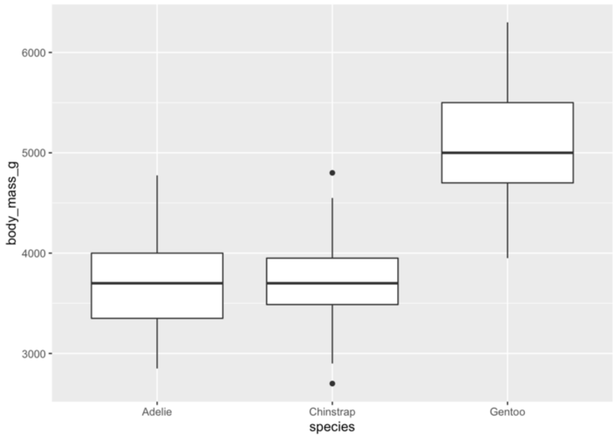
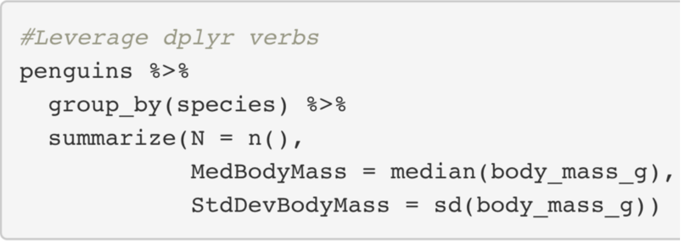
STAT 380 –Data Wrangling Part 1 (Lecture 2)

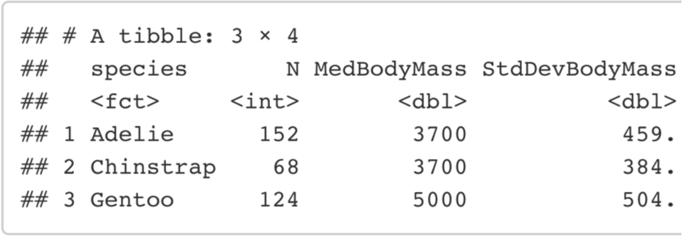
These notes largely follow from Chapters 4.1 of *Modern Data Science with R*.

EXAMPLE 1: (Builds on Example 8 from Lecture 1.) In addition to the boxplots generated in the last example (reproduced for your convenience below), we would support the boxplots with summary statistics. i) Based on the boxplots, what have you learned about the penguins? ii) For each species, find the number of penguins, median of the body masses, and the standard deviation of the body masses. Modify the given code to obtain the summary statistics listed below.





```
#Leverage dplyr verbs
penguins %>%
  group_by(species) %>%
  summarize(N = n(),
            MedBodyMass = median(body_mass_g),
            StdDevBodyMass = sd(body_mass_g))
```



```
## # A tibble: 3 × 4
##   species       N MedBodyMass StdDevBodyMass
##   <fct>     <int>       <dbl>          <dbl>
## 1 Adelie      152        3700           459.
## 2 Chinstrap    68        3700           384.
## 3 Gentoo      124        5000           504.
```

EXAMPLE 2: In the file STAT380_L2.Rmd, two methods are presented for finding the summary statistics in Example 1. Issues with NA's aside, explain why Method 2 is preferred to Method 1.

Definition: Data wrangling is the process of reforming, summarizing, and combining data to make it more suitable for a given purpose.

NOTE: The *tidyverse* is a collection of R packages that are commonly used for data wrangling and data visualization. The dplyr package, which is part of the tidyverse, contains a number of important data wrangling functions. In particular, we regularly use the following “dplyr verbs”:

- filter() - select a subset of rows (observations, cases), often according to their values
- arrange() - reorder/sort the rows
- select() - select a subset of columns (variables, features, attributes)
- mutate() – add (e.g., create new variables) or modify existing columns
- summarize() – aggregate data across rows
  - often paired with group_by() to find the summary statistics for each level of a categorical variable

NOTE: The *Data transformation with dplyr cheatsheet* at https://posit.co/resources/cheatsheets/ is a great resource.

