

# Projeto 1 EDA - Pisa

*Fabio Carvalho Lima*

*10/11/2019*

Exploratory Data Analysis (**EDA**) - é o processo de analisar e visualizar o dado para ganhar um melhor entendimento do dado e insights. Há vários estágios que envolvem o processo de EDA, mas o mais comuns que são executados por um analista de dados são:

1. Importar o dado.
2. Limpeza do dado.
3. Processamento e organização.
4. Visualização do dado.

Para este primeiro projeto de EDA, iremos executar todos os processos descritos anteriormente e para ser capaz de fazer todos os passos, iremos usar algumas ferramentas do R, tais como:

1. biblioteca - Tidyverse para tornar o dado em formato tidy - data frames ou tibbles.
2. Algumas funções básicas para manipular os dados como , strsplit (), cbind(), matrix (), dentre outras.
3. biblioteca - corrrplot para fazer plots de correlação.

O conjunto de dados que será usado será o Pisa (*Programa Internacional de Avaliação de Alunos*) é uma rede mundial de desempenho escolar, realizado pela primeira vez em 2000 e repetido a cada 3 anos. Usaremos este dataset para fazer o EDA. Utilizaremos os dados do ano 2013-2015.

Para executar os passos 1-3 utilizaremos os seguintes passos:

## Importação, limpeza e organização dos dados.

```
dataframe.raw <- read.csv(file = "../data/Pisa_MeanPerformance_2013_2015.csv", fileEncoding = "UTF-8-BOM")
head(dataframe.raw)
```

```
## Country.Name Country.Code
## 1 Albania ALB
## 2 Albania ALB
## 3 Albania ALB
## 4 Albania ALB
## 5 Albania ALB
## 6 Albania ALB
##
## Series.Name Series.Code
## 1 PISA: Mean performance on the mathematics scale LO.PISA.MAT
## 2 PISA: Mean performance on the mathematics scale. Female LO.PISA.MAT.FE
## 3 PISA: Mean performance on the mathematics scale. Male LO.PISA.MAT.MA
## 4 PISA: Mean performance on the reading scale LO.PISA.REA
## 5 PISA: Mean performance on the reading scale. Female LO.PISA.REA.FE
## 6 PISA: Mean performance on the reading scale. Male LO.PISA.REA.MA
## X2013..YR2013. X2014..YR2014. X2015..YR2015.
## 1 NA NA 413.2
## 2 NA NA 417.8
## 3 NA NA 408.5
## 4 NA NA 405.3
## 5 NA NA 434.6
## 6 NA NA 375.8
```

