Homework 4: Introduction To Tidyverse

Shreya Rao sr3843

June 1, 2021

```
knitr::opts_chunk$set(echo = TRUE)
library(formatR)
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60),tidy=TRUE)
```

Instructions

Please submit both pdf and Rmd files (or html and Rmd files).

Part II: Split/Apply/Combine and tidyverse warm-up

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

Consider the following **loop** that computes the mean of each quantitative variable split by species and stores the computed means in a matrix named **MeanFlowers**.

```
# define a matrix of zeros
MeanFlowers <- matrix(0, nrow = 4, ncol = 3)

# define a character vector corresponding to the numeric
# variable names
measurements <- c("Sepal.Length", "Sepal.Width", "Petal.Length",
    "Petal.Width")

# name the rows and columns of the matrix MeanFlowers
rownames(MeanFlowers) <- measurements
colnames(MeanFlowers) <- c("setosa", "versicolor", "virginica")

# Loop
for (j in measurements) {
    #-- R code goes here ----
    MeanFlowers[j, ] <- round(tapply(iris[, j], iris[, "Species"],
        mean), 4)
}
MeanFlowers
```

```
##              setosa versicolor virginica
## Sepal.Length  5.006      5.936     6.588
## Sepal.Width   3.428      2.770     2.974
## Petal.Length  1.462      4.260     5.552
## Petal.Width   0.246      1.326     2.026
```

Problem 1

Replicate the above loop using the **Split/Apply/Combine** model with base R commands.

```
mean_species <- function(df) {
    return(apply(df[, c("Sepal.Length", "Sepal.Width", "Petal.Length",
        "Petal.Width")], 2, mean))
}

species <- split(iris, iris$Species)
sapply(species, mean_species)
```

```
##              setosa versicolor virginica
## Sepal.Length  5.006      5.936     6.588
## Sepal.Width   3.428      2.770     2.974
## Petal.Length  1.462      4.260     5.552
## Petal.Width   0.246      1.326     2.026
```

Problem 2

Repeat question 1 by constructing a pipe, including the **split()** function from base R and **map_df()** from the **purrr** package.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(purrr)
iris %>%
    split(.$Species) %>%
    map_df(mean_species, .id = "var")
```

```
## # A tibble: 3 x 5
##   var        Sepal.Length Sepal.Width Petal.Length Petal.Width
##   <chr>             <dbl>       <dbl>        <dbl>       <dbl>
## 1 setosa             5.01        3.43         1.46       0.246
## 2 versicolor         5.94        2.77         4.26       1.33
## 3 virginica          6.59        2.97         5.55       2.03
```

Part II: More tidyverse with CDC cancer data

Consider the Center of Disease Control data set **BYSITE_new.csv**, which describes the incidence and mortality counts of several types of cancer over time. The variables of interest are: **YEAR**, **RACE**, **SITE**, **EVENT_TYPE**, **COUNT** and **POPULATION**.

Problem 3

Load in the dataset **BYSITE_new.csv** using the appropriate function from the **readr** package. Display the dimension of the cancer tibble.

```
library(readr)
cancer <- read_csv("BYSITE_new.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   YEAR = col_character(),
##   RACE = col_character(),
##   SEX = col_character(),
##   SITE = col_character(),
##   EVENT_TYPE = col_character(),
##   COUNT = col_character(),
##   POPULATION = col_double()
## )
```

```
dim(cancer)
```

```
## [1] 44982     7
```

```
cancer_rows <- nrow(cancer)
```

Base R code for reference.

```
# Base R code for reference cancer <-
# read.csv('BYSITE_new.csv',header=T) dim(cancer)
```

Problem 4

Using Base R or tydyverse functions, identify any strange symbols that are recorded in the **COUNT** variable. Once you have identified the symbols, use functions from the **dplyr** package to remove any rows in the cancer tibble containing these symbols and then convert **COUNT** to a numeric mode.

