Welcome to The Carpentries Etherpad!

This pad is synchronized as you type, so that everyone viewing this page sees the same text. This allows you to collaborate seamlessly on documents.

Use of this service is restricted to members of The Carpentries community; this is not for general purpose use (for that, try https://etherpad.wikimedia.org).

Users are expected to follow our code of conduct: https://docs.carpentries.org/topic_folders/policies/code-of-conduct.html

----------------------------------------------------------------------------

**6 September 2021**

**Data organisation with Spreadsheet**
Instructor - Ms Zine Sapula; Helper- Ms Marissa Griesel
Lesson notes for Spreadsheets: https://datacarpentry.org/spreadsheets-socialsci/

**Attendance Register**
Please write your surname, name and email address

- Sebastian Mosidi, Sebastian.mosidi@nwu.ac.za
- Caroline F Ajilogba, carolfad@gmail.com
- Zine Sapula, zine.sapula@ump.ac.za
- Renato Alves, renato.alves@embl.de
- Henok Solomon, hsolomon@uwc.ac.za
- Kristina Michl, kristina.michl@tugraz.at
- Marissa Griesel, griesel.marissa@gmail.com
- Hanne Vlietinck, hanne,vlietinck@uhasselt.be
- Rabin, Fadiyah, fadiyah.r@gmail.com
- Margaret Wanjiku, margaretwmugure@gmail.com
- Mamsi Khantsi, mamsikhantsi@gmail.com
- Barulaganye Hulela,hulelab@biust.ac.bw
- Doris Gombani, dgombani@staff.bou.ac.bw
- Mark de Vos m.a.de.vos@gmail.com
- Ajani Temitope Akinwumi 4112465@myuwc.ac.za
- Zuko Mgugudo zmgugudo@gmail.com
- Olukosi Ibitoye olukosiibitoye@yahoo.com
- Elizabeth Gyamfi, lizzgyamfi@gmail.com
- Federico Milani. federico.milani80@gmail.com

Zoom details - https://carpentries.zoom.us/j/94947751065?pwd=Y1pnaHJSdHdyVUpPNjNRWkdLMGhVQT09
Use passcode 367203

Schedule: https://nwu-eresearch.github.io/2021-09-06-CCZA_DC_2021/

Session 1
Download the following three files

1. SAFI_clean.csv - https://ndownloader.figshare.com/files/11492171
2. SAFI_messy.xlsx- https://ndownloader.figshare.com/files/11502824
3. SAFI_dates.xlsx- https://ndownloader.figshare.com/files/11502827

**Exercise 1: Formatting spreadsheets**
We're going to take a messy version of the SAFI data and describe how we would clean it up.

1. Download the messy data.

2. Open up the data in a spreadsheet program.
3. Notice that there are two tabs. Two researchers conducted the interviews, one in Mozambique and the other in Tanzania. They both structured their data tables in a different way. Now, you're the person in charge of this project and you want to be able to start analyzing the data.
4. With the person next to you, identify what is wrong with this spreadsheet. Discuss the steps you would need to take to clean up the two tabs, and to put them all together in one spreadsheet.

**Important** Do not forget our first piece of advice, to create a new file (or tab) for the cleaned data, never modify your original (raw) data.
After you go through this exercise, we'll discuss as a group what was wrong with this data and how you would fix it.

**Exercise 2: Metadata**
Download a clean version of this dataset and open the file with your spreadsheet program. This data has many more variables that were not included in the messy spreadsheet and is formatted according to tidy data principles.
Discuss this data with a partner and make a list of some of the types of metadata that should be recorded about this dataset. It may be helpful to start by asking yourself, "What is not immediately obvious to me about this data? What questions would I need to know the answers to in order to analyze and interpret this data?"

**Exercise 3: Dates**
Download and open the SAFI_dates.xlsx file. This file contains a subset of the data from the SAFI interviews, including the dates on which the interviews were conducted.
Choose the tab of the spreadsheet that corresponds to the way you format dates in your location (either day first DD_MM_YEAR, or month first MM_DD_YEAR).
Extract the components of the date to new columns. For this we can use the built in Excel functions:

```
=MONTH()
=DAY()
=YEAR()
```

Apply each of these formulas to its entire column. Make sure the new column is formatted as a number and not as a date.
We now have each component of our date isolated in its own column. This will allow us to group our data with respect to month, year, or day of month for our analyses and will also prevent problems when passing data between different versions of spreadsheet software (as for example when sharing data with

collaborators in different countries).

