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



---

2023-12-13
Welcome! Please open this etherpad and sign in

0. jason tallant, data manager, UM Biological Station, data management/landscape ecology/remote sensing

1. sarah arcos, postdoc, UM microbiology and immunology, virus evolution, influenza

2. Madisyn Guza, Lab tech, EEB Baucom Lab - Plant genetics

3. Kelley Rivenburgh, graduate student, Social Work, Program Evaluation & Applied Research pathway

4. Cindy Sopko, Neuroimaging Core Manager, Michigan Alzheimer's Disease Center-UMich, data management/data wrangling/pipeline creation

5. Nicholas Birdsong, Political Science & Public Policy PhD student

6. Kyle Clark, University of Michigan Library, Stamps School of Art & Design

7. Rain Lemieux, undergraduate student

8. Ishika, Graduate student, Department of Biophysics

9. Brett Meyer, Graduate student, Museum of Anthropological Archaeology

10. Greta Colford, Graduate Student, Mechanical Engineering

11. Elena Lee, graduate student, Earth and Environmental Sciences (IsoPaleo Lab), paleoclimate/isotope geochemistry/landscape evolution

12. Emad Sanei, PhD Candidate, Environmental Engineering, Earth and Ecosystem Science Program, Central Michigan University

13. Isabella Camilleri, MCDB Master's student, microbiology research

14. Qilong WANG, undergraduate student, LSA

15. Yewande Taiwo, Neuroimaging Data Specialist, Michigan Alzheimer's Disease Center,

16. Richie Hodel, Smithsonian/UMich EEB Dept, manipulating genomic data

17. tal ness, Social Sciences Resident Librarian and Linguistics Librarian, Hatcher Library

18. Jinny Yang, PhD student, EEB department, microbial ecology

19. Katherine Colleran, Research Area Specialist, U-M Direct Brain Interface Lab

20. Wenzhen Zhu, Postdoc MMED dept.

---

# R for Plotting

un_reports: https://github.com/UMSWC/curriculum/raw/gh-pages/files/un-report.zip

For OneDrive users: you can find your desktop and the un-report folder by going to: C -> Users -> uniqname -> OneDrive umich -> Desktop -> un-report

R Cheatsheets: https://posit.co/resources/cheatsheets/

Contents of gdp_population.r

```
library(tidyverse)
gapminder_1997 <- read_csv("gapminder_1997.csv")
View(gapminder_1997)

name <- "Ben"
name

age <- 26
age

name_character <- "Harry Potter"
name_character

test <- read_csv("data/gapminder_1997.csv")

Sys.Date() #outputs the current date, used for knowing when I last ran

getwd() #output current directory

sum(5,6)
```

```
round(3.1415)

round(3.1415, digits = 2)

read_csv(file = 'gapminder_1997.csv')
```

#Use code comments to provide context for the work that we are doing in our current script

```
ggplot(data=gapminder_1997) +
  aes(x=gdpPercap) +
  labs(x = 'GDP Per Capita') +
  aes(y = lifeExp) +
  labs(y = "Life Expectancy") +
  geom_point()
  labs(title = "Do people in wealthy countries live longer?")
  aes(color = continent) +
  scale_color_brewer(palette = "Set1") +
  aes(size = pop/1000000) +
  labs(size = "Population (in millions)") +
  aes(shape = continent)


ggplot(data=gapminder_1997) +
  aes(x=gdpPercap, y = lifeExp, color = continent, size = pop/1000000) +
  labs(x = "GDP Per Capita", y = "Life Expectancy",
       title = "Do people in wealthy countries live longer",
       size = "Population (in millions)") +
  geom_point() +
  scale_color_brewer(palette = "Set1")

gapminder_data <- read_csv("gapminder_data.csv")
dim(gapminder_data)
head(gapminder_data)

R plotting day 1 exercise
x axis - year
y axis - life expectancy,
color - continent
```

plot data in points

Solution (multiple correct solutions exist, this is just an option)

```
ggplot(gapminder_data, aes(x = year, y = lifeExp, color = continent)) +
  geom_point() +
  scale_color_brewer(palette = "Set1") +
  labs(x = "Year", y = "Life Expectancy")
```

# Unix Shell

PS1=`$ '
pwd

Glossary of unix commands

- root: the very top of the file system tree
- absolute path: the location of a specific file or directory starting from the root of the file system tree
- relative path: the location of a specific file or directory starting from where you currently are in the file system tree

