Universidade Federal do Ceará

Departamento de Economia Agrícola

Programa de pós-graduação em Economia Rural

Introdução à análise de dados em R

Organizador: Professor Edward Martins Costa

Ministrante: Helson Gomes de Souza

Aula V: Estatística aplicada com R

1. Modelos Lineares

1.1 O Modelo Linear Geral

Os modelos lineares são aqueles com forma funcional linear tanto na parte independente quanto na parte explicada. Esses modelos podem ser estimados por qualquer estimador, mas são convencionalmente atribuídos ao estimador de Mínimos Quadrados Ordinários (MQO). Em suma, um modelo linear simples possui a seguinte especificação:

$$Y = \alpha + \beta X + \varepsilon$$

Em que Y é uma variável dependente, α é o intercepto da reta de regressão, β é o coeficiente angular da reta de regressão e ε é o termo de erro idiossincrático que mede a diferença entre o valor predito e o valor real de cada observação. Estimar esta equação por MQO significa encontrar um valor para α e β que satisfaçam a seguinte condição:

$$min \left(\sum (Y - \alpha - \beta X)^2 \right)$$

Caso Y seja explicado por mais de uma variável, podemos descrever X como sendo uma matriz de variáveis explicativas. Tem-se então um modelo linear múltiplo ou geral.

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} + \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix} \begin{pmatrix} X_{1,1} & X_{1,2} & \dots & X_{1,n} \\ X_{2,1} & X_{2,2} & \dots & X_{2,n} \\ \vdots & \vdots & \dots & \vdots \\ X_{n,1} & X_{n,2} & \dots & X_{n,n} \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

É conhecido que tanto no caso do modelo linear simples quanto no caso do modelo linear geral, a estimação via MQO gera um valor para β equivalente a:

$$\beta = (X'X)^{-1}X'Y$$

Em R, a estimação de um modelo linear via MQO é feita com o uso da função *lm()*. Para demonstrar, vamos usar a base de carros disponível no R para medir o efeito da quantidade de cilíndros no desempenho do carro.

In [4]:
```
# Regressão linear simples
modelo_linear  <- lm(formula = mpg ~ cyl, data = mtcars)
summary(modelo_linear)
```

```
Call:
lm(formula = mpg ~ cyl, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.9814 -2.1185  0.2217  1.0717  7.5186 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  37.8846     2.0738   18.27  < 2e-16 ***
cyl          -2.8758     0.3224   -8.92 6.11e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.206 on 30 degrees of freedom
Multiple R-squared:  0.7262,    Adjusted R-squared:  0.7171 
F-statistic: 79.56 on 1 and 30 DF,  p-value: 6.113e-10
```

1.2 Regressão linear clusterizada

Em uma regressão linear clusterizada existe uma dupla matriz $(Y_{i,g}, X_{i,g})$ em que i se refere ao indivíduo em específico e g é um cluster que subdivide as informações, sendo $i = 1, \ldots n_g$ e $g = 1, \ldots G$. Deixe $\boldsymbol{Y}_g = \left(Y_{1g}, \ldots, Y_{n_g g}\right)'$ e $\boldsymbol{X}_g = \left(X_{1g}, \ldots, X_{n_g g}\right)'$ representarem o n_g x 1 vetor de variáveis dependentes e a matriz n_g k de regressores do g-ésimo cluster. Podemos escrever a regressão clusterizada como:

$$Y_{ig} = X'_{ig}\beta + e_{ig}$$

ou

$$Y_g = X_g\beta + e_g$$

Em que $\boldsymbol{e}_g = \left(e_{1g}, \ldots, e_{n_g g}\right)'$ é um vetor $n_g \times 1$ de distúrbios aleatórios. O estimador MQO pode ser derivado como:

$$\begin{aligned}\widehat{\beta} &= \left(\sum_{g=1}^{G}\sum_{i=1}^{n_g} X_{ig}X_{ig}'\right)^{-1}\left(\sum_{g=1}^{G}\sum_{i=1}^{n_g} X_{ig}Y_{ig}\right)\\ &= \left(\sum_{g=1}^{G} X_g'X_g\right)^{-1}\left(\sum_{g=1}^{G} X_g'Y_g\right)\\ &= \left(X_g'X_g\right)^{-1}\left(X_g'Y_g\right)\end{aligned}$$

Nesse caso, a média do estimador MQO pode ser obtida fazendo:

$$\begin{aligned}\widehat{\beta} - \beta &= \left(\sum_{g=1}^{G} \boldsymbol{X}_g'\boldsymbol{X}_g\right)^{-1}\left(\sum_{g=1}^{G} \boldsymbol{X}_g'\boldsymbol{e}_g\right)\\ E[\widehat{\beta} - \beta \mid X] &= \left(\sum_{g=1}^{G} X_g'X_g\right)^{-1}\left(\sum_{g=1}^{G} X_g'E\left[e_g \mid X\right]\right)\\ &= \left(\sum_{g=1}^{G} X_g'X_g\right)^{-1}\left(\sum_{g=1}^{G} X_g'E\left[e_g \mid X_g\right]\right)\\ &= 0\end{aligned}$$

E a variância do estimador MQO é obtida fazendo:

$$\begin{aligned}\operatorname{var}\left[\left(\sum_{g=1}^{G} \boldsymbol{X}_g'\boldsymbol{e}_g\right) \mid \boldsymbol{X}\right] &= \sum_{g=1}^{G} \operatorname{var}\left[\boldsymbol{X}_g'\boldsymbol{e}_g \mid \boldsymbol{X}_g\right]\\ &= \sum_{g=1}^{G} \boldsymbol{X}_g'\mathbb{E}\left[\boldsymbol{e}_g\boldsymbol{e}_g' \mid \boldsymbol{X}_g\right]\boldsymbol{X}_g\\ &= \sum_{g=1}^{G} \boldsymbol{X}_g'\Sigma_g\boldsymbol{X}_g\\ &\overset{\text{def}}{=} \Omega_n\\ \boldsymbol{V}_{\widehat{\beta}} = \operatorname{var}[\hat{\beta} \mid \boldsymbol{X}] &= \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\Omega_n\left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\end{aligned}$$

Em R podemos realizar esse procedimento por meio do uso do pacote *miceadds*. Este pacote disponibiliza uma função de nome *lm.cluster* que facilita a estimação de uma regressão linear clusterizada. Instale e libere a biblioteca para o uso.

In [1]:
```
install.packages("miceadds")
library(miceadds)
```

```
Installing package into '/home/helson/R/x86_64-pc-linux-gnu-library/3.
6'
(as 'lib' is unspecified)

also installing the dependency 'mice'


Loading required package: mice


Attaching package: 'mice'


The following object is masked from 'package:stats':

    filter


The following objects are masked from 'package:base':

    cbind, rbind


* miceadds 3.10-28 (2020-07-29 21:56:24)
```

In [3]:
```
# Vamos usar a base de dados mtcars do R para exemplificar.
# Vamos fazer uma regressão linear onde queremos obter a relação
# entre o peso do carro (wt) e o consumo de combustível
# dado em milhas por galão de combustível (mpg)
# a regressão será clusterizada pelo número de cilindros (cyl).
summary(lm.cluster(mpg ~ wt, data = mtcars, cluster = mtcars$cyl))
```

```
Loading required namespace: sandwich


R^2= 0.75283

             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 37.285126  3.9219084  9.506883 1.964601e-21
wt          -5.344472  0.9603426 -5.565172 2.618941e-08
```

