Matéria de Análise de Dados – Ciências Biomédicas

Aula 11 – Regressão Múltipla

James Hunter

24 de março de 2017

Nesta aula, vamos considerar **regressão múltipla**, ou seja, a inclusão de duas ou mais variáveis independentes na analise. Esta é uma extensão direta do que aprendemos na aula sobre regressão simples e regressão polinomial.

Regressão Múltipla

Vamos usar o dataset "Prestige" que descreve quais ocupações (entre os 102 no dataset) em Canadá têm as melhores reputações entre Canadenses e porque. Este dataset se encontra no pacote `car`, ligado ao texto sobre regressão **An R Companion to Applied Regression** de John Fox e Sanford Weisberg.

As variáveis no dataset são os seguintes:

- `education` : The average number of years of education for occupational incumbents in the 1971 Census of Canada.
- `income` : The average income of occupational incumbents, in dollars, in the 1971 Census.
- `women` : The percentage of occupational incumbents in the 1971 Census who were women.
- `prestige` : The average prestige rating for the occupation obtained in a sample survey conducted in Canada in 1966.
- `census` : The code of the occupation in the standard 1971 Census occupational classification.
- `type` : Professional and managerial (prof), white collar (wc), blue collar (bc), or missing (NA).

Nós queremos usar esses dados para desenvolver um modelo que mostra como nós podemos predizer quais profissões têm o maior prestígio.

Carregar os Pacotes Necessários

```
suppressMessages(library(tidyverse))
suppressPackageStartupMessages(library(DescTools))
suppressPackageStartupMessages(library(knitr))
suppressPackageStartupMessages(library(car))
suppressPackageStartupMessages(library(psych))
suppressPackageStartupMessages(library(broom))
suppressPackageStartupMessages(library(nortest))
suppressMessages(library(mosaic))
options(scipen = 5)
```

Como sempre, começamos com uma revisão dos dados:

Revisão dos Dados

```
data("Prestige")
occ <- rownames(Prestige)
head(Prestige)
```

```
##                       education income women prestige census type
## gov.administrators        13.11  12351 11.16     68.8   1113 prof
## general.managers          12.26  25879  4.02     69.1   1130 prof
## accountants               12.77   9271 15.70     63.4   1171 prof
## purchasing.officers       11.42   8865  9.11     56.8   1175 prof
## chemists                  14.62   8403 11.68     73.5   2111 prof
## physicists                15.64  11030  5.13     77.6   2113 prof
```

```
summary(Prestige)
```

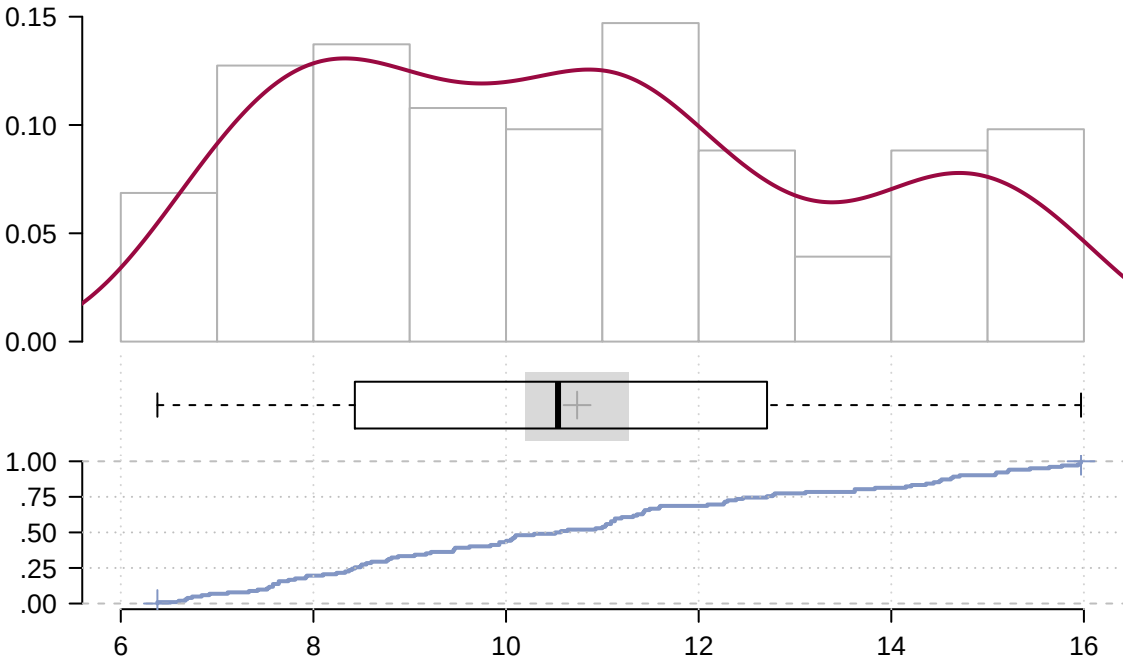
```
##    education          income          women           prestige
##  Min.   : 6.380   Min.   :  611   Min.   : 0.000   Min.   :14.80
##  1st Qu.: 8.445   1st Qu.: 4106   1st Qu.: 3.592   1st Qu.:35.23
##  Median :10.540   Median : 5930   Median :13.600   Median :43.60
##  Mean   :10.738   Mean   : 6798   Mean   :28.979   Mean   :46.83
##  3rd Qu.:12.648   3rd Qu.: 8187   3rd Qu.:52.203   3rd Qu.:59.27
##  Max.   :15.970   Max.   :25879   Max.   :97.510   Max.   :87.20
##      census       type
##  Min.   :1113   bc  :44
##  1st Qu.:3120   prof:31
##  Median :5135   wc  :23
##  Mean   :5402   NA's: 4
##  3rd Qu.:8312
##  Max.   :9517
```

```
Desc(Prestige[,c(1:4,6)], plotit = TRUE) # Não precisa census
```

```
## ------------------------------------------------------------------------
## Describe Prestige[, c(1:4, 6)] (data.frame):
##
## data.frame:  102 obs. of  5 variables
##
##   Nr  ColName    Class    NAs       Levels
##   1   education  numeric  .
##   2   income     integer  .
##   3   women      numeric  .
##   4   prestige   numeric  .
##   5   type       factor   4 (3.9%)  (3): 1-bc, 2-prof, 3-wc
##
##
## ------------------------------------------------------------------------
## 1 - education (numeric)
##
##   length       n     NAs   unique       0s     mean   meanCI
##      102     102       0       96        0  10.7380  10.2021
##           100.0%    0.0%             0.0%           11.2740
##
##      .05     .10     .25   median      .75      .90      .95
##   6.8440  7.5220  8.4450  10.5400  12.6475  14.7030  15.4290
```

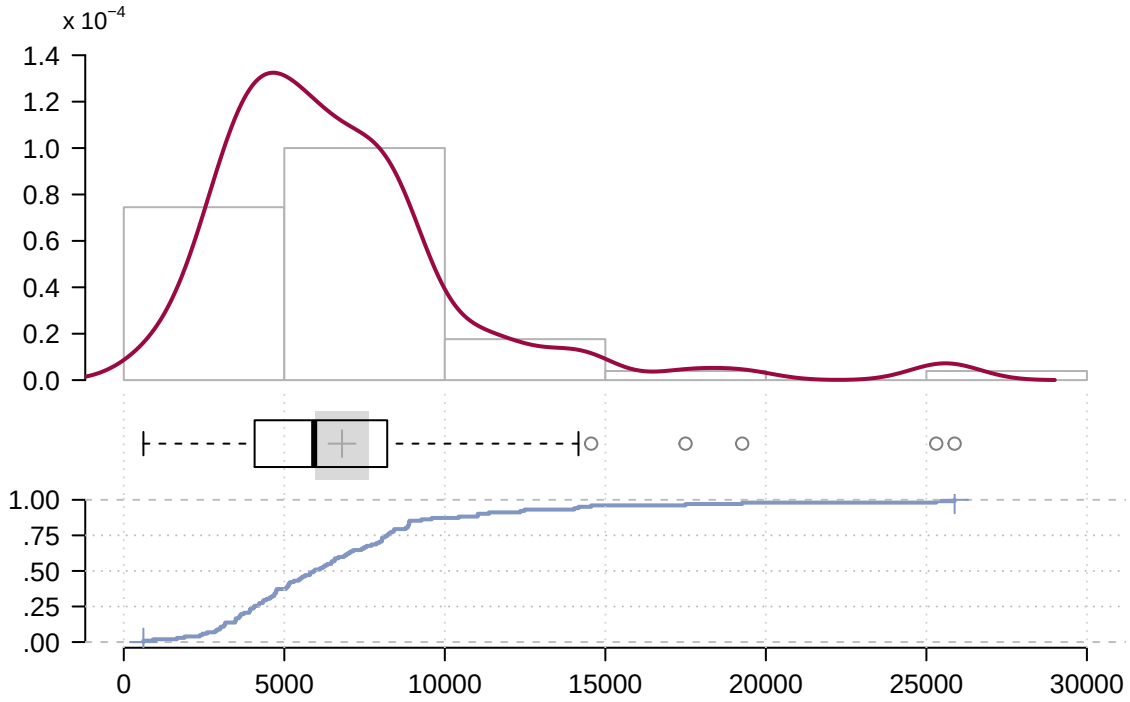
```
##
##    range      sd   vcoef      mad      IQR     skew     kurt
##   9.5900  2.7284  0.2541   3.1505   4.2025   0.3248  -1.0284
##
## lowest : 6.38, 6.6, 6.67, 6.69, 6.74
## highest: 15.64, 15.77, 15.94, 15.96, 15.97
```

1 − education (numeric)



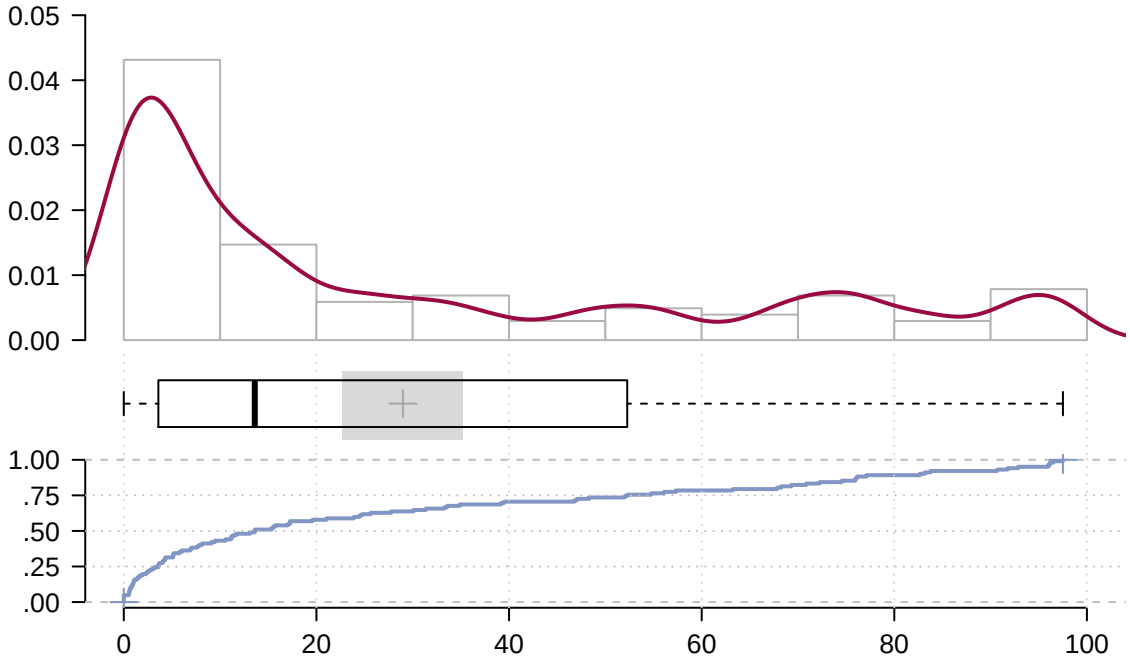

```
## ------------------------------------------------------------------------
## 2 - income (integer)
##
##      length         n       NAs    unique        0s       mean     meanCI
##         102       102         0       100         0   6'797.90   5'963.92
##                100.0%      0.0%               0.0%              7'631.88
##
##         .05       .10       .25    median       .75        .90        .95
##    2'455.30  3'026.00  4'106.00  5'930.50  8'187.25  11'029.30  14'156.45
##
##       range        sd     vcoef       mad       IQR       skew       kurt
##   25'268.00  4'245.92      0.62  3'060.83  4'081.25       2.13       6.29
##
## lowest : 611, 918, 1'656, 1'890, 2'370
## highest: 14'558, 17'498, 19'263, 25'308, 25'879
```

2 − income (integer)



