Diabetes Data Vizualization

Visualization Homework 2

Hannah Damico

17 Feb. 2023

Data Curation/ Data Mining Tasks

**Task 1. Combine all the data files into a single long file. Do this with code, not by copying and pasting. Show your code and include the name of the tool or language you used.* *

Tool/language used: Python Code:

```
import glob
import os

files = glob.glob("./Case_Studies_BIOS824/Diabetes-Data/*")

for file in files:
    new_file = file + ".txt"
    os.rename(file, new_file)

""" Read in Diabetes Data Text Files """

import pandas as pd
import os

def parse_data_Patient(file):
    df = pd.read_csv(
        file, sep="\t", header=None, names=["Date", "Time", "Code", "Value"]
    )
    # Could not convert to date because dates were in
    # different formats and had missing values
    # Commented out for this reason
    # df["Date"] = pd.to_datetime(df["Date"], format="%m-%d-%Y")
    # df["Time"] = pd.to_datetime(df["Time"], format="%H:%M").dt.time
    # df["Unique_ID"] = df["Date"].dt.strftime("%Y-%m-%d") + "_" + df["Code"].astype(str)

    return df

data_directory = "./Case_Studies_BIOS824/Diabetes-Data"
combined_df = pd.DataFrame()

for filename in os.listdir(data_directory):
    if filename.endswith(".txt"):
        file_path = os.path.join(data_directory, filename)
        patient_df = parse_data_Patient(file_path)
        combined_df = pd.concat([combined_df, patient_df])

combined_df.to_csv("combined_data.csv", index=False)
```

Import Data

```
combined_data <- read_csv("../combined_data.csv")
```

```
## Rows: 29330 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Date, Time, Value
## dbl (1): Code
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# View(combined_data)
str(combined_data)
```

```
## spc_tbl_ [29,330 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Date : chr [1:29330] "08-23-1990" "08-23-1990" "08-23-1990" "08-23-1990" ...
##  $ Time : chr [1:29330] "07:19" "07:23" "12:16" "12:35" ...
##  $ Code : num [1:29330] 59 33 60 33 62 70 65 33 64 58 ...
##  $ Value: chr [1:29330] "219" "7" "119" "5" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Date = col_character(),
##   ..   Time = col_character(),
##   ..   Code = col_double(),
##   ..   Value = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Adjust data structure

```
# Read Date as date
combined_data$Date <- mdy(combined_data$Date)
# Read Time as time
combined_data$Time <- hm(combined_data$Time)
# Read Value as numeric
combined_data$Value <- as.numeric(combined_data$Value)
```

Year	Total
1988	1051
1989	6245
1990	9514
1991	12480

Task 2. Sort the data so that the rows are in order, earliest date to latest date. Submit your date ordered dataset with your homework. Also submit your code, the language or tool you used for the sorting.

Tool/language used: R Studio

```
# Sort by descending Date
combined_data <- combined_data %>%
  arrange((Date))

# combined_data %>% arrange((Date))
```

Task 3. What is the count of the total number of measurements in the dataset?

ANSWER: 29,278

```
# Total number of measurements equals number of values that are not missing
combined_data %>%
  count(!is.na(Value))
```

```
## # A tibble: 2 x 2
##   `!is.na(Value)`     n
##   <lgl>           <int>
## 1 FALSE              52
## 2 TRUE            29278
```

Task 4. What is the total number of measurements per year in the dataset? List counts by year in a table and include a table with your answers.

ANSWER:
1988 - 1051
1989 - 6245
1990 - 9514
1991 - 12,480

```
# Save unique years to a vector
years <- combined_data %>%
  distinct(year(Date)) %>%
  drop_na() %>%
  as.vector()


combined_data %>%
  group_by(year(Date)) %>%
  rename("Year" = "year(Date)") %>%
  summarise(Total = n()) %>%
  drop_na() %>%
  kbl() %>%
  kable_classic(full_width = F)
```

Weekday	Total
Sunday	4055
Monday	4370
Tuesday	4253
Wednesday	4334
Thursday	4229
Friday	4131
Saturday	3918

Code	Total
33	9518
34	3830
58	3518
62	3160
60	2771
48	1883

Task 5. What is the count of the total number of measurements by day. That is, count of the tests done on a Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday. Include the table of the counts by day.

Tool/Language used: R, lubridate package