1.3 Modelo de Probabilidade Linear

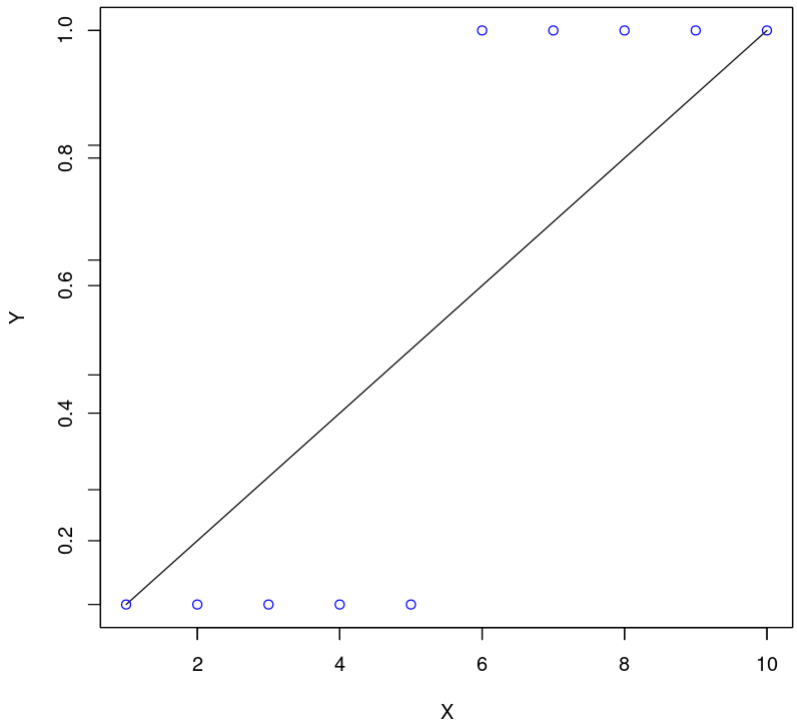
O modelo de probabilidade linear é uma especificação utilizada para quando temos uma variável dependente binária. Nesse caso, $\widehat{Y}$ mostra a probabilidade de sucesso dado um conjunto de covariadas X. Nesse caso, a especificação do modelo segue-se como:

$$P(Y = 1|X) = \alpha + \beta X + e$$

Com $Y \in \{0, 1\}$. O MPL retorna uma média das probabilidades individuais podendo ser representado como:

In [30]:
```
plot(x= 1:10, y = seq(.1,1,.1), type = "line", ylab = "Y", xlab = "X")
par(new=T)
plot(x=1:10, y=c(rep(0,5), rep(1,5)), type = "p", col = "blue",  xlab =
```

```
Warning message in plot.xy(xy, type, ...):
“gráfico do tipo 'line' vai ser truncado para o primeiro caractere”
Warning message in plot.window(...):
“"labels" não é um parâmetro gráfico”
Warning message in plot.xy(xy, type, ...):
“"labels" não é um parâmetro gráfico”
Warning message in box(...):
“"labels" não é um parâmetro gráfico”
Warning message in title(...):
“"labels" não é um parâmetro gráfico”
```



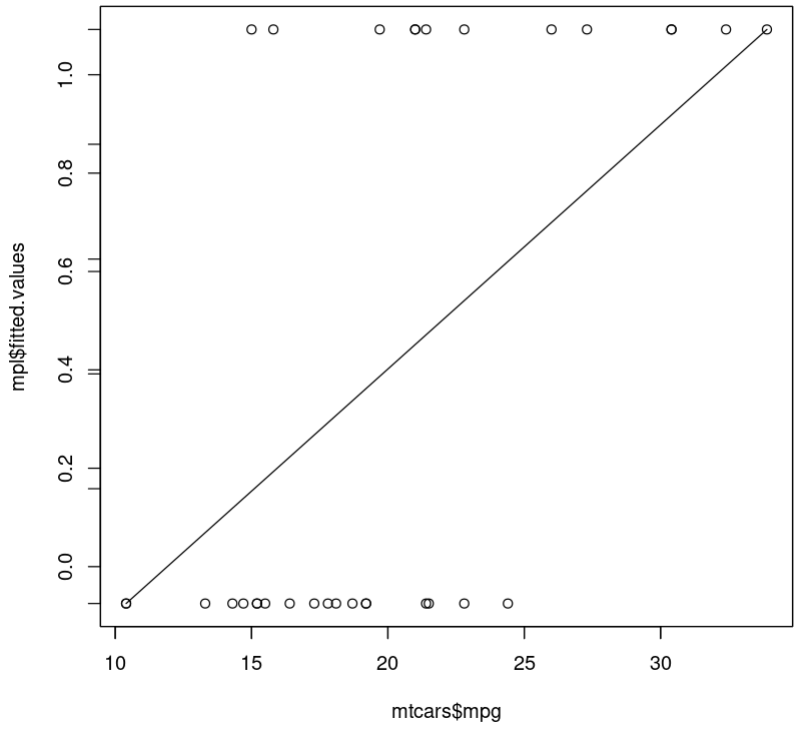
A implementação do mpl em R é semelhante aos comandos usados para estimar o modelo linear geral, com a diferença de que agora Y é uma variável binária. Para exemplificar, vamos usar as informações que o R disponibiliza sobre carros para estimar a probabilidade de o indivíduo comprar um carro com transmissão automática dado o seu desempenho com combustível.

In [32]:
```
mpl  <- lm(data = mtcars, am ~ mpg)
summary(mpl)
```

```
Call:
lm(formula = am ~ mpg, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.6203 -0.2962 -0.1054  0.2519  0.8466 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.59149    0.25336  -2.335 0.026448 *  
mpg          0.04966    0.01209   4.106 0.000285 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4059 on 30 degrees of freedom
Multiple R-squared:  0.3598,    Adjusted R-squared:  0.3385 
F-statistic: 16.86 on 1 and 30 DF,  p-value: 0.000285
```

In [43]:
```
plot( y = mpl$fitted.values, x = mtcars$mpg, type = "l")
par(new = T)
plot( y = mtcars$am, x = mtcars$mpg, type = "p",  xlab = "", ylab = "",
```

```
Warning message in plot.window(...):
“"labels" não é um parâmetro gráfico”
Warning message in plot.xy(xy, type, ...):
“"labels" não é um parâmetro gráfico”
Warning message in box(...):
“"labels" não é um parâmetro gráfico”
Warning message in title(...):
“"labels" não é um parâmetro gráfico”
```



1.3.1 Problemas com o MPL

- Não se pode garantir a homocedasticidade dos resíduos.
- $\widehat{Y}$ pode não estar contido no intervalo {0,1}.

2 O modelo Logit

O modelo logit possui a mesma funcionalidade do MPL, no entanto, este não se encaixa na classe de modelos lineares. A lógica desse modelo consiste em estimar um vetor de parâmetros β usando uma função do tipo logística. Assim, o preditor deste modelo conseguiria retornar valores mais precisos para a probabilidade de sucesso.

O modelo logit pode ser representado como:

$$p = \Lambda\left(x'\beta\right) = \frac{e^{x\beta}}{1 + e^{x'\beta}}$$

Em que p é a probabilidade de sucesso, X é uma matriz de regressores, β é o vetor de parâmetros a ser estimado e $\Lambda(.)$ é uma função de distribuição acumulada (FDA) do tipo logística, com $\Lambda(z) = \frac{e^z}{(1+e^z)} = \frac{1}{(1+e^{-z})}$.

Para especificar, considere um modelo da seguinte estrutura:

$$Y = X\beta + \varepsilon \qquad \text{com} \qquad Y \in \{0, 1\}$$

A FDA logística será:

$$P(y = 1 \mid x) = \frac{e^{x'\beta}}{1 + e^{x'\beta}}$$

A função de Log-verossimilhança será:

$$\ln L(\beta) = \sum \left[y \ln\left(\frac{\log\left(x'\beta\right)}{1 + \exp(x'\beta)} \right) + (1 - y) \ln\left(\frac{1}{1 + \exp(x'\beta)} \right) \right] = 0$$

A CPO será:

$$\begin{aligned} \frac{\partial lnL}{\partial \beta} &= \left[\sum \frac{xy}{1 + exp\,(x'\beta)} - \sum \frac{(1 - y)x}{1 + \exp(-x'\beta)} \right] = 0 \\ &= \sum_{i=1}^{N} \left[y_i\, F\left(-x'\beta\right) - (1 - y_i)\, F\left(x_i'\beta\right) \right] x_i = 0 \end{aligned}$$

Se p é a probabilidade de sucesso, então a probabilidade de fracasso é 1- p. Usando esse conceito na equação de p podemos calcular uma razão de chances dada por $\frac{p}{1-p}$. Fazendo isso chegamos a:

$$\frac{p}{1 - p} = exp(x'\beta)$$

Já o efeito marginal é calculado como:

$$\begin{aligned} \text{emg} &= \frac{\partial P}{\partial x_i} \\ \text{como } P &= \frac{\exp\left(x'\beta\right)}{1+\exp(x'\beta)} \\ emg &= \frac{\beta exp\left(x'\beta\right)\cdot\left[1+\exp\left(x'\beta\right)\right] - \beta \exp\left(x'\beta\right)\cdot\left[exp\left(y'\beta\right)\right]}{\left[1+\exp(x'\beta)\right]^2} \\ \text{emg} &= \frac{\beta \exp\left(x'\beta\right)}{[1+exp(x'\beta)]^2} \\ emg &= p(1 - p)\beta \end{aligned}$$

Em R, o pacote *aod* faz todo esse processo, instale-o e enseguida libere a biblioteca.

