Assignment 1 | Big Data Analytics Capstone

Michael Rose

Titanic Data

Preparation

Load Data

First we must load the data in. Since this is in xls format, we will use the `readxl` package.

```
# load data
titanic <- readxl::read_xls("/home/michael/Desktop/SS/BigDataCapstone/Titanic.xls")
```

```
## New names:
## * CLASS -> CLASS..1
## * `` -> `..5`
## * CLASS -> CLASS..6
```

CLASS..1	AGE	SEX	SURVIVE	..5	CLASS..6
1	1	1	1	NA	0=crew
1	1	1	1	NA	1=first
1	1	1	1	NA	2=second
1	1	1	1	NA	3=third
1	1	1	1	NA	NA
1	1	1	1	NA	AGE

From the table above, we see that it read in 5 columns and added a 6th which seems to originally be a legend. We need to clean this data frame up.

Create Factor Encodings

```
# clean up column names
titanic %<>% rename("class" = CLASS..1, "age" = AGE, "sex" = SEX, "survive" = SURVIVE)

# change doubles to factors for manipulation
titanic %<>% mutate_if(is.numeric, as.factor)

# change factor levels according to legend
titanic %<>% mutate(class = recode(class,
                                   "0" = "crew",
                                   "1" = "first",
                                   "2" = "second",
                                   "3" = "third"),
                    age = recode(age,
                                 "0" = "child",
                                 "1" = "adult"),
```

```
                    sex = recode(sex,
                                 "0" = "female",
                                 "1" = "male"),
                    survive = recode(survive,
                                     "0" = "no",
                                     "1" = "yes")) %>%
  select(class, age, sex, survive)

# check for missing values
titanic %>%
  select_if(function(x) any(is.na(x))) %>%
  summarize_each(funs(sum(is.na(.))))
```

```
## # A tibble: 1 x 0
```

class	age	sex	survive
first	adult	male	yes
first	adult	male	yes
first	adult	male	yes
first	adult	male	yes
first	adult	male	yes
first	adult	male	yes

This dataframe looks much tidier, and we have no missing values.

Problem 1

Dataset : Titanic

Question: What are the proportions of each factor in regards to survival rate?

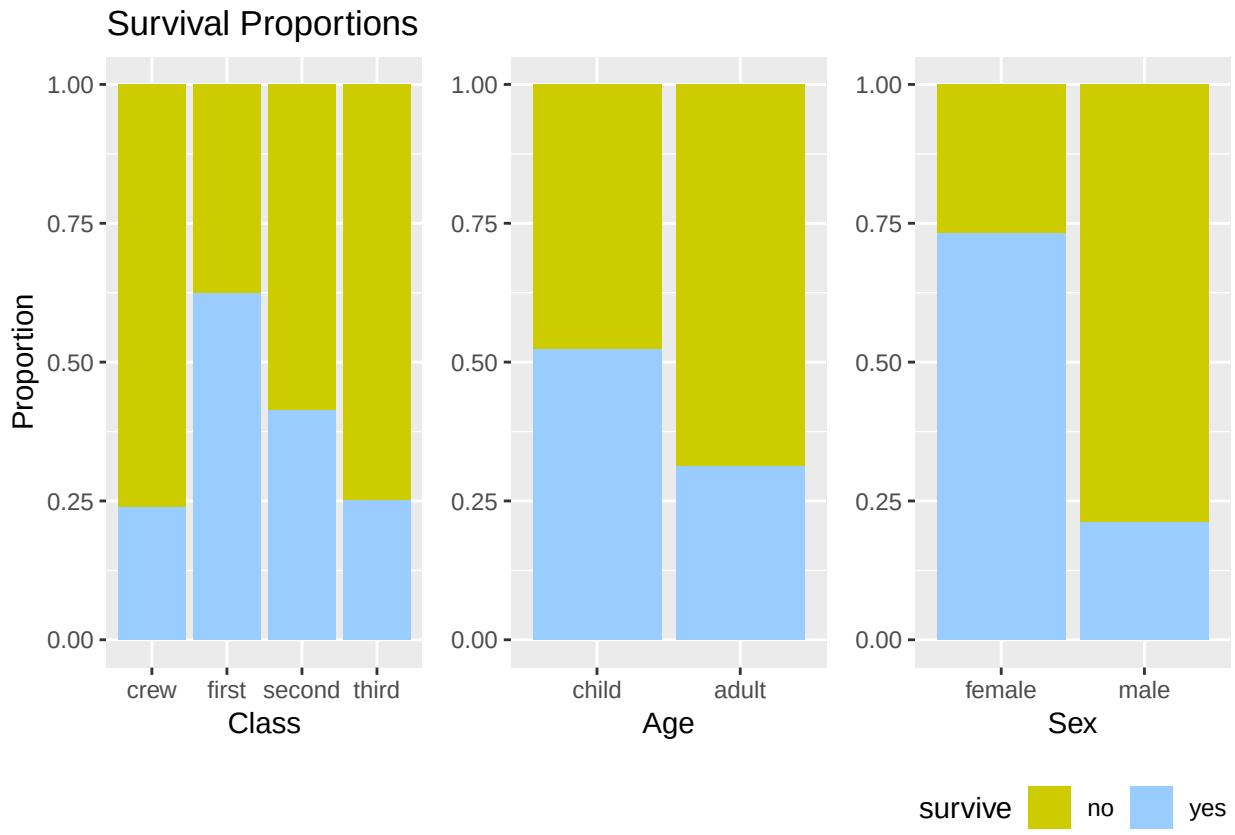
Methods Used:

