


IMD0033 - Probabilidade

Aula 16 - Correlação e covariância

Ivanovitch Silva
Outubro, 2017



Agenda

- Correlação e covariância
- Coeficiente de correlação

Atualizar o repositório

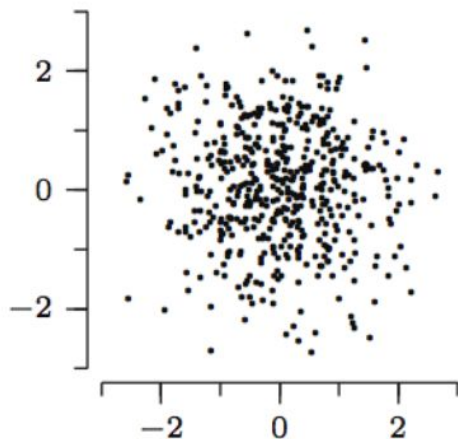
```
git clone https://github.com/ivanovitchm/IMD0033_Probabilidade.git
```

Ou

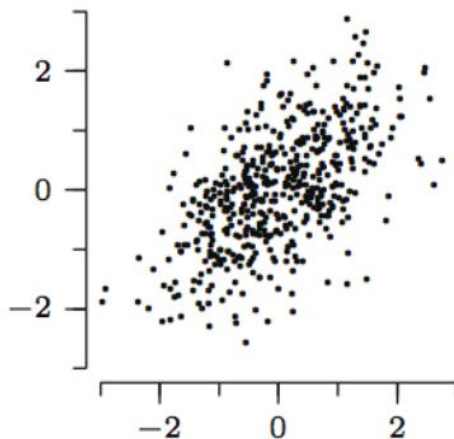
```
git pull
```

Tipos de correlação e intensidade

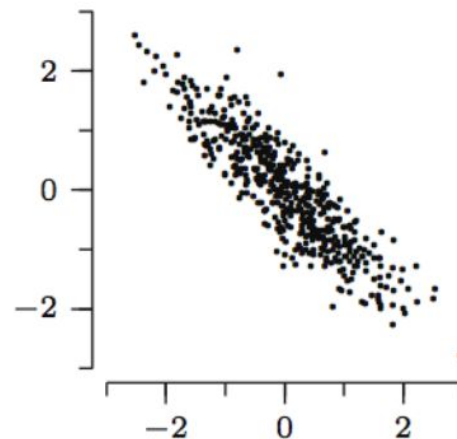
Gráficos de Dispersão



Não correlacionado

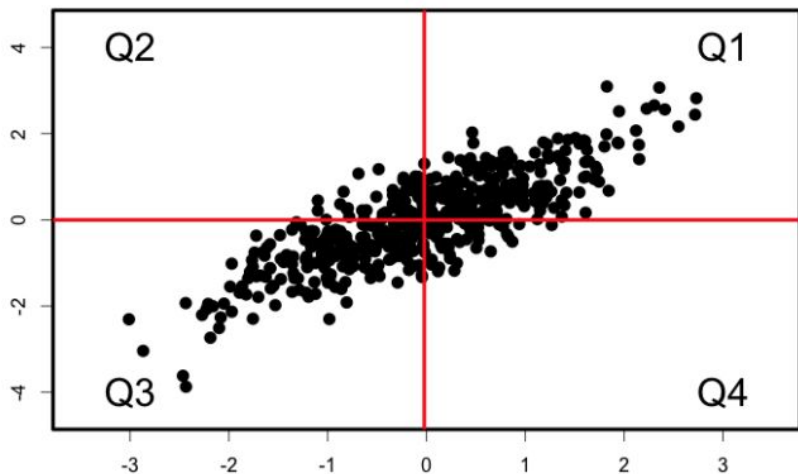


Correlação positiva, fraca



Correlação negativa, forte

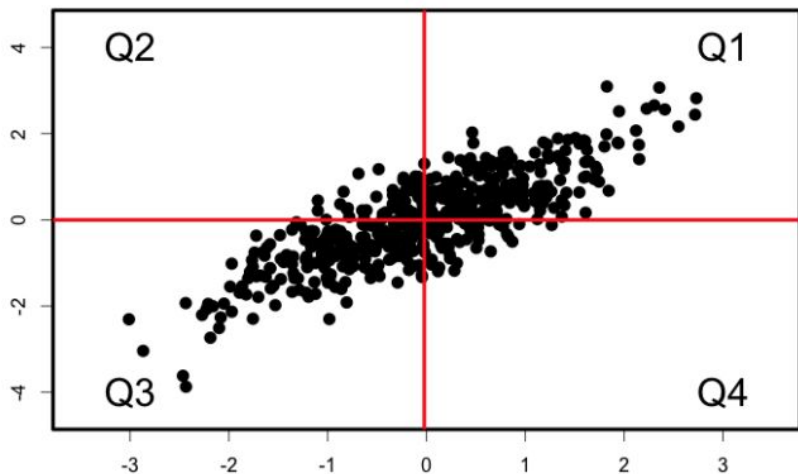
Analizando os gráficos de dispersão



É possível incluir no gráfico de dispersão as retas vertical e horizontal que passam, respectivamente, pelas médias amostrais \bar{x} e \bar{y}

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{e} \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

Analizando os gráficos de dispersão



$(x_i - \bar{x})$ Desvio da média para a variável x

$(y_i - \bar{y})$ Desvio da média para a variável y

$(x_i - \bar{x})(y_i - \bar{y})$ Produto dos desvios

Como se comporta o produto dos desvios para Q1, Q2, Q3, Q4?

