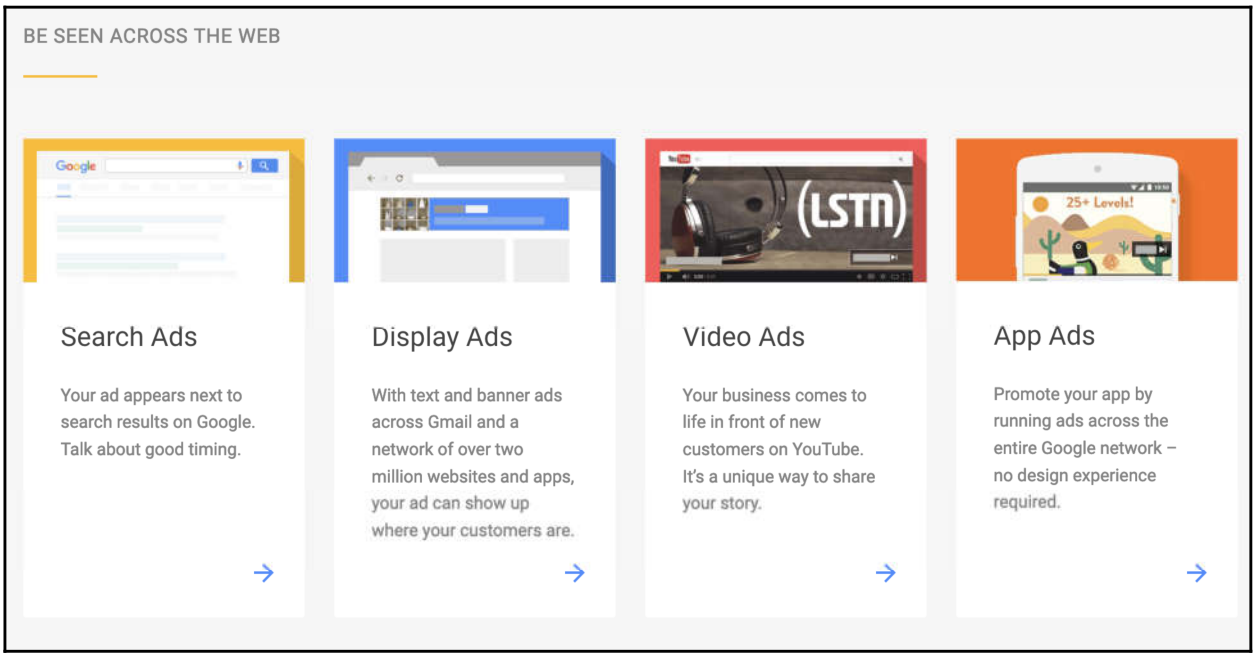
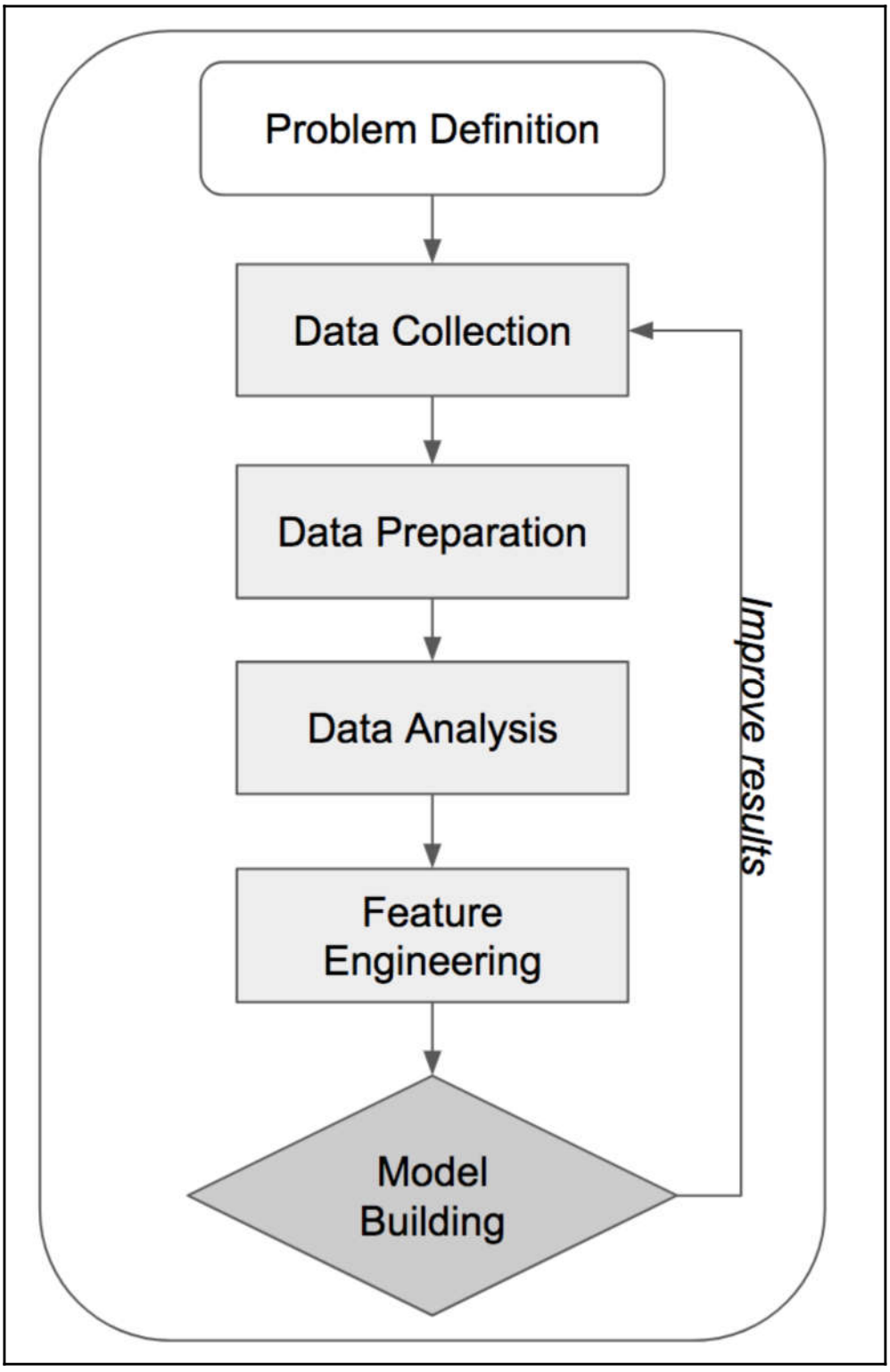
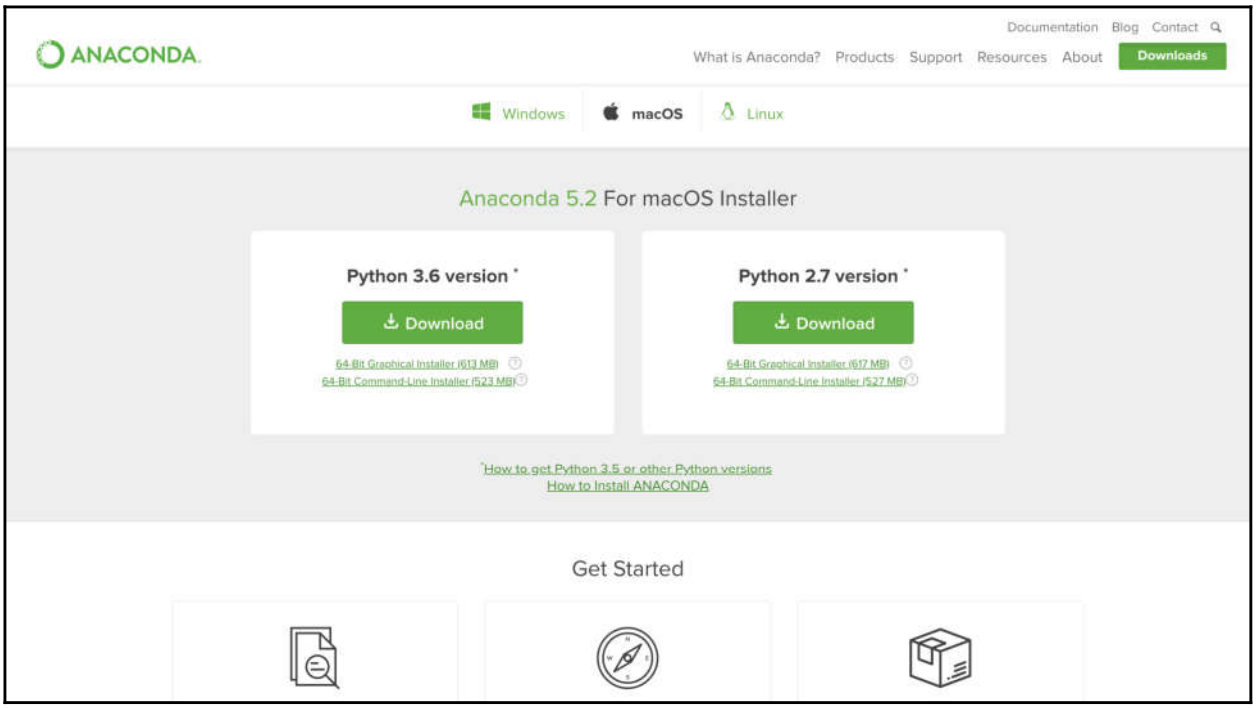
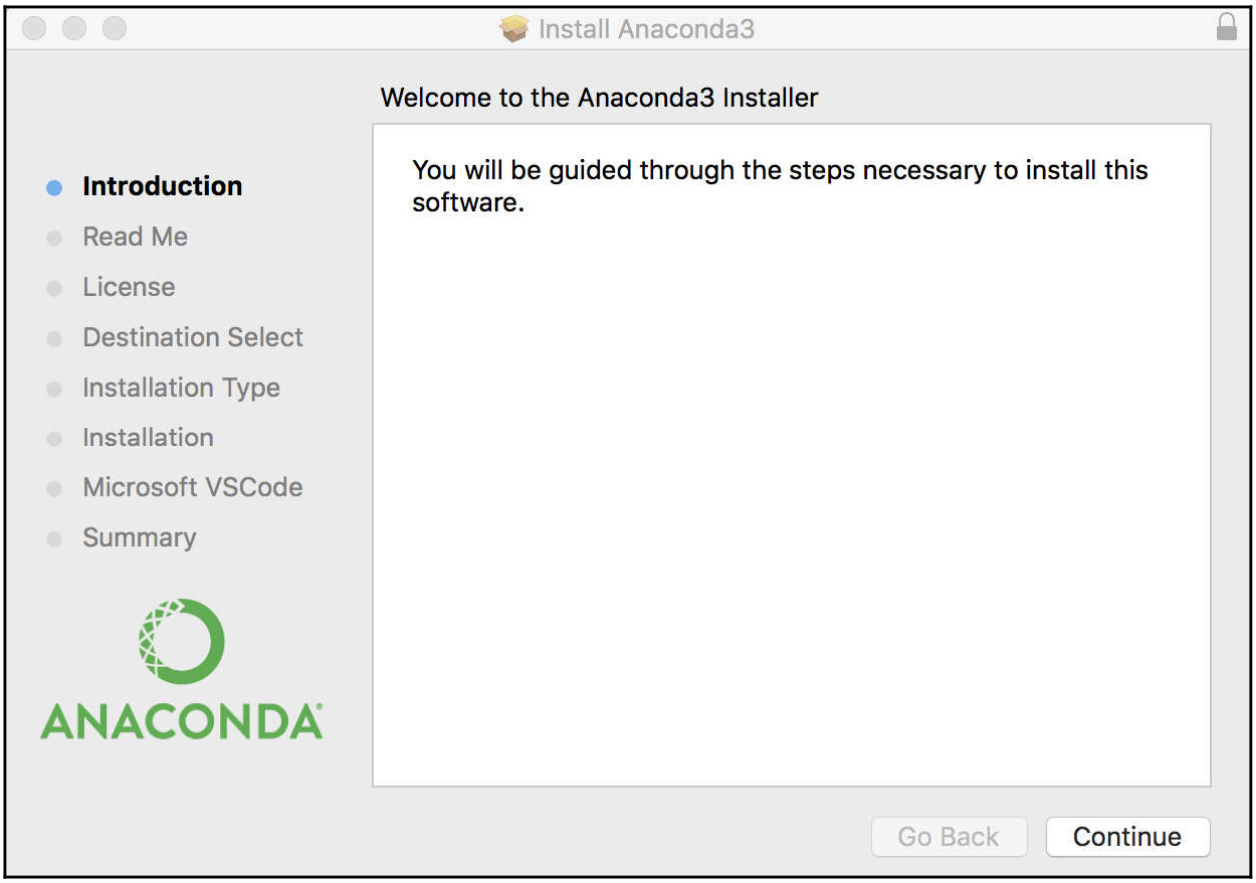
Chapter 1: Data Science and Marketing

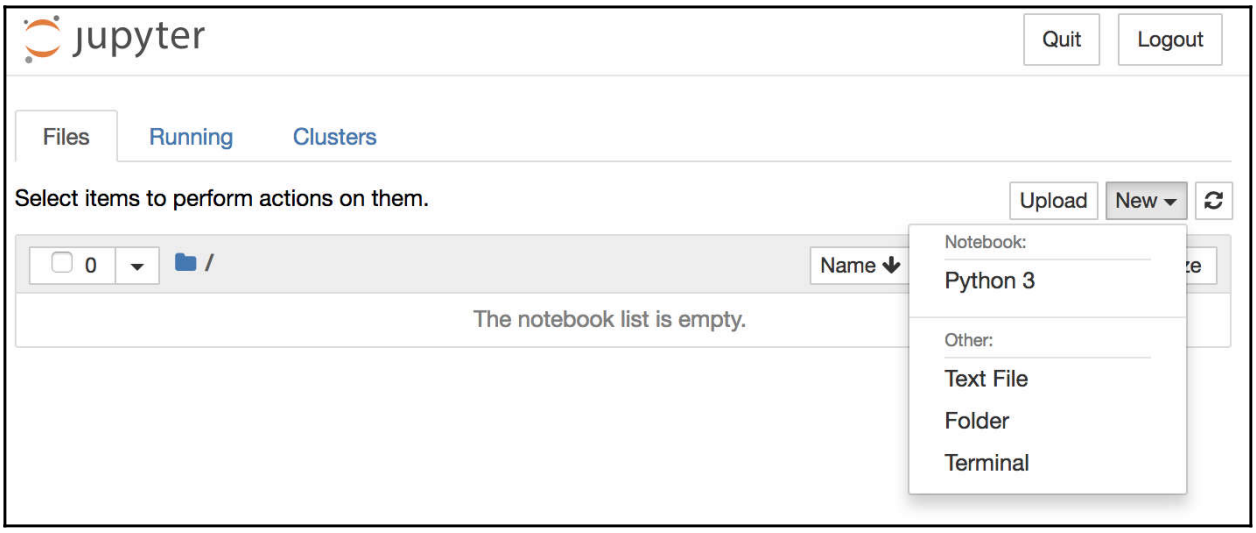


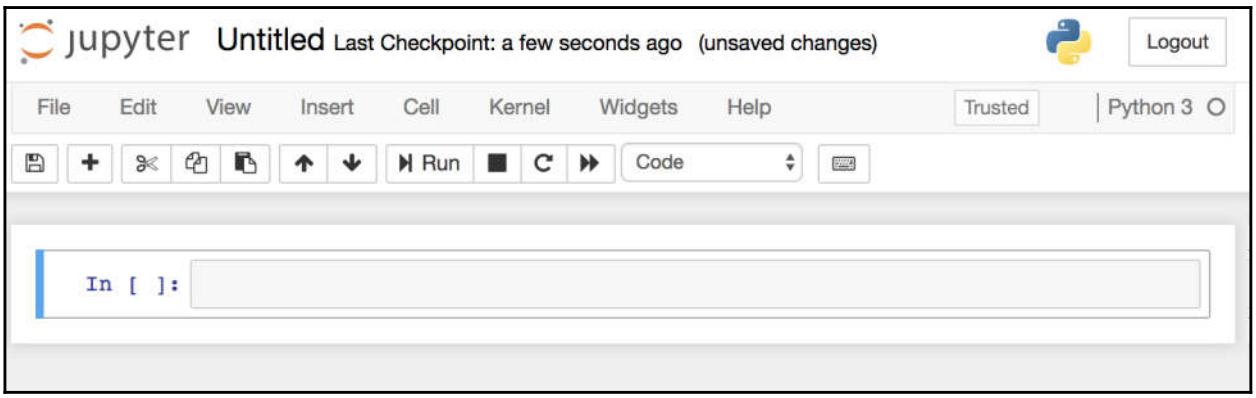

Problem Definition
Data Collection
Data Preparation
Data Analysis
Feature Engineering
Model Building
Improve results


Documentation Blog Contact
ANACONDA.
What is Anaconda? Products Support Resources About
Downloads
Windows
macOS
Linux
Anaconda 5.2 For macOS Installer
Python 3.6 version *
Download
64-Bit Graphical Installer (613 MB)
64-Bit Command-Line Installer (523 MB)
Python 2.7 version *
Download
64-Bit Graphical Installer (617 MB)
64-Bit Command-Line Installer (527 MB)
*How to get Python 3.5 or other Python versions
How to Install ANACONDA
Get Started




```
Yoons-MBP:python yoonhyuph$ jupyter notebook
[I 16:21:39.646 NotebookApp] JupyterLab beta preview extension loaded from /anaconda3/lib/python3.6/site-packages/jupyterlab
[I 16:21:39.646 NotebookApp] JupyterLab application directory is /anaconda3/share/jupyter/lab
[I 16:21:39.656 NotebookApp] Serving notebooks from local directory: /Users/yoonhyuph/Documents/data-science-for-marketing/ch.1/python
[I 16:21:39.656 NotebookApp] 0 active kernels
[I 16:21:39.656 NotebookApp] The Jupyter Notebook is running at:
[I 16:21:39.656 NotebookApp] http://localhost:8888/?token=c1ae6b23458efbfc677d199954be6124bb03ec8399775408
[I 16:21:39.656 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 16:21:39.657 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8888/?token=c1ae6b23458efbfc677d199954be6124bb03ec8399775408&token=c1ae6b23458efbfc677d199954be6124bb03ec8399775408
[I 16:21:39.808 NotebookApp] Accepting one-time-token-authenticated connection from ::1
```


jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
The notebook list is empty.
Notebook:
Python 3
Other:
Text File
Folder
Terminal


jupyter
Untitled
Last Checkpoint: a few seconds ago
(unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
In [ ]:

In [1]:
```
import numpy as np
from sklearn.linear_model import LogisticRegression
```

In [2]:
```
input_data = np.array([
    [0, 0],
    [0.25, 0.25],
    [0.5, 0.5],
    [1, 1],
])
```

In [3]:
```
output_data = [
    0,
    0,
    1,
    1
]
```

In [4]:
```
logit_model = LogisticRegression()
```

In [5]:
```
logit_model.fit(input_data, output_data)
```

Out[5]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

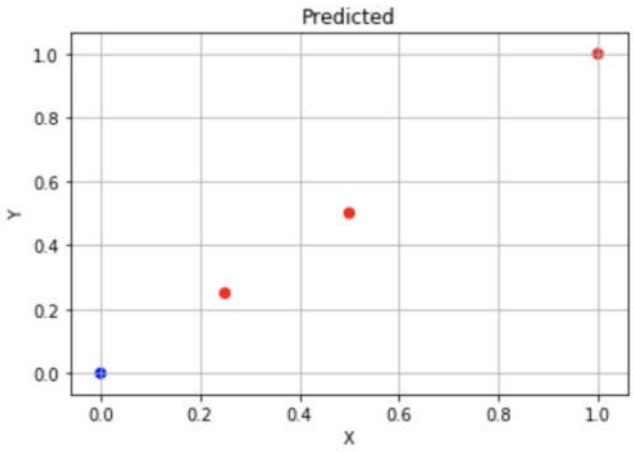
In [6]:
```
logit_model.coef_
```

Out[6]:
```
array([[0.43001235, 0.43001235]])
```

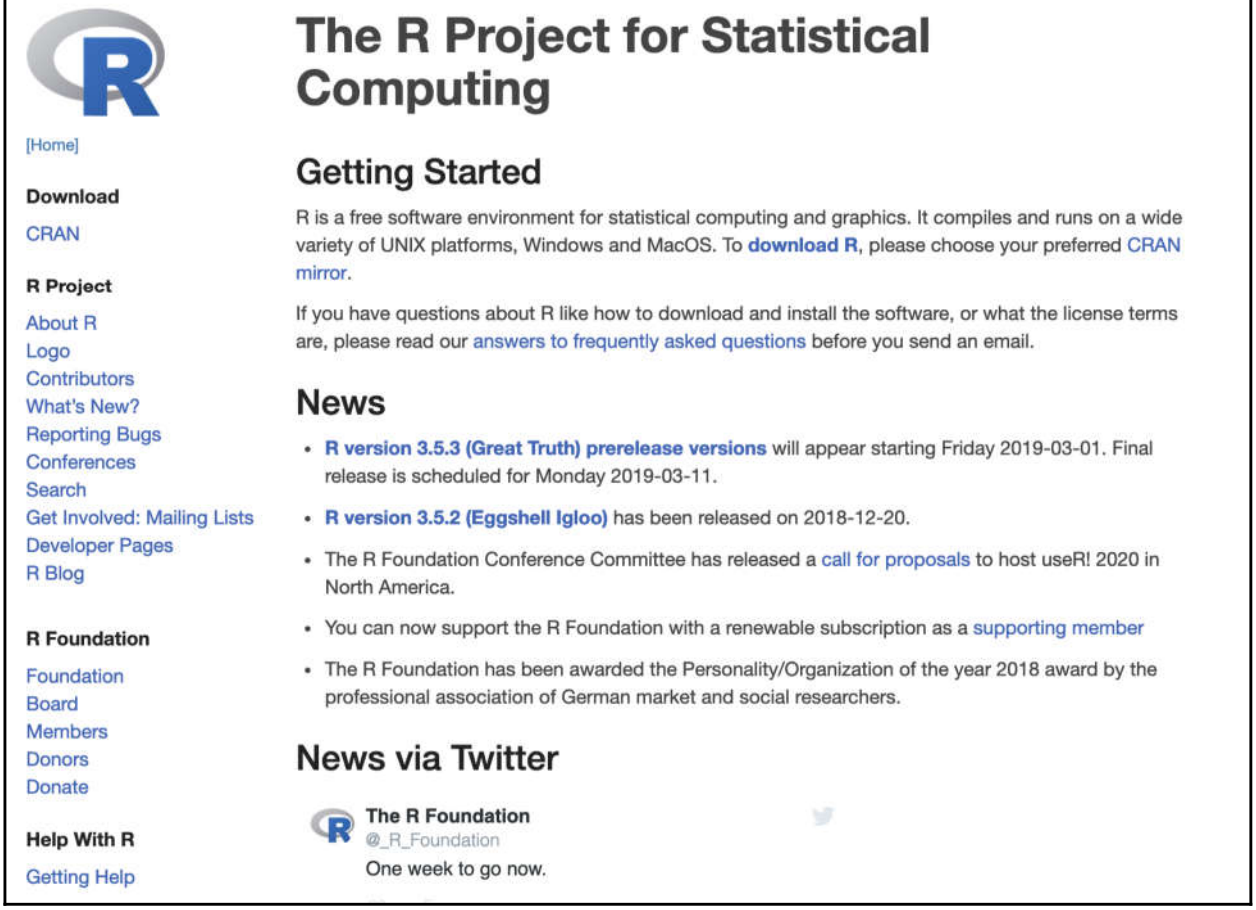
In [7]:
```
logit_model.intercept_
```

Out[7]:
```
array([-0.18498028])
```

In [13]:
```
plt.scatter(
    x=input_data[:,0],
    y=input_data[:,1],
    color=[('red' if x == 1 else 'blue') for x in predicted_output]
)
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Predicted')
plt.grid()
plt.show()
```



In [ ]:



[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting Bugs
Conferences
Search
Get Involved: Mailing Lists
Developer Pages
R Blog

R Foundation

Foundation
Board
Members
Donors
Donate

Help With R

Getting Help

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

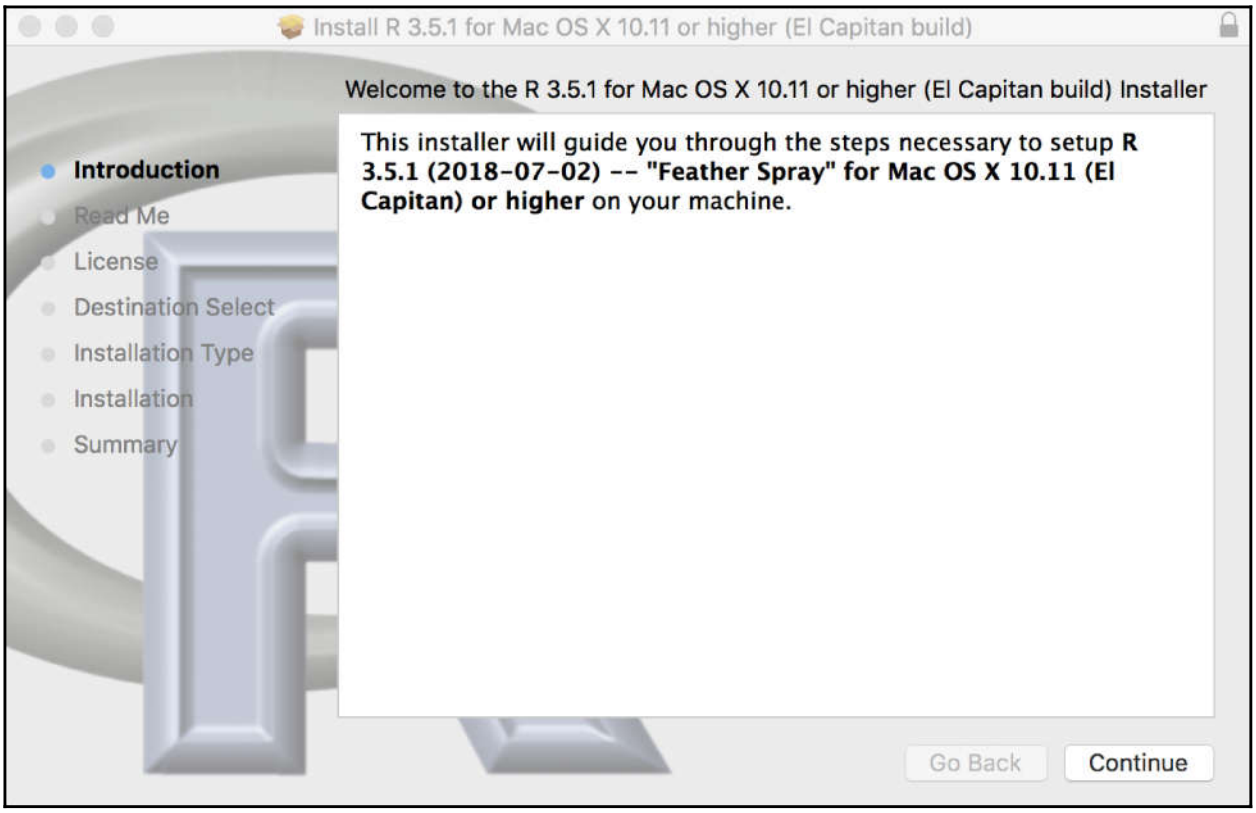
If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

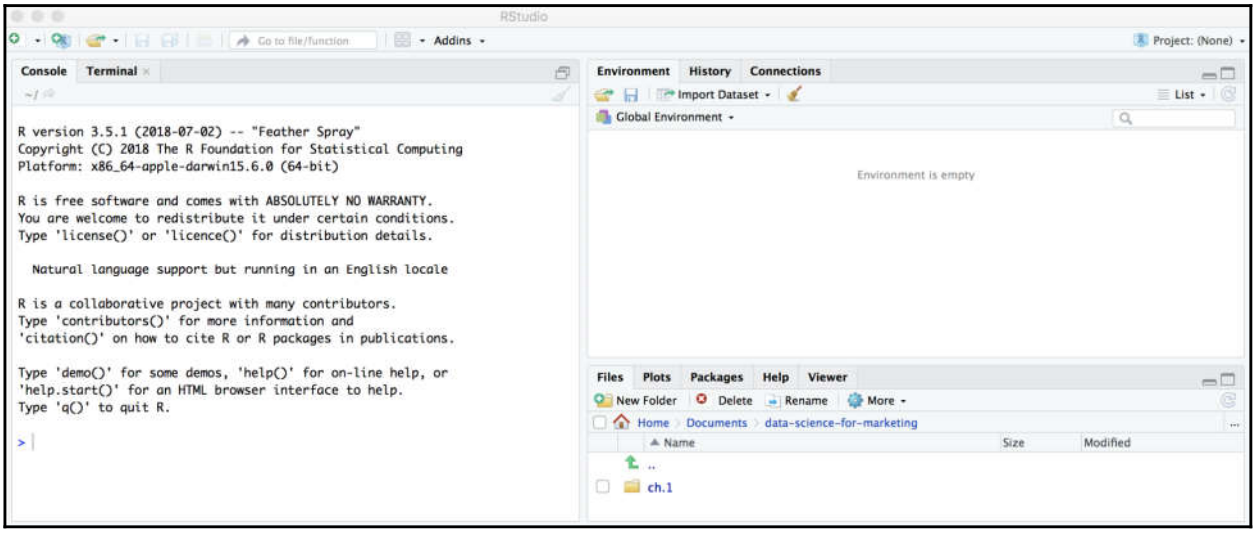
News

- **R version 3.5.3 (Great Truth) prerelease versions** will appear starting Friday 2019-03-01. Final release is scheduled for Monday 2019-03-11.
- **R version 3.5.2 (Eggshell Igloo)** has been released on 2018-12-20.
- The R Foundation Conference Committee has released a call for proposals to host useR! 2020 in North America.
- You can now support the R Foundation with a renewable subscription as a supporting member
- The R Foundation has been awarded the Personality/Organization of the year 2018 award by the professional association of German market and social researchers.

News via Twitter

The R Foundation
@_R_Foundation
One week to go now.


Install R 3.5.1 for Mac OS X 10.11 or higher (El Capitan build)
Welcome to the R 3.5.1 for Mac OS X 10.11 or higher (El Capitan build) Installer
Introduction
Read Me
License
Destination Select
Installation Type
Installation
Summary
This installer will guide you through the steps necessary to setup R 3.5.1 (2018-07-02) -- "Feather Spray" for Mac OS X 10.11 (El Capitan) or higher on your machine.
Go Back
Continue

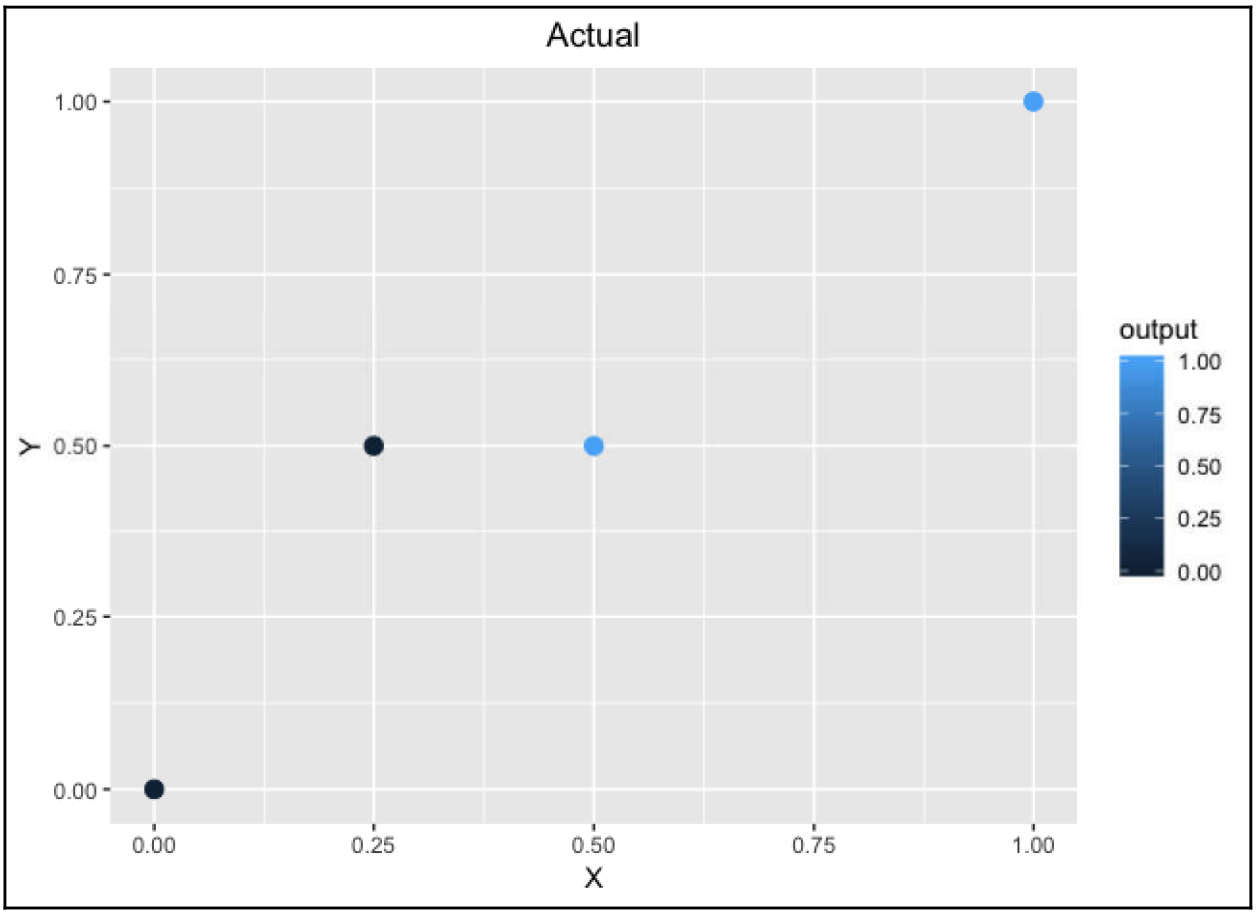

RStudio
Addins
Project: (None)
Console
Terminal
R version 3.5.1 (2018-07-02) -- "Feather Spray"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Documents
data-science-for-marketing
Name
Size
Modified
ch.1

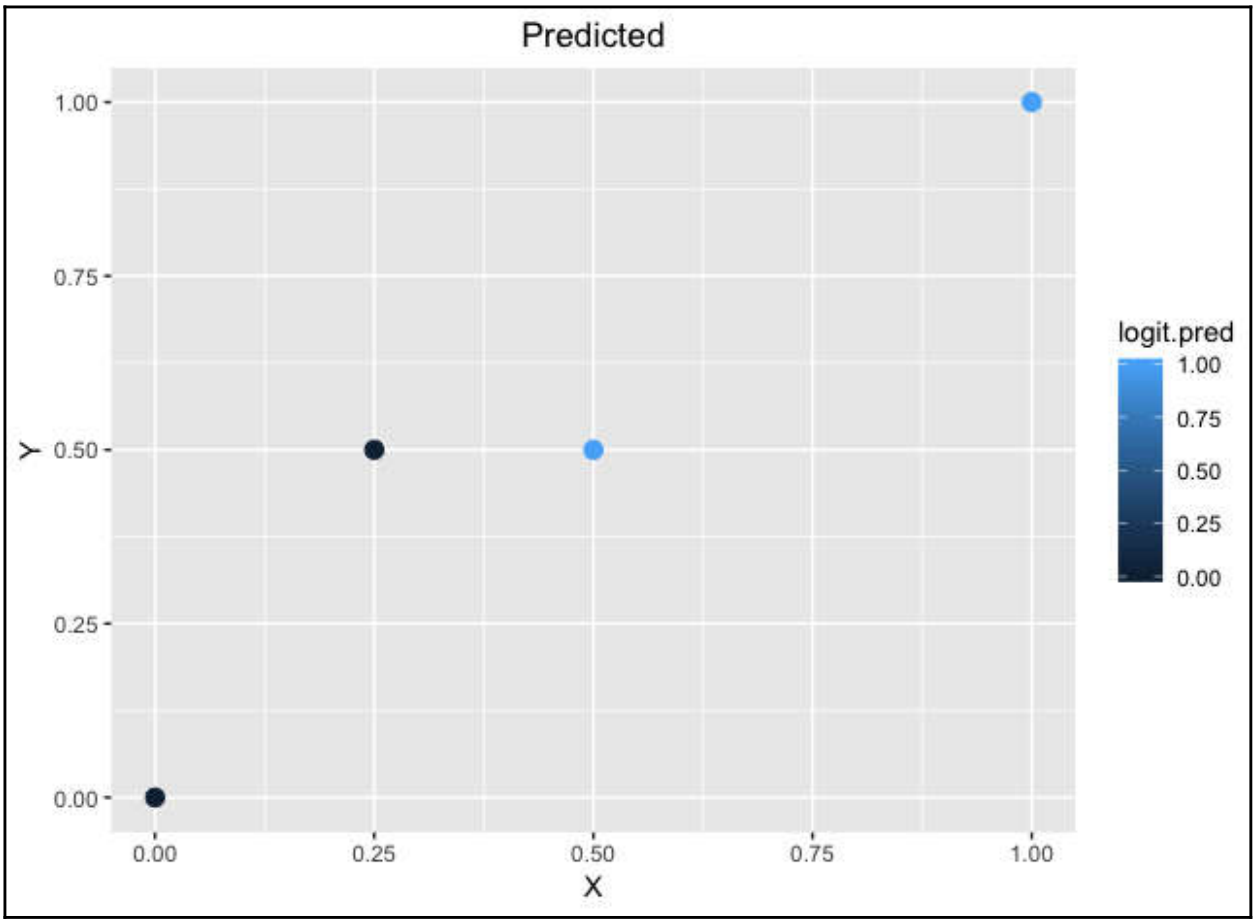
```
> data
     X   Y output
1 0.00 0.0      0
2 0.25 0.5      0
3 0.50 0.5      1
4 1.00 1.0      1
```

```
> # Show Fitted Results
> summary(logit.fit)

Call:
glm(formula = output ~ X + Y, family = binomial, data = data)

Deviance Residuals:
         1           2           3           4
-1.140e-05  -6.547e-06   1.516e-05   2.110e-08

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    -23.46   75250.33       0        1
X              189.81  570847.71       0        1
Y              -97.12  556850.90       0        1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 5.5452e+00  on 3  degrees of freedom
Residual deviance: 4.0252e-10  on 1  degrees of freedom
AIC: 6

Number of Fisher Scoring iterations: 23
```

```
> # Plotting Library
> require(ggplot2)
Loading required package: ggplot2
Warning message:
In library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE,  :
  there is no package called 'ggplot2'
```

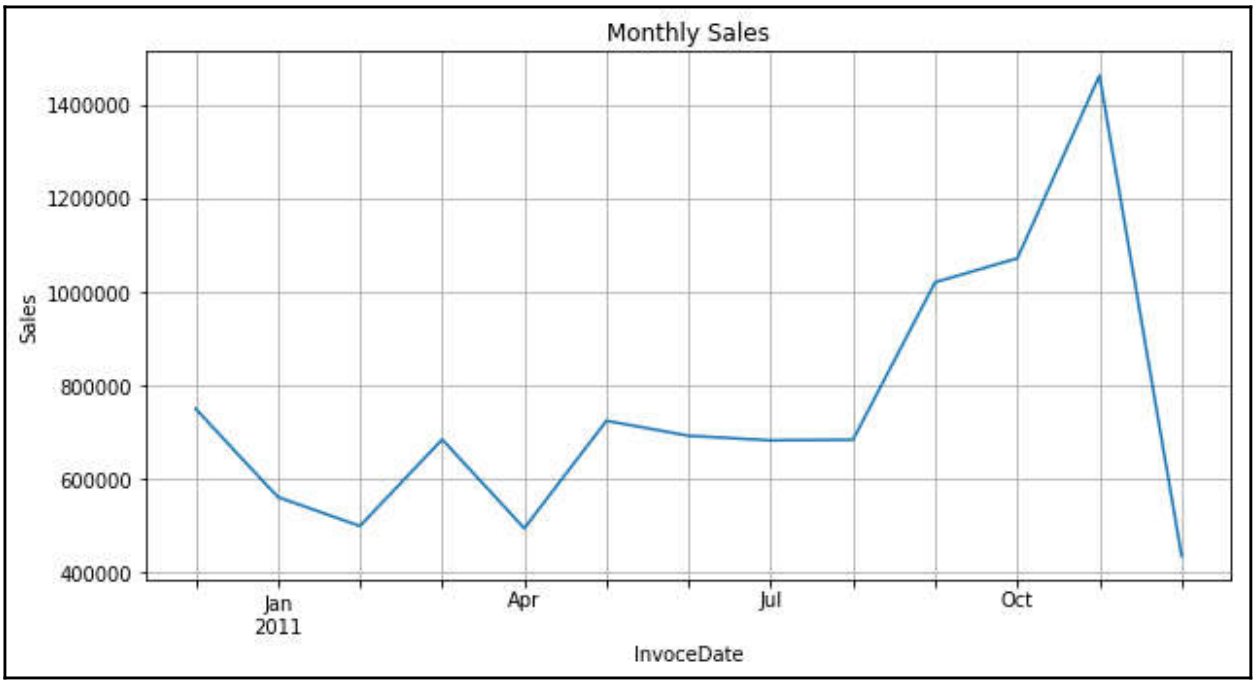
```
> install.packages("ggplot2")
also installing the dependencies ‘colorspace’, ‘RColorBrewer’, ‘dichromat’, ‘munsell’, ‘labeli
ng’, ‘digest’, ‘gtable’, ‘lazyeval’, ‘plyr’, ‘reshape2’, ‘scales’, ‘viridisLite’, ‘withr’

trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/colorspace_1.3-2.tgz'
Content type 'application/x-gzip' length 443683 bytes (433 KB)
==================================================
downloaded 433 KB
```




Predicted
1.00
0.75
0.50
0.25
0.00
Y
0.00
0.25
0.50
0.75
1.00
X
logit.pred
1.00
0.75
0.50
0.25
0.00

Chapter 2: Key Performance Indicators and Visualizations

	Date	TotalSales
1	2010-12-01	748957.020
2	2011-01-01	560000.260
3	2011-02-01	498062.650
4	2011-03-01	683267.080
5	2011-04-01	493207.121
6	2011-05-01	723333.510
7	2011-06-01	691123.120
8	2011-07-01	681300.111
9	2011-08-01	682680.510
10	2011-09-01	1019687.622
11	2011-10-01	1070704.670
12	2011-11-01	1461756.250
13	2011-12-01	433668.010



Campaign	Cost	Customers Acquired	CPA	Sales	Sales per Customer	Value of Campaign
Happy Hour Event	$25,000	40	$625	$50,000	$1,250	$25,000
Webinar	$2,000	10	$200	$5,000	$500	$3,000
Radio Commercial	$7,000	50	$140	$6,000	$120	($1,000)

```
import pandas as pd
```

```
df = pd.read_csv('../data/bank-additional-full.csv', sep=';')
```

```
df['conversion'] = df['y'].apply(lambda x: 1 if x == 'yes' else 0)
```

```
df.head()
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	pdays	previous	poutcome	emp.var.rate	cons.price.idx
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	...	999	0	nonexistent	1.1	93.994
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	...	999	0	nonexistent	1.1	93.994
2	37	services	married	high.school	no	yes	no	telephone	may	mon	...	999	0	nonexistent	1.1	93.994
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	...	999	0	nonexistent	1.1	93.994
4	56	services	married	high.school	no	no	yes	telephone	may	mon	...	999	0	nonexistent	1.1	93.994

5 rows × 22 columns

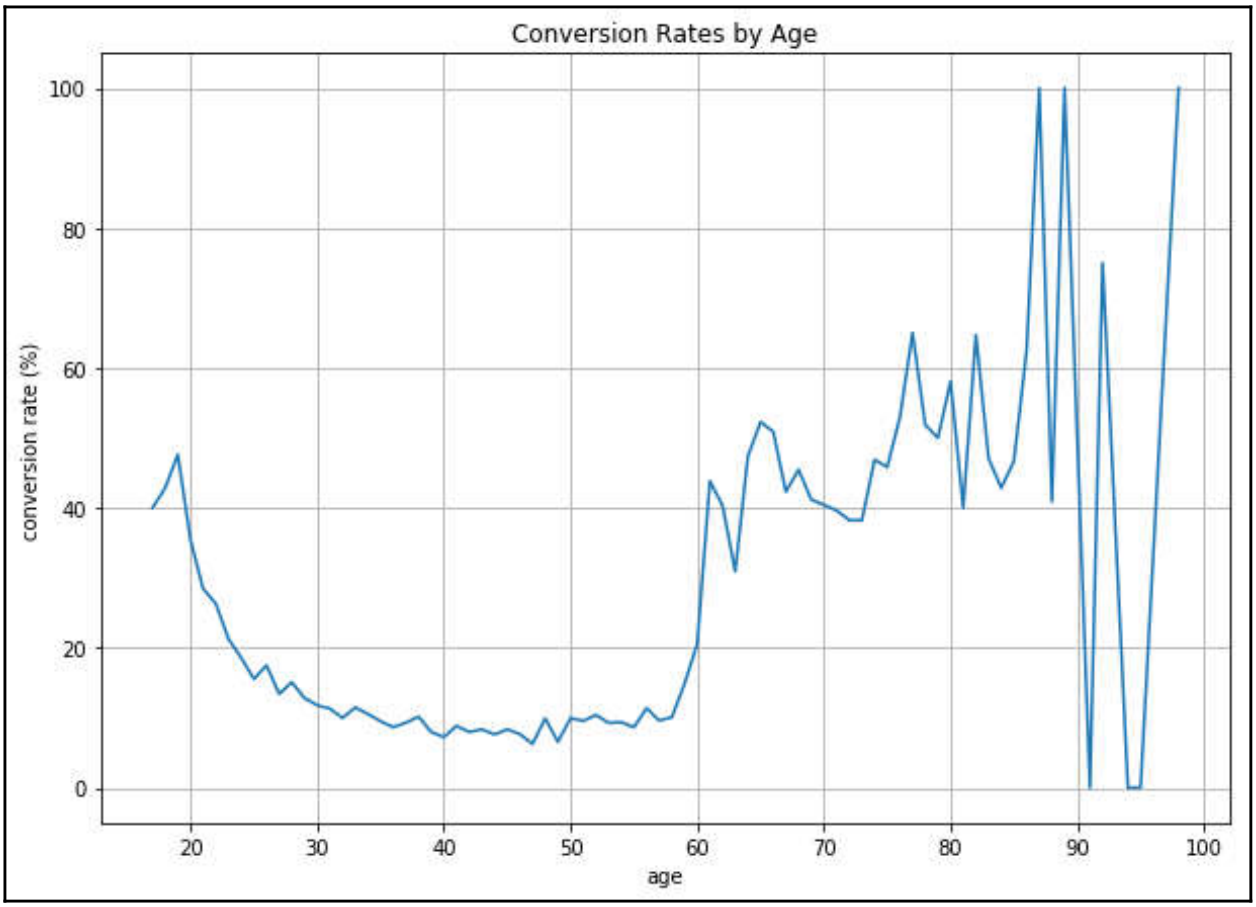
```
print('total conversions: %i out of %i' % (df.conversion.sum(), df.shape[0]))
```

```
total conversions: 4640 out of 41188
```

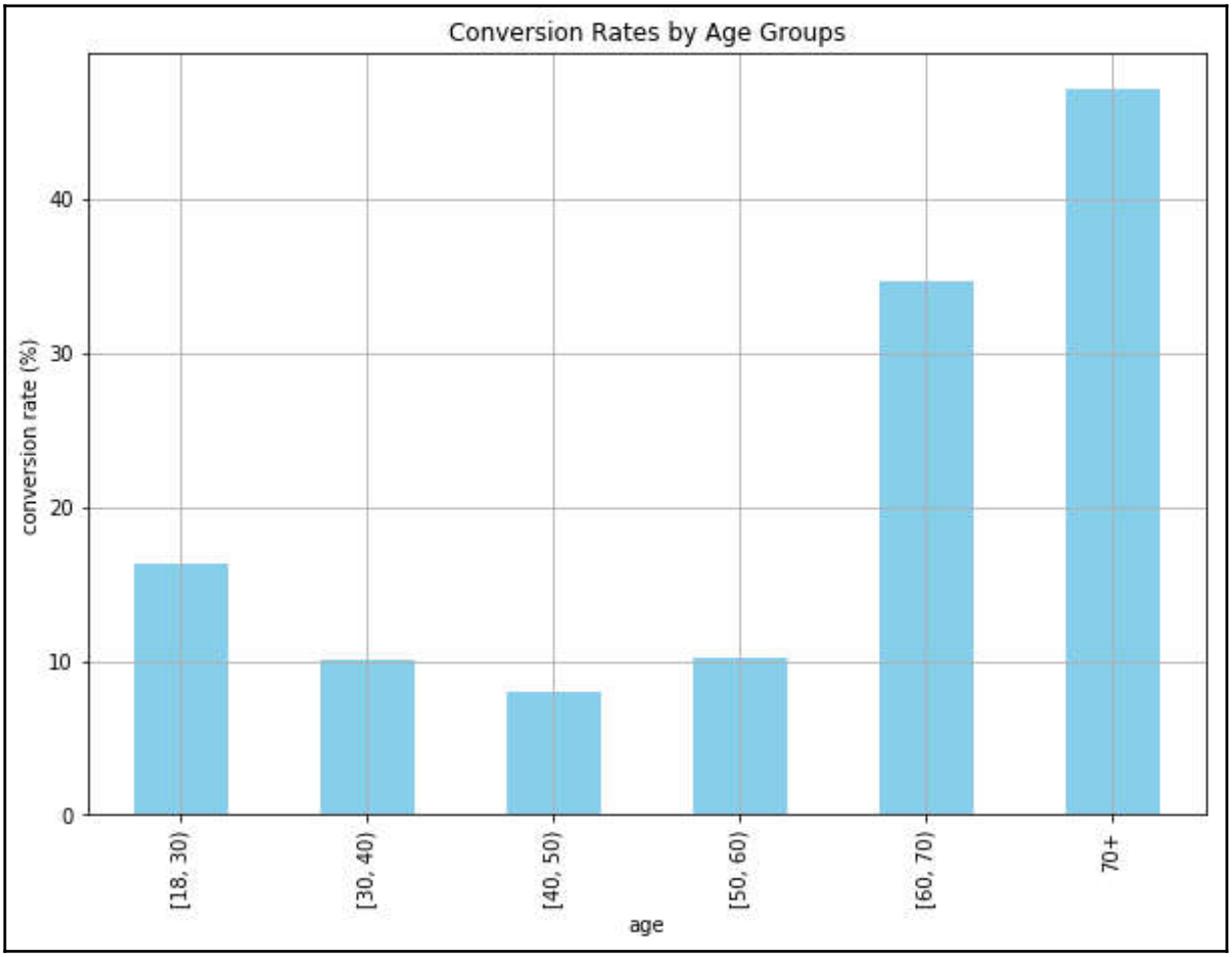
```
print('conversion rate: %0.2f%%' % (df.conversion.sum() / df.shape[0] * 100.0))
```

```
conversion rate: 11.27%
```

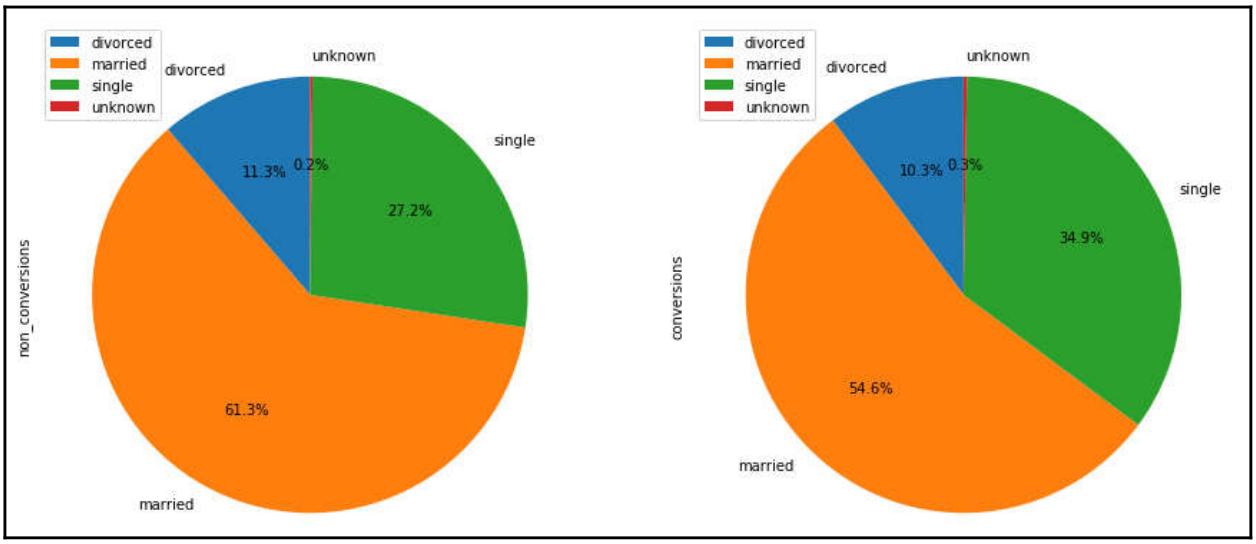
age	conversion
17	40.000000
18	42.857143
19	47.619048
20	35.384615
21	28.431373
22	26.277372
23	21.238938
24	18.574514
25	15.551839


Conversion Rates by Age
conversion rate (%)
age
0
20
40
60
80
100
20
30
40
50
60
70
80
90
100

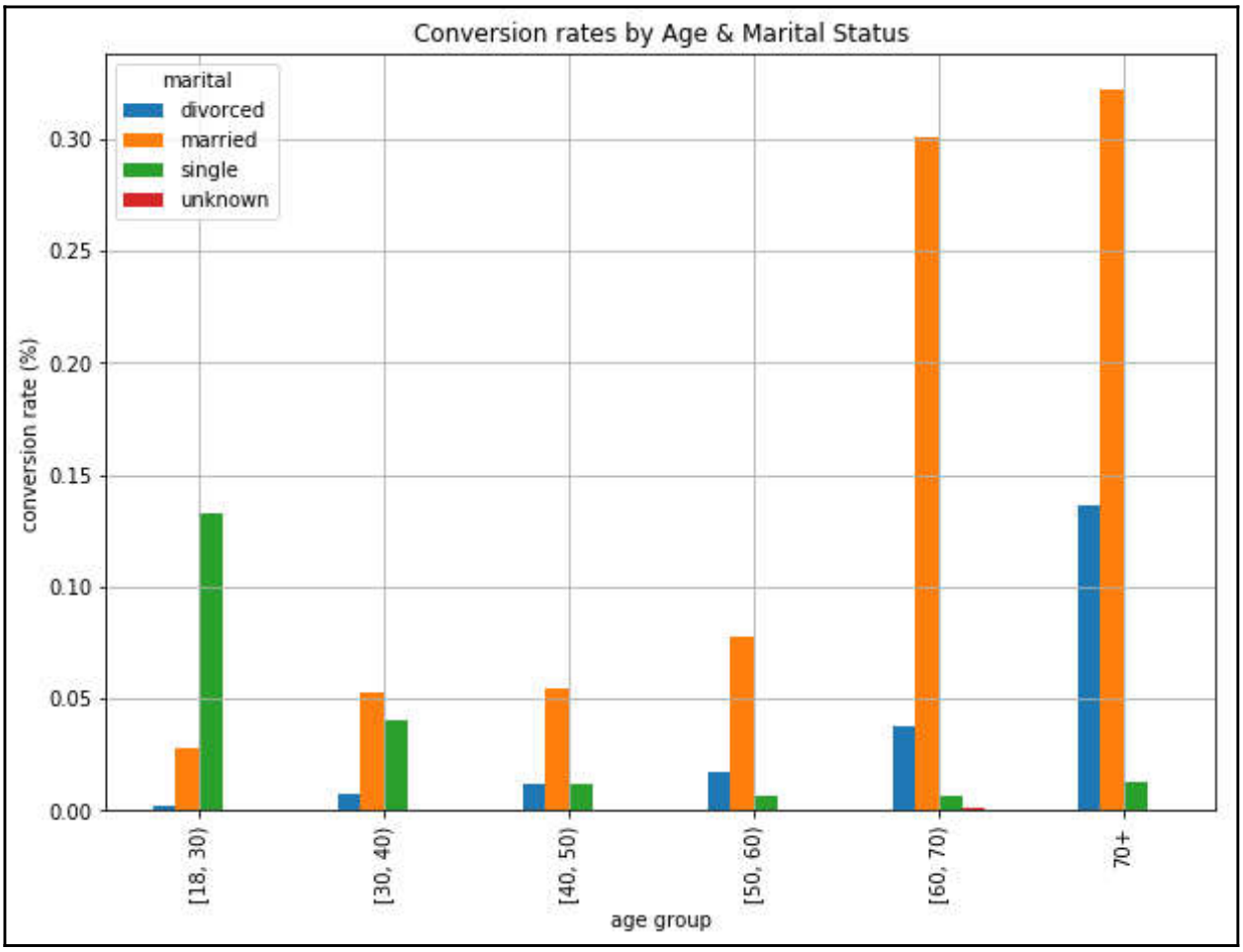
age_group	conversion
70+	47.121535
[18, 30)	16.263891
[30, 40)	10.125162
[40, 50)	7.923238
[50, 60)	10.157389
[60, 70)	34.668508

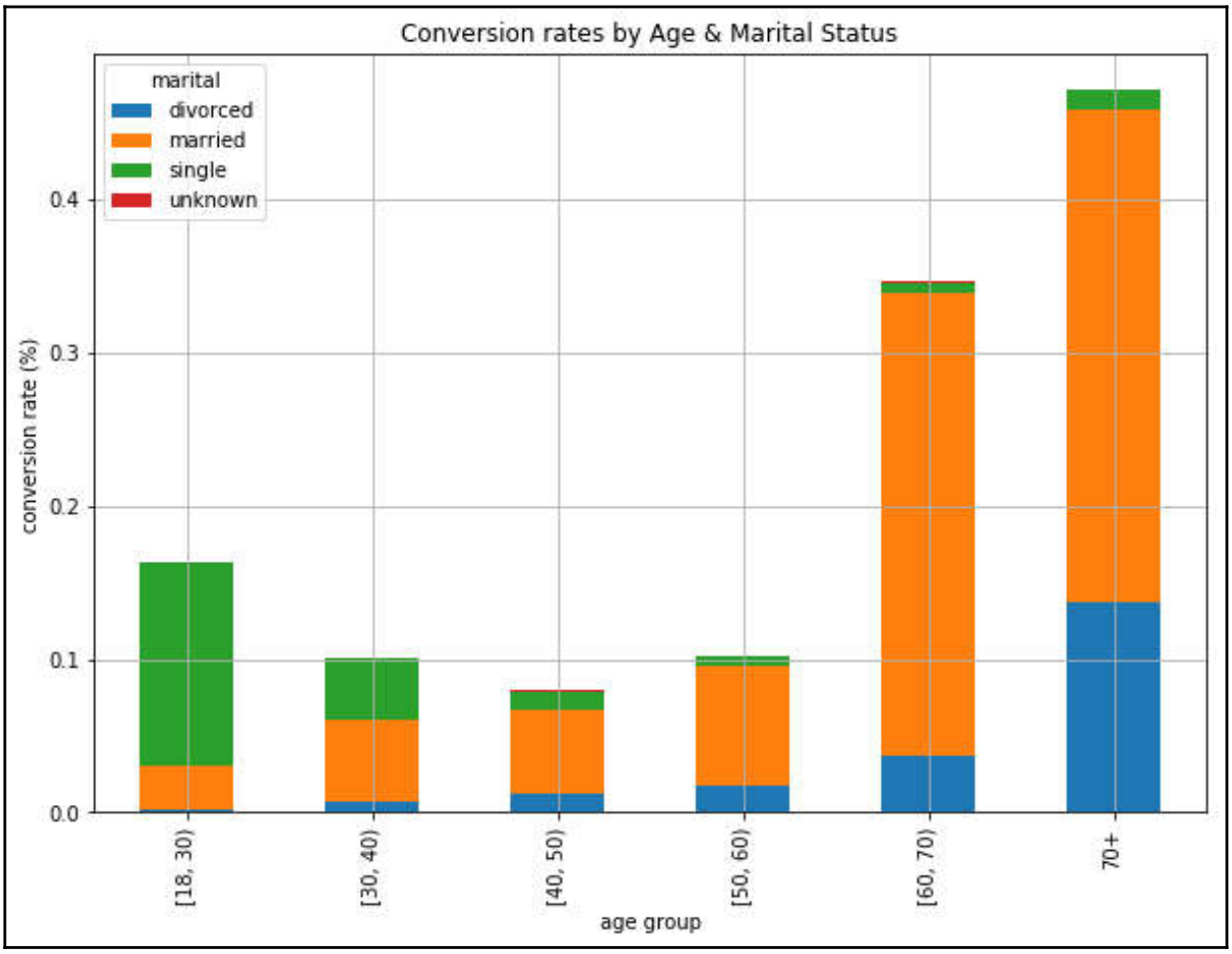

Conversion Rates by Age Groups
conversion rate (%)
0
10
20
30
40
[18, 30)
[30, 40)
[40, 50)
[50, 60)
[60, 70)
70+
age

marital	non_conversions	conversions
divorced	4136	476
married	22396	2532
single	9948	1620
unknown	68	12



marital age_group	divorced	married	single	unknown
70+	0.136461	0.321962	0.012793	0.000000
[18, 30)	0.002117	0.027871	0.132475	0.000176
[30, 40)	0.007557	0.052958	0.040383	0.000354
[40, 50)	0.011970	0.054627	0.012350	0.000285
[50, 60)	0.017342	0.077674	0.006412	0.000146
[60, 70)	0.037293	0.301105	0.006906	0.001381

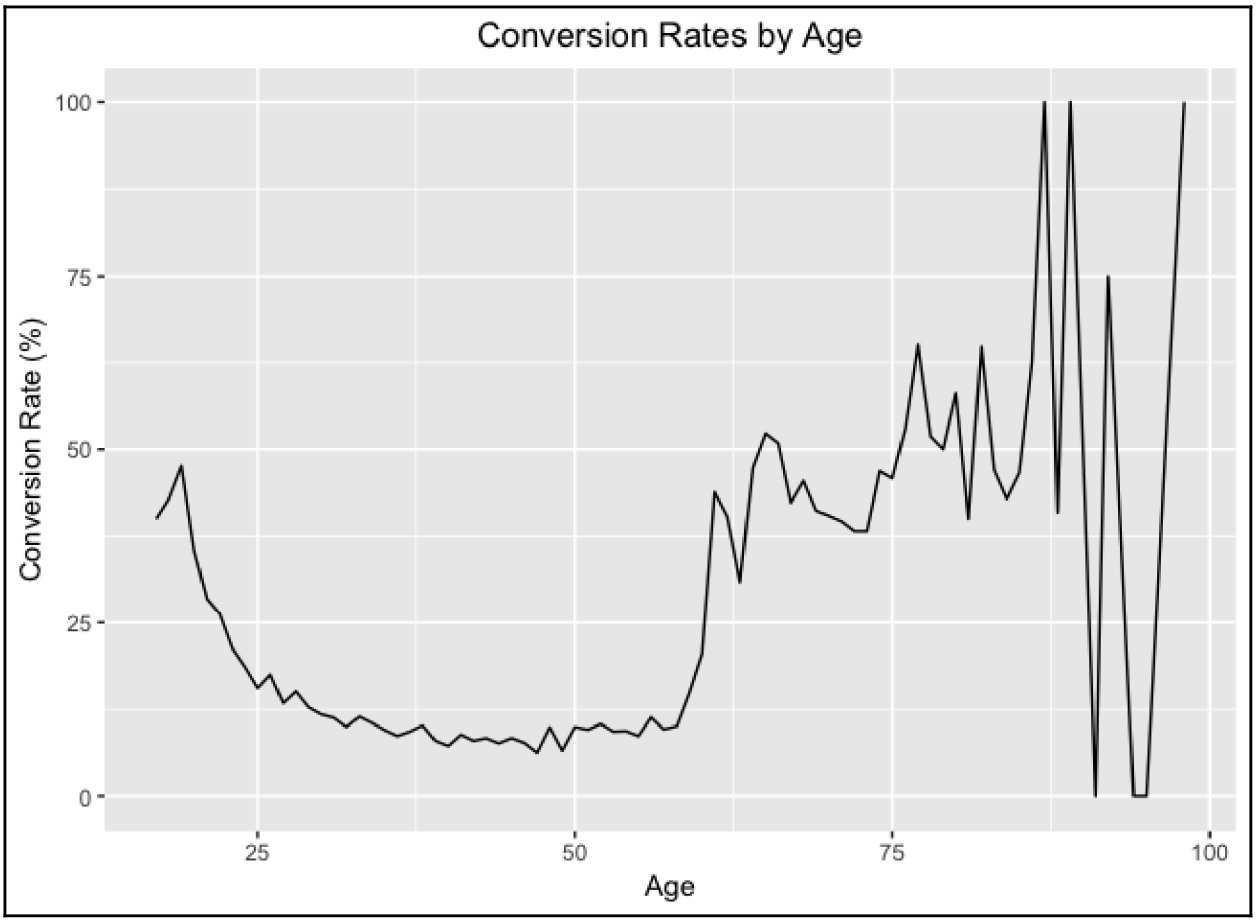

Conversion rates by Age & Marital Status
marital
divorced
married
single
unknown
conversion rate (%)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
[18, 30)
[30, 40)
[40, 50)
[50, 60)
[60, 70)
70+
age group



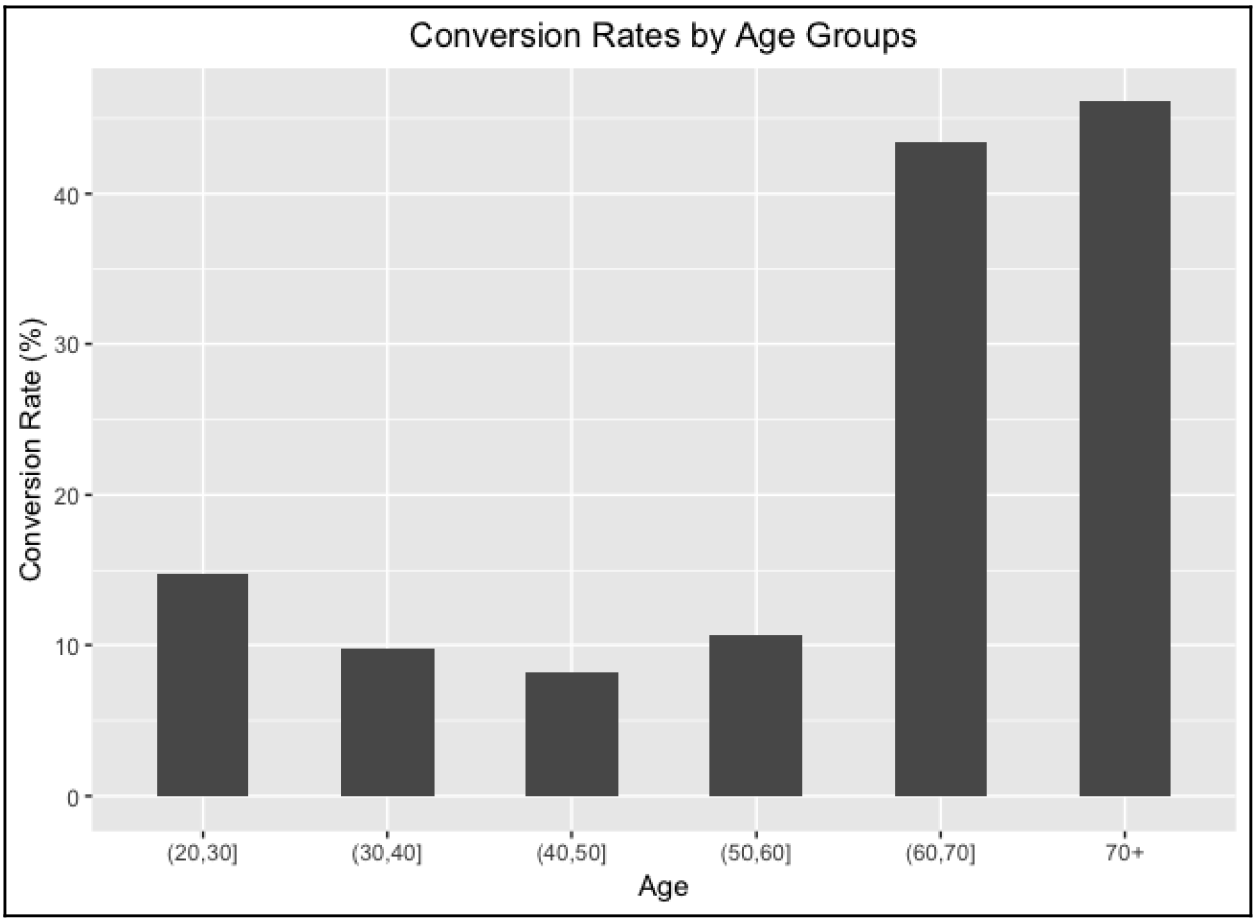
	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome
1	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	261	1	999	0	nonexistent
2	57	services	married	high.school	unknown	no	no	telephone	may	mon	149	1	999	0	nonexistent
3	37	services	married	high.school	no	yes	no	telephone	may	mon	226	1	999	0	nonexistent
4	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	151	1	999	0	nonexistent
5	56	services	married	high.school	no	no	yes	telephone	may	mon	307	1	999	0	nonexistent
6	45	services	married	basic.9y	unknown	no	no	telephone	may	mon	198	1	999	0	nonexistent
7	59	admin.	married	professional.course	no	no	no	telephone	may	mon	139	1	999	0	nonexistent
8	41	blue-collar	married	unknown	unknown	no	no	telephone	may	mon	217	1	999	0	nonexistent
9	24	technician	single	professional.course	no	yes	no	telephone	may	mon	380	1	999	0	nonexistent

