# Programming in R

Blake Pappas

September 26, 2022

## Load the Appropriate Packages

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(ggplot2)
```

## Import and Tidy the Data

```r
gss <- select(gss_cat, marital, race, relig, age, tvhours) %>%
  mutate(marital = marital %>% fct_infreq() %>% fct_rev()) %>% # Order of frequency
  mutate(race = race %>% fct_infreq() %>% fct_rev()) %>% # Order of frequency
  mutate(relig = relig %>% fct_infreq() %>% fct_rev()) # Order of frequency
```

## Assert That the Data Types Are Correct

```r
is.factor(gss$marital)
```

```
## [1] TRUE
```

```r
is.factor(gss$race)
```

```
## [1] TRUE
```

```
is.factor(gss$relig)
```

```
## [1] TRUE
```

```
is.numeric(gss$age)
```

```
## [1] TRUE
```

```
is.numeric(gss$tvhours)
```

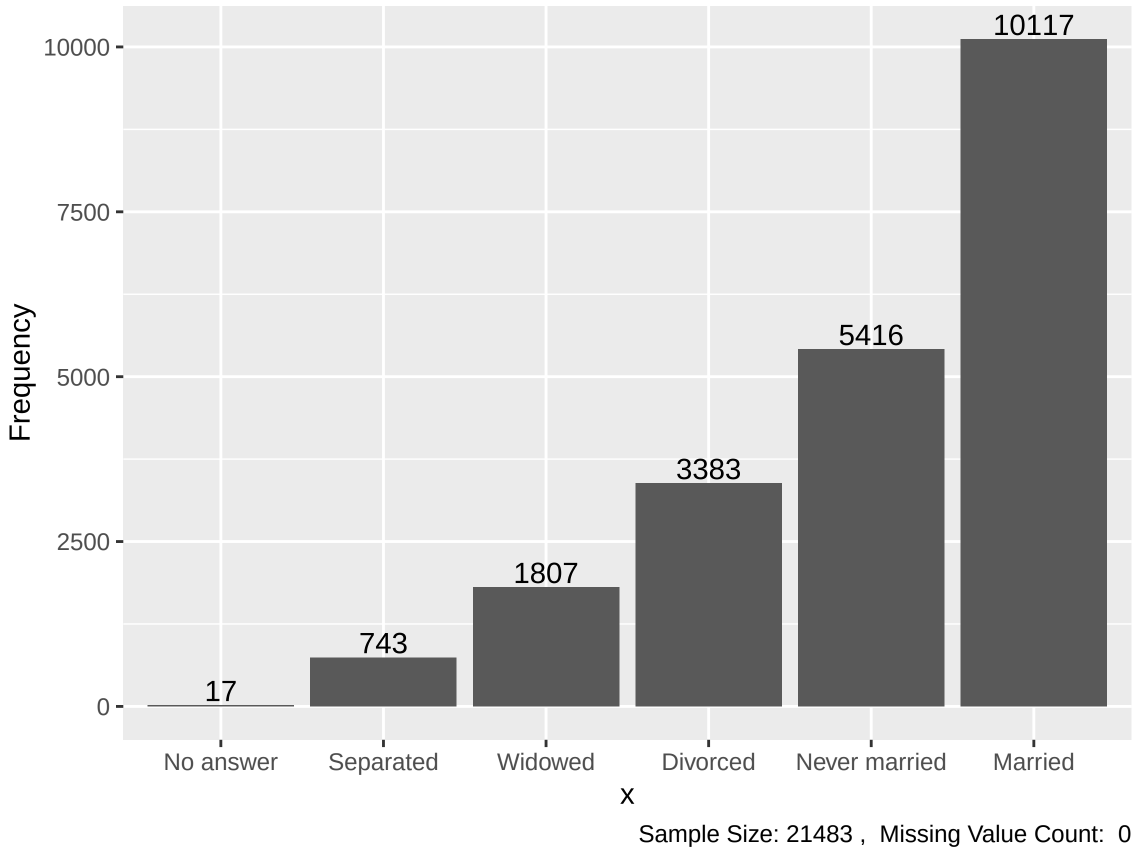
```
## [1] TRUE
```

### Create the Factor Variable Function

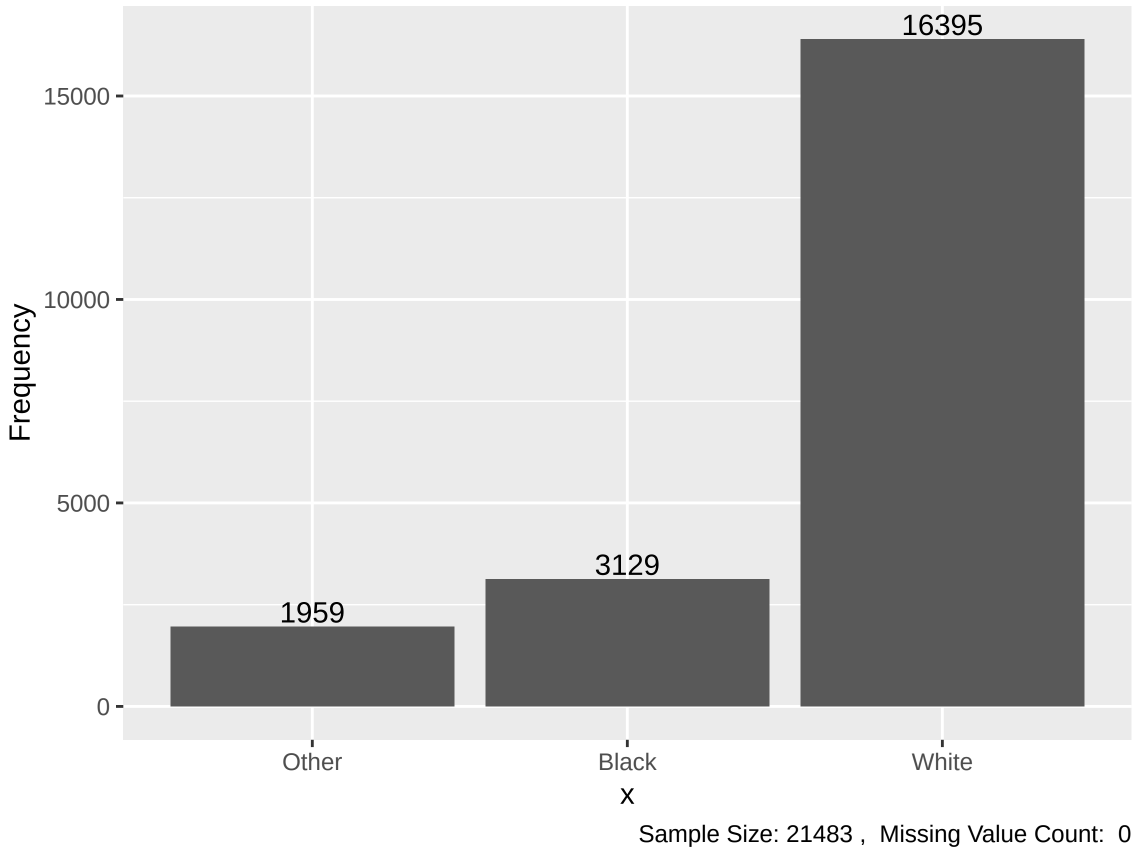
```
function_1 <- function(x) {
  if (is.factor(x)) {
    return(ggplot(data = gss, aes(x = x)) +
             geom_bar() +
             geom_text(aes(label = ..count..), stat = "count", vjust = -0.2, colour = "black") +
             labs(title = "", subtitle = "", caption = paste("Sample Size:", nrow(gss), ", ", "Missing
             ylab("Frequency")) +
             xlab()
  } else {
    return("Error: Variable data type must be a factor")
  }
}
```

### Test the Factor Variable Function (`function_1`)

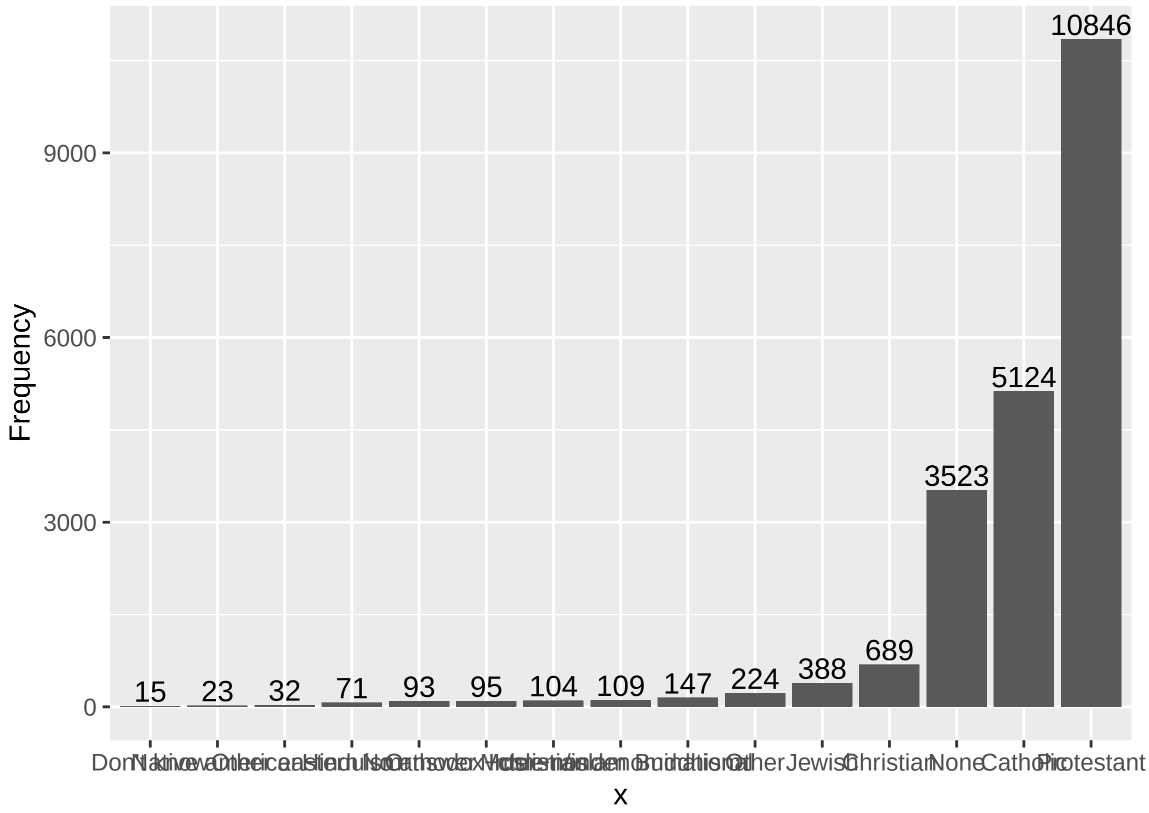
```
function_1(gss$marital)
```



