

Name : Snehanshu Shankar

**Statricks Project (Web Scraping using Python) + Data Cleaning,
Analyzing & Visualizing the Data collected from
“<http://www.boattrader.com>”**

Index

1.	Introduction:	2
2.	Python Code:	2
3.	Console Output:	5
4.	CSV Output:	6
5.	Data Cleaning:	7
6.	Data Analysis :	8
6.1	Model 1:	8
6.2	Model2:	8
6.3	Model3:	9
7.	Data Visualization:	10
8.	Conclusion:	14

1. Introduction:

In the teamlead "Statricks Project", I have wrote a web scraper in python for <http://www.boattrader.com> website, which automates the below mentioned task as per asked in the Leada project.

To find the URL for each and every boat being listed on the entire website.

For every boat, fetch and output the following info into a csv file:

- Boat Maker/Model
- Seller Contact Number
- Price

In addition to above mentioned attributes, I have also gathered **Year, Boat Category and Boat Class** information for each and every boat. Finally, I have used R for data cleaning, data analysis & visualization.

2. Python Code:

The below mentioned python code is written to fetch the all the individual boat details from the web (<http://www.boattrader.com>). The search for the boat of any type and category results in 137257 web pages. Further, each page has 28 or more listing of boats. The below code starts to gather data from the first page and then iterates till the end of the pages.

To run the script with available python environment and libraries, one needs only to change the path for the directory where the csv file will be created and make sure indentation is correct. While loop is used to iterate over the different web pages and for loop is used to iterate over each individual boat in a page. Further, code is explained with the mentioned comments.

```
# In Python
import requests
#import bs4 # import main module first
from bs4 import BeautifulSoup, SoupStrainer
import os, csv
```

```

# Given a raw url, returns the html of the page.
def get_raw_html(url):
    r = requests.get(url)
    ad_page_html = r.text
    return ad_page_html
# Variable to store start point.
page = 1
count_per_page = 28
os.chdir("C:\Users\sneh\Desktop\python_intro")
with open("BoatDetails.csv", "wb") as toWrite:
    writer = csv.writer(toWrite)
#Below code creates the column header in csv excel file.
    writer.writerow(["BOAT URL", "PRICE", "MAKE", "CONTACT", "BOAT CLASS", "BOAT
CATEGORY", "YEAR"])

#while loop which iterates each page for listing all the boats on a page.
while page <= 137257:
    print page
    url_boat = "http://www.boattrader.com/search-results/NewOrUsed-any/Type-any/Category-
all/Zip-33613/Radius-4000/Sort-Length:DESC/Page-%s,28?" %page
    #Code to call the method which generates raw HTML for a URL.
    raw_html = get_raw_html(url_boat)
    #Soup object is created for operations on data.
    soup = BeautifulSoup(raw_html, "html.parser")
    #for loop which iterates for collecting data from individual boats.
    for boat in soup.find_all('a', {'data-reporting-click-listing-type': 'standard listing'}):
        # followed by finding the <a> for each of the <a>
        links = boat.find_all('a', href=True)
        # Check for data availability.
        if len(links) != 0:
            link = links[0]
            url = link['href'].encode('utf-8')
            print "-----Page"+str(page)+"-----"
            print url
            individual_page_html= get_raw_html("http:"+url)
            soup1 = BeautifulSoup(individual_page_html, "html.parser")
            make = soup1.find_all('span', {'class': 'bd-make'})
            # Check for data availability, if data is not available, NA will be stored in the make field.
            if len(make) !=0:
                makeBoat = make[0].get_text().strip() #strip() method is used to trim spaces.
            else:
                makeBoat = "NA"

