

Métodos Quantitativos

Prof. Dr. A. L. Korzenowski

Aula 03: Modelos Lineares de Regressão e Generalizados

Multiple (Linear) Regression

A regressão linear simples é utilizada para analisar relações entre variáveis contínuas. Para fazer a regressão no R a função é `lm()`, para linear models. A seguir temos um exemplo de aplicação das funções e discutiremos os conceitos em conjunto com os resultados.

```
# Multiple Linear Regression Example
require(car)
attach(mtcars)
fit <- lm(mpg ~ hp + wt + vs, data=mtcars)
summary(fit) # show results

##
## Call:
## lm(formula = mpg ~ hp + wt + vs, data = mtcars)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.4667 -1.4857 -0.4296  1.0341  5.7384
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  35.38267     2.42564   14.587 1.31e-14 ***
## hp           -0.02542     0.01100    -2.312  0.0284 *
## wt           -3.78003     0.63985   -5.908 2.35e-06 ***
## vs            1.36771     1.35296    1.011  0.3207
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.592 on 28 degrees of freedom
## Multiple R-squared:  0.8329, Adjusted R-squared:  0.815
## F-statistic: 46.52 on 3 and 28 DF,  p-value: 5.276e-11

# Other useful functions
coefficients(fit) # model coefficients

## (Intercept)          hp          wt          vs
## 35.38267257 -0.02542217 -3.78003115  1.36770864

confint(fit, level=0.95) # CIs for model parameters
```

```
##           2.5 %      97.5 %
## (Intercept) 30.41396676 40.351378382
## hp          -0.04794918 -0.002895161
## wt          -5.09070585 -2.469356445
## vs          -1.40370546  4.139122747
```

```
fitted(fit) # predicted values
```

```
##      Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
##      22.682552      21.718644      25.616447      21.801142
##  Hornet Sportabout      Valiant      Duster 360      Merc 240D
##      17.930485      21.002145      15.659529      23.115907
##      Merc 230      Merc 280      Merc 280C      Merc 450SE
##      22.428177      20.620147      20.620147      15.421955
##      Merc 450SL      Merc 450SLC  Cadillac Fleetwood  Lincoln Continental
##      16.707165      16.518164      10.325964      9.414017
##  Chrysler Imperial      Fiat 128      Honda Civic      Toyota Corolla
##      9.331306      26.756449      29.323678      28.161583
##      Toyota Corona      Dodge Challenger      AMC Javelin      Camaro Z28
##      24.966654      18.263637      18.584940      14.638921
##  Pontiac Firebird      Fiat X1-9      Porsche 914-2      Lotus Europa
##      16.399573      27.758158      24.979988      28.158489
##      Ford Pantera L      Ferrari Dino      Maserati Bora      Volvo 142E
##      16.688520      20.463106      13.371534      23.470878
```

```
residuals(fit) # residuals
```

```
##      Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
##      -1.68255202      -0.71864407      -2.81644693      -0.40114213
##  Hornet Sportabout      Valiant      Duster 360      Merc 240D
##      0.76951472      -2.90214536      -1.35952917      1.28409282
##      Merc 230      Merc 280      Merc 280C      Merc 450SE
##      0.37182326      -1.42014688      -2.82014688      0.97804521
##      Merc 450SL      Merc 450SLC  Cadillac Fleetwood  Lincoln Continental
##      0.59283462      -1.31816383      0.07403626      0.98598341
##  Chrysler Imperial      Fiat 128      Honda Civic      Toyota Corolla
##      5.36869353      5.64355068      1.07632205      5.73841714
##      Toyota Corona      Dodge Challenger      AMC Javelin      Camaro Z28
##      -3.46665373      -2.76363709      -3.38493974      -1.33892076
##  Pontiac Firebird      Fiat X1-9      Porsche 914-2      Lotus Europa
##      2.80042734      -0.45815757      1.02001176      2.24151138
##      Ford Pantera L      Ferrari Dino      Maserati Bora      Volvo 142E
##      -0.88852035      -0.76310615      1.62846634      -2.07087785
```

```
anova(fit) # anova table
```

```
## Analysis of Variance Table
##
## Response: mpg
##      Df Sum Sq Mean Sq  F value    Pr(>F)
## hp      1  678.37   678.37  100.9378 8.682e-11 ***
## wt      1  252.63   252.63   37.5893 1.287e-06 ***
```

```
## vs      1    6.87    6.87    1.0219    0.3207
## Residuals 28 188.18    6.72
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
vcov(fit) # covariance matrix for model parameters
```

```
##           (Intercept)           hp           wt           vs
## (Intercept)  5.88374720 -0.0113149889 -0.911920623 -2.468756850
## hp          -0.01131499  0.0001209413 -0.003153067  0.008499696
## wt          -0.91192062 -0.0031530667  0.409408892  0.130892197
## vs          -2.46875685  0.0084996961  0.130892197  1.830502316
```

```
influence(fit) # regression diagnostics
```

```