In [2]:
```
install.packages("aod")
library(aod)
```

Vamos usar o mesmo exemplo do mpl, só que agora, estimaremos um modelo logit.

In [2]:
```
modlogit <- glm(am ~ mpg, data=mtcars, family=binomial(link = "logit"))
summary(modlogit)
```

```
Call:
glm(formula = am ~ mpg, family = binomial(link = "logit"), data = mtc
ars)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5701  -0.7531  -0.4245   0.5866   2.0617

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -6.6035     2.3514  -2.808  0.00498 **
mpg           0.3070     0.1148   2.673  0.00751 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.230  on 31  degrees of freedom
Residual deviance: 29.675  on 30  degrees of freedom
```

2.1 Razão de chances

Podemos calcular a razão de chances de cada variável fazendo o exponencial dos coeficientes do modelo estimado.

In [9]:
```
exp(coef(modlogit)[2])
```

mpg: 1.35937928836939

2.2 Efeito marginal

Para calcular os efeitos marginais precisamos usar uma biblioteca auxiliar chamada *mfx*. Instale a biblioteca e libere-a para o uso.

In [3]:
```
install.packages("mfx")
library(mfx)
```

```
Loading required package: sandwich

Loading required package: lmtest

Loading required package: zoo


Attaching package: ‘zoo’


The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric


Loading required package: MASS

Loading required package: betareg
```

Use a função *logitmfx* para calcular os efeitos marginais.

In [11]:
```
logitmfx(formula = am ~ mpg, data = mtcars)
```

```
Call:
logitmfx(formula = am ~ mpg, data = mtcars)

Marginal Effects:
       dF/dx Std. Err.      z    P>|z|
mpg 0.073235  0.028307 2.5872 0.009675 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Probabilidades preditas

Nesse caso, precisamos usar a função *predict*. Esta função assume a seguinte estrutura.

predict(object, newdata, type)

Em que *object* é o objeto que contém os coeficientes do modelo logit, *newdata* é um banco de dados que descreve as condições para o cálculo da probabilidade e *type* é o tipo de predição

In [16]:
```
df <- data.frame(mpg = mean(mtcars$mpg))
predict(modlogit, newdata=df, type="response")
```

1: 0.39289997890346

3 O modelo Probit

O modelo probit tem a mesma lógica do modelo logit, a diferença é que em vez de usar uma FDA logística, no modelo probit utiliza-se uma FDA normal. O modelo probit especifica a seguinte probabilidade condicional:

$$p = \Phi\left(\mathbf{x}'\boldsymbol{\beta}\right) = \int_{-\infty}^{\mathbf{x}'\beta} \phi(z)dz$$

Em que $\Phi(z)$ é uma FDA normal com derivada $\phi(z) = (1/\sqrt{2\pi})\exp\left(-z^2/2\right)$. Inserindo a equação de probabilidade na função de verossimilhança, aplicando o logaríimo natural e derivando em relação a β, tem-se:

$$\sum_{i=1}^{N} w_i \left(y_i - \Phi\left(\mathbf{x}_i'\boldsymbol{\beta}\right)\right)\mathbf{x}_i = \mathbf{0}$$

Em que $w_i = \phi\left(\mathbf{x}_i'\boldsymbol{\beta}\right) / \left[\Phi\left(\mathbf{x}_i'\boldsymbol{\beta}\right)\left(1 - \Phi\left(\mathbf{x}_i'\boldsymbol{\beta}\right)\right)\right]$ é um peso que varia de acordo com as observações.

Para encontrar o efeito marginal de um modelo probit, basta derivar a equação de probabilidade condicional em relação a β, o que resulta em:

$$\begin{aligned}\partial p_i/\partial x_{ij} &= \phi\left(\mathbf{x}_i'\boldsymbol{\beta}\right)\beta_j \\ &= \phi\left(\Phi^{-1}\left(p_i\right)\right)\beta_j\end{aligned}$$

Em R, a aplicação do probit também é bem semelhante àquela utilizada no modelo logit. É necessário apenas indicar o tipo de distribuição utilizado na função de verossimilhança.

In [1]:
```
modprobit <- glm(am ~ mpg, data=mtcars, family=binomial(link = "probit"))
summary(modprobit)
```

```
Call:
glm(formula = am ~ mpg, family = binomial(link = "probit"), data = mtc
ars)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5613  -0.7711  -0.4134   0.5858   2.0602

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.91021    1.24654  -3.137  0.00171 **
mpg          0.18227    0.06114   2.981  0.00287 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 43.230  on 31  degrees of freedom
Residual deviance: 29.559  on 30  degrees of freedom
AIC: 33.559

Number of Fisher Scoring iterations: 6
```

Para calcular os efeitos marginais, utilizamos a função *proitmfx* do pacote *mfx* de acordo com o especificado abaixo.

In [5]:
```
probitmfx(formula = am ~ mpg, data = mtcars)
```

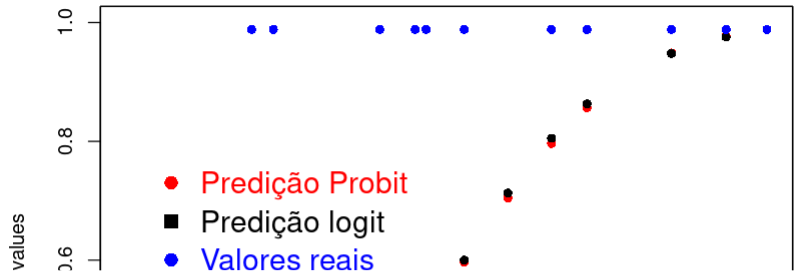
```
Call:
probitmfx(formula = am ~ mpg, data = mtcars)

Marginal Effects:
       dF/dx Std. Err.      z    P>|z|
mpg 0.070507  0.024274 2.9047 0.003677 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Qual modelo usar, logit ou probit

Ambos os modelos são bastante semelhantes e retornarão coeficientes praticamente idênticos. A diferença principal entre os dois modelos é que o probit possui uma curva de predição mais acentuada.

In [10]:
```
plot( y = modprobit$fitted.values, x = mtcars$mpg,
     type = "p", col = "red", pch = 16)
par(new = T)
plot( y = modlogit$fitted.values, x = mtcars$mpg,
     type = "p", xlab = "", ylab = "",
     yaxt = "n", col = "black", pch = 16)
par(new = T)
plot( y = mtcars$am, x = mtcars$mpg, type = "p",
     xlab = "", ylab = "",
     yaxt = "n", labels = F, col = "blue", pch = 16)
legend(11,.8,
        cex = 1.5,
        bty = "n",
        legend = c("Predição Probit", "Predição logit",
                   "Valores reais"),
        text.col = c("red", "black", "blue"),
        col = c("red", "black", "blue"),
        pch = c(16,15))
```

```
Warning message in plot.window(...):
“"labels" não é um parâmetro gráfico”
Warning message in plot.xy(xy, type, ...):
“"labels" não é um parâmetro gráfico”
Warning message in box(...):
“"labels" não é um parâmetro gráfico”
Warning message in title(...):
“"labels" não é um parâmetro gráfico”
```



4 O modelo Tobit

O modelo tobit parte do princípio que uma determinada variável latente y linear nos seus regressores, com erros homocedásticos normalmente distribuídos com média zero e variância constante, ou seja:

$$y = x'\beta + \varepsilon \qquad com \qquad \varepsilon \sim \mathcal{N}\left[0, \sigma^2\right]$$

De maneira que y pode ser censurada ou truncada em um determinado limiar, L. Para demonstrar, vamos supor o caso de uma cesura para baixo, de modo que:

$$y = \begin{cases} y^* & \text{if } y^* > 0 \\ - & \text{if } y^* \leq 0 \end{cases}$$

Nesse caso, existirá uma função de densidade censurada, $f^*(y)$, e uma FDA normal padrão no limiar inferior, tal que:

$$\begin{aligned} F^*(0) &= \Pr\left[y^* \leq 0\right] \\ &= \Pr\left[\mathbf{x}'\boldsymbol{\beta} + \varepsilon \leq 0\right] \\ &= \Phi\left(-\mathbf{x}'\boldsymbol{\beta}/\sigma\right) \\ &= 1 - \Phi\left(\mathbf{x}'\boldsymbol{\beta}/\sigma\right) \end{aligned}$$

