MEDI 504 - Lab 2

Instructor's Version

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

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 Pima Indian heritage.

The dataset 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 norm.

Task 1:

First import the dataset into the workspace:

```
# import the dataset into the workspace
diab <- read_csv("data/diabetes.csv") # It reads the CSV file and assigns to diab object
```

```
## 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 DiabetesPedigre~
##         <dbl>   <dbl>         <dbl>         <dbl>   <dbl> <dbl>            <dbl>
## 1           6     148            72            35       0  33.6            0.627
## 2           1      85            66            29       0  26.6            0.351
## 3           8     183            64             0       0  23.3            0.672
## 4           1      89            66            23      94  28.1            0.167
## 5           0     137            40            35     168  43.1            2.29
## 6           5     116            74             0       0  25.6            0.201
## # ... with 2 more variables: Age <dbl>, Outcome <dbl>
```

Please use the code chunk below to perform any modifications to original dataset.

```
# Add any modification that is done to a dataset here
spec(diab)
```

```
## cols(
##   Pregnancies = col_double(),
##   Glucose = col_double(),
##   BloodPressure = col_double(),
##   SkinThickness = col_double(),
##   Insulin = col_double(),
##   BMI = col_double(),
##   DiabetesPedigreeFunction = col_double(),
##   Age = col_double(),
##   Outcome = col_double()
## )
```

