Progetto

Andrea Corvaglia at All

17 agosto 2019

Contents

Descriptive analysis on Y

```
data <- read.csv("../data/Laptop2.csv")
str(data)
```

```
## 'data.frame':    1303 obs. of  22 variables:
##  $ X                 : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Company           : Factor w/ 19 levels "Acer","Apple",..: 2 2 8 2 2 1 2 2 3 1 ...
##  $ Product           : Factor w/ 618 levels "110-15ACL (A6-7310/4GB/500GB/W10)",..: 302 300 51 302 302 59 302 300 6
##  $ TypeName          : Factor w/ 6 levels "2 in 1 Convertible",..: 5 5 4 5 5 4 5 5 5 5 ...
##  $ Inches            : num  13.3 13.3 15.6 15.4 13.3 15.6 15.4 13.3 14 14 ...
##  $ ScreenResolution  : Factor w/ 40 levels "1366x768","1440x900",..: 24 2 9 26 24 1 26 2 9 16 ...
##  $ Cpu               : Factor w/ 118 levels "AMD A10-Series 9600P 2.4GHz",..: 55 53 64 75 57 15 74 53 96 73 ...
##  $ Ram               : int  8 8 8 16 8 4 16 8 16 8 ...
##  $ Memory            : Factor w/ 38 levels "1024GB HDD","1024GB HDD +  1024GB HDD",..: 8 6 17 29 17 26 16 16 29 17
##  $ Gpu               : Factor w/ 110 levels "AMD FirePro W4190M",..: 59 52 54 10 60 18 61 52 98 62 ...
##  $ OpSys             : Factor w/ 9 levels "Android","Chrome OS",..: 5 5 6 5 5 7 4 5 7 7 ...
##  $ Weight            : num  1.37 1.34 1.86 1.83 1.37 2.1 2.04 1.34 1.3 1.6 ...
##  $ Price             : num  1340 899 575 2537 1804 ...
##  $ Frequenza         : num  2.3 1.8 2.5 2.7 3.1 3 2.2 1.8 1.8 1.6 ...
##  $ Risoluzione       : Factor w/ 15 levels "1366x768","1440x900",..: 11 2 4 13 11 1 13 2 4 4 ...
##  $ Pixel             : int  4096000 1296000 2073600 5184000 4096000 1049088 5184000 1296000 2073600 2073600 ...
##  $ GpuCompany        : Factor w/ 4 levels "AMD","ARM","Intel",..: 3 3 3 1 3 1 3 3 4 3 ...
##  $ MemoriaSSD        : int  128 0 256 512 256 0 0 0 512 256 ...
##  $ SolidStateDisk    : Factor w/ 2 levels "False","True": 2 1 2 2 2 1 1 1 2 2 ...
##  $ TotalMemory       : int  128 128 256 512 256 500 256 256 512 256 ...
##  $ dedicated_GPU     : Factor w/ 2 levels "False","True": 1 1 1 2 1 2 1 1 2 1 ...
##  $ Aggregated_Company: Factor w/ 10 levels "Acer","Apple",..: 2 2 5 2 2 1 2 2 3 1 ...
```

```
head(data,3)
```

```
##   X Company     Product  TypeName Inches
## 1 1   Apple MacBook Pro Ultrabook   13.3
## 2 2   Apple Macbook Air Ultrabook   13.3
## 3 3      HP      250 G6  Notebook   15.6
##                      ScreenResolution                        Cpu Ram
## 1 IPS Panel Retina Display 2560x1600       Intel Core i5 2.3GHz   8
## 2                            1440x900       Intel Core i5 1.8GHz   8
## 3                   Full HD 1920x1080 Intel Core i5 7200U 2.5GHz   8
##                Memory                          Gpu OpSys Weight   Price
## 1           128GB SSD Intel Iris Plus Graphics 640 macOS   1.37 1339.69
## 2 128GB Flash Storage       Intel HD Graphics 6000 macOS   1.34  898.94
## 3           256GB SSD        Intel HD Graphics 620 No OS   1.86  575.00
##   Frequenza Risoluzione   Pixel GpuCompany MemoriaSSD SolidStateDisk
## 1       2.3   2560x1600 4096000      Intel        128           True
## 2       1.8    1440x900 1296000      Intel          0          False
## 3       2.5   1920x1080 2073600      Intel        256           True
##   TotalMemory dedicated_GPU Aggregated_Company
## 1         128         False              Apple
## 2         128         False              Apple
## 3         256         False                 HP
```

```
summary(data)
```

```
##        X             Company                  Product
##  Min.   :   1.0   Dell   :297   XPS 13            :  30
##  1st Qu.: 331.5   Lenovo :297   Inspiron 3567     :  29
##  Median : 659.0   HP     :274   250 G6            :  21
##  Mean   : 660.2   Asus   :158   Legion Y520-15IKBN:  19
##  3rd Qu.: 990.5   Acer   :103   Vostro 3568       :  19
##  Max.   :1320.0   MSI    : 54   Inspiron 5570     :  18
##                   (Other):120   (Other)           :1167
##                TypeName       Inches
##  2 in 1 Convertible:121   Min.   :10.10
##  Gaming            :205   1st Qu.:14.00
##  Netbook           : 25   Median :15.60
##  Notebook          :727   Mean   :15.02
##  Ultrabook         :196   3rd Qu.:15.60
##  Workstation       : 29   Max.   :18.40
##
##                                 ScreenResolution
##  Full HD 1920x1080                         :507
##  1366x768                                  :281
##  IPS Panel Full HD 1920x1080               :230
##  IPS Panel Full HD / Touchscreen 1920x1080: 53
##  Full HD / Touchscreen 1920x1080           : 47
##  1600x900                                  : 23
##  (Other)                                   :162
##                           Cpu           Ram
##  Intel Core i5 7200U 2.5GHz :190   Min.   : 2.000
##  Intel Core i7 7700HQ 2.8GHz:146   1st Qu.: 4.000
##  Intel Core i7 7500U 2.7GHz :134   Median : 8.000
##  Intel Core i7 8550U 1.8GHz : 73   Mean   : 8.382
##  Intel Core i5 8250U 1.6GHz : 72   3rd Qu.: 8.000
##  Intel Core i5 6200U 2.3GHz : 68   Max.   :64.000
##  (Other)                    :620
##                    Memory                         Gpu
##  256GB SSD             :412   Intel HD Graphics 620  :281
##  1024GB HDD            :224   Intel HD Graphics 520  :185
##  500GB HDD             :132   Intel UHD Graphics 620 : 68
##  512GB SSD             :118   Nvidia GeForce GTX 1050: 66
##  128GB SSD +  1024GB HDD: 94   Nvidia GeForce GTX 1060: 48
##  128GB SSD             : 76   Nvidia GeForce 940MX   : 43
##  (Other)               :247   (Other)                :612
##         OpSys          Weight          Price        Frequenza
##  Windows 10:1072   Min.   :0.690   Min.   : 174   Min.   :0.900
##  No OS     :  66   1st Qu.:1.500   1st Qu.: 599   1st Qu.:2.000  
##  Linux     :  62   Median :2.040   Median : 977   Median :2.500  
##  Windows 7 :  45   Mean   :2.039   Mean   :1124   Mean   :2.299  
##  Chrome OS :  27   3rd Qu.:2.300   3rd Qu.:1488   3rd Qu.:2.700  
##  macOS     :  13   Max.   :4.700   Max.   :6099   Max.   :3.600  
##  (Other)   :  18                                                 
##     Risoluzione      Pixel          GpuCompany    MemoriaSSD   
##  1920x1080:841   Min.   :1049088   AMD   :180   Min.   :  0.0  
##  1366x768 :308   1st Qu.:1440000   ARM   :  1   1st Qu.:  0.0  
##  3840x2160: 43   Median :2073600   Intel :722   Median :128.0  
##  3200x1800: 27   Mean   :2168807   Nvidia:400   Mean   :170.5  
##  1600x900 : 23   3rd Qu.:2073600                3rd Qu.:256.0  
##  2560x1440: 23   Max.   :8294400                Max.   :512.0  
##  (Other)  : 38                                                 
##  SolidStateDisk  TotalMemory     dedicated_GPU Aggregated_Company
##  False:476      Min.   :   8.0   False:723     Dell   :297       
##  True :827      1st Qu.: 256.0   True :580     Lenovo :297       
##                 Median : 500.0                 HP     :274       
##                 Mean   : 620.1                 Asus   :158       
##                 3rd Qu.:1024.0                 Acer   :103       
##                 Max.   :2560.0                 MSI    : 54       
##                                                (Other):120
```

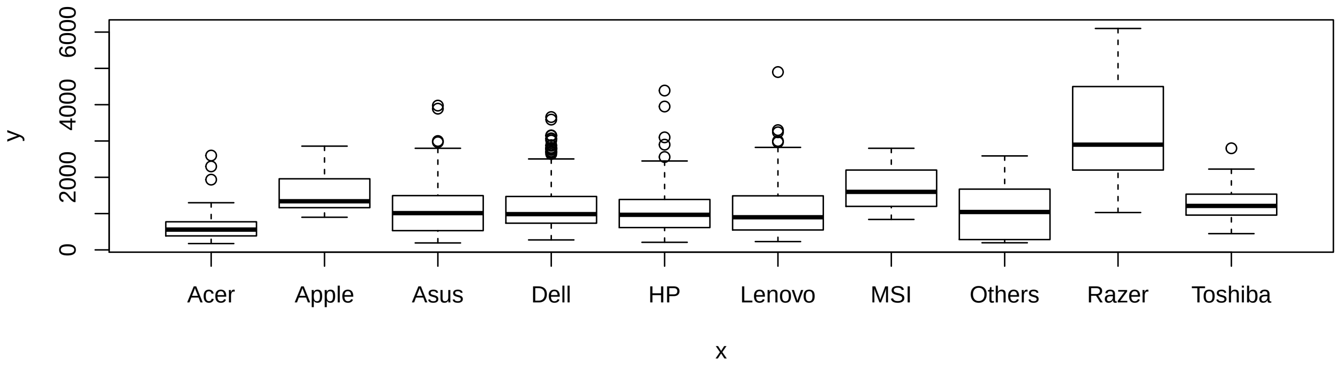
```
nums <- sapply(data, is.numeric)
var_numeric <- data[,nums]
head(var_numeric)
```

```
##   X Inches Ram Weight   Price Frequenza   Pixel MemoriaSSD TotalMemory
## 1 1   13.3   8   1.37 1339.69       2.3 4096000        128         128
## 2 2   13.3   8   1.34  898.94       1.8 1296000          0         128
## 3 3   15.6   8   1.86  575.00       2.5 2073600        256         256
## 4 4   15.4  16   1.83 2537.45       2.7 5184000        512         512
## 5 5   13.3   8   1.37 1803.60       3.1 4096000        256         256
## 6 6   15.6   4   2.10  400.00       3.0 1049088          0         500
```

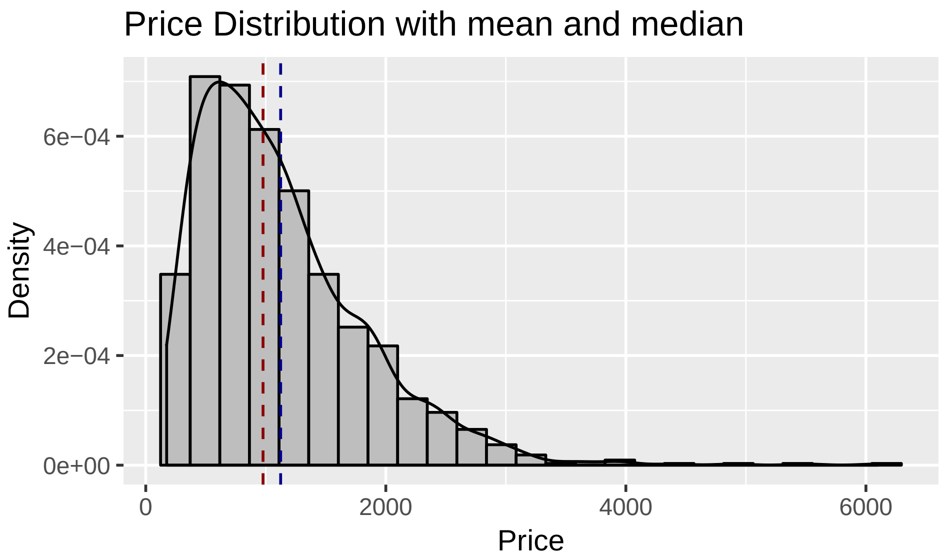
```
sapply(data, function(x)(sum(is.na(x)))) # Non ci sono missing data!
```

```
##                  X            Company            Product 
##                  0                  0                  0 
##           TypeName             Inches   ScreenResolution 
##                  0                  0                  0 
##                Cpu                Ram             Memory 
##                  0                  0                  0 
##                Gpu              OpSys             Weight 
##                  0                  0                  0 
##              Price          Frequenza        Risoluzione 
##                  0                  0                  0 
##              Pixel         GpuCompany        MemoriaSSD 
##                  0                  0                  0 
##     SolidStateDisk        TotalMemory      dedicated_GPU 
##                  0                  0                  0 
## Aggregated_Company 
##                  0
```

```
plot(data$Aggregated_Company,data$Price)
```

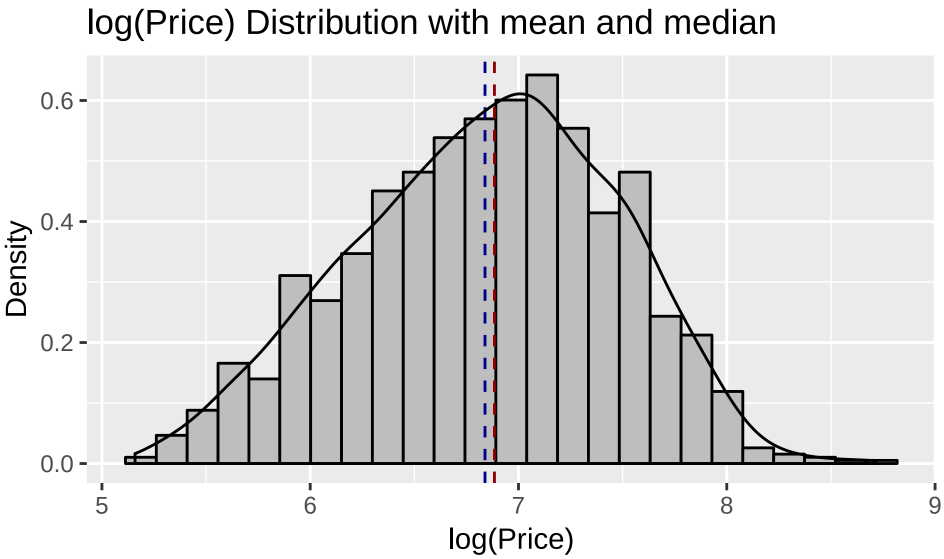


```
library(ggplot2)
ggplot(data,aes(x = Price)) + geom_histogram(aes(y =..density..), bins= 25, fill = "grey",color ="black") +
        geom_vline(xintercept = quantile(data$Price, 0.50), color = "dark red", lty = 2) +geom_vline(xintercept = mean
        labs(x = "Price", y ="Density") + ggtitle("Price Distribution with mean and median") +geom_density()
```



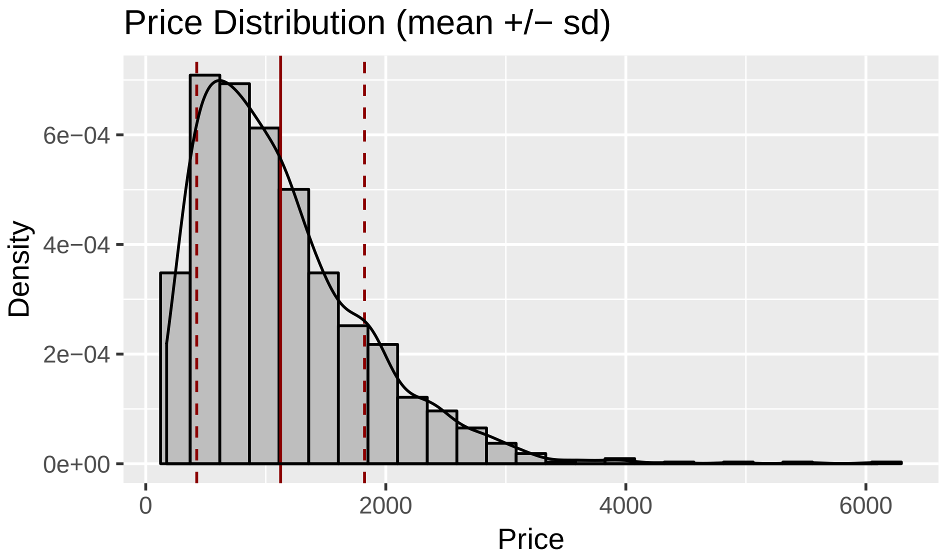
Quite skewed to the right, mean > median: we could try to apply a correction like Log(Y)

```
ggplot(data,aes(x = log(Price))) + geom_histogram(aes(y =..density..),bins= 25, fill = "grey", color ="black") +
  geom_vline(xintercept = quantile(log(data$Price), 0.50), color = "dark red", lty = 2) + geom_vline(xintercept = mean
  labs(x = "log(Price)", y ="Density") +ggtitle("log(Price) Distribution with mean and median")+  geom_density()
```

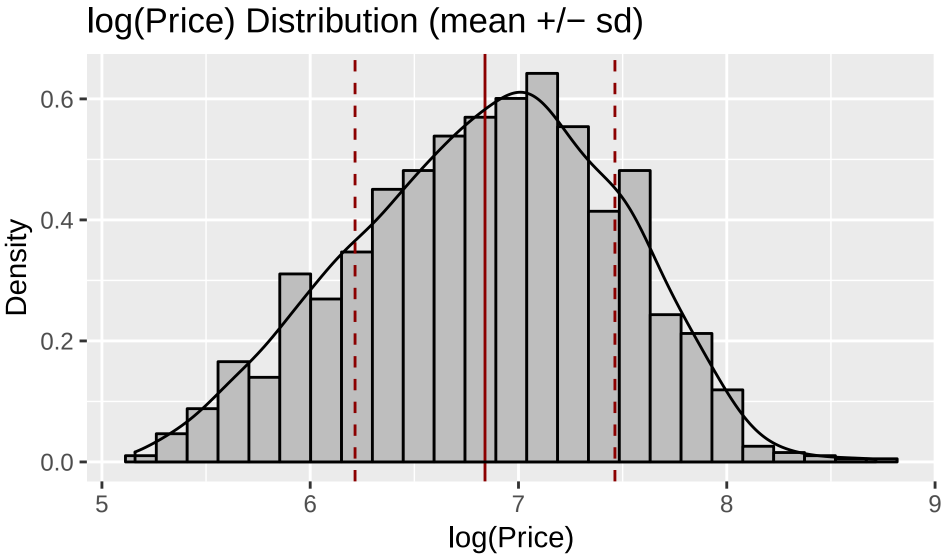


Now the distribution is looking a bit better (as regards normality)