- ggplot
- geom_bar
- xlab
- ylab
- scale_fill_manual
- theme
- ggtitle
- ggarrange

Visualization Results:

```
p_class <- titanic %>%
  ggplot(aes(x = class, fill = survive)) +
  geom_bar(position = "fill") +
  xlab("Class") + ylab("Proportion") +
  scale_fill_manual(values = c("#CCCC00", "#99CCFF")) +
  theme(legend.position = "none") +
  ggtitle("Survival Proportions")
```

```
p_age <- titanic %>%
  ggplot(aes(x = age, fill = survive)) +
  geom_bar(position = "fill") +
  xlab("Age") + ylab(NULL) +
  scale_fill_manual(values = c("#CCCC00", "#99CCFF")) +
  theme(legend.position = "none")

p_sex <- titanic %>%
  ggplot(aes(x = sex, fill = survive)) +
  geom_bar(position = "fill") +
  xlab("Sex") + ylab(NULL) +
  scale_fill_manual(values = c("#CCCC00", "#99CCFF")) +
  theme(legend.position = "bottom")

ggarrange(p_class, p_age, p_sex, nrow = 1)
```



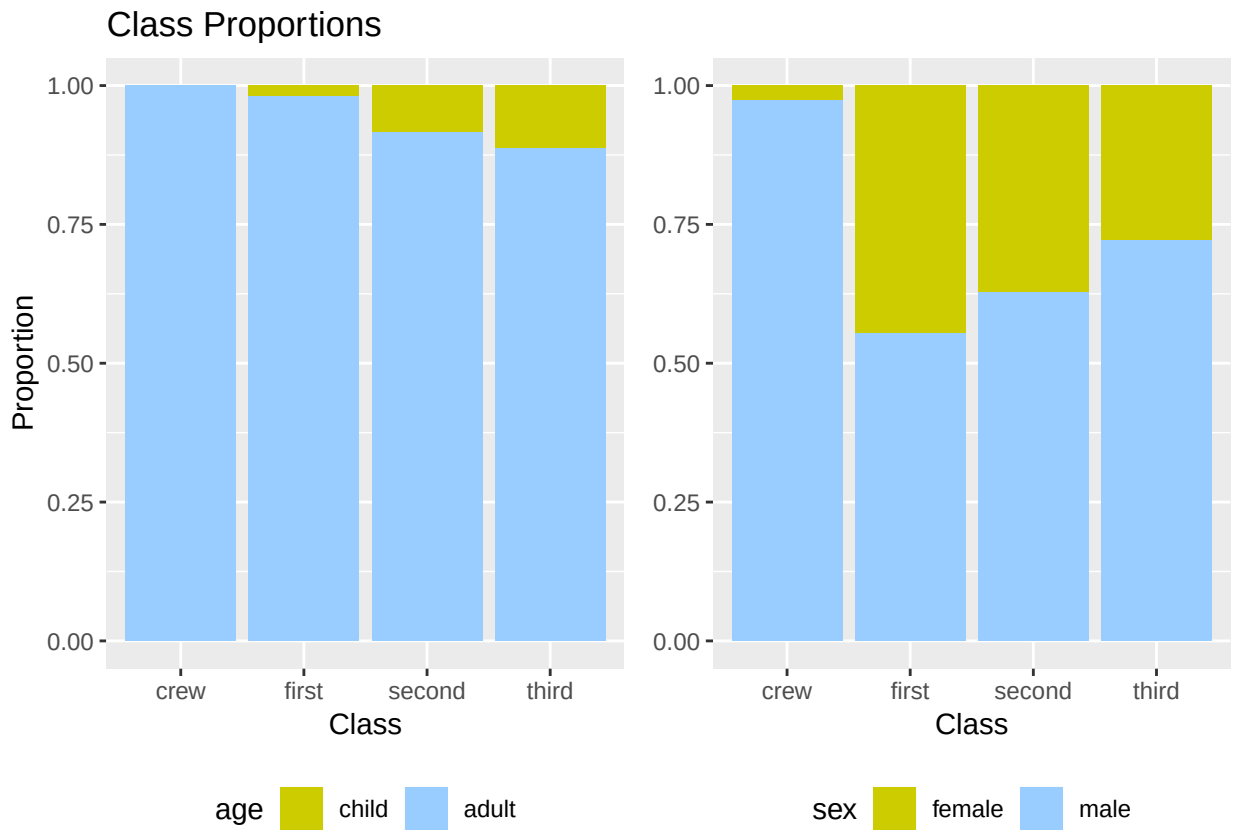
From the plots above, we see the following:

- The crew took the largest hit in mortality. It seems that many of them went down with the ship.
- First class had the most people who survived. This is strange, because the majority of people didn't sail first class. It seems that the poorer you were, the more likely you were to not survive.

class	n
crew	885
first	325
second	285
third	706

- adults and men were more likely to die, as they seem to have saved the women and children first.

```
p_class_age <- titanic %>%
  ggplot(aes(x = class, fill = age)) +
  geom_bar(position = "fill") +
  xlab("Class") + ylab("Proportion") +
  scale_fill_manual(values = c("#CCCC00", "#99CCFF")) +
  theme(legend.position = "bottom") +
  ggtitle("Class Proportions")

p_class_sex <- titanic %>%
  ggplot(aes(x = class, fill = sex)) +
  geom_bar(position = "fill") +
  xlab("Class") + ylab(NULL) +
  scale_fill_manual(values = c("#CCCC00", "#99CCFF")) +
  theme(legend.position = "bottom")

ggarrange(p_class_age, p_class_sex, nrow = 1)
```



From the plots above, we see the following:

- The crew consisted entirely of adults.
- As the class decreased, the proportion of children increased
- First class contained the largest proportion of females, with a steady decline in females by class.