```
combined_data %>%
  group_by(wday(Date, label = TRUE, abbr = FALSE)) %>%
  summarise(Total = n()) %>%
  drop_na() %>%
  rename("Weekday" = `wday(Date, label = TRUE, abbr = FALSE)`) %>%
  kbl() %>%
  kable_classic(full_width = F)
```

Task 6. What is the count of the number of measurements by each test code field. List the counts by code field and describe the method you used and include any script you used.
The first 6 codes and frequencies are shown in the table below.

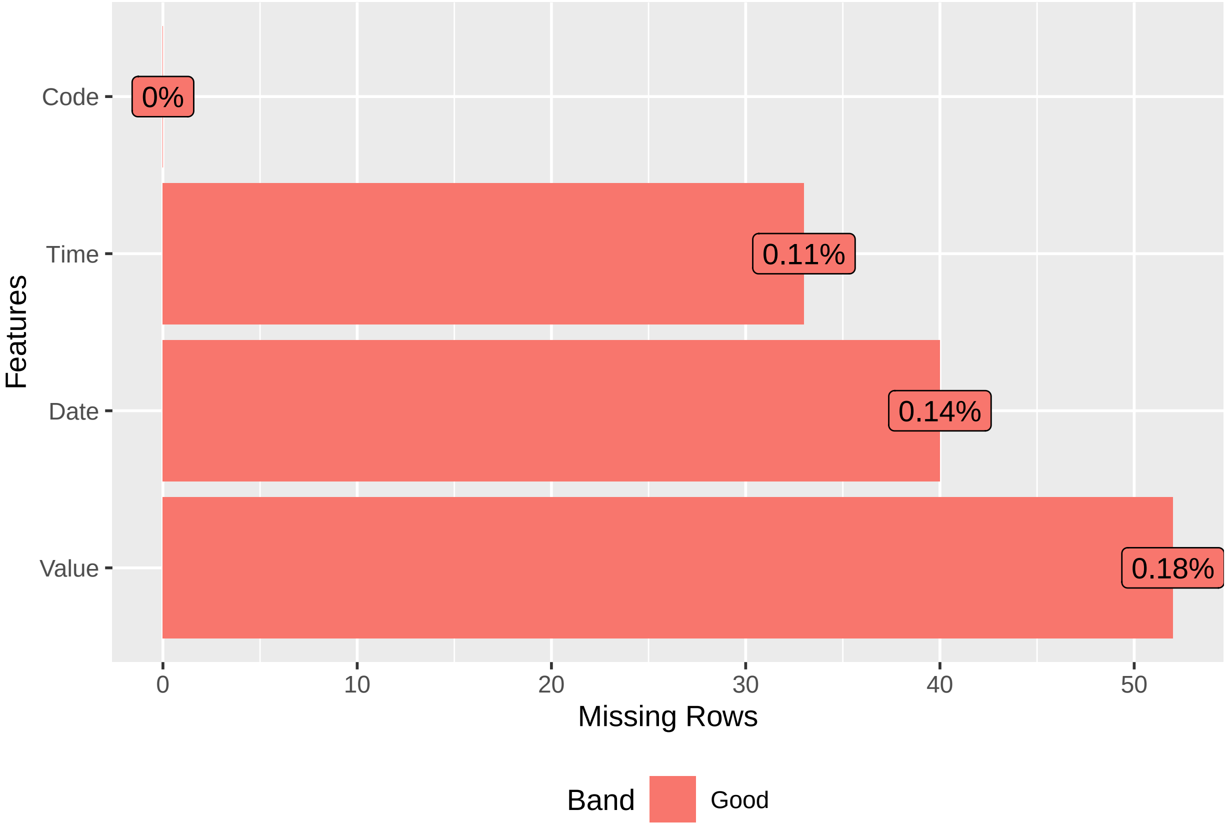
```
combined_data %>%
  group_by(Code) %>%
  count(name = "Total") %>%
  arrange(desc(Total)) %>%
  head() %>%
  kbl() %>%
  kable_classic(full_width = F)
```

Tool/language used: R Studio
Description of method: After grouping by the code variable, the count was taken of each instance of each specific Code and then placed in descending order under the Total column.

Task 7. In the above tasks, have you found any anomalies in the dataset? For instance, measurements without dates, with invalid test codes? How many rows have invalid data? Describe how you identified anomalous/missing data.
ANSWER: Missing data was found in the initial exploratory steps when reading in the data. There are a total of 125 missing observations with 40 belonging to the Date variable, 33 to Time, 0 to Code, and 52 for the Value variable. Additional missingness was introduced when converting the Time variable to a Date class using the lubridate package, but since this missingness only accounts for $< 1\%$ of the data (as seen in the plot below), it can be filtered out.

View Missing Data

```
plot_missing(combined_data)
```



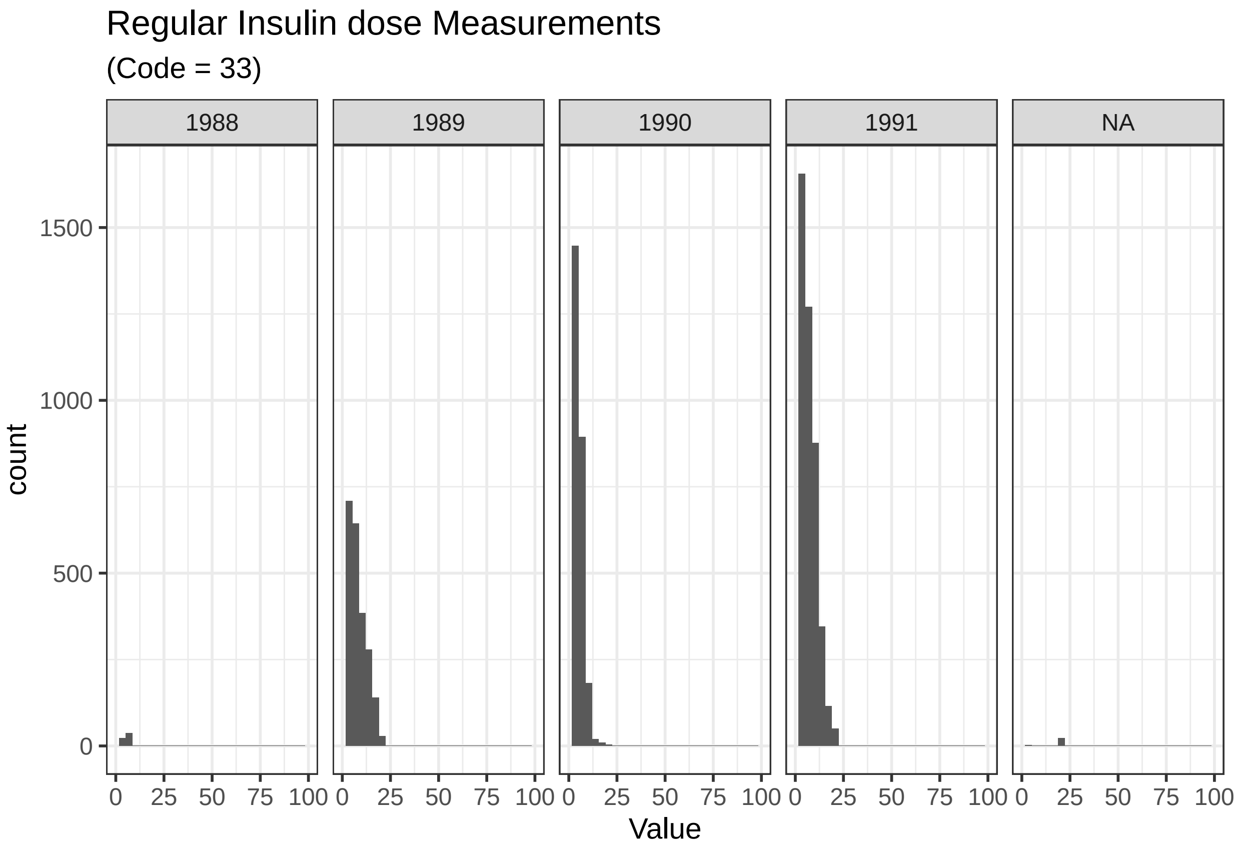
```
combined_data %>% summarise_all(~ sum(is.na(.)))
```