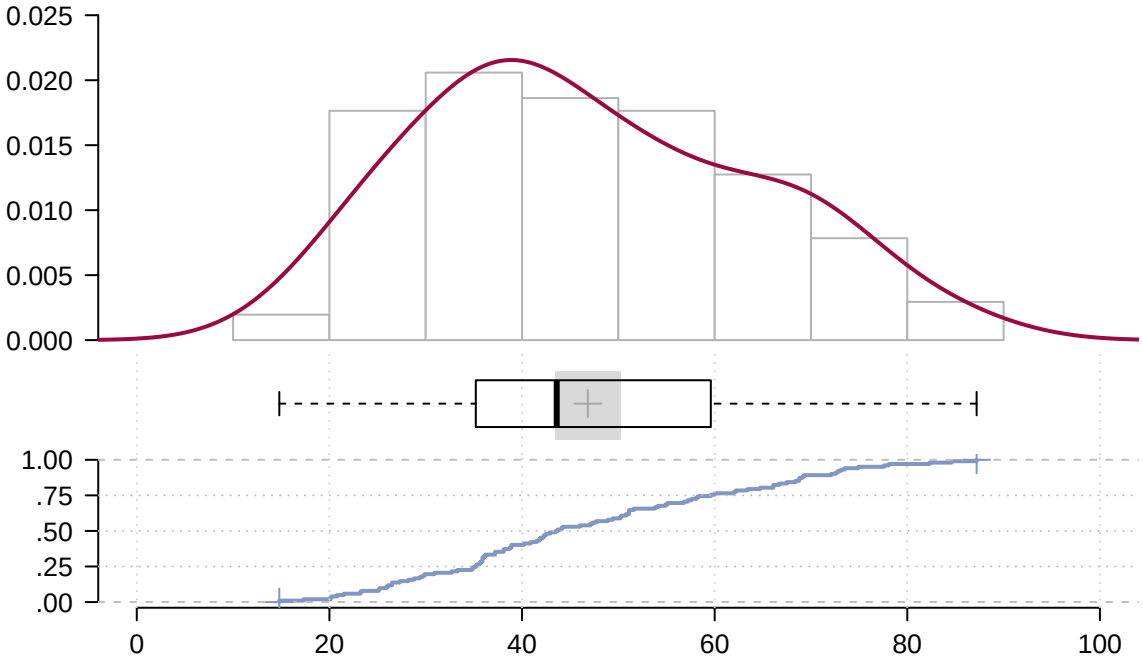

```
## ------------------------------------------------------------------------
## 3 - women (numeric)
##
##    length        n      NAs   unique       0s     mean   meanCI
##       102      102        0       96        5  28.9790  22.7477
##             100.0%     0.0%              4.9%           35.2104
##
##       .05      .10      .25   median      .75      .90      .95
##    0.5220   0.7830   3.5925  13.6000  52.2025  82.1040  92.8050
##
##     range       sd    vcoef      mad      IQR     skew     kurt
##   97.5100  31.7249   1.0948  18.7327  48.6100   0.8988  -0.6758
##
## lowest : 0.0 (5), 0.52, 0.56, 0.58, 0.61
## highest: 95.97, 96.12, 96.14, 96.53, 97.51
```

3 − women (numeric)

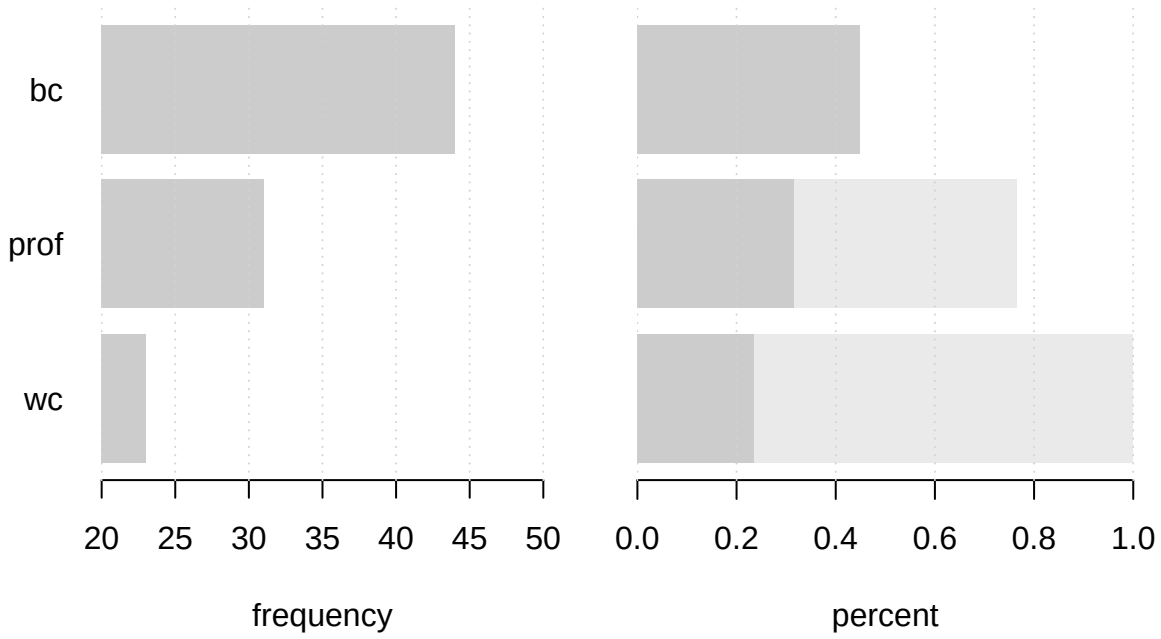



```
## ------------------------------------------------------------------------
## 4 - prestige (numeric)
##
##   length       n     NAs  unique      0s    mean  meanCI
##      102     102       0      92       0  46.833  43.454
##           100.0%    0.0%            0.0%          50.213
##
##      .05     .10     .25  median     .75     .90     .95
##   21.585  25.920  35.225  43.600  59.275  71.820  74.830
##
##    range      sd   vcoef     mad     IQR    skew    kurt
##   72.400  17.204   0.367  19.200  24.050   0.329  -0.793
##
## lowest : 14.8, 17.3, 20.1, 20.2, 20.8
## highest: 77.6, 78.1, 82.3, 84.6, 87.2
```

4 − prestige (numeric)




```
## ------------------------------------------------------------------------
## 5 - type (factor)
##
##   length      n    NAs unique levels  dupes
##      102     98      4      3      3      y
##           96.1%   3.9%
##
##    level  freq   perc  cumfreq  cumperc
## 1     bc    44  44.9%       44    44.9%
## 2   prof    31  31.6%       75    76.5%
## 3     wc    23  23.5%       98   100.0%
```




```
prestnorm <- unlist(ad.test(Prestige$prestige)[2])
educnorm <- unlist(ad.test(Prestige$education)[2])
incnorm <- unlist(ad.test(Prestige$income)[2])
mulhnorm <- unlist(ad.test(Prestige$women)[2])
paste("Normalidade de prestige per Anderson-Darling (valor-p):",
      round(prestnorm,3))
```

```
## [1] "Normalidade de prestige per Anderson-Darling (valor-p): 0.023"
```

```
paste("Normalidade de education per Anderson-Darling (valor-p):",
      round(educnorm,3))
```

```
## [1] "Normalidade de education per Anderson-Darling (valor-p): 0.002"
```

```
paste("Normalidade de income per Anderson-Darling (valor-p):",
      round(incnorm,3))
```

```
## [1] "Normalidade de income per Anderson-Darling (valor-p): 0"
```

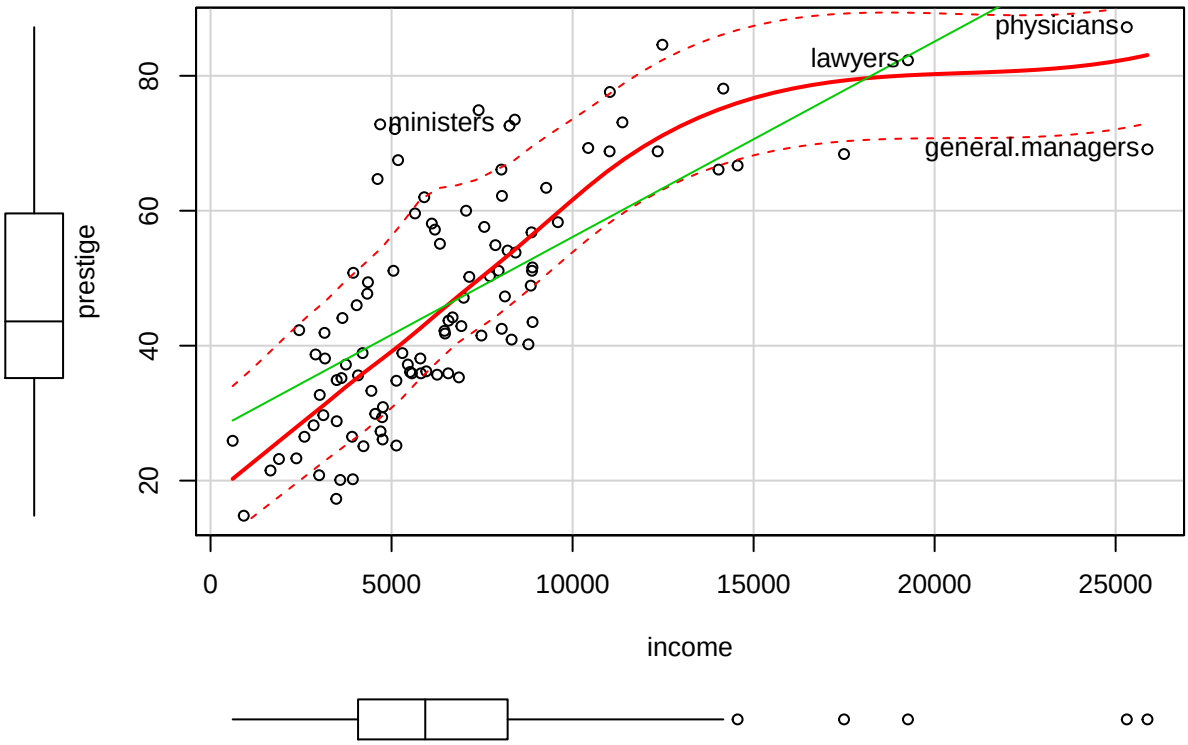
```
paste("Normalidade de women per Anderson-Darling (valor-p):",
      round(mulhnorm,3))
```

```
## [1] "Normalidade de women per Anderson-Darling (valor-p): 0"
```

```
cor(Prestige[,c(4,1:3)])
```

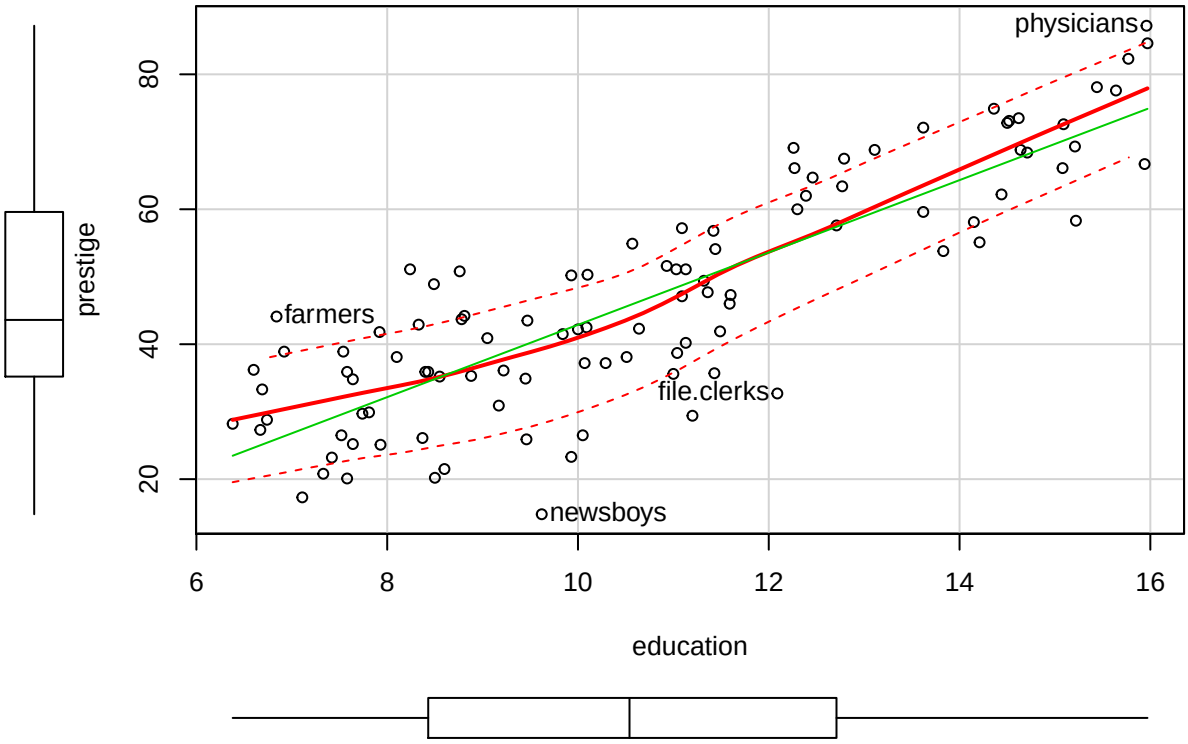
```
##             prestige  education     income       women
## prestige   1.0000000 0.85017689  0.7149057 -0.11833419
## education  0.8501769 1.00000000  0.5775802  0.06185286
## income     0.7149057 0.57758023  1.0000000 -0.44105927
## women     -0.1183342 0.06185286 -0.4410593  1.00000000
```

```
car::scatterplot(prestige ~ income, data = Prestige, id.n = 4, labels = occ)
```



