**R AND RSTUDIO INSTALLATION AND TUTORIAL**

TEMPLATE: Please copy content/questions to your own document.

Go to File → Make a Copy → Save to your own Drive.

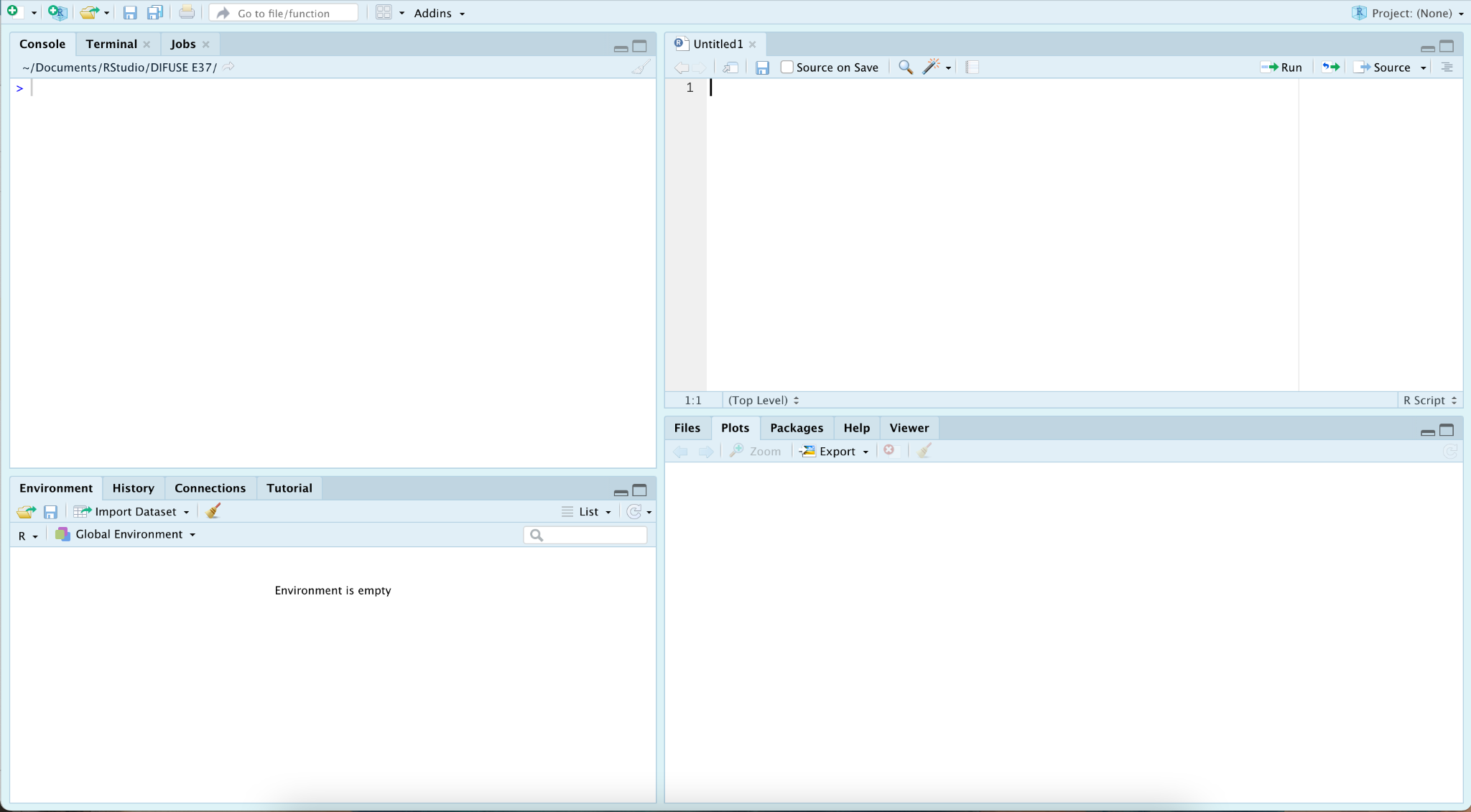
Your name:

**INSTALLATION – BEFORE COMING TO CLASS**

1. (Slide 3 - *Setup RStudio*)  
   Install R software.
   1. Follow the instructions for download on this website: <https://www.r-project.org/>  
        
      The easiest way to install R is from a CRAN (Comprehensive R Archive Network) repository *near your current location*.
   2. After successfully installing, an R Console might open. You may close this.
2. (Slide 3 - *Setup RStudio*)  
   Install RStudio IDE – an environment for you to run, track, and output your R scripts. It includes a console for direct code execution, tools for plotting and keeping track of your variable, and more.
   1. You can download the RStudio IDE Desktop here: <https://rstudio.com/products/rstudio/>.
   2. You might want to watch their explanatory video at RStudio IDE Overview: <https://www.rstudio.com/products/rstudio/?wvideo=520zbd3tij>.