Problem 2

Additional R Courses

This section will focus on information gained from the datacamp courses

- **Functional Programming with purrr**
- **Intermediate Functional Programming with purrr**.
- **Working with Web Data in R**

Dataset

The New York State government has an open data portal that allows a user to get data in JSON format. Here is a link to the dataset used : NY State Retail Food Stores

Methods Used:

- fromJSON
- str
- map_* variants
- safe_extract
- flatten_chr
- set_names
- paste0
- glimpse
- head
- replace
- is.na
- mutate_* variants
- select
- summary
- bind_cols
- readOGR
- coordinates
- proj4string
- names
- CRS
- SPtransform

```
# grab JSON data
food_markets_raw <- jsonlite::fromJSON("https://data.ny.gov/api/views/9a8c-vfzj/rows.json?accessType=DO
```

What does the JSON representation look like?

```r
str(food_markets_raw, max.level = 1)
```

```
## List of 2
##  $ meta:List of 1
##  $ data:List of 29389
```

We see that our JSON consists of 2 components, meta (metadata), and data. Lets look at the different structures:

meta

```r
str(food_markets_raw$meta, max.level = 1, list.len = 5)
```

```
## List of 1
##  $ view:List of 37
```

Our metadata is a single list. We can look into its nested structure and pull the columns component. This tells us what data we have for each food market in the data component. Lets grab that component and place it in a character vector.

```r
# look at json data
# jsonedit(food_markets_raw[[c("meta", "view")]])

c_names <- food_markets_raw[[c("meta", "view", "columns")]] %>%
  map_chr("name")

# output of column names
c_names %>% kabletable()
```

x
sid
id
position
created_at
created_meta
updated_at
updated_meta
meta
County
License Number
Operation Type
Establishment Type
Entity Name
DBA Name
Street Number
Street Name
Address Line 2
Address Line 3
City
State
Zip Code
Square Footage
Location

data

```
str(food_markets_raw$data, max.level = 1, list.len = 5)
```

```
## List of 29389
##  $ :List of 23
##  $ :List of 23
##  $ :List of 23
##  $ :List of 23
##  $ :List of 23
##   [list output truncated]
```

So we have a data set of lists for 29,389 different food markets. Lets focus on this data for now

```
# focus on data component
food_markets <- food_markets_raw$data

# add column names to the food markets
food_markets %<>% map(set_names, c_names)

# look at one of the columns
food_markets %>%
  map_chr("DBA Name") %>%
  head(10) %>%
  kabletable()
```

x
KNOX STORE
RITE AID #04805
KISS MART 802
MICKEYS CORNER
KATIES PATH 2 SNACKS
KUHAR FAMILY FARM
GREEN ACRES OF NEW YORK
HISSHO SUSHI@ST PETERS
HICKORY FARMS #11322
DMV NEWS

Our dataframe is shaping up nicely. From looking at the json information above (it won't knit to PDF, but the code is available commented), we see that 22 of the 23 variables can be easily extracted. Location holds some unparsed JSON that must be handled separately.

Creating a Dataframe

```
# grab target names
names_to_process <- c_names[c_names != "Location"]

# create a function to extract with the presence of NULLs
safe_extract <- function(grabbed_list, item){
  res <- grabbed_list[item]
  null_found <- map_lgl(res, is.null)
  res[null_found] <- NA
  res
}

# extract from food markets
markets_frame <- food_markets %>%
  map_df(safe_extract, names_to_process)

markets_frame %>% head() %>% kabletable(scale = TRUE)
```

sid	id	position	created_at	created_meta	updated_at	updated_meta	meta	County	License Number	Operation Type	Establishment Type	Entity Name	DBA Name	Street Number	Street Name	Address Line 2	Address Line 3	City	State	Zip Code	Square Footage
5	E1677F10-F9EE-4A88-8CAD-1EA78CEFA487	5	1533248982	708543	1533248982	708543	NA	Albany	730927	Store	JAC	NAPREDAK LLC	KNOX STORE	2160	BERNE ALTAMONT RD			KNOX	NY	12107	700
13	7FB28271-3F01-4AB3-A2D8-5534906802E9	13	1533248982	708543	1533248982	708543	NA	Albany	732394	Store	A	WALGREEN EASTERN CO INC	RITE AID #04805	2463	US ROUTE 9W			RAVENA	NY	12143	10000
2925	E871D5F1-545A-4466-B530-35920A4E6217	2925	1533248983	708543	1533248983	708543	NA	Erie	148458	Store	JAC	DELTA SONIC CAR WASH SYSTEMS INC	KISS MART 802	NA	RIDGE AT LANGER ST			WEST SENECA	NY	14224	2400
47	1374C7C7-3F4D-4332-A72B-EFB12B7A70F3	47	1533248982	708543	1533248982	708543	NA	Albany	730735	Store	A	PERUNOVIC MIODRAG	MICKEYS CORNER	6	EMPIRE STATE PLAZA			ALBANY	NY	12228	0
71	FB17F4AB-A8F3-4153-A77D-81F271F6FB80	71	1533248982	708543	1533248982	708543	NA	Albany	734731	Store	A	BROOKS CAITLYN	KATIES PATH 2 SNACKS	160	STATE ST JUSTICE BLVD			ALBANY	NY	12223	600
77	936865B2-D3D0-41A7-B557-FCF0D62DA740	77	1533248982	708543	1533248982	708543	NA	Albany	722162	Store	JAC	KUHAR FAMILY FARM LLC	KUHAR FAMILY FARM	26	CR 353			RENSSELAERVILLE	NY	12147	400

