52414:Final_exam_R

Gil Shiloh

24 7 2021

Q0.Submission Instructions (Please read carefully)

The exam will be submitted **individually** by uploading the solved exam `Rmd` and `html` files to the course `moodle`.

Please name your files as `52414-HomeExam_ID.Rmd` and `52414-HomeExam_ID.html` where `ID` is replaced by your ID number (do **not** write your name in the file name or in the exam itself).

The number of points for each sub-question is indicated next to it, with 105 points overall. The total grade will be at most 100.

Once you click on the `moodle` link for the home exam, the exam will start and you have three days (72 hours) to complete and submit it.

The exam will be available from July 18th to July 30th. The last submission time is June 30th at 23:59.

You may use all course materials, the web and other written materials and R libraries.

You are NOT allowed to discuss any of the exam questions/materials with other students.

Analysis and Presentation of Results:

Write your answers and explanations in the text of the `Rmd` file (*not* in the `code`).

The text of your answers should be next to the relevant code, plots and tables and refer to them, and not at a separate place at the end.

You need to explain every step of your analysis. When in doubt, a more detailed explanation is better than omitting explanations.

Give informative titles, axis names and names for each curve/bar in your graphs.

In some graphs you may need to change the graph limits. If you do so, please include the outlier points you have removed in a separate table.

Add informative comments explaining your code

Whenever possible, use *objective* and *specific* terms and quantities learned in class, and avoid *subjective* and *general* unquantified statements. For example:

`Good:` "We see a 2.5-fold increase in the curve from Jan. 1st to March 1st".

`Bad:` "The curve goes up at the beginning".

`Good:` "The median is 4.7. We detected five outliers with distance > 3 standard deviations from the median".

`Bad:` "The five points on the sides seem far from the middle".

Sometimes `Tables` are the best way to present your results (e.g. when asked for a list of items). Exclude irrelevant

rows/columns. Display clearly items' names in your `Tables`.

Show numbers in plots/tables using standard digits and not scientific display.

That is: 90000000 and not 9e+06.

Round numbers to at most 3 digits after the dot - that is, 9.456 and not 9.45581451044

Some questions may require data wrangling and manipulation which you need to

decide on. The instructions may not specify precisely the exact plot you should use

(for example: `show the distribution of ...`). In such cases, you should decide what and how to show the results.

When analyzing real data, use your best judgment if you encounter missing values, negative values, NaNs, errors in the data etc. (e.g. excluding them, zeroing negative values..) and mention what you have done in your analysis in such cases.

Required libraries are called in the `Rmd` file. Install any library missing from your `R` environment. You are allowed to add additional libraries if you want.

If you do so, *please add them at the start of the Rmd file, right below the existing libraries, and explain what libraries you've added, and what is each new library used for*.

Q1. Two Armies Simulation (45 pt)

Consider two armies of 10 `R` loving statisticians and 10 `Python` loving statisticians, facing each other in a shootout, fighting to the death over which language is better.

Once the battle starts, assume that each statistician tries to shoot as fast as she can, where the time until shooting has an exponential distribution with $\lambda = 1$. After a shot is fired, the statistician keeps firing, with the time to the next shot again distributed as $exp(1)$. Each statistician keeps shooting until she is shot and killed herself by a statistician from the opposing army, and leaves the battle. The times until shooting the next bullet for all statisticians and all shots are independent.

At each shot, the statistician chooses as target **uniformly at random** a member from the remaining **living members** of the opposing army.

The battle keeps going until all persons from one of the armies die, and then the other army is declared the `winner` .

Let X be the number of remaining statisticians from the `winner` army when the battle ends.

Throughout this question, assume that statisticians are **perfect shooters**, and always hit their target (the choice of the target changes however between different sub-questions below).

a. (5pt) Describe in words a simulation strategy to estimate $E[X]$ and $Var(X)$, including how would you simulate a battle between the two armies.

Hint: remember that the exponential distribution has a memoryless property: $Pr(T > t) = Pr(T > t + s|T > s), \forall t, s > 0$.

You can perform the simulations in this question exactly as described, which may take many minutes to run, or perform **simpler** and **faster** simulations using probabilistic arguments, provided that they are **equivalent** to the description in the question.

(For example, if you were requested to simulate n i.i.d. $Bernouli(p)$ random variables and report their sum, you could argue that instead it is enough to simulate a single $Bionomial(n, p)$ random variable).

b. (8pt) Simulate $1,000$ random battles as described in the question and use them to estimate $E[X]$ and $Var(X)$ from the random simulations.

It is recommended to write a function for the simulation and call it, such that the simulation function can be used also in the subsequent sub-questions.

c. (8pt) Now, change n, the number of statisticians in each army, to be $n = 10, 20, 40, \ldots, 10240$ (each time multiplying n by two), and let X_n be the random variable counting the number of remaining winners when starting with n statisticians in each army. (so the variable X from (a.) corresponds to X_{10}).

For each value of n simulate 100 random battles and estimate $\mu_n \equiv E[X_n]$.

Plot your estimate vs. n.

Find a simple function $f(n)$ such that it holds that $\mu_n \approx f(n)$ based on the plot.

(**Hint:** you can use log-scale).

d. (8pt) In this sub-question, assume that all statisticians in both armies have used their programming language too much so they became to hate it, and therefore in each shot they aim and kill a random member from their **own** army (including possibly themselves).

Modify the simulation to accommodate this case, and repeat the simulation, plot and finding a function $f(n)$ as in (c.) for this case.

Explain in words the differences in results between the two cases.

e. (8pt) In this sub-question, assume that all statisticians in both armies are **completely drunk**, and shoot randomly one of the **remaining persons alive** (from both armies) including themselves (they still always hit their target).

Repeat (d.) for this case. Are the results similar or different? why?

f. (8pt) Finally, suppose in this sub-question that statisticians that are shot become zombies instead of being killed, and can still keep shooting at statisticians from the opposing army (as in (a.), (b.)).

