



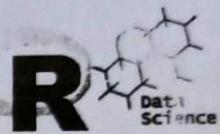
R for Data Science

Bài 11: *Visualization*

Phòng LT & Mạng

https://csc.edu.vn/lap-trinh-va-cSDL/R-Programming-Language-for-Data-Science_190

2020



Vai trò của trực quan hóa dữ liệu



Định nghĩa:

- “The use of computer-supported, visual representations of abstract data to amplify cognition.” [Card et al., 1999]
- Trực quan hóa dữ liệu: Là việc sử dụng các biểu diễn trực quan của dữ liệu trừu tượng thông qua sự hỗ trợ của máy tính để mở rộng nhận thức. (tạm dịch)



Vai trò của trực quan hóa dữ liệu

□ Định nghĩa:

- “The representation and presentation of data to facilitate understanding.” [Kirk, 2016]
- Là sự diễn tả và trình bày dữ liệu để tạo điều kiện cho việc hiểu biết về dữ liệu. (tạm dịch)



Vai trò của trực quan hóa dữ liệu

□ Tại sao cần trực quan hóa dữ liệu?



Dữ liệu khó hiểu...



Vai trò của trực quan hóa dữ liệu

□ Tại sao cần trực quan hóa dữ liệu?



Quá nhiều số...



Vai trò của trực quan hóa dữ liệu



□ Tại sao cần trực quan hóa dữ liệu?

A	B	C	D	G
Sales Team Review				
2	Salesperson	Region Covered	February 2017 Sales	Cost of Sales
3	Jeffrey Burke	Oklahoma	\$ 28,000	\$ 2,460
4	Amy Fernandez	North Carolina	\$ 23,138	\$ 1,521
5	Mark Hayes	Massachusetts	\$ 25,092	\$ 1,530
6	Judith Ray	California	\$ 21,839	\$ 1,923
7	Randy Graham	South Carolina	\$ 23,342	\$ 2,397
8	Christina Foster	Delaware	\$ 23,368	\$ 1,500
9	Judy Green	Texas	\$ 21,510	\$ 1,657
10	Paula Hall	Virginia	\$ 21,314	\$ 2,418
11	February Sales Total		\$ 187,603	\$ 15,406
12				

Các cuộc họp nhảm chán với các spreadsheet...



Vai trò của trực quan hóa dữ liệu

□ Ưu điểm

- Giúp người xem thấy trực quan và dễ hiểu
- Giúp giải quyết vấn đề
- Kể câu chuyện về dữ liệu trong thời gian rất ngắn



Vai trò của trực quan hóa dữ liệu



□ Nguyên tắc thiết kế:

- "Good data visualization is:
 - 1. Trustworthy
 - 2. Accessible
 - 3. Elegant" - Andy Kirk*

Theo: Kirk, A. *Data Visualisation: A handbook for Data Driven Design.*
SAGE publications, 2016.



Vai trò của trực quan hóa dữ liệu

❑ Nguyên tắc thiết kế:

- Trustworthy

- Sự tin tưởng khó kiểm, dễ mất
- Tính trung thực và toàn vẹn cần phải có ở mọi nơi trong quá trình thực hiện data science



Vai trò của trực quan hóa dữ liệu

❑ Nguyên tắc thiết kế:

- Accessible

- Cần phải biết ai là người sẽ xem các biểu đồ
- Hiểu mục đích của việc trực quan hóa



Vai trò của trực quan hóa dữ liệu

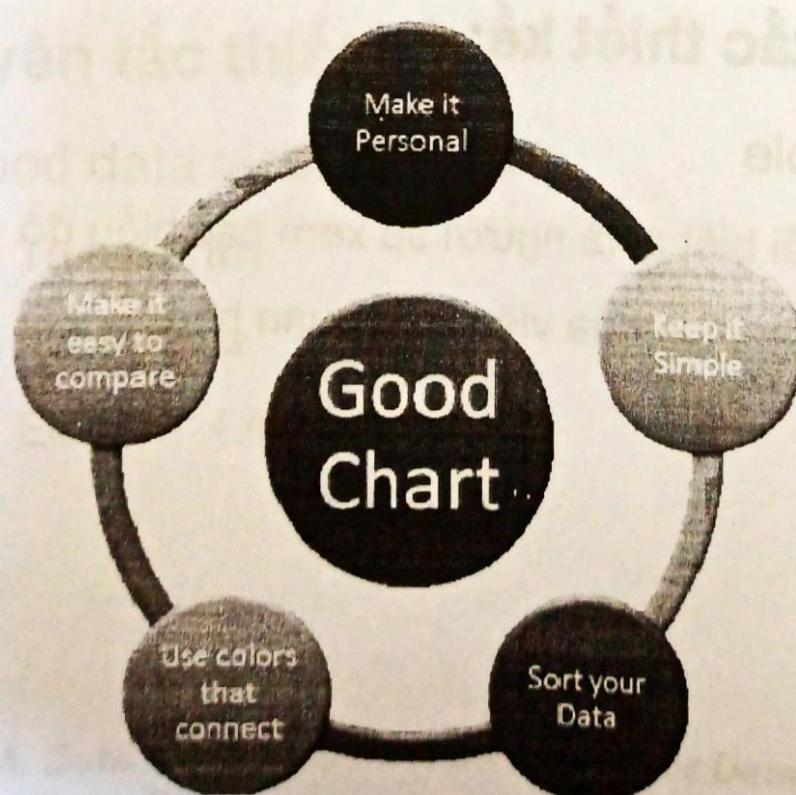
❑ Nguyên tắc thiết kế:

- Elegant

- Tập trung vào các yếu tố liên quan
- Có phong cách riêng nếu có thể
- Suy nghĩ cách thiết kế trước khi thể hiện



Vai trò của trực quan hóa dữ liệu



1. Pie chart
2. Bar chart
3. Boxplot
4. Histogram
5. Line graph
6. Scatterplot



Pie chart

□ R có rất nhiều thư viện để tạo biểu đồ - chart và đồ thị - graph. Pie chart là một biểu đồ tròn được chia thành nhiều phần với nhiều màu sắc khác nhau. Mỗi thành phần được gán nhãn và các giá trị số tương ứng thể hiện trên biểu đồ.

