HARVARD X - FINAL ASSESSMENT SECTION 1

Notebook by Isabella Heder

Course: Data Science by Harvard X on EDX

Platform used to run the codes and import libraries and datasets:

–> **RStudio**

```
library(gtools)
library(tidyverse)
```

LIBRARIES:

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.2
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

EXERCISES:

EXERCISE 1)

In the 200m dash finals in the Olympics, 8 runners compete for 3 medals (order matters). In the 2012 Olympics, 3 of the 8 runners were from Jamaica and the other 5 were from different countries. The three medals were all won by Jamaica (Usain Bolt, Yohan Blake, and Warren Weir).

Use the information above to help you answer the following four questions.

A) How many different ways can the 3 medals be distributed across 8 runners? And how many different ways can the three medals be distributed among the 3 runners from Jamaica?

Code:

```
medals <- permutations(8,3)
nrow(medals)
```

```
## [1] 336
```

For the first part of this exercise, I used the permutation function because, in this case, order matters;

Each medal is different from the other, so a person can get a Gold medal, which would be different from getting a Silver medal.

“nrow” counts the number of rolls –> and that is the number of possible ways of distributing the medals;

Code:

```
jamaica <- permutations(3, 3)
nrow(jamaica)
```

```
## [1] 6
```

For the second part of the exercise I used the same thinking as the previous one, but for this case, the number of medals is equal to the number of runners, so 3x3 would be equivalent to all the different ways the medals could be distributed.

C) What is the probability that all 3 medals are won by Jamaica?

SOLVING BY LOGIC: 6/336 → 1/56 → 0.0178

For the logic solution I just used the two information I gathered before, and divided one by the other to get the probability.

For the code, I used the **same** logic using the nrow function.

```
nrow(jamaica)/nrow(medals)
```

```
## [1] 0.01785714
```

MONTE CARLO:

D) Run a Monte Carlo simulation on this vector representing the countries of the 8 runners in this race:

```
runners <- c("Jamaica", "Jamaica", "Jamaica", "USA", "UK", "France", "Ireland", "Italy")

set.seed(1)
B <- 10000
monte_carlo <- replicate(B, {
  winners <- sample(runners, 3, replace = FALSE)
  all(winners == "Jamaica")})
mean(monte_carlo)
```

```
## [1] 0.0174
```

To create the Monte Carlo, I created the B variable for 10000 simulations (number commonly used at the course), and applied it to the Monte Carlo using the replicate function to run the simulation 10 thousand times.

The “winners” variable uses sample to randomly get 3 runners, not replacing them: meaning that a runner can´t get two medals.

Then, to see if all the three people to get the medal were Jamaicans, I used the “all( )” function, which checks if the expression is true or false.

Lastly, I used mean( ) to see the proportion of the 10k simulations that had the Jamaican runners as winners.

EXERCISE 2:

A restaurant manager wants to advertise that his lunch special offers enough choices to eat different meals every day of the year. He doesn't think his current special actually allows that number of choices, but wants to change his special if needed to allow at least 365 choices.

A meal at the restaurant includes 1 entree, 2 sides, and 1 drink. He currently offers a choice of 1 entree from a list of 6 options, a choice of 2 different sides from a list of 6 options, and a choice of 1 drink from a list of 2 options.

A) How many meal combinations are possible with the current menu?

```
6 * nrow(combinations(6,2)) * 2
```

```
## [1] 180
```

Knowing that the only rule is that the sides should be different from one another, I applied the combination function to gather all the possibilities for the sides, considering that each combo should not have the same ones on the same meal.

So, 6 choices for entrees, 15 for sides (result of the combination), and 2 for drinks == 180 different meals.

B) The manager has one additional drink he could add to the special. How many combinations are possible if he expands his original special to 3 drink options?

```
6 * 15 * 3
```

```
## [1] 270
```

6 choices for entrees, 15 for sides, and 3 for drinks == 270 different meals.

C) The manager decides to add the third drink but needs to expand the number of options. The manager would prefer not to change his menu further and wants to know if he can meet his goal by letting customers choose more sides.

How many meal combinations are there if customers can choose from 6 entrees, 3 drinks, and select 3 sides from the current 6 options?

```
nrow(combinations(6, 3))
```

```
## [1] 20
```

```
6 * 20 * 3
```

```
## [1] 360
```

Same logic, just changing the numbers for the combination function.

D) The manager is concerned that customers may not want 3 sides with their meal. He is willing to increase the number of entree choices instead, but if he adds too many expensive options it could eat into profits. He wants to know how many entree choices he would have to offer in order to meet his goal.

- **Write a function that takes a number of entree choices and returns the number of meal combinations possible given that number of entree options, 3 drink choices, and a selection of 2 sides from 6 options.**

- **Use sapply() to apply the function to entree option counts ranging from 1 to 12.**

What is the minimum number of entree options required in order to generate more than 365 combinations?

```
n_entrees <- function(N) {
    result <- 15 * 3 * N
    result }
N <- seq(1, 12)
n_entrees_apply <- sapply(N, n_entrees)
n_entrees_apply
```

```
##  [1]  45  90 135 180 225 270 315 360 405 450 495 540
```

My answer: 9

For the range, I created the N value (1:12).