```

```

year = soup1.find_all('span', {'class': 'bd-year'})
# Check for data availability, if data is not available, NA will be stored in the year.
if len(year)!=0:
    yearBoat = year[0].get_text().strip()
else:
    yearBoat = "NA"
indvDetailContact = soup1.find_all('a', {'class': 'phone'})
if len(indvDetailContact) !=0:
    contact = indvDetailContact[0].get_text().strip()
else:
    contact = "NA"
sellerPrice = soup1.find_all('span', {'class': 'bd-price contact-toggle'})
# Check for data availability, if data is not available, NA will be stored.
if len(sellerPrice)!=0:
    price = sellerPrice[0].get_text().strip()
else:
    price = "NA"
boattype = soup1.find_all('div', {'class': 'collapsible open'})
# Check for data availability, if data is not available, NA will be stored..
if len(boattype) !=0:
    tab = boattype[0].find_all('td')
    boatClass = tab[0].get_text().strip()
    boatCategory = tab[1].get_text().strip()
else:
    boatClass = "NA"
    boatCategory = "NA"
model = "Make : " + makeBoat+ " / Year : " + yearBoat

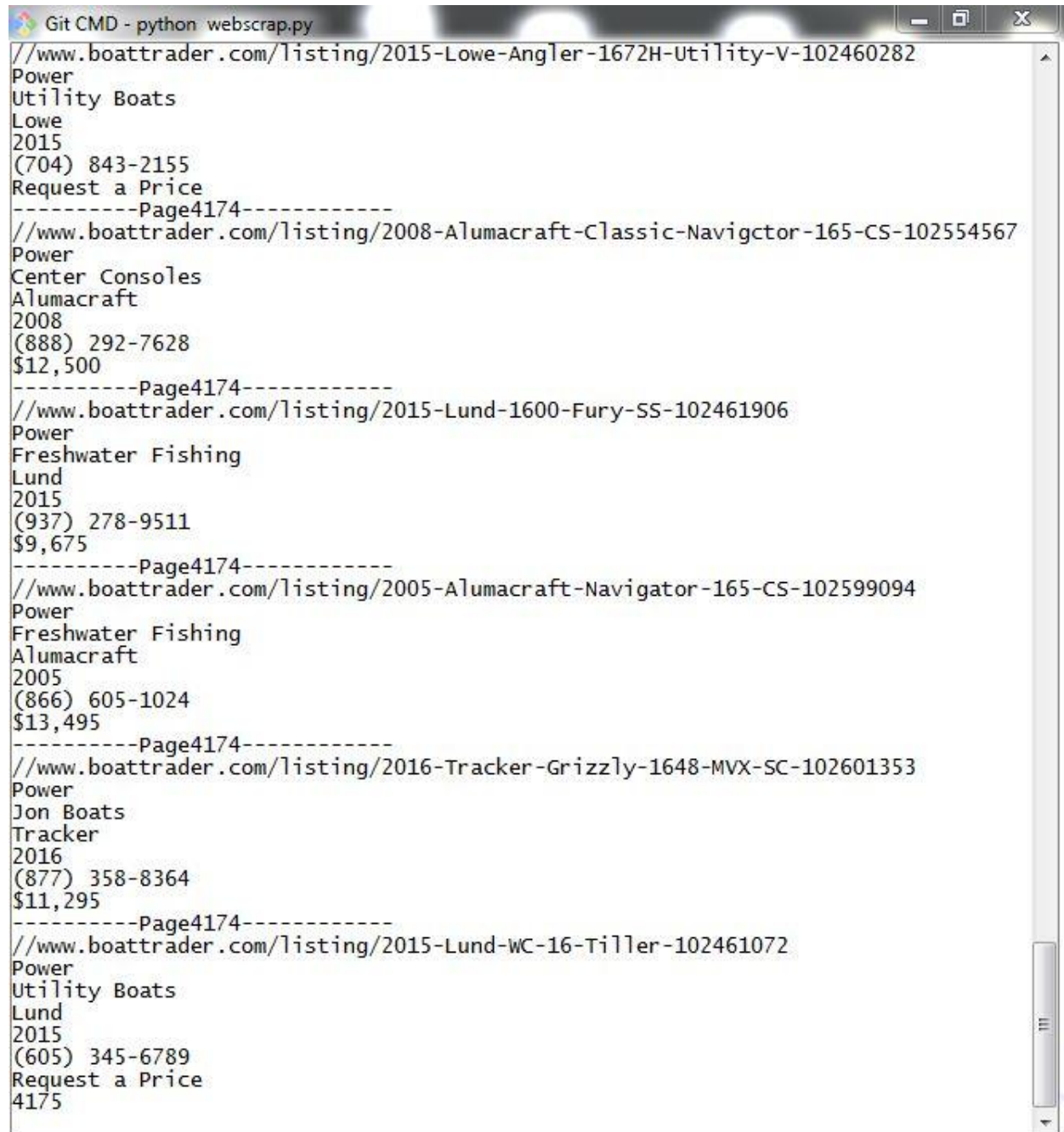
#Print Statements to check values on console for output.
print boatClass
print boatCategory
print makeBoat
print yearBoat
print contact
print price

#Below code is to write each row (individual boat data) to the csv file.
writer.writerow([url,price,makeBoat, contact, boatClass,boatCategory,yearBoat])
# Increment the page for next web page.
page = page + 1

```

3. Console Output:

Below is the sample output for my python script execution on console.



```
Git CMD - python webscrap.py
//www.boattrader.com/listing/2015-Lowe-Angler-1672H-Utility-V-102460282
Power
Utility Boats
Lowe
2015
(704) 843-2155
Request a Price
-----Page4174-----
//www.boattrader.com/listing/2008-Alumacraft-Classic-Navigtor-165-CS-102554567
Power
Center Consoles
Alumacraft
2008
(888) 292-7628
$12,500
-----Page4174-----
//www.boattrader.com/listing/2015-Lund-1600-Fury-SS-102461906
Power
Freshwater Fishing
Lund
2015
(937) 278-9511
$9,675
-----Page4174-----
//www.boattrader.com/listing/2005-Alumacraft-Navigator-165-CS-102599094
Power
Freshwater Fishing
Alumacraft
2005
(866) 605-1024
$13,495
-----Page4174-----
//www.boattrader.com/listing/2016-Tracker-Grizzly-1648-MVX-SC-102601353
Power
Jon Boats
Tracker
2016
(877) 358-8364
$11,295
-----Page4174-----
//www.boattrader.com/listing/2015-Lund-WC-16-Tiller-102461072
Power
Utility Boats
Lund
2015
(605) 345-6789
Request a Price
4175
```

Figure: Output of command execution on Console.

