STAT3622_Final project

Tang Yihan

2024-04-26

UID: 3036051639

Name: Tang Yihan

File name: STAT 3622 Final project R markdown

Note: This file only contains the R code. Other parts that include Python (model and analysis) are presented in separate files.

Table of Contents:

Data loading

First, load in the necessary libraries.

The base dataset is the ICUSTAYS.csv. We first merge it with patient information dataset.

Similarly, we load in hospital admission, diagnosis, and ICD mapping dataset. Eventually, we get an integrated dataset of ICU information, consisting of over 60,000 rows and 23 columns. Note: This process usually takes a few minutes locally due to the large volume of data.

```r
setwd("/Users/hanktang/Desktop/ML_project_environments/mimic-iii-clinical-database-1.4")
admissions <- read.csv("ADMISSIONS.csv")
icu$ethnicity <- NA
icu$insurance <- NA
icu$admit_time <- NA
icu$discharge_time <- NA
icu$admission_type <- NA
icu$diagnosis <- NA

for (i in 1:nrow(icu)) {
  hadm_id <- icu$HADM_ID[i]
  matching_row <- admissions$HADM_ID == hadm_id

  # If a match is found, retrieve the 'dob' value from the Patients dataset
  if (any(matching_row)) {
    icu$ethnicity[i] <- admissions$ETHNICITY[matching_row]
    icu$insurance[i] <- admissions$INSURANCE[matching_row]
    icu$admit_time[i] <- admissions$ADMITTIME[matching_row]
    icu$discharge_time[i] <- admissions$DISCHTIME[matching_row]
    icu$admission_type[i] <- admissions$ADMISSION_TYPE[matching_row]
    icu$diagnosis[i] <- admissions$DIAGNOSIS[matching_row]
  }
}


DIAGNOSIS_ICD <- read.csv("DIAGNOSES_ICD.csv")
icu$ICD_diagnosis <- NA

for (i in 1:nrow(icu)) {
  hadm_id <- icu$HADM_ID[i]
  matching_row <- DIAGNOSIS_ICD$HADM_ID == hadm_id

  if (any(matching_row)) {
    icu$ICD_diagnosis[i] <- DIAGNOSIS_ICD$ICD9_CODE[matching_row]
  }
}

PROCEDURES_ICD <- read.csv("PROCEDURES_ICD.csv")
icu$ICD_procedures <- NA

for (i in 1:nrow(icu)) {
  hadm_id <- icu$HADM_ID[i]
  matching_row <- PROCEDURES_ICD$HADM_ID == hadm_id

  if (any(matching_row)) {
    icu$ICD_procedures[i] <- PROCEDURES_ICD$ICD9_CODE[matching_row]
  }
}
```

Now, we have loaded all the necessary data into the ICU dataset. We can start to work on EDA and visualization.

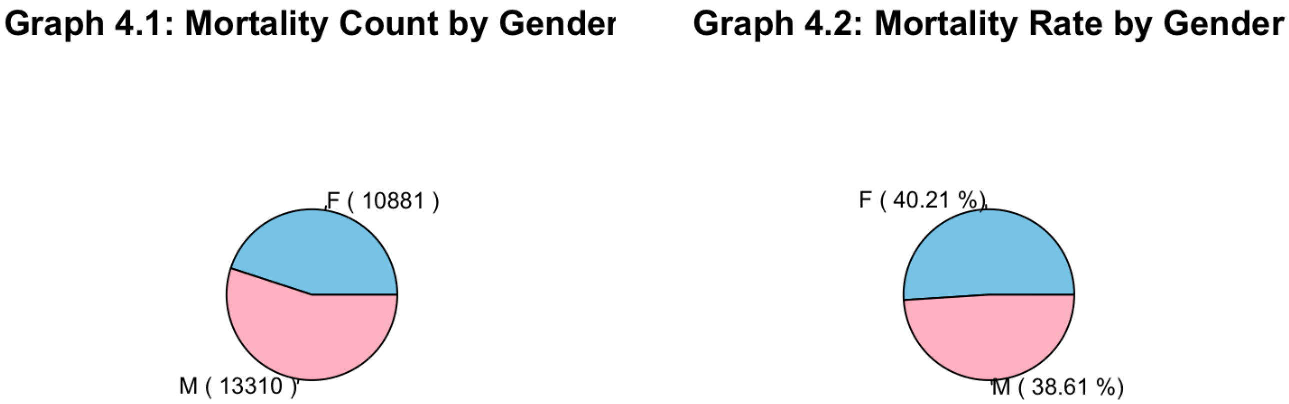
There are two objectives in this project. The first objective is mortality prediction in the next 48 hours. The second objective is the length of stay of patients in ICU. First, we look into in-hospital mortality scenario. There are several extraneous variables such as gender, race, last ICU ward, and so on. We would like to generate visualization based on each of the abovementioned variables to examine whether they have significant impacts on the mortality outcome of patients.