Em que Φ é uma FDA normal padrão. Além disso, o processo de estimação é baseado em uma função de densidade censurada dada por:

$$f(y) = \left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{-\frac{1}{2\sigma^2}\left(y - \mathbf{x}'\boldsymbol{\beta}\right)^2\right\}\right]^d \left[1 - \Phi\left(\frac{\mathbf{x}'\boldsymbol{\beta}}{\sigma}\right)\right]^{1-d}$$

Em que:

$$d = \begin{cases} 1 & \text{if } y > L \\ 0 & \text{if } y = L \end{cases}$$

A função de log-verossimilhança desse processo é:

$$\ln \mathrm{L}_N\left(\boldsymbol{\beta}, \sigma^2\right) = \sum_{i=1}^{N}\left\{d_i\left(-\frac{1}{2}\ln 2\pi - \frac{1}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\left(y_i - \mathbf{x}_i'\boldsymbol{\beta}\right)^2\right) + (1-d_i)\ln\left(1 - \Phi\right.\right.$$

E as CPO's são:

$$\frac{\partial \ln \mathrm{L}_N}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{N}\frac{1}{\sigma^2}\left(d_i\left(y_i - \mathbf{x}_i'\boldsymbol{\beta}\right) - (1-d_i)\frac{\sigma\phi_i}{(1-\Phi_i)}\right)\mathbf{x}_i = \mathbf{0}$$

$$\frac{\partial \ln \mathrm{L}_N}{\partial \sigma^2} = \sum_{i=1}^{N}\left\{d_i\left(-\frac{1}{2\sigma^2} + \frac{\left(y_i - \mathbf{x}_i'\boldsymbol{\beta}\right)^2}{2\sigma^4}\right) + (1-d_i)\frac{\phi_i \mathbf{x}_i'\boldsymbol{\beta}}{(1-\Phi_i)}\frac{1}{2\sigma^3}\right\} = 0$$

Caso a variável latente seja truncada para baixo, a função de log-verossimilhança é:

$$\ln \mathrm{L}_N\left(\boldsymbol{\beta}, \sigma^2\right) = \sum_{i=1}^{N}\left\{-\frac{1}{2}\ln\sigma^2 - \frac{1}{2}\ln 2\pi - \frac{1}{2\sigma^2}\left(y_i - \mathbf{x}_i'\boldsymbol{\beta}\right)^2 - \ln\Phi\left(\mathbf{x}_i'\boldsymbol{\beta}/\sigma\right)\right\}$$

Os efeitos marginais do modelo tobit são:

$$\begin{aligned} &Variável Latente \Rightarrow \partial \mathrm{E}\left[y^* \mid \mathbf{x}\right]/\partial \mathbf{x} = \boldsymbol{\beta} \\ &Truncada \Rightarrow \partial \mathrm{E}[y, y > 0 \mid \mathbf{x}]/\partial \mathbf{x} = \left\{1 - w\lambda(w) - \lambda(w)^2\right\}\boldsymbol{\beta} \\ &Censurada \Rightarrow \partial \mathrm{E}[y \mid \mathbf{x}]/\partial \mathbf{x} = \Phi(w)\boldsymbol{\beta} \end{aligned}$$

Em que $w = x'\beta/\sigma$

Para implementar o modelo tobit em R precisamos usar o pacote *AER*. Instale-o e em seguida libere a biblioteca para o uso.

In [25]:
```
install.packages("AER")
library(AER)
```

```
Loading required package: car

Loading required package: carData

Loading required package: survival
```

Vamos usar a base de dados sobre matrimônio disponibilizada por Greene (2003). Nesse caso, iremos verificar o efeito do matrimônio dobre o nível de felicidade dos indivíduos. A variável que representa o matrimônio (affairs) é uma variável binária com valor 1 se o indivíduo é casado e zero caso contrário. Já o nível de felicidade (rating) é uma variável truncada entre 1 e 5, de maneira que:

1 ⇒ Muito infeliz.

2 ⇒ Um pouco infeliz.

3 ⇒ Mais ou menos feliz.

4 ⇒ Feliz.

5 ⇒ Muito feliz.

In [27]:
```
data("Affairs")
modtobit <- tobit(rating ~ affairs + age + yearsmarried +
                  religiousness + occupation, left = 1,
                  right = 5, data = Affairs)
summary(modtobit)
```

```
Call:
tobit(formula = rating ~ affairs + age + yearsmarried + religiousness
+
    occupation, left = 1, right = 5, data = Affairs)

Observations:
         Total  Left-censored     Uncensored Right-censored
           601             16            353            232

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    5.06531    0.35737  14.174  < 2e-16 ***
affairs       -0.11187    0.02146  -5.212 1.87e-07 ***
age           -0.01388    0.01217  -1.140  0.25410
yearsmarried  -0.05677    0.02069  -2.744  0.00606 **
religiousness  0.06685    0.06288   1.063  0.28770
occupation     0.03868    0.03926   0.985  0.32449
Log(scale)     0.47252    0.04189  11.280  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Scale: 1.604

Gaussian distribution
Number of Newton-Raphson Iterations: 3
Log-likelihood: -873.9 on 7 Df
Wald-statistic:  73.7 on 5 Df, p-value: 1.7366e-14
```

5 Dados em Painel

De uma maneira geral, um painel de dados é um conjunto de dados de corte transversal empilhados de acordo com um determinado intervalo de tempo.

Considere $i = 1, 2, \dots N$ como sendo os indivíduos de um determinado banco de dados, de maneira que, este banco de dados dispõe de informações sobre esses indivíduos nos períodos de tempo $t = 1, 2, \dots, T$. Deixe $Y = (Y_1, Y_2, \dots Y_N)'$ denotar o vetor $nx1$ de informações sobre a variável dependente Y_i. Além disso, considere X como sendo uma matriz (n*t x k) de variáveis independentes.

5.1 O modelo pooled

Considere a seguinte equação para a especificação do modelo pooled:

$$Y_{i,t} = \beta X_{i,t} + e_{i,t}$$

Em que β é um vetor k x 1 de parâmetros desconhecidos e e é o termo de erro.

Em linhas gerais, o modelo pooled consiste em uma estimação convencional de MQO para os dados empilhados, dada a suposição de que a média dos resíduos é estritamente independente dos regressores, ou seja:

$$E[e_{it}|X_{i,t}] = 0$$

De uma maneira geral, a hipótese de independência estrita requer que não haja correlação entre os termos de erro e os valores presentes, passados ou futuros de X_i. Em outras palavras, a estimação de um modelo pooled só é aplicável caso X_i seja exógeno.

Para encontrar os valores do vetor β, considere o seguinte procedimento:

$$\begin{aligned}
e_{i,t} &= Y_{i,t} - \hat{\beta}X_{i,t} \\
e'_{i,t}e_{i,t} &= \left(Y_{i,t} - \hat{\beta}X_{i,t}\right)' \left(Y_{i,t} - \hat{\beta}X_{i,t}\right) \\
e_{i,t}e'_{i,t} &= Y'_{it}Y_{it} - \hat{\beta}X'_{it}Y_{it} - Y'_{it}X_{it}\hat{\beta} + \hat{\beta}X'_{it}X_{it}\hat{\beta} \\
e_{i,t}e'_{i,t} &= Y'_{it}Y_{it} - 2\hat{\beta}X'_{it}Y_{it} + \hat{\beta}' X'_{it}X_{it}\beta
\end{aligned}$$

Derivando em relação a β, tem-se:

$$\begin{aligned}
\frac{\partial e_{i,t}e'_{i,t}}{\partial \hat{\beta}} &= -2X'_{it}Y_{it} + 2X'_{it}X_{it}\hat{\beta} = 0 \\
X'_{it}Y_{it} &= X'_{it}X_{it}\hat{\beta} \\
\hat{\beta}_{pool} &= \left(X'_{it}X_{it}\right)^{-1}\left(X'_{it}Y_{it}\right)
\end{aligned}$$

