

Progetto

Andrea Corvaglia at All

17 agosto 2019

Contents

Descriptive analysis on Y	1
Test on a mean (justify H0) on Y and confidence limits.	12
Test two means, two variances (Y vs X) .	13
Association/chi square among some couples of categorical Xj	14
Anova one way Y = Xj, for a categorical X	18
Anova two way Y = Xj Xk for some categorical X	32
Ancova Y = all covariates (qualitative +quantitative)	40

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 3 ...
## $ TypeName       : Factor w/ 6 levels "2 in 1 Convertible",...: 5 5 4 5 5 4 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 5 ...
## $ 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 1 ...
## $ 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 ...
## $ 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)
```

```
##      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
## 4 4   Apple MacBook Pro Ultrabook 15.4
## 5 5   Apple MacBook Pro Ultrabook 13.3
## 6 6   Acer   Aspire 3 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
## 4 IPS Panel Retina Display 2880x1800      Intel Core i7 2.7GHz 16
## 5 IPS Panel Retina Display 2560x1600      Intel Core i5 3.1GHz 8
## 6                               1366x768      AMD A9-Series 9420 3GHz 4
##
##                               Memory          Gpu      OpSys Weight
## 1                               128GB SSD Intel Iris Plus Graphics 640      macOS 1.37
## 2 128GB Flash Storage          Intel HD Graphics 6000      macOS 1.34
## 3                               256GB SSD      Intel HD Graphics 620      No OS 1.86
## 4                               512GB SSD          AMD Radeon Pro 455      macOS 1.83
## 5                               256GB SSD Intel Iris Plus Graphics 650      macOS 1.37
## 6                               500GB HDD          AMD Radeon R5 Windows 10 2.10
##
##      Price Frequenza Risoluzione Pixel GpuCompany MemoriaSSD
## 1 1339.69      2.3 2560x1600 4096000 Intel      128
## 2 898.94      1.8 1440x900 1296000 Intel      0
## 3 575.00      2.5 1920x1080 2073600 Intel      256
## 4 2537.45      2.7 2880x1800 5184000 AMD      512
## 5 1803.60      3.1 2560x1600 4096000 Intel      256
## 6 400.00      3.0 1366x768 1049088 AMD      0
##
##      SolidStateDisk TotalMemory dedicated_GPU Aggregated_Company
## 1      True      128      False      Apple
## 2      False      128      False      Apple
## 3      True      256      False      HP
## 4      True      512      True      Apple
## 5      True      256      False      Apple
## 6      False      500      True      Acer
```

```
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
```

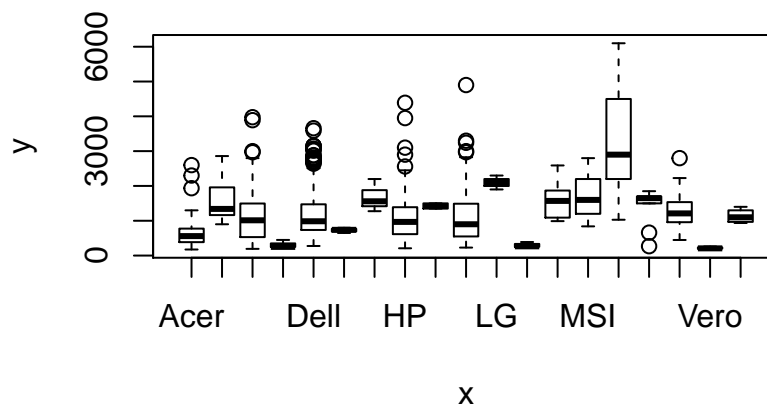
```
data$Weight<-as.numeric(data$Weight)
data$Ram<-as.numeric(data$Ram)
```

```
supply(data, function(x)(sum(is.na(x))))
```

```
## 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
```

```
# Non ci sono missing data!
```

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

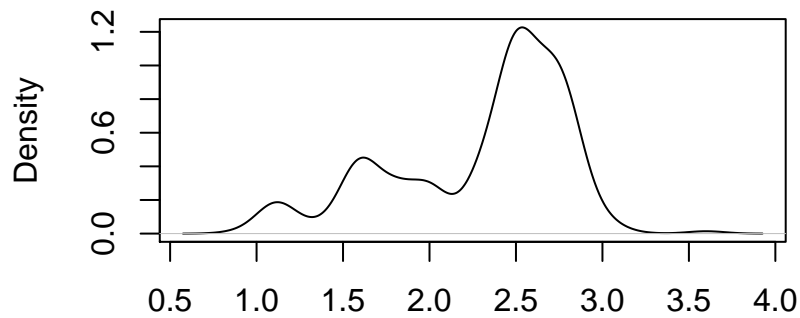


```
class(data$Ram)
```

```
## [1] "numeric"
```

```
plot(density(data$Frequenza))
```

density.default(x = data\$Frequenza)

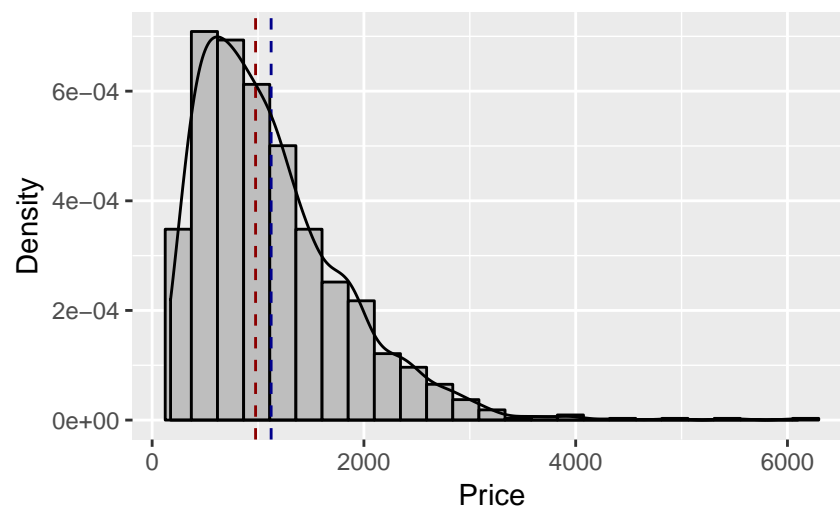


N = 1303 Bandwidth = 0.1086

```
#hist(data$Price, breaks=25, probability=TRUE)
#lines(density(data$Price))
```

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

Price Distribution with mean and median



Quite skewed to the right, mean > median

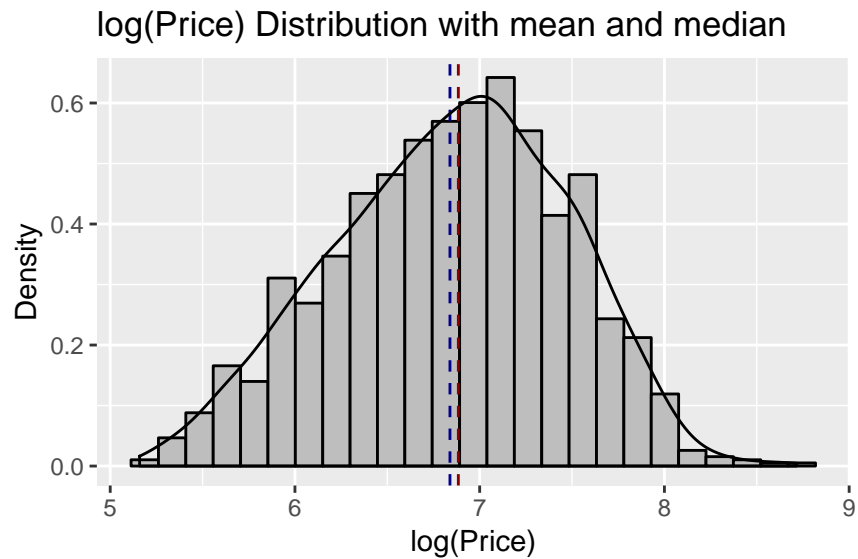
We could try to apply a correction like Log(Y)

```
data$LogPrice=log(data$Price)
ggplot(data,aes(x = log(Price))) +
  geom_histogram(aes(y =..density..),
```

```

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

```



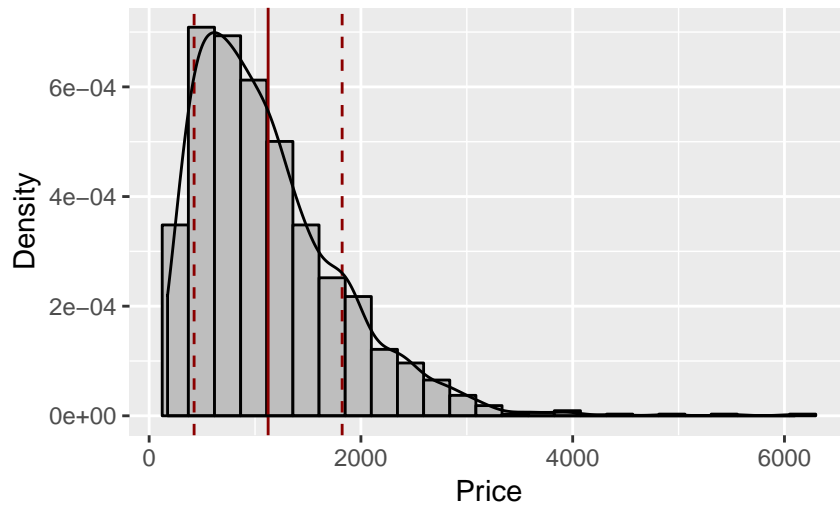
Now the distribution is looking a bit better (as regards normality)

```

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$Price), color = "dark red", lty = 2) +
  geom_vline(xintercept = mean(data$Price) - sd(data$Price), color = "dark red", lty = 2) +
  labs(x = "Price", y ="Density") +
  ggtitle("Price Distribution (mean +/- sd)") +
  geom_density()

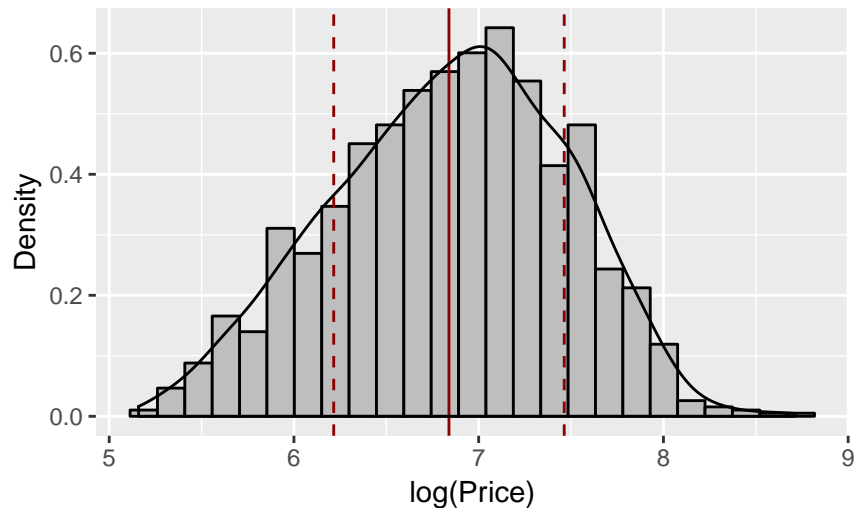
```

Price Distribution (mean \pm sd)



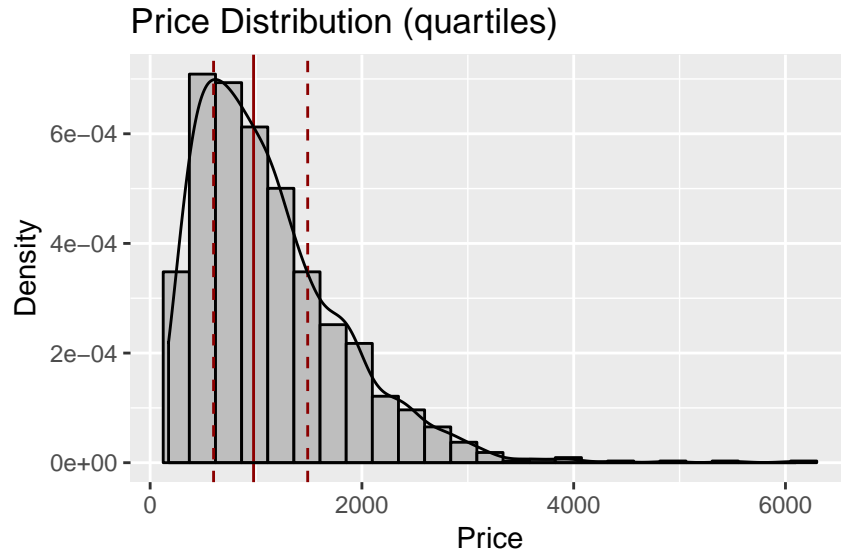
```
ggplot(data,aes(x = log(Price))) +
  geom_histogram(aes(y =..density..),
    bins= 25,
    fill = "grey",
    color ="black") +
  geom_vline(xintercept = mean(data$LogPrice), color = "dark red") +
  geom_vline(xintercept = mean(data$LogPrice) + sd(data$LogPrice), color = "dark red", lty = 2) +
 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

log(Price) Distribution (mean \pm sd)

