WEEK 6

Introduction to Coding for Web Design and Data Viz

DATA VISUALIZATION

Digital storytelling at the confluence of science, art, and technology

AGENDA FOR TODAY

Uniting raster and vector graphics for our graphical abstract:

Figure Critique
Intro to Web Design

Scientific Figures

A critique and celebration

SCIENTIFIC FIGURES

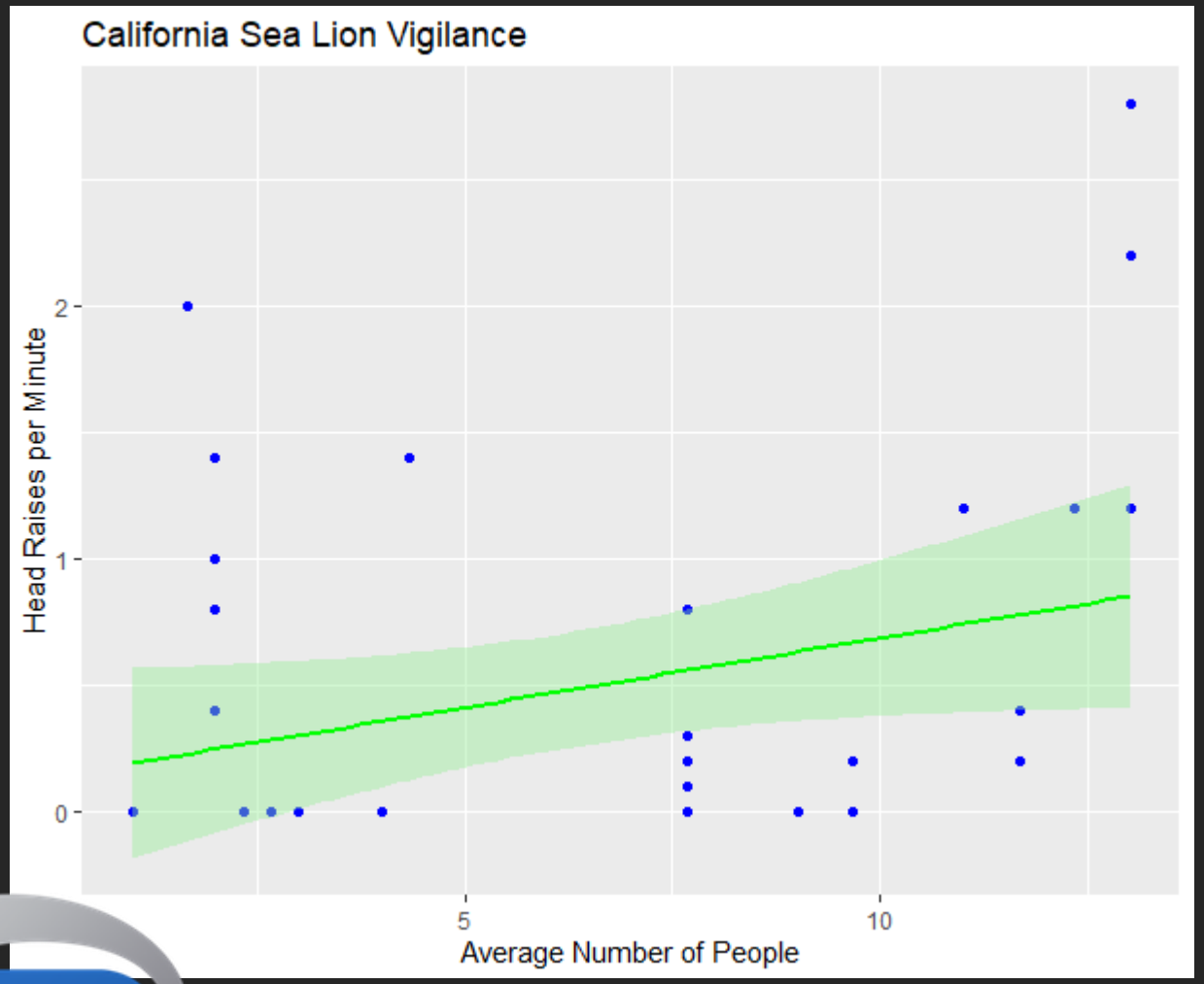
Critique/Celebration!

We will display a few animations and discuss each one.

Some discussion points:

1) How do your eyes travel across the figure?
2) How do the colors, style, text (especially), and graphics influence the message?
3) Describe the story the figure tells.
4) What questions are you still wondering after looking at the figure?

Please be mindful and conscientious while providing feedback by participating in the discussion, turning on video whenever possible, and making space for everyone to be heard. We want to create a supportive environment where we feel comfortable sharing our work.

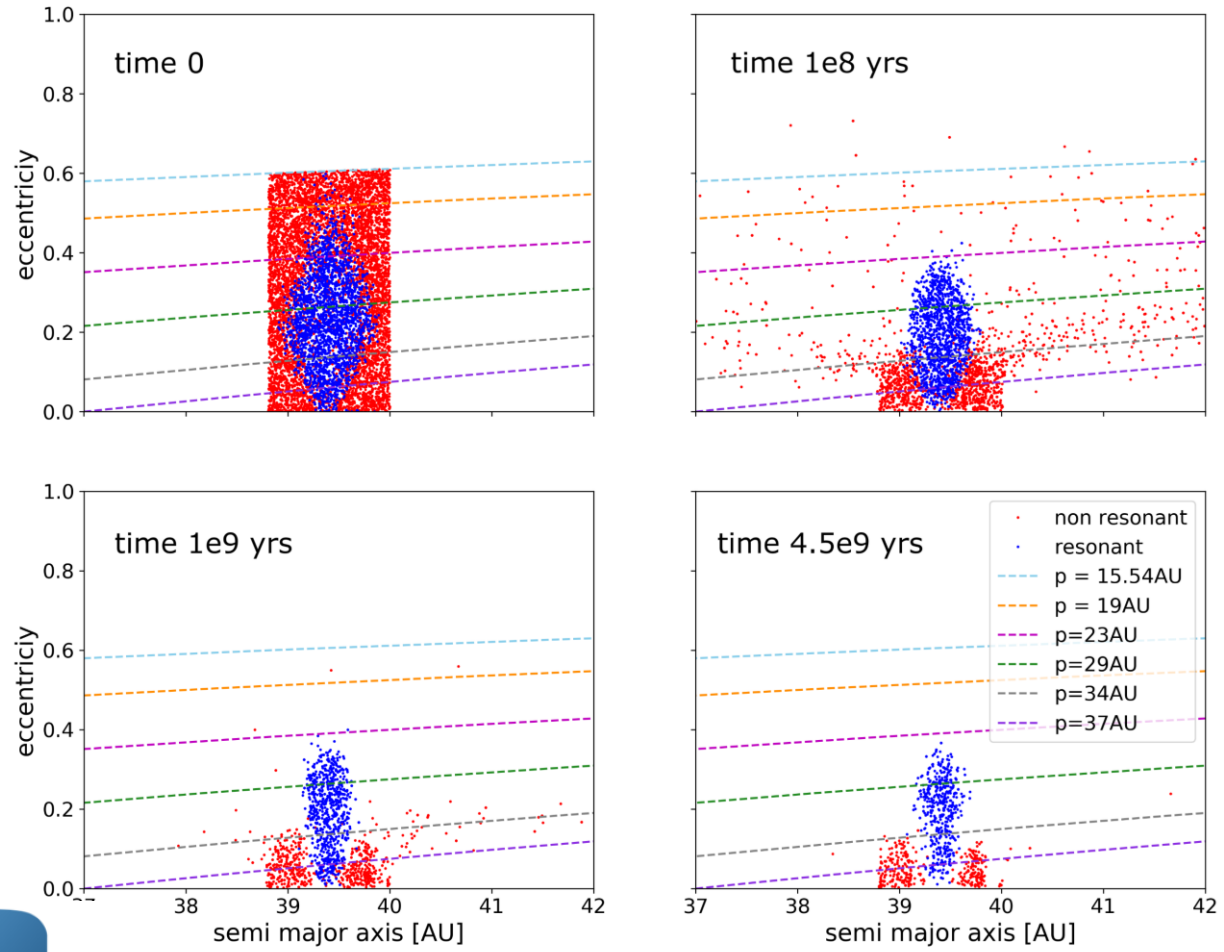



```r
#Linnea Gullikson
#Data Visualization
rm(list=ls())
setwd("C:/Users/Linnea/Desktop/DataViz")
getwd()vigilance <- read.csv("sealion.csv")
str(vigilance)
names(vigilance)
head(vigilance)
summary(vigilance)

library(datasets)
library(tidyverse)
ggplot(vigilance,
       aes(x=People, y=Head_Raises)) +
  geom_point(color="blue") +
  labs(x='Average Number of People',
       y='Head Raises per Minute',
       title = 'California Sea LionVigilance') +
  geom_smooth(method = 'lm',
              color = "green",
              fill = "lightgreen")
```

R LANGUAGE

Artist: Linnea Gullikson




```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import glob
get_ipython().run_line_magic('matplotlib', 'inline'
)
# read in the data. There are 150+ files for each t
ime, so bring them all together and concatenate

csv_time0_data = glob.glob("/Users/arceliahermosill
o/Research/Jan_final_int/Jan*/Short*/Short_Integrat
ion_time_0.0/Jan*.csv")
csv_time1e7_data = glob.glob("/Users/arceliahermosi
llo/Research/Jan_final_int/Jan*/Short*/Short_Integr
ation_time_10000000.0/Jan*.csv")
csv_time1e8_data = glob.glob("/Users/arceliahermosi
llo/Research/Jan_final_int/Jan*/Short*/Short_Integr
ation_time_100000000.0/Jan*.csv")
csv_time1e9_data = glob.glob("/Users/arceliahermosi
llo/Research/Jan_final_int/Jan*/Short*/Short_Integr
ation_time_1000000000.0/Jan*.csv")
csv_time4_5e9_data = glob.glob("/Users/arceliahermo
sillo/Research/Jan_final_int/Jan*/Short*/Short_Inte
gration_time_4500000000.0/Jan*.csv")
```

PYTHON

Artist: Arcelia Hermosillo Ruiz

```
# ## Some functions so I'm not copying pasting

# In[3]:

# concatenate and clean my data
def concatenate_clean(glob_files):
    dat_list = []
    #append all the data files together
    for glob1 in glob_files:
        dat_list.append(pd.read_csv(glob1))

    #now concatenate them to one data frame
    concat_dat = pd.concat(dat_list, ignore_index = True)
#     for some reason they have an unnamed column. remove that
    if "Unnamed: 0" in concat_dat.keys():
        concat_dat.drop("Unnamed: 0", "columns", inplace = True)
    if "Unnamed: 0.1" in concat_dat.keys():
        concat_dat.drop("Unnamed: 0.1", "columns", inplace = True)
    if "Unnamed: 0.1.1" in concat_dat.keys():
        concat_dat.drop("Unnamed: 0.1.1", "columns", inplace = True)
        # ok actually decided not to cut negative a just yet
        # I kind of want to see why that happens
#     # then drop all particles that have a < 0
#     idx_a_negative = concat_dat.index[concat_dat['a'] < 0]
#     concat_dat.drop(idx_a_negative, inplace = True)
    return concat_dat
```

Arcelia decided to write functions to perform some actions in her plot so that she didn't have to write the same instructions again.

This function strings things together and cleans the data.

