Lab3: Exploratory Data Analysis

Alex Ptacek

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`.

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

Load Packages

```
library(tidyverse)
library(skimr)
library(GGally)
library(ggpubr)
library(scales)
```

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.

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.

1. Load Data

```
diabetes <- read_csv("https://raw.githubusercontent.com/georgehagstrom/DATA607/refs/heads/ma
```

2. Run skim function on ***diabetes***

```
diabetes_skim <- skim(diabetes)
diabetes_skim[c(2,3:6)] |> print()
```

```
# A tibble: 9 x 5
  skim_variable            n_missing complete_rate numeric.mean numeric.sd
  <chr>                        <int>         <dbl>        <dbl>      <dbl>
1 Pregnancies                      0             1        3.85       3.37
2 Glucose                          0             1      121.        32.0
3 BloodPressure                    0             1       69.1       19.4
4 SkinThickness                    0             1       20.5       16.0
5 Insulin                          0             1       79.8      115.
6 BMI                              0             1       32.0        7.88
7 DiabetesPedigreeFunction         0             1        0.472      0.331
8 Age                              0             1       33.2       11.8
9 Outcome                          0             1        0.349      0.477
```

```
diabetes_skim[c(2,7:11)] |> print()
```

```
# A tibble: 9 x 6
  skim_variable      numeric.p0 numeric.p25 numeric.p50 numeric.p75 numeric.p100
  <chr>                   <dbl>       <dbl>       <dbl>       <dbl>        <dbl>
1 Pregnancies                 0           1           3           6           17
2 Glucose                     0          99         117         140.         199
3 BloodPressure               0          62          72          80          122
4 SkinThickness               0           0          23          32           99
```

```
5 Insulin                  0           0          30.5       127.          846
6 BMI                      0          27.3        32          36.6          67.1
7 DiabetesPedigreeF~       0.078       0.244       0.372       0.626         2.42
8 Age                     21          24          29          41            81
9 Outcome                  0           0           0           1             1
```

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?

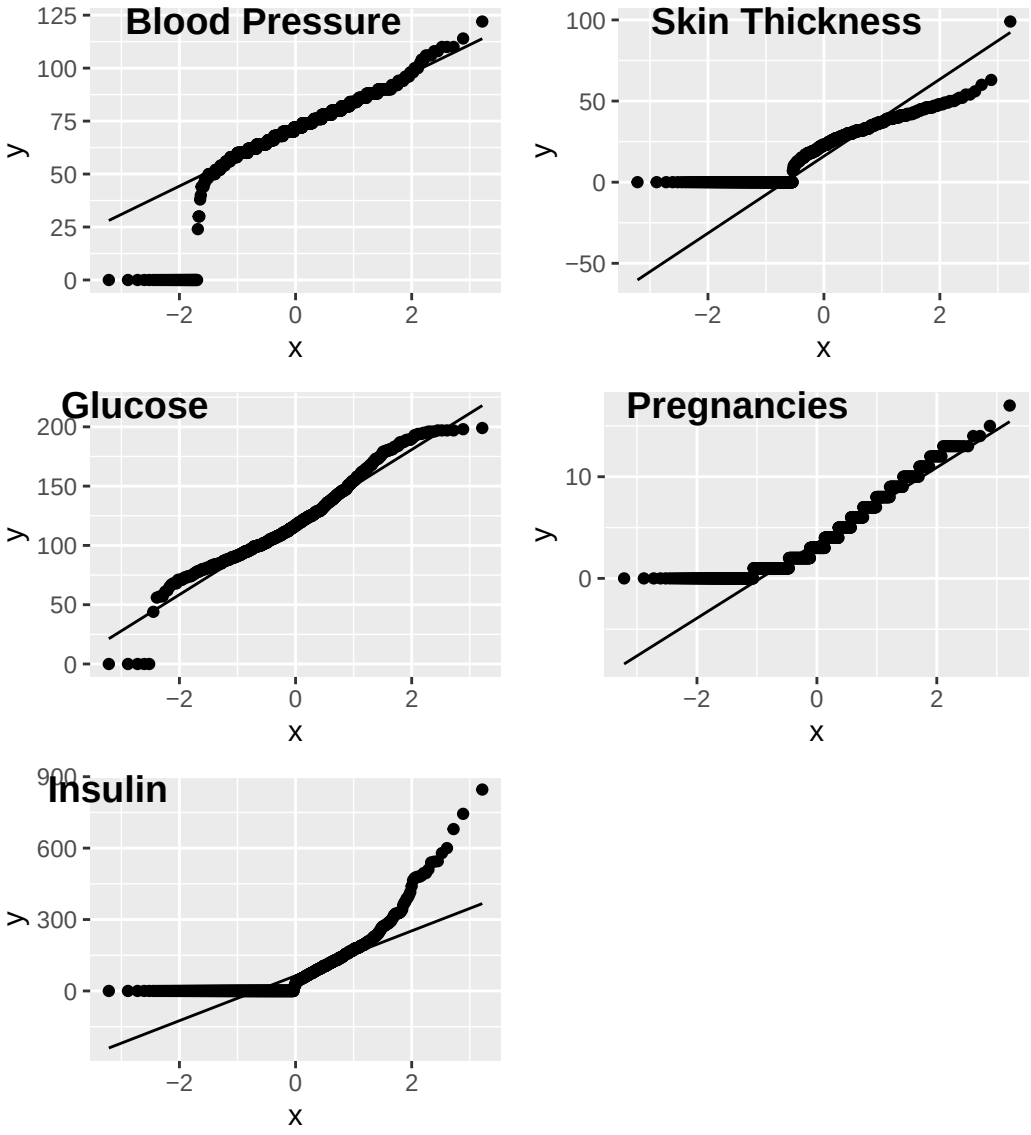
Answer: The highest amount of pregnancies being 17 seems too high. Blood pressure and skin thickness can never be 0, so that must be an error. Glucose and Insulin can be at 0 but it's apparently very rare, especially if they have a sugary drink before the test, so this is also suspicious.