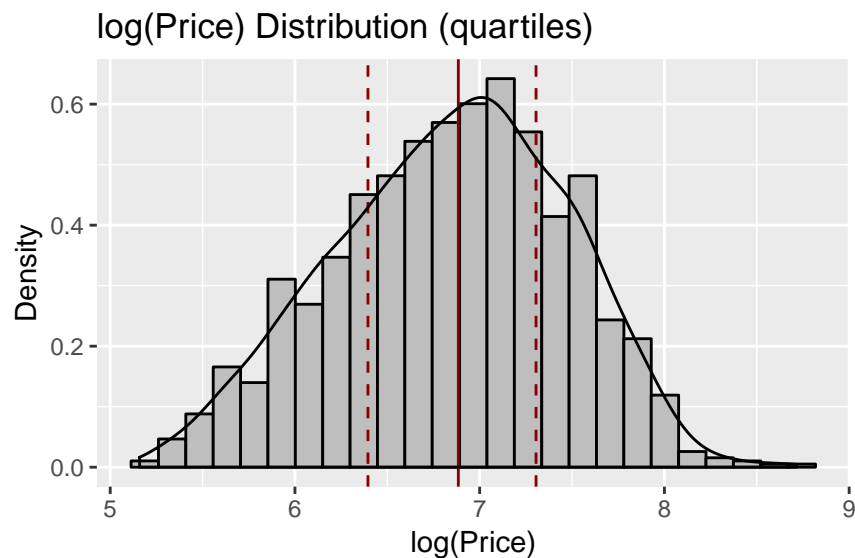


```
ggplot(data,aes(x = Price)) +
  geom_histogram(aes(y =..density..),
    bins= 25,
    fill = "grey",
    color ="black") +
  geom_vline(xintercept = quantile(data$Price, 0.25), color = "dark red",lty = 2) +
```

```
geom_vline(xintercept = quantile(data$Price, 0.5), color = "dark red", ) +
geom_vline(xintercept = quantile(data$Price, 0.75), color = "dark red", lty = 2) +
labs(x = "Price", y = "Density") +
ggtitle("Price Distribution (quartiles)") +
geom_density()
```

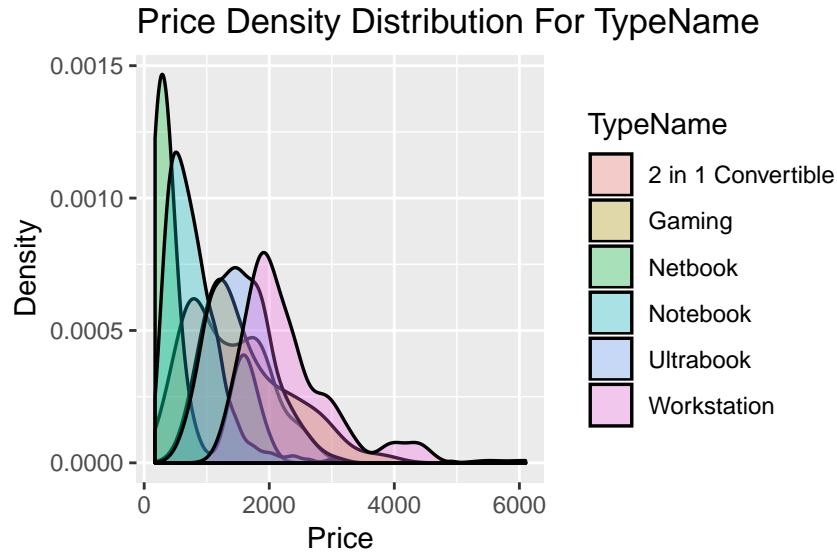


```
ggplot(data,aes(x = log(Price))) +
  geom_histogram(aes(y = ..density..),
    bins= 25,
    fill = "grey",
    color = "black") +
  geom_vline(xintercept = quantile(data$LogPrice, 0.25), color = "dark red",lty = 2) +
  geom_vline(xintercept = quantile(data$LogPrice, 0.5), color = "dark red", ) +
  geom_vline(xintercept = quantile(data$LogPrice, 0.75), color = "dark red", lty = 2) +
  labs(x = "log(Price)", y = "Density") +
  ggtitle("log(Price) Distribution (quartiles)") +
  geom_density()
```

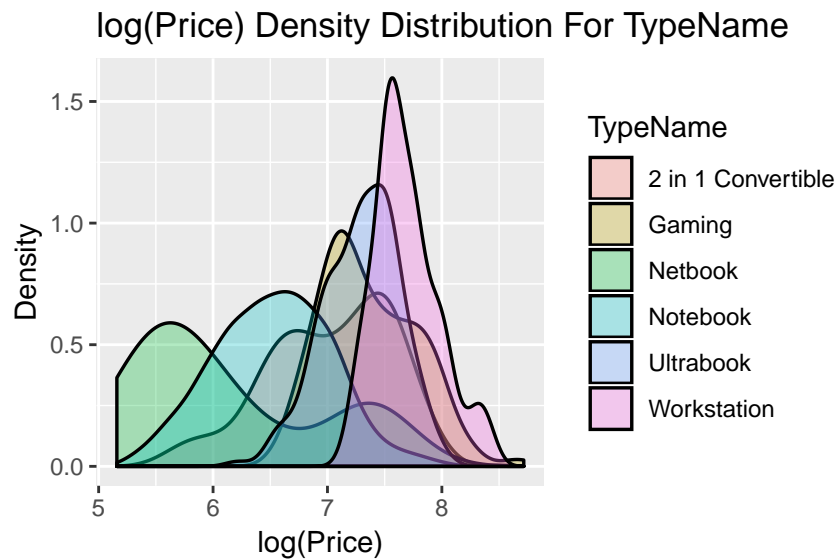


Descrittive variabile dipendente price

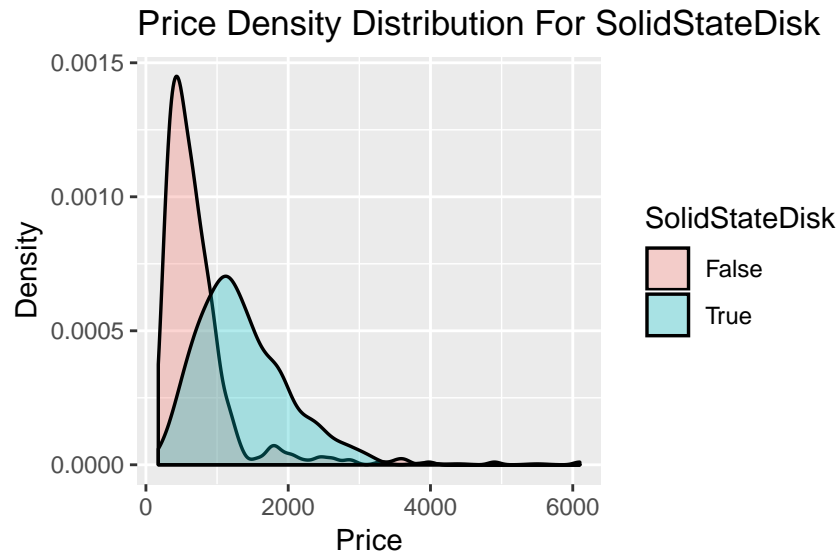
```
ggplot(data, aes(x = Price, fill = TypeName)) +  
  geom_density(size = 0.6, alpha = .3) +  
  labs(x = "Price", y = "Density", fill = "TypeName") +  
  ggtitle("Price Density Distribution For TypeName")
```



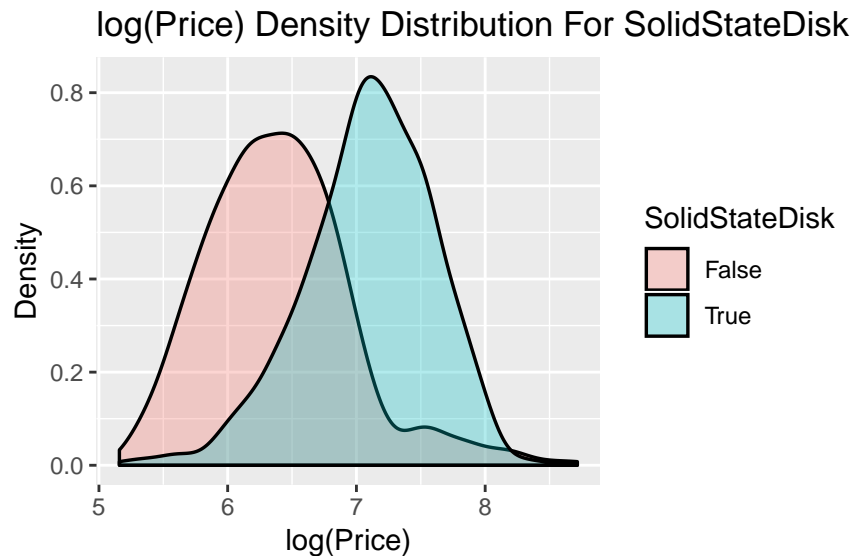
```
ggplot(data, aes(x = log(Price), fill = TypeName)) +  
  geom_density(size = 0.6, alpha = .3) +  
  labs(x = "log(Price)", y = "Density", fill = "TypeName") +  
  ggtitle("log(Price) Density Distribution For TypeName")
```



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



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

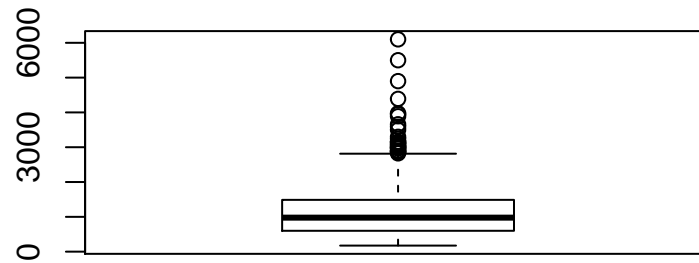


```
library(psych)
describe(data$Price)
```

```
##      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
```

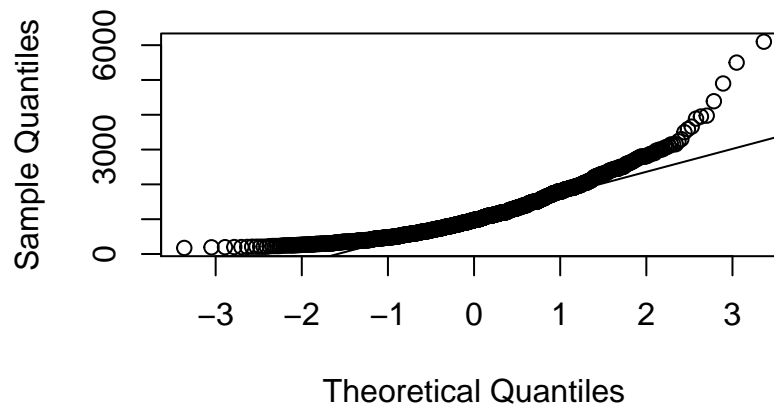
```
library(nortest)
# NORMALITA'

boxplot(data$Price)
```



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

Normal Q-Q Plot



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

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

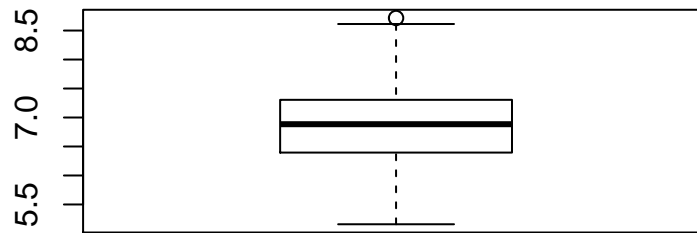
```
ad.test(data$Price)
```

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

```
#wilcox.test(data$Price, conf.int = TRUE, mu = ) #worth it?  
#if(!require(Envstats)) install.packages("EnvStats")
```

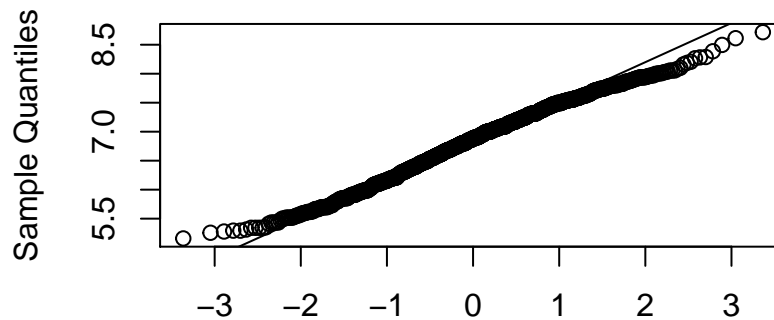
Trying with the log correction:

```
# Correzione NORMALITA'  
library(nortest)  
boxplot(data$LogPrice)
```



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

Normal Q-Q Plot



Theoretical Quantiles

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

```
##
##  Shapiro-Wilk normality test
##
## data:  data$LogPrice
## W = 0.99252, p-value = 3.628e-06
```

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

```
##
##  Anderson-Darling normality test
##
## data:  data$LogPrice
## 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$Price)
Apple<-data$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

#FIXME: var test?
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
```

```
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
```

```
# 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
```

```
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
```

```
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
```

Correlazione per variabili quantitative

```
# seleziona solo variabili quantitative
```

```
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
##      LogPrice
## 1 7.200194
## 2 6.801216
## 3 6.354370
## 4 7.838915
## 5 7.497540
## 6 5.991465
```

```
var_numeric$X=NULL
```

```
# Matrice di correlazione
```

```
R<-cor(var_numeric)
```

```
R
```

```
##              Inches      Ram      Weight      Price Frequenza
## Inches      1.00000000 0.2379928 0.82763110 0.06819667 0.3078698
## Ram         0.23799280 1.0000000 0.38387409 0.74300714 0.3680005
## Weight      0.82763110 0.3838741 1.00000000 0.21036980 0.3204336
## Price       0.06819667 0.7430071 0.21036980 1.00000000 0.4302931
## Frequenza   0.30786980 0.3680005 0.32043359 0.43029310 1.0000000
## Pixel       -0.08639917 0.3963585 -0.04403379 0.51548639 0.1352935
## MemoriaSSD -0.12617118 0.4642349 -0.09500459 0.55288979 0.2482924
## TotalMemory 0.53805897 0.3489632 0.54952713 0.15783025 0.2421317
## LogPrice    0.04432871 0.6848033 0.15167383 0.92758068 0.5041461
##              Pixel  MemoriaSSD TotalMemory  LogPrice
## Inches      -0.08639917 -0.12617118 0.53805897 0.04432871
## Ram         0.39635848 0.46423485 0.34896315 0.68480333
## Weight      -0.04403379 -0.09500459 0.54952713 0.15167383
## Price       0.51548639 0.55288979 0.15783025 0.92758068
## Frequenza   0.13529350 0.24829236 0.24213174 0.50414608
## Pixel       1.00000000 0.36076909 0.06334134 0.48490475
## MemoriaSSD 0.36076909 1.00000000 -0.16285476 0.61685264
## TotalMemory 0.06334134 -0.16285476 1.00000000 0.15678005
## LogPrice    0.48490475 0.61685264 0.15678005 1.00000000
```

```
# Test di correlazione. (Spearsman's o Kendall tau)
```

```
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
```

```
#corrgram(var_numeric)
```

