Introduction to Generalized Linear Models

with applications in R

Dae-Jin Lee
dlee@bcamath.org

Basque Center for Applied Mathematics

http://idaejin.github.io/bcam-courses/

Outline

Short reminder on linear models

Introduction to Generalized Linear Models

Linear Models

A quick reminder

- We want to explain a variable y (`response`) using some other variables $x_1, x_2, ..., x_p$ (`explanatory`, `independent`, `covariates`).
- **Linear regression** assumes that y_i can be explained by linear combinations of $x_1, ..., x_p$, i.e.:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} + \epsilon_i$$

- In **matrix notation**:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \text{where } \boldsymbol{X} = [1 : x_1 : ... : x_p]$$

where $\boldsymbol{\epsilon}$ represents the error between $\boldsymbol{Y}$ and $\boldsymbol{X}\boldsymbol{\beta}$.

- It is typical to assume $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2)$ and use Maximum Likelihood (ML) to compute the estimator $\boxed{\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}}$[1]

[1] MLE of $\boldsymbol{\beta}$ is equivalent to Least Squares Estimator for Normal/Gaussian Data

Simple linear regression in R

Intro.R

- This script contains R commands to fit a linear model to simulated data

```
> set.seed(1234)
> n <- 50
> x <- seq(1,n)
>  beta0 <- 15
>  beta1 <- 0.5
> sigma <- 3 # standard deviation of the errors
> eps <- rnorm(n,mean=0,sd=3) # generate gaussian random errors
> # Generate random data
>  y <- beta0 + beta1*x  +  eps
```

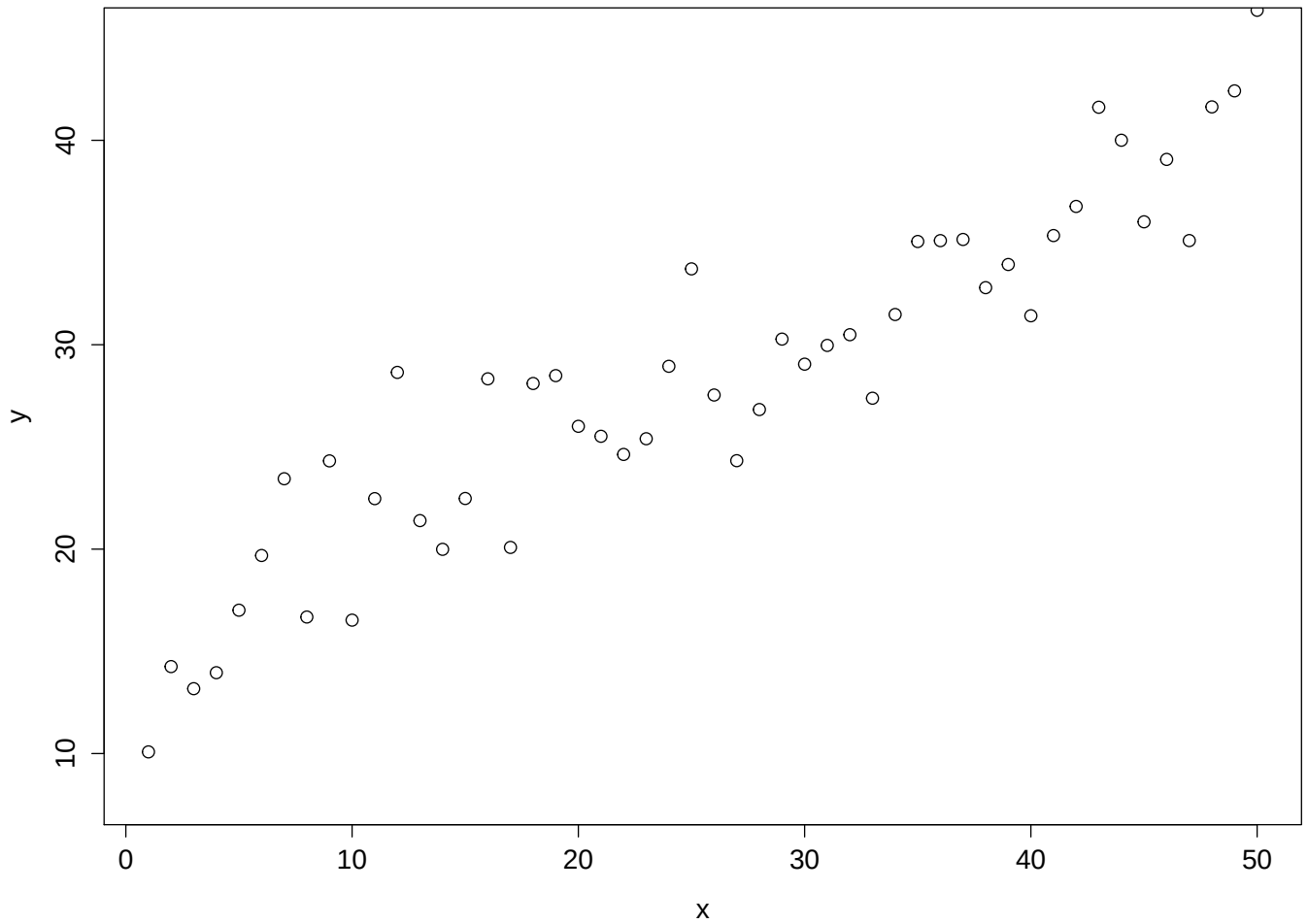
Intro.R (cont.)

- xy-plot

```
> plot(x,y,ylim = c(8,45), cex=1.3, xlab = "x", ylab="y")
> # correlation between x and y
> cor(x,y)
[1] 0.9332733
```

