Part A. Writing some basic R functions

The function imputeNA returns a modified copy of data with missing values (NAs) imputed. Continuous variables (numeric types) are imputed using the median or mean of the non-missing values. Categorical variables are imputed using the mode.

```
testdf <- data.frame(
  row.names=c("Jack", "Rosa", "Dawn", "Vicki", "Blake", "Guillermo"),
  age=c(24, 23, NA, 25, 32, 19), city=c("Harlem", NA, "Queens", "Brooklyn", "Brooklyn", NA),
gpa=c(3.5, 3.6, 4.0, NA, 3.8, NA))

testdf
```

```
##           age     city gpa
## Jack       24   Harlem 3.5
## Rosa       23     <NA> 3.6
## Dawn       NA   Queens 4.0
## Vicki      25 Brooklyn  NA
## Blake      32 Brooklyn 3.8
## Guillermo  19     <NA>  NA
```

```
get_mode <- function(x) {
  unique_vals <- unique(x[! x %in% c(NA)])
  tab <- tabulate(match(x, unique_vals))
  mode <- unique_vals[tab == max(tab)]
  if(length(mode) > 1){
    mode <- mode[1]
  }
  mode
}

# get_mode(c("Harlem", NA, "Queens", "Brooklyn", "Brooklyn", NA))
```

```
imputeNA <- function(data, use.mean=TRUE) {
  for(i in 1:ncol(data)) {
    if(is.numeric(data[ , i])) {
      if(use.mean == TRUE){
        data[ , i][is.na(data[ , i])] <- mean(data[ , i], na.rm = TRUE)
      }else{
        data[ , i][is.na(data[ , i])] <- median(data[ , i], na.rm = TRUE)
      }
    }else{
      data[ , i][is.na(data[ , i])] <- get_mode(data[ , i])
    }
  }
  data
}

imputed_df <- imputeNA(testdf)
imputed_df
```

```
##            age     city   gpa
## Jack      24.0   Harlem 3.500
## Rosa      23.0 Brooklyn 3.600
## Dawn      24.6   Queens 4.000
## Vicki     25.0 Brooklyn 3.725
```

```
## Blake     32.0 Brooklyn 3.800
## Guillermo 19.0 Brooklyn 3.725
```

```
imputed_df <- imputeNA(testdf, TRUE)
imputed_df
```

```
##            age     city   gpa
## Jack      24.0   Harlem 3.500
## Rosa      23.0 Brooklyn 3.600
## Dawn      24.6   Queens 4.000
## Vicki     25.0 Brooklyn 3.725
## Blake     32.0 Brooklyn 3.800
## Guillermo 19.0 Brooklyn 3.725
```

```
imputed_df <- imputeNA(testdf, FALSE)
imputed_df
```

```
##           age     city gpa
## Jack       24   Harlem 3.5
## Rosa       23 Brooklyn 3.6
## Dawn       24   Queens 4.0
## Vicki      25 Brooklyn 3.7
## Blake      32 Brooklyn 3.8
## Guillermo  19 Brooklyn 3.7
```

The function countNA returns a named numeric vector giving the count of missing values (NAs) for each row or each column of data (depending on the value of byrow). The names of the result are the rownames() or colnames() of data, whichever is appropriate.

```
countNA <- function(data, byrow = FALSE) {
  if(byrow) {
    counts <- rowSums(is.na(data))
  }else{
    counts <- sapply(data, function(x) sum(is.na(x)))
  }
  counts
}

countNA(testdf)
```

```
##  age city  gpa
##    1    2    2
```

```
countNA(testdf, byrow=TRUE)
```

```
##      Jack      Rosa      Dawn     Vicki     Blake Guillermo
##         0         1         1         1         0         2
```

Part B. Using in-built R datasets to create basic plots.