1.1 Mortality count by gender

```
mortality_data <- icu[icu$death_flag == 1, ]

gender_counts <- table(mortality_data$gender)

par(mfrow = c(1, 2))

# Create a pie chart
pie(gender_counts,
    labels = paste(names(gender_counts), "(", gender_counts, ")"),
    main = "Graph 1.1: Mortality Count by Gender",
    col = c("skyblue", "pink"),
    cex = 0.8)
```

1.2 Mortality rate by gender

```
icu_gender_counts <- table(icu$gender)
mortality_gender_counts <- table(mortality_data$gender)
proportions <- mortality_gender_counts / icu_gender_counts
pie(proportions,
    labels = paste(names(proportions), "(", round(proportions * 100, 2), "%)"),
    main = "Graph 1.2: Mortality Rate by Gender",
    col = c("skyblue", "pink"),
    cex = 0.8)
par(mfrow = c(1, 1))
```

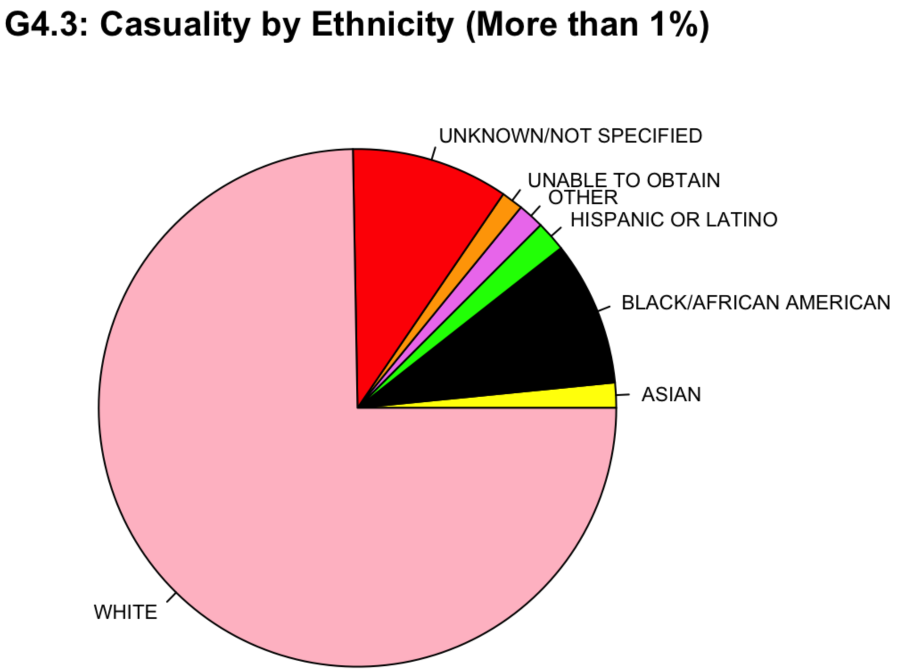


As shown in the graph 1.1 and 1.2, the mortality rate of male and female patients are generally similar. This means that we do not have to separate males and females in later modelling.

Then, we examine data of different ethnicities.