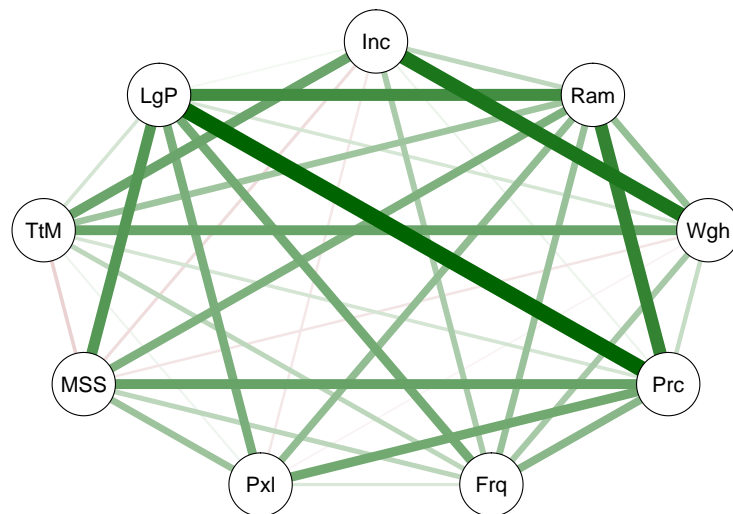
```
# Correlazione come grafo
```

```
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
## LogPrice     0.04 0.68  0.15 0.93      0.50  0.48      0.62
##
##      TotalMemory LogPrice
## Inches          0.54    0.04
## Ram              0.35    0.68
## Weight           0.55    0.15
## Price            0.16    0.93
## Frequenza        0.24    0.50
## Pixel            0.06    0.48
## MemoriaSSD      -0.16    0.62
## TotalMemory      1.00    0.16
## LogPrice         0.16    1.00
```

```
# plot corr matrix: green positive red negative
```

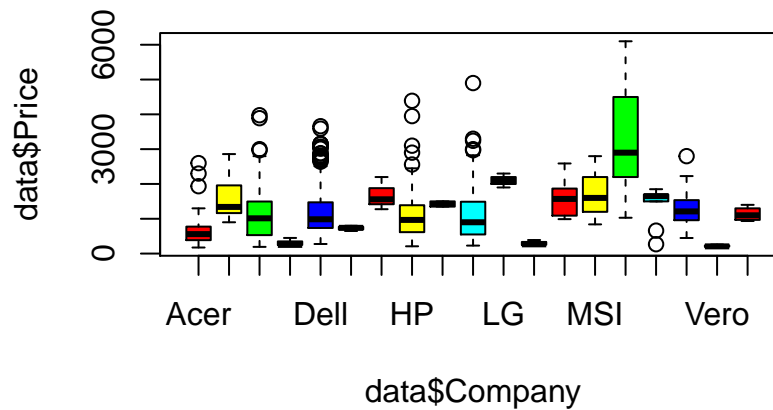
```
qgraph(detcor, shape="circle", posCol="darkgreen", negCol="darkred")
```

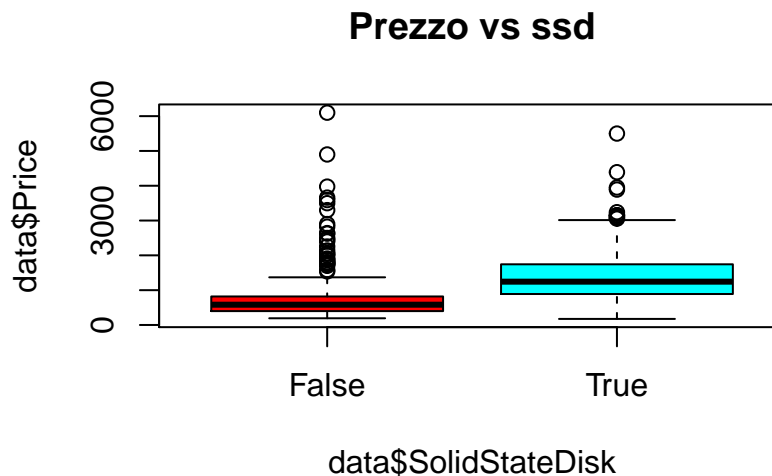
Boxplot di confronto (pre-anova)

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

Boxplot Prezzo per compagnia



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



Anova one way $Y = X_j$, for a categorical X

A una via

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



```
##
## Call:
## lm(formula = Price ~ SolidStateDisk, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1180.9  -375.9  -132.5   237.0  5377.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)       722.00     28.84   25.03  <2e-16 ***
## SolidStateDiskTrue  632.89     36.20   17.48  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 629.3 on 1301 degrees of freedom
## Multiple R-squared:  0.1902, Adjusted R-squared:  0.1896
## F-statistic: 305.6 on 1 and 1301 DF, p-value: < 2.2e-16
drop1(lmA, test = 'F')

## Single term deletions
##
## Model:
## Price ~ SolidStateDisk
##              Df Sum of Sq      RSS   AIC F value    Pr(>F)
## <none>                 515163583 16796
## SolidStateDisk  1 121011379 636174961 17069   305.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```

anova(lmA)

## Analysis of Variance Table
##
## Response: Price
##           Df      Sum Sq   Mean Sq F value    Pr(>F)
## SolidStateDisk    1 121011379 121011379   305.6 < 2.2e-16 ***
## Residuals      1301  515163583    395975
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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

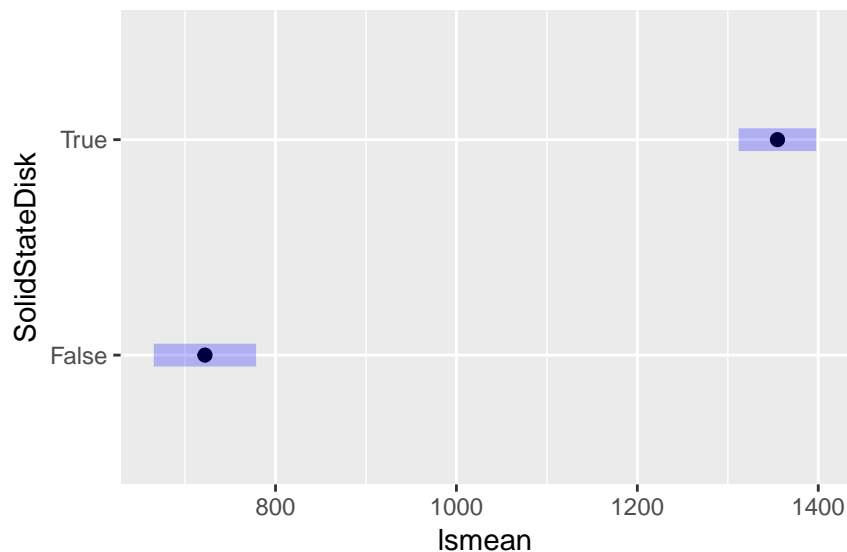
## contrast      estimate      SE    df t.ratio p.value
## False - True      -633 36.2 1301  -17.482 <.0001

ls_SolidStateDisk$lsmeans

## SolidStateDisk lsmean      SE    df lower.CL upper.CL
## False           722 28.8 1301      665      779
## True           1355 21.9 1301     1312     1398
##
## Confidence level used: 0.95

plot(ls_SolidStateDisk$lsmeans, alpha = .05)

```



```
str(data)
```



```

## 'data.frame':    1303 obs. of  23 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 3 ...
## $ TypeName       : Factor w/ 6 levels "2 in 1 Convertible",...: 5 5 4 5 5 4 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 5 ...
## $ Ram            : num  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 1
## $ 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 1
## $ 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 ...
## $ LogPrice : num 7.2 6.8 6.35 7.84 7.5 ...
```

```
lm_gpu_test=lm(Price~GpuCompany, data = data) #just a try
summary(lm_gpu_test)
```

```
##
```

```
## Call:
```

```
lm(formula = Price ~ GpuCompany, data = data)
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1030.9  -489.6  -140.9   367.8  4609.1
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    775.65     48.54  15.980 < 2e-16 ***
## GpuCompanyARM  -116.65    653.03  -0.179   0.858
## GpuCompanyIntel  232.58     54.25   4.287 1.95e-05 ***
## GpuCompanyNvidia 714.23     58.45  12.220 < 2e-16 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 651.2 on 1299 degrees of freedom
```

```
## Multiple R-squared:  0.1341, Adjusted R-squared:  0.1321
```

```
## F-statistic: 67.03 on 3 and 1299 DF, p-value: < 2.2e-16
```

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

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## Price ~ GpuCompany
```

```
##              Df Sum of Sq      RSS      AIC F value    Pr(>F)
## <none>                550892912 16888
## GpuCompany  3  85282050 636174961 17069   67.031 < 2.2e-16 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm_gpu_test)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: Price
##           Df      Sum Sq Mean Sq F value    Pr(>F)
## GpuCompany   3  85282050 28427350  67.031 < 2.2e-16 ***
## Residuals 1299 550892912   424090
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(data$dedicated_GPU)
```

```
## False True
##    723   580
```

```
lm_gpu_test=lm(Price~dedicated_GPU, data = data) #FIXME: seems not really worth it
summary(lm_gpu_test)
```

```
##
## Call:
## lm(formula = Price ~ dedicated_GPU, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1069.2   -523.2   -169.2    391.3   4830.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1007.74      25.56   39.43 < 2e-16 ***
## dedicated_GPUPrue    260.48      38.30    6.80 1.59e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 687.2 on 1301 degrees of freedom
## Multiple R-squared:  0.03432,    Adjusted R-squared:  0.03358
## F-statistic: 46.24 on 1 and 1301 DF,  p-value: 1.59e-11
```

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

```
## Single term deletions
##
## Model:
## Price ~ dedicated_GPU
##           Df Sum of Sq      RSS   AIC F value    Pr(>F)
## <none>                 614339895 17026
## dedicated_GPU  1  21835067 636174961 17069  46.241 1.59e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm_gpu_test)
```

```
## Analysis of Variance Table
##
## Response: Price
##           Df      Sum Sq Mean Sq F value    Pr(>F)
## dedicated_GPU   1  21835067 21835067  46.241 1.59e-11 ***
## Residuals    1301 614339895   472206
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmB = lm(Price ~ Company, data=data)
summary(lmB)

##
## Call:
## lm(formula = Price ~ Company, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2317.1  -452.8  -127.4   288.5  3812.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      626.78      63.43   9.881 < 2e-16 ***
## CompanyApple      937.42     154.14   6.082 1.57e-09 ***
## CompanyAsus       477.39      81.53   5.856 6.03e-09 ***
## CompanyChuwi     -312.48     377.06  -0.829 0.407416
## CompanyDell       559.29      73.62   7.597 5.80e-14 ***
## CompanyFujitsu    102.22     377.06   0.271 0.786352
## CompanyGoogle    1050.89     377.06   2.787 0.005397 **
## CompanyHP         441.00      74.41   5.927 3.96e-09 ***
## CompanyHuawei      797.22     459.62   1.735 0.083065 .
## CompanyLenovo     459.61      73.62   6.243 5.81e-10 ***
## CompanyLG        1472.22     377.06   3.904 9.93e-05 ***
## CompanyMediacom  -331.78     251.46  -1.319 0.187270
## CompanyMicrosoft  985.53     270.37   3.645 0.000278 ***
## CompanyMSI       1102.13     108.16  10.190 < 2e-16 ***
## CompanyRazer     2719.37     251.46  10.814 < 2e-16 ***
## CompanySamsung    786.67     223.77   3.515 0.000454 ***
## CompanyToshiba    641.04     112.51   5.698 1.50e-08 ***
## CompanyVero      -409.35     328.08  -1.248 0.212365
## CompanyXiaomi     506.69     328.08   1.544 0.122740
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 643.8 on 1284 degrees of freedom
## Multiple R-squared:  0.1635, Adjusted R-squared:  0.1518
## F-statistic: 13.94 on 18 and 1284 DF,  p-value: < 2.2e-16
```

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

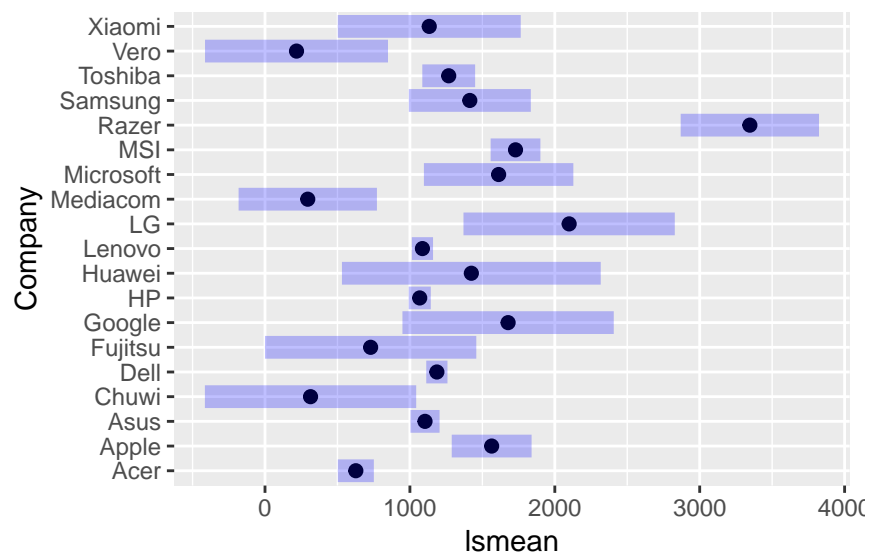
## Single term deletions
##
## Model:
## Price ~ Company
##      Df Sum of Sq      RSS   AIC F value    Pr(>F)
## <none>                 532160971 16873
## Company 18 104013991 636174961 17069  13.943 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lmB)

## Analysis of Variance Table
##
```

```
## Response: Price
##           Df      Sum Sq Mean Sq F value    Pr(>F)
## Company      18 104013991 5778555  13.943 < 2.2e-16 ***
## Residuals 1284 532160971  414456
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(lsmmeans)
ls_Company = lsmmeans(lmB, pairwise ~ Company, adjust = 'tukey')
#ls_Company$contrasts #FIXME: too long to be printed
#ls_Company$lsmmeans #mFIXME: aybe only the plot is enough?
plot(ls_Company$lsmmeans, alpha = .05)
```



```
lmB_agg = lm(Price ~ Aggregated_Company, data=data) #seems to be fine
library(lmB_agg)
```

