TA Session 9: Data Visualization – Base R

Harris Coding Camp

Summer 2024

General Guidelines

You may encounter some functions we did not cover in the lectures. This will give you some practice on how to use a new function for the first time. You can try following steps:

1. Start by typing `?new_function` in your Console to open up the help page.
2. Read the help page of this `new_function`. The description might be a bit technical for now. That's OK. Pay attention to the Usage and Arguments, especially the argument `x` or `x,y` (when two arguments are required).
3. At the bottom of the help page, there are a few examples. Run the first few lines to see how it works.
4. Apply it in your questions.

It is highly likely that you will encounter error messages while doing this exercise. Here are a few steps that might help get you through it:

1. Locate which line is causing this error first.
2. Check if you have a typo in the code. Sometimes your group members can spot a typo faster than you.
3. If you enter the code without any typo, try googling the error message. Scroll through the top few links see if any of them helps.
4. Try working on the next few questions while waiting for help by TAs.

I. Customising simple plots I

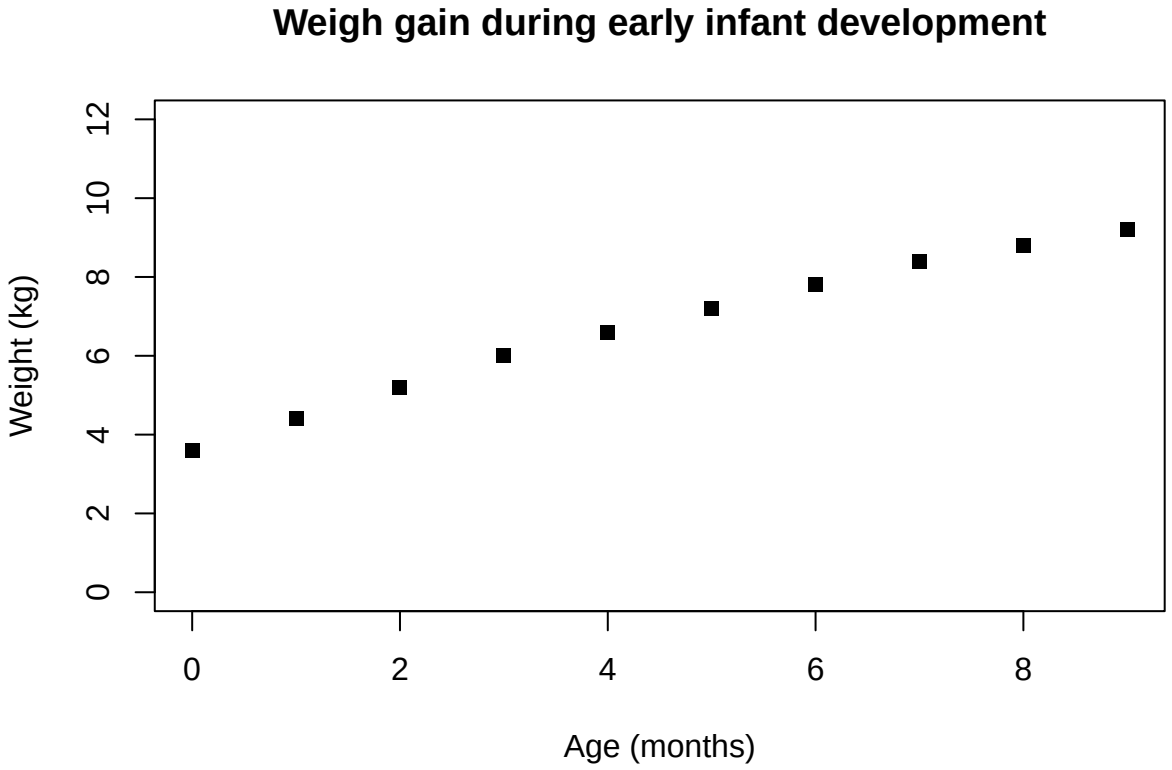
1. Let's plot a baby's growth rate. First, load the dataset `weight_chart.txt` and give it a name (e.g. weight.chart). Don't forget to change the working directory if needed! Then, take a look at the dataset.

```
weight.chart <- read.delim("weight_chart.txt")
weight.chart
```

```
##    Age Weight
## 1    0    3.6
## 2    1    4.4
## 3    2    5.2
## 4    3    6.0
## 5    4    6.6
## 6    5    7.2
## 7    6    7.8
## 8    7    8.4
## 9    8    8.8
## 10   9    9.2
```

2. We're going to use a scatterplot to plot the two values against each other. Replicate the plot by modifying the code below.

```
# adjust the code
plot([fill in] ~ [fill in],
     type = "?",            # plot type
     pch = 15,              # filled square
     cex = 1,               # size of pch symbols
     data = "[fill in]",    # data from...?
     ylim = c(?,?),         # fill in the y-axis limits
     ylab = "[fill in]",
     xlab = "[fill in]",
     main = "Weigh gain during early infant development"
     )
```


Weigh gain during early infant development
Weight (kg)
0
2
4
6
8
10
12
0
2
4
6
8
Age (months)