The dataframe looks really nice. We have everything well formatted except for location. It looks really small here because it has been scaled down to fit in the width of this pdf. Here is an overview of the current columns:

```
markets_frame %>% glimpse()
```

```
## Observations: 29,389
## Variables: 22
## $ sid                 <int> 5, 13, 2925, 47, 71, 77, 80, 83, 3118, 11...
## $ id                  <chr> "E1677F10-F9EE-4A88-8CAD-1EA78CEFA487", "...
## $ position            <int> 5, 13, 2925, 47, 71, 77, 80, 83, 3118, 11...
## $ created_at          <int> 1533248982, 1533248982, 1533248983, 15332...
```

```
## $ created_meta         <chr> "708543", "708543", "708543", "708543", "...
## $ updated_at           <int> 1533248982, 1533248982, 1533248983, 15332...
## $ updated_meta         <chr> "708543", "708543", "708543", "708543", "...
## $ meta                 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ County               <chr> "Albany", "Albany", "Erie", "Albany", "Al...
## $ `License Number`     <chr> "730927", "732394", "148458", "730735", "...
## $ `Operation Type`     <chr> "Store", "Store", "Store", "Store", "Stor...
## $ `Establishment Type` <chr> "JAC   ", "A     ", "JAC   ", "A     ", "...
## $ `Entity Name`        <chr> "NAPREDAK LLC                    ", "WAL...
## $ `DBA Name`           <chr> "KNOX STORE             ", "RITE AID #048...
## $ `Street Number`      <chr> "2160", "2463", NA, "6", "160", "26", "28...
## $ `Street Name`        <chr> "BERNE ALTAMONT RD           ", "US ROUTE...
## $ `Address Line 2`     <chr> "             ", "              ", "      ...
## $ `Address Line 3`     <chr> "               ", "               ", "  ...
## $ City                 <chr> "KNOX              ", "RAVENA           ...
## $ State                <chr> "NY", "NY", "NY", "NY", "NY", "NY", "NY",...
## $ `Zip Code`           <chr> "12107", "12143", "14224", "12228", "1222...
## $ `Square Footage`     <chr> "700", "10000", "2400", "0", "600", "400"...
```

Add Location

To add Location we need to unpack the Location components from the JSON file.

```
# Grab location
location_raw <- food_markets %>%
  map("Location")

# look at one of the location variables
location_raw[[345]]
```

```
## [[1]]
## [1] "{\"address\":\"965 CTY\",\"city\":\"NIVERVILLE\",\"state\":\"NY\",\"zip\":\"12130\"}"
##
## [[2]]
## NULL
##
## [[3]]
## NULL
##
## [[4]]
## NULL
##
## [[5]]
## [1] TRUE
```

We have a keyless JSON with a series of nameless lists. Now we should extract names from the meta component and apply them.

```
# grab location items
locs <- which(c_names == "Location")

# pull locations from JSON metadata
```

```
location_meta <- food_markets_raw[[c("meta", "view", "columns")]][[locs]]

# location names
loc_names <- location_meta[["subColumnTypes"]] %>% flatten_chr()

# apply names
location_raw %<>% map(set_names, loc_names)

# check earlier location to see results
location_raw[[345]]
```

```
## $human_address
## [1] "{\"address\":\"965 CTY\",\"city\":\"NIVERVILLE\",\"state\":\"NY\",\"zip\":\"12130\"}"
##
## $latitude
## NULL
##
## $longitude
## NULL
##
## $machine_address
## NULL
##
## $needs_recoding
## [1] TRUE
```

Looks better! Now we know what each of these lists means. Next up we need to parse the human address section. We will do the following:

- parse the human adress section
- bind the elements to a dataframe
- add "ha_" to each name to avoid problems when we combine these

```
# create replacement address for NULL addresses
replace_string <- "{\"address\":\"0000 NO RD\",\"city\":\"NONE\",\"state\":\"NY\",\"zip\":\"00000\"}"

ha <- location_raw %>%
  map("human_address", .null = NA) %>%
  replace(is.na(.), replace_string) %>%
  flatten_chr() %>%
  map_df(fromJSON) %>%
  set_names(paste0("ha_", names(.)))

ha %>% head() %>% kabletable()
```

ha_address	ha_city	ha_state	ha_zip
2160 BERNE ALTAMONT RD	KNOX	NY	12107
2463 US ROUTE	RAVENA	NY	12143
0000 NO RD	NONE	NY	00000
6 EMPIRE STATE PLAZA	ALBANY	NY	12228
160 STATE ST JUSTICE BLVD	ALBANY	NY	12223
26 CR	RENSSELAERVILLE	NY	12147

Handling the Remaining Variables

We still need to grab the rest of the variables from the Locations data frame.

```
everything_else <- location_raw %>%
  map_df(safe_extract, loc_names[loc_names != "human_address"]) %>%
  mutate_at(vars(latitude, longitude), as.numeric)

everything_else %>% head() %>% kabletable()
```

latitude	longitude	machine_address	needs_recoding
NA	NA	NA	TRUE
NA	NA	NA	TRUE
NA	NA	NA	NA
NA	NA	NA	TRUE
NA	NA	NA	TRUE
NA	NA	NA	TRUE

This looks a little off, considering all the latitudes and longitudes display NA. Lets take a closer look to make sure there isn't an error lurking somewhere.

```
everything_else %>%
  select(latitude, longitude) %>%
  map(summary)
```

```
## $latitude
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   40.51   40.70   40.84   41.45   42.50   44.99    5415
##
## $longitude
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  -79.76  -74.18  -73.95  -74.62  -73.86  -71.94    5415
```

```
dim(everything_else)[[1]]
```

```
## [1] 29389
```

We see from the output above that, while there are 5415 NAs for (latitude, longitude), we still have 29389 - 5415 = 23974 value containing points. We also see that our latitudes are in the [40.51, 44.99] range and out longitudes are in the [-79.76, -71.94] range. This coincides the with the New York Latitude and Longitude Map which indicates we should lie within [40, 50] and [-80, -72].

