

# Analisi dataset mtcars

David Marabottini

2025-10-15

```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(data.table)
```

```
printn <- function(..., mode = "", sameLn=FALSE) {  
  endColumn = ifelse(sameLn == T, "", "\n")  
  if (knitr::is_latex_output() || mode == "") {  
    cat(..., endColumn)  
  } else {  
    cat(paste0("\033[", mode, "m"), ..., "\033[0m", endColumn, sep = "")  
  }  
}
```

```
data("mtcars")  
head(mtcars)
```

```
##           mpg cyl  disp  hp  drat    wt  qsec vs am gear carb  
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46 0  1   4    4  
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02 0  1   4    4  
## Datsun 710      22.8   4  108  93 3.85 2.320 18.61 1  1   4    1  
## Hornet 4 Drive  21.4   6  258 110 3.08 3.215 19.44 1  0   3    1  
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02 0  0   3    2  
## Valiant        18.1   6  225 105 2.76 3.460 20.22 1  0   3    1
```

```
str(mtcars)
```

```
## 'data.frame':   32 obs. of  11 variables:  
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...  
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...  
##  $ disp: num  160 160 108 258 360 ...  
##  $ hp : num  110 110 93 110 175 105 245 62 95 123 ...  
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...  
##  $ wt : num  2.62 2.88 2.32 3.21 3.44 ...  
##  $ qsec: num  16.5 17 18.6 19.4 17 ...  
##  $ vs : num  0 0 1 1 0 1 0 1 1 1 ...  
##  $ am : num  1 1 1 0 0 0 0 0 0 0 ...  
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...  
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

I valori sono tutti numerici

Dalla documentazione

```
help(mtcars)
```

```
## avvio in corso del server httpd per la guida ... fatto
```

A data frame with 32 observations on 11 (numeric) variables.

mpg	Miles/(US) gallon	<b>Variabile quantitativa continua</b> , consumo auto
cyl	Number of cylinders	<b>Variabile quantitativa discreta</b> , numero di cilindri
disp	Displacement (cu.in.)	<b>Variabile quantitativa continua</b> , Volume totale spazzato dal pistone
hp	Gross horsepower	<b>Variabile quantitativa discreta</b> , Cavalli
drat	Rear axle ratio	<b>Variabile quantitativa continua</b> , rapporto tra il numero di rotazioni compiute dall'albero di trasmissione e il numero di rotazioni compiute dall'asse della ruota (o semiasse) posteriore.
wt	Weight (1000 lbs)	<b>Variabile quantitativa continua</b> , peso
qsec	1/4 mile time	<b>Variabile quantitativa continua</b> , Migli per quarto di secondo
vs	Engine (0 = V-shaped, 1 = straight)	<b>Variabile Bernoulliana</b> , Tipo di motore
am	Transmission (0 = automatic, 1 = manual)	<b>Variabile Bernoulliana</b> , Tipo di cambio
gear	Number of forward gears	<b>Variabile quantitativa discreta</b> , nr marce
carb	Number of carburetors	<b>Variabile quantitativa discreta</b> , nr carburatori

L'unica variabile qualitativa presente è l'indice che indica marca e modello auto volendo si potrebbero creare variabili aggiuntive

```
which(is.na(mtcars))
```

```
## integer(0)
```

Non sono presenti variabili a null

```
to_factorize = c('cyl', 'hp', 'vs', 'am', 'gear', 'carb')
mtcars_factorized <- unserialize(serialize(mtcars, NULL))
```

```

for(i in to_factorize) {
  name = paste(i, '_factorized')
  printn(i, mode="1;36")
  mtcars_factorized[[name]] = factor(mtcars[[i]])
  printn(mtcars_factorized[[i]])
}

```

```

## cyl
## 6 6 4 6 8 6 8 4 4 6 6 8 8 8 8 8 4 4 4 4 8 8 8 8 4 4 4 8 6 8 4
## hp
## 110 110 93 110 175 105 245 62 95 123 123 180 180 180 205 215 230 66 52 65 97 150 150 245 175 66 91 1
## vs
## 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 1
## am
## 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 1 1 1 1
## gear
## 4 4 4 3 3 3 3 4 4 4 4 3 3 3 3 3 3 4 4 4 3 3 3 3 3 4 5 5 5 5 4
## carb
## 4 4 1 1 2 1 4 2 2 4 4 3 3 3 4 4 4 1 2 1 1 2 2 4 2 1 2 2 4 6 8 2

```

Calcolo media, mediana, deviazione standard e range per tutte le variabili di cui ha senso farlo

