Chapter 01: Transitioning from Data Developer to Data Scientist

	New Data	New Data Source
Data Developer	☑	☑
Data Scientist		☑

	Data Quality
Data Developer	☑
Data Scientist	☑

	Query	Mining
Data Developer	☑	
Data Scientist	☑	☑

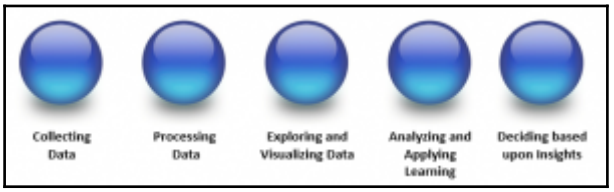
	Performance
Data Developer	☑
Data Scientist	☑

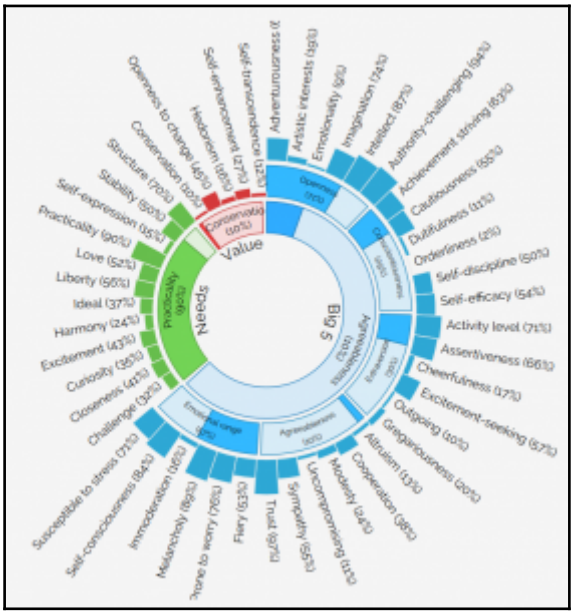
	Financial Reporting
Data Developer	☑
Data Scientist	

	Visualization
Data Developer	☑
Data Scientist	☑

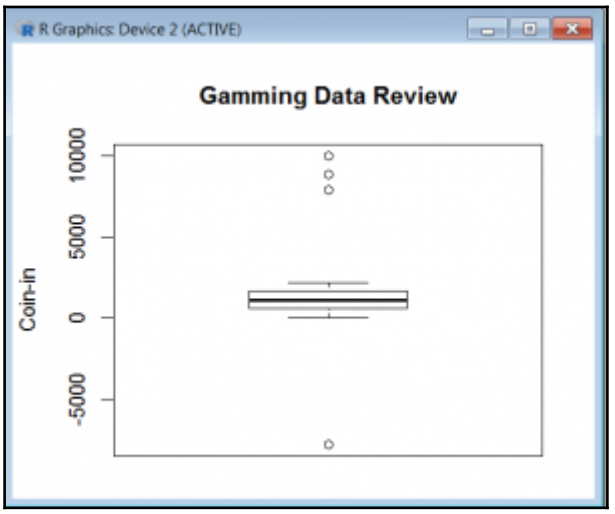
	Tools of the Trade
Data Developer	☑
Data Scientist	☑

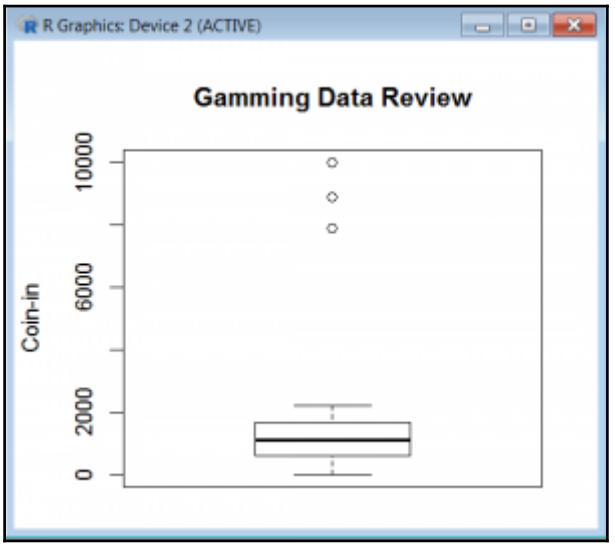
Chapter 02: Declaring the Objectives

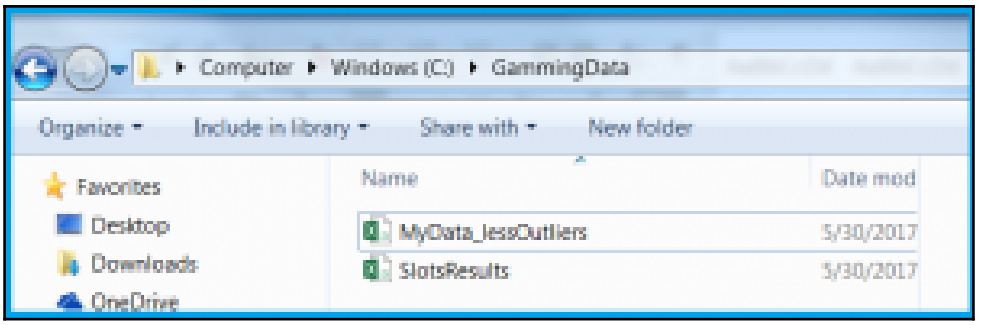


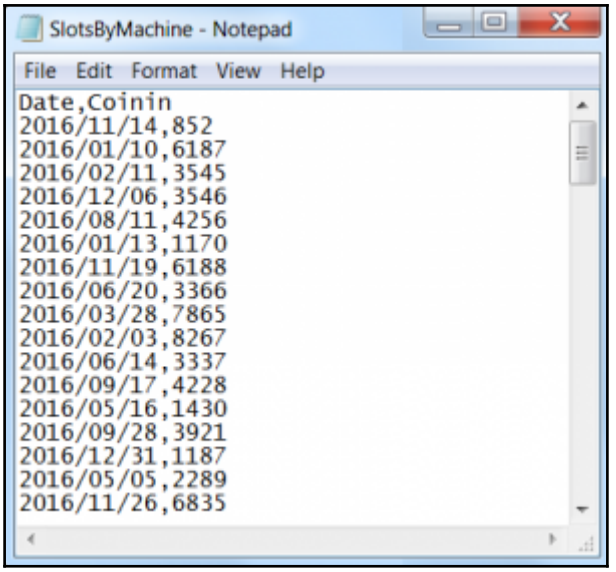


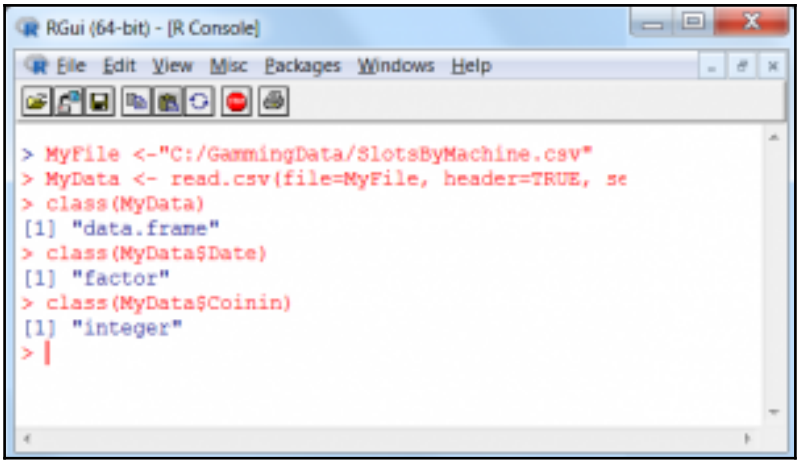
Chapter 03: A Developer's Approach to Data Cleaning

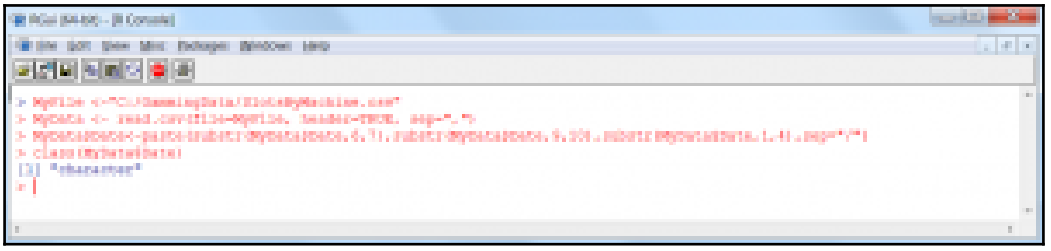


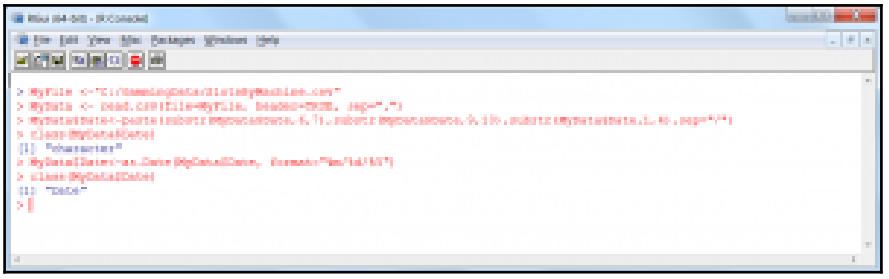


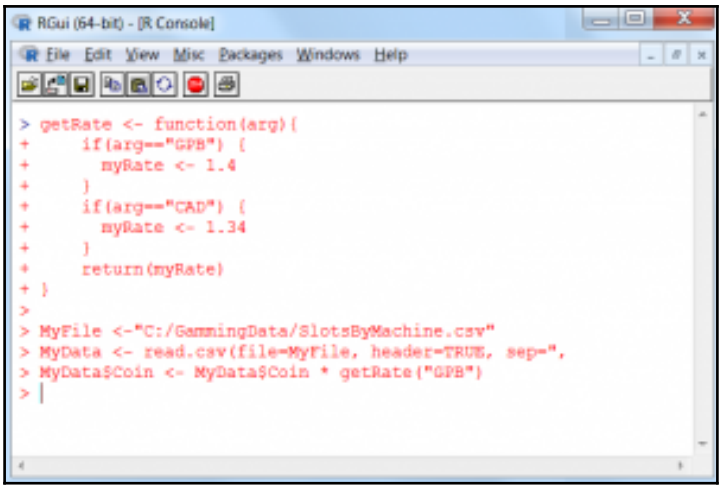

Computer ▸ Windows (C:) ▸ GammingData
Organize
Include in library
Share with
New folder
Favorites
Desktop
Downloads
OneDrive
Name
Date mod
MyData_lessOutliers
5/30/2017
SlotsResults
5/30/2017