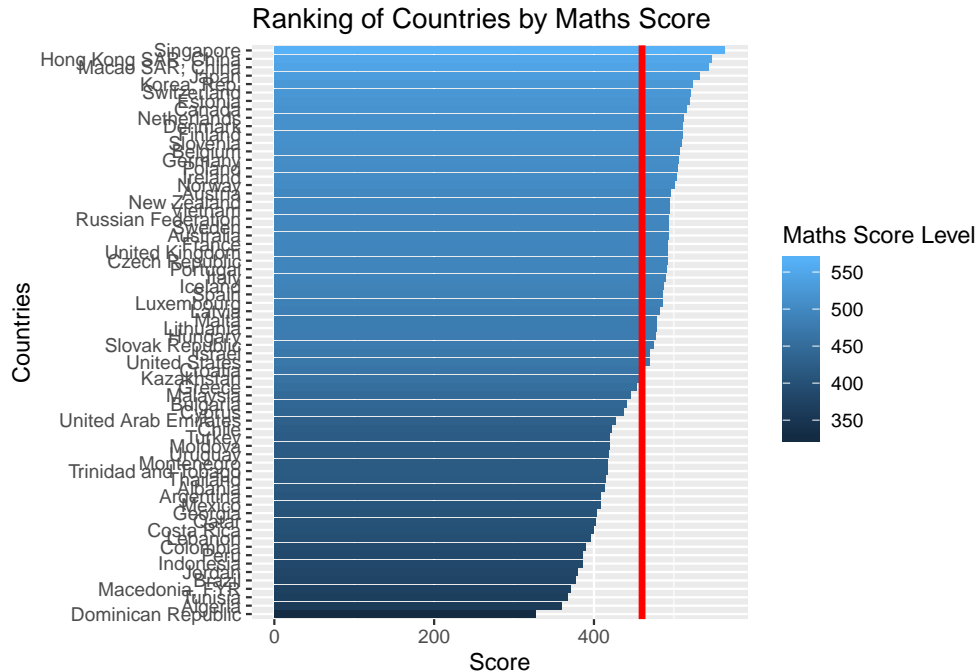


Figure 1: Notas de matemática

Para limpar o conjunto de dados precisamos organizar as informações, o termo específico para esse processo é “data wrangling”. Ou para ser mais específico deixar os *dados Tidy* (dados arrumados). As ferramentas que utilizaremos para limpar e organizar os dados pertencem ao **tidyr**, um pacote que fornece diversas ferramentas para ajudar-nos a arrumar os dados bagunçados. O **tidyr** é um membro do núcleo do tidyverse.

- Cada coluna no dataset deve corresponder a um único país. Podemos ver na variável **\$ Country.Name** que estão nas linhas. Para executar esse passo devemos utilizar a função `spread(key = Series.Code, value = X2015..YR2015.)`
- Por inspeção do dado importado vemos que há inúmeras colunas com todas as observações sem nenhuma informação *NAs*. Aqui manteremos as colunas e linhas com informação relevante apenas, usaremos a função `drop_na()` - eliminar *NAs* e faremos ao mesmo tempo um subset dos dados.
- Renomear a coluna **Series Code** para um nome de melhor entendimento através da função `rename()`.

Podemos fazer todas operações anteriormente descritas encadeadas com o uso do pipe.

Estando os dados em formato tidy, agora podemos partir para a visualização dos dados.

## Visualização

- Barplot - Ranking das notas de matemática por países
  - Ranking das notas de matemática:
  - Ranking das notas de ciências:
  - Ranking das notas de leitura:

```
pisa2015 %>%
  ggplot(aes(x = reorder(Country.Name, Reading), y = Reading)) +
  geom_bar(stat = 'identity', aes(fill = Reading)) +
  coord_flip() +
```