```
## # A tibble: 1 x 4
##    Date  Time  Code Value
##   <int> <int> <int> <int>
## 1    40    33     0    52
```

Task 8. Thinking about visualization, what do you think the impact of invalid or incompletely specified data on visualization might be? If you are not familiar with faceted searching, look up that term. What impact do you think invalid or poorly specified data will have on a faceted search?

ANSWER: Missingness will cause issue during visualization when trying to sort data into groups. For instance, if we want to plot Regular Insulin dose (Code = 33) measurements by year, the missing values in year will show up in the faceted plot as seen below:

```
combined_data %>%
  filter(Code == 33) %>%
  group_by(year(Date)) %>%
  ggplot() +
  geom_histogram(aes(Value)) +
  facet_grid(.~ year(Date)) +
  ggtitle("Regular Insulin dose Measurements", subtitle = "(Code = 33)") +
  xlim(0,100) + theme_bw()
```



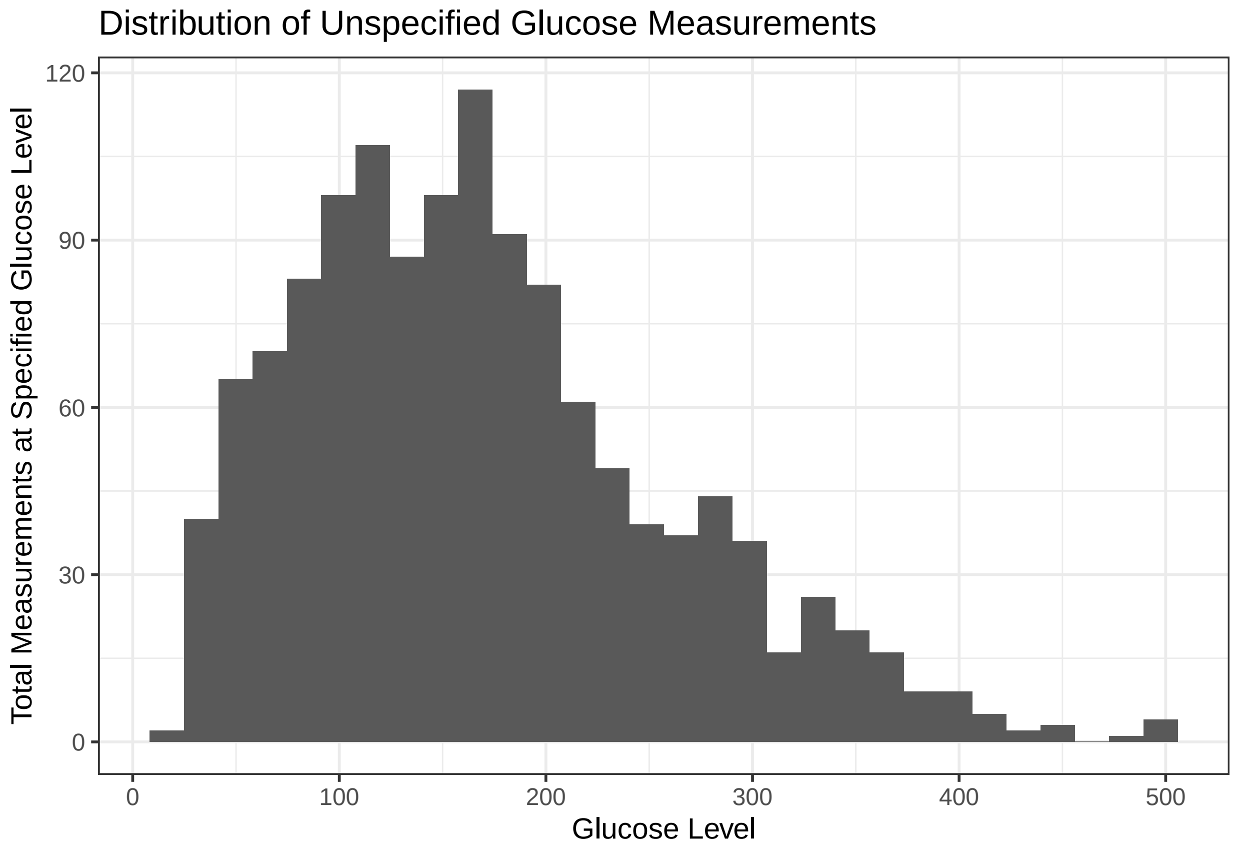
Data Visualization Tasks

Task 1. Graph the distribution of blood glucose levels using the data in the datafile.

Tool used: R Studio
Did you need to preprocess the data to visualize it? Yes, had to filter for code = 48, 55

