First import data and all necessary packages as well as libraries:

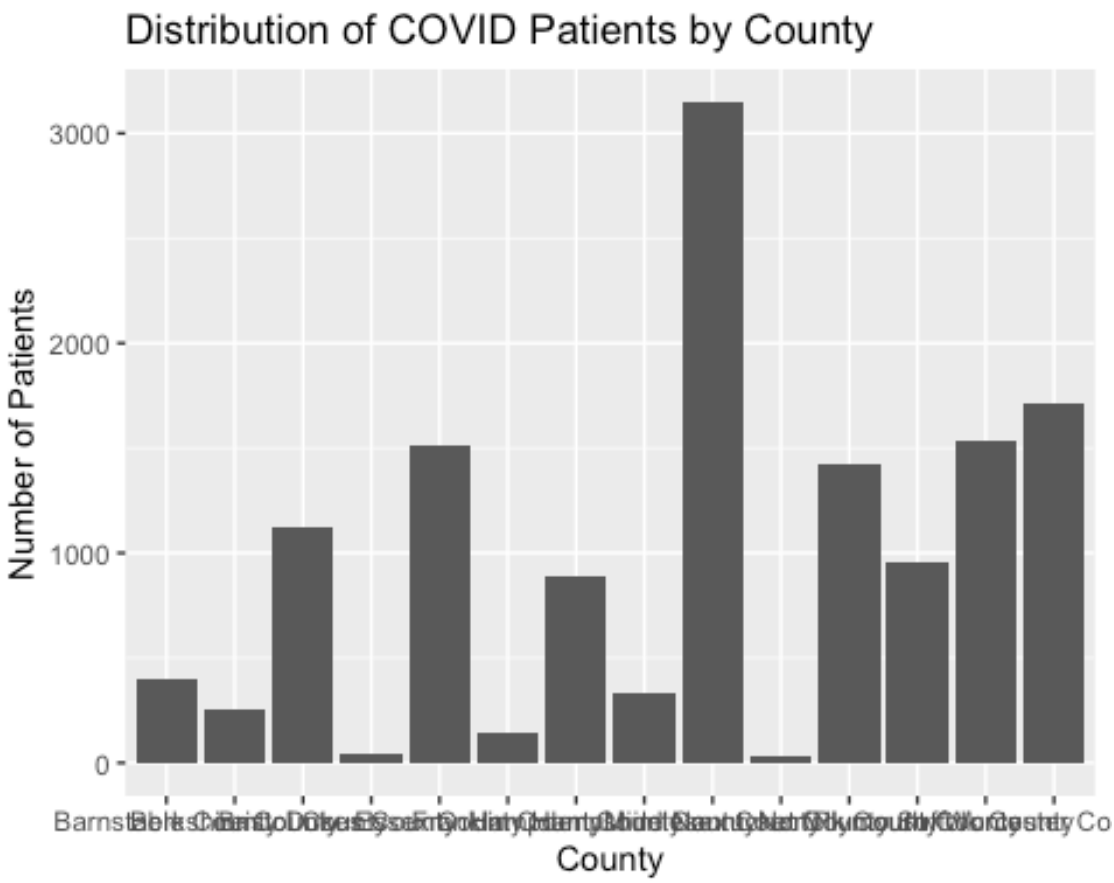
```
library(readr)
library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ lubridate 1.9.3     ✔ tibble    3.2.1
## ✔ purrr     1.0.2     ✔ tidyr     1.3.1

## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force
all conflicts to become errors

library(stats)

encountersUG <- read_csv("encountersUG.csv")

## New names:
## Rows: 238327 Columns: 13
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: ","
chr
## (5): Id, PATIENT, ENCOUNTERCLASS, DESCRIPTION, REASONDESCRIPTION dbl (6):
...1,
## CODE, BASE_ENCOUNTER_COST, TOTAL_CLAIM_COST, PAYER_COVERAGE,... dttm (2):
## START, STOP
## 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.
## • `` -> `...1`

conditionsUG <- read_csv("conditionsUG.csv")

## New names:
## Rows: 86215 Columns: 7
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: ","
