STAT 4410/8416 Homework 2

lastName firstName

Due on October 7, 2021

1. The data set `tips` contains tip amounts for different party sizes as well as total bill amounts per payment. We can get the data from the reshape2 package as follows:

```
library(reshape2)
tips.dat <- tips
```

Now answer the following questions:

    a. Compute the tip rate, dividing tip by total bill, and create a new column called `tip.rate` in the dataframe `tips.dat`. Demonstrate your results by showing the head of `tips.dat`.

```
tips.dat$tip.rate <-  tips.dat$tip/tips.dat$total_bill
head(tips.dat)
```

```
##   total_bill  tip     sex smoker day   time size   tip.rate
## 1      16.99 1.01 Female     No Sun Dinner    2 0.05944673
## 2      10.34 1.66   Male     No Sun Dinner    3 0.16054159
## 3      21.01 3.50   Male     No Sun Dinner    3 0.16658734
## 4      23.68 3.31   Male     No Sun Dinner    2 0.13978041
## 5      24.59 3.61 Female     No Sun Dinner    4 0.14680765
## 6      25.29 4.71   Male     No Sun Dinner    4 0.18623962
```

    b. Draw a side-by-side violin plot of the tip rate for each party size. Order the party sizes by the median tip rate. Provide your code as well as your plot. Which party size is responsible for the highest median tip rate?

```
library(ggplot2)
p <- ggplot(tips.dat, aes(x=factor(size), y=tip.rate)) + geom_violin() + stat_summary(fun=media
n, geom="point", size=2, color="red")
sort(with(tips.dat, tapply(tip.rate, size, median)), decreasing = T)
```

```
##         1         6         3         2         4         5
## 0.2027521 0.1628912 0.1593231 0.1561042 0.1466993 0.1213887
```

```
p
```

0.6

0.4

0.2

tip.rate

1   2   3   4   5   6
factor(size)
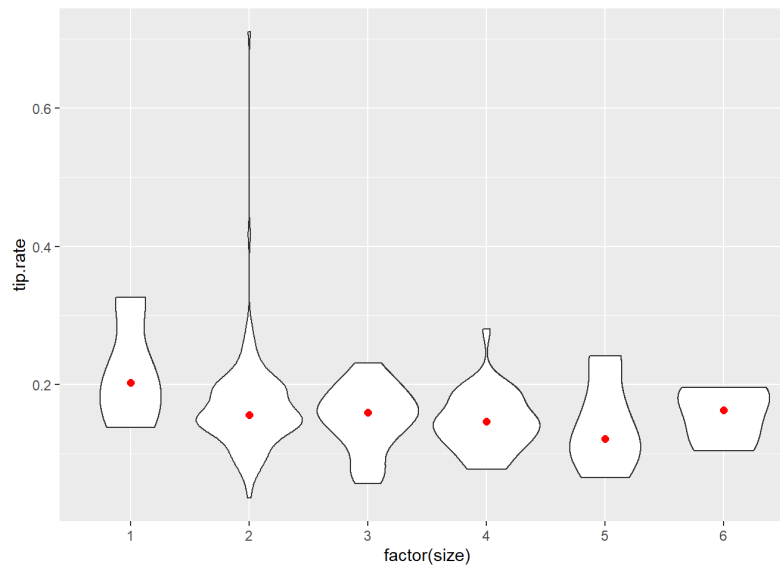
Party size 1 is responsible for the highest median tip rate.

