A Shallow Dive into Deep Sea Data

Sarah Solie and Arielle Fogel

7/18/2018

Introduction

The datasets

This data expedition will utilize the **World Ocean Atlas (WOA) database** to explore two deep sea physical oceanography parameters: **temperature** and **salinity**. The WOA database is compiled and maintained by the National Oceanic and Atmospheric Administaration and contains data for World Ocean climatologies across depth, location, and over time. These data are publicly and freely available online and can be accessed at: https://www.nodc.noaa.gov/OC5/woa13/woa13data.html

Today, we will explore the World Ocean, specfically assessing temperature and salinity across location and oceanic depth. You will be assigned to either TEAM TEMPERATURE or THE SALINITY SQUAD and will investigate one variable.

The data you will download are computed averages for the time span from 1955 through 2012 and at 1° spatial resolution. Use the link above to access and download the dataset for your oceanographic variable. Before downloading the data, make sure to select the following on the left hand side of the page:

- "CSV" under available formats
- 1° grid
- "Statistical mean"

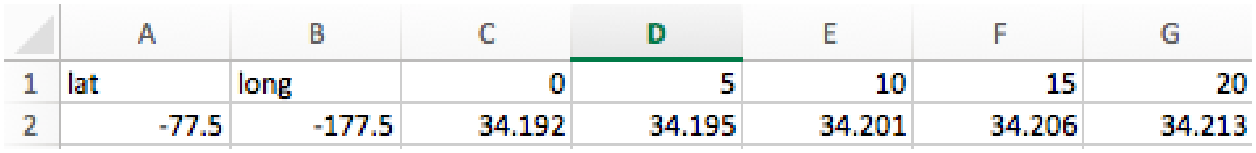
Save the data file to your desktop and name it “woa_temp1.csv” or “woa_sal1.csv”.

Inspect the data (part 1)

Open your data file in Excel or Numbers. What do you notice? What is in each row? What is in each column? Are there headers? Variable explanations? What’s going on???

Tidy the data (part 1)

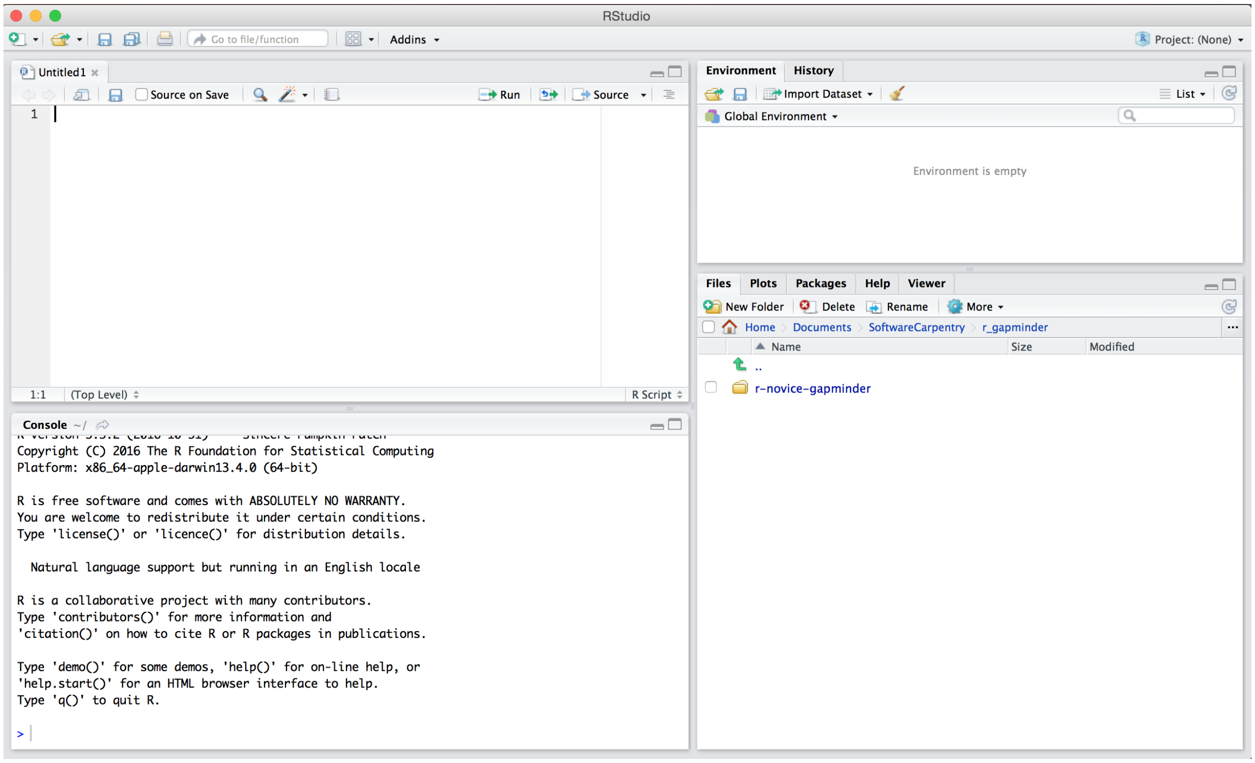
Let’s clean this data file up a bit – it will make our lives easier when we begin to work with this file later in R.
1. Delete row 1 by highlighting the row, right clicking, and selecting “Delete”.
2. Rename A1 “lat” and B1 “long”.
3. Remove the explanation from C1 so that the cell simply reads “0”.
4. Resave your file as woa_temp1.csv or woa_sal1.csv once the headers look something like this:



	A	B	C	D	E	F	G
1	lat	long	0	5	10	15	20
2	-77.5	-177.5	34.192	34.195	34.201	34.206	34.213

