Coding Clinic 1

Stat 405

2019-01-26

Hi, we're Ajjit and Alex!

Overview

- Directories
- Data Read in
- Filtering, Selecting, and Renaming Dataframes

Get Started

Directories: What are they?

- Just a location for storing files
- You can think of them as folders

Working Directories

- By default, R has a **working directory**
- This is where R will look, by default, for files you ask it to load
- To find R's current working directory:

```
getwd()
```

```
## [1] "C:/Users/ajjit/Google Drive/Documents/senior_spring_classes"
```

- To set R's current working directory to something else:

```
setwd("C:/Users/ajjit/Downloads")
```

Why are directories important?

- When you're reading data into R, you have two choices:
  - Provide full path to file
  - Provide relative path to file (from current working directory)
- For example you can do

```
data = read.csv("C:/Users/ajjit/Downloads/test_data.csv")
```

- Or you can do

```
setwd("C:/Users/ajjit/Downloads")
data = read.csv("test_data.csv")
```

- Second method is preferred if you're reading in multiple files/ working on a big project

How to find filepaths

- Mac:
  - Find the file in Finder > Right Click > Get Info > Where
  - Or open Terminal, drag and drop the file into the screen, and it will output the full path
- Windows:
  - Find the file in File Explorer > Right Click > Properties > Location
  - Or use R's `file.choose()` function

Exercise 1

- Download data on Chicago's bike share stations from https://tinyurl.com/ybfk995s and save it in your downloads folder
- Read in the data using `read.csv()` and the full filepath. Name the dataframe `stations`
- Read in the data using `read.csv()` but now use `setwd()` and the relative filepath.

Exercise 1

- Download data on Chicago's bike share stations from https://tinyurl.com/ybfk995s and save it in your downloads folder
- Read in the data using `read.csv()` and the full filepath. Name the dataframe `stations`
- Read in the data using `read.csv()` but now use `setwd()` and the relative filepath.

```
stations = read.csv("C:/Users/ajjit/Downloads/Divvy_Bicycle_Stations.
                    stringsAsFactors = F)
```

Selecting columns in a dataframe

- Way to access dataframe itmes is `df[rows, columns]`

Selecting columns in a dataframe

- Way to access dataframe itmes is `df[rows, columns]`
- Suppose you only need Station Addresses, Latitude, and Longitude

```
stations_reduced = stations[,c('Address', 'Latitude', 'Longitude')]
head(stations_reduced, n =2)
```

```
##                   Address Latitude Longitude
## 1 Jeffery Blvd & 71st St 41.76664 -87.57645
## 2 Loomis St & Archer Ave 41.84163 -87.65743
```

Selecting columns in a dataframe

- Way to access dataframe itmes is `df[rows, columns]`
- Suppose you only need Station Addresses, Latitude, and Longitude

```
stations_reduced = stations[,c('Address', 'Latitude', 'Longitude')]
head(stations_reduced, n =2)
```

```
##                   Address Latitude Longitude
## 1 Jeffery Blvd & 71st St 41.76664 -87.57645
## 2 Loomis St & Archer Ave 41.84163 -87.65743
```

Exercise: Create a new dataframe called `station1` that selects only the ID and Station.Name columns

- Bonus: How would you do this with column indexes rather than column names?

Filtering rows in a dataframe

- Say you want to find all large bike stations with 55 or greater stations in service

```
stations[stations$stations.in.Service > 55,]
```

Filtering rows in a dataframe

- Say you want to find all large bike stations with 55 or greater stations in service

```
stations[stations$stations.in.Service > 55,]
```

Exercise: How many stations have less than 12 Total stations AND less than 10 stations in Service. You will need to use the & operator.