1. Using the police_killings dataset from fivethirtyeight package, we would like to visualize the distribution of Americans killed by police by race and income. First, use the na.omit() function to remove missing data from the dataset. Then, visualize the count of Americans killed of each race/ethnicity, broken out by national quintile of household income.

```
library(fivethirtyeight)
```

```
## Some larger datasets need to be installed separately, like senators and
```

```
## house_district_forecast. To install these, we recommend you install the
## fivethirtyeightdata package by running:
## install.packages('fivethirtyeightdata', repos =
## 'https://fivethirtyeightdata.github.io/drat/', type = 'source')
```

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

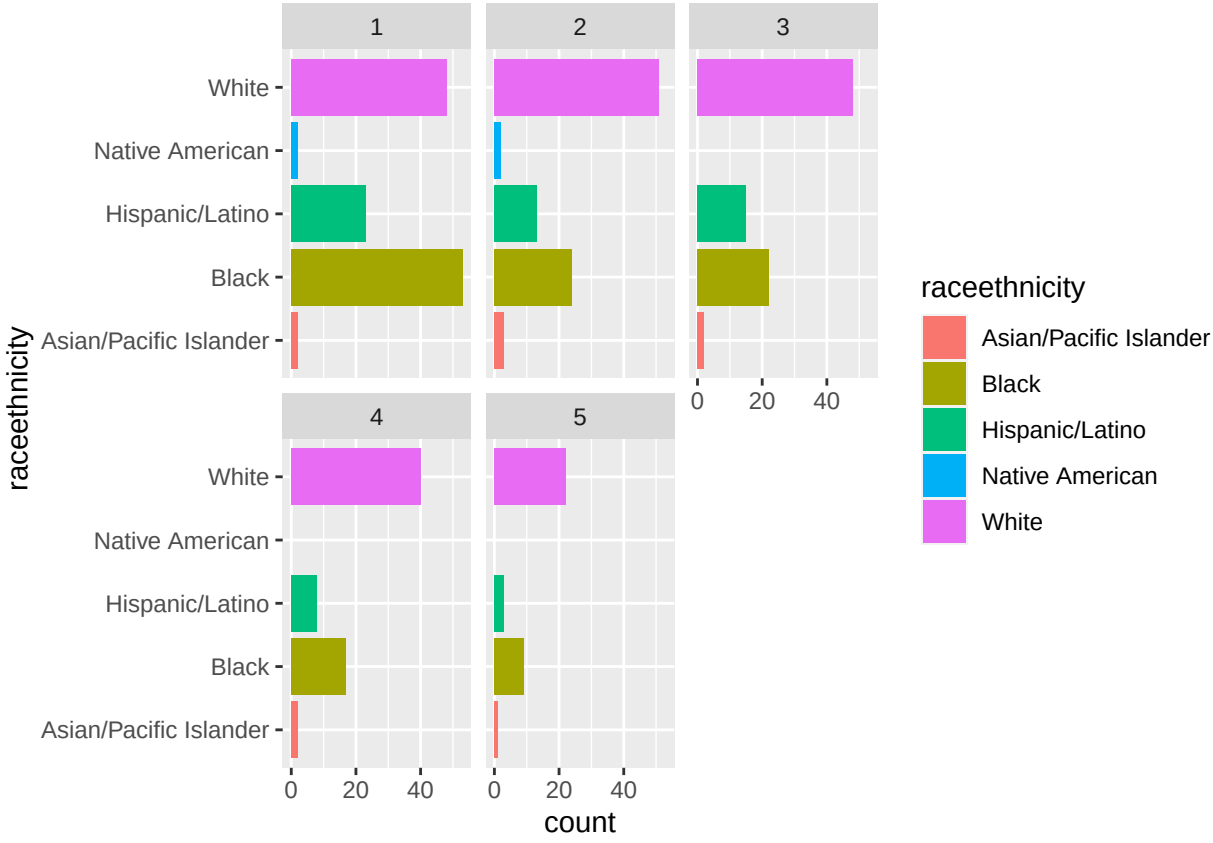
```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
```

```
ggplot(data = police_killings, mapping = aes(x=raceethnicity, fill=raceethnicity)) + geom_histogram(sta
coord_flip()
```