Artist: Arcelia Hermosillo Ruiz

This function makes plots for each set of data.

```
def make_plots(data, png_name):
    res_idx = data.index[data['resonance'] ==1.0][::1001]
    nores_idx = data.index[data['resonance'] ==0.0][::1001]

    x19 = np.linspace(19,60,18)
    x24 = np.linspace(24,60,18)
    x29 = np.linspace(29,60,18)
    x34 = np.linspace(34,60,18)
    x37 = np.linspace(37,60,18)
    x15_54 = np.linspace(15.54,60,18)
    #print(x)
    eccentricity_from_peri_19 = []
    eccentricity_from_peri_24 = []
    eccentricity_from_peri_29 = []
    eccentricity_from_peri_34 = []
    eccentricity_from_peri_37 = []
    eccentricity_from_peri_15_54 = []
    for ep in range(len(x19)):
        ecc19 = 1-(19/x19[ep])
        ecc24 = 1-(24/x24[ep])
        ecc29 = 1-(29/x29[ep])
        ecc34 = 1-(34/x34[ep])
        ecc37 = 1-(37/x37[ep])
        ecc15_54 = 1-(15.54/x15_54[ep])
        eccentricity_from_peri_19.append(ecc19)
        eccentricity_from_peri_24.append(ecc24)
        eccentricity_from_peri_29.append(ecc29)
        eccentricity_from_peri_34.append(ecc34)
        eccentricity_from_peri_37.append(ecc37)
        eccentricity_from_peri_15_54.append(ecc15_54)
```

```
fig, ax = plt.subplots(figsize = (8,5))

    ax.plot(data.loc[nores_idx, 'a'], data.loc[nores_idx,'e'], 'r.', markersize =
 2, label = "non resonant")
    ax.plot(data.loc[res_idx, 'a'], data.loc[res_idx,'e'], 'b.', markersize = 2,
label = "resonant")
    ax.set_xlim(37,42)
    ax.set_ylim(0,1)
    ax.set_ylabel("eccentriciy", fontsize = 18)
    ax.set_xlabel("semi major axis [AU]", fontsize = 18)
    ax.plot(x15_54,eccentricity_from_peri_15_54, color = 'yellow', linestyle = 'd
ashed', label = "p = 15.54AU")
    ax.plot(x19,eccentricity_from_peri_19, color = 'darkorange', linestyle = 'das
hed', label = "p = 19AU")
    ax.plot(x24,eccentricity_from_peri_24, color = 'm', linestyle = 'dashed', lab
el = "p=23AU")
    ax.plot(x29,eccentricity_from_peri_29, color = 'forestgreen', linestyle = 'da
shed', label = "p=29AU")
    ax.plot(x34,eccentricity_from_peri_34, color = 'grey', linestyle = 'dashed',
label = "p=34AU")
    ax.plot(x37,eccentricity_from_peri_37, color = 'blueviolet', linestyle = 'das
hed', label = "p=37AU")

    # We change the fontsize of minor ticks label
    ax.tick_params(axis='both', which='major', labelsize=15)
    ax.tick_params(axis='both', which='minor', labelsize=8)
    plt.legend(loc = "upper right", fontsize = 15)
#     plt.savefig('JanPlots/{}.png'.format(png_name), dpi=300)
    plt.show()
```

Artist: Arcelia Hermosillo Ruiz

This function makes subplots.

```
def make_subplots(data1, data2, data3, data4, png_name):

    # getting the indices of the data for 2 separate conditi
ons (in resonance and not resonance)
    res_idx1 = data1.index[data1['resonance'] ==1.0][::1001]
    nores_idx1 = data1.index[data1['resonance'] ==0.0][::100
1]

    res_idx2 = data2.index[data2['resonance'] ==1.0][::1001]
    nores_idx2 = data2.index[data2['resonance'] ==0.0][::100
1]

    res_idx3 = data3.index[data3['resonance'] ==1.0][::1001]
    nores_idx3 = data3.index[data3['resonance'] ==0.0][::100
1]

    res_idx4 = data4.index[data4['resonance'] ==1.0][::1001]
    nores_idx4 = data4.index[data4['resonance'] ==0.0][::100
1]
```

```
    # making some arrays for lines that will be in the plot
    # these lines represent equal pericenter (closest approach of an orbital obje
ct to the sun)
    x19 = np.linspace(19,60,18)
    x24 = np.linspace(24,60,18)
    x29 = np.linspace(29,60,18)
    x34 = np.linspace(34,60,18)
    x37 = np.linspace(37,60,18)
    x15_54 = np.linspace(15.54,60,18)
    #print(x)
    eccentricity_from_peri_19 = []
    eccentricity_from_peri_24 = []
    eccentricity_from_peri_29 = []
    eccentricity_from_peri_34 = []
    eccentricity_from_peri_37 = []
    eccentricity_from_peri_15_54 = []
    for ep in range(len(x19)):
        ecc19 = 1-(19/x19[ep])
        ecc24 = 1-(24/x24[ep])
        ecc29 = 1-(29/x29[ep])
        ecc34 = 1-(34/x34[ep])
        ecc37 = 1-(37/x37[ep])
        ecc15_54 = 1-(15.54/x15_54[ep])
        eccentricity_from_peri_19.append(ecc19)
        eccentricity_from_peri_24.append(ecc24)
        eccentricity_from_peri_29.append(ecc29)
        eccentricity_from_peri_34.append(ecc34)
        eccentricity_from_peri_37.append(ecc37)
        eccentricity_from_peri_15_54.append(ecc15_54)
```

Artist: Arcelia Hermosillo Ruiz

```
    # initialize the subplot
    fig, axs = plt.subplots(2,2, figsize = (15,12), sharex = True, sharey = True)

    # first squar in subplot
    # plotting below, then setting limits on plots, setting labels and fontsize
    axs[0,0].plot(data1.loc[nores_idx1, 'a'], data1.loc[nores_idx1,'e'], 'r.', markersize = 2, label = "non resonant")
    axs[0,0].plot(data1.loc[res_idx1, 'a'], data1.loc[res_idx1,'e'], 'b.', markersize = 2, label = "resonant")
    axs[0,0].set_xlim(37,42)
    axs[0,0].set_ylim(0,1)
    axs[0,0].set_ylabel("eccentriciy", fontsize = 18)
#     axs[0,0].set_xlabel("semi major axis [AU]", fontsize = 18)
    axs[0,0].plot(x15_54,eccentricity_from_peri_15_54, color = 'skyblue', linestyle = 'dashed', label = "p = 15.54AU")
    axs[0,0].plot(x19,eccentricity_from_peri_19, color = 'darkorange', linestyle = 'dashed', label = "p = 19AU")
    axs[0,0].plot(x24,eccentricity_from_peri_24, color = 'm', linestyle = 'dashed', label = "p=23AU")
    axs[0,0].plot(x29,eccentricity_from_peri_29, color = 'forestgreen', linestyle = 'dashed', label = "p=29AU")
    axs[0,0].plot(x34,eccentricity_from_peri_34, color = 'grey', linestyle = 'dashed', label = "p=34AU")
    axs[0,0].plot(x37,eccentricity_from_peri_37, color = 'blueviolet', linestyle = 'dashed', label = "p=37AU")

    # We change the fontsize of minor ticks label
    axs[0,0].tick_params(axis='both', which='major', labelsize=15)
    axs[0,0].tick_params(axis='both', which='minor', labelsize=8)
#     plt.legend(loc = "upper right", fontsize = 15)
#     plt.savefig('JanPlots/{}.png'.format(png_name), dpi=300)

    axs[0,1].plot(data2.loc[nores_idx2, 'a'], data2.loc[nores_idx2,'e'], 'r.', markersize = 2, label = "non resonant")
    axs[0,1].plot(data2.loc[res_idx2, 'a'], data2.loc[res_idx2,'e'], 'b.', markersize = 2, label = "resonant")
    axs[0,1].set_xlim(37,42)
    axs[0,1].set_ylim(0,1)
#     axs[0,1].set_ylabel("eccentriciy", fontsize = 18)
#     axs[0,1].set_xlabel("semi major axis [AU]", fontsize = 18)
    axs[0,1].plot(x15_54,eccentricity_from_peri_15_54, color = 'skyblue', linestyle = 'dashed', label = "p = 15.54AU")
    axs[0,1].plot(x19,eccentricity_from_peri_19, color = 'darkorange', linestyle = 'dashed', label = "p = 19AU")
    axs[0,1].plot(x24,eccentricity_from_peri_24, color = 'm', linestyle = 'dashed', label = "p=23AU")
    axs[0,1].plot(x29,eccentricity_from_peri_29, color = 'forestgreen', linestyle = 'dashed', label = "p=29AU")
    axs[0,1].plot(x34,eccentricity_from_peri_34, color = 'grey', linestyle = 'dashed', label = "p=34AU")
    axs[0,1].plot(x37,eccentricity_from_peri_37, color = 'blueviolet', linestyle = 'dashed', label = "p=37AU")

    # We change the fontsize of minor ticks label
    axs[0,1].tick_params(axis='both', which='major', labelsize=15)
    axs[0,1].tick_params(axis='both', which='minor', labelsize=8)

    axs[1,0].plot(data3.loc[nores_idx3, 'a'], data3.loc[nores_idx3,'e'], 'r.', markersize = 2, label = "non resonant")
    axs[1,0].plot(data3.loc[res_idx3, 'a'], data3.loc[res_idx3,'e'], 'b.', markersize = 2, label = "resonant")
    axs[1,0].set_xlim(37,42)
    axs[1,0].set_ylim(0,1)
    axs[1,0].set_ylabel("eccentriciy", fontsize = 18)
    axs[1,0].set_xlabel("semi major axis [AU]", fontsize = 18)
    axs[1,0].plot(x15_54,eccentricity_from_peri_15_54, color = 'skyblue', linestyle = 'dashed', label = "p = 15.54AU")
    axs[1,0].plot(x19,eccentricity_from_peri_19, color = 'darkorange', linestyle = 'dashed', label = "p = 19AU")
    axs[1,0].plot(x24,eccentricity_from_peri_24, color = 'm', linestyle = 'dashed', label = "p=23AU")
    axs[1,0].plot(x29,eccentricity_from_peri_29, color = 'forestgreen', linestyle = 'dashed', label = "p=29AU")
    axs[1,0].plot(x34,eccentricity_from_peri_34, color = 'grey', linestyle = 'dashed', label = "p=34AU")
    axs[1,0].plot(x37,eccentricity_from_peri_37, color = 'blueviolet', linestyle = 'dashed', label = "p=37AU")
```

