Stat 5810, Section 003
Statistical Visualization II
Spring 2019
Homework 3 - Extra Credit

ShaunMicheal Bartschi

A01975136

May 3, 2019

Stat 5810/6910 Statistical Visualization II Spring 2019

Homework Assignment 3 (4/11/2019)

50 (+ 12.5 EC) Points — Due Wednesday 5/1/2019 (via Canvas by 11:59pm)

Special Instructions: This HW consists of four questions. All questions are worth 25 points. You have to work on at least two of these questions. If you decide to work on three or all four questions, your two highest scores will directly count as your HW 3 score. 50% of your third highest score will be awarded as extra credit for the course. This will allow you to make up for any point deductions because of a late HW submission earlier in the semester or any other points you may have lost earlier on. The fourth highest score will just be dropped So, if you get (18, 25, 2, 23) points out of these four questions, the $25 + 23 = 48$ points count directly for HW 3, $0.5 \cdot 18 = 9$ points count as extra credit, and the 2 points are dropped.

It is totally up to you to choose on which of these four questions you want to work. But, arrange them in the proper numeric order in your HW submission.

As always, include your R code for each question! Load all required R packages and the data in the first part of a question. Also see the general instructions on the last page of this HW.

(iii) **Shiny App for Bad Color Graphic from HW 2** (25 Points):

Build a shiny app that allows a user to toggle between the original graphic with the bad colors and a graphic with improved colors. These should be your graphics / improvements from HW 2.

Moreover, add a 2nd selection mechanism that let's a user check for various color-blind appearances of these graphics, i.e., the original and the improved one. This should be based on the *dichromat* R package and the three color–blindness types deutan, protan, and tritan. Overall, your app should be able to show six differently colored graphics: two sets of color (original and improved) × three types of color–blindness. Include screenshots that show your shiny app and the resulting graphics for all of these six settings.

Adding a toggle button that allows you to add/choose different symbols or sizes (or what else you may have done in HW 2 to further improve your original graph) is optional.

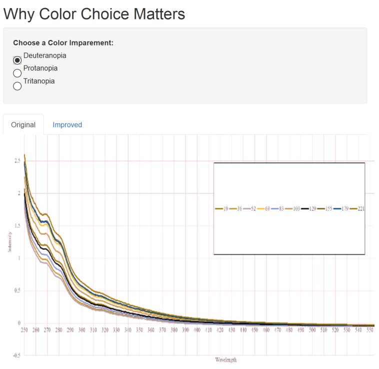
When you develop such an app, follow the principle from class: Start with the most basic app that just shows your original bad graph (the default). Then add one control option after the other.

