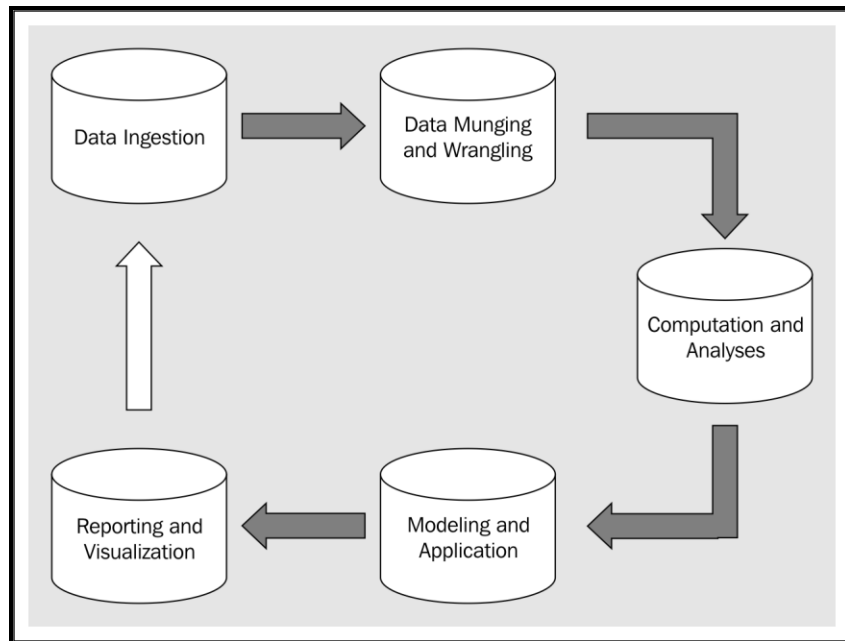


Chapter 1: Preparing Your Data Science Environment



```
R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (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.

[R.app GUI 1.62 (6558) x86_64-apple-darwin10.8.0]

[Workspace restored from /Users/seanmurphy/.RData]
[History restored from /Users/seanmurphy/.Rapp.history]

>
```

RStudio interface showing a console window on the left and a script editor on the right. The console displays the R version 3.0.2 (2013-09-25) -- "Frisbee Sailing" and various help messages. The script editor shows R code for loading and processing data from a CSV file named 'all.csv'.

```

1 Exploring the beans.
2
3
4 Loading stuff we need.
5
6
7 library(plyr)
8
9 setwd("~/Desktop/DR/Beans")
10 all <- read.csv("~/Desktop/DR/Beans/all.csv")
11
12
13
14
15
16
17
18 #Drop the deadbeat from Calazan
19 all <- subset(all, association!="")
20
21 #Eliminate observations with sum = NA or sum = 0
22 all <- subset(all, !is.na(sum))
23 all <- subset(all, sum > 0.0)
24

```

The Environment pane shows the following data objects:

Object	Details
AdTrial	1 obs. of 61 variables
AdTrials	31096 obs. of 61 variables
best_df	5 obs. of 18 variables
bob	num [1:2, 1:2] 78 109 2576 2879
champ_df	1 obs. of 18 variables
contest	164 obs. of 18 variables
not_champ_df	1 obs. of 18 variables
results	5267 obs. of 14 variables
results_boostNA	3841 obs. of 14 variables
results_champ_won	1518 obs. of 14 variables
results_client_decision	226 obs. of 13 variables
results_upset	640 obs. of 14 variables
temp_df	2 obs. of 18 variables

The Values pane shows:

```

ad_trial_ids int [1:7389] NA 13538 13544 13552 13562 13624 NA NA 13905 14031...
adtrialid   20862L

```

The Files pane shows a list of files in the current directory, including R scripts, R Markdown files, and other documents.

Install Packages dialog box. The 'Install from:' dropdown is set to 'Repository (CRAN)'. The 'Install to Library:' dropdown is set to '/Library/Frameworks/R.framework/Versions/3.0/Resources/l'. The 'Install dependencies' checkbox is checked. The 'Install' and 'Cancel' buttons are visible at the bottom.

Install from: Configuring Repositories

Repository (CRAN)

Packages (separate multiple with space or comma):

Install to Library: /Library/Frameworks/R.framework/Versions/3.0/Resources/l

Install dependencies

Install Cancel

```
3. seanmurphy@SPM: ~ (zsh)
Last login: Tue Apr 22 16:40:14 on ttys005
→ ~ virtualenv
You must provide a DEST_DIR
Usage: virtualenv [OPTIONS] DEST_DIR

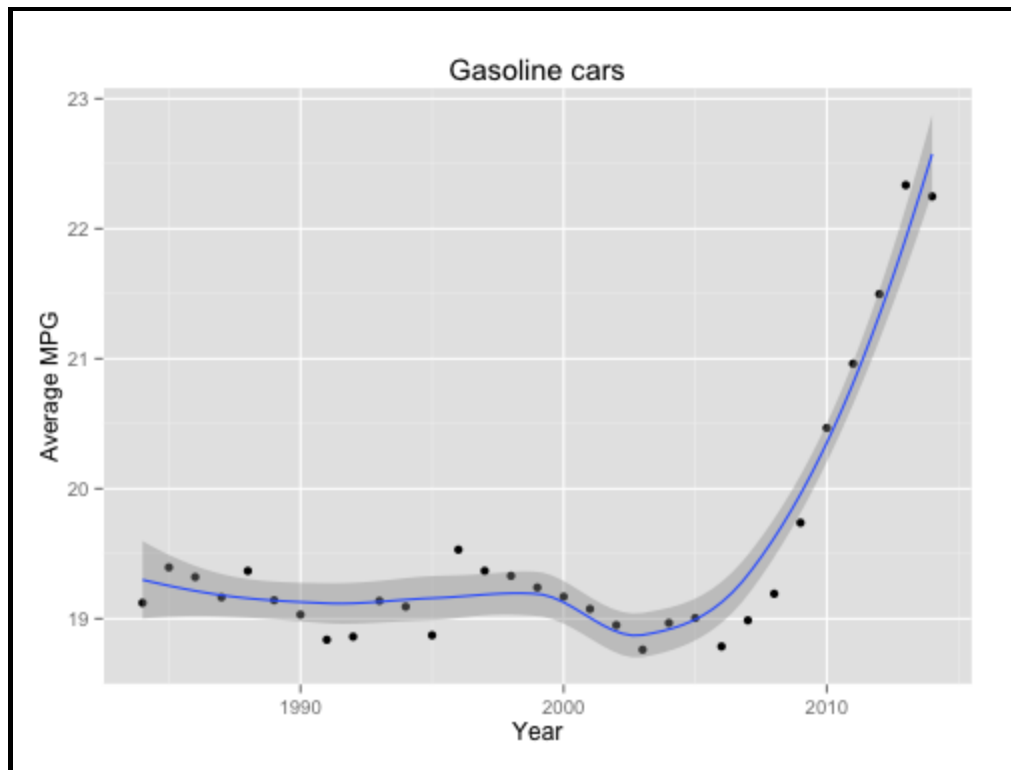
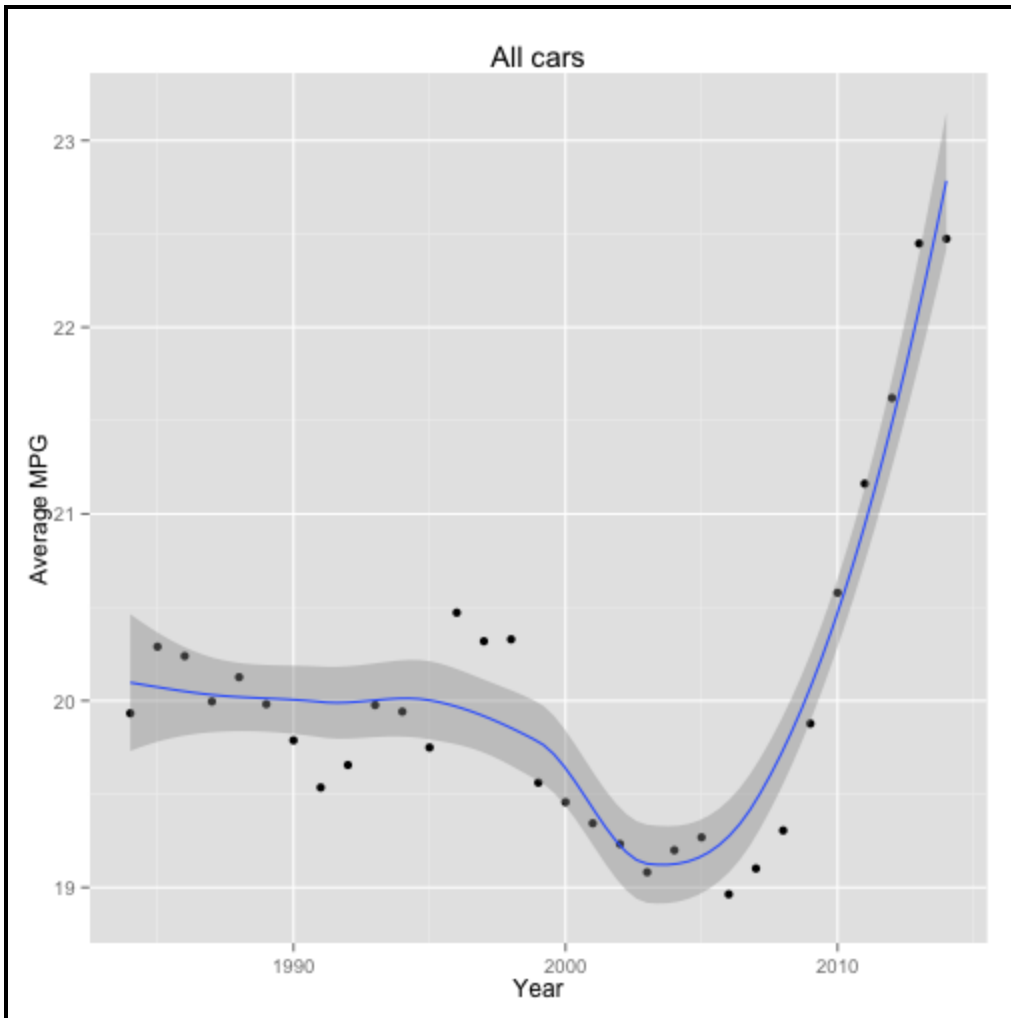
Options:
  --version            show program's version number and exit
  -h, --help          show this help message and exit
  -v, --verbose       Increase verbosity.
  -q, --quiet         Decrease verbosity.
  -p PYTHON_EXE, --python=PYTHON_EXE
                    The Python interpreter to use, e.g.,
                    --python=python2.5 will use the python2.5 interpreter
                    to create the new environment. The default is the
                    interpreter that virtualenv was installed with
                    (/usr/bin/python)
  --clear             Clear out the non-root install and start from scratch.
  --no-site-packages DEPRECATED. Retained only for backward compatibility.
                    Not having access to global site-packages is now the
                    default behavior.
  --system-site-packages
                    Give the virtual environment access to the global
                    site-packages.
  --always-copy       Always copy files rather than symlinking.
  --unzip-setuptools Unzip Setuptools when installing it.
  --relocatable       Make an EXISTING virtualenv environment relocatable.
                    This fixes up scripts and makes all .pth files
                    relative.
  --no-setuptools     Do not install setuptools (or pip) in the new
                    virtualenv.
  --no-pip            Do not install pip in the new virtualenv.
  --extra-search-dir=DIR
                    Directory to look for setuptools/pip distributions in.
                    This option can be used multiple times.
  --never-download   DEPRECATED. Retained only for backward compatibility.
                    This option has no effect. Virtualenv never downloads
                    pip or setuptools.
  --prompt=PROMPT    Provides an alternative prompt prefix for this
                    environment.
  --setuptools        DEPRECATED. Retained only for backward compatibility.
                    This option has no effect.
  --distribute        DEPRECATED. Retained only for backward compatibility.
                    This option has no effect.

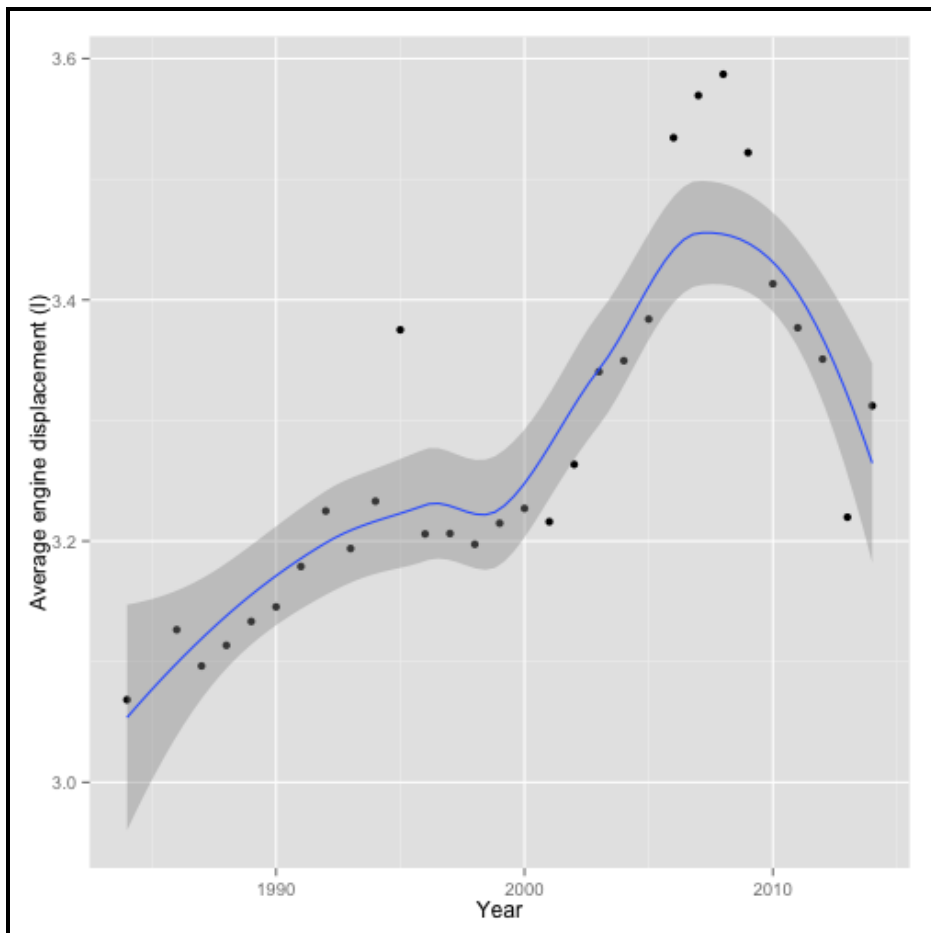
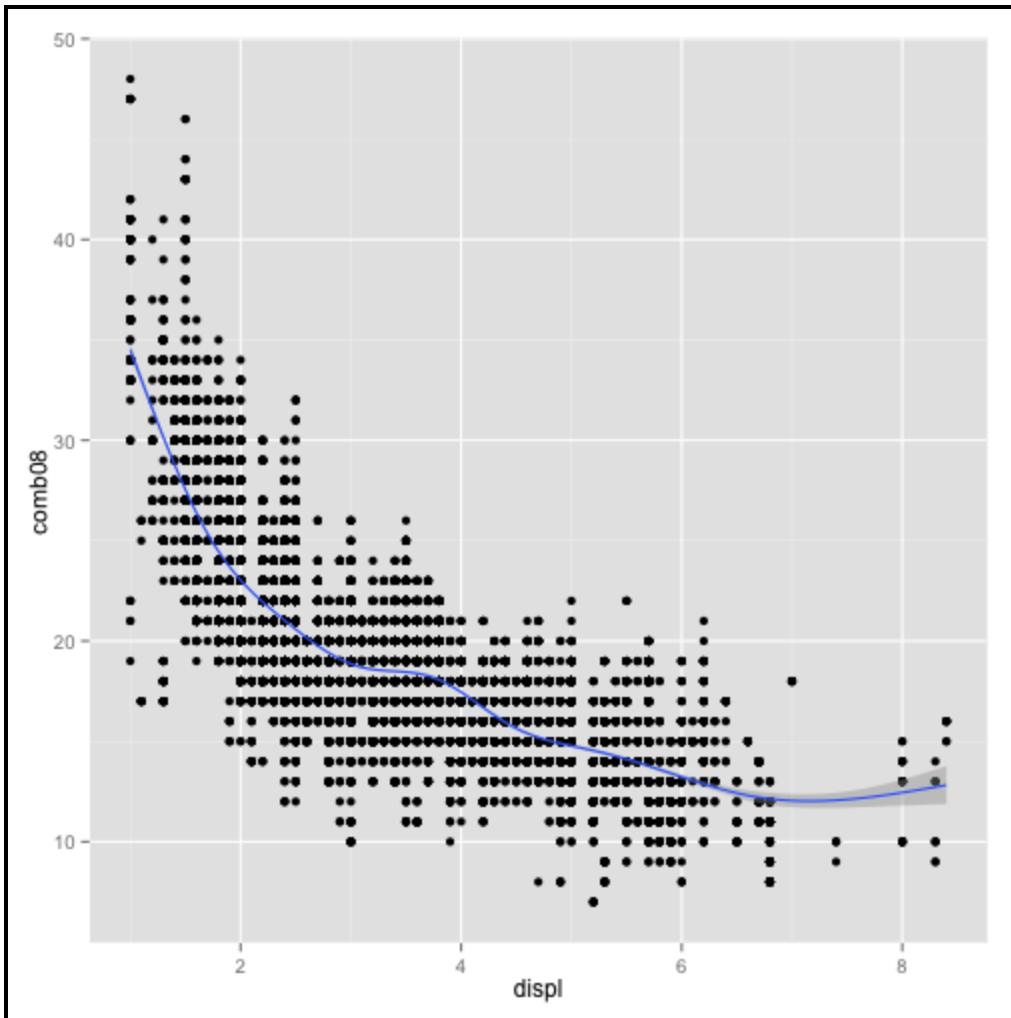
→ ~
```

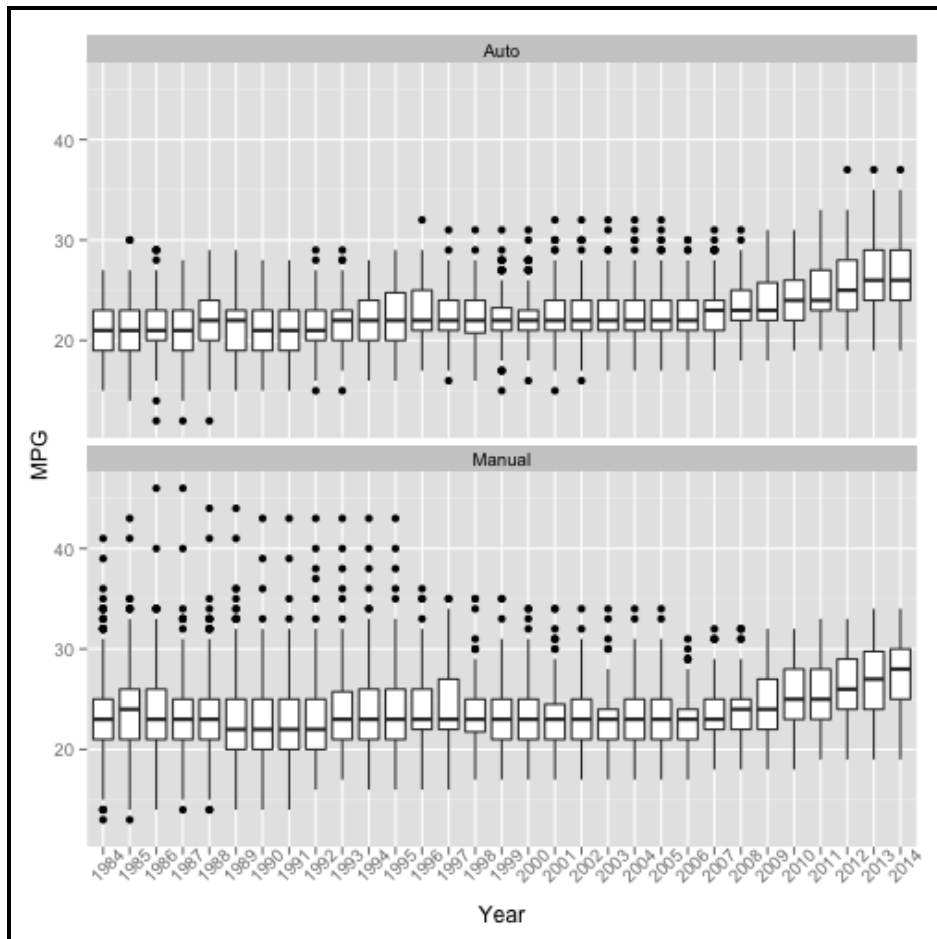
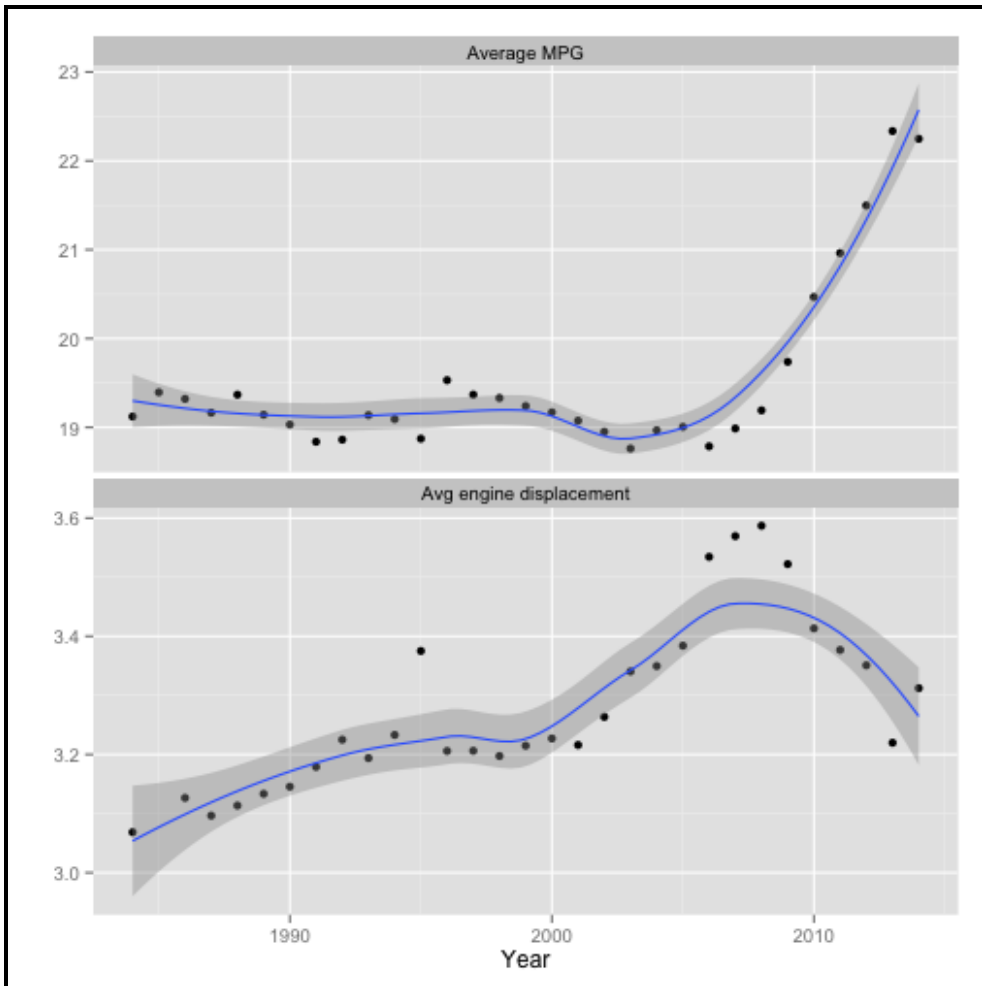
Chapter 2: Driving Visual Analysis with Automobile Data (R)

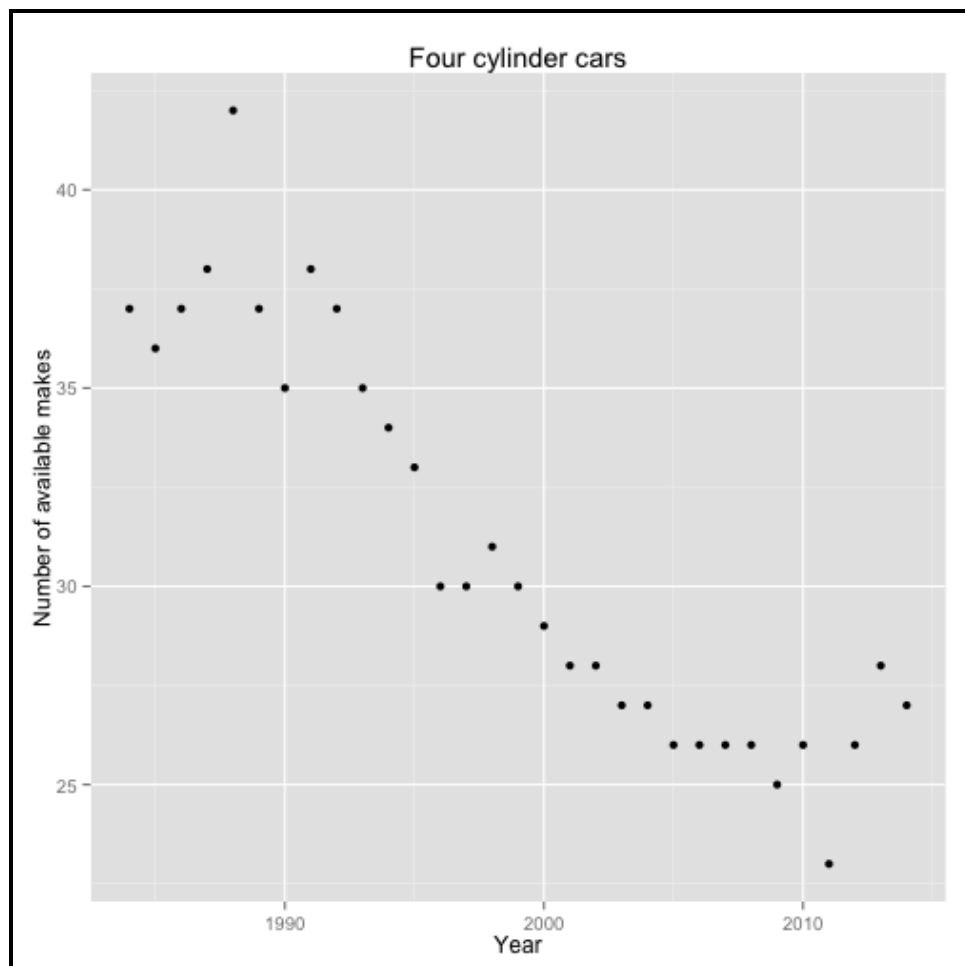
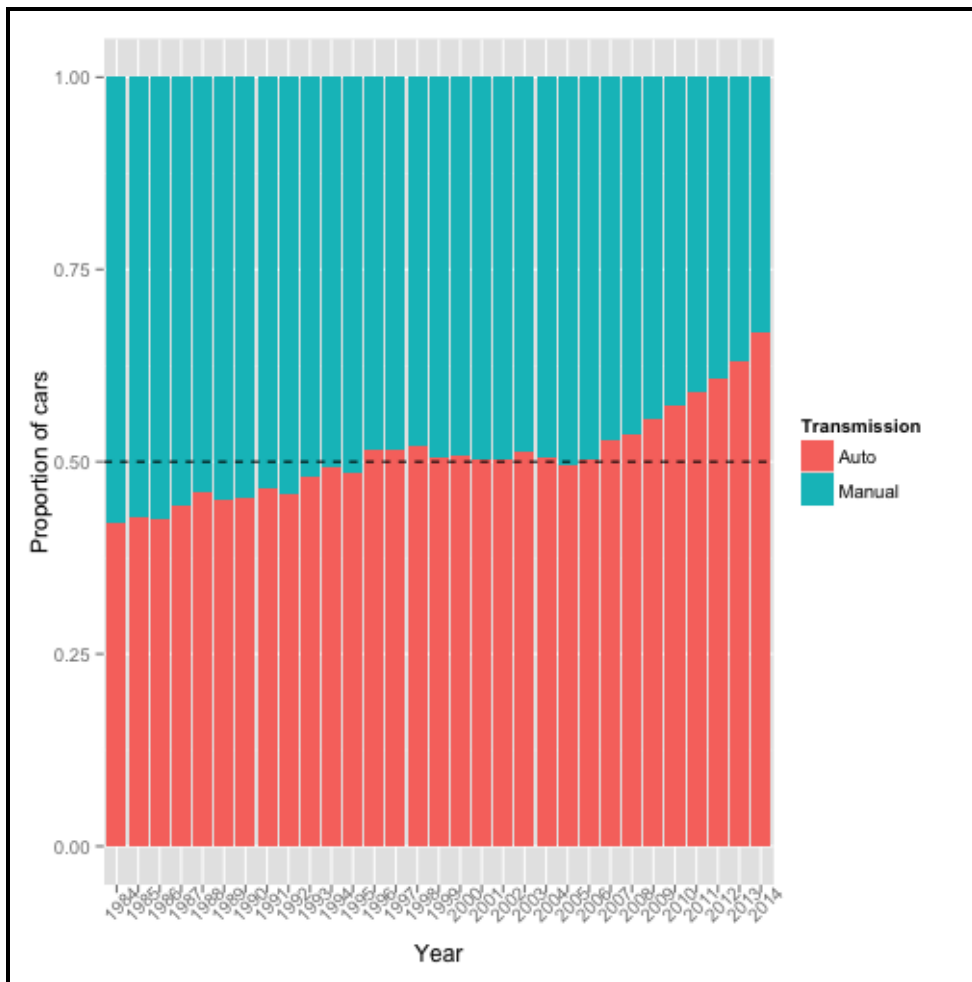
```
> names(vehicles)
 [1] "barrels08"      "barrelsA08"      "charge120"      "charge240"      "city08"         "city08U"
 [7] "cityA08"       "cityA08U"       "cityCD"        "cityE"         "cityUF"        "co2"
[13] "co2A"         "co2TailpipeAGpm" "co2TailpipeGpm" "comb08"        "comb08U"       "combA08"
[19] "combA08U"     "combE"          "combinedCD"    "combinedUF"    "cylinders"     "displ"
[25] "drive"        "engId"          "eng_dscr"     "feScore"       "fuelCost08"    "fuelCostA08"
[31] "fuelType"     "fuelType1"     "ghgScore"     "ghgScoreA"    "highway08"     "highway08U"
[37] "highwayA08"   "highwayA08U"   "highwayCD"    "highwayE"     "highwayUF"     "hlv"
[43] "hpv"         "id"            "lv2"         "lv4"         "make"         "model"
[49] "mpgData"     "phevBlended"   "pv2"         "pv4"         "range"        "rangeCity"
[55] "rangeCityA"  "rangeHwy"     "rangeHwyA"   "trany"       "UCity"        "UCityA"
[61] "UHighway"   "UHighwayA"    "VClass"     "year"        "youSaveSpend"  "guzzler"
[67] "trans_dscr"  "tCharger"     "sCharger"    "atvType"     "fuelType2"    "rangeA"
[73] "evMotor"    "mfrCode"     "trany2"
> |
```

