

MEDI 504 - Lab 2

Student Version - Allison Jung

```
# loading required libraries
library(readr)
library(cowplot)
library(ggthemes)
library(tidyverse)
library(ggribes)
library(utf8)
```

The biomedical data, we are going to use is obtained from Kaggle website. You can find this dataset (**diabetes**) in a data folder. The dataset contains information about diabetes of cohort of sample subjects. This dataset arises from a research study of the National Institute of Diabetes and Digestive and Kidney Diseases (Smith et al. 1988). The purpose of the dataset is to predict whether or not a patient has diabetes. It is based on certain test measurements included in the dataset. Here, the patients are all females at least 21 years old of Indian heritage.

The datasets consists of several medical predictors/features and one target/response variable named as Outcome:

- **Pregnancies** - Number of times pregnant
- **Glucose** - Plasma glucose concentration a 2 hours in an oral glucose tolerance test
- **BloodPressure** - Diastolic blood pressure (mm Hg)
- **SkinThickness** - Triceps skin fold thickness (mm)
- **Insulin** - 2-Hour serum insulin (mu U/ml)
- **BMI** - Body mass index (weight in kg/(height in m)²)
- **DiabetesPedigreeFunction** - Diabetes pedigree function
- **Age** - Age (years)
- **Outcome** - Diabetic outcome is given as binary, where “0” refers to normal.

Task 1:

First import the dataset into the workspace:

```
# import the dataset into the workspace
diab <- read_csv("~/Desktop/ubc/medi 504/lab2_files/data/diabetes.csv")

## Rows: 768 Columns: 9
## -- Column specification -----
## Delimiter: ","
## dbl (9): Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, D...
##
## 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.
```

```
head(diab)
```

```
## # A tibble: 6 x 9
##   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
##   <dbl>      <dbl>         <dbl>         <dbl>    <dbl> <dbl>
## 1         6      148           72           35         0  33.6
## 2         1       85           66           29         0  26.6
## 3         8      183           64            0         0  23.3
## 4         1       89           66           23        94  28.1
## 5         0      137           40           35       168  43.1
## 6         5      116           74            0         0  25.6
## # i 3 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <dbl>
```

Please use the code chunk below to perform specified modifications:

1. You should generate the `age_decade` column, which categorizes the patients' ages into decades based on the `Age` column.
2. Convert `age_decade` and `Outcome` columns into factors
3. Recode the `Outcome` column as "Diabetes-Positive" (when `Outcome == 1`) and "Diabetes-Negative" (when `Outcome == 0`).

Please print the head of the dataset to reflect your modifications

```
# Add any modification that is done to a dataset here
diab_age_decade <- diab %>%
  mutate(age_decade = Age - (Age %% 10)) %>%
  mutate(Outcome = as.factor(Outcome), age_decade = as.factor(age_decade)) %>%
  mutate(Outcome = fct_recode(Outcome, "Diabetes-Positive" = "1", "Diabetes-Negative" = "0"))

# check labels for new factors (print head)
head(diab_age_decade)
```

```
## # A tibble: 6 x 10
##   Pregnancies Glucose BloodPressure SkinThickness Insulin   BMI
##   <dbl>      <dbl>         <dbl>         <dbl>    <dbl> <dbl>
## 1         6      148           72           35         0  33.6
## 2         1       85           66           29         0  26.6
## 3         8      183           64            0         0  23.3
## 4         1       89           66           23        94  28.1
## 5         0      137           40           35       168  43.1
## 6         5      116           74            0         0  25.6
## # i 4 more variables: DiabetesPedigreeFunction <dbl>, Age <dbl>, Outcome <fct>,
## #   age_decade <fct>
```

Reproduce the following figures using the `diab` dataset:

Note: the following colours are used in the plot: "#0072B2" "#E69F00"

