

ADA Homework 2

Samantha Rabinowitz, sar4357

3/3/2020

Challenge 1

The R code below will load the 'movies.csv' dataset from GitHub and organize the data into a tibble.

```
library(tidyverse)

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

## v ggplot2 3.2.1      v purrr  0.3.3
## v tibble  2.1.3      v dplyr  0.8.3
## v tidyr   1.0.0      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.4.0

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

f <- "https://raw.githubusercontent.com/difiore/ADA-datasets/master/IMDB-movies.csv"
d <- read_csv(f, col_names = T)

## Parsed with column specification:
## cols(
##   tconst = col_character(),
##   titleType = col_character(),
##   primaryTitle = col_character(),
##   startYear = col_double(),
##   runtimeMinutes = col_double(),
##   genres = col_character(),
##   averageRating = col_double(),
##   numVotes = col_double(),
##   nconst = col_character(),
##   director = col_character()
## )

glimpse(d)

## Observations: 28,938
## Variables: 10
## $ tconst      <chr> "tt0002130", "tt0002844", "tt0003037", "tt0003165", ...
```

```
## $ titleType      <chr> "movie", "movie", "movie", "movie", "movie", "movie"...
## $ primaryTitle   <chr> "Dante's Inferno", "Fantômas: In the Shadow of the G...
## $ startYear      <dbl> 1911, 1913, 1913, 1913, 1913, 1914, 1914, 1914, 1914...
## $ runtimeMinutes <dbl> 68, 54, 61, 90, 85, 78, 148, 59, 61, 82, 195, 59, 72...
## $ genres         <chr> "Adventure,Drama,Fantasy", "Crime,Drama", "Crime,Dra...
## $ averageRating  <dbl> 7.0, 7.0, 7.0, 7.0, 6.5, 6.5, 7.1, 6.9, 6.2, 6.3, 6....
## $ numVotes       <dbl> 2082, 1877, 1307, 1010, 1686, 1068, 2907, 1126, 1207...
## $ nconst         <chr> "nm0078205", "nm0275421", "nm0275421", "nm0275421", ...
## $ director       <chr> "Francesco Bertolini", "Louis Feuillade", "Louis Feu..."
```