R Studio

We’ll be working with RStudio today. R is a free software program that provides powerful tools for statistical computing and graphics, and RStudio is an integrated development environment frequently used for easier programming in R. Once you have downloaded R (https://cran.r-project.org/) and RStudio (http://www.rstudio.com/products/rstudio/), go ahead and open RStudio. Select File > New File > R Script. Your R programming environment should now look like this:



```
R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin Patch"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.4.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Writing R Code!

The panel in the upper left corner of your environment is the script editor, and this is where we'll write and execute code. The script editor enables you to modify and reuse code as needed, and transfers executed code to the "console", where it can no longer be modified. To execute any command in RStudio, keep the cursor in the line of code you plan to execute, and hit "command + enter". (You can also **highlight a chunk of code and use the same keystroke to run the selected block.**) Alternatively, you can select the relevant code and click "Run" in the upper right corner of your console.

if your computer is from China, run this line of code before starting:
Sys.setenv(LANG = "en")

The first step in writing any piece of code is always setting up your environment. For this data exploration, we will need to install and then load a couple of nifty, add-on R packages. These include:

- the "tidyverse" package which includes two packages that we will use in this expedition: "ggplot2", and "dplyr"

- the "ggpubr" package

The first time you want to use an add-on R package, you must first install it using the install.packages command (this only needs to be done once). Then, you must load the library using the library command (this needs to be done every time you create a new R file).

Go ahead and install and then load your libraries!

```
#install the tidyverse package
#install.packages("tidyverse")
#install the ggpubr package
#install.packages("ggpubr")
#load the tidyverse library
library(tidyverse)
#load the ggpubr library
library(ggpubr)
```

Next, we need to import our dataset into RStudio. We need to write a bit of code that locates the data file we've stored on our desktop to make it accessible for our use here in RStudio. We're going to tell R to "read" and name the data file with the following command:

```
#read in the salinity dataset and name it "salinity"
#use check.names=FALSE to ensure that depth values remain unaltered
salinity <- read.csv("~/Desktop/woa_sal1.csv", check.names=FALSE)
```

or

```
#read in the temperature dataset and name it "temp"
temp <- read.csv("~/Desktop/woa_temp1.csv", check.names=FALSE)
```

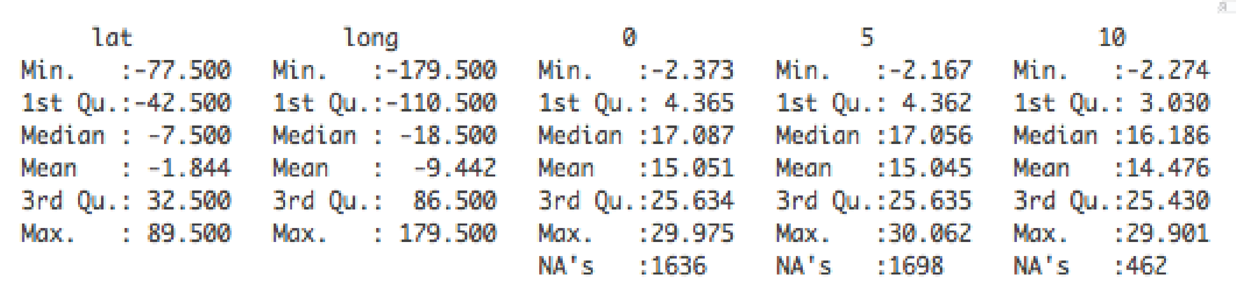
Inspect the data (part 2)

Does your "Global Environment" now include your data file? How many observations are in your dataset? How many variables? Click on the data file to view the dataset. How is depth represented here?

You can also summarize your data to get a statistical overview of your data. You should write this bit of code in the **"Console"** rather than the text-editor, since we don't need to redo this every time we run the code.

```
#summarize the data
summary(salinity)
summary(temp)
```

The first row of that summary should look like this (these values are for the temperature file, though the salinity file will be similarly structured):



```
      lat              long                0                 5                10
 Min.   :-77.500   Min.   :-179.500   Min.   :-2.373   Min.   :-2.167   Min.   :-2.274
 1st Qu.:-42.500   1st Qu.:-110.500   1st Qu.: 4.365   1st Qu.: 4.362   1st Qu.: 3.030
 Median : -7.500   Median : -18.500   Median :17.087   Median :17.056   Median :16.186
 Mean   : -1.844   Mean   :  -9.442   Mean   :15.051   Mean   :15.045   Mean   :14.476
 3rd Qu.: 32.500   3rd Qu.:  86.500   3rd Qu.:25.634   3rd Qu.:25.635   3rd Qu.:25.430
 Max.   : 89.500   Max.   : 179.500   Max.   :29.975   Max.   :30.062   Max.   :29.901
                                      NA's   :1636     NA's   :1698     NA's   :462
```

Tidy the data (part 2)

We need to rearrange the dataset to make it easier to work with depth as a variable in our analysis. To do so, we're going to gather and flip the data frame such that we have a table that is x (~4 million) rows by 4 columns using the following code:

```
#gather and flip the data frame, creating a new variable, "depth"
#also, name this new data frame "tidy_sal"
tidy_sal <- salinity %>%
  gather(`0`:`5500`, key = "depth", value = "salinity")
#use the summary command on the new gather and flipped data frame.
#inspect each variable and make sure they are in the correct format (e.g. numeric, character, etc.).
#Run summary(tidy_sal) in the console

#uh oh! depth is being treated as a character string when it should be a number!
#change depth to a numeric variable
tidy_sal$depth <- as.numeric(tidy_sal$depth)