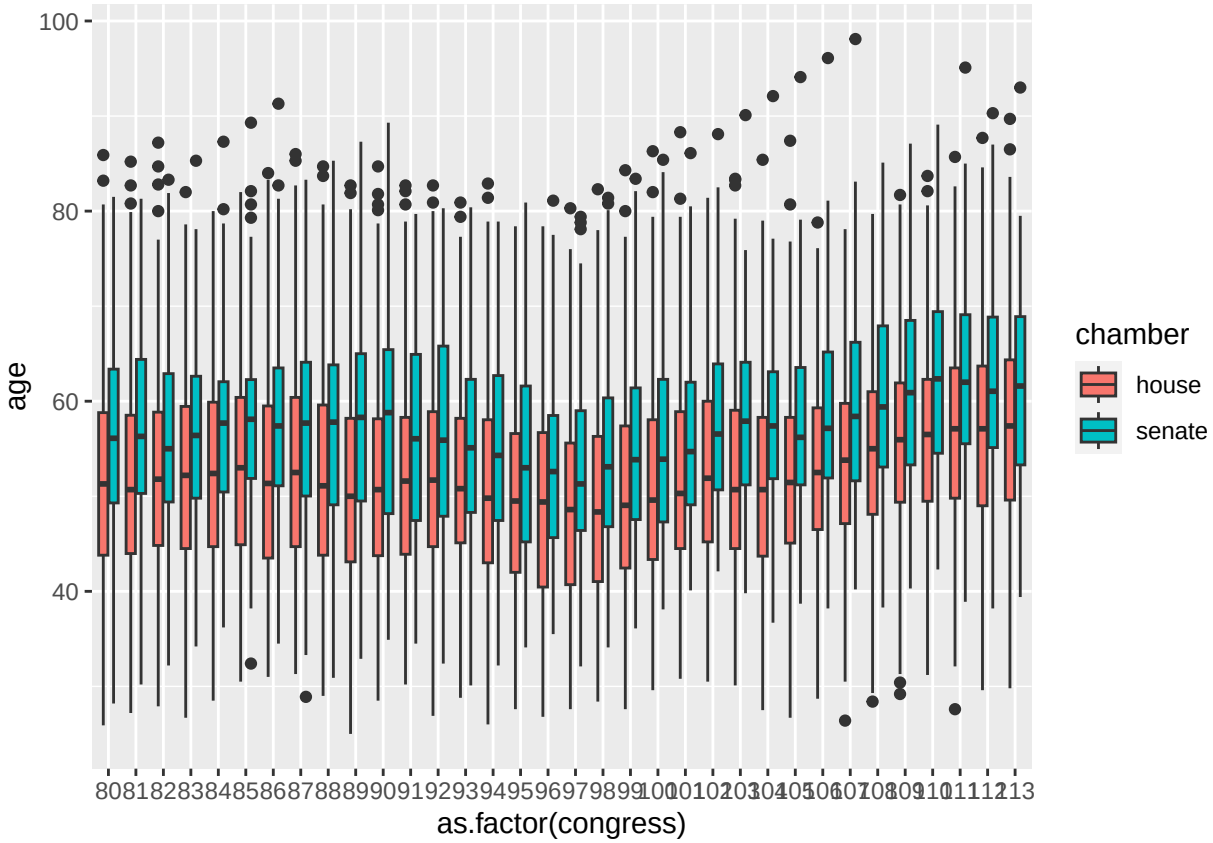
```
## Warning in geom_histogram(stat = "count"): Ignoring unknown parameters:
## `binwidth`, `bins`, and `pad`
```



Majority killings are in the black and white ethnicity. For the lowest income quintiles, Black race people are killed most and White ethnicity people are killed for all other income groups. Asian/Pacific Islander and Native American account for very less percent of our dataset.

