Exploratory Data Analysis: Health Outcomes and NYC Car Crashes

Tai Chou-Kudu

Overview

This is a two part lab where each part will focus on a different dataset: the first part will use a dataset containing a series of diagnostic measurements taken on members of the Akimel O'odham people (an indigenous group living in the Southwestern United States who are also called the Pima) to understand diabetes risk (click here to download diabetes.csv), and the second dataset contains information on traffic accidents in New York City in the months of July and August of this year, and was compiled by NYC Open Data (click here to download crashes.csv).

For this problem set you will need to install the `skimr` and `GGally` packages, and in particular the functions `skim` and `ggpairs`.

```r
#|echo: FALSE
library(tidyr)
library(readr)
library(ggplot2)
library(dplyr)
library(janitor)
library(skimr)
library(GGally)
```

We will also explore the concept of an *inlier*, which is an erroneous value that occurs in the interior of the distribution of a variable, rather than in the tails of the variable. The US Census published an article on the problem of inliers here

Part 1: Health Diagnostics and Diabetes Incidence

Problem 1: Data Description and Outliers.

Load `diabetes.csv` into R and take a look at the data using the `skimr` package (make sure to install it if you don't have it). Skimr provides a tidy summary function called `skim`. Use `skim` on the data frame that you loaded from diabetes.csv.

```r
diabetes <- readr::read_csv(here::here('diabetes.csv'))
```

```r
cat("\\footnotesize\n")
```

```r
skim(diabetes)
```

Table 1: Data summary

Name	diabetes
Number of rows	768
Number of columns	9
Column type frequency:	
numeric	9
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Pregnancies	0	1	3.85	3.37	0.00	1.00	3.00	6.00	17.00	
Glucose	0	1	120.89	31.97	0.00	99.00	117.00	140.25	199.00	
BloodPressure	0	1	69.11	19.36	0.00	62.00	72.00	80.00	122.00	
SkinThickness	0	1	20.54	15.95	0.00	0.00	23.00	32.00	99.00	
Insulin	0	1	79.80	115.24	0.00	0.00	30.50	127.25	846.00	
BMI	0	1	31.99	7.88	0.00	27.30	32.00	36.60	67.10	
DiabetesPedigreeFunction	0	1	0.47	0.33	0.08	0.24	0.37	0.63	2.42	
Age	0	1	33.24	11.76	21.00	24.00	29.00	41.00	81.00	
Outcome	0	1	0.35	0.48	0.00	0.00	0.00	1.00	1.00	

```r
cat("\\footnotesize\n")
```

Skim will list several variables. Pregnancies is the past number of pregnancies (this dataset includes women 21 years or older), glucose describes the concentration of glucose in the blood after an oral glucose tolerance test (drinking a sugary drink and measuring two hours later), skin thickness is the result of a skinfold thickness test taken at the triceps (upper arm), Insulin is the insulin concentration in the blood taken at the same time as the glucose measurement (Insulin is a hormone that transports glucose into cells), BMI is "Body Mass Index", Diabetes Pedigree Function is a measure of diabetes risk based on the family history of diabetes for each patient (this is an engineered feature) and outcome is equal to 1 if the patient was diagnosed with diabetes with 5 years and 0 otherwise.

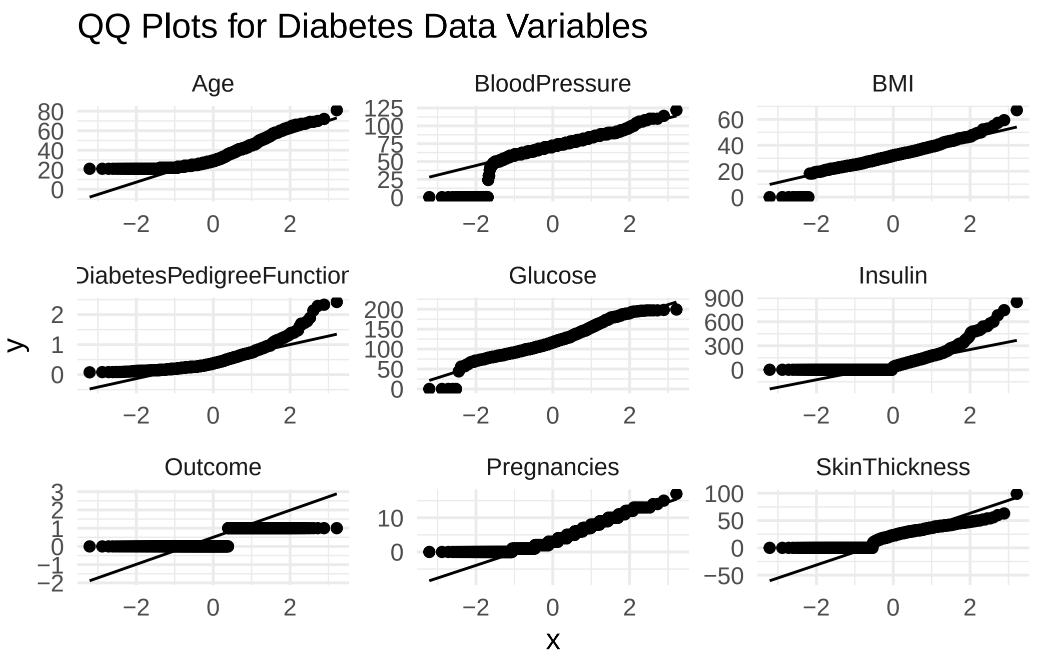
a) Skim should show no missing data, but should indicate potential data issues. Do any of the percentile ranges (p0, p25, p50, p75, or p100) for the reported variables suggest a potential problem?