#the temperature code is as follows:
tidy_temp <- temp %>%
  gather(`0`:`5500`, key = "depth", value = "temperature")
#summarize and inspect the data using summary(tidy_temp)
tidy_temp$depth <- as.numeric(tidy_temp$depth)
```

You should again inspect and summarize the new tidy datasets. Now, how is depth represented in the data file? How many observations? How many variables?

Exploratory Data Analysis

R is a powerful tool for statistical analyses, so let's use it to explore how temperature and salinity vary over depth in the three primary climate zones: **polar, temperate, and tropical**. Specifically, we will answer the following questions:

1. *How does the average temperature/salinity at 3000m differ from the average surface temperature/salinity?*
(i) Create facet-wrapped boxplots by climate zone
(ii) Conduct a t-test to compare means

2. *How can we describe temperature/salinity trends across depth?*
(i) Create facet-wrapped scatter plots by climate zone

(ii) Plot averages to describe trends across depth

To answer these questions, we'll need to do some further data organization to make our datasets more manageable. We can outline what we need to do as follows:

```
# filter the dataset to include only those lat-long coordinates with depth observations
#to at least 3300m

# create a new variable, "climate zone", and assign each lat-long coordinate to its
#appropriate climate zone using the mutate() function

# plot the data using boxplots

# analyze the data using t-tests

# plot the data using scatterplots

# plot average values across depth to describe trends
```

Filter and subset the data.

These datasets are MASSIVE (on the order of 4 million observations), and will take a long time to analyze without access to supercomputing power. Since we're interested in variation across depth, we'll make things more manageable by subsetting our data to exclude inshore/shallow coordinates.

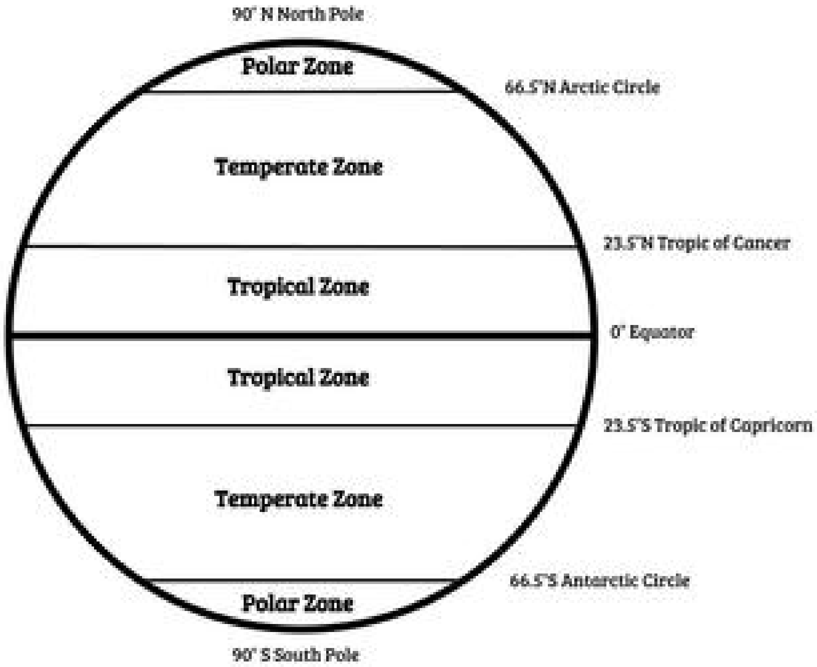
```
#filter dataset to include only those coordinates with depth observations to at least 3300m
#count the number of temperature observations for each lat-long coordinate
tidy_temp2 <- tidy_temp %>%  group_by(lat, long) %>%
  summarise(temp_obs=sum(!is.na(temperature)))
#subset the dataset to include only those lat-long coordinates with at least
#80 temperature observations (80 observations = 3300m)
filter_temp <- subset(tidy_temp2, temp_obs>80)
temp_filtered <- inner_join(tidy_temp, filter_temp, by=c("lat"="lat", "long"="long"))

#the code for salinity is as follows:
tidy_sal2 <- tidy_sal %>%  group_by(lat, long) %>%
  summarise(salinity_obs=sum(!is.na(salinity)))
filter_sal <- subset(tidy_sal2, salinity_obs>80)
sal_filtered <- inner_join(tidy_sal, filter_sal, by=c("lat"="lat", "long"="long"))
```

Be sure to summarize and inspect the data as before. Does anything look out of place?

Create a climate zone designation variable using mutate().

We're going to use latitude to define and designate climate zones. Designations are as follows:

Tropical latitudes: -23.5° to 23.5° ; Temperate latitudes: -66.5° to -23.5° and 23.5° to 66.5° ; Polar latitudes: less than -66.5° and greater than 66.5°



https://ecdn.teacherspayteachers.com/thumbitem/Climate-Zones-Diagram-3610545-1516756656/original-3610545-1.jpg

```
#assign climate variables based on latitude
temp_analysis <- temp_filtered %>%
  mutate(climate=ifelse(abs(lat)<23.5, "tropical",
                        ifelse(abs(lat)>66.5, "polar", "temperate")))
#ensure "climate" is treated as a factor
temp_analysis$climate <- as.factor(temp_analysis$climate)
#summarize and inspect the data using summary(temp_analysis)