```
> #### 1. Aggregate Conversion Rate ####
> sprintf("total conversions: %i out of %i", sum(conversionsDF$conversion), nrow(conversionsDF))
[1] "total conversions: 4640 out of 41188"
> sprintf("conversion rate: %0.2f%%", sum(conversionsDF$conversion)/nrow(conversionsDF)*100.0)
[1] "conversion rate: 11.27%"
```

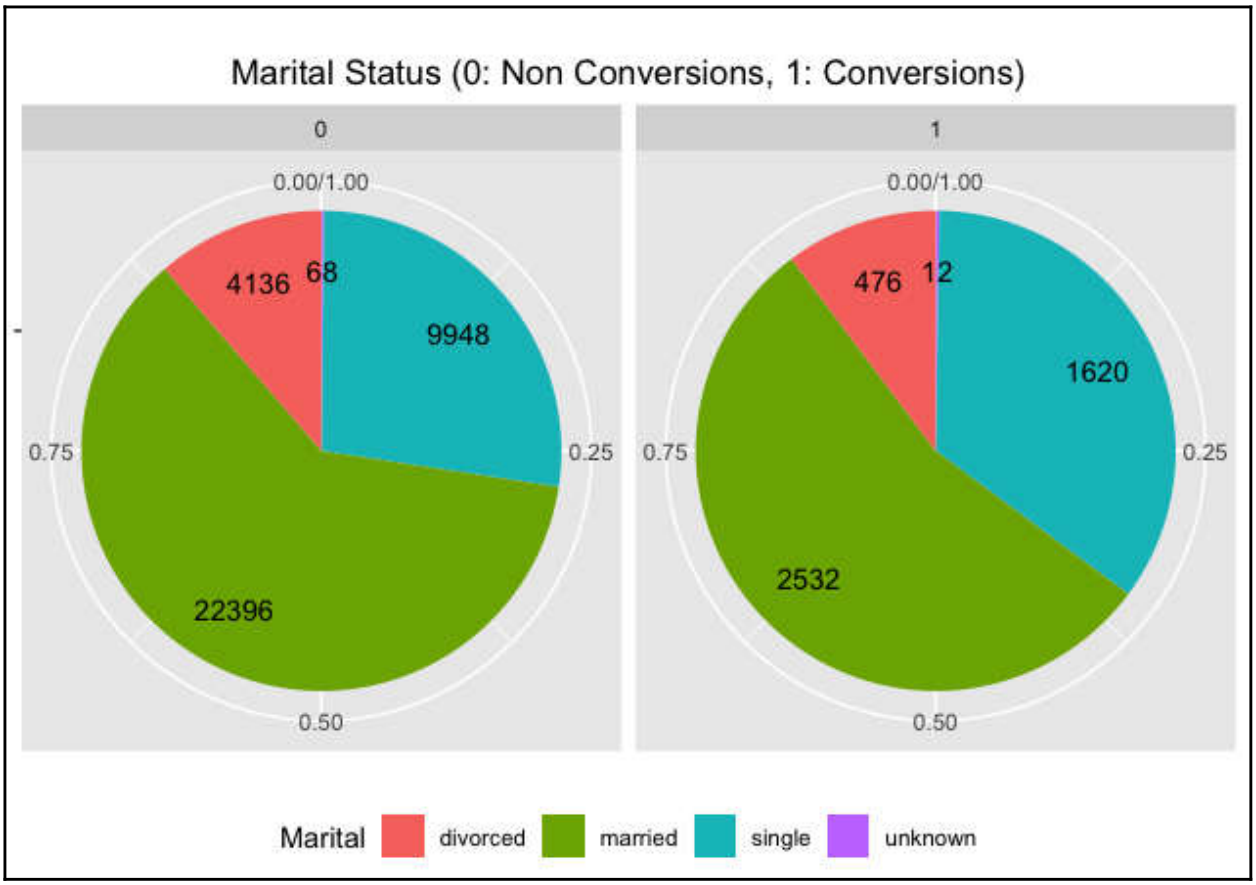
	Age	TotalCount	NumConversions	ConversionRate
1	17	5	2	40.000000
2	18	28	12	42.857143
3	19	42	20	47.619048
4	20	65	23	35.384615
5	21	102	29	28.431373
6	22	137	36	26.277372
7	23	226	48	21.238938
8	24	463	86	18.574514
9	25	598	93	15.551839
10	26	698	122	17.478510
11	27	851	114	13.396005
12	28	1001	151	15.084915
13	29	1453	186	12.801101
14	30	1714	202	11.785298
15	31	1947	220	11.299435
16	32	1846	184	9.967497
17	33	1833	210	11.456628
18	34	1745	184	10.544413



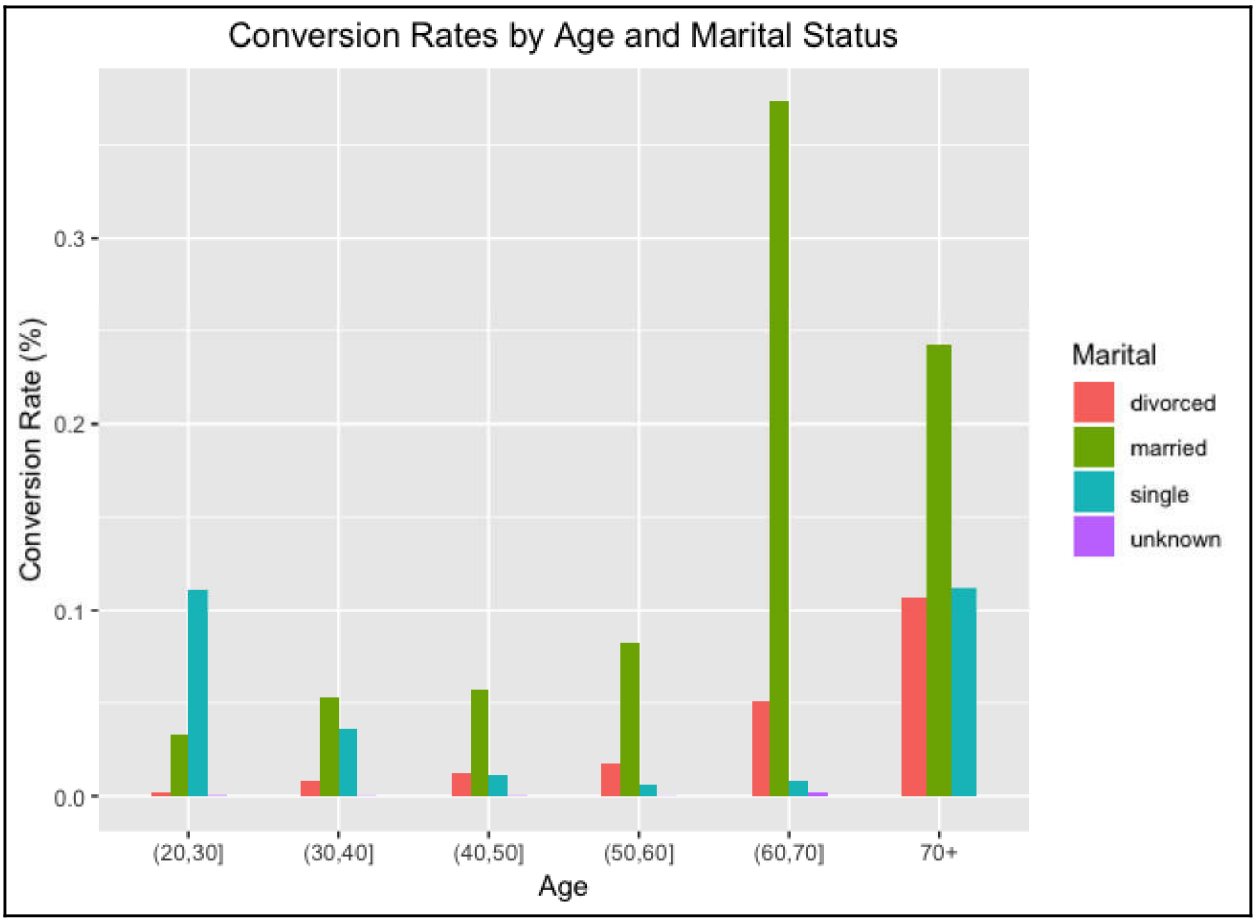
	AgeGroup	TotalCount	NumConversions	ConversionRate
1	(20,30]	7243	1067	14.731465
2	(30,40]	16385	1597	9.746720
3	(40,50]	10240	837	8.173828
4	(50,60]	6270	668	10.653907
5	(60,70]	488	212	43.442623
6	70+	562	259	46.085409

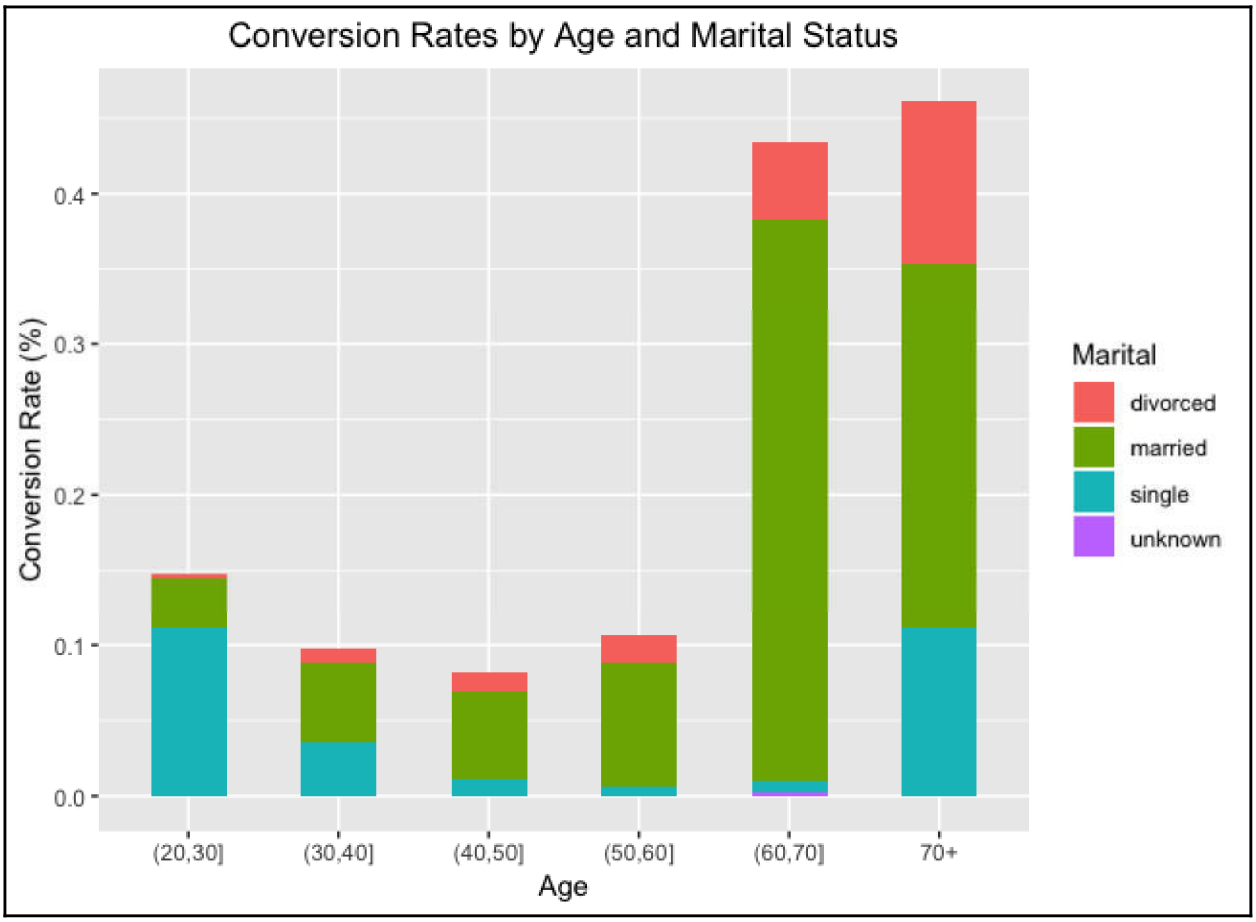

Conversion Rates by Age Groups
Conversion Rate (%)
0
10
20
30
40
(20,30]
(30,40]
(40,50]
(50,60]
(60,70]
70+
Age

	Marital	Conversion	Count
1	divorced	0	4136
2	divorced	1	476
3	married	0	22396
4	married	1	2532
5	single	0	9948
6	single	1	1620
7	unknown	0	68
8	unknown	1	12

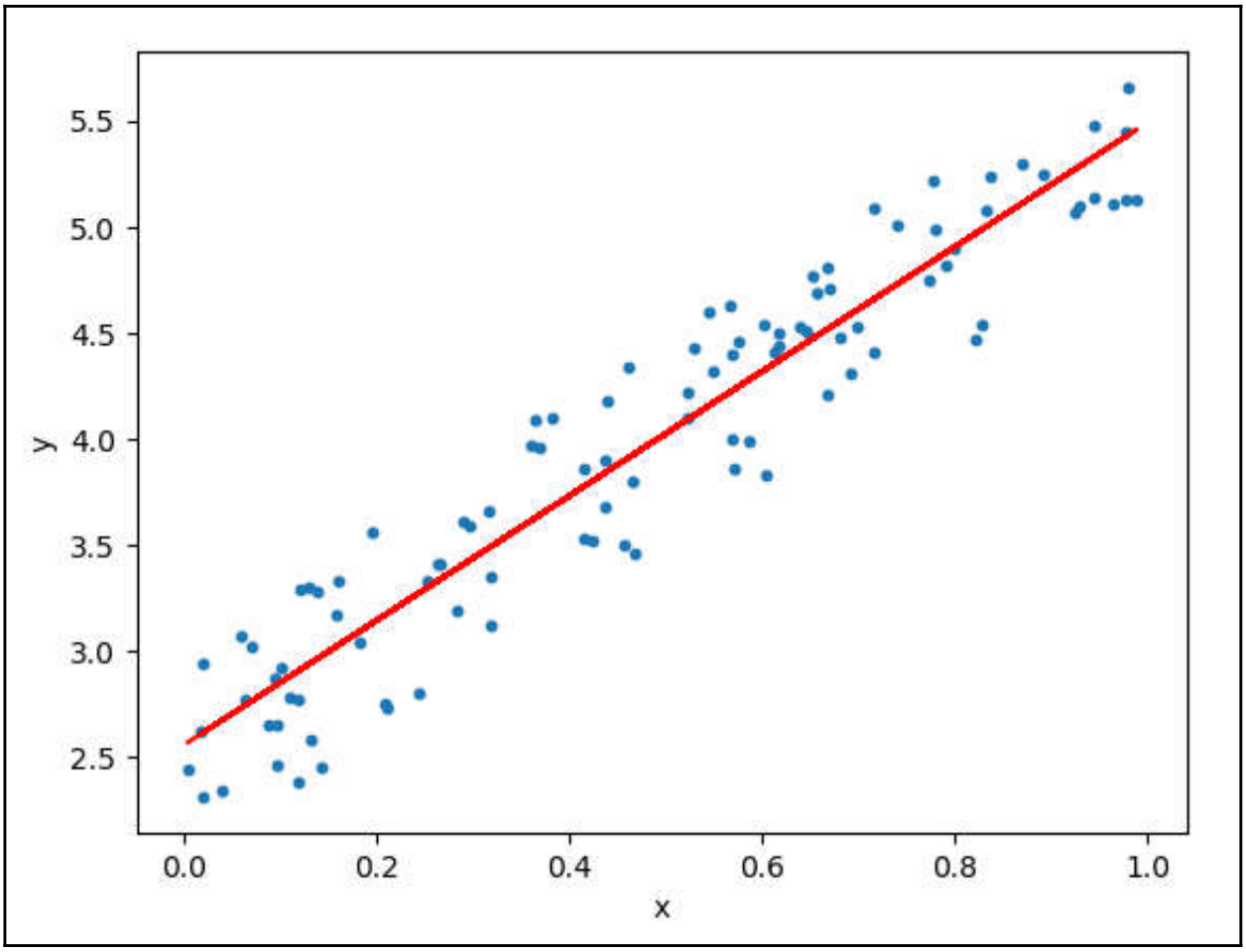

Marital Status (0: Non Conversions, 1: Conversions)
0
1
0.00/1.00
0.25
0.50
0.75
4136
68
9948
22396
476
12
1620
2532
Marital
divorced
married
single
unknown

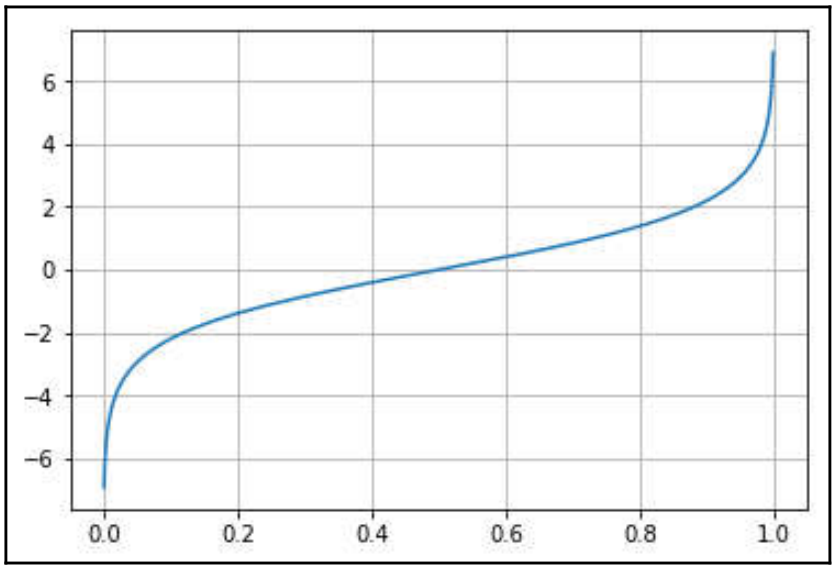
	AgeGroup	Marital	Count	NumConversions	TotalCount	ConversionRate
1	(20,30]	divorced	229	18	7243	0.0024851581
2	(20,30]	married	2389	242	7243	0.0334115698
3	(20,30]	single	4612	804	7243	0.1110037277
4	(20,30]	unknown	13	3	7243	0.0004141930
5	(30,40]	divorced	1505	135	16385	0.0082392432
6	(30,40]	married	9705	867	16385	0.0529142508
7	(30,40]	single	5139	591	16385	0.0360695758
8	(30,40]	unknown	36	4	16385	0.0002441257
9	(40,50]	divorced	1548	126	10240	0.0123046875
10	(40,50]	married	7383	588	10240	0.0574218750
11	(40,50]	single	1295	120	10240	0.0117187500
12	(40,50]	unknown	14	3	10240	0.0002929687


Conversion Rates by Age and Marital Status
Conversion Rate (%)
0.0
0.1
0.2
0.3
(20,30]
(30,40]
(40,50]
(50,60]
(60,70]
70+
Age
Marital
divorced
married
single
unknown


Conversion Rates by Age and Marital Status
Conversion Rate (%)
0.0
0.1
0.2
0.3
0.4
(20,30]
(30,40]
(40,50]
(50,60]
(60,70]
70+
Age
Marital
divorced
married
single
unknown

Chapter 3: Drivers behind Marketing Engagement






```
df.shape
```

```
(9134, 24)
```

```
df.head()
```

	Customer	State	Customer Lifetime Value	Response	Coverage	Education	Effective To Date	EmploymentStatus	Gender	Income	...	Months Since Policy Inception	Number of Open Complaints	Number of Policies	C
0	BU79786	Washington	2763.519279	No	Basic	Bachelor	2/24/11	Employed	F	56274	...	5	0	1	
1	QZ44356	Arizona	6979.535903	No	Extended	Bachelor	1/31/11	Unemployed	F	0	...	42	0	8	
2	AI49188	Nevada	12887.431650	No	Premium	Bachelor	2/19/11	Employed	F	48767	...	38	0	2	
3	WW63253	California	7645.861827	No	Basic	Bachelor	1/20/11	Unemployed	M	0	...	65	0	7	C
4	HB64268	Washington	2813.692575	No	Basic	Bachelor	2/3/11	Employed	M	43836	...	44	0	1	

5 rows × 24 columns

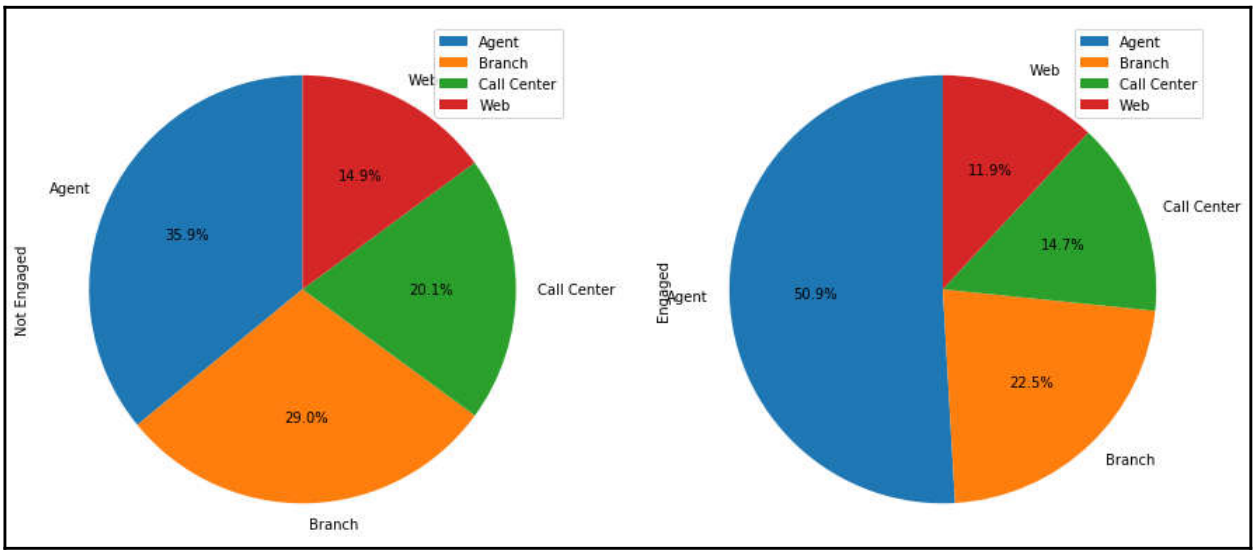
```
engagement_rate_df
```

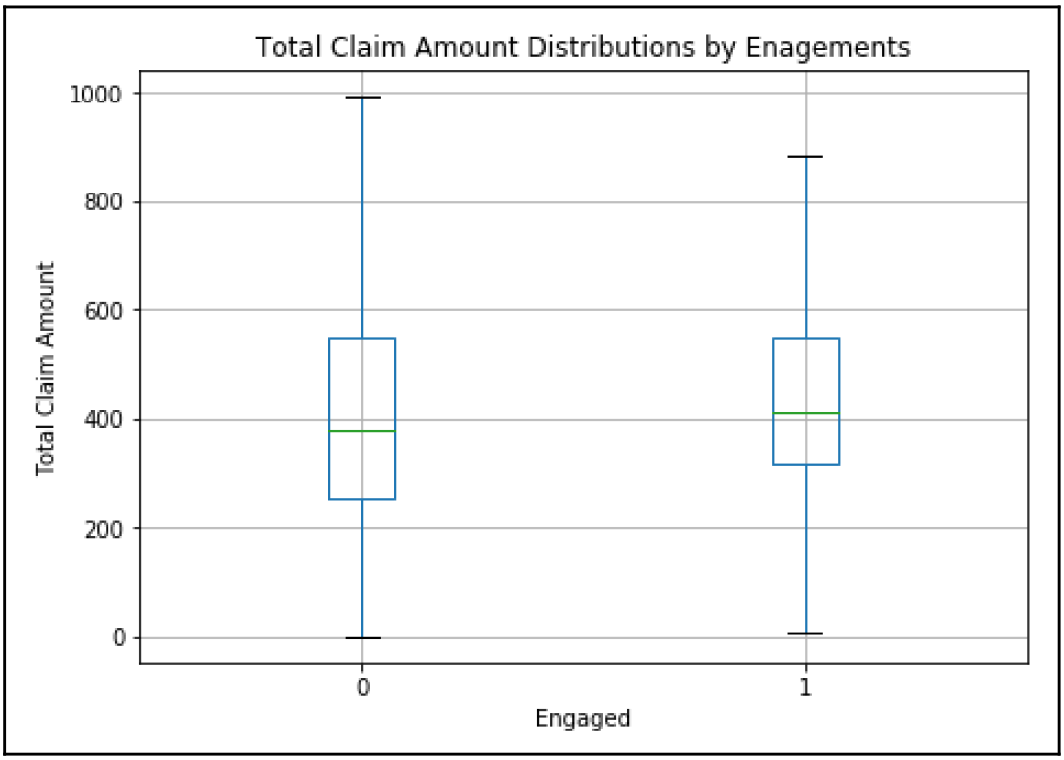
Engaged	Response
0	85.679877
1	14.320123

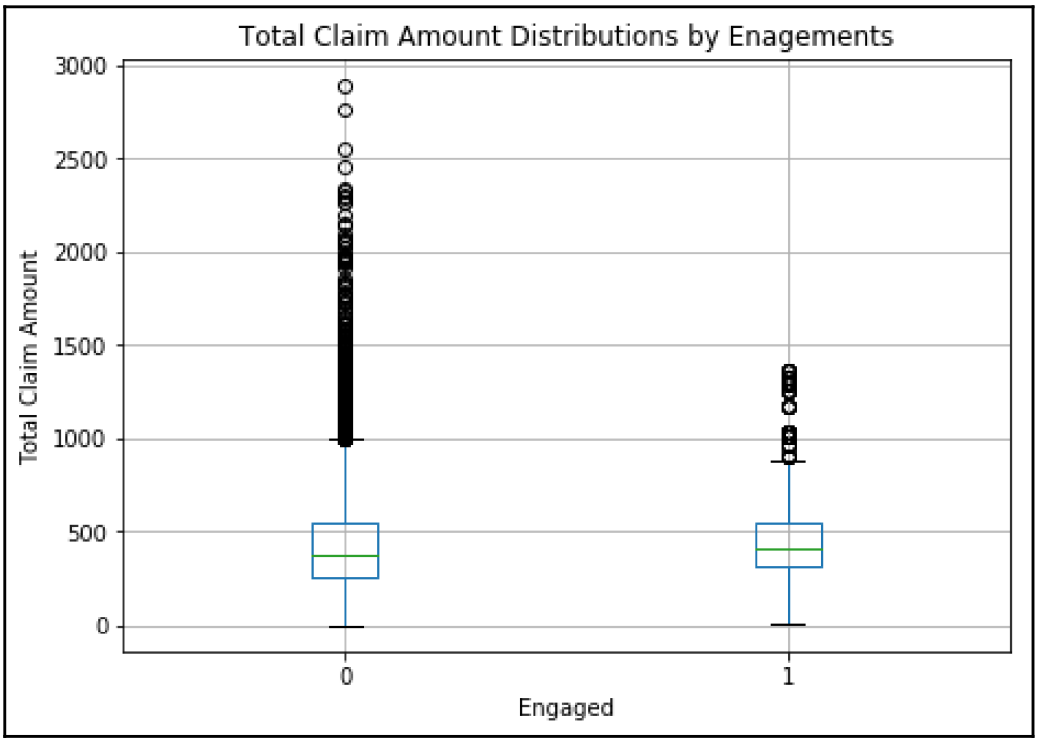
```
engagement_rate_df.T
```

Engaged	0	1
Response	85.679877	14.320123

Sales Channel	Not Engaged	Engaged
Agent	2811	666
Branch	2273	294
Call Center	1573	192
Web	1169	156




Total Claim Amount Distributions by Enagements
Total Claim Amount
1000
800
600
400
200
0
0
1
Engaged


Total Claim Amount Distributions by Enagements
Total Claim Amount
3000
2500
2000
1500
1000
500
0
0
1
Engaged

```
df['Income'].dtype
```

```
dtype('int64')
```

```
df['Customer Lifetime Value'].dtype
```

```
dtype('float64')
```

```
df.describe()
```

	Customer Lifetime Value	Income	Monthly Premium Auto	Months Since Last Claim	Months Since Policy Inception	Number of Open Complaints	Number of Policies	Total Claim Amount	Engaged
count	9134.000000	9134.000000	9134.000000	9134.000000	9134.000000	9134.000000	9134.000000	9134.000000	9134.000000
mean	8004.940475	37657.380009	93.219291	15.097000	48.064594	0.384388	2.966170	434.088794	0.143201
std	6870.967608	30379.904734	34.407967	10.073257	27.905991	0.910384	2.390182	290.500092	0.350297
min	1898.007675	0.000000	61.000000	0.000000	0.000000	0.000000	1.000000	0.099007	0.000000
25%	3994.251794	0.000000	68.000000	6.000000	24.000000	0.000000	1.000000	272.258244	0.000000
50%	5780.182197	33889.500000	83.000000	14.000000	48.000000	0.000000	2.000000	383.945434	0.000000
75%	8962.167041	62320.000000	109.000000	23.000000	71.000000	0.000000	4.000000	547.514839	0.000000
max	83325.381190	99981.000000	298.000000	35.000000	99.000000	5.000000	9.000000	2893.239678	1.000000

Logit Regression Results

Dep. Variable:	Engaged	**No. Observations:**	9134
Model:	Logit	**Df Residuals:**	9126
Method:	MLE	**Df Model:**	7
Date:	Tue, 04 Sep 2018	**Pseudo R-squ.:**	-0.02546
Time:	17:00:30	**Log-Likelihood:**	-3847.1
converged:	True	**LL-Null:**	-3751.6
		LLR p-value:	1.000

	coef	std err	z	P>|z|	[0.025	0.975]
Customer Lifetime Value	-6.741e-06	5.04e-06	-1.337	0.181	-1.66e-05	3.14e-06
Income	-2.857e-06	1.03e-06	-2.766	0.006	-4.88e-06	-8.33e-07
Monthly Premium Auto	-0.0084	0.001	-6.889	0.000	-0.011	-0.006
Months Since Last Claim	-0.0202	0.003	-7.238	0.000	-0.026	-0.015
Months Since Policy Inception	-0.0060	0.001	-6.148	0.000	-0.008	-0.004
Number of Open Complaints	-0.0829	0.034	-2.424	0.015	-0.150	-0.016
Number of Policies	-0.0810	0.013	-6.356	0.000	-0.106	-0.056
Total Claim Amount	0.0001	0.000	0.711	0.477	-0.000	0.000

```
gender_values
```

```
array([0, 0, 0, ..., 1, 1, 1])
```

```
gender_labels
```

```
Index(['F', 'M'], dtype='object')
```

```
logit_fit.summary()
```

Logit Regression Results

Dep. Variable:	Engaged	**No. Observations:**	9134
Model:	Logit	**Df Residuals:**	9132
Method:	MLE	**Df Model:**	1
Date:	Tue, 04 Sep 2018	**Pseudo R-squ.:**	-0.2005
Time:	17:31:15	**Log-Likelihood:**	-4503.7
converged:	True	**LL-Null:**	-3751.6
		LLR p-value:	1.000

	coef	std err	z	P>|z|	[0.025	0.975]
GenderFactorized	-1.1266	0.047	-24.116	0.000	-1.218	-1.035
EducationFactorized	-0.6256	0.021	-29.900	0.000	-0.667	-0.585

```
logit_fit.summary()
```

Logit Regression Results

Dep. Variable:	Engaged	**No. Observations:**	9134
Model:	Logit	**Df Residuals:**	9124
Method:	MLE	**Df Model:**	9
Date:	Tue, 04 Sep 2018	**Pseudo R-squ.:**	-0.02454
Time:	17:30:53	**Log-Likelihood:**	-3843.7
converged:	True	**LL-Null:**	-3751.6
		LLR p-value:	1.000

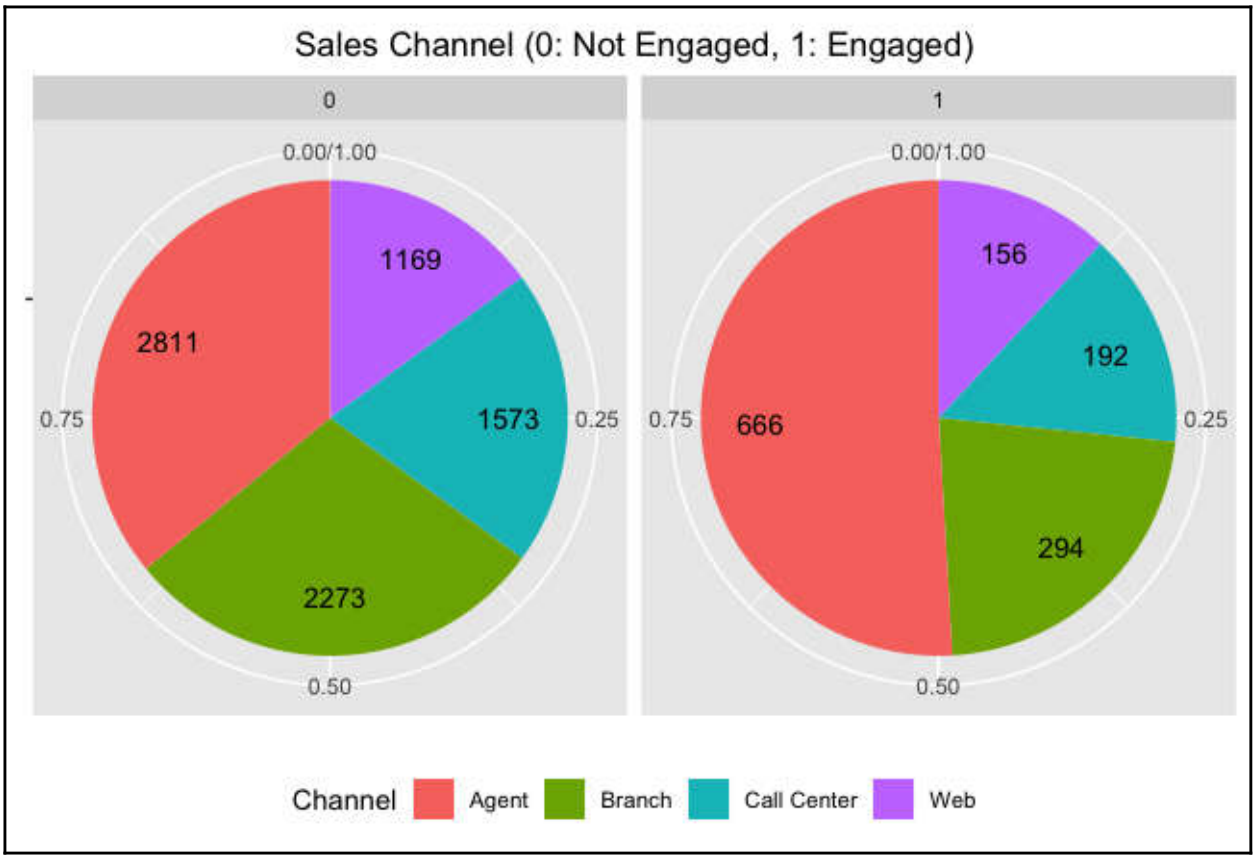
	coef	std err	z	P>|z|	[0.025	0.975]
Customer Lifetime Value	-6.909e-06	5.03e-06	-1.373	0.170	-1.68e-05	2.96e-06
Income	-2.59e-06	1.04e-06	-2.494	0.013	-4.63e-06	-5.55e-07
Monthly Premium Auto	-0.0081	0.001	-6.526	0.000	-0.011	-0.006
Months Since Last Claim	-0.0194	0.003	-6.858	0.000	-0.025	-0.014
Months Since Policy Inception	-0.0057	0.001	-5.827	0.000	-0.008	-0.004
Number of Open Complaints	-0.0813	0.034	-2.376	0.017	-0.148	-0.014
Number of Policies	-0.0781	0.013	-6.114	0.000	-0.103	-0.053
Total Claim Amount	0.0001	0.000	0.943	0.346	-0.000	0.000
GenderFactorized	-0.1500	0.058	-2.592	0.010	-0.263	-0.037
EducationFactorized	-0.0070	0.027	-0.264	0.792	-0.059	0.045

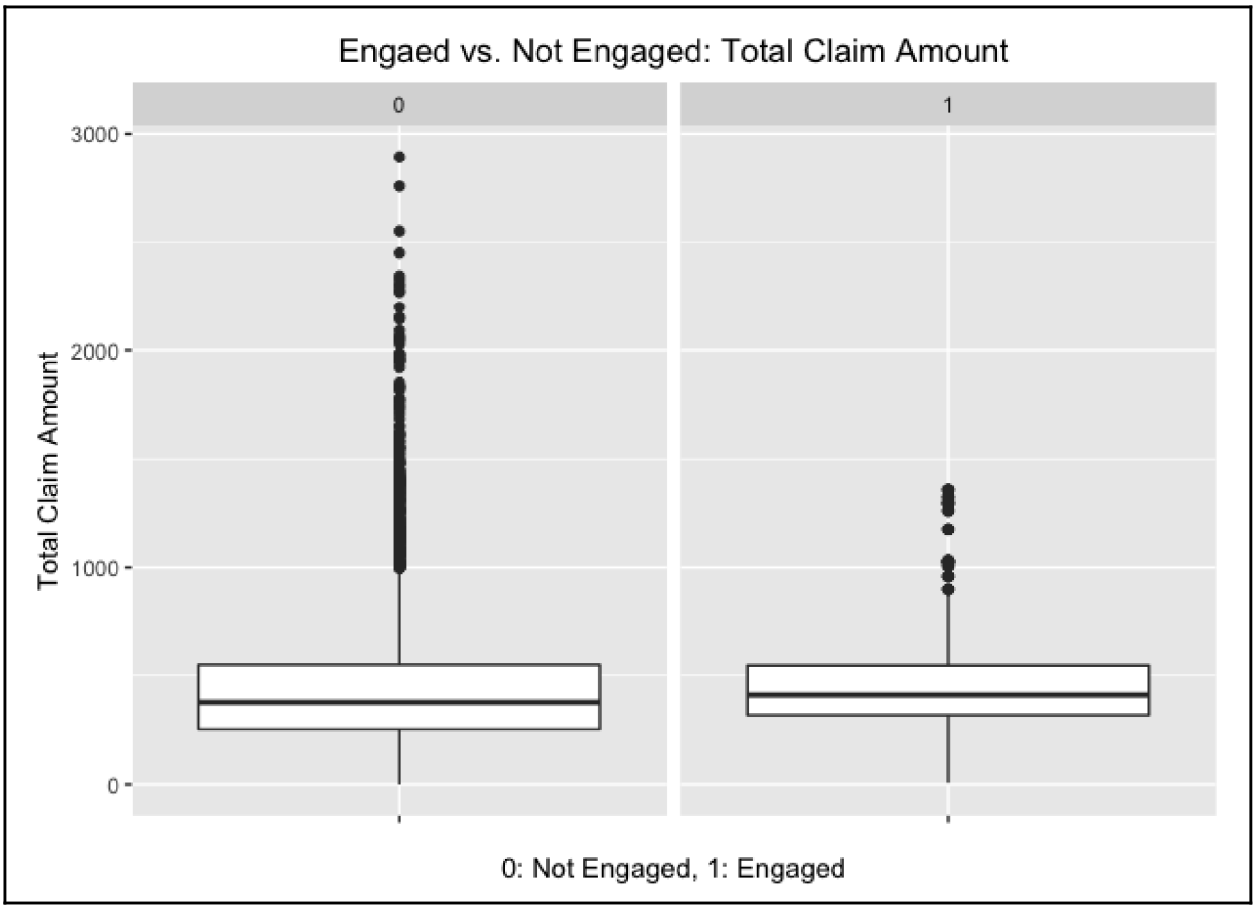
	Customer	State	Customer.Lifetime.Value	Response	Coverage	Education	Effective.To.Date	EmploymentStatus	Gender	Income	Location.Code
1	BU79786	Washington	2763.519	No	Basic	Bachelor	2/24/11	Employed	F	56274	Suburban
2	QZ44356	Arizona	6979.536	No	Extended	Bachelor	1/31/11	Unemployed	F	0	Suburban
3	AI49188	Nevada	12887.432	No	Premium	Bachelor	2/19/11	Employed	F	48767	Suburban
4	WW63253	California	7645.862	No	Basic	Bachelor	1/20/11	Unemployed	M	0	Suburban
5	HB64268	Washington	2813.693	No	Basic	Bachelor	2/3/11	Employed	M	43836	Rural
6	OC83172	Oregon	8256.298	Yes	Basic	Bachelor	1/25/11	Employed	F	62902	Rural
7	XZ87318	Oregon	5380.899	Yes	Basic	College	2/24/11	Employed	F	55350	Suburban
8	CF85061	Arizona	7216.100	No	Premium	Master	1/18/11	Unemployed	M	0	Urban
9	DY87989	Oregon	24127.504	Yes	Basic	Bachelor	1/26/11	Medical Leave	M	14072	Suburban
10	BQ94931	Oregon	7388.178	No	Extended	College	2/17/11	Employed	F	28812	Urban
11	SX51350	California	4738.992	No	Basic	College	2/21/11	Unemployed	M	0	Suburban
12	VQ65197	California	8197.197	No	Basic	College	1/6/11	Unemployed	F	0	Suburban
13	DP39365	California	8798.797	No	Premium	Master	2/6/11	Employed	M	77026	Urban
14	SJ95423	Arizona	8819.019	Yes	Basic	High School or Below	1/10/11	Employed	M	99845	Suburban
15	IL66569	California	5384.432	No	Basic	College	1/18/11	Employed	M	83689	Urban

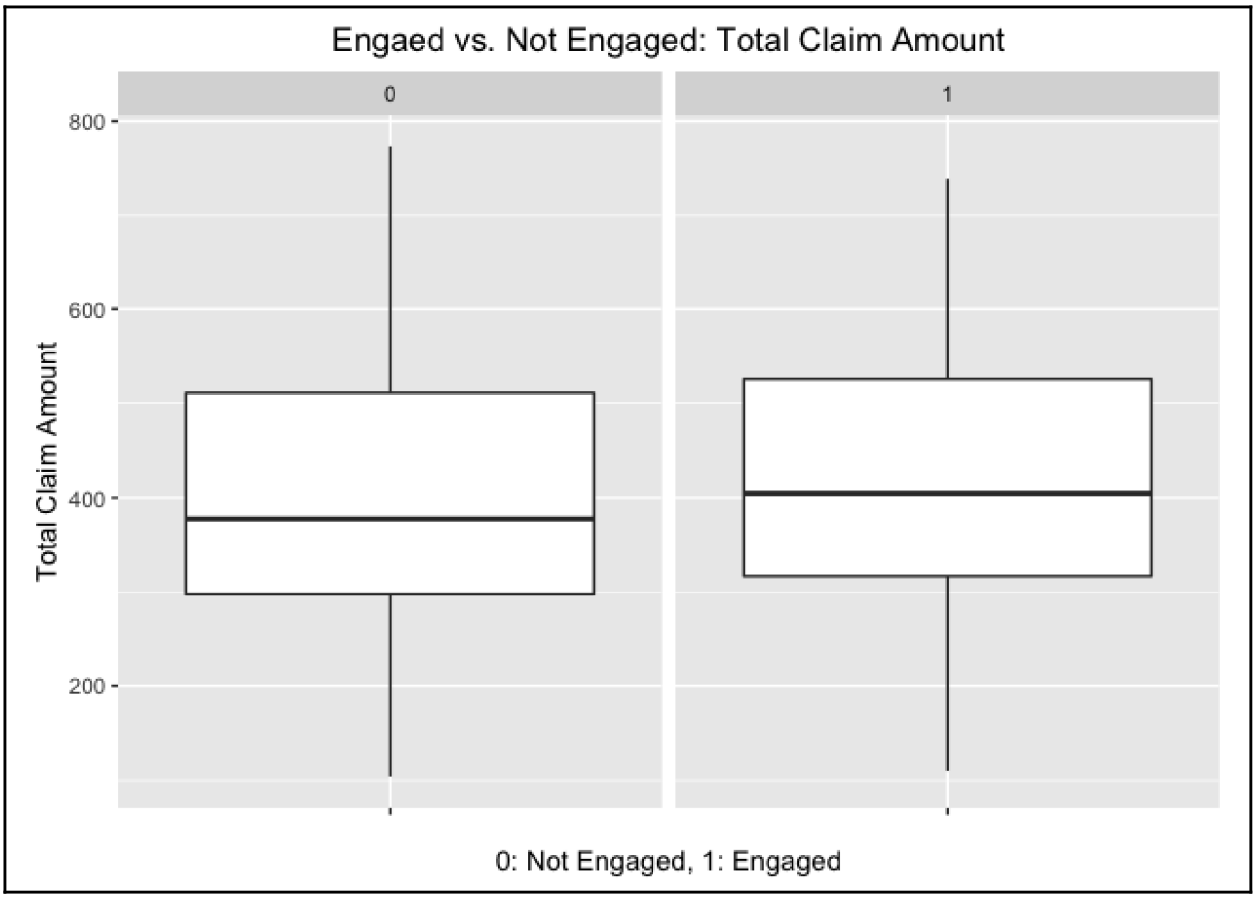
	Engaged	Count	Percentage
1	0	7826	85.67988
2	1	1308	14.32012

	0	1
Count	7826.00000	1308.00000
Percentage	85.67988	14.32012

	Engaged	Channel	Count
1	0	Agent	2811
2	0	Branch	2273
3	0	Call Center	1573
4	0	Web	1169
5	1	Agent	666
6	1	Branch	294
7	1	Call Center	192
8	1	Web	156


Sales Channel (0: Not Engaged, 1: Engaged)
0
1
0.00/1.00
0.25
0.50
0.75
1169
2811
1573
2273
156
192
666
294
Channel
Agent
Branch
Call Center
Web


Engaed vs. Not Engaged: Total Claim Amount
0
1
3000
2000
1000
0
Total Claim Amount
0: Not Engaged, 1: Engaged




```
> sapply(df, class)
                     Customer                         State       Customer.Lifetime.Value                      Response
                     "factor"                      "factor"                     "numeric"                      "factor"
                     Coverage                     Education             Effective.To.Date                EmploymentStatus
                     "factor"                      "factor"                      "factor"                      "factor"
                       Gender                       Income                 Location.Code                Marital.Status
                     "factor"                     "integer"                      "factor"                      "factor"
        Monthly.Premium.Auto      Months.Since.Last.Claim Months.Since.Policy.Inception     Number.of.Open.Complaints
                     "integer"                     "integer"                     "integer"                     "integer"
          Number.of.Policies                  Policy.Type                        Policy                Renew.Offer.Type
                     "integer"                      "factor"                      "factor"                      "factor"
              Sales.Channel           Total.Claim.Amount                 Vehicle.Class                  Vehicle.Size
                     "factor"                     "numeric"                      "factor"                      "factor"
                     Engaged
                     "numeric"
```

```
   Customer          State      Customer.Lifetime.Value Response      Coverage                     Education    Effective.To.Date
AA10041:   1   Arizona   :1703   Min.   : 1898           No :7826   Basic   :5568   Bachelor            :2748   1/10/11: 195
AA11235:   1   California:3150   1st Qu.: 3994           Yes:1308   Extended:2742   College             :2681   1/27/11: 194
AA16582:   1   Nevada    : 882   Median : 5780                      Premium : 824   Doctor              : 342   2/14/11: 186
AA30683:   1   Oregon    :2601   Mean   : 8005                                      High School or Below:2622   1/26/11: 181
AA34092:   1   Washington: 798   3rd Qu.: 8962                                      Master              : 741   1/17/11: 180
AA35519:   1                    Max.   :83325                                                                  1/19/11: 179
(Other):9128                                                                                                   (Other):8019
```

```
> # get numeric columns
> continuousDF <- select_if(df, is.numeric)
> colnames(continuousDF)
[1] "Customer.Lifetime.Value"       "Income"                       "Monthly.Premium.Auto"          "Months.Since.Last.Claim"
[5] "Months.Since.Policy.Inception" "Number.of.Open.Complaints"    "Number.of.Policies"            "Total.Claim.Amount"
[9] "Engaged"
```

```
> summary(logit.fit)

Call:
glm(formula = Engaged ~ ., family = binomial, data = continuousDF)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7629  -0.5704  -0.5477  -0.5216   2.1018

Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                 -1.787e+00  1.234e-01 -14.476   <2e-16 ***
Customer.Lifetime.Value     -6.327e-06  4.863e-06  -1.301   0.1933
Income                       2.042e-06  1.092e-06   1.869   0.0616 .
Monthly.Premium.Auto        -1.194e-04  1.226e-03  -0.097   0.9224
Months.Since.Last.Claim     -4.489e-03  2.987e-03  -1.503   0.1329
Months.Since.Policy.Inception  2.125e-04  1.073e-03   0.198   0.8429
Number.of.Open.Complaints   -3.257e-02  3.379e-02  -0.964   0.3351
Number.of.Policies          -2.443e-02  1.283e-02  -1.904   0.0569 .
Total.Claim.Amount           2.772e-04  1.463e-04   1.895   0.0581 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7503.3  on 9133  degrees of freedom
Residual deviance: 7488.1  on 9125  degrees of freedom
AIC: 7506.1

Number of Fisher Scoring iterations: 4
```

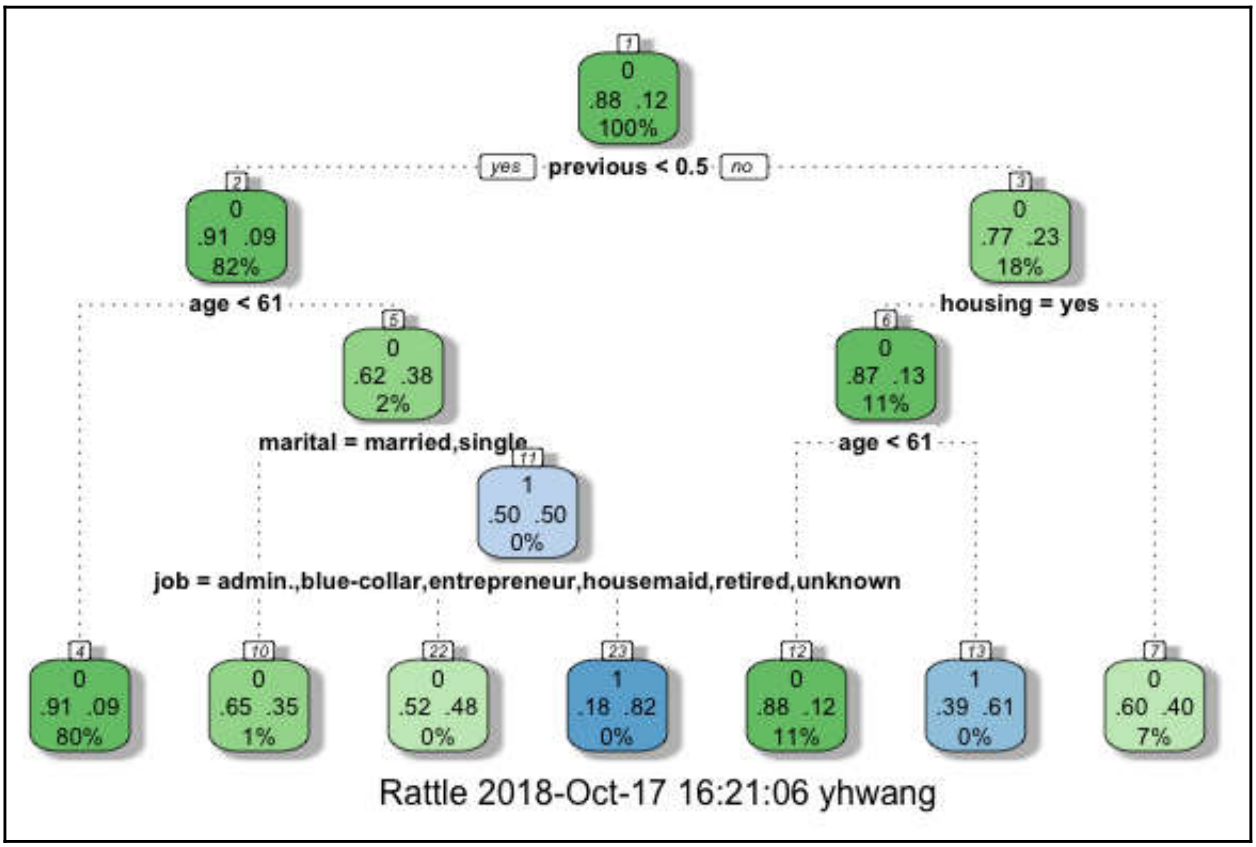
```
> summary(logit.fit)

