Homework 5

Name: Arthur Viegas Eguia

I worked with: Sarah, Allison, and Jenna at the stats help lab

Click the "Knit" button in RStudio to knit this file to a pdf.

Problem 1: Regular expression

a.

answer: There were 16314 baby names that end in vowels in 2017

```
pattern <- "(.{1,})[aeiouy]\\b"
babynames %>%
  filter(year == 2017) %>%
  select(name) %>%
  unlist() -> names
sum(str_count(names, pattern))
## [1] 16314
```

b.

answer: 4 names matched this pattern, Edward, Eduard, Edgard, Edvard

```
pattern2 <- "^Ed(.{1,})rd$"
x <- c("Edward", "Eddard", "Ned")
str_view(x, pattern2)
## [1] | <Edward>
## [2] | <Eddard>
str_view(names, pattern2)
## [18478] | <Edward>
## [22089] | <Eduard>
## [25079] | <Edgard>
## [31050] | <Edvard>
sum(str_count(names, pattern2))
## [1] 4
```

c.

answer: Yes, there are 19 names like that, the names are Aya, Aiya, Yui, Iya, Aoi, Yue, Yoyo, Yu, Yiyi, Io, Ai, Yaa, Yi, Aaya, Oya, Yi, Yao, Yu, Ayo

```
pattern3 <- "^([aeiouyAEIOUY]{1,})$"
str_view(names, pattern3)
##   [790] | <Aya>
##  [4552] | <Aiya>
##  [5993] | <Yui>
##  [7185] | <Iya>
```

```
##  [7831] | <Aoi>
##  [9858] | <Yue>
## [10600] | <Yoyo>
## [11513] | <Yu>
## [12605] | <Yiyi>
## [13161] | <Io>
## [14105] | <Ai>
## [15700] | <Yaa>
## [15720] | <Yi>
## [15810] | <Aaya>
## [17671] | <Oya>
## [24940] | <Yi>
## [27890] | <Yao>
## [28974] | <Yu>
## [30708] | <Ayo>
sum(str_count(names, pattern3))
## [1] 19
```

d.

answer: The length of the vector is 2960

```
pattern4 <- "^.{4}$"
result <- babynames %>%
  filter(year == 2017) %>%
  mutate(is_4_rows = str_detect(name, pattern4)) %>%
  filter(is_4_rows) %>%
  dplyr::pull(name)

result %>%
  length()
## [1] 2960
```

e.

answer: The palindrome names are anna, elle, emme, adda, alla, izzi, luul, avva, otto, anna

```
pattern5 <- "(.)(.)\\2\\1"
result %>%
  str_to_lower() %>%
  str_subset(pattern5)
##  [1] "anna" "elle" "emme" "adda" "alla" "izzi" "luul" "avva" "otto" "anna"
```

Problem 2: Energy autocorrelation

a.

```
energy <- readr::read_csv("https://raw.githubusercontent.com/deepbas/statdatasets/main/energy.csv",
                     col_type = cols(
                      .default = col_double(),
                       Timestamp = col_datetime(format = ""),
                       dayWeek = col_factor(levels=c("Mon","Tues","Wed","Thurs","Fri","Sat","Sun"))
                      ))
```

```r
x <- energy %>%
  arrange(Timestamp) %>%    # making sure sorted by time
  pull("Olin_Hall_of_Science")
acf_out <- acf(
  x,   # time series
  na.action = na.pass,    # skips over NAs
  lag.max = 4,   # max lag
  plot = FALSE)   # don't plot
acf_out
##
## Autocorrelations of series 'x', by lag
##
##     0     1     2     3     4
## 1.000 0.956 0.950 0.934 0.917
acf_out$acf   # autocorr values
## , , 1
##
##           [,1]
## [1,] 1.0000000
## [2,] 0.9556181
## [3,] 0.9502154
## [4,] 0.9344803
## [5,] 0.9169001
acf_out$lag   # lag values
## , , 1
##
##      [,1]
## [1,]    0
## [2,]    1
## [3,]    2
## [4,]    3
## [5,]    4
```

a.