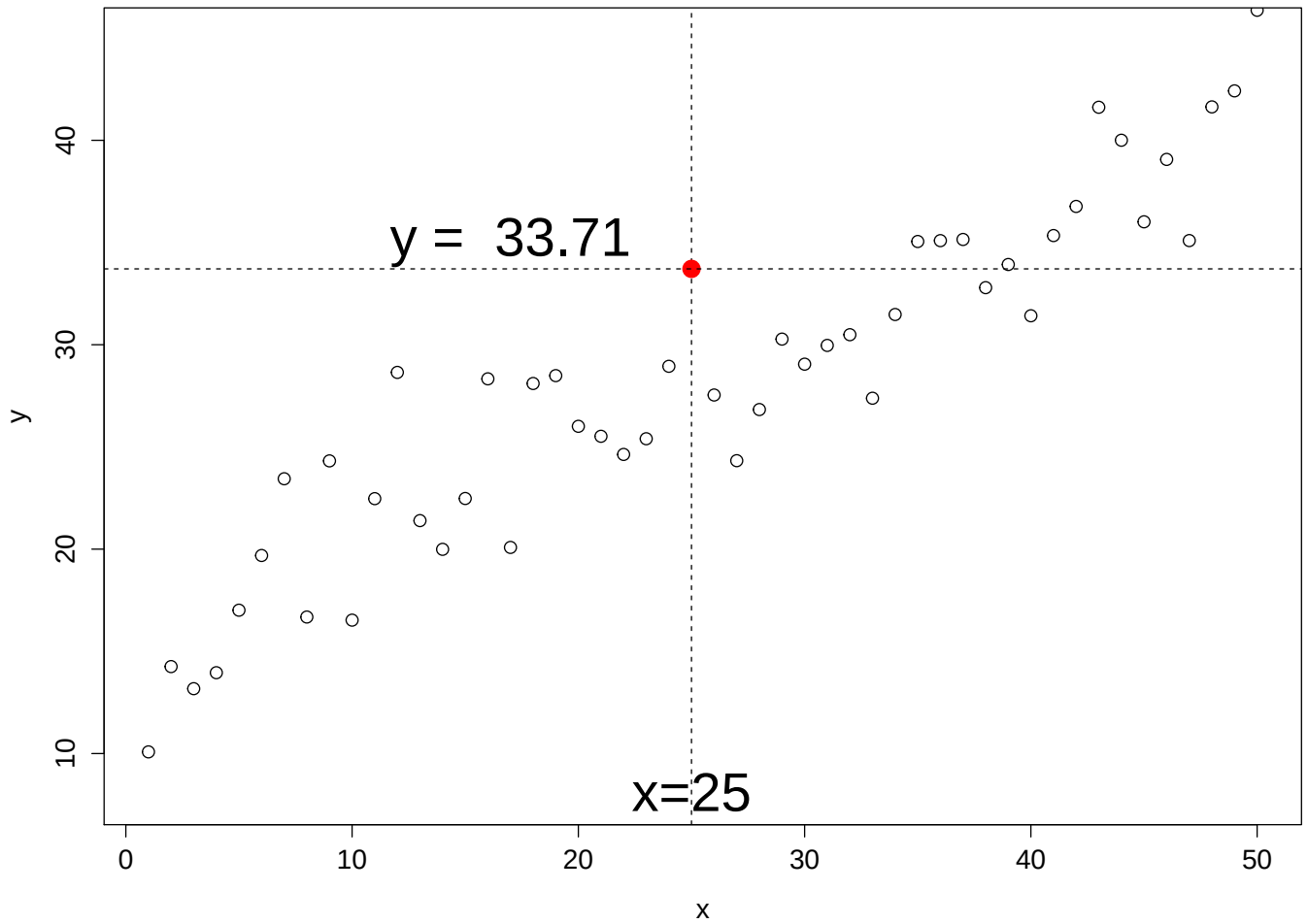
Simple linear regression in R

Linear fit: find a straight line that best approximate the data



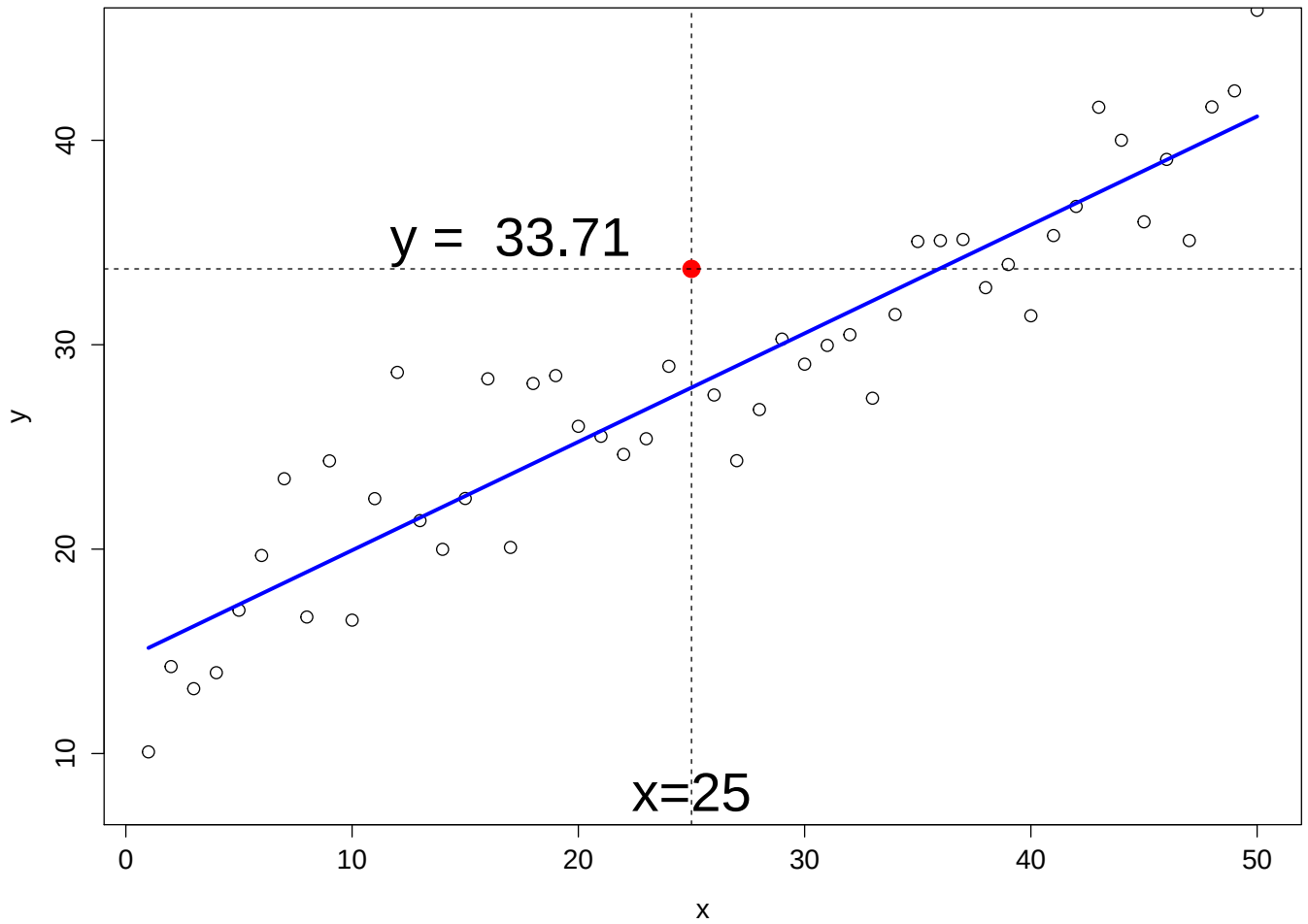
Simple linear regression in R

Linear fit: find a straight line that best approximate the data



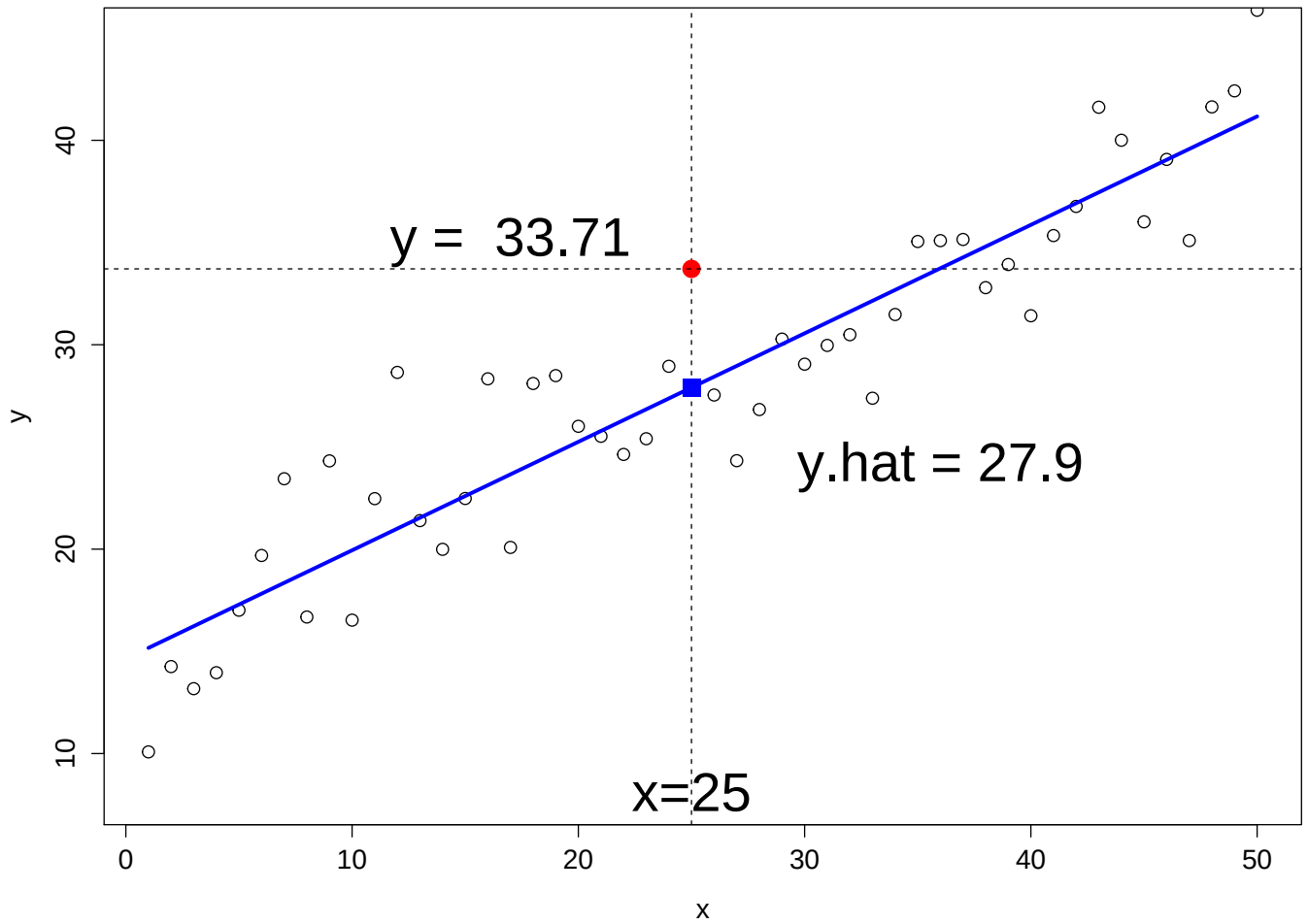
Simple linear regression in R

Linear fit: find a straight line that best approximate the data



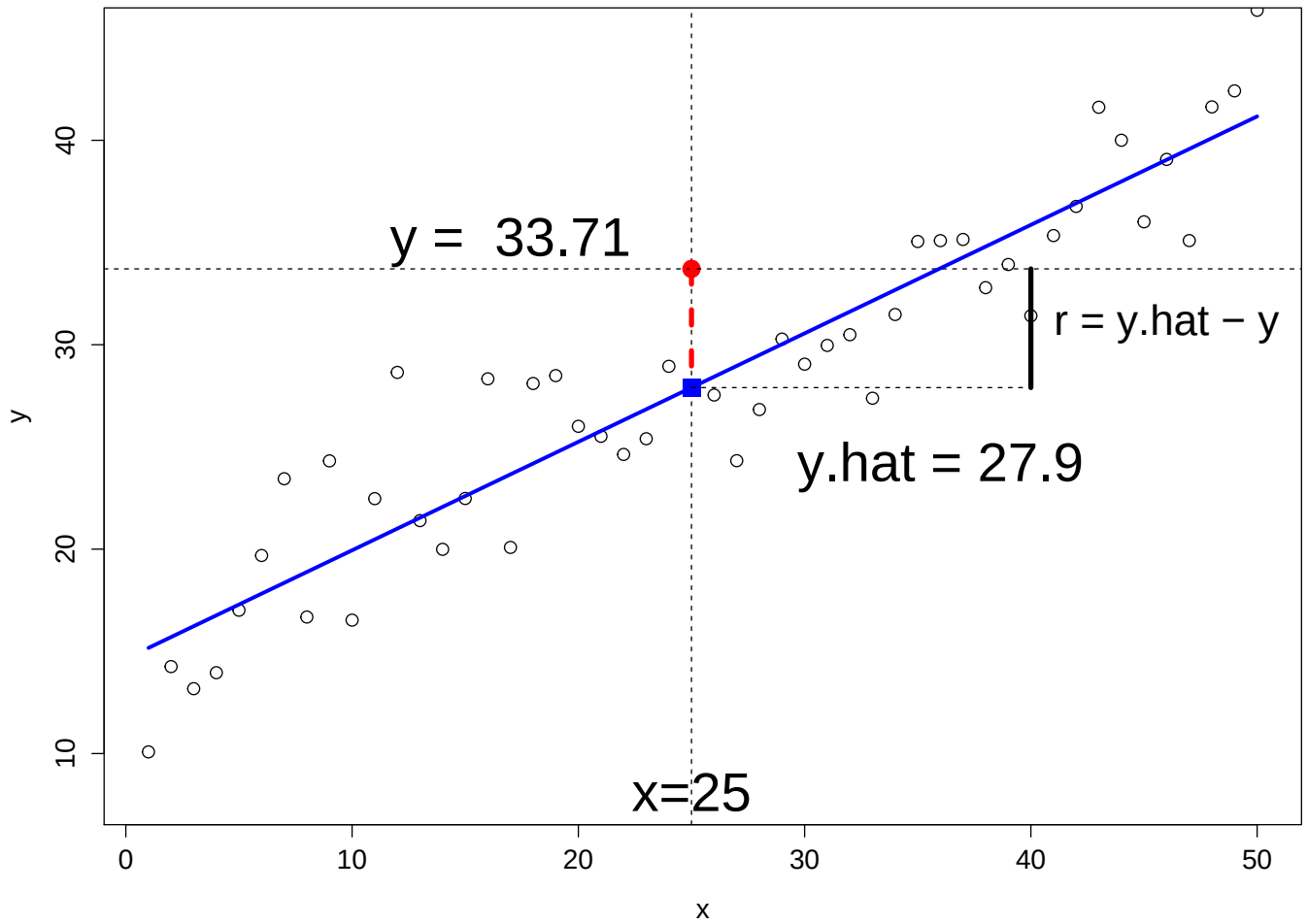
Simple linear regression in R

Linear fit: find a straight line that best approximate the data



Simple linear regression in R

Linear fit: find a straight line that best approximate the data



Some few formulas

- **Ordinary Least Squares**

$$\min_{\beta_0,\beta_1} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- $\hat{\beta}_1 = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} = \frac{\mathsf{Cov}_{x,y}}{\mathsf{Var}_x}$ and $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$
- In matrix form:

$$\boldsymbol{X} = [1 : x]$$
$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$$

where $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}_0, \hat{\boldsymbol{\beta}}_1)$

Intro.R (cont.)

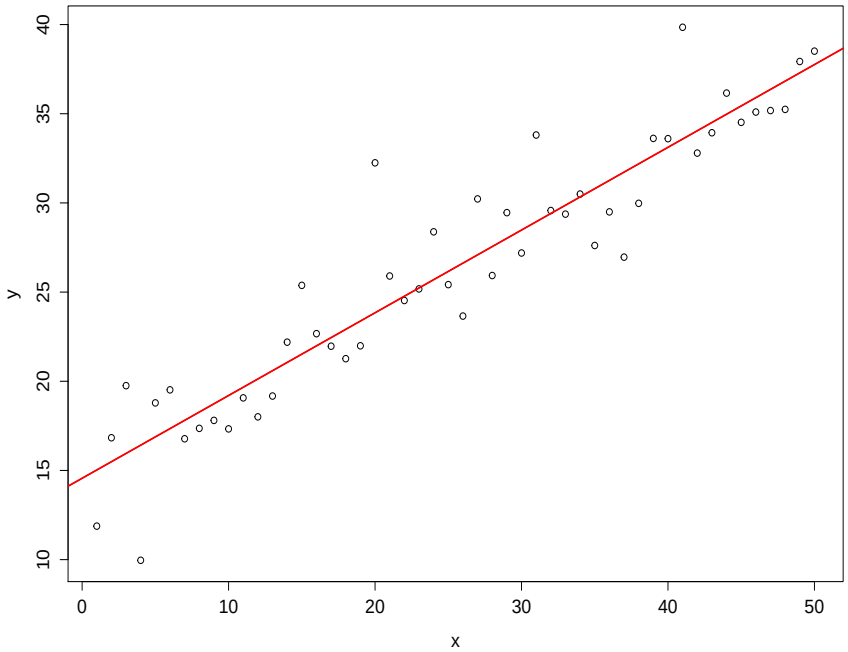
```
> # Using lm()
>
> lin.mod <- lm(y~x)
> lin.mod


Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
    14.5618       0.4639


> coefficients(lin.mod)


(Intercept)           x
 14.5618350   0.4638826
```

Intro.R (cont.)

```
> summary(lin.mod)


Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-6.4545 -1.4126 -0.5366  1.1734  8.4080

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.56184    0.75500   19.29   <2e-16 ***
x            0.46388    0.02577   18.00   <2e-16 ***
---
Signif. codes:  0
```

Intro.R (cont.)

```
> plot(x,y)
> abline(lin.mod,lwd=2,col="red")
```



- **How can you interpret β_0 and β_1?**

Some useful commands for `lm` objects

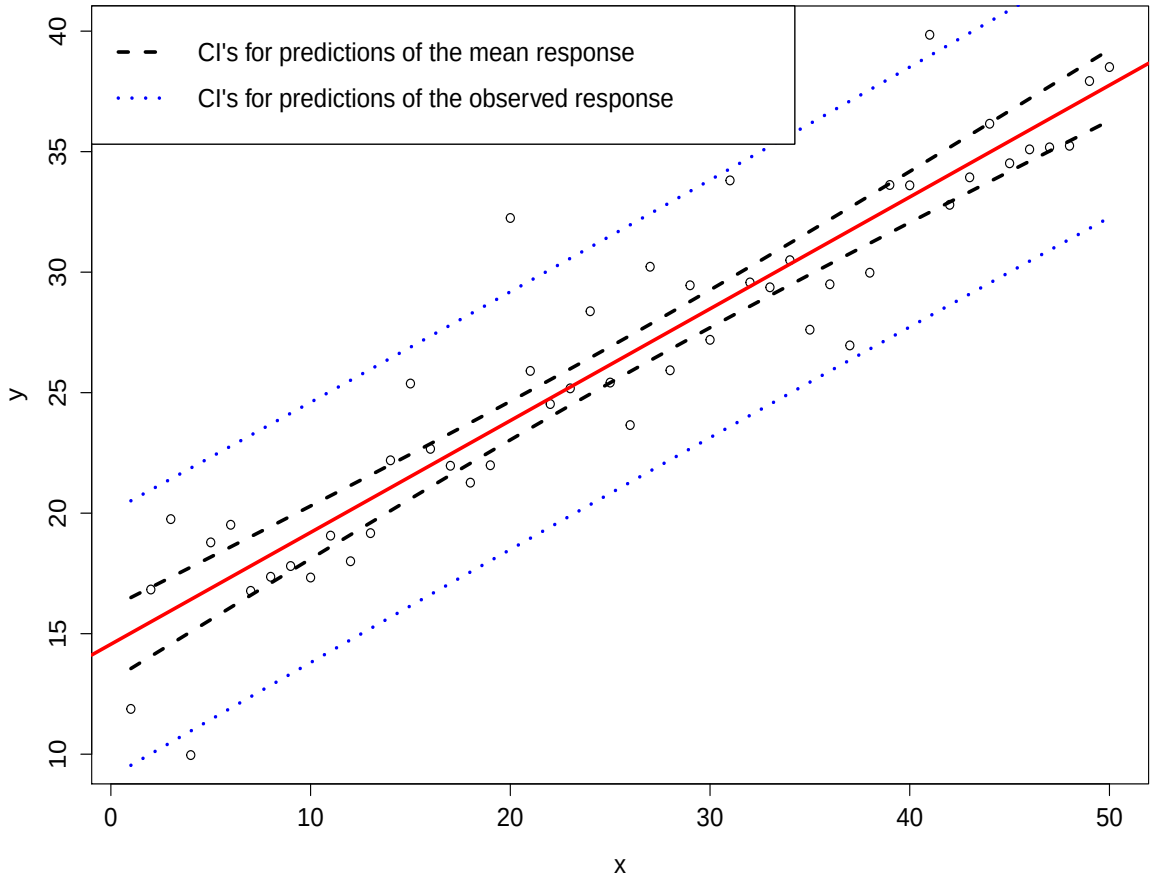
See Faraway's (2002) book (Chapters 1-7)

R commands	
`print()`	Short summary
`summary()`	Summary table
`coef()`	Estimated coefficients
`predict()`	Predict new values
`confint()`	Confident intervals of the estimated parameters
`fitted.values()`	Fitted values of the model
`residuals()`	Residuals of the fitted model
`deviance()`	Deviance
`logLik()`	Logarithm of the likelihood and degrees of freedom (df)

More options

See the `Intro.R` script for the `R` code.

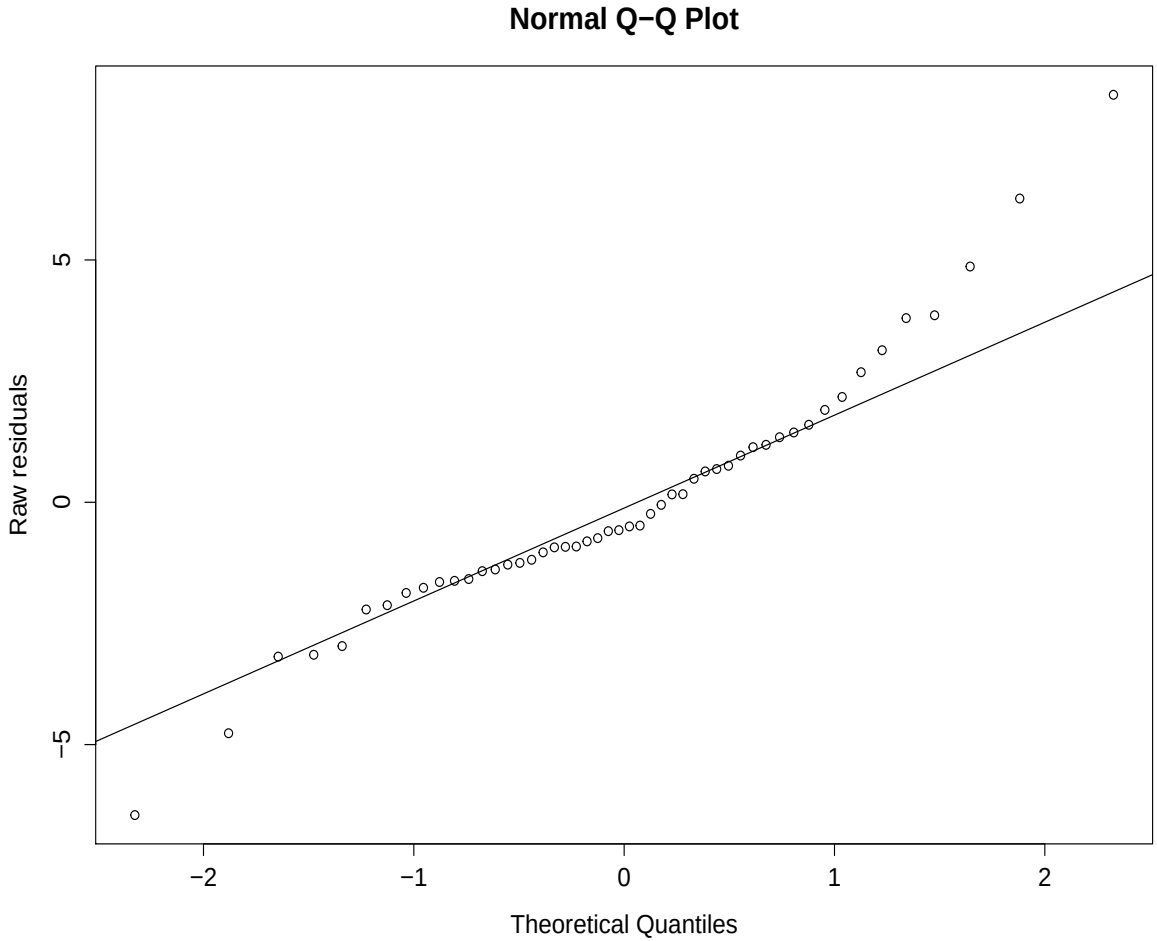
E.g: CI's for predictions



More options

See the `Intro.R` script for the `R` code.

