AMAAS: Using the package "dplyr" for data manipulation

Benjamin Utting (bju23@cam.ac.uk)

September 10, 2018

Analytical Methods in Anthropology and Archaeology (AMAAS)

Lent 2018

The following packages are required for running this document: *tidyverse*, *devtools*

```r
require(bioanth)
  devtools::install_github("geanes/bioanth")
require(devtools)
require(knitr)
require(tidyverse)
```

Introduction

This R Markdown document is intended to accompany a workshop on using the package "dplyr" for manipulating data

Why use dplyr?

- dplyr is more consistent (and easier to learn) than base functions
- dplyr is faster
- dplyr is integrated into the tidyverse

For this demonstration, we will be using Dr. Benjamin Auerbach's osteometric dataset (available at http://web.utk.edu/~auerbach/DATA.htm).

- This dataset includes samples of humans throughout the Holocene
- ~1,500 observations from 59 locations
- Dataset includes measurements from the humerus/radius, femur/tibia, and pelvis

6 Verbs to be introduced today:

1. filter() for selecting rows
2. select() for selecting columns
3. arrange() for arranging data
4. mutate() for adding new columns
5. group_by()/summarize() for summarizing data
6. sample_n()/sample_frac() for taking random samples

Note the intuitive syntax!

A first look at our data

```r
View(goldman)
str(goldman)
summary(goldman)
```

```r
#For variable name definitions
?goldman

#Where are specimens housed?
summary(goldman$Inst)

#How many of each sex is represented?
summary(goldman$Sex)
#985 males and 543 females with 10 indeterminates (5 likely male, 5 likely female)
#Important to note that this sample is sex-biased!

#Where do our specimens come from?
summary(goldman$Location)
#Many different locations from all around the world: however, this column is arranged poorly
  #Varying levels of resolution for each observation
  #Some specimens have three levels of location information (e.g. Aleutian Islands,
  #Alaska, United States) whereas others only have one (e.g. Peru)
  #How do we address this issue?
```

How to separate our "Locations" column into three separate columns

```r
separated_location <- goldman %>%
  separate(Location, into = c("Region", "Subregion", "Country"), sep = ",", fill = "left")
View(separated_location)

#This function returned "character" class data, so we must transform it into "factor"
  #data in order to explore it
separated_location$Region <- as.factor(separated_location$Region)
separated_location$Subregion <- as.factor(separated_location$Subregion)
separated_location$Country <- as.factor(separated_location$Country)

#Explore data using "summary()" function
summary(separated_location$Region)
summary(separated_location$Subregion)
summary(separated_location$Country)
```

1. filter()

Works similarly to the "subset()" function in base R

Pulls out specific observations (rows)

```r
#We are interested in specimens with a left humerus maximum length (LHML) greater than or
  #equal to 300 mm
goldman %>%
  filter(LHML >= 300)
#Narrowed down our sample to 780 specimens

#Which specimens have a LHML >= 300 and ("&" or ",") a right humerus maximum length
  #>= 300?
goldman %>%
  filter(LHML >= 300 & RHML >= 300)
```

```
#OR
goldman %>%
  filter(LHML >= 300, RHML >= 300)
#710 total specimens have both a left humerus >= 300 mm and a right humerus >= 300 mm

#Which specimens have a LHML greater than or equal to 300 or (|) a right
  #humerus maximum length greater to or equal to 300?
goldman %>%
  filter(LHML >= 300 | RHML >= 300)
#938 total specimens have a left humerus >= 300 mm OR a right humerus >= 300 mm

#Which specimens come from Asia?
specimens_asia <- goldman %>%
  filter(Location == "Andaman Islands" | Location == "China" | Location == "Indonesia" |
           Location == "Japan" | Location == "Malaysia" |
           Location == "Philippine Islands" | Location == "China")
#144 total specimens

#Which specimens are housed in New York (AMNH), DC (NMNH), or Paris (MdH)?
goldman %>%
  filter(Inst %in% c("AMNH", "NMNH", "MdH"))
#Total of 656 observations
```

2. select()

Selects columns

Please note that if you get the error message "Error in select () : unused argument ()", R is using the "select()" function from the MASS package.

In order to fix this, you can either unload MASS package or use "dplyr::select()"

```
#Select "Location" from the dataset
goldman %>%
  select(Location)

#Select "Institution, Location, Sex, Left Humerus Maximum Length, and Right
  #Humerus Maximum Length from the dataset
goldman %>%
  select(Inst, Location, Sex, LHML, RHML)

#We are NOT interested in Bi-iliac breadth (for whatever reason)
goldman %>%
  select(-BIB)

#Rearrange data: we want "Location" to appear first
goldman %>%
  select(Location, everything())
```

3. arrange()

Sorts or orders data

```
#Sort by Left Humerus Maximum Length
goldman %>%
  arrange(LHML)

#Sort by LHML with the highest value appearing first
goldman %>%
  arrange(desc(LHML))

#Sort by multiple conditions (in order)
goldman %>%
  arrange(Location, LHML)
#This will give us specimens grouped by location (alphabetically) and then by LHML
```

4. mutate()

Creates new columns

```
#Ratio of left humerus maximum length to right humerus maximum length
goldman %>%
  mutate(hum.ratio = LHML/RHML)

#Multiple columns
goldman %>%
  mutate(hum.ratio = LHML/RHML,
         fem.ratio = LFML/RFML)
```

5. group_by() and summarize()

Summarizes data based on a variable (column)

```
#Difference in left maximum humeral lengths by sex
goldman %>%
  group_by(Sex) %>%
  summarize(mean(na.omit(LHML)))
#Note the use of "na.omit" to remove "NA" values - otherwise, you will get "NA" as a result

#Name new column
goldman %>%
  group_by(Sex) %>%
  summarize(LHML_Mean = mean(na.omit(LHML)))
```

6. sample_n() and sample_frac()

Randomly select a portion of your data (a number or a percentage)

```
#100 observations with sample_n()
n_sample <- goldman %>%
  sample_n(100)
View(n_sample)
#100 samples selected

#10% of observations
```

```
frac_sample <- goldman %>%
  sample_frac(.10)
View(frac_sample)
#154 samples selected
```

Putting it all together with pipes

Pipes are loaded in with the package "magrittr", but I think it loads automatically with library(tidyverse).

When reading code, pronounce "%>%" as "then".

We want to do multiple things to our data in one go, so we use pipes (%>%) to accomplish this.

1. Remove the "BIB" (Bi-iliac Breadth) column using "select()" function
2. We only want to see specimens from the American Museum of Natural History: use "filter()"
3. We want to include a new column in our data (LHML/RHML), so we use the "mutate()" function
4. We want to see our data arranged by Location and LHML (with the highest LHML first)

```
AMNH_data <- goldman %>% #name our dataset "AMNH_data" and specify "goldman" as our source (THEN)
  select(-BIB) %>% #1. Remove bi-iliac breadth (THEN)
  filter(Inst == "AMNH") %>% #2. Specify our "Inst" as "AMNH" (THEN)
  mutate(hum.ratio = LHML/RHML) %>% #3. Add and define our new column (THEN)
  arrange(Location, desc(LHML)) #4. Arrange "AMNH_data" by location first and LHML second

View(AMNH_data)
```