FinalProject@PDA (mailto:FinalProject@PDA)

Priyanka Settipalli

2024-04-21

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
Cellphone <- read.csv("C:/Users/91630/OneDrive/Desktop/PDA/PDA project/CellPhone Data/Cellpho
ne.csv")
```

```
#Data cleaning and preparation

#summary Of Cellphone data
summary(Cellphone)
```

```
##    Product_id         Price           Sale            weight     
##  Min.   :  10.0   Min.   : 614   Min.   :  10.0   Min.   : 66.0  
##  1st Qu.: 237.0   1st Qu.:1734   1st Qu.:  37.0   1st Qu.:134.1  
##  Median : 774.0   Median :2258   Median : 106.0   Median :153.0  
##  Mean   : 675.6   Mean   :2216   Mean   : 621.5   Mean   :170.4  
##  3rd Qu.:1026.0   3rd Qu.:2744   3rd Qu.: 382.0   3rd Qu.:170.0  
##  Max.   :1339.0   Max.   :4361   Max.   :9807.0   Max.   :753.0  
##   resoloution          ppi           cpu.core        cpu.freq    
##  Min.   : 1.40   Min.   :121.0   Min.   :0.000   Min.   :0.000  
##  1st Qu.: 4.80   1st Qu.:233.0   1st Qu.:4.000   1st Qu.:1.200  
##  Median : 5.15   Median :294.0   Median :4.000   Median :1.400  
##  Mean   : 5.21   Mean   :335.1   Mean   :4.857   Mean   :1.503  
##  3rd Qu.: 5.50   3rd Qu.:428.0   3rd Qu.:8.000   3rd Qu.:1.875  
##  Max.   :12.20   Max.   :806.0   Max.   :8.000   Max.   :2.700  
##   internal.mem         ram             RearCam         Front_Cam     
##  Min.   :  0.0   Min.   :0.000   Min.   : 0.00   Min.   : 0.000  
##  1st Qu.:  8.0   1st Qu.:1.000   1st Qu.: 5.00   1st Qu.: 0.000  
##  Median : 16.0   Median :2.000   Median :12.00   Median : 5.000  
##  Mean   : 24.5   Mean   :2.205   Mean   :10.38   Mean   : 4.503  
##  3rd Qu.: 32.0   3rd Qu.:3.000   3rd Qu.:16.00   3rd Qu.: 8.000  
##  Max.   :128.0   Max.   :6.000   Max.   :23.00   Max.   :20.000  
##     battery       thickness     
##  Min.   : 800   Min.   : 5.100  
##  1st Qu.:2040   1st Qu.: 7.600  
##  Median :2800   Median : 8.400  
##  Mean   :2842   Mean   : 8.922  
##  3rd Qu.:3240   3rd Qu.: 9.800  
##  Max.   :9500   Max.   :18.500  
```

```
#Datatype of all entities
str(Cellphone)
```

```
## 'data.frame':    161 obs. of  14 variables:
##  $ Product_id  : int  203 880 40 99 880 947 774 947 99 1103 ...
##  $ Price       : int  2357 1749 1916 1315 1749 2137 1238 2137 1315 2580 ...
##  $ Sale        : int  10 10 10 11 11 12 13 13 14 15 ...
##  $ weight      : num  135 125 110 118 125 ...
##  $ resoloution : num  5.2 4 4.7 4 4 5.5 4 5.5 4 5.1 ...
##  $ ppi         : int  424 233 312 233 233 401 233 401 233 432 ...
##  $ cpu.core    : int  8 2 4 2 2 4 2 4 2 4 ...
##  $ cpu.freq    : num  1.35 1.3 1.2 1.3 1.3 2.3 1.2 2.3 1.3 2.5 ...
##  $ internal.mem: num  16 4 8 4 4 16 8 16 4 16 ...
##  $ ram         : num  3 1 1.5 0.512 1 2 1 2 0.512 2 ...
##  $ RearCam     : num  13 3.15 13 3.15 3.15 16 2 16 3.15 16 ...
##  $ Front_Cam   : num  8 0 5 0 0 8 0 8 0 2 ...
##  $ battery     : int  2610 1700 2000 1400 1700 2500 1560 2500 1400 2800 ...
##  $ thickness   : num  7.4 9.9 7.6 11 9.9 9.5 11.7 9.5 11 8.1 ...
```

```
#Number of Null values in Cellphone data
sum(is.na(Cellphone))
```

```
## [1] 0
```

```
#Number of Duplicate values  in Cellphone data
sum(duplicated(Cellphone))
```

```
## [1] 0
```

```
#Correlations between all entities
library(tidyverse) # for data manipulation
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