5.2 O modelo de efeitos aleatórios

Para descrever o modelo de efeitos aleatórios, considere inicialmente que o termo de erro da regressão, $e_{i,t}$ segue a seguinte estrutura:

$$e_{i,t} = u_i + \varepsilon_{i,t}$$

Em que u_i é um efeito individual específico e ε é o termo de erro idiossincrático. Considerando este conceito, o modelo de regressão pode ser escrito como:

$$Y_{i,t} = \beta X_{i,t} + u_i + \varepsilon_{i,t}$$

Ou

$$Y_{i,t} = \beta X_{i,t} + 1u_i + \varepsilon_{i,t}$$

No modelo de efeitos aleatórios assume-se que:

$$
\begin{aligned}
\mathbb{E}\left[\varepsilon_{it} \mid \boldsymbol{X}_i\right] &= 0 \\
\mathbb{E}\left[\varepsilon_{it}^2 \mid \boldsymbol{X}_i\right] &= \sigma_\varepsilon^2 \\
\mathbb{E}\left[\varepsilon_{it}\varepsilon_{jt} \mid \boldsymbol{X}_i\right] &= 0 \qquad \forall j \neq t \\
\mathbb{E}\left[u_i \mid \boldsymbol{X}_i\right] &= 0 \\
\mathbb{E}\left[u_i^2 \mid \boldsymbol{X}_i\right] &= \sigma_u^2 \\
\mathbb{E}\left[u_i\varepsilon_{it} \mid \boldsymbol{X}_i\right] &= 0
\end{aligned}
$$

Assumir uma estrutura de efeitos aleatórios implica em considerar que o vetor de erros e_i para o indivíduo i possui a seguinte estrutura de covariância:

$$
\begin{aligned}
\mathbb{E}\left[\boldsymbol{e}_i \mid \boldsymbol{X}_i\right] &= 0 \\
\mathbb{E}\left[\boldsymbol{e}_i\boldsymbol{e}_i' \mid \boldsymbol{X}_i\right] &= \mathbf{1}_i\mathbf{1}_i'\sigma_u^2 + \boldsymbol{I}_i\sigma_\varepsilon^2 \\
&= \begin{pmatrix}
\sigma_u^2 + \sigma_\varepsilon^2 & \sigma_u^2 & \cdots & \sigma_u^2 \\
\sigma_u^2 & \sigma_u^2 + \sigma_\varepsilon^2 & \cdots & \sigma_u^2 \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_u^2 & \sigma_u^2 & \cdots & \sigma_u^2 + \sigma_\varepsilon^2
\end{pmatrix} \\
&= \sigma_\varepsilon^2\Omega_i
\end{aligned}
$$

Em que I_i é uma matriz identidade de dimensã T.

Dadas essas hipóteses, o vetor de parâmetros β do modelo de efeitos aleatórios é obtido considerando uma estimação de minimos quadrados generalizados (ou GLS) acordo como demonstrado a seguir.

5.2.1 O estimador de efeitos aleatórios

Considere Considere $\sigma_\varepsilon^2\Omega_i$ como sendo a matriz de covariância, onde Ω_i é uma matriz não singular simétrica.

Defina Ω_i como sendo igual a $K'K$ em que K é a raíz quadrada de Ω_i.

Defina:

$$
\begin{aligned}
z &= K^{-1}Y \\
B &= k^{-1}X \\
g &= K^{-1}\varepsilon \\
&\text{O que implica em} \\
z &= B\hat{\beta} + g
\end{aligned}
$$

Organizando a equação de z de acordo com um processo de obtenção de um estimador de MQO, tem-se:

$$g'g = (\boldsymbol{z} - \boldsymbol{B}\hat{\boldsymbol{\beta}})'(\boldsymbol{z} - \boldsymbol{B}\hat{\boldsymbol{\beta}})$$

Note que: $z - B\hat{\beta} = K^{-1}Y - K^{-1}X\hat{\beta}$ Então:

$$= \left(\boldsymbol{K}^{-1}\boldsymbol{Y} - \boldsymbol{K}^{-1}\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)' \left(\boldsymbol{K}^{-1}\boldsymbol{Y} - \boldsymbol{K}^{-1}\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)$$

Colocando K^{-1} fora dos parênteses

$$= (\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'[\boldsymbol{K}']^{-1}\boldsymbol{K}^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

$$= (\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})'\Omega^{-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\boldsymbol{\beta}})$$

O que equivale a:

$$g'g = Y'\Omega^{-1}Y - 2\hat{\beta}' X'\Omega^{-1}Y + 2X'\Omega^{-1}X\hat{\beta}$$

Derivando parcialmente em relação a $\hat{\beta}$, obtém-se:

$$\frac{\partial g'g}{\partial \hat{\beta}} = -2X'\Omega^{-1}Y + 2X'\Omega^{-1}X\hat{\beta} = 0$$

$$X'\Omega^{-1}Y = X'\Omega^{-1}X\hat{\beta}$$

$$\hat{\beta} = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}Y$$

Ou seja: **Quando se utiliza o estimador de efeitos aleatórios um estimador de mínimos quadrados generalizados está sendo usado. Este processo de estimação utiliza como base a matriz $\sigma_\varepsilon^2\Omega_i$.**

5.3 O modelo de efeitos fixos

Se a estrutura estocástica de u_i é desconhecida e/ou possivelmente correlacionada com $X_{i,t}$, então o termo u_i é conhecido como efeito fixo.

Nota: A correlação entre u_i e $X_{i,t}$ torna os estimadores dos modelos pooled e de efeitos aleatórios viesados.

Dado que a não exogeneidade de u_i gera estimadores viesados, a alternativa mais comum para contornar este problema é utilizar alguma técnica para eliminar o termo u_i da equação.

5.3.1 A transformação de within

Considere a seguinte equação de regressão:

$$Y_{i,t} = \beta X_{i,t} + u_i + \varepsilon_{i,t}$$

Agora considere esta mesma equação em termos médios: *Note que u_i é constante.

$$\bar{Y} = \beta\bar{X} + u_i + \bar{\varepsilon}$$

Agora considere a diferença entre estas duas equações:

$$Y_{i,t} - \bar{Y} = \beta X_{i,t} - \beta\bar{X} + u_i - u_i + \varepsilon_{i,t} - \bar{\varepsilon}$$

O que implica em:

$$\dot{Y}_{it} = \dot{X}_{it}\beta + \dot{\varepsilon}_{it}$$

Em que $\dot{Y}_{it} = Y_{i,t} - \bar{Y}$, $\dot{X}_{it} = X_{i,t} - \bar{X}$ e $\dot{\varepsilon}_{it} = \varepsilon_{i,t} - \bar{\varepsilon}$.

Note que o termo u_i foi eliminado da equação de regressão, fazendo com que a endogeneidade proveniente da correlação entre u_i e ε_{it} não seja mais um problema na estimativa.

A principal consequência do uso do estimador de within é que ele gera parâmetros não tendenciosos e eficientes. No entanto, uma consequência negativa merece ser destacada. Uma vez que a equação de regressão passa a ser escrita em desvios em relação à média temporal, **todos os regressores invariantes no tempo são eliminados neste processo.**

5.3.2 O estimador de efeitos fixos (ou estimador de within)

Para encontrar o estimador de efeitos fixos, considere o procedimento usual de uma estimação de MQO aplicada à equação escrita pela transformação de within:

$$\begin{aligned}
\dot{Y} &= \dot{X}\hat{\beta} + \dot{\varepsilon} \\
\dot{\varepsilon}'\dot{\varepsilon} &= \left(\dot{Y} - \dot{X}\hat{\beta}\right)' \left(\dot{Y} - X\hat{\beta}\right) \\
\dot{\varepsilon}'\dot{\varepsilon} &= \dot{Y}_i'\dot{Y} - \hat{\beta}'\dot{X}'\dot{Y} - \dot{Y}'\dot{X}\hat{\beta} + \hat{\beta}'\dot{X}'\dot{X}\hat{\beta} \\
\dot{\varepsilon}'\dot{\varepsilon} &= \dot{Y}'\dot{Y} - 2\hat{\beta}'\dot{X}'\dot{Y} + \hat{\beta}'\dot{X}'\dot{X}\hat{\beta}
\end{aligned}$$

Derivando em relação a $\hat{\beta}$, tem-se:

$$\begin{aligned}
\frac{\partial \dot{\varepsilon}'\dot{\varepsilon}}{\partial \hat{\beta}} &= -2\dot{X}'\dot{Y} + 2\dot{X}'\dot{X}\hat{\beta} = 0 \\
\dot{X}'\dot{Y} &= \dot{X}'\dot{X}\hat{\beta} \\
\hat{\beta}_{fe} &= \left(\dot{X}'\dot{X}\right)^{-1}\left(\dot{X}'\dot{Y}\right)
\end{aligned}$$

5.3.3 Estimação do tipo between

O estimador between é semelhante ao estimador de within, diferenido apenas no grupo usado para extrair as médias. Neste estimador os desvios são calculados em relação às médias individuais de cada observação, ou seja:

$$\begin{aligned}
Y^{\star} &= Y_{i,t} - \bar{Y}_i \\
X^{\star} &= X_{i,t} - \bar{X}_i \\
\varepsilon^{\star} &= \varepsilon_{i,t} - \bar{\varepsilon}_i
\end{aligned}$$

