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

Workshop Links:

Post workshop Survey:
https://ucsb.co1.qualtrics.com/jfe/form/SV_9AJwIjSYRmQFutM

RStudio Image: https://carpentryworkshop.lsit.ucsb.edu/
- you may need to do an extra install for palmerpenguins with install.packages("palmerpenguins")

Workshop Website: https://ucsbcarpentry.github.io/2022-07-26-ucsb-R/
Workshop Zoom: https://ucsb.zoom.us/j/83786617934?pwd=c0VlRFB1WEZ6UXlFeVpWQ1hhM0g3dz09

Workshop Lessons; https://ucsbcarpentry.github.io/CustomDC-R/
daily survey: https://forms.gle/cKHHpR9dW1WFo2Yb6

Instructor Scripts
Manipulating, Analyzing, and Exporting Data Part 1: https://github.com/UCSBCarpentry/2022-07-26-ucsb-R/blob/gh-pages/code/dplyr_pt_1.R
Manipulating, Analyzing, and Exporting Data Part 2: https://github.com/UCSBCarpentry/2022-07-26-ucsb-R/blob/gh-pages/code/dplyr_pt_2.R
Time Series Visualization: https://github.com/UCSBCarpentry/2022-07-26-ucsb-R/blob/gh-pages/code/vis_time_series.R

Introductions

Name, ucsb/other institution affiliation, favorite icecream

Christopher Kibler, UCSB Geography, dark chocolate
Dave Hunter, Westmont College Math/CS, Cookie Dough
Emily Fox, UCSB Sociology, mint chip
Laurie Van De Werfhorst, UCSB Bren School, vanilla w/cherries
Angie Torrico, UCSB undergrad: environmental studies, strawberry

Amanda Maheras | UCSB Molecular, Cellular and Developmental Biology PhD Student | Black Cherry
Edwin Gao, UCSB PSTAT & Econ, strawberry
Mackenzie Taradalsky, UCSB Econ & Philosophy, mint chip
Pippa Lin, UCSB Statistics and Probability Undergrad, white chocolate
Jennifer Rink | UCSB PSTAT Undergrad | Chocolate Macadamia
Lucero Torres Ojeda | UCSB Environmental Studies Undergrad | Rocky Road
Gabriel Franco, Geology, raspberry
Lizette Rivera, UCSB Economics
Denis Lomov, UCSB, 1 year PhD in Political Science, pineapple
Adam Parison/UCSB Classics/Peanut butter
Yoobin Won, UCSB PSTAT, matcha
Justin Gutierrez, UCSB Biology, peanut butter
Abhishek Sharma, UCSB Mechanical Engineering, Butterscotch
Matthew Rosen, UCSB Pre-Bio Undergrad, Strawberry
Kaitlyn Deen, UCSB Psych & Brain Sciences, Chocolate with Oreo
Joyce Chen, UCSB math&stats undergrad
Natalia Almanza UCSB economics , cookies n cream
Madison Avila, Sociology, chocolate
Fatima Gonzalez, UCSB Psych & Brain Sciences, chocolate
Martha Garcia, UCSB Psych and Brain Sciences undergrad, chocolate
Dong Li UCSB Bren School

Setup packages - you'll need these for the workshop

```
library(tidyverse)
library(hexbin)
library(patchwork)
library(RSQLite)
```

or compile install.packages(c("tidyverse", "hexbin", "patchwork", "RSQLite"))
- but this may take a while

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

For Pipes
ctrl-shift-m on linux/windows
cmd-shit-m on mac

Dataframe = spreadsheet form workable in Rstudio/R. We will be working with these during th workshop

head(data) will give us the first few rows of a dataset
view(data) will pop up a new tab with the complete dataset

NA values are indication that some data may be missing
is.na to set missing-ness (useful if we want to find NAs or omit them)

summary(data) tabulate all the individual records in the dataset and spit out some summary statistics

[ ] vs ( )
[ ] = used for vectors and dataframes, with dataframes to specify [rows, columns]
( ) = for calling a function in R

DAY 1 Whiteboard Content:

Data Wrangling
"Reproducibility"

select() = selects columns, can unselect columns with a minus (-)
filter() = filters rows

Pipe
%>% = percent sign, greater than, percent sign; you may read it as "and then" or "pipe to"

- example: select() %>% filter() = select and then filter

mutate() = creates new columns

Order for Challenge:
1. new column (mutate)
2. filter out
3. select columns

Challenge

Create a new data frame from the penguins data that meets the following criteria: contains only the species column and a new column called flipper_length_cm containing the length of the penguin flipper values (currently in mm) converted to centimeters. In this flipper_length_mm column, there are no NAs and all values are less than 200. Hint: think about how the commands should be ordered to produce this data frame!