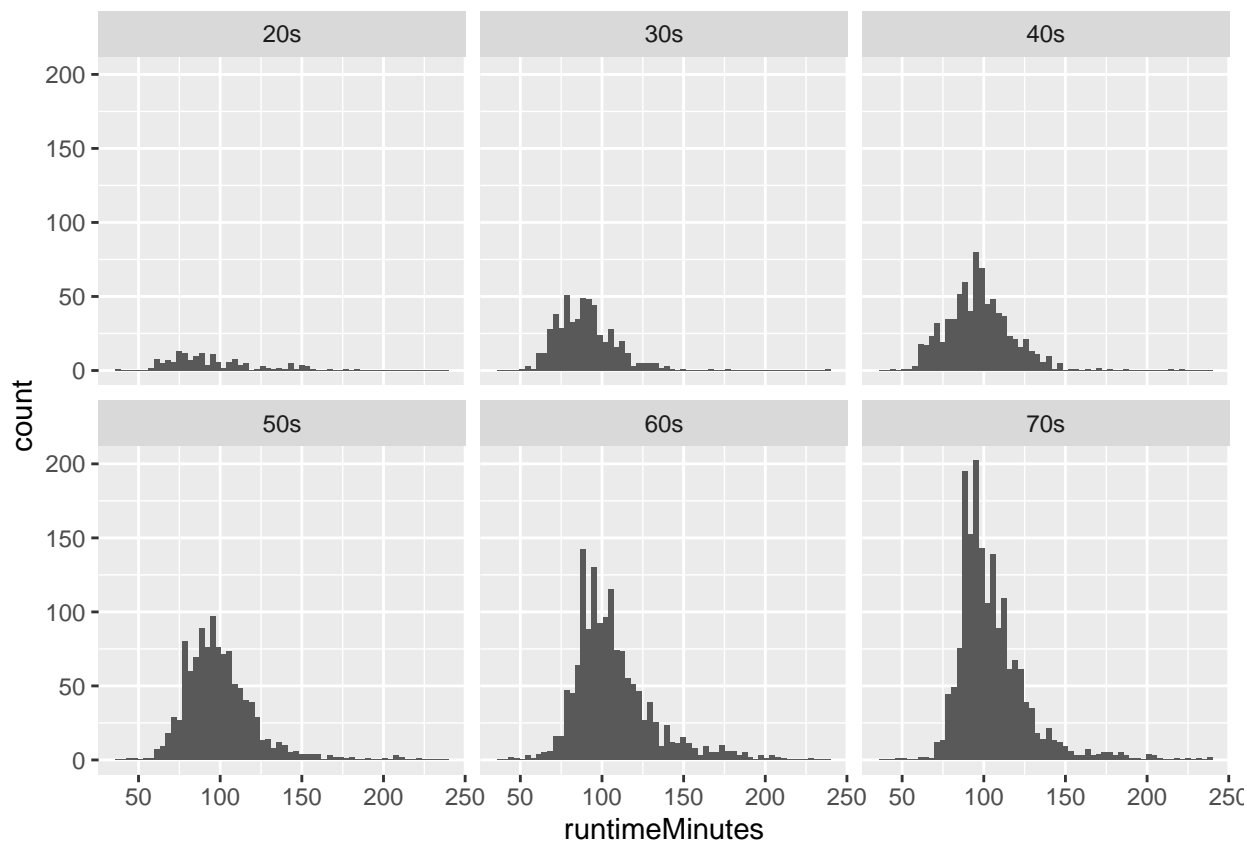
The following code will filter the dataset to just include movies from 1920 to 1979 and movies that are less than 4 hours long. Columns were also added to make **startYear** a new variable called **decade**.

```
d1 <- d %>%
  filter(startYear >= "1920" & startYear <="1979" & runtimeMinutes < 240) %>%
  mutate(decade = case_when(startYear >=1920 & startYear<=1929 ~ "20s",
                             startYear >=1930 & startYear<=1939 ~ "30s",
                             startYear >=1940 & startYear<=1949 ~ "40s",
                             startYear >=1950 & startYear<=1959 ~ "50s",
                             startYear >=1960 & startYear<=1969 ~ "60s",
                             startYear >=1970 & startYear<=1979 ~ "70s"))
d1 %>% glimpse()
```

```
## Observations: 5,741
## Variables: 11
## $ tconst         <chr> "tt0010323", "tt0011000", "tt0011130", "tt0011237", ...
## $ titleType      <chr> "movie", "movie", "movie", "movie", "movie", "movie"...
## $ primaryTitle   <chr> "The Cabinet of Dr. Caligari", "Leaves From Satan's ...
## $ startYear      <dbl> 1920, 1920, 1920, 1920, 1920, 1920, 1920, 1920, 1920...
## $ runtimeMinutes <dbl> 76, 167, 82, 76, 73, 107, 90, 77, 145, 90, 79, 75, 1...
## $ genres         <chr> "Fantasy,Horror,Mystery", "Drama", "Drama,Horror,Sci...
## $ averageRating  <dbl> 8.1, 6.7, 7.0, 7.2, 6.7, 7.1, 7.4, 6.2, 7.4, 6.7, 6....
## $ numVotes       <dbl> 52649, 1047, 4561, 6128, 1063, 2053, 1927, 1356, 479...
## $ nconst         <chr> "nm0927468", "nm0003433", "nm0731910", "nm0091380", ...
## $ director       <chr> "Robert Wiene", "Carl Theodor Dreyer", "John S. Robe...
## $ decade        <chr> "20s", "20s", "20s", "20s", "20s", "20s", "20s", "20..."
```

The code below utilizes *ggplot* to plot histograms of the distribution of **runtimeMinutes** for each decade.

```
d1 %>%
  ggplot(aes(x=runtimeMinutes)) + geom_histogram(bins=60) + facet_wrap(~ decade)
```



