Final project

## Group Members

Baskar Dakshinamoorthy, Guillermo Gonzalez, Thomas Freitas, Yen Yung Geszvain

## Project Summary

Corey Oil, LTD. is an oil distributor/wholesaler located in Wisconsin. 40 years of oil selling history has contributed a highly loyal customer group. The dedication to providing product service, knowledge, and competitive prices promotes excellent customer relationship and stable selling history. However, the challenge is to identify the strength of the products. For example, what type of oil is selling and what products are not.

This group project aims to prepare and collect a dataset that is valuable for data and business analysis. The goal of the project is to turn the data into valuable proposals and suggestions via descriptive analytics, predictive analytics, and quantitative analytics.

## Dataset

The team has selected an invoice dataset from the Corey Oil database.

## Developing Questions

1. What is the dataset?
2. What is the value of the dataset? Why is it important?
3. How can we turn raw data into readable analysis and reports?
4. How can we improve the business operations with the analysis and recommendation? What are the business areas being affected for the improvement?
5. What effect will the project impact the business results?

## Data Gathering

We collect the invoice data from Corey Oil database. We filtered the dataset to include only 2018 invoice data. As the market evolves, we want to ensure that we analyze the data which is current and up to date.

Corey\_Oil <- **read.csv**("Corey\_Oil\_dataset\_0909.csv", header=TRUE,sep=";")  
**summary**(Corey\_Oil)

## TCode Refn CNUM   
## : 3 Min. : 25 Min. : 18.24   
## - Unlined : 5 1st Qu.: 90408 1st Qu.:103585.00   
## (12-15 oz): 2 Median : 97392 Median :106965.00   
## CM : 148 Mean : 134530 Mean :108428.98   
## DM : 7 3rd Qu.: 105917 3rd Qu.:111143.00   
## IN :14648 Max. :3129700 Max. :160149.00   
##   
## SHIPDATE DUEDATE PNUM   
## 6/7/2018 0:00:00 : 220 6/3/2018 0:00:00 : 172 2UL/RF :2095   
## 5/24/2018 0:00:00: 205 3/2/2018 0:00:00 : 167 12HS :1449   
## 1/4/2018 0:00:00 : 192 6/17/2018 0:00:00: 157 22LS :1071   
## 1/15/2018 0:00:00: 191 6/8/2018 0:00:00 : 155 1379B : 746   
## 5/8/2018 0:00:00 : 186 5/3/2018 0:00:00 : 153 12LS : 631   
## 5/23/2018 0:00:00: 183 7/12/2018 0:00:00: 153 5W30SB : 471   
## (Other) :13636 (Other) :13856 (Other):8350   
## PNAME1 PRICE   
## 87 Unleaded Reformulated Gasoline - Cardlock :2095 Min. : 0.000   
## #2 Ultra Low Dyed Diesel :1449 1st Qu.: 2.330   
## #2 Ultra Low Sulfur On Road Diesel - Cardlock :1071 Median : 2.590   
## Rain Guard Windshield Wash Pre-Mixed -25\xba F: 746 Mean : 38.194   
## #2 Ultra Low Sulfur On Road Diesel : 631 3rd Qu.: 8.568   
## Severe Duty 5W30 Synthetic Blend : 471 Max. :4758.600   
## (Other) :8350 NA's :183   
## COST TotalSales MEAS UM   
## 2.25 : 442 0.00 : 523 Min. : 1.00 G :11447   
## 2.04 : 350 4.64 : 260 1st Qu.: 1.00 E : 3163   
## 0.00 : 322 5.46 : 218 Median : 1.00 : 193   
## 2.06 : 241 : 183 Mean : 44.51 P : 5   
## 2.16 : 237 3.31 : 152 3rd Qu.: 1.00 Q : 2   
## 1.65 : 224 3.37 : 138 Max. :53225.00 10o : 1   
## (Other):12997 (Other):13339 NA's :183 (Other): 2   
## UP QSHIP STNAME   
## BULK :9203 Min. : 0.01 Cardlock Credit Card Customer: 1869   
## DRUM :2140 1st Qu.: 2.00 Groundskeepers : 434   
## CASE : 805 Median : 16.60 Fred`s Sanitary Service LLC : 216   
## PAIL : 586 Mean : 124.47 Evergreen Property Management: 145   
## ITEM : 497 3rd Qu.: 114.00 Auto Parts & Service : 135   
## Box : 420 Max. :12448.72 Arbor Experts : 133   
## (Other):1162 NA's :193 (Other) :11881   
## STCITY STZIP   
## North Lake:2476 53064 :2454   
## Hartland :1871 53029 :1855   
## Milwaukee :1338 53066 : 546   
## Waukesha : 675 53186 : 400   
## Oconomowoc: 532 53072 : 382   
## Pewaukee : 377 53916 : 373   
## (Other) :7544 (Other):8803

## Data Preprocessing

Before preprocessing, the Corey Oil dataset had 28000 rows and 32 variables. All these data sets contained information regarding various attributes of Corey Oil. The data sets contained numerous NA values, which were best dealt with by omitting them, as working on substantial values makes more sense than assuming some calculated values such as mean. Only the columns that are relevant to the analysis of data have been kept, while the remaining columns were eliminated from the dataset. After preprocessing the data we ended up with 14620 rows and 17 Variables. In summary, the data preprocessing phase provides a subset of usable data for the project, consisting of only the columns that need to be worked with and eliminating Null/NA values.

# Following is the code for data preprocessing

*#Check if there is any NAs*  
**any**(**is.na**(Corey\_Oil))

## [1] TRUE

*#Remove Columns that have empty Rows*  
Corey\_Oil <- **na.omit**(Corey\_Oil)  
*#Total Number of NAs in each column*  
**colSums**(**is.na**(Corey\_Oil))

## TCode Refn CNUM SHIPDATE DUEDATE PNUM   
## 0 0 0 0 0 0   
## PNAME1 PRICE COST TotalSales MEAS UM   
## 0 0 0 0 0 0   
## UP QSHIP STNAME STCITY STZIP   
## 0 0 0 0 0

*#Remove decimal points*  
Corey\_Oil**$**Refn <- **as.character**(**gsub**(".00", "", Corey\_Oil**$**Refn))  
Corey\_Oil**$**CNUM <- **as.character**(**gsub**(".00", "", Corey\_Oil**$**CNUM))  
*#Subsetting only UM = G that are needed for our analysis to a new dataframe and where Total Amount is Not null*  
Corey\_Oil\_Analysis <- **sqldf**("select \* from Corey\_Oil where UM = 'G'")  
**head.matrix**(Corey\_Oil\_Analysis,n=10)

## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 IN 85273 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 2 IN 85276 105386 1/1/2018 0:00:00 1/1/2018 0:00:00 2UL/RF  
## 3 IN 85272 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 4 IN 85269 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 5 IN 85268 101 1/1/2018 0:00:00 1/31/2018 0:00:00 22LS  
## 6 IN 85267 105047 1/1/2018 0:00:00 2/15/2018 0:00:00 2UL/RF  
## 7 IN 320125 101384 1/2/2018 0:00:00 1/12/2018 0:00:00 12HS  
## 8 IN 320132 105175 1/2/2018 0:00:00 1/12/2018 0:00:00 12HS  
## 9 IN 320131 107785 1/2/2018 0:00:00 1/2/2018 0:00:00 12HS  
## 10 IN 320130 101621 1/2/2018 0:00:00 1/12/2018 0:00:00 12HS  
## PNAME1 PRICE COST TotalSales  
## 1 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 2 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 3 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 4 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 5 #2 Ultra Low Sulfur On Road Diesel - Cardlock 2.24 2.14 4.80  
## 6 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 7 #2 Ultra Low Dyed Diesel 2.39 1.97 4.73  
## 8 #2 Ultra Low Dyed Diesel 2.39 1.97 4.73  
## 9 #2 Ultra Low Dyed Diesel 2.39 1.97 4.73  
## 10 #2 Ultra Low Dyed Diesel 2.39 1.97 4.73  
## MEAS UM UP QSHIP STNAME STCITY STZIP  
## 1 1 G BULK 4.36 Cardlock Credit Card Customer North Lake 53064  
## 2 1 G BULK 33.99 Groundskeepers Hartland 53029  
## 3 1 G BULK 8.10 Cardlock Credit Card Customer North Lake 53064  
## 4 1 G BULK 12.27 Cardlock Credit Card Customer North Lake 53064  
## 5 1 G BULK 19.85 Cardlock Credit Card Customer North Lake 53064  
## 6 1 G BULK 26.89 Evergreen Property Management Hartland 53029  
## 7 1 G BULK 181.90 Sindberg, Dorothy Oconomowoc 53066  
## 8 1 G BULK 220.90 Westhoff, Larry/Maria Ixonia 53036  
## 9 1 G BULK 203.90 Christopherson, Robert Ixonia 53036  
## 10 1 G BULK 210.80 Kailing, Maxine Hartland 53029

*#Total Number of NAs in each column*  
**colSums**(**is.na**(Corey\_Oil\_Analysis))

## TCode Refn CNUM SHIPDATE DUEDATE PNUM   
## 0 0 0 0 0 0   
## PNAME1 PRICE COST TotalSales MEAS UM   
## 0 0 0 0 0 0   
## UP QSHIP STNAME STCITY STZIP   
## 0 0 0 0 0

**sqldf**("select count(\*) from Corey\_Oil\_Analysis")

## count(\*)  
## 1 11447

**str**(Corey\_Oil\_Analysis)

## 'data.frame': 11447 obs. of 17 variables:  
## $ TCode : Factor w/ 6 levels ""," - Unlined",..: 6 6 6 6 6 6 6 6 6 6 ...  
## $ Refn : chr "85273" "85276" "85272" "85269" ...  
## $ CNUM : chr "101" "105386" "101" "101" ...  
## $ SHIPDATE : Factor w/ 163 levels "1/1/2018 0:00:00",..: 1 1 1 1 1 1 12 12 12 12 ...  
## $ DUEDATE : Factor w/ 272 levels "","1.00","1/1/2018 0:00:00",..: 27 3 27 27 27 49 6 6 14 6 ...  
## $ PNUM : Factor w/ 904 levels "","(I)TIM-3628-50",..: 295 295 295 295 239 295 198 198 198 198 ...  
## $ PNAME1 : chr "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" ...  
## $ PRICE : num 1.94 1.94 1.94 1.94 2.24 1.94 2.39 2.39 2.39 2.39 ...  
## $ COST : Factor w/ 1729 levels "","0.00","0.15",..: 88 88 88 88 404 88 110 110 110 110 ...  
## $ TotalSales: Factor w/ 4527 levels "","0.00","0.01",..: 1785 1785 1785 1785 2591 1785 2584 2584 2584 2584 ...  
## $ MEAS : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ UM : Factor w/ 8 levels "","10o","e","E",..: 5 5 5 5 5 5 5 5 5 5 ...  
## $ UP : Factor w/ 26 levels "","1","30DRUM",..: 10 10 10 10 10 10 10 10 10 10 ...  
## $ QSHIP : num 4.36 33.99 8.1 12.27 19.85 ...  
## $ STNAME : Factor w/ 2169 levels "","1:28 Automotive",..: 292 737 292 292 292 557 1741 2091 338 994 ...  
## $ STCITY : chr "North Lake" "Hartland" "North Lake" "North Lake" ...  
## $ STZIP : Factor w/ 400 levels "",".",".53219",..: 89 58 89 89 89 58 92 63 63 58 ...

*#Convert data type - convert the value to characters then numbers*  
Corey\_Oil\_Analysis**$**COST <- **as.numeric**(**as.character**(Corey\_Oil\_Analysis**$**COST))  
*#Data Preview*  
**head.matrix**(Corey\_Oil\_Analysis,n=10)

## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 IN 85273 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 2 IN 85276 105386 1/1/2018 0:00:00 1/1/2018 0:00:00 2UL/RF  
## 3 IN 85272 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 4 IN 85269 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 5 IN 85268 101 1/1/2018 0:00:00 1/31/2018 0:00:00 22LS  
## 6 IN 85267 105047 1/1/2018 0:00:00 2/15/2018 0:00:00 2UL/RF  
## 7 IN 320125 101384 1/2/2018 0:00:00 1/12/2018 0:00:00 12HS  
## 8 IN 320132 105175 1/2/2018 0:00:00 1/12/2018 0:00:00 12HS  
## 9 IN 320131 107785 1/2/2018 0:00:00 1/2/2018 0:00:00 12HS  
## 10 IN 320130 101621 1/2/2018 0:00:00 1/12/2018 0:00:00 12HS  
## PNAME1 PRICE COST TotalSales  
## 1 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 2 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 3 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 4 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 5 #2 Ultra Low Sulfur On Road Diesel - Cardlock 2.24 2.14 4.80  
## 6 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37  
## 7 #2 Ultra Low Dyed Diesel 2.39 1.97 4.73  
## 8 #2 Ultra Low Dyed Diesel 2.39 1.97 4.73  
## 9 #2 Ultra Low Dyed Diesel 2.39 1.97 4.73  
## 10 #2 Ultra Low Dyed Diesel 2.39 1.97 4.73  
## MEAS UM UP QSHIP STNAME STCITY STZIP  
## 1 1 G BULK 4.36 Cardlock Credit Card Customer North Lake 53064  
## 2 1 G BULK 33.99 Groundskeepers Hartland 53029  
## 3 1 G BULK 8.10 Cardlock Credit Card Customer North Lake 53064  
## 4 1 G BULK 12.27 Cardlock Credit Card Customer North Lake 53064  
## 5 1 G BULK 19.85 Cardlock Credit Card Customer North Lake 53064  
## 6 1 G BULK 26.89 Evergreen Property Management Hartland 53029  
## 7 1 G BULK 181.90 Sindberg, Dorothy Oconomowoc 53066  
## 8 1 G BULK 220.90 Westhoff, Larry/Maria Ixonia 53036  
## 9 1 G BULK 203.90 Christopherson, Robert Ixonia 53036  
## 10 1 G BULK 210.80 Kailing, Maxine Hartland 53029