I also launched multiple instances of Git console for parallel data collection from web. Below is the image to show two launched Git console instances.

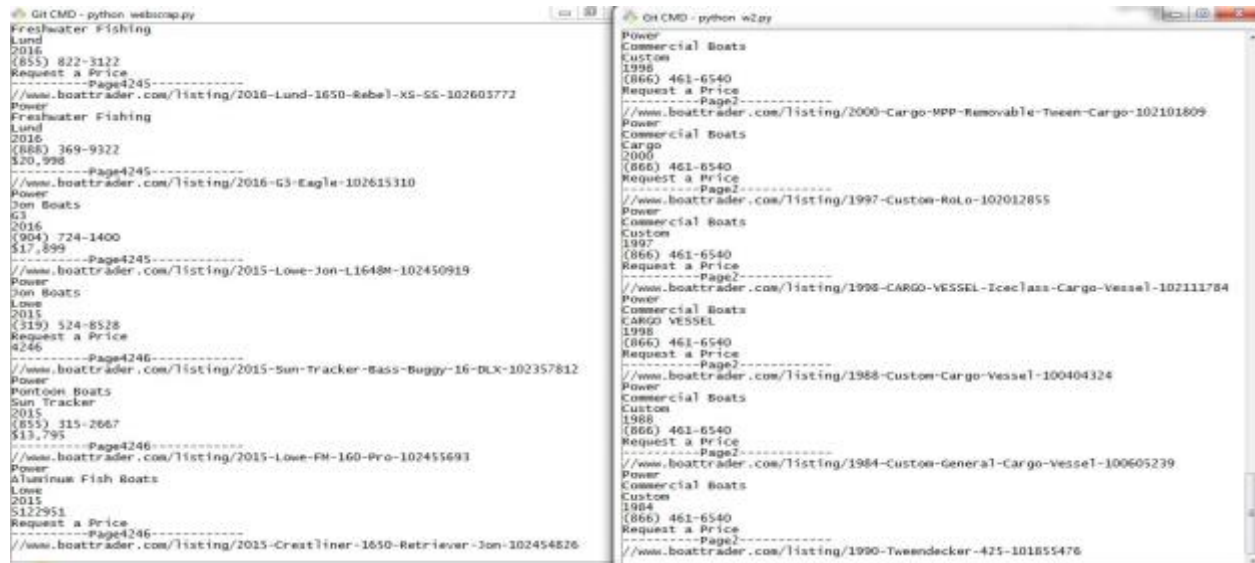


Figure: Two instances launched for fast retrieval of data from web.

4. CSV Output:

Below is mentioned csv file that generated after execution of the above python script.

