CSC 465 Final Project: "Magnificent Seven"

Introduction:

The City of Chicago is known for many qualities such as architecture, arts, culture, sports teams, but perhaps among the unique qualities are the "neighborhoods" of Chicago. Chicago neighborhoods are much more than just names, boundaries, and landmarks but are expressions of the diverse cultures and ethnicities of the makeup. One of the many ways by which this is expressed is in the variety of foods available. For our project, we chose to look "behind the scenes" of Restaurant Week, the Taste of Chicago, and other celebrations of Chicago's cuisines into food inspection data as maintained by the Chicago Department of Public Health.

Through the Chicago Health Atlas, we were able to obtain over 160,000 rows of restaurant inspection data spanning seven plus years. The set contains qualitative data to include Name, Facility Type, Address, Risk Level, Violations, and Results, and geographical data to include Address, Latitude / Longitude, Census Tract, Community Areas, and Wards. Quantitative data can be derived by calculating the number of inspections, pass / fail frequency and rates, inspection types, and risk levels.

We posed questions such as, "what restaurants have the most failures?", "what area of the City has the most passing food establishments?", "do daycare centers frequently violate food handling rules?" and other questions too numerous to list but overall, we want to inform the audience of the state of food safety here in Chicago.

The following terms are referenced throughout our project and the definitions are provided for common understanding:

Risk Category – restaurants are classified into 3 risk categories with 1 being the highest and 3 being the lowest. The "risk" is defined by the complexity of the food prep where serving pre-packaged boxed foods purchased from vendors constitutes low risk and preparing food from scratch, on site, constitutes high risk. The risk level also determines frequency of inspections where Risk 1 receive more frequent inspections than Risk 3

Inspection Type – the inspection types are classified as "canvass", the most common type of inspection; "consultation", where the owner requests an inspection prior to opening; "complaint", which is performed in response to complaint(s) against the facility; "license", required prior to the facility receiving its license to operate; "suspect food poisoning", performed when complaints of illness are received; and "task-force inspection", when the inspection is performed on a bar or tavern.

Results – the results of inspection can be "Pass", "Pass with Condition", or "Fail". Pass is defined as not receiving any critical or serious violations (violations 1 – 14 and 15 – 29); pass with conditions is defined as those establishments with critical or serious violations but were corrected during the inspection; and fail is defined as those with critical or serious violations and did not correct for them during the inspection. Other results can be "Out of Business" and "Not Located" which are self-explanatory.

Exploratory Analysis:

The initial explorations of the data revealed several things requiring attention as part of cleaning the data to include incomplete entries, misspelled words, missing locations, and absent results. We used high level explorations in R to show any portions of missing or incomplete variables.

Through the exploration, we identified over 150 unique "Business Types" that are comparable. For example, the entries of "taverns", "brewery", "TAVERN", "bar", all seem to represent a similar group of facilities. Table 1.1 shows some of the top results by type from the original dataset. We condensed the

variations and coded each into 16 general “Business Types” as follows: “Hotels”, “Theaters”, “Farms”, “Bars”, “Food Halls”, “Events”, “Hospitals”, “Kitchens”, “Adult Daycare”, “Child Daycare”, “Food Trucks”, “Storefronts”, “Schools”, “Restaurants”, “Kitchen”, and “Other”. These newly created categories provided for a better understanding of the overall breakdown.

As part of the exploratory analysis, we looked for gaps in time as a line graph, Figure 1.1. The line graph shows the total inspections by month in 2017 with a significant decline in July. Upon further exploration, this decline is reflective of the high numbers of inspections taking place in schools and July represents the first full month of summer vacation.

Following examining the data by time, we explored the data by geography. Figure 1.2 exhibits a Tableau map indicating count of total food inspections per zip code within the city of Chicago with the darker the color and larger the size representing the higher counts. We can see the majority of the inspections of 2017 centered on the central part of Chicago. The team also decided at this point to frame the visualizations around the City of Chicago and omit those zip codes outside the city limits.

	Facility_Type <fctr>	Total_Inspections <int>
182	TAVERN	19
68	Daycare (Under 2 Years)	16
86	GAS STATION	16
133	Mobile Frozen Desserts Vendor	12
25	BREWERY	11
180	tavern	8
192	Wholesale	8
64	DAYCARE	7
170	Shared Kitchen User (Long Term)	7
14	BANQUET	6

21–30 of 195 rows Previous

Table 1.1

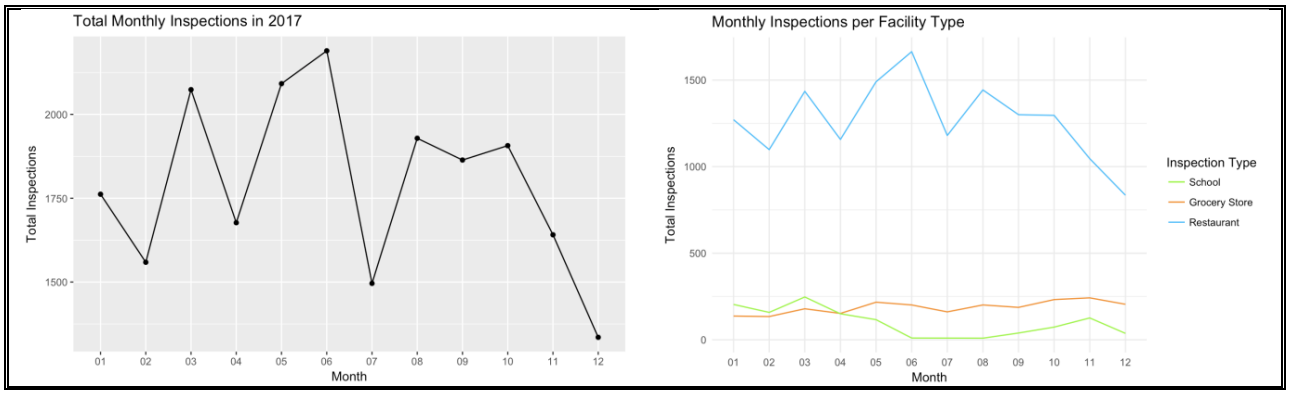


Figure 1.1

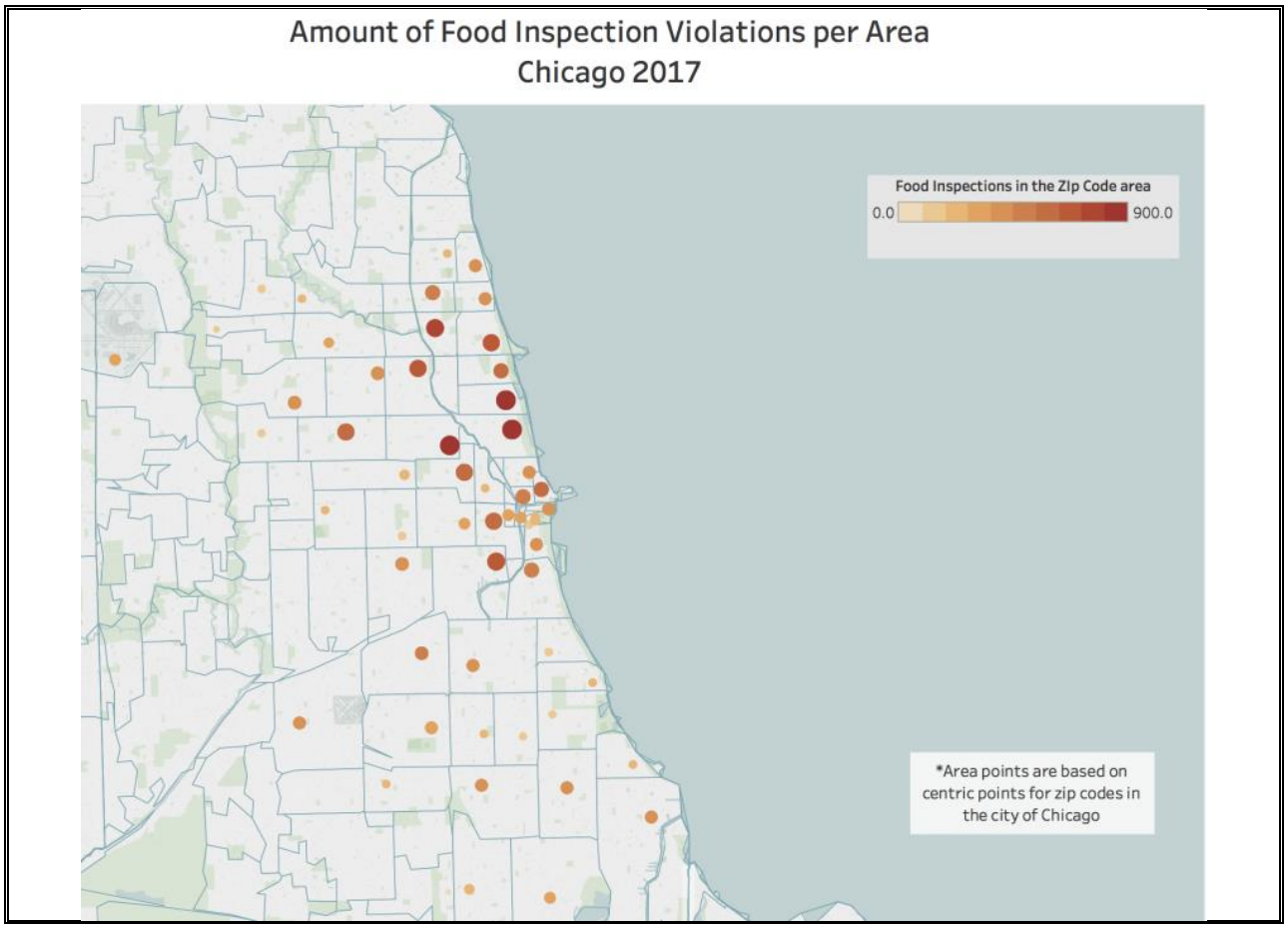


Figure 2.1

Visualizations

Stacked Bar Chart on a Radial Axis

To jumpstart the analysis, it is necessary to show how the newly created classes break down. The stacked bar graph in Figure 2.1 demonstrates the newly created categories (business types) and their corresponding count of food inspections and results filtered by "completed" food inspections as indicated by "Fail", "Pass", and "Pass with Conditions". We utilized R / ggplot and several R functions due to the flexibility of the program to represent the data in a more catching and please fashion. The final categorical variables are represented as individual bars with corresponding values in logarithmic scale, represented on polar coordinates to create this visual.

The colors chosen show the contrast between failed and passed inspections. Because the visual is meant to only portray how the business types vary and how differently made up they are, the scale isn't shown as part of the final visual. It has to be noted that this bar graph does show that there is a slight trend of comparable proportions in the breakdown of results, as portrayed by the colors of each bar indicating that there is, perhaps, a trend to how these inspections come about.

As noted in the first steps of the exploratory analysis, there were several business types that make up most of the data and some that are not represented at all. Even after the data was modified to more condensed "umbrella" business types, some were severely misrepresented in amount. Due to that, several business types were chosen to be explored further.