```
#for exercise 1, rename the outcome with proper labels,
#which will need to be combined with the exercise 2 age
#split into decades code to make the figure easier to make for exercise 3

diab_outcome <- diab %>%
  mutate( Outcome = as.factor(Outcome),
          Outcome = fct_recode(Outcome, "Diabetes-Positive" = "1",
                               "Diabetes-Negative" = "0"),
          age_decade = case_when(Age <30 ~ '20',
                                 Age <40 ~ '30',
                                 Age <50 ~ '40',
                                 Age <60 ~ '50',
                                 Age <70 ~ '60',
                                 Age <80 ~ '70',
                                 Age <90 ~ '80'),
          age_decade = as.factor(age_decade))

diab_outcome
```

```
## # A tibble: 768 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
##  7           3      78            50            32      88  31
##  8          10     115             0             0       0  35.3
##  9           2     197            70            45     543  30.5
## 10           8     125            96             0       0   0
## # ... with 758 more rows, and 4 more variables: DiabetesPedigreeFunction <dbl>,
## #   Age <dbl>, Outcome <fct>, age_decade <fct>
```

Reproduce the following figures using the `diab` dataset:

1.1 Violin

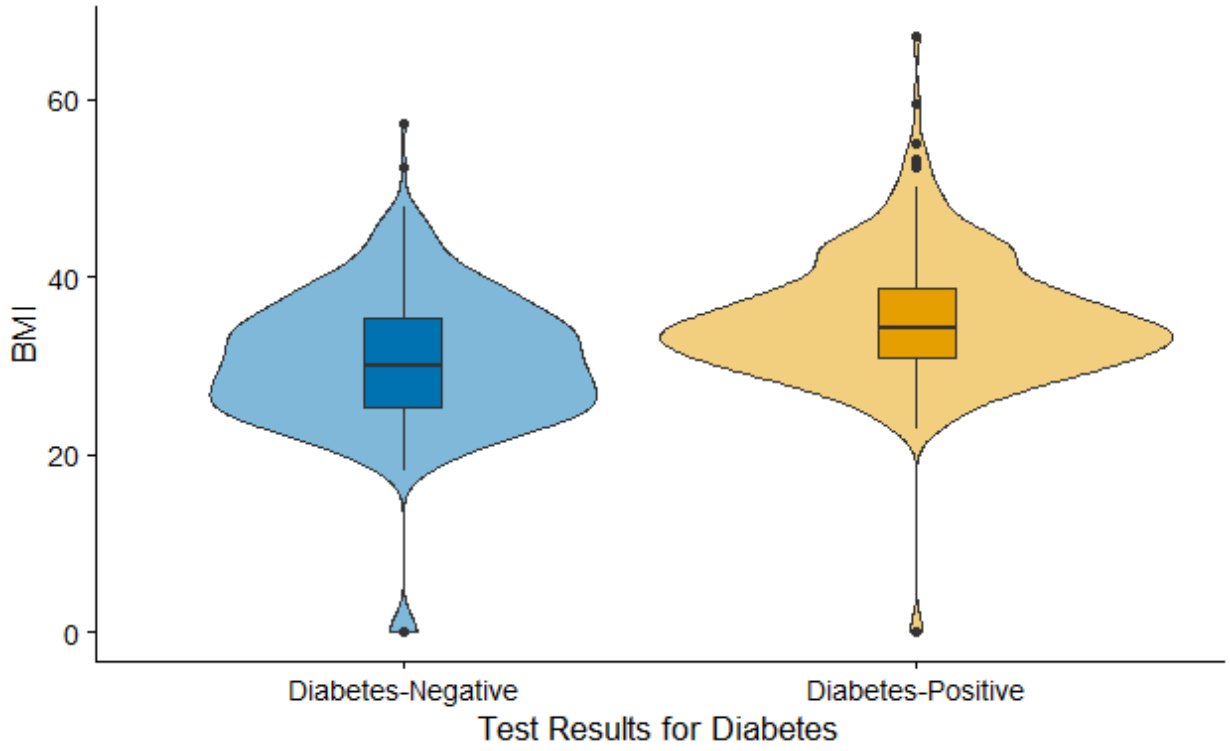
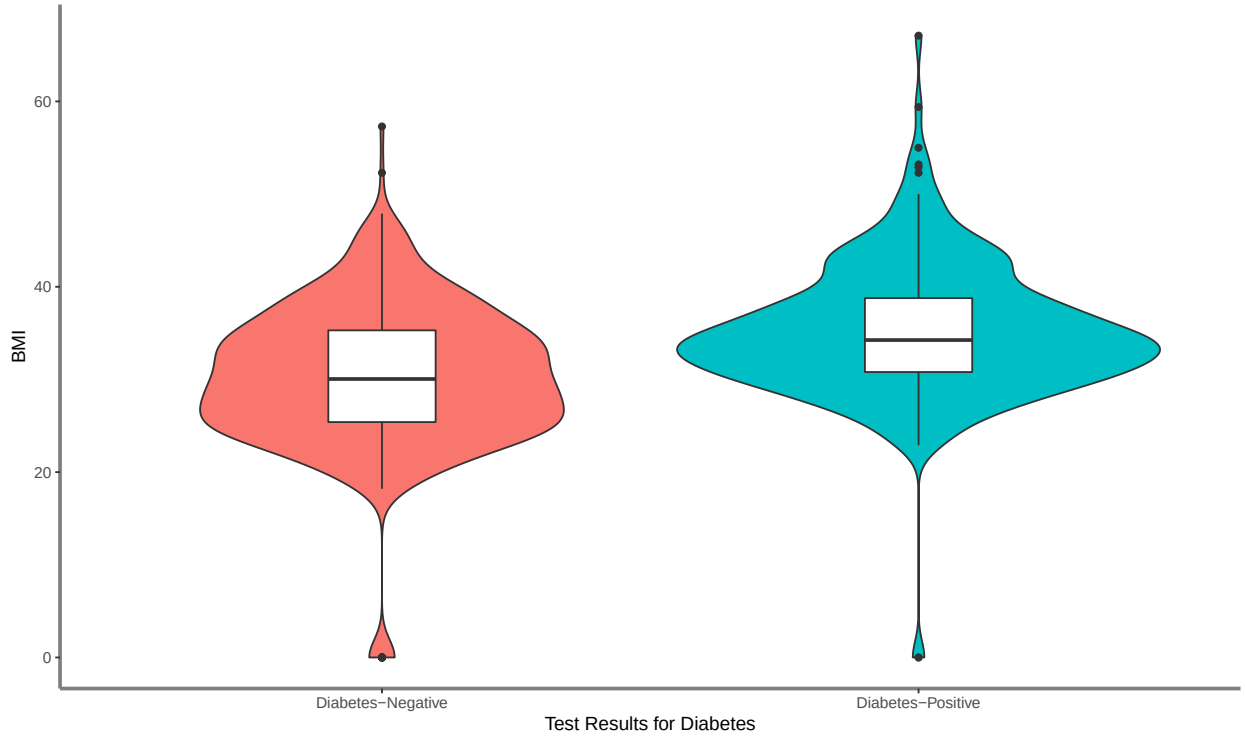