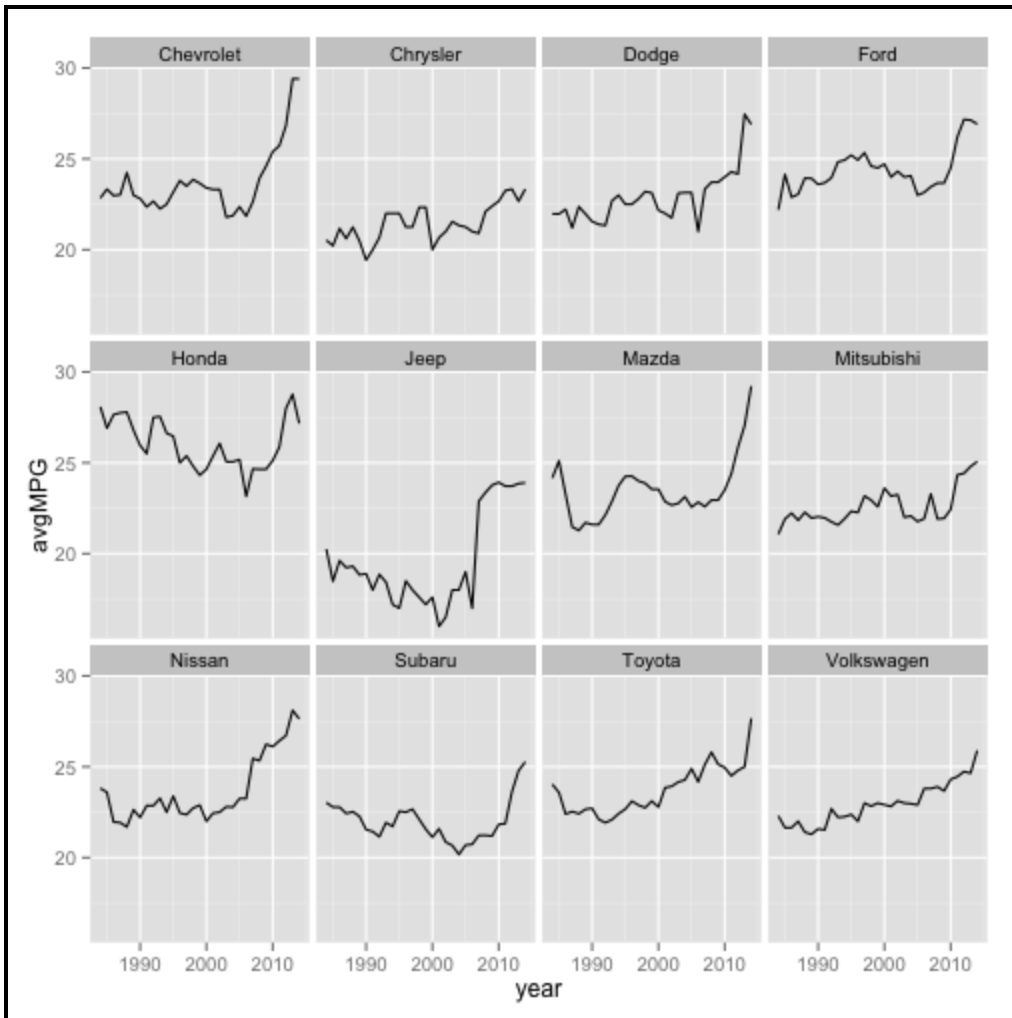
```
> with(vehicles, table(sCharger, year))
      year
sCharger 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005
      S    0    0    0    0    0    4    4    2    5    5    3    5    6    5    12   12   14   20   26   29   33   30
      year
sCharger 2006 2007 2008 2009 2010 2011 2012 2013 2014
      S   37   28   35   19   18   20   28   42   57
> |
```











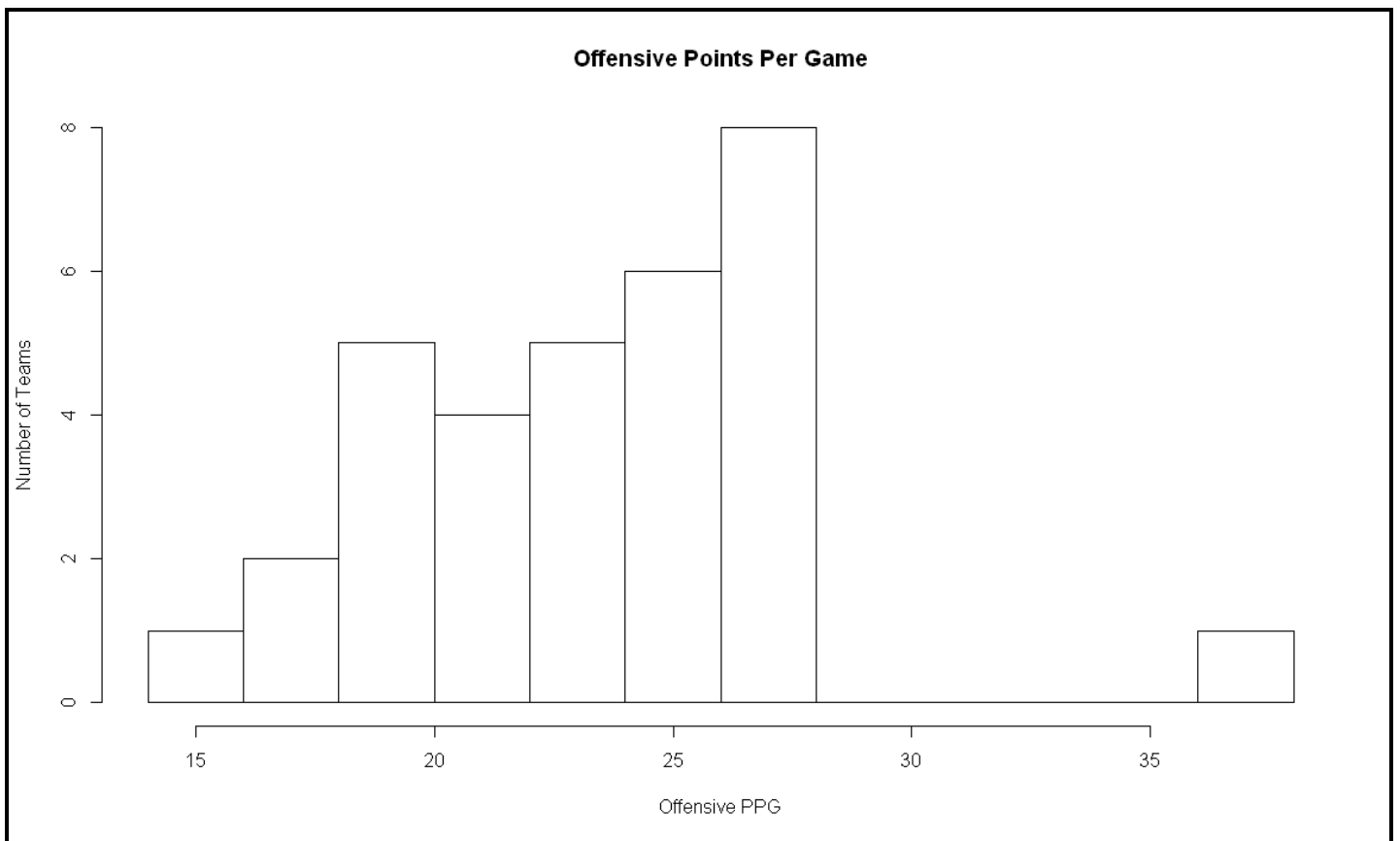
Chapter 3: Simulating American Football Data (R)

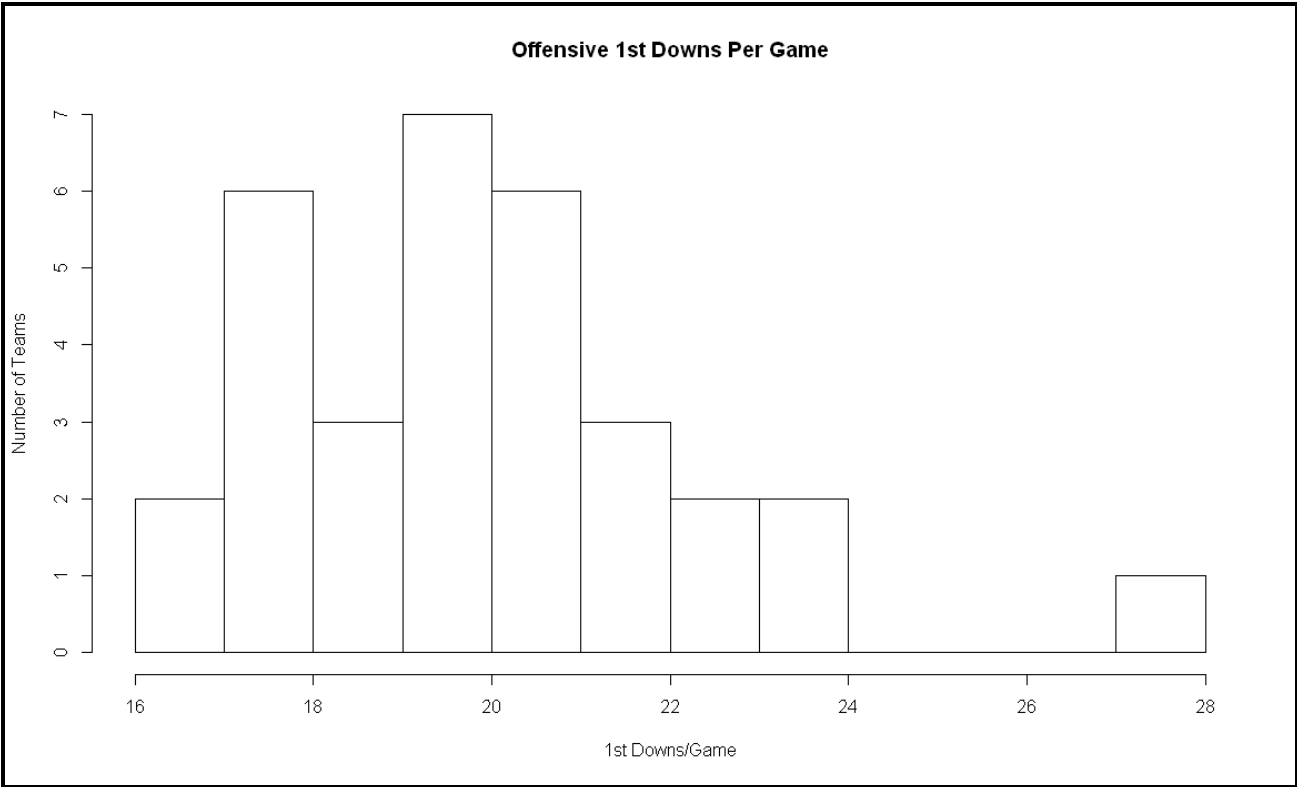
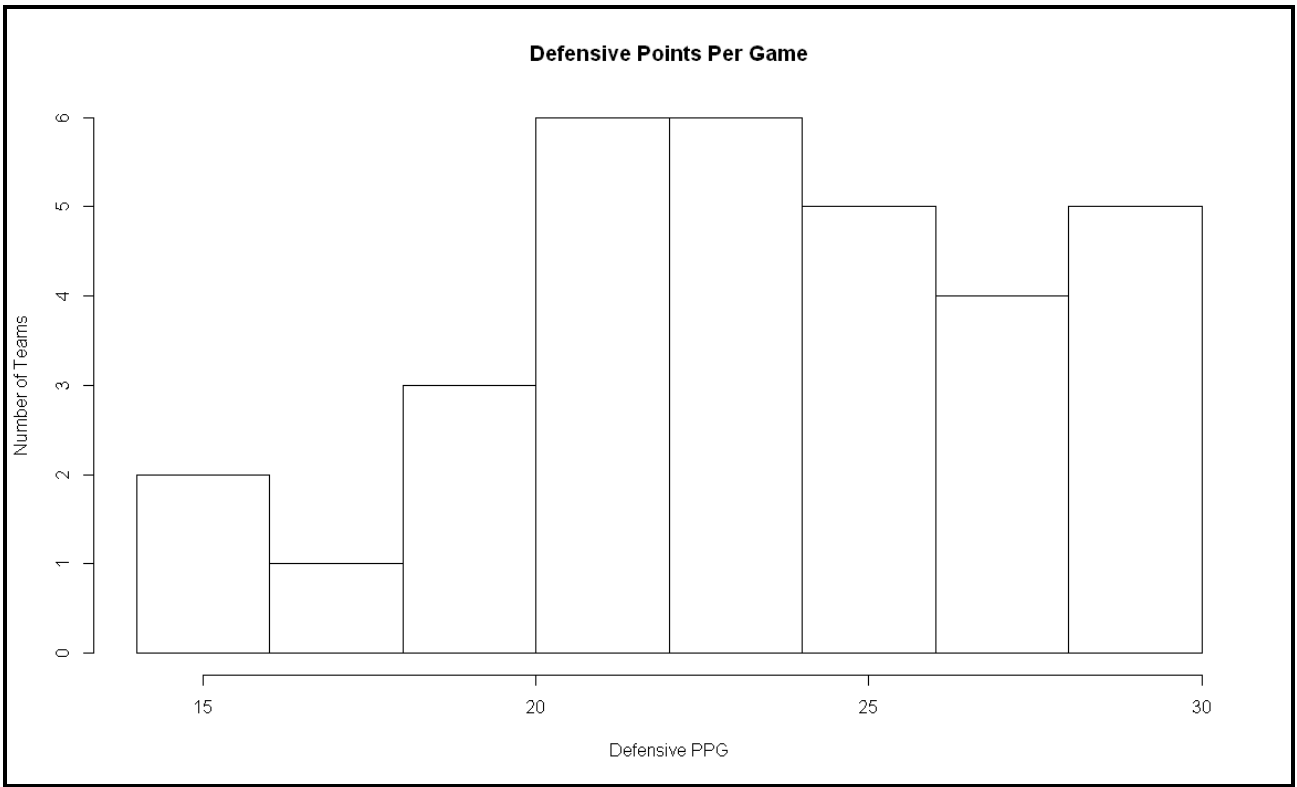
	Team	G	Pts/G	Yds/G	PassYds/G	RushYds/G	1stD/G	3rdM	3rdD%
1	Denver Broncos	16	37.9	457.3	340.3	117.1	27.2	95	46.3
2	Chicago Bears	16	27.8	381.8	267.6	114.3	21.5	83	42.1
3	New England Patriots	16	27.8	384.5	255.4	129.1	23.6	83	37.6
4	Philadelphia Eagles	16	27.6	417.3	256.9	160.4	22.2	83	39.0
5	Dallas Cowboys	16	27.4	341.1	247.1	94.0	20.3	63	35.0
6	Kansas City Chiefs	16	26.9	337.3	208.8	128.5	20.2	74	34.7
7	Cincinnati Bengals	16	26.9	368.2	258.5	109.7	20.6	92	40.9
8	Seattle Seahawks	16	26.1	339.0	202.3	136.8	19.2	76	37.3
9	Green Bay Packers	16	26.1	400.3	266.8	133.5	21.9	89	41.2
10	New Orleans Saints	16	25.9	399.4	307.4	92.1	22.4	97	43.9
11	San Francisco 49ers	16	25.4	323.8	186.2	137.6	17.9	77	36.5
12	San Diego Chargers	16	24.8	393.3	270.5	122.8	23.3	101	49.0
13	Detroit Lions	16	24.7	392.1	280.1	112.0	21.9	95	43.0
14	Minnesota Vikings	16	24.4	344.3	214.2	130.1	19.3	78	36.4

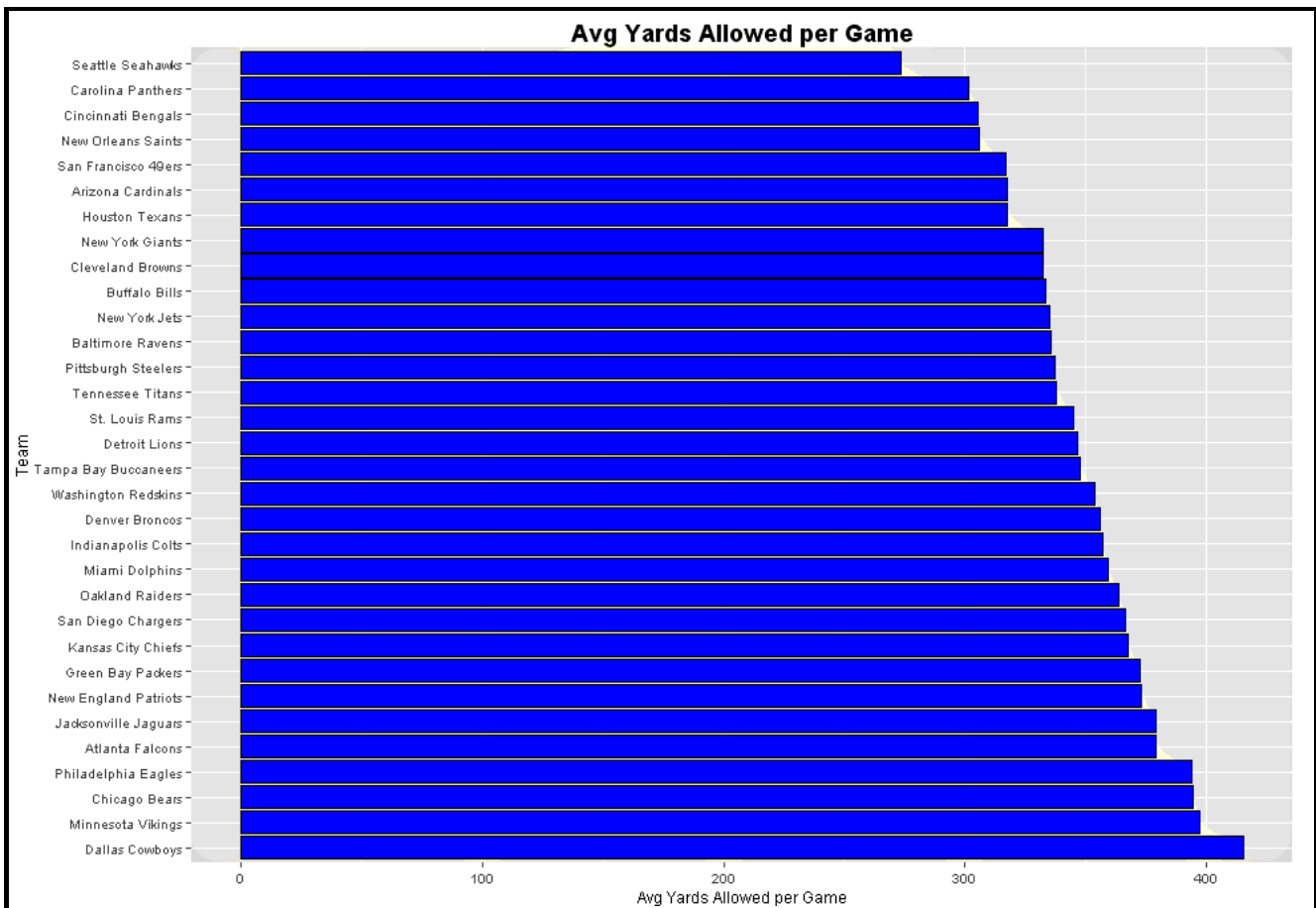
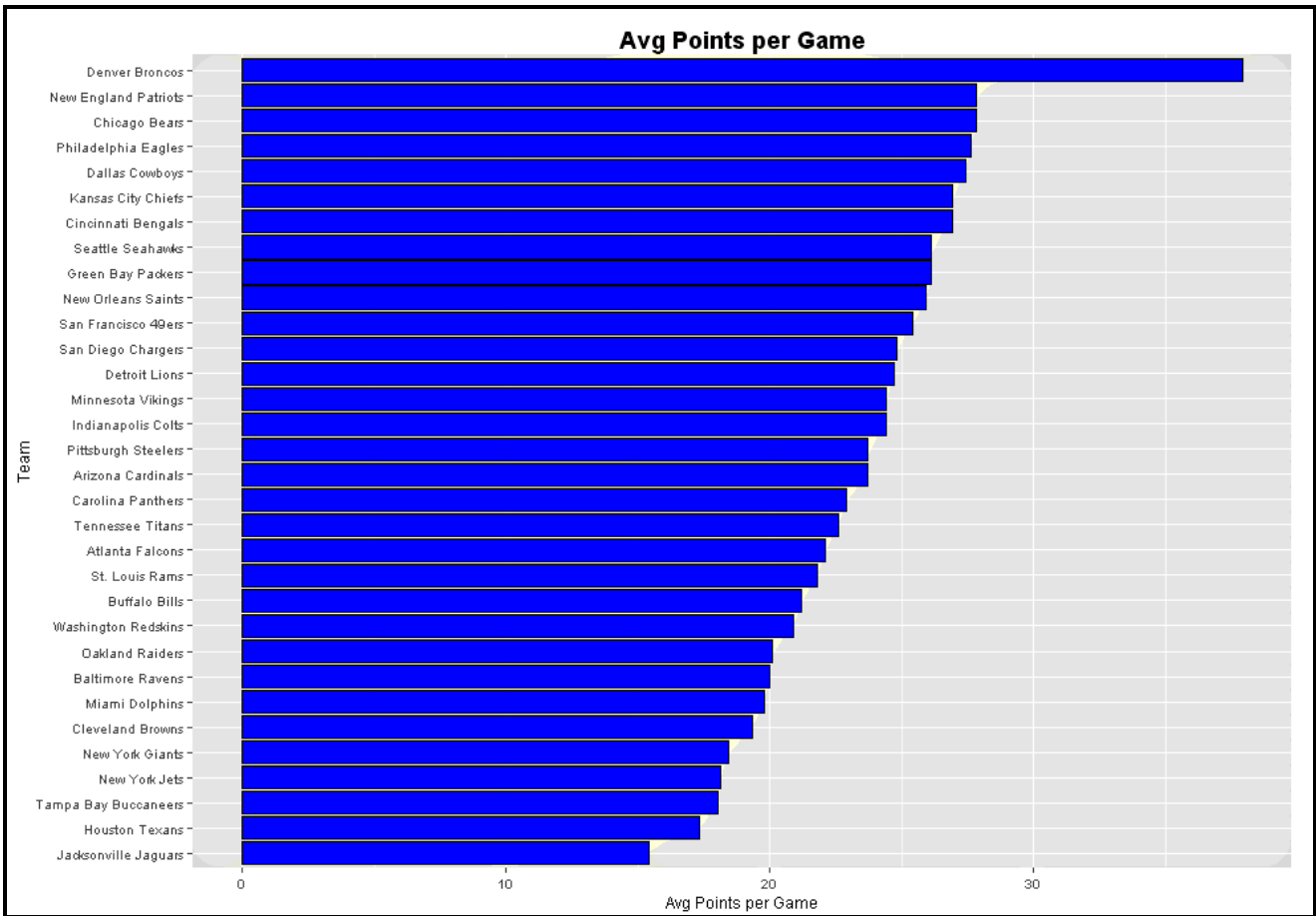
	Team	G	Pts/G	Yds/G	PassYds/G	RushYds/G	1stD/G	3rdM	3rdD%	4thM	4thD%	Pen	PYds
1	Denver Broncos	16	37.9	457.3	340.3	117.1	27.2	95	46.3	8	88.9	117	1000
2	Chicago Bears	16	27.8	381.8	267.6	114.3	21.5	83	42.1	8	61.5	85	708
3	New England Patriots	16	27.8	384.5	255.4	129.1	23.6	83	37.6	7	46.7	69	625
4	Philadelphia Eagles	16	27.6	417.3	256.9	160.4	22.2	83	39.0	7	50.0	95	846
5	Dallas Cowboys	16	27.4	341.1	247.1	94.0	20.3	63	35.0	4	66.7	102	867
6	Kansas City Chiefs	16	26.9	337.3	208.8	128.5	20.2	74	34.7	4	36.4	101	774
7	Cincinnati Bengals	16	26.9	368.2	258.5	109.7	20.6	92	40.9	12	57.1	102	1000
8	Seattle Seahawks	16	26.1	339.0	202.3	136.8	19.2	76	37.3	6	54.5	128	1183
9	Green Bay Packers	16	26.1	400.3	266.8	133.5	21.9	89	41.2	9	69.2	86	801
10	New Orleans Saints	16	25.9	399.4	307.4	92.1	22.4	97	43.9	9	47.4	95	817
11	San Francisco 49ers	16	25.4	323.8	186.2	137.6	17.9	77	36.5	7	63.6	103	845
12	San Diego Chargers	16	24.8	393.3	270.5	122.8	23.3	101	49.0	5	83.3	95	765
13	Detroit Lions	16	24.7	392.1	280.1	112.0	21.9	95	43.0	5	35.7	110	925
14	Minnesota Vikings	16	24.4	344.3	214.2	130.1	19.3	78	36.4	10	52.6	70	695

	Team	G	Pts/G	Yds/G	RushYds/G	PassYds/G	Int	IntTD	FFum
1	Arizona Cardinals	16	20.3	317.4	84.4	233.0	20	4	11
2	Pittsburgh Steelers	16	23.1	337.2	115.6	221.6	10	3	14
3	San Diego Chargers	16	21.8	366.5	107.8	258.7	11	1	9
4	Philadelphia Eagles	16	23.9	394.0	104.3	289.8	19	1	11
5	New York Jets	16	24.2	334.9	88.3	246.7	13	1	9
6	New Orleans Saints	16	19.0	305.7	111.6	194.1	12	0	10
7	New York Giants	16	23.9	332.3	108.9	223.3	17	2	13
8	San Francisco 49ers	16	17.0	316.9	95.9	221.0	18	2	11
9	Seattle Seahawks	16	14.4	273.6	101.6	172.0	28	3	15
10	Baltimore Ravens	16	22.0	335.5	105.4	230.1	16	1	14
11	Houston Texans	16	26.8	317.6	122.4	195.2	7	1	9
12	Jacksonville Jaguars	16	28.1	379.4	131.8	247.6	11	1	14
13	Carolina Panthers	16	15.1	301.3	86.9	214.3	20	4	11

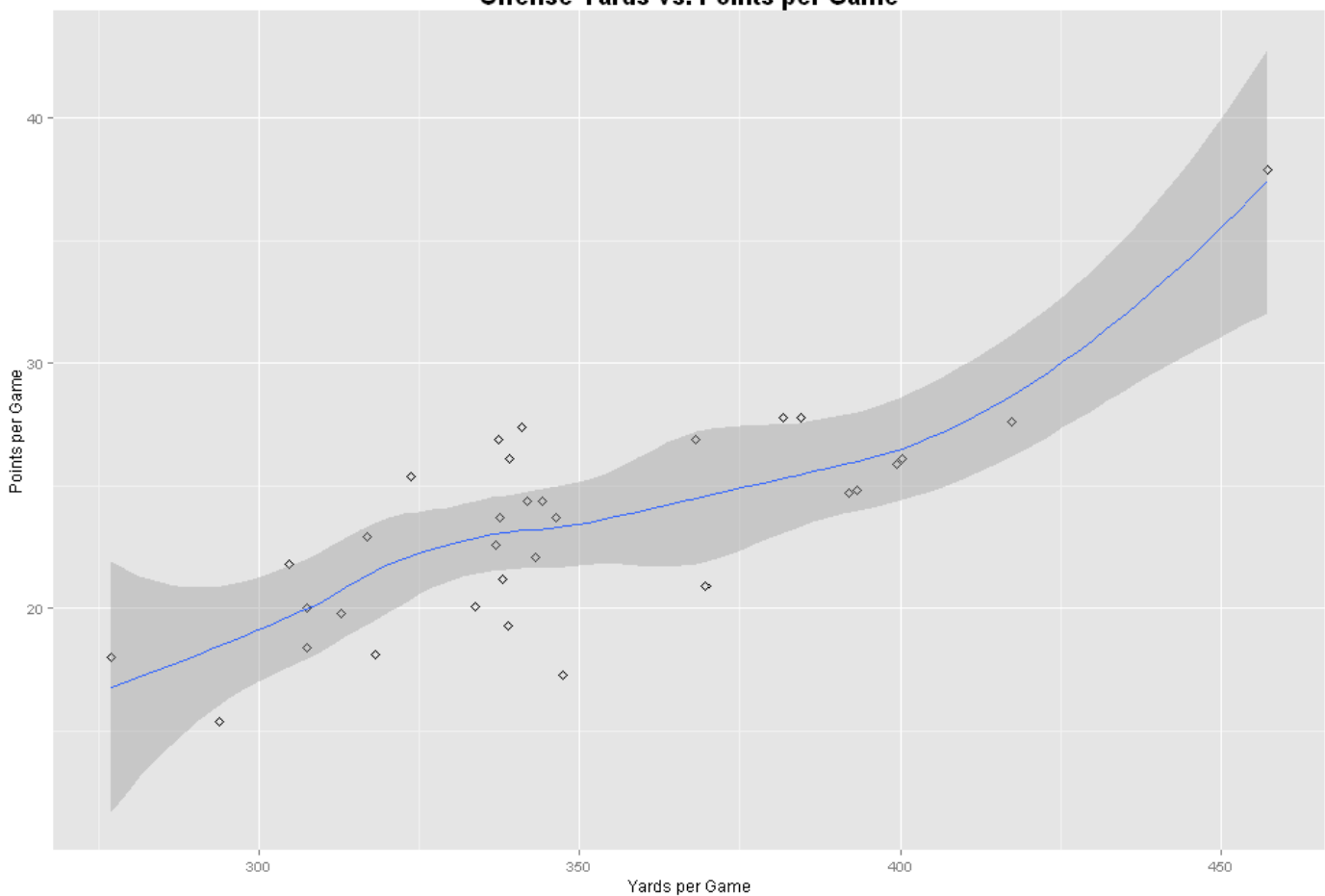
	Team	G	Pts/G	Yds/G	RushYds/G	PassYds/G	Int	IntTD	FFum	DefTD	Tack	PD	Sack
1	Arizona Cardinals	16	20.3	317.4	84.4	233.0	20	4	11	5	686	103	47
2	Pittsburgh Steelers	16	23.1	337.2	115.6	221.6	10	3	14	3	637	72	34
3	San Diego Chargers	16	21.8	366.5	107.8	258.7	11	1	9	1	678	70	35
4	Philadelphia Eagles	16	23.9	394.0	104.3	289.8	19	1	11	2	739	101	37
5	New York Jets	16	24.2	334.9	88.3	246.7	13	1	9	1	599	64	41
6	New Orleans Saints	16	19.0	305.7	111.6	194.1	12	0	10	0	564	69	49
7	New York Giants	16	23.9	332.3	108.9	223.3	17	2	13	2	742	83	34
8	San Francisco 49ers	16	17.0	316.9	95.9	221.0	18	2	11	5	681	87	38
9	Seattle Seahawks	16	14.4	273.6	101.6	172.0	28	3	15	4	575	90	44
10	Baltimore Ravens	16	22.0	335.5	105.4	230.1	16	1	14	1	585	101	40
11	Houston Texans	16	26.8	317.6	122.4	195.2	7	1	9	1	640	71	32
12	Jacksonville Jaguars	16	28.1	379.4	131.8	247.6	11	1	14	2	703	72	31
13	Carolina Panthers	16	15.1	301.3	86.9	214.3	20	4	11	4	615	67	60
14	Tampa Bay Buccaneers	16	24.3	348.0	110.1	237.9	21	3	13	3	676	75	35



