


# ¡Hola!



Eu sou professora do DFNAE, aqui na UERJ.

Comecei usar R faz uns meses e gosto muito.

Com alguns exemplos hoje eu vou mostrar como **R** é  **lindo, fácil de aprender, e útil** para a física.



# Mas para quê aprender *mais outra* linguagem?

*-“Professora! Mas eu já tive que aprender <C++, Fortran, Python, etc.> “*

- Por quê usar? Para quê serve?
- Como começar usar? Quais ferramentas instalar?
- Mini-tutorial com exemplos de laboratório
- Conclusões e avaliação das vantagens e desvantagens





# Por quê R?

*O quê que é R?*

“ *R is an **open source** programming **language** and software **environment** for statistical computing and graphics [...]*

—Wikipedia em inglês  
Accessado 22/11/2017

# Por quê usar?



- É software livre
  - de graça!
  - o código fonte está disponível livremente
  - colaborativo: qualquer pessoa pode criar e desenvolver novos pacotes
- Tem um ambiente de trabalho integrado (consola)
- Tem estruturas análises estatísticos e gráficos
  - ... mais vários pacotes extras para fazer tudo mais bonito
- É usado por empresas como Google, Pfizer, Merck, Bank of América, InterContinental, e Shell, e também pelo IBGE.

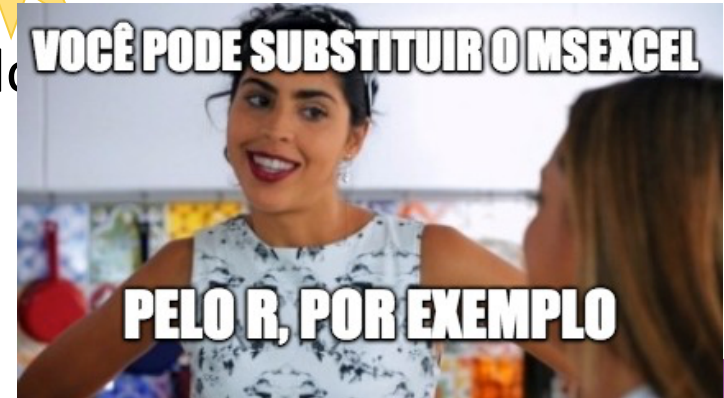


—Fonte: “Apostila do curso: Software Estatístico Libre R”  
Prof. Marcelo Rubens Amaral, IME –UERJ

[http://www.ime.uerj.br/~mrubens/slae/ApostilaCurso\\_Inovuerj.pdf](http://www.ime.uerj.br/~mrubens/slae/ApostilaCurso_Inovuerj.pdf)

# Para quê serve?

- Fazer cálculos estatísticos a partir de dados (quaisquer)
  - análise dos seus dados experimentais para obter resultados
  - análises das incertezas e erros
- Ajustes lineares
- Testes estatísticos
- Fazer gráficos lindos e limpinhos de dados teóricos em só um par de linhas
- Substituir o excel pelo R, por exemplo!






# Cómo começar usar?

Quais programas baixar? Pacotes? (Re)cursos online?



# 1. Baixar programas

- Obter  do site <http://www.r-project.org>
  - Tem instaladores pre-compilados para muitas plataformas, ou código fonte para compilar.
  - Para instruções no Windows (e uma boa introdução), ver apostila do Prof. Marcelo Rubens Amaral – IME UERJ [http://www.ime.uerj.br/~mrubens/slae/ApostilaCurso\\_Inovuerj.pdf](http://www.ime.uerj.br/~mrubens/slae/ApostilaCurso_Inovuerj.pdf)
- Eu gostei do **RStudio** <http://www.rstudio.com>
  - GUI mais prático para ter consola, gráficos, historial, scripts, ajuda, tudo na mão.





## 2. Instalar Pacotes

Já no ambiente de programação, instalar pacotes úteis:

```
install.packages("<nome>")
```

Carregar na console ou código fonte, chamando:

```
library(<nome>)
```

Exemplo:

```
>install.packages("latex2exp")  
>library(latex2exp)
```

Outros\*: plotly, Rcmdr, ggplot2...

