

# MAD-CB



## Regressão Linear – 2

# Revisão de Regressão Linear Simples

- Vamos seguir a construção e interpretação de um modelo de regressão simples do início até o fim
- Mostra processo de preparação e análise dos dados

# Processo - 5 Passos

- 1 Preparar os Dados
- 2 Explorar os Dados
- 3 Construir Modelo
- 4 Interpretar Resultados
- 5 Apurar Modelo

# Dataset - Women

- Vem de programa R como um exemplo

```
##      height weight
## 1         58    115
## 2         59    117
## 3         60    120
## 4         61    123
## 5         62    126
## 6         63    129
## 7         64    132
## 8         65    135
## 9         66    139
## 10        67    142
## 11        68    146
## 12        69    150
## 13        70    154
## 14        71    159
```

# Passo 1 – Traduzir a Idioma e as Medidas

- Abrir o dataset e olhar a estrutura

```
str(women) #mostrar a estrutura de data.frame
```

```
## 'data.frame':    15 obs. of  2 variables:  
## $ height: num  58 59 60 61 62 63 64 65 66 67 ...  
## $ weight: num  115 117 120 123 126 129 132 135 139 142 ...
```

# Comanda transmute

- `transmute` cria novas variáveis e apaga as originais.
  - ▶ `mutate` preserva as originais
- Colocar os resultados num dataframe novo - `mulheres`.

```
mulheres <- transmute(women,  
  alturacm = round(height*2.54, 2),  
  pesokg = round(weight/2.2, 2))
```

# Dados - mulheres

##		alturacm	pesokg
## 1		147.32	52.27
## 2		149.86	53.18
## 3		152.40	54.55
## 4		154.94	55.91
## 5		157.48	57.27
## 6		160.02	58.64
## 7		162.56	60.00
## 8		165.10	61.36
## 9		167.64	63.18
## 10		170.18	64.55
## 11		172.72	66.36
## 12		175.26	68.18
## 13		177.80	70.00
## 14		180.34	72.27
## 15		182.88	74.55



# Estrutura de mulheres

```
str(mulheres)
```

```
## 'data.frame':    15 obs. of  2 variables:  
## $ alturacm: num  147 150 152 155 157 ...  
## $ pesokg : num  52.3 53.2 54.5 55.9 57.3 ...
```

```
str(women)
```

```
## 'data.frame':    15 obs. of  2 variables:  
## $ height: num  58 59 60 61 62 63 64 65 66 67 ...  
## $ weight: num  115 117 120 123 126 129 132 135 139 142 ...
```

## Passo 2 - Analise Exploratória

- Utilize pacote DescTools - analise mais completa das variáveis com gráficos

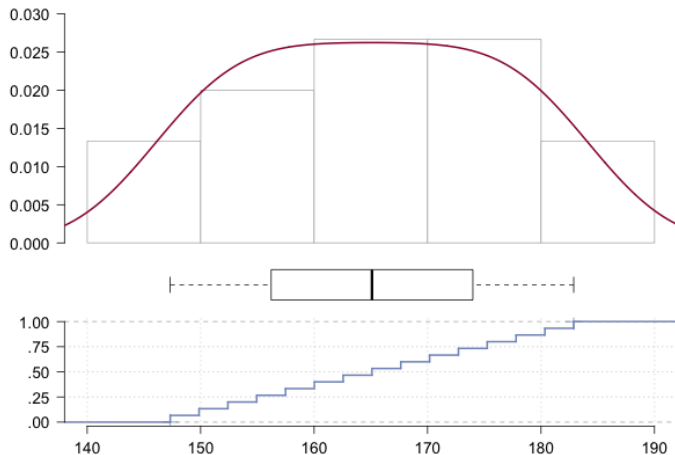
# Desc – altura

```
Desc(mulheres$alturacm, plotit = FALSE)
```

```
## -----
## mulheres$alturacm (numeric)
##
##      length      n      NAs  unique      Os      mean  meanCI
##         15      15       0      = n      0    165.100  158.809
##          100.0%    0.0%          0.0%          171.391
##
##      .05      .10      .25  median      .75      .90      .95
## 149.098 150.876 156.210 165.100 173.990 179.324 181.102
##
##      range      sd      vcoef      mad      IQR      skew      kurt
##   35.560   11.359   0.069   15.063   17.780 1.696e-15  -1.441
##
## lowest : 147.32, 149.86, 152.4, 154.94, 157.48
## highest: 172.72, 175.26, 177.8, 180.34, 182.88
```

```
alturacm com plotit = TRUE
```