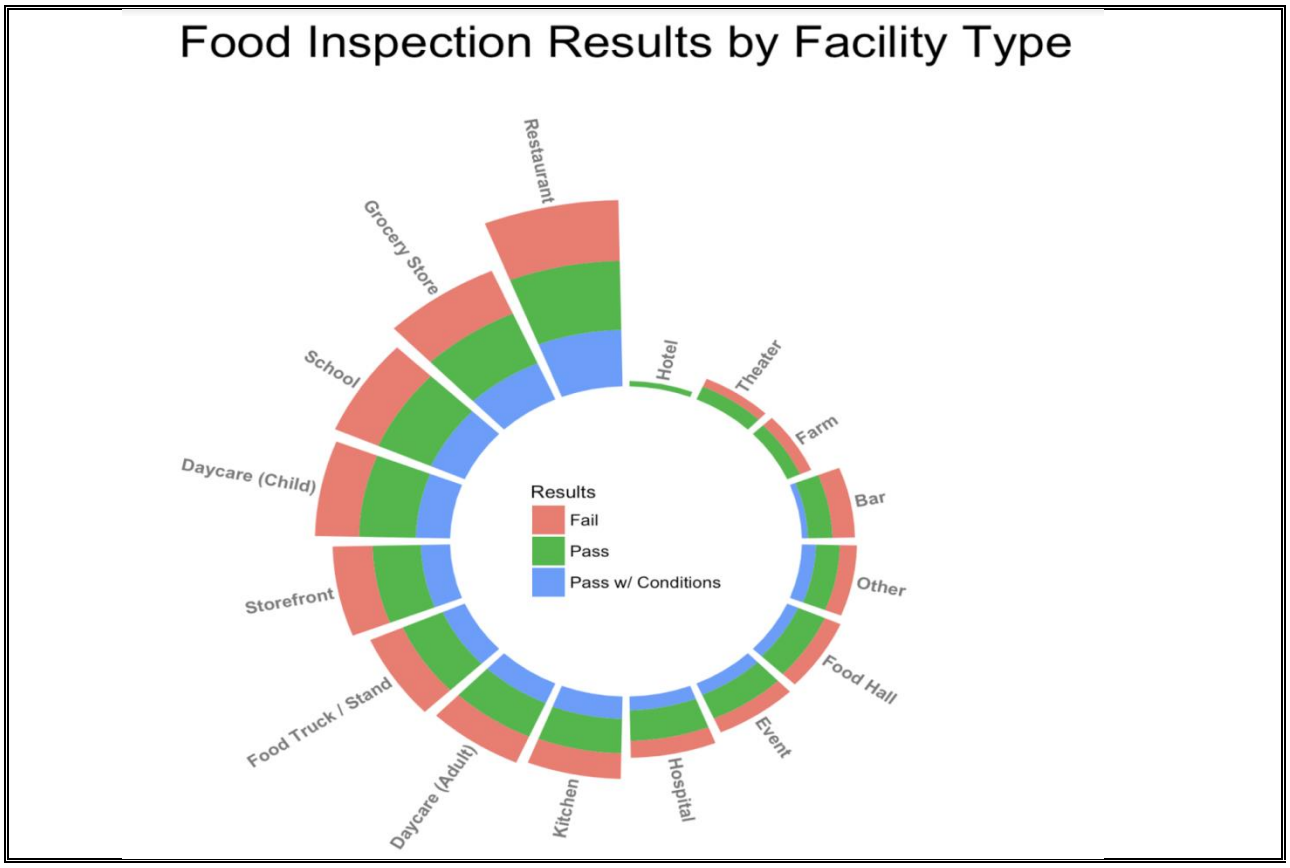


Figure 2.1

Heat Map

In line with the exploratory analyses and to complement the bar graph, we created a heat map to visualize potential variances in pass/fail rates by facility type and seasonality (i.e. by time). The heat map (generated in Tableau) above exhibits the average results coded as 1 for "Pass", 1.5 for "Pass w/ Conditions" and 2 for "Fail". The results are broken out by month on the columns and facility type on the rows. Those facility types with average at or near 1.0 and below 1.25 had higher percentage of facilities with a passing rate; between 1.25 and below 1.75 had on average conditional pass rate; and those above 1.75 had more facilities with a high fail rate. The color codes the results in a traffic light pattern with green for pass, gold for conditional pass and red for fail. This provides another layer for the general audience to visually identify the pass/fail average rate labels quickly upon observing the heat map.

A first glance at the map reveals there are 4 instances across 3 facility types and 3 months in which the average rate was 100% fail. One point of interest that stands out in the heat map is that "Bars" appear to have the most questionable food inspections given the frequency of monthly averages in gold and red, with February and October averaging 100% fail rates. A closer review of the data reveals that each instance of fail rates in February and October were inspection and re-inspection of the same facility in

which both respective facilities failed in both instances. Moreover, of the 38 inspections for bars, 16 resulted in fails – a 42% fail rate!

Whereas "Bars" had 100% fail rates in February and October, "Food Halls" (e.g. banquet halls, cafeterias, etc.) are Tier 1 facilities and exhibited a high fail rate in July. In the "Food Halls" category, 8 of 44, or 18%, inspections resulted in fails. However, Food Halls also had 5 months in which they achieved 100% pass rate (average at 1.0). July appears be an anomaly based on prevalent rate of other months. A deeper look into the detail for July reveals only one inspection was completed in this facility type and an outlier.

The "Event" group also had a 100% fail rate in October despite a relatively stable average pass rate for all other months. As with "Food Halls", a deeper look into the detail revealed an outlier as the fail rate is based on one observation. "Hotels" has only 2 observations which are reflected in the two points on the heat map in February and March. Unfortunately, the lack of data for hotels does not allow for a valuable assessment of the pass / fail rates for hotels. The higher frequency facility types such as "Restaurants", "Schools", and "Daycares" are also of interest to most audiences due to their high traffic and frequency of use. Fortunately, these facility types reveal higher pass rates.

There does not appear to be much time effect on the level of facility type with respect to months with higher pass or fail rates. This view would imply that the months have little correlation with the pass / fail rates. As we have noted with the "Food Halls" and "Bar" examples, facility type seems to have an influence on the pass / fail rate. In short, it appears that certain facilities tend to have higher pass / fail rates. It would be of particularly interest to examine the relationship between facility and zip code / area / neighborhood to gauge if location has any impact to pass / fail rates.

Food Inspections Results by Facility Type

	Inspection Date											
Facility Type Group	January	February	March	April	May	June	July	August	September	October	November	December
Bar	1.400	2.000	1.333	1.000		1.500	1.000	1.500	1.333	2.000	1.429	1.000
Daycare (Adult)	1.400	1.417	1.150	1.375	1.063	1.217	1.375	1.304	1.260	1.391	1.136	1.333
Daycare (Child)	1.217	1.216	1.189	1.223	1.198	1.165	1.316	1.189	1.200	1.187	1.256	1.252
Event			1.000	1.125	1.200	1.375	1.150	1.250	1.250	2.000	1.000	
Farm		1.500			1.500		1.333	1.500				1.250
Food Hall	1.000	1.000	1.000	1.214	1.400	1.000	2.000	1.167	1.333	1.250	1.000	1.200
Food Truck / Stand	1.250	1.325	1.344	1.225	1.409	1.245	1.100	1.444	1.400	1.091	1.000	1.286
Grocery Store	1.281	1.348	1.289	1.332	1.262	1.298	1.338	1.272	1.295	1.315	1.250	1.288
Hospital	1.167		1.000	1.500	1.167	1.263		1.150	1.200	1.100	1.250	1.000
Hotel		1.000	1.000									
Kitchen	1.286	1.188	1.350	1.300	1.333	1.237	1.136	1.421	1.286	1.231	1.222	1.000
Restaurant	1.269	1.283	1.281	1.299	1.280	1.292	1.292	1.287	1.297	1.288	1.288	1.260
School	1.281	1.300	1.288	1.264	1.227	1.000		1.370	1.272	1.219	1.165	1.236
Storefront	1.342	1.240	1.333	1.448	1.350	1.275	1.266	1.220	1.256	1.372	1.229	1.130
Theater			1.000			1.500		1.500	1.333	1.000		

Avg. Results Coded: 1.000 – 2.000

Average of Results Coded (color) broken down by Inspection Date Month vs. Facility Type Group. The data is filtered on Results Coded, which ranges from 1 to 2. The view is filtered on Facility Type Group, which excludes Null and Other.

Figure 3.1

Stacked Bar Graph of the Top Restaurants (by number of locations)

Food inspection does lead one to think "restaurants" and as students, we do find ourselves dining out for convenience, time, and price. We decided to look into the 20 most common restaurants as defined by the number of different locations and sorted them by names. This 100% stacked bar chart shows the

percentage of each result (including pass, pass with conditions, fail). The green indicates the pass rate; the yellow indicates the pass with condition rate; and the red shows the fail rate. Then they are sorted by pass rate. The result shows that Starbucks in Chicago area has the highest pass rate, of almost 80%, followed by Corner Bakery Café and Potbelly Sandwich. Restaurants like KFC, McDonald's, Subway, Dunkin Donuts, etc., are ranked in the middle. While Domino's Pizza and 7-Eleven have the lowest pass rate at 56%.

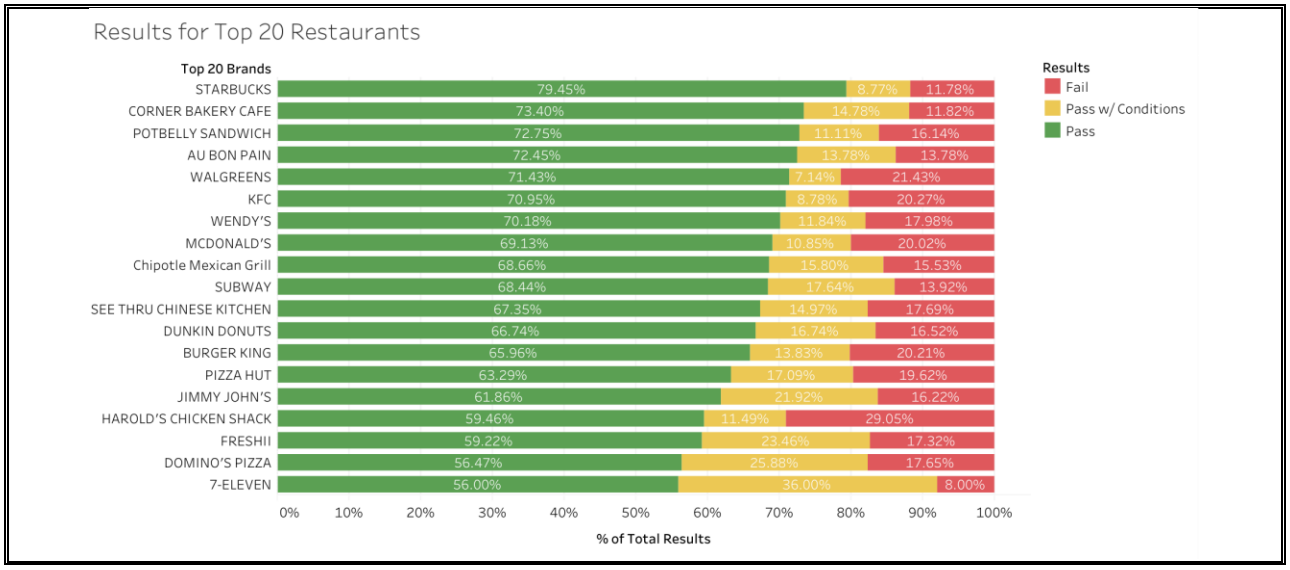


Figure 4.1

Force Directed Graph/Network Graph

In addition to finding out the pass / fail rate of some of the more popular restaurants / cafes, we thought it would be interesting to look at the descriptions of the violations to figure out "why?". We decided to use a "Force Directed" graph to describe the network and connections. Each node in Figure 5.1 represents a word used in the descriptive part of the restaurant inspection violations and each edge reveals the connection between two nodes. The thicknesses of the edges represent the total number of times that two words appeared together and thus the strengths of connection.