2. Using the congress_age dataset, we would like to visualize the distribution of ages in US Congress. Used box-and-whiskey plots to visualize the distribution of ages for each congress number (#80 through #113), broken out by the congress chamber (House and Senate).

```
ggplot(congress_age, aes(x = as.factor(congress), y = age, fill = chamber)) +
geom_boxplot()
```



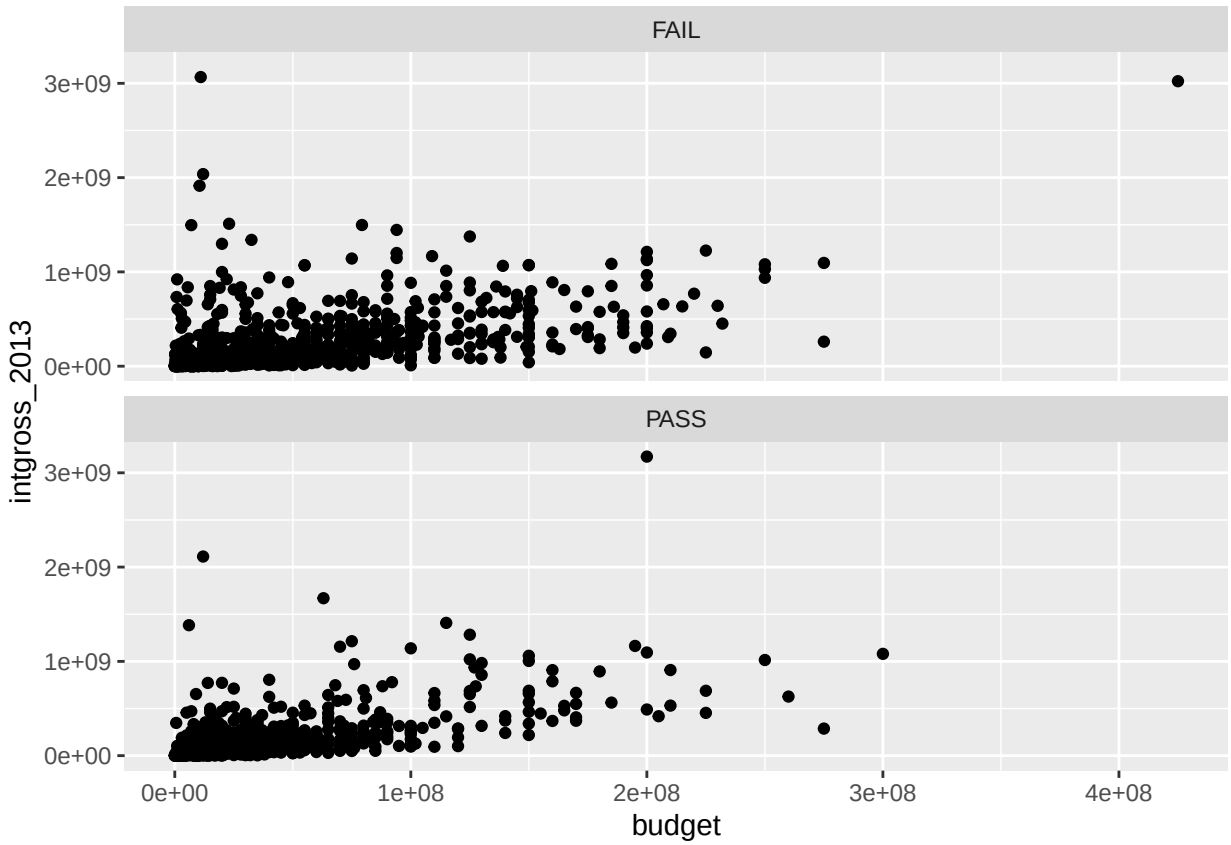
For both Senate and House chamber, the median is almost same from chamber numbers 80-90, then there is a dip in the median and remains same for chamber numbers 90-100, then there is an increase in the median from chamber numbers 100-113.

The median age for Senate chamber is higher (older in age) than House chamber across all chamber numbers.

