DATA 622: Machine Learning and Big Data: HW4 (Final Project)

Gabriel Campos

Last edited May 20, 2024

Assigment Description

Exploratory analysis and essay

Assignment

1. Choose a dataset You get to decide which dataset you want to work on. The data set must be different from the ones used in previous homeworks You can work on a problem from your job, or something you are interested in. You may also obtain a dataset from sites such as Kaggle, Data.Gov, Census Bureau, USGS or other open data portals.
2. Select one of the methodologies studied in weeks 1-10, and another methodology from weeks 11-15 to apply in the new dataset selected.
3. To complete this task:.
   a. Describe the problem you are trying to solve.
   b. Describe your datasets and what you did to prepare the data for analysis.
   c. Methodologies you used for analyzing the data
   d. What's the purpose of the analysis performed
   e. Make your conclusions from your analysis. Please be sure to address the business impact (it could be of any domain) of your solution.

Deliverable

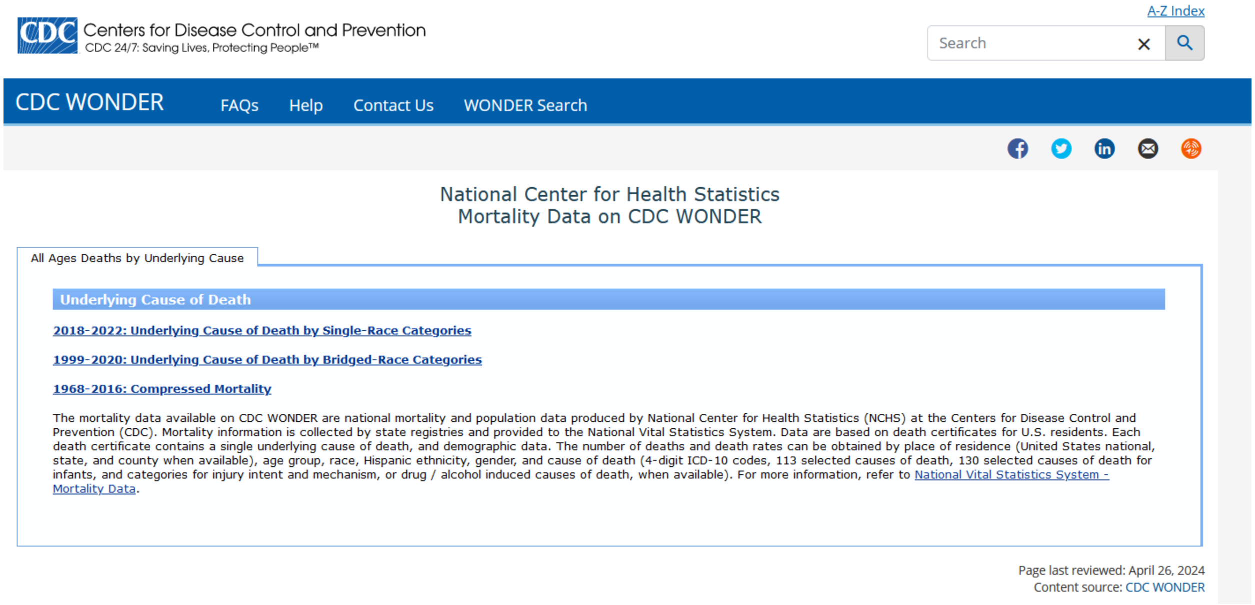
1. Your final presentation (essay or video) should include:
   1. The traditional R file or Python file and essay,
   2. An Essay (minimum 500 word document) or Video (5 to 8 minutes recording) Include the execution and explanation of your code. The video can be recorded on any platform of your choice (Youtube, Free Cam).

Libraries

```r
library(Amelia)
library(car)
library(caret)
library(corrplot)
library(Cubist)
library(DataExplorer)
library(dplyr)
library(e1071)
library(earth)
library(forcats)
library(forecast)
library(fpp3)
library(gbm)
library(ggplot2)
library(ggforce)
library(gridExtra)
library(kableExtra)
library(MASS)
library(Metrics)
library(mice)
library(mlbench)
library(party)
library(psych)
library(pROC)
library(randomForest)
library(RANN)
library(RColorBrewer)
library(readr)
library(readxl)
library(rpart)
library(rpart.plot)
library(stringr)
library(summarytools)
library(tidyr)
library(tidymodels)
library(VIM)
library(earth)
library(randomForest)
```

Overview

Data retrived came from Centers of Disease Control and Prevention website. The data selected was the 2018-2022 Underlying Cause of Deaths by Single: Race Categories. The queries completed from this site is limited to 75,000 observations. In order to limit the data, State was filtered to strictly NYS and data was collected in yearly batches then merged.


CDC Centers for Disease Control and Prevention
CDC 24/7: Saving Lives, Protecting People™
A-Z Index
Search
CDC WONDER
FAQs
Help
Contact Us
WONDER Search
All Ages Deaths by Underlying Cause

National Center for Health Statistics
Mortality Data on CDC WONDER

Underlying Cause of Death

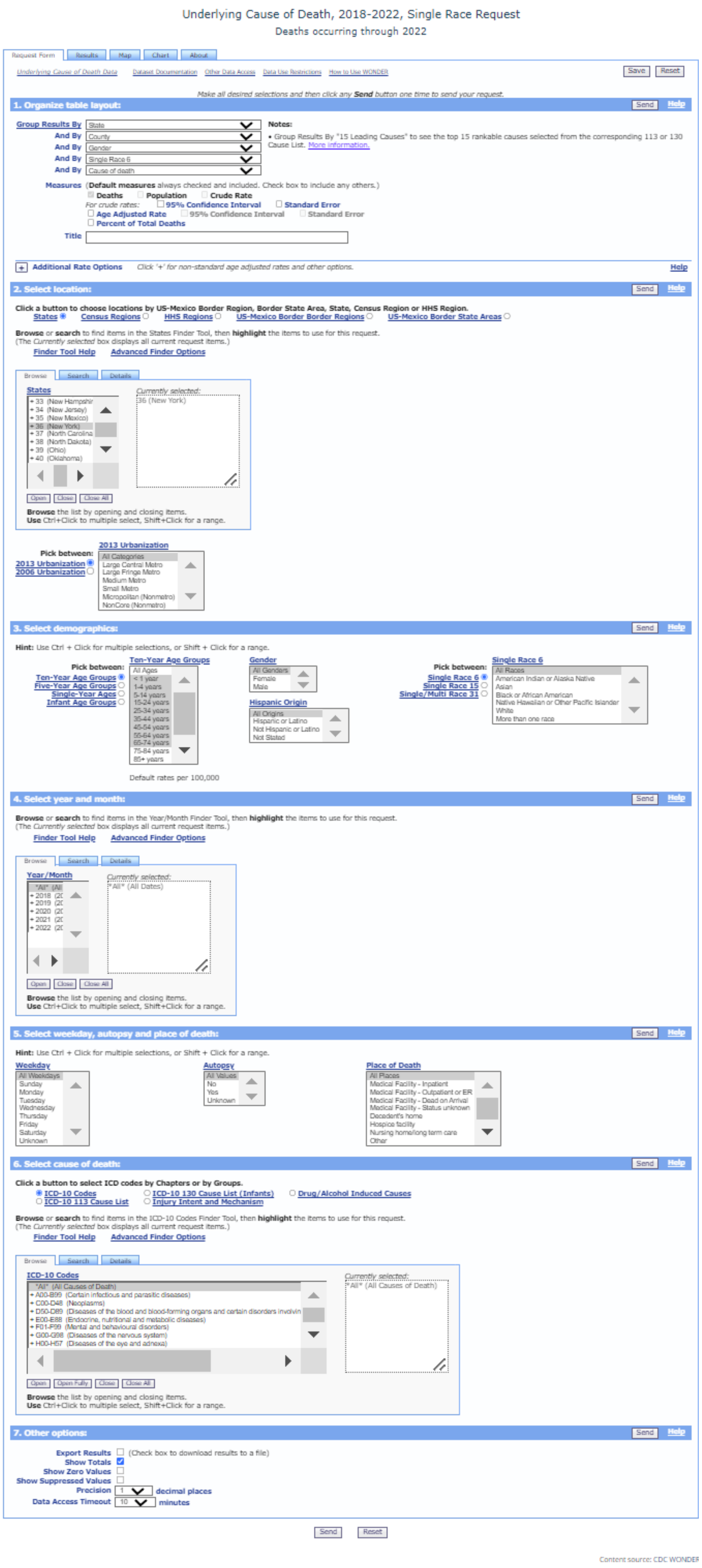
2018-2022: Underlying Cause of Death by Single-Race Categories

1999-2020: Underlying Cause of Death by Bridged-Race Categories

1968-2016: Compressed Mortality

The mortality data available on CDC WONDER are national mortality and population data produced by National Center for Health Statistics (NCHS) at the Centers for Disease Control and Prevention (CDC). Mortality information is collected by state registries and provided to the National Vital Statistics System. Data are based on death certificates for U.S. residents. Each death certificate contains a single underlying cause of death, and demographic data. The number of deaths and death rates can be obtained by place of residence (United States national, state, and county when available), age group, race, Hispanic ethnicity, gender, and cause of death (4-digit ICD-10 codes, 113 selected causes of death, 130 selected causes of death for infants, and categories for injury intent and mechanism, or drug / alcohol induced causes of death, when available). For more information, refer to National Vital Statistics System - Mortality Data.