1.3 Mortality count by Ethnicity (>1%)

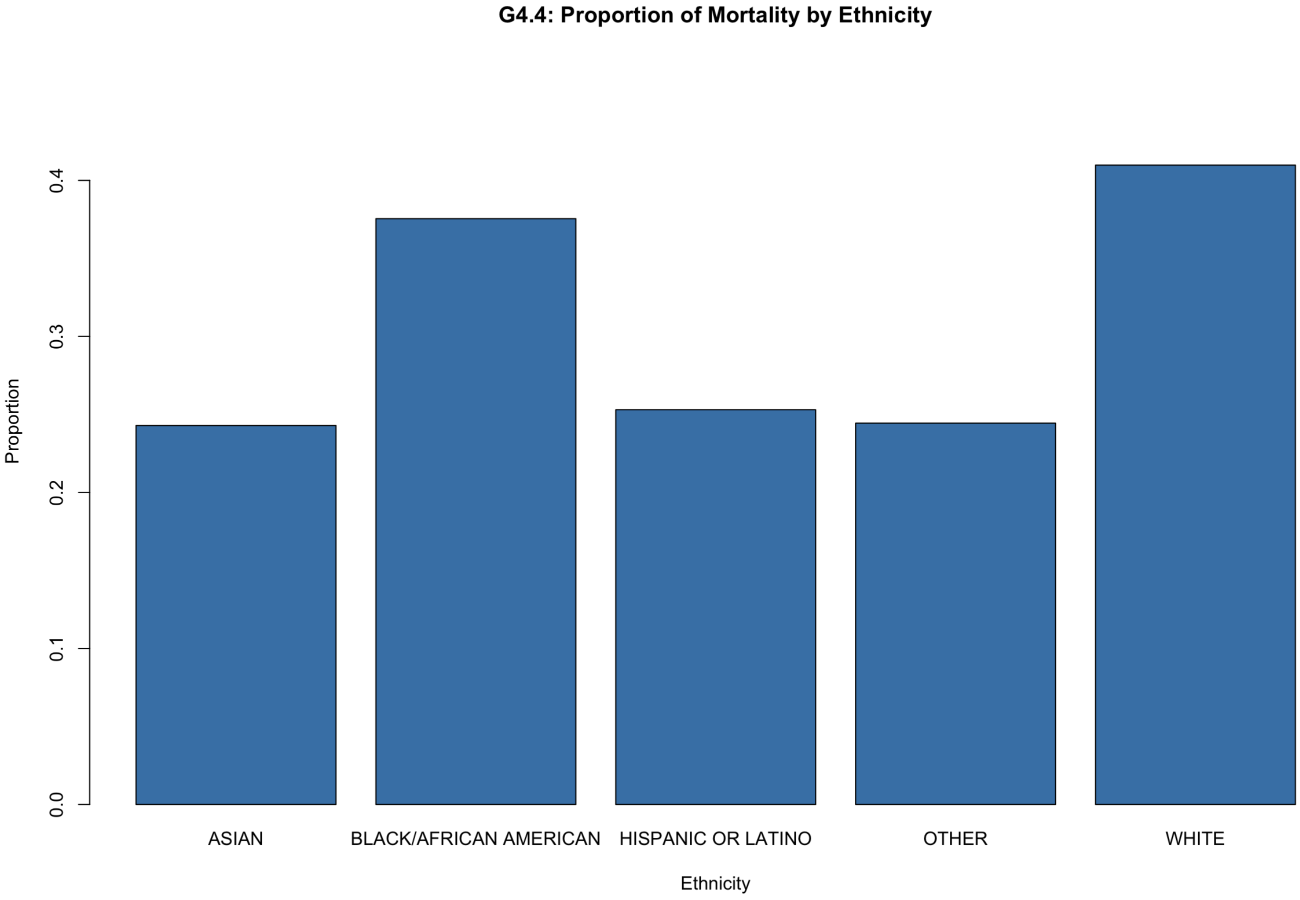
```
ethnicity_counts <- table(mortality_data$ethnicity)
filtered_counts <- ethnicity_counts[ethnicity_counts > 240]
colors = c("yellow", "black", "green", "violet", "orange","red", "pink", "cyan")
pie_1 <- pie(filtered_counts, main = "G1.3: Casuality by Ethnicity (More than 1%)", cex = 0.7, col = col
```



1.4 Proportion of Mortality by Ethnicity

```r
icu_ethnicity_counts <- table(icu$ethnicity)
mortality_ethnicity_counts <- table(mortality_data$ethnicity)

group_to_drop <- "UNKNOWN/NOT SPECIFIED"
filtered_ethnicities <- names(icu_ethnicity_counts)[icu_ethnicity_counts > 1200 & names(icu_ethnicity_c
filtered_icu_counts <- icu_ethnicity_counts[filtered_ethnicities]
filtered_mortality_counts <- mortality_ethnicity_counts[filtered_ethnicities]

# Calculate proportions
proportions <- filtered_mortality_counts / filtered_icu_counts

# Set the upper margin size
par(mar = c(5, 4, 2, 2) + 0.1)

# Create a bar chart with adjusted y-axis limits
barplot(proportions,
        main = "G1.4: Proportion of Mortality by Ethnicity",
        xlab = "Ethnicity",
        ylab = "Proportion",
        col = "steelblue",
        ylim = c(0, max(proportions) * 1.2))
```

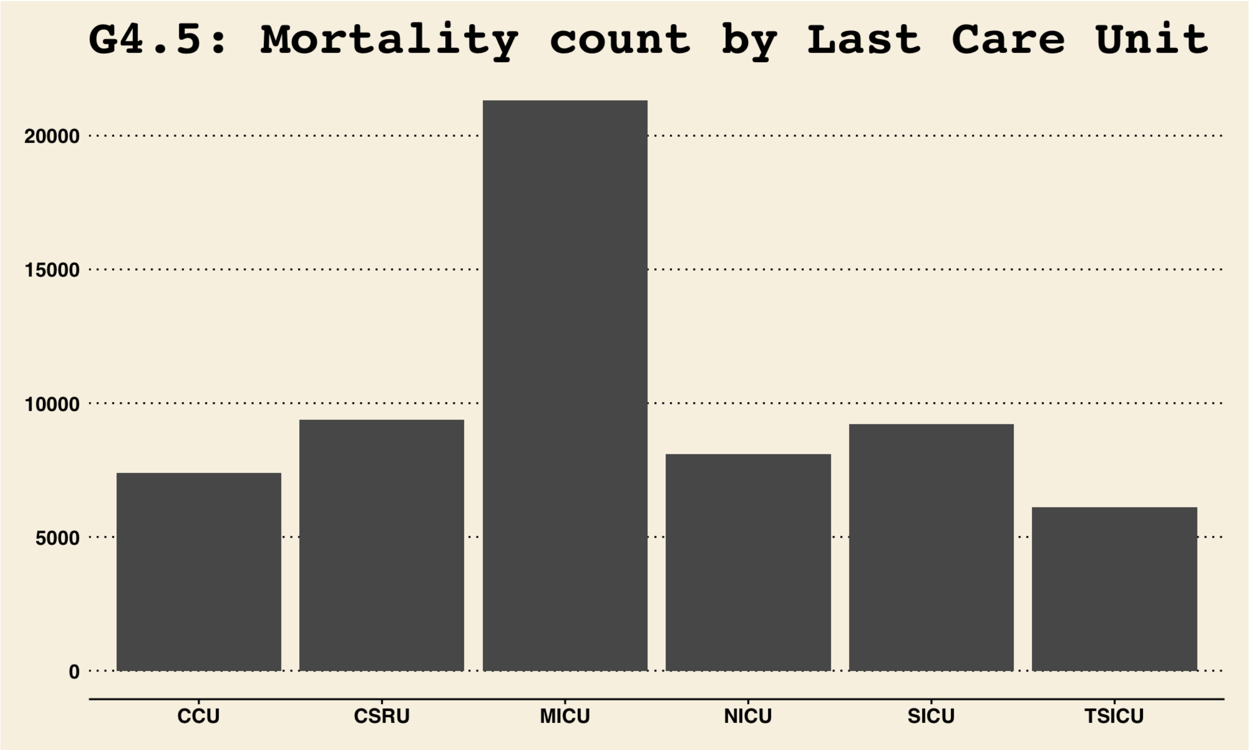
```
# Add labels to the bars
text(x = barplot(proportions) - 0.3,
     y = proportions + 0.02,
     labels = paste0(round(proportions * 100, 2), "%"),
     pos = 3,
     col = "black")
```