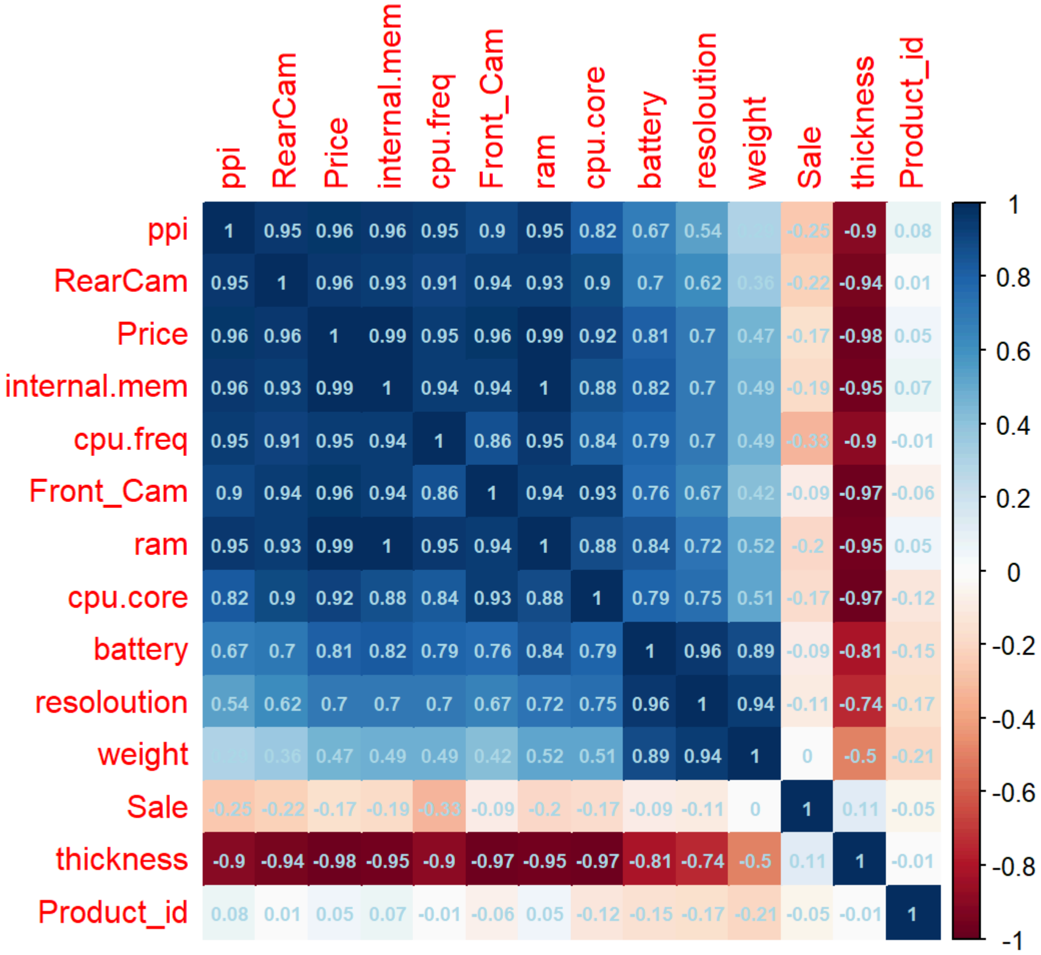
```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.4.4     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
Cellphone_data = cor(Cellphone, method = "spearman")
correlation_matrix <- cor(Cellphone_data, use = "pairwise.complete.obs")



#using corrplot function
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

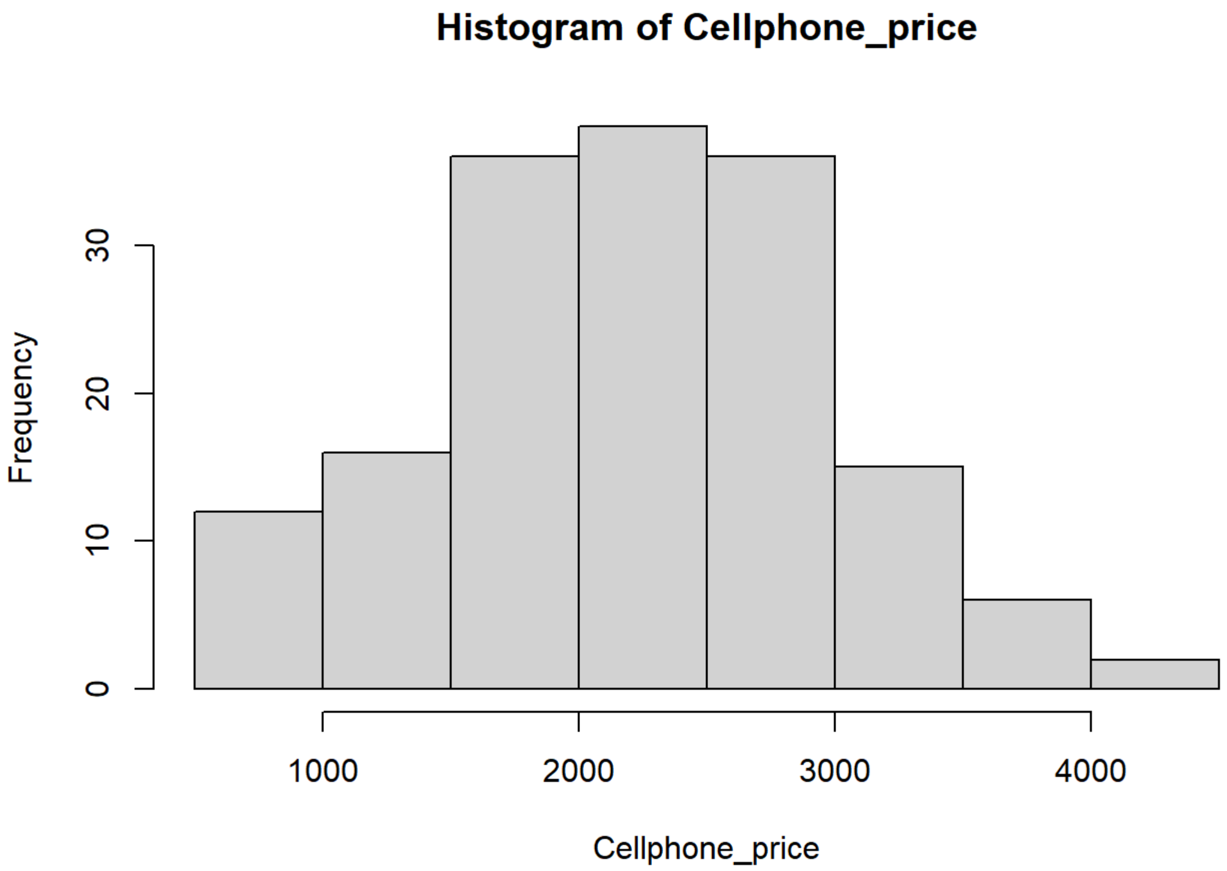
```
## corrplot 0.92 loaded
```

```
corrplot(correlation_matrix, method = "color", addCoef.col = "lightblue",order = "AOE", numbe
r.cex = 0.60)
```

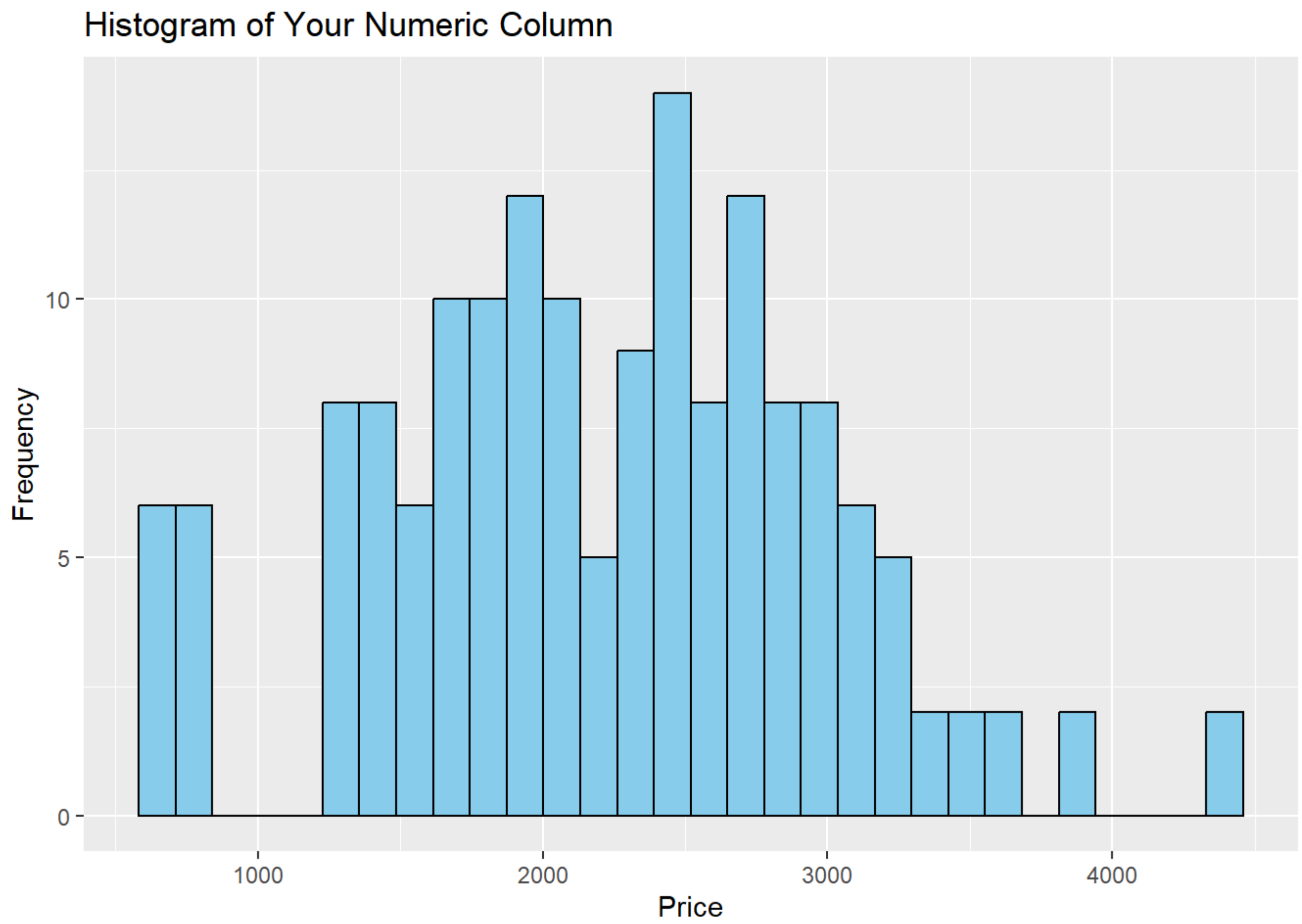


	ppi	RearCam	Price	internal.mem	cpu.freq	Front_Cam	ram	cpu.core	battery	resoloution	weight	Sale	thickness	Product_id
ppi	1	0.95	0.96	0.96	0.95	0.9	0.95	0.82	0.67	0.54	0.29	-0.25	-0.9	0.08
RearCam	0.95	1	0.96	0.93	0.91	0.94	0.93	0.9	0.7	0.62	0.36	-0.22	-0.94	0.01
Price	0.96	0.96	1	0.99	0.95	0.96	0.99	0.92	0.81	0.7	0.47	-0.17	-0.98	0.05
internal.mem	0.96	0.93	0.99	1	0.94	0.94	1	0.88	0.82	0.7	0.49	-0.19	-0.95	0.07
cpu.freq	0.95	0.91	0.95	0.94	1	0.86	0.95	0.84	0.79	0.7	0.49	-0.33	-0.9	-0.01
Front_Cam	0.9	0.94	0.96	0.94	0.86	1	0.94	0.93	0.76	0.67	0.42	-0.09	-0.97	-0.06
ram	0.95	0.93	0.99	1	0.95	0.94	1	0.88	0.84	0.72	0.52	-0.2	-0.95	0.05
cpu.core	0.82	0.9	0.92	0.88	0.84	0.93	0.88	1	0.79	0.75	0.51	-0.17	-0.97	-0.12
battery	0.67	0.7	0.81	0.82	0.79	0.76	0.84	0.79	1	0.96	0.89	-0.09	-0.81	-0.15
resoloution	0.54	0.62	0.7	0.7	0.7	0.67	0.72	0.75	0.96	1	0.94	-0.11	-0.74	-0.17
weight	0.29	0.36	0.47	0.49	0.49	0.42	0.52	0.51	0.89	0.94	1	0	-0.5	-0.21
Sale	-0.25	-0.22	-0.17	-0.19	-0.33	-0.09	-0.2	-0.17	-0.09	-0.11	0	1	0.11	-0.05
thickness	-0.9	-0.94	-0.98	-0.95	-0.9	-0.97	-0.95	-0.97	-0.81	-0.74	-0.5	0.11	1	-0.01
Product_id	0.08	0.01	0.05	0.07	-0.01	-0.06	0.05	-0.12	-0.15	-0.17	-0.21	-0.05	-0.01	1