Putting the Dataframes Together

Now that we have the markets, human address and everything else data frames, let's combine them.

```
# combine dataframes
markets_frame <- bind_cols(markets_frame, ha, everything_else)

# remove leading / trailing white space
```

```r
markets_frame %<>% mutate_if(is.character, trimws)

markets_frame %>% head() %>% kabletable(scale = TRUE)
```

sid	id	position	created_at	created_meta	updated_at	updated_meta	meta	County	License Number	Operation Type	Establishment Type	Entity Name	DBA Name
5	E1677F10-F9EE-4A88-8CAD-1EA78CEFA487	5	1533248982	708543	1533248982	708543	NA	Albany	730927	Store	JAC	NAPREDAK LLC	KNOX STORE
13	7FB28271-3F01-4AB3-A2D8-5534906802E9	13	1533248982	708543	1533248982	708543	NA	Albany	732394	Store	A	WALGREEN EASTERN CO INC	RITE AID #04805
2925	E871D5F1-545A-4466-B530-35920A4E6217	2925	1533248983	708543	1533248983	708543	NA	Erie	148458	Store	JAC	DELTA SONIC CAR WASH SYSTEMS INC	KISS MART 802
47	1374C7C7-3F4D-4332-A72B-EFB12B7A70F3	47	1533248982	708543	1533248982	708543	NA	Albany	730735	Store	A	PERUNOVIC MIODRAG	MICKEYS CORNER
71	FB17F4AB-A8F3-4153-A77D-81F271F6FB80	71	1533248982	708543	1533248982	708543	NA	Albany	734731	Store	A	BROOKS CAITLYN	KATIES PATH 2 SNACKS
77	936865B2-D3D0-41A7-B557-FCF0D62DA740	77	1533248982	708543	1533248982	708543	NA	Albany	722162	Store	JAC	KUHAR FAMILY FARM LLC	KUHAR FAMILY FARM

sid	Street Number	Street Name	Address Line 2	Address Line 3	City	State	Zip Code	Square Footage	ha_address	ha_city	ha_state	ha_zip	latitude	longitude	machine_address	needs_recoding
5	2160	BERNE ALTAMONT RD			KNOX	NY	12107	700	2160 BERNE ALTAMONT RD	KNOX	NY	12107	NA	NA	NA	TRUE
13	2463	US ROUTE 9W			RAVENA	NY	12143	10000	2463 US ROUTE	RAVENA	NY	12143	NA	NA	NA	TRUE
2925	NA	RIDGE AT LANGER ST			WEST SENECA	NY	14224	2400	0000 NO RD	NONE	NY	00000	NA	NA	NA	NA
47	6	EMPIRE STATE PLAZA			ALBANY	NY	12228	0	6 EMPIRE STATE PLAZA	ALBANY	NY	12228	NA	NA	NA	TRUE
71	160	STATE ST JUSTICE BLVD			ALBANY	NY	12223	600	160 STATE ST JUSTICE BLVD	ALBANY	NY	12223	NA	NA	NA	TRUE
77	26	CR 353			RENSSELAERVILLE	NY	12147	400	26 CR	RENSSELAERVILLE	NY	12147	NA	NA	NA	TRUE

Finally, our table is all together. Once again, the table display is small due to the width of a pdf file. Here is an overview of the different columns:

```r
markets_frame %>% glimpse()
```

```
## Observations: 29,389
## Variables: 30
## $ sid                  <int> 5, 13, 2925, 47, 71, 77, 80, 83, 3118, 11...
## $ id                   <chr> "E1677F10-F9EE-4A88-8CAD-1EA78CEFA487", "...
## $ position             <int> 5, 13, 2925, 47, 71, 77, 80, 83, 3118, 11...
## $ created_at           <int> 1533248982, 1533248982, 1533248983, 15332...
## $ created_meta         <chr> "708543", "708543", "708543", "708543", "...
## $ updated_at           <int> 1533248982, 1533248982, 1533248983, 15332...
## $ updated_meta         <chr> "708543", "708543", "708543", "708543", "...
## $ meta                 <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ County               <chr> "Albany", "Albany", "Erie", "Albany", "Al...
## $ `License Number`     <chr> "730927", "732394", "148458", "730735", "...
## $ `Operation Type`     <chr> "Store", "Store", "Store", "Store", "Stor...
## $ `Establishment Type` <chr> "JAC", "A", "JAC", "A", "A", "JAC", "JAC"...
## $ `Entity Name`        <chr> "NAPREDAK LLC", "WALGREEN EASTERN CO INC"...
## $ `DBA Name`           <chr> "KNOX STORE", "RITE AID #04805", "KISS MA...
## $ `Street Number`      <chr> "2160", "2463", NA, "6", "160", "26", "28...
## $ `Street Name`        <chr> "BERNE ALTAMONT RD", "US ROUTE 9W", "RIDG...
## $ `Address Line 2`     <chr> "", "", "", "", "", "", "", "", "", "", "...
## $ `Address Line 3`     <chr> "", "", "", "", "", "", "", "", "", "", "...
## $ City                 <chr> "KNOX", "RAVENA", "WEST SENECA", "ALBANY"...
## $ State                <chr> "NY", "NY", "NY", "NY", "NY", "NY", "NY",...
## $ `Zip Code`           <chr> "12107", "12143", "14224", "12228", "1222...
## $ `Square Footage`     <chr> "700", "10000", "2400", "0", "600", "400"...
## $ ha_address           <chr> "2160 BERNE ALTAMONT RD", "2463 US ROUTE"...
## $ ha_city              <chr> "KNOX", "RAVENA", "NONE", "ALBANY", "ALBA...
## $ ha_state             <chr> "NY", "NY", "NY", "NY", "NY", "NY", "NY",...
## $ ha_zip               <chr> "12107", "12143", "00000", "12228", "1222...
## $ latitude             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ longitude            <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ machine_address      <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ needs_recoding       <lgl> TRUE, TRUE, NA, TRUE, TRUE, TRUE, TRUE, T...
```

