

Introdução ao R

Anderson Castro Soares de Oliveira

Regressão e Correlação

- Correlação permite verificar o grau de associação entre duas variáveis

Regressão e Correlação

- Correlação permite verificar o grau de associação entre duas variáveis

Regressão e Correlação

- Correlação permite verificar o grau de associação entre duas variáveis
- Correlação de Pearson dados normais

Régressão e Correlação

- Correlação permite verificar o grau de associação entre duas variáveis
- Correlação de Pearson dados normais
- Correlação de Kendall e Spearman dados não normais.

```
cor.test(x, y,  
alternative = c("two.sided", "less", "greater"),  
conf.level = 0.95,),  
method = c("pearson", "kendall", "spearman")
```

Regressão e Correlação

```
> ##Correlação entre Peso e Altura  
> cor.test(P,A)
```

Pearson's product-moment correlation

data: P and A

t = 3.3135, df = 27, p-value = 0.002629

alternative hypothesis: true correlation is not equal to zero

95 percent confidence interval:

0.2131712 0.7553306

sample estimates:

cor

0.5376696

Régressão e Correlação

```
> ##Correlação entre Peso e Idade  
> cor.test(P, IDADE, method = "spearman")
```

Spearman's rank correlation rho

data: P and IDADE

S = 4015.886, p-value = 0.9554

alternative hypothesis: true rho is not equal to 0
sample estimates:

rho

0.01086558

Régressão e Correlação

- Régressão Linear Simples

$$y = a + bx + \epsilon$$

em que:

Régressão e Correlação

- Régressão Linear Simples

$$y = a + bx + \epsilon$$

em que:

- a é coeficiente linear, interpretado como o valor da variável de dependente quando a variável indenpente é igual a 0;

Regressão e Correlação

- Regressão Linear Simples

$$y = a + bx + \epsilon$$

em que:

- a é coeficiente linear, interpretado como o valor da variável de dependente quando a variável independente é igual a 0;
- b é coeficiente de regressão, interpretado como acréscimo na variável dependente para a variação de uma unidade na variável.

Régressão e Correlação

- Régressão Linear Simples

$$y = a + bx + \epsilon$$

em que:

- a é coeficiente linear, interpretado como o valor da variável de dependente quando a variável independente é igual a 0;
- b é coeficiente de regressão, interpretado como acréscimo na variável dependente para a variação de uma unidade na variável.
- ϵ são os erros aleatórios de uma população normal, com média 0 e variância constante.

Regressão e Correlação

```
> ##modelo de regressão entre Peso e Altura  
> model=lm (P~A)  
> model
```

Call:

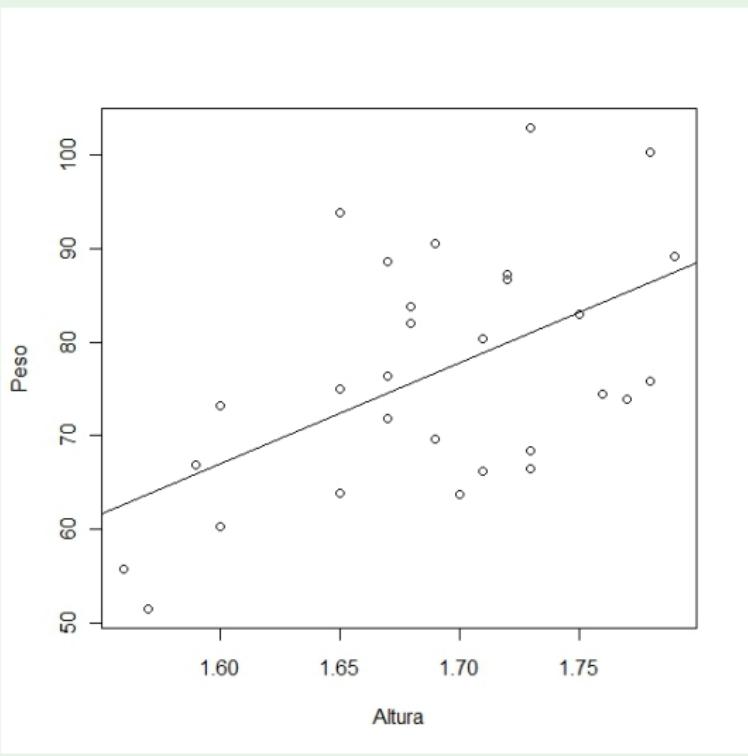
```
lm(formula = P ~ A)
```

Coefficients:

(Intercept)	A
-106.0	108.1

Regressão e Correlação

```
> plot (A,P,xlab="Altura",ylab="Peso", )  
> abline (model)
```



Rregressão e Correlação

```
#modelo sem intercepto  
> model1=lm(P~A-1)  
> summary(model1)
```

Call:

```
lm(formula = P ~ A - 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-19.8057	-10.1725	0.0609	7.9025	24.3275

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
A	45.418	1.257	36.14	<2e-16 ***

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

Residual standard error: 11.44 on 28 degrees of freedom

Multiple R-squared: 0.979, Adjusted R-squared: 0.9783

F-statistic: 1306 on 1 and 28 DF, p-value: < 2.2e-16

Regressão e Correlação

```
> ##análise de variância para o modelo ajustado  
> anova(model)
```

Analysis of Variance Table

Response: P

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
A	1	1311.3	1311.33	10.979	0.002629 **
Residuals	27	3224.7	119.43		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'

Regressão e Correlação

```
> ##gerar os resíduos  
> erro=resid(model1)  
> ##teste de normalidade  
> shapiro.test(erro)
```

Shapiro-Wilk normality test

```
data: erro  
W = 0.9679, p-value = 0.5055
```

Régressão e Correlação

```
> ##calcula a previsão e seu intervalo de confiança  
> prev=predict(model,interval = "confidence")  
> plot(A,P,xlab="Altura",ylab="Peso")  
> previsao=cbind(A,prev) ##adicionar altura ao conjunto  
> previsao=previsao[order(previsao[,1]),] ##ordenar os dados p  
> lines(previsao[,1:2]) ##plotar a previsão  
> lines(previsao[,1],previsao[,3],lty=2) ##plotar limite infer  
> lines(previsao[,1],previsao[,4],lty=2) ##plotar limite super
```