Prova pratica del 6 giugno 2018

Corso di Data Mining
Laurea Magistrale in Informatica
Università degli Studi di Padova

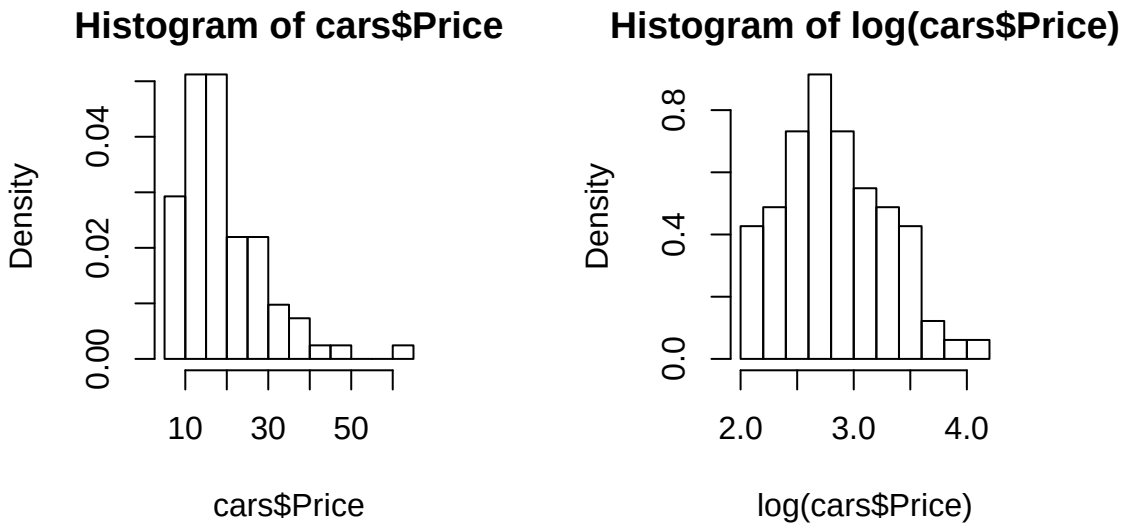
a.a. 2017/2018

Annamaria Guolo

1 Dataset `Cars93`

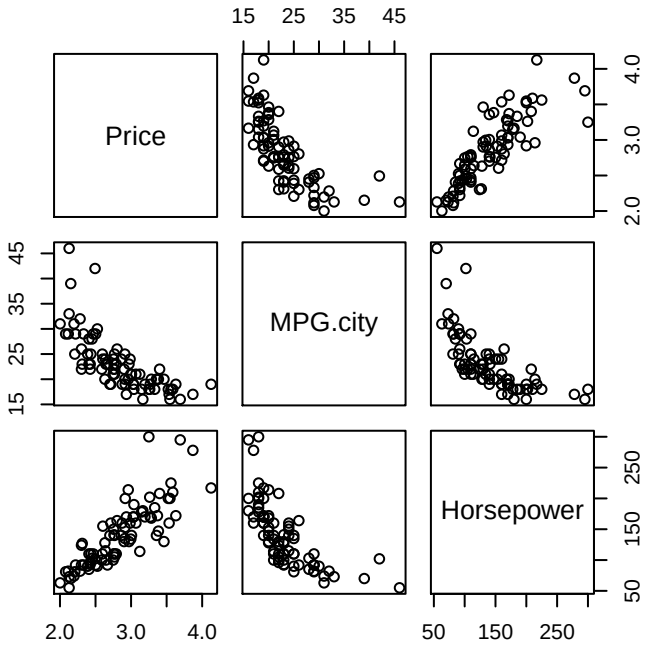
```
load('cars.RData')
dim(cars)

## [1] 82 22
```

```
par(mfrow=c(1,2))
hist(cars$Price, prob=TRUE)
hist(log(cars$Price), prob=TRUE)
```

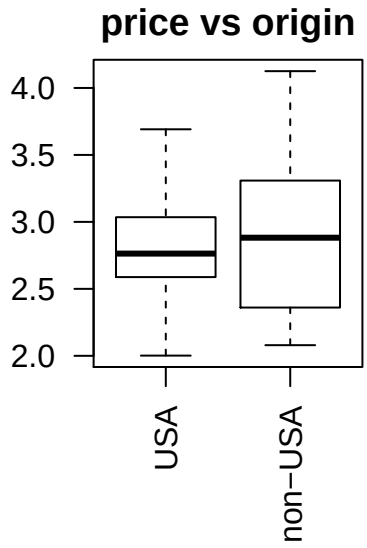


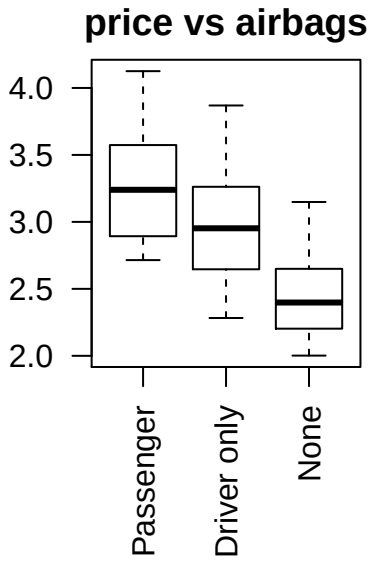
```r
cars$Price <- log(cars$Price)
dati <- cars[,c('Price','MPG.city','Horsepower','Origin', 'AirBags')]
dim(dati)

## [1] 82  5
```

```r
pairs(dati[,1:3])
```

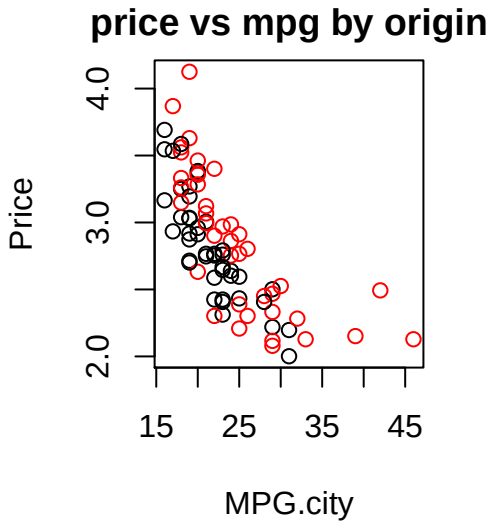


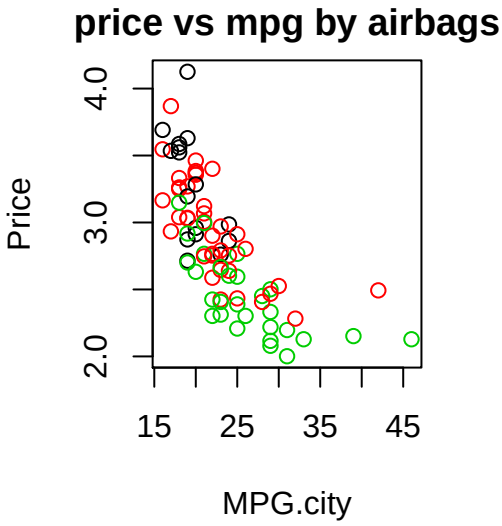

```r
par(mfrow=c(1,2))
boxplot(dati$Price~dati$Origin, main='price vs origin', las=2)
boxplot(dati$Price~dati$AirBags, main='price vs airbags', las=2)
```