The R code below will compute the population mean and population standard deviation in **runtimeMinutes** for each decade and store the values in a new dataframe, *results*.

```
library(radiant)
```

```
## Loading required package: radiant.data
```

```
## Loading required package: magrittr
```

```
##
```

```
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
```

```
##
```

```
##   set_names
```

```
## The following object is masked from 'package:tidyr':
```

```
##
```

```
##   extract
```

```
## Loading required package: lubridate
```

```
##
```

```
## Attaching package: 'lubridate'
```

```

## The following object is masked from 'package:base':
##
##     date

## Registered S3 method overwritten by 'radiant.data':
##   method      from
##   print.gtable gtable

##
## Attaching package: 'radiant.data'

## The following objects are masked from 'package:lubridate':
##
##     month, wday

## The following object is masked from 'package:forcats':
##
##     as_factor

## The following objects are masked from 'package:purrr':
##
##     is_double, is_empty, is_numeric

## The following object is masked from 'package:ggplot2':
##
##     diamonds

## Loading required package: radiant.design

## Loading required package: mvtnorm

## Loading required package: radiant.basics

## Loading required package: radiant.model

## Loading required package: radiant.multivariate

results <- d1 %>% group_by(decade) %>% summarize(mean=mean(runtimeMinutes),
                                                pop_sd=sdpop(runtimeMinutes))
results %>% glimpse()

## Observations: 6
## Variables: 3
## $ decade <chr> "20s", "30s", "40s", "50s", "60s", "70s"
## $ mean    <dbl> 95.95513, 90.24254, 97.30341, 99.64168, 106.82781, 105.06640
## $ pop_sd  <dbl> 27.43508, 18.64468, 20.55356, 21.54956, 24.30888, 21.35976

```

The following code will generate a function to calculate the standard error of the mean for each decade as well as a single sample of 100 movies from each decade and calculate the sample mean and standard deviation for each decade. Additionally, the SE around each population mean for each decade will be estimated using the standard deviation and sample size of these samples.

```
std_error <- function(x) {
  sd(x) / sqrt(length(x))
}
d1 %>% group_by(decade) %>% sample_n(100, replace=FALSE) %>%
  summarize(mean(runtimeMinutes), sd(runtimeMinutes), std_error(runtimeMinutes))
```

```
## # A tibble: 6 x 4
##   decade `mean(runtimeMinutes)` `sd(runtimeMinutes)` `std_error(runtimeMinutes)`
##   <chr>          <dbl>          <dbl>          <dbl>
## 1 20s             96.7             29.0             2.90
## 2 30s             89.3             16.5             1.65
## 3 40s             96.6             19.4             1.94
## 4 50s            100.             22.6             2.26
## 5 60s            108.             26.6             2.66
## 6 70s            104.             22.1             2.21
```

The code below will write a function to calculate the standard error of the mean for each decade using the population standard deviation for purposes of comparison to the values obtained from the sample created above.

```
pop_std_error <- function(x) {
  sdpop(x) / sqrt(length(x))
}
d1 %>% group_by(decade) %>%
  summarize(mean = mean(runtimeMinutes),
            sdpop = sdpop(runtimeMinutes),
            pop_se = pop_std_error(runtimeMinutes),
            length(decade))
```

```
## # A tibble: 6 x 5
##   decade mean sdpop pop_se `length(decade)`
##   <chr> <dbl> <dbl> <dbl>          <int>
## 1 20s    96.0  27.4  2.20            156
## 2 30s    90.2  18.6  0.805           536
## 3 40s    97.3  20.6  0.731           791
## 4 50s    99.6  21.5  0.652          1094
## 5 60s   107.   24.3  0.646          1417
## 6 70s   105.   21.4  0.511          1747
```

Challenge 3

The R code below will load the ‘zombies.csv’ dataset from GitHub and organize the data into a tibble.

```
f <- "https://raw.githubusercontent.com/difiore/ADA-datasets/master/zombies.csv"
d <- read_csv(f, col_names = T)
```

```
## Parsed with column specification:
## cols(
##   id = col_double(),
##   first_name = col_character(),
##   last_name = col_character(),
```

```
## gender = col_character(),
## height = col_double(),
## weight = col_double(),
## zombies_killed = col_double(),
## years_of_education = col_double(),
## major = col_character(),
## age = col_double()
## )
```

```
glimpse(d)
```

```
## Observations: 1,000
## Variables: 10
## $ id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
## $ first_name  <chr> "Sarah", "Mark", "Brandon", "Roger", "Tammy", "A...
## $ last_name   <chr> "Little", "Duncan", "Perez", "Coleman", "Powell"...
## $ gender      <chr> "Female", "Male", "Male", "Male", "Female", "Mal...
## $ height      <dbl> 62.88951, 67.80277, 72.12908, 66.78484, 64.71832...
## $ weight      <dbl> 132.0872, 146.3753, 152.9370, 129.7418, 132.4265...
## $ zombies_killed <dbl> 2, 5, 1, 5, 4, 1, 0, 4, 9, 2, 4, 4, 2, 5, 4, 2, ...
## $ years_of_education <dbl> 1, 3, 1, 6, 3, 4, 4, 0, 3, 3, 4, 3, 1, 5, 5, 2, ...
## $ major       <chr> "medicine/nursing", "criminal justice administra...
## $ age         <dbl> 17.64275, 22.58951, 21.91276, 18.19058, 21.10399...
```