□ Để vẽ các loại biểu đồ, install.pakages("plotrix")



Pie chart

□ Sử dụng pie() để tạo biểu đồ

`pie(x, labels, radius, main, col, clockwise)`

- Trong đó:

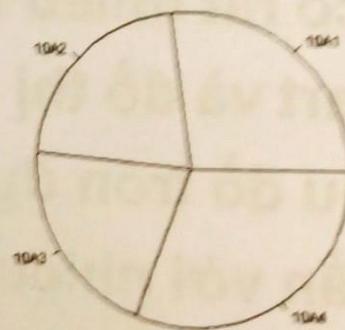
- x: là một vector chứa các giá trị số được dùng trong pie chart
- labels: là một vector các nhãn của từng thành phần
- radius: bán kính của đường tròn pie chart (giá trị từ -1 đến 1)
- main: tiêu đề của pie chart
- col: bảng màu
- clockwise: một giá trị logic quy định các thành phần vẽ theo chiều kim đồng hồ hoặc ngược chiều kim đồng hồ



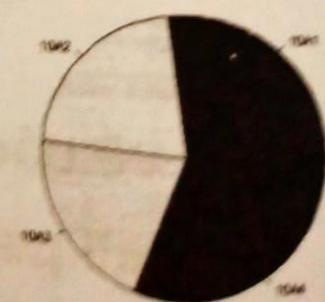
Pie chart

- Ví dụ:

```
> # Create data for the graph.  
> x <- c(46, 37, 35, 53)  
> labels <- c("10A1", "10A2", "10A3", "10A4")  
> # Give the chart file a name.  
> png(file = "~/R/chuong14/Hinh/classes_pie.jpg")  
> # Plot the chart.  
> pie(x, labels)  
> # Save the file.  
> dev.off()  
null device  
1
```



```
> # create data for the graph.  
> x <- c(46, 37, 35, 53)  
> labels <- c("10A1", "10A2", "10A3", "10A4")  
> # Give the chart file a name.  
> png(file = "~/R/chuong14/Hinh/classes_pie_colors.jpg")  
> # Plot the chart with title and rainbow color pallet.  
> # length of the pallet should be same as the number of values => length(x)  
> pie(x, labels, main = "Classes pie chart", col = rainbow(length(x)))  
> # Save the file.  
> dev.off()  
null device  
1
```



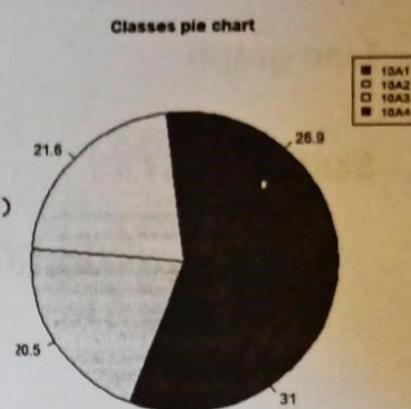
Pie chart

□ Tính tỷ lệ % cho các thành phần và chú thích cho biểu đồ

```
piepercent<- round(100*x/sum(x), 1)
pie(x, labels = piepercent, radius, main,...)
legend(position, labels, cex = character_expansion, fill
= color_pattern)
```

- Ví dụ:

```
> # Create data for the graph.
> x <- c(46, 37, 35, 53)
> labels <- c("10A1", "10A2", "10A3", "10A4")
> #percentage
> piepercent<- round(100*x/sum(x), 1)
> # Give the chart file a name.
> png(file = "~/R/chuong14/Hinh/classes_pie_percentage_legends.jpg")
> # Plot the chart.
> pie(x, labels = piepercent, main = "Classes pie chart", col = rainbow(length(x)))
> # Insert legends for chart
> legend("topright", labels, cex = 0.8, fill = rainbow(length(x)))
> # Save the file.
> dev.off()
null device
1
```



R programming language for Data Science

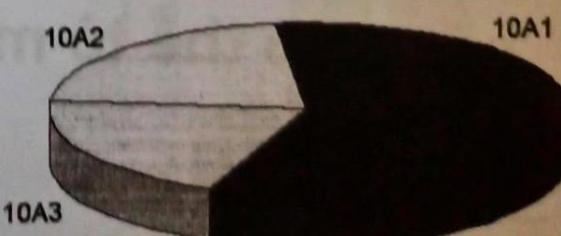
Pie chart

□ 3D Pie chart

- Sử dụng thư viện library(plotrix)
- Và phương thức pie3D(x, labels = labels, main =
 pie_name, col = color_pattern[, explore = 0.1])
- Ví dụ:

```
> # Create data for the graph.
> x <- c(46, 37, 35, 53)
> labels <- c("10A1", "10A2", "10A3", "10A4")
> # Get the library.
> library(plotrix)
> # Give the chart file a name.
> png(file = "~/R/chuong14/Hinh/classes_pie_3D.jpg")
> # Plot the chart.
> pie3D(x, labels = labels, main = "Classes Pie Chart", col = rainbow(length(x)))
> # Save the file.
> dev.off()
null device
1
```

Classes Pie Chart



R programming language for Data

Nội dung

1. Pie chart
2. Bar chart
3. Boxplot
4. Histogram
5. Line graph
6. Scatterplot



Bar chart

□ Bar chart là một biểu đồ khối với dữ liệu là các khối hình chữ nhật có chiều cao tỷ lệ thuận với giá trị.

□ Sử dụng barplot() để tạo biểu đồ khối với cả trục tung và trục hoành và cũng có thể tô màu cho các khối hình chữ nhật.



Bar chart

□ Sử dụng barplot() để tạo biểu đồ

`barplot(H, xlab, ylab, main, names.arg, col)`

- Trong đó:

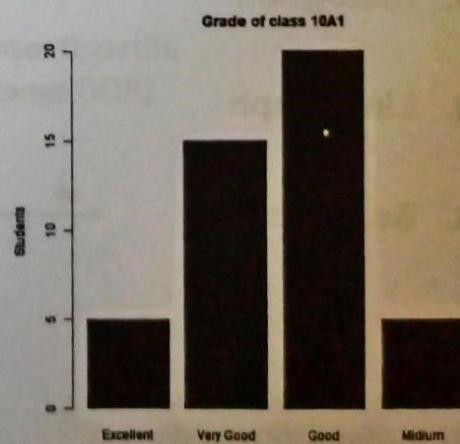
- H: vector hoặc matrix chứa các giá trị số trong bar chart
- xlab: nhãn của trục hoành
- ylab: nhãn của trục tung
- main: tiêu đề của bar chart
- names.arg: vector các nhãn của từng khối
- col: màu sắc cho các khối



Bar chart

- Ví dụ:

```
> # Create the data for the chart.
> H <- c(5,15,20,5)
> M <- c("Excellent", "Very Good", "Good", "Midium")
> # Give the chart file a name.
> png(file = "~/R/chuong14/Hinh/10A1_grade_barchart.png")
>
> # Plot the bar chart.
> barplot(H,names.arg = M,xlab = "Grade",ylab = "Students",col = "blue",
+         main = "Grade of class 10A1",border = "red")
> # Save the file.
> dev.off()
null device
1
```