Page last reviewed: April 26, 2024
Content source: CDC WONDER



Underlying Cause of Death, 2018-2022, Single Race Request

Deaths occurring through 2022

Underlying Cause of Death Data Dataset Documentation Other Data Access Data Use Restrictions How to Use WONDER

Make all desired selections and then click any ***Send*** *button one time to send your request.*

1. Organize table layout:

Group Results By: State
And By: County
And By: Gender
And By: Single Race 6
And By: Cause of death

Notes:

- Group Results By "15 Leading Causes" to see the top 15 rankable causes selected from the corresponding 113 or 130 Cause List. More information.

Measures (**Default measures** always checked and included. Check box to include any others.)
☑ **Deaths** □ **Population** □ **Crude Rate**
For crude rates: □ **95% Confidence Interval** □ **Standard Error**
□ **Age Adjusted Rate** □ **95% Confidence Interval** □ **Standard Error**
□ **Percent of Total Deaths**

Title:

[+] Additional Rate Options *Click '+' for non-standard age adjusted rates and other options.* Help

2. Select location:

Click a button to choose locations by US-Mexico Border Region, Border State Area, State, Census Region or HHS Region.
States ● Census Regions ○ HHS Regions ○ US-Mexico Border Border Regions ○ US-Mexico Border State Areas ○

Browse or **search** to find items in the States Finder Tool, then **highlight** the items to use for this request.
(The *Currently selected* box displays all current request items.)
Finder Tool Help Advanced Finder Options

Browse | Search | Details

States
+ 33 (New Hampshir
+ 34 (New Jersey)
+ 35 (New Mexico)
+ 36 (New York)
+ 37 (North Carolina
+ 38 (North Dakota)
+ 39 (Ohio)
+ 40 (Oklahoma)

Currently selected:
36 (New York)

Open | Close | Close All

Browse the list by opening and closing items.
Use Ctrl+Click to multiple select, Shift+Click for a range.

Pick between:
2013 Urbanization ●
2006 Urbanization ○

2013 Urbanization
All Categories
Large Central Metro
Large Fringe Metro
Medium Metro
Small Metro
Micropolitan (Nonmetro)
NonCore (Nonmetro)

3. Select demographics:

Hint: Use Ctrl + Click for multiple selections, or Shift + Click for a range.

Pick between:
Ten-Year Age Groups ●
Five-Year Age Groups ○
Single-Year Ages ○
Infant Age Groups ○

Ten-Year Age Groups
All Ages
< 1 year
1-4 years
5-14 years
15-24 years
25-34 years
35-44 years
45-54 years
55-64 years
65-74 years
75-84 years
85+ years

Default rates per 100,000

Gender
All Genders
Female
Male

Hispanic Origin
All Origins
Hispanic or Latino
Not Hispanic or Latino
Not Stated

Pick between:
Single Race 6 ●
Single Race 15 ○
Single/Multi Race 31 ○

Single Race 6
All Races
American Indian or Alaska Native
Asian
Black or African American
Native Hawaiian or Other Pacific Islander
White
More than one race

4. Select year and month:

Browse or **search** to find items in the Year/Month Finder Tool, then **highlight** the items to use for this request.
(The *Currently selected* box displays all current request items.)
Finder Tool Help Advanced Finder Options

Browse | Search | Details

Year/Month
"All" (All
+ 2018 (20
+ 2019 (20
+ 2020 (20
+ 2021 (20
+ 2022 (20

Currently selected:
All (All Dates)

Open | Close | Close All

Browse the list by opening and closing items.
Use Ctrl+Click to multiple select, Shift+Click for a range.

5. Select weekday, autopsy and place of death:

Hint: Use Ctrl + Click for multiple selections, or Shift + Click for a range.

Weekday
All Weekdays
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Unknown

Autopsy
All Values
No
Yes
Unknown

Place of Death
All Places
Medical Facility - Inpatient
Medical Facility - Outpatient or ER
Medical Facility - Dead on Arrival
Medical Facility - Status unknown
Decedent's home
Hospice facility
Nursing home/long term care
Other

6. Select cause of death:

Click a button to select ICD codes by Chapters or by Groups.
● ICD-10 Codes ○ ICD-10 130 Cause List (Infants) ○ Drug/Alcohol Induced Causes
○ ICD-10 113 Cause List ○ Injury Intent and Mechanism

Browse or **search** to find items in the ICD-10 Codes Finder Tool, then **highlight** the items to use for this request.
(The *Currently selected* box displays all current request items.)
Finder Tool Help Advanced Finder Options

Browse | Search | Details

ICD-10 Codes
"All" (All Causes of Death)
+ A00-B99 (Certain infectious and parasitic diseases)
+ C00-D48 (Neoplasms)
+ D50-D89 (Diseases of the blood and blood-forming organs and certain disorders involvin
+ E00-E88 (Endocrine, nutritional and metabolic diseases)
+ F01-F99 (Mental and behavioural disorders)
+ G00-G98 (Diseases of the nervous system)
+ H00-H57 (Diseases of the eye and adnexa)

Currently selected:
All (All Causes of Death)

Open | Open Fully | Close | Close All

Browse the list by opening and closing items.
Use Ctrl+Click to multiple select, Shift+Click for a range.

7. Other options:

Export Results □ (Check box to download results to a file)
Show Totals ☑
Show Zero Values □
Show Suppressed Values □
Precision 1 decimal places
Data Access Timeout 10 minutes

Send | Reset

Content source: CDC WONDER

Load Data

We will first load in the data that is required for this analysis.

```
cdc_ucd_df_2018 <- as_tibble(read_tsv(url_git_2018,
                                      show_col_types = FALSE)
                             )%>%
                               dplyr::select(-1)%>%
                               rename(Race = `Single Race 6`,
                                      `Race Code` = `Single Race 6 Code`)

cdc_ucd_df_2019 <- as_tibble(read_tsv(url_git_2019,
                                      show_col_types = FALSE)
                             )%>%
                               dplyr::select(-1)%>%
                               rename(Race = `Single Race 6`,
                                      `Race Code` = `Single Race 6 Code`)

cdc_ucd_df_2020 <- as_tibble(read_tsv(url_git_2020,
                                      show_col_types = FALSE)
                             )%>%
                               dplyr::select(-1)%>%
                               rename(Race = `Single Race 6`,
                                      `Race Code` = `Single Race 6 Code`)

cdc_ucd_df_2021 <- as_tibble(read_tsv(url_git_2021,
                                      show_col_types = FALSE)
                             )%>%
                               dplyr::select(-1)%>%
                               rename(Race = `Single Race 6`,
                                      `Race Code` = `Single Race 6 Code`)

cdc_ucd_df_2022 <- as_tibble(read_tsv(url_git_2022,
                                      show_col_types = FALSE)
                             )%>%
                               dplyr::select(-1)%>%
                               rename(Race = `Single Race 6`,
                                      `Race Code` = `Single Race 6 Code`)

cdc_ucd_df <- bind_rows(cdc_ucd_df_2018,
                        cdc_ucd_df_2020,
                        cdc_ucd_df_2021,
                        cdc_ucd_df_2022)
```

Exporatory Analysis (EDA)

First, we can preview our dataset.

```
glimpse(cdc_ucd_df)
```

```
## Rows: 12,356
## Columns: 13
## $ Year                  <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, ~
## $ `Year Code`           <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, ~
## $ County                <chr> "Albany County, NY", "Albany County, NY", "Alban~
## $ `County Code`         <dbl> 36001, 36001, 36001, 36001, 36001, 36001, 36001,~
## $ Gender                <chr> "Female", "Female", "Female", "Female", "Female"~
## $ `Gender Code`         <chr> "F", "F", "F", "F", "F", "F", "F", "F", "F", "F"~
## $ Race                  <chr> "White", "White", "White", "White", "White", "Wh~
## $ `Race Code`           <chr> "2106-3", "2106-3", "2106-3", "2106-3", "2106-3"~
## $ `Cause of death`      <chr> "Septicaemia, unspecified", "Colon, unspecified ~
## $ `Cause of death Code` <chr> "A41.9", "C18.9", "C25.9", "C34.9", "C50.9", "C5~
## $ Deaths                <dbl> 19, 10, 24, 78, 37, 16, 25, 18, 143, 14, 73, 21,~
## $ Population            <dbl> 119942, 119942, 119942, 119942, 119942, 119942, ~
## $ `Crude Rate`          <chr> "Unreliable", "Unreliable", "20.0", "65.0", "30.~
```

The dataset consists of 12,356 rows and 13 columns. Most of the variables are categorical, except for the "Deaths" column indicating the count for this type of observation.