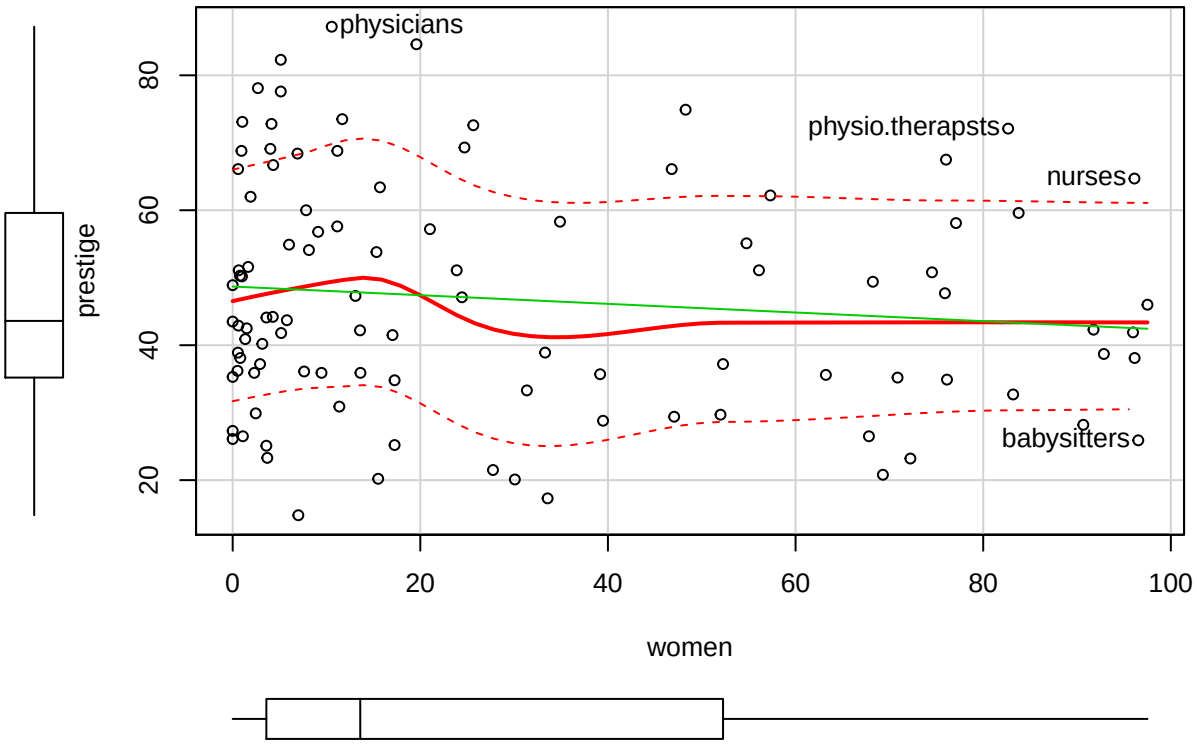

```
## general.managers          lawyers        ministers       physicians
##                2                17               20               24
```

```
scatterplot(prestige ~ education, data = Prestige, id.n = 4, labels = occ)
```



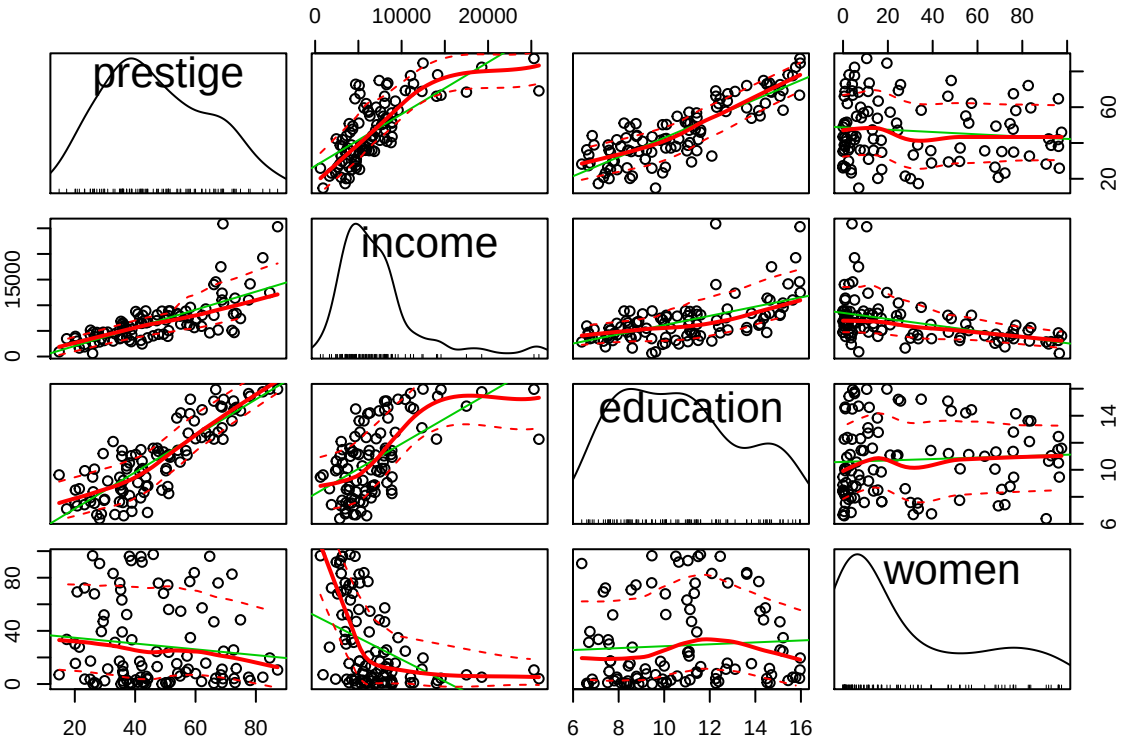

```
##  physicians file.clerks     newsboys      farmers
##          24          41           53           67
```

```
scatterplot(prestige ~ women, data = Prestige, id.n = 4, labels = occ)
```




```
##        physicians             nurses physio.therapsts       babysitters
##                24                 27                 29                 63
```

```
scatterplotMatrix(~ prestige + income + education + women,
                  data = Prestige, span = 0.7)
```



NB., os parâmetros `id.n` e `labels` na função `scatterplot` permitem a identificação automática dos pontos extremos. Pode aprender mais sobre essas opções na página help da função `scatterplot`.

Podemos ver nesta analise exploratória que as variáveis numéricas `prestige` e `education` têm distribuições aceitavelmente normal. Entretanto, as variáveis `income` e `women` têm uma assimetria bastante positiva (a direta). Podemos ver esse problema igualmente no gráfico da distribuição da variável que no scatterplot contra prestígio, onde a falta de linearidade indica que uma transformação pode ajudar dar esta variável um formato que cairia dentro das premissas da regressão.

Podemos anotar também, que `education` e `income` têm correlações bastante altas com a variável dependente `prestige`, mas a porcentagem da mulheres em um grupo ocupacional tem pouco relação (-0,118). Assim, quando construímos o modelo, esta variável explicará pouco da variância. A transformação não vai mudar bastante esta falta de relação.

O `scatterplotMatrix` é uma forma alternativa de apresentar as quatro variáveis numéricas. A função vem da pacote `car` mas uma outra versão fica na pacote `lattice`.

Transformação da Variável `women`

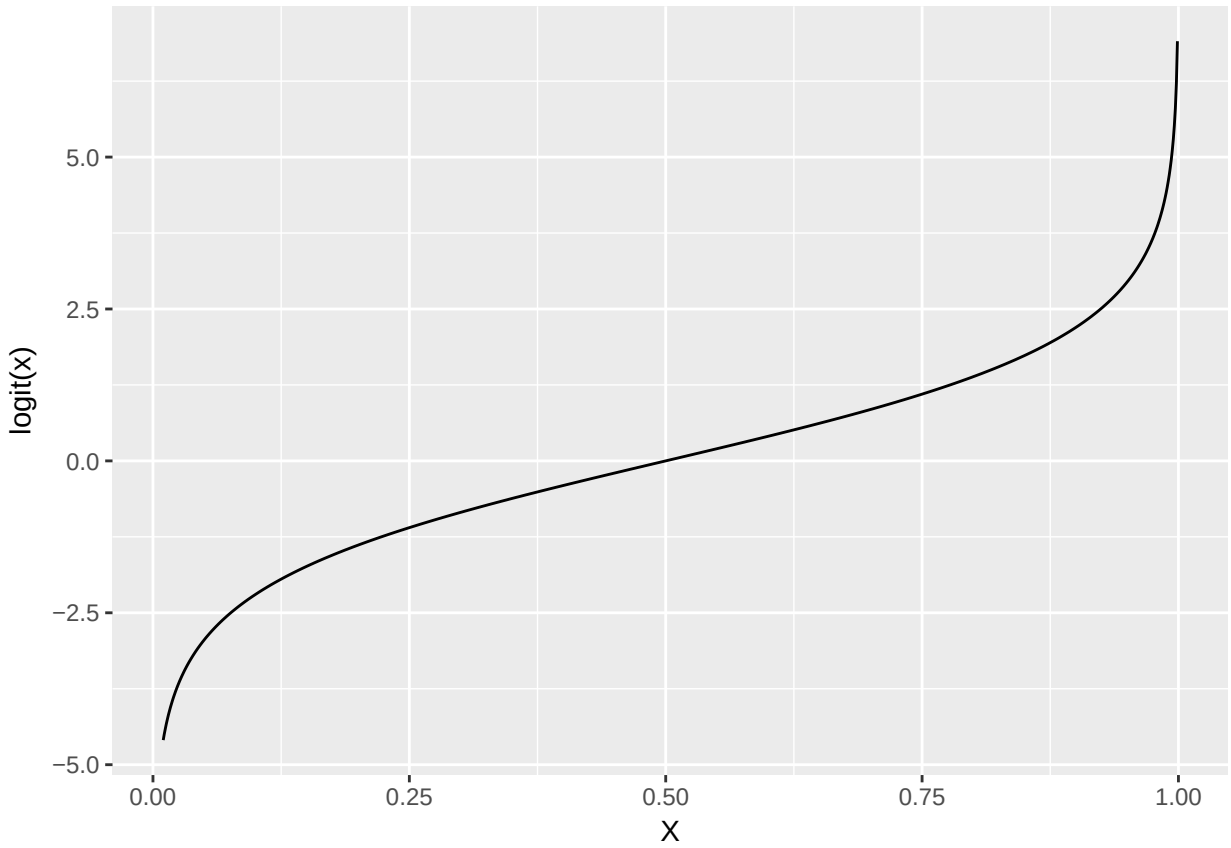
Há duas funções candidatas para transformação de `women`, logarítmica e logit. As duas são variantes da mesma ideia básica. Com a logarítmica, nós simplesmente tomamos o logaritmo dos valores da variável. É importante de lembrar que para os fins de estatística, os logaritmos nas bases diferentes (neperianos, comuns, ou base 2) são equivalentes. Aqui, nós vamos usar o logaritmo de base 10 (comum). Eles todos mudam a escala da variável. Este o que fazemos quando calculamos um fold change da contagem do vírus HIV.

Com o logit, aplicamos a função seguinte para a variável:

$$logit(x) = log_e \frac{x}{1-x}$$

Este função se aplica para variáveis que têm valores entre 0 e 1. A curva de `logit` tem a forma de um "S" que permite conversão de valores entre 0 e 1 em uma linha continua, como a figura abaixo mostra.

```
logitcurve <- tibble(x = seq(.01, .999, .001), logitx = logit(x))
logitgr <- ggplot(data = logitcurve, mapping = aes(x = x, y = logitx))
logitgr <- logitgr + geom_line()
logitgr <- logitgr + labs(x = "X", y = "logit(x)")
logitgr
```



Vamos experimentar com essas duas transformações e ver o efeito delas.