#salinity code
sal_analysis <- sal_filtered %>%
  mutate(climate=ifelse(abs(lat)<23.5, "tropical",
                        ifelse(abs(lat)>66.5, "polar", "temperate")))
sal_analysis$climate <- as.factor(sal_analysis$climate)
#summarize and inspect the data using summary(sal_analysis)
```

(1) Compare salinity/temperature at 0m and 3000m in the three climate zones using boxplots

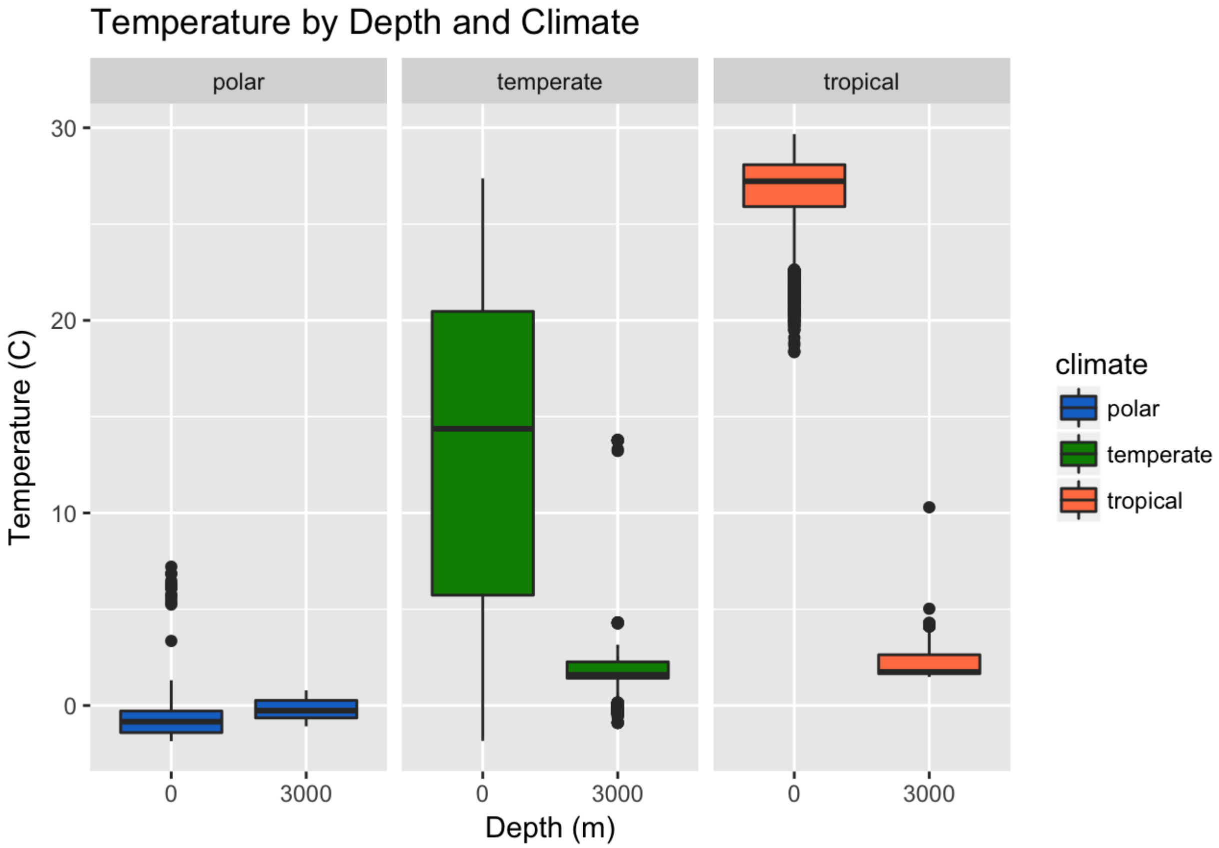
We're going to create boxplots to get a broad sense of how salinity/temperature might vary with depth in the three climate zones. Let's make some predictions!

1. Do you expect the same trends across all three climate zones?
2. How much variation do you expect within each climate zone? At each depth?

(i) Create Boxplots

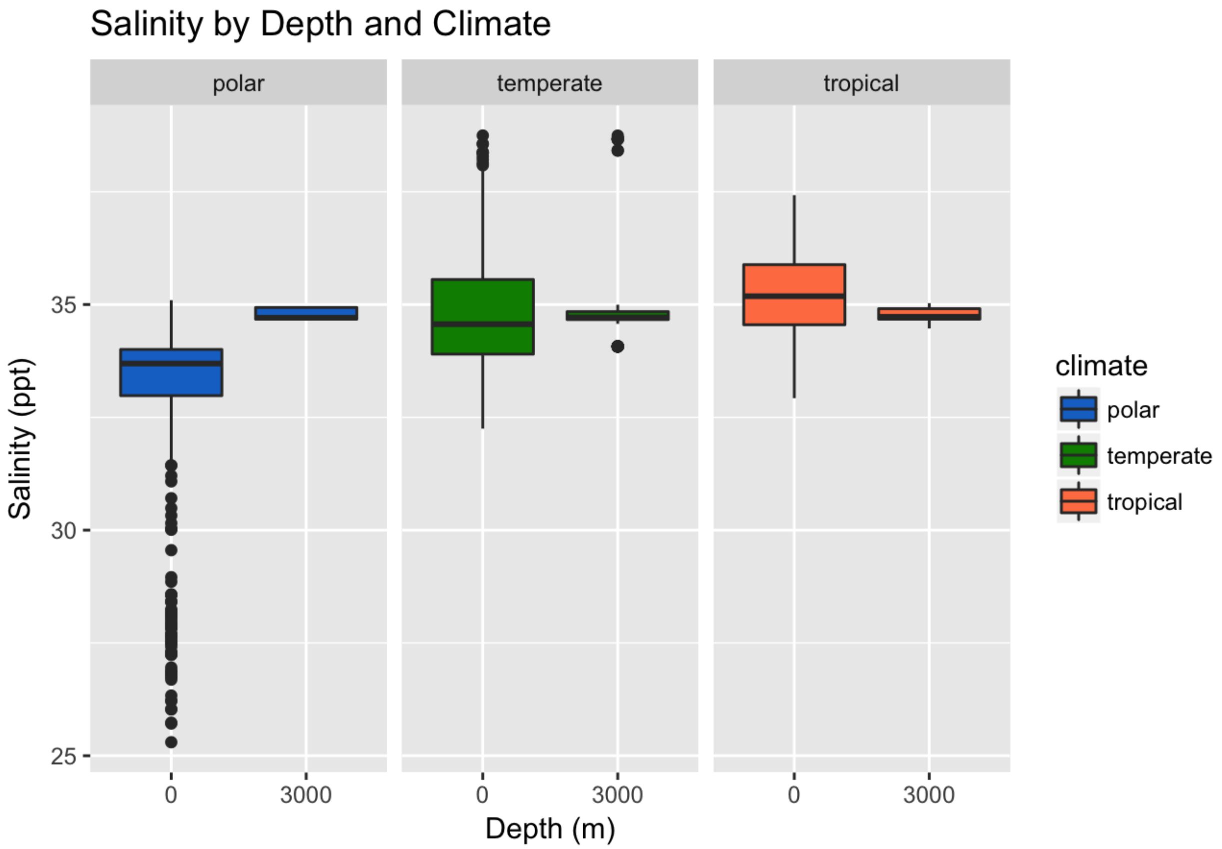
```
#select temperature data for analysis (depth = 0m and depth = 3000m)
box_temp <- filter(temp_analysis, depth == "0"|depth == "3000")
```