## $hat
##           Mazda RX4           Mazda RX4 Wag           Datsun 710           Hornet 4 Drive
##           0.15955082           0.14938930           0.07709486           0.08932495
##   Hornet Sportabout           Valiant           Duster 360           Merc 240D
##           0.05978585           0.10779325           0.11754812           0.12328101
##           Merc 230           Merc 280           Merc 280C           Merc 450SE
##           0.08750504           0.10667764           0.10667764           0.06959973
##           Merc 450SL           Merc 450SLC   Cadillac Fleetwood   Lincoln Continental
##           0.05773921           0.05860010           0.18589561           0.20936132
##   Chrysler Imperial           Fiat 128           Honda Civic           Toyota Corolla
##           0.18929309           0.08351815           0.12297672           0.10144147
##           Toyota Corona   Dodge Challenger           AMC Javelin           Camaro Z28
##           0.07407973           0.07939791           0.07841548           0.10408473
##   Pontiac Firebird           Fiat X1-9           Porsche 914-2           Lotus Europa
##           0.06310802           0.09476984           0.23357466           0.17564226
##           Ford Pantera L           Ferrari Dino           Maserati Bora           Volvo 142E
##           0.20736109           0.09816578           0.45237562           0.07597098
##
```

```
## $coefficients
##           (Intercept)           hp           wt           vs
## Mazda RX4          -0.6701915630  1.868451e-03  0.055437565  0.3547319314
## Mazda RX4 Wag      -0.2535971772  8.895852e-04  0.010271398  0.1455056622
## Datsun 710          -0.1121754754 -5.072644e-04  0.056503096 -0.2070079342
## Hornet 4 Drive      0.0499091717 -2.301442e-05 -0.012347062 -0.0470284162
## Hornet Sportabout   0.0933585062 -1.213916e-04 -0.006739381 -0.0646700638
## Valiant             0.4493044105  4.966094e-04 -0.147353686 -0.3422315290
## Duster 360          0.0330057789 -1.618218e-03  0.051081150 -0.0185575260
## Merc 240D           -0.0426200523 -1.171432e-03  0.071807710  0.0667466568
## Merc 230            -0.0322847532 -7.627502e-05  0.012675916  0.0352582833
## Merc 280            0.2634522722 -2.871483e-04 -0.056654659 -0.2028303057
## Merc 280C           0.5231670850 -5.702230e-04 -0.112505589 -0.4027831641
## Merc 450SE          0.0211990784 -3.720355e-04  0.029221418 -0.0635164509
## Merc 450SL          0.0417136752 -1.223082e-04  0.004458206 -0.0421816015
## Merc 450SLC         -0.0833352502  3.050462e-04 -0.014186746  0.0925127514
## Cadillac Fleetwood -0.0165546949 -4.161816e-05  0.007998501 -0.0005295614
## Lincoln Continental -0.2774505382 -4.480908e-04  0.117049910  0.0127360728
## Chrysler Imperial  -1.5695775945 -3.464814e-04  0.543089256  0.1830699320
## Fiat 128            0.6065349111 -1.621683e-03 -0.081025102  0.1930429916
```

```

## Honda Civic      0.2472235146 -2.955575e-04 -0.051822369 0.0027611128
## Toyota Corolla   0.9560708833 -7.031607e-04 -0.223031932 0.1467285022
## Toyota Corona    -0.0387464811 -6.371407e-04 0.043276080 -0.2834815844
## Dodge Challenger -0.4361978415 1.908471e-03 -0.025120559 0.3274423848
## AMC Javelin      -0.5760534740 2.188557e-03 -0.011716659 0.4067087081
## Camaro Z28       0.0867682198 -1.380430e-03 0.024970155 -0.0258602743
## Pontiac Firebird 0.1766960714 -1.011284e-03 0.049131940 -0.2126056035
## Fiat X1-9        -0.0680509094 7.036414e-05 0.014829997 -0.0132543419
## Porsche 914-2    0.5747826811 -1.397443e-03 -0.063905643 -0.2802403858
## Lotus Europa     0.3061340212 2.460076e-03 -0.209530554 0.2104860663
## Ford Pantera L   -0.0009676913 -1.771054e-03 0.074473639 -0.0317058306
## Ferrari Dino     -0.1734474130 -1.404777e-04 0.041503954 0.0779023705
## Maserati Bora    -0.5142933493 7.939587e-03 -0.224157581 0.3742948191
## Volvo 142E       0.1178751635 -5.341431e-04 -0.004483151 -0.2174519770
##
## $sigma
## Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
## 2.616269      2.635741      2.579009      2.638764
## Hornet Sportabout Valiant      Duster 360      Merc 240D
## 2.635582      2.572934      2.625270      2.626778
## Merc 230      Merc 280      Merc 280C      Merc 450SE
## 2.638941      2.624119      2.576796      2.632782
## Merc 450SL      Merc 450SLC Cadillac Fleetwood Lincoln Continental
## 2.637386      2.627025      2.639957      2.631365
## Chrysler Imperial Fiat 128      Honda Civic      Toyota Corolla
## 2.377572      2.383801      2.630722      2.369035
## Toyota Corona Dodge Challenger AMC Javelin      Camaro Z28
## 2.547334      2.581152      2.551303      2.625930
## Pontiac Firebird Fiat X1-9      Porsche 914-2      Lotus Europa
## 2.580619      2.638377      2.630464      2.596899
## Ford Pantera L Ferrari Dino      Maserati Bora      Volvo 142E
## 2.633008      2.635470      2.605814      2.607245
##
## $wt.res
## Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
## -1.68255202    -0.71864407    -2.81644693    -0.40114213
## Hornet Sportabout Valiant      Duster 360      Merc 240D
## 0.76951472     -2.90214536    -1.35952917     1.28409282
## Merc 230      Merc 280      Merc 280C      Merc 450SE
## 0.37182326     -1.42014688    -2.82014688     0.97804521
## Merc 450SL      Merc 450SLC Cadillac Fleetwood Lincoln Continental
## 0.59283462     -1.31816383    0.07403626      0.98598341
## Chrysler Imperial Fiat 128      Honda Civic      Toyota Corolla
## 5.36869353     5.64355068     1.07632205      5.73841714
## Toyota Corona Dodge Challenger AMC Javelin      Camaro Z28
## -3.46665373    -2.76363709    -3.38493974     -1.33892076
## Pontiac Firebird Fiat X1-9      Porsche 914-2      Lotus Europa
## 2.80042734     -0.45815757     1.02001176      2.24151138
## Ford Pantera L Ferrari Dino      Maserati Bora      Volvo 142E
## -0.88852035    -0.76310615     1.62846634     -2.07087785