chr
## (3): PATIENT, ENCOUNTER, DESCRIPTION dbl (2): ...1, CODE date (2): START,
STOP
## 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.
## • `` -> `...1`

patientsUG <- read_csv("patientsUG.csv")

## New names:
## Rows: 9264 Columns: 11
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: ","
chr
## (7): Id, MARITAL, RACE, GENDER, CITY, STATE, COUNTY dbl (2): ...1, ZIP
date
## (2): BIRTHDATE, DEATHDATE
## 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.
## • `` -> `...1`
```

1. Write the code to analyse the distribution of COVID patients (confirmed or suspected) across counties. Write the code to investigate the distribution of the patients across age groups (e.g., 0-18, 19-35, 36-50, 51+). Visualise both the findings using the histogram. Explain your findings.

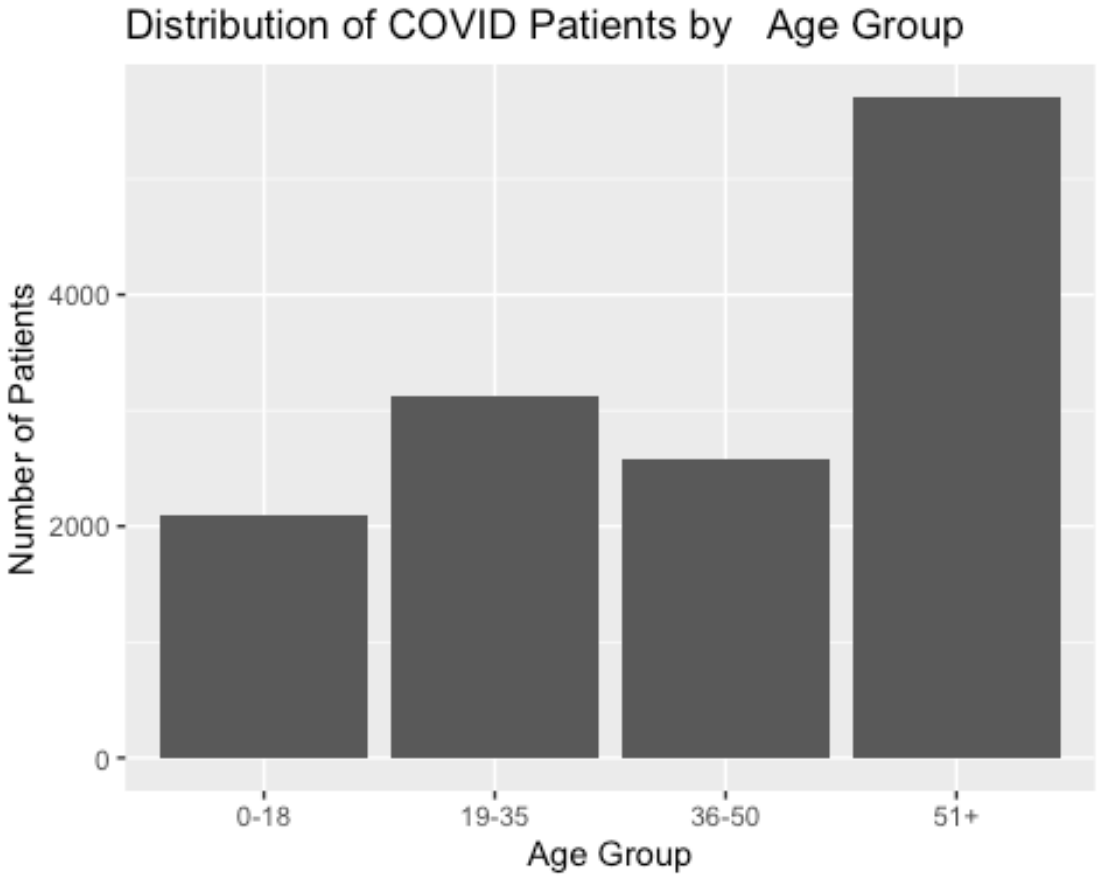
```
#Filter confirm or suspected Covid patients only
covid_condition <- conditionsUG %>%
  filter(CODE %in% c("840544004", "840539006"))

#Make sure data is valid
view(covid_condition)

#Merge covid condition with patient
covid_data <- covid_condition %>%
  left_join(patientsUG, by = c("PATIENT" = "Id"))
```

```r
#Make sure data is valid
view(covid_data)

#Analyse distribution accross countries
county_dist <- covid_data %>%
  count(COUNTY) %>%
  ggplot(aes(x = COUNTY, y = n)) +
  geom_histogram(bins = 4, stat = "identity") +
  labs(title = "Distribution of COVID Patients by County", x = "County", y =
"Number of Patients")

## Warning in geom_histogram(bins = 4, stat = "identity"): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`

county_dist
```




```r
#Calculate age based on BirthDate (assuming DeathDate is empty for living
patients)
covid_data <- covid_data %>%
  mutate(Age = as.integer((Sys.Date() - BIRTHDATE) / 365.25)) %>%  #Adjust
for leap years
  mutate(AgeGroup = case_when(
    Age < 19 ~ "0-18",
    Age < 36 ~ "19-35",
```

```
    Age < 51 ~ "36-50",
    TRUE ~ "51+"
  ))

#Create distribution graph
age_dist <- covid_data %>%
  count(AgeGroup) %>%
  ggplot(aes(x = AgeGroup, y = n)) +
  geom_histogram(bins = 4, stat = "identity") +
  labs(title = "Distribution of COVID Patients by   Age Group", x = "Age
Group", y = "Number of Patients")

## Warning in geom_histogram(bins = 4, stat = "identity"): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`

age_dist
```



=> from the findings above, it can be seen that most COVID patients come from Middlesex County Most patients age 51+ while the rest are almost evenly distributed to other age ranges. Patients age 0-18 have the least chance of having or suspected having COVID.