```
library(stringr)
# identifying rows in COUNT with symbols
problems <- cancer$COUNT[grepl(pattern = "[^0-9]", cancer$COUNT)]
head(problems)
```

```
## [1] "." "." "." "." "." "."
```

```r
length(problems)
```

```
## [1] 6828
```

```r
# replace all symbols with ' '
cancer$COUNT <- str_replace(cancer$COUNT, pattern = "[^0-9]",
    replacement = " ")

# remove rows with symbols
cancer <- cancer %>%
    filter(COUNT != " ")

# calculating number of rows in new cancer df
new_rows <- nrow(cancer)
new_rows
```

```
## [1] 38154
```

```r
# new cancer rows must be equal to (old rows - problem rows)
cancer_rows - length(problems)
```

```
## [1] 38154
```

```r
# converting COUNT to numeric type
cancer$COUNT <- as.numeric(cancer$COUNT)

head(cancer)
```

```
## # A tibble: 6 x 7
##   YEAR  RACE      SEX    SITE              EVENT_TYPE COUNT POPULATION
##   <chr> <chr>     <chr>  <chr>             <chr>      <dbl>      <dbl>
## 1 1999  All Races Female Acute Lymphocytic Incidence   1647  139034769
## 2 1999  All Races Female Acute Lymphocytic Mortality    582  142237295
## 3 2000  All Races Female Acute Lymphocytic Incidence   1777  140494755
## 4 2000  All Races Female Acute Lymphocytic Mortality    591  143719004
## 5 2001  All Races Female Acute Lymphocytic Incidence   1843  143603977
## 6 2001  All Races Female Acute Lymphocytic Mortality    644  145077463
```

Problem 5

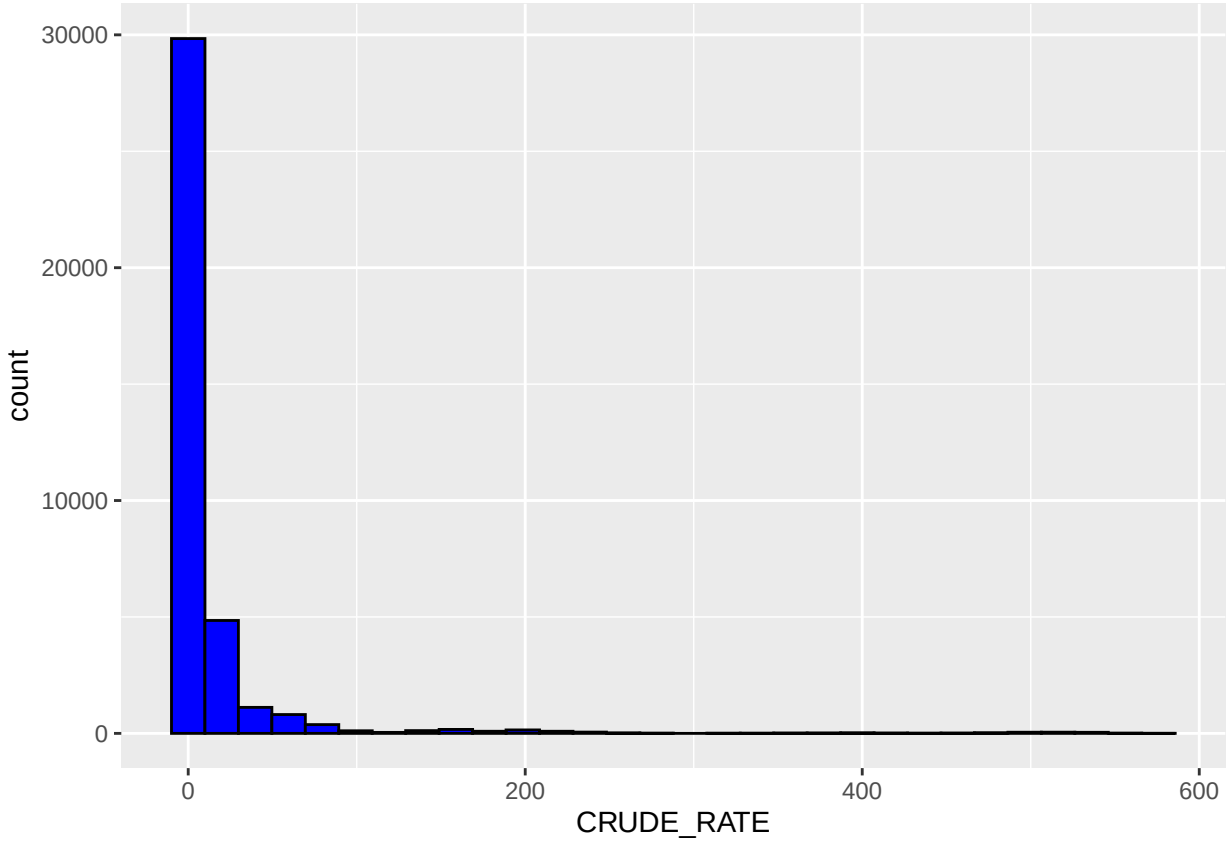
For a specific tumor and population, a crude rate is calculated by dividing the number of new cancers observed during a given time period by the corresponding number of people in the population at risk. For cancer, the result is usually expressed as an annual rate per 100,000 persons at risk. https://ci5.iarc.fr/ci5plus/pages/glossary.aspx

In reference to our data, this quantity can be calculated by:

$$\text{CRUDE RATE} = 100000 * \frac{\text{COUNT}}{\text{POPULATION}}$$

Using relevant functions from the **dplyr** package, create a new variable in your dataframe (or tibble) called **CRUDE_RATE**. Then using base R graphics or ggplot, create a histogram of **CRUDE_RATE**. Note that the crude rates are not bounded between [0,1] because they are calculated per 100,000 persons at risk.