```
#Histogram
Cellphone_price <- Cellphone$Price
hist(Cellphone_price)
```



```
library(ggplot2)
ggplot(Cellphone, aes(x=Price))+geom_histogram(bins = 30, fill = "skyblue", color = "black")
+
  labs(title = "Histogram of Your Numeric Column",
       x = "Price",
       y = "Frequency")
```



```
summary_plot <- summary(Cellphone$Price)

Median <- summary_plot["Median"]
Median
```

```
## Median
##   2258
```

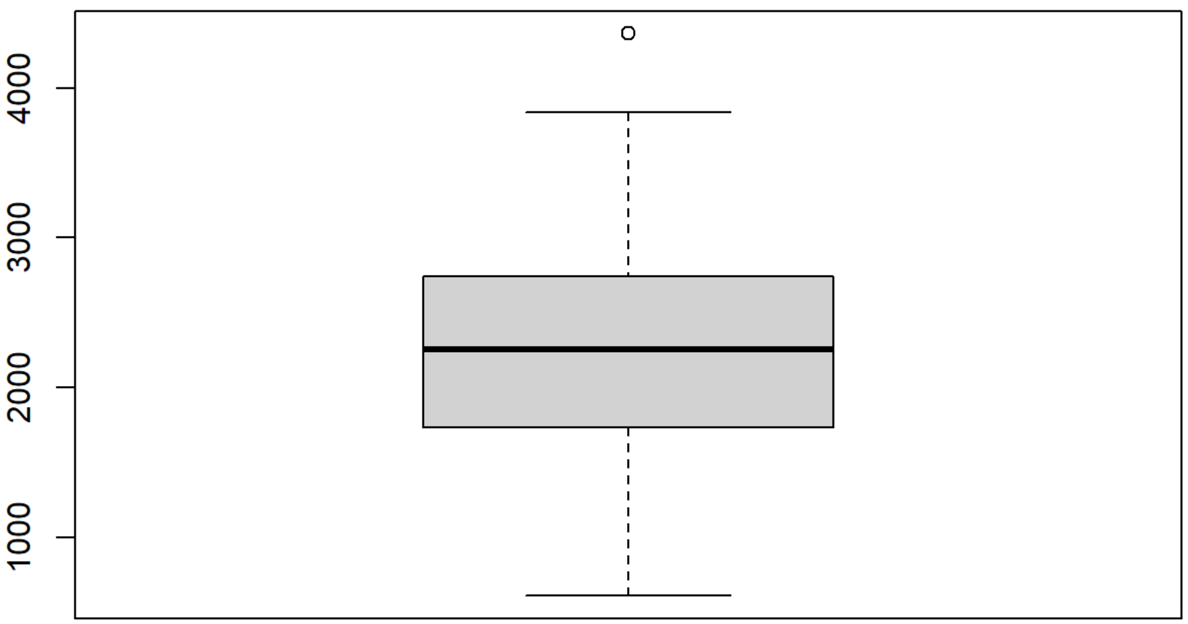
```
Q1 <- summary_plot["1st Qu."]
Q1
```