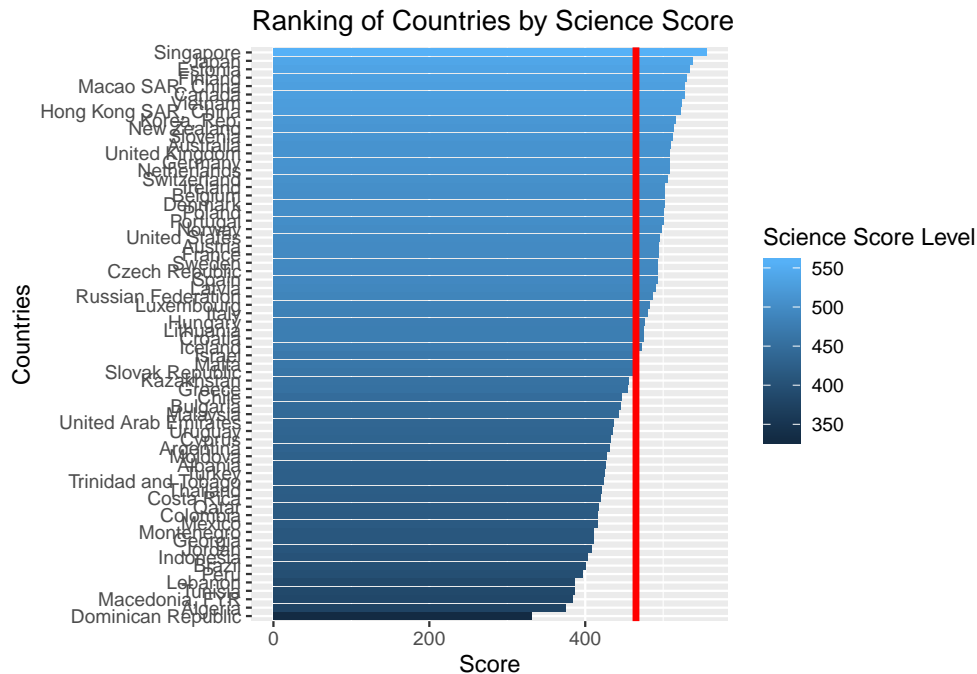


Figure 2: Notas de ciências

```
scale_fill_gradient(name = "Reading Score Level") +
labs(title = 'Ranking of Countries by Reading Score',
y = 'Score', x = 'Countries') +
geom_hline(yintercept = mean(pisa2015$Reading),size = 1.5, color = 'red') +
theme_gray()
```

Para criar boxplots precisaremos trabalhar o data frame **pisa2015**, ele está em formato que nos impede de plotar as informações em um boxplot, que pede como entrada um data frame, com uma coluna x e outra y.

## 2. Boxplots

No gráfico acima podemos separar as disciplinas e os gêneros, para isso vamos usar a função **strsplit()** - separa os elementos de vetor de caracter x, em substrings que separadas de acordo com um separador.

```
S <- numeric(408)      # create an empty vector
for (i in 1:length(pisaLong$Score)) {
  S[i] <- strsplit(pisaLong$Score[i], ".", fixed = TRUE)
}
```

Agora temos uma lista com 408 componentes, cada um contém 2 sub-componentes, “disciplina: Science” e “Gender”, vamos chamar esse data frame de **df\_S**.

Agora podemos combinar os data frames **pisaLong** e **df\_S** e nomear o resultado como **pisaWide**, usando a função **cbind()** que funciona para vetores, matrizes ou data frames.

Agora temos um data frame mais organizado e informativo. Agora podemos criar múltiplos gráficos, utilizando a função **face\_wrap()** ou **face\_grid**.

Vamos gerar um outro gráfico criando as facetas por teste (Math, Science and Reading).

Olhando os gráficos acima, já podemos ter algum insight sobre como os homens e mulheres que participaram desta avaliação, se saíram nas provas, do boxplot acima vemos que homens só saíram melhores com uma