The variable: Skin thickness is likely missing values, since a skin thickness value of 0 seems unrealistic. The same goes for BMI and blood pressure variables.

b) Further investigate the dataset to find potentially problematic variables using a qq-plot (`geom_qq`) or `group_by` combined with `count` and `arrange`. For which variables do you find repeated values and what are those values? Do you believe these values represent real measurements or could they correspond to missing data? Do the repeated variables occur in the same rows or different rows?

```
long_data <- diabetes %>%
  pivot_longer(cols = where(is.numeric), names_to = "variable", values_to = "value")

ggplot(long_data, aes(sample = value)) +
  geom_qq() +
  geom_qq_line() +
  facet_wrap(~ variable, scales = "free") +
  theme_minimal() +
  ggtitle("QQ Plots for Diabetes Data Variables")
```



```
diabetes %>%
  group_by(BloodPressure) %>%
  count() %>%
  arrange(BloodPressure)
```

```
# A tibble: 47 x 2
# Groups:   BloodPressure [47]
   BloodPressure     n
```

```
            <dbl> <int>
 1              0    35
 2             24     1
 3             30     2
 4             38     1
 5             40     1
 6             44     4
 7             46     2
 8             48     5
 9             50    13
10             52    11
# i 37 more rows
```

```r
diabetes %>%
  group_by(SkinThickness) %>%
  count() %>%
  arrange(SkinThickness)
```

```
# A tibble: 51 x 2
# Groups:   SkinThickness [51]
   SkinThickness     n
           <dbl> <int>
 1             0   227
 2             7     2
 3             8     2
 4            10     5
 5            11     6
 6            12     7
 7            13    11
 8            14     6
 9            15    14
10            16     6
# i 41 more rows
```

```r
diabetes %>%
  group_by(BMI) %>%
  count() %>%
  arrange(BMI)
```

```
# A tibble: 248 x 2
# Groups:   BMI [248]
     BMI     n
   <dbl> <int>
 1   0      11
 2  18.2     3
 3  18.4     1
 4  19.1     1
 5  19.3     1
 6  19.4     1
 7  19.5     2
 8  19.6     3
 9  19.9     1
10  20       1
# i 238 more rows
```

```
repeated_rows <- diabetes %>%
  filter(BloodPressure == 0 | BMI == 0 | SkinThickness == 0)

print(repeated_rows)
```

```
# A tibble: 231 x 9
   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
         <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>
 1           8     183            64             0       0  23.3
 2           5     116            74             0       0  25.6
 3          10     115             0             0       0  35.3
 4           8     125            96             0       0   0
 5           4     110            92             0       0  37.6
 6          10     168            74             0       0  38
 7          10     139            80             0       0  27.1
 8           7     100             0             0       0  30
 9           7     107            74             0       0  29.6
10           8      99            84             0       0  35.4
# i 221 more rows
# i 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <dbl>
```

Write an overview of which values are missing and replace all missing values with NA for the next stage of analysis.

```
diabetes_cleaned <- diabetes %>%
  mutate(across(c(SkinThickness, BloodPressure), ~ na_if(.,0)))
```