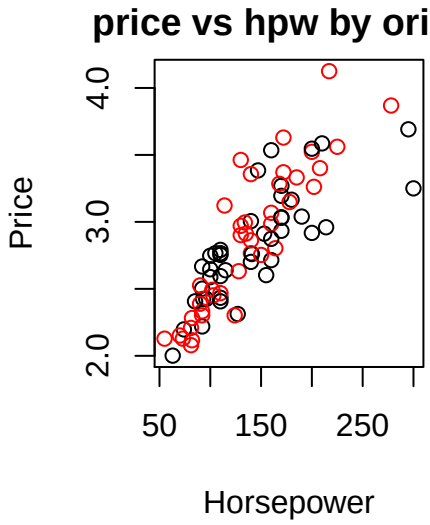


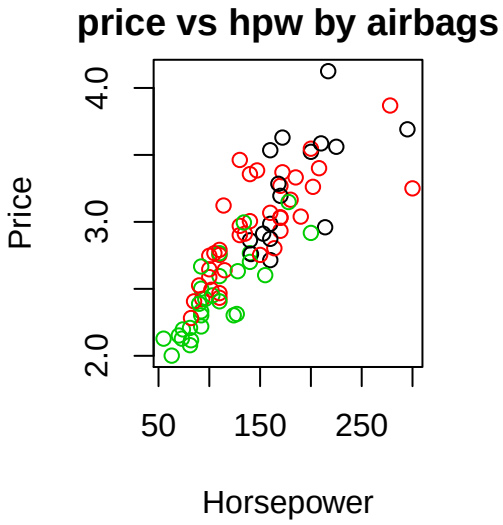

```
par(mfrow=c(1,2))
with(dati, plot(Price~MPG.city, col=Origin, main='price vs mpg by origin'))
with(dati, plot(Price~MPG.city, col=AirBags, main='price vs mpg by airbags'))
```






```
par(mfrow=c(1,2))
with(dati, plot(Price~Horsepower, col=Origin, main='price vs hpw by origin'))
with(dati, plot(Price~Horsepower, col=AirBags, main='price vs hpw by airbags'))
```






```
m <- lm(Price ~ MPG.city*Origin + MPG.city*AirBags +
        Horsepower*Origin + Horsepower*AirBags, data=dati)
summary(m)

##
## Call:
## lm(formula = Price ~ MPG.city * Origin + MPG.city * AirBags +
##     Horsepower * Origin + Horsepower * AirBags, data = dati)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39500 -0.08529 -0.01662  0.05968  0.45144
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.379775   0.787801   5.559  4.6e-07 ***
## MPG.city                        -0.089639   0.028814  -3.111   0.0027 **
## Originnon-USA                   -0.965181   0.493493  -1.956   0.0545 .
## AirBagsDriver only              -0.551433   0.834176  -0.661   0.5107
## AirBagsNone                     -1.733909   0.898427  -1.930   0.0577 .
## Horsepower                       0.002707   0.001625   1.666   0.1002
## MPG.city:Originnon-USA           0.024210   0.014798   1.636   0.1063
## MPG.city:AirBagsDriver only      0.035555   0.029661   1.199   0.2347
## MPG.city:AirBagsNone             0.062951   0.030362   2.073   0.0418 *
## Originnon-USA:Horsepower         0.004312   0.001350   3.195   0.0021 **
## AirBagsDriver only:Horsepower -0.001284   0.001826  -0.704   0.4840
## AirBagsNone:Horsepower           0.001252   0.002347   0.534   0.5953
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

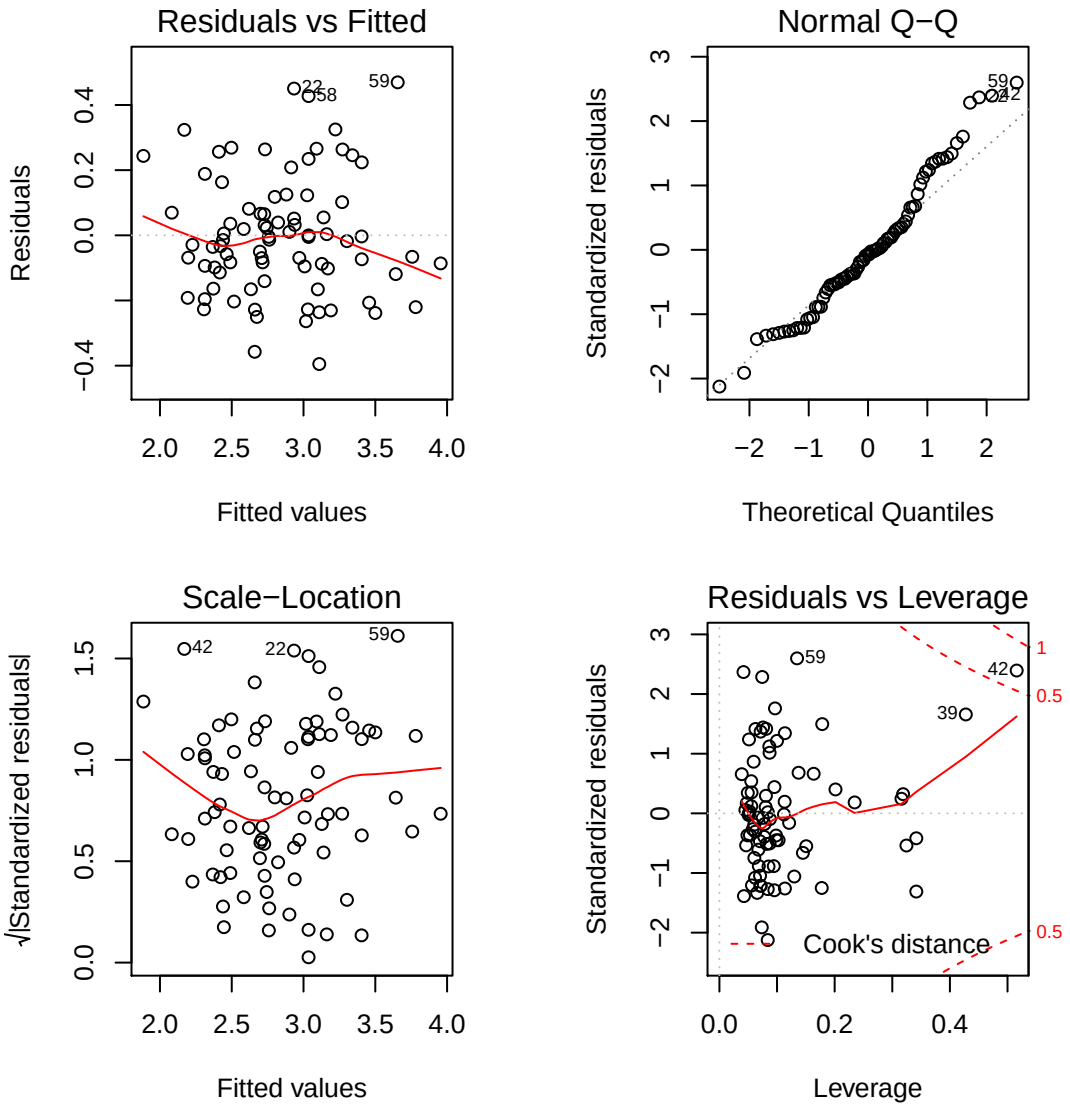
```
##
## Residual standard error: 0.1916 on 70 degrees of freedom
## Multiple R-squared:  0.8561,Adjusted R-squared:  0.8335
## F-statistic: 37.86 on 11 and 70 DF,  p-value: < 2.2e-16