**mulheres\$altura (numeric)**

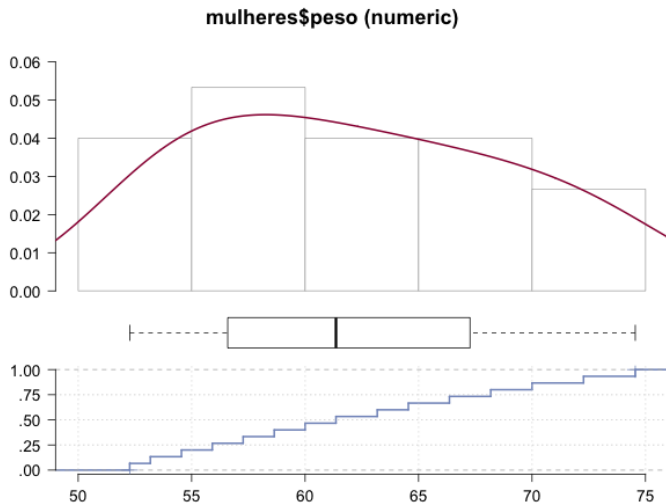


# Desc – peso

```
Desc(mulheres$pesokg, plotit = FALSE)
```

```
## -----  
## mulheres$pesokg (numeric)  
##  
##      length      n      NAs  unique      0s      mean  meanCI  
##          15      15        0      = n      0  62.151  58.250  
##          100.0%    0.0%          0.0%          66.053  
##  
##      .05      .10      .25  median      .75      .90      .95  
##  52.907  53.728  56.590  61.360  67.270  71.362  72.954  
##  
##      range      sd    vcoef      mad      IQR      skew      kurt  
##   22.280    7.045    0.113    8.080   10.680    0.228   -1.344  
##  
## lowest : 52.27, 53.18, 54.55, 55.91, 57.27  
## highest: 66.36, 68.18, 70.0, 72.27, 74.55
```

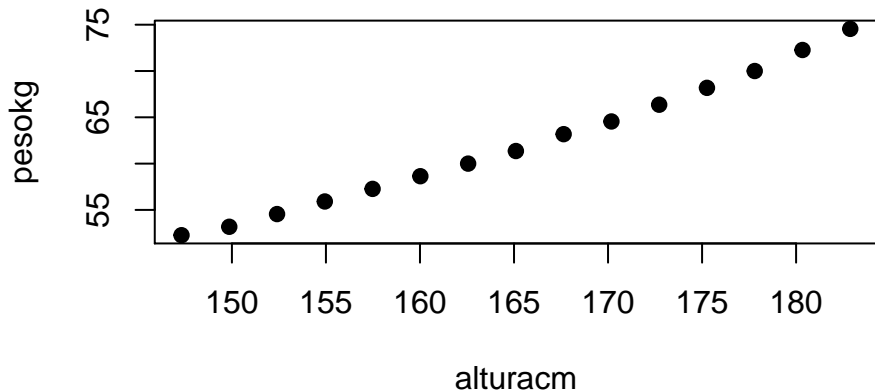
pesokg com plotit = TRUE



# Plotagem de Dispersão (*Scatterplot*)

```
plot(pesokg ~ alturacm, data = mulheres, pch = 19,  
     main = "Mulheres Idade 30 - 39")
```