1.1 Violin

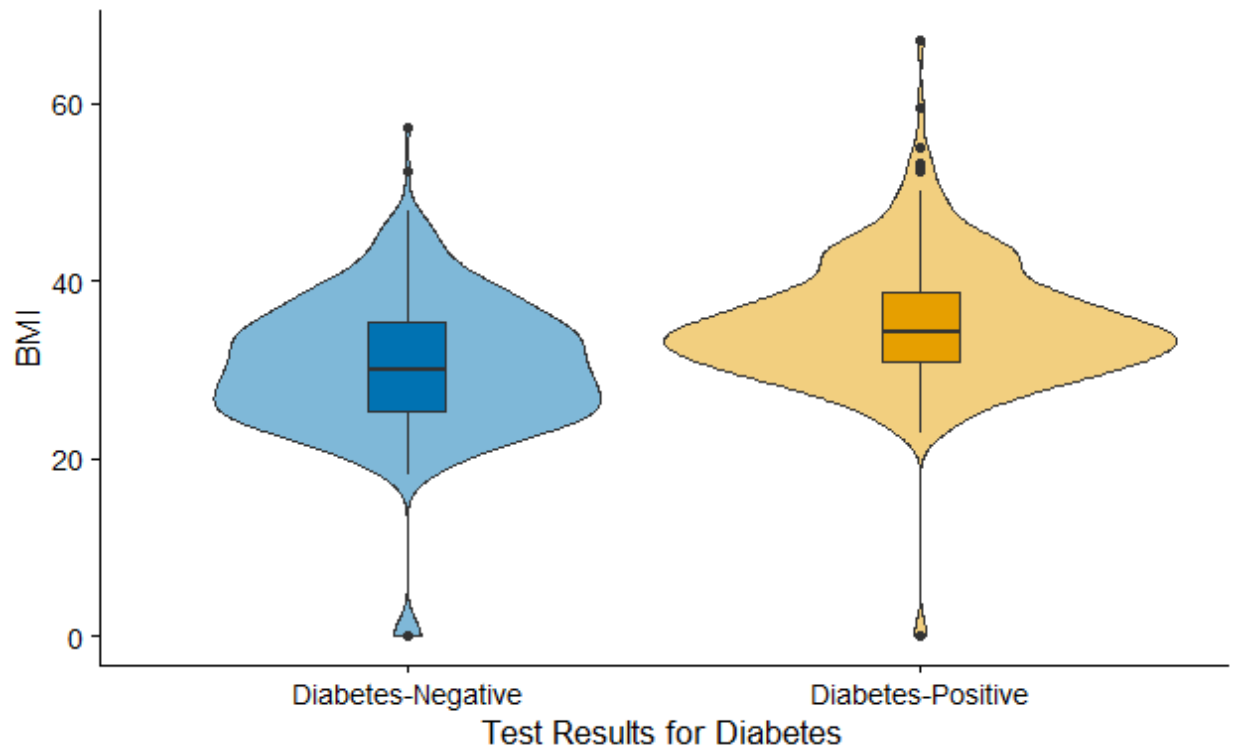
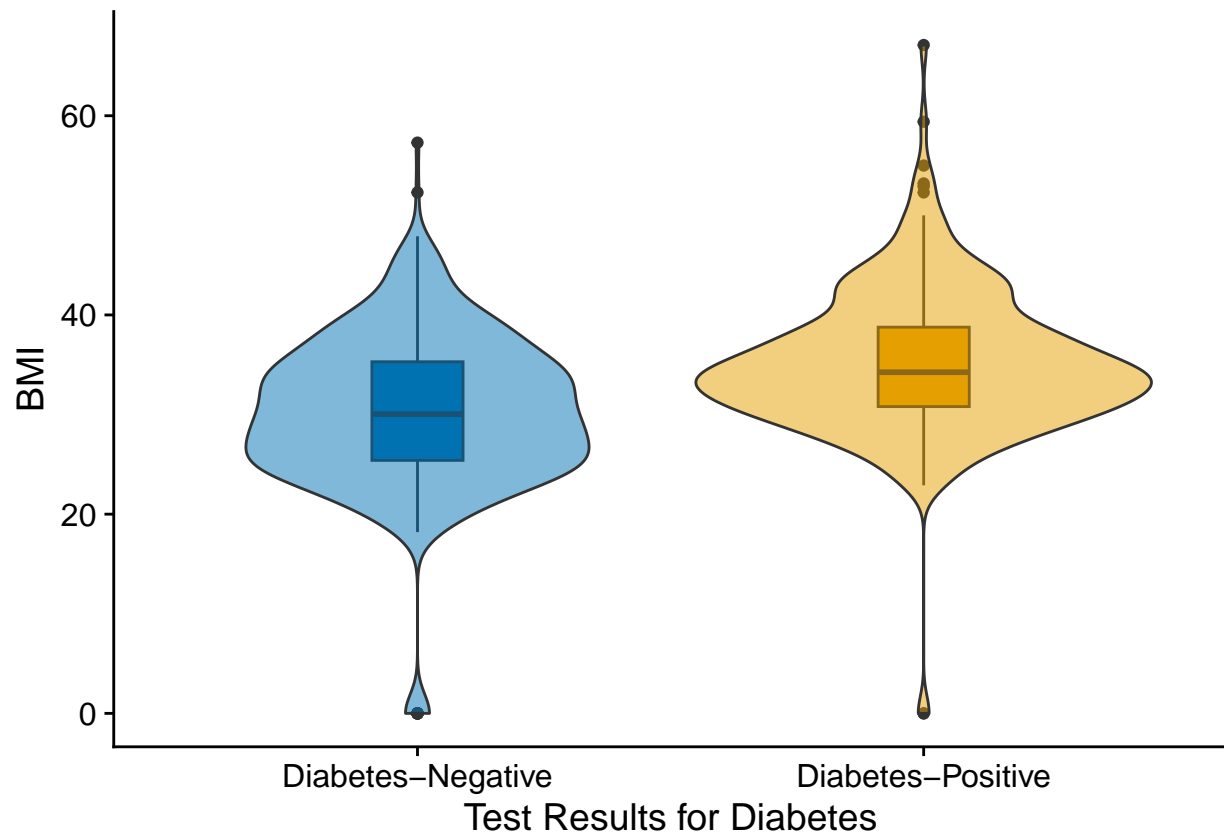


Figure 1: Exercise 1.1

```
diab_age_decade %>%
  ggplot(aes(x = Outcome, y = BMI, group = Outcome, fill = Outcome)) +
  geom_boxplot(width = 0.18) +
  geom_violin(alpha = 0.5) +
  theme_cowplot() +
  labs(x = "Test Results for Diabetes", y = "BMI") +
  scale_fill_manual(values=c("#0072B2", "#E69F00")) +
  theme(legend.position = "none")
```



1.2 Ridge Plot

```
diab_age_decade %>%  
  ggplot(aes(x = Pregnancies, y = age_decade)) +  
  geom_density_ridges(fill = "#0072B2", alpha = 0.4) +  
  theme_classic() +  
  labs(x = "Number of pregnancies", y = "Age (decades)") +  
  theme(legend.position = "none")
```

```
## Picking joint bandwidth of 0.87
```

