Assignment: HWA01

Sean Paolo

Wed 07 Feb 2024

General Template Instuctions

To complete and submit such type of exercises, please do the following: Your Homework Assignments (HWAs) can be written and submitted either as **.Rmd** (reproducible files) or as documents that combine text with code (in a **.PDF** format). Submit **always** your **PDF** and optionally the **Rmd** (if you encounter any issues or you have a special reason to do so).

A. A simple .Rmd template is provided here.

B. Alternatively, open a plain R script and save it as for example LAST_First_HWA##_ddmmyyyy.R. Keep short name-versions.

Also enter the current assignment (e.g., HWA01), your name, and the current date at the top of your document. When working on a task, always indicate which task you are answering with appropriate comments.

Edit your ANSWERS whenever is asked, namely write the script, run the chuck and type a brief reply to the question asked using the output.

Apply appropriate modifications withing the chunks below in order to get the output!

Here is an example how your file JACKSON_Jane_HWA01_08022021.Rmd could look:

```r
# ~~~~~~~~~~~~~~~~~~~~~~~~~~
# Exercise 1:

# Adding numbers:
1 + 2
```

```
## [1] 3
```

```r
# ~~~~~~~~~~~~~~~~~~~~~~~~~~
# Exercise 2:

# Draw 100 samples from a standard normal distribution:
x <- rnorm(100)

# Conduct a t-test on the sample:
t.test(x)
```

```
##
##  One Sample t-test
##
## data:  x
```

```
## t = 0.4006, df = 99, p-value = 0.6896
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1496979  0.2254337
## sample estimates:
## mean of x
## 0.0378679
```

```
# etc. ...
```

C. Complete as many exercises as you can.

D. Upload your *PDF* assignment (and Rmd script only when asked) on Teams. If any problems send it to parentheses, like this email thanos.manos@cyu.com.

IMPORTANT NOTEs:

(a) I am not suppose to run your script to evaluate your HWAs. The code must appear inside each respective chunk accompanied by the output!

(b) Make sure your chunks work fine. You can run them one chunk at at time when working, debugging etc. If you clock "*knit*", you run the whole script and at times it will be time consuming and unnecessary.

(c) Mind the *line-breaking* when you prepare your PDF!

E. Your final files should include both the commands script and the output. Hence, modify the corresponding fields below accordingly.

Creating and evaluating objects

1. Let's see how we interact with R by creating some simple objects and applying basic functions to them:

1a. R can be used to create (e.g., numeric) objects and evaluate them, as with any regular calculator:

```
a <- 1 # assigns "1" to an object a
b <- 2 # assigns "2" to an object b
a + b  # applies "+" to a and b, and prints the result
```

```
## [1] 3
```

```
sum(a, b) # applies the function sum() to a and b, and prints the result
```

```
## [1] 3
```

```
a <- 100 # to change an object, it must be re-assigned
a + b
```

```
## [1] 102
```

```
sum(a, b)
```

```
## [1] 102
```

```r
# Note that evaluating
# A + B
# would yield an Error, as R is case-sensitive!
```

1b. Creating some vectors:

```r
x <- 1:10 # creates a sequence of numbers (integers from 1 to 10) and assigns it to a vector x
y <- 10:1
x + y # applies "+" to each element of the vectors x and y, and prints the result
```

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

```r
z <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) # a generic way to create a vector
w <- c("a", "vector", "of", "characters")
x == z # applies "==" to each element of the vectors x and z
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```r
x[3] # returns the 3rd element of x
```

```
## [1] 3
```

```r
y[3] # ...
```

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

```r
w[3:4] # ...
```

```
## [1] "of"         "characters"
```

```r
x > 5 # applies "+" to each element of the vectors x and y, and prints the result
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
```

```r
x[x > 5] # ...
```

```
## [1]  6  7  8  9 10
```

1c. Basic sampling:

```r
sample(c(0, 1), size = 1) # randomly draws 1 sample from c(0, 1)
```

```
## [1] 0
```

```r
coin = c("heads", "tails") # defines the outcomes of a coin
# sample(coin, size = 10) # tries to randomly draws 10 samples from our coin (flip it 10 times), but ..
sample(coin, size = 10, replace = TRUE) # ... works!
```

```
##  [1] "tails" "heads" "heads" "heads" "heads" "tails" "heads" "tails" "tails"
## [10] "tails"
```

```r
# Randomly assigning students to groups:
n.students <- 16
n.groups <- 4
groups <- 1:n.groups
sample(groups, size = n.students, replace = TRUE)
```

```
##  [1] 4 1 3 3 3 1 4 3 2 2 3 1 1 2 2 4
```

Installing and loading packages

R is not just a single program, but an entire universe of code from a community of developers (with all the benefits and costs of such diversity). Packages allow to import and use R code of other people.

2. Let's install and load the yarrr package (by Nathaniel Phillips) that contains many datasets (like pirates) and functions (like pirateplot).