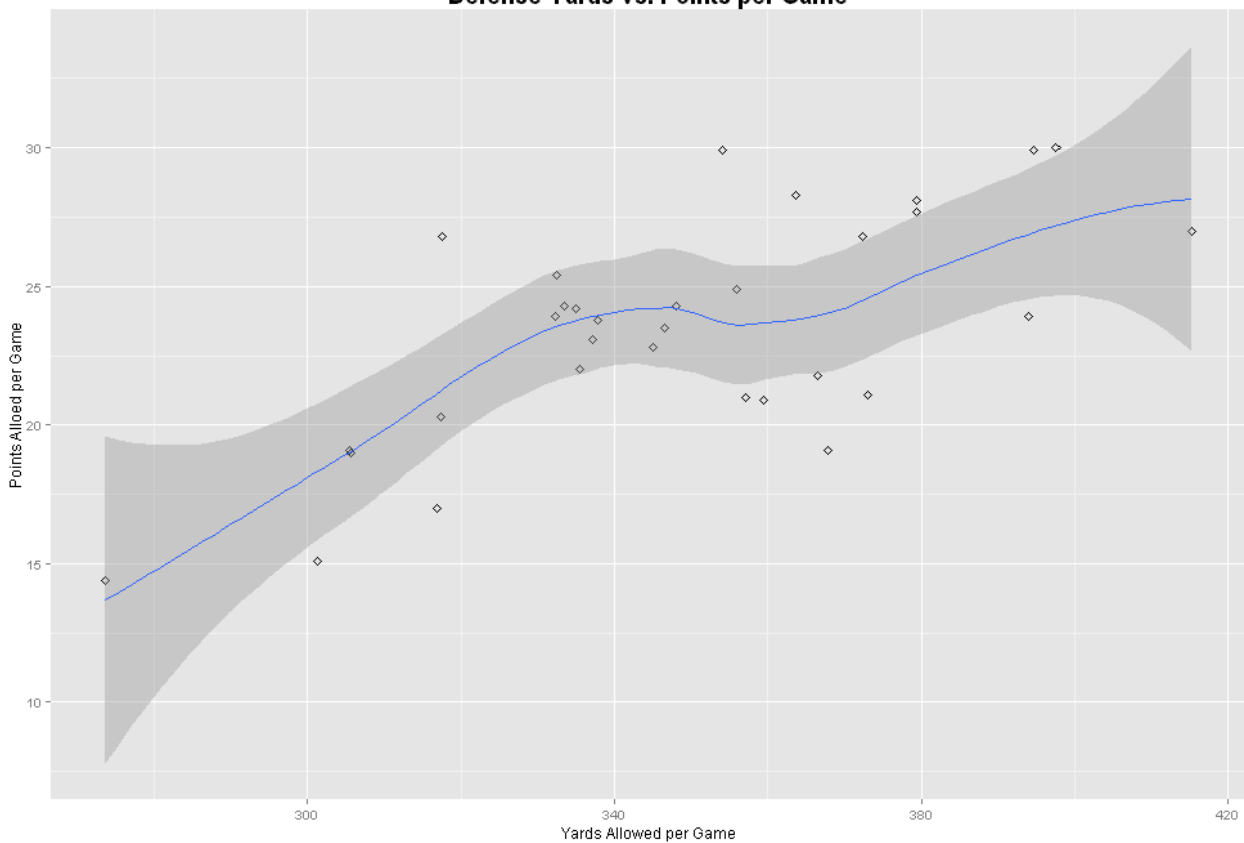


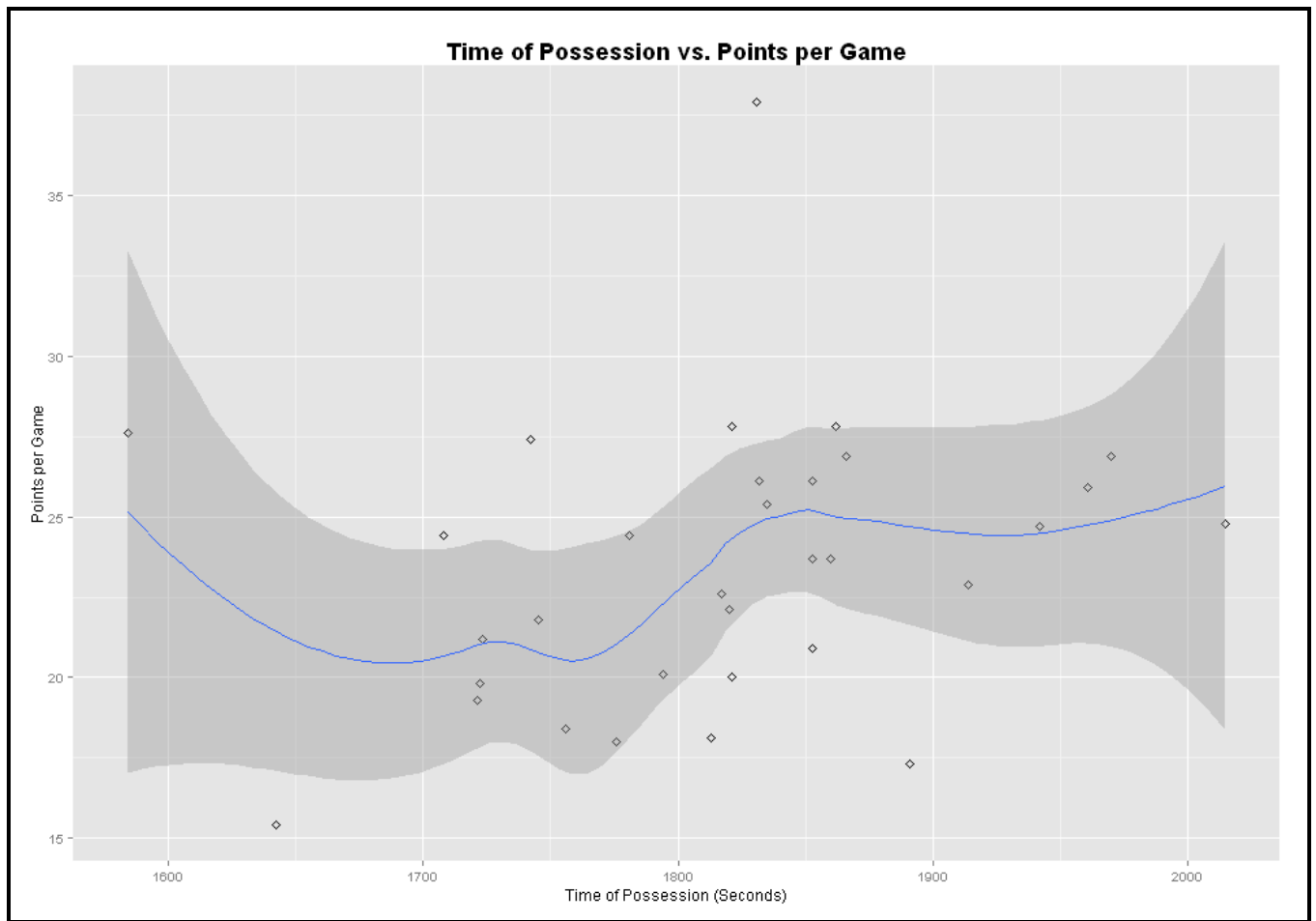


Offense Yards vs. Points per Game



Defense Yards vs. Points per Game





	Team	OPassStrength	ORushStrength	OffStrength	DPassStrength	DRushStrength	DefStrength
1	Chicago Bears	55.67073	44.12121	53.25710	48.13243	0.00000	15.84545
2	New Orleans Saints	79.93902	17.21212	52.92617	81.23939	64.67532	73.44351

	Team	OPassStrength	ORushStrength	OffStrength	DPassStrength	DRushStrength	DefStrength	Net_Pass
1	Chicago Bears	55.67073	44.12121	53.25710	48.13243	0.00000	15.84545	-25.56866
2	New Orleans Saints	79.93902	17.21212	52.92617	81.23939	64.67532	73.44351	31.80660

	Team	OPassStrength	ORushStrength	OffStrength	DPassStrength	DRushStrength	DefStrength	Net_Pass	Net_Rush	Net_Total
1	Chicago Bears	55.67073	44.12121	53.25710	48.13243	0.00000	15.84545	-25.56866	-20.55411	-20.18641
2	New Orleans Saints	79.93902	17.21212	52.92617	81.23939	64.67532	73.44351	31.80660	17.21212	37.00072

	Week	HomeTeam	AwayTeam
1	1	San Diego Chargers	Denver Broncos
2	1	Indianapolis Colts	Chicago Bears
3	1	San Francisco 49ers	Philadelphia Eagles
4	1	Tennessee Titans	Dallas Cowboys
5	1	Baltimore Ravens	Cincinnati Bengals
6	1	Detroit Lions	Seattle Seahawks
7	1	New England Patriots	Green Bay Packers
8	1	Minnesota Vikings	Pittsburgh Steelers
9	1	Washington Redskins	Carolina Panthers
10	1	Atlanta Falcons	St. Louis Rams
11	1	Tampa Bay Buccaneers	Buffalo Bills
12	1	New Orleans Saints	Oakland Raiders
13	1	Kansas City Chiefs	Miami Dolphins
14	1	Arizona Cardinals	New York Giants

	Team	Wins	Losses
1	Denver Broncos	0	0
2	Chicago Bears	0	0
3	New England Patriots	0	0
4	Philadelphia Eagles	0	0
5	Dallas Cowboys	0	0
6	Kansas City Chiefs	0	0
7	Cincinnati Bengals	0	0
8	Seattle Seahawks	0	0
9	Green Bay Packers	0	0
10	New Orleans Saints	0	0
11	San Francisco 49ers	0	0
12	San Diego Chargers	0	0
13	Detroit Lions	0	0
14	Minnesota Vikings	0	0

	Winner	Loser
1	Denver Broncos	San Diego Chargers
2	Indianapolis Colts	Chicago Bears
3	San Francisco 49ers	Philadelphia Eagles
4	Tennessee Titans	Dallas Cowboys
5	Cincinnati Bengals	Baltimore Ravens
6	Seattle Seahawks	Detroit Lions
7	Green Bay Packers	New England Patriots
8	Pittsburgh Steelers	Minnesota Vikings
9	Carolina Panthers	Washington Redskins
10	St. Louis Rams	Atlanta Falcons
11	Buffalo Bills	Tampa Bay Buccaneers
12	New Orleans Saints	Oakland Raiders
13	Kansas City Chiefs	Miami Dolphins

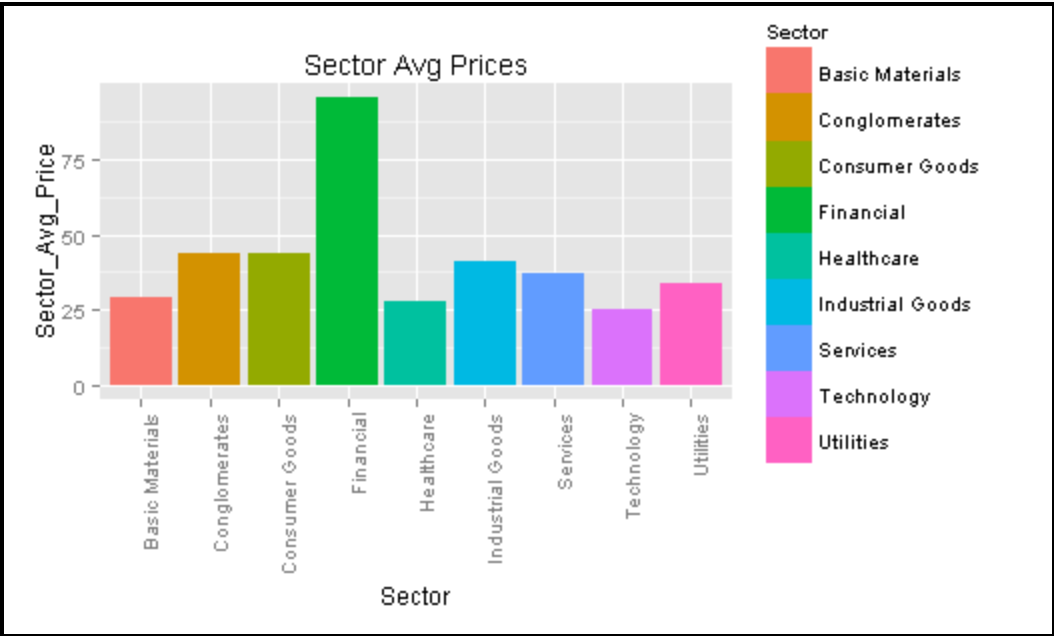
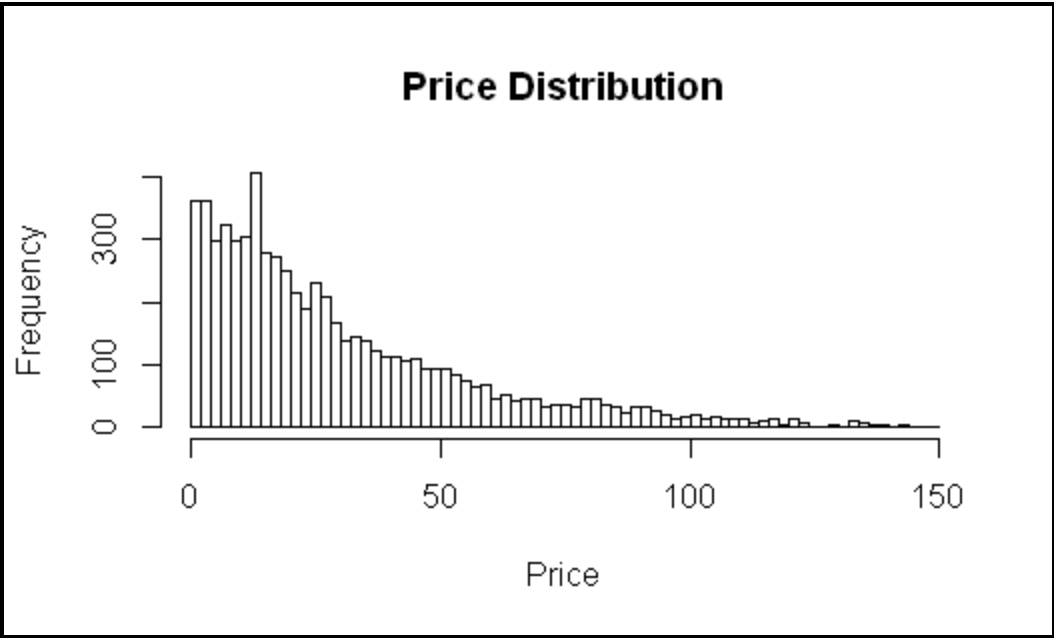
	row.names	Team	Wins	Losses
1	8	Seattle Seahawks	50	0
2	1	Denver Broncos	49	1
3	10	New Orleans Saints	47	3
4	13	Detroit Lions	45	5
5	7	Cincinnati Bengals	44	6
6	4	Philadelphia Eagles	43	7
7	18	Carolina Panthers	43	7
8	11	San Francisco 49ers	42	8
9	16	Arizona Cardinals	35	15
10	12	San Diego Chargers	33	17
11	22	Buffalo Bills	32	18
12	9	Green Bay Packers	31	19
13	31	Houston Texans	31	19

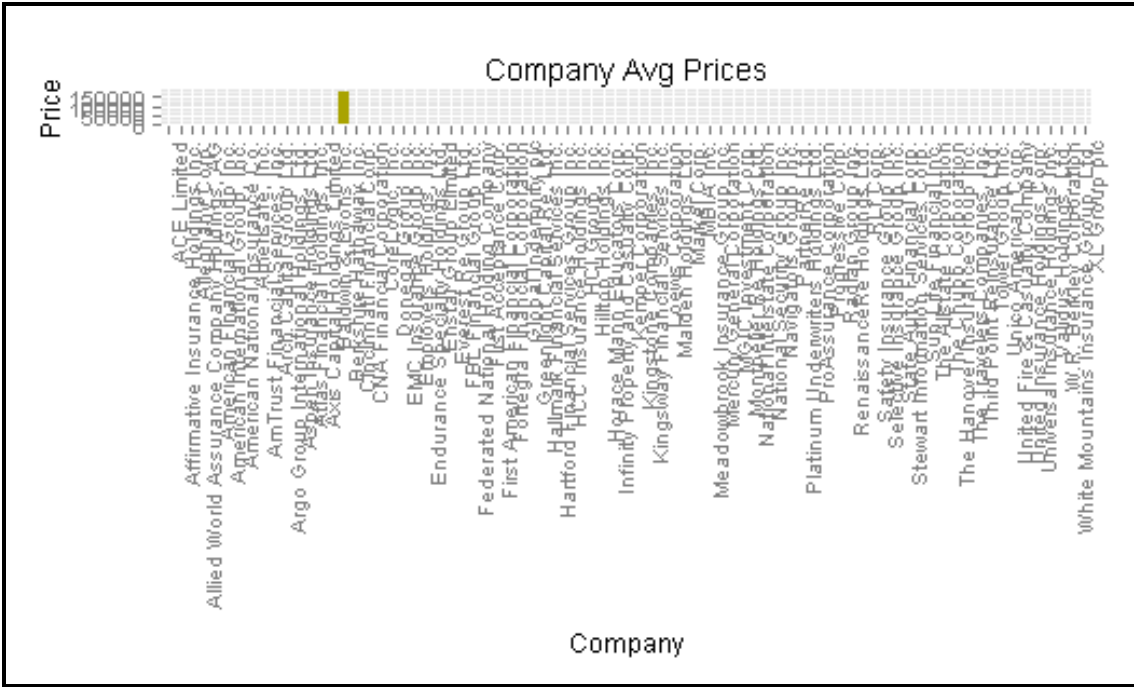
Team	Wins	Losses
Denver Broncos	13	3
Seattle Seahawks	13	3
New England Patriots	12	4
Carolina Panthers	12	4
San Francisco 49ers	12	4
Cincinnati Bengals	11	5
Indianapolis Colts	11	5
Kansas City Chiefs	11	5
New Orleans Saints	11	5
Philadelphia Eagles	10	6

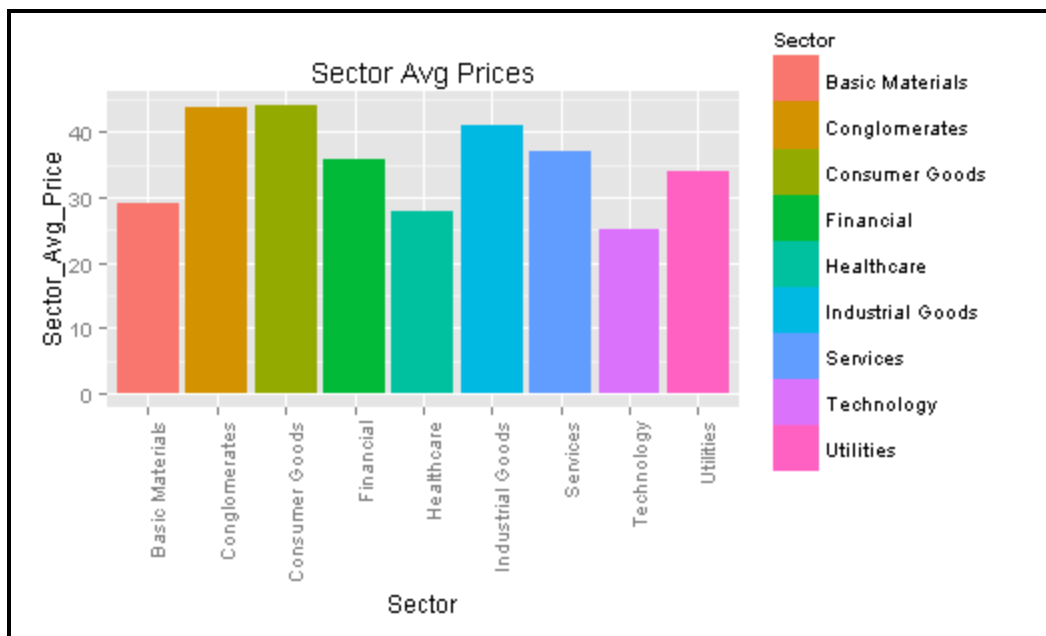
Chapter 4: Modeling Stock Market Data (R)

Home News Screener Maps Groups Portfolio Insider Futures Forex Collaborate Store Elite <small>NEW</small> Search ticker, company or profile													
My Presets		Order: Ticker	Asc	Signal: None (all stocks)	Tickers:		Filters						
Filters: 0													
Exchange	Any	Index	Any	Sector	Any	Industry	Any	Country	Any				
Market Cap.	Any	Dividend Yield	Any	Float Short	Any	Analyst Recom.	Any	Option/Short	Any				
Earnings Date	Any	Average Volume	Any	Relative Volume	Any	Current Volume	Any	Price	Any				
Reset (0)													
Overview	Valuation	Financial	Ownership	Performance	Technical	Custom	Charts	Tickers	Quotes	Basic	TA	News	Snapshot
Total: 6706 #1 save as portfolio Page 1/336													
No.	Ticker	Company	Sector	Industry	Country	Market Cap	P/E	Price	Change	Volume			
1	A	Agilent Technologies Inc.	Healthcare	Medical Laboratories & Research	USA	19.51B	28.06	58.93	0.89%	1,623,230			
2	AA	Alcoa, Inc.	Basic Materials	Aluminum	USA	10.82B	40.44	10.11	-5.43%	74,458,744			
3	AADR	WCM/BNY Mellon Focused Growth ADR ETF	Financial	Exchange Traded Fund	USA	-	-	37.45	0.19%	4,113			
4	AAIT	iShares MSCI AC Asia Information Tech	Financial	Exchange Traded Fund	USA	-	-	32.22	4.47%	1,814			
5	AAL	American Airlines Group Inc.	Services	Major Airlines	USA	9.86B	25.97	29.35	-0.24%	12,806,994			
6	AAMC	Altisource Asset Management Corporation	Financial	Asset Management	USA	2.41B	-	1024.00	4.06%	12,062			
7	AAME	Atlantic American Corp.	Financial	Life Insurance	USA	85.62M	8.93	4.02	-0.25%	14,397			
8	AAN	Aaron's, Inc.	Services	Rental & Leasing Services	USA	2.20B	16.46	28.97	0.42%	976,533			
9	AAOI	Applied Optoelectronics, Inc.	Technology	Semiconductor - Integrated Circuits	USA	176.46M	-	14.00	-1.13%	48,220			
10	AAON	AAON Inc.	Industrial Goods	General Building Materials	USA	1.15B	34.91	31.42	-0.48%	100,471			
11	AAP	Advance Auto Parts Inc.	Services	Auto Parts Stores	USA	8.41B	20.87	115.64	1.84%	678,570			
12	AAPL	Apple Inc.	Consumer Goods	Electronic Equipment	USA	481.86B	13.45	532.94	-0.67%	10,884,522			
13	AAT	American Assets Trust, Inc.	Financial	REIT - Office	USA	1.27B	639.40	31.97	2.24%	75,899			
14	AAU	Almaden Minerals Ltd.	Basic Materials	Gold	Canada	72.06M	-	1.20	-1.64%	67,248			
15	AAV	Advantage Oil & Gas Ltd.	Basic Materials	Oil & Gas Drilling & Exploration	Canada	734.15M	-	4.36	2.11%	129,611			
16	AAWW	Atlas Air Worldwide Holdings Inc.	Services	Air Services, Other	USA	1.08B	9.66	43.00	2.58%	265,005			









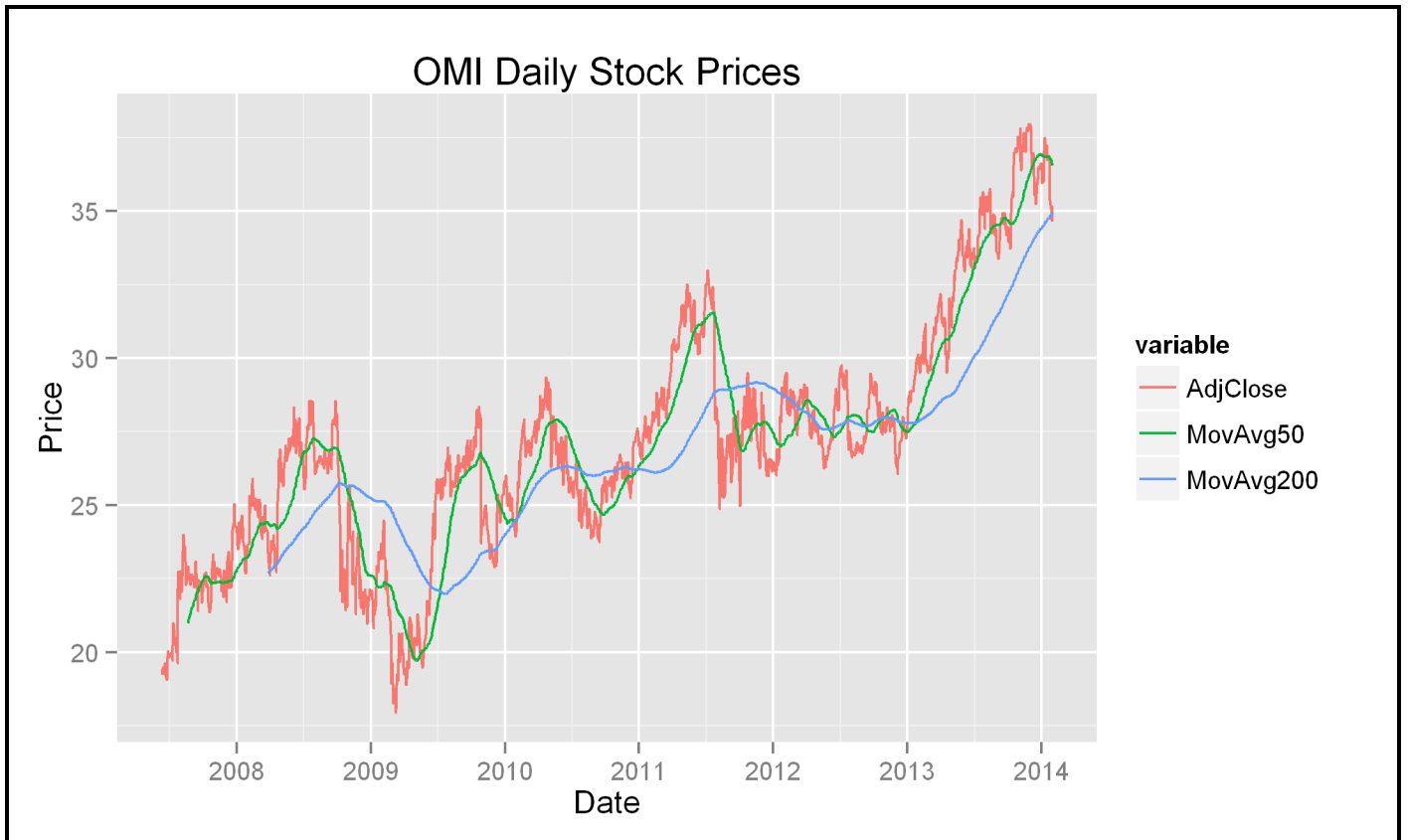
Sector	Industry	Country	Market.Cap	P.E	Forward.P.E	PEG	P.S	P.B	P.Cash
Healthcare	Medical Laboratories & Research	USA	19823.59	28.52	16.78	2.97	2.92	3.75	7.41
Basic Materials	Aluminum	USA	12062.38	NA	19.33	NA	0.52	1.15	8.39
Financial	Exchange Traded Fund	USA	NA	NA	NA	NA	NA	NA	NA
Financial	Exchange Traded Fund	USA	NA	NA	NA	NA	NA	NA	NA
Services	Major Airlines	USA	11645.76	30.67	6.61	0.87	0.46	NA	1.72
Financial	Asset Management	USA	2348.00	NA	NA	NA	78.27	671.14	23.00
Financial	Life Insurance	USA	83.71	8.73	NA	NA	0.52	0.86	2.24
Services	Rental & Leasing Services	USA	2297.49	17.15	15.81	2.34	1.02	1.85	7.43
Technology	Semiconductor - Integrated Circuits	USA	159.82	NA	13.24	NA	2.17	NA	21.89
Industrial Goods	General Building Materials	USA	1035.39	31.31	24.94	3.13	3.18	6.39	47.28
Services	Auto Parts Stores	USA	9090.47	22.56	15.17	1.46	1.42	6.19	16.02
Consumer Goods	Electronic Equipment	USA	480222.89	13.29	11.60	0.68	2.76	3.70	11.80
Financial	REIT - Office	USA	1296.44	651.20	95.76	132.90	5.11	1.99	19.73
Basic Materials	Gold	Canada	99.68	NA	NA	NA	332.28	2.27	6.83
Basic Materials	Oil & Gas Drilling & Exploration	Canada	520.70	NA	NA	NA	2.19	0.53	NA
Services	Air Services, Other	USA	789.65	7.06	8.05	3.07	0.48	0.62	2.70
Financial	Exchange Traded Fund	USA	NA	NA	NA	NA	NA	NA	NA
Financial	Asset Management	USA	2176.37	17.35	13.70	2.17	14.53	1.53	NA
Healthcare	Diagnostic Substances	USA	857.43	50.66	41.31	2.89	4.77	4.59	8.41
Industrial Goods	Industrial Equipment & Components	Switzerland	58963.99	20.54	15.88	1.77	1.42	3.35	12.61
Healthcare	Drug Manufacturers - Major	USA	81300.48	19.67	14.20	1.47	4.33	22.77	8.47
Services	Drugs Wholesale	USA	15598.96	43.42	15.76	3.13	0.16	6.94	44.89
Financial	Regional - Mid-Atlantic Banks	USA	480.18	26.65	10.90	3.33	3.80	1.66	1.69
Services	Education & Training Services	USA	91.80	NA	NA	NA	0.61	NA	1.71
Services	Business Services	USA	2150.47	93.64	41.22	5.70	4.41	6.95	27.61
Services	Trucking	USA	828.76	54.47	12.45	5.45	0.36	1.59	5.87
Services	Auto Dealerships	USA	1469.79	14.83	10.95	0.95	0.28	3.12	1130.61

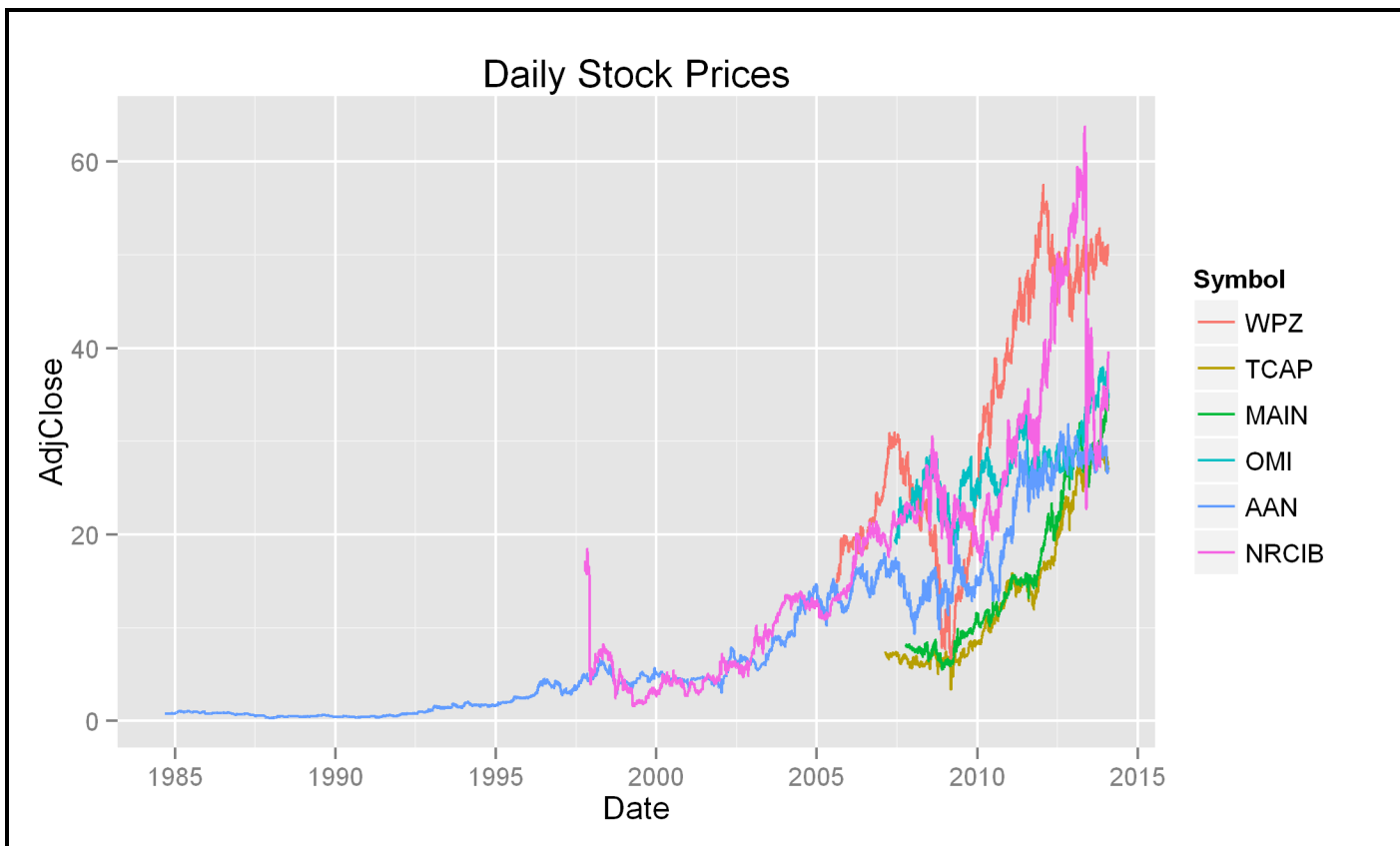
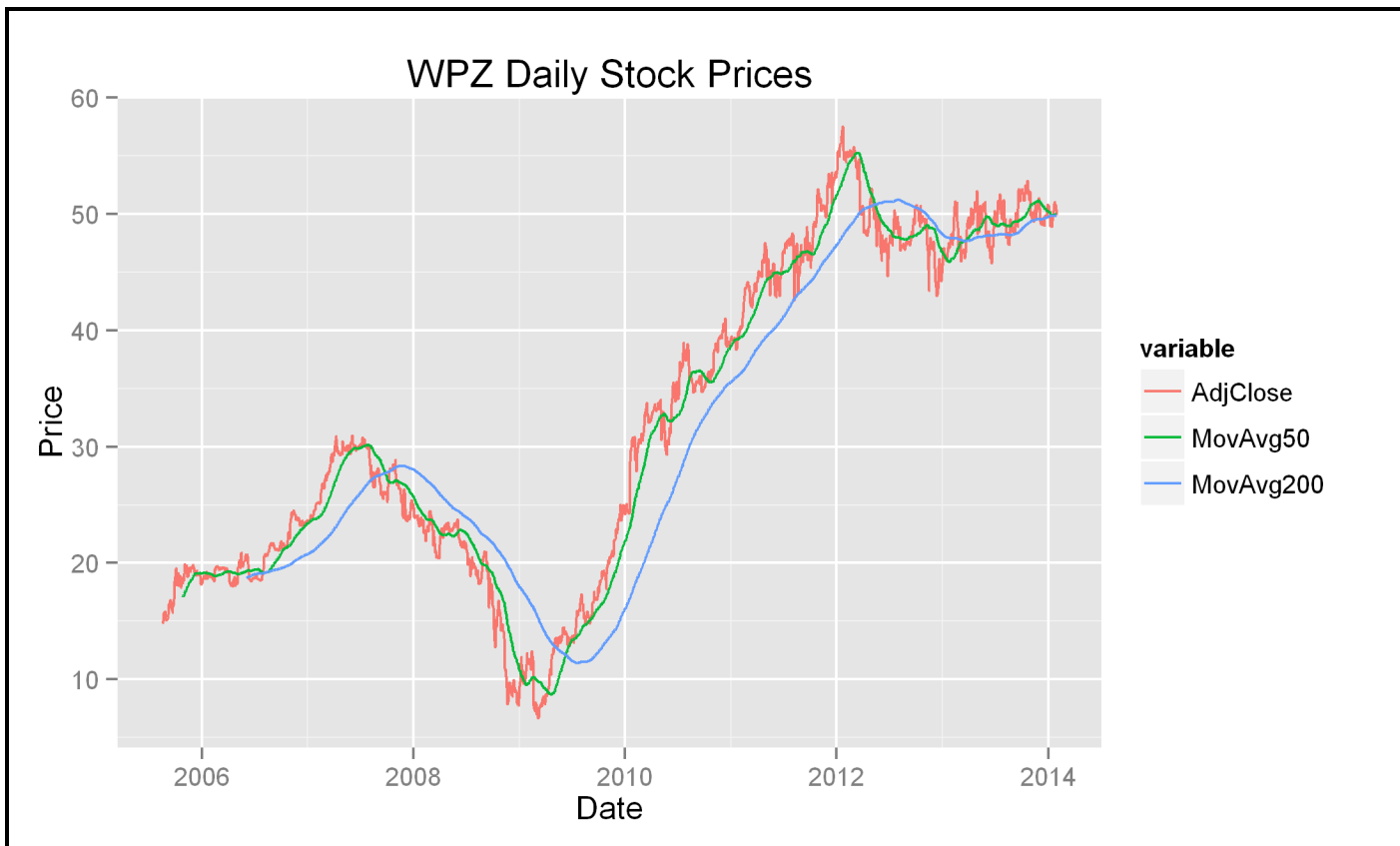
Sector	variable	value
Healthcare	P.E	28.52
Services	P.E	30.67
Financial	P.E	8.73
Services	P.E	17.15
Industrial Goods	P.E	31.31
Services	P.E	22.56
Consumer Goods	P.E	13.29
Financial	P.E	651.2
Services	P.E	7.06
Financial	P.E	17.35
Healthcare	P.E	50.66
Industrial Goods	P.E	20.54
Healthcare	P.E	19.67
Services	P.E	43.42
Financial	P.E	26.65