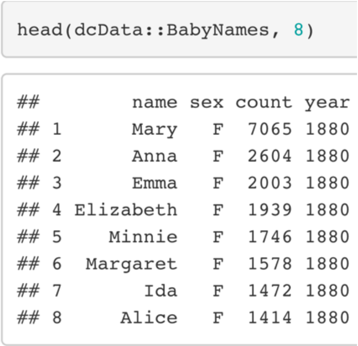
NOTE: These functions provide the ***verbs*** for a language of data manipulation. All verbs work similarly:

1. The first argument is a data frame.
2. The subsequent arguments describe what to do with the data frame, using the variable (column) names (without quotes).
3. The result is a new data frame.
4. dplyr functions never modify their inputs (if you want to save the result, you’ll need to use the assignment operator, <-)

Tidyverse Command Chains

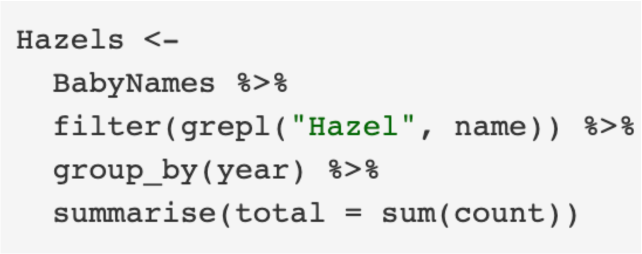
NOTE: It is common practice to combine multiple dplyr verbs into a single command chain.

- Each link in the chain is a “data verb” or “data move” with its arguments
  - The very first link is typically a data table/data frame.
  - Links are connected by the pipe: %>%
- Often, but not always, you will store the result of the chain in a named object
  - This is done with the assignment operator, <-
- New line for each link
- Note that %>% is at the end of each line. **Except**
  - Hazels <- is an assignment
  - Last line has no %>% (otherwise R thinks there’s more)



```
head(dcData::BabyNames, 8)
```

```
##          name sex count year
## 1        Mary   F  7065 1880
## 2        Anna   F  2604 1880
## 3        Emma   F  2003 1880
## 4 Elizabeth   F  1939 1880
## 5      Minnie   F  1746 1880
## 6    Margaret   F  1578 1880
## 7         Ida   F  1472 1880
## 8       Alice   F  1414 1880
```



```
Hazels <-
  BabyNames %>%
  filter(grepl("Hazel", name)) %>%
  group_by(year) %>%
  summarise(total = sum(count))
```