```

Gráficos de Diagnóstico provém avaliação de heterocedasticidade, normalidade e observações influentes (potenciais outliers). Copie e cole o seguinte código para avaliar graficamente os pressupostos da Regressão Linear e testar a normalidade dos resíduos.

```
# diagnostic plots
layout(matrix(c(1,2,3,4),2,2)) # optional 4 graphs/page
plot(fit)

# executando o teste de normalidade
require(nortest)
lillie.test(residuals(fit))
```

Diagnósticos da Regressão

É possível que deseje-se maior rigos na avaliação dos pressupostos da regressão. Uma excelente revisão dos diagnósticos de regressão é fornecida por John Fox, apropriadamente chamado de Overview of Regression Diagnostics. O pacote **car** do Dr. Fox fornece utilitários avançados para modelagem de regressão.

Vamos assumir que estamos ajustando um modelo linear múltiplo nos dados da base **mtcars**.

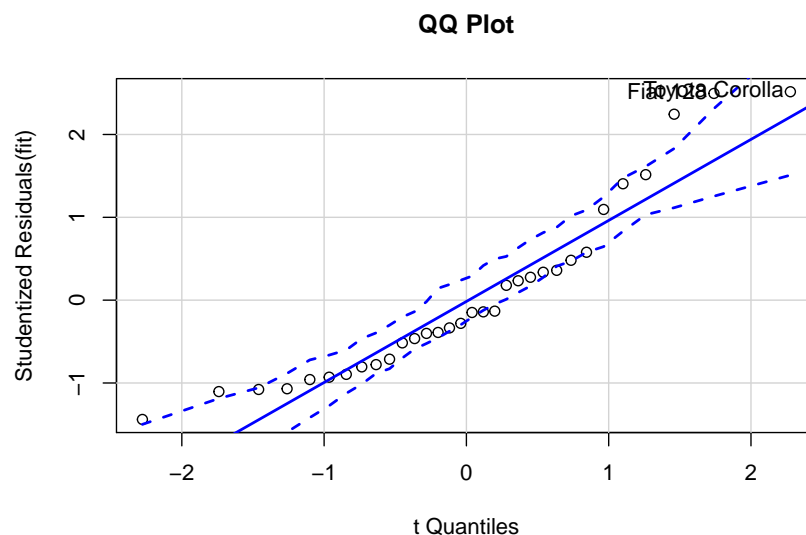
```
fit <- lm(mpg~disp+hp+wt+drat, data=mtcars)
```

Identificando os Outliers...

```
outlierTest(fit) # Bonferonni p-value for most extreme obs
```