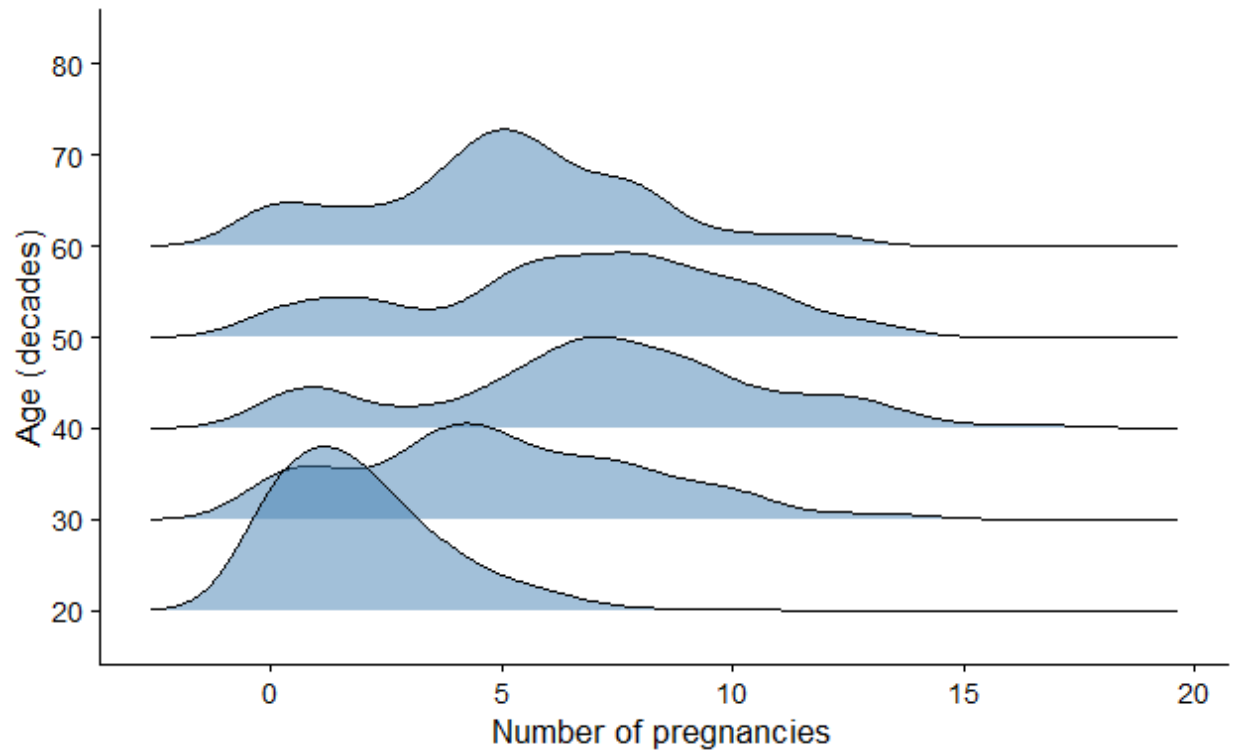
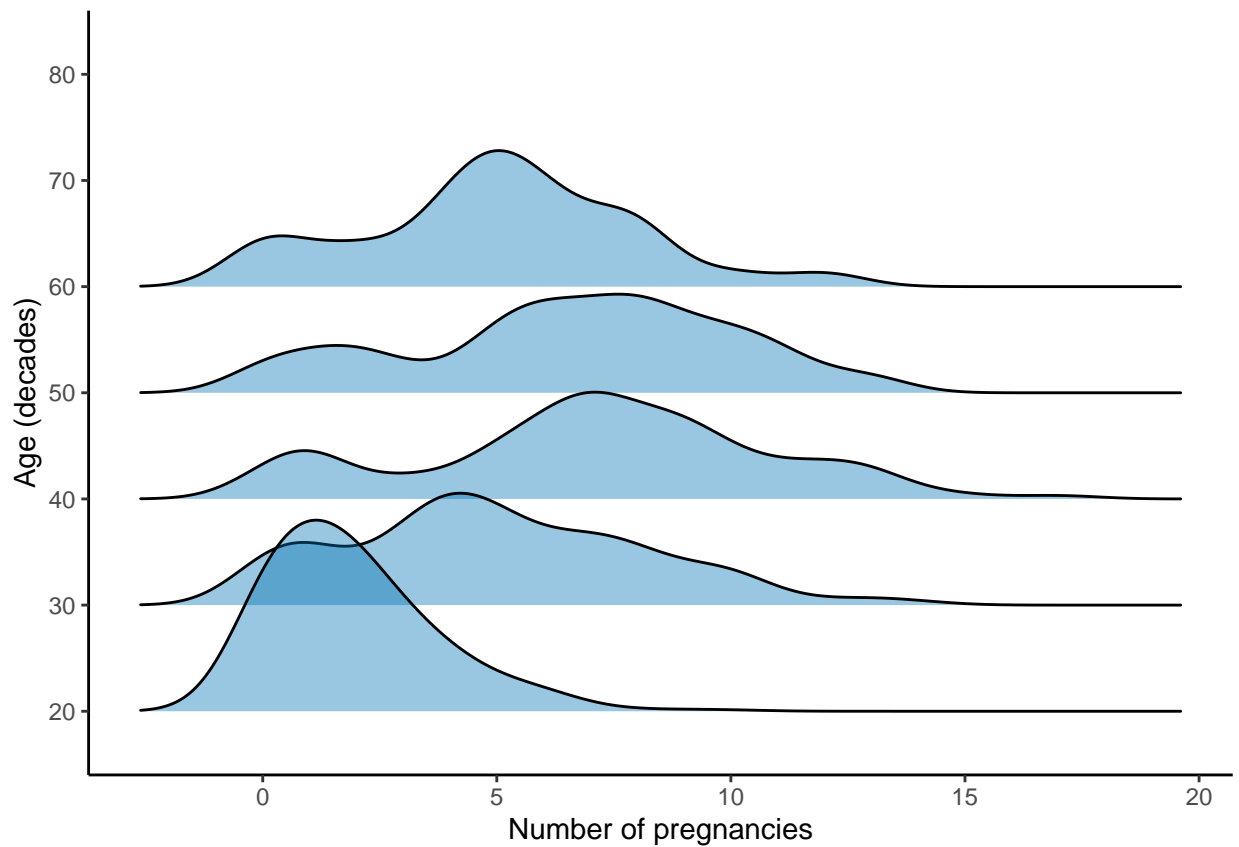


