QUIZ #3 SOLUTIOS

MSAN 593

August 2, 2018

Instructions

1. No computer, no notes or electronic devices permitted in this quiz.
2. You may only use a pencil and eraser or pen.
3. Write your name at the top of the first page of this quiz.
4. You have 45 minutes to complete the quiz.
5. This quiz is designed to test your knowledge of `dplyr`, `magrittr` and to a lesser extent, `ggplot`. When writing answers to questions, you may assume that the aforementioned packages are loaded. Using base `R` functions that have a `dplyr`, `magrittr` or `ggplot` equivalent will result in a reduction or complete loss of grades for a given question.

Question 1 (4 pts)

Financial data on annualized returns in the data frame `myDF` follows:

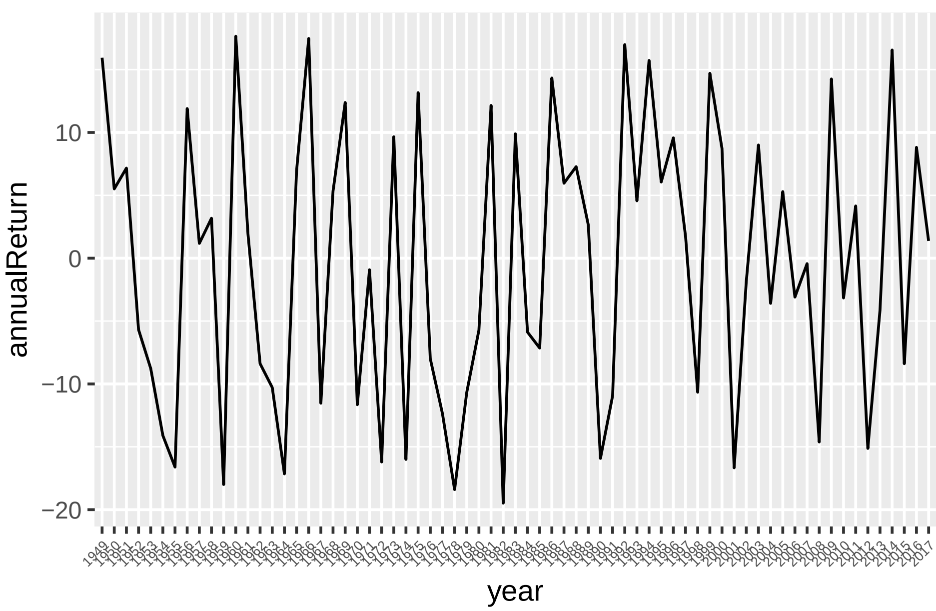
```
x <- as.character(1949:2017)
y <- runif(length(x), -20, 20)
myDF <- data.frame(year = x, annualReturn = y)
```

```
tibble::glimpse(myDF)
```

```
## Observations: 69
## Variables: 2
## $ year         <fct> 1949, 1950, 1951, 1952, 1953, 1954, 1955, 1956, 1...
## $ annualReturn <dbl> 15.944007, 5.512106, 7.164837, -5.685614, -8.7694...
```

Without coercioning any of the data, write the code that creates the following plot (*don't worry about rotating or scaling x-axis tick labels*):

```
myDF %>% ggplot() + geom_line(aes(x = year, y = annualReturn),
    group = 1) + theme(axis.text.x = element_text(angle = 45,
    hjust = 1, size = 5))
```



Question 2 (14 = 3 + 5 + 6 pts)

```
census <- read_csv("~/Desktop/2015census.csv")
```

Selected data from the 2015 census has been imported into the data frame `census`, as shown below.

```
census %>% select(2:13) %>% glimpse()
```

```
## Observations: 74,001
## Variables: 12
## $ State    <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama"...
## $ County   <chr> "Autauga", "Autauga", "Autauga", "Autauga", "Autauga"...
## $ TotalPop <int> 1948, 2156, 2968, 4423, 10763, 3851, 2761, 3187, 1091...
## $ Men      <int> 940, 1059, 1364, 2172, 4922, 1787, 1210, 1502, 5486, ...
## $ Women    <int> 1008, 1097, 1604, 2251, 5841, 2064, 1551, 1685, 5429,...
## $ Hispanic <dbl> 0.9, 0.8, 0.0, 10.5, 0.7, 13.1, 3.8, 1.3, 1.4, 0.4, 0...
## $ White    <dbl> 87.4, 40.4, 74.5, 82.8, 68.5, 72.9, 74.5, 84.0, 89.5,...
## $ Black    <dbl> 7.7, 53.3, 18.6, 3.7, 24.8, 11.9, 19.7, 10.7, 8.4, 12...
## $ Native   <dbl> 0.3, 0.0, 0.5, 1.6, 0.0, 0.0, 0.0, 3.1, 0.0, 0.0, 1.3...
## $ Asian    <dbl> 0.6, 2.3, 1.4, 0.0, 3.8, 0.0, 0.0, 0.0, 0.0, 0.3, 0.0...
## $ Pacific  <dbl> 0.0, 0.0, 0.3, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0...
## $ Citizen  <int> 1503, 1662, 2335, 3306, 7666, 2642, 2060, 2391, 7778,...
```

Answer the following questions. Where appropriate, you must use functions from `dplyr` as well as piping notation from `magrittr`.