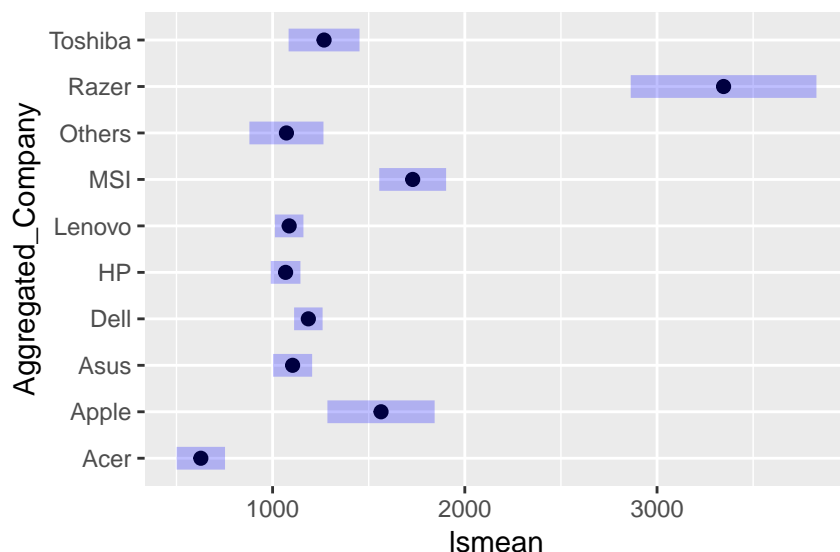
```
##
## Call:
## lm(formula = Price ~ Aggregated_Company, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2317.1  -465.0  -137.4   312.9  3812.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      626.78      64.19   9.765 < 2e-16 ***
## Aggregated_CompanyApple      937.42     155.98   6.010 2.41e-09 ***
## Aggregated_CompanyAsus       477.39      82.50   5.787 9.00e-09 ***
## Aggregated_CompanyDell       559.29      74.49   7.508 1.11e-13 ***
## Aggregated_CompanyHP         441.00      75.29   5.857 5.96e-09 ***
## Aggregated_CompanyLenovo      459.61      74.49   6.170 9.12e-10 ***
## Aggregated_CompanyMSI       1102.13     109.45  10.070 < 2e-16 ***
## Aggregated_CompanyOthers      445.30      117.32   3.795 0.000154 ***
## Aggregated_CompanyRazer      2719.37     254.45  10.687 < 2e-16 ***
## Aggregated_CompanyToshiba      641.04     113.85   5.631 2.20e-08 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 651.4 on 1293 degrees of freedom
## Multiple R-squared:  0.1375, Adjusted R-squared:  0.1315
## F-statistic: 22.9 on 9 and 1293 DF,  p-value: < 2.2e-16
drop1(lmB_agg, test = 'F')

## Single term deletions
##
## Model:
## Price ~ Aggregated_Company
##              Df Sum of Sq      RSS   AIC F value    Pr(>F)
## <none>                    548712016 16895
## Aggregated_Company  9  87462945 636174961 17069    22.9 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(lmB_agg)
```

```
## Analysis of Variance Table
##
## Response: Price
##              Df      Sum Sq Mean Sq F value    Pr(>F)
## Aggregated_Company  9  87462945 9718105    22.9 < 2.2e-16 ***
## Residuals        1293 548712016  424371
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ls_Company_agg = lsmeans(lmB_agg, pairwise ~ Aggregated_Company, adjust = 'tukey')
#ls_Company_agg$contrasts #FIXME: too long to be printed
#ls_Company_agg$lsmeans #mFIXME: aybe only the plot is enough?
plot(ls_Company_agg$lsmeans, alpha = .05) #i guess from here seems fine to leave "Razer" alone?
```



```
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
```

```
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
```

```
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
```

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

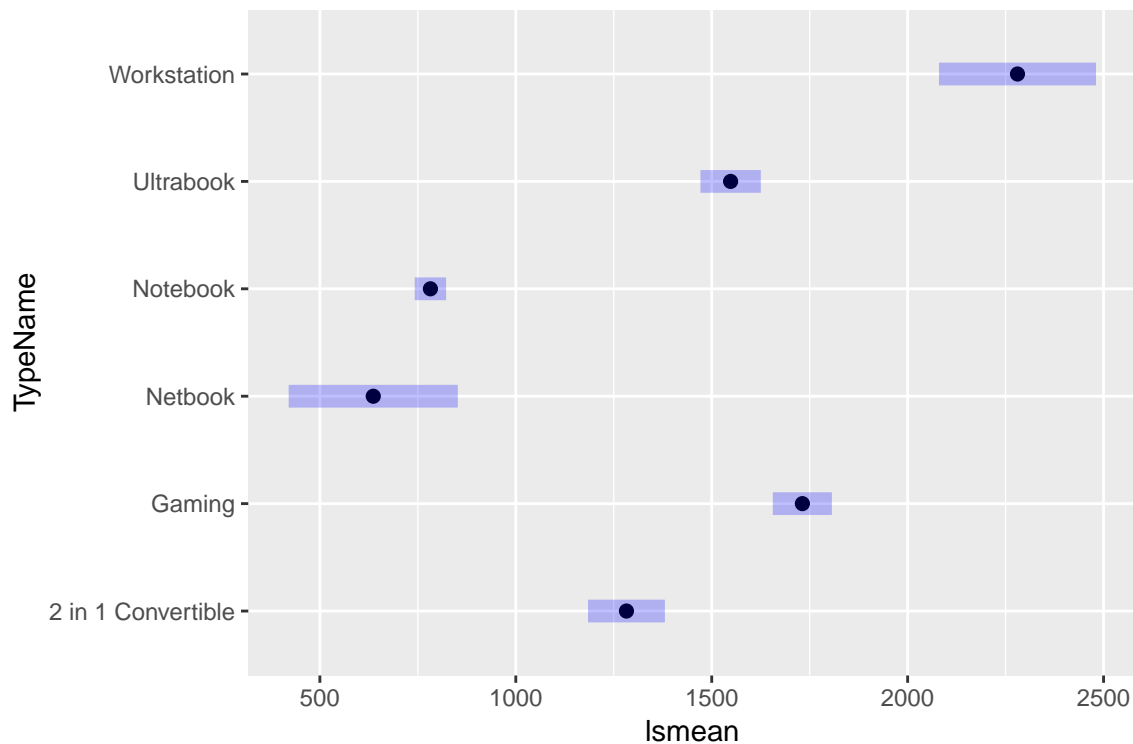
```
## contrast              estimate      SE    df t.ratio p.value
## 2 in 1 Convertible - Gaming      -449  63.1 1297  -7.119 <.0001
## 2 in 1 Convertible - Netbook      646 120.9 1297   5.347 <.0001
## 2 in 1 Convertible - Notebook     500  54.0 1297   9.263 <.0001
## 2 in 1 Convertible - Ultrabook    -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
```

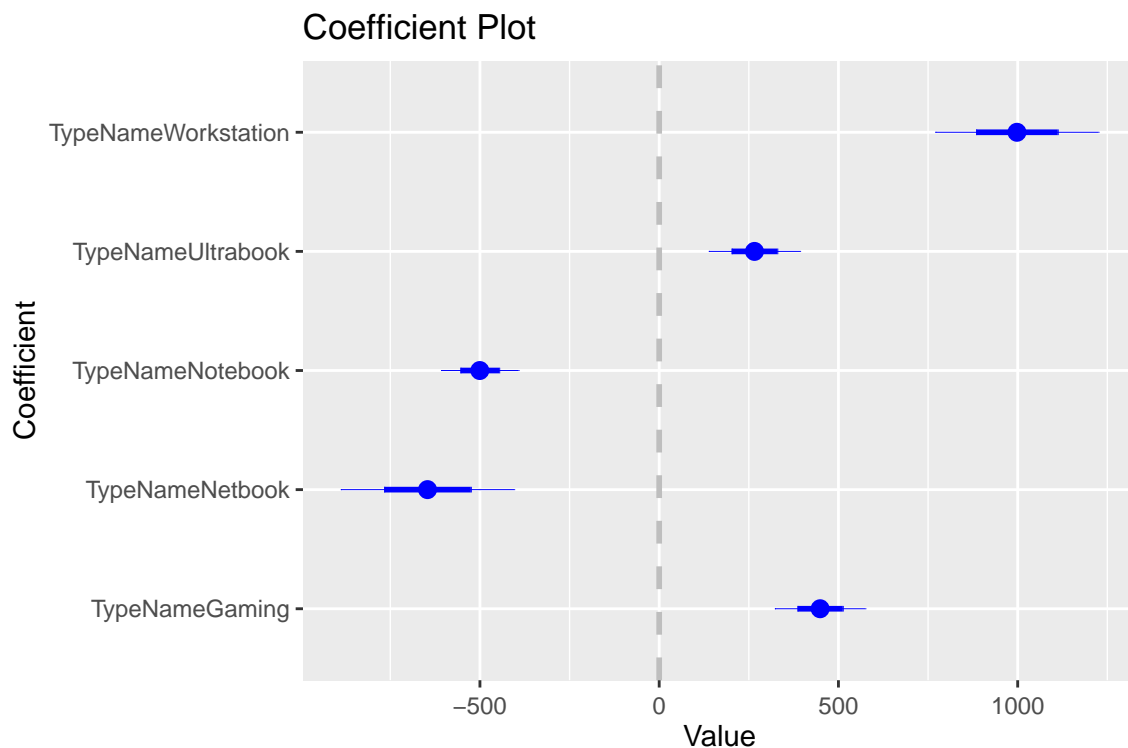
```
ls_TypeName$lsmeans
```

```
## 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
```

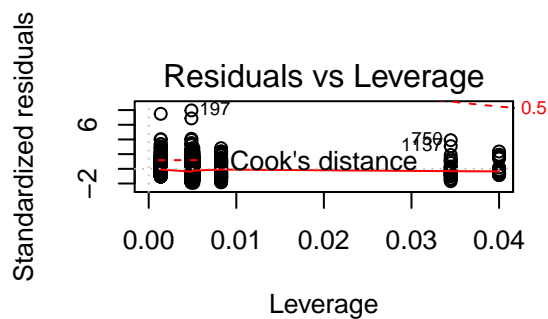
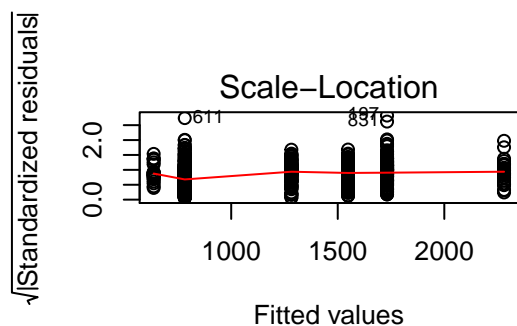
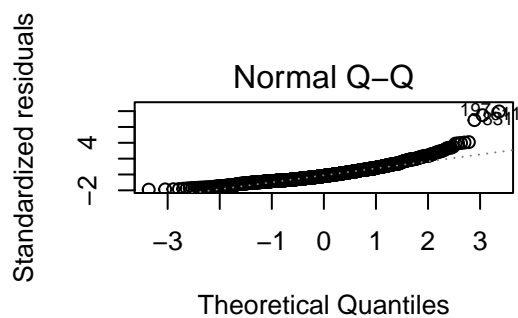
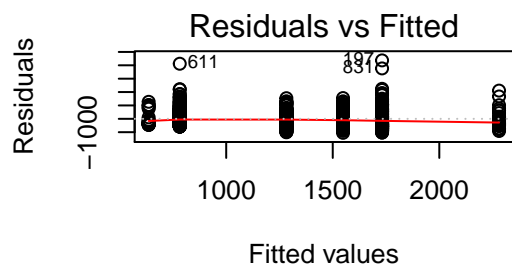
```
plot(ls_TypeName$lsmeans, alpha = .05)
```