```
combined_data %>%
  filter(Code == c(48,57)) %>%
  ggplot() +
  geom_histogram(aes(Value)) +
  theme_bw() +
  ggtitle("Distribution of Unspecified Glucose Measurements") +
  ylab("Total Measurements at Specified Glucose Level") +
  xlab("Glucose Level")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Task 2. Generate a table distribution of blood glucose levels by day of the week for the data in the datafile and then graph it

Tool used:R studio

```
# Create a dataframe including weekday
df_weekday <- combined_data %>%
  group_by(wday(Date, label = TRUE, abbr = FALSE)) %>%
  rename("Weekday" = `wday(Date, label = TRUE, abbr = FALSE)`) %>%
  summarise(Value, Code = Code)
```

```
Glucose_by_Weekday <- combined_data  %>%
  group_by(wday(Date, label = TRUE, abbr = FALSE)) %>%
  rename("Weekday" = `wday(Date, label = TRUE, abbr = FALSE)`) %>%
  summarise(Glucose_Level = Value) %>%
  ungroup()

df_weekday %>%
  filter(Code == c(48, 57)) %>%
  ggplot() +
  geom_histogram(aes(Value)) +
  facet_grid(. ~ Weekday) +
  theme_bw() +
  ggtitle("Distribution of Unspecified Glucose Measurements",
          subtitle = "By Weekdays") +
  xlab("Glucose Level") +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=0.5))
```

Weekday	Avg_Glucose_Level
Sunday	172.6216
Monday	165.1312
Tuesday	152.7402
Wednesday	171.7124
Thursday	171.8774
Friday	166.1900
Saturday	175.5981

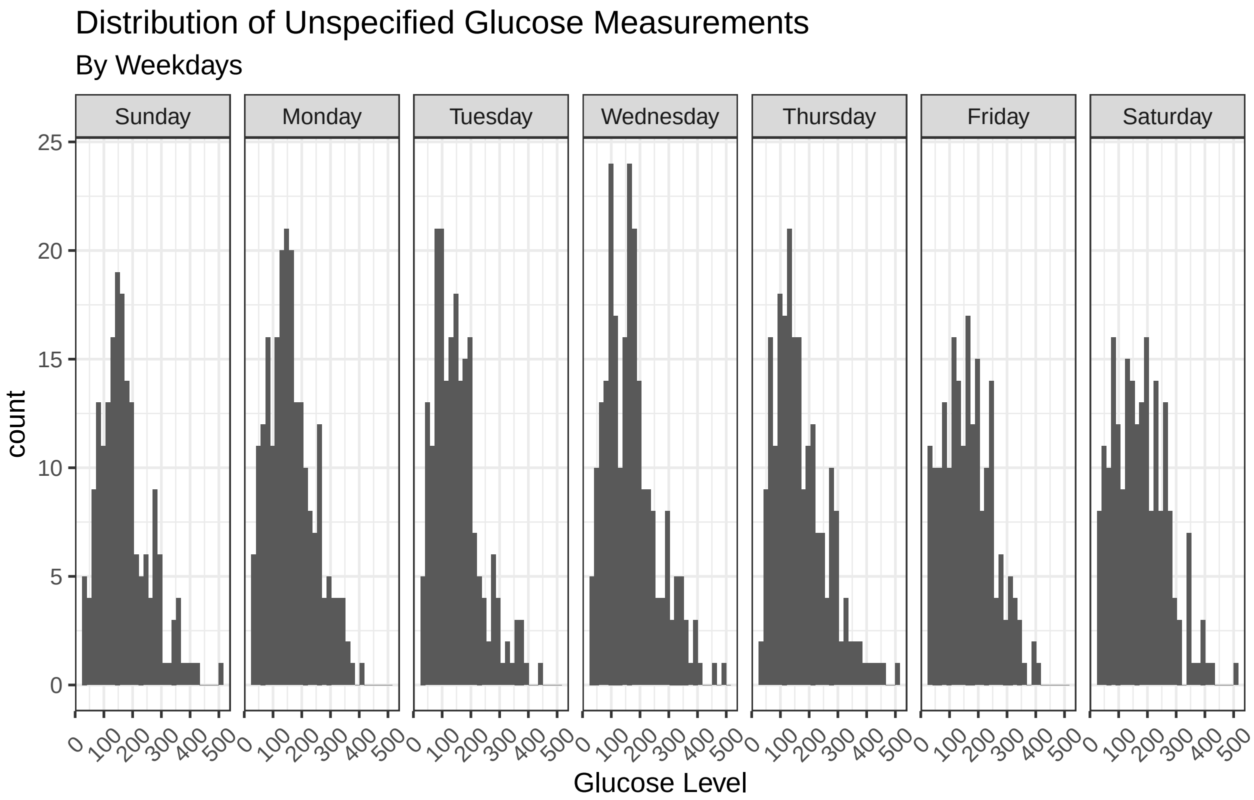


Table:

```r
df_weekday %>%
  filter(Code == c(48, 57)) %>%
  group_by(Weekday) %>%
  summarize(Avg_Glucose_Level = mean(Value, na.rm = T)) %>%
  kbl() %>%
  kable_classic(full_width = F)
```

```r
CI_table <- Glucose_by_Weekday  %>%
  group_by(Weekday) %>%
  summarize(Avg_Glucose_Level = mean(Glucose_Level, na.rm = T))
mean <- mean(CI_table$Avg_Glucose_Level)
sd <- sd(CI_table$Avg_Glucose_Level)
n <- length(CI_table$Avg_Glucose_Level)

CI <- round(c(mean - qt(0.95,n-1)*sd/sqrt(n), mean + qt(0.95,n-1)*sd/sqrt(n)),2)
CI
```

```
## [1] 63.51 84.62
```

Did you need to preprocess the data to visualize it?
ANSWER: Yes, had to filter for Code = 48 & 57 to consider unspecified glucose measurements only.

