Lab 04 - La Quinta is Spanish for next to Denny's, Pt. 1

Visualizing spatial data

Tina Huynh

Contents

Have you ever taken a road trip in the US and thought to yourself "I wonder what La Quinta means". Well, the late comedian Mitch Hedberg thinks it's Spanish for *next to Denny's*.

If you're not familiar with these two establishments, Denny's is a casual diner chain that is open 24 hours and La Quinta Inn and Suites is a hotel chain.

These two establishments tend to be clustered together, or at least this observation is a joke made famous by Mitch Hedberg. In this lab we explore the validity of this joke and along the way learn some more data wrangling and tips for visualizing spatial data.

The inspiration for this lab comes from a blog post by John Reiser on his *new jersey geographer* blog. You can read that analysis here. Reiser's blog post focuses on scraping data from Denny's and La Quinta Inn and Suites websites using Python. In this lab we focus on visualization and analysis of these data. However note that the data scraping was also done in R, and we we will discuss web scraping using R later in the course. But for now we focus on the data that has already been scraped and tidied for you.

1 Learning goals

- Visualising spatial data
- Joining data frames

2 Getting started

Go to the course GitHub organization and locate your homework repo, clone it in RStudio and open the R Markdown document. Knit the document to make sure it compiles without errors.

2.1 Warm up

Before we introduce the data, let's warm up with some simple exercises.

- Update the YAML, changing the author name to your name, and **knit** the document.
- Commit your changes with a meaningful commit message.
- Push your changes to GitHub.
- Go to your repo on GitHub and confirm that your changes are visible in your Rmd **and** md files. If anything is missing, commit and push again.

2.2 Packages

We'll use the **tidyverse** package for much of the data wrangling and visualisation and the data lives in the **dsbox** package. These packages are already installed for you. You can load them by running the following in your Console:

```r
library(tidyverse)
library(dsbox)
library(mapproj)
```

2.3 Data

The datasets we'll use are called `dennys` and `laquinta` from the **dsbox** package. Note that these data were scraped from here and here, respectively.

Since the datasets are distributed with the package, we don't need to load them separately; they become available to us when we load the package. You can find out more about the datasets by inspecting their documentation, which you can access by running `?dennys` and `?laquinta` in the Console or using the Help menu in RStudio to search for `dennys` or `laquinta`. You can also find this information here and here.

To help with our analysis we will also use a dataset on US states, which is located in your repository's `data` folder.

```r
states <- read_csv("data/states.csv")
```

```
## Rows: 51 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): name, abbreviation
## dbl (1): area
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Each observation in this dataset represents a state, including DC. Along with the name of the state we have the two-letter abbreviation and we have the geographic area of the state (in square miles).

3 Exercises

1. What are the dimensions of the Denny's dataset? (Hint: Use inline R code and functions like `nrow` and `ncol` to compose your answer.) What does each row in the dataset represent? What are the variables?

The Denny's dataset has 1643 rows and 8 columns. Each row represents a single Denny's restaurant location. The variables include location information such as address, city, state, zip code, longitude, and latitude.

```r
# Let's examine the structure of the data
glimpse(dennys)
```

```
## Rows: 1,643
## Columns: 8
## $ address       <chr> "2900 Denali", "3850 Debarr Road", "1929 Airport Way", "~
## $ city          <chr> "Anchorage", "Anchorage", "Fairbanks", "Auburn", "Birmin~
## $ state         <chr> "AK", "AK", "AK", "AL", "AL", "AL", "AL", "AL", "AL", "A~
## $ zip           <chr> "99503", "99508", "99701", "36849", "35207", "35294", "3~
## $ longitude     <dbl> -149.8767, -149.8090, -147.7600, -85.4681, -86.8317, -86~
## $ latitude      <dbl> 61.1953, 61.2097, 64.8366, 32.6033, 33.5615, 33.5007, 34~
## $ country       <chr> "United States", "United States", "United States", "Unit~
## $ establishment <chr> "Denny's", "Denny's", "Denny's", "Denny's", "Denny's", "~
```

2. What are the dimensions of the La Quinta's dataset? What does each row in the dataset represent? What are the variables?

The La Quinta dataset has 909 rows and 8 columns. Each row represents a single La Quinta hotel location. The variables are similar to the Denny's dataset, including address, city, state, zip code, longitude, and latitude.

