Problem Set Week 1

Bruno Nadalic Sotic

13/01/2023

Table of Contents

## Directions for the student

* Put all R code in code chunks and verbal answers outside code chunks.
* If you get a piece of R code to work, set the code chunk option eval=FALSE to ensure the document can still be knitted.
* Use tidyverse functions whenever possible.
* Comment your code to communicate your intentions.
* Ensure that the R Markdown document knits without problems into a PDF or Word document.
* Submit the R Markdown document on Canvas (under Assignments) before the deadline.

#Load all libraries in this code chunk.  
library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✔ ggplot2 3.4.0 ✔ purrr 0.3.4  
## ✔ tibble 3.1.6 ✔ dplyr 1.0.8  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.0  
## ✔ readr 2.1.2 ✔ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(dplyr)  
library(ggplot2)  
library (readr)

## Data

**sleepers.csv**. The effect of two drugs (variable *group*) on the increase in hours of sleep compared to a control condition (variable *extra*) in 67 patients (variable *ID*). The patient’s age group (*age*, 1 = 18-25, 2 = 26-35, 3 = 36-50, 4 = 50+) is provided as well.

## Questions

1. Import the data in R with readr:: and count the missing values on each variable using R code.

#import data, assuming the WD is set.   
#Alternatively: readr::read\_csv("C:/Users/sotic/Desktop/DWR1/sleeprs.csv")  
  
sleepers <- readr::read\_csv("sleepers.csv")

## Rows: 134 Columns: 4  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): group  
## dbl (3): ID, age, extra  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

#Count missing values for each column  
sleepers %>%   
 summarize\_all(funs(sum(is.na(.))))

## Warning: `funs()` was deprecated in dplyr 0.8.0.  
## ℹ Please use a list of either functions or lambdas:  
##   
## # Simple named list: list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)  
##   
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))

## # A tibble: 1 × 4  
## ID group age extra  
## <int> <int> <int> <int>  
## 1 0 0 3 0

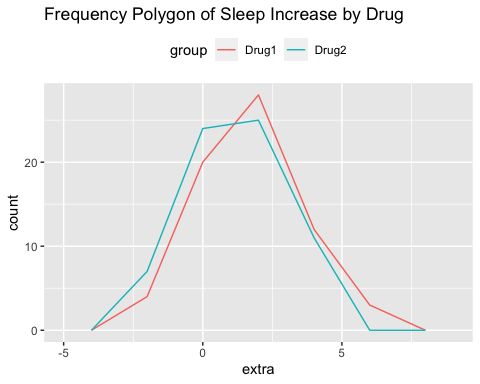
### Q1 Answer:

Only *‘Age’* seems to have three missing values.

| Grading | Max points | Awarded |  |
| --- | --- | --- | --- |
| Ex. 1 | 1 | 1 | Well done. Note that the only package you need to load is tidyverse. The other packages you load int he first chunk are all part of the tidyverse and loaded automatically. The solution of assuming the data is inside the same directory as the RMarkdown file is the simplest one for collaboration. Keep doing that for this course. |

1. Use ggplot2:: to create one figure containing a frequency polygon of sleep increase for each drug. Use different colors for the two polygons. Which drug seems to increase sleep more?

#Plot  
ggplot(data = sleepers, aes(x=extra, fill=group, color=group)) +   
 geom\_freqpoly(binwidth=2) +  
 scale\_fill\_manual(name="Drug", values=c("Drug 1" = "red", "Drug 2" = "blue"), labels=c("Drug 1", "Drug 2")) +  
 ggtitle("Frequency Polygon of Sleep Increase by Drug") +  
 theme(legend.position = "top")



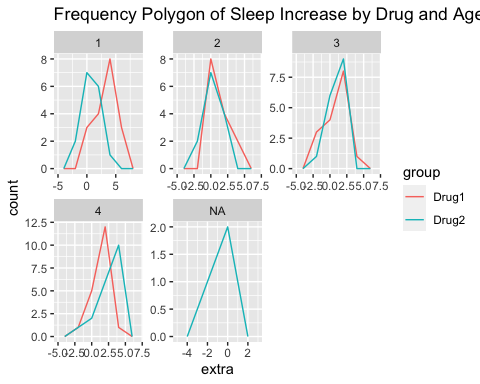
### Q2 Answer:

Based on visual inspection, both drugs seem to follow an innitial pattern. However, Drug 1 seems to increase sleep more at the peak. It is important to note that this might not be a correct conclusion. The graph shows the frequency of extra sleep for each drug group, but it does not provide information on the average extra sleep or the overall effect of the drug. To determine which drug is more effective at increasing sleep, we would need additional information such as the average increase in sleep for each drug group and a statistical analysis comparing the two groups. It is also important to note that the graph is based on observational data, which may not be enough to draw a conclusion about causality, a controlled experiment would be needed to establish causality.

| Grading | Max points | Awarded |  |
| --- | --- | --- | --- |
| Ex. 2 | 1 | 1 | Well done. Nice touch on editing labels! |

1. For each age group, create a figure as in Question 2 using facetting. Again, use ggplot2::. Which drug works better (gives more additional sleep) for which age group?