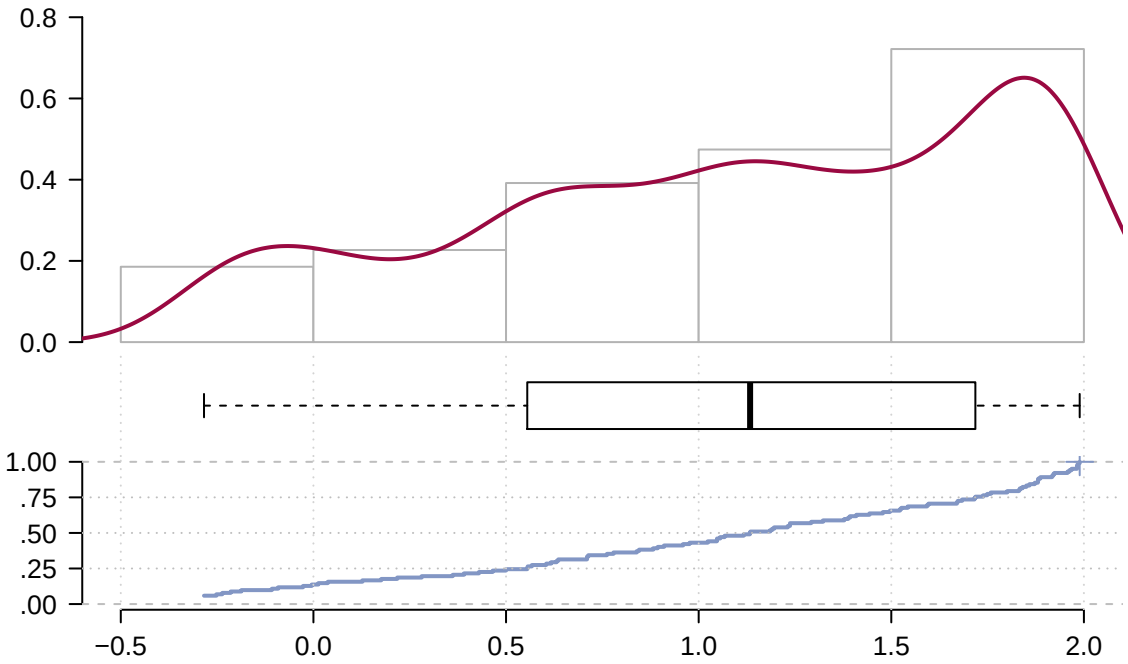
Transformação Log

Nesta transformação, vamos criar uma nova variável (usando `mutate` do pacote `dplyr`) do logaritmo de `women`. Mas, primeiro devemos resolver um problema.

```
logw <- log10(Prestige$women)
Desc(logw, plotit = TRUE)
```

```
## ------------------------------------------------------------------------
## logw (numeric)
##
##   length       n    NAs  unique    0s  mean  meanCI
##      102     102      0      96     0  -Inf      NA
##           100.0%   0.0%          0.0%           NA
##
##      .05     .10    .25  median   .75   .90     .95
##    -0.28   -0.11   0.56    1.13  1.72  1.91    1.97
##
##    range      sd  vcoef     mad   IQR  skew    kurt
##      Inf      NA     NA    0.87  1.16    NA      NA
##
## lowest : -Inf (5), -0.28, -0.25, -0.24, -0.21
## highest: 1.98, 1.98, 1.98, 1.98, 1.99
```

```
## Warning in bplt(at[i], wid = width[i], stats = z$stats[, i], out = z$out[z
## $group == : Outlier (-Inf) in boxplot 1 is not drawn
```



Este cálculo mostra que não podemos calcular um log para todos os valores de `women`. O logaritmo de 0 não existe (R retorna um valor de `-Inf`) e no dataset, `women` mostra 5 profissões em que não tem presença nenhuma das mulheres.

Profissões Sem Mulheres

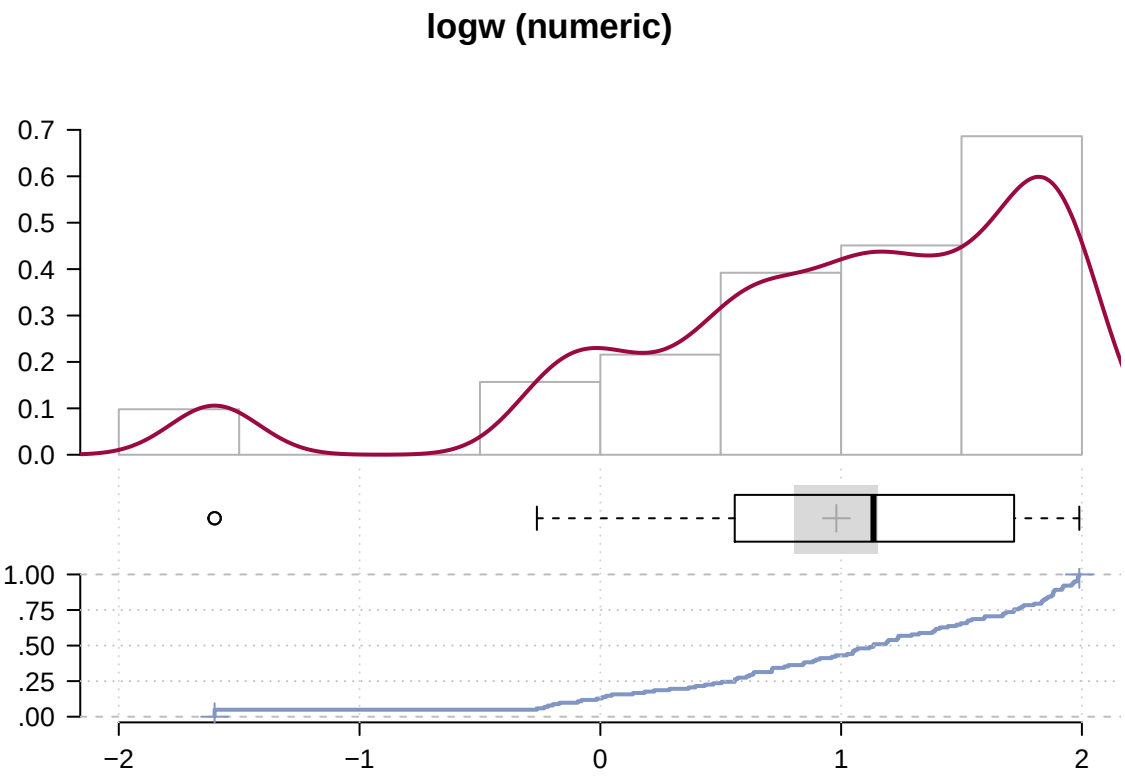
```
##                       education income women prestige census type
## firefighters               9.47   8895     0     43.5   6111   bc
## rotary.well.drillers       8.88   6860     0     35.3   7711   bc
## railway.sectionmen         6.67   4696     0     27.3   8715   bc
## train.engineers            8.49   8845     0     48.9   9131   bc
## longshoremen               8.37   4753     0     26.1   9313   bc
```

Nós podemos lidar com este problema facilmente. Nós vamos aumentar `women` por um pequeno valor fixo, 0,025. Assim, não teremos os logaritmos impossíveis e podemos usar o cálculo. Antes de pôr esta transformação definitivamente no dataset, vamos fazer mais um experimento.

```
logw <- log10(Prestige$women + 0.025)
Desc(logw, plotit = TRUE)
```

```
## ------------------------------------------------------------------------
## logw (numeric)
##
##        length             n         NAs      unique          0s
##           102           102           0          96           0
##                      100.0%        0.0%                   0.0%
##
##           .05           .10         .25      median         .75
##   -0.26206553  -0.09261506  0.55840823  1.13433604  1.71789815
##
##         range            sd       vcoef         mad         IQR
##    3.59122048   0.88367693  0.90152415  0.86435593  1.15948992
```

```
##
##          mean      meanCI
##    0.98020328  0.80663267
##                1.15377390
##
##           .90         .95
##    1.91440358  1.96768689
##
##          skew        kurt
##   -1.15274553  1.19275590
##
## lowest : -1.60205999 (5), -0.26360350, -0.23284413, -0.21824463, -0.19722627
## highest: 1.98224861, 1.9829267, 1.98301704, 1.98477477, 1.98916049
```



Esta vez, temos um resultado bem melhor. Todos os valores têm logaritmos e o boxplot abaixo da curva de densidade mostra que a distribuição dentro da IQR (interquartile range) fica simétrica.

Transformação Logit

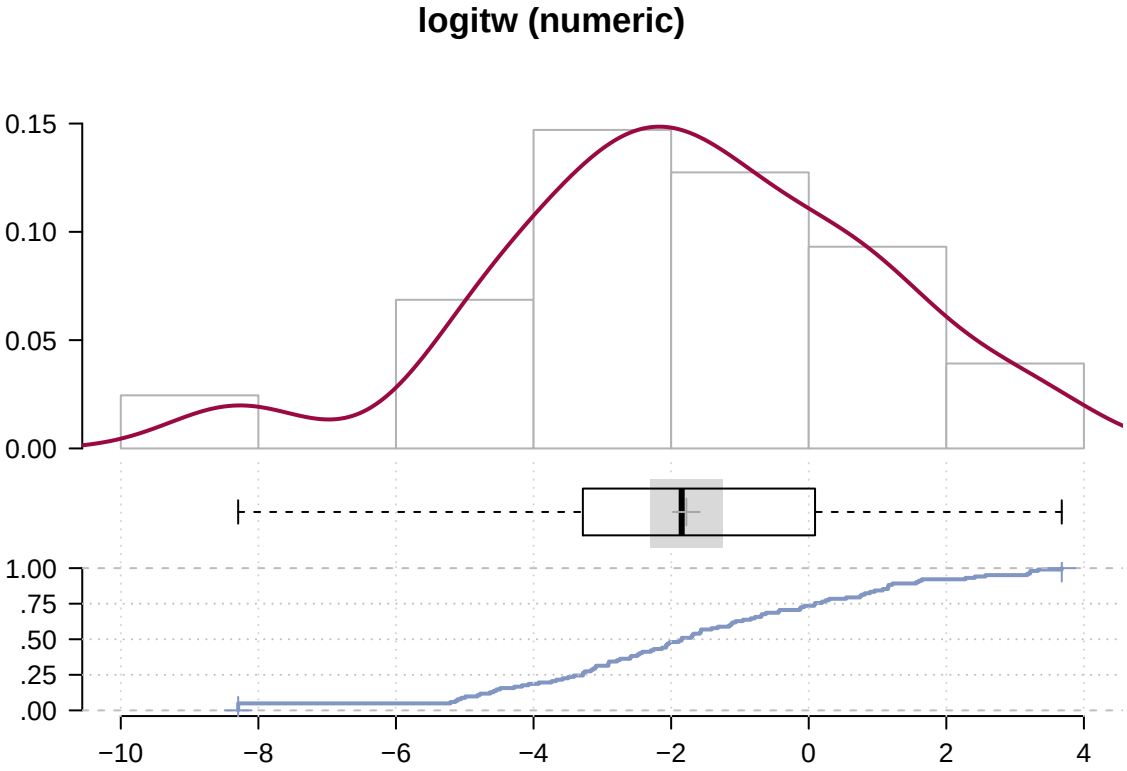
Agora, vamos experimentar com a transformação logit. O pacote `car` tem uma função `logit` que calcula diretamente a transformação usando a formula acima. Porque a função vai tomar um logaritmo, precisamos de novo corrigir os valores de `women` somando 0.025 aos valores existentes.

A função logit precisa também ter os números entre 0 e 1. Então, nós vamos dividir `women` por 100 para pôr de volta em termos decimais invés das porcentagens em que eles aparecem no dataset.

```
logitw <- car::logit((Prestige$women + 0.025)/100)
Desc(logitw, plotit = TRUE)
```

```
## ------------------------------------------------------------------------
```