2a. Installing a package:

```r
#install.packages("yarrr") # installs a package
############################################################
### REMEMBER: You only need to install a package ONCE!   ###
####
```

2b. Loading a (previously installed) package:

```r
library("yarrr")          # loads a package
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
## "ndiMatrix" of class "replValueSp"; definition not updated
```

Exploring a dataset

3. The pirates dataset included in the yarrr package contains data from a survey of 1,000 pirates.

3a. Get basic information on the pirates dataset:

```r
?pirates
```

3b. How many rows and columns are in this dataset?

```r
nrow(pirates) # number of rows / cases
```

```
## [1] 1000
```

```r
ncol(pirates) # number of columns / variables
```

```
## [1] 17
```

3c. View the first few rows of the pirates dataset:

```r
head(pirates)
```

```
##   id    sex age height weight headband college tattoos tchests parrots
## 1  1   male  28 173.11   70.5      yes   JSSFP       9       0       0
## 2  2   male  31 209.25  105.6      yes   JSSFP       9      11       0
## 3  3   male  26 169.95   77.1      yes    CCCC      10      10       1
## 4  4 female  31 144.29   58.5       no   JSSFP       2       0       2
## 5  5 female  41 157.85   58.4      yes   JSSFP       9       6       4
## 6  6   male  26 190.20   85.4      yes    CCCC       7      19       0
##   favorite.pirate sword.type eyepatch sword.time beard.length
## 1    Jack Sparrow    cutlass        1       0.58           16
## 2    Jack Sparrow    cutlass        0       1.11           21
## 3    Jack Sparrow    cutlass        1       1.44           19
## 4    Jack Sparrow   scimitar        1      36.11            2
## 5            Hook    cutlass        1       0.11            0
## 6    Jack Sparrow    cutlass        1       0.59           17
##             fav.pixar grogg
## 1      Monsters, Inc.    11
## 2              WALL-E     9
## 3          Inside Out     7
## 4          Inside Out     9
## 5          Inside Out    14
## 6 Monsters University     7
```

3d. Show the structure of the pirates dataset:

```r
str(pirates)
```

```
## 'data.frame':    1000 obs. of  17 variables:
##  $ id             : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ sex            : chr  "male" "male" "male" "female" ...
##  $ age            : num  28 31 26 31 41 26 31 31 28 30 ...
##  $ height         : num  173 209 170 144 158 ...
##  $ weight         : num  70.5 105.6 77.1 58.5 58.4 ...
##  $ headband       : chr  "yes" "yes" "yes" "no" ...
##  $ college        : chr  "JSSFP" "JSSFP" "CCCC" "JSSFP" ...
##  $ tattoos        : num  9 9 10 2 9 7 9 5 12 12 ...
##  $ tchests        : num  0 11 10 0 6 19 1 13 37 69 ...
##  $ parrots        : num  0 0 1 2 4 0 7 7 2 4 ...
##  $ favorite.pirate: chr  "Jack Sparrow" "Jack Sparrow" "Jack Sparrow" "Jack Sparrow" ...
##  $ sword.type     : chr  "cutlass" "cutlass" "cutlass" "scimitar" ...
##  $ eyepatch       : num  1 0 1 1 1 1 0 1 0 1 ...
##  $ sword.time     : num  0.58 1.11 1.44 36.11 0.11 ...
##  $ beard.length   : num  16 21 19 2 0 17 1 1 1 25 ...
##  $ fav.pixar      : chr  "Monsters, Inc." "WALL-E" "Inside Out" "Inside Out" ...
##  $ grogg          : num  11 9 7 9 14 7 9 12 16 9 ...
```

3e. Show the entire dataset in a new window:

```r
View(pirates)
```

Basic descriptives for numeric vectors

4. To obtain basic descriptives for numeric data, you can apply in-built R functions to numeric vectors.

4a. What is the mean age? Apply the function mean() to the vector pirates$age:

```r
mean(pirates$age)
```

```
## [1] 27.36
```

4b. What is the tallest pirate? Apply the function max() to the vector pirates$height.

```r
max(pirates$height)
```

```
## [1] 209.25
```

Answer: The height of the tallest pirate is 209.25 and this is just a demonstration: 4.

4c. What was the mean and median weight of the pirates?

```r
mean(pirates$weight)
```

```
## [1] 69.7446
```

```r
median(pirates$weight)
```

```
## [1] 69.55
```

Answer: The mean and median weight of the pirates is 69.7446 and 69.55 respectively.

Basic descriptives for non-numeric data

5. Non-numeric data are typically summarized in frequency tables:

5a. How many pirates are there of each sex?

```r
table(pirates$sex)
```

```
##
## female   male  other
##    464    490     46
```

5b. How many pirates are there of each age?

6. To collapse cases over 2 variables, you can use the **aggregate()** function:

6a. What is the mean age for each sex?

```r
# There is an issue here:
#aggregate(formula = age ~ sex,
#          data = pirates,
#          FUN = mean)
```

6b. What is the mean beard length (beard.length) for each sex?