Include only the final version of your R code. Make sure that you change to `eval=FALSE` for this question when you compile your Rnw document.

```
> library(shiny)
> library(shinythemes)
> library(dichromat)
> library(ggplot2)
> library(RColorBrewer)
> library(png)
> setwd("C:/Users/Shaun/Desktop/StatVis/StatVis2/HW3/EC")
> #help from https://shiny.rstudio.com/gallery/image-output.html
> #and also http://bit.ly/Shiny-updateTabsetPanel
>
> spectroscopy <- read.csv(file = "Spectroscopy.csv",
+                          header=TRUE, sep=",")
> better.col <- brewer.pal(10, "Spectral")
> my.plot <- ggplot(data = spectroscopy, aes(x=Wavelength)) +
+         xlab("Wavelength (nm)") +
+         ylab("Absorbance") +
+         ggtitle("UV/Vis Spectoscopy of Cd-Se") +
+         scale_color_brewer(name="Time in \nSeconds",
+                              labels = c("19s","36s","52s","69s",
+                                         "83s","103s","129s","155s",
+                                         "179s","221s"),
+                             palette = "RdYlBu") +
+         xlim(250,600) +
+         theme(legend.position = c(.8,.7),
+               plot.title = element_text(hjust = 0.5))
> ui <- fluidPage(
+   # App title ----
+   titlePanel("Why Color Choice Matters"),
+
+   # Sidebar layout with input and output definitions ----
+   sidebarLayout(
+
+     # Sidebar panel for inputs ----
+     sidebarPanel(# Input: Selector for choosing graph ----
+       radioButtons("color", "Choose a Color Imparement:",
+                    c("Deuteranopia", "Protanopia", "Tritanopia"))
+
+     ),
+
+     # Main panel for displaying outputs ----
+     mainPanel(
+       tabsetPanel(id = "graph",
+                    tabPanel(title = "Original",
+                             imageOutput("original.plot")),
+                    tabPanel(title = "Improved",
+                             plotOutput("improved.plot"))
+                    )
+     )
+   )
+ )
> server <- function(input, output) {
+   output$original.plot <- renderImage({
+     if (input$color == "Deuteranopia") {
+       return(list(src = "Deuteranopia_graph.png",
+         contentType = "image/png",
+         width = 725,
+         height = 450,
+         alt = "deutan bad graph"
+       ))
+     }
+     else if (input$color == "Protanopia") {
+       return(list(src = "Protanopia_graph.png",
+         contentType = "image/png",
+         width = 725,
+         height = 450,
+         alt = "protan bad graph"
+       ))
+     }
+     else if (input$color == "Tritanopia") {
+       return(list(src = "Tritanopia_graph.png",
+         contentType = "image/png",
+         width = 725,
+         height = 450,
+         alt = "tritan bad graph"
+       ))
+     }
+   }, deleteFile = FALSE)
+
+   output$improved.plot <- renderPlot({
+     if (input$color == "Deuteranopia") {
+       new.col <- dichromat(better.col,"deutan")
+       deutan <- my.plot +
+         geom_line(aes(y=t19, color = new.col[1])) +
+         geom_line(aes(y=t36, color = new.col[2])) +
+         geom_line(aes(y=t52, color = new.col[3])) +
+         geom_line(aes(y=t69, color = new.col[4])) +
+         geom_line(aes(y=t83, color = new.col[5])) +
+         geom_line(aes(y=t103, color = new.col[6])) +
+         geom_line(aes(y=t129, color = new.col[7])) +
+         geom_line(aes(y=t155, color = new.col[8])) +
+         geom_line(aes(y=t179, color = new.col[9])) +
+         geom_line(aes(y=t221, color = new.col[10]))
+       print(deutan)
+     }
+     else if (input$color == "Protanopia") {
+       new.col <- dichromat(better.col,"protan")
+       protan <- my.plot +
+         geom_line(aes(y=t19, color = new.col[1])) +
+         geom_line(aes(y=t36, color = new.col[2])) +
+         geom_line(aes(y=t52, color = new.col[3])) +
+         geom_line(aes(y=t69, color = new.col[4])) +
+         geom_line(aes(y=t83, color = new.col[5])) +
+         geom_line(aes(y=t103, color = new.col[6])) +
+         geom_line(aes(y=t129, color = new.col[7])) +
+         geom_line(aes(y=t155, color = new.col[8])) +
+         geom_line(aes(y=t179, color = new.col[9])) +
+         geom_line(aes(y=t221, color = new.col[10]))
+       print(protan)
+     }
+     else if (input$color == "Tritanopia") {
+       new.col <- dichromat(better.col,"tritan")
+       tritan <- my.plot +
+         geom_line(aes(y=t19, color = new.col[1])) +
+         geom_line(aes(y=t36, color = new.col[2])) +
+         geom_line(aes(y=t52, color = new.col[3])) +
+         geom_line(aes(y=t69, color = new.col[4])) +
+         geom_line(aes(y=t83, color = new.col[5])) +
+         geom_line(aes(y=t103, color = new.col[6])) +
+         geom_line(aes(y=t129, color = new.col[7])) +
+         geom_line(aes(y=t155, color = new.col[8])) +
+         geom_line(aes(y=t179, color = new.col[9])) +
+         geom_line(aes(y=t221, color = new.col[10]))
+       print(tritan)
+     }
+   })
+ }
> shinyApp(ui = ui, server = server)
```

Original

Protanopia



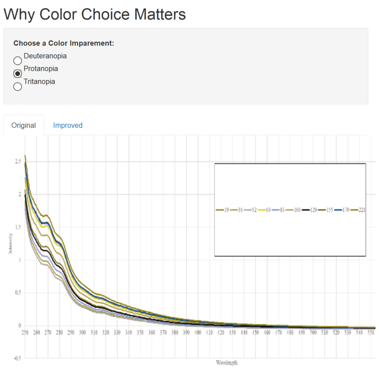
Why Color Choice Matters

Choose a Color Imparement:

● Deuteranopia
○ Protanopia
○ Tritanopia

Original | Improved

Deuteranopia



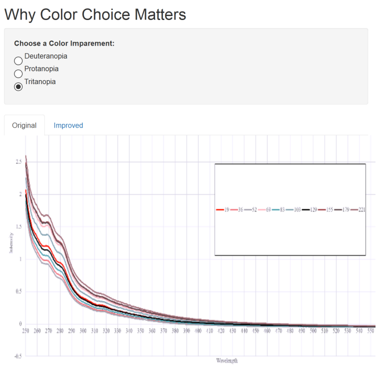
Why Color Choice Matters

Choose a Color Imparement:

○ Deuteranopia
● Protanopia
○ Tritanopia

Original | Improved

Tritanopia



Why Color Choice Matters

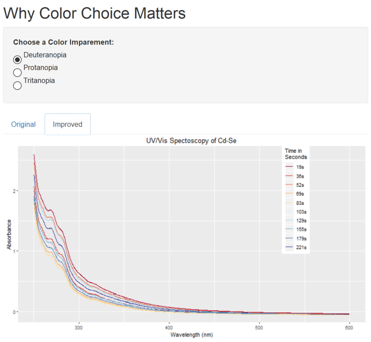
Choose a Color Imparement:

○ Deuteranopia
○ Protanopia
● Tritanopia

Original | Improved

Improved

Protanopia



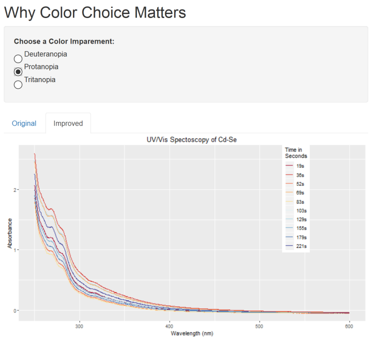
Why Color Choice Matters

Choose a Color Imparement:

● Deuteranopia
○ Protanopia
○ Tritanopia

Original | Improved

Deuteranopia



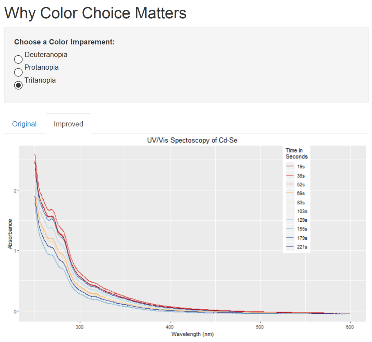
Why Color Choice Matters

Choose a Color Imparement:

○ Deuteranopia
● Protanopia
○ Tritanopia

Original | Improved

Tritanopia



Why Color Choice Matters

Choose a Color Imparement:

○ Deuteranopia
○ Protanopia
● Tritanopia

Original | Improved

(iv) **Using the Grand Tour for an Interactive Clustering** (25 Points):

Load the file *hw03_mystery.csv* into *GGobi*, the *rggobi* R package, or any other standalone software or R package that allows you to apply a **brush–tour** strategy as introduced in the *Grand Tour* section of our lecture notes.

I have asked our computer support people to install *GGobi/rggobi* **on one or two publicly accessible computers in our department. If you are interested in working on this question, but do not have access to** *GGobi/rggobi* **on your own computer, you will hopefully be able to use one of the computers here in the department within the next few days. I will post an announcement in Canvas when** *GGobi/rggobi* **have been installed on a departmental computer.**

My instructions below are written under the assumption that you are using *GGobi*.

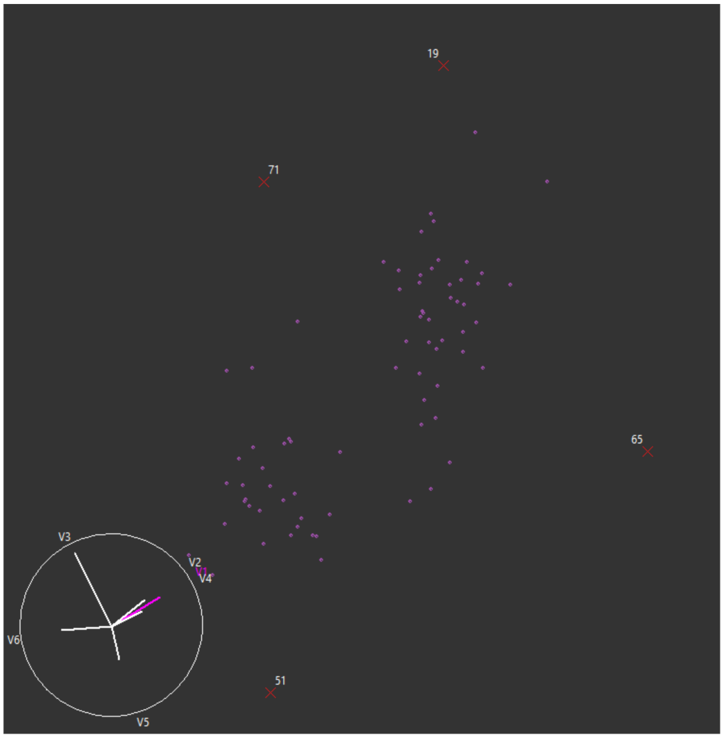
For this question, you do not have to include any R code.