Call:
glm(formula = Engaged ~ factor(Education), family = binomial,
    data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6211  -0.5746  -0.5440  -0.5287   2.0184

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                       -1.83575    0.05538 -33.146   <2e-16 ***
factor(Education)College           0.11816    0.07719   1.531   0.1258
factor(Education)Doctor            0.28819    0.15258   1.889   0.0589 .
factor(Education)High School or Below -0.06137    0.08019  -0.765   0.4441
factor(Education)Master            0.19191    0.11407   1.682   0.0925 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7503.3  on 9133  degrees of freedom
Residual deviance: 7492.4  on 9129  degrees of freedom
AIC: 7502.4

Number of Fisher Scoring iterations: 4
```

```
> summary(logit.fit)

Call:
glm(formula = Engaged ~ factor(Education) + factor(Gender), family = binomial,
    data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6247  -0.5713  -0.5409  -0.5256   2.0238

Coefficients:
                                   Estimate Std. Error z value Pr(>|z|)
(Intercept)                        -1.84803    0.06257 -29.537   <2e-16 ***
factor(Education)College            0.11782    0.07720   1.526   0.1269
factor(Education)Doctor             0.28759    0.15259   1.885   0.0595 .
factor(Education)High School or Below -0.06173    0.08019  -0.770   0.4415
factor(Education)Master             0.19223    0.11407   1.685   0.0919 .
factor(Gender)M                     0.02534    0.05979   0.424   0.6717
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7503.3  on 9133  degrees of freedom
Residual deviance: 7492.3  on 9128  degrees of freedom
AIC: 7504.3

Number of Fisher Scoring iterations: 4
```

```
> colnames(continuousDF)
 [1] "Customer.Lifetime.Value"       "Income"                        "Monthly.Premium.Auto"
 [4] "Months.Since.Last.Claim"       "Months.Since.Policy.Inception" "Number.of.Open.Complaints"
 [7] "Number.of.Policies"            "Total.Claim.Amount"            "Engaged"
[10] "Gender"                        "Education"
```

```
> summary(logit.fit)

Call:
glm(formula = Engaged ~ ., family = binomial, data = continuousDF)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.7905  -0.5739  -0.5427  -0.5095   2.1431

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -1.837e+00  1.342e-01 -13.693   <2e-16 ***
Customer.Lifetime.Value       -6.065e-06  4.872e-06  -1.245   0.2132
Income                         2.044e-06  1.094e-06   1.867   0.0618 .
Monthly.Premium.Auto          -4.619e-04  1.237e-03  -0.374   0.7087
Months.Since.Last.Claim       -4.717e-03  2.993e-03  -1.576   0.1150
Months.Since.Policy.Inception  1.856e-04  1.074e-03   0.173   0.8627
Number.of.Open.Complaints     -3.448e-02  3.378e-02  -1.021   0.3075
Number.of.Policies            -2.392e-02  1.285e-02  -1.862   0.0626 .
Total.Claim.Amount             3.471e-04  1.487e-04   2.335   0.0196 *
GenderM                        1.537e-02  6.017e-02   0.255   0.7984
EducationCollege               1.216e-01  7.731e-02   1.573   0.1158
EducationDoctor                3.107e-01  1.532e-01   2.028   0.0425 *
EducationHigh School or Below -7.456e-02  8.056e-02  -0.925   0.3547
EducationMaster                2.065e-01  1.149e-01   1.798   0.0722 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7503.3  on 9133  degrees of freedom
Residual deviance: 7475.5  on 9120  degrees of freedom
AIC: 7503.5

Number of Fisher Scoring iterations: 4
```

Chapter 4: From Engagement to Conversion




```
df.shape
```

```
(45211, 17)
```

```
df.head()
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no

```
conversion_rate_df
```

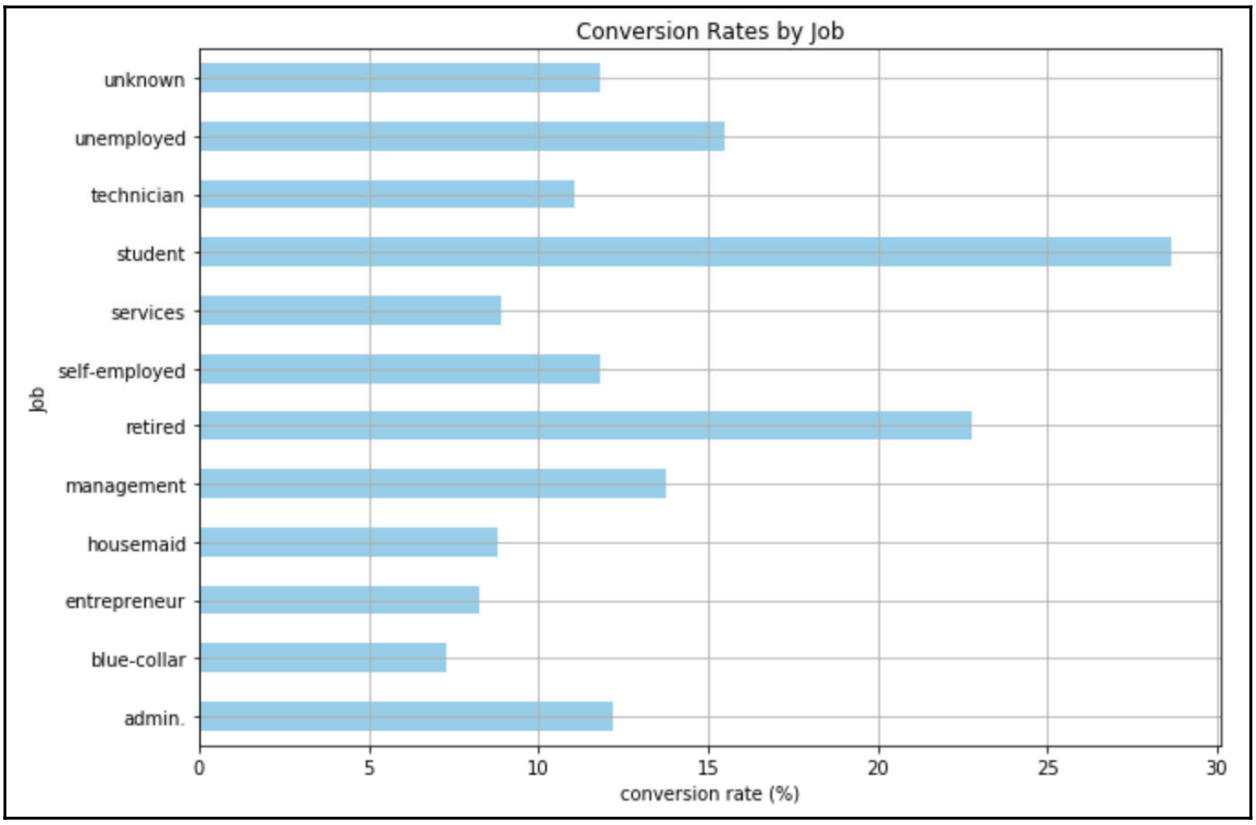
	y
conversion	
0	88.30152
1	11.69848

```
conversion_rate_df.T
```

conversion	**0**	**1**
y	88.30152	11.69848

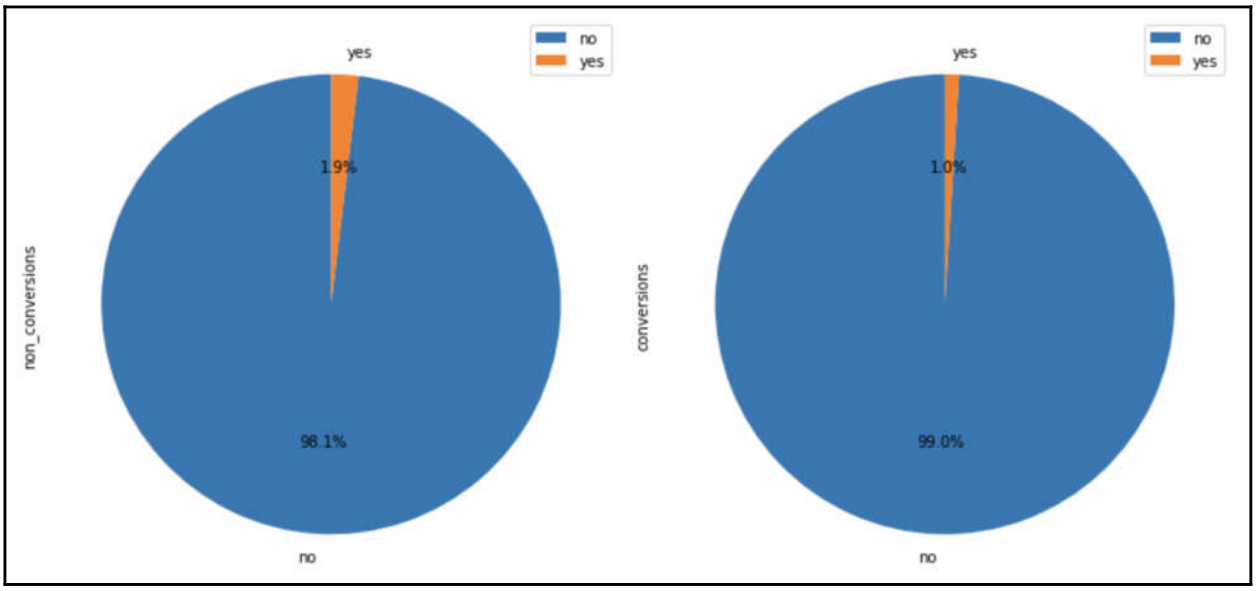
```
conversion_rate_by_job
```

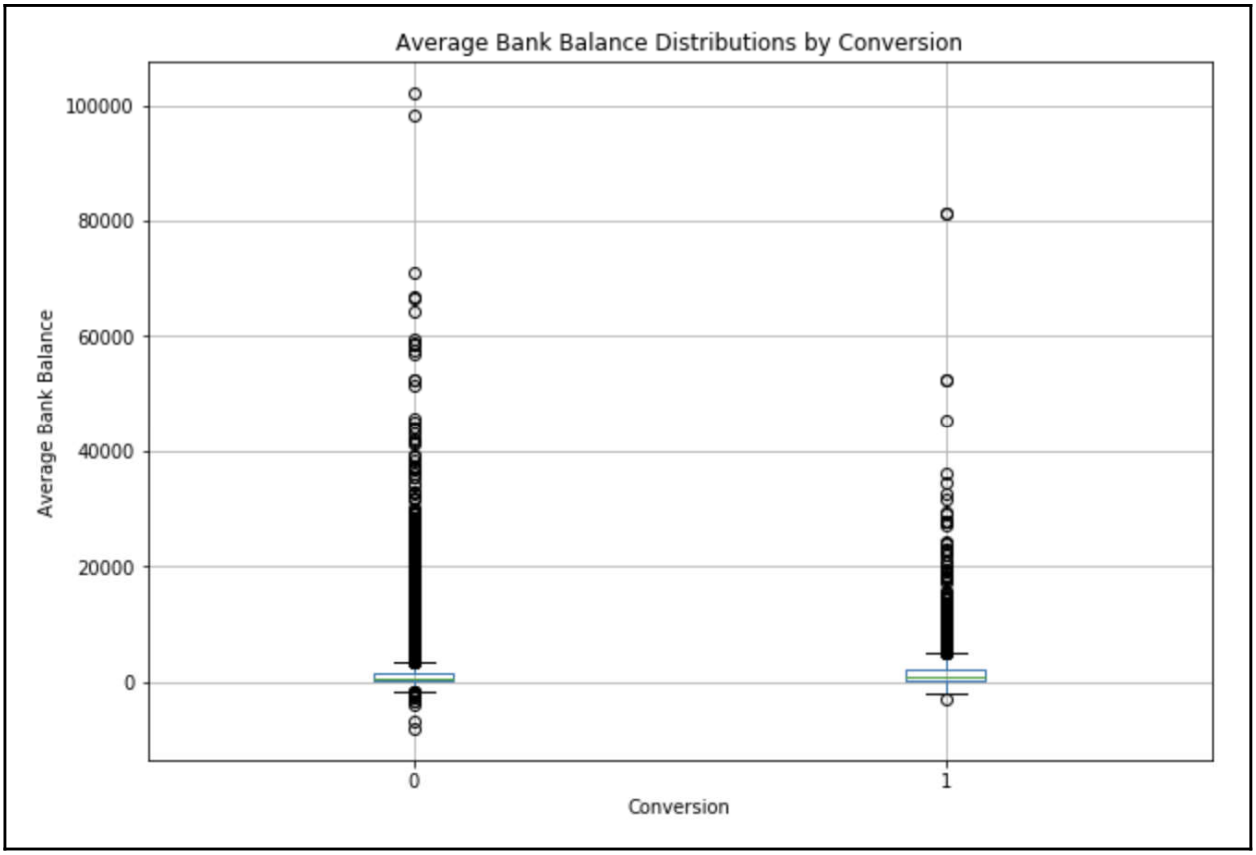
```
job
admin.           12.202669
blue-collar       7.274969
entrepreneur      8.271688
housemaid         8.790323
management       13.755551
retired          22.791519
self-employed    11.842939
services          8.883004
student          28.678038
technician       11.056996
unemployed       15.502686
unknown          11.805556
Name: conversion, dtype: float64
```

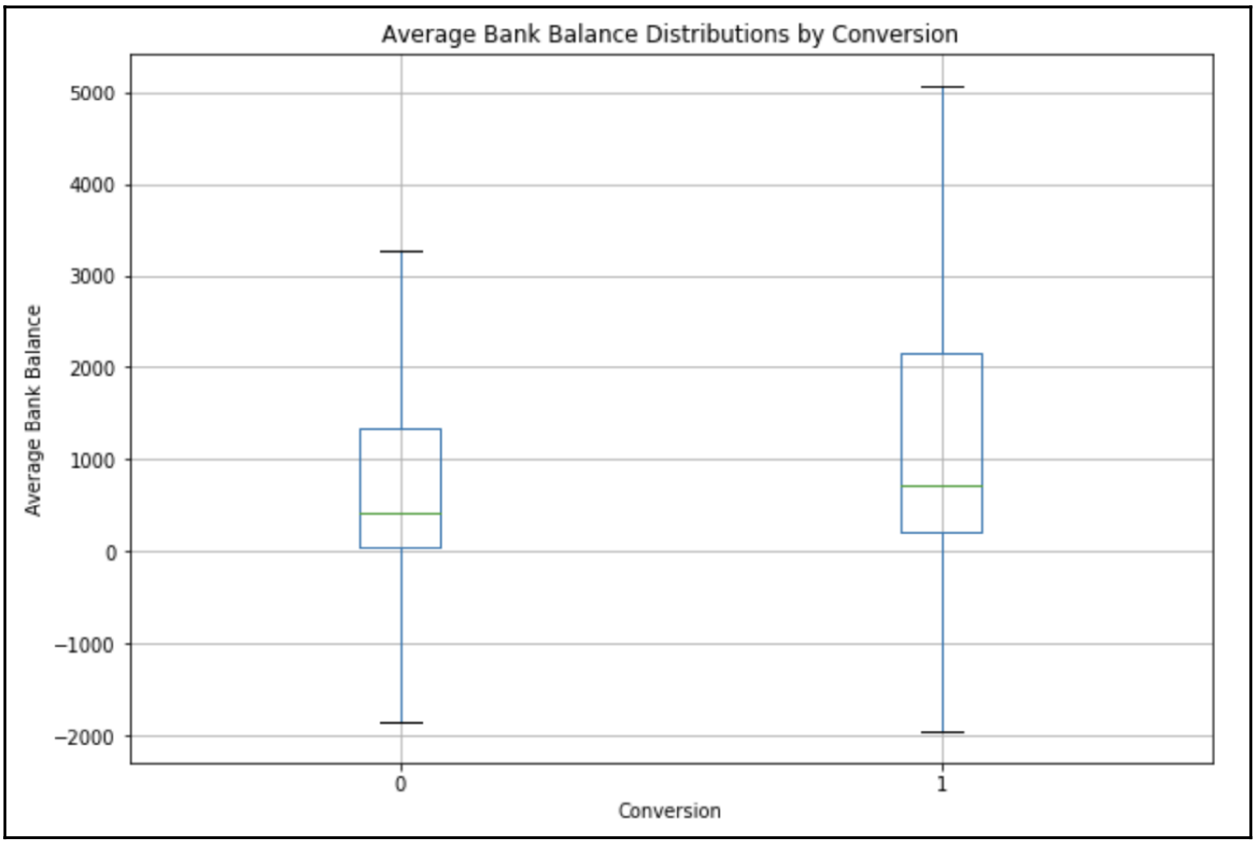



```
default_by_conversion_df
```

conversion	0	1
default		
no	39159	5237
yes	763	52

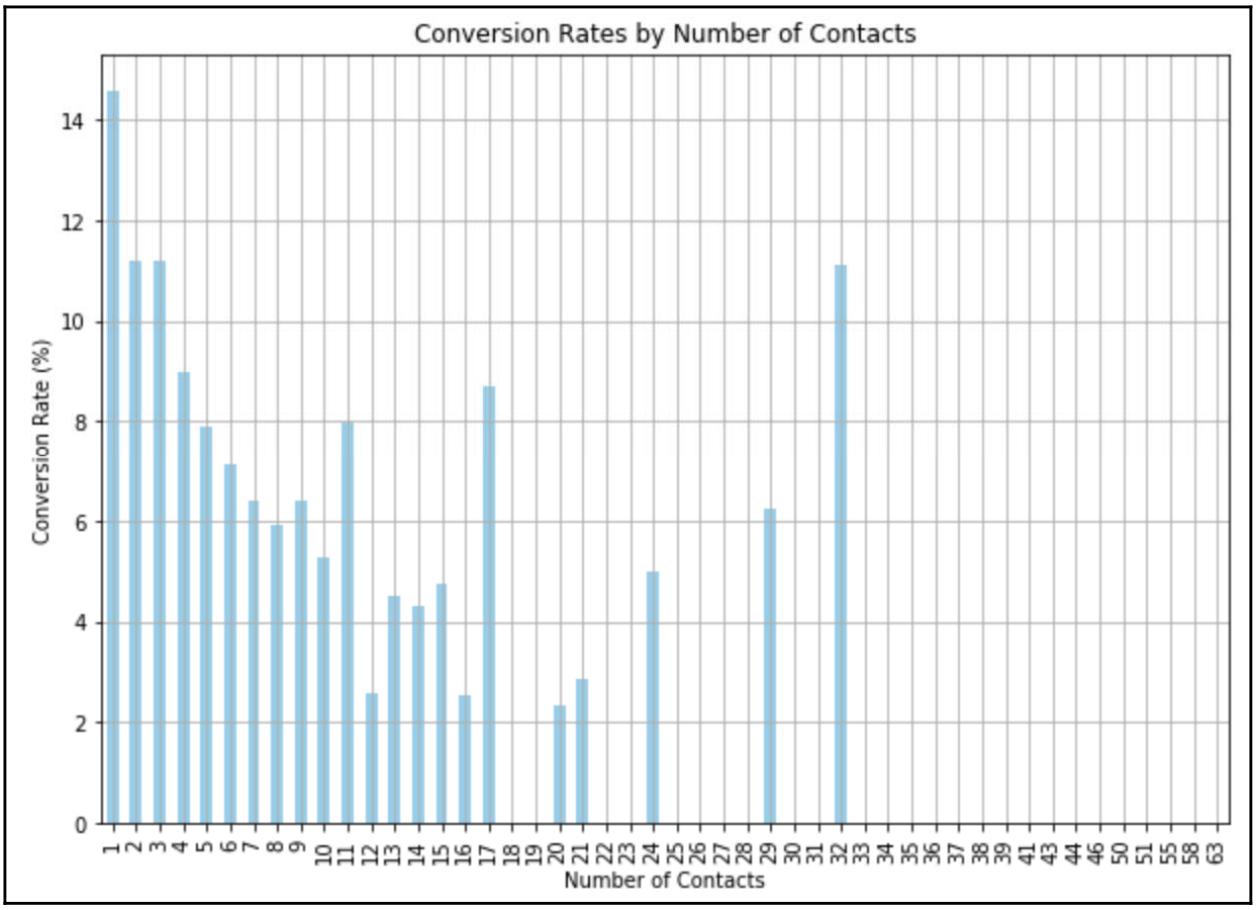

no
yes
yes
1.9%
98.1%
no
non_conversions
no
yes
yes
1.0%
99.0%
no
conversions


Average Bank Balance Distributions by Conversion
100000
80000
60000
40000
20000
0
Average Bank Balance
0
1
Conversion


Average Bank Balance Distributions by Conversion
Average Bank Balance
5000
4000
3000
2000
1000
0
−1000
−2000
0
1
Conversion

```
pd.DataFrame(conversions_by_num_contacts)
```

campaign	conversion
1	14.597583
2	11.203519
3	11.193624
4	9.000568
5	7.879819
6	7.126259
7	6.394558
8	5.925926
9	6.422018
10	5.263158




```
df['month'].unique()
```

```
array(['may', 'jun', 'jul', 'aug', 'oct', 'nov', 'dec', 'jan', 'feb',
       'mar', 'apr', 'sep'], dtype=object)
```

```
df['month'].unique()
```

```
array([ 5,  6,  7,  8, 10, 11, 12,  1,  2,  3,  4,  9])
```

```
df.groupby('month').count()['conversion']
```

```
month
1       1403
2       2649
3        477
4       2932
5      13766
6       5341
7       6895
8       6247
9        579
10       738
11      3970
12       214
Name: conversion, dtype: int64
```

```
df['job'].unique()
```

```
array(['management', 'technician', 'entrepreneur', 'blue-collar',
       'unknown', 'retired', 'admin.', 'services', 'self-employed',
       'unemployed', 'housemaid', 'student'], dtype=object)
```

	job_admin.	job_blue-collar	job_entrepreneur	job_housemaid	job_management	job_retired	job_self-employed	job_services	job_student	job_technician	job_unemployed	job_unknown
0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	1

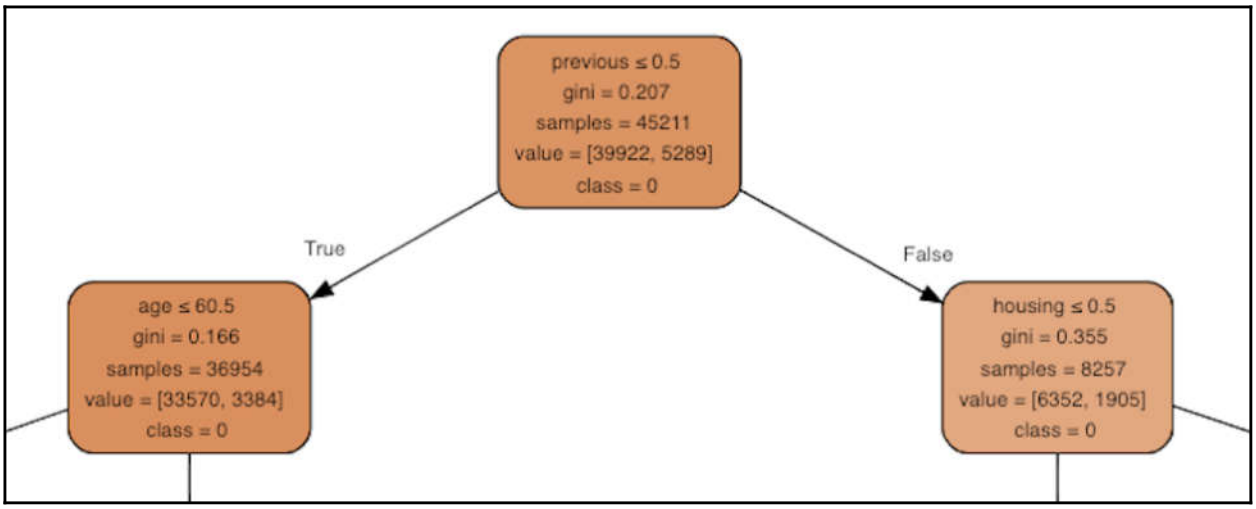
	age	job	marital	education	default	balance	housing	loan	contact	day	...	job_entrepreneur	job_housemaid	job_management	job_retired
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	...	0	0	1	0
1	44	technician	single	secondary	no	29	yes	no	unknown	5	...	0	0	0	0
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	...	1	0	0	0
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	...	0	0	0	0
4	33	unknown	single	unknown	no	1	no	no	unknown	5	...	0	0	0	0

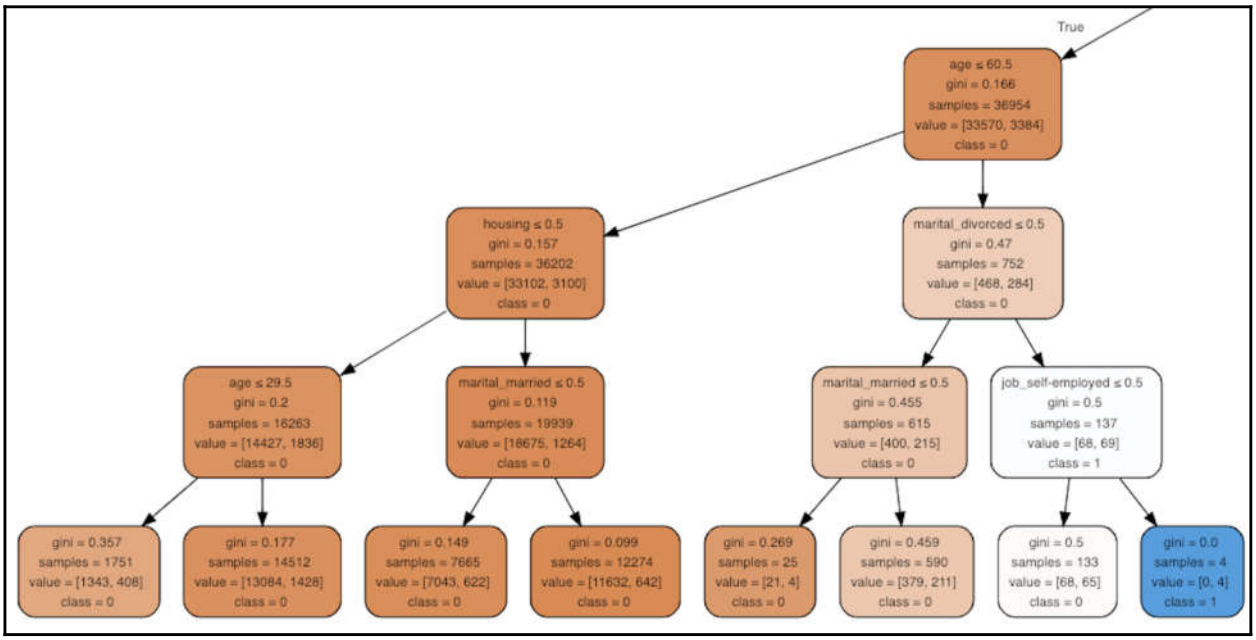
```
marital_encoded_df.head()
```

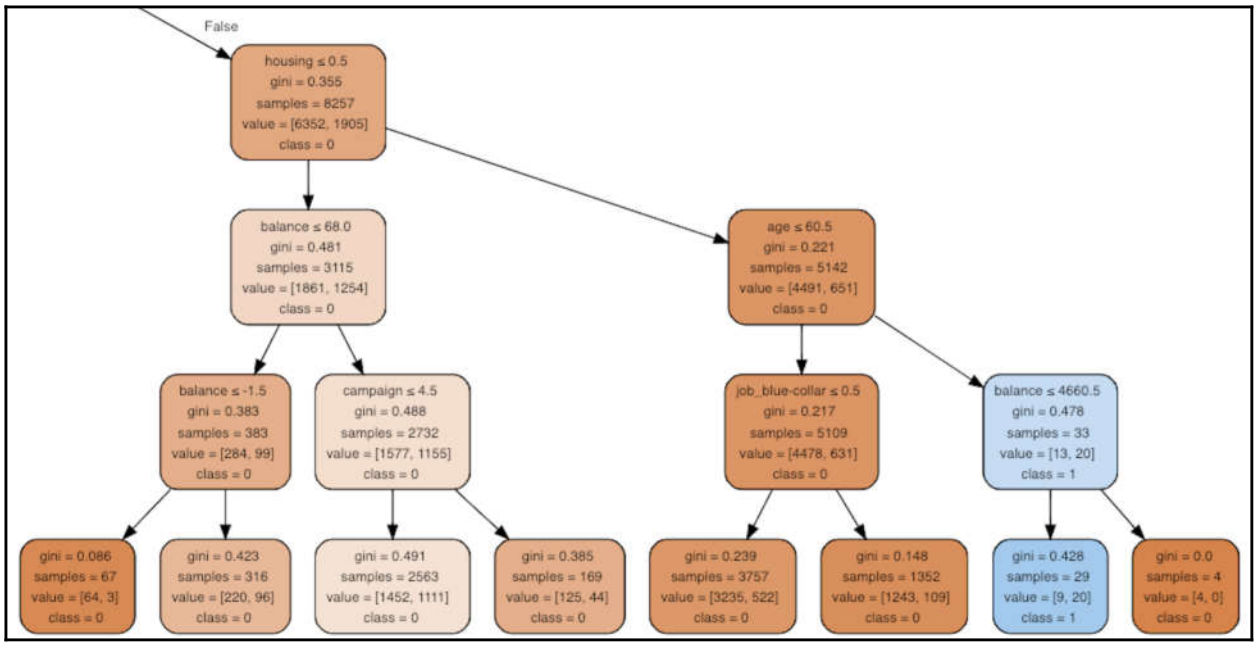
	marital_divorced	marital_married	marital_single
0	0	1	0
1	0	0	1
2	0	1	0
3	0	1	0
4	0	0	1

```
features
```

```
['age',
 'balance',
 'campaign',
 'previous',
 'housing',
 'job_admin.',
 'job_blue-collar',
 'job_entrepreneur',
 'job_housemaid',
 'job_management',
 'job_retired',
 'job_self-employed',
 'job_services',
 'job_student',
 'job_technician',
 'job_unemployed',
 'job_unknown',
 'marital_divorced',
 'marital_married',
 'marital_single']
```


previous ≤ 0.5
gini = 0.207
samples = 45211
value = [39922, 5289]
class = 0
True
False
age ≤ 60.5
gini = 0.166
samples = 36954
value = [33570, 3384]
class = 0
housing ≤ 0.5
gini = 0.355
samples = 8257
value = [6352, 1905]
class = 0

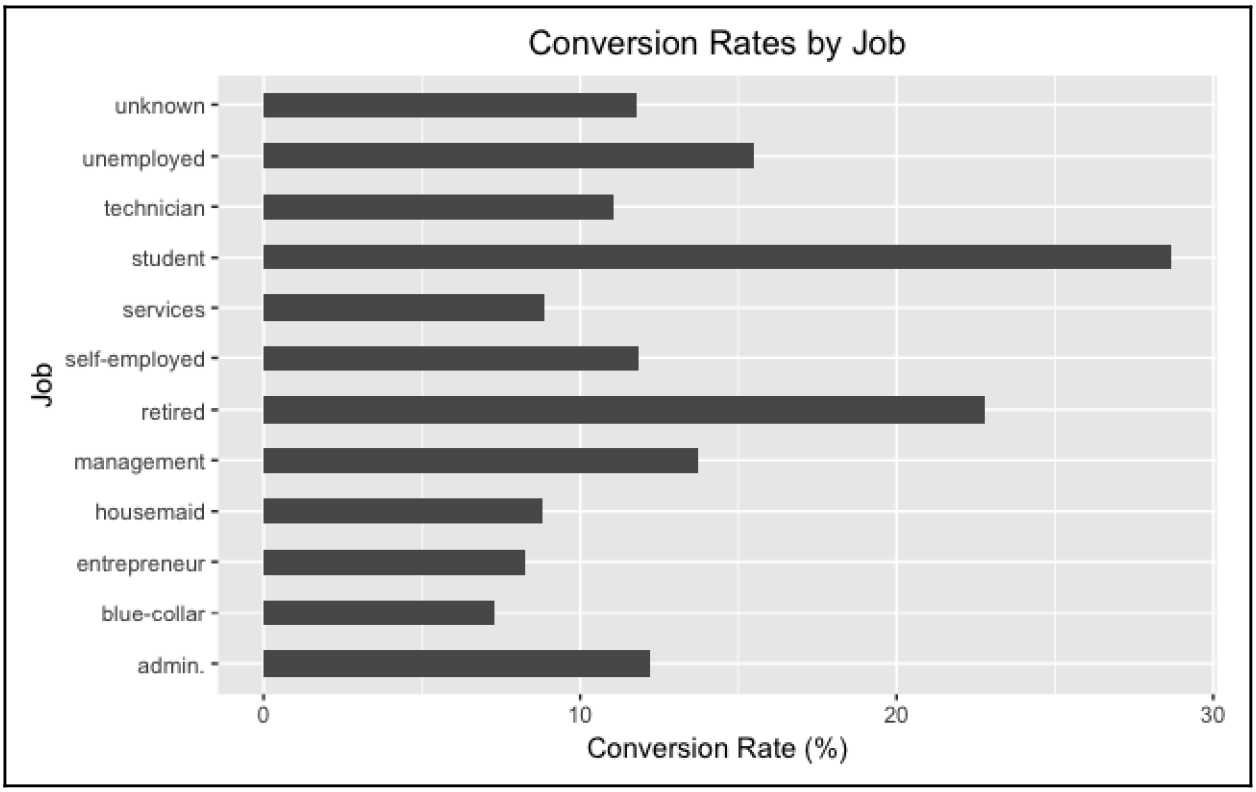

True
age ≤ 60.5
gini = 0.166
samples = 36954
value = [33570, 3384]
class = 0
housing ≤ 0.5
gini = 0.157
samples = 36202
value = [33102, 3100]
class = 0
marital_divorced ≤ 0.5
gini = 0.47
samples = 752
value = [468, 284]
class = 0
age ≤ 29.5
gini = 0.2
samples = 16263
value = [14427, 1836]
class = 0
marital_married ≤ 0.5
gini = 0.119
samples = 19939
value = [18675, 1264]
class = 0
marital_married ≤ 0.5
gini = 0.455
samples = 615
value = [400, 215]
class = 0
job_self-employed ≤ 0.5
gini = 0.5
samples = 137
value = [68, 69]
class = 1
gini = 0.357
samples = 1751
value = [1343, 408]
class = 0
gini = 0.177
samples = 14512
value = [13084, 1428]
class = 0
gini = 0.149
samples = 7665
value = [7043, 622]
class = 0
gini = 0.099
samples = 12274
value = [11632, 642]
class = 0
gini = 0.269
samples = 25
value = [21, 4]
class = 0
gini = 0.459
samples = 590
value = [379, 211]
class = 0
gini = 0.5
samples = 133
value = [68, 65]
class = 0
gini = 0.0
samples = 4
value = [0, 4]
class = 1



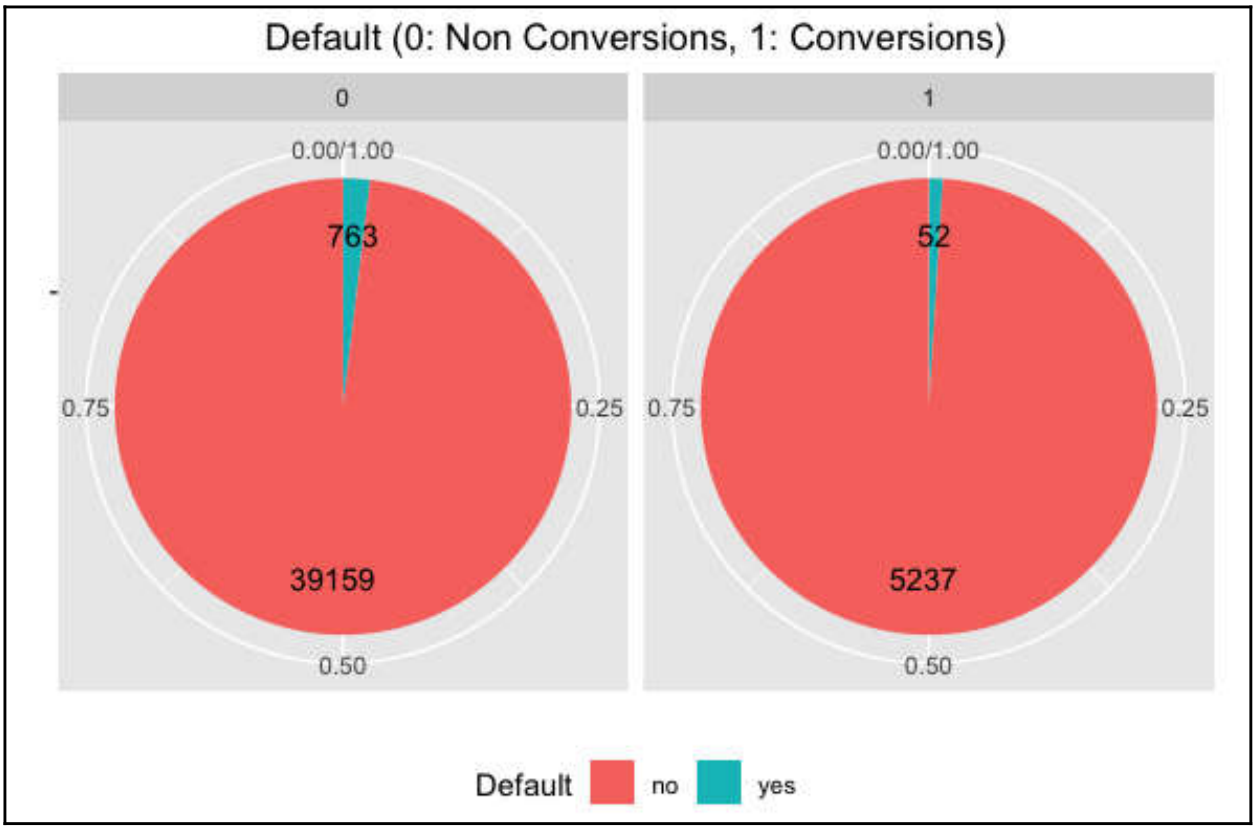
	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
1	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
2	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
3	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
4	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
5	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
6	35	management	married	tertiary	no	231	yes	no	unknown	5	may	139	1	-1	0	unknown	no
7	28	management	single	tertiary	no	447	yes	yes	unknown	5	may	217	1	-1	0	unknown	no
8	42	entrepreneur	divorced	tertiary	yes	2	yes	no	unknown	5	may	380	1	-1	0	unknown	no
9	58	retired	married	primary	no	121	yes	no	unknown	5	may	50	1	-1	0	unknown	no
10	43	technician	single	secondary	no	593	yes	no	unknown	5	may	55	1	-1	0	unknown	no

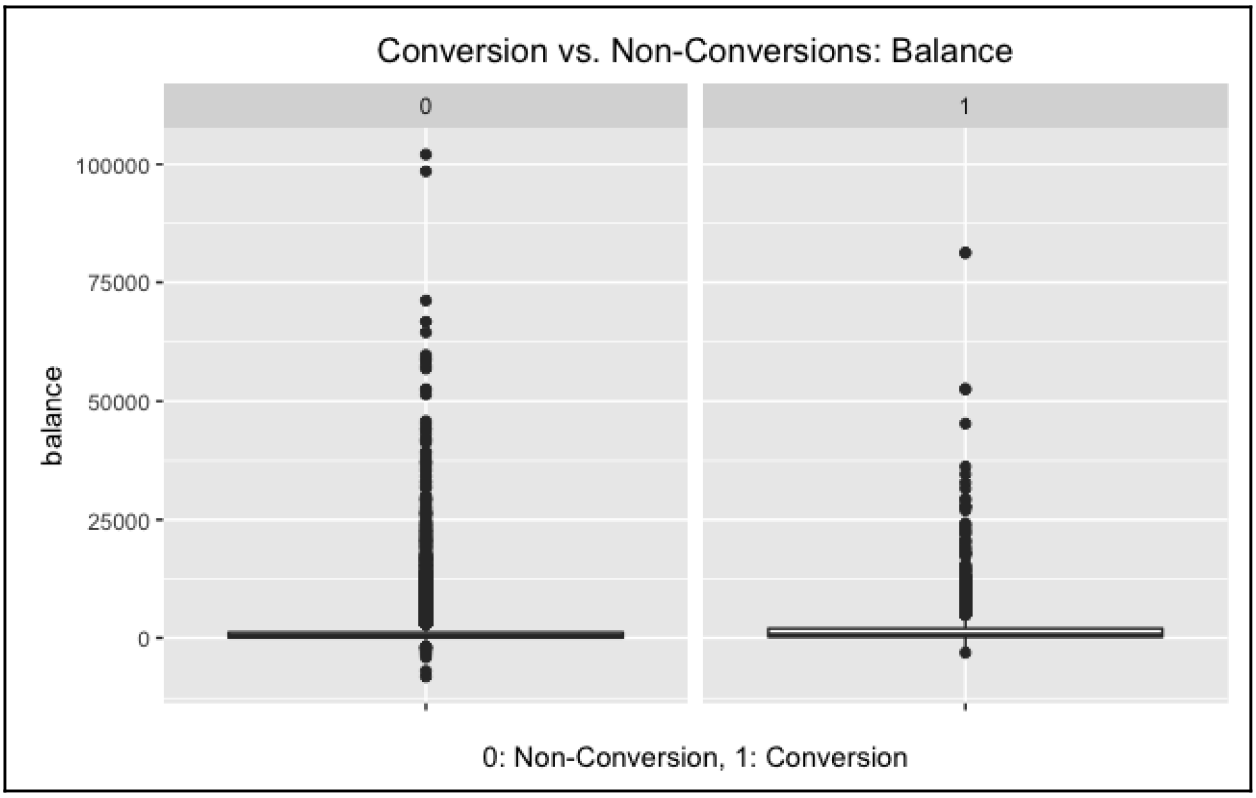
```
> sprintf("conversion rate: %0.2f%%", sum(df$conversion)/nrow(df)*100.0)
[1] "conversion rate: 11.70%"
```

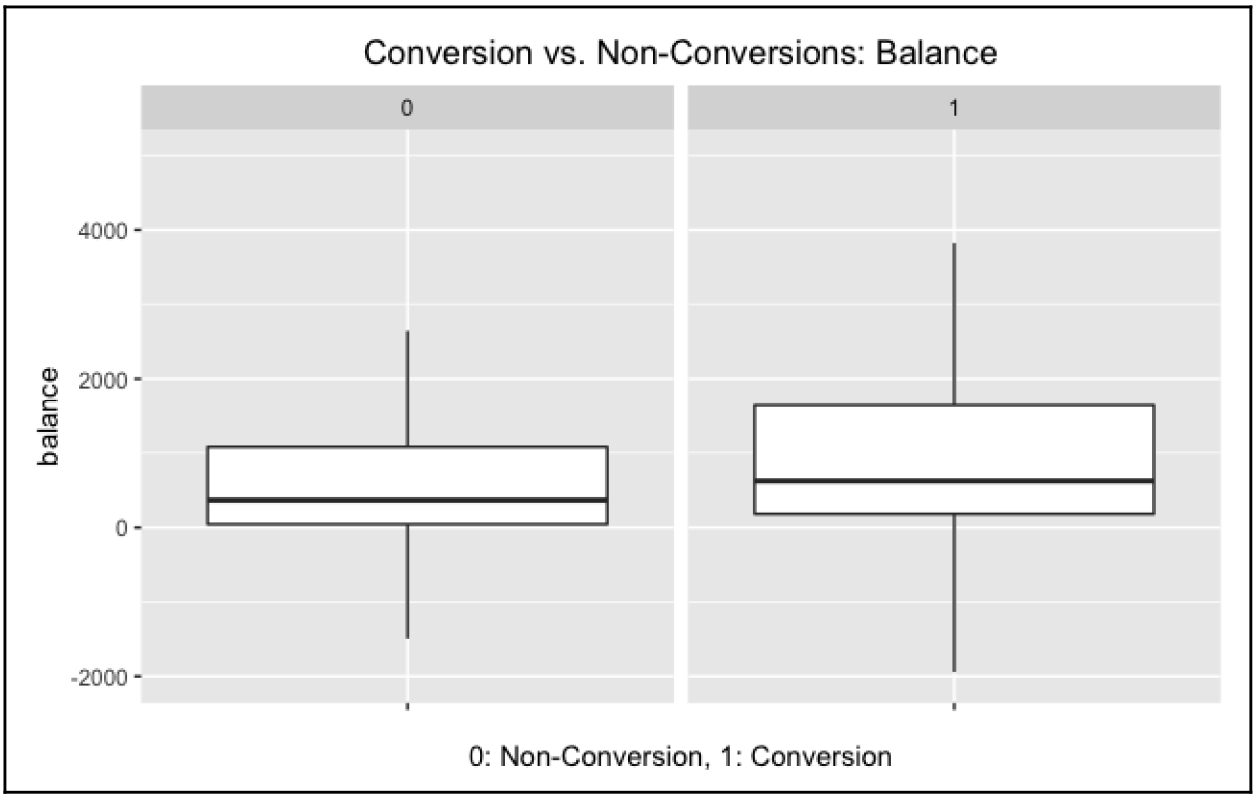
	Job	Count	NumConversions	ConversionRate
1	admin.	5171	631	12.202669
2	blue-collar	9732	708	7.274969
3	entrepreneur	1487	123	8.271688
4	housemaid	1240	109	8.790323
5	management	9458	1301	13.755551
6	retired	2264	516	22.791519
7	self-employed	1579	187	11.842939
8	services	4154	369	8.883004
9	student	938	269	28.678038
10	technician	7597	840	11.056996
11	unemployed	1303	202	15.502686
12	unknown	288	34	11.805556



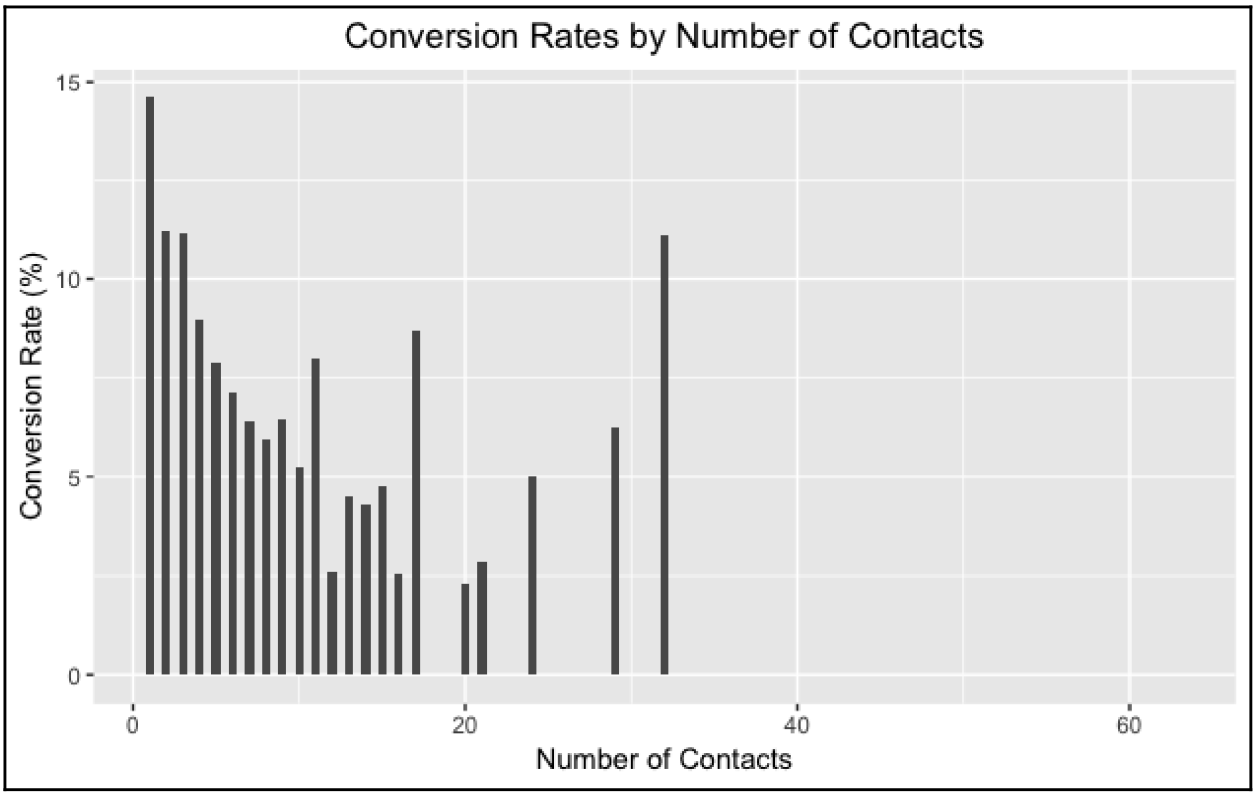
	Default	Conversion	Count
1	no	0	39159
2	no	1	5237
3	yes	0	763
4	yes	1	52


Default (0: Non Conversions, 1: Conversions)
0
1
0.00/1.00
763
39159
52
5237
0.75
0.25
0.50
Default
no
yes


Conversion vs. Non-Conversions: Balance
0
1
100000
75000
50000
25000
0
balance
0: Non-Conversion, 1: Conversion


Conversion vs. Non-Conversions: Balance
0
1
4000
2000
0
-2000
balance
0: Non-Conversion, 1: Conversion

	Campaign	Count	NumConversions	ConversionRate
1	1	17544	2561	14.597583
2	2	12505	1401	11.203519
3	3	5521	618	11.193624
4	4	3522	317	9.000568
5	5	1764	139	7.879819
6	6	1291	92	7.126259
7	7	735	47	6.394558
8	8	540	32	5.925926
9	9	327	21	6.422018
10	10	266	14	5.263158
11	11	201	16	7.960199
12	12	155	4	2.580645
13	13	133	6	4.511278
14	14	93	4	4.301075

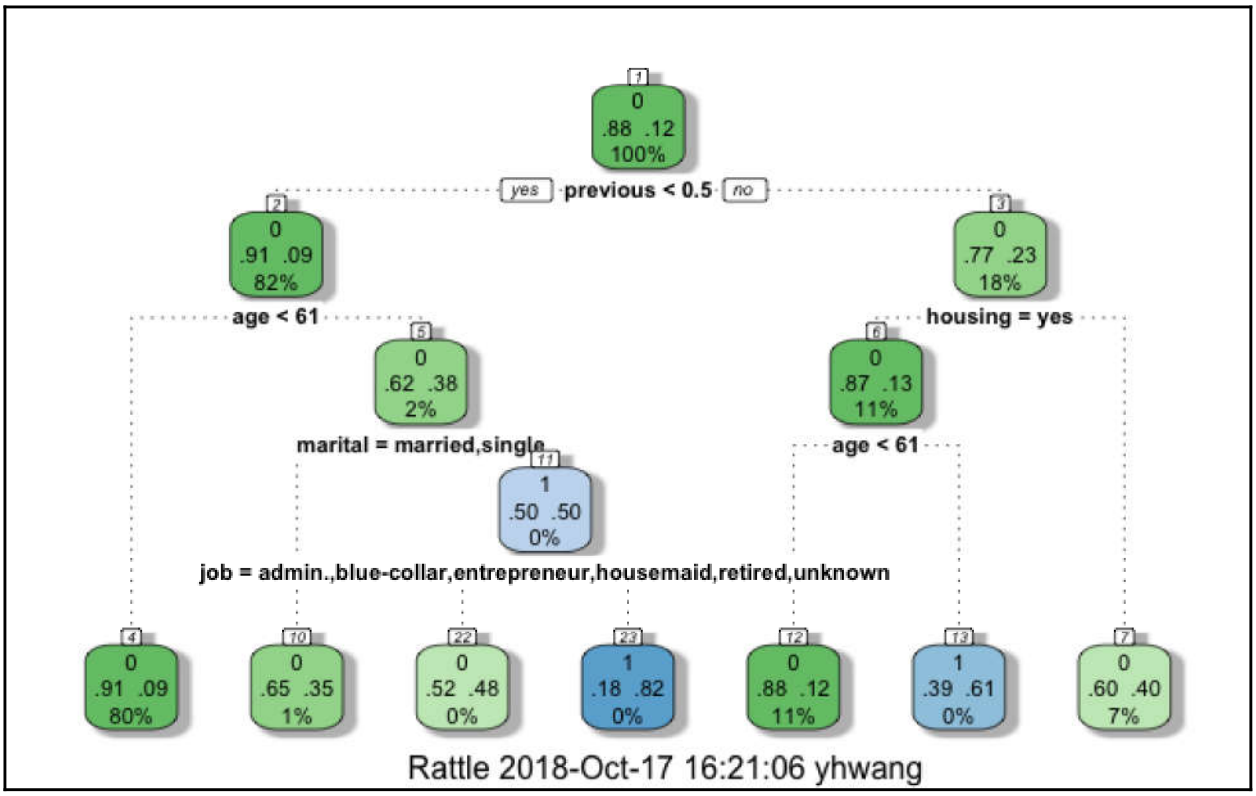



```
> unique(df$month)
 [1] may jun jul aug oct nov dec jan feb mar apr sep
Levels: apr aug dec feb jan jul jun mar may nov oct sep
```

```
> month.abb
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
```

```
> match(unique(df$month), months)
 [1]  5  6  7  8 10 11 12  1  2  3  4  9
```

```
> df %>%
+   group_by(month) %>%
+   summarise(Count=n())
# A tibble: 12 x 2
    month Count
   <fctr> <int>
 1    apr  2932
 2    aug  6247
 3    dec   214
 4    feb  2649
 5    jan  1403
 6    jul  6895
 7    jun  5341
 8    mar   477
 9    may 13766
10    nov  3970
11    oct   738
12    sep   579
```


1
0
.88 .12
100%
yes
previous < 0.5
no
2
0
.91 .09
82%
3
0
.77 .23
18%
age < 61
housing = yes
5
0
.62 .38
2%
6
0
.87 .13
11%
marital = married,single
age < 61
11
1
.50 .50
0%
job = admin.,blue-collar,entrepreneur,housemaid,retired,unknown
4
0
.91 .09
80%
10
0
.65 .35
1%
22
0
.52 .48
0%
23
1
.18 .82
0%
12
0
.88 .12
11%
13
1
.39 .61
0%
7
0
.60 .40
7%
Rattle 2018-Oct-17 16:21:06 yhwang

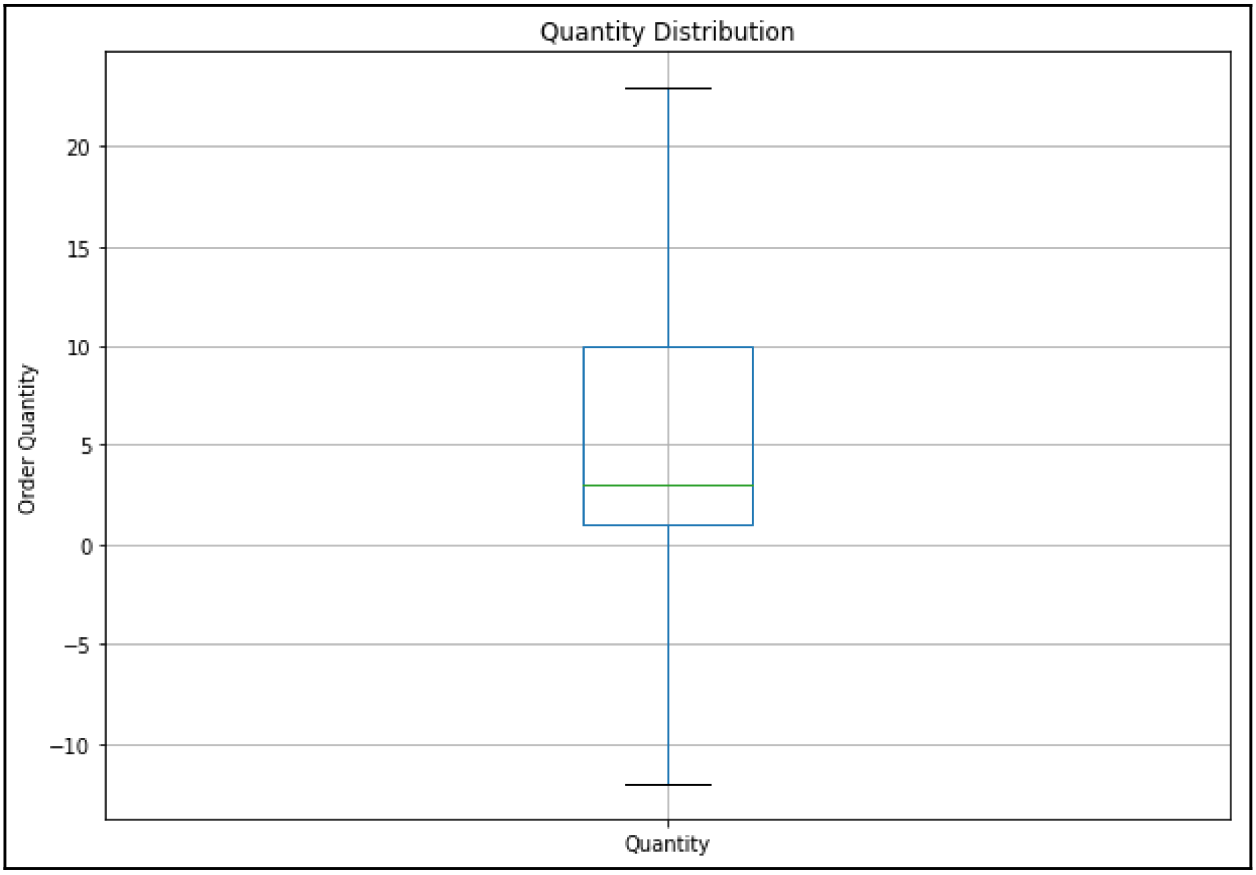
Chapter 5: Product Analytics

```
df.shape
```