6c. How many pirates are wearing a headband? What is the median age of pirates for each combination of sex and headband?

```
table(pirates$headband)
```

```
##
##  no yes
## 113 887
```

```
#aggregate(formula = age ~ sex + headband,
#           data = pirates,
#           FUN = median)
```

Basic Plots

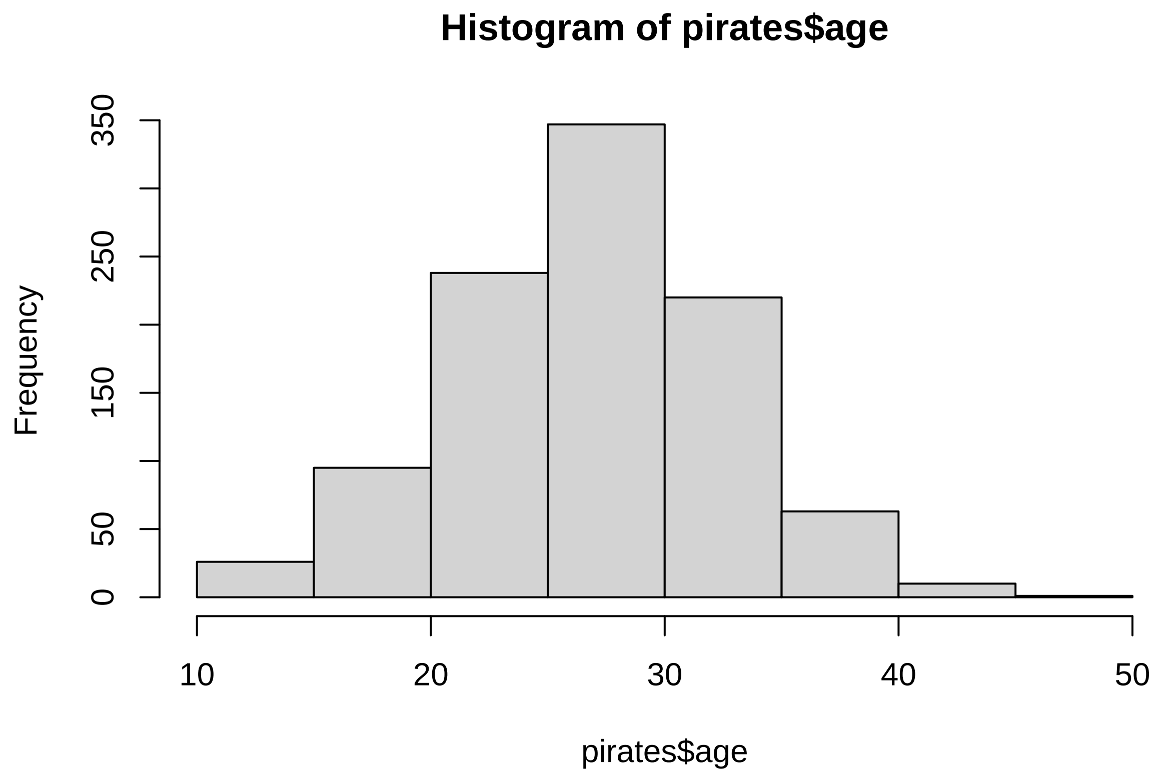
Let's explore some basic plotting commands!

Histograms

7. Basic histograms show some variable's distribution of values:

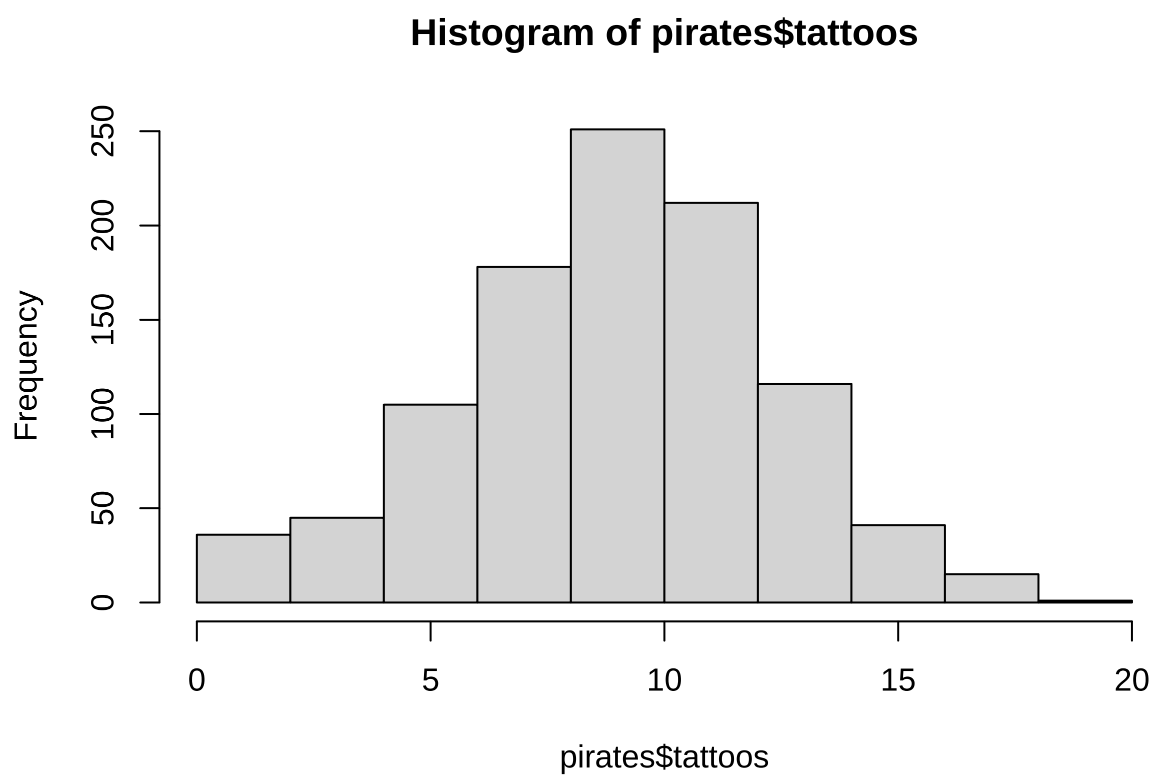
7a. What is the distribution of pirate ages?

```
hist(x = pirates$age)
```