Figure 1: Exercise 1.1

```
# Plot

diab_violin <- diab_outcome %>%
  ggplot(aes(Outcome, BMI)) +
  geom_violin(aes(fill = Outcome)) +
  geom_boxplot(width = 0.2) +
  #label, BMi doesnt need label, explanatory already
  xlab("Test Results for Diabetes")  +
```

```
  #visuals
  theme(panel.background = element_rect(fill = "white"),
        legend.position = 'none',
        axis.line = element_line(size = 1, colour = "gray50", linetype=1)
        )

diab_violin
```



1.2 Ridge Plot

```
# Plot

diab_ridge <- diab_outcome %>%
  ggplot(aes(x = Pregnancies, y = age_decade)) +
  geom_density_ridges(alpha = 0.6, fill = 'cadetblue') +
  #titles
  xlab("Number of Pregnancies") +
  ylab("Age (decades)") +
  #visuals
  theme(panel.background = element_rect(fill = "white"),
        axis.line = element_line(size = 1, colour = "gray50", linetype=1))

diab_ridge
```

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

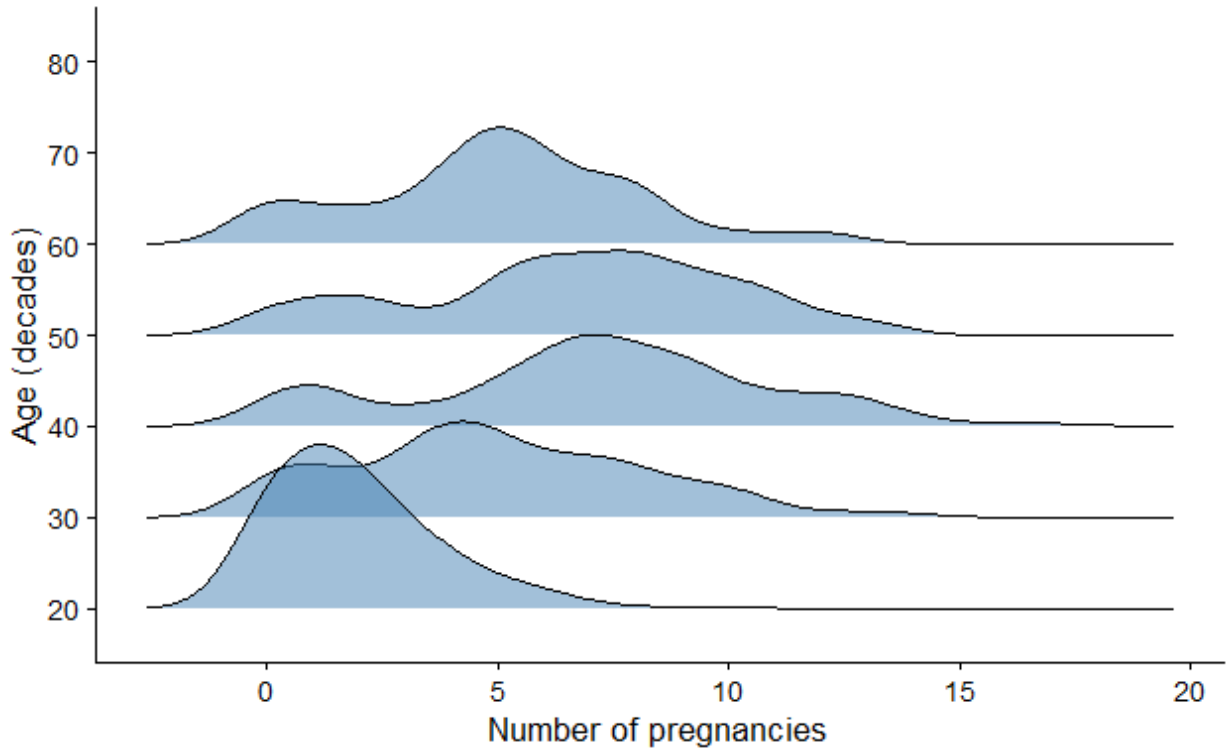
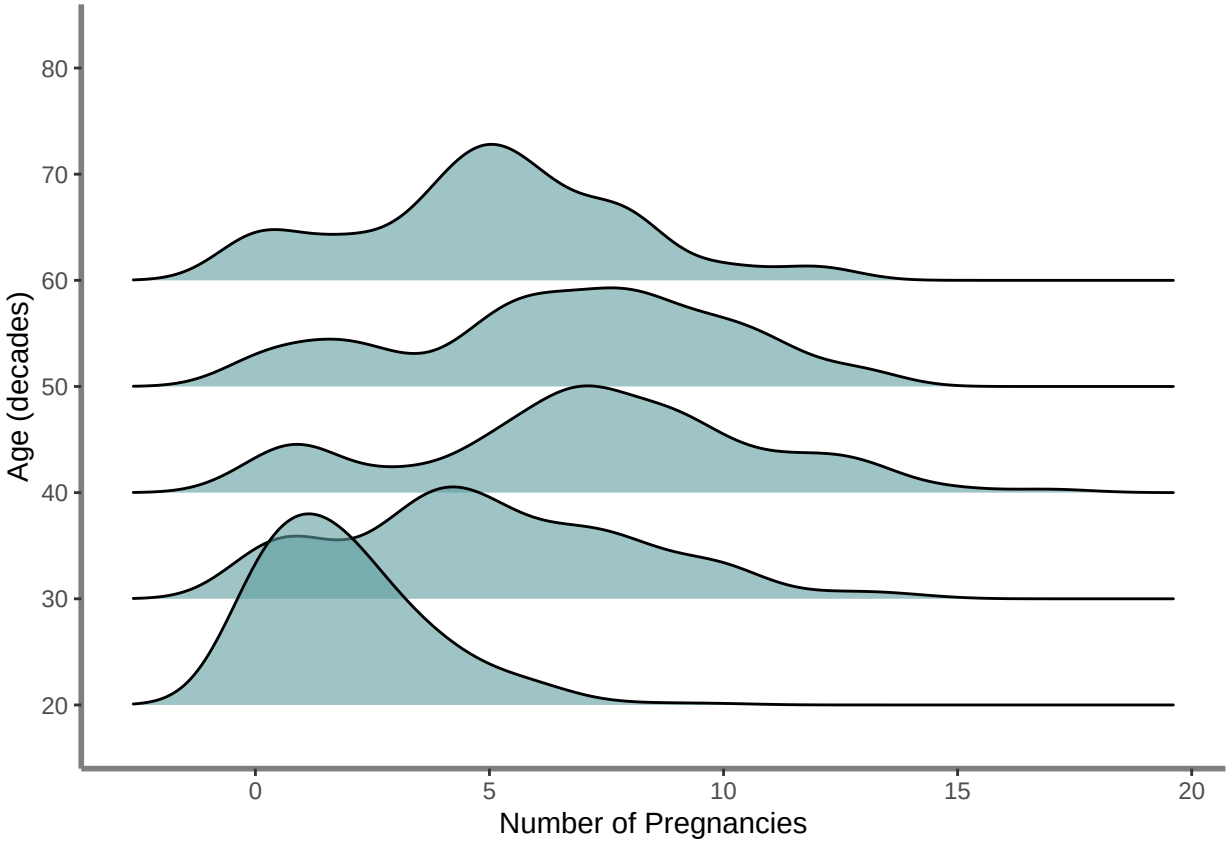