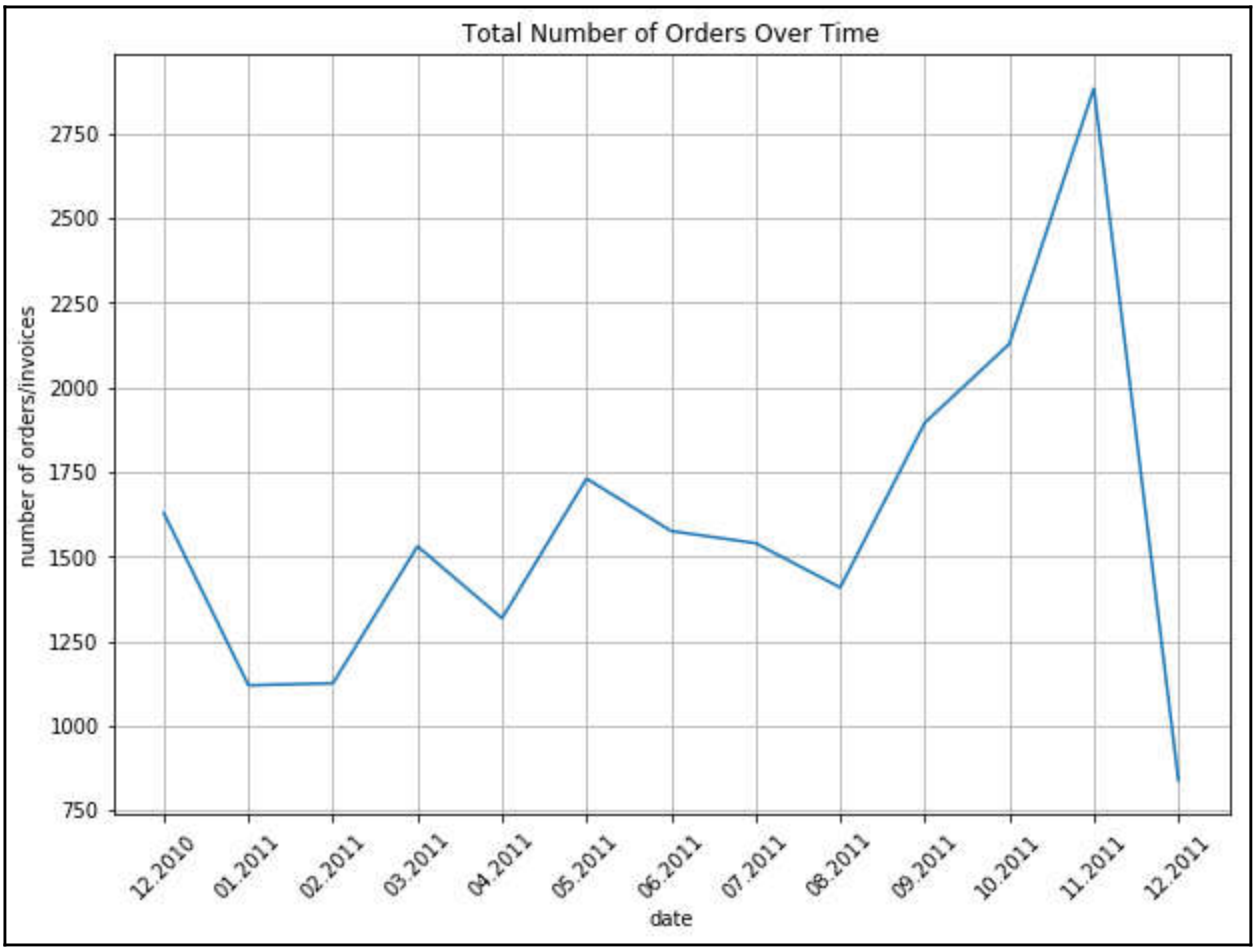
```
(541909, 8)
```

```
df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

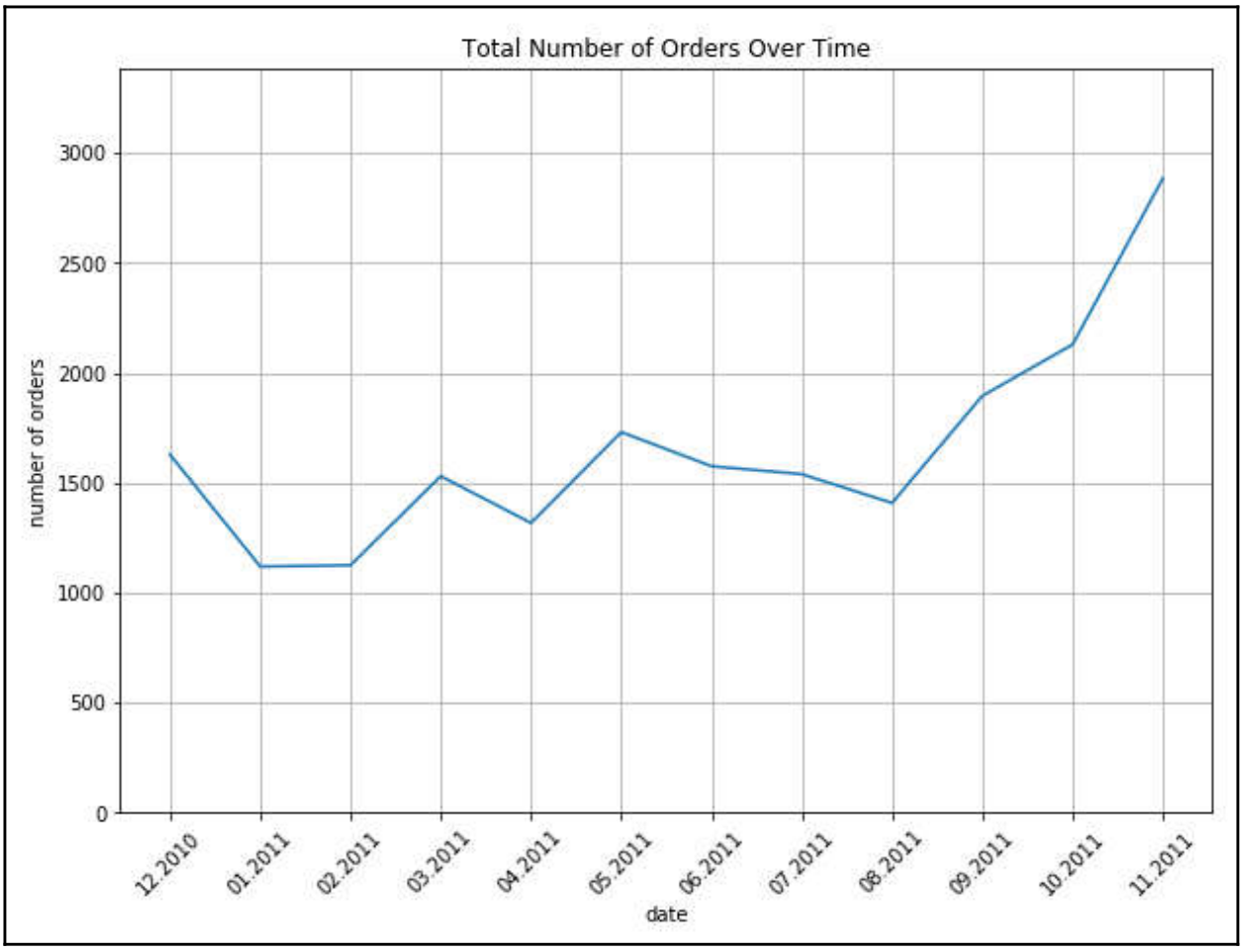


```
InvoiceDate
2010-12-31    1629
2011-01-31    1120
2011-02-28    1126
2011-03-31    1531
2011-04-30    1318
2011-05-31    1731
2011-06-30    1576
2011-07-31    1540
2011-08-31    1409
2011-09-30    1896
2011-10-31    2129
2011-11-30    2884
2011-12-31     839
Freq: M, Name: InvoiceNo, dtype: int64
```



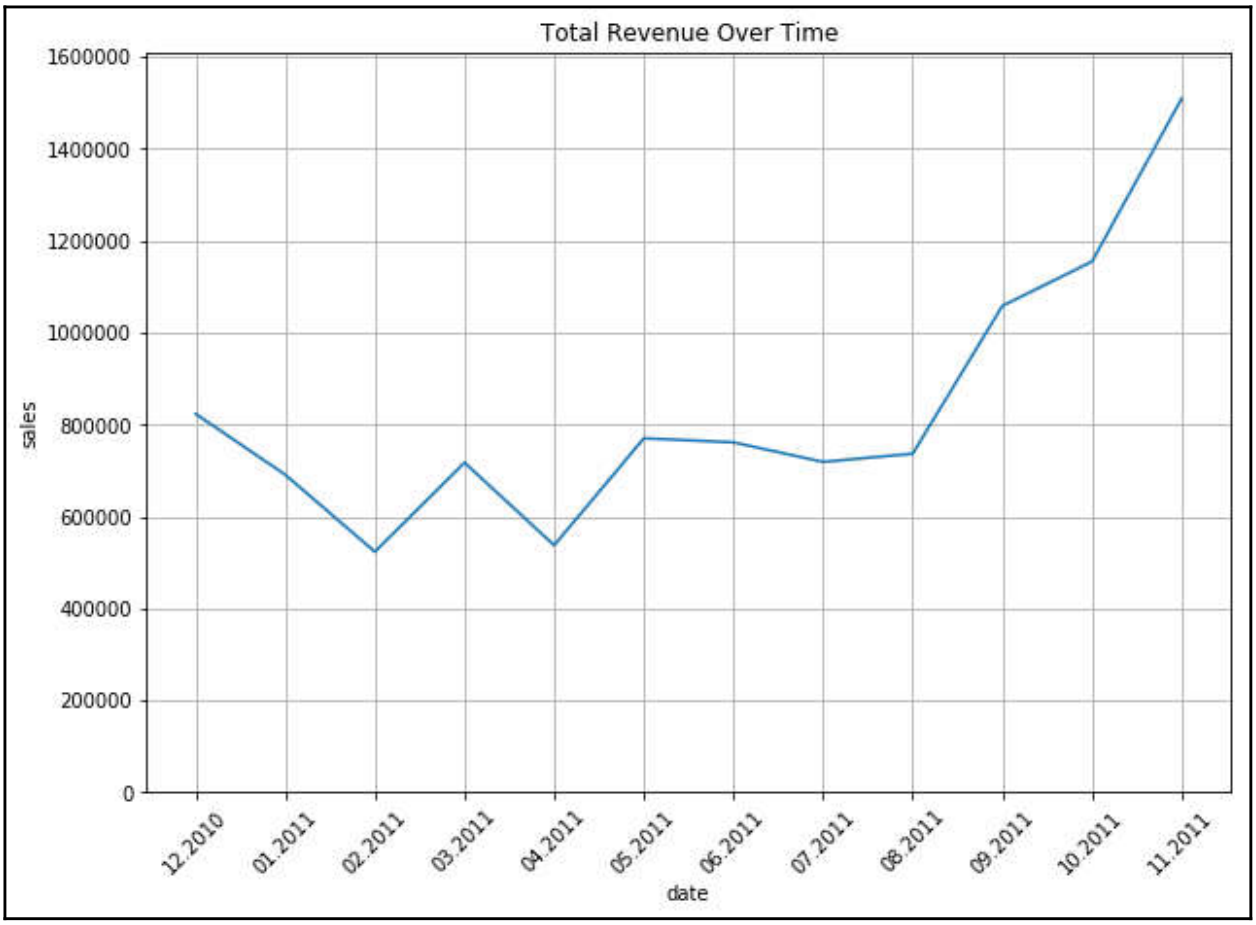

```
print('Min date: %s\nMax date: %s' % (invoice_dates.min(), invoice_dates.max()))
```

```
Min date: 2011-12-01 08:33:00
Max date: 2011-12-09 12:50:00
```


Total Number of Orders Over Time
number of orders
0
500
1000
1500
2000
2500
3000
12.2010
01.2011
02.2011
03.2011
04.2011
05.2011
06.2011
07.2011
08.2011
09.2011
10.2011
11.2011
date

```
monthly_revenue_df
```

```
InvoiceDate
2010-12-31     823746.140
2011-01-31     691364.560
2011-02-28     523631.890
2011-03-31     717639.360
2011-04-30     537808.621
2011-05-31     770536.020
2011-06-30     761739.900
2011-07-31     719221.191
2011-08-31     737014.260
2011-09-30    1058590.172
2011-10-31    1154979.300
2011-11-30    1509496.330
Freq: M, Name: Sales, dtype: float64
```




```
df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Sales
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom	15.30
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom	22.00
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom	20.34

```
invoice_customer_df.head()
```

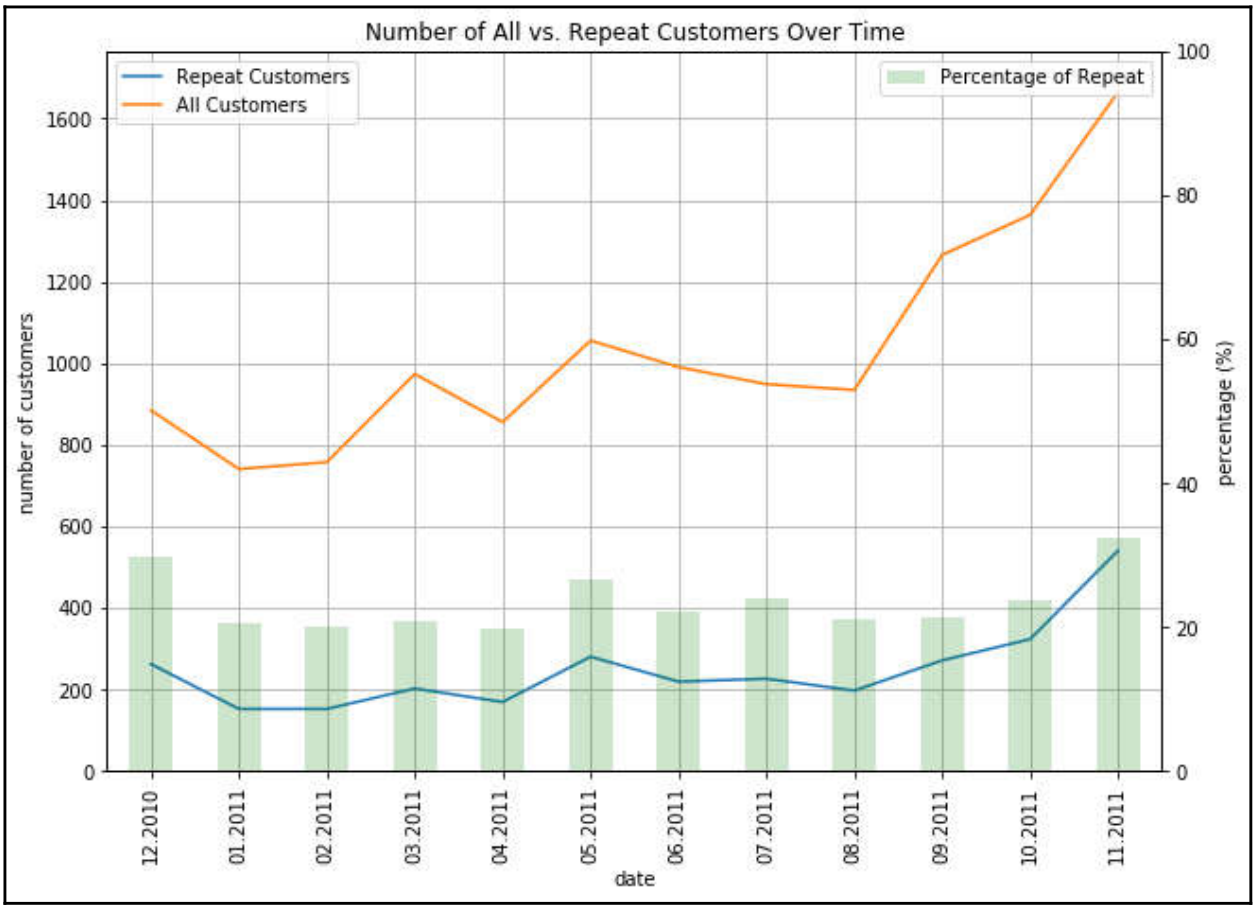
	InvoiceNo	InvoiceDate	Sales	CustomerID	Country
0	536365	2010-12-01 08:26:00	139.12	17850.0	United Kingdom
1	536366	2010-12-01 08:28:00	22.20	17850.0	United Kingdom
2	536367	2010-12-01 08:34:00	278.73	13047.0	United Kingdom
3	536368	2010-12-01 08:34:00	70.05	13047.0	United Kingdom
4	536369	2010-12-01 08:35:00	17.85	13047.0	United Kingdom

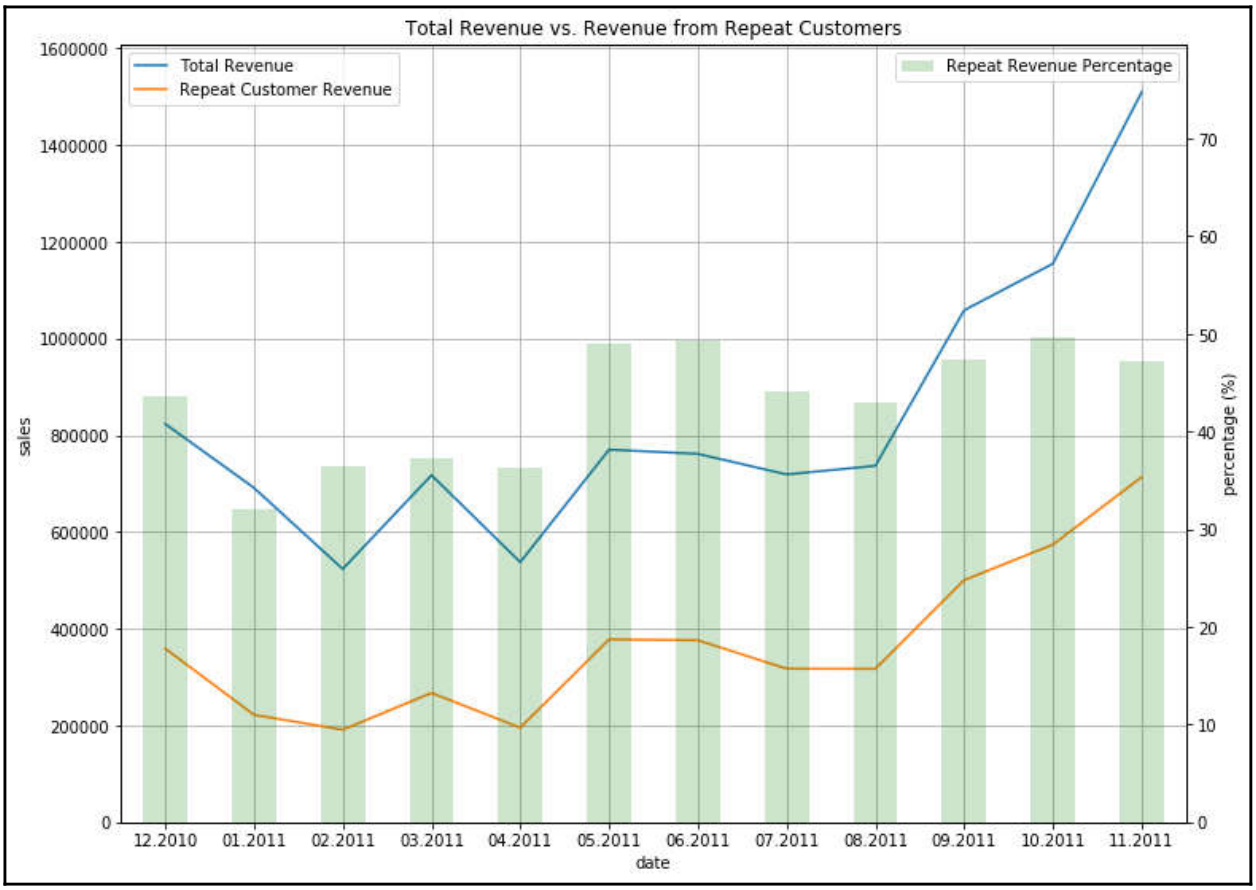
```
monthly_repeat_customers_df
```

```
InvoiceDate
2010-12-31    263
2011-01-31    153
2011-02-28    153
2011-03-31    203
2011-04-30    170
2011-05-31    281
2011-06-30    220
2011-07-31    227
2011-08-31    198
2011-09-30    272
2011-10-31    324
2011-11-30    541
Freq: M, Name: CustomerID, dtype: int64
```

```
monthly_unique_customers_df
```

```
InvoiceDate
2010-12-31     885
2011-01-31     741
2011-02-28     758
2011-03-31     974
2011-04-30     856
2011-05-31    1056
2011-06-30     991
2011-07-31     949
2011-08-31     935
2011-09-30    1266
2011-10-31    1364
2011-11-30    1665
Freq: M, Name: CustomerID, dtype: int64
```


Number of All vs. Repeat Customers Over Time
Repeat Customers
All Customers
Percentage of Repeat
number of customers
percentage (%)
date
0
200
400
600
800
1000
1200
1400
1600
0
20
40
60
80
100
12.2010
01.2011
02.2011
03.2011
04.2011
05.2011
06.2011
07.2011
08.2011
09.2011
10.2011
11.2011

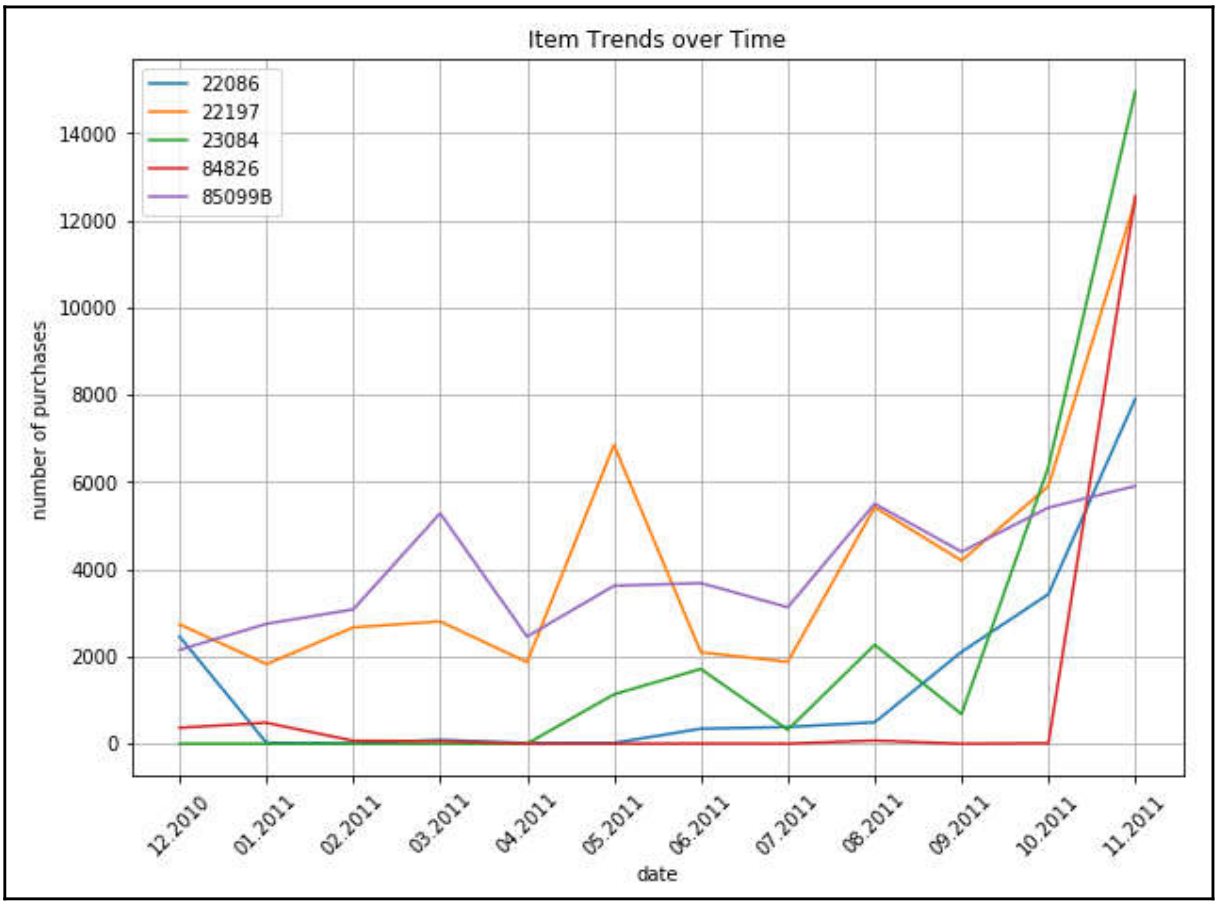

Total Revenue vs. Revenue from Repeat Customers
Total Revenue
Repeat Customer Revenue
Repeat Revenue Percentage
sales
percentage (%)
date
1600000
1400000
1200000
1000000
800000
600000
400000
200000
0
70
60
50
40
30
20
10
0
12.2010
01.2011
02.2011
03.2011
04.2011
05.2011
06.2011
07.2011
08.2011
09.2011
10.2011
11.2011

		Quantity
InvoiceDate	StockCode	
2010-12-31	10002	251
	10120	16
	10125	154
	10133	130
	10135	411
	11001	74
	15034	45
	15036	161
	15039	20

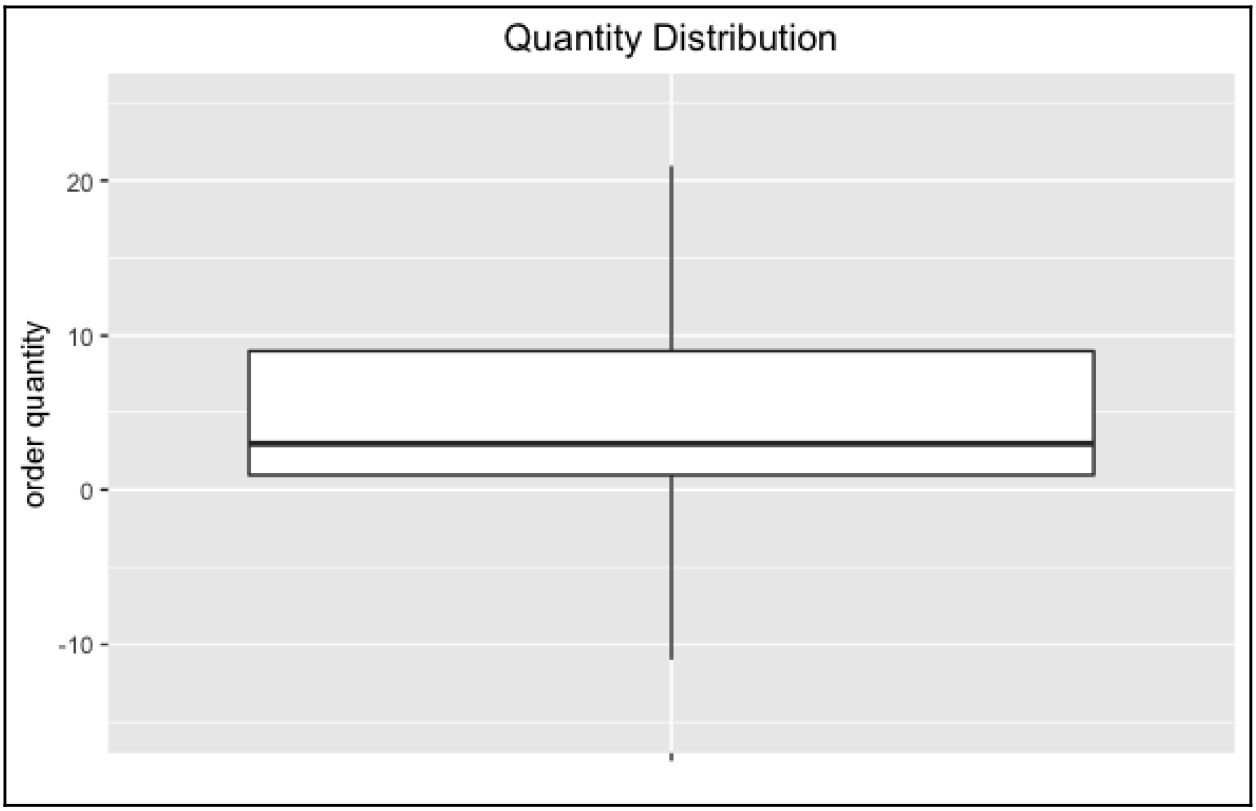
	InvoiceDate	StockCode	Quantity
0	2011-11-30	23084	14954
1	2011-11-30	84826	12551
2	2011-11-30	22197	12460
3	2011-11-30	22086	7908
4	2011-11-30	85099B	5909
5	2011-11-30	22578	5366
6	2011-11-30	84879	5254
7	2011-11-30	22577	5003
8	2011-11-30	85123A	4910
9	2011-11-30	84077	4559

		Quantity
InvoiceDate	StockCode	
2010-12-31	22086	2460
	22197	2738
	84826	366
	85099B	2152
2011-01-31	22086	24
	22197	1824
	84826	480
	85099B	2747
2011-02-28	22086	5
	22197	2666
	84826	66
	85099B	3080

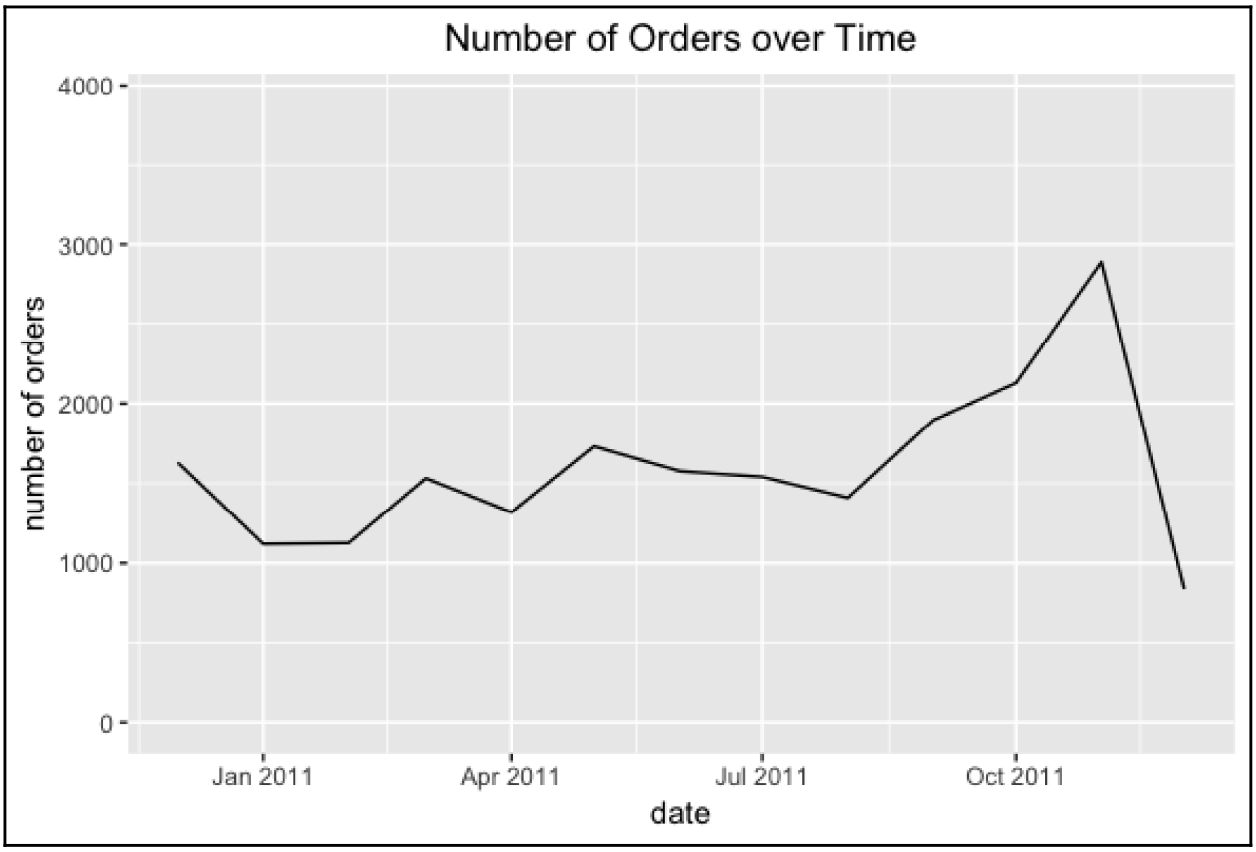
StockCode InvoiceDate	22086	22197	23084	84826	85099B
2010-12-31	2460.0	2738.0	0.0	366.0	2152.0
2011-01-31	24.0	1824.0	0.0	480.0	2747.0
2011-02-28	5.0	2666.0	0.0	66.0	3080.0
2011-03-31	87.0	2803.0	0.0	60.0	5282.0
2011-04-30	13.0	1869.0	0.0	1.0	2456.0
2011-05-31	17.0	6849.0	1131.0	0.0	3621.0
2011-06-30	344.0	2095.0	1713.0	4.0	3682.0
2011-07-31	383.0	1876.0	318.0	2.0	3129.0
2011-08-31	490.0	5421.0	2267.0	72.0	5502.0
2011-09-30	2106.0	4196.0	680.0	0.0	4401.0
2011-10-31	3429.0	5907.0	6348.0	11.0	5412.0
2011-11-30	7908.0	12460.0	14954.0	12551.0	5909.0



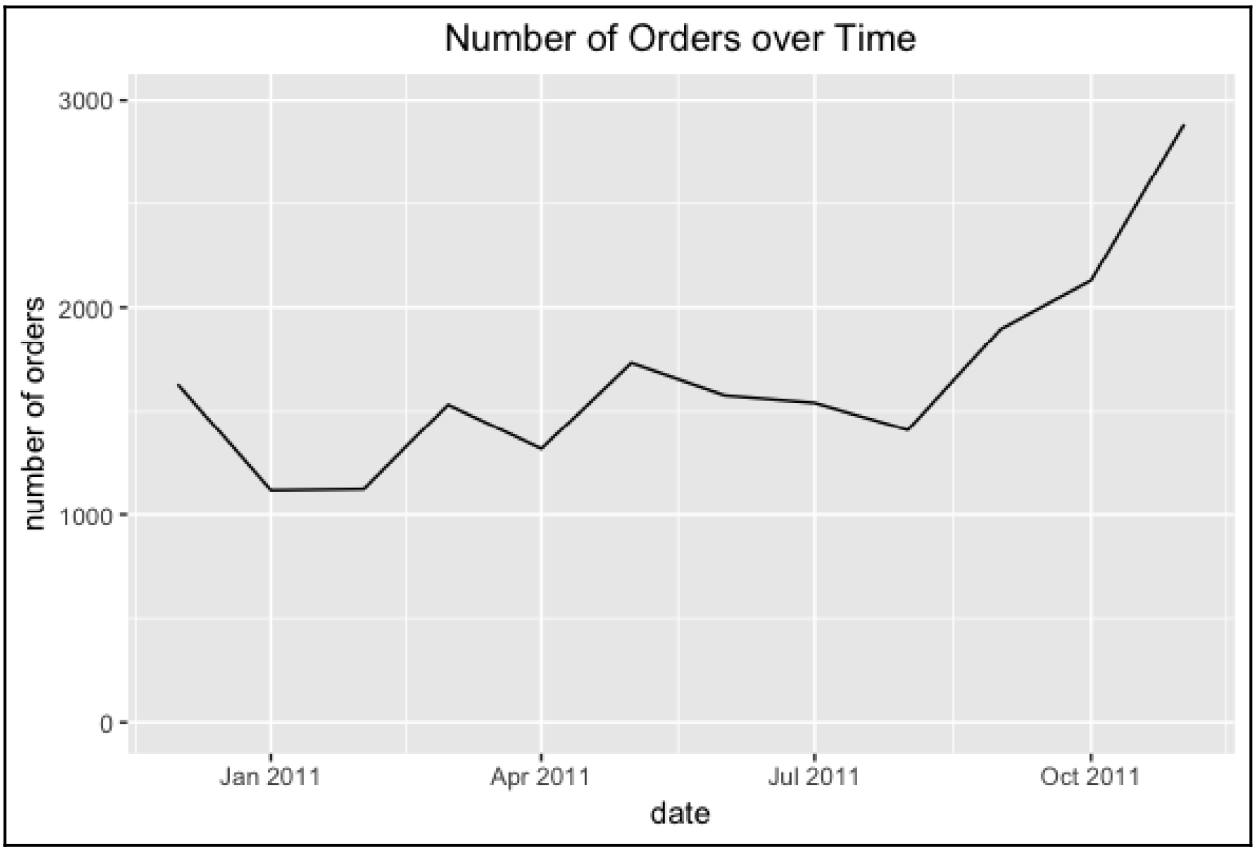
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Sales
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	15.30
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom	22.00
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34
5	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34
6	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850	United Kingdom	15.30
7	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850	United Kingdom	25.50
8	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850	United Kingdom	11.10
9	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850	United Kingdom	11.10
10	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047	United Kingdom	54.08


Quantity Distribution
order quantity
20
10
0
-10

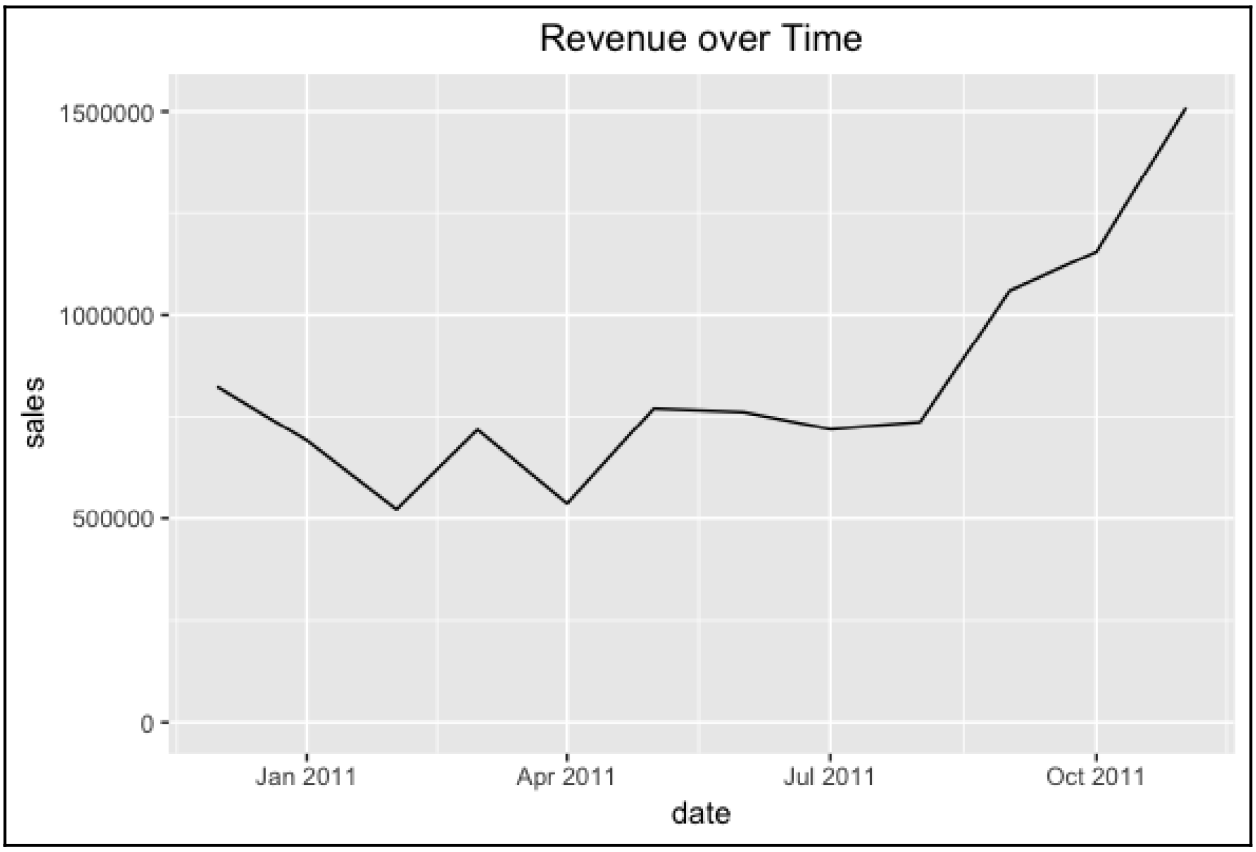
	InvoiceDate	NumOrders
1	2010-12-01	1629
2	2011-01-01	1120
3	2011-02-01	1126
4	2011-03-01	1531
5	2011-04-01	1318
6	2011-05-01	1731
7	2011-06-01	1576
8	2011-07-01	1540
9	2011-08-01	1409
10	2011-09-01	1896
11	2011-10-01	2129
12	2011-11-01	2884




```
> summary(df[which(df$InvoiceDate >= as.Date("2011-12-01")),"InvoiceDate"])
  InvoiceDate
 Min.   :2011-12-01 08:33:00
 1st Qu.:2011-12-04 12:32:00
 Median :2011-12-05 17:28:00
 Mean   :2011-12-05 20:37:49
 3rd Qu.:2011-12-08 09:20:00
 Max.   :2011-12-09 12:50:00
```


Number of Orders over Time
3000
2000
1000
0
number of orders
Jan 2011
Apr 2011
Jul 2011
Oct 2011
date

	InvoiceDate	Sales
1	2010-12-01	823746.1
2	2011-01-01	691364.6
3	2011-02-01	523631.9
4	2011-03-01	717639.4
5	2011-04-01	537808.6
6	2011-05-01	770536.0
7	2011-06-01	761739.9
8	2011-07-01	719221.2
9	2011-08-01	737014.3
10	2011-09-01	1058590.2
11	2011-10-01	1154979.3
12	2011-11-01	1509496.3



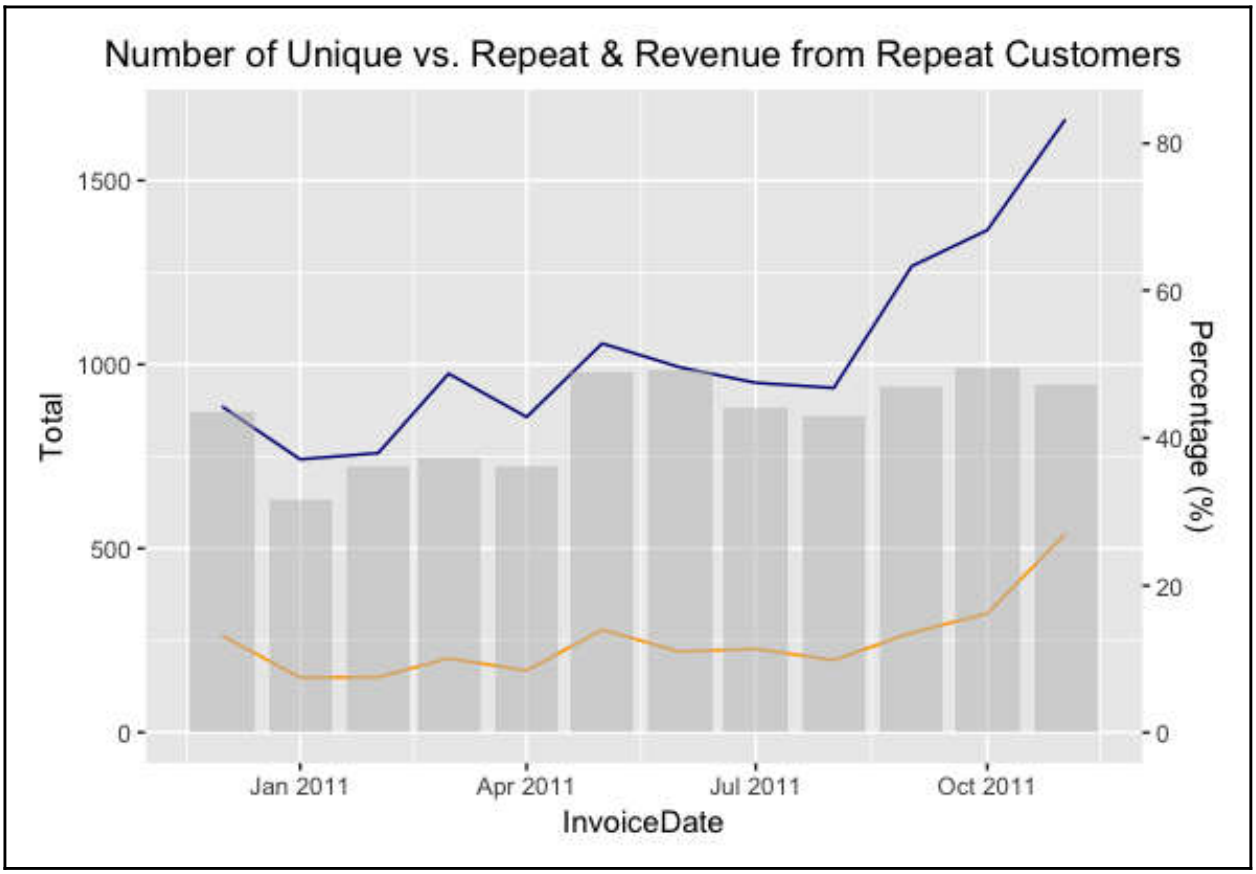
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Sales
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom	15.30
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom	22.00
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34
5	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom	20.34
6	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850	United Kingdom	15.30
7	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850	United Kingdom	25.50
8	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850	United Kingdom	11.10
9	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850	United Kingdom	11.10
10	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047	United Kingdom	54.08

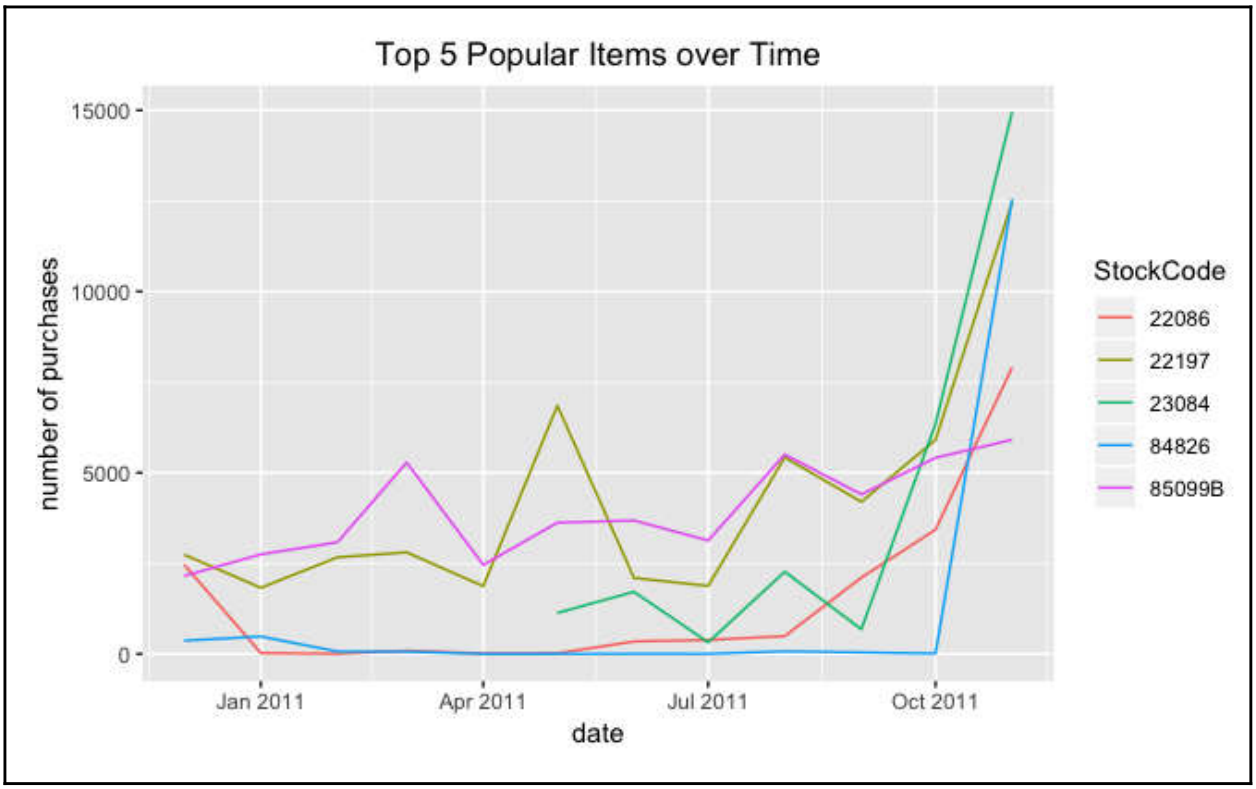
	InvoiceNo	InvoiceDate	CustomerID	Sales
1	536365	2010-12-01 08:26:00	17850	139.12
2	536366	2010-12-01 08:28:00	17850	22.20
3	536367	2010-12-01 08:34:00	13047	278.73
4	536368	2010-12-01 08:34:00	13047	70.05
5	536369	2010-12-01 08:35:00	13047	17.85
6	536370	2010-12-01 08:45:00	12583	855.86
7	536371	2010-12-01 09:00:00	13748	204.00
8	536372	2010-12-01 09:01:00	17850	22.20
9	536373	2010-12-01 09:02:00	17850	259.86
10	536374	2010-12-01 09:09:00	15100	350.40

	InvoiceDate	CustomerID	Count	Sales
1	2010-12-01	12347	1	711.79
2	2010-12-01	12348	1	892.80
3	2010-12-01	12370	2	1868.02
4	2010-12-01	12377	1	1001.52
5	2010-12-01	12383	1	600.72
6	2010-12-01	12386	1	258.90
7	2010-12-01	12395	2	679.92
8	2010-12-01	12417	1	291.34
9	2010-12-01	12423	1	237.93
10	2010-12-01	12427	1	303.50

	InvoiceDate	Count	Sales
1	2010-12-01	263	359170.6
2	2011-01-01	149	219339.8
3	2011-02-01	150	190084.8
4	2011-03-01	201	266773.7
5	2011-04-01	168	194860.1
6	2011-05-01	279	377802.3
7	2011-06-01	219	376084.2
8	2011-07-01	227	317475.0
9	2011-08-01	196	316278.3
10	2011-09-01	271	496818.5
11	2011-10-01	323	573221.8
12	2011-11-01	540	713522.2

	InvoiceDate	Count
1	2010-12-01	886
2	2011-01-01	742
3	2011-02-01	759
4	2011-03-01	975
5	2011-04-01	857
6	2011-05-01	1057
7	2011-06-01	992
8	2011-07-01	950
9	2011-08-01	936
10	2011-09-01	1267
11	2011-10-01	1365
12	2011-11-01	1666


Number of Unique vs. Repeat & Revenue from Repeat Customers
Total
1500
1000
500
0
80
60
40
20
0
Percentage (%)
Jan 2011
Apr 2011
Jul 2011
Oct 2011
InvoiceDate


Top 5 Popular Items over Time
15000
10000
5000
0
number of purchases
Jan 2011
Apr 2011
Jul 2011
Oct 2011
date
StockCode
22086
22197
23084
84826
85099B

Chapter 6: Recommending the Right Products

		Items				
		A	B	C	D	E
Users	1	0	1	0	1	0
	2	1	1	1	0	1
	3	0	0	1	0	0
	4	1	0	1	0	1
	5	0	1	0	0	1

```
df = pd.read_excel(io='../data/Online Retail.xlsx', sheet_name='Online Retail')
```

```
df.shape
```

```
(541909, 8)
```

```
df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

```
df['CustomerID'].isna().sum()
```

```
133361
```

```
df.loc[df['CustomerID'].isna()].head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
622	536414	22139	NaN	56	2010-12-01 11:52:00	0.00	NaN	United Kingdom
1443	536544	21773	DECORATIVE ROSE BATHROOM BOTTLE	1	2010-12-01 14:32:00	2.51	NaN	United Kingdom
1444	536544	21774	DECORATIVE CATS BATHROOM BOTTLE	2	2010-12-01 14:32:00	2.51	NaN	United Kingdom
1445	536544	21786	POLKADOT RAIN HAT	4	2010-12-01 14:32:00	0.85	NaN	United Kingdom
1446	536544	21787	RAIN PONCHO RETROSPOT	2	2010-12-01 14:32:00	1.66	NaN	United Kingdom

```
df.shape
```

```
(531285, 8)
```

```
df = df.dropna(subset=['CustomerID'])
```

```
df.shape
```

```
(397924, 8)
```

StockCode	10002	10080	10120	10125	10133	10135	11001	15030	15034	15036
CustomerID										
12481.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	36.0
12483.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
12484.0	NaN	NaN	NaN	NaN	NaN	NaN	16.0	NaN	NaN	NaN
12488.0	NaN	NaN	NaN	NaN	NaN	10.0	NaN	NaN	NaN	NaN
12489.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

StockCode	10002	10080	10120	10125	10133	10135	11001	15030	15034	15036
CustomerID										
12481.0	0	0	0	0	0	0	0	0	0	1
12483.0	0	0	0	0	0	0	0	0	0	0
12484.0	0	0	0	0	0	0	1	0	0	0
12488.0	0	0	0	0	0	1	0	0	0	0
12489.0	0	0	0	0	0	0	0	0	0	0

```
user_user_sim_matrix.head()
```

	0	1	2	3	4	5	6	7	8	9	...	4329	4330	4331	4332	4333	4334	4335	
0	1.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0.
1	0.0	1.000000	0.063022	0.046130	0.047795	0.038484	0.0	0.025876	0.136641	0.094742	...	0.0	0.029709	0.052668	0.0	0.032844	0.062318	0.0	0.
2	0.0	0.063022	1.000000	0.024953	0.051709	0.027756	0.0	0.027995	0.118262	0.146427	...	0.0	0.064282	0.113961	0.0	0.000000	0.000000	0.0	0.
3	0.0	0.046130	0.024953	1.000000	0.056773	0.137137	0.0	0.030737	0.032461	0.144692	...	0.0	0.105868	0.000000	0.0	0.039014	0.000000	0.0	0.
4	0.0	0.047795	0.051709	0.056773	1.000000	0.031575	0.0	0.000000	0.000000	0.033315	...	0.0	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0.

5 rows × 4339 columns

```
user_user_sim_matrix.head()
```

CustomerID	12346.0	12347.0	12348.0	12349.0	12350.0	12352.0	12353.0	12354.0	12355.0	12356.0	...	18273.0	18274.0	18276.0	18277.0	182
CustomerID																
12346.0	1.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	0.000000	...	0.0	0.000000	0.000000	0.0	0.000
12347.0	0.0	1.000000	0.063022	0.046130	0.047795	0.038484	0.0	0.025876	0.136641	0.094742	...	0.0	0.029709	0.052668	0.0	0.032
12348.0	0.0	0.063022	1.000000	0.024953	0.051709	0.027756	0.0	0.027995	0.118262	0.146427	...	0.0	0.064282	0.113961	0.0	0.000
12349.0	0.0	0.046130	0.024953	1.000000	0.056773	0.137137	0.0	0.030737	0.032461	0.144692	...	0.0	0.105868	0.000000	0.0	0.039
12350.0	0.0	0.047795	0.051709	0.056773	1.000000	0.031575	0.0	0.000000	0.000000	0.033315	...	0.0	0.000000	0.000000	0.0	0.000

5 rows × 4339 columns

```
user_user_sim_matrix.loc[12350.0].sort_values(ascending=False)
```

```
CustomerID
12350.0    1.000000
17935.0    0.183340
12414.0    0.181902
12652.0    0.175035
16692.0    0.171499
16754.0    0.171499
12814.0    0.171499
12791.0    0.171499
16426.0    0.166968
16333.0    0.161690
12475.0    0.161690
```

```
items_to_recommend_to_B
```

```
{20615,
 20652,
 21171,
 21832,
 21864,
 21908,
 21915,
 22348,
 22412,
 22620,
 '79066K',
 '79191C',
 '84086C'}
```

```
df.loc[
    df['StockCode'].isin(items_to_recommend_to_B),
    ['StockCode', 'Description']
].drop_duplicates().set_index('StockCode')
```

StockCode	Description
21832	CHOCOLATE CALCULATOR
21915	RED HARMONICA IN BOX
22620	4 TRADITIONAL SPINNING TOPS
79066K	RETRO MOD TRAY
21864	UNION JACK FLAG PASSPORT COVER
79191C	RETRO PLASTIC ELEPHANT TRAY
21908	CHOCOLATE THIS WAY METAL SIGN
20615	BLUE POLKADOT PASSPORT COVER
20652	BLUE POLKADOT LUGGAGE TAG
22348	TEA BAG PLATE RED RETROSPOT
22412	METAL SIGN NEIGHBOURHOOD WITCH
21171	BATHROOM METAL SIGN
84086C	PINK/PURPLE RETRO RADIO

```
item_item_sim_matrix
```

StockCode	10002	10080	10120	10125	10133	10135	11001	15030	15034	15036	...	90214V	90214W	90214Y	90214Z
StockCode															
10002	1.000000	0.000000	0.094868	0.090351	0.062932	0.098907	0.095346	0.047673	0.075593	0.090815	...	0.000000	0.000000	0.000000	0.000000
10080	0.000000	1.000000	0.000000	0.032774	0.045655	0.047836	0.000000	0.000000	0.082261	0.049413	...	0.000000	0.000000	0.000000	0.000000
10120	0.094868	0.000000	1.000000	0.057143	0.059702	0.041703	0.060302	0.060302	0.095618	0.028718	...	0.000000	0.000000	0.000000	0.000000
10125	0.090351	0.032774	0.057143	1.000000	0.042644	0.044682	0.043073	0.000000	0.051224	0.030770	...	0.000000	0.000000	0.000000	0.000000
10133	0.062932	0.045655	0.059702	0.042644	1.000000	0.280097	0.045002	0.060003	0.071358	0.057152	...	0.000000	0.000000	0.000000	0.000000
10135	0.098907	0.047836	0.041703	0.044682	0.280097	1.000000	0.094304	0.062869	0.074767	0.044911	...	0.073721	0.000000	0.060193	0.000000
11001	0.095346	0.000000	0.060302	0.043073	0.045002	0.094304	1.000000	0.045455	0.072075	0.075765	...	0.000000	0.000000	0.000000	0.000000
15030	0.047673	0.000000	0.060302	0.000000	0.060003	0.062869	0.045455	1.000000	0.108112	0.129884	...	0.000000	0.000000	0.000000	0.000000
15034	0.075593	0.082261	0.095618	0.051224	0.071358	0.074767	0.072075	0.108112	1.000000	0.231694	...	0.000000	0.000000	0.000000	0.000000
15036	0.090815	0.049413	0.028718	0.030770	0.057152	0.044911	0.075765	0.129884	0.231694	1.000000	...	0.000000	0.000000	0.000000	0.000000
15039	0.062284	0.030124	0.026261	0.056274	0.052262	0.054759	0.019795	0.158362	0.235412	0.207400	...	0.000000	0.000000	0.000000	0.000000
16008	0.043033	0.062439	0.027217	0.077762	0.121867	0.014188	0.041030	0.123091	0.081325	0.078161	...	0.000000	0.000000	0.000000	0.000000

```
top_10_similar_items
```

```
[23166, 23165, 23167, 22993, 23307, 22722, 22720, 22666, 23243, 22961]
```

StockCode	Description
23166	MEDIUM CERAMIC TOP STORAGE JAR
23165	LARGE CERAMIC TOP STORAGE JAR
23167	SMALL CERAMIC TOP STORAGE JAR
22993	SET OF 4 PANTRY JELLY MOULDS
23307	SET OF 60 PANTRY DESIGN CAKE CASES
22722	SET OF 6 SPICE TINS PANTRY DESIGN
22720	SET OF 3 CAKE TINS PANTRY DESIGN
22666	RECIPE BOX PANTRY YELLOW DESIGN
23243	SET OF TEA COFFEE SUGAR TINS PANTRY
22961	JAM MAKING SET PRINTED

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
5	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
6	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850	United Kingdom
7	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850	United Kingdom
8	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
9	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
10	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047	United Kingdom

```
> sum(is.na(df$CustomerID))
[1] 133361
```

```
> head(df[which(is.na(df$CustomerID)),])
# A tibble: 6 x 8
  InvoiceNo StockCode                   Description Quantity         InvoiceDate UnitPrice CustomerID        Country
      <chr>     <chr>                         <chr>    <dbl>              <dttm>     <dbl>      <dbl>          <chr>
1    536414     22139                          <NA>       56 2010-12-01 11:52:00      0.00         NA United Kingdom
2    536544     21773 DECORATIVE ROSE BATHROOM BOTTLE        1 2010-12-01 14:32:00      2.51         NA United Kingdom
3    536544     21774 DECORATIVE CATS BATHROOM BOTTLE        2 2010-12-01 14:32:00      2.51         NA United Kingdom
4    536544     21786               POLKADOT RAIN HAT        4 2010-12-01 14:32:00      0.85         NA United Kingdom
5    536544     21787           RAIN PONCHO RETROSPOT        2 2010-12-01 14:32:00      1.66         NA United Kingdom
6    536544     21790              VINTAGE SNAP CARDS        9 2010-12-01 14:32:00      1.66         NA United Kingdom
```

```
> # current DataFrame shape
> dim(df)
[1] 531285      8
>
> # remove records with NA
> df <- na.omit(df)
> dim(df)
[1] 397924      8
```

	CustomerID	10002	10080	10120	10123C	10124A	10124G	10125	10133	10135	11001	15030	15034	15036	15039
315	12731	1	0	0	0	0	0	1	0	0	0	0	0	0	0
316	12732	0	0	0	0	0	0	0	0	0	0	0	0	0	0
317	12733	0	0	0	0	0	0	0	0	0	0	0	0	0	0
318	12734	0	0	0	0	0	0	0	0	0	0	0	0	0	0
319	12735	0	0	0	0	0	0	1	0	0	0	0	0	0	0
320	12736	0	0	0	0	0	0	0	0	0	0	0	0	0	0
321	12738	0	0	0	0	0	0	0	0	0	0	0	0	0	0
322	12739	0	0	0	0	0	0	0	0	0	0	0	0	0	0
323	12740	0	0	0	0	0	0	0	0	0	0	0	0	0	0
324	12743	0	0	0	0	0	0	0	0	0	0	0	0	0	0
325	12744	0	0	0	0	0	0	0	0	0	0	0	0	0	0
326	12747	0	0	0	0	0	0	0	0	0	0	0	0	0	0
327	12748	1	0	1	0	0	0	0	1	1	1	1	1	1	1

	CustomerID	10002	10080	10120	10123C	10124A	10124G	10125	10133	10135	11001	15030	15034	15036	15039
315	12731	3	0	0	0	0	0	5	0	0	0	0	0	0	0
316	12732	0	0	0	0	0	0	0	0	0	0	0	0	0	0
317	12733	0	0	0	0	0	0	0	0	0	0	0	0	0	0
318	12734	0	0	0	0	0	0	0	0	0	0	0	0	0	0
319	12735	0	0	0	0	0	0	1	0	0	0	0	0	0	0
320	12736	0	0	0	0	0	0	0	0	0	0	0	0	0	0
321	12738	0	0	0	0	0	0	0	0	0	0	0	0	0	0
322	12739	0	0	0	0	0	0	0	0	0	0	0	0	0	0
323	12740	0	0	0	0	0	0	0	0	0	0	0	0	0	0
324	12743	0	0	0	0	0	0	0	0	0	0	0	0	0	0
325	12744	0	0	0	0	0	0	0	0	0	0	0	0	0	0
326	12747	0	0	0	0	0	0	0	0	0	0	0	0	0	0
327	12748	1	0	2	0	0	0	0	5	1	2	1	5	5	2

