Assessment 4

Exploratory Data Analysis and Visualization

Ziyang Lin

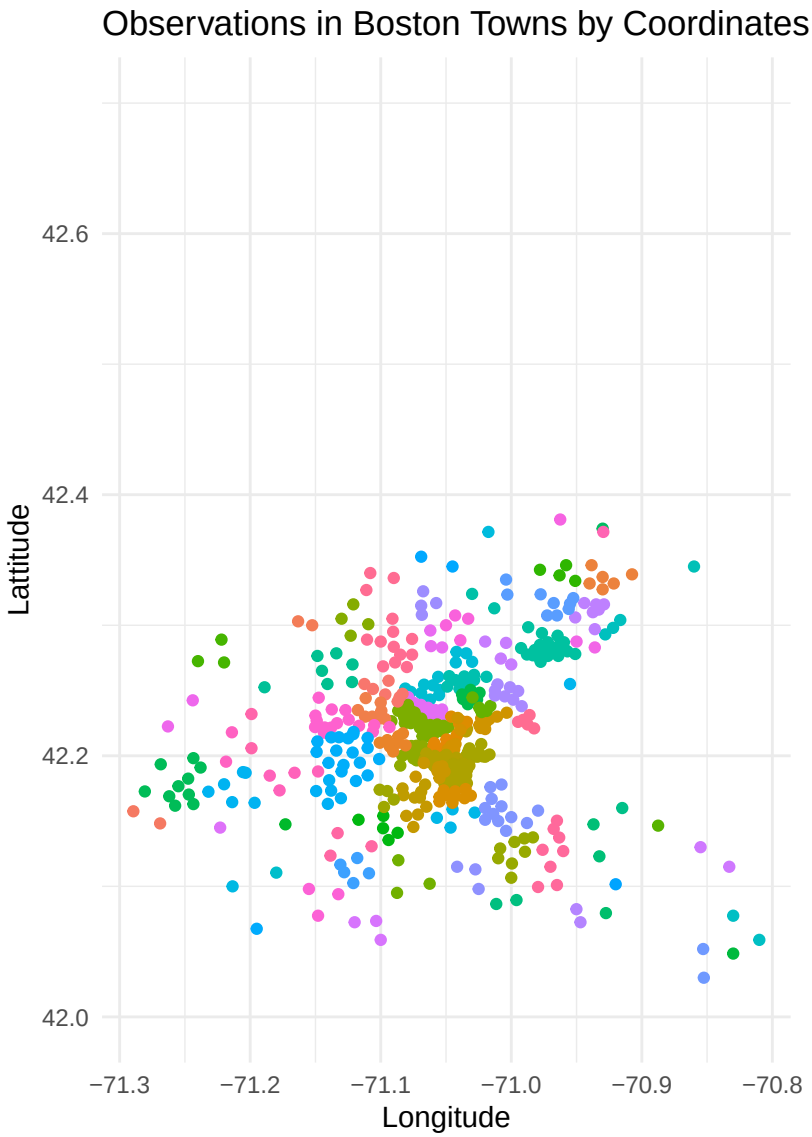
Mar 15, 2022

Question 1

(a).

We first load the data set and select the required columns, then construct a visualization to plot data points in a map of Boston towns:

```
BostonData <- boston.c %>%
  select(TOWN, LON, LAT, CMEDV) # select only required columns
ggplot(data=BostonData, aes(x=LON, y=LAT)) +
  geom_point(aes(col=TOWN)) + # plot town location by LON and LAT
  ggtitle("Observations in Boston Towns by Coordinates") +
  coord_equal(ylim=c(42.0, 42.7)) + theme_bw() +
  scale_size(range=c(0.5, 3.5)) + theme_minimal() +
  theme(legend.position="none") +
  labs(x="Longitude", y="Lattitude")
```