Then she uses the functions to make each plot, changes colors and labels, and then puts the plots together using `make_subplots`.

```
    # We change the fontsize of minor ticks label
    axs[1,0].tick_params(axis='both', which='major', labelsize=15)
    axs[1,0].tick_params(axis='both', which='minor', labelsize=8)

        # We change the fontsize of minor ticks label
    axs[0,1].tick_params(axis='both', which='major', labelsize=15)
    axs[0,1].tick_params(axis='both', which='minor', labelsize=8)

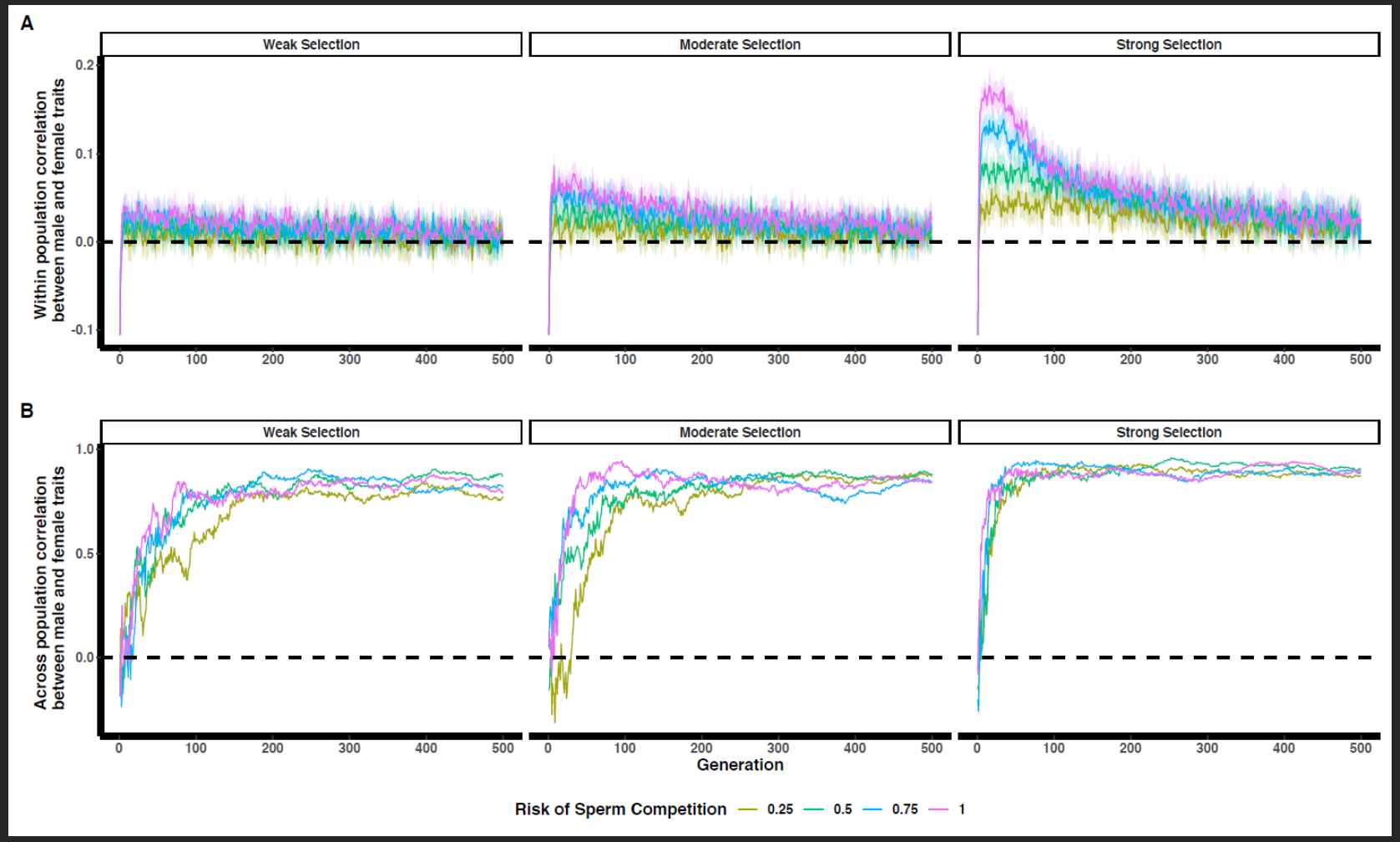
    axs[1,1].plot(data4.loc[nores_idx4, 'a'], data4.loc[nores_idx4,'e'], 'r.', markersize = 2, label = "non resonant")
    axs[1,1].plot(data4.loc[res_idx4, 'a'], data4.loc[res_idx4,'e'], 'b.', markersize = 2, label = "resonant")
    axs[1,1].set_xlim(37,42)
    axs[1,1].set_ylim(0,1)
#     axs[1,1].set_ylabel("eccentriciy", fontsize = 18)
    axs[1,1].set_xlabel("semi major axis [AU]", fontsize = 18)
    axs[1,1].plot(x15_54,eccentricity_from_peri_15_54, color = 'skyblue', linestyle = 'dashed', label = "p = 15.54AU")
    axs[1,1].plot(x19,eccentricity_from_peri_19, color = 'darkorange', linestyle = 'dashed', label = "p = 19AU")
    axs[1,1].plot(x24,eccentricity_from_peri_24, color = 'm', linestyle = 'dashed', label = "p=23AU")
    axs[1,1].plot(x29,eccentricity_from_peri_29, color = 'forestgreen', linestyle = 'dashed', label = "p=29AU")
    axs[1,1].plot(x34,eccentricity_from_peri_34, color = 'grey', linestyle = 'dashed', label = "p=34AU")
    axs[1,1].plot(x37,eccentricity_from_peri_37, color = 'blueviolet', linestyle = 'dashed', label = "p=37AU")

    # We change the fontsize of minor ticks label
    axs[1,1].tick_params(axis='both', which='major', labelsize=15)
    axs[1,1].tick_params(axis='both', which='minor', labelsize=8)
    axs[1,1].legend(loc = "upper right", fontsize = 15)
    plt.savefig('JanPlots/{}.png'.format(png_name), dpi=300)
    fig.tight_layout()
    plt.show()

# In[5]:
time0_df = concatenate_clean(csv_time0_data)
# In[6]:
time_1e7_df = concatenate_clean(csv_time1e7_data)
time_1e8_df = concatenate_clean(csv_time1e8_data)
time_1e9_df = concatenate_clean(csv_time1e9_data)
time_45e9_df = concatenate_clean(csv_time4_5e9_data)

# In[57]:
make_subplots(time0_df, time_1e8_df, time_1e9_df, time_45e9_df, "4time_sub")
```

Artist: Arcelia Hermosillo Ruiz




```
## Loading Libraries and setting working directory


```{r libraries}
setwd("~/Desktop/my_papers/Projects/Phenotypic_Paper/Code")
library(patchwork)
library(tidyverse)
library(feather)
```


Processing Data

Loading up the data and filtering out simulation runs I don't need.

```{r data}
run_data<-data.frame(read_feather("Data_each_generation_wide.feather"))
run_data_clean<-run_data[run_data$PopN==500 & run_data$Model=="Tradeoff",]%>%
  select(-c(Model,PopN))%>%
  mutate(ID=paste(A,Level,sep="_"))

run_data_across_pop_cor<- run_data_clean%>%
  filter(!Generation==0) %>%
  group_by(Generation,Level,A,ID)%>%
  summarise(intercor=cor(Pheno_Average_F,Pheno_Average_M))
```


Setting up plotting themes

Setting up my prefered plotting theme.

```{r plotthemes}
mytheme<-theme_classic()+theme(legend.position ="bottom",#this puts legend on the bottom
                               axis.title = (element_text(face="bold")),#this makes the axis title
                               axis.line=element_line(color="black",size=2),#Makes the axis line b
                               text=element_text(size=12,face="bold"))#makes all the text larger a
theme_set(mytheme)

A.labs<-c("Weak Selection","Moderate Selection","Strong Selection")
names(A.labs)<-c("10","5","1")
```


Plot 1

Making subplot 1 of within population correlations.

```{r plot1}
Plot1<-ggplot(run_data_clean,aes(x=Generation,y=Cor,color=Level,fill=Level))+
  stat_summary(fun.data = mean_cl_boot,geom="ribbon",alpha=.25,color = NA,fun.args = list(B=1000))
  stat_summary(fun= mean,geom="line",size=0.5)+
  facet_grid(.~A,labeller=labeller(A=A.labs))+
  scale_color_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E7
  scale_fill_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76
  geom_hline(yintercept = 0,size=1,color="black",linetype="dashed")+
  ylab("Within population correlation \n between male and female traits")+theme(legend.position =
Plot1
```


##Making subplot 2

Subplot 2 of across population correlations

```{r plot2}
Plot2<-ggplot(run_data_across_pop_cor,aes(x=Generation,y=intercor,color=Level,fill=Level))+
  geom_line()+
  facet_grid(.~A,labeller=labeller(A=A.labs))+
  scale_color_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E7
  scale_fill_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76
  geom_hline(yintercept = 0,size=1,color="black",linetype="dashed")+
  ylab("Across population correlation \n between male and female traits")
Plot2
```


##Putting together final plots
```

R LANGUAGE

Artist: Matt Kustra

Matt reads in the data, sets up his plot themes, and then plots subplots in ggplot with custom hexcode colors.

````
---
title: "Figure_for_Publication_Kustra"
author: "Matt Kustra"
date: "2/15/2021"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## Loading Libraries and setting working directory


```{r libraries}
setwd("~/Desktop/my_papers/Projects/Phenotypic_Paper/Code")
library(patchwork)
library(tidyverse)
library(feather)
```

## Processing Data

Loading up the data and filtering out simulation runs I don't need.

```{r data}
run_data<-data.frame(read_feather("Data_each_generation_wide.feather"))
run_data_clean<-run_data[run_data$PopN==500 & run_data$Model=="Tradeoff",]%>%
  select(-c(Model,PopN))%>%
  mutate(ID=paste(A,Level,sep="_"))

run_data_across_pop_cor<- run_data_clean%>%
  filter(!Generation==0) %>%
  group_by(Generation,Level,A,ID)%>%
  summarise(intercor=cor(Pheno_Average_F,Pheno_Average_M))
```

## Setting up plotting themes

Setting up my prefered plotting theme.

```{r plotthemes}
mytheme<-theme_classic()+theme(legend.position ="bottom",#this puts legend on the bottom
                               axis.title = (element_text(face="bold")),#this makes the axis titles in bold,
                               axis.line=element_line(color="black",size=2),#Makes the axis line blk & thicker
                               text=element_text(size=12,face="bold"))#makes all the text larger and bold
theme_set(mytheme)

A.labs<-c("Weak Selection","Moderate Selection","Strong Selection")
names(A.labs)<-c("10","5","1")
```

   ## Plot 1

Making subplot 1 of within population correlations.

```{r plot1}
Plot1<-ggplot(run_data_clean,aes(x=Generation,y=Cor,color=Level,fill=Level))+
  stat_summary(fun.data = mean_cl_boot,geom="ribbon",alpha=.25,color = NA,fun.args = list(B=1000))+
  stat_summary(fun= mean,geom="line",size=0.5)+
  facet_grid(.~A,labeller=labeller(A=A.labs))+
  scale_color_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76BF3"))+
  scale_fill_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76BF3"))+
  geom_hline(yintercept = 0,size=1,color="black",linetype="dashed")+
  ylab("Within population correlation \n between male and female traits")+theme(legend.position = "none")
Plot1
```

##Making subplot 2

Subplot 2 of across population correlations

```{r plot2}
Plot2<-ggplot(run_data_across_pop_cor,aes(x=Generation,y=intercor,color=Level,fill=Level))+
  geom_line()+
  facet_grid(.~A,labeller=labeller(A=A.labs))+
  scale_color_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76BF3"))+
  scale_fill_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76BF3"))+
  geom_hline(yintercept = 0,size=1,color="black",linetype="dashed")+
  ylab("Across population correlation \n between male and female traits")
Plot2
```

##Putting together final plots

```{r finalplot,fig.width=12,fig.height=8}
(Plot1+xlab(""))/Plot2+plot_annotation(tag_levels = "A")
```
````

Artist: Matt Kustra

Figure_for_Publication_Kustra

Matt Kustra

2/15/2021

Loading Libraries and setting working directory

```
setwd("~/Desktop/my_papers/Projects/Phenotypic_Paper/Code")
library(patchwork)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.3     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(feather)
```

Processing Data

Loading up the data and filtering out simulation runs I don't need.

```
run_data<-data.frame(read_feather("Data_each_generation_wide.feather"))
run_data_clean<-run_data[run_data$PopN==500 & run_data$Model=="Tradeoff",]%>%
  select(-c(Model,PopN))%>%
  mutate(ID=paste(A,Level,sep="_"))

run_data_across_pop_cor<- run_data_clean%>%
  filter(!Generation==0) %>%
  group_by(Generation,Level,A,ID)%>%
  summarise(intercor=cor(Pheno_Average_F,Pheno_Average_M))
```

```
## `summarise()` regrouping output by 'Generation', 'Level', 'A' (override with `.groups` argument)
```

Setting up plotting themes

Setting up my prefered plotting theme.

```
mytheme<-theme_classic()+theme(legend.position ="bottom",#this puts legend on the bottom
                               axis.title = (element_text(face="bold")),#this makes the axis titles in bold,
                               axis.line=element_line(color="black",size=2),#Makes the axis line black and  thicker
                               text=element_text(size=12,face="bold"))#makes all the text larger and bold
theme_set(mytheme)

A.labs<-c("Weak Selection","Moderate Selection","Strong Selection")
names(A.labs)<-c("10","5","1")
```