**tail.matrix**(Corey\_Oil\_Analysis,n=10)

## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 11438 IN 108697 103796 9/6/2018 0:00:00 9/16/2018 0:00:00 2490WAY68BK  
## 11439 IN 108695 112082 9/6/2018 0:00:00 9/16/2018 0:00:00 DEF  
## 11440 IN 108692 105926 9/6/2018 0:00:00 10/6/2018 0:00:00 AF110  
## 11441 IN 108685 112258 9/6/2018 0:00:00 9/16/2018 0:00:00 DEF  
## 11442 IN 108682 102490 9/6/2018 0:00:00 9/16/2018 0:00:00 5W20SB  
## 11443 IN 108701 112453 9/6/2018 0:00:00 9/6/2018 0:00:00 5W20SB  
## 11444 IN 107601 112084 9/6/2018 0:00:00 9/16/2018 0:00:00 1379B  
## 11445 IN 107910 111811 9/6/2018 0:00:00 9/16/2018 0:00:00 1379B  
## 11446 IN 303023 102 1/29/2019 0:00:00 2/8/2016 0:00:00 12LS  
## 11447 IN 130 102245 10/22/2019 0:00:00 11/1/2015 0:00:00 12HS  
## PNAME1 PRICE COST TotalSales  
## 11438 Waylube 68 9.70 4.93 47.89  
## 11439 Diesel Exhaust Fluid (DEF) 1.73 1.20 2.08  
## 11440 ThermaCool HD ELC NF Concentrate (red) 8.52 5.57 47.46  
## 11441 Diesel Exhaust Fluid (DEF) 1.24 0.85 1.06  
## 11442 Severe Duty 5W20 Synthetic Blend 6.34 3.58 22.74  
## 11443 Severe Duty 5W20 Synthetic Blend 5.25 3.58 18.83  
## 11444 Rain Guard Windshield Wash Pre-Mixed -25<ba> F 1.27 0.70 0.89  
## 11445 Rain Guard Windshield Wash Pre-Mixed -25<ba> F 1.15 0.70 0.80  
## 11446 #2 Ultra Low Sulfur On Road Diesel 1.35 1.12 1.52  
## 11447 #2 Ultra Low Dyed Diesel 2.41 1.78 4.32  
## MEAS UM UP QSHIP STNAME STCITY STZIP  
## 11438 5 G PAIL 1.0 Kald Tool & Die Corp Richfield 53076  
## 11439 55 G DRUM 3.0 Ball Auto & Truck Parts, Inc Two Rivers 54241  
## 11440 55 G DRUM 1.0 De Angelis Construction, LLC Milwaukee 53225  
## 11441 1 G BULK 240.0 Gleason Roll Off & Recycling Racine 53403  
## 11442 1 G BULK 90.0 Village Auto Repair Hartland 53056  
## 11443 1 G BULK 80.0 Meineke - Racine Racine 53406  
## 11444 1 G BULK 222.6 Kunes Country Ford of Antioch Antioch 60002  
## 11445 1 G BULK 113.2 Illinois Recovery/Kenosha Nissan Kenosha 53104  
## 11446 1 G BULK 35.0 COREY OIL LTD. DELIVERY EXP. BEAVER DAM 53916  
## 11447 1 G BULK 460.0 L F GEORGE INC WAUKESHA 53186

**summary**(Corey\_Oil\_Analysis)

## TCode Refn CNUM   
## : 0 Length:11447 Length:11447   
## - Unlined : 0 Class :character Class :character   
## (12-15 oz): 0 Mode :character Mode :character   
## CM : 36   
## DM : 0   
## IN :11411   
##   
## SHIPDATE DUEDATE PNUM   
## 6/7/2018 0:00:00 : 167 6/3/2018 0:00:00 : 126 2UL/RF :2095   
## 5/24/2018 0:00:00: 165 3/2/2018 0:00:00 : 123 12HS :1449   
## 1/4/2018 0:00:00 : 155 6/8/2018 0:00:00 : 123 22LS :1071   
## 1/15/2018 0:00:00: 152 3/1/2018 0:00:00 : 121 1379B : 746   
## 5/23/2018 0:00:00: 152 6/9/2018 0:00:00 : 121 12LS : 631   
## 5/29/2018 0:00:00: 150 6/22/2018 0:00:00: 120 5W30SB : 471   
## (Other) :10506 (Other) :10713 (Other):4984   
## PNAME1 PRICE COST TotalSales   
## Length:11447 Min. : 0.000 Min. : 0.000 4.64 : 260   
## Class :character 1st Qu.: 2.270 1st Qu.: 1.960 5.46 : 218   
## Mode :character Median : 2.490 Median : 2.170 3.31 : 152   
## Mean : 3.996 Mean : 2.779 3.37 : 138   
## 3rd Qu.: 5.575 3rd Qu.: 3.240 5.74 : 127   
## Max. :939.700 Max. :157.210 5.14 : 119   
## (Other):10433   
## MEAS UM UP QSHIP   
## Min. : 1.000 G :11447 BULK :9203 Min. : 0.02   
## 1st Qu.: 1.000 : 0 DRUM :1695 1st Qu.: 8.00   
## Median : 1.000 10o : 0 PAIL : 334 Median : 31.06   
## Mean : 9.925 e : 0 KEG : 84 Mean : 153.62   
## 3rd Qu.: 1.000 E : 0 Cargo Tank: 51 3rd Qu.: 150.20   
## Max. :330.000 L : 0 30DRUM : 30 Max. :7504.00   
## (Other): 0 (Other) : 50   
## STNAME STCITY   
## Cardlock Credit Card Customer :1815 Length:11447   
## Groundskeepers : 428 Class :character   
## Fred`s Sanitary Service LLC : 212 Mode :character   
## Evergreen Property Management : 139   
## Dousman Transport Co Inc (cardlock): 130   
## Arbor Experts : 129   
## (Other) :8594   
## STZIP   
## 53064 :2298   
## 53029 :1774   
## 53066 : 448   
## 53916 : 314   
## 53186 : 295   
## 53072 : 268   
## (Other):6050

*#this is to correct the totalsales column*  
Corey\_Oil\_Analysis**$**TotalSales <- (Corey\_Oil\_Analysis**$**PRICE **\*** Corey\_Oil\_Analysis**$**MEAS **\*** Corey\_Oil\_Analysis**$**QSHIP)  
Corey\_Oil\_Analysis**$**TotalSales <- **round**(Corey\_Oil\_Analysis**$**TotalSales, 2)  
*#this is to create a totalcost column*  
Corey\_Oil\_Analysis**$**TotalCost <- (Corey\_Oil\_Analysis**$**COST **\*** Corey\_Oil\_Analysis**$**MEAS **\*** Corey\_Oil\_Analysis**$**QSHIP)  
Corey\_Oil\_Analysis**$**TotalCost <- **round**(Corey\_Oil\_Analysis**$**TotalCost, 2)  
*#this is to create a Profit column*  
Corey\_Oil\_Analysis**$**Profit <- (Corey\_Oil\_Analysis**$**TotalSales **-** Corey\_Oil\_Analysis**$**TotalCost)  
Corey\_Oil\_Analysis**$**Profit <- **round**(Corey\_Oil\_Analysis**$**Profit, 2)  
**head**(Corey\_Oil\_Analysis)

## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 IN 85273 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 2 IN 85276 105386 1/1/2018 0:00:00 1/1/2018 0:00:00 2UL/RF  
## 3 IN 85272 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 4 IN 85269 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 5 IN 85268 101 1/1/2018 0:00:00 1/31/2018 0:00:00 22LS  
## 6 IN 85267 105047 1/1/2018 0:00:00 2/15/2018 0:00:00 2UL/RF  
## PNAME1 PRICE COST TotalSales MEAS  
## 1 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 8.46 1  
## 2 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 65.94 1  
## 3 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 15.71 1  
## 4 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 23.80 1  
## 5 #2 Ultra Low Sulfur On Road Diesel - Cardlock 2.24 2.14 44.46 1  
## 6 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 52.17 1  
## UM UP QSHIP STNAME STCITY STZIP TotalCost  
## 1 G BULK 4.36 Cardlock Credit Card Customer North Lake 53064 7.54  
## 2 G BULK 33.99 Groundskeepers Hartland 53029 58.80  
## 3 G BULK 8.10 Cardlock Credit Card Customer North Lake 53064 14.01  
## 4 G BULK 12.27 Cardlock Credit Card Customer North Lake 53064 21.23  
## 5 G BULK 19.85 Cardlock Credit Card Customer North Lake 53064 42.48  
## 6 G BULK 26.89 Evergreen Property Management Hartland 53029 46.52  
## Profit  
## 1 0.92  
## 2 7.14  
## 3 1.70  
## 4 2.57  
## 5 1.98  
## 6 5.65

## Date Grouping

The data were cleaned up in order to show better plots that are associated with specific items. These items consisted of the highest selling, most frequently ordered, most expensive to the company, and the highest profiting products for the company. The analysis of these data is all very similar. We chose to find the top 30 products for each category. The Corey Oil data frame was aggregated and ranked by the highest values and matched to the PNUM. The scatter plots show these values.

# Data Grouping - TOP 30 by Total Sales

Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis[,**c**(6,7,10)]  
**str**(Corey\_Oil\_Analysis\_Rank)

## 'data.frame': 11447 obs. of 3 variables:  
## $ PNUM : Factor w/ 904 levels "","(I)TIM-3628-50",..: 295 295 295 295 239 295 198 198 198 198 ...  
## $ PNAME1 : chr "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" ...  
## $ TotalSales: num 8.46 65.94 15.71 23.8 44.46 ...

Corey\_Oil\_Analysis\_Rank**$**TotalSales <- **as.numeric**(**as.character**(Corey\_Oil\_Analysis\_Rank**$**TotalSales))  
Corey\_Oil\_Analysis\_Rank <- **aggregate**(Corey\_Oil\_Analysis\_Rank**$**TotalSales, **list**(PNUM=Corey\_Oil\_Analysis\_Rank**$**PNUM), sum)  
Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis\_Rank[**order**(**-**Corey\_Oil\_Analysis\_Rank**$**x),]  
  
**head.matrix**(Corey\_Oil\_Analysis\_Rank,n=30)

## PNUM x  
## 22 12HS 1159855.81  
## 23 12LS 1098406.28  
## 74 5W30SB 274142.66  
## 212 METHANOL 203542.60  
## 40 1UL/RF 182298.94  
## 158 DS15W40 163474.05  
## 152 DEF 159153.14  
## 15 110B 139526.02  
## 27 1379B 121668.00  
## 71 5W20SB 116725.58  
## 204 HY46 116545.25  
## 84 622515001097 110029.09  
## 201 HY32 93381.73  
## 126 80565-29811 82262.70  
## 45 22LS 82160.23  
## 90 622723001097 78230.53  
## 58 2UL/RF 76275.34  
## 64 41550 65982.03  
## 75 5W30SYN 63917.40  
## 222 THP 59914.57  
## 139 AF32 57498.24  
## 127 80565-30911 55644.01  
## 133 AF111 52795.39  
## 89 622721001097 52352.63  
## 36 1K 46410.94  
## 138 AF31 45592.31  
## 39 1UL/C 44639.85  
## 160 E85 43405.33  
## 137 AF30 42152.24  
## 19 11HS 34353.75

Corey\_Oil\_Analysis\_TOP30Sales <- **sqldf**("select \* from Corey\_Oil where UM = 'G' and PNUM in ('2UL/RF','12HS','22LS','22HS','1UL/RF','2P/RF','DS15W40','5W30SB','80565-29811','2K','AF31','622515001097','2CAM2','5W30SYN','11HS','METHANOL','5W20SB','1P/RF','1UL/C','622723001097','661460008097','0W20SYN','HY32','THP','110B','80565-30911','HY46','1K','622721001097')")  
Corey\_Oil\_Analysis\_TOP30Sales**$**TotalSales <- **as.numeric**(**as.character**(Corey\_Oil\_Analysis\_TOP30Sales**$**TotalSales))  
**head**(Corey\_Oil\_Analysis\_TOP30Sales)

## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 IN 85273 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 2 IN 85276 105386 1/1/2018 0:00:00 1/1/2018 0:00:00 2UL/RF  
## 3 IN 85272 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 4 IN 85269 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 5 IN 85268 101 1/1/2018 0:00:00 1/31/2018 0:00:00 22LS  
## 6 IN 85267 105047 1/1/2018 0:00:00 2/15/2018 0:00:00 2UL/RF  
## PNAME1 PRICE COST TotalSales MEAS  
## 1 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## 2 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## 3 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## 4 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## 5 #2 Ultra Low Sulfur On Road Diesel - Cardlock 2.24 2.14 4.80 1  
## 6 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## UM UP QSHIP STNAME STCITY STZIP  
## 1 G BULK 4.36 Cardlock Credit Card Customer North Lake 53064  
## 2 G BULK 33.99 Groundskeepers Hartland 53029  
## 3 G BULK 8.10 Cardlock Credit Card Customer North Lake 53064  
## 4 G BULK 12.27 Cardlock Credit Card Customer North Lake 53064  
## 5 G BULK 19.85 Cardlock Credit Card Customer North Lake 53064  
## 6 G BULK 26.89 Evergreen Property Management Hartland 53029

# Data Grouping - TOP 30 by Quantity Shipped

Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis[,**c**(6,7,14)]  
**str**(Corey\_Oil\_Analysis\_Rank)

## 'data.frame': 11447 obs. of 3 variables:  
## $ PNUM : Factor w/ 904 levels "","(I)TIM-3628-50",..: 295 295 295 295 239 295 198 198 198 198 ...  
## $ PNAME1: chr "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" ...  
## $ QSHIP : num 4.36 33.99 8.1 12.27 19.85 ...

Corey\_Oil\_Analysis\_Rank**$**QSHIP <-**as.numeric**(**as.character**(Corey\_Oil\_Analysis\_Rank**$**QSHIP))  
Corey\_Oil\_Analysis\_Rank <- **aggregate**(Corey\_Oil\_Analysis\_Rank**$**QSHIP, **list**(PNUM=Corey\_Oil\_Analysis\_Rank**$**PNUM), sum)  
Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis\_Rank[**order**(**-**Corey\_Oil\_Analysis\_Rank**$**x),]  
**head.matrix**(Corey\_Oil\_Analysis\_Rank,n=30)

## PNUM x  
## 22 12HS 475824.45  
## 23 12LS 462295.30  
## 152 DEF 124161.60  
## 212 METHANOL 102844.08  
## 27 1379B 94016.60  
## 40 1UL/RF 80082.30  
## 74 5W30SB 41975.40  
## 58 2UL/RF 34655.72  
## 45 22LS 33648.65  
## 223 TT 27106.00  
## 39 1UL/C 20318.80  
## 15 110B 20308.20  
## 71 5W20SB 17410.80  
## 158 DS15W40 15721.50  
## 204 HY46 14166.00  
## 36 1K 11500.60  
## 19 11HS 11291.40  
## 139 AF32 10118.80  
## 84 622515001097 9523.90  
## 138 AF31 9464.50  
## 201 HY32 9187.40  
## 38 1P/RF 8755.10  
## 44 22HS 7512.77  
## 90 622723001097 6731.60  
## 127 80565-30911 6686.40  
## 174 FUSE110A 6539.47  
## 126 80565-29811 6208.10  
## 133 AF111 5617.30  
## 222 THP 5513.20  
## 64 41550 5467.00

Corey\_Oil\_Analysis\_TOP30QSHIP <- **sqldf**("select \* from Corey\_Oil where UM = 'G' and PNUM in ('12HS','12LS','DEF','METHANOL','1379B','1UL/RF','5W30SB','2UL/RF','22LS','TT','1UL/C','110B','5W20SB','DS15W40','HY46','1K','11HS','AF32','622515001097','AF31','HY32','1P/RF','22HS','622723001097','80565-30911','FUSE110A','80565-29811','AF111','THP','41550')")  
**head**(Corey\_Oil\_Analysis\_TOP30QSHIP)

## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 IN 85273 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 2 IN 85276 105386 1/1/2018 0:00:00 1/1/2018 0:00:00 2UL/RF  
## 3 IN 85272 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 4 IN 85269 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 5 IN 85268 101 1/1/2018 0:00:00 1/31/2018 0:00:00 22LS  
## 6 IN 85267 105047 1/1/2018 0:00:00 2/15/2018 0:00:00 2UL/RF  
## PNAME1 PRICE COST TotalSales MEAS  
## 1 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## 2 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## 3 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## 4 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## 5 #2 Ultra Low Sulfur On Road Diesel - Cardlock 2.24 2.14 4.80 1  
## 6 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 3.37 1  
## UM UP QSHIP STNAME STCITY STZIP  
## 1 G BULK 4.36 Cardlock Credit Card Customer North Lake 53064  
## 2 G BULK 33.99 Groundskeepers Hartland 53029  
## 3 G BULK 8.10 Cardlock Credit Card Customer North Lake 53064  
## 4 G BULK 12.27 Cardlock Credit Card Customer North Lake 53064  
## 5 G BULK 19.85 Cardlock Credit Card Customer North Lake 53064  
## 6 G BULK 26.89 Evergreen Property Management Hartland 53029

# Data Grouping - TOP 30 by Total Cost

Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis[,**c**(6,7,18)]  
**str**(Corey\_Oil\_Analysis\_Rank)

## 'data.frame': 11447 obs. of 3 variables:  
## $ PNUM : Factor w/ 904 levels "","(I)TIM-3628-50",..: 295 295 295 295 239 295 198 198 198 198 ...  
## $ PNAME1 : chr "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" ...  
## $ TotalCost: num 7.54 58.8 14.01 21.23 42.48 ...