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?

Answer: It seems my suspicions have been confirmed regarding blood pressure, skin thickness, glucose, and insulin measurements. For these variables, the 0 values appear to be clear inliers. Additionally, insulin seems to have outlier values >600. Pregnancies is more difficult to interpret. The 17 pregnancies value I called out does not appear to be an outlier. The 0 values look like inliers, but it's difficult to determine if these are actually errors, given that 46.8% of women in the U.S. were childless in the 2020 census. This repeated 0 value also seems to follow the trend of the pregancies q-q plot, as a whole, because there are many repeated values for almost every number of pregnancies.

```
bp_qq <- diabetes |>
  ggplot(aes(sample = BloodPressure)) +
  geom_qq()+
  geom_qq_line()

skin_thickness_qq <- diabetes |>
  ggplot(aes(sample = SkinThickness)) +
  geom_qq()+
  geom_qq_line()

glucose_qq <- diabetes |>
  ggplot(aes(sample = Glucose)) +
  geom_qq()+
  geom_qq_line()
```

```r
preg_qq <- diabetes |>
  ggplot(aes(sample = Pregnancies)) +
  geom_qq()+
  geom_qq_line()

insulin_qq <- diabetes |>
  ggplot(aes(sample = Insulin)) +
  geom_qq()+
  geom_qq_line()

ggarrange(bp_qq, skin_thickness_qq, glucose_qq, preg_qq, insulin_qq,
          labels = c("Blood Pressure", "Skin Thickness", "Glucose", "Pregnancies", "Insulin"),
          nrow = 3, ncol = 2)
```



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

Answer: Continuing from my previous analysis, I believe that all of the 0 values for blood pressure, skin thickness, glucose, and insulin are actually missing values. I have decided to leave all 0 values in for pregnancies, because it is not obvious that there are errors here.

```
diabetes_clean <- diabetes |>
  mutate(BloodPressure = if_else(BloodPressure == 0, NA, BloodPressure),
         SkinThickness = if_else(SkinThickness == 0, NA, SkinThickness),
```