Figure 2: Exercise 1.2



1.3 Composite Plot: scatter plot and boxplot

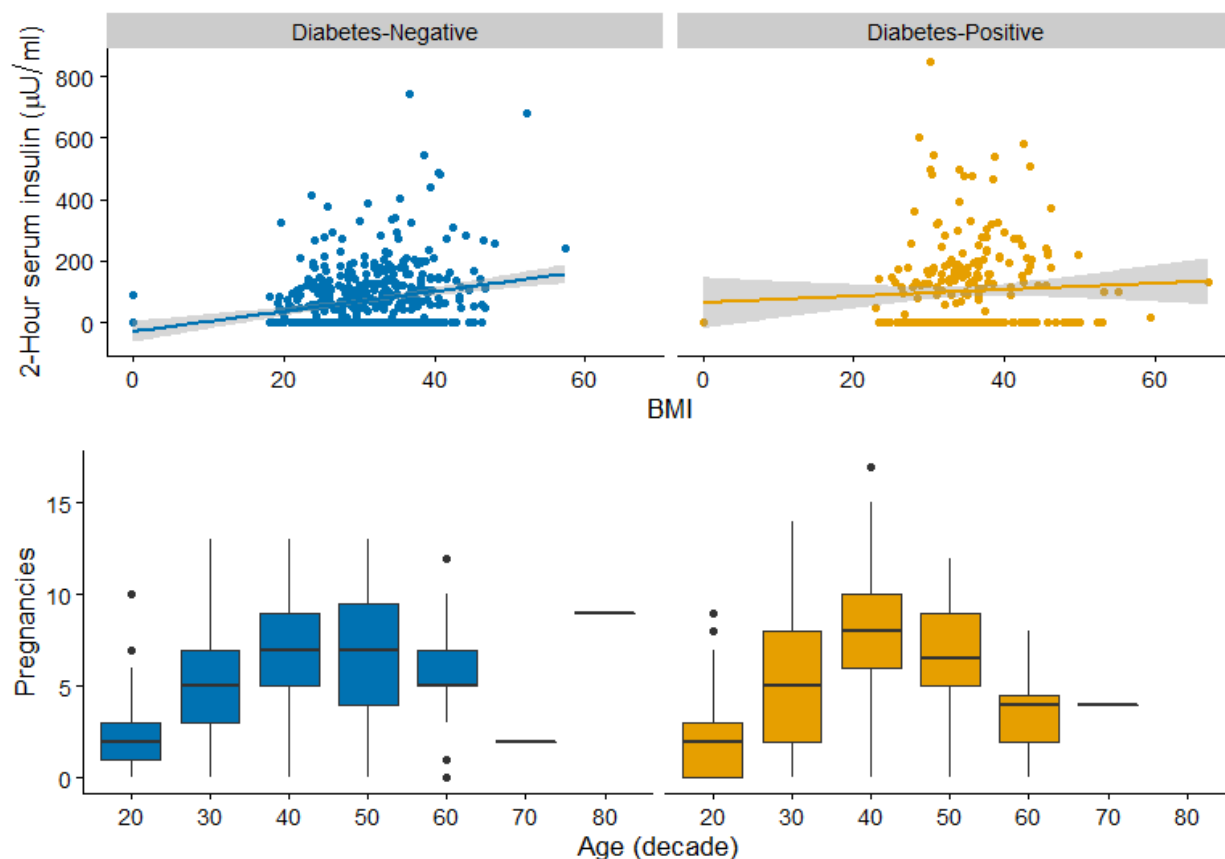


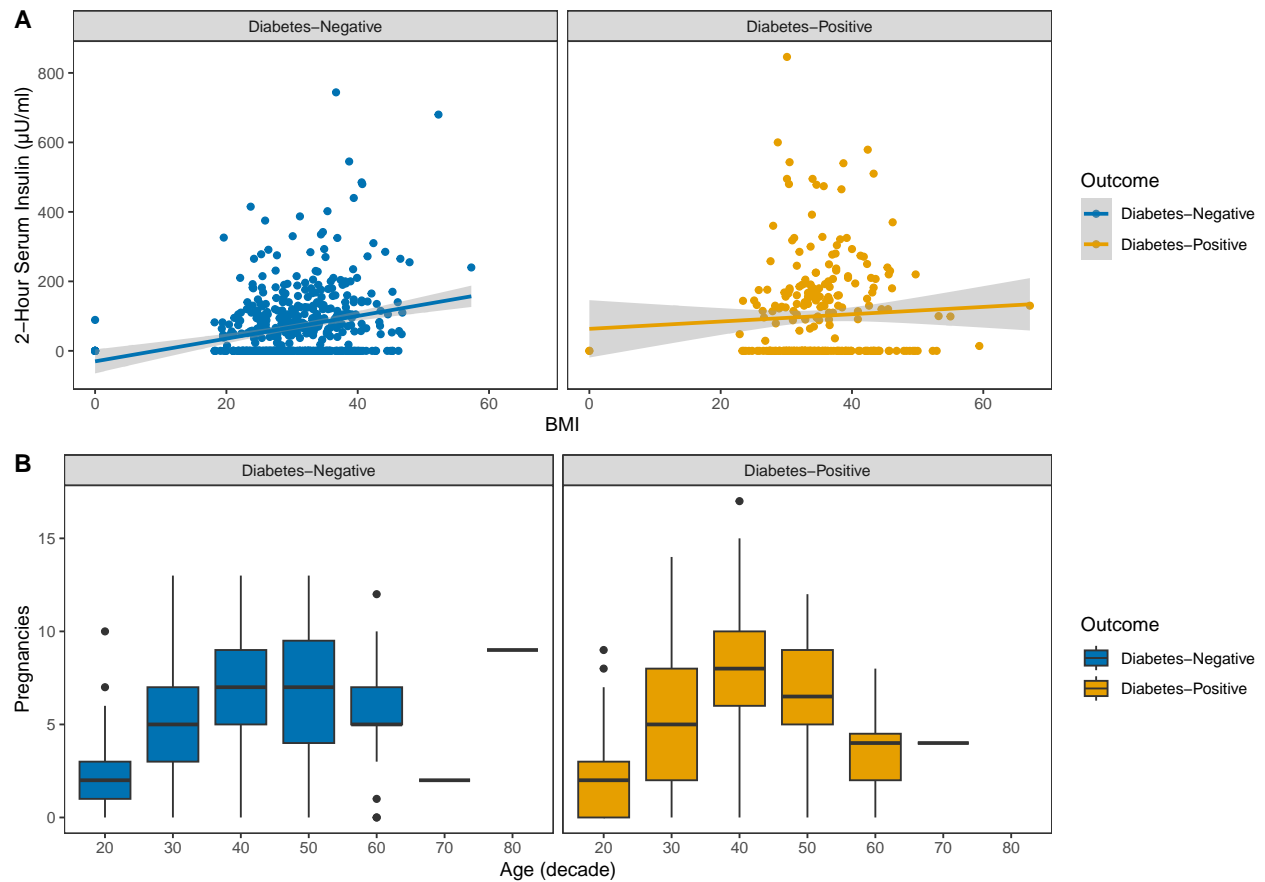
Figure 3: Exercise 1.3

```
# use plot_grid & facet_grid
# Scatter plot: BMI vs. 2-hour Serum Insulin
p1 <- diab_age_decade %>%
  ggplot(aes(x = BMI, y = Insulin, group = Outcome, color = Outcome)) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "lm") +
  theme_bw() +
  facet_grid(~ Outcome) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  labs(x = "BMI", y = "2-Hour Serum Insulin (µU/ml)") +
  scale_color_manual(values = c("#0072B2", "#E69F00"))

# Boxplot: Age vs. Pregnancies
p2 <- ggplot(diab_age_decade, aes(x = age_decade, y = Pregnancies, fill = Outcome)) +
  geom_boxplot() +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  facet_grid(~ Outcome) +
  labs(x = "Age (decade)", y = "Pregnancies") +
  scale_fill_manual(values = c("#0072B2", "#E69F00"))
```

```
# Combine the scatter plot and boxplot
composite_plot <- plot_grid(
  p1,
  p2,
  labels = c("A", "B"),
  ncol = 1
)

# Display the composite plot
print(composite_plot)
```



Task 2: Apply principles of effective visualizations

This exercise was adapted from emilyriederer `ugliest-ggplot-theme.R` to demonstrate the wide variety of `ggplot2` `theme()` options.

Just as a reminder, refer to the principles of effective visualizations when completing this task.

Let's look at this plot:

```
# Plot
ggplot(diab_age_decade %>% filter(Age<50),
  mapping = aes(x = BloodPressure, y = Pregnancies, col = age_decade)) +
  geom_point(size = 5) +
```

Principles of Effective Visualizations		
Principle	Definition	Examples
• Proportional Ink	The amount of ink used to indicate a value should be proportional to the value itself.	Truncating the y-axis on a bar chart to exaggerate the difference between bars violates the principle of proportional ink.
• Data:ink ratio	Remove distracting visual elements to focus attention on the data	Lighten line weights, remove backgrounds, never use 3D or special effects, remove avoid unnecessary/redundant labels.
• Labels & legends	Use axes labels and titles to highlight/communicate data	Never leave your data column names as axes labels! Generally good to add a title.
• Overplotting	With large datasets, points overlap, resulting in large clouds of data	To fix overplotting, could plot just a sample subset of the data, use alpha, and use smaller points. Or, jitter - but check if appropriate!
• Visualization choice	Must be informed by the data you have, the research question being asked and the audience that cares.	Pick the simplest plot that best shows most/all of the data needed to answer the research question. If you only have summary statistics, cannot show distributions. Tailor the visualization to your audience (within reason) but don't dumb it down.
• Colour & Accessibility	Colour can be used to encode information or for aesthetics/style/design. However, colour can also be distracting if used inappropriately or poorly.	Choose a perceptually uniform colour palette; can be sequential or diverging for quantitative data. Opt for colour-blind friendly palettes. Categorical data can use qualitative colour schemes.