As shown in graph 1.3 and 1.4, although the gender makeup of the dataset is unbalanced, the mortality rates of patients of different races do not shown a great disparity. Particularly, the mortality rates of the two dominant classes are similar.

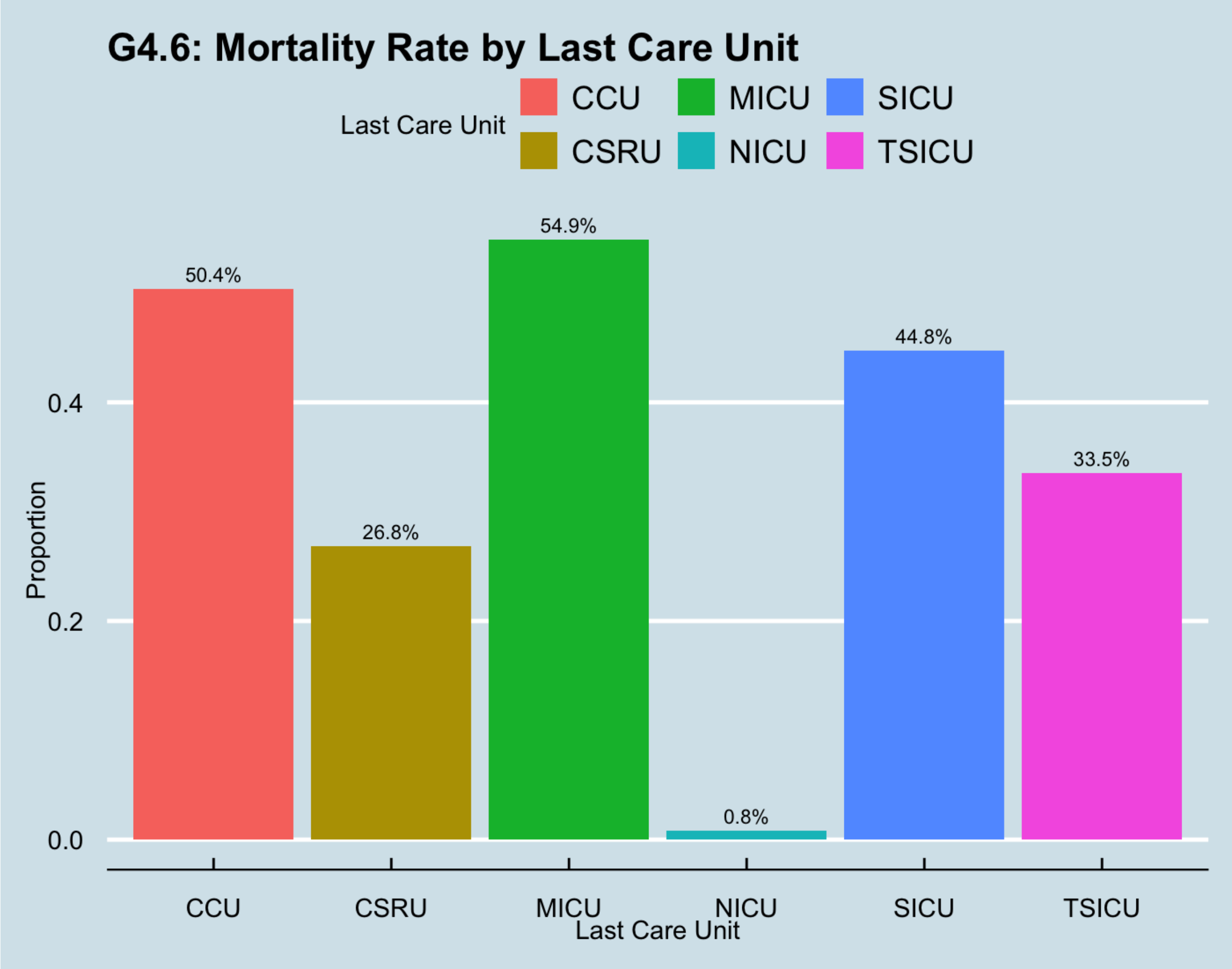
1.5 Mortality count by Last Care Unit (ICU ward type)

```
# Create a stacked bar plot
ggplot(data = icu, aes(x = LAST_CAREUNIT, fill = death_flag)) +
  geom_bar() +
  labs(x = "Last Care Unit", y = "Count", fill = "Mortality") +
  ggtitle("G1.5: Mortality count by Last Care Unit") +
  theme_wsj()
```