\*dependendo do objetivo



# 3. Cursos e referencias

- Os modulos básicos do datacamp são gratuitos <http://www.datacamp.com>
- O curso do prof. M.R. Amaral do IME  
<http://www.ime.uerj.br/~mrubens/slae/OficinaEstatisticaSemFronteiras2017.html>
- Site “Estatística é com R” da UFF da profa. Luciane Ferreira Alcoforado (vídeos)  
[http://www.estadisticacomr.uff.br/?page\\_id=691](http://www.estadisticacomr.uff.br/?page_id=691)
- Outros tutoriais em inglês :
  - <http://www.r-tutor.com/>
  - <https://www.tutorialspoint.com/r/>
- A propria ajuda do R

```
help(plot)
```



# Mini-Tutorial de R

Baseado em alguns exemplos de laboratório



# Usando R para Distribuições e Histogramas

Em R posso definir um conjunto simples de medições na forma de um vetor.

comentário

*#defino um vetor com o comando c() e asigno o nome com a seta a esquerda <-*

comando

```
meuvetordedados<-c(620.35, 621.27, 483.87, 618.28,  
                    626.57, 603.36 , 626.57 , 626.57)
```

*#quando eu chamar ele, o conteudo é impresso dessa forma*  
meuvetordedados

resultado

```
## [1] 620.35 621.27 483.87 618.28 626.57 603.36 626.57  
626.57
```

# Analizando um conjunto de dados

Sobre o conjunto posso fazer varias operações (funções do R que operam sobre um vetor) p.ex.: ordenar, fazer a media, variância, desvio padrão

```
#ordenar os dados
sort(meuvetordedados)
## [1] 483.87 603.36 618.28 620.35 621.27 626.57 626.57 626.57
#avaliar a media
mean(meuvetordedados)
## [1] 603.355
#a variância
var(meuvetordedados)
## [1] 2389.43
#e o desvio padrão (populacional)
sd(meuvetordedados)
## [1] 48.88179
```



# Asignar valores e nomes e imprimir texto na tela

E posso dar a esses resultados nomes de variáveis e imprimir na tela o resultado resumido:

```
media<-mean(meuvetordedados)
variancia<-var(meuvetordedados)
desvio<-sd(meuvetordedados)

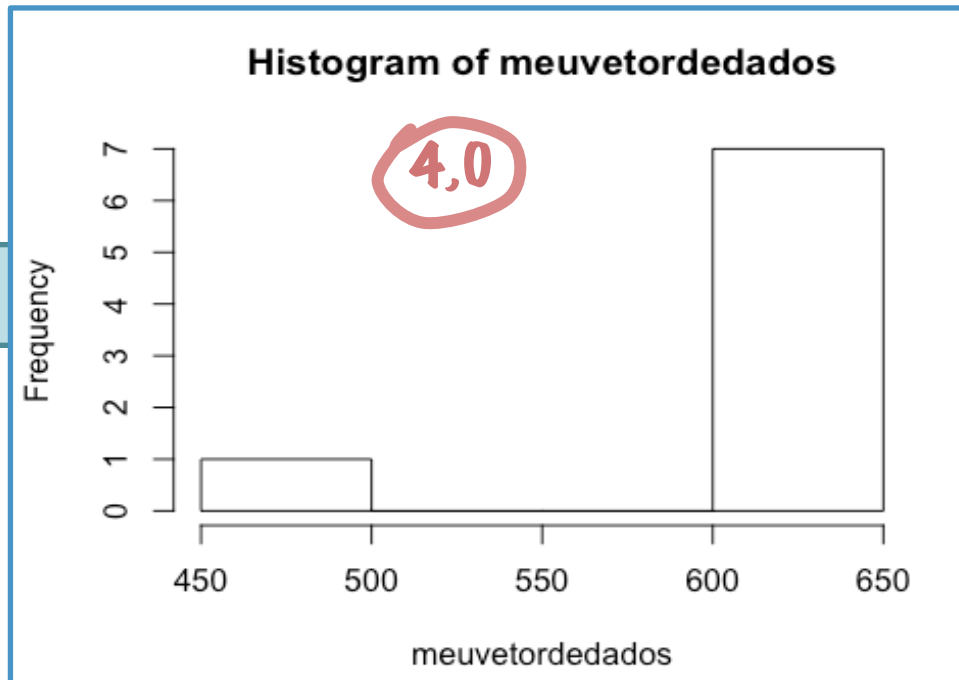
print(paste( "A media do comprimento de onda é ",media , " nm; a
variância ", variancia , " nm^{2} e o desvio padrão ", desvio, "nm"))

## [1] "A media do comprimento de onda é 603.355 nm; a variância
2389.42954285714 nm^{2} e o desvio padrão 48.8817915266732 nm"
```