Corey\_Oil\_Analysis\_Rank**$**TotalCost <- **as.numeric**(**as.character**(Corey\_Oil\_Analysis\_Rank**$**TotalCost))  
Corey\_Oil\_Analysis\_Rank <- **aggregate**(Corey\_Oil\_Analysis\_Rank**$**TotalCost, **list**(PNUM=Corey\_Oil\_Analysis\_Rank**$**PNUM), sum)  
Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis\_Rank[**order**(**-**Corey\_Oil\_Analysis\_Rank**$**x),]  
**head.matrix**(Corey\_Oil\_Analysis\_Rank,n=30)

## PNUM x  
## 22 12HS 1005671.27  
## 23 12LS 988013.81  
## 212 METHANOL 185805.45  
## 74 5W30SB 163439.33  
## 40 1UL/RF 159795.67  
## 15 110B 122195.86  
## 152 DEF 111145.40  
## 158 DS15W40 108057.03  
## 45 22LS 72853.95  
## 27 1379B 70508.68  
## 64 41550 69418.95  
## 71 5W20SB 68593.09  
## 58 2UL/RF 68357.03  
## 84 622515001097 61119.27  
## 204 HY46 61056.78  
## 126 80565-29811 57580.14  
## 201 HY32 50998.21  
## 90 622723001097 50243.77  
## 139 AF32 47841.86  
## 75 5W30SYN 38880.16  
## 39 1UL/C 38076.64  
## 36 1K 36165.89  
## 89 622721001097 35349.03  
## 127 80565-30911 33917.27  
## 222 THP 33885.51  
## 137 AF30 32141.85  
## 41 217D7D 31024.22  
## 133 AF111 27302.58  
## 19 11HS 27294.15  
## 215 RV100 27268.92

Corey\_Oil\_Analysis\_TOP30TotalCost <- **sqldf**("select \* from Corey\_Oil\_Analysis where UM = 'G' and PNUM in ('12HS','12LS','DEF','METHANOL','1379B','1UL/RF','5W30SB','2UL/RF','22LS','TT','1UL/C','110B','5W20SB','DS15W40','HY46','1K','11HS','AF32','622515001097','AF31','HY32','1P/RF','22HS','622723001097','80565-30911','FUSE110A','80565-29811','AF111','THP','41550')")  
**head**(Corey\_Oil\_Analysis\_TOP30TotalCost)

## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 IN 85273 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 2 IN 85276 105386 1/1/2018 0:00:00 1/1/2018 0:00:00 2UL/RF  
## 3 IN 85272 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 4 IN 85269 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 5 IN 85268 101 1/1/2018 0:00:00 1/31/2018 0:00:00 22LS  
## 6 IN 85267 105047 1/1/2018 0:00:00 2/15/2018 0:00:00 2UL/RF  
## PNAME1 PRICE COST TotalSales MEAS  
## 1 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 8.46 1  
## 2 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 65.94 1  
## 3 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 15.71 1  
## 4 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 23.80 1  
## 5 #2 Ultra Low Sulfur On Road Diesel - Cardlock 2.24 2.14 44.46 1  
## 6 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 52.17 1  
## UM UP QSHIP STNAME STCITY STZIP TotalCost  
## 1 G BULK 4.36 Cardlock Credit Card Customer North Lake 53064 7.54  
## 2 G BULK 33.99 Groundskeepers Hartland 53029 58.80  
## 3 G BULK 8.10 Cardlock Credit Card Customer North Lake 53064 14.01  
## 4 G BULK 12.27 Cardlock Credit Card Customer North Lake 53064 21.23  
## 5 G BULK 19.85 Cardlock Credit Card Customer North Lake 53064 42.48  
## 6 G BULK 26.89 Evergreen Property Management Hartland 53029 46.52  
## Profit  
## 1 0.92  
## 2 7.14  
## 3 1.70  
## 4 2.57  
## 5 1.98  
## 6 5.65

# Data Grouping - TOP 30 by Total Profit

Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis[,**c**(6,7,19)]  
**str**(Corey\_Oil\_Analysis\_Rank)

## 'data.frame': 11447 obs. of 3 variables:  
## $ PNUM : Factor w/ 904 levels "","(I)TIM-3628-50",..: 295 295 295 295 239 295 198 198 198 198 ...  
## $ PNAME1: chr "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" ...  
## $ Profit: num 0.92 7.14 1.7 2.57 1.98 ...

Corey\_Oil\_Analysis\_Rank**$**Profit <- **as.numeric**(**as.character**(Corey\_Oil\_Analysis\_Rank**$**Profit))  
Corey\_Oil\_Analysis\_Rank <- **aggregate**(Corey\_Oil\_Analysis\_Rank**$**Profit, **list**(PNUM=Corey\_Oil\_Analysis\_Rank**$**PNUM), sum)  
Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis\_Rank[**order**(**-**Corey\_Oil\_Analysis\_Rank**$**x),]  
**head.matrix**(Corey\_Oil\_Analysis\_Rank,n=30)

## PNUM x  
## 22 12HS 154184.54  
## 74 5W30SB 110703.33  
## 23 12LS 110392.47  
## 204 HY46 55488.47  
## 158 DS15W40 55417.02  
## 27 1379B 51159.32  
## 84 622515001097 48909.82  
## 71 5W20SB 48132.49  
## 152 DEF 48007.74  
## 201 HY32 42383.52  
## 90 622723001097 27986.76  
## 222 THP 26029.06  
## 133 AF111 25492.81  
## 75 5W30SYN 25037.24  
## 126 80565-29811 24682.56  
## 40 1UL/RF 22503.27  
## 127 80565-30911 21726.74  
## 138 AF31 18826.68  
## 212 METHANOL 17737.15  
## 15 110B 17330.16  
## 89 622721001097 17003.60  
## 160 E85 16522.36  
## 206 HY68 14638.73  
## 12 10W30SB 11483.92  
## 177 FUSE116 11413.07  
## 36 1K 10245.05  
## 137 AF30 10010.39  
## 139 AF32 9656.38  
## 45 22LS 9306.28  
## 9 0W20SYN 8650.09

Corey\_Oil\_Analysis\_TOP30Profit <- **sqldf**("select \* from Corey\_Oil\_Analysis where UM = 'G' and PNUM in ('12HS','12LS','DEF','METHANOL','1379B','1UL/RF','5W30SB','2UL/RF','22LS','TT','1UL/C','110B','5W20SB','DS15W40','HY46','1K','11HS','AF32','622515001097','AF31','HY32','1P/RF','22HS','622723001097','80565-30911','FUSE110A','80565-29811','AF111','THP','41550')")  
**head**(Corey\_Oil\_Analysis\_TOP30Profit)

## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 IN 85273 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 2 IN 85276 105386 1/1/2018 0:00:00 1/1/2018 0:00:00 2UL/RF  
## 3 IN 85272 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 4 IN 85269 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 5 IN 85268 101 1/1/2018 0:00:00 1/31/2018 0:00:00 22LS  
## 6 IN 85267 105047 1/1/2018 0:00:00 2/15/2018 0:00:00 2UL/RF  
## PNAME1 PRICE COST TotalSales MEAS  
## 1 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 8.46 1  
## 2 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 65.94 1  
## 3 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 15.71 1  
## 4 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 23.80 1  
## 5 #2 Ultra Low Sulfur On Road Diesel - Cardlock 2.24 2.14 44.46 1  
## 6 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 52.17 1  
## UM UP QSHIP STNAME STCITY STZIP TotalCost  
## 1 G BULK 4.36 Cardlock Credit Card Customer North Lake 53064 7.54  
## 2 G BULK 33.99 Groundskeepers Hartland 53029 58.80  
## 3 G BULK 8.10 Cardlock Credit Card Customer North Lake 53064 14.01  
## 4 G BULK 12.27 Cardlock Credit Card Customer North Lake 53064 21.23  
## 5 G BULK 19.85 Cardlock Credit Card Customer North Lake 53064 42.48  
## 6 G BULK 26.89 Evergreen Property Management Hartland 53029 46.52  
## Profit  
## 1 0.92  
## 2 7.14  
## 3 1.70  
## 4 2.57  
## 5 1.98  
## 6 5.65

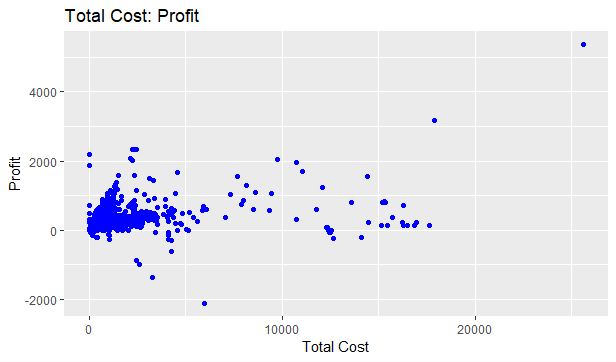
# Scatterplots of Profit vs. Sales and quantity shipped

Visualization of our Profit data using a Scatterplot to give us an idea of the company's revenue. A majority of our profit is from lower cost sales and profit. A bulk of our quantity shipped leads to the best Profit margin.

profit.sales <- ggplot(Corey\_Oil\_Analysis\_TOP30Profit, aes(x=Corey\_Oil\_Analysis\_TOP30Profit$TotalCost, y=Corey\_Oil\_Analysis\_TOP30Profit$Profit)) + geom\_point(color="blue")

profit.sales <- profit.sales + xlab("Total Cost") + ylab("Profit") + ggtitle("Total Cost: Profit")

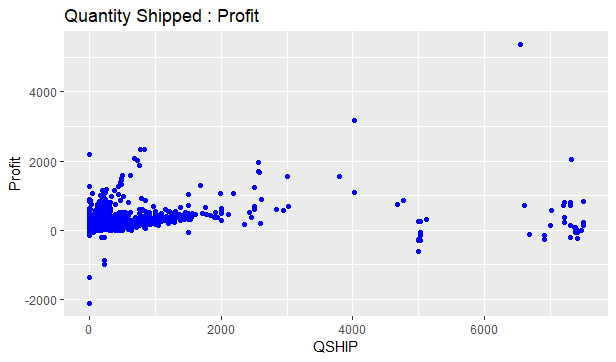
profit.sales



qship.profit <- ggplot(Corey\_Oil\_Analysis\_TOP30Profit, aes(x=Corey\_Oil\_Analysis\_TOP30Profit$QSHIP, y=Corey\_Oil\_Analysis\_TOP30Profit$Profit)) + geom\_point(color="blue")

qship.profit <- qship.profit + xlab("QSHIP") + ylab("Profit") + ggtitle("Quantity Shipped : Profit")

qship.profit



**head**(Corey\_Oil\_Analysis)

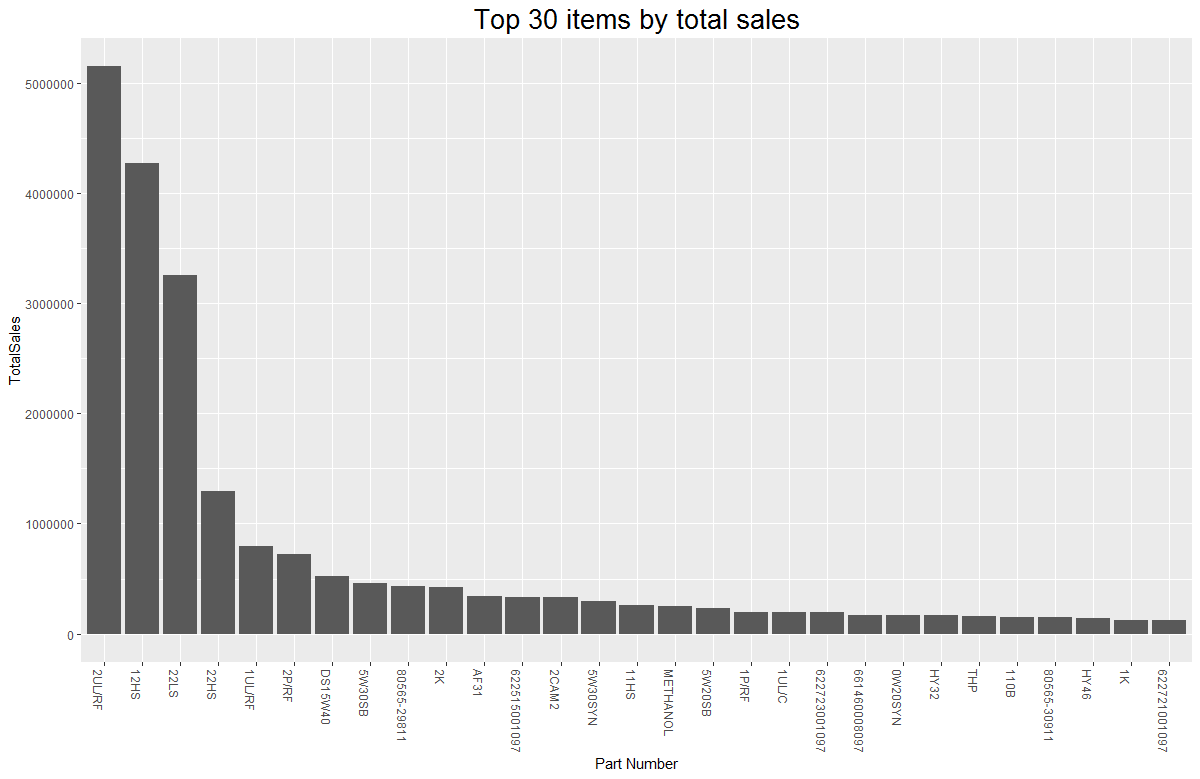
## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 IN 85273 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 2 IN 85276 105386 1/1/2018 0:00:00 1/1/2018 0:00:00 2UL/RF  
## 3 IN 85272 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 4 IN 85269 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 5 IN 85268 101 1/1/2018 0:00:00 1/31/2018 0:00:00 22LS  
## 6 IN 85267 105047 1/1/2018 0:00:00 2/15/2018 0:00:00 2UL/RF  
## PNAME1 PRICE COST TotalSales MEAS  
## 1 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 8.46 1  
## 2 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 65.94 1  
## 3 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 15.71 1  
## 4 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 23.80 1  
## 5 #2 Ultra Low Sulfur On Road Diesel - Cardlock 2.24 2.14 44.46 1  
## 6 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 52.17 1  
## UM UP QSHIP STNAME STCITY STZIP TotalCost  
## 1 G BULK 4.36 Cardlock Credit Card Customer North Lake 53064 7.54  
## 2 G BULK 33.99 Groundskeepers Hartland 53029 58.80  
## 3 G BULK 8.10 Cardlock Credit Card Customer North Lake 53064 14.01  
## 4 G BULK 12.27 Cardlock Credit Card Customer North Lake 53064 21.23  
## 5 G BULK 19.85 Cardlock Credit Card Customer North Lake 53064 42.48  
## 6 G BULK 26.89 Evergreen Property Management Hartland 53029 46.52  
## Profit  
## 1 0.92  
## 2 7.14  
## 3 1.70  
## 4 2.57  
## 5 1.98  
## 6 5.65