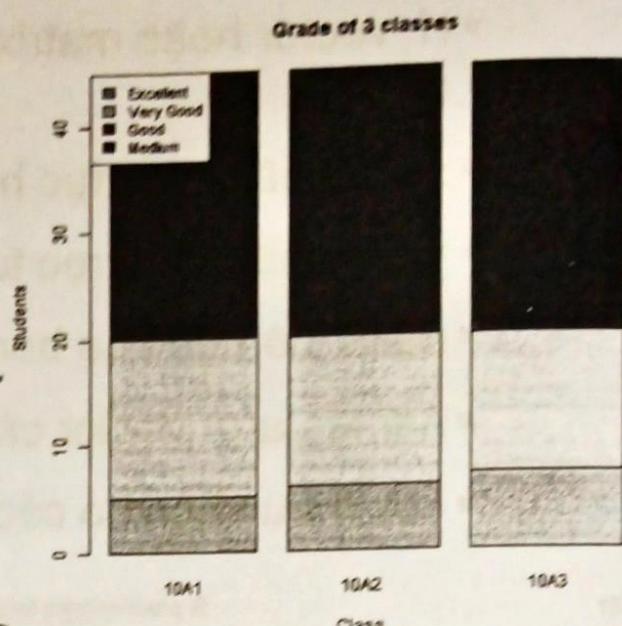
Bar chart

□ Biểu đồ nhóm xếp chồng

- Khi muốn vẽ các nhóm trên cùng một biểu đồ ta sử dụng dữ liệu đầu vào là matrix

- Ví dụ:

```
# Create the input vectors.
colors <- c("green","orange","brown", "red") # for 4 grades
classes <- c("10A1","10A2","10A3")
grades <- c("Excellent","Very Good","Good","Medium")
# Create the matrix of the values.
Values <- matrix(c(5, 6, 7, 15, 14, 13, 20, 21, 22, 5, 4, 3),
                  nrow = 4, ncol = 3, byrow = TRUE)
# Give the chart file a name.
png(file = "~/R/chuong14/Hinh/grade_barchart_stacked.png")
# Create the bar chart.
barplot(Values, main = "Grade of 3 classes", names.arg = classes,
        xlab = "Class", ylab = "Students", col = colors)
# Add the legend to the chart.
legend("topleft", grades, cex = 0.9, fill = colors)
# Save the file.
dev.off()
```



Nội dung

1. Pie chart
2. Bar chart
3. Boxplot
4. Histogram
5. Line graph
6. Scatterplot

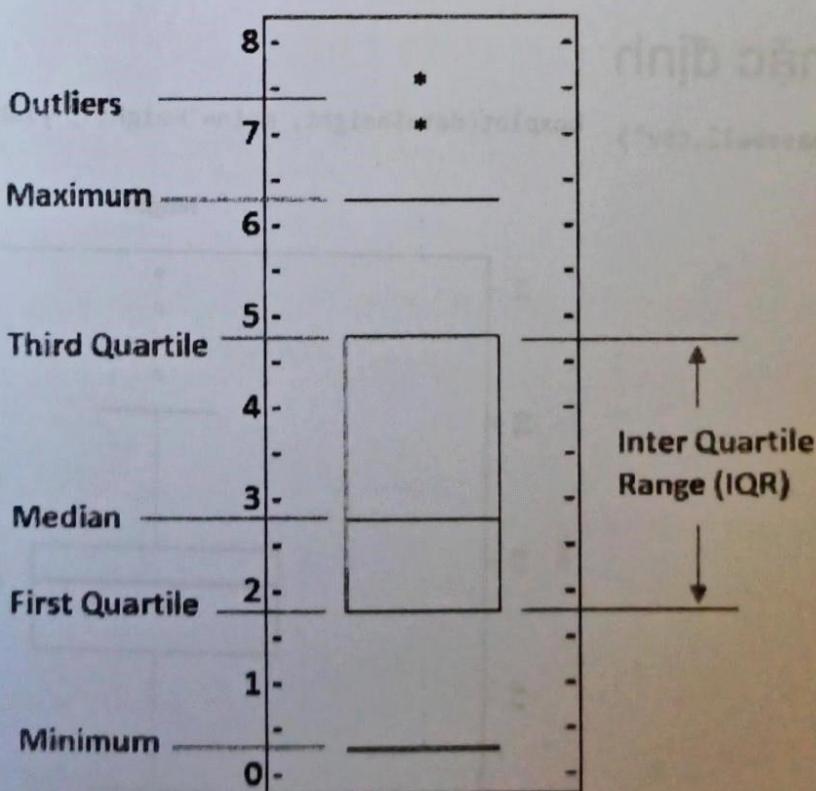


Boxplot

- Boxplot là một biểu thể hiện dữ liệu liên tục của một thuộc tính. Nó giúp chúng ta quan sát dữ liệu và phát hiện ngoại lệ (outlier)
- Boxplot chia tập dữ liệu thành các khoảng phần tư (quartiles). Phần thân của biểu đồ là một hộp box (vì thế mới có tên gọi là boxplot), đi từ Q1 đến Q3.



Boxplot



Boxplot

☐ Sử dụng boxplot () để tạo biểu đồ

`boxplot(x, data, notch, varwidth, names, xlab, ylab, main)`

- Trong đó:

- x: là một vector hoặc fomular chứa các giá trị số được dùng trong boxplot
- data: dataframe
- notch: đặt bằng TRUE khi muốn vẽ thêm notch
- varwidth: đặt bằng TRUE khi muốn vẽ chiều rộng của box tương ứng với kích thước của mẫu
- names: in tên của nhóm dưới từng boxplot
- xlab, ylab: tên trên trục hoành, trục tung
- main: tiêu đề của boxplot

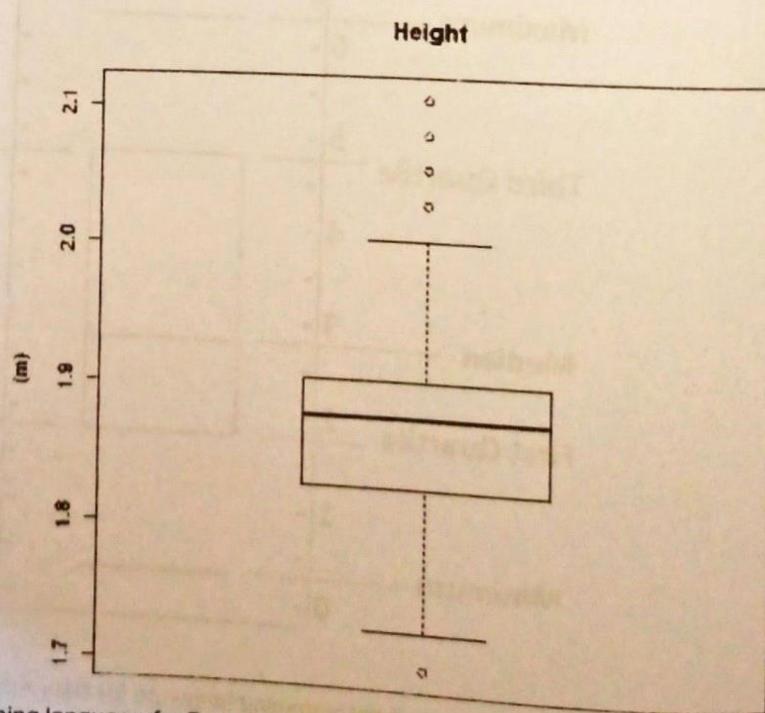


Boxplot

- Dạng mặc định

```
data <- read.csv("baseball.csv")    boxplot(data$height, main="Height", ylab="(m)")  
head(data)
```

