CS112 Assignment 1

Akmarzhan Abylay

January 2020

1 Preprocessing

[1]:
```
#downloading the data into R
foo <- read.csv("https://tinyurl.com/yb4phxx8")

#8 columns that represent dates, which we need to be careful about
date.columns <- c(11, 12, 14, 15, 16, 17, 18, 25)

#loops through the columns that involve dates
for(i in date.columns) {
  #identifies which values are missing in the i-th column
  which_values_are_missing <- which(as.character(foo[, i]) == "")

  #missing values that we found above are replaces by NA, since
  #it is easier for R to process them rather than blank space
  foo[which_values_are_missing, i] <- NA

  #these columns are in the "factor" value form, which means that
  #R stores the data in these columns as levels (categories,
  #instead of Date values), which is why we convert them to the
  #"Date" class to manipulate easier
  foo[, i] <- as.Date(as.character(foo[, i]))
}

#find rows which don't have a rating in the foo dataset
which.have.NAs <- which(is.na(foo$Rating == TRUE))

#remove all these rows from foo and save as new_foo; we will work with the new␣
 ↪cleaned dataset, although it is also possible to work with the raw one - only␣
 ↪where the dates are updated; the results are different, but the difference is␣
 ↪negligible; also, I prefer using the cleaned dataset as it will give us more␣
 ↪accurate data on projects that have been already processed and that were rated␣
 ↪(rather than the ones with no rating)
new_foo <- foo[-which.have.NAs, ]
```

2 Questions

2.1 Question 1

When projects are approved, they are approved for a certain period (until the time of "original completion date"). While projects are active, this "original" completion date is often extended, and then there is a "revised" completion date. You have been told that project duration at approval is generally about two years (24 months). In other words, (purportedly) when projects are approved, the difference between the original project completion date and the approval date is (supposedly) approximately 24 months.

(a) Is this claim true? Explain.

Since we only need to consider the non-missing Circulation Dates after 2009-01-01, we will first get rid of the "NA" values and then filter out the data to include entries later than 2009-01-01.

[2]:
```
foo_noNA <- new_foo[!is.na(new_foo$CirculationDate), ]
df <- foo_noNA[which(foo_noNA$CirculationDate >= as.Date("2009-01-01")), ]

#check:
min(df$CirculationDate)
```

2009-01-14

To address this question, we will need to consider the difference between the average original project completion date and the average approval date as stated. Once again, we will consider only those entries which have a non-missing original completion date.

[3]:
```
no_od <- which(is.na(df$OriginalCompletionDate))
df_Dates <- df[-no_od,] #deleting the rows with missing original completion dates
expected_duration <- mean(df_Dates$OriginalCompletionDate) -␣
 ↪mean(df_Dates$ApprovalDate)
expected_duration
```

Time difference of 650.9313 days

Answer:

As we see, the time difference is 650.9313 days, which is a bit less than the approximate 730 days. Depending on what people consider acceptable, this claim is partially true, since it is close to 2 years, but still not 2 years on average.

(b) Has the length of project delay, measured as the difference between "OriginalCompletion-Date" and "RevisedCompletionDate", changed over time (consider projects circulated earlier and circulated later)?

You will need to make a choice of how to deal with missing information, which you should

explicitly discuss. Be sure to also discuss mean delays, median delays, and the interquartile range of delays (using the "quantile" function).

[4]:
```
#create a new column with circulation years
df_Dates$CirculationYear <- format(df_Dates$CirculationDate, "%Y")

#we need to make sure that there is no missing information
sum(is.na(df_Dates$RevisedCompletionDate))
```

0

In the above code, I made sure that there is no missing data in the Revised Completion Date column (and we already deleted NA values of the Original Completion Date earlier). Now we will add a new column "Circulation Year" to get some sense of how the data changed throughout the years.