A equação de regressão é:

$$Y_{i,t} - \bar{Y} = \hat{\beta}\left(X_{i,t} - \bar{X}_i\right) + \varepsilon_{i,t} - \bar{\varepsilon}_i$$
$$Y^{\star} = \hat{\beta}X^{\star} + \varepsilon^{\star}$$

O erro quadrático é:

$$\varepsilon^{\star\prime}\varepsilon^{\star} = Y^{\star} - \hat{\beta}X^{\star}$$
$$\varepsilon^{\star\prime}\varepsilon^{\star} = Y^{\star\prime}Y^{\star} - 2\hat{\beta}' X^{\star\prime}Y^{\star} + \hat{\beta}\prime X^{\star\prime}X^{\star}\hat{\beta}$$

Derivando em relação a β:

$$\frac{\partial\varepsilon^{\star\prime}\varepsilon^{\star}}{\partial\beta} = -2X^{\star\prime}Y^{\star} + 2X^{\star\prime}X^{\star}\hat{\beta} = 0$$
$$X^{\star\prime}X^{\star}\hat{\beta} = X^{\star\prime}Y^{\star}$$
$$\hat{\beta} = \left(X^{\star\prime}X^{\star}\right)^{-1}\left(X^{\star\prime}Y^{\star}\right)$$

5.3.4 O estimador de primeira diferença

A transformação de within não é a única maneira de eliminar os efeitos individuais específicos. Outra solução utilizada para resolver este problema é usar uma estimação com variáveis em primeira diferença. Para realizar este procedimento, considere que:

$$\Delta Y_i = Y_{i,t} - Y_{i,t-1}$$
$$\Delta X_i = X_{i,t} - X_{i,t-1}$$
$$\Delta\varepsilon_i = \varepsilon_{i,t} - \varepsilon_{i,t-1}$$

Reescrevendo a equação de regressão de acordo com esses conceitos, tem-se:

$$\Delta Y_i = \Delta X\hat{\beta} + \Delta\varepsilon_i$$

Nesse caso, a equação dos erros quadráticos é dada por:

$$\Delta\varepsilon_i'\Delta\varepsilon_i = \Delta Y_i'\Delta Y_i - 2\hat{\beta}'\Delta X'\Delta Y_i + \hat{\beta}'\Delta X'\Delta X\hat{\beta}$$

Derivando em relação a $\hat{\beta}$, obtém-se:

$$\frac{\partial\Delta\varepsilon_i'\Delta\varepsilon_i}{\partial\hat{\beta}} = -2\Delta X'\Delta Y_i + 2\Delta X'\Delta X\hat{\beta}$$
$$\Delta X'\Delta Y_i = \Delta X'\Delta X\hat{\beta}$$
$$\hat{\beta} = \left(\Delta X'\Delta X\right)^{-1}\Delta X'\Delta Y_i$$

5.3.5 Regressão com variáveis dummy

Considere d_i como sendo um vetor de N variáveis dummy onde o i-ésimo elemento indica o i-ésimo indivíduo. Nesse caso, o i-ésimo elemento de d_i é igual a 1 e os demais elementos são iguais a zero. Considere a equação de regressão apenas com os efeitos individuais específicos

e com o termo de erro:

$$Y = Du + \varepsilon$$

Em que $Du = diag\{1T_1, \ldots, 1T_N\}$.

Considere agora uma estimação de u por MQO. Nesse caso,tem-se:

$$\hat{u} = \left(D'D\right)^{-1}\left(D'Y\right)$$

O componente de erro estimado desse processo é:

$$\hat{\varepsilon} = \left(I_N - D\left(D'D\right)\right)^{-1}\left(D'\right) = \dot{Y}$$

A estimação por variáveis dummy consiste em estimar a seguinte equação de regressão por MQO:

$$Y = X\hat{\beta} + D\hat{u} + \hat{\varepsilon}$$

O estimador de variáveis dummy e o estimador de within possuem o mesmo valor, uma vez que os seus termos de erro são idênticos.

Para demonstrar considere uma matriz M, de tal forma que $\boldsymbol{D} = \text{diag}\left\{\mathbf{1}_{T_1}, \ldots, \mathbf{1}_{T_N}\right\}$ e $\boldsymbol{M_D} = \boldsymbol{I}_n - \boldsymbol{D}\left(\boldsymbol{D}'\boldsymbol{D}\right)^{-1}\boldsymbol{D}'$. Sendo que $\boldsymbol{M_D} = \text{diag}\{\boldsymbol{M}_1, \ldots, \boldsymbol{M}_N\}$. Com essa notação, tem-se que:

$$\boldsymbol{M_D}\boldsymbol{Y} = \dot{\boldsymbol{Y}} = \begin{pmatrix} \dot{\boldsymbol{Y}}_1 \\ \vdots \\ \dot{\boldsymbol{Y}}_N \end{pmatrix}, \quad \boldsymbol{M_D}\boldsymbol{X} = \dot{\boldsymbol{X}} = \begin{pmatrix} \dot{\boldsymbol{X}}_1 \\ \vdots \\ \dot{\boldsymbol{X}}_N \end{pmatrix}$$

A equação de regressão, nesse caso, passa a ser:

$$\boldsymbol{M_D}Y = \boldsymbol{M_D}X\hat{\beta} + \boldsymbol{M_D}u + \varepsilon$$

A estimação por MQO segue-se como:

$$\begin{aligned} \widehat{\beta}_{\text{dv}} &= \left(\boldsymbol{X}'\boldsymbol{M_D}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}'\boldsymbol{M_D}\boldsymbol{Y}\right) \\ &= \left(\boldsymbol{X}'\boldsymbol{M_D}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}'\boldsymbol{M_D}(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{D}u + \boldsymbol{\varepsilon})\right) \\ &= \beta + \left(\boldsymbol{X}'\boldsymbol{M_D}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}'\boldsymbol{M_D}\boldsymbol{\varepsilon}\right) \end{aligned}$$

Note que $\left(\boldsymbol{X}'\boldsymbol{M_D}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}'\boldsymbol{M_D}\boldsymbol{\varepsilon}\right) = 0$ O que implica em:

$$\widehat{\beta}_{\text{dv}} = \beta$$

5.4 Dados em painel usando R

No R, o usuário pode estimar as equações anteriormente descritas usando um pacote de nome *plm*. Para utilizar a biblioteca, instale-a e libere-a para uso:

In [1]:
```
install.packages("plm")
library(plm)
```

Vamos usar um painel de dados disponibilizado pela própria biblioteca para fazer a demonstração de uso. Este banco de dados possui informações sobre mão de obra, capital, trabalho e produto de um conjunto de empresas do Reino Unido.

In [2]:
```
data("EmplUK", package="plm")
head(EmplUK)
```

A data.frame: 6 × 7

	firm	year	sector	emp	wage	capital	output
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1	1977	7	5.041	13.1516	0.5894	95.7072
2	1	1978	7	5.600	12.3018	0.6318	97.3569
3	1	1979	7	5.015	12.8395	0.6771	99.6083
4	1	1980	7	4.715	13.8039	0.6171	100.5501
5	1	1981	7	4.093	14.2897	0.5076	99.5581
6	1	1982	7	3.166	14.8681	0.4229	98.6151

Desejamos estimar uma função de produção usando estes dados. Com isso, a nossa equação de regressão é:

$$ln(Y) = \alpha + \hat{\beta}_1 ln(k) + \hat{\beta}_2 ln(L) + \eta u + \varepsilon$$

Em que Y é o produto, L é a mão de obra e k é o capital.

Precisamos então, transformar as variáveis em logarítimo.

