STT 301 Homework Assignment 2

Shawn Santo

September 26, 2018

```
library(knitr)
opts_chunk$set(comment = NA, message = FALSE, warning = FALSE)
```

Homework Assignment 2 is due Wednesday, October 3 at 12:40pm EST.

Instructions

You must complete this individual homework assignment using R Markdown. Submit the R Markdown file, which should have a `.Rmd` extension, via the dropbox on D2L.

Some of the questions are open-ended, and there is not a correct or incorrect answer. Written responses to questions should be incisive.

Rubric

- **Total**: 10 points.
- **Correctness**: Point values for the question and their respective parts are listed. Partial credit is available. Hard-coded solutions will not receive full credit.
- **Knitting**: Deduction of 0.5 points if the Rmd file does not knit for any reason.
- **Style**: Use a third-level header to off-set each question in your solutions - as is done below. For questions with multiple parts (part a, part b, etc), use fourth-level headers to off-set the parts in your solutions - as is done below. Use code comments for subsubparts. Coding style is very important. You will receive a deduction of up to 1.0 point if you do not adhere to good coding style. What I am looking for in terms of style includes:
    - appropriate variable use and naming
    - appropriate function use
    - good code commenting
    - consistent code syntax
- **Late Submission**: Late homework will not be accepted.

Please do not include the above Rubric, Instructions, and homework deadline sections in your solutions.

Introduction

The opioid epidemic is the deadliest drug overdose crisis in US history - on track to kill more people over the next ten years than currently live in entire cities like Baltimore or Miami. More people died last year from drug overdoses than American soldiers during the Vietnam War. In recent times, the opioid crisis has become an issue that the country can no longer ignore.

So far in 2018, anti-opioid ads on television have aired in congressional and gubernatorial races more than 50,000 times across 25 states, said The Wall Street Journal. At this point in 2014, Kentucky was the only state with political ads mentioning opioids that aired only 70 times. In other words, political ads against opioids have skyrocketed roughly 714 times when compared to this time in the 2014 midterm election cycle.

The above is an excerpt from an article, “Opioid Crisis Emerges As Top Campaign Theme For Midterms”.

Connecticut is one of the states with a high number of opiod deaths. For this assignment you will work with Connecticut’s open data on Accidental Drug Related Deaths 2012-2017. The website that hosts the data and other related resources is available here (https://data.ct.gov/Health-and-Human-Services/Accidental-Drug-Related-Deaths-2012-2017/rybz-nyjw). You will need to read the website first to understand the data and variables. There are some other interesting features on the website for data visualization. By the end of this course you will have the capabilities to replicate and expand on those visualizations.

You will need the following two packages for a plot to generate at the end of this assignment. Install the packages with `install.packages("dplyr")` and `install.packages("ggplot2")`. Do this in your console and do not include it in your .Rmd file. After installation you will want to load the packages with the `library` function. Include the below two lines of code as your second R chunk in your .Rmd file. Remember, a package only needs to be installed once, but each time you want to use it you need to load it into your workspace.

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

Below is code to read in the data. The `opiod` object is a data frame with 4083 rows and 32 columns. This should be your third R chunk.

```
opiod <- read.csv("https://data.ct.gov/api/views/rybz-nyjw/rows.csv?accessType=DOWNLOAD"
,
                  stringsAsFactors = FALSE)
names(opiod)
```

```
 [1] "CaseNumber"            "Date"
 [3] "Sex"                   "Race"
 [5] "Age"                   "Residence.City"
 [7] "Residence.State"       "Residence.County"
 [9] "Death.City"            "Death.State"
[11] "Death.County"          "Location"
[13] "DescriptionofInjury"   "InjuryPlace"
[15] "ImmediateCauseA"       "Heroin"
[17] "Cocaine"               "Fentanyl"
[19] "Oxycodone"             "Oxymorphone"
[21] "EtOH"                  "Hydrocodone"
[23] "Benzodiazepine"        "Methadone"
[25] "Amphet"                "Tramad"
[27] "Morphine..not.heroin." "Other"
[29] "Any.Opioid"            "MannerofDeath"
[31] "AmendedMannerofDeath"  "DeathLoc"
```

Question 1 - Data exploration and cleaning (6 points)

Part a (0.75 points)

1. Create a data frame called `opiod_dem` that contains all the rows of `opiod`, but only contains the columns `Sex`, `Race`, `Age`.
2. Compute the mean, median, and standard deviation for `Age` from `opiod_dem`.
3. Create a boxplot for the variable `Age` from `opiod_dem`. Include a label for the y-axis that reads, “Age at time of death”.

Part b (4 points)

The function `table` builds a contingency table of the counts at each combination of factor levels. Try it out with `table(opiod_dem$Sex)`. You should notice that for 4 of the deaths a value was not provided in the data for `Sex`. In the data set empty values are as “”.