Question 1

Now that we have all this information, what does it look like on a map?

First we must load our shape file, which contains all the coordinates of our New York map.

```
# read in shapefile for NY state
NY_state <- readOGR("/home/michael/Desktop/SS/BigDataCapstone", layer = "nys")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/home/michael/Desktop/SS/BigDataCapstone", layer: "nys"
## with 1 features
## It has 53 fields
```

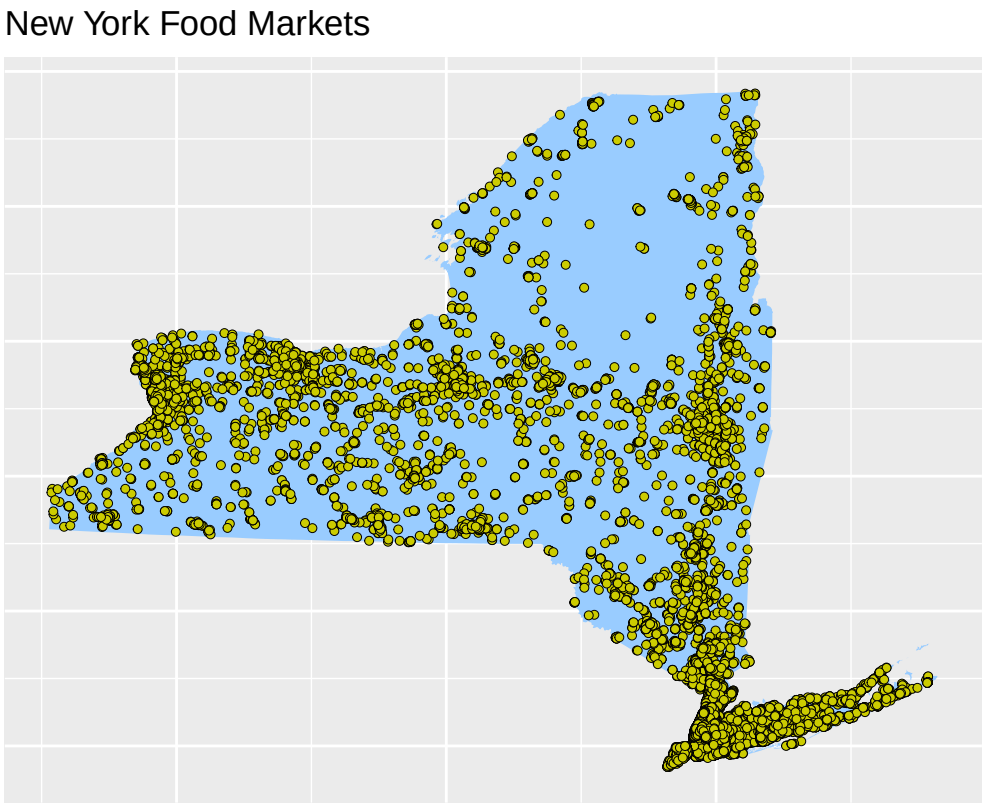
Before we plot our points, we need to change the orientation of the latitude and longitude points. Currently they are **unprojected**, whereas our map is in a projected coordinate system. So we much project these points before we plot.

```
# grab non-NA lat / long components
map_markets_frame <- markets_frame[!is.na(markets_frame$latitude) & !is.na(markets_frame$longitude), ]

# coerce square footage to numeric
markets_frame %<>% mutate(`Square Footage` = as.numeric(`Square Footage`))
map_markets_frame %<>% mutate(`Square Footage` = as.numeric(`Square Footage`))

# grab coordinates
coordinates(map_markets_frame) <- ~longitude + latitude

# set projected coordinate system
proj4string(map_markets_frame) <- CRS("+proj=longlat +datum=NAD83")

# add projected coordinates to map
map_markets_frame %<>% spTransform(CRS(proj4string(NY_state)))

# create geo data dataframe
geo_data <- data.frame(coordinates(map_markets_frame))

# set coord names
names(geo_data) <- c("x", "y")
```

Now we can plot it:

```
ggplot() +
  geom_polygon(data = NY_state, aes(x = long, y = lat, group = group), fill = "#99CCFF") +
  labs(x = "", y = "", title = "New York Food Markets") +
  theme(axis.ticks = element_blank(),
        axis.text = element_blank()) +
  geom_point(data = geo_data, aes(x = x, y = y), shape = 21, fill = "#CCCC00", color = "black", stroke =
  coord_equal(ratio = 1)
```

```
## Regions defined for each Polygons
```