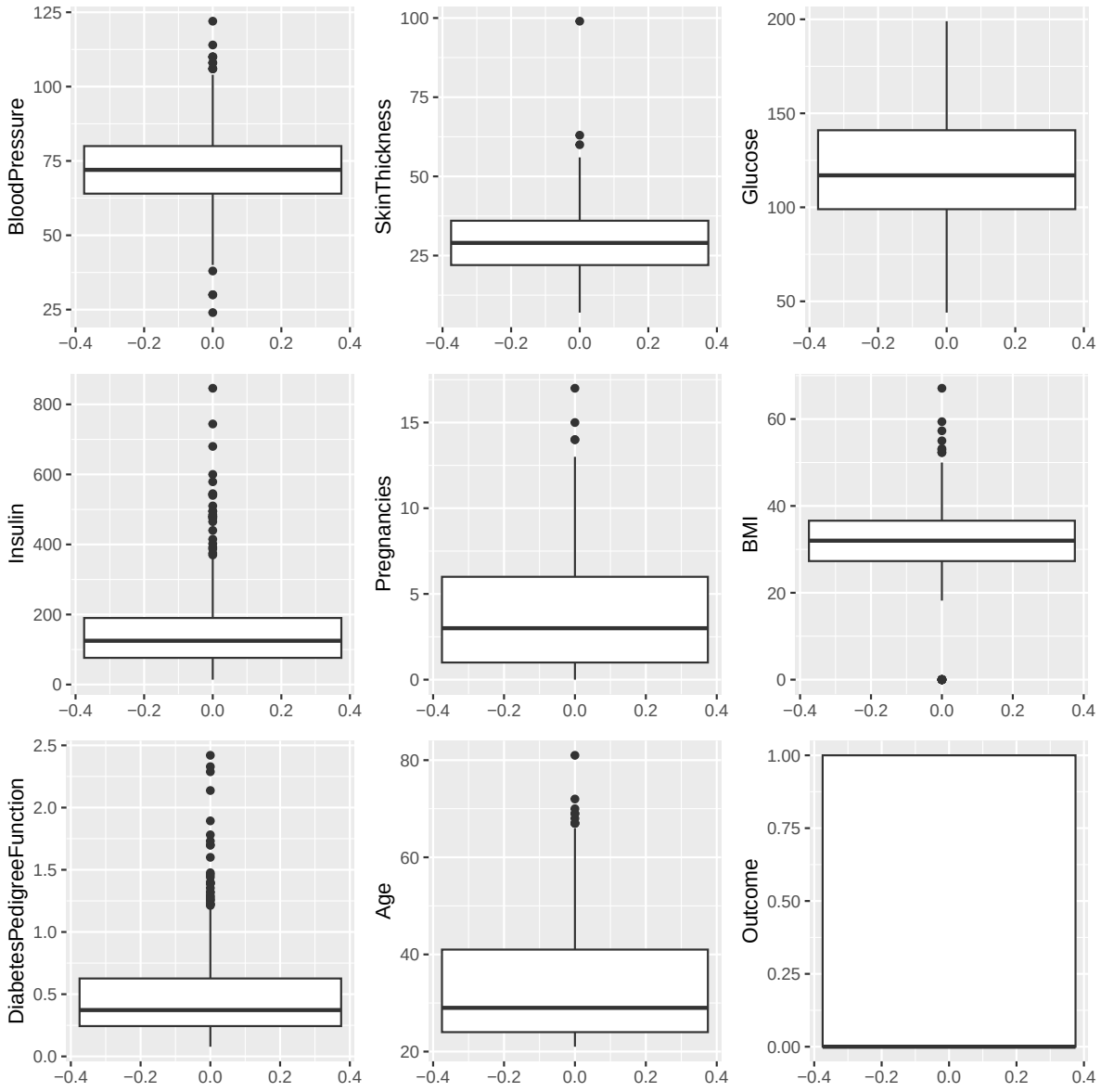
```
         Glucose = if_else(Glucose == 0, NA, Glucose),
         Insulin = if_else(Insulin == 0, NA, Insulin))
```

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.

Answer: Insulin has the most outliers. I think the blood pressure outliers on the lower end are probably from measurement error because those blood pressures seem too low.

```
bp_box <- diabetes_clean |>
  ggplot(aes(y = BloodPressure)) +
  geom_boxplot()

skin_thickness_box <- diabetes_clean |>
  ggplot(aes(y = SkinThickness)) +
  geom_boxplot()

glucose_box <- diabetes_clean |>
  ggplot(aes(y = Glucose)) +
  geom_boxplot()

insulin_box <- diabetes_clean |>
  ggplot(aes(y = Insulin)) +
  geom_boxplot()

preg_box <- diabetes_clean |>
  ggplot(aes(y = Pregnancies)) +
  geom_boxplot()

bmi_box <- diabetes_clean |>
  ggplot(aes(y = BMI)) +
  geom_boxplot()

diab_ped_funct_box <- diabetes_clean |>
  ggplot(aes(y = DiabetesPedigreeFunction)) +
  geom_boxplot()

age_box <- diabetes_clean |>
  ggplot(aes(y = Age)) +
  geom_boxplot()
```

```r
outcome_box <- diabetes_clean |>
  ggplot(aes(y = Outcome)) +
  geom_boxplot()

ggarrange(bp_box, skin_thickness_box, glucose_box, insulin_box, preg_box,
          bmi_box, diab_ped_funct_box, age_box, outcome_box,
          nrow = 3, ncol = 3)
```