[5]:
```
#creating a delay column
df_Dates$Delay <- df_Dates$RevisedCompletionDate -␣
 ↪df_Dates$OriginalCompletionDate
#this is an array of years to use in the for loop
years <- c(2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018)

#setting up the variables (arrays) to store the data in
df_mean <- c()
df_median <- c()
df_IQR <- c()
#looping through each year
for (year in years) {
    #choosing a subset of Delay values based on the circulation year
    subset1 <- subset(df_Dates, subset = CirculationYear==year, select=Delay)
    #extracting numeric values to get the mean/median/IQR
    subset1 <- as.numeric(as.character(subset1$Delay))
    #finding the needed values
    df_mean <- c(df_mean, mean(subset1))
    df_median <- c(df_median, median(subset1))
    df_IQR <- c(df_IQR, quantile(subset1, 0.75)-quantile(subset1, 0.25))
}

#creating a data frame in which to store and show the found values
delayByYear <- data.frame("CirculationYear"=years, "mean.delay"=df_mean, "median.
 ↪delay"=df_median, "IQR.delay"=df_IQR)
delayByYear
```

CirculationYear	mean.delay	median.delay	IQR.delay
2009	657.7644	549.0	610.00
2010	634.0000	547.0	511.25
2011	584.7664	426.0	610.00
2012	538.2390	487.0	409.50
2013	503.5893	396.0	493.00

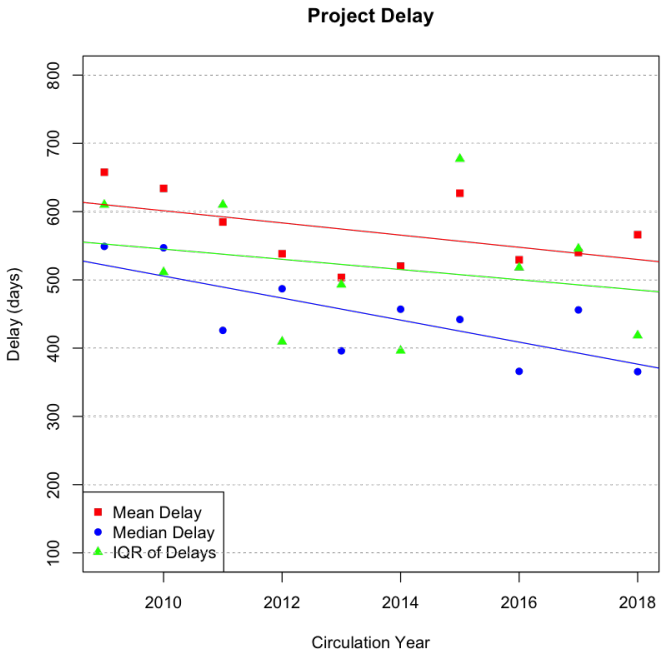
CirculationYear	mean.delay	median.delay	IQR.delay
2014	520.1750	457.0	396.25
2015	626.9821	442.0	677.25
2016	529.3702	366.0	518.00
2017	540.1632	456.0	545.75
2018	566.1905	365.5	418.50

As you could see, the delays fluctuate over the years, and to better see the pattern, it would be best if we plotted it.

[6]:
```
#plots the mean, median, IQR by year in different styles to see better
plot(delayByYear$CirculationYear, delayByYear$mean.delay, pch=15, col="red",␣
 ↪ylim=c(100, 800), xlab="Circulation Year", ylab="Delay (days)", main="Project␣
 ↪Delay")
points(delayByYear$CirculationYear, delayByYear$median.delay, pch=16, col="blue")
points(delayByYear$CirculationYear, delayByYear$IQR.delay, pch=17, col="green")
legend("bottomleft", pch=c(15,16,17), col=c("red","blue","green"),␣
 ↪legend=c("Mean Delay", "Median Delay", "IQR of Delays"))
grid(nx=NA, ny=NULL)

lm1 <- lm(mean.delay ~ CirculationYear, data=delayByYear)
# lines(lm1, years)
lm2 <- lm(median.delay ~ CirculationYear, data=delayByYear)
lm3 <- lm(IQR.delay ~ CirculationYear, data=delayByYear)

abline(lm1, col="red")
abline(lm2, col="blue")
abline(lm3, col="green")
```



Answer:

There is no obvious trend in the delays since the mean, median, and IQR mostly fluctuate. Still, when we plot the line of best fit for each of the variables, we see that there is a slight negative association, possibly corresponding to the decreasing delay for later projects. This suggests that there might be an improvement in project planning as the delay gets smaller as time passes.