answer: Function written

```r
autocor_fun <- function(KWHVect, maxLag){
  acf_out <- acf(
    KWHVect, #Time series
    na.action = na.pass,
    lag.max = maxLag,
    plot = FALSE
    )


  return(data.frame(autocor = as.vector(acf_out$acf),
                    lag = 0:maxLag))
}
autocor_fun(x, 4)
##     autocor lag
## 1 1.0000000   0
## 2 0.9556181   1
## 3 0.9502154   2
```

```
## 4 0.9344803   3
## 5 0.9169001   4
```

b.

answer: write your answer here

```
energy %>%
  arrange(Timestamp) %>%    # making sure sorted by time
  select("Sayles-Hill" ,"Language_&_Dining_Center", "Olin_Hall_of_Science") %>%
  map_df(autocor_fun, maxLag = 4, .id = "buildings")
##                   buildings   autocor lag
## 1               Sayles-Hill 1.0000000   0
## 2               Sayles-Hill 0.9356347   1
## 3               Sayles-Hill 0.9286133   2
## 4               Sayles-Hill 0.9147799   3
## 5               Sayles-Hill 0.8935076   4
## 6  Language_&_Dining_Center 1.0000000   0
## 7  Language_&_Dining_Center 0.9548038   1
## 8  Language_&_Dining_Center 0.9415759   2
## 9  Language_&_Dining_Center 0.9259386   3
## 10 Language_&_Dining_Center 0.9080089   4
## 11     Olin_Hall_of_Science 1.0000000   0
## 12     Olin_Hall_of_Science 0.9556181   1
## 13     Olin_Hall_of_Science 0.9502154   2
## 14     Olin_Hall_of_Science 0.9344803   3
## 15     Olin_Hall_of_Science 0.9169001   4
```

c.

answer: write your answer here

```
energy_narrow <- energy %>%
  mutate(month = month(month, label=TRUE)) %>%
  pivot_longer(
    cols = 9:90,
    names_to = "building",
    values_to = "energyKWH")

#Special thanks to Jenna from the Stats Help Lab
energy_narrow %>%
  group_by(building) %>%
  arrange(building, Timestamp) %>%
  filter(building == "Sayles-Hill" |
           building == "Language_&_Dining_Center" |
           building == "Olin_Hall_of_Science") %>%
  reframe(auto_correlation = autocor_fun(energyKWH, 4)) %>%
  unnest(auto_correlation)
## # A tibble: 15 x 3
##    building                 autocor   lag
##    <chr>                      <dbl> <int>
##  1 Language_&_Dining_Center   1         0
##  2 Language_&_Dining_Center   0.955     1
##  3 Language_&_Dining_Center   0.942     2
##  4 Language_&_Dining_Center   0.926     3
```

```
##  5 Language_&_Dining_Center   0.908     4
##  6 Olin_Hall_of_Science       1         0
##  7 Olin_Hall_of_Science       0.956     1
##  8 Olin_Hall_of_Science       0.950     2
##  9 Olin_Hall_of_Science       0.934     3
## 10 Olin_Hall_of_Science       0.917     4
## 11 Sayles-Hill                1         0
## 12 Sayles-Hill                0.936     1
## 13 Sayles-Hill                0.929     2
## 14 Sayles-Hill                0.915     3
## 15 Sayles-Hill                0.894     4
```