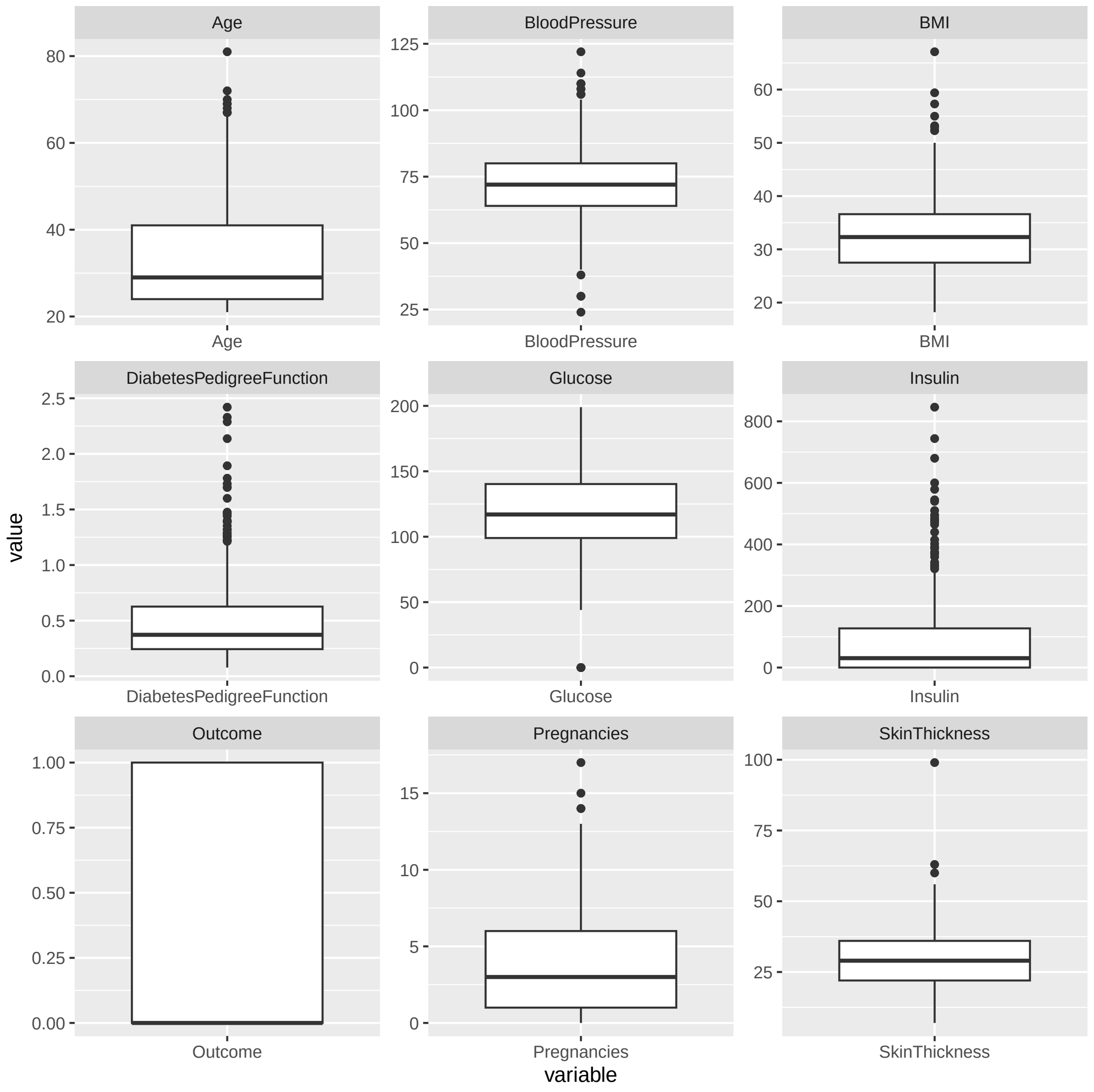
```
diabetes_cleaned <- diabetes_cleaned %>%
  mutate(BMI = na_if(BMI,0.0))
```

BMI, Skin Thickness, and Blood Pressure have missing values, as shown by the QQ Plots and further investigation by grouping by / count / arrange, and manually inspecting rows with 0 values.

c) Perform Tukey Box plots on each variable to identify potential outliers. Which variables have the most outliers? Are there any outliers that you think come from measurement error? If so remove them.

```
long_data_cleaned <- diabetes_cleaned %>%
  pivot_longer(cols = where(is.numeric), names_to = "variable", values_to = "value")

ggplot(long_data_cleaned, aes(x = variable, y = value)) +
  geom_boxplot() +
  facet_wrap(~ variable, scales = "free", nrow = 3)
```



The variables: Insulin and Diabetes Pedigree Function have the most outliers. I don't know enough about insulin or how it's measured to make a decision to remove outliers. Same with DPF. Someone with medical domain knowledge or research study-specific knowledge would have to give their input in order to enact further decisions. Skin Thickness outliers make sense to remove, because how much can that truly vary? An outlier of 100, while the median is closer to 30, sounds like a huge gap. It could be a measurement error.

```r
diabetes_cleaned %>%
  group_by(SkinThickness) %>%
  count() %>%
  arrange(desc(SkinThickness))
```

```
# A tibble: 51 x 2
# Groups:   SkinThickness [51]
   SkinThickness     n
           <dbl> <int>
 1            99     1
 2            63     1
 3            60     1
 4            56     1
 5            54     2
 6            52     2
 7            51     1
 8            50     3
 9            49     3
10            48     4
# i 41 more rows
```