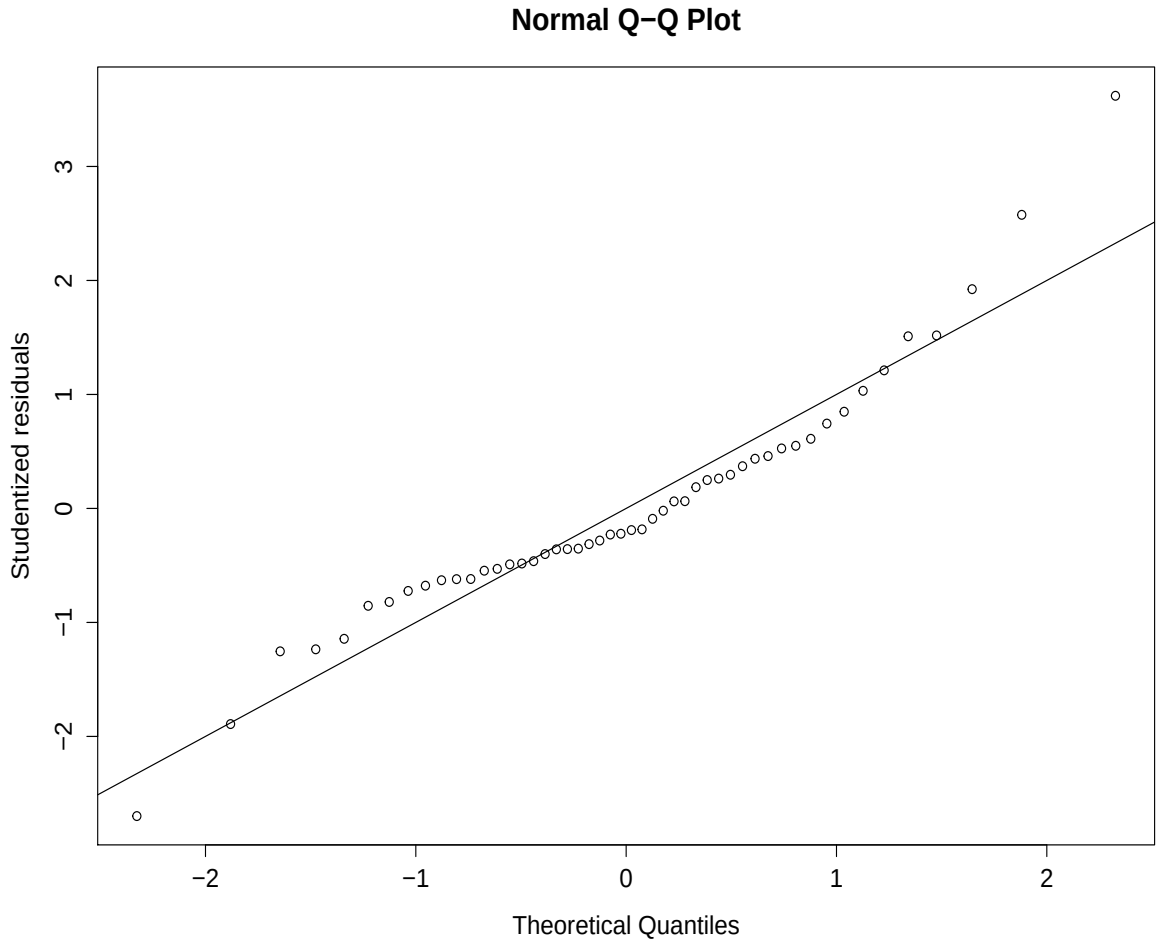
E.g: QQ-plots of raw residuals



More options

See the `Intro.R` script for the `R` code.

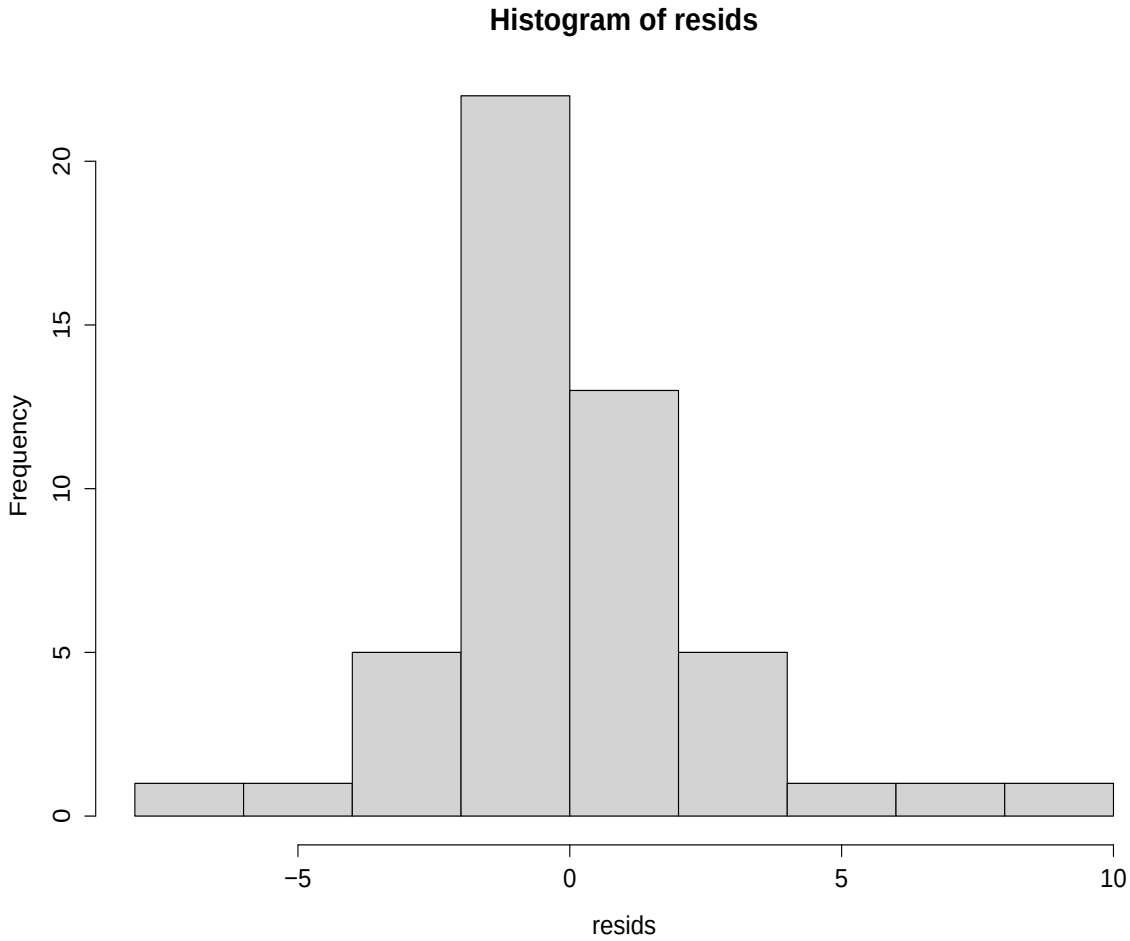
E.g: QQ-plots of studentized residuals



More options

See the `Intro.R` script for the `R` code.

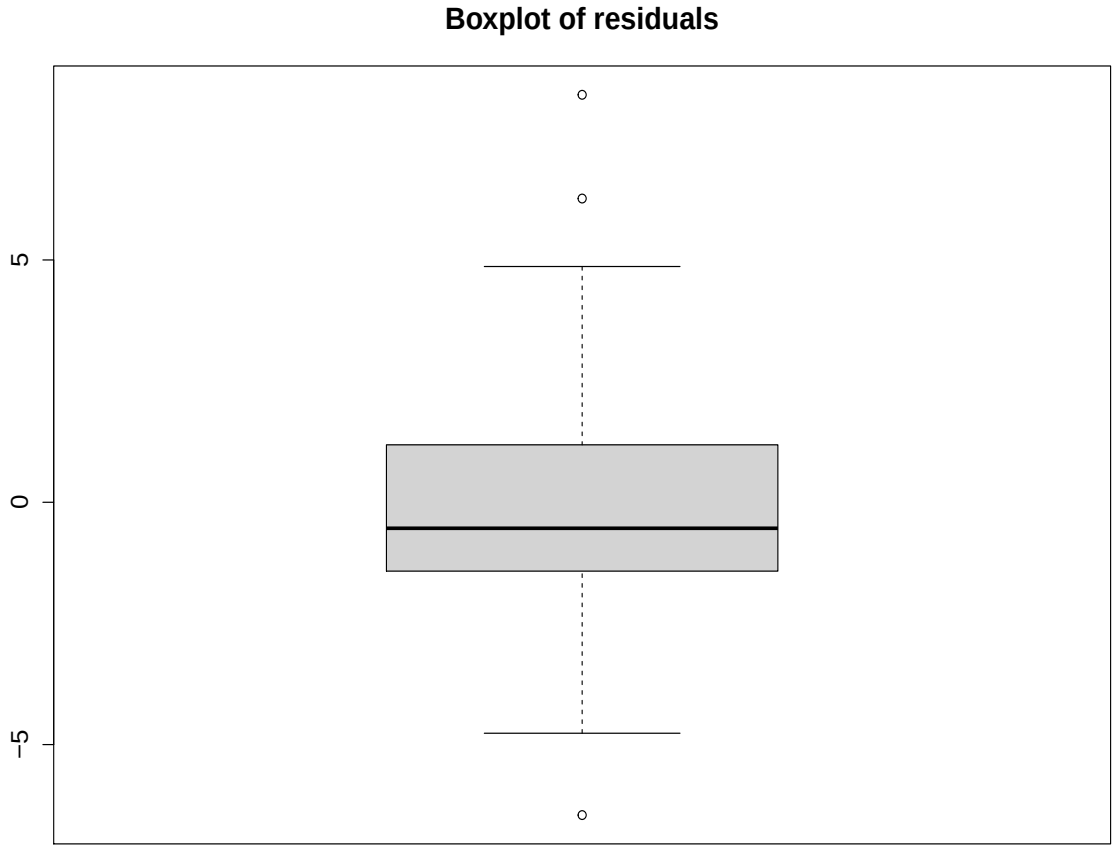
E.g: Histogram of raw residuals



More options

See the `Intro.R` script for the `R` code.

E.g: Boxplot of raw residuals



Multiple linear regression

- We have more explanatory variables:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p + \epsilon$$

- In matrix notation:

$$\boldsymbol{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{pmatrix}$$

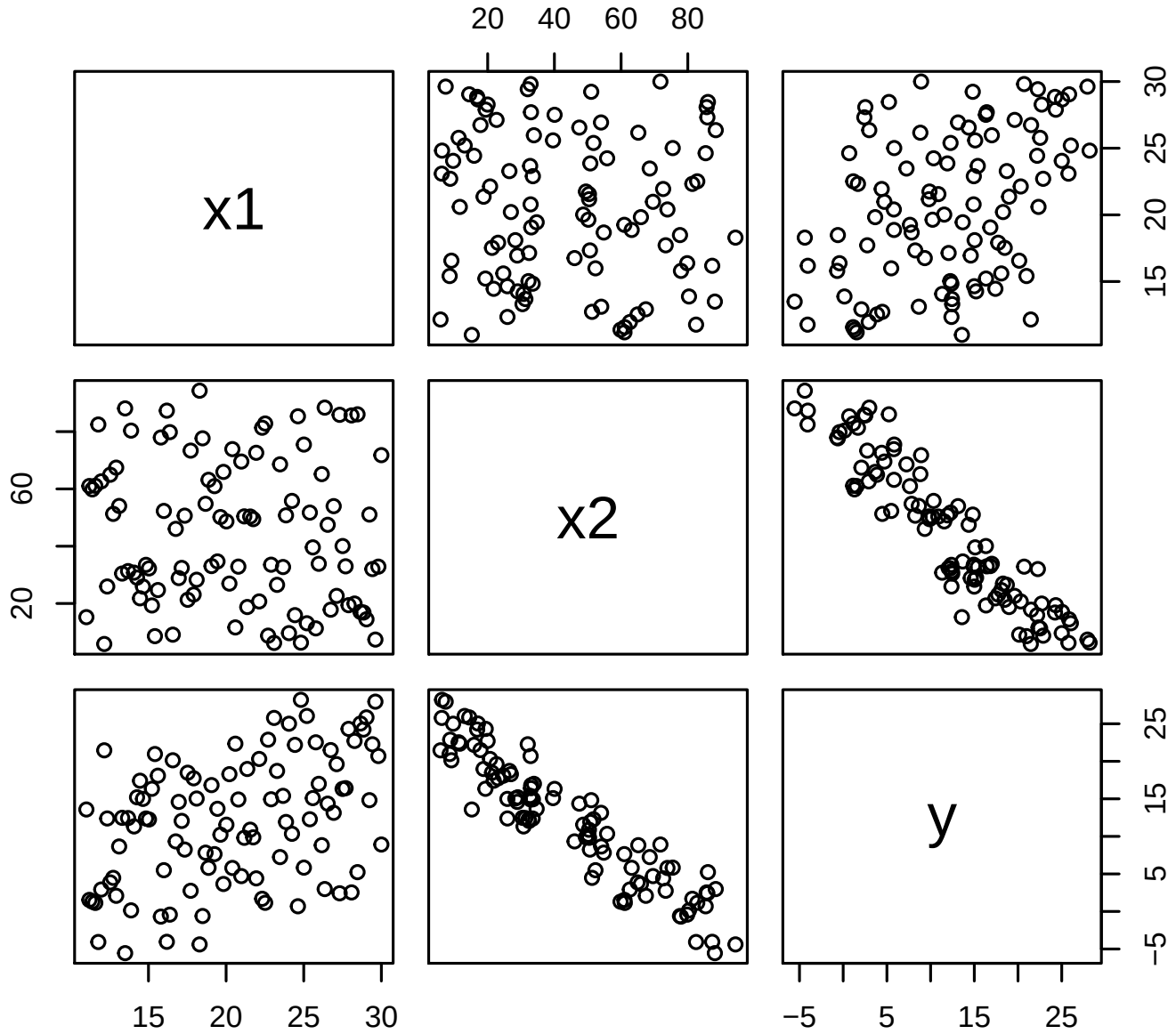
- Some results:
  - Hat-matrix: $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$
  - Predicted values $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y} = \boldsymbol{X}\hat{\beta}$
  - Residuals: $\hat{\boldsymbol{\epsilon}} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{y}'(\boldsymbol{I} - \boldsymbol{H})(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y} = \boldsymbol{y}'(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}$
  - Estimated variance: $\sigma^2 = \frac{\hat{\boldsymbol{\epsilon}}'\hat{\boldsymbol{\epsilon}}}{n-p}$