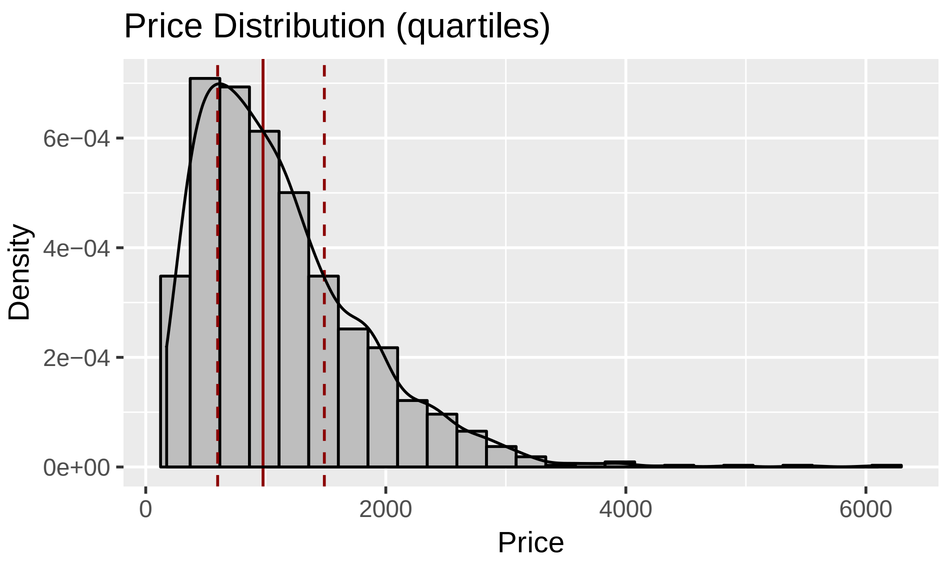
```
ggplot(data,aes(x = Price)) + geom_histogram(aes(y =..density..), bins= 25, fill = "grey", color ="black") +
  geom_vline(xintercept = mean(data$Price), color = "dark red") + geom_vline(xintercept = mean(data$Price) + sd(data$P
  geom_vline(xintercept = mean(data$Price) - sd(data$Price), color = "dark red", lty = 2) +labs(x = "Price", y ="Densi
```



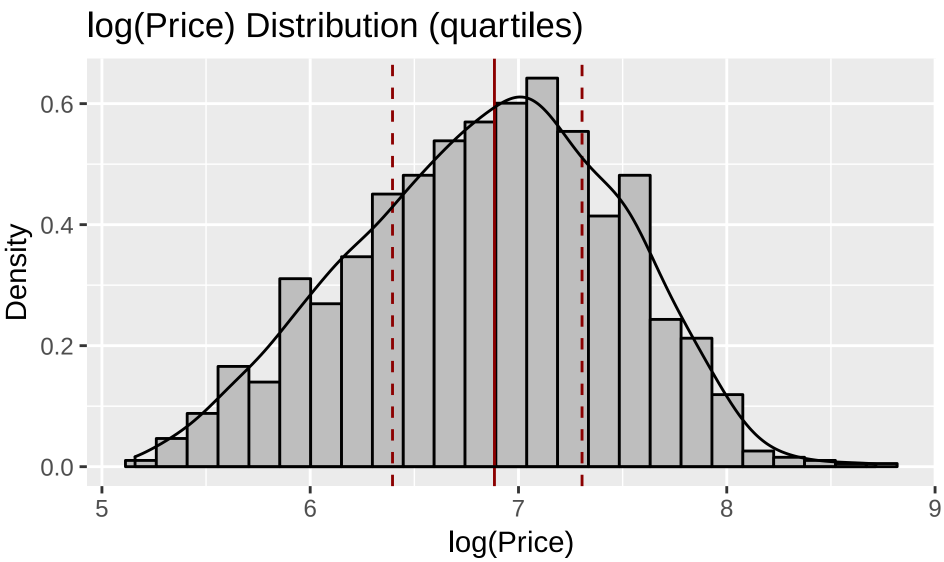
```
ggplot(data,aes(x = log(Price))) +geom_histogram(aes(y =..density..), bins= 25,fill = "grey",color ="black") +
  geom_vline(xintercept = mean(log(data$Price)), color = "dark red") + geom_vline(xintercept = mean(log(data$Price)) +
  geom_vline(xintercept = mean(log(data$Price)) - sd(log(data$Price)), color = "dark red", lty = 2) + labs(x = "log(Pr
```



```
ggplot(data,aes(x = Price)) + geom_histogram(aes(y =..density..), bins= 25, fill = "grey", color ="black") +
  geom_vline(xintercept = quantile(data$Price, 0.25), color = "dark red",lty = 2) + geom_vline(xintercept = quantile(d
  geom_vline(xintercept = quantile(data$Price, 0.75), color = "dark red", lty = 2) + labs(x = "Price", y ="Density") +
```

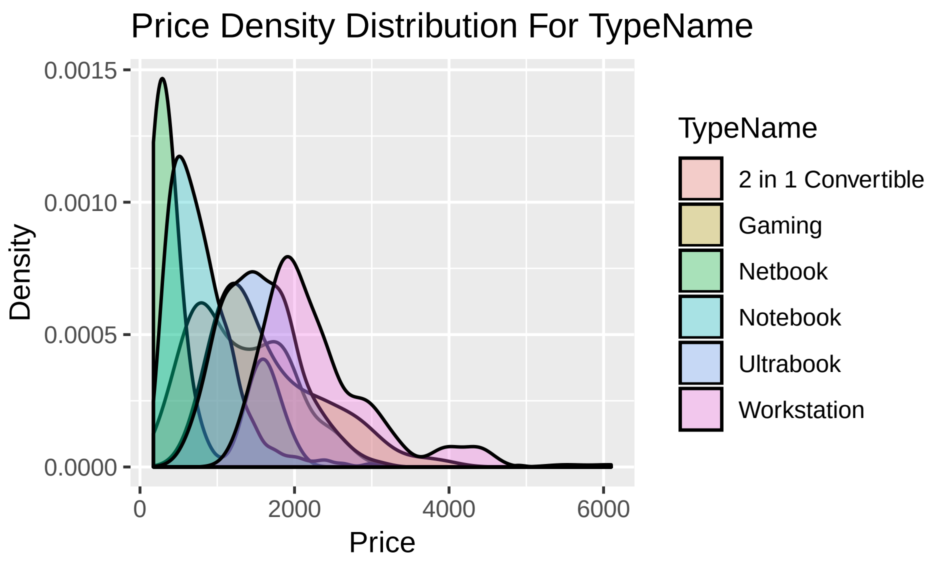


```
ggplot(data,aes(x = log(Price))) + geom_histogram(aes(y =..density..), bins= 25, fill = "grey", color ="black") +
  geom_vline(xintercept = quantile(log(data$Price), 0.25), color = "dark red",lty = 2) + geom_vline(xintercept = quant
  geom_vline(xintercept = quantile(log(data$Price), 0.75), color = "dark red", lty = 2) + labs(x = "log(Price)", y ="D
```

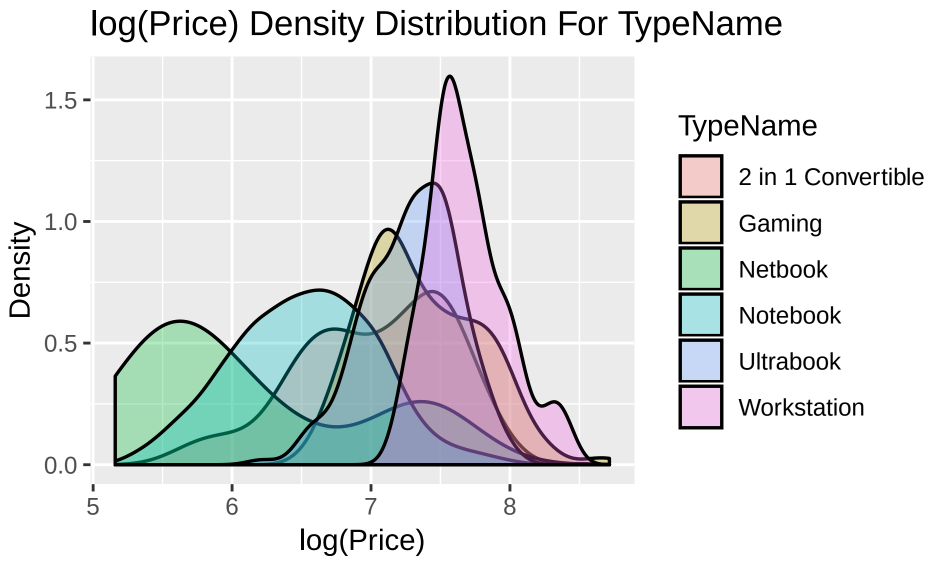


Descrittive variabile dipendente price

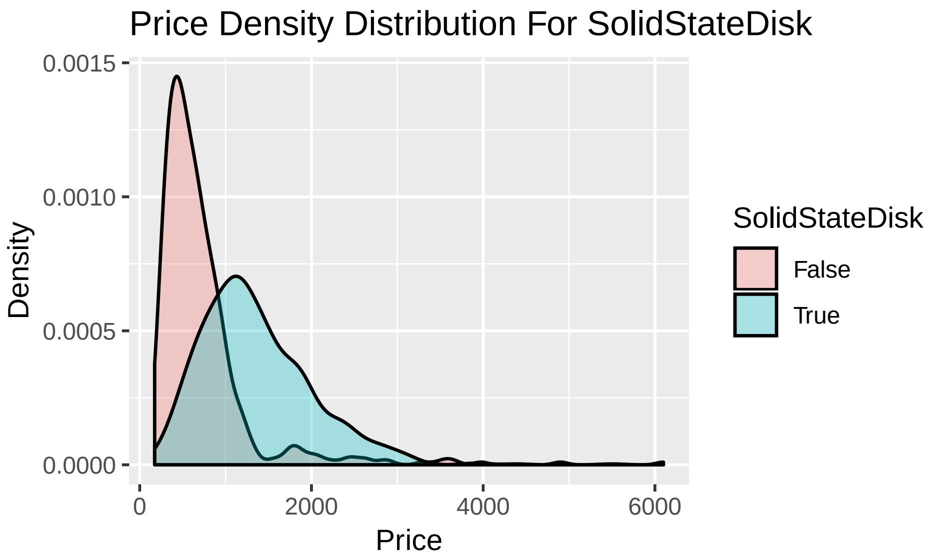
```
ggplot(data, aes(x = Price, fill = TypeName)) + geom_density(size = 0.6, alpha = .3) + labs(x = "Price", y ="Density",
```



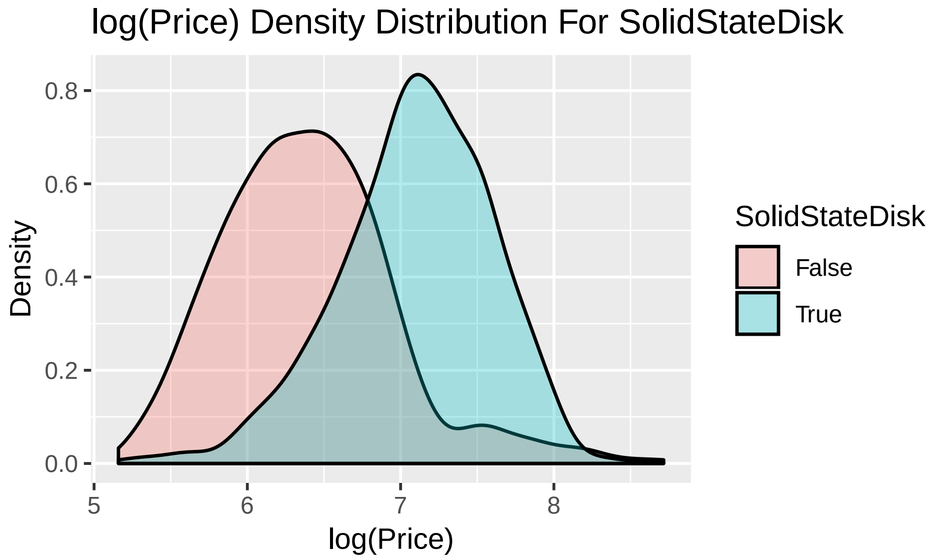
```
ggplot(data, aes(x = log(Price), fill = TypeName)) + geom_density(size = 0.6, alpha = .3) +labs(x = "log(Price)", y ="
```



```
ggplot(data, aes(x = Price, fill = SolidStateDisk)) + geom_density(size = 0.6, alpha = .3) +labs(x = "Price", y ="Dens
```



```
ggplot(data, aes(x = log(Price), fill = SolidStateDisk)) + geom_density(size = 0.6, alpha = .3) + labs(x = "log(Price)
```



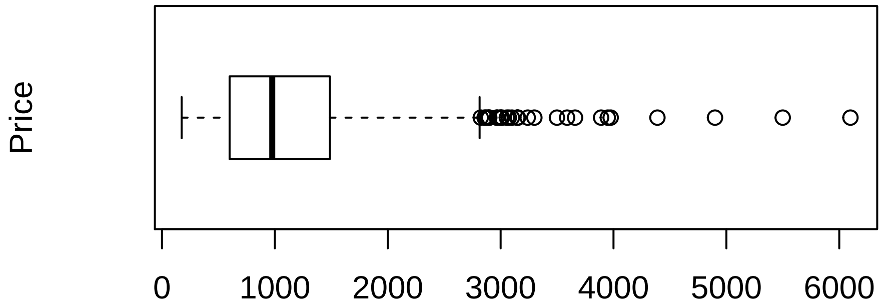
```
library(psych)
describe(data$Price)
```

```
##    vars    n    mean     sd median trimmed    mad min  max range skew
## X1    1 1303 1123.69 699.01    977 1038.47 619.73 174 6099  5925 1.52
##    kurtosis    se
## X1     4.34 19.36
```

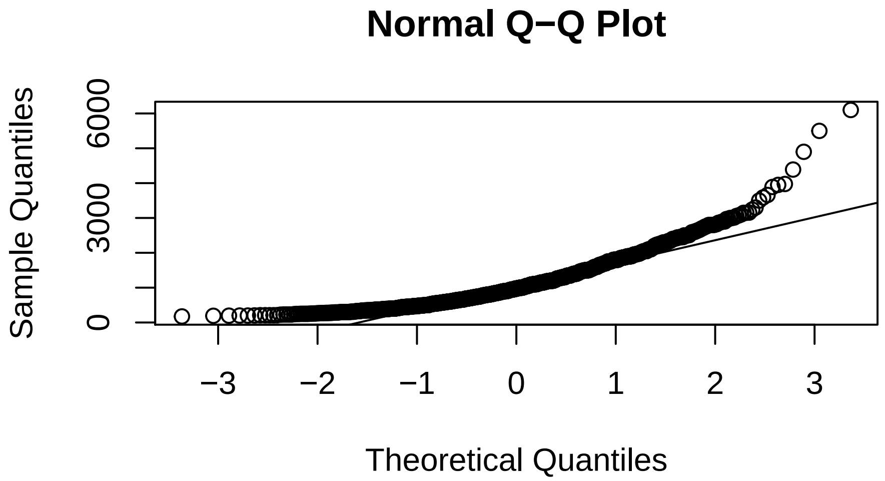
```
describe(log(data$Price))
```

```
##    vars    n mean   sd median trimmed  mad  min  max range  skew kurtosis
## X1    1 1303 6.84 0.62   6.88    6.85 0.65 5.16 8.72  3.56 -0.17    -0.47
##      se
## X1 0.02
```

```
library(nortest) # test per ipotesi di normalità
boxplot(data$Price, horizontal = T, ylab = c("Price") )
```



```
qqnorm(data$Price);qqline(data$Price)
```



```
shapiro.test(data$Price)
```

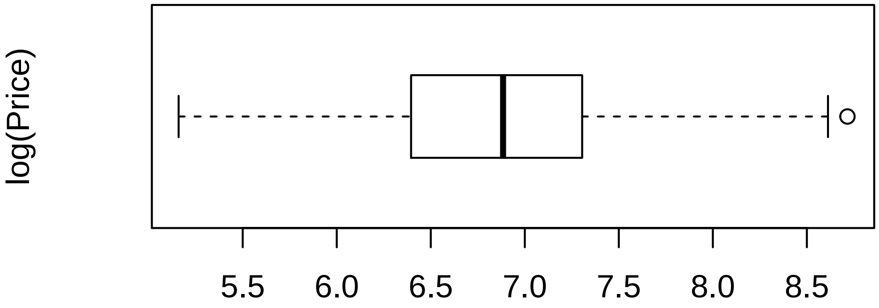
```
##
##  Shapiro-Wilk normality test
##
## data:  data$Price
## W = 0.89382, p-value < 2.2e-16
```

```
ad.test(data$Price)
```

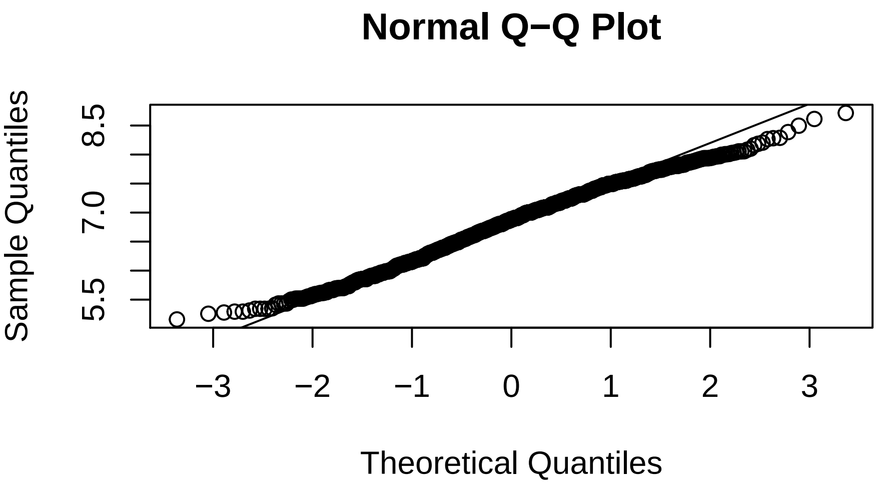
```
##
##  Anderson-Darling normality test
##
## data:  data$Price
## A = 28.319, p-value < 2.2e-16
```

Trying with the log correction:

```
# Correzione NORMALITA'
library(nortest)
boxplot(log(data$Price), ylab="log(Price)", horizontal = T)
```



```
qqnorm(log(data$Price));qqline(log(data$Price))
```



```
shapiro.test(log(data$Price)) #better than before, but still not normal according to shapiro
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log(data$Price)
## W = 0.99252, p-value = 3.628e-06
```

```
ad.test(log(data$Price))
```

```
##
##  Anderson-Darling normality test
##
## data:  log(data$Price)
## A = 2.5942, p-value = 1.515e-06
```

Test on a mean (justify H0) on Y and confidence limits.