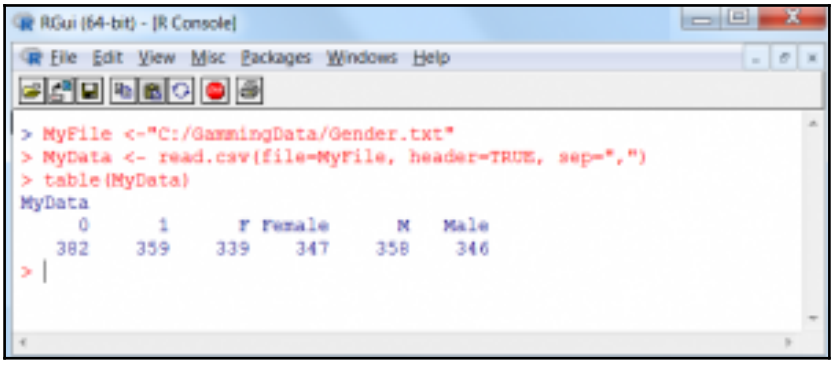

SlotsByMachine - Notepad
File Edit Format View Help
Date,Coinin
2016/11/14,852
2016/01/10,6187
2016/02/11,3545
2016/12/06,3546
2016/08/11,4256
2016/01/13,1170
2016/11/19,6188
2016/06/20,3366
2016/03/28,7865
2016/02/03,8267
2016/06/14,3337
2016/09/17,4228
2016/05/16,1430
2016/09/28,3921
2016/12/31,1187
2016/05/05,2289
2016/11/26,6835

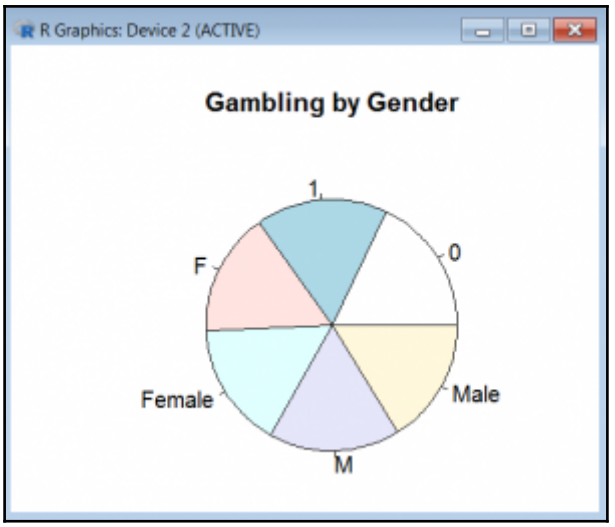

RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help
> MyFile <-"C:/GammingData/SlotsByMachine.csv"
> MyData <- read.csv(file=MyFile, header=TRUE, se
> class(MyData)
[1] "data.frame"
> class(MyData$Date)
[1] "factor"
> class(MyData$Coinin)
[1] "integer"

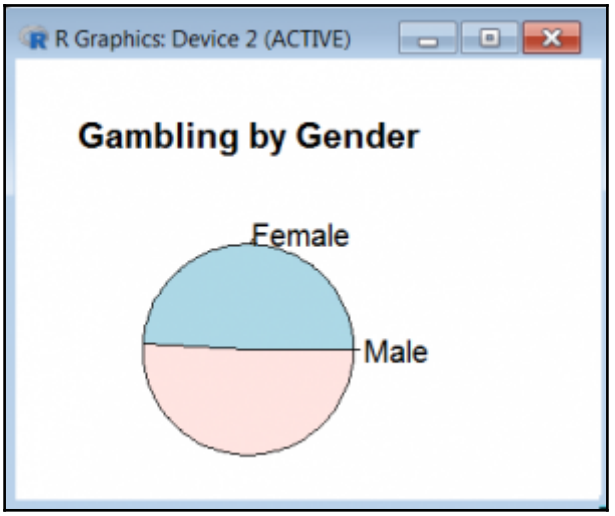


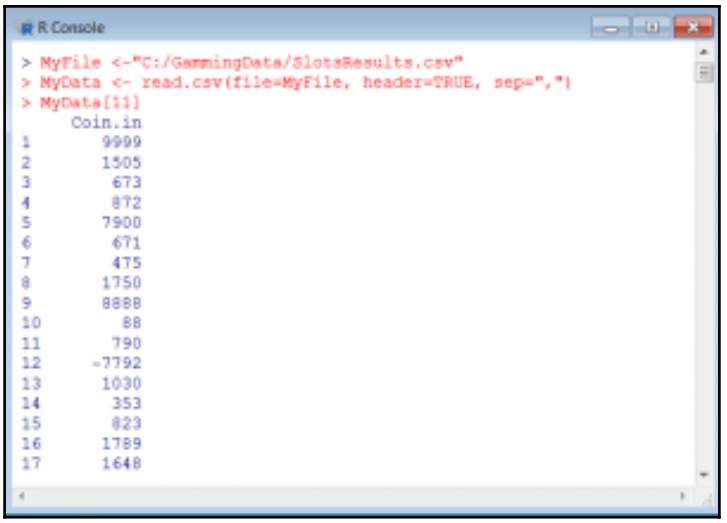


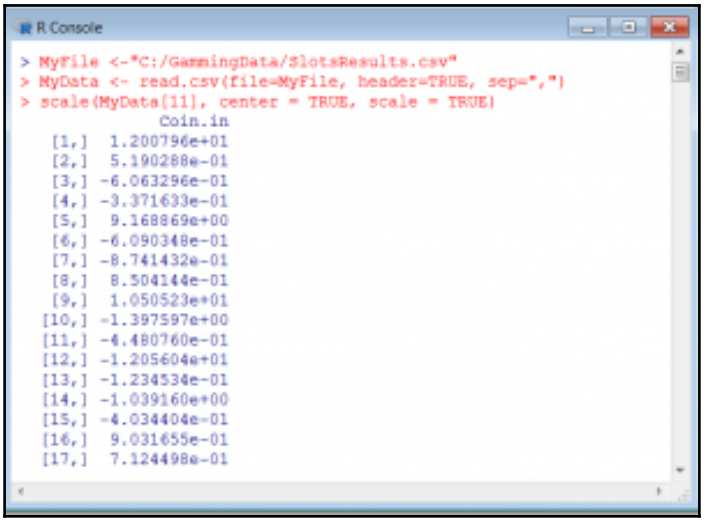

RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help
> getRate <- function(arg){
+ if(arg=="GPB") {
+ myRate <- 1.4
+ }
+ if(arg=="CAD") {
+ myRate <- 1.34
+ }
+ return(myRate)
+ }
>
> MyFile <-"C:/GammingData/SlotsByMachine.csv"
> MyData <- read.csv(file=MyFile, header=TRUE, sep=",
> MyData$Coin <- MyData$Coin * getRate("GPB")
>


RGui (64-bit) - [R Console]
File Edit View Misc Packages Windows Help
> MyFile <-"C:/GammingData/Gender.txt"
> MyData <- read.csv(file=MyFile, header=TRUE, sep=",")
> table(MyData)
MyData
0 1 F Female M Male
382 359 339 347 358 346
>

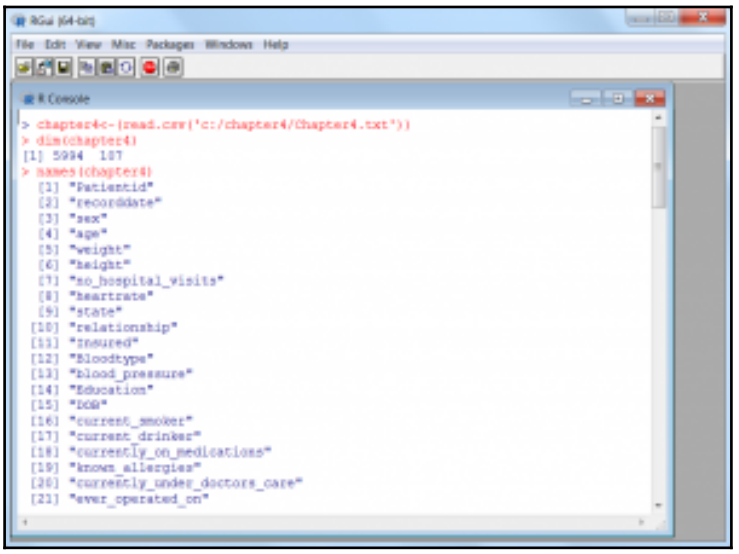

R Graphics: Device 2 (ACTIVE)
Gambling by Gender
1
0
F
Female
Male
M


R Graphics: Device 2 (ACTIVE)
Gambling by Gender
Female
Male


R Console
> MyFile <-"C:/GammingData/SlotsResults.csv"
> MyData <- read.csv(file=MyFile, header=TRUE, sep=",")
> MyData[11]
Coin.in
1 9999
2 1505
3 673
4 872
5 7900
6 671
7 475
8 1750
9 8888
10 88
11 790
12 -7792
13 1030
14 353
15 823
16 1789
17 1648