II. Customising simple plots II

In this exercise, we want to plot out the petunia plants data in different ways.

```
# load data first
library(readr)
flower <- read_csv("flower.csv")
```

```
## Rows: 96 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): treat, nitrogen
## dbl (6): block, height, weight, leafarea, shootarea, flowers
##
## 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.
```

```
flower
```

```
## # A tibble: 96 x 8
##    treat nitrogen block height weight leafarea shootarea flowers
##    <chr> <chr>    <dbl>  <dbl>  <dbl>    <dbl>     <dbl>   <dbl>
##  1 tip   medium       1    7.5   7.62     11.7      31.9       1
##  2 tip   medium       1   10.7  12.1      14.1      46        10
##  3 tip   medium       1   11.2  12.8       7.1      66.7      10
##  4 tip   medium       1   10.4   8.78     11.9      20.3       1
##  5 tip   medium       1   10.4  13.6      14.5      26.9       4
##  6 tip   medium       1    9.8  10.1      12.2      72.7       9
##  7 tip   medium       1    6.9  10.1      13.2      43.1       7
##  8 tip   medium       1    9.4  10.3      14        28.5       6
##  9 tip   medium       2   10.4  10.5      10.5      57.8       5
## 10 tip   medium       2   12.3  13.5      16.1      36.9       8
## # i 86 more rows
```

1. Let's plot this! Generate a scatter plot with `shootarea` on the y-axis and `weight` on the x-axis. Don't forget to also include the labels and title.
2. Add a linear regression line on the same plot.
3. Start with the same plot as above, and then annotate it by red the data points corresponding to weight which is at least 15.

III. Global Attacks Against Aid-Workers

In this part of the exercise (revised from previous Stats I HW), we will examine trends over time in the number of attacks and incidents involving aid-workers. There is a large literature discussing whether providing aid to low and middle income countries also creates incentives for armed groups to use violence in order to appropriate the distributed aid.[1] We will not be trying to answer that question or to quantify the relationship between the amount of aid and violence against aid-workers. Our goal here is a simple description of an annual time-series of data spanning 1997 to 2020. We obtained the data from https://aidworkersecurity.org/ and have simplified it for the purposes of this exercise.[2]

The `aid_workers_security_incidents_annual_level_1997_2020.csv` data set contains the following variables:

Name	Description
`year`	Calendar year
`number_incidents`	The total number of recorded incidents in that year
`total_affected`	The total number of people affected across all incidents in that year

1. Read the data set into R.
2. Create a new variable with the number of affected people per-incident.
3. Create a new variable that is equal to 1 for any year after 2007 and 0 otherwise (that variable should equal to 0 during 1997 to 2007, and equal to 1 during 2008 to 2020).
4. What is the mean value of the total number of affected people during the 1997 to 2007 period? What is that number for the 2008 to 2020 period?
5. Next, you should use the `plot()` command to produce three different graphs. We will start with a graph for the number of incidents (y-axis) in each year (x-axis). Make sure that the axes are labeled and that your plot has an appropriate title.
6. Now produce two additional graphs. One for the total number of affected people in each year, and one for the number of affected people per-incident in each year.
7. In 1-2 sentences, briefly describe the over-time patterns in the first two graphs you produced (the number of incidents and the total number of affected people).
8. In 1-2 sentences, what is qualitatively different about the third graph you produced (the number of affected people per-incident each year)?

[1] For more details, read the introduction and background sections of Crost, Benjamin, Joseph Felter, and Patrick Johnston. 2014. "Aid Under Fire: Development Projects and Civil Conflict." *American Economic Review* 104 (6): 1833-56.

[2] We simply downloaded the entire data set and aggregated over all counties and types of incidents.

IV. (Optional) Revisit: ISLR Chapter 2 Q8

As a quick reminder, this exercise relates to the College data set, which can be found in the file College.csv. It contains a number of variables for 777 different universities and colleges in the US. If you forget some details from this exercise, review TA session 4.

1. Setup – we will repeat parts 1-3 from TA session 4 before we move forward. Fill in your working directory and run the following code. Finally, use `View(college)` to make sure the data has been properly processed.

```
#set your working directory -- fill in your code after this line

#read in the file College.csv using read.csv() with option `stringsAsFactors=T`
college <- read.csv('College.csv', stringsAsFactors = T)

#Give data frame college rownames
rownames(college) <- college[, 1]

# college[, -1] will generate a subset with all but the first column
college <- college[, -1]

# as.factor() can turn a character column to a factor column, so that we can use it to plot later on
college$Private <- as.factor(college$Private)
```

2. (Optional) Use the `pairs()` function to produce a scatterplot matrix of the first ten columns or variables of the data. Recall that you can reference the first ten columns of a matrix A using `A[, 1:10]`.

```
pairs(...)
```

3. Create a new qualitative variable, called `Elite`, by binning the `Top10perc` variable. We are going to divide universities into two groups based on whether or not the proportion of students coming from the top 10% of their high school classes exceeds 50%. Instead of following the steps used in TA session 4, we want to use `ifelse` function. Complete and run the code.

```
college$Elite <- factor(ifelse(...))
```

4. Use the `summary()` function to see how many elite universities there are. Then, use the `plot()` function to produce side-by-side boxplots of `Outstate` versus `Elite`.

5. Use the `hist()` function to produce some histograms with differing numbers of bins/breaks for a few of the quantitative variables. Hint: You may find the command `par(mfrow = c(2, 2))` useful: it will divide the print window into four regions so that four plots can be made simultaneously. Modifying the arguments to this function will divide the screen in other ways.

```
par(mfrow = c(2, 2))
hist(college$Books, col = 2, breaks = 50, xlab = "Books", ylab = "Count")
# complete the following 3 lines
hist(college$...)
hist(college$...)
hist(college$...)
```

6. Continue exploring the data, and provide a brief summary of what you discover.