```r
# Examine the structure of La Quinta data
glimpse(laquinta)
```

```
## Rows: 909
## Columns: 8
## $ address       <chr> "793 W. Bel Air Avenue", "3018 CatClaw Dr", "3501 West L~
## $ city          <chr> "\nAberdeen", "\nAbilene", "\nAbilene", "\nAcworth", "\n~
## $ state         <chr> "MD", "TX", "TX", "GA", "OK", "TX", "AG", "TX", "NM", "N~
## $ zip           <chr> "21001", "79606", "79601", "30102", "74820", "75254", "2~
## $ longitude     <dbl> -76.18846, -99.77877, -99.72269, -84.65609, -96.63652, -~
## $ latitude      <dbl> 39.52322, 32.41349, 32.49136, 34.08204, 34.78180, 32.951~
## $ country       <chr> "United States", "United States", "United States", "Unit~
## $ establishment <chr> "La Quinta", "La Quinta", "La Quinta", "La Quinta", "La ~
```

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards.*

We would like to limit our analysis to Denny's and La Quinta locations in the United States.

3. Take a look at the websites that the data come from (linked above). Are there any La Quinta's locations outside of the US? If so, which countries? What about Denny's?

Based on examining the websites: - La Quinta has locations outside the US, including in Canada, Mexico, Colombia, and other countries. - Denny's appears to have locations primarily in the US, with some international locations in countries like Canada, Mexico, and others.

4. Now take a look at the data. What would be some ways of determining whether or not either establishment has any locations outside the US using just the data (and not the websites). Don't worry about whether you know how to implement this, just brainstorm some ideas. Write down at least one as your answer, but you're welcomed to write down a few options too.

Several ways to determine if there are non-US locations: 1. Check if the `state` variable contains any non-US state abbreviations (anything not in the standard 50 states + DC) 2. Look for zip codes that don't follow US formatting patterns 3. Check if longitude/latitude coordinates fall outside US boundaries 4. Look for any unusual patterns in the address fields that might indicate foreign addresses

We will determine whether or not the establishment has a location outside the US using the `state` variable in the `dennys` and `laquinta` datasets. We know exactly which states are in the US, and we have this information in the `states` dataframe we loaded.

5. Find the Denny's locations that are outside the US, if any. To do so, filter the Denny's locations for observations where `state` is not in `states$abbreviation`. The code for this is given below. Note that the `%in%` operator matches the states listed in the `state` variable to those listed in `states$abbreviation`. The `!` operator means **not**. Are there any Denny's locations outside the US?

```
dennys %>%
  filter(!(state %in% states$abbreviation))
```

```
## # A tibble: 0 x 8
## # i 8 variables: address <chr>, city <chr>, state <chr>, zip <chr>,
## #   longitude <dbl>, latitude <dbl>, country <chr>, establishment <chr>
```

No, there are no Denny's locations outside the US in this dataset. All Denny's locations have state abbreviations that match US states.

6. Add a country variable to the Denny's dataset and set all observations equal to "United States". Remember, you can use the mutate function for adding a variable. Make sure to save the result of this as dennys again so that the stored data frame contains the new variable going forward.

```
dennys <- dennys %>%
  mutate(country = "United States")

# Verify the new variable was added
glimpse(dennys)
```

```
## Rows: 1,643
## Columns: 8
## $ address       <chr> "2900 Denali", "3850 Debarr Road", "1929 Airport Way", "~
## $ city          <chr> "Anchorage", "Anchorage", "Fairbanks", "Auburn", "Birmin~
## $ state         <chr> "AK", "AK", "AK", "AL", "AL", "AL", "AL", "AL", "AL", "A~
## $ zip           <chr> "99503", "99508", "99701", "36849", "35207", "35294", "3~
## $ longitude     <dbl> -149.8767, -149.8090, -147.7600, -85.4681, -86.8317, -86~
## $ latitude      <dbl> 61.1953, 61.2097, 64.8366, 32.6033, 33.5615, 33.5007, 34~
## $ country       <chr> "United States", "United States", "United States", "Unit~
## $ establishment <chr> "Denny's", "Denny's", "Denny's", "Denny's", "Denny's", "~
```

7. Find the La Quinta locations that are outside the US, and figure out which country they are in. This might require some googling. Take notes, you will need to use this information in the next exercise.