```
## 1st Qu.
##    1734
```

```
Q3 <- summary_plot["3rd Qu."]
Q3
```

```
## 3rd Qu.
##    2744
```

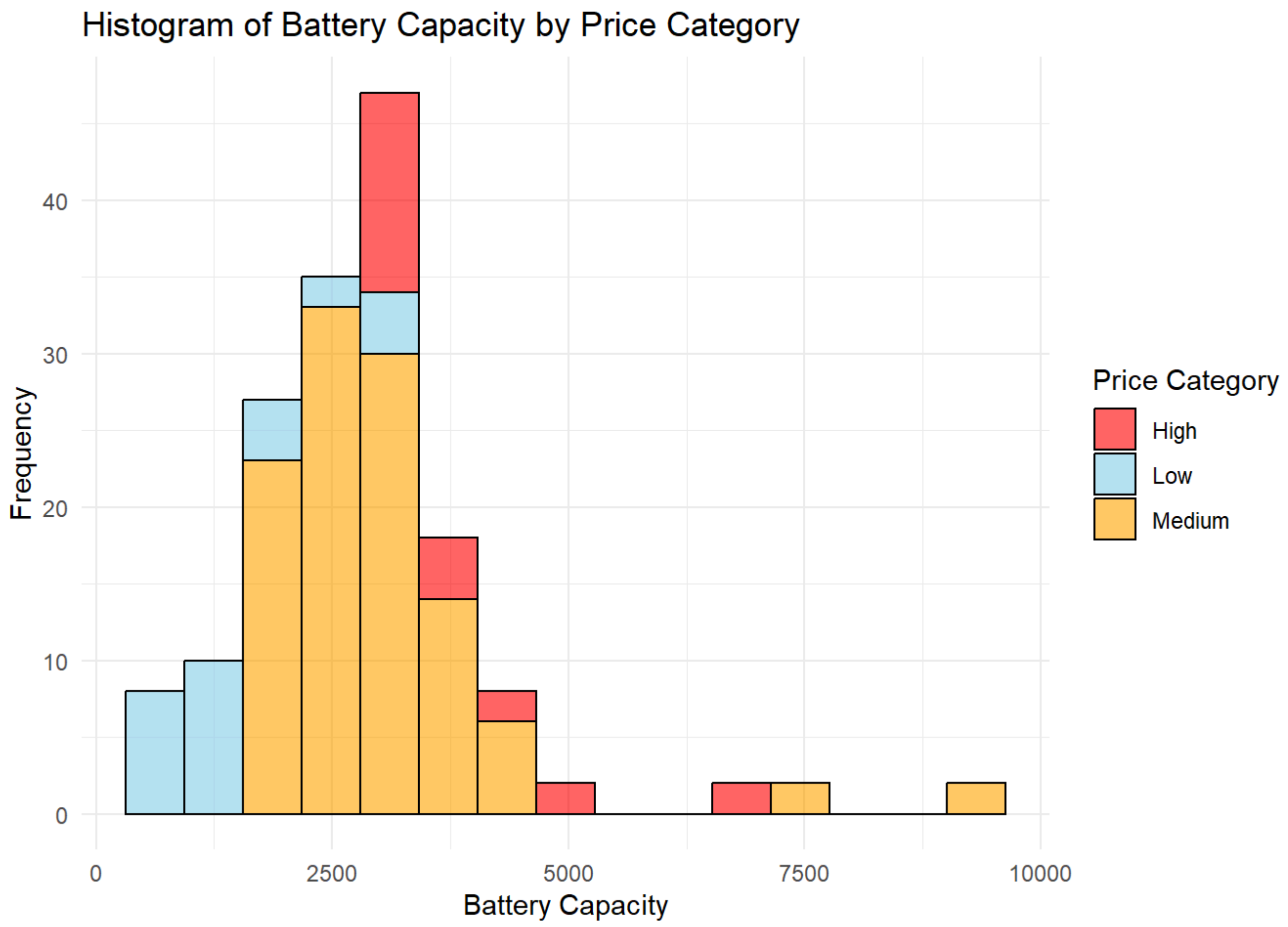
```
#Boxplot
boxplot(Cellphone_price)
```



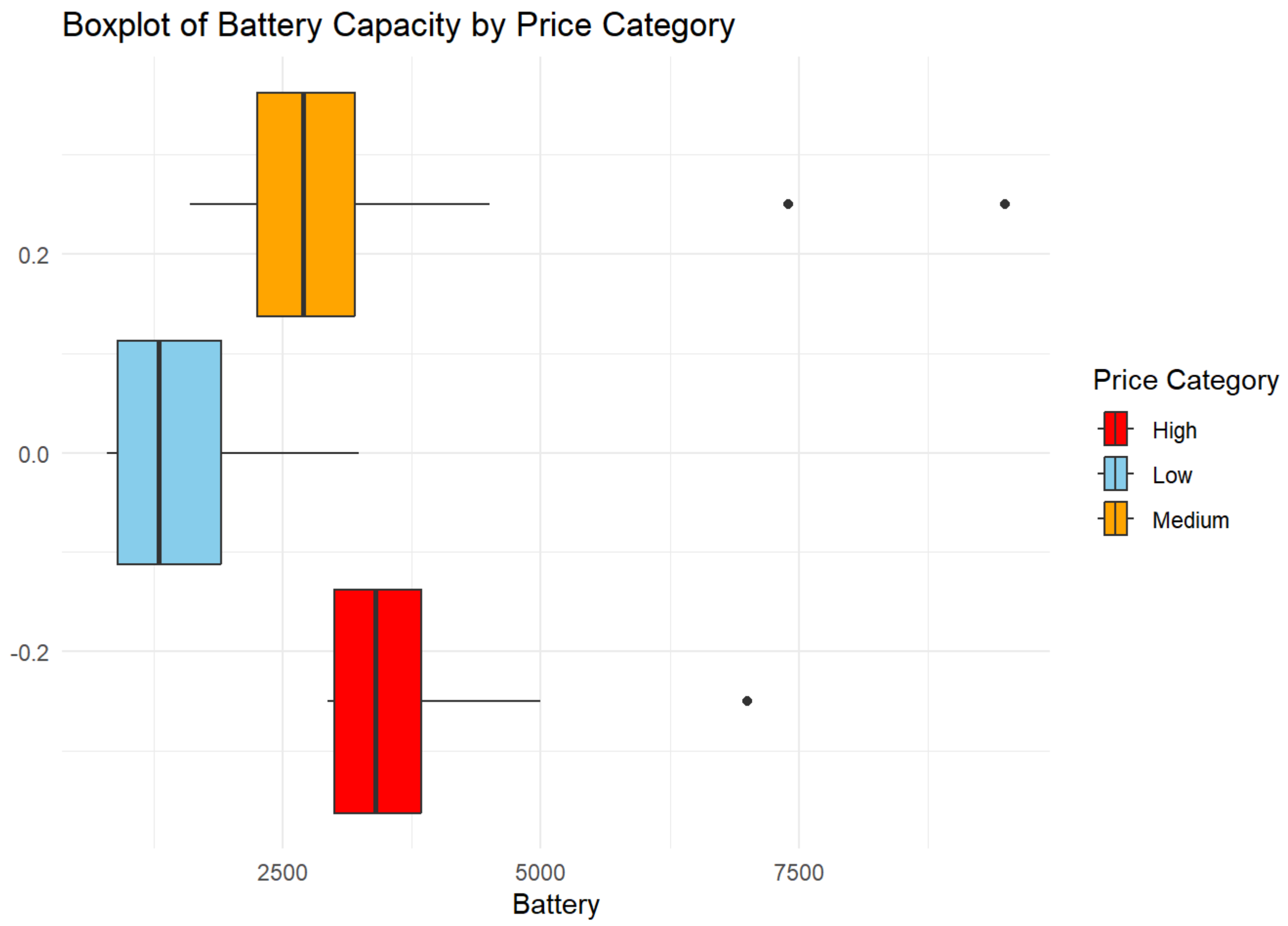
```
summary(Cellphone_price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     614    1734    2258    2216    2744    4361
```

```
#Histogram Overlay
library(dplyr)
Cellphone_new <- dplyr::mutate(Cellphone, HighMediumLowprice = case_when(Price <= 1500 ~ 'Low', Price > 1500 & Price <= 3000 ~ 'Medium', Price > 3000 ~ 'High'))

#Histogram with specified changes

ggplot(Cellphone_new, aes(x=battery, fill = HighMediumLowprice))+geom_histogram(alpha = 0.6, bins = 15, color ="black")+
  labs(
    title = "Histogram of Battery Capacity by Price Category",
    x = "Battery Capacity",
    y = "Frequency"
  )+
  theme_minimal() +
  scale_fill_manual(
    name = "Price Category",
    values = c ("Low" = "Skyblue",
                "Medium" = "Orange",
                "High" = "red")
  )
```



```
#Boxplot overlay
ggplot(data = Cellphone_new, mapping = aes( x= battery))+geom_boxplot(aes(fill = HighMediumLo
wprice))+
  labs(
    title = "Boxplot of Battery Capacity by Price Category",
    x = "Battery",
  ) +
  theme_minimal() +
  scale_fill_manual(
    name = "Price Category",
    values = c("Low" = "Skyblue",
               "Medium" = "Orange",
               "High" = "red")
  )
```



```
#Hypothesis Testing

#F-test
var.test(Cellphone$Price, Cellphone$battery, alternative = "two.sided" )
```

```
##
##  F test to compare two variances
##
## data:  Cellphone$Price and Cellphone$battery
## F = 0.31579, num df = 160, denom df = 160, p-value = 1.328e-12
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2313851 0.4309932
## sample estimates:
## ratio of variances
##          0.3157933
```

```
result_two_sided <- t.test(Cellphone$Price, Cellphone$battery, alternative = "two.sided" )
result_two_sided
```

```
##
##  Welch Two Sample t-test
##
## data:  Cellphone$Price and Cellphone$battery
## t = -5.0697, df = 251.89, p-value = 7.723e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -869.8963 -383.1347
## sample estimates:
## mean of x mean of y
##  2215.596  2842.112
```

```
result_less <- t.test(Cellphone$Price, Cellphone$battery, alternative = "less")
result_less
```

```
##
##  Welch Two Sample t-test
##
## data:  Cellphone$Price and Cellphone$battery
## t = -5.0697, df = 251.89, p-value = 3.862e-07
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -422.495
## sample estimates:
## mean of x mean of y
##  2215.596  2842.112
```

```
result_greater <- t.test(Cellphone$Price, Cellphone$battery, alternative = "greater")
result_greater
```

```
##
##  Welch Two Sample t-test
##
## data:  Cellphone$Price and Cellphone$battery
## t = -5.0697, df = 251.89, p-value = 1
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  -830.5361       Inf
## sample estimates:
## mean of x mean of y
##  2215.596  2842.112
```

```
#Case2
var.test(Cellphone$Price, Cellphone$weight, alternative = "two.sided" )
```