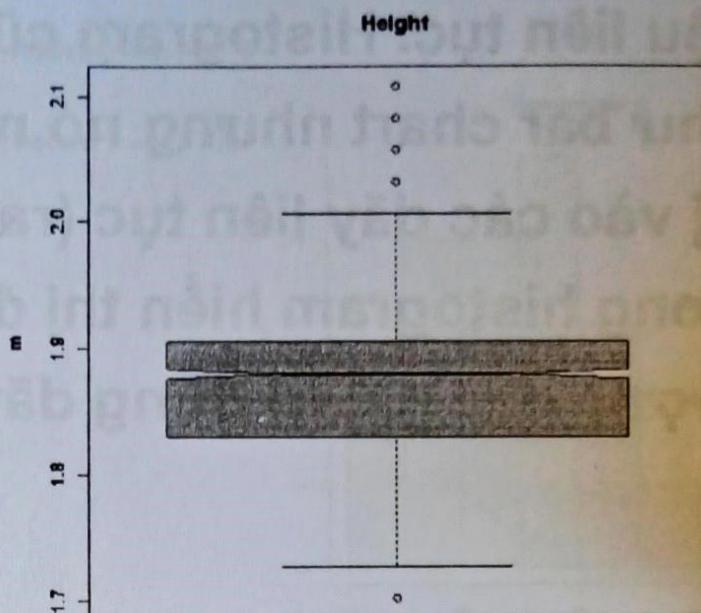
X	height	weight
0	1.8796	81.64656
1	1.8796	97.52228
2	1.8288	95.25432
3	1.8288	95.25432
4	1.8542	85.27530
5	1.7526	79.83219



Boxplot

- Dạng có notch tại median

```
boxplot(data$height, data = data,
        ylab = "m",
        notch = TRUE,
        varwidth = TRUE,
        col = c("green"),
        main = "Height")
```



Nội dung

1. Pie chart
2. Bar chart
3. Boxplot
4. Histogram
5. Line graph
6. Scatterplot



Histogram

❑ Histogram là một dạng đồ thị hiển thị dạng phân phối tần suất của một tập dữ liệu liên tục. Histogram cũng tương tự như bar chart nhưng nó nhóm các giá trị vào các dãy liên tục (range). Mỗi khối trong histogram hiển thị độ cao của số lượng các giá trị trong dãy liên tục đó.



Histogram

❑ Sử dụng `hist()` để tạo biểu đồ
`hist(v, main, xlab, xlim, ylim, breaks, col, border)`

• Trong đó:

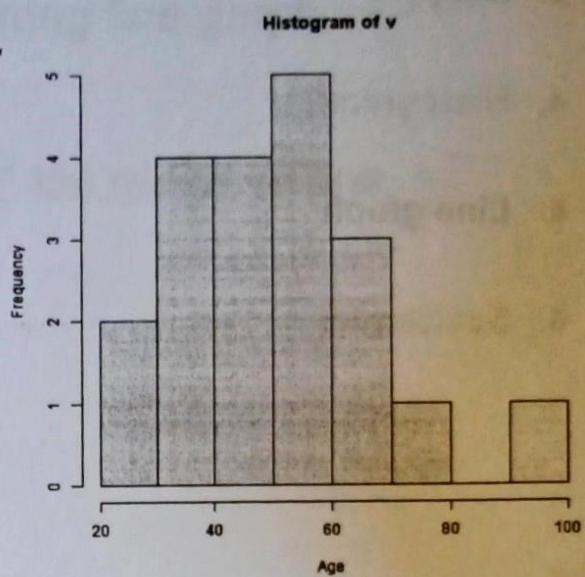
- v: là một vector chứa các giá trị số được dùng trong histogram
- main: tiêu đề của histogram
- xlab: tên trục hoành
- xlim: dãy các giá trị trên trục hoành
- ylim: dãy các giá trị trên trục tung
- breaks: độ rộng của mỗi thanh
- col: bảng màu
- border: màu đường viền của mỗi khối



Histogram

- Ví dụ:

```
> # Create data for the graph.
> # dataset of 20-age samples
> v <- c(36, 25, 38, 46, 55, 68, 72, 55, 36, 38, 67, 45,
+      22, 48, 91, 46, 52, 61, 58, 55)
>
> # Give the chart file a name.
> png(file = "~/R/chuong15/Hinh/ages_histogram.png")
>
> # Create the histogram.
> hist(v,xlab = "Age",col = "orange",border = "blue"
>
> # Save the file.
> dev.off()
null device
1
```



R programming language for Data Science



Histogram

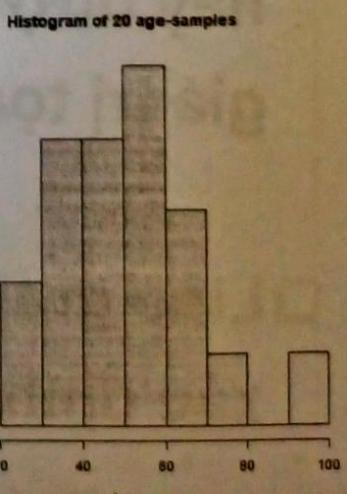
- Ví dụ: khi bổ sung thêm các thông số

```
# Create data for the graph.
# dataset of 20 age-samples
v <- c(36, 25, 38, 46, 55, 68, 72, 55, 36, 38, 67, 45,
     22, 48, 91, 46, 52, 61, 58, 55)

# Give the chart file a name.
png(file = "~/R/chuong15/Hinh/ages_histogram.png")

# create the histogram.
hist(v, main = "Histogram of 20 age-samples", xlab = "Age",
      xlim = c(0, 100), ylim = c(0, 5), col = "orange",
      border = "blue", breaks = 5)

# Save the file.
dev.off()
```



R programming language for Data Science

Nội dung

1. Pie chart
2. Bar chart
3. Boxplot
4. Histogram
5. Line graph
6. Scatterplot



Line graph

□ **Line graph là một biểu đồ nối một loạt các điểm với nhau bằng cách vẽ đường nối giữa các điểm. Các điểm này được sắp xếp theo một trong các giá trị tọa độ (thường là x).**

□ **Line graph thường được sử dụng để xác định xu hướng trong dữ liệu.**



Line graph



□ Sử dụng plot() để tạo biểu đồ

`plot(v, main, type, col, xlab, ylab)`

• Trong đó:

- v: vector chứa các giá trị số trong line graph
- main: tiêu đề của line graph
- type: "p" khi muốn vẽ điểm, "l" khi muốn vẽ line, "o" khi muốn vẽ cả điểm và line
- xlab: nhãn của trục hoành
- ylab: nhãn của trục tung
- col: màu sắc điểm và line