PNUM<-Corey\_Oil\_Analysis**$**PNUM  
TotalSales<-Corey\_Oil\_Analysis**$**TotalSales  
  
*#turns off scientific notation*  
**options**(scipen = 999)  
*#some graphs from the top30 data, aggregated by pname1*  
*#plotting the top30 by sales, ship, costs, and profit in pareto chart*

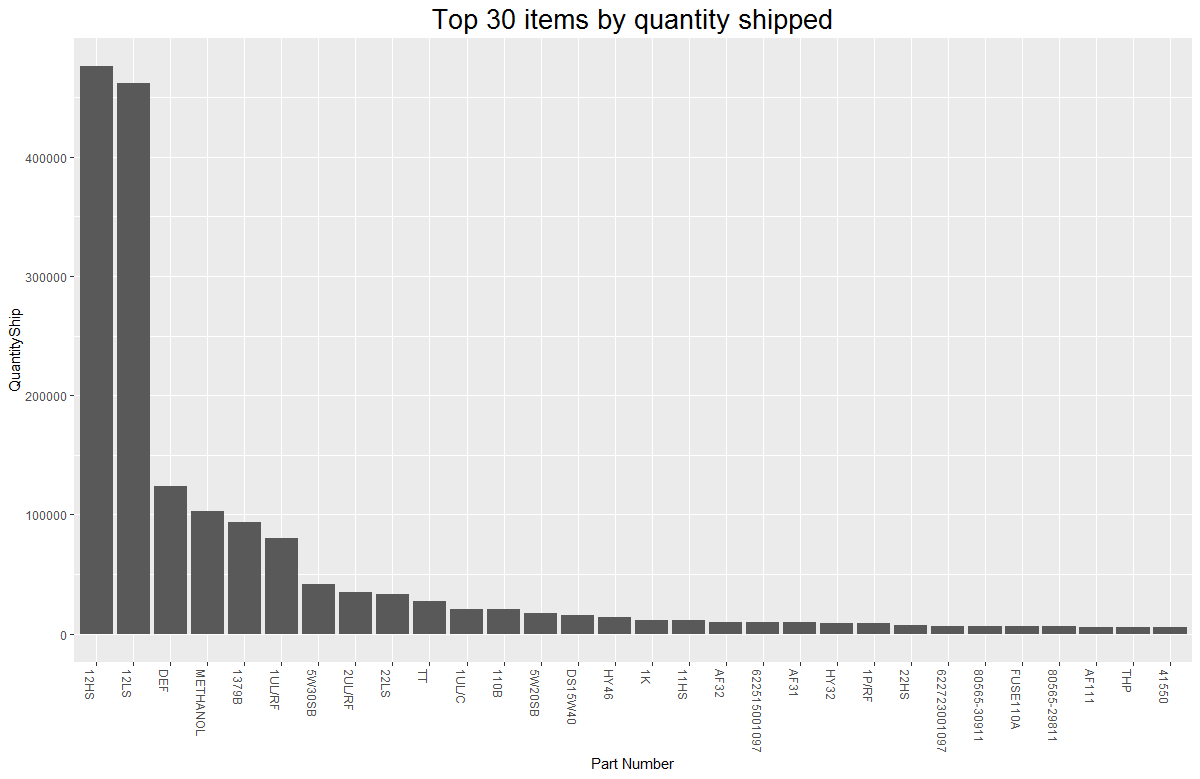
# Bar plots of Top 30 sales, costs, quantity shipped, and profit by part number

The following four charts show the top 30 items in descending order in the dataframe based on total sales, total cost, total numbers shipped, or most profitable. There are a few standout products such as 12HS and 12LS which were most profitable. The items with the greatest sales were not always the most profitable though since some of them are also the highest cost to the supplier. The coding for this was done in a similar manner for all the plots. The data was aggregated, sorted, summed and plotted using ggplot.

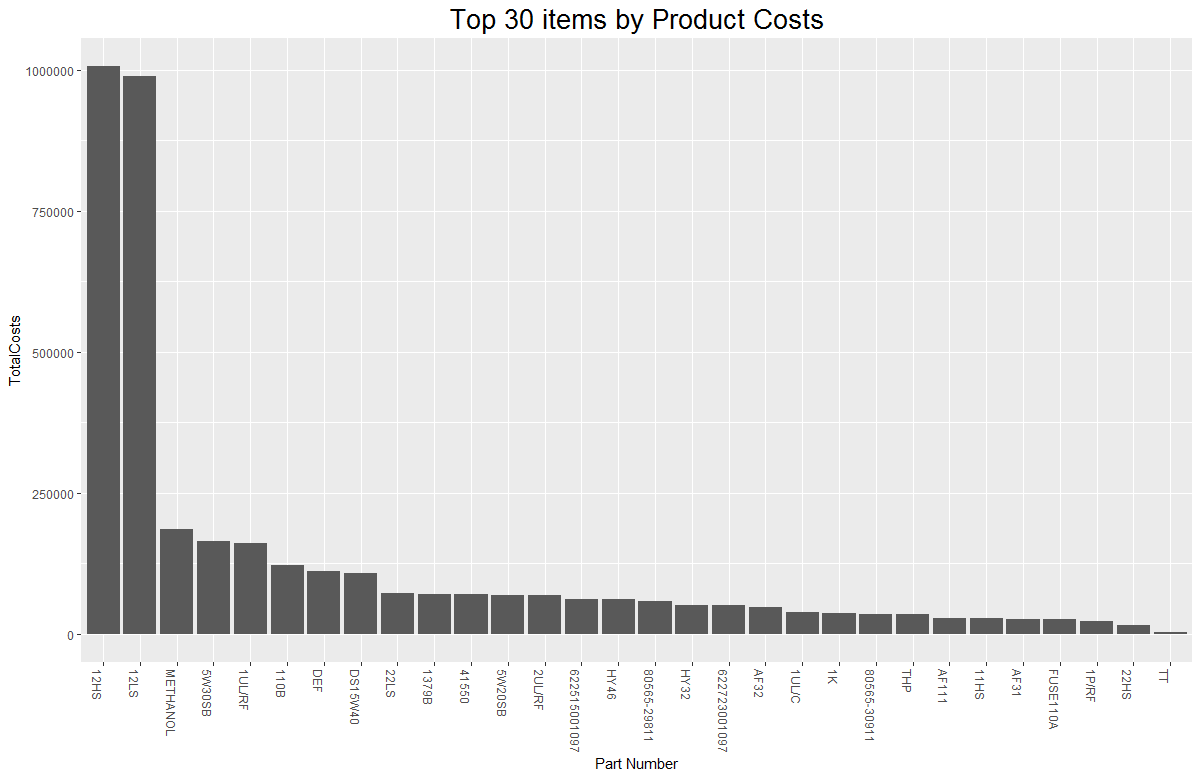
TOP30Sales <- **setNames**(**aggregate**(Corey\_Oil\_Analysis\_TOP30Sales**$**TotalSales, by=**list**(Corey\_Oil\_Analysis\_TOP30Sales**$**PNUM), FUN=sum), **c**("PartNumber","TotalSales"))  
TOP30Sales <- **ggplot**(data = TOP30Sales, **aes**(x= **reorder** (PartNumber, **-**TotalSales), y=TotalSales)) **+** **geom\_bar**(stat="identity")  
TOP30Sales <- TOP30Sales **+** **theme**(axis.text.x = **element\_text**(angle=**-**90, hjust=0)) **+** **ggtitle**("Top 30 items by total sales")  
TOP30Sales <- TOP30Sales **+** **scale\_x\_discrete**(name= "Part Number")  
TOP30Sales <- TOP30Sales **+** **theme**(plot.title = **element\_text**(size=20, hjust=0.5))  
TOP30Sales



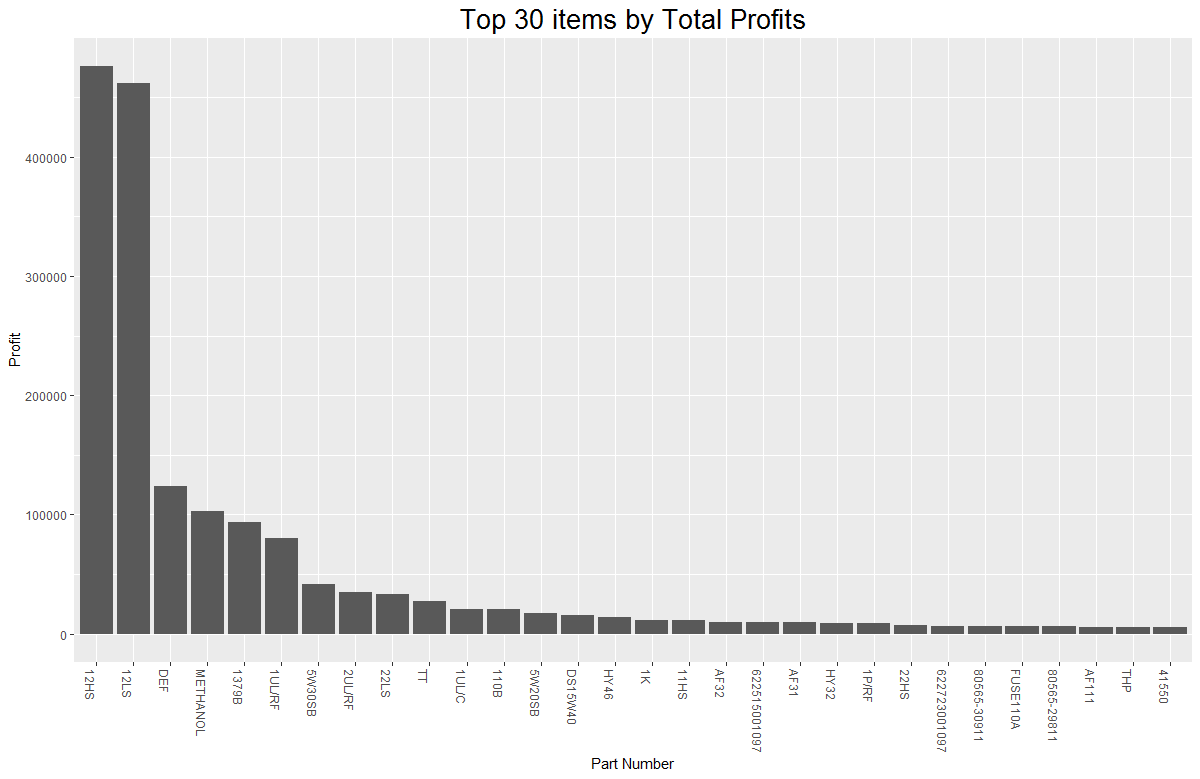
TOP30Ship<- **setNames**(**aggregate**(Corey\_Oil\_Analysis\_TOP30QSHIP**$**QSHIP, by=**list**(Corey\_Oil\_Analysis\_TOP30QSHIP**$**PNUM), FUN=sum), **c**("PartNumber","QuantityShip"))  
TOP30Ship <- **ggplot**( data = TOP30Ship, **aes**(x=**reorder** (PartNumber, **-**QuantityShip), y=QuantityShip)) **+** **geom\_bar**(stat="identity")  
TOP30Ship <- TOP30Ship **+** **theme**(axis.text.x = **element\_text**(angle=**-**90, hjust=0)) **+ggtitle**("Top 30 items by quantity shipped")  
TOP30Ship <- TOP30Ship **+** **scale\_x\_discrete**(name="Part Number")  
TOP30Ship <- TOP30Ship **+** **theme**(plot.title = **element\_text**(size=20, hjust=0.5))  
TOP30Ship



TOP30Cost <- **setNames**(**aggregate**(Corey\_Oil\_Analysis\_TOP30TotalCost**$**TotalCost, by=**list**(Corey\_Oil\_Analysis\_TOP30TotalCost**$**PNUM), FUN=sum), **c**("PartNumber","TotalCosts"))  
TOP30Cost <- **ggplot**( data = TOP30Cost, **aes**(x=**reorder** (PartNumber, **-**TotalCosts), y=TotalCosts)) **+** **geom\_bar**(stat="identity")  
TOP30Cost <- TOP30Cost **+** **theme**(axis.text.x = **element\_text**(angle=**-**90, hjust=0)) **+ggtitle**("Top 30 items by Product Costs")  
TOP30Cost <- TOP30Cost **+** **scale\_x\_discrete**(name="Part Number")  
TOP30Cost <- TOP30Cost **+** **theme**(plot.title = **element\_text**(size=20, hjust=0.5))  
TOP30Cost



TOP30Profit<- **setNames**(**aggregate**(Corey\_Oil\_Analysis\_TOP30QSHIP**$**QSHIP, by=**list**(Corey\_Oil\_Analysis\_TOP30Profit**$**PNUM), FUN=sum), **c**("PartNumber","Profit"))  
TOP30Profit <- **ggplot**( data = TOP30Profit, **aes**(x=**reorder** (PartNumber, **-**Profit), y=Profit)) **+** **geom\_bar**(stat="identity")  
TOP30Profit <- TOP30Profit **+** **theme**(axis.text.x = **element\_text**(angle=**-**90, hjust=0)) **+ggtitle**("Top 30 items by Total Profits")  
TOP30Profit <- TOP30Profit **+** **scale\_x\_discrete**(name="Part Number")  
TOP30Profit <- TOP30Profit **+** **theme**(plot.title = **element\_text**(size=20, hjust=0.5))  
TOP30Profit



# Bar plot showing Sales, Costs and Profit

The following bar plot is an interesting graph. The total sales bar shown represents the total sales that the company had. The total cost bar represents the amount that the product cost the company. As previously stated, when they are subtracted from each other, the leftover amount is the profit for the company. The second and third bar should equal the first bar. To compute those, the sum command was used for all three of those columns.