Linear models in R

Multiple regression model

- Let us consider a multiple regression
- See `Intro.R`
- **TO DO:** Estimate a multiple regression model

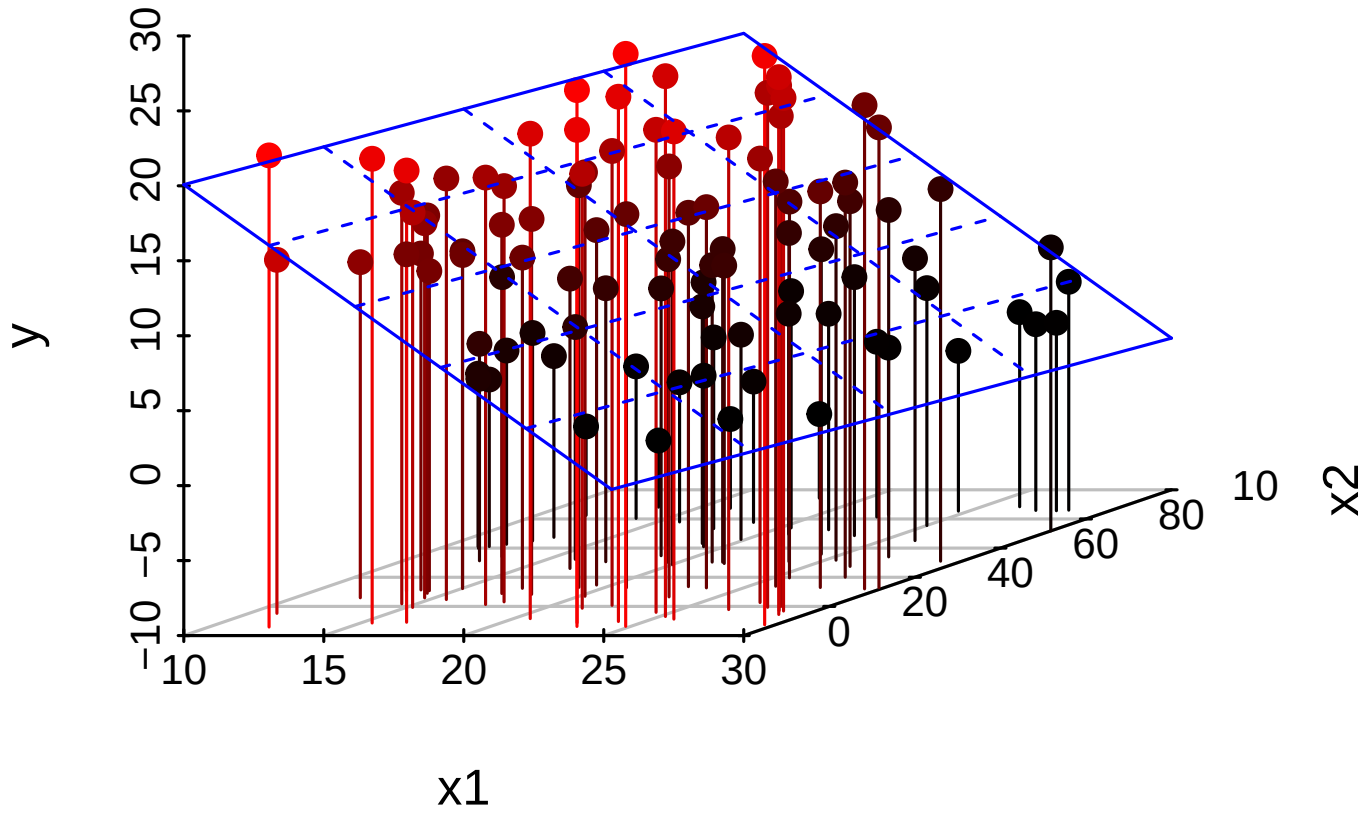
Linear models in R



Linear models in R

See Intro.R

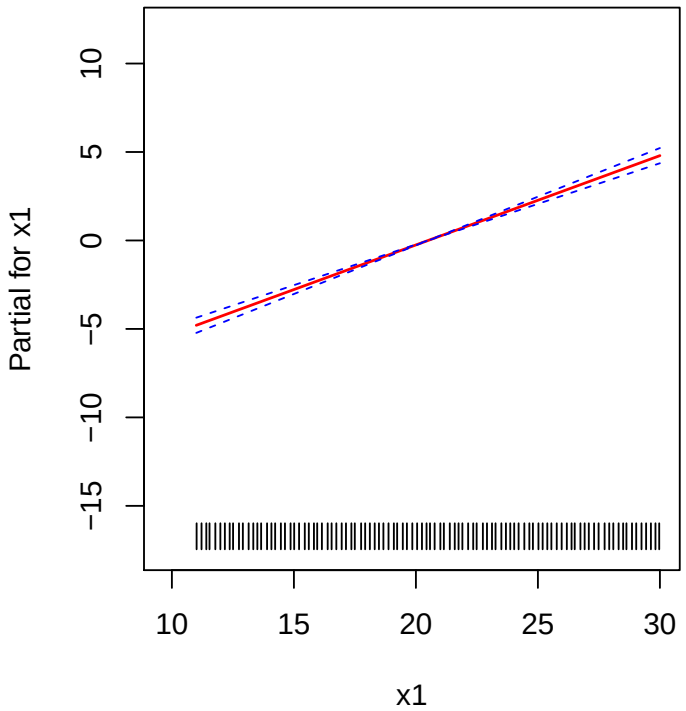
```
> library(scatterplot3d)
> ss<-scatterplot3d(df1,angle=35, pch =19, box=FALSE, type="h", highlight.3d=TRUE)
> ss$plane3d(mod1,lty.box="solid",col="blue")
```

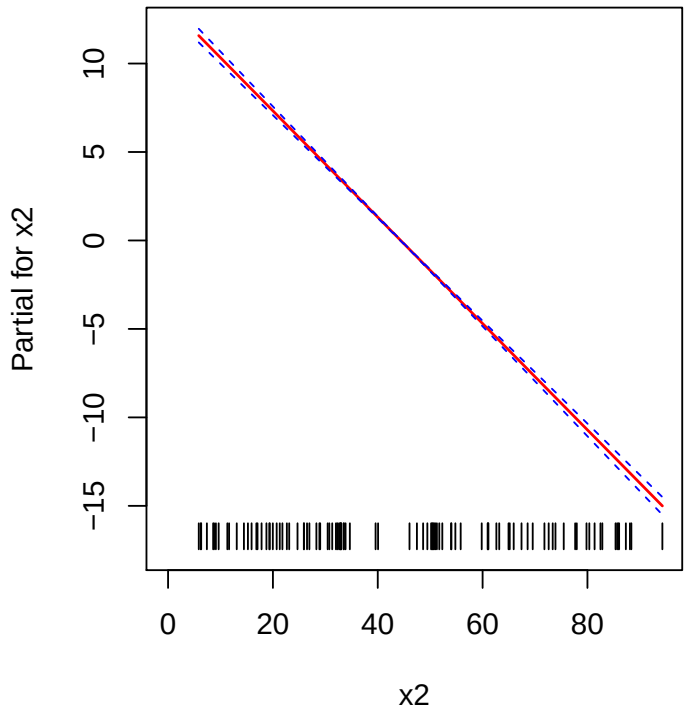


Linear models in R

See Intro.R

```
> termplot(mod1,rug=TRUE,se=TRUE,col.se="blue")
```





Linear models in R

- The principal argument of `lm` is a `formula`

R syntax	Mathematical syntax
`y ~ x1+x2`	$y = \alpha + \beta_1 x_1 + \beta_2 x_2$
`y ~ x1+x2-1`	$y = \beta_1 x_1 + \beta_2 x_2$ (without **intercept**)
`y ~ x1+I(X1^2)`	$y = \alpha + \beta_1 x_1 + \beta_2 x_2^2$
`y ~ x1+x2+x1:x2`	$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2$ (where x2 is categorical with 2 levels)
`y ~ x1*x2`	Equivalent to previous model
`y ~ x + fac`	`fac` is a categorical variable with different levels
`y ~ x + fac + fac:x`	`fac:x` allows a different slope for each different level of `fac`

When considering categorical variables, we need to set the `contrasts` attribute for the `factor`. See `?contrasts?` or `?C`

```
options(contrasts=c("contr.treatment", "contr.poly"))
```

Linear models with factor variables

Epidemiology survey at Comunidad de Madrid

See `Intro.R`

```
> rm(list=ls()) # Remove all previous variables
> salud <- read.table("data/salud.txt",header=TRUE, dec=",")
> class(salud)

[1] "data.frame"

> dim(salud)

[1] 7357   11
```

- Usually data are organized as a data frame/matrix by `rows` (cases/individuals) and `columns` (variables)

Example

Epidemiology survey at Comunidad de Madrid

The `data.frame` contains data from a survey conducted by the Service of Epidemiology of Comunidad de Madrid. The interest of the study was to know which variables influence the perception of health

Example

Epidemiology survey at Comunidad de Madrid

The `data.frame` contains data from a survey conducted by the Service of Epidemiology of Comunidad de Madrid. The interest of the study was to know which variables influence the perception of health

See `Intro.R` (cont.)

```
> names(salud)

 [1] "sexo"    "g01"     "g02"     "peso"    "altura"  "con_tab" "anio"
 [8] "educa"   "imc"     "bebedor" "edad"
```

Example

Variables

- `sexo`: 1 if male and 2 if female

Example

Variables

- `sexo`: 1 if male and 2 if female
- g01: answer to the question *"En general, cómo considera usted que es su salud?"* with 5 levels: 1= *Muy buena*, 2= *Buena*, 3= *Regular*, 4= *Mala*, 5= *Muy mala*

Example

Variables

- `sexo`: 1 if male and 2 if female
- g01: answer to the question *"En general, cómo considera usted que es su salud?"* with 5 levels: 1= *Muy buena*, 2= *Buena*, 3= *Regular*, 4= *Mala*, 5= *Muy mala*
- g02: g01 variable recoded into two categories 1= if the individual is considered healthy 0= if not

Example

Variables

- `sexo`: 1 if male and 2 if female
- g01: answer to the question *"En general, cómo considera usted que es su salud?"* with 5 levels: 1= *Muy buena*, 2= *Buena*, 3= *Regular*, 4= *Mala*, 5= *Muy mala*
- g02: g01 variable recoded into two categories 1= if the individual is considered healthy 0= if not
- `peso`: weight in kilograms

Example

Variables

- `sexo`: 1 if male and 2 if female
- g01: answer to the question *"En general, cómo considera usted que es su salud?"* with 5 levels: 1= *Muy buena*, 2= *Buena*, 3= *Regular*, 4= *Mala*, 5= *Muy mala*
- g02: g01 variable recoded into two categories 1= if the individual is considered healthy 0= if not
- `peso`: weight in kilograms
- `con_tab`: smoking habits (*consumo de tabaco*), with 2 categories: =1 *no fumador/fumador ocasional*, =2 *fumador diario/ex-fumador*.

Example

Variables

- `sexo`: 1 if male and 2 if female
- g01: answer to the question *"En general, cómo considera usted que es su salud?"* with 5 levels: 1= *Muy buena*, 2= *Buena*, 3= *Regular*, 4= *Mala*, 5= *Muy mala*
- g02: g01 variable recoded into two categories 1= if the individual is considered healthy 0= if not
- `peso`: weight in kilograms
- `con_tab`: smoking habits (*consumo de tabaco*), with 2 categories: =1 *no fumador/fumador ocasional*, =2 *fumador diario/ex-fumador*.
- `anio`: Year of the survey

Example

Variables

- `sexo`: 1 if male and 2 if female
- `g01`: answer to the question *"En general, cómo considera usted que es su salud?"* with 5 levels: 1= *Muy buena*, 2= *Buena*, 3= *Regular*, 4= *Mala*, 5= *Muy mala*
- `g02`: `g01` variable recoded into two categories 1= if the individual is considered healthy 0= if not
- `peso`: weight in kilograms
- `con_tab`: smoking habits (*consumo de tabaco*), with 2 categories: =1 *no fumador/fumador ocasional*, =2 *fumador diario/ex-fumador*.
- `anio`: Year of the survey
- `educa`: Education level with four categories: *Bajo*, *Medio-Bajo*, *Medio-Alto*, *Alto*.