R Console
> MyFile <-"C:/GammingData/SlotsResults.csv"
> MyData <- read.csv(file=MyFile, header=TRUE, sep=",")
> scale(MyData[11], center = TRUE, scale = TRUE)
Coin.in
[1,] 1.200796e+01
[2,] 5.190288e-01
[3,] -6.063296e-01
[4,] -3.371633e-01
[5,] 9.168869e+00
[6,] -6.090348e-01
[7,] -8.741432e-01
[8,] 8.504144e-01
[9,] 1.050523e+01
[10,] -1.397597e+00
[11,] -4.480760e-01
[12,] -1.205604e+01
[13,] -1.234534e-01
[14,] -1.039160e+00
[15,] -4.034404e-01
[16,] 9.031655e-01
[17,] 7.124498e-01

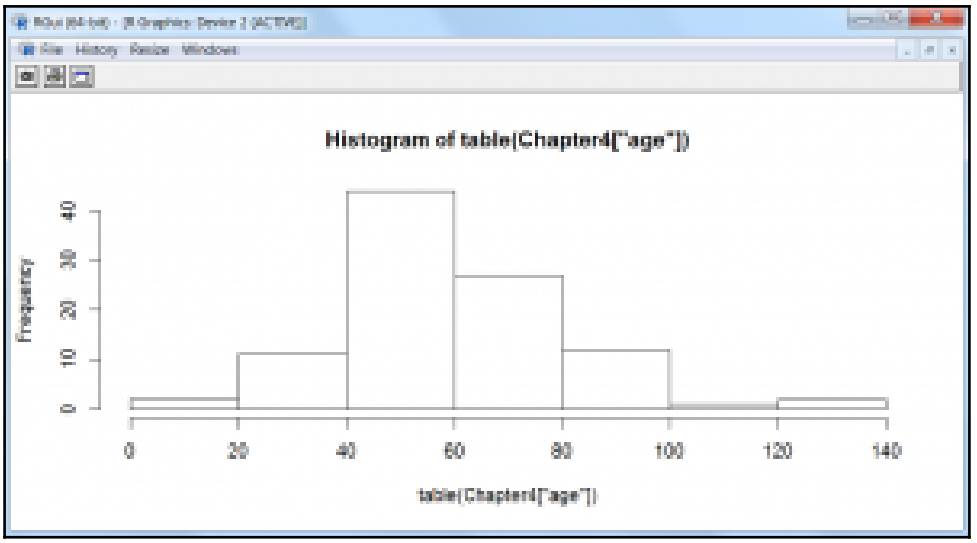
Chapter 04: Data Mining and the Database Developer



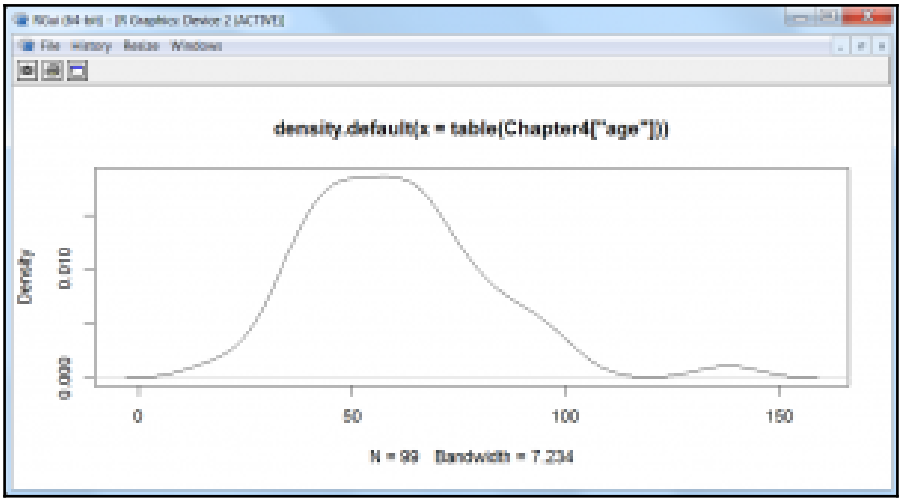

```
> table(chapter4["current_smoker"])

  No  Yes
5466  528
>
```

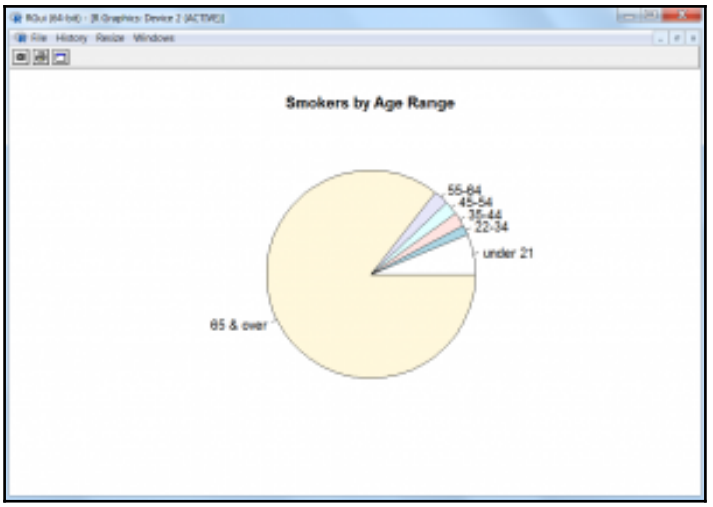
```
> range(Chapter4["age"])
[1]  1 99
>
```

```
> Chapter4<-read.csv('c:/chapter4/Chapter4.txt')
> hist(table(Chapter4["age"]))
>
```




```
> plot(density(table(Chapter4["age"])))
>
```



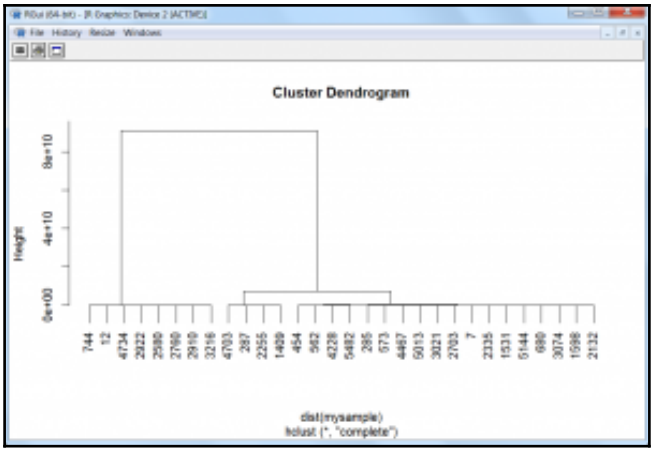



```
> Chapter4<-read.csv('c:/chapter4/Chapter4_NA.csv')
> nrow(Chapter4)
[1] 5994
> Chapter4<-na.omit(Chapter4)
> nrow(Chapter4)
[1] 5988
>
```

```
> # --- create a subset of smokers only
> mysub <- subset(Chapter4,Chapter4["current_smoker"]=="Yes")
> # --- confirm row count
> nrow(mysub)
[1] 528
>
```

```
> # --- create a random sample of 30 smokers
> mysample <- mysub[sample(1:nrow(mysub), 30,
+     replace=FALSE),]
> # --- confirm the row count in our sample
> nrow(mysample)
[1] 30
>
```

```
> # -- perform hierarchical cluster
> smokerclust<-hclust(dist(mysample))
> # -- create results in a dendrogram
> plot(smokerclust)
>
```

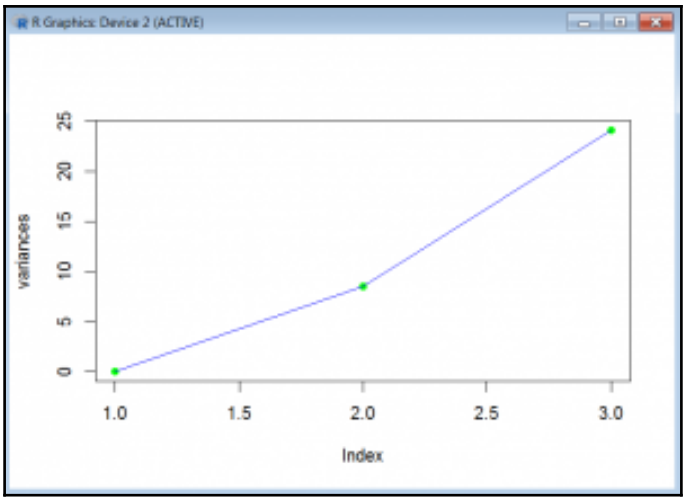



```
> Chapter4<-read.csv('c:/chapter4/varainces.csv')
> var(Chapter4["No_servings_per_week_skim_milk"])
                           No_servings_per_week_skim_milk
No_servings_per_week_skim_milk                  0.003160316
>
```