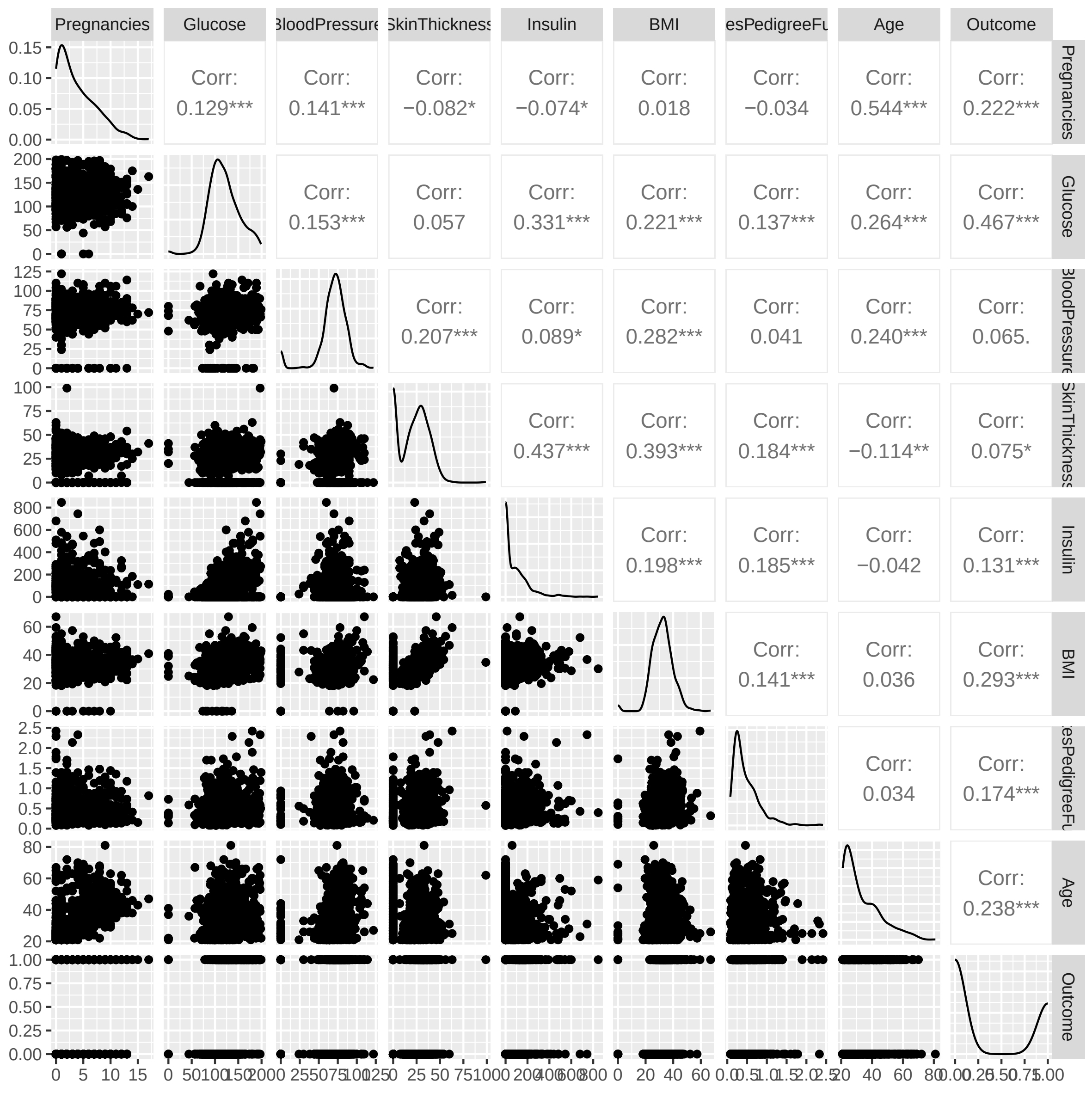
```
diabetes_cleaned <- diabetes_cleaned %>%
  mutate(SkinThickness = na_if(SkinThickness,99))
```