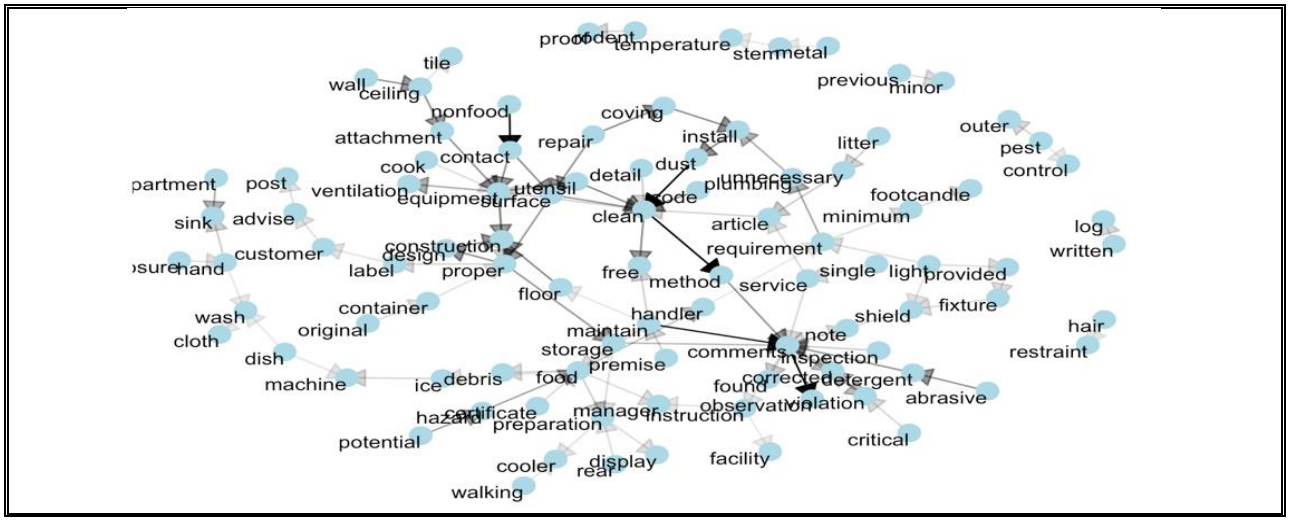


Figure 5.1

We can see that there are five small centers (“equipment”, “clean”, “construction”, “comments”, “food”) that obviously have much more edges connected with other nodes, which means they are keywords and violations are mostly related to them. At the same time, these centers and nodes make up of five small groups which deliver more precise information from different perspectives.

Directed edges helps us better understand these relationships/information. For example, the group with “equipment” at the center may reveal that equipment related to cooking (like cooking utensil) and kitchen (like wall, ceiling) should receive greater attention as it’s often cited. For the group with “comment” at the center, we may get additional information which is beyond regular violation rules, like service and observations about facilities that need to be corrected.

This network graph tells detailed information about violations among words. According to this information, restaurant violations can be improved to include but not limited to the following: Cleaning should be detail-oriented using correct methods; care of maintenance and service should be taken beyond cleaning; proper food preparation and disposal methods should be followed; and restaurant construction and equipment need to be routinely repaired and maintained.

Choropleth of Daycare Facilities

When our group decided on using food inspection data, the initial thoughts went to restaurants we frequent and other similar venues. Since our exploratory analysis indicated interesting results for the category “Daycare” and assuming choosing the best daycare is of great concern for parents, we decided to dive deeper into this subset. While we were fairly confident parents take into account the reputation of the daycare, the philosophical approach of the center, convenience of location, and perhaps religious and ethnic factors, we were not as confident as to whether or not parents consider food handling / food safety records as part of their decision.

As mentioned in the introduction, our chosen data set required cleaning to consolidate all those categories that fit into the more general category of “Daycare” and as with the previous analyses, we framed the visual to include those with definitive results within the geographical boundaries of the City of Chicago. We chose to represent the percentage of “Pass” and “Fail” using calculation functions within Tableau as a chloropleth in figure 4.1.

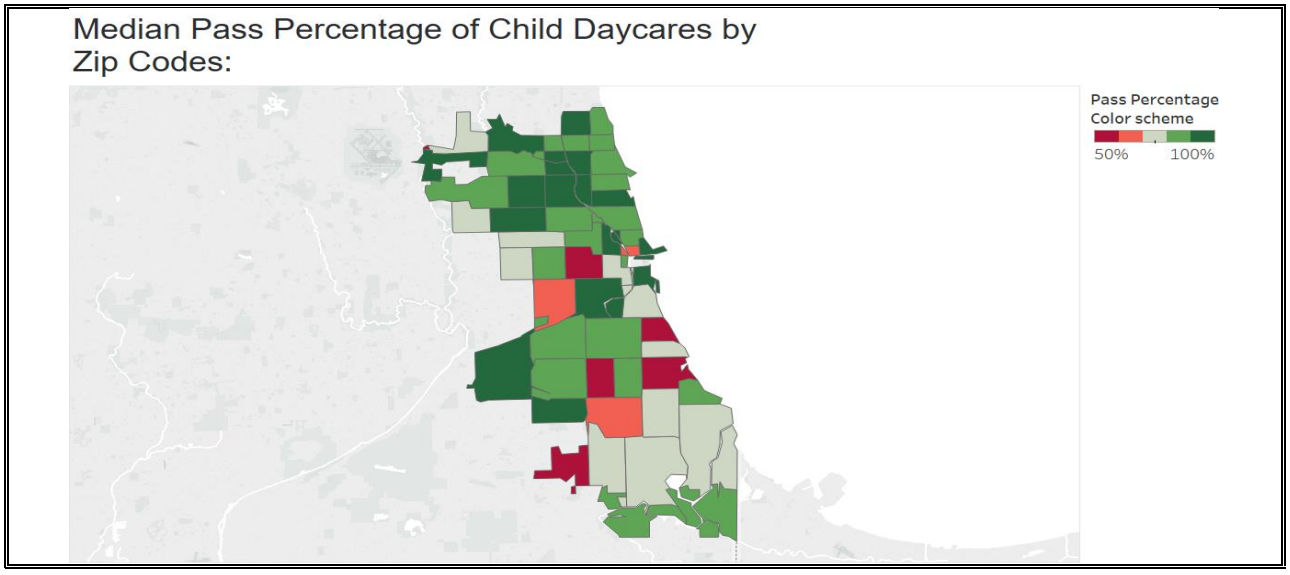


Figure 6.1

Figure 6.1 shows the median inspection passing rates of the daycare facilities by zip code in the city of Chicago. We chose the Red-Green Diverging color scheme to provide emphasis on the pass / fail rates to the viewer. Red indicates median of less than 60% passing rates while green median of greater than 90% passing rates.

This visual may be used by families to consider, among other factors, the target neighborhood(s) to which to relocate.

Treemap of Violations for Failed Adult Daycare Facilities

Adult Daycare facilities was considered an interesting population to look into to further to understand why these types of facilities in Chicago would fail food inspections. All Adult Daycare facilities have a Risk Level 1 (highest risk) given that food is usually prepared in the facility. Other concerns regarding this population are related to the fact that the population as a whole is aging, and Adult Daycare facilities are growing in number and importance in our country as the "Baby Boomer" generation begins to depend on these types of facilities.

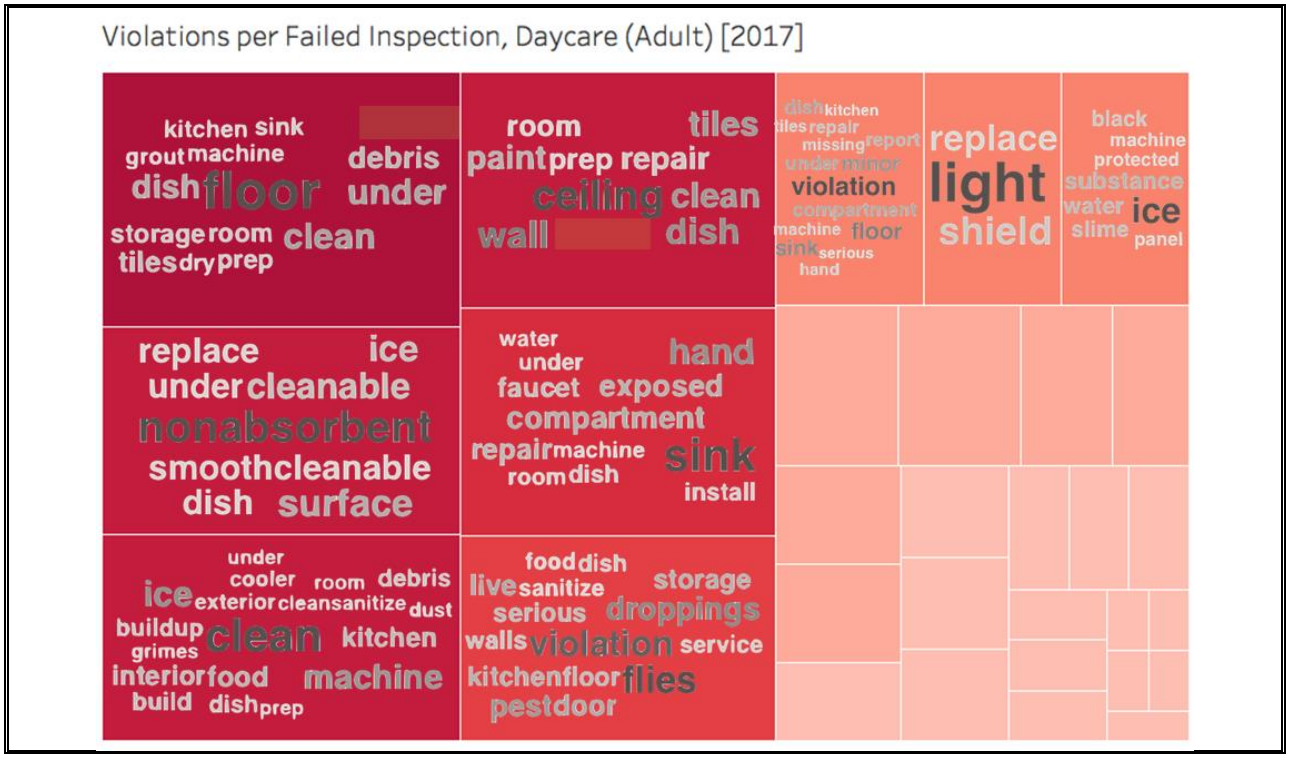


Figure 7.1

I was interested in looking at the most prevalent violations for failed Adult Daycare facility inspections in 2017, and the types of comments made for the violations. Given that 2 different types of data were being explored, two types of visuals, a tree map and word clouds, were used to create one final visual.

Data was extracted as a .json file, read, cleaned and transformed in R. Since two different types of visuals were used to create one final visual, two R scripts were written, and two separate text files were exported, then read into Tableau. The first visualization was a tree map which used a sequential color scheme from red because red is often associated with "failed" results. A darker red and larger boxed area indicates a larger count of the specific violation. Because there is more meaning in the comments associated with the violations than the title of the violation itself, a second type of visualization was created by creating word maps of the comments for each violation. Words showing up in the comments

with a greater frequency have a darker color, with less frequent words having a lighter color. A grayscale for the word maps was used because it contrasted the most with the sequential red color scheme used for the tree map. Each word map was then overlaid onto the area representing its respective violation in the tree map; violation titles and numbers were left out of the tree map to not distract from the word clouds. This was performed by creating a word cloud from the comments of each violation in Tableau, cutting the background color of each word map image, then pasting it onto its violation cell in the tree map. This was a pretty manual process given that this capability is not in Tableau. It is also important to note that innocuous words in the violation comments were removed so as to not distract from meaningful words in the word map. Such words included: the, of, all, there, etc. Another point to mention is that the bold Helvetica font was chosen specifically because of the process of cutting way the background of the word map. The bold Helvetica font had the crispest edges, and allowed for the most readability of the words once the background was deleted.

Based on the final visualization, it is easy to glean that most prevalent violations have to do with surfaces like walls, floors, countertops, etc. However, it is also interesting to note that two high-count violations include pests (with words like 'droppings', 'pests' and 'flies' appearing in a word map) and dirty appliances (by seeing words like 'black', 'slime' and 'machine' in another violation's word map). By combining two different visuals into one, a lot of information it represented at one time (the size and color of the tree map show that different violations show up more than others, while the color, size, and words of the word maps describe the actual violations given).

Ultimately, this data can be used to begin to understand what types of violations Adult Daycare facilities in Chicago are encountering. It is essential that aging populations live in safe and healthy environments, including those where food is prepared. Understanding the violations that are most frequently occurring can help legislators, facility owners/managers, and organizations in honing in on what types of issues may need extra funding (i.e. assistance in maintaining safe floors), or provide education to staff in these facilities on maintaining clean and safe environments.

Analysis and Discussion

With the size of our group and the variety of data, we chose to conduct the analyses and discussions with the visuals. Please see the content accompanying each visual for analysis and discussion.