**EXPLORING R – IN CLASS**

1. (Slide 4 - *RStudio Startup*)  
   Start RStudio. You will see something like this (panels might be ordered differently):  
     
   *File Window* – if you make a script, it will live in this corner  
   *Console* – run commands directly here  
   *Environment* – lists current objects (starts empty)  
   *Viewer* – view plots, files, packages, and help menus

If you have text that pops up in your console, environment, or viewer windows, you can use the “broomstick” icon there to clear it out  
  


1. (Slide 5 - *Helpful Commands #1*)  
   Here are some useful commands for learning the capabilities and requirements of a function:  
     
   help.search("term") – where “term” is a word that might appear in the help files  
   ? term– similar to above; for example: ? hist  
   help() – where you insert an R command in the (), like help(t.test)  
   summary(variable) – provides a summary of a particular variable  
   # – creates a comment line (#### after a comment makes that comment a “header”)  
   1. Test your console by typing: print("Hello World!") in the console and pressing Enter.
2. (Slide 6 - *Writing and Executing Code*)  
   We will be saving and executing code in an R script file – within the *File Window*.
   1. Open a new R script file:  
        
      Go to File → New File → R Script, which will open a blank sheet, or click on the ‘new file’ icon () at the top left corner of the RStudio window.
   2. Save your file as “R\_intro\_<first name>\_<last name>.R” in a spot on your computer that you will remember.  
        
      For example, if my name is Andrew Roznere, then I will save the file as R\_intro\_Andrew\_Roznere.R
   3. To execute a line of code in your R script file:  
        
      Have your text cursor indicator on the line of code you want to execute and press **Cmd + Enter**.
   4. *Option for running the whole script:* **Cmd + Shift + Enter**
3. (Slide 7 - *Load Data*)  
   In your R script file, load R’s [air quality dataset](https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/airquality.html) via R’s saved dataset variable name:
   1. Type in the R script file:  
        
      air\_quality\_data\_full <- airquality  
        
      and press **Cmd + Enter** to execute the line.
   2. Note how the operation <- assigns the result of an operation (right side) to an object (left side). This is different from other coding languages that use “=”.
   3. *Problems with loading airquality?* Download the file R\_intro\_dataset.csv from Canvas and load it into RStudio.  
        
      air\_quality\_data\_full <- read.csv(file.choose())  
        
      A window should pop up – navigate to the file *R\_intro\_dataset.csv* and select it.
4. (Slide 8 - *Set Working Directory*)  
   We need to set our “working directory '' for R to know where to look and save files – the file path that R will navigate to when asked to read a csv file or to export one.
   1. Run the command: getwd() in your command line to see what your current working directory is. What folder is R currently looking at (the last folder in the filepath)?
   2. In your Documents director (using Finder or Files Viewer, not RStudio), create a folder called “RStudio”. Within the “RStudio” folder, create another folder called “ENGS 37 Air Quality”.  
        
      In RStudio Console, run the command setwd() to set your working directory to end at the “ENGS 37 Air Quality” folder.
      1. On a macOS, your path command might look like:  
           
         setwd("/Users/Andrew/Documents/RStudio/ENGS 37 Air Quality")
      2. Within setwd(), your folders are separated by forward slashes, and the order starts at a top level folder.
5. (Slide 9 – *More About airquality*)Info on dataset: Today’s dataset – collected by the New York State Department of Conservation and the National Weather Service – contains daily readings (05/01/1973 – 09/30/1973) of the following air quality measurements:

| **Column #** | **Variable** | **Type** | **Name** | **Unit** |
| --- | --- | --- | --- | --- |
| 1 | Ozone | integer | Ozone | ppb |
| 2 | Solar.R | integer | Solar R | lang |
| 3 | Wind | numeric | Wind | mph |
| 4 | Temp | integer | Temperature | degrees F |
| 5 | Month | integer | Month | 1–12 |
| 6 | Day | integer | Day of Month | 1–31 |

\*Numeric value corresponds to a floating point value.  
\*\*For more detailed information on dataset, refer to: [air quality dataset](https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/airquality.html).