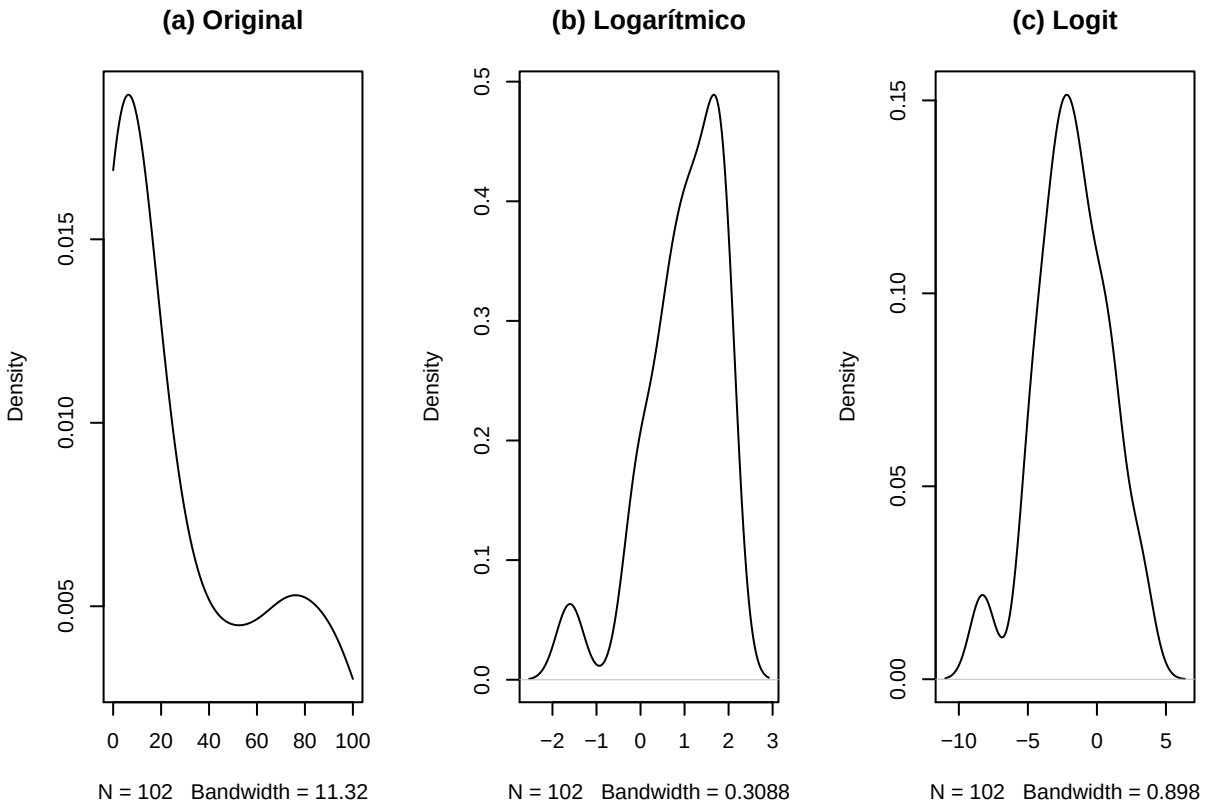
```
## logitw (numeric)
##
##       length            n          NAs       unique           0s         mean
##          102          102            0           96            0  -1.7784686
##                   100.0%         0.0%                      0.0%
##
##          .05          .10          .25       median          .75          .90
##   -5.2031133   -4.8103114   -3.2825422   -1.8467930    0.0891595    1.5286568
##
##        range           sd        vcoef          mad          IQR         skew
##   11.9718191    2.6661995   -1.4991547    2.5664699    3.3717017   -0.2678432
##
##       meanCI
##   -2.3021599
##   -1.2547773
##
##          .95
##    2.5613734
##
##         kurt
##    0.0516243
##
## lowest : -8.2937996 (5), -5.2066748, -5.1354464, -5.1016286, -5.0529302
## highest: 3.1767525, 3.2164865, 3.221896, 3.3331889, 3.6780195
```



O gráfico seguinte mostra um resumo das transformações

```
par(mfrow = c(1, 3))
with(Prestige, {
   plot(density(women, from = 0, to = 100),
```

```
        main = "(a) Original")
   plot(density(logw), main = "(b) Logarítmico")
   plot(density(logitw), main = "(c) Logit")
})
```




```
par(mfrow = c(1, 1))
```

Neste caso, as duas transformações produziram resultados semelhantes e podemos trabalhar com a transformação logit.

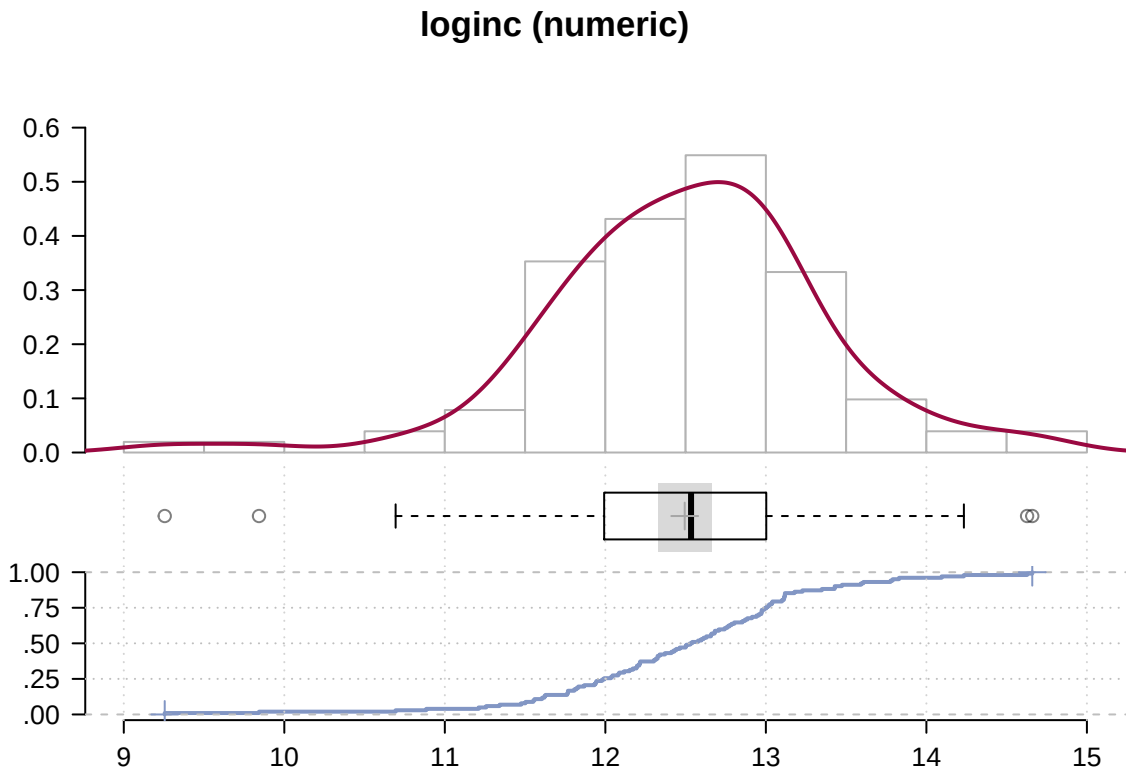
Transformação da Variável `income`

Dado a experiência que temos com a distribuição da renda numa população, podemos usar a transformação logarítmico para nossa variável `income`.

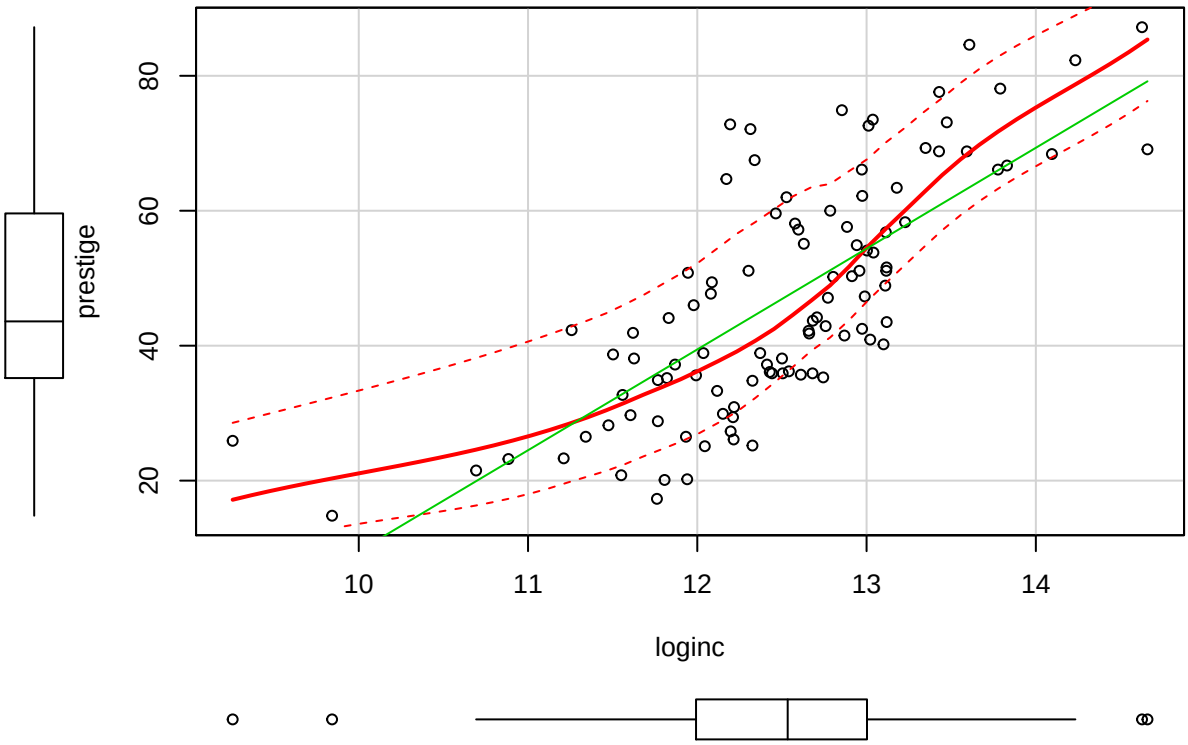
```
loginc <- log2(Prestige$income)
Desc(loginc, plotit = TRUE)
```

```
## ------------------------------------------------------------------------
## loginc (numeric)
##
##      length          n        NAs     unique         0s       mean
##         102        102          0        100          0  12.494472
##                  100.0%       0.0%                0.0%
##
##         .05        .10        .25     median        .75        .90
##   11.261567  11.563127  12.003396  12.533921  12.999152  13.429054
##
```

```
##       range         sd      vcoef        mad        IQR       skew
##    5.404466   0.853281   0.068293   0.724458   0.995756  -0.504819
##
##      meanCI
##   12.326872
##   12.662073
##
##         .95
##   13.789169
##
##        kurt
##    1.929901
##
## lowest : 9.255029, 9.84235, 10.693487, 10.884171, 11.210671
## highest: 13.829525, 14.094902, 14.233545, 14.627306, 14.659494
```




```
occ <- rownames(Prestige)
scatterplot(prestige ~ loginc, data = Prestige, labels = occ)
```



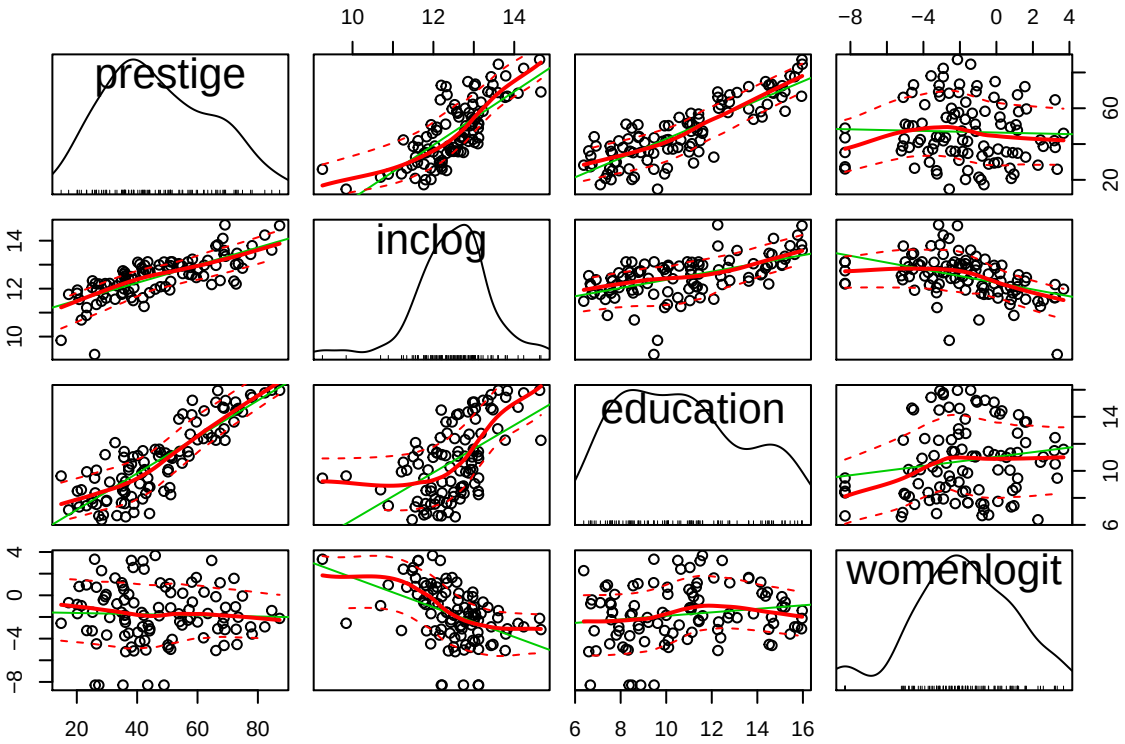
Agora, a distribuição da renda tem um formato muito mais normal e a transformação tira o formato de segundo grau que podíamos ver no scatterplot da renda não modificado.