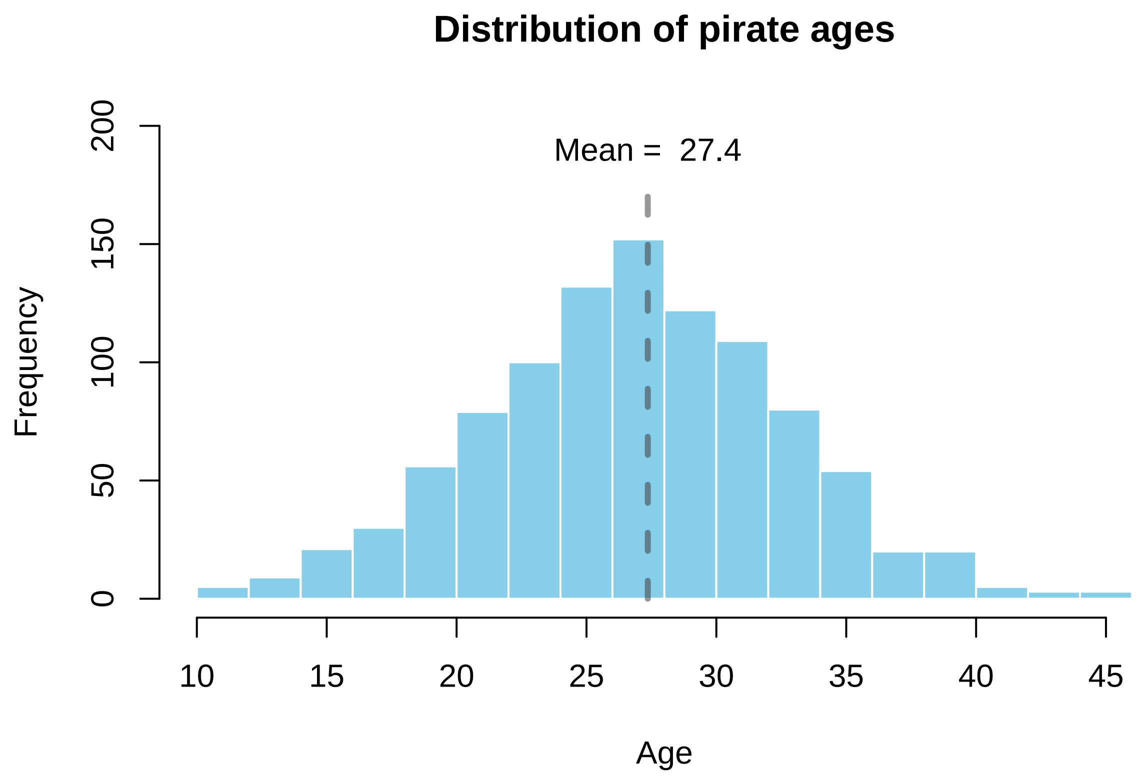
7b. What is the distribution of (the number of) pirate tattoos?

```r
hist(x = pirates$tattoos)
```