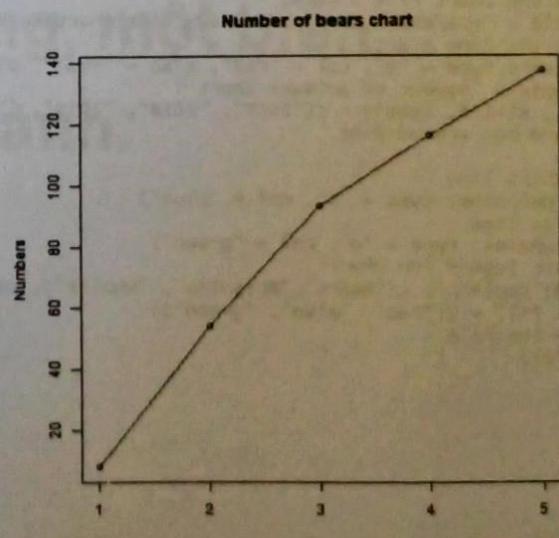


Line graph



• Ví dụ

```
# Create the data for the chart.  
bears <- c(8, 54, 93, 116, 137)  
  
# Give the chart file a name.  
png(file = "~/R/chuong15/Hinh/bears_line_chart.jpg")  
  
# Plot the bar chart.  
plot(bears, type = "o", col = "red", xlab = "Year", ylab = "Numbers",  
     main = "Number of bears chart")  
  
# Save the file.  
dev.off()
```



Line graph

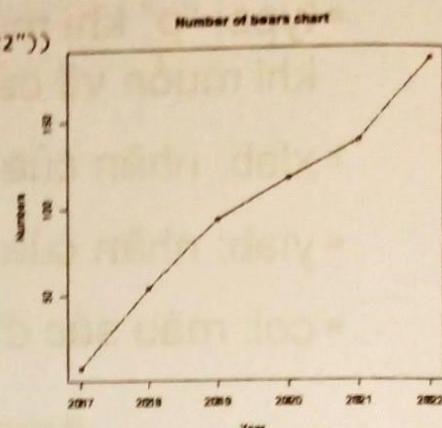
- Có thêm nhãn cho các điểm theo trục hoành sử dụng axis()

- Ví dụ:

```
# Create the data for the chart.
bears <- c(8, 54, 93, 116, 137)

# Give the chart file a name.
png(file = "~/R/chuong15/Hinh/bears_line_chart_axis.jpg")

# Plot the bar chart.
plot(bears, type = "o", col = "red", xlab = "Year", ylab = "Numbers",
     main = "Number of bears chart")
axis(1, at=1:6, labels = c("2017", "2018", "2019", "2020", "2021", "2022"))
# save the file.
dev.off()
```



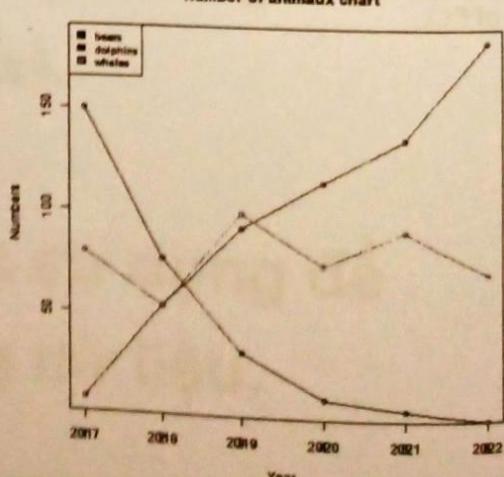
Line graph

- Nhiều line trên cùng một line chart

- Ví dụ:

```
# Create the data for the chart.
bears <- c(8, 54, 93, 116, 137, 184)
dolphins <- c(150, 77, 32, 11, 6, 1)
whales <- c(80, 54, 100, 76, 93, 72)
# calculate range from 0 to max value of bears, dolphins and whales
q_range <- range(0, bears, dolphins, whales)
# Give the chart file a name.
png(file = "~/R/chuong15/Hinh/animaux_line_chart_axis.jpg")

# Plot the line chart.
plot(bears, type = "o", col = "red", xlab = "Year", ylab = "Numbers",
     main = "Number of animaux chart")
axis(1, at=1:6, labels = c("2017", "2018", "2019", "2020", "2021", "2022"))
# create box around plot
box()
# dolphins line
lines(dolphins, type = "o", col = "blue")
# whales line
lines(whales, type = "o", col = "green")
# insert legend for chart
legend("topleft", c("bears", "dolphins", "whales"), cex = 0.8,
       fill = c("red", "blue", "green"))
# save the file.
dev.off()
```



1. Pie chart
2. Bar chart
3. Boxplot
4. Histogram
5. Line graph
6. Scatterplot



Scatterplot

□ **Scatterplot hiển thị nhiều điểm trên mặt phẳng Cartesian. Mỗi điểm đại diện cho giá trị của hai biến, một biến được chọn trên trục tung, một biến được chọn trên trục hoành.**



Scatterplot

□ Sử dụng plot() để vẽ biểu đồ

`plot(x, y, main, xlab, ylab, xlim, ylim, axes)`

- Trong đó:

- x: tập dữ liệu giá trị trên trục hoành
- y: tập dữ liệu giá trị trên trục tung
- main: tiêu đề của scatterplot
- xlab: nhãn của trục hoành
- ylab: nhãn của trục tung
- xlim: các giới hạn của các giá trị trên trục hoành được sử dụng để vẽ
- ylim: các giới hạn của các giá trị trên trục tung được sử dụng để vẽ
- axes: cho biết cả 2 trục được vẽ trên biểu đồ hay không

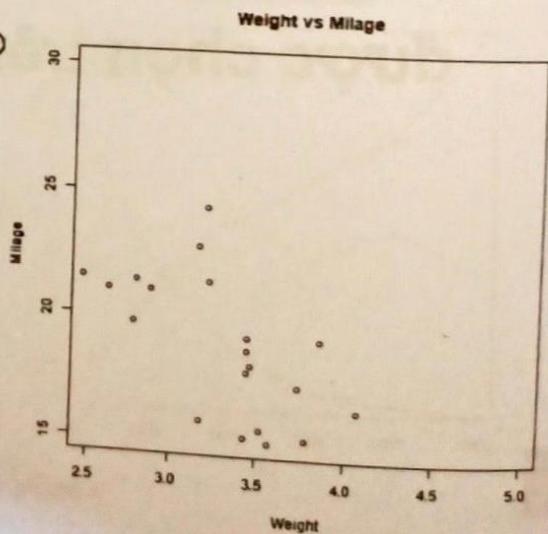


Scatterplot



- Ví dụ:

```
> # Get the input values.
> input <- mtcars[,c('wt', 'mpg')]
> print(head(input))
      wt  mpg
Mazda RX4     2.620 21.0
Mazda RX4 Wag 2.875 21.0
Datsun 710    2.320 22.8
Hornet 4 Drive 3.215 21.4
Hornet Sportabout 3.440 18.7
Valiant       3.460 18.1
> # Give the chart file a name.
> png(file = "~/R/chuong15/Hinh/mtcars_scatterplot.png")
> # Plot the chart for cars with weight between
> # 2.5 to 5 and mileage between 15 and 30.
> plot(x = input$wt, y = input$mpg,
+       xlab = "Weight",
+       ylab = "Milage",
+       xlim = c(2.5,5),
+       ylim = c(15,30),
+       main = "Weight vs Milage"
+ )
> # Save the file.
> dev.off()
null device
1
```