Example

Variables

- `sexo`: 1 if male and 2 if female
- `g01`: answer to the question *"En general, cómo considera usted que es su salud?"* with 5 levels: 1= *Muy buena*, 2= *Buena*, 3= *Regular*, 4= *Mala*, 5= *Muy mala*
- `g02`: `g01` variable recoded into two categories 1= if the individual is considered healthy 0= if not
- `peso`: weight in kilograms
- `con_tab`: smoking habits (*consumo de tabaco*), with 2 categories: =1 *no fumador/fumador ocasional*, =2 *fumador diario/ex-fumador*.
- `anio`: Year of the survey
- `educa`: Education level with four categories: *Bajo*, *Medio-Bajo*, *Medio-Alto*, *Alto*.
- `imc`: Body Mass Index (*Indice de masa corporal*), i.e. $\text{BMI} = \text{Weight}/\text{Height}^2$

Example

Variables

- `sexo`: 1 if male and 2 if female
- `g01`: answer to the question *"En general, cómo considera usted que es su salud?"* with 5 levels: 1= *Muy buena*, 2= *Buena*, 3= *Regular*, 4= *Mala*, 5= *Muy mala*
- `g02`: `g01` variable recoded into two categories 1= if the individual is considered healthy 0= if not
- `peso`: weight in kilograms
- `con_tab`: smoking habits (*consumo de tabaco*), with 2 categories: =1 *no fumador/fumador ocasional*, =2 *fumador diario/ex-fumador*.
- `anio`: Year of the survey
- `educa`: Education level with four categories: *Bajo*, *Medio-Bajo*, *Medio-Alto*, *Alto*.
- `imc`: Body Mass Index (*Indice de masa corporal*), i.e. $\text{BMI} = \text{Weight}/\text{Height}^2$
- `bebedor`: *"How often do you drink alcohol?* with three levels: *Poco/Nada*, *Ocasionalmente*, and *Frecuentemente*

Example

Variables

- `sexo`: 1 if male and 2 if female
- `g01`: answer to the question *"En general, cómo considera usted que es su salud?"* with 5 levels: 1= *Muy buena*, 2= *Buena*, 3= *Regular*, 4= *Mala*, 5= *Muy mala*
- `g02`: `g01` variable recoded into two categories 1= if the individual is considered healthy 0= if not
- `peso`: weight in kilograms
- `con_tab`: smoking habits (*consumo de tabaco*), with 2 categories: =1 *no fumador/fumador ocasional*, =2 *fumador diario/ex-fumador*.
- `anio`: Year of the survey
- `educa`: Education level with four categories: *Bajo*, *Medio-Bajo*, *Medio-Alto*, *Alto*.
- `imc`: Body Mass Index (*Indice de masa corporal*), i.e. $\text{BMI} = \text{Weight}/\text{Height}^2$
- `bebedor`: *"How often do you drink alcohol?* with three levels: *Poco/Nada*, *Ocasionalmente*, and *Frecuentemente*
- `edad`: age

```
> summary(salud)
```

```
      sexo            g01             g02              peso       
 Min.   :1.000   Min.   :1.000   Min.   :0.0000   Min.   : 34.00  
 1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.0000   1st Qu.: 59.00  
 Median :2.000   Median :2.000   Median :1.0000   Median : 69.00  
 Mean   :1.502   Mean   :2.057   Mean   :0.8089   Mean   : 69.51  
 3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.: 79.00  
 Max.   :2.000   Max.   :5.000   Max.   :1.0000   Max.   :175.00  
     altura         con_tab          anio          educa            imc       
 Min.   :120.0   Min.   :1.000   Min.   :2001   Min.   :1.000   Min.   :14.69  
 1st Qu.:161.0   1st Qu.:1.000   1st Qu.:2002   1st Qu.:2.000   1st Qu.:21.72  
 Median :168.0   Median :2.000   Median :2003   Median :3.000   Median :24.06  
 Mean   :168.5   Mean   :1.549   Mean   :2003   Mean   :2.921   Mean   :24.39  
 3rd Qu.:175.0   3rd Qu.:2.000   3rd Qu.:2004   3rd Qu.:4.000   3rd Qu.:26.53  
 Max.   :200.0   Max.   :2.000   Max.   :2004   Max.   :4.000   Max.   :62.87  
    bebedor           edad       
 Min.   :0.0000   Min.   :18.00  
 1st Qu.:0.0000   1st Qu.:28.00  
 Median :1.0000   Median :39.00  
 Mean   :0.6397   Mean   :39.24  
 3rd Qu.:1.0000   3rd Qu.:49.00  
 Max.   :2.0000   Max.   :64.00  
```

- We can access to the variable with $ symbol as for example `salud$sexo`

```
> summary(salud$sexo)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   1.502   2.000   2.000
```

- We can access to the variable with $ symbol as for example `salud$sexo`

```
> summary(salud$sexo)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   1.502   2.000   2.000
```

- By default R considers all the variables as **numeric**

- We can access to the variable with `$` symbol as for example `salud$sexo`

```
> summary(salud$sexo)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   1.502   2.000   2.000
```

- By default `R` considers all the variables as **numeric**
- There are 4 categorical variables: `sexo`, `con_tab`, `educa` and `bebedor`, we need to convert them to `factor` variables, i.e.:

- We can access to the variable with `$` symbol as for example `salud$sexo`

```
> summary(salud$sexo)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   1.502   2.000   2.000
```

- By default `R` considers all the variables as **numeric**
- There are 4 categorical variables: `sexo`, `con_tab`, `educa` and `bebedor`, we need to convert them to `factor` variables, i.e.:

- We can access to the variable with `$` symbol as for example `salud$sexo`

```
> summary(salud$sexo)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   1.000   2.000   1.502   2.000   2.000
```

- By default `R` considers all the variables as **numeric**
- There are 4 categorical variables: `sexo`, `con_tab`, `educa` and `bebedor`, we need to convert them to `factor` variables, i.e.:

```
>  salud$g02     <- factor(salud$g02)
>  salud$sexo    <- factor(salud$sexo)
>  salud$con_tab <- factor(salud$con_tab)
>  salud$educa   <- factor(salud$educa)
>  salud$bebedor <- factor(salud$bebedor)
```

- Then, now

```
> summary(salud)
```

```
 sexo          g01          g02           peso            altura      con_tab
 1:3666   Min.   :1.000   0:1406   Min.   : 34.00   Min.   :120.0   1:3320
 2:3691   1st Qu.:2.000   1:5951   1st Qu.: 59.00   1st Qu.:161.0   2:4037
          Median :2.000            Median : 69.00   Median :168.0
          Mean   :2.057            Mean   : 69.51   Mean   :168.5
          3rd Qu.:2.000            3rd Qu.: 79.00   3rd Qu.:175.0
          Max.   :5.000            Max.   :175.00   Max.   :200.0
     anio          educa           imc          bebedor         edad
 Min.   :2001   1: 546   Min.   :14.69   0:2977   Min.   :18.00
 1st Qu.:2002   2:1928   1st Qu.:21.72   1:4054   1st Qu.:28.00
 Median :2003   3:2442   Median :24.06   2: 326   Median :39.00
 Mean   :2003   4:2441   Mean   :24.39            Mean   :39.24
 3rd Qu.:2004            3rd Qu.:26.53            3rd Qu.:49.00
 Max.   :2004            Max.   :62.87            Max.   :64.00
```

- Using `attach` command we can have direct access to the variables

  ```
  > attach(salud)
  ```

- Some interesting commands to analyze variables are `table`, `tapply`, `xtabs`

```
> table(sexo)

sexo
   1    2
3666 3691
```

- Some interesting commands to analyze variables are `table`, `tapply`, `xtabs`

```
> table(sexo)

sexo
   1    2
3666 3691
```

```
> tapply(peso,bebedor,FUN=mean)

       0        1        2
67.30971 70.86384 72.75767
```

- Some interesting commands to analyze variables are `table`, `tapply`, `xtabs`

```
>  table(sexo)

sexo
   1    2
3666 3691
```

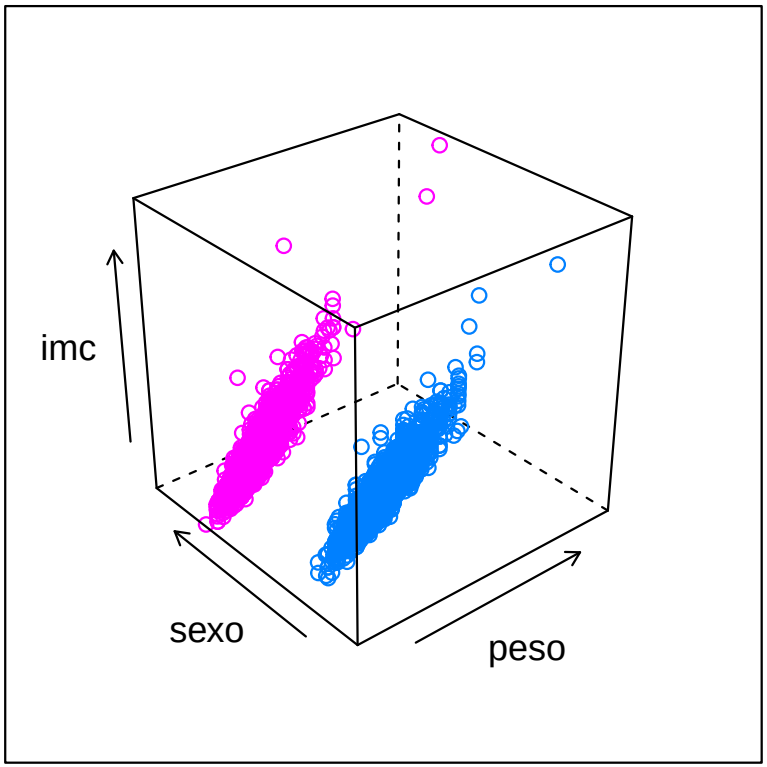
```
> tapply(peso,bebedor,FUN=mean)

       0        1        2
67.30971 70.86384 72.75767
```

```
> xtabs(~sexo+bebedor,data=salud)

    bebedor
sexo    0    1    2
   1 1015 2425  226
   2 1962 1629  100
```

Preliminary Exploratory Data Analysis

- Let us consider the variables: **body mass index** (`imc`), **weight** (`peso`) and `sexo`

```
> cloud( imc ~ peso*sexo, groups = sexo, data=salud)
```



Linear models in R

- First, we are interested in studying the relationship between the **body mass index** (imc) and the **weight** (peso)

```
> modelo1 <- lm(imc ~ peso , data = salud)
> modelo1

Call:
lm(formula = imc ~ peso, data = salud)

Coefficients:
(Intercept)         peso
     8.7608       0.2248
```

- The lm object modelo1 inmediately gives the parameters
- A lm object is a list of several objects

Linear models in R

```
> attributes(modelo1)

$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"
```

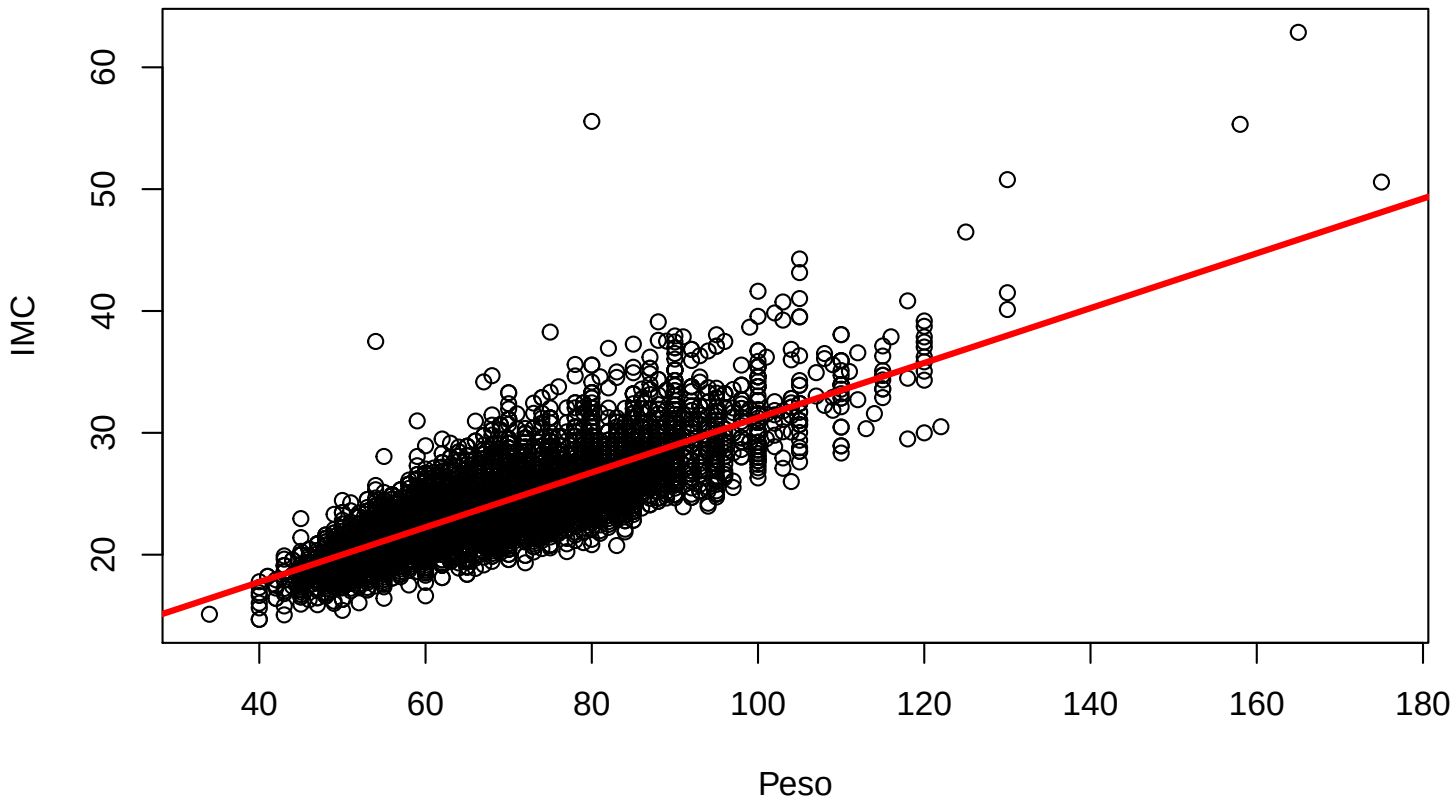
- We can access to each attribute using `$`, i.e. `modelo1$fitted.values`, etc ...

Linear models in R

```
> summary(modelo1)


Call:
lm(formula = imc ~ peso, data = salud)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6705 -1.4898 -0.2098  1.2121 28.8139

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 8.760772   0.136390   64.23   <2e-16 ***
peso        0.224816   0.001926  116.75   <2e-16 ***
---
Signif. codes:  0
```

Linear models in R

```
> plot(peso,imc,xlab="Peso",ylab="IMC")
> abline(modelo1,lwd=3,col=2)
```



Linear models in R

- Now, we are interested in including the variable `sexo`

```
>   modelo2 <- lm(imc~peso+sexo,data=salud)
>   summary(modelo2)

Call:
lm(formula = imc ~ peso + sexo, data = salud)

Residuals:
    Min      1Q  Median      3Q     Max
-7.4773 -1.3337 -0.0957  1.2043 26.8843

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.329184   0.171741   19.39   <2e-16 ***
peso        0.284001   0.002164  131.25   <2e-16 ***
sexo2       2.626471   0.058921   44.58   <2e-16 ***
---
Signif. codes:  0
```

- How can you interpret the coefficients?