(a) (5 Points) Start the grand tour and select the six variables called *V1, . . ., V6*. Do **not** select the variable *Row*. In the grand tour window, select *Options* → *Show Axes* to show the projection of the variable axes. Adjust the speed of the tour as desired.

First look for obvious outliers in the grand tour. These are single points that behave rather differently from all other points in the tour. Once you have spotted an outlier, stop the tour and brush this point with a big red X. Continue and find the next outlier, etc. There are ≥ 1 and ≤ 5 outliers. How many outliers did you find?

Continue the tour and find a projection where all of your outliers are visible and well separated from the other data points at the same time. Such a projection exists. Do some fine adjustment of this projection with the mouse. Which variables contribute most to this projection? Also include a screenshot (or photo taken with your phone) of this projection as part of your answer. Make sure that the variable axes are visible.

I noticed **4** total outliers that seemed reasonably significant to me. In the screenshot below, *V3* contributes the most, followed by *V6* and *V1*.



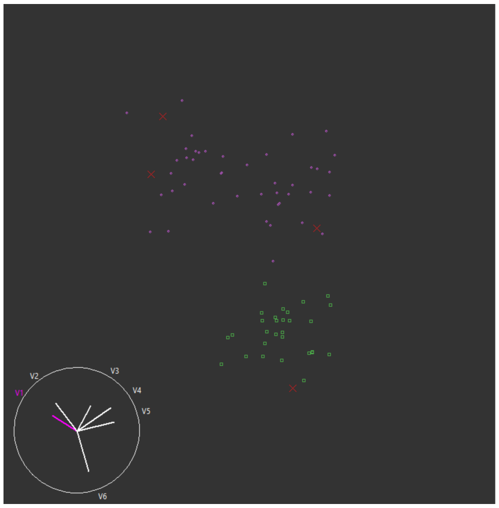
(b) (4 Points) Now focus on clusters via the grand tour. There are ≥ 2 and ≤ 5

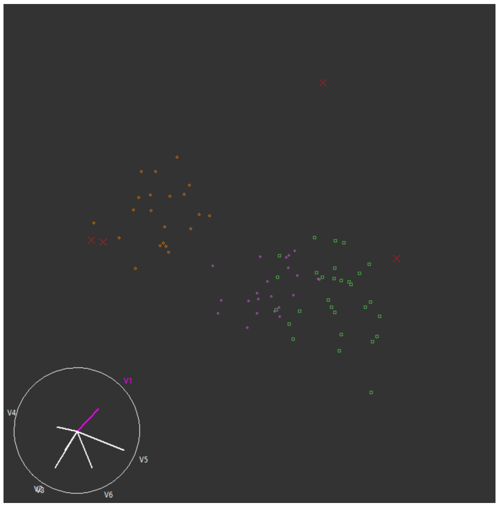
clusters in this data set. Whenever you think you identified a cluster, i.e., a group of points that moves in a different direction, persistently brush it with a new color and symbol.

If you paused the tour a fraction of a second too late, do some adjustment of the projection with the mouse to get back to a more interesting projection. Do not brush your previously detected outliers again!

After each new cluster you detected, take a screenshot (or photo) and include it as part of your answer. As before, make sure that the variable axes are visible. How many main clusters did you detect?