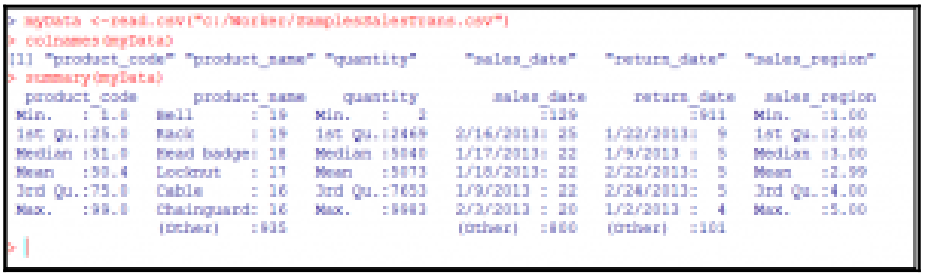
```
> Chapter4<-read.csv('c:/chapter4/varainces.csv')
> var(Chapter4["No_servings_per_week_skim_milk"])
                           No_servings_per_week_skim_milk
No_servings_per_week_skim_milk                  0.003160316
> var(Chapter4["No_servings_per_week_regular_or_diet_soda"])
                           No_servings_per_week_regular_or_diet_soda
No_servings_per_week_regular_or_diet_soda                  8.505655
>
```

```
> var(Chapter4["No_servings_per_week_water"])
                           No_servings_per_week_water
No_servings_per_week_water                   24.10477
>
```

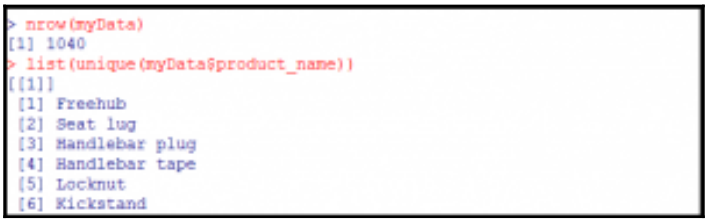
```
> Chapter4<-read.csv('c:/chapter4/varainces.csv')
> c1<-var(Chapter4["No_servings_per_week_skim_milk"])
> c2<-var(Chapter4["No_servings_per_week_regular_or_diet_soda"])
> c3<-var(Chapter4["No_servings_per_week_water"])
> variances<-c(c1, c2, c3)
> plot(variances, pch=16, col="green")
> lines(variances, col="blue")
>
```



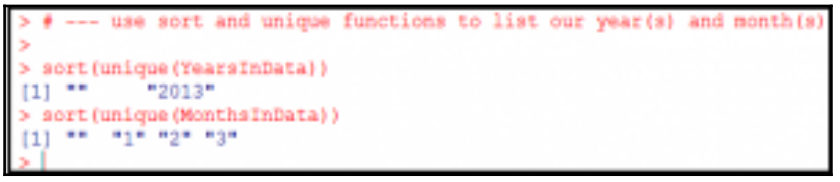
Chapter 05: Statistical Analysis for the Database Developer



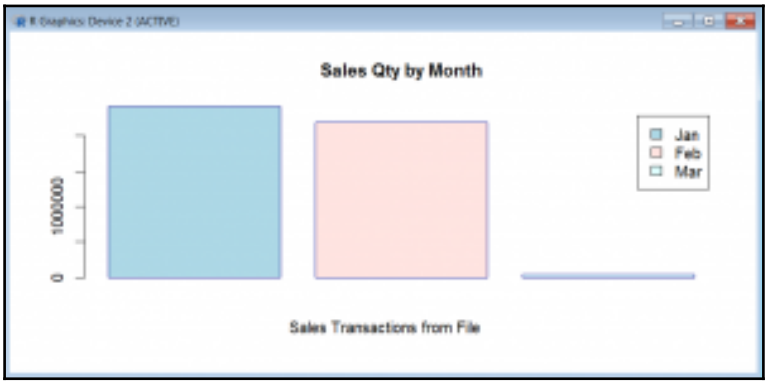
```
> myData <-read.csv("C:/Worker/SampleSalesTrans.csv")
> colnames(myData)
[1] "product_code"  "product_name"  "quantity"      "sales_date"    "return_date"   "sales_region"
> summary(myData)
  product_code      product_name    quantity        sales_date      return_date    sales_region
 Min.   :  1.0   Bell      : 19   Min.   :   2          :1129          :911   Min.   :1.00
 1st Qu.: 25.0   Rack      : 19   1st Qu.:2469   2/16/2013:  25   1/22/2013:  9   1st Qu.:2.00
 Median : 51.0   Head badge: 18   Median :5040   1/17/2013:  22   1/9/2013 :  5   Median :3.00
 Mean   : 50.4   Locknut   : 17   Mean   :5073   1/18/2013:  22   2/22/2013:  5   Mean   :2.99
 3rd Qu.: 75.0   Cable     : 16   3rd Qu.:7653   1/9/2013 :  22   2/28/2013:  5   3rd Qu.:4.00
 Max.   : 98.0   Chainguard: 16   Max.   :9983   2/3/2013 :  20   1/2/2013 :  4   Max.   :5.00
                 (other)   :835                  (other)  :800   (other)  :101
>
```



```
> nrow(myData)
[1] 1040
> list(unique(myData$product_name))
[[1]]
 [1] Freehub
 [2] Seat lug
 [3] Handlebar plug
 [4] Handlebar tape
 [5] Locknut
 [6] Kickstand
```



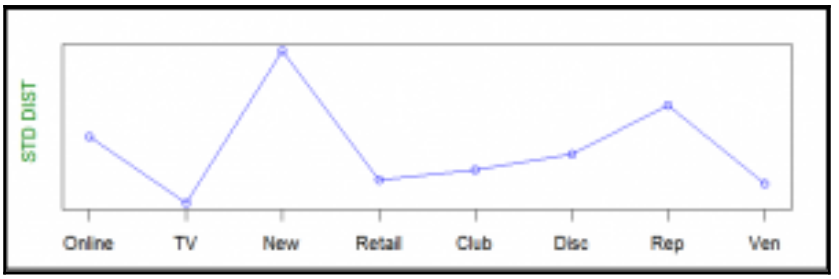
```
> # --- use sort and unique functions to list our year(s) and month(s)
>
> sort(unique(YearsInData))
[1] ""     "2013"
> sort(unique(MonthsInData))
[1] ""  "1" "2" "3"
>
```

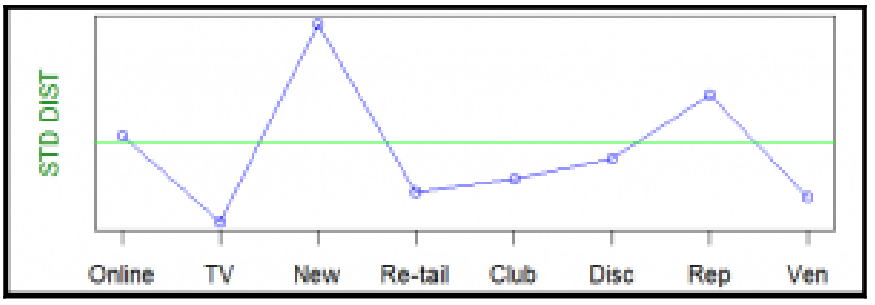


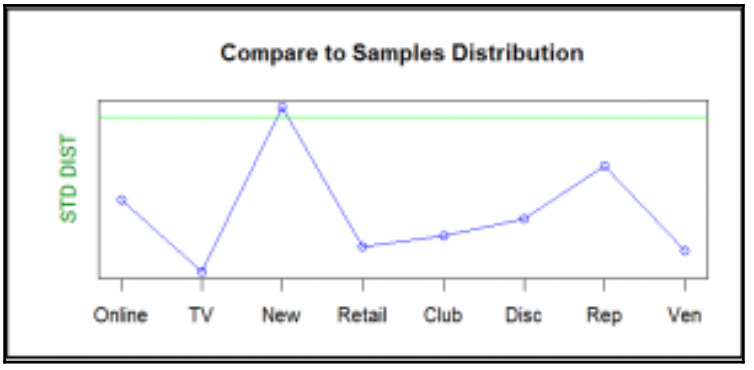
```
> a <-read.csv("c:/Rocker/SampleSalesTrans_2.csv", sep=",", header = TRUE, skip = 0)
> colnames(a)
[1] "product_code" "product_name" "quantity"     "sales_date"   "return_date"  "sales_region" "sale_type"
>
```

```
> summary(a)
 product_code
Min.   : 1.00
1st Qu.:25.00
Median :50.00
Mean   :49.55
3rd Qu.:74.00
Max.   :99.00


Barrel
Bicycle brake cable
Cap
Chainwheel
1) the disc component of a disc brake.[1][2] another name for a derailleur - a device that allows the handlebars and
Bar ends
(Other)
   quantity        sales_date      return_date    sales_region         sale_type
Min.   :  10          : 279          :1960   Min.   :1.000   Vendor       :256
1st Qu.:2598   1/22/2014:  27   2/2/2014 :   7   1st Qu.:1.000   Club         :249
Median :5122   2/27/2013:  27   1/7/2014 :   6   Median :3.000   Repeat       :247
Mean   :5044   2/12/2014:  26   2/14/2014:   6   Mean   :2.995   Online       :245
3rd Qu.:7488   1/3/2014 :  24   2/13/2014:   6   3rd Qu.:4.000   Retailer     :244
Max.   :9956   2/2/2013 :  24   2/17/2014:   6   Max.   :5.000   New Customer:176
               (Other)  :1832   (Other)  : 249                   (Other)      :523
>
```