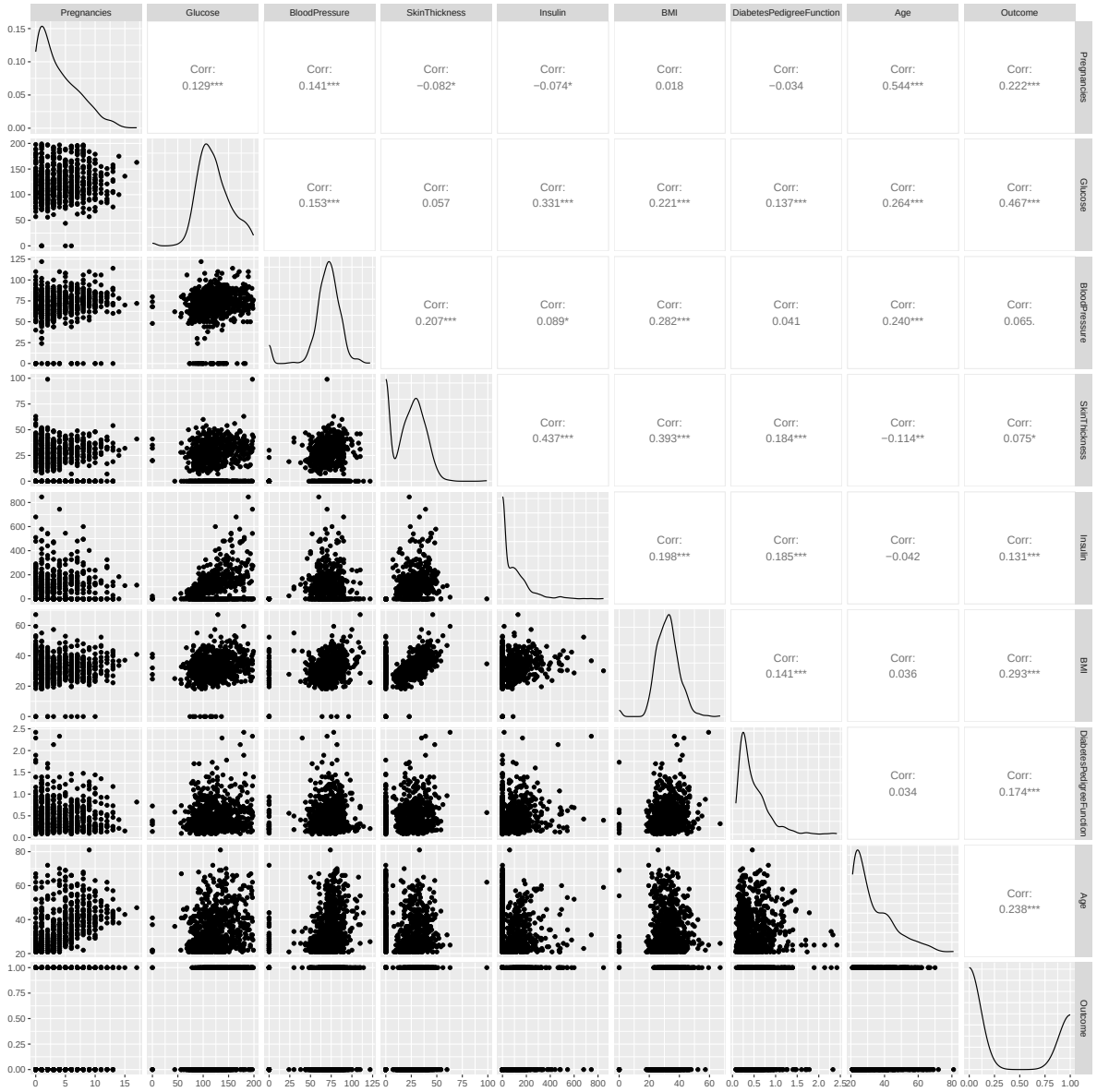
```r
diabetes_clean <- diabetes_clean |>
  mutate(BloodPressure = if_else(BloodPressure < 30, NA, BloodPressure)) |>
  drop_na()
```