All statisticians aim at and hit a random **living** (non-zombie) member from the opposing army. The battle ends when all members of a certain army become zombies, and then X_n records the number of remaining living (non-zombie) statisticians in the other army.

Repeat the simulation, plot and finding a function $f(n)$ as in (c.) for this case.

Explain in words the differences in results between the this and the previous cases.

Solutions:

Q2. Analysis and Visualization of Twitter Data (60 pt)

a. (4pt) Download and read the tweets dataset file `New-years-resolutions-DFE.csv` available here (https://github.com/DataScienceHU/DataAnalysisR_2021/blob/master/New-years-resolutions-DFE.csv).

The data represents new year's resolutions tweets by American users wishing to change something in their life at the stat of the year 2015, downloaded from here (https://data.world/crowdflower/2015-new-years-resolutions#).

Make sure that the tweets `text` column has `character` type.

Show the top and bottom two rows of the resulting data-frame.

b. (5pt) Create a new column with tweet times, of class `times` , with the time of the day for each tweet, in the format: `Hours:Minutes:Seconds` (see `DateTimeClasses` for more). For example, the first entry in the column corresponding to the time of the first tweet should be: `10:48:00` .

The class `times` stores and displays times in the above format, but also treats them as numeric values between zero and one in units of days. For example, the time `10:48:00` corresponds to the value: $(10 + 48/60)/24 = 0.45$.

Make a histogram showing the number of tweets in every hour of the 24 hours in a day (that is, the bins are times between `00:00` and `00:59` , between `01:00` and `01:59` etc.).

At which hours do we see the most/fewest tweets?

c. (6pt) Plot the distribution of tweets `text` lengths (in characters) made by `females` and `males` separately. Who writes longer tweets?

Repeat, but this time plot the tweets lengths distribution for tweets in the four different regions of the US

(`Midwest` , `Northeast` , `South` and `West`). Report the major differences in lengths between regions.

Finally, show the tweets lengths distribution for tweets for the 10 different categories given in `Resolution_Category` . Report the major differences in lengths between categories.

d. (8pt) Compute the number of occurrences of each word in the `text` of all the tweets. Ignore upper/lower case differences.

Remove words containing the special characters: `#` , `@` , `&` , `-` , `.` , `:` and `?` .

Remove also non-informative words: `resolution` , `rt` , `2015` and the empty word.

Plot the top 100 remaining words in a word cloud, using the `wordcloud2` package.

e. (8pt) Find for each of the top (most frequent) 100 words from 2.(d.) and each of the 10 tweet categories, the fraction of tweets from this category where the word appears, and list them in a 100×10 table F, with f_{ij} indicating the frequency of word i in category j.

That is, if for example there were 200 tweets in the category `Humor` , and 30 of them contained the word `joke` , then the frequency was 0.15.

Finally, for each of the 10 categories we want to find the most `characteristic` words, i.e. words appearing more frequently in this category compared to other categories:

Formally, compute for each word i and each category j the difference between the frequency in the category and the maximum over frequencies in other categories: $d_{ij} = f_{ij} - \max_{k \neq j} f_{ik}$.

(For example, if the word `joke` had frequency 0.15 in `Humor` , and the next highest frequency for this word in other categories is 0.1, then the difference for this word is 0.05).

Find for each category j of the 10 categories the 3 `characteristic` words with the highest differences d_{ij}. Show a table with the 10 categories and the 3 `characteristic` words you have found for each of them. Do the words make sense for the categories?

f. (5pt) Plot the number of tweets in each of the 10 categories shown in `Resolution_Category` .

Next, compute and show in a table of size 10×4 the number of tweets for each of the 10 categories from users in each of the four regions of the USA: `Midwest` , `Northeast` , `South` and `West` .

g. (8pt) We want to test the null hypothesis that users in different `regions` have the same distribution over `categories` for their resolutions, using the Pearson chi-square statistic:

$$

S = *{i=1}^{10}* {j=1}^{4}

$$

where o_{ij} is the number of tweets on category i from region j computed in the table in the previous sub-question, assuming some indexing for the categories and regions (for example, $j = 1, 2, 3, 4$ for `Midwest` , `Northeast` , `South` and `West` , respectively, and similarly for the categories). The expected counts e_{ij} are given by:

$$

e_{ij} = {o_{}}

$$

where $o_{i\bullet}$ is the sum over the i'th row (over all regions), $o_{\bullet j}$ the sum over the j'th column (over all categories) and $o_{\bullet\bullet}$ the sum over all observations in the table. These expected counts correspond to independence between the row (categories) and column (regions) according to the null hypothesis.

Compute and report the test statistic for the table computed in 2.(f).

Use the approximation $S \sim \chi^2(27)$ to compute a p-value for the above test (there are $(4-1) \times (10-1) = 27$ degrees of freedom). Would you reject the null hypothesis?

Finally, repeat the analysis (computing a table, χ^2-statistic and p-value) but this time split tweets by `gender` (`male` and `female`) instead of by `region` , to get a 10×2 table. Is there a significant difference in the distribution of categories between males and females?

h. (8pt) Use the following simulation to create a randomized dataset of `(category, region)` pairs for the tweets:

For each tweet in the dataset keep the real `category` (from the column `Resolution_Category`) but change the `region` randomly by shuffling (permuting) the regions column in a random order, such that the total number of tweets from each region remains the same.

Repeat this simulation $N = 1,000$ times, each time creating a new shuffled random data, with the `category` column remaining the same and the `region` column shuffled each time in a random order.

For each such simulation indexed i compute the `category` -by- `region` occurance table and the resulting χ^2 test statistic from 2.(g.) and call it S_i .

Plot the empirical density distribution of the S_i randomized test statistics and compare it to the theoretical density of the $\chi^2(27)$ distribution. Are the distributions similar?