```

to_describe = c('mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'gear', 'carb')
for (i in to_describe){
  curcol = mtcars_factorized[[i]]
  printn(i, mode="1;36")
  summary = summary(curcol)
  for (name in names(summary)) {
    printn(name, ": ", mode="1", sameLn = T)
    printn(summary[[name]])
  }
  printn('Deviazione Standard', mode="1", sameLn = T)
  printn(sd(curcol))
  printn("Range:" , mode="1", sameLn = T)
  printn(diff(range(curcol)))
  printn('-----')
}

```

```

## mpg
## Min. : 10.4
## 1st Qu.: 15.425
## Median : 19.2
## Mean : 20.09062
## 3rd Qu.: 22.8
## Max. : 33.9
## Deviazione Standard 6.026948
## Range: 23.5
## -----
## cyl
## Min. : 4
## 1st Qu.: 4
## Median : 6

```

```

## Mean : 6.1875
## 3rd Qu. : 8
## Max. : 8
## Deviazione Standard 1.785922
## Range: 4
## -----
## disp
## Min. : 71.1
## 1st Qu. : 120.825
## Median : 196.3
## Mean : 230.7219
## 3rd Qu. : 326
## Max. : 472
## Deviazione Standard 123.9387
## Range: 400.9
## -----
## hp
## Min. : 52
## 1st Qu. : 96.5
## Median : 123
## Mean : 146.6875
## 3rd Qu. : 180
## Max. : 335
## Deviazione Standard 68.56287
## Range: 283
## -----
## drat
## Min. : 2.76
## 1st Qu. : 3.08
## Median : 3.695
## Mean : 3.596563
## 3rd Qu. : 3.92
## Max. : 4.93
## Deviazione Standard 0.5346787
## Range: 2.17
## -----
## wt
## Min. : 1.513
## 1st Qu. : 2.58125
## Median : 3.325
## Mean : 3.21725
## 3rd Qu. : 3.61
## Max. : 5.424
## Deviazione Standard 0.9784574
## Range: 3.911
## -----
## qsec
## Min. : 14.5
## 1st Qu. : 16.8925
## Median : 17.71
## Mean : 17.84875
## 3rd Qu. : 18.9
## Max. : 22.9
## Deviazione Standard 1.786943

```

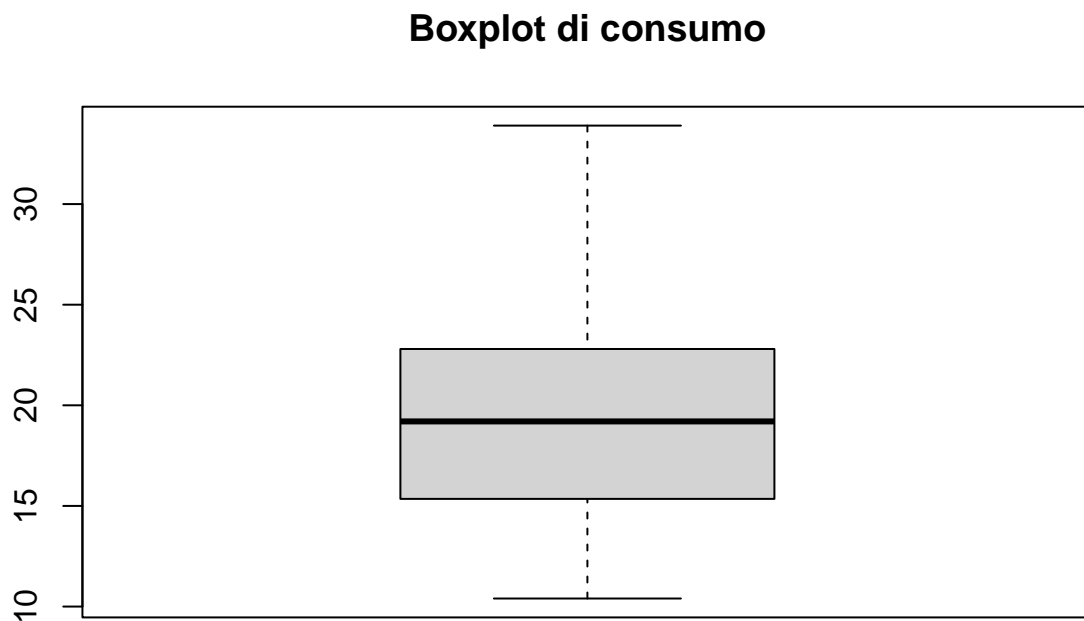
```
## Range: 8.4
## -----
## gear
## Min. : 3
## 1st Qu.: 3
## Median : 4
## Mean : 3.6875
## 3rd Qu.: 4
## Max. : 5
## Deviazione Standard 0.7378041
## Range: 2
## -----
## carb
## Min. : 1
## 1st Qu.: 2
## Median : 2
## Mean : 2.8125
## 3rd Qu.: 4
## Max. : 8
## Deviazione Standard 1.6152
## Range: 7
## -----
```