Creating new columns in a dataframe

- If you want to create a new column:

```
# This creates a column of 0's
stations$new_column = rep(0, nrow(stations))
```

- Often new columns will be functions of other columns. Say you want the total number of stations rounded to the nearest 10

```
stations$rounded_total_stations = round(stations$Total.Docks,
                                        digits = -1)
```

Creating new columns in a dataframe

- If you want to create a new column:

```
# This creates a column of 0's
stations$new_column = rep(0, nrow(stations))
```

- Often new columns will be functions of other columns. Say you want the total number of stations rounded to the nearest 10

```
stations$rounded_total_stations = round(stations$Total.Docks,
                                        digits = -1)
```

Exercise: create a column called `PctstationsInService` which is the percentage of stations in service at each station. Then create a column called `NeedsToBeFixed` which is 1 if the percentage of stations in service is greater than 95% or 0 otherwise

Manipulating strings

- `stringr`: R package that vecrorizes string manipulation and makes it easy
- Install and load it by running:

```
install.packages('stringr')
library(stringr)
```

stringr continued

- Say you want to find all stations that are on 71st street. How would you go about this?

stringr continued

- Say you want to find all stations that are on 71st street. How would you go about this?

```
street_72_log = str_detect(stations$Address, "71") #logical vector
street_72 =     str_subset(stations$Address, "71") #actual values
street_72
```

```
## [1] "Jeffery Blvd & 71st St"      "Stony Island Ave & 71st St"
## [3] "Calumet Ave & 71st St"       "Cottage Grove Ave & 71st St"
```

stringr continued

- Say you want to find all stations that are on 71st street. How would you go about this?

```
street_72_log = str_detect(stations$Address, "71") #logical vector
street_72 =     str_subset(stations$Address, "71") #actual values
street_72
```

```
## [1] "Jeffery Blvd & 71st St"      "Stony Island Ave & 71st St"
## [3] "Calumet Ave & 71st St"       "Cottage Grove Ave & 71st St"
```

- Or what if the data was badly coded and all the 71st were actually 72nd?

stringr continued

- Say you want to find all stations that are on 71st street. How would you go about this?

```
street_72_log = str_detect(stations$Address, "71") #logical vector
street_72 =     str_subset(stations$Address, "71") #actual values
street_72
```

```
## [1] "Jeffery Blvd & 71st St"      "Stony Island Ave & 71st St"
## [3] "Calumet Ave & 71st St"       "Cottage Grove Ave & 71st St"
```

- Or what if the data was badly coded and all the 71st were actually 72nd?

```
str_replace_all(street_72, "71st", "72nd")
```

```
## [1] "Jeffery Blvd & 72nd St"      "Stony Island Ave & 72nd St"
## [3] "Calumet Ave & 72nd St"       "Cottage Grove Ave & 72nd St"
```

stringr continued

- Say you want to find all stations that are on 71st street. How would you go about this?

```
street_72_log = str_detect(stations$Address, "71") #logical vector
street_72 =     str_subset(stations$Address, "71") #actual values
street_72
```

```
## [1] "Jeffery Blvd & 71st St"      "Stony Island Ave & 71st St"
## [3] "Calumet Ave & 71st St"       "Cottage Grove Ave & 71st St"
```

- Or what if the data was badly coded and all the 71st were actually 72nd?

```
str_replace_all(street_72, "71st", "72nd")
```

```
## [1] "Jeffery Blvd & 72nd St"      "Stony Island Ave & 72nd St"
## [3] "Calumet Ave & 72nd St"       "Cottage Grove Ave & 72nd St"
```

- Other useful functions: `str_replace()`, `str_to_lower()`, `str_to_title`.

Renaming Dataframe columns

- To access columns names

```
colnames(stations)
```

Renaming Dataframe columns

- To access columns names

```
colnames(stations)
```

- To change all column names at once. Think about what the column names will be in this case:

```
colnames(stations) = as.character(seq(1, 10, by =1))
```

Renaming Dataframe columns

- To access columns names

```
colnames(stations)
```

- To change all column names at once. Think about what the column names will be in this case:

```
colnames(stations) = as.character(seq(1, 10, by =1))
```

- To change one column's name

```
colnames(stations)[colnames(stations) ==
                     'Station.Name']  = 'station_name'
```

Putting it all together

Exercise: Rename all columns so that they follow these rules. Replace all periods with underscores (_), and make every column name lowercase. As an example: `This.Title` would become `this_title`.

That's all!