Appendix A: Individual Contributions

Monica Choto

My individual contributions to this project pertain to proposing the initial dataset (found in the Chicago data portal), exploratory analysis, and a deeper dive on several aspects of the dataset, most importantly restaurants, although the latter wasn't presented as part of our final project. As a major foodie, my initial proposal to the team goal was to look at the areas of Chicago that had the most food inspections and how the results played out. Upon initial explorations, I was able to notice that a lot of these food inspections are done as part of a regular process or re-inspections rather than due to complaints and reports of potential disease, which led to the decision of exploring other areas of the dataset rather than focusing solely on restaurants. For instance - the graph below shows that the grand majority of inspections types are 'canvass inspections' and 'canvass reinspections' which are simply a frequently performed inspection due to the restaurant risk. A close 4th is 'license' which is an inspection done at the request of the owner. Not a very interesting property to explore further, as seen on the tree map in the appendix. The tree map uses size to represent the number of inspections, color per inspection type, text for the month number and result. It's a good way to see that most of the inspections of the dataset are merely pre-scheduled.

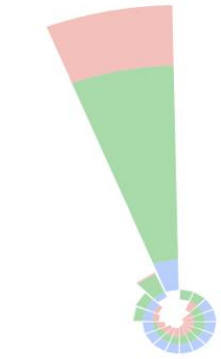
I therefore set to explore the greater dataset. Some of the ground-laying work indicated that we were right to think we should probably break down the data to explore further. As I explored the variables, I noticed that there were over 150 business types - which was a bit ridiculous taking into account that most of these different categories seemed to belong together. Erin had conveniently set to break them down already. I provided the first iterations of our maps to the team (included in several of the homework) as well as

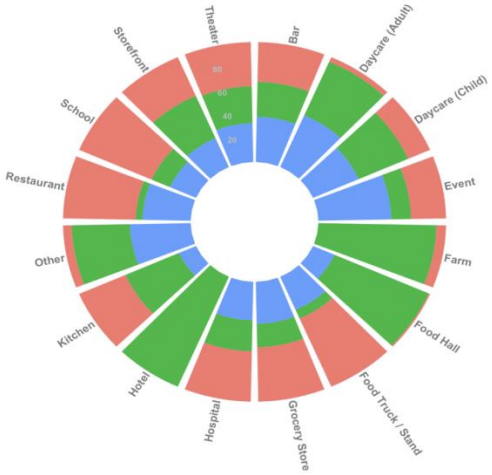
To represent the data as a whole, I wanted to be able to portray the reason why we chose to explore the dataset in such a way - which to the naked eye might have seemed compartmentalized. Even after trimming the dataset by business type, removing the missing values and incomplete cases, we ended up with 16 different variables, which were in no way easy to represent - at least not in a way that explained certain detail that I wanted to make clear. After trying several iterations or 'boring' and 'incomplete' graphs, I ended up going with a a radial axis stacked bar chart, because it didn't occupy a lot of space, allowed flexibility with fills and ordering, and still helped portray the message of how the distribution of business types varied. I chose to do this in R mainly because I knew that the graphing functions are fantastic. I computed the total occurrences of each business type and result in the dataset, and applied a logarithmic scale so that the data could be represented in a readable manner. I chose for the fills in the bars to be specific to the results per business type to see if there were any trends, and chose to have polar coordinates so that I could manipulate the y-axis limits to negative values and a circumference. The ylim coordinates for the coord_polar() function used in the final function account for a few things - negative values so that the circle has white space in the middle and that they aren't 'cones' and there's enough space for each bar. Start from 0 allows it to start from the axis rather than the center of the circle.

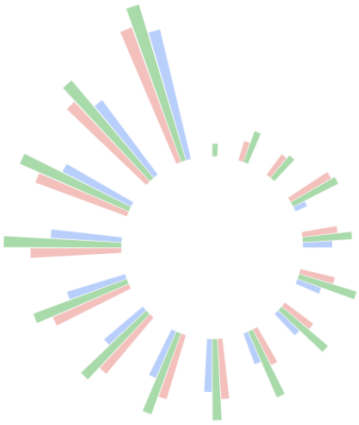
My initial attempts are below. The first one is a true scale of the data showing a stacked bar chart. The second is a grouped, logarithmic scaled, bar chart, and the second is a stacked bar chart scaled at a 100%, with every color representing the percentage of the whole made up by each result where each bar represents the number of occurrences of every result per business type. The fourth is the initial iteration of the final result – which is before ordering. Although the stacked bar graph showing percentage of total does portray the trend well, it failed to show the distribution of the data, which is why I chose the graph provided in the paper. While trying this I noticed that there definitely is a trend on

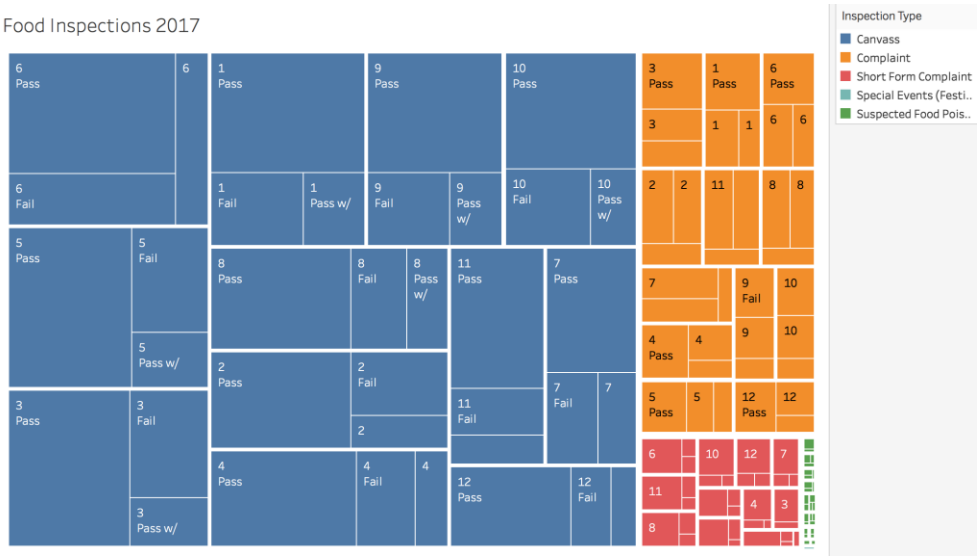
the proportion of passed vs failed vs passed w/conditions results per business type, so I decided to explore a bit further to see if there was any trend to exploit. Unfortunately, most everything pointed towards a 'no' without additional info that wasn't present in the data.

I learned that the details in data visualization is more important than I realized – I'm used to picking colors that seem logical, but I'd never stopped to think about the theory of why or how they make sense. Likewise, I really never thought there were people that did visualizations as bad as some examples you showed – until I had it happen in a meeting by a coworker recently. Staring at his graph during the presentation, I realized – I had no idea what he was trying to portray. Therefore, I will move forward with more caution in colors, ratios, labels, and specifically clutter.









Code for final version:

```
lab2=as.data.frame((unique(unlist(test$Facility.Group))))

#lab2
totes=ddply(test,~id.y+Facility.Group,summarise,sum=sum(TotalN))
#totes
lab2$tots=totes$sum
colnames(lab2)=c("Facility.Group","Total")
lab2$Newlev=reorder(lab2$Facility.Group,lab2$Total)
lab2=lab2[order(lab2$Total),]
lab2$id=seq(from=1,to=16,by=1)
number_of_bar2=nrow(lab2)
angle2= 90 - 360 * (lab2$id-0.5) /number_of_bar2
lab2$hjust<-ifelse( angle2 < -90, 1, 0)
lab2$angle<-ifelse(angle2 < -90, angle2+180, angle2)
lab2=lab2%>% arrange(Total)
lab3= merge(test,lab2,by="Facility.Group")
lab3$Newlev=reorder(lab3$Facility.Group,lab3$Total.y)



p3 = ggplot(lab3) +
   geom_bar(aes(x=Newlev, y=TotalN,fill=Results),stat="identity") +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    legend.text = element_text(size=10),
    legend.position = c(0.5,0.5),
    legend.key.size = unit(0.7, "cm"),
    panel.grid = element_blank()) +
    ylim(-20,25) + coord_polar(start=0)
lab3


p3+ ggtitle("Food Inspection Results by Facility Type") +
  theme(
    plot.title=element_text(hjust=0.5,size=28,margin=margin(1,1,1,15,"cm"))) +
geom_text(data=lab2,aes(x=Newlev,y=Total+0.5,label=Facility.Group,hjust=hjust),fontfac
e="bold",size=3.7,angle=lab2$angle,alpha=0.6,inherit.aes=FALSE)
InsCare=FoodInspData_allViol[FoodInspData_allViol$Results %in% c("Pass","Fail","Pass
```

```
w/ Conditions"),]
alltypes = as.data.frame(ddply(InsCare, .(InsCare$Facility.Group, InsCare$Results),
nrow))
colnames(alltypes)=c('Facility.Group','Results','Total')
alltypes
#ddply(alltypes,.(Facility.Group,Results),summarize,sum=sum(Inspection.ID))
#allo = alltypes[order(-alltypes$Total),]
allo=alltypes
#allo=allo[1:90,]
allo=filter(allo,!is.na(Facility.Group))
allo=filter(allo,Total>0)
allo$id=as.numeric(as.factor(allo$Facility.Group))
#allo = allo[order(-as.factor(allo$Facility.Group),]
allo
ddply(allo,~id+Facility.Group,summarise,sum=sum(Total))

lab=as.data.frame((unique(unlist(allo$Facility.Group))))
lab$id=(unique(unlist(allo$id)))
colnames(lab)=c("Facility.Group","id")
lab
test=merge(allo,lab,by="Facility.Group")
Test
Initial iterations:
test$TotalN=log(test$Total)

p2 = ggplot(test,aes(x=Facility.Group, y= TotalN ,group=Facility.Group,fill=Results))
+
   geom_bar(stat="identity",alpha=0.5) +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    legend.text = element_text(size=15),
    panel.grid = element_blank()) +
    ylim(-2.5,25) + coord_polar(start=0)
p2
test$Total
log(test$Total)
lab2=as.data.frame((unique(unlist(test$Facility.Group))))
lab2$id=unique(unlist(test$id.y))
lab2
totes=ddply(test,~id.y+Facility.Group,summarise,sum=sum(TotalN))
totes
lab2$tots=totes$sum
lab2
colnames(lab2)=c("Facility.Group","id","Total")
number_of_bar2=nrow(lab2)
angle2= 90 - 360 * (lab2$id-0.5) /number_of_bar2
lab2$hjust<-ifelse( angle2 < -90, 1, 0)
lab2$angle<-ifelse(angle2 < -90, angle2+180, angle2)
lab2=lab2%>% arrange(lab2$id,Total)
lab2

p2+ ggtitle("Food Inspection Results by Facility Type") +
theme(plot.title=element_text(hjust=0.5,size=28))+
geom_text(data=lab2,aes(x=id,y=Total+0.5,label=Facility.Group,hjust=hjust),fontface="b
old",size=5,angle=lab2$angle,alpha=0.6,inherit.aes=FALSE)
totes=ddply(lab3,~id.y+Facility.Group,summarise,sum=sum(Total.x))

data=data.frame(ID=lab3$Newlev,Results=lab3$Results,Total=(log(lab3$Total.x)*100))
data=data[data$Results %in% c("Pass","Fail","Pass w/ Conditions"),]
data=filter(data,!is.na(Total))
data=filter(data,Total>0)
```

```
data
#data = data %>% arrange(ID, Total)
# Set a number of 'empty bar' to add at the end of each group
empty_bar=4
to_add = data.frame( matrix(NA, empty_bar*nlevels(data$ID), ncol(data)) )
colnames(to_add) = colnames(data)
to_add$ID=rep(levels(data$ID), each=empty_bar)
data=rbind(data, to_add)
data=data %>% arrange(ID)
data$id=seq(1, nrow(data))

data
# Get the name and the y position of each label
label_data=data
number_of_bar=nrow(label_data)
angle= 90 - 360 * (label_data$ID-0.5) /number_of_bar     # I substract 0.5 because the
letter must have the angle of the center of the bars. Not extreme right(1) or extreme
left (0)
label_data$hjust<-ifelse( angle < -90, 1, 0)
label_data$angle<-ifelse(angle < -90, angle+180, angle)

# prepare a data frame for base lines
base_data=data %>%
  group_by(ID) %>%
  summarize(start=min(id), end=max(id) - empty_bar) %>%
  rowwise() %>%
  mutate(title=mean(c(start, end)))

# prepare a data frame for grid (scales)
grid_data = base_data
grid_data$end = grid_data$end[ c( nrow(grid_data), 1:nrow(grid_data)-1)] + 1
grid_data$start = grid_data$start - 1
grid_data=grid_data[-1,]


p5 = ggplot(data,aes(x=as.factor(id), y=Total,fill=Results)) +
  geom_bar(aes(x=as.factor(id), y=Total, fill=Results), stat="identity", alpha=0.5) +
  theme_minimal() +
  theme(
    axis.text = element_blank(),
    axis.title = element_blank(),
    legend.text = element_text(size=10),
    panel.grid = element_blank()) +
    ylim(-500,900) + coord_polar(start=0)

p5 + ggtitle("Food Inspection Results by Facility Type")
```

Uei Lei

With so many visualization talented team members, my main contribution to the team and project consisted of serving as the liaison and coordinating the final products of the presentation and write up. I was also able to provide background and context on food inspection given my work at the health department that generated the data. Although I do not have visuals represented in the final write up, I did explore the data and produce visuals as submitted in homework sets 3 and 4.