1.6 Mortality Rate by Last Care Unit

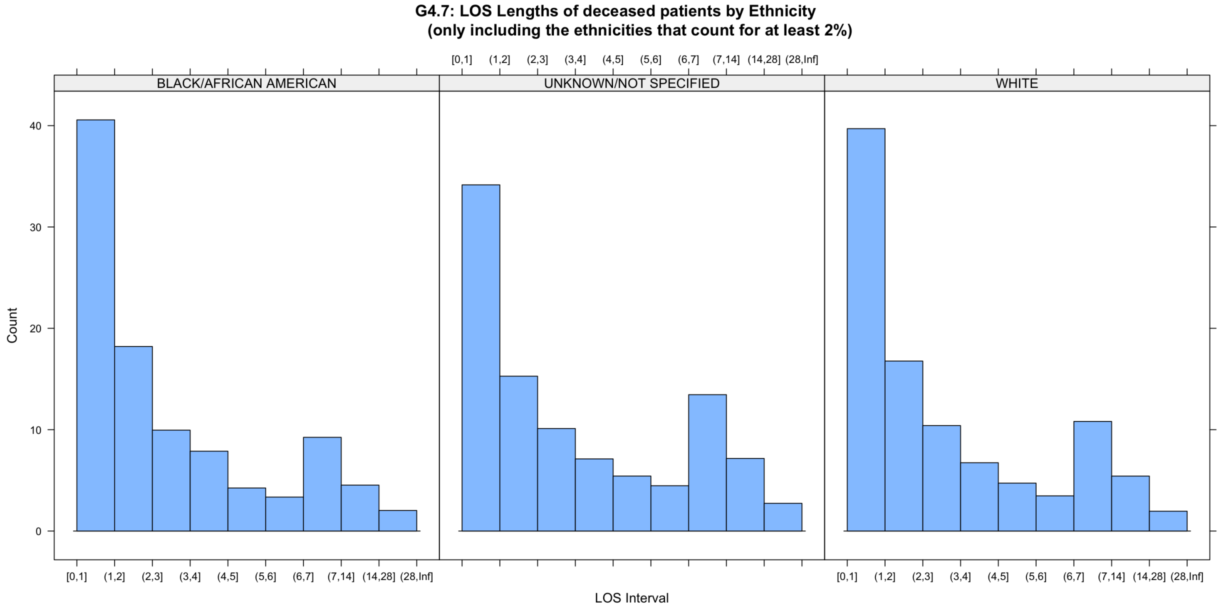
```r
proportions <- icu %>%
  group_by(LAST_CAREUNIT) %>%
  summarize(proportion = sum(death_flag == 1) / n())

ggplot(data = proportions, aes(x = LAST_CAREUNIT, y = proportion, fill = LAST_CAREUNIT)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = scales::percent(proportion)), vjust = -0.5, size = 3) +  # Add text labels
  labs(x = "Last Care Unit", y = "Proportion", fill = "Last Care Unit") +
  ggtitle("G1.6: Mortality Rate by Last Care Unit") +
  theme_economist()
```



As shown in graph 1.5 and 1.6, the mortality rate in NICU (neonatal Intensive Care Unit) is extremely low compared with other ICU wards. This prompts the author to separate minors from the later modelling analysis. Also, it is widely recognized in the medical community that minors have clearly different physiological patterns from adults.

1.7 LOS Lengths of deceased patients by Ethnicity

```
intervals <- c(0, 1, 2, 3, 4, 5, 6, 7, 14, 28, Inf)
mortality_data$LOS_interval <- cut(mortality_data$LOS, breaks = intervals, include.lowest = TRUE)

majority_races <- names(ethnicity_counts[ethnicity_counts > 1300])
selected_data <- mortality_data[mortality_data$ethnicity %in% majority_races, ]

histogram(~ LOS_interval | ethnicity, data = selected_data,
          layout = c(length(majority_races), 1),
          breaks = 10,
          main = "G1.7: LOS Lengths of deceased patients by Ethnicity
          (only including the ethnicities that count for at least 2%)",
          xlab = "LOS Interval",
          ylab = "Count")
```