Do you see different distributions by day?
ANSWER: Distributions varied slightly by day of the week with an average mean Unspecified Glucose Measurement level of 74.067 and a 95% confidence interval around the average mean unspecified glucose measurement of 63.51, 84.62. There are slight variations in the distributions by day where Friday and Saturday appear to have a wider spread in measurement values more similar to each other, while Sunday - Thursday have smaller spreads. Saturday and Sunday have the highest mean unspecified glucose measurements.

Suggest a potential explanation (hypothesis) for the different distributions?
ANSWER: The differences in distributions may depend on when the measurements were taken during the day. There is data present in this dataset that accounts for this, so it may be better to instead look at the distribution of glucose measurements pre-meal and post-meal across days to assess for differences that consider the effect of meal intake on glucose levels. Individuals may consume different types or amounts of food on the weekends compared to weekdays due to a difference in routine.

Give a short experimental design that would let you test your hypothesis:
ANSWER: As stated above, we could instead assess differences depending on pre-meal or post-meal glucose measurements. We propose a design where pre-breakfast measurements are assessed across the days of the week, and the mean pre-breakfast glucose measurement is compared across days of the week.

Task 3. Graph the distribution of test codes and test code frequencies (count of a specific test code) by day of the week.

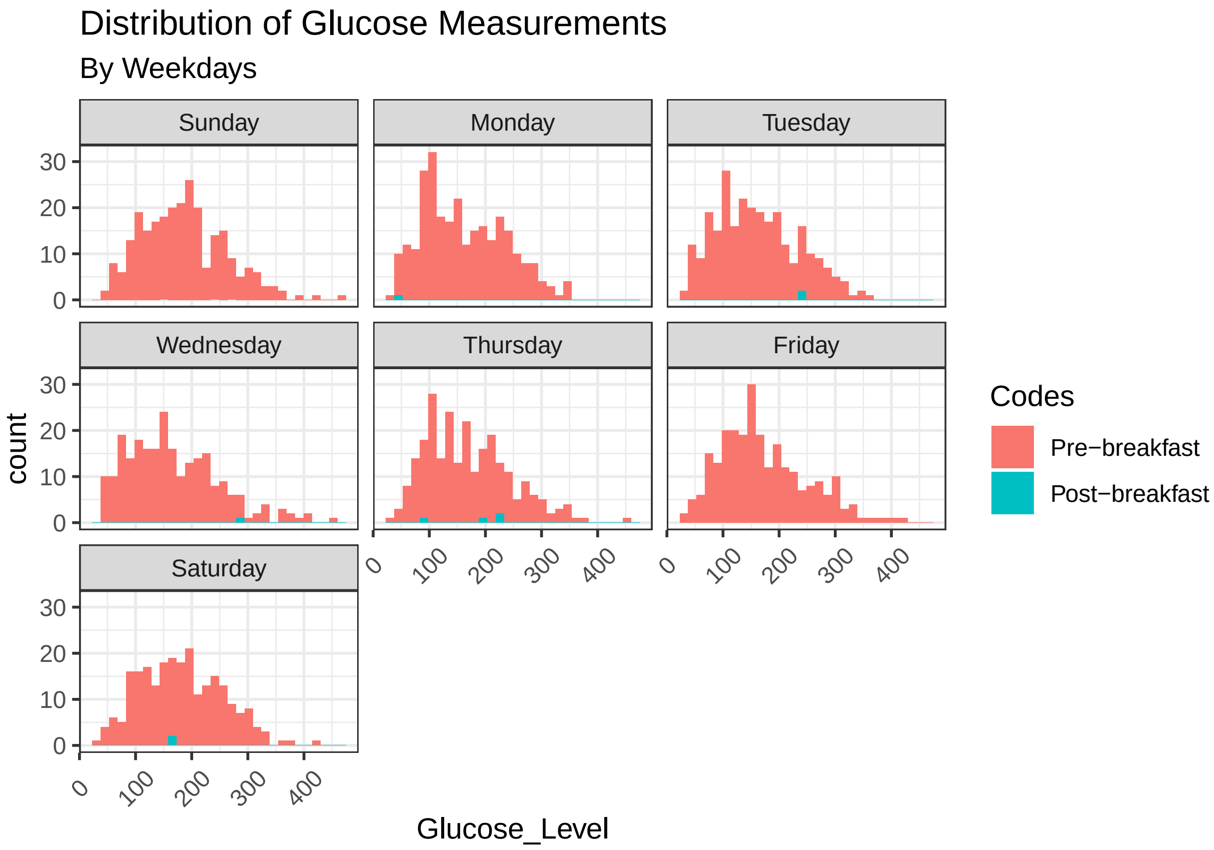
Graph:

```
Pre_Breakfast_Gluc_WD <- combined_data %>%
  filter(Code == c(58, 59)) %>%
  group_by(wday(Date, label = TRUE, abbr = FALSE)) %>%
  rename("Weekday" = `wday(Date, label = TRUE, abbr = FALSE)`) %>%
  summarise(Glucose_Level = Value, Code = Code) %>%
  ungroup()
```