Linear models in R

- Write the Equation of the model

Linear models in R

- Write the Equation of the model

```
> coefficients(modelo2)

(Intercept)        peso       sexo2
  3.3291844   0.2840009   2.6264714
```

Linear models in R

- Write the Equation of the model

```
> coefficients(modelo2)

(Intercept)        peso       sexo2
  3.3291844   0.2840009   2.6264714
```

- **Model:**

$$imc = \beta_0 + \beta_1 * peso + \beta_2 * sexo2$$

Linear models in R

- Write the Equation of the model

```
> coefficients(modelo2)

(Intercept)        peso       sexo2
  3.3291844   0.2840009   2.6264714
```

- **Model:**

$$imc = \beta_0 + \beta_1 * peso + \beta_2 * sexo2$$

- Note that with categorical variables, `lm` compares each level to the reference level, intercept being the mean of the reference group

Linear models in R

- Write the Equation of the model

```
> coefficients(modelo2)

(Intercept)        peso       sexo2
  3.3291844   0.2840009   2.6264714
```

- **Model:**

$$imc = \beta_0 + \beta_1 * peso + \beta_2 * sexo2$$

- Note that with categorical variables, `lm` compares each level to the reference level, intercept being the mean of the reference group
- Hence, $sexo2$ is a dummy variable taking value $= 1$ (when `sexo` is *female*) and $= 0$ when is *male*

Linear models in R

- Write the Equation of the model

```
> coefficients(modelo2)

(Intercept)        peso       sexo2
  3.3291844   0.2840009   2.6264714
```

- **Model:**

$$imc = \beta_0 + \beta_1 * peso + \beta_2 * sexo2$$

- Note that with categorical variables, `lm` compares each level to the reference level, intercept being the mean of the reference group
- Hence, $sexo2$ is a dummy variable taking value $= 1$ (when `sexo` is *female*) and $= 0$ when is *male*
- How can we interpret β_2 ?

Linear models in R

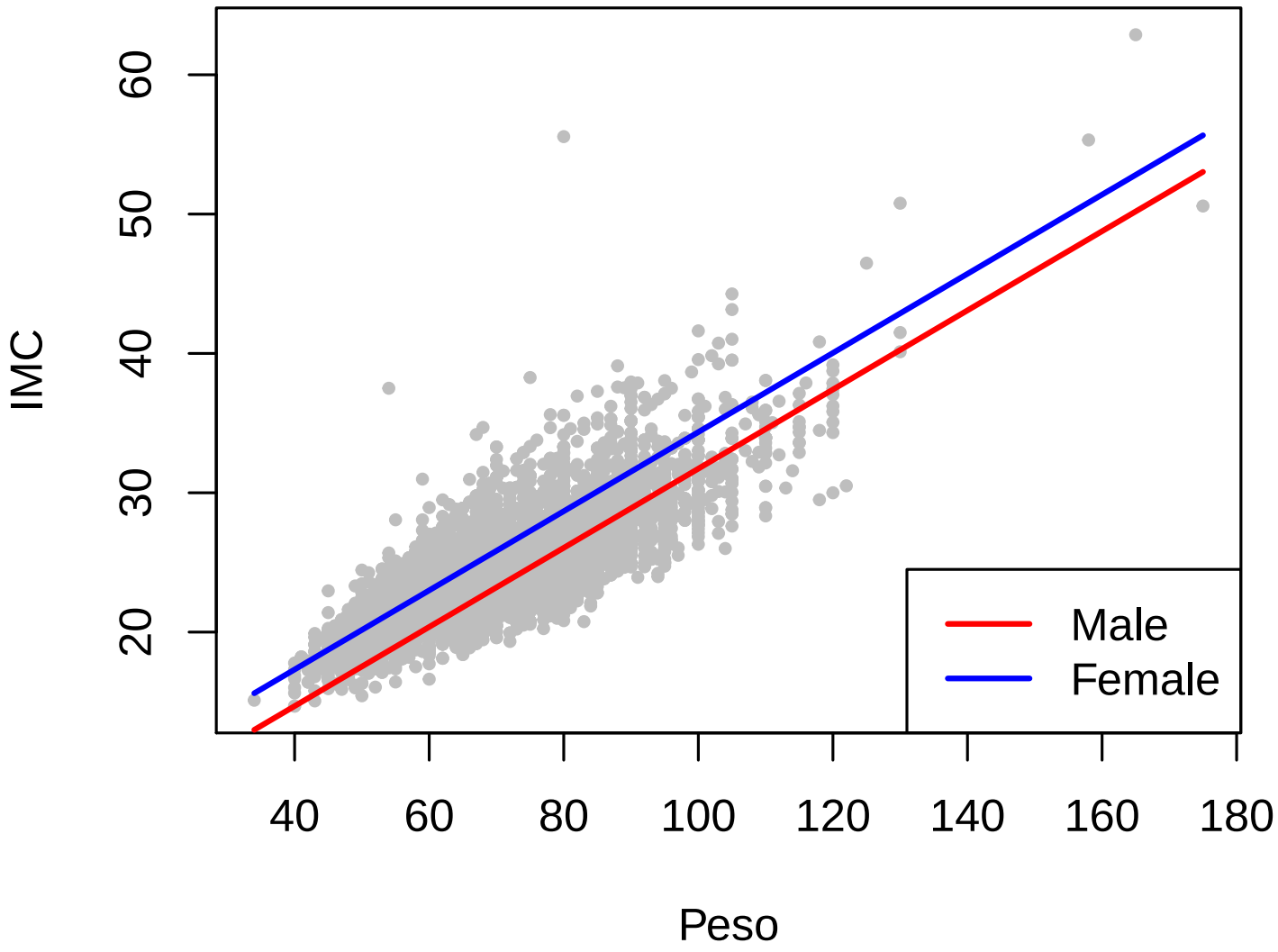
- `modelo2` is fitting two lines for each level of the variable `sexo`. How can we obtain these two lines?
- First,

```
> y.ajustados.mod2 <- fitted.values(modelo2)
```

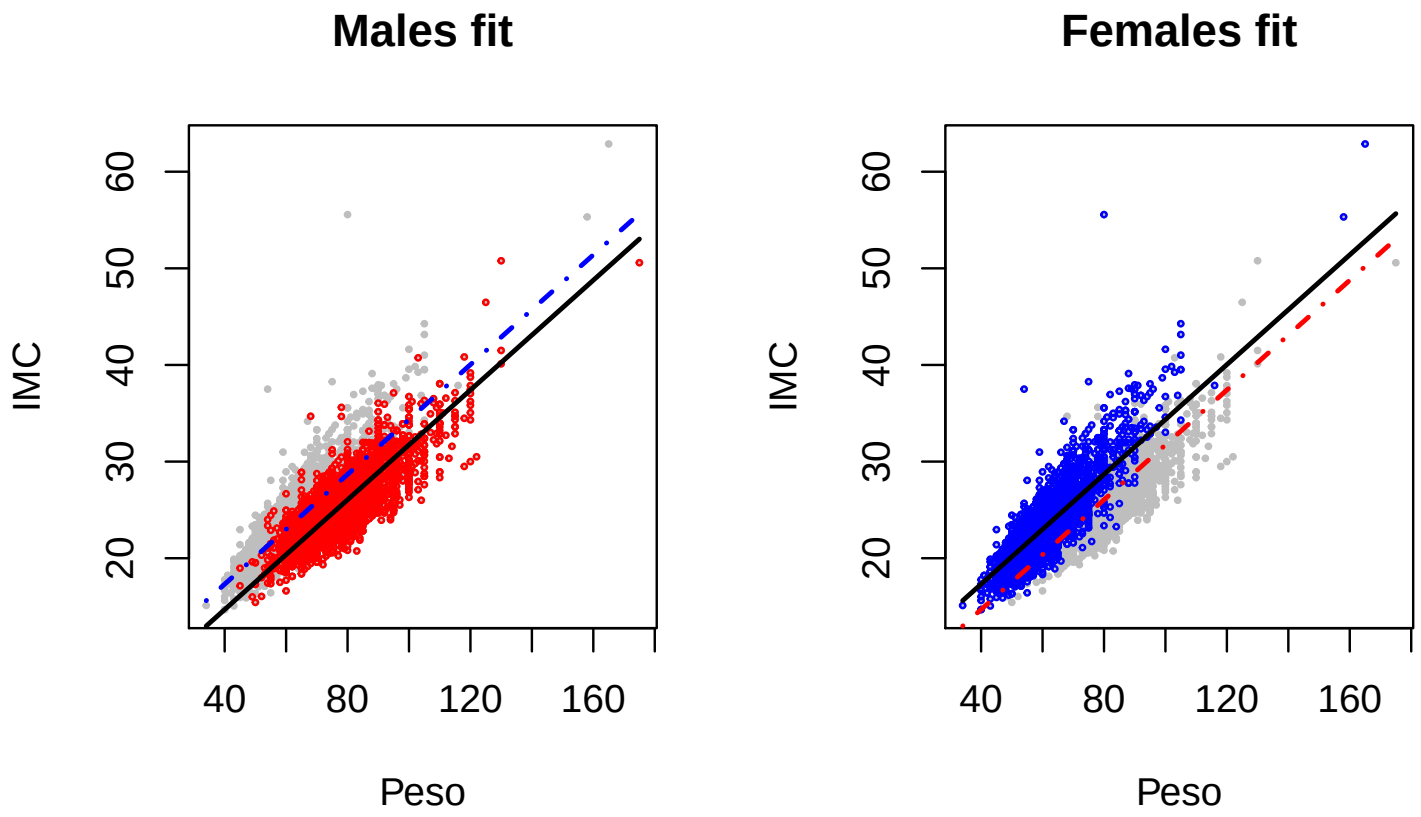
gives all the fitted values given the values of `peso` and `sexo`

- But, we are interested in each particular line
- We can use the function `predict.lm` or simply `predict`
- See `Intro.R` script

Linear models in R



Linear models in R



- Is `modelo2` realistic? Can we propose a better alternative?

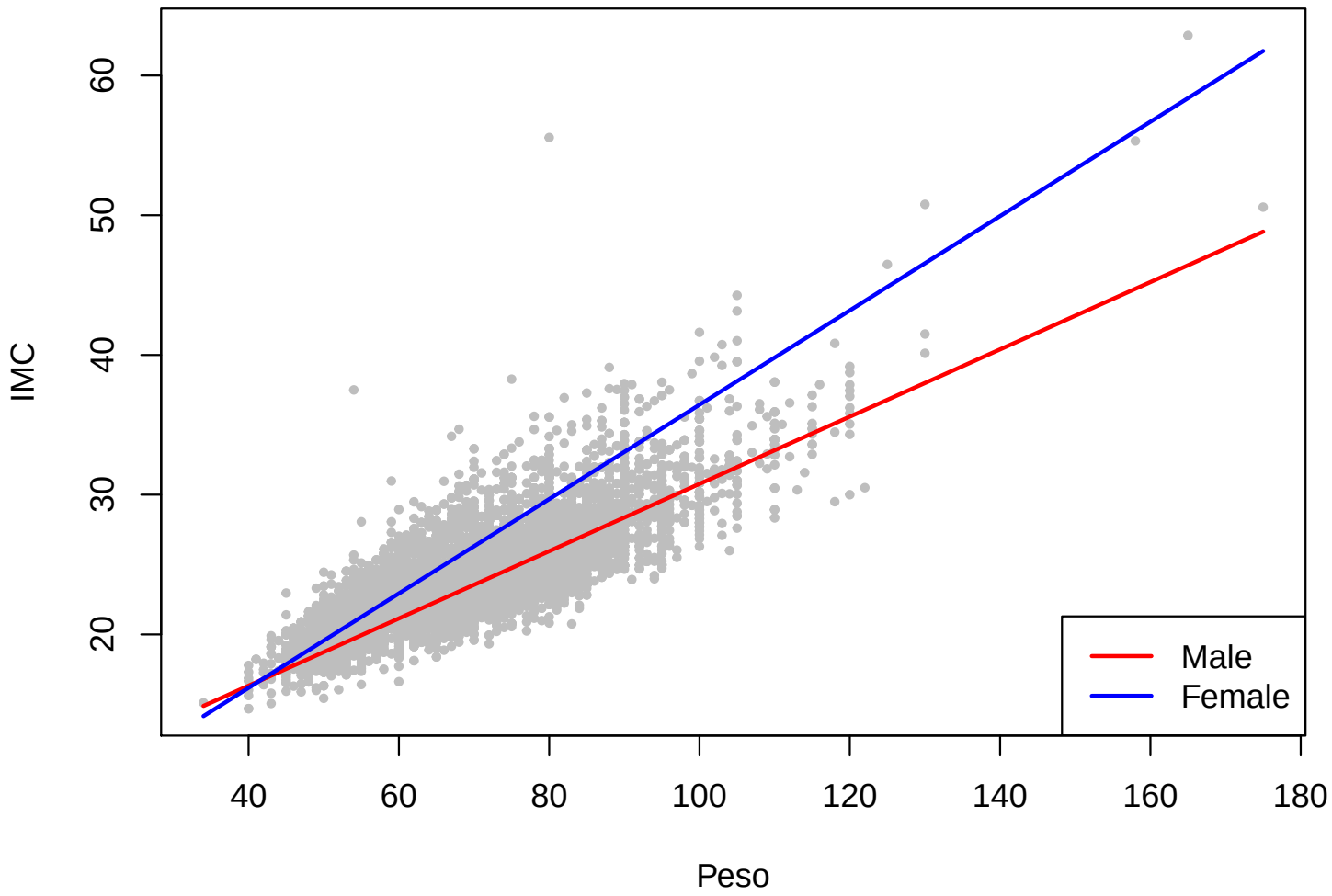
Linear models in R

including interactions

```
> modelo3 <- lm(imc~peso+sexo+peso:sexo)
> summary(modelo3)


Call:
lm(formula = imc ~ peso + sexo + peso:sexo)

Residuals:
    Min      1Q  Median      3Q     Max
-7.7643 -1.2680 -0.0487  1.1384 25.8760

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.700237   0.221165   30.30   <2e-16 ***
peso         0.240722   0.002810   85.67   <2e-16 ***
sexo2       -4.023201   0.294134  -13.68   <2e-16 ***
peso:sexo2   0.096865   0.004204   23.04   <2e-16 ***
---
Signif. codes:  0
```

Linear models in R

including interactions



Linear models in R

Variable selection

- There are several methods for variable selection in linear models
  - stepwise selection (`stepAIC()` from the `MASS` package)
  - dropping one-term (`dropterm()` from the `MASS` package)
  - all-subsets regression (`regsubsets` from `leaps` package)
- We will use the function `anova` which compares the deviance (residuals variance) and applies a F-test and AIC (Akaike Information Criteria)
- Now we include an additional variable such as `bebedor`

```
> modelo4 <- lm(imc~peso+sexo+sexo:peso+bebedor+bebedor:peso+sexo:bebedor)
```

- Or equivalently

```
> modelo4 <- lm(imc~sexo*peso+bebedor*peso+sexo:bebedor)
```

Linear models in R

Variable selection

```
>   anova(modelo4)

Analysis of Variance Table

Response: imc
               Df Sum Sq Mean Sq    F value    Pr(>F)
peso            1  68924   68924 18628.3242 < 2.2e-16 ***
sexo            1   7911    7911  2138.1357 < 2.2e-16 ***
bebedor         2    107      53    14.4303 5.563e-07 ***
peso:sexo       1   1918    1918   518.4414 < 2.2e-16 ***
peso:bebedor    2     54      27     7.2363 0.0007251 ***
sexo:bebedor    2     17       8     2.2802 0.1023316
Residuals    7347  27183       4
---
Signif. codes:  0
```

Linear models in R

Variable selection

```
>   modelo5 <- lm(imc~peso+sexo+peso:sexo+bebedor+bebedor:peso)
> anova(modelo5,modelo4)


Analysis of Variance Table

Model 1: imc ~ peso + sexo + peso:sexo + bebedor + bebedor:peso
Model 2: imc ~ peso + sexo + sexo:peso + bebedor + bebedor:peso + sexo:bebedor
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1   7349 27200
2   7347 27183  2    16.873 2.2802 0.1023
```

- Observe that, we compare the models as: `anova(null.model,alternative.model)`, and hence, `null.model` is the simplest model. And then, with a p-value > 0.05 we do not reject the null hypothesis.