As has been mentioned, the delays have been fluctuating for these 9 years. For example, the average delay decreased from 658 days in 2009 to 503 days in 2013 before increasing once again to 627 in 2015 and then decreasing to 529 in 2016. At the same time, the median delay fluctuates too but is much smaller than the average delay (mostly 549 to 365). There is a similar trend in the IQR, which shows the middle 50% values and which is also a good measure of spread since it is not that affected by outliers. In the plot, we see that it is fluctuating but generally decreasing over time.

(c) How does the original planned project duration differ from actual duration (if the actual duration is measured as the duration between "ApprovalDate" and "RevisedCompletionDate")?

We will print out the needed mean, median, and IQR values to understand the data better. Below is the code for the actual and expected durations.

[7]:
```
#creating an Actual Duration column
df_Dates$ActualDuration <- df_Dates$RevisedCompletionDate - df_Dates$ApprovalDate

#Actual Duration statistics
paste ("Actual Duration:")
mean(df_Dates$ActualDuration)
median(df_Dates$ActualDuration)
quantile(df_Dates$ActualDuration)
IQR(df_Dates$ActualDuration)

#finding the Expected Duration
df_Dates$ExpectedDuration <- df_Dates$OriginalCompletionDate -␣
 ↪df_Dates$ApprovalDate

#statistics for the Expected Duration
paste("Expected Duration")
mean(df_Dates$ExpectedDuration)
median(df_Dates$ExpectedDuration)
quantile(df_Dates$ExpectedDuration)
IQR(df_Dates$ExpectedDuration)
```

'Actual Duration:'

```
Time difference of 1220.019 days
```

```
Time difference of 1120.5 days
```

```
Time differences in days
     0%     25%     50%     75%    100%
  56.00  838.25 1120.50 1482.75 4395.00
```

644.5

'Expected Duration'

```
Time difference of 650.9313 days
```

```
Time difference of 599.5 days
```

```
Time differences in days
     0%     25%     50%     75%    100%
  18.00  406.25  599.50  792.00 3369.00
```

385.75

Answer:

By definition, in the code, the delay of the project is the same thing as the difference between the revised project duration and the initially planned project duration. This means that our results should not be very different from what we have stated in section (b). As seen from the quantile table, there are outliers (more big values) in both the actual duration and revised duration data. Also, the median is smaller than the mean in both data, which means that the distributions are right-skewed.

All descriptive stats in the actual duration data is much bigger (almost twice) than the expected duration, which suggests that there must have been a delay (which is true). On average, the actual duration and the expected duration would differ by 569 days, which means that the teams would extend their original completion day by almost 1.5 years, on average.

2.2 Question 2

What % of projects completed between 2010 and now were rated 0? What % over the same time period were rated 1? What % were rated 2? What % were rated 3? Answer these questions using a table or a figure. Provide a title and an explanatory sentence or two that provides the numerical % results rounded to the nearest percentage point.

As I understood, the Circulation Date represents the moment the project was officially closed and "circulated," while the Revised Completion Date was the moment when it was completed. Thus, I took the Revised Completion Date as the filter, since it is asking for the projects "completed" between 2010 and now.

[8]:
```
df2010 <- foo_noNA[which(foo_noNA$RevisedCompletionDate >= as.
 ↪Date("2010-01-01")), ]
```

```
print("Distribution of Project Ratings")
prop.table(table(df2010$Rating)) * 100
```

```
[1] "Distribution of Project Ratings"
```

```
        0         1         2         3
 2.124312 11.487018 72.147915 14.240755
```

Answer:

As we can see from the table above, 2.1% of the projects received a rating of 0, 11.5% of 1, 72.1% of 2, and 14.2% of 3. This shows that the majority of the projects received a 2.