2. Filter those patients in the dataset that have contracted COVID-19 or Suspected COVID-19; ; what are the top 10 most common conditions (symptoms) related to the patients? Do the conditions differ between genders? Provide a table to rank the top 10 conditions for male and female patients separately. Elaborate on the findings.

```
#Merge covid patient with covid condition and encounter symptom from dataset
found from previous question and count top 10
covid_symptom_top10 <- inner_join(covid_data %>% select(PATIENT),
encountersUG %>% select(PATIENT, DESCRIPTION), by = "PATIENT") %>% #select
column PATIENT from both table and DESCRIPTION from encounterUG only
  count(DESCRIPTION) %>% #count number of DESRIPTION
  arrange(desc(n)) %>%
  head(10)  #Top 10 conditions
```

```
## Warning in inner_join(covid_data %>% select(PATIENT), encountersUG %>%
select(PATIENT, : Detected an unexpected many-to-many relationship between
`x` and `y`.
## ℹ Row 1 of `x` matches multiple rows in `y`.
## ℹ Row 1 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
print(covid_symptom_top10)
```

```
## # A tibble: 10 × 2
##    DESCRIPTION                                      n
##    <chr>                                        <int>
##  1 General examination of patient (procedure) 134592
##  2 Encounter for check up (procedure)          42579
##  3 Follow-up encounter                         22733
##  4 Well child visit (procedure)                14706
##  5 Encounter for symptom                       14243
##  6 Encounter for symptom (procedure)           13688
##  7 Encounter for problem                        9651
##  8 Encounter for problem (procedure)            7217
##  9 Prenatal initial visit                       4954
## 10 Prenatal visit                               4395
```

```
#Find top 10 conditions base on gender from dataset found from previous
question
covid_symptom_gender <- inner_join(covid_data %>% select(PATIENT, GENDER),
encountersUG %>% select(PATIENT, DESCRIPTION), by = "PATIENT") %>%
  count(DESCRIPTION, GENDER) %>% #Count both DESCRIPTION and GENDER
  arrange(GENDER, desc(n)) %>% #Arrange and group top 10 DESCRIPTION
  group_by(GENDER) %>% #Group GENDER separately
  top_n(10)
```

```
## Warning in inner_join(covid_data %>% select(PATIENT, GENDER), encountersUG
%>% : Detected an unexpected many-to-many relationship between `x` and `y`.