*#this is a simple barplot with sale info*  
allsales <- **sum**(Corey\_Oil\_Analysis**$**TotalSales)  
allcosts <- **sum**(Corey\_Oil\_Analysis**$**TotalCost)  
allprofits <- **sum**(Corey\_Oil\_Analysis**$**Profit)  
allchart <- **c**(allsales,allcosts,allprofits)  
**barplot**(allchart, main="Totals for Sales, Costs, and Profits", names.arg=**c**("Total Sales", "Total Costs", "Total Profit"))



## Linear Regression

A linear regression analysis was performed. If the line of best fit can be computed, the expected sales and profit can be calculated from the cost of the product. By calling the lm() function, we were able to get the summary and see that the formula for the regression is “Sales =1.096 (Cost) + 77.” The R-squared value of this regression is 0.9715, which is a very good line fit. The P value is extremely low, (P< 2.2^10-16), meaning these analyses are statistically significant. A plot was made from these data using three variables, cost (x-axis), sales (y-axis), and profit (darkness of blue). The plot also includes the formula for the line and the R^2 value. Along with this, the code has a function that requests the cost of an item and returns the expected profit from that sale using the regression analysis data.

# linear regression using cost to determine sales and possibly profit

salescostprofitreg <- **lm**(TotalSales **~** TotalCost, data=Corey\_Oil\_Analysis)  
**summary** (salescostprofitreg)

##   
## Call:  
## lm(formula = TotalSales ~ TotalCost, data = Corey\_Oil\_Analysis)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2766.4 -75.3 -49.9 24.9 4440.2   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 77.019016 1.945636 39.59 <0.0000000000000002 \*\*\*  
## TotalCost 1.096403 0.001756 624.22 <0.0000000000000002 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 194.3 on 11445 degrees of freedom  
## Multiple R-squared: 0.9715, Adjusted R-squared: 0.9715   
## F-statistic: 3.897e+05 on 1 and 11445 DF, p-value: < 0.00000000000000022

# This plot has linear modeling added and a function to test the profit based on the cost

salescostprofit <- **ggplot**(Corey\_Oil\_Analysis, **aes**(y=TotalSales,x=TotalCost,color=Profit))  
salescostprofit <- salescostprofit **+** **geom\_point**(size=1)  
salescostprofit <- salescostprofit **+** **ggtitle**("Sales, Cost, and Profit")  
salescostprofit <- salescostprofit **+** **theme**(plot.title = **element\_text**(size=20, hjust=0.5))  
salescostprofit <- salescostprofit **+** **xlim**(0, 31000) **+** **ylim**(0, 31000)  
salescostprofit <- salescostprofit **+** **geom\_abline**(intercept = 77, slope =1.09 )  
salescostprofit <- salescostprofit **+** **geom\_text**(x = 15000, y = 30000, label = "Y=1.09X + 77, R^2 = 0.9715")  
salescostprofit

# this function allows you to input a cost, and get the expected profit

estimatedprofit<- **function**(testcost)  
{  
 expectedsales <- (testcost **\*** 1.09) **+** 77  
 expectedprofit <- (expectedsales**-**testcost)  
 **return** (expectedprofit)  
}

# 2327 is the expected profit for cost @ $25000

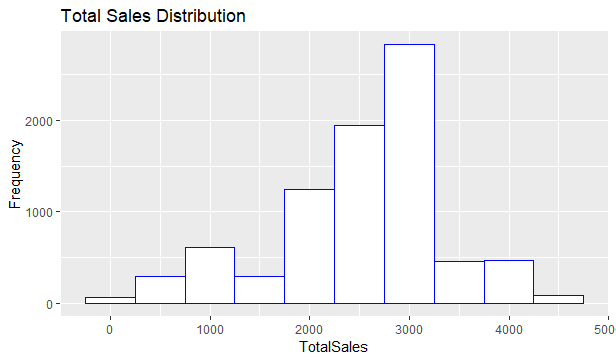
**estimatedprofit** (25000)

## [1] 2327

## Histogram of total sales vs count

Descriptive representation using a histogram to show us our distribution of Total Sales. Our frequency of Total Sales lies heavily in the middle producing a normal distribution with our largest sales in between 2000 and 3000.

tot.sales <- **ggplot**(Corey\_Oil\_Analysis\_TOP30Sales, **aes**(x=TotalSales)) **+** **geom\_histogram**(binwidth=25, color="blue", fill="white")  
tot.sales



## Map Plots

The data of sales by specific zip codes was aggregated and merged. The data were plotted in a map of the USA by zip code using ggplot2. The size of each dot represents the sales for that zip code. The most obvious part of this map is that most of the data come directly from Wisconsin. Therefore the same data were compiled only for Wisconsin and a new map was made. The map is only showing the Wisconsin data. Along with this plot, a similar plot was made of using the total profit within the zip codes. There are slight differences between both of these plots.

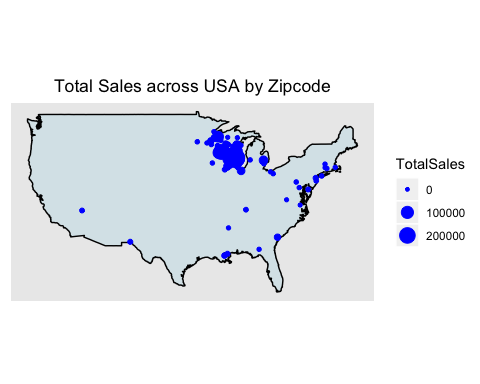
removeaxis <- **theme**(  
 axis.text = **element\_blank**(),  
 axis.line = **element\_blank**(),  
 axis.ticks = **element\_blank**(),  
 panel.border = **element\_blank**(),  
 panel.grid = **element\_blank**(),  
 axis.title = **element\_blank**())  
  
**data**(zipcode)  
Corey\_Oil\_Analysis\_LatLon <- **merge**(Corey\_Oil\_Analysis, zipcode, by.x="STZIP", by.y = "zip")  
**head**(Corey\_Oil\_Analysis\_LatLon)

## STZIP TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 01062 IN 97941 150615 5/23/2018 0:00:00 5/23/2018 0:00:00 E85  
## 2 02346 IN 102189 150596 6/29/2018 0:00:00 6/29/2018 0:00:00 FUSE116  
## 3 06095 IN 91243 150609 3/7/2018 0:00:00 3/7/2018 0:00:00 FUSE116  
## 4 06232 IN 106658 150597 8/15/2018 0:00:00 8/15/2018 0:00:00 E98  
## 5 08701 IN 89889 150117 3/26/2018 0:00:00 3/26/2018 0:00:00 FUSE100  
## 6 08859 IN 90160 150529 2/28/2018 0:00:00 3/10/2018 0:00:00 E98  
## PNAME1 PRICE COST TotalSales MEAS UM UP QSHIP  
## 1 Fuse E85R Racing Ethanol 2.73 1.96 589.68 54 G DRUM 4  
## 2 Fuse Ignitor 16 12.71 6.46 381.30 5 G PAIL 6  
## 3 Fuse Ignitor 16 7.11 4.65 383.94 54 G DRUM 1  
## 4 Fuse E98R Racing Ethanol 2.75 1.96 891.00 54 G DRUM 6  
## 5 Fuse Ignitor 1.0 5.73 4.43 1237.68 54 G DRUM 4  
## 6 Fuse E98R Racing Ethanol 2.81 2.33 606.96 54 G DRUM 4  
## STNAME STCITY TotalCost Profit city state  
## 1 Sias, Travis Florence 423.36 166.32 Florence MA  
## 2 Robert Rose Middleboro 193.80 187.50 Middleboro MA  
## 3 John Blase Windsor 251.10 132.84 Windsor CT  
## 4 Jon Lachapelle Andover 635.04 255.96 Andover CT  
## 5 Jesel Inc Lakewood 956.88 280.80 Lakewood NJ  
## 6 Double O Contracting Parlin 503.28 103.68 Parlin NJ  
## latitude longitude  
## 1 42.32423 -72.67915  
## 2 41.88958 -70.89406  
## 3 41.85730 -72.66654  
## 4 41.73701 -72.37151  
## 5 40.08278 -74.20940  
## 6 40.46185 -74.30343

*#aggregate to get totals by zip*  
oil\_agg\_totalsales\_by\_zip<- **setNames**(**aggregate**(Corey\_Oil\_Analysis\_LatLon**$**TotalSales, by=**list**(Corey\_Oil\_Analysis\_LatLon**$**STZIP), FUN=sum), **c**("Zipcode","TotalSales"))  
oil\_agg\_totalsales\_by\_zip<- **merge**(x=oil\_agg\_totalsales\_by\_zip, y=zipcode, by.x = "Zipcode", by.y = "zip")  
**head**(oil\_agg\_totalsales\_by\_zip)

## Zipcode TotalSales city state latitude longitude  
## 1 01062 589.68 Florence MA 42.32423 -72.67915  
## 2 02346 381.30 Middleboro MA 41.88958 -70.89406  
## 3 06095 383.94 Windsor CT 41.85730 -72.66654  
## 4 06232 891.00 Andover CT 41.73701 -72.37151  
## 5 08701 1237.68 Lakewood NJ 40.08278 -74.20940  
## 6 08859 2907.36 Parlin NJ 40.46185 -74.30343

*#this gives us the total cost of all purchases nationwide*  
mapUSA <- **map\_data**("usa")  
mapUSA <- **ggplot**(data = mapUSA) **+** **geom\_polygon**(**aes**(x = long, y = lat, group = group), fill = "light blue", color = "black", alpha = 0.3) **+** **coord\_fixed**(1.3)  
mapUSA <- mapUSA **+** **geom\_point**(data= oil\_agg\_totalsales\_by\_zip, **aes**(size=TotalSales, x = longitude, y = latitude), color="blue")  
mapUSA <- mapUSA **+** **scale\_size\_continuous**(range = **c**(range = **c**(1, 6)))  
mapUSA <- mapUSA **+** removeaxis **+** **ggtitle**("Total Sales across USA by Zipcode")  
mapUSA <- mapUSA **+** **theme**(plot.title = **element\_text**(hjust = 0.5))  
mapUSA



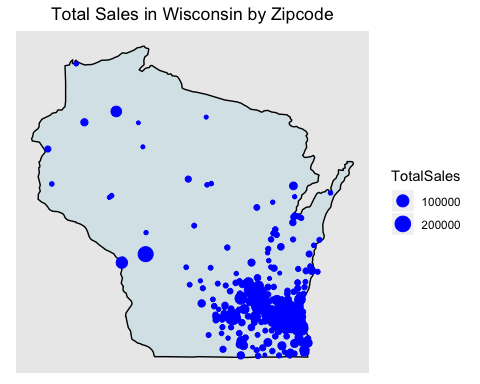
*#This will get rid of the non WI data*  
Corey\_Oil\_Analysis\_LatLonWI <- Corey\_Oil\_Analysis\_LatLon[Corey\_Oil\_Analysis\_LatLon**$**state **==** "WI", ]  
*#this will rerun the aggregates*  
oil\_agg\_totalsales\_by\_zipWI<- **setNames**(**aggregate**(Corey\_Oil\_Analysis\_LatLonWI**$**TotalSales, by=**list**(Corey\_Oil\_Analysis\_LatLonWI**$**STZIP), FUN=sum), **c**("Zipcode","TotalSales"))  
oil\_agg\_totalsales\_by\_zipWI<- **merge**(x=oil\_agg\_totalsales\_by\_zipWI, y=zipcode, by.x = "Zipcode", by.y = "zip")  
**head**(oil\_agg\_totalsales\_by\_zipWI)

## Zipcode TotalSales city state latitude longitude  
## 1 53004 3243.26 Belgium WI 43.49894 -87.86211  
## 2 53005 41753.26 Brookfield WI 43.06087 -88.09478  
## 3 53006 127.63 Brownsville WI 43.62400 -88.51089  
## 4 53007 34197.31 Butler WI 43.10836 -88.06893  
## 5 53012 1847.79 Cedarburg WI 43.30541 -87.99794  
## 6 53013 1451.45 Cedar Grove WI 43.57527 -87.84597

*#since most of the sales are in Wisconsin, this is wisconsin only*  
states <- **map\_data**("state")  
**dim**(states)

## [1] 15537 6

wisconsin <- **subset**(states, region **%in%** **c**("wisconsin"))  
mapWI <- **ggplot**(data = wisconsin) **+** **geom\_polygon**(**aes**(x = long, y = lat, group = group), fill = "light blue", color = "black", alpha = 0.3) **+** **coord\_fixed**(1.3)  
mapWI <- mapWI **+** **geom\_point**(data= oil\_agg\_totalsales\_by\_zipWI, **aes**(size=TotalSales, x = longitude, y = latitude), color="blue")  
mapWI <- mapWI **+** **scale\_size\_continuous**(range = **c**(range = **c**(1, 6)))  
mapWI <- mapWI **+** removeaxis **+** **ggtitle**("Total Sales in Wisconsin by Zipcode")  
mapWI <- mapWI **+** **theme**(plot.title = **element\_text**(hjust = 0.5))  
mapWI



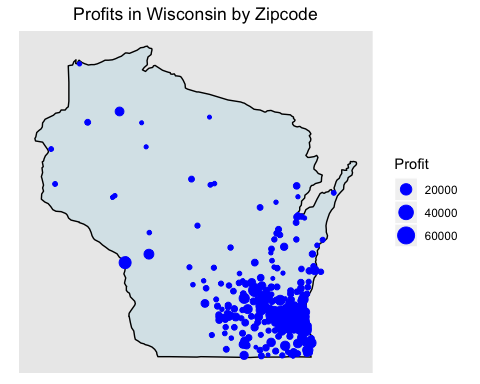
*#These are the total profits for Wisconsin*  
oil\_agg\_profits\_by\_zipWI<- **setNames**(**aggregate**(Corey\_Oil\_Analysis\_LatLonWI**$**Profit, by=**list**(Corey\_Oil\_Analysis\_LatLonWI**$**STZIP), FUN=sum), **c**("Zipcode","Profit"))  
oil\_agg\_profits\_by\_zipWI<- **merge**(x=oil\_agg\_profits\_by\_zipWI, y=zipcode, by.x = "Zipcode", by.y = "zip")  
**head**(oil\_agg\_profits\_by\_zipWI)

## Zipcode Profit city state latitude longitude  
## 1 53004 1247.91 Belgium WI 43.49894 -87.86211  
## 2 53005 13999.86 Brookfield WI 43.06087 -88.09478  
## 3 53006 22.98 Brownsville WI 43.62400 -88.51089  
## 4 53007 13087.87 Butler WI 43.10836 -88.06893  
## 5 53012 790.63 Cedarburg WI 43.30541 -87.99794  
## 6 53013 652.30 Cedar Grove WI 43.57527 -87.84597

states <- **map\_data**("state")  
**dim**(states)

## [1] 15537 6

wisconsin <- **subset**(states, region **%in%** **c**("wisconsin"))  
mapWIProfit <- **ggplot**(data = wisconsin) **+** **geom\_polygon**(**aes**(x = long, y = lat, group = group), fill = "light blue", color = "black", alpha = 0.3) **+** **coord\_fixed**(1.3)  
mapWIProfit <- mapWIProfit **+** **geom\_point**(data= oil\_agg\_profits\_by\_zipWI, **aes**(size=Profit, x = longitude, y = latitude), color="blue")  
mapWIProfit <- mapWIProfit **+** **scale\_size\_continuous**(range = **c**(range = **c**(1, 6)))  
mapWIProfit <- mapWIProfit **+** removeaxis **+** **ggtitle**("Profits in Wisconsin by Zipcode")  
mapWIProfit <- mapWIProfit **+** **theme**(plot.title = **element\_text**(hjust = 0.5))  
mapWIProfit



## Neural Network Analysis

The objective of this Neural Network is to predict the total Sales of a product (Y) We use Price, cost, Qship as Input Variables. We scaled the data using Max\_Min Normalization and passed it to the training and test set. A training set is used to find the relationship between the dependent and independent variables while the test set assesses the performance of the model. We use 60% of the dataset as a training set. The assignment of the data to training and test set is done using random sampling. We perform random sampling on R using sample ( )function. We have used set.seed( ) to generate the same random sample each time and maintain consistency

library(neuralnet,pos=17)

Corey\_Oil\_neuralnet<-sqldf(“select TotalSales,price,cost,QSHIP from Corey\_Oil\_Analysis”)

colSums(is.na(Corey\_Oil\_neuralnet))

summary(Corey\_Oil\_neuralnet)

dim(Corey\_Oil\_neuralnet)

# Scaling of Data

# MAX-MIN NORMALIZATION

normalize <- function(x) {

return ((x - min(x)) / (max(x) - min(x)))

}

maxmindf <- as.data.frame(lapply(Corey\_Oil\_neuralnet, normalize))

dim(maxmindf)

## Before we actually call the neuralnetwork() function we need to create a formula to insert into the machine learning model.