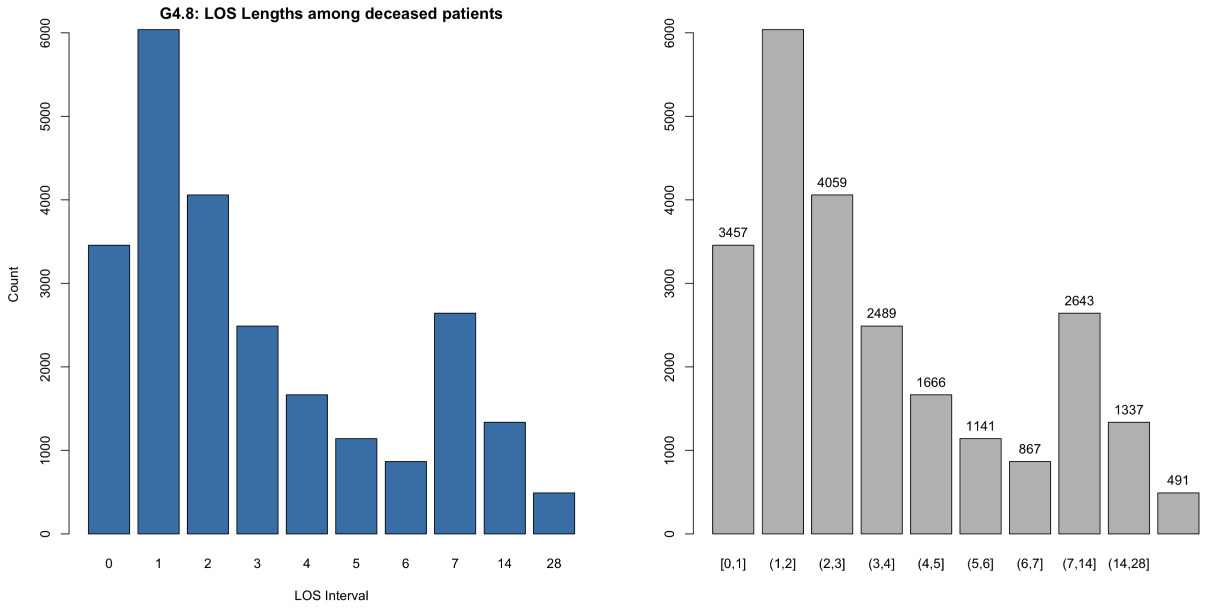
This part of visualization examines whether ethnicity is an important extraneous variable. As is shown in graph 1.7, the length of stay of patients of different races generally follow the same distribution. The distributions are all with a heavy tail, with 0-1 days being the dominant interval. Therefore, we do not need to design treatment groups for patients of different races for objective 1.

Part 2: Here is the length of stay (LOS) part visualization.

In part 2, the author focuses on visualization related to objective 2, the length of stay of patients in ICU. It should be noted that the end of stay in ICU means either discharge or death.

1.8 Overview of LOS (Length of Stay) of deceased patients

```
par(mfrow=c(1,2))
intervals <- c(0, 1, 2, 3, 4, 5, 6, 7, 14, 28, Inf)
mortality_data$LOS_interval <- cut(mortality_data$LOS, breaks = intervals, include.lowest = TRUE)

mortality_subset_counts <- table(mortality_data$LOS_interval)

ordered_intervals <- sort(intervals[-length(intervals)])

barplot(mortality_subset_counts,
        names.arg = ordered_intervals,
        xlab = "LOS Interval",
        ylab = "Count",
        main = "G1.8: LOS Lengths among deceased patients",
        col = "steelblue")

text_pos <- barplot(mortality_subset_counts)  # Get the x-coordinates of the bars
text_height <- mortality_subset_counts + 3  # Adjust the height of the text above the bars

text(x = text_pos,
     y = text_height,
     labels = mortality_subset_counts,
     pos = 3)

par(mfrow=c(1,1))
```

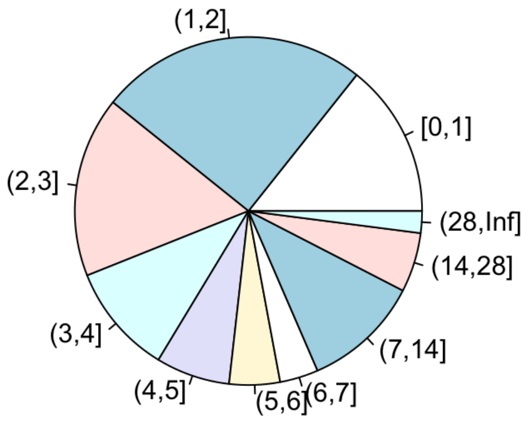


G4.8: LOS Lengths among deceased patients

1.9 Proportion of LOS Lengths in ICU among deceased patients.

```
pie(mortality_subset_counts, labels = names(mortality_subset_counts),
    main = "G1.9: Proportion of LOS Lengths of deceased patients.")
```

G4.9: Proportion of LOS Lengths of deceased patients.