```r
cancer <- cancer %>%
    mutate(CRUDE_RATE = 1e+05 * COUNT/as.numeric(POPULATION))

cancer %>%
    ggplot() + geom_histogram(aes(CRUDE_RATE), bins = 30, col = 1,
    fill = "blue")
```



Problem 6

Compute the average incidence rate of prostate cancer for each level of **RACE**. To solve this problem, students must build a pipe (**magrittr** package) and utilize the appropriate functions from the **dplyr** package. Also compare your results to a base R solution. Include both the tidyverse and base R solutions in your final write-up. **Note:** before computing the average incidence rates, students should filter the data as follows:

i. Extract the rows corresponding to **EVENT_TYPE** level **Incidence**

ii. Extract the rows corresponding to **SITE** level **Prostate**

iii. Extract the rows corresponding to **SEX** level **Male**

iv. Remove the rows corresponding to **YEAR** level **2010-2014**

v. Remove the rows corresponding to **RACE** level **All Races**

First filter the dataset:

```
# Base R
filter_baseR <- cancer[cancer$EVENT_TYPE == "Incidence" & cancer$SITE ==
    "Prostate" & cancer$SEX == "Male" & cancer$YEAR != "2010-2014" &
    cancer$RACE != "All Races", ]

# dplyr
filter_dp <- cancer %>%
    filter(EVENT_TYPE == "Incidence", SITE == "Prostate", SEX ==
        "Male", YEAR != "2010-2014", RACE != "All Races")
```

Compute the average incidence rate of prostate cancer for each level of **RACE**.

```
# Base R
mean_race_baseR <- tapply(filter_baseR$CRUDE_RATE, filter_baseR$RACE,
    mean)
mean_race_baseR
```

```
## American Indian/Alaska Native        Asian/Pacific Islander
##                       41.93982                       51.99519
##                          Black                       Hispanic
##                      152.72852                       53.92810
##                          White
##                      142.85196
```