We can take also take a look at the summary statistics for each of the numeric variables.

```
describe(cdc_ucd_df)
```

```
##                      vars     n      mean        sd median   trimmed       mad
## Year                    1 12131   2020.31      1.45   2021   2020.39      1.48
## Year Code               2 12131   2020.31      1.45   2021   2020.39      1.48
## County*                 3 12131     30.34     16.05     30     30.48     16.31
## County Code             4 12131  36061.18     32.78  36061  36061.57     32.62
## Gender*                 5 12131      1.51      0.50      2      1.51      0.00
## Gender Code*            6 12131      1.51      0.50      2      1.51      0.00
## Race*                   7 12131      3.56      0.90      4      3.76      0.00
## Race Code*              8 12131      1.89      0.43      2      1.93      0.00
## Cause of death*         9 12131     84.52     59.69     70     81.94     68.20
## Cause of death Code*   10 12131     88.22     50.97     86     86.84     66.72
## Deaths                 11 12131     38.16     79.76     19     23.98     11.86
## Population             12 12131 278183.67 206485.84 280155 264969.96 297173.83
## Crude Rate*            13 12131    908.29    378.14   1195    969.61      0.00
##                        min    max  range  skew kurtosis      se
## Year                  2018   2022      4 -0.50    -1.04    0.01
## Year Code             2018   2022      4 -0.50    -1.04    0.01
## County*                  1     61     60 -0.05    -0.79    0.15
## County Code          36001  36123    122 -0.09    -0.80    0.30
## Gender*                  1      2      1 -0.03    -2.00    0.00
## Gender Code*             1      2      1 -0.03    -2.00    0.00
## Race*                    1      4      3 -1.70     1.23    0.01
## Race Code*               1      4      3 -0.57     1.82    0.00
## Cause of death*          1    190    189  0.35    -1.34    0.54
## Cause of death Code*     1    190    189  0.17    -1.00    0.46
## Deaths                  10   2341   2331 10.75   176.82    0.72
## Population            4648 668250 663602  0.37    -1.23 1874.74
## Crude Rate*              1   1195   1194 -0.97    -0.49    3.43
```

Year is notable being calculated due to its formatting, which will need addressing.Deaths, have a noteable average of 38.16 but a standard deviation of 79.76, indicating a large window of fluctuating deaths.

```
summary(cdc_ucd_df)
```

```
##       Year        Year Code       County           County Code
##  Min.   :2018   Min.   :2018   Length:12356       Min.   :36001
##  1st Qu.:2020   1st Qu.:2020   Class :character   1st Qu.:36039
##  Median :2021   Median :2021   Mode  :character   Median :36061
##  Mean   :2020   Mean   :2020                      Mean   :36061
##  3rd Qu.:2022   3rd Qu.:2022                      3rd Qu.:36083
##  Max.   :2022   Max.   :2022                      Max.   :36123
##  NA's   :225    NA's   :225                       NA's   :225
##     Gender           Gender Code            Race            Race Code
##  Length:12356       Length:12356       Length:12356       Length:12356
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Cause of death     Cause of death Code     Deaths          Population
##  Length:12356       Length:12356        Min.   :  10.00   Min.   :  4648
##  Class :character   Class :character    1st Qu.:  13.00   1st Qu.: 83932
##  Mode  :character   Mode  :character    Median :  19.00   Median :280155
##                                         Mean   :  38.16   Mean   :278184
##                                         3rd Qu.:  35.00   3rd Qu.:485306
##                                         Max.   :2341.00   Max.   :668250
##                                         NA's   :225       NA's   :225
##   Crude Rate
##  Length:12356
##  Class :character
##  Mode  :character
##
##
##
##
```

```
apply(cdc_ucd_df, 2, function(x) sum(is.na(x)))
```

```
##                 Year           Year Code              County         County Code
##                  225                 225                 225                 225
##               Gender         Gender Code                Race           Race Code
##                  225                 225                 225                 225
##       Cause of death Cause of death Code              Deaths          Population
##                  225                 225                 225                 225
##           Crude Rate
##                  225
```

NA Proportions

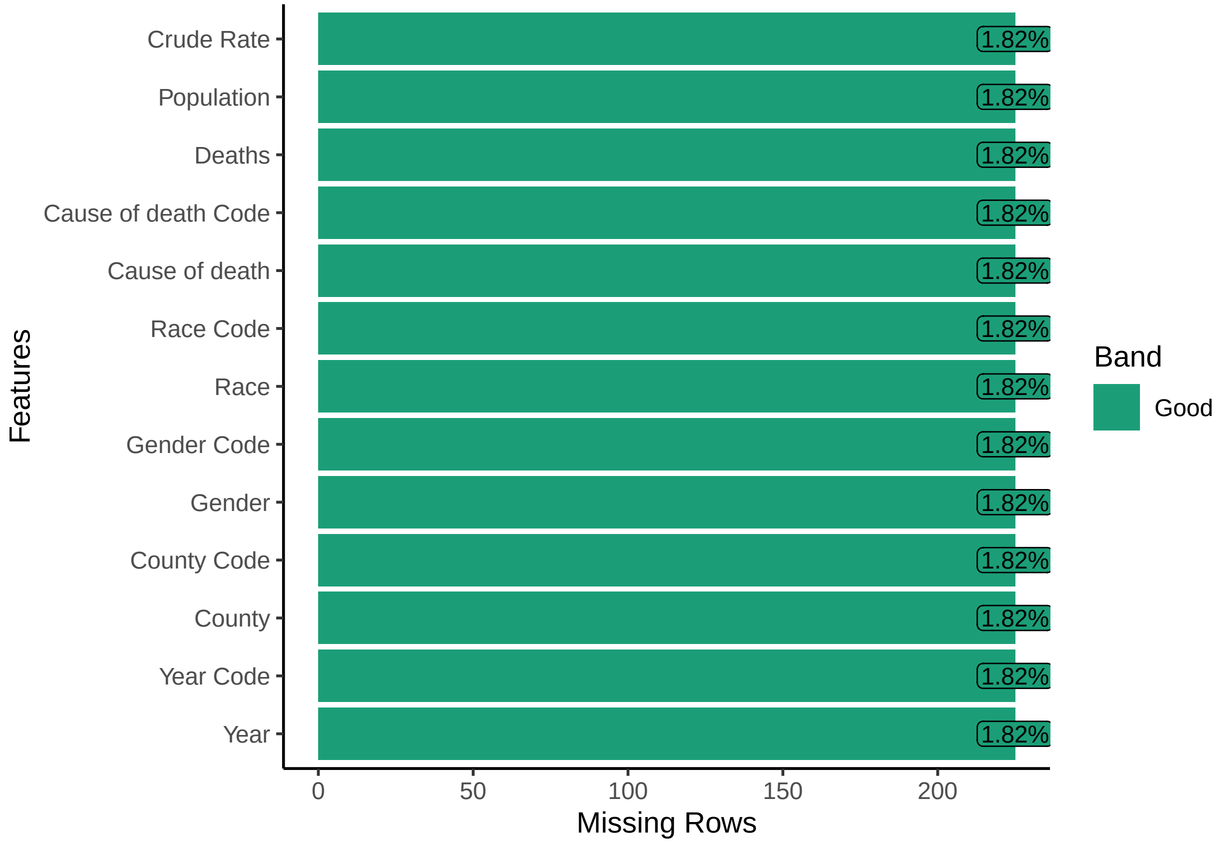
We can view if any variable is without NAs below

```
data.frame(missing = colSums(is.na(cdc_ucd_df))) |>
  filter(missing == 0) |>
  rownames()
```