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

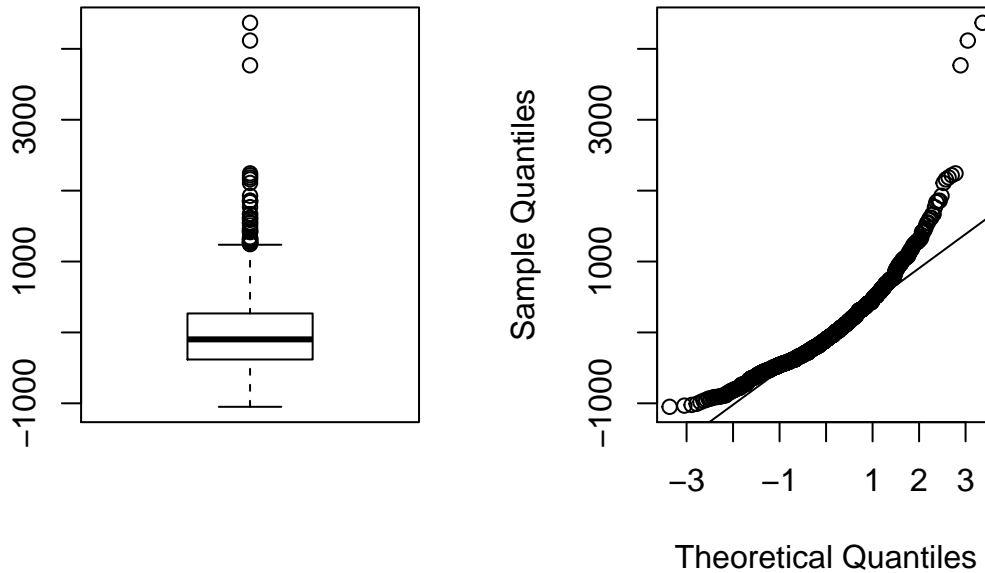


```
par(mfrow = c(2,2))
plot(lmC)
```



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

Normal Q-Q Plot



```
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
```

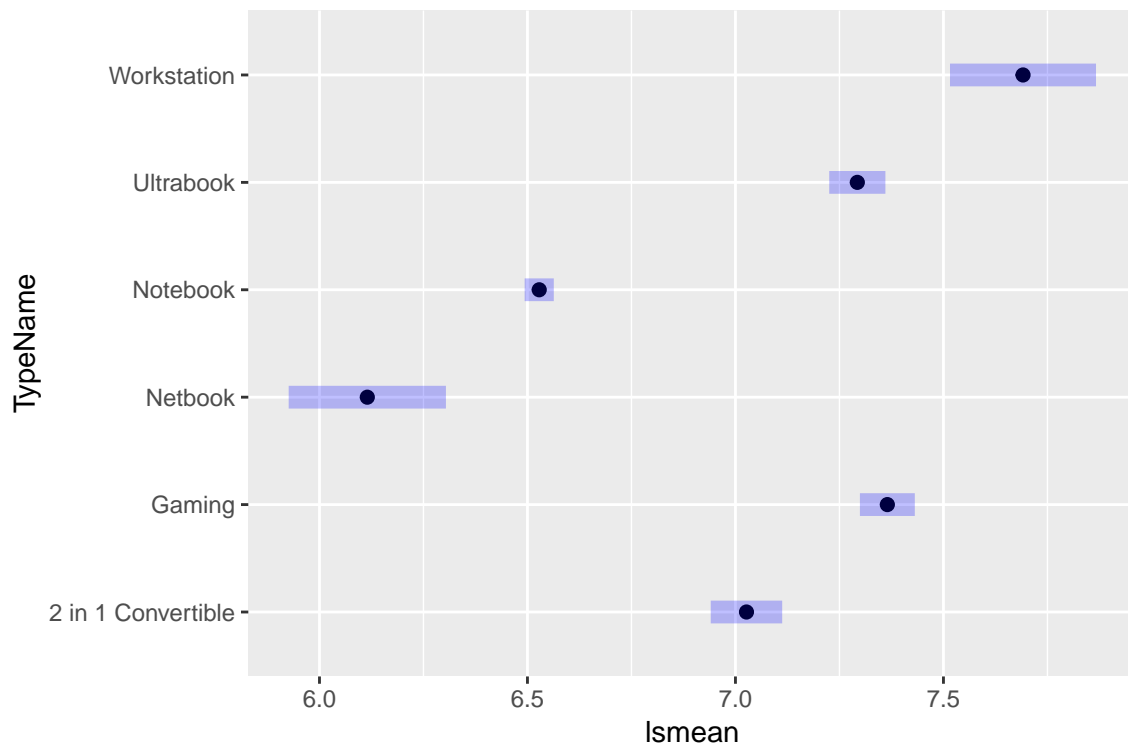
```
ls_TypeName_log = lsmeans(lmC_log, pairwise ~ TypeName, adjust = 'tukey')
ls_TypeName_log$contrasts
```

```
## 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
```

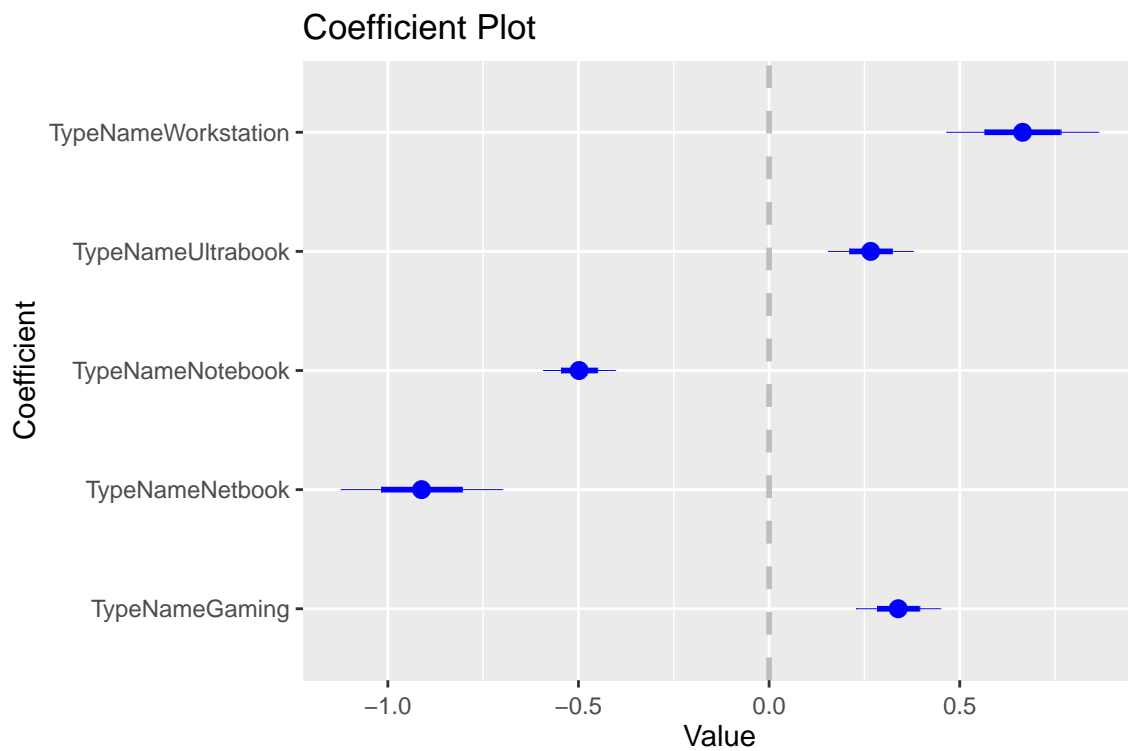
```
ls_TypeName_log$lsmeans
```

```
##   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
```

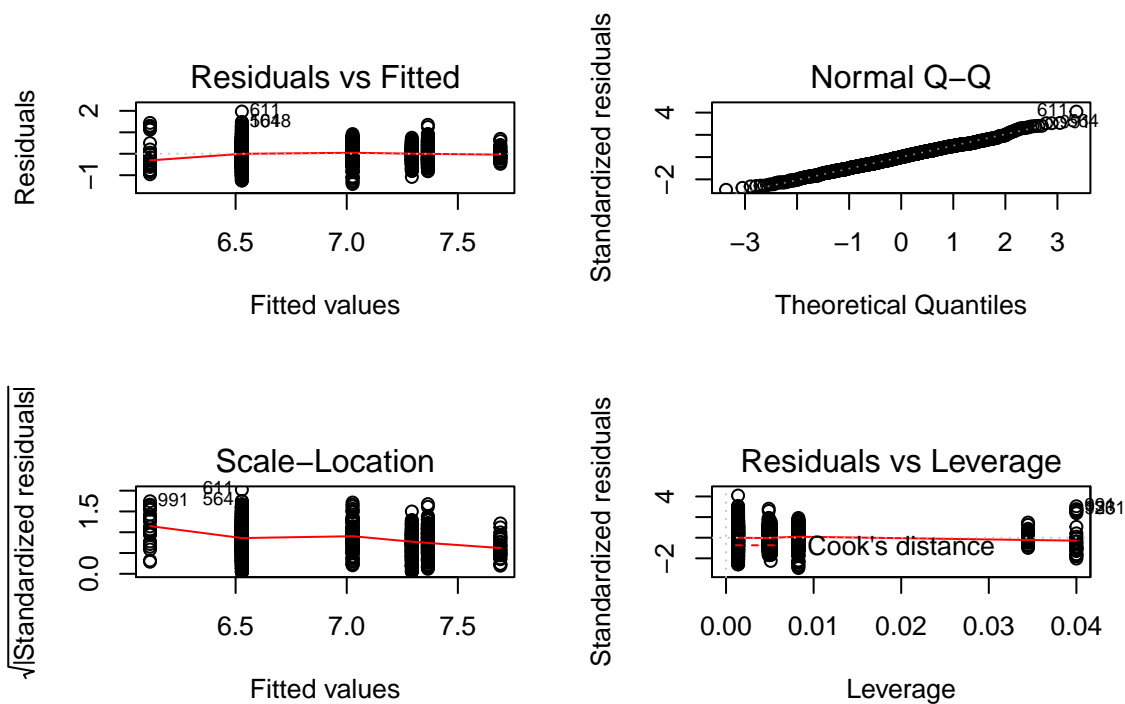
```
plot(ls_TypeName_log$lsmeans, alpha = .05)
```



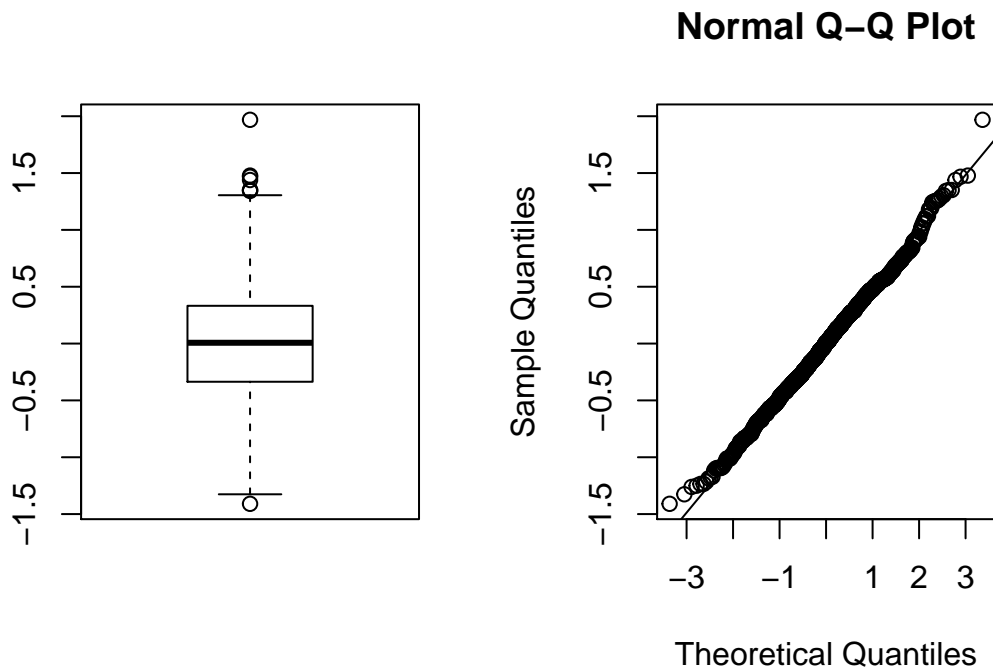
```
coefplot(lmC_log, intercept = FALSE)
```



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



```
 #(not) normal distribution of residuals
par(mfrow=c(1,2))
boxplot(lmC_log$residuals)
qqnorm(lmC_log$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 $Y = X_j X_k$ for some categorical X

A due vie

```
# Con interazione
```

```
lmC = lm(Price ~ Company*TypeName , data=data)
```

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

```
## Single term deletions
```

```
##
```

```
## Model:
```

```
## Price ~ Company * TypeName
```

```
##           Df Sum of Sq      RSS   AIC F value    Pr(>F)
```

```
## <none>                 320739568 16273
```

```
## Company:TypeName 25  29159364 349898932 16336  4.5602 1.181e-12 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



```
#summary(lmC) #FIXME: too long to be printed
```

```
lmC = lm(Price ~ Company+TypeName , data=data)
```

```
# type I effects A, B/A C/A,B
```

```
anova(lmC)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: Price
```

```
##           Df      Sum Sq Mean Sq F value    Pr(>F)
```

```
## Company      18 104013991  5778555  21.123 < 2.2e-16 ***
```

```
## TypeName       5 182262038 36452408 133.246 < 2.2e-16 ***
```

```
## Residuals 1279 349898932   273572
```

```
## ---
```

```
## 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:
```

```
## Price ~ Company + TypeName
```

```
##           Df Sum of Sq      RSS      AIC F value    Pr(>F)
```

```
## <none>                 349898932 16336
```

```
## Company      18 42619448 392518380 16450   8.6549 < 2.2e-16 ***
```

```
## TypeName       5 182262038 532160971 16873 133.2460 < 2.2e-16 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lmC)
```

```
##
```

```
## Call:
```

```
## lm(formula = Price ~ Company + TypeName, data = data)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -2147.6  -343.2   -81.9    243.1   4081.9
```

```
##
```

```
## Coefficients:
```

```
##           Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      991.52      69.88  14.189 < 2e-16 ***
```

```
## CompanyApple      383.70     132.62   2.893  0.00388 **
```

```
## CompanyAsus       168.81      67.79   2.490  0.01290 *
```

```
## CompanyChuwi     -180.68     306.47  -0.590  0.55559
```

```
## CompanyDell       350.73      60.52   5.796 8.56e-09 ***
```

```
## CompanyFujitsu    234.02     306.47   0.764  0.44525
```

```
## CompanyGoogle     497.17     309.44   1.607  0.10837
```

```
## CompanyHP         337.48      60.85   5.546 3.55e-08 ***
```

```
## CompanyHuawei      243.50     375.96   0.648  0.51731
```

```
## CompanyLenovo     322.12      60.24   5.348 1.05e-07 ***
```

```
## CompanyLG         918.50     309.44   2.968  0.00305 **
```

```
## CompanyMediacom   -270.91     204.43  -1.325  0.18534
```

```
## CompanyMicrosoft  431.81     223.95   1.928  0.05406 .
```