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
##               rstudent unadjusted p-value Bonferroni p
## Toyota Corolla  2.51597          0.01838      0.58816
```

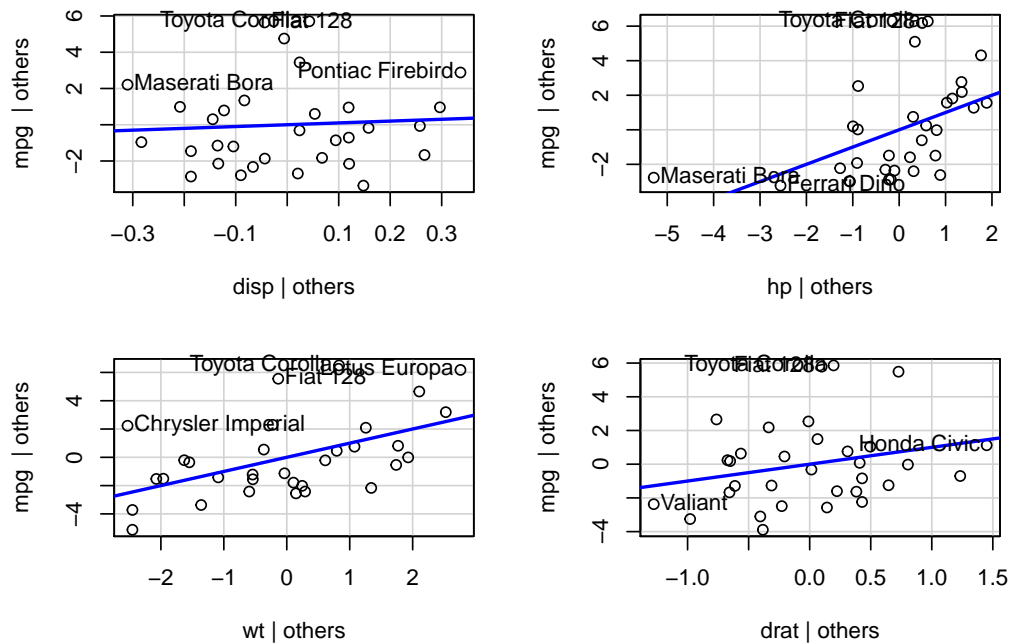
```
qqPlot(fit, main="QQ Plot") #qq plot for studentized resid
```



```
##      Fiat 128 Toyota Corolla
##      18          20
```

```
leveragePlots(fit) # leverage plots
```

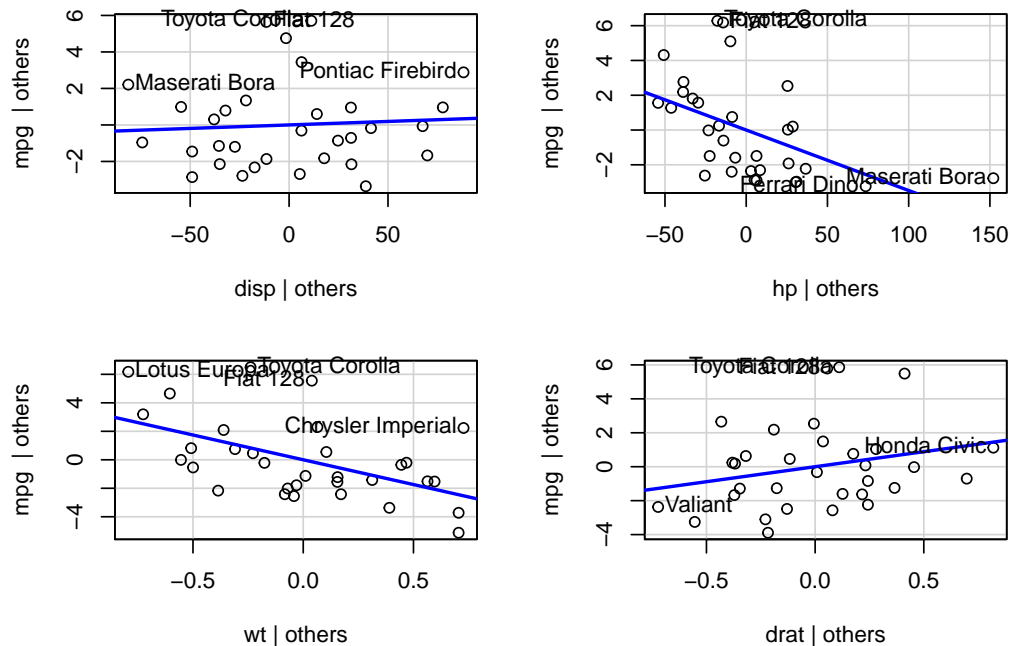
Leverage Plots



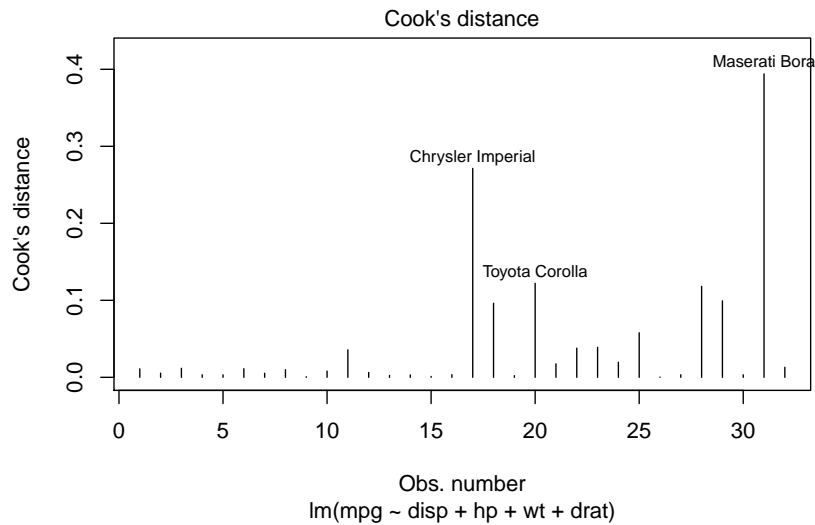
Avaliando a influência das variáveis...