ggplot(data = sleepers, aes(x = extra, fill = group, color = group)) +  
 geom\_freqpoly(binwidth = 2) +  
 facet\_wrap(~ age, scales = "free") + #"free" parameter allows each facet to have its own scales.  
 ggtitle("Frequency Polygon of Sleep Increase by Drug and Age Group") +  
 scale\_fill\_manual(name="Drug", values=c("Drug 1" = "red", "Drug 2" = "blue"), labels=c("Drug 1", "Drug 2"))



### Q3 Answer

Based on the graphs, ‘Drug 1’ seems to perform slightly better for age groups in the categories of 1, 2 and 4. However, ‘Drug 2’ works slightly better for age group ‘3’. It is important to note that there are also missing values under ‘Drug 2’ as evident in the last plot. We could exclude that data and plot from the analysis and visualization by adding “na.rm = TRUE” under ggplot(), but I decided to leave it as it highlights whats lost.

| Grading | Max points | Awarded |  |
| --- | --- | --- | --- |
| Ex. 3 | 2 | 2 | Well done. For future reference, usually it’s best to remove NAs before plotting, unless they are of interest to the viewer for some reason. In this case, plotting the NAs doesn’t contribute to understanding the feature of the data you are presenting (namely diff. between drugs.) |

1. Use dplyr:: to create a table (tibble/data frame) containing the number of observations, the minimum, maximum, and average sleep increase for each drug.

### Q4 Answer:

sleepers %>%  
 group\_by(group) %>%  
 summarize(n = n(),  
 min = min(extra, na.rm = TRUE),  
 max = max(extra, na.rm = TRUE),  
 mean = mean(extra, na.rm = TRUE))

## # A tibble: 2 × 5  
## group n min max mean  
## <chr> <int> <dbl> <dbl> <dbl>  
## 1 Drug1 67 -2.07 5.98 1.61  
## 2 Drug2 67 -2.72 4.98 1.32

| Grading | Max points | Awarded |  |
| --- | --- | --- | --- |
| Ex. 1 | 1 | 1 |  |

1. Calculate the difference in extra sleep for each patient between drug 2 and drug 1, again using dplyr::. Show the first 10 rows of the results.

Tip: Sort the data and, in the end, only retain one case for each patient.

### Q5 Answer:

sleepers\_diff <- sleepers %>%   
 spread(group, extra) %>%   
 group\_by(ID) %>%   
 mutate(extra\_diff = `Drug2` - `Drug1`) %>%   
 select(ID, extra\_diff)   
  
head(sleepers\_diff,10)

## # A tibble: 10 × 2  
## # Groups: ID [10]  
## ID extra\_diff  
## <dbl> <dbl>  
## 1 1 -4.59   
## 2 2 2.46   
## 3 3 -4.48   
## 4 4 2.27   
## 5 5 -2.12   
## 6 6 -0.590  
## 7 7 3.94   
## 8 8 -0.385  
## 9 9 3.09   
## 10 10 -0.385

| Grading | Max points | Awarded |  |
| --- | --- | --- | --- |
| Ex. 5 | 2 | 2 | Well done. You are already using some of the code we will talk about next for data tidying. Note that a newer version of spread() is pivot\_wider(). We’ll use that in the tutorial. You can just add the head() function to the set of pipes instead of saving out sleepers\_diff first. |

1. Create a plot to explore the covariation (association) between two continuous variables in *your project’s data set* that are of interest to you. Use comments in the R code to justify the choices that you made to create this plot.

Note: You may use grouped summaries (aggregation) to create continuous variables.

### Q6 Answer:

For this, we can use the ‘Responders2009.csv’ from my data project, since this csv contains the most variables. Given that the dataset is in German, it would be helpful if the variable names would be translated to English. We can run a quick function to do that using Googles API.

install.packages("translate") # install the package if not already installed  
library(translate)  
  
# Authorization through the API key  
Sys.setenv("google\_api\_credidentials"="path/to/your/key.json") #Authorize via account. Creditentials are personal and stored on my machine. Hence, this is left blank.  
  
# read in the German dataset  
data <- read.csv("NLResponders2009.csv", header = TRUE, sep = ",")  
  
# translate the data to English  
translated\_data <- translate(data, source = "de", target = "en")  
  
# write the translated data to a new CSV file  
write.csv(translated\_data, file = "Responders2009.csv", row.names = FALSE)

data <- read\_csv("Responders2009.csv")

## New names:  
## • `D23\_02` -> `D23\_02...647`  
## • `D23\_02` -> `D23\_02...648`  
## • `D23\_02` -> `D23\_02...649`  
## • `D23\_02` -> `D23\_02...650`  
## • `D23\_02` -> `D23\_02...651`

## Warning: One or more parsing issues, see `problems()` for details

## Rows: 225 Columns: 715  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (366): A29\_02, A29\_03, A29\_04, A30\_01, A30\_02, A31\_C4\_neg, A31\_C1\_neut, ...  
## dbl (349): ID\_Proband, Responder, Location, Treatment, Participation1, Parti...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

head(data)