2.3 Question 3

Repeat problem 2, but this time limit your analysis purely to policy and advisory technical assistance ("PATA") projects.

[9]:
```
#choosing PATA projects
df_PATA <- df2010[which(df2010$Type == "PATA"),]

print("Distribution of Project Ratings (PATA only)")
table(df_PATA$Rating) / length(df_PATA$Rating) * 100
```

```
[1] "Distribution of Project Ratings (PATA only)"
```

```
        0         1         2         3
 1.098901  8.058608 72.161172 18.681319
```

Answer:

As we can see from the table above, 1.1% of the projects received a rating of 0, 8.1% of 1, 72.2% of 2, and 18.7% of 3. This shows that the majority of the projects received a 2, while more projects received a 3 than 0 and 1, combined.

2.4 Question 4

Identify the top 10% of projects by "Revised.Amount" and the bottom 10% of projects by "RevisedAmount" ("RevisedAmount" shows the final project budget). Compare the ratings of these projects. Can you draw a causal conclusion about the effect of budget size on ratings? Why or why not?

[10]:
```
#finding the top and bottom 10% of projects by RevisedAmount
df_Top <- df[which(df$RevisedAmount >= quantile(df$RevisedAmount, 0.9)),]
df_Bottom <- df[which(df$RevisedAmount <= quantile(df$RevisedAmount, 0.1)),]

#compare ratings
print("The difference")
```

```
(table(df_Top$Rating) / length(df_Top$Rating) * 100) -
  (table(df_Bottom$Rating) / length(df_Bottom$Rating) * 100)
# The differences seem to be quite small (i.e., about 2% difference max)

#printing the necessary code
print("Top 10%")
table(df_Top$Rating) / length(df_Top$Rating)
print("Bottom 10%")
table(df_Bottom$Rating) / length(df_Bottom$Rating)
print(paste("Length of Top 10% data:", length(df_Top$Rating)))
print(paste("Length of Bottom 10% data:", length(df_Bottom$Rating)))
```

```
[1] "The difference"
```

```
        0         1         2         3
 1.261549 -1.226013  2.167733 -2.203269
```

```
[1] "Top 10%"
```

```
         0          1          2          3
0.02380952 0.10714286 0.73809524 0.13095238
```

```
[1] "Bottom 10%"
```

```
         0          1          2          3
0.01119403 0.11940299 0.71641791 0.15298507
```

```
[1] "Length of Top 10% data: 168"
[1] "Length of Bottom 10% data: 268"
```

Answer:

We can't compare the number of projects with different ratings since these two data pieces have very different lengths (168 vs. 268), which is why we would want to identify the difference between the proportions with respect to the lengths of the datasets. The difference seems to be quite small (at max 2%) for causal inferences, so we would want to test that using the difference of proportions test for significance.

Null Hypothesis: $p_1 = p_2$

Alternative Hypothesis: $p_1 != p_2$

Here p_1 and p_2 correspond to the proportion in the top and bottom 10%, respectively. We set our α, the significance level equal to 0.05. Below is the function that computes the corresponding p-values.

[11]:
```
proportion <- function (n1, n2, p1, p2){

    p_values <- c()
```

```
    #finding the number of people who got the corresponding ratings
    y2 <- round(n2*p2)
    y1 <- round(n1*p1)

    for (item in 1:length(p1)) { #performing the test for each of the existent␣
↪ratings
        p_c <- (y1[item]+y2[item])/(n1+n2)
        p1_1 <- p1[item]
        p2_2 <- p2[item]

        #we are using a general formula for the z-score for the difference
        #of proportions test I am not discussing where specifically this formula
        #came from, since it is unnecessary for the scope of this exact␣
↪assignment and it would take too many words, which would be unnecessary extra␣
↪work for you
        #to read
        z_score <- (p1_1-p2_2)/(sqrt(p_c*(1-p_c)*(1/n1+1/n2)))
        p <- 2*pnorm(-abs(z_score)) #calculating the p-value
        p_values <- c(p_values, p)
    }

    for (rating in 1:length(p_values)){ #printing out
    print(paste("The p-value in proportion difference test for rating",␣
↪rating-1, "is", p_values[rating]))
}
}

n1 <- 168
n2 <- 268

p1 <- c(0.02380952, 0.10714286, 0.73809524, 0.13095238)
p2 <- c(0.01119403, 0.11940299, 0.71641791, 0.15298507)

proportion(n1, n2, p1, p2)
```

```
[1] "The p-value in proportion difference test for rating 0 is
0.307737852571226"
[1] "The p-value in proportion difference test for rating 1 is
0.695793944479363"
[1] "The p-value in proportion difference test for rating 2 is
0.621860173925173"
[1] "The p-value in proportion difference test for rating 3 is
0.524250291611972"
```