In [3]:
```
library(dplyr)
library(tidyr)
EmplUK <- EmplUK %>%
mutate(lny = log(output), lnk = log(capital), lnl = log(emp))
head(EmplUK)
```

```
Attaching package: ‘dplyr’


The following objects are masked from ‘package:plm’:

    between, lag, lead


The following objects are masked from ‘package:stats’:

    filter, lag


The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

A data.frame: 6 × 10

	firm	year	sector	emp	wage	capital	output	lny	lnk	lnl
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1	1977	7	5.041	13.1516	0.5894	95.7072	4.561294	-0.5286502	1.617604
2	1	1978	7	5.600	12.3018	0.6318	97.3569	4.578384	-0.4591824	1.722767
3	1	1979	7	5.015	12.8395	0.6771	99.6083	4.601245	-0.3899363	1.612433
4	1	1980	7	4.715	13.8039	0.6171	100.5501	4.610656	-0.4827242	1.550749
5	1	1981	7	4.093	14.2897	0.5076	99.5581	4.600741	-0.6780615	1.409278
6	1	1982	7	3.166	14.8681	0.4229	98.6151	4.591224	-0.8606196	1.152469

Antes de estimar os parâmetros, primeiro, é preciso informar ao R que o data frame em questão possui dados do tipo empilhados. Para tanto, é preciso usar a função *pdata.frame* indicando a variável de identificação do indivíduo e do tempo.

In [14]:
```
EmplUK  <- pdata.frame(EmplUK, index = c("firm", "year"))
head(EmplUK)
```

A pdata.frame: 6 × 10

	firm	year	sector	emp	wage	capital	output	lny	lnk	lnl
	<fct>	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1-1977	1	1977	7	5.041	13.1516	0.5894	95.7072	4.561294	-0.5286502	1.617604
1-1978	1	1978	7	5.600	12.3018	0.6318	97.3569	4.578384	-0.4591824	1.722767
1-1979	1	1979	7	5.015	12.8395	0.6771	99.6083	4.601245	-0.3899363	1.612433
1-1980	1	1980	7	4.715	13.8039	0.6171	100.5501	4.610656	-0.4827242	1.550749
1-1981	1	1981	7	4.093	14.2897	0.5076	99.5581	4.600741	-0.6780615	1.409278
1-1982	1	1982	7	3.166	14.8681	0.4229	98.6151	4.591224	-0.8606196	1.152469

In [11]:
```
# Estimação do tipo pooled
within  <- plm(lny ~ lnl + lnk, model = "within", effect = "individual",
summary(within)
```

```
Oneway (individual) effect Within Model

Call:
plm(formula = lny ~ lnl + lnk, data = EmplUK, effect = "individual",
    model = "within")

Unbalanced Panel: n = 140, T = 7-9, N = 1031

Residuals:
      Min.    1st Qu.     Median    3rd Qu.       Max.
-0.2543348 -0.0470753  0.0027198  0.0537619  0.2221578

Coefficients:
    Estimate Std. Error t-value  Pr(>|t|)
lnl 0.195481   0.018427 10.6086 < 2.2e-16 ***
lnk 0.047473   0.016602  2.8594  0.004343 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Total Sum of Squares:    7.5308
Residual Sum of Squares: 5.3384
R-Squared:      0.29112
Adj. R-Squared: 0.17869
F-statistic: 182.548 on 2 and 889 DF, p-value: < 2.22e-16
```

In [19]:
```
# Estimação por efeitos fixos (Wihin estimator)
pooled  <- plm(lny ~ lnl + lnk, model = "pooling", data = EmplUK)
summary(pooled)
```

```
Pooling Model

Call:
plm(formula = lny ~ lnl + lnk, data = EmplUK, model = "pooling")

Unbalanced Panel: n = 140, T = 7-9, N = 1031

Residuals:
     Min.   1st Qu.    Median   3rd Qu.      Max.
-0.180162 -0.066317 -0.020487  0.069179  0.226970

Coefficients:
             Estimate Std. Error  t-value Pr(>|t|)
(Intercept)  4.6211557  0.0080291 575.5507  < 2e-16 ***
lnl          0.0131545  0.0052630   2.4994  0.01259 *
lnk         -0.0067216  0.0046629  -1.4415  0.14975
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Total Sum of Squares:    0.0926
```

In [17]:
```
# Estimação por efeitos fixos (Between estimator)
between  <- plm(lny ~ lnl + lnk, model = "between", effect = "individual
summary(between)
```

```
Oneway (individual) effect Between Model

Call:
plm(formula = lny ~ lnl + lnk, data = EmplUK, effect = "individual",
    model = "between")

Unbalanced Panel: n = 140, T = 7-9, N = 1031
Observations used in estimation: 140

Residuals:
      Min.    1st Qu.     Median    3rd Qu.       Max.
-0.0738884 -0.0263931 -0.0028441  0.0154776  0.1031986

Coefficients:
             Estimate Std. Error  t-value Pr(>|t|)
(Intercept)  4.6303509  0.0095931 482.6744   <2e-16 ***
lnl          0.0057768  0.0062814   0.9197   0.3594
lnk         -0.0045246  0.0055735  -0.8118   0.4183
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

In [12]:
```
# Regressão com variáveis dummy
regdv <- lm(lny ~ lnl + lnk +  as.factor(firm), data = EmplUK)
summary(regdv)
```

```
Call:
lm(formula = lny ~ lnl + lnk + as.factor(firm), data = EmplUK)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.25433 -0.04707  0.00272  0.05376  0.22216 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)        4.338888   0.045632  95.084  < 2e-16 ***
lnl                0.195481   0.018427  10.609  < 2e-16 ***
lnk                0.047473   0.016602   2.859 0.004343 ** 
as.factor(firm)2  -0.715160   0.057452 -12.448  < 2e-16 ***
as.factor(firm)3  -0.411911   0.050185  -8.208 7.87e-16 ***
as.factor(firm)4  -0.408275   0.051753  -7.889 8.92e-15 ***
as.factor(firm)5  -0.769653   0.059957 -12.837  < 2e-16 ***
as.factor(firm)6   0.453075   0.046690   9.704  < 2e-16 ***
as.factor(firm)7   0.216908   0.048665   4.457 9.37e-06 ***
as.factor(firm)8   0.181622   0.047571   3.818 0.000144 ***
```

In [13]:
```
# Estimação com efeitos aleatórios
random_ef  <- plm(lny ~ lnl + lnk, model = "random", data = EmplUK)
summary(random_ef)
```

```
Oneway (individual) effect Random Effect Model
   (Swamy-Arora's transformation)

Call:
plm(formula = lny ~ lnl + lnk, data = EmplUK, model = "random")

Unbalanced Panel: n = 140, T = 7-9, N = 1031

Effects:
                    var   std.dev share
idiosyncratic 0.0060049 0.0774915 0.893
individual    0.0007215 0.0268614 0.107
theta:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2630  0.2630  0.2630  0.2725  0.2859  0.3069

Residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.166737 -0.066741 -0.011672 -0.000024  0.069209  0.203924

Coefficients:
              Estimate Std. Error  z-value  Pr(>|z|)
(Intercept)  4.6151549  0.0103916 444.1216 < 2.2e-16 ***
lnl          0.0183949  0.0067765   2.7145  0.006637 **
lnk         -0.0079307  0.0060117  -1.3192  0.187099
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Total Sum of Squares:    14.278
Residual Sum of Squares: 8.2442
R-Squared:      0.42266
Adj. R-Squared: 0.42154
Chisq: 14.9631 on 2 DF, p-value: 0.00056339
```

5.5 Teste de Hausman

O teste de hausma é uma medida estatística para testar a hipótese nula de que a correlação entre os resíduos e os efeitos individuais específicos é igual a zero. Formalmente, o teste de Hausman é escrito como:

$$H = \left(\hat{\beta}_{\text{fe}} - \hat{\beta}_{\text{re}}\right)' \widehat{\text{var}}\left[\hat{\beta}_{\text{fe}} - \hat{\beta}_{\text{re}}\right]^{-1} \left(\hat{\beta}_{\text{fe}} - \hat{\beta}_{\text{re}}\right)$$
$$= \left(\hat{\beta}_{\text{fe}} - \hat{\beta}_{\text{re}}\right)' \left(\widehat{V}_{\text{fe}} - \hat{V}_{\text{re}}\right)^{-1} \left(\hat{\beta}_{\text{fe}} - \hat{\beta}_{\text{re}}\right)$$

Em que fe e re representam os efeitos fixos e aleatórios, respectivamente.

O teste de Hausman possui distribuição χ^2 com k graus de liberdade em que k é o número de variáveis independentes.

- H0 não rejeitada ⇒ A correlação entre os termos de erro e os efeitos individuais específicos é diferente de zero: **O modelo de efeitos fixos é preferível aos modelos pooled e de efeitos aleatórios**
- H0 aceita ⇒ A correlação entre os termos de erro e os efeitos individuais específicos é nula: **A escolha entre os modelos de efeitos fixos e de efeitos aleatórios fica a critério do usuário**

Para calcular o teste de hausman no R, podemos utilizar a função *phtest* do pacote *plm*.