What I've learned is perhaps more of a reinforcement in that with 20+ years in public health and healthcare delivery, policy, planning, my experience and skill sets are more adept at strategic thinking, goal setting, and managing projects and less so to production of visualizations despite inherent penchant for analytics and visualizations. I should say, at this point in time for from my teammates, I

learned I have a steep learning curve to become comfortable navigating and using these new tools of the devil (i.e. visualization software). Also, as I go through the rest of the program, it would be good for my individual growth to focus on technical aspects of projects rather than fulfilling the coordinating role to which I am accustomed.

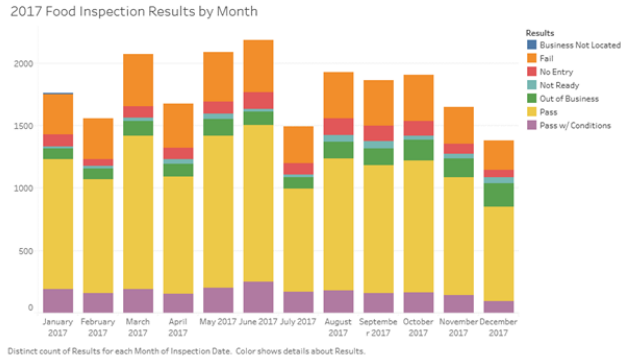
Joy Matias

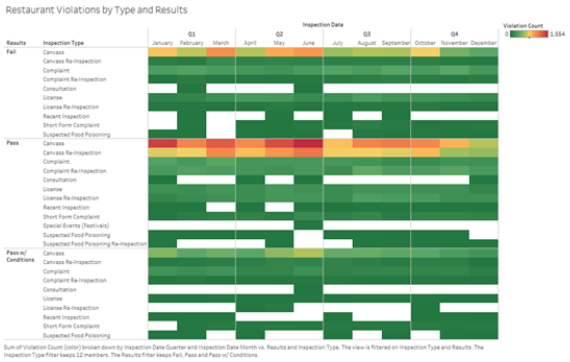
My individual contribution to the project originated with a bar graph of the different results (pass/fail/pass with conditions/out of business/not ready/business not located/no entry). I noted through the bar graph that the pass, pass w/conditions and the fail were the results of interest as the other result options did not truly measure performance of facilities. The bar graph gives very little information with regard to underlying factors of pass/fail rates. It is not apparent whether the pass/fail rates have seasonality although we note that there appears to be seasonality with regard to the amounts of inspections performed on facilities with the 1st half of the year having higher amount of inspections peaking in June and a low point in July. From the bar graph, the question of the inspection types and their relationship to pass/fail rates arose in discussions which then yielded the heat maps exploration.

The first heat map below shows restaurant violations by type and results. Here, we note that the seasonality we noticed in the bar graph appears to correlate with the violation count as for all results (pass/pass w/conditions/fail). It is also interesting that the higher frequency of violations is in the canvas and canvas re-inspection types. These are random inspections on facilities based on the risk type. It is peculiar that the violation counts appear higher for the passing results; however, we would need to re-examine the violation types in specificity to truly understand what constitutes fail violations verses pass violations.

The second heat map was made to explore the restaurant violations over month by zip code. The highest violation count noted occur in the 60639 zip code followed by 60614. It is also curious that the 60607 zip code has one of the highest violation counts in October despite being relatively neutral most other months. The questions which arose from the bar graphs and heat map begged the questions that yielded the final visualization of heatmap with avg pass/fail rate by facility type. Discussions among the group directed the final visualization by facility type which originally suggested that the heatmap be specific to schools; however, in revising the heatmap among different facility types, it was evident that the heat map was more value added for aggregated data as on a facility type level the data points were too few to display effectively on a heatmap.

After creating the final heatmap, I also leveraged the violations heatmap by zipcode and month to further review the results of the 60614 zip code. As this is the DePaul and surrounding area zip code, I thought it appropriate and curious to review the restaurants which failed their inspections and their violation counts. This was the word cloud that was created for the final assignment. Although it was not used for this paper based on the inability to make generalized conclusions about the food inspection data, the detail is of unique interest to the campus population.





Erin Murphy (Tree Map + Word Cloud)

I participated in the group in various ways, including writing the R code to process any exported .json file of the data that was downloaded from the website. This code created different formats of text files which were used for the tree map and word clouds I created for the final presentation and report. An initial visualization I created at the beginning of the quarter on our group's data was a word cloud (which was a major reason why I created the R code that was used to create my final visualization). One thing I learned from creating the R code was the melt() function (first to separate violation comments, then to separate all the words from the comments).

I also joined all of the group meetings to discuss our project, met all of our project deliverable deadlines, and provided feedback to my group members. Along with this, I ended up cleaning the data to group

different facility types into a smaller number of groups in order to help with our groups' analyses better. For the final portions of the group project, I created my final visualizations for the presentation, and wrote up the section of my topic for our final paper.

Since there were group members who had already chosen to look at the overall dataset, and some of the larger groups that were represented in the data (i.e. restaurants), I chose what I thought would be an interesting group to delve into further, Adult Daycare facilities. Because of this, I was able to look at the data in all types of ways and used multiple techniques in R and Tableau. As mentioned, I created a tree map and word clouds, but I also played around with a network map (which did not really amount to anything, except that I believe I got the R code to work).

I also learned a lot more about how to look at visualizations and how to create them in terms of things I had not thought about before like color choices, divergent / sequential color schemes, and how we perceive images (i.e. aligning bar graphs or when to choose one type of visualization over another). I also learned a lot about the different types of visualizations that are available (like heat maps and choropleths), and more powerfully, how to create them in R and Tableau. The first R lab was super insightful!

Initial Analysis

From the exploratory analysis performed by Monica which showed that Adult Care facilities had the largest percent of failed inspections, I chose to look into that facility group further. My initial analysis included a tree map of the entire Adult Care facility population (which included Results of: Pass, Pass w/ Conditions, Fail, No Entry, Not Ready and Out of Business).

The initial data contains 1 record per inspection, and each inspection (which has its own unique Inspection ID and occurs on 1 date), can have multiple violations. Violations and their respective comments, are contained in 1 cell, and are delimited by | (pipe) characters. For example, the below inspection has 3 violations for its single inspection date:

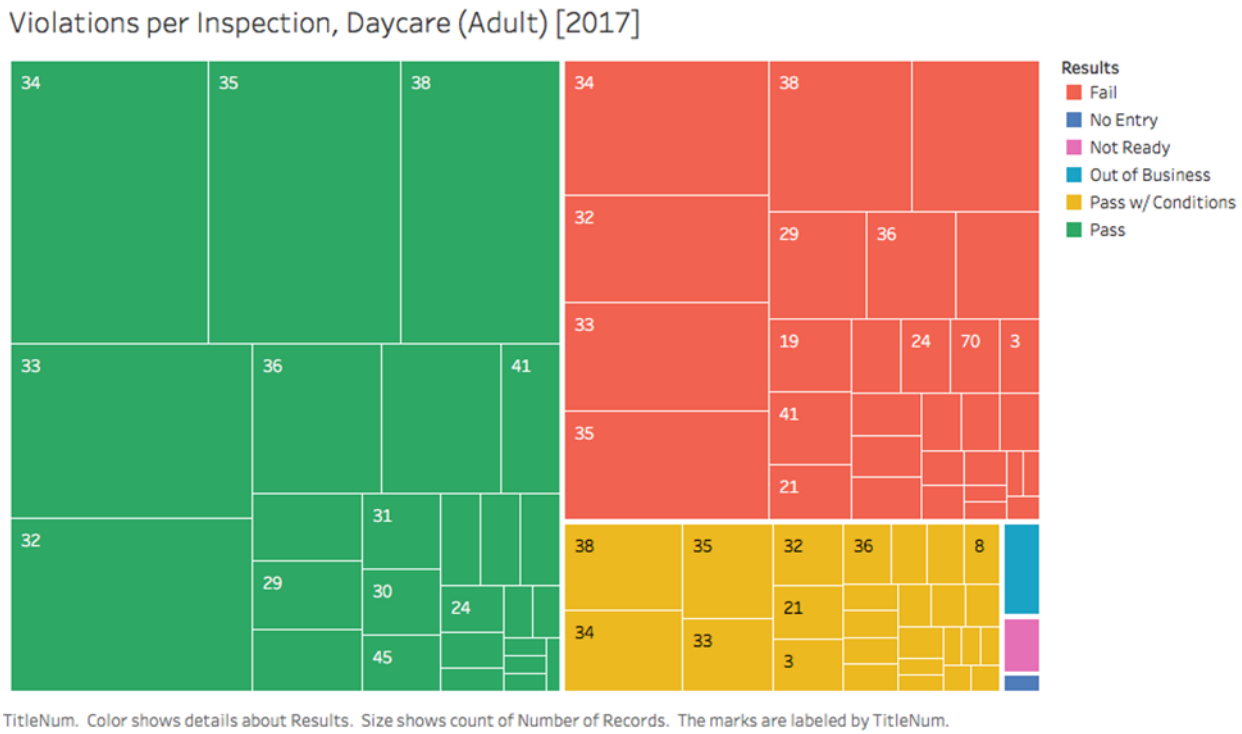
Inspection ID	Inspection Date	Inspection Type	Results	Violations
2129692	12/27/2017	Canvass Re-Inspection	Pass	29. PREVIOUS MINOR VIOLATION(S) CORRECTED 7-42-090 - Comments: BACK FLOW PREVENTION DEVICES INSTALLED AT ICE MACHINE AND COFFEE MACHINE WATERLINES. | 34. FLOORS: CONSTRUCTED PER CODE, CLEANED, GOOD REPAIR, COVING INSTALLED, DUST-LESS CLEANING METHODS USED - Comments: CORRECTED. | 38. VENTILATION: ROOMS AND EQUIPMENT VENTED AS REQUIRED: PLUMBING: INSTALLED AND MAINTAINED - Comments: CORRECTED.

Instead of looking at the number of facilities their inspection results, I wanted to look at the number and types of violations for Adult Care facilities based on their respective inspection results. An R script was written, utilizing the melt() function, to split out the violations so that each record in the data represented a violation, per inspection ID / inspection date:

Inspection ID	Inspection Date	Inspection Type	Results	Violations

Inspection ID	Inspection Date	Inspection Type	Results	Violations
2129692	12/27/2017	Canvass Re-Inspection	Pass	29. PREVIOUS MINOR VIOLATION(S) CORRECTED 7-42-090 - Comments: BACK FLOW PREVENTION DEVICES INSTALLED AT ICE MACHINE AND COFFEE MACHINE WATERLINES.
2129692	12/27/2017	Canvass Re-Inspection	Pass	34. FLOORS: CONSTRUCTED PER CODE, CLEANED, GOOD REPAIR, COVING INSTALLED, DUST-LESS CLEANING METHODS USED - Comments: CORRECTED.
2129692	12/27/2017	Canvass Re-Inspection	Pass	38. VENTILATION: ROOMS AND EQUIPMENT VENTED AS REQUIRED: PLUMBING: INSTALLED AND MAINTAINED - Comments: CORRECTED.