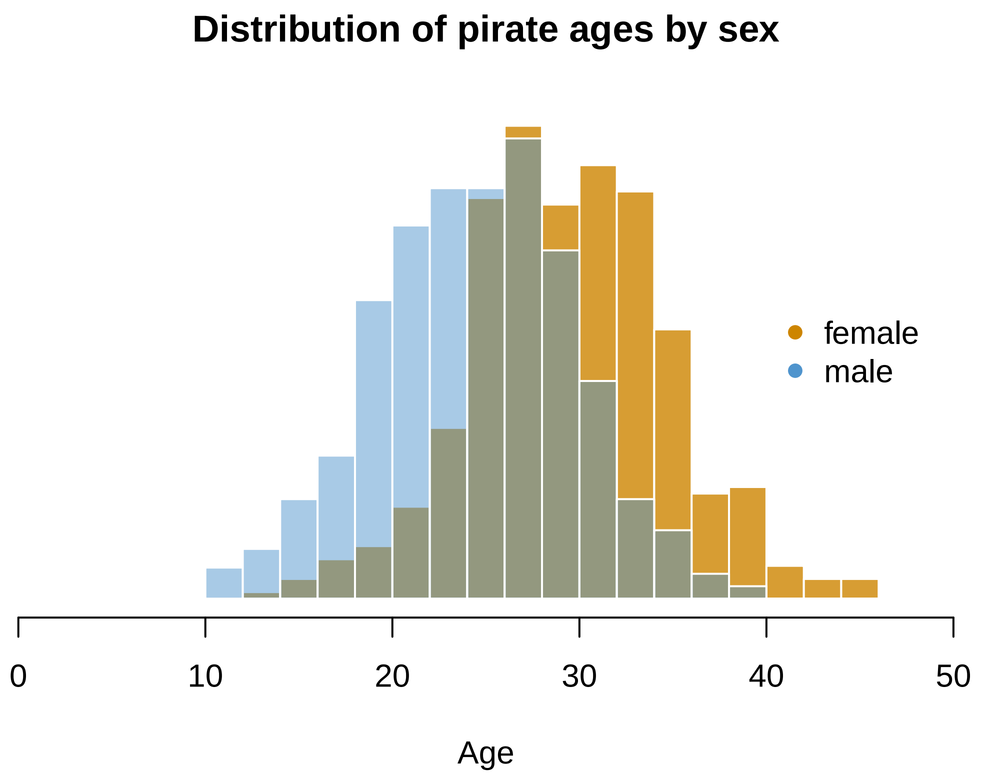
7c. To get more fancy histograms you can set and customize many parameters:

```r
ymax <- 200

hist(x = pirates$age,
     breaks = 15,
     main = "Distribution of pirate ages",
     col = "skyblue",
     border = "white",
     xlab = "Age",
     ylim = c(0, ymax))

# Add the mean as a text label:
text(x = mean(pirates$age), y = (ymax - 10),
     labels = paste("Mean = ", round(mean(pirates$age), 1)))

# Add a vertical dashed line at the mean:
segments(x0 = mean(pirates$age), y0 = 0,
         x1 = mean(pirates$age), y1 = (ymax - 20),
         col = gray(.2, .5),
         lwd = 3,
         lty = 2)
```



7d. Combining multiple histograms:

```
## 2 overlapping histograms of pirate ages for females and males:

# (a) Start with the female data:
hist(x = pirates$age[pirates$sex == "female"],
     main = "Distribution of pirate ages by sex",
     col = transparent("orange3", .2),
     border = "white",
     xlab = "Age",
     breaks = seq(0, 50, 2),
     probability = TRUE,
     ylab = "",
     yaxt = "n")

# (b) add male data:
hist(x = pirates$age[pirates$sex == "male"],
     add = TRUE,
     probability = TRUE,
     border = "white",
     breaks = seq(0, 50, 2),
     col = transparent("steelblue3", .5))

# (c) add a legend:
legend(x = 40,
       y = .05,
       col = c("orange3", "steelblue3"),
       legend = c("female", "male"),
       pch = 16,
       bty = "n")
```