```
# tidyverse
mean_race_dp <- filter_dp %>%
    group_by(RACE) %>%
    summarise(avg_incidence_rate = mean(CRUDE_RATE))
mean_race_dp
```

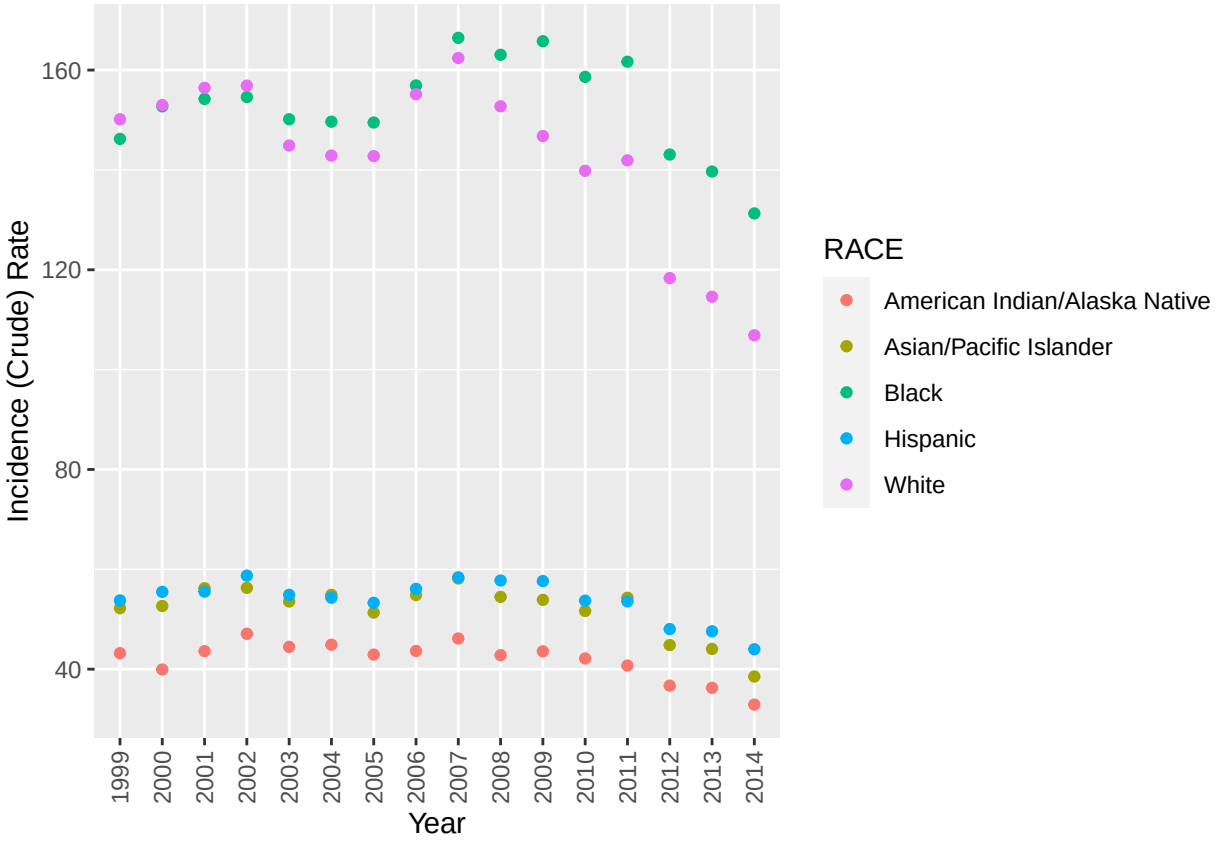
```
## # A tibble: 5 x 2
##   RACE                           avg_incidence_rate
##   <chr>                                       <dbl>
## 1 American Indian/Alaska Native               41.9
## 2 Asian/Pacific Islander                      52.0
## 3 Black                                      153.
## 4 Hispanic                                    53.9
## 5 White                                      143.
```

Problem 7

Create a plot in base R or ggplot that shows the incidence rate (**CRUDE_RATE**) as a function of time (**YEAR**), split by the levels of **RACE**. Make sure to include a legend and label the graphic appropriately. Before constructing the graphic, perform the data wrangling tasks using a pipe and functions from the **dplyr** package, i.e., the same filtering tasks from problem 6. Students can use some base R functions in the pipe if needed and the plotting code can be included inside or outside the pipe.

```
graph_base <- cancer %>%
    filter(EVENT_TYPE == "Incidence", SITE == "Prostate", SEX ==
        "Male", YEAR != "2010-2014", RACE != "All Races") %>%
    ggplot() + geom_point(aes(x = YEAR, y = CRUDE_RATE, col = RACE)) +
```