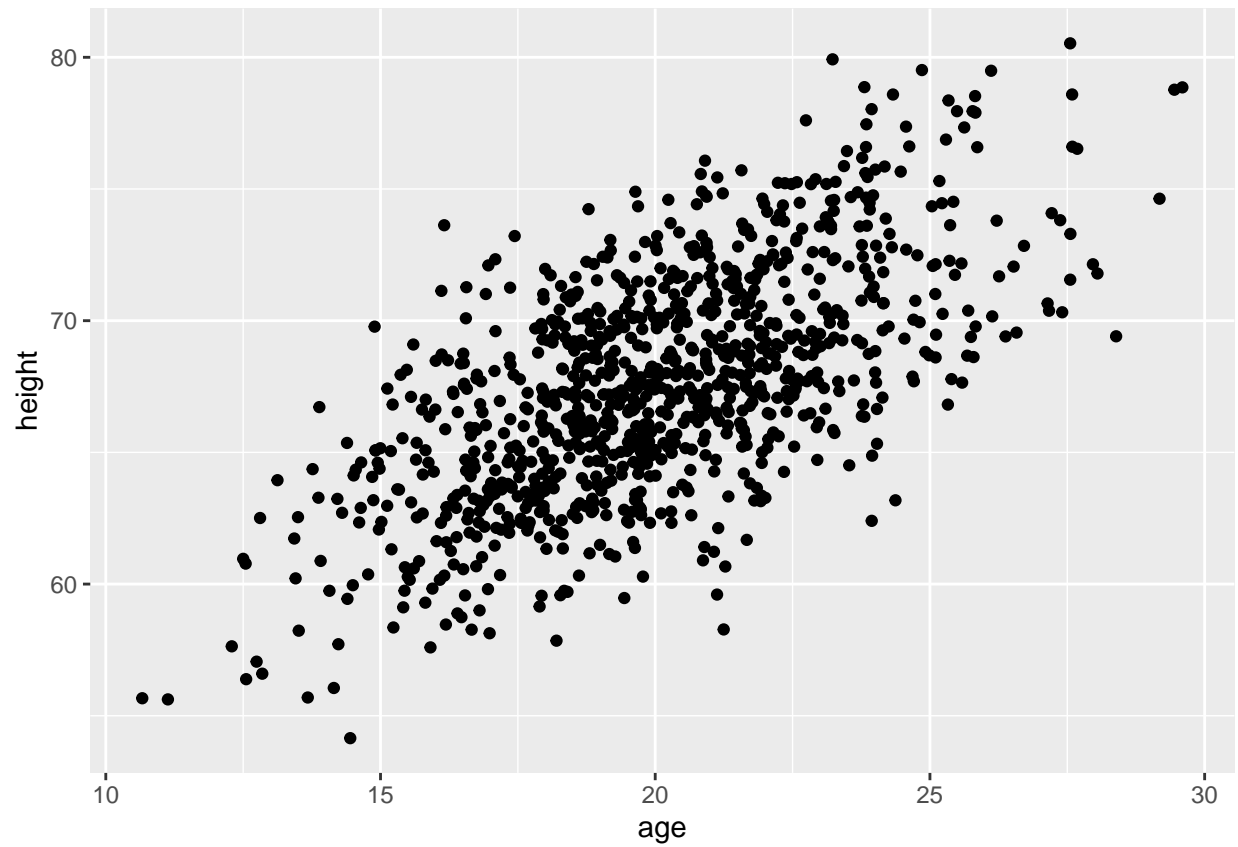
The code below will calculate the population mean and standard deviation for each quantitative random variable.

```
d %>% summarize(height_mean=mean(height),
                 height_sd=sdpop(height),
                 weight_mean=mean(weight),
                 weight_sd=sdpop(weight),
                 age_mean=mean(age),
                 age_sd=sdpop(age),
                 n_zombies_mean=mean(zombies_killed),
                 n_zombies_sd=sdpop(zombies_killed),
                 ed_mean=mean(years_of_education),
                 ed_sd=sdpop(years_of_education))
```

