Data analysis and visualization in R

UC Merced R curriculum

Andrea Sánchez-Tapia - Sara Ribeiro Mortara
¡liibre! - RLadies+

2020-10-29

last time

- we talked about **matrices** and **lists** using function `matrix()` as an example
- we talked about data frame objects, `str()`, `dim()`, `nrow()`, `ncol()`, and subsetting `[rows, columns]`
- we downloaded a file, read it into disk, removed rows with NAs and saved it back into a **processed** data folder
- we talked about **factors**: in R>4.0 you need to specify them with `factor()`

today

- exploratory data analysis [**Why** do we plot our data?]
- basic plotting functions [**How** do we plot our data?]

Exploratory data analysis

exploratory data analysis (EDA)

- control the quality of your data
- support the selection of statistical procedures
- evaluate if data attend the **assumptions** of the statistical tests (ex. normality)
- suggest hypotheses for the relationship of your data and new studies
- **EDA is NOT data wrangling or manipulation**
- your hypotheses based on theory are **central** to guide these analyses

exploratory data analysis (EDA)

- EDA can take 20-50% of your analysis time
- it should be performed during the data collection
- uses a lot of visual techniques
- EDA will help you **understand** your data

what we need to know about our data

- do they contain NAs? do we have a lot of zeroes?
- how are the variables distributed? are they centered? are they symmetric? bimodal?
- are there outliers?
- do the variables follow some distribution?
- do they need to be transformed?
- are the variables related? what is the shape of the relationship between variables? (ex. linear)
- was the sampling effort sufficient?

what we need to know about our data

- central tendency measures: mean, median, mode
- variation/dispersion measures: range, range width, variance, standard deviation, variation coefficient
- data distribution: quantiles, inter-quantile ranges, *boxplots*, *histograms*.
- relationship between variables: *scatterplots*, correlations, linear models

The Anscombe quartet

The Anscombe quartet

```
data("anscombe")
dim(anscombe)
```

```
## [1] 11  8
```

```
head(anscombe)
```

```
##   x1 x2 x3 x4   y1   y2    y3   y4
## 1 10 10 10  8 8.04 9.14  7.46 6.58
## 2  8  8  8  8 6.95 8.14  6.77 5.76
## 3 13 13 13  8 7.58 8.74 12.74 7.71
## 4  9  9  9  8 8.81 8.77  7.11 8.84
## 5 11 11 11  8 8.33 9.26  7.81 8.47
## 6 14 14 14  8 9.96 8.10  8.84 7.04
```

The Anscombe quartet

```
class(anscombe)
```

```
## [1] "data.frame"
```

```
str(anscombe)
```

```
## 'data.frame':    11 obs. of  8 variables:
##  $ x1: num  10 8 13 9 11 14 6 4 12 7 ...
##  $ x2: num  10 8 13 9 11 14 6 4 12 7 ...
##  $ x3: num  10 8 13 9 11 14 6 4 12 7 ...
##  $ x4: num  8 8 8 8 8 8 8 19 8 8 ...
##  $ y1: num  8.04 6.95 7.58 8.81 8.33 ...
##  $ y2: num  9.14 8.14 8.74 8.77 9.26 8.1 6.13 3.1 9.13 7.26 ...
##  $ y3: num  7.46 6.77 12.74 7.11 7.81 ...
##  $ y4: num  6.58 5.76 7.71 8.84 8.47 7.04 5.25 12.5 5.56 7.91 ...
```

Central tendency measures

```r
mean(anscombe$x1)
```

```
## [1] 9
```

```r
mean(anscombe$x2)
```

```
## [1] 9
```

```r
mean(anscombe$x3)
```

```
## [1] 9
```

```r
mean(anscombe$x4)
```

```
## [1] 9
```

Central tendency measures

```
apply(anscombe[,1:4], 2, mean)
```

```
## x1 x2 x3 x4
##  9  9  9  9
```

```
apply(anscombe[,5:8], 2, mean)
```

```
##       y1       y2       y3       y4
## 7.500909 7.500909 7.500000 7.500909
```

```
apply(anscombe, 2, var)
```

```
##        x1        x2        x3        x4        y1        y2        y3        y4
## 11.000000 11.000000 11.000000 11.000000  4.127269  4.127629  4.122620  4.123249
```

Correlations

```
cor(anscombe$x1, anscombe$y1)
```

```
## [1] 0.8164205
```

```
cor(anscombe$x2, anscombe$y2)
```

```
## [1] 0.8162365
```

```
cor(anscombe$x3, anscombe$y3)
```

```
## [1] 0.8162867
```

```
cor(anscombe$x4, anscombe$y4)
```

```
## [1] 0.8165214
```

Linear regression parameters

- remember a linear regression: **y = a + bx**, where a is the intercept and b is the slope

```
m1 <- lm(anscombe$y1 ~ anscombe$x1)
m2 <- lm(anscombe$y2 ~ anscombe$x2)
m3 <- lm(anscombe$y3 ~ anscombe$x3)
m4 <- lm(anscombe$y4 ~ anscombe$x4)

coef(m1)
```

```
## (Intercept) anscombe$x1
##   3.0000909   0.5000909
```

```
coef(m2)
```

```
## (Intercept) anscombe$x2
##    3.000909    0.500000
```