Next, I created the function with the "result" variable, which multiplies the meal options just like I did previously.

The only difference is that it takes the N value as the number of entree options.

Because the manager wants 365 combinations, the answer is 9 (405 options for meals).

E) The manager isn't sure he can afford to put that many entree choices on the lunch menu and thinks it would be cheaper for him to expand the number of sides. He wants to know how many sides he would have to offer to meet his goal of at least 365 combinations.

- **Write a function that takes a number of side choices and returns the number of meal combinations possible given 6 entree choices, 3 drink choices, and a selection of 2 sides from the specified number of side choices.**
- **Use sapply() to apply the function to side counts ranging from 2 to 12.**

What is the minimum number of side options required in order to generate more than 365 combinations?

```
side_choices <- function(n) {
   6 * nrow(combinations(n, 2)) * 3 }
meals <- sapply(2:12, side_choices)
meals
```

```
##  [1]   18   54  108  180  270  378  504  648  810  990 1188
```

My answer: 7, because it starts from 2.

For this exercise, I created the function containing the same formula I previously used, but for this time, knowing that the number of options for sides are not 100% defined yet, I set the 'n' variable inside the combination function.

To apply the side options range (1:12) to the side_choices I used the sapply( ) function.

By doing that I define the 'n' value as 2:12, meaning it will go from the number 2 to the number 12.

Because the manager wants 365 combinations, the answer is 7 (378 options for meals), because the range doesn't start from 1, it starts from 2.

EXERCISES 3 AND 4:

Case-control studies help determine whether certain exposures are associated with outcomes such as developing cancer. The built-in dataset esoph contains data from a case-control study in France comparing people with esophageal cancer (cases, counted in ncases) to people without esophageal cancer (controls, counted in ncontrols) that are carefully matched on a variety of demographic and medical characteristics. The study compares alcohol intake in grams per day (alcgp) and tobacco intake in grams per day (tobgp) across cases and controls grouped by age range (agegp).

The dataset is available in base R and can be called with the variable name esoph:

```
data(esoph)
head(esoph)
```

```
##   agegp     alcgp    tobgp ncases ncontrols
## 1 25-34 0-39g/day 0-9g/day      0        40
## 2 25-34 0-39g/day    10-19      0        10
## 3 25-34 0-39g/day    20-29      0         6
## 4 25-34 0-39g/day      30+      0         5
## 5 25-34     40-79 0-9g/day      0        27
## 6 25-34     40-79    10-19      0         7
```

You will be using this dataset to answer the following four multi-part questions (Questions 3-6).

The following three parts have you explore some basic characteristics of the dataset.

Each row contains one group of the experiment. Each group has a different combination of age, alcohol consumption, and tobacco consumption. The number of cancer cases and number of controls (individuals without cancer) are reported for each group.

3) How many groups are in the study? How many cases and controls are there?

```
nrow(esoph)
```

```
## [1] 88
```

```
all_cases <- sum(esoph$ncases)
all_cases
```

```
## [1] 200
```

```
all_controls <- sum(esoph$ncontrols)
all_controls
```

```
## [1] 775
```

“Each row contains one group of the experiment” –> that’s why I used nrow to answer the first question.

“The number of cancer cases and number of controls (individuals without cancer) are reported for each group” –> each row contains columns with the number of cases and controls.

To answer the 2nd and 3rd question, I used the sum function, to sum all the values on each column and apply it to a variable for later use.

esoph$ncases –> the $ sign is used to access a value (column) from the table.

4.A) What is the probability that a subject in the highest alcohol consumption group is a cancer case?

```
head(esoph$alcgp)
```

```
## [1] 0-39g/day 0-39g/day 0-39g/day 0-39g/day 40-79     40-79
## Levels: 0-39g/day < 40-79 < 80-119 < 120+
```

Firstly, I needed to know each section from the alcgp column, to get the information about the highest alcohol consumption group.

```
esoph %>%
    dplyr::filter(alcgp == "120+") %>%
    summarize(ncases = sum(ncases), ncontrols = sum(ncontrols)) %>%
    mutate(p_case = ncases / (ncases + ncontrols)) %>%
    pull(p_case)
```

```
## [1] 0.6716418
```

Then, I applied a filter to the table, to select only the rows with the highest alcohol consumption group and summarized the totals of controls and cases, to get the total.

To get the probability, I divided the number of cases (total) by the sum of all the occasions (ncases + ncontrols) –> both totals (summarized).

4.b) What is the probability that a subject in the lowest alcohol consumption group is a cancer case?

```
esoph %>%
    dplyr::filter(alcgp == "0-39g/day") %>%
    summarize(ncases = sum(ncases), ncontrols = sum(ncontrols)) %>%
    mutate(p_case = ncases / (ncases + ncontrols)) %>%
    pull(p_case)
```

```
## [1] 0.06987952
```

For this exercise I used the same logic as the previous one, but instead of getting the highest alcohol consumption group, I got the lowest alcohol consumption group.

