Homework 4

Maxime Grossman (UNI: mmg2240)

6/10/2021

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

Problem 1

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

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.6.3
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
library(readr)
```

```
iris.split <- split(iris, iris$Species)


three.mean <- function(df){

  return(apply(df[, c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")], 2, mean))

}

iris.split.mean <- sapply(iris.split, three.mean)
```

```
iris.split.mean
```

```
##              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(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:plyr':
##
##     compact
```

```
Measurement <- c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")

iris %>%
  split(iris$Species) %>%
  map_df(three.mean) %>%
  cbind(Measurement,.)
```

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

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.

Base R code for reference.

```
# Base R code for reference
cancer <- read.csv("BYSITE_new.csv",header=T)
dim(cancer)
```

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

Solution goes below

```
cancer <- read_csv("BYSITE_new.csv", col_names = TRUE)
```

```
## Parsed with 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()
## )
```

```
cancer <- as_tibble(cancer)

head(cancer)
```

```
## # A tibble: 6 x 7
##   YEAR  RACE      SEX    SITE              EVENT_TYPE COUNT POPULATION
##   <chr> <chr>     <chr>  <chr>             <chr>      <chr>      <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
```

```
dim(cancer)
```

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

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.

```
head(table(cancer$COUNT), 50)
```

```
##
##      .      ~    100   1000 100008  10002  10004 100043  10008 100099   1001
##    633   6195     59      3      1      1      1      1      1      1      5
## 100126  10013  10019   1002  10022  10028   1003  10032  10035 100386   1004
##      1      2      1      6      2      1      7      1      1      1      8
##  10042  10045  10047  10048   1005  10050 100516  10054  10057 100590   1006
##      2      2      2      1      4      1      1      1      2      1      8
##  10062  10064  10065  10066  10068  10069   1007  10071  10072  10074 100743
##      1      1      1      1      1      1      4      1      1      1      1
##  10076  10079   1008  10080  10081 100849
##      1      2      6      2      2      1
```

```r
tail(table(cancer$COUNT), 50)
```

```
##
##   990  9904 99068  9907  9908 99086   991  9918   992  9921  9922  9923  9924
##     2     1     1     1     2     1     4     1     1     1     2     1     1
##  9926 99266  9927 99279  9929 99293   993 99360  9939   994  9941 99410  9942
##     1     1     1     1     1     1     4     1     1     6     1     1     1
##   995  9951 99514  9956  9958  9959   996   997  9973  9975  9976  9978   998
##     1     2     1     2     2     1     5     5     1     1     1     1     6
##  9983 99843  9985  9986  9987  9989   999 99914  9992  9998  9999
##     1     1     1     1     2     2     4     1     1     1     1
```

```r
# select(cancer, COUNT)

dim(cancer)
```

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

```r
#cancer %>%
#  filter(COUNT >= 0)

#cancer %>%
#  filter(COUNT == "." | COUNT == "~")

#cancer %>%
 # filter(COUNT != "." | COUNT != "~")


#cancer %>%
 # which(COUNT == ".")


v1 <- which(cancer$COUNT == ".")
v2 <- which(cancer$COUNT == "~")

v3 <- c(v1,v2)


length(v1)
```

```
## [1] 633
```

```r
length(v2)
```

```
## [1] 6195
```

```r
length(v3)
```

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

```r
cancer <- cancer[-v3,]

dim(cancer)
```

```
## [1] 38154     7
```

```r
44982-(633+6195)
```

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

```
cancer$COUNT <- as.numeric(cancer$COUNT)