As we could see, the two-tailed difference of the proportions significance test showed us that all p-values are much larger than our base 0.05, which suggests that we will not reject our null hypothesis. It is because there is no observed significant difference between the proportions of different ratings from the two datasets with top and bottom 10% by Revised Amount. This may

suggest that there is no significant correlation between the budget size and rating. Let's analyze more data below.

[12]:
```
#I will only look into Dept and Country data, since although Cluster data had
#some differences percentage-wise, if we look at the counts, we will identify
#a lot of missing data. I will also only look at the counts, since there are
# different categories and the percentages might not give the best information
#(e.g., there might be, for example, a 100% difference because of the lack of␣
 ↪that exact project type)

table(df_Top$Dept) - table(df_Bottom$Dept)
```

```
      AED*       AGD      AGDX      ARDD       AWD      AWD*      BDCO
         0         0         0         0         0         0         0
     BPMS      CPSO      CRPN       CTL      CWRD       DOC      EARD    ECRD**
         0         0         0        -2         1         0        -6        -1
     ERCD      IDBD       IED       IFD      IRDD       IWD      MKRD       OAG
       -19         0         0         0         0         0         0         0
      OAI       OCO      OCRP      OESD       OGC     OOMP       OPO      OPPP
       -2        -1        -2         0         0         0         0         0
     OREI       ORM      PARD      PPFD       PSD       PSG      PSOD       RMU
       -6         0        -8        -3         0         0       -10         0
     SARD      SDCC      SERD    SERD** SFSP-AUS       SPD        TD      VPO1
      -27        10       -20         0         0        -2        -2         0
     VPPC       VPW
         0         0
```

[13]:
```
(table(df_Top$Country) / length(df_Top$Country) * 100) -
  (table(df_Bottom$Country) / length(df_Bottom$Country) * 100)
```

```
       AFG         ARM         AZE         BAN         BHU         BRU
2.60305615  0.00000000  0.00000000 -5.22388060 -1.11940299  0.00000000
       CAM         COO         FIJ         FSM         GEO         IND
-2.54086709 -0.37313433 -0.37313433 -0.74626866 -1.11940299  4.14889837
       INO         KAZ         KGZ         KIR         KOR         LAO
0.73738451 -0.37313433 -1.49253731  0.22210377  0.00000000  0.07107321
       MAL         MLD         MON         MYA         NAU         NEP
-0.37313433  0.59523810  0.66631130 -0.75515281 -1.49253731 -0.82622601
       PAK         PAL         PHI         PNG         PRC         REG
3.34044065  0.00000000 -0.38201848 -0.52416489 -1.94562900 11.24733475
       RMI         SAM         SIN         SOL         SRI         TAJ
-0.74626866  0.00000000  0.00000000 -0.15103056 -1.49253731 -0.37313433
       TAP         THA         TIM         TKM         TON         TUV
0.00000000 -2.23880597  0.22210377 -0.37313433  0.22210377  0.00000000
       UZB         VAN         VIE
0.22210377 -0.74626866  1.48365316
```

For the top 10%, there are more projects in SDCC and Pakistan (PAK). For the bottom 10%, there are more projects in SARD, ERCD, AND SERD and Bangladesh (BAN). Based on these differences in other categories and no significant difference in the rating of the top and bottom 10% of projects by Revised Amount, it seems that the rating is not correlated to the budget. However, there is a possibility that the rating or the budget might be associated with any other project properties, such as department or country. We wouldn't be able to make any valid conclusions or causal inferences since we have observational data, where the data is not randomly distributed "treatment" and "control" to make comparisons.