Figure 4: Exercise 2

```

facet_grid(Outcome ~ age_decade, switch = "y") +
theme(
  plot.background = element_rect(fill = "lightyellow"),
  plot.title = element_text(size = 30, hjust = 0.25),
  plot.caption = element_text(size = 10, face = "italic", angle = 25),

  panel.background = element_rect(fill = 'lightblue', colour = 'darkred', size = 4),
  panel.border = element_rect(fill = NA, color = "green", size = 2),
  panel.grid.major.x = element_line(color = "purple", linetype = 2),
  panel.grid.minor.y = element_blank(),

  axis.title.x = element_text(face = "bold.italic", color = "blue"),
  axis.title.y = element_text(family = "mono", face = "bold", size = 20, hjust = 0.25),
  axis.text = element_text(face = "italic", size = 15),
  # note that axis.text options from above are inherited
  axis.text.x.bottom = element_text(angle = 180),

  # generally will want to match w plot background
  legend.background = element_rect(fill = "orangered4"),
  legend.key = element_rect(fill = "orange"),
  legend.direction = "horizontal",
  legend.position = "bottom") +

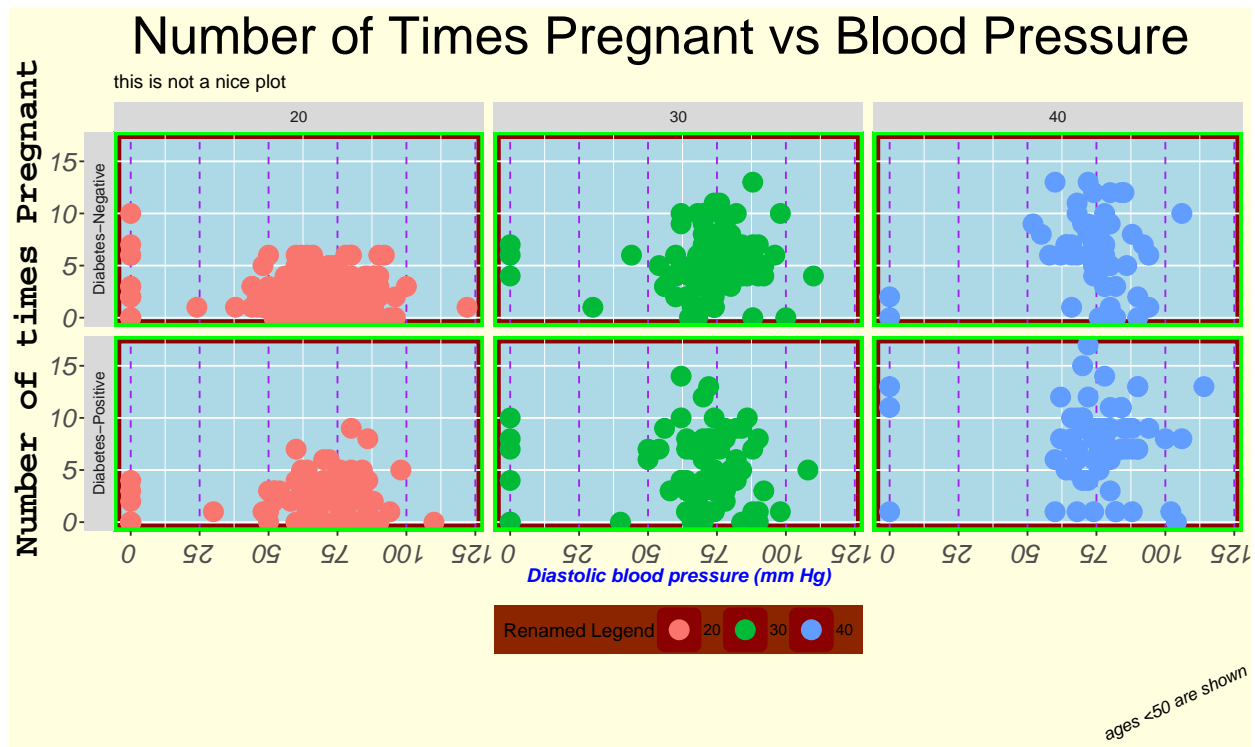
labs(title = "Number of Times Pregnant vs Blood Pressure",
     subtitle = "this is not a nice plot",
     x = "Diastolic blood pressure (mm Hg)",
     y = "Number of times Pregnant",
     caption = "ages <50 are shown",

```



```
col = "Renamed Legend")
```

```
## Warning: The 'size' argument of 'element_rect()' is deprecated as of ggplot2 3.4.0.
## i Please use the 'linewidth' argument instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



2.1 Summarize what is the role of arguments `plot.*`, `panel.*`, `axis.*`, and `legend.*`

`plot.*` arguments:

1. **`plot.background`**: Sets the overall background color of the plot area. Here, it is light yellow, which may distract from the actual data visualization.
2. **`plot.title`**: Adjusts the size and position of the plot title. A larger size and center alignment aim to make the title stand out.
3. **`plot.caption`**: Modifies the size and style of the caption text at the bottom of the plot. The angle adds visual interest but may reduce readability.

`panel.*` arguments:

1. **`panel.background`**: Sets the background color and border for the panel where the data is plotted. The light blue with a dark red border creates a contrast but might be too bold.
2. **`panel.border`**: Specifies the border's appearance around the plotting panel. The green border is prominent, possibly leading to distractions.
3. **`panel.grid.major.x`**: Customizes the major grid lines on the x-axis, making them purple and dashed. This emphasizes the x-axis but could clutter the visual if too pronounced.
4. **`panel.grid.minor.y`**: Hides the minor grid lines on the y-axis, potentially reducing visual noise.

axis.* arguments:

1. **axis.title.x**: Alters the font style and color of the x-axis title, making it bold and italicized in blue, which draws attention to it.
2. **axis.title.y**: Sets the y-axis title's font family, style, size, and alignment. Using a mono font and bold style may differentiate it from other text but could clash with the overall theme.
3. **axis.text**: Modifies the appearance of the axis text labels. Using italic and larger size improves legibility but might not match the plot's overall theme.
4. **axis.text.x.bottom**: Specifically adjusts the angle of the x-axis text, rotating it by 180 degrees, which is likely confusing for viewers.

legend.* arguments:

1. **legend.background**: Changes the legend's background color to a dark orange, which may clash with the light background of the plot.
2. **legend.key**: Sets the fill color for the legend keys (color swatches), which are orange in this case, furthering the color clash.
3. **legend.direction**: Specifies the orientation of the legend items as horizontal, which is good for compact space usage.
4. **legend.position**: Moves the legend to the bottom of the plot, which can make it easier to read if well-designed but could also interfere with the plot's data.

2.2 Based on your interpretation of this bad plot, what research question do you think the plot-creators are trying to answer with this plot?

Based on the provided plot, it appears the plot-creators are trying to answer the question: “**How does blood pressure relate to the number of pregnancies among patients with different diabetes outcomes and age categories?**” They likely aim to investigate any patterns or correlations between these two variables (blood pressure and pregnancies) while distinguishing between different diabetes statuses and age groups.

2.3 Using the principles of effective visualization, correct the figure. Comment on the steps you took to improve the plot and explain your choice.