T-test

```
# One sample
ref <- mean(data$Price) #FIXME: trova ref
Apple<-data$Price[data$Company=="Apple"]
t.test(Apple,mu=ref,alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  Apple
## t = 3.5944, df = 20, p-value = 0.000906
## alternative hypothesis: true mean is greater than 1123.687
## 95 percent confidence interval:
##  1352.823      Inf
## sample estimates:
## mean of x
##  1564.199
```

```
# Wilcoxon Signed Rank Test
wilcox.test(Apple, mu=ref, conf.int = TRUE)
```

```
##
##  Wilcoxon signed rank test
##
## data:  Apple
## V = 206, p-value = 0.0008516
## alternative hypothesis: true location is not equal to 1123.687
## 95 percent confidence interval:
##  1234.50 1829.26
## sample estimates:
## (pseudo)median
##       1514.275
```

```
library(EnvStats)
varTest(sample(data$Price), sigma.squared = (sd(data$Price)*sd(data$Price)))
```

```
##
##  Chi-Squared Test on Variance
##
## data:  sample(data$Price)
## Chi-Squared = 1302, df = 1302, p-value = 0.9896
## alternative hypothesis: true variance is not equal to 488613.6
## 95 percent confidence interval:
##  453149.5 528432.0
## sample estimates:
## variance
## 488613.6
```

Test two means, two variances (Y vs X) .

```
#Two sample
Other <-data$Price[data$Company!="Apple"]
wilcox.test(Apple, Other, alternative = "g")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Apple and Other
## W = 19689, p-value = 0.0001358
## alternative hypothesis: true location shift is greater than 0
```

```
# F test sulla varianza
var.test(Apple, Other, alternative = "two.sided")
```

```
##
##  F test to compare two variances
##
## data:  Apple and Other
## F = 0.64574, num df = 20, denom df = 1281, p-value = 0.2401
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3755878 1.3509884
## sample estimates:
## ratio of variances
##          0.6457382
```

Association/chi square among some couples of categorical Xj

Variabili qualitative: tabella di contingenza e chi quadro

```
b<-data
b.table<-table(b$SolidStateDisk,b$TypeName)
b.table
```

```
##
##         2 in 1 Convertible Gaming Netbook Notebook Ultrabook Workstation
##   False                 29     32      13      376        19           7
##   True                   92    173      12      351       177          22
```

```r
prop.table(b.table,2)
```

```
##
##         2 in 1 Convertible     Gaming    Netbook   Notebook  Ultrabook
##   False         0.23966942 0.15609756 0.52000000 0.51719395 0.09693878
##   True          0.76033058 0.84390244 0.48000000 0.48280605 0.90306122
##
##         Workstation
##   False  0.24137931
##   True   0.75862069
```

```r
# chi square test

chisq.test(b.table)
```

```
##
##  Pearson's Chi-squared test
##
## data:  b.table
## X-squared = 184.66, df = 5, p-value < 2.2e-16
```

```r
chi=chisq.test(b.table)
chi_norm=chi$statistic/(nrow(b)*min(nrow(b.table)-1,ncol(b.table)-1))
chi_norm
```

```
## X-squared
## 0.1417156
```

```r
summary(b.table)
```

```
## Number of cases in table: 1303
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 184.66, df = 5, p-value = 5.42e-38
```

```r
#Proviamo SolidStateDisk vs dedicated_GPU FIXME: check
b<-data
b.table<-table(b$SolidStateDisk,b$dedicated_GPU)
b.table
```

```
##
##         False True
##   False   285  191
##   True    438  389
```

```r
prop.table(b.table,2)
```

```
##
##             False      True
##   False 0.3941909 0.3293103
##   True  0.6058091 0.6706897
```

```r
# chi square test

chisq.test(b.table)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  b.table
## X-squared = 5.5664, df = 1, p-value = 0.01831
```

```r
chi=chisq.test(b.table)
chi_norm=chi$statistic/(nrow(b)*min(nrow(b.table)-1,ncol(b.table)-1))
chi_norm
```

```
##  X-squared
## 0.00427199
```

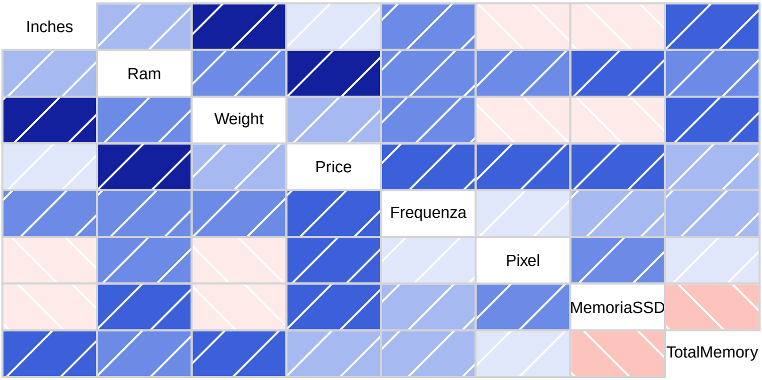
```
summary(b.table)
```

```
## Number of cases in table: 1303
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 5.843, df = 1, p-value = 0.01564
```

Correlazione per variabili quantitative

```
# seleziona solo variabili quantitative
nums <- sapply(data, is.numeric)
var_numeric <- data[,nums]
var_numeric$X=NULL
```

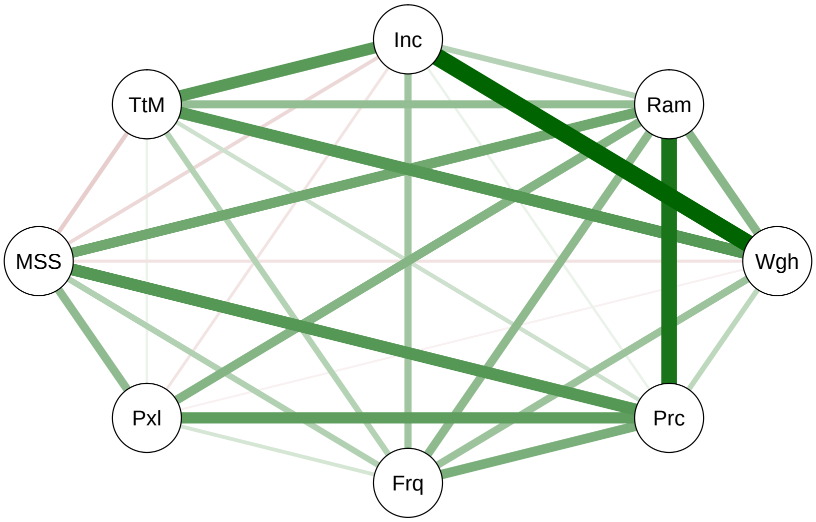
```
# Test di correlazione. (Spearman's o Kendall tau)
#if(!require(corrgram)) install.packages("corrgram")
library(corrgram)
corrgram(var_numeric)
```



```
# Correlazione come grafo
library(qgraph)
detcor=cor(as.matrix(var_numeric), method="pearson")
round(detcor, 2)
```

```
##             Inches  Ram Weight Price Frequenza Pixel MemoriaSSD
## Inches        1.00 0.24   0.83  0.07      0.31 -0.09      -0.13
## Ram           0.24 1.00   0.38  0.74      0.37  0.40       0.46
## Weight        0.83 0.38   1.00  0.21      0.32 -0.04      -0.10
## Price         0.07 0.74   0.21  1.00      0.43  0.52       0.55
## Frequenza     0.31 0.37   0.32  0.43      1.00  0.14       0.25
## Pixel        -0.09 0.40  -0.04  0.52      0.14  1.00       0.36
## MemoriaSSD   -0.13 0.46  -0.10  0.55      0.25  0.36       1.00
## TotalMemory   0.54 0.35   0.55  0.16      0.24  0.06      -0.16
##             TotalMemory
## Inches             0.54
## Ram                0.35
## Weight             0.55
## Price              0.16
## Frequenza          0.24
## Pixel              0.06
## MemoriaSSD        -0.16
## TotalMemory        1.00
```

```
# plot corr matrix: green positive red negative
qgraph(detcor, shape="circle", posCol="darkgreen", negCol="darkred")
```

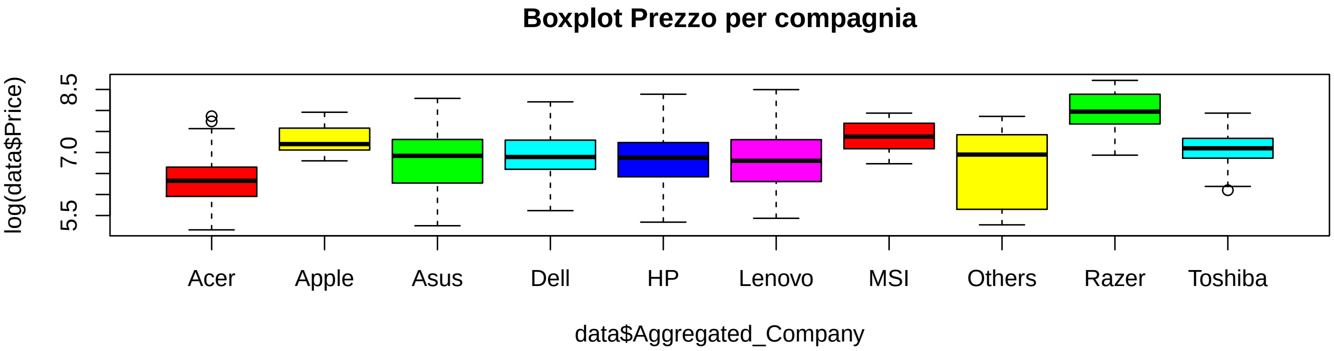


```r
cor.test(var_numeric$Inches, var_numeric$Weight)
```

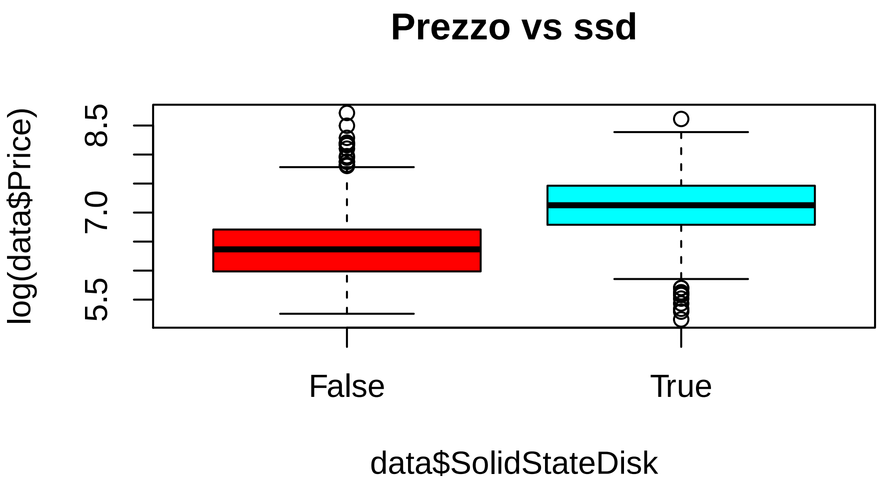
```
##
##  Pearson's product-moment correlation
##
## data:  var_numeric$Inches and var_numeric$Weight
## t = 53.187, df = 1301, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8097181 0.8440031
## sample estimates:
##       cor
## 0.8276311
```

Boxplot di confronto (pre-anova)

```r
boxplot(log(data$Price)~data$Aggregated_Company, main="Boxplot Prezzo per compagnia", col= rainbow(6), horizontal = F)
```



```r
boxplot(log(data$Price)~data$SolidStateDisk, main="Prezzo vs ssd", col= rainbow(2), horizontal = F)
```



Anova one way Y = Xj, for a categorical X

```
lmA = lm(log(Price) ~ SolidStateDisk, data=data)
summary(lmA)
```

```
##
## Call:
## lm(formula = log(Price) ~ SolidStateDisk, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9365 -0.3314  0.0015  0.3422  2.3221
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.39376    0.02404  265.95   <2e-16 ***
## SolidStateDiskTrue  0.70179    0.03018   23.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5245 on 1301 degrees of freedom
## Multiple R-squared:  0.2936, Adjusted R-squared:  0.2931
## F-statistic: 540.8 on 1 and 1301 DF,  p-value: < 2.2e-16
```

```
drop1(lmA, test = 'F')
```

```
## Single term deletions
##
## Model:
## log(Price) ~ SolidStateDisk
##                Df Sum of Sq    RSS     AIC F value    Pr(>F)
## <none>                      357.92 -1679.6
## SolidStateDisk  1    148.79 506.71 -1228.7  540.84 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lmA)
```

```
## Analysis of Variance Table
##
## Response: log(Price)
##                  Df Sum Sq Mean Sq F value    Pr(>F)
## SolidStateDisk    1 148.79 148.791  540.84 < 2.2e-16 ***
## Residuals      1301 357.92   0.275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

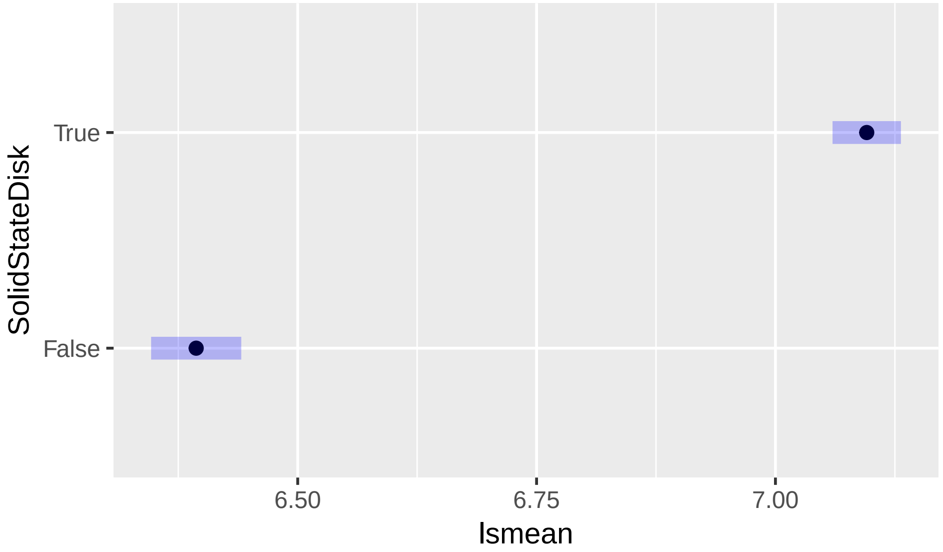
```
library(lsmeans)
ls_SolidStateDisk = lsmeans(lmA,pairwise ~ SolidStateDisk,adjust = 'tukey')
ls_SolidStateDisk$contrasts
```

```
##  contrast     estimate     SE   df t.ratio p.value
##  False - True   -0.702 0.0302 1301 -23.256 <.0001
##
## Results are given on the log (not the response) scale.
```