# Desenhar um histograma

E também posso já simplesmente graficar a distribuição de frequências num histograma:

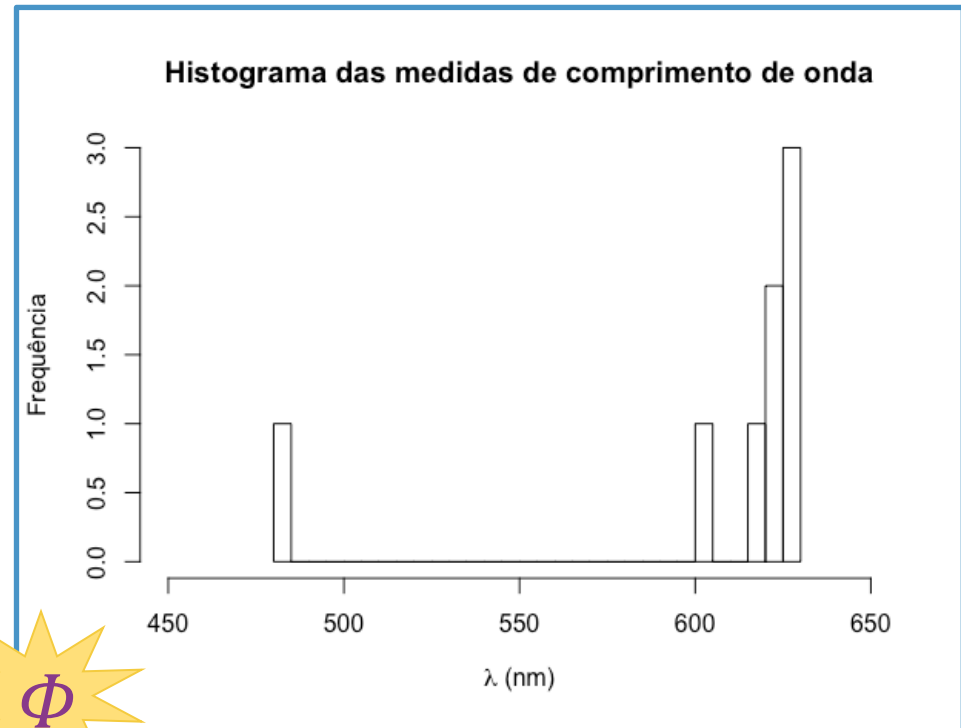
```
hist(meuvetordedados)
```



# Desenhar um histograma nota 10

O gráfico pode ser melhorado (e muito) se utilizarmos algumas opções

```
hist(meuvetordedados,
breaks=50,xlim=c(450,650),
ylim=c(0,3),
main="Histograma das medidas
de comprimento de onda",
xlab=TeX("$\\lambda$ (nm)"),
ylab="Frequência" )
```



\*Os dados são medições do comprimento de onda de um laser HeNe usando interferência por fenda dupla (Exp. Young)





# Tabelas e sistemas de varias variáveis

- Conjunto de medidas de 2 (ou mais) grandezas
- Por exemplo os dados coletados pela turma 2 nessa terça-feira foram salvos num arquivo texto. Eu posso ler esse arquivo no R:

```
meuarquivo<-file("~/Desktop/UERJStuff/fotoeletrico_dados_turma2.txt")
meufotoe<-read.table(meuarquivo,header=TRUE)
meufotoe
```