```
# added variable plots
avPlots(fit)
```

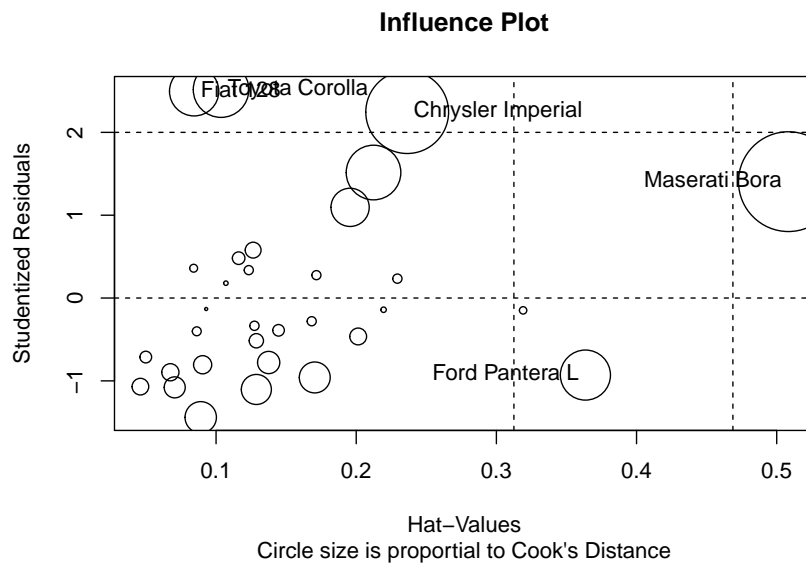
Added-Variable Plots



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



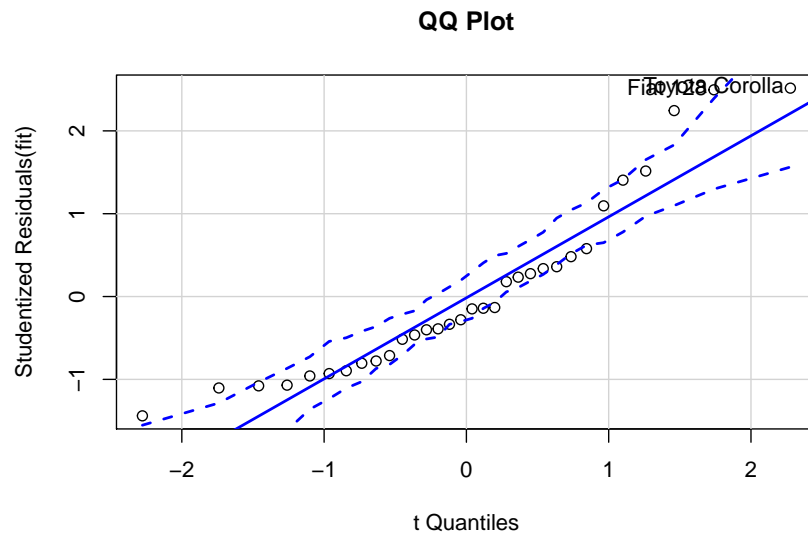
```
# Influence Plot
influencePlot(fit, id.method="identify", main="Influence Plot", sub="Circle size is proportional to Cook's Distance")
```



	StudRes	Hat	CookD
Chrysler Imperial	2.2446894	0.23636908	0.27133817
Fiat 128	2.4958292	0.08435766	0.09615527
Toyota Corolla	2.5159696	0.10355543	0.12213690
Ford Pantera L	-0.9300234	0.36351495	0.09929551
Maserati Bora	1.4051504	0.50833966	0.39406393

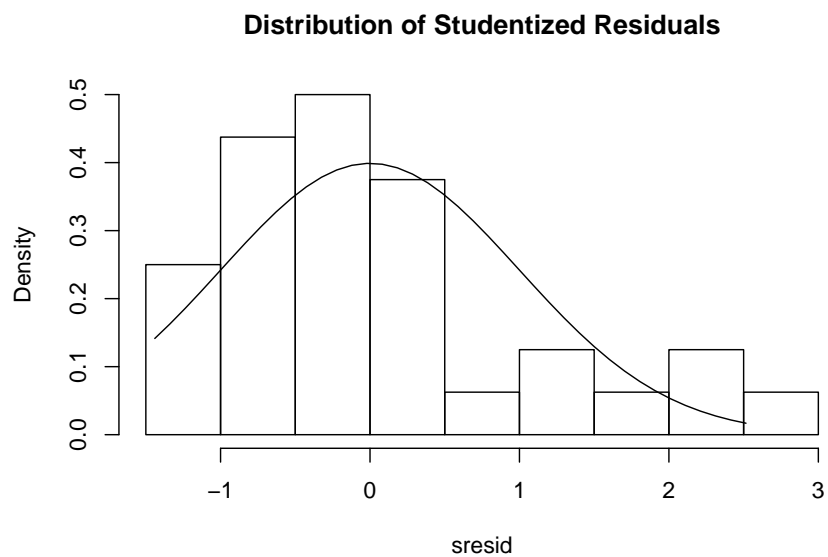
Avaliando a normalidade dos resíduos...