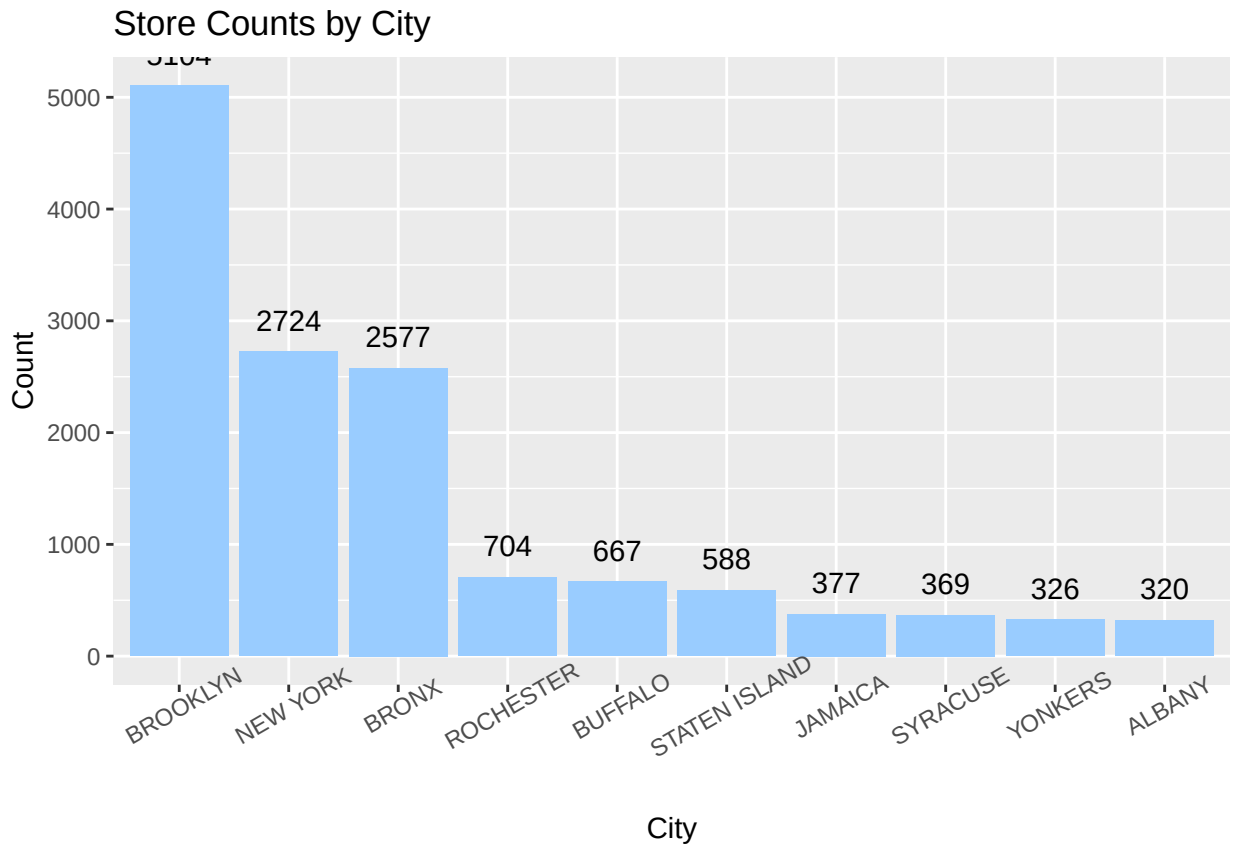


As expected, New York City is packed with Food shops. Certain areas of upstate New York are particularly sparse.

Question 2

Which cities have the largest number of food shops?

```
markets_frame %>%
  group_by(City) %>%
  count(sort = TRUE) %>%
  head(10) %>%
ggplot() +
  geom_col(aes(x = reorder(City, -n), y = n), fill = "#99CCFF") +
  xlab("City") + ylab("Count") +
  theme(axis.text.x = element_text(angle = 30)) +
  geom_text(aes(label = n, x = City, y = n), vjust = -1) +
  ggtitle("Store Counts by City")
```



From the plot above, we see that Brooklyn has roughly twice as many food shops as the rest of New York City. We also see that Bronx has roughly as many food shops as NY, NY.

Question 3

How are store sizes distributed?

```
markets_frame %>% select('Square Footage') %>% summary()
```