Linear regression coefficients

```
mlist <- list(m1, m2, m3, m4)
lapply(mlist, coef)
```

```
## [[1]]
## (Intercept) anscombe$x1
##   3.0000909   0.5000909
##
## [[2]]
## (Intercept) anscombe$x2
##    3.000909    0.500000
##
## [[3]]
## (Intercept) anscombe$x3
##   3.0024545   0.4997273
##
## [[4]]
## (Intercept) anscombe$x4
##   3.0017273   0.4999091
```

Let's plot the Anscombe data

```
#par(mfrow = c(2,2),
#    las = 1,
#    bty = "l")
plot(anscombe$y1 ~ anscombe$x1)
abline(mlist[[1]])
plot(anscombe$y2 ~ anscombe$x2)
abline(mlist[[2]])
plot(anscombe$y3 ~ anscombe$x3)
abline(mlist[[3]])
plot(anscombe$y4 ~ anscombe$x4)
abline(mlist[[4]])
#par(mfrow=c(1, 1))
```

one example EDA workflow

```r
data(iris)
#head(iris)
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

how many observations do we have?

```
table(iris$Species)
plot(iris$Species) #barplot is the default funciton when you plot a categorical va
```

central tendency measures

```r
mean(iris$Sepal.Length)
median(iris$Sepal.Length)

## for each species:
tapply(X = iris$Sepal.Length,
       INDEX = iris$Species,
       FUN = mean)

tapply(X = iris$Sepal.Length,
       INDEX = iris$Species,
       FUN = median)
```

central tendency measures

```r
freqf <- sort(table(iris$Sepal.Length),
              decreasing = TRUE)
freqf[1] #the most common value (mode) is 5, it appears 10 times
```

data dispersion measures

```r
range(iris$Sepal.Length)
```

```
## [1] 4.3 7.9
```

```r
diff(range(iris$Sepal.Length))
```

```
## [1] 3.6
```

data dispersion measures

- variance, standard deviation

```
var(iris$Petal.Length) # variance
sd(iris$Petal.Length) #standard deviation
sd(iris$Petal.Length) / mean(iris$Petal.Length) * 100 # variation coefficient
```

data dispersion measures

- for each species?

```
tapply(X = iris$Sepal.Length, INDEX = iris$Species, FUN = sd)
tapply(X = iris$Sepal.Width, INDEX = iris$Species, FUN = sd)
```

data distribution: quantiles and inter-quantile range (IQR)

```
quantile(iris$Petal.Length) #quantiles
```

```
##   0%  25%  50%  75% 100% 
## 1.00 1.60 4.35 5.10 6.90
```

```
quantile(iris$Petal.Length, probs = c(0.05, 0.5, 0.95)) #other quantiles
```

```
##   5%  50%  95% 
## 1.30 4.35 6.10
```

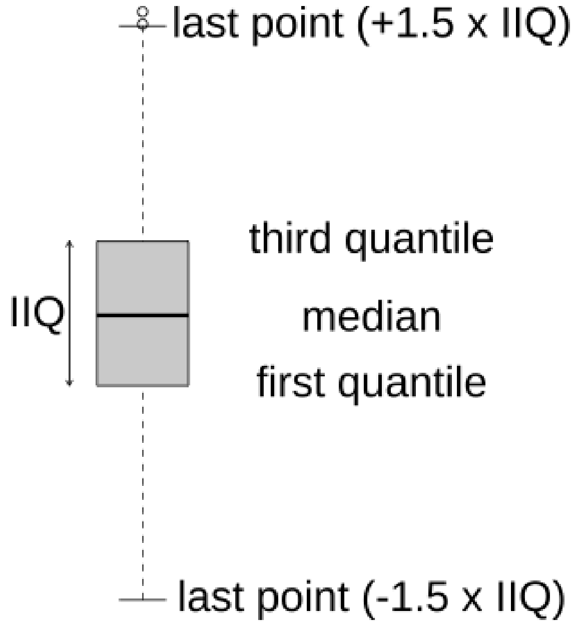
```
IQR(iris$Petal.Length) #inter-quantile range
```