```
# Modified Plot
# Improved Plot
ggplot(diab_age_decade %>% filter(Age < 50),
       mapping = aes(x = BloodPressure, y = Pregnancies, col = age_decade)) +
  geom_point(size = 3, alpha = 0.5) + # Adjust point size and add transparency
  facet_grid(Outcome ~ age_decade, switch = "y") +
  theme_minimal() + # Change to a minimal theme for cleaner presentation
  labs(title = "Pregnancies vs. Blood Pressure by Diabetes Outcome",
       subtitle = "Examining the relationship among patients under 50 years",
       x = "Diastolic Blood Pressure (mmHg)",
       y = "Number of Pregnancies",
       caption = "Data for ages < 50",
       col = "Age Decade") + # Clearer legend title
  theme(
    plot.title = element_text(size = 24, hjust = 0.5), # Center title with a more reasonable size
    plot.subtitle = element_text(size = 16, hjust = 0.5), # Center subtitle
```

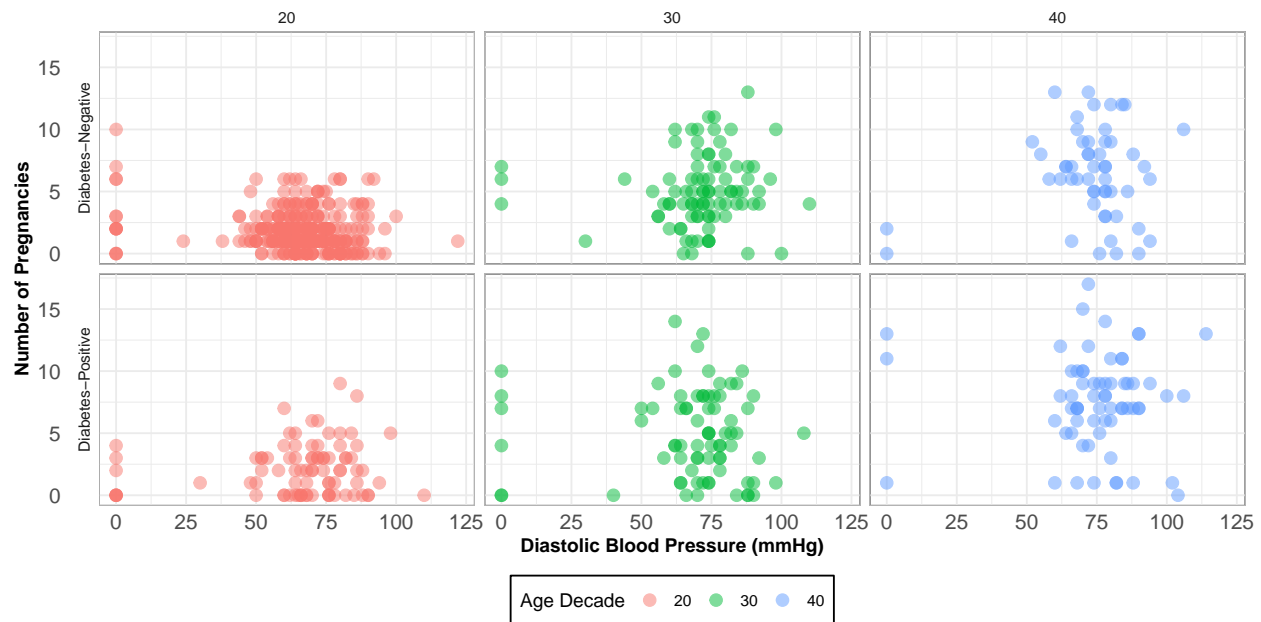
```

panel.background = element_rect(fill = 'white'), # Use a white panel for a cleaner look
panel.border = element_rect(fill = NA, color = "grey70"), # More subtle border
legend.position = "bottom", # Keep the legend position
legend.background = element_rect(fill = "transparent"), # Transparent legend background
axis.title.x = element_text(face = "bold", color = "black"),
axis.title.y = element_text(face = "bold", color = "black"),
axis.text = element_text(size = 12)
)

```

Pregnancies vs. Blood Pressure by Diabetes Outcome

Examining the relationship among patients under 50 years



Data for ages < 50

Task 3: Advanced Figure Design

Using iris dataset reproduce the following plot:

```

# Scatter plot: Sepal.Length vs. Sepal.Width
p3 <- iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, group = Species, color = Species)) +
  geom_point() +
  geom_smooth(formula = y ~ x, method = "lm") +
  theme_bw() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_line(color = "lightgray"),
        panel.grid.minor.y = element_blank()) +
  theme(legend.position = "none") +
  facet_grid(~ Species) +
  labs(x = "Sepal.Length", y = "Sepal.Width")

```

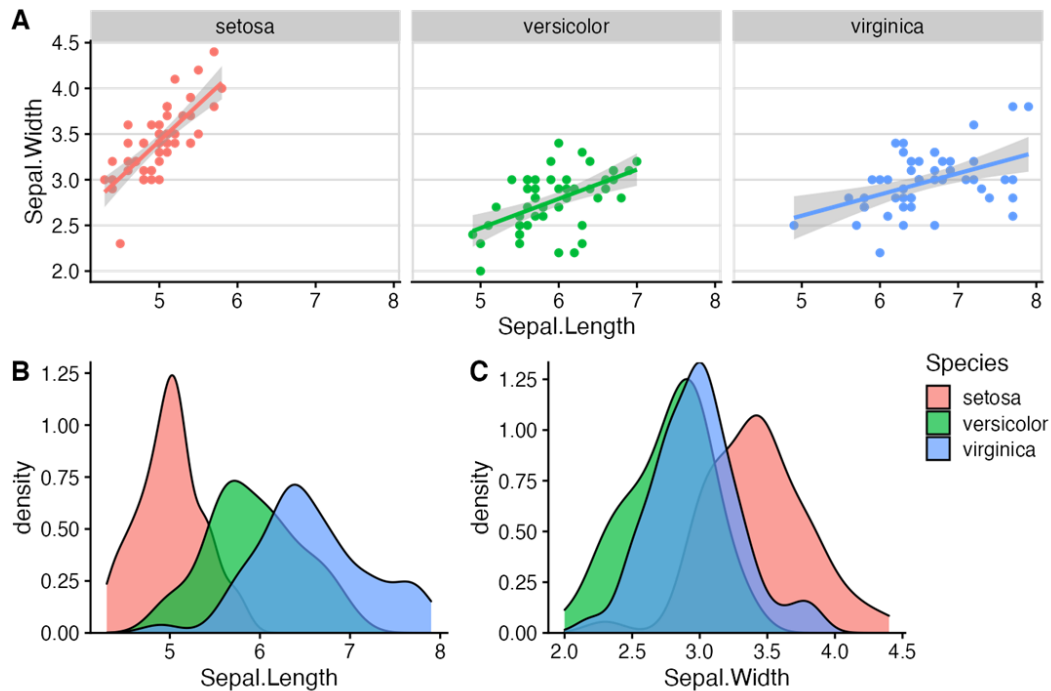


Figure 5: Plot title.