## ℹ Row 1 of `x` matches multiple rows in `y`.
```

```
## ℹ Row 1 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.

## Selecting by n

#Create table to rank top 10 M and F
covid_symptom_gender_top10 <- covid_symptom_gender %>%
  pivot_wider(names_from = GENDER, values_from = n) %>%
  arrange(desc(M), desc(F))

print(covid_symptom_gender_top10)

## # A tibble: 13 × 3
##    DESCRIPTION                                     F     M
##    <chr>                                       <int> <int>
##  1 General examination of patient (procedure) 68272 66320
##  2 Encounter for check up (procedure)         16003 26576
##  3 Follow-up encounter                        11161 11572
##  4 Encounter for symptom                       7354  6889
##  5 Well child visit (procedure)                7897  6809
##  6 Encounter for symptom (procedure)           7202  6486
##  7 Encounter for problem (procedure)             NA  4730
##  8 Encounter for problem                       4936  4715
##  9 Urgent care clinic (procedure)                NA  2040
## 10 Outpatient procedure                          NA  1682
## 11 Prenatal initial visit                      4954    NA
## 12 Prenatal visit                              4395    NA
## 13 Consultation for treatment                  3570    NA
```

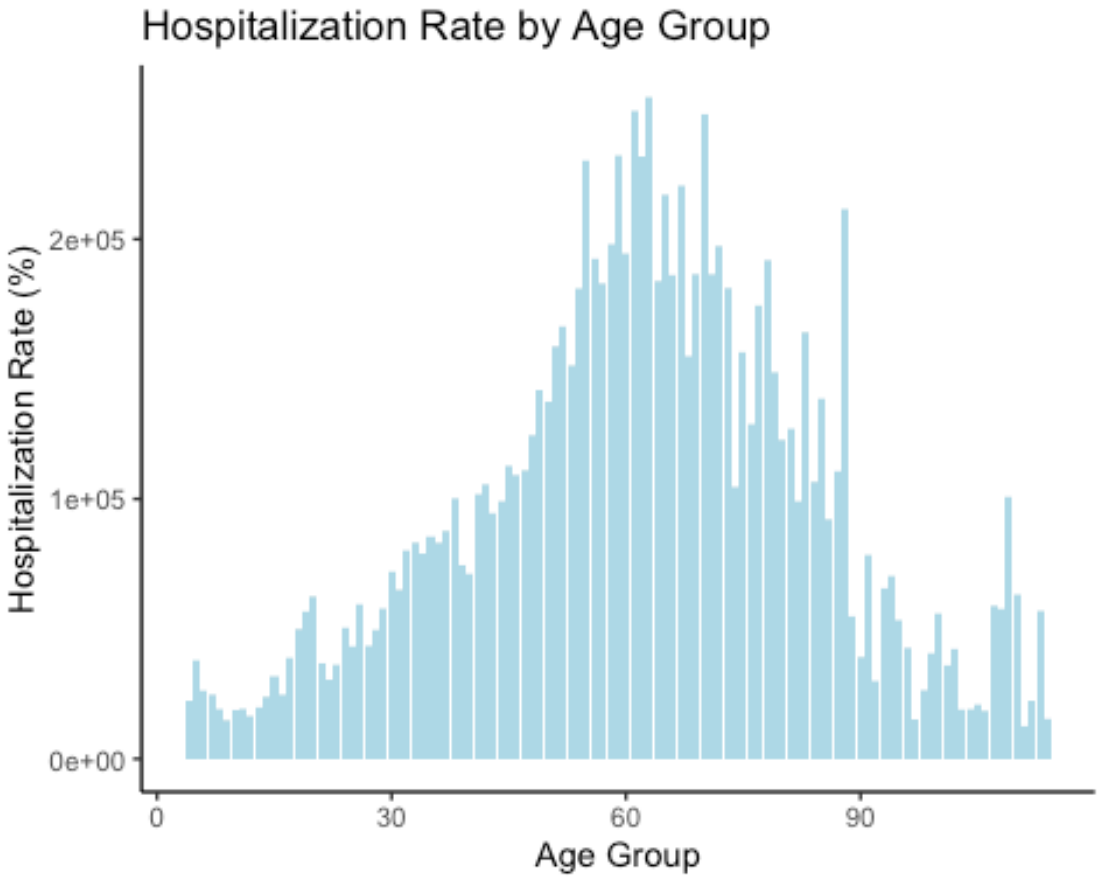
=> From the finding above, it can be seen that most contracted COVID or suspected COVID cased are discovered through General examination or Encounter for check up. Although most appeared symptoms are similar between Male and Female, there are still differences between symtoms.