m2 <- lm(Price ~ MPG.city*Origin + MPG.city*AirBags +
        Horsepower*Origin + Horsepower, data=dati)
summary(m2)

##
## Call:
## lm(formula = Price ~ MPG.city * Origin + MPG.city * AirBags +
##     Horsepower * Origin + Horsepower, data = dati)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41763 -0.10868 -0.01736  0.09490  0.47174
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   4.6369195  0.5789999   8.008 1.48e-11 ***
## MPG.city                     -0.0967070  0.0249110  -3.882 0.000227 ***
## Originnon-USA                -0.9630420  0.4778120  -2.016 0.047583 *
## AirBagsDriver only           -0.9820711  0.4588888  -2.140 0.035735 *
## AirBagsNone                  -1.5747972  0.4781940  -3.293 0.001537 **
## Horsepower                    0.0020803  0.0009024   2.305 0.024035 *
## MPG.city:Originnon-USA        0.0244596  0.0143250   1.707 0.092043 .
## MPG.city:AirBagsDriver only   0.0461953  0.0228896   2.018 0.047299 *
## MPG.city:AirBagsNone          0.0617675  0.0231669   2.666 0.009466 **
## Originnon-USA:Horsepower      0.0042282  0.0013200   3.203 0.002025 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1918 on 72 degrees of freedom
## Multiple R-squared:  0.8517,Adjusted R-squared:  0.8331
## F-statistic: 45.94 on 9 and 72 DF,  p-value: < 2.2e-16

anova(m2, m)

## Analysis of Variance Table
##
## Model 1: Price ~ MPG.city * Origin + MPG.city * AirBags + Horsepower *
##     Origin + Horsepower
## Model 2: Price ~ MPG.city * Origin + MPG.city * AirBags + Horsepower *
##     Origin + Horsepower * AirBags
```

```
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     72 2.6482
## 2     70 2.5692  2  0.079042 1.0768 0.3463
```

```
m3 <- lm(Price ~ MPG.city + MPG.city*AirBags +
        Horsepower*Origin + Horsepower, data=dati)
summary(m3)
```

```
##
## Call:
## lm(formula = Price ~ MPG.city + MPG.city * AirBags + Horsepower *
##     Origin + Horsepower, data = dati)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39499 -0.11102 -0.01388  0.09679  0.46957
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   4.1685575  0.5165479   8.070 1.04e-11 ***
## MPG.city                     -0.0809710  0.0234453  -3.454 0.000924 ***
## AirBagsDriver only           -1.0298488  0.4640052  -2.219 0.029561 *
## AirBagsNone                  -1.6581393  0.4818969  -3.441 0.000963 ***
## Horsepower                    0.0029973  0.0007346   4.080 0.000114 ***
## Originnon-USA                -0.1810797  0.1380565  -1.312 0.193754
## MPG.city:AirBagsDriver only   0.0486987  0.0231403   2.104 0.038778 *
## MPG.city:AirBagsNone          0.0646503  0.0234065   2.762 0.007262 **
## Horsepower:Originnon-USA      0.0025645  0.0009021   2.843 0.005794 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1943 on 73 degrees of freedom
## Multiple R-squared:  0.8457,Adjusted R-squared:  0.8288
## F-statistic:    50 on 8 and 73 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(m3)
```



Introduciamo qualche polinomio

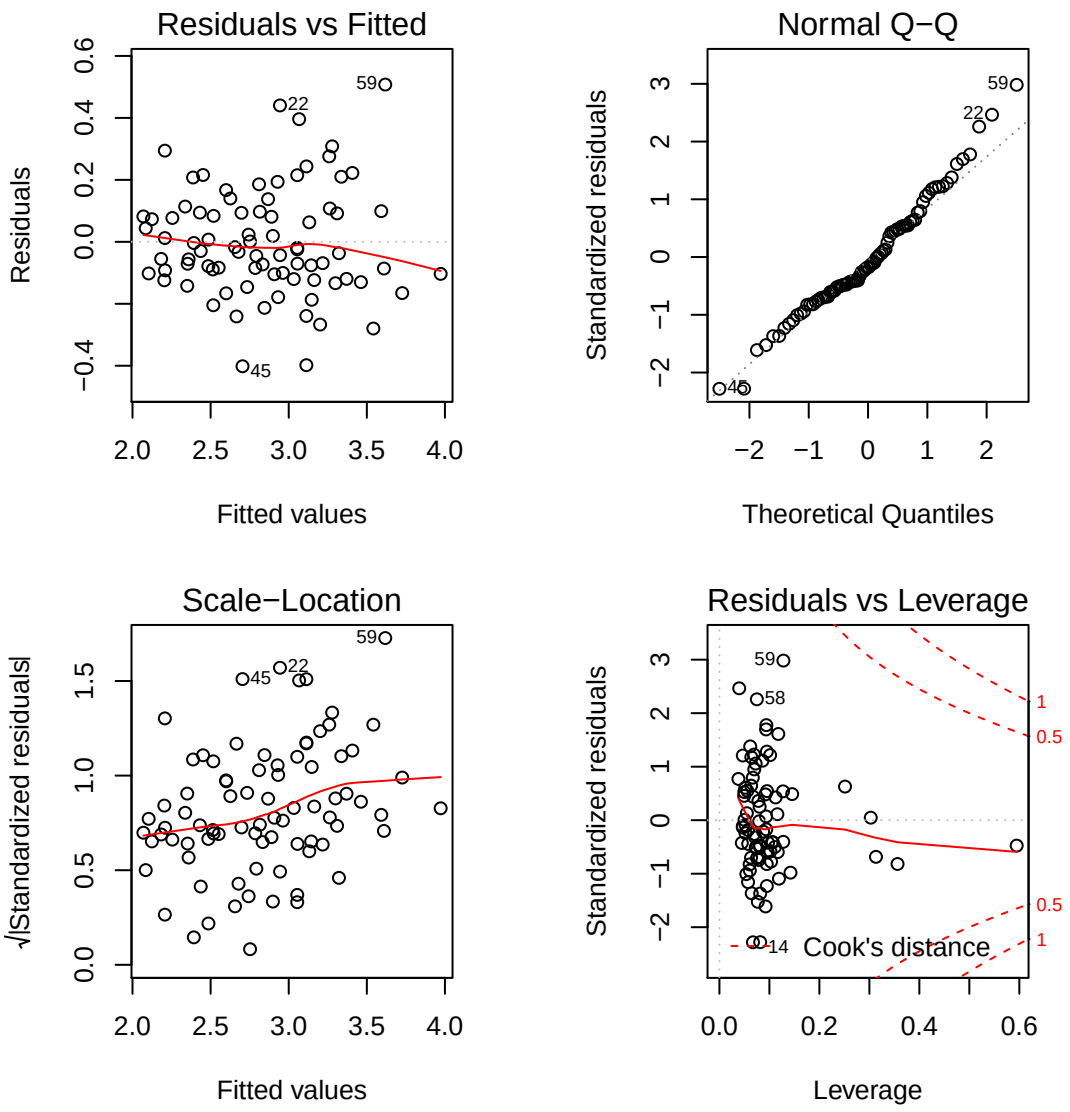
```
m4 <- lm(Price ~ MPG.city + I(MPG.city^2) + MPG.city*AirBags +
        Horsepower*Origin + Horsepower + I(Horsepower^2), data=dati)
summary(m4)

