Settlers of Antarctica - Module 1 - Day 4

Keaton Wilson

12/4/2019

Day 4 - 2-dimensional data and introduction to the tidyverse

Learning Objectives

1. Students will practice programming fundamentals like installing and loading packages, using helpfiles and navigating file structure.

2. Students will be able to use the fundamental functions of the tidyverse: select, filter, mutate, summarize and the pipe.

3. Students will be able to compare base R methodology and tidyverse methodology for filtering and indexing dataframes.

4. Students will able to use functions above to gain insight from real-world data.

Tidyverse - what is it?

Differenet programming languages have different syntax... and tidyverse is a package (or set of packages) offered in R that have similar goals and a set of unified syntax designed specifically to work with 2-dimensional data. When I started learning R, none of this was available, but the ecosystem has changed a lot in the last 10 years, and this is the set of tools I use to work with data every day.

Getting the tidyverse (and what's in it) - Group Challenge and discussion

Let's begin by scoping out what is in the tidyverse. Go to www.tidyverse.org and browse the site. In your groups, answer the 3 questions below, and be prepared to report out at the end:
1. What are the core packages in the tidyverse?
2. Give a one sentence summary about what each package does.
3. What is one question you have about the tidyverse/core packages?

Practice with the tidyverse

We're going to use a real data set on climate change from Berkeley. It outlines temperatures in major cities across the world since 1750. We're going to use this data set to learn the tidyverse!

First things first, let's load it up.

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------------------- tidy
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts -------------------------------------------------------------------------- tidyverse_
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# download.file("https://tinyurl.com/vvsykhm")

climate_df = read_csv("../data/GlobalLandTemperaturesByMajorCity.csv")
```

```
## Parsed with column specification:
## cols(
##   dt = col_date(format = ""),
##   AverageTemperature = col_double(),
##   AverageTemperatureUncertainty = col_double(),
##   City = col_character(),
##   Country = col_character(),
##   Latitude = col_character(),
##   Longitude = col_character()
## )
```

```
# call the climate data... this looks a bit different
#climate_df

# It's a tibble! Like a dataframe but better. :) Lots of information to dissect here. Let's look at ano

glimpse(climate_df)
```

```
## Observations: 239,177
## Variables: 7
## $ dt                           <date> 1849-01-01, 1849-02-01, 1849-03-01, ...
## $ AverageTemperature           <dbl> 26.704, 27.434, 28.101, 26.140, 25.42...
## $ AverageTemperatureUncertainty <dbl> 1.435, 1.362, 1.612, 1.387, 1.200, 1....
## $ City                         <chr> "Abidjan", "Abidjan", "Abidjan", "Abi...
## $ Country                      <chr> "Côte D'Ivoire", "Côte D'Ivoire", "Cô...
## $ Latitude                     <chr> "5.63N", "5.63N", "5.63N", "5.63N", "...
## $ Longitude                    <chr> "3.23W", "3.23W", "3.23W", "3.23W", "...
```

```
#
```

Select()ing columns

Let's use our first function, select.

```r
# Select at its core allows you pick out a selection of columns from your data
# Can folks remember how do this with base R?

# climate_df$dt
# climate_df[,1:2]

#Select does this, but with more power.
select(climate_df, dt)
```

```
## # A tibble: 239,177 x 1
##    dt
##    <date>
##  1 1849-01-01
##  2 1849-02-01
##  3 1849-03-01
##  4 1849-04-01
##  5 1849-05-01
##  6 1849-06-01
##  7 1849-07-01
##  8 1849-08-01
##  9 1849-09-01
## 10 1849-10-01
## # ... with 239,167 more rows
```

```r
# Multiple columns
select(climate_df, dt, City, Country)
```

```
## # A tibble: 239,177 x 3
##    dt         City    Country
##    <date>     <chr>   <chr>
##  1 1849-01-01 Abidjan Côte D'Ivoire
##  2 1849-02-01 Abidjan Côte D'Ivoire
##  3 1849-03-01 Abidjan Côte D'Ivoire
##  4 1849-04-01 Abidjan Côte D'Ivoire
##  5 1849-05-01 Abidjan Côte D'Ivoire
##  6 1849-06-01 Abidjan Côte D'Ivoire
##  7 1849-07-01 Abidjan Côte D'Ivoire
##  8 1849-08-01 Abidjan Côte D'Ivoire
##  9 1849-09-01 Abidjan Côte D'Ivoire
## 10 1849-10-01 Abidjan Côte D'Ivoire
## # ... with 239,167 more rows
```