4.c) Given that a person is a case, what is the probability that they smoke 10g or more a day?

```
smoke_case <- esoph %>%
    dplyr::filter(tobgp != "0-9g/day") %>%
    pull(ncases) %>%
    sum()

smoke_case / all_cases
```

```
## [1] 0.61
```

For this exercise I created the smoke_case variable, which filters the table by the value asked and sums the number of cases on this filtered group.

Then, I divided the variable by all the cases to get the probability asked on this exercise.

4.d) Given that a person is a control, what is the probability that they smoke 10g or more a day?

```
smoke_case <- esoph %>%
    dplyr::filter(tobgp != "0-9g/day") %>%
    pull(ncontrols) %>%
    sum()

smoke_case / all_controls
```

```
## [1] 0.4232258
```

For this exercise I created the smoke_case variable, which filters the table by the value asked, and sums the number of control on this filtered group.

Then, I divided the variable by all the controls to get the probability asked on this exercise.

EXERCISES 5 AND 6:

The following four parts look at probabilities related to alcohol and tobacco consumption among the cases.

5.a) For cases, what is the probability of being in the highest alcohol group?

```
highest_alc <- subset(esoph, alcgp == "120+")
highest_alc_cases <- sum(highest_alc$ncases)
highest_alc_cases
```

```
## [1] 45
```

```
all_cases
```

```
## [1] 200
```

On this exercise I used a different way of “filtering”, since I created a variable that contains only a selected part of the dataset.

Then to get the values I wanted, I used the sum( ) function and applied it to the cases column on the variable I created before.

```
45 / 200
```

```
## [1] 0.225
```

Lastly, having both values I needed to calculate the probability.

I divided one by the other and got the answer for the exercise.

5.b) For cases, what is the probability of being in the highest tobacco group?

```
highest_tab <- subset(esoph, tobgp == "30+")
highest_tab_cases <- sum(highest_tab$ncases)
highest_tab_cases
```

```
## [1] 31
```

```
31 / 200
```

```
## [1] 0.155
```

For this exercise, I used the same logic as the one before.

5.c) For cases, what is the probability of being in the highest alcohol group and the highest tobacco group?

```
highest_tab_alc <- subset(esoph, tobgp == "30+" & alcgp == "120+")
highest_tab_alc_cases <- sum(highest_tab_alc$ncases)
highest_tab_alc_cases
```

```
## [1] 10
```

```
10 / 200
```

```
## [1] 0.05
```

For this exercise, I used the same logic as the others, but with a small difference: I applied the same concepts, but "filtered" the dataset by two values at the same time, using "&" as "and".

5.d) For cases, what is the probability of being in the highest alcohol group or the highest tobacco group?

```
highest_tab_or_alc_cases <- highest_alc_cases + highest_tab_cases - highest_tab_alc_cases

highest_tab_or_alc_cases
```

```
## [1] 66
```

```
66 / 200
```

```
## [1] 0.33
```

For this exercise, I created a variable to get the highest alcohol group or the highest tobacco group, but never both at the same time.

That's why I subtracted the intersection of the groups after adding them together.

6.a) For controls, what is the probability of being in the highest alcohol group?

```
high_alc_controls <- esoph %>%
    dplyr::filter(alcgp == "120+") %>%
    pull(ncontrols) %>%
    sum()
p_control_high_alc <- high_alc_controls/all_controls
p_control_high_alc
```

```
## [1] 0.0283871
```

Just like exercise 4.A I created a variable to get the value I wanted, and then, divided it by the total of controls.

6.b) For controls, what is the probability of being in the highest tobacco group?

```
high_tob_controls <- esoph %>%
  dplyr::filter(tobgp == "30+") %>%
  pull(ncontrols) %>%
  sum()

p_control_high_tob <- high_tob_controls/all_controls
p_control_high_tob
```

```
## [1] 0.06580645
```

Just like exercise 4.A I created a variable to get the value I wanted, and then, divided it by the total of controls.

6.c) For controls, what is the probability of being in the highest alcohol group and the highest tobacco group?

```
high_alc_tob_controls <- esoph %>%
  dplyr::filter(alcgp == "120+" & tobgp == "30+") %>%
  pull(ncontrols) %>%
  sum()

p_control_high_alc_tob <- high_alc_tob_controls/all_controls
p_control_high_alc_tob
```

```
## [1] 0.003870968
```

Just like exercise 4.A I created a variable to get the value I wanted, and then, divided it by the total of controls.

6.d) For controls, what is the probability of being in the highest alcohol group or the highest tobacco group?

```
p_control_either_highest <- p_control_high_alc + p_control_high_tob - p_control_high_alc_tob

p_control_either_highest
```

```
## [1] 0.09032258
```

Just like exercise 4.A I created a variable to get the value I wanted, and then, divided it by the total of controls.

6.f) How many times more likely are cases than controls to be in the highest alcohol group or the highest tobacco group?

```
0.33 / p_control_either_highest
```

```
## [1] 3.653571
```

→ I got the ‘0.33’ value from exercise 5.d

THANK YOU FOR READING THIS MATERIAL!

Created by Isabella Heder