	12346	12347	12348	12349	12350	12352	12353	12354	12355	12356	12357
1	1.00000000	0.000000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
2	0.00000000	1.000000000	0.06302187	0.04612963	0.04779549	0.03848368	0.00000000	0.02587601	0.13664059	0.09474177	0.06026203
3	0.00000000	0.063021872	1.00000000	0.02495326	0.05170877	0.02775637	0.00000000	0.02799463	0.11826248	0.14642685	0.00000000
4	0.00000000	0.046129628	0.02495326	1.00000000	0.05677330	0.13713714	0.00000000	0.03073651	0.03246137	0.14469154	0.15338899
5	0.00000000	0.047795490	0.05170877	0.05677330	1.00000000	0.03157545	0.00000000	0.00000000	0.00000000	0.03331483	0.02119044
6	0.00000000	0.038483684	0.02775637	0.13713714	0.03157545	1.00000000	0.00000000	0.10256785	0.03610791	0.08941411	0.06824795
7	0.00000000	0.000000000	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000	0.00000000	0.13867505	0.06868028	0.13105561
8	0.00000000	0.025876015	0.02799463	0.03073651	0.00000000	0.10256785	0.00000000	1.00000000	0.00000000	0.05410898	0.06883378
9	0.00000000	0.136640586	0.11826248	0.03246137	0.00000000	0.03610791	0.13867505	0.00000000	1.00000000	0.15238786	0.07269657
10	0.00000000	0.094741770	0.14642685	0.14469154	0.03331483	0.08941411	0.06868028	0.05410898	0.15238786	1.00000000	0.09600999
11	0.00000000	0.060262033	0.00000000	0.15338899	0.02119044	0.06824795	0.13105561	0.06883378	0.07269657	0.09600999	1.00000000
12	0.00000000	0.000000000	0.05913124	0.12984549	0.06726728	0.10832372	0.13867505	0.00000000	0.00000000	0.15238786	0.14539314
13	0.00000000	0.141447018	0.01457410	0.13601306	0.00000000	0.13349296	0.00000000	0.05385554	0.07583705	0.03755910	0.10153281
14	0.00000000	0.057694975	0.02080626	0.09137637	0.07100716	0.05082055	0.00000000	0.25628391	0.02706660	0.08043011	0.08526479

```
> top10SimilarCustomersTo12350
 [1] 12350 17935 12414 12652 12791 12814 16692 16754 16426 12475 16333
```

```
> itemsBoughtByA
 [1] "CustomerID" "20615"      "20652"      "21171"      "21832"      "21864"      "21866"
 [8] "21908"      "21915"      "22348"      "22412"      "22551"      "22557"      "22620"
[15] "79066K"     "79191C"     "84086C"     "POST"
```

```
> itemsBoughtByB
 [1] "CustomerID" "20657"      "20659"      "20828"      "20856"      "21051"      "21866"
 [8] "21867"      "22208"      "22209"      "22210"      "22211"      "22449"      "22450"
[15] "22551"      "22553"      "22557"      "22640"      "22659"      "22749"      "22752"
[22] "22753"      "22754"      "22755"      "23290"      "23292"      "23309"      "85099B"
[29] "POST"
```

```
> itemsToRecommendToB
 [1] "20615"  "20652"  "21171"  "21832"  "21864"  "21908"  "21915"  "22348"  "22412"  "22620"
[11] "79066K" "79191C" "84086C"
```

	StockCode	Description
1	20615	BLUE POLKADOT PASSPORT COVER
2	20652	BLUE POLKADOT LUGGAGE TAG
3	21171	BATHROOM METAL SIGN
4	21832	CHOCOLATE CALCULATOR
5	21864	UNION JACK FLAG PASSPORT COVER
6	21908	CHOCOLATE THIS WAY METAL SIGN
7	21915	RED HARMONICA IN BOX
8	22348	TEA BAG PLATE RED RETROSPOT
9	22412	METAL SIGN NEIGHBOURHOOD WITCH
10	22620	4 TRADITIONAL SPINNING TOPS
11	79066K	RETRO MOD TRAY
12	79191C	RETRO PLASTIC ELEPHANT TRAY
13	84086C	PINK/PURPLE RETRO RADIO

	10002	10080	10120	10123C	10124A	10124G	10125	10133	10135	11001	15030	15034
10002	1.00000000	0.00000000	0.09486833	0.09128709	0.00000000	0.00000000	0.09035079	0.06293168	0.09890707	0.09534626	0.04767313	0.07559289
10080	0.00000000	1.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.03277368	0.04565544	0.04783649	0.00000000	0.00000000	0.08226127
10120	0.09486833	0.00000000	1.00000000	0.11547005	0.00000000	0.00000000	0.05714286	0.05970223	0.04170288	0.06030227	0.06030227	0.09561829
10123C	0.09128709	0.00000000	0.11547005	1.00000000	0.00000000	0.00000000	0.16495722	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
10124A	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000	0.44721360	0.06388766	0.04449942	0.00000000	0.00000000	0.00000000	0.00000000
10124G	0.00000000	0.00000000	0.00000000	0.00000000	0.44721360	1.00000000	0.07142857	0.04975186	0.00000000	0.00000000	0.00000000	0.00000000
10125	0.09035079	0.03277368	0.05714286	0.16495722	0.06388766	0.07142857	1.00000000	0.04264445	0.04468166	0.04307305	0.00000000	0.05122408
10133	0.06293168	0.04565544	0.05970223	0.00000000	0.04449942	0.04975186	0.04264445	1.00000000	0.28009746	0.04500225	0.06000300	0.07135782
10135	0.09890707	0.04783649	0.04170288	0.00000000	0.00000000	0.00000000	0.04468166	0.28009746	1.00000000	0.09430419	0.06286946	0.07476672
11001	0.09534626	0.00000000	0.06030227	0.00000000	0.00000000	0.00000000	0.04307305	0.04500225	0.09430419	1.00000000	0.04545455	0.07207500
15030	0.04767313	0.00000000	0.06030227	0.00000000	0.00000000	0.00000000	0.00000000	0.06000300	0.06286946	0.04545455	1.00000000	0.10811250
15034	0.07559289	0.08226127	0.09561829	0.00000000	0.00000000	0.00000000	0.05122408	0.07135782	0.07476672	0.07207500	0.10811250	1.00000000
15036	0.09081532	0.04941327	0.02871833	0.00000000	0.06421613	0.03589791	0.03076964	0.05715161	0.04491139	0.07576539	0.12988352	0.23169352
15039	0.06228411	0.03012376	0.02626129	0.00000000	0.05872202	0.00000000	0.05627419	0.05226191	0.05475857	0.01979519	0.15836152	0.23541181
15044A	0.04343722	0.00000000	0.00000000	0.00000000	0.06142951	0.06868028	0.00000000	0.06833943	0.07160414	0.02070788	0.04141577	0.13134182
15044B	0.07905694	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.03316791	0.03475240	0.02512595	0.05025189	0.11952286
15044C	0.07233642	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.06535653	0.06069670	0.04769723	0.02299002	0.04598005	0.12758946

```
> top10SimilarItemsTo23166
 [1] "23166" "23165" "23167" "22993" "23307" "22722" "22720" "22666" "23243" "22961"
[11] "23306"
```

	StockCode	Description
1	23166	MEDIUM CERAMIC TOP STORAGE JAR
2	23165	LARGE CERAMIC TOP STORAGE JAR
3	23167	SMALL CERAMIC TOP STORAGE JAR
4	22993	SET OF 4 PANTRY JELLY MOULDS
5	23307	SET OF 60 PANTRY DESIGN CAKE CASES
6	22722	SET OF 6 SPICE TINS PANTRY DESIGN
7	22720	SET OF 3 CAKE TINS PANTRY DESIGN
8	22666	RECIPE BOX PANTRY YELLOW DESIGN
9	23243	SET OF TEA COFFEE SUGAR TINS PANTRY
10	22961	JAM MAKING SET PRINTED
11	23306	SET OF 36 DOILIES PANTRY DESIGN

Chapter 7: Exploratory Analysis for Customer Behavior

```
df = pd.read_csv('../data/WA_Fn-UseC_-Marketing-Customer-Value-Analysis.csv')
```

```
df.shape
```

```
(9134, 24)
```

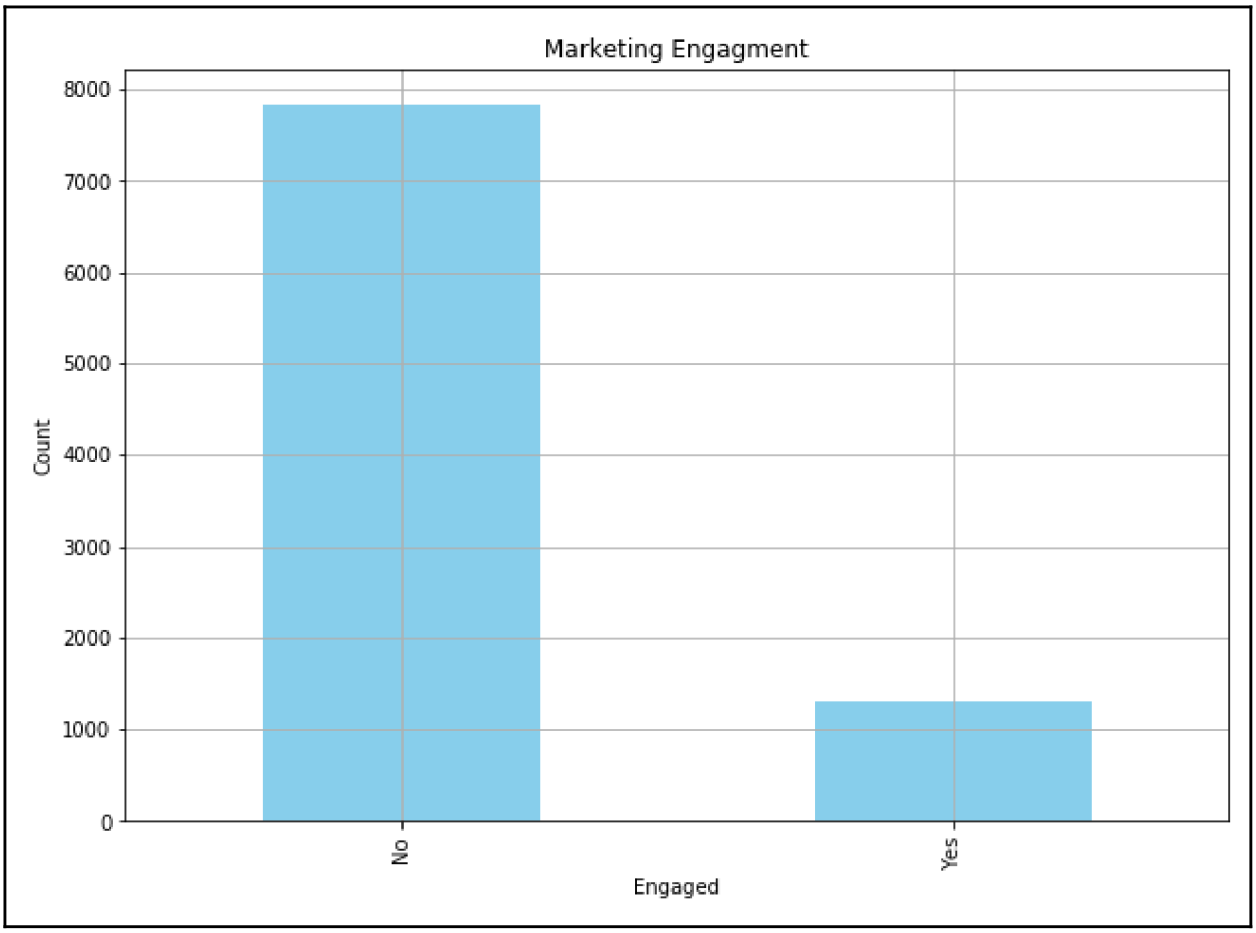
```
df.head()
```

	Customer	State	Customer Lifetime Value	Response	Coverage	Education	Effective To Date	EmploymentStatus	Gender	Income	...	Months Since Policy Inception	Number of Open Complaints	Number of Policies
0	BU79786	Washington	2763.519279	No	Basic	Bachelor	2/24/11	Employed	F	56274	...	5	0	1
1	QZ44356	Arizona	6979.535903	No	Extended	Bachelor	1/31/11	Unemployed	F	0	...	42	0	8
2	AI49188	Nevada	12887.431650	No	Premium	Bachelor	2/19/11	Employed	F	48767	...	38	0	2
3	WW63253	California	7645.861827	No	Basic	Bachelor	1/20/11	Unemployed	M	0	...	65	0	7
4	HB64268	Washington	2813.692575	No	Basic	Bachelor	2/3/11	Employed	M	43836	...	44	0	1

5 rows × 24 columns

```
df.groupby('Response').count()['Customer']
```

```
Response
No      7826
Yes     1308
Name: Customer, dtype: int64
```

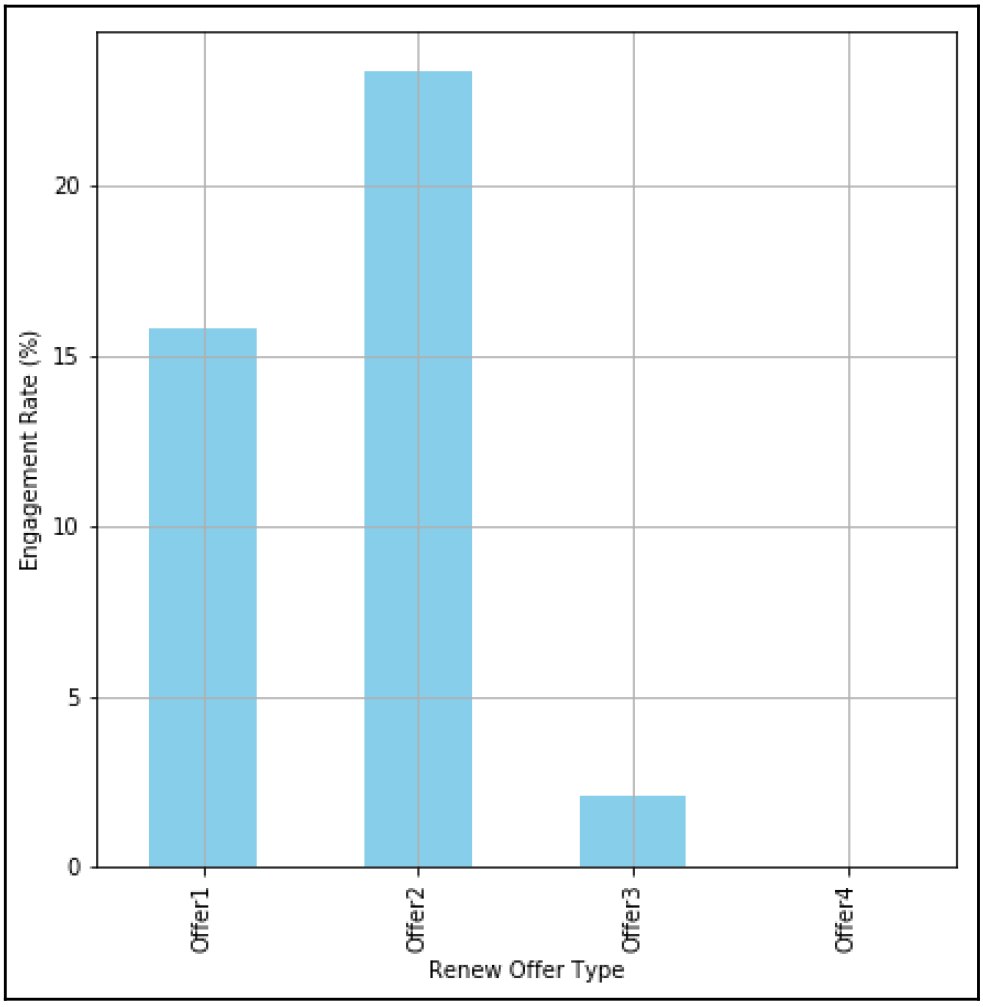



```
df.groupby('Response').count()['Customer']/df.shape[0]
```

```
Response
No      0.856799
Yes     0.143201
Name: Customer, dtype: float64
```

```
by_offer_type_df = df.loc[
    df['Response'] == 'Yes'
].groupby([
    'Renew Offer Type'
]).count()['Customer']/df.groupby('Renew Offer Type').count()['Customer']

by_offer_type_df
```

```
Renew Offer Type
Offer1    0.158316
Offer2    0.233766
Offer3    0.020950
Offer4         NaN
Name: Customer, dtype: float64
```

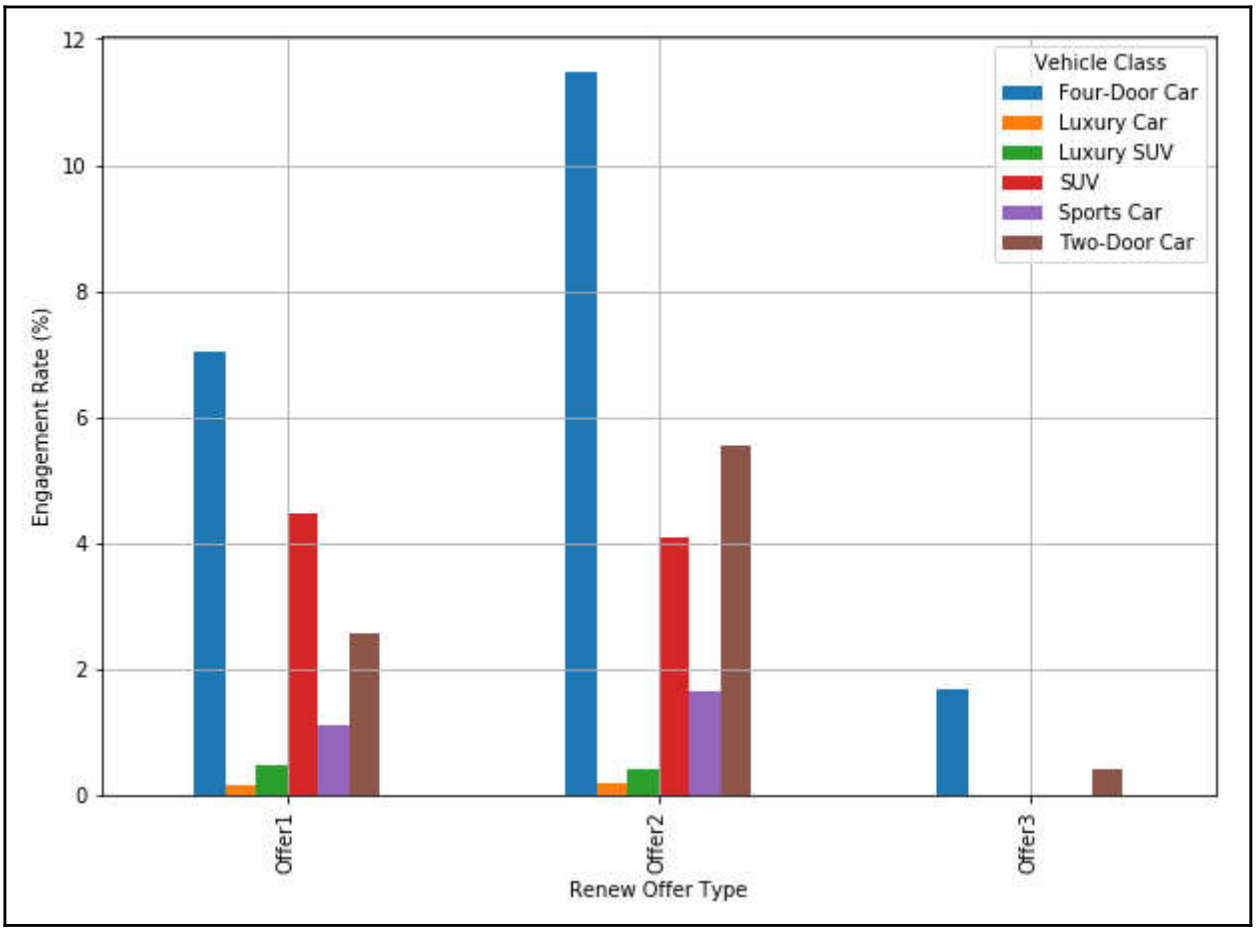


```
by_offer_type_df = df.loc[
    df['Response'] == 'Yes'
].groupby([
    'Renew Offer Type', 'Vehicle Class'
]).count()['Customer']/df.groupby('Renew Offer Type').count()['Customer']

by_offer_type_df
```

```
Renew Offer Type  Vehicle Class
Offer1            Four-Door Car    0.070362
                  Luxury Car       0.001599
                  Luxury SUV       0.004797
                  SUV              0.044776
                  Sports Car       0.011194
                  Two-Door Car     0.025586
Offer2            Four-Door Car    0.114833
                  Luxury Car       0.002051
                  Luxury SUV       0.004101
                  SUV              0.041012
                  Sports Car       0.016405
                  Two-Door Car     0.055366
Offer3            Four-Door Car    0.016760
                  Two-Door Car     0.004190
Name: Customer, dtype: float64
```

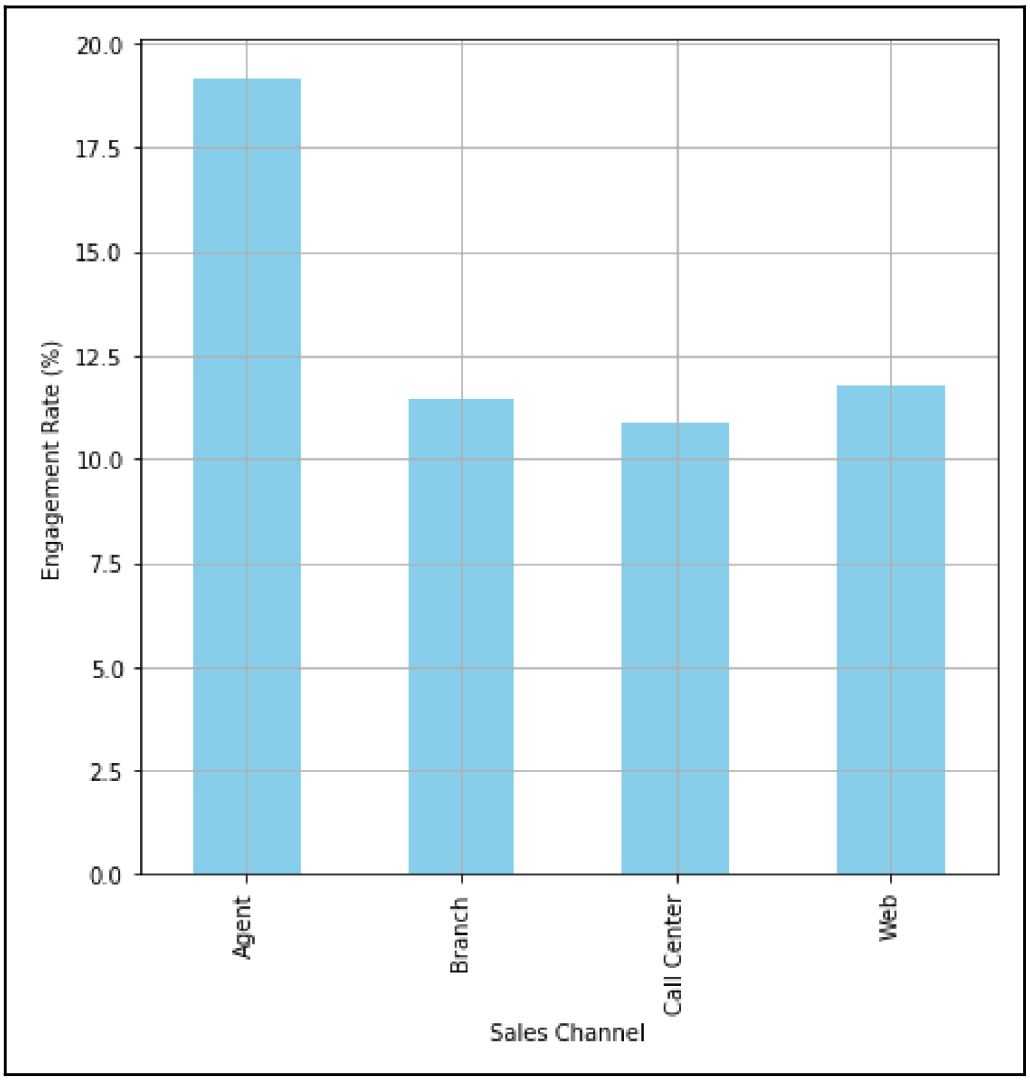
```
by_offer_type_df = by_offer_type_df.unstack().fillna(0)
by_offer_type_df
```

Vehicle Class	Four-Door Car	Luxury Car	Luxury SUV	SUV	Sports Car	Two-Door Car
Renew Offer Type						
Offer1	0.070362	0.001599	0.004797	0.044776	0.011194	0.025586
Offer2	0.114833	0.002051	0.004101	0.041012	0.016405	0.055366
Offer3	0.016760	0.000000	0.000000	0.000000	0.000000	0.004190




```
by_sales_channel_df = df.loc[
    df['Response'] == 'Yes'
].groupby([
    'Sales Channel'
]).count()['Customer']/df.groupby('Sales Channel').count()['Customer']

by_sales_channel_df
```

```
Sales Channel
Agent          0.191544
Branch         0.114531
Call Center    0.108782
Web            0.117736
Name: Customer, dtype: float64
```

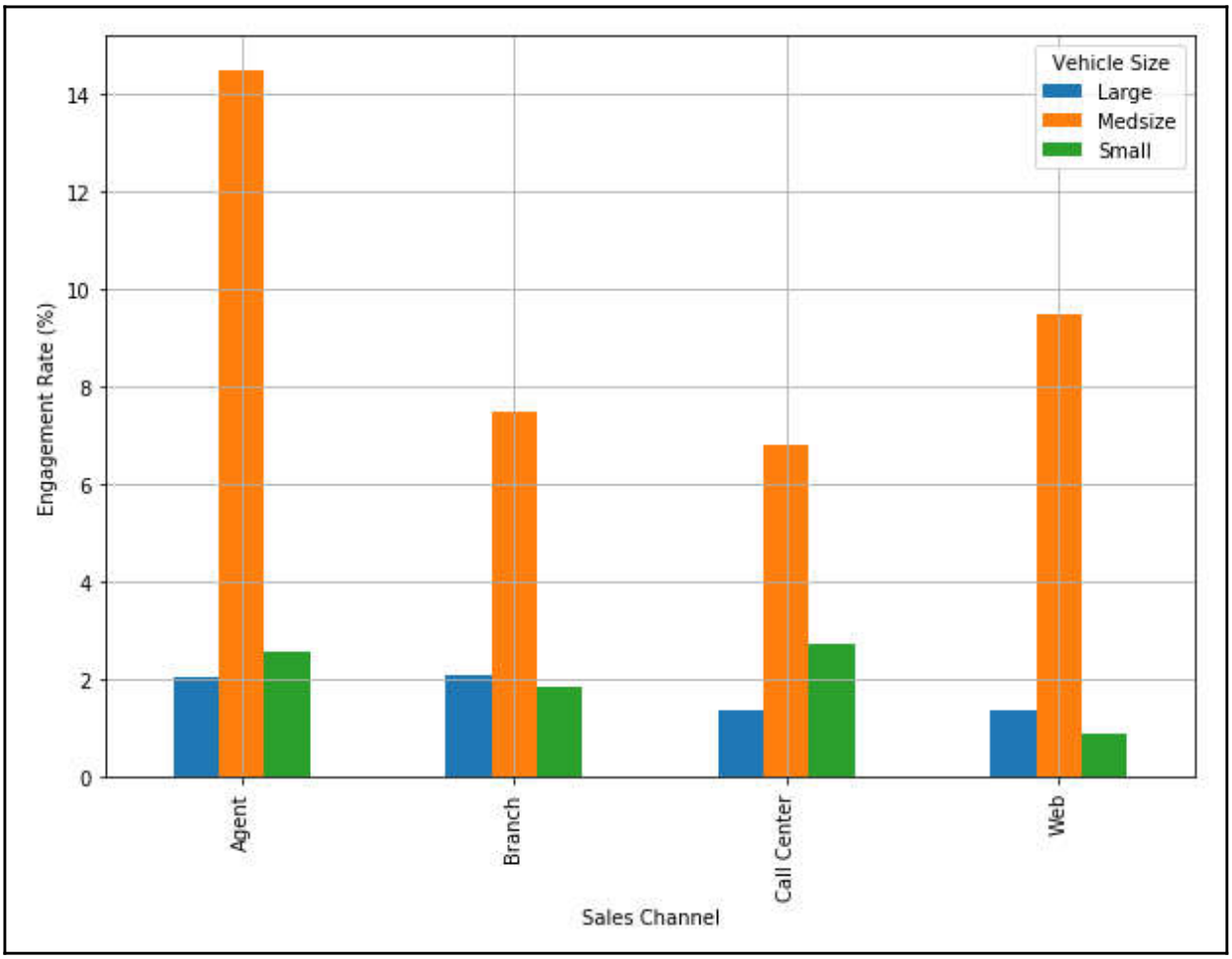

20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Engagement Rate (%)
Agent
Branch
Call Center
Web
Sales Channel

```
by_sales_channel_df = df.loc[
    df['Response'] == 'Yes'
].groupby([
    'Sales Channel', 'Vehicle Size'
]).count()['Customer']/df.groupby('Sales Channel').count()['Customer']

by_sales_channel_df
```

```
Sales Channel  Vehicle Size
Agent          Large           0.020708
               Medsize         0.144953
               Small           0.025884
Branch         Large           0.021036
               Medsize         0.074795
               Small           0.018699
Call Center    Large           0.013598
               Medsize         0.067989
               Small           0.027195
Web            Large           0.013585
               Medsize         0.095094
               Small           0.009057
Name: Customer, dtype: float64
```

```
by_sales_channel_df = by_sales_channel_df.unstack().fillna(0)
by_sales_channel_df
```

Vehicle Size	Large	Medsize	Small
Sales Channel			
Agent	0.020708	0.144953	0.025884
Branch	0.021036	0.074795	0.018699
Call Center	0.013598	0.067989	0.027195
Web	0.013585	0.095094	0.009057

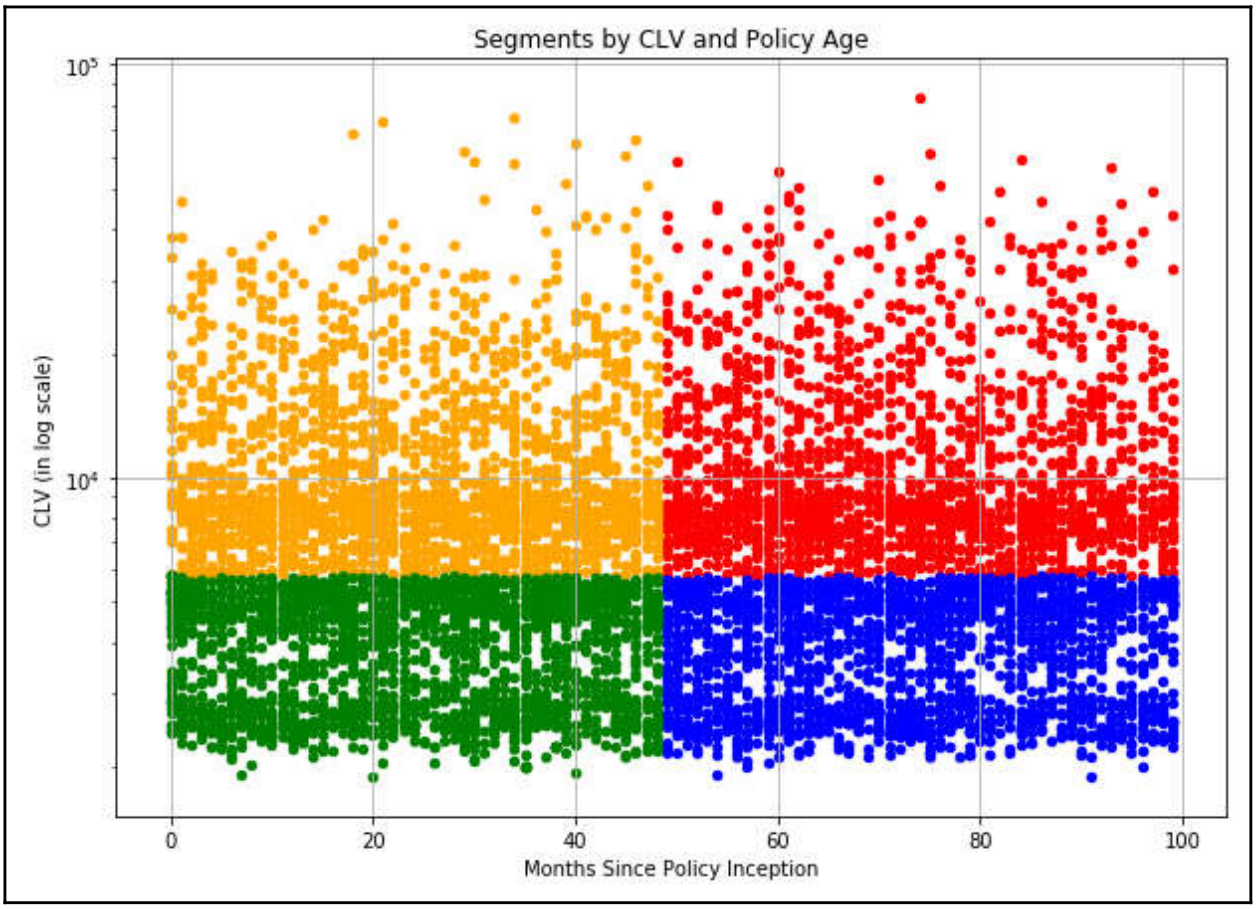

Vehicle Size
Large
Medsize
Small
Engagement Rate (%)
0
2
4
6
8
10
12
14
Agent
Branch
Call Center
Web
Sales Channel

```
df['Customer Lifetime Value'].describe()
```

```
count     9134.000000
mean      8004.940475
std       6870.967608
min       1898.007675
25%       3994.251794
50%       5780.182197
75%       8962.167041
max      83325.381190
Name: Customer Lifetime Value, dtype: float64
```

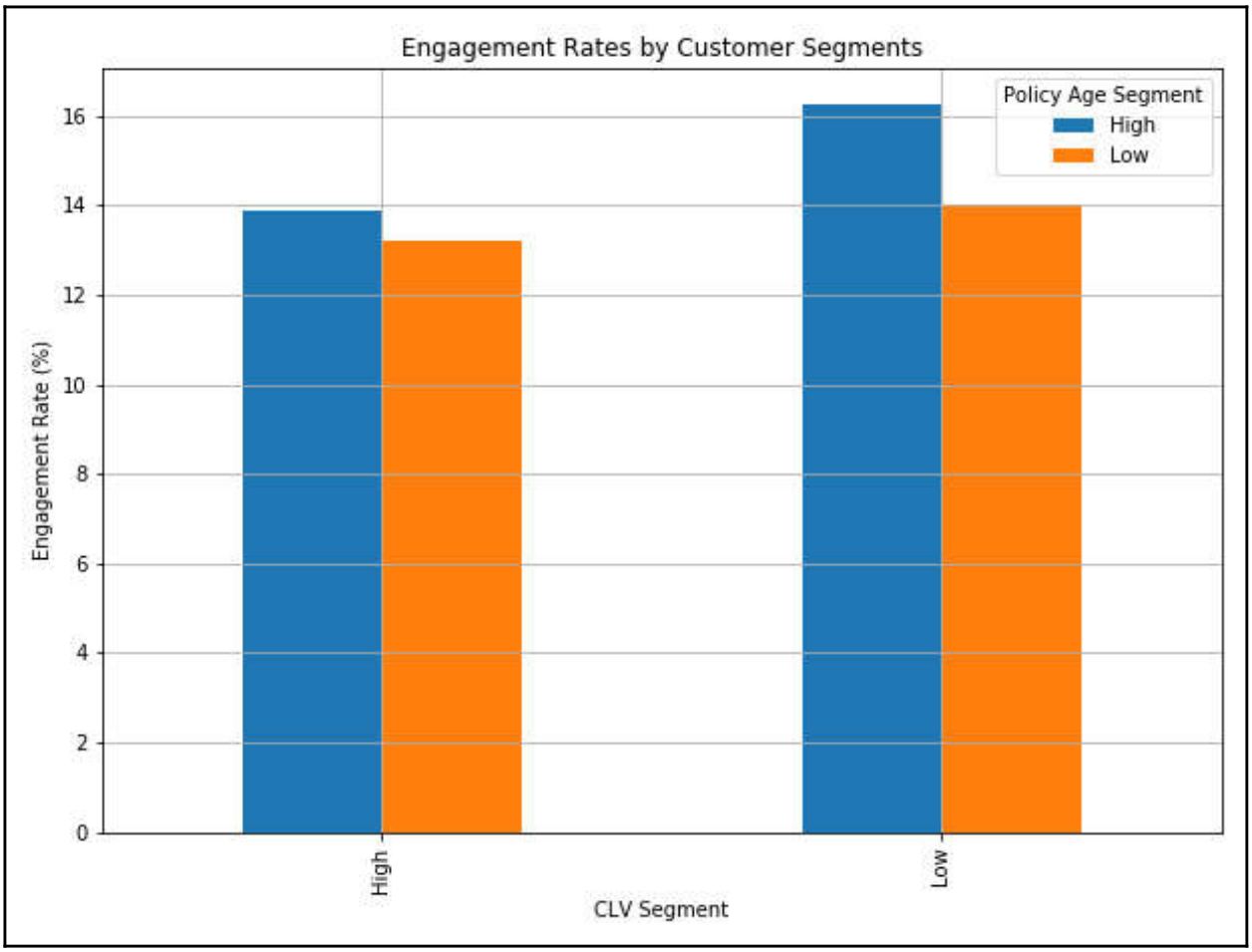
```
df['Months Since Policy Inception'].describe()
```

```
count    9134.000000
mean       48.064594
std        27.905991
min         0.000000
25%        24.000000
50%        48.000000
75%        71.000000
max        99.000000
Name: Months Since Policy Inception, dtype: float64
```


Segments by CLV and Policy Age
10^5
10^4
CLV (in log scale)
0
20
40
60
80
100
Months Since Policy Inception

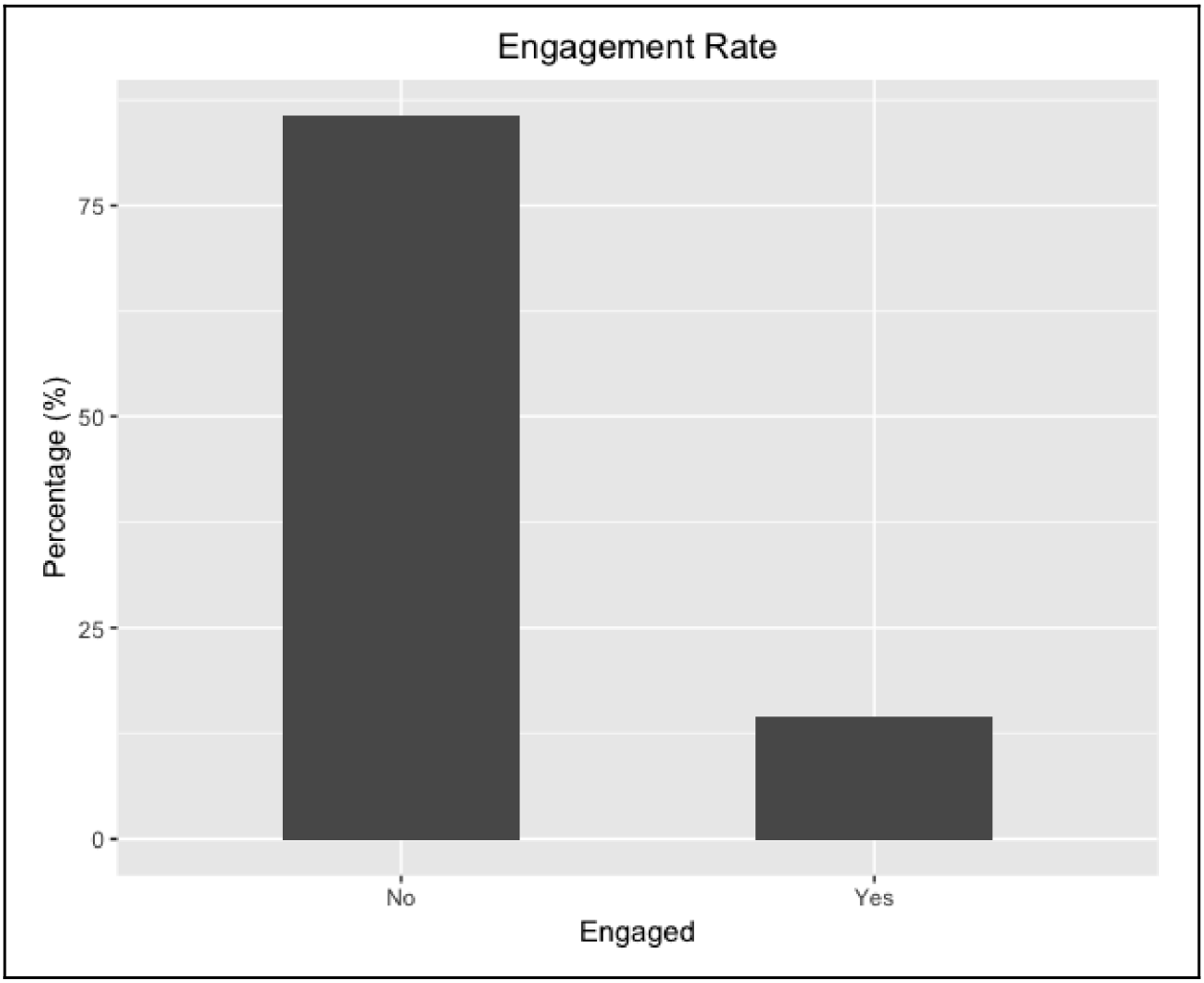
```
engagment_rates_by_segment_df = df.loc[
    df['Response'] == 'Yes'
].groupby(
    ['CLV Segment', 'Policy Age Segment']
).count()['Customer']/df.groupby(
    ['CLV Segment', 'Policy Age Segment']
).count()['Customer']

engagment_rates_by_segment_df
```

```
CLV Segment  Policy Age Segment
High         High                  0.138728
             Low                   0.132067
Low          High                  0.162450
             Low                   0.139957
Name: Customer, dtype: float64
```

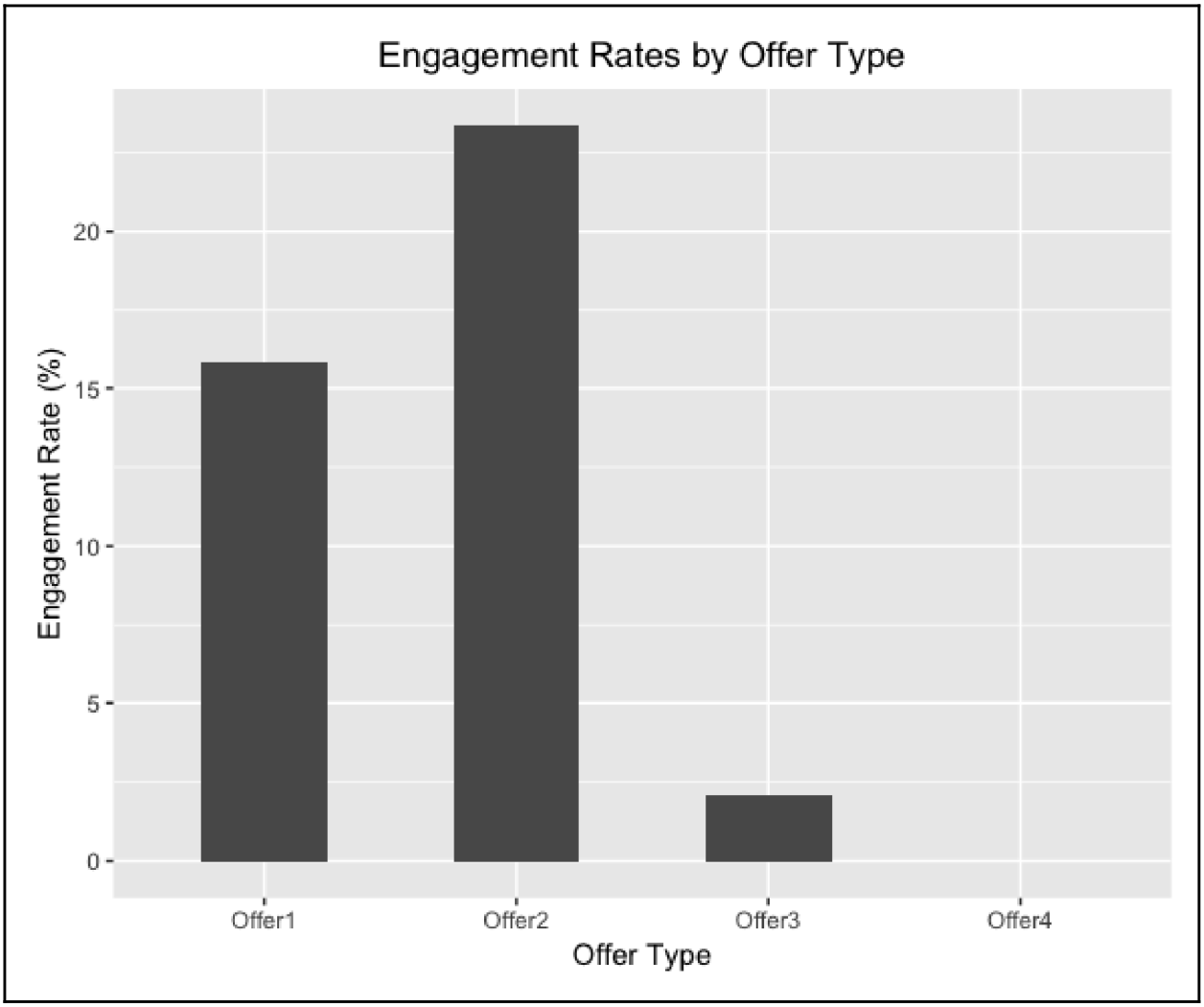


	Customer	State	Customer.Lifetime.Value	Response	Coverage	Education	Effective.To.Date	EmploymentStatus	Gender	Income
1	BU79786	Washington	2763.519	No	Basic	Bachelor	2/24/11	Employed	F	56274
2	QZ44356	Arizona	6979.536	No	Extended	Bachelor	1/31/11	Unemployed	F	0
3	AI49188	Nevada	12887.432	No	Premium	Bachelor	2/19/11	Employed	F	48767
4	WW63253	California	7645.862	No	Basic	Bachelor	1/20/11	Unemployed	M	0
5	HB64268	Washington	2813.693	No	Basic	Bachelor	2/3/11	Employed	M	43836
6	OC83172	Oregon	8256.298	Yes	Basic	Bachelor	1/25/11	Employed	F	62902
7	XZ87318	Oregon	5380.899	Yes	Basic	College	2/24/11	Employed	F	55350
8	CF85061	Arizona	7216.100	No	Premium	Master	1/18/11	Unemployed	M	0
9	DY87989	Oregon	24127.504	Yes	Basic	Bachelor	1/26/11	Medical Leave	M	14072
10	BQ94931	Oregon	7388.178	No	Extended	College	2/17/11	Employed	F	28812
11	SX51350	California	4738.992	No	Basic	College	2/21/11	Unemployed	M	0
12	VQ65197	California	8197.197	No	Basic	College	1/6/11	Unemployed	F	0
13	DP39365	California	8798.797	No	Premium	Master	2/6/11	Employed	M	77026
14	SJ95423	Arizona	8819.019	Yes	Basic	High School or Below	1/10/11	Employed	M	99845
15	IL66569	California	5384.432	No	Basic	College	1/18/11	Employed	M	83689

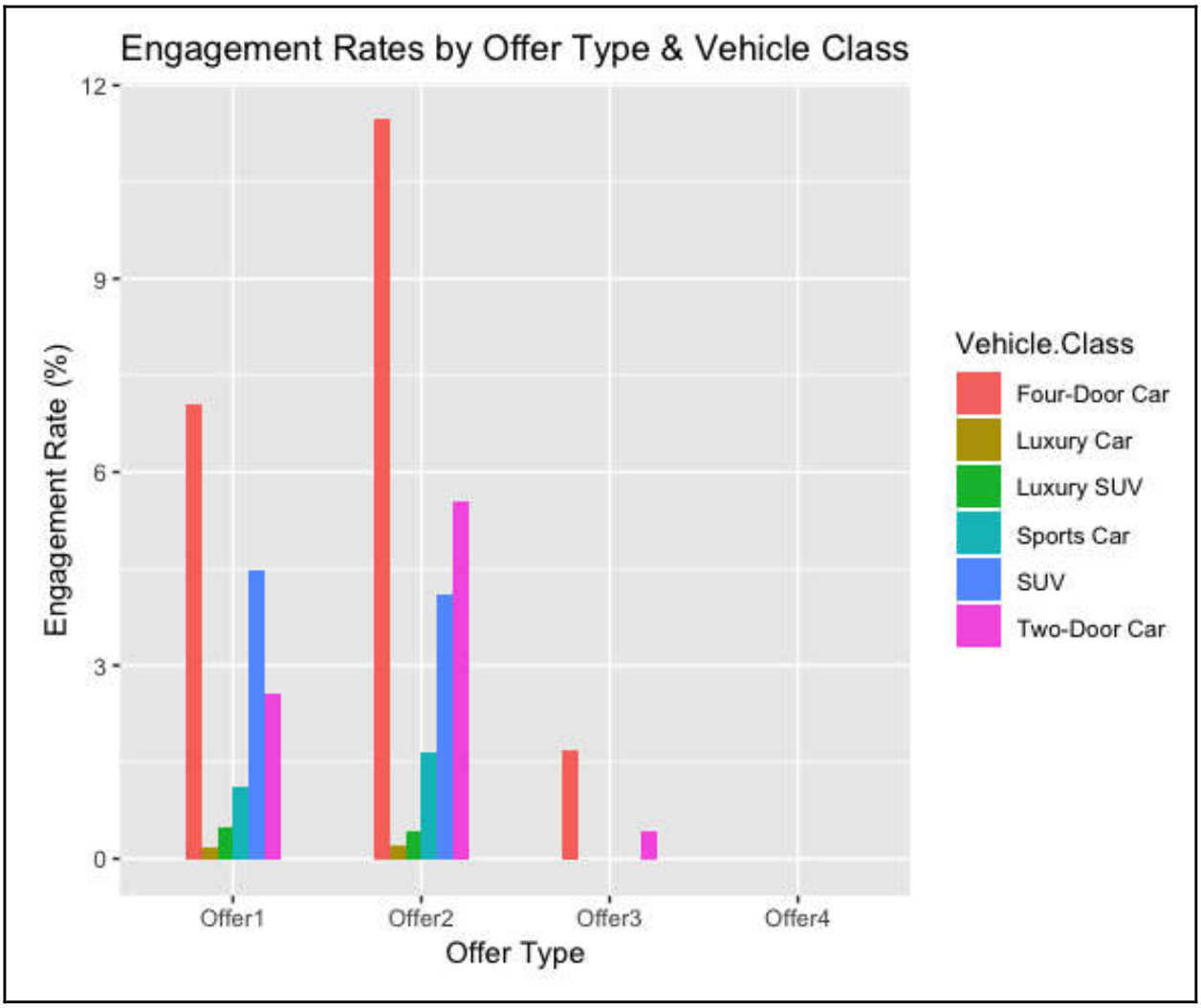
	Response	Count	EngagementRate
1	No	7826	85.67988
2	Yes	1308	14.32012



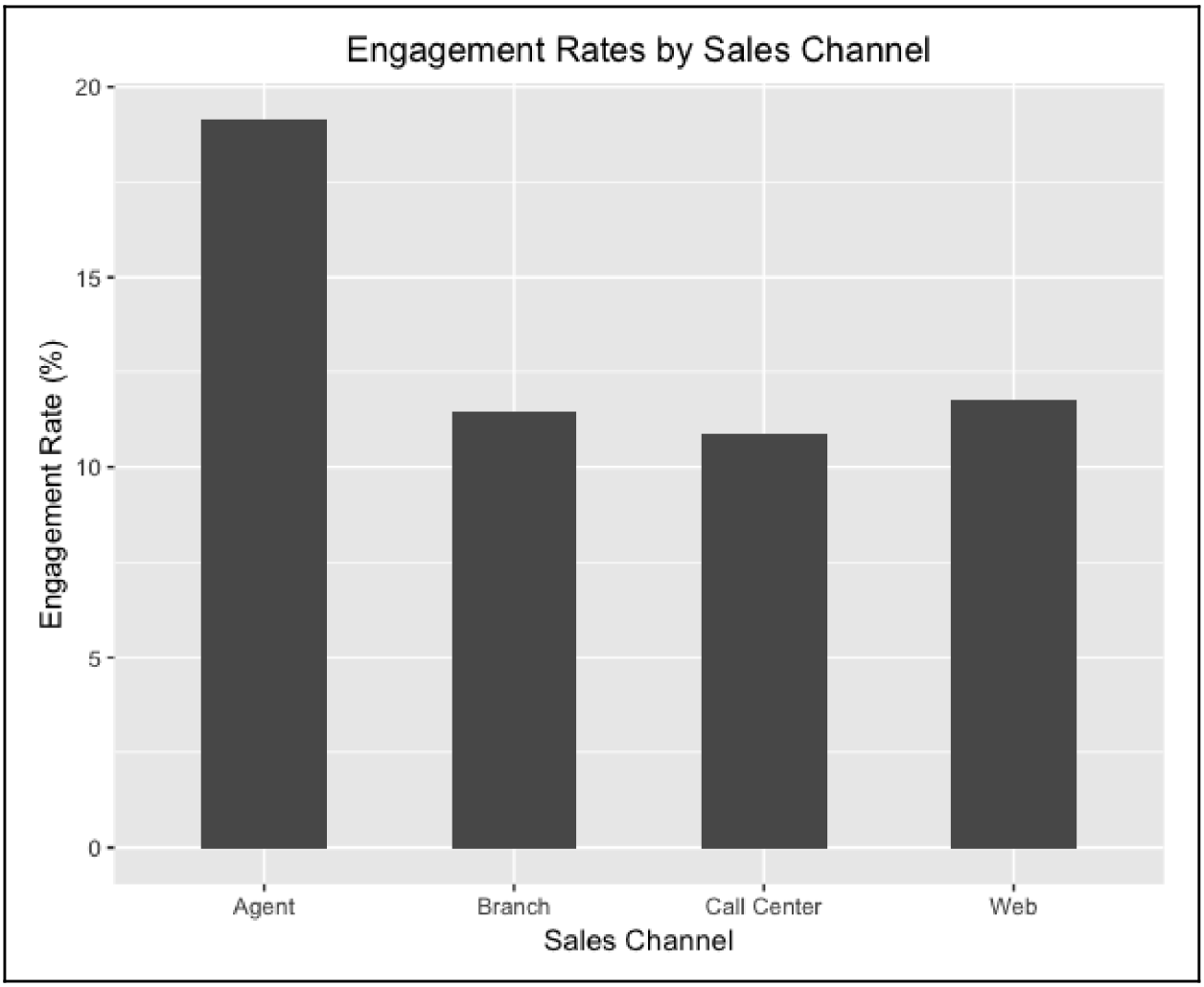
	Renew.Offer.Type	Count	NumEngaged	EngagementRate
1	Offer1	3752	594	15.831557
2	Offer2	2926	684	23.376623
3	Offer3	1432	30	2.094972
4	Offer4	1024	0	0.000000



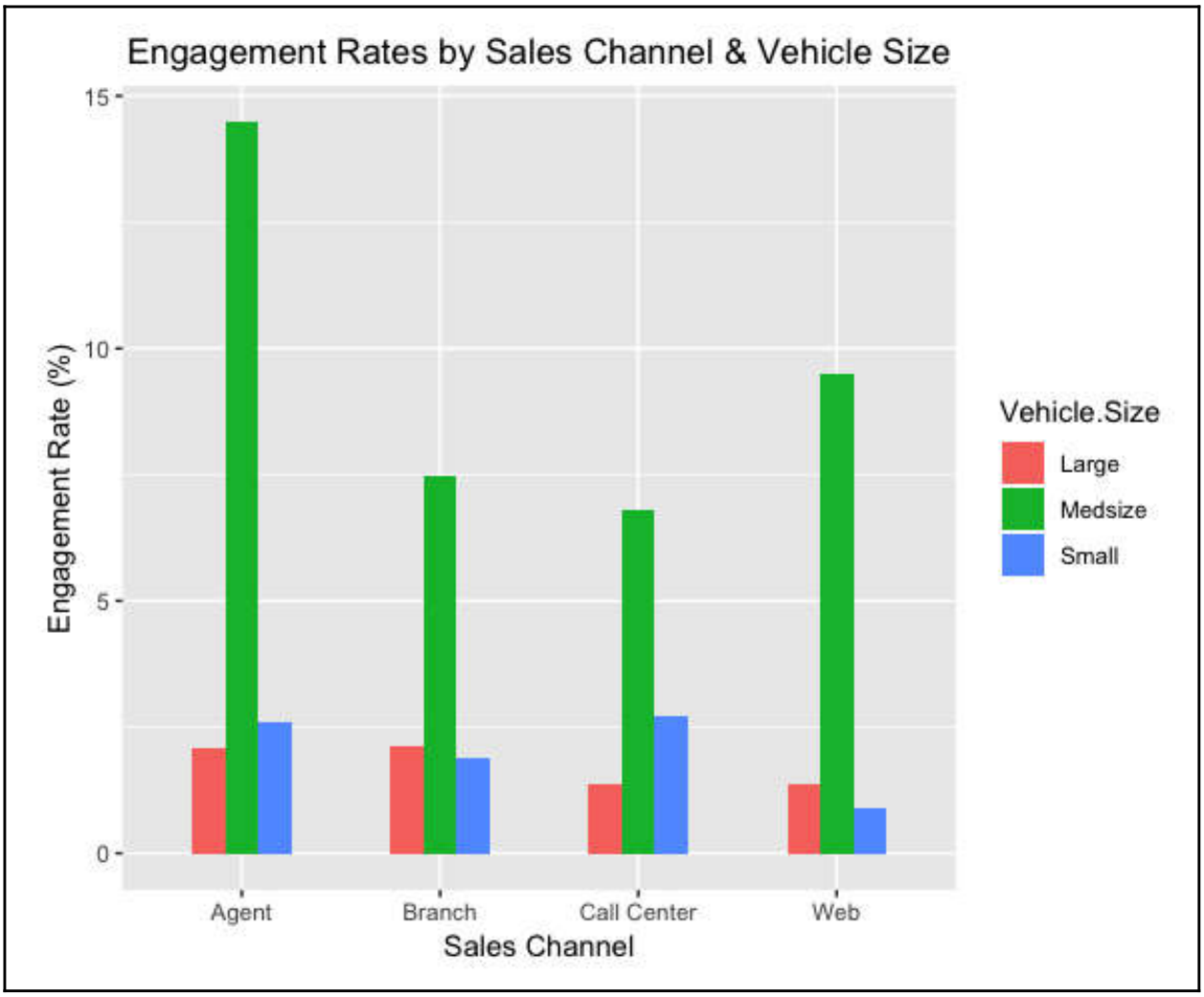
	Renew.Offer.Type	Vehicle.Class	NumEngaged	Count	EngagementRate
1	Offer1	Four-Door Car	264	3752	7.0362473
2	Offer1	Luxury Car	6	3752	0.1599147
3	Offer1	Luxury SUV	18	3752	0.4797441
4	Offer1	Sports Car	42	3752	1.1194030
5	Offer1	SUV	168	3752	4.4776119
6	Offer1	Two-Door Car	96	3752	2.5586354
7	Offer2	Four-Door Car	336	2926	11.4832536
8	Offer2	Luxury Car	6	2926	0.2050581
9	Offer2	Luxury SUV	12	2926	0.4101162
10	Offer2	Sports Car	48	2926	1.6404648
11	Offer2	SUV	120	2926	4.1011620
12	Offer2	Two-Door Car	162	2926	5.5365687
13	Offer3	Four-Door Car	24	1432	1.6759777
14	Offer3	Luxury Car	0	1432	0.0000000
15	Offer3	Luxury SUV	0	1432	0.0000000
16	Offer3	Sports Car	0	1432	0.0000000
17	Offer3	SUV	0	1432	0.0000000
18	Offer3	Two-Door Car	6	1432	0.4189944
19	Offer4	Four-Door Car	0	1024	0.0000000
20	Offer4	Luxury Car	0	1024	0.0000000
21	Offer4	Luxury SUV	0	1024	0.0000000
22	Offer4	Sports Car	0	1024	0.0000000
23	Offer4	SUV	0	1024	0.0000000
24	Offer4	Two-Door Car	0	1024	0.0000000