```
# qq plot for studentized resid
qqPlot(fit, main="QQ Plot")
```



```
##      Fiat 128 Toyota Corolla
##      18         20
```

```
# distribution of studentized residuals
require(MASS)
sresid <- studres(fit)
hist(sresid, freq=FALSE,
     main="Distribution of Studentized Residuals")
xfit<-seq(min(sresid),max(sresid),length=40)
yfit<-dnorm(xfit)
lines(xfit, yfit)
```

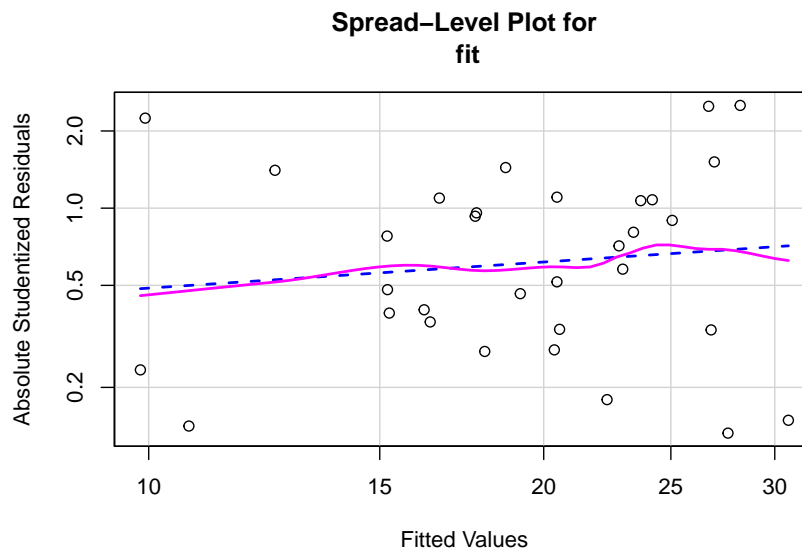


Avaliando a heterocedasticidade...


```
# non-constant error variance test
ncvTest(fit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.429672, Df = 1, p = 0.23182
```

```
# plot studentized residuals vs. fitted values
spreadLevelPlot(fit)
```



```
##
## Suggested power transformation: 0.6616338
```

Avaliando a presença de multicolinearidade...

```
vif(fit) # variance inflation factors
```

```
##      disp      hp      wt      drat
## 8.209402 2.894373 5.096601 2.279547
```

```
sqrt(vif(fit)) > 2 # problem?
```

```
## disp  hp  wt drat
## TRUE FALSE TRUE FALSE
```

Avaliando a independência dos resíduos - Erros não autocorrelacionados...

```
durbinWatsonTest(fit)
```

```
## lag Autocorrelation D-W Statistic p-value
## 1 0.100862 1.735915 0.256
## Alternative hypothesis: rho != 0
```

Comparando modelos

Você pode comparar modelos com a função `anova()`. O seguinte código verifica o impacto de `wt` e `vs` na predição de `mpg` em relação a um modelo que utiliza apenas `hp` como variável preditora.