```
ls_SolidStateDisk$lsmeans
```

```
##  SolidStateDisk lsmean      SE   df lower.CL upper.CL
##  False           6.394 0.02404 1301    6.347    6.441
##  True            7.096 0.01824 1301    7.060    7.131
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

```
plot(ls_SolidStateDisk$lsmeans, alpha = .05)
```



```
#FIXME: tolto Price vs Aggregated_company, non ricordo se tenere o nop
lmC = lm(Price ~ TypeName, data=data)
summary(lmC)
```

```
##
## Call:
## lm(formula = Price ~ TypeName, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1049.2  -381.7   -98.1   267.6  4367.6
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1282.40      50.01  25.642  < 2e-16 ***
## TypeNameGaming         448.98      63.07   7.119 1.79e-12 ***
## TypeNameNetbook       -646.17     120.86  -5.347 1.06e-07 ***
## TypeNameNotebook      -500.32      54.01  -9.263  < 2e-16 ***
## TypeNameUltrabook      265.83      63.60   4.180 3.12e-05 ***
## TypeNameWorkstation    997.96     113.74   8.774  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 550.1 on 1297 degrees of freedom
## Multiple R-squared:  0.383,  Adjusted R-squared:  0.3806
## F-statistic:   161 on 5 and 1297 DF,  p-value: < 2.2e-16
```

```r
drop1(lmC, test = 'F')
```

```
## Single term deletions
##
## Model:
## Price ~ TypeName
##          Df Sum of Sq       RSS   AIC F value    Pr(>F)
## <none>                392518380 16450
## TypeName  5 243656581 636174961 17069  161.02 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lmC)
```

```
## Analysis of Variance Table
##
## Response: Price
##             Df    Sum Sq  Mean Sq F value    Pr(>F)
## TypeName     5 243656581 48731316  161.02 < 2.2e-16 ***
## Residuals 1297 392518380   302636
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

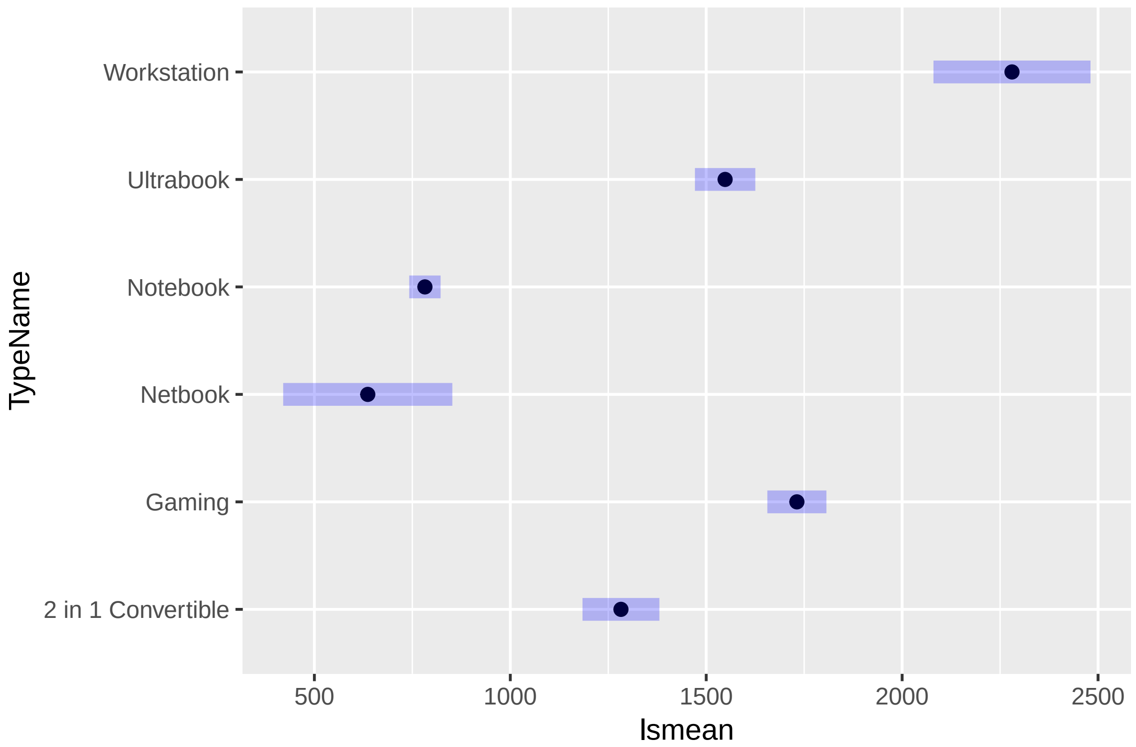
```r
ls_TypeName = lsmeans(lmC,pairwise ~ TypeName,adjust = 'tukey')
ls_TypeName$contrasts
```

```
##  contrast                         estimate    SE   df t.ratio p.value
##  2 in 1 Convertible - Gaming          -449  63.1 1297  -7.119 <.0001
##  2 in 1 Convertible - Netbook          646 120.9 1297   5.347 <.0001
##  2 in 1 Convertible - Notebook         500  54.0 1297   9.263 <.0001
##  2 in 1 Convertible - Ultrabook       -266  63.6 1297  -4.180 0.0004
##  2 in 1 Convertible - Workstation     -998 113.7 1297  -8.774 <.0001
##  Gaming - Netbook                     1095 116.5 1297   9.397 <.0001
##  Gaming - Notebook                     949  43.5 1297  21.821 <.0001
##  Gaming - Ultrabook                    183  55.0 1297   3.333 0.0114
##  Gaming - Workstation                 -549 109.1 1297  -5.030 <.0001
##  Netbook - Notebook                   -146 111.9 1297  -1.303 0.7833
##  Netbook - Ultrabook                  -912 116.8 1297  -7.806 <.0001
##  Netbook - Workstation               -1644 150.1 1297 -10.951 <.0001
##  Notebook - Ultrabook                 -766  44.3 1297 -17.304 <.0001
##  Notebook - Workstation              -1498 104.2 1297 -14.383 <.0001
##  Ultrabook - Workstation              -732 109.5 1297  -6.689 <.0001
##
## P value adjustment: tukey method for comparing a family of 6 estimates
```

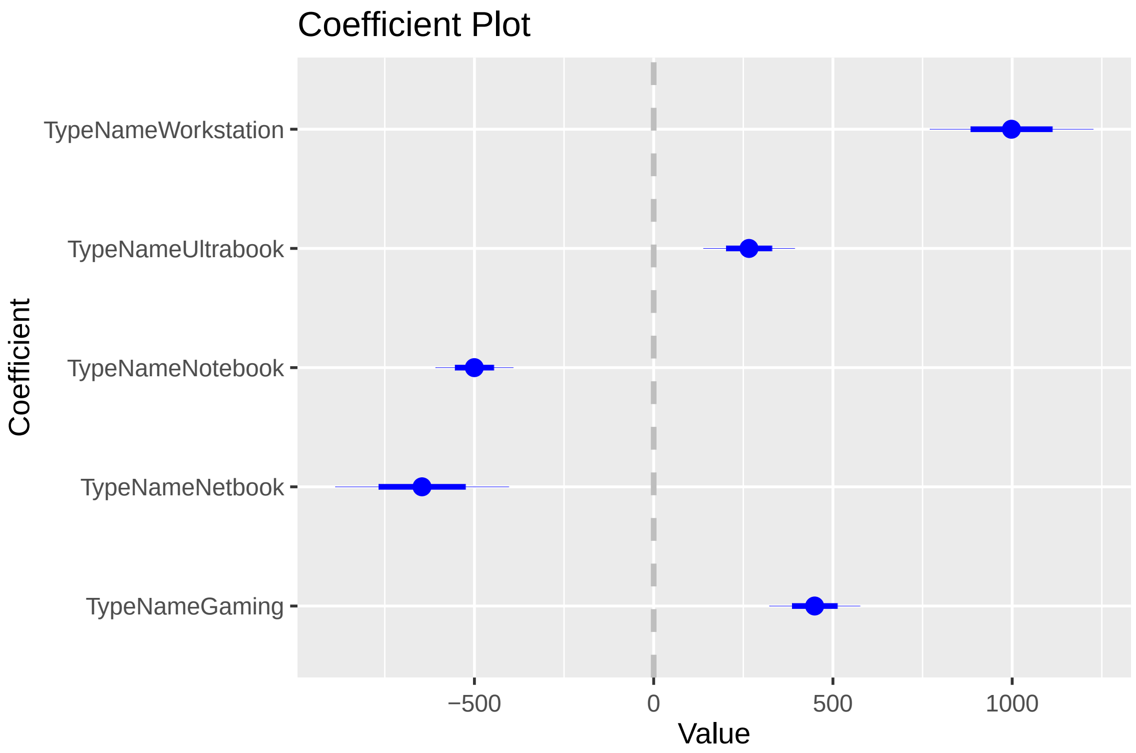
```r
ls_TypeName$lsmeans
```

```
##  TypeName           lsmean    SE   df lower.CL upper.CL
##  2 in 1 Convertible   1282  50.0 1297     1184     1381
##  Gaming               1731  38.4 1297     1656     1807
##  Netbook               636 110.0 1297      420      852
##  Notebook              782  20.4 1297      742      822
##  Ultrabook            1548  39.3 1297     1471     1625
##  Workstation          2280 102.2 1297     2080     2481
##
## Confidence level used: 0.95
```

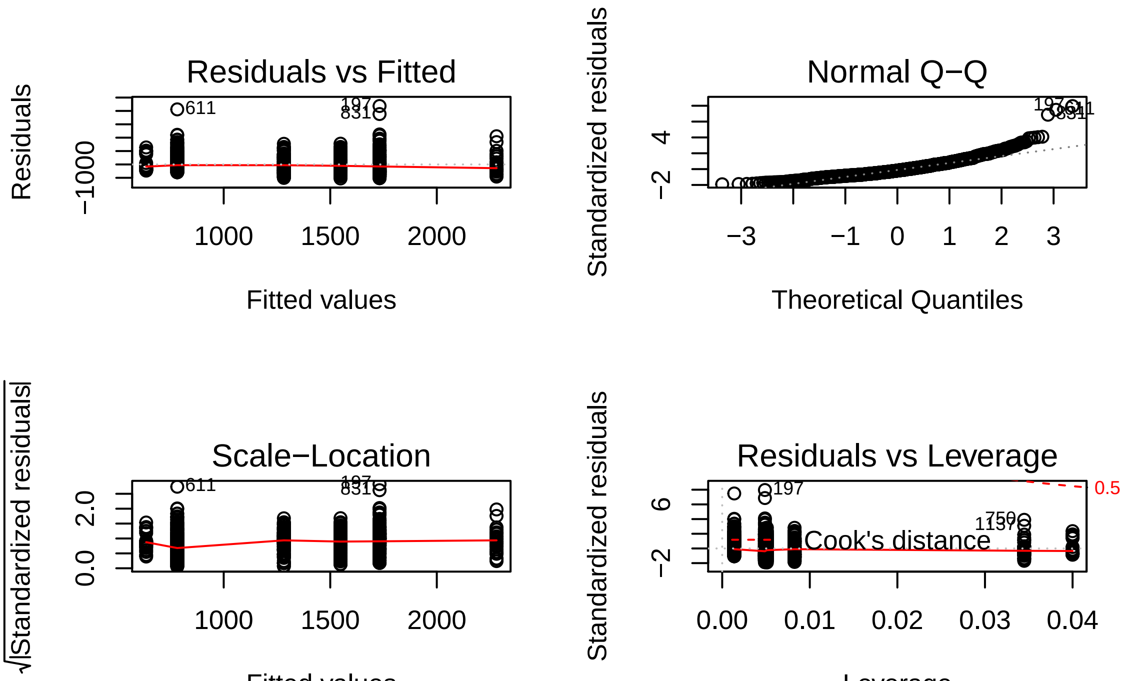
```r
plot(ls_TypeName$lsmeans, alpha = .05)
```



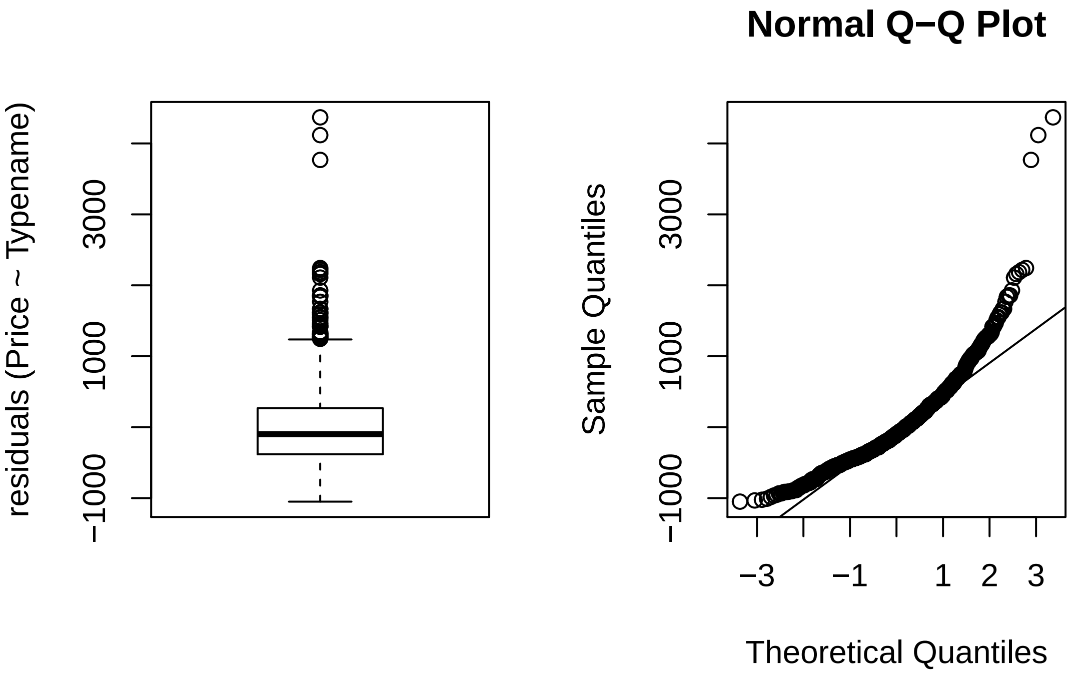
```
library(coefplot)
#library(forestmodel)
coefplot(lmC, intercept = FALSE)
```



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



```
#(not) normal distribution of residuals
par(mfrow=c(1,2))
boxplot(lmC$residuals, ylab="residuals (Price ~ Typename)")
qqnorm(lmC$residuals);qqline(lmC$residuals)
```



```
ad.test(lmC$residuals)
```

```
##
##  Anderson-Darling normality test
##
## data:  lmC$residuals
## A = 22.667, p-value < 2.2e-16
```

```
shapiro.test(lmC$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lmC$residuals
## W = 0.89641, p-value < 2.2e-16
```

```
#let's try again with the log correction
lmC_log = lm(log(Price) ~ TypeName, data=data)
summary(lmC_log)#R^2 increases
```

```
##
## Call:
## lm(formula = log(Price) ~ TypeName, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40971 -0.33589  0.00698  0.33215  1.96853
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           7.02648    0.04379 160.456  < 2e-16 ***
## TypeNameGaming        0.33865    0.05522   6.133 1.15e-09 ***
## TypeNameNetbook      -0.91149    0.10583  -8.613  < 2e-16 ***
## TypeNameNotebook     -0.49823    0.04729 -10.534  < 2e-16 ***
## TypeNameUltrabook     0.26648    0.05569   4.785 1.91e-06 ***
## TypeNameWorkstation  0.66479    0.09959   6.675 3.65e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4817 on 1297 degrees of freedom
## Multiple R-squared:  0.4061, Adjusted R-squared:  0.4038
## F-statistic: 177.4 on 5 and 1297 DF,  p-value: < 2.2e-16
```

```
drop1(lmC_log, test = 'F')
```

```
## Single term deletions
##
## Model:
## log(Price) ~ TypeName
##          Df Sum of Sq    RSS     AIC F value    Pr(>F)
## <none>                300.95 -1897.5
## TypeName  5    205.76 506.71 -1228.7  177.36 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lmC_log)
```

```
## Analysis of Variance Table
##
## Response: log(Price)
##             Df Sum Sq Mean Sq F value    Pr(>F)
## TypeName     5 205.76  41.152  177.36 < 2.2e-16 ***
## Residuals 1297 300.95   0.232
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

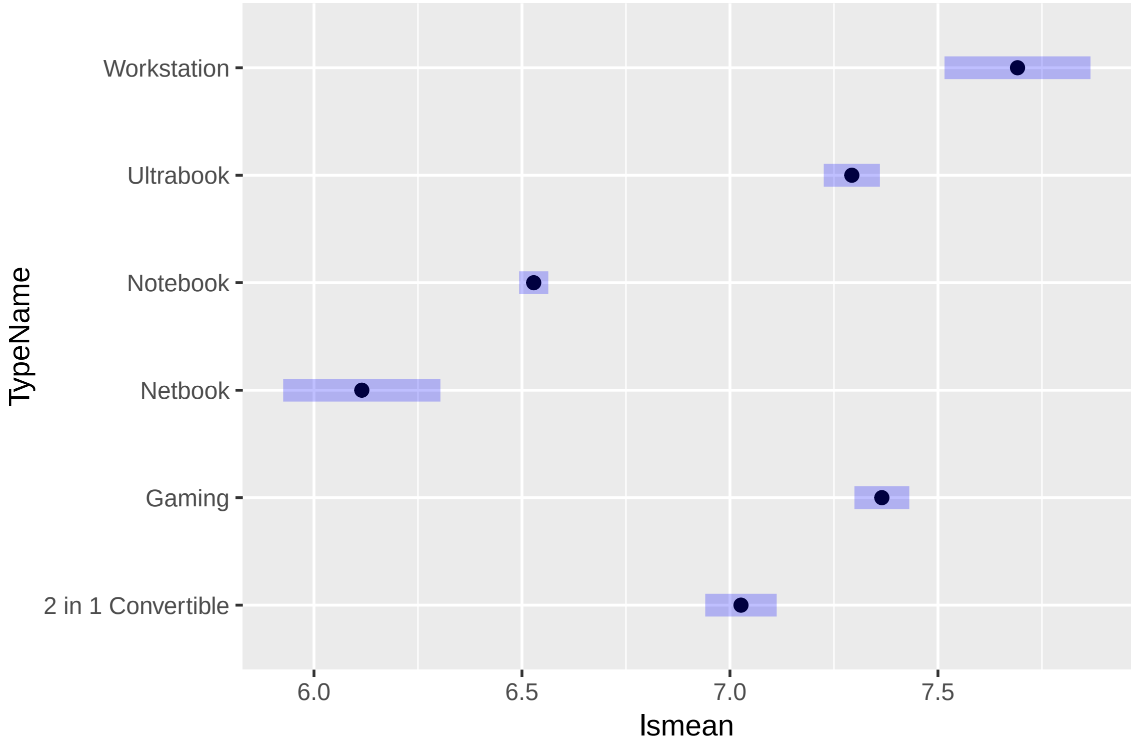
```
ls_TypeName_log = lsmeans(lmC_log,pairwise ~ TypeName,adjust = 'tukey')
ls_TypeName_log$contrasts
```

```
##  contrast                         estimate     SE   df t.ratio p.value
##  2 in 1 Convertible - Gaming       -0.3387 0.0552 1297  -6.133 <.0001
##  2 in 1 Convertible - Netbook       0.9115 0.1058 1297   8.613 <.0001
##  2 in 1 Convertible - Notebook      0.4982 0.0473 1297  10.534 <.0001
##  2 in 1 Convertible - Ultrabook    -0.2665 0.0557 1297  -4.785 <.0001
##  2 in 1 Convertible - Workstation  -0.6648 0.0996 1297  -6.675 <.0001
##  Gaming - Netbook                   1.2501 0.1020 1297  12.251 <.0001
##  Gaming - Notebook                  0.8369 0.0381 1297  21.970 <.0001
##  Gaming - Ultrabook                 0.0722 0.0481 1297   1.500 0.6644
##  Gaming - Workstation              -0.3261 0.0956 1297  -3.413 0.0087
##  Netbook - Notebook                -0.4133 0.0980 1297  -4.218 0.0004
##  Netbook - Ultrabook               -1.1780 0.1023 1297 -11.515 <.0001
##  Netbook - Workstation             -1.5763 0.1315 1297 -11.990 <.0001
##  Notebook - Ultrabook              -0.7647 0.0388 1297 -19.725 <.0001
##  Notebook - Workstation            -1.1630 0.0912 1297 -12.750 <.0001
##  Ultrabook - Workstation           -0.3983 0.0958 1297  -4.156 0.0005
##
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 6 estimates
```