Podemos inserir essas transformações em nosso dataset e olhar de novo nas correlações e o `scatterplotMatrix` das variáveis. Anote que preciso pôr de volta os nomes para as profissões porque `mutate` tira eles quando cria novas colunas.

```
Prestige <- Prestige %>% mutate(womenlogit = logit((women + 0.025)/100),
                    inclog = log2(income))
rownames(Prestige) <- occ
cor(Prestige[,c(4,1,7:8)])
```

```
##                prestige education  womenlogit     inclog
## prestige    1.00000000 0.8501769 -0.03476255  0.7410561
## education   0.85017689 1.0000000  0.16670369  0.5481051
## womenlogit -0.03476255 0.1667037  1.00000000 -0.4386544
## inclog      0.74105613 0.5481051 -0.43865438  1.0000000
```

```
scatterplotMatrix(~ prestige + inclog + education + womenlogit,
                  data = Prestige, span = 0.7)
```



Regressão com os Regressores Numéricos

Nós vamos começar com um modelo de regressão com somente os variáveis numéricos: `income` (modificada a `inclog`), `education` e `women` (modificada a `womenlogit`). O quarto regressor, `type` é uma variável categórica e precisa de tratamento especial que explicarei na próxima parte deste aula.

Para construir o modelo, nós vamos usar os mesmos símbolos que na última aula (regressão polinomial) na formula: "~" (til) para separar a variável dependente e as independentes. E, usamos o "+" para separar as variáveis independentes. Existem outros símbolos que podem ser usados, mas relatam para tipos de modelos mais avançados. Lembre que a notação de formula em R **não** demanda que você especifica o dataframe antes de todas as variáveis porque a função `lm` reconhece o data frame no parâmetro `data =`.

```
fit1 <- lm(prestige ~ inclog + education + womenlogit, data = Prestige)
summary(fit1)
```

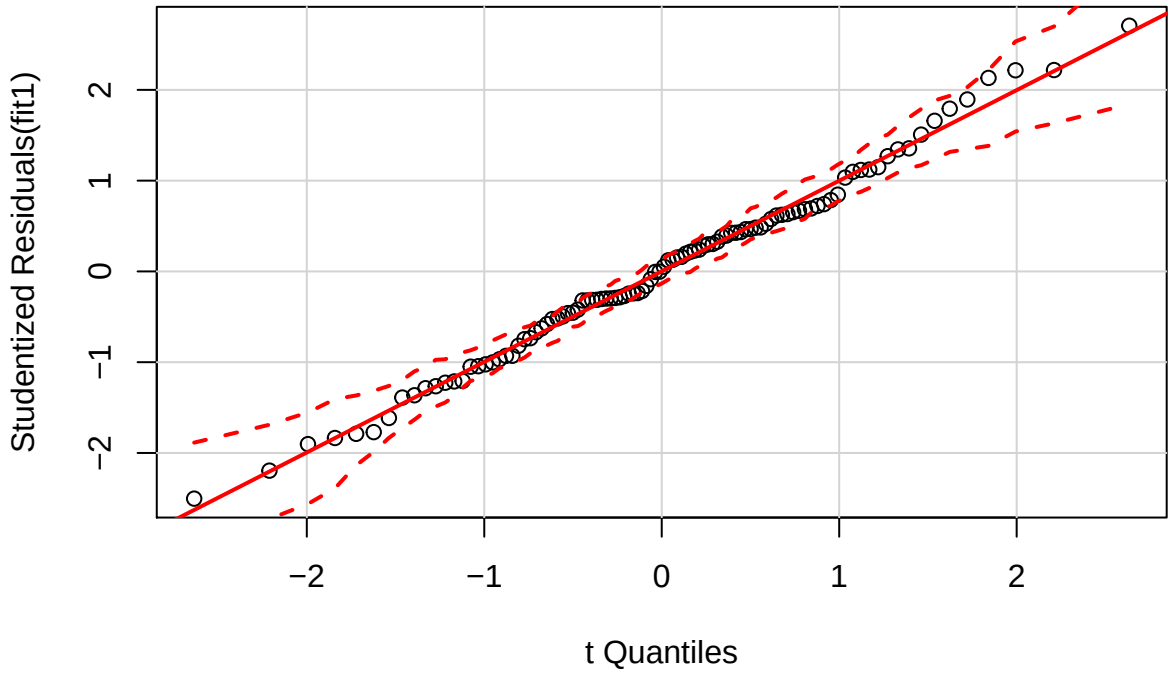
```
##
## Call:
## lm(formula = prestige ~ inclog + education + womenlogit, data = Prestige)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.2972  -4.1876   0.1766   4.3429  18.4359
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -103.0222    13.3980  -7.689 1.16e-11 ***
## inclog         8.7821     1.2999   6.756 1.02e-09 ***
## education      3.7967     0.3705  10.247  < 2e-16 ***
## womenlogit     0.3609     0.3529   1.023    0.309
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.143 on 98 degrees of freedom
## Multiple R-squared:  0.8327, Adjusted R-squared:  0.8276
## F-statistic: 162.6 on 3 and 98 DF,  p-value: < 2.2e-16
```

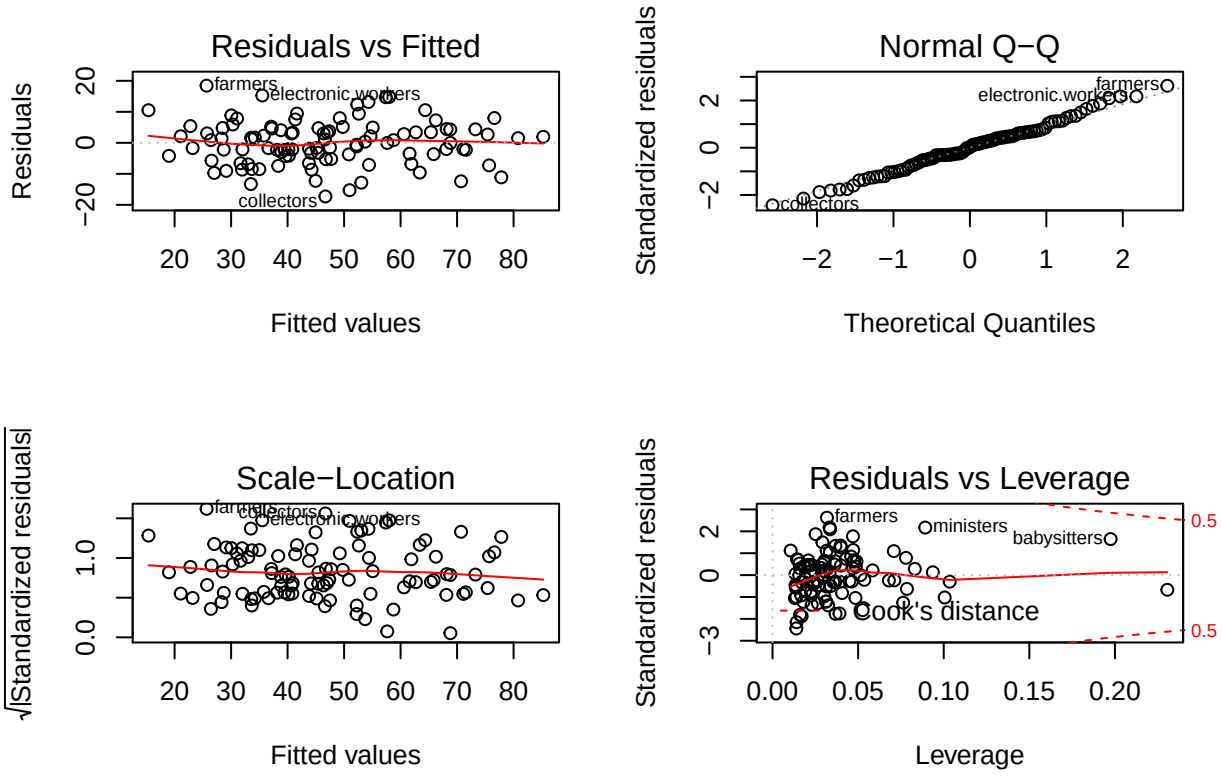
```
anova(fit1)
```

```
## Analysis of Variance Table
##
## Response: prestige
##            Df  Sum Sq Mean Sq  F value Pr(>F)
## inclog      1 16417.5 16417.5 321.7644 <2e-16 ***
## education   1  8424.3  8424.3 165.1066 <2e-16 ***
## womenlogit  1    53.4    53.4   1.0456  0.309
## Residuals  98  5000.3    51.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
qqPlot(fit1)
```




```
par(mfrow=c(2,2))
plot(fit1)
```




```
par(mfrow=c(1,1))
```

Os resultados mostram que o modelo descreve bem a relação entre prestígio e as previsores. Como pode ser visto na quadra ANOVA, a variável `education` tem a melhor relação a variável dependente `prestige` com a valor t de 10.247 seguida pela variável transformada da renda, `inclog`. Finalmente, a contribuição da participação feminina tem efeito insignificante sobre prestígio de uma ocupação (p = 0.309).

Os gráficos analíticos mostram que não tem problemas com as premissas de linearidade, independência e normalidade no modelo.

Então este modelo diz: prestige = -105.42 + 9.00 * log_2(income) + 3.78 * education + 0.62 * participação feminina

Este modelo explica aproximadamente 83% de variância dos dados.

Opção - Modelo Sem Variável `women`

Porque a variável women parece de contribuir pouco para o modelo, vamos considerar uma versão sem esta variável

```
fit2 <- lm(prestige ~ inclog + education, data = Prestige)
summary(fit2)
```

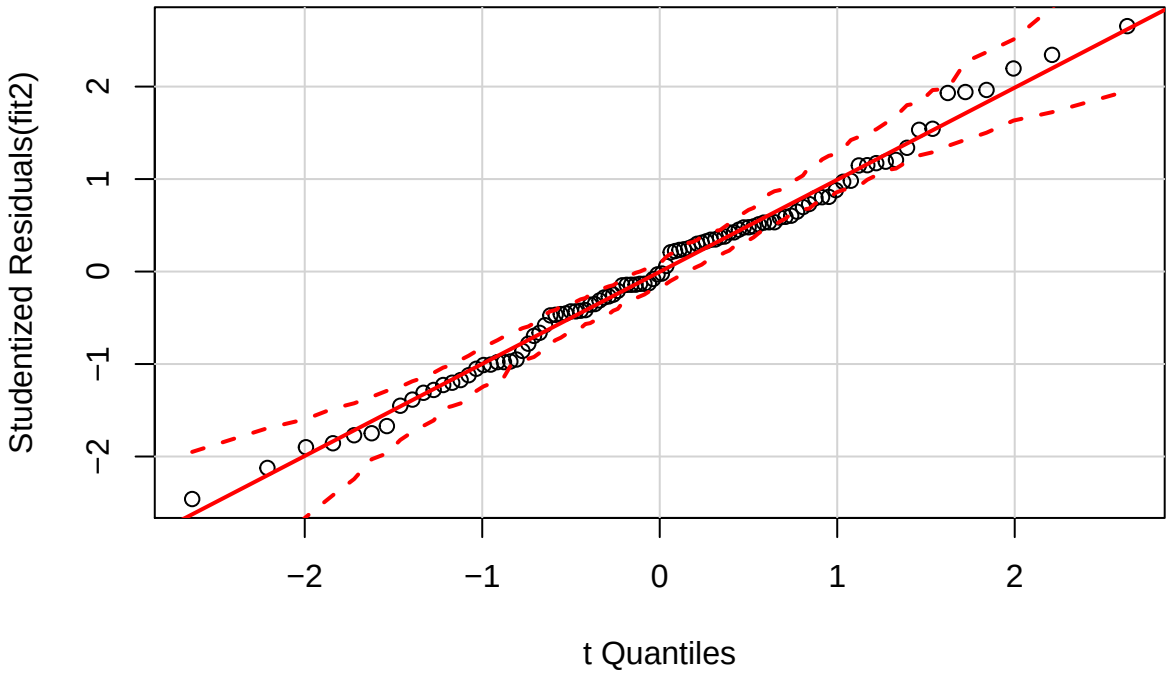
```
##
## Call:
## lm(formula = prestige ~ inclog + education, data = Prestige)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.0346  -4.5657  -0.1857   4.0577  18.1270
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -95.1940    10.9979  -8.656 9.27e-14 ***
## inclog        7.9278     0.9961   7.959 2.94e-12 ***
## education     4.0020     0.3115  12.846  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.145 on 99 degrees of freedom
## Multiple R-squared:  0.831,  Adjusted R-squared:  0.8275
## F-statistic: 243.3 on 2 and 99 DF,  p-value: < 2.2e-16
```

```
anova(fit2)
```

```
## Analysis of Variance Table
##
## Response: prestige
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## inclog     1 16417.5 16417.5  321.62 < 2.2e-16 ***
## education  1  8424.3  8424.3  165.03 < 2.2e-16 ***
## Residuals 99  5053.6    51.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
qqPlot(fit2)
```