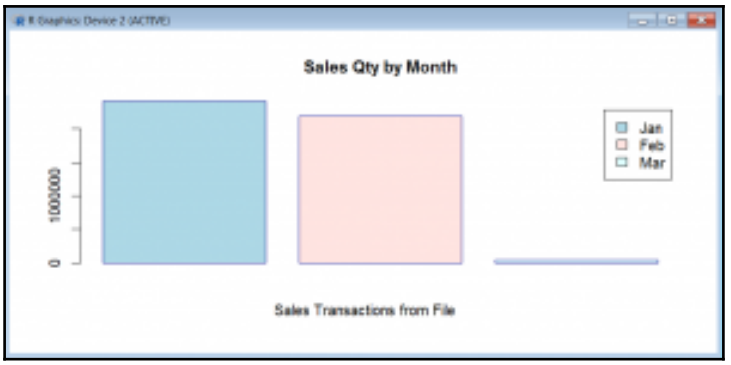
```
> # --- mean for all sale types:
> MeanAll <-mean(data.df [["quantity"]])
>
> # --- standard deviation for all sales types:
> StdDAll<-sd(data.df[["quantity"]])
>
> # --- median for all sales types:
> MeanAll <-mean(data.df [["quantity"]])
> MeanAll
[1] 5043.971
> StdDAll
[1] 2876.796
> MeanAll
[1] 5043.971
>
```

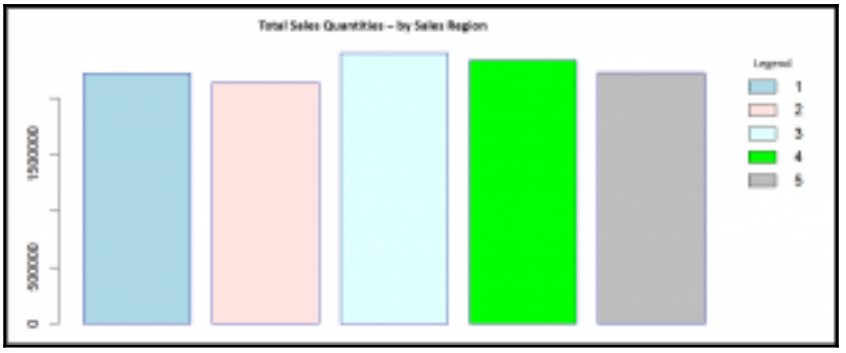




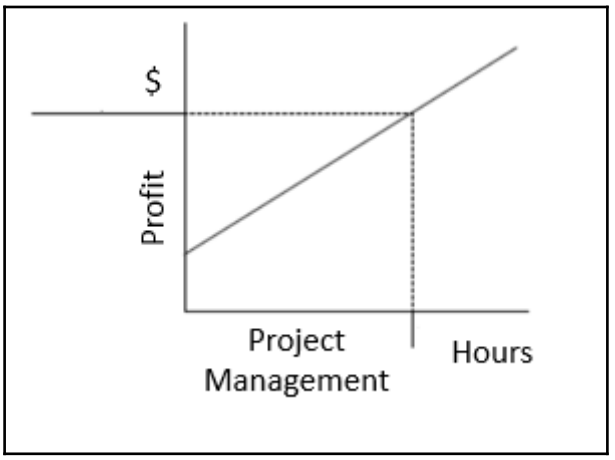


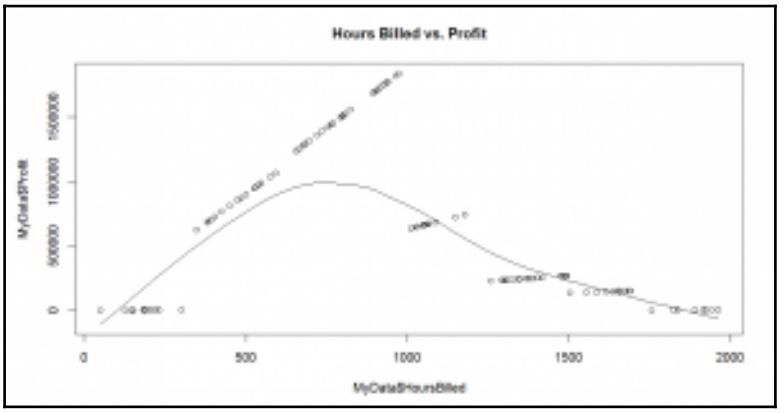

```
> df
        group  max     mean       sd median min     sum
1      Online 9902 5160.888 2803.941 5100.0  33 1470853
2          TV 9970 4516.475 2803.657 4305.0  30 1151701
3 New Customer 9954 5052.507 2986.567 4966.5 122 1394492
4    Retailer 9906 4972.775 2832.253 4862.0  10 1412268
5        Club 9980 5282.920 2843.501 5560.5  30 1521481
6  Discounted 9918 5408.142 2862.614 5622.0  10 1449382
7      Repeat 9950 4937.014 2920.933 5068.0  90 1416923
8      Vendor 9848 4987.676 2827.729 5137.0  40 1476352
> |
```

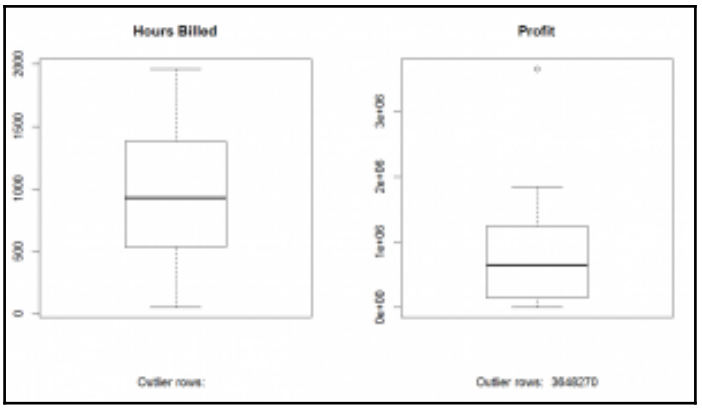

Sales Qty by Month
Jan
Feb
Mar
1000000
0
Sales Transactions from File

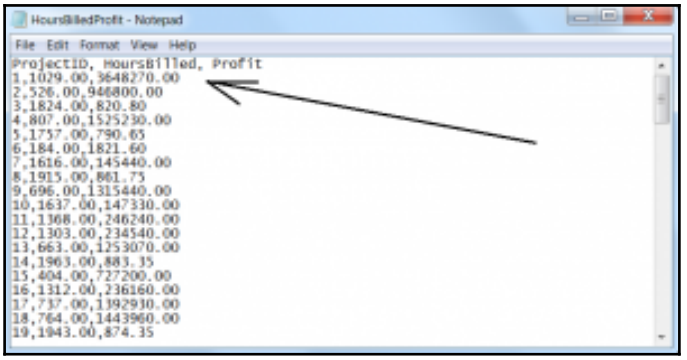

Total Sales Quantities – by Sales Region
Legend
1
2
3
4
5
1500000
500000
0

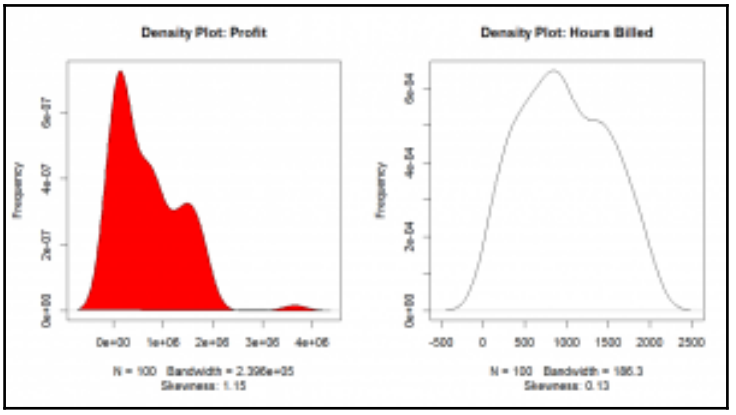
Chapter 06: Database Progression to Database Regression





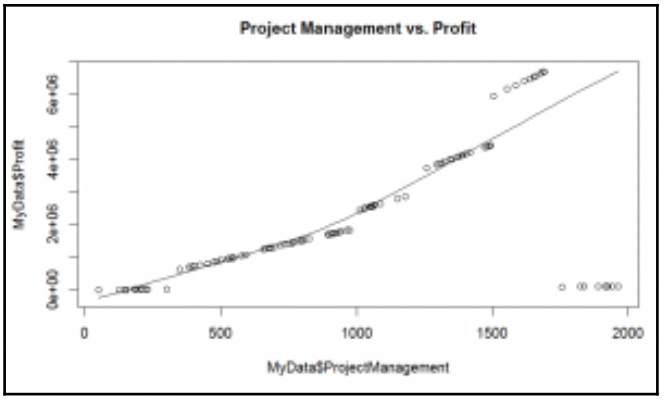







```
> cor(MyData$HoursBilled, MyData$Profit)
[1] -0.2578628
>
```

```
> MyData <- read.csv(file="c:/Worker/ProjectManagementProfit0.csv", header=TRUE, sep=",")
> cor(MyData$ProjectManagement, MyData$Profit)
[1] 0.9130142
>
```




```
> # --- build linear regression model on full data
> alinearMod <- lm(ProjectManagement ~ Profit, data=MyData)
> print(alinearMod)

Call:
lm(formula = ProjectManagement ~ Profit, data = MyData)

Coefficients:
(Intercept)       Profit
      6.180        1.629

>
```

```
> # --- predict project profitability
> ProfitPred <- predict(lmMod, testData)
> ProfitPred
        1         3         4         5         7        14        23        29        33        41
1838.5783  561.3375  833.4984  590.7587 1690.0242  582.5466  747.4471  707.8118  565.5875 1297.7161
       44        59        61        80        89        78        82        88        98       100
1342.8028  885.3103  865.7436  783.3871 1029.6436 1083.5952  792.6467 1020.2491  814.6033  732.8004
>
```