is.numeric(cancer$COUNT)
```

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

We first use the table function to look at the summarized data. Immediately we notice that there are 633 entries with a "." as COUNT, and 6,195 entries with a "~" as COUNT. Note: 633 + 6,195 = 6,828.

We can use the which() function to figure out which rows contain these two symbols. After deleting these rows, we find the length of our new object, which is now 38,154.

This length corroborates perfectly with our observation because starting out with 44,982 rows, if we subtract (633 + 6,195) rows, we end up with 38,154 rows.

As a double check, we force the remaining values as numeric with as.numeric(), and do a check using is.numeric() on the new object, which returns TRUE. All values are now integers and there are no strange characters.

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

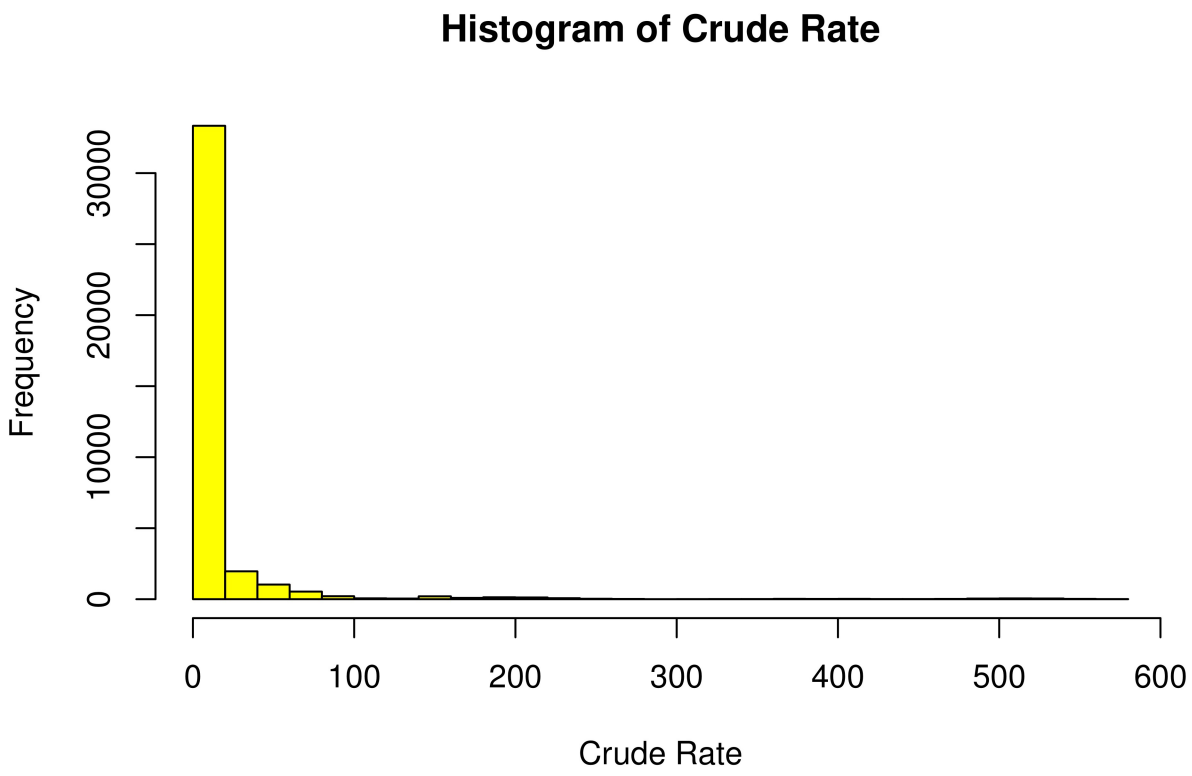
In reference to our data, this quantity can be calculated by:

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

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.

```
cancer <- cancer %>%
  mutate(CRUDE_RATE = 100000*COUNT/POPULATION)

hist(cancer$CRUDE_RATE, breaks=30, main = "Histogram of Crude Rate", xlab = "Crude Rate", col="yellow")
```



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**

##**Solution goes below**

First we do it via dplyr:

```
#levels(factor(cancer$RACE))

#levels(factor(cancer$SITE))

head(cancer)
```

```
## # A tibble: 6 x 8
```

```
##   YEAR  RACE      SEX    SITE             EVENT_TYPE COUNT POPULATION CRUDE_RATE
##   <chr> <chr>     <chr>  <chr>            <chr>      <dbl>      <dbl>      <dbl>
## 1 1999  All Races Female Acute Lymphocyt~ Incidence   1647  139034769      1.18
## 2 1999  All Races Female Acute Lymphocyt~ Mortality    582  142237295      0.409
## 3 2000  All Races Female Acute Lymphocyt~ Incidence   1777  140494755      1.26
## 4 2000  All Races Female Acute Lymphocyt~ Mortality    591  143719004      0.411
## 5 2001  All Races Female Acute Lymphocyt~ Incidence   1843  143603977      1.28
## 6 2001  All Races Female Acute Lymphocyt~ Mortality    644  145077463      0.444
```

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

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