##		Cor	Comprimento.nm.	Frequência.x10.14Hz.	Potencial.V.
## 1		Amarelo	569.96	5.19	0.71
## 2		Verde	546.07	5.49	0.80
## 3		Azul	435.83	6.88	1.25
## 4		Violeta	404.66	7.41	1.51
## 5		Ultravioleta	364.90	8.22	1.83

# Tabelas e data.frame

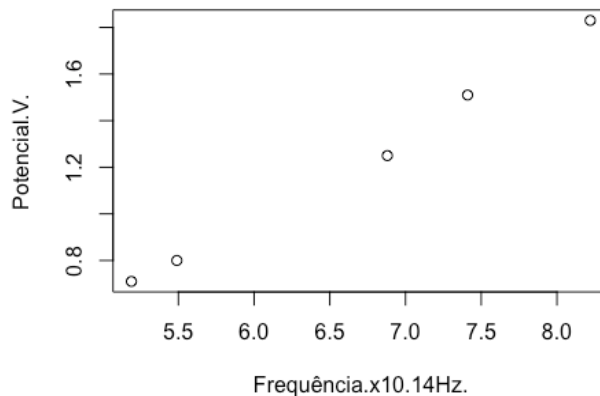
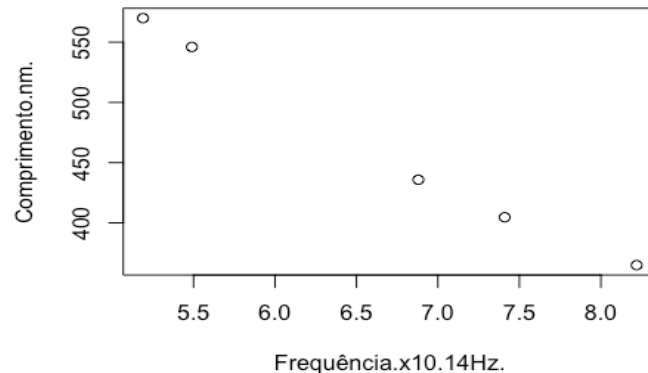
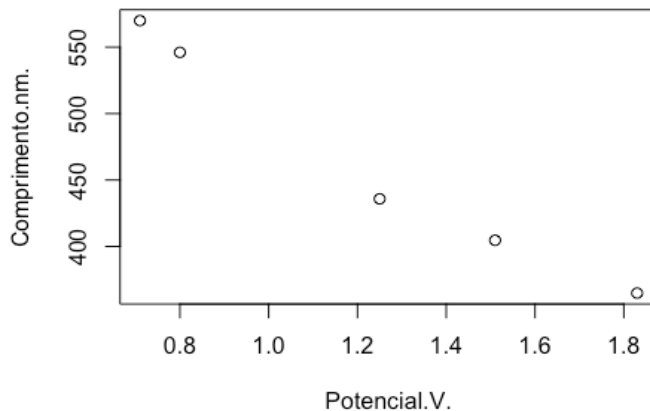
- O objeto onde foi salva a tabela é um "data frame" que contém 5 observações de 4 variáveis

```
str(meufotoe)
## 'data.frame':    5 obs. of  4 variables:
##  $ Cor                : Factor w/ 5 levels "Amarelo","Azul",...: 1 4 2
##  $ Comprimento.nm.     : num  570 546 436 405 365
##  $ Frequência.x10.14Hz.: num  5.19 5.49 6.88 7.41 8.22
##  $ Potencial.V.        : num  0.71 0.8 1.25 1.51 1.83
```