	Sector	SAvgPE	SAvgPEG	SAvgPS	SAvgPB	SAvgPrice
1	Basic Materials	42.87945	5.390194	35.677311	10.203838	29.22257
2	Conglomerates	20.79571	1.045000	1.532000	58.426316	40.11000
3	Consumer Goods	30.29197	3.446652	1.380000	4.712809	42.40154
4	Financial	32.88929	5.403305	12.335628	4.465120	35.38289
5	Healthcare	38.44733	12.175091	184.600614	9.349106	27.94912
6	Industrial Goods	32.73892	3.314206	1.856246	3.765014	40.78930
7	Services	44.43990	3.927596	1.992289	33.536609	36.45865
8	Technology	59.85766	4.749591	9.386424	4.697576	24.87183
9	Utilities	27.20184	97.133068	7.979917	2.030339	34.01273

	row.names	Ticker	Company	RelValIndex
1	1	CGA	China Green Agriculture, Inc.	10
2	4	UAIL	CVR Partners, LP	8
3	7	YONG	Yongye International, Inc.	8
4	8	AVD	American Vanguard Corp.	9
5	16	MOS	The Mosaic Company	8
6	27	ARSD	Arabian American Development Company	8
7	32	CE	Celanese Corporation	8
8	33	LIND	Landec Corp.	10
9	36	ASH	Ashland Inc.	8
10	38	ACET	Aceto Corp.	10
11	40	DOW	The Dow Chemical Company	8
12	42	SQM	Chemical & Mining Co. of Chile Inc.	8
13	43	FF	FutureFuel Corp.	10

Sector	Industry	No.	Ticker	Company	Country
Basic Materials	Specialty Chemicals	6538	WPZ	Williams Partners L.P.	USA
Financial	Credit Services	5786	TCAP	Triangle Capital Corporation	USA
Financial	Diversified Investments	3694	MAII	Main Street Capital Corporation	USA
Services	Medical Equipment Wholesale	4434	OMI	Owens & Minor Inc.	USA
Services	Rental & Leasing Services	8	AAR	Aaron's, Inc.	USA
Services	Research Services	4257	HRCIB	National Research Corp.	USA







Chapter 5: Visually Exploring Employment Data (R)

```

Console D:/Github/practical-data-science/two/ ↵
> head(ann2012)
  area_fips own_code industry_code agglvl_code size_code year qtr disclosure_code
1:    01000      0         10          50         0 2012  A
2:    01000      1         10          51         0 2012  A
3:    01000      1        102          52         0 2012  A
4:    01000      1       1021          53         0 2012  A
5:    01000      1       1022          53         0 2012  A
6:    01000      1       1023          53         0 2012  A
  annual_avg_estabs_count annual_avg_emplvl total_annual_wages taxable_annual_wages
1:                116233          1828248  3.792883e-313      6.632697e-314
2:                 1252           56031  2.072269e-314      0.000000e+00
3:                 1252           56031  2.072269e-314      0.000000e+00
4:                  599          11734  3.555500e-315      0.000000e+00
5:                   2             13  2.155134e-318      0.000000e+00
6:                   17            161  6.053830e-317      0.000000e+00
  annual_contributions annual_avg_wkly_wage avg_annual_pay
1:      2.07203e-315          808          41990
2:      0.00000e+00          1440          74857
3:      0.00000e+00          1440          74857
4:      0.00000e+00          1179          61330
5:      0.00000e+00           662          34437
6:      0.00000e+00          1468          76343
> |

```

```

> ann2012 <- fread('data/2012.annual.singlefile.csv', sep=',',
+                 colClasses=c('character','integer','integer','integer','integer',
+                               'integer','character',rep('integer',8)))
Read 3556289 rows and 15 (of 15) columns from 0.191 GB file in 00:00:04
Warning message:
In fread("data/2012.annual.singlefile.csv", sep = ",", colClasses = c("character", :
Some columns have been read as type 'integer64' but package bit64 isn't loaded. Those column
s will display as strange looking floating point data. There is no need to reload the data. Ju
st require(bit64) to obtain the integer64 print method and print the data again.

```

```

> head(ann2012)
  area_fips own_code industry_code agglvl_code size_code year qtr disclosure_code
1:    01000      0         10          50         0 2012  A
2:    01000      1         10          51         0 2012  A
3:    01000      1        102          52         0 2012  A
4:    01000      1       1021          53         0 2012  A
5:    01000      1       1022          53         0 2012  A
6:    01000      1       1023          53         0 2012  A
  annual_avg_estabs_count annual_avg_emplvl total_annual_wages taxable_annual_wages
1:                116233          1828248  76768801894      13424728725
2:                 1252           56031  4194319351           0
3:                 1252           56031  4194319351           0
4:                  599          11734  719641114           0
5:                   2             13    436204           0
6:                   17            161  12253089           0
  annual_contributions annual_avg_wkly_wage avg_annual_pay
1:      419383612          808          41990
2:              0          1440          74857
3:              0          1440          74857
4:              0          1179          61330
5:              0           662          34437
6:              0          1468          76343
> |

```

```

> intersect(names(agglvl),names(ann2012))
[1] "agglvl_code"
> intersect(names(industry), names(ann2012))
[1] "industry_code"
> intersect(names(area), names(ann2012))
[1] "area_fips"
> intersect(names(ownership),names(ann2012))
[1] "own_code"
> intersect(names(size), names(ann2012))
[1] "size code"

```

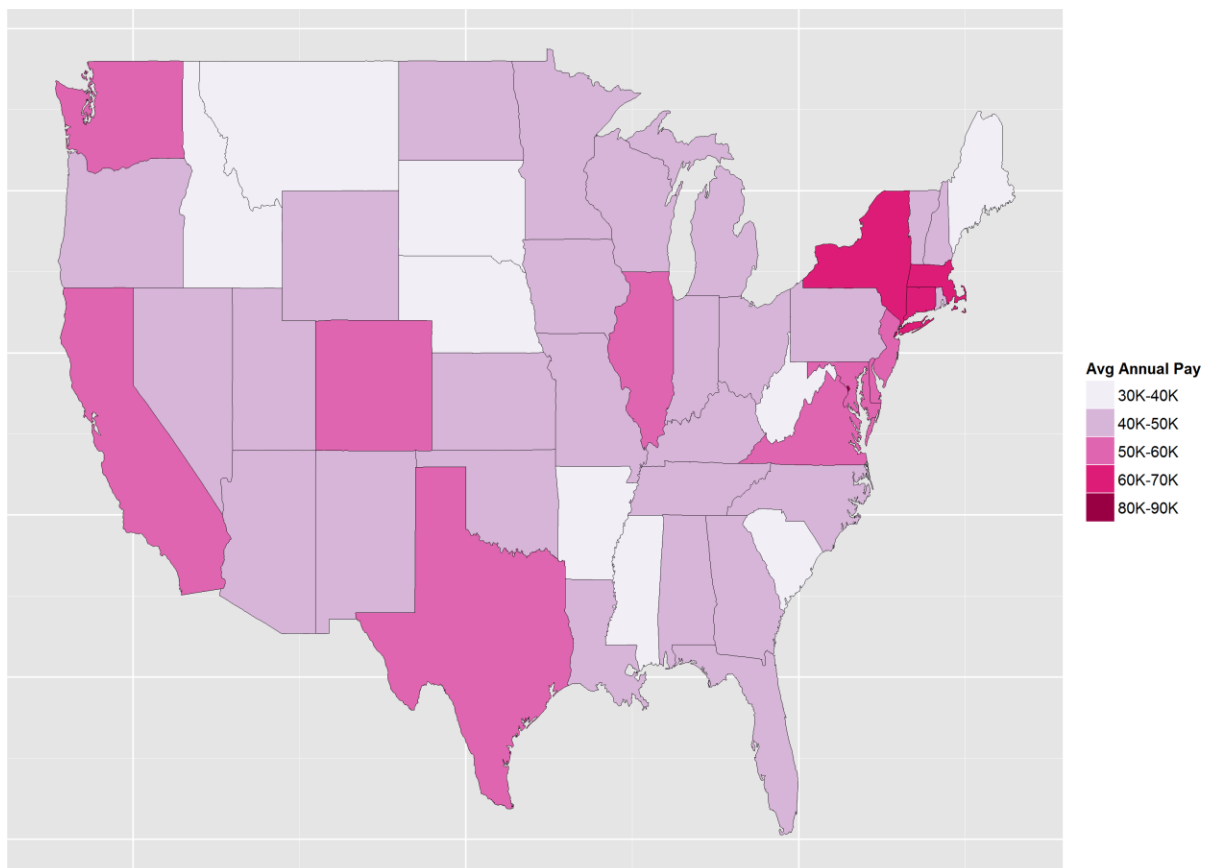
```

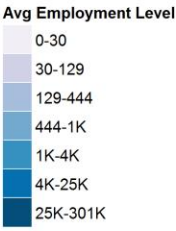
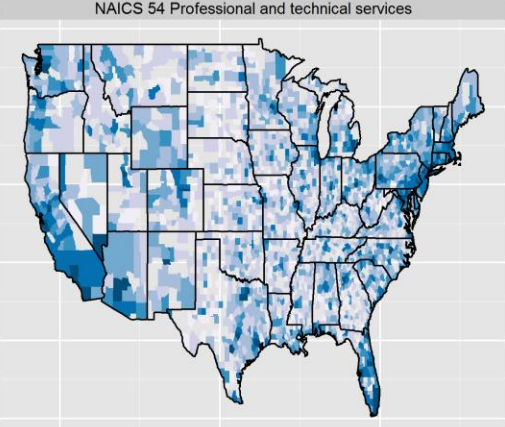
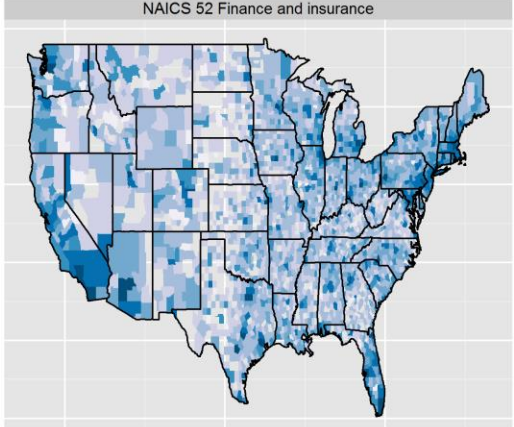
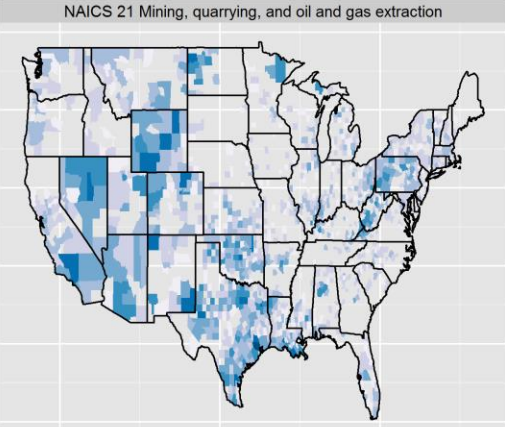
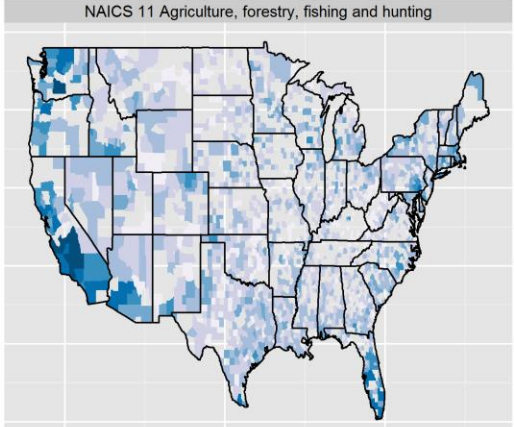
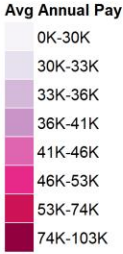
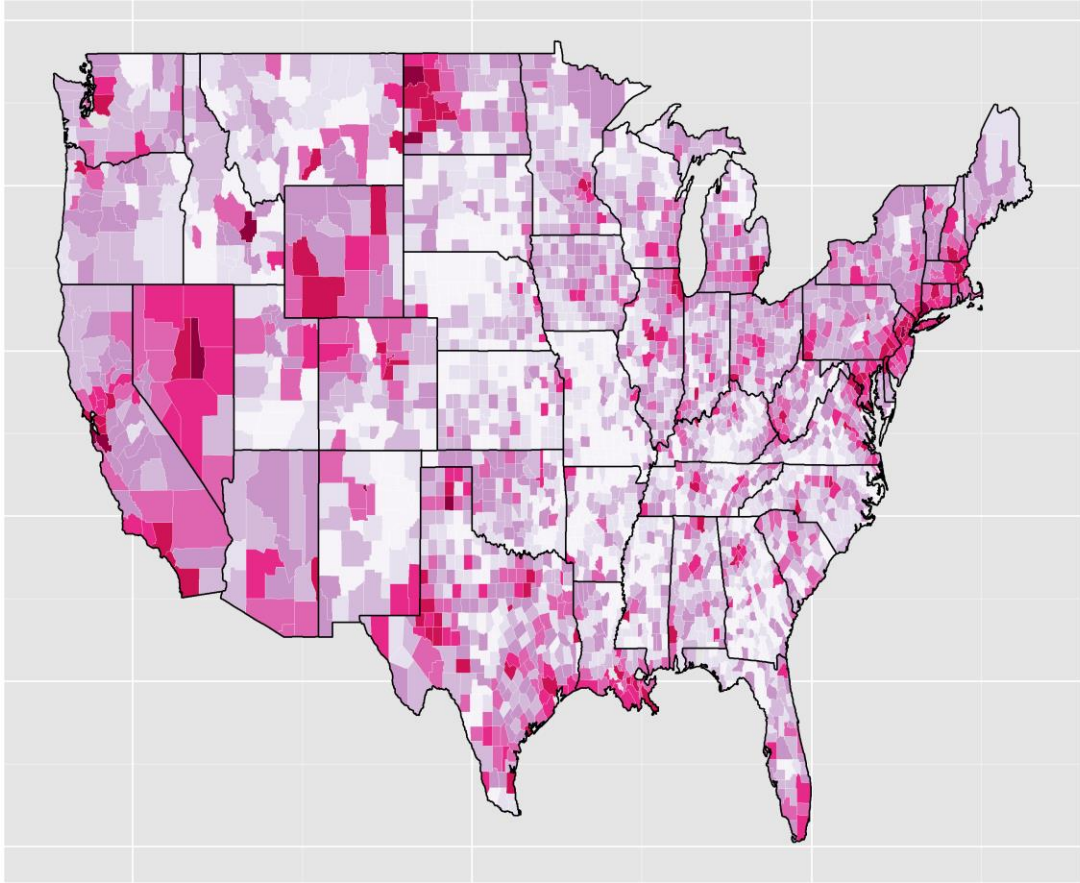
> head(ann2012full)
 area_fips own_code industry_code agglvl_code size_code year qtr disclosure_code
1      01000      0          10          50         0 2012  A
2      01000      1          10          51         0 2012  A
3      01000      1         102          52         0 2012  A
4      01000      1        1021          53         0 2012  A
5      01000      1        1022          53         0 2012  A
6      01000      1        1023          53         0 2012  A
 annual_avg_estabs_count annual_avg_emplvl total_annual_wages taxable_annual_wages
1                116233          1828248          76768801894          13424728725
2                 1252           56031           4194319351              0
3                 1252           56031           4194319351              0
4                  599           11734           719641114              0
5                   2             13             436204              0
6                   17            161           12253089              0
 annual_contributions annual_avg_wkly_wage avg_annual_pay
1          419383612             808           41990
2                   0             1440           74857
3                   0             1440           74857
4                   0             1179           61330
5                   0              662           34437
6                   0             1468           76343
                                agglvl_title                                industry_title
1                                State, Total Covered                        Total, all industries
2      State, Total -- by ownership sector                                Total, all industries
3      State, by Domain -- by ownership sector                            Service-providing
4 State, by Supersector -- by ownership sector Trade, transportation, and utilities
5 State, by Supersector -- by ownership sector                                Information
6 State, by Supersector -- by ownership sector                            Financial activities
                                own_title                                size_title
1      Total Covered All establishment sizes
2 Federal Government All establishment sizes
3 Federal Government All establishment sizes
4 Federal Government All establishment sizes
5 Federal Government All establishment sizes
6 Federal Government All establishment sizes

```

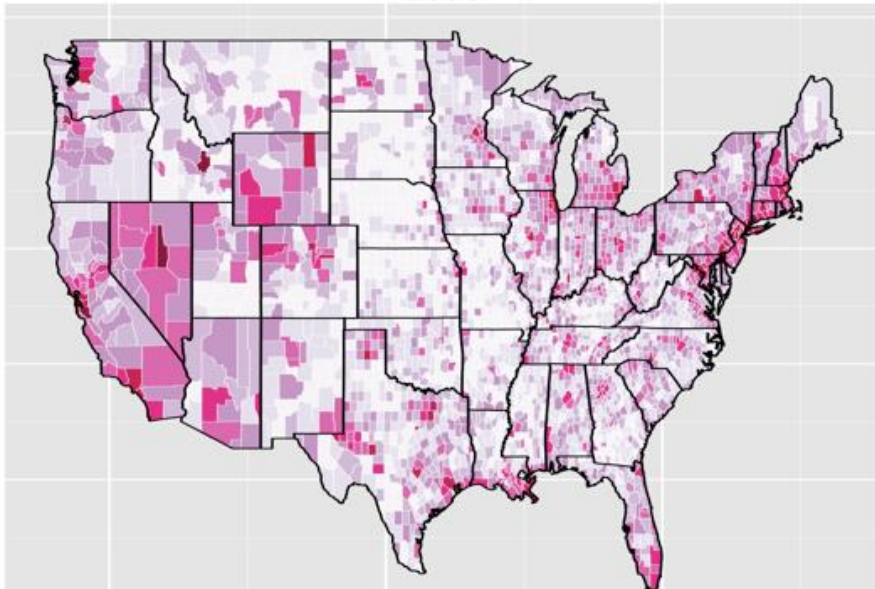
```
> head(area)
  area_fips          area_title
1    US000          U.S. TOTAL
2    USCMS    U.S. Combined Statistical Areas (combined)
3    USMSA U.S. Metropolitan Statistical Areas (combined)
4    USNMS U.S. Nonmetropolitan Area Counties (combined)
5    01000          Alabama -- Statewide
6    01001      Autauga County, Alabama
```

```
> head(state.fips)
  fips ssa region division abb  polynome
1    1  1  3      6    AL    alabama
2    4  3  4      8    AZ    arizona
3    5  4  3      7    AR    arkansas
4    6  5  4      9    CA    california
5    8  6  4      8    CO    colorado
6    9  7  1      1    CT    connecticut
```