```
cancer.filter
```

```
## # A tibble: 80 x 8
##    YEAR  RACE                SEX   SITE   EVENT_TYPE COUNT POPULATION CRUDE_RATE
##    <chr> <chr>               <chr> <chr>  <chr>      <dbl>      <dbl>      <dbl>
##  1 1999  American Indian/Al~ Male  Prost~ Incidence    590    1365653       43.2
##  2 2000  American Indian/Al~ Male  Prost~ Incidence    577    1444839       39.9
##  3 2001  American Indian/Al~ Male  Prost~ Incidence    671    1538799       43.6
##  4 2002  American Indian/Al~ Male  Prost~ Incidence    749    1590329       47.1
##  5 2003  American Indian/Al~ Male  Prost~ Incidence    735    1652897       44.5
##  6 2004  American Indian/Al~ Male  Prost~ Incidence    769    1712685       44.9
##  7 2005  American Indian/Al~ Male  Prost~ Incidence    762    1775711       42.9
##  8 2006  American Indian/Al~ Male  Prost~ Incidence    805    1844390       43.6
##  9 2007  American Indian/Al~ Male  Prost~ Incidence    885    1917949       46.1
## 10 2008  American Indian/Al~ Male  Prost~ Incidence    855    1997376       42.8
## # ... with 70 more rows
```

```
# average incidence rate

cancer.filter %>%
  split(.$RACE) %>%
  map(~ mean(.$CRUDE_RATE)/100000)
```

```
## $`American Indian/Alaska Native`
## [1] 0.0004193982
##
## $`Asian/Pacific Islander`
## [1] 0.0005199519
##
## $Black
## [1] 0.001527285
##
## $Hispanic
## [1] 0.000539281
##
## $White
## [1] 0.00142852
```

```r
# average incidence rate per 100,000 people

cancer.filter %>%
  split(.$RACE) %>%
  map(~ mean(.$CRUDE_RATE))
```

```
## $`American Indian/Alaska Native`
## [1] 41.93982
##
## $`Asian/Pacific Islander`
## [1] 51.99519
##
## $Black
## [1] 152.7285
##
## $Hispanic
## [1] 53.9281
##
## $White
## [1] 142.852
```

Since the crude rate is the count divided by the population times 100,000, this value is essentially the incidence rate but scaled up by 100,000. We simply take the mean of this value per each race and divide back by 100,000 to get the average incidence rate.

If we want the average incidence rate per 100,000 people, we take the average of the crude rate per each race.

Now we do it via Base R:

```r
# Base R

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

cancer.split <- split(cancer, cancer$RACE)

mean1 <- function(df){
  mean(df$CRUDE_RATE)
}

# We can use either of these options:

sapply(cancer.split, mean1)
```

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

```
ddply(cancer, .(RACE), mean1)
```

```
##                             RACE        V1
## 1 American Indian/Alaska Native  41.93982
## 2        Asian/Pacific Islander  51.99519
## 3                         Black 152.72852
## 4                      Hispanic  53.92810
## 5                         White 142.85196
```

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.

Solution goes below

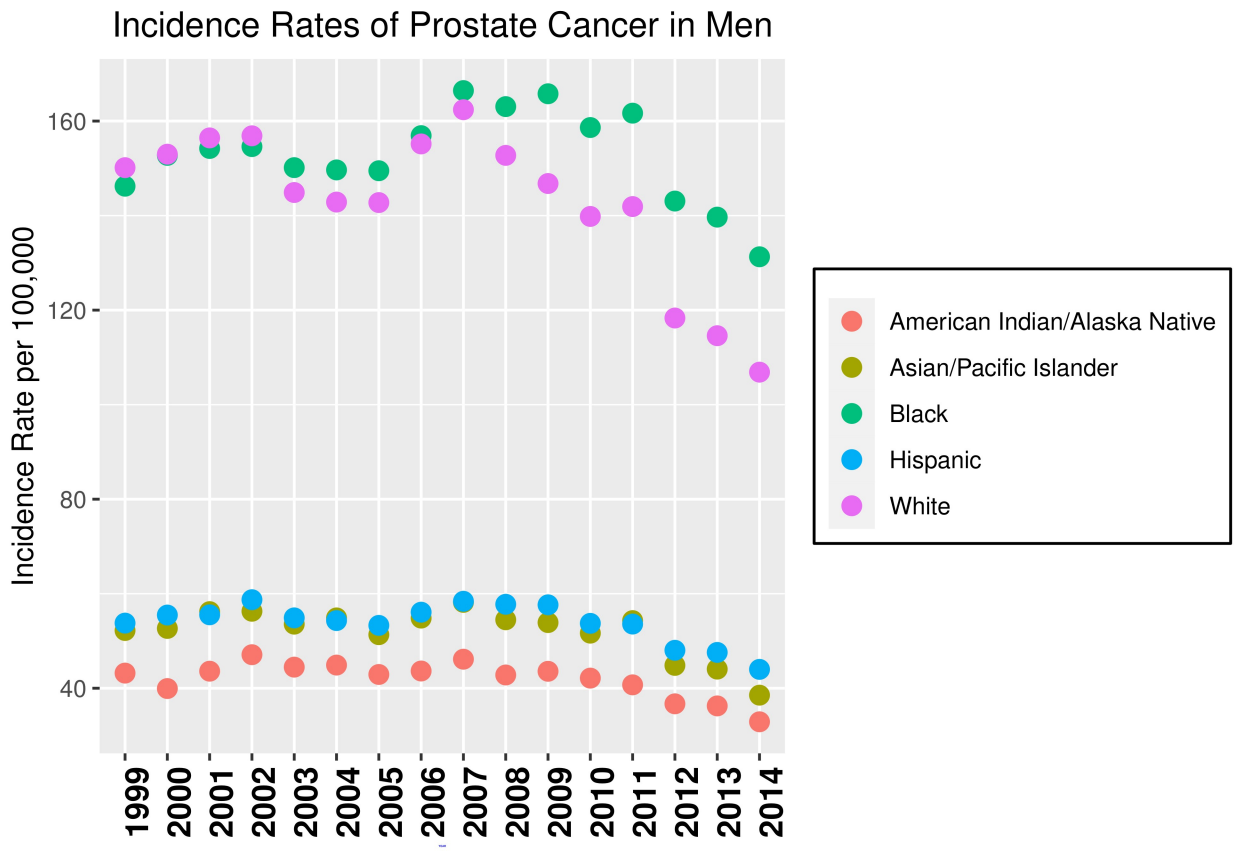
```
# The results of the same data wrangling tasks from problem 6 were
# stored as our variable cancer.filter