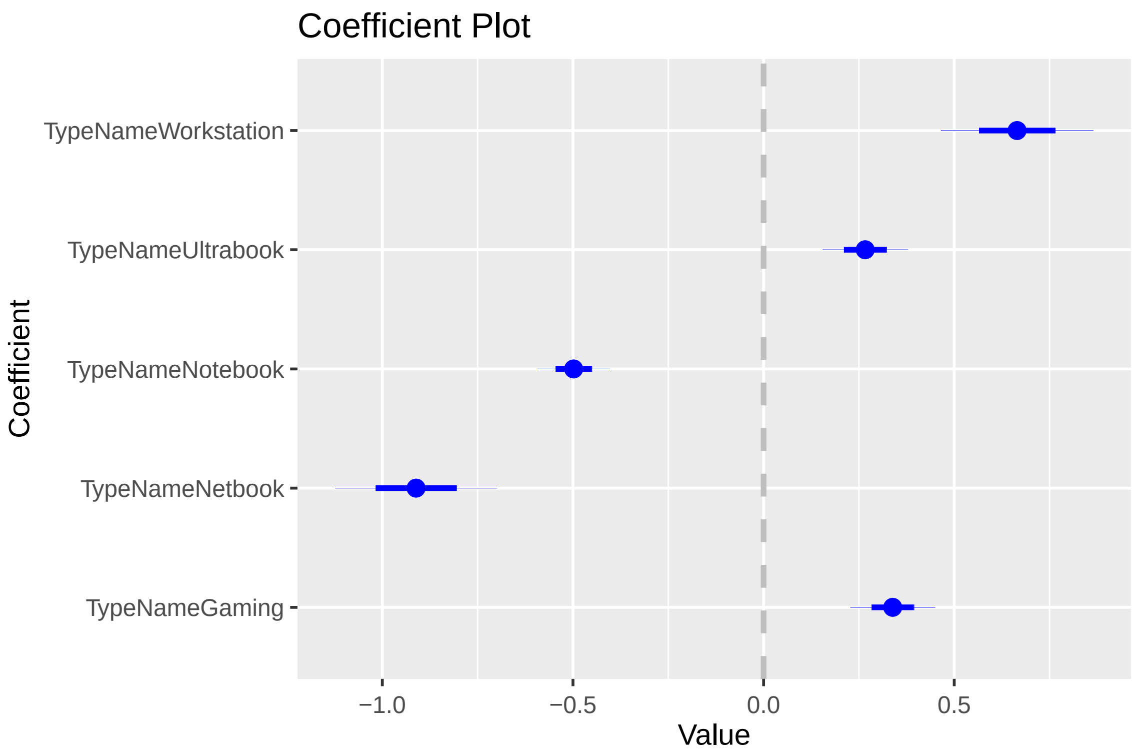
```
ls_TypeName_log$lsmeans
```

```
##  TypeName            lsmean     SE   df lower.CL upper.CL
##  2 in 1 Convertible    7.03 0.0438 1297     6.94     7.11
##  Gaming                7.37 0.0336 1297     7.30     7.43
##  Netbook               6.11 0.0963 1297     5.93     6.30
##  Notebook              6.53 0.0179 1297     6.49     6.56
##  Ultrabook             7.29 0.0344 1297     7.23     7.36
##  Workstation           7.69 0.0894 1297     7.52     7.87
##
## Results are given on the log (not the response) scale.
## Confidence level used: 0.95
```

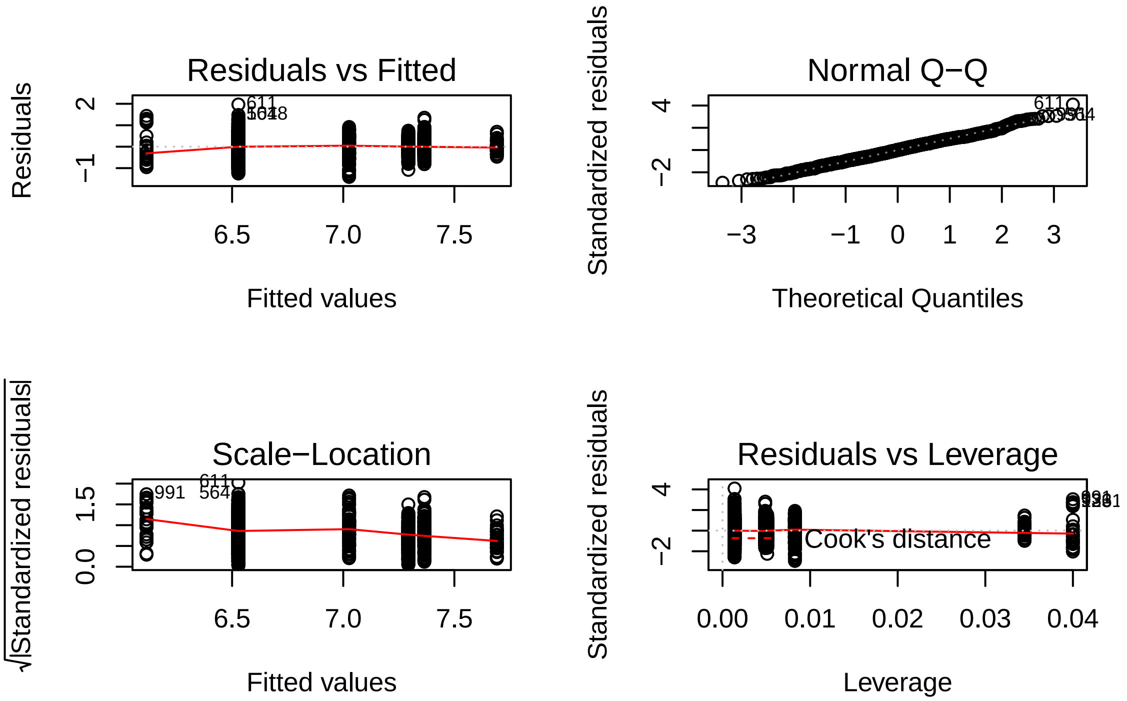
```
plot(ls_TypeName_log$lsmeans, alpha = .05)
```



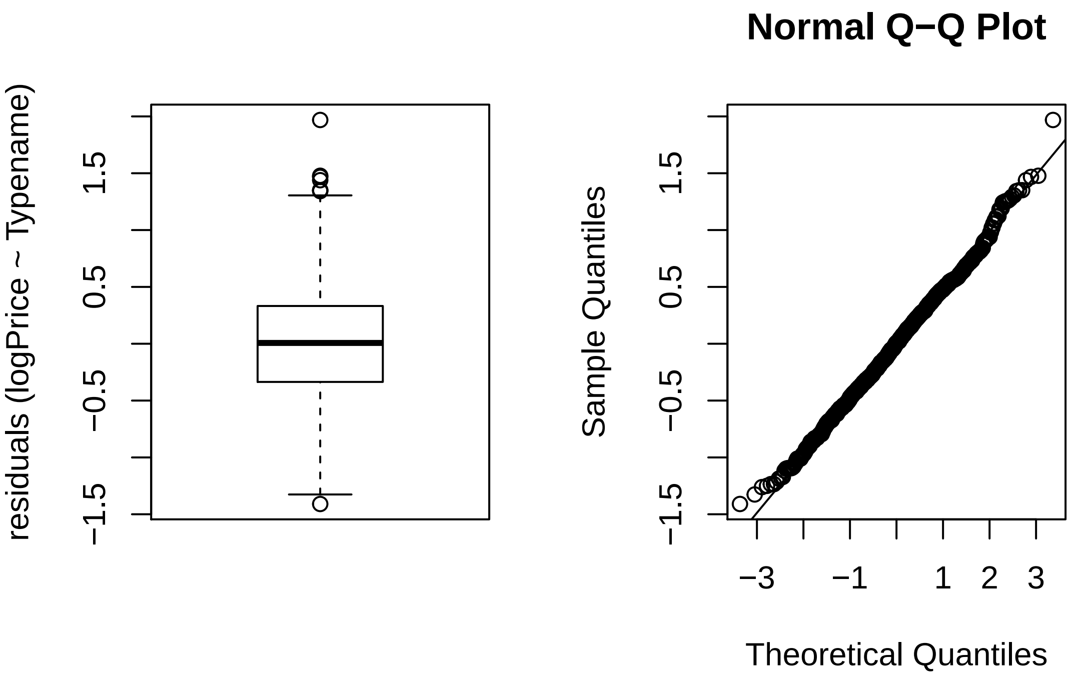
```
coefplot(lmC_log, intercept = FALSE)
```



```r
par(mfrow = c(2,2))
plot(lmC_log)
```



```r
#(not) normal distribution of residuals
par(mfrow=c(1,2))
boxplot(lmC_log$residuals, ylab="residuals (logPrice ~ Typename)")
qqnorm(lmC_log$residuals);qqline(lmC_log$residuals)
```



```
ad.test(lmC_log$residuals) #normal now!
```

```
##
##  Anderson-Darling normality test
##
## data:  lmC_log$residuals
## A = 0.51757, p-value = 0.1886
```

```
shapiro.test(lmC_log$residuals) #borderline now!
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lmC_log$residuals
## W = 0.99764, p-value = 0.05462
```

Anova two way Y = Xj Xk for some categorical X

A due vie

```
lmC = lm(log(Price) ~ Aggregated_Company+TypeName  , data=data)
summary(lmC)
```

```
##
## Call:
## lm(formula = log(Price) ~ Aggregated_Company + TypeName, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.27740 -0.31902 -0.01106  0.32333  1.96234
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  6.70454    0.06136 109.267  < 2e-16 ***
## Aggregated_CompanyApple      0.35606    0.11619   3.064  0.00223 **
## Aggregated_CompanyAsus       0.18638    0.05956   3.129  0.00179 **
## Aggregated_CompanyDell       0.41217    0.05316   7.753 1.81e-14 ***
## Aggregated_CompanyHP         0.40221    0.05347   7.523 1.00e-13 ***
## Aggregated_CompanyLenovo   0.33805    0.05293   6.387 2.36e-10 ***
## Aggregated_CompanyMSI      0.36890    0.08666   4.257 2.22e-05 ***
## Aggregated_CompanyOthers   0.09942    0.08452   1.176  0.23971    
## Aggregated_CompanyRazer    0.95348    0.18209   5.236 1.91e-07 ***
## Aggregated_CompanyToshiba  0.68605    0.08093   8.477  < 2e-16 ***
## TypeNameGaming             0.33433    0.05752   5.813 7.74e-09 ***
## TypeNameNetbook           -0.85521    0.10165  -8.413  < 2e-16 ***
## TypeNameNotebook          -0.50815    0.04557 -11.152  < 2e-16 ***
## TypeNameUltrabook          0.23848    0.05538   4.306 1.79e-05 ***
## TypeNameWorkstation        0.58959    0.09595   6.145 1.06e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4596 on 1288 degrees of freedom
## Multiple R-squared:  0.463,  Adjusted R-squared:  0.4572 
## F-statistic: 79.32 on 14 and 1288 DF,  p-value: < 2.2e-16
```

```
# type I effects A, B/A   C/A,B
anova(lmC)
```

```
## Analysis of Variance Table
## 
## Response: log(Price)
##                     Df  Sum Sq Mean Sq F value    Pr(>F)    
## Aggregated_Company    9  66.642   7.405   35.05 < 2.2e-16 ***
## TypeName              5 167.965  33.593  159.01 < 2.2e-16 ***
## Residuals          1288 272.102   0.211                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

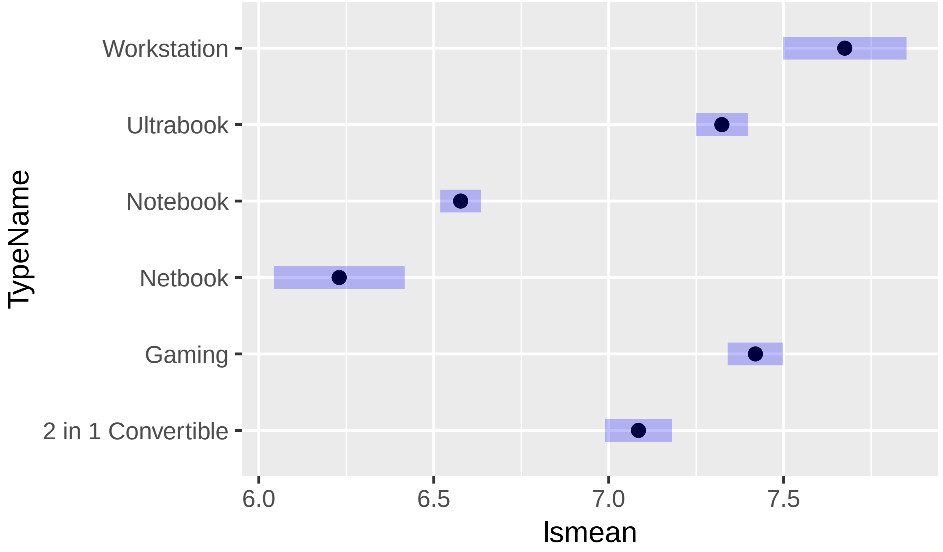
```
# type III effects A/B,C , B/A,C   C/A,B
drop1(lmC, test="F")
```

```
## Single term deletions
## 
## Model:
## log(Price) ~ Aggregated_Company + TypeName
##                    Df Sum of Sq    RSS     AIC F value    Pr(>F)    
## <none>                          272.10 -2010.8                      
## Aggregated_Company  9    28.845 300.95 -1897.5  15.171 < 2.2e-16 ***
## TypeName            5   167.965 440.07 -1394.4 159.014 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# contrasti
library(lsmeans)
ls=lsmeans(lmC, pairwise ~ TypeName ,adjust="tukey")
ls$lsmeans
```

```
##  TypeName           lsmean     SE   df lower.CL upper.CL
##  2 in 1 Convertible   7.08 0.0490 1288     6.99     7.18
##  Gaming               7.42 0.0405 1288     7.34     7.50
##  Netbook              6.23 0.0954 1288     6.04     6.42
##  Notebook             6.58 0.0296 1288     6.52     6.63
##  Ultrabook            7.32 0.0379 1288     7.25     7.40
##  Workstation          7.67 0.0900 1288     7.50     7.85
## 
## Results are averaged over the levels of: Aggregated_Company 
## Results are given on the log (not the response) scale. 
## Confidence level used: 0.95
```

```r
plot(ls$lsmeans, alpha = .05) # plot lsmeans and 95% confid int
```



```r
ls$contrasts # contrasts between predicted lsmeans
```

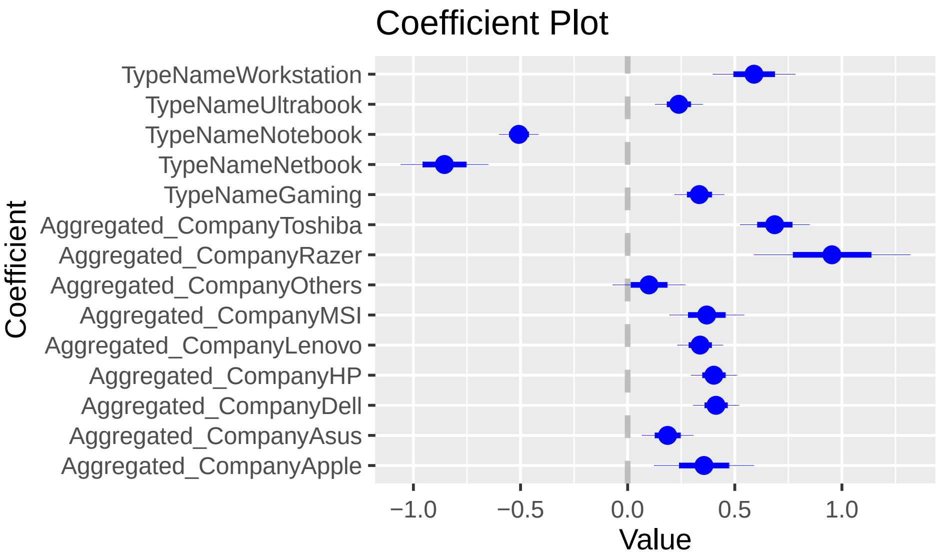
```
##  contrast                             estimate     SE   df t.ratio p.value
##  2 in 1 Convertible - Gaming           -0.3343 0.0575 1288  -5.813 <.0001
##  2 in 1 Convertible - Netbook           0.8552 0.1016 1288   8.413 <.0001
##  2 in 1 Convertible - Notebook          0.5081 0.0456 1288  11.152 <.0001
##  2 in 1 Convertible - Ultrabook        -0.2385 0.0554 1288  -4.306 0.0003
##  2 in 1 Convertible - Workstation      -0.5896 0.0959 1288  -6.145 <.0001
##  Gaming - Netbook                       1.1895 0.1005 1288  11.840 <.0001
##  Gaming - Notebook                      0.8425 0.0434 1288  19.418 <.0001
##  Gaming - Ultrabook                     0.0959 0.0531 1288   1.804 0.4636
##  Gaming - Workstation                  -0.2553 0.0952 1288  -2.681 0.0797
##  Netbook - Notebook                    -0.3471 0.0938 1288  -3.698 0.0031
##  Netbook - Ultrabook                   -1.0937 0.0992 1288 -11.029 <.0001
##  Netbook - Workstation                 -1.4448 0.1263 1288 -11.444 <.0001
##  Notebook - Ultrabook                  -0.7466 0.0396 1288 -18.833 <.0001
##  Notebook - Workstation                -1.0977 0.0877 1288 -12.516 <.0001
##  Ultrabook - Workstation               -0.3511 0.0932 1288  -3.769 0.0024
##
## Results are averaged over the levels of: Aggregated_Company
## Results are given on the log (not the response) scale.
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```r
# if at least one contrast is significant, the variable is significant in the anova table

c= contrast(ls, method = "eff") # contrast among predicted lsmeans and overall lsmean
c
```