```
function_1(gss$race)
```



```
function_1(gss$relig)
```



Sample Size: 21483 , Missing Value Count: 0

```
function_1(gss$age)
```

```
## [1] "Error: Variable data type must be a factor"
```

```
function_1(gss$tvhours)
```

```
## [1] "Error: Variable data type must be a factor"
```

### Create the Numeric Variable Function

```
function_2 <- function(x) {
  if (nrow(gss) <= 10 && is.numeric(x) == TRUE) {
    boxplot(x, horizontal = TRUE, range = 0, ylim = endaxis, axes = FALSE, col= "grey", add = FALSE, sta
text(x = fivenum(x), labels = fivenum(x), y = 1.25) # Labels for quartiles and median values
  } else if (nrow(gss) <= 10 && is.numeric(x) == FALSE) {
    return("Error: Variable data type must be numeric")
  } else if (nrow(gss) > 10 && is.numeric(x) == TRUE) {
    # Histogram
    mean.x <- mean(x, na.rm = TRUE)
```

```r
    sd.x <- sd(x, na.rm = TRUE)
    hist(x, main = "",xlab = "x", col = "blue", label = TRUE, plot = TRUE, freq = T)

    # Labels for mean and standard deviation
    text(x = c(round(mean.x, digits = 2), round(mean.x + sd.x, digits = 2), round(mean.x - sd.x, digits
  } else if (nrow(gss) > 10 && is.numeric(x) == FALSE) {
    return("Error: Variable data type must be numeric")
  }
}
```

### Test the Numeric Variable Function (`function_2`)