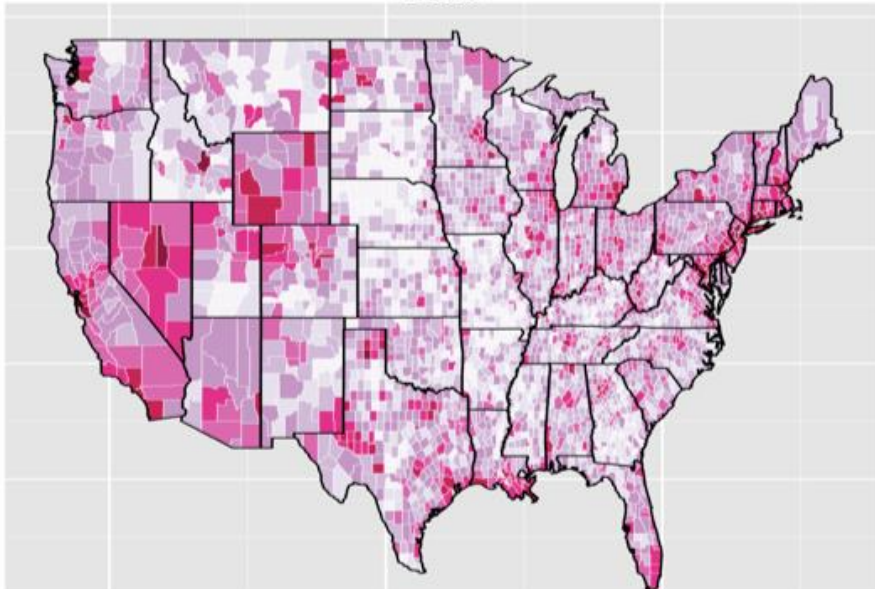




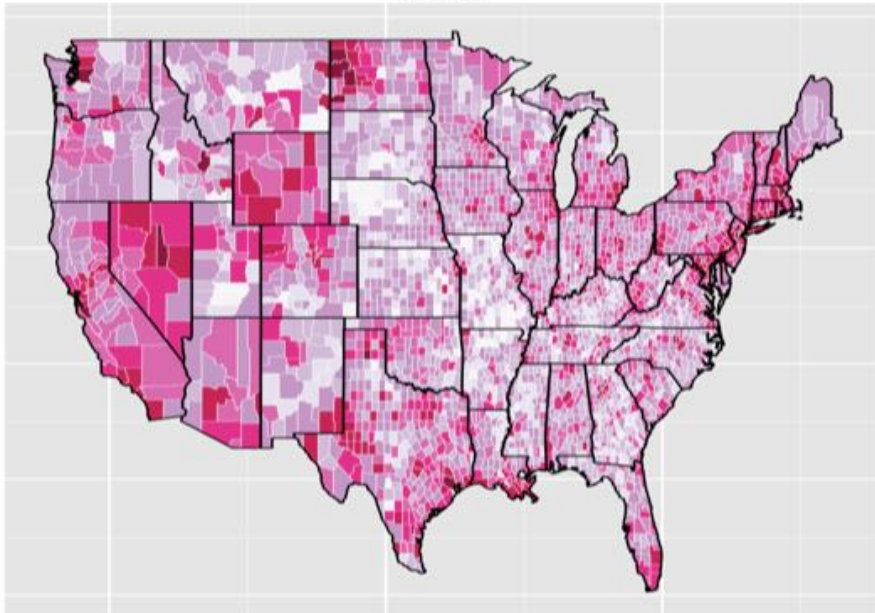
2006



2008



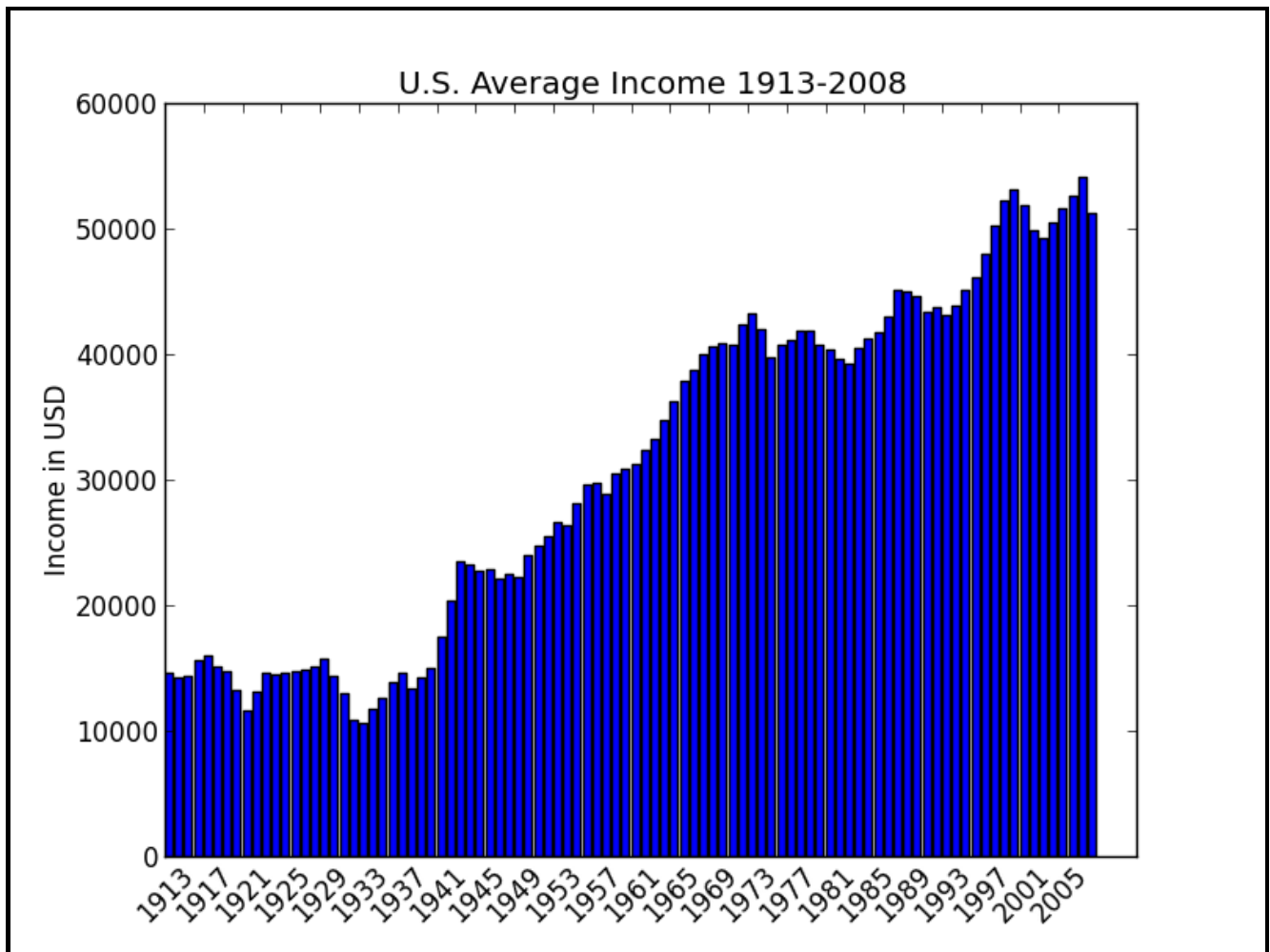
2012

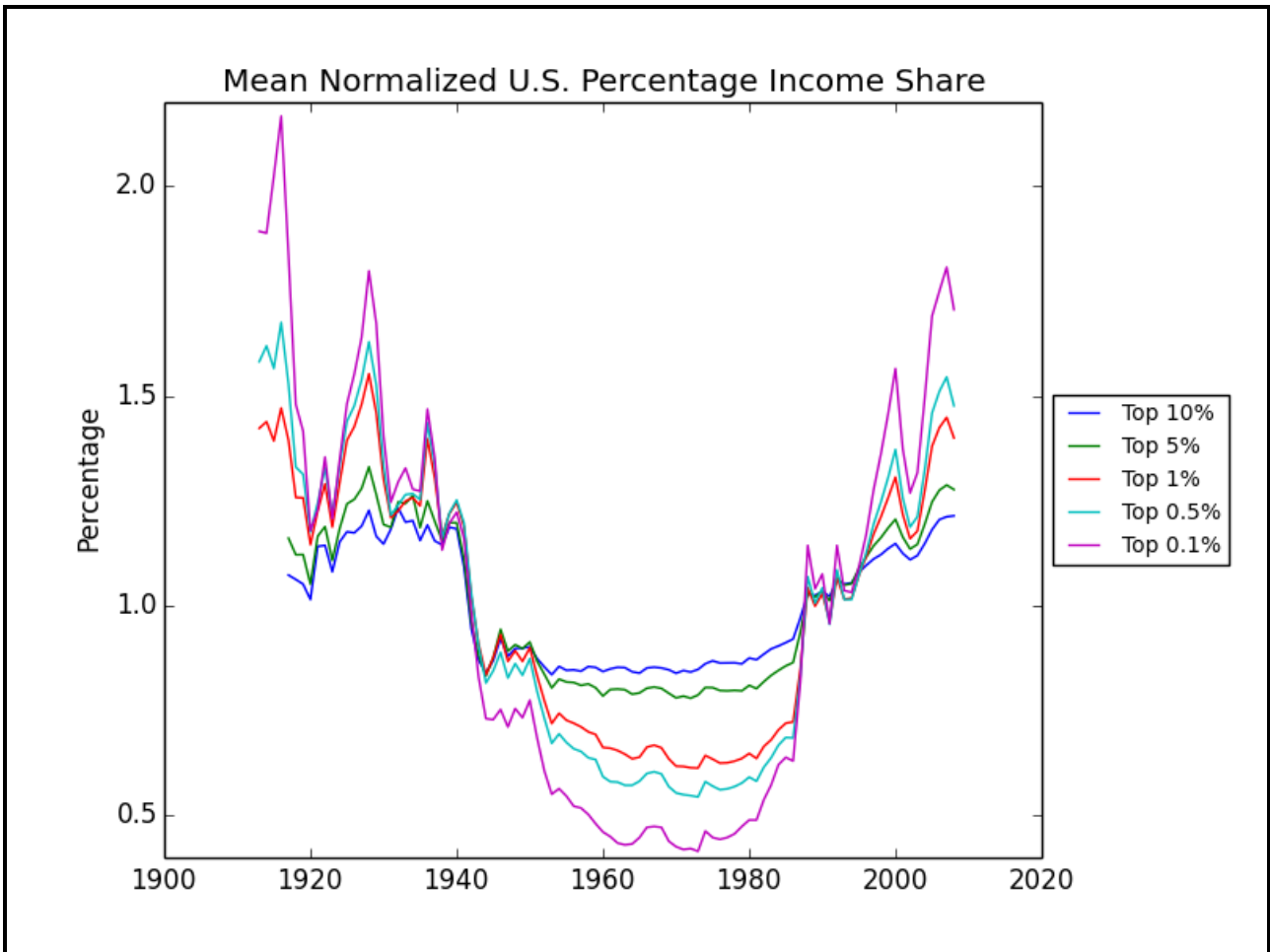
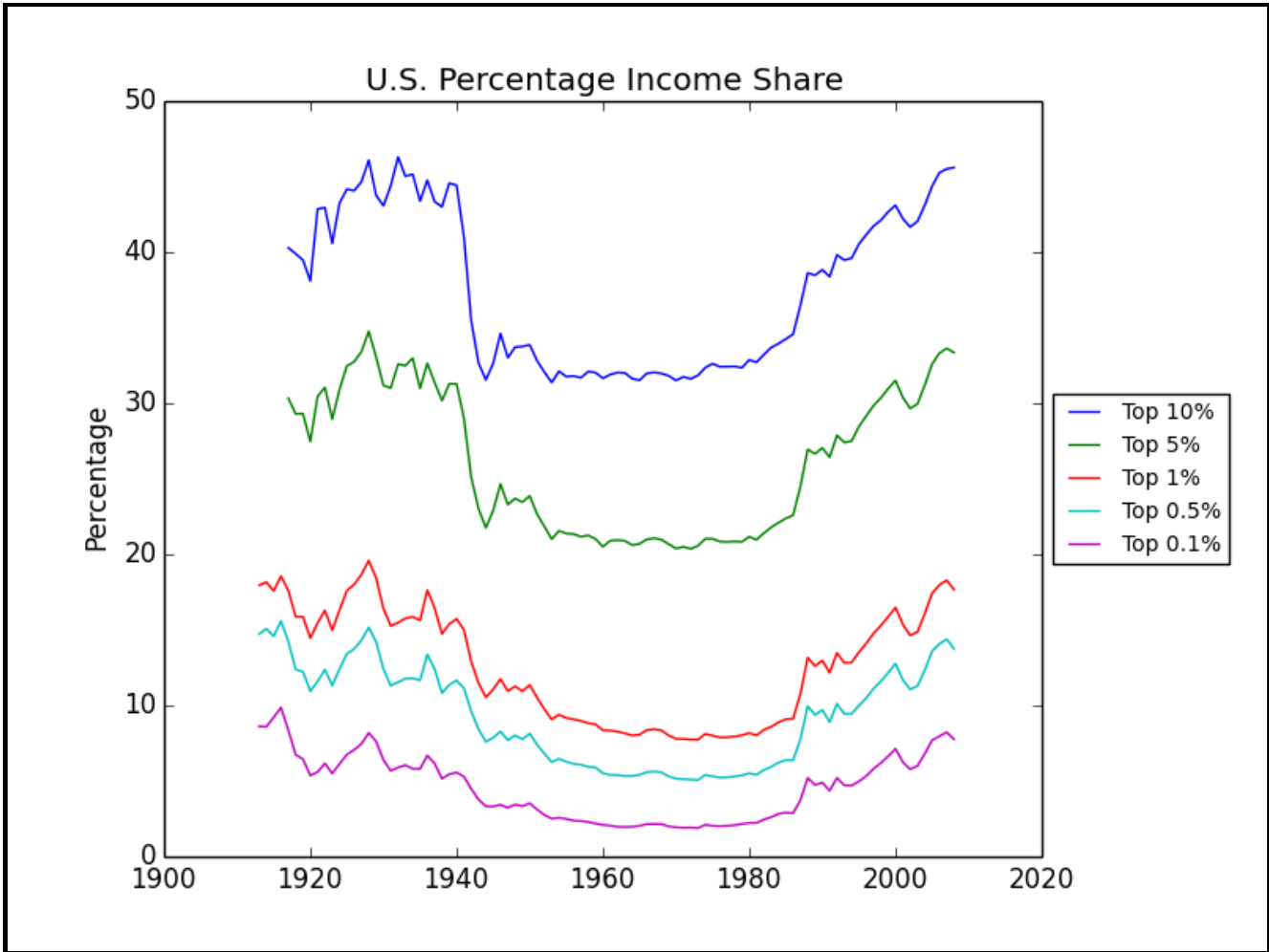


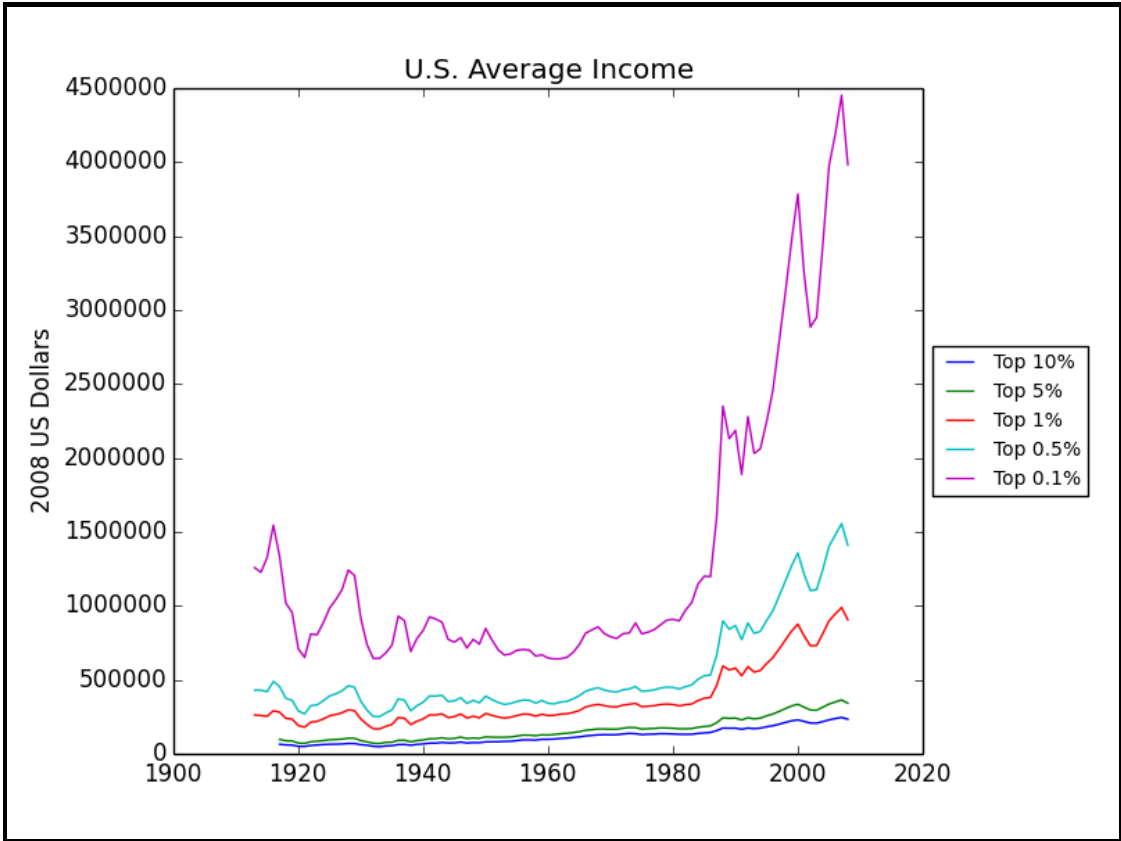
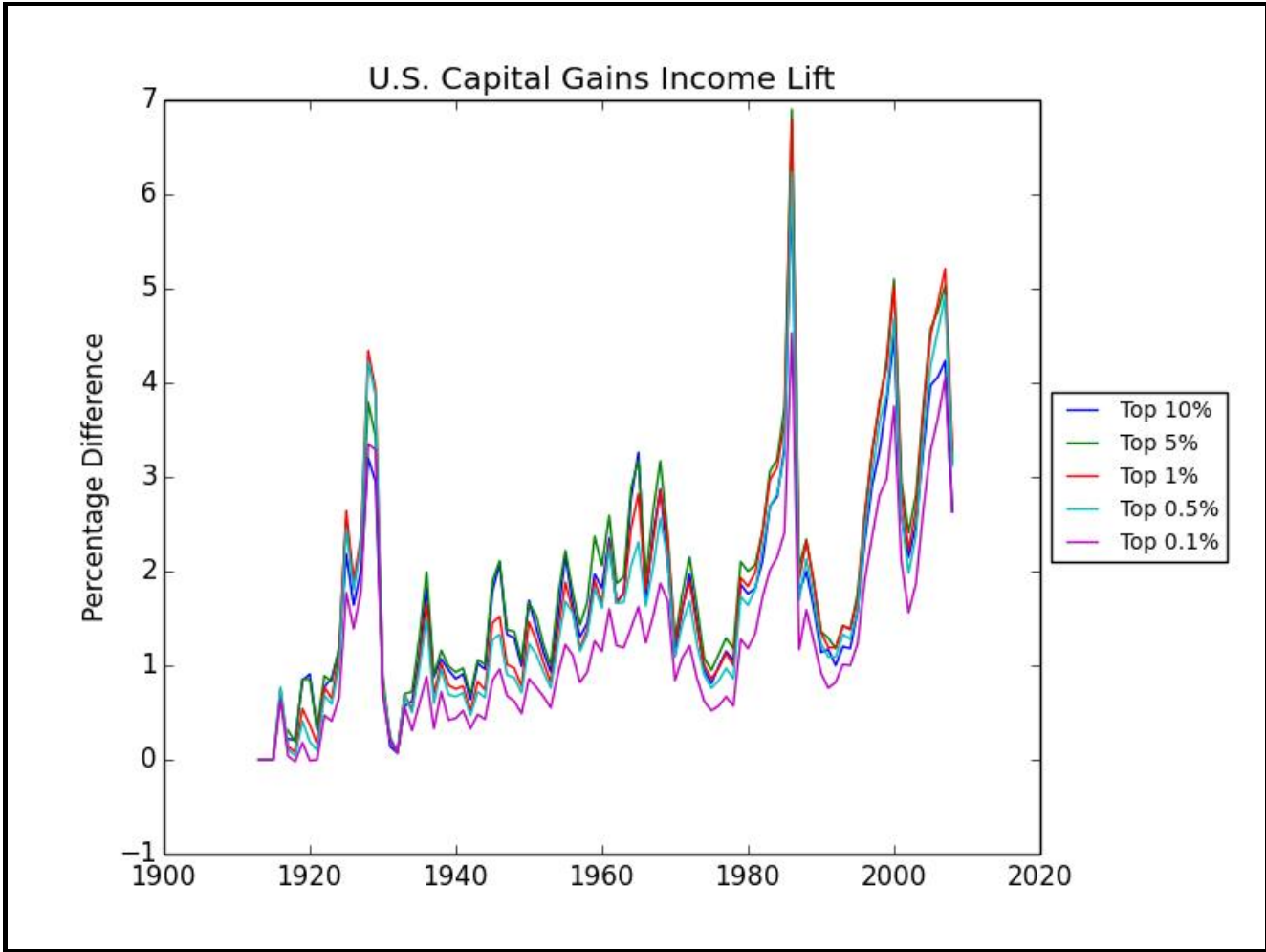
```
> opload
ng.do test replications elapsed relative user.self sys.self user.child sys.child
4 FREAD 5 16.29 1.000 15.85 0.43 NA NA
3 LOAD 5 79.67 4.891 79.10 0.50 NA NA
1 CSV 5 189.30 11.621 160.82 2.54 NA NA
2 CSVZIP 5 212.02 13.015 182.46 1.55 NA NA
```

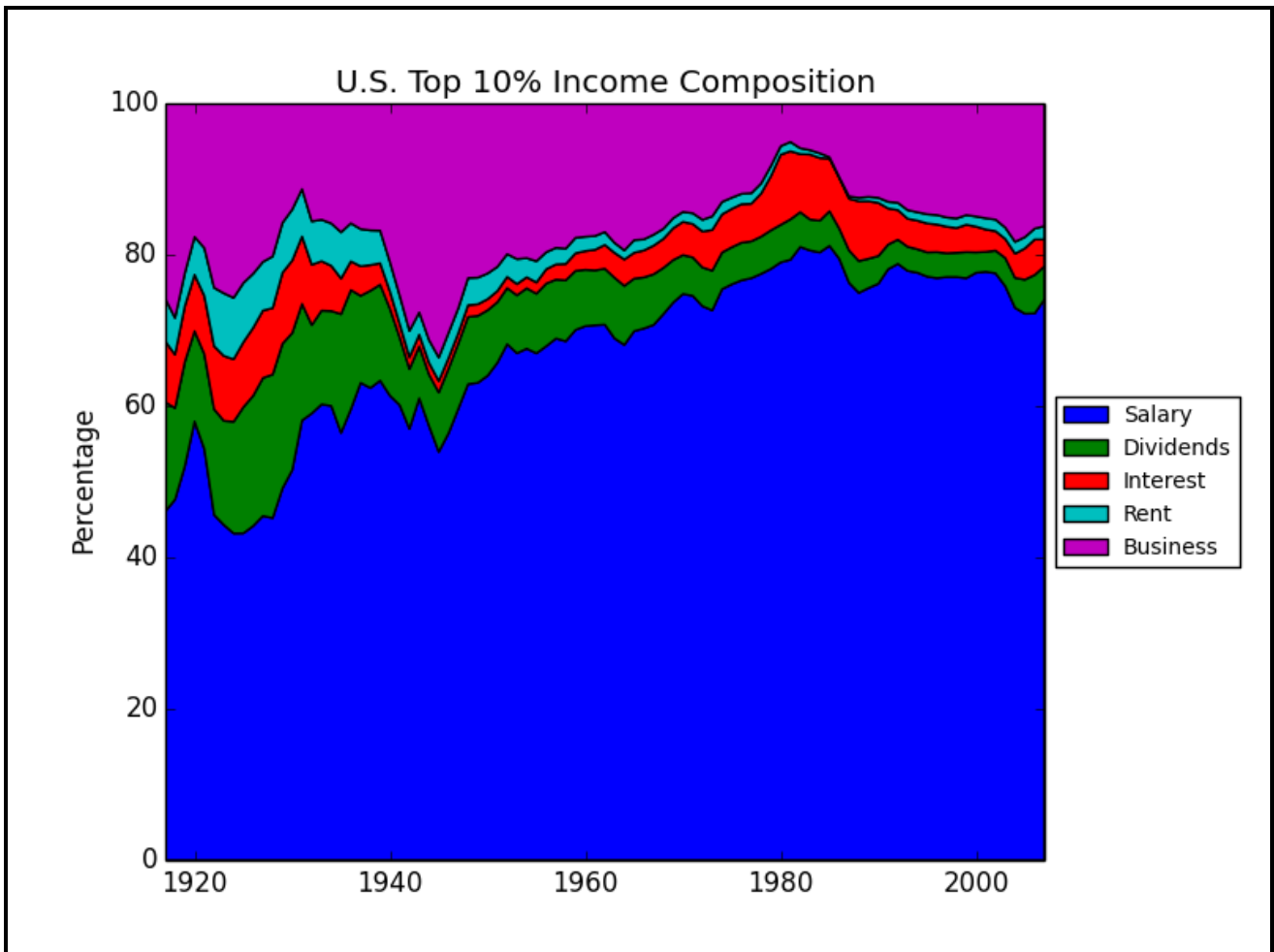
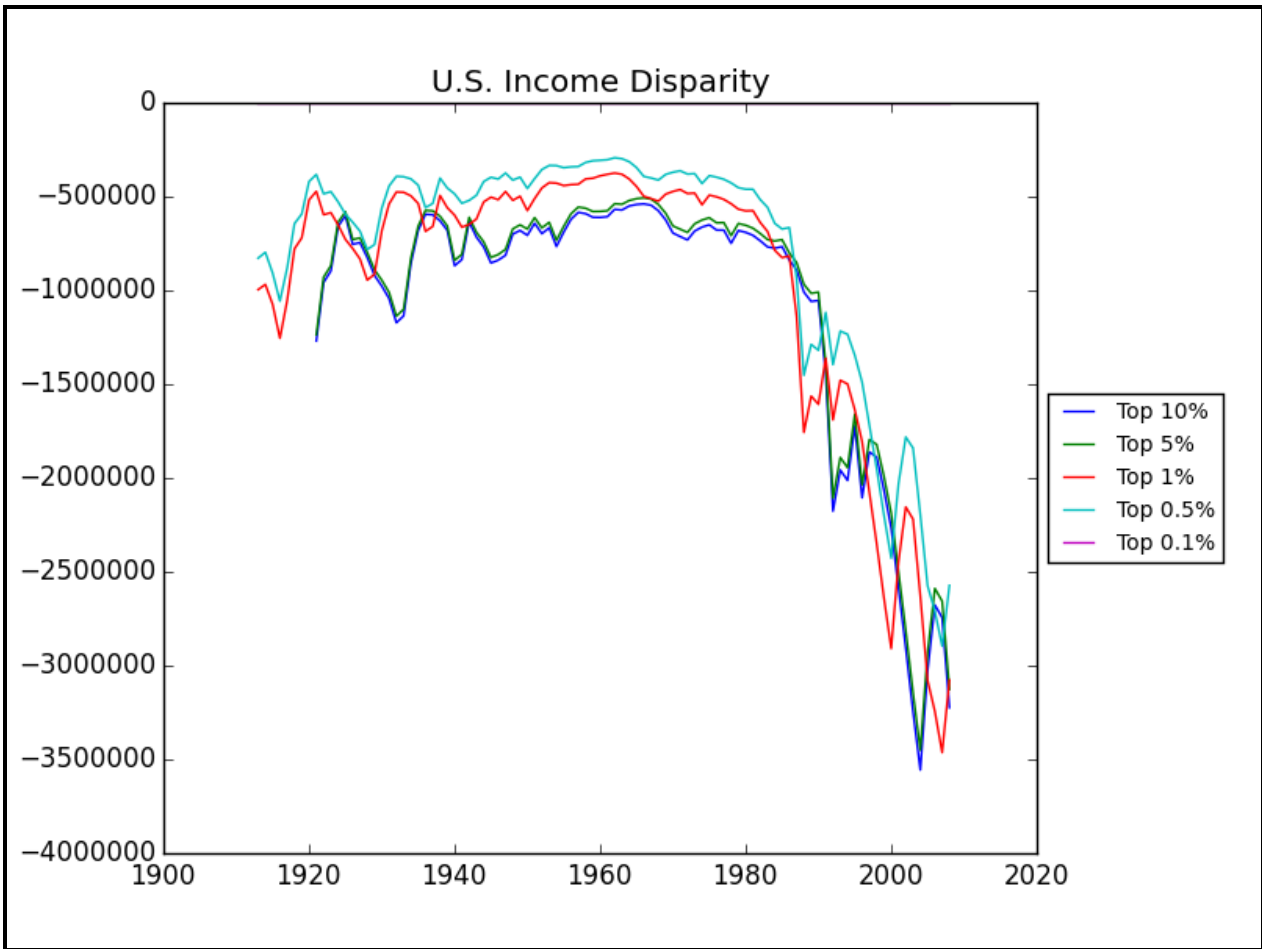
```
> op
test replications elapsed relative user.self sys.self user.child sys.child
1 DT 5 0.41 1.000 0.41 0.00 NA NA
4 DPLYR2 5 4.90 11.951 4.24 0.63 NA NA
3 DPLYR 5 5.40 13.171 4.52 0.86 NA NA
2 PLYR 5 97.70 238.293 95.46 1.67 NA NA
5 MERGE 5 207.14 505.220 204.14 2.54 NA NA
```

Chapter 6: Creating Application-oriented Analyses Using Tax Data (Python)

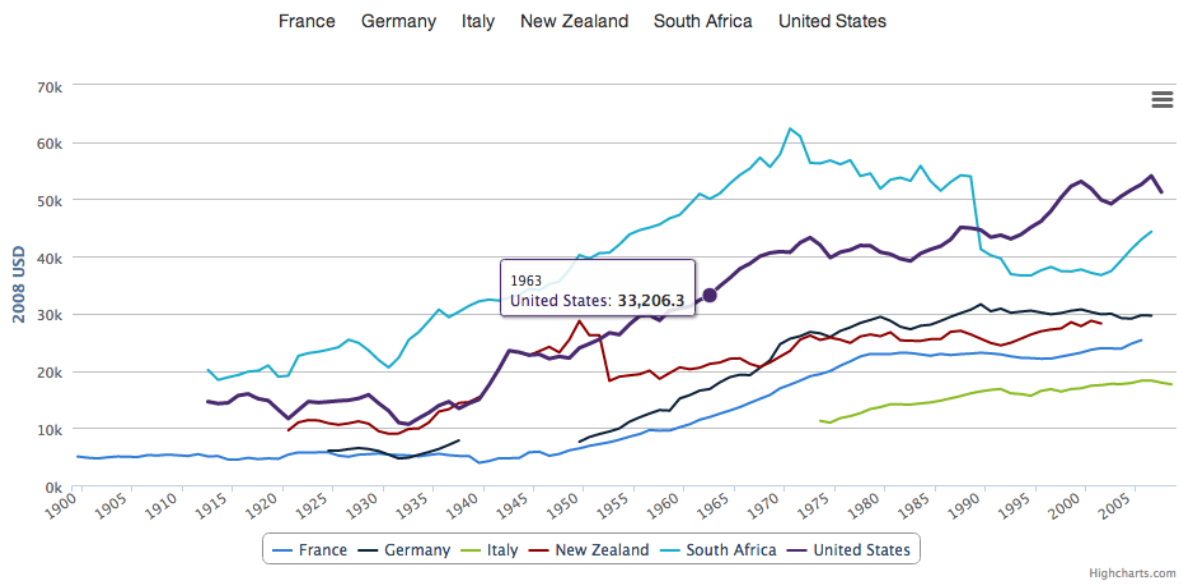




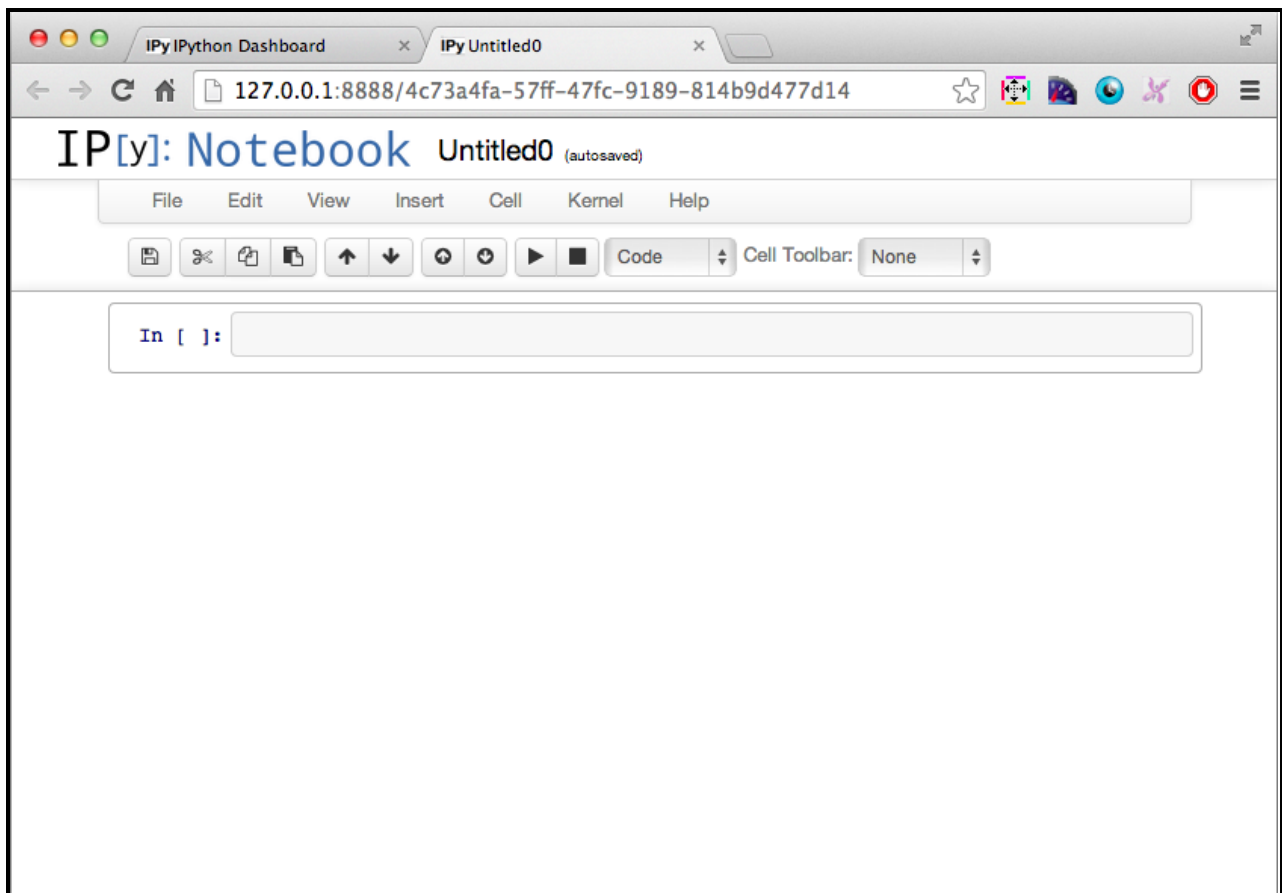
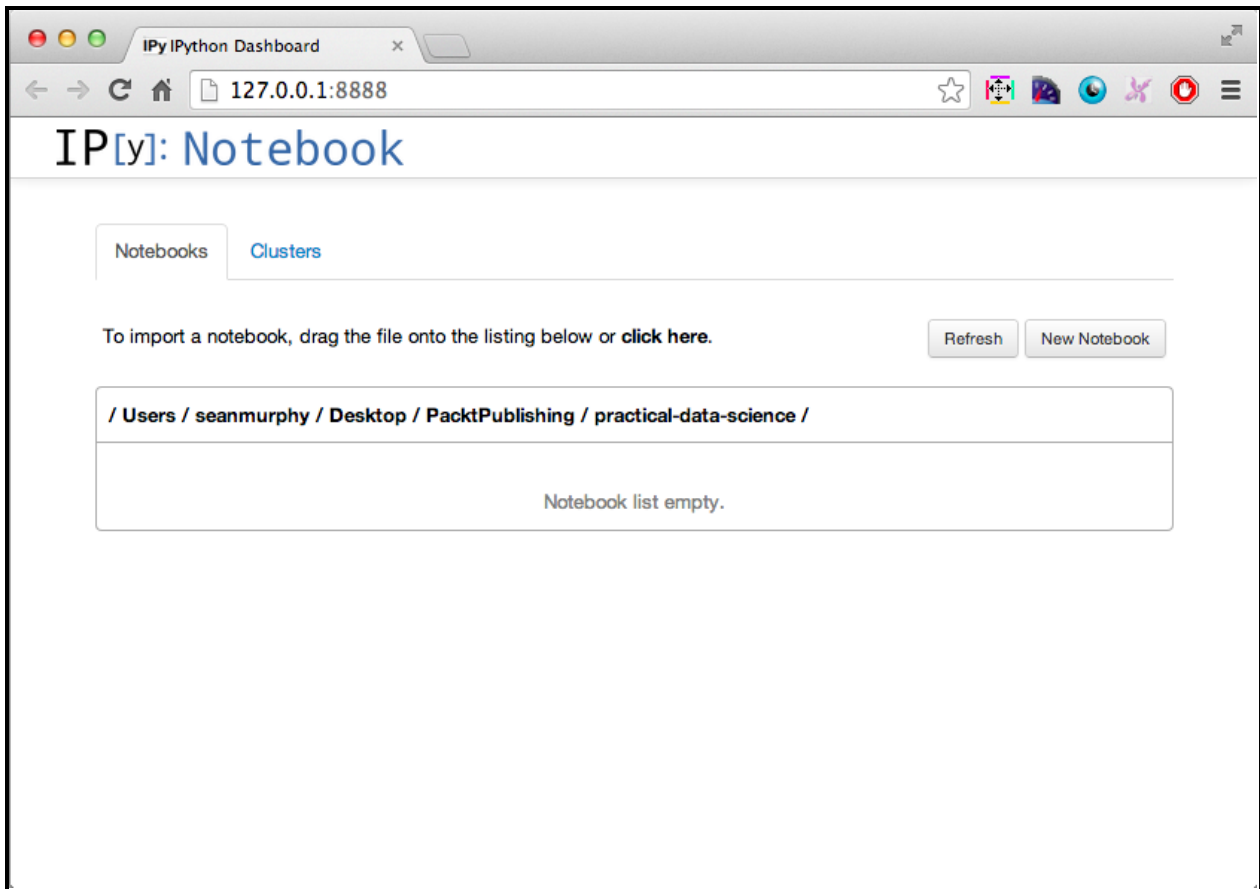




Average Income per Family, 1900 - 2009



Chapter 7: Driving Visual Analyses with Automobile Data (Python)



The screenshot shows a web browser window with two tabs: "IPy IPython Dashboard" and "IPy Untitled0". The address bar displays the URL "127.0.0.1:8888/4c73a4fa-57ff-47fc-9189-814b9d477d14". The notebook title is "IP[y]: Notebook Untitled0 (unsaved changes)". The menu bar includes "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". The toolbar contains icons for saving, undo, redo, and running code. The first code cell contains the input "In [1]: 5+5" and the output "Out[1]: 10". A second empty code cell is visible below it, labeled "In []:".