Finally, compute the empirical p-value, comparing the test statistic S computed on the real data in 2.(g.) to the $1,000$ random statistics:

$$

= *{i=1}^N 1*{{S_i S}}.

$$

How different from the p-value obtained via the chi-square approximation?

i. (8pt) Compute for each of the 50 states (and `DC` - District of Columbia) in the US the number of tweets made by users from this state.

Next, load the `usmap` library that contains the variable `statepop` .

Use this variable to compute the number of tweets per million residents for each state.

Remove `DC` and use the `usmap` package to make a map of USA states, where each state is colored by the number of tweets per million residents.

Report the three states with the maximal and minimal number.

Solutions:

1

a

The strategy simulation for finding E[x] will be: I will create 2 vectors of 10 which distributed by exp(1), the first shooter will be the value with the minimum time (to shoot). The first shooter will reduce(kill) the other vector by 1 in a random spot by (uniformly distribution). and than I will random again for the shooter exp(1) plus the time for the first shot. I will run this simulation until one vector will be 0.I will save the survivors each time in a new vector. E[x] will be the average number of survivors every time. var[x] will be the distance of each number in the vector of survivors from the average power by 2.

b

Simulate 1,000 random battles as described in the question and use them to estimate E[X] and Var(X) from the random simulations

```
battle = function(n_1){
  team1 <- rexp(n_1,1)
  team2 <- rexp(n_1,1)
  while((length(team1) > 0) & (length(team2) > 0)){
    if (min(team1) < min(team2)){
      x <- rdunif(1,1,length(team2))
      team2 <- team2[-x]
      team1[which.min(team1)] <- team1[which.min(team1)] + rexp(1)
    }
    else{
      x <- rdunif(1,1,length(team1))
      team1 <- team1[-x]
      team2[which.min(team2)] <- team2[which.min(team2)] + rexp(1)
    }
  }
  return(max((length(team1)),length(team2)))
}
```

Run the simulation k times for 2 pairs of teams of 10.

```
res <- c()
for( i in (1:1000)){
  res[i] <- battle(10)
}
mean_battle <- mean(res)
var_battle <- var(res)
mean_and_var <- data.frame(mean=round(mean_battle,3),var=round(var_battle,3))
knitr::kable(mean_and_var, caption = "Mean & Var")
```

Mean & Var

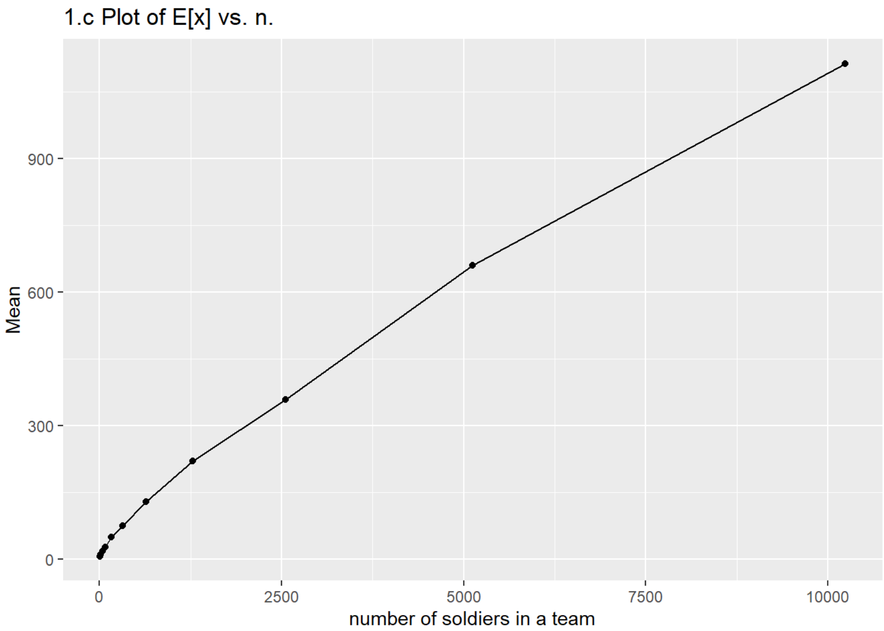
mean	var
5.642	5.225

c

Now, change n, the number of statisticians in each army, to be n=10,20,40,…,10240 (each time multiplying n by two), and let Xn be the random variable counting the number of remaining winners when starting with n statisticians in each army. (so the variable X from (a.) corresponds to X10).

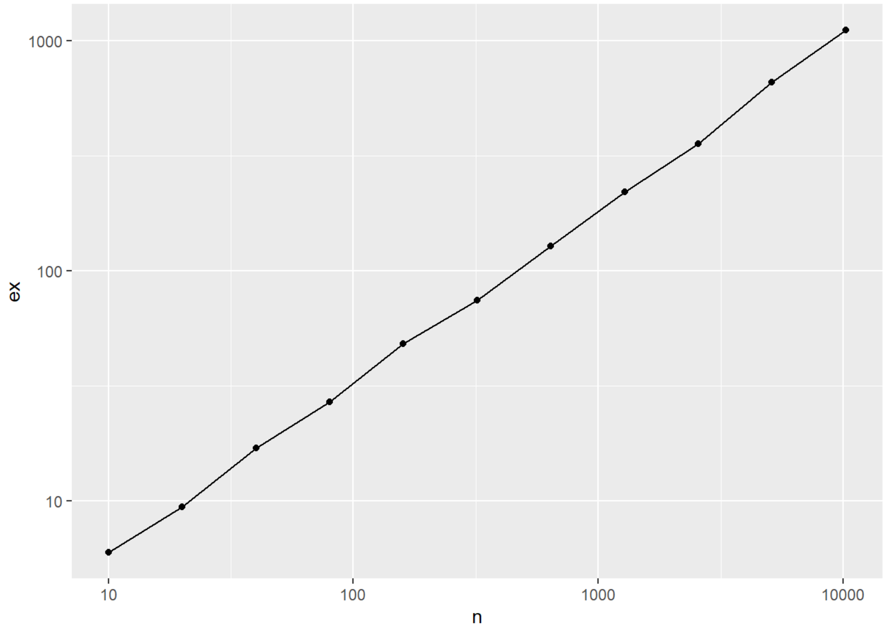
```
ex <- c()
res_c <- c()
n <- c(10,20,40,80,160,320,640,1280,2560,5120,10240)
for( i in 1:length(n)){
  for( j in (1:100)){
    res_c[j] <- battle(n[i])}
    ex[i] <- mean(res_c)
}
df_ex <- as.data.frame(cbind(n,ex))
```

```
ggplot(data= df_ex, aes(x=n,y=ex)) + geom_line() +
    geom_point() + labs(title="1.c Plot of E[x] vs. n.") + xlab("number of soldiers in a team") + ylab("Mean")
```