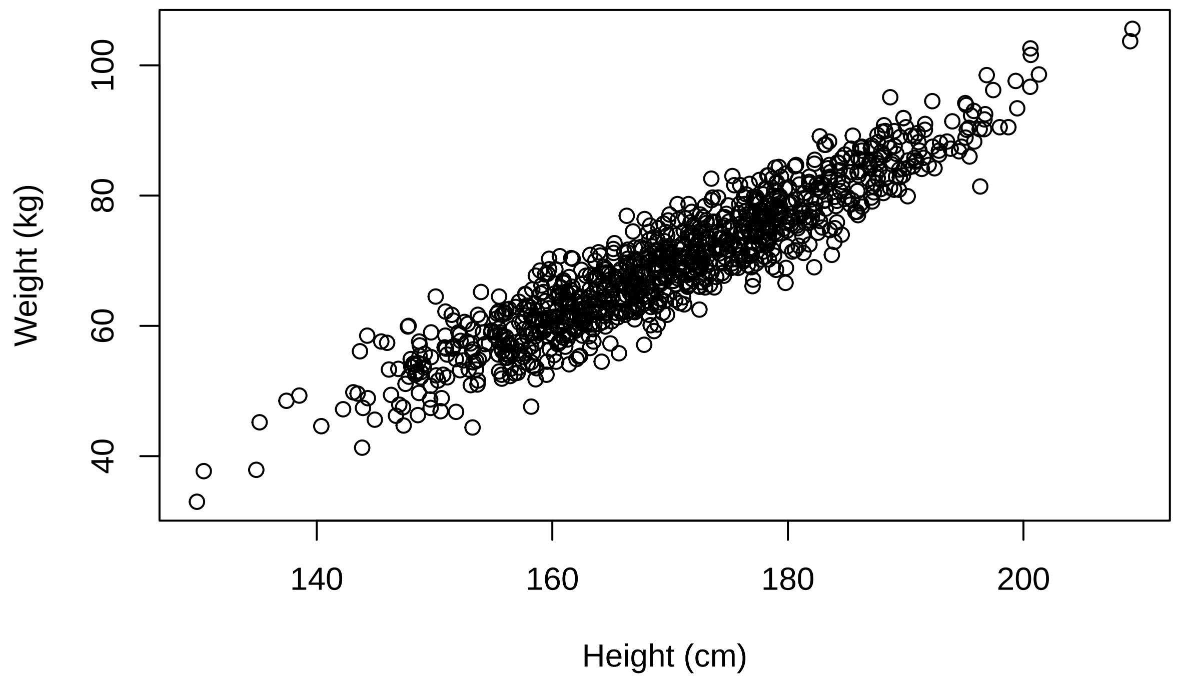


Scatterplots

8. Scatterplots show relations between 2 numeric variables:

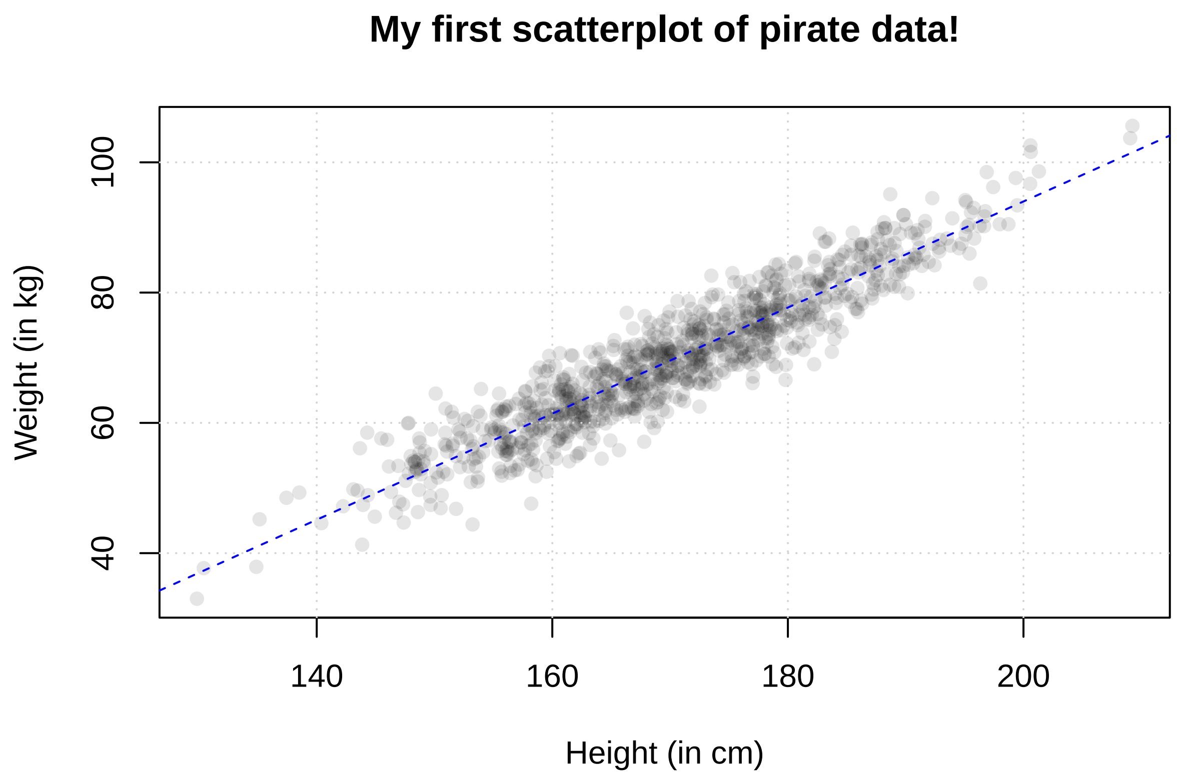
8a. Basic scatterplot of height and weight of pirates:

```
## 8A: A simple scatterplot of pirate height and weight
plot(x = pirates$height,
     y = pirates$weight,
     xlab = "Height (cm)",
     ylab = "Weight (kg)")
```



8b. A fancier scatterplot of the same data with some additional arguments:

```
# Create main plot:
plot(x = pirates$height,
     y = pirates$weight,
     main = 'My first scatterplot of pirate data!',
     xlab = 'Height (in cm)',
     ylab = 'Weight (in kg)',
     pch = 16,     # filled circles
     col = gray(0, .1)) # transparent gray

# Add gridlines:
grid()

# Create a linear regression model:
model <- lm(formula = weight ~ height,
            data = pirates)

# Add regression line to the plot:
abline(model,
       col = 'blue', lty = 2)
```