1. (Slide 10 - *Viewing Data*)  
   Learn a little about the data with these useful commands. Run these one at a time and answer the corresponding questions:
   1. air\_quality\_data\_full  
        
      How many entries are there in total? Find line 43 and paste it below, what day and month is it? What is the wind speed on that day (with units)?
   2. colnames(air\_quality\_data\_full)  
        
      Paste your output below. What character surrounds the column names?
   3. str(air\_quality\_data\_full)  
        
      Paste your output below. There is an additional column here that shows the words “int” or “num” for each variable, what do “int” and “num” mean?
   4. head(air\_quality\_data\_full).  
        
      How many rows does the head function show? Paste your output below.
   5. length(air\_quality\_data\_full$Ozone) – $ notation denotes columns.  
        
      Paste your output, what does this number correspond to?
2. (Slide 11 - *Cleaning Data*)  
   We will only be using 2 variables (Ozone and Month). Thus, we will take a subset of this dataset:  
     
   Execute this line, selecting only Ozone and Month:  
     
   air\_quality\_data <- subset(air\_quality\_data\_full, select=c(Ozone, Month))  
   1. *Fun tip:* One can also choose the columns by what you do not want included:  
        
      air\_quality\_data <- subset(air\_quality\_data\_full, select=-c(Wind, Solar.R, Day, Temp)
3. (Slide 12 - *Verifying Data*) Make sure you have **2** variables/columns (**Ozone** and **Month**) and **153** lines of data. Sometimes a line gets lost due to programming or file formatting errors; always double check that your data are correct before proceeding.  
     
   In the *Environment* panel, double-click on air\_quality\_data – your new subset. This will open up the data sheet in the Viewer panel. In addition, what command(s) that you already learned can you use to show that you only have Ozone and Month now?
   1. *Hint*: Look at Step 5 (Slide 9)
4. (Slide 14 - *Plot 1: Histogram*)  
   We will use the function hist to create a histogram on our data. Let’s check what hist can do for us. Execute this line: help(hist)  
     
   hist function can take many different arguments, but at least one is required: *x*. The other arguments that you should know are *breaks*, *main*, *xlab*, *ylab*, *xlim*, and *ylim*. From the help function, summarize what these arguments represent and best use for.  
     
   x:   
   breaks:   
   main, xlab, ylab:

xlim, ylim:

1. (Slide 14 - *Plot 1: Histogram*)  
   Create a histogram for ALL of the months data on ozone, using the default options:  
     
   hist(air\_quality\_data$Ozone)  
     
   Your plot will appear in the Plots panel. Paste your plot → oh! check the next step to see how to do this. How many bars are there by default?
2. (Slide 15 - *Saving Plots*)How to save plots? You will need to know how to save your figures and graphs either as an Image or as a PDF. To save the figure/graph/plot, in the *Plots* window, click Export, then choose one of the options:  
    - “Save as Image…”, or   
    - “Save as PDF…”, or  
    - “Copy to Clipboard…”.
3. (Slide 16-18)  
   We will now add more parameters/options to the hist function:
   1. (Slide 16 - *Plot 1: Histogram (con’t)*)  
      This code includes an x-axis label, but deletes the graph’s title because it won’t be needed once you write a good caption.  
        
      hist(air\_quality\_data$Ozone, xlab="ppb", main="")  
      hist(air\_quality\_data$Ozone, xlab="ug/L", main="")  
        
      Run both commands and paste your plots. What argument deleted the title?
   2. (Slide 16 - *Plot 1: Histogram (con’t)*)  
      To make the notation prettier you can correspond the expression (value) to a variable name, and use it later.  
        
      hist(air\_quality\_data$Ozone, xlab=expression(paste(mu,"g/",L)), main="")  
        