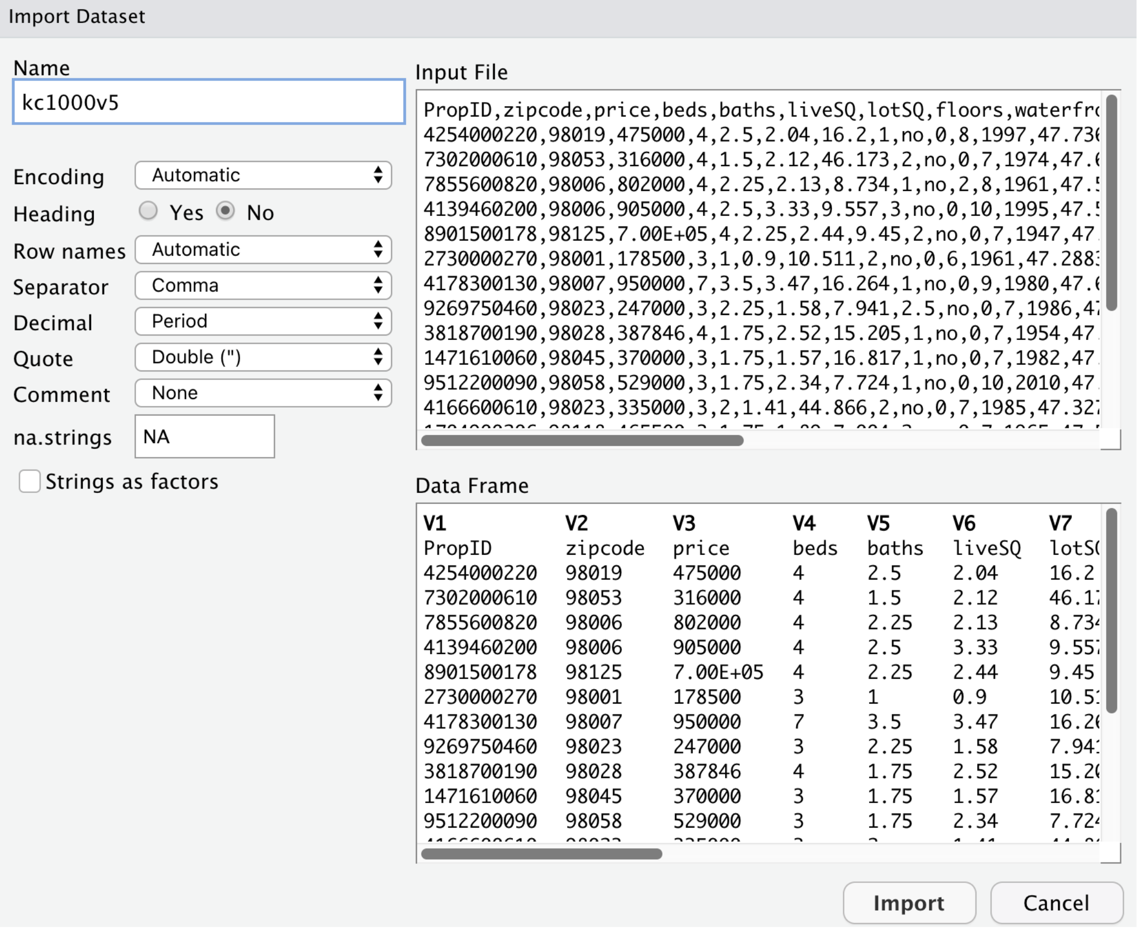
EXAMPLE 3: Many datasets come from packages in R; however, it is also common that our data are from an external source. The purpose of this example is to practice reading in data from different sources. CAUTION: There are many ways to read in data. I am presenting one method.

NOTE: When downloading datasets from Canvas, use Chrome. For Mac users, avoid Safari and avoid opening .csv files in Numbers.

a. We want to read the file kc1000v5.csv that is found on Canvas into R. What does .csv represent?

b. If the file is NOT downloaded to your working directory (use getwd() to find the current working directory), then you will have to specify a path to the file. One way to handle this is to use the Import Dataset interface found in the Environment tab. Here are the steps:

a) In the Environment window (upper right) click on Import Dataset
b) select From Text (base)
c) Browse to where you stored the file kc1000v5.csv (likely in Downloads) and select Open
d) Edit the name (upper left) if desired
e) You will see a preview in the Data Frame portion. If it looks reasonable, select IMPORT
f) PRO TIP: This sequence of actions will cause R to create some code in the Console. Copy this code into your Markdown document.



Name: kc1000v5

Encoding: Automatic
Heading: ○ Yes ● No
Row names: Automatic
Separator: Comma
Decimal: Period
Quote: Double (")
Comment: None
na.strings: NA
□ Strings as factors

Input File

```
PropID,zipcode,price,beds,baths,liveSQ,lotSQ,floors,waterfro
4254000220,98019,475000,4,2.5,2.04,16.2,1,no,0,8,1997,47.73
7302000610,98053,316000,4,1.5,2.12,46.173,2,no,0,7,1974,47.6
7855600820,98006,802000,4,2.25,2.13,8.734,1,no,2,8,1961,47.5
4139460200,98006,905000,4,2.5,3.33,9.557,3,no,0,10,1995,47.5
8901500178,98125,7.00E+05,4,2.25,2.44,9.45,2,no,0,7,1947,47
2730000270,98001,178500,3,1,0.9,10.511,2,no,0,6,1961,47.288
4178300130,98007,950000,7,3.5,3.47,16.264,1,no,0,9,1980,47.6
9269750460,98023,247000,3,2.25,1.58,7.941,2.5,no,0,7,1986,47
3818700190,98028,387846,4,1.75,2.52,15.205,1,no,0,7,1954,47
1471610060,98045,370000,3,1.75,1.57,16.817,1,no,0,7,1982,47
9512200090,98058,529000,3,1.75,2.34,7.724,1,no,0,10,2010,47
4166600610,98023,335000,3,2,1.41,44.866,2,no,0,7,1985,47.32
```