## The neuralnetwork() function won’t accept the typical decimal R format for a formula involving all features (e.g. y ~.).

## However, we can use a simple script to create the expanded formula and save us some typing:

feats <- names(maxmindf[,2:4])

f <- paste(feats,collapse=‘+’)

f <- paste(‘TotalSales ~’,f)

f

f <- as.formula(f)

f

# Create Training and Testing Data Set using Random Sampling

samplesize = 0.60 \* nrow(maxmindf)

samplesize

set.seed(80)

index=sample(seq\_len ( nrow(maxmindf)),size= samplesize)

trainset=maxmindf[index,]

dim(trainset)

testset=maxmindf[ -index,]

dim(testset)

head(testset)

# Run the NeuralNetwork using hidden=c(1,2)

Corey\_Neuralt<-neuralnet(f,trainset,hidden=2,lifesign = “minimal”,linear.output = TRUE,threshold = 0.01)

plot(Corey\_Neuralt,rep=“best”)

Corey\_Neuralt$result.matrix

# Model Validation

# 

results<-data.frame(actual=testsetnet.result)

head(results)

percentCorrect2 <- (results[1,1]+results[2,2])/(results[1,1]+results[1,2]+results[2,1]+results[2,2])\*100

percentCorrect2

## Model Validation and plotting neuralNet Prediction vs Actual Total Sales

pred\_neuralnet.scaled <- Corey\_Neuralt1.resultsTotalSales)-min(maxmindfTotalSales)

real.values <- (testsetTotalSales)-min(maxmindfTotalSales)

MSE.neuralnetModel <- sum((real.values - pred\_neuralnet.scaled)^2)/nrow(testset)

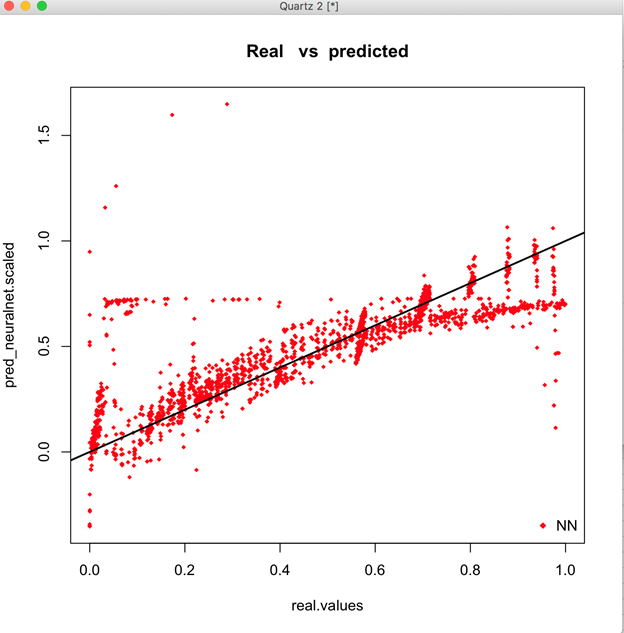
MSE.neuralnetModel

plot(real.values, pred\_neuralnet.scaled, col=‘red’,main=‘Real Sales vs predicted Sales’,pch=18,cex=0.7)

abline(0,1,lwd=2)

legend(‘bottomright’,legend=‘NN’,pch=18,col=‘red’, bty=‘n’)

## 



## Machine Learning

We applied association rules mining to our dataset. We focus on the association among quantity shipped, prices, and costs. We applied three different models to execute the prediction and testing methods. Each model returns different prediction results. We evaluate results and utilize the best model base on fitting.

*# Step 1: Load the data*  
**str**(Corey\_Oil\_Analysis)

## 'data.frame': 11447 obs. of 19 variables:  
## $ TCode : Factor w/ 6 levels ""," - Unlined",..: 6 6 6 6 6 6 6 6 6 6 ...  
## $ Refn : chr "85273" "85276" "85272" "85269" ...  
## $ CNUM : chr "101" "105386" "101" "101" ...  
## $ SHIPDATE : Factor w/ 163 levels "1/1/2018 0:00:00",..: 1 1 1 1 1 1 12 12 12 12 ...  
## $ DUEDATE : Factor w/ 272 levels "","1.00","1/1/2018 0:00:00",..: 27 3 27 27 27 49 6 6 14 6 ...  
## $ PNUM : Factor w/ 904 levels "","(I)TIM-3628-50",..: 295 295 295 295 239 295 198 198 198 198 ...  
## $ PNAME1 : chr "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" ...  
## $ PRICE : num 1.94 1.94 1.94 1.94 2.24 1.94 2.39 2.39 2.39 2.39 ...  
## $ COST : num 1.73 1.73 1.73 1.73 2.14 1.73 1.97 1.97 1.97 1.97 ...  
## $ TotalSales: num 8.46 65.94 15.71 23.8 44.46 ...  
## $ MEAS : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ UM : Factor w/ 8 levels "","10o","e","E",..: 5 5 5 5 5 5 5 5 5 5 ...  
## $ UP : Factor w/ 26 levels "","1","30DRUM",..: 10 10 10 10 10 10 10 10 10 10 ...  
## $ QSHIP : num 4.36 33.99 8.1 12.27 19.85 ...  
## $ STNAME : Factor w/ 2169 levels "","1:28 Automotive",..: 292 737 292 292 292 557 1741 2091 338 994 ...  
## $ STCITY : chr "North Lake" "Hartland" "North Lake" "North Lake" ...  
## $ STZIP : Factor w/ 400 levels "",".",".53219",..: 89 58 89 89 89 58 92 63 63 58 ...  
## $ TotalCost : num 7.54 58.8 14.01 21.23 42.48 ...  
## $ Profit : num 0.92 7.14 1.7 2.57 1.98 ...

Corey\_Oil\_Analysis\_m <- **data.frame**(Corey\_Oil\_Analysis[1**:**1000,**c**(8,9,14)])   
**str**(Corey\_Oil\_Analysis\_m)

## 'data.frame': 1000 obs. of 3 variables:  
## $ PRICE: num 1.94 1.94 1.94 1.94 2.24 1.94 2.39 2.39 2.39 2.39 ...  
## $ COST : num 1.73 1.73 1.73 1.73 2.14 1.73 1.97 1.97 1.97 1.97 ...  
## $ QSHIP: num 4.36 33.99 8.1 12.27 19.85 ...

**colnames**(Corey\_Oil\_Analysis\_m)[**colSums**(**is.na**(Corey\_Oil\_Analysis\_m)) **>** 0] *# find which columns in the dataframe contain NAs.*

## character(0)

Corey\_Oil\_Analysis\_m**$**QSHIP[**is.na**(Corey\_Oil\_Analysis\_m**$**QSHIP)] <- **mean**(Corey\_Oil\_Analysis\_m**$**QSHIP, na.rm=TRUE) *# find the NAs in column "QSHIP" and replace them by the mean value of this column*  
Corey\_Oil\_Analysis\_m**$**PRICE[**is.na**(Corey\_Oil\_Analysis\_m**$**PRICE)] <- **mean**(Corey\_Oil\_Analysis\_m**$**PRICE, na.rm=TRUE)*# find the NAs in column "PRICE" and replace those NAs by the mean value of this column*  
*# --------------------------------------------------------------------*  
*# Step 2: Create train and test data sets*  
*# create a list of random index for Corey\_Oil\_Analysis\_m data and store the index in a variable called "ranIndex"*  
*#*   
**dim**(Corey\_Oil\_Analysis\_m)

## [1] 1000 3

Corey\_Oil\_Analysis\_m[1**:**5,]

## PRICE COST QSHIP  
## 1 1.94 1.73 4.36  
## 2 1.94 1.73 33.99  
## 3 1.94 1.73 8.10  
## 4 1.94 1.73 12.27  
## 5 2.24 2.14 19.85

randIndex <- **sample**(1**:dim**(Corey\_Oil\_Analysis\_m)[1])  
**head**(randIndex)

## [1] 707 625 783 896 324 465

**length**(randIndex)

## [1] 1000

Corey\_Oil\_Analysis\_m[148,]

## PRICE COST QSHIP  
## 148 2.43 1.98 233.9

Corey\_Oil\_Analysis\_m[45,]

## PRICE COST QSHIP  
## 45 2.39 1.96 166.6

*#*   
*# # In order to split data, create a 2/3 cutpoint and round the number*  
cutpoint2\_3 <- **floor**(2**\*dim**(Corey\_Oil\_Analysis\_m)[1]**/**3)  
*# check the 2/3 cutpoint*  
cutpoint2\_3

## [1] 666

*#*   
*# create train data set, which contains the first 2/3 of overall data*  
*#*   
trainData <- Corey\_Oil\_Analysis\_m[randIndex[1**:**cutpoint2\_3],]  
**dim**(trainData)

## [1] 666 3

**head**(trainData)

## PRICE COST QSHIP  
## 707 5.38 3.47 1.00  
## 625 2.24 2.08 32.27  
## 783 1.94 1.73 12.75  
## 896 2.24 2.08 2.92  
## 324 2.39 2.02 300.00  
## 465 10.86 6.35 90.00

*#*   
*# create test data, which contains the left 1/3 of the overall data*  
*#*   
testData <- Corey\_Oil\_Analysis\_m[randIndex[(cutpoint2\_3**+**1)**:dim**(Corey\_Oil\_Analysis\_m)[1]],]  
**dim**(testData) *# check test data set*

## [1] 334 3

**head**(trainData)

## PRICE COST QSHIP  
## 707 5.38 3.47 1.00  
## 625 2.24 2.08 32.27  
## 783 1.94 1.73 12.75  
## 896 2.24 2.08 2.92  
## 324 2.39 2.02 300.00  
## 465 10.86 6.35 90.00

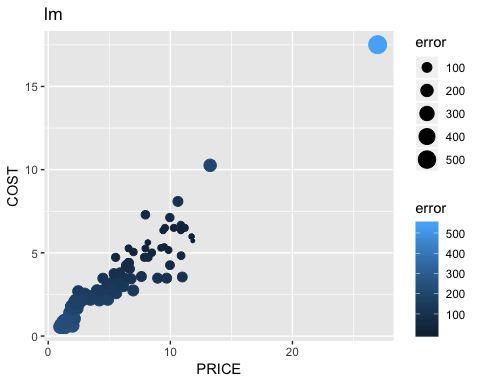
*#------------------------------------------------------lm model*  
model <- **lm**(QSHIP **~**.,data=trainData)  
lmPred <- **predict**(model,testData)  
  
**str**(lmPred)

## Named num [1:334] 194 184 193 194 -21 ...  
## - attr(\*, "names")= chr [1:334] "780" "662" "435" "84" ...

compTable3 <- **data.frame**(testData[,1],lmPred)  
**colnames**(compTable3) <- **c**("test","Pred")  
**sqrt**(**mean**((compTable3**$**test**-**compTable3**$**Pred)**^**2))

## [1] 174.5609

*#lm plot*  
compTable3**$**error <- **abs**(compTable3**$**test **-** compTable3**$**Pred)  
Plot3 <- **data.frame**(compTable3**$**error, testData**$**PRICE, testData**$**COST)  
**colnames**(Plot3) <- **c**("error","PRICE","COST")  
lm.plot <- **ggplot**(Plot3, **aes**(x=PRICE,y=COST)) **+** **geom\_point**(**aes**(size=error, color=error)) **+** **ggtitle**("lm")  
lm.plot



*# --------------------------------------------------------------------KSVM*  
*# Step 3: Build a Model using KSVM & visualize the results*  
*# 1) Build a model to predict QSHIP and name it "svmOutput"*  
*# This is the Training step*  
svmOutput <- **ksvm**(QSHIP**~**., *# set "QSHIP" as the target predicting variable; "." means use all other variables to predict "QSHIP"*  
 data = trainData, *# specify the data to use in the analysis*  
 kernel = "rbfdot", *# kernel function that projects the low dimensional problem into higher dimensional space*  
 kpar = "automatic",*# kpar refer to parameters that can be used to control the radial function kernel(rbfdot)*  
 C = 10, *# C refers to "COST of Constrains"*  
 cross = 10, *# use 10 fold cross validation in this model*  
 prob.model = TRUE *# use probability model in this model*  
)  
*# check the model*  
svmOutput

## Support Vector Machine object of class "ksvm"   
##   
## SV type: eps-svr (regression)   
## parameter : epsilon = 0.1 cost C = 10   
##   
## Gaussian Radial Basis kernel function.   
## Hyperparameter : sigma = 330.164408218941   
##   
## Number of Support Vectors : 280   
##   
## Objective Function Value : -769.8646   
## Training error : 0.432814   
## Cross validation error : 176868.5   
## Laplace distr. width : 120.1747

*# 2) Test the model with the testData data set*  
svmPred <- **predict**(svmOutput, *# use the built model "svmOutput" to predict*   
 testData, *# use testData to generate predictions*  
 type = "votes" *# request "votes" from the prediction process*  
)  
  
  
*# create a comparison data frame that contains the exact "QSHIP" value and the predicted "QSHIP" value*  
*# use for RMSE calc*   
  
compTable <- **data.frame**(testData[,1], svmPred[,1])  
*# change the column names to "test" and "Pred"*  
**colnames**(compTable) <- **c**("test","Pred")  
  
*# compute the Root Mean Squared Error*  
**sqrt**(**mean**((compTable**$**test**-**compTable**$**Pred)**^**2)) *#A smaller value indicates better model performance.*

## [1] 138.9826

*# 3) Plot the results*  
*# compute absolute error for each case*  
compTable**$**error <- **abs**(compTable**$**test **-** compTable**$**Pred)  
*# create a new dataframe contains error, tempreture and COST*  
svmPlot <- **data.frame**(compTable**$**error, testData**$**PRICE, testData**$**COST, testData**$**QSHIP)  
*# assign column names*  
**colnames**(svmPlot) <- **c**("error","PRICE","COST", "QSHIP")  
*# polt result using ggplot, setting "PRICE" as x-axis and "COST" as y-axis*  
plot.ksvm <- **ggplot**(svmPlot, **aes**(x=PRICE,y=COST)) **+**   
 *# use point size and color shade to illustrate how big is the error*  
 **geom\_point**(**aes**(size=error, color=error))**+**  
 **ggtitle**("ksvm")  
  