Problem 2: Pair Plots

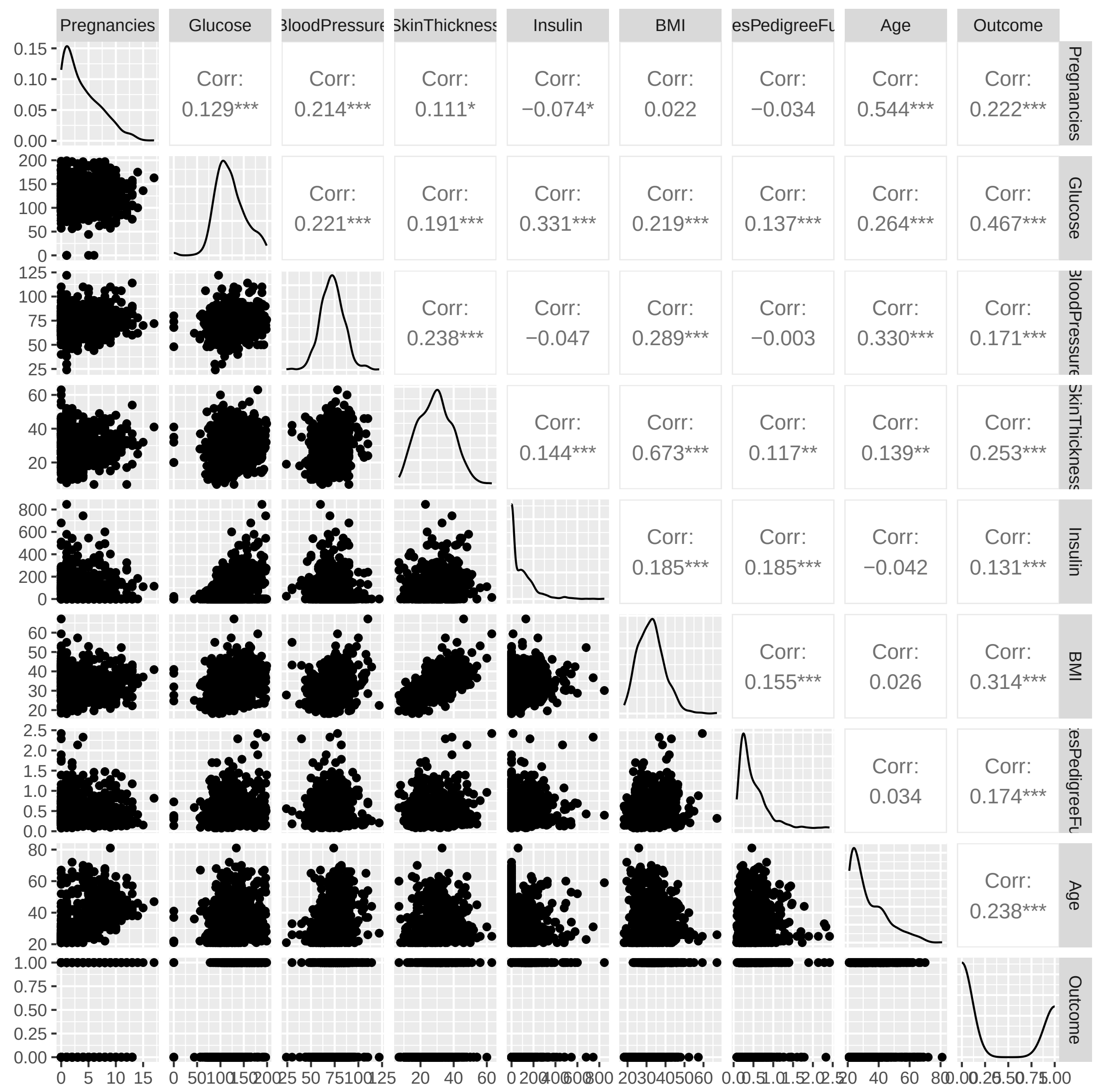
Use the `GGally` package and its function `ggpair` on both the original dataset and the cleaned dataset. Which correlations change the most? What are the strongest correlations between variables overall and with the `Outcome`?

- Remark: This dataset has been used an model dataset for the construction of binary classifiers using machine learning and there are a large number of published studies showing these analyses. However, many of these analyses did not exclude the missing values erroneously coded as zero, as is discussed in this interesting paper by Breault, leading to highly degraded accuracy.

```
ggpairs(diabetes)
```



```
ggpairs(diabetes_cleaned)
```



Skin Thickness's correlations changed a lot. BMI and Skin Thickness's correlation increased by ~ 0.3. S.T.'s correlations with glucose and blood pressure also increased. S.T.'s correlation with insulin decreased.

Glucose and BMI have the strongest correlation with outcome. Skin thickness, age, and pregnancies also have notable correlations, but not as strong compared to glucose and BMI.

Part 2: Car Crashes in NYC

Problem 3: Finding Inliers and Missing Data

Load the NYC car crash dataset using `read_csv`. You can download the data from the course website by clicking here.

```r
crashes <- readr::read_csv(here::here('crashes.csv'))
crashes <- janitor::clean_names(crashes)
```

```r
cat("\\scriptsize\n")
```

```r
skim_summary <- skim(crashes)
skim_summary %>%
  dplyr::filter(n_missing > 0)
```

Table 3: Data summary

Name	crashes
Number of rows	14720
Number of columns	29
Column type frequency:	
character	15
numeric	3
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
borough	4559	0.69	5	13	0	5	0
location	9120	0.38	10	23	0	4754	0
on_street_name	4179	0.72	6	32	0	2167	0
cross_street_name	7440	0.49	6	32	0	2342	0
off_street_name	10541	0.28	10	36	0	4037	0
contributing_factor_vehicle_1	90	0.99	5	53	0	51	0
contributing_factor_vehicle_2	3305	0.78	5	53	0	35	0
contributing_factor_vehicle_3	13349	0.09	11	30	0	15	0
contributing_factor_vehicle_4	14381	0.02	11	30	0	9	0
contributing_factor_vehicle_5	14610	0.01	11	30	0	4	0
vehicle_type_code_1	214	0.99	2	38	0	117	0
vehicle_type_code_2	4769	0.68	2	35	0	140	0
vehicle_type_code_3	13471	0.08	2	35	0	34	0
vehicle_type_code_4	14401	0.02	4	35	0	12	0
vehicle_type_code_5	14613	0.01	5	35	0	4	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
zip_code	4560	0.69	10889.33	533.32	10001.00	10456.00	11208.00	11238.00	11697.0	
latitude	9120	0.38	39.99	5.42	0.00	40.66	40.72	40.77	41.4	
longitude	9120	0.38	-72.58	9.84	-74.33	-73.96	-73.92	-73.86	0.0	

```r
cat("\\scriptsize\n")
```

```r
cat("\\scriptsize\n")
```

```r
skim(crashes)
```