```
#you know the drill...summarize and inspect the data using summary(box_temp)

#treat depth as a categorical variable
box_temp$depth <- as.factor(box_temp$depth)

#make a boxplot with depth on the x axis and temperature on the y
temp_boxplot <- ggplot(data = box_temp, aes(x = depth, y = temperature, fill = climate)) +
  geom_boxplot() +
  #create an intuitive color scheme
  scale_fill_manual(values = c("dodgerblue3", "green4", "coral")) +
  #plot by climate
  facet_wrap(~climate) +
  #label axes
  xlab("Depth (m)") + ylab("Temperature (C)") +
  #add a title
  ggtitle("Temperature by Depth and Climate")

#visualize the boxplot
temp_boxplot
```



```
#Same for salinity

#select data for analysis (depth = 0m and depth = 3000m)
box_sal <- filter(sal_analysis, depth == "0"|depth == "3000")

#treat depth as a categorical variable
```

```
box_sal$depth <- as.factor(box_sal$depth)

#make a boxplot with depth on the x axis and temperature on the y
sal_boxplot <- ggplot(box_sal, aes(depth, salinity, fill = climate)) +
  geom_boxplot() +
  scale_fill_manual(values = c("dodgerblue3", "green4", "coral")) +
  facet_wrap(~climate) +
  xlab("Depth (m)") + ylab("Salinity (ppt)") +
  ggtitle("Salinity by Depth and Climate")

#visualize the boxplot
sal_boxplot
```



3. What do these boxplots tell you about your data?

4. Were your predictions correct?

5. Are there any outliers in your plots? Should there be?

6. Use what you have learned about physical oceanography to generate some hypotheses about what these outliers might represent.

(ii) Compare Means (t-test)

You probably have a general idea of how to describe the relationship between surface and depth values based on a visual assessment of your boxplots, but we will use a t-test to describe those relationships with statistics. A t-test will test the hypothesis that two means are, in fact, different from one another. The null hypothesis is that the means are the same, and this result would yield a t-value = 0. The p-value here tells us whether our rejection of the null hypothesis is statistically significant, given the sample size. Generally, we take **p = 0.05 as the cutoff for significance**.

```r
#compare temperature means
#polar temperature
polar_temp <- filter(box_temp, climate =="polar")
t.test(temperature~depth, polar_temp)

#temperate temperature
temperate_temp <- filter(box_temp, climate =="temperate")
t.test(temperature~depth, temperate_temp)

#tropical temperature
tropical_temp <- filter(box_temp, climate =="tropical")
t.test(temperature~depth, tropical_temp)

#compare salinity means
#polar salinity
polar_sal <- filter(box_sal, climate =="polar")
t.test(salinity~depth, polar_sal)

#temperate salinity
temperate_sal <- filter(box_sal, climate =="temperate")
t.test(salinity~depth, temperate_sal)