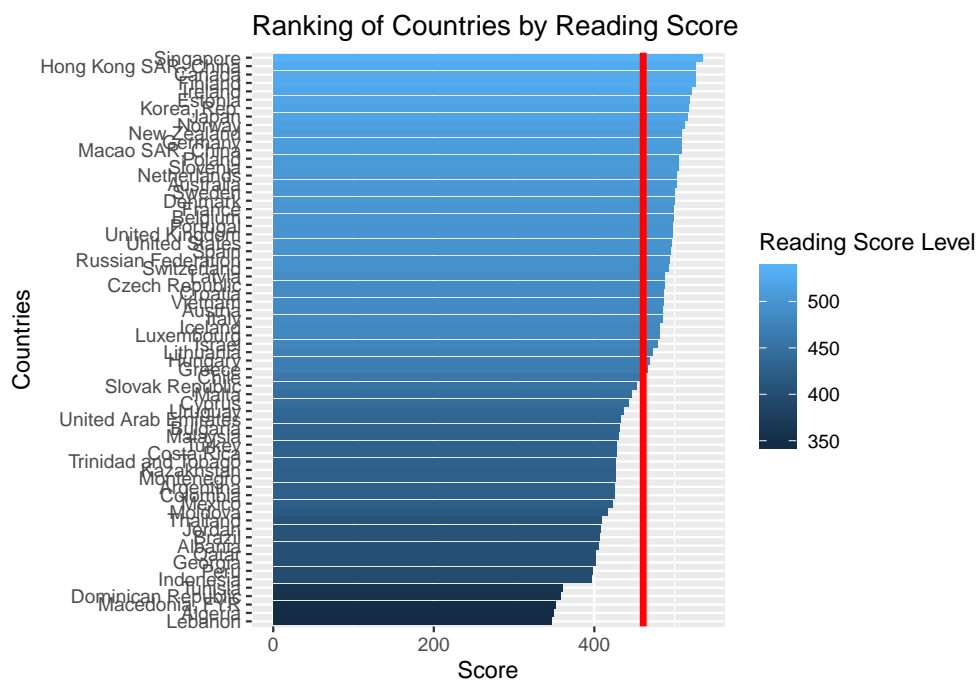


Figure 3: Notas de leitura

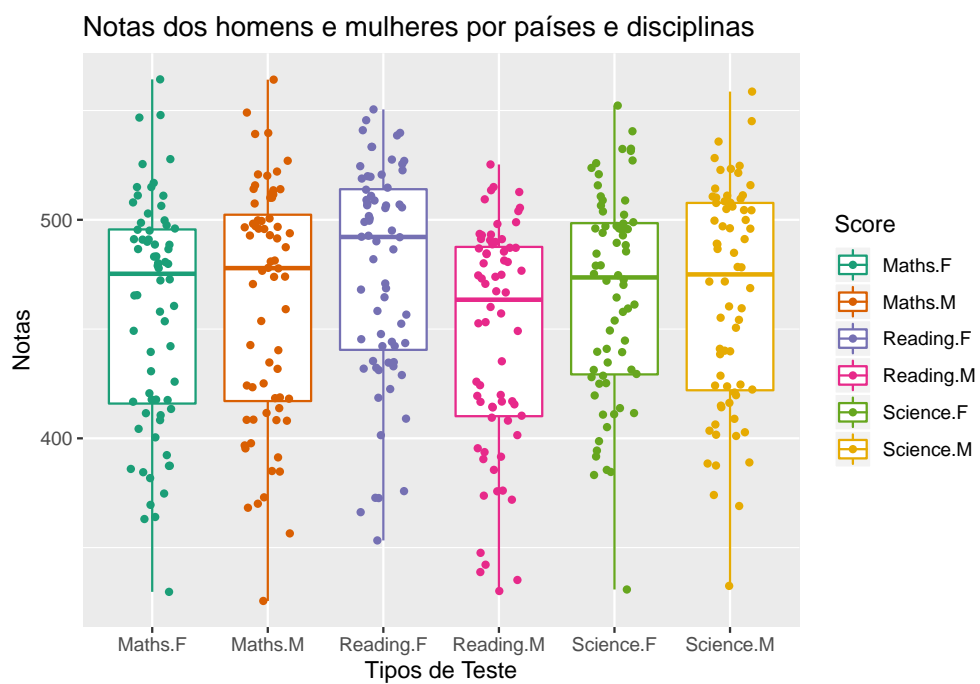


Figure 4: Boxplots das notas

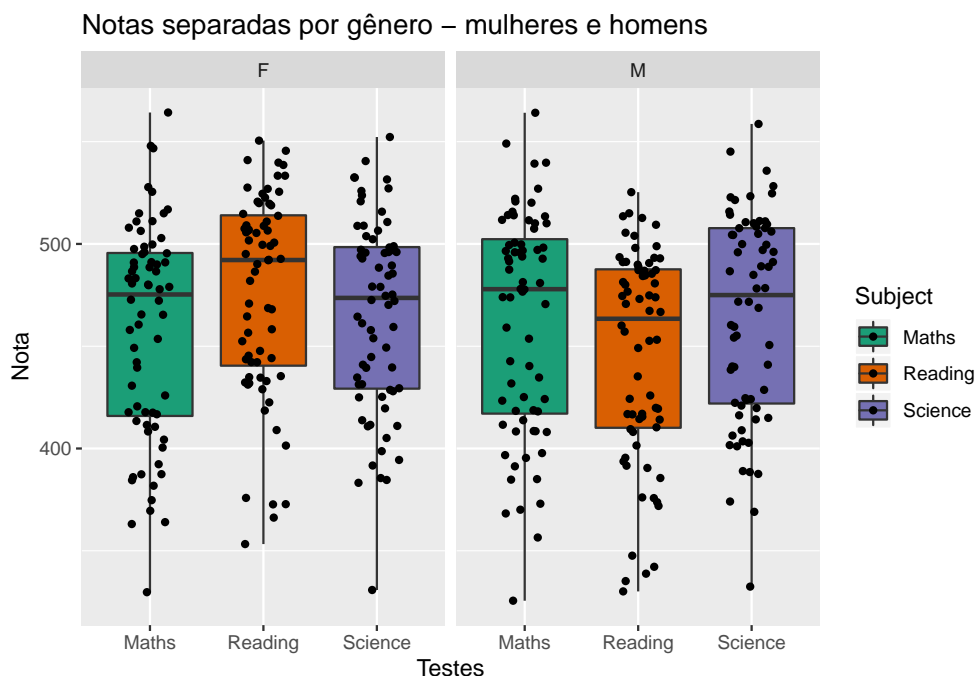


Figure 5: Boxplots das notas separadas por gênero

vantagem que pode ser considerada pequena em matemática e ciências, já na parte de leitura/interpretação as mulheres se saíram muito melhores do que os homens. Contudo, não podemos concluir que a situação geral é essa, olhando apenas o boxplot. Para tirar confirmar essa hipótese ou refutar, vamos estudar os dados com outras ferramentas para termos outros insights sobre este dataset.

Como vamos comparar a performance entre os homens e mulheres em cada teste (*matemática, ciências e leitura*) para todos os países participantes, precisaremos calcular a diferença percentual para cada tema, entre os homens e mulheres e depois então plotar o gráfico para analisarmos.

Vamos agora plotar as notas de matemática considerando essa nova informação que calculamos anteriormente.

Podemos tirar alguns insights do plot anterior aqui:

- Em geral, os homens saíram melhor do que as mulheres em matemática, na maioria dos países, podemos concluir que os homens tiraram notas melhores do que as mulheres.
- Interessante, em Singapura e Hongkong homens e mulheres se saíram igualmente bem, podemos checar isso no gráfico, onde a diferença nas notas em torno de zero em cada dos países citados. Isto é um insight interessante para os governadores desses locais, porque nós não queremos grandes diferenças nas performances entre homens e mulheres em educação. Já vimos existe uma grande nas notas dos testes de leitura, com clara superioridade das mulheres. Vamos então checar as notas de ciências utilizando a mesma metodologia que fizemos no gráfico anterior das notas de matemática.

### 3. Gráficos de Correlação

Para montar o gráfico de correlação, vamos primeiro calcular a correlação entre as variáveis numéricas, separando fazendo um subset do data frame **pisa2015**, através, dos seguintes comandos.

```
##           Maths.F Maths.M Reading.F Reading.M Science.F Science.M
## Maths.F      1.0000  0.9846   0.9377   0.9178   0.9711   0.9547
## Maths.M      0.9846  1.0000   0.9313   0.9468   0.9576   0.9758
## Reading.F    0.9377  0.9313   1.0000   0.9663   0.9556   0.9440
## Reading.M    0.9178  0.9468   0.9663   1.0000   0.9284   0.9693
```