Table 6: Data summary

Name	crashes
Number of rows	14720
Number of columns	29
Column type frequency:	
character	16
difftime	1
numeric	12
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
crash_date	0	1.00	10	10	0	61	0
borough	4559	0.69	5	13	0	5	0
location	9120	0.38	10	23	0	4754	0
on_street_name	4179	0.72	6	32	0	2167	0
cross_street_name	7440	0.49	6	32	0	2342	0
off_street_name	10541	0.28	10	36	0	4037	0
contributing_factor_vehicle_1	90	0.99	5	53	0	51	0
contributing_factor_vehicle_2	3305	0.78	5	53	0	35	0
contributing_factor_vehicle_3	13349	0.09	11	30	0	15	0
contributing_factor_vehicle_4	14381	0.02	11	30	0	9	0
contributing_factor_vehicle_5	14610	0.01	11	30	0	4	0
vehicle_type_code_1	214	0.99	2	38	0	117	0
vehicle_type_code_2	4769	0.68	2	35	0	140	0
vehicle_type_code_3	13471	0.08	2	35	0	34	0
vehicle_type_code_4	14401	0.02	4	35	0	12	0
vehicle_type_code_5	14613	0.01	5	35	0	4	0

Variable type: difftime

skim_variable	n_missing	complete_rate	min	max	median	n_unique
crash_time	0	1	0 secs	86340 secs	50820 secs	1405

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
zip_code	4560	0.69	10889.33	533.32	10001.00	10456.00	11208.00	11238.00	11697.0	
latitude	9120	0.38	39.99	5.42	0.00	40.66	40.72	40.77	41.4	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
longitude	9120	0.38	-72.58	9.84	-74.33	-73.96	-73.92	-73.86	0.0	
number_of_persons_injured	0	1.00	0.63	0.91	0.00	0.00	0.00	1.00	12.0	
number_of_persons_killed	0	1.00	0.00	0.06	0.00	0.00	0.00	0.00	4.0	
number_of_pedestrians_injured	0	1.00	0.09	0.30	0.00	0.00	0.00	0.00	7.0	
number_of_pedestrians_killed	0	1.00	0.00	0.04	0.00	0.00	0.00	0.00	4.0	
number_of_cyclist_injured	0	1.00	0.07	0.26	0.00	0.00	0.00	0.00	2.0	
number_of_cyclist_killed	0	1.00	0.00	0.01	0.00	0.00	0.00	0.00	1.0	
number_of_motorist_injured	0	1.00	0.45	0.90	0.00	0.00	0.00	1.00	12.0	
number_of_motorist_killed	0	1.00	0.00	0.05	0.00	0.00	0.00	0.00	2.0	
collision_id	0	1.00	4745479.90	4398.30	4737173.00	4741762.75	4745506.50	4749205.25	4755325.0	

```r
cat("\\scriptsize\n")
```

a) Which variables have missing data (use `skim` or another tool of your choosing)? Some missing values have a different interpretation than others- what does it mean when `VEHICLE TYPE CODE 2` is missing compared to `LATITUDE`?

   Borough, Location, On Street Name, Cross Street Name, Off Street Name, Contributing Vehicle 1-5 (5 variables), Vehicle Type Code 1-5 (5 variables), Zip Code, Latitude, and Longitude. Vehicle Type Code 2 missing may not indicate missing data, perhaps a vehicle is assigned a category from 1 - 5 based on its type. Missing latitude, however, may indicate missing the data, the GPS coordinates.

b) Latitude and Longitude have the same number of missing values. Verify that they always occur in the same row. Check the counts of latitude and longitude values- do you find any hidden missing values? If so recode them as NA.

```r
crashes %>%
  group_by(latitude) %>%
  count() %>%
  arrange(desc(n))
```

```
# A tibble: 4,591 x 2
# Groups:   latitude [4,591]
   latitude     n
      <dbl> <int>
 1     NA    9120
 2      0     101
 3     40.7     7
 4     40.8     7
 5     40.6     6
 6     40.6     6
 7     40.7     6
 8     40.7     6
 9     40.7     6
10     40.8     6
# i 4,581 more rows
```

```r
crashes %>%
  group_by(longitude) %>%
  count() %>%
  arrange(desc(n))
```

```
# A tibble: 4,440 x 2
# Groups:   longitude [4,440]
   longitude     n
```

```
       <dbl> <int>
 1        NA  9120
 2         0   101
 3     -73.9    11
 4     -73.9     7
 5     -73.9     7
 6     -74.1     6
 7     -74.1     6
 8     -74.0     6
 9     -73.9     6
10     -73.9     6
# i 4,430 more rows
```

```r
crashes <- crashes %>%
  mutate(latitude = if_else(latitude == 0.00000, NA_real_, latitude))

crashes <- crashes %>%
  mutate(longitude = if_else(longitude == 0.00000, NA_real_, longitude))
```

```r
both_missing <- which(is.na(crashes$latitude) & is.na(crashes$longitude))
both_missing
```

```r
one_missing <- which(xor(is.na(crashes$latitude), is.na(crashes$longitude)))
one_missing
```

I recoded 0.00000 as NA for Latitude and Longitude. Rows with NA values were all the same rows for both columns.