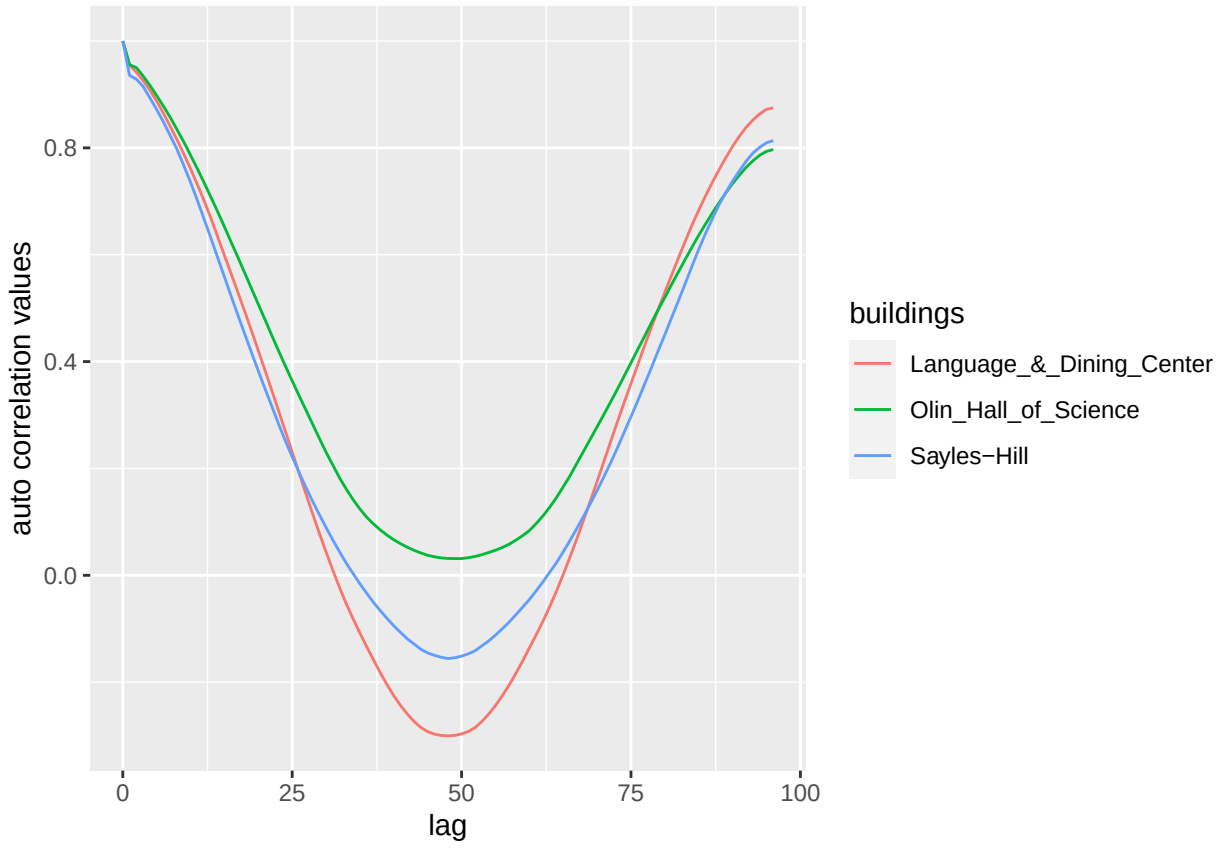
d.

answer: When the lag is 0, the auto correlation values are very high, over 0.80, then they start to decrease in a shape that resembles a parabola.

The auto correlation values reach a minimum at around 50, where they are either close to 0, or below 0.

After that, the auto correlation value starts to increase, in a shape similar to a parabola, and gets to values close to 0.8.

```
energy %>%
  arrange(Timestamp) %>%    # making sure sorted by time
  select("Sayles-Hill" ,
         "Language_&_Dining_Center",
         "Olin_Hall_of_Science") %>%
  map_df(autocor_fun,
         maxLag = 96,
         .id = "buildings") %>%
  ggplot(aes(x = lag, y = autocor, color = buildings)) +
  geom_line() + labs(y = "auto correlation values", x = "lag")
```



Problem 3: weather

a.

answer:

```
library(nasaweather)
for(coeff in 5:11){
  str_c(colnames(atmos)[coeff],
        (sd(atmos[[coeff]],
             na.rm = TRUE)/
        mean(atmos[[coeff]],
             na.rm = TRUE)), sep = " ") ->output
  print(output)
}
## [1] "surftemp 0.0157716107833334"
## [1] "temp 0.015868379223734"
## [1] "pressure 0.0458542184057587"
## [1] "ozone 0.0709975641748931"
## [1] "cloudlow 0.564537783240515"
## [1] "cloudmid 0.634662357561836"
## [1] "cloudhigh 1.02927630626123"
```

b.