Linear models in R

Variable selection

```
> library(MASS)
>   dropterm(modelo4,test="F")


Single term deletions

Model:
imc ~ peso + sexo + sexo:peso + bebedor + bebedor:peso + sexo:bebedor
             Df Sum of Sq   RSS     AIC F Value     Pr(F)
<none>                    27183  9635.3
peso:sexo     1   1795.56 28979 10103.8  485.30 < 2.2e-16 ***
peso:bebedor  2     47.23 27231  9644.0    6.38  0.001699 **
sexo:bebedor  2     16.87 27200  9635.8    2.28  0.102332
---
Signif. codes:  0
```

- Using the function `dropterm` in `MASS` package, we found the same conclusion, in this case the null hypothesis is that the term is $= 0$, then for a p-value $< 0{,}05$ we reject the null hypothesis

Linear models in R

Extensions

- **Q:** Can we do something better than straight lines ?

Linear models in R

Extensions

- **Q:** Can we do something better than straight lines ?
- **A:** YES

Linear models in R

Extensions

- **Non-linear models** are beyond this course, but still we can use `lm` to fit different alternatives

Linear models in R

Extensions

- **Non-linear models** are beyond this course, but still we can use `lm` to fit different alternatives

Polynomial, exponential and natural cubic spline regression models:

1. **Polynomial regression of degree p:**

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + ... + \beta_p x^{p-1}$$

Linear models in R

Extensions

- **Non-linear models** are beyond this course, but still we can use `lm` to fit different alternatives

Polynomial, exponential and natural cubic spline regression models:

1. **Polynomial regression of degree p:**

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + ... + \beta_p x^{p-1}$$

2. **Exponential models:**

$$y = \alpha \exp(\beta x) \Longleftrightarrow \log(y) = \log(\alpha) + \beta x$$

Linear models in R

Extensions

- **Non-linear models** are beyond this course, but still we can use `lm` to fit different alternatives

Polynomial, exponential and natural cubic spline regression models:

1. **Polynomial regression of degree p:**

$$y = \alpha + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + ... + \beta_p x^{p-1}$$

2. **Exponential models:**

$$y = \alpha \exp(\beta x) \Longleftrightarrow \log(y) = \log(\alpha) + \beta x$$

3. **Natural cubic spline models:**

$$y = \alpha + \mathsf{ns}(x, \mathsf{df})$$

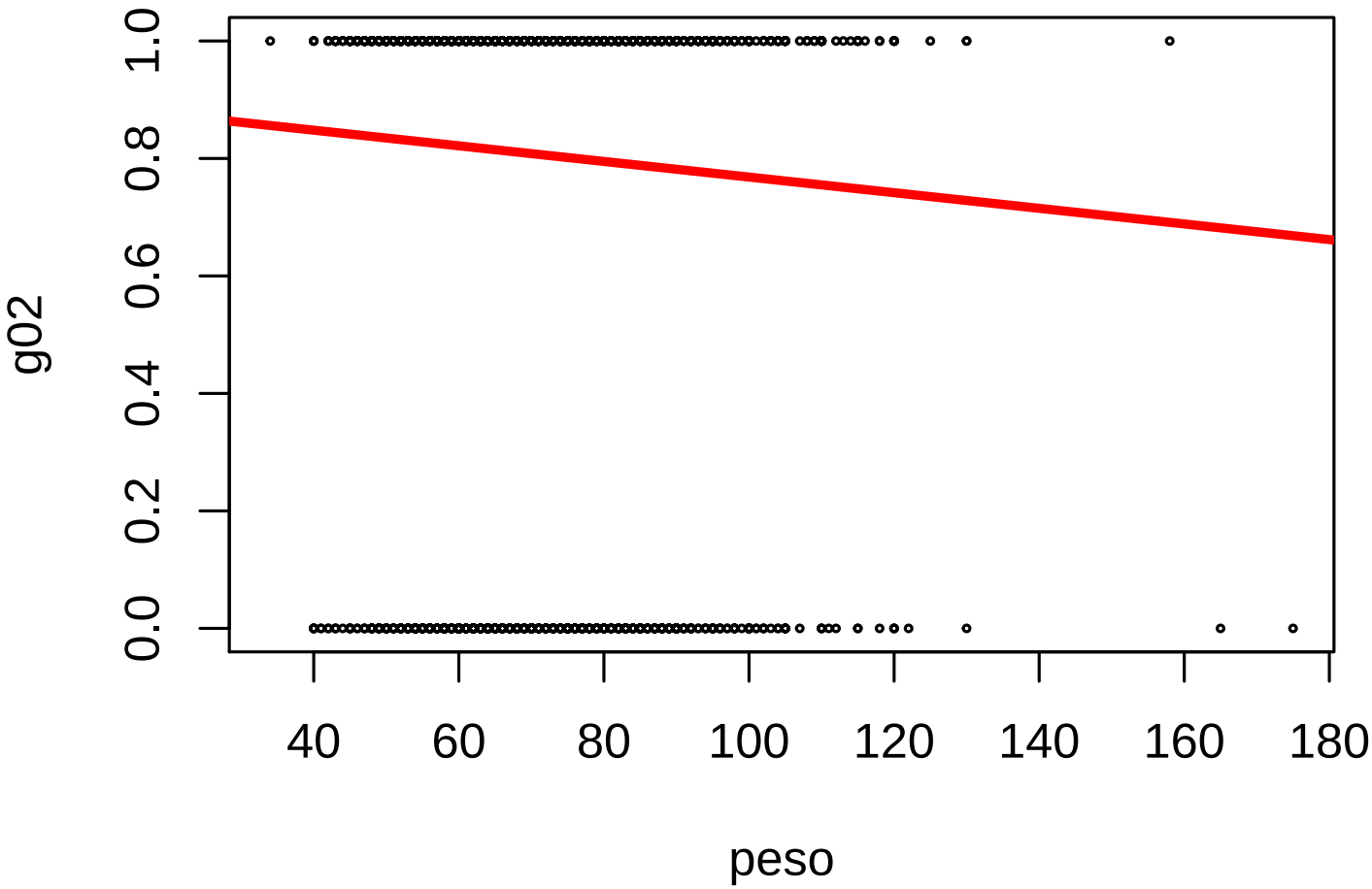
Why GLM's?

Motivation example: Health perception data

- Consider now the variable **"Health perception"** (`g02`) in `salud` data frame
- We want to relate `g02` with `peso`
- Fit a linear model
- Would you use a LM?

Why GLM's?

Motivation example: Health perception data (cont.)

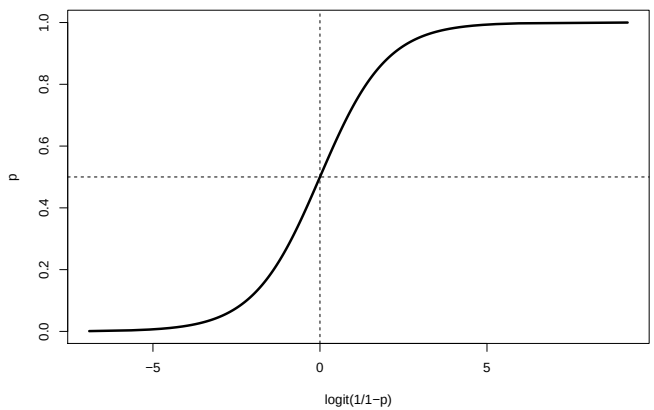


Why GLM's?

Motivation example: Health perception data (cont.)

- Berkson (1944) proposed the logistic regression model using the **logit** function
- **Definition:** **logit** of a number p between 0 and 1 is given by

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$



- Instead of working with a response variable $\in (0, 1)$ we transform the data to the **logit**.
- John Nelder in 1972 introduced the GLM theory, logistic regression is a particular case of a GLM.

Why GLM's?

Motivation example: Turbines experiment

- Nelson (1982) performed an experiment to determine the relationship between the time in use and the number of fissures in turbines

See `Intro.R`

```
> turbinas <- read.table("data/fisuras.txt",header=TRUE)
```

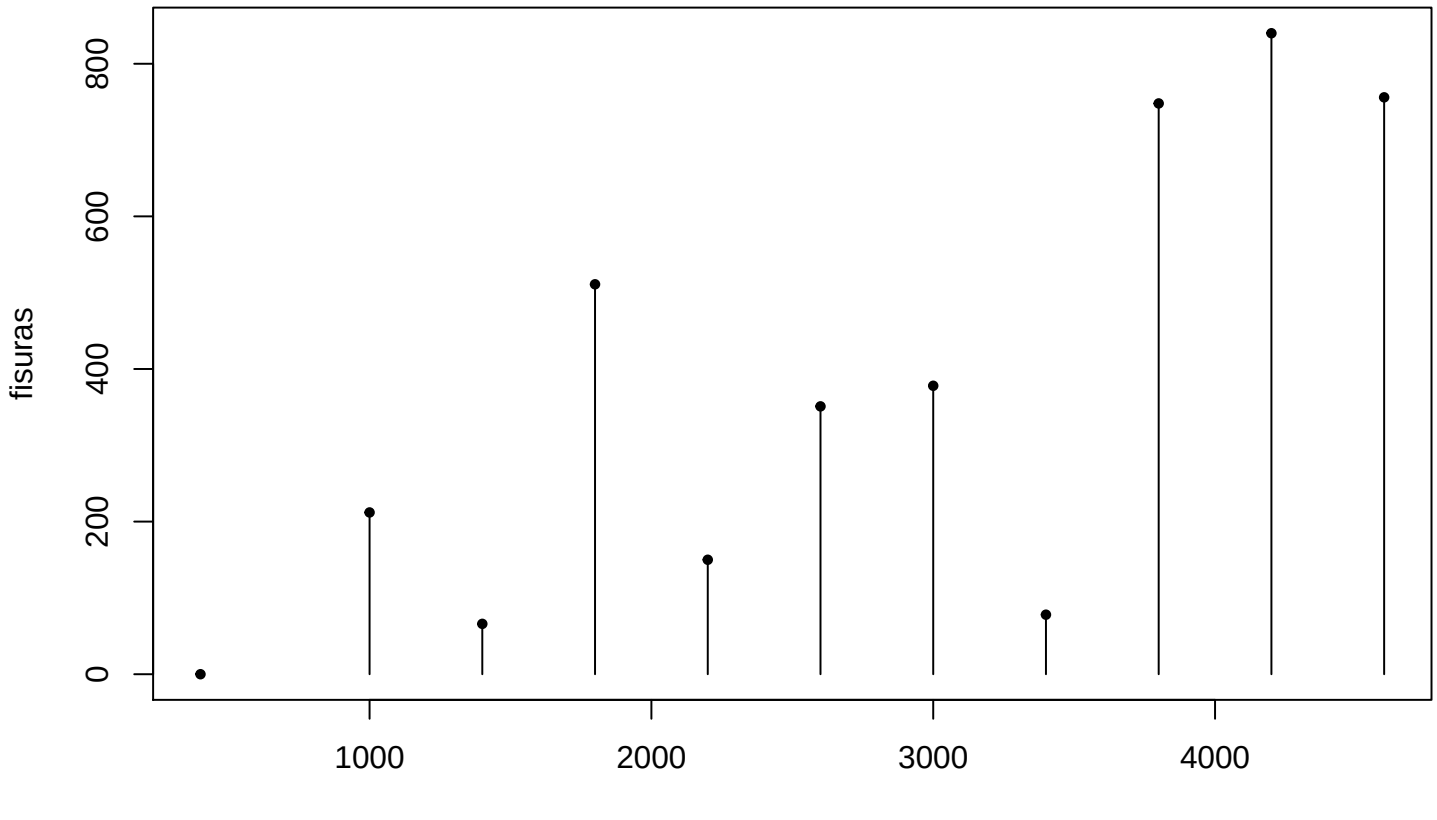
- The response variable is the number of fissures (discrete count data), and the exploratory variable is the number of hours in use.
- Let us plot the data.

Why GLM's?

Motivation example: Turbines experiment (cont.)

See `Intro.R`

```
> plot(turbinas,type="h"); points(turbinas,cex=.6,pch=19)
```



Why GLM's?

Motivation example: Turbines experiment (cont.)

- The data follows a **Poisson** distribution, i.e. $\boldsymbol{y} \sim \mathcal{P}\text{ois}(\boldsymbol{\mu})$
- Hence, theoretically $\mathbb{E}[\boldsymbol{y}] = \mathbb{V}\text{ar}[\boldsymbol{y}] = \boldsymbol{\mu}$
- Fit a LM

```
> ex2<- lm(fisuras~horas,data=turbinas)
```

Why GLM's?

Motivation example: Turbines experiment (cont.)