Figure 2: Exercise 1.2



1.3 Composite Plot: scatter plot and baxplot plot

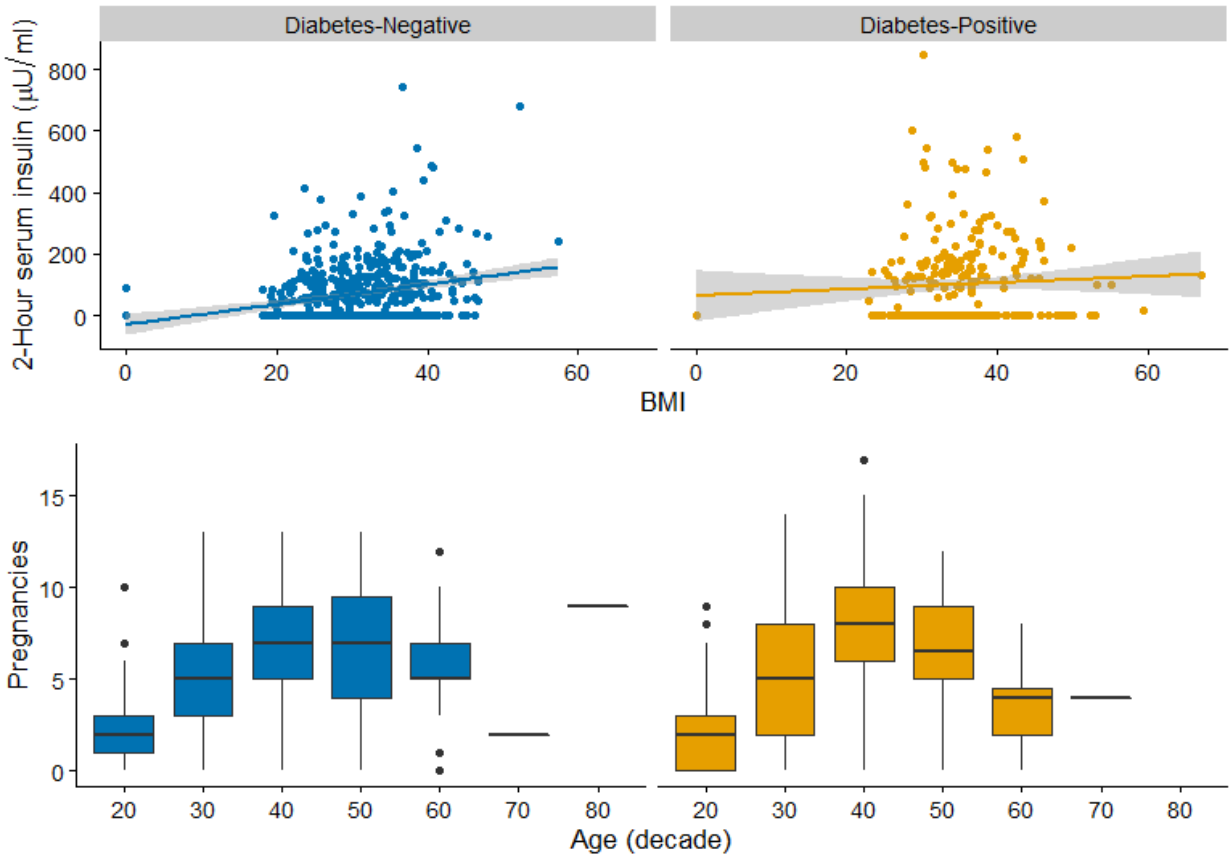
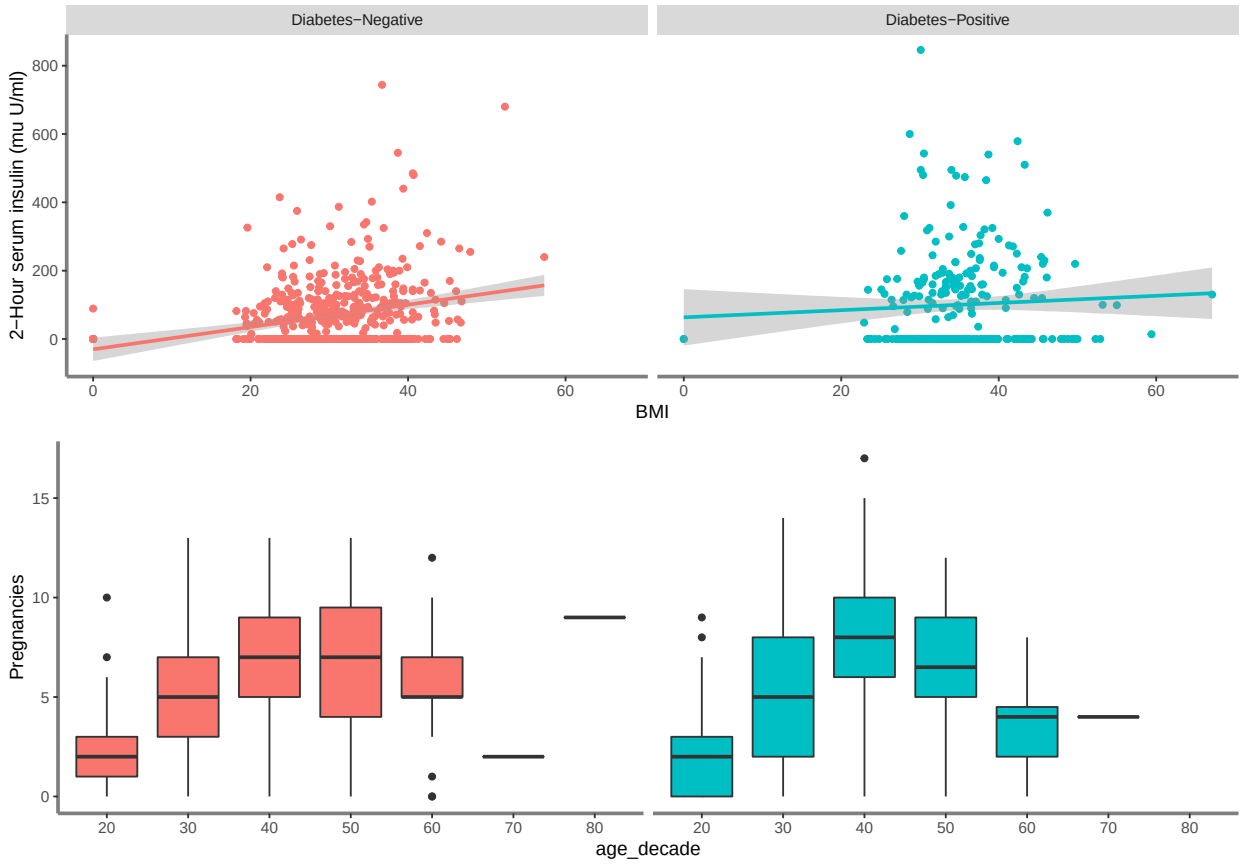


Figure 3: Exrcise 1.3

```
# Plot
# will take previously created groups from previous two exercises

serum_bmi_plot <- diab_outcome %>%
  ggplot(aes(BMI, Insulin, color = Outcome)) +
  facet_wrap(Outcome ~.) +
  #first layer
  geom_point() +
  #second layer
  geom_smooth(formula = "y~x", method = 'lm' ) +
  #theme and visual
  theme(panel.background = element_rect(fill = "white")) +
  theme(axis.line = element_line(size = 1, colour = "gray50", linetype=1))+
  ylab("2-Hour serum insulin (mu U/ml)") +
  theme(legend.position = 'none')

#serum_bmi_plot
#first half of plot works, move to plot 2

preg_age_plot <- diab_outcome %>%
  ggplot(aes(age_decade, Pregnancies, fill= Outcome)) +
```

```
  facet_wrap(Outcome ~.)+
  #layer 1
  geom_boxplot()+
  #theme and visual
  theme(panel.background = element_rect(fill = "white"),
        strip.text.x = element_blank(),
        axis.line = element_line(size = 1, colour = "gray50", linetype=1),
        legend.position = 'none')


#preg_age_plot
#second half works

#combine plots
plot_grid(serum_bmi_plot, preg_age_plot, ncol = 1)
```