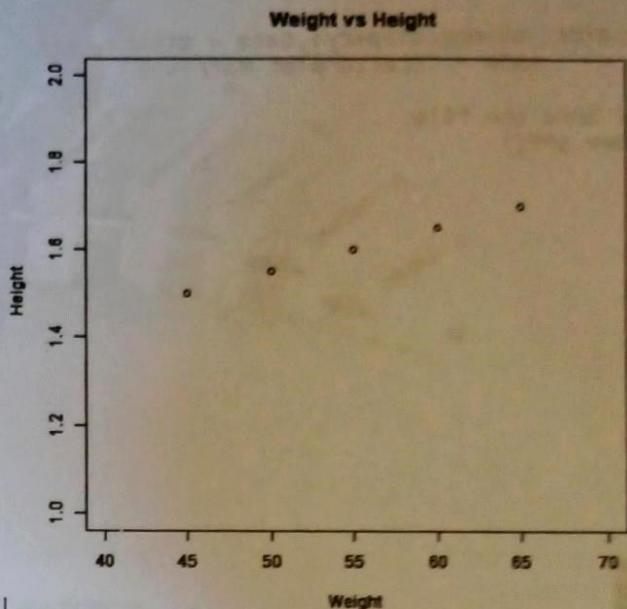


Scatterplot



• Ví dụ:

```
# Get the input values.  
bmi.data <- data.frame(  
  id = c(1:5),  
  name = c("Ann", "Dan", "Rosy", "Ryan", "Angela"),  
  height = c(1.5, 1.55, 1.6, 1.65, 1.7),  
  weight = c(45, 50, 55, 60, 65),  
  stringsAsFactors = FALSE  
)  
print(bmi.data)  
# Give the chart file a name.  
png(file = "~/R/chuong15/Hinh/bmi_scatterplot.png")  
# Plot the chart for persons with weight between  
# 40 to 70 and height between 1.0 and 2.0.  
plot(x = bmi.data$weight, y = bmi.data$height,  
  xlab = "Weight",  
  ylab = "Height",  
  xlim = c(40, 70),  
  ylim = c(1.0, 2.0),  
  main = "Weight vs Height")  
)  
# Save the file.  
dev.off()
```



R programming language for Data Science

Scatterplot



□ Scatterplot Matrix

• Khi chúng ta có nhiều hơn hai biến (variable) và chúng ta muốn tìm mối quan hệ giữa một biến và biến còn lại thì ta dùng scatterplot matrix.

• Sử dụng **pairs(formula, data)** để tạo biểu đồ:

- formula: đại diện cho loạt các biến được sử dụng trong pairs.
- data: đại diện cho bộ dữ liệu mà các biến sẽ được thực hiện



R programming language for Data Science

Scatterplot

• Ví dụ:

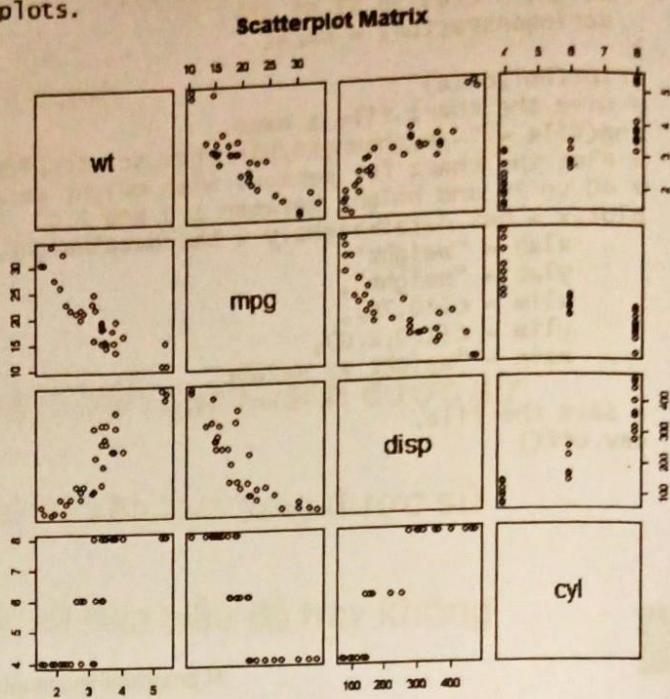
```
# Give the chart file a name.
png(file = "~/R/chuong16/Hinh/scatterplot_matrices.png")

# Plot the matrices between 4 variables giving 12 plots.

# One variable with 3 others and total 4 variables

pairs(~wt+mpg+disp+cyl,data = mtcars,
      main = "Scatterplot Matrix")

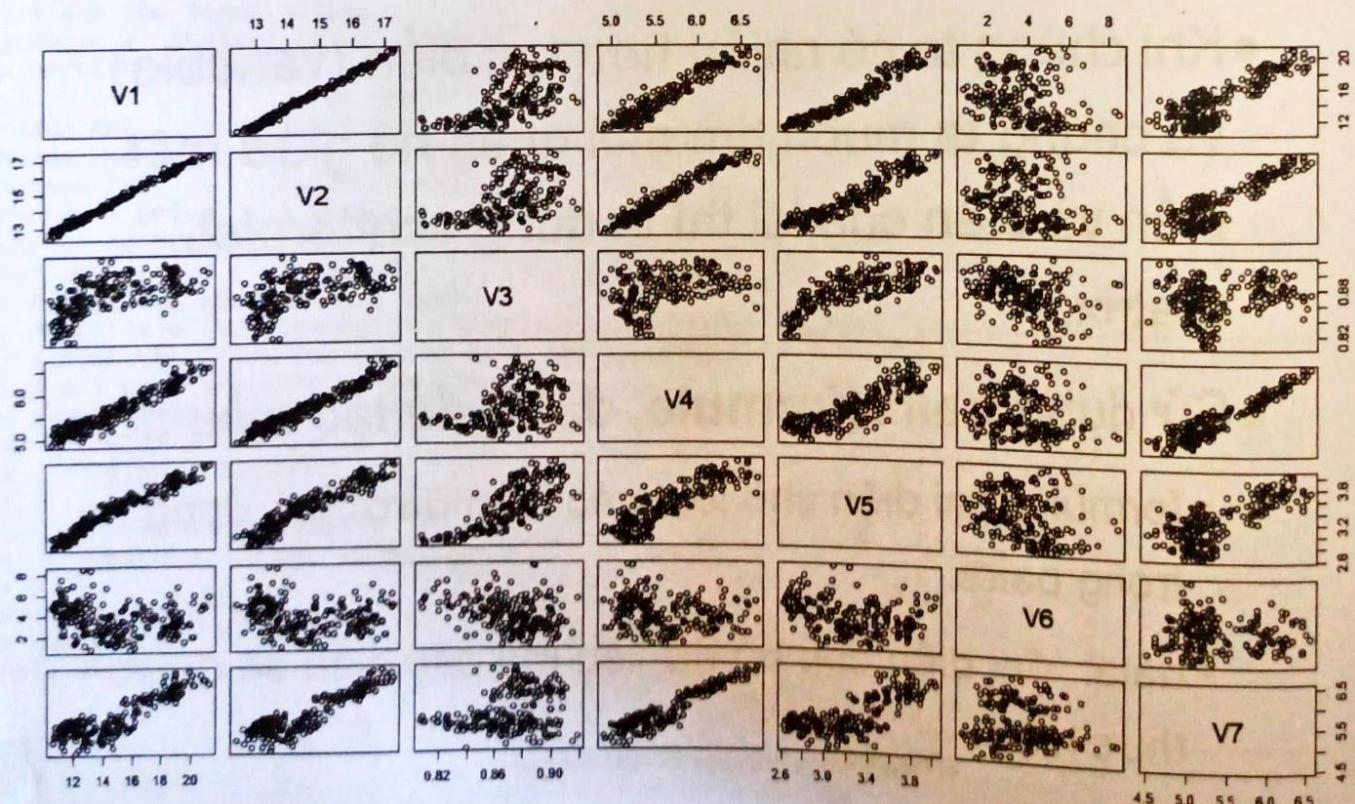
# Save the file.
dev.off()
```