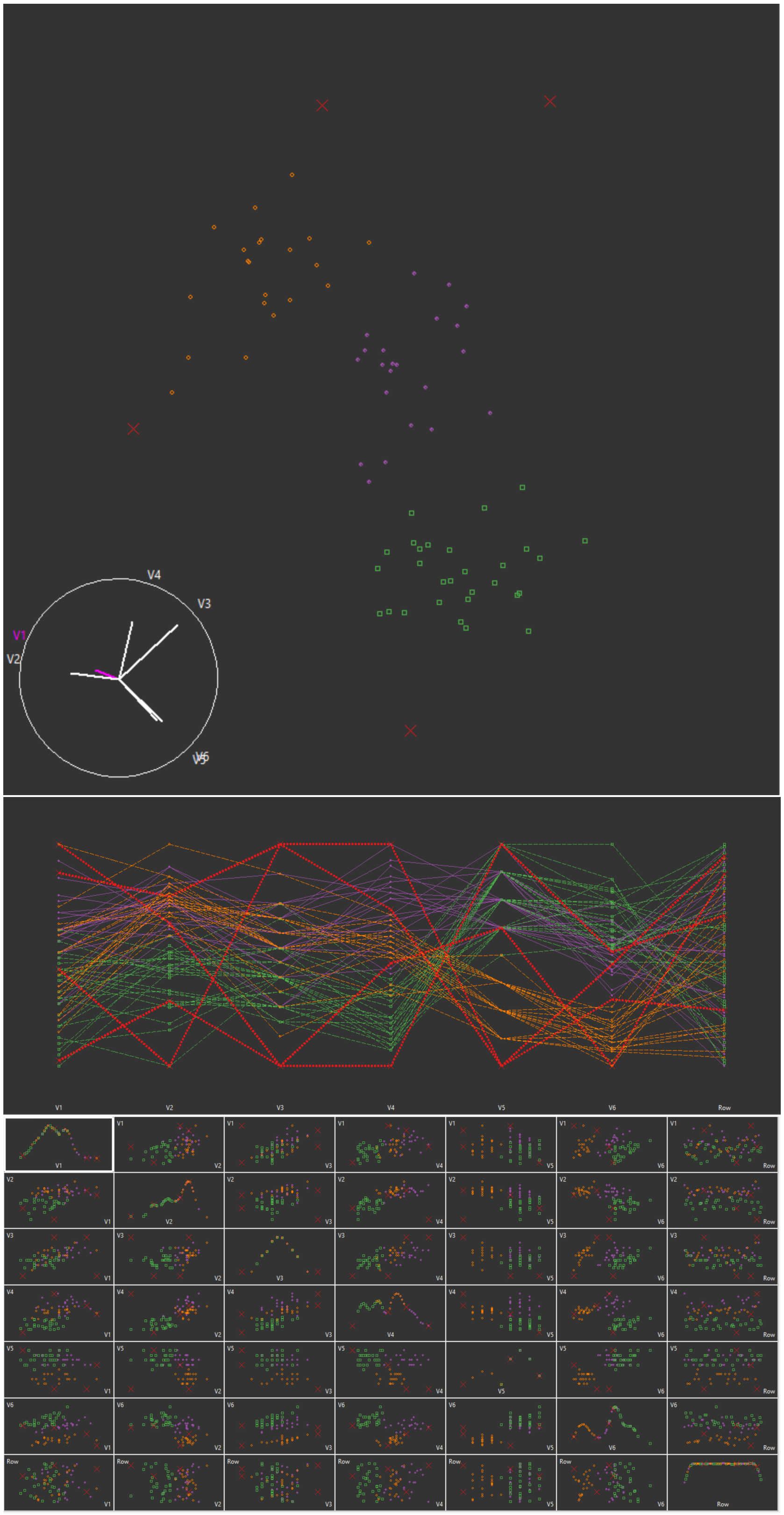
I identified **3** distinct clusters. I've attached two screenshots in the order I identified the groupings below:.





(c) (4 Points) Find a projection in the grand tour that separates your clusters and the outliers reasonably well. There may not be a perfect projection that separates all clusters and the outliers at the same time.

Also activate a parallel coordinate plot (PCP) and a scatterplot matrix of all six variables and *Row*. Make sure that the variables appear in the order *V1, . . ., V6, Row*. If this is not the case pull the axes in the PCP until they appear in the right place in *GGobi*. For the scatterplot matrix, unselect variables and select them in the proper order again in *GGobi*. Include a screenshot of the final projection from the grand tour, the PCP, and the scatterplot matrix.


V4
V3
V1
V2
V6
V5
V1
V2
V3
V4
V5
V6
Row

(d) (4 Points) Revisit your outliers: What are the *Row* numbers for your outliers? Note that *Identify* in *GGobi* provides you with the case number (or row label used by *GGobi*). The *Row* number in our data set is the *GGobi* case number +1. This becomes obvious in *Tools* → *Data Viewer*.