      Run the command and paste your plot. The expression command creates an expression object and assigns it to xlab, so what does the paste function do?
   3. (Slide 17- *Plot 1: Histogram (con’t)*)  
      Change the number of bins in a histogram. R determines the number of bins (or breaks) to put in a histogram that makes it look "prettiest" using something called the Sturges method. Alternatively, you can tell R how many bins you would like it to create:  
        
      hist(air\_quality\_data$Ozone, xlab="ppb", main="", breaks=22)  
      hist(air\_quality\_data$Ozone, xlab="ppb", main="", breaks=5)  
        
      Choose a number of bin/breaks that you think is most appropriate, paste that plot, and explain why that number is most appropriate.
   4. (Slide 18 - *Plot 1: Histogram (con’t)*)  
      Notice that you can change the look of the histogram by changing the range of values shown on the x-axis:  
        
      hist(air\_quality\_data$Ozone, xlab="ppb", main="", breaks=22, xlim=c(0, 200))  
        
      Choose an xlim that you think is best, explain why, and paste your plot.
   5. (Slide 18 - *Plot 1: Histogram (con’t)*)  
      Play around with the other options (and more that you find in the help documentation for hist()): create a plot that you think is best, paste it below, and describe the distribution you see (i.e., shape, center, spread)**.**Hint: hist() has lots of options, so look at its help page for details.
4. (Slide 19-20 - *Refining and Segmenting Data*)  
   Of course, we do not really want that histogram for the whole dataset – we really want to look at the distribution of each month separately.  
     
   Let’s create a histogram on ozone for each month. This can be done most easily in a brute-force fashion, by creating subsets of the data for example by pulling out the ~30 rows for each month at a time:  
     
   hist(air\_quality\_data$Ozone[1:31], xlab="ppb", main="May")  
   hist(air\_quality\_data$Ozone[32:61], xlab="ppb", main="June")  
   hist(air\_quality\_data$Ozone[62:92], xlab="ppb", main="July")  
   hist(air\_quality\_data$Ozone[93:123], xlab="ppb", main="August")  
   hist(air\_quality\_data$Ozone[124:153], xlab="ppb", main="September")  
     
   The above ranges is a brute-force method, so instead let’s use logical selection, e.g.:  
     
   hist(air\_quality\_data$Ozone[air\_quality\_data$Month==5], xlab="ppb", main="May")  
   1. What goes inside the square brackets after air\_quality\_data$Ozone (looking for a general answer)? Think about what the code in the brackets does.
   2. Run air\_quality\_data$Month==5 alone, what does this return (describe the output)?
   3. Now, explain how the logical selection method works, and why this might be better than a brute-force approach.
   4. Paste a plot for each month and describe the distribution (i.e., shape, spread, outliers) of each plot.
5. (Slide 21 - *R Libraries*)  
   Next, let's try using a package to accomplish the same thing using a formula. First we will need to install the package *lattice*. But what is a library?  
     
   *“R packages are a collection of R functions, compiled code and sample data. They are stored under a directory called "library" in the R environment. By default, R installs a set of packages during installation. More packages are added later, when they are needed for some specific purpose. When we start the R console, only the default packages are available by default. Other packages which are already installed have to be loaded explicitly to be used by the R program that is going to use them.” [source:* [*https://www.tutorialspoint.com/r/r\_packages.htm*](https://www.tutorialspoint.com/r/r_packages.htm)*]*  
     
   (Slide 22 - *Installing and Loading lattice*)  
   Execute the following lines:  
     
   install.packages("lattice") # Install the library **library**(lattice) # Call/include the library  
     
   *IMPORTANT:* After installing lattice, you will not have to install it again – but you will need to call it in your script via the **library** function to use its commands.
6. (Slide 23 - *Segmenting with lattice*)  
   Describe at least one similarity and one difference between this function and the previous hist function. You may use the following function call to help out. ? histogram
7. (Slide 23 - *Segmenting with lattice*)  
   The following arguments will be needed for histogram plots: *x*, *data*, and *type*. From the previous step’s help type function, summarize what these arguments represent and used for:  
     
   *x*:   
   *data*:   
   *type*:
8. (Slide 23 - *Segmenting with lattice*)  
   Execute the following lines *one at a time*:  
     
   histogram(~ Ozone | Month, data=air\_quality\_data)  
   histogram(~ Ozone | Month, data=air\_quality\_data, type="count")  
     
   Notice the difference between the two histograms? Paste your plots and describe the differences.
9. (Slide 24 - Custom Graphics)  
   Histograms are great for quickly looking at the data, but what if you want to change the look of the graphic before putting it in your write-up? Let's build up a graphic from scratch. R is awesome for producing publication-ready graphics. As a result, the number of graphical parameters you can control is overwhelming.  
     