	Sales.Channel	Count	NumEngaged	EngagementRate
1	Agent	3477	666	19.15444
2	Branch	2567	294	11.45306
3	Call Center	1765	192	10.87819
4	Web	1325	156	11.77358

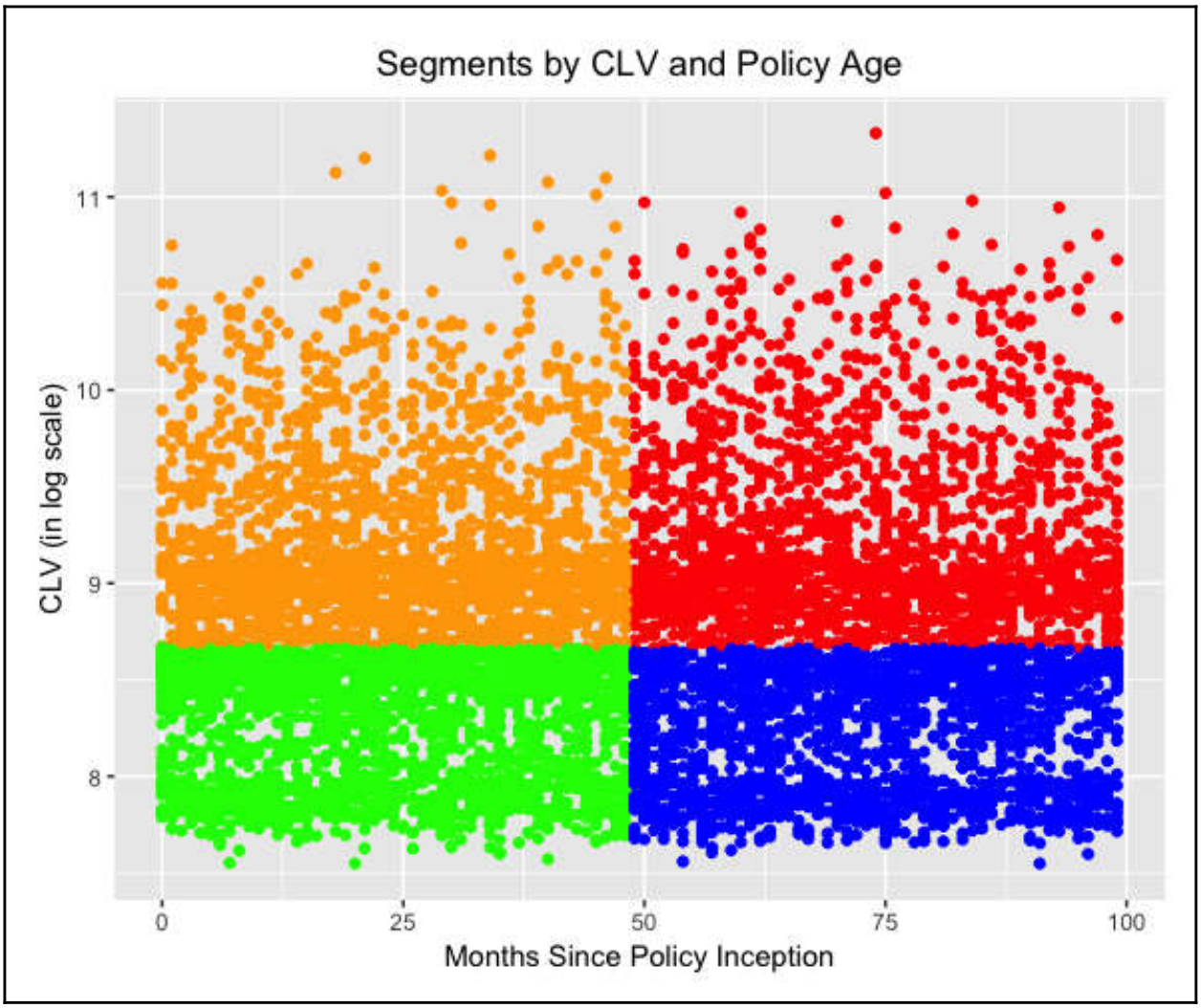

Engagement Rates by Sales Channel
Engagement Rate (%)
20
15
10
5
0
Agent
Branch
Call Center
Web
Sales Channel

	Sales.Channel	Vehicle.Size	NumEngaged	Count	EngagementRate
1	Agent	Large	72	3477	2.0707506
2	Agent	Medsize	504	3477	14.4952545
3	Agent	Small	90	3477	2.5884383
4	Branch	Large	54	2567	2.1036229
5	Branch	Medsize	192	2567	7.4795481
6	Branch	Small	48	2567	1.8698870
7	Call Center	Large	24	1765	1.3597734
8	Call Center	Medsize	120	1765	6.7988669
9	Call Center	Small	48	1765	2.7195467
10	Web	Large	18	1325	1.3584906
11	Web	Medsize	126	1325	9.5094340
12	Web	Small	12	1325	0.9056604

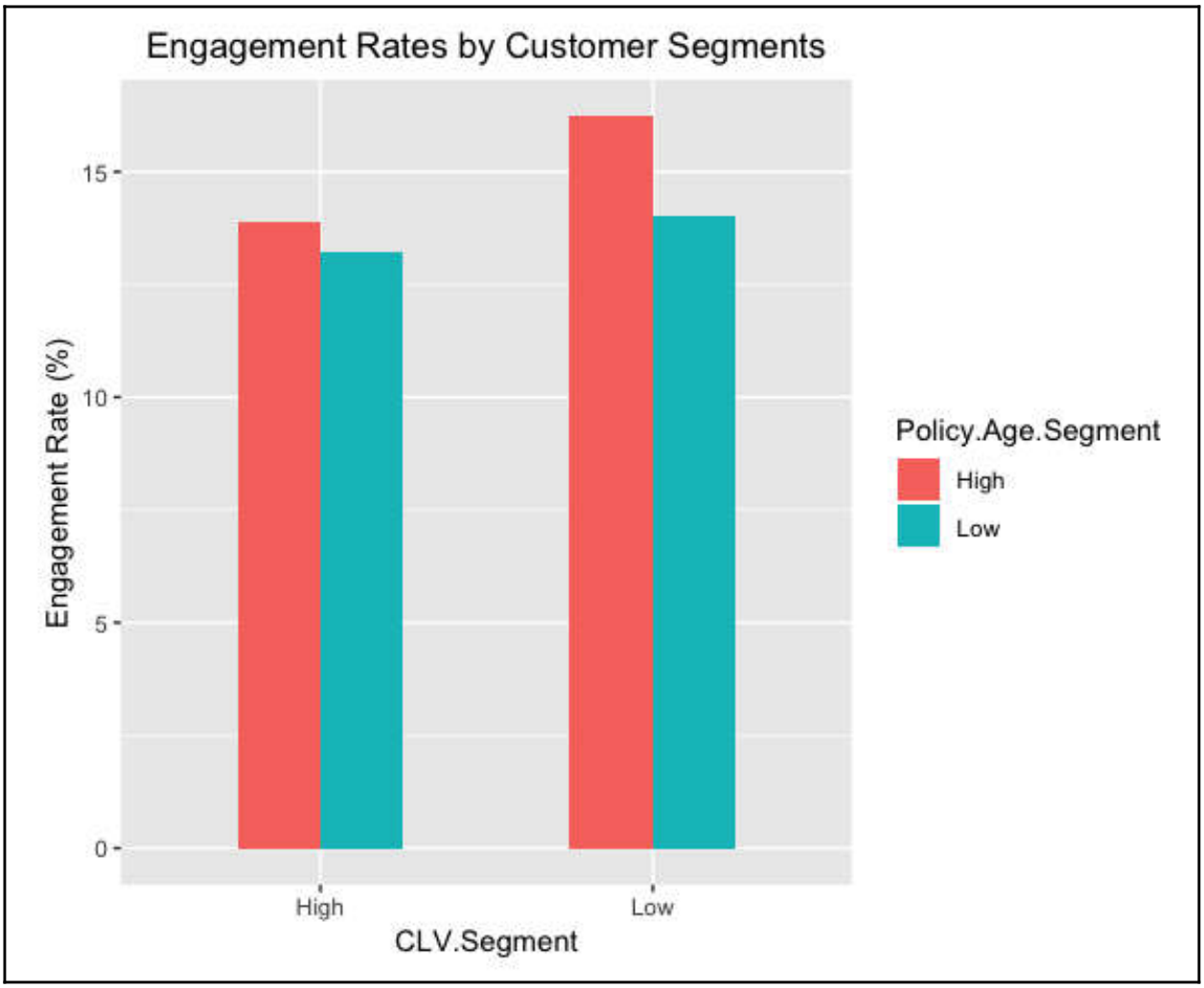



```
> summary(df$Customer.Lifetime.Value)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1898    3994    5780    8005    8962   83325
```

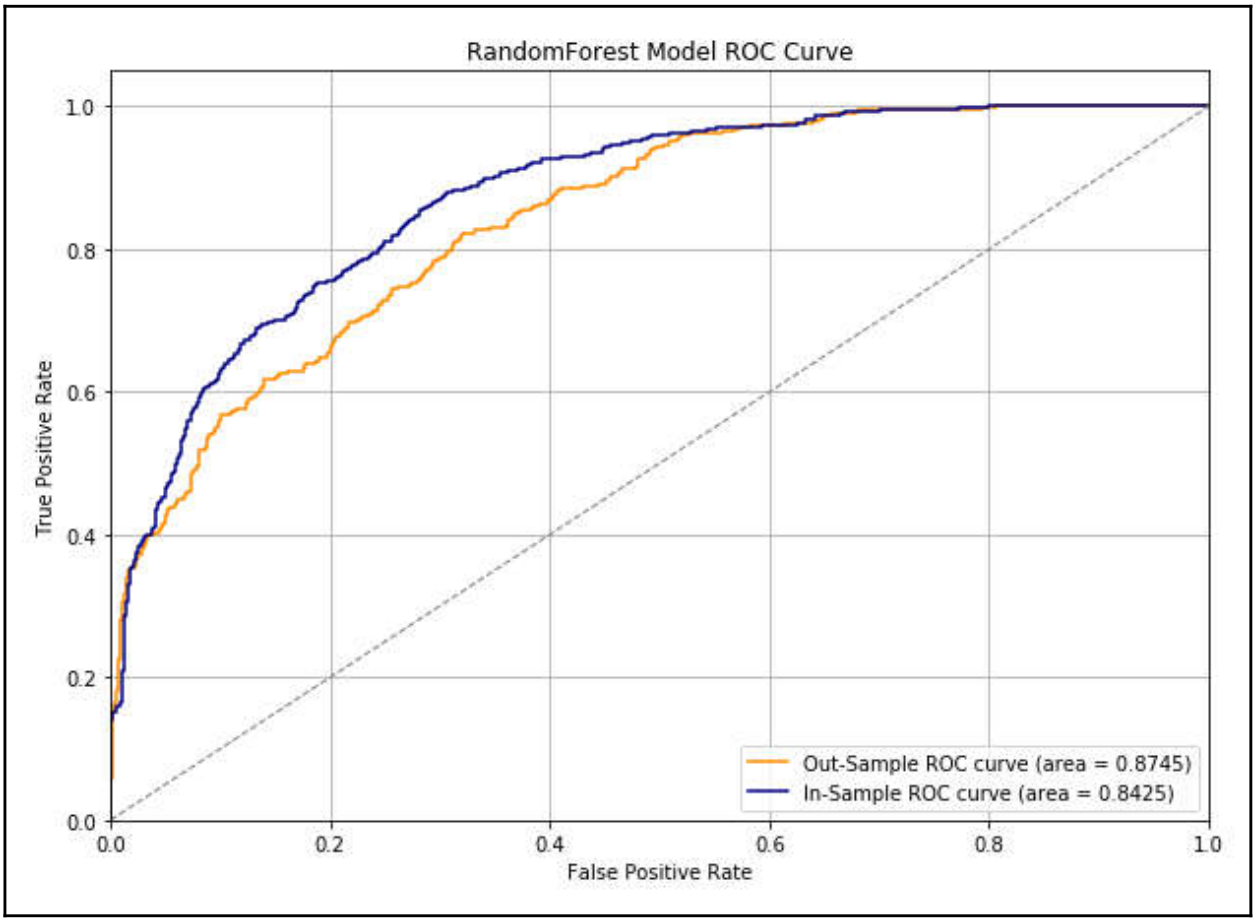
```
> summary(df$Months.Since.Policy.Inception)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   24.00   48.00   48.06   71.00   99.00
```



	CLV.Segment	Policy.Age.Segment	Count	NumEngaged	EngagementRate
1	High	High	2249	312	13.87283
2	High	Low	2317	306	13.20673
3	Low	High	2253	366	16.24501
4	Low	Low	2315	324	13.99568


Engagement Rates by Customer Segments
Engagement Rate (%)
15
10
5
0
High
Low
CLV.Segment
Policy.Age.Segment
High
Low

Chapter 8: Predicting the Likelihood of Marketing Engagement



```
df.head()
```

	Customer	State	Customer Lifetime Value	Response	Coverage	Education	Effective To Date	EmploymentStatus	Gender	Income	...	Months Since Policy Inception	Number of Open Complaints	Number of Policies
0	BU79786	Washington	2763.519279	No	Basic	Bachelor	2/24/11	Employed	F	56274	...	5	0	1
1	QZ44356	Arizona	6979.535903	No	Extended	Bachelor	1/31/11	Unemployed	F	0	...	42	0	8
2	AI49188	Nevada	12887.431650	No	Premium	Bachelor	2/19/11	Employed	F	48767	...	38	0	2
3	WW63253	California	7645.861827	No	Basic	Bachelor	1/20/11	Unemployed	M	0	...	65	0	7
4	HB64268	Washington	2813.692575	No	Basic	Bachelor	2/3/11	Employed	M	43836	...	44	0	1

```
df['Engaged'].mean()
```

```
0.14320122618786948
```

	Sales.Channel.Agent	Sales.Channel.Branch	Sales.Channel.Call Center	Sales.Channel.Web
0	1	0	0	0
1	1	0	0	0
2	1	0	0	0
3	0	0	1	0
4	1	0	0	0
5	0	0	0	1
6	1	0	0	0
7	1	0	0	0
8	1	0	0	0
9	0	1	0	0

```
sample_df.head()
```

	Customer.Lifetime.Value	Income	Monthly.Premium.Auto	Months.Since.Last.Claim	Months.Since.Policy.Inception	Number.of.Open.Complaints	Number.of.Policie
0	2763.519279	56274	69	32	5	0	
1	6979.535903	0	94	13	42	0	
2	12887.431650	48767	108	18	38	0	
3	7645.861827	0	106	18	65	0	
4	2813.692575	43836	73	12	44	0	

5 rows × 51 columns

```
sample_df.shape
```

```
(9134, 51)
```

```
x_train.shape
```

```
(6393, 50)
```

```
x_test.shape
```

```
(2741, 50)
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf_model = RandomForestClassifier(
    n_estimators=200,
    max_depth=5
)
```

```
rf_model.fit(X=x_train, y=y_train)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=5, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=200, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
```

- Individual Trees

```
rf_model.estimators_
```

```
[DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=5,
             max_features='auto', max_leaf_nodes=None,
             min_impurity_decrease=0.0, min_impurity_split=None,
             min_samples_leaf=1, min_samples_split=2,
             min_weight_fraction_leaf=0.0, presort=False,
             random_state=1182049216, splitter='best'),
 DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=5,
             max_features='auto', max_leaf_nodes=None,
             min_impurity_decrease=0.0, min_impurity_split=None,
             min_samples_leaf=1, min_samples_split=2,
             min_weight_fraction_leaf=0.0, presort=False,
             random_state=829317093, splitter='best'),
 DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=5,
             max_features='auto', max_leaf_nodes=None,
             min_impurity_decrease=0.0, min_impurity_split=None,
             min_samples_leaf=1, min_samples_split=2,
             min_weight_fraction_leaf=0.0, presort=False,
             random_state=1398037487, splitter='best'),
 DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=5,
             max_features='auto', max_leaf_nodes=None,
             min_impurity_decrease=0.0, min_impurity_split=None,
             min_samples_leaf=1, min_samples_split=2,
             min_weight_fraction_leaf=0.0, presort=False,
             random_state=831979291, splitter='best'),
```

```
rf_model.estimators_[0].predict(x_test)[:10]
```

```
array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.])
```

```
rf_model.estimators_[1].predict(x_test)[:10]
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
rf_model.estimators_[2].predict(x_test)[:10]
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

```
rf_model.estimators_[3].predict(x_test)[:10]
```

```
array([0., 0., 1., 0., 0., 0., 0., 0., 0., 0.])
```

```
rf_model.estimators_[4].predict(x_test)[:10]
```

```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.])
```

- Feature Importances

```
rf_model.feature_importances_
```

```
array([0.06054531, 0.09003091, 0.05340668, 0.02954197, 0.05362482,
       0.01076425, 0.02145626, 0.07377831, 0.04641028, 0.00755703,
       0.00755832, 0.00568473, 0.0067975 , 0.01018243, 0.0102477 ,
       0.00495166, 0.00113173, 0.00146122, 0.00683007, 0.00633034,
       0.00360746, 0.00122614, 0.00120962, 0.00099013, 0.00150983,
       0.00156154, 0.00167344, 0.00093062, 0.00100503, 0.00273003,
       0.00141377, 0.00098851, 0.00111641, 0.00475984, 0.03401539,
       0.00604427, 0.29381144, 0.02655533, 0.03369894, 0.01470135,
       0.01590602, 0.00454194, 0.00414489, 0.00268673, 0.0043107 ,
       0.00546048, 0.00556966, 0.00578405, 0.00193898, 0.00781593])
```

```
feature_importance_df = pd.DataFrame(list(zip(rf_model.feature_importances_, all_features)))
feature_importance_df.columns = ['feature.importance', 'feature']

feature_importance_df.sort_values(by='feature.importance', ascending=False)
```

	feature.importance	feature
36	0.293811	EmploymentStatus.Retired
1	0.090031	Income
7	0.073778	Total.Claim.Amount
0	0.060545	Customer.Lifetime.Value
4	0.053625	Months.Since.Policy.Inception
2	0.053407	Monthly.Premium.Auto
8	0.046410	Sales.Channel.Agent
34	0.034015	EmploymentStatus.Employed
38	0.033699	Marital.Status.Divorced
3	0.029542	Months.Since.Last.Claim
37	0.026555	EmploymentStatus.Unemployed
6	0.021456	Number.of.Policies
40	0.015906	Marital.Status.Single
39	0.014701	Marital.Status.Married
5	0.010764	Number.of.Open.Complaints
14	0.010248	Vehicle.Size.Small

- Accuracy, Precision, and Recall

```
from sklearn.metrics import accuracy_score, precision_score, recall_score
```

```
in_sample_preds = rf_model.predict(x_train)
out_sample_preds = rf_model.predict(x_test)
```

```
print('In-Sample Accuracy: %0.4f' % accuracy_score(y_train, in_sample_preds))
print('Out-of-Sample Accuracy: %0.4f' % accuracy_score(y_test, out_sample_preds))
```

```
In-Sample Accuracy: 0.8724
Out-of-Sample Accuracy: 0.8818
```

```
print('In-Sample Precision: %0.4f' % precision_score(y_train, in_sample_preds))
print('Out-of-Sample Precision: %0.4f' % precision_score(y_test, out_sample_preds))
```

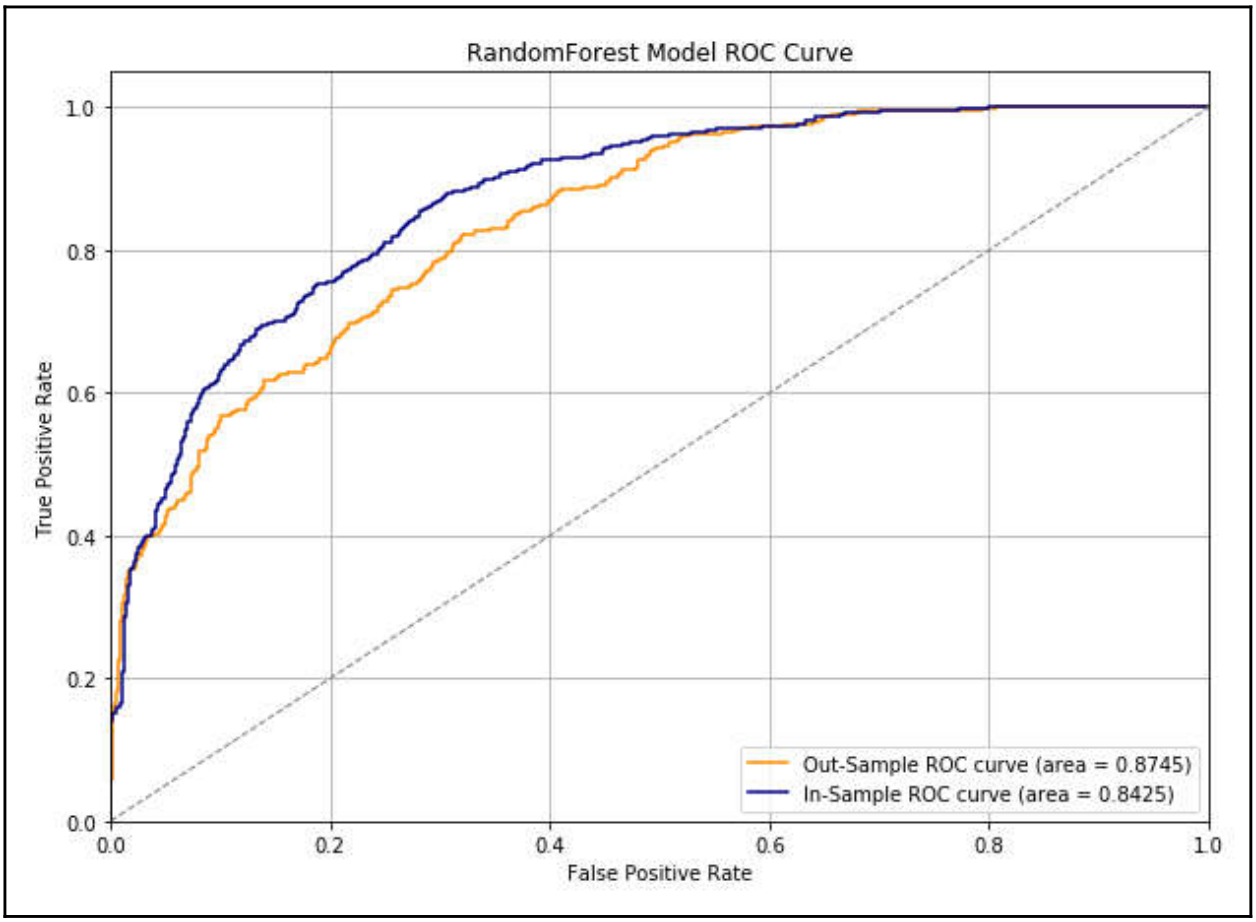
```
In-Sample Precision: 0.9919
Out-of-Sample Precision: 0.9423
```

```
print('In-Sample Recall: %0.4f' % recall_score(y_train, in_sample_preds))
print('Out-of-Sample Recall: %0.4f' % recall_score(y_test, out_sample_preds))
```

```
In-Sample Recall: 0.1311
Out-of-Sample Recall: 0.1324
```

```
in_sample_roc_auc = auc(in_sample_fpr, in_sample_tpr)
out_sample_roc_auc = auc(out_sample_fpr, out_sample_tpr)

print('In-Sample AUC: %0.4f' % in_sample_roc_auc)
print('Out-Sample AUC: %0.4f' % out_sample_roc_auc)
```

```
In-Sample AUC: 0.8745
Out-Sample AUC: 0.8425
```



	Customer	State	Customer.Lifetime.Value	Response	Coverage	Education	Effective.To.Date	EmploymentStatus	Gender	Income
1	BU79786	Washington	2763.519	No	Basic	Bachelor	2/24/11	Employed	F	56274
2	QZ44356	Arizona	6979.536	No	Extended	Bachelor	1/31/11	Unemployed	F	0
3	AI49188	Nevada	12887.432	No	Premium	Bachelor	2/19/11	Employed	F	48767
4	WW63253	California	7645.862	No	Basic	Bachelor	1/20/11	Unemployed	M	0
5	HB64268	Washington	2813.693	No	Basic	Bachelor	2/3/11	Employed	M	43836
6	OC83172	Oregon	8256.298	Yes	Basic	Bachelor	1/25/11	Employed	F	62902
7	XZ87318	Oregon	5380.899	Yes	Basic	College	2/24/11	Employed	F	55350
8	CF85061	Arizona	7216.100	No	Premium	Master	1/18/11	Unemployed	M	0
9	DY87989	Oregon	24127.504	Yes	Basic	Bachelor	1/26/11	Medical Leave	M	14072
10	BQ94931	Oregon	7388.178	No	Extended	College	2/17/11	Employed	F	28812
11	SX51350	California	4738.992	No	Basic	College	2/21/11	Unemployed	M	0
12	VQ65197	California	8197.197	No	Basic	College	1/6/11	Unemployed	F	0
13	DP39365	California	8798.797	No	Premium	Master	2/6/11	Employed	M	77026
14	SJ95423	Arizona	8819.019	Yes	Basic	High School or Below	1/10/11	Employed	M	99845
15	IL66569	California	5384.432	No	Basic	College	1/18/11	Employed	M	83689

```
> mean(df$Engaged)
[1] 0.1432012
```

	Sales.ChannelAgent	Sales.ChannelBranch	Sales.ChannelCall Center	Sales.ChannelWeb	Vehicle.SizeMedsize	Vehicle.SizeSmall	Vehicle.ClassLuxury Car
1	1	0	0	0	1	0	0
2	1	0	0	0	1	0	0
3	1	0	0	0	1	0	0
4	0	0	1	0	1	0	0
5	1	0	0	0	1	0	0
6	0	0	0	1	1	0	0
7	1	0	0	0	1	0	0
8	1	0	0	0	1	0	0
9	1	0	0	0	1	0	0
10	0	1	0	0	1	0	0
11	1	0	0	0	0	1	0
12	1	0	0	0	1	0	0
13	1	0	0	0	1	0	0
14	0	1	0	0	1	0	0
15	0	0	1	0	1	0	0

```
> dim(encodedDF)
[1] 9134   42
>
> dim(trainX)
[1] 6393   42
>
> dim(testX)
[1] 2741   42
```

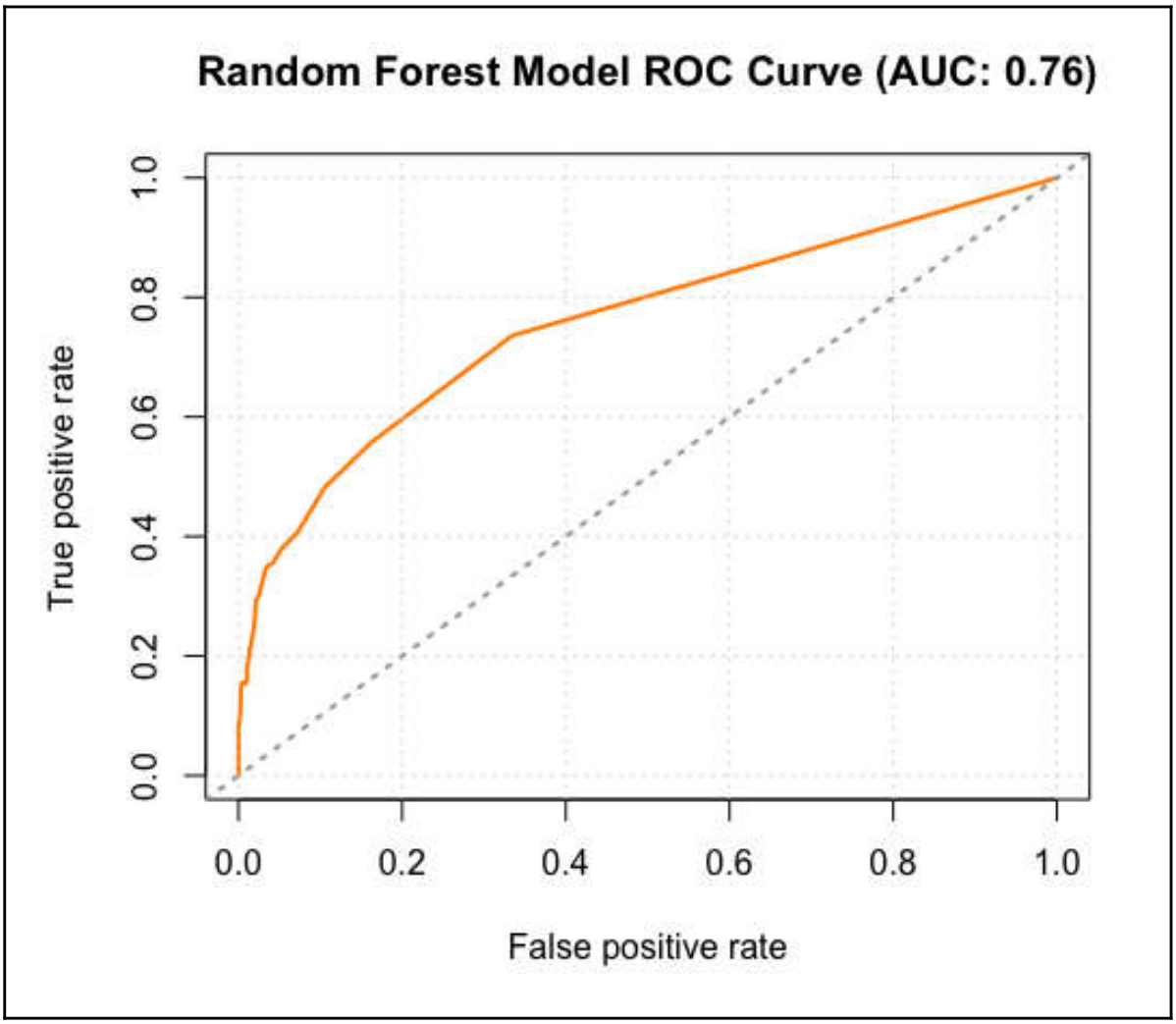
Name	Type	Value
rfModel	list [18] (S3: randomForest)	List of length 18
call	language	randomForest(x = trainX, y = factor(trainY), ntree = 200, maxnodes = 24)
type	character [1]	'classification'
predicted	factor	Factor with 6393 levels: "0", "0", "0", "0", "0", "0" , ...
err.rate	double [200 x 3]	0.12019 0.11917 0.12560 0.12535 0.12870 0.12930 0.00884 0.00792 0.00802 0.0...
confusion	double [2 x 3]	5.49e+03 8.06e+02 1.00e+01 8.90e+01 1.82e-03 9.01e-01
votes	double [6393 x 2] (S3: matrix	1.0000 1.0000 1.0000 0.9861 1.0000 1.0000 0.0000 0.0000 0.0000 0.0139 0.0000 ...
oob.times	double [6393]	80 74 75 72 72 68 ...
classes	character [2]	'0' '1'
importance	double [42 x 1]	8.807 1.574 1.074 1.301 0.954 2.164 ...
importanceSD	NULL	Pairlist of length 0
localImportance	NULL	Pairlist of length 0
proximity	NULL	Pairlist of length 0
ntree	double [1]	200
mtry	double [1]	6
forest	list [14]	List of length 14
y	factor	Factor with 6393 levels: "0", "0", "0", "0", "1", "0" , ...
test	NULL	Pairlist of length 0
inbag	NULL	Pairlist of length 0

```
> getTree(rfModel, 1)
   left daughter right daughter split var split point status prediction
1              2              3        42   302.37692      1          0
2              4              5         1     0.50000      1          0
3              6              7        35  2017.56718      1          0
4              8              9        39    11.50000      1          0
5             10             11        42    49.16778      1          0
6             12             13        38     9.00000      1          0
7             14             15        39    96.50000      1          0
8             16             17         3     0.50000      1          0
9             18             19        35  3884.07793      1          0
10            20             21        26     0.50000      1          0
11            22             23        24     0.50000      1          0
12             0              0         0     0.00000     -1          1
13            24             25         5     0.50000      1          0
14            26             27        36 10261.50000      1          0
15             0              0         0     0.00000     -1          1
16            28             29         5     0.50000      1          0
17             0              0         0     0.00000     -1          1
18            30             31        36 63155.50000      1          0
19            32             33        22     0.50000      1          0
20            34             35        39    33.50000      1          0
```

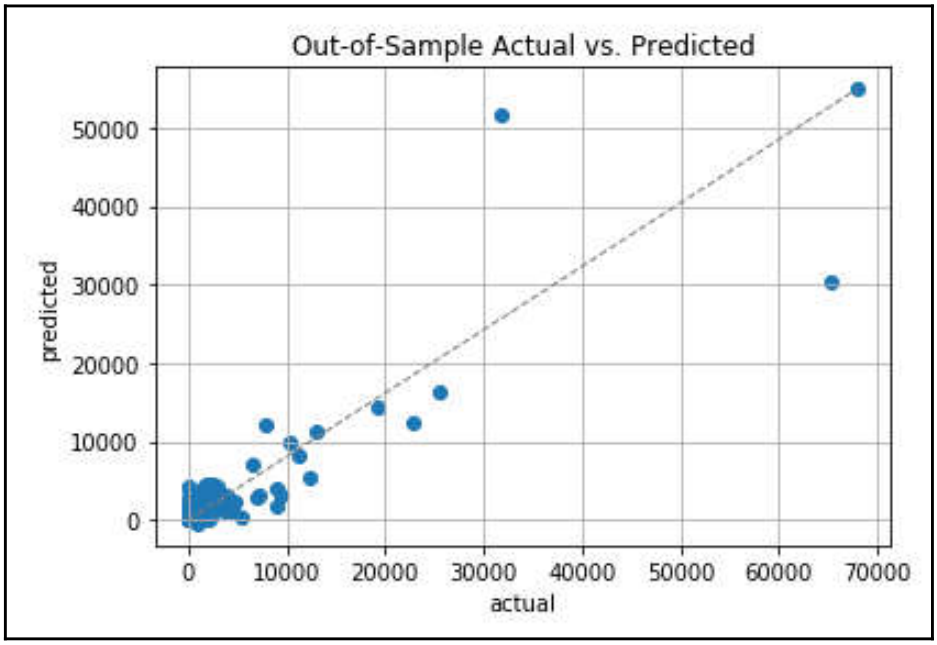
```
> predict(rfModel, trainX, predict.all=TRUE)$individual
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
2    "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"
3    "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"
4    "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"
5    "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"
7    "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "0"  "1"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"   "0"
```

```
> importance(rfModel)
                                MeanDecreaseGini
Sales.ChannelAgent                    8.80679847
Sales.ChannelBranch                   1.57399368
Sales.ChannelCall Center              1.07412042
Sales.ChannelWeb                      1.30109148
Vehicle.SizeMedsize                   0.95446772
Vehicle.SizeSmall                     2.16432825
Vehicle.ClassLuxury Car               0.14581500
Vehicle.ClassLuxury SUV               0.09509184
Vehicle.ClassSports Car               0.69120913
Vehicle.ClassSUV                      0.95105991
Vehicle.ClassTwo-Door Car             0.51935534
PolicyCorporate L2                    0.43019126
PolicyCorporate L3                    0.29868539
PolicyPersonal L1                     0.28883213
PolicyPersonal L2                     0.16178421
PolicyPersonal L3                     0.23481424
PolicySpecial L1                      0.36156105
PolicySpecial L2                      0.06004588
PolicySpecial L3                      0.63030395
Policy.TypePersonal Auto              0.16764506
Policy.TypeSpecial Auto               0.36371783
EmploymentStatusEmployed              4.94663314
EmploymentStatusMedical Leave         0.58598988
EmploymentStatusRetired              70.39553508
EmploymentStatusUnemployed            5.81568328
Marital.StatusMarried                 3.86505447
Marital.StatusSingle                  2.64214736
EducationCollege                      1.35925811
EducationDoctor                       0.54873538
EducationHigh School or Below         0.88504666
EducationMaster                       1.09701383
CoverageExtended                      0.89261823
CoveragePremium                       0.57386404
GenderM                               0.87952114
```

```
> # - Accuracy, Precision, and Recall
> inSampleAccuracy <- mean(trainY == inSamplePreds)
> outSampleAccuracy <- mean(testY == outSamplePreds)
> print(sprintf('In-Sample Accuracy: %0.4f', inSampleAccuracy))
[1] "In-Sample Accuracy: 0.8756"
> print(sprintf('Out-Sample Accuracy: %0.4f', outSampleAccuracy))
[1] "Out-Sample Accuracy: 0.8636"
>
> inSamplePrecision <- sum(inSamplePreds & trainY) / sum(inSamplePreds)
> outSamplePrecision <- sum(outSamplePreds & testY) / sum(outSamplePreds)
> print(sprintf('In-Sample Precision: %0.4f', inSamplePrecision))
[1] "In-Sample Precision: 0.9717"
> print(sprintf('Out-Sample Precision: %0.4f', outSamplePrecision))
[1] "Out-Sample Precision: 0.8980"
>
> inSampleRecall <- sum(inSamplePreds & trainY) / sum(trainY)
> outSampleRecall <- sum(outSamplePreds & testY) / sum(testY)
> print(sprintf('In-Sample Recall: %0.4f', inSampleRecall))
[1] "In-Sample Recall: 0.1151"
> print(sprintf('Out-Sample Recall: %0.4f', outSampleRecall))
[1] "Out-Sample Recall: 0.1065"
```


Random Forest Model ROC Curve (AUC: 0.76)
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

Chapter 9: Customer Lifetime Value




```
df = pd.read_excel('../data/Online Retail.xlsx', sheet_name='Online Retail')
```

```
df.shape
```

```
(541909, 8)
```

```
df.head()
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

```
print('Date Range: %s ~ %s' % (df['InvoiceDate'].min(), df['InvoiceDate'].max()))
```

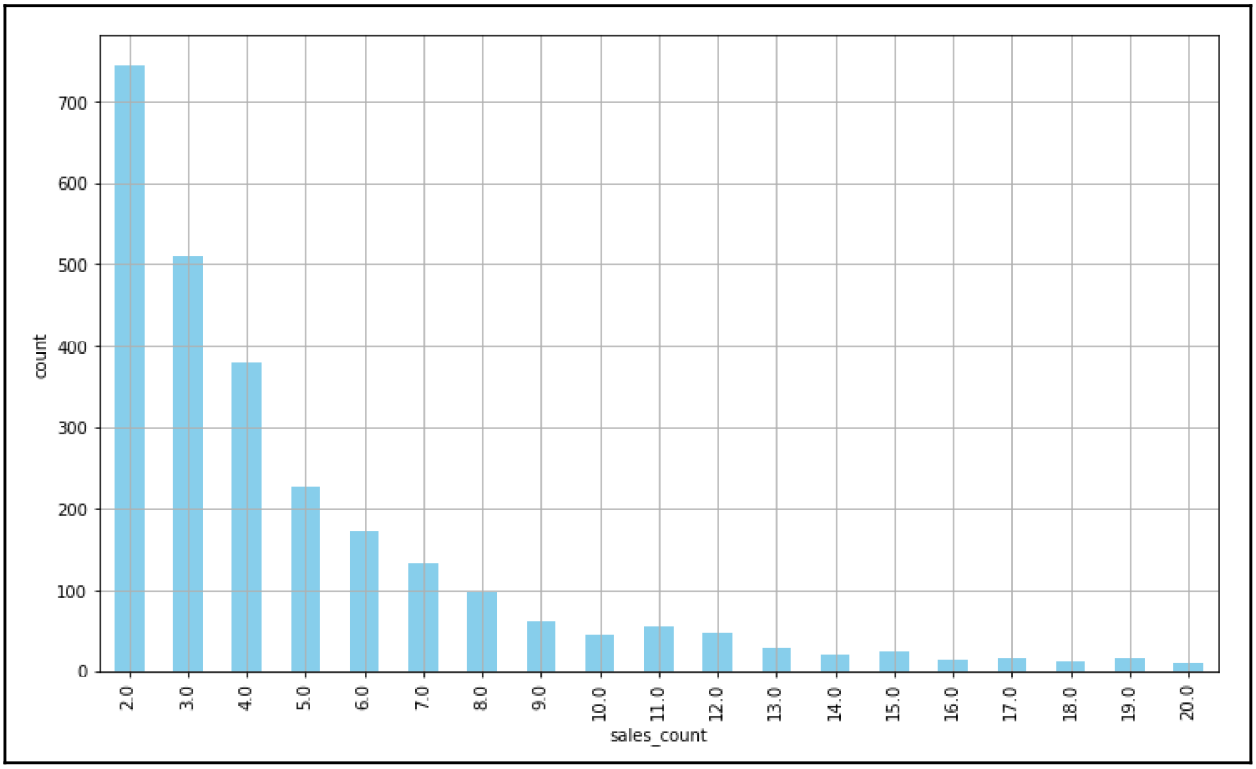
```
Date Range: 2010-12-01 08:26:00 ~ 2011-12-09 12:50:00
```

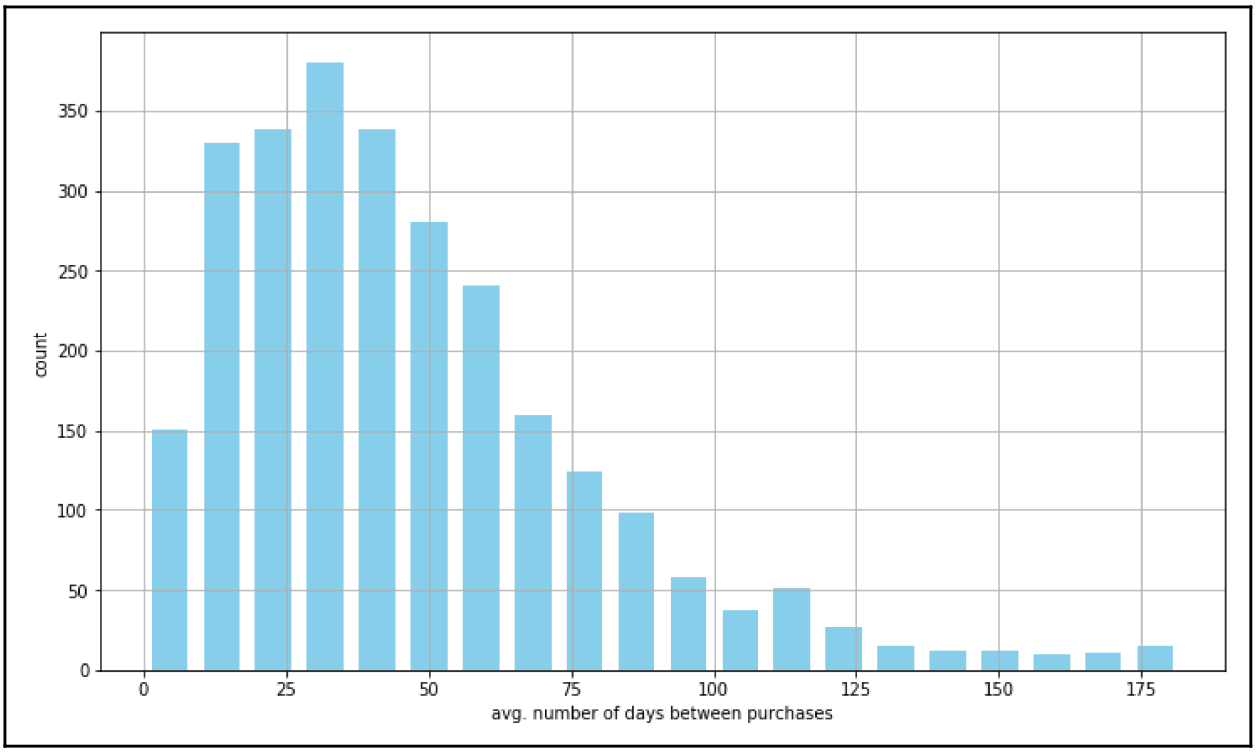
```
orders_df
```

CustomerID	InvoiceNo	Sales	InvoiceDate
12346.0	**541431**	77183.60	2011-01-18 10:01:00
12347.0	**537626**	711.79	2010-12-07 14:57:00
	542237	475.39	2011-01-26 14:30:00
	549222	636.25	2011-04-07 10:43:00
	556201	382.52	2011-06-09 13:01:00
	562032	584.91	2011-08-02 08:48:00
	573511	1294.32	2011-10-31 12:25:00
12348.0	**539318**	892.80	2010-12-16 19:09:00
	541998	227.44	2011-01-25 10:42:00
	548955	367.00	2011-04-05 10:47:00
	568172	310.00	2011-09-25 13:13:00

```
summary_df
```

	Sales					InvoiceDate			
	min	max	sum	avg	count	min	max	purchase_duration	purchase_frequency
CustomerID									
12346.0	77183.60	77183.60	77183.60	77183.600000	1.0	2011-01-18 10:01:00	2011-01-18 10:01:00	0	0.000000
12347.0	382.52	1294.32	4085.18	680.863333	6.0	2010-12-07 14:57:00	2011-10-31 12:25:00	327	54.500000
12348.0	227.44	892.80	1797.24	449.310000	4.0	2010-12-16 19:09:00	2011-09-25 13:13:00	282	70.500000
12349.0	1757.55	1757.55	1757.55	1757.550000	1.0	2011-11-21 09:51:00	2011-11-21 09:51:00	0	0.000000
12350.0	334.40	334.40	334.40	334.400000	1.0	2011-02-02 16:01:00	2011-02-02 16:01:00	0	0.000000
12352.0	120.33	840.30	2506.04	313.255000	8.0	2011-02-16 12:33:00	2011-11-03 14:37:00	260	32.500000
12353.0	89.00	89.00	89.00	89.000000	1.0	2011-05-19 17:47:00	2011-05-19 17:47:00	0	0.000000
12354.0	1079.40	1079.40	1079.40	1079.400000	1.0	2011-04-21 13:11:00	2011-04-21 13:11:00	0	0.000000
12355.0	459.40	459.40	459.40	459.400000	1.0	2011-05-09 13:49:00	2011-05-09 13:49:00	0	0.000000
12356.0	58.35	2271.62	2811.43	937.143333	3.0	2011-01-18 09:50:00	2011-11-17 08:40:00	302	100.666667




count
avg. number of days between purchases
0
25
50
75
100
125
150
175
0
50
100
150
200
250
300
350

```
data_df.head(10)
```

	CustomerID	InvoiceDate	sales_sum	sales_avg	sales_count
0	12346.0	2011-03-31	77183.60	77183.600	1.0
1	12347.0	2010-12-31	711.79	711.790	1.0
2	12347.0	2011-03-31	475.39	475.390	1.0
3	12347.0	2011-06-30	1018.77	509.385	2.0
4	12347.0	2011-09-30	584.91	584.910	1.0
5	12347.0	2011-12-31	1294.32	1294.320	1.0
6	12348.0	2010-12-31	892.80	892.800	1.0
7	12348.0	2011-03-31	227.44	227.440	1.0
8	12348.0	2011-06-30	367.00	367.000	1.0
9	12348.0	2011-09-30	310.00	310.000	1.0

```
data_df.head(10)
```

	CustomerID	InvoiceDate	sales_sum	sales_avg	sales_count	M
0	12346.0	2011-03-31	77183.60	77183.600	1.0	M_4
1	12347.0	2010-12-31	711.79	711.790	1.0	M_5
2	12347.0	2011-03-31	475.39	475.390	1.0	M_4
3	12347.0	2011-06-30	1018.77	509.385	2.0	M_3
4	12347.0	2011-09-30	584.91	584.910	1.0	M_2
5	12347.0	2011-12-31	1294.32	1294.320	1.0	M_1
6	12348.0	2010-12-31	892.80	892.800	1.0	M_5
7	12348.0	2011-03-31	227.44	227.440	1.0	M_4
8	12348.0	2011-06-30	367.00	367.000	1.0	M_3
9	12348.0	2011-09-30	310.00	310.000	1.0	M_2

```
features_df.head(10)
```

CustomerID	sales_avg_M_2	sales_avg_M_3	sales_avg_M_4	sales_avg_M_5	sales_count_M_2	sales_count_M_3	sales_count_M_4	sales_count_M_5	sales_sum_M
12346.0	NaN	NaN	77183.600	NaN	NaN	NaN	1.0	NaN	N
12347.0	584.91	509.385	475.390	711.79	1.0	2.0	1.0	1.0	584
12348.0	310.00	367.000	227.440	892.80	1.0	1.0	1.0	1.0	310
12350.0	NaN	NaN	334.400	NaN	NaN	NaN	1.0	NaN	N
12352.0	316.25	NaN	312.362	NaN	2.0	NaN	5.0	NaN	632
12353.0	NaN	89.000	NaN	NaN	NaN	1.0	NaN	NaN	N
12354.0	NaN	1079.400	NaN	NaN	NaN	1.0	NaN	NaN	N
12355.0	NaN	459.400	NaN	NaN	NaN	1.0	NaN	NaN	N
12356.0	NaN	481.460	2271.620	NaN	NaN	1.0	1.0	NaN	N
12358.0	484.86	NaN	NaN	NaN	1.0	NaN	NaN	NaN	484

```
response_df.head(10)
```

	CustomerID	CLV_3M
5	12347.0	1294.32
10	12349.0	1757.55
14	12352.0	311.73
20	12356.0	58.35
21	12357.0	6207.67
25	12359.0	2876.85
28	12360.0	1043.78
33	12362.0	2119.85
37	12364.0	299.06
41	12370.0	739.28

	CLV_3M	CustomerID	sales_sum_M_5	sales_sum_M_4	sales_sum_M_3	sales_sum_M_2	sales_count_M_5	sales_count_M_4	sales_count_M_3	sales_count
9219	0.00	12346.0	0.00	77183.60	0.00	0.00	0.0	1.0	0.0	
5	1294.32	12347.0	711.79	475.39	1018.77	584.91	1.0	1.0	2.0	
9219	0.00	12348.0	892.80	227.44	367.00	310.00	1.0	1.0	1.0	
9219	0.00	12350.0	0.00	334.40	0.00	0.00	0.0	1.0	0.0	
14	311.73	12352.0	0.00	1561.81	0.00	632.50	0.0	5.0	0.0	
9219	0.00	12353.0	0.00	0.00	89.00	0.00	0.0	0.0	1.0	
9219	0.00	12354.0	0.00	0.00	1079.40	0.00	0.0	0.0	1.0	
9219	0.00	12355.0	0.00	0.00	459.40	0.00	0.0	0.0	1.0	
20	58.35	12356.0	0.00	2271.62	481.46	0.00	0.0	1.0	1.0	
9219	0.00	12358.0	0.00	0.00	0.00	484.86	0.0	0.0	0.0	

```
coef = pd.DataFrame(list(zip(all_features, reg_fit.coef_)))
coef.columns = ['feature', 'coef']

coef
```

	feature	coef
0	sales_avg_M_2	-0.053913
1	sales_avg_M_3	0.162335
2	sales_avg_M_4	0.241964
3	sales_avg_M_5	-0.550508
4	sales_count_M_2	41.247136
5	sales_count_M_3	40.512827
6	sales_count_M_4	62.766692
7	sales_count_M_5	-7.927177
8	sales_sum_M_2	0.533077
9	sales_sum_M_3	0.053559
10	sales_sum_M_4	-0.214531
11	sales_sum_M_5	0.604951

- R-Squared

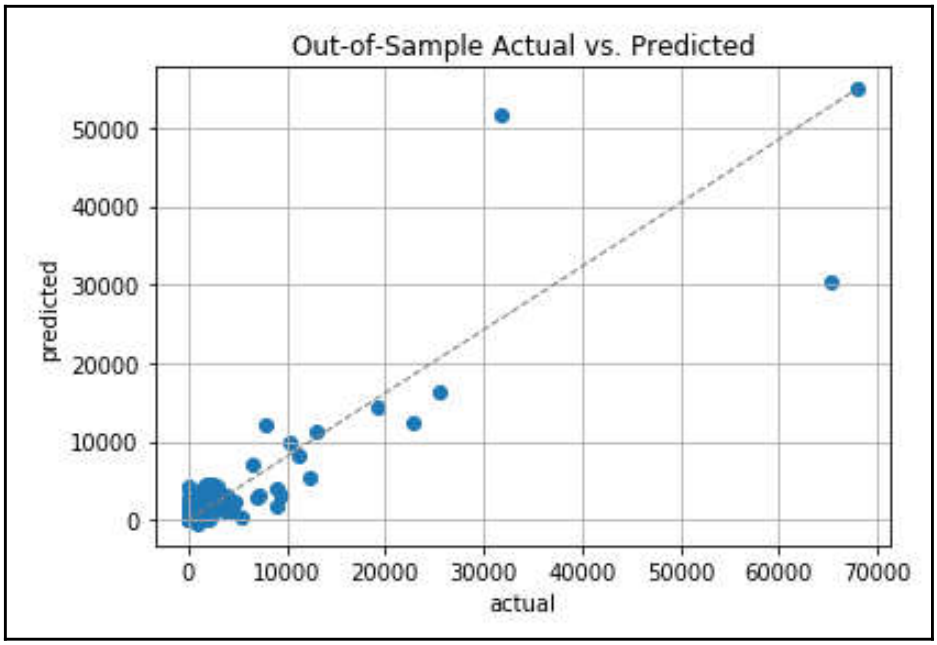
```
print('In-Sample R-Squared: %0.4f' % r2_score(y_true=y_train, y_pred=train_preds))
print('Out-of-Sample R-Squared: %0.4f' % r2_score(y_true=y_test, y_pred=test_preds))
```

```
In-Sample R-Squared: 0.4445
Out-of-Sample R-Squared: 0.7947
```

- Median Absolute Error

```
print('In-Sample MSE: %0.4f' % median_absolute_error(y_true=y_train, y_pred=train_preds))
print('Out-of-Sample MSE: %0.4f' % median_absolute_error(y_true=y_test, y_pred=test_preds))
```

```
In-Sample MSE: 178.2854
Out-of-Sample MSE: 178.7393
```

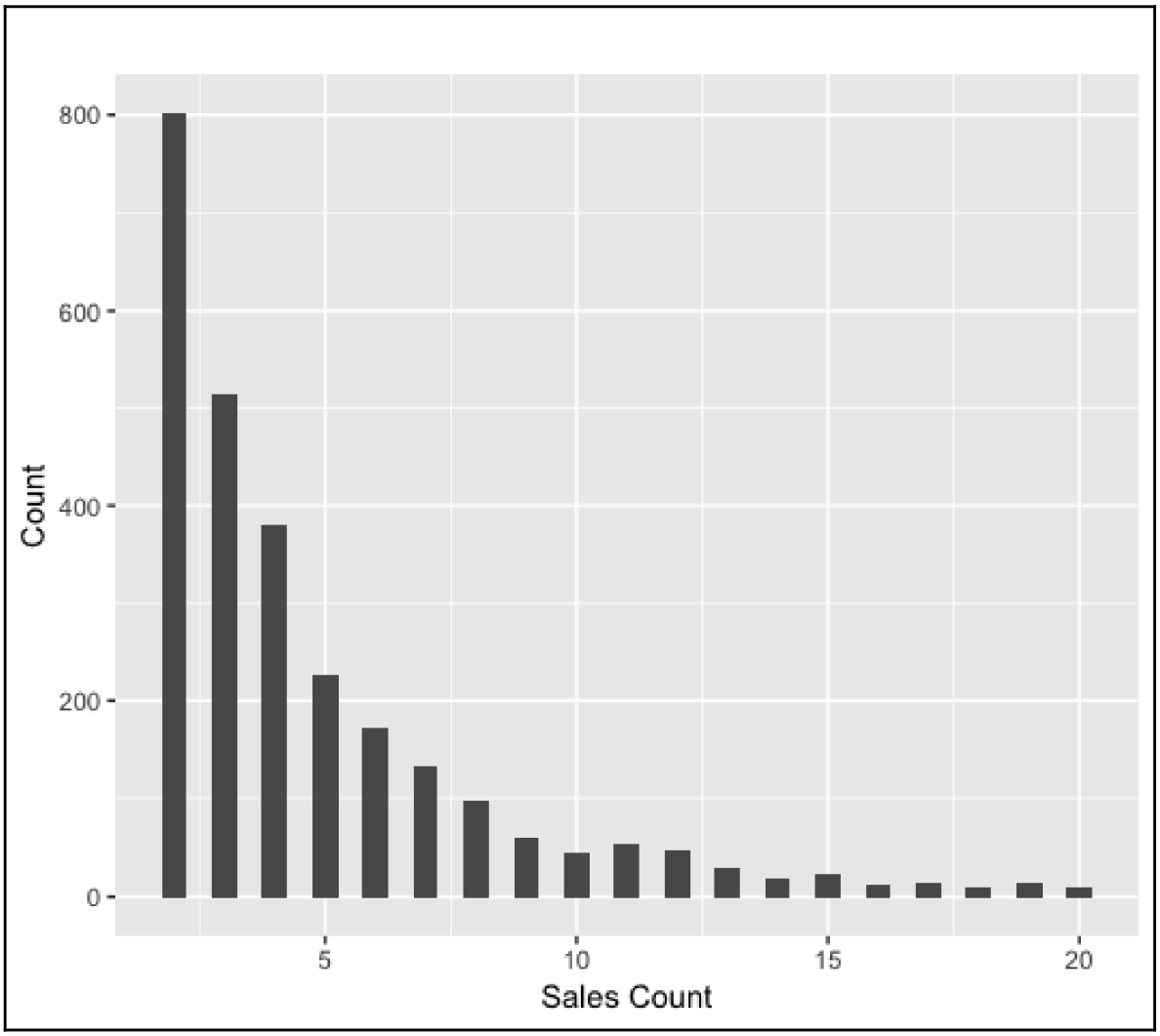


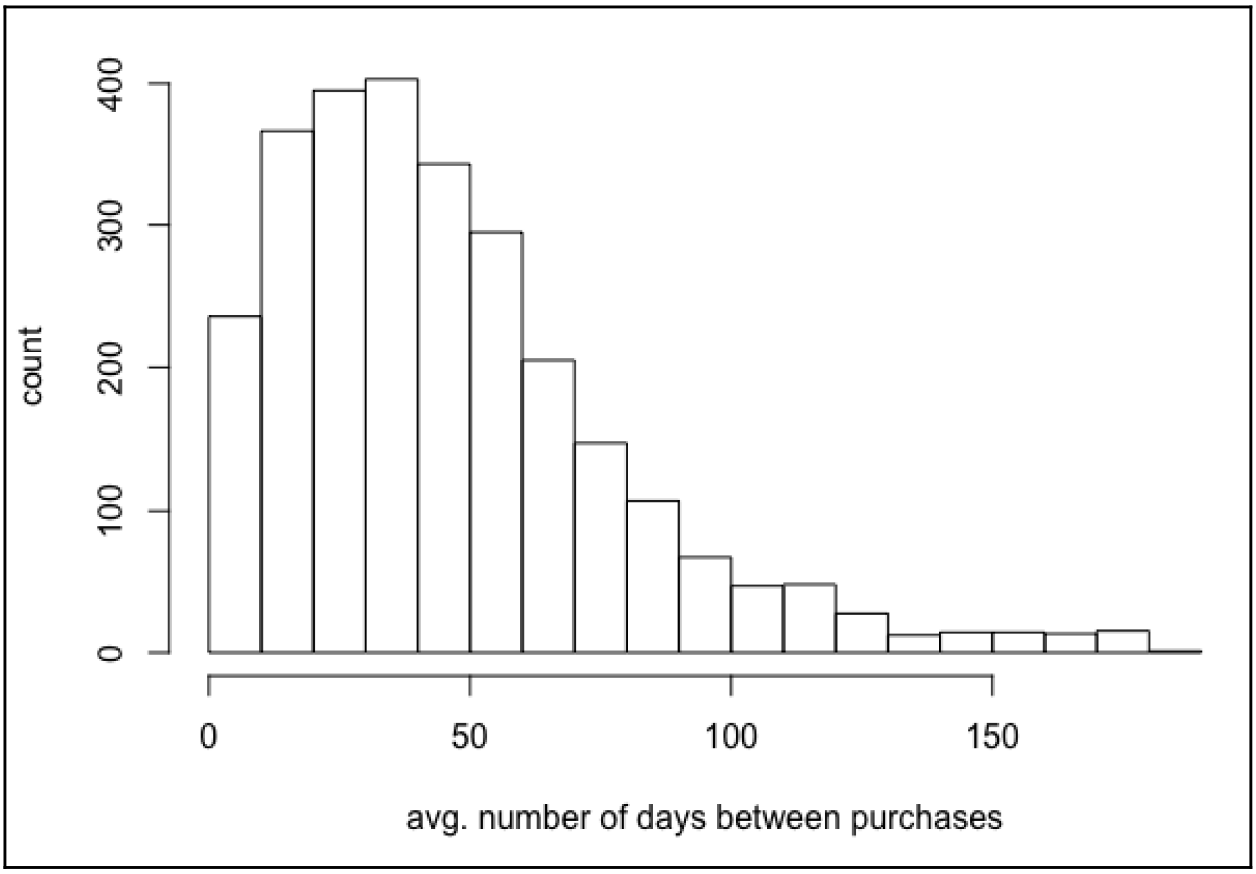
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
5	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
6	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850	United Kingdom
7	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850	United Kingdom
8	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
9	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
10	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047	United Kingdom
11	536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	2010-12-01 08:34:00	2.10	13047	United Kingdom
12	536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	2010-12-01 08:34:00	2.10	13047	United Kingdom
13	536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	2010-12-01 08:34:00	3.75	13047	United Kingdom
14	536367	22310	IVORY KNITTED MUG COSY	6	2010-12-01 08:34:00	1.65	13047	United Kingdom
15	536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	2010-12-01 08:34:00	4.25	13047	United Kingdom

```
> sprintf("Date Range: %s ~ %s", min(df$InvoiceDate), max(df$InvoiceDate))
[1] "Date Range: 2010-12-01 08:26:00 ~ 2011-12-09 12:50:00"
```