3. Write the code to analyse the factors that might influence the hospitalisation rate (ambulatory, emergency, inpatient, urgent care) for the COVID patient (confirmed or suspected) in the dataset. Any factors in the dataset, such as age, gender, zip code, marital status, race and county, can be considered. Pick 2 of the factors and explain if there is a trend that explains the variation.

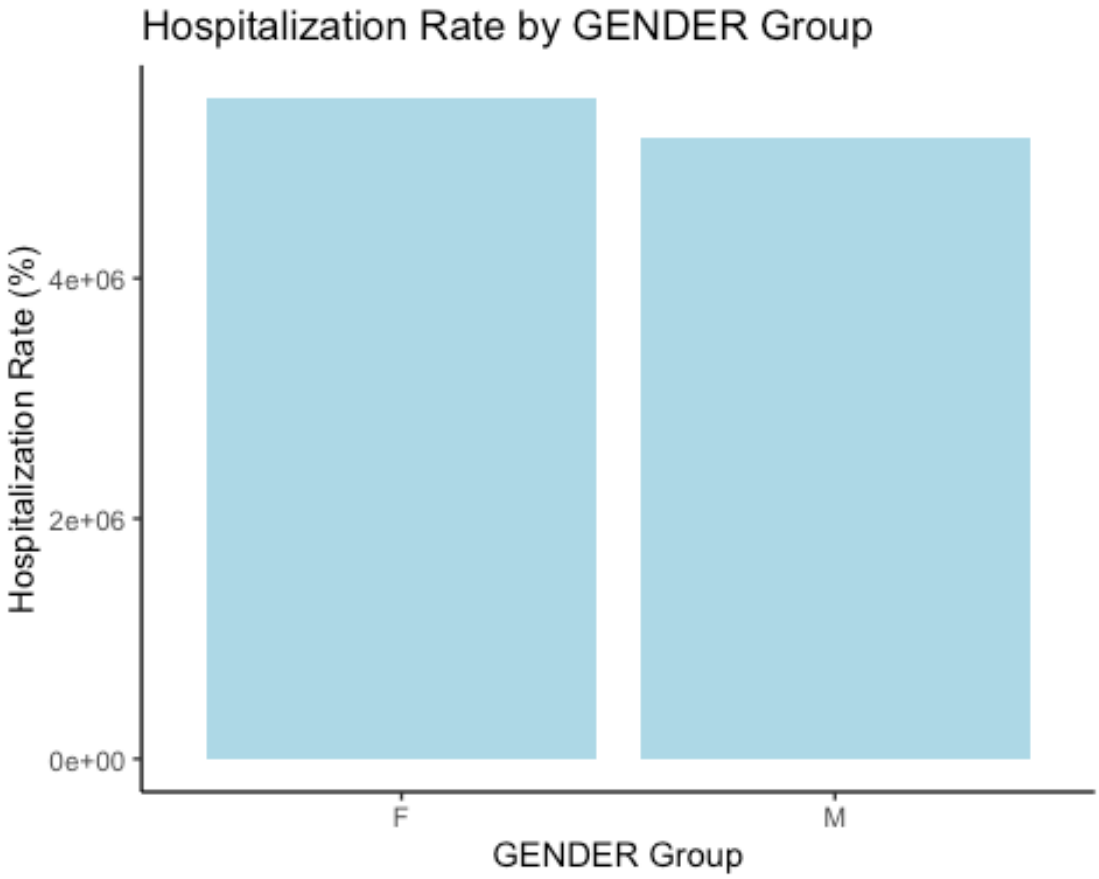
```
#Merge covid_data with encounter class column from encountersUG
covid_encounter_class <- inner_join(covid_data %>% select(PATIENT, GENDER,
Age, ZIP, COUNTY), encountersUG %>% select(PATIENT, ENCOUNTERCLASS), by =
"PATIENT")

## Warning in inner_join(covid_data %>% select(PATIENT, GENDER, Age, ZIP,
COUNTY), : Detected an unexpected many-to-many relationship between `x` and
`y`.