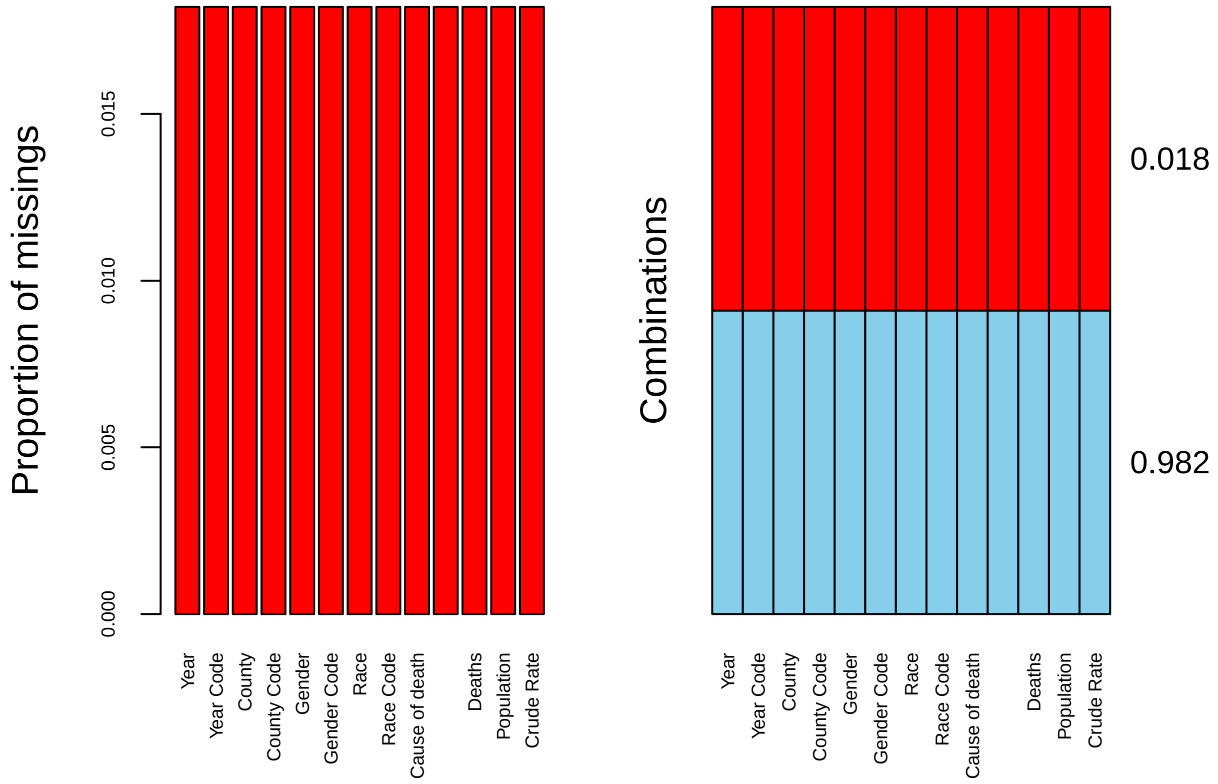
```
## character(0)
```

Considering all values have 225 NA, it is important to understand how much this would impact the overall data.

```
plot_missing(cdc_ucd_df,
             missing_only = T,
             ggtheme = theme_classic(),
             theme_config = list(legend.position = c("right")),
             geom_label_args = list("size" = 3, "label.padding" = unit(0.1, "lines")))
```



```
VIM::aggr(cdc_ucd_df, numbers=T, sortVars=T, bars = FALSE,
          cex.axis = .6)
```



```
##
##  Variables sorted by number of missings:
##             Variable      Count
##                 Year 0.01820978
##            Year Code 0.01820978
##               County 0.01820978
##          County Code 0.01820978
##               Gender 0.01820978
##          Gender Code 0.01820978
##                 Race 0.01820978
##            Race Code 0.01820978
##       Cause of death 0.01820978
##  Cause of death Code 0.01820978
##               Deaths 0.01820978
##           Population 0.01820978
##           Crude Rate 0.01820978
```

We can see that all 11 variables is missing 1.8% of values, which means the NA count of 66 observations noted from the summary is not of great concern, therefore I will actively make the decision to remove it.

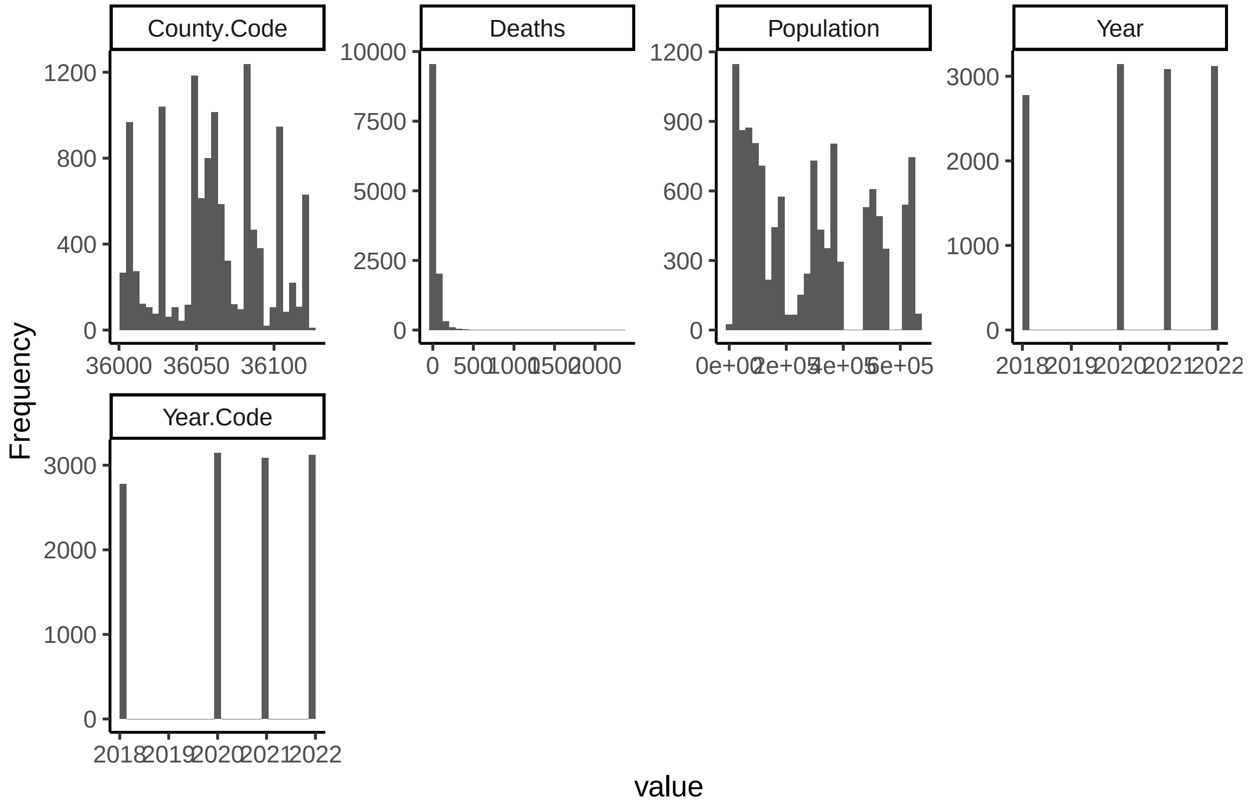
```
cdc_ucd_df<-na.omit(cdc_ucd_df)
```

```
# kable(cdc_ucd_df$`Cause of death`, format = "html", row.names = TRUE) %>%
#   kable_styling(full_width = FALSE)
```

Distributions

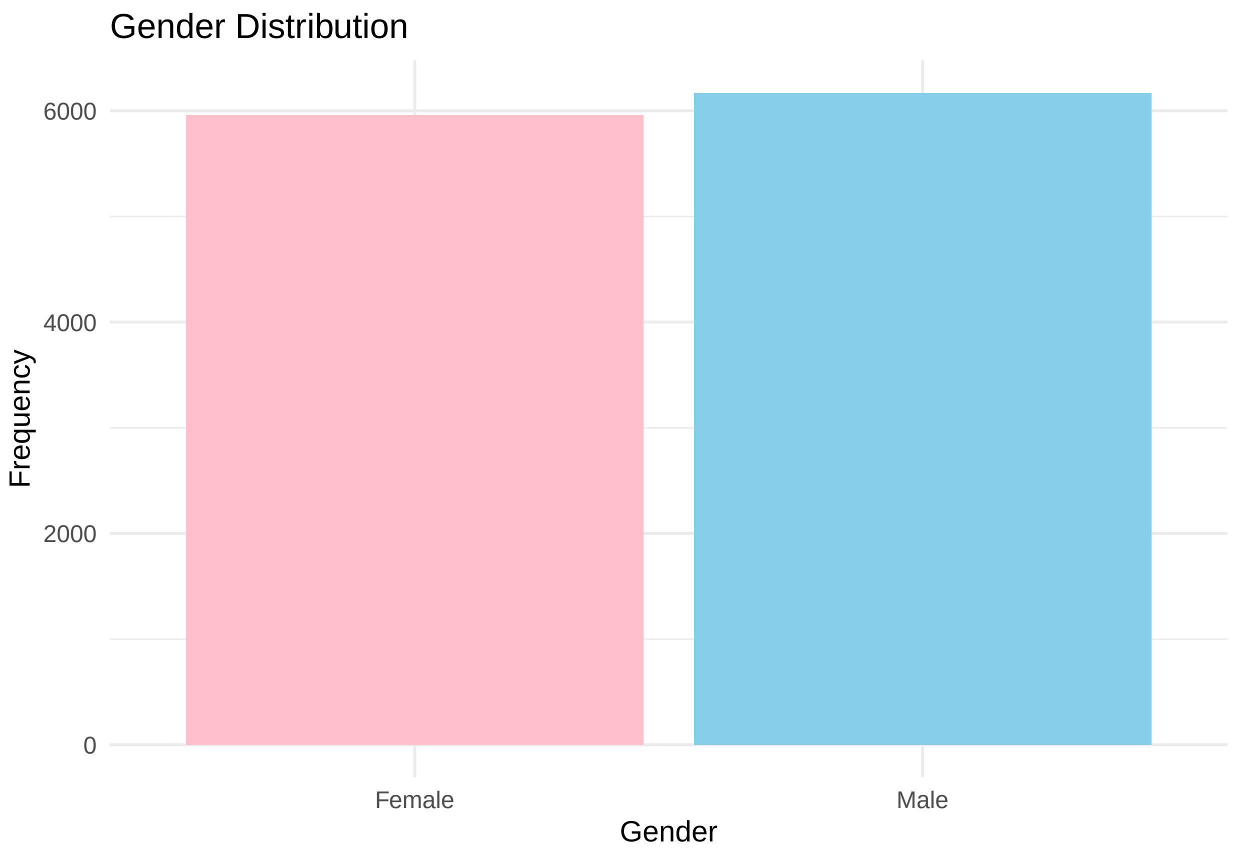
We will now take a look at the distributions of the numeric variables.

```r
DataExplorer::plot_histogram(cdc_ucd_df, nrow = 4L, ncol = 4L, ggtheme = theme_classic())
```