2.5 Question 5

Imagine your manager asks you to apply Jeremy Howard's drivetrain model to the problem of optimal budget-setting to minimize project completion delays (i.e., the difference between revised and original completion dates).

Answer:

- ***(a) decision problem or objective?***

Minimize the gap between the actual completion date and the original planned completion date (delay).

- ***(b) lever or levers?***

Allocation of the budget (or time management strategies, number of people in the team).

- ***(c) ideal RCT design?***

It will be tough to match projects by their properties because they are mostly very different. However, randomly assigning different budgets to different projects will more or less be the best we can get from such a complex environment as all of the characteristics of the project will just overlap. We then could measure the difference in delays between those different budgets, possibly getting an unbiased result. We can also do blinding, which requires masking the identity of the treatment from investigators and data analysts. Participants are not included because it would be hard not to know how much money is allocated to their own project since they are the ones spending the money afterward.

- ***(d) dependent variable(s) and independent variable(s) in the modeler***

Independent variable: budget allocation, dependent variable: project delay.

- ***(e) And—Why would running RCTs and modeling/optimizing over RCT results be preferable to using (observational, non-RCT) "foo" data?***

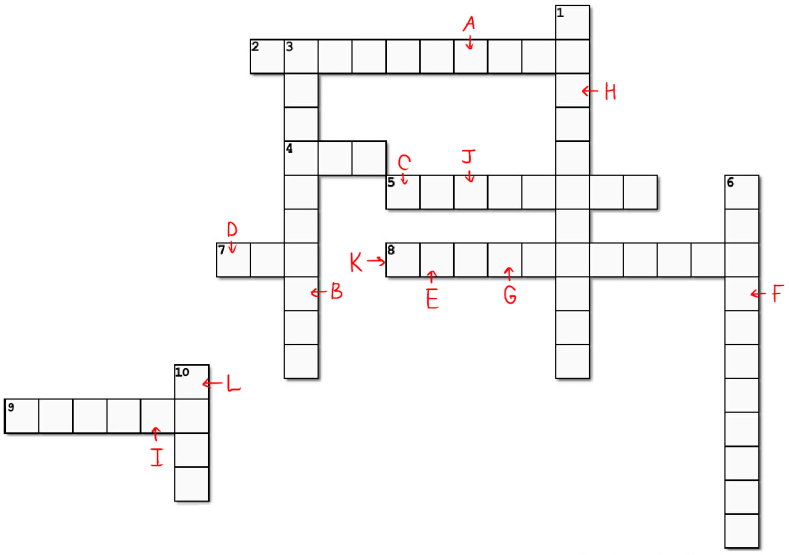
In observational data, projects which have different allocated budgets are not necessarily comparable in reality. The results may be biased by some other confounding variables, such as the country or department. Thus, we can't make conclusions about the causal connections between, per se, delays, and the budget. RCTs would be preferable since they eliminate bias in treatment assignment (e.g., selection bias, which is introduced by selecting a sample which is not representative of the population, is eliminated by randomized assignment). It also ensures blinding, so there is no examiner bias (i.e., when the experimenter's expectations can affect the participants' performance and the study results).

3 Bonus Question

3.1 Prompt

Congrats on making it through the first weeks of CS112! Below is a refresher for what you have been doing these days in the form of a crossword. Try to answer all the questions (and even the calculations!), and based on your answer, it will lead to an exciting website.

We have marked capital alphabetical letters, such as A, B, C, D, etc. in red to the locations of the characters we will need. The actual answer characters corresponding to these locations will be used in the website URL, which will look approximately like this: **https://ABCDEFG.com/FHIJKLD**, but instead of the capital alphabetical letters, you need to input the corresponding characters (or numbers!) from the answers (all-lowercase!). The produced website will lead you to some interesting data and plot. Let's get started!