c. Generate a similar plot to the one you created in question 2b for each day (instead of party size) and facet by sex and smoker. Is the shape of the violin plot similar for each faceted condition?

```
p <- ggplot(tips.dat, aes(x=factor(day), y=tip.rate)) + geom_violin() + stat_summary(fun=median,
geom="point", size=2, color="red") + facet_wrap(vars(sex, smoker))
sort(with(tips.dat, tapply(tip.rate, size, median)), decreasing = T)
```

```
##          1          6          3          2          4          5
## 0.2027521 0.1628912 0.1593231 0.1561042 0.1466993 0.1213887
```
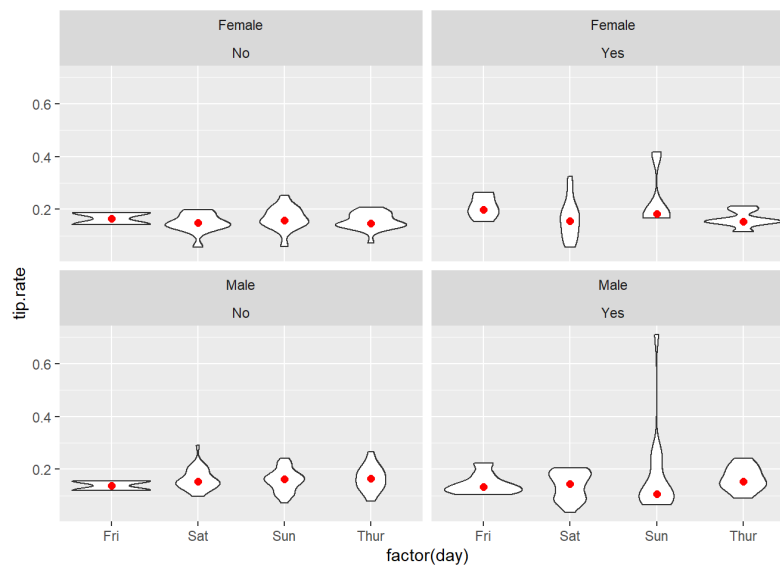
```
p
```

Female
No

Female
Yes

0.6

0.4

0.2

Male
No

Male
Yes

0.6

0.4

0.2

tip.rate

Fri   Sat   Sun   Thur   Fri   Sat   Sun   Thur
factor(day)

The shapes of the plot look to have similarities between yes and no, but not much between female and male.

2. We can generate an nxk matrix M and a vector V of length k for some specific values of n and k as follows:

```
set.seed(321)
n <- 9
k <- 5
V <- sample(seq(50), size = k, replace = TRUE)
M <- matrix(rnorm(n * k), ncol = k)
```

a. Now, carefully review the following for-loop. Rewrite the code so that you perform the same job without a loop.

```
A <- M
A <- t(t(A)/V)
round(A, digits = 4)
```

```
##            [,1]    [,2]    [,3]    [,4]    [,5]
## [1,]   0.0173 -0.0134 -0.0169 -0.0784 -0.0348
## [2,]  -0.0070  0.0183  0.0547 -0.0899  0.0633
## [3,]  -0.0339  0.0426 -0.0387  0.0766 -0.0154
## [4,]  -0.0167  0.1886 -0.0242 -0.0818  0.0254
## [5,]  -0.0049 -0.0119 -0.0542 -0.0157  0.0122
## [6,]   0.0072 -0.1500  0.0192  0.0904 -0.0304
## [7,]   0.0464  0.0079 -0.0265  0.0494  0.0536
## [8,]  -0.0007 -0.0457  0.0191  0.0823  0.0021
## [9,]   0.0136  0.0848 -0.0633 -0.0532  0.0426
```

b. Now do the same experiment for $n = 900$ and $k = 500$. Which runs faster, your code or the for-loop? Demonstrate this using the function `system.time()`.

```r
F_loop <- function() {
  set.seed(321)
n <- 900
k <- 500
V <- sample(seq(50), size = k, replace = TRUE)
M <- matrix(rnorm(n * k), ncol = k)

  X <- M
for(i in seq(n)) {
  X[i, ] <- round(M[i, ] / V, digits = 4)
}
X
}

F_noloop <- function() {
  set.seed(321)
n <- 900
k <- 500
V <- sample(seq(50), size = k, replace = TRUE)
M <- matrix(rnorm(n * k), ncol = k)

  A <- M
A <- t(t(A)/V)
round(A, digits = 4)
}

system.time(F_loop())
```

```
##    user  system elapsed
##    0.19    0.00    0.19
```

```r
system.time(F_noloop())
```

```
##    user  system elapsed
##    0.08    0.01    0.11
```

The function without a loop happened faster, as expected.