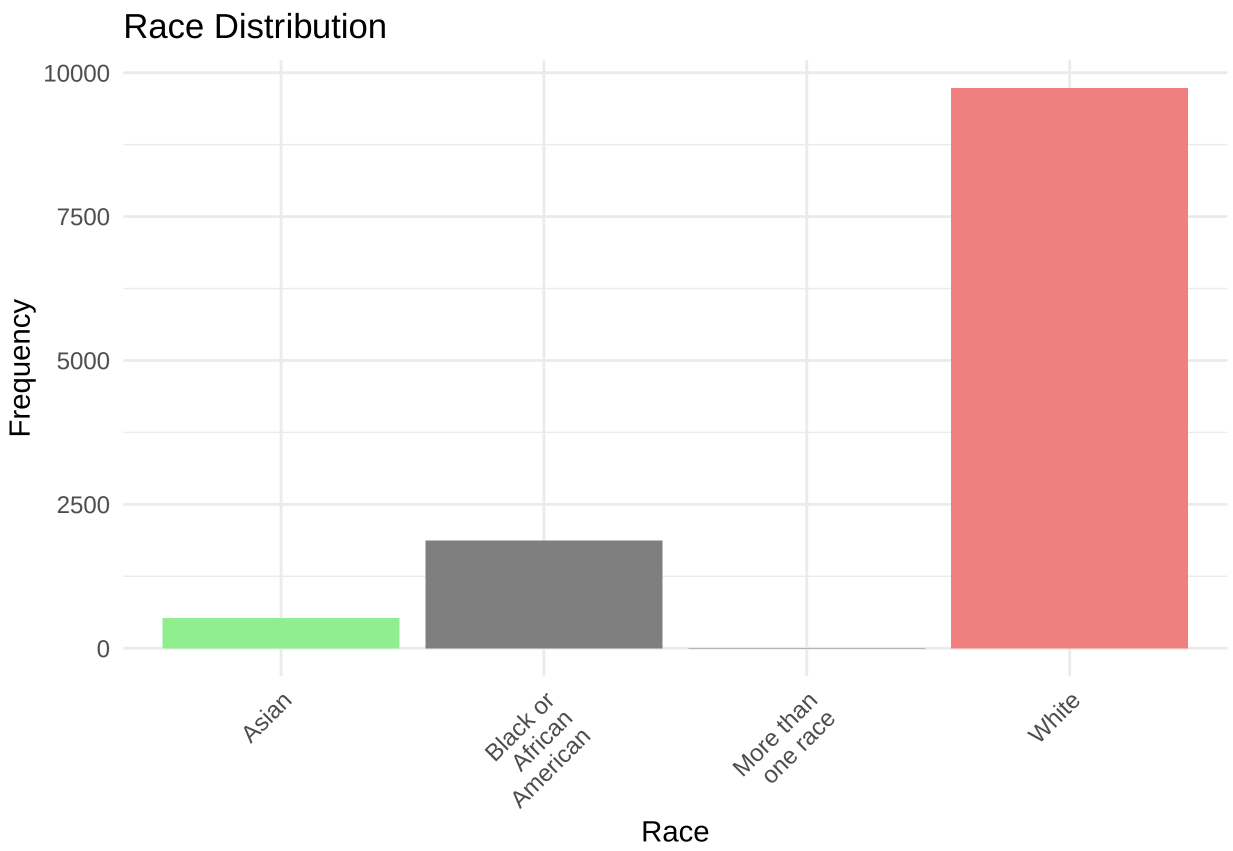


It appears none of the numeric values `County Code`, `Population` or `Deaths` is normally distributed

```r
# Create bar plot for gender distribution
ggplot(cdc_ucd_df, aes(x = Gender, fill = Gender)) +
  geom_bar() +
  labs(title = "Gender Distribution", x = "Gender", y = "Frequency") +
  theme_minimal() +  # Change the theme to minimal
  theme(legend.position = "none") +  # Remove legend
  scale_fill_manual(values = c("Male" = "skyblue",
                               "Female" = "pink"))  # Custom fill colors
```

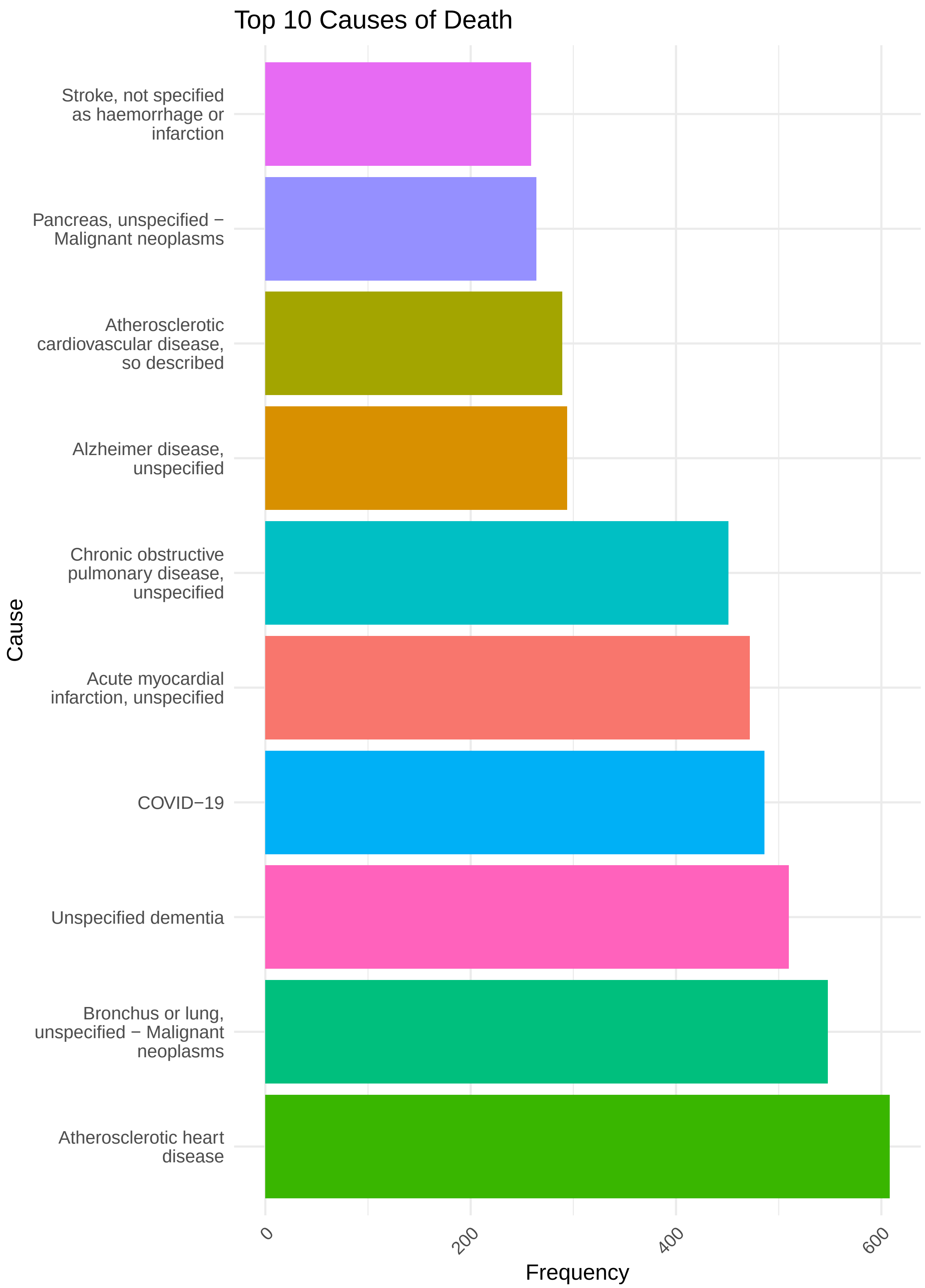


```
# Create bar plot for race distribution
ggplot(cdc_ucd_df, aes(x = str_wrap(`Race`, width = 10),
                                   fill = `Race`)) +
  geom_bar() +
  labs(title = "Race Distribution", x = "Race", y = "Frequency") +
  theme_minimal() +  # Change the theme to minimal
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 45,
                                   hjust = 1)) +  # Remove legend
  scale_fill_manual(values = c("Asian" = "lightgreen",
                               "Black" = "lightblue",
                               "White" = "lightcoral"))  # Custom fill colors
```