Here's another R ecology lesson that covers SQL: https://datacarpentry.org/R-ecology-lesson/

Dave's script from Tuesday is here: https://math.westmont.edu/dc/episode04a.R

Day 2 Notes

Review of Day 1

- select, filter, mutate:
- dealing with NAs

Zoom chat notes:

- The ! reverses TRUE and FALSE values, so is.na(value) tells you if a value is NA and ! is.na(value) tells you if a value is not NA.
- https://www.rstudio.com/resources/cheatsheets/
- The dplyr and ggplot cheat sheets cover the topics that we'll be learning today
- Question: What did summarize do in this analysis?
- Answer: Summarize made a new data frame instead of adding columns to the old data frame. The new data frame has three rows (one for each observed value of sex).
- Question: Where did the other 2 penguins go, given that there were 11 penguins with NA values?
- Answer: They did not have any measurements at all, so we removed them at the beginning. The other nine penguins had some measurements.
- If you prefer a dark theme, you can change it in Tools > Global Options > Appearance > Editor Theme
- Question: What does geom_point() do?
- Answer: It creates a scatter plot with the input parameters in the ggplot() function.
- Any aes() parameters in the first line of code will automatically be applied to all of the individual layers
- In this case, adding the color to the geom_point() makes it so only the points change color
- In theory, you can skip the aes() call in the first line, but then you have to add it to each layer individually
- It's almost always advisable to set your x and y parameters in the first line
- There are literally hundreds of parameters in ggplot. The best way to approach ggplot is to learn the structure and theory behind it, and then google the names of the specific parameters you need.

Challenge:

```
# 1. How many penguins are in each island surveyed?
#
penguins %>%
  group_by(island) %>%
  summarise(num_penguins = n()) %>%
  view()
# 2. Use group_by() and summarize() to find the mean,
#    min, and max bill length for each species (using species).
#    Also add the number of observations (hint: see ?n).
#
# 3. What was the heaviest animal measured in each year?
#    Return the columns year, island, species, and body_mass_g.
```

Dave's script is available here:

https://math.westmont.edu/dc/episode04b.R

Visualization using ggplot

penguins_plot <- ggplot(data = penguins_comp, mapping = aes(x = body_mass_g, y = flipper_length_mm))

Challenge:

Scatter plots can be useful exploratory tools for small datasets. For data sets with large numbers of observations, such as the surveys_complete data set, overplotting of points can be a limitation of scatter plots. One strategy for handling such settings is to use hexagonal binning of observations. The plot space is tessellated into hexagons. Each hexagon is assigned a color based on the number of observations that fall within its boundaries. To use hexagonal binning with ggplot2, first install the R package hexbin from CRAN:

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

Then use the geom_hex() function:

```
penguins_plot +
  geom_hex()
```

What are the relative strengths and weaknesses of a hexagonal bin plot compared to a scatter plot? Examine the above scatter plot and compare it with the hexagonal bin plot that you created.

ggplot cheat sheet: https://github.com/rstudio/cheatsheets/blob/main/data-visualization-2.1.pdf

Challenge:

Boxplots are useful summaries, but hide the shape of the distribution. For example, if there is a bimodal distribution, it would not be observed with a boxplot. An alternative to the boxplot is the violin plot (sometimes known as a beanplot), where the shape (of the density of points) is drawn.
Replace the box plot with a violin plot; see geom_violin(). Try making a new plot to explore the distribution of another variable within each species.
Some suggested explorations:

- Create boxplot for flipper_length_mm. Overlay the boxplot layer on a jitter layer to show actual measurements.
- Add color to the data points on your boxplot according to the plot from which the observation was located (island). Hint: Check the class for island.

Whiteboard:
Good plots address questions such as ...
DISTRIBUTION
RELATIONSHIP
COMPOSITION

Great book: *The Visual Display of Quantitative information*, by Tufte

This works:

```
penguins_raw_subset <- penguins_raw %>%
    select("Island", "Species", "Date Egg") %>%
    filter(year(penguins_raw[["Date Egg"]]) == 2008) %>%
    rename(date_egg = "Date Egg")

daily_counts <- penguins_raw_subset %>%
  count(date_egg, Species)

ggplot(data = daily_counts, aes(x = date_egg, y = n, color = Species)) +
  geom_line()
```

Or better yet:

```
penguins_raw_subset <- penguins_raw %>%
    select("Island", "Species", "Date Egg") %>%
    rename(date_egg = "Date Egg") %>%
    filter(year(date_egg) == 2008)

daily_counts <- penguins_raw_subset %>%
  count(date_egg, Species)

ggplot(data = daily_counts, aes(x = date_egg, y = n, color = Species)) +
  geom_line()
```