(a) Compute the total population by state. The result/output should look exactly like the following table, noting that the following table is truncated for ease of exposition (not need to account for `knitr`):

```
census %>% group_by(State) %>% summarise(totalPopulation = sum(TotalPop)) %>%
  arrange(desc(totalPopulation)) %>% slice(1:5) %>% knitr::kable()
```

State	totalPopulation
California	38421464
Texas	26538614
New York	19673174
Florida	19645772
Illinois	12873761

(b) Compute the percentage of **national population** in each state. The result/output should look **exactly** like this, although it is truncated in length for ease of exposition):

```
census %>% group_by(State) %>% summarise(totalPopulation = sum(TotalPop)) %>%
    mutate(prct = totalPopulation/sum(totalPopulation) * 100) %>%
    arrange(desc(prct)) %>% select(-totalPopulation) %>% slice(1:5) %>%
    knitr::kable()
```

State	prct
California	12.003028
Texas	8.290775
New York	6.145983
Florida	6.137422
Illinois	4.021817

(c) Compute the relative number of US citizens in each state. The result/output should look **exactly** like this, although it is truncated in length for ease of exposition):

```
census %>% group_by(State) %>% summarize(totalPopulation = sum(TotalPop),
    totalCitizens = sum(Citizen)) %>% mutate(prct = totalCitizens/totalPopulation) %>%
    arrange(desc(prct)) %>% select(State, prct) %>% slice(1:5) %>%
    knitr::kable()
```

State	prct
Maine	0.7887097
Vermont	0.7869787
West Virginia	0.7863413
New Hampshire	0.7703740
Montana	0.7699328

Question 3 (22 = 4 + 5 + 2 + 6 + 5 pts)

A data frame containing information on Texas housing data, called `DF_texas`, is as follows:

```r
DF_texas <- ggplot2::txhousing
glimpse(ggplot2::txhousing)
```

```
## Observations: 8,602
## Variables: 9
## $ city      <chr> "Abilene", "Abilene", "Abilene", "Abilene", "Abilene...
## $ year      <int> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000...
## $ month     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5...
## $ sales     <dbl> 72, 98, 130, 98, 141, 156, 152, 131, 104, 101, 100, ...
## $ volume    <dbl> 5380000, 6505000, 9285000, 9730000, 10590000, 139100...
## $ median    <dbl> 71400, 58700, 58100, 68600, 67300, 66900, 73500, 750...
## $ listings  <dbl> 701, 746, 784, 785, 794, 780, 742, 765, 771, 764, 72...
## $ inventory <dbl> 6.3, 6.6, 6.8, 6.9, 6.8, 6.6, 6.2, 6.4, 6.5, 6.6, 6....
## $ date      <dbl> 2000.000, 2000.083, 2000.167, 2000.250, 2000.333, 20...
```

(a) Rewrite the following code using pipes (`magrittr`) and `dplyr`:

```r
DF_texas[(DF_texas$year == 2010) & (DF_texas$month == 4), c(1,
    4, 5)]
```

````
```r
DF_texas %>% filter(year == 2010, month == 4) %>% select(city,
    sales, volume)
```
````

````
```
## # A tibble: 46 x 3
##    city                  sales    volume
##    <chr>                 <dbl>     <dbl>
##  1 Abilene                 161  18788002
##  2 Amarillo                293  39634272
##  3 Arlington               451  64885372
##  4 Austin                 2230 513122847
##  5 Bay Area                508  92783572
##  6 Beaumont                200  26818968
##  7 Brazoria County          93  12067420
##  8 Brownsville              74   9238019
##  9 Bryan-College Station   233  38691043
## 10 Collin County          1254 290057387
## # ... with 36 more rows
```
````

(b) A manager is interested in finding the year in which the fraction of homes sold to homes listed for sale, to be called `salesPrct`, is maximal for each city. The expected output is below, which is truncated for ease of exposition. Write the code required to generate the following table (don't worry about truncating or formatting table with `kable`).

```
DF_texas %>% mutate(salesPrct = sales/listings * 100) %>% group_by(city) %>%
    top_n(1, salesPrct) %>% select(city, year, salesPrct) %>%
    head() %>% knitr::kable()
```

city	year	salesPrct
Abilene	2005	34.63855
Amarillo	2015	31.24424
Arlington	2015	85.66775
Austin	2000	54.74150
Bay Area	2015	39.59888
Beaumont	2006	27.11443