```
## 
##  F test to compare two variances
## 
## data:  Cellphone$Price and Cellphone$weight
## F = 68.393, num df = 160, denom df = 160, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  50.11199 93.34191
## sample estimates:
## ratio of variances 
##           68.39261
```

```
result_two_sided <- t.test(Cellphone$Price, Cellphone$weight, alternative = "two.sided" )
result_two_sided
```

```
## 
##  Welch Two Sample t-test
## 
## data:  Cellphone$Price and Cellphone$weight
## t = 33.537, df = 164.68, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1924.762 2165.579
## sample estimates:
## mean of x mean of y 
## 2215.5963  170.4261
```

```
result_less <- t.test(Cellphone$Price, Cellphone$weight, alternative = "less")
result_less
```

```
## 
##  Welch Two Sample t-test
## 
## data:  Cellphone$Price and Cellphone$weight
## t = 33.537, df = 164.68, p-value = 1
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 2146.045
## sample estimates:
## mean of x mean of y 
## 2215.5963  170.4261
```

```
result_greater <- t.test(Cellphone$Price, Cellphone$weight, alternative = "greater")
result_greater
```

```
## 
##  Welch Two Sample t-test
## 
## data:  Cellphone$Price and Cellphone$weight
## t = 33.537, df = 164.68, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  1944.295      Inf
## sample estimates:
## mean of x mean of y 
## 2215.5963  170.4261
```

```
#Simple Linear Regression

linear_model <- lm(Price ~ battery, data = Cellphone)
summary(linear_model)
```

```
## 
## Call:
## lm(formula = Price ~ battery, data = Cellphone)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1819.60  -390.14    18.25   436.39  1382.92 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.321e+03  1.164e+02  11.353  < 2e-16 ***
## battery     3.147e-01  3.692e-02   8.522 1.14e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 638.5 on 159 degrees of freedom
## Multiple R-squared:  0.3135, Adjusted R-squared:  0.3092 
## F-statistic: 72.62 on 1 and 159 DF,  p-value: 1.142e-14
```

```
set.seed(9)
```

```
i <- sample(2, nrow(Cellphone), replace = TRUE, prob = c(0.85, 0.25))
Cellphone_training <- Cellphone[i==1,]
Cellphone_testing <- Cellphone[i==2,]
summary(Cellphone_training)
```

```
##    Product_id         Price           Sale            weight
##  Min.   :  14.0   Min.   : 614   Min.   :  10.0   Min.   : 66.0
##  1st Qu.: 198.0   1st Qu.:1676   1st Qu.:  40.0   1st Qu.:133.5
##  Median : 774.0   Median :2137   Median : 109.0   Median :152.0
##  Mean   : 662.8   Mean   :2163   Mean   : 609.9   Mean   :166.5
##  3rd Qu.:1058.0   3rd Qu.:2744   3rd Qu.: 367.0   3rd Qu.:170.0
##  Max.   :1339.0   Max.   :3837   Max.   :9807.0   Max.   :753.0
##   resoloution          ppi         cpu.core        cpu.freq
##  Min.   : 1.400   Min.   :121   Min.   :0.000   Min.   :0.000
##  1st Qu.: 4.700   1st Qu.:233   1st Qu.:4.000   1st Qu.:1.200
##  Median : 5.100   Median :294   Median :4.000   Median :1.350
##  Mean   : 5.125   Mean   :330   Mean   :4.661   Mean   :1.464
##  3rd Qu.: 5.500   3rd Qu.:428   3rd Qu.:8.000   3rd Qu.:1.875
##  Max.   :12.200   Max.   :806   Max.   :8.000   Max.   :2.700
##   internal.mem         ram            RearCam        Front_Cam
##  Min.   :  0.00   Min.   :0.000   Min.   : 0.00   Min.   : 0.000
##  1st Qu.:  8.00   1st Qu.:1.000   1st Qu.: 5.00   1st Qu.: 0.000
##  Median : 16.00   Median :2.000   Median :12.00   Median : 5.000
##  Mean   : 22.78   Mean   :2.111   Mean   :10.22   Mean   : 4.312
##  3rd Qu.: 32.00   3rd Qu.:3.000   3rd Qu.:16.00   3rd Qu.: 8.000
##  Max.   :128.00   Max.   :6.000   Max.   :23.00   Max.   :20.000
##     battery       thickness
##  Min.   : 800   Min.   : 5.100
##  1st Qu.:2040   1st Qu.: 7.700
##  Median :2600   Median : 8.500
##  Mean   :2704   Mean   : 9.043
##  3rd Qu.:3200   3rd Qu.: 9.900
##  Max.   :9500   Max.   :18.500
```

```
summary(Cellphone_testing)
```

```
##    Product_id         Price          Sale            weight     
##  Min.   :  10.0   Min.   : 614   Min.   :  15.0   Min.   : 69.8  
##  1st Qu.: 361.5   1st Qu.:1909   1st Qu.:  33.5   1st Qu.:145.0  
##  Median : 769.5   Median :2333   Median :  90.5   Median :161.0  
##  Mean   : 714.3   Mean   :2375   Mean   : 656.5   Mean   :182.4  
##  3rd Qu.:1020.0   3rd Qu.:2748   3rd Qu.: 453.0   3rd Qu.:178.5  
##  Max.   :1339.0   Max.   :4361   Max.   :4408.0   Max.   :489.0  
##   resoloution          ppi           cpu.core       cpu.freq    
##  Min.   : 1.400   Min.   :129.0   Min.   :0.00   Min.   :0.000  
##  1st Qu.: 5.000   1st Qu.:234.5   1st Qu.:4.00   1st Qu.:1.275  
##  Median : 5.480   Median :376.5   Median :4.00   Median :1.525  
##  Mean   : 5.468   Mean   :350.5   Mean   :5.45   Mean   :1.620  
##  3rd Qu.: 5.700   3rd Qu.:429.0   3rd Qu.:8.00   3rd Qu.:1.950  
##  Max.   :10.100   Max.   :806.0   Max.   :8.00   Max.   :2.500  
##   internal.mem        ram           RearCam        Front_Cam     
##  Min.   :  0.0   Min.   :0.004   Min.   : 0.00   Min.   : 0.000  
##  1st Qu.:  8.0   1st Qu.:1.000   1st Qu.: 5.00   1st Qu.: 2.000  
##  Median : 16.0   Median :2.000   Median :12.00   Median : 5.000  
##  Mean   : 29.7   Mean   :2.488   Mean   :10.85   Mean   : 5.082  
##  3rd Qu.: 32.0   3rd Qu.:3.250   3rd Qu.:16.00   3rd Qu.: 8.000  
##  Max.   :128.0   Max.   :6.000   Max.   :23.00   Max.   :16.000  
##     battery       thickness     
##  Min.   : 800   Min.   : 5.100  
##  1st Qu.:2175   1st Qu.: 7.500  
##  Median :3050   Median : 8.300  
##  Mean   :3260   Mean   : 8.555  
##  3rd Qu.:3630   3rd Qu.: 9.475  
##  Max.   :7400   Max.   :14.100
```

```
dim(Cellphone_training)
```

```
## [1] 121  14
```

```
dim(Cellphone_testing)
```

```
## [1] 40 14
```

```
#Model 1 using stepwise model selection:Forward model propogation

library(MASS)
```

```
## 
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
## 
##     select
```

```
intercept_only <- lm(Price ~ 1, data = Cellphone_training[, 1:14])
all_model <-lm(Price ~ ., data = Cellphone_training[, 1:14])
forward <- stepAIC(intercept_only, direction = "forward", scope = formula(all_model))
```