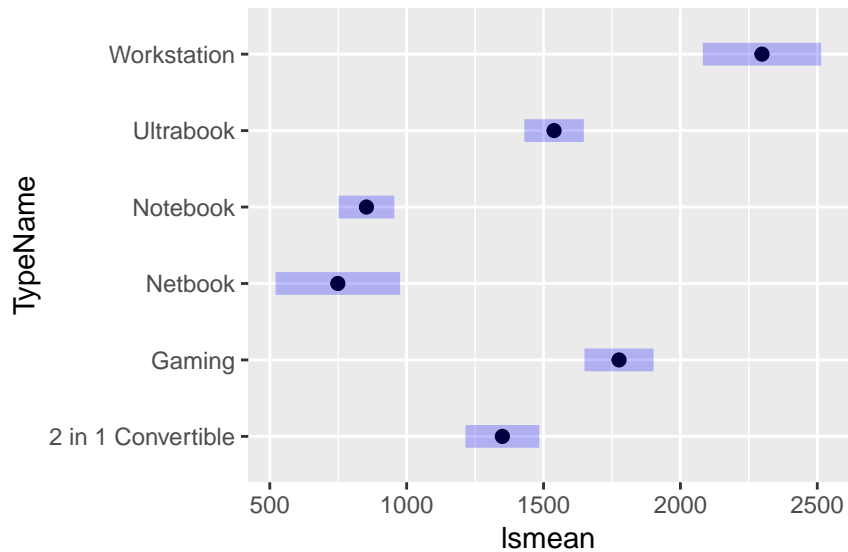
```
## CompanyMSI          311.10      98.62    3.155  0.00165 **
## CompanyRazer        1996.14     207.24    9.632  < 2e-16 ***
## CompanySamsung       438.88     183.82    2.388  0.01710 *
## CompanyToshiba       601.45      92.12    6.529  9.52e-11 ***
## CompanyVero          -277.55     266.70   -1.041  0.29821
## CompanyXiaomi         295.72     267.40    1.106  0.26896
## TypeNameGaming       426.29      65.51    6.507  1.10e-10 ***
## TypeNameNetbook      -600.94     115.75   -5.192  2.42e-07 ***
## TypeNameNotebook     -496.54      51.98   -9.552  < 2e-16 ***
## TypeNameUltrabook     188.98      63.81    2.962  0.00312 **
## TypeNameWorkstation   948.46     109.22    8.684  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 523 on 1279 degrees of freedom
## Multiple R-squared:  0.45, Adjusted R-squared:  0.4401
## F-statistic: 45.5 on 23 and 1279 DF, p-value: < 2.2e-16
```

```
# contrasti
library(lsmeans)
ls=lsmeans(lmC, #FIXME: @Andrea, c'era lmB ma credo tu volessi scrivere lmC, in case check it
            pairwise ~ TypeName ,
            adjust="tukey")
ls$lsmeans
```

```
## TypeName      lsmean    SE    df lower.CL upper.CL
## 2 in 1 Convertible  1350  68.9 1279    1214    1485
## Gaming            1776  64.4 1279    1649    1902
## Netbook           749 115.9 1279     521     976
## Notebook          853  52.0 1279     751     955
## Ultrabook         1538  55.5 1279    1430    1647
## Workstation       2298 110.1 1279    2082    2514
##
```

```
## Results are averaged over the levels of: Company
## Confidence level used: 0.95
```

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



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

```
## contrast          estimate    SE    df t.ratio p.value
## 2 in 1 Convertible - Gaming      -426  65.5 1279  -6.507 <.0001
## 2 in 1 Convertible - Netbook       601 115.7 1279   5.192 <.0001
## 2 in 1 Convertible - Notebook      497  52.0 1279   9.552 <.0001
## 2 in 1 Convertible - Ultrabook    -189  63.8 1279  -2.962 0.0367
## 2 in 1 Convertible - Workstation  -948 109.2 1279  -8.684 <.0001
## Gaming - Netbook                1027 114.5 1279   8.972 <.0001
## Gaming - Notebook                923  49.4 1279  18.671 <.0001
## Gaming - Ultrabook              237  61.1 1279   3.882 0.0015
## Gaming - Workstation            -522 108.3 1279  -4.820 <.0001
## Netbook - Notebook             -104 107.0 1279  -0.975 0.9258
## Netbook - Ultrabook            -790 113.3 1279  -6.969 <.0001
## Netbook - Workstation          -1549 143.8 1279 -10.774 <.0001
## Notebook - Ultrabook           -686  46.5 1279 -14.754 <.0001
## Notebook - Workstation         -1445  99.8 1279 -14.475 <.0001
## Ultrabook - Workstation        -759 106.4 1279  -7.138 <.0001
##
```

```
## Results are averaged over the levels of: Company
## P value adjustment: tukey method for comparing a family of 6 estimates
```

```
# if at least one contrast is significant, the variable
# is significant in the anova table # drop1 effects
```

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

```
## $lsmeans
## contrast          estimate    SE    df t.ratio p.value
## 2 in 1 Convertible effect     -77.7  47.9 1279  -1.623 0.1048
## Gaming effect                348.6  46.0 1279   7.583 <.0001
## Netbook effect              -678.6  90.2 1279  -7.521 <.0001
## Notebook effect             -574.2  31.8 1279 -18.032 <.0001
## Ultrabook effect             111.3  43.8 1279   2.542 0.0134
## Workstation effect           870.7  84.6 1279  10.287 <.0001
```

```
##
## Results are averaged over the levels of: Company
## P value adjustment: fdr method for 6 tests
##
## $contrasts
## contrast estimate SE df t.ratio
## 2 in 1 Convertible - Gaming effect -150.6 71.6 1279 -2.103
## 2 in 1 Convertible - Netbook effect 876.6 121.9 1279 7.192
## 2 in 1 Convertible - Notebook effect 772.2 51.2 1279 15.077
## 2 in 1 Convertible - Ultrabook effect 86.7 57.9 1279 1.498
## 2 in 1 Convertible - Workstation effect -672.8 74.0 1279 -9.093
## Gaming - Netbook effect 1302.9 123.6 1279 10.544
## Gaming - Notebook effect 1198.5 55.4 1279 21.649
## Gaming - Ultrabook effect 513.0 60.9 1279 8.416
## Gaming - Workstation effect -246.5 77.4 1279 -3.186
## Netbook - Notebook effect 171.2 107.0 1279 1.600
## Netbook - Ultrabook effect -514.3 110.5 1279 -4.655
## Netbook - Workstation effect -1273.7 119.6 1279 -10.649
## Notebook - Ultrabook effect -409.9 55.3 1279 -7.416
## Notebook - Workstation effect -1169.3 71.6 1279 -16.325
## Ultrabook - Workstation effect -483.8 84.4 1279 -5.730
## p.value
## 0.0411
## <.0001
## <.0001
## 0.1345
## <.0001
## <.0001
## <.0001
## <.0001
## <.0001
## 0.0018
## 0.1177
## <.0001
## <.0001
## <.0001
## <.0001
## <.0001
##
## Results are averaged over the levels of: Company
## P value adjustment: fdr method for 15 tests
```

```
library(coefplot)
coefplot(lmC, intercept=FALSE) #FIXME: @Andrea, same goes here
```



ANOVA k way

```
lmK = lm(Price ~ Company+TypeName+SolidStateDisk , data=data)
summary(lmK)
```

```
##
## Call:
## lm(formula = Price ~ Company + TypeName + SolidStateDisk, data = data)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-2220.9	-304.8	-66.1	212.1	4268.4

```
##
## Coefficients:
```

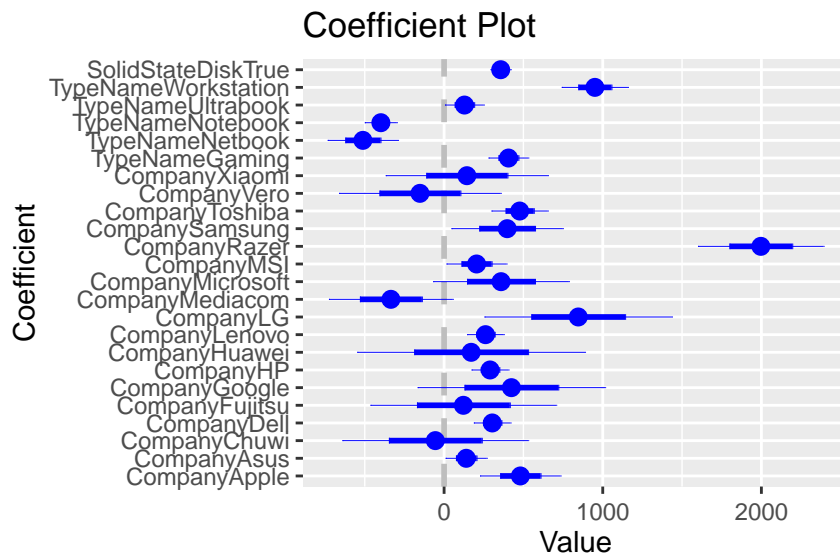
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	767.82	69.59	11.034	< 2e-16 ***
CompanyApple	480.95	126.85	3.792	0.000157 ***
CompanyAsus	139.64	64.74	2.157	0.031210 *
CompanyChuwi	-55.11	292.67	-0.188	0.850666
CompanyDell	303.66	57.90	5.244	1.83e-07 ***
CompanyFujitsu	121.68	292.63	0.416	0.677611
CompanyGoogle	424.48	295.36	1.437	0.150913
CompanyHP	290.08	58.22	4.982	7.14e-07 ***
CompanyHuawei	170.81	358.83	0.476	0.634132
CompanyLenovo	261.18	57.74	4.524	6.64e-06 ***
CompanyLG	845.81	295.36	2.864	0.004256 **
CompanyMediacom	-335.25	195.17	-1.718	0.086087 .
CompanyMicrosoft	359.12	213.81	1.680	0.093275 .
CompanyMSI	204.88	94.58	2.166	0.030485 *
CompanyRazer	1996.74	197.77	10.096	< 2e-16 ***
CompanySamsung	397.79	175.45	2.267	0.023539 *
CompanyToshiba	476.72	88.60	5.380	8.83e-08 ***
CompanyVero	-151.98	254.75	-0.597	0.550877
CompanyXiaomi	143.73	255.53	0.562	0.573878
TypeNameGaming	405.95	62.54	6.491	1.22e-10 ***
TypeNameNetbook	-511.64	110.74	-4.620	4.22e-06 ***
TypeNameNotebook	-398.41	50.36	-7.910	5.52e-15 ***

```
## TypeNameUltrabook      128.51      61.13    2.102 0.035727 *
## TypeNameWorkstation    950.57     104.22    9.120 < 2e-16 ***
## SolidStateDiskTrue     356.87      31.73   11.248 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 499.1 on 1278 degrees of freedom
## Multiple R-squared:  0.4995, Adjusted R-squared:  0.4901
## F-statistic: 53.15 on 24 and 1278 DF,  p-value: < 2.2e-16
```

```
drop1(lmK, test="F") # type III SS
```

```
## Single term deletions
##
## Model:
## Price ~ Company + TypeName + SolidStateDisk
##           Df Sum of Sq      RSS   AIC  F value    Pr(>F)
## <none>                 318382025 16216
## Company          18  38122976 356505001 16327    8.5015 < 2.2e-16 ***
## TypeName           5 124171618 442553644 16634   99.6861 < 2.2e-16 ***
## SolidStateDisk     1  31516907 349898932 16336  126.5103 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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



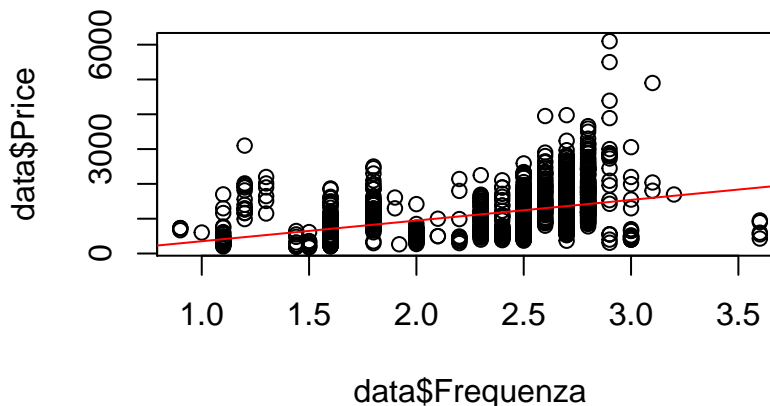
Regressione lineare

```
lmA<-lm(Price ~ Frequenza , data=data)
summary(lmA)
```

```
##
## Call:
## lm(formula = Price ~ Frequenza, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1467.6  -453.8  -119.6   327.6  4618.2
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -241.84      81.32  -2.974   0.003 **
## Frequenza     594.02     34.55  17.194 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 631.2 on 1301 degrees of freedom
## Multiple R-squared:  0.1852, Adjusted R-squared:  0.1845
## F-statistic: 295.6 on 1 and 1301 DF, p-value: < 2.2e-16
```

```
plot(data$Frequenza,data$Price)
abline(lmA,col="red")
```