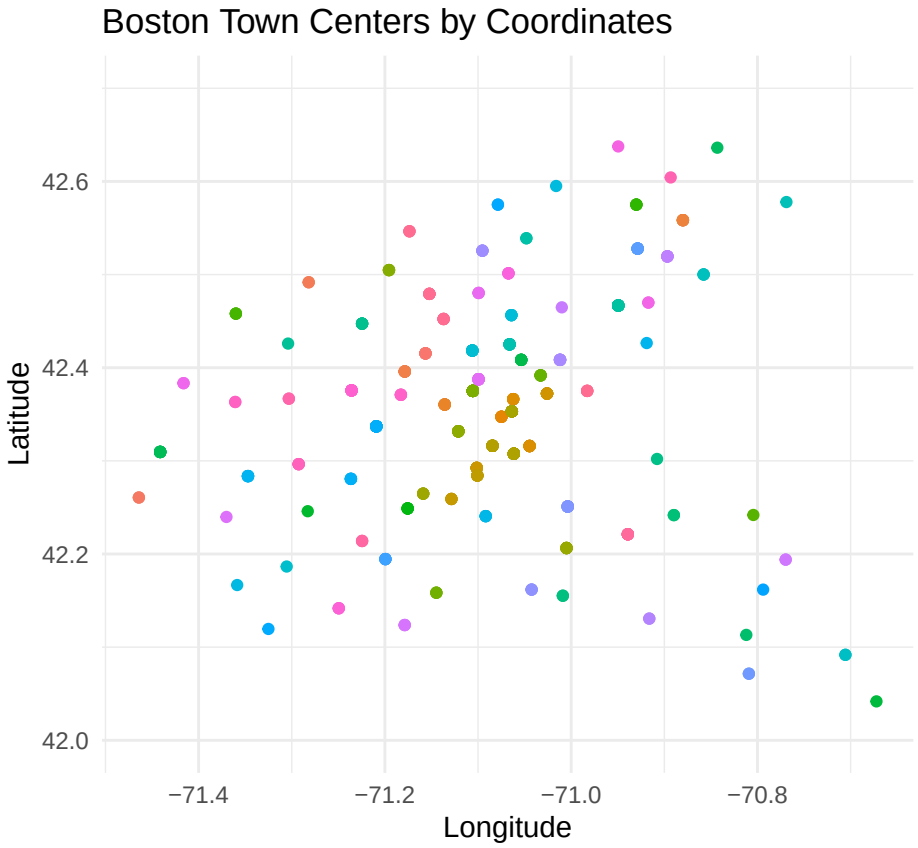
If the town coordinates are specified correctly, then we would expect the observations in the same town appear to be clustered together. However, this is not the case in our constructed figure as we see points with different colors (that are from different towns) are stacked together, which means that the coordinates that are given in the dataset must be incorrect. Note that the legend for towns is hidden since it is unable to fit in.

(b).

Now we load the correct coordinates, and then join with `BostonData` to get a new data frame that contains the wrong coordinates and the true town center coordinates. The visualization of the town centers is as follow.

```
boston.tc <- read.csv("~/Desktop/BostonTownCentres.csv")
names(boston.tc) <- c("TOWN", "LAT_CTR", "LON_CTR")
boston.join <- left_join(boston.tc, BostonData, by="TOWN")
ggplot(data=boston.join, aes(x=LON_CTR, y=LAT_CTR)) +
  geom_point(aes(col=TOWN)) + # plot town location by LON and LAT
  ggtitle("Boston Town Centers by Coordinates") +
  coord_equal(ylim=c(42.0, 42.7)) + theme_bw() +
```

```
  scale_size(range=c(0.5, 3.5)) + theme_minimal() +
  theme(legend.position="none") +
  labs(x="Longitude", y="Latitude")
```



These centers seem deviated from the centroids of observations in each town as in the previous figure.

(c).

Next, we average out the `LON` and `LAT` from `boston.c` to get the centroid for each town, and then according to the true town center coordinate, we add the displacement back to each observation. The two new columns with corrected coordinates are `LAT_C` and `LON_C`.

```
boston.join$LON_C <- numeric(nrow(boston.join))
boston.join$LAT_C <- numeric(nrow(boston.join))
for (i in unique(boston.join$TOWN)) {
  wrong_centroid <- c(mean(boston.join[boston.join$TOWN==i,]$LAT),
                      mean(boston.join[boston.join$TOWN==i,]$LON)) # get mean of lon and lat
  true_centroid <- c(mean(boston.join[boston.join$TOWN==i,]$LAT_CTR),
                     mean(boston.join[boston.join$TOWN==i,]$LON_CTR)) # get true town centers

  displacement <- true_centroid - wrong_centroid # compute displacement
  # add the displacement back to match true centroid
  boston.join[boston.join$TOWN==i,]$LAT_C <-
    boston.join[boston.join$TOWN==i,]$LAT + displacement[1]
  boston.join[boston.join$TOWN==i,]$LON_C <-
    boston.join[boston.join$TOWN==i,]$LON + displacement[2]
}
```