```
## Start:  AIC=1603.03
## Price ~ 1
## 
##                 Df Sum of Sq      RSS    AIC
## + ram           1  54149739 13337879 1408.8
## + ppi           1  48047344 19440274 1454.4
## + RearCam       1  44953058 22534560 1472.3
## + thickness     1  38886958 28600660 1501.2
## + cpu.freq      1  38375704 29111913 1503.3
## + internal.mem  1  37164780 30322838 1508.2
## + cpu.core      1  32330868 35156750 1526.1
## + Front_Cam     1  30919487 36568131 1530.9
## + battery       1  19701242 47786376 1563.3
## + resoloution   1  14342533 53145085 1576.1
## + Sale          1   3551950 63935668 1598.5
## + Product_id    1   3292239 64195379 1599.0
## + weight        1   1540195 65947423 1602.2
## <none>                      67487618 1603.0
## 
## Step:  AIC=1408.85
## Price ~ ram
## 
##                 Df Sum of Sq      RSS    AIC
## + thickness     1   6399854  6938025 1331.8
## + cpu.core      1   6123149  7214730 1336.5
## + RearCam       1   4025187  9312692 1367.4
## + ppi           1   3501321  9836558 1374.0
## + cpu.freq      1   3168699 10169180 1378.0
## + Front_Cam     1   1371349 11966530 1397.7
## + resoloution   1   1213065 12124814 1399.3
## + internal.mem  1    528972 12808906 1406.0
## + Sale          1    440241 12897638 1406.8
## + battery       1    391210 12946669 1407.2
## <none>                      13337879 1408.8
## + Product_id    1    162744 13175135 1409.4
## + weight        1     12502 13325376 1410.7
## 
## Step:  AIC=1331.77
## Price ~ ram + thickness
## 
##                 Df Sum of Sq     RSS    AIC
## + ppi           1   1869639 5068385 1295.8
## + RearCam       1   1373465 5564560 1307.1
## + cpu.core      1   1318076 5619949 1308.3
## + cpu.freq      1    609846 6328179 1322.6
## + Front_Cam     1    325828 6612197 1327.9
## <none>                      6938025 1331.8
## + weight        1     57391 6880634 1332.8
## + Sale          1     52150 6885875 1332.8
## + Product_id    1     40949 6897075 1333.0
## + resoloution   1      8175 6929849 1333.6
## + battery       1      4546 6933479 1333.7
## + internal.mem  1       116 6937908 1333.8
## 
## Step:  AIC=1295.77
## Price ~ ram + thickness + ppi
##
##                 Df Sum of Sq     RSS    AIC
## + cpu.core      1   1138817 3929568 1267.0
## + Front_Cam     1    447996 4620390 1286.6
## + RearCam       1    287697 4780688 1290.7
## + battery       1    169745 4898640 1293.7
## <none>                      5068385 1295.8
## + resoloution   1     49621 5018764 1296.6
## + weight        1     30175 5038211 1297.0
## + cpu.freq      1     29112 5039274 1297.1
## + Product_id    1     23584 5044802 1297.2
## + internal.mem  1     23519 5044866 1297.2
## + Sale          1     13797 5054588 1297.4
##
## Step:  AIC=1266.98
## Price ~ ram + thickness + ppi + cpu.core
##
##                 Df Sum of Sq     RSS    AIC
## + internal.mem  1    307691 3621877 1259.1
## + Front_Cam     1     91674 3837894 1266.1
## + RearCam       1     82550 3847018 1266.4
## <none>                      3929568 1267.0
## + Product_id    1     40100 3889468 1267.7
## + cpu.freq      1     18859 3910709 1268.4
## + battery       1     15161 3914407 1268.5
## + Sale          1     10991 3918577 1268.6
## + weight        1      3963 3925605 1268.9
## + resoloution   1      1762 3927807 1268.9
##
## Step:  AIC=1259.11
## Price ~ ram + thickness + ppi + cpu.core + internal.mem
##
##                Df Sum of Sq     RSS    AIC
## + RearCam      1    144005 3477872 1256.2
## + Sale         1     88570 3533307 1258.1
## + cpu.freq     1     69412 3552465 1258.8
## <none>                     3621877 1259.1
## + Front_Cam    1     44994 3576883 1259.6
## + battery      1     17614 3604263 1260.5
## + Product_id   1     15937 3605940 1260.6
## + weight       1      3418 3618459 1261.0
## + resoloution  1       728 3621149 1261.1
##
## Step:  AIC=1256.2
## Price ~ ram + thickness + ppi + cpu.core + internal.mem + RearCam
##
##                Df Sum of Sq     RSS    AIC
## + Sale         1    216242 3261630 1250.4
## + cpu.freq     1     59248 3418624 1256.1
## <none>                     3477872 1256.2
## + battery      1     29805 3448067 1257.2
## + Product_id   1     12638 3465235 1257.8
## + Front_Cam    1      7786 3470086 1257.9
## + resoloution  1        76 3477796 1258.2
## + weight       1         7 3477865 1258.2
```

```
##
## Step:  AIC=1250.44
## Price ~ ram + thickness + ppi + cpu.core + internal.mem + RearCam +
##     Sale
##
##                Df Sum of Sq     RSS    AIC
## + cpu.freq      1     56195 3205435 1250.3
## <none>                     3261630 1250.4
## + Front_Cam     1     44156 3217475 1250.8
## + Product_id    1     22019 3239612 1251.6
## + battery       1     17614 3244016 1251.8
## + resoloution   1        58 3261572 1252.4
## + weight        1        13 3261618 1252.4
##
## Step:  AIC=1250.33
## Price ~ ram + thickness + ppi + cpu.core + internal.mem + RearCam +
##     Sale + cpu.freq
##
##                Df Sum of Sq     RSS    AIC
## + Front_Cam     1     72942 3132493 1249.5
## <none>                     3205435 1250.3
## + Product_id    1     22795 3182640 1251.5
## + resoloution   1      9245 3196190 1252.0
## + battery       1      5383 3200052 1252.1
## + weight        1      3848 3201587 1252.2
##
## Step:  AIC=1249.55
## Price ~ ram + thickness + ppi + cpu.core + internal.mem + RearCam +
##     Sale + cpu.freq + Front_Cam
##
##                Df Sum of Sq     RSS    AIC
## <none>                     3132493 1249.5
## + Product_id    1     36943 3095550 1250.1
## + battery       1     14030 3118463 1251.0
## + resoloution   1      2418 3130075 1251.5
## + weight        1        10 3132483 1251.5
```

```
summary(forward)
```

```
## 
## Call:
## lm(formula = Price ~ ram + thickness + ppi + cpu.core + internal.mem + 
##     RearCam + Sale + cpu.freq + Front_Cam, data = Cellphone_training[, 
##     1:14])
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -343.80 -112.37  -11.89  101.94  388.30 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1393.79244  151.79197   9.182 2.73e-15 ***
## ram           110.18262   30.68543   3.591 0.000493 ***
## thickness     -54.55738   11.03889  -4.942 2.76e-06 ***
## ppi             1.02407    0.24764   4.135 6.91e-05 ***
## cpu.core       60.77391   10.59850   5.734 8.57e-08 ***
## internal.mem    6.45253    1.45475   4.435 2.17e-05 ***
## RearCam        13.18298    5.30603   2.485 0.014465 *  
## Sale           -0.03589    0.01171  -3.064 0.002740 ** 
## cpu.freq       74.27175   42.80013   1.735 0.085460 .  
## Front_Cam       9.30274    5.78636   1.608 0.110741    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 168 on 111 degrees of freedom
## Multiple R-squared:  0.9536, Adjusted R-squared:  0.9498 
## F-statistic: 253.4 on 9 and 111 DF,  p-value: < 2.2e-16
```

```
forward$anova
```

```
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## Price ~ 1
## 
## Final Model:
## Price ~ ram + thickness + ppi + cpu.core + internal.mem + RearCam + 
##     Sale + cpu.freq + Front_Cam
## 
## 
##              Step Df    Deviance Resid. Df Resid. Dev      AIC
## 1                                      120   67487618 1603.031
## 2           + ram  1 54149739.03       119   13337879 1408.850
## 3     + thickness  1  6399854.34       118    6938025 1331.765
## 4           + ppi  1  1869639.27       117    5068385 1295.772
## 5      + cpu.core  1  1138817.06       116    3929568 1266.978
## 6  + internal.mem  1   307691.04       115    3621877 1259.112
## 7       + RearCam  1   144004.79       114    3477872 1256.203
## 8          + Sale  1   216242.05       113    3261630 1250.436
## 9      + cpu.freq  1    56195.25       112    3205435 1250.333
## 10    + Front_Cam  1    72942.07       111    3132493 1249.547
```