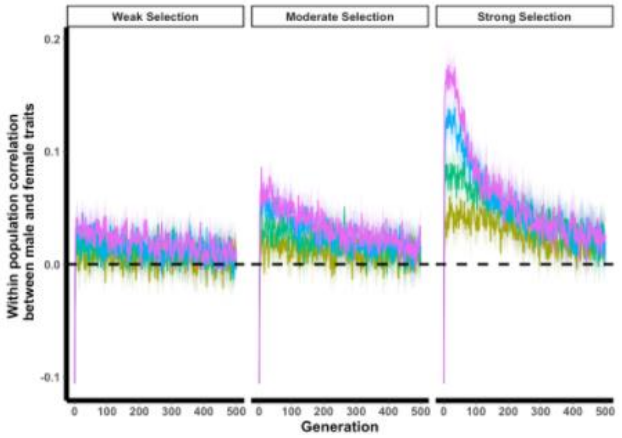
Plot 1

Making subplot 1 of within population correlations.

```
Plot1<-ggplot(run_data_clean,aes(x=Generation,y=Cor,color=Level,fill=Level))+
  stat_summary(fun.data = mean_cl_boot,geom="ribbon",alpha=.25,color = NA,fun.args = list(B=1000))+
  stat_summary(fun= mean,geom="line",size=0.5)+
  facet_grid(.~A,labeller=labeller(A=A.labs))+
  scale_color_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76BF3"))+
  scale_fill_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76BF3"))+
  geom_hline(yintercept = 0,size=1,color="black",linetype="dashed")+
  ylab("Within population correlation \n between male and female traits")+theme(legend.position = "none")
Plot1
```

```
## Warning: Removed 6571 rows containing non-finite values (stat_summary).

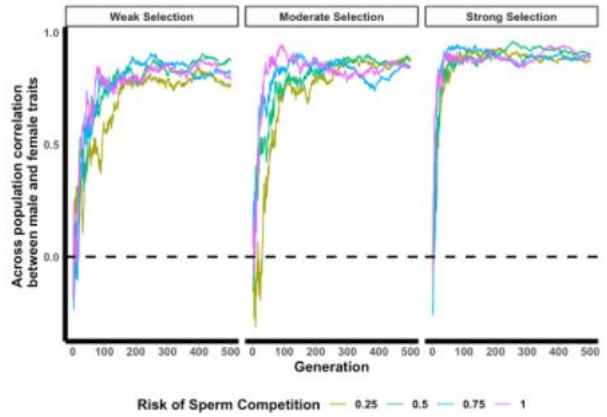
## Warning: Removed 6571 rows containing non-finite values (stat_summary).
```



Making subplot 2

Subplot 2 of across population correlations

```
Plot2<-ggplot(run_data_across_pop_cor,aes(x=Generation,y=intercor,color=Level,fill=Level))+
  geom_line()+
  facet_grid(.~A,labeller=labeller(A=A.labs))+
  scale_color_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76BF3"))+
  scale_fill_manual(name="Risk of Sperm Competition",values=c("#A3A500","#00BF7D" ,"#00B0F6","#E76BF3"))+
  geom_hline(yintercept = 0,size=1,color="black",linetype="dashed")+
  ylab("Across population correlation \n between male and female traits")
Plot2
```



rmarkdown

www.rstudio.com

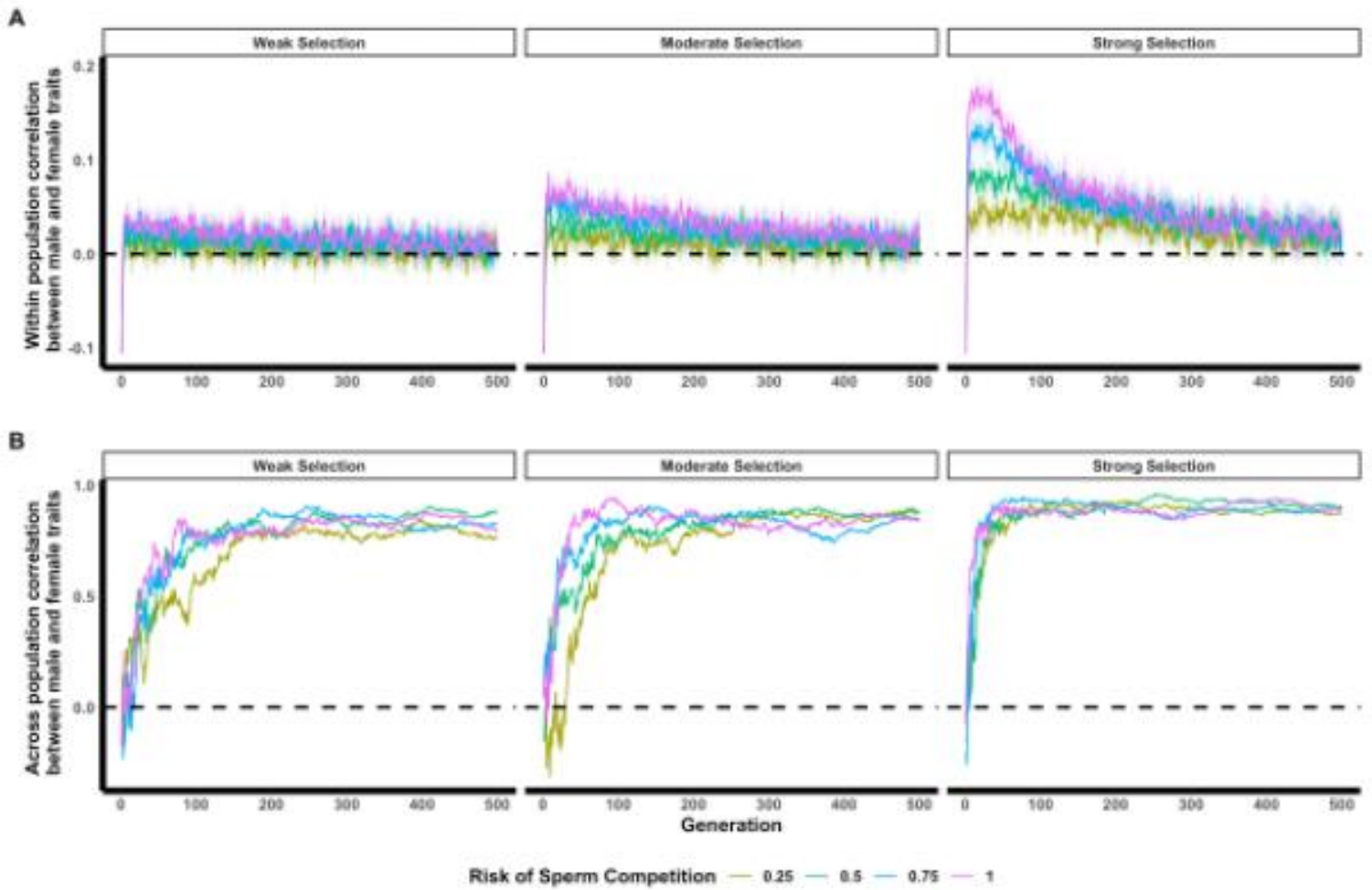
Artist: Matt Kustra

Putting together final plots

```
(Plot1+xlab(""))/Plot2+plot_annotation(tag_levels = "A")
```

```
## Warning: Removed 6571 rows containing non-finite values (stat_summary).

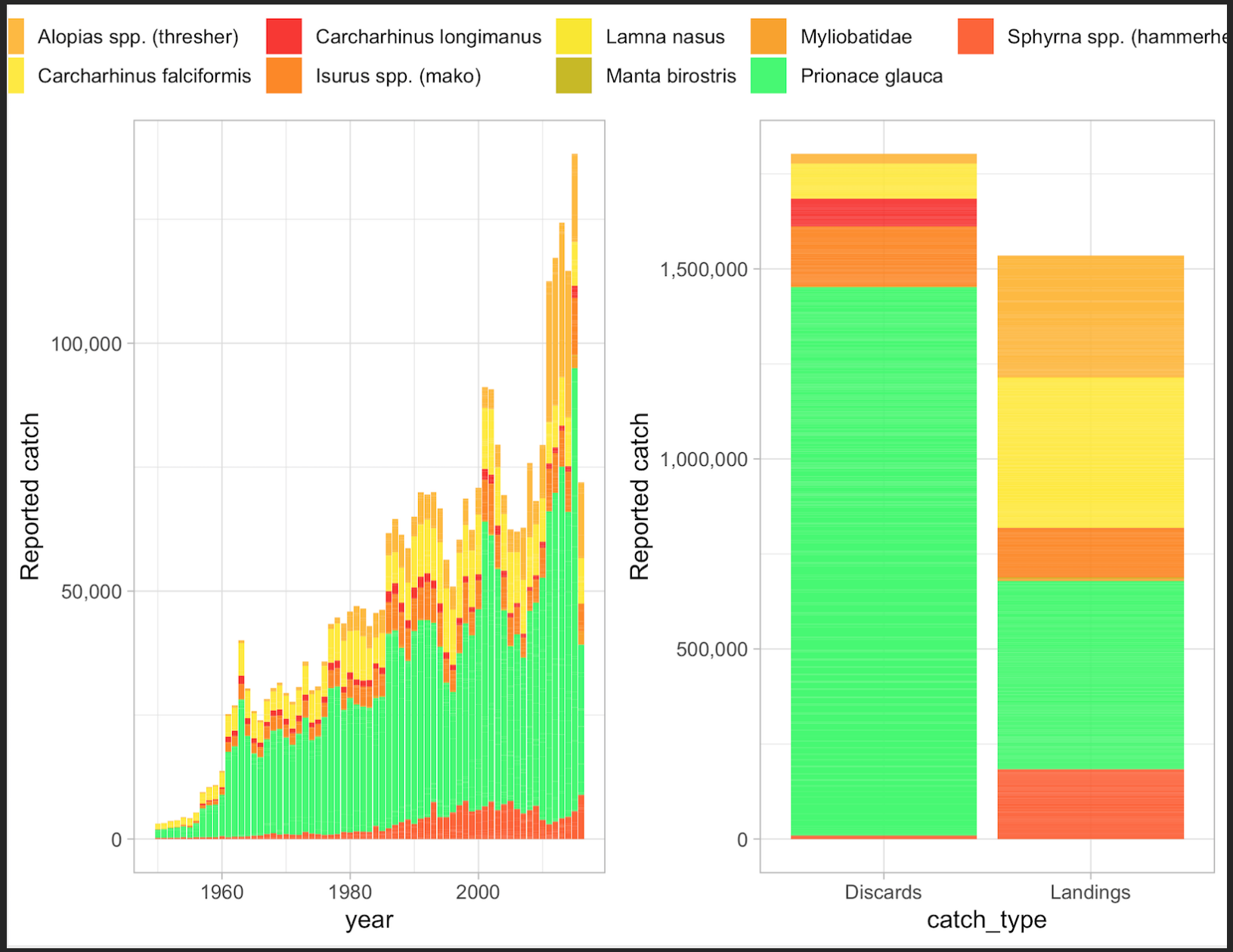
## Warning: Removed 6571 rows containing non-finite values (stat_summary).
```



rmarkdown

www.rstudio.com

Artist: Matt Kustra




```
---
title: "Shark_Data_3"
author: "Megan"
date: "2/18/2021"
output: html_document
---

```{r code 03}
rm(list=ls())
setwd("~/Documents")
 iattc_SAU_data<-read.csv("~/Documents/Another_Shark_03/Data_for _03/iattc_

library(dplyr)
library(ggplot2)
library(dplyr)
library(tidyr)
library(scales)
library(RColorBrewer)
library(gtable)
library(rpivotTable)
library(tidyverse)
library(ggpubr)
library(viridis)
library(hrbrthemes)