(c) The manager, her interest peaked, would like to see in which years the highest `salesPrct`'s occurred, to explore whether or not there was a dominant year for cities. Working from the previous code block (above), I have begun the next line of code for you. Add the final lines of code necessary to generate the following table, which is ordered according to `n`:

```
DF_texas %>% mutate(salesPrct = sales/listings * 100) %>% group_by(city) %>%
    top_n(1, salesPrct) %>% select(city, year, salesPrct) %>%
    ungroup() %>% count(year) %>% arrange(desc(n)) %>% head() %>%
    knitr::kable()
```

year	n
2015	16
2014	5
2005	4
2006	4
2000	3
2007	3

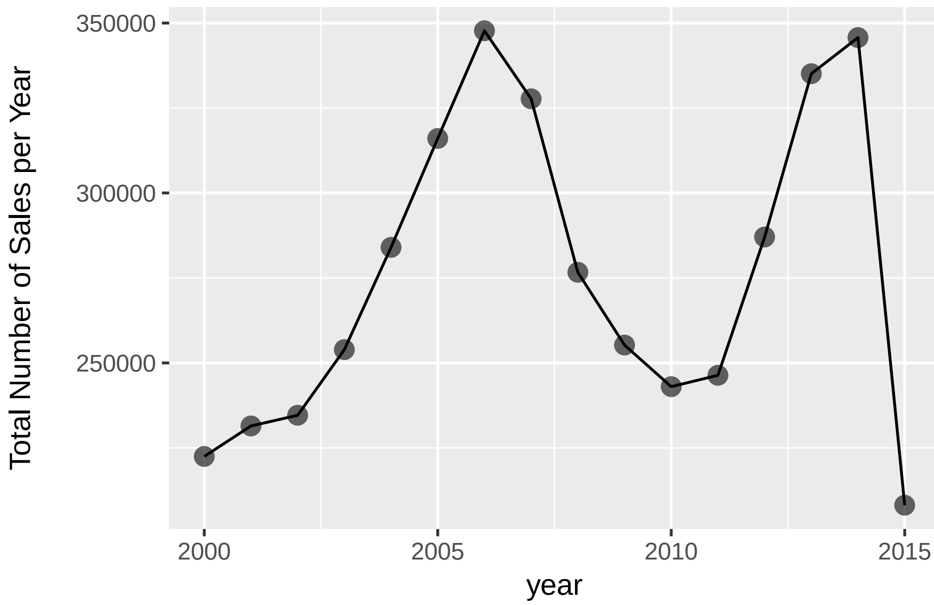
(d) The manager is interested in exploring some descriptive statistics related to `listings` and `sales`, namely to compute their mean, median and standard deviation. Write *concise* code to compute the mean, median and standard deviation of both `listings` and `sales` by city. The output should look like the following table:

```
DF_texas %>% select(city, listings, sales) %>% group_by(city) %>%
    summarize_all(funs(mean(., na.rm = T), median(., na.rm = T),
        sd(., na.rm = T))) %>% head() %>% knitr::kable()
```

city	listings_mean	sales_mean	listings_median	sales_median	listings_sd	sales_sd
Abilene	813.4086	150.4866	801.0	146	130.0180	40.01161
Amarillo	1286.2143	238.6524	1263.5	242	159.1457	59.74122
Arlington	1945.3172	423.9840	1909.0	425	774.8074	102.94510
Austin	8696.4118	1996.6898	9095.0	1910	2340.1084	578.21479
Bay Area	2999.2097	502.6150	2791.0	489	837.7091	131.53899
Beaumont	1332.0802	177.0588	1307.0	176	298.6503	41.95494

(e) The `sales` variable contains information on **how many** houses were sold in a given month. Write code in a single series of pipes—i.e., do not store any variables—that will generate the following graph (*don't worry about labeling y-axis*):

```
ggplot2::txhousing %>% group_by(year) %>% summarise(salesSum = sum(sales,
    na.rm = T)) %>% ggplot(aes(x = year, y = salesSum)) + geom_line() +
    geom_point(size = 3, alpha = 0.6) + ylab("Total Number of Sales per Year\n")
```



Question 4 (5 pts)

The `cars` data set describes the relationship between the `speed` of cars and the associated stopping `distance`. **n.b.** There are multiple observations (rows) for a given speed, i.e., they tested the stopping `distance` of various cars at the same `speed`. A summary of the `cars` data set is provided below.

```
glimpse(cars)
```

```
## Observations: 50
## Variables: 2
## $ speed <dbl> 4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12, 12, 12, 12, 13...
## $ dist  <dbl> 2, 10, 4, 22, 16, 10, 18, 26, 34, 17, 28, 14, 20, 24, 28...
```

```
head(cars)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

Write the code that generates the following graph, plotting **mean** stopping distance (y) against speed (x).

```
cars %>% group_by(speed) %>% summarize(meanDist = mean(dist)) %>%
    ggplot() + geom_line(aes(x = speed, y = meanDist), group = 1) +
    geom_point(aes(x = speed, y = meanDist), size = 3, alpha = 0.6)
```