based on the plot I'll do a scale of y = log(x) to get a linear line.

```
ggplot(df_ex) + aes(x = n, y = ex) + geom_point() + geom_line() + scale_x_log10() + scale_y_log10()
```



d

In this sub-question, assume that all statisticians in both armies have used their programming language too much so they became to hate it, and therefore in each shot they aim and kill a random member from their own army (including possibly themselves).

```
battle_hate = function(n_1){
  team1 <- rexp(n_1,1)
  team2 <- rexp(n_1,1)
  while((length(team1) > 0) & (length(team2) > 0)){
    if (min(team1) < min(team2)){
      team1[which.min(team1)] <- team1[which.min(team1)] + rexp(1)}
    else{team2[which.min(team2)] <- team2[which.min(team2)] + rexp(1)}
    x <- rdunif(1,1,length(team1)+length(team2))
    if(x>length(team1)){x <- x-length(team1)
      team2 <- team2[-x]}
      else{team1 <- team1[-x]}}
  return(max(length(team1),length(team2)))}
```

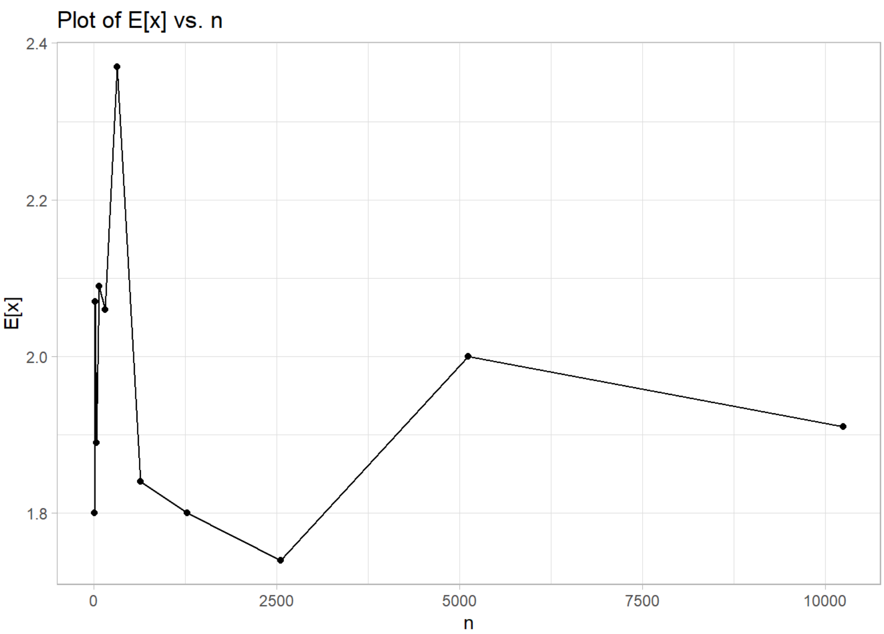
```
res_d <- c()
for( i in (1:1000)){
  res_d[i] <- battle_hate(10)
}
mean_battle_hate <- mean(res_d)
var_battle_hate <- var(res_d)
mean_var_hate <- data.frame(mean=round(mean_battle_hate,3),var=round(var_battle_hate,3))
knitr::kable(mean_var_hate, caption = "Mean & Var")
```

Mean & Var

mean	var
1.841	1.267

```
expt <- c()
res_d_2 <- c()
for( i in 1:length(n)){
  for( j in (1:100)){
    res_d_2[j] <- battle_hate(n[i])}
  expt[i] <- mean(res_d_2)
}
E_x_d <- data.frame(n = n, mean = expt)
```

```
ggplot(data= E_x_d, aes(x=n,y=expt)) + geom_line() +
    geom_point() + theme_light() + labs(title="Plot of E[x] vs. n") + xlab("n") + ylab("E[x]")
```



In this sub quastion each army is shooting at their own army, so there's no interaction between the armys. each army has the same distribution of shooting rate so I would expect to see the same resaults in both of the armys. In the graph I can see that the diffence between the E[x] of the last quastion to this one is bigger than 4. Because in the last quastion there was an interaction between the teams so the first shooter gives an advantage to his team.

e

In this sub-question, assume that all statisticians in both armies are completely drunk, and shoot randomly one of the remaining persons alive (from both armies) including themselves (they still always hit their target).

```
battle_drunk = function(n_1){
  team1 <- rexp(n_1,1)
  team2 <- rexp(n_1,1)
  all_team <- cbind(team1,team2)
  while((length(team1) > 0) & (length(team2) > 0)){
    if (min(team1) < min(team2)){
      team1[which.min(team1)] <- team1[which.min(team1)] + rexp(1)}
    else{team2[which.min(team2)] <- team2[which.min(team2)] + rexp(1)}
      x <- rdunif(1,1,length(team1)+length(team2))
    if(x>length(team1)){x <- x-length(team1)
      team2 <- team2[-x]}
    else{team1 <- team1[-x]}}
  return(max(length(team1),length(team2)))}
```