1	BOAT URL	PRICE	MAKE	CONTACT	BOAT CLASS	BOAT CATEGORY	YEAR
2	//www.boattrader.com/listing/2015-Crestlin	Request a Price	Crestliner	(903) 363-8990	Power	Aluminum Fish Boats	2015
3	//www.boattrader.com/listing/2016-Lund-1	\$18,795	Lund	(888) 235-6516	Power	Freshwater Fishing	2016
4	//www.boattrader.com/listing/2015-Lund-1	Request a Price	Lund	(218) 385-2855	Power	Freshwater Fishing	2015
5	//www.boattrader.com/listing/2015-Carolin	\$14,292	Carolina Skiff	(888) 351-3975	Power	Center Consoles	2015
6	//www.boattrader.com/listing/2016-Mako-J	\$31,595	Mako	(252) 977-2191	Power	Center Consoles	2016
7	//www.boattrader.com/listing/2015-Lowe-F	Request a Price	Lowe	(973) 398-0251	Power	Jon Boats	2015
8	//www.boattrader.com/listing/2015-Key-W	Request a Price	Key West	(855) 822-8291	Power	Center Consoles, Spor	2015
9	//www.boattrader.com/listing/2016-Mako-J	\$27,995	Mako	(252) 977-2191	Power	Center Consoles	2016
10	//www.boattrader.com/listing/2009-Duracr	\$10,500	Duracraft	(877) 634-5582	Power	Aluminum Fish Boats,	2009
11	//www.boattrader.com/listing/2016-Tracker	\$10,595	Tracker	(855) 315-4532	Power	Jon Boats	2016
12	//www.boattrader.com/listing/2015-Lowe-S	Request a Price	Lowe	(931) 762-6710	Power	Bass Boats	2015
13	//www.boattrader.com/listing/2007-Lowe-F	\$9,000	Lowe	(602) 399-9367	Power	Fish And Ski	2007
14	//www.boattrader.com/listing/2011-Lowe-S	Request a Price	Lowe	(605) 996-0337	Power	Bass Boats	2011
15	//www.boattrader.com/listing/2013-Carolin	\$10,995	Carolina Skiff	(888) 620-4044	Power	Saltwater Fishing, Cer	2013
16	//www.boattrader.com/listing/2016-Tracker	\$3,799	Tracker	(334) 794-2598	Power	Jon Boats	2016
17	//www.boattrader.com/listing/2015-Rogue-	\$15,665	Rogue Marine	(866) 465-6585	Power	Aluminum Fish Boats	2015
18	//www.boattrader.com/listing/2016-Tracker	\$14,295	Tracker	(855) 315-3698	Power	Jon Boats	2016
19	//www.boattrader.com/listing/2015-Lowe-S	Request a Price	Lowe	(877) 246-0500	Power	Bass Boats	2015
20	//www.boattrader.com/listing/2015-Lund-1	Request a Price	Lund	(207) 746-5621	Power	Freshwater Fishing	2015
21	//www.boattrader.com/listing/2015-Cast-ar	Request a Price	Cast and blast	(877) 782-7267	Power	Aluminum Fish Boats	2015
22	//www.boattrader.com/listing/2016-Tracker	\$10,595	Tracker	(888) 830-6194	Power	Jon Boats	2016
23	//www.boattrader.com/listing/1997-Seaswi	\$9,999	Seaswirl Stripe	(888) 732-9115	Power	Center Consoles	1997
24	//www.boattrader.com/listing/2015-Crestlin	Request a Price	Crestliner	(765) 647-4619	Power	Bass Boats	2015
25	//www.boattrader.com/listing/2016-Bennin	Request a Price	Bennington	(888) 872-1284	Power	Pontoon Boats	2016
26	//www.boattrader.com/listing/2016-Lowe-S	Request a Price	Lowe	(866) 270-9985	Power	Bass Boats, Freshwate	2016
27	//www.boattrader.com/listing/2015-Crestlin	Request a Price	Crestliner	(409) 769-9890	Power	Bass Boats	2015
28	//www.boattrader.com/listing/2015-Lowe-F	Request a Price	Lowe	(541) 882-5834	Power	Jon Boats	2015

Figure: CSV file output; data scrapped from the web (<http://www.boattrader.com>).

5. Data Cleaning:

I have fetched 7201 boat records from the website. The data set is further cleaned using R. All the rows where “Request a Price” is mentioned, that means that seller has no disclosed or mentioned the price on the website. In order to know the price, one has to contact the seller directly. All such rows will have to be eliminated for the analysis, including the blank data or NA’s from the rows of any field. 6172 rows have been retained after cleaning operation using below mentioned R code.

```
boatData <- read.csv("C:/Users/sneh/Desktop/python_intro/BoatDataset/BoatDetails.csv")
attach(boatData)
#In below code, I have taken 2016 as a latest boat model as a base and calculated how old a boat is in
#Year column.
boat<- data.frame(Price=boatData$PRICE,
Make=boatData$MAKE,Class=boatData$BOAT.CLASS,Category=boatData$BOAT.CATEGORY, Year=2016-
boatData$YEAR)
attach(boat)
#Removing “Request a Price” and “NA” values from the rows.
boatClean1<- subset(boat, (boat$Price != NA | boat$Price!="Request a Price"))
#Removing “$” from the price.
boatClean2<- data.frame(Price=gsub("$", "", boatClean1$Price,fixed=TRUE), boatClean1[,2:5])
#Removing “,” from the price.
CleanedBoatData<- data.frame(Price= as.numeric(gsub(",", "", boatClean2$Price)), boatClean2[,2:5])
```

CleanedBoatData is the dataset for further analysis.

```
> summary(CleanedBoatData)
```

Price		Make		Class		Category		Year	
Min.	: 299	Yamaha	:1152	Power	:1570	Personal watercraft:	1726	Min.	: 0.00
1st Qu.:	10999	Sea-Doo	: 377	PWC	:1785	Cruisers	: 415	1st Qu.:	1.00
Median :	30228	Hunter	: 236	Sails	:1905	House Boats	: 413	Median :	9.00
Mean :	334260	Beneteau:	215	Small Boats:	912	Motor Yachts	: 284	Mean :	13.87
3rd Qu.:	179000	Catalina:	149			Kayak	: 162	3rd Qu.:	25.00
Max.	:123456792	Allmand :	146			Sloop	: 138	Max.	:126.00
		(other)	:3897			(other)	:3034		

Figure: Summary of Cleaned Data Set.