Across

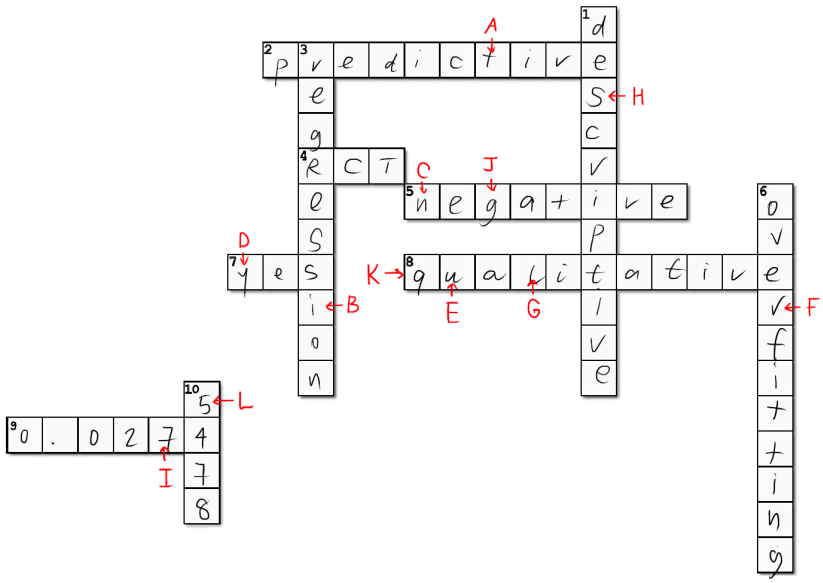
2. As an organizer, you want to know how many water bottles you'll need to prepare for CIVITAS. What kind of inference is this?
4. Our beloved study design that eliminates bias by randomly assigning treatment and control, and comparing their effects.
5. Your average increase in y associated with a one-unit increase in x is below zero, what kind of correlation is this?
7. Do you like CS112?
8. A scientific method of observation to gather non-numerical data.
9. Find the correlation coefficient between the rating and the budget after Question 2. Round to the 4th decimal point.

Down

1. You want to know what your popularity rating is within the Minerva community. What kind of inference would you want to make?
3. A statistical tool to determine the strength of the relationship between a dependent and multiple independent variables.
6. What happens when when a function is too closely fit to a limited set of data points (i.e., focuses on noise too much)?
10. Calculate how many unknown (i.e., NA) values there are in the Rating column within the 'foo' dataset.

3.2 Answer Key and Notes

If students answer all questions correctly, they will get something like this:



Across

2. As an organizer, you want to know how many water bottles you'll need to prepare for CIVITAS. What kind of inference is this?
4. Our beloved study design that eliminates bias by randomly assigning treatment and control, and comparing their effects.
5. Your average increase in y associated with a one-unit increase in x is below zero, what kind of correlation is this?
7. Do you like CS112?
8. A scientific method of observation to gather non-numerical data.
9. Find the correlation coefficient between the rating and the budget after Question 2. Round to the 4th decimal point.

Down

1. You want to know what your popularity rating is within the Minerva community. What kind of inference would you want to make?
3. A statistical tool to determine the strength of the relationship between a dependent and multiple independent variables.
6. What happens when when a function is too closely fit to a limited set of data points (i.e., focuses on noise too much)?
10. Calculate how many unknown (i.e., NA) values there are in the Rating column within the 'foo' dataset.

This will map them to this website: https://tinyurl.com/rs7gq5y, which has an interesting parametric graph that looks like a portrait of Einstein:) My idea was to create a task that would be exciting for students to complete in a challenge-type. This is not a very hard task, but it will be a good fit to get them started with the assignment or as a refresher/first task. Also, the portrait of Einstein was just a random thing, and instead, to motivate students to complete this task, we could, for example, make it optional and link the website with the answer key for the next assignment (from last year).

4 Code

Here is the link to the gist with the complete code for this assignment.