head(cancer.filter)
```

```
## # A tibble: 6 x 8
##   YEAR  RACE                 SEX   SITE   EVENT_TYPE COUNT POPULATION CRUDE_RATE
##   <chr> <chr>                <chr> <chr>  <chr>      <dbl>      <dbl>       <dbl>
## 1 1999  American Indian/Ala~ Male  Prost~ Incidence    590    1365653        43.2
## 2 2000  American Indian/Ala~ Male  Prost~ Incidence    577    1444839        39.9
## 3 2001  American Indian/Ala~ Male  Prost~ Incidence    671    1538799        43.6
## 4 2002  American Indian/Ala~ Male  Prost~ Incidence    749    1590329        47.1
## 5 2003  American Indian/Ala~ Male  Prost~ Incidence    735    1652897        44.5
## 6 2004  American Indian/Ala~ Male  Prost~ Incidence    769    1712685        44.9
```

```
# plot(cancer.filter$YEAR, cancer.filter$CRUDE_RATE, col = "RACE") ### USE THIS

ggplot(cancer.filter) +
  geom_point(mapping = aes(x=YEAR, y = CRUDE_RATE, color = RACE), size = 3)+
  labs(title = "Incidence Rates of Prostate Cancer in Men", y = "Incidence Rate per 100,000") +
  theme(plot.title = element_text(hjust=0.5)) +
  theme(legend.background = element_rect(fill = "white", color = "black", linetype = 1), legend.title =
  theme(axis.text.x = element_text(face="bold", color="black", size=12, angle=90)) +
  theme(axis.title.x = element_text(color = "blue", size = 1, face = "bold"))
```



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.

Solution goes below

```
cancer.lm <- function(df){
  return(coef(lm(CRUDE_RATE ~ YEAR, data = df)))