```
# compare models
fit2 <- lm(mpg ~ hp, data=mtcars)
anova(fit2,fit)

## Analysis of Variance Table
##
## Model 1: mpg ~ hp
## Model 2: mpg ~ disp + hp + wt + drat
##   Res.Df    RSS Df Sum of Sq    F    Pr(>F)
## 1      30 447.67
## 2      27 182.84  3    264.84 13.036 1.885e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Seleção de Variáveis

A seleção de um subconjunto de variáveis preditoras de um conjunto maior (por exemplo, seleção por etapas) é um tópico controverso. Você pode executar a seleção passo a passo (avançar, retroceder, ambos) usando a função `stepAIC()` do pacote **MASS**. O `stepAIC()` realiza a seleção do modelo passo a passo pelo AIC exato.

```
# Stepwise Regression
require(MASS)
fit <- lm(mpg ~ disp + hp + drat + wt + qsec + vs, data=mtcars)
step <- stepAIC(fit, direction="both")

## Start:  AIC=67.42
## mpg ~ disp + hp + drat + wt + qsec + vs
##
##      Df Sum of Sq    RSS    AIC
## - vs   1      0.224 170.13 65.466
## - disp 1      4.178 174.08 66.201
## - qsec 1      5.760 175.66 66.491
## <none>          169.91 67.424
## - hp   1     12.051 181.96 67.617
## - drat 1     14.800 184.71 68.097
## - wt   1     75.423 245.33 77.180
##
## Step:  AIC=65.47
## mpg ~ disp + hp + drat + wt + qsec
##
##      Df Sum of Sq    RSS    AIC
## - disp 1      3.974 174.10 64.205
## <none>          170.13 65.466
## - hp   1     11.886 182.01 65.627
## - qsec 1     12.708 182.84 65.772
## - drat 1     15.506 185.63 66.258
```

```
## + vs      1      0.224 169.91 67.424
## - wt      1     81.394 251.52 75.978
##
## Step:  AIC=64.21
## mpg ~ hp + drat + wt + qsec
##
##           Df Sum of Sq   RSS   AIC
## - hp      1      9.418 183.52 63.891
## - qsec    1      9.578 183.68 63.919
## <none>                174.10 64.205
## - drat    1     11.956 186.06 64.331
## + disp    1      3.974 170.13 65.466
## + vs      1      0.021 174.08 66.201
## - wt      1    113.882 287.99 78.310
##
## Step:  AIC=63.89
## mpg ~ drat + wt + qsec
##
##           Df Sum of Sq   RSS   AIC
## <none>                183.52 63.891
## - drat    1     11.942 195.46 63.908
## + hp      1      9.418 174.10 64.205
## + disp    1      1.506 182.02 65.627
## + vs      1      0.147 183.37 65.865
## - qsec    1     85.720 269.24 74.156
## - wt      1    275.686 459.21 91.241
```

```
step$anova # display results
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## mpg ~ disp + hp + drat + wt + qsec + vs
##
## Final Model:
## mpg ~ drat + wt + qsec
##
##
##      Step Df  Deviance Resid. Df Resid. Dev      AIC
## 1              25    169.9049 67.42409
## 2  - vs      1 0.2242432      26    170.1291 65.46630
## 3 - disp    1 3.9743508      27    174.1035 64.20525
## 4  - hp      1 9.4180913      28    183.5216 63.89108
```

Como alternativa, você pode executar a regressão de todos os subconjuntos usando a função **leaps()** do pacote **leaps**. No código a seguir, o **nbest** indica o número de subconjuntos de cada tamanho a serem relatados. Aqui, os dez melhores modelos serão relatados para cada tamanho de subconjunto (1 preditor, 2 preditores etc.). Copie e cole o código no Console do R para verificar os resultados.

```
# All Subsets Regression
require(leaps)
leaps<-regsubsets(mpg ~ disp + hp + drat + wt + qsec + vs, data=mtcars,nbest=10)
```

```
# view results
summary(leaps)
# plot a table of models showing variables in each model.
# models are ordered by the selection statistic.
plot(leaps, scale="r2")
# plot statistic by subset size
require(car)
subsets(leaps, statistic="rsq")
```

Indo além...

O pacote **relaimpo** fornece medidas de importância relativa para cada um dos preditores no modelo. Consulte **help(calc.relimp)** para obter detalhes sobre as quatro medidas de importância relativa fornecidas.

Modelos Lineares Generalizados

Modelos lineares generalizados são ajustados com a função **glm()**. A forma da função é **glm(formula, family=familytype(link=linkfunction), data=)**. De acordo com as características da variável de resposta, a função link é definida. A mais comum é quando a resposta é binária. Neste caso temos um modelo de regressão logística e o link será “**logit**”. Consulte a **help(glm)** para outras opções de modelagem. Consulte **help(family)** para outras funções de vínculo permitidas para cada família.