In [16]:
```
phtest(within, random_ef)
```

```
        Hausman Test

data:  lny ~ lnl + lnk
chisq = 350.85, df = 2, p-value < 2.2e-16
alternative hypothesis: one model is inconsistent
```

6 Uma breve discussão sobre a programação linear

O objetivo da programação linear, de uma maneira geral, é resolver problemas de maximização/minimização quando as equações objetivo e as equações das restrições são lineares. Algumas técnicas são utilizadas no âmbito da programação linear para resolver esses problemas, como o **método gráfico** e o **método simplex**.

Para exemplificar considere o problema da dieta ilustrado por Chang (1982, p.548).

- Para manter a sua saúde, uma pessoa necessita preencher certos requisitos mínimos de consumo diário de diversos tipos de nutrientes. Suponhamos, por simplicidade, que apenas três tipos de nutrientes sejam ecessários: Cálcio, proteínas e calorias. Além disso suponha também que a dieta da pessoa em questão consiste em apenas dois alimentos, I e II, cujos preços são demonstrados na tabela abaixo. Sabendo que o preço do alimento I é 0,60 e o que o preço do alimento II é 1,00, qual a combinação de alimentos que satisfaz os requisitos diários e gera o menor custo?

	Alimento I (kg)	Alimento II (kg)	Requisito mínimo diário
Cálcio	10	4	20
Proteína	5	5	20
Calorias	2	6	12

A Equação do problema é:

$$C = 0,6x_1 + x_2$$
Sujeito a:
$$10x_1 + 4x_2 \geq 20$$
$$5x_1 + 5x_2 \geq 20$$
$$2x_1 + 6x_2 \geq 12$$
$$x_1, x_2 \neq 0$$

Pela solução gráfica, teríamos que checar os valores dos interceptos verticais e horizontais de cada equação de restrição supondo a igualdade.

- Restrição 1

(0,5)-(2,0)

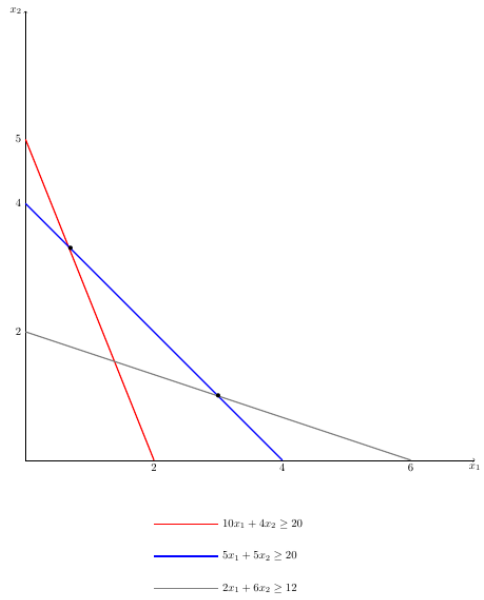
- Restrição 2

(0,4)-(4,0)

- Restrição 3

(0,2)-- (6,0)

In [12]:
```
library("IRdisplay")
display_png(file="solucao_grafica.png", width = 250)
```



Existem duas soluções posíveis:

(1) Restrição 1 = Restrição 2

$$10x_1 + 4x_2 = 20$$

$$5x_1 + 5x_2 = 20$$

Resolvendo simultaneamente estas equações chega-se ao seguinte resultado:

$$\left(\frac{2}{3}; \frac{10}{3}\right)$$

A segunda solução possível é:

(2) Restrição 2 = Restrição 3

$$5x_1 + 5x_2 = 20$$

$$2x_1 + 6x_2 = 12$$

Resolvendo simultaneamente estas equações chega-se ao seguinte resultado:

$$(3; 1)$$

Substituindo esses dois pontos na solução objetivo, tem-se:

$$0,6x_1 + x_2 = 0,6\left(\frac{2}{3}\right) + \frac{10}{3} = 4$$

$$0,6x_1 + x_2 = 0,6 * 3 + 1 = 2,8$$

Como $2,8 \leq 4$, então conclui-se que a dieta ideal é composta por 3 unidades do alimento I e 1 unidade do alimento II.

No R este problema pode ser resolvido facilmente por meio do uso do pacote *linprog*. Para fazer uso da biblioteca, instale-a e libere-a para uso.

In [1]:
```
install.packages("linprog")
library(linprog)
```

```
Loading required package: lpSolve
```

In [29]:
```
# Primeiro é preciso indicar os coeficientes da função objetivo
coef_obj = c(0.6,1)
# Agora vamos indicar os nomes dos objetos
names(coef_obj)  <- c("Alimento I", "Alimento II")
# Em seguida, é preciso indicar os valores de x1 e x2
# em cada restrição
r1 <- c(10,4)
r2 <- c(5,5)
r3 <- c(2,6)
# Vamos deixar os coeficientes das restrições em um único
# objeto de nome coef_r
coef_r  <- rbind(r1,r2,r3)
# Em seguida é preciso indicar o valor dos coeficientes
# independentes em cada restrição
coef_ind <- c(20,20,12)
# Agora vamos dar nomes as restrições
names(coef_ind) <- c("Cálcio", "Proteínas", "Calorias")
# Use o comando solveLP para resolver o problema indicando
# maximum = TRUE caso se trate
# de um problema de maximização e maximum = FALSE caso se
# trate de um problema de minimização
solveLP(coef_obj, coef_ind, coef_r, maximum=TRUE)
```

```
Results of Linear Programming / Linear Optimization

Objective function (Maximum): 101.538

Iterations in phase 1: 0
Iterations in phase 2: 2
Solution
               opt
Alimento I  1.38462
Alimento II 1.53846

Basic Variables
               opt
Alimento I  1.38462
Alimento II 1.53846
S Proteínas 5.38462

Constraints
           actual dir bvec    free    dual dual.reg
Cálcio    20.0000  <=   20 0.00000 3.46154 12.00000
Proteínas 14.6154  <=   20 5.38462 0.00000  5.38462
Calorias  12.0000  <=   12 0.00000 2.69231  8.00000

All Variables (including slack variables)
               opt cvec min.c     max.c     marg marg.reg
Alimento I  1.38462   40    10  75.00000       NA       NA
Alimento II 1.53846   30    16 120.00000       NA       NA
S Cálcio    0.00000    0  -Inf   3.46154 -3.46154       12
S Proteínas 5.38462    0    NA   4.66667  0.00000       NA
S Calorias  0.00000    0  -Inf   2.69231 -2.69231        8
```

6.1 Resolvendo um broblema de minimização

Suponha que queiremos resolver o seguinte problema de minimização:

$$min \quad C = 0,6x1 + x2$$

Sujeito a:

$$10x1 + 4x2 >= 20$$
$$5x1 + 5x2 >= 20$$
$$2x1 + 6x2 >= 12$$
$$x1 >= 0$$
$$x2 >= 0$$

In [2]:
```
# Primeiro vamos indicar quais são os coeficientes da função objetivo
cobj <- c(0.6,1)
# Vamos dar nomes às variáveis que acompanham esses coeficientes
names(cobj) <- c("Bem1","Bem2")

# Vamos indicar o vetor dos coeficientes independentes das restrições
b <- c(20,20,12)
names(b) <- c("calcio","proteína","calorias")

# Coeficientes das equações de restrição
r1 <- c(10,4)
r2 <- c(5,5)
r3 <- c(2,6)

# matriz A com os coeficientes das restrições
A <- rbind(r1,r2,r3)

# Resolvendo o programa
solveLP(cobj,b,A,maximum=FALSE,
        const.dir = c( ">=", ">=", ">=" ))
```

```
Results of Linear Programming / Linear Optimization

Objective function (Minimum): 2.8

Iterations in phase 1: 3
Iterations in phase 2: 1
Solution
     opt
Bem1   3
Bem2   1

Basic Variables
         opt
Bem1       3
Bem2       1
S calcio  14

Constraints
         actual dir bvec free dual dual.reg
calcio       34  >=   20   14 0.00 14.00000
proteína     20  >=   20    0 0.08 10.00000
calorias     12  >=   12    0 0.10  9.33333

All Variables (including slack variables)
           opt cvec      min.c     max.c marg marg.reg
Bem1         3  0.6 -0.8666667 1.0000000   NA       NA
Bem2         1  1.0 -1.4000000 1.8000000   NA       NA
S calcio    14  0.0 -0.0307692 0.0666667 0.00       NA
S proteína   0  0.0 -0.0800000       Inf 0.08 10.00000
S calorias   0  0.0 -0.1000000       Inf 0.10  9.33333
```