6. Data Analysis :

6.1 Model 1:

In the below regression model, I have taken Price as an independent variable and class a dependent variable to see the price trend as per boat class.

```
> model1<-lm(CleanedBoatData$Price~ CleanedBoatData$Class)
> summary(model1)

> model1<-lm(CleanedBoatData$Price~ CleanedBoatData$Class)
> summary(model1)

Call:
lm(formula = CleanedBoatData$Price ~ CleanedBoatData$Class)

Residuals:
    Min       1Q   Median       3Q      Max
-1121113  -107941  -11315    2157 122334679

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    1122113     47278   23.73  <2e-16 ***
CleanedBoatData$ClassPWC -1111171     64816  -17.14  <2e-16 ***
CleanedBoatData$ClassSails -986273     63854  -15.45  <2e-16 ***
CleanedBoatData$ClassSmall Boats -1096865     77994  -14.06  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1873000 on 6168 degrees of freedom
Multiple R-squared:  0.05761, Adjusted R-squared:  0.05715
F-statistic: 125.7 on 3 and 6168 DF, p-value: < 2.2e-16
```

Figure: Model 1

In the above model, R squared is only approx 5%. Therefore, it is not an efficient model and the variation in price is not explained by boat class in itself.

6.2 Model2:

In this regression model, I have taken Price as an independent variable, Class and Year as a dependent variable to see the price trend as per boat class and Year.

```
> model2<-lm(CleanedBoatData$Price~ CleanedBoatData$Year +
CleanedBoatData$Class)
> summary(model2)
```



```
> model2<-lm(CleanedBoatData$Price~ CleanedBoatData$Year + CleanedBoatData$Class)
> summary(model2)
```

Call:
lm(formula = CleanedBoatData\$Price ~ CleanedBoatData\$Year + CleanedBoatData\$Class)

Residuals:

Min	1Q	Median	3Q	Max
-1467461	-219790	-31887	40793	122253483

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	1486456	61514	24.16	<2e-16	***
CleanedBoatData\$Year	-20225	2205	-9.17	<2e-16	***
CleanedBoatData\$ClassPWC	-1426046	72968	-19.54	<2e-16	***
CleanedBoatData\$ClassSails	-831055	65648	-12.66	<2e-16	***
CleanedBoatData\$ClassSmall Boats	-1371638	83066	-16.51	<2e-16	***

signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1861000 on 6167 degrees of freedom
Multiple R-squared: 0.07029, Adjusted R-squared: 0.06968
F-statistic: 116.6 on 4 and 6167 DF, p-value: < 2.2e-16

Figure: Model 2

In the above model, again R squared is only approx 7%. Therefore, it is also not an efficient model and the variation in price is not explained by boat class and year.

6.3 Model3:

In the below non parametric regression model, I have taken log(Price) as an independent variable, Class and Year as a dependent variable to see the price trend as per boat class and Year. The price of the boat of all classes decreases with the increase in years. P values of the attributes are also significant to justify the outcome analysis.

```
> model3<-lm(log(CleanedBoatData$Price)~ CleanedBoatData$Year +
CleanedBoatData$Class)
> summary(model3)
> model3<-lm(log(CleanedBoatData$Price)~ CleanedBoatData$Year + CleanedBoatData$Class)
> summary(model3)
```

Call:
lm(formula = log(CleanedBoatData\$Price) ~ CleanedBoatData\$Year +
CleanedBoatData\$Class)

Residuals:

Min	1Q	Median	3Q	Max
-5.9050	-0.4385	0.0154	0.4665	6.3209

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	13.665732	0.035099	389.35	<2e-16	***
CleanedBoatData\$Year	-0.038770	0.001258	-30.81	<2e-16	***
CleanedBoatData\$ClassPWC	-4.333149	0.041634	-104.08	<2e-16	***
CleanedBoatData\$ClassSails	-1.550035	0.037458	-41.38	<2e-16	***
CleanedBoatData\$ClassSmall Boats	-4.619620	0.047396	-97.47	<2e-16	***

signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.062 on 6167 degrees of freedom
Multiple R-squared: 0.7014, Adjusted R-squared: 0.7012
F-statistic: 3622 on 4 and 6167 DF, p-value: < 2.2e-16

Figure: Model 3

In the above model, R squared is approx 70%. It is comparatively a very efficient model over previous models in which the 70% variation is explained by boat class and year.

Below is the model where boat class is a factor with base of boat class "Power".

Model: $\log(\text{Price}) = 13.67 - 0.04 * \text{Year} - 4.33 * \text{ClassPWC} - 1.55 * \text{ClassSails} - 4.62 * \text{ClassSmallBoats}$.