R programming language

Scatterplot

```
df <- read.table("~/R/chuong15/seeds_dataset.txt", header = F)
print(df)
pairs(~V1+V2+V3+V4+V5+V6+V7, data=df)
```



Chapter 11: Visualization

Exercise 1: Painters school – PIE

- Sử dụng thư viện MASS
- Trong MASS có dữ liệu painters
- Đọc dữ liệu painters => cho biết kiểu dữ liệu, số dòng, số cột
- Lấy dữ liệu school trong painters => cho biết kiểu dữ liệu school (Factor)
- Chuyển school thành table (dùng phương thức table(school)). In kết quả
- Vẽ pie chart dựa trên kết quả này
- Vẽ pie chart với vector màu tùy chọn

Exercise 2: PIE 3D

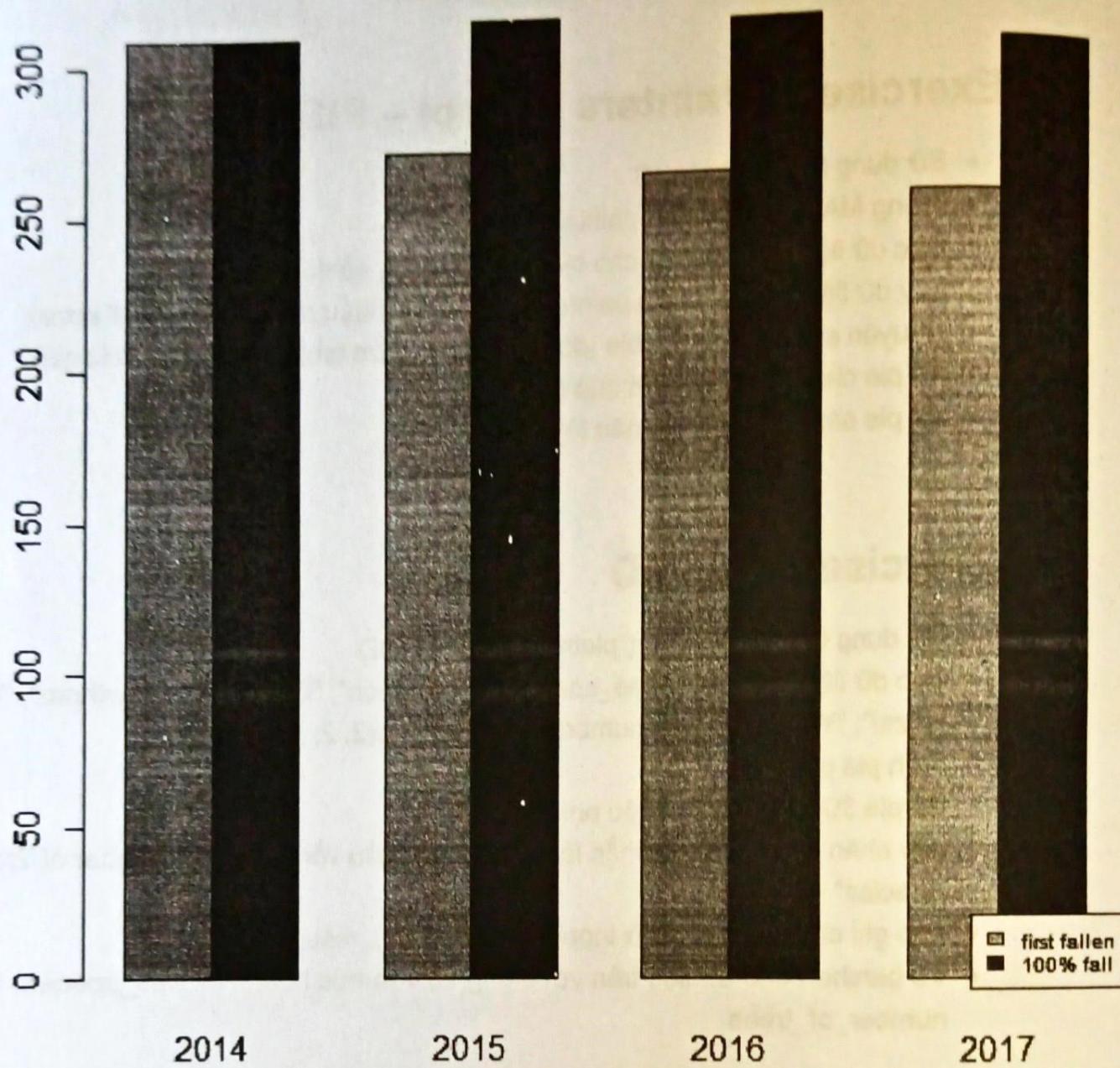
- Sử dụng thư viện library("plotrix") để vẽ pie 3D
- Tạo dữ liệu như sau: Tree_species <- c("Beech", "Chestnut", "Hawthorn", "Red maple", "Witch Hazel", "Yellow Birch"), number_of_trees <- c(2, 2, 1, 4, 1, 1)
- Tính pie percent
- Vẽ pie 3D dựa trên dữ liệu phía trên
- Lấy nhãn hiển thị từng phần là số cây, màu cầu vồng, tên là "Number of Trees of each Species"
- Tạo ghi chú cho pie 3D ở top right, lấy tên cây, màu cầu vồng
- Vẽ barchart cho dữ liệu trên với thông tin trên trục hoành là Tree_species, trên trục tung là number_of_trees

Exercise 3: BAR CHART nhóm cột

- Cho dữ liệu: year <- c(2014, 2015, 2016, 2017), first_fallen <- c(309, 271, 263, 255), one_hundred_fallen <- c(309, 314, 312, 304)
- Hãy vẽ biểu đồ nhóm cột như hình sau:

First and last dates of leaf fall in Yellow Birch

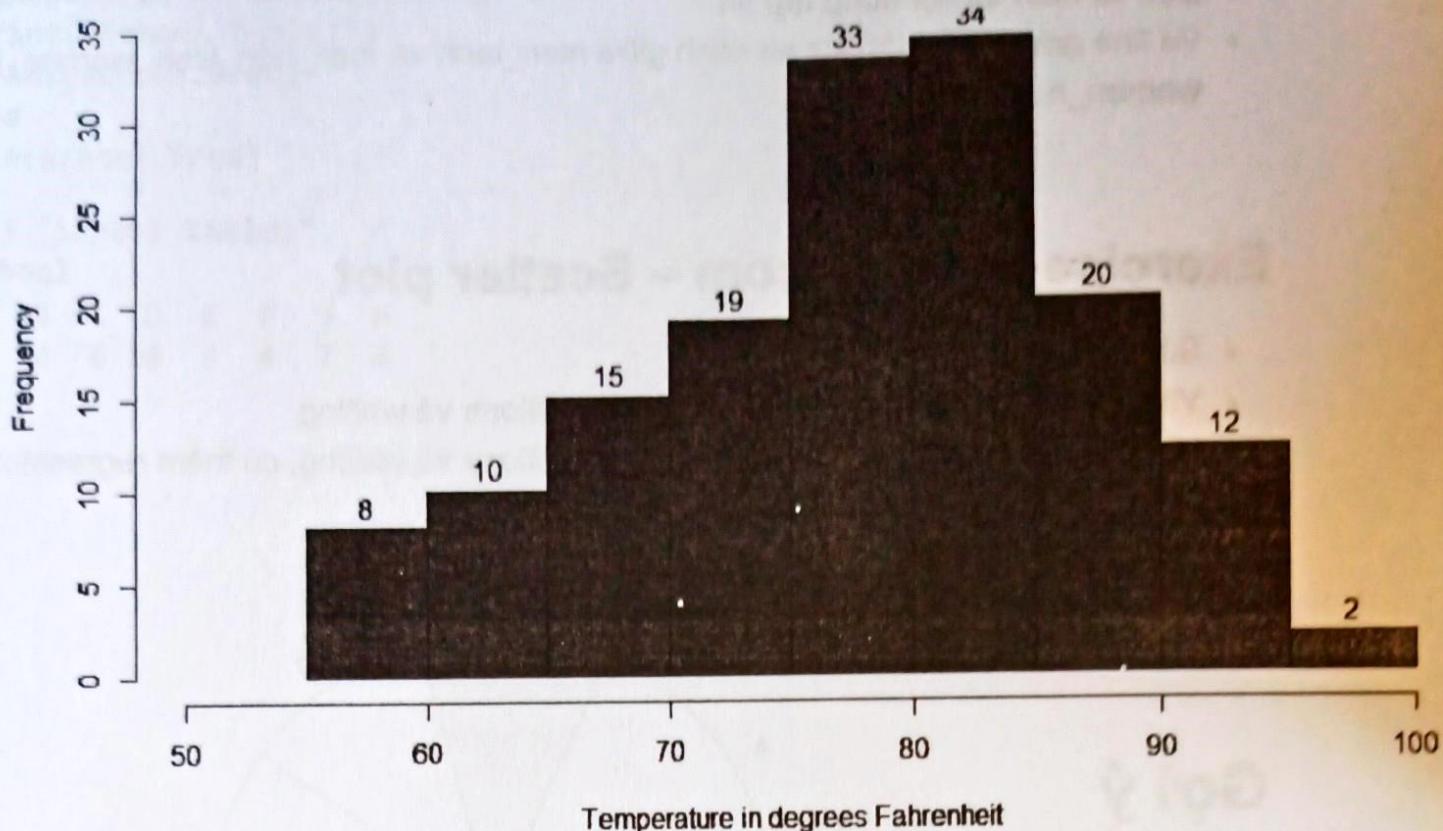
T H



Exercise 4: Airquality - Histogram

- Sử dụng dữ liệu airquality
- In thông tin về dữ liệu này
- Tạo biến chứa dữ liệu Temp của airquality
- Vẽ histogram liên quan đến biến chứa Temp này với số cột phân chia dữ liệu là 5.
- Thay vì trục tung là Frequency => hãy chuyển thành đồ thị có trục tung là Density
- Hãy vẽ một đồ thị có số lượng mẫu trên mỗi cột (có 9 cột) như hình sau:

Maximum daily temperature at La Guardia Airport



- Hãy vẽ một đồ thị chỉ có các cột giá trị 55, 60, 70, 75, 80, 90, 100

Exercise 5: AirPassengers - Histogram

- Sử dụng dữ liệu AirPassengers
- In thông tin về dữ liệu này
- Vẽ histogram thể hiện thông tin AirPassengers với 10 khoảng (cột), màu cầu vồng, dùng density

Exercise 6: Chol - Histogram

- Cung cấp tập tin chol.txt
- Đọc nội dung tập tin vào
- Cho biết kiểu của nội dung
- In thông tin nội dung
- Vẽ histogram thể hiện thông tin AGE, có gắn thông tin số lượng trên mỗi cột

Exercise 7: Leaves had fallen – Line graph

- Sử dụng dữ liệu: years <- c(2014, 2015, 2016, 2017), first_fallen <- c(309, 271, 263, 255)
one_hundred_fallen <- c(309, 314, 312, 304)
- Vẽ biểu đồ hiển thị cả first_fallen và one_hundred_fallen

Exercise 8: Tech_com – line graph



- Cung cấp tập tin Tech_comp.csv
- Đọc và hiển thị nội dung tập tin
- Vẽ line graph thể hiện sự so sánh giữa men_tech và men_non_tech, women_tech và women_non_tech

Exercise 9: Tech_com – Scatter plot

- Sử dụng dữ liệu có sẵn faithful
- Vẽ biểu đồ thể hiện mối quan hệ giữa eruptions và waiting
- Vẽ biểu đồ thể hiện mối quan hệ giữa eruptions và waiting, có thêm regression line

Gợi ý

Exercise 1: Painters school – PIE

```
In [1]: library(MASS)
```

```
In [2]: print(paste("Data type:", class(painters)))
print(paste("Nrows:", nrow(painters), ", ncols:", ncol(painters)))
```

```
[1] "Data type: data.frame"
[1] "Nrows: 54 , ncols: 5"
```

```
In [3]: school <- painters$School
print(paste("School type:", class(school)))
print("School:")
print(school)
```

```
[1] "School type: factor"
[1] "School:"
[1] A A A A A A A A A A A B B B B B C C C C C C D D D D D D D D D D E E E E E
E
[39] E F F F F G G G G G G H H H H
Levels: A B C D E F G H
```

```
In [4]: # chuyen factor thanh table  
school.freq <- table(school)  
print("School table:")  
print(school.freq)  
#ve  
pie(school.freq)
```

```
[1] "School table:"
```

```
school
```

	A	B	C	D	E	F	G	H
10	10	6	6	10	7	4	7	4