Problem 2: Pair Plots

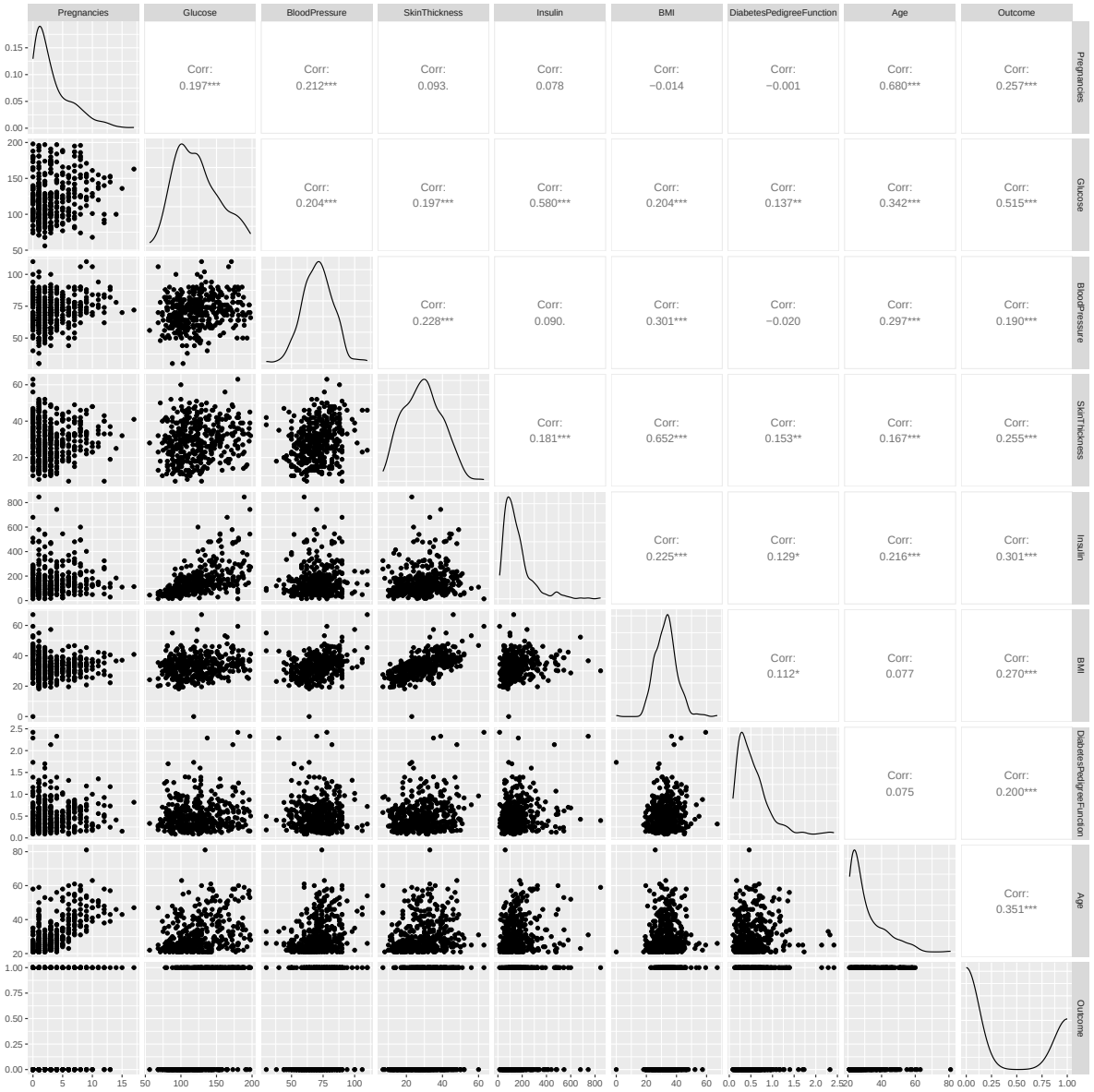
Use the `GGally` package and its function `ggpair` on both the original dataset and the cleaned dataset. Which correlations change the most? What are the strongest correlations between variables overall and with the `Outcome`?

*Answer: Some of the variables that change the most are Pregnancies and Blood Pressure, Outcome and Insulin, and Age and Insulin. The highest correlation overall is between Pregnancies and Age with a coefficient of 0.680. The highest correlation with Outcome is Glucose at 0.515.

```r
ggpairs(diabetes)
```




```
ggpairs(diabetes_clean)
```



- 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.

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.

```
nyc_car_crash <- read_csv("https://raw.githubusercontent.com/georgehagstrom/DATA607/refs/hea
```

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`?