## ℹ Row 1 of `x` matches multiple rows in `y`.
```

```
## ℹ Row 1 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```r
#Create hospitalization rate by category
hospitalisation_rate <- covid_encounter_class %>%
  group_by(ENCOUNTERCLASS) %>%
  summarise(n = n()) %>%
  mutate(rate = n / nrow(covid_encounter_class) * 100)  #Convert count to
percentage

#Hospitalization rate by Age Group
age_group <- covid_encounter_class %>%
  group_by(Age) %>%
  left_join(hospitalisation_rate, by = "ENCOUNTERCLASS") %>%
  ggplot(aes(x = Age, y = rate)) +
  geom_col(fill = "lightblue") +
  labs(title = "Hospitalization Rate by Age Group", x = "Age Group", y =
"Hospitalization Rate (%)") +
  theme_classic()

print(age_group)
```



```r
#Hospitalization rate by GENDER Group
gender_group <- covid_encounter_class %>%
  group_by(GENDER) %>%
  left_join(hospitalisation_rate, by = "ENCOUNTERCLASS") %>%
  ggplot(aes(x = GENDER, y = rate)) +
  geom_col(fill = "lightblue") +
  labs(title = "Hospitalization Rate by GENDER Group", x = "GENDER Group", y
= "Hospitalization Rate (%)") +
  theme_classic()

print(gender_group)
```

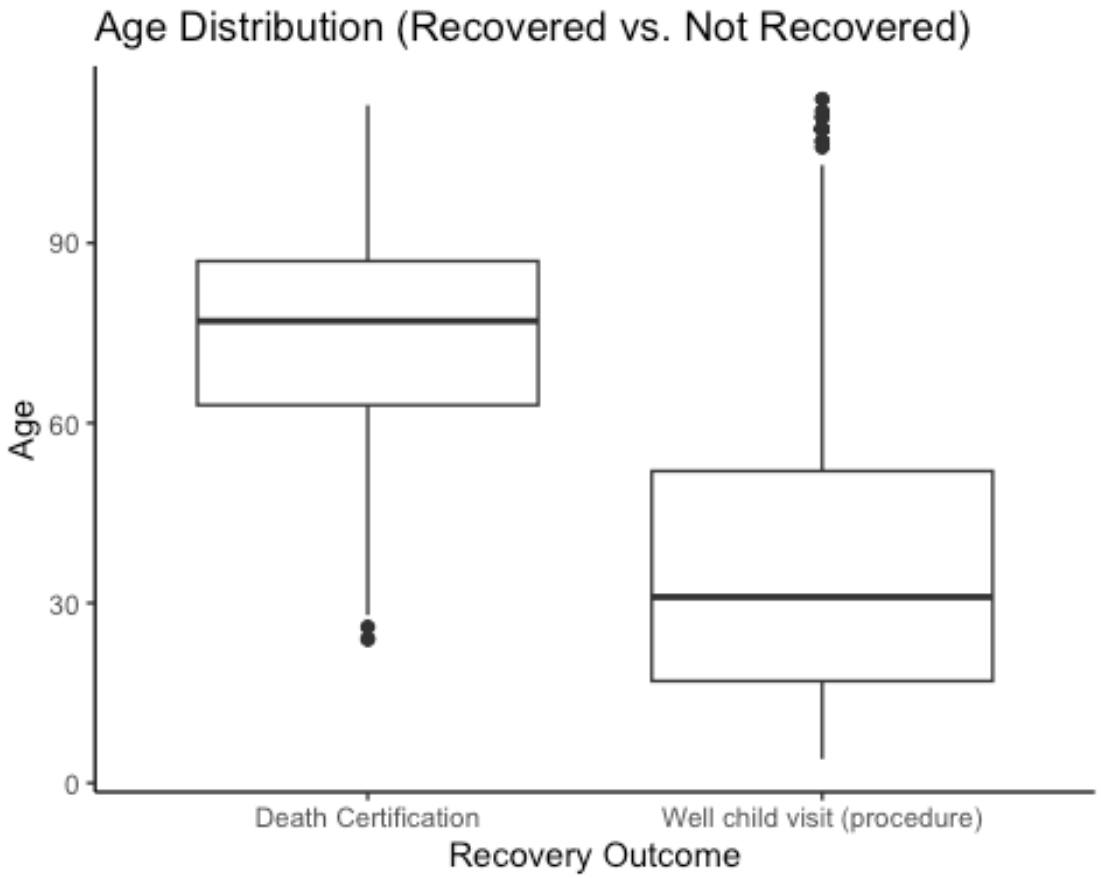


=> From the findings above, it can be seen that hospitalisation rate increase as patients turn into their 60s. Although there are not a big gap, Female has slightly greater hospitalisation rate compare to Male.