**Opening a CSV file in Excel:**
If you do not import the sheet with the correct column seperators, all of your column headings may be grouped together in one cell. To rectify this, follow these basic steps:

1. Open a new Excel sheet.
2. Click the Data tab, then From Text.
3. Select the CSV file that has the data clustered into one column.
4. Select Delimited, then make sure the File Origin is Unicode UTF-8.
5. Select Comma (this is Affinity's default list separator). ...
6. Finally, click Finish.
7. Remember to Save your document!

**Exercise 4: Data validation**
Apply a new data validation rule to one of the other numeric columns in this data table. Discuss with the person sitting next to you what a reasonable rule would be for the column you've selected. Be sure to create an informative input message.

**To turn prevent invalid entries, if the named range has blank cells (text values):**

1. Select the cell that contains a data validation list.
2. Choose Data|Validation.
3. On the Settings tab, remove the check mark from the Ignore blank box.
4. Click OK.

Please complete the sticky notes with one positive and one negative thing:
https://jamboard.google.com/d/1GA919O4UIKC5Y3K-RSliSHYnfTTGqeHK2prTSB1mA08/edit?usp=sharing

--------------------------------------------------------------------------------
**7 September 2021**

**Day 2: Data Cleaning with OpenRefine**
https://datacarpentry.org/openrefine-socialsci/

**Zoom details:** https://carpentries.zoom.us/j/94947751065?pwd=Y1pnaHJSdHdyVUpPNjNRWkdLMGhVQT09
Use passcode 367203

**Schedule**: https://nwu-eresearch.github.io/2021-09-06-CCZA_DC_2021/

Download and install **OpenRefine**: https://openrefine.org/download.html
If necessary install **Java** from https://java.com
Download **data**: https://ndownloader.figshare.com/files/7823341

**Attendance Register**
**Please write your surname, name and email address**

- Marissa Griesel, griesel.marissa@gmail.com

- Nkwanyana, Akhona, 3223935@myuwc.ac.za
- Henok Solomon, hsolomon@uwc.ac.za
- Zine Sapula, zine.sapula@ump.ac.za
- Zuko Mgugudo
- Temitope Ajani, 4112465@myuwc.ac.za
- Barulaganye Hulela,hulelab@biust.ac.bw
- Renato Alves, renato.alves@embl.de
- Kristina Michl, kristina.michl@tugraz.at
- Fadiyah Rabin, fadiyah.r@gmail.com
- namoune rachida,namounerachida@gmail.com
- Khantsi Mamsi, mamsikhantsi@gmail.com
- Elizabeth Gyamfi, lizzgyamfi@gmail.com
- Doris Gombani, dgombani@staff.bou.ac.bw
- Mark de Vos@ru.ac.za

**Session 1: Introduction to OpenRefine**

Additional info on Clustering with OpenRefine:
https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth

More Help Documentation: https://docs.openrefine.org/

Tea break - 10:20-10:45

**Please send us Feedback:**
https://jamboard.google.com/d/1AQrmr1oVXMeDroPpUBm4MOGa587ZtXSa6JNdisYOavI/edit?usp=sharing

**Session 2 : Data Analysis and Visualisation with R**

Downloads: https://datacarpentry.org/r-socialsci/setup.html
Installing tidyverse:

```
install.packages("tidyverse")
library(tidyverse)
```

Download and save the following files to your computer:
https://datacarpentry.org/r-socialsci/data/SAFI.json (right click and "save as")

**8 September 2021**

**Day 3: Data Analysis and Visualisation with R**

Zoom details: https://carpentries.zoom.us/j/94947751065?pwd=Y1pnaHJSdHdyVUpPNjNRWkdLMGhVQT09
Use passcode 367203

Schedule: https://nwu-eresearch.github.io/2021-09-06-CCZA_DC_2021/

Lesson content: https://datacarpentry.org/r-socialsci/

**Files for the R Lesson**

The lesson uses SAFI_clean.csv. The direct download link for this file is: https://ndownloader.figshare.com/files/11492171.

The json episode uses SAFI.json. The downlink link is https://datacarpentry.org/r-socialsci/data/SAFI.json.