```
#mean absolute error on testing dataset for forward propogation model

library(MLmetrics)
```

```
## Warning: package 'MLmetrics' was built under R version 4.3.3
```

```
##
## Attaching package: 'MLmetrics'
```

```
## The following object is masked from 'package:base':
##
##     Recall
```

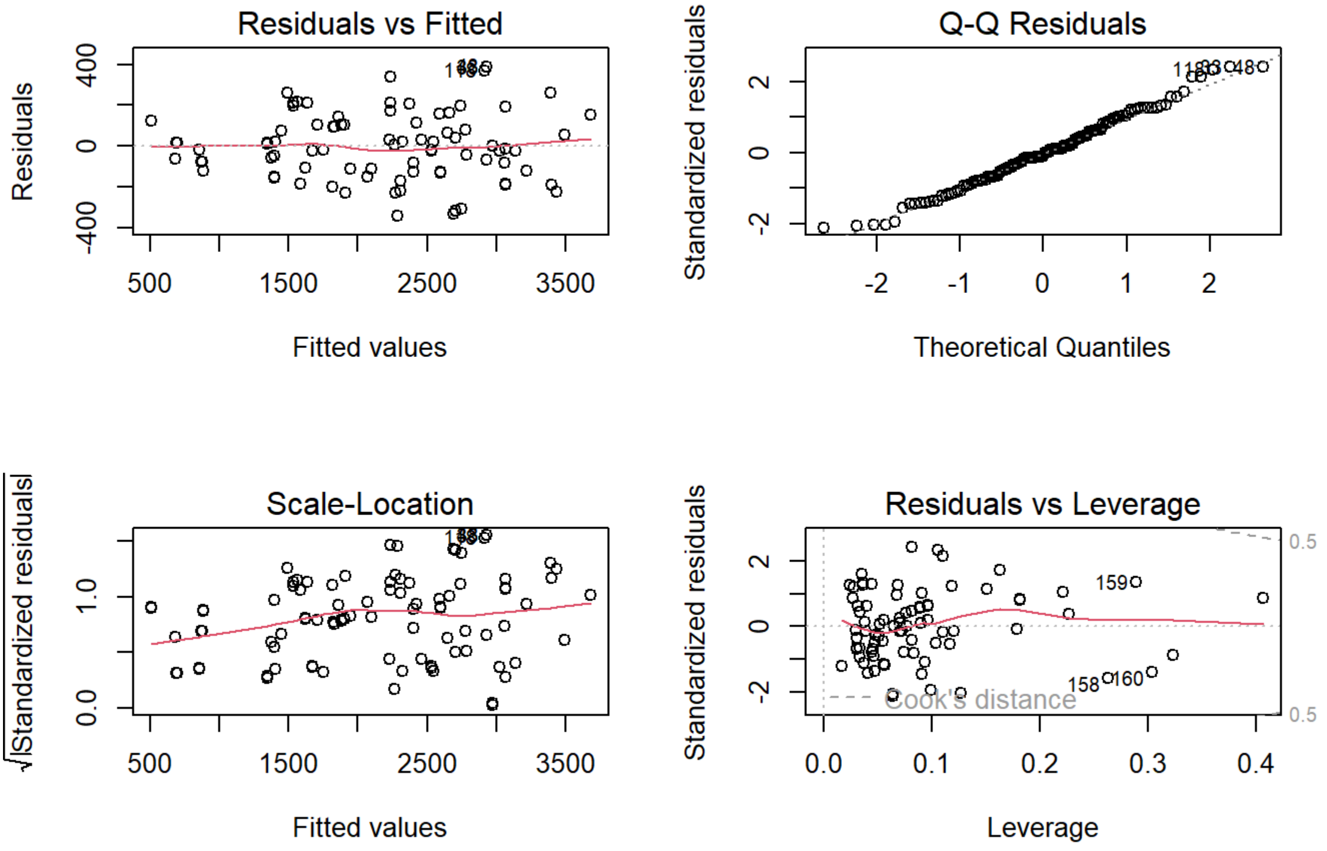
```
forward_pred <- predict(object = forward, newdata = Cellphone_testing[ ,1:14])
MAE(y_pred = forward_pred, y_true = Cellphone_testing$Price)
```

```
## [1] 188.1548
```

```
MSE(y_pred = forward_pred, y_true = Cellphone_testing$Price)
```

```
## [1] 55986.22
```

```
par(mfrow=c(2,2))
plot(forward)
```



```
#backward model propogation
backward <- stepAIC(all_model, direction = "backward")
```

```
## Start:  AIC=1250.47
## Price ~ Product_id + Sale + weight + resoloution + ppi + cpu.core + 
##     cpu.freq + internal.mem + ram + RearCam + Front_Cam + battery + 
##     thickness
## 
##                Df Sum of Sq     RSS    AIC
## - resoloution   1      6487 2960966 1248.7
## - weight        1     23402 2977881 1249.4
## - Product_id    1     44569 2999048 1250.3
## <none>                      2954479 1250.5
## - Front_Cam     1     64004 3018483 1251.1
## - cpu.freq      1     78314 3032793 1251.6
## - RearCam       1    119256 3073735 1253.3
## - battery       1    129941 3084420 1253.7
## - Sale          1    167389 3121867 1255.1
## - ram           1    187761 3142240 1255.9
## - ppi           1    415551 3370029 1264.4
## - internal.mem  1    462512 3416991 1266.1
## - thickness     1    515247 3469725 1267.9
## - cpu.core      1    742237 3696716 1275.6
## 
## Step:  AIC=1248.73
## Price ~ Product_id + Sale + weight + ppi + cpu.core + cpu.freq + 
##     internal.mem + ram + RearCam + Front_Cam + battery + thickness
## 
##                Df Sum of Sq     RSS    AIC
## <none>                      2960966 1248.7
## - Product_id    1     59367 3020334 1249.1
## - Front_Cam     1     61126 3022092 1249.2
## - cpu.freq      1     77178 3038144 1249.8
## - RearCam       1    114292 3075258 1251.3
## - weight        1    119847 3080813 1251.5
## - battery       1    134467 3095433 1252.1
## - Sale          1    163980 3124946 1253.3
## - ram           1    197441 3158407 1254.5
## - ppi           1    436712 3397679 1263.4
## - internal.mem  1    456027 3416993 1264.1
## - thickness     1    729147 3690113 1273.4
## - cpu.core      1    735938 3696904 1273.6
```

```
summary(backward)
```

```
##
## Call:
## lm(formula = Price ~ Product_id + Sale + weight + ppi + cpu.core +
##     cpu.freq + internal.mem + ram + RearCam + Front_Cam + battery +
##     thickness, data = Cellphone_training[, 1:14])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -354.68 -106.77    5.64  117.30  373.06
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1383.10971  151.76268   9.114 4.81e-15 ***
## Product_id      0.05866    0.03986   1.472  0.14406
## Sale           -0.02970    0.01214  -2.446  0.01608 *
## weight         -1.10203    0.52709  -2.091  0.03889 *
## ppi             1.02498    0.25682   3.991  0.00012 ***
## cpu.core       57.40122   11.07912   5.181 1.03e-06 ***
## cpu.freq       72.74387   43.35658   1.678  0.09628 .
## internal.mem    5.92154    1.45193   4.078 8.70e-05 ***
## ram            86.42169   32.20396   2.684  0.00843 **
## RearCam        10.89953    5.33832   2.042  0.04361 *
## Front_Cam       8.85155    5.92805   1.493  0.13831
## battery         0.10203    0.04607   2.215  0.02889 *
## thickness     -56.67940   10.99063  -5.157 1.15e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 165.6 on 108 degrees of freedom
## Multiple R-squared:  0.9561, Adjusted R-squared:  0.9513
## F-statistic: 196.1 on 12 and 108 DF,  p-value: < 2.2e-16
```