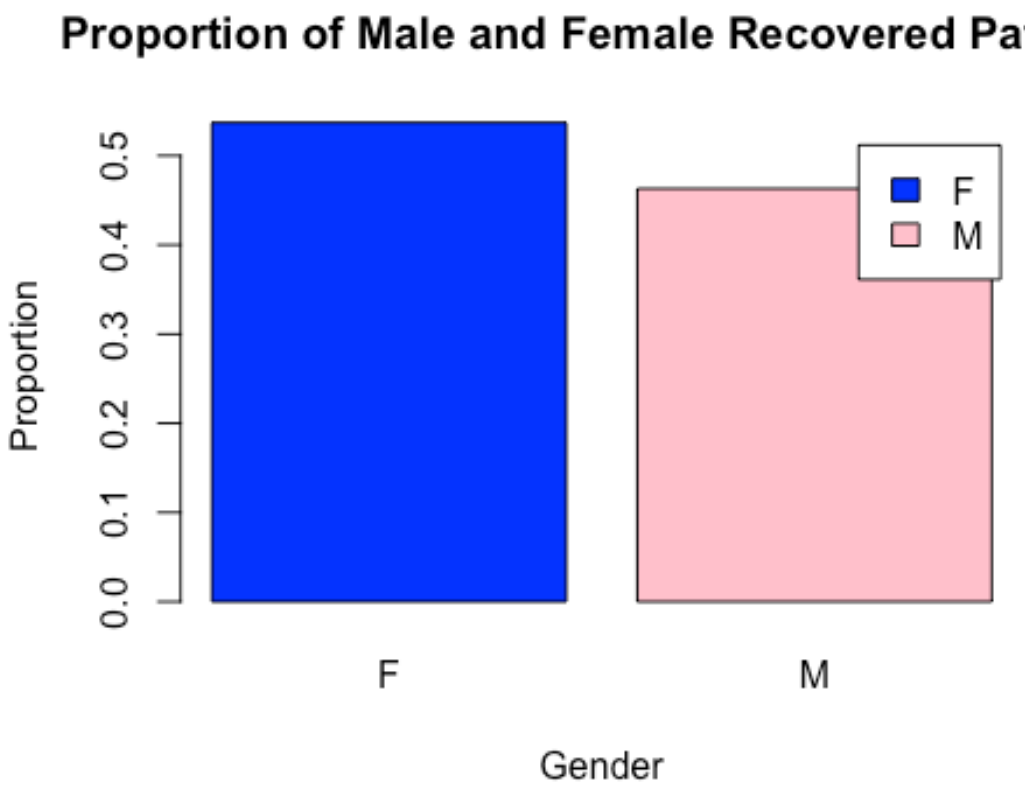
4. Write the code to investigate the characteristics of patients (confirmed or suspected) who recover from COVID-19 compared to those who don't. Consider factors such as demographics (age, gender, zip code), symptoms, and timeline of diagnosis and recovery. Analyse how these factors impact the recovery outcome.