```
## [1] 3.5
```

```
summary(iris$Petal.Length)
```

data distribution: boxplot

```
boxplot(iris$Petal.Length)
```

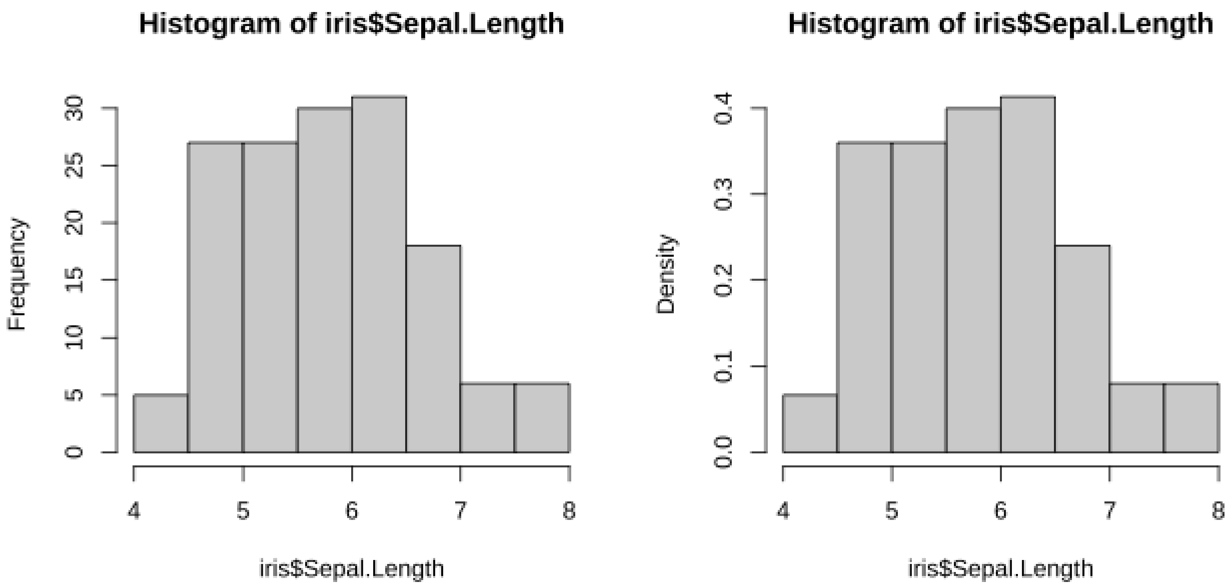


histogram

```
hist(iris$Sepal.Width)
hist(iris$Sepal.Length)
hist(iris$Petal.Length)
```

histogram types

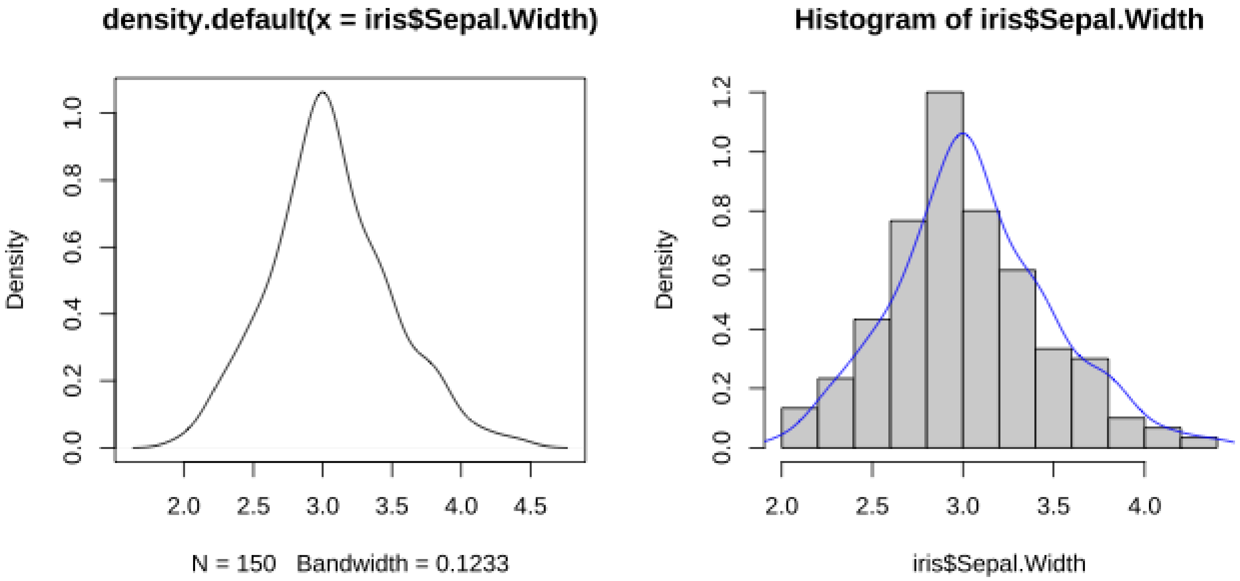
```r
par(mfrow = c(1,2))
hist(iris$Sepal.Length)
hist(iris$Sepal.Length, probability = TRUE) # empirical probabilistic density curv
```



```r
par(mfrow = c(1,1))
```

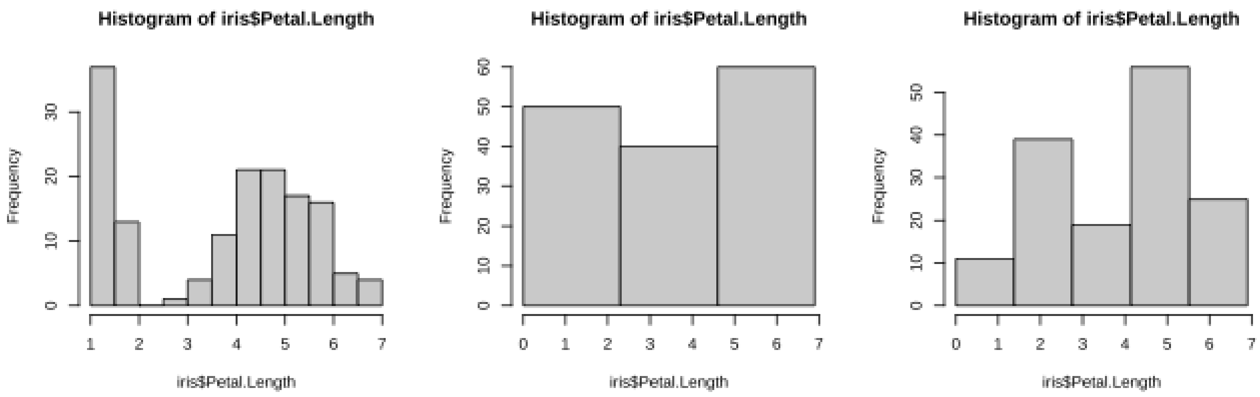
histogram types