#group by genera
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name == "Alopias
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name == "Alopias"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name == "Alopias
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name == "Sphyrna"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name == "Sphyrna
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name == "Sphyrna
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name == "Sphyrna
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name == "Isurus o
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name == "Isurus"]

iattc<- iattc_SAU_data %>%
  filter( scientific_name =="Carcharhinus longimanus"|
          scientific_name =="Carcharhinus falciformis"|
          scientific_name =="Alopias spp. (thresher)"|
          scientific_name=="Isurus spp. (mako)"|
          scientific_name=="Sphyrna spp. (hammerhead)"|
          scientific_name=="Lamna nasus"|
          scientific_name=="Manta birostris"|
          scientific_name=="Myliobatidae"|
          scientific_name=="Prionace glauca")

timeline<-iattc%>%
  #group_by(year, scientific_name, tonnes)%>%
  #summarise(total = sum(tonnes)) %>%
  ggplot(aes(x= year, y= tonnes,fill=scientific_name)) +
  geom_col() +
  scale_y_continuous(labels = comma)+
  labs(y="Reported catch", fill="Species")+
```

Artist: Megan Penland
```

Megan loads in the data and all required packages

Megan groups data by genus

Megan filters data

Plotting with custom colors according to IUCN Status

```
---
title: "Shark_Data_3"
author: "Megan"
date: "2/18/2021"
output: html_document
---

```{r code 03}
rm(list=ls())
setwd("~/Documents")
 iattc_SAU_data<-
read.csv("~/Documents/Another_Shark_03/Data_for _03/iattc_sau2.cs
v",sep=",", header=TRUE)

library(dplyr)
library(ggplot2)
library(dplyr)
library(tidyr)
library(scales)
library(RColorBrewer)
library(gtable)
library(rpivotTable)
library(tidyverse)
library(ggpubr)
library(viridis)
library(hrbrthemes)

#group by genera
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name ==
"Alopias superciliosus"] <- "Alopias spp. (thresher)"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name ==
"Alopias"] <- "Alopias spp. (thresher)"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name ==
"Alopias vulpinus"] <- "Alopias spp. (thresher)"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name ==
"Sphyrna"] <- "Sphyrna spp. (hammerhead)"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name ==
"Sphyrna lewini"] <- "Sphyrna spp. (hammerhead)"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name ==
"Sphyrna mokorran"] <- "Sphyrna spp. (hammerhead)"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name ==
"Sphyrna zygaena"] <- "Sphyrna spp. (hammerhead)"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name ==
"Isurus oxyrinchus"] <- "Isurus spp. (mako)"
iattc_SAU_data$scientific_name[iattc_SAU_data$scientific_name ==
"Isurus"] <- "Isurus spp. (mako)"

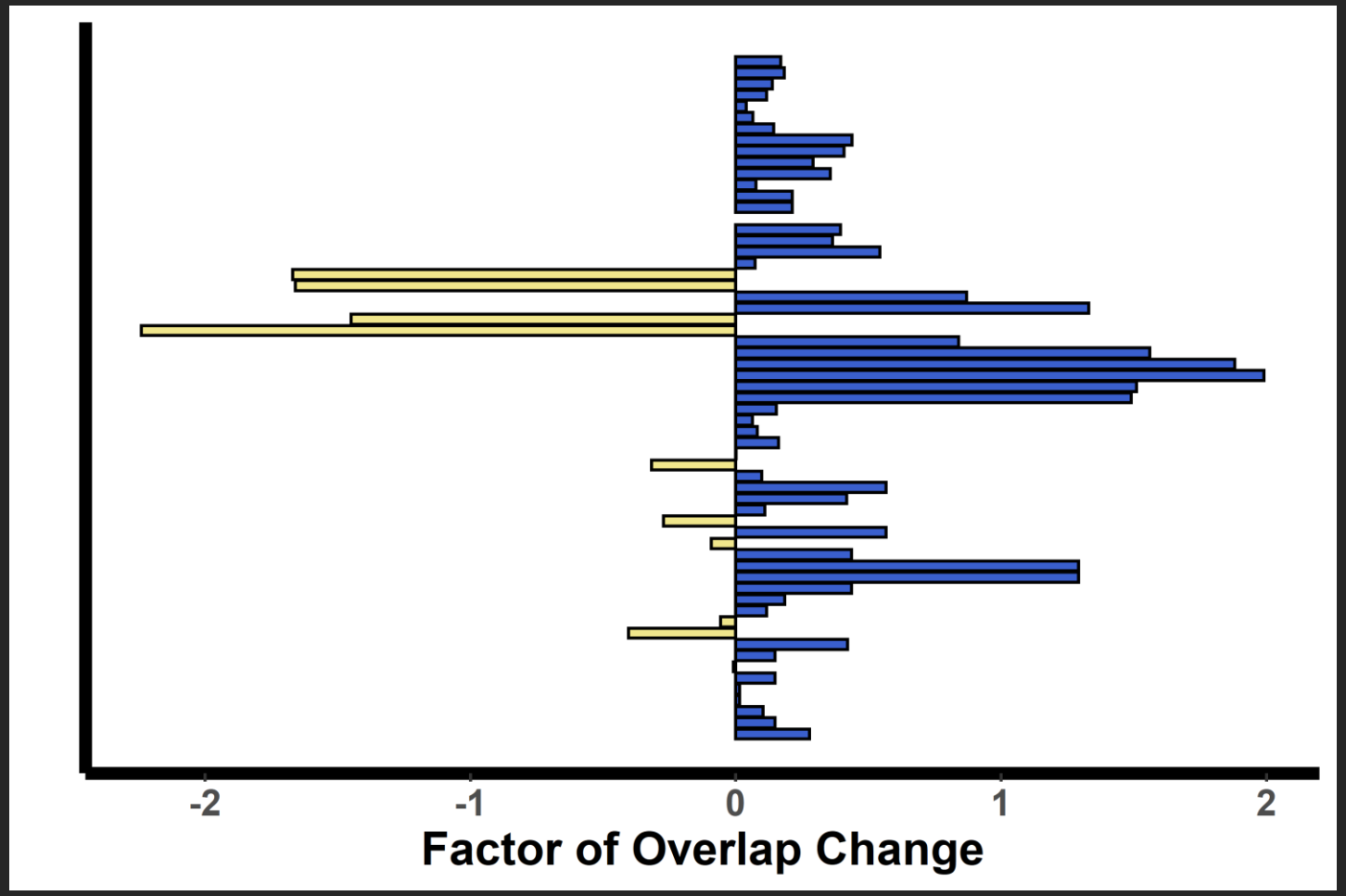
iattc<- iattc_SAU_data %>%
  filter( scientific_name =="Carcharhinus longimanus"|
          scientific_name =="Carcharhinus falciformis"|
          scientific_name =="Alopias spp. (thresher)"|
          scientific_name=="Isurus spp. (mako)"|
          scientific_name=="Sphyrna spp. (hammerhead)"|
          scientific_name=="Lamna nasus"|
          scientific_name=="Manta birostris"|
          scientific_name=="Myliobatidae"|
          scientific_name=="Prionace glauca")

timeline<-iattc%>%
  #group_by(year, scientific_name, tonnes)%>%
  #summarise(total = sum(tonnes)) %>%
  ggplot(aes(x= year, y= tonnes,fill=scientific_name)) +
  geom_col() +
  scale_y_continuous(labels = comma)+
  labs(y="Reported catch", fill="Species")+
  scale_fill_manual(values = c("#FFBA31","#FFEB29","#FA362B","
#FF890A","#FAE900","#C7BA00","#FAA316","#3AFB6C","#FF6330"))+
  theme_light()+
  theme(legend.title = element_blank())+
  theme(legend.position="top")

catch_type<-iattc%>%
  ggplot(aes(x= catch_type, y= tonnes,fill=scientific_name)) +
  geom_col() +
  scale_y_continuous(labels = comma)+
  labs(y="Reported catch", fill="Species")+
  scale_fill_manual(values = c("#FFBA31","#FFEB29","#FA362B","
#FF890A","#FAE900","#C7BA00","#FAA316","#3AFB6C","#FF6330"))+
  theme_light()+
  theme(legend.title = element_blank())
  theme(legend.position="top")

ggarrange(timeline, catch_type, common.legend = TRUE)
ggsave("iattc_sau.tiff")
```

Artist: Megan Penland
```



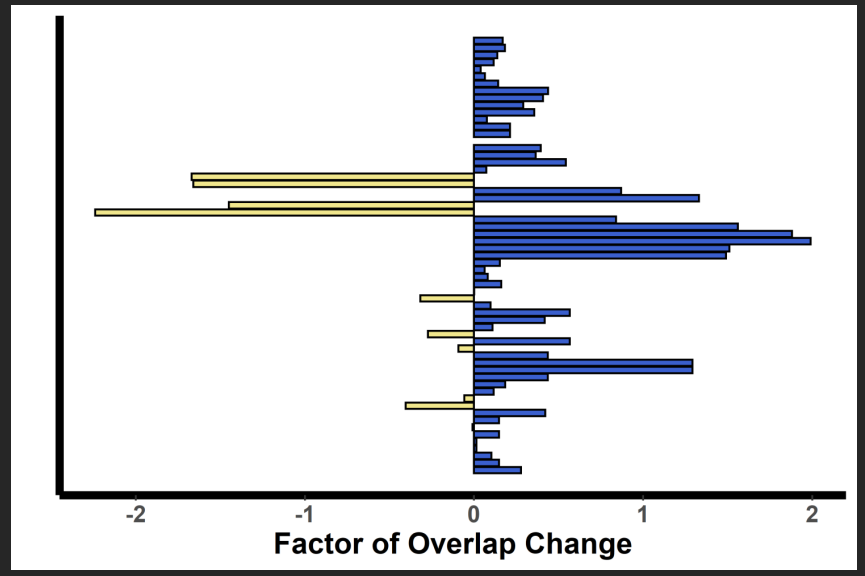

```
---
title: "Data_Viz_Overlap_Fig"
author: "Ishana Shukla"
date: "2/18/2021"
output: pdf_document
---
```{r}
library(tidyverse)
setwd ("C:/Users/ishana/Documents/R/Human")
getwd()
dat <- read.csv("overlap.csv", header=TRUE)
str(dat)
head(dat)
```


```{r setup, include=FALSE}
#Makes the basic plot, flips the coordinates, adds a littl
overlap <-ggplot(data=dat,aes(x=id,y=Factor, fill=Sign,leg
  geom_bar(stat="identity", color="black", show.legend = F
#Choosing appropriate colors (aka yellows for daytime, blu
overlap<- overlap + scale_fill_manual(name="Sign",values=c
#Altering the x and y axis labels
overlap <- overlap+xlab(" ")+ylab("Factor of Overlap Chang
#Changing the formatting of the axis titles
overlap <-overlap + theme(axis.title = (element_text(face=
                                        text=element_text(size=15,
#Removing the tick marks for the y axis, as we don't reall
overlap<-overlap + theme(axis.text.y = element_blank(),
                         axis.ticks.y = element_blank())
#Checking to make sure everything is running smoothly
overlap
#save the figure
ggsave("Figure1_highres.png",overlap,width = 6,height=4,un
```

Artist: Ishana Shukla
```

Ishana loads in data and packages

```
---
title: "Data_Viz_Overlap_Fig"
author: "Ishana Shukla"
date: "2/18/2021"
output: pdf_document
---
```{r}
library(tidyverse)
setwd ("C:/Users/ishana/Documents/R/Human")
getwd()
dat <- read.csv("overlap.csv", header=TRUE)
str(dat)
head(dat)
```
```