# Diagramas de dispersão e correlações

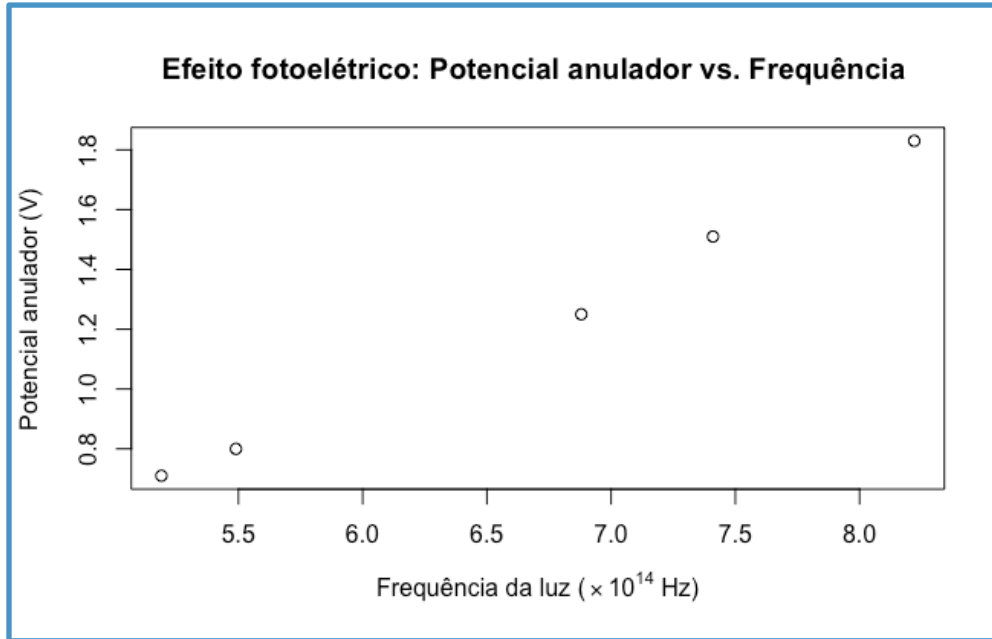
- Posso graficar as correlações entre as variáveis numéricas

```
plot(meufotoe[c(3,2)])  
plot(meufotoe[c(3,4)])  
plot(meufotoe[c(4,2)])
```



# Diagramas de dispersão e correlações

```
plot(meufotoe[c(3,4)], main="Efeito fotoelétrico: Potencial anulador vs. Frequência ",  
xlab= TeX("Frequência da luz ( $\times 10^{14}$  Hz)"),ylab="Potencial anulador (V)")
```



# Avaliar covariância e correlação

- Posso avaliar a variância de cada variável e a covariância entre elas

```
# seja entre como matriz covariância
cov(meufotoe[c(3,4)])
##              Frequência.x10.14Hz. Potencial.V.
## Frequência.x10.14Hz.          1.64297          0.6042
## Potencial.V.                  0.60420          0.2234

# ou como o valor da covariância entre os dois vetores
cov(meufotoe$Frequência.x10.14Hz.,meufotoe$Potencial.V.)
## [1] 0.6042

cor(meufotoe[c(3,4)])
##              Frequência.x10.14Hz. Potencial.V.
## Frequência.x10.14Hz.          1.0000000          0.9972966
## Potencial.V.                  0.9972966          1.0000000

cor(meufotoe$Frequência.x10.14Hz.,meufotoe$Potencial.V.)
## [1] 0.9972966
```

uma forma de  
selecionar uma  
coluna do  
data.frame

# Outro exemplo

- No caso de um **pêndulo simples** eu tenho 2 medidas, comprimento  $L$  e período medido (para 20 oscilações nesse caso)... mas a relação linear esta entre  $L$  e o período de uma oscilação ao quadrado ( $/4\pi^2$  e obteria  $g$ )

```
#comprimentos da linha do pendulo em metros
meuscomprimentos<-c(0.923, 0.825, 0.787,0.738,0.653)

#tempo de 20 osc em segundos
meustempos<- c(39.21, 37.33, 36.10, 35.53, 33.25)

meupendulo<-cbind.data.frame(meuscomprimentos,meustempos,meustempos/20.,(meustempos/20)^2/(4*pi^2))
meupendulo
```