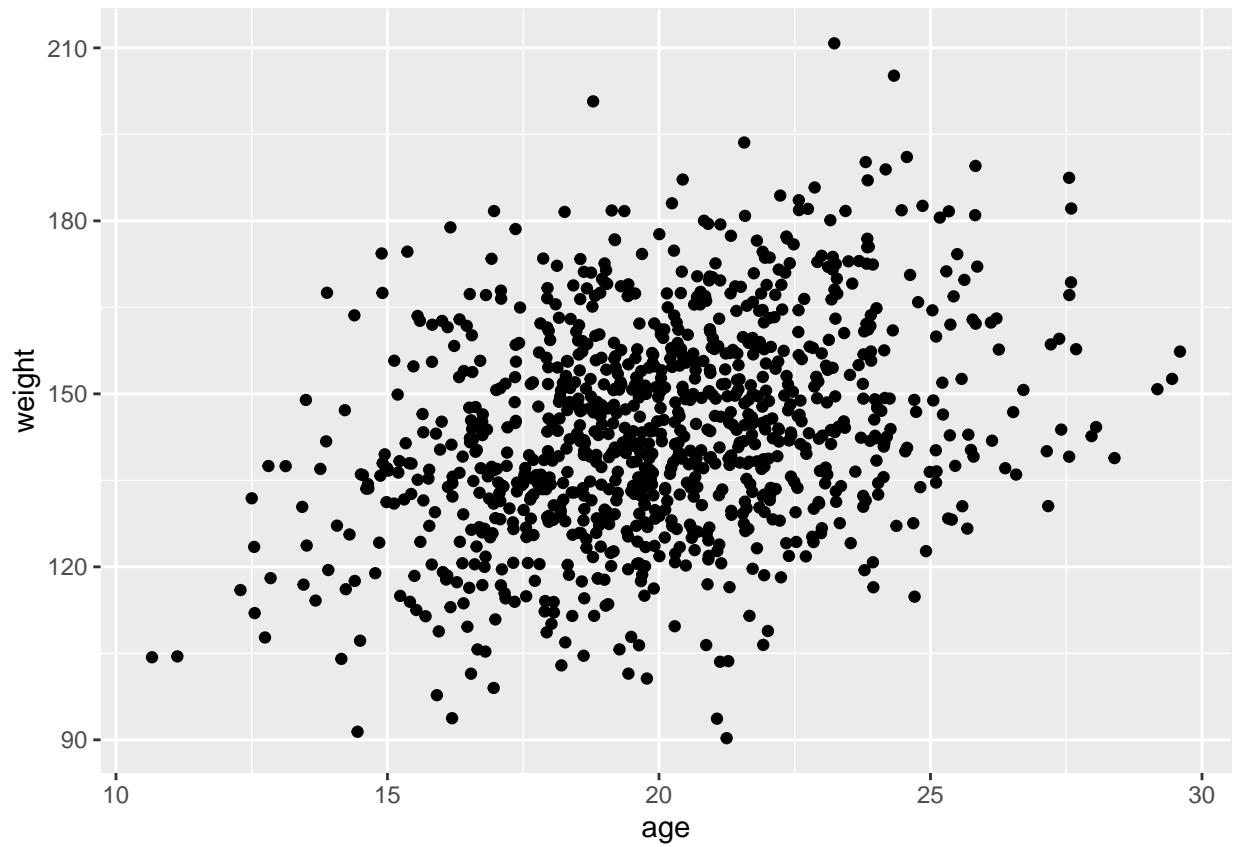
```
## # A tibble: 1 x 10
##   height_mean height_sd weight_mean weight_sd age_mean age_sd n_zombies_mean
##   <dbl>      <dbl>    <dbl>    <dbl>    <dbl> <dbl>      <dbl>
## 1      67.6      4.31     144.     18.4     20.0  2.96      2.99
## # ... with 3 more variables: n_zombies_sd <dbl>, ed_mean <dbl>, ed_sd <dbl>
```

The following will utilize *ggplot2* to make scatterplots of height and weight in relation to age.

```
d %>% ggplot(aes(x=age,y=height)) + geom_point()
```



```
d %>% ggplot(aes(x=age,y=weight)) + geom_point()
```



The scatterplot generated for height in relation to age shows a distinct positive relationship. The scatterplot generated for weight in relation to age does not show as strong of a relationship however there is still a positive tendency to the relationship between the two variables.