```
par(mfrow = c(1,2))
plot(density(iris$Sepal.Width))
hist(iris$Sepal.Width, probability = TRUE) # empirical probabilistic density curve
lines(density(iris$Sepal.Width), col="blue")
```



histogram breaks

```
par(mfrow = c(1,3))
hist(iris$Petal.Length)
hist(iris$Petal.Length,
     breaks = seq(0, max(iris$Petal.Length), length = 4))
hist(iris$Petal.Length,
     breaks = seq(0, max(iris$Petal.Length), length = 6))
```



```
par(mfrow = c(1,1))
```

relationships between variables: scatterplot

```
x <- iris$Petal.Length
y <- iris$Petal.Width
plot(x, y)
```

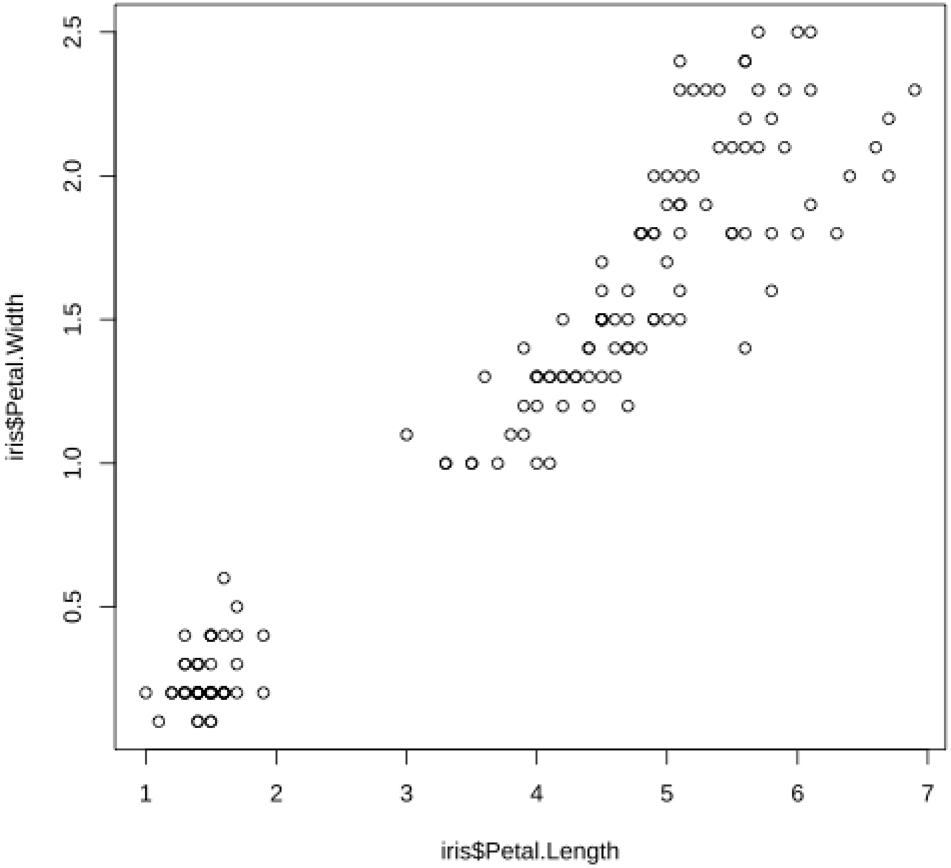
relationship between variables: correlation