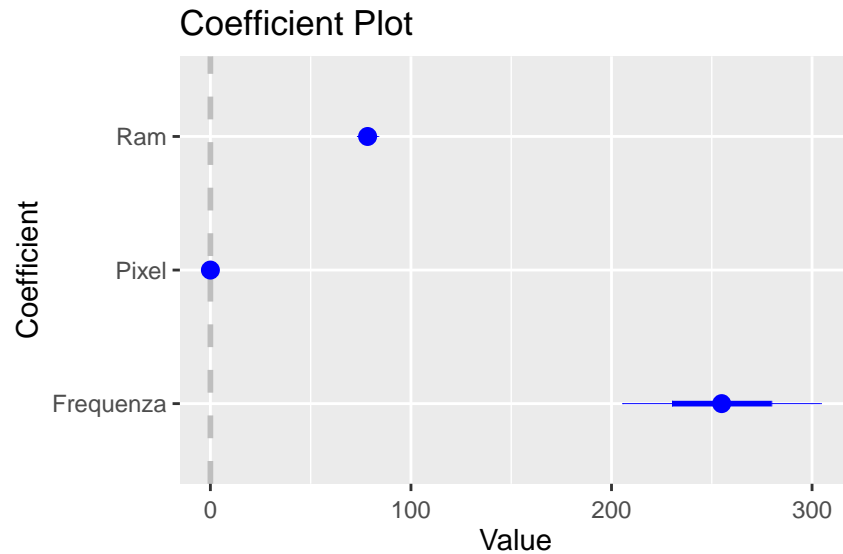


```
lmA<-lm(Price ~ Frequenza+Pixel+Ram , data=data)
summary(lmA)
```

```
##
## Call:
## lm(formula = Price ~ Frequenza + Pixel + Ram, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1785.72  -257.23   -66.06   191.11  2791.53
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.076e+02  5.547e+01  -7.349 3.52e-13 ***
## Frequenza    2.549e+02  2.474e+01  10.306 < 2e-16 ***
## Pixel        1.329e-04  9.117e-06  14.575 < 2e-16 ***
## Ram          7.839e+01  2.658e+00  29.488 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 420.2 on 1299 degrees of freedom
## Multiple R-squared:  0.6395, Adjusted R-squared:  0.6386
## F-statistic: 768 on 3 and 1299 DF, p-value: < 2.2e-16
```



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



Ancova Y = all covariates (qualitative +quantitative)

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

```
##
## Call:
## lm(formula = Price ~ Company + TypeName + SolidStateDisk + Frequenza +
##     Pixel + Ram, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1881.48  -214.66   -31.28   165.08  1905.88
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.544e+02  6.752e+01  -2.286  0.02240 *
## CompanyApple    2.629e+02  9.105e+01   2.888  0.00394 **
## CompanyAsus     5.442e+01  4.625e+01   1.177  0.23950
## CompanyChuwi   -9.371e+01  2.098e+02  -0.447  0.65521
## CompanyDell     1.134e+02  4.166e+01   2.721  0.00660 **
## CompanyFujitsu  6.262e+01  2.087e+02   0.300  0.76421
## CompanyGoogle   3.043e+02  2.122e+02   1.434  0.15186
## CompanyHP       2.073e+02  4.167e+01   4.975  7.41e-07 ***
## CompanyHuawei    5.645e+01  2.559e+02   0.221  0.82547
## CompanyLenovo   1.301e+02  4.141e+01   3.141  0.00172 **
## CompanyLG       6.774e+02  2.107e+02   3.216  0.00133 **
## CompanyMediacom -9.928e+01  1.403e+02  -0.708  0.47932
## CompanyMicrosoft 2.374e+02  1.527e+02   1.554  0.12034
## CompanyMSI      2.116e+02  6.742e+01   3.139  0.00173 **
## CompanyRazer    1.105e+03  1.441e+02   7.673  3.32e-14 ***
## CompanySamsung  8.825e+01  1.256e+02   0.703  0.48241
```



```
## CompanyToshiba      2.941e+02  6.359e+01  4.625 4.12e-06 ***
## CompanyVero         5.760e+00  1.825e+02  0.032 0.97483
## CompanyXiaomi      -8.005e+00  1.823e+02 -0.044 0.96498
## TypeNameGaming     -3.957e+01  4.846e+01 -0.816 0.41443
## TypeNameNetbook    -1.126e+02  8.012e+01 -1.406 0.16010
## TypeNameNotebook   -2.507e+02  3.672e+01 -6.828 1.33e-11 ***
## TypeNameUltrabook   9.767e+01  4.373e+01  2.233 0.02570 *
## TypeNameWorkstation 7.122e+02  7.560e+01  9.420 < 2e-16 ***
## SolidStateDiskTrue  1.589e+02  2.341e+01  6.790 1.71e-11 ***
## Frequenza          1.742e+02  2.339e+01  7.448 1.74e-13 ***
## Pixel              8.775e-05  8.303e-06 10.568 < 2e-16 ***
## Ram                6.744e+01  2.568e+00 26.266 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 355.7 on 1275 degrees of freedom
## Multiple R-squared:  0.7464, Adjusted R-squared:  0.741
## F-statistic: 139 on 27 and 1275 DF, p-value: < 2.2e-16
```

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

```
## Single term deletions
##
## Model:
## Price ~ Company + TypeName + SolidStateDisk + Frequenza + Pixel +
##      Ram
##
##          Df Sum of Sq      RSS   AIC  F value    Pr(>F)
## <none>                161324026 15336
## Company          18 13792368 175116394 15406   6.0559 2.398e-14 ***
## TypeName           5 35840561 197164587 15587 56.6521 < 2.2e-16 ***
## SolidStateDisk     1  5833506 167157532 15380 46.1042 1.714e-11 ***
## Frequenza          1  7019435 168343461 15389 55.4770 1.736e-13 ***
## Pixel              1 14131083 175455109 15443 111.6829 < 2.2e-16 ***
## Ram                1  87292336 248616362 15897 689.9018 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
data$LogPrice=NULL
data$Product=NULL
data$X=NULL
str(data)
```

```
## 'data.frame': 1303 obs. of 20 variables:
## $ Company      : Factor w/ 19 levels "Acer","Apple",...: 2 2 8 2 2 1 2 2 3 1 ...
## $ 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 ...
## $ Ram          : num 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 1 ...
```

```
## $ 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 ...
## $ 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 ...
```

```
lm_full = lm(Price ~ ., data = data)
#summary(lm_full) #FIXME: wayyy too long to be printed, R^2 = 0.9586
anova(lm_full, test="F")
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: Price
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
## Company	18	104013991	5778555	114.6831	< 2.2e-16 ***
## TypeName	5	182262038	36452408	723.4463	< 2.2e-16 ***
## Inches	1	6163570	6163570	122.3242	< 2.2e-16 ***
## ScreenResolution	36	108074619	3002073	59.5801	< 2.2e-16 ***
## Cpu	110	95329933	866636	17.1995	< 2.2e-16 ***
## Ram	1	34947028	34947028	693.5700	< 2.2e-16 ***
## Memory	34	17103911	503056	9.9838	< 2.2e-16 ***
## Gpu	88	34266807	389396	7.7281	< 2.2e-16 ***
## OpSys	6	3524179	587363	11.6570	1.186e-12 ***
## Weight	1	944	944	0.0187	0.8911
## Residuals	1002	50487942	50387		

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

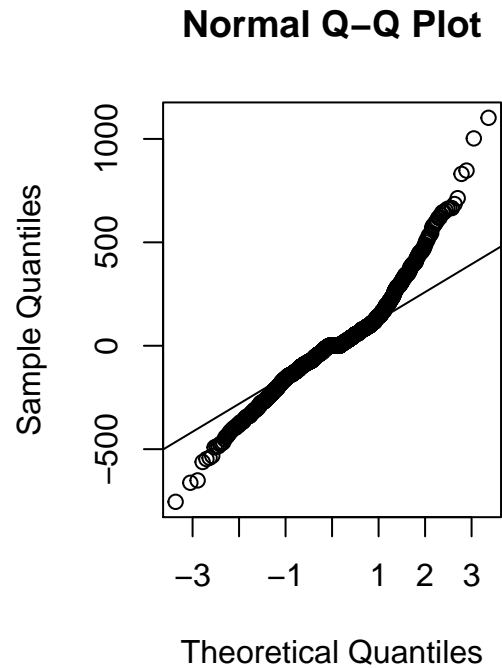
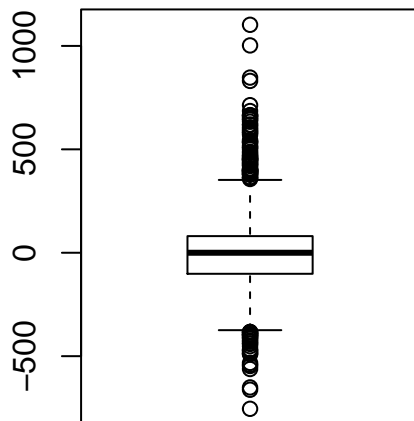
```
## Single term deletions
```

```
##
```

```
## Model:
```

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

	Df	Sum of Sq	RSS	AIC	F value	Pr(>F)
## <none>			50487942	14368		
## Company	6	648147	51136089	14373	2.1439	0.04621 *
## TypeName	5	3680356	54168297	14450	14.6083	7.662e-14 ***
## Inches	1	210771	50698713	14371	4.1830	0.04109 *
## ScreenResolution	23	5329402	55817344	14453	4.5987	7.463e-12 ***
## Cpu	88	16403980	66891922	14559	3.6995	< 2.2e-16 ***
## Ram	1	4479889	54967831	14477	88.9093	< 2.2e-16 ***
## Memory	31	10454540	60942481	14551	6.6930	< 2.2e-16 ***
## Gpu	86	29370368	79858310	14793	6.7778	< 2.2e-16 ***



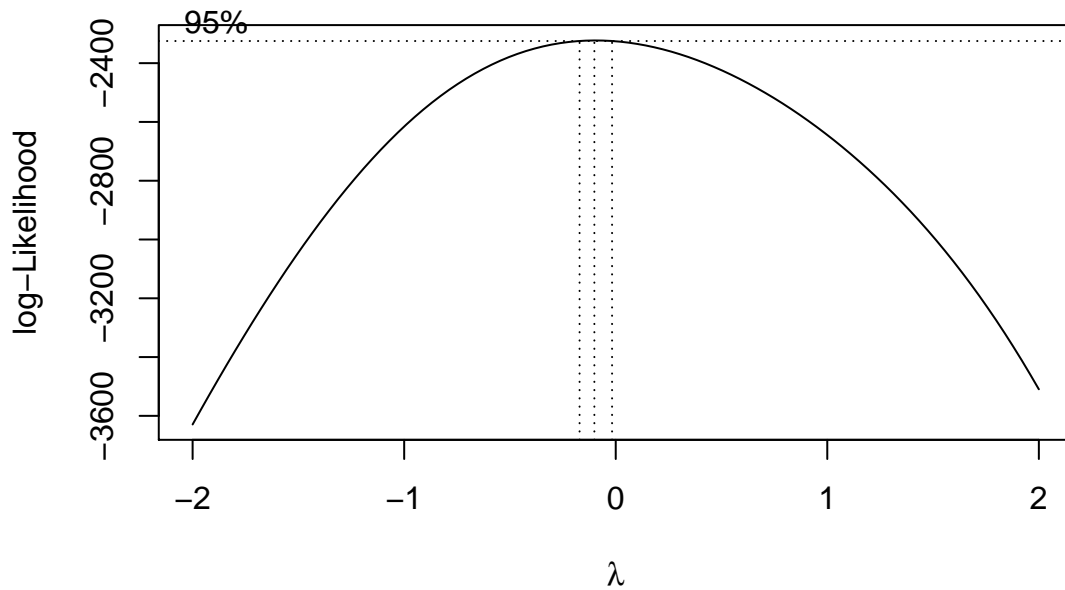
```
#tests
ad.test(lm_full$residuals)

##
##  Anderson-Darling normality test
##
## data:  lm_full$residuals
## A = 19.734, p-value < 2.2e-16

shapiro.test(lm_full$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  lm_full$residuals
## W = 0.94932, p-value < 2.2e-16

library(MASS)
#to justify log correction
boxcoxreg1<-boxcox(lm_full, plotit=T)
```



```
which.max(boxcoxreg1$y)
```

```
## [1] 48
```

```
lambda=boxcoxreg1$x[which.max(boxcoxreg1$y)]
```

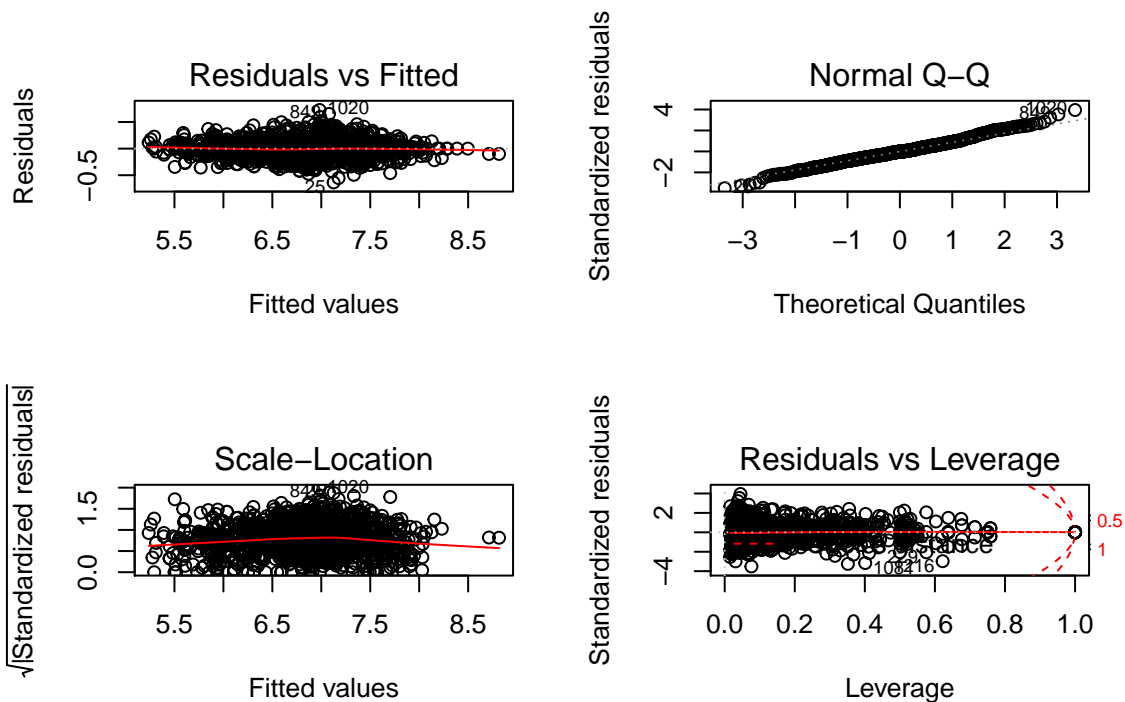
```
lambda #FIXME: not really 0, one should actually apply  $((y)^{\lambda} - 1) / \lambda$  but meh
```