After splitting the data by violation, I was able to count the violations by Result and visually inspect the most popular violation by Result, as illustrated by the following tree map (see the visualization below entitled, 'Violations per Inspection, Daycare (Adult) [2017]').



At first, I was confused because the Pass and Fail results had the same Violations. But looking at the data further, it this makes senses because a) they are probably commonly cited violations, and b) as seen in the above example, when a re-inspection is performed, the previous violation is commented on (such as being corrected). Given this, I became more interested in the types of comments given to the failed inspections, and therefore decided two things: 1) I was only going to look at failed inspections (because those are more interesting), and 2) I wanted to get a better idea of the key words in the comments, and not necessarily the violation type. This ultimately resulted in a tree map of all failed inspections for Adult Care facilities, with overlaying word clouds over the most common violations.

The word clouds were created by taking the violation data (above), and performing a second melt() function on the violations so that each word, by violation, by inspection ID / inspection date had its own record. This allowed for the ability to count words per violation, and ultimately create an individual word cloud per violation.

The following R code was used to create the text files for the tree map visual, and the word clouds. Both text files were read into Tableau, which was used to create the visuals.

TREE MAP R CODE

```
library(RJSONIO) # for reading in the json file and converting to a dataframe
library(gdata) # for the trim function
library(reshape) # for transposing the lists to rows
library(stringr)
library(ggplot2)
library(ggraph)
library(igraph)

document <- fromJSON("/Users/erinmurphy/Documents/school/CSC
465/Project/FoodInspData_AdultDay2017.json")

#parse data
data_insp<-document[['data']]

#extract data -- everything but geographical data
grabInfo<-function(var){
  print(paste("Variable", var, sep=" "))
  sapply(data_insp, function(x) returnData(x, var))
}

returnData<-function(x, var){
  if(!is.null( x[[var]])){
    return( trim(x[[var]]))
  }else{
    return(NA)
  }
}
InspDataDF <- data.frame(sapply(1:22, grabInfo), stringsAsFactors=FALSE)

#extract geographical data, and combine
grabGeoInfo<-function(val){

  l<- length(data_insp[[1]][[val]])
  tmp<-lapply(1:l, function(y)

    sapply(data_insp, function(x){

      if(!is.null(x[[val]][[y]])){
        return(x[[val]][[y]])
      }else{
        return(NA)
      }

    })
  )
}

#format latitude and longitue into data frams
InspDataLat <- data.frame(do.call("cbind", grabGeoInfo(23)), stringsAsFactors=FALSE)
InspDataLong <- data.frame(do.call("cbind", grabGeoInfo(24)), stringsAsFactors=FALSE)

#combine with original dataset
```

```
InspDataDF<-cbind(InspDataDF, InspDataLat, InspDataLong)

#get column names from the data
columns <- document[['meta']][['view']][['columns']]

#get data column names
getNames <- function(x){
  if(is.null(columns[[x]]$subColumnTypes)){
    return(columns[[x]]$name)
  }else{
    return(columns[[x]]$subColumnTypes)
  }
}

InspNames <- unlist(sapply(1:24, getNames))
InspNames

#assign column names
names(InspDataDF)<-InspNames
names(InspDataDF)

# split the violations into a list to be able to transpose to rows
InspDataDF$Violations_lst <- as.list(strsplit(InspDataDF$Violations, "[|]"))

# transpose all violations
violations.melt <- melt(InspDataDF$Violations_lst)
violations.melt$rnames <- violations.melt$L1

InspDataDF$rnames <- as.integer(rownames(InspDataDF))

# split the violations into a list to be able to transpose to rows
InspDataDF$Violations_lst <- as.list(strsplit(InspDataDF$Violations, "[|]"))

# transpose all violations
violations.melt <- melt(InspDataDF$Violations_lst)
violations.melt$rnames <- violations.melt$L1

InspDataDF$rnames <- as.integer(rownames(InspDataDF))

# merge inspection comments with restaurant data
FoodInspData_Adult <- merge(InspDataDF, violations.melt,by=c("rnames"))
FoodInspData_Adult <- FoodInspData_Adult[,c("Inspection ID", "License #", "Facility
Type", "Risk", "Inspection Date", "Inspection Type", "Latitude", "Longitude",
"Results", "value")]


# split comments from violation title
FoodInspData_Adult$v.value <- lapply(FoodInspData_Adult$value, as.character)
FoodInspData_Adult$end1 <- str_locate(FoodInspData_Adult$v.value, " - Comments: ")
FoodInspData_Adult$end2 <- nchar(FoodInspData_Adult$v.value, type = "chars") + 1

FoodInspData_Adult$v.title <-
substr(FoodInspData_Adult$value,1,FoodInspData_Adult$end1)
FoodInspData_Adult$v.comment <-
tolower(substr(FoodInspData_Adult$value,(FoodInspData_Adult$end1 + 13),
FoodInspData_Adult$end2))

FoodInspData_Adult <- FoodInspData_Adult[,c("Inspection ID", "License #", "Facility
Type", "Risk", "Inspection Date", "Inspection Type", "Latitude", "Longitude",
"Results", "v.title", "v.comment")]
names(FoodInspData_Adult)

FoodInspData_Adult_fail <- subset(FoodInspData_Adult, FoodInspData_Adult$Results ==
```

```
"Fail")
FoodInspData_Adult_pass <- subset(FoodInspData_Adult, FoodInspData_Adult$Results ==
"Pass")
FoodInspData_Adult_passCond <- subset(FoodInspData_Adult, FoodInspData_Adult$Results
== "Pass w/ Conditions")


# export final list of words
write.table(FoodInspData_Adult, "/Users/erinmurphy/Documents/school/CSC
465/Project/FoodInspData_adult2017.txt", sep="|")

WORD CLOUD R CODE:
library(RJSONIO) # for reading in the json file and converting to a dataframe
library(gdata) # for the trim function
library(reshape) # for transposing the lists to rows
library(stringr)

document <- fromJSON("/Users/erinmurphy/Documents/school/CSC
465/Project/FoodInspData_AdultDay2017.json")

#parse data
data_insp<-document[['data']]

#extract data -- everything but geographical data
grabInfo<-function(var){
  print(paste("Variable", var, sep=" "))
  sapply(data_insp, function(x) returnData(x, var))
}

returnData<-function(x, var){
  if(!is.null( x[[var]])){
    return( trim(x[[var]]))
  }else{
    return(NA)
  }
}
InspDataDF <- data.frame(sapply(1:22, grabInfo), stringsAsFactors=FALSE)

#extract geographical data, and combine
grabGeoInfo<-function(val){

  l<- length(data_insp[[1]][[val]])
  tmp<-lapply(1:l, function(y)

    sapply(data_insp, function(x){

      if(!is.null(x[[val]][[y]])){
        return(x[[val]][[y]])
      }else{
        return(NA)
      }

    })
  )
}

#format latitude and longitue into data frams
InspDataLat <- data.frame(do.call("cbind", grabGeoInfo(23)), stringsAsFactors=FALSE)
InspDataLong <- data.frame(do.call("cbind", grabGeoInfo(24)), stringsAsFactors=FALSE)

#combine with original dataset
InspDataDF<-cbind(InspDataDF, InspDataLat, InspDataLong)
```

```
#get column names from the data
columns <- document[['meta']][['view']][['columns']]

#get data column names
getNames <- function(x){
  if(is.null(columns[[x]]$subColumnTypes)){
    return(columns[[x]]$name)
  }else{
    return(columns[[x]]$subColumnTypes)
  }
}

InspNames <- unlist(sapply(1:24, getNames))

#assign column names
names(InspDataDF)<-InspNames
#head(InspDataDF)

# split the violations into a list to be able to transpose to rows
InspDataDF$Violations_lst <- as.list(strsplit(InspDataDF$Violations, "[|]"))

# transpose all violations
violations.melt <- melt(InspDataDF$Violations_lst)
violations.melt$rnames <- violations.melt$L1

InspDataDF$rnames <- as.integer(rownames(InspDataDF))

# merge inspection comments with restaurant data
FoodInspData_wrdmap <- merge(InspDataDF, violations.melt,by=c("rnames"))
FoodInspData_wrdmap <- FoodInspData_wrdmap[,c("Facility Type", "Risk", "Inspection
Type", "Results", "value")]


# split comments from violation title
FoodInspData_wrdmap$v.value <- lapply(FoodInspData_wrdmap$value, as.character)
FoodInspData_wrdmap$end1 <- str_locate(FoodInspData_wrdmap$v.value, " - Comments: ")
FoodInspData_wrdmap$end2 <- nchar(FoodInspData_wrdmap$v.value, type = "chars") + 1

FoodInspData_wrdmap$v.title <-
tolower(substr(FoodInspData_wrdmap$value,1,FoodInspData_wrdmap$end1))
FoodInspData_wrdmap$v.comment <-
tolower(substr(FoodInspData_wrdmap$value,(FoodInspData_wrdmap$end1 + 13),
FoodInspData_wrdmap$end2))

# final word map data
FoodInsp_WordMap <- FoodInspData_wrdmap[, c("Facility Type", "Risk", "Inspection
Type", "Results"
                                            , "v.title", "v.comment")]

fac.type <- as.data.frame(unique(FoodInsp_WordMap$`Facility Type`))

titles <- as.data.frame(unique(FoodInsp_WordMap$v.title))

# remove special characters and meaningless words (like 'the')
FoodInsp_WordMap$v.comment_lst <- as.list(strsplit(FoodInsp_WordMap$v.comment, " "))
comment_lst <- melt(FoodInsp_WordMap$v.comment_lst)
class(comment_lst)
comment_lst$wrd_lst <- lapply(comment_lst$value, as.character)
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[[:punct:]]", "")
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[[:blank:]]", "")
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[[:digit:]]", "")
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[[:space:]]", "")
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[$]", "")
```

```
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[=]", "")
comment_lst$len <- nchar(comment_lst$wrd_lst)

comment_lst <- subset(comment_lst, len > 2, select=c(wrd_lst, len, value, L1))
# remove fluff and verbs
comment_lst <- subset(comment_lst, !(wrd_lst %in%
c("the","and","for","had","not","are","has","now","all","from","with"

,"must","off","around","able","after","again","also","another","any","anything"

,"been","before","being","better","but","call","andor","area","areas","both","city"

,"comp","control","corrected","instructed","maintain","need","needs","needed"

,"noted","observed","same","shall","than","that","this","use","used","while","along"

,"inside","near","provide","front","proof","remove","site","test","who","said"

,"three","two","one","small","where","more","when","next","kit","detail","have"

,"throughout","out","aand","about","actual","instd","proper","chicago","provided"

,"during","previous","was","following","comments","etc","were","other","such"
                                                    ,"some","name","bar")))


comment_lst$rnames <- comment_lst$L1
FoodInsp_WordMap$rnames <- as.integer(rownames(FoodInsp_WordMap))

# merge inspection details with word list
FoodInsp_WordMap_lst <- merge(FoodInsp_WordMap, comment_lst,by=c("rnames"))
names(FoodInsp_WordMap_lst)

FoodInsp_WordMap_lst$title <- trimws(FoodInsp_WordMap_lst$v.title, which = c("both",
"left", "right"))
FoodInsp_WordMap_lst$end1 <- str_locate(FoodInsp_WordMap_lst$title, ". ")
FoodInsp_WordMap_lst$titleChar <-
tolower(substr(FoodInsp_WordMap_lst$title,1,FoodInsp_WordMap_lst$end1 - 1))
FoodInsp_WordMap_lst <- FoodInsp_WordMap_lst[,c("Facility Type", "Risk", "Inspection
Type","titleChar", "Results", "wrd_lst")]

# export final list of words
write.table(FoodInsp_WordMap_lst, "/Users/erinmurphy/Documents/school/CSC
465/Project/FoodInsp_WordMapAdult_lst.txt", sep="|")
```