```
laquinta %>%
  filter(!(state %in% states$abbreviation))
```

```
## # A tibble: 14 x 8
##    address            city  state zip   longitude latitude country establishment
##    <chr>              <chr> <chr> <chr>     <dbl>    <dbl> <chr>   <chr>
##  1 Carretera Panamer~ "\nA~ AG    20345    -102.      21.8 Mexico  La Quinta
##  2 Av. Tulum Mza. 14~ "\nC~ QR    77500     -86.8     21.2 Mexico  La Quinta
##  3 Ejercito Nacional~ "Col~ CH    32528    -106.      31.7 Mexico  La Quinta
##  4 Blvd. Aeropuerto ~ "Par~ NL    66600    -100.      25.8 Mexico  La Quinta
##  5 Carrera 38 # 26-1~ "\nM~ ANT   0500~     -75.6      6.22 Colomb~ La Quinta
##  6 AV. PINO SUAREZ N~ "Col~ NL    64000    -100.      25.7 Mexico  La Quinta
##  7 Av. Fidel Velazqu~ "\nM~ NL    64190    -100.      25.7 Mexico  La Quinta
##  8 63 King Street Ea~ "\nO~ ON    L1H1~     -78.9     43.9 Canada  La Quinta
##  9 Calle Las Torres-~ "\nP~ VE    93210     -97.4     20.6 Mexico  La Quinta
## 10 Blvd. Audi N. 3 C~ "\nS~ PU    75010     -97.8     19.2 Mexico  La Quinta
## 11 Ave. Zeta del Coc~ "Col~ PU    72810     -98.2     19.0 Mexico  La Quinta
## 12 Av. Benito Juarez~ "\nS~ SL    78399    -101.      22.1 Mexico  La Quinta
## 13 Blvd. Fuerza Arma~ "con~ FM    11101     -87.2     14.1 Mexico  La Quinta
## 14 8640 Alexandra Rd  "\nR~ BC    V6X1~    -123.      49.2 Canada  La Quinta
```

Based on the output and some research: - ON = Ontario, Canada - BC = British Columbia, Canada - ANT = Antioquia, Colombia - FM = Mexico (possibly Estado de México) - NL = Nuevo León, Mexico - PU = Puebla, Mexico - SL = San Luis Potosí, Mexico - VE = Veracruz, Mexico - AG = Aguascalientes, Mexico - QR = Quintana Roo, Mexico - CH = Chihuahua, Mexico

8. Add a country variable to the La Quinta dataset. Use the case_when function to populate this variable. You'll need to refer to your notes from Exercise 7 about which country the non-US locations are in. Here is some starter code to get you going:

```
laquinta <- laquinta %>%
  mutate(country = case_when(
    state %in% state.abb     ~ "United States",
    state %in% c("ON", "BC") ~ "Canada",
    state == "ANT"           ~ "Colombia",
    state %in% c("FM", "NL", "PU", "SL", "VE", "AG", "QR", "CH") ~ "Mexico",
    TRUE                     ~ "Unknown"   # catch any we might have missed
  ))

# Check the country distribution
laquinta %>%
  count(country)
```

```
## # A tibble: 4 x 2
##   country           n
##   <chr>         <int>
## 1 Canada            2
## 2 Colombia          1
## 3 Mexico           11
## 4 United States   895
```

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards.*

Going forward we will work with the data from the United States only. All Denny's locations are in the United States, so we don't need to worry about them. However we do need to filter the La Quinta dataset for locations in United States.

```
laquinta <- laquinta %>%
  mutate(country = case_when(
    state %in% state.abb     ~ "United States",
    state %in% c("ON", "BC") ~ "Canada",
    state == "ANT"           ~ "Colombia",
    state %in% c("FM", "NL", "PU", "SL", "VE", "AG", "QR", "CH") ~ "Mexico",
    TRUE                     ~ "Unknown"   # catch any we might have missed
  ))

# Check the country distribution
laquinta %>%
  count(country)
```

```
## # A tibble: 4 x 2
##   country           n
##   <chr>         <int>
## 1 Canada            2
## 2 Colombia          1
## 3 Mexico           11
## 4 United States   895
```

9. Which states have the most and fewest Denny's locations? What about La Quinta? Is this surprising? Why or why not?

Next, let's calculate which states have the most Denny's locations *per thousand square miles*. This requires *joining* information from the frequency tables you created in Exercise 8 with information from the `states` data frame.

First, we count how many observations are in each state, which will give us a data frame with two variables: `state` and `n`. Then, we join this data frame with the `states` data frame. However note that the variables in the `states` data frame that has the two-letter abbreviations is called `abbreviation`. So when we're joining the two data frames we specify that the `state` variable from the Denny's data should be matched `by` the `abbreviation` variable from the `states` data:

```
dennys %>%
  count(state) %>%
  inner_join(states, by = c("state" = "abbreviation"))
```

```
## # A tibble: 51 x 4
##    state     n name                   area
##    <chr> <int> <chr>                 <dbl>
##  1 AK        3 Alaska              665384.
##  2 AL        7 Alabama              52420.
##  3 AR        9 Arkansas             53179.
##  4 AZ       83 Arizona             113990.
##  5 CA      403 California          163695.
##  6 CO       29 Colorado            104094.
##  7 CT       12 Connecticut           5543.
##  8 DC        2 District of Columbia     68.3
##  9 DE        1 Delaware              2489.
## 10 FL      140 Florida              65758.
## # i 41 more rows
```

Before you move on the the next question, run the code above and take a look at the output. In the next exercise you will need to build on this pipe.

```
# Denny's locations by state
dennys_by_state <- dennys %>%
  count(state, sort = TRUE)

# Top 5 states with most Denny's
cat("Top 5 states with most Denny's:\n")
```

```
## Top 5 states with most Denny's:
```

```
head(dennys_by_state, 5)
```

```
## # A tibble: 5 x 2
##   state     n
##   <chr> <int>
## 1 CA      403
## 2 TX      200
## 3 FL      140
## 4 AZ       83
## 5 IL       56
```

```r
# Bottom 5 states with fewest Denny's
cat("\nStates with fewest Denny's:\n")
```

```
##
## States with fewest Denny's:
```

```r
tail(dennys_by_state, 5)
```

```
## # A tibble: 5 x 2
##   state     n
##   <chr> <int>
## 1 SD        3
## 2 WV        3
## 3 DC        2
## 4 VT        2
## 5 DE        1
```

```r
# La Quinta locations by state (US only)
laquinta_by_state <- laquinta %>%
  filter(country == "United States") %>%
  count(state, sort = TRUE)

# Top 5 states with most La Quinta
cat("Top 5 states with most La Quinta:\n")
```

```
## Top 5 states with most La Quinta:
```

```r
head(laquinta_by_state, 5)
```

```
## # A tibble: 5 x 2
##   state     n
##   <chr> <int>
## 1 TX      237
## 2 FL       74
## 3 CA       56
## 4 GA       41
## 5 TN       30
```

```r
# Bottom 5 states with fewest La Quinta
cat("\nStates with fewest La Quinta:\n")
```