```
# Creating the density variable
data("iris")
iris$density <- iris$Sepal.Length * iris$Sepal.Width * iris$Petal.Length

# Adjusting the plot to show density on the y-axis
p4 <- iris %>%
  ggplot(aes(x = Sepal.Width, y = ..density.., fill = Species, color = Species)) +
  geom_density(color = "black", alpha = 0.4) +
  theme_classic() +
  labs(x = "Sepal.Width", y = "Density") +
  scale_fill_manual(values = c("red", "green", "blue"))

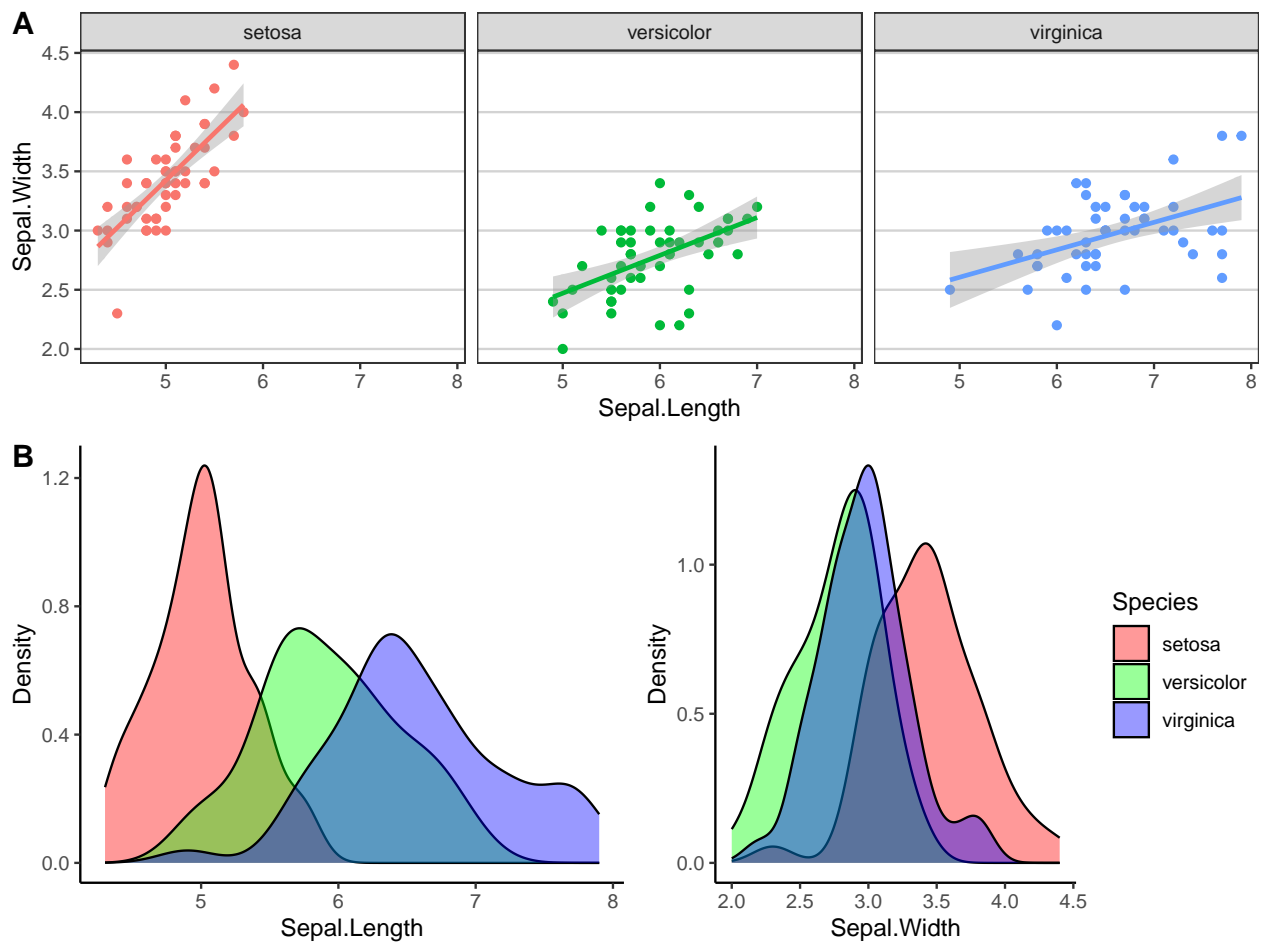
p5 <- iris %>%
  ggplot(aes(x = Sepal.Length, y = ..density.., fill = Species, color = Species)) +
  geom_density(color = "black", alpha = 0.4) +
  theme_classic() +
  labs(x = "Sepal.Length", y = "Density") +
  theme(legend.position = "none") +
  scale_fill_manual(values = c("red", "green", "blue"))

# Combine the scatter plot and density plots
composite_plot <- plot_grid(
  p3,
  plot_grid(p5, p4, ncol = 2),
  labels = c("A", "B", "C"),
  ncol = 1,
  label_size = 14,
  align = "v",
```

```
axis = "tb",
rel_heights = c(1, 1.2)
)
```

```
## Warning: The dot-dot notation ('..density..') was deprecated in ggplot2 3.4.0.
## i Please use 'after_stat(density)' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
# Display the composite plot
print(composite_plot)
```



Task 4: Making Ethical Data Decisions

Imagine that during a survey, the patients were asked to answer the question of their ethnicity:

The Canadian Census identifies the following categories in its Census of the Population (see options below). Please indicate how you self-identify (you can select more than one category). This self-identification is not intended as an indication of one's place of origin, citizenship, language or culture and recognizes that there are differences both between and among subgroups of persons of colour. If you are of mixed-descent,

please indicate this by selecting all that apply, rather than using the “other” line unless parts of your self-identification do not appear in this list.

- Indigenous person of Canada (First Nations, Inuit, Métis)
- Indigenous (outside of Canada)
- Arab
- Black
- Chinese (including Hong Kong and Macau)
- Filipino
- Japanese
- Korean
- Latin, Central, or South American (e.g. Brazilian, Chilean, Colombian, Mexican)
- South Asian (e.g. Indian, Pakistani, Sri Lankan, etc.)
- Southeast Asian (e.g. Cambodian, Indonesian, Laotian Vietnamese, etc)
- West Asian (e.g. Afghan, Iranian, Syrian, etc)
- White
- None of the above
- Prefer not to answer

Once you download the data, examine how “messy” the data input format.