### Homens são melhores em matemática?

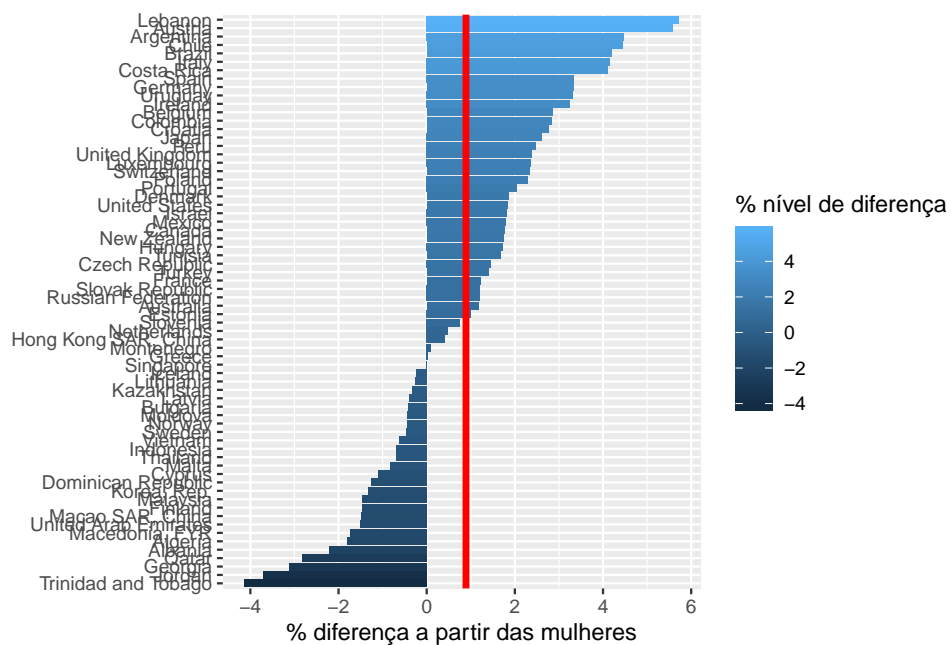


Figure 6: notas e diferenças percentuais

### Homens são melhores em ciências?

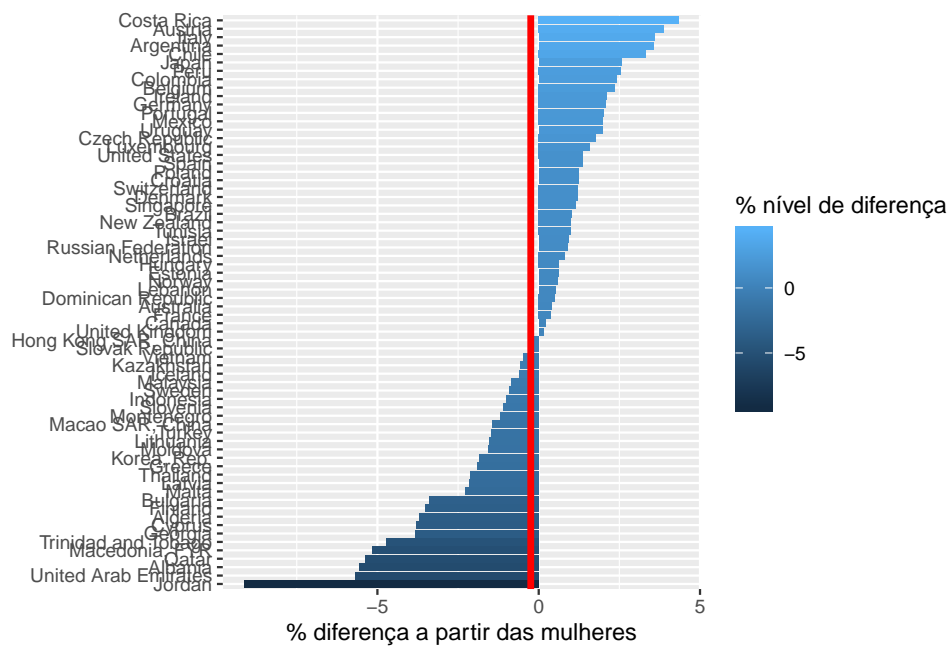


Figure 7: Notas e diferenças percentuais das notas em ciências

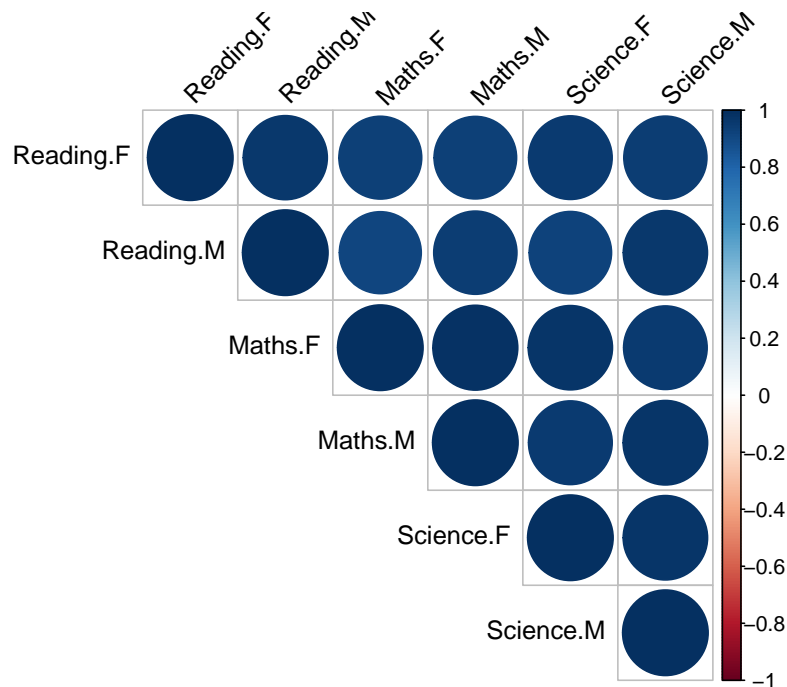


Figure 8: gráfico de correlação

```
## Science.F 0.9711 0.9576 0.9556 0.9284 1.0000 0.9737
## Science.M 0.9547 0.9758 0.9440 0.9693 0.9737 1.0000
```

Nós agora podemos calcular o **p-value** para verificar se a correlação é significativa.

```
##           Maths.M Reading.F Reading.M Science.F Science.M
## Maths.M      1.00      -0.95      -0.49      0.23      0.45
## Reading.F    -0.95       1.00       0.45     -0.28     -0.57
## Reading.M    -0.49       0.45       1.00     -0.77      0.18
## Science.F     0.23     -0.28     -0.77      1.00      0.23
## Science.M     0.45     -0.57      0.18      0.23      1.00
##
## n= 6
##
##
## P
##           Maths.M Reading.F Reading.M Science.F Science.M
## Maths.M           0.0040      0.3271      0.6653      0.3757
## Reading.F 0.0040           0.3731      0.5870      