3. We want to generate a plot of US arrest data (USArrests). Please provide the detailed codes to answer the following questions.

   a. Obtain USA state boundary coordinates data for generating a USA map using function `map_data()` and store the data in `mdat`. Display the first few rows of data from `mdat`, noticing that there is a column called `order` that contains the true order of the coordinates.

```r
library(maps)
mdat <- map_data("state")
head(mdat)
```

```
##          long      lat group order  region subregion
## 1 -87.46201 30.38968    1     1 alabama      <NA>
## 2 -87.48493 30.37249    1     2 alabama      <NA>
## 3 -87.52503 30.37249    1     3 alabama      <NA>
## 4 -87.53076 30.33239    1     4 alabama      <NA>
## 5 -87.57087 30.32665    1     5 alabama      <NA>
## 6 -87.58806 30.32665    1     6 alabama      <NA>
```

b. You will find USA crime data in the data frame called `USArrests`. Standardize the crime rates and create a new column called `state` so that all state names are in lower case. Store this new data in an object called `arrest` and report the first few rows of `arrest`.

```
arrest <- scale(USArrests)
state <- tolower(state.name)
arrest <- cbind(arrest, state)
head(arrest)
```

```
##             Murder               Assault              UrbanPop
## Alabama    "1.24256408388112"   "0.782839347089918"  "-0.520906614581632"
## Alaska     "0.507862482177576"  "1.10682252261763"   "-1.21176419360236"
## Arizona    "0.0716334061660956" "1.47880320563092"   "0.998980059263978"
## Arkansas   "0.232349381538747"  "0.230868011005672"  "-1.07359267779822"
## California "0.278268231645218"  "1.26281442194578"   "1.75892339618678"
## Colorado   "0.0257145560596241" "0.398859287205225"  "0.860808543459832"
##             Rape                  state
## Alabama    "-0.00341647301516242" "alabama"
## Alaska     "2.48420294114997"     "alaska"
## Arizona    "1.04287838787833"     "arizona"
## Arkansas   "-0.184916601945666"   "arkansas"
## California "2.06782029242705"     "california"
## Colorado   "1.86496720715179"     "colorado"
```

c. Merge the two data sets `mdat` and `arrest` by state name. Note: merging will change the order of the coordinates data. So, order the data back to the original order and store the merged-ordered data in `odat`. Report the first few rows of data from `odat`.

```
odat <- merge(mdat,arrest, by.x="region", by.y="state")
odat <- as.data.frame(odat)
odat <- odat[order(odat[,5]),]
head(odat)
```

```
##     region      long     lat group order subregion         Murder
## 1 alabama -87.46201 30.38968    1     1      <NA> 1.24256408388112
## 2 alabama -87.48493 30.37249    1     2      <NA> 1.24256408388112
## 6 alabama -87.52503 30.37249    1     3      <NA> 1.24256408388112
## 7 alabama -87.53076 30.33239    1     4      <NA> 1.24256408388112
## 8 alabama -87.57087 30.32665    1     5      <NA> 1.24256408388112
## 9 alabama -87.58806 30.32665    1     6      <NA> 1.24256408388112
##             Assault            UrbanPop                       Rape
## 1 0.782839347089918 -0.520906614581632 -0.00341647301516242
## 2 0.782839347089918 -0.520906614581632 -0.00341647301516242
## 6 0.782839347089918 -0.520906614581632 -0.00341647301516242
## 7 0.782839347089918 -0.520906614581632 -0.00341647301516242
## 8 0.782839347089918 -0.520906614581632 -0.00341647301516242
## 9 0.782839347089918 -0.520906614581632 -0.00341647301516242
```