#tropical salinity
tropical_sal <- filter(box_sal, climate =="tropical")
t.test(salinity~depth, tropical_sal)
```

7. Are the mean values different at the surface vs. at 3000m depth in each climate zone?
8. How confident are you in rejecting the null hypothesis?

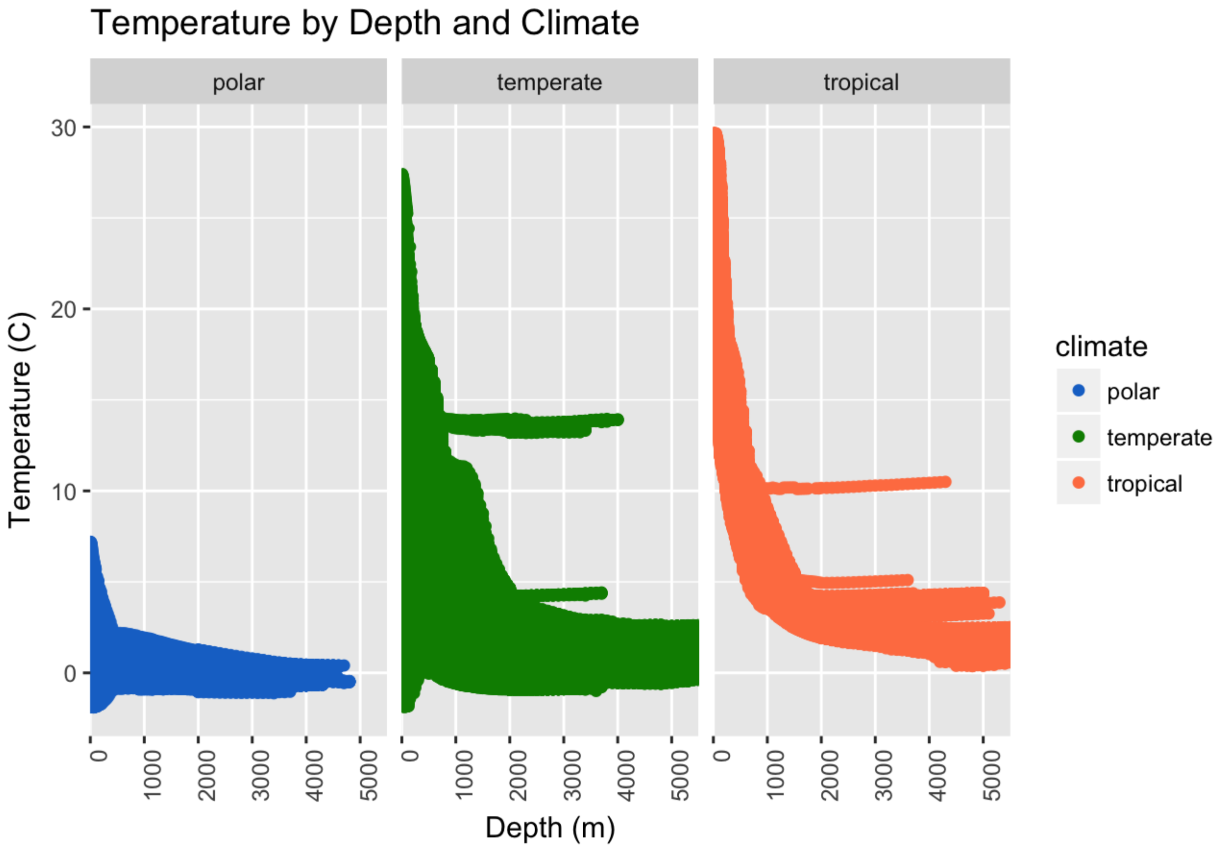
(2) Describing trends across depth

Now we're going use scatterplots to take a closer look at variation across continuous depth. These plots will include every observation at each depth for a total of ~1 million observations, so you can expect the code to take a little while to run. Be patient.

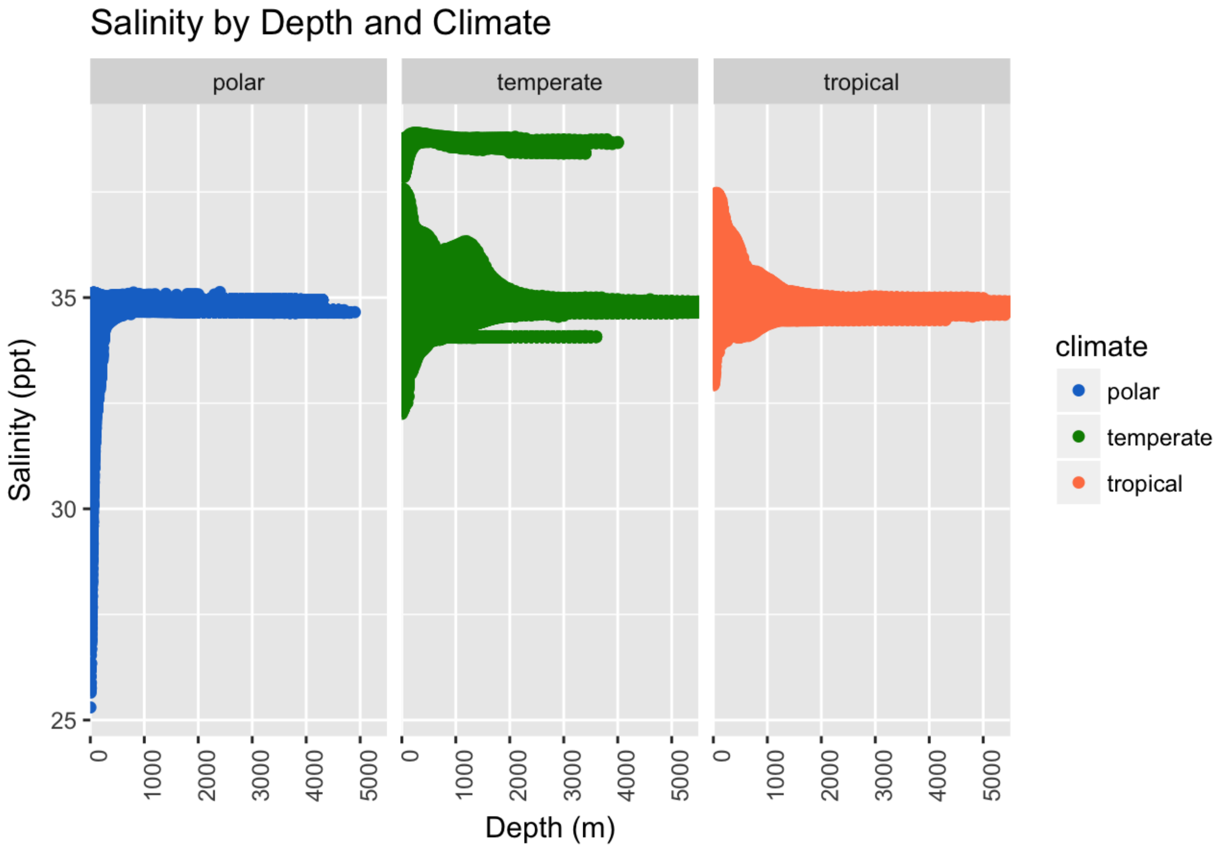
(i) Make a scatterplot of temperature/salinity as a function of depth

The first part of this code should look really familiar