It also appears that deaths among men are higher than women, and among races, more deaths occurred for individuals classified as "white".

```r
# Calculate frequency of each cause of death
top_10_ca_freq<- table(cdc_ucd_df$`Cause of death`)

# Select the top 10 causes of death
top_10_ca <- names(sort(top_10_ca_freq, decreasing = TRUE))[1:10]

# Filter data to include only the top 10 causes of death
top_10_ca_data <- subset(cdc_ucd_df, `Cause of death` %in% top_10_ca)

# Create the plot with sorted values
ggplot(top_10_ca_data, aes(x = reorder(str_wrap(`Cause of death`, width = 23), -table(`Cause of death`)
  geom_bar() +
  labs(title = "Top 10 Causes of Death", x = "Cause", y = "Frequency") +
  theme_minimal() +  # Change the theme to minimal
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1)) +   # Rotate x-axis labels by 45
  coord_flip()
```


Top 10 Causes of Death
Stroke, not specified as haemorrhage or infarction
Pancreas, unspecified − Malignant neoplasms
Atherosclerotic cardiovascular disease, so described
Alzheimer disease, unspecified
Chronic obstructive pulmonary disease, unspecified
Acute myocardial infarction, unspecified
COVID−19
Unspecified dementia
Bronchus or lung, unspecified − Malignant neoplasms
Atherosclerotic heart disease
Cause
0
200
400
600
Frequency

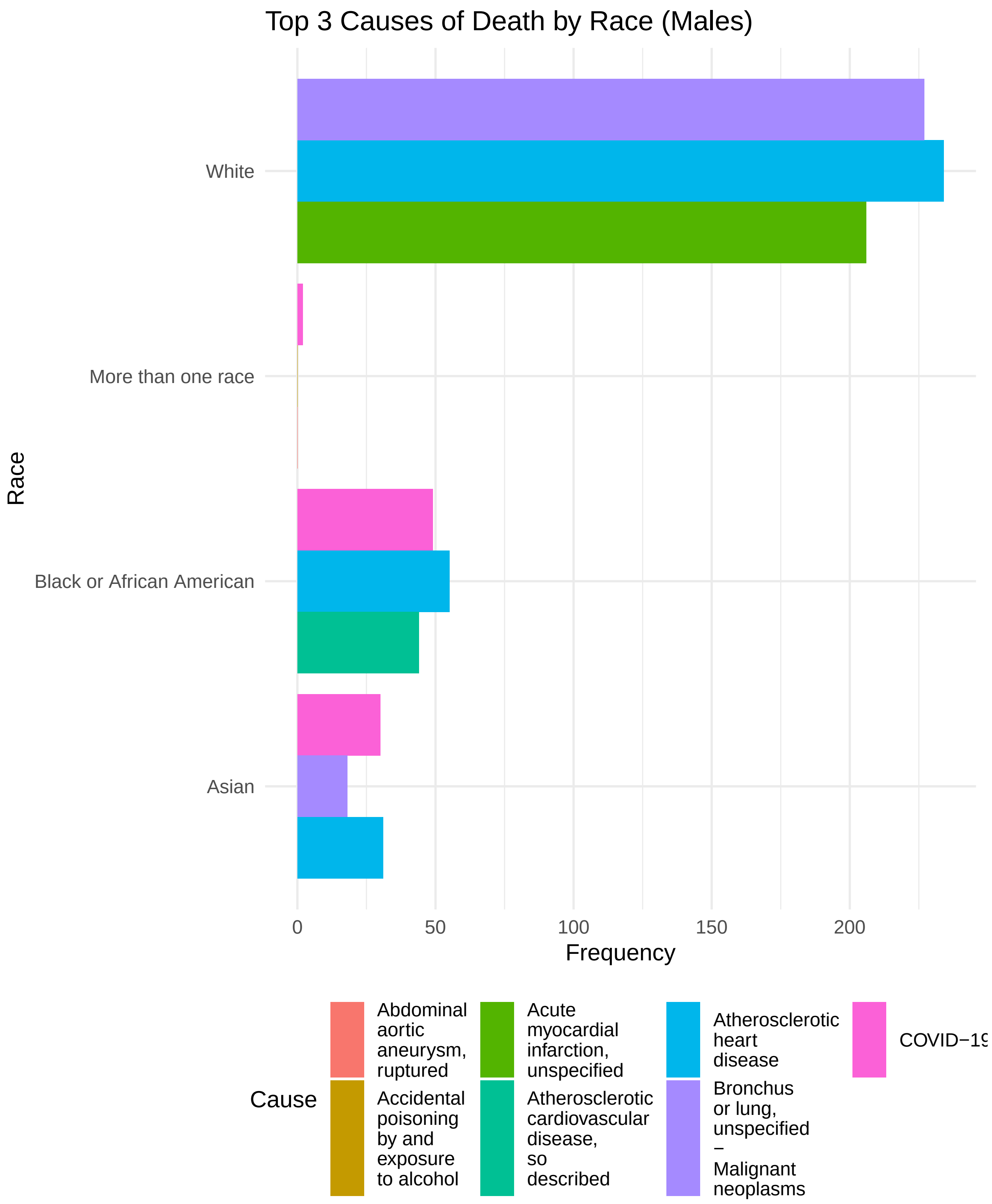
My main concern with the data is the most prominent causes of death in NYC so above I identified the top 10 for 2018-2022, and see if my models will help identify the most prominent causes it expects. I am actually surprised that COVID-19 is not ranked #1 considering the years my dataset consists of.

```
# Calculate frequency of each cause of death by race
ca_freq_rc <- table(cdc_ucd_df$Race, cdc_ucd_df$`Cause of death`,
                    cdc_ucd_df$Gender)

# Convert the frequency table to a data frame
ca_freq_rc_df <- as.data.frame.table(ca_freq_rc)

# Rename columns
names(ca_freq_rc_df) <- c("Race", "Cause","Gender", "Frequency")

# Sort by frequency in descending order
ca_freq_rc_df <-
  ca_freq_rc_df[order(ca_freq_rc_df$Frequency,
                             decreasing = TRUE),]
```

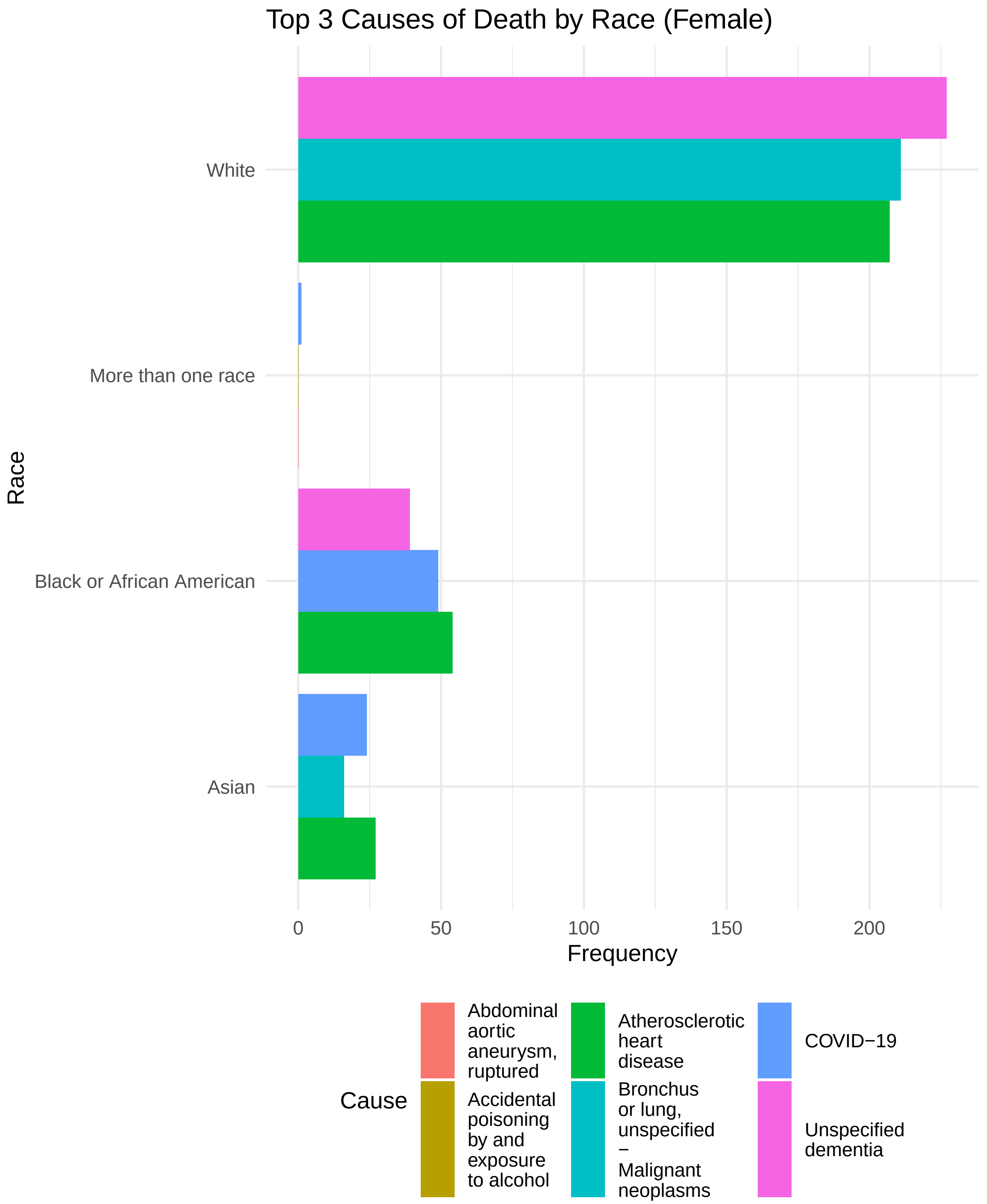
```
unique(cdc_ucd_df$Gender)
```

```
## [1] "Female" "Male"
```