```
## $lsmeans
##  contrast                    estimate     SE   df t.ratio p.value
##  2 in 1 Convertible effect     0.0335 0.0420 1288   0.797 0.4256
##  Gaming effect                 0.3678 0.0404 1288   9.112 <.0001
##  Netbook effect               -0.8217 0.0792 1288 -10.379 <.0001
##  Notebook effect              -0.4747 0.0278 1288 -17.081 <.0001
##  Ultrabook effect              0.2720 0.0378 1288   7.192 <.0001
##  Workstation effect            0.6231 0.0744 1288   8.378 <.0001
##
## Results are averaged over the levels of: Aggregated_Company
## P value adjustment: fdr method for 6 tests
##
## $contrasts
##  contrast                                 estimate     SE   df t.ratio
##  2 in 1 Convertible - Gaming effect        -0.1338 0.0629 1288  -2.127
##  2 in 1 Convertible - Netbook effect        1.0557 0.1070 1288   9.865
##  2 in 1 Convertible - Notebook effect       0.7086 0.0449 1288  15.790
##  2 in 1 Convertible - Ultrabook effect     -0.0380 0.0504 1288  -0.754
##  2 in 1 Convertible - Workstation effect   -0.3891 0.0650 1288  -5.986
##  Gaming - Netbook effect                    1.3900 0.1084 1288  12.825
##  Gaming - Notebook effect                   1.0430 0.0485 1288  21.496
##  Gaming - Ultrabook effect                  0.2964 0.0532 1288   5.573
##  Gaming - Workstation effect               -0.0548 0.0680 1288  -0.806
##  Netbook - Notebook effect                 -0.1466 0.0939 1288  -1.561
##  Netbook - Ultrabook effect                -0.8932 0.0968 1288  -9.228
##  Netbook - Workstation effect              -1.2443 0.1050 1288 -11.851
##  Notebook - Ultrabook effect               -0.5461 0.0478 1288 -11.415
##  Notebook - Workstation effect             -0.8972 0.0629 1288 -14.260
##  Ultrabook - Workstation effect            -0.1506 0.0736 1288  -2.047
##  p.value
##  0.0458
##  <.0001
##  <.0001
##  0.4510
##  <.0001
##  <.0001
##  <.0001
##  <.0001
##  0.4505
##  0.1369
##  <.0001
##  <.0001
##  <.0001
##  <.0001
##  0.0511
##
## Results are averaged over the levels of: Aggregated_Company
## P value adjustment: fdr method for 15 tests
```

```r
library(coefplot)
coefplot(lmC, intercept=FALSE)
```



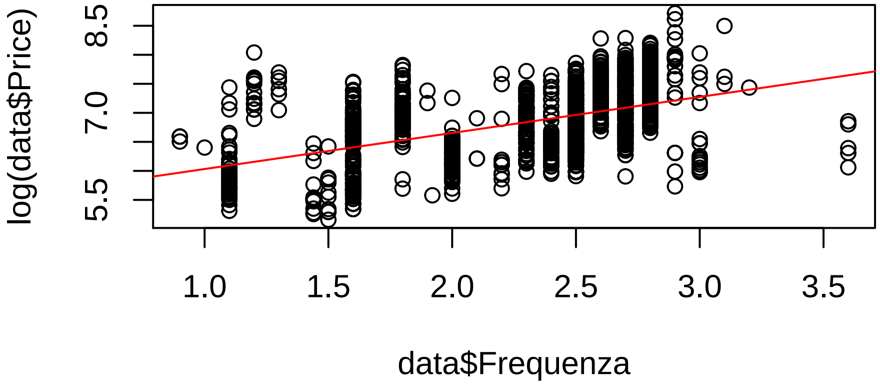
#FIXME: ho rimosso “anova k-way”, tutti d’accordo?

Regressione lineare

```r
lmA1<-lm(log(Price) ~ Frequenza  , data=data)
summary(lmA1)
```

```
##
## Call:
## lm(formula = log(Price) ~ Frequenza, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58596 -0.43023  0.00587  0.40113  1.88247
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.41132    0.06944   77.93   <2e-16 ***
## Frequenza    0.62114    0.02950   21.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.539 on 1301 degrees of freedom
## Multiple R-squared:  0.2542, Adjusted R-squared:  0.2536
## F-statistic: 443.3 on 1 and 1301 DF,  p-value: < 2.2e-16
```

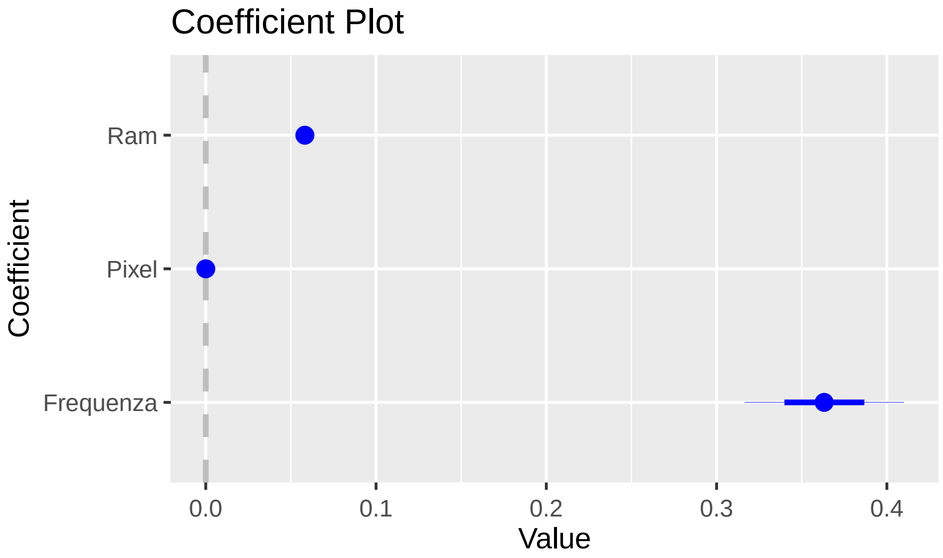
```r
plot(data$Frequenza,log(data$Price))
abline(lmA1,col="red")
```



```r
lmA2<-lm(log(Price) ~ Frequenza+Pixel+Ram  , data=data)
summary(lmA2)
```

```
##
## Call:
## lm(formula = log(Price) ~ Frequenza + Pixel + Ram, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92388 -0.29048  0.00741  0.28110  1.36597
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.266e+00  5.227e-02  100.76   <2e-16 ***
## Frequenza   3.632e-01  2.331e-02   15.58   <2e-16 ***
## Pixel       1.152e-07  8.591e-09   13.41   <2e-16 ***
## Ram         5.821e-02  2.505e-03   23.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3959 on 1299 degrees of freedom
## Multiple R-squared:  0.5981, Adjusted R-squared:  0.5972
## F-statistic: 644.5 on 3 and 1299 DF,  p-value: < 2.2e-16
```

```
coefplot(lmA2, intercept=FALSE)
```



Ancova Y = all covariates (qualitative +quantitative)

```
lmK = lm(log(Price) ~ Aggregated_Company+TypeName+SolidStateDisk+ Frequenza+Pixel+Ram  , data=data)
```

```
summary(lmK)
```

```
##
## Call:
## lm(formula = log(Price) ~ Aggregated_Company + TypeName + SolidStateDisk +
##     Frequenza + Pixel + Ram, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.06590 -0.20002 -0.00696  0.21244  1.11366
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5.484e+00  5.830e-02  94.070  < 2e-16 ***
## Aggregated_CompanyApple    3.718e-01  7.910e-02   4.701 2.87e-06 ***
## Aggregated_CompanyAsus     9.850e-02  4.027e-02   2.446  0.01458 *
## Aggregated_CompanyDell     2.147e-01  3.626e-02   5.922 4.08e-09 ***
## Aggregated_CompanyHP       2.644e-01  3.628e-02   7.290 5.42e-13 ***
## Aggregated_CompanyLenovo   1.469e-01  3.605e-02   4.076 4.87e-05 ***
## Aggregated_CompanyMSI      2.374e-01  5.871e-02   4.044 5.57e-05 ***
## Aggregated_CompanyOthers   2.731e-02  5.724e-02   0.477  0.63332
## Aggregated_CompanyRazer    3.069e-01  1.254e-01   2.446  0.01457 *
## Aggregated_CompanyToshiba  3.693e-01  5.533e-02   6.674 3.70e-11 ***
## TypeNameGaming            -8.882e-02  4.201e-02  -2.114  0.03468 *
## TypeNameNetbook           -4.098e-01  6.969e-02  -5.880 5.23e-09 ***
## TypeNameNotebook          -2.964e-01  3.188e-02  -9.298  < 2e-16 ***
## TypeNameUltrabook          9.970e-02  3.768e-02   2.646  0.00825 **
## TypeNameWorkstation        3.371e-01  6.576e-02   5.127 3.40e-07 ***
## SolidStateDiskTrue         2.891e-01  2.031e-02  14.234  < 2e-16 ***
## Frequenza                  2.751e-01  1.989e-02  13.831  < 2e-16 ***
## Pixel                      6.417e-08  7.187e-09   8.929  < 2e-16 ***
## Ram                        4.562e-02  2.227e-03  20.488  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.3098 on 1284 degrees of freedom
## Multiple R-squared:  0.7568, Adjusted R-squared:  0.7533
## F-statistic: 221.9 on 18 and 1284 DF,  p-value: < 2.2e-16
```

```
drop1(lmK, .~., test="F")
```

```
## Single term deletions
##
## Model:
## log(Price) ~ Aggregated_Company + TypeName + SolidStateDisk +
##     Frequenza + Pixel + Ram
##                    Df Sum of Sq    RSS     AIC F value    Pr(>F)
## <none>                          123.25 -3034.7
## Aggregated_Company  9    10.390 133.65 -2947.2  12.026 < 2.2e-16 ***
## TypeName            5    29.658 152.91 -2763.8  61.792 < 2.2e-16 ***
## SolidStateDisk      1    19.450 142.71 -2845.8 202.620 < 2.2e-16 ***
## Frequenza           1    18.364 141.62 -2855.7 191.304 < 2.2e-16 ***
## Pixel               1     7.653 130.91 -2958.2  79.724 < 2.2e-16 ***
## Ram                 1    40.293 163.55 -2668.1 419.752 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ls=lsmeans(lmK,pairwise ~ Aggregated_Company ,adjust="tukey")
c= contrast(ls, method = "eff")
c #FIXME: too long to be printed
```

```
## $lsmeans
##  contrast       estimate     SE   df t.ratio p.value
##  Acer effect     -0.2037 0.0333 1284 -6.113  <.0001
##  Apple effect     0.1681 0.0664 1284  2.532  0.0229
##  Asus effect     -0.1052 0.0281 1284 -3.749  0.0006
##  Dell effect      0.0110 0.0234 1284  0.471  0.6378
##  HP effect        0.0607 0.0249 1284  2.442  0.0246
##  Lenovo effect   -0.0568 0.0241 1284 -2.359  0.0264
##  MSI effect       0.0337 0.0470 1284  0.717  0.5260
##  Others effect   -0.1764 0.0459 1284 -3.847  0.0006
##  Razer effect     0.1031 0.1089 1284  0.947  0.4299
##  Toshiba effect   0.1655 0.0447 1284  3.705  0.0006
##
## Results are averaged over the levels of: TypeName, SolidStateDisk
## P value adjustment: fdr method for 10 tests
##
## $contrasts
##  contrast                 estimate     SE   df t.ratio p.value
##  Acer - Apple effect     -0.31706 0.0893 1284 -3.550  0.0014
##  Acer - Asus effect      -0.04376 0.0442 1284 -0.990  0.4146
##  Acer - Dell effect      -0.16000 0.0396 1284 -4.042  0.0002
##  Acer - HP effect        -0.20970 0.0396 1284 -5.293  <.0001
##  Acer - Lenovo effect    -0.09218 0.0387 1284 -2.380  0.0388
##  Acer - MSI effect       -0.18267 0.0572 1284 -3.195  0.0038
##  Acer - Others effect     0.02743 0.0553 1284  0.496  0.7154
##  Acer - Razer effect     -0.25213 0.1071 1284 -2.355  0.0388
##  Acer - Toshiba effect   -0.31453 0.0498 1284 -6.320  <.0001
##  Apple - Asus effect      0.32805 0.0714 1284  4.594  <.0001
##  Apple - Dell effect      0.21180 0.0681 1284  3.110  0.0048
##  Apple - HP effect        0.16211 0.0690 1284  2.349  0.0388
##  Apple - Lenovo effect    0.27962 0.0687 1284  4.071  0.0002
##  Apple - MSI effect       0.18913 0.0812 1284  2.330  0.0391
##  Apple - Others effect    0.39924 0.0757 1284  5.275  <.0001
##  Apple - Razer effect     0.11968 0.1181 1284  1.013  0.4146
##  Apple - Toshiba effect   0.05727 0.0749 1284  0.765  0.5558
##  Asus - Dell effect      -0.06150 0.0392 1284 -1.568  0.1890
##  Asus - HP effect        -0.11120 0.0402 1284 -2.768  0.0136
##  Asus - Lenovo effect     0.00632 0.0388 1284  0.163  0.9279
##  Asus - MSI effect       -0.08417 0.0538 1284 -1.566  0.1890
##  Asus - Others effect     0.12593 0.0552 1284  2.283  0.0424
##  Asus - Razer effect     -0.15363 0.1054 1284 -1.458  0.2176
##  Asus - Toshiba effect   -0.21603 0.0500 1284 -4.322  0.0001
##  Dell - HP effect         0.00505 0.0366 1284  0.138  0.9279
##  Dell - Lenovo effect     0.12257 0.0353 1284  3.471  0.0016
##  Dell - MSI effect        0.03207 0.0543 1284  0.591  0.6569
##  Dell - Others effect     0.24218 0.0526 1284  4.602  <.0001
##  Dell - Razer effect     -0.03738 0.1044 1284 -0.358  0.8105
##  Dell - Toshiba effect   -0.09978 0.0470 1284 -2.123  0.0611
##  HP - Lenovo effect       0.17226 0.0357 1284  4.828  <.0001
##  HP - MSI effect          0.08177 0.0556 1284  1.470  0.2176
##  HP - Others effect       0.29187 0.0532 1284  5.491  <.0001
##  HP - Razer effect        0.01231 0.1057 1284  0.116  0.9279
##  HP - Toshiba effect     -0.05009 0.0471 1284 -1.064  0.4044
##  Lenovo - MSI effect     -0.03575 0.0550 1284 -0.650  0.6272
##  Lenovo - Others effect   0.17436 0.0535 1284  3.257  0.0033
##  Lenovo - Razer effect   -0.10520 0.1054 1284 -0.999  0.4146
##  Lenovo - Toshiba effect -0.16760 0.0474 1284 -3.533  0.0014
##  MSI - Others effect      0.26485 0.0739 1284  3.582  0.0013
##  MSI - Razer effect      -0.01471 0.1133 1284 -0.130  0.9279
##  MSI - Toshiba effect    -0.07711 0.0698 1284 -1.104  0.3915
##  Others - Razer effect   -0.22482 0.1159 1284 -1.939  0.0912
##  Others - Toshiba effect -0.28722 0.0678 1284 -4.233  0.0001
##  Razer - Toshiba effect  -0.00766 0.1454 1284 -0.053  0.9580
##
## Results are averaged over the levels of: TypeName, SolidStateDisk
## P value adjustment: fdr method for 45 tests
```

```r
data$Product=NULL
data$X=NULL
data$Company=NULL #uso solo Aggregated_Company
data$Gpu=NULL #uso solo Gpu_company
data$dedicated_GPU=NULL
data$ScreenResolution=NULL #uso solo Pixels
data$Risoluzione=NULL #uso solo Pixels
data$Cpu=NULL #uso solo Frequenza
data$Memory=NULL #uso solo MemorySSD, TotalMemory e SolidStateDisk