a) Many of the geographic values are missing, but geographic information is redundant in multiple variables in the dataset. For example, with effort you could determine the borough of an accident from the zip code, the latitude and longitude, or the streets (not part of the assignment for this week). Consider the borough variable- what percentage of the missing values of borough have values present of *at least* one of zip code or latitude. What about if we include all the street name variables? What fraction of rows don't have any detailed location information (latitude, zip code, or street names)?

```r
percentage <- crashes %>%
  filter(is.na(borough)) %>%
  summarise(present_in_zip_or_lat = mean(!is.na(zip_code) | !is.na(latitude)) * 100) %>%
  pull(present_in_zip_or_lat)

print(percentage)
```

```
[1] 45.22922
```

```r
fraction_no_loc_info <- crashes %>%
  filter(is.na(borough)) %>%
  filter(!is.na(latitude) | !is.na(zip_code) |
         !is.na(on_street_name) | !is.na(cross_street_name) | !is.na(off_street_name)) %>%
  summarise(fraction = n() / sum(is.na(borough))) %>%
  pull(fraction)

print(fraction_no_loc_info)
```

```
[1] 1
```

45% of the rows with missing borough data have at least a zip code or latitude present, while 100% of rows with missing borough data have at least one location identifier, including street name data.

a) The `CRASH TIME` variable has no missing values. Compute the count of how many times each individual time occurs in the crash data set. This will suggest that there are some inliers in the data. Compute summary statistics on the count data, and determine how many inliers there are (define an inlier as a data value where the count is an outlier, i.e. the count of that value is greater than 1.5*IQR + P75, i.e. 1.5 times the interquartile range past the 75th percentile for the distribution of counts for values of that variable.) For which inliers do you believe the time is most likely to be accurate? For which is it least likely to be accurate and why do you think so?

```r
threshold <- crashes %>%
  group_by(crash_time) %>%
  count() %>%
  summarise(
    iqr_CT = IQR(n),
    p75_CT = quantile(n, 0.75),
    outlier_of_count = quantile(n, 0.75) + (1.5 * IQR(n))
  ) %>%
  pull(outlier_of_count)


inliers <- crashes %>%
  count(crash_time) %>%
  filter(n <= threshold)

print(inliers)
```

```
# A tibble: 1,405 x 2
   crash_time     n
   <time>     <int>
 1 00'00"       306
 2 01'00"        13
 3 02'00"         7
 4 03'00"         2
 5 04'00"         3
 6 05'00"        23
 7 06'00"         6
 8 07'00"         4
 9 08'00"        15
10 09'00"         5
# i 1,395 more rows
```

The time is least likely inaccurate for 00:00:00 entries. This could potentially be coded in if the time was unknown, pointing to potential missing data for some of the 306 entries. For the non 00:00:00 entries, time is more likely to be accurately coded, although entries with precise minutes may be more precise than potentially estimated times, directly on the hour.

Problem 4: Finding Patterns in the Data

Formulate a question about crash data in NYC and make visualizations to explore your question. It could be related to the geographical distribution of accidents, the timing of accidents, which types of vehicles lead to more or less dangerous accidents, or anything else you want. Write comments/notes describing your observations in each visualizations you create and mention how these observations impact your initial hypotheses.

Useful questions to consider when you observe a pattern:

- Could this pattern be due to coincidence (i.e. random chance)?
- How can you describe the relationship implied by the pattern?
- How strong is the relationship implied by the pattern?

- What other variables might affect the relationship?
- Does the relationship change if you look at individual subgroups of the data?