```
cor(x, y)
```

```
## [1] 0.9628654
```

- when is a correlation high? (~0.7?)

let's go back to out scatterplot



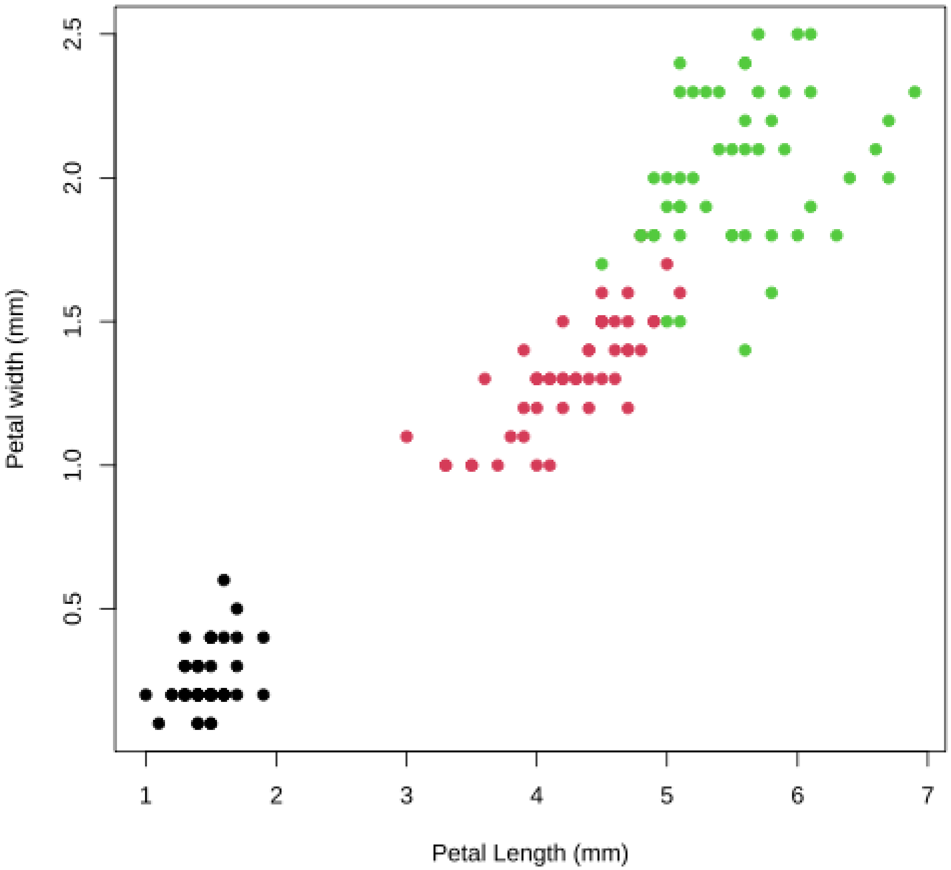
plotting basics

- All parameters for plotting are in function `par()`
- `pch`, `cex`, `xlab`, `ylab`, `las`
- `par(mfrow = c(1, 2))`

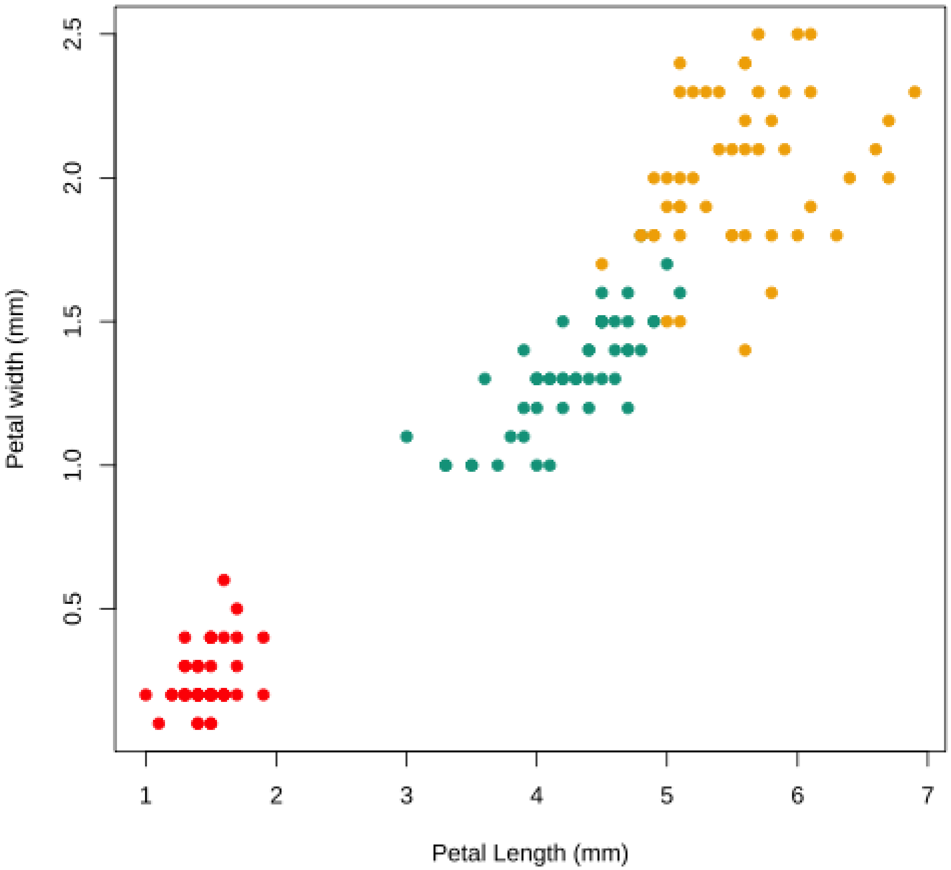
let's go back to out scatterplot