## Mulheres Idade 30 - 39



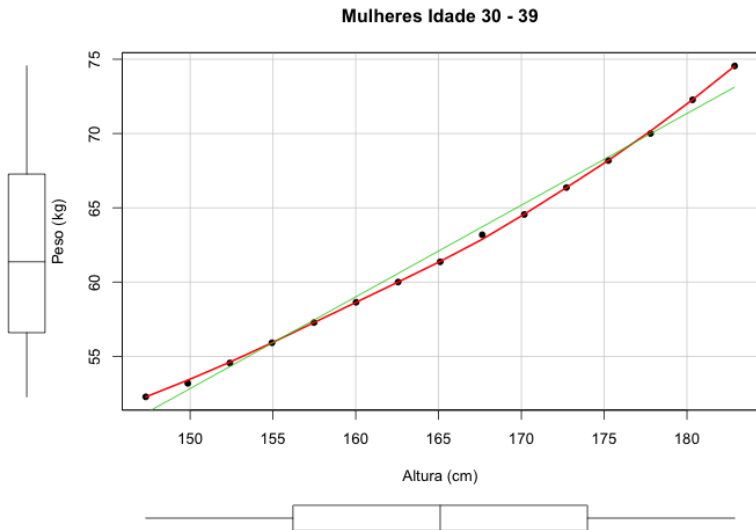
# Scatterplot Avançado (de pacote car)

```
library(car)
scatterplot(pesokg ~ alturacm,
            data=mulheres,
            spread=FALSE, smooth = TRUE,
            pch=19,
            main="Mulheres Idade 30 - 39",
            xlab="Altura (cm)",
            ylab="Peso (kg)")
```

- car tem uma variedade das funções que ajuda com a regressão



# Scatterplot



## Passo 3 – Construir o Modelo

```
fit <- lm(pesokg ~ alturacm, data = mulheres)
summary(fit)
```

# Resumo do Modelo

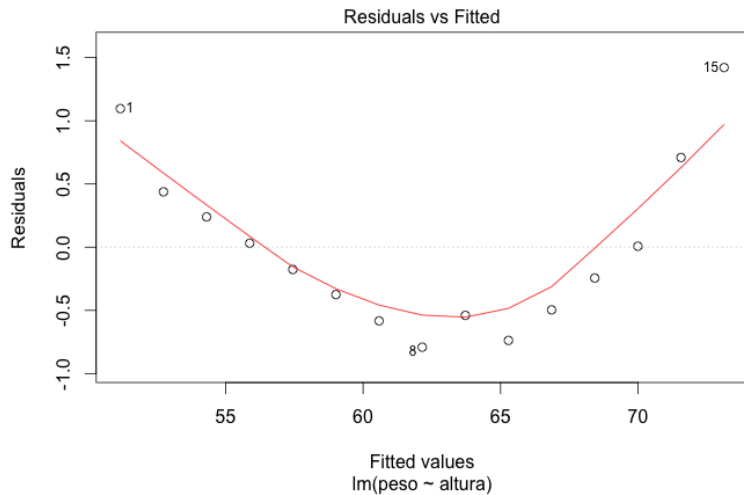
```
##
## Call:
## lm(formula = pesokg ~ alturacm, data = mulheres)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.7913 -0.5178 -0.1767  0.3388  1.4212
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -39.78260     2.70019  -14.73 1.72e-09 ***
## alturacm      0.61741     0.01632   37.83 1.10e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6936 on 13 degrees of freedom
## Multiple R-squared:  0.991, Adjusted R-squared:  0.9903
## F-statistic: 1431 on 1 and 13 DF, p-value: 1.099e-14
```