```r
#Make a scatterplot of temperature as a function depth, faceted by climate.
temp_scatterplot <- ggplot(temp_analysis, aes(depth, temperature, color = climate)) +
  geom_point() +
  scale_color_manual(values = c("dodgerblue3", "green4", "coral")) +
  ylab("Temperature (C)") +
  #the x axis gets ugly with all of these depth values, so we'll select a few landmarks
  scale_x_discrete(name = "Depth (m)", limits = (0:5500), breaks = seq(0, 5500, 1000)) +
  #and rotate the axis text
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  facet_wrap(~climate) +
  ggtitle("Temperature by Depth and Climate")
#visualize the scatterplot (this will take a while)
temp_scatterplot
```



```
#same for salinity
sal_scatterplot <- ggplot(sal_analysis, aes(depth, salinity, color = climate)) +
  geom_point() +
  scale_color_manual(values = c("dodgerblue3", "green4", "coral")) +
  ylab("Salinity (ppt)") +
  scale_x_discrete(name = "Depth (m)", limits = (0:5500), breaks = seq(0, 5500, 1000)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  facet_wrap(~climate) +
  ggtitle("Salinity by Depth and Climate")
#visualize the scatterplot (this will take a while)
sal_scatterplot
```



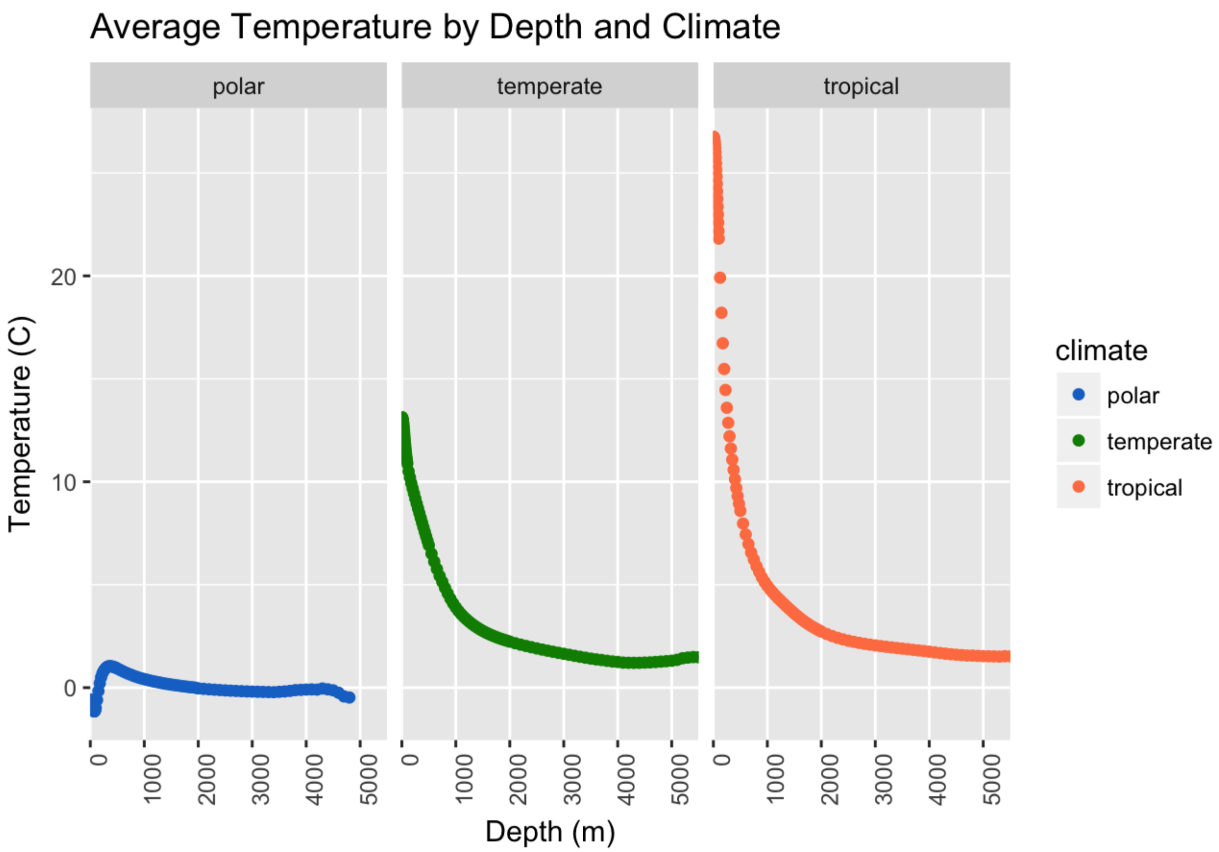
Cool! You have made some graphs! What can you say about them?

9. Do your data follow some generalizable trend?
10. Are there any anomalous points? If so, how might you interpret these data? What could be going on here?

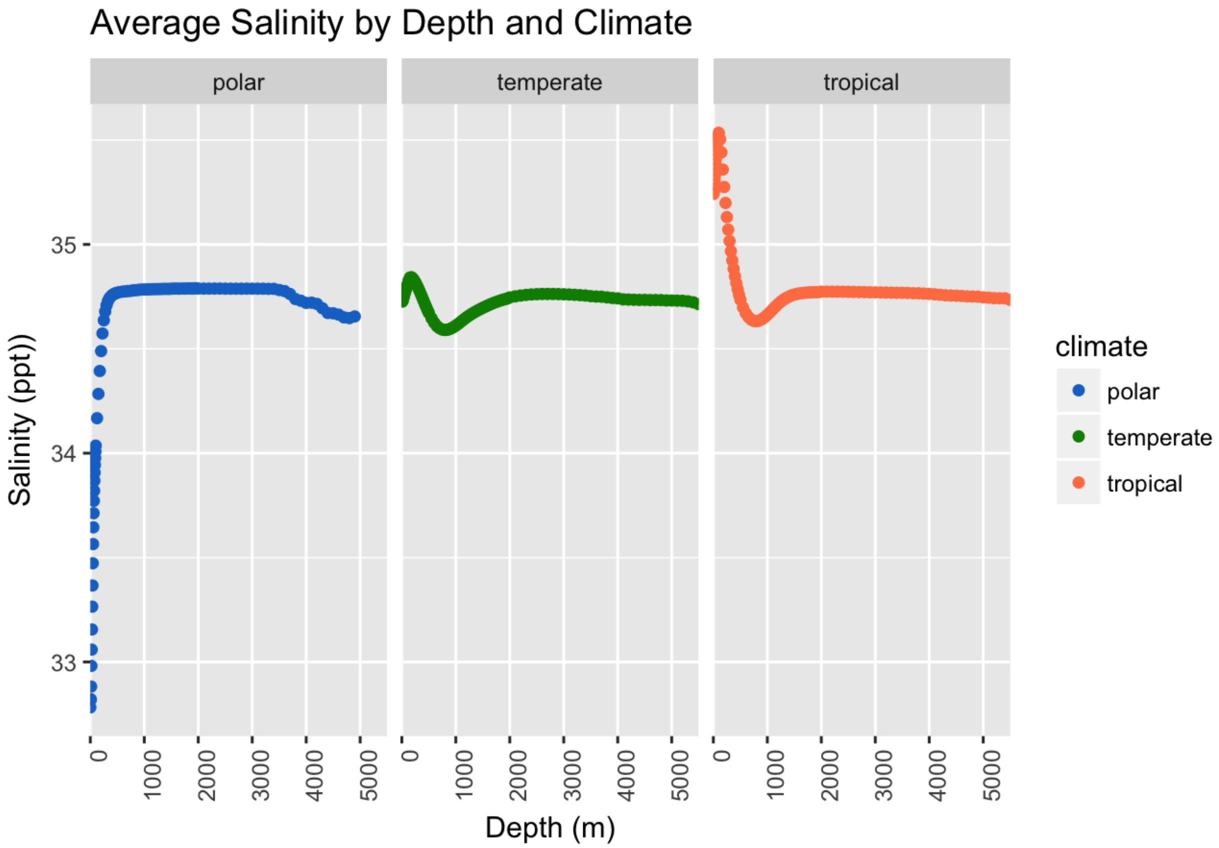
(ii) Look for trends in the data using the average temperature/salinity for each depth.