##
## Call:
## lm(formula = Price ~ MPG.city + I(MPG.city^2) + MPG.city * AirBags +
##     Horsepower * Origin + Horsepower + I(Horsepower^2), data = dati)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42761 -0.10789 -0.01798  0.07663  0.48467
##
```

```
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   5.025e+00  6.511e-01   7.718 5.57e-11 ***
## MPG.city                     -1.610e-01  3.276e-02  -4.915 5.51e-06 ***
## I(MPG.city^2)                 1.589e-03  5.122e-04   3.102  0.00275 **
## AirBagsDriver only           -8.655e-01  4.513e-01  -1.918  0.05915 .
## AirBagsNone                  -1.307e+00  5.007e-01  -2.610  0.01102 *
## Horsepower                    5.194e-03  2.925e-03   1.776  0.08002 .
## Originnon-USA                -2.147e-01  1.287e-01  -1.668  0.09963 .
## I(Horsepower^2)              -8.886e-06  7.359e-06  -1.208  0.23124
## MPG.city:AirBagsDriver only  4.083e-02  2.270e-02   1.799  0.07632 .
## MPG.city:AirBagsNone         5.027e-02  2.442e-02   2.058  0.04323 *
## Horsepower:Originnon-USA     2.802e-03  8.485e-04   3.303  0.00150 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1804 on 71 degrees of freedom
## Multiple R-squared:  0.8706,Adjusted R-squared:  0.8524
## F-statistic: 47.77 on 10 and 71 DF,  p-value: < 2.2e-16

m5 <- lm(Price ~ MPG.city + I(MPG.city^2) + MPG.city*AirBags +
        Horsepower*Origin + Horsepower, data=dati)
summary(m5)

##
## Call:
## lm(formula = Price ~ MPG.city + I(MPG.city^2) + MPG.city * AirBags +
##     Horsepower * Origin + Horsepower, data = dati)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41374 -0.11280 -0.01521  0.08978  0.48156
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   5.3610074  0.5904104   9.080 1.48e-13 ***
## MPG.city                     -0.1655608  0.0326519  -5.070 2.97e-06 ***
## I(MPG.city^2)                 0.0017380  0.0004988   3.485 0.000843 ***
## AirBagsDriver only           -0.7383496  0.4402112  -1.677 0.097828 .
## AirBagsNone                  -1.1120040  0.4754297  -2.339 0.022117 *
## Horsepower                    0.0017854  0.0007675   2.326 0.022830 *
## Originnon-USA                -0.2184269  0.1290364  -1.693 0.094825 .
## MPG.city:AirBagsDriver only  0.0336744  0.0219807   1.532 0.129905
## MPG.city:AirBagsNone         0.0398835  0.0229308   1.739 0.086256 .
## Horsepower:Originnon-USA     0.0029133  0.0008462   3.443 0.000963 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.181 on 72 degrees of freedom
## Multiple R-squared:  0.8679,Adjusted R-squared:  0.8514
## F-statistic: 52.58 on 9 and 72 DF,  p-value: < 2.2e-16

m6 <- lm(Price ~ MPG.city + I(MPG.city^2) + AirBags +
        Horsepower*Origin + Horsepower, data=dati)
summary(m6)

##
## Call:
## lm(formula = Price ~ MPG.city + I(MPG.city^2) + AirBags + Horsepower *
##     Origin + Horsepower, data = dati)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40160 -0.10436 -0.03112  0.09647  0.50771
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               4.8593183  0.4789902  10.145 1.18e-15 ***
## MPG.city                 -0.1463280  0.0285375  -5.128 2.27e-06 ***
## I(MPG.city^2)             0.0020305  0.0004625   4.390 3.70e-05 ***
## AirBagsDriver only       -0.0771377  0.0572891  -1.346  0.18226
## AirBagsNone              -0.2893483  0.0684304  -4.228 6.65e-05 ***
## Horsepower                0.0018794  0.0007705   2.439  0.01712 *
## Originnon-USA            -0.2093899  0.1298484  -1.613  0.11109
## Horsepower:Originnon-USA  0.0027986  0.0008497   3.294  0.00152 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1823 on 74 degrees of freedom
## Multiple R-squared:  0.8623,Adjusted R-squared:  0.8493
## F-statistic: 66.22 on 7 and 74 DF,  p-value: < 2.2e-16

anova(m6, m5)

## Analysis of Variance Table
##
## Model 1: Price ~ MPG.city + I(MPG.city^2) + AirBags + Horsepower * Origin +
##     Horsepower
## Model 2: Price ~ MPG.city + I(MPG.city^2) + MPG.city * AirBags + Horsepower *
##     Origin + Horsepower
```