d. All the columns of `odat` are not necessary for our analysis. So, obtain a subset by selecting only the columns long, lat, group, region, Murder, Assault, UrbanPop, and Rape. Store the data in `sdat` and report the first few rows.

```
sdat <- subset(odat, select = c(long, lat, group, region, Murder, Assault, UrbanPop, Rape))
head(sdat)
```

```
##        long     lat group  region         Murder           Assault
## 1 -87.46201 30.38968    1 alabama 1.24256408388112 0.782839347089918
## 2 -87.48493 30.37249    1 alabama 1.24256408388112 0.782839347089918
## 6 -87.52503 30.37249    1 alabama 1.24256408388112 0.782839347089918
## 7 -87.53076 30.33239    1 alabama 1.24256408388112 0.782839347089918
## 8 -87.57087 30.32665    1 alabama 1.24256408388112 0.782839347089918
## 9 -87.58806 30.32665    1 alabama 1.24256408388112 0.782839347089918
##             UrbanPop                       Rape
## 1 -0.520906614581632 -0.00341647301516242
## 2 -0.520906614581632 -0.00341647301516242
## 6 -0.520906614581632 -0.00341647301516242
## 7 -0.520906614581632 -0.00341647301516242
## 8 -0.520906614581632 -0.00341647301516242
## 9 -0.520906614581632 -0.00341647301516242
```

e. Melt the data frame `sdat` with id variables long, lat, group, region. Store the molten data in `msdat` and report the first few rows of data.

```
msdat <- melt(sdat, id=c("long","lat","group","region"))
head(msdat)
```

```
##        long     lat group  region variable            value
## 1 -87.46201 30.38968    1 alabama   Murder 1.24256408388112
## 2 -87.48493 30.37249    1 alabama   Murder 1.24256408388112
## 3 -87.52503 30.37249    1 alabama   Murder 1.24256408388112
## 4 -87.53076 30.33239    1 alabama   Murder 1.24256408388112
## 5 -87.57087 30.32665    1 alabama   Murder 1.24256408388112
## 6 -87.58806 30.32665    1 alabama   Murder 1.24256408388112
```

f. The molten data frame `msdat` is now ready to be plotted. Create a plot showing the USA state map, fill by value, and `facet_wrap` with variable. Please don't add any legend and make sure that facetting labels are identified so that we can compare the facetted plots.

```
library(maps)
library(scales)

map <- ggplot(msdat, aes(x=long, y=lat, fill=value, group=group)) + facet_wrap(msdat$variable)+
geom_polygon(fill="grey65", colour = alpha("white", 1/2), size = 0.2)+ theme_bw()
map + scale_fill_gradient(low="white", high="grey20")
```


g. Now examine the plot you have generated in question (f) and answer the following questions based on what you see in the plot.

```
\
i. For each crime, name two states with its highest rate.
ii. Do you think a larger urban population is indicative of a higher murder rate? Why or why no
t?
```

rape: nevada and california murder: georgia and missisipi assault: north carolina and florida

It is not indicative, as a state like missisipi has a lower urban pop but is the second highest murder rate.

h. In question (3b) we standardized the crime rates. Why do you think we did this? Explain what would happen if we did not standardize the data.

We did this because the number of arrests for each crime is different orders of magnitude. Assault happening over 10 times more then anything else would dominate the visual space of whatever form we graph it in.

i. In question (3c) we ordered the data after merging. Why do you think we had to do this? Explain what would happen if we did not.

One of the biggest reasons is that we use head to check our data. With the data not in order we cannot see if everything changed correctly.

4. Life expectancy data for four countries can be obtained from the world bank database found at github (http://mamajumder.github.io/data-science/data/life-expectancy.csv). It contains life expectancy in years for different genders. Now answer the following questions.

a. Read the data from the above link and display the first few rows of data.

```
url <- "http://mamajumder.github.io/data-science/data/life-expectancy.csv"
life.dat <- read.table(url, header=T, sep=",")
head(life.dat)
```

```
##   year    sex Bangladesh  India Pakistan  USA
## 1 1960 female     46.224 40.391   46.655 73.1
## 2 1960   male     47.787 42.329   46.223 66.6
## 3 1961 female     46.731 41.125   47.564 73.6
## 4 1961   male     48.445 43.052   47.156 67.1
## 5 1962 female     47.254 41.876   48.426 73.5
## 6 1962   male     49.104 43.784   48.044 66.9
```