	CustomerID	InvoiceNo	Sales	InvoiceDate
1	12346	541431	77183.60	2011-01-18 10:01:00
2	12347	537626	711.79	2010-12-07 14:57:00
3	12347	542237	475.39	2011-01-26 14:30:00
4	12347	549222	636.25	2011-04-07 10:43:00
5	12347	556201	382.52	2011-06-09 13:01:00
6	12347	562032	584.91	2011-08-02 08:48:00
7	12347	573511	1294.32	2011-10-31 12:25:00
8	12348	539318	892.80	2010-12-16 19:09:00
9	12348	541998	227.44	2011-01-25 10:42:00
10	12348	548955	367.00	2011-04-05 10:47:00
11	12348	568172	310.00	2011-09-25 13:13:00
12	12349	577609	1757.55	2011-11-21 09:51:00
13	12350	543037	334.40	2011-02-02 16:01:00
14	12352	544156	296.50	2011-02-16 12:33:00
15	12352	545323	144.35	2011-03-01 14:57:00

	CustomerID	SalesMin	SalesMax	SalesSum	SalesAvg	SalesCount	InvoiceDateMin	InvoiceDateMax	PurchaseDuration	PurchaseFrequency
1	12346	77183.60	77183.60	77183.60	77183.6000	1	2011-01-18 10:01:00	2011-01-18 10:01:00	0	0.000000
2	12347	382.52	1294.32	4085.18	680.8633	6	2010-12-07 14:57:00	2011-10-31 12:25:00	327	54.500000
3	12348	227.44	892.80	1797.24	449.3100	4	2010-12-16 19:09:00	2011-09-25 13:13:00	282	70.500000
4	12349	1757.55	1757.55	1757.55	1757.5500	1	2011-11-21 09:51:00	2011-11-21 09:51:00	0	0.000000
5	12350	334.40	334.40	334.40	334.4000	1	2011-02-02 16:01:00	2011-02-02 16:01:00	0	0.000000
6	12352	120.33	840.30	2506.04	313.2550	8	2011-02-16 12:33:00	2011-11-03 14:37:00	260	32.500000
7	12353	89.00	89.00	89.00	89.0000	1	2011-05-19 17:47:00	2011-05-19 17:47:00	0	0.000000
8	12354	1079.40	1079.40	1079.40	1079.4000	1	2011-04-21 13:11:00	2011-04-21 13:11:00	0	0.000000
9	12355	459.40	459.40	459.40	459.4000	1	2011-05-09 13:49:00	2011-05-09 13:49:00	0	0.000000
10	12356	58.35	2271.62	2811.43	937.1433	3	2011-01-18 09:50:00	2011-11-17 08:40:00	302	100.666667
11	12357	6207.67	6207.67	6207.67	6207.6700	1	2011-11-06 16:07:00	2011-11-06 16:07:00	0	0.000000
12	12358	484.86	484.86	484.86	484.8600	1	2011-07-12 10:04:00	2011-07-12 10:04:00	0	0.000000
13	12359	547.50	2876.85	6372.58	1593.1450	4	2011-01-12 12:43:00	2011-10-13 12:47:00	274	68.500000
14	12360	534.70	1083.58	2662.06	887.3533	3	2011-05-23 09:43:00	2011-10-18 15:22:00	148	49.333333
15	12361	189.90	189.90	189.90	189.9000	1	2011-02-25 13:51:00	2011-02-25 13:51:00	0	0.000000




count
0
100
200
300
400
0
50
100
150
avg. number of days between purchases

	CustomerID	Quarter	SalesSum	SalesAvg	SalesCount
1	12346	2011-01-01	77183.60	77183.6000	1
2	12347	2011-01-01	1187.18	593.5900	2
3	12347	2011-04-01	636.25	636.2500	1
4	12347	2011-07-01	967.43	483.7150	2
5	12347	2011-10-01	1294.32	1294.3200	1
6	12348	2011-01-01	1120.24	560.1200	2
7	12348	2011-04-01	367.00	367.0000	1
8	12348	2011-10-01	310.00	310.0000	1
9	12349	2012-01-01	1757.55	1757.5500	1
10	12350	2011-01-01	334.40	334.4000	1
11	12352	2011-04-01	1561.81	312.3620	5
12	12352	2011-10-01	944.23	314.7433	3
13	12353	2011-07-01	89.00	89.0000	1
14	12354	2011-04-01	1079.40	1079.4000	1
15	12355	2011-04-01	459.40	459.4000	1

	CustomerID	Quarter	SalesSum	SalesAvg	SalesCount
1	12346	Q5	77183.60	77183.6000	1
2	12347	Q5	1187.18	593.5900	2
3	12347	Q4	636.25	636.2500	1
4	12347	Q3	967.43	483.7150	2
5	12347	Q2	1294.32	1294.3200	1
6	12348	Q5	1120.24	560.1200	2
7	12348	Q4	367.00	367.0000	1
8	12348	Q2	310.00	310.0000	1
9	12349	Q1	1757.55	1757.5500	1
10	12350	Q5	334.40	334.4000	1
11	12352	Q4	1561.81	312.3620	5
12	12352	Q2	944.23	314.7433	3
13	12353	Q3	89.00	89.0000	1
14	12354	Q4	1079.40	1079.4000	1
15	12355	Q4	459.40	459.4000	1

	CustomerID	SalesSum.Q2	SalesSum.Q3	SalesSum.Q4	SalesSum.Q5	SalesAvg.Q2	SalesAvg.Q3	SalesAvg.Q4	SalesAvg.Q5	SalesCount.Q2	SalesCount.Q3
1	12346	0.00	0.00	0.00	77183.60	0.0000	0.00000	0.0000	77183.6000	0	0
2	12347	1294.32	967.43	636.25	1187.18	1294.3200	483.71500	636.2500	593.5900	1	2
3	12348	310.00	0.00	367.00	1120.24	310.0000	0.00000	367.0000	560.1200	1	0
4	12350	0.00	0.00	0.00	334.40	0.0000	0.00000	0.0000	334.4000	0	0
5	12352	944.23	0.00	1561.81	0.00	314.7433	0.00000	312.3620	0.0000	3	0
6	12353	0.00	89.00	0.00	0.00	0.0000	89.00000	0.0000	0.0000	0	1
7	12354	0.00	0.00	1079.40	0.00	0.0000	0.00000	1079.4000	0.0000	0	0
8	12355	0.00	0.00	459.40	0.00	0.0000	0.00000	459.4000	0.0000	0	0
9	12356	0.00	0.00	481.46	2271.62	0.0000	0.00000	481.4600	2271.6200	0	0
10	12357	6207.67	0.00	0.00	0.00	6207.6700	0.00000	0.0000	0.0000	1	0
11	12358	0.00	484.86	0.00	0.00	0.0000	484.86000	0.0000	0.0000	0	1
12	12359	2876.85	1109.32	0.00	2386.41	2876.8500	1109.32000	0.0000	1193.2050	1	1
13	12360	1578.48	1083.58	0.00	0.00	789.2400	1083.58000	0.0000	0.0000	2	1
14	12361	0.00	0.00	189.90	0.00	0.0000	0.00000	189.9000	0.0000	0	0
15	12362	2949.84	773.01	974.34	0.00	589.9680	386.50500	487.1700	0.0000	5	2

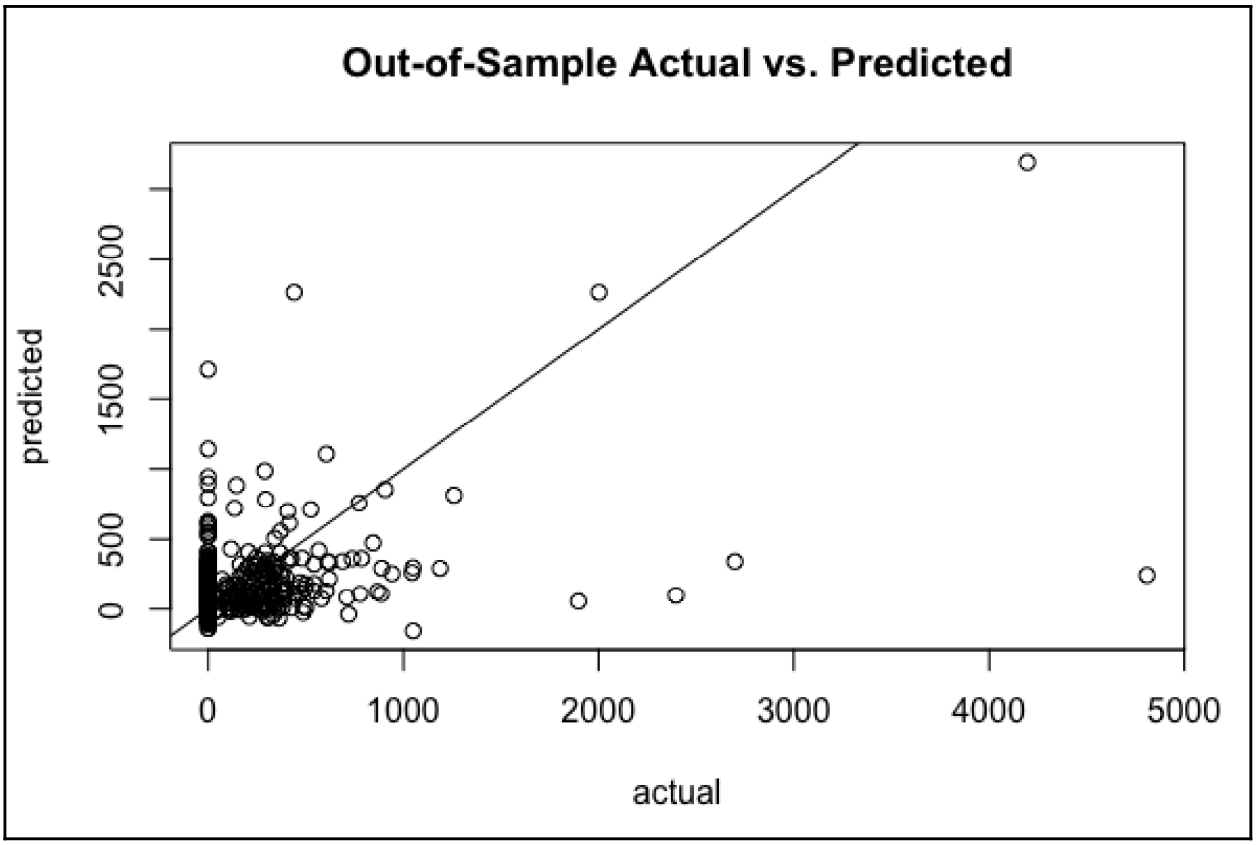
	CustomerID	CLV_3_Month
1	12349	1757.55
2	12356	58.35
3	12375	227.20
4	12380	1040.39
5	12388	286.40
6	12391	460.89
7	12395	265.83
8	12406	1794.05
9	12421	178.48
10	12427	239.72
11	12429	905.52
12	12433	2843.29
13	12437	491.01
14	12438	2016.78
15	12444	936.64

SalesSum.Q3	SalesSum.Q4	SalesSum.Q5	SalesAvg.Q2	SalesAvg.Q3	SalesAvg.Q4	SalesAvg.Q5	SalesCount.Q2	SalesCount.Q3	SalesCount.Q4	SalesCount.Q5	CLV_3_Month
0.00	0.00	77183.60	0.0000	0.00000	0.0000	77183.6000	0	0	0	1	0.00
967.43	636.25	1187.18	1294.3200	483.71500	636.2500	593.5900	1	2	1	2	0.00
0.00	367.00	1120.24	310.0000	0.00000	367.0000	560.1200	1	0	1	2	0.00
0.00	0.00	334.40	0.0000	0.00000	0.0000	334.4000	0	0	0	1	0.00
0.00	1561.81	0.00	314.7433	0.00000	312.3620	0.0000	3	0	5	0	0.00
89.00	0.00	0.00	0.0000	89.00000	0.0000	0.0000	0	1	0	0	0.00
0.00	1079.40	0.00	0.0000	0.00000	1079.4000	0.0000	0	0	1	0	0.00
0.00	459.40	0.00	0.0000	0.00000	459.4000	0.0000	0	0	1	0	0.00
0.00	481.46	2271.62	0.0000	0.00000	481.4600	2271.6200	0	0	1	1	58.35
0.00	0.00	0.00	6207.6700	0.00000	0.0000	0.0000	1	0	0	0	0.00
484.86	0.00	0.00	0.0000	484.86000	0.0000	0.0000	0	1	0	0	0.00
1109.32	0.00	2386.41	2876.8500	1109.32000	0.0000	1193.2050	1	1	0	2	0.00
1083.58	0.00	0.00	789.2400	1083.58000	0.0000	0.0000	2	1	0	0	0.00
0.00	189.90	0.00	0.0000	0.00000	189.9000	0.0000	0	0	1	0	0.00
773.01	974.34	0.00	589.9680	386.50500	487.1700	0.0000	5	2	2	0	0.00
0.00	299.10	0.00	252.9000	0.00000	299.1000	0.0000	1	0	1	0	0.00
0.00	0.00	0.00	334.2600	0.00000	0.0000	0.0000	3	0	0	0	0.00
0.00	641.38	0.00	0.0000	0.00000	320.6900	0.0000	0	0	2	0	0.00
0.00	938.39	1868.02	739.2800	0.00000	938.3900	934.0100	1	0	1	2	0.00

```
> summary(regFit)

Call:
lm(formula = CLV_3_Month ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-6486.9   -98.5   -17.8    43.8 11969.0

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -54.722109  10.987551  -4.980 6.67e-07 ***
SalesSum.Q2     0.120008   0.005131  23.389  < 2e-16 ***
SalesSum.Q3    -0.105788   0.007839 -13.495  < 2e-16 ***
SalesSum.Q4     0.012862   0.012574   1.023 0.306414
SalesSum.Q5    -0.045571   0.015615  -2.918 0.003542 **
SalesAvg.Q2     0.028505   0.022521   1.266 0.205697
SalesAvg.Q3     0.292372   0.026064  11.218  < 2e-16 ***
SalesAvg.Q4    -0.173427   0.031389  -5.525 3.55e-08 ***
SalesAvg.Q5     0.062404   0.016671   3.743 0.000185 ***
SalesCount.Q2  38.608748   6.002764   6.432 1.44e-10 ***
SalesCount.Q3  23.972964   7.543423   3.178 0.001497 **
SalesCount.Q4  21.170655   8.186019   2.586 0.009746 **
SalesCount.Q5  27.466354   7.424798   3.699 0.000220 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 435.8 on 3310 degrees of freedom
Multiple R-squared:  0.4363,    Adjusted R-squared:  0.4342
F-statistic: 213.5 on 12 and 3310 DF,  p-value: < 2.2e-16
```

```
> sprintf('In-Sample R-Squared: %0.4f', inSampleR2)
[1] "In-Sample R-Squared: 0.4557"
> sprintf('Out-of-Sample R-Squared: %0.4f', outOfSampleR2)
[1] "Out-of-Sample R-Squared: 0.1235"
```

```
> sprintf('In-Sample MAE: %0.4f', inSampleMAE)
[1] "In-Sample MAE: 69.6753"
> sprintf('Out-of-Sample MAE: %0.4f', outOfSampleMAE)
[1] "Out-of-Sample MAE: 66.9589"
```



Chapter 10: Data-Driven Customer Segmentation

```
df.head()
```

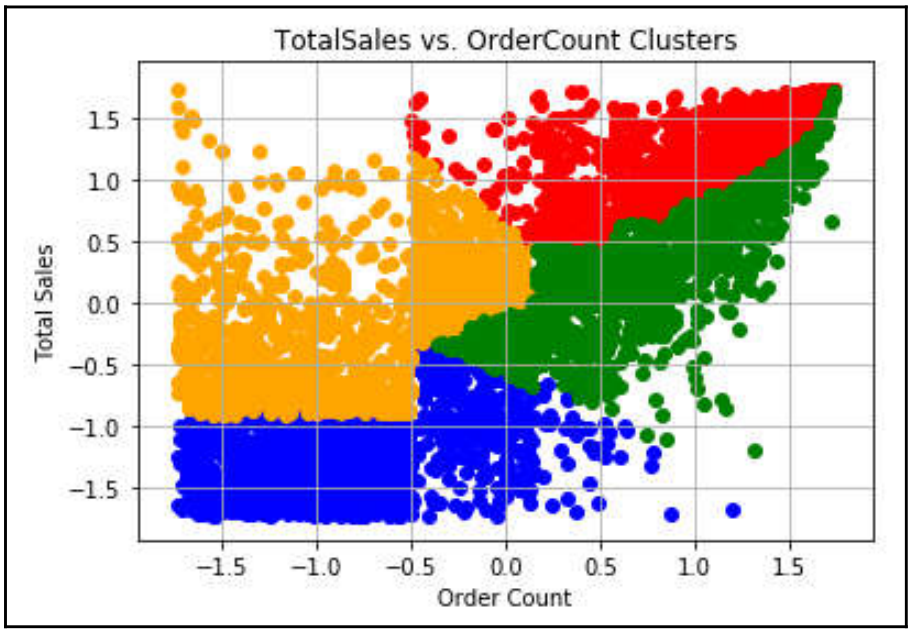
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

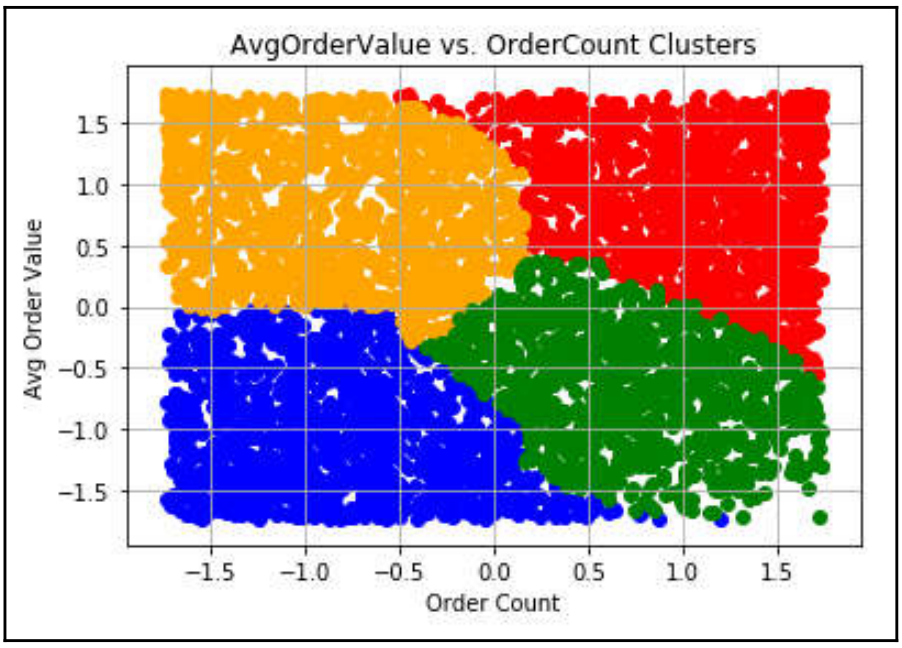
CustomerID	TotalSales	OrderCount	AvgOrderValue
12346.0	77183.60	1	77183.600000
12347.0	4085.18	6	680.863333
12348.0	1797.24	4	449.310000
12349.0	1757.55	1	1757.550000
12350.0	334.40	1	334.400000
12352.0	2506.04	8	313.255000
12353.0	89.00	1	89.000000
12354.0	1079.40	1	1079.400000
12355.0	459.40	1	459.400000
12356.0	2811.43	3	937.143333
12357.0	6207.67	1	6207.670000
12358.0	484.86	1	484.860000
12359.0	6372.58	4	1593.145000
12360.0	2662.06	3	887.353333
12361.0	189.90	1	189.900000

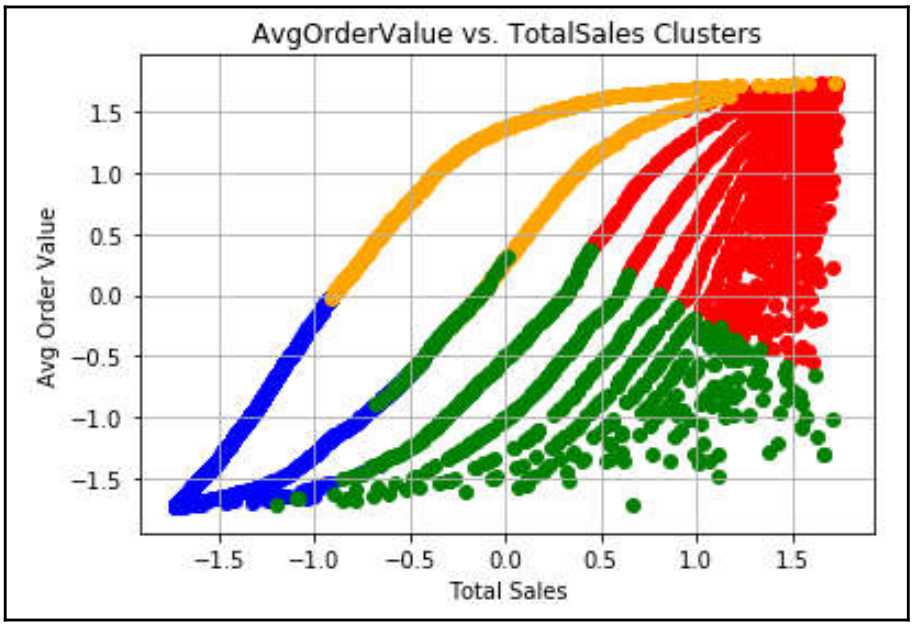
CustomerID	TotalSales	OrderCount	AvgOrderValue
12346.0	4290.0	1.0	4298.0
12347.0	3958.0	3470.0	3888.0
12348.0	3350.0	2861.0	3303.0
12349.0	3321.0	2.0	4238.0
12350.0	1241.0	3.0	2561.0
12352.0	3630.0	3774.0	2360.0
12353.0	119.0	4.0	201.0
12354.0	2781.0	5.0	4151.0
12355.0	1670.0	6.0	3354.0
12356.0	3724.0	2346.0	4082.0
12357.0	4111.0	7.0	4295.0
12358.0	1738.0	8.0	3447.0
12359.0	4117.0	2862.0	4225.0
12360.0	3680.0	2347.0	4057.0
12361.0	607.0	9.0	1186.0

CustomerID	TotalSales	OrderCount	AvgOrderValue
12346.0	1.724999	-1.731446	1.731446
12347.0	1.457445	1.064173	1.401033
12348.0	0.967466	0.573388	0.929590
12349.0	0.944096	-1.730641	1.683093
12350.0	-0.732148	-1.729835	0.331622
12352.0	1.193114	1.309162	0.169639
12353.0	-1.636352	-1.729029	-1.570269
12354.0	0.508917	-1.728223	1.612981
12355.0	-0.386422	-1.727417	0.970690
12356.0	1.268868	0.158357	1.557375
12357.0	1.580746	-1.726611	1.729029
12358.0	-0.331622	-1.725805	1.045637
12359.0	1.585581	0.574194	1.672617
12360.0	1.233409	0.159163	1.537228
12361.0	-1.243079	-1.724999	-0.776471

	TotalSales	OrderCount	AvgOrderValue
count	4.298000e+03	4.298000e+03	4.298000e+03
mean	9.952744e-17	-1.231371e-16	5.719018e-17
std	1.000000e+00	1.000000e+00	1.000000e+00
min	-1.731446e+00	-1.731446e+00	-1.731446e+00
25%	-8.657232e-01	-8.657232e-01	-8.657232e-01
50%	0.000000e+00	0.000000e+00	0.000000e+00
75%	8.657232e-01	8.657232e-01	8.657232e-01
max	1.731446e+00	1.731446e+00	1.731446e+00


TotalSales vs. OrderCount Clusters
Total Sales
Order Count
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


AvgOrderValue vs. OrderCount Clusters
Avg Order Value
Order Count
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5




```
Silhouette Score for 4 Clusters: 0.4113
Silhouette Score for 5 Clusters: 0.3771
Silhouette Score for 6 Clusters: 0.3784
Silhouette Score for 7 Clusters: 0.3906
Silhouette Score for 8 Clusters: 0.3810
```

CustomerID	TotalSales	OrderCount	AvgOrderValue	Cluster
12346.0	1.724999	-1.731446	1.731446	0
12347.0	1.457445	1.064173	1.401033	2
12348.0	0.967466	0.573388	0.929590	2
12349.0	0.944096	-1.730641	1.683093	0
12350.0	-0.732148	-1.729835	0.331622	0
12352.0	1.193114	1.309162	0.169639	2
12353.0	-1.636352	-1.729029	-1.570269	3
12354.0	0.508917	-1.728223	1.612981	0
12355.0	-0.386422	-1.727417	0.970690	0
12356.0	1.268868	0.158357	1.557375	2
12357.0	1.580746	-1.726611	1.729029	0
12358.0	-0.331622	-1.725805	1.045637	0
12359.0	1.585581	0.574194	1.672617	2
12360.0	1.233409	0.159163	1.537228	2
12361.0	-1.243079	-1.724999	-0.776471	3

```
array([[-0.13330681, -0.84982057,  0.79745159],
       [ 0.21794823,  0.715536  , -0.64337832],
       [ 1.20630621,  1.00552238,  0.86837366],
       [-1.24675221, -0.7971239 , -1.06197333]])
```

Description	StockCode
JUMBO BAG RED RETROSPOT	1143
REGENCY CAKESTAND 3 TIER	1078
WHITE HANGING HEART T-LIGHT HOLDER	1072
LUNCH BAG RED RETROSPOT	937
PARTY BUNTING	865

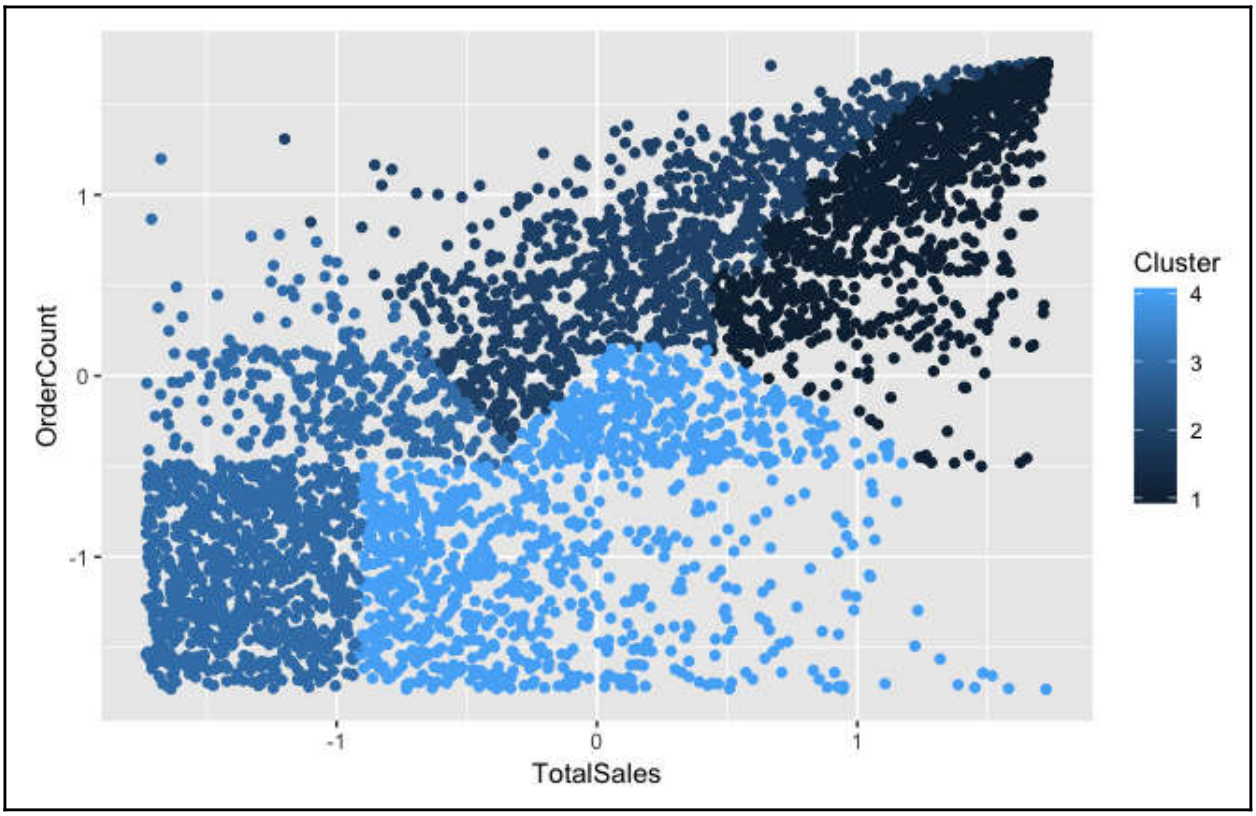
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
5	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
6	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850	United Kingdom
7	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850	United Kingdom
8	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
9	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
10	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047	United Kingdom
11	536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	2010-12-01 08:34:00	2.10	13047	United Kingdom
12	536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	2010-12-01 08:34:00	2.10	13047	United Kingdom
13	536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	2010-12-01 08:34:00	3.75	13047	United Kingdom
14	536367	22310	IVORY KNITTED MUG COSY	6	2010-12-01 08:34:00	1.65	13047	United Kingdom
15	536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	2010-12-01 08:34:00	4.25	13047	United Kingdom

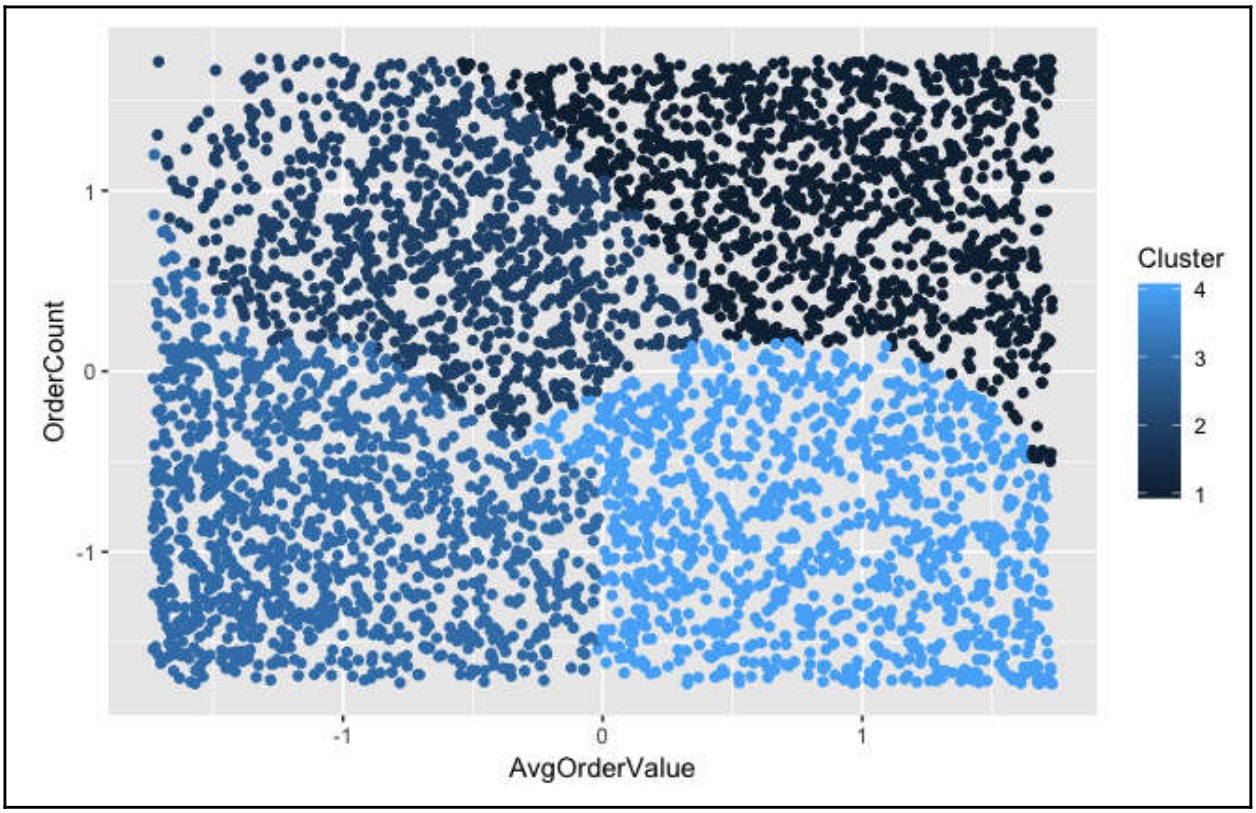
	CustomerID	TotalSales	OrderCount	AvgOrderValue
1	12346	77183.60	1	77183.6000
2	12347	4085.18	6	680.8633
3	12348	1797.24	4	449.3100
4	12349	1757.55	1	1757.5500
5	12350	334.40	1	334.4000
6	12352	2506.04	8	313.2550
7	12353	89.00	1	89.0000
8	12354	1079.40	1	1079.4000
9	12355	459.40	1	459.4000
10	12356	2811.43	3	937.1433
11	12357	6207.67	1	6207.6700
12	12358	484.86	1	484.8600
13	12359	6372.58	4	1593.1450
14	12360	2662.06	3	887.3533
15	12361	189.90	1	189.9000

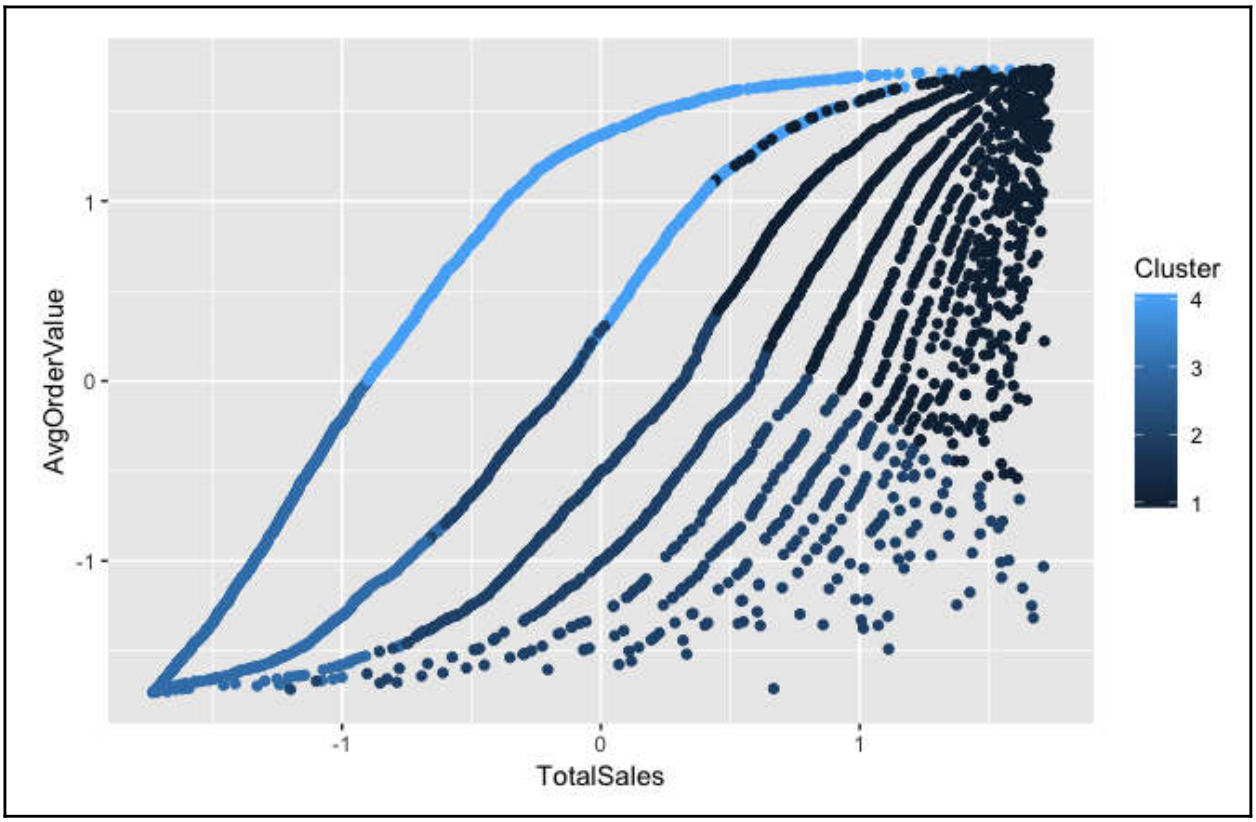
	CustomerID	TotalSales	OrderCount	AvgOrderValue
1	12346	4290.0	1	4298.0
2	12347	3958.0	3473	3885.0
3	12348	3350.0	2862	3299.0
4	12349	3321.0	2	4237.0
5	12350	1241.0	3	2554.0
6	12352	3630.0	3776	2357.0
7	12353	119.0	4	201.0
8	12354	2781.0	5	4148.0
9	12355	1670.0	6	3347.0
10	12356	3724.0	2346	4079.0
11	12357	4111.0	7	4294.0
12	12358	1738.0	8	3445.0
13	12359	4117.0	2863	4224.0
14	12360	3680.0	2347	4055.0
15	12361	607.0	9	1181.0

	CustomerID	TotalSales	OrderCount	AvgOrderValue
1	12346	1.72499932	-1.7314464	1.73144641
2	12347	1.45744512	1.0665903	1.39861543
3	12348	0.96746633	0.5741939	0.92636614
4	12349	0.94409563	-1.7306405	1.68228736
5	12350	-0.73214757	-1.7298346	0.32598095
6	12352	1.19311446	1.3107738	0.16722138
7	12353	-1.63635184	-1.7290287	-1.57026918
8	12354	0.50891711	-1.7282229	1.61056349
9	12355	-0.38642241	-1.7274170	0.96504867
10	12356	1.26886776	0.1583566	1.55495734
11	12357	1.58074570	-1.7266111	1.72822287
12	12358	-0.33162215	-1.7258052	1.04402552
13	12359	1.58558102	0.5749998	1.67181084
14	12360	1.23340877	0.1591625	1.53561607
15	12361	-1.24307940	-1.7249993	-0.78050074

```
> summary(normalizedDF)
   CustomerID       TotalSales.V1        OrderCount.V1        AvgOrderValue.V1
 Min.   :12346   Min.   :-1.7314464   Min.   :-1.7314464   Min.   :-1.7314464
 1st Qu.:13815   1st Qu.:-0.8660254   1st Qu.:-0.8657232   1st Qu.:-0.8657232
 Median :15300   Median : 0.0000000   Median : 0.0000000   Median : 0.0000000
 Mean   :15302   Mean   : 0.0000000   Mean   : 0.0000000   Mean   : 0.0000000
 3rd Qu.:16781   3rd Qu.: 0.8657232   3rd Qu.: 0.8657232   3rd Qu.: 0.8657232
 Max.   :18287   Max.   : 1.7314464   Max.   : 1.7314464   Max.   : 1.7314464
> sapply(normalizedDF, sd)
   CustomerID    TotalSales    OrderCount AvgOrderValue
    1720.983         1.000         1.000         1.000
```


OrderCount
1
0
-1
-1
0
1
TotalSales
Cluster
4
3
2
1


OrderCount
1
0
-1
-1
0
1
AvgOrderValue
Cluster
4
3
2
1



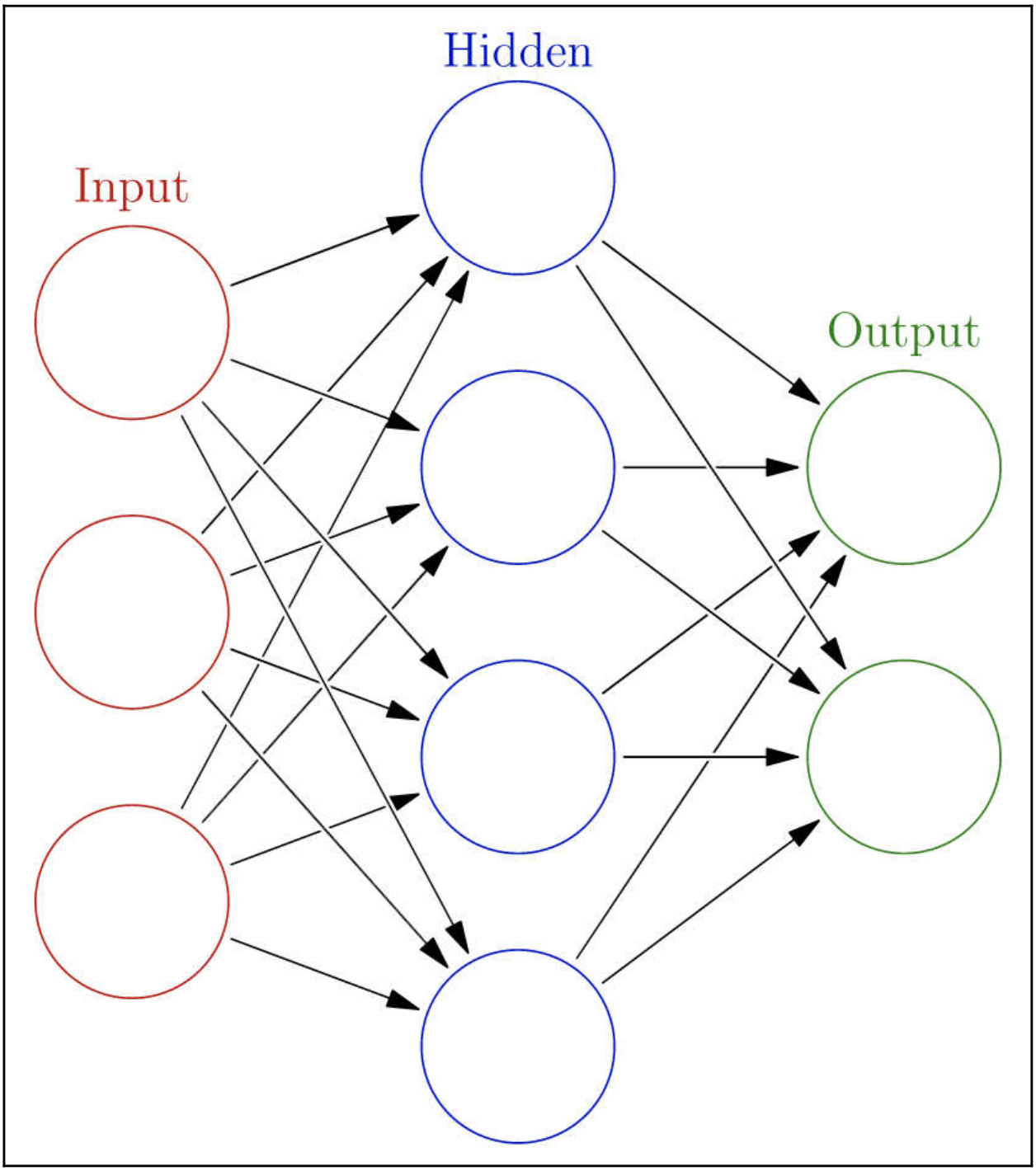

```
[1] "Silhouette Score for 4 Clusters: 0.4117"
[1] "Silhouette Score for 5 Clusters: 0.3831"
[1] "Silhouette Score for 6 Clusters: 0.3778"
[1] "Silhouette Score for 7 Clusters: 0.3915"
[1] "Silhouette Score for 8 Clusters: 0.3716"
```

	CustomerID	TotalSales	OrderCount	AvgOrderValue	Cluster
1	12346	1.72499932	-1.7314464	1.73144641	4
2	12347	1.45744512	1.0665903	1.39861543	1
3	12348	0.96746633	0.5741939	0.92636614	1
4	12349	0.94409563	-1.7306405	1.68228736	4
5	12350	-0.73214757	-1.7298346	0.32598095	4
6	12352	1.19311446	1.3107738	0.16722138	1
7	12353	-1.63635184	-1.7290287	-1.57026918	3
8	12354	0.50891711	-1.7282229	1.61056349	4
9	12355	-0.38642241	-1.7274170	0.96504867	4
10	12356	1.26886776	0.1583566	1.55495734	1
11	12357	1.58074570	-1.7266111	1.72822287	4
12	12358	-0.33162215	-1.7258052	1.04402552	4
13	12359	1.58558102	0.5749998	1.67181084	1
14	12360	1.23340877	0.1591625	1.53561607	1
15	12361	-1.24307940	-1.7249993	-0.78050074	3

```
> # cluster centers
> cluster$centers
  TotalSales OrderCount AvgOrderValue
1  0.2132451  0.7112607    -0.6432146
2 -0.1314293 -0.8520880     0.7984693
3 -1.2460079 -0.7960747    -1.0616594
4  1.2059015  1.0076634     0.8661864
```

```
> df[which(df$CustomerID %in% highValueCustomers),] %>%
+   group_by(Description) %>%
+   summarise(Count=n()) %>%
+   arrange(desc(Count))
# A tibble: 3,659 x 2
   Description                         Count
   <chr>                               <int>
 1 JUMBO BAG RED RETROSPOT              1147
 2 REGENCY CAKESTAND 3 TIER             1086
 3 WHITE HANGING HEART T-LIGHT HOLDER   1079
 4 LUNCH BAG RED RETROSPOT               938
 5 PARTY BUNTING                         869
 6 ASSORTED COLOUR BIRD ORNAMENT         828
 7 SET OF 3 CAKE TINS PANTRY DESIGN      730
 8 LUNCH BAG  BLACK SKULL.               701
 9 POSTAGE                               696
10 PACK OF 72 RETROSPOT CAKE CASES       690
```

Chapter 11: Retaining Customers



```
df.head(10)
```

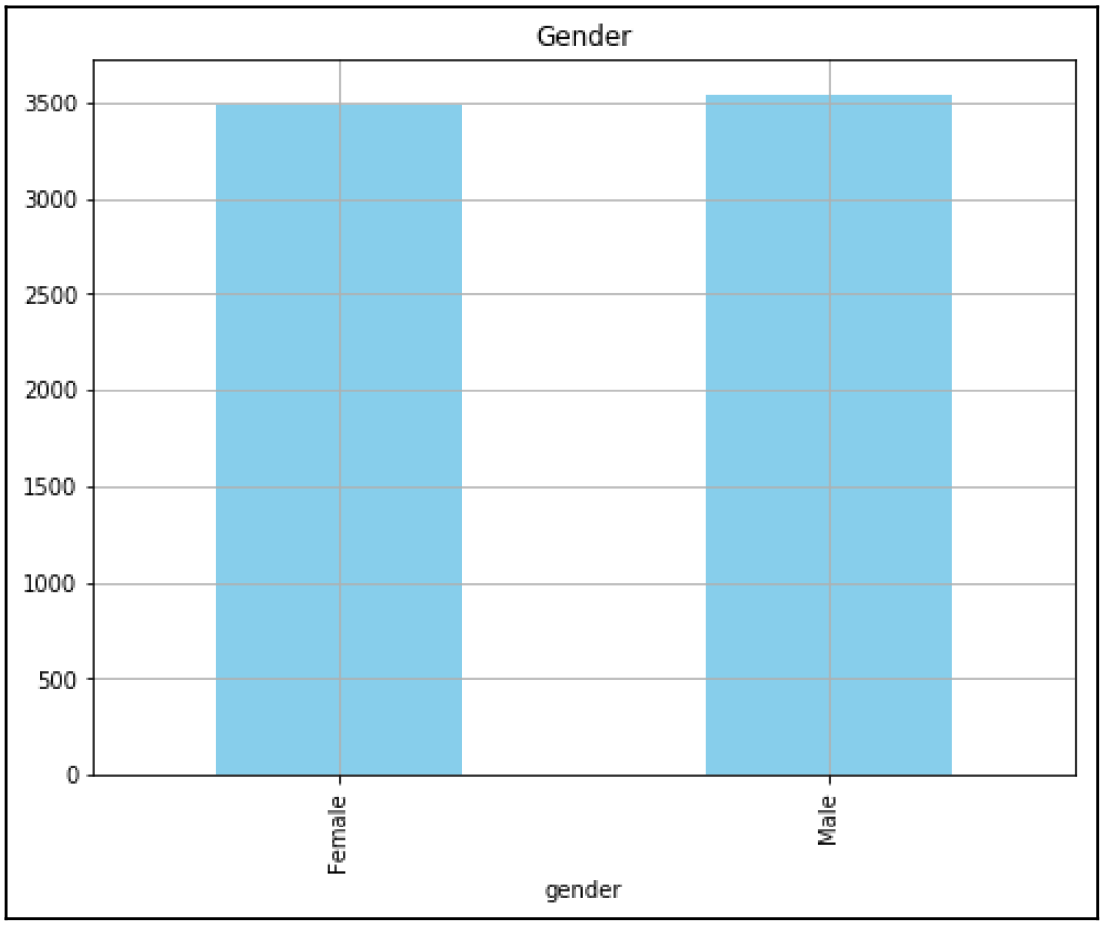
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSup
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	
5	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	...	Yes	
6	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	...	No	
7	6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL	Yes	...	No	
8	7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	...	Yes	
9	6388-TABGU	Male	0	No	Yes	62	Yes	No	DSL	Yes	...	No	

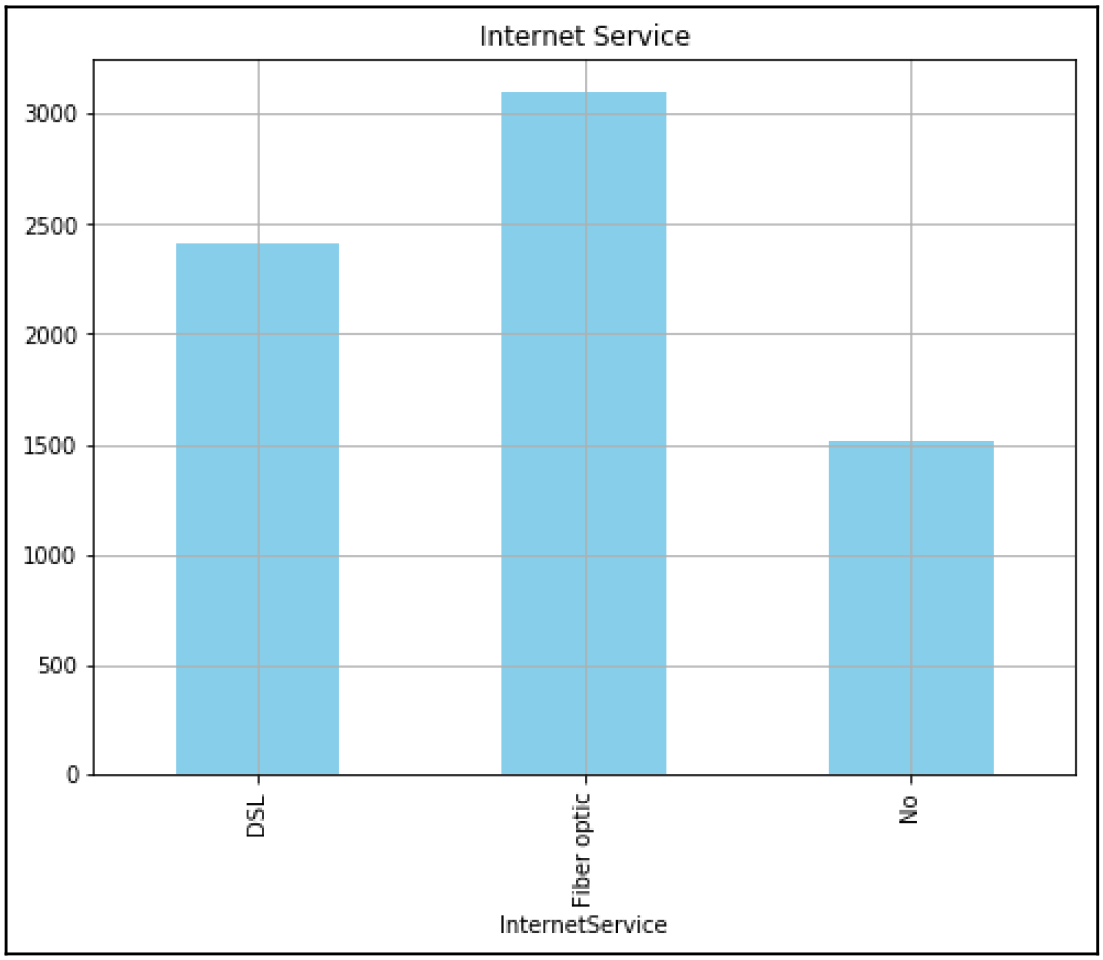
	tenure	MonthlyCharges	TotalCharges
count	7032.000000	7032.000000	7032.000000
mean	32.421786	64.798208	2283.300441
std	24.545260	30.085974	2266.771362
min	1.000000	18.250000	18.800000
25%	9.000000	35.587500	401.450000
50%	29.000000	70.350000	1397.475000
75%	55.000000	89.862500	3794.737500
max	72.000000	118.750000	8684.800000

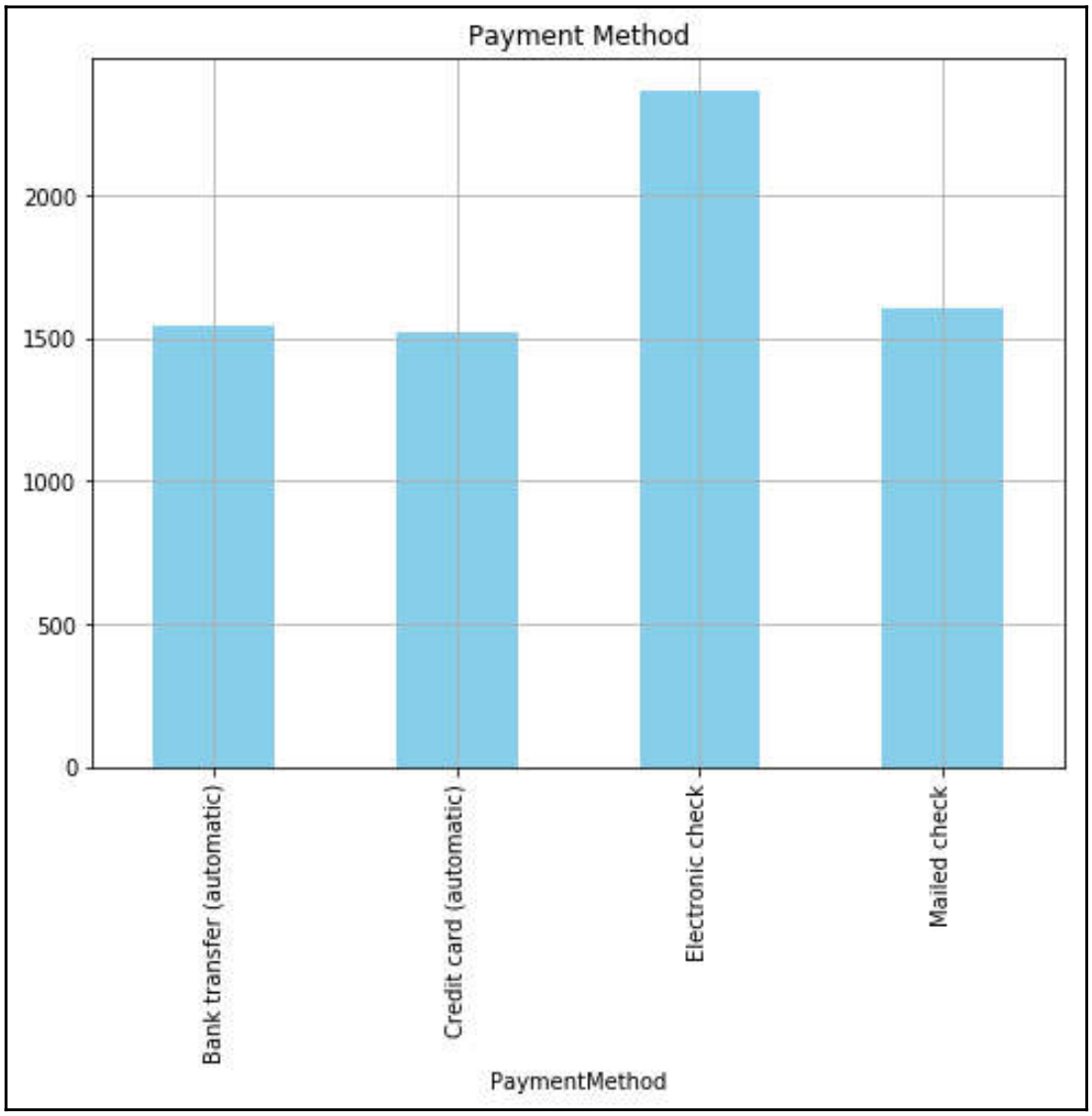
```
df[['tenure', 'MonthlyCharges', 'TotalCharges']].describe()
```

	tenure	MonthlyCharges	TotalCharges
count	7.032000e+03	7.032000e+03	7.032000e+03
mean	-1.028756e-16	4.688495e-14	7.150708e-15
std	1.000000e+00	1.000000e+00	1.000000e+00
min	-1.280157e+00	-1.882268e+00	-2.579056e+00
25%	-9.542285e-01	-7.583727e-01	-6.080585e-01
50%	-1.394072e-01	3.885103e-01	1.950521e-01
75%	9.198605e-01	8.004829e-01	8.382338e-01
max	1.612459e+00	1.269576e+00	1.371323e+00

```
customerID 7032
gender 2
SeniorCitizen 2
Partner 2
Dependents 2
tenure 72
PhoneService 2
MultipleLines 3
InternetService 3
OnlineSecurity 3
OnlineBackup 3
DeviceProtection 3
TechSupport 3
StreamingTV 3
StreamingMovies 3
Contract 3
PaperlessBilling 2
PaymentMethod 4
MonthlyCharges 1584
TotalCharges 6530
Churn 2
```


Gender
3500
3000
2500
2000
1500
1000
500
0
Female
Male
gender


Internet Service
3000
2500
2000
1500
1000
500
0
DSL
Fiber optic
No
InternetService


Payment Method
2000
1500
1000
500
0
Bank transfer (automatic)
Credit card (automatic)
Electronic check
Mailed check
PaymentMethod