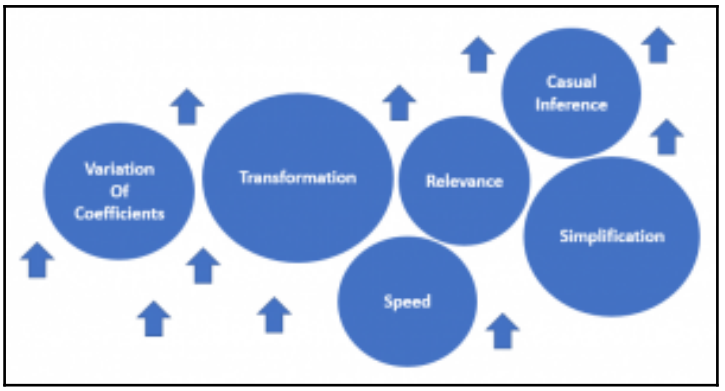
```
> summary (lmMod)

Call:
lm(formula = ProjectManagement ~ Profit, data = trainingData)

Residuals:
    Min      1Q  Median      3Q     Max
-516.48 -210.30  -43.53   59.69 1360.63

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.655e+02  6.448e+01   8.769 3.03e-13 ***
Profit      1.757e-04  2.136e-05   8.229 3.39e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 383.4 on 78 degrees of freedom
Multiple R-squared:  0.4647,    Adjusted R-squared:  0.4578
F-statistic: 67.71 on 1 and 78 DF,  p-value: 3.393e-12

>
```

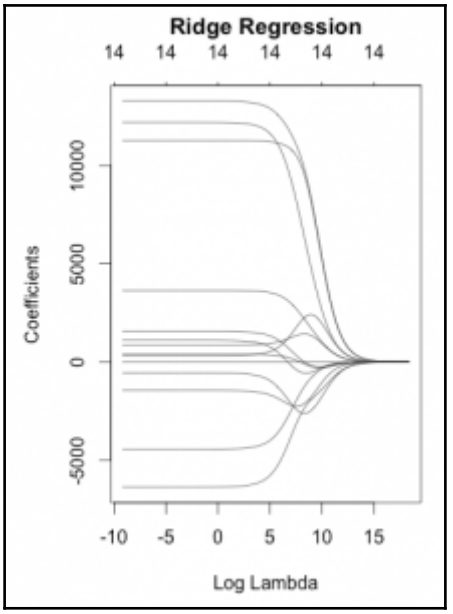
```
> actuals_preds <- data.frame(cbind(actuals=testData$ProjectManagement, predicteds=distPred))
> correlation_accuracy <- cor(actuals_preds)
> head(actuals_preds)
   actuals predicteds
1     1029  1004.8763
2     1624   581.3375
4      807   653.8984
5     1757   580.7547
7     1614  1690.0342
14    1963   582.5466
>
```

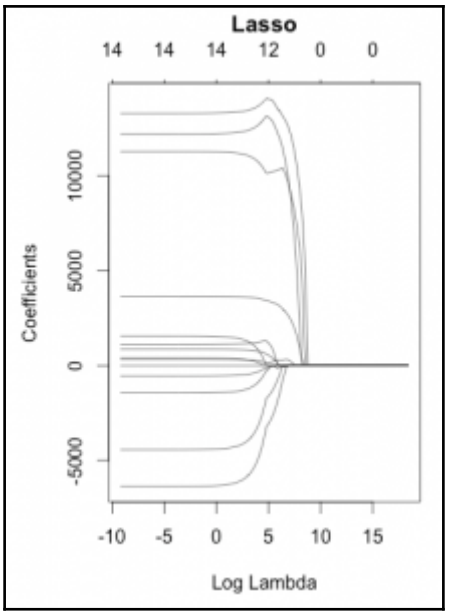
Chapter 07: Regularization for Database Improvement

```
> ols <- lm(y~ x1 + x2 + x3)
> summary(ols)

Call:
lm(formula = y ~ x1 + x2 + x3)

Residuals:
     Min       1Q   Median       3Q      Max
-1.19698 -0.28592  0.04026  0.24016  1.20322

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.4293     0.4916  -0.873   0.3954
x1            1.7851     0.4812   3.710   0.0019 **
x2            0.7119     0.4622   1.540   0.1430
x3            0.2839     0.5122   0.554   0.5870
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6306 on 16 degrees of freedom
Multiple R-squared:  0.4831,    Adjusted R-squared:  0.3862
F-statistic: 4.984 on 3 and 16 DF,  p-value: 0.0125

>
```

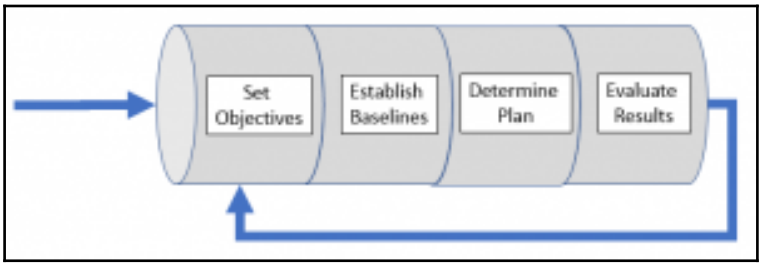
```
> # --- Fit model using ridge regression using independent variables
>
> ridge <- lm.ridge (y ~ x1 + x2 + x3, lambda = seq(0, .1, .001))
> summary(ridge)
       Length Class  Mode
coef   303    -none- numeric
scales   3    -none- numeric
Inter    1    -none- numeric
lambda 101    -none- numeric
ym       1    -none- numeric
xm       3    -none- numeric
GCV    101    -none- numeric
kHKB     1    -none- numeric
kLW      1    -none- numeric
>
```


Variation
Of
Coefficients
Transformation
Relevance
Casual
Inference
Simplification
Speed

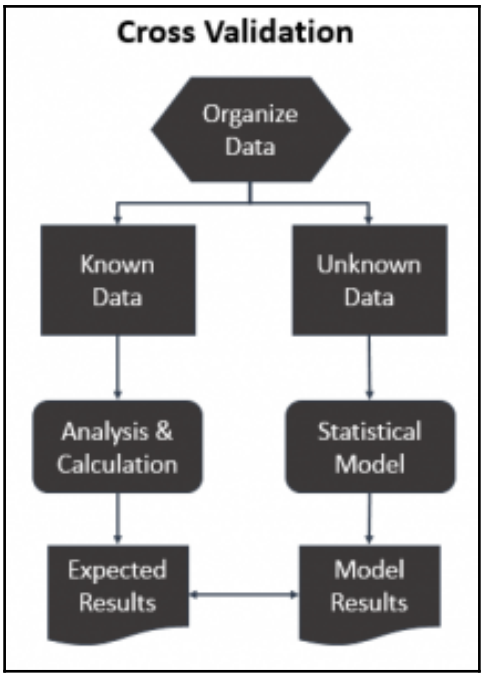

Ridge Regression
14 14 14 14 14 14
Coefficients
10000
5000
0
-5000
-10 -5 0 5 10 15
Log Lambda


Lasso
14 14 14 12 0 0
Coefficients
10000
5000
0
-5000
-10 -5 0 5 10 15
Log Lambda

Chapter 08: Database Development and Assessment

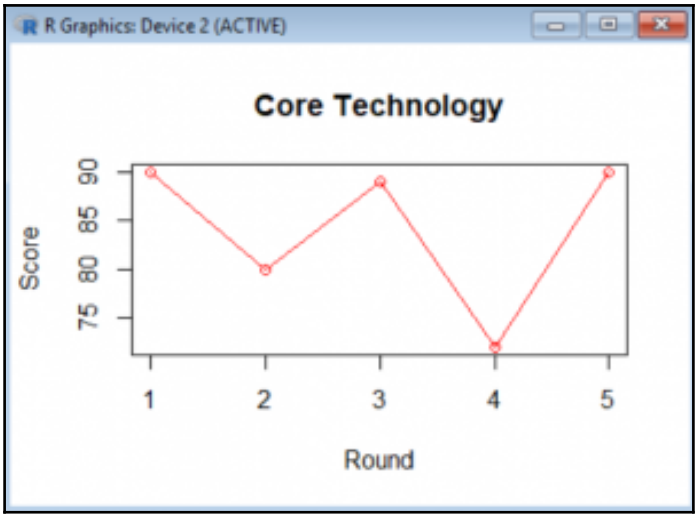


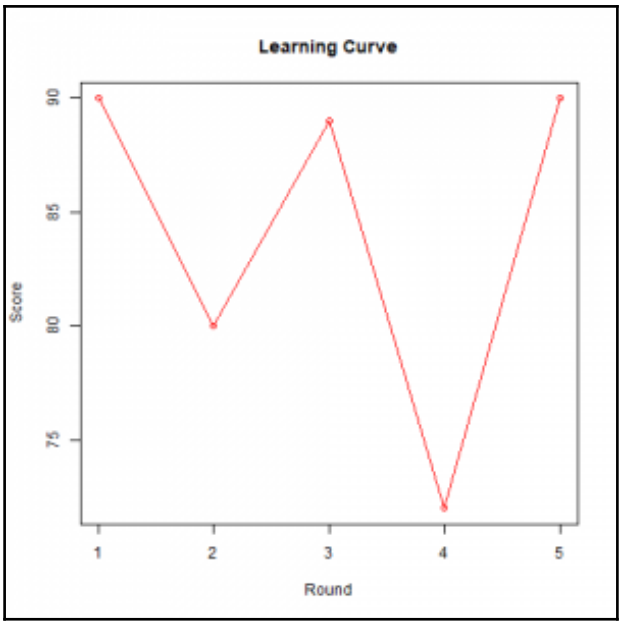

```
> ols <- lm(y~ x1 + x2 + x3)
> summary(ols)

Call:
lm(formula = y ~ x1 + x2 + x3)

Residuals:
     Min       1Q   Median       3Q      Max
-1.19698 -0.28592  0.04026  0.24016  1.20322

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.4293     0.4916  -0.873   0.3954
x1            1.7851     0.4812   3.710   0.0019 **
x2            0.7119     0.4622   1.540   0.1430
x3            0.2839     0.5122   0.554   0.5870
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6306 on 16 degrees of freedom
Multiple R-squared:  0.4831,    Adjusted R-squared:  0.3862
F-statistic: 4.984 on 3 and 16 DF,  p-value: 0.0125

>
```