Data Frame

V1	V2	V3	V4	V5	V6	V7
PropID	zipcode	price	beds	baths	liveSQ	lotS
4254000220	98019	475000	4	2.5	2.04	16.2
7302000610	98053	316000	4	1.5	2.12	46.1
7855600820	98006	802000	4	2.25	2.13	8.73
4139460200	98006	905000	4	2.5	3.33	9.55
8901500178	98125	7.00E+05	4	2.25	2.44	9.45
2730000270	98001	178500	3	1	0.9	10.5
4178300130	98007	950000	7	3.5	3.47	16.2
9269750460	98023	247000	3	2.25	1.58	7.94
3818700190	98028	387846	4	1.75	2.52	15.2
1471610060	98045	370000	3	1.75	1.57	16.8
9512200090	98058	529000	3	1.75	2.34	7.72

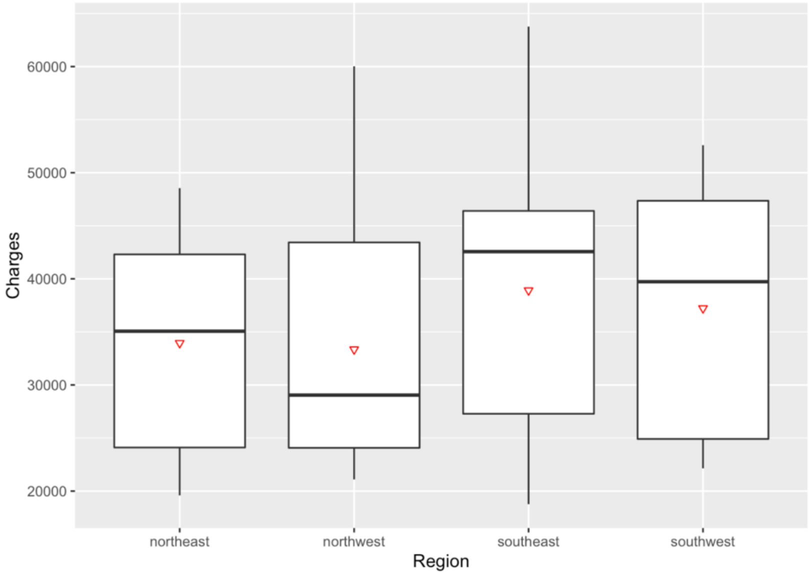
c. Download the file Insurance.txt from Canvas. Read the dataset into R and name the resulting dataset Ins. How many observations and how many variables are included in the dataset?

NOTE: We do not ***always*** use read.csv(). Other useful functions include:

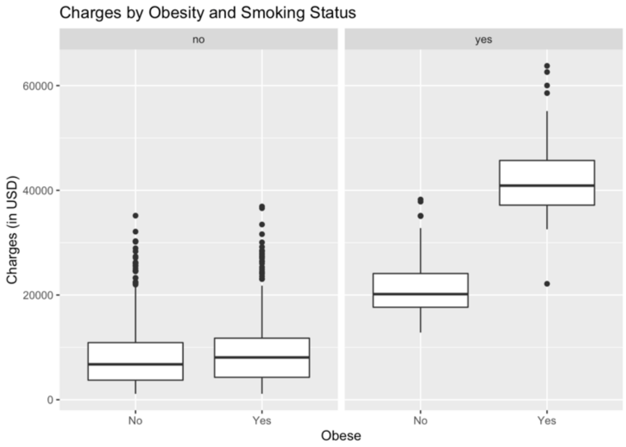
EXAMPLE 4: (dplyr practice) Using Insurance.txt,

a. find the mean and standard deviation for medical charges for smokers over 40 broken down by region.

b. Create side-by-side boxplots showing the medical charges for smokers over 40 in each region. Add red triangles (point down) to show the mean.



c. Recreate the plot shown below. What have you learned about the relationship between the variables? NOTE: Obese takes on a value of “Yes” if the person’s BMI is 30.0 or higher.