answer: write your answer here

```r
coefficient_of_variation <- function(x){
  sd(x, na.rm = TRUE)/mean(x, na.rm = TRUE)
}
atmos %>%
  select(-(1:4)) %>%
  map_dfc(coefficient_of_variation)
## # A tibble: 1 x 7
##   surftemp   temp pressure  ozone cloudlow cloudmid cloudhigh
##      <dbl>  <dbl>    <dbl>  <dbl>    <dbl>    <dbl>     <dbl>
## 1   0.0158 0.0159   0.0459 0.0710    0.565    0.635      1.03
```

c.

```r
# e.g. named vector with names x and y and values 1 and 2
c(x = 1, y = 2)
## x y
## 1 2
```

answer:

```r
my_stats <- function(vect){
  c(mean = mean(vect, na.rm = TRUE),
    sd = sd(vect, na.rm = TRUE),
    min = min(vect, na.rm = TRUE),
    Q1 = quantile(vect, 0.25, na.rm = TRUE),
    median = median(vect, na.rm = TRUE),
    Q3 = quantile(vect, 0.75, na.rm = TRUE),
    max = max(vect, na.rm = TRUE))
}
my_stats(c(1,2,3,4,5))
##     mean       sd      min   Q1.25%   median   Q3.75%      max
## 3.000000 1.581139 1.000000 2.000000 3.000000 4.000000 5.000000
```

d.

answer: write your answer here

```r
atmos %>%
  select(-(1:4)) %>%
  map_df(my_stats) %>%
  mutate(variables = (colnames(atmos)[-(1:4)]))
## # A tibble: 7 x 8
##    mean    sd   min `Q1.25%` median `Q3.75%`    max variables
##   <dbl> <dbl> <dbl>    <dbl>  <dbl>    <dbl>  <dbl> <chr>
## 1 296.   4.67 266      294.   297.     299.   315.  surftemp
## 2 298.   4.73 269.     296.   299.     301.   310   temp
## 3 985.  45.2  615      995   1000     1000   1000   pressure
## 4 267.  19.0  232      254    264      276    390   ozone
## 5  26.2 14.8    0.5     15     23.5     34.5   84.5 cloudlow
## 6  15.3  9.69   0        7.5   14       22     83.5 cloudmid
## 7  12.0 12.4    0        1.5    8.5     18.5   62.5 cloudhigh
```

e.

answer:

```r
#I worked with Sarah and Allison for this question
atmos %>%
  group_by(year) %>%
  summarise(my_stats = my_stats(temp)) %>%
  mutate(stat = c("mean", "sd", "min", "Q1", "median","Q3", "max")) %>%
  pivot_wider(names_from = "stat",
              values_from = "my_stats")
## # A tibble: 6 x 8
## # Groups:   year [6]
##    year  mean    sd   min    Q1 median    Q3   max
##   <int> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
## 1  1995  297.  4.94  269.  296.   298.  300.  308.
## 2  1996  297.  4.55  272.  295    298.  300.  308.
## 3  1997  298.  4.77  273.  296    299.  301.  308.
## 4  1998  299.  4.65  273.  296.   300.  302.  310
## 5  1999  298.  4.49  273.  296.   299.  301.  310.
## 6  2000  298.  4.75  273.  296    300.  302.  309.
```

Problem 4:

```r
#install.packages("titanic")
library(titanic)
set.seed(12233)
df = tibble(titanic_train)  #load dataset
```

a.