```
ethnicity <- read_csv("~/Desktop/ubc/medi 504/lab2_files/data/ethnicity_data.csv")

## Rows: 485 Columns: 2
## -- Column specification -----
## Delimiter: ","
## chr (1): Ethnicity
## dbl (1): ID
##
## 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(ethnicity)
```

Imagine you were tasked to present the data for the conference/board meeting to show the ethnic diversity of your patients.

4.1 Wrangling practice

First, try to convert the dataset in a more tidy form (For instance, making ethnicity labels shorter: **Chinese** (including Hong Kong and Macau) transformed to **Chinese**). Hint: you can create a new column for each ethnicity, and add value 1 if the person selected that category , and 0 if it was not selected.

```
ethnicity$White <- ifelse(grepl("White", ethnicity$Ethnicity), 1, 0)

## Warning in grepl("White", ethnicity$Ethnicity): unable to translate 'Indigenous
## person of Canada (First Nations, Inuit, M<e9>tis),South Asian (e.g. Indian,
## Pakistani, Sri Lankan, etc.) ,West Asian (e.g. Afghan, Iranian, Syrian, etc)
## (optional) please specify:,'White' to a wide string

## Warning in grepl("White", ethnicity$Ethnicity): input string 14 is invalid
```

```
## Warning in grepl("White", ethnicity$Ethnicity): unable to translate 'Indigenous  
## person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide string
```

```
## Warning in grepl("White", ethnicity$Ethnicity): input string 96 is invalid
```

```
## Warning in grepl("White", ethnicity$Ethnicity): unable to translate 'Indigenous  
## person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide string
```

```
## Warning in grepl("White", ethnicity$Ethnicity): input string 135 is invalid
```

```
## Warning in grepl("White", ethnicity$Ethnicity): unable to translate 'Indigenous  
## person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide string
```

```
## Warning in grepl("White", ethnicity$Ethnicity): input string 211 is invalid
```

```
## Warning in grepl("White", ethnicity$Ethnicity): unable to translate 'Indigenous  
## person of Canada (First Nations, Inuit, M<e9>tis)' to a wide string
```

```
## Warning in grepl("White", ethnicity$Ethnicity): input string 236 is invalid
```

```
## Warning in grepl("White", ethnicity$Ethnicity): unable to translate 'Indigenous  
## person of Canada (First Nations, Inuit, M<e9>tis)' to a wide string
```

```
## Warning in grepl("White", ethnicity$Ethnicity): unable to translate 'Indigenous  
## person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide string
```

```
## Warning in grepl("White", ethnicity$Ethnicity): unable to translate 'Indigenous  
## person of Canada (First Nations, Inuit, M<e9>tis)' to a wide string
```

```
## Warning in grepl("White", ethnicity$Ethnicity): unable to translate 'Indigenous  
## person of Canada (First Nations, Inuit, M<e9>tis),Chinese (including Hong Kong  
## and Macau)' to a wide string
```

```
ethnicity$Chinese <- ifelse(grepl("Chinese", ethnicity$Ethnicity), 1, 0)
```

```
## Warning in grepl("Chinese", ethnicity$Ethnicity): unable to translate  
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),South Asian (e.g.  
## Indian, Pakistani, Sri Lankan, etc.) ,West Asian (e.g. Afghan, Iranian, Syrian,  
## etc) (optional) please specify:,White' to a wide string
```

```
## Warning in grepl("Chinese", ethnicity$Ethnicity): input string 14 is invalid
```

```
## Warning in grepl("Chinese", ethnicity$Ethnicity): unable to translate  
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide  
## string
```

```
## Warning in grepl("Chinese", ethnicity$Ethnicity): input string 96 is invalid
```

```

## Warning in grepl("Chinese", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide
## string

## Warning in grepl("Chinese", ethnicity$Ethnicity): input string 135 is invalid

## Warning in grepl("Chinese", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide
## string

## Warning in grepl("Chinese", ethnicity$Ethnicity): input string 211 is invalid

## Warning in grepl("Chinese", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis)' to a wide string

## Warning in grepl("Chinese", ethnicity$Ethnicity): input string 236 is invalid

## Warning in grepl("Chinese", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis)' to a wide string

## Warning in grepl("Chinese", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide
## string

## Warning in grepl("Chinese", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis)' to a wide string

## Warning in grepl("Chinese", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),Chinese
## (including Hong Kong and Macau)' to a wide string

ethnicity$Southeast <- ifelse(grepl("Southeast", ethnicity$Ethnicity), 1, 0)

## Warning in grepl("Southeast", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),South Asian (e.g.
## Indian, Pakistani, Sri Lankan, etc.) ,West Asian (e.g. Afghan, Iranian, Syrian,
## etc) (optional) please specify:,White' to a wide string

## Warning in grepl("Southeast", ethnicity$Ethnicity): input string 14 is invalid

## Warning in grepl("Southeast", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide
## string

## Warning in grepl("Southeast", ethnicity$Ethnicity): input string 96 is invalid

## Warning in grepl("Southeast", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide
## string

```



```
## Warning in grepl("Southeast", ethnicity$Ethnicity): input string 135 is invalid

## Warning in grepl("Southeast", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide
## string

## Warning in grepl("Southeast", ethnicity$Ethnicity): input string 211 is invalid

## Warning in grepl("Southeast", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis)' to a wide string

## Warning in grepl("Southeast", ethnicity$Ethnicity): input string 236 is invalid

## Warning in grepl("Southeast", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis)' to a wide string

## Warning in grepl("Southeast", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),White' to a wide
## string

## Warning in grepl("Southeast", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis)' to a wide string

## Warning in grepl("Southeast", ethnicity$Ethnicity): unable to translate
## 'Indigenous person of Canada (First Nations, Inuit, M<e9>tis),Chinese
## (including Hong Kong and Macau)' to a wide string
```

```
view(ethnicity)
```

4.2 Visualization Practice

Second, Try to create a visual to effectively show the diversity of patients' ethnicity. It can be either table or a figure. This task is not easy, so it is okay that your visual wont be perfect. Please discuss what aspects of your figure still needs improvement and describe or describe how you see the ideal images in this case.

Submit Document

- knit the document to pdf file. You need to have a TinyTeX to be able to knit the document:

```
install.packages('tinytex')
tinytex::install_tinytex()
```

- make sure the document is tidy, code chunks has sufficient comments, your writing is clear and has no typos

References

Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In Proceedings of the Symposium on Computer Applications and Medical Care (pp. 261–265). IEEE Computer Society Press.