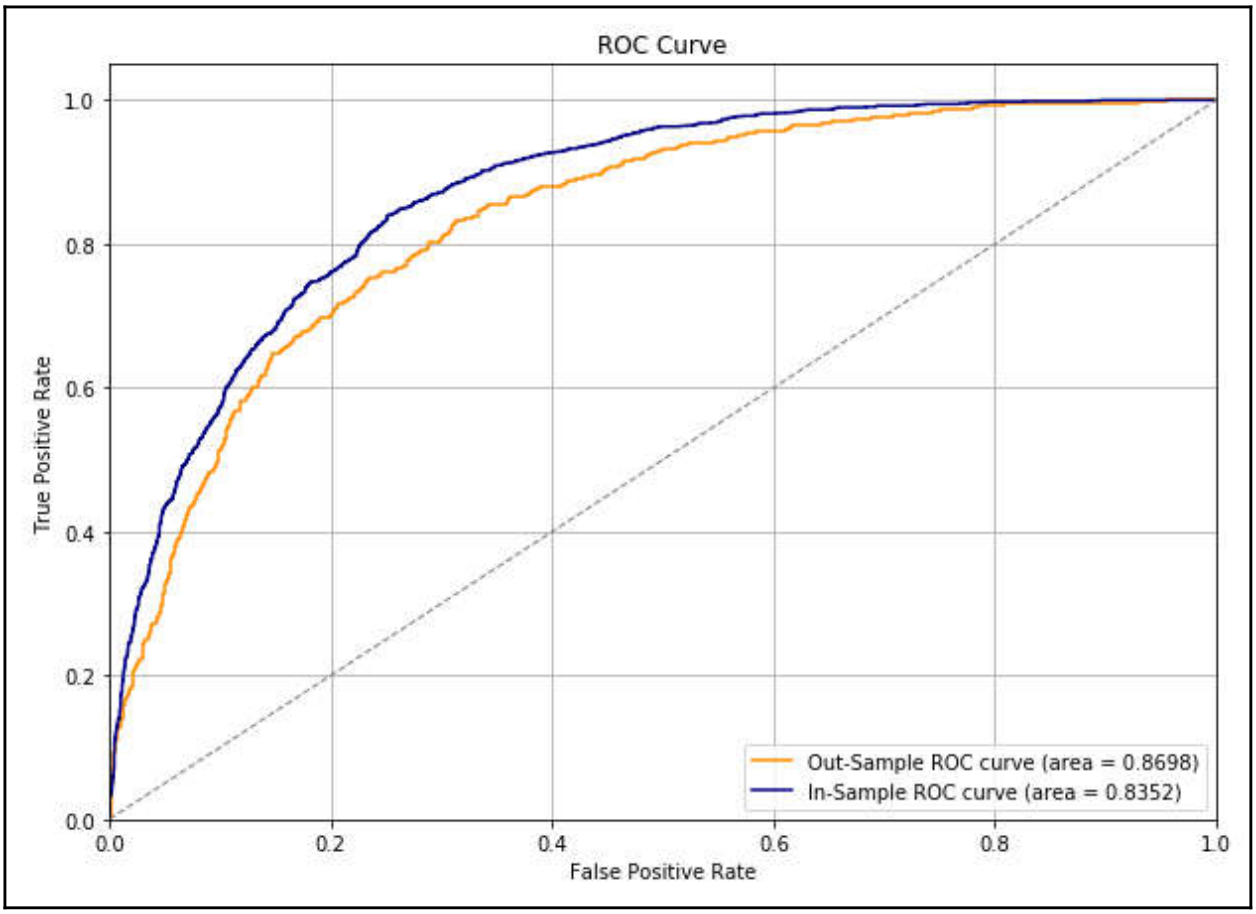
```
sample_set.head(10)
```

	tenure	MonthlyCharges	TotalCharges	Churn	genderFemale	genderMale	SeniorCitizen0	SeniorCitizen1	PartnerNo	PartnerYes	...	StreamingMoviesYes
0	-1.280157	-1.054244	-2.281382	0	1	0	1	0	0	1	...	0
1	0.064298	0.032896	0.389269	0	0	1	1	0	1	0	...	0
2	-1.239416	-0.061298	-1.452520	1	0	1	1	0	1	0	...	0
3	0.512450	-0.467578	0.372439	0	0	1	1	0	1	0	...	0
4	-1.239416	0.396862	-1.234860	1	1	0	1	0	1	0	...	0
5	-0.994970	0.974468	-0.147808	1	1	0	1	0	1	0	...	1
6	-0.424595	0.786142	0.409363	0	0	1	1	0	1	0	...	0
7	-0.913487	-1.059891	-0.791550	0	1	0	1	0	1	0	...	0
8	-0.180148	1.059269	0.696733	1	1	0	1	0	0	1	...	1
9	1.205048	0.009088	0.783956	0	0	1	1	0	1	0	...	0

```
Epoch 1/50
4922/4922 [==============================] - 0s 78us/step - loss: 0.7690 - acc: 0.3779
Epoch 2/50
4922/4922 [==============================] - 0s 14us/step - loss: 0.5783 - acc: 0.7324
Epoch 3/50
4922/4922 [==============================] - 0s 14us/step - loss: 0.4511 - acc: 0.7891
Epoch 4/50
4922/4922 [==============================] - 0s 15us/step - loss: 0.4181 - acc: 0.8046
Epoch 5/50
4922/4922 [==============================] - 0s 15us/step - loss: 0.4093 - acc: 0.8094
Epoch 6/50
4922/4922 [==============================] - 0s 18us/step - loss: 0.4058 - acc: 0.8137
Epoch 7/50
4922/4922 [==============================] - 0s 20us/step - loss: 0.4048 - acc: 0.8070
Epoch 8/50
4922/4922 [==============================] - 0s 16us/step - loss: 0.4027 - acc: 0.8106
Epoch 9/50
4922/4922 [==============================] - 0s 15us/step - loss: 0.4007 - acc: 0.8086
Epoch 10/50
4922/4922 [==============================] - 0s 13us/step - loss: 0.3994 - acc: 0.8094
Epoch 11/50
4922/4922 [==============================] - 0s 13us/step - loss: 0.3984 - acc: 0.8111
Epoch 12/50
4922/4922 [==============================] - 0s 17us/step - loss: 0.3983 - acc: 0.8098
Epoch 13/50
4922/4922 [==============================] - 0s 15us/step - loss: 0.3982 - acc: 0.8133
Epoch 14/50
4922/4922 [==============================] - 0s 13us/step - loss: 0.3972 - acc: 0.8123
Epoch 15/50
4922/4922 [==============================] - 0s 13us/step - loss: 0.3960 - acc: 0.8113
```

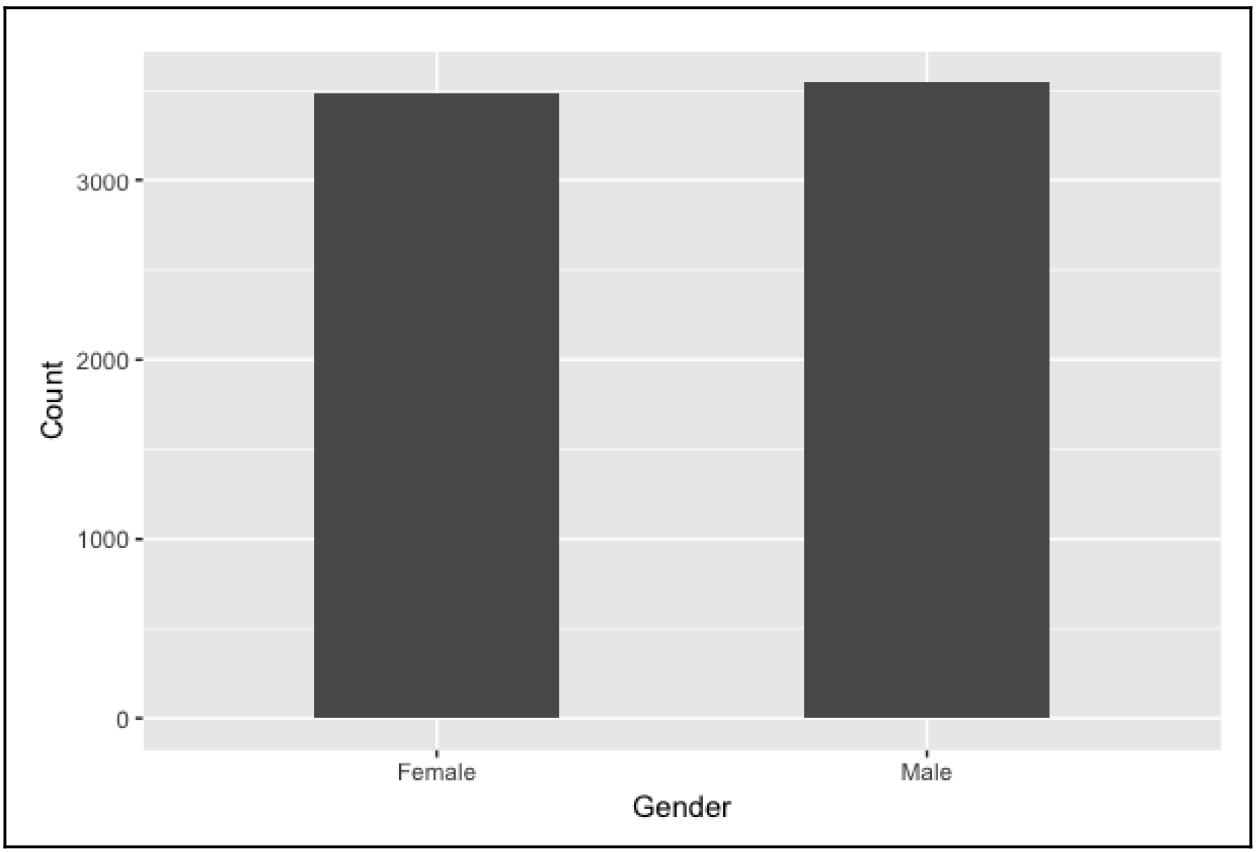
```
In-Sample Accuracy: 0.8151
Out-of-Sample Accuracy: 0.7910


In-Sample Precision: 0.6733
Out-of-Sample Precision: 0.6638


In-Sample Recall: 0.5583
Out-of-Sample Recall: 0.5169
```

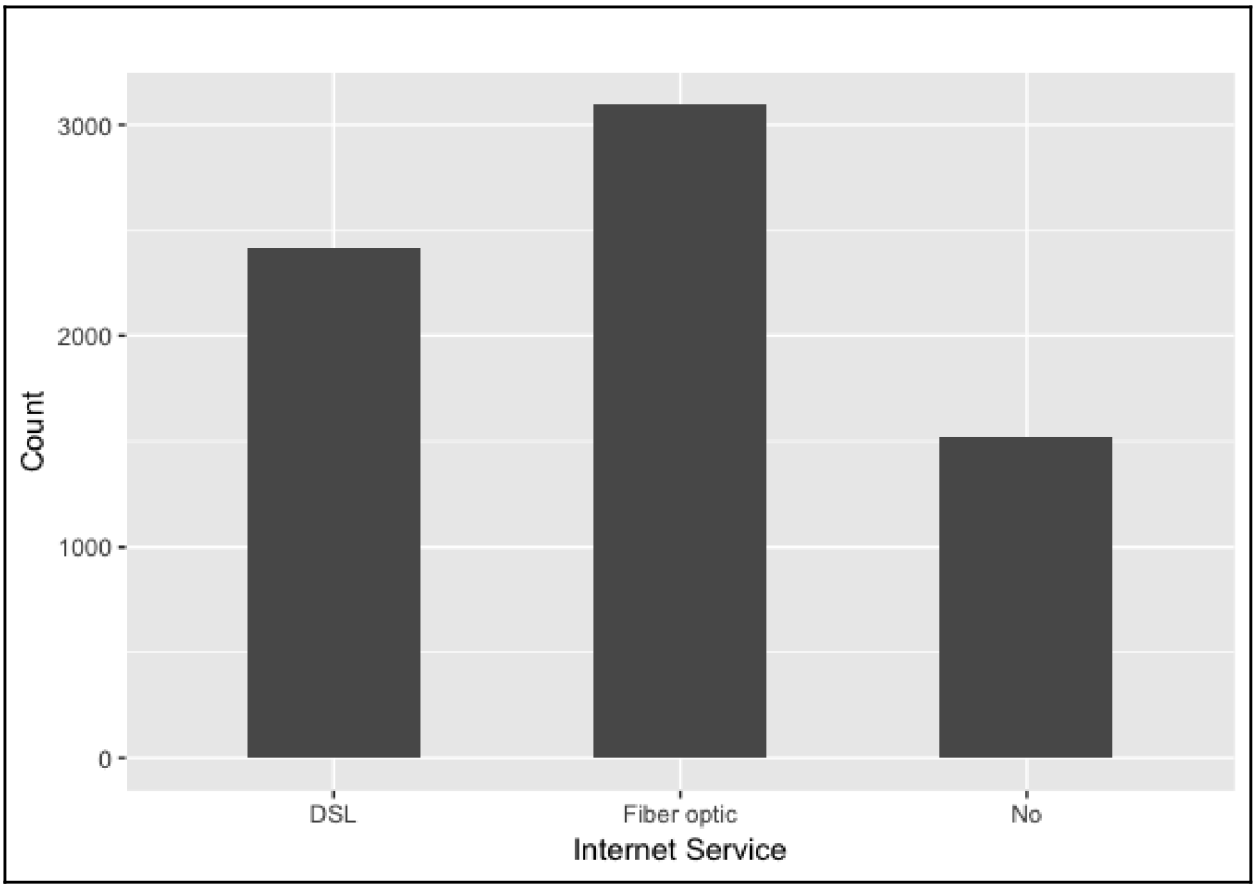
```
In-Sample AUC: 0.8698
Out-Sample AUC: 0.8352
```

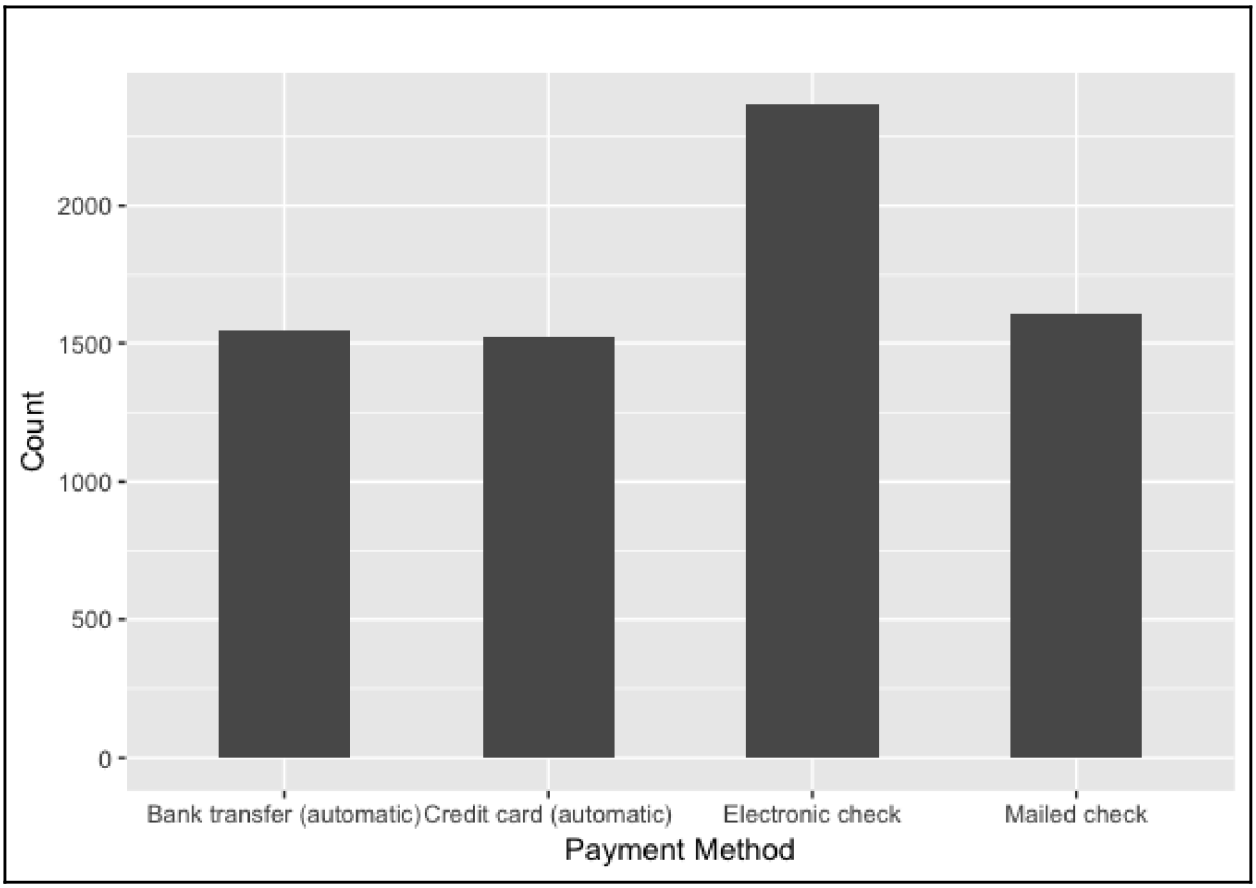


	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity
1	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No
2	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes
3	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes
4	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes
5	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No
6	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No
7	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No
8	6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL	Yes
9	7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No
10	6388-TABGU	Male	0	No	Yes	62	Yes	No	DSL	Yes
11	9763-GRSKD	Male	0	Yes	Yes	13	Yes	No	DSL	Yes
12	7469-LKBCI	Male	0	No	No	16	Yes	No	No	No internet service
13	8091-TTVAX	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No
14	0280-XJGEX	Male	0	No	No	49	Yes	Yes	Fiber optic	No
15	5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic	Yes

```
> apply(df, 2, function(x) length(unique(x)))
      customerID           gender    SeniorCitizen          Partner       Dependents
            7032                2                2                2                2
          tenure     PhoneService    MultipleLines  InternetService   OnlineSecurity
              72                2                3                3                3
    OnlineBackup DeviceProtection      TechSupport      StreamingTV  StreamingMovies
               3                3                3                3                3
        Contract PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges
               3                2                4             1584             6530
           Churn
               2
```




3000
2000
1000
0
Count
DSL
Fiber optic
No
Internet Service



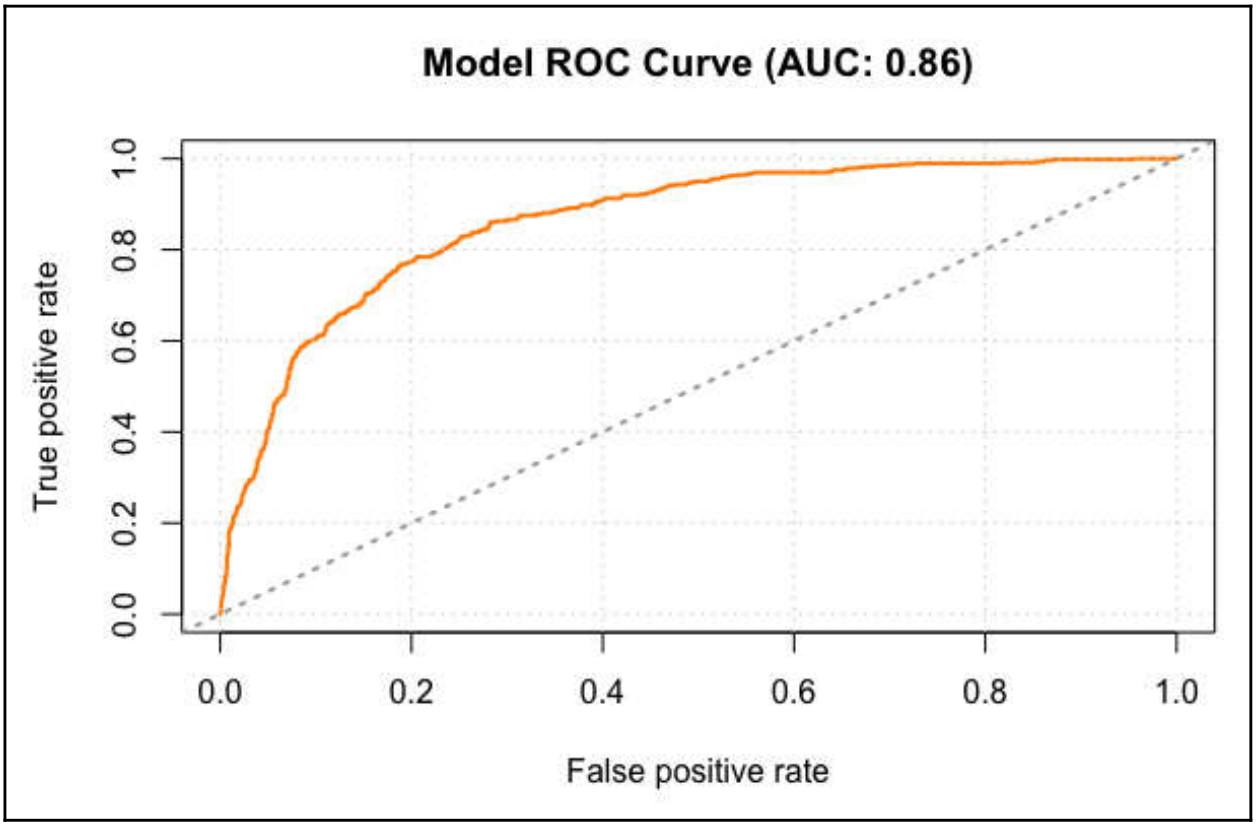

```
> summary(sampleDF[,c("tenure", "MonthlyCharges", "TotalCharges")])
     tenure         MonthlyCharges      TotalCharges
 Min.   :-1.2802   Min.   :-1.8823   Min.   :-2.5791
 1st Qu.:-0.9542   1st Qu.:-0.7584   1st Qu.:-0.6081
 Median :-0.1394   Median : 0.3885   Median : 0.1951
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.9199   3rd Qu.: 0.8005   3rd Qu.: 0.8382
 Max.   : 1.6125   Max.   : 1.2696   Max.   : 1.3713
> apply(sampleDF[,c("tenure", "MonthlyCharges", "TotalCharges")], 2, sd)
       tenure MonthlyCharges   TotalCharges
            1              1              1
```

```
> summary(df[,c("tenure", "MonthlyCharges", "TotalCharges")])
     tenure       MonthlyCharges    TotalCharges
 Min.   : 1.00   Min.   : 18.25   Min.   :  18.8
 1st Qu.: 9.00   1st Qu.: 35.59   1st Qu.: 401.4
 Median :29.00   Median : 70.35   Median :1397.5
 Mean   :32.42   Mean   : 64.80   Mean   :2283.3
 3rd Qu.:55.00   3rd Qu.: 89.86   3rd Qu.:3794.7
 Max.   :72.00   Max.   :118.75   Max.   :8684.8
> apply(df[,c("tenure", "MonthlyCharges", "TotalCharges")], 2, sd)
        tenure MonthlyCharges    TotalCharges
      24.54526       30.08597     2266.77136
```

	tenure	MonthlyCharges	TotalCharges	gender	Partner	Dependents	PhoneService	PaperlessBilling	Churn	MultipleLines.No	MultipleLines.No phone service	MultipleLines.Yes	InternetService.DSL
1	-1.28015700	-1.054243906	-2.281382138	0	1	0	0	1	0	0	1	0	1
2	0.06429811	0.032896403	0.389269024	1	0	0	1	0	0	1	0	0	1
3	-1.23941594	-0.061298166	-1.452520489	1	0	0	1	1	1	1	0	0	1
4	0.51244982	-0.467578124	0.372439102	1	0	0	0	0	0	0	1	0	1
5	-1.23941594	0.396862254	-1.234860074	0	0	0	1	1	1	1	0	0	0
6	-0.99496955	0.974467637	-0.147808199	0	0	0	1	1	1	0	0	1	0
7	-0.42459466	0.786142215	0.409363467	1	0	1	1	1	0	0	0	1	0
8	-0.91348743	-1.059891256	-0.791549959	0	0	0	0	0	0	0	1	0	1
9	-0.18014827	1.059268949	0.696733168	0	1	0	1	1	1	0	0	1	0
10	1.20504791	0.009088278	0.783955768	1	0	1	1	0	0	1	0	0	1
11	-0.79126423	-0.187819078	-0.362936119	1	1	1	1	1	0	1	0	0	1
12	-0.66904104	-1.818925526	-0.740522515	1	0	0	1	0	0	1	0	0	0
13	1.04208365	0.986248047	1.098049907	1	1	0	1	0	0	0	0	1	0
14	0.67541407	1.041511473	1.020482172	1	0	0	1	1	1	0	0	1	0
15	-0.30237146	1.070472377	0.615752488	1	0	0	1	1	0	1	0	0	0

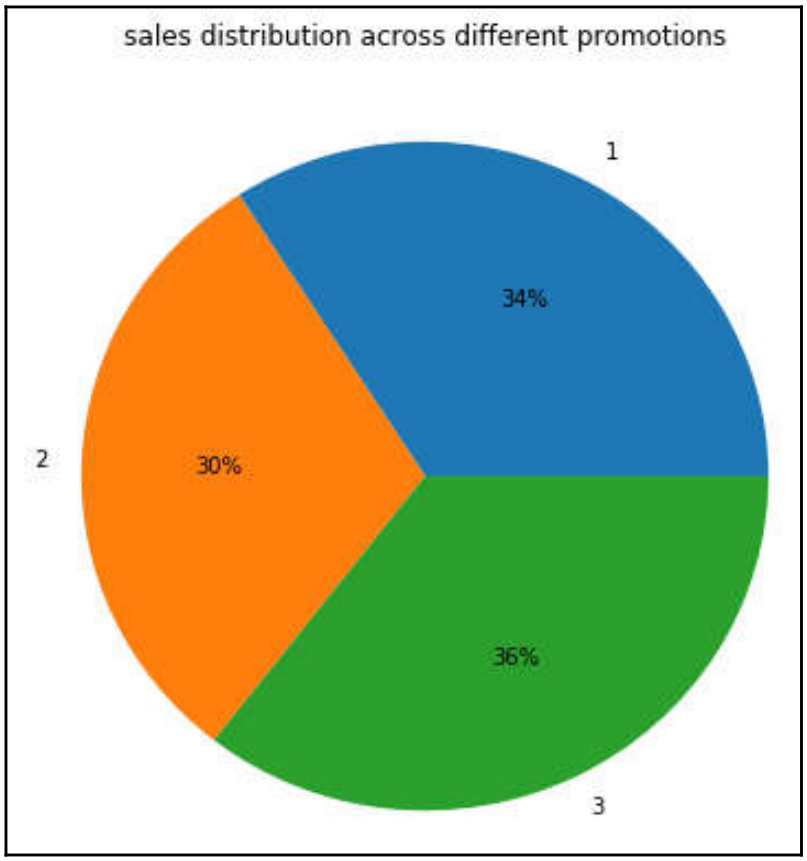
```
Train on 3937 samples, validate on 985 samples
Epoch 1/50
3937/3937 [==============================] - 1s 202us/step - loss: 0.6851 - acc: 0.7290 - val_loss: 0.6650 - val_acc: 0.7391
Epoch 2/50
3937/3937 [==============================] - 0s 62us/step - loss: 0.6026 - acc: 0.7330 - val_loss: 0.5225 - val_acc: 0.7391
Epoch 3/50
3937/3937 [==============================] - 0s 55us/step - loss: 0.4869 - acc: 0.7330 - val_loss: 0.4533 - val_acc: 0.7391
Epoch 4/50
3937/3937 [==============================] - 0s 52us/step - loss: 0.4520 - acc: 0.7330 - val_loss: 0.4428 - val_acc: 0.7391
Epoch 5/50
3937/3937 [==============================] - 0s 48us/step - loss: 0.4436 - acc: 0.7711 - val_loss: 0.4384 - val_acc: 0.7980
Epoch 6/50
3937/3937 [==============================] - 0s 48us/step - loss: 0.4384 - acc: 0.7917 - val_loss: 0.4359 - val_acc: 0.7929
Epoch 7/50
3937/3937 [==============================] - 0s 49us/step - loss: 0.4350 - acc: 0.7932 - val_loss: 0.4318 - val_acc: 0.7949
Epoch 8/50
3937/3937 [==============================] - 0s 54us/step - loss: 0.4314 - acc: 0.7945 - val_loss: 0.4296 - val_acc: 0.7939
Epoch 9/50
3937/3937 [==============================] - 0s 50us/step - loss: 0.4298 - acc: 0.7971 - val_loss: 0.4272 - val_acc: 0.7919
Epoch 10/50
3937/3937 [==============================] - 0s 46us/step - loss: 0.4278 - acc: 0.7988 - val_loss: 0.4255 - val_acc: 0.7939
```

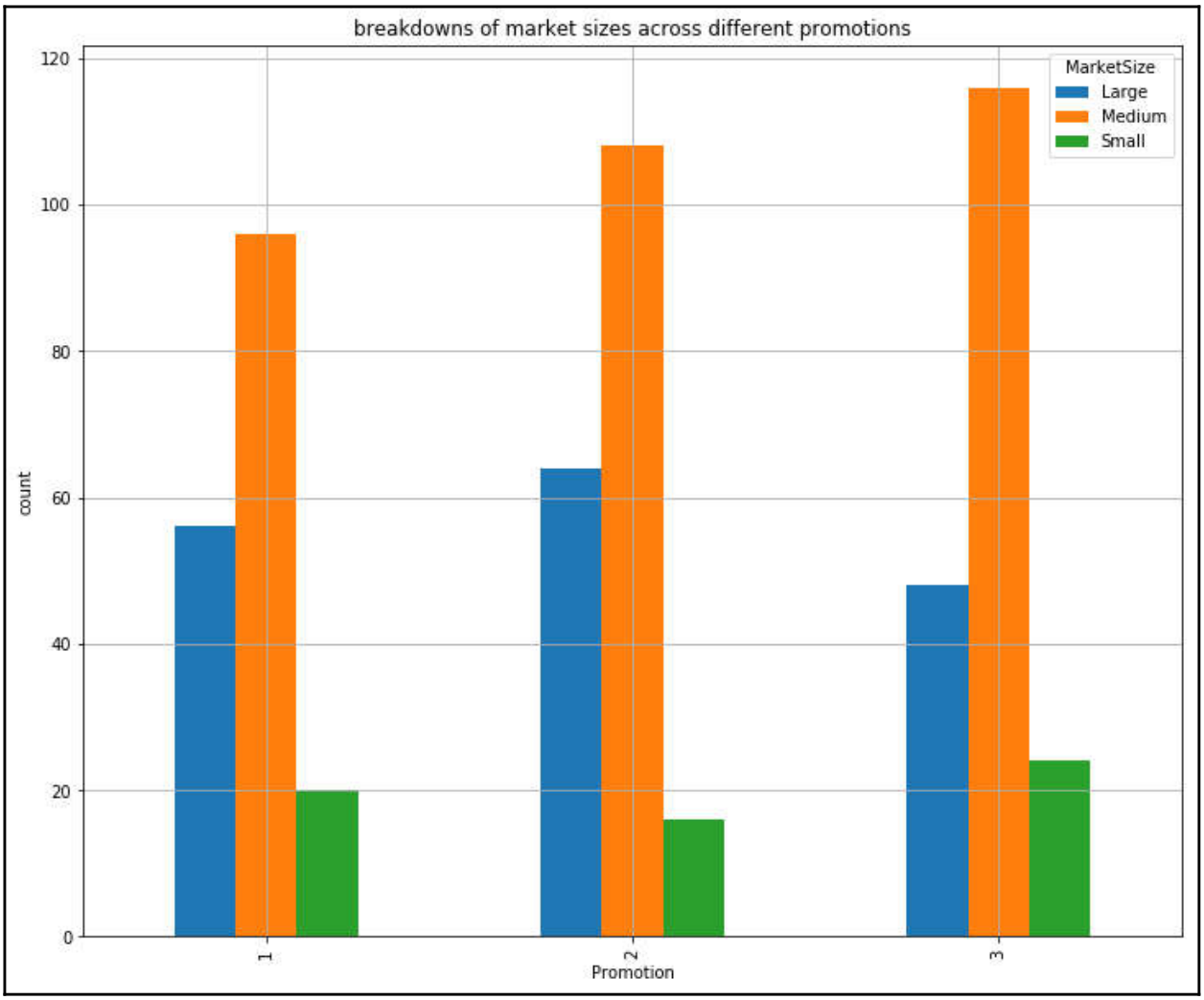
```
> print(sprintf('In-Sample Accuracy: %0.4f', inSampleAccuracy))
[1] "In-Sample Accuracy: 0.8035"
> print(sprintf('Out-Sample Accuracy: %0.4f', outSampleAccuracy))
[1] "Out-Sample Accuracy: 0.8275"
> print(sprintf('In-Sample Precision: %0.4f', inSamplePrecision))
[1] "In-Sample Precision: 0.6638"
> print(sprintf('Out-Sample Precision: %0.4f', outSamplePrecision))
[1] "Out-Sample Precision: 0.7165"
> print(sprintf('In-Sample Recall: %0.4f', inSampleRecall))
[1] "In-Sample Recall: 0.5283"
> print(sprintf('Out-Sample Recall: %0.4f', outSampleRecall))
[1] "Out-Sample Recall: 0.5811"
```

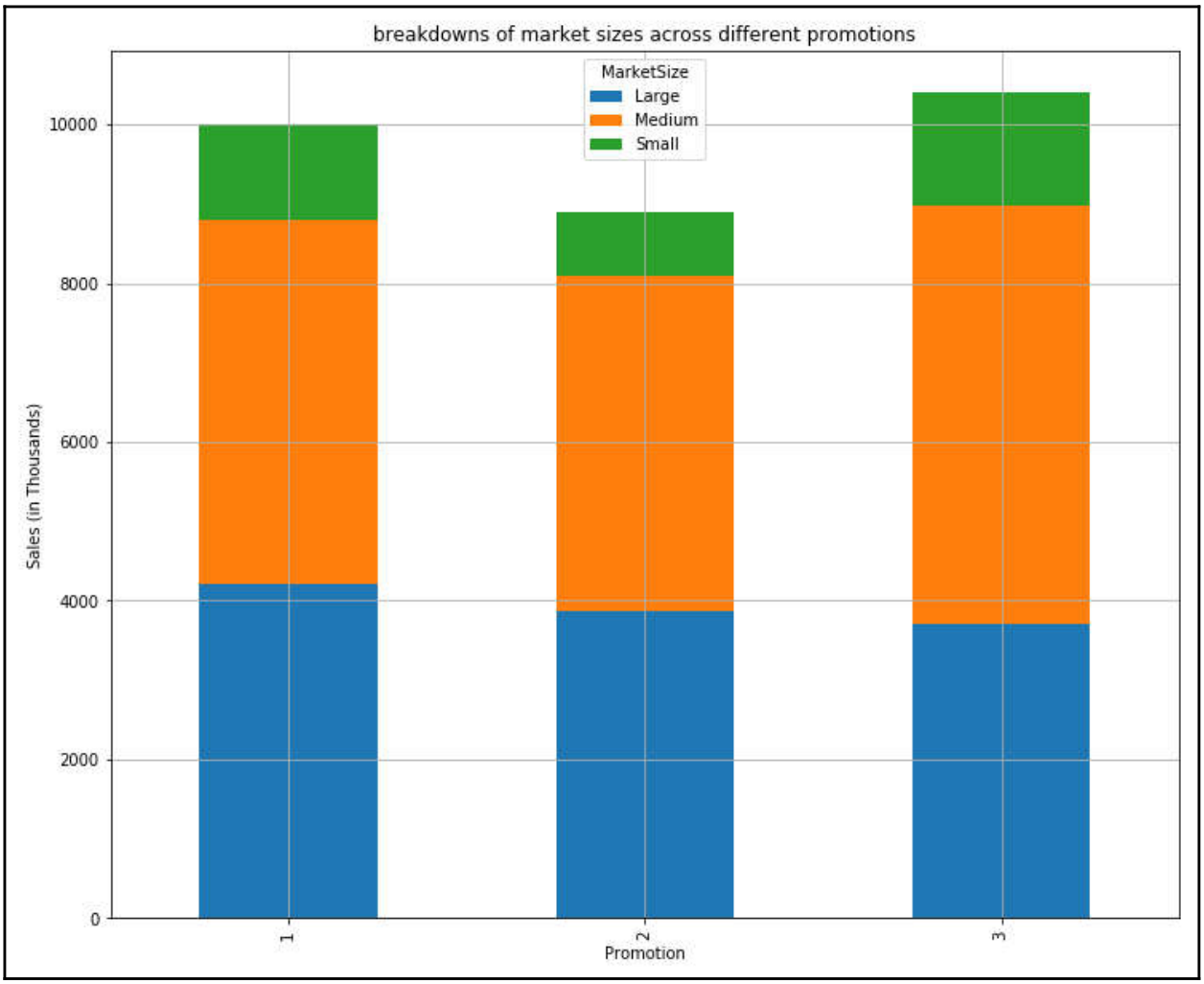


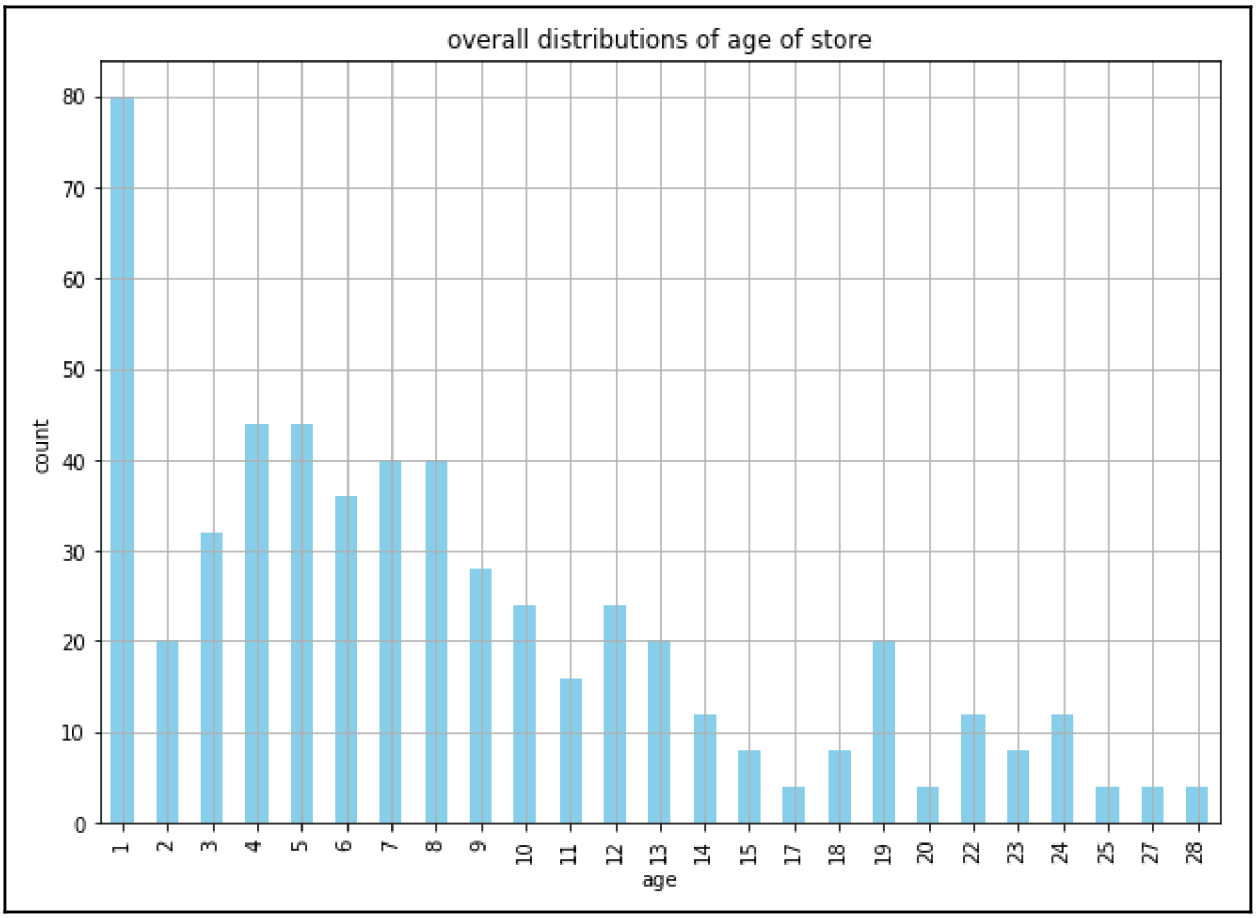
Chapter 12: A/B Testing for Better Marketing Strategy

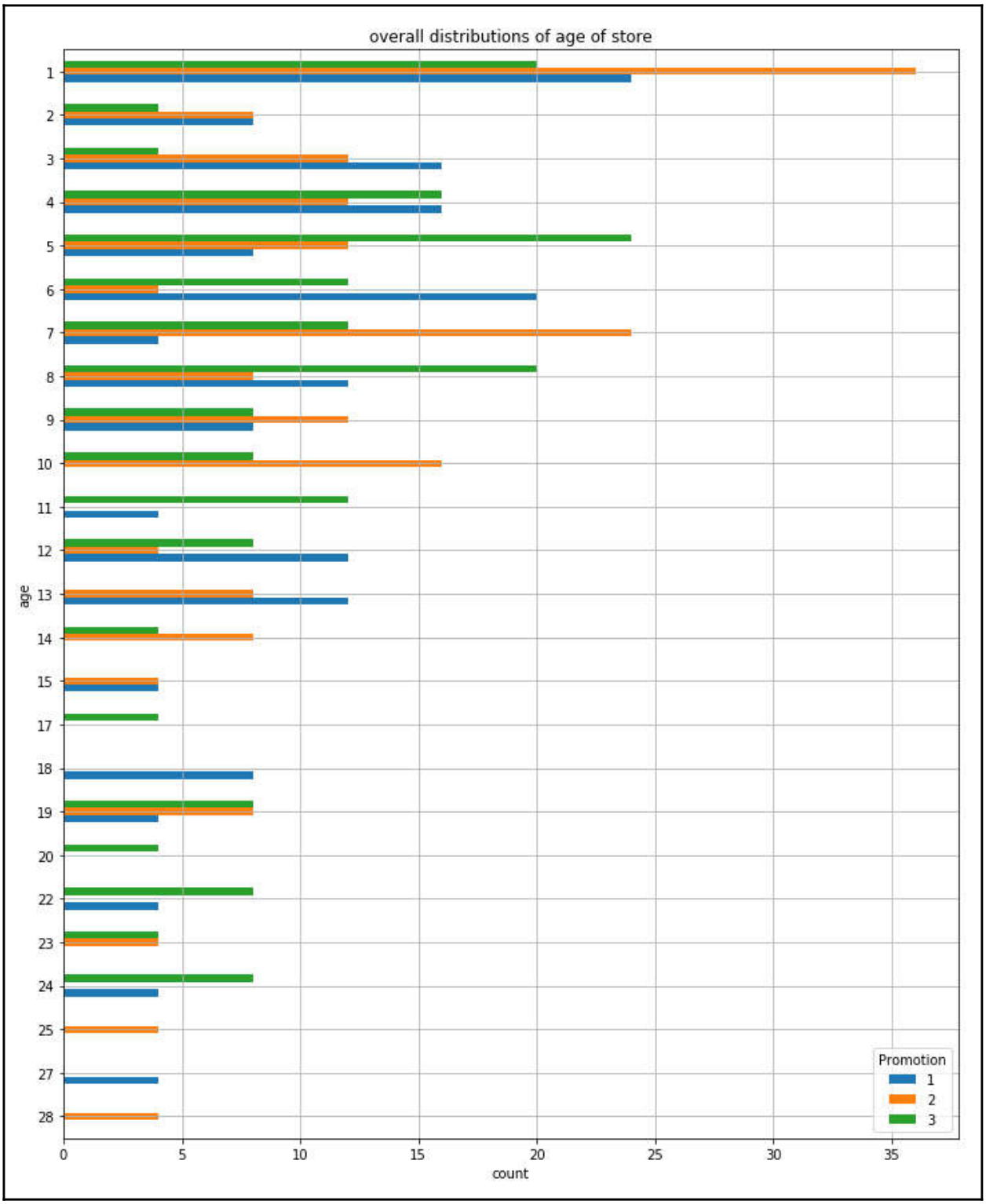
	MarketID	MarketSize	LocationID	AgeOfStore	Promotion	Week	SalesInThousands
0	1	Medium	1	4	3	1	33.73
1	1	Medium	1	4	3	2	35.67
2	1	Medium	1	4	3	3	29.03
3	1	Medium	1	4	3	4	39.25
4	1	Medium	2	5	2	1	27.81
5	1	Medium	2	5	2	2	34.67
6	1	Medium	2	5	2	3	27.98
7	1	Medium	2	5	2	4	27.72
8	1	Medium	3	12	1	1	44.54
9	1	Medium	3	12	1	2	37.94
10	1	Medium	3	12	1	3	45.49
11	1	Medium	3	12	1	4	34.75
12	1	Medium	4	1	2	1	39.28
13	1	Medium	4	1	2	2	39.80
14	1	Medium	4	1	2	3	24.77


sales distribution across different promotions
1
34%
2
30%
36%
3


breakdowns of market sizes across different promotions
MarketSize
Large
Medium
Small
count
120
100
80
60
40
20
0
1
2
3
Promotion

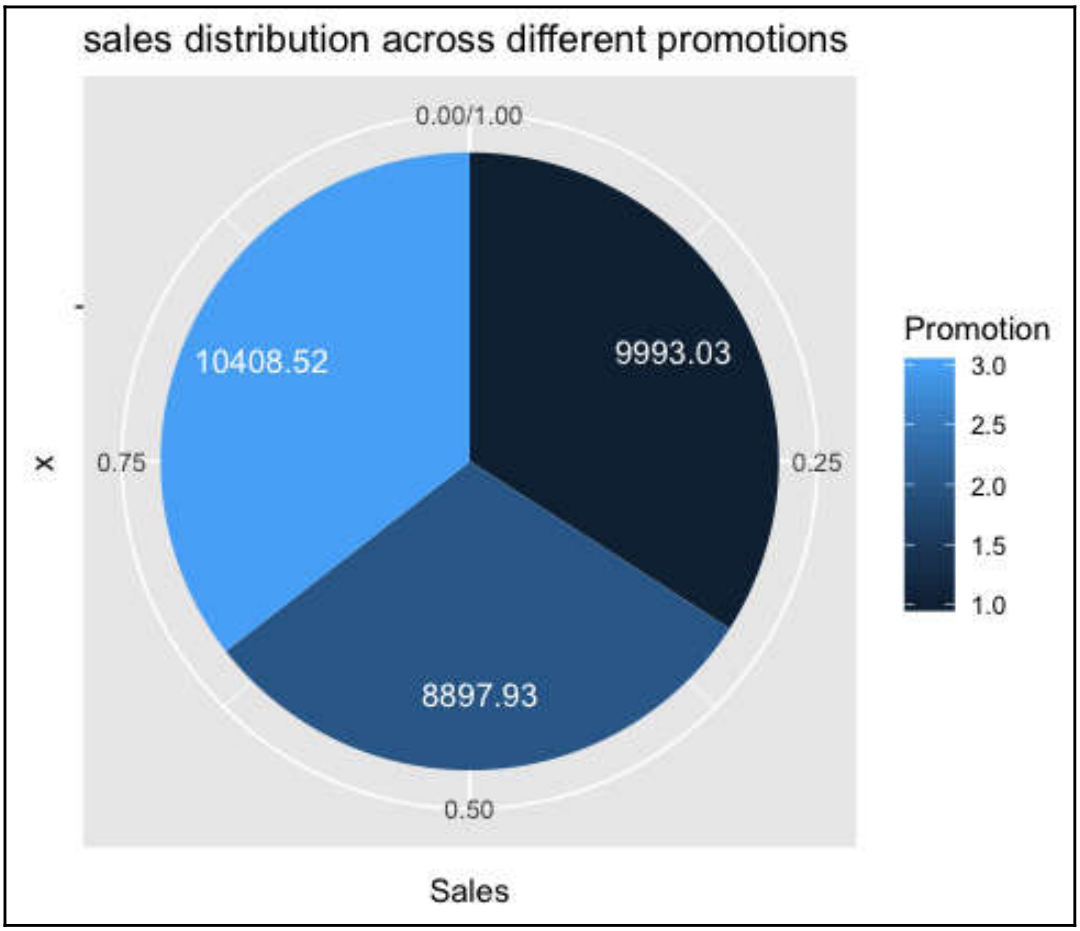

breakdowns of market sizes across different promotions
MarketSize
Large
Medium
Small
10000
8000
6000
4000
2000
0
Sales (in Thousands)
1
2
3
Promotion

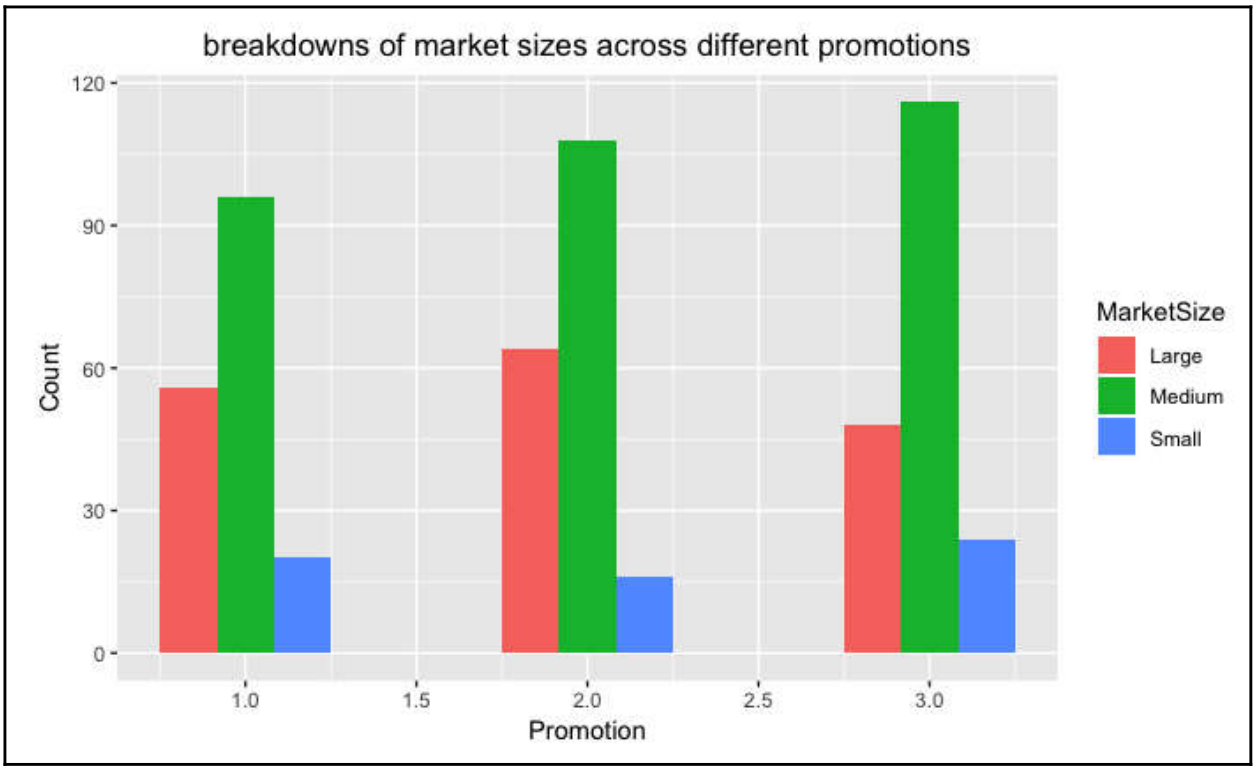

overall distributions of age of store
count
age
0
10
20
30
40
50
60
70
80
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
17
18
19
20
22
23
24
25
27
28

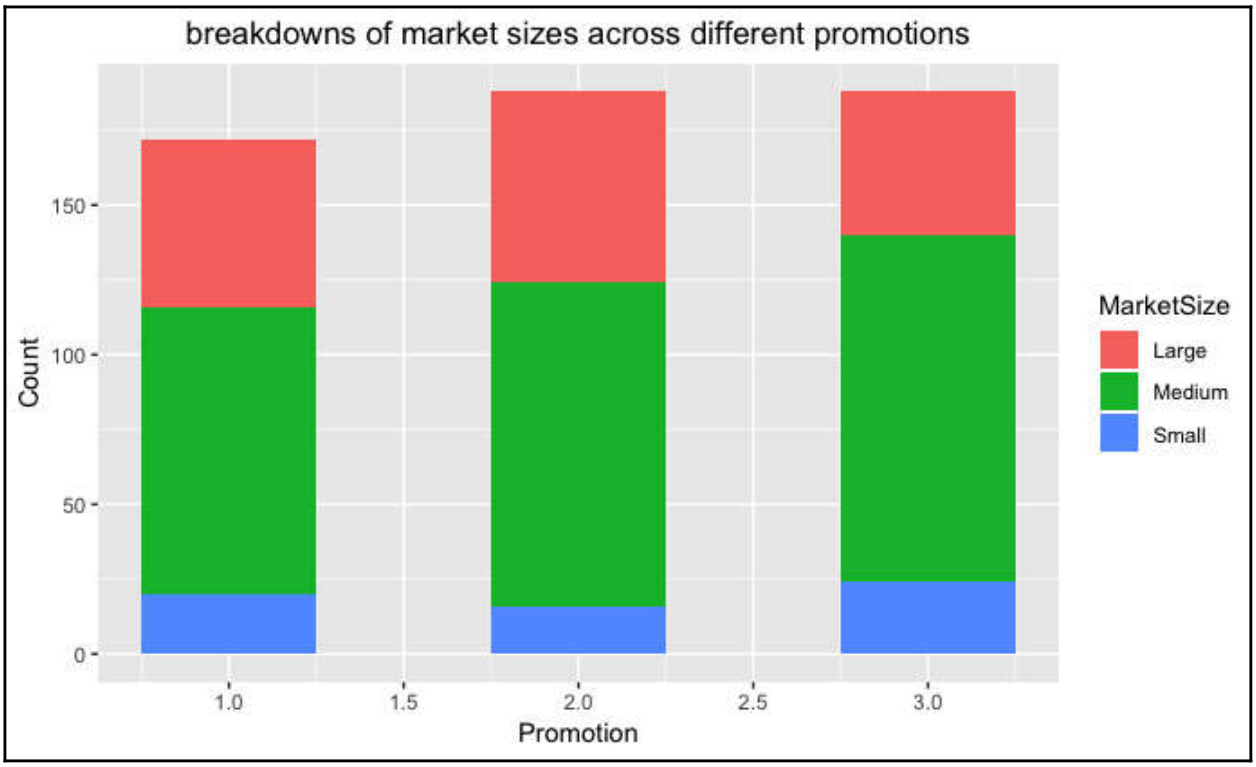

overall distributions of age of store
age
count
Promotion
1
2
3
0
5
10
15
20
25
30
35
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
17
18
19
20
22
23
24
25
27
28

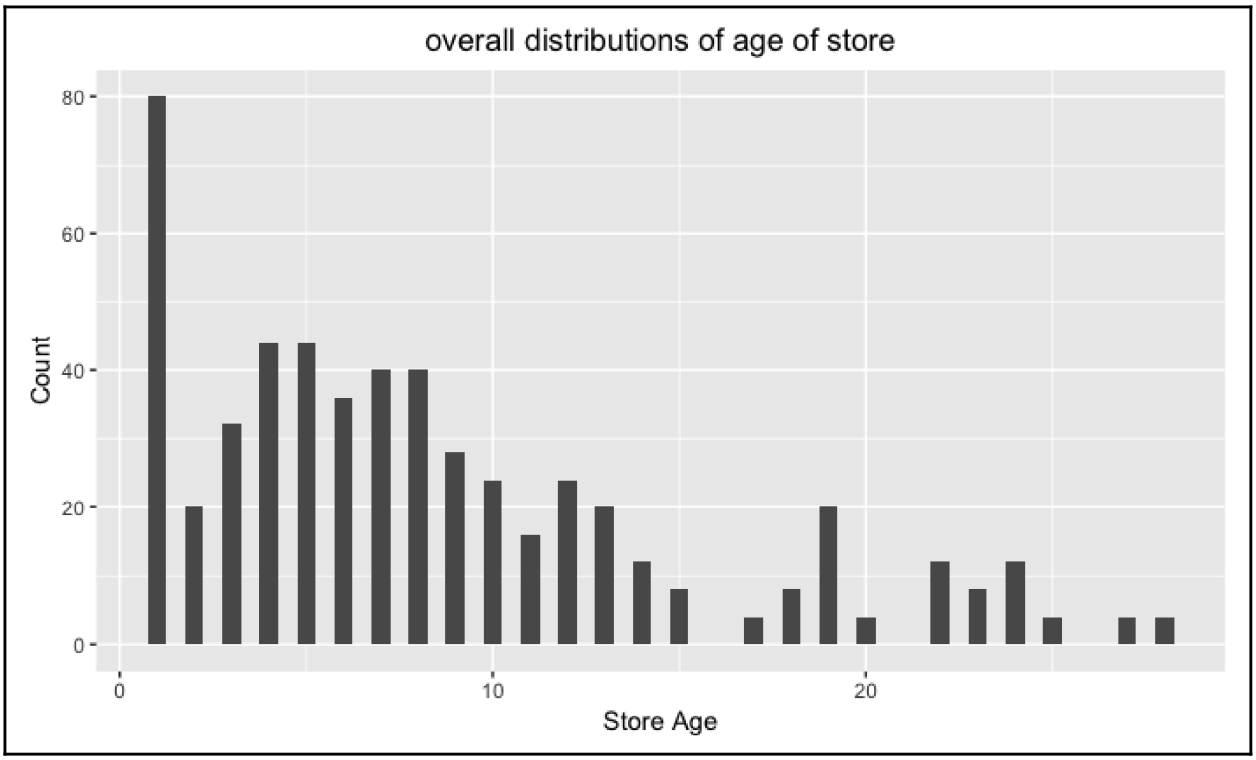
Promotion	count	mean	std	min	25%	50%	75%	max
1	172.0	8.279070	6.636160	1.0	3.0	6.0	12.0	27.0
2	188.0	7.978723	6.597648	1.0	3.0	7.0	10.0	28.0
3	188.0	9.234043	6.651646	1.0	5.0	8.0	12.0	24.0

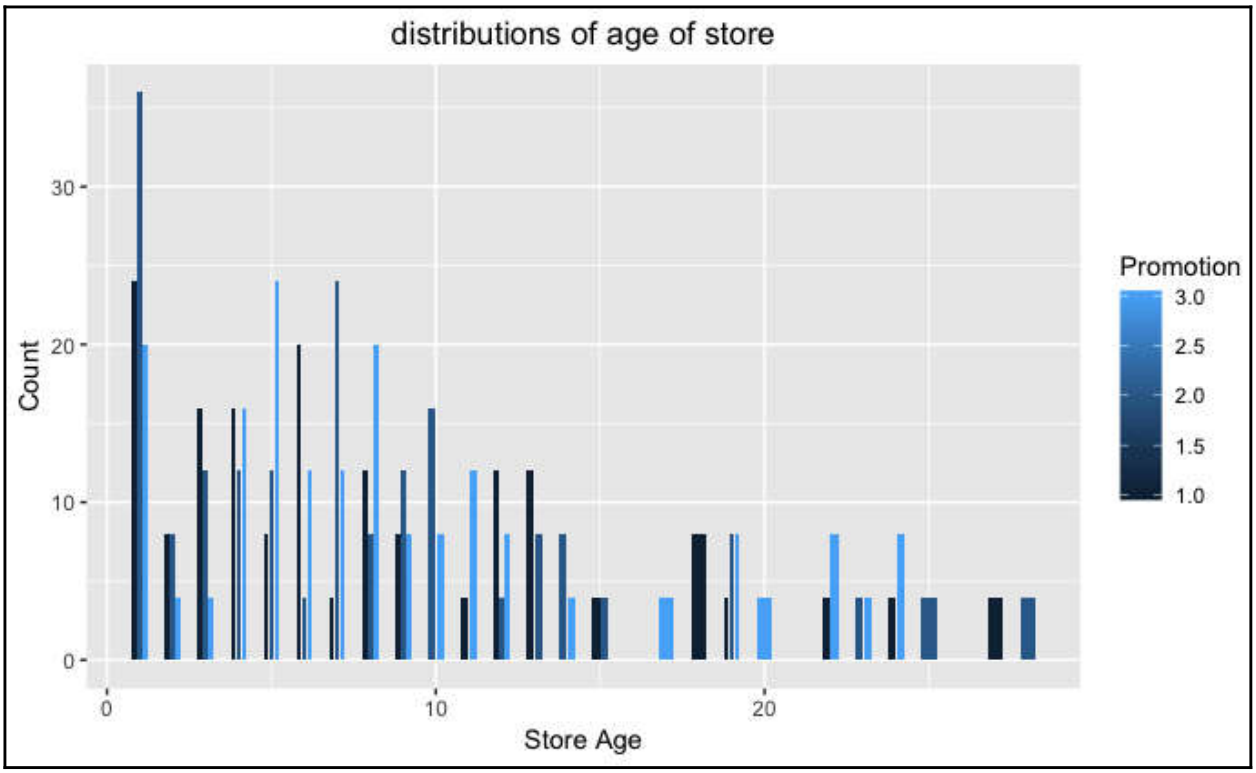
	MarketID	MarketSize	LocationID	AgeOfStore	Promotion	Week	SalesInThousands
1	1	Medium	1	4	3	1	33.73
2	1	Medium	1	4	3	2	35.67
3	1	Medium	1	4	3	3	29.03
4	1	Medium	1	4	3	4	39.25
5	1	Medium	2	5	2	1	27.81
6	1	Medium	2	5	2	2	34.67
7	1	Medium	2	5	2	3	27.98
8	1	Medium	2	5	2	4	27.72
9	1	Medium	3	12	1	1	44.54
10	1	Medium	3	12	1	2	37.94
11	1	Medium	3	12	1	3	45.49
12	1	Medium	3	12	1	4	34.75
13	1	Medium	4	1	2	1	39.28
14	1	Medium	4	1	2	2	39.80
15	1	Medium	4	1	2	3	24.77


sales distribution across different promotions
0.00/1.00
0.25
0.50
0.75
10408.52
9993.03
8897.93
Promotion
3.0
2.5
2.0
1.5
1.0
Sales


breakdowns of market sizes across different promotions
120
90
60
30
0
Count
1.0
1.5
2.0
2.5
3.0
Promotion
MarketSize
Large
Medium
Small


breakdowns of market sizes across different promotions
Count
150
100
50
0
1.0
1.5
2.0
2.5
3.0
Promotion
MarketSize
Large
Medium
Small


overall distributions of age of store
Count
80
60
40
20
0
0
10
20
Store Age




```
> tapply(df$AgeOfStore, df$Promotion, summary)
$`1`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   6.000   8.279  12.000  27.000

$`2`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   7.000   7.979  10.000  28.000

$`3`
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   5.000   8.000   9.234  12.000  24.000
```

Chapter 13: What's Next?