```r
#Create table then filter recovered and non-recovered groups (DESCRIPTION:
Death Certification)
```

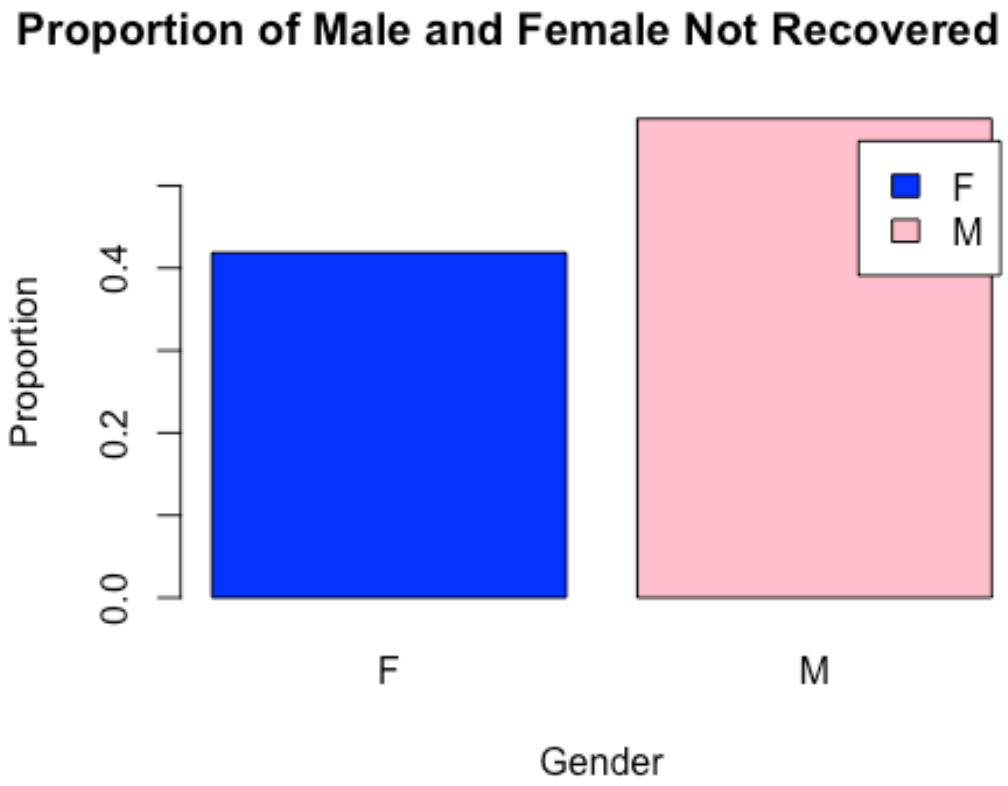
```r
covid_recover <- inner_join(covid_data %>% select(PATIENT, START, STOP,
GENDER, Age, ZIP), encountersUG %>% select(PATIENT, DESCRIPTION), by =
"PATIENT")

## Warning in inner_join(covid_data %>% select(PATIENT, START, STOP, GENDER,
: Detected an unexpected many-to-many relationship between `x` and `y`.
## ℹ Row 1 of `x` matches multiple rows in `y`.
## ℹ Row 1 of `y` matches multiple rows in `x`.
## ℹ If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.

not_recovered <- covid_recover %>%
  filter(DESCRIPTION == "Death Certification")

recovered <- covid_recover %>%
  filter(DESCRIPTION == "Well child visit (procedure)")

#Combine demographics
combined_data <- rbind(recovered, not_recovered)

#Age distribution by recovery outcome
ggplot(combined_data, aes(x = DESCRIPTION, y = Age)) +
  geom_boxplot() +
  labs(title = "Age Distribution (Recovered vs. Not Recovered)", x =
"Recovery Outcome", y = "Age") +
  theme_classic()
```




```r
#Gender proportion for recovered patient
recovered_proportion <- table(recovered$GENDER) / nrow(recovered)

#Create bar plot for found proportion
print(barplot(recovered_proportion, main = "Proportion of Male and Female
Recovered Patients", xlab = "Gender", ylab = "Proportion", col = c("blue",
"pink"), legend.text = rownames(recovered_proportion)))
```




```
##      [,1]
## [1,]  0.7
## [2,]  1.9

#Gender proportion for not_recover patient
not_recover_proportion <- table(not_recovered$GENDER) / nrow(not_recovered)

#Create bar plot for found proportion
print(barplot(not_recover_proportion, main = "Proportion of Male and Female
Not Recovered Patients", xlab = "Gender", ylab = "Proportion", col =
c("blue", "pink"), legend.text = rownames(not_recover_proportion)))
```




```
##      [,1]
## [1,]  0.7
## [2,]  1.9
```

=> From above findings, it can be seen that younger patients have higher chance of recovering from COVID compare to older age patients. Male patients also have higher chances of being recovered from COVID.