- pwd: Print working directory - prints the *absolute path* from the *root* directory to the directory where you currently are.
- ls: List files - lists files in the current directory. You can provide a path to list files to another directory as well (ls [path]).
- cd [path]: Change directories - move to another folder.
- mkdir: Make directory - creates a new directory
- ..: This will move you up one level in the file system tree
- mv: Move - move a file to a new location (mv [file] [/path/to/new/location]) OR remaning a file (mv [oldfilename] [newfilename])
- less: - quick way to view a document without using a full text editor
- man: Manual - allows you to view the bash manual for another command (e.g. man ls)
- -h/--help: Help - argument that pulls up the help manual for a program
- nano: a user-friendly text editor
- : Wildcard - matches zero of more characters in a filename

Challenge 1: using tha manual pages

in order of size: ls -S
in order of last time they were edited: ls -t
more information about files: ls -l
more information with unit suffixes: ls -l -h OR ls -lh
in order of size and more information about files: ls -Sl

size-> ls -s, time-> ls -t,

1. ls -S
2. ls -t
3. ls -l
4. ls -l -h
5. ls -S -l

1. ls - S

2.  ls - t
3.  ls - l
4. ls -l -h
5. ls -S -l

1. ls -S
2. ls -t
3. ls -l
4.ls -l -h
5. ls -S -l

1. ls -S
2. ls -t
3. ls -l
4. ls -h
5. ls -s -T

1. ls -S
2. ls -t
3. ls -l
4. ls -l -h
5. ls -l -S

Challenge 2: paths

Madisyn Guza : 5,7, 8, 9

Kelley Rivenburgh 5, 8, 9

Cindy Sopko

Nicholas Birdsong 8, 9, 5, 7

Kyle Clark

Rain Lemieux: 9, 8, 5, 7

Ishika 5, 7, 8, 9

Brett Meyer 5, 8, 9

Greta Colford, 5, 7, 8, 9

Elena Lee

Emad Sanei

Isabella Camilleri: 5, 7, 8, and 9

Qilong WANG：5, 7, 8 and 9

Yewande Taiwo

tal ness

Jinny Yang

Katherine Colleran: 5, 7, 8, 9

Wenzhen Zhu

# Day 2

Attendance

1. Yewande Taiwo
2. tal ness
3. Madisyn Guza
Cindy Sopko
5. Rain Lemieux :)
Kelley Rivenburgh
7. Kyle Clark
8. Qilong WANG
9. Greta Colford
10. Emad Sanei
11. Qinnan Yang
12. Elena Lee
13. Isabella Camilleri
14. Brett Meyer
15.
16. Katherine Colleran
17. Wenzhen Zhu

## R for Data Analysis

```
library(tidyverse)
gapminder_data <- read_csv("gapminder_data.csv")
```

```
summarize(gapminder_data, averageLifeExp = mean(lifeExp))
```

```
gapminder_data %>%
```

```
    • summarize(averageLifeExp = mean(lifeExp))

# this is a brief list of notes to myself
# %>%
# shift + command + m
# shift + control + m

gapminder_data_summarized <- gapminder_data %>%
    • summarize(averageLifeExp = mean(lifeExp))

# danger!
gapminder_data <- gapminder_data %>%
    • summarize(averageLIfeExp = mean(lifeExp))
gapminder_data <- read_csv("gapminder_data.csv")

gapminder_data %>%
  summarize(recent_year = max(year))

#filter: filter a dataset for observations that match a condition
gapminder_data %>%
  filter(year == 2007) %>%
  summarize(averageLifeExp = mean(lifeExp))
```

Exercise 1: what is the mean GDP per capita for the first year in the data set?

1. Yewande Taiwo
2. tal ness
3. Madisyn Guza = 3725
Cindy Sopko = 3725
5. Rain Lemieux :)

```
gapminder_data %>%
  summarize(EarliestYear=min(year))
gapminder_data %>%
  filter(year==1952) %>%
  summarize(average=mean(gdpPercap))
```

Answer= 3725
Kelley Rivenburgh 3725.276
7. Kyle Clark
8. Qilong WANG 3725.

```
gapminder_data %>%
  filter(year == min(year)) %>%
  summarize(average = mean(gdpPercap))
```

9. Greta Colford 3725
10. Emad Sanei 3725

11. Qinnan Yang
12. Elena Lee 3725
13. Isabella Camilleri 3725.
14. Brett Meyer 3725
15. Ishika 3725
16. Katherine Colleran 3725
17. Wenzhen Zhu 3725.