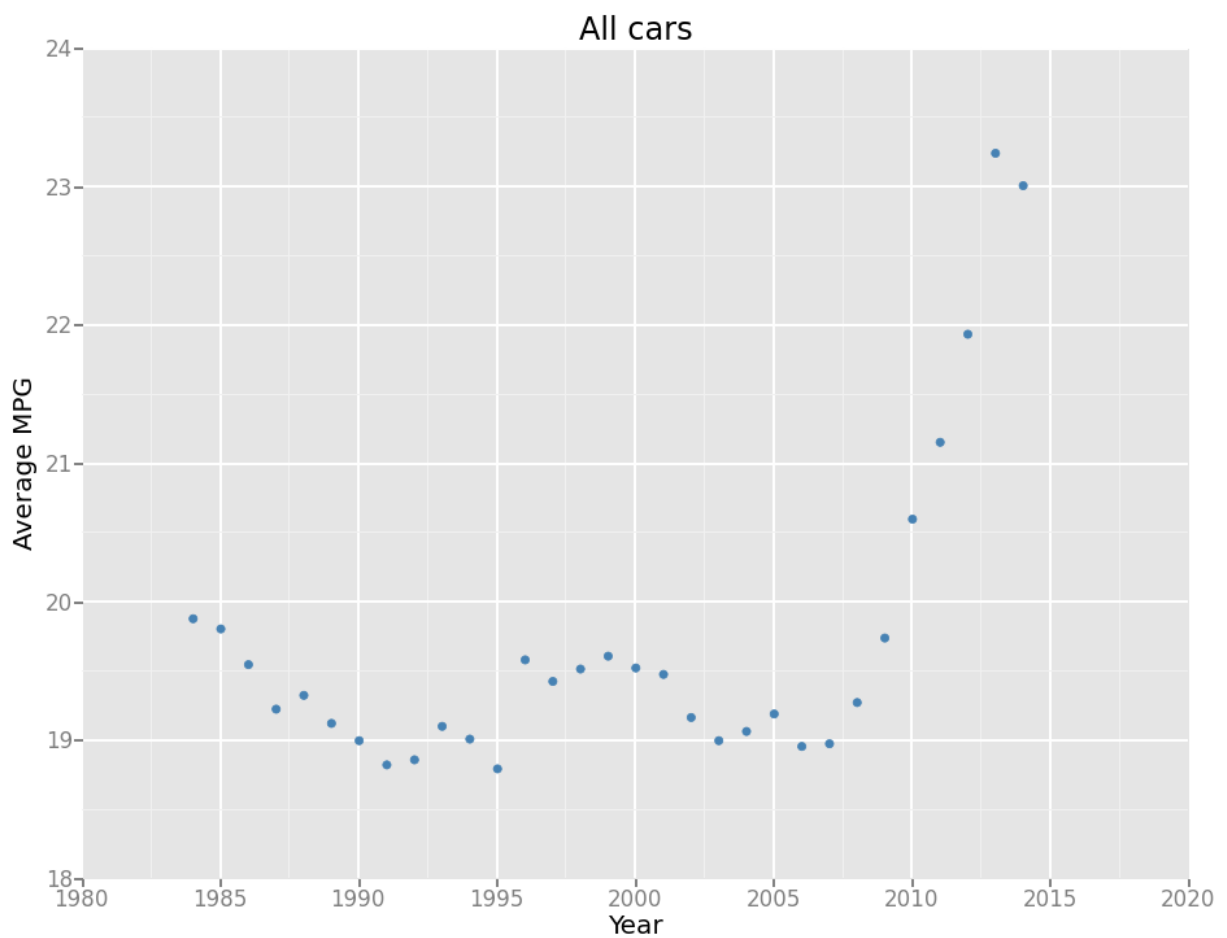
```

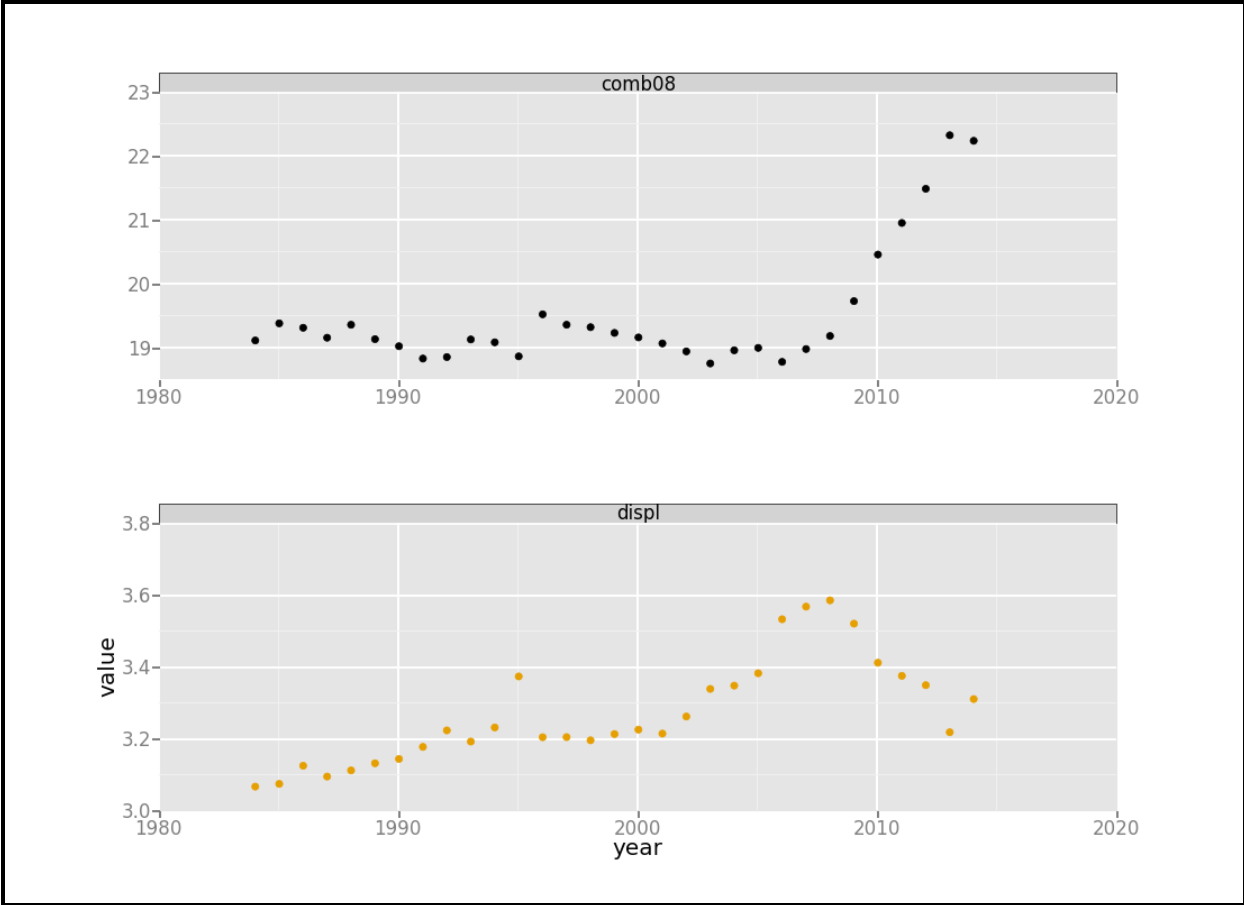
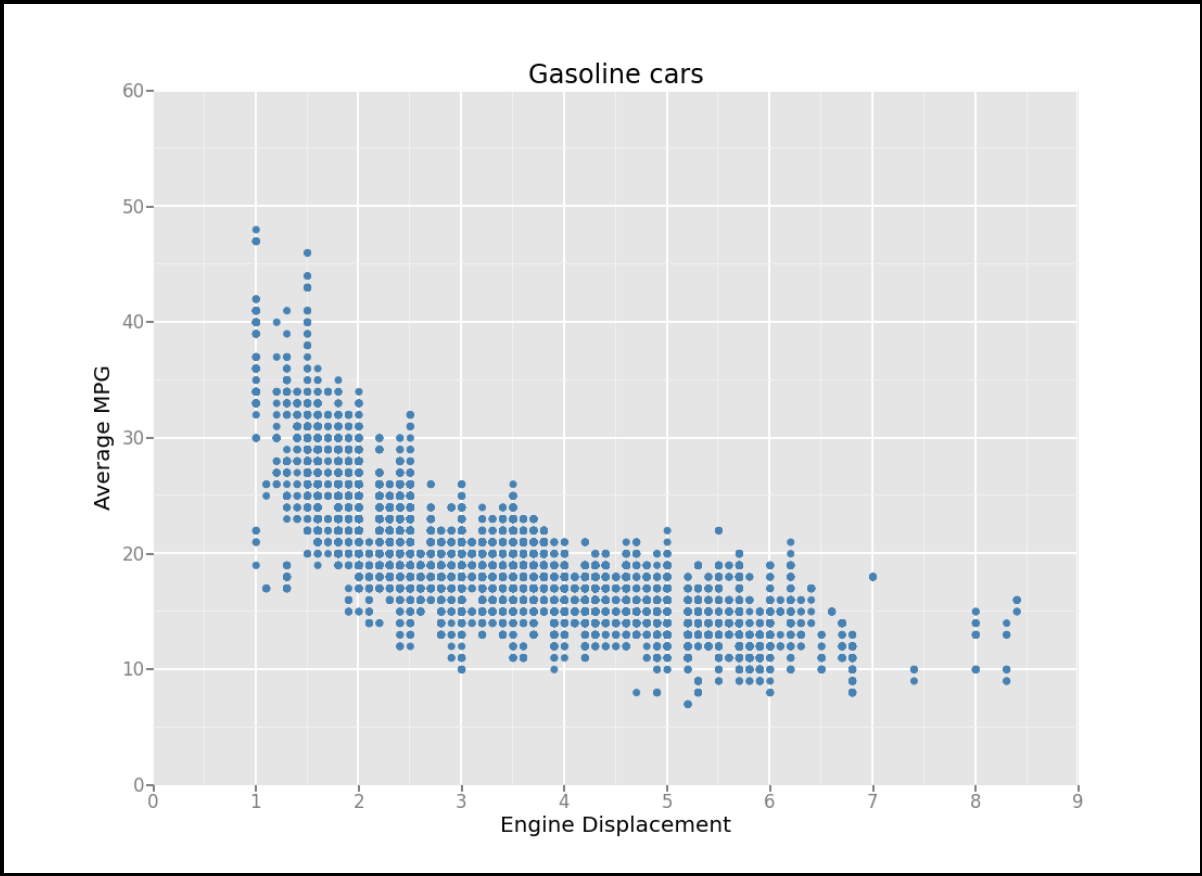
/Library/Python/2.7/site-packages/pandas/io/parsers.py:1139: DtypeWarning: Columns (22,23,70,71,72,73) have mixed types. Specify dtype option on import or set low_memory=False.
  data = self._reader.read(nrows)
Out[2]: <bound method DataFrame.head of
   barrels08 barrelsA08 charge120 charge240 city08 city08U cityA08 \
0    15.689436         0         0         19         0         0
1    29.950562         0         0         9         0         0
2    12.195570         0         0        23         0         0
3    29.950562         0         0        10         0         0
4    17.337486         0         0        17         0         0
5    14.964294         0         0        21         0         0
6    13.184400         0         0        22         0         0
7    13.733750         0         0        23         0         0
8    12.657024         0         0        23         0         0
9    13.184400         0         0        23         0         0
10   12.657024         0         0        23         0         0
11   15.689436         0         0        18         0         0
12   13.733750         0         0        21         0         0
13   15.689436         0         0        18         0         0
14   25.336022         0         0        12         0         0

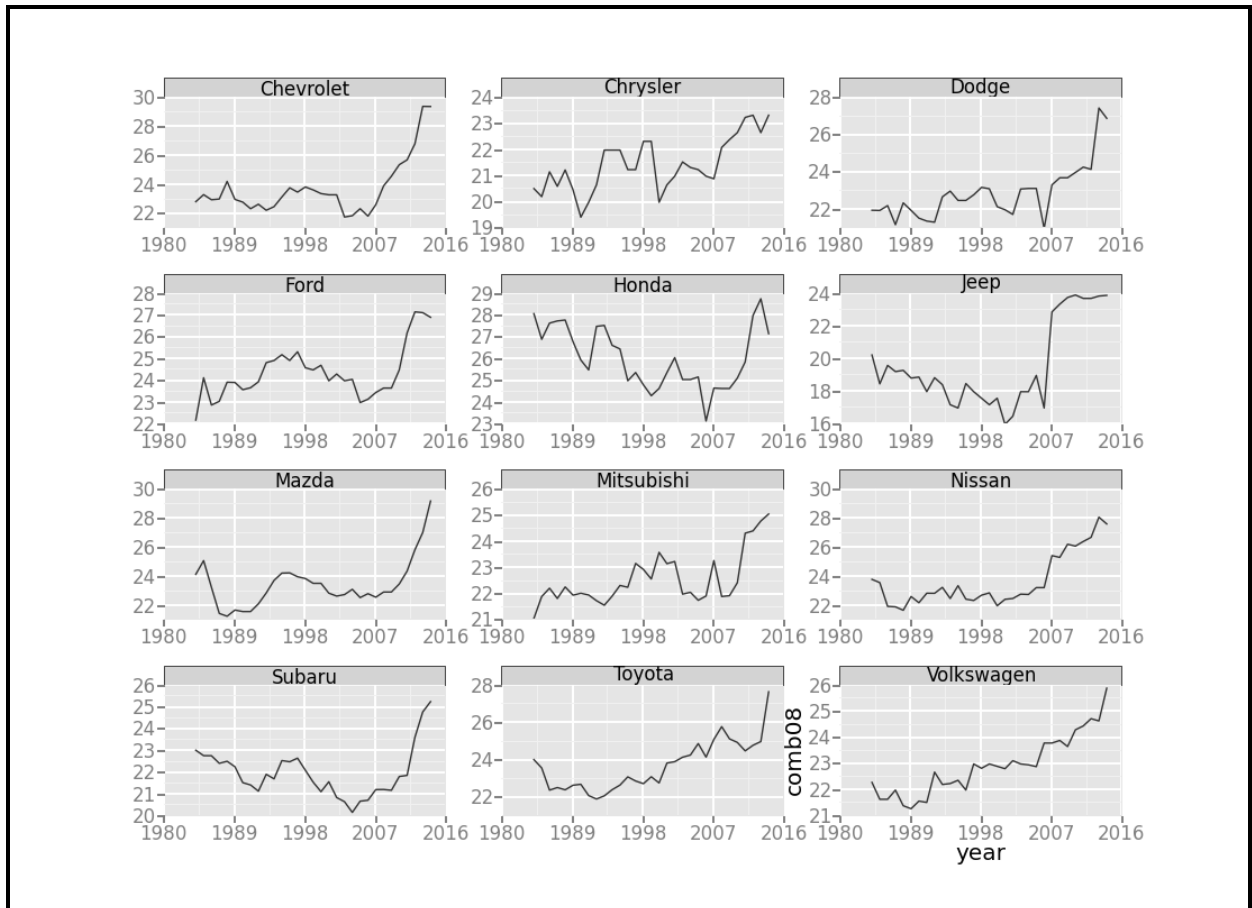
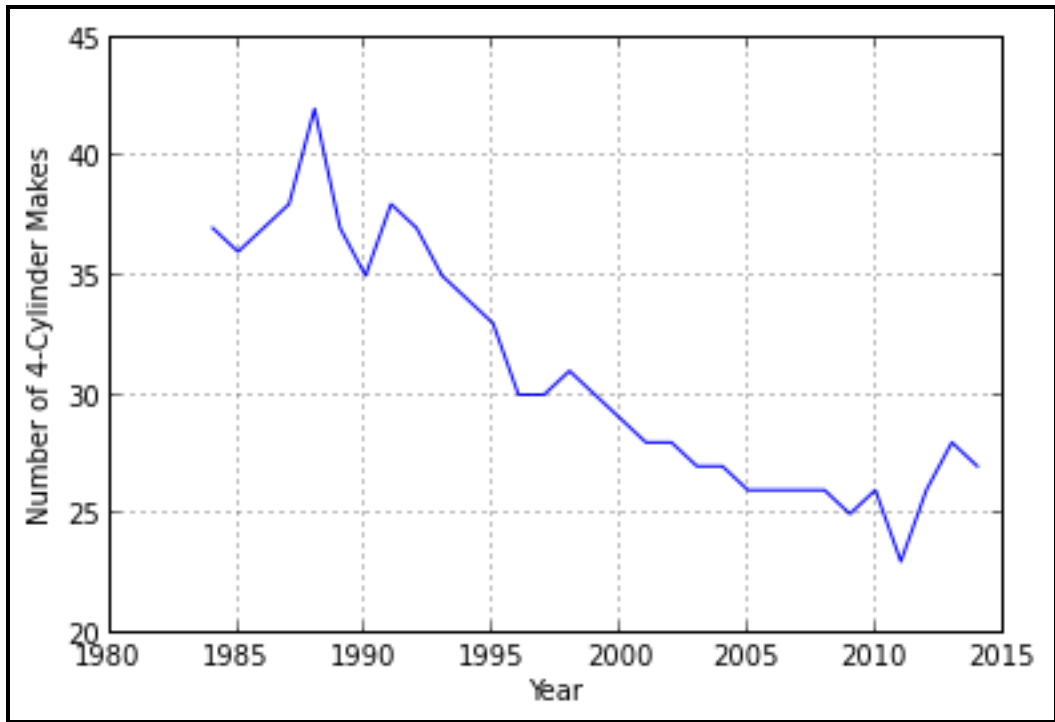
```

```
In [3]: pd.value_counts(vehicles.trany)
```

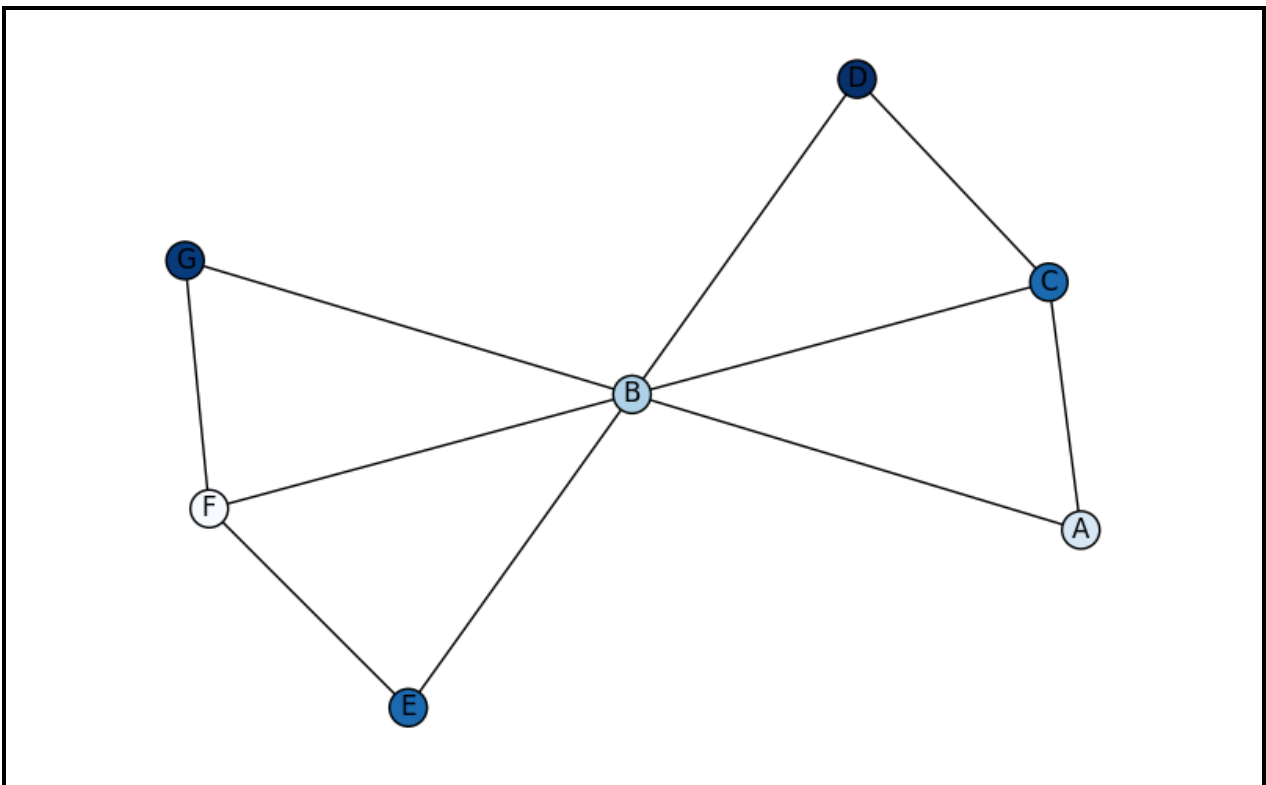
```
Out[3]: Automatic 4-spd      11029  
Manual 5-spd      8213  
Automatic 3-spd      3151  
Automatic 5-spd      2149  
Manual 6-spd      2009  
Automatic (S6)      1877  
Manual 4-spd      1483  
Automatic 6-spd      1093  
Automatic (S5)      813  
Automatic (variable gear ratios)  534  
Automatic 7-spd      493  
Automatic (S8)      366
```



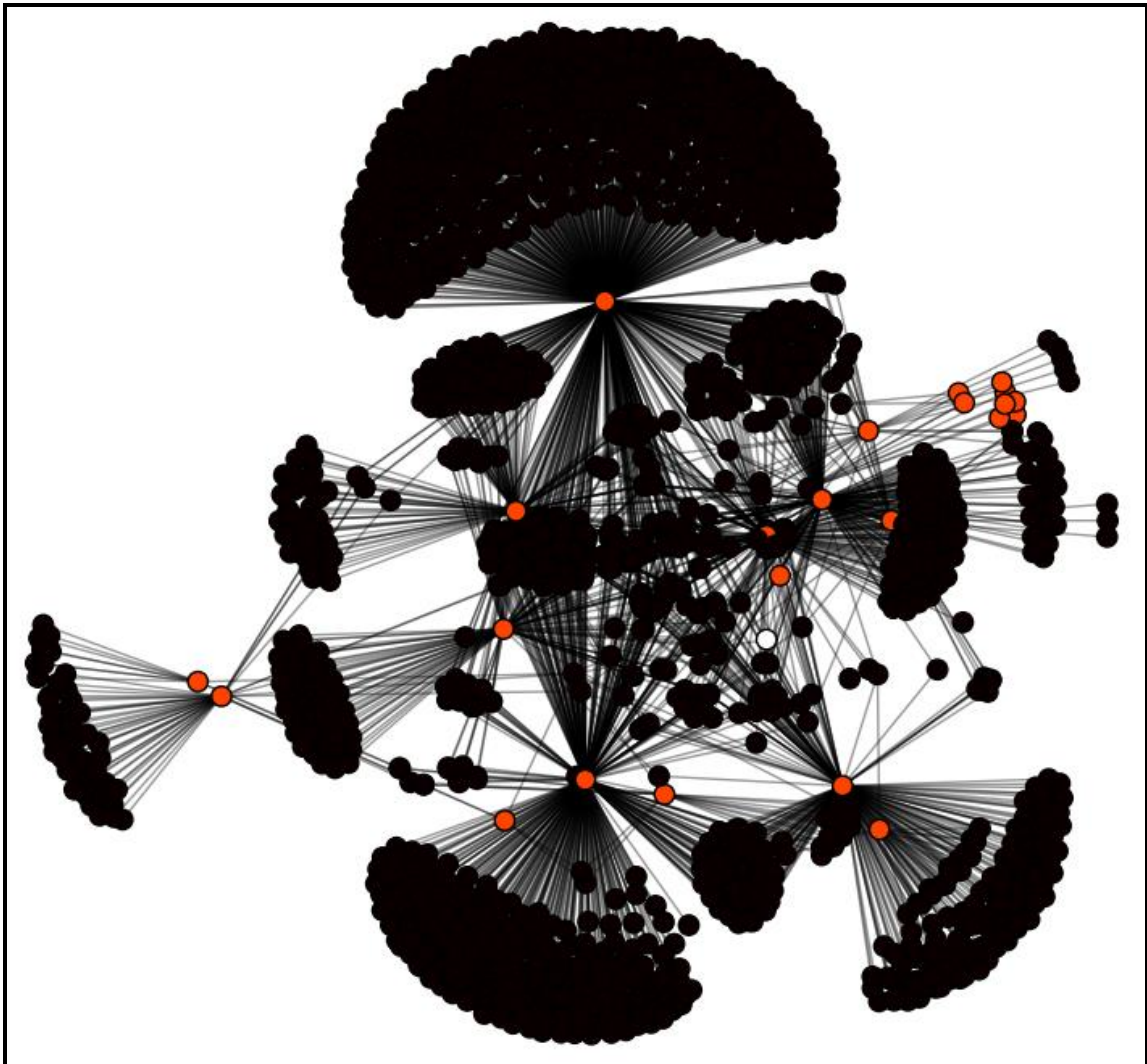
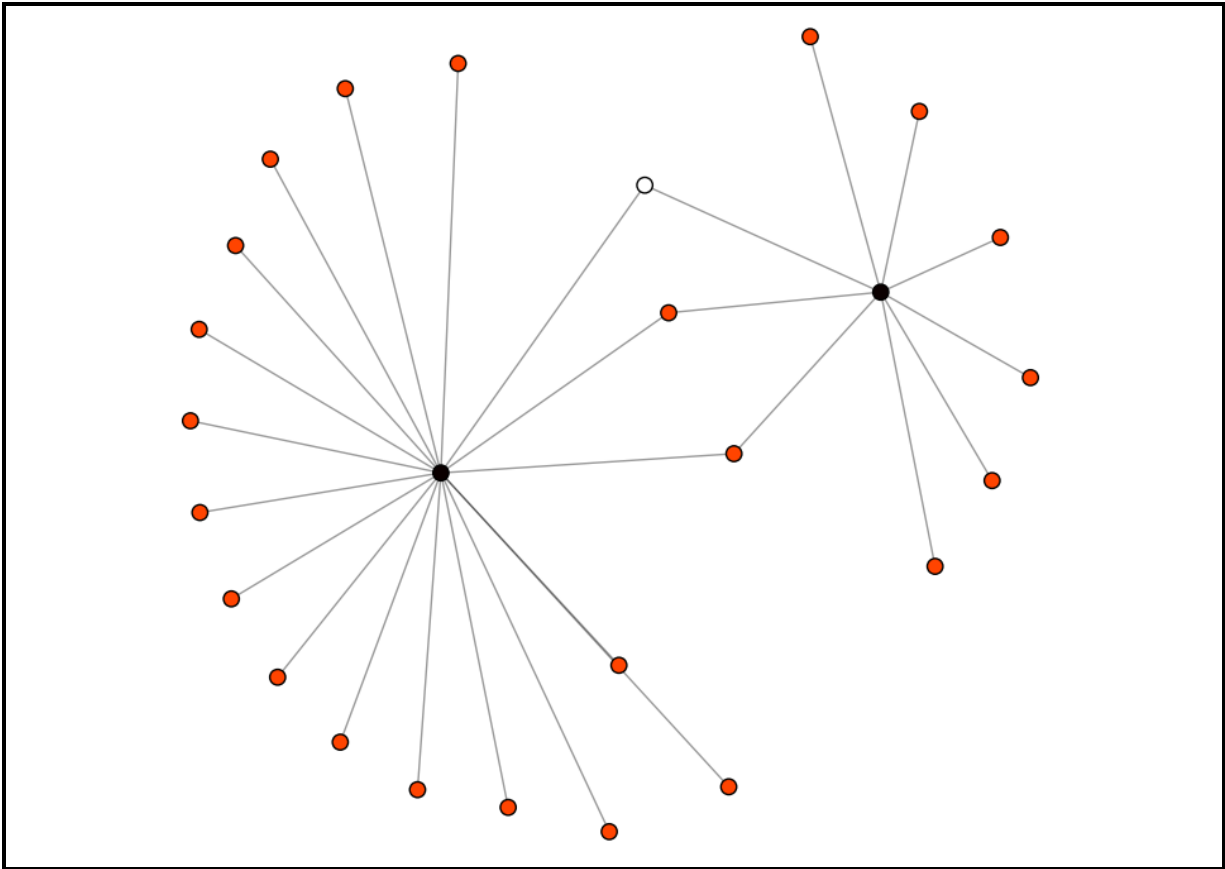


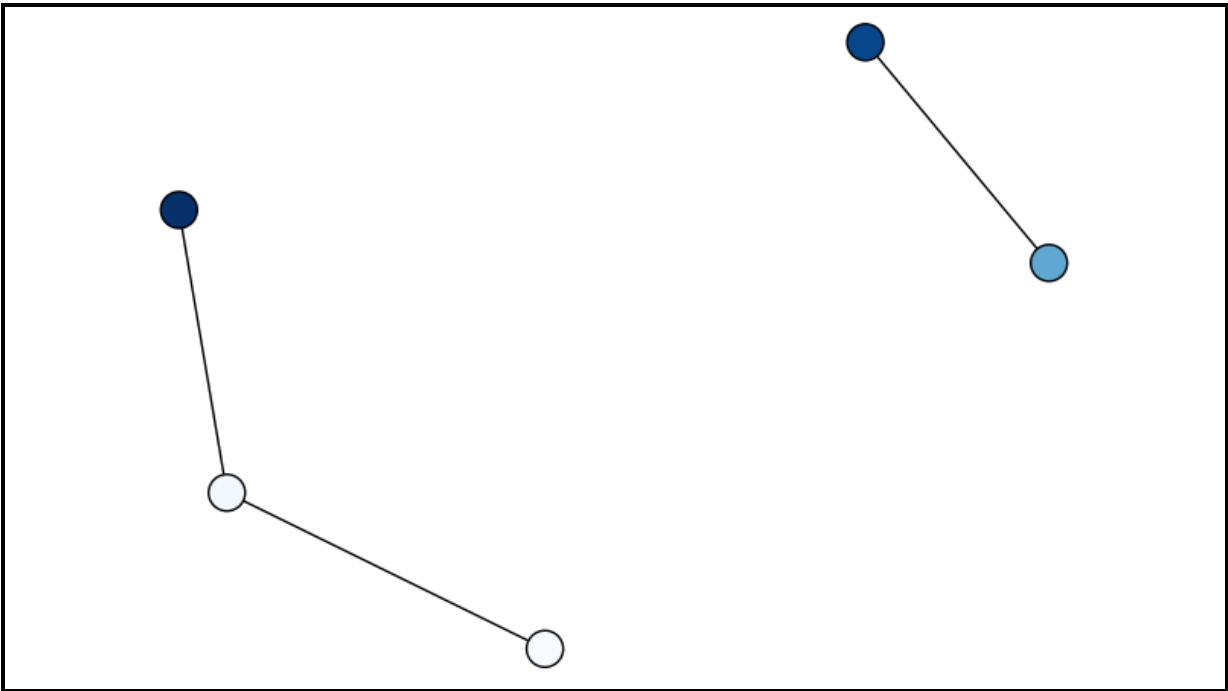
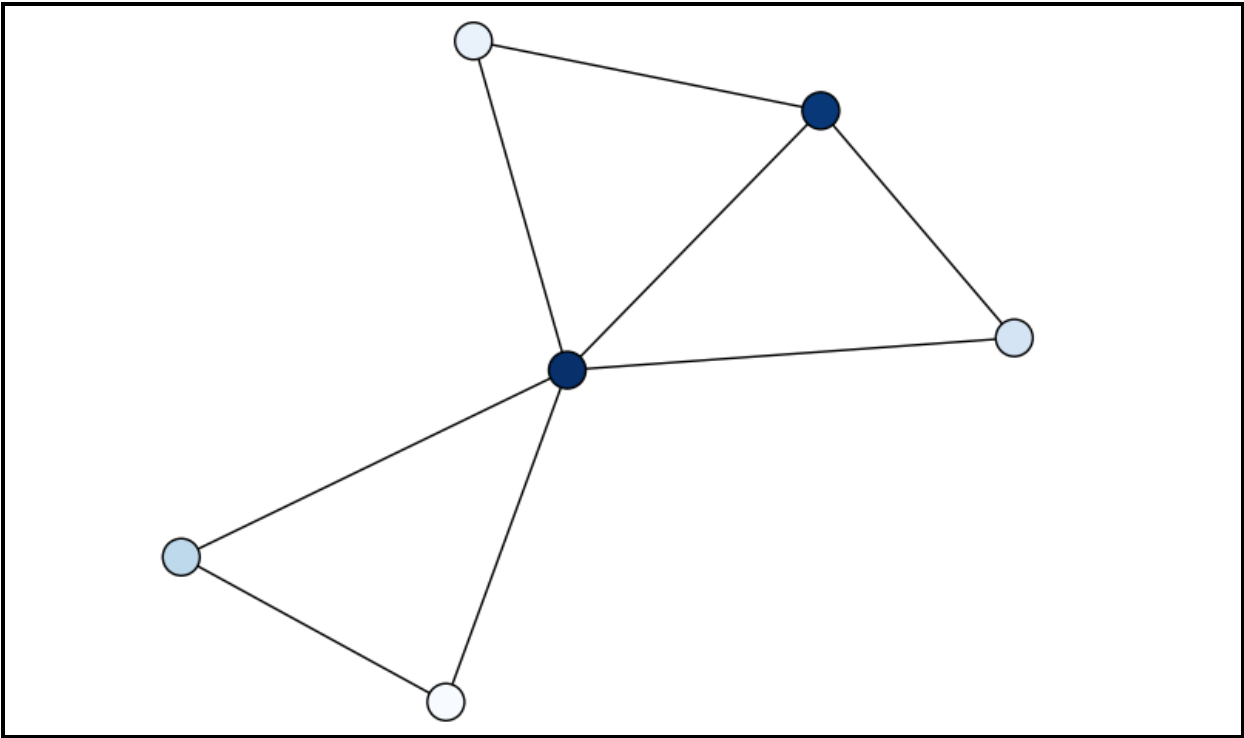