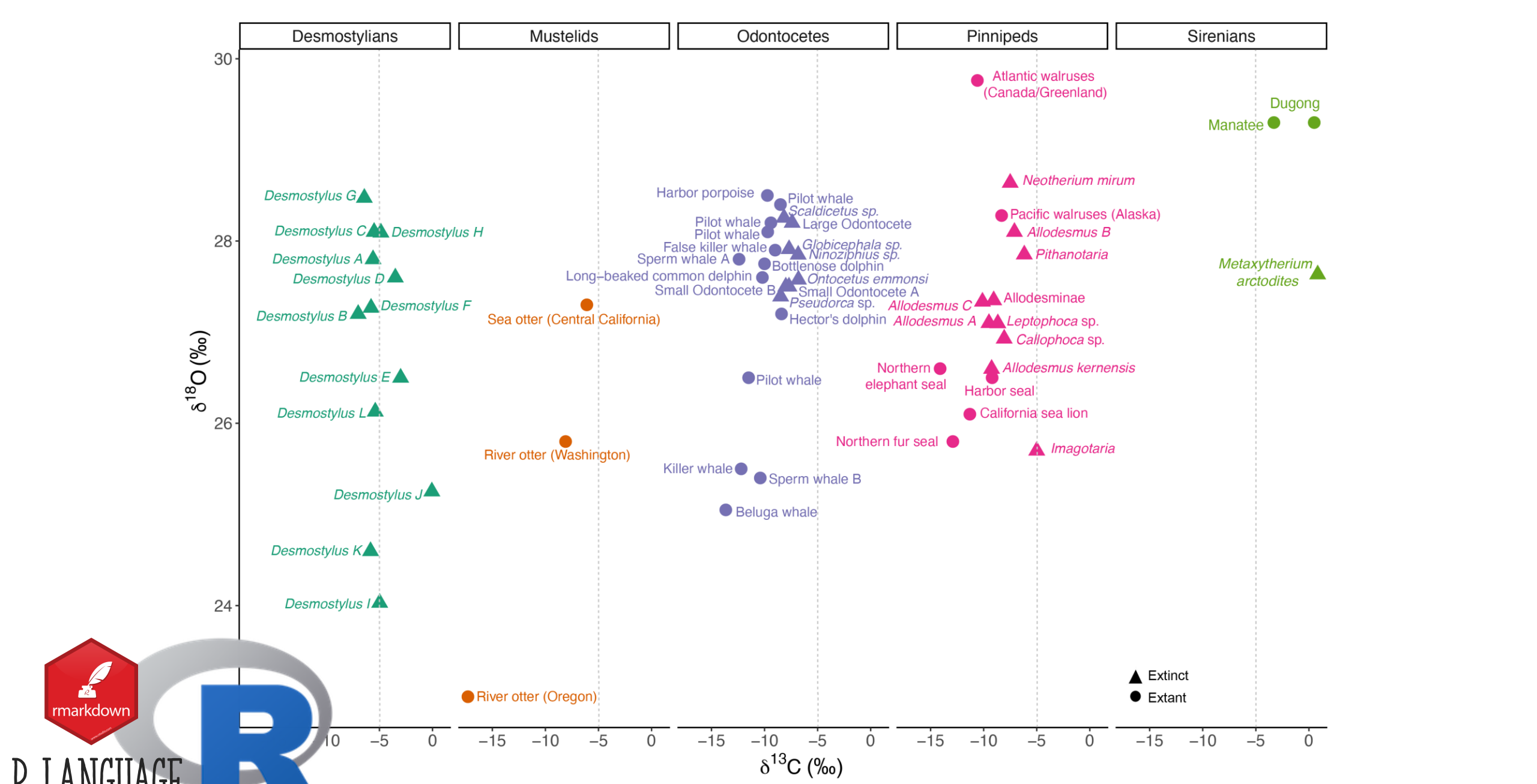
Ishana creates the plot with ggplot geom_bar

```
```{r setup, include=FALSE}
#Makes the basic plot, flips the coordinates, adds a little color, and removes the legend
overlap <-ggplot(data=dat,aes(x=id,y=Factor, fill=Sign,legend=FALSE)) +
  geom_bar(stat="identity", color="black", show.legend = F) + coord_flip()
#Choosing appropriate colors (aka yellows for daytime, blues for nighttime)
overlap<- overlap + scale_fill_manual(name="Sign",values=c("Khaki","royalblue3"))
#Altering the x and y axis labels
overlap <- overlap+xlab(" ")+ylab("Factor of Overlap Change")
#Changing the formatting of the axis titles
overlap <-overlap + theme(axis.title = (element_text(face="bold")),
                                  text=element_text(size=15,face="bold"))
#Removing the tick marks for the y axis, as we don't really need to see case number IDs!
overlap<-overlap + theme(axis.text.y = element_blank(),
                 axis.ticks.y = element_blank())
#Checking to make sure everything is running smoothly
overlap
#save the figure
ggsave("Figure1_highres.png",overlap,width = 6,height=4,units="in",dpi=500)
```
```

rmarkdown

R LANGUAGE

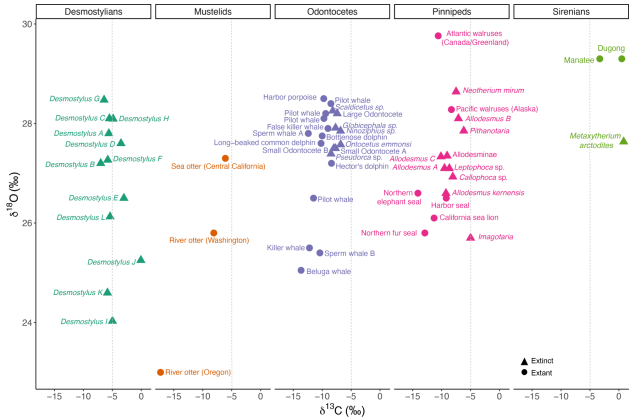
Artist: Ishana Shukla



rmarkdown

R LANGUAGE

Artist: Ana Valenzuela Toro



Data Visualization Assignment#3

Ana M. Valenzuela Toro

February 18, 2021

This is the script used to generate the faceted scatter plot depicting the carbon and oxygen stable isotopes of extant and extinct marine mammals from the eastern and western coast of North America.

Start by cleaning the console and remove previous data and figures.

```
rm(list=ls())
graphics.off()
```

Add the libraries needed

```
library(ggplot2)
library(plotly)
library(ggrepel)
```

Add the corresponding data set

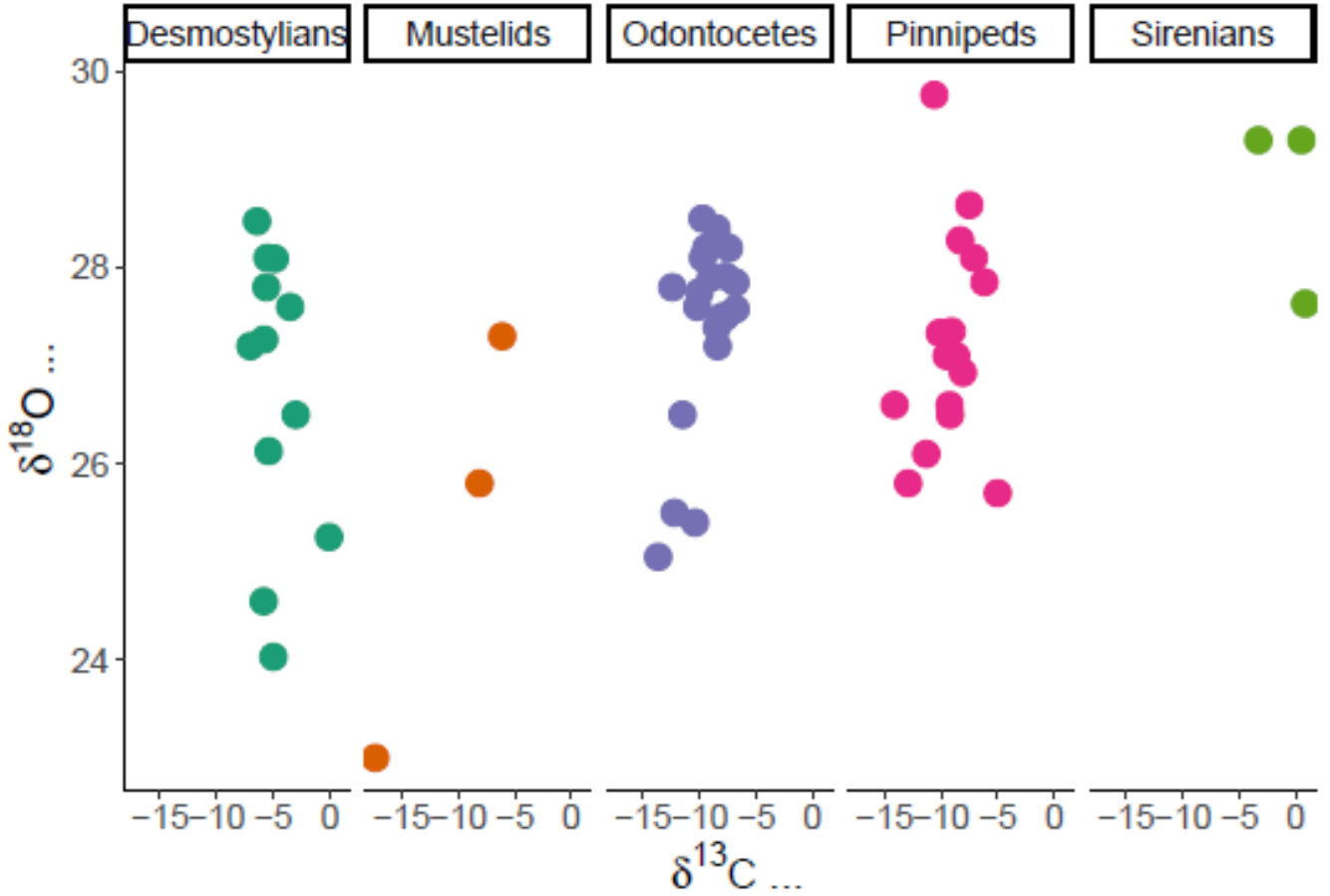
```
Data <-read.csv("isotopes.csv")
```

Make a simple scatter plot with the carbon and oxygen data faceted and colored by taxonomic groups.

```
sp<-ggplot(Data, aes(x=d13C, y=d18O.corrected, color=Group)) +
  geom_point(size=4)+theme_classic()+
  facet_grid(. ~ Group)+ scale_color_brewer(palette="Dark2")+
  theme(legend.position = "none", text=element_text(size=15))+
  xlab(expression({delta}^13*C-'\u2030'))+
  ylab(expression({delta}^18*O-'\u2030'))
```