```
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     74 2.4582
## 2     72 2.3578  2   0.10037 1.5325  0.223
```

```
par(mfrow=c(2,2))
plot(m6)
```



Vince modello m6.
Proviamo qualche splines

```
sp.mpg <- smooth.spline(x=dati$MPG.city, y=dati$Price, cv=TRUE)

## Warning in smooth.spline(x = dati$MPG.city, y = dati$Price, cv = TRUE): cross-validat
with non-unique 'x' values seems doubtful
```

```r
sp.mpg
```

```
## Call:
## smooth.spline(x = dati$MPG.city, y = dati$Price, cv = TRUE)
##
## Smoothing Parameter  spar= 0.6725017  lambda= 0.01109883 (14 iterations)
## Equivalent Degrees of Freedom (Df): 3.872711
## Penalized Criterion (RSS): 0.635975
## PRESS(l.o.o. CV): 0.0776602
```

```r
## df=4
sp.hp <- smooth.spline(x=dati$Horsepower, y=dati$Price, cv=TRUE)
```

```
## Warning in smooth.spline(x = dati$Horsepower, y = dati$Price, cv = TRUE): cross-valid
with non-unique 'x' values seems doubtful
```

```r
sp.hp
```

```
## Call:
## smooth.spline(x = dati$Horsepower, y = dati$Price, cv = TRUE)
##
## Smoothing Parameter  spar= 0.9666355  lambda= 0.01974355 (15 iterations)
## Equivalent Degrees of Freedom (Df): 3.623617
## Penalized Criterion (RSS): 3.064721
## PRESS(l.o.o. CV): 0.0636254
```

```r
## df=4
library(gam)
m.gam <- gam(Price ~ s(MPG.city,4) + AirBags + s(Horsepower,4) * Origin, data=dati)
summary(m.gam)
```

```
##
## Call: gam(formula = Price ~ s(MPG.city, 4) + AirBags + s(Horsepower,
##     4) * Origin, data = dati)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41064 -0.09747 -0.02337  0.09748  0.48616
##
## (Dispersion Parameter for gaussian family taken to be 0.0342)
##
##     Null Deviance: 17.8554 on 81 degrees of freedom
## Residual Deviance: 2.3583 on 69.0002 degrees of freedom
## AIC: -30.2928
##
## Number of Local Scoring Iterations: 2
##
```

```
## Anova for Parametric Effects
##                          Df Sum Sq Mean Sq F value    Pr(>F)
## s(MPG.city, 4)            1 9.0655  9.0655 265.239 < 2.2e-16 ***
## AirBags                   2 1.7935  0.8967  26.237 3.355e-09 ***
## s(Horsepower, 4)          1 1.0816  1.0816  31.646 3.661e-07 ***
## Origin                    1 0.6390  0.6390  18.695 5.059e-05 ***
## s(Horsepower, 4):Origin   1 0.3513  0.3513  10.279   0.00204 **
## Residuals                69 2.3583  0.0342
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                          Npar Df Npar F     Pr(F)
## (Intercept)
## s(MPG.city, 4)                 3 7.1717 0.0002924 ***
## AirBags
## s(Horsepower, 4)               3 0.5969 0.6191572
## Origin
## s(Horsepower, 4):Origin
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## possiamo mettere horsepower come lineare
m.gam2 <- gam(Price ~ s(MPG.city,4) + AirBags + Horsepower * Origin, data=dati)
summary(m.gam2)
```

```
##
## Call: gam(formula = Price ~ s(MPG.city, 4) + AirBags + Horsepower *
##     Origin, data = dati)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40518 -0.10858 -0.03005  0.10143  0.49944
##
## (Dispersion Parameter for gaussian family taken to be 0.0336)
##
##     Null Deviance: 17.8554 on 81 degrees of freedom
## Residual Deviance: 2.4185 on 72.0001 degrees of freedom
## AIC: -34.2288
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## s(MPG.city, 4)     1 9.4374  9.4374 280.962 < 2.2e-16 ***
## AirBags            2 1.8205  0.9103  27.099 1.683e-09 ***
```

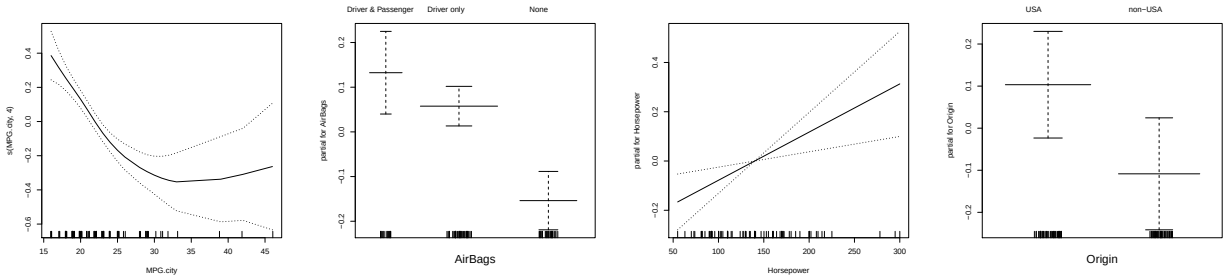
```
## Horsepower          1 1.0142  1.0142  30.193 5.602e-07 ***
## Origin              1 0.6788  0.6788  20.209 2.598e-05 ***
## Horsepower:Origin   1 0.3684  0.3684  10.969  0.001451 **
## Residuals          72 2.4185  0.0336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                    Npar Df Npar F     Pr(F)
## (Intercept)
## s(MPG.city, 4)           3  6.749 0.0004478 ***
## AirBags
## Horsepower
## Origin
## Horsepower:Origin
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m.gam, m.gam2)
```

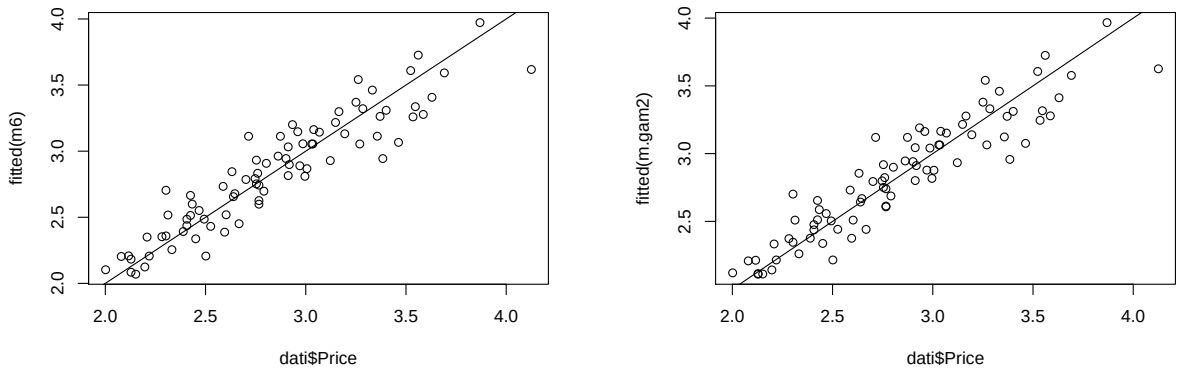
```
## Analysis of Deviance Table
##
## Model 1: Price ~ s(MPG.city, 4) + AirBags + s(Horsepower, 4) * Origin
## Model 2: Price ~ s(MPG.city, 4) + AirBags + Horsepower * Origin
##   Resid. Df Resid. Dev      Df  Deviance Pr(>Chi)
## 1        69     2.3583
## 2        72     2.4184 -2.9999 -0.060107    0.624
```