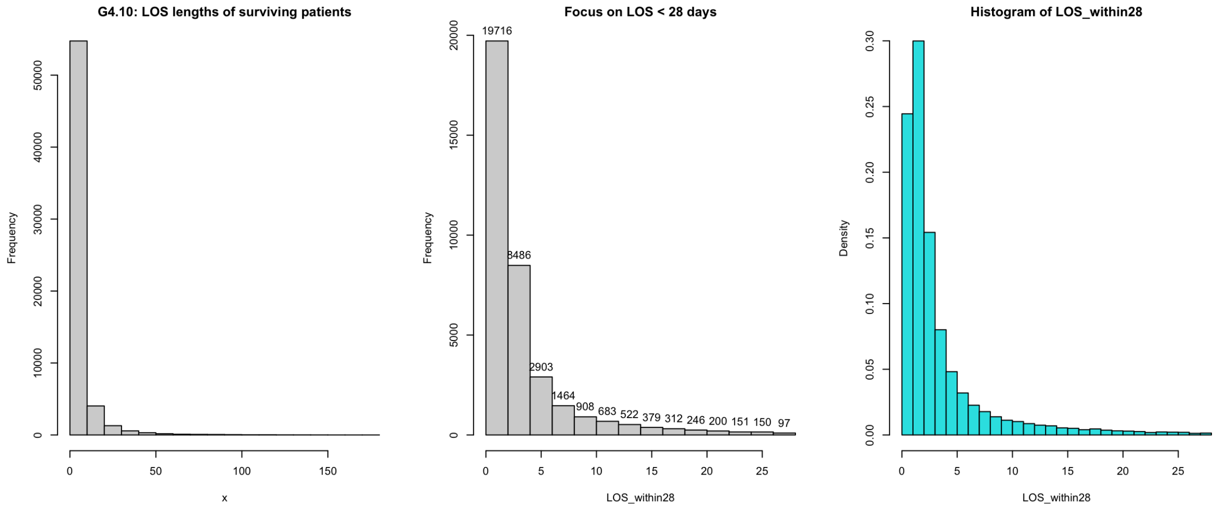


In graph 1.8 and 1.9, the author looks into the distribution of length of stay in ICU among deceased patients.

This distribution will be compared with that of surviving patients in later sections.

1.10 Overview of LOS (Length of Stay) of surviving patients

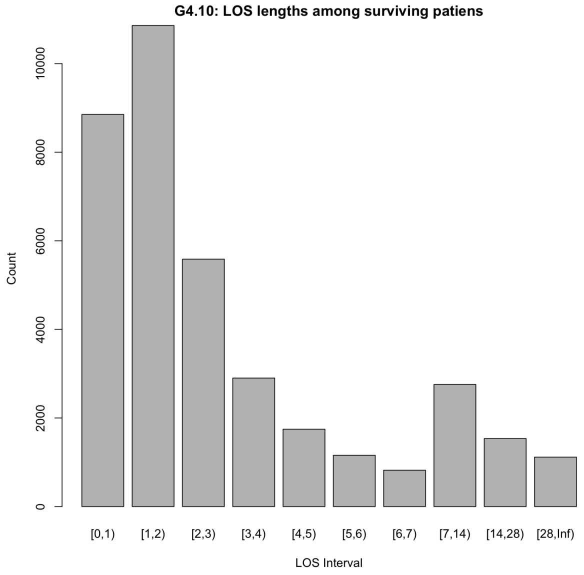
```
x = icu$LOS
alive_data <- icu[icu$death_flag == 0, ]

par(mfrow=c(1,3))
hist(x, main = "G1.10: LOS lengths of surviving patients")

LOS_within28 <- alive_data[alive_data$LOS < 28, ]$LOS
LOS_within28 <- LOS_within28[!is.na(LOS_within28)]
hist_data <- hist(LOS_within28, main="Focus on LOS < 28 days")
text(hist_data$mids, hist_data$counts, labels = hist_data$counts, pos = 3, offset = 0.5)

hist(LOS_within28, breaks=28, freq = F, col=5)

intervals <- c(0, 1, 2, 3, 4, 5, 6, 7, 14, 28, Inf)
alive_data$LOS_interval <- cut(alive_data$LOS, breaks = intervals, right = FALSE)
split_data <- split(alive_data, alive_data$LOS_interval)
```

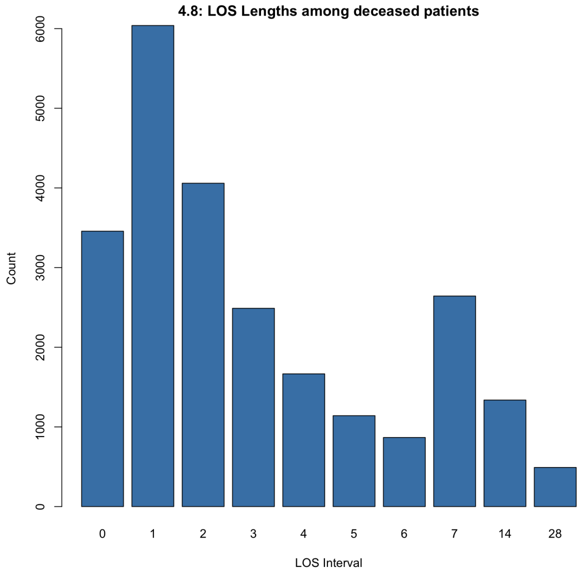


In graph 1.10, the author looks into the distribution of length of stay in ICU among surviving patients. This distribution exhibits a clear heavy tail as well as a gradual decrease.