```
for (i in to_describe){
  printn(i, mode="1;36")
  outliers = boxplot(mtcars_factorized[[i]], plot=F)$out
  message = paste('sono presenti i seguenti outliers: ', outliers, sep = ' ')
  if(length(outliers) == 0) {
    printn('Non sono presnti outliers', mode="1")
  } else {
    printn("sono presenti i seguenti outlier: ", mode="1", sameLn = T)
    printn(paste(outliers, sep = ", "))
  }
  printn('-----')
}
```

```
## mpg
## Non sono presnti outliers
## -----
## cyl
## Non sono presnti outliers
## -----
## disp
## Non sono presnti outliers
## -----
## hp
## sono presenti i seguenti outlier: 335
## -----
## drat
## Non sono presnti outliers
## -----
## wt
## sono presenti i seguenti outlier: 5.424 5.345
## -----
```

```
## qsec
## sono presenti i seguenti outlier:  22.9
## -----
## gear
## Non sono presnti outliers
## -----
## carb
## sono presenti i seguenti outlier:  8
## -----
```

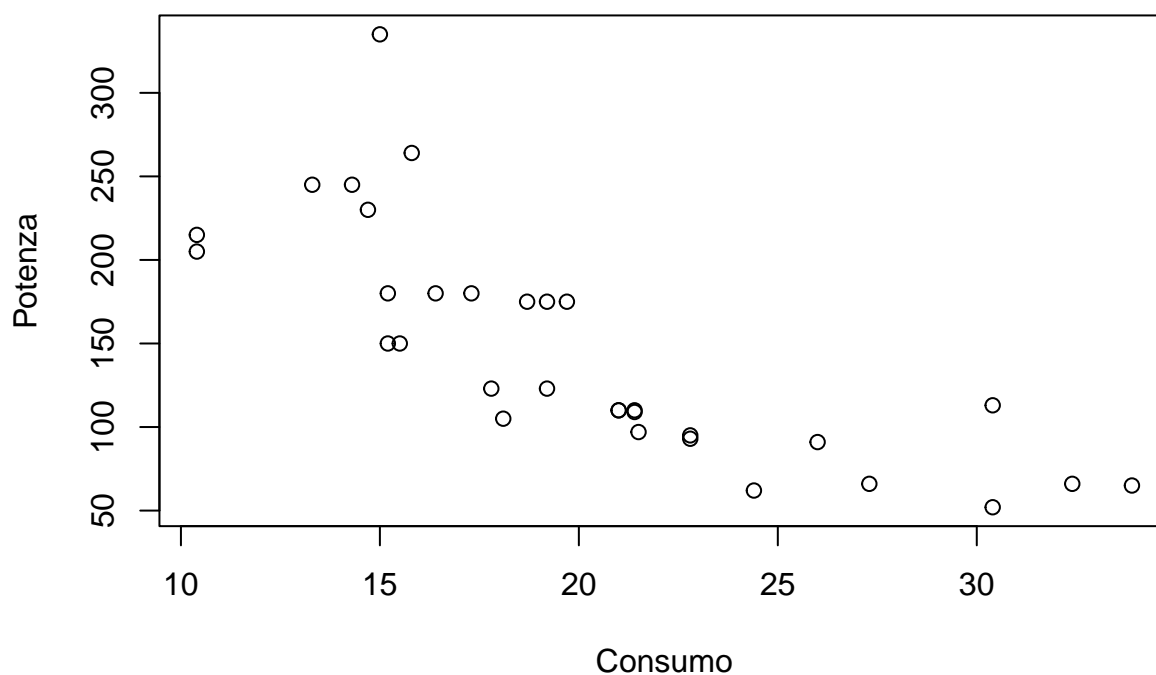
```
boxplot(mtcars$mpg, main="Boxplot di consumo")
```



Dal boxplot si evince che sebbene l'area tra il primo e il terzo percentile sia abbastanza equilibrata, e la mediana risulta ben posizionata in base a queste 2 aree, il grafico presenta una coda leggermente più lunga verso destra, anche se i valori si presentano ben posizionati dentro il range della normalità e non si segnalano outliers