```
par(mfrow=c(1,4))
plot(m.gam2, se=TRUE)
```

```
## Warning in preplot.gam(x, terms = terms):  No terms saved for "a:b" style interaction
terms
```



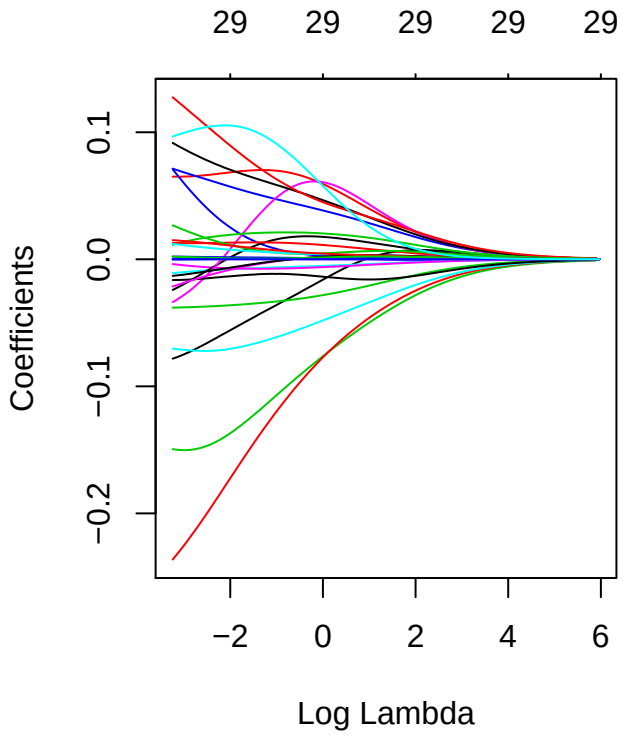
```r
par(mfrow=c(1,2))
plot(dati$Price, fitted(m6))
abline(0,1)
plot(dati$Price, fitted(m.gam2))
abline(0,1)
```



Regolarizzazione

```r
library(glmnet)
m.lm <- lm(Price ~ ., data=cars)
X <- model.matrix(m.lm)[,-1]
y <- cars$Price
m.ridge <- glmnet(x=X, y=y, alpha=0)
```

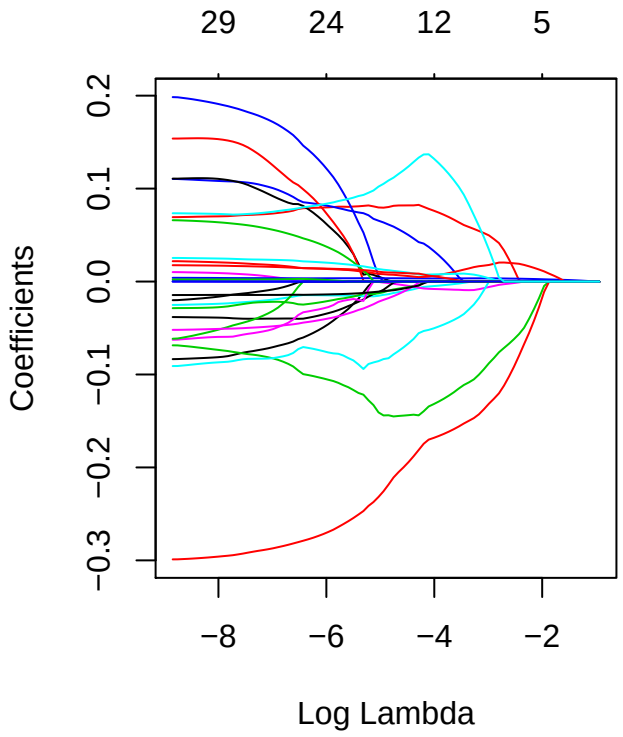
```r
plot(m.ridge, xvar='lambda')
```




```
set.seed(2906)
m.ridge.cv <- cv.glmnet(x=X, y=y, alpha=0)
m.ridge.min <- glmnet(x=X, y=y, alpha=0, lambda=m.ridge.cv$lambda.min)
cbind(coef(m.lm), coef(m.ridge.min))

## 30 x 2 sparse Matrix of class "dgCMatrix"
##                                            s0
## (Intercept)         3.196812e+00  1.905316e+00
## TypeLarge          -8.627820e-02 -6.228203e-02
## TypeMidsize         6.715068e-02  6.703580e-02
## TypeSmall          -6.171040e-02 -1.435832e-01
## TypeSporty          2.055918e-01  3.399529e-02
## MPG.city           -2.675308e-02 -8.318536e-03
## MPG.highway         1.121144e-02 -6.632907e-03
## AirBagsDriver only -3.653130e-02 -4.846891e-03
## AirBagsNone        -3.014483e-01 -1.894577e-01
## DriveTrainFront    -2.853966e-02 -3.699152e-02
## DriveTrainRear      1.138201e-01  6.047151e-02
## Cylinders4         -9.952024e-02 -7.171012e-02
## Cylinders5         -6.879863e-02 -5.098646e-03
## Cylinders6          1.062756e-01  7.485797e-02
## Cylinders8          1.494754e-01  9.876736e-02
## EngineSize         -7.478061e-02  1.779324e-02
## Horsepower          4.249307e-03  1.733677e-03
```