---

```
#One possible solution:
#What is the average GDP per capita for the first year in the dataset
gapminder_data %>%
  filter(year == min(year)) %>%
  summarize(averageGDP = mean(gdpPercap))
```

Pre tidy (thinking about what the pipe is doing):
The above code can be written without pipes in two other ways:

1. this option relies on nesting the functions you wish to apply to the data (this option works, but is not necessarily easy to read, which was the goal of the tidyverse packages)

```
summarize(filter(gapminder_data, year == min(year)), averageGDP =
mean(gdpPercap))
```

2. this option saves each intermediate step as a new object

```
gapminder_filtered <- filter(gapminder_data, year == min(year))
gapminder_summarized <- summarize(gapminder_filtered, averageGDP =
mean(gdpPercap))
```

---

#group_by command

```
gapminder_data %>%
```

- `group_by(year) %>%`
- `summarize(average = mean(lifeExp))`

---

Challenge: What is the average life expectancy by continent?

1. Yewande Taiwo
2. tal ness
3. Madisyn Guza :

Africa 48.9
Americas 64.7
Asia 60.1
Europe 71.9

Oceania 74.3
Cindy Sopko1
Africa 48.9
2 Americas 64.7
3 Asia 60.1
4 Europe 71.9
5 Oceania 74.3
5. Rain Lemieux :)
Africa 48.92 Americas 64.73 Asia 60.14 Europe 71.95 Oceania 74.3
Kelley Rivenburgh
Africa 48.9, Americas 64.7, Asia 60.1, Europe 71.9, Oceania 74.3
7. Kyle Clark
8. Qilong WANG
9. Greta Colford
10. Emad Sanei
11. Qinnan Yang
12. Elena Lee
13. Isabella Camilleri

- 1 Africa 48.9
- 2 Americas 64.7
- 3 Asia 60.1
- 4 Europe 71.9
- 5 Oceania 74.3

14. Brett Meyer
Africa 48.9
Americas 64.7
Asia 60.1
Europe 71.9
Oceania 74.3
15. Ishika Africa 48.9 Americas 64.7 Asia 60.1 Europe 71.9 Oceania 74.3
16. Katherine Colleran
1 Africa 48.9
2 Americas 64.7
3 Asia 60.1
4 Europe 71.9
5 Oceania 74.3
17. Wenzhen Zhu1 Africa 48.9
2 Americas 64.73 Asia 60.14 Europe 71.95 Oceania 74.3

One solution:

```
gapminder_data %>%
  group_by(continent) %>%
  summarize(average = mean(lifeExp))
```

---

```
gapminder_data %>%
```

```
  group_by(continent) %>%
  summarize(average = mean(lifeExp), min = min(lifeExp))

#mutate

gapminder_data %>%
  mutate(gdp = pop*gdpPercap)

gapminder_data %>%
  mutate(gdp = pop*gdpPercap,
         popInMullions = pop/1000000)

# select

gapminder_data %>%
  select(pop, year)

gapminder_data %>%
  select(-continent)
```

Challenge: produce a data frame that has only the columns country, continent, year, and lifeExp

1. Yewande Taiwo
2. tal ness
3. Madisyn Guza :

```
gapminder_data %>%
```

- select(-pop,-gdpPercap)

Cindy Sopko
5. Rain Lemieux :)

```
gapminder_data %>%
  select(country, continent, year, lifeExp)
```

Kelley Rivenburgh
7. Kyle Clark
8. Qilong WANG

```
gapminder_data %>%
  select(country, continent, year, lifeExp)
```

9. Greta Colford

- gapminder_data %>%
- select(-gdpPercap,-pop)

10. Emad Sanei
11. Qinnan Yang
12. Elena Lee

```
gapminder_data %>%
  select(-gdpPercap, -pop)
```

13. Isabella Camilleri

- gapminder_data %>%

- select(-pop, -gdpPercap)
- country year continent lifeExp
- <chr> <dbl> <chr> <dbl>

14. Brett Meyer
15.Ishika
16. Katherine Colleran
gapminder_data %>%
select(-pop, -gdpPercap)
17. Wenzhen Zhu

- gapminder_data %>% dsfsdfadfadfasdfasdfdf
- select(country, continent, year, lifeExp)

One solution:

```
gapminder_data %>%
  select(-pop, -gdpPercap)

gapminder_data %>%
  select(year, starts_with("c"))

gapminder_data %>%
  select(ends_with("p"))

?select

gapminder_data %>%
  select(year, starts_with("con"))
```