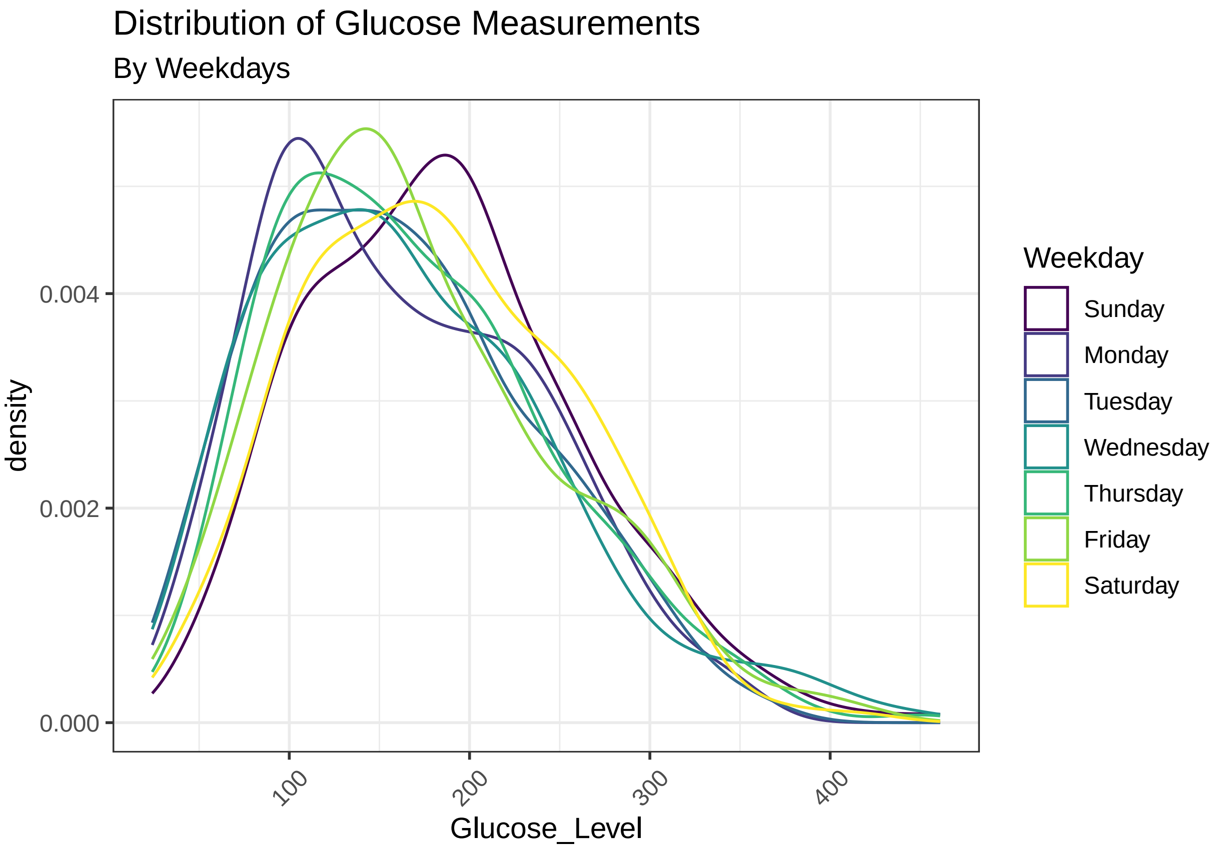
```
## `summarise()` has grouped output by 'Weekday'. You can override using the
## `.groups` argument.
```

```
Pre_Breakfast_Gluc_WD %>%
  drop_na() %>%
  ggplot() +
  geom_histogram(aes(Glucose_Level, fill = factor(Code))) +
  facet_wrap(~ Weekday) +
  theme_bw() +
  ggtitle("Distribution of Glucose Measurements",
          subtitle = "By Weekdays") +
  theme(axis.text.x = element_text(
    angle = 45,
    vjust = 0.5,
    hjust = 0.5
  )) + labs(fill = "Codes") +
  scale_fill_discrete(name = "Codes", labels = c("Pre-breakfast", "Post-breakfast"))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
Pre_Breakfast_Gluc_WD %>%
  filter(Code == 58) %>%
  drop_na() %>%
  ggplot() +
  geom_density(aes(Glucose_Level, color = Weekday)) +
  theme_bw() +
  ggtitle("Distribution of Glucose Measurements",
          subtitle = "By Weekdays") +
  theme(axis.text.x = element_text(
    angle = 45,
    vjust = 0.5,
    hjust = 0.5
  ))
```



Do you see different distributions by day?
ANSWER: The pre-breakfast glucose measurements distributions are similar across weekdays, with slight differences in the spread across days.

Can you come up with a potential explanation for the different distributions?
The differences in distributions may now be more difficult to explain without more context. There are subtle differences across days of the week where Tuesday-Friday appear to have distributions more similar to each other and Saturday has a wider spread of pre-breakfast glucose measurements. Glucose levels can be related to food, alcohol, exhaustion or stress for example, these factors may be more or less present at different points of the week that could affect measurement levels.

```
CI_table2 <- Pre_Breakfast_Gluc_WD  %>%
  group_by(Weekday) %>%
  summarize(Avg_Glucose_Level = mean(Glucose_Level, na.rm = T)) %>%
  drop_na()
mean2 <- mean(CI_table2$Avg_Glucose_Level)
sd2 <- sd(CI_table2$Avg_Glucose_Level)
n2 <- length(CI_table2$Avg_Glucose_Level)

CI2 <- round(c(mean2 - qt(0.95,n2-1)*sd2/sqrt(n2), mean2 + qt(0.95,n2-1)*sd2/sqrt(n2)),2)
CI2
```

```
## [1] 164.40 178.43
```

Task 4. Mirroring tasks 2 and 3, take the five most frequent test codes, and plot the distribution and frequency of blood glucose levels by day of the week for those codes.