**Attendance Register**
**Please write your surname, name and email address**

- Temitope Ajani, 4112465@myuwc.ac.za
- Henok Solomon, hsolomon@uwc.ac.za
- Barulaganye Hulela,hulelab@biust.ac.bw
- Zuko Mgugudo, zmgugudo@gmail.com
- Nkwanyana, Akhona, 3223935@myuwc.ac.za
- Mark de Vos, m.devos@ru.ac.za
- Fadiyah Rabin, fadiyah.r@gmail.com
- Renato Alves
- Kristina Michl, kristina.michl@tugraz.at
- Marissa Griesel, griesel.marissa@gmail.com
- Elizabeth Gyamfi
- Doris Gombani, dgombani@staff.bou.ac.bw
- Caroline F Ajilogba, carolfad@gmail.com
- Zine Sapula, zine.sapula@ump.ac.za
- Rachida Namoune, namounerachida@gmail.com
- Mark de vos m.devos@ru.ac.za

**Episode 1-Before we Start**

**Episode 2-Introduction to R**

Exercise 1:
Create two variables r_length and r_width and assign them values. It should be noted that, because length is a built-in R function, R Studio might add “()” after you type length and if you leave the parentheses you will get unexpected results. This is why you might see other programmers abbreviate common words. Create a third variable r_area and give it a value based on the current values of r_length and r_width. Show that changing the values of either r_length and r_width does not affect the value of r_area.

Exercise 2:
We’ve seen that atomic vectors can be of type character, numeric (or double), integer, and logical. But what happens if we try to mix these types in a single vector?

Exercise 3:

What will happen in each of these examples? (hint: use class() to check the data type of your objects):

```
num_char <- c(1, 2, 3, "a")
num_logical <- c(1, 2, 3, TRUE)
char_logical <- c("a", "b", "c", TRUE)
tricky <- c(1, 2, 3, "4")
```

Exercise 4:
How many values in combined_logical are "TRUE" (as a character) in the following example:

```
num_logical <- c(1, 2, 3, TRUE)
char_logical <- c("a", "b", "c", TRUE)
combined_logical <- c(num_logical, char_logical)
```

Exercise 5:
Extract the elements where the household members are greater than or equal to 4 and less than or equal to 7

```
household_members[household_members>=4 & household_members<=7]
```

.. that extracted 7 and 6

How would we extract only possessions "car" and "bicycle"?
By using the == (equals to) and | (or) operators:

```
possessions[possessions == "car" | possessions == "bicycle"]
```

This will return

```
[1] "car"     "bicycle"
```

in the console.

Exercise 6:

- Using this vector of rooms, create a new vector with the NAs removed.

```
rooms <- c(1, 2, 1, 1, NA, 3, 1, 3, 2, 1, 1, 8, 3, 1, NA, 1)
```

- Use the function median() to calculate the median of the rooms vector.

- Use R to figure out how many households in the set use more than 2 rooms for sleeping.

Solutions?
newRooms<-rooms[!is.na(rooms)] # I chose to set up a new vector.  Can't manage to do it in a single command tho... oops... was I supposed to just post the output?
median(newRooms)

One possible way:

```
rooms <- c(1, 2, 1, 1, NA, 3, 1, 3, 2, 1, 1, 8, 3, 1, NA, 1)
rooms_no_na <- rooms[!is.na(rooms)]
# or
rooms_no_na <- na.omit(rooms)
# 2.
median(rooms, na.rm = TRUE)
```

Output will then be:

```
[1] 1
```

# 3.

```
rooms_above_2 <- rooms_no_na[rooms_no_na > 2]
length(rooms_above_2)
```

Output will then be:

```
[1] 4
```

**You can revisit the lesson from today here**:
https://datacarpentry.org/r-socialsci/01-intro-to-r/index.html

**Please give us feedback on today's lesson:**
https://jamboard.google.com/d/1sAluGII8F16UEcmzbKinF5izFFtX8pFYKk9-89ha094/edit?usp=sharing
------------------------------------------------------------------------------------------------------------

**9 September**

**Day 4**

**Episode 3-Starting with Data**

Zoom details: https://carpentries.zoom.us/j/94947751065?pwd=Y1pnaHJSdHdyVUpPNjNRWkdLMGhVQT09
Use passcode 367203

Schedule: https://nwu-eresearch.github.io/2021-09-06-CCZA_DC_2021/

Lesson content: https://datacarpentry.org/r-socialsci/

Attendance Register
Please write your surname, name and email address
Ajani, Temitope 4112465@myuwc.ac.za
Kristina Michl, kristina.michl@gmx.at
Fuloria, Sanjay, sanjayfuloria@gmail.com
zine.sapula,zine.sapula@ump.ac.za
Zuko Mgugudo
Henok Solomon, hsolomon@uwc.ac.za
Barulaganye Hulela,hulelab@biust.ac.bw
Khantsi Mamsi, mamsikhantsi@gmail.com
Nkwanyana, Akhona, 3223935@myuwc.ac.za
Rabin, Fadiyah, fadiyah.r@gmail.com
Elizabeth Gyamfi
Doris Gombani, dgombani@staff.bou.ac.bw
Rachida Namoune, namounerachida@gmail.com
m.devos@ru.ac.za mark de vos

Files for the R Lesson

The lesson uses SAFI_clean.csv. The direct download link for this file is: https://ndownloader.figshare.com/files/11492171.