Answer: Borough, Location, On Street Name, Cross Street Name, Off Street Name, Contributing Factor Vehicle (1, 2, 3, 4, 5), Vehicle Type Code (1, 2, 3, 4, 5), Zip Code, Latitude, and Longitude all have missing values. When `VEHICLE TYPE CODE 2` is missing, it could mean that there wasn't a second vehicle involved in the accident. When `LATITUDE` is missing, it means that the latitude coordinates are missing erroneously.

```
crash_skim <- skim(nyc_car_crash)
crash_skim[c(2,3:6)] |> print()
```

```
# A tibble: 29 x 5
   skim_variable             n_missing complete_rate character.min character.max
   <chr>                         <int>         <dbl>         <int>         <int>
 1 CRASH DATE                        0         1                10            10
 2 BOROUGH                        4559         0.690             5            13
 3 LOCATION                       9120         0.380            10            23
 4 ON STREET NAME                 4179         0.716             6            32
 5 CROSS STREET NAME              7440         0.495             6            32
 6 OFF STREET NAME               10541         0.284            10            36
 7 CONTRIBUTING FACTOR VEHI~        90         0.994             5            53
 8 CONTRIBUTING FACTOR VEHI~      3305         0.775             5            53
 9 CONTRIBUTING FACTOR VEHI~     13349         0.0931           11            30
10 CONTRIBUTING FACTOR VEHI~     14381         0.0230           11            30
# i 19 more rows
```

```
crash_skim[c(2,7:11)] |> print()
```

```
# A tibble: 29 x 6
   skim_variable         character.empty character.n_unique character.whitespace
   <chr>                           <int>             <int>                <int>
```

```
 1 CRASH DATE                          0                61                    0
 2 BOROUGH                             0                 5                    0
 3 LOCATION                            0              4754                    0
 4 ON STREET NAME                      0              2167                    0
 5 CROSS STREET NAME                   0              2342                    0
 6 OFF STREET NAME                     0              4037                    0
 7 CONTRIBUTING FACTOR ~               0                51                    0
 8 CONTRIBUTING FACTOR ~               0                35                    0
 9 CONTRIBUTING FACTOR ~               0                15                    0
10 CONTRIBUTING FACTOR ~               0                 9                    0
# i 19 more rows
# i 2 more variables: difftime.min <drtn>, difftime.max <drtn>
```

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.

Answer: All geographic coordinate NA's occur in both latitude and longitude. There are also missing values in the form of 0 that I changed to NA

```
#check observations where latitude is NA to see if there are any longitude values
nyc_car_crash |>
  filter(is.na(LATITUDE)) |>
  group_by(LATITUDE, LONGITUDE) |>
  summarise(n = n())
```

```
# A tibble: 1 x 3
# Groups:   LATITUDE [1]
  LATITUDE LONGITUDE     n
     <dbl>     <dbl> <int>
1       NA        NA  9120
```

```
#check observations where longitude is NA to see if there are any latitude values
nyc_car_crash |>
  filter(is.na(LONGITUDE)) |>
  group_by(LONGITUDE, LATITUDE) |>
  summarise(n = n())
```

```
# A tibble: 1 x 3
# Groups:   LONGITUDE [1]
  LONGITUDE LATITUDE     n
      <dbl>    <dbl> <int>
1        NA       NA  9120
```

```r
#check the most common coordinates for hidden missing values
nyc_car_crash |>
  group_by(LONGITUDE, LATITUDE) |>
  count() |>
  arrange(desc(n))
```

```
# A tibble: 4,755 x 3
# Groups:   LONGITUDE, LATITUDE [4,755]
   LONGITUDE LATITUDE     n
       <dbl>    <dbl> <int>
 1      NA       NA    9120
 2       0        0     101
 3     -73.9     40.7     7
 4     -73.9     40.8     7
 5     -74.1     40.6     6
 6     -74.1     40.6     6
 7     -74.0     40.8     6
 8     -73.9     40.8     6
 9     -73.9     40.8     6
10     -73.9     40.7     6
# i 4,745 more rows
```

```r
#replace all 0's in latitude and longitude with NA
nyc_car_crash_clean <- nyc_car_crash |>
  mutate(LATITUDE = if_else(LATITUDE == 0, NA, LATITUDE)) |>
  mutate(LONGITUDE = if_else(LONGITUDE == 0, NA, LONGITUDE))
```

c) 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)?

Answer: 45% of observations missing a Borough value have either a Zip Code of Latitude value. 97% have either Zip Code, Latitude, or On Street Name. 3% don't have any detailed information.

```r
nyc_car_crash_clean |>
  filter(is.na(BOROUGH)) |>
```

```
  group_by(BOROUGH) |>
  count()
```

```
# A tibble: 1 x 2
# Groups:   BOROUGH [1]
  BOROUGH     n
  <chr>   <int>
1 <NA>     4559
```

```
nyc_car_crash_clean |>
  filter(is.na(BOROUGH)) |>
  filter(`ZIP CODE` != is.na(`ZIP CODE`) |
           LATITUDE != is.na(LATITUDE)) |>
  group_by(BOROUGH) |>
  summarise(n = n())
```

```
# A tibble: 1 x 2
  BOROUGH     n
  <chr>   <int>
1 <NA>     2062
```

```
percent(2062/4559)
```

```
[1] "45%"
```

```
nyc_car_crash_clean |>
  filter(is.na(BOROUGH)) |>
  filter(`ZIP CODE` != is.na(`ZIP CODE`) |
           LATITUDE != is.na(LATITUDE) |
           `ON STREET NAME` != is.na(`ON STREET NAME`)) |>
  group_by(BOROUGH) |>
  summarise(n = n())
```

```
# A tibble: 1 x 2
  BOROUGH     n
  <chr>   <int>
1 <NA>     4417
```

```
percent(4417/4559)
```

```
[1] "97%"
```

```
nyc_car_crash_clean |>
  filter(is.na(BOROUGH), is.na(LATITUDE), is.na(`ZIP CODE`), is.na(`ON STREET NAME`)) |>
  count()
```

```
# A tibble: 1 x 1
      n
  <int>
1   142
```

```
percent(142/4559)
```

```
[1] "3%"
```

d) 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?