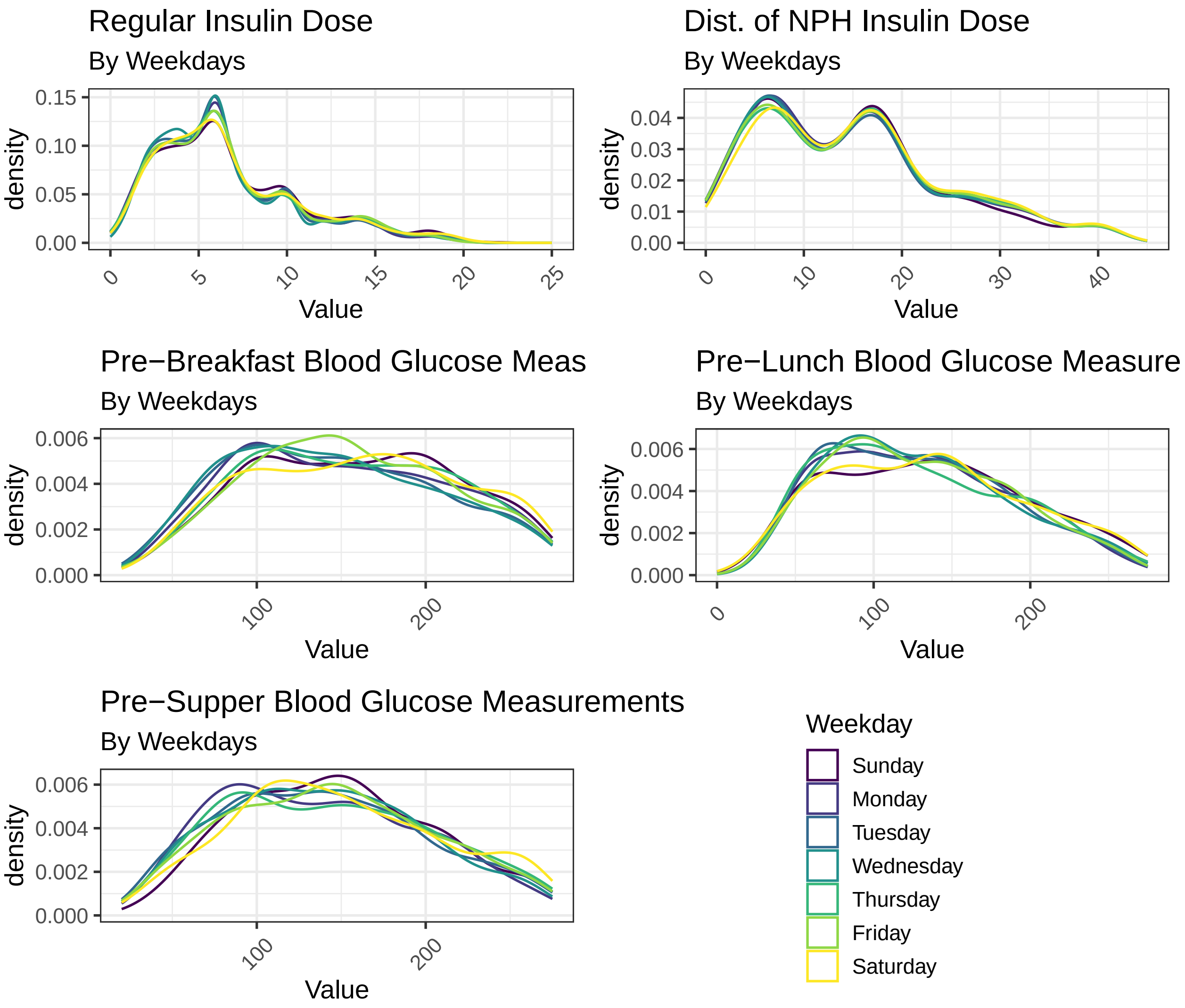
```
# Table of Code frequencies
glucose_measurement_codes <- c(48,57,58,59,60,61,62,63,64)
```

```
most_freq_codes <- combined_data %>%
  filter(Code %in% glucose_measurement_codes) %>%
  group_by(Code) %>%
  count(name = "Total") %>%
  arrange(desc(Total)) %>%
  head(5)
most_freq_codes
```

```
## # A tibble: 5 x 2
## # Groups:   Code [5]
##    Code Total
##   <dbl> <int>
## 1    58  3518
## 2    62  3160
## 3    60  2771
## 4    48  1883
## 5    57   990
```

```
# Create a function for more efficient code plotting
plot_codes <- function(Codes, xlim1 = 0,
                       xlim2 = 30,
                       title = NULL,
                       legend.position = "right"){
  df_weekday %>%
  filter(Code == Codes) %>%
  drop_na() %>%
  ggplot() +
  geom_density(aes(Value, color = Weekday)) +
  theme_bw() +
  ggtitle(paste(title),
          subtitle = "By Weekdays") +
  theme(axis.text.x = element_text(
    angle = 45,
    vjust = 0.5,
    hjust = 0.5
  )) + xlim(xlim1,xlim2) +
    theme(legend.position = legend.position, legend.key.size = unit(0.5, 'cm'))
}
```