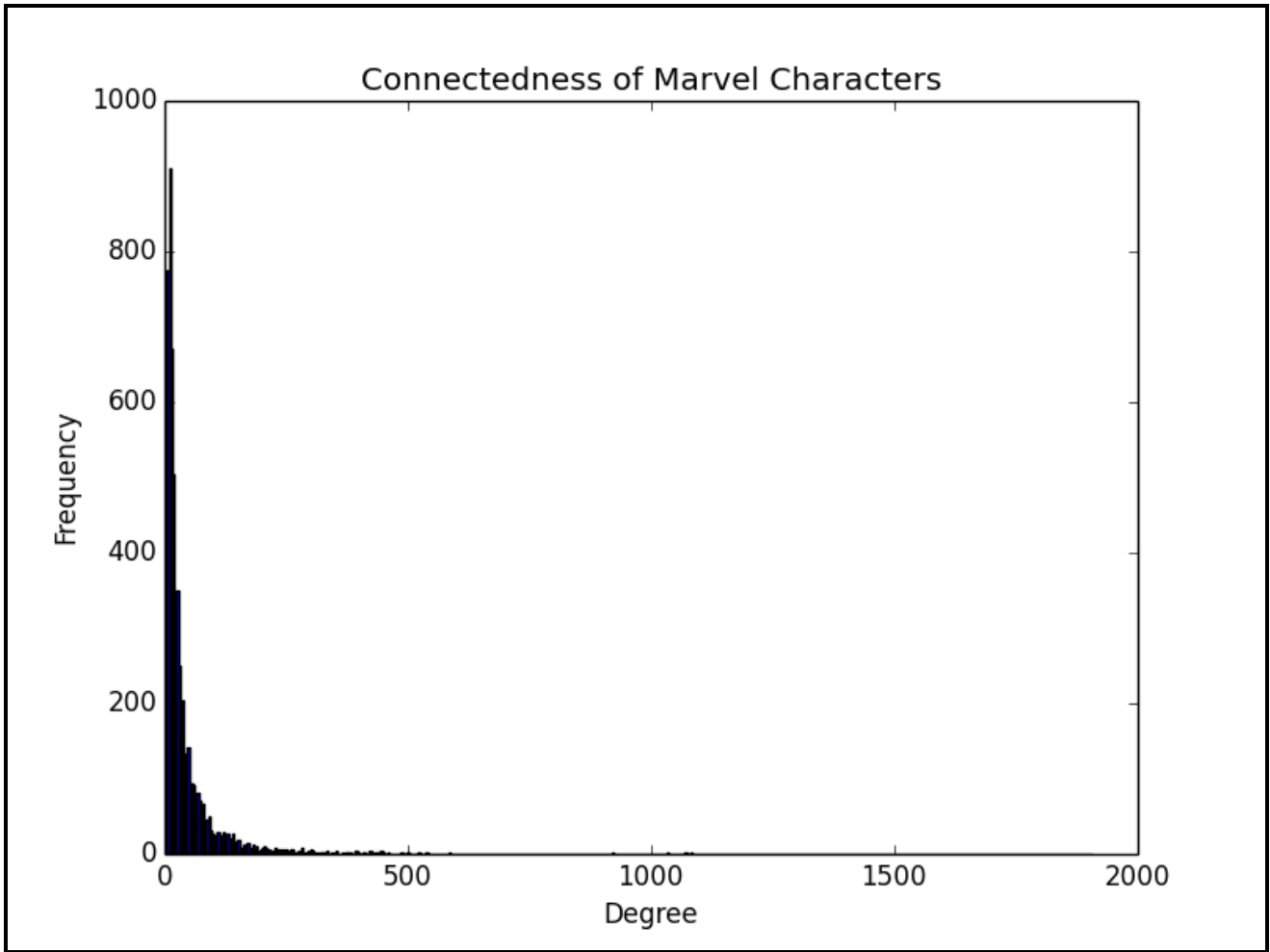
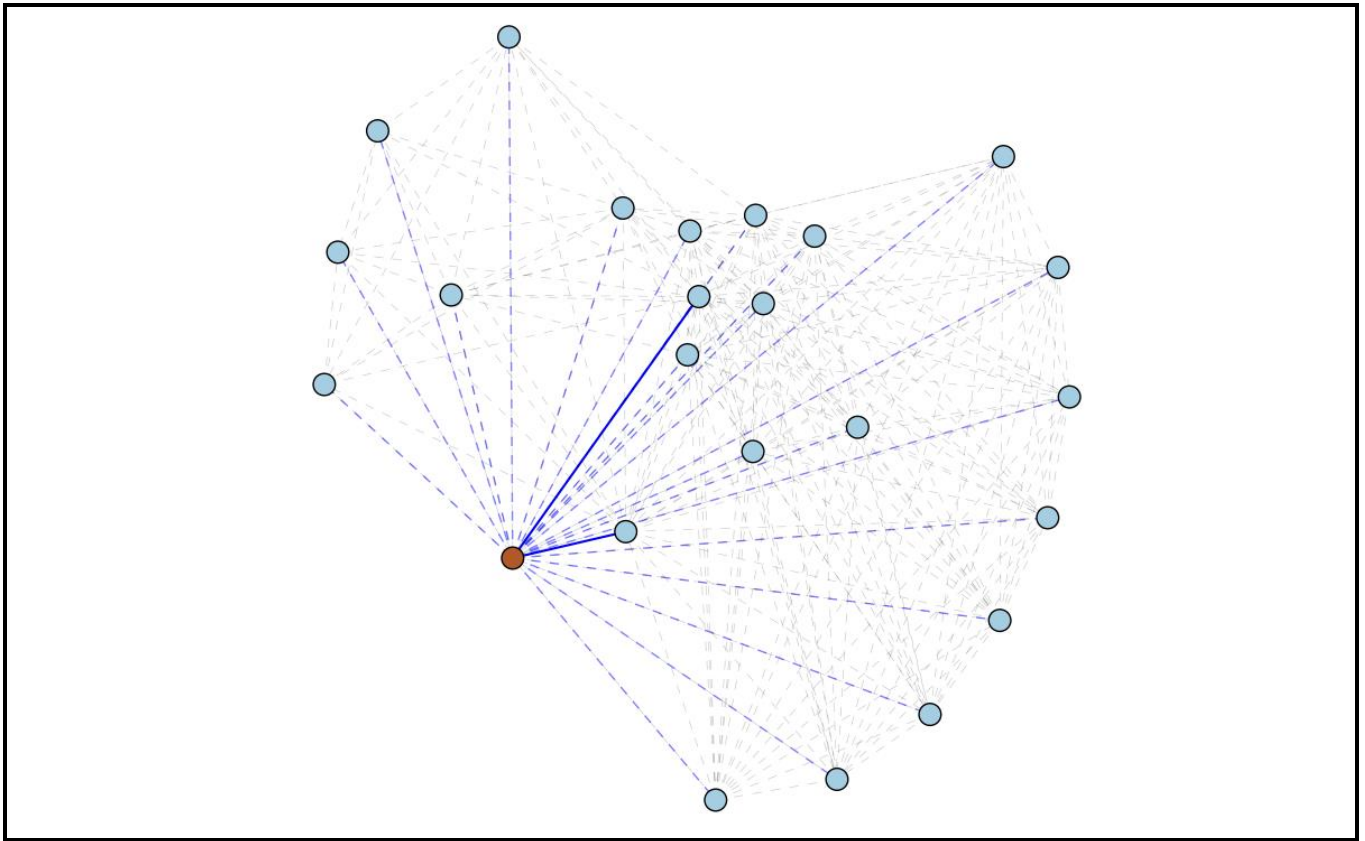
Chapter 8: Working with Social Graphs (Python)

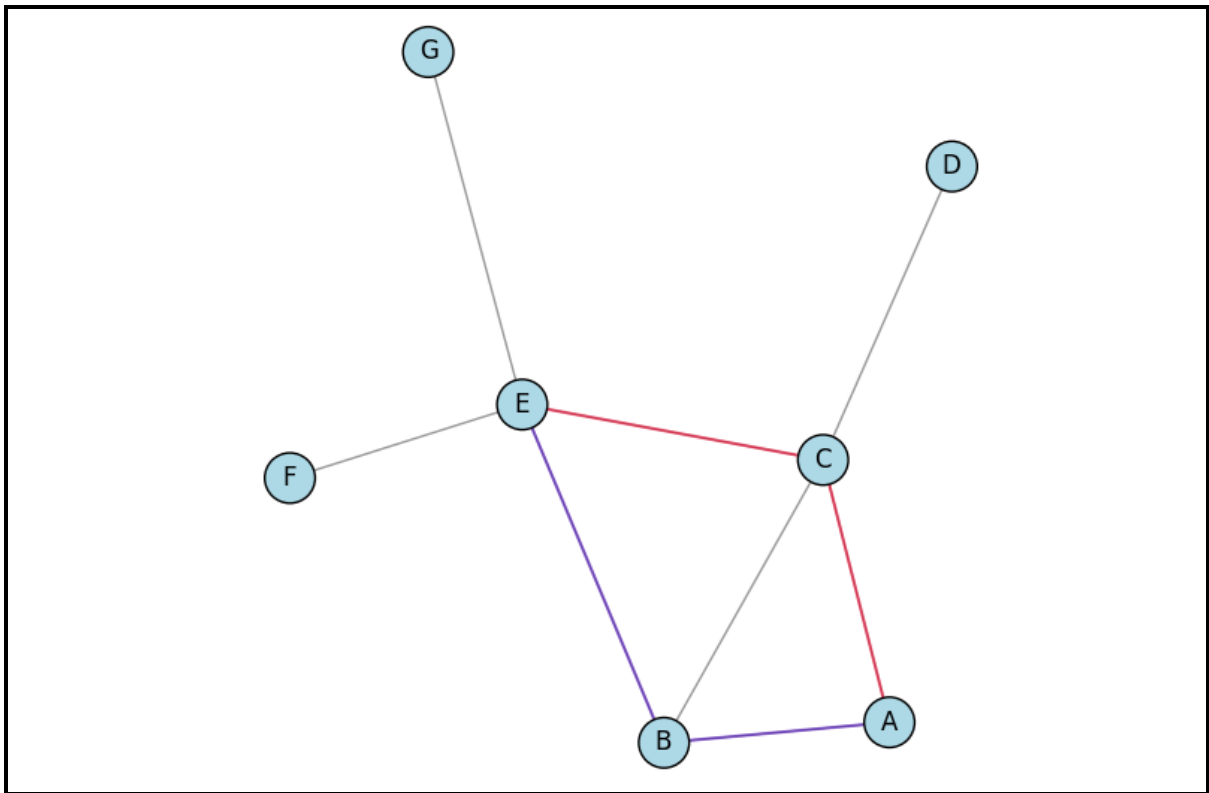
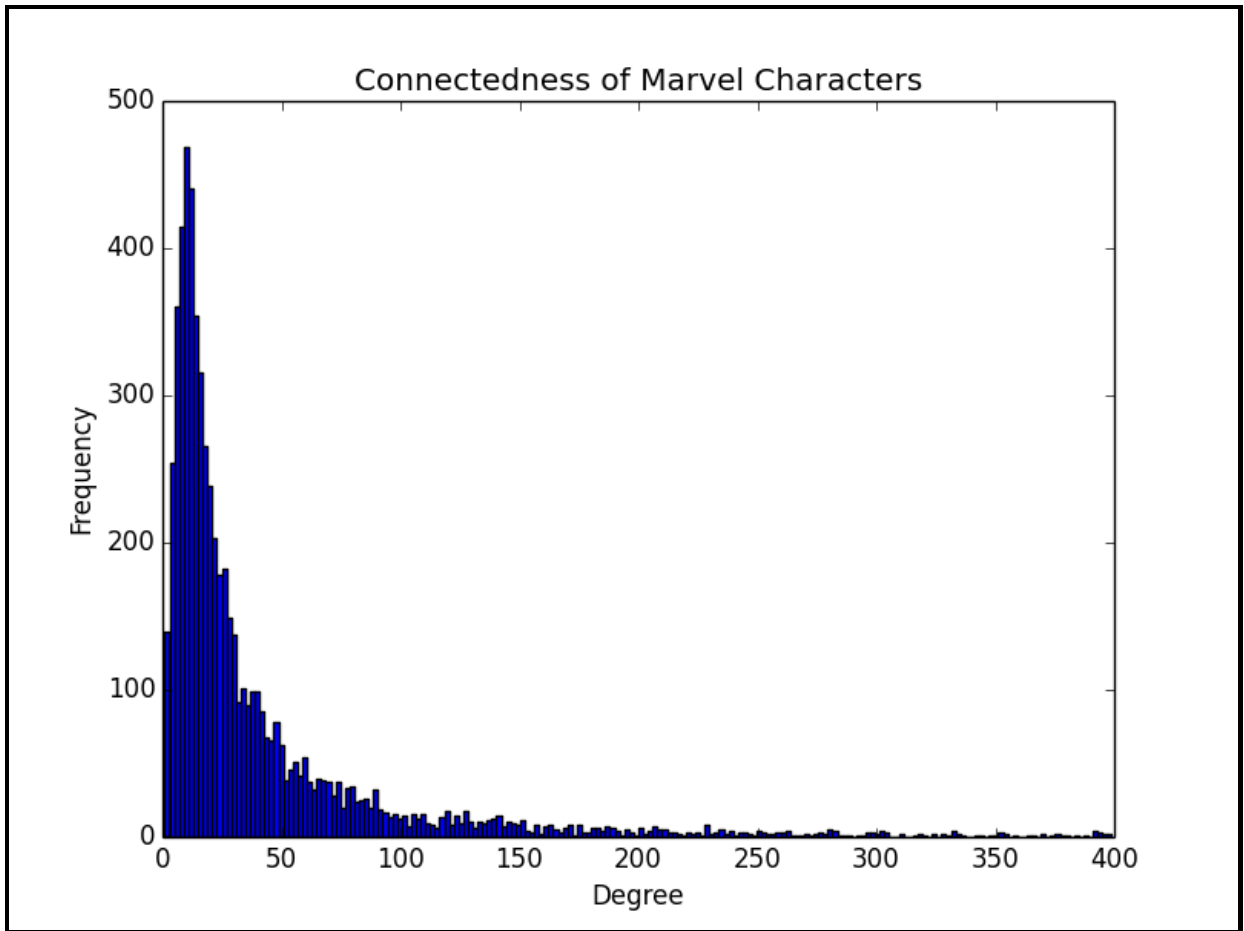


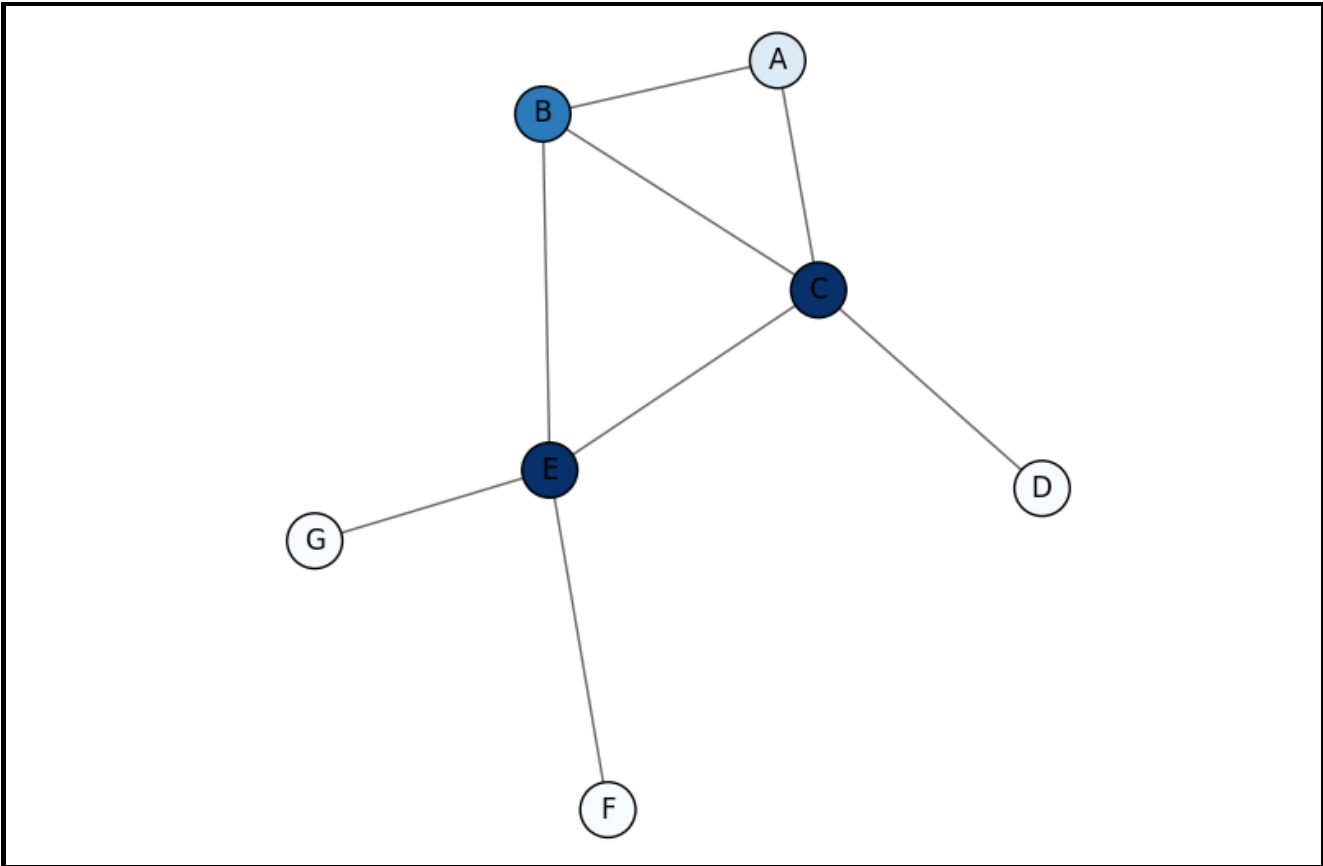
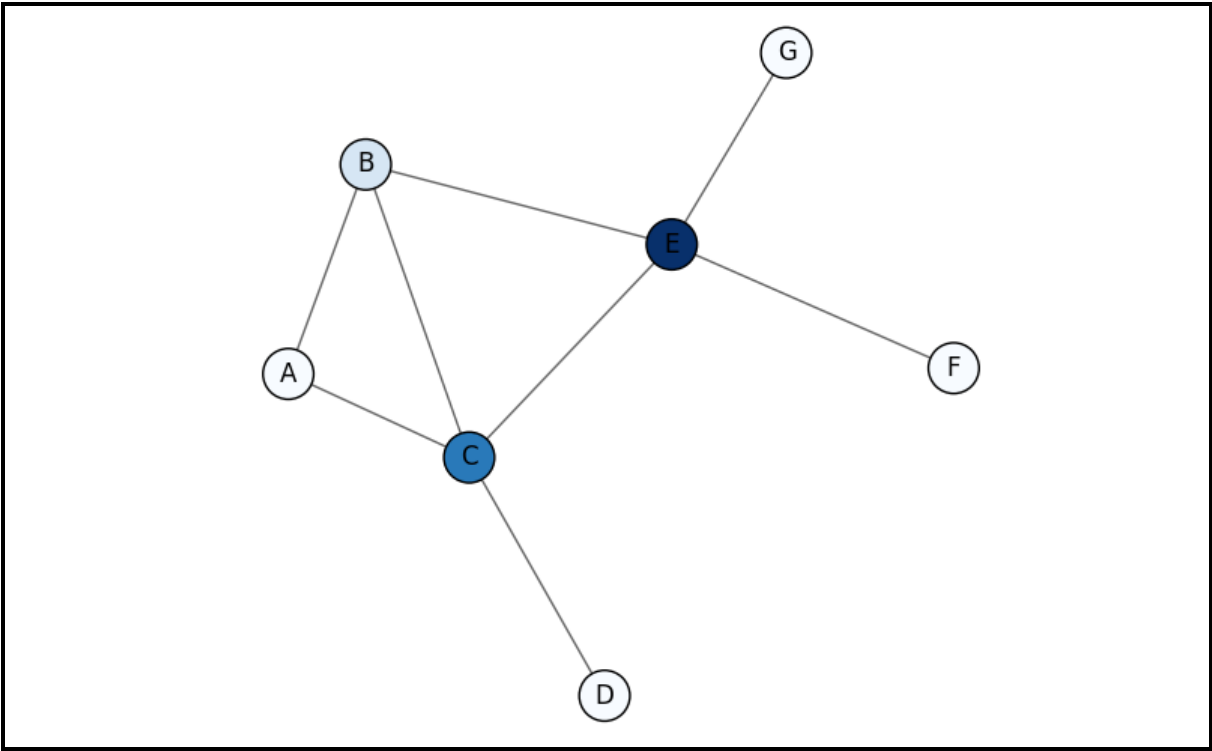
	A	B	C	D
A	-	1	1	0
B	1	-	1	0
C	1	1	-	1
D	0	0	1	-



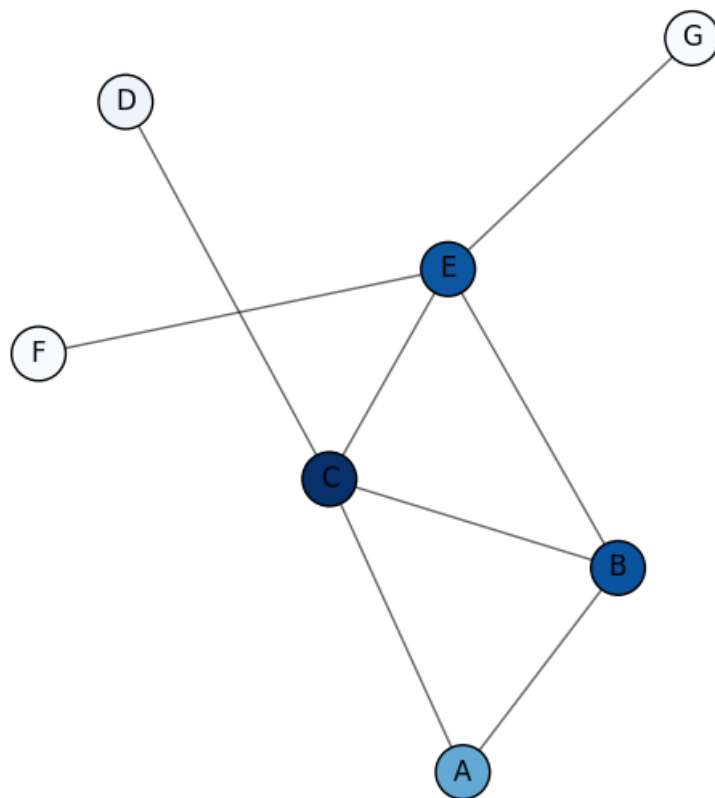


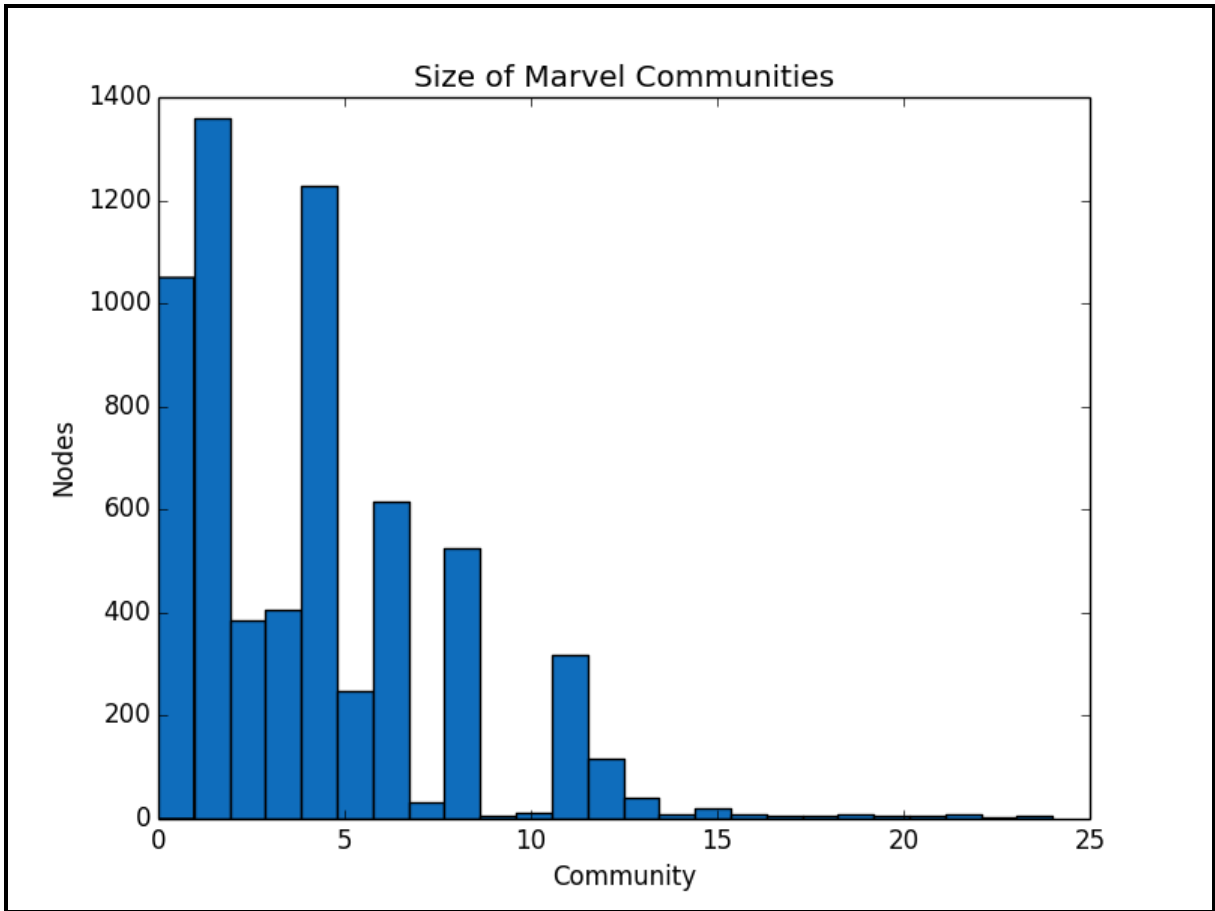
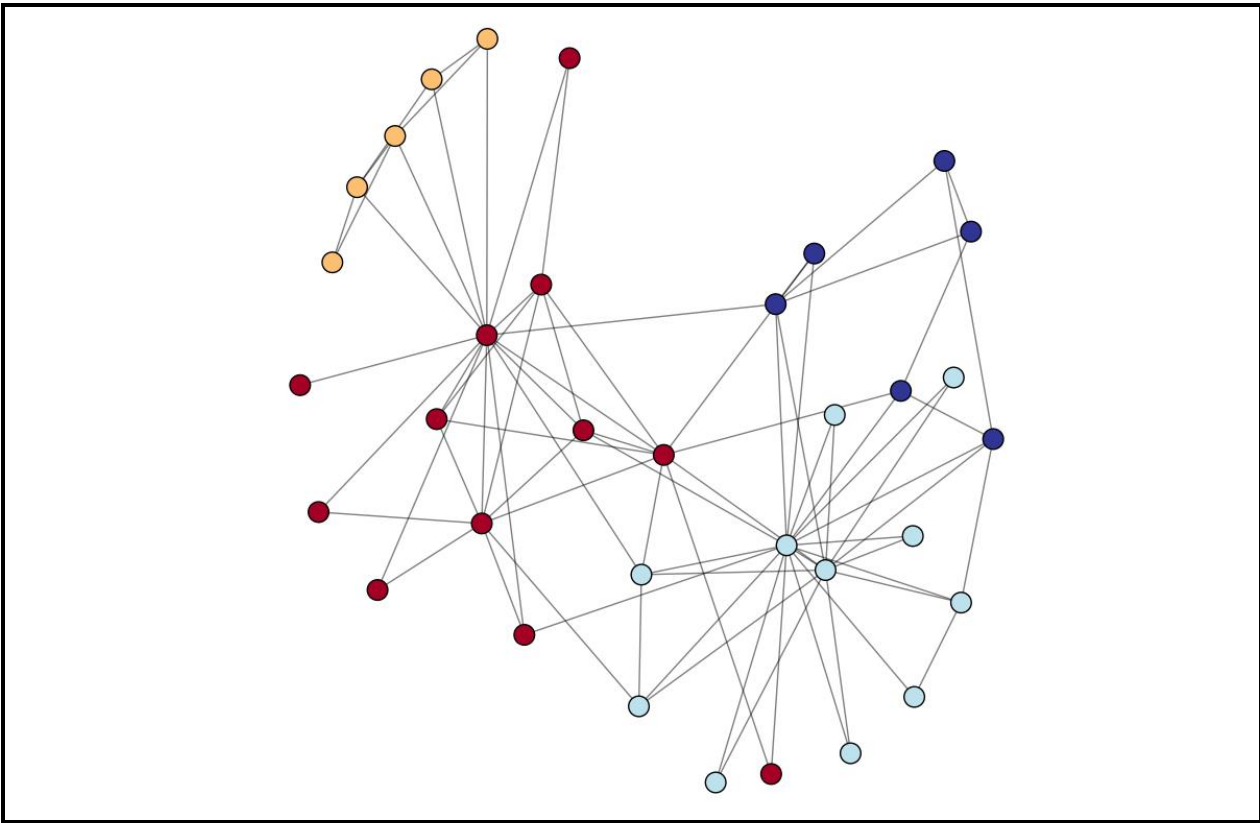


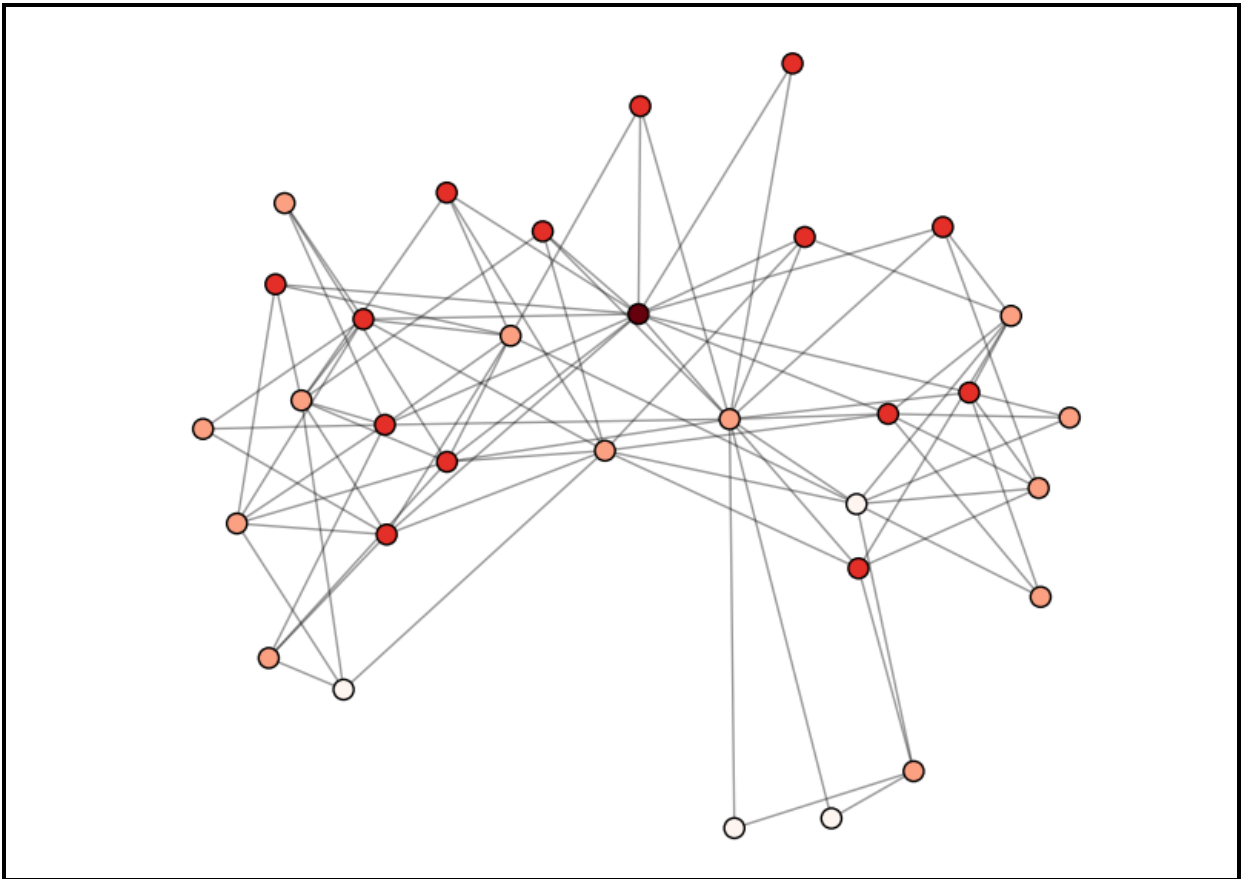
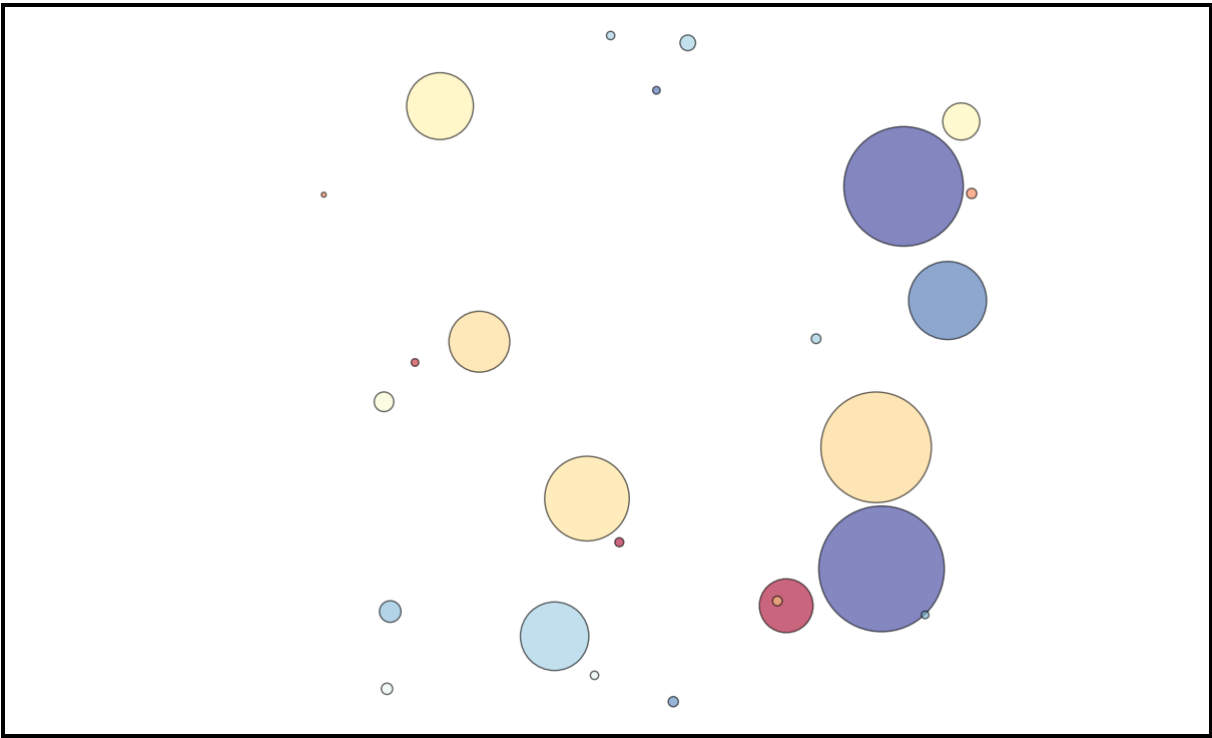