lm_full = lm(log(Price) ~ ., data = data)
summary(lm_full)
```

```
##
## Call:
## lm(formula = log(Price) ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.92215 -0.18933 -0.00294  0.18376  1.01631
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.295e+00  2.419e-01  21.887  < 2e-16 ***
## TypeNameGaming             -1.522e-01  4.890e-02  -3.113 0.001890 **
## TypeNameNetbook            -4.176e-01  6.844e-02  -6.102 1.39e-09 ***
## TypeNameNotebook           -2.480e-01  3.357e-02  -7.387 2.72e-13 ***
## TypeNameUltrabook           8.461e-02  3.570e-02   2.370 0.017947 *
## TypeNameWorkstation         2.386e-01  6.678e-02   3.573 0.000366 ***
## Inches                     -2.686e-02  1.281e-02  -2.097 0.036174 *
## Ram                         3.658e-02  2.401e-03  15.233  < 2e-16 ***
## OpSysChrome OS             2.837e-01  2.143e-01   1.323 0.185942    
## OpSysLinux                 2.184e-01  2.121e-01   1.030 0.303317    
## OpSysMac OS X              9.874e-01  2.329e-01   4.240 2.40e-05 ***
## OpSysmacOS                 6.901e-01  2.262e-01   3.051 0.002329 ** 
## OpSysNo OS                 6.266e-02  2.112e-01   0.297 0.766688    
## OpSysWindows 10            3.412e-01  2.081e-01   1.640 0.101289    
## OpSysWindows 10 S          4.502e-01  2.345e-01   1.920 0.055075 .  
## OpSysWindows 7             6.572e-01  2.126e-01   3.092 0.002033 ** 
## Weight                     4.250e-02  2.830e-02   1.502 0.133381    
## Frequenza                  2.606e-01  1.918e-02  13.584  < 2e-16 ***
## Pixel                      5.252e-08  6.815e-09   7.706 2.61e-14 ***
## GpuCompanyARM              2.396e-01  2.983e-01   0.803 0.422027    
## GpuCompanyIntel            1.590e-01  2.735e-02   5.814 7.70e-09 ***
## GpuCompanyNvidia           2.521e-01  3.075e-02   8.197 5.97e-16 ***
## MemoriaSSD                 4.321e-04  9.656e-05   4.475 8.32e-06 ***
## SolidStateDiskTrue         2.196e-01  3.042e-02   7.218 9.05e-13 ***
## TotalMemory                1.361e-04  2.417e-05   5.630 2.21e-08 ***
## Aggregated_CompanyApple           NA         NA      NA       NA    
## Aggregated_CompanyAsus     9.357e-02  3.837e-02   2.439 0.014876 *  
## Aggregated_CompanyDell     2.535e-01  3.464e-02   7.320 4.38e-13 ***
## Aggregated_CompanyHP       2.672e-01  3.531e-02   7.568 7.26e-14 ***
## Aggregated_CompanyLenovo   1.732e-01  3.528e-02   4.910 1.03e-06 ***
## Aggregated_CompanyMSI      2.091e-01  5.729e-02   3.650 0.000273 ***
## Aggregated_CompanyOthers   8.039e-02  5.647e-02   1.424 0.154815    
## Aggregated_CompanyRazer    4.003e-01  1.174e-01   3.408 0.000674 ***
## Aggregated_CompanyToshiba  3.659e-01  5.276e-02   6.935 6.45e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2862 on 1270 degrees of freedom
## Multiple R-squared:  0.7947, Adjusted R-squared:  0.7895 
## F-statistic: 153.6 on 32 and 1270 DF,  p-value: < 2.2e-16
```

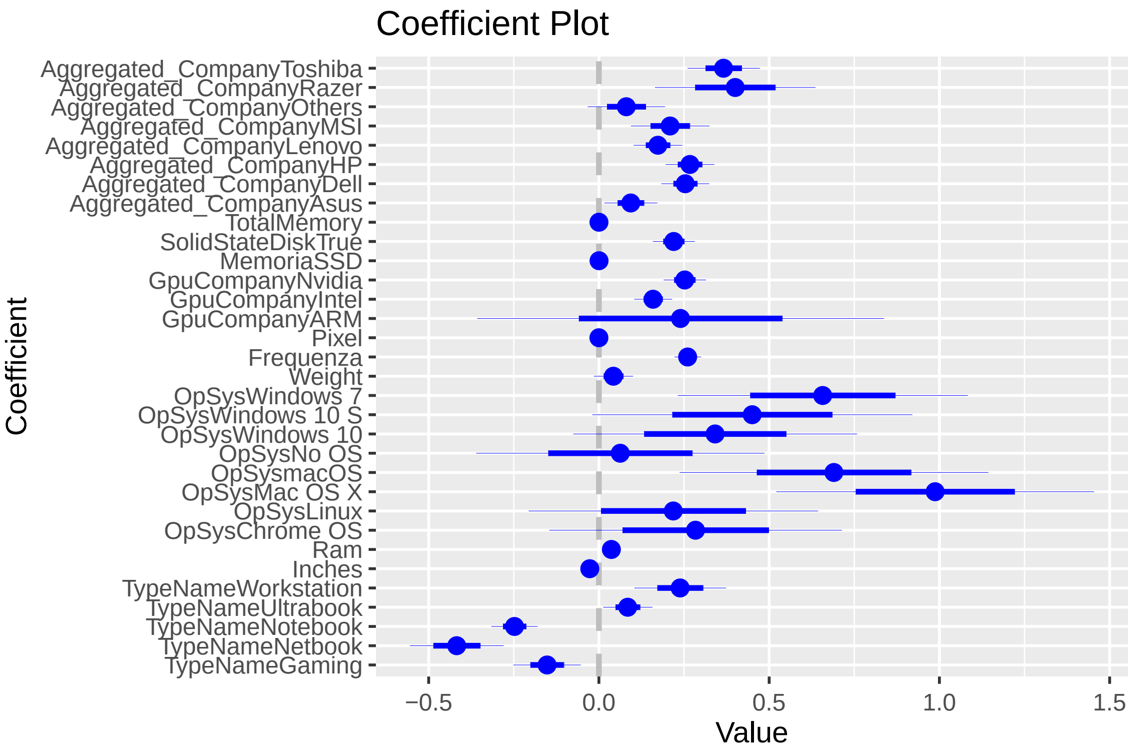
```
anova(lm_full, test="F")
```

```
## Analysis of Variance Table
## 
## Response: log(Price)
##                      Df  Sum Sq Mean Sq   F value    Pr(>F)    
## TypeName              5 205.762  41.152  502.4427 < 2.2e-16 ***
## Inches                1   2.374   2.374   28.9820 8.694e-08 ***
## Ram                   1 101.674 101.674 1241.3753 < 2.2e-16 ***
## OpSys                 8  23.287   2.911   35.5399 < 2.2e-16 ***
## Weight                1   0.145   0.145    1.7725    0.1833    
## Frequenza             1  24.973  24.973  304.9023 < 2.2e-16 ***
## Pixel                 1  10.680  10.680  130.3919 < 2.2e-16 ***
## GpuCompany            3   3.748   1.249   15.2515 9.402e-10 ***
## MemoriaSSD            1  15.007  15.007  183.2197 < 2.2e-16 ***
## SolidStateDisk        1   3.231   3.231   39.4446 4.626e-10 ***
## TotalMemory           1   2.734   2.734   33.3833 9.509e-09 ***
## Aggregated_Company    8   9.075   1.134   13.8504 < 2.2e-16 ***
## Residuals          1270 104.019   0.082                        
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

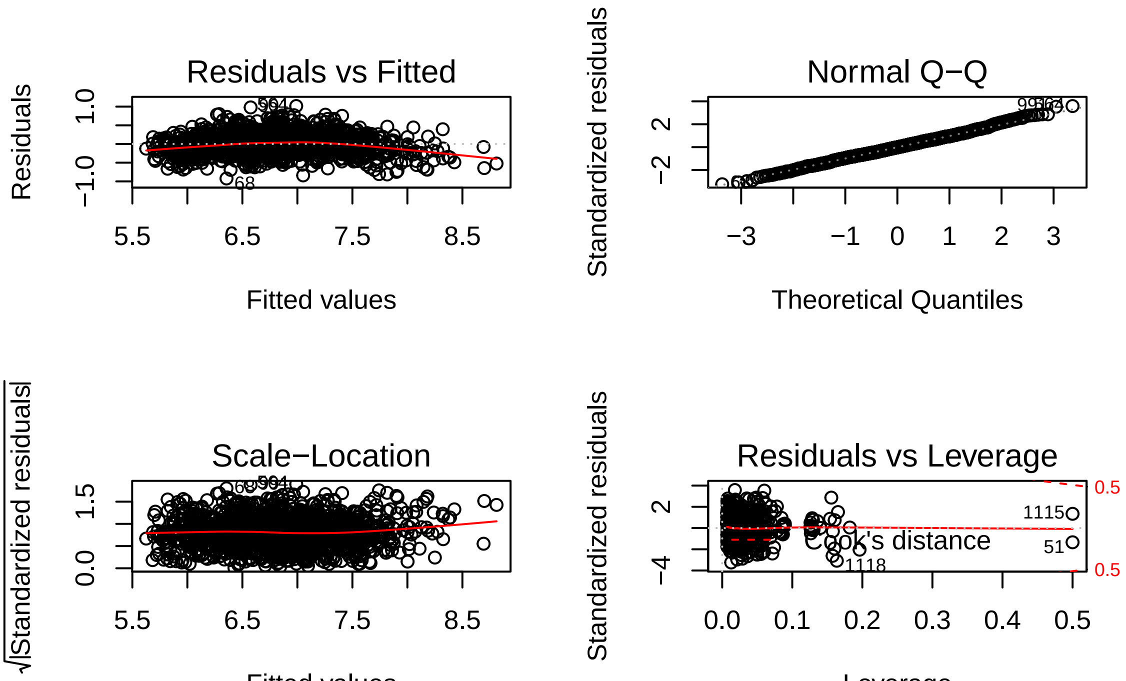
```
drop1(lm_full, test="F")
```

```
## Single term deletions
## 
## Model:
## log(Price) ~ TypeName + Inches + Ram + OpSys + Weight + Frequenza + 
##     Pixel + GpuCompany + MemoriaSSD + SolidStateDisk + TotalMemory + 
##     Aggregated_Company
##                    Df Sum of Sq    RSS     AIC  F value    Pr(>F)
## <none>                          104.02 -3227.8
## TypeName            5   17.2030 121.22 -3038.4  42.0072 < 2.2e-16 ***
## Inches              1    0.3602 104.38 -3225.3   4.3982   0.03617 *
## Ram                 1   19.0067 123.03 -3011.1 232.0592 < 2.2e-16 ***
## OpSys               7   10.0159 114.03 -3122.0  17.4697 < 2.2e-16 ***
## Weight              1    0.1847 104.20 -3227.5   2.2556   0.13338
## Frequenza           1   15.1135 119.13 -3053.0 184.5257 < 2.2e-16 ***
## Pixel               1    4.8634 108.88 -3170.2  59.3792 2.608e-14 ***
## GpuCompany          3    5.6591 109.68 -3164.8  23.0314 1.610e-14 ***
## MemoriaSSD          1    1.6403 105.66 -3209.4  20.0266 8.322e-06 ***
## SolidStateDisk      1    4.2673 108.29 -3177.4  52.1009 9.048e-13 ***
## TotalMemory         1    2.5966 106.62 -3197.7  31.7023 2.210e-08 ***
## Aggregated_Company  8    9.0753 113.09 -3134.8  13.8504 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

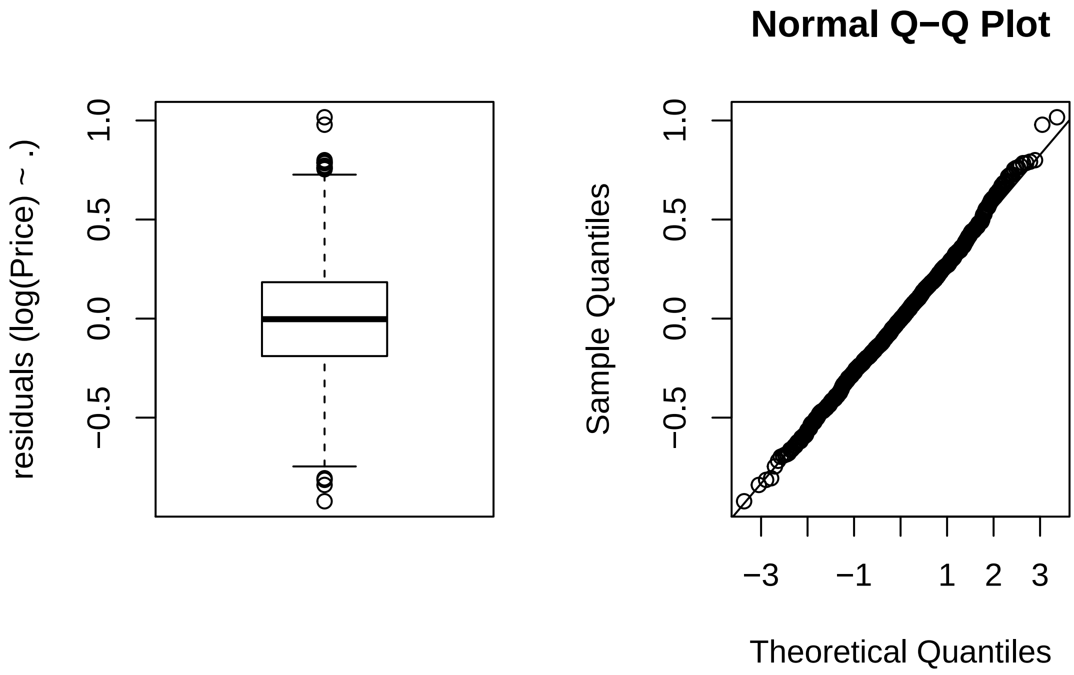
```
coefplot(lm_full, intercept=FALSE)
```



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



```r
par(mfrow=c(1,1))
par(mfrow=c(1,2))
boxplot(lm_full$residuals, ylab="residuals (log(Price) ~ .)")
qqnorm(lm_full$residuals);qqline(lm_full$residuals)
```



```r
#normality tests
ad.test(lm_full$residuals)
```

```
##
##  Anderson-Darling normality test
##
## data:  lm_full$residuals
## A = 0.47935, p-value = 0.2341
```

```
shapiro.test(lm_full$residuals)
```

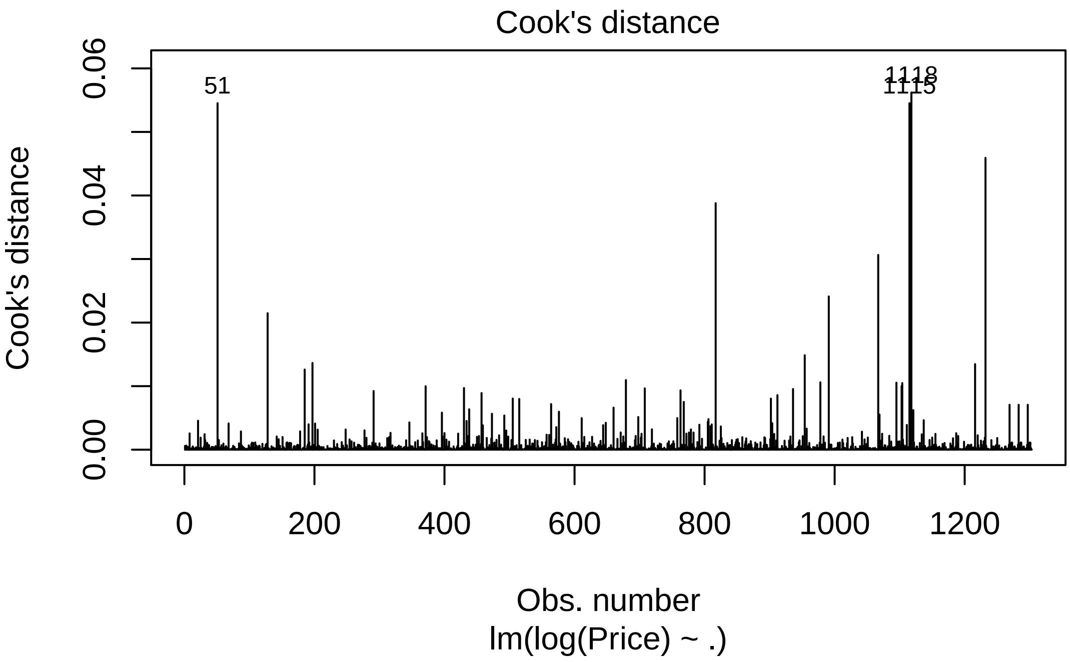
```
##
##  Shapiro-Wilk normality test
##
## data:  lm_full$residuals
## W = 0.99827, p-value = 0.2046
```

A look over outliers

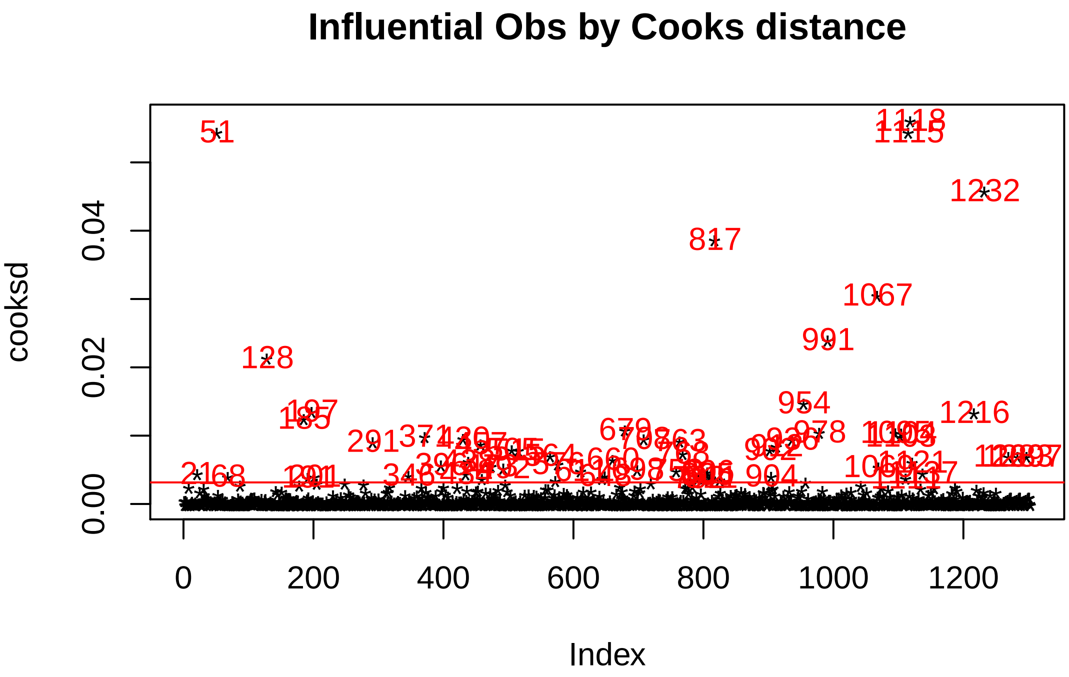
```
cooksd <- cooks.distance(lm_full) #Cook's Distance
cooksda=data.frame(cooksd)
summary(cooksd)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## 0.0000000 0.0000602 0.0002372 0.0009733 0.0006906 0.0562069         1
```

```
cutoff <- 4/((nrow(data)-length(lm_full$coefficients)-2)) # identify D values > 4/(n-k-1)
plot(lm_full, which=4, cook.levels=cutoff)# Cook's D plot
```



```
plot(cooksd, pch="*", cex=1, main="Influential Obs by Cooks distance") # plot cook's distance
abline(h = cutoff, col="red") # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4*mean(cooksd, na.rm=T),names(cooksd),""), col="red")#add la
```



```r
#extract influencial obs
influential <- as.numeric(names(cooksd)[(cooksd > cutoff)]) # influential row numbers
influ=data.frame(data[cooksd > cutoff, ])
filtered_data <- data[ !(row.names(data) %in% c(influential)), ]
dim(influ); dim(data); dim(filtered_data)
```