Task 2: Apply principles of effective visualizations

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

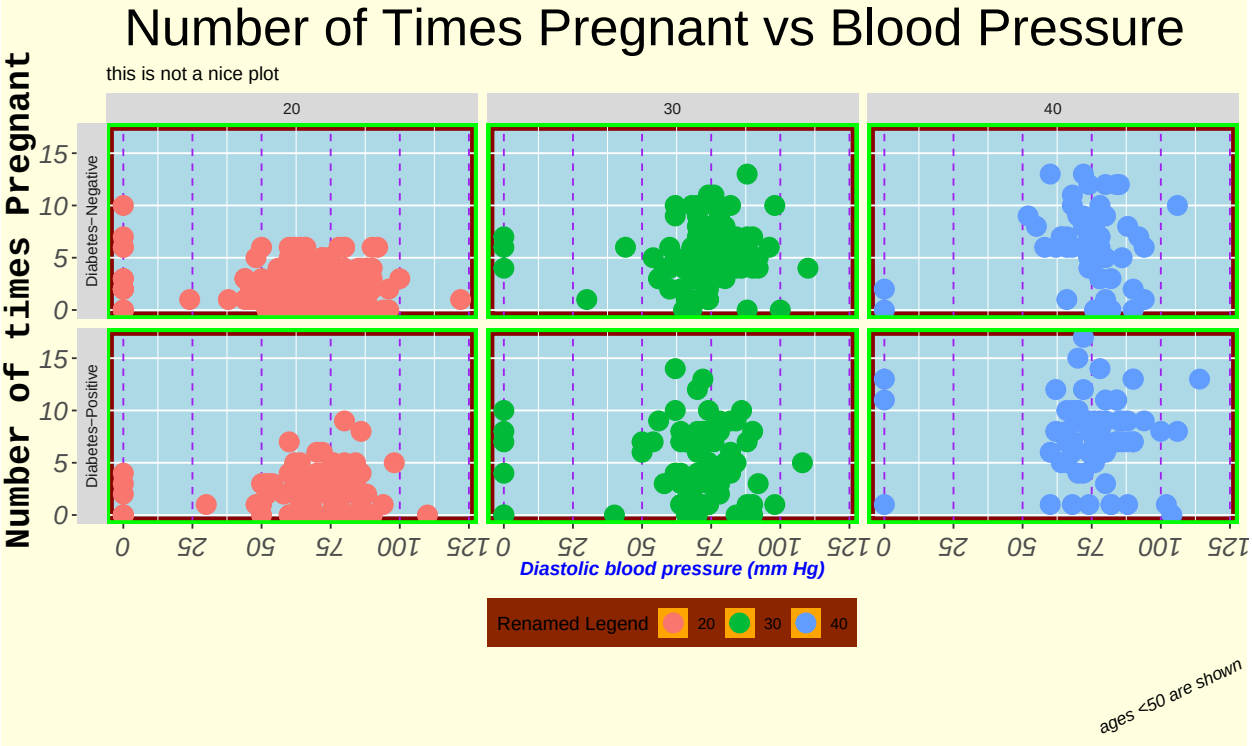
Just as a reminder, refer to the principles of effective visualizations when completing this task.

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.

Let's look at this plot:

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



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

plot. is used in the theme section, and is used prior to the other half of the argument such as background to communicate to R that the aspects that need to be customized are referring to the entire plot, the surrounding of the smaller data panels contained within. Plot is the encompassing, and targets the caption, title and background overall

panel. refers to the smaller data panels (figures) contained within the plot, anything passed to this argument will only affect the visuals of the data plots, but all that are contained within the plot

axis. allows customization of stylistic choices made to the x and y axes for each panel: fonts and sizing

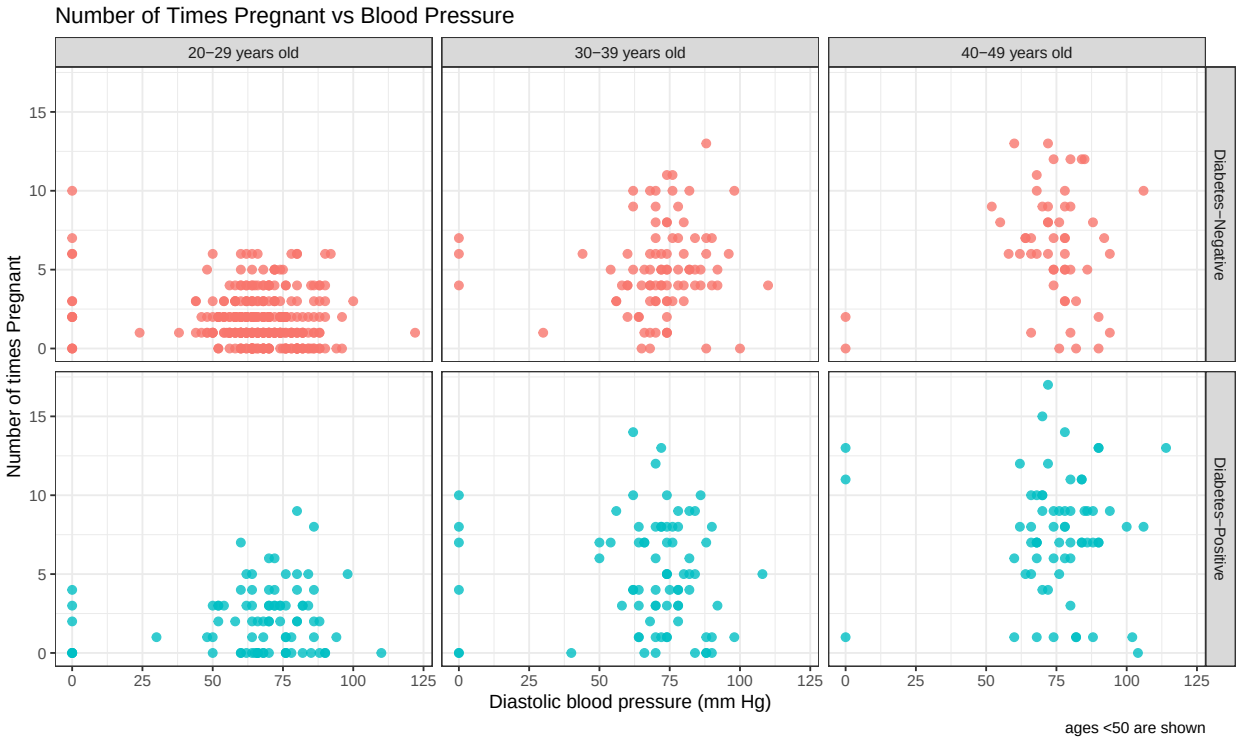
legend. argument modulates the visuals of the legend for the entire plot, and allows for customization of fonts, sizing

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?

This is studying relationships of the variables - an inferential question. But they are looking to see whether the outcome for the diabetes test will affect the relationship between diastolic blood pressure and the number of times one is pregnant. They break down these panels by age, by decade, as age is a factor that will affect blood pressure and the number of times a person will be pregnant. Notably, they excluded participants over 50 years old.