  
*# Step 4: Create a "goodQSHIP" variable*  
*# calculate average QSHIP*  
meanTotalSales <- **mean**(Corey\_Oil\_Analysis\_m**$**QSHIP,na.rm=TRUE)  
*# create a new variable named "goodQSHIP" in train data set*  
*# goodQSHIP = 0 if QSHIP is below average QSHIP*  
*# goodQSHIP = 1 if QSHIP is eaqual or above the average QSHIP*  
trainData**$**goodQSHIP <- **ifelse**(trainData**$**QSHIP**<**meanTotalSales, 0, 1)  
*# do the same thing for test dataset*  
testData**$**goodQSHIP <- **ifelse**(testData**$**QSHIP**<**meanTotalSales, 0, 1)  
*# remove "QSHIP" from train data*  
trainData <- trainData[,**-**1]  
*# remove "QSHIP" from test data*  
testData <- testData[,**-**1]  
  
*# Step 5: See if we can do a better job predicting 'good' and 'bad' shipments*  
*# convert "goodQSHIP" in train data from numeric to factor*  
trainData**$**goodQSHIP <- **as.factor**(trainData**$**goodQSHIP)  
*# convert "goodQSHIP" in test data from numeric to factor*  
testData**$**goodQSHIP <- **as.factor**(testData**$**goodQSHIP)  
  
*# 1) Build a model*   
*# build a model using ksvm function and use all other variables to predict*  
svmGood <- **ksvm**(goodQSHIP**~**., *# set "QSHIP" as target variable; "." means use all other variables to predict "QSHIP"*  
 data=trainData, *# specify the data to use in the analysis*  
 kernel="rbfdot", *# kernel function that projects the low dimensional problem into higher dimensional space*  
 kpar="automatic",*# kpar refer to parameters that can be used to control the radial function kernel(rbfdot)*  
 C=10, *# C refers to "COST of Constrains"*  
 cross=10, *# use 10 fold cross validation in this model*  
 prob.model=TRUE *# use probability model in this model*  
)  
*# check the model*  
svmGood

## Support Vector Machine object of class "ksvm"   
##   
## SV type: C-svc (classification)   
## parameter : cost C = 10   
##   
## Gaussian Radial Basis kernel function.   
## Hyperparameter : sigma = 12.5713919050905   
##   
## Number of Support Vectors : 75   
##   
## Objective Function Value : -192.2901   
## Training error : 0.007508   
## Cross validation error : 0.008978   
## Probability model included.

*# 2) Test the model*  
goodPred <- **predict**(svmGood, *# use model "svmGood" to predict*  
 testData *# use testData to do the test*  
)  
*# create a dataframe that contains the exact "goodQSHIP" value and the predicted "goodQSHIP"*  
compGood1 <- **data.frame**(testData[,3], goodPred)  
*# change column names*  
**colnames**(compGood1) <- **c**("test","Pred")  
*# Compute the percentage of correct cases*  
perc\_ksvm <- **length**(**which**(compGood1**$**test**==**compGood1**$**Pred))**/dim**(compGood1)[1]  
perc\_ksvm

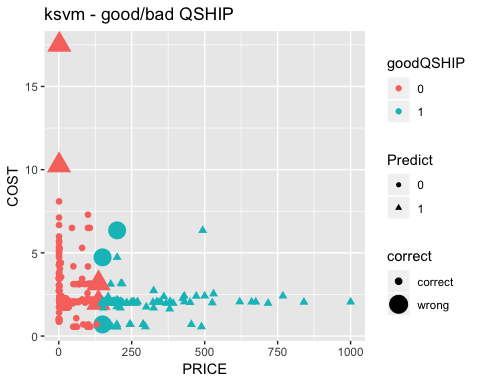
## [1] 0.9790419

*# Confusion Matrix*  
*#*   
results <- **table**(test=compGood1**$**test, pred=compGood1**$**Pred)  
**print**(results)

## pred  
## test 0 1  
## 0 210 4  
## 1 3 117

*# pred*  
*# test 0 1*  
*# 0 210 4 # read horizontal,, 0 class, 210 identified correctly, 4 incorrectly*  
*# 1 3 117 # 1 class, 3 identified incorrectly, 117 correctly*  
  
  
  
  
*# 3) Plot the results.*   
*# determine the prediction is "correct" or "wrong" for each case*  
compGood1**$**correct <- **ifelse**(compGood1**$**test**==**compGood1**$**Pred,"correct","wrong")  
*# create a new dataframe contains correct, QSHIP and PRICE, and goodZone*  
Plot\_ksvm <- **data.frame**(compGood1**$**correct,testData**$**QSHIP,testData**$**COST,testData**$**goodQSHIP,compGood1**$**Pred)  
*# change column names*  
**colnames**(Plot\_ksvm) <- **c**("correct","PRICE","COST","goodQSHIP","Predict")  
*# polt result using ggplot*  
*# size representing correct/wrong; color representing actual good/bad day; shape representing predicted good/bad day.*  
plot.ksvm.good <- **ggplot**(Plot\_ksvm, **aes**(x=PRICE,y=COST)) **+**   
 **geom\_point**(**aes**(size=correct,color=goodQSHIP,shape = Predict))**+**  
 **ggtitle**("ksvm - good/bad QSHIP")  
  
plot.ksvm.good

## Warning: Using size for a discrete variable is not advised.



*# --------------------------------------------------------------------SVM Model*  
svmGood2 <- **svm**(goodQSHIP**~**.,data=trainData,kernel="radial",C=10,cross=10,prob.model=TRUE)  
svmGood2

##   
## Call:  
## svm(formula = goodQSHIP ~ ., data = trainData, kernel = "radial",   
## C = 10, cross = 10, prob.model = TRUE)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.5   
##   
## Number of Support Vectors: 167

*#test the svm model*  
goodPred2 <- **predict**(svmGood2,testData)  
  
compGood2 <- **data.frame**(testData[,3],goodPred2)  
**colnames**(compGood2) <- **c**("test","Pred")  
perc\_svm <- **length**(**which**(compGood2**$**test**==**compGood2**$**Pred))**/dim**(compGood2)[1]  
perc\_svm

## [1] 0.9730539

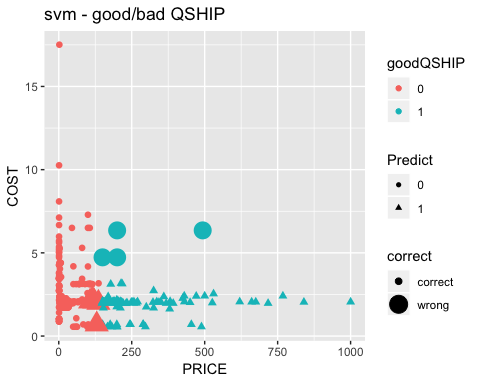
*#confusion matrix*  
results2 <- **table**(test=compGood2**$**test,pred=compGood2**$**Pred)  
**print**(results2)

## pred  
## test 0 1  
## 0 209 5  
## 1 4 116

*# pred*  
*# test 0 1*  
*# 0 209 5 # read horizontal,, 0 class, 209 identified correctly, 5 incorrectly*  
*# 1 4 116 # 1 class, 4 identified incorrectly, 116 correctly*

*#plot results of svm model*  
compGood2**$**correct <- **ifelse**(compGood2**$**test**==**compGood2**$**Pred,"correct","wrong")  
plot.svm <- **data.frame**(compGood2**$**correct,testData**$**QSHIP,testData**$**COST,testData**$**goodQSHIP,compGood2**$**Pred)  
**colnames**(plot.svm) <- **c**("correct","PRICE","COST","goodQSHIP","Predict")  
plot.svm.good <- **ggplot**(plot.svm,**aes**(x=PRICE,y=COST)) **+** **geom\_point**(**aes**(size=correct,color=goodQSHIP,shape=Predict)) **+** **ggtitle**("svm - good/bad QSHIP")  
plot.svm.good

## Warning: Using size for a discrete variable is not advised.



*#-------------------------------------------------------------------build a naive bayes model*  
*#Use the Function "naiveBayes"*  
*# build a model using naiveBayes function and use all other variables to predict*  
svmGood3 <- **naiveBayes**(goodQSHIP**~**., *# set "QSHIP" as target variable; "." means use all other variables to predict "QSHIP"*  
 data=trainData, *# specify the data to use in the analysis*  
 kernel="rbfdot", *# kernel function that projects the low dimensional problem into higher dimensional space*  
 kpar="automatic",*# kpar refer to parameters that can be used to control the radial function kernel(rbfdot)*  
 C=10, *# C refers to "COST of Constrains"*  
 cross=10, *# use 10 fold cross validation in this model*  
 prob.model=TRUE *# use probability model in this model*  
)  
*# check the model*  
svmGood3

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace, kernel = "rbfdot",   
## kpar = "automatic", C = 10, cross = 10, prob.model = TRUE)  
##   
## A-priori probabilities:  
## Y  
## 0 1   
## 0.6501502 0.3498498   
##   
## Conditional probabilities:  
## COST  
## Y [,1] [,2]  
## 0 2.799307 1.7743049  
## 1 1.933476 0.6921739  
##   
## QSHIP  
## Y [,1] [,2]  
## 0 30.92753 37.12601  
## 1 396.96996 863.64634

*# 2) Test the model*  
goodPred3 <- **predict**(svmGood3, *# use model "svmGood" to predict*  
 testData *# use testData to do the test*  
)  
*# create a dataframe that contains the exact "goodQSHIP" value and the predicted "goodQSHIP"*  
compGood3 <- **data.frame**(testData[,3], goodPred3)  
*# change column names*  
**colnames**(compGood3) <- **c**("test","Pred")  
*# Compute the percentage of correct cases*  
naiveBayes <- **length**(**which**(compGood3**$**test**==**compGood3**$**Pred))**/dim**(compGood3)[1]  
naiveBayes

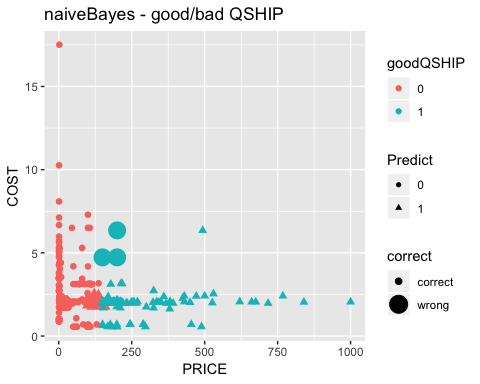
## [1] 0.9850299

*# Confusion Matrix*  
*#*   
results3 <- **table**(test=compGood3**$**test, pred=compGood3**$**Pred)  
**print**(results3)

## pred  
## test 0 1  
## 0 212 2  
## 1 3 117

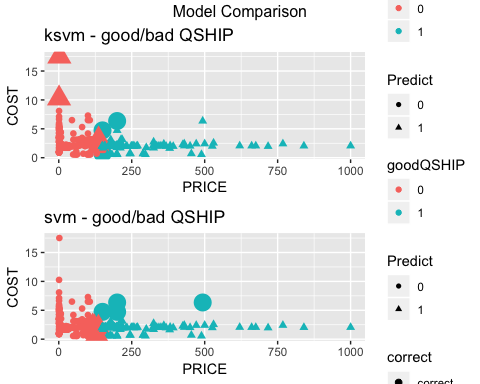
*# pred*  
*# test 0 1*  
*# 0 212 2 # read horizontal,, 0 class, 212 identified correctly, 2 incorrectly*  
*# 1 3 117 # 1 class, 3 identified incorrectly, 117 correctly*  
  
  
*#plot results of svm model*  
compGood3**$**correct <- **ifelse**(compGood3**$**test**==**compGood3**$**Pred,"correct","wrong")  
plot.naiveBayes <- **data.frame**(compGood3**$**correct,testData**$**QSHIP,testData**$**COST,testData**$**goodQSHIP,compGood3**$**Pred)  
**colnames**(plot.naiveBayes) <- **c**("correct","PRICE","COST","goodQSHIP","Predict")  
plot.nb.good <- **ggplot**(plot.naiveBayes,**aes**(x=PRICE,y=COST)) **+** **geom\_point**(**aes**(size=correct,color=goodQSHIP,shape=Predict)) **+** **ggtitle**("naiveBayes - good/bad QSHIP")  
plot.nb.good

## Warning: Using size for a discrete variable is not advised.



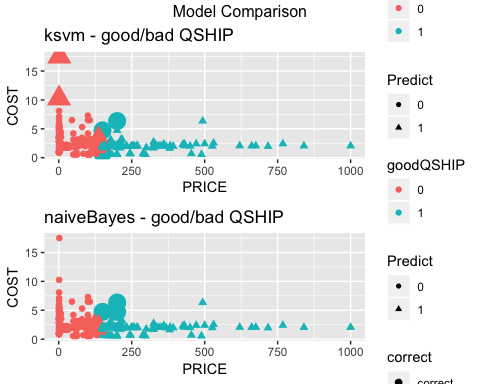
*#5. Show all three results in one window using gridArrange function*  
**grid.arrange**(plot.ksvm.good,plot.svm.good, ncol=1, nrow=2, top="Model Comparison")

## Warning: Using size for a discrete variable is not advised.  
  
## Warning: Using size for a discrete variable is not advised.



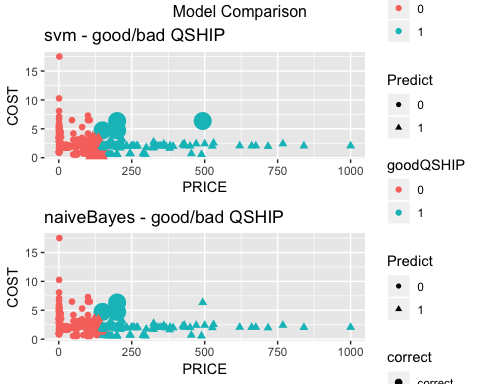
**grid.arrange**(plot.ksvm.good,plot.nb.good, ncol=1, nrow=2, top="Model Comparison")

## Warning: Using size for a discrete variable is not advised.  
  
## Warning: Using size for a discrete variable is not advised.



**grid.arrange**(plot.svm.good,plot.nb.good, ncol=1, nrow=2, top="Model Comparison")

## Warning: Using size for a discrete variable is not advised.  
  
## Warning: Using size for a discrete variable is not advised.



Naive bayes model has the best prediction model among all three models.