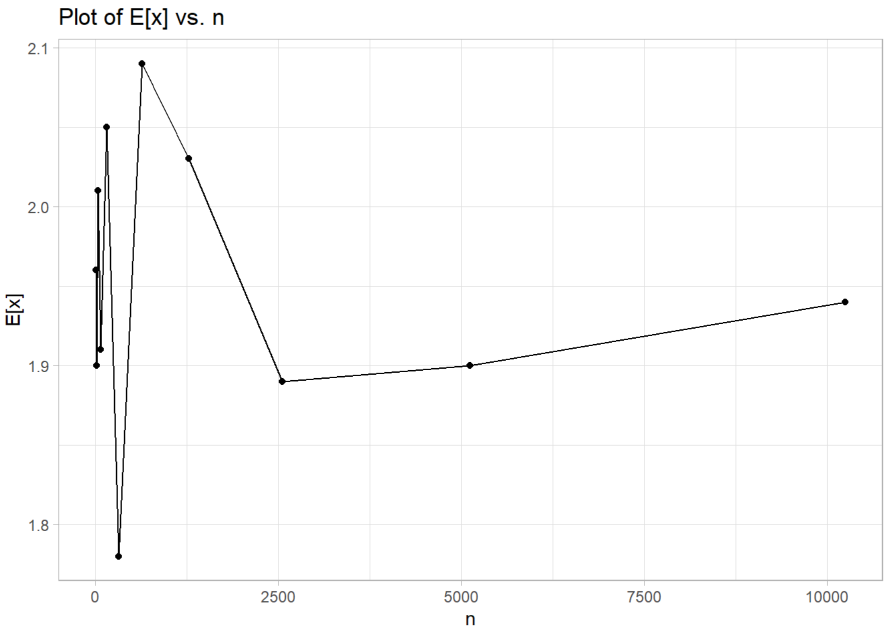
```r
res_drunk <- c()
for( i in (1:1000)){
  res_drunk[i] <- battle_drunk(10)
}
mean_battle_drunk <- mean(res_drunk)
var_battle_drunk <- var(res_drunk)
mean_var_drunk <- data.frame(mean=round(mean_battle_drunk,3),var=round(var_battle_drunk,3))
knitr::kable(mean_var_drunk, caption = "Mean & Var")
```

Mean & Var

mean	var
1.896	1.459

```r
ex_e <- c()
res_d_3 <- c()
for( i in 1:length(n)){
  for( j in (1:100)){
    res_d_3[j] <- battle_drunk(n[i])}
  ex_e[i] <- mean(res_d_3)
}
E_x_e <- data.frame(n = n, mean = ex_e)
```

```r
ggplot(data= E_x_e, aes(x=n,y=ex_e)) + geom_line() +
    geom_point() + theme_light() + labs(title="Plot of E[x] vs. n") + xlab("n") + ylab("E[x]")
```



Are the results similar or different? why? As the resault shows I can understand that in this case the resault is not coming from a known function for us. in this case evryone can shoot at everyone from both of the teams. so the difference from d is that there's more choice of who to shoot, from those reasons I got more variety in that case.

f

Finally, suppose in this sub-question that statisticians that are shot become zombies instead of being killed, and can still keep shooting at statisticians from the opposing army (as in (a.), (b.)).

```r
battle_zom = function(n_1){
  zom_1 <- 0
  zom_2 <- 0
  team1 <- rexp(n_1,1)
  team2 <- rexp(n_1,1)
  while(zom_1 < n_1 & zom_2 < n_1){
    if (min(team1) < min(team2)){
      zom_2 <- zom_2 + 1
      team1[which.min(team1)] <- team1[which.min(team1)] + rexp(1)
    }
    else{
      zom_1 <- zom_1 +1
      team2[which.min(team2)] <- team2[which.min(team2)] + rexp(1)
    }
  }
  return(n_1 - min(zom_1,zom_2))
}
```

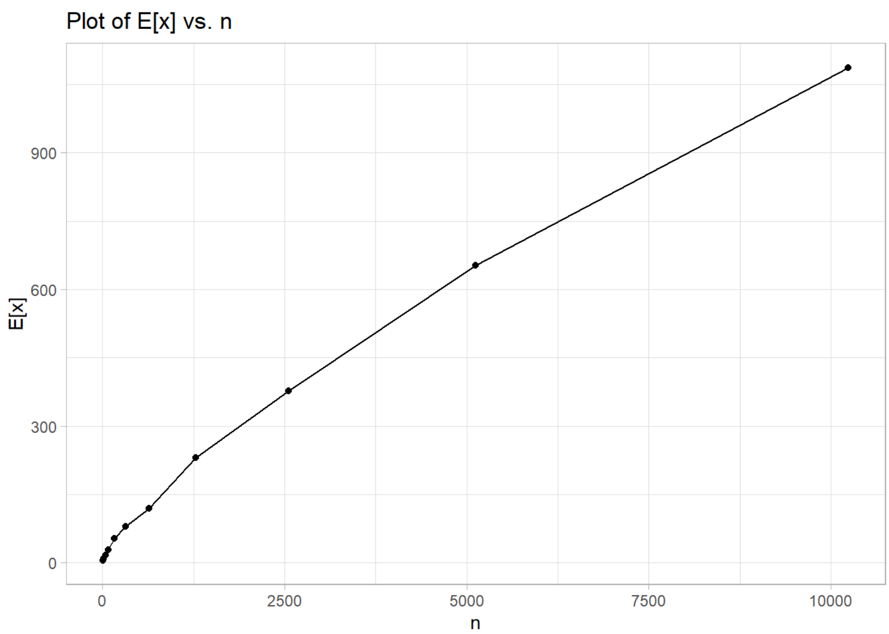
```
res_zom <- c()
for( i in (1:1000)){
  res_zom[i] <- battle_zom(10)
}
mean_battle_zom <- mean(res_zom)
var_battle_zom <- var(res_zom)
mean_var_zom <- data.frame(mean=round(mean_battle_zom,3),var=round(var_battle_zom,3))
knitr::kable(mean_var_zom, caption = "Mean & Var")
```