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.

```r
#recode the age decade column so it is better explained in the label, rather than just leaving as 20, 3
diab_2 <- diab_outcome %>%
  mutate(
    age_decade2 = fct_recode(age_decade,
      "20-29 years old" = "20",
      "30-39 years old" = "30",
      "40-49 years old" = "40",
      "50-59 years old" = "50",
      "60-69 years old" = "60",
      "70-79 years old" = "70",
      "80-89 years old" = "80"))

# Modified Plot
ggplot(diab_2 %>%  filter(Age<50),
       mapping = aes(x = BloodPressure, y = Pregnancies, col = Outcome)) +
  geom_point(size = 2, alpha = 0.8) +
  facet_grid(Outcome ~ age_decade2) +
  #theme elements
  theme_bw()+
  theme(
    #remove legend
    legend.position = "none") +

  #titles
  labs(title = "Number of Times Pregnant vs Blood Pressure",
       x = "Diastolic blood pressure (mm Hg)",
       y = "Number of times Pregnant",
       caption = "ages <50 are shown")
```



- I did not change the visualization type, but did change the theme elements to improve visibility, following the provided table above. I believe the current plot type is suitable for this research question
- proportional ink was not a problem in this, so I did not change this
- data:ink ratio: this needed a lot of work: redundant label: top label and colouring and labels for showing age completely unnecessary. Reduced the clutter to only have top label for age and removed colour with legend. Decided to colour by the diabetes outcome, because that will come closer to illustrating the relationship that the research question is trying to address. The background needs cleaning, it has way too many lines crossing through that do not show anything, while I did keep lines in, theyre minimal and do not distract, while also there to help with finding values up close if needed (personally help me identify the data points easier)
- labels and legend: change to consistent fonts and colours. Removed title subtitle, it was not adding any useful information - axes already illustrate with more conciseness
- points overlap too much, do not see anything, so will reduce the point size, and increase transparency
- colour is one of the worst - too many colours selected for this, reduced colour variation, and only used where necessary

Task 3: Advanced Figure Design

Using `iris` dataset reproduce the following plot:

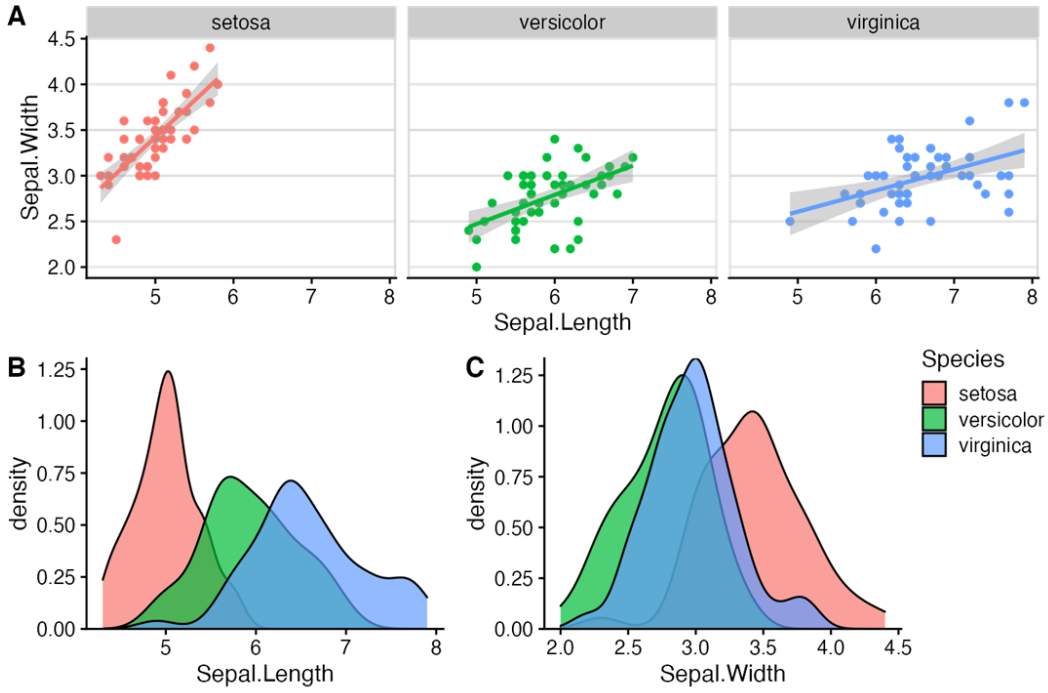
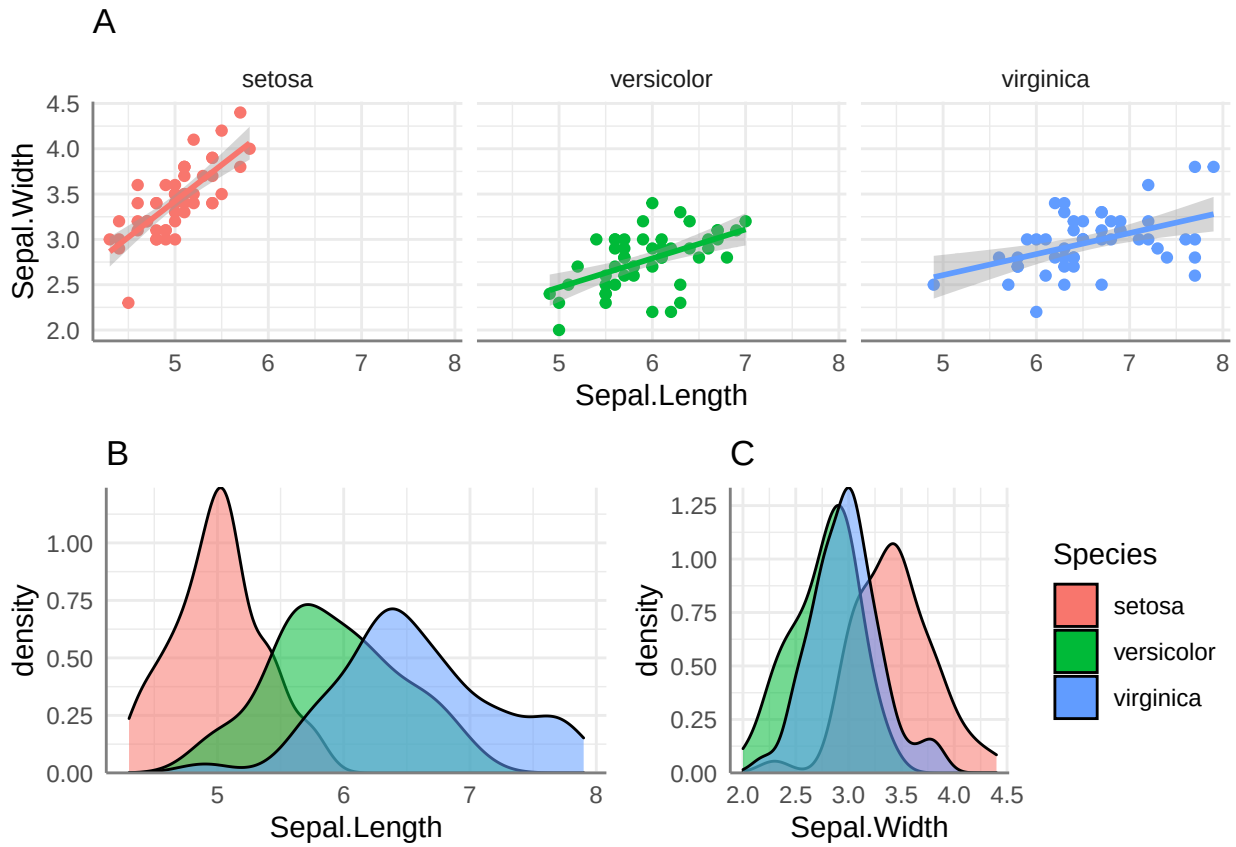