Regressão Logística

A regressão logística é útil quando você está prevendo um resultado binário de um conjunto de variáveis preditivas contínuas. É frequentemente preferido em relação à análise da função discriminante devido às suas suposições menos restritivas.

```
fitlog <- glm(am ~ mpg + hp + qsec, data=mtcars, family=binomial())
summary(fitlog) # display results
```

```
##
## Call:
## glm(formula = am ~ mpg + hp + qsec, family = binomial(), data = mtcars)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.25597  -0.04564  -0.00055   0.00211   1.84909
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  23.712985  44.069519   0.538   0.591
## mpg          2.296430   1.623232   1.415   0.157
## hp           0.007295   0.047199   0.155   0.877
## qsec        -4.040952   2.890460  -1.398   0.162
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 43.2297  on 31  degrees of freedom
## Residual deviance:  7.4802  on 28  degrees of freedom
```

```
## AIC: 15.48
##
## Number of Fisher Scoring iterations: 9
```

```
confint(fitlog) # 95% CI for the coefficients
```

```
##              2.5 %      97.5 %
## (Intercept) -56.2578666 147.4033040
## mpg          0.6039345   7.5517535
## hp           -0.0946645   0.1335095
## qsec         -14.7757379  -0.9522515
```

```
exp(coef(fitlog)) # exponentiated coefficients
```

```
## (Intercept)      mpg      hp      qsec
## 1.988010e+10 9.938640e+00 1.007322e+00 1.758072e-02
```

```
exp(confint(fitlog)) # 95% CI for exponentiated coefficients
```

```
##              2.5 %      97.5 %
## (Intercept) 3.694188e-25 1.038584e+64
## mpg          1.829302e+00 1.904079e+03
## hp           9.096781e-01 1.142832e+00
## qsec         3.828059e-07 3.858713e-01
```

```
predict(fitlog, type="response") # predicted values
```

## Mazda RX4	Mazda RX4 Wag	Datsun 710	Hornet 4 Drive
## 9.980273e-01	9.813569e-01	8.246110e-01	7.413055e-03
## Hornet Sportabout	Valiant	Duster 360	Merc 240D
## 3.006604e-01	1.575206e-07	3.437816e-03	3.495827e-01
## Merc 230	Merc 280	Merc 280C	Merc 450SE
## 1.411858e-07	5.232093e-03	1.869433e-05	4.878419e-04
## Merc 450SL	Merc 450SLC	Cadillac Fleetwood	Lincoln Continental
## 1.715327e-03	2.746120e-06	5.832503e-11	1.197652e-10
## Chrysler Imperial	Fiat 128	Honda Civic	Toyota Corolla
## 1.307190e-05	1.000000e+00	1.000000e+00	1.000000e+00
## Toyota Corona	Dodge Challenger	AMC Javelin	Camaro Z28
## 8.532481e-04	4.224726e-04	3.733650e-05	1.968927e-03
## Pontiac Firebird	Fiat X1-9	Porsche 914-2	Lotus Europa
## 5.455810e-01	9.999728e-01	9.999999e-01	1.000000e+00
## Ford Pantera L	Ferrari Dino	Maserati Bora	Volvo 142E
## 9.653987e-01	9.994971e-01	8.327656e-01	1.809437e-01

```
residuals(fitlog, type="deviance") # residuals
```

## Mazda RX4	Mazda RX4 Wag	Datsun 710	Hornet 4 Drive
## 6.284293e-02	1.940055e-01	6.210370e-01	-1.219891e-01
## Hornet Sportabout	Valiant	Duster 360	Merc 240D
## -8.457172e-01	-5.612854e-04	-8.299083e-02	-9.275140e-01

```
##          Merc 230          Merc 280          Merc 280C          Merc 450SE
##      -5.313864e-04      -1.024288e-01      -6.114655e-03      -3.123975e-02
##          Merc 450SL          Merc 450SLC  Cadillac Fleetwood  Lincoln Continental
##      -5.859693e-02      -2.343555e-03      -1.080047e-05      -1.547678e-05
##  Chrysler Imperial          Fiat 128          Honda Civic          Toyota Corolla
##      -5.113118e-03          6.677698e-05          1.024527e-04          2.854518e-05
##          Toyota Corona  Dodge Challenger          AMC Javelin          Camaro Z28
##      -4.131858e-02      -2.907101e-02      -8.641435e-03      -6.278324e-02
##  Pontiac Firebird          Fiat X1-9  Porsche 914-2          Lotus Europa
##      -1.255974e+00          7.372978e-03          3.514364e-04          3.107240e-06
##          Ford Pantera L          Ferrari Dino          Maserati Bora          Volvo 142E
##          2.653830e-01          3.171982e-02          6.049845e-01          1.849091e+00
```

```
#confusion matrix
table(mtcars$am, predict(fitlog, type="response") > 0.5)
```

```
##
##      FALSE TRUE
##      0      18      1
##      1       1     12
```

Atividade

1. A partir da base de dados **Seguro_Residencial.xlsx** utilizada na Aula 02, construa um modelo de regressão para prever se um determinado lançamento é fraudulento. Considere o código a seguir para separar a base de dados em uma base de treinamento e outra de teste. Apresente a evolução dos resultados até o seu melhor modelo e justifique por que este modelo foi o escolhido.

```
require(caret)
indice.treino <- createDataPartition(y=1:dim(mydata)[1], p=0.75, list=FALSE)
treino = mydata[indice.treino, ]
teste = mydata[-indice.treino, ]
```