```
#calculate the average temperature for each depth in a given climate zone
temp_analysis_mean <- temp_analysis %>%
  group_by(climate, depth) %>% summarize(avg_temp=mean(temperature, na.rm=T))
#you know the drill...summarize and inspect the data with summary(temp_analysis_mean)

# same for salinity
#calculate the average salinity for each depth in a given climate zone
sal_analysis_mean <- sal_analysis %>%
  group_by(climate, depth) %>% summarize(avg_sal=mean(salinity, na.rm=T))
#you know the drill...summarize and inspect the data with summary(sal_analysis_mean)

#re-plot the scatterplots from (2i) but this time, use the new data frame with the #avg_temp/avg_sal as
temp_mean_scatterplot <- ggplot(temp_analysis_mean, aes(depth, avg_temp, color = climate)) +
  geom_point() +
  scale_color_manual(values = c("dodgerblue3", "green4", "coral")) +
  ylab("Temperature (C)") +
  #the x axis gets really ugly with all of these depth values, so we'll select a few landmarks
```

```r
  scale_x_discrete(name = "Depth (m)", limits = (0:5500), breaks = seq(0, 5500, 1000)) +
  #and rotate the axis text
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  facet_wrap(~climate) +
  ggtitle("Average Temperature by Depth and Climate")
#visualize the scatterplot
temp_mean_scatterplot
```



```r
#same for salinity
sal_mean_scatterplot <- ggplot(sal_analysis_mean, aes(depth, avg_sal, color = climate)) +
  geom_point() +
  scale_color_manual(values = c("dodgerblue3", "green4", "coral")) +
  ylab("Salinity (ppt))") +
  scale_x_discrete(name = "Depth (m)", limits = (0:5500), breaks = seq(0, 5500, 1000)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  facet_wrap(~climate) +
  ggtitle("Average Salinity by Depth and Climate")
#visualize the scatterplot
sal_mean_scatterplot
```



Take a moment to visually assess these new plots.
11. Do your average value plots reflect the trends we observed across the entire dataset?
12. What are some benefits of summarizing data with average values?
13. What do we lose when plotting averages? How might that affect our interpretation of the data?

Concluding Remarks

We hope that we have helped you to understand the power of R as a statistical tool, and the utility of RStudio as a working environment for using R to run analyses and make awesome graphics! If you'd like more practice with RStudio, you can make just a few modifications to the code we've written here and use it to analyze the other physical oceanography databases available through the World Ocean Atlas.

Love Coding in R? Want more? Here are some awesome resources:

- **R for Data Science**
- A comprehensive instructional text for beginners
- Includes explanations and examples with annotated code
- Available at http://r4ds.had.co.nz/

- **R Cheat Sheets**
- Free PDFs with helpful tips for commonly-used functions
- Available at https://www.rstudio.com/resources/cheatsheets/

- **Stack Overflow**

- An online community with thousands of users who help troubleshoot each other's code
- https://stackoverflow.com/

- **GOOGLE!**
- Just about every problem you'll encounter has been solved and talked about online. Google is my most valuable trouble-shooting resource.