WORD CLOUD R CODE:

```
library(RJSONIO) # for reading in the json file and converting to a dataframe
library(gdata) # for the trim function
library(reshape) # for transposing the lists to rows
library(stringr)

document <- fromJSON("/Users/erinmurphy/Documents/school/CSC
465/Project/FoodInspData_AdultDay2017.json")

#parse data
data_insp<-document[['data']]

#extract data -- everything but geographical data
grabInfo<-function(var){
  print(paste("Variable", var, sep=" "))
  sapply(data_insp, function(x) returnData(x, var))
}
```

```
returnData<-function(x, var){
  if(!is.null( x[[var]])){
    return( trim(x[[var]]))
  }else{
    return(NA)
  }
}
InspDataDF <- data.frame(sapply(1:22, grabInfo), stringsAsFactors=FALSE)

#extract geographical data, and combine
grabGeoInfo<-function(val){

  l<- length(data_insp[[1]][[val]])
  tmp<-lapply(1:l, function(y)

    sapply(data_insp, function(x){

      if(!is.null(x[[val]][[y]])){
        return(x[[val]][[y]])
      }else{
        return(NA)
      }

    })
  )
}

#format latitude and longitue into data frams
InspDataLat <- data.frame(do.call("cbind", grabGeoInfo(23)), stringsAsFactors=FALSE)
InspDataLong <- data.frame(do.call("cbind", grabGeoInfo(24)), stringsAsFactors=FALSE)

#combine with original dataset
InspDataDF<-cbind(InspDataDF, InspDataLat, InspDataLong)

#get column names from the data
columns <- document[['meta']][['view']][['columns']]

#get data column names
getNames <- function(x){
  if(is.null(columns[[x]]$subColumnTypes)){
    return(columns[[x]]$name)
  }else{
    return(columns[[x]]$subColumnTypes)
  }
}

InspNames <- unlist(sapply(1:24, getNames))

#assign column names
names(InspDataDF)<-InspNames
#head(InspDataDF)

# split the violations into a list to be able to transpose to rows
InspDataDF$Violations_lst <- as.list(strsplit(InspDataDF$Violations, "[|]"))

# transpose all violations
violations.melt <- melt(InspDataDF$Violations_lst)
violations.melt$rnames <- violations.melt$L1

InspDataDF$rnames <- as.integer(rownames(InspDataDF))

# merge inspection comments with restaurant data
```

```
FoodInspData_wrdmap <- merge(InspDataDF, violations.melt,by=c("rnames"))
FoodInspData_wrdmap <- FoodInspData_wrdmap[,c("Facility Type", "Risk", "Inspection
Type", "Results", "value")]


# split comments from violation title
FoodInspData_wrdmap$v.value <- lapply(FoodInspData_wrdmap$value, as.character)
FoodInspData_wrdmap$end1 <- str_locate(FoodInspData_wrdmap$v.value, " - Comments: ")
FoodInspData_wrdmap$end2 <- nchar(FoodInspData_wrdmap$v.value, type = "chars") + 1

FoodInspData_wrdmap$v.title <-
tolower(substr(FoodInspData_wrdmap$value,1,FoodInspData_wrdmap$end1))
FoodInspData_wrdmap$v.comment <-
tolower(substr(FoodInspData_wrdmap$value,(FoodInspData_wrdmap$end1 + 13),
FoodInspData_wrdmap$end2))

# final word map data
FoodInsp_WordMap <- FoodInspData_wrdmap[, c("Facility Type", "Risk", "Inspection
Type", "Results"
                                            , "v.title", "v.comment")]

fac.type <- as.data.frame(unique(FoodInsp_WordMap$`Facility Type`))

titles <- as.data.frame(unique(FoodInsp_WordMap$v.title))

# remove special characters and meaningless words (like 'the')
FoodInsp_WordMap$v.comment_lst <- as.list(strsplit(FoodInsp_WordMap$v.comment, " "))
comment_lst <- melt(FoodInsp_WordMap$v.comment_lst)
class(comment_lst)
comment_lst$wrd_lst <- lapply(comment_lst$value, as.character)
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[[:punct:]]", "")
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[[:blank:]]", "")
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[[:digit:]]", "")
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[[:space:]]", "")
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[$]", "")
comment_lst$wrd_lst <- str_replace_all(comment_lst$wrd_lst, "[=]", "")
comment_lst$len <- nchar(comment_lst$wrd_lst)

comment_lst <- subset(comment_lst, len > 2, select=c(wrd_lst, len, value, L1))
# remove fluff and verbs
comment_lst <- subset(comment_lst, !(wrd_lst %in%
c("the","and","for","had","not","are","has","now","all","from","with"

,"must","off","around","able","after","again","also","another","any","anything"

,"been","before","being","better","but","call","andor","area","areas","both","city"

,"comp","control","corrected","instructed","maintain","need","needs","needed"

,"noted","observed","same","shall","than","that","this","use","used","while","along"

,"inside","near","provide","front","proof","remove","site","test","who","said"

,"three","two","one","small","where","more","when","next","kit","detail","have"

,"throughout","out","aand","about","actual","instd","proper","chicago","provided"

,"during","previous","was","following","comments","etc","were","other","such"
                                                    ,"some","name","bar")))



comment_lst$rnames <- comment_lst$L1
```

```
FoodInsp_WordMap$rnames <- as.integer(rownames(FoodInsp_WordMap))

# merge inspection details with word list
FoodInsp_WordMap_lst <- merge(FoodInsp_WordMap, comment_lst,by=c("rnames"))
names(FoodInsp_WordMap_lst)

FoodInsp_WordMap_lst$title <- trimws(FoodInsp_WordMap_lst$v.title, which = c("both",
"left", "right"))
FoodInsp_WordMap_lst$end1 <- str_locate(FoodInsp_WordMap_lst$title, ". ")
FoodInsp_WordMap_lst$titleChar <-
tolower(substr(FoodInsp_WordMap_lst$title,1,FoodInsp_WordMap_lst$end1 - 1))
FoodInsp_WordMap_lst <- FoodInsp_WordMap_lst[,c("Facility Type", "Risk", "Inspection
Type","titleChar", "Results", "wrd_lst")]

# export final list of words
write.table(FoodInsp_WordMap_lst, "/Users/erinmurphy/Documents/school/CSC
465/Project/FoodInsp_WordMapAdult_lst.txt", sep="|")
```

Jiaxi Peng

My main contribution in this team was to analyze the performance of restaurants (i.e. pass/fail rate) in Chicago area.

At the very beginning, I created a choropleth map (Figure 1) to visual the distribution of different results for all restaurants by zip code. I chose continuous color for different areas, so that we can see clearly darker areas have higher pass rate while lighter areas have lower pass rate. Overall, the restaurants in southwest, northwest area have higher pass rate; but restaurants in northeast and southeast part have lower pass rate.

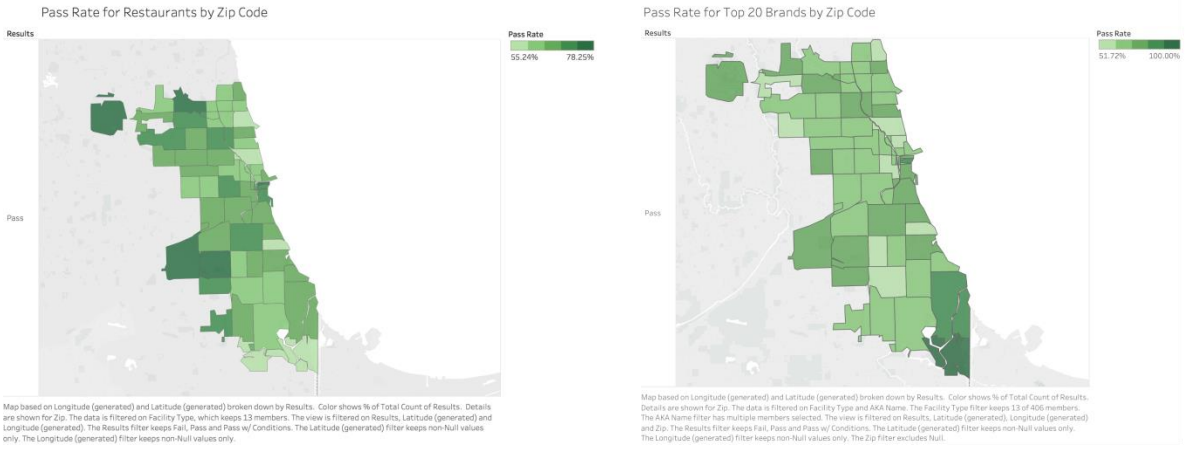


Figure 1

Next, since I care more about the performance of more common restaurants (where my friends and I always go have a bite), I sorted restaurants names by frequency and chose most 20 frequent (brands like Starbucks, KFC, McDonald's, Subway, Dunkin Donuts, etc) and created another choropleth map (Figure 2). This map shows totally different contribution from Figure 1. For these 20 most common brands, the branches in southeast have highest pass rate, some of which are even 100%; while in the Lincoln park area, west loop, and southwest, the pass rate are the lowest of only about 50%.

Figure 2

At last, I created a 100% stacked bar chart (which I presented in class) to show exact pass/ pass with condition/ fail rate for these 20 most common restaurants, and made suggestions which restaurants are best places to go.

From this class, I have learnt different visualization techniques which is a good way to present your findings and conclusions in class or in work. More importantly, though different examples and assignments, I have a deep understanding of how to use different techniques under different situations and for different purpose.

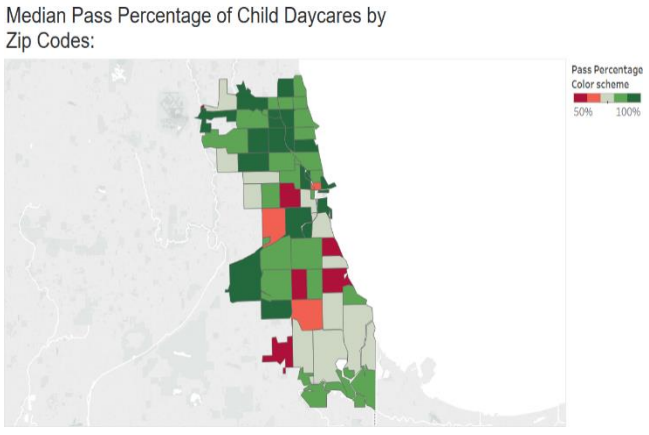
Neetu Singh

While working with the team, I contributed for managing logistics such as taking initiatives for setting up regular project meetings and collaborating with the team to identify right communication channel for the group discussions. I also worked with the team to ensure there is no overlap of visuals created by the team members.

Being a mother, I realized Chicago Inspections dataset should be filtered to generate insights for daycares, so I cleaned the dataset for daycares in R programming and explored the data in both Tableau and R programming to understand what visuals can present the data efficiently and effectively to families and parents. While trying Tukey graph, stacked bar, tree map, violin plot, and choropleth map during data exploration, I understood that Choropleth map would present the data to the families in the right format. This not only shows the health of the day cares but also informs about the location on the Chicago map.

Chicago Food Inspection data needed to be cleaned to explore child day-care data by cleaning some issues for example: there were some discrepancies in day care's name such as punctuations, special characters, and brackets etc. and some of daycares have missing information about inspection results etc. There are some daycares which were not in business or were not ready yet, so I excluded during the cleaning process to make the data more relevant.

While analyzing the daycare inspection data, I identified the number of inspections on each daycare over 8 years period. Out of these number of inspections I identified number of Pass and number of fail inspections. Based on this Pass and Fail numbers, I calculated the percentage of Pass and Fail for each day care. I did data cleaning in R and then visualized these findings by Choropleth map in Tableau.



In addition, I performed text analysis for inspection codes and comments, and created text clouds for Pass and Fail inspections for daycares to help users understand what are the common issues in failed inspections and how pattern is different between Pass and Fail inspections.