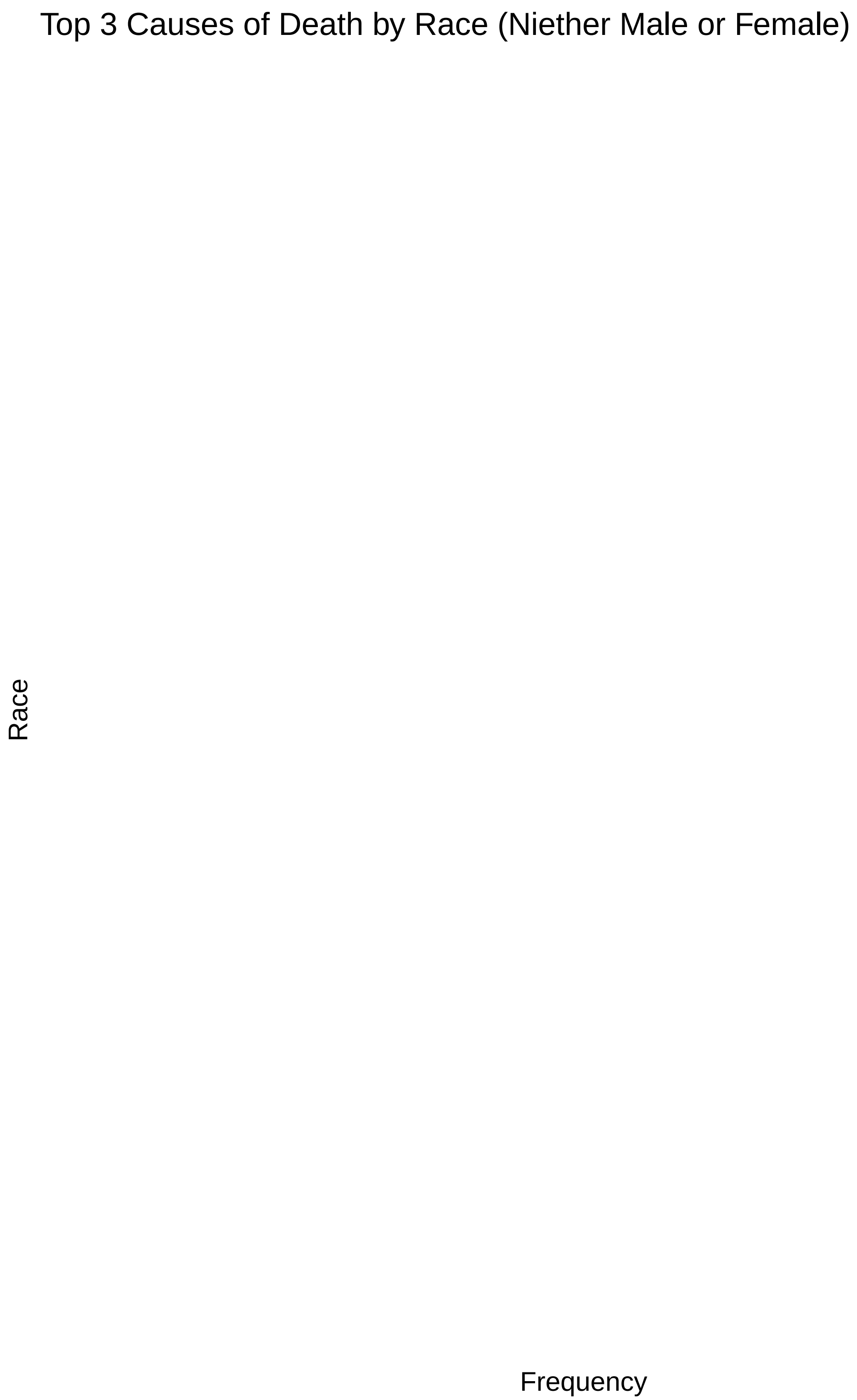
```
# Filter to keep only the top 3 causes of death for each race
top_3_causes <- do.call(rbind,
                        lapply(split(ca_freq_rc_df,
                                     list(ca_freq_rc_df$Race, ca_freq_rc_df$Gender)),
                               function(x) {
                                 race_gender <- unique(x$Race)[1]
                                 gender <- unique(x$Gender)[1]
                                 head(x[order(-x$Frequency), ], 3)
                               }))

top_3_causes_m <- subset(top_3_causes, Gender == "Male")
top_3_causes_f <- subset(top_3_causes, Gender == "Female")
top_3_causes_o <- subset(top_3_causes, Gender == "NA")


# Create the plot for males
ggplot(top_3_causes_m, aes(x = Race, y = Frequency, fill = Cause)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Top 3 Causes of Death by Race (Males)", x = "Race", y = "Frequency") +
  theme_minimal() +
  theme(legend.position = "bottom") +  # Position the legend at the bottom
  scale_fill_discrete(labels = function(x) str_wrap(x, width = 10))+
  # Manually wrap legend labels
  coord_flip()
```



```r
# Create the plot for females
ggplot(top_3_causes_f, aes(x = Race, y = Frequency, fill = Cause)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Top 3 Causes of Death by Race (Female)", x = "Race", y = "Frequency") +
  theme_minimal() +
  theme(legend.position = "bottom") +  # Position the legend at the bottom
  scale_fill_discrete(labels = function(x) str_wrap(x, width = 10))+
  # Manually wrap legend labels
  coord_flip()
```



```r
# Create the plot for females
ggplot(top_3_causes_o, aes(x = Race, y = Frequency, fill = Cause)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Top 3 Causes of Death by Race (Niether Male or Female)", x = "Race", y = "Frequency") +
  theme_minimal() +
  theme(legend.position = "bottom") +  # Position the legend at the bottom
  scale_fill_discrete(labels = function(x) str_wrap(x, width = 10))+
  # Manually wrap legend labels
  coord_flip()
```



I looked into how much, top causes of death may vary by race and gender, and decided in my model, going into that level of granularity may not be needed at the present time. I can however explore looking to this at a future date for my own purposes.

```
(unreliable_count <- sum(cdc_ucd_df$`Crude Rate` == "Unreliable", na.rm = TRUE))
```

```
## [1] 6420
```

Crude Rate also appears to have to many `Unreliable` values, approximately 6420, therefore I removed the column altogether.

Transformation

Preprocessing

First redundant categorical data which is any variable labeled `Code`. I also remove

```
cdc_model<-cdc_ucd_df%>%
                dplyr::select(-c(`Year Code`,`County Code`, `Race Code`,`Gender Code`,`Cause of death C
```

From here, the multiple classifications are set with `as.factor` and `Gender` is simply set to character. From there we preprocess the data, and use `predict()` for our model.

```
cdc_model <- cdc_model %>%
  mutate(
    County = as.factor(`County`),
    Gender = as.factor(Gender),
    Race = as.factor(str_trim(Race)),
    `Cause of death` = as.factor(`Cause of death`),
    Year = as.Date(paste0(Year, "-01-01"))  # Convert Year to Date with January 1st as the date
  ) %>%
  predict(preProcess(., method = c("center", "scale")), .)
```

Running predict to normalize the data made the values appear unlikely with negative decimals. For the purposes of this project I will move forward but resolving this for later interpretation may prove difficult.