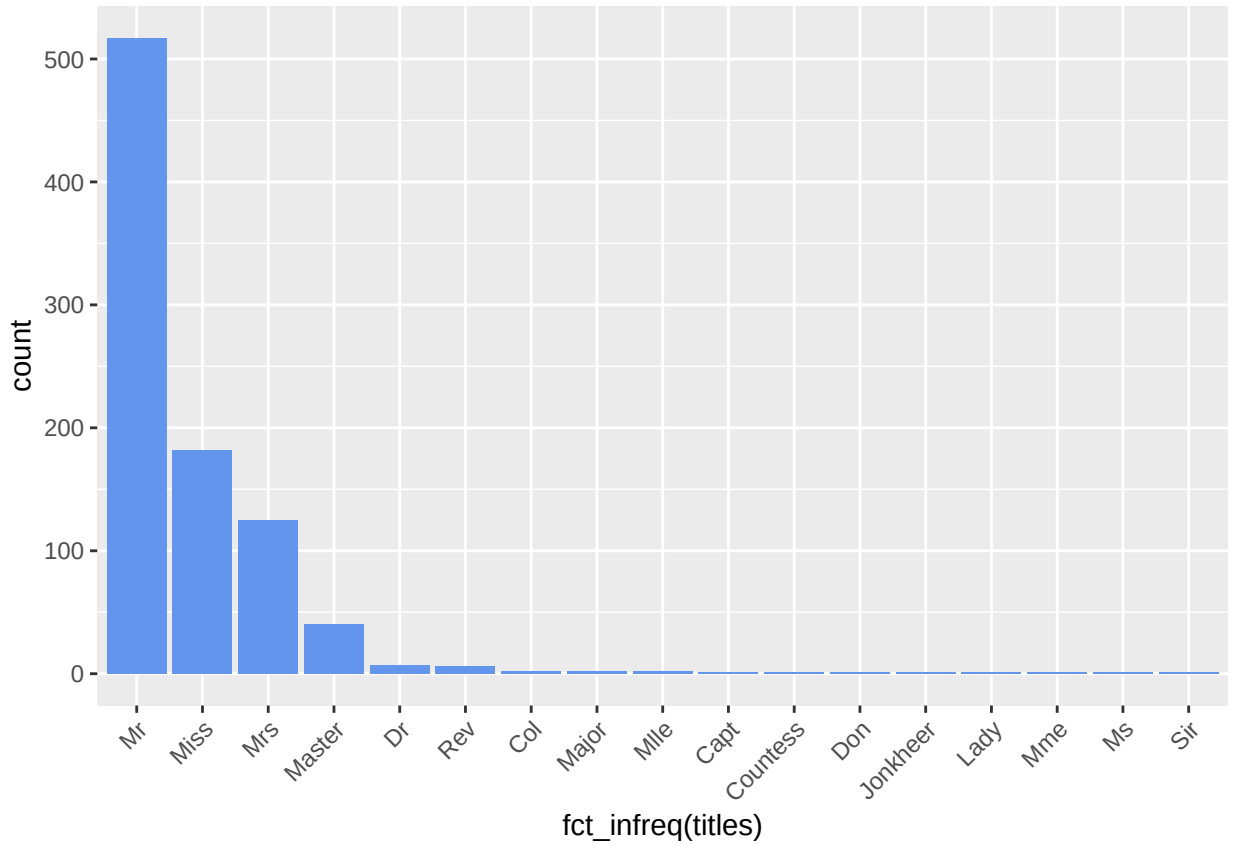
answer:

```r
reg <- "(?<=\\s)[:alnum:]+(?=\\.)"
str_extract(df$Name, reg) ->titanicTitles
```

b.

answer:

```r
titanic_titles_df <- data.frame(titles = titanicTitles)
titanic_titles_df %>%
  mutate(titles = factor(titles)) %>%
  ggplot(aes(x = fct_infreq(titles))) +
  geom_bar(fill = "cornflowerblue") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```