```
plot(iris$Petal.Length, iris$Petal.Width)
plot(iris$Petal.Length, iris$Petal.Width,
     xlab = "Petal Length (mm)",
     ylab = "Petal width (mm)", pch = 19)
lmod <- lm(Petal.Width ~ Petal.Length, data = iris)
coef(lmod)
abline(lmod, col = "red")
```

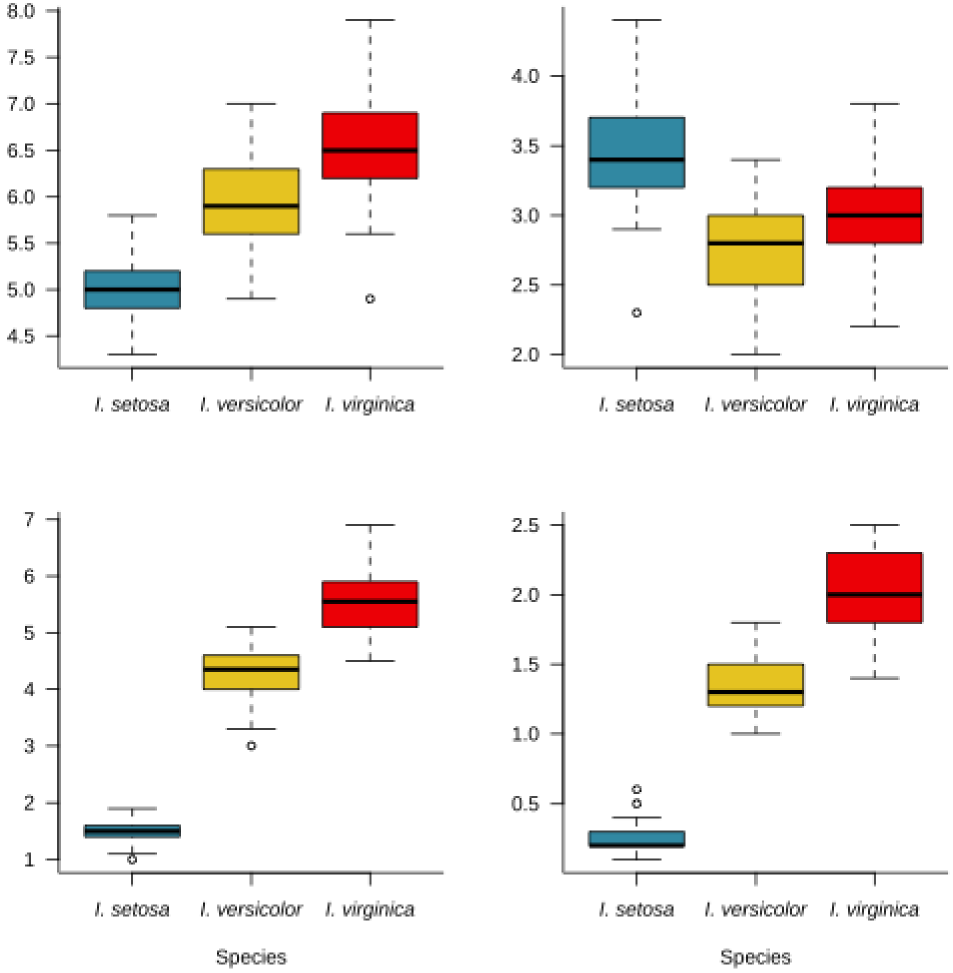
what about species?



what about species?



let's go back to boxplots



plot devices

- `plot()` opens a new graphic "device": a new window
- `hist()`, `barplot()`, `boxplot()` also call for new devices
- `points()` and `lines()` do not open new devices and need to be executed after `plot()` calls
- new `plot()` calls reset the graphic device.
- `dev.off()` turns off the current plot device

saving plots