Call the function sp to see the plot

```
sp
```

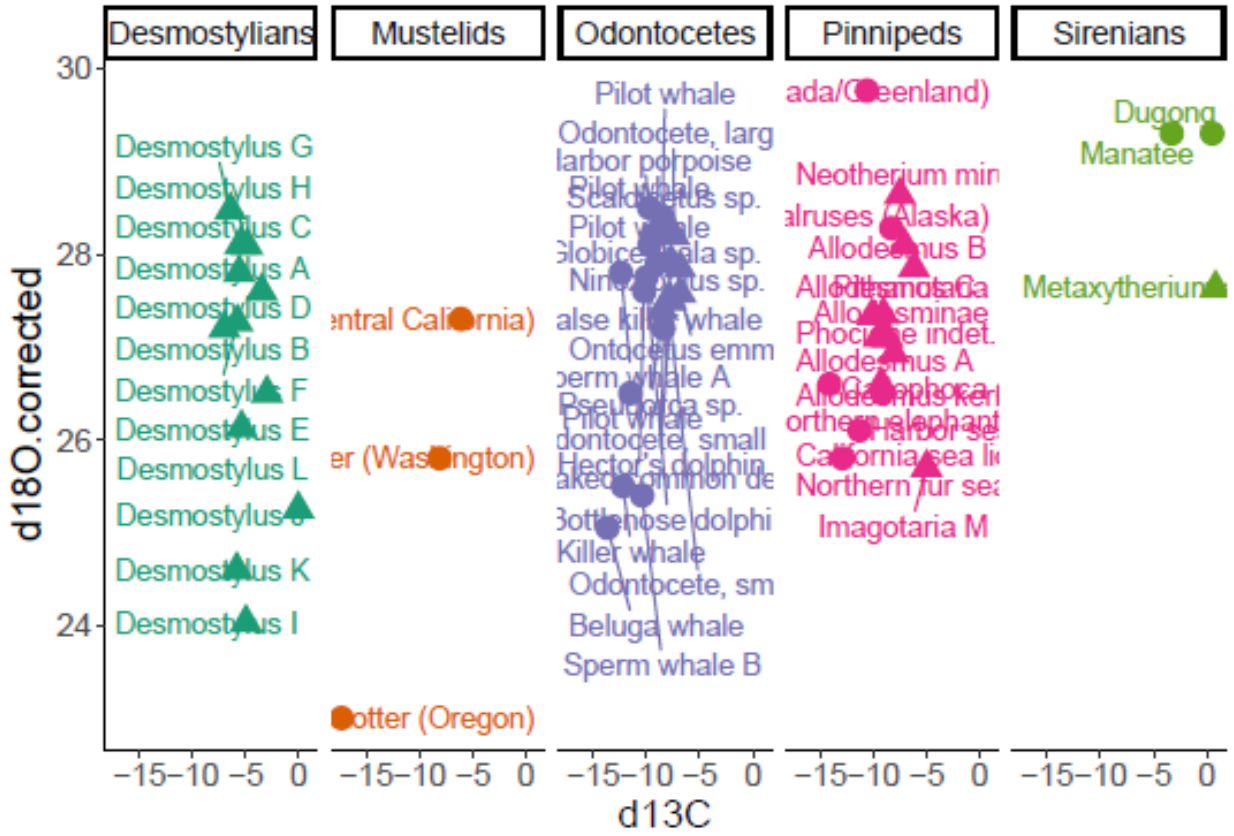


Make the same scatter plot but add labels to each point and different shapes for extinct and extant animals

```
Labelsp<-ggplot(Data, aes(x=d13C, y=d18O.corrected, color=Group, shape=Category, label=Species)) +
  geom_point(size=4)+theme_classic()+
  geom_text_repel()+ facet_grid(. ~ Group)+ scale_color_brewer(palette="Dark2")+
  theme(legend.position = "none", text=element_text(size=15))
```

Call the plot.

```
Labelsp
```



Add per mil symbols to the axis labels

```
Final<-Labelsp+xlab(expression({delta}^13*C-'\u2030'))+
  ylab(expression({delta}^18*O-'\u2030'))
```

Call the function just made to see the plot

```
Final
```

Set working directory to save plots

```
setwd("~/Documents/UCSC/2021/Data Visualization/R")
```

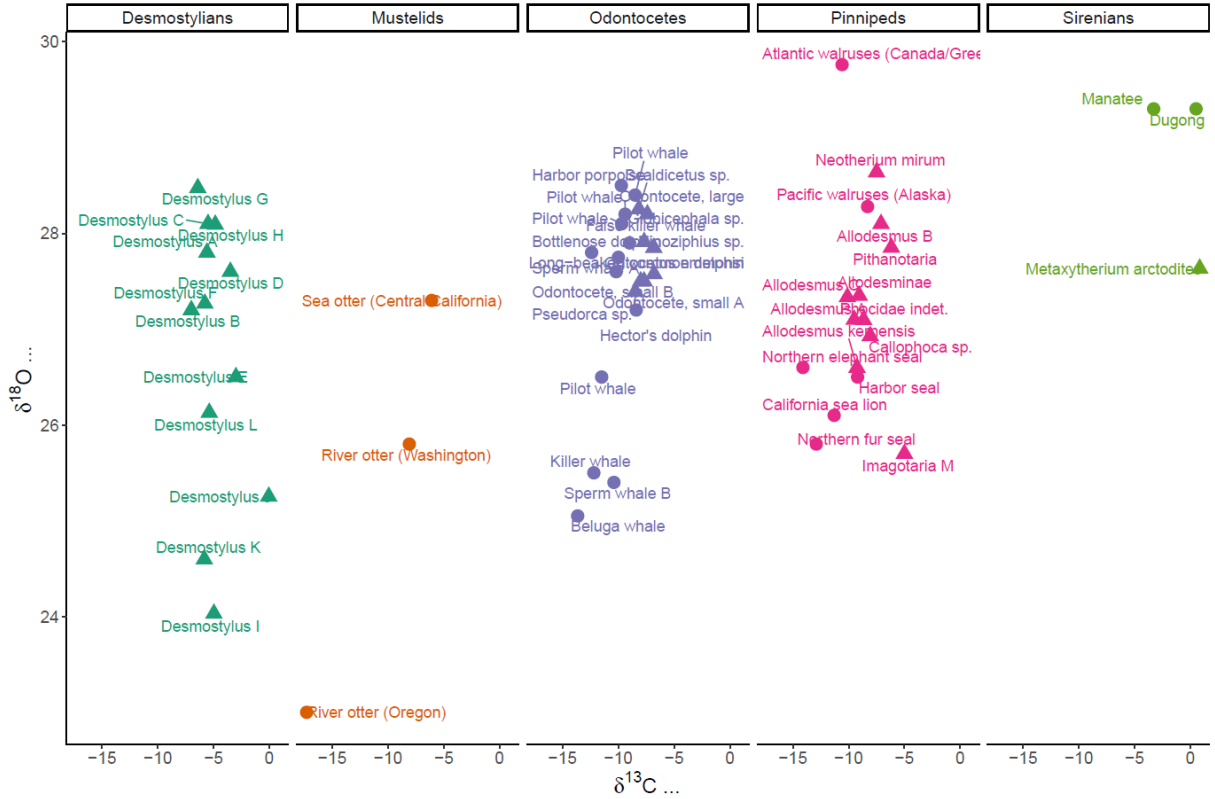
Save the plot in pdf

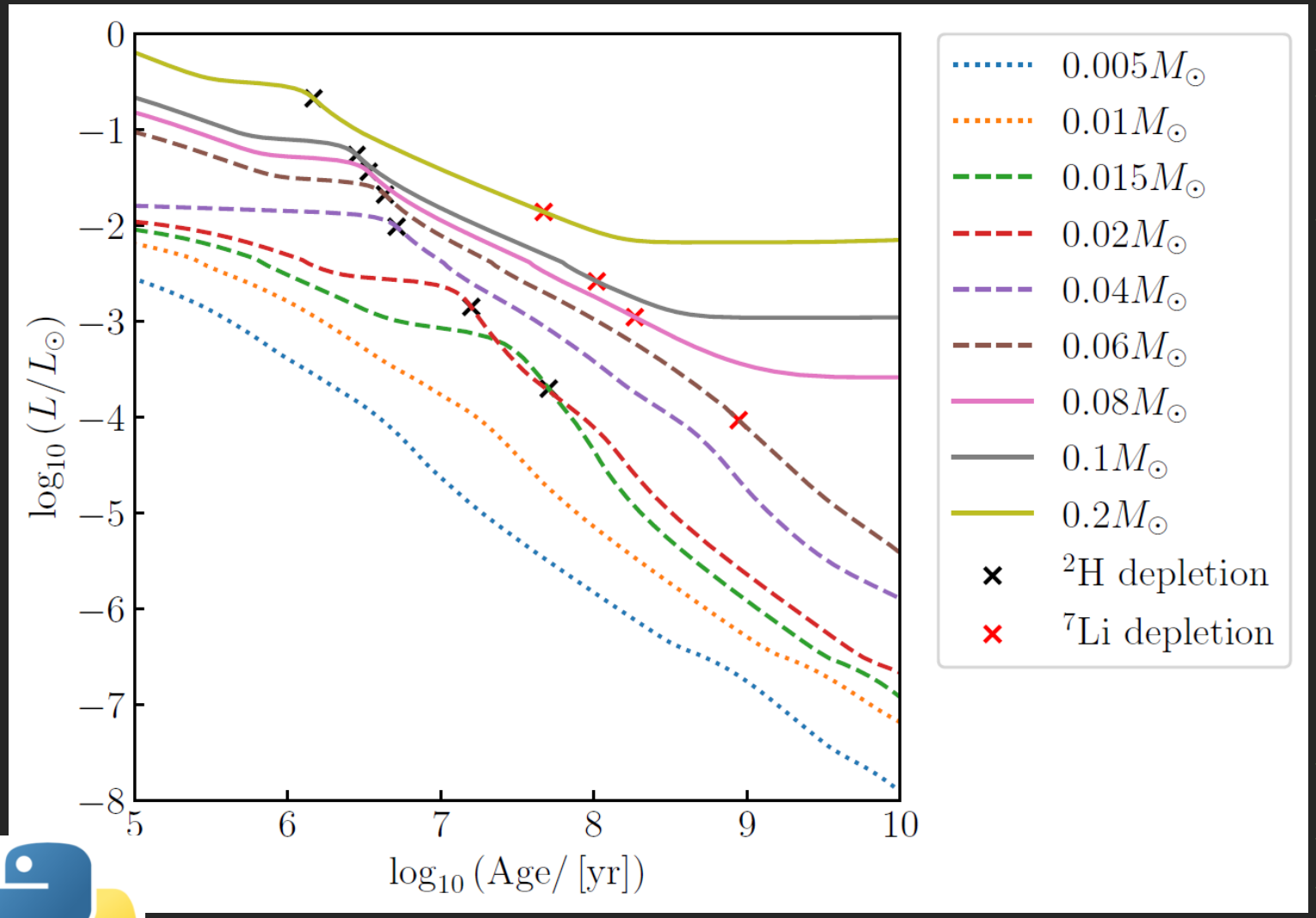
```
ggsave("Final.pdf", width = 30, height = 20, units = "cm", dpi = 300)
```

Save the plot in png

```
ggsave("Final.png", width = 30, height = 20, units = "cm", dpi = 300)
```

FIN






```
import mesa_reader as mr
import matplotlib.pyplot as plt
import numpy as np
import rytools as rt
rt.plot.plottex()

…

# HERE'S THE BEGINNING
if do['c']:
  masses = ['0.005', '0.01', '0.015', '0.02', '0.04','0.06', '0.08', '0.1

  for mass in masses:
#   Generate the points that we will plot with scatter.
    hist[mass] = mr.MesaData(mass + 'MSun/LOGS/history.data')
    logs[mass] = mr.MesaLogDir(mass + 'MSun/LOGS')
    h2_dep_idx[mass] = np.where(hist[mass].center_h2 < 1.e-2 * hist[mass]
    li7_dep_idx[mass] = np.where(hist[mass].center_li7 < 1e-2 * hist[mass
    print("Mass = " + mass + 'MSun')
    if float(mass) < 0.015: obj[mass] = 'p'
    elif float(mass) < 0.08: obj[mass] = 'b'
    else: obj[mass] = 's'
    print("  Burns deuterium: ", len(h2_dep_idx[mass]) > 0)
    print("  Burns lithium: ", len(li7_dep_idx[mass]) > 0)

    if len(h2_dep_idx[mass]) > 0:
      deut_x = np.append(deut_x, hist[mass].log_star_age[h2_dep_idx[mass]
      deut_y = np.append(deut_y, hist[mass].log_L[h2_dep_idx[mass][0]])
    if len(li7_dep_idx[mass]) > 0:
      lith_x = np.append(lith_x, hist[mass].log_star_age[li7_dep_idx[mass
      lith_y = np.append(lith_y, hist[mass].log_L[li7_dep_idx[mass][9]])


  fig, ax = plt.subplots()
  for mass in masses:
#   Plot the luminosity as a function of age
    ax.plot(hist[mass].log_star_age, hist[mass].log_L, label = r'$' + str
ar_mass[0],3)) + r'M_\odot$', ls = lsvals[obj[mass]])
# Make sure both axes have ticks facing inwards
  ax.tick_params(which = 'both', direction = 'in')
# Set the limits to some nice integers
  ax.set_xlim(5, 10)
  ax.set_ylim(-8, 0)
# TeX the labels for better typesetting
  ax.set_xlabel(r'$\log_{10}\lp\text{Age}/\ls\si{\year}\rs\rp$')
  ax.set_ylabel(r'$\log_{10}\lp L/L_\odot\rp$')
# Make the scatter plots
  ax.scatter(deut_x, deut_y, color = 'black', marker = 'x', label = '$^2$
  ax.scatter(lith_x, lith_y, color = 'red', marker = 'x', label = '$^7$Li
#  ax.legend(loc = 0, ncol = 3)
```