## Word Cloud

We use word cloud analysis to analyze the frequency of keywords. Our target is to find the most frequent customer by running the analysis against STNAME. STNAME is also known as customer delivery address name.

mlk <- **readLines**(**file**("MLK.txt"))

## Warning in readLines(file("MLK.txt")): incomplete final line found on  
## 'MLK.txt'

mlk <- mlk[**which**(mlk **!=** "")] *#remove all blank lines in the text*  
  
*#Create a term matrix*  
*#interprets each element of the "mlk" as a document and create a vector source*  
words.vec <- **VectorSource**(mlk)  
*#create a Corpus, a "Bag of Words"*  
words.corpus <- **Corpus**(words.vec)  
*#first step transformation: make all of the letters in "words.corpus" lowercase*  
words.corpus <- **tm\_map**(words.corpus, **content\_transformer**(tolower))

## Warning in tm\_map.SimpleCorpus(words.corpus, content\_transformer(tolower)):  
## transformation drops documents

*#second step transformation: remove the punctuation in "words.corpus"*  
words.corpus <- **tm\_map**(words.corpus, removePunctuation)

## Warning in tm\_map.SimpleCorpus(words.corpus, removePunctuation):  
## transformation drops documents

*#third step transformation: remove numbers in "words.corpus"*  
words.corpus <- **tm\_map**(words.corpus, removeNumbers)

## Warning in tm\_map.SimpleCorpus(words.corpus, removeNumbers): transformation  
## drops documents

*#final step transformation: take out the "stop" words, such as "the", "a" and "at"*  
words.corpus <- **tm\_map**(words.corpus, removeWords, **stopwords**("english"))

## Warning in tm\_map.SimpleCorpus(words.corpus, removeWords,  
## stopwords("english")): transformation drops documents

*#create a term-document matrix "tdm"*  
tdm <- **TermDocumentMatrix**(words.corpus)  
  
*#Create a list of counts for each word*  
*#convert tdm into a matrix called "m"*  
m <- **as.matrix**(tdm)  
  
*#create a list of counts for each word named "wordCounts"*  
wordCounts <- **rowSums**(m)  
wordCounts[1**:**10]

## stname card cardlock credit customer   
## 1 4026 4302 4026 4026   
## groundskeepers evergreen management property associated   
## 782 314 507 345 95

*#sum the total number of words and store the value to "totalWords"*  
totalWords <- **sum**(wordCounts)  
totalWords

## [1] 82229

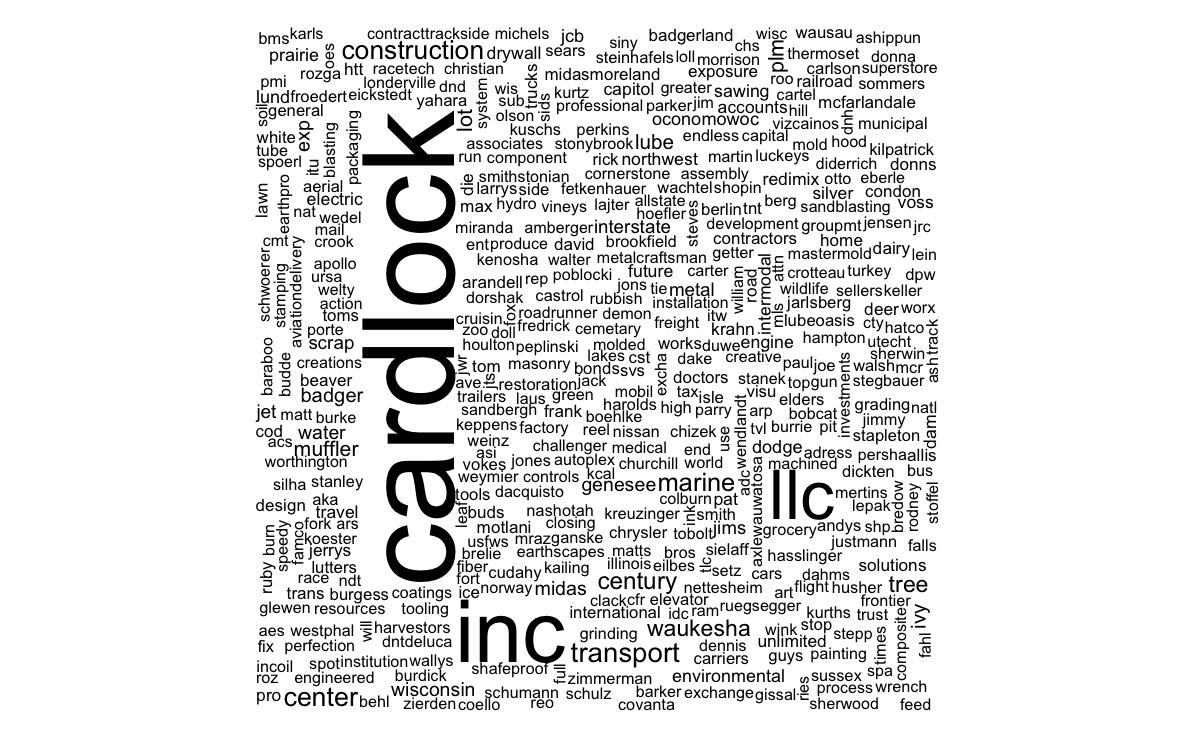
*#create a vector "words" that contains all the words in "wordCounts"*  
words <- **names**(wordCounts)  
**head**(words)

## [1] "stname" "card" "cardlock" "credit"   
## [5] "customer" "groundskeepers"

*#sort words in "wordCounts" by frequency*  
wordCounts <- **sort**(wordCounts, decreasing=TRUE)  
*#check the first several items in "wordCounts" to see if it is built correctly*  
**head**(wordCounts)

## cardlock card credit customer auto inc   
## 4302 4026 4026 4026 2833 2732

*#Build Word Cloud*  
cloudFrame<-**data.frame**(word=**names**(wordCounts),freq=wordCounts)  
**wordcloud**(cloudFrame**$**word,cloudFrame**$**freq)



The most frequent keyword is “Cardlock”. Cardlock is the gas station Corey oil owns. The frequency of keyword is 4302. Base on this result, we can run analysis to find out what products the client purchases at the Cardlock station.

# Data Grouping - Cardlock Sales by Total Sales

Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis[,**c**(6,7,10)]  
**str**(Corey\_Oil\_Analysis\_Rank)

## 'data.frame': 11447 obs. of 3 variables:  
## $ PNUM : Factor w/ 904 levels "","(I)TIM-3628-50",..: 295 295 295 295 239 295 198 198 198 198 ...  
## $ PNAME1 : chr "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" "87 Unleaded Reformulated Gasoline - Cardlock" ...  
## $ TotalSales: num 8.46 65.94 15.71 23.8 44.46 ...

Corey\_Oil\_Analysis\_Rank**$**TotalSales <- **as.numeric**(**as.character**(Corey\_Oil\_Analysis\_Rank**$**TotalSales))  
Corey\_Oil\_Analysis\_Rank <- **aggregate**(Corey\_Oil\_Analysis\_Rank**$**TotalSales, **list**(PNUM=Corey\_Oil\_Analysis\_Rank**$**PNUM), sum)  
Corey\_Oil\_Analysis\_Rank <- Corey\_Oil\_Analysis\_Rank[**order**(**-**Corey\_Oil\_Analysis\_Rank**$**x),]  
  
**head.matrix**(Corey\_Oil\_Analysis\_Rank,n=30)

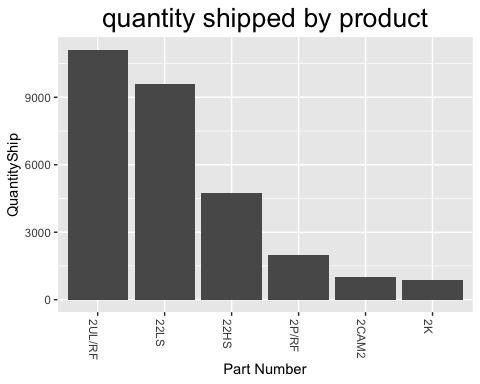
## PNUM x  
## 22 12HS 1159855.81  
## 23 12LS 1098406.28  
## 74 5W30SB 274142.66  
## 212 METHANOL 203542.60  
## 40 1UL/RF 182298.94  
## 158 DS15W40 163474.05  
## 152 DEF 159153.14  
## 15 110B 139526.02  
## 27 1379B 121668.00  
## 71 5W20SB 116725.58  
## 204 HY46 116545.25  
## 84 622515001097 110029.09  
## 201 HY32 93381.73  
## 126 80565-29811 82262.70  
## 45 22LS 82160.23  
## 90 622723001097 78230.53  
## 58 2UL/RF 76275.34  
## 64 41550 65982.03  
## 75 5W30SYN 63917.40  
## 222 THP 59914.57  
## 139 AF32 57498.24  
## 127 80565-30911 55644.01  
## 133 AF111 52795.39  
## 89 622721001097 52352.63  
## 36 1K 46410.94  
## 138 AF31 45592.31  
## 39 1UL/C 44639.85  
## 160 E85 43405.33  
## 137 AF30 42152.24  
## 19 11HS 34353.75

Corey\_Oil\_Analysis\_CardlockSales <- **sqldf**("select \* from Corey\_Oil\_Analysis where UM = 'G' and STNAME = 'Cardlock Credit Card Customer'")  
Corey\_Oil\_Analysis\_CardlockSales **$**TotalSales <- **as.numeric**(**as.character**(Corey\_Oil\_Analysis\_CardlockSales **$**TotalSales))  
**head**(Corey\_Oil\_Analysis\_CardlockSales )

## TCode Refn CNUM SHIPDATE DUEDATE PNUM  
## 1 IN 85273 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 2 IN 85272 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 3 IN 85269 101 1/1/2018 0:00:00 1/31/2018 0:00:00 2UL/RF  
## 4 IN 85268 101 1/1/2018 0:00:00 1/31/2018 0:00:00 22LS  
## 5 IN 85312 101 1/2/2018 0:00:00 2/1/2018 0:00:00 2K  
## 6 IN 85310 101 1/2/2018 0:00:00 2/1/2018 0:00:00 2UL/RF  
## PNAME1 PRICE COST TotalSales MEAS  
## 1 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 8.46 1  
## 2 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 15.71 1  
## 3 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 23.80 1  
## 4 #2 Ultra Low Sulfur On Road Diesel - Cardlock 2.24 2.14 44.46 1  
## 5 Kerosene - Cardlock 3.00 2.31 15.30 1  
## 6 87 Unleaded Reformulated Gasoline - Cardlock 1.94 1.73 50.52 1  
## UM UP QSHIP STNAME STCITY STZIP TotalCost  
## 1 G BULK 4.36 Cardlock Credit Card Customer North Lake 53064 7.54  
## 2 G BULK 8.10 Cardlock Credit Card Customer North Lake 53064 14.01  
## 3 G BULK 12.27 Cardlock Credit Card Customer North Lake 53064 21.23  
## 4 G BULK 19.85 Cardlock Credit Card Customer North Lake 53064 42.48  
## 5 G BULK 5.10 Cardlock Credit Card Customer North Lake 53064 11.78  
## 6 G BULK 26.04 Cardlock Credit Card Customer North Lake 53064 45.05  
## Profit  
## 1 0.92  
## 2 1.70  
## 3 2.57  
## 4 1.98  
## 5 3.52  
## 6 5.47

# Histogram - Cardlock Sales by Total Sales

Cardlock<- **setNames**(**aggregate**(Corey\_Oil\_Analysis\_CardlockSales**$**QSHIP, by=**list**(Corey\_Oil\_Analysis\_CardlockSales**$**PNUM), FUN=sum), **c**("PartNumber","QuantityShip"))  
Cardlock <- **ggplot**( data = Cardlock, **aes**(x=**reorder** (PartNumber, **-**QuantityShip), y=QuantityShip)) **+** **geom\_bar**(stat="identity")  
Cardlock <- Cardlock **+** **theme**(axis.text.x = **element\_text**(angle=**-**90, hjust=0)) **+ggtitle**("quantity shipped by product")  
Cardlock <- Cardlock **+** **scale\_x\_discrete**(name="Part Number")  
Cardlock <- Cardlock **+** **theme**(plot.title = **element\_text**(size=20, hjust=0.5))  
Cardlock



After filtering the data to only include Cardlock client, we identified the products Corey Oil sold are 2UL/RF, 22LS, 22HS, 2P/RF, 2CAM2, and 2K.

## Skills Used

# packages

ggplot2,reshape2,maps,mapdata,zipcode,sqldf,gdata,kernlab,e1071 ,gridExtra,neuralnet

# Functions

read.csv() - Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.  
summary() - It is a generic function used to produce result summaries of the results of various model fitting functions.  
is.na() - It indicates which elements are missing.  
any() - Given a set of logical vectors, is at least one of the values true?  
na.omit() - It returns the object with incomplete cases removed.  
gsub() - Replace the first occurrence of a pattern with sub or replace all occurrences with gsub.  
round() - It rounds the values in its first argument to the specified number of decimal places (default 0).  
head(), tail() - Returns the first or last parts of a vector, matrix, table, data frame or function.  
str() - Compactly display the internal structure of an R object, a diagnostic function and an alternative to the summary  
as.numeric() - Creates or coerces objects of type “numeric”.  
as.character() - The function returns a string of 1’s and 0’s or a character vector of features depending on the nature of the fingerprint supplied.  
as.factor() - Convert a column into a factor column.  
as.matrix() - It returns all values of a Raster\* object as a matrix.  
aggregate() - It splits the data into subsets, computes summary statistics for each, and returns the result in a convenient form.  
list() - Functions to construct, coerce and check for both kinds of R lists.  
lm() - lm is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance.  
merge() - Merge two data frames by common columns or row names, or do other versions of database join operations.  
setNames() - This is a convenience function that sets the names on an object and returns the object. It is most useful at the end of a function definition where one is creating the object to be returned and would prefer not to store it under a name just so the names can be assigned.  
colSums() - Form row and column sums and means for numeric arrays (or data frames).  
dim() - Retrieve or set the dimension of an object.  
Set.seed() - set.seed in the simEd package allows the user to simultaneously set the initial seed for both the stats and simEd variate generators.  
data.frame() - The function data.frame() creates data frames, tightly coupled collections of variables which share many of the properties of matrices and of lists, used as the fundamental data structure by most of R’s modeling software.  
legend() - This function can be used to add legends to plots.  
sqrt() - Computes the square root of the specified float value.  
mean() - Generic function for the (trimmed) arithmetic mean.  
rownames(), colnames() - Retrieve or set the row or column names of a matrix-like object.  
ifelse() - It returns a value with the same shape as test which is filled with elements selected from either yes or no depending on whether the elements of the test is TRUE or FALSE.  
length() - Get or set the length of vectors (including lists) and factors, and of any other R object for which a method has been defined.  
table() - It uses the cross-classifying factors to build a contingency table of the counts at each combination of factor levels.  
readLines() - Read some or all text lines from a connection.  
rowSums() - Sum values of Raster objects by row or column.  
sum() - It returns the sum of all the values present in its arguments.  
names() - Functions to get or set the names of an object.

## Conclusion

Corey Oil has years of wholesale history in Wisconsin. Through this project, Corey Oil was able to identify its strength in different products. Top 30 analyses identified the best selling products by total sales, total quantity shipped, total profits, and product costs. The products identified are 12HS, 12LS, 5W30SB, 2UL/RF. They are off-road diesel, on-road diesel, 5W30 semi-synthetic motor oil, and gasoline, respectively. Since Corey Oil is a wholesaler, the total profits showed the characteristic of low profit but high quantity overall. By running the linear regression analysis, we can see the relationship between total costs and total sales. The prediction model has helped Corey Oil to confirm the product profit is around 9.3% (2327/25000). The map plots help the company understand where the majority of sales are located. The deliveries are very condensed, mostly in north and south Milwaukee, WI. Word count analysis helped Corey Oil identified the most frequent transactions which occurred mostly at Cardlock gas station. To sum up, by running different analyses, Corey Oil can understand its strength and extend to more profitable business strategy.