- to save plots in base R, new graphic devices must be called: `png()`, `pdf()`, etc- (check `capabilities()`)
- basic recommended formats: **.png** and **.tiff** because they are not lossy (try not to use **.jpeg**)
- `png()` calls for a new graphic device *different from the graphic window*
    - plot code
- `dev.off()` to close the device and save.

you won't see the plot when you do that

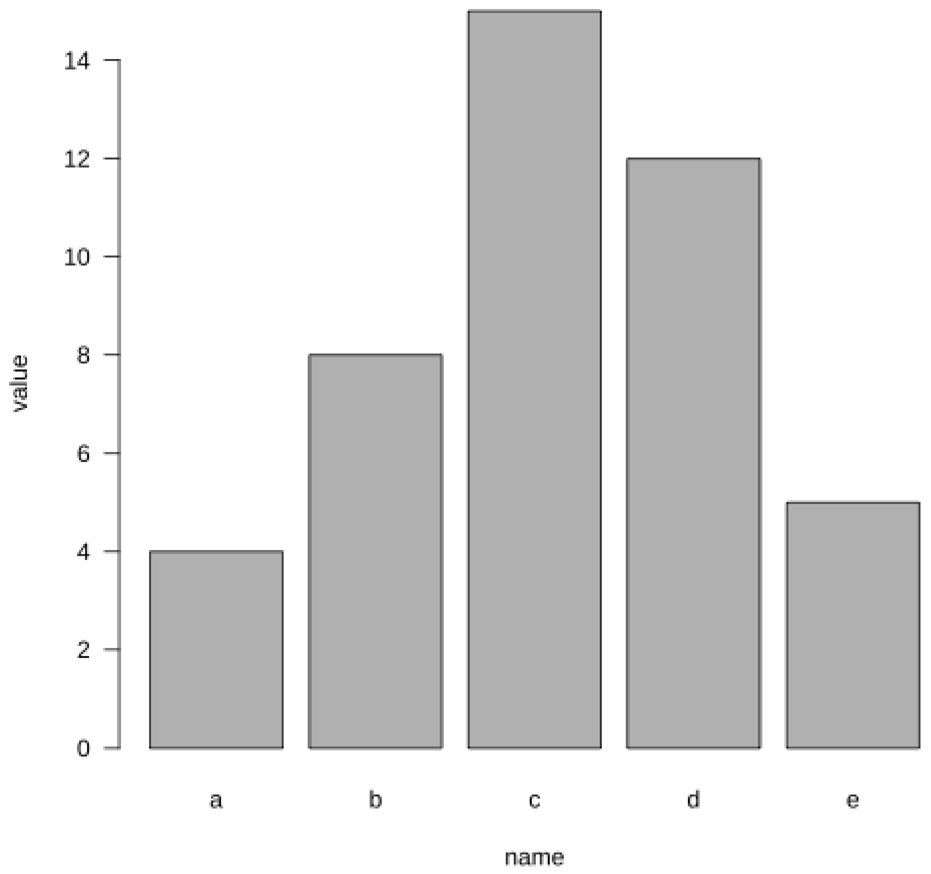
saving our plot

data visualization has many don'ts

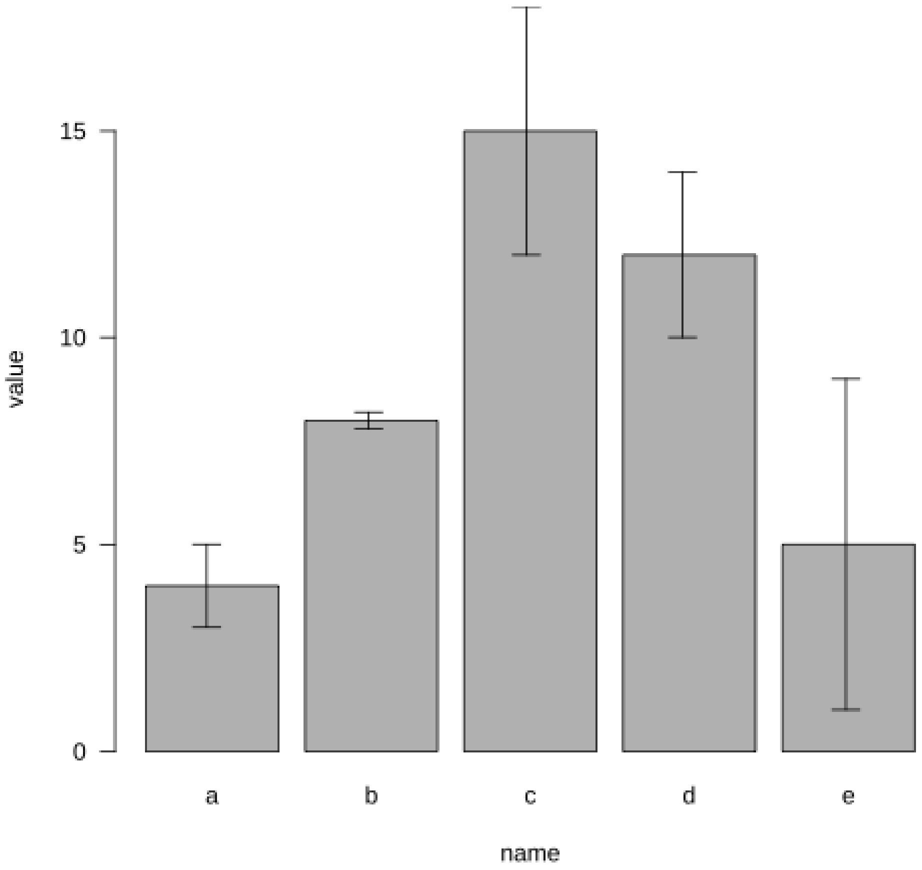
many

there's always a better option than a pie chart

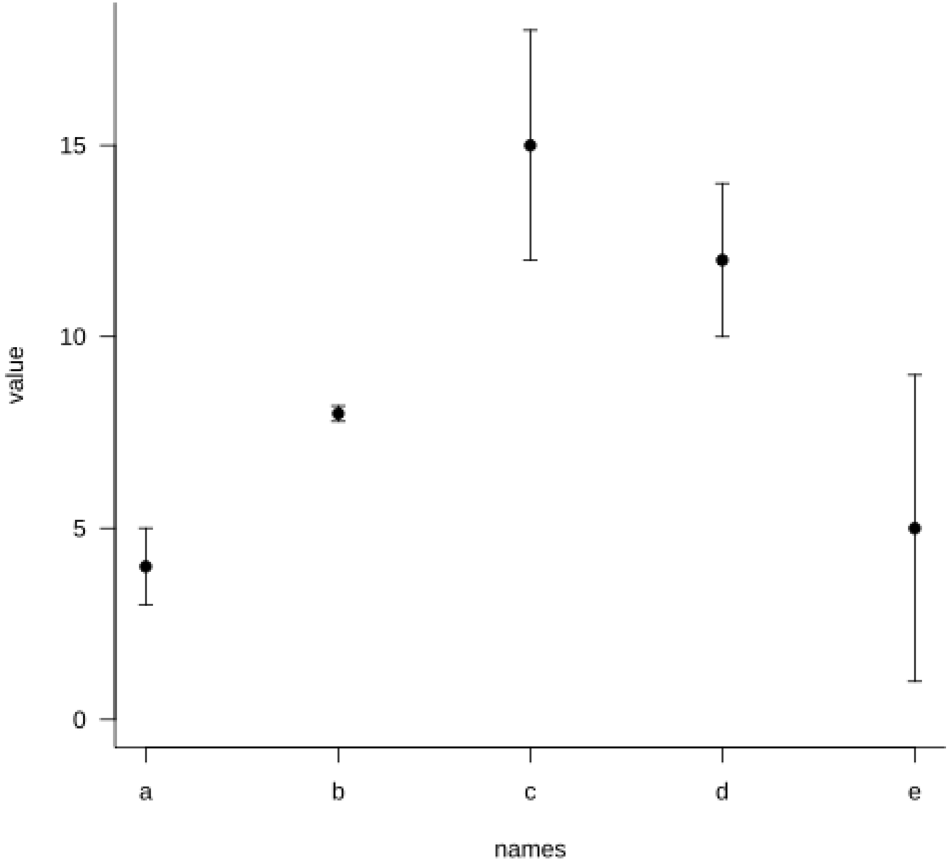
barplots are not always very informative



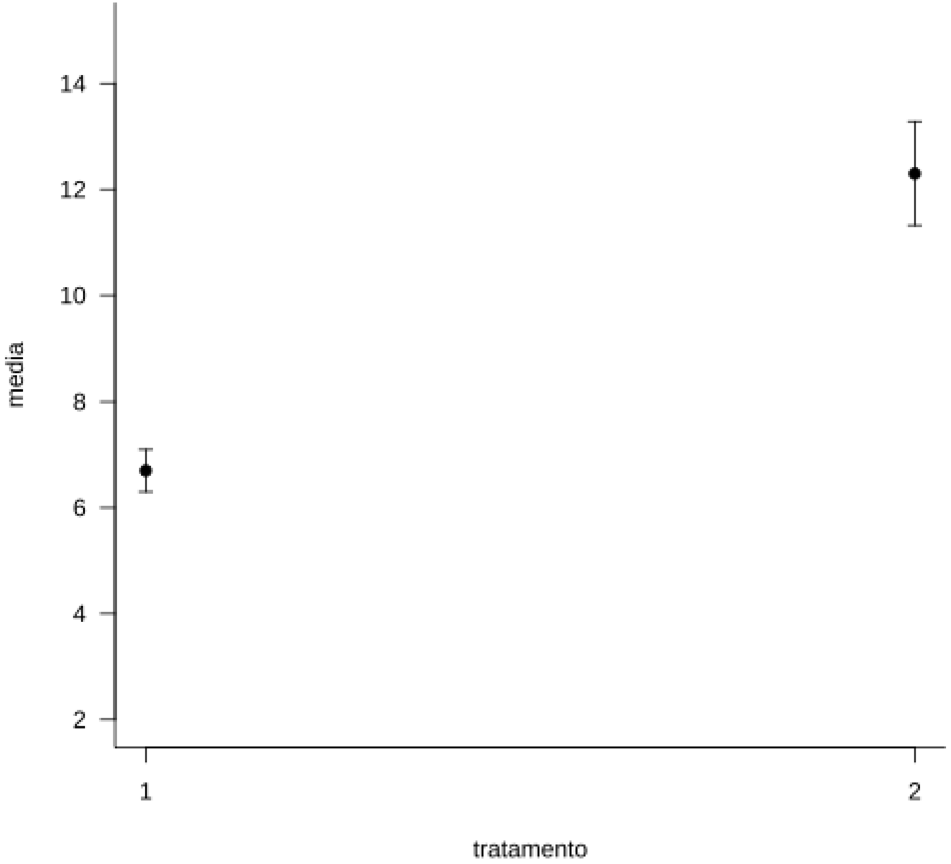
better with error bars...



but maybe don't even make a barplot



...or maybe don't even make a graph



make a table or say it in the text

Treatment	Effect
1	6.7 ± 0.4
2	12.3 ± 0.98

some basic tips in general

- only make plots when you really need to
- don't spend more ink and colors than you need to
- don't fool your reader (no y-axis tampering, no undue transformation)
- show error measures

some basic tips in R

- use `las = 1` for your axis labels
- use `bty = "l"` for your boxes
- change to at least `pch = 19`
- use `xlab` and `ylab`
- save to png and pdf formats

Statistical procedures: package stats

- Linear regression: `lm()`
- Analysis of variance: `anova()`, `aov()`
- t-tests: `t.test()`
- p-values correction: `p.adjust()`

R TASKVIEWS https://cran.r-project.org/web/views/