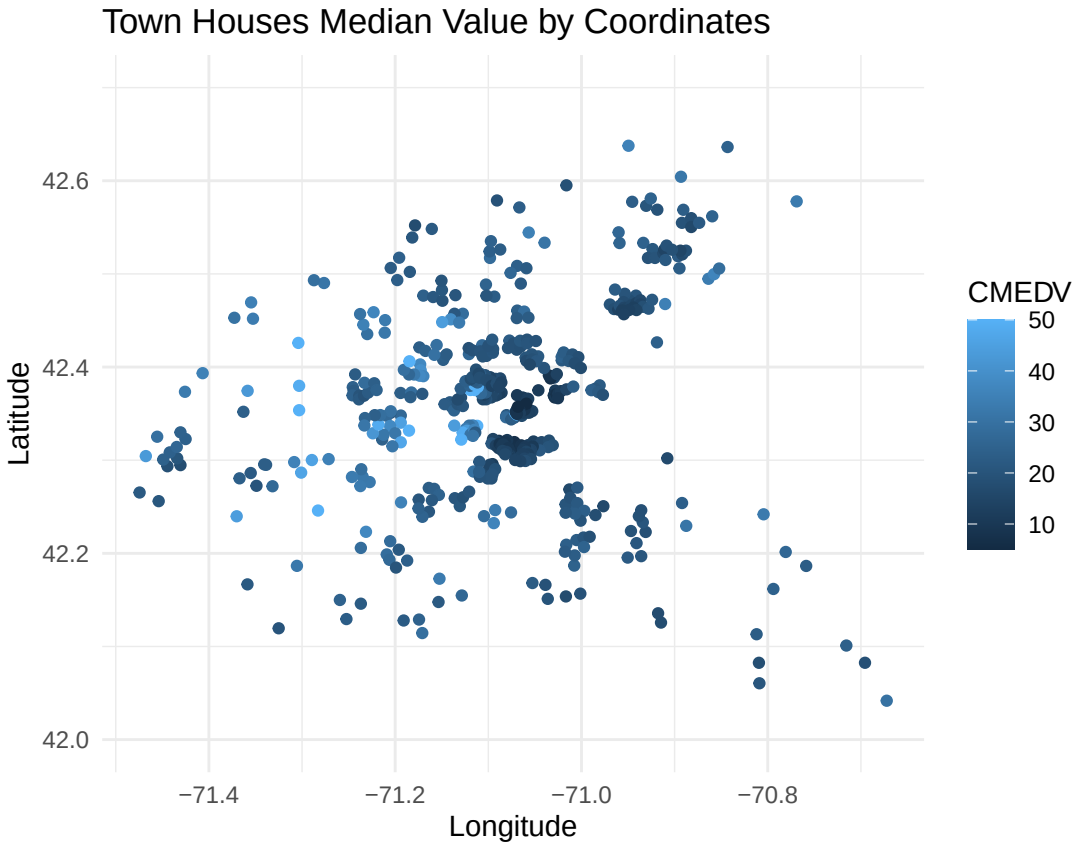
```r
head(boston.join)
```

```
##        TOWN  LAT_CTR   LON_CTR      LON     LAT CMEDV     LON_C    LAT_C
## 1 Arlington 42.41537 -71.15644 -71.0870 42.2416  23.1 -71.14847 42.40772
## 2 Arlington 42.41537 -71.15644 -71.0855 42.2450  23.6 -71.14697 42.41112
## 3 Arlington 42.41537 -71.15644 -71.0833 42.2475  22.6 -71.14477 42.41362
## 4 Arlington 42.41537 -71.15644 -71.0940 42.2575  29.4 -71.15547 42.42362
## 5 Arlington 42.41537 -71.15644 -71.1125 42.2550  23.2 -71.17397 42.42112
## 6 Arlington 42.41537 -71.15644 -71.1060 42.2512  24.6 -71.16747 42.41732
```

(d).

Finally, we construct a visualisation that shows the spatial distribution of the median value of owner-occupied housing in Greater Boston in 1970 using the corrected coordinates.

```r
ggplot(data=boston.join, aes(x=LON_C, y=LAT_C)) +
  geom_point(aes(col=CMEDV)) + # plot town location by LON and LAT+
  coord_equal(ylim=c(42.0, 42.7)) + theme_bw() +
  scale_size(range=c(0.5, 3.5)) + theme_minimal() +
  theme(legend.position="right") +
  labs(
    x="Longitude", y="Latitude", title="Town Houses Median Value by Coordinates"
  )
```



Question 2

The data set is `us_states_df` from `spData` package, we construct the choropleth the same way as we did in Lab 6:

```
data("us_states_df")
states_map <- map_data("state") # get map data
us_states_df$`Median Income` <- us_states_df$median_income_10
states_data <- as.data.frame(us_states_df)
states_data$region <- tolower(states_data$state) # match column name
fact_join <- left_join(states_map, states_data, by="region") # join data set for a single plot
ggplot(fact_join, aes(x=long, y=lat, group=group)) +
  geom_polygon(aes(fill=`Median Income`), color="white") +
  labs(
    x="Longitude", y="Lattitude",
    title="Median Income of 2010 by States"
  )
```