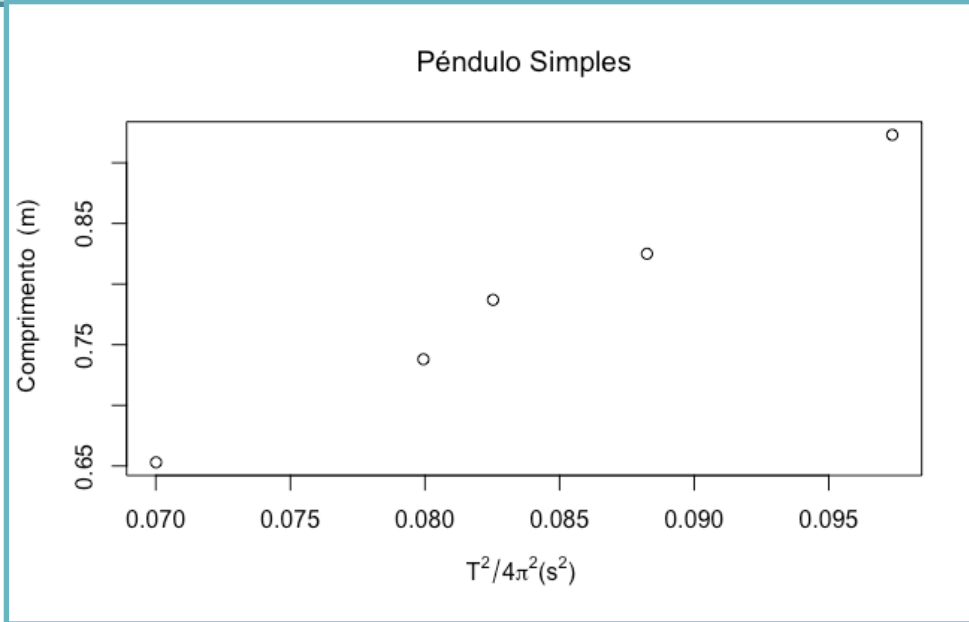
##	meuscomprimentos	meustempos	meustempos/20	(meustempos/20)^2/(4 * pi^2)
## 1	0.923	39.21	1.9605	0.09735852
## 2	0.825	37.33	1.8665	0.08824625
## 3	0.787	36.10	1.8050	0.08252674
## 4	0.738	35.53	1.7765	0.07994120
## 5	0.653	33.25	1.6625	0.07001056

# Gráfico



- De novo graficando a dispersão observamos a correlação

```
plot(meupendulo[c(4,1)],main=TeX("Pêndulo Simples"),ylab=" Comprimento (m)", xlab= TeX("$T^2/4\\pi^2 (s^2)$ "))
```



# Ajuste linear

- Para fazer o ajuste linear posso criar minha propria função

```
mmq<-function(x,y) {  
  # tenho 2 variaveis de observação (xi,yi) nos quais quero ajustar a reta  $y=a*x+b$   
  # pelo metodo de minimos quadrados  
  
  sigma_xy <- cov(x,y)  
  sigma_xx <- var(x)  
  meanx <- mean(x)  
  meany <- mean(y)  
  nobs<-length(y)  
  
  a <- sigma_xy/sigma_xx  
  b <- meany - a* meanx  
  print (paste("a =",a, "e b= ",b))  
  
  sumd2<-sum((y-a*x-b)^2)  
  ey<-sqrt(sumd/(nobs-2))  
  sigma_a <- ey/sqrt(sigma_xx)/nobs  
  sigma_b <- sigma_a*sqrt(mean(x^2))  
  
  print (paste("sigma_a = ",sigma_a,"sigma_b = ",sigma_b))  
  
  return(c(a, sigma_a, b, sigma_b))  
}
```