PYTHON

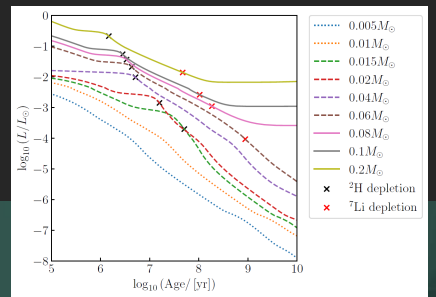
Artist: Ricardo Yarza

```
# HERE'S THE BEGINNING
if do['c']:
  masses = ['0.005', '0.01', '0.015', '0.02', '0.04','0.06', '0.08', '0.1', '0.2']

  for mass in masses:
#   Generate the points that we will plot with scatter.
    hist[mass] = mr.MesaData(mass + 'MSun/LOGS/history.data')
    logs[mass] = mr.MesaLogDir(mass + 'MSun/LOGS')
    h2_dep_idx[mass] = np.where(hist[mass].center_h2 < 1.e-
2 * hist[mass].center_h2[0])[0]
    li7_dep_idx[mass] = np.where(hist[mass].center_li7 < 1e-
2 * hist[mass].center_li7[0])[0]
    print("Mass = " + mass + 'MSun')
    if float(mass) < 0.015: obj[mass] = 'p'
    elif float(mass) < 0.08: obj[mass] = 'b'
    else: obj[mass] = 's'
    print("  Burns deuterium: ", len(h2_dep_idx[mass]) > 0)
    print("  Burns lithium: ", len(li7_dep_idx[mass]) > 0)

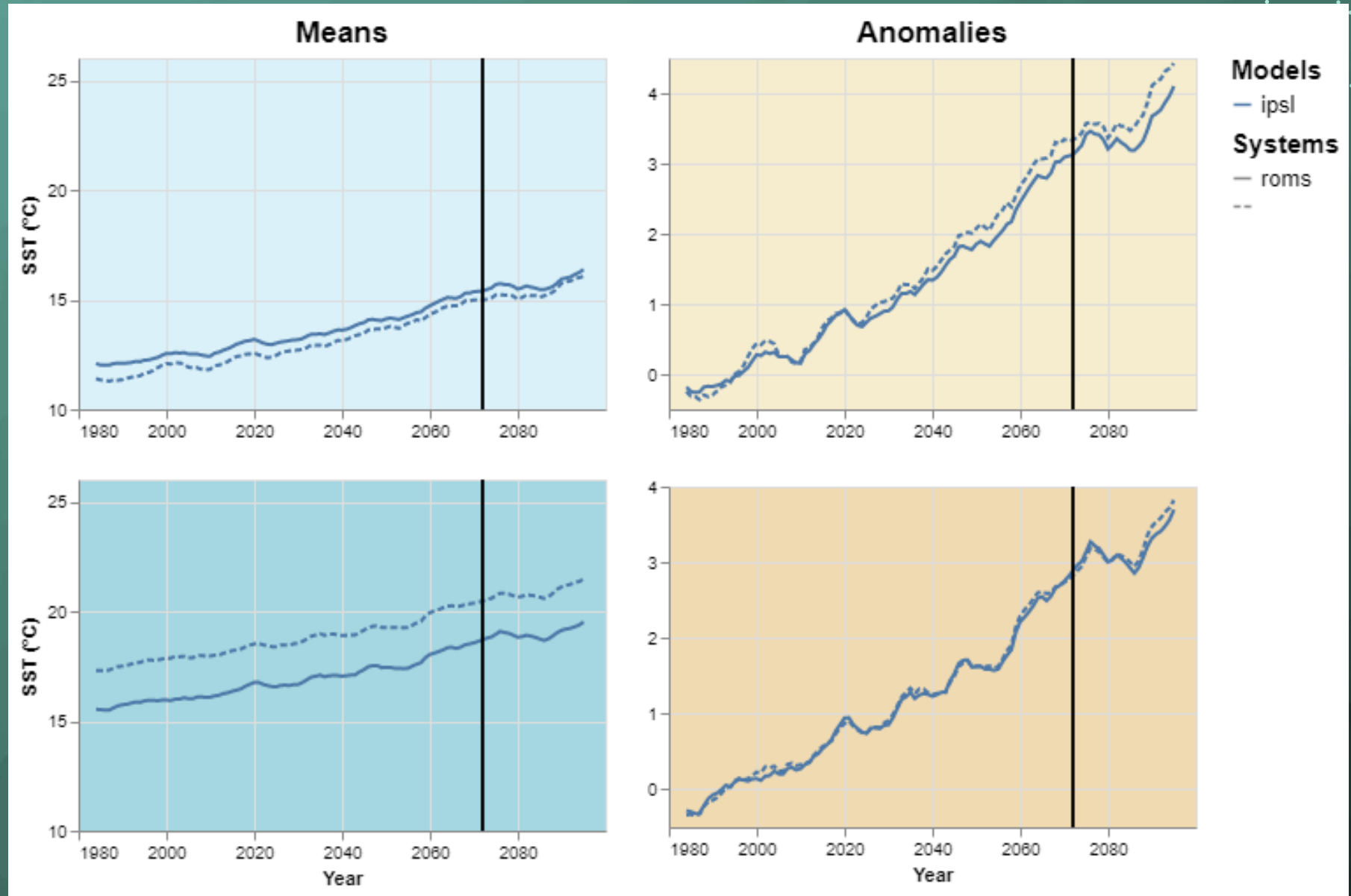
    if len(h2_dep_idx[mass]) > 0:
      deut_x = np.append(deut_x, hist[mass].log_star_age[h2_dep_idx[mass][0]])
      deut_y = np.append(deut_y, hist[mass].log_L[h2_dep_idx[mass][0]])
    if len(li7_dep_idx[mass]) > 0:
      lith_x = np.append(lith_x, hist[mass].log_star_age[li7_dep_idx[mass][9]])
      lith_y = np.append(lith_y, hist[mass].log_L[li7_dep_idx[mass][9]])
```

```
  fig, ax = plt.subplots()
  for mass in masses:
#   Plot the luminosity as a function of age
    ax.plot(hist[mass].log_star_age, hist[mass].log_L,
 label = r'$' + str(round(hist[mass].star_mass[0],3))
+ r'M_\odot$', ls = lsvals[obj[mass]])
# Make sure both axes have ticks facing inwards
  ax.tick_params(which = 'both', direction = 'in')
# Set the limits to some nice integers
  ax.set_xlim(5, 10)
  ax.set_ylim(-8, 0)
# TeX the labels for better typesetting
  ax.set_xlabel(r'$\log_{10}\lp\text{Age}/\ls\si{\year
}\rs\rp$')
  ax.set_ylabel(r'$\log_{10}\lp L/L_\odot\rp$')
# Make the scatter plots
  ax.scatter(deut_x, deut_y, color = 'black', marker =
 'x', label = '$^2$H depletion')
  ax.scatter(lith_x, lith_y, color = 'red', marker = '
x', label = '$^7$Li depletion')
#  ax.legend(loc = 0, ncol = 3)
# Put the legend outside of the figure because it does
n't fit very well inside
  ax.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxe
spad=0., ncol = 1)
# Make it square
  ax.set_aspect(5/8)
# Fix whitespace and whatnot
  plt.tight_layout()
# Save as lossless PDF
  plt.savefig('age_vs_L_several' + do['fmt'], bbox_inc
hes = 'tight')
```



PYTHON

Artist: Ricardo Yarza



https://observablehq.com/@aith/coastal-chart-prototype/3

Artist: Ari Iramanesh – interactive web prototype using Vega Lite

Intro to Coding for Web Design

Intro to plotting in R, Python, and Javascript (D3).

WHAT IS A WEBSITE?

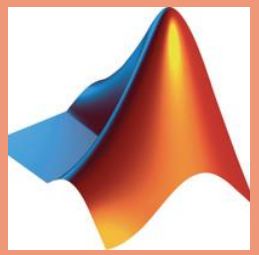
A website is really just a folder of HTML files with instructions for what graphics and text the browser should display on the screen.

A server “serves” those files to anyone browsing your website.

WHAT SHOULD WE USE?

Exploring workflows in R, Javascript, and Python.

MATLAB (not free): High-level multi-paradigm programming language and interactive environment for numerical computation, visualization, and programming



R Studio

Install packages with RSTUDIO CONSOLE

EDIT & RUN CODE:

RSTUDIO

PUBLISH & SHARE CODE

RMARKDOWN

LET PEOPLE INTERACT:

RSHINY

JS

Observable: Include libraries by:

d3=require('d3');

HTML: Include libraries inline

<script src="link"></script>

EDIT & RUN CODE:

VISUAL STUDIO CODE

with HTML, CSS, JAVASCRIPT

PUBLISH & SHARE CODE

OBSERVABLE

LET PEOPLE INTERACT:

PUBLISH WEBPAGE

Install packages through package manager like ANACONDA or in your terminal:

```
conda install pip
```

EDIT & RUN CODE:

SPYDER

through ANACONDA

PUBLISH & SHARE CODE

JUPYTER NOTEBOOK

LET PEOPLE INTERACT:

PUBLISH WEBPAGE

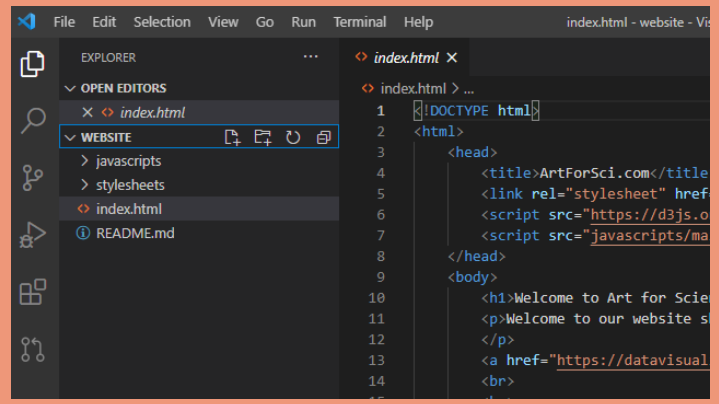
OUR WEB WORKFLOW

Local → Remote

VISUAL STUDIO CODE

Create HTML, CSS, & Javascript files.

Preview changes in browser with Live Server Extension.

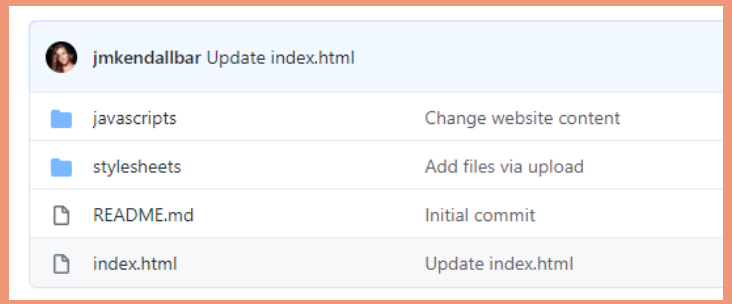


GITHUB

Make a GitHub account.

Create a repository called "Website"

Drop your website files in there.



NETLIFY

Deploy your website from your GitHub repository.

`https://www.netlify.com/`

GOOGLE DOMAINS

Buy your own domain and link it!

`https://domains.google.com/`

POST-MEETING TO DO'S

- Due in two weeks: Interactive Website
  - Use HTML/CSS/Javascript, RShiny, Observable, or Jupyter Notebooks to create an interactive web interface where anyone can interact with your data. Use this as an opportunity to submit something you would like feedback on for your final project.
- Next week, we will be exploring data access online through APIs and providing feedback and help for your interactive websites. We will have a guest appearance by someone who uses open-source APIs to perform scientific research.