```r
function_2(gss$marital)
```

```
## [1] "Error: Variable data type must be numeric"
```

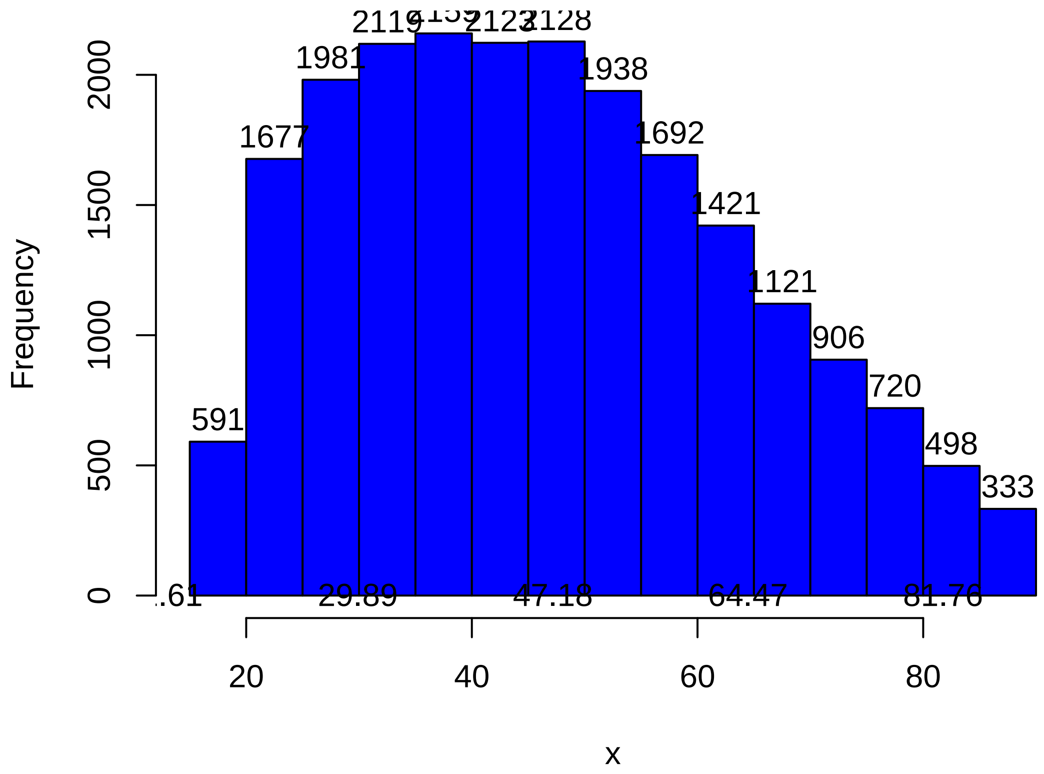
```r
function_2(gss$race)
```

```
## [1] "Error: Variable data type must be numeric"
```

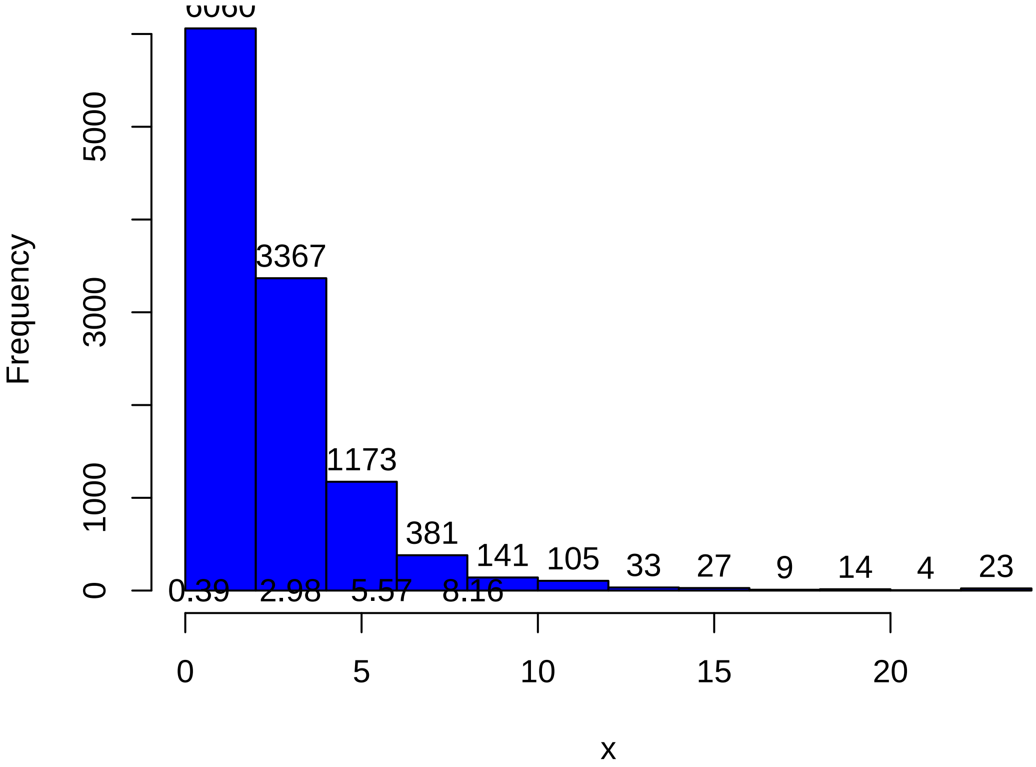
```r
function_2(gss$relig)
```

```
## [1] "Error: Variable data type must be numeric"
```

```r
function_2(gss$age)
```



```
function_2(gss$tvhours)
```


6060
3367
1173
381
141
105
33
27
9
14
4
23
0.39
2.98
5.57
8.16
Frequency
0
1000
3000
5000
0
5
10
15
20
x