```
backward$anova
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## Price ~ Product_id + Sale + weight + resoloution + ppi + cpu.core +
##     cpu.freq + internal.mem + ram + RearCam + Front_Cam + battery +
##     thickness
##
## Final Model:
## Price ~ Product_id + Sale + weight + ppi + cpu.core + cpu.freq +
##     internal.mem + ram + RearCam + Front_Cam + battery + thickness
##
##
##            Step Df Deviance Resid. Df Resid. Dev      AIC
## 1                                 107    2954479 1250.468
## 2 - resoloution  1 6487.362       108    2960966 1248.734
```

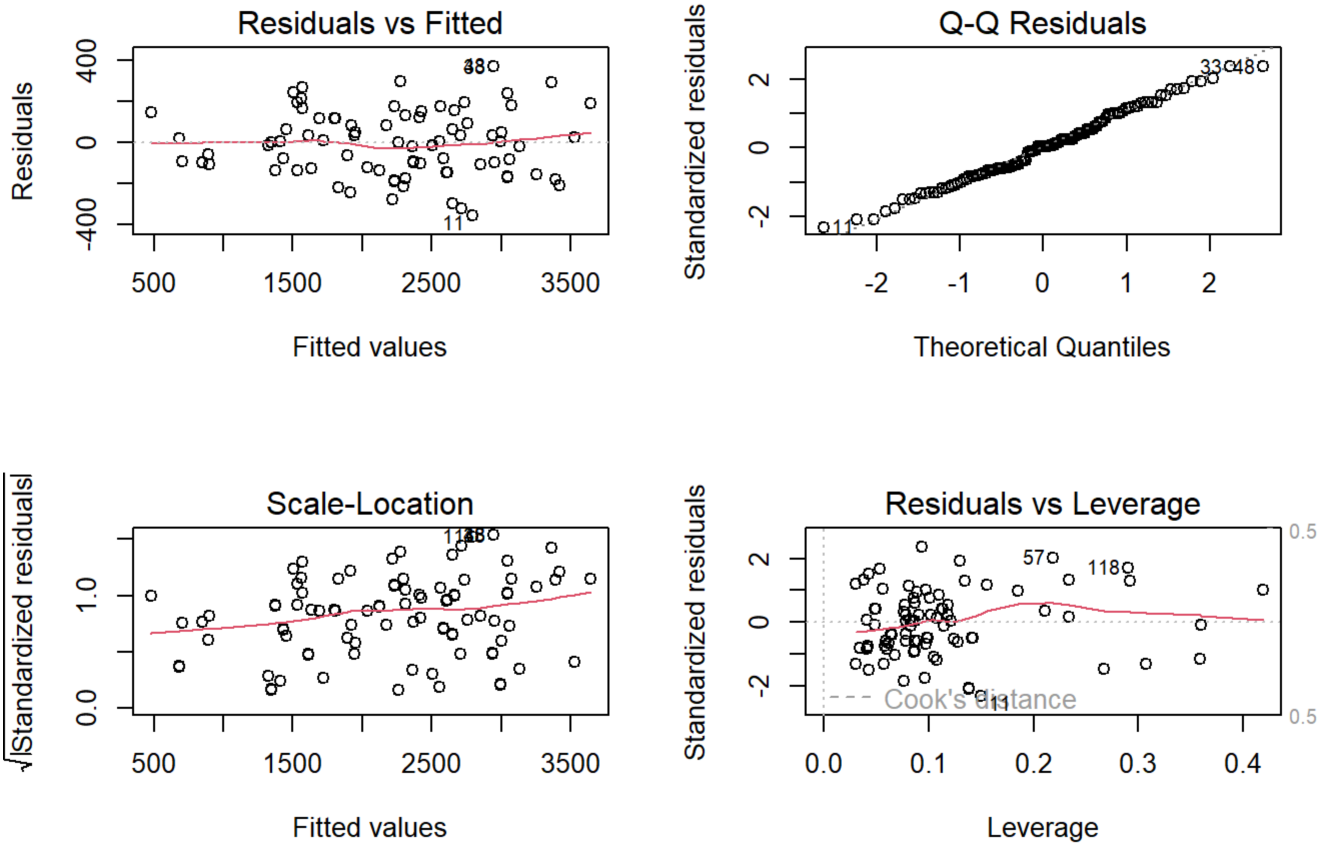
```
#calculate mae and mse for backward

library(MLmetrics)
backward_pred <- predict(object = backward, newdata = Cellphone_testing[ ,1:14])
MAE(y_pred = backward_pred, y_true = Cellphone_testing$Price)
```

```
## [1] 169.2486
```

```
MSE(y_pred = backward_pred, y_true = Cellphone_testing$Price)
```

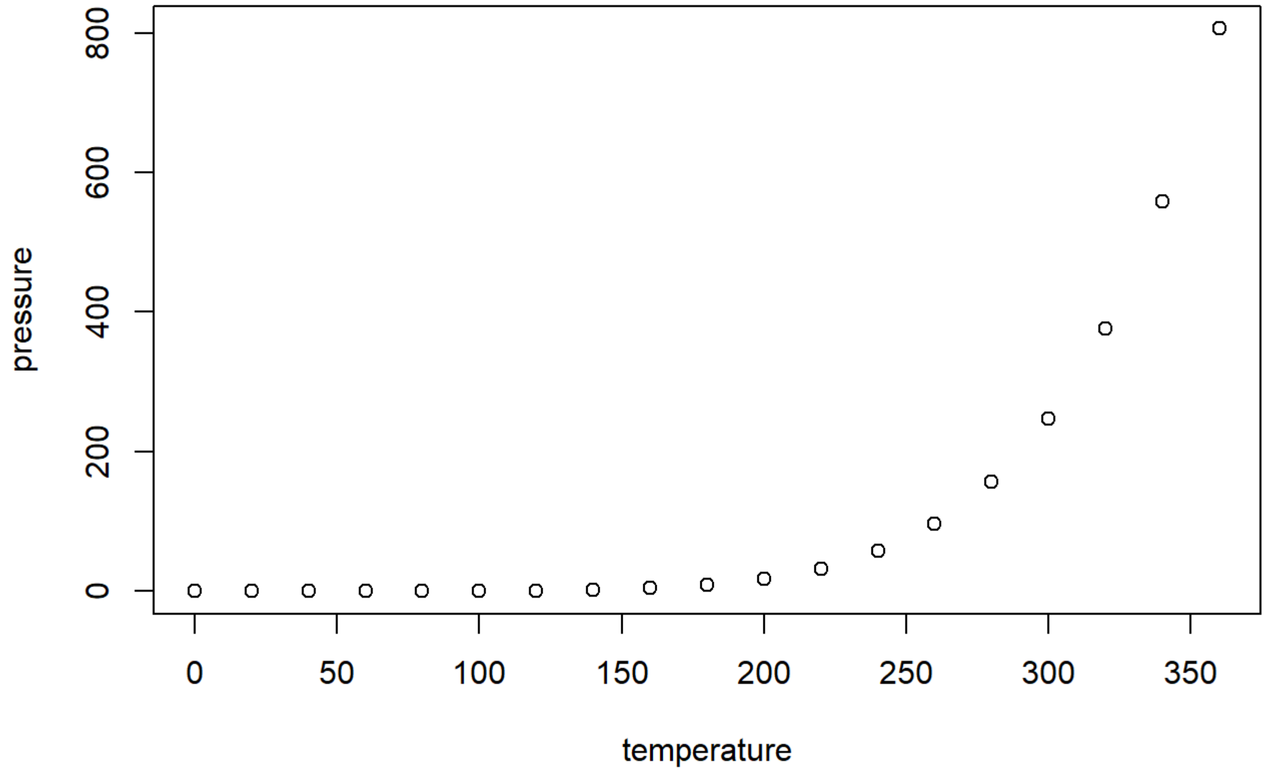
```
## [1] 44396.01
```

```
#plotting
par(mfrow = c(2,2))
plot(backward)
```



## Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.