```
plot(mtcars_factorized$mpg, mtcars_factorized$hp, main="consumo vs potenza", xlab="Consumo", ylab="Potenza")
```

## consumo vs potenza



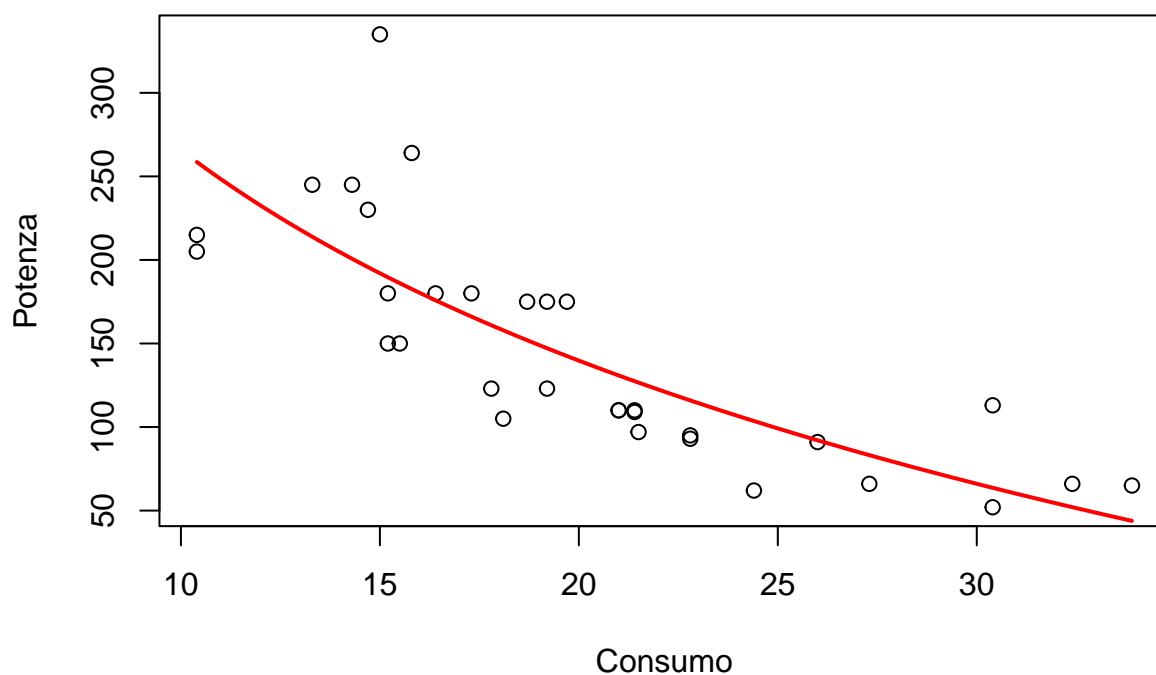
```
modello <- lm(hp ~ log(mpg), data = mtcars_factorized)
modello
```

```
##
## Call:
## lm(formula = hp ~ log(mpg), data = mtcars_factorized)
##
## Coefficients:
## (Intercept)      log(mpg)
##      684.3        -181.8
```

```
summary_model = summary(modello)
m = summary_model$coefficients[, 1][['log(mpg)']]
q = summary_model$coefficients[, 1][['(Intercept)']]
```