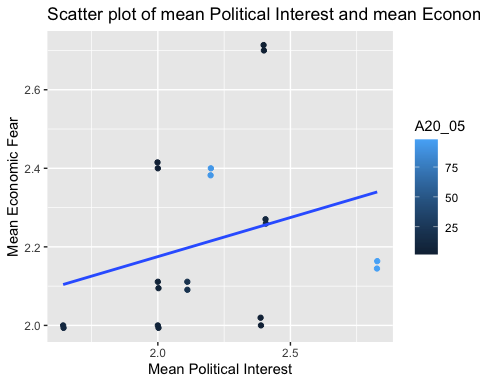
## # A tibble: 6 × 715  
## ID\_Proband Responder Location Treatment Participation1 Participation2  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 1 3 5 1 1  
## 2 2 2 3 5 1 1  
## 3 3 3 3 5 1 1  
## 4 4 4 3 5 1 1  
## 5 5 5 3 5 1 1  
## 6 6 6 3 5 1 1  
## # … with 709 more variables: Participation3 <dbl>, Participation4 <dbl>,  
## # A01 <dbl>, A02 <dbl>, A03 <dbl>, A04\_01 <dbl>, A04\_03 <dbl>, A05 <dbl>,  
## # A06 <dbl>, A07 <dbl>, A08 <dbl>, A09 <dbl>, A10 <dbl>, A11 <dbl>,  
## # A12 <dbl>, A13 <dbl>, A14 <dbl>, A15 <dbl>, A16 <dbl>, A17\_03 <dbl>,  
## # A18 <dbl>, A19 <dbl>, A20\_04 <dbl>, A20\_05 <dbl>, A21\_03 <dbl>, A22 <dbl>,  
## # A23\_01 <dbl>, A23\_02 <dbl>, A23\_03 <dbl>, A23\_04 <dbl>, A23\_05 <dbl>,  
## # A23\_06 <dbl>, A24\_01 <dbl>, A24\_02 <dbl>, A25\_01c1 <dbl>, A25\_01c2 <dbl>, …

Most of the data consist of coded values, ordinals and integers. To explore the association between two continuous variables, we can plot the mean *Political Interest* (A01) and *Economic Fear* (A14) across the different *Years* (A20\_05). Given that there are missing values in the dataset (denoted with ‘99’), those values should be removed.

#remove missing values  
data\_clean <- data[data$A20\_05 != 99 & data$A01 != 99 & data$A14 != 99,]  
  
#group the data by A20\_05  
grouped\_data <- group\_by(data\_clean, A20\_05)  
  
#create new variable mean\_A01, mean\_A14  
grouped\_data <- summarize(grouped\_data, mean\_A01 = mean(A01), mean\_A14 = mean(A14))  
  
#create scatter plot  
ggplot(data = grouped\_data, aes(x = mean\_A01, y = mean\_A14, color = A20\_05)) +  
 geom\_point() +  
 geom\_jitter() +  
 geom\_smooth(method = "lm", se = FALSE) + #fit a regression line  
 ggtitle("Scatter plot of mean Political Interest and mean Economic Fear across Years") +  
 xlab("Mean Political Interest") + ylab("Mean Economic Fear") #+guides(color=guide\_legend(title="Years"))

## `geom\_smooth()` using formula = 'y ~ x'

## Warning: The following aesthetics were dropped during statistical transformation: colour  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?



This code first removes missing values (99) from the data set by subsetting the data where A20\_05 (Years), A01 (Political Interest), and A14 (Economic Fear) are not equal to 99. Then it groups the data by Years and creates new variables mean Political Interest, mean Economic Fear which are the mean of A01, A14 respectively. Then it creates a scatter plot of the new variables using ggplot package, also it color codes the points according to the value of A20\_05 (Years).

The plot would indicate no association among the two variables across the years.

| Grading | Max points | Awarded |  |
| --- | --- | --- | --- |
| Ex. 6 | 2 | 1 | I appreciate the extra work you did here for translation. A couple of things though - install.packages(‘translate’) doesn’t work, because the package has since been removed from CRAN. I guess you installed it in some other way? Make sure you consistently use tidverse code (e.g. read\_csv and filter(A20\_05!=99)) instead of base r (e.g. read.csv, data[data$A20\_05!=99,]). Importantly, use pipes. I will start removing points for not using tidyverse codes starting in problem set. Now, about the actual graph: The type of graph is appropriate to represent the relationship between 2 continuous variables, but there are some issues here. First, I think you accidentally picked the wrong variale. A20\_05 is people’s knowledge about the “Deutschlandplan” and missing values on this variable are coded as 99, 98, or 93 (see codebook). Second, I don’t understand the purpose of computing averages per year. Why not show all datapoints? if you wanted to see differences in the relationship between x and y per year, facetting may be more appropriate. Finally, since you are already working with labels - note that you can also change labels for the legend (e.g. guides(fill=guide\_legend(title=“New Legend Title”))) |
| Flawless knitting | 1 | 1 |  |
| **Total** | 10 | 9 |  |