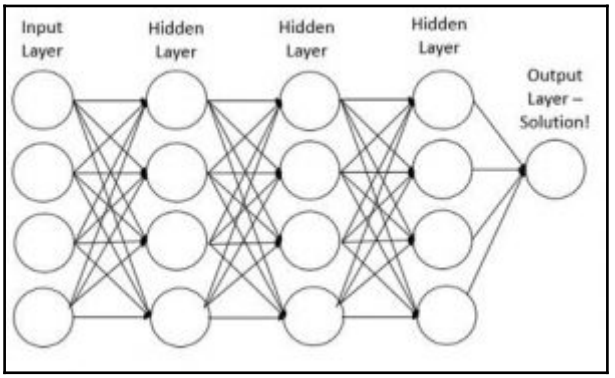
Characteristic	Validation Percent	Round 1 Percent	Round 2 Percent	Round 3 Percent	Round 4 Percent	Round 5 Percent	Average
Core Technology	90%	90%	80%	89%	72%	90%	85%
FT Project Manage	17%	38%	17%	30%	15%	17%	15%
FT Client Resource	79%	78%	77%	79%	80%	78%	79%
Sub-Contracted	92%	9%	5%	15%	44%	79%	41%
Time & Materials	89%	99%	99%	89%	79%	99%	89%
Not to Exceed	90%	69%	69%	69%	59%	70%	71%
Formal QA	99%	89%	89%	91%	52%	99%	93%
On-site	75%	78%	79%	81%	84%	88%	81%
Remote	89%	99%	99%	89%	99%	19%	82%

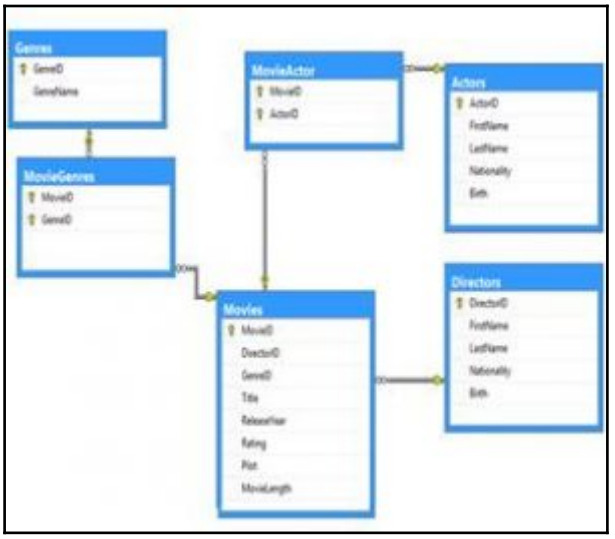
Characteristic	Validation Percent	Round 1 Percent	Round 2 Percent	Round 3 Percent	Round 4 Percent	Round 5 Percent	Average
Core Technology	90%	90%	80%	89%	72%	90%	85%
FT Project Manage	37%	36%	37%	30%	35%	37%	35%
FT Client Resource	79%	78%	77%	79%	80%	78%	79%
Sub-Contracted	52%	9%	5%	15%	44%	79%	41%
Time & Materials	69%	99%	99%	89%	79%	99%	89%
Not to Exceed	90%	69%	69%	69%	59%	70%	71%
Formal QA	99%	89%	89%	91%	92%	99%	93%
On-site	75%	78%	79%	81%	84%	88%	81%
Remote	89%	99%	99%	89%	99%	19%	82%



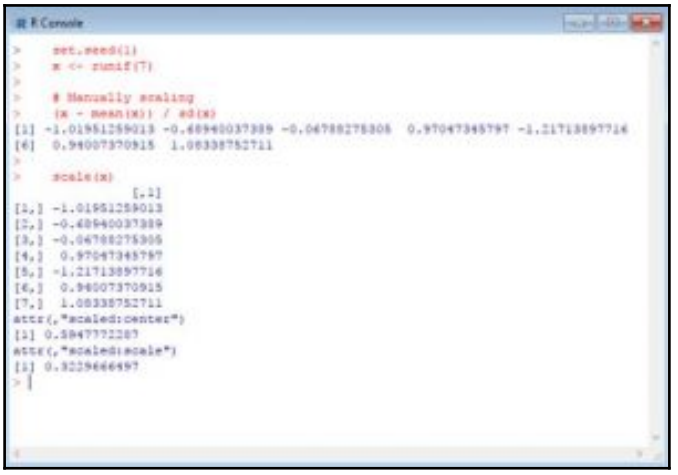

Learning Curve
Score
90
85
80
75
1
2
3
4
5
Round

Chapter 09: Databases and Neural Networks

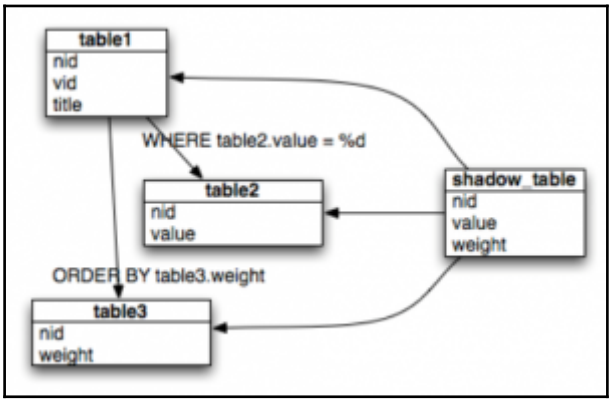




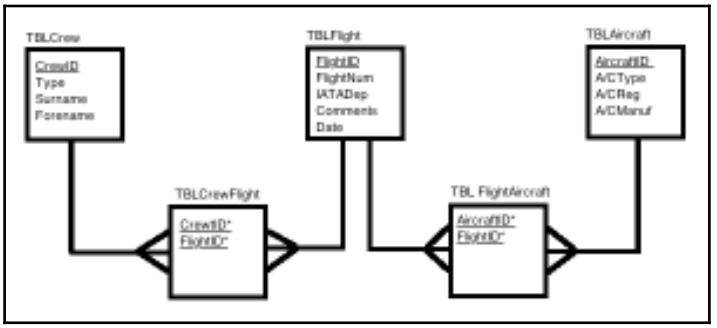
Input	Expected Output	Neural Net Output
1	1	0.9623402772
4	2	2.0083461217
9	3	2.9958221776
16	4	4.0009548085
25	5	5.0028838579
36	6	5.9975810435
49	7	6.9968278722
64	8	8.0070028670
81	9	9.0019220736
100	10	9.9222007864

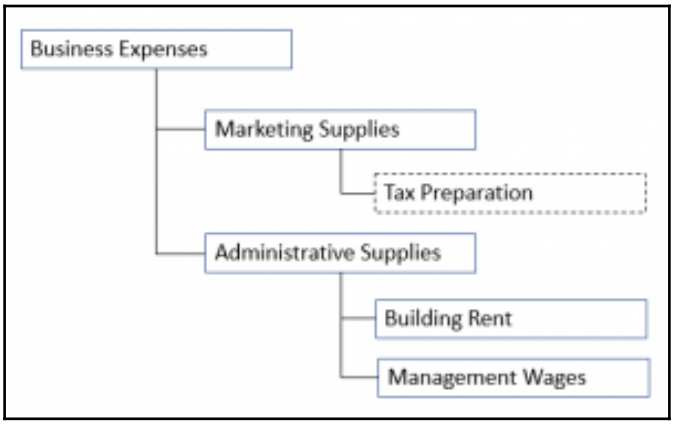


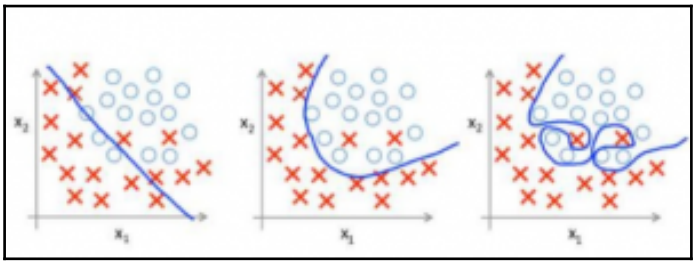
Chapter 10: Boosting your Database

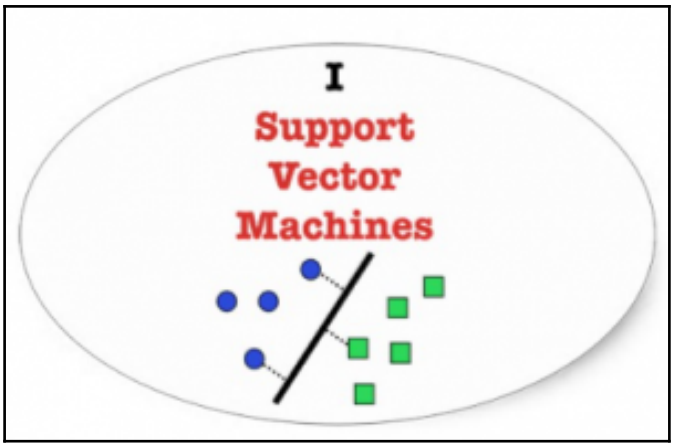


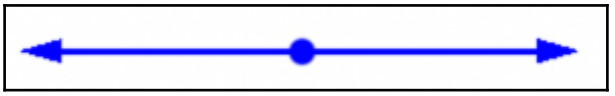
Chapter 11: Database Classification using Support Vector Machines