```
plot(mtcars_factorized$mpg, mtcars_factorized$hp, main="consumo vs potenza", xlab="Consumo", ylab="Potenza",
     curve(q + m * log(x),
           add = TRUE, col = "red", lwd = 2))
```

## consumo vs potenza

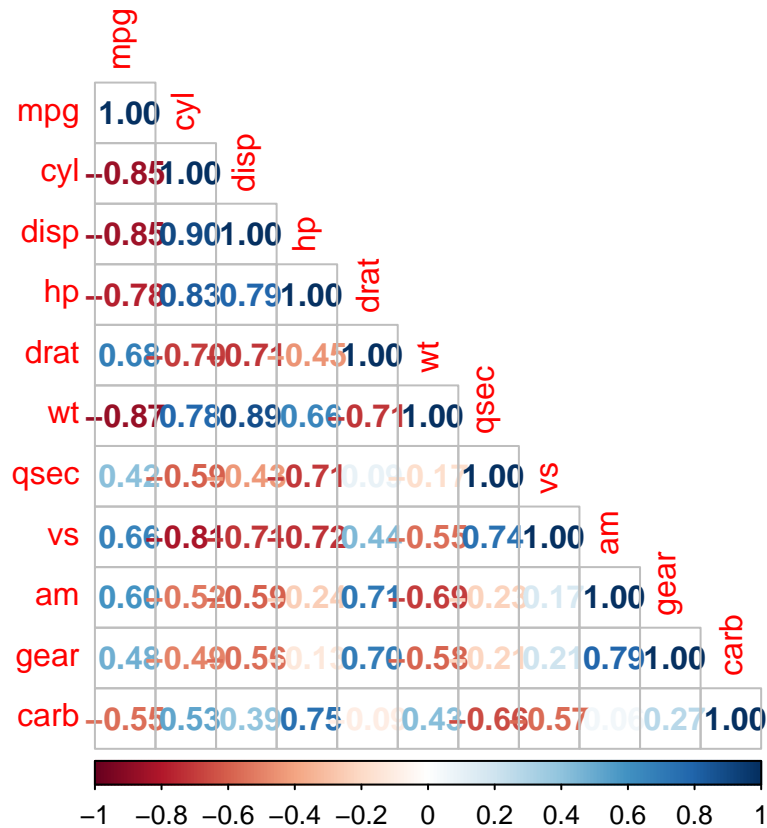


Dal grafico si vede che esiste approssimare l'esistenza di una relazione logaritmica negativa tra consumo e potenza del tipo

potenza è circa  $684 - 181 \cdot \log(\text{consumo})$  nelle macchine di questo grafico

```
cormap = cor(mtcars, use = "complete.obs")
corrplot(cormap,
  method = "number",
  type = "lower",
  number.cex = 1.0)
```





```
cormap[1:5, 'mpg']
```

```
##      mpg      cyl      disp      hp      drat
## 1.0000000 -0.8521620 -0.8475514 -0.7761684  0.6811719
```

le 5 colonne che hanno le relazioni lineari più forti con mpg sono:

cyl: relazione lineare inversa di 0.852, molto forte

disp: relazione lineare inversa di 0.847, molto forte

hp: relazione lineare inversa di 0.77, garantisce una buona correlazione

drat: relazione lineare diretta di 0.68, garantisce una discreta correlazione lineare

```
mod = lm(mpg ~ wt + hp + cyl, data = mtcars)
summary(mod)
```

```
##
## Call:
## lm(formula = mpg ~ wt + hp + cyl, data = mtcars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.9290 -1.5598 -0.5311  1.1850  5.8986
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 38.75179    1.78686  21.687 < 2e-16 ***
## wt          -3.16697    0.74058  -4.276 0.000199 ***
## hp          -0.01804    0.01188  -1.519 0.140015
## cyl         -0.94162    0.55092  -1.709 0.098480 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.512 on 28 degrees of freedom
## Multiple R-squared:  0.8431, Adjusted R-squared:  0.8263
## F-statistic: 50.17 on 3 and 28 DF,  p-value: 2.184e-11
```

Da questa analisi ho notato quanto segue:

- il consumo è pressappoco inversamente proporzionale a cilindrata, grandezza del pistone, potenza (con scala logaritmica) e direttamente influenzato da rear axle ratio (non so cosa sia)
- Ho notato un pattern interessante nella relazione tra consumo e potenza, come se il decrescere del risparmio energetico si smorzi a tendere
- Il dataset è molto piccolo (32 sole marche) ed è improbabile che questa quantità sia significativa nel mondo delle auto