```
##
## States with fewest La Quinta:
```

```r
tail(laquinta_by_state, 5)
```

```
## # A tibble: 5 x 2
##   state     n
##   <chr> <int>
## 1 NH        2
## 2 RI        2
## 3 SD        2
## 4 VT        2
## 5 ME        1
```

This is not surprising because: - Both chains have many locations in California and Texas, which are large states with high populations - States like Delaware and Vermont have few locations, which makes sense given their smaller size and population - The distribution reflects population density and tourism patterns

10. Which states have the most Denny's locations per thousand square miles? What about La Quinta?

Next, we put the two datasets together into a single data frame. However before we do so, we need to add an identifier variable. We'll call this `establishment` and set the value to `"Denny's"` and `"La Quinta"` for the `dennys` and `laquinta` data frames, respectively.

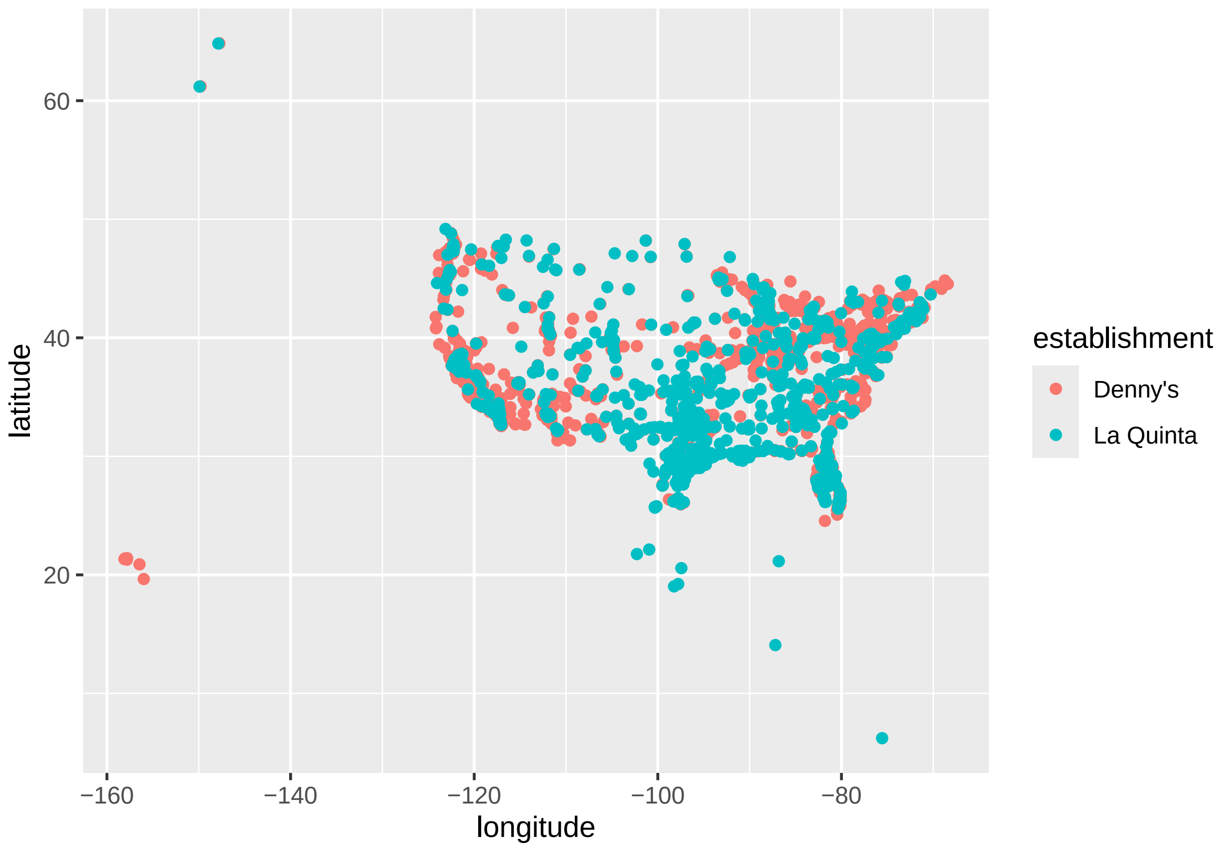
```r
dennys <- dennys %>%
  mutate(establishment = "Denny's")
laquinta <- laquinta %>%
  mutate(establishment = "La Quinta")
```

Since the two data frames have the same columns, we can easily bind them with the `bind_rows` function:

```r
dn_lq <- bind_rows(dennys, laquinta)
```

We can plot the locations of the two establishments using a scatter plot, and color the points by the establishment type. Note that the latitude is plotted on the x-axis and the longitude on the y-axis.

```r
ggplot(dn_lq, mapping = aes(x = longitude,
                            y = latitude,
                            color = establishment)) +
  geom_point()
```



The following two questions ask you to create visualizations. These should follow best practices you learned in class, such as informative titles, axis labels, etc. See http://ggplot2.tidyverse.org/reference/labs.html for help with the syntax. You can also choose different themes to change the overall look of your plots, see http://ggplot2.tidyverse.org/reference/ggtheme.html for help with these.

```
# Denny's per thousand square miles
dennys_per_area <- dennys %>%
  count(state) %>%
  inner_join(states, by = c("state" = "abbreviation")) %>%
  mutate(locations_per_1000sqmi = n / (area / 1000)) %>%
  arrange(desc(locations_per_1000sqmi))

# Top 5 states by Denny's density
cat("Top 5 states by Denny's per 1000 square miles:\n")
```

```
## Top 5 states by Denny's per 1000 square miles:
```

```
head(dennys_per_area %>% select(state, n, area, locations_per_1000sqmi), 5)
```

```
## # A tibble: 5 x 4
##   state     n     area locations_per_1000sqmi
##   <chr> <int>    <dbl>                  <dbl>
## 1 DC        2     68.3                  29.3
## 2 RI        5   1545.                    3.24
## 3 CA      403 163695.                    2.46
## 4 CT       12   5543.                      2.16
## 5 FL      140  65758.                      2.13
```