```
## RPM                 -2.262640e-05  6.715301e-05
## Rev.per.mile         5.184731e-05  3.279234e-05
## Man.trans.availYes -2.523624e-02 -1.436887e-02
## Fuel.tank.capacity  2.338568e-02  1.286404e-02
## Passengers           6.762413e-02  1.424003e-02
## Length               7.108963e-04  2.187435e-04
## Wheelbase            2.615676e-02  8.514346e-03
## Width               -5.269866e-02 -1.097788e-02
## Turn.circle         -1.458606e-02 -8.346716e-03
## Rear.seat.room       1.817791e-02  1.207728e-02
## Luggage.room         2.134666e-03  1.057164e-03
## Weight              -8.556264e-05  7.191159e-05
## Originnon-USA        7.456516e-02  1.051326e-01

m.lasso <- glmnet(x=X, y=y, alpha=1)
```

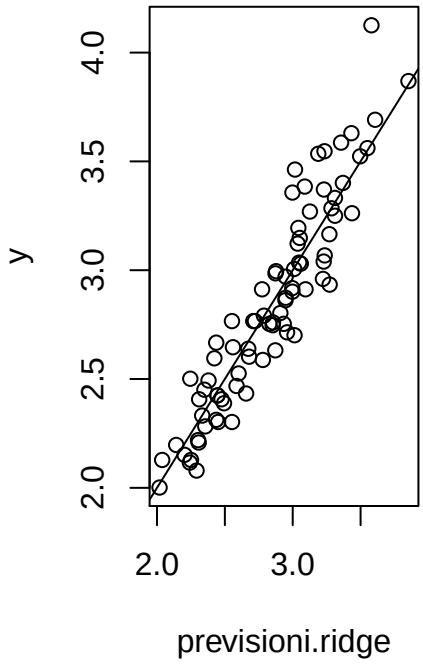
```
plot(m.lasso, xvar='lambda')
```

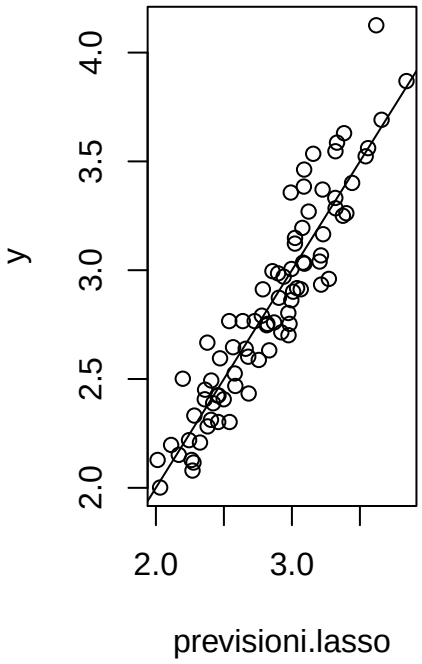



```
set.seed(2906)
m.lasso.cv <- cv.glmnet(x=X, y=y, alpha=1)
m.lasso.min <- glmnet(x=X, y=y, alpha=1, lambda=m.lasso.cv$lambda.min)
cbind(coef(m.lm), coef(m.ridge.min), coef(m.lasso.min))
```

```
## 30 x 3 sparse Matrix of class "dgCMatrix"
##                                              s0            s0
## (Intercept)          3.196812e+00  1.905316e+00  2.746836e+00
## TypeLarge           -8.627820e-02 -6.228203e-02  .
## TypeMidsize          6.715068e-02  6.703580e-02  8.086533e-02
## TypeSmall           -6.171040e-02 -1.435832e-01 -1.446790e-01
## TypeSporty           2.055918e-01  3.399529e-02  .
## MPG.city            -2.675308e-02 -8.318536e-03 -1.367076e-02
## MPG.highway          1.121144e-02 -6.632907e-03 -1.425334e-03
## AirBagsDriver only -3.653130e-02 -4.846891e-03 -5.067999e-03
## AirBagsNone         -3.014483e-01 -1.894577e-01 -2.188422e-01
## DriveTrainFront     -2.853966e-02 -3.699152e-02 -9.883541e-03
## DriveTrainRear       1.138201e-01  6.047151e-02  6.220368e-02
## Cylinders4          -9.952024e-02 -7.171012e-02 -8.088757e-02
## Cylinders5          -6.879863e-02 -5.098646e-03  .
## Cylinders6           1.062756e-01  7.485797e-02  1.748844e-03
## Cylinders8           1.494754e-01  9.876736e-02  .
## EngineSize          -7.478061e-02  1.779324e-02  .
## Horsepower           4.249307e-03  1.733677e-03  3.554784e-03
## RPM                 -2.262640e-05  6.715301e-05  .
## Rev.per.mile         5.184731e-05  3.279234e-05  8.081379e-06
## Man.trans.availYes -2.523624e-02 -1.436887e-02  .
## Fuel.tank.capacity  2.338568e-02  1.286404e-02  7.583332e-03
## Passengers           6.762413e-02  1.424003e-02  .
## Length               7.108963e-04  2.187435e-04  .
## Wheelbase            2.615676e-02  8.514346e-03  1.192336e-02
## Width               -5.269866e-02 -1.097788e-02 -1.778649e-02
## Turn.circle         -1.458606e-02 -8.346716e-03 -9.965819e-03
## Rear.seat.room       1.817791e-02  1.207728e-02  1.001628e-02
## Luggage.room         2.134666e-03  1.057164e-03  .
## Weight              -8.556264e-05  7.191159e-05  .
## Originnon-USA        7.456516e-02  1.051326e-01  1.080180e-01

previsioni.ridge <- predict(m.ridge.min, newx=X)
previsioni.lasso <- predict(m.lasso.min, newx=X)
```

```
par(mfrow=c(1,2))
plot(previsioni.ridge, y)
abline(0,1)
plot(previsioni.lasso, y)
abline(0,1)
```






```
min(m.ridge.cv$cvm)
```

```
## [1] 0.04177318
```

```
min(m.lasso.cv$cvm)
```

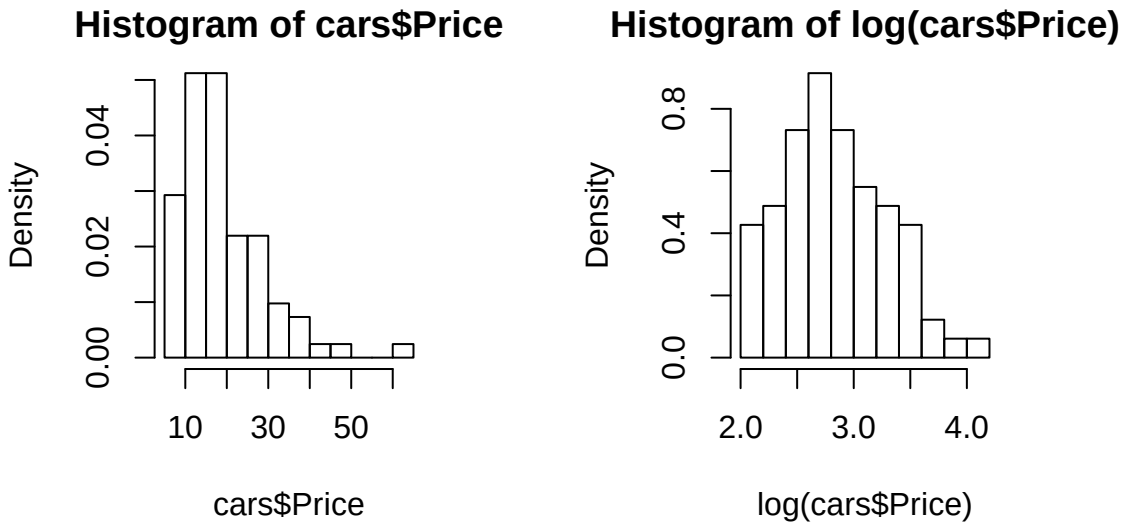
```
## [1] 0.04508531
```