```
## [1] -0.1010101
```

```
lm_full_t = lm(log(Price) ~ ., data = data)
```

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

```
plot(lm_full_t) #quite better
```



```
ad.test(lm_full_t$residuals) #not really
```

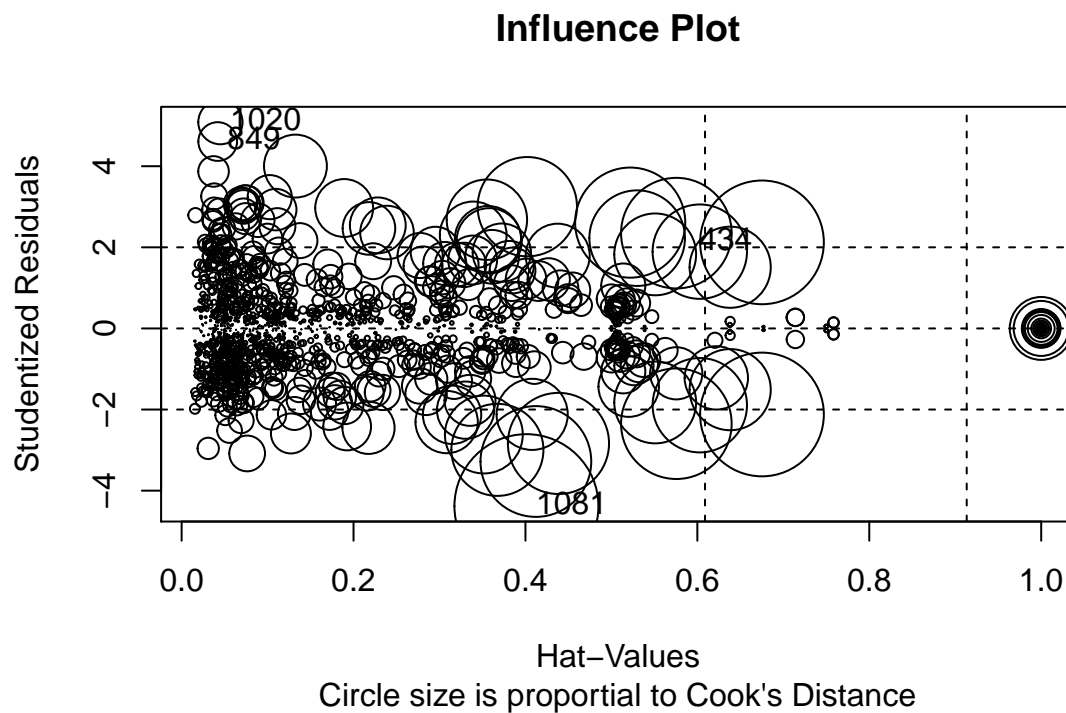
```
##  
## Anderson-Darling normality test  
##  
## data: lm_full_t$residuals  
## A = 7.3367, p-value < 2.2e-16
```

```
shapiro.test(lm_full_t$residuals) #not really
```

```
##  
## Shapiro-Wilk normality test  
##  
## data: lm_full_t$residuals  
## W = 0.98508, p-value = 2.573e-10
```

A look over outliers

```
library(car)  
influencePlot(lm_full,main="Influence Plot", sub="Circle size is proportional to Cook's Distance" )
```

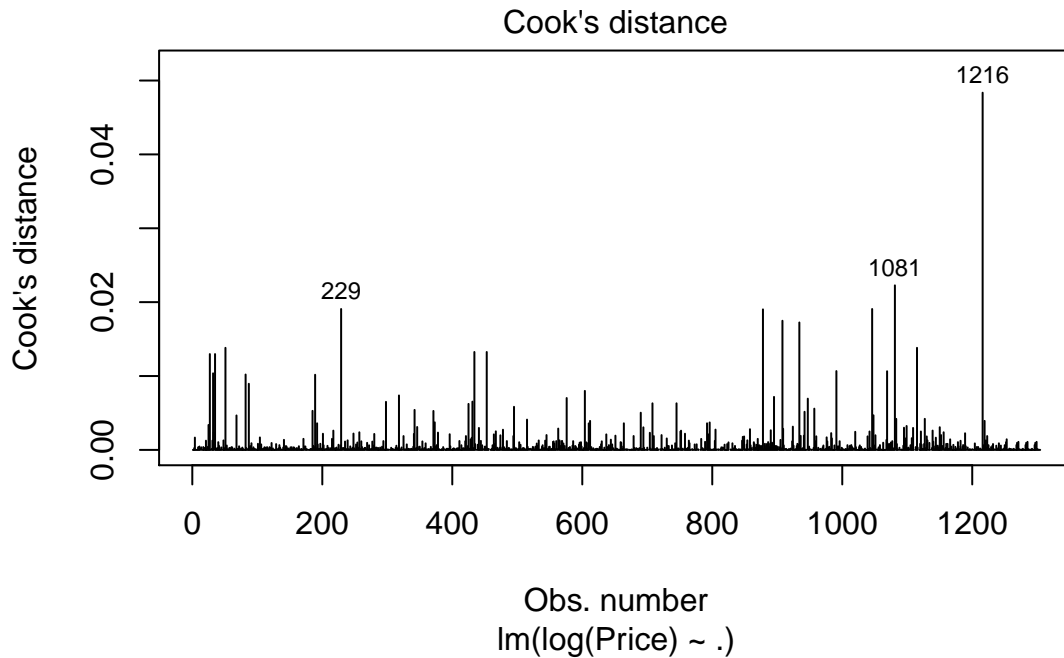


```
##      StudRes      Hat      CookD  
## 13      NaN 1.00000000      NaN  
## 15      NaN 1.00000000      NaN  
## 434  2.121962 0.67517279 0.030985293  
## 849  4.608379 0.04197115 0.003029829  
## 1020 5.086222 0.04512122 0.003962863  
## 1081 -4.382893 0.40114339 0.041986565
```

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

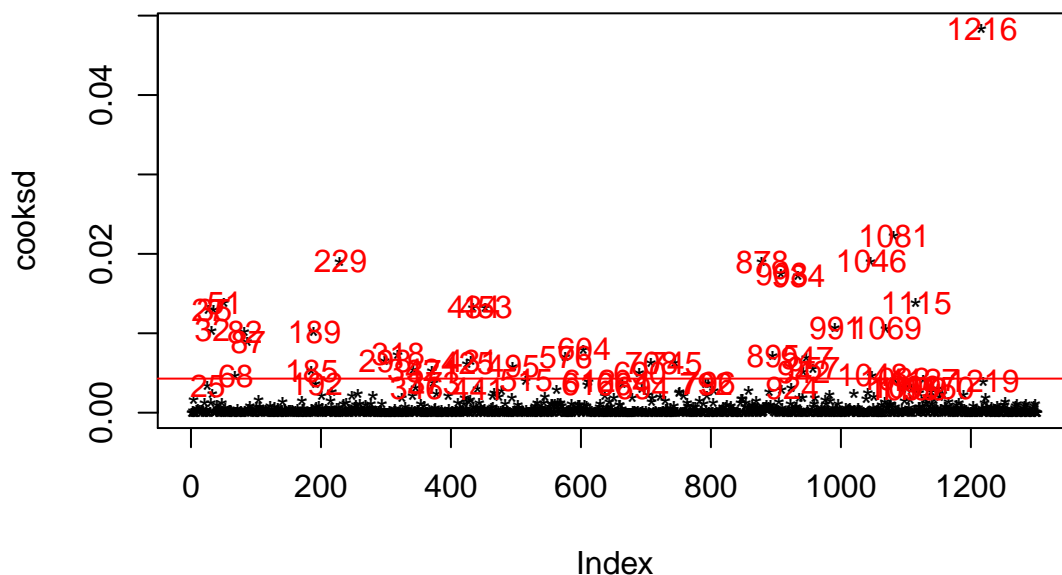
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
## 0.00000 0.00003 0.00013 0.00073 0.00053 0.04835      91
```

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



```
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 labels
```

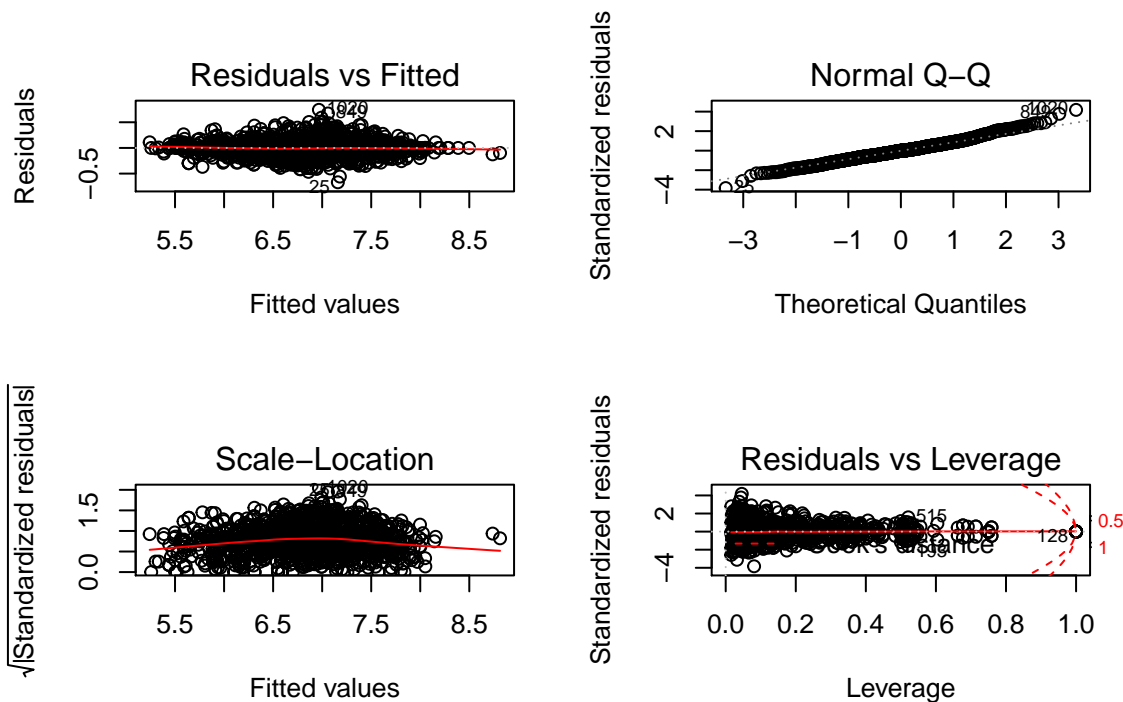
Influential Obs by Cooks distance



```

#extract influential obs
influential <- as.numeric(names(cooksdata)[(cooksdata > cutoff)]) # influential row numbers
influ=data.frame(data[cooksdata > cutoff, ])
filtered_data <- data[ !(row.names(data) %in% c(influential)), ]
#Outlier rimossi
lm_full_t_no_OUTliers = lm(log(Price) ~ ., data = filtered_data)
par(mfrow=c(2,2))
plot(lm_full_t_no_OUTliers)

```



```

#summary(lm_full_t_no_OUTliers) #FIXME: too long to be printed, R^2=0.9727
ncvTest(lm_full_t_no_OUTliers)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.672606, Df = 1, p = 0.19591

null = lm(log(Price) ~ 1, data = filtered_data)
full = lm(log(Price) ~ ., data = filtered_data)
lm_fit = stepAIC(null, scope = list(upper = full), direction = "both", trace = FALSE)
drop1(lm_fit, test = 'F')

```

```

## Single term deletions
##
## Model:
## log(Price) ~ Cpu + Memory + OpSys + Gpu + TypeName + ScreenResolution +
##      Company + Ram + Inches
##


|        | Df | Sum of Sq | RSS    | AIC     | F value | Pr(>F)        |
|--------|----|-----------|--------|---------|---------|---------------|
| <none> |    |           | 32.192 | -4060.0 |         |               |
| Cpu    | 84 | 15.4886   | 47.680 | -3731.1 | 5.5731  | < 2.2e-16 *** |
| Memory | 33 | 9.9446    | 42.136 | -3785.4 | 9.1083  | < 2.2e-16 *** |
| OpSys  | 5  | 5.9088    | 38.101 | -3856.8 | 35.7188 | < 2.2e-16 *** |
| Gpu    | 86 | 10.7637   | 42.956 | -3867.1 | 3.7830  | < 2.2e-16 *** |


```



```

## TypeName          4      2.1581 34.350 -3985.9 16.3072 6.159e-13 ***
## ScreenResolution 29      4.7135 36.905 -3945.1  4.9126 1.419e-15 ***
## Company           14      3.1033 35.295 -3971.5  6.6999 3.640e-13 ***
## Ram               1      1.5490 33.741 -4002.5 46.8194 1.375e-11 ***
## Inches            1      1.0441 33.236 -4021.6 31.5567 2.528e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```