The json episode uses SAFI.json. The downlink link is https://datacarpentry.org/r-socialsci/data/SAFI.json.

Read about tibbles at

#Getting  Interview data to console

```
getwd()
library(tidyverse)
library(here)

interviews <- read_csv(
  here("data", "SAFI_clean.csv"),
  na = "NULL")
```

Follow the code : https://datacarpentry.org/r-socialsci/02-starting-with-data/index.html
LAST Code before the exercise:  interviews["village"]

Exercise 1

1. Create a tibble (interviews_100) containing only the data in row 100 of the interviews dataset.
2. Notice how nrow() gave you the number of rows in the tibble?
   - Use that number to pull out just that last row in the tibble.
   - Compare that with what you see as the last row using tail() to make sure it's meeting expectations.
   - Pull out that last row using nrow() instead of the row number.
   - Create a new tibble (interviews_last) from that last row.
3. Using the number of rows in the interviews dataset that you found in question 2, extract the row that is in the middle of the dataset. Store the content of this middle row in an object named interviews_middle. (hint: This dataset has an odd number of rows, so finding the middle is a bit trickier than dividing n_rows by 2. Use the median( ) function and what you've learned about sequences in R to extract the middle row!
4. Combine nrow() with the - notation above to reproduce the behavior of head(interviews), keeping just the first through 6th rows of the interviews dataset.

Solution

*## 1.*

```
interviews_100 <- interviews[100, ]

## 2.# Saving `n_rows` to improve readability and reduce duplication

n_rows <- nrow(interviews)
interviews_last <- interviews[n_rows, ]

## 3.

interviews_middle <- interviews[median(1:n_rows), ]
interviews_middle

## 4.
interviews_head <- interviews[-(7:n_rows), ]
interviews_head
head(interviews)
```

Exercise 2

- Rename the levels of the factor to have the first letter in uppercase: “No”,”Undetermined”, and “Yes”.
- Now that we have renamed the factor level to “Undetermined”, can you recreate the barplot such that “Undetermined” is last (after “Yes”)?

Solution:

```
## Rename levels. Note we need to keep the original level ordering when renaming.levels(memb_assoc) <- c("No", "Undetermined", "Yes")

## Reorder levels. Note we need to use the new level names.memb_assoc <- factor(memb_assoc, levels = c("No", "Yes", "Undetermined"))plot(memb_assoc)
```

Exercise 3

**Session 4: Formatting Dates**

```
select(interviews, village, no_membrs, months_lack_food)
select(interviews, village:respondent_wall_type)
filter(interviews, village == "Chirodzo")

##Pipe shortcut  ctrl+shift+M
```

Exercise 1
Using pipes, subset the interviews data to include interviews where respondents were members of an irrigation association (memb_assoc) and retain only the columns affect_conflicts, liv_count, and no_meals.

Solution:

Exercise 2

Create a new dataframe from the interviews data that meets the following criteria: contains only the village column and a new column called total_meals containing a value that is equal to the total number of meals served in the household per day on average (no_membrs times no_meals). Only the rows where total_meals is greater than 20 should be shown in the final dataframe.

**Hint**: think about how the commands should be ordered to produce this data frame!

Solution:

```
interviews_total_meals <- interviews %>%
   mutate(total_meals = no_membrs * no_meals) %>%
   filter(total_meals > 20) %>%
   select(village, total_meals)
```

Exercise 3:

1. How many households in the survey have an average of two meals per day? Three meals per day? Are there any other numbers of meals represented?

2. Use group_by() and summarize() to find the mean, min, and max number of household members for each village. Also add the number of observations (hint: see ?n).

3. What was the largest household interviewed in each month?

Overall feedback for Day 4

Feedback
https://jamboard.google.com/d/1JgoflDhBUIn2F6tGxMXzQojTgYUePVnj3bV2F3832fA/edit?usp=sharing

Day 5, September 10, 2021

Zoom details: https://carpentries.zoom.us/j/94947751065?pwd=Y1pnaHJSdHdyVUpPNjNRWkdLMGhVQT09
Use passcode 367203

Schedule: https://nwu-eresearch.github.io/2021-09-06-CCZA_DC_2021/

Lesson content: https://datacarpentry.org/r-socialsci/

Attendance Register
Please write your surname, name and email address

- Marissa Griesel, griesel.marissa@gmail.com
- Sanjay Fuloria, sanjayfuloria@gmail.com
- Henok Solomon, hsolomon2uwc.ac.za
- Zuko Mgugudo, zmgugudo@gmail.com