Mean & Var

mean	var
3.459	4.008

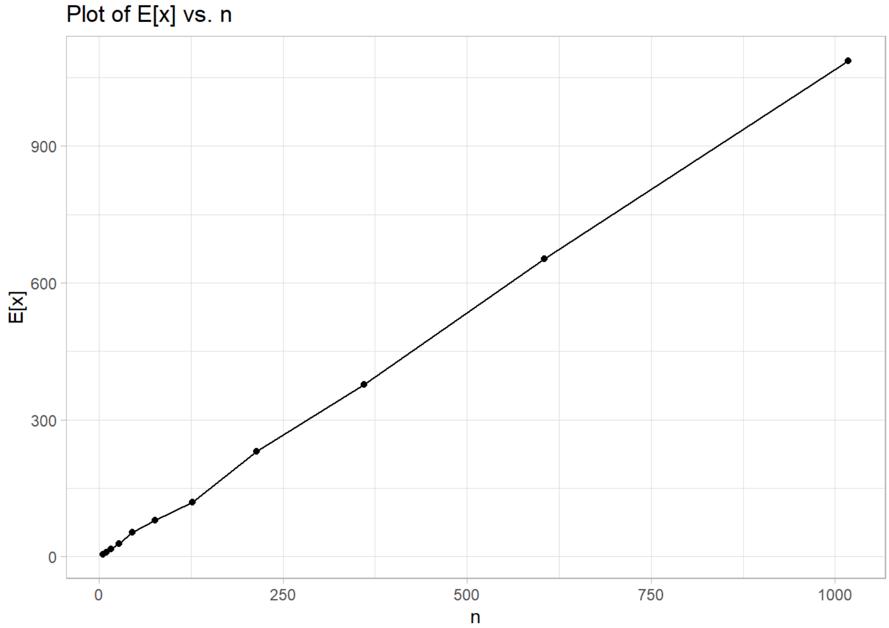
```
ex_zom <- c()
res_zom <- c()
x_zom <- c(10,20,40,80,160,320,640,1280,2560,5120,10240)
for( i in 1:length(x_zom)){
  for( j in (1:100)){
    res_zom[j] <- battle(x_zom[i])}
  ex_zom[i] <- mean(res_zom)
}
df_zom <- as.data.frame(cbind(x_zom,ex_zom))
```

```
ggplot(data = df_zom, aes(x=x_zom,y = ex_zom)) + geom_line() +
    geom_point() + theme_light() + labs(title="Plot of E[x] vs. n") + xlab("n") + ylab("E[x]")
```



I can see that the graph comes from the function of log(x) and log(y) so the transaction that I will do will be x^(3/4). So I'll make the nacessery transaction.

```
ggplot(data = df_zom, aes(x=x_zom^(3/4),y = ex_zom)) + geom_line() +
    geom_point() + theme_light() + labs(title="Plot of E[x] vs. n") + xlab("n") + ylab("E[x]")
```



The difference this time is that people who died are turning into zombies and can continue shooting. The difference from the first battle is that this time the zombies continue to shoot so theres no adventage of being the first one to shoot, so I would expect the E[x] to be lower and the var[x] to be lower too. This time there's an interaction between the teams but it's less important because the zombies continue to shoot so from that reason the E[x] and the var[x] are higher than the last battle.

2

a

```
data_new_year <- data.frame(read.csv('C://Users/97254/Downloads/New-years-resolutions-DFE.csv'))
attach(data_new_year)
text <- as.character(text)
class(text)
```

```
## [1] "character"
```

```
knitr::kable(head(data_new_year,2))
```

other_topic	resolution_topics	gender	name	Resolution_Category	retweet_count	text	tweet_coord	tweet_created	twe
Read moore books, read less facebook.	Eat healthier	female	Dena_Marina	Health & Fitness	0	#NewYearsResolution :: Read more books, No scrolling FB/checking email b4 breakfast, stay dedicated to PT/yoga to squash my achin' back!		12/31/14 10:48	12/
	Humor about Personal Growth and Interests Resolutions	female	ninjagirl325	Humor	1	#NewYearsResolution Finally master @ZJ10 's part of Kitchen Sink		12/31/14 10:47	12/

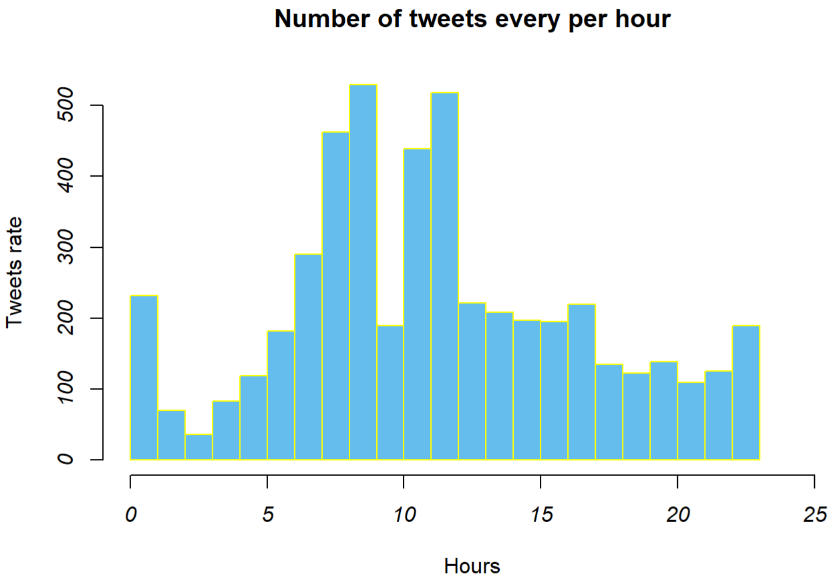
```
knitr::kable(tail(data_new_year,2))
```

	other_topic	resolution_topics	gender	name	Resolution_Category	retweet_count	text	tweet_coord	tweet_create
5010		Join a startup	female	itsmeJajael	Career	NA	RT @kscmaghirang: To have an excellent job before or after graduation #NewYearsResolution		12/31/14 9:48
5011	humor on resolutions	Improve my body	female	_LeahHarrell	Health & Fitness	NA	RT @tompycan: #NewYearsResolution on Jan1: "I'm really going to get in shape this year!"		