- The data follows a **Poisson** distribution, i.e. $\boldsymbol{y} \sim \mathcal{P}\text{ois}(\boldsymbol{\mu})$
- Hence, theoretically $\mathbb{E}[\boldsymbol{y}] = \mathbb{V}\text{ar}[\boldsymbol{y}] = \boldsymbol{\mu}$
- Fit a LM

```
> ex2<- lm(fisuras~horas,data=turbinas)
```

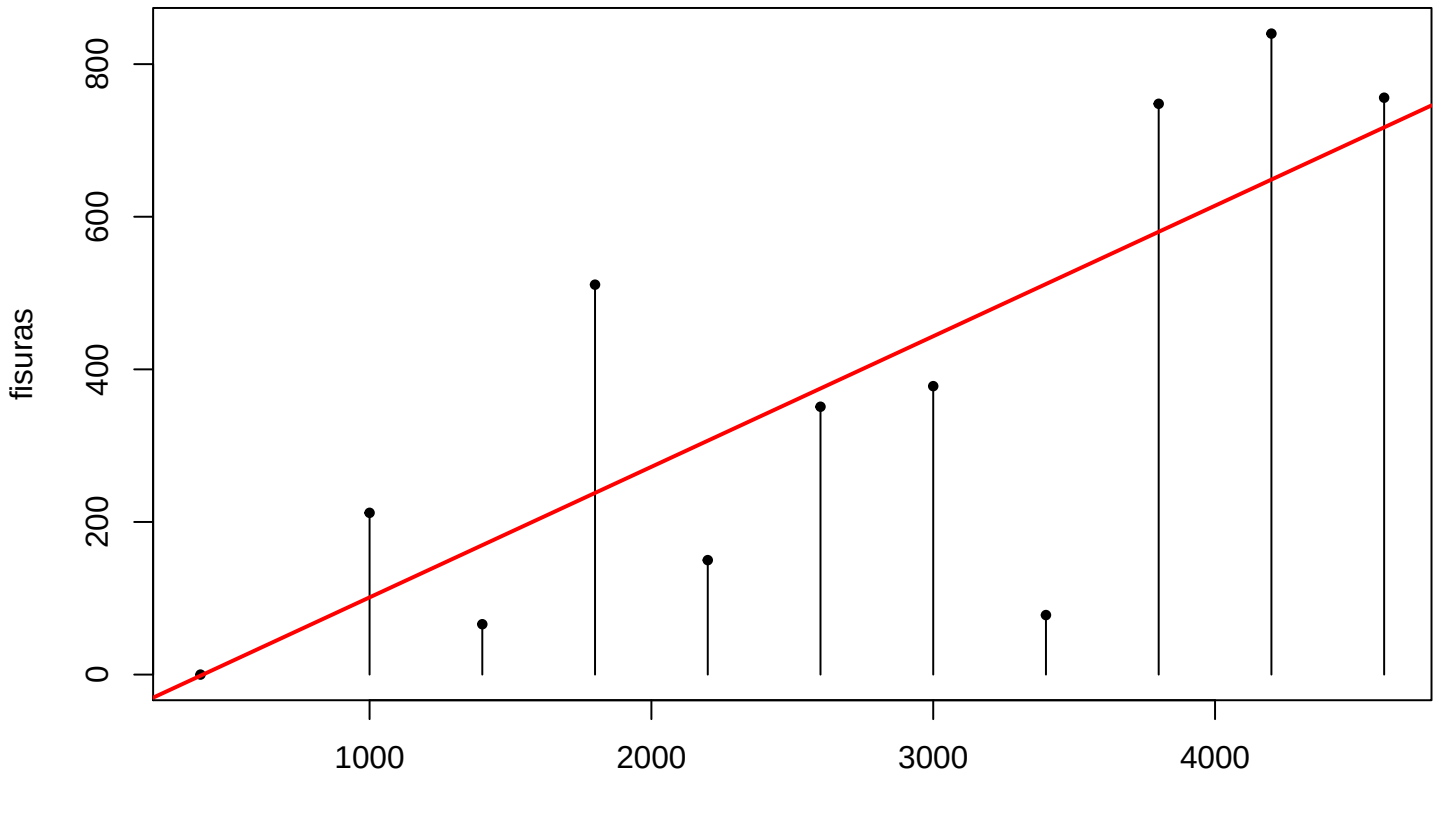
- Do you think is a good model?

```
> ex2$fitted
        1         2         3         4         5         6         7         8
 -1.47929 101.17751 169.61538 238.05325 306.49112 374.92899 443.36686 511.80473
        9        10        11
580.24260 648.68047 717.11834
```

Why GLM's?

Motivation example: Turbines experiment (cont.)

```
> plot(turbinas,type="h"); points(turbinas,cex=.6,pch=19)
> abline(ex2,col=2,lwd=2)
```



GLM's

Main Idea

- Remember in LM's we have $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2)$, where we are basically modelling the (conditional) mean, i.e.

$$\mathbb{E}(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{\mu}$$

- When $\boldsymbol{y}$ is not Normal/Gaussian, it is still good to relate the mean $\boldsymbol{\mu}$ with the linear predictor $\boldsymbol{X}\beta$.
- **Solution: Exponential families** of distributions

GLM's

Main Idea

- Remember in LM's we have $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{X}\boldsymbol{\beta}, \sigma^2)$, where we are basically modelling the (conditional) mean, i.e.

$$\mathbb{E}(\boldsymbol{y}|\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{\mu}$$

- When $\boldsymbol{y}$ is not Normal/Gaussian, it is still good to relate the mean $\boldsymbol{\mu}$ with the linear predictor $\boldsymbol{X}\beta$.
- **Solution:** **Exponential families** of distributions
- There are **two fundamental issues** in the notion of GLM's:
  1. The distribution of the response $\boldsymbol{y}$, and
  2. the model that relates the mean response to the regression variables

GLM's

Exponential families (Fisher, 1984)

- Members of the exponential family of distributions all have probability density (or probability mass) functions that can be expressed in the form

$$f(y;\boldsymbol{\theta},\phi) = exp\left\{\frac{y\boldsymbol{\theta} - b(\boldsymbol{\theta})}{a(\phi)} + c(y,\phi)\right\}$$

where, in each case, $a(.)$, $b(.)$ and $c(.)$ are specific functions. The parameter $\boldsymbol{\theta}$ is a *canonical location parameter* and ϕ is a *dispersion parameter*.

GLM's

Exponential families (Fisher, 1984)

- Members of the exponential family of distributions all have probability density (or probability mass) functions that can be expressed in the form

$$f(y;\boldsymbol{\theta},\phi) = exp\left\{\frac{y\boldsymbol{\theta} - b(\boldsymbol{\theta})}{a(\phi)} + c(y,\phi)\right\}$$

  where, in each case, $a(.)$, $b(.)$ and $c(.)$ are specific functions. The parameter $\boldsymbol{\theta}$ is a *canonical location parameter* and ϕ is a *dispersion parameter*.

- The most popular members of the EF are
  - Bernoulli, Binomial and Multinomial Distributions ($\boldsymbol{y}$ is categorical)
  - Poisson and Negative Binomial ($\boldsymbol{y}$'s are counts)
  - Normal distribution ($\boldsymbol{y}$ is continuous)
  - Exponential and Gamma distributions ($\boldsymbol{y}$ is continuous and strictly positive)
  - among many others

GLM's

Exponential families (Fisher, 1984) (cont.)

- $\mathbb{E}[\boldsymbol{y}_i] = \boldsymbol{\mu}_i$
- $\mathbb{V}ar[\boldsymbol{y}] = h(\boldsymbol{\mu}_i)\phi$, where $h(\cdot)$ is a positive function of $\boldsymbol{\mu}_i$, and $\phi > 0$ (dispersion or scale parameter)

GLM's

Exponential families (Fisher, 1984) (cont.)

- $\mathbb{E}[\boldsymbol{y}_i] = \boldsymbol{\mu}_i$
- $\mathbb{V}ar[\boldsymbol{y}] = h(\boldsymbol{\mu}_i)\phi$, where $h(\cdot)$ is a positive function of $\boldsymbol{\mu}_i$, and $\phi > 0$ (dispersion or scale parameter)

- It is important to notice the range of possible values for $\boldsymbol{\mu}_i$ for each distribution of an EF is not the same
  - Normal: $-\infty < \boldsymbol{\mu}_i < \infty$
  - Poisson: $\boldsymbol{\mu}_i > 0$
  - Bernoulli: $0 < \boldsymbol{\mu}_i < 1$

GLM's

- To understand a GLM, let us go back to the standard linear regression model

$$\mathbb{E}[\boldsymbol{y}] = \boldsymbol{\mu} = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p = \boldsymbol{X}\boldsymbol{\beta} \quad (\textit{Linear predictor or } \boldsymbol{\eta})$$

  where there is a linear relation between $\boldsymbol{X}$ and $\boldsymbol{\mu}$.
- In **LM**'s the mean $\boldsymbol{\mu}$ is directly *linked* to the linear predictor, i.e. $\boldsymbol{\mu} = \boldsymbol{\eta}$

GLM's

- To understand a GLM, let us go back to the standard linear regression model

$$\mathbb{E}[\boldsymbol{y}] = \boldsymbol{\mu} = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p = \boldsymbol{X}\boldsymbol{\beta} \quad (\textit{Linear predictor or } \boldsymbol{\eta})$$

  where there is a linear relation between $\boldsymbol{X}$ and $\boldsymbol{\mu}$.
- In **LM**'s the mean $\boldsymbol{\mu}$ is directly *linked* to the linear predictor, i.e. $\boldsymbol{\mu} = \boldsymbol{\eta}$
- **E.g:** when the response variable is **binary** (**e.g.:** Health perception data), $\boldsymbol{y} \sim$ Bernoulli(p)

$$\begin{aligned} \mathbb{P}r(y=1) &= p \\ \mathbb{P}r(y=0) &= 1-p \\ \mathbb{E}[y] &= 0 \times \mathbb{P}r(y=0) + 1 \times \mathbb{P}r(y=1) = p \end{aligned}$$

$$\Downarrow$$

Linear relationship between X and $p \rightarrow$ gives wrong or misleading results

GLM's

- To understand a GLM, let us go back to the standard linear regression model

$$\mathbb{E}[\boldsymbol{y}] = \boldsymbol{\mu} = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p = \boldsymbol{X}\boldsymbol{\beta} \quad (\textit{Linear predictor or } \boldsymbol{\eta})$$

  where there is a linear relation between $\boldsymbol{X}$ and $\boldsymbol{\mu}$.
- In **LM**'s the mean $\boldsymbol{\mu}$ is directly *linked* to the linear predictor, i.e. $\boldsymbol{\mu} = \boldsymbol{\eta}$
- **E.g:** when the response variable is **binary** (**e.g.:** Health perception data), $\boldsymbol{y} \sim$ Bernoulli(p)

$$\begin{aligned} \mathbb{P}r(y=1) &= p \\ \mathbb{P}r(y=0) &= 1-p \\ \mathbb{E}[y] &= 0 \times \mathbb{P}r(y=0) + 1 \times \mathbb{P}r(y=1) = p \end{aligned}$$

$$\Downarrow$$

  Linear relationship between X and $p \rightarrow$ gives wrong or misleading results
- **Solution:** **Link functions**

GLM's

A GLM:

- **DO NOT** establish a linear relationship between $\boldsymbol{X}$ and $\boldsymbol{\mu}$.

GLM's

A GLM:

- **DO NOT** establish a linear relationship between $\boldsymbol{X}$ and $\boldsymbol{\mu}$.
- **DO** establish a linear relationship between $\boldsymbol{X}$ and $\boldsymbol{\mu}$ through **link functions** $g(\cdot)$

$$g(\mathbb{E}[\boldsymbol{y}]) = \beta_0 + \beta_1 x_1 + \ldots + x_p \beta_p = \boldsymbol{X}\boldsymbol{\beta},$$

  depending on $\boldsymbol{y}$, we have different choices for $g(\cdot)$.
- g is a monotonic invertible function that maps the value of $\boldsymbol{\mu}$ onto $(-\infty, \infty)$
- g^{-1} is now as the **inverse link function**
- The link function g is also called **canonical link** o natural link: that transforms the mean to the location parameter

$$\boldsymbol{\eta} = g(\boldsymbol{\mu}) = \boldsymbol{\theta} \Rightarrow g \quad \text{is a canonical link}$$

* We will choose it by default in our examples

Canonical links in GLM's

Exponential Family	Canonical Link
Normal	$\boldsymbol{X\beta} = \boldsymbol{\mu}$ (identity)
Binomial	$\boldsymbol{X\beta} = \ln\left(\frac{p}{1-p}\right)$ (logistic)
Poisson	$\boldsymbol{X\beta} = \ln(\boldsymbol{\mu})$ (log link)
Exponential	$\boldsymbol{X\beta} = \frac{1}{\boldsymbol{\mu}}$ (reciprocal)
Gamma	$\boldsymbol{X}\beta = \frac{1}{\boldsymbol{\mu}}$ (reciprocal)

- We can view the selection of the link function similar as the choice of a transformation on the response.
- The link function is a transformation on the *population mean* not the data.
- More details in Mc Cullagh and Nelder (1989, Chapter 2)

Estimation and interpretation of $\boldsymbol{\beta}$

- For Normal data we used MLE to obtain $\hat{\boldsymbol{\beta}}_{MLE} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$
- In general, for GLM's MLE is not useful to obtain $\hat{\boldsymbol{\beta}}_{MLE}$

Estimation and interpretation of $\boldsymbol{\beta}$

- For Normal data we used MLE to obtain $\hat{\boldsymbol{\beta}}_{MLE} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$
- In general, for GLM's MLE is not useful to obtain $\hat{\boldsymbol{\beta}}_{MLE}$
- We need to be approximate $\hat{\boldsymbol{\beta}}$ using particular algorithms until convergence

  e.g: Iterative Reweighted Least Squares based on Newton-Raphson method (details skipped)

Estimation and interpretation of $\boldsymbol{\beta}$

- For Normal data we used MLE to obtain $\hat{\boldsymbol{\beta}}_{MLE} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$
- In general, for GLM's MLE is not useful to obtain $\hat{\boldsymbol{\beta}}_{MLE}$
- We need to be approximate $\hat{\boldsymbol{\beta}}$ using particular algorithms until convergence

  e.g: Iterative Reweighted Least Squares based on Newton-Raphson method (details skipped)

- The interpretation of $\boldsymbol{\beta}$ depends on the link function g.
- In LM $\boldsymbol{\beta}$ is the effect of a unit change in $\boldsymbol{X}$ on $\boldsymbol{y}$
- In GLM's $\boldsymbol{\beta}$'s are interpretted as a unit change in x_i on $g(\mu_i)$
- $g(\mu_i)$ is not on the same scale y_i, instead μ_i is

  * We will discuss this in the data examples.

GLM's

Summary

Components of a GLM

1. **Random component:** $\boldsymbol{y}$ is a vector of random iid components acording to a specific exponential family distribution with mean $\boldsymbol{\mu}$.
2. **Systematic component:** is the linear predictor $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$. This describes how the location of the response distribution changes with the predictors x_i $i = 1 \dots p$
3. **Link function:** is a monotonic differentiable function which links the mean and the linear predictor

$$\boldsymbol{\eta} = g(\boldsymbol{\mu}) \qquad \mathbb{E}[\boldsymbol{y}] = \boldsymbol{\mu} = g^{-1}(\boldsymbol{\eta})$$

References

Berkson, J. (1944). *Application of the logistic function to bio-assay*. *Journal of the American Statistical Association*, Vol. 39, No. 227) 39 (227): 357???65.

Faraway, J. (2002). *Practical regression and ANOVA using R*. http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf

Faraway, J. (2004). *Linear Models with R*. Chapman Hall/CRC Texts in Statistical Science

Fisher, R. (1934). *The new properties of mathematical likelihood*. *Proceedings of the Royal Statistical Society of London, A*, 144:285–307.

McCullagh, P. and Nelder, J. (1989). *Generalized Linear Models*. Chapman & Hall, New York.

Nelson, W. (1982). *Applied Life Data Analysis*. New York: John Wiley & Sons.

bcam basque center for applied mathematics