```
#Text rotated based on code from https://forum.posit.co/t/boxplot-how-to-rotate-x-axis-labels-to-45/871
```

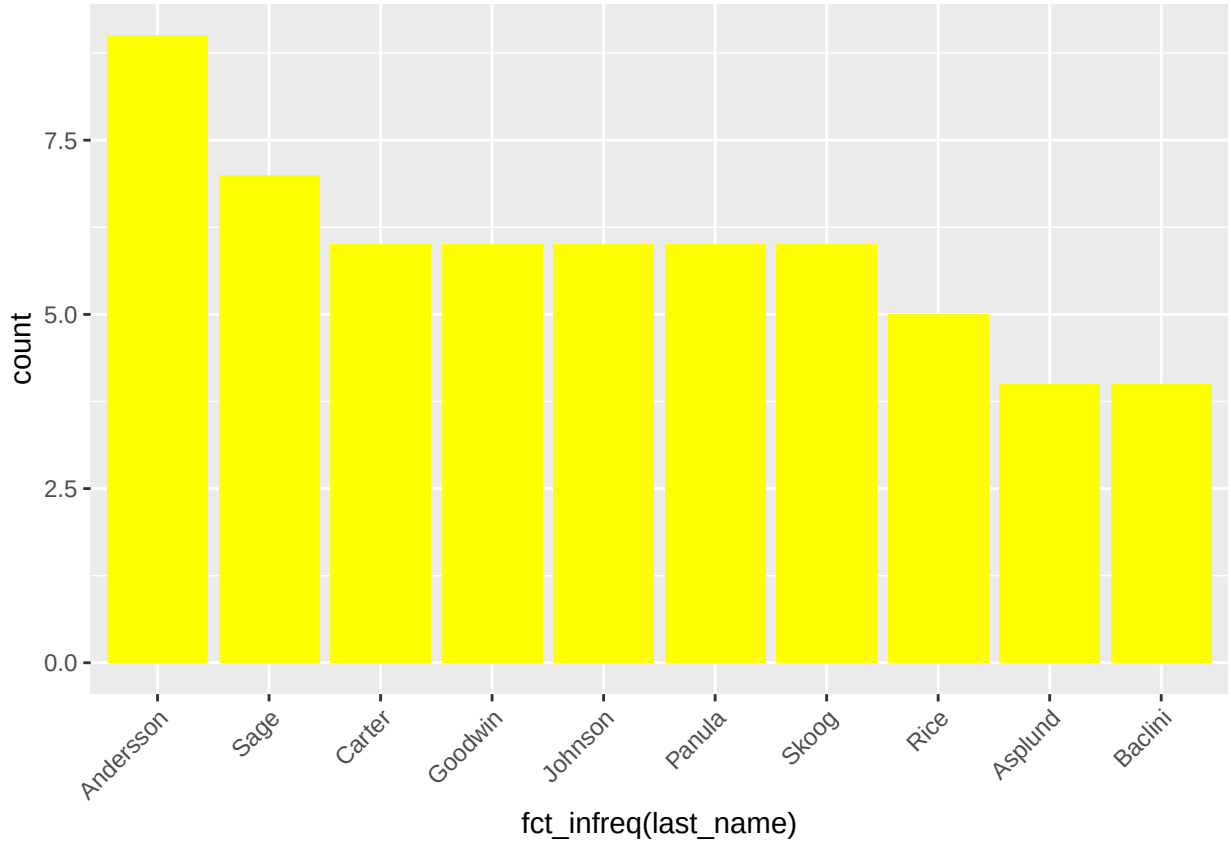
c.

answer:

```
reg <- ".+(?=\\,)"
str_extract(df$Name, reg) -> titanicLastNames
last_names_df <- data.frame(last_name = titanicLastNames)
most_popular_last_names <- last_names_df %>%
  group_by(last_name) %>%
  summarise(popular_last_name = n()) %>%
  arrange(desc(popular_last_name)) %>%
  slice(1:10)
most_popular_last_names
## # A tibble: 10 x 2
##    last_name popular_last_name
##    <chr>                <int>
##  1 Andersson                9
##  2 Sage                     7
##  3 Carter                   6
##  4 Goodwin                  6
##  5 Johnson                  6
##  6 Panula                   6
##  7 Skoog                    6
##  8 Rice                     5
##  9 Asplund                  4
```

```
## 10 Baclini                   4
```

```
last_names_df %>%
  filter(last_name %in% most_popular_last_names$last_name) %>%
  mutate(last_name = factor(last_name)) %>%
  ggplot(aes(x = fct_infreq(last_name))) +
  geom_bar(fill = "yellow") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))
```



d.

answer:

```
mostPopularByLetter <- function(df, firstLetter, lastLetter){
  reg <- ".+(?=\\,)"
  str_extract(df$Name, reg) -> allLastNames
  filterLetters <- str_c("^",
                         str_to_upper(firstLetter),
                         ".*",
                         str_to_lower(lastLetter),
                         "$")
  filteredNames <- str_extract(allLastNames, filterLetters)
  names_df <- data.frame(last_name = filteredNames) %>%
    drop_na() %>%
    mutate(last_name = factor(last_name)) %>%
    ggplot(aes(x = fct_infreq(last_name))) +
```

```
    geom_bar(fill = "pink") +
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1))


}
res <- mostPopularByLetter(df, "S", "m")
res
```