on Jan3: "I'm learning to love my body the way it‰_ | |12/31/14 9:51 |12/31/14 | 550348000000000000|shenandoah conservatory |VA |Eastern Time (US & Canada) |South |

b

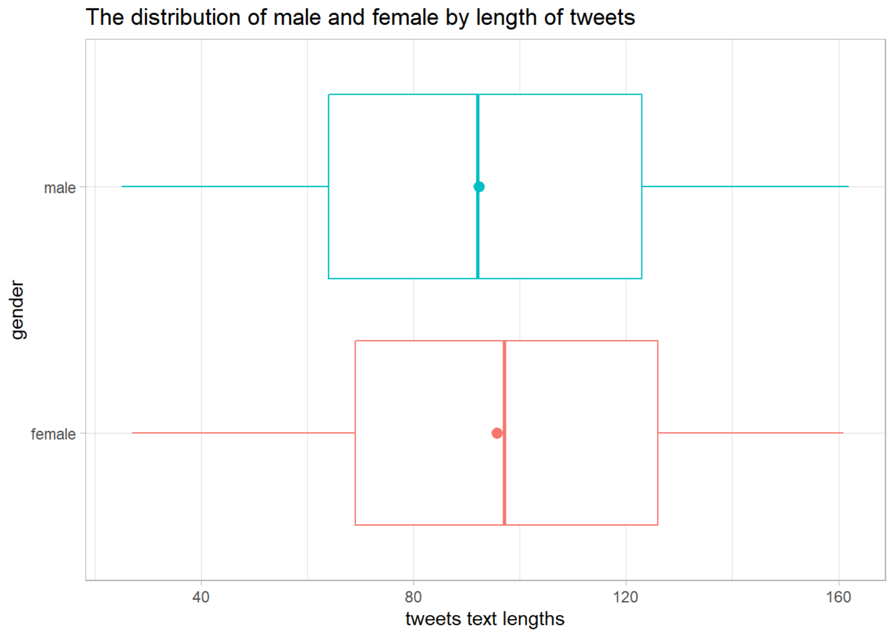
```
data_new_year$clock <- as.chron(tweet_created, "%m/%y/%d %H:%M")
data_new_year$clock <- as.times(data_new_year$clock)
hist(hours(data_new_year$clock), breaks=24, main="Number of tweets every per hour",ylab = "Tweets rate", xlab = "Hours" ,col
='#65BDED', border="yellow", font.axis=3,xlim = c(0,25))
```



From the graph I can understand that the time that people tweet the most is between 8:00-10:00 in the morning and after a break of an hour they are tweeting strong in their lunch break from 11:00 - 13:00.

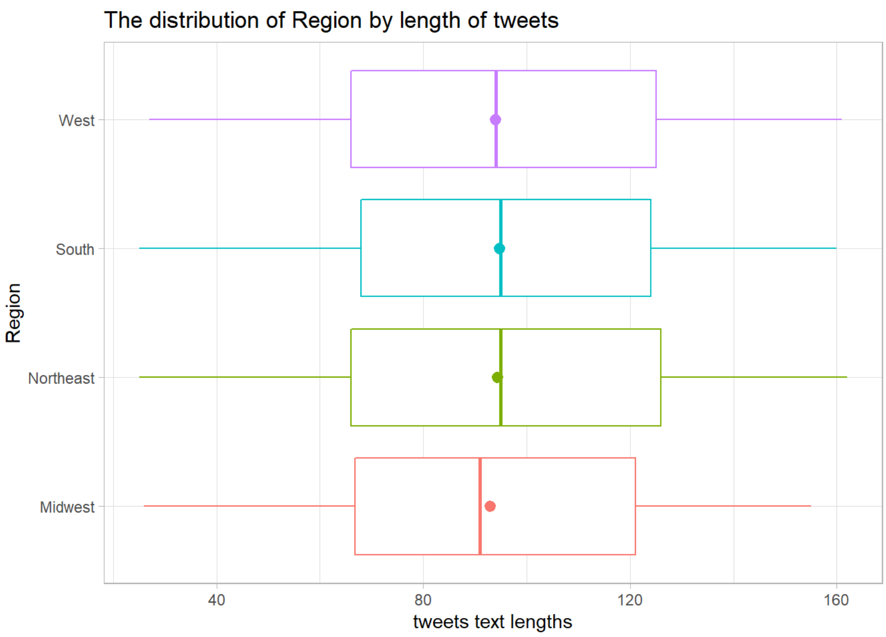
c

```
data_new_year$length <- nchar(data_new_year$text)
ggplot(data = data_new_year,aes(x=gender,y = length, col=gender))+
  geom_boxplot()+
  theme_light()+ stat_summary(fun="mean") +
  theme(legend.position="none") +
  xlab("gender")+
  ylab("tweets text lengths")+
  coord_flip()+
  labs(title="The distribution of male and female by length of tweets")
```