- Ajani Temitope, 41122465@myuwc.ac.za
- Barulaganye Hulela,hulelab@biust.ac.bw
- Mark de Vos m.devos@ru.ac.za
- Nkwanyana, Akhona, 3223935@myuwc.ac.za
- Sebastian Mosidi, Sebastian.mosidi@nwu.ac.za
- Elizabeth Gyamfi
- Zine Sapula, zine.sapula@ump.ac.za
- Doris Gombani, dgombani@staff.bou.ac.bw
- Rabin, Fadiyah, fadiyah.r@gmail.com
- Caroline F Ajilogba,carolfad@gmail.com
- Namoune Rachida, namounerachida@gmail.com

Exercise 1

1. Create a new dataframe (named interviews_months_lack_food) that has one column for each month and records TRUE or FALSE for whether each interview respondent was lacking food in that month.

Exercise 2

1. How many months (on average) were respondents without food if they did belong to an irrigation association? What about if they didn’t?

---------------------------------------------

**Code to create the new file for the next session:**

```
interviews_plotting <- interviews %>%
## pivot wider by items_owned
  separate_rows(items_owned, sep = ";") %>%
## if there were no items listed, changing NA to no_listed_items
  replace_na(list(items_owned = "no_listed_items")) %>%
  mutate(items_owned_logical = TRUE) %>%
  pivot_wider(names_from = items_owned,
              values_from = items_owned_logical,
              values_fill = list(items_owned_logical = FALSE)) %>%
## pivot wider by months_lack_food
  separate_rows(months_lack_food, sep = ";") %>%
  mutate(months_lack_food_logical = TRUE) %>%
  pivot_wider(names_from = months_lack_food,
              values_from = months_lack_food_logical,
              values_fill = list(months_lack_food_logical = FALSE)) %>%
## add some summary columns
  mutate(number_months_lack_food = rowSums(select(., Jan:May))) %>%
  mutate(number_items = rowSums(select(., bicycle:car)))
```

```
write_csv (interviews_plotting, file = "data_output/interviews_plotting.csv")
##remember to change the path of the directory "data_output" to your working directory
##(you can use getwd() to see the path to yours)
```

------------------------------

Data visualization

Exercise 3

1. Use what you just learned to create a scatter plot of rooms by village with the respondent_wall_type showing in different colours. Does this seem like a good way to display the relationship between these variables? What other kinds of plots might you use to show this type of data?

**A possible solution:**

```
interviews_plotting %>%
  ggplot(aes(x = village, y = rooms)) +
  geom_jitter(aes(color = respondent_wall_type),
              alpha = 0.5,
              width = 0.2,
              height = 0.2)
```

Exercise 4

1. Boxplots are useful summaries, but hide the *shape* of the distribution. For example, if the distribution is bimodal, we would not see it in a boxplot. An alternative to the boxplot is the violin plot, where the shape (of the density of points) is drawn.

- Replace the box plot with a violin plot; see geom_violin().

**Possible solution:**

```
interviews_plotting %>%
  ggplot(aes(x = respondent_wall_type, y = rooms)) +
  geom_violin(alpha = 0) +
  geom_jitter(alpha = 0.5, color = "tomato")
```

2. So far, we’ve looked at the distribution of room number within wall type. Try making a new plot to explore the distribution of another variable within wall type.

- Create a boxplot for liv_count for each wall type. Overlay the boxplot layer on a jitter layer to show actual measurements.

**Possible solution:**

```
interviews_plotting %>%
   ggplot(aes(x = respondent_wall_type, y = liv_count)) +
   geom_boxplot(alpha = 0) +
   geom_jitter(alpha = 0.5, width = 0.2, height = 0.2)
```

3. Add colour to the data points on your boxplot according to whether the respondent is a member of an irrigation association (memb_assoc).

**Possible solution:**

```
interviews_plotting %>%
  ggplot(aes(x = respondent_wall_type, y = liv_count)) +
  geom_boxplot(alpha = 0) +
  geom_jitter(aes(color = memb_assoc), alpha = 0.5, width = 0.2, height = 0.2)
```

Exercise 5

Create a bar plot showing the proportion of respondents in each village who are or are not part of an irrigation association (memb_assoc). Include only respondents who answered that question in the calculations and plot. Which village had the lowest proportion of respondents in an irrigation association?

----------------------------

Please remember to complete the POST WORKSHOP SURVEY:
https://carpentries.typeform.com/to/UgVdRQ?slug=2021-09-06-CCZA_DC_2021