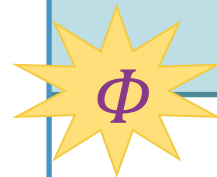
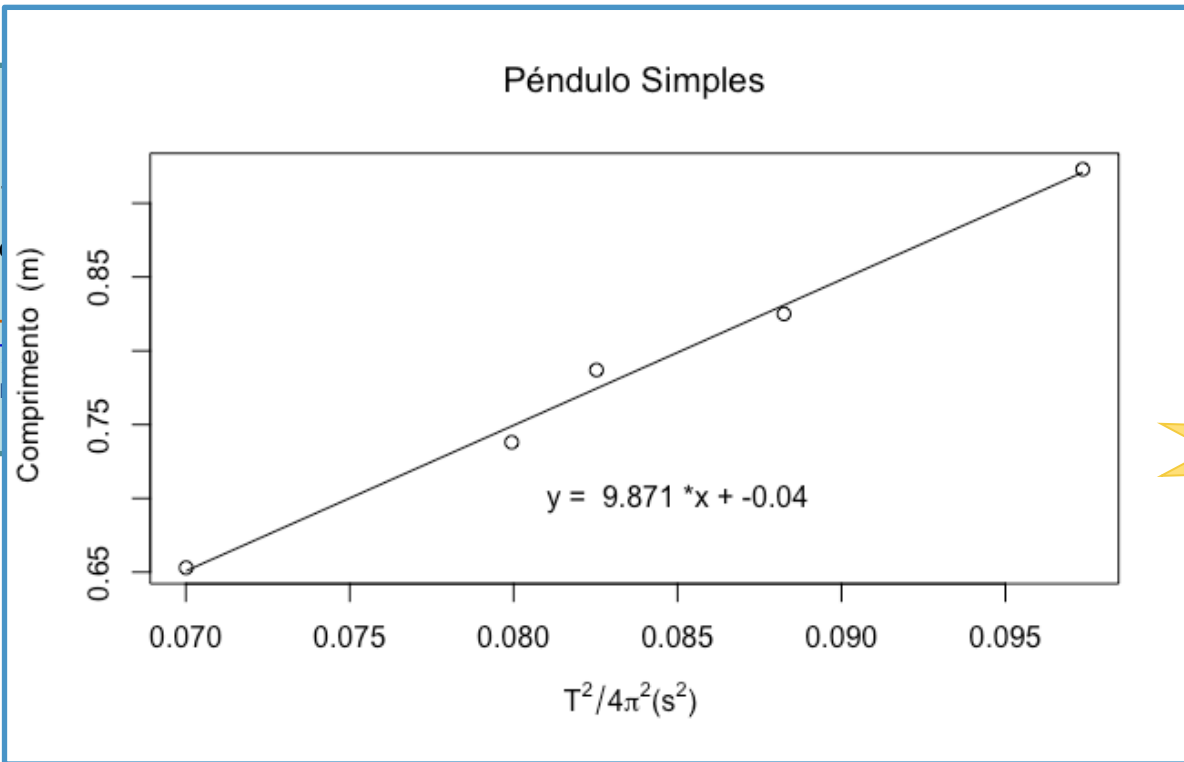


# Resultado do ajuste

- O resultado dos parâmetros da reta pela função é

```
res<- mmq(meupendulo)
## [1] "a = 9.871"
## [1] "sigma = 0.005"

plot(meupendulo,
     xlab= TeX("T^2/4*pi^2 (s^2)"),
     ylab= TeX("Comprimento (m)"),
     curve(res[1]*x + res[2], col="red", lty=1),
     text(0.085, 0.7,
          TeX("y = 9.871 *x + -0.04"),
          "*x +", font=1, col="red", cex=1.2))
```



# Outras opções de ajuste

- Posso usar outros metodos para resolver o mesmo problema

```
m <- cbind(1, meupendulo$(meustempos/20)^2/(4 * pi^2))
yo <- meupendulo$meuscomprimentos

# com algebra linear
solve(t(m) %*% m) %*% t(m) %*% yo
##           [,1]
## [1,] -0.04019797
## [2,]  9.87121497

# com uma função preexistente do R
lm.fit(m,yo)

## $coefficients
##           x1           x2
## -0.04019797  9.87121497
##
## $residuals
## [1]  0.002151105 -0.005899719  0.012558805 -0.010918843  0.002108651
## (...)
```



# CONCLUSÃO

# Vantagens



- Gratis!
- Fácil de aprender usar
- Muitos pacotes disponiveis (tidyverse, graficos 3d, ajustes)
- Resultados lindos e precisos (compete com outros)

# Desvantagens

- Zero 😂
- Tem muitos pacotes e tem que pesquisar, testar, (errar), etc.
- Como Python, pode demorar em alguns processos (onde C ou C++ dão conta melhor)