# Tabela ANOVA

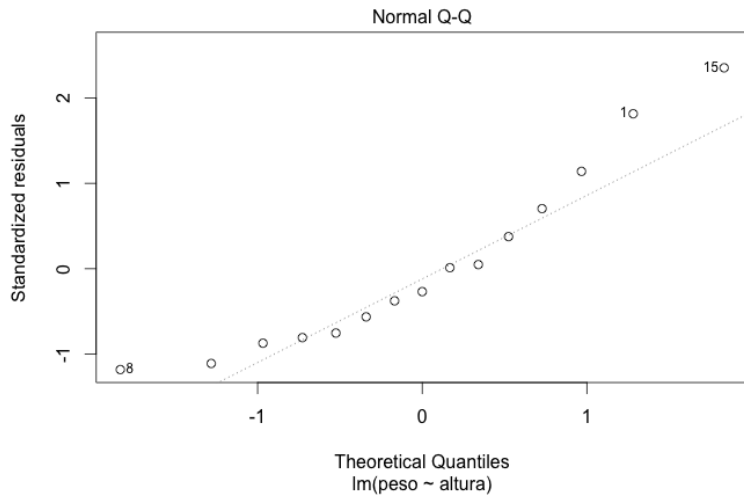
```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: pesokg
##           Df Sum Sq Mean Sq F value    Pr(>F)
## alturacm   1 688.60   688.60  1431.4 1.099e-14 ***
## Residuals 13   6.25     0.48
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

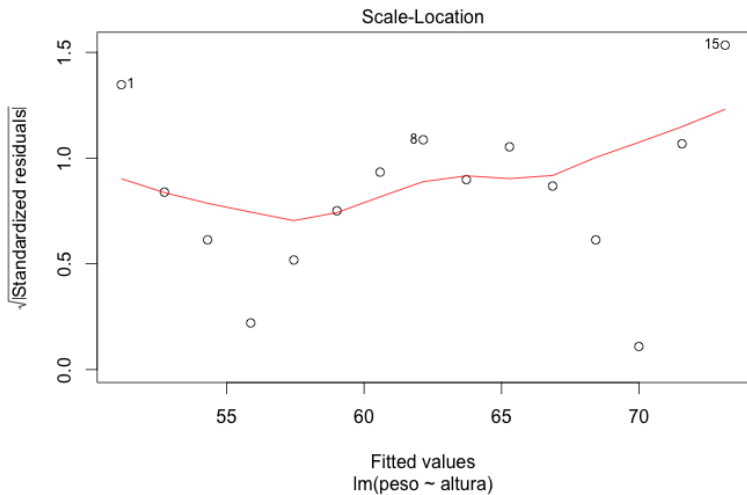
# Plotagem de fit – 4 Gráficos



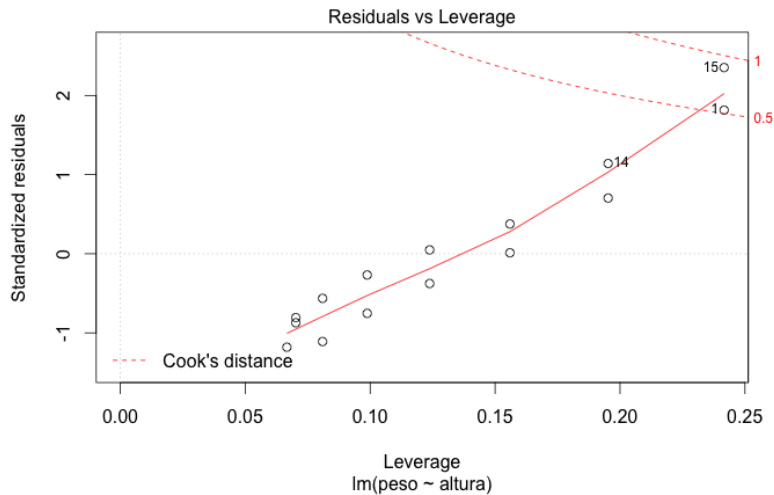
# Q-Q Normal Gráfico



# Scale - Location



# Outliers - Alavancagem

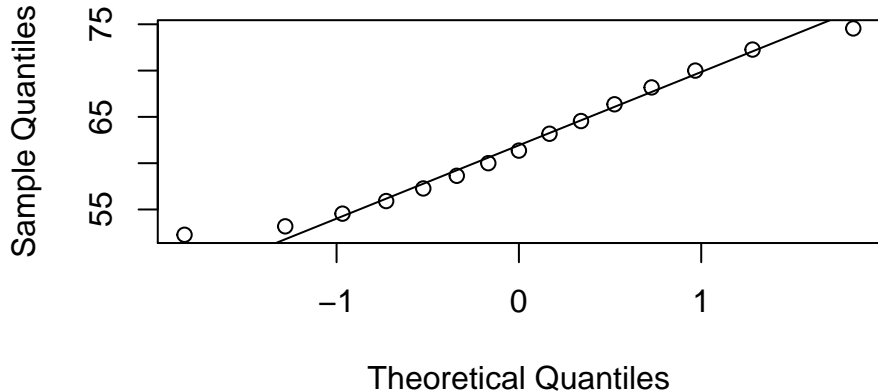




# Duas Outras Versões de Q-Q Normal – de R Base