For Class Power : $\log(\text{Price}) = 13.67 - 0.04 * \text{Year}$. { ClassPWC , ClassSails, ClassSmallBoats =0}

For Class Sail : $\log(\text{Price}) = 13.67 - 0.04 * \text{Year} - 1.55 * \text{ClassSails}$. { ClassPWC , ClassSmallBoats =0}

For Class Small Boat : $\log(\text{Price}) = 13.67 - 0.04 * \text{Year} - 4.62 * \text{ClassSmallBoats}$. { ClassPWC , ClassSails =0}

For Class PWC : $\log(\text{Price}) = 13.67 - 0.04 * \text{Year} - 4.33 * \text{ClassPWC}$. { ClassSails, ClassSmallBoats =0}

We can interpret that the dependent variable (price) changes by 100*(coefficient) % for a 1 unit increase in the independent variable when all other variable in the model are kept constant. we'd say that a increase of one year of the age of boat would results in a 3 percent change in the boat price.

7. Data Visualization:

```
> plot(CleanedBoatData$Class, log(CleanedBoatData$Price))
```

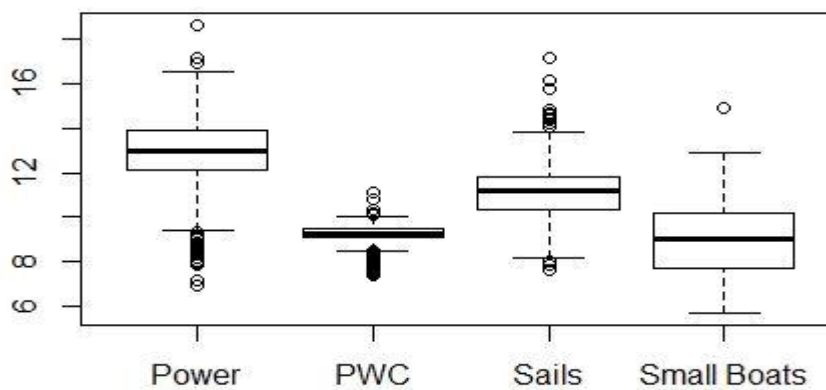


Figure: log(Price) vs Class Plot

Below is ggplot diagram to show the identity bar plot of boat price variation as per year, more older the less price it has.

```
> g<-ggplot(CleanedBoatData,aes(Year,Price)) +geom_bar(stat = "identity")
> g
```

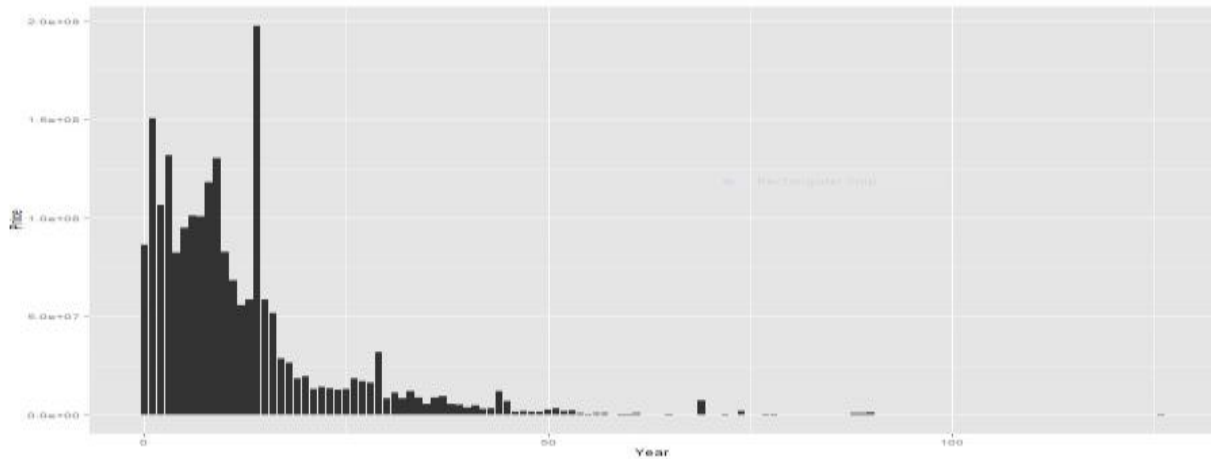


Figure: log(Price) vs Boat Class Plot

Clustering

Below is the clustering diagram using “fpc” library and R code to show 4 clusters, specifically according to class of boat (Power, Sail, PWC & Small boats).

```
> d<-data.frame(CleanedBoatData$Class, log(CleanedBoatData$Price),
CleanedBoatData$Year)
> boatDta <- d[-1]
> kMeanfit <- kmeans(boatDta, 4)
> library(fpc)
> plotcluster(boatDta, kMeanfit$cluster)
```