Figure 4: Plot title.

```
#panel A
plot_1 <- iris %>%
  ggplot(aes(Sepal.Length, Sepal.Width)) +
  geom_point(aes(color = Species))+
```

```
  facet_wrap(Species ~.) +
  geom_smooth(method = 'lm', formula = 'y~x', aes(color = Species))+
  #theme elements
  theme_minimal()+
  theme(
    #remove legend
    legend.position = 'none',
    axis.line = element_line(colour = "grey50")
  )+
  labs(
    title = "A"
  )
#panel B
plot_2 <- iris %>%
  ggplot(aes(Sepal.Length, fill = Species, alpha = 0.5))+
  geom_density()+
  #theme elements
  theme_minimal()+
  theme(
    #remove legend
    legend.position = 'none',
    axis.line = element_line(colour = "grey50")
  )+
  labs(
    title = "B"
  )+
  #rescale the y axes, force start at 0
  scale_y_continuous(breaks=c(0.00, 0.25, 0.5, 0.75, 1.00, 1.25), expand = c(0,0))

plot_3 <- iris %>%
  ggplot(aes(Sepal.Width, fill = Species))+
  geom_density(aes(alpha = 0.5))+
  #theme elemends
  theme_minimal()+
  theme(
    axis.line = element_line(colour = "grey50")
  )+
  labs(
    #do not remove legend
    title = "C"
  )+
  #rescale the y axes, force start at 0
  scale_y_continuous(breaks=c(0.00, 0.25, 0.5, 0.75, 1.00, 1.25), expand = c(0,0))+
  #remove legend for alpha value
  scale_alpha(guide = 'none')

#combine bottom panel first side by side
bottom_panel = plot_grid(plot_2, plot_3)

#plot final graph and stack on top of each other
plot_grid(plot_1, bottom_panel, ncol = 1)
```



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

Import the dataset `ethnicity_data.csv` and examine how "messy" the data input format.

```
ethnicity_data <- read_csv('data/ethnicity_data_fix.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.
```

```
head(ethnicity_data)
```

```
## # A tibble: 6 x 2
##      ID Ethnicity
##   <dbl> <chr>
## 1     1 White
## 2     2 Chinese (including Hong Kong and Macau)
## 3     3 White
## 4     4 Chinese (including Hong Kong and Macau)
## 5     5 Chinese (including Hong Kong and Macau)
## 6     6 South Asian (e.g. Indian, Pakistani, Sri Lankan, etc.)
```

```
tail(ethnicity_data)
```

```
## # A tibble: 6 x 2
##      ID Ethnicity
##   <dbl> <chr>
## 1   480 <NA>
## 2   481 <NA>
## 3   482 <NA>
## 4   483 <NA>
## 5   484 <NA>
## 6   485 None of the above
```

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.

```r
# tidying up dataset
# I'll drop the NAs first, and then see the counts of each for the time being
ethnicity_count <- ethnicity_data %>%
  drop_na() %>%
  group_by(Ethnicity)%>%
  summarise(count = n())

#this shows me that there is a big untidyness due
#to people clicking more than one, and it being recorded as an individual category

#for now will delete all of those instances so I could
#get an idea of what I'm looking at, and unfortunately
#have to consider how to break up those instances in the future

ethnicity_count <- ethnicity_count %>%
  filter(!count == 1)


#will try to at least make them factors first

ethnicity_count <- ethnicity_count %>%
  mutate(
    Ethnicity = as.factor(Ethnicity))


ethnicity_count <- ethnicity_count %>%
  mutate(Ethnicity = fct_recode(Ethnicity, "Canada Indigenous" =
                                  "Indigenous person of Canada (First Nations, Inuit, Metis)",
                                "Latin, Central, or South American" =
                                  "Latin, Central, or South American (e.g. Brazilian, Chilean, Colombian, Mex
                                "South Asian" =
                                  "South Asian (e.g. Indian, Pakistani, Sri Lankan, etc.)",
                                "Southeast Asian" =
                                  "Southeast Asian (e.g. Cambodian, Indonesian, Laotian Vietnamese, etc)",
                                )
  )

#I tried to reorganize this even at the bare level,
#but I was having issues and it was not working here,
#when I try to do the factor recoding, there was a big problem
#with the Metis word, had to change in the csv file

levels(ethnicity_count$Ethnicity)
```