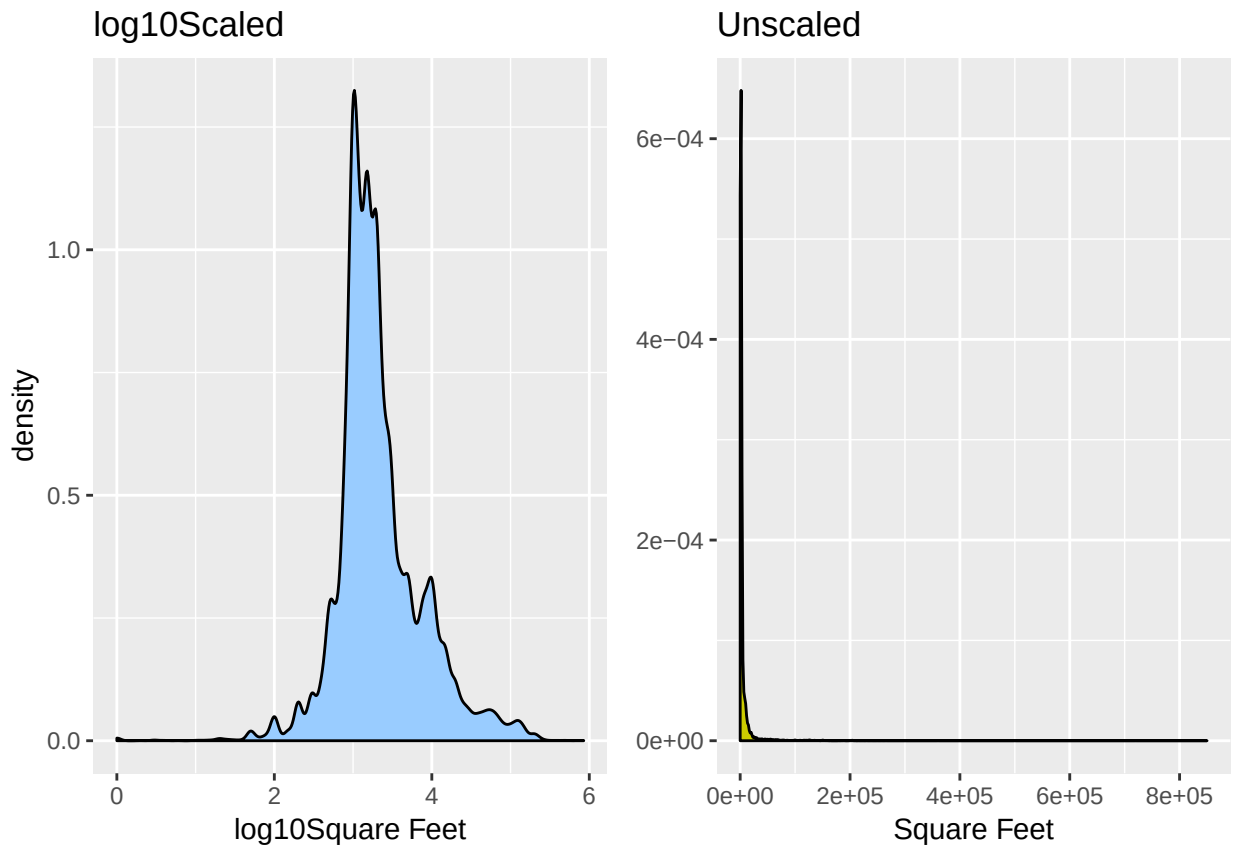
```
##  Square Footage
##  Min.   :     0
##  1st Qu.:   600
##  Median :  1400
##  Mean   :  5078
##  3rd Qu.:  3000
##  Max.   :850000
```

We see from a summary the following:

- Some stores claim 0 sqft
- The median square footage is 1718 sqft
- The mean is 6276 sqft. The difference between this and the median indicates pull from outliers
- The max size is a whopping 850,000 sqft

Lets visualize the distribution:

```
p_unscaled <- ggplot(markets_frame) +
  geom_density(aes(x = `Square Footage`), fill = "#CCCC00") +
  ggtitle("Unscaled") +
  ylab(NULL) + xlab("Square Feet")

p_scaled <- ggplot(markets_frame) +
  geom_density(aes(x = log10(`Square Footage`)), fill = "#99CCFF") +
  ggtitle(paste0(expression(log10), "Scaled")) +
  xlab(paste0(expression(log10), "Square Feet"))

ggarrange(p_scaled, p_unscaled, nrow = 1)
```



We see from the plots above that:

- Square footage is roughly normally distributed around 3500 sqft
- The 850,000 square foot outlier really skews the plot.

Question 4

Who exactly are the outliers?

```
markets_frame %>% arrange(desc(`Square Footage`)) %>% select(`Square Footage`, `Entity Name`, ha_city) %
```

Square Footage	Entity Name	ha_city
850000	TOPS MARKETS LLC	LANCASTER
720000	G&C FOOD DISTRIBUTORS&BROKERS INC	SYRACUSE
500000	PRICE CHOPPER OPERATING CO INC	FULTON
500000	PRICE CHOPPER OPERATING CO INC	CANTON
400003	RONALD L ABBEY III	PAINTED POST
360000	395 FOODS INC	NORTHPORT
300000	BONDUELLE USA INC	OAKFIELD
265000	L&D ACQUISITION LLC	NAPLES
258000	WAL-MART STORES EAST LP	ALBANY
250000	TARGET CORPORATION	WATERTOWN

The largest food retailer in New York is Top Markets LLC. Of note is that none of these goliath sized retailers are located in New York City or any of its Boroughs.

Question 5

Which stores are the most represented overall? Which franchise has the most food outlets?

```
markets_frame %>% group_by(`Entity Name`) %>% count(sort = TRUE) %>% head(10) %>%
  ggplot() +
  geom_col(aes(x = reorder(`Entity Name`, -n), y = n), fill = "#99CCFF") +
  xlab("Store Owner") + ylab("Count") +
  theme(axis.text.x = element_text(angle = 90)) +
  geom_text(aes(label = n, x = `Entity Name`, y = n), vjust = -1) +
  ggtitle("Top 10 Stores by Number of Shops")
```

Of the stores above:

- There are 3 pharmacies: CVS, Walgreens, and Rite Aid
- There are 3 gas stations: Stewarts, Speedway, and United Refining (Kwik Fill)
- There are 3 discount stores: Dolgencorp (Dollar General), Dollar Tree, and Family Dollar
- There is one general convenience store: 7-eleven