3. Using the bechdel dataset, we would like to investigate if there is a relationship between passing the Bechdel test and the amount of money spent and made from a movie. The Bechdel test is a basic set of criteria designed to reveal trends of gender bias in the movies. The test asks: does a movie (1) have at least two female characters (2) who talk to each other (3) about something other than a man

```
ggplot(bechdel) +
geom_point(mapping = aes(x = budget, y = intgross_2013)) +
  facet_wrap(~ binary, nrow = 2)
```

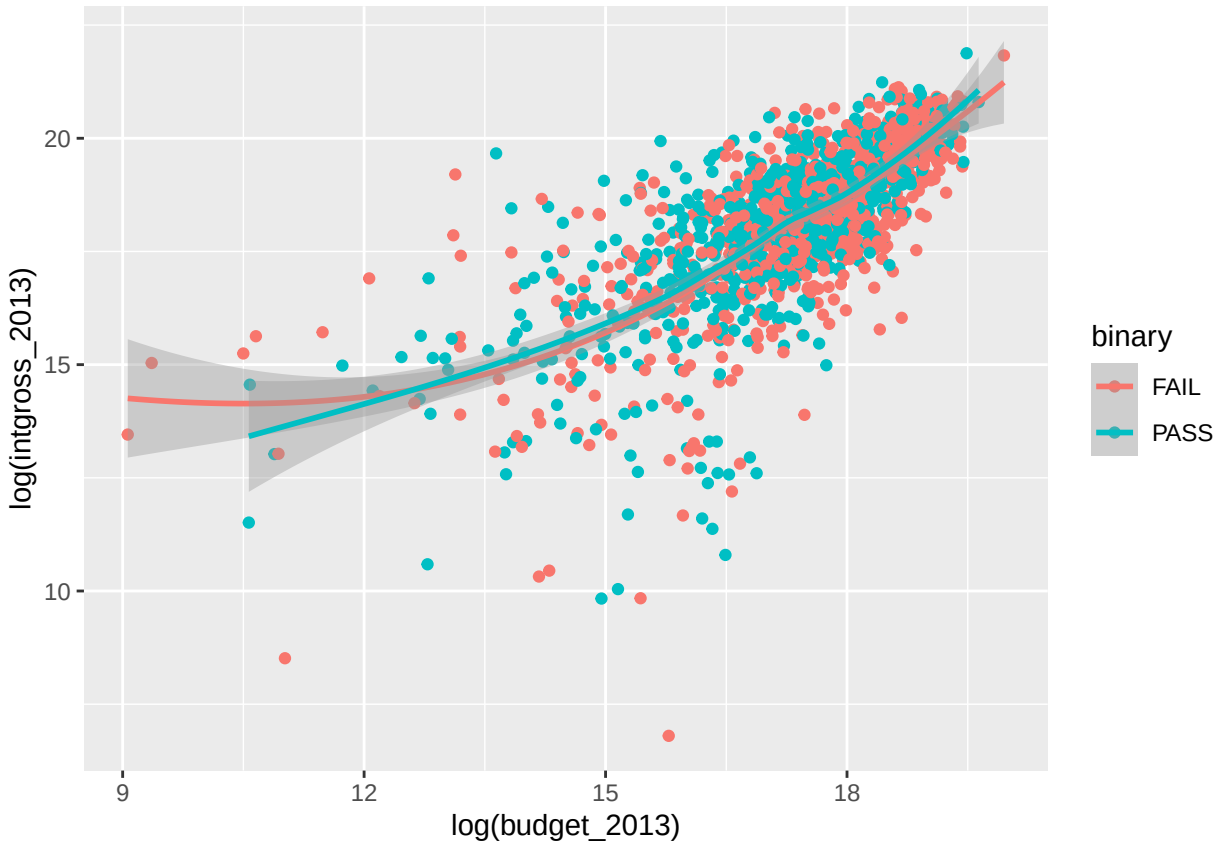
```
## Warning: Removed 11 rows containing missing values (`geom_point()`).
```




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

ggplot(bechdel, aes(x = log(budget_2013), y = log(intgross_2013), color = binary)) +
geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Passing or failing the Bechdel test does not affect the relationship between the budget and gross income. There is a positive correlation between the movie budget and gross income. There are a few outliers.