```r
select(climate_df, dt:Country, -City)
```

```
## # A tibble: 239,177 x 4
##    dt         AverageTemperature AverageTemperatureUncertainty Country
##    <date>                  <dbl>                         <dbl> <chr>
##  1 1849-01-01               26.7                          1.44 Côte D'Ivoire
##  2 1849-02-01               27.4                          1.36 Côte D'Ivoire
##  3 1849-03-01               28.1                          1.61 Côte D'Ivoire
##  4 1849-04-01               26.1                          1.39 Côte D'Ivoire
##  5 1849-05-01               25.4                          1.2  Côte D'Ivoire
```

```
##  6 1849-06-01                24.8                           1.40 Côte D'Ivoire
##  7 1849-07-01                24.1                           1.25 Côte D'Ivoire
##  8 1849-08-01                23.6                           1.26 Côte D'Ivoire
##  9 1849-09-01                23.7                           1.23 Côte D'Ivoire
## 10 1849-10-01                25.3                           1.18 Côte D'Ivoire
## # ... with 239,167 more rows
```

```
# Similar process for rows - though we typically don't have rownames in R
slice(climate_df, 1:4)
```

```
## # A tibble: 4 x 7
##   dt         AverageTemperatu~ AverageTemperat~ City  Country Latitude Longitude
##   <date>                 <dbl>            <dbl> <chr> <chr>   <chr>    <chr>
## 1 1849-01-01              26.7             1.44 Abid~ Côte D~ 5.63N    3.23W
## 2 1849-02-01              27.4             1.36 Abid~ Côte D~ 5.63N    3.23W
## 3 1849-03-01              28.1             1.61 Abid~ Côte D~ 5.63N    3.23W
## 4 1849-04-01              26.1             1.39 Abid~ Côte D~ 5.63N    3.23W
```

Filtering a Dataframe with filter()

`filter()` allows you filter rows by certain conditions. Recall that we did this a bit with base R.

```
# base R
climate_df[climate_df$Country == "United States",]
```

```
## # A tibble: 8,455 x 7
##    dt         AverageTemperat~ AverageTemperat~ City  Country Latitude Longitude
##    <date>                <dbl>            <dbl> <chr> <chr>   <chr>    <chr>
##  1 1743-11-01             5.44             2.20 Chic~ United~ 42.59N   87.27W
##  2 1743-12-01            NA               NA    Chic~ United~ 42.59N   87.27W
##  3 1744-01-01            NA               NA    Chic~ United~ 42.59N   87.27W
##  4 1744-02-01            NA               NA    Chic~ United~ 42.59N   87.27W
##  5 1744-03-01            NA               NA    Chic~ United~ 42.59N   87.27W
##  6 1744-04-01             8.77            2.36 Chic~ United~ 42.59N   87.27W
##  7 1744-05-01            11.6             2.10 Chic~ United~ 42.59N   87.27W
##  8 1744-06-01            18.0             1.99 Chic~ United~ 42.59N   87.27W
##  9 1744-07-01            21.7             1.79 Chic~ United~ 42.59N   87.27W
## 10 1744-08-01            NA               NA    Chic~ United~ 42.59N   87.27W
## # ... with 8,445 more rows
```

```
# filter
filter(climate_df, Country == "United States")
```

```
## # A tibble: 8,455 x 7
##    dt         AverageTemperat~ AverageTemperat~ City  Country Latitude Longitude
##    <date>                <dbl>            <dbl> <chr> <chr>   <chr>    <chr>
##  1 1743-11-01             5.44             2.20 Chic~ United~ 42.59N   87.27W
##  2 1743-12-01            NA               NA    Chic~ United~ 42.59N   87.27W
##  3 1744-01-01            NA               NA    Chic~ United~ 42.59N   87.27W
##  4 1744-02-01            NA               NA    Chic~ United~ 42.59N   87.27W
##  5 1744-03-01            NA               NA    Chic~ United~ 42.59N   87.27W
```