Answer: There are 199 values that qualify as inliers. I believe the values may be accurate for times with detail to the minute. The most common values are mostly rounded to the hour or half hour. I believe those times that are rounded probably contain inaccuracy, because it is common for people to round the time, even though it does not accurately refect the true time.

```
crash_clean_skim <- nyc_car_crash_clean |>
  group_by(`CRASH TIME`) |>
  summarise(n = n()) |>
  arrange(desc(n)) |>
  skim()

crash_clean_skim[c(2,3:6)] |> print()
```

```
# A tibble: 2 x 5
  skim_variable n_missing complete_rate difftime.min difftime.max
  <chr>             <int>         <dbl> <drtn>       <drtn>
1 CRASH TIME            0             1  0 secs      86340 secs
2 n                     0             1 NA secs         NA secs
```

```
crash_clean_skim[c(2,7:11)] |> print()
```

```
# A tibble: 2 x 6
  skim_variable difftime.median difftime.n_unique numeric.mean numeric.sd
  <chr>         <time>                      <int>        <dbl>      <dbl>
1 CRASH TIME    12:12                        1405         NA         NA
2 n                NA                          NA         10.5       18.1
# i 1 more variable: numeric.p0 <dbl>
```

```
crash_time_inliers <- nyc_car_crash_clean |>
  group_by(`CRASH TIME`) |>
  summarise(n = n()) |>
  arrange(desc(n)) |>
  filter(n > (1.5*6 + 9))
```

Problem 4: Finding Patterns in the Data

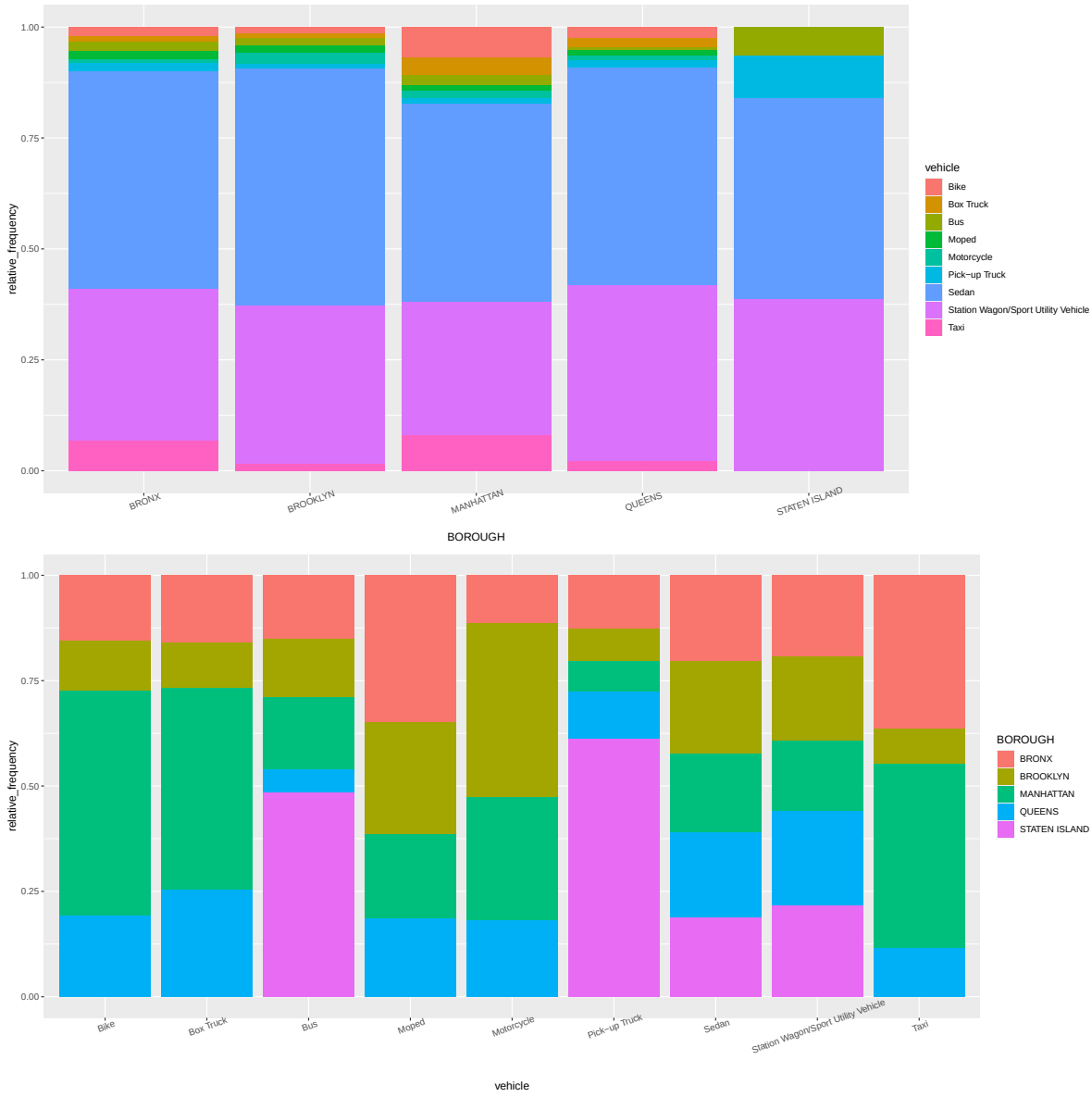
Formulate a question about crash data in NYC and make visualizations to explore your question. It could be related to the geographical distribution of accidents, the timing of accidents, which types of vehicles lead to more or less dangerous accidents, or anything else you want. Write comments/notes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