```
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5,
        hjust = 1)) + labs(y = "Incidence (Crude) Rate", x = "Year")
graph_base
```



Problem 8

Fit five simple linear regression models, one for each level of **RACE**, relating the incidence rate (**CRUDE_RATE**) as a function of time (**YEAR**). Collect the estimated slopes, t-statistics and p-values of your estimated models. The collection of slopes describe whether cancer has increased or decreased over the selected time period and the p-values describe if the increase or decrease is statistically significant. Solve this problem using a pipe and functions from the **dplyr** and **purrr** packages. **Note:** use the same filtered data from problem 4 and problem 4 in this analysis.

Some hints: (i) this exercise is a natural extension of problem 7; (ii) if needed, students can also define their own functions used in the pipe; (iii) students are not required to use a single pipe to solve this question but it's a fun challenge if interested.

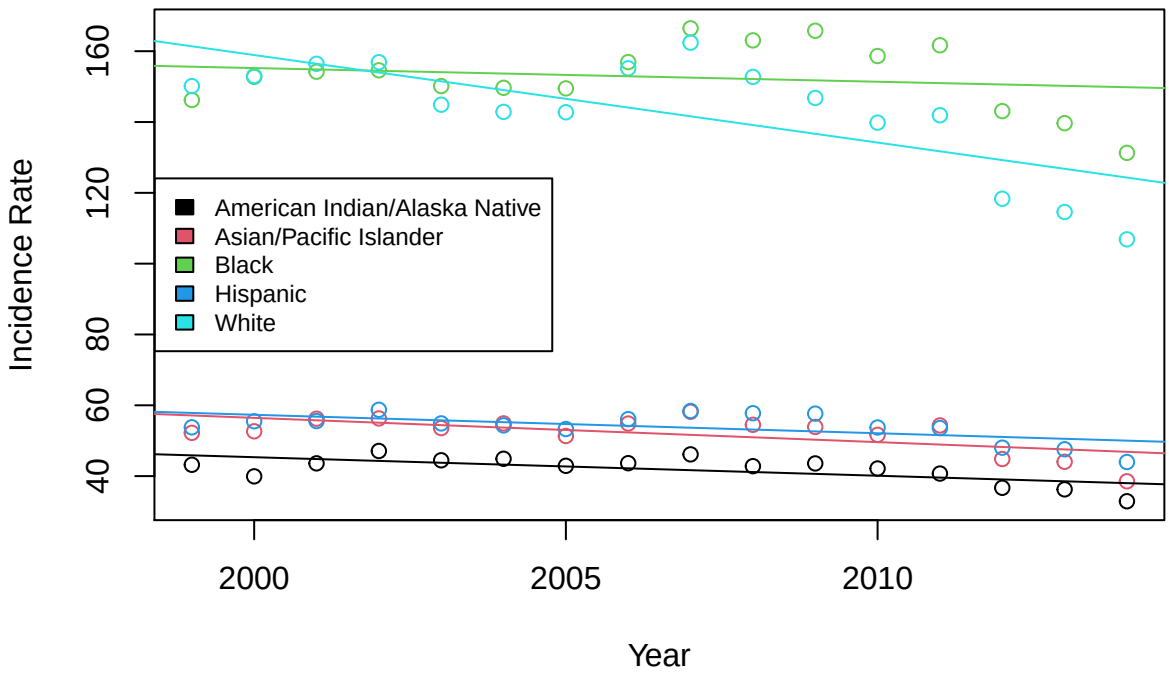
```
summaries <- filter_dp %>%
    split(.$RACE) %>%
    map(~lm(CRUDE_RATE ~ as.numeric(YEAR), data = .)) %>%
    map(summary) %>%
    map(coefficients)

lm_slopes <- matrix(ncol = 3, nrow = 5)
for (i in 1:length(summaries)) {
```

```
    lm_slopes[i, ] <- summaries[[i]][2, c(1, 3, 4)]
}
rownames(lm_slopes) <- names(summaries)
colnames(lm_slopes) <- c("Slope Estimate", "t value", "p-value")
lm_slopes
```

```
##                                Slope Estimate    t value     p-value
## American Indian/Alaska Native     -0.5237751 -3.2369333 0.005965782
## Asian/Pacific Islander            -0.6844835 -3.0127263 0.009313557
## Black                             -0.3857198 -0.7256263 0.480022889
## Hispanic                          -0.5212330 -2.8040621 0.014067392
## White                             -2.4717620 -4.0014809 0.001312233
```

```
# The slopes all indicate that cancer rates have decreased
# over the years, with rates in the White group showing the
# steepest slope.  The p-values indicate that the decrease is
# statistically significant for all races expect for that of
# Black. (assuming the significance level alpha is set at
# 0.05)
```

```
# plot of regression lines
legend <- rep(NA, 5)
plot(y = filter_baseR$CRUDE_RATE, x = filter_baseR$YEAR, col = as.factor(filter_baseR$RACE),
    xlab = "Year", ylab = "Incidence Rate")
for (i in 1:length(summaries)) {
    abline(a = summaries[[i]][1, 1], b = summaries[[i]][2, 1],
        col = i)
    legend[i] <- names(summaries)[i]
}
legend("left", legend = legend, fill = unique(as.factor(filter_baseR$RACE)),
    cex = 0.75)
```


Incidence Rate
160
120
80
60
40
American Indian/Alaska Native
Asian/Pacific Islander
Black
Hispanic
White
2000
2005
2010
Year