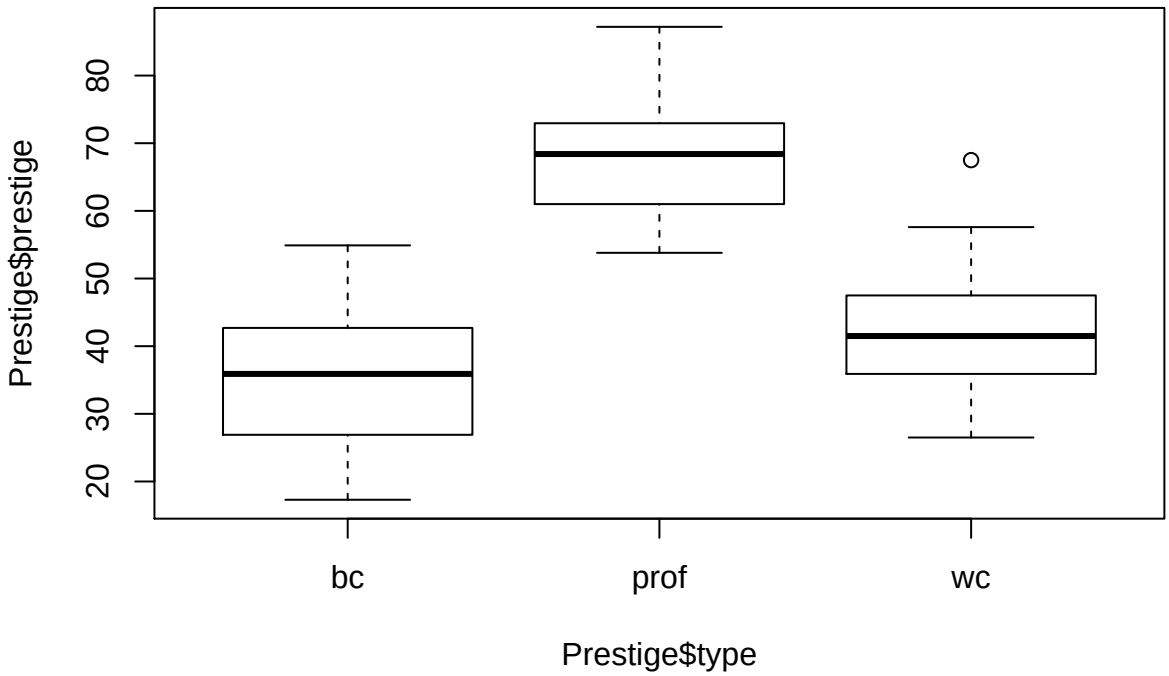


Como indica os R^2, o modelo fica o mesmo sem a variável `women`. Seguindo a dica da Navalha de Ockham (se você ter um número menor dos termos e não perder informação, melhor), vamos deixar fora das versões futuros.

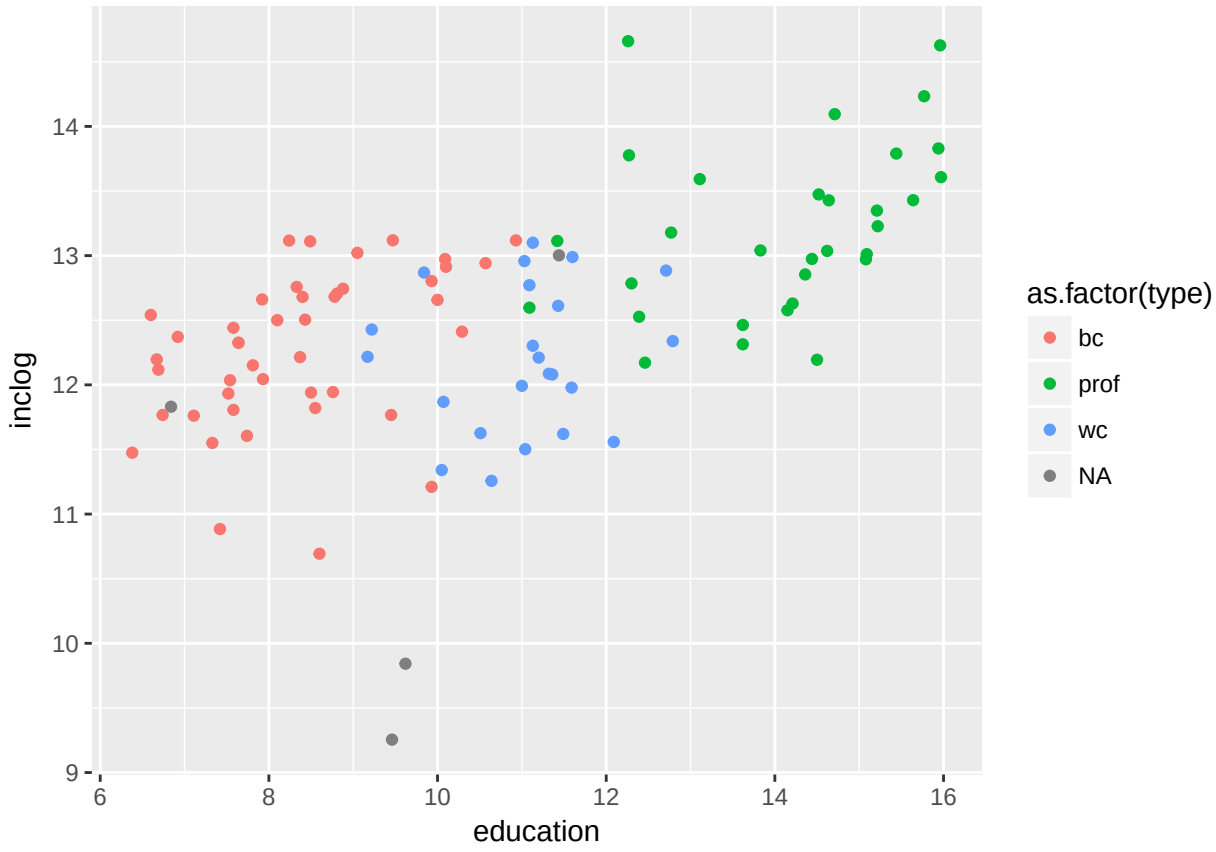
Inclusão da Variável `type` no modelo

A variável `type` é nominal e tem três categorias: `bc` (operário), `prof` (profissional-executivo) e `wc` (colarinho branco). Nós podemos mostrar o efeito desta variável em dois gráficos: o primeiro mostrando a distribuição de `prestige` para cada grupo e o segundo mostrando a interação entre `education`, `inclog` e `type`, retratando `type` em cores.

```
plot(Prestige$prestige ~ Prestige$type)
```




```
qplot(education, inclog, data = Prestige, col = as.factor(type))
```



Estes dois gráficos claramente mostram que o prestígio de cada `type` é bastante diferente. O scatterplot também mostra que type tem uma forte relação com renda e com educação porque os três cores agrupam em três áreas distintas do espaço (faltando só uma pouca sobreposição). Uma das coisas que tentamos de fazer com regressão e outros modelos estatísticos é definir claramente essas diferenças entre grupos para ajudar nos processos de previsão e classificação.

A inclusão das variáveis nominais é possível e até muito importante (se a variável seja gênero ou raça, por exemplo). Entretanto, precisa um pouco de cuidado na especificação deles no modelo.

A primeira coisa que precisamos fazer é acertar que a variável `type`, nossa variável nominal, fica na forma de um `factor` em R. Neste caso, a variável vem nessa forma. Se fosse necessário para modificar ela para um `factor`, precisa chamar a função `factor` para fazer: `type <- factor(type)`.

Quando incluímos `type` numa formula de modelo em R, o programa automaticamente cria regressores chamados `contrasts` para os níveis da variável. Podemos ver esses `contrasts` no formato do matriz do modelo do `type`, ou seja, o formato em que o programa trata dos níveis da variável. Aqui, mostro só alguns casos. (Lembre que regressão está calculado no formato de matrizes usando álgebra linear dentro do programa. Esta é uma das poucas vezes que vou mostrar o que acontece dentro da "caixa preta".)

```
kable(with(Prestige, model.matrix(~ type)[c(1:5, 50:55),]))
```

	(Intercept)	typeprof	typewc
1	1	1	0
2	1	1	0
3	1	1	0
4	1	1	0
5	1	1	0
51	1	0	1
52	1	0	1

	(Intercept)	typeprof	typewc
54	1	0	0
55	1	0	1
56	1	0	1
57	1	0	1

A primeira coluna do matriz é todos 1's que representa o intercepto do modelo. As outras colunas representam variáveis "dummy" que o software criou para executar o modelo. Normalmente, quando temos fatores, nós usamos uma técnica de estatística chamado Analise de Variância (ANOVA). Mas, ANOVA e regressão são primas muito próximas e podemos executar nosso modelo no formato de regressão.

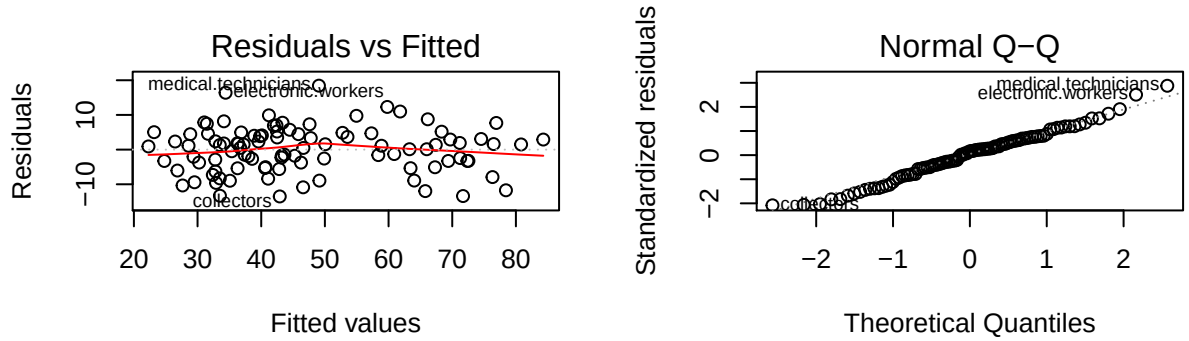
```
fit3 <- lm(prestige ~ inclog + education + type, data = Prestige)
summary(fit3)
```

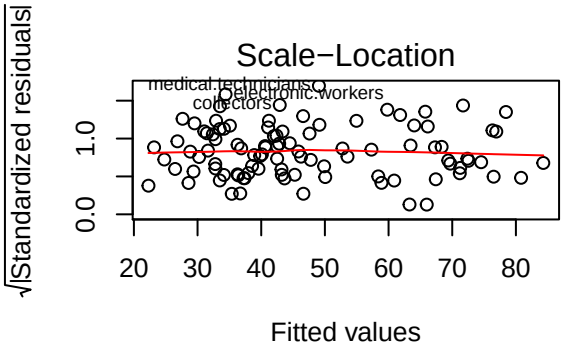
```
##
## Call:
## lm(formula = prestige ~ inclog + education + type, data = Prestige)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.511  -3.746   1.011   4.356  18.438
##
## Coefficients:
##             Estimate Std. Error t value     Pr(>|t|)
## (Intercept) -81.2019    13.7431  -5.909 0.0000000563 ***
## inclog        7.2694     1.1900   6.109 0.0000000231 ***
## education     3.2845     0.6081   5.401 0.0000005058 ***
## typeprof      6.7509     3.6185   1.866       0.0652 .
## typewc       -1.4394     2.3780  -0.605       0.5465
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.637 on 93 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.8555, Adjusted R-squared:  0.8493
## F-statistic: 137.6 on 4 and 93 DF,  p-value: < 2.2e-16
```

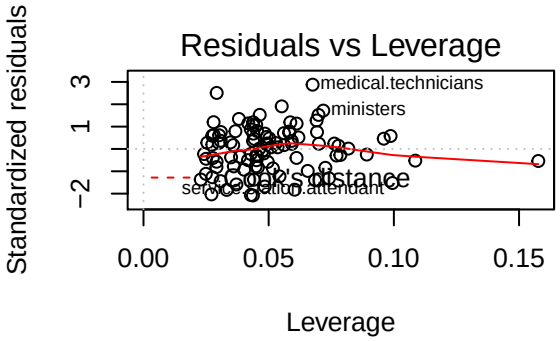
```
anova(fit3)
```

```
## Analysis of Variance Table
##
## Response: prestige
##           Df  Sum Sq Mean Sq  F value    Pr(>F)
## inclog     1 15998.5 15998.5 363.2209 < 2.2e-16 ***
## education  1  7783.1  7783.1 176.7028 < 2.2e-16 ***
## type       2   469.1   234.5   5.3247  0.006465 **
## Residuals 93  4096.3    44.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

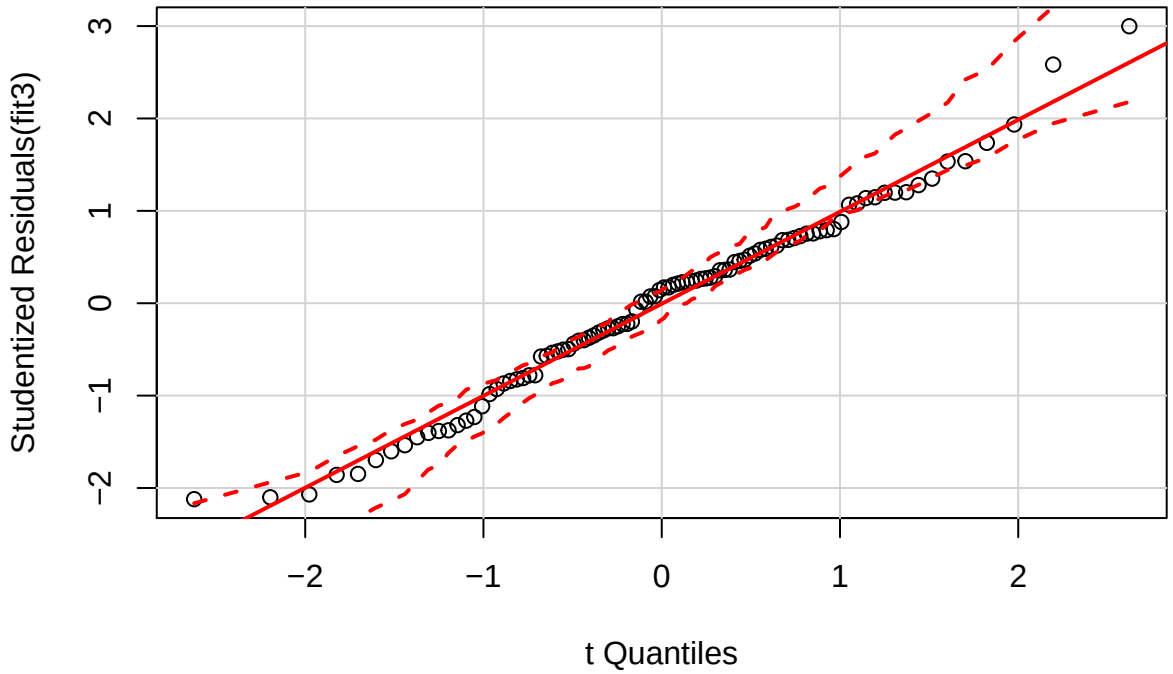
```
par(mfrow=c(2,2))
plot(fit3)
```