```
## preferisco ridge
```

SECOND VERSION, REDUCED DATASET

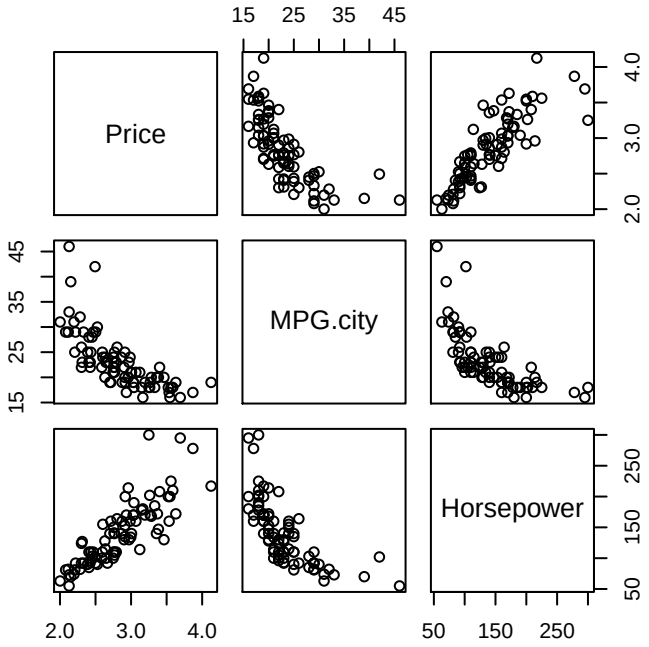
```
load('cars.RData')
dim(cars)

## [1] 82 22
```

```
par(mfrow=c(1,2))
hist(cars$Price, prob=TRUE)
hist(log(cars$Price), prob=TRUE)
```

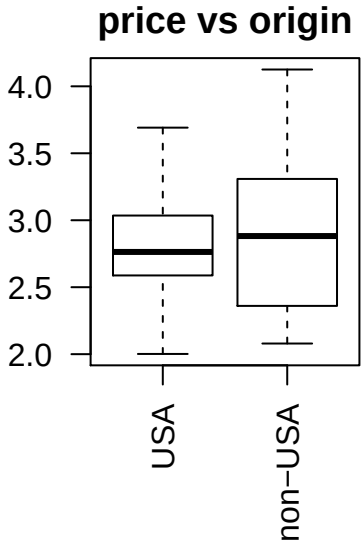


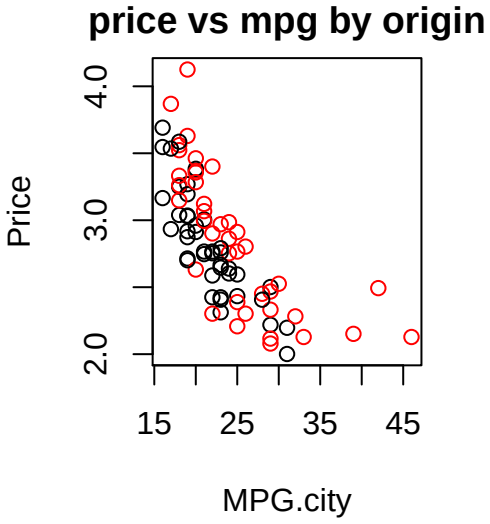

```
cars$Price <- log(cars$Price)
dati <- cars[,c('Price','MPG.city','Horsepower','Origin')]
dim(dati)

## [1] 82  4
```

```
pairs(dati[,1:3])
```




```
par(mfrow=c(1,2))
boxplot(dati$Price~dati$Origin, main='price vs origin', las=2)
with(dati, plot(Price~MPG.city, col=Origin, main='price vs mpg by origin'))
```






```
m <- lm(Price ~ MPG.city*Origin + I(MPG.city^2)*Origin + Horsepower*Origin, data=dati)
summary(m)

##
```

```
## Call:
## lm(formula = Price ~ MPG.city * Origin + I(MPG.city^2) * Origin +
##     Horsepower * Origin, data = dati)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52287 -0.14462 -0.00057  0.10125  0.53403
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  5.2759081  1.2252233   4.306 5.03e-05 ***
## MPG.city                    -0.1883572  0.0951647  -1.979   0.0515 .
## Originnon-USA               -1.1483180  1.4568209  -0.788   0.4331
## I(MPG.city^2)                0.0026976  0.0019447   1.387   0.1696
## Horsepower                   0.0021848  0.0010349   2.111   0.0381 *
## MPG.city:Originnon-USA       0.0518264  0.1049018   0.494   0.6227
## Originnon-USA:I(MPG.city^2) -0.0008052  0.0020584  -0.391   0.6968
## Originnon-USA:Horsepower     0.0041508  0.0015946   2.603   0.0112 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2079 on 74 degrees of freedom
## Multiple R-squared:  0.8209,Adjusted R-squared:  0.804
## F-statistic: 48.47 on 7 and 74 DF,  p-value: < 2.2e-16

m2 <- lm(Price ~ MPG.city*Origin  + Horsepower*Origin, data=dati)
summary(m2)

##
## Call:
## lm(formula = Price ~ MPG.city * Origin + Horsepower * Origin,
##     data = dati)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48898 -0.14568 -0.02352  0.11614  0.59247
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.6647245  0.4095182   8.949 1.67e-13 ***
## MPG.city                  -0.0575537  0.0134752  -4.271 5.57e-05 ***
## Originnon-USA             -1.5773687  0.5190098  -3.039 0.003252 **
## Horsepower                 0.0027760  0.0009903   2.803 0.006416 **
## MPG.city:Originnon-USA     0.0430341  0.0156118   2.757 0.007311 **
## Originnon-USA:Horsepower   0.0054792  0.0014446   3.793 0.000297 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2183 on 76 degrees of freedom
## Multiple R-squared:  0.7973,Adjusted R-squared:  0.7839
## F-statistic: 59.77 on 5 and 76 DF,  p-value: < 2.2e-16

anova(m, m2)

## Analysis of Variance Table
##
## Model 1: Price ~ MPG.city * Origin + I(MPG.city^2) * Origin + Horsepower *
##     Origin
## Model 2: Price ~ MPG.city * Origin + Horsepower * Origin
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     74 3.1970
## 2     76 3.6202 -2  -0.42312 4.8969 0.01006 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```