b. Generate a plot showing trend lines of life expectancy by year. Color them by sex and facet by country. Include your code with the plot.

```
life.datA <- subset(life.dat, select = -c(India,Pakistan,USA))
life.datB <- subset(life.dat, select = -c(Bangladesh,Pakistan,USA))
life.datC <- subset(life.dat, select = -c(Bangladesh,India,USA))
life.datD <- subset(life.dat, select = -c(Bangladesh,India,Pakistan))
life.datA$country = 1
life.datB$country = 2
life.datC$country = 3
life.datD$country =4
colnames(life.datA) <- c("year","sex","life","country")
colnames(life.datB) <- c("year","sex","life","country")
colnames(life.datC) <- c("year","sex","life","country")
colnames(life.datD) <- c("year","sex","life","country")
life.dat <- rbind(life.datA,life.datB,life.datC,life.datD)
ggplot(life.dat,
       aes(x=year,y=life, color=sex)) +
  geom_point() +
  geom_smooth(method="lm", se=F) +facet_wrap(life.dat$country)
```
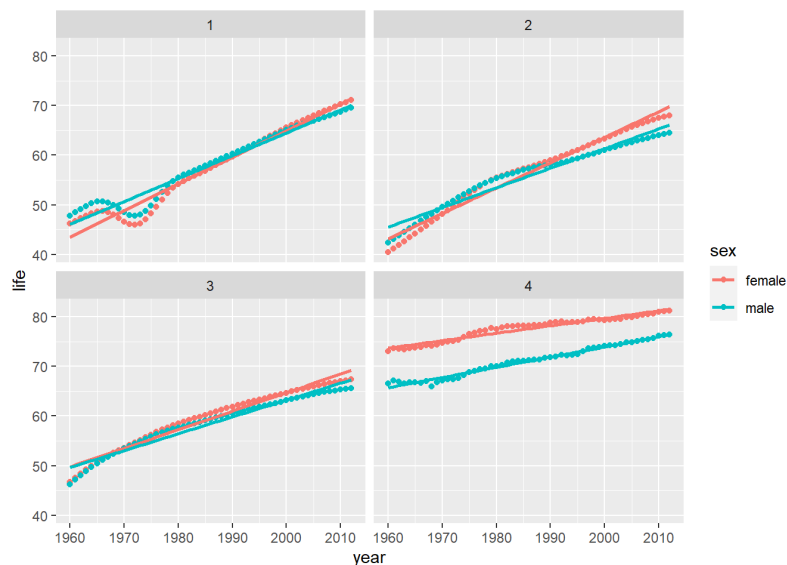
c. Explain what interesting features you noticed in the plot you made in question 4b. The USA was the only country where the different sexes had different life expectancy.

5. For the following questions please use data frame `tips`

a. Create a bar chart that shows the average tip by day.

```
avgtips = aggregate(x = tips$tip, by = list(tips$day), FUN=mean)
colnames(avgtips) <- c('Day', "Average_tip")
```

b. Compute the average tip, total tip, and average size grouped by smoker and day. i.e., For each combination of smoker and day you should have a row of these summaries. Report these results in a nice table.

```
avgtipsd = aggregate(x = tips$tip, by = list(tips$day,tips$smoker), FUN=mean)
colnames(avgtipsd) <- c('Day','smoker', "Average_tip")
totaltipd = aggregate(x = tips$tip, by = list(tips$day,tips$smoker), FUN=sum)
colnames(totaltipd) <- c('Day','smoker', "total_tip")
avgsized = aggregate(x = tips$size, by = list(tips$day,tips$smoker), FUN=mean)
colnames(avgsized) <- c('Day','smoker', "Average_size")
tipadd <- subset(totaltipd, select = c(total_tip))
sizeadd <- subset(avgsized, select = c(Average_size))
totaltipd <- as.data.frame(totaltipd)
tipadd <- as.data.frame(tipadd)
sizeadd <- as.data.frame(sizeadd)
avgtipsd$total_tip <- tipadd
avgtipsd$Average_size <- sizeadd
avgtipsd
```

```
##      Day smoker Average_tip total_tip Average_size
## 1   Fri     No    2.812500     11.25     2.250000
## 2   Sat     No    3.102889    139.63     2.555556
## 3   Sun     No    3.167895    180.57     2.929825
## 4  Thur     No    2.673778    120.32     2.488889
## 5   Fri    Yes    2.714000     40.71     2.066667
## 6   Sat    Yes    2.875476    120.77     2.476190
## 7   Sun    Yes    3.516842     66.82     2.578947
## 8  Thur    Yes    3.030000     51.51     2.352941
```