0.2370
## Reading.M 0.3271 0.3731           0.0761      0.7300
## Science.F 0.6653 0.5870      0.0761           0.6651
## Science.M 0.3757 0.2370      0.7300      0.6651
```

Quanto menor for o **p-value**, mais significativo é a correlação. O nosso objetivo aqui é começar a entender o uso dessa função para o cálculo de correlação entre variáveis no R. Para este dataset, era de se esperar que as variáveis fossem correlacionadas.

Para plotar esse resultado vamos usar o pacote **corrplot**.

```
##           Reading.F Reading.M Maths.F Maths.M Science.F Science.M
## Reading.F      1.0000      0.9663 0.9377 0.9313      0.9556 0.9440
```

## Reading.M	0.9663	1.0000	0.9178	0.9468	0.9284	0.9693
## Maths.F	0.9377	0.9178	1.0000	0.9846	0.9711	0.9547
## Maths.M	0.9313	0.9468	0.9846	1.0000	0.9576	0.9758
## Science.F	0.9556	0.9284	0.9711	0.9576	1.0000	0.9737
## Science.M	0.9440	0.9693	0.9547	0.9758	0.9737	1.0000

Podemos interpretar esse gráfico da seguinte maneira, quanto mais forte fica a cor e maior o tamanho das bolas, maior é a correlação. Este gráfico é um resultado visual, do que já havíamos visto na matrix de correlação.