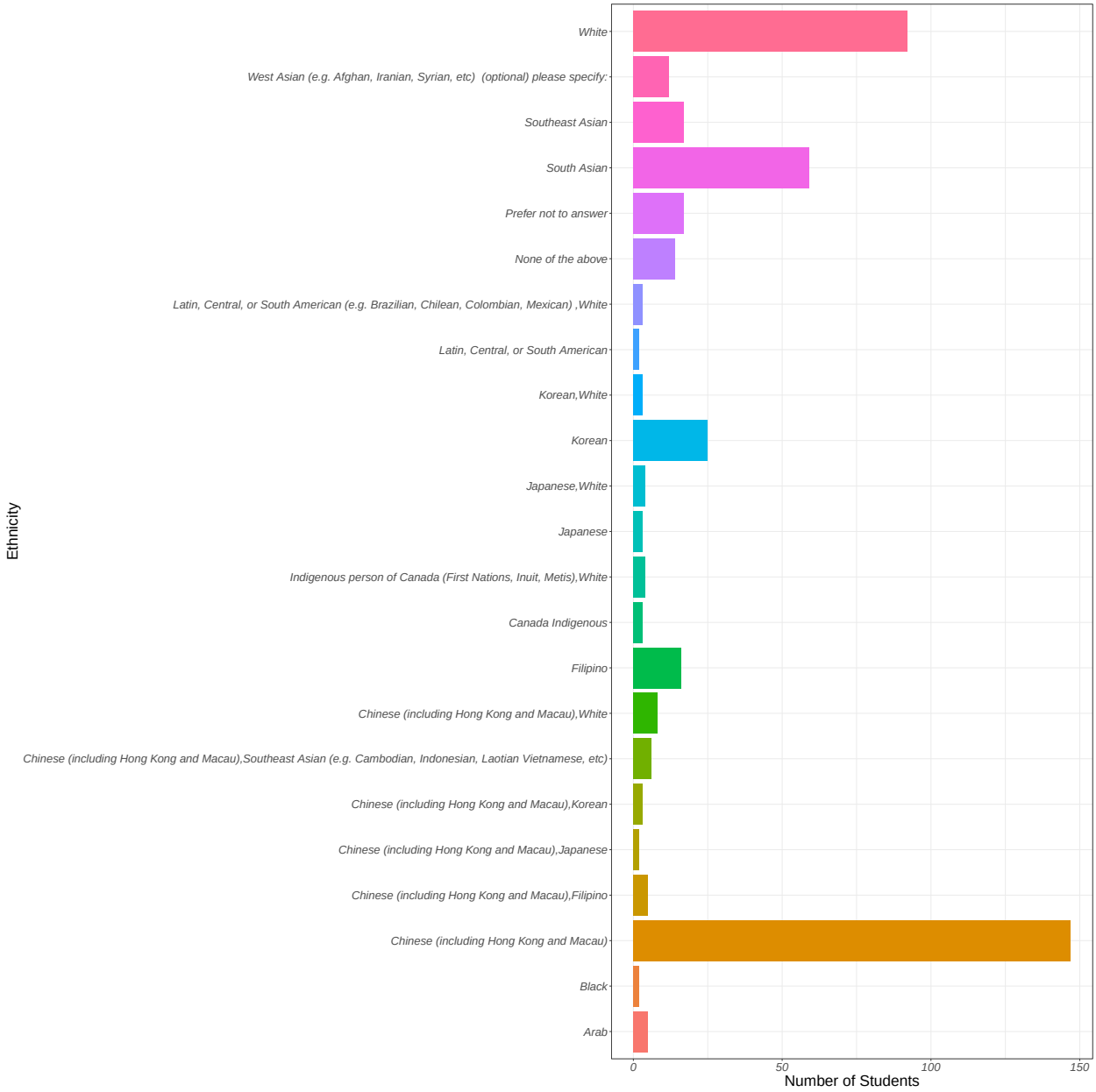
```
##  [1] "Arab"
##  [2] "Black"
##  [3] "Chinese (including Hong Kong and Macau)"
##  [4] "Chinese (including Hong Kong and Macau),Filipino"
##  [5] "Chinese (including Hong Kong and Macau),Japanese"
##  [6] "Chinese (including Hong Kong and Macau),Korean"
##  [7] "Chinese (including Hong Kong and Macau),Southeast Asian (e.g. Cambodian, Indonesian, Laotian Vi
##  [8] "Chinese (including Hong Kong and Macau),White"
##  [9] "Filipino"
```

```
## [10] "Canada Indigenous"
## [11] "Indigenous person of Canada (First Nations, Inuit, Metis),White"
## [12] "Japanese"
## [13] "Japanese,White"
## [14] "Korean"
## [15] "Korean,White"
## [16] "Latin, Central, or South American"
## [17] "Latin, Central, or South American (e.g. Brazilian, Chilean, Colombian, Mexican) ,White"
## [18] "None of the above"
## [19] "Prefer not to answer"
## [20] "South Asian"
## [21] "Southeast Asian"
## [22] "West Asian (e.g. Afghan, Iranian, Syrian, etc)  (optional) please specify:"
## [23] "White"
```

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 okey 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.

```
# loaded extra libraries
ethn_plot <- ethnicity_count %>%
  ggplot(aes(count, Ethnicity, fill = Ethnicity)) +
  geom_bar(stat='identity')+
  theme_bw()+
  theme(
    legend.position = 'none',
    axis.text = element_text(face = "italic", size = 15, angle = 0),
    axis.title.x = element_text(size = 20),
    axis.title.y = element_text(size = 20)

  )+
  xlab("Number of Students")


ethn_plot
```



This is not a helpful graph at all, because there is again that issue of individual selecting more than one ethnicity and that not registering as an individual in both groups, but rather putting themselves into their own group. So this graph does not currently show all the samples that were collected, not making it a good graph. Other issues include how long some of the ethnicities are, which is again tied to the issue I mentioned earlier of people being more than one ethnicity. In an ideal circumstance, there would be several panels. One panel will show the individual columns for ethnicities, where people selected more than one will also show up in the graph more than once. Another panel will give a distribution of how many people selected more than one ethnicity and a graph showing those invididuals values - this could perhaps be more closely tied to the ID of participant. Or this could be a graph that has a heat map of overlaps in ethnicity. The ideal solution is to tidy the data considerably more, and then etiher show percentages of people having one ethnicity vs more than one or have another bar graph to accompany it in a second panel in addition to what I show below.

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.