Failed Inspections

Passed Inspections

Learnings from the team: -

Since team consists of both full time and online students, I learned a lot from experienced professionals in the team. I learned to effectively collaborate with the team members in both online and face to face settings, and critically validate the solution before finally presenting it to the audience. Also, as I worked with the team, I learned new techniques such as data melting to clean the textual data.

Major learning from the class:

Biggest learnings from the class were the application of the data analysis and visualization in the real-world context, and importance of various facets of the visuals such as color schemes and interactive graphs in the data story. I learned all the visuals ranging from simple to complex ones, and applied most of them during exploration for class assignments and the project. Even though I have good understanding of different visuals now, I am more equipped on identifying what visuals would better communicate the story to the audience in different contexts.

Additional Analysis – R Code for child daycare data exploration:

```
install.packages("dplyr")
library(dplyr)


#reading Food inspection database into R work space (please change the
working directory)
fread<-read.csv("C:/Data Science/Neetu's
project/Food_Inspections2.csv",header=TRUE)


#reading Day care names into R work space (please change the working
directory)
careread<-read.csv("C:/Data Science/Neetu's
```

```
project/percentage.csv",header=TRUE)


#Initializing two numeric vectors with same length as number of daycares to
store percentage values
x<-vector("numeric", length = nrow(careread))
y<-vector("numeric", length = nrow(careread))
c<-vector("numeric", length = nrow(careread))
#comparing each day care with the food inspection dataset and
#then identify pass and fail percentage of each day care
for(i in 1:nrow(careread))
{

#matching day care address inside food inspection database to find a unique
match
interim<-  fread[grep(careread$Address[i],fread$Address),]


#counting number of Passes and fails for the selected daycare
passfail<-count(interim, Results)
p1<-passfail$n[1]
p2<-passfail$n[2]


if(is.na(passfail$n[2]))
{
 #when there are only passes or only fails, ensuring that fails and passes
are captured
 #currectly to calculate the percentages
if(passfail$Results[1]=="Pass"){
p2<-p1
p1<-0
}
else{
p2<-0}
}
#count of number of inspections
c[i]<-p1+p2
#calculation of the percentage of fails
x[i]<-p1/(p1+p2)
#calculation of the percentage of passes
y[i]<-p2/(p1+p2)
}


#Writting passes and fails as percentage to the file.
```

```
careread$Failpct<-x
careread$Passpct<-y
careread$Count<-c
write.csv(careread,"C:/Data Science/Neetu's project/percentage.csv")
```

Wenyi Yan

In the group, I take charge of data analysis of restaurant part, providing network graph making which is about restaurant violation keywords. In addition, I participated group discussion actively and came up with new ideas like to research on bad performing restaurants (with high risk level and high failure rate) in Chicago which has very practical meaning. In exploratory analysis part, I did some complementary graphs like interactive time series line graph of food inspection frequency in recent 3 years and heat map about distribution of risk level of establishments in recent 3 years to help group better explore the data, although they didn't show up in main body because of the limitation on graph amount.

Except network graph showed above, I also created many other graphs from different perspectives to dig more into restaurant data, such as bar graph which is to find Top 20 worst restaurants; choropleth which is to get the distribution of these restaurants; word cloud which is to refine keywords of violations; and leaflet map of restaurants broken down by fail and pass results which is interactive. I'll put the codes and whole graphs I did in Appendix B.

In this course, I learnt many data visualization techniques to discover insight in data and learnt how to use these techniques properly for different purposes. For example, I used word map to discover violation keywords and network graph to discover relationships between them. Besides, I learnt how to collect different visualizations together and tell a complete story about my findings. For example, I wanted to tell a story about bad performing restaurants in Chicago, so I used bar graph to find Top 20 bad performing restaurants, then used choropleth to discover the distributions of these restaurants with high risk level and high failure rate, and used word cloud and network graph on violation, exploring and analyzing what made them perform bad.

What's more, I learned a new domain of knowledge – NLP (Natural Language Processing) and a new statistics theory – TF-IDF (Term Frequency–Inverse Document Frequency) while doing word cloud and network graph. To process text data in "Violations" in the dataset, I used tm packages in R to remove stop words, lowercase and stem words, replace and delete inappropriate words. To refine the most important words in violations, I used TF-IDF to compute weights according to the frequency of usage inside an individual document as opposed to the entire dataset. Especially processed text data in "Violations" column using tm package. Then, I used TF-IDF theory to find out important UniGrams and BiGrams: I used UniGrams to make word cloud about most important words in violations(graph attached in appendix) while using BiGrams to make this network graph. For color selection, I took light blue on node and black on edge which are distinct on white background. For node size, I choose size 5 which is proper represent the location of each node. To show main but also complete relationships I choose words occur more than 1500 times.

R code:

```
#For final project
library(tidyverse)
library(leaflet)
library(tidytext)
library(lubridate)
library(wordcloud)
library(igraph)
library(ggraph)
library(DT)
library(tm)
library(caret)
####Preparation: dataset processing####
FIR = read.csv("~/desktop/ProjData/FoodInspection_17R.csv")
FIR_Network <- FIR[,c('Inspection.ID','Risk','Results','Violations')]
#text data processing
FIR_Network$Violations <- tolower(FIR_Network$Violations)
FIR_Network$Violations <- str_replace_all(FIR_Network$Violations,
"[[:punct:]]", "")
FIR_Network$Violations <- str_replace_all(FIR_Network$Violations,
"[[:digit:]]", "")
gsub(paste0(StopWordsCustom,collapse = "|"),"", FIR_Network$Violations)
FIR_Network$Violations<- gsub(paste0(StopWordsCustom,collapse = "|"),"",
```

```
FIR_Network$Violations)

write.csv(FIR_Network, "~/desktop/ProjData/FIR_Network.csv")

#### Main Graph Making ######
fillColor = "#FFA07A"
fillColor2 = "#F1C40F"
FoodInspections = read_csv("~/desktop/ProjData/FIR_Network.csv")
FoodInspectionsReduced = FoodInspections %>%
  mutate(InspectionID = `Inspection.ID`) %>%
  select(InspectionID,Violations,Results)
FoodInspectionWords <- FoodInspectionsReduced %>%
  unnest_tokens(word, Violations) %>%
  filter(!word %in% stop_words$word) %>%
  count(Results, word, sort = TRUE) %>%
  ungroup()

#Term Frequency of Words (TF)
FoodInspectionResultsWords <- FoodInspectionWords
TotalWordsPerResult <- FoodInspectionResultsWords %>%
  group_by(Results) %>%
  summarize(total = sum(n))
FoodInspectionResultsWords <- left_join(FoodInspectionResultsWords,
TotalWordsPerResult)
FoodInspectionResultsWords = FoodInspectionResultsWords %>% filter
(!is.na(Results))
ggplot(FoodInspectionResultsWords, aes(n/total, fill = Results)) +
  geom_histogram(bins = 30, show.legend = FALSE) +
  xlim(NA, 0.0009) +
  facet_wrap(~Results, ncol = 2, scales = "free_y") + theme_bw()

#TF-IDF of Unigrams (One Word)
FoodInspectionWords_TF_IDF <- FoodInspectionWords  %>%
  bind_tf_idf(word, Results, n)
#Choose words with low IDF
LowIDF = FoodInspectionWords_TF_IDF %>%
  arrange((idf)) %>%
  select(word,idf)
#Get the Unique Words with LowIDF
UniqueLowIDF = unique(LowIDF$word)
plot_FoodInspectionWords_TF_IDF <- FoodInspectionWords_TF_IDF %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word))))

#plot 1: Word cloud for UniGrams
plot_FoodInspectionWords_TF_IDF2 = plot_FoodInspectionWords_TF_IDF %>%
top_n(100)
plot_FoodInspectionWords_TF_IDF2 %>%
  with(wordcloud(word, tf_idf, max.words = 50,colors=brewer.pal(8, "Dark2")))

#TF-IDF Bigrams
FoodInspectionWordsBiGram <- FoodInspectionsReduced %>%
  unnest_tokens(bigram, Violations, token = "ngrams", n = 2)
bigrams_separated <- FoodInspectionWordsBiGram %>%
  separate(bigram, c("word1", "word2"), sep = " ")
bigrams_filtered <- bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
```

```
  filter(!word2 %in% stop_words$word)
# new bigram counts:
bigram_counts <- bigrams_filtered %>%
  count(word1, word2, sort = TRUE)
bigrams_united <- bigrams_filtered %>%
  unite(bigram, word1, word2, sep = " ")
bigram_tf_idf <- bigrams_united %>%
  count(Results, bigram) %>%
  bind_tf_idf(bigram, Results, n)
plot_FoodInspectionWords_TF_IDF <- bigram_tf_idf %>%
  arrange(desc(tf_idf)) %>%
  mutate(bigram = factor(bigram, levels = rev(unique(bigram))))

#plot 2: Force-directed graph for relationship among words
count_bigrams <- function(dataset) {
  dataset %>%
    unnest_tokens(bigram, Violations, token = "ngrams", n = 2) %>%
    separate(bigram, c("word1", "word2"), sep = " ") %>%
    filter(!word1 %in% stop_words$word,
           !word2 %in% stop_words$word) %>%
    count(word1, word2, sort = TRUE)
}
visualize_bigrams <- function(bigrams) {
  set.seed(2016)
  a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
  bigrams %>%
    graph_from_data_frame() %>%
    ggraph(layout = "fr") +
    geom_edge_link(aes(edge_alpha = n), show.legend = FALSE, arrow = a) +
    geom_node_point(color = "lightblue", size = 5) +
    geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
    labs(title = "Relationship Between Keywords of Restaurant Violations")+
    theme_void()
}
visualize_bigrams_individual <- function(bigrams) {
  set.seed(2016)
  a <- grid::arrow(type = "closed", length = unit(.15, "inches"))
  bigrams %>%
    graph_from_data_frame() %>%
    ggraph(layout = "fr") +
    geom_edge_link(aes(edge_alpha = n), show.legend = FALSE, arrow =
a,end_cap = circle(.07, 'inches')) +
    geom_node_point(color = "lightblue", size = 5) +
    geom_node_text(aes(label = name), vjust = 1, hjust = 1) +
    theme_void()
}
FoodInspectionsReduced_Bigrams <- FoodInspectionsReduced %>%
  count_bigrams()
FoodInspectionsReduced_Bigrams %>%
  filter(n > 1500) %>%
  visualize_bigrams()

####Other Exploratory Graphs####
#Plot 3: Bar graph for Top 20 bad-performing restaurants(with high risk)
cData%>%
  select(DBA.Name,Risk,Latitude,Longitude)%>%
  group_by(DBA.Name,Risk,Latitude,Longitude)%>%
```

```
  summarise(Tot=n())->res_by_high_risk

res_by_high_risk<-arrange(res_by_high_risk,-Tot)
write.csv(res_by_high_risk,'~/desktop/rbhr.csv',sep=",", row.names=FALSE)
# use tableau to graph

#Plot 4:choropleth for discribtion of high risk restaurants by zip code
# use tableau to graph

#plot 5: line graph for time series about inspection frequency
hchart(tseries, name = "test") %>%
  hc add theme(hc theme 538()) %>%
  hc_credits(enabled = TRUE, style = list(fontSize = "13px")) %>%
  hc_title(text = "line graph for time series about inspection frequency in
2015-2017") %>%
  hc_legend(enabled = TRUE)

#Plot 6:leaflet map of food place broken down by results(pass and fail)
ResultsPassORFail = c("Pass","Fail")
factpal <-
colorFactor(c("gray","red","purple","yellow","orange","green","blue"),
                        FoodInspections$Results)
FoodInspectionsSubSet = FoodInspections %>%
  sample_n(8e3) %>%
  filter(Results %in%  ResultsPassORFail)
leaflet(FoodInspectionsSubSet) %>% addProviderTiles("Esri.NatGeoWorldMap")
%>%
  addCircles(lng = ~Longitude, lat = ~Latitude,radius = 1,
             color = ~factpal(Results))  %>%
  addLegend("bottomright", pal = factpal, values = ~Results,
            title = "Locations of Food Places in Chicago",
            opacity = 1)
```