c. Create a bar chart that shows average tip by day, faceted by smoker.

```
ggplot(avgtipsd)+facet_wrap(avgtipsd$smoker)
```

No	Yes

d. In questions 5a and 5c, we plotted a summary of our data which does not show us the whole picture. In practice, we would like to see all of the data. What plot do you suggest would serve a similar purpose to the one in question 5c? In other words, what would be a better plot to show than tips by day, facetted by smoker? Please produce this plot and include your code.

```
'The best plot would be faceted by a grid, taking the base tips and faceting it by every possibl
e combination of factors'
```

```
## [1] "The best plot would be faceted by a grid, taking the base tips and faceting it by every
possible combination of factors"
```

6. We have the following data set:

```
myDat <- read.csv("http://mamajumder.github.io/data-science/data/reshape-source.csv")
kable(myDat)
```

player	track	walking	cycling
1	A	408	43
1	B	402	31
1	C	386	41
2	A	373	53
2	B	404	41
2	C	422	30
3	A	403	25
3	B	393	46
3	C	422	48

We want to reshape the data and produce the following output:

player	variable	A	B	C
1	walking	408	402	386
1	cycling	43	31	41
2	walking	373	404	422
2	cycling	53	41	30
3	walking	403	393	422
3	cycling	25	46	48

Provide code that will produce this desired output. Demonstrate your answer by displaying the output as well.

```
myDat <- as.data.frame(myDat)
myDat <- subset(myDat, select = -c(1))
myDatA <- subset(t(myDat), select = -c(4,5,6,7,8,9))
myDatB <- subset(t(myDat), select = -c(1,2,3,7,8,9))
myDatC <- subset(t(myDat), select = -c(1,2,3,4,5,6))
myDat <- rbind(myDatA,myDatB,myDatC)
playerorder <- c(1,1,2,2,3,3)
po <- t(t(playerorder))
'cbind(po,myDat)'
```

```
## [1] "cbind(po,myDat)"
```

```
kable(myDat)
```

	1	2	3
track	A	B	C
walking	408	402	386
cycling	43	31	41
track	A	B	C
walking	373	404	422
cycling	53	41	30
track	A	B	C
walking	403	393	422
cycling	25	46	48

7. Ordering the factor In class, we have seen how to order factors. Suppose we have the following data about a certain value obtained during particular months of the year;

```
month <- c("July", "June", "September", "May", "October", "August")
value <- c(35, 72, 14, 23, 60, 105)
df <- data.frame(month, value)
```

Now please do the following:

   a. Convert the month column of dataframe `df` into a factor column. Demonstrate that it is indeed converted into a factor column.

```
df$month <- factor(df$month)
is.factor(df$month)
```

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

   b. Now generate a bar chart showing the value for different months.

   c. Notice the order of the levels of the months is not natural, instead the plot shows the dictionary order. Now, order the bars according to the natural order of the levels of the class (months of the year as they appear in chronological order) and regenerate the bar graph.