Are you surprised how your outliers look in the PCP and the scatterplot matrix? I would be ... Is there a single axis in the PCP that would have allowed you to detect a specific outlier (which outlier and which axis)? Is there any pairwise scatterplot in the scatterplot matrix that would have allowed you to detect a specific outlier (which outlier and which pairwise scatterplot)? Using the variable axes information from the grand tour may help.

Verbally describe the characteristics of the outliers you detected.

The row number for the outliers I identified are **20, 52, 66,** and **72**. In looking at the PCP and the scatterplot matrix, I was a little surprised (even given our earlier observations) that they were all easily identifiable on *V3* as either the highest or the lowest values. This is true for all of the outliers in both the PCP and the scatterplot matrix.

General characteristics of the outliers is there tendancy to bounce all over the PCP compaired to the tendancies of the clusters that were identified, that is to say that they ignore the overall behavior of the data. In the scatterplot matrix however, it is much more difficult to identify general trends outside of *V3*. In fact, I would go so far as to say that if I didn't already believe they were outliers from other visualizations, I would have had not likely been able to pick them out from just the scatterplot matrix.

(e) (4 Points) Revisit your clusters: How many did you detect visually? Are there any variables in the PCP that would have allowed you to identify the same clusters? Which ones? Is this a clear separation?

Are there any pairwise scatterplots in the scatterplot matrix that would have allowed you to detect the same clusters? Which ones? Is this a clear separation? When you refer to your clusters, identify them by the colors and glyphs you used, e.g., the cluster made up by the "yellow o" and the cluster made up by the "purple +".

I visually detected three clusters, and viewing the PCP, I believe that examining *V5* and *V6* I would have been able to identify the orange o group as distinct. Then, I might have been able to further identify the green □ by noting the pattern between *V4* and *V5*, and gaining evidence of this after examining *V2*.

There are only 2 plots in my scatterplot matrix that might allow me to have identified the three clusters they were *V4*-*V5* and *V4*-*V6*, and even then the seperation is relatively minamal, and thus difficult to identify.

(f) (4 Points) Overall, did we benefit from the grand tour, or could we just have used a PCP and/or a scatterplot matrix? Be specific and justify your answer. Compare which variable axes had most impact in the screenshots (or photos)

you created from the grand tour with subsets of these variables in the PCP and in the pairwise scatterplots.

I believe that we benefited considerably from the use of the Grand Tour. It allowed us the ability to patterns quickly, and with less ambiguity. While I believe that it is definitely possible to identify these patterns in the PCP and the Scatterplot Matrix, the Grand Tour made for much quicker exploritory data analysis. Then, viewing the PCP and the Scatterplot matrix served to confirm those initial observations.

Compairing the Variable axes in the Grand Tour, the PCP and the Scatterplot matrix, it appear that *V3*, *V4*, *V5*, and *V6* had the most impact on the the subsetting of the data as I have identified it.

General Instructions

(i) Create a single pdf document, using R Markdown, Sweave, or knitr. When you take this course at the 6000–level, you have to use LaTeX in combination with Sweave or knitr. You only have to submit this one document.

(ii) Include a title page that contains your name, your A–number, the number of the assignment, the submission date, and any other relevant information.

(iii) Start your answers to each main question on a new page (continuing with the next part of a question on the same page is fine). Clearly label each question and question part.

(iv) Show your R code for each question part!

(v) Before you submit your homework, check that you follow all recommendations from Google's R Style Guide (see `https://google.github.io/styleguide/Rguide.xml`). Moreover, make sure that your R code is consistent, i.e., that you use the same type of assignments and the same type of quotes throughout your entire homework.

(vi) Give credit to external sources, such as stackoverflow or help pages. Be specific and include the full URL where you found the help (or from which help page you got the information). Consider R code from such sources as "legacy code or third–party code" that does not have to be adjusted to Google's R Style (even though it would be nice, in particular if you only used a brief code segment).

(vii) **Not following the general instructions outlined above will result in point deductions!**

(viii) For general questions related to this homework, please use the corresponding discussion board in Canvas! I will try to reply as quickly as possible. Moreover, if one of you knows an answer, please post it. It is fine to refer to web pages and R commands, but do not provide the exact R command with all required arguments or which of the suggestions from a stackoverflow web page eventually worked for you! This will be the task for each individual student!

(ix) Submit your single pdf file via Canvas by the submission deadline. Late submissions will result in point deductions as outlined on the syllabus.