```
##  6 1744-04-01             8.77             2.36 Chic~ United~ 42.59N   87.27W
##  7 1744-05-01            11.6              2.10 Chic~ United~ 42.59N   87.27W
##  8 1744-06-01            18.0              1.99 Chic~ United~ 42.59N   87.27W
##  9 1744-07-01            21.7              1.79 Chic~ United~ 42.59N   87.27W
## 10 1744-08-01            NA               NA    Chic~ United~ 42.59N   87.27W
## # ... with 8,445 more rows
```

```
# Pretty similar, but easy to chain stuff together
filter(climate_df, Country == "United States" & AverageTemperature > 25)
```

```
## # A tibble: 61 x 7
##    dt         AverageTemperat~ AverageTemperat~ City  Country Latitude Longitude
##    <date>                <dbl>            <dbl> <chr> <chr>   <chr>    <chr>
##  1 1761-07-01             27.8            2.39  Chic~ United~ 42.59N   87.27W
##  2 1868-07-01             25.1            0.699 Chic~ United~ 42.59N   87.27W
##  3 1900-08-01             25.2            0.49  Chic~ United~ 42.59N   87.27W
##  4 1921-07-01             25.6            0.264 Chic~ United~ 42.59N   87.27W
##  5 1947-08-01             26.4            0.199 Chic~ United~ 42.59N   87.27W
##  6 1955-08-01             25.3            0.153 Chic~ United~ 42.59N   87.27W
##  7 1959-08-01             25.1            0.205 Chic~ United~ 42.59N   87.27W
##  8 1988-08-01             25.4            0.305 Chic~ United~ 42.59N   87.27W
##  9 1995-08-01             25.9            0.283 Chic~ United~ 42.59N   87.27W
## 10 2012-07-01             25.9            0.516 Chic~ United~ 42.59N   87.27W
## # ... with 51 more rows
```

```
# Also worth noting here that we haven't saved any of this. We need to write to a new object.

us_df = filter(climate_df, Country == "United States")
```

Group Challenge

Work with the climate data we've using this class period. Construct a small set of code that does the following:
1. Slims down the full dataframe to one that contains the columns dt, AverageTemperature and City.
2. Filters the data for Paris and Mexico City (hint:this may be hard than you think... remember to do quality control on your data) with an average temperature less than 22.
3. Name this new dataframe "cold_paris_and_mexico_city"

```
# Piped
cold_paris_and_mexico_city = climate_df %>%
  select(dt, AverageTemperature, City) %>%
  filter(City == "Paris" | City == "Mexico",
         AverageTemperature < 22)

# Not piped
slim = select(climate_df, dt, AverageTemperature, City)
filtered = filter(slim, City == "Paris" | City == "Mexico",
                  AverageTemperature < 22)
cold_paris_and_mexico_city = filtered
```

The pipe

You can use the pipe operator to chain tidyverse functions together. This has a twofold benefit - it removes the clutter of creating a lot of intermediate objects in your workspace, but also makes things very human-readable.

```
climate_df %>%
  select(dt, AverageTemperature, City) %>%
  filter(City == "Paris" | City == "Mexico",
         AverageTemperature < 22)
```

```
## # A tibble: 5,306 x 3
##    dt         AverageTemperature City
##    <date>                  <dbl> <chr>
##  1 1835-01-01               13.1 Mexico
##  2 1835-02-01               12.3 Mexico
##  3 1835-03-01               13.9 Mexico
##  4 1835-04-01               16.3 Mexico
##  5 1835-05-01               16.9 Mexico
##  6 1835-06-01               16.5 Mexico
##  7 1835-07-01               15.4 Mexico
##  8 1835-08-01               15.2 Mexico
##  9 1835-09-01               14.2 Mexico
## 10 1835-10-01               13.4 Mexico
## # ... with 5,296 more rows
```

Learning new verbs - Group Assignment

You've had an introduction to the tidyverse verbs, but there a few more to learn. We're going to practice extracting information from the web, which is a key tool in the toolkit of data scientists.

Figure out what the `mutate()` and `summarize()` functions do. Use the web (particularly useful links include the tidyverse website, and a help-forum for stats and programming called stack overflow). Here are the specifics: 1) Be able to briefly describe each function
2) Come up with an example from the climate data frame we've been working on.
As always, be ready to present to your peers.