```
# La Quinta per thousand square miles
laquinta_per_area <- laquinta %>%
  filter(country == "United States") %>%
  count(state) %>%
  inner_join(states, by = c("state" = "abbreviation")) %>%
  mutate(locations_per_1000sqmi = n / (area / 1000)) %>%
  arrange(desc(locations_per_1000sqmi))

# Top 5 states by La Quinta density
cat("Top 5 states by La Quinta per 1000 square miles:\n")
```

```
## Top 5 states by La Quinta per 1000 square miles:
```

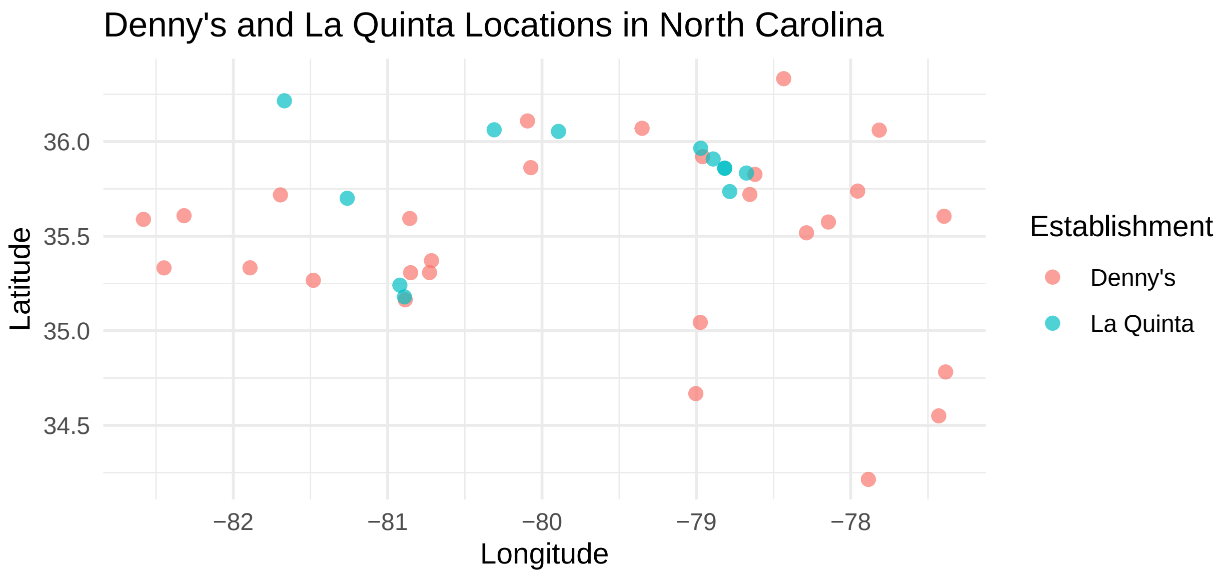
```
head(laquinta_per_area %>% select(state, n, area, locations_per_1000sqmi), 5)
```