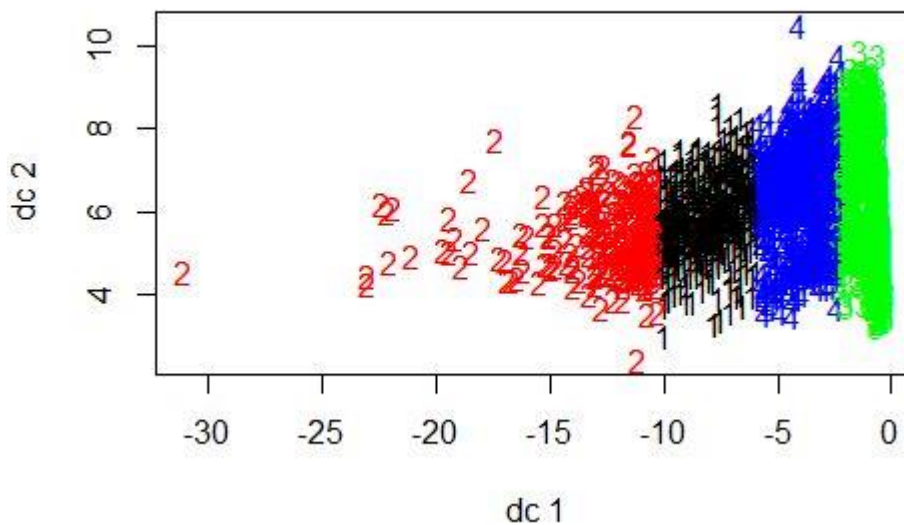


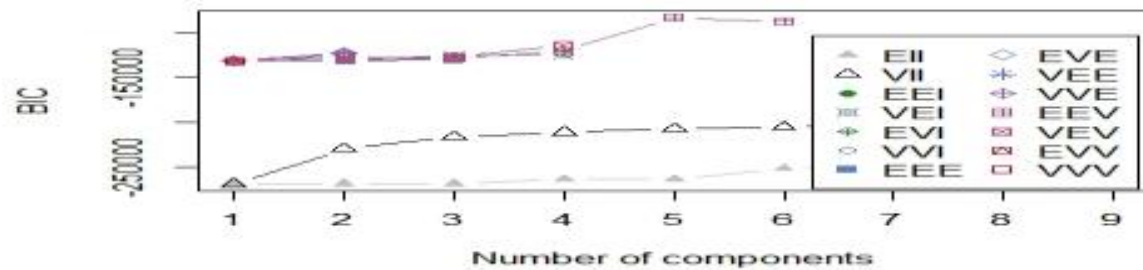
Figure: Clustering Diagram

Model-based clustering plots are described below:

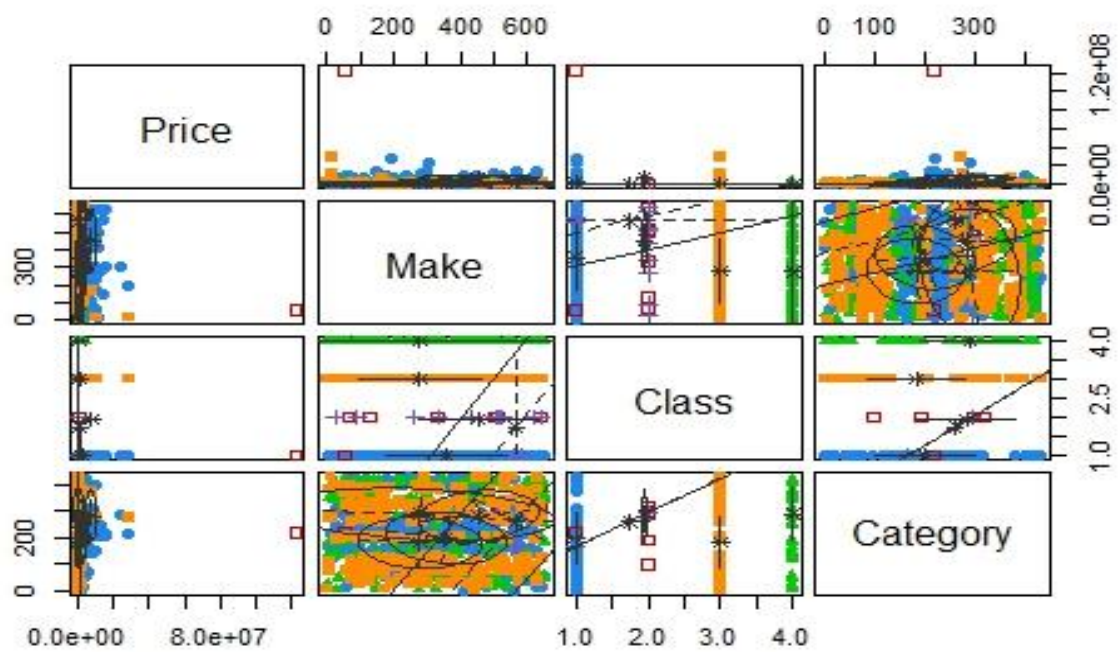
R Code:

```
> library(CleanedBoatData)
```