```
## [1] 68 13
```

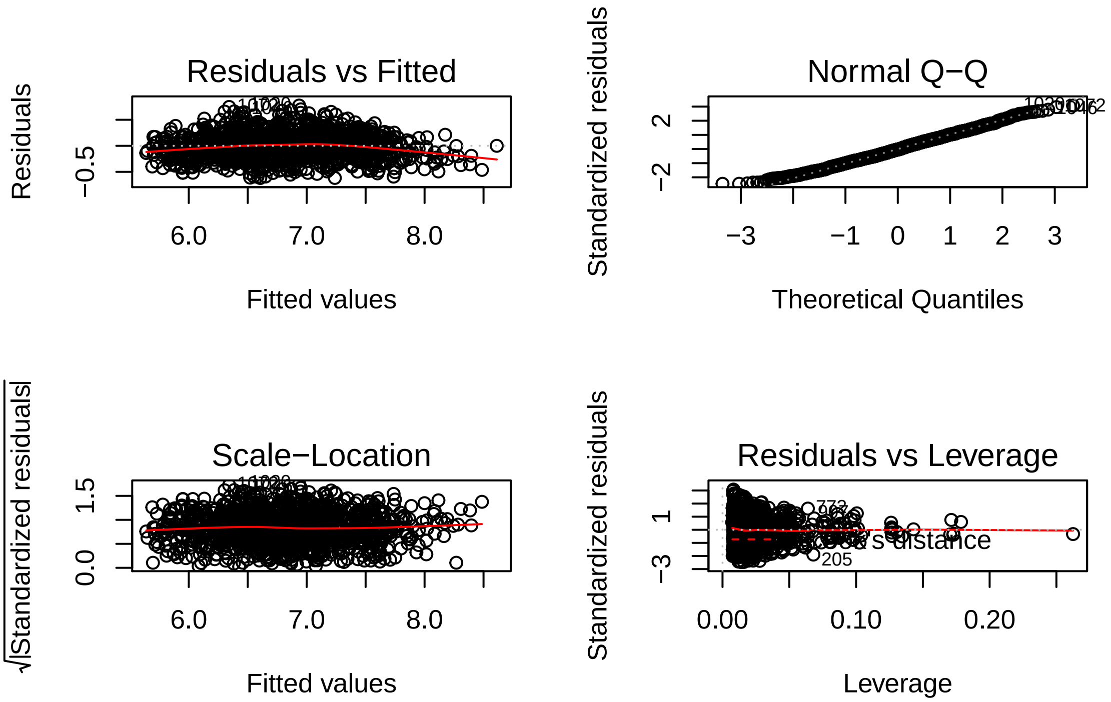
```
## [1] 1303   13
```

```
## [1] 1236   13
```

```r
#removed outliers
lm_full_t_no_OUTliers = lm(log(Price) ~ ., data = filtered_data)
summary(lm_full_t_no_OUTliers)
```

```
##
## Call:
## lm(formula = log(Price) ~ ., data = filtered_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61803 -0.17506 -0.00775  0.17186  0.77302
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                5.657e+00  1.536e-01  36.824  < 2e-16 ***
## TypeNameGaming            -1.166e-01  4.534e-02  -2.573  0.01021 *
## TypeNameNetbook           -3.902e-01  7.902e-02  -4.938 8.99e-07 ***
## TypeNameNotebook          -2.145e-01  3.057e-02  -7.018 3.74e-12 ***
## TypeNameUltrabook          7.777e-02  3.263e-02   2.383  0.01731 *
## TypeNameWorkstation        2.963e-01  6.291e-02   4.710 2.77e-06 ***
## Inches                    -3.568e-02  1.167e-02  -3.058  0.00228 **
## Ram                        3.731e-02  2.564e-03  14.555  < 2e-16 ***
## OpSysLinux                -5.027e-02  7.745e-02  -0.649  0.51640
## OpSysMac OS X              7.106e-01  1.158e-01   6.139 1.13e-09 ***
## OpSysmacOS                 4.042e-01  1.029e-01   3.929 9.00e-05 ***
## OpSysNo OS                -2.267e-01  7.859e-02  -2.884  0.00400 **
## OpSysWindows 10            7.851e-02  7.050e-02   1.114  0.26571
## OpSysWindows 10 S          2.550e-01  1.295e-01   1.969  0.04916 *
## OpSysWindows 7             3.791e-01  8.149e-02   4.652 3.65e-06 ***
## Weight                    3.921e-02  2.627e-02   1.493  0.13582    
## Frequenza                 2.577e-01  1.745e-02  14.763  < 2e-16 ***
## Pixel                     6.074e-08  6.667e-09   9.111  < 2e-16 ***
## GpuCompanyARM             3.010e-01  2.685e-01   1.121  0.26259    
## GpuCompanyIntel           1.636e-01  2.450e-02   6.675 3.76e-11 ***
## GpuCompanyNvidia          2.450e-01  2.773e-02   8.835  < 2e-16 ***
## MemoriaSSD                3.895e-04  9.203e-05   4.232 2.49e-05 ***
## SolidStateDiskTrue        2.479e-01  2.826e-02   8.774  < 2e-16 ***
## TotalMemory               1.621e-04  2.316e-05   7.000 4.24e-12 ***
## Aggregated_CompanyApple          NA         NA      NA       NA    
## Aggregated_CompanyAsus    7.400e-02  3.467e-02   2.135  0.03298 *  
## Aggregated_CompanyDell    2.272e-01  3.102e-02   7.326 4.35e-13 ***
## Aggregated_CompanyHP      2.369e-01  3.171e-02   7.472 1.52e-13 ***
## Aggregated_CompanyLenovo  1.443e-01  3.162e-02   4.562 5.58e-06 ***
## Aggregated_CompanyMSI     1.640e-01  5.141e-02   3.190  0.00146 ** 
## Aggregated_CompanyOthers  2.303e-02  5.698e-02   0.404  0.68613    
## Aggregated_CompanyRazer   2.530e-01  2.614e-01   0.968  0.33331    
## Aggregated_CompanyToshiba 3.475e-01  4.710e-02   7.378 2.98e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2529 on 1204 degrees of freedom
## Multiple R-squared:  0.8244, Adjusted R-squared:  0.8198 
## F-statistic: 182.3 on 31 and 1204 DF,  p-value: < 2.2e-16
```

```r
par(mfrow=c(2,2))
plot(lm_full_t_no_OUTliers)
```



```r
library(car)
ncvTest(lm_full_t_no_OUTliers)
```

```
## Non-constant Variance Score Test 
## Variance formula: ~ fitted.values 
## Chisquare = 0.001667699, Df = 1, p = 0.96743
```

```r
null = lm(log(Price) ~ 1, data = filtered_data)
full = lm(log(Price) ~ ., data = filtered_data)
library(MASS)
lm_fit = stepAIC(null, scope = list(upper = full), direction = "both", trace = FALSE)
```

```r
summary(lm_fit)
```

```
##
## Call:
## lm(formula = log(Price) ~ Ram + TypeName + SolidStateDisk + Frequenza +
##     OpSys + Pixel + Aggregated_Company + GpuCompany + TotalMemory +
##     MemoriaSSD + Inches + Weight, data = filtered_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61803 -0.17506 -0.00775  0.17186  0.77302
##
## Coefficients: (1 not defined because of singularities)
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                5.657e+00  1.536e-01  36.824  < 2e-16 ***
## Ram                        3.731e-02  2.564e-03  14.555  < 2e-16 ***
## TypeNameGaming            -1.166e-01  4.534e-02  -2.573  0.01021 *
## TypeNameNetbook           -3.902e-01  7.902e-02  -4.938 8.99e-07 ***
## TypeNameNotebook          -2.145e-01  3.057e-02  -7.018 3.74e-12 ***
## TypeNameUltrabook          7.777e-02  3.263e-02   2.383  0.01731 *
## TypeNameWorkstation        2.963e-01  6.291e-02   4.710 2.77e-06 ***
## SolidStateDiskTrue         2.479e-01  2.826e-02   8.774  < 2e-16 ***
## Frequenza                  2.577e-01  1.745e-02  14.763  < 2e-16 ***
## OpSysLinux                -5.027e-02  7.745e-02  -0.649  0.51640
## OpSysMac OS X              7.106e-01  1.158e-01   6.139 1.13e-09 ***
## OpSysmacOS                 4.042e-01  1.029e-01   3.929 9.00e-05 ***
## OpSysNo OS                -2.267e-01  7.859e-02  -2.884  0.00400 **
## OpSysWindows 10            7.851e-02  7.050e-02   1.114  0.26571
## OpSysWindows 10 S          2.550e-01  1.295e-01   1.969  0.04916 *
## OpSysWindows 7             3.791e-01  8.149e-02   4.652 3.65e-06 ***
## Pixel                      6.074e-08  6.667e-09   9.111  < 2e-16 ***
## Aggregated_CompanyApple           NA         NA      NA       NA
## Aggregated_CompanyAsus     7.400e-02  3.467e-02   2.135  0.03298 *
## Aggregated_CompanyDell     2.272e-01  3.102e-02   7.326 4.35e-13 ***
## Aggregated_CompanyHP       2.369e-01  3.171e-02   7.472 1.52e-13 ***
## Aggregated_CompanyLenovo   1.443e-01  3.162e-02   4.562 5.58e-06 ***
## Aggregated_CompanyMSI      1.640e-01  5.141e-02   3.190  0.00146 **
## Aggregated_CompanyOthers   2.303e-02  5.698e-02   0.404  0.68613
## Aggregated_CompanyRazer    2.530e-01  2.614e-01   0.968  0.33331
## Aggregated_CompanyToshiba  3.475e-01  4.710e-02   7.378 2.98e-13 ***
## GpuCompanyARM              3.010e-01  2.685e-01   1.121  0.26259
## GpuCompanyIntel            1.636e-01  2.450e-02   6.675 3.76e-11 ***
## GpuCompanyNvidia           2.450e-01  2.773e-02   8.835  < 2e-16 ***
## TotalMemory                1.621e-04  2.316e-05   7.000 4.24e-12 ***
## MemoriaSSD                 3.895e-04  9.203e-05   4.232 2.49e-05 ***
## Inches                    -3.568e-02  1.167e-02  -3.058  0.00228 **
## Weight                     3.921e-02  2.627e-02   1.493  0.13582
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2529 on 1204 degrees of freedom
## Multiple R-squared:  0.8244, Adjusted R-squared:  0.8198
## F-statistic: 182.3 on 31 and 1204 DF,  p-value: < 2.2e-16
```

```r
drop1(lm_fit, test = 'F')
```

```
## Single term deletions
##
## Model:
## log(Price) ~ Ram + TypeName + SolidStateDisk + Frequenza + OpSys +
##     Pixel + Aggregated_Company + GpuCompany + TotalMemory + MemoriaSSD +
##     Inches + Weight
##                    Df Sum of Sq    RSS     AIC  F value    Pr(>F)    
## <none>                           76.981 -3367.2                      
## Ram                 1   13.5450 90.526 -3168.9 211.8485 < 2.2e-16 ***
## TypeName            5   13.0233 90.004 -3184.1  40.7378 < 2.2e-16 ***
## SolidStateDisk      1    4.9216 81.902 -3292.6  76.9749 < 2.2e-16 ***
## Frequenza           1   13.9357 90.916 -3163.6 217.9593 < 2.2e-16 ***
## OpSys               6    9.8885 86.869 -3229.9  25.7766 < 2.2e-16 ***
## Pixel               1    5.3071 82.288 -3286.8  83.0054 < 2.2e-16 ***
## Aggregated_Company  8    7.6931 84.674 -3265.5  15.0404 < 2.2e-16 ***
## GpuCompany          3    5.2492 82.230 -3291.7  27.3661 < 2.2e-16 ***
## TotalMemory         1    3.1331 80.114 -3319.9  49.0032 4.235e-12 ***
## MemoriaSSD          1    1.1450 78.126 -3351.0  17.9087 2.494e-05 ***
## Inches              1    0.5978 77.578 -3359.7   9.3505  0.002278 ** 
## Weight              1    0.1424 77.123 -3367.0   2.2277  0.135818    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

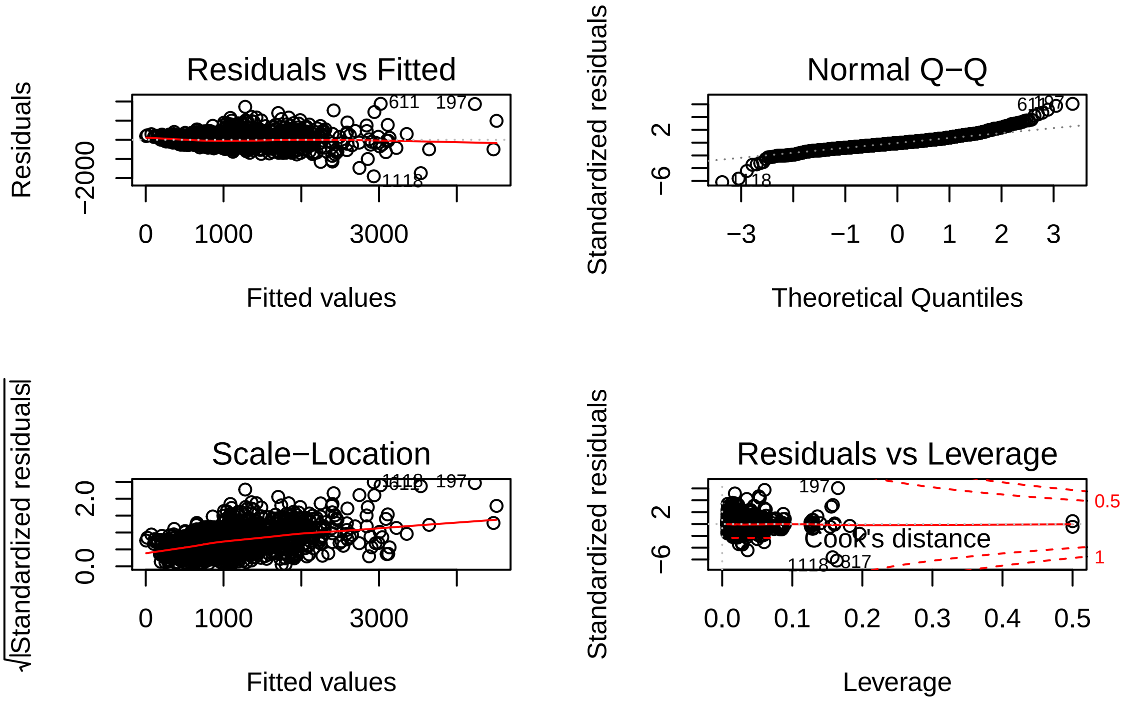
APPENDIX

no log model and a log justification

```
lm_full_no_log = lm(Price ~ ., data = data)
summary(lm_full_no_log)
```

```
## 
## Call:
## lm(formula = Price ~ ., data = data)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1903.51  -188.44   -30.52   157.09  1877.64 
## 
## Coefficients: (1 not defined because of singularities)
##                           Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             -4.284e+02  2.847e+02  -1.505 0.132627    
## TypeNameGaming          -1.079e+02  5.754e+01  -1.875 0.061028 .  
## TypeNameNetbook         -1.331e+02  8.055e+01  -1.653 0.098631 .  
## TypeNameNotebook        -2.220e+02  3.951e+01  -5.620 2.34e-08 ***
## TypeNameUltrabook        1.185e+02  4.202e+01   2.819 0.004890 ** 
## TypeNameWorkstation      5.933e+02  7.859e+01   7.550 8.28e-14 ***
## Inches                  -2.889e+01  1.507e+01  -1.917 0.055517 .  
## Ram                      5.882e+01  2.826e+00  20.812  < 2e-16 ***
## OpSysChrome OS           2.147e+02  2.522e+02   0.851 0.394870    
## OpSysLinux               1.462e+02  2.496e+02   0.586 0.558155    
## OpSysMac OS X            5.554e+02  2.741e+02   2.026 0.042928 *  
## OpSysmacOS               5.683e+02  2.662e+02   2.135 0.032974 *  
## OpSysNo OS              -2.565e+00  2.485e+02  -0.010 0.991766    
## OpSysWindows 10          2.318e+02  2.449e+02   0.946 0.344163    
## OpSysWindows 10 S        3.409e+02  2.760e+02   1.235 0.216907    
## OpSysWindows 7           5.972e+02  2.502e+02   2.387 0.017113 *  
## Weight                   1.625e+02  3.331e+01   4.879 1.20e-06 ***
## Frequenza                1.777e+02  2.258e+01   7.869 7.62e-15 ***
## Pixel                    8.450e-05  8.020e-06  10.536  < 2e-16 ***
## GpuCompanyARM           -4.869e+01  3.511e+02  -0.139 0.889718    
## GpuCompanyIntel          1.926e+02  3.219e+01   5.983 2.84e-09 ***
## GpuCompanyNvidia         1.623e+02  3.619e+01   4.485 7.94e-06 ***
## MemoriaSSD               4.719e-01  1.136e-01   4.153 3.51e-05 ***
## SolidStateDiskTrue       5.015e+01  3.580e+01   1.401 0.161533    
## TotalMemory              2.571e-02  2.844e-02   0.904 0.366155    
## Aggregated_CompanyApple         NA         NA      NA       NA    
## Aggregated_CompanyAsus   8.675e+01  4.516e+01   1.921 0.054930 .  
## Aggregated_CompanyDell     1.563e+02  4.076e+01   3.834 0.000132 ***
## Aggregated_CompanyHP       2.189e+02  4.156e+01   5.267 1.63e-07 ***
## Aggregated_CompanyLenovo   1.620e+02  4.152e+01   3.902 0.000100 ***
## Aggregated_CompanyMSI      2.784e+02  6.743e+01   4.128 3.89e-05 ***
## Aggregated_CompanyOthers   1.740e+02  6.646e+01   2.618 0.008950 **
## Aggregated_CompanyRazer    1.183e+03  1.382e+02   8.559  < 2e-16 ***
## Aggregated_CompanyToshiba  2.989e+02  6.209e+01   4.815 1.65e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 336.8 on 1270 degrees of freedom
## Multiple R-squared:  0.7735, Adjusted R-squared:  0.7678
## F-statistic: 135.6 on 32 and 1270 DF,  p-value: < 2.2e-16
```

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



```
ad.test(lm_full_no_log$residuals)
```

```
##
##  Anderson-Darling normality test
##
## data:  lm_full_no_log$residuals
## A = 15.82, p-value < 2.2e-16
```

```
shapiro.test(lm_full_no_log$residuals)
```

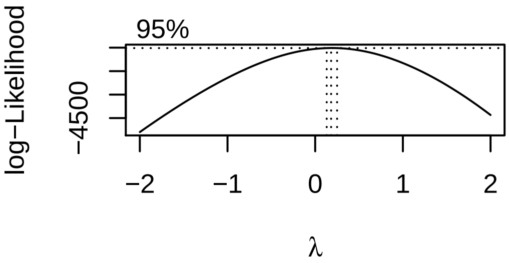
```
##
##  Shapiro-Wilk normality test
##
## data:  lm_full_no_log$residuals
## W = 0.93651, p-value < 2.2e-16
```

```
library(MASS)
boxcoxreg1<-boxcox(lm_full_no_log, plotit=T) #to justify log correction
which.max(boxcoxreg1$y)
```

```
## [1] 55
```

```
lambda=boxcoxreg1$x[which.max(boxcoxreg1$y)]
lambda #not exactly lambda= 0 but almost, one could also apply y'=((y^lambda) - 1) / lambda
```

```
## [1] 0.1818182
```


log−Likelihood
λ