Models

SVM

SVM is a model that is ideal for high-dimensional data, so I attempted to utilize it for this data set and use of `eps-regression` is based on the fact that the annual data can be considered continous.

```
# Set seed for reproducibility
set.seed(1234)
#
# # Process the data: trim whitespace and convert to factors
# cdc_model <- cdc_model %>%
#   mutate(
#     County = as.factor(str_trim(County)),
#     Gender = as.factor(str_trim(Gender)),
#     Race = as.factor(str_trim(Race)),
```

```r
#     `Cause of death` = as.factor(str_trim(`Cause of death`))
#   )

# Remove rows with NA values
cdc_model <- na.omit(cdc_model)

# Split the data
training.samples <- cdc_model$Deaths %>%
  createDataPartition(p = 0.8, list = FALSE)

train_df <- cdc_model[training.samples, ]
test_df <- cdc_model[-training.samples, ]

# Identify and remove constant variables in the training set
constant_vars <- sapply(train_df, function(x) length(unique(x)) == 1)
train_df <- train_df[, !constant_vars]

# Ensure the same columns are removed from the test set
test_df <- test_df[, colnames(train_df)]

# Fit the SVM model
svm_model <- svm(formula = Deaths ~ ., data = train_df, type = 'eps-regression')

# Print the SVM model
print(svm_model)
```

```
##
## Call:
## svm(formula = Deaths ~ ., data = train_df, type = "eps-regression")
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
##        cost:  1
##       gamma:  0.00390625
##     epsilon:  0.1
##
##
## Number of Support Vectors:  5171
```

```r
# Predict using the SVM model
predictions_SVM <- predict(svm_model, newdata = test_df)

# Combine predictions with the test dataset
predictions_SVM <- data.frame(Predicted = predictions_SVM, test_df)

# Print predictions
# print(predictions_SVM)
```

Note SVM-Kernel: radial is the default

```r
predictions_SVM <- predict(svm_model, newdata = test_df) %>%
  bind_cols(test_df)
```

```
## New names:
## * `` -> `...1`
```

```r
predictions_SVM$...1 <- as.numeric(predictions_SVM$...1)
```

Performance and Comparison

```r
MAE <- MAE(predictions_SVM$Deaths, predictions_SVM$...1)
RMSE <- RMSE(predictions_SVM$Deaths, predictions_SVM$...1)
R2 <- R2(predictions_SVM$Deaths, predictions_SVM$...1)

# Create a data frame to store the results
a_svm <- data.frame(Model = "SVM",
                MAE = MAE,
                RMSE = RMSE,
                R2 = R2)

# Print the results
print(a_svm)
```

```
##   Model       MAE     RMSE        R2
## 1   SVM 0.4065106 1.089014 0.2689261
```

Not suprisingly the model did very poorly. Choosing a dataset with only 1 numeric value was a drastic change from previous data, therefore learning how to best utilize SVM or manipulate the data for accuracy is something I will look further into.

Random Forest Regression Tree

Random Forest Regression Tree is an obvious choice considering most of the data is categorical. The only challenge I foresee is creating a clear visual.

```r
# Calculate frequency of each cause of death
top_10_ca_freq<- table(cdc_model$`Cause of death`)

# Select the top 10 causes of death
top_10_ca <- names(sort(top_10_ca_freq, decreasing = TRUE))[1:10]

# Filter data to include only the top 10 causes of death
top_10_ca_data <- subset(cdc_model, `Cause of death` %in% top_10_ca)

top_10_ca_data<-top_10_ca_data%>%
                  dplyr::select(-Population)
```

```
set.seed(1234)

cdc_rf_model <- top_10_ca_data

#split
training_cdc_samples <- cdc_rf_model$Deaths %>%
  createDataPartition(p = 0.8, list = FALSE)

train_cdc  <- cdc_rf_model[training_cdc_samples, ]
test_cdc <- cdc_rf_model[-training_cdc_samples, ]

#train using rpart, cp- complexity, smaller # = more complexity,
#method- anova is for regression
tree_cdc <- rpart(Deaths ~., data = train_cdc, cp = 0.004,  method = 'anova')

#visualize
# rpart.plot(tree_cdc)
# print(tree_1k1)
```

```
# Open a PNG graphics device
png("tree_cdc.png", width = 1000, height = 600, res=300)  # Adjust width and height as needed

# Plot the tree using rpart.plot
rpart.plot(tree_cdc)
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

```
# Close the graphics device and save the plot as "tree_cdc.png"
dev.off()
```

```
## pdf
##   2
```

Because the Tree model was difficult to see I made it into a PNG. However because of there were an excessive amount of variables, most labels are still difficult to view.The PNG will be provided with my submission.

```
# Open a PNG graphics device with high resolution
png("tree_cdc_hd.png", width = 2040, height = 1200, res = 300)  # 300 DPI

# Plot the tree using rpart.plot with custom text settings
rpart.plot(tree_cdc, extra = 101, type = 3, under = TRUE,  faclen = 0, varlen = 0, snip = TRUE,
           cex = .3, # Increase font size
           branch.lty = 1, branch.lwd = .5, # Set branch line type and width
           main = "Decision Tree for CDC Data", # Add a main title
           split.cex = .5, split.box.col = "lightblue", split.border.col = "blue") # Customize split no
```

```
## Warning: labs do not fit even at cex 0.15, there may be some overplotting
```

```
## Warning: ignoring snip=TRUE for png device
```

```
# Close the graphics device and save the plot as "tree_cdc_hd.png"
dev.off()
```

```
## pdf
##   2
```

Predictions

```
predictions_tree <- predict(tree_cdc, newdata = test_cdc) %>%
  bind_cols(test_cdc )

predictions_tree$...1 <- as.numeric(predictions_tree$...1)
```

```
decision_tree_model <- data.frame(Model = "Decision Tree 1",

MAE = ModelMetrics::mae(predictions_tree$Deaths, predictions_tree$...1),
#rmse Root Mean Squared Error
RMSE = ModelMetrics::rmse(predictions_tree$Deaths, predictions_tree$...1),
#r squared
R2 = caret::R2(predictions_tree$Deaths, predictions_tree$...1)
)

decision_tree_model
```

```
##             Model       MAE      RMSE        R2
## 1 Decision Tree 1 0.4034347 0.8857448 0.6272207
```

Essay

The objective of my project is to utilize recent mortality data to train a model that can potentially predict its impact on specific demographics. The data was sourced from the Center for Disease Control (CDC) website, specifically from the dataset titled "2018-2022 Underlying Cause of Death by Single Race Categories." This dataset is primarily categorical, presenting a personal challenge since my previous work throughout the semester primarily involved numerical data.

This project aims to identify underlying causes of death that may disproportionately affect different communities. Factors such as social disparity, limited access to healthy foods, excessive access to fast food, limited access to parks and outdoor activities, and availability of sports facilities by region could all be contributing factors. Although this project does not delve into these specific factors, it applies the skills learned this semester to localize these underlying causes of death by county and demographic. Visualizations created for race and gender groupings illustrate the potential of this data to inform stakeholders and decision-makers, with the goal of improving the lives of various regions and communities.

Data collection faced obstacles due to the CDC website's maximum query output of 75,000 observations. To mitigate this challenge, the data was limited to New York State and collected in annual batches before merging. During my exploratory data analysis (EDA), I prioritized data integrity, assessing for missing values and identifying 225 NA observations. Given their negligible impact on the dataset, these observations were omitted from the models. The numerical data included deaths and population counts, which were not normalized. With an average death toll of 38 and a standard deviation of 79, there was concern about the model's fit. Plotting the distributions confirmed that the death count was right-skewed and not normalized.

All categorical data included both coded and descriptive text formats. For this project, the descriptive text was used to enhance readability. Redundant codes were removed, categories were set as factors, and the year was treated as a date value for preprocessing.

Given that Support Vector Machine (SVM) models perform well with high-dimensional data, I employed this technique for the dataset. Based on guidance from R documentation and Stack Exchange sites, I determined that 'eps-regression' was the optimal parameter for the 'type' argument, considering the continuous nature of the annual data. However, the results were as follows:

Model	MAE	RMSE	R^2
SVM	0.4065	1.0890	0.2689

The SVM model proved to be a poor fit.

Subsequently, I utilized a random forest model, generating the image tree_cdc_hd.png to better visualize the results. The results were:

Model	MAE	RMSE	R^2
Decision Tree 1	0.4034	0.8857	0.6272

The decision tree model demonstrated a better fit compared to the SVM model.

In this project, I deliberately retained the noted errors as learning opportunities for future endeavors. Firstly, while I believe my data selection was appropriate, I need to further investigate the transformation of numerical values. Specifically, the normalization process via the predict() function resulted in negative decimal values, posing a challenge in accurately predicting future outcomes. Additionally, using the Code definitions for categories in the Random Forest model could have enhanced the model's visualization. However, this introduced interpretation challenges. Moving forward, I intend to continue refining this dataset to address these common interpretative issues, thereby improving the clarity and utility of my models in similar future projects.