   We will be using the par function (feel free to check it out: ? par ).  
     
   Now we will set some graphical parameters ourselves. Build the figure we want:  
     
   par(mfrow=c(5,1), mar=c(4,2,2,2)+0.01, oma=c(2,2,0,0))  
   hist(air\_quality\_data$Ozone[1:31], xlab="", main="May", xlim=c(0, 200))  
   hist(air\_quality\_data$Ozone[32:61], xlab="", main="June", xlim=c(0, 200))  
   hist(air\_quality\_data$Ozone[62:92], xlab="", main="July", xlim=c(0, 200))  
   hist(air\_quality\_data$Ozone[93:123], xlab="", main="August", xlim=c(0, 200))  
   hist(air\_quality\_data$Ozone[124:153], xlab="", main="September", xlim=c(0, 200))  
   mtext("ppb", 1, outer=TRUE, cex=1.5)  
   mtext("Frequency", 2, outer=TRUE, cex=1.5)  
     
   Paste your final plot and explain at least two pros and two cons of this plot compared to the earlier ones.
10. (Slides 25-26 - *Box Plots*)  
    Create box plots for the whole dataset and for each month.  
      
    boxplot(air\_quality\_data$Ozone, xlab="ppb", main="")  
    boxplot(air\_quality\_data$Ozone ~ air\_quality\_data$Month, xlab="ppb", ylab="Month", main="")  
      
    *Trouble?* If you are having trouble plotting the boxplot due to the previous par window formatting, run this before the boxplots:  
    par(mfrow=c(1,1), mar=c(4,2,2,2)+0.01, oma=c(2,2,0,0))  
    1. Paste your plots. Describe what you see in the box plots, and how this is similar or different from the histograms.
    2. Which months are captured well by the histogram? By the boxplot? Which are captured poorly by either or both of these months? Is there perhaps a third option worth pursuing?
    3. *Fun tip:* A neat trick for boxplots:  
         
       points(air\_quality\_data$Month, air\_quality\_data$Ozone, col="red", cex=0.5)
11. (Slide 27 - *Numeric Summaries*)  
    Numeric summaries – mean & SD vs. 5-number summary – can be important for discussing the distribution of each variable, if appropriate.
    1. To calculate all of the numeric summaries by month:  
         
       by(air\_quality\_data$Ozone, air\_quality\_data$Month, summary)

Write the output in the table below. Also, highlight the row (month) that has the greatest potential error. Hint: what do the NAs in the data mean? Look at your data in a way that shows you NAs to learn more about this.

| **Month** | **Min.** | **1st Qu.** | **Median** | **Mean** | **3rd Qu.** | **Max.** | **NA’s** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

1. (Slide 28 - *Cleaning 2: Removing NAs*)  
   At times, calls/functions might not work properly with your dataset as it includes NA’s. To clean or remove NA’s from datasets:  
     
   nas <- is.na(air\_quality\_data) # Returns a logical array of TRUEs where NAs are in the original set. New array is same size as data  
     
   air\_quality\_data\_cleaned <- air\_quality\_data[complete.cases(air\_quality\_data), ] # Note: we need the comma and space in the brackets to select all rows and columns to check  
     
   Check this cleaned up dataset – what is the number of the last row? Are there that many rows?
2. (Slide 29 - *More Plots: ggplot and tidyverse*)  
   Tidyverse package and examples.  
     
   To install and load:  
     
   install.packages("tidyverse")  
   install.packages("ggplot2")  
   **library**(tidyverse)  
   **library**(ggplot2)  
     