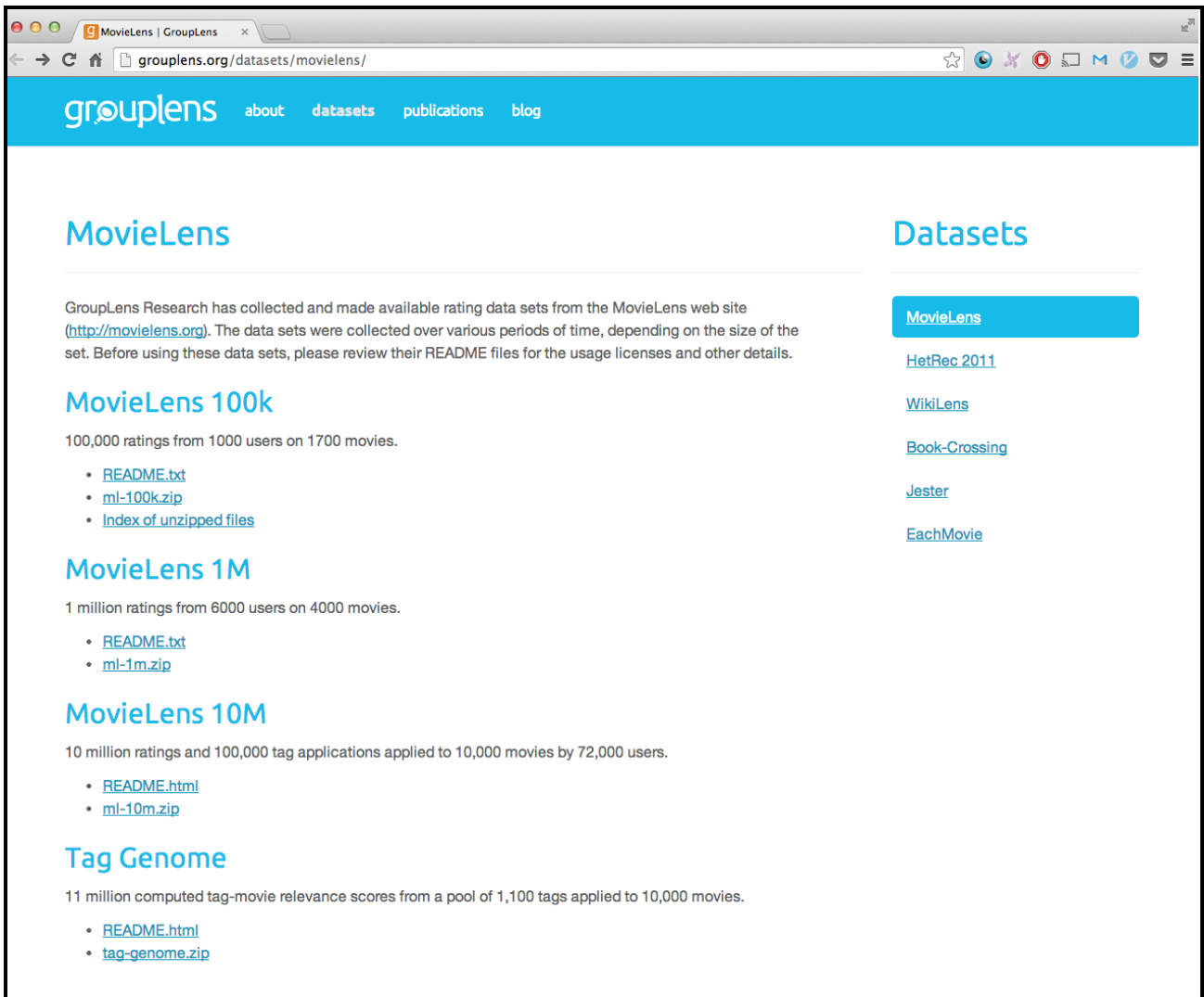
	A	B	C	D	E	F	G
A	-	1	1	0	0	0	0
B	1	-	1	0	1	0	0
C	1	1	-	1	1	0	0
D	0	0	1	-	0	0	0
E	0	1	1	0	-	1	1
F	0	0	0	0	1	-	0
G	0	0	0	0	1	0	-







Chapter 9: Recommending Movies at Scale (Python)



The screenshot shows the MovieLens website on a browser. The page has a blue header with the 'grouplens' logo and navigation links for 'about', 'datasets', 'publications', and 'blog'. The main content is divided into two columns. The left column is titled 'MovieLens' and contains a paragraph about the data collection process, followed by sections for 'MovieLens 100k', 'MovieLens 1M', 'MovieLens 10M', and 'Tag Genome', each with a brief description and a list of links to README files and data zip files. The right column is titled 'Datasets' and features a blue button for 'MovieLens' and links to 'HetRec 2011', 'WikiLens', 'Book-Crossing', 'Jester', and 'EachMovie'.

$$rating = \frac{C \times m \times \sum stars}{C + n}$$

$$Pearson\ Correlation = \frac{cov(A, B)}{\sigma_A \sigma_B}$$

$$\text{Pearson Correlation} = \frac{\sum_{i=1}^n (A_i - \text{mean}(A)) \times \sum_{i=1}^n (B_i - \text{mean}(B))}{\sqrt{\sum_{i=1}^n (A_i - \text{mean}(A))^2} \times \sqrt{\sum_{i=1}^n (B_i - \text{mean}(B))^2}}$$

$$R \approx P \times Q^T = \hat{R}$$

$$e_{ij} = (r_{ij} - \hat{r}_{ij})^2$$

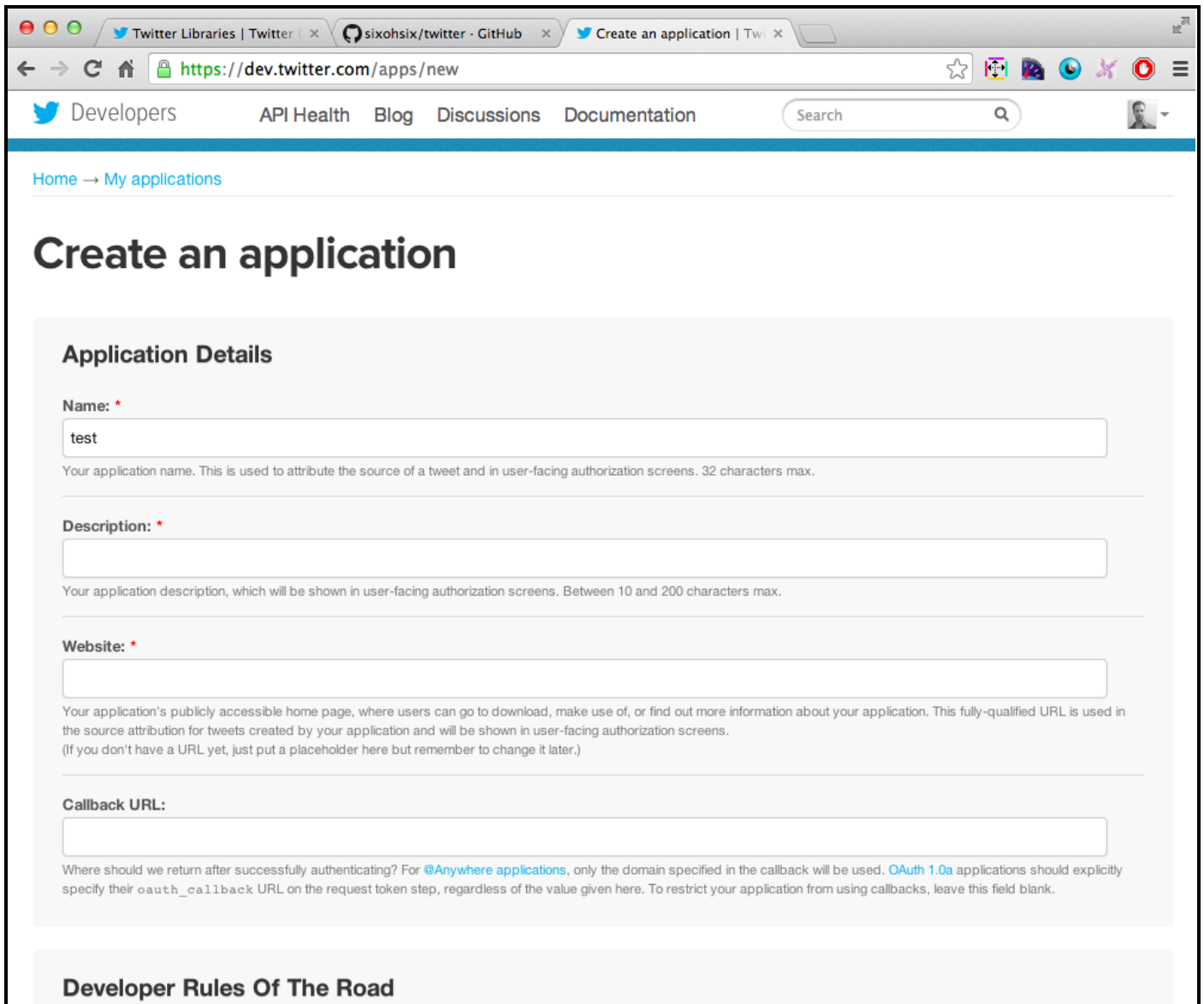
$$\frac{\partial}{\partial p_{ik}} e_{ij} = -2(r_{ij} - \hat{r}_{ij})(q_{kj}) = -2e_{ij}q_{kj}$$

$$\frac{\partial}{\partial q_{ik}} e_{ij} = -2(r_{ij} - \hat{r}_{ij})(p_{ik}) = -2e_{ij}p_{ik}$$

$$p'_{ik} = p_{ik} + \alpha \frac{\partial}{\partial p_{ik}} e_{ij} = p_{ik} + 2\alpha e_{ij} q_{kj}$$

$$q'_{kj} = q_{kj} + \alpha \frac{\partial}{\partial q_{kj}} e_{ij} = q_{kj} + 2\alpha e_{ij} p_{ik}$$

Chapter 10: Harvesting and Geolocating Twitter Data (Python)



The screenshot shows a web browser window with the URL `https://dev.twitter.com/apps/new`. The page title is "Create an application" and it is part of the "My applications" section. The main heading is "Create an application". Below this, there is a section titled "Application Details" with four form fields:

- Name:** * (required) - Input field contains "test". Below it: "Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max."
- Description:** * (required) - Empty input field. Below it: "Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max."
- Website:** * (required) - Empty input field. Below it: "Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)"
- Callback URL:** - Empty input field. Below it: "Where should we return after successfully authenticating? For @Anywhere applications, only the domain specified in the callback will be used. OAuth 1.0a applications should explicitly specify their `oauth_callback` URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank."

At the bottom of the page, there is a section titled "Developer Rules Of The Road".

Sean_test_app2 | Twitter

https://apps.twitter.com/app/6099242/keys

Application Management

Sean_test_app2

Test OAuth

Details Settings **API Keys** Permissions

Application settings

Keep the "API secret" a secret. This key should never be human-readable in your application.

API key	zL5wUPhZZBiDLF4sjk5Pmbpkf
API secret	bvmcEsBYsG6FmkP58TpzQxDxc9liXr2qWOxgTYaxooA6EeBaYr
Access level	Read-only (modify app permissions)
Owner	sayhitosean
Owner ID	57840341

Application actions

Regenerate API keys Change App Permissions

Your access token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.

Token actions

Create my access token

Sean_test_app2 | Twitter

https://apps.twitter.com/app/6099242/keys

Sean_test_app2

Test OAuth

Details Settings API Keys Permissions

Application settings

Keep the "API secret" a secret. This key should never be human-readable in your application.

API key	zL5wUPhZZBiDLf4sjk5Pmbpkf
API secret	bvmcEsBYsG6FmkP58TpzQxDxc9iIXr2qWOxgTYaxooA6EeBaYr
Access level	Read-only (modify app permissions)
Owner	sayhitosean
Owner ID	57840341

Application actions

Regenerate API keys Change App Permissions

Your access token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access token	57840341-RUGnfl9fYVb386NRrgmOPxy6L4EwR6m7r33ygdqA3
Access token secret	wVTIqT6V2dapfoq9iY2HwRVRrNlzBKty39DjMDKwT9gzm
Access level	Read-only
Owner	sayhitosean
Owner ID	57840341

Token actions

Regenerate my access token Revoke token access


```

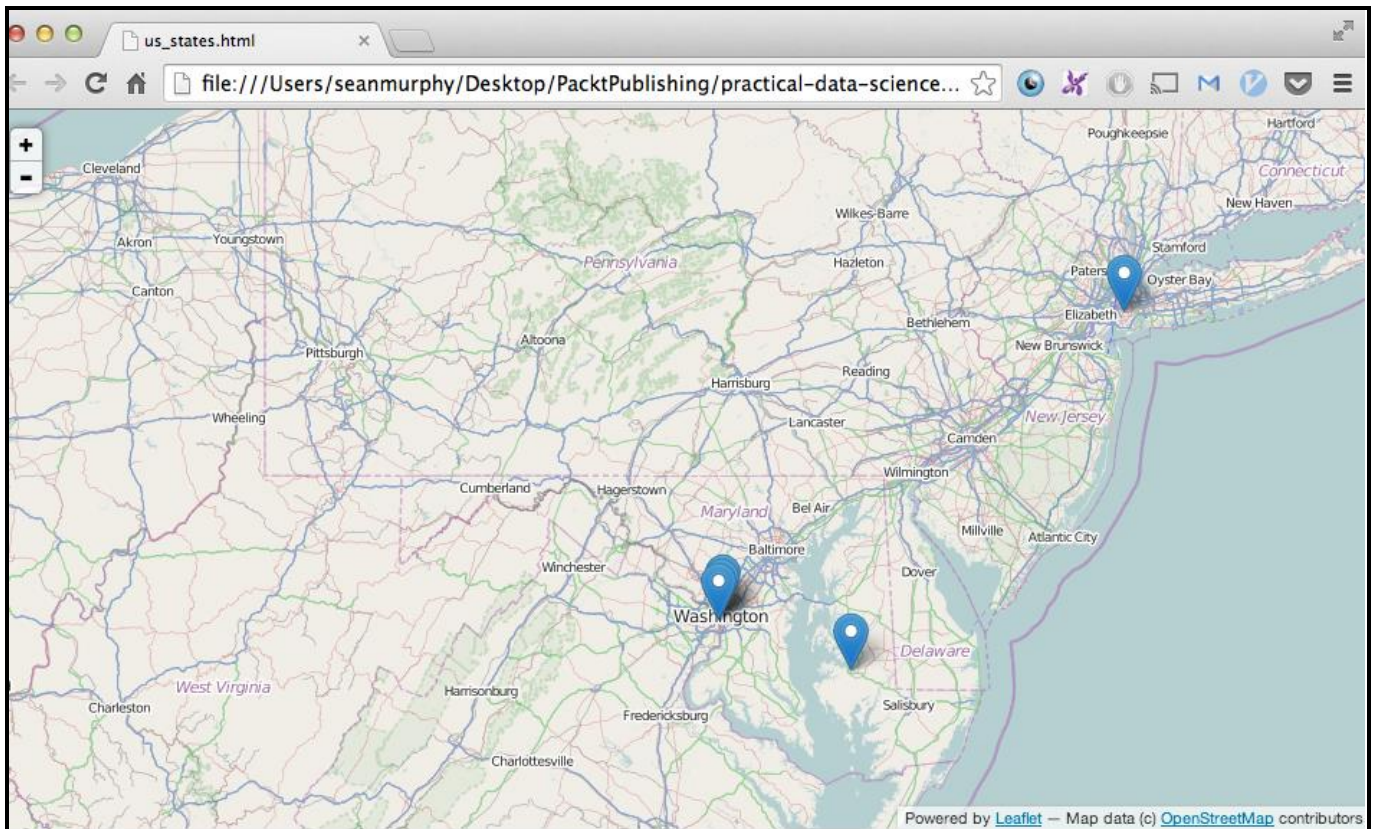
In [7]: twitter.
twitter.access_token
twitter.access_token_url
twitter.add_list_member
twitter.api_url
twitter.api_version
twitter.app_key
twitter.app_secret
twitter.authenticate_url
twitter.client
twitter.client_args
twitter.construct_api_url
twitter.create_block
twitter.create_favorite
twitter.create_friendship
twitter.create_list
twitter.create_list_members
twitter.create_place
twitter.create_saved_search
twitter.cursor
twitter.delete_list
twitter.delete_list_member
twitter.delete_list_members
twitter.destroy_block
twitter.destroy_direct_message
twitter.destroy_favorite
twitter.destroy_friendship
twitter.destroy_saved_search
twitter.destroy_status
twitter.encode
twitter.get
twitter.get_account_settings
twitter.get_application_rate_limit_status
twitter.get_authentication_tokens
twitter.get_available_trends
twitter.get_closest_trends
twitter.get_contributees
twitter.get_contributors
twitter.get_direct_message
twitter.get_direct_messages
twitter.get_favorites
twitter.get_followers_ids
twitter.get_followers_list
twitter.get_friends_ids
twitter.get_friends_list
twitter.get_geo_info
twitter.get_home_timeline
twitter.get_incoming_friendship_ids
twitter.get_lastfunction_header
twitter.get_list_members
twitter.get_list_memberships
twitter.get_list_statuses
twitter.get_list_subscribers
twitter.get_list_subscriptions
twitter.get_mentions_timeline
twitter.get_oembed_tweet
twitter.get_outgoing_friendship_ids
twitter.get_place_trends
twitter.get_privacy_policy
twitter.get_profile_banner_sizes
twitter.get_retweeters_ids
twitter.get_retweets
twitter.get_saved_searches
twitter.get_sent_messages
twitter.get_similar_places
twitter.get_specific_list
twitter.get_supported_languages
twitter.get_tos
twitter.get_twitter_configuration
twitter.get_user_ids_of_blocked_retweets
twitter.get_user_suggestions
twitter.get_user_suggestions_by_slug
twitter.get_user_suggestions_statuses_by_slug
twitter.get_user_timeline
twitter.html_for_tweet
twitter.invalidate_token
twitter.is_list_member
twitter.is_list_subscriber
twitter.list_block_ids
twitter.list_blocks
twitter.lookup_friendships
twitter.lookup_user
twitter.oauth_token
twitter.oauth_token_secret
twitter.oauth_version
twitter.obtain_access_token
twitter.post
twitter.remove_profile_banner
twitter.report_spam
twitter.request
twitter.request_token_url
twitter.retweet
twitter.retweeted_of_me
twitter.reverse_geocode
twitter.search
twitter.search_gen
twitter.search_geo
twitter.search_users
twitter.send_direct_message
twitter.show_friendship
twitter.show_lists
twitter.show_owned_lists
twitter.show_saved_search
twitter.show_status
twitter.show_user
twitter.subscribe_to_list
twitter.unicode2utf8
twitter.unsubscribe_from_list
twitter.update_account_settings
twitter.update_delivery_service
twitter.update_friendship
twitter.update_list
twitter.update_profile
twitter.update_profile_background_image
twitter.update_profile_banner_image
twitter.update_profile_colors
twitter.update_profile_image
twitter.update_status
twitter.update_status_with_media
twitter.verify_credentials

```

```

Number of Twitter users who either are our friend or follow you (union):
980
Number of Twitter users who follow you and are your friend (intersection):
205
Number of Twitter users you follow that don't follow you (set difference):
354
Number of Twitter users who follow you that you don't follow (set difference):
421

```



Chapter 11: Optimizing Numerical Code with NumPy and SciPy (Python)

```
3. seanmurphy@SPM: ~/Desktop/PacktPublishing/practical-data-science/eleven/asa (zsh)
12078.1 angstrom squared.
49755288 function calls in 65.221 seconds

Ordered by: standard name
ncalls  tottime  percall  cumtime  percall  filename:lineno(function)
  1  0.000  0.000  0.000  0.000  <string>:1(<module>)
  1  0.000  0.000  0.000  0.000  __future__.py:48(<module>)
  1  0.000  0.000  0.000  0.000  __future__.py:74(_Feature)
  7  0.000  0.000  0.000  0.000  __future__.py:75(__init__)
  1  0.004  0.004  65.221  65.221  asa.py:10(<module>)
  1  0.002  0.002  0.003  0.003  asa.py:17(generate_sphere_points)
2302  4.727  0.002  13.759  0.006  asa.py:33(find_neighbor_indices)
  1  26.332  26.332  65.134  65.134  asa.py:50(calculate_asa)
  1  0.001  0.001  65.214  65.214  asa.py:94(main)
  1  0.000  0.000  0.000  0.000  cProfile.py:5(<module>)
  1  0.000  0.000  0.000  0.000  cProfile.py:66(Profile)
  1  0.000  0.000  0.000  0.000  getopt.py:15(<module>)
  1  0.000  0.000  0.000  0.000  getopt.py:38(GetoptError)
  1  0.000  0.000  0.000  0.000  getopt.py:51(getopt)
  1  0.001  0.001  0.001  0.001  hashlib.py:55(<module>)
  6  0.000  0.000  0.000  0.000  hashlib.py:94(__get_openssl_constructor)
  1  0.000  0.000  0.000  0.000  molecule.py:1(<module>)
  1  0.001  0.001  0.001  0.001  molecule.py:108(add_radii)
  1  0.000  0.000  0.000  0.000  molecule.py:133(Molecule)
  1  0.000  0.000  0.077  0.077  molecule.py:135(__init__)
  1  0.000  0.000  0.000  0.000  molecule.py:144(atoms)
  1  0.000  0.000  0.000  0.000  molecule.py:150(clear)
2302  0.002  0.000  0.002  0.000  molecule.py:159(insert_atom)
  1  0.007  0.007  0.077  0.077  molecule.py:169(read_pdb)
2302  0.042  0.000  0.065  0.000  molecule.py:41(AtomFromPdbLine)
  1  0.000  0.000  0.000  0.000  molecule.py:5(Atom)
2302  0.016  0.000  0.019  0.000  molecule.py:6(__init__)
  1  0.000  0.000  0.000  0.000  posixpath.py:127(dirname)
  1  0.000  0.000  0.000  0.000  posixpath.py:68(join)
  1  0.000  0.000  0.000  0.000  random.py:100(seed)
  1  0.002  0.002  0.002  0.002  random.py:40(<module>)
  1  0.000  0.000  0.000  0.000  random.py:649(WichmannHill)
  1  0.000  0.000  0.000  0.000  random.py:72(Random)
  1  0.000  0.000  0.000  0.000  random.py:799(SystemRandom)
  1  0.000  0.000  0.000  0.000  random.py:91(__init__)
5296902  3.571  0.000  8.909  0.000  vector3d.py:134(pos_distance)
32378818  25.560  0.000  25.560  0.000  vector3d.py:138(pos_distance_sq)
  1  0.000  0.000  0.000  0.000  vector3d.py:16(Vector3d)
4605  0.002  0.000  0.002  0.000  vector3d.py:18(__init__)
  1  0.000  0.000  0.000  0.000  vector3d.py:197(Matrix3d)
  1  0.001  0.001  0.004  0.004  vector3d.py:3(<module>)
2302  0.001  0.000  0.001  0.000  vector3d.py:49(set)
  1  0.000  0.000  0.000  0.000  {_hashlib.openssl_md5}
  1  0.000  0.000  0.000  0.000  {_hashlib.openssl_sha1}
  1  0.000  0.000  0.000  0.000  {_hashlib.openssl_sha224}
  1  0.000  0.000  0.000  0.000  {_hashlib.openssl_sha256}
  1  0.000  0.000  0.000  0.000  {_hashlib.openssl_sha384}
  1  0.000  0.000  0.000  0.000  {_hashlib.openssl_sha512}
  1  0.000  0.000  0.000  0.000  {binascii.hexlify}
  1  0.000  0.000  0.000  0.000  {eval}
  1  0.000  0.000  0.000  0.000  {function seed at 0x10c4d9140}
  6  0.000  0.000  0.000  0.000  {getattr}
  6  0.000  0.000  0.000  0.000  {globals}
6933  0.003  0.000  0.003  0.000  {len}
```

```

3. seanmurphy@SPM: ~/Desktop/PacktPublishing/practical-data-science/eleven/asa (zsh)
.ce/eleven/asa (zsh)  ..data-science (zsh)
→ asa git:(master) python -m line_profiler asa.py.lprof
Timer unit: 1e-06 s

File: asa.py
Function: calculate_asa at line 48
Total time: 309.418 s

Line #      Hits          Time Per Hit   % Time  Line Contents
-----
48           @profile
49           def calculate_asa(atoms, probe, n_sphere_point=960):
50             """
51             Returns list of accessible surface areas of the atoms, using the probe
52             and atom radius to define the surface.
53             """
54             1          3352   3352.0    0.0    sphere_points = generate_sphere_points(n_sphere_point)
55
56             1           3     3.0    0.0    const = 4.0 * math.pi / len(sphere_points)
57             1           5     5.0    0.0    test_point = Vector3d()
58             1           1     1.0    0.0    areas = []
59             2303        6931     3.0    0.0    for i, atom_i in enumerate(atoms):
60
61                 2302       26792447  11638.8    8.7        neighbor_indices = find_neighbor_indices(atoms, probe, i)
62                 2302        11042     4.8    0.0        n_neighbor = len(neighbor_indices)
63                 2302        3152     1.4    0.0        j_closest_neighbor = 0
64                 2302        3888     1.7    0.0        radius = probe + atom_i.radius
65
66                 2302        2854     1.2    0.0        n_accessible_point = 0
67                 2212222       2738037    1.2    0.9        for point in sphere_points:
68                     2209920       2724147    1.2    0.9            is_accessible = True
69
70                     2209920       4044723    1.8    1.3            test_point.x = point[0]*radius + atom_i.pos.x
71                     2209920       3620414    1.6    1.2            test_point.y = point[1]*radius + atom_i.pos.y
72                     2209920       3606615    1.6    1.2            test_point.z = point[2]*radius + atom_i.pos.z
73
74                     2209920       5969487    2.7    1.9            cycled_indices = range(j_closest_neighbor, n_neighbor)
75                     2209920       6354112    2.9    2.1            cycled_indices.extend(range(j_closest_neighbor))
76
77                     27184207       32946560    1.2   10.6            for j in cycled_indices:
78                         27081916       36679189    1.4   11.9                atom_j = atoms[neighbor_indices[j]]
79                         27081916       39541780    1.5   12.8                r = atom_j.radius + probe
80                         27081916       97550238    3.6   31.5                diff_sq = pos_distance_sq(atom_j.pos, test_point)
81                         27081916       36229084    1.3   11.7                if diff_sq < r*r:
82                             2107629       2504817    1.2    0.8                    j_closest_neighbor = j
83                             2107629       2647353    1.3    0.9                    is_accessible = False
84                             2107629       2645371    1.3    0.9                    break
85                         2209920       2650233    1.2    0.9                if is_accessible:
86                             102291        133391    1.3    0.0                    n_accessible_point += 1
87
88                 2302        4193     1.8    0.0                area = const*n_accessible_point*radius*radius
89                 2302        4480     1.9    0.0                areas.append(area)
90             1           2     2.0    0.0    return areas

→ asa git:(master) >

```

```
→ asa git:(master) x python -m line_profiler asa_np_v1.py.lprof
Timer unit: 1e-06 s
```

```
File: asa_np_v1.py
Function: find_neighbor_indices at line 31
Total time: 37.1705 s
```

Line #	Hits	Time	Per Hit	% Time	Line Contents
31					@profile
32					def find_neighbor_indices(atoms, probe, k):
33					"""
34					Returns list of indices of atoms within probe distance to atom k.
35					"""
36	2302	3492	1.5	0.0	neighbor_indices = []
37	2302	2143	0.9	0.0	atom_k = atoms[k]
38	2302	2710	1.2	0.0	radius = atom_k.radius + probe + probe
39	2302	28179	12.2	0.1	indices = range(k)
40	2302	40282	17.5	0.1	indices.extend(range(k+1, len(atoms)))
41	5299204	4324380	0.8	11.6	for i in indices:
42	5296902	4708375	0.9	12.7	atom_i = atoms[i]
43	5296902	22013127	4.2	59.2	dist = pos_distance(atom_k.pos, atom_i.pos)
44	5296902	5924582	1.1	15.9	if dist < radius + atom_i.radius:
45	98310	121311	1.2	0.3	neighbor_indices.append(i)
46	2302	1891	0.8	0.0	return neighbor_indices

```
→ asa git:(master) x python kernprof.py -l asa_np_v1.py "1R0R.pdb"
12114.7 angstrom squared.
```

```
Wrote profile results to asa_np_v1.py.lprof
```

```
→ asa git:(master) x python -m line_profiler asa_np_v1.py.lprof
Timer unit: 1e-06 s
```

```
File: asa_np_v1.py
Function: find_neighbor_indices at line 18
Total time: 0.346843 s
```

Line #	Hits	Time	Per Hit	% Time	Line Contents
18					@profile
19					def find_neighbor_indices(points, radii, probe, k):
20					"""
21					Returns list of indices of atoms within probe distance to atom k.
22					"""
23	2302	12455	5.4	3.6	radius = radii[k] + probe + probe
24	2302	36325	15.8	10.5	test_radii = (radius + radii) ** 2
25					
26	2302	234933	102.1	67.7	dist_sq = np.sum(((points - points[k,:])** 2), axis=1)
27					
28	2302	19970	8.7	5.8	neighbor_indices = (dist_sq < test_radii)
29					
30					#Must remove self distance
31	2302	3478	1.5	1.0	neighbor_indices[k] = False
32					
33					#Need to return a list/array of integers not a boolean array
34	2302	39682	17.2	11.4	return (np.where(neighbor_indices)[0])