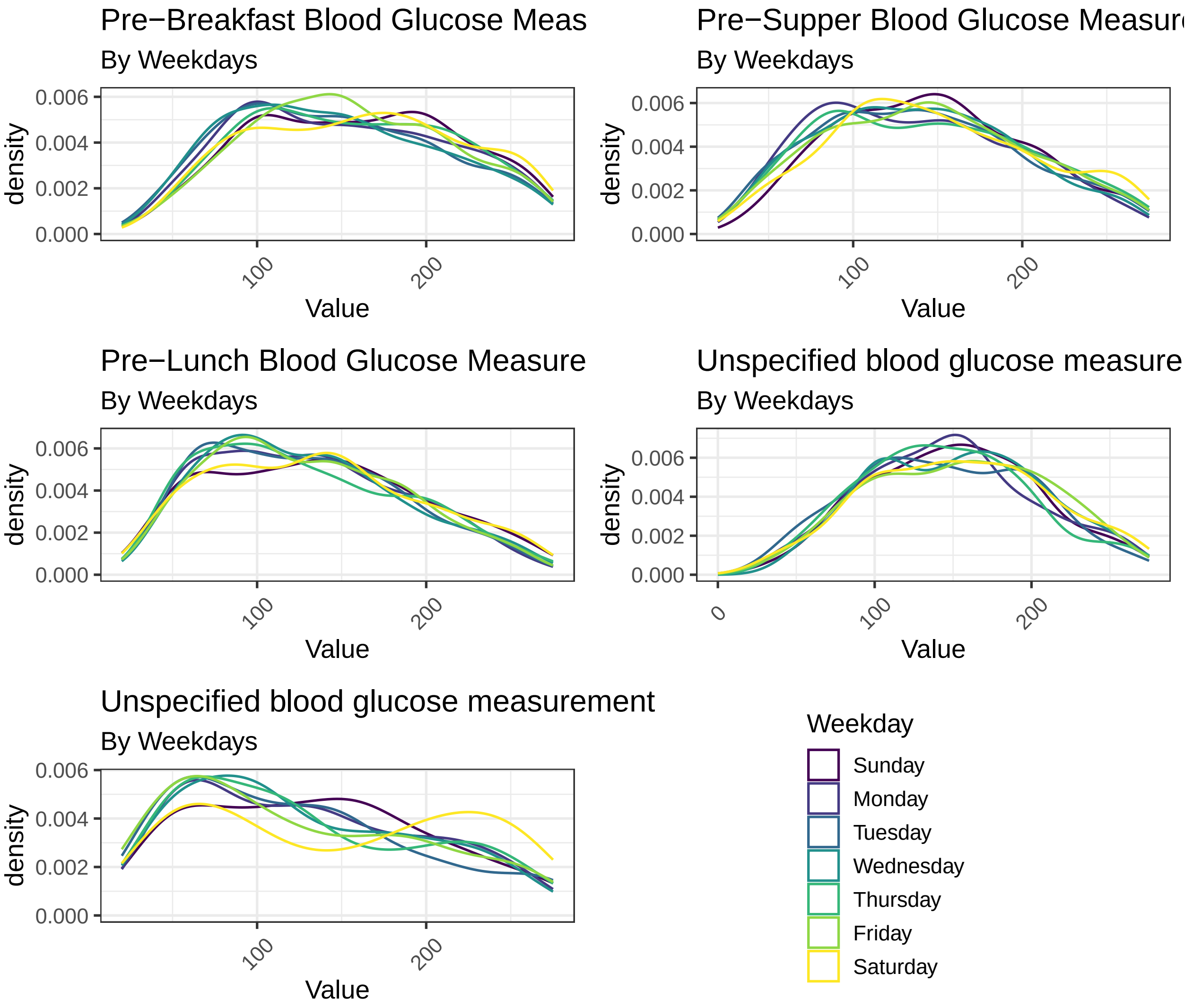
```
require(gridExtra)
grid.arrange(
plot_codes(33, 0, 25,"Regular Insulin Dose",
           legend.position = "none") ,
plot_codes(34, 0, 45,"Dist. of NPH Insulin Dose",
           legend.position = "none") ,
plot_codes(58, 20, 275,"Pre-Breakfast Blood Glucose Measurements",
           legend.position = "none"),
plot_codes(60, 0, 275, "Pre-Lunch Blood Glucose Measurements",
           legend.position = "none"),
plot_codes(62, 20, 275, "Pre-Supper Blood Glucose Measurements",
           legend.position = "none"),
cowplot::get_legend(plot_codes(62, 20, 275,
                               "Pre-Supper Blood Glucose Measurements")), ncol = 2)
```



Compare with the distribution of blood glucose levels from the other test codes. Are they different?

When comparing across the codes with the highest frequencies of measurements, we note that Codes 33 and 34 are included with the most frequent measurements, however, these codes measure Regular Insulin dose and NPH Insulin Dose respectively. Therefore, we cannot compare these distributions ot the Blood Glucose distributions since the range of values differs. Had the question stated: *Take the five most frequent test codes among Blood Glucose Measurements*, then we would have codes = 58, 62, 60, 48, 57, in that order. The plots regarding only blood glucose measurements appear below.

```
require(gridExtra)
grid.arrange(
plot_codes(58, 20, 275,
           "Pre-Breakfast Blood Glucose Measurements",
           legend.position = "none") ,
plot_codes(62, 20, 275,
           "Pre-Supper Blood Glucose Measurements",
           legend.position = "none") ,
plot_codes(60, 20, 275,
           "Pre-Lunch Blood Glucose Measurements",
           legend.position = "none"),
plot_codes(48, 0, 275,
           "Unspecified blood glucose measurement",
           legend.position = "none"),
plot_codes(57, 20, 275,
           "Unspecified blood glucose measurement",
           legend.position = "none"),
cowplot::get_legend(plot_codes(62, 20, 275,
                               "Pre-Supper Blood Glucose Measurements")), ncol = 2)
```



Does this change your hypothesis from either task 2 or task 3? Why or why not?
ANSWER: There still appears to be slight variation in the glucose measurements throughout the weekdays, but in the pre-meal glucose measurements, we observe far less variation across days of the week as compared to unspecified glucose measurements taken at any point during the day, possibly at inconsistent times across days. From this, we can infer that comparing pre-meal glucose measurements across days of the week, as suggested in our hypothesis, would be a more appropriate way to answer questions regarding any differences in glucose measurement distributions.

End of homework 2