   Go through each example, paste your plot, and in one sentence describe the possible purpose of designing such a plot.
   1. Histograms
      1. (Slide 30) Ex. 1:  
         qplot(air\_quality\_data$Ozone,  
          geom = "histogram", # graph type  
          binwidth = 3, # bin size  
          main = "Histogram of Ozone", # figure title  
          xlab = "ppb", # label for x axis  
          fill = I("green"), # fill color  
          col = I("black"), # outline color  
          alpha = I(.5), # fill transparency (between 0-see through and 1-opaque)  
          xlim = c(0, 200)) # x axis range
      2. (Slide 31) Ex. 2:  
         ggplot(data = air\_quality\_data, aes(air\_quality\_data$Ozone)) +  
          geom\_histogram(breaks = seq(0, 200, by = 3),  
          col = "black",  
          aes(fill = ..count..)) +  
          scale\_fill\_gradient("Count", low = "green", high = "red") +  
          labs(title = "Histogram of Ozone", x = "ppb", y = "Count")
      3. (Slide 32) Ex. 3:  
         multi <- air\_quality\_data %>% #tidyverse's "piping" functionality with "%>%"  
          ggplot(aes(x = Ozone, color = Month, fill = Month)) +  
          geom\_histogram(alpha = 0.5, binwidth = 3) +  
          theme(legend.position = "none") +  
          xlab("") +  
          ylab("ppb") +  
          facet\_wrap(~Month)  
         multi
   2. Boxplots
      1. (Slide 33) Ex. 1:  
         outliboxplot <- air\_quality\_data %>%  
          ggplot(aes(x = Month, y = Ozone, fill = Month)) +  
          geom\_boxplot(outlier.color = "black", outlier.shape = 8, outlier.size = 2) + # outlier boxplot  
          stat\_summary(fun = mean, geom = "point", shape = 23, size = 4) # add mean diamond  
         outliboxplot
      2. (Slide 34) Ex. 2:  
         dotnboxplot <- air\_quality\_data %>%  
          ggplot(aes(x = Month, y = Ozone)) +   
          geom\_boxplot() +   
          geom\_jitter(color = "black", size = 0.9, alpha = 0.8) +  
          theme(panel.border = element\_blank(),  
          panel.background = element\_blank(),  
          panel.grid.major = element\_blank(),  
          panel.grid.minor = element\_blank(),  
          axis.line = element\_line(color = "black"))  
         dotnboxplot
   3. (Side 35-36) Numeric Summaries  
        
      ? summarise  
        
      The following calls/functions will not perform correctly if there are NA’s in your dataset. Please remove them beforehand:  
        
      air\_quality\_data\_cleaned <- air\_quality\_data[complete.cases(air\_quality\_data), ]
      1. (Slide 35) Ex. 1:  
         air\_quality\_data\_cleaned %>%  
          group\_by(Month) %>%  
          summarise(count = n(),  
          df = count-1,  
          mean = mean(Ozone),  
          sd = sd(Ozone))
      2. (Slide 36) Ex. 2:  
         air\_quality\_data\_cleaned %>%  
          group\_by(Month) %>%  
          summarise(count = n(),  
          df = count-1,  
          min = min(Ozone),  
          Q1 = quantile(Ozone, 0.25),  
          M = median(Ozone),  
          Q3 = quantile(Ozone, 0.75),  
          max = max(Ozone))
3. (Slide 37 - *Final Plot: Your Choice*)  
   Finally, create a figure (possibly with multiple panels) comparing the distributions, this time, of **Wind** or **Temperature**. The exact nature of the figure is up to you; if you have multiple panels, just be sure to label the separate panels. You should also be sure to include descriptions of the distribution (i.e., shape, spread, outliers) of each month or note any observations of the data that you had.  
     
   Ask your TA for help if needed.

**R RESOURCES**

* [R Resources maintained by UCLA](https://stats.oarc.ucla.edu/r/)
* [R Bootcamp](https://www.jaredknowles.com/r-bootcamp/)
* [Quick-R](https://www.statmethods.net/) (*reference guide*)
* [RStudio Cheatsheets](https://www.rstudio.com/resources/cheatsheets/)
* [R Graph Gallery](https://www.r-graph-gallery.com/index.html) *(Tidyverse and ggplot2 help*)
* [GGPlot Color Tricks](https://www.datanovia.com/en/blog/ggplot-colors-best-tricks-you-will-love/) (*for color-blind-friendly graph-making*)
* [Stack Overflow](https://stackoverflow.com/questions/tagged/r) (*public forum for script-writing questions*)
* [R for Data Science](https://r4ds.had.co.nz/) (*by Hadley Wickham*)
* [Kickstarter guide](https://paulvanderlaken.com/2017/10/18/learn-r/) (*for new R users*)
* [Curated list of packages and tools](https://github.com/qinwf/awesome-R/blob/master/README.md) (*on Github*)