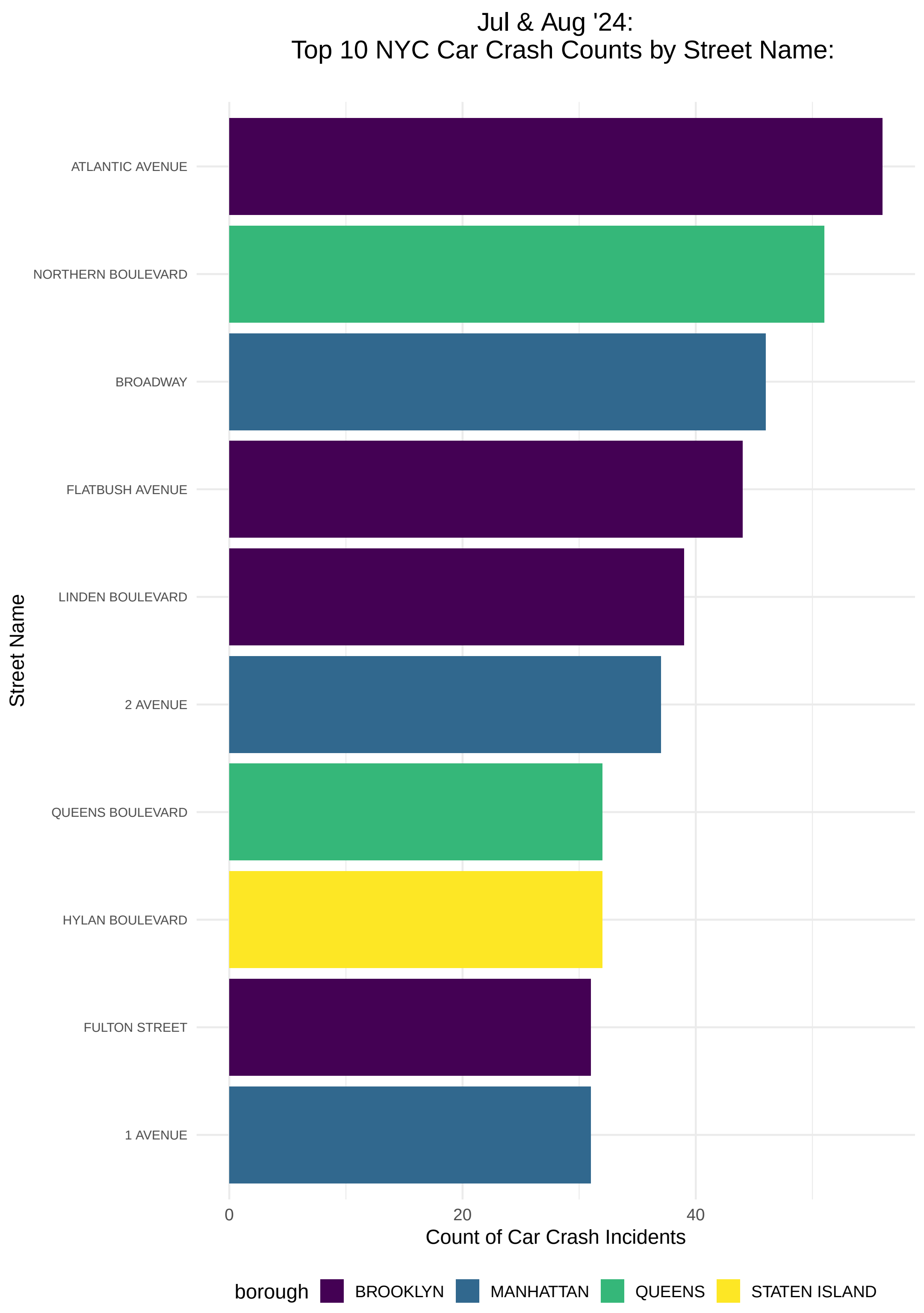
My question is: **Which streets have the highest recorded incidences of crashes? How can this information contribute to improving urban planning?**

```
crashes %>%
  filter(!is.na(borough)) %>%
  group_by(borough) %>%
  count() %>%
  arrange(desc(n))
```

```
# A tibble: 5 x 2
# Groups:   borough [5]
  borough           n
  <chr>         <int>
1 BROOKLYN       3476
2 QUEENS         2837
3 MANHATTAN      1846
4 BRONX          1619
5 STATEN ISLAND   383
```

```
result <- crashes %>%
  filter(!is.na(borough) & !is.na(on_street_name)) %>%
  group_by(borough, on_street_name) %>%
  count() %>%
  arrange(desc(n)) %>%
  ungroup() %>%
  slice_max(n, n = 10) %>%
  mutate(borough = as.factor(borough), on_street_name = as.factor(on_street_name))
```

```
ggplot(result, aes(x = reorder(on_street_name, n), y = n, fill = borough)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Jul & Aug '24:
  Top 10 NYC Car Crash Counts by Street Name:
       ",
       x = "Street Name",
       y = "Count of Car Crash Incidents") +
  scale_fill_viridis_d() +
  theme_minimal() +
  theme(legend.position = "bottom",
        legend.key.size = unit(0.5, "cm"),
        axis.text.y = element_text(size = 7),
        plot.title = element_text(size = 14, hjust = 0.5))
```



Jul & Aug '24:
Top 10 NYC Car Crash Counts by Street Name:
Street Name
ATLANTIC AVENUE
NORTHERN BOULEVARD
BROADWAY
FLATBUSH AVENUE
LINDEN BOULEVARD
2 AVENUE
QUEENS BOULEVARD
HYLAN BOULEVARD
FULTON STREET
1 AVENUE
0
20
40
Count of Car Crash Incidents
borough
BROOKLYN
MANHATTAN
QUEENS
STATEN ISLAND

We can see that Brooklyn has the highest number of crashes, while Staten Island has the lowest number of crashes. Brooklyn is home to the 4 streets with the highest car crash incidents. Atlantic Avenue and Fulton Street are certainly known to be busy areas, as a New Yorker. I don't have the domain knowledge or driving experience to know if there's tricky rotaries/ multiple-road lanes in these Top 10 street areas, but exploring road features would be the next direction I'd take this analysis in. If we explored time of crash, we could determine which streets have more crashes during daytime vs nighttime. High counts for nighttime crashes could point to a need for improved lighting by roads or in front of signs. Another important distinction to make would be what portion of crashes are caused by drunk-driving versus not. I could even turn the current box plot into a stacked box plot, with this information. Another interesting exploration would be to map socio-economic status of neighborhood/street and explore correlations with the Top 10 street names. The subway system often takes the longest to fix or upgrade in low socio economic neighborhoods. I'd hypothesize that some of these street names may be in low to mid level socio-economic neighborhoods. One last takeaway, as a resident of New York, is to be careful when walking by Atlantic Avenue area.