1. Subset `opiod_dem` to remove the rows where the variable `Sex` has “” as a value. Save the resulting data frame as `opiod_dem_filter`.
2. Subset `opiod_dem_filter` to remove the rows where the variable `Race` has “” as a value. Save the resulting data frame as `opiod_dem_filter`.
3. Use the function `table` to build a table for the variable `Sex` from `opiod_dem_filter`.
4. Use the function `table` to create a two way contingency table for the variables `Race` and `Sex` from `opiod_dem_filter`. Save the resulting table as an object named `dem`.
5. Turn `dem` into a data frame with the following code:
   `dem_df <- data.frame(female = dem[, 1], male = dem[, 2])`.
6. Add a column named `sums` to `dem_df` that is the row sum.
7. From 6, calculate which race has the highest percentage of female deaths, and which race has the highest percentage of male deaths.
8. Repeat the calculation in 6, but only for races with a minimum of 100 deaths.

Part c (1.25 points)

You will now get back to working with the data frame object `opiod`.

1. Display the unique counties for the variable `Death.County`. You should observe that three counties do not make sense. One is “”, another is “USA”, and another is " FAIRFIELD" as opposed to “FAIRFIELD”, note the extra space. You will correct each of these issues in 2-4.
2. Update `opiod` so " FAIRFIELD" is “FAIRFIELD” for the variable `Death.County`.
3. Update `opiod` so “” is “NOT RECORDED” for the variable `Death.County`.
4. Update `opiod` so “USA” is “NOT RECORDED” for the variable `Death.County`.
5. Create a barplot with the function `barplot` for the number of deaths from the variable `Death.County`. As a hint, use the `table` function to generate counts for each county in `Death.County`. If you did steps 2-4 correctly, you should have nine categories. Include the graphical parameter `cex.names = 0.5` to shrink the county names so they fit in the plot window.
6. Update `opiod` so “” values for the variable `Race` are “Unknown”.

Question 2 - County summary function (3 points)

Create a function named `county.summary` that has two arguments: `df` and `county`. The `opiod` data frame will be passed to the argument `df`, and a county in Connecticut will be passed to the `county` argument as a character. See below for some examples of `county.summary` in action.

Your function, `county.summary`, should return a list with the following information:

- total deaths in the specified county,

- mean age of death in the specified county,
- median age of death in the specified county,
- a demographics data frame for county deaths based on race and gender.

An idea of how the function should work: the county name passed into the function should filter the data frame based on the variable `Death.County`; compute the necessary information from the filtered data frame; return the results in a list and in the format you see below. Include a check at the begining of your function that will give an error to the user if a county is entered that is not in Connecticut.

```
county.summary(df = opiod, county = "LITCHFIELD")
```

```
$total.deaths
[1] 187

$mean.age
[1] 40.1016

$median.age
[1] 41

$demographics

                  Female Male
  Asian, Other         0    1
  Black                1    0
  Hispanic, White      1    3
  White               53  128
```

```
county.summary(opiod, "HARTFORD")
```

```
$total.deaths
[1] 1022

$mean.age
[1] 42.34868

$median.age
[1] 43

$demographics

                  Unknown Female Male
  Asian Indian          0      1    2
  Asian, Other          0      1    3
  Black                 0     15   67
  Hispanic, Black       0      1    3
  Hispanic, White       1     24  132
  Other                 0      1    1
  Unknown               0      2    5
  White                 0    212  551
```

```
county.summary(opiod, "hartford")
```

```
Error in county.summary(opiod, "hartford"): County is not located in CT!
```

```
county.summary(opiod, "INGHAM")
```

```
Error in county.summary(opiod, "INGHAM"): County is not located in CT!
```

Question 3 Choropleth map (1 point)

Part a

1. Use the `opiod` data frame to create a data frame the contains two columns. Column 2 should contain the counts of total deaths in that county. Column 1 should contain the county's name. You should work with the `Death.County` variable.
2. Add the names "subregion" and "count" to the data frame in 1.
3. Remove the "NOT RECORDED" row from your data frame.
4. Pass your data frame into the function `ct.choropleth` given below. Explain ways to improve the plot to make it more informative.

```
ct.choropleth <- function(df){

  # generate county and state boundaries
  ct.state <- map_data("state", region = "connecticut")
  ct.county.df <- map_data("county", region = "connecticut")

  # convert county names to lower case
  county.df <- mutate_all(df, funs(tolower))

  # merge data frames to pass a single data frame to ggplot
  choropleth <- inner_join(ct.county.df, county.df, by = "subregion")

  # convert counts to type numeric
  choropleth$count <- as.numeric(choropleth$count)

  # generate choropleth
  ct.plot <- ggplot(choropleth, aes(long, lat, group = group)) +
    geom_polygon(aes(fill = count), alpha = 0.75, color = "white") +
    geom_polygon(data = ct.county.df, colour = "white", fill = NA) +
    geom_polygon(data = ct.state, color = "black", fill = NA)+
    scale_fill_gradient2(low = "yellow", mid = "orange", high = "red") +
    ggtitle("Opiod deaths in Connecticut by county") +
    labs(fill = "Deaths") +
    theme_void()

  return(ct.plot)
}
```

Student Feedback

This part is optional. It will be available on all assignments. Feel free to answer on all, some, or none.

1. How is the course going for you in terms of pace and/or difficulty level?
2. Do you have any concerns or suggestions?
3. Please provide any other comments you may have.