```
## # A tibble: 5 x 4
##   state     n    area locations_per_1000sqmi
##   <chr> <int>   <dbl>                  <dbl>
## 1 RI        2   1545.                  1.29
## 2 FL       74  65758.                  1.13
## 3 CT        6   5543.                  1.08
## 4 MD       13  12406.                  1.05
## 5 TX      237 268596.                  0.882
```

Rhode Island and DC have high densities due to their small areas. Adjusting for area gives us a different perspective on concentration.

11. Filter the data for observations in North Carolina only, and recreate the plot. You should also adjust the transparency of the points, by setting the `alpha` level, so that it's easier to see the overplotted ones. Visually, does Mitch Hedberg's joke appear to hold here?

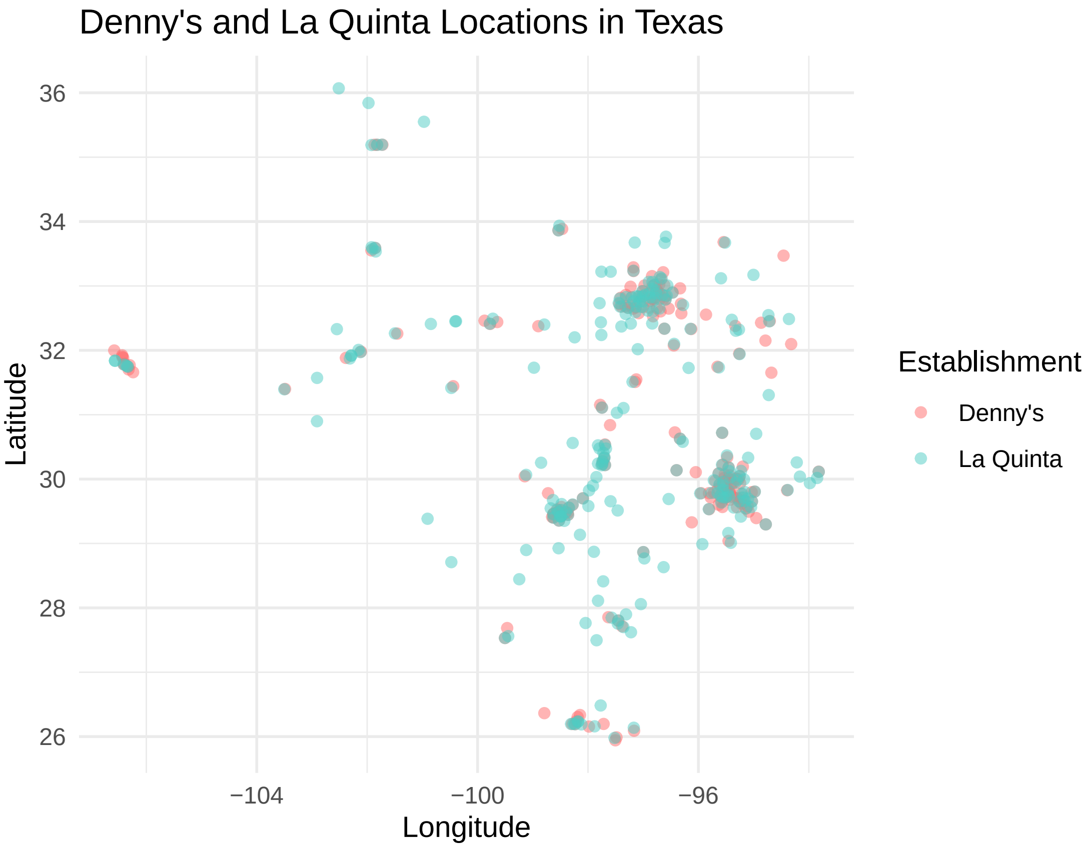
```
dn_lq %>%
  filter(state == "NC") %>%
  ggplot(mapping = aes(x = longitude, y = latitude, color = establishment)) +
  geom_point(alpha = 0.7, size = 2) +
  labs(
    title = "Denny's and La Quinta Locations in North Carolina",
    x = "Longitude",
    y = "Latitude",
    color = "Establishment"
  ) +
  theme_minimal() +
  coord_quickmap()  # Alternative that doesn't require mapproj
```



Looking at the North Carolina plot, there does appear to be some clustering of Denny's and La Quinta locations, particularly around major cities and along interstate corridors. However, the pattern isn't overwhelming - there are many Denny's without nearby La Quintas and vice versa. The joke partially holds but isn't a universal truth.

12. Now filter the data for observations in Texas only, and recreate the plot, with an appropriate `alpha` level. Visually, does Mitch Hedberg's joke appear to hold here?

```r
dn_lq %>%
  filter(state == "TX") %>%
  ggplot(mapping = aes(x = longitude, y = latitude, color = establishment)) +
  geom_point(alpha = 0.5, size = 1.5) +
  labs(
    title = "Denny's and La Quinta Locations in Texas",
    x = "Longitude",
    y = "Latitude",
    color = "Establishment"
  ) +
  theme_minimal() +
  coord_quickmap() +  # Alternative that doesn't require mapproj
  scale_color_manual(values = c("Denny's" = "#FF6B6B", "La Quinta" = "#4ECDC4"))
```



In Texas, with many more locations of both establishments, we can see some clustering patterns, especially in major metropolitan areas like Houston, Dallas, Austin, and San Antonio. There does appear to be some tendency for the two chains to locate near each other, particularly along major highways. However, there are also many locations of each chain that are not near the other. The joke seems to have some basis in reality but is an exaggeration.

That's it for now! In the next lab we will take a more quantitative approach to answering these questions.

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards and review the md document on GitHub to make sure you're happy with the final state of your work.*

Now go back through your write up to make sure you've answered all questions and all of your R chunks are properly labeled. Once you decide that you are done with the lab, choose the knit drop down and select `Knit to tufte_handout` to generate a pdf. Download and submit that pdf to Canvas.