Comparison between LOS lengths among surviving and deceased patients

```
par(mfrow=c(1,2))
subset_counts <- sapply(split_data, nrow)
barplot(subset_counts, main = "G1.10: LOS lengths among surviving patiens",
        xlab = "LOS Interval", ylab = "Count")

barplot(mortality_subset_counts,
        names.arg = ordered_intervals,
        xlab = "LOS Interval",
        ylab = "Count",
        main = "1.8: LOS Lengths among deceased patients",
        col = "steelblue")
```

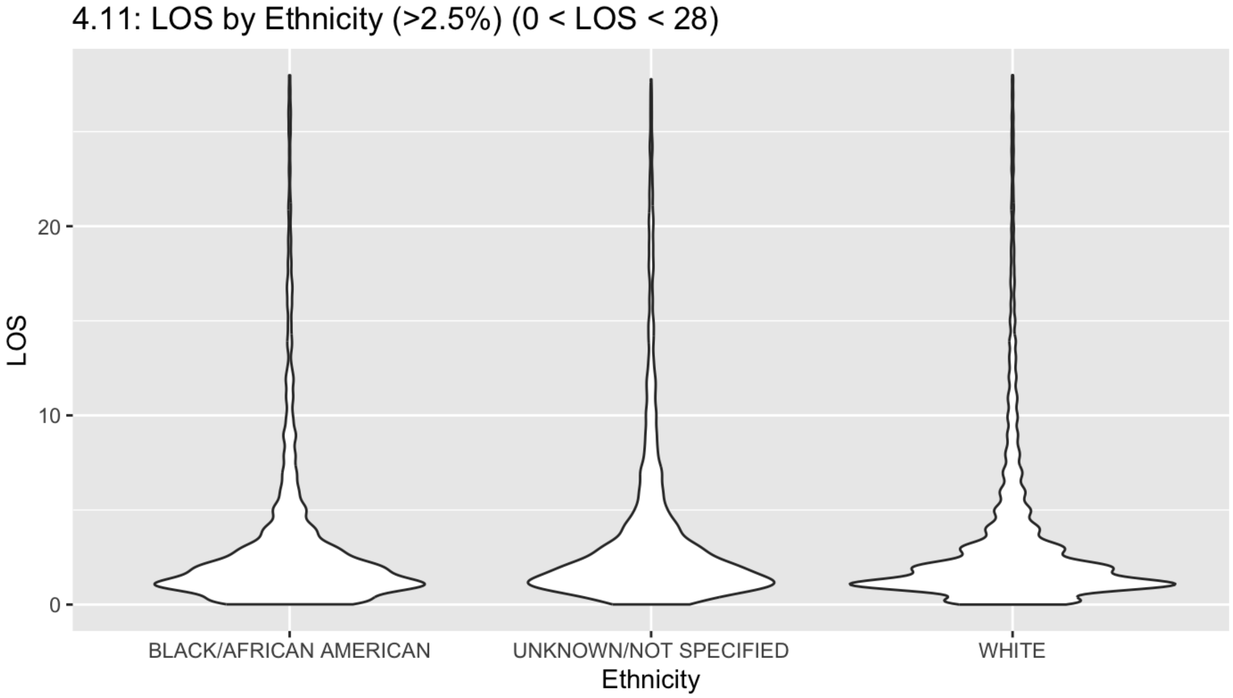




Here is a comparison between the Length Of Stay distribution of surviving and deceased patients. There is an important insight from the comparison. The LOS of both groups of patients exhibit a heavy tail and both lean towards shorter time intervals. This provides previous information for how we devise our second objective Length Of Stay. If we were to devise it as a linear regression problem, a linear function would be fitted to favor the shorter length of stay heavily, thus sacrificing precision in longer length of stay. This is not ideal because patients who may stay in ICU for more than 4 days will be incorrectly treated by the model. Therefore, a more sensible approach would be to model the problem as a multi-class classification problem. Hence, we will use Cohen's Cappa Score as metrics of model performance.

1.11 Length Of Stay by Ethnicity (>2.5%)

```r
par(mfrow=c(1,1))
majority_races <- names(ethnicity_counts[ethnicity_counts > 1400])
filtered_alive <- subset(alive_data, ethnicity %in% majority_races)

qplot(data = subset(filtered_alive, LOS > 0 & LOS < 28),
      x = ethnicity, y = LOS, geom = "violin",
      main = "1.11: LOS by Ethnicity (>2.5%) (0 < LOS < 28)",
      xlab = "Ethnicity", ylab = "LOS")
```



From graph 1.11, we observe that ethnicity does not seems to impact the LOS of patients. Therefore, we do not need to devise treatment groups for ethnicity in the second objective.

Overall, we find that ethnicity and gender are not important extraneous variables. We do not need to separate the patients by their ethnicity or gender.

For the remaining part of Python code, please visit: https://github.com/Hank-Tang/MIMIC-III-for-STAT3622/tree/main