Covariância

Covariância é uma medida que reflete a forma como duas variáveis variam conjuntamente.

$$\text{cov}(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)}{n}$$

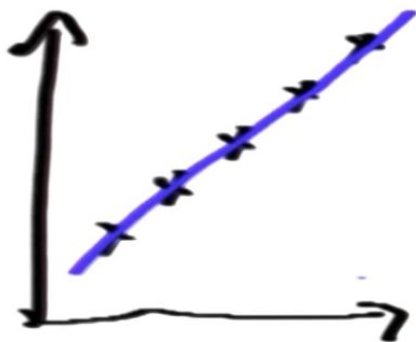
Coeficiente de correlação

A covariância, no entanto, não fornece uma medida da intensidade da relação, já que depende das unidades em que as variáveis são expressas. Uma maneira de contornar este problema é através da padronização dos dados

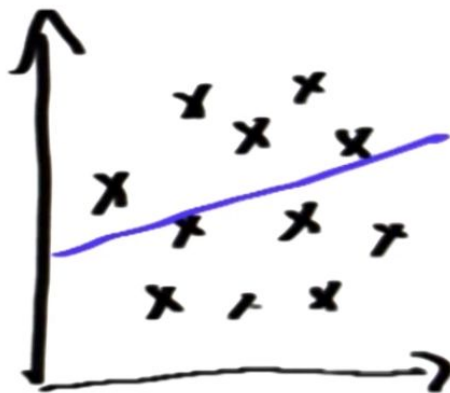
$$\frac{cov(\mathbf{x}, \mathbf{y})}{\sigma_x \sigma_y}$$

Coeficiente de correlação

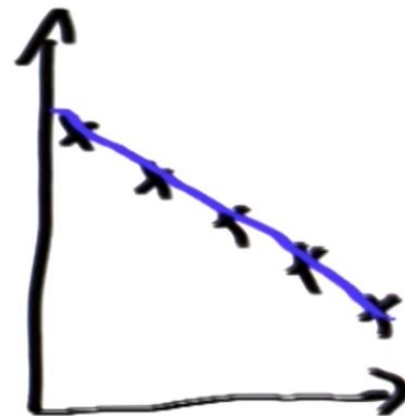
$\in [-1 \dots 1]$



$\gamma = 1$



0



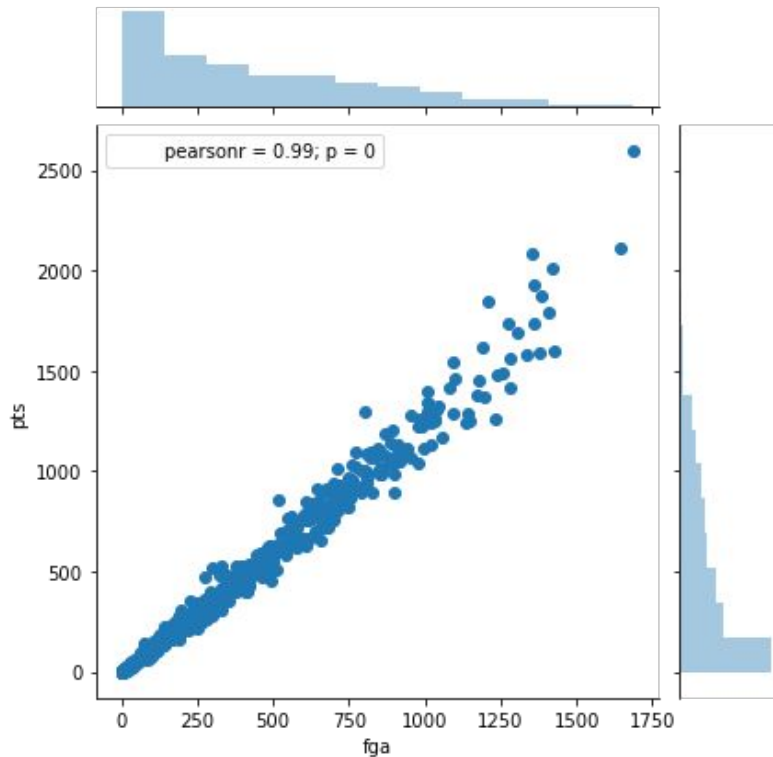
-1

Pearsonr

```
from scipy.stats.stats import pearsonr
```

```
# The pearsonr function will find the correlation between two columns of data.  
# It returns the r value and the p value. We'll learn more about p values later on.  
r, p_value = pearsonr(nba["fga"], nba["pts"])
```

Seaborn Jointplot



```
# Draw a plot of two variables  
# with bivariate and univariate graphs.  
sns.jointplot(x=nba["fga"], y=nba["pts"])
```