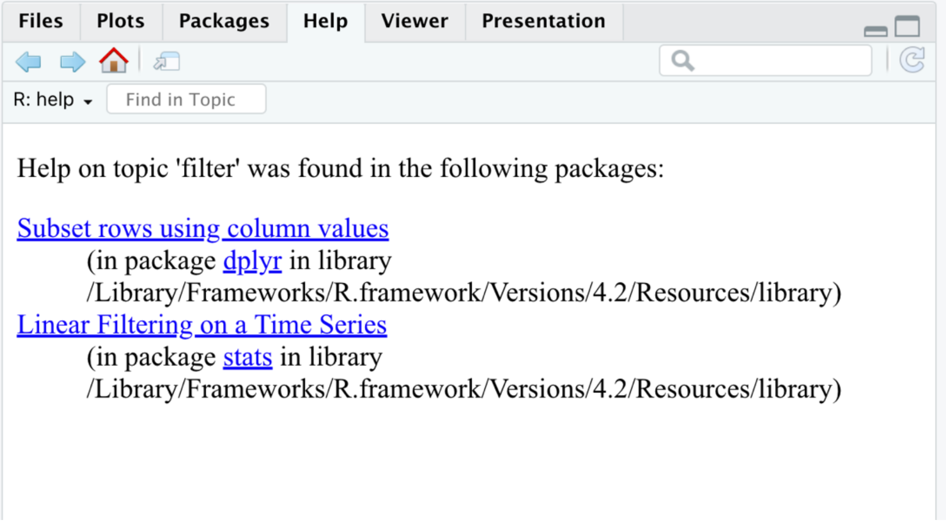


EXAMPLE 5: (A note about masking.) It is important to understand the concept of ‘masked’ objects in R. Restart your RStudio session.

a. In the Console, run the command “library(dplyr)”. You will see the following message in the Console.

```
> library(dplyr)

Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

b. Look at the help file documentation for filter by entering ?filter (or help(“filter”)) in the Console. Notice that there are two objects (specifically functions) named ‘filter’. One is from the stats library (base R) and the other is from the dplyr library. If we call the filter function which one will R use?



Help on topic 'filter' was found in the following packages:

Subset rows using column values
(in package dplyr in library
/Library/Frameworks/R.framework/Versions/4.2/Resources/library)
Linear Filtering on a Time Series
(in package stats in library
/Library/Frameworks/R.framework/Versions/4.2/Resources/library)

c. Consider the following. Explain the cause of the error and fix it.

```r
library(palmerpenguins)
library(dplyr)
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

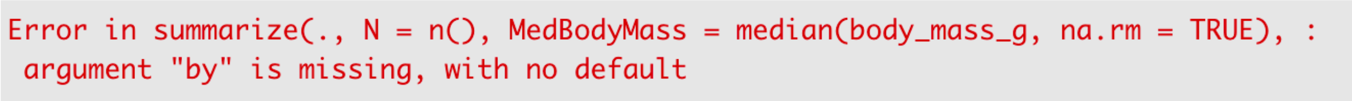
```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
penguins %>%
  group_by(species) %>%
  summarize(N = n(),
            MedBodyMass = median(body_mass_g, na.rm = TRUE),
            StdDevBodyMass = sd(body_mass_g, na.rm = TRUE))
```



```
Error in summarize(., N = n(), MedBodyMass = median(body_mass_g, na.rm = TRUE), :
  argument "by" is missing, with no default
```