```
par(mfrow=c(1,1))
qqPlot(fit3)
```



Este modelo não mudou o resultado final muito. O R^2 aumentou um pouco e todas as três variáveis parecem de ter um efeito significativo no modelo. Realmente, inclui a variável nominal para mostrar que pode ser feito. Mas, baseado em o que vimos nos gráficos anteriores, ela parece de dizer muito da mesma coisa que renda e educação. Então, este representa um caso em que tem "autocorrelação", ou seja, as variáveis não são realmente independentes porque expliquem o mesmo comportamento. Por exemplo, ocupações "blue-collar"

normalmente ganham menos que uma ocupação "profissional". Então as duas variáveis estão explicando o mesmo fenômeno. Autocorrelação é um efeito estatístico que pode estragar os modelos de regressão, mas neste caso estava benigna.

A função `aov` produz uma tabela ANOVA. Esta é exatamente a mesma que a regressão múltipla produz. Eles são os mesmos modelos.

Nós podemos avaliar a relação entre renda e anos de educação e `type` por incluir no modelo diretamente a interação entre essas variáveis. Usamos o símbolo ":" (dois pontos) para criar um termo para interação no modelo. Vamos olhar primeiro nas variáveis "dummy" que a regressão criará para `education` and `type` e depois executar o modelo com os dois novos termos de interação.

```
model.matrix(~ type + education + education:type, data = Prestige)[c(1:5, 50:55),]
```

```
##                            (Intercept) typeprof typewc education
## gov.administrators                   1        1      0     13.11
## general.managers                     1        1      0     12.26
## accountants                          1        1      0     12.77
## purchasing.officers                  1        1      0     11.42
## chemists                             1        1      0     14.62
## commercial.travellers                1        0      1     11.13
## sales.clerks                         1        0      1     10.05
## service.station.attendant            1        0      0      9.93
## insurance.agents                     1        0      1     11.60
## real.estate.salesmen                 1        0      1     11.09
## buyers                               1        0      1     11.03
##                            typeprof:education typewc:education
## gov.administrators                      13.11             0.00
## general.managers                        12.26             0.00
## accountants                             12.77             0.00
## purchasing.officers                     11.42             0.00
## chemists                                14.62             0.00
## commercial.travellers                    0.00            11.13
## sales.clerks                             0.00            10.05
## service.station.attendant                0.00             0.00
## insurance.agents                         0.00            11.60
## real.estate.salesmen                     0.00            11.09
## buyers                                   0.00            11.03
```

```
fit5 <- lm(prestige ~ inclog + education + type +
             inclog:type + education:type, data = Prestige)
summary(fit5)
```

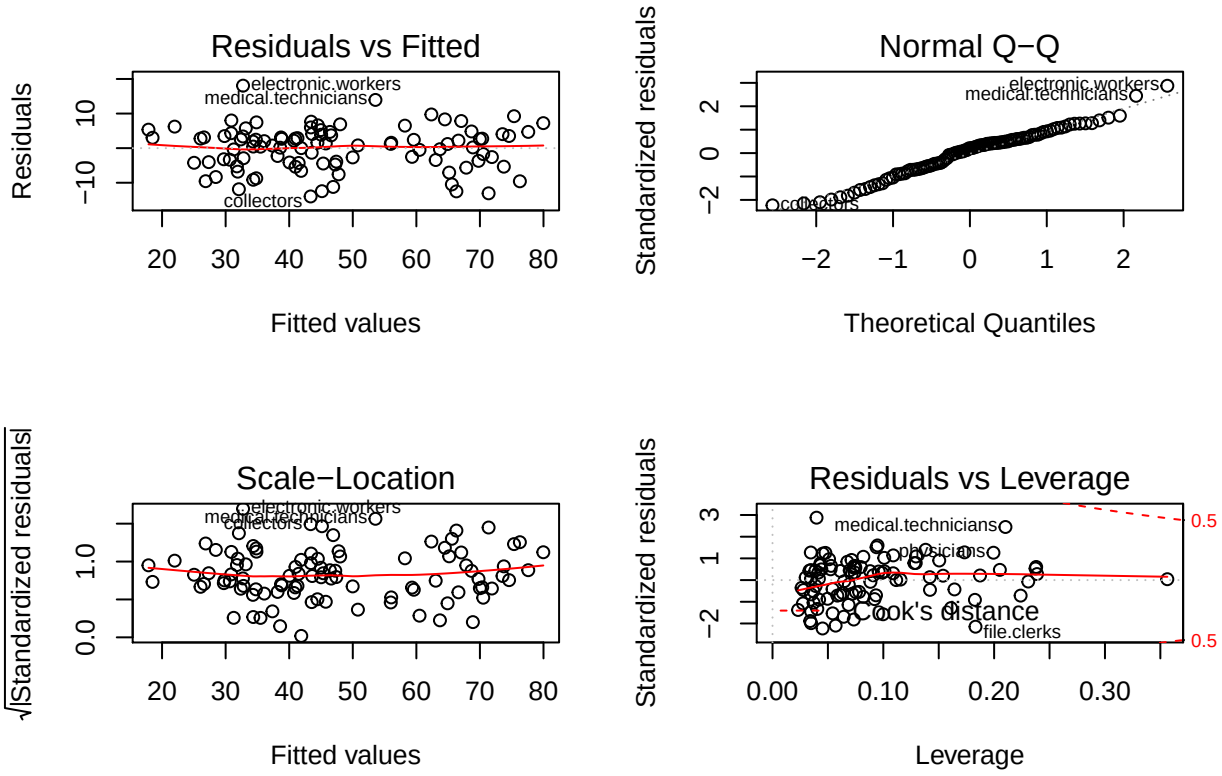
```
##
## Call:
## lm(formula = prestige ~ inclog + education + type + inclog:type +
##     education:type, data = Prestige)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.970  -4.124   1.206   3.829  18.059
##
## Coefficients:
##                     Estimate Std. Error t value     Pr(>|t|)
## (Intercept)         -120.0459    20.1576  -5.955 0.0000000507 ***
## inclog                11.0782     1.8063   6.133 0.0000000232 ***
```

```
## education             2.3357     0.9277   2.518      0.01360 *
## typeprof             85.1601    31.1810   2.731      0.00761 **
## typewc               30.2412    37.9788   0.796      0.42800
## inclog:typeprof      -6.5356     2.6167  -2.498      0.01434 *
## inclog:typewc        -5.6530     3.0519  -1.852      0.06730 .
## education:typeprof    0.6974     1.2895   0.541      0.58998
## education:typewc      3.6400     1.7589   2.069      0.04140 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.409 on 89 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.871,  Adjusted R-squared:  0.8595
## F-statistic: 75.15 on 8 and 89 DF,  p-value: < 2.2e-16
```

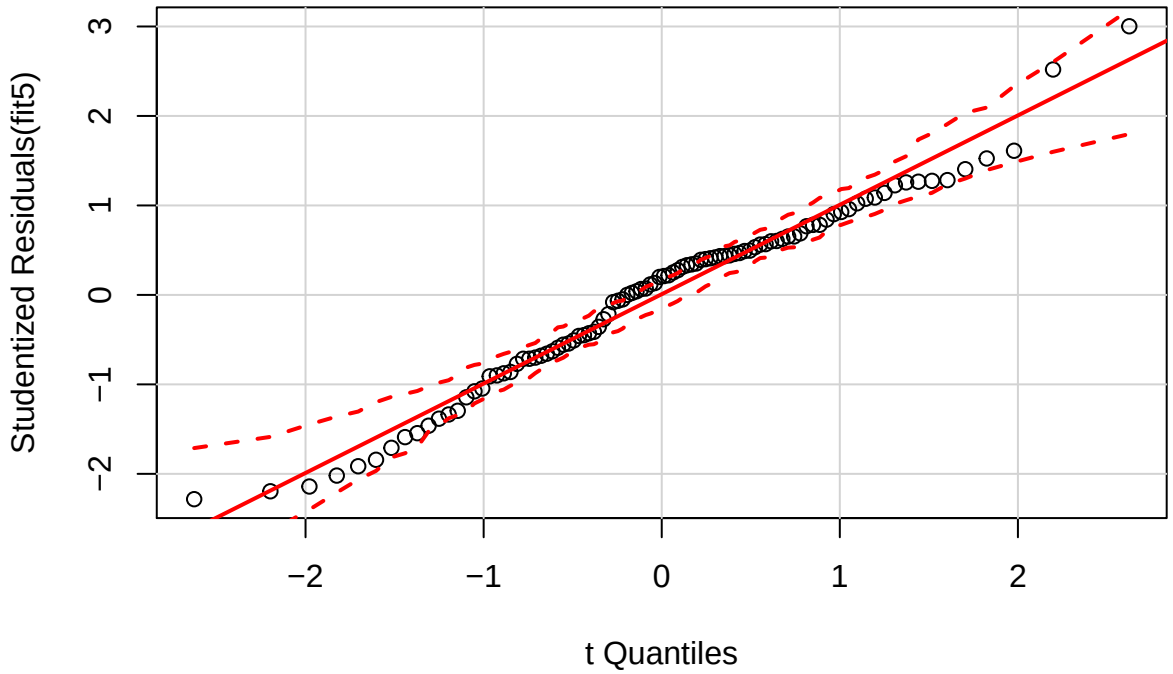
```
anova(fit5)
```

```
## Analysis of Variance Table
##
## Response: prestige
##                Df  Sum Sq Mean Sq  F value    Pr(>F)
## inclog          1 15998.5 15998.5 389.5234 < 2.2e-16 ***
## education       1  7783.1  7783.1 189.4987 < 2.2e-16 ***
## type            2   469.1   234.5   5.7103  0.004642 **
## inclog:type     2   262.1   131.1   3.1911  0.045873 *
## education:type  2   178.8    89.4   2.1762  0.119474
## Residuals      89  3655.4    41.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,2))
plot(fit5)
```




```
par(mfrow=c(1,1))
qqPlot(fit5)
```



Como nós podemos ver na tabela dos coeficientes, `inclog` agora é tem o maior valor t e representa a variável que melhor explique o resultado `prestige`. Por causa do impacto das interações entre `type` e `education`, essas variáveis parecem de perder muito força na explicação de como Canadenses avaliam o prestígio das ocupações. O modelo agora diz que alguém com uma ocupação que renda muito vai ser mais bem pensado que alguém que tem uma ocupação que demanda muitos anos de escolaridade mas não renda tanto. (Pense

em nos!) Agora, o modelo está tomando em conta a autocorrelação entre `education` e `type`. Assim, temos uma visão mais sofisticada de prestígio que nos modelos anteriores. O valor p da interação entre `inclog` e a categoria `prof` do `type` reforça essa ideia que as ocupações com o maior prestígio são aquelas que são classificadas como profissional **e** rendam mais.

Com regressão podemos construir modelos que vão bem além as análises simplórias que estudamos antes.

Na próxima aula, apresentarei a **regressão logistica**, um tipo de regressão que podemos usar quando temos uma variável dependente binária. Por exemplo, quando queremos prever se um paciente tem ou não tem infecção de HIV, nós podemos avaliar vários fatores quantitativos e qualitativos para estimar a probabilidade que o paciente tem o vírus (mesmo antes de receber os resultados do testes de carga viral e das células T CD4+).