```
qqnorm(mulheres$pesokg)  
qqline(mulheres$pesokg)
```

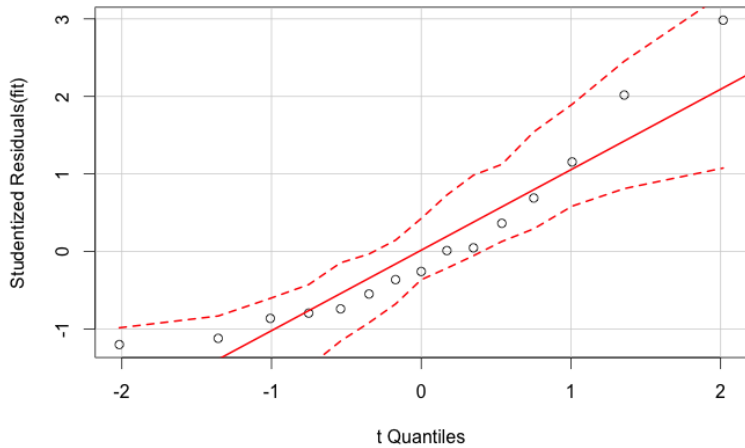
## Normal Q–Q Plot



## Do Pacote car – qqPlot(fit)

```
library(car)  
qqPlot(fit, main = "Q-Q Normal")
```

### Q-Q Normal



# Regressão Polinomial

- Solução potencial para o problema de forma quadrática
- Usa a forma de  $y = x + x^2$
- Parecido com a álgebra que aprendemos no ensino médio
- em R, precisamos usar a função  $I(x^2)$  porque  $\hat{\phantom{x}}$  tem significado nas formulas
- $I(x^2)$  quer dizer “não interpreta o  $\hat{\phantom{x}}$  como parte da formula. Deixe ele ser tratado com o significado normal”

## -Código em R

```
fitpoly <- lm(pesokg ~ alturacm + I(alturacm^2), data = mulheres)
summary(fitpoly)
anova(fitpoly)
par(mfrow=c(2,2))
plot(fitpoly)
par(mfrow=c(1,1))
qqPlot(fitpoly,
       main = "Q-Q Normal Regressão Polinomial")
```

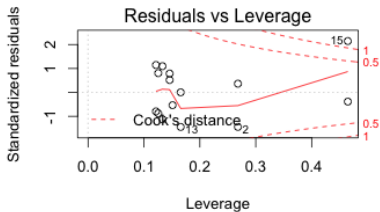
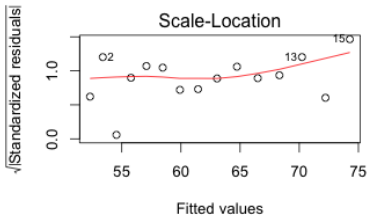
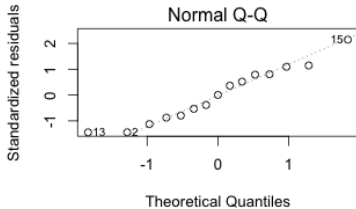
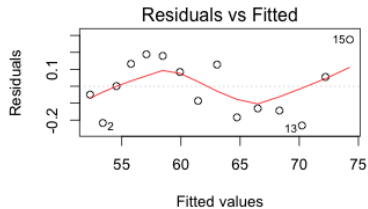
# Resumo fitpoly

```
##
## Call:
## lm(formula = pesokg ~ alturacm + I(alturacm^2), data = mulheres)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.231604 -0.137179  0.000539  0.129965  0.275529
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept)  119.0833006   11.5095658   10.346 0.00000024757 ***
## alturacm     -1.3156140    0.1398583   -9.407 0.00000069060 ***
## I(alturacm^2)  0.0058541    0.0004234   13.827 0.00000000982 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1754 on 12 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9994
## F-statistic: 1.128e+04 on 2 and 12 DF,  p-value: < 2.2e-16
```

```
## Analysis of Variance Table
##
## Response: pesokg
##
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
alturacm	1	688.60	688.60	22373.36	< 2.2e-16 ***
I(alturacm^2)	1	5.88	5.88	191.19	0.000000009822 ***
Residuals	12	0.37	0.03		

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```





## Q-Q Normal Regressão Polinomial