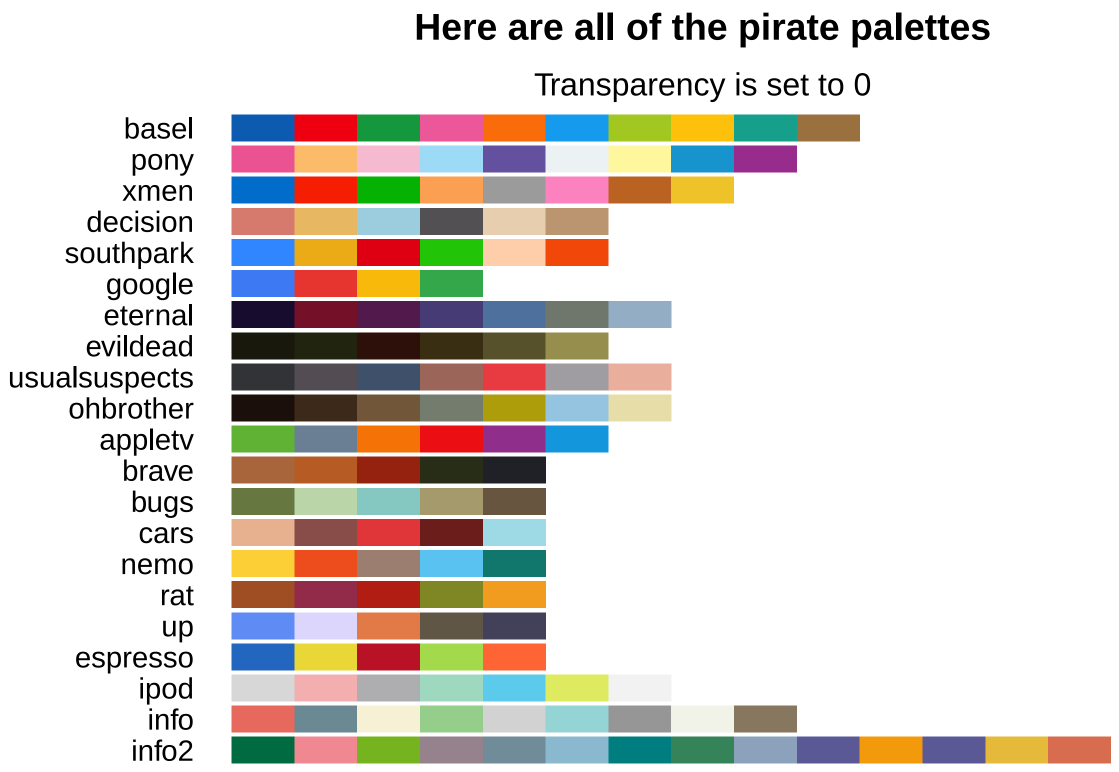


Color palettes

9. To obtain prettier colors, the yarrr package offers some pre-designed color palettes:

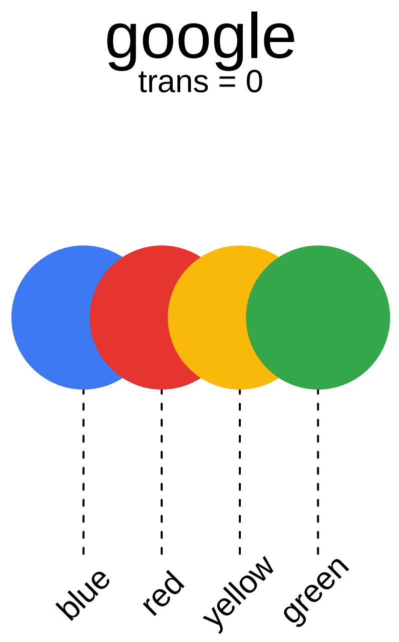
9a. Look at all the available palettes from piratepal():

```r
piratepal()
```



9b. Look at some specific palette in more detail:

```r
piratepal(palette = "google", plot.result = TRUE)
```



9c. Look at some other palettes in more detail.

9d. Using the pony palette in a fancy scatterplot of pirate height and weight:

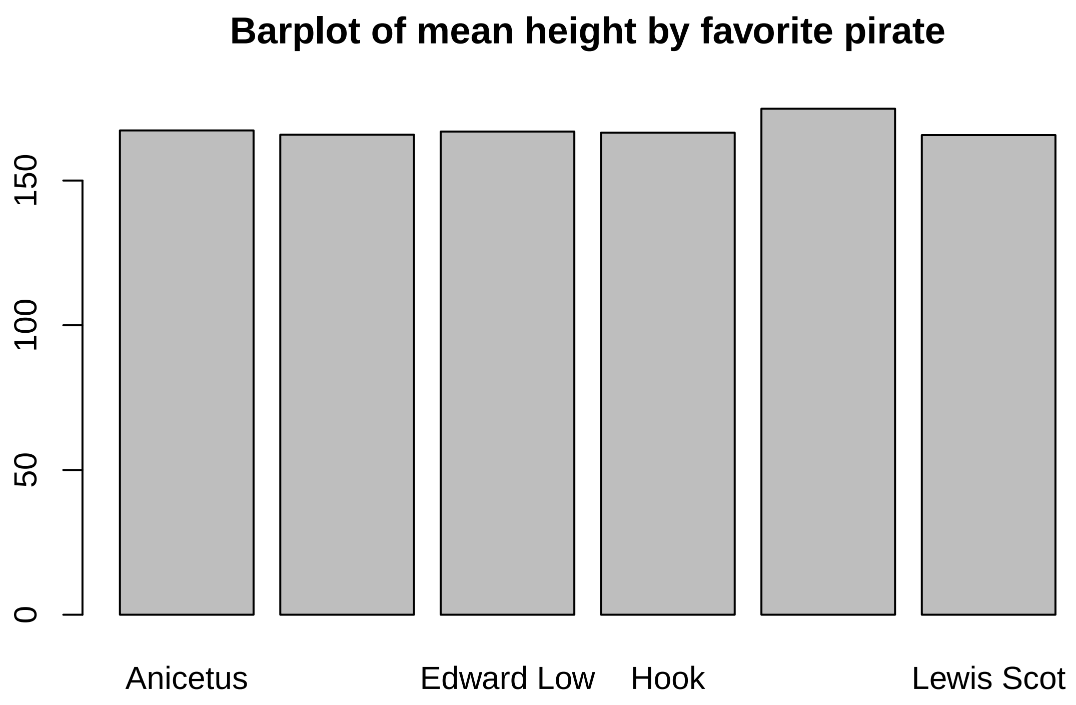
```r
my.cols <- piratepal(palette = "pony",
                     trans = .2,
                     length.out = nrow(pirates))

# Create the plot:
plot(x = pirates$height, y = pirates$weight,
     main = "Random scatterplot with My Little Pony Colors",
     xlab = "Pony height",
     ylab = "Pony weight",
     pch = 21,  # Round symbols with borders
     cex = 2,  # magnifying factor of plot text and symbols
     col = "white",  # white border
     bg = my.cols,   # random colors
     bty = "n"       # no plot border
)

# Add gridlines:
grid()
```