}

cancer.filter %>%
```

```
  split(.$RACE) %>%
  map(cancer.lm)
```

```
## $`American Indian/Alaska Native`
## (Intercept)    YEAR2000    YEAR2001    YEAR2002    YEAR2003    YEAR2004
##  43.2027755  -3.2675301   0.4026596   3.8943975   1.2646051   1.6974834
##    YEAR2005    YEAR2006    YEAR2007    YEAR2008    YEAR2009    YEAR2010
##  -0.2903872   0.4430911   2.9402658  -0.3966138   0.3796624  -1.0567463
##    YEAR2011    YEAR2012    YEAR2013    YEAR2014
##  -2.4823210  -6.4986435  -6.9386349 -10.2985207
##
## $`Asian/Pacific Islander`
## (Intercept)    YEAR2000    YEAR2001    YEAR2002    YEAR2003    YEAR2004
##  52.2164629   0.4388911   3.9916236   4.0729087   1.3406037   2.6663844
##    YEAR2005    YEAR2006    YEAR2007    YEAR2008    YEAR2009    YEAR2010
##  -0.8850718   2.6270152   5.9536182   2.2695056   1.6771809  -0.5579489
##    YEAR2011    YEAR2012    YEAR2013    YEAR2014
##   2.0952340  -7.3828442  -8.1664599 -13.6810314
##
## $Black
## (Intercept)    YEAR2000    YEAR2001    YEAR2002    YEAR2003    YEAR2004
##  146.218494    6.523624    7.988680    8.381552    3.932762    3.444570
##    YEAR2005    YEAR2006    YEAR2007    YEAR2008    YEAR2009    YEAR2010
##    3.276701   10.707239   20.235143   16.843757   19.560415   12.417749
##    YEAR2011    YEAR2012    YEAR2013    YEAR2014
##   15.455377   -3.138004   -6.540807  -14.928396
##
## $Hispanic
## (Intercept)    YEAR2000    YEAR2001    YEAR2002    YEAR2003    YEAR2004
## 53.78091892  1.70647981  1.75629739  4.95374773  1.10741858  0.53830471
##    YEAR2005    YEAR2006    YEAR2007    YEAR2008    YEAR2009    YEAR2010
## -0.48313806  2.31036672  4.61668200  3.98011569  3.86805858 -0.05307447
##    YEAR2011    YEAR2012    YEAR2013    YEAR2014
## -0.20130531 -5.75299061 -6.20196842 -9.79015823
##
## $White
## (Intercept)    YEAR2000    YEAR2001    YEAR2002    YEAR2003    YEAR2004
##  150.143775    2.846938    6.299292    6.743226   -5.254714   -7.276712
##    YEAR2005    YEAR2006    YEAR2007    YEAR2008    YEAR2009    YEAR2010
##   -7.386155    5.025250   12.266237    2.598463   -3.353652  -10.318130
##    YEAR2011    YEAR2012    YEAR2013    YEAR2014
##   -8.228553  -31.809051  -35.552296  -43.269103
```

```
cancer.filter$YEAR <- as.numeric(cancer.filter$YEAR)

cancer.summ <- cancer.filter %>%
  split(.$RACE) %>%
  map(~lm(CRUDE_RATE ~ YEAR, .)) %>%
  map(summary)

# slopes

slopes <- NA
```

```r
slopes[1] <- coef(cancer.summ$`American Indian/Alaska Native`)[2,1]
slopes[2] <- coef(cancer.summ$`Asian/Pacific Islander`)[2,1]
slopes[3] <- coef(cancer.summ$Black)[2,1]
slopes[4] <- coef(cancer.summ$Hispanic)[2,1]
slopes[5] <- coef(cancer.summ$White)[2,1]


print("Slopes for American Indian/Alaska Native, Asian/Pacific Islander, Black, Hispanic, White:")
```

```
## [1] "Slopes for American Indian/Alaska Native, Asian/Pacific Islander, Black, Hispanic, White:"
```

```r
slopes
```

```
## [1] -0.5237751 -0.6844835 -0.3857198 -0.5212330 -2.4717620
```

```r
# p-value

p.values <- NA


p.values[1] <- coef(cancer.summ$`American Indian/Alaska Native`)[2,4]
p.values[2] <- coef(cancer.summ$`Asian/Pacific Islander`)[2,4]
p.values[3] <- coef(cancer.summ$`Black`)[2,4]
p.values[4] <- coef(cancer.summ$`Hispanic`)[2,4]
p.values[5] <- coef(cancer.summ$`White`)[2,4]



print("Slopes for American Indian/Alaska Native, Asian/Pacific Islander, Black, Hispanic, White:")
```

```
## [1] "Slopes for American Indian/Alaska Native, Asian/Pacific Islander, Black, Hispanic, White:"
```

```r
p.values
```

```
## [1] 0.005965782 0.009313557 0.480022889 0.014067392 0.001312233
```

```r
# t-statistic

t.stats <- NA

t.stats[1] <- coef(cancer.summ$`American Indian/Alaska Native`)[2,3]
t.stats[2] <- coef(cancer.summ$`Asian/Pacific Islander`)[2,3]
t.stats[3] <- coef(cancer.summ$`Black`)[2,3]
t.stats[4] <- coef(cancer.summ$`Hispanic`)[2,3]
t.stats[5] <- coef(cancer.summ$`White`)[2,3]



print("Slopes for American Indian/Alaska Native, Asian/Pacific Islander, Black, Hispanic, White:")
```

```
## [1] "Slopes for American Indian/Alaska Native, Asian/Pacific Islander, Black, Hispanic, White:"
```

```r
t.stats
```

```
## [1] -3.2369333 -3.0127263 -0.7256263 -2.8040621 -4.0014809
```