Stringr: a package for string manipulation in R
https://rstudio.github.io/cheatsheets/html/strings.html

```
#filter country start with "A"
library(stringr)
gapminder_data %>%
  filter(str_starts(.$country, "A"))

gapminder_data %>%
  select(country, continent, year, lifeExp) %>%
  pivot_wider(names_from = year, values_from = lifeExp)

# new dataset
getwd()

gapminder_data_2007 <- read_csv("gapminder_data.csv") %>%
  filter(year == 2007 & continent == "Americas")

tmp <- read_csv("co2-un-data.csv")
```

read_csv("co2-un-data.csv", skip = 1)

co2_emissions_dirty <- read_csv("co2-un-data.csv", skip = 2, col_names = c("region", "country", "year", "series", "value", "footnotes", "source"))

read_csv("co2-un-data.csv", skip = 1) %>%

- rename(country = ...2)

co2_emissions_dirty %>%

- select(country, year, series, value)

- rename strings within data columns with mutate() and recode():

co2_emissions_dirty %>%

- select(country, year, series, value) %>%
- mutate(series = recode(series, "Emissions (thousand metric tons of carbon dioxide)" = "total_emissions",
    - "Emissions per capita (metric tons of carbon dioxide)" = "per_capita_emissions"))

- separate series into new columns:

co2_emissions_dirty %>%

- select(country, year, series, value) %>%
- mutate(series = recode(series, "Emissions (thousand metric tons of carbon dioxide)" = "total_emissions",
    - "Emissions per capita (metric tons of carbon dioxide)" = "per_capita_emissions")) %>%
- pivot_wider(names_from = series, values_from = value)

This link has some nice visuals for how pivot_wider and pivot_longer work: https://epirhandbook.com/en/pivoting-data.html

- extract values close to year == 2007

co2_emissions_dirty %>%

- select(country, year, series, value) %>%
- mutate(series = recode(series, "Emissions (thousand metric tons of carbon dioxide)" = "total_emissions",
    - "Emissions per capita (metric tons of carbon dioxide)" = "per_capita_emissions")) %>%
- pivot_wider(names_from = series, values_from = value) %>%
- count(year)

Challenge: create data frame with the columns country, year, series, and value, rename the values in

series, separate the series data into new columns and extract data from the year 2005

One solution:

```
co2_emissions_dirty %>%
  select(country, year, series, values) %>%
  mutate(series = recode(series,
                         "Emissions (thousand metric tons of carbon dioxide)" =
"total_emissions",
                         "Emissions per capita(metric tons of carbon dioxide)" =
"per_capita_emissions")) %>%
    pivot_wider(names_from = series, values_from = value) %>%
  filter(year == 2005) %>%
  select(-year)

co2_emissions <-  co2_emissions_dirty %>%
  select(country, year, series, values) %>%
  mutate(series = recode(series,
        "Emissions (thousand metric tons of carbon dioxide)" = "total_emissions",
        "Emissions per capita(metric tons of carbon dioxide)" =
"per_capita_emissions")) %>%
```

- pivot_wider(names_from = series, values_from = values) %>%
- filter(year == 2005) %>%
- select(-year)

- merge CO2 and gapminder data:

how would we merge dataframe? look at columns that each dataset has in common, here country

common issues whem merging:

- incomplete data
- differences in spelling (either column names or values)

```
inner_join(gapminder_data, co2_emissions)

inner_join(gapmider_data, co2_emissions, by = "country")
```

you can merge by multiple columns using the by argument, for example: by = c("country", "year")

```
gapminder_co2 <- inner_join(gapminder_data_2007, co2_emissions, by = country)

ggplot(gapminder_co2, aes(x = gdpPercap, y = per_capita_emissions)) +
```

- geom_point() +
- labs(x = "GDP (per capita)",
  - y = "CO2 emitted (per capita)",
  - title = "Strong association between a nation's GDP and CO2 production")

# Writing Reports with R Markdown

https://r4ds.hadley.nz/ - R for Data Science
https://epirhandbook.com/en/reports-with-r-markdown.html#yaml-metadata - Epi Handbook
https://bookdown.org/ how to you rmarkdown to write a book in html, pdf, kindle format, etc.
https://bookdown.org/yihui/rmarkdown/xaringan.html Make presentation slides (with interactive plots and figures!)
http://nickstrayer.me/RMarkdown_Sites_tutorial/ Make your own personal website using Rmarkdown and host it using Github

jt helper: knitr::opts_knit$set(root.dir = rprojroot::find_rstudio_root_file())

in the R terminal:

- make a new directory named 'reports' in un-reports

- mkdir reports

- make a new rmarkdown file

- new file
- rmarkdown
- title: this is the title of the document (not the file name)
- author: who is creating this file (should auto fill your name)
- date: date of file creation
- select hmtl as output for today
- * when a new rmarkdown file is created, a template will be created
- save the new rmarkdown file in the reports directory

markdown structure:

- header: this is information about how the computer will render the document, all header information is contained between three dashes:
    - ---
    - ---

- code chunks are pieces of code that is denoted by the ```,  for example:
    - ```{r setup, include = FALSE}
        - data <- read_csv("path_to_file")
    - ```
    - - the r inside {} tells the computer what language the code is
    - - 'setup' is the name of the code chunk, this is optional
    - - settings  for how that code is run are set after the ,
    - code chunk options (some):
    - include: whether or not that code and output is shown in the report
        - TRUE - code and output are included in report
        - FALSE - code and output are not included in report
    - echo: whether the code is shown in the report or not
        - FALSE - output is included in report, but code is not

rendering reports:

- in R, rendering is completed by the package 'knitr'
- rendering is often referred to as 'knitting' in r due to the package nomeclature

Shortcut for adding new R code chunks:

- 1. click the green c with a + in the script header
- 2. ctrl + alt + i  / option + cmd + i

```{r packages}
- library(tidyverse)
```

You can run all code in a code chunk by clicking the green arrow on the right hand side of the code chunk

Report text is written outside of code chunks.

read data into report:

```{r data}
- gapminder_1997 <- read_csv("data/gapminder_1997.csv")
```

challenge 1:

- - create a new code chunk,
- - use ggplot to show gpd per capita ~ life expectancy in point format, color by continent, update legend to show pop size in millions
- one solution:
- ```{r gdp_lifeexp_1997}
  - ggplot(data = gapminder_1997) +
    - aes(x = gdpPercap, y = lifeExp, color = continent, size = pop/1000000) +
    - geom_point() +
    - labs(x = "GDP per capita", y = "Life Expectancy", size = "population (in millions)")
- ```

more helpful hints about rmarkdown: https://rstudio.github.io/cheatsheets/html/rmarkdown.html?_gl=1*1pbh072*_ga*MTg0NDYyNzM3Ny4xNzAyNTY4NTY0*_ga_2C0WZ1JHG0*MTcwMjU4MDIxNC4yLjAuMTcwMjU4MDIxNC4wLjAuMA..

```{r}
- nCountries <- gapminder_1997 %>%
  - select(country) %>%
  - n_distinct()
- minGDP <- gapminder_1997 %>%
  - summarize(round(min(gdpPercap))) %>%
  - pull()
```

challenge:

- create a new object named maxGDP containing the value of the maximum GDP per capita

```{r}

- nCountries <- gapminder_1997 %>%
    - select(country) %>%
    - n_distinct()
- minGDP <- gapminder_1997 %>%
    - summarize(round(min(gdpPercap))) %>%
    - pull()
- maxGDP <- gapminder_1997 %>%
    - summarize(round(max(gdpPercap))) %>%
    - pull()

```

Formatting rmarkdown documents:

- building tables:
- option 1 - outside code chunk (manual)
    - |Header 1|Header 2|
    - |-----------------|-----------------|
    - |row 1, column 1| row 1, column 2|
    - |row 2, column 1|row 2, column 2|
    - filled in with your information:
    - |Summary of data|
    - |-----------------|-----------------|
    - |Number of Countries|`r nCountries`|
    - |Minimum GDP per capita|`r minGDP`|
    - |Maximum GDP per capita|`r maxGDP`|
    - * you can include values stored in r objects within your text. To do this, values are contained in single `, as shown above. The r is required to tell the rendering package what language is being used followed by the object name where the value is saved.
- option 2 - inside code chunk (there are many ways to do this, this is just one)

https://umcarpentries.org/intro-curriculum-r/05-r-markdown/index.html#integrating-it-all-together-paired-exercise

Modern statistics for modern biology https://web.stanford.edu/class/bios221/book/01-chap.html
R for data science by Hadley wickham: https://r4ds.hadley.nz/
Interactive tutorials: https://posit.cloud/learn/primers

Post Workshop Survey: https://carpentries.typeform.com/to/UgVdRQ?slug=2023-12-13-UMich