Barplots

Barplots allow comparisons between categories of a variable:

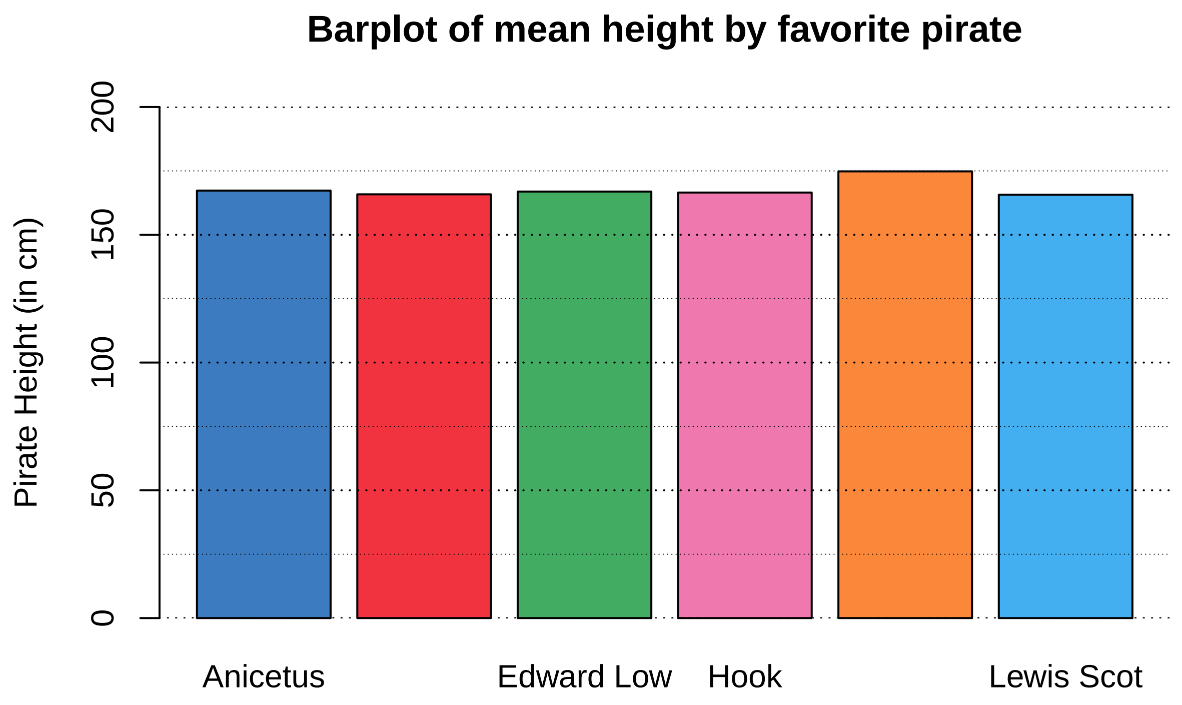
10.a Calculate mean height for each favorite.pirate:

```r
pirate.heights <- aggregate(height ~ favorite.pirate,
                            data = pirates,
                            FUN = mean)

barplot(pirate.heights$height,
        main = "Barplot of mean height by favorite pirate",
        names.arg = pirate.heights$favorite.pirate)
```



10b. The same barplot, but with additional customizations:

```r
barplot(pirate.heights$height,
        ylim = c(0, 200),
        ylab = "Pirate Height (in cm)",
        main = "Barplot of mean height by favorite pirate",
        names.arg = pirate.heights$favorite.pirate,
        col = piratepal("basel", trans = .2))

abline(h = seq(0, 200, 25), lty = 3, lwd = c(1, .5))
```



Pirateplots

11.a A so-called pirateplot shows the raw values, means and distributions of a numeric variable (like height) by the levels of some categorical variables (like favorite.pirate):

```r
pirateplot(formula = height ~ favorite.pirate,
           data = pirates,
           main = "Pirateplot of height by favorite pirate")
```

11b. Create a pirateplot of height by sex and eyepatch.

```r
pirateplot(formula = height ~ pirates$sex,
           data = pirates,
           main = "Pirateplot of height by sex pirate")
```

```r
pirateplot(formula = height ~ pirates$eyepatch,
           data = pirates,
           main = "Pirateplot of height by eyepatch pirate")
```