The question I would like to explore is what is the relationship between the borough that the accident occurred and the vehicle type that was involved. First, I found the top 10 most common vehicles involved in these accidents and filtered the original dataset to just these vehicles, creating a new dataset. Sedan and SUVs have the highest counts, by far, so I'm now expecting most accidents in each borough to be caused by those 2 vehicles. I decided to analyze the relative frequency of each of these top vehicles within each borough. To do that, I computed the total accidents per borough involving one of the top 10 vehicles, then created a table which calculated the relative frequency. Lastly, I created stacked bar charts comparing borough, vehicle, and relative frequency by proportion. In the top chart, it becomes clear that Sedans and SUVs were a large majority of accidents in all Boroughs. We can also see that taxis were pretty commonly involved in Manhattan and the Bronx, and Pick-up Trucks were relatively common in Staten Island. Based on the bottom graph, we also discover that accidents involving Sedans and SUVs were fairly evenly distributed across all Boroughs. Most

Pick-up Truck accidents occurred in Staten Island and most Bike and Box Truck accidents happened in Manhattan.

```
top_vehicles <- nyc_car_crash_clean |>
  group_by(`VEHICLE TYPE CODE 1`) |>
  summarise(n = n()) |>
  arrange(desc(n)) |>
  head(10) |>
  mutate(vehicle = `VEHICLE TYPE CODE 1`)

top_vehicle_crashes <- nyc_car_crash_clean |>
  filter(`VEHICLE TYPE CODE 1` == c(top_vehicles$vehicle))

top_vehicle_boroughs <- top_vehicle_crashes |>
  group_by(BOROUGH) |>
  summarise(n = n())

relative_frequency_crashes <- top_vehicle_crashes |>
  group_by(BOROUGH, `VEHICLE TYPE CODE 1`) |>
  summarise(n = n()) |>
  mutate(borough_total =
           case_when(
             BOROUGH == "BRONX" ~ 149,
             BOROUGH == "BROOKLYN" ~ 328,
             BOROUGH == "MANHATTAN" ~ 174,
             BOROUGH == "QUEENS" ~ 280,
             BOROUGH == "STATEN ISLAND" ~ 31,
             BOROUGH == NA ~ 420)) |>
  group_by(BOROUGH, `VEHICLE TYPE CODE 1`) |>
  mutate(relative_frequency = n/borough_total) |>
  mutate(vehicle = `VEHICLE TYPE CODE 1`) |>
  filter(BOROUGH != is.na(BOROUGH))

borough_makeup <- relative_frequency_crashes |>
  ggplot(aes(x = BOROUGH, y = relative_frequency, fill = vehicle)) +
  geom_bar(stat = "identity", position = "fill") +
  theme(axis.text.x = element_text(angle = 20))

vehicle_makeup <- relative_frequency_crashes |>
  ggplot(aes(x = vehicle, y = relative_frequency, fill = BOROUGH)) +
  geom_bar(stat = "identity", position = "fill") +
  theme(axis.text.x = element_text(angle = 20))
```

```
ggarrange(borough_makeup, vehicle_makeup,
          nrow = 2, ncol = 1)
```



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?