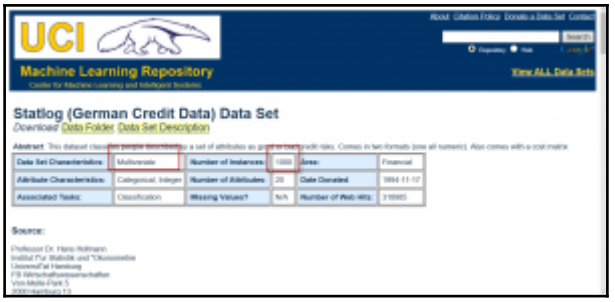


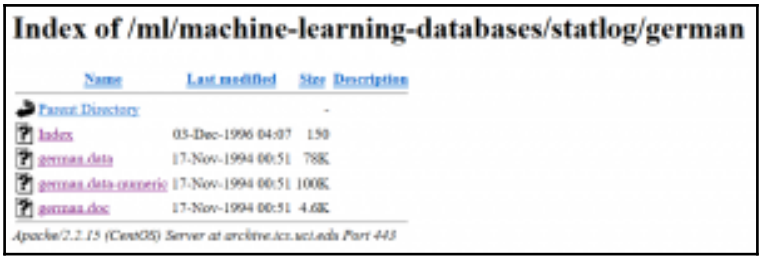












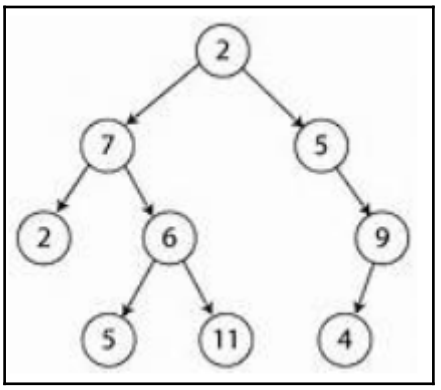
Index of /ml/machine-learning-databases/statlog/german

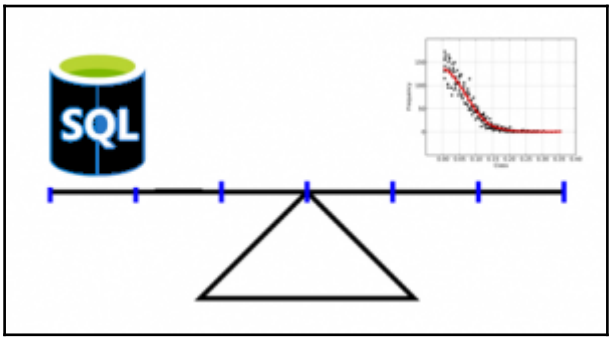
Name	Last modified	Size	Description
Parent Directory		-	
index	03-Dec-1996 04:07	150	
german.data	17-Nov-1994 00:51	78K	
german.data-numeric	17-Nov-1994 00:51	100K	
german.doc	17-Nov-1994 00:51	4.6K	

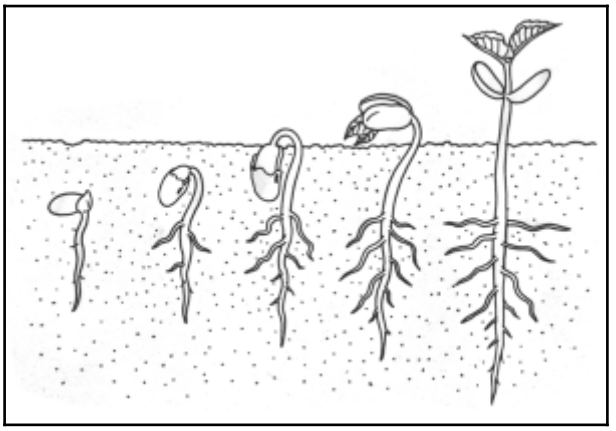
Apache/2.2.15 (CentOS) Server at archive.ics.uci.edu Port 443

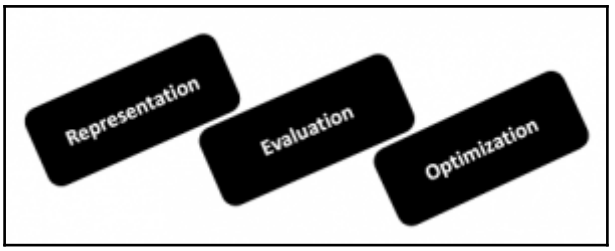
```
A11 6 A34 A43 1169 A65 A75 4 A93 A101 4 A121 67 A143 A152 2 A173 1 A192 A201 1
A12 48 A32 A43 5951 A61 A73 2 A92 A101 2 A121 22 A143 A152 1 A173 1 A191 A201 2
A14 12 A34 A46 2096 A61 A74 2 A93 A101 3 A121 49 A143 A152 1 A172 2 A191 A201 1
A11 42 A32 A42 7882 A61 A74 2 A93 A103 4 A122 45 A143 A153 1 A173 2 A191 A201 1
A11 24 A33 A40 4870 A61 A73 3 A93 A101 4 A124 53 A143 A153 2 A173 2 A191 A201 2
A14 36 A32 A46 9055 A65 A73 2 A93 A101 4 A124 35 A143 A153 1 A172 2 A192 A201 1
A14 24 A32 A42 2835 A63 A75 3 A93 A101 4 A122 53 A143 A152 1 A173 1 A191 A201 1
A12 36 A32 A41 6948 A61 A73 2 A93 A101 2 A123 35 A143 A151 1 A174 1 A192 A201 1
A14 12 A32 A43 3059 A64 A74 2 A91 A101 4 A121 61 A143 A152 1 A172 1 A191 A201 1
A12 30 A34 A40 5234 A61 A71 4 A94 A101 2 A123 28 A143 A152 2 A174 1 A191 A201 2
A12 12 A32 A40 1295 A61 A72 3 A92 A101 1 A123 25 A143 A151 1 A173 1 A191 A201 2
A11 48 A32 A49 4308 A61 A72 3 A92 A101 4 A122 24 A143 A151 1 A173 1 A191 A201 2
A12 12 A32 A43 1567 A61 A73 1 A92 A101 1 A123 22 A143 A152 1 A173 1 A192 A201 1
A11 24 A34 A40 1199 A61 A75 4 A93 A101 4 A123 60 A143 A152 2 A172 1 A191 A201 2
A11 15 A32 A40 1403 A61 A73 2 A92 A101 4 A123 28 A143 A151 1 A173 1 A191 A201 1
A11 24 A32 A43 1282 A62 A73 4 A92 A101 2 A123 32 A143 A152 1 A172 1 A191 A201 2
A14 24 A34 A43 2424 A65 A75 4 A93 A101 4 A122 53 A143 A152 2 A173 1 A191 A201 1
A11 30 A30 A49 8072 A65 A72 2 A93 A101 3 A123 25 A141 A152 3 A173 1 A191 A201 1
A12 24 A32 A41 12579 A61 A75 4 A92 A101 2 A124 44 A143 A153 1 A174 1 A192 A201 2
A14 24 A32 A43 3430 A63 A75 3 A93 A101 2 A123 31 A143 A152 1 A173 2 A192 A201 1
A14 9 A34 A40 2134 A61 A73 4 A93 A101 4 A123 48 A143 A152 3 A173 1 A192 A201 1
A11 6 A32 A43 2647 A63 A73 2 A93 A101 3 A121 44 A143 A151 1 A173 2 A191 A201 1
A11 10 A34 A40 2241 A61 A72 1 A93 A101 3 A121 48 A143 A151 2 A172 2 A191 A202 1
A12 12 A34 A41 1804 A62 A72 3 A93 A101 4 A122 44 A143 A152 1 A173 1 A191 A201 1
A14 10 A34 A42 2069 A65 A73 2 A94 A101 1 A123 26 A143 A152 2 A173 1 A191 A202 1
A11 6 A32 A42 1374 A61 A73 1 A93 A101 2 A121 36 A141 A152 1 A172 1 A192 A201 1
A14 6 A30 A43 426 A61 A75 4 A94 A101 4 A123 39 A143 A152 1 A172 1 A191 A201 1
```

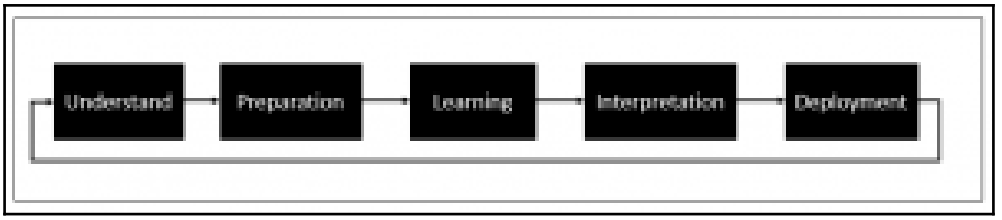
Chapter 12: Database Structures and Machine Learning








Representation
Evaluation
Optimization


Understand
Preparation
Learning
Interpretation
Deployment