8. Install the `babynames` package with `install.packages()`. This package includes data from the Social Security Administration about American baby names over a wide range of years. Generate a plot of the reported proportion of babies born with the name Angelica over time. Do you notice anything odd about the plotted data? (Hint: you should) If so, describe the issue and generate a new plot that adjusts for this problem. Make sure you show both plots along with all code that was used to generate them. x The graph jumps from really big to really small numbers, because we didn't seperate by gender.

```
baby.dat <- babynames::babynames
ggplot(subset(baby.dat, name %in% c("Angelica")),aes(x=year,y=n)) +geom_line(aes(color="white"))
```
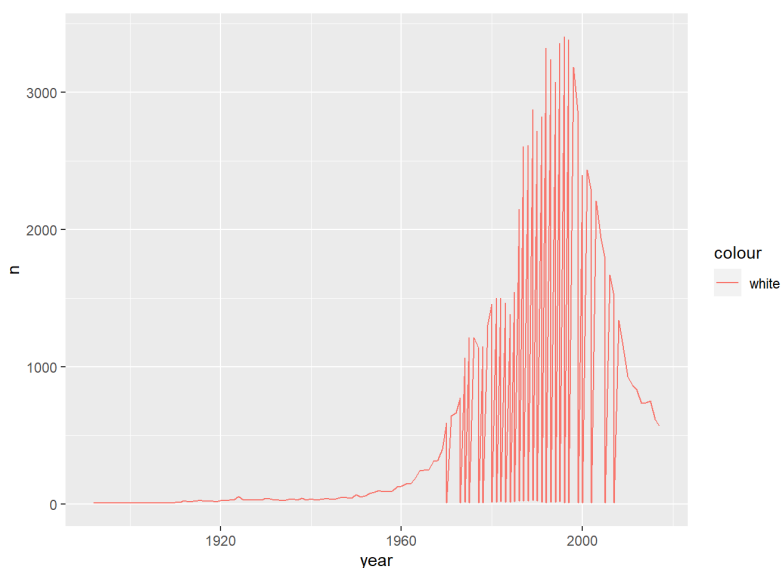


```
ggplot(subset(baby.dat, name %in% c("Angelica")),aes(x=year,y=n)) +geom_line(aes(color = sex))
```
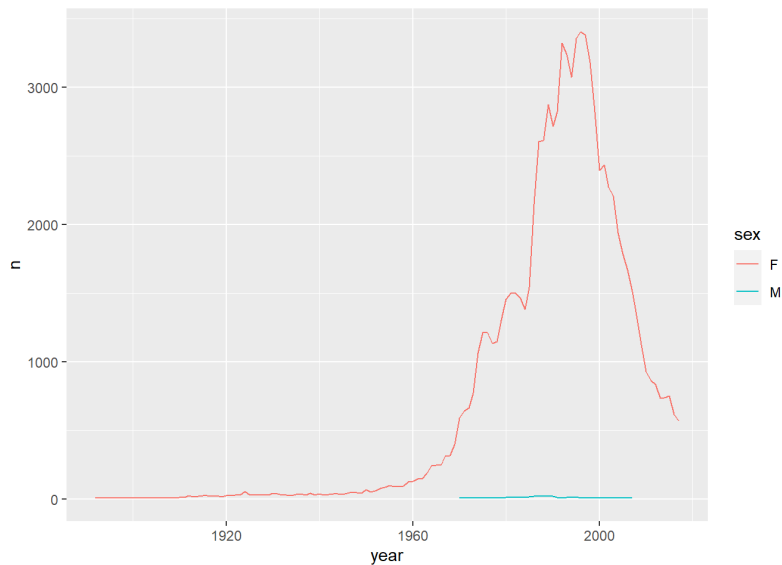
9. Bonus (2 points) for undergraduates and mandatory for graduate students. Suppose we have a vector of data as follows:

```
myVector <- c(-15, -10, -5, 0, 5, 10, 15, 20)
```

    a. Using the function `tapply()`, separately compute the means of the first three values, next two values, and the last three values of `myVector`. Show your code. Your result should be: -10.0, 2.5, 15.0.

```
ind <- list(c(1,1,1,2,2,3,3,3))
tapply(myVector,ind,mean)
```

```
##      1     2     3
## -10.0   2.5  15.0
```

    b. Now repeat question 9a, but instead of computing means, you will compute the sum of squares. Again, show your code. Your result should be: 350, 25, 725.

```
myvectorsquared <- myVector^2
tapply(myvectorsquared,ind,sum)
```

```
##    1   2   3
## 350  25 725
```