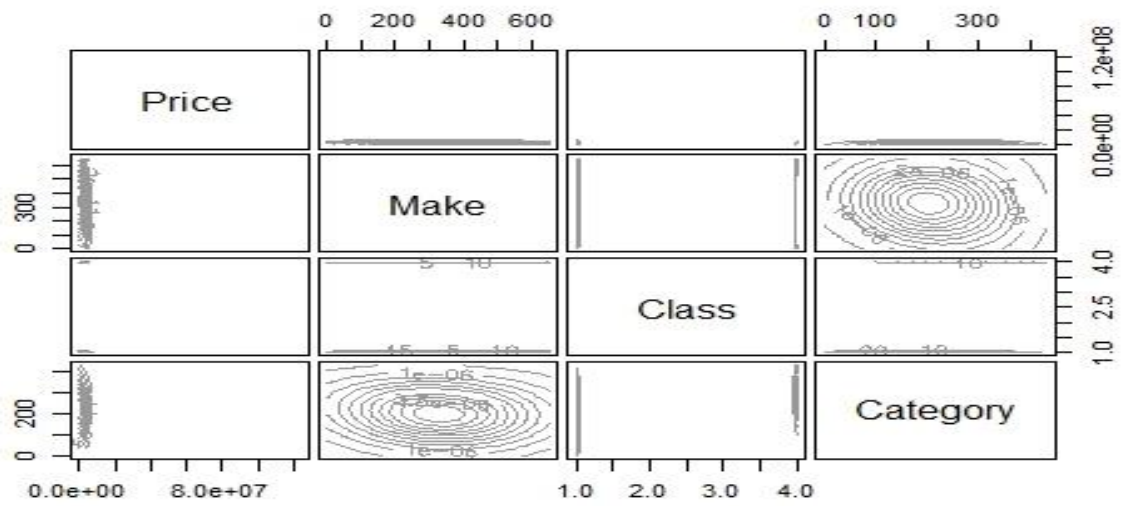
```
> fit <- Mclust(CleanedBoatData[,1:4])
> plot(fit)# plot results
1: BIC
```



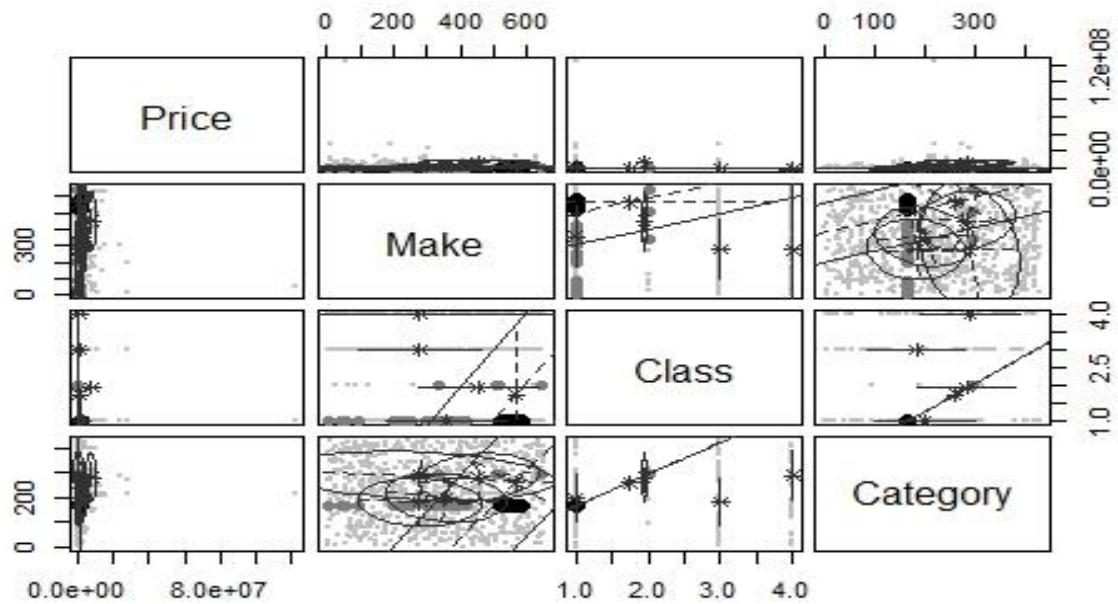
2: classification



3: density



4: uncertainty



8. Conclusion:

I have successfully found and stored the URL into a csv file for each and every boat being listed on the entire website (<http://www.boattrader.com>).

For every boat, I have included Boat Model, Price, Seller Contact, Year, Boat Category and Boat Class. Furthermore, I have used R for data cleaning, data Analysis & Visualization of the “Boattrader” data set.