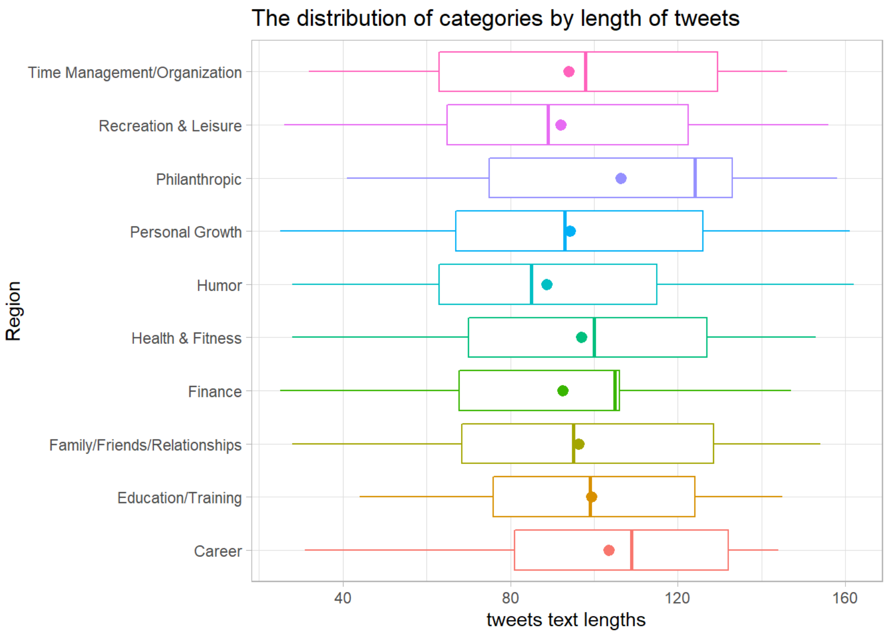
I can see that the average is similar between male and female. there is difference of 3 in advantage of the females.

```
data_new_year$length <- nchar(data_new_year$text)
ggplot(data = data_new_year,aes(x=tweet_region,y = length, col=tweet_region))+
  geom_boxplot()+
  theme_light()+ stat_summary(fun="mean") +
  theme(legend.position="none") +
  xlab("Region")+
  ylab("tweets text lengths")+
  coord_flip()+
  labs(title="The distribution of Region by length of tweets")
```



The average of length of tweets between the Regions is similar between the 4 regions.

```
data_new_year$length <- nchar(data_new_year$text)
ggplot(data = data_new_year,aes(x=Resolution_Category,y = length, col=Resolution_Category))+
  geom_boxplot()+
  theme_light()+ stat_summary(fun="mean") +
  theme(legend.position="none") +
  xlab("Region")+
  ylab("tweets text lengths")+
  coord_flip()+
  labs(title="The distribution of categories by length of tweets")
```



The average length of philantropic category is highest between all categories. and the average length of tweets of Humor is the smallest.

d

```
text <- sort(table(unlist(strsplit(tolower(data_new_year$text), " "))), decreasing = TRUE)
clean_words <- setNames(data.frame(text), c("word", "count")) %>% anti_join(stop_words, by = "word")
clean_words <- clean_words[!str_detect(pattern ="#|@|&|\\.|-|:|\\?|\\s|\\d|\\!", string = clean_words$word),]
clean_words <- clean_words %>% filter(!clean_words$word %in% c("resolution","rt","2015"))
clean_words$word <-  str_replace_all(clean_words$word,"[^[:alnum:]]", "")
wordcloud2(data = clean_words[1:100,], shape = 'cardiod', gridSize = 10,backgroundColor="pink")
```

e

```
count_cat <- data_new_year %>% group_by(Resolution_Category) %>% summarise(number = n())
data_freq <- data.frame(matrix(NA, nrow =100, ncol = 10))
data_freq <- setNames(data_freq, as.vector(count_cat$Resolution_Category))
rownames(data_freq) <- make.names(clean_words$word[1:100], unique = TRUE)
for(i in 1:10){
  for(j in 1:100){
    c <- 0
    for(k in 1:5011){
      if ((grepl(row.names(data_freq)[j], data_new_year[k,7], fixed = TRUE)) & (data_new_year[k,5] == colnames(data_freq)
[i])){
        c <- c+1
      }
    }
    data_freq[j,i] <- c/count_cat[i,2]
  }
}
```

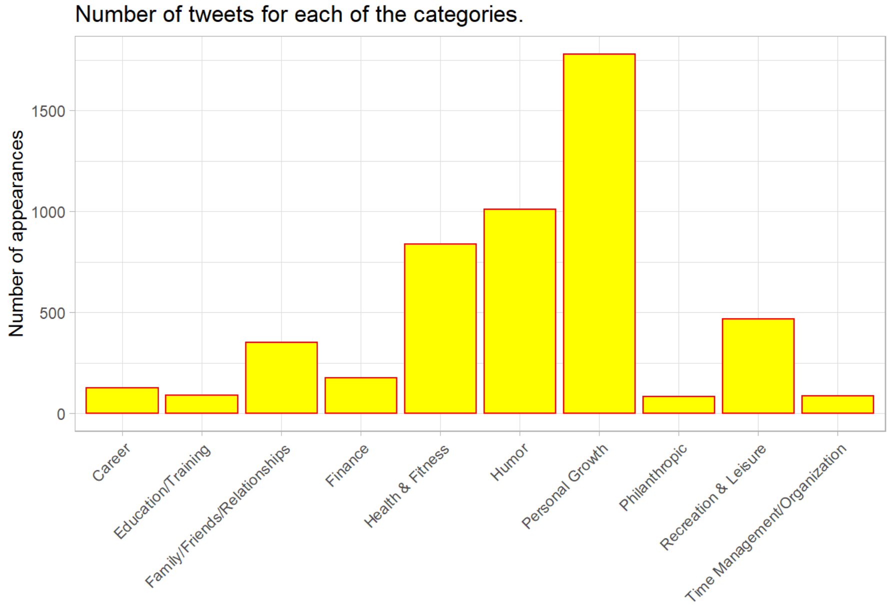
```
data_freq_1 <- data_freq
for (i in 1:100){for(j in 1:10){data_freq_1[i,j] <- data_freq_1[i,j] - max(data_freq_1[i,-j])}}
top_3 <- data.frame(matrix(NA, nrow =3, ncol = 10))
colnames(top_3) <- as.vector(count_cat$Resolution_Category)
for (i in 1:10) {top_3[,i] <- rownames(top_n(data_freq_1,3, data_freq_1[i])[i])}
top_3
```

```
##   Career Education/Training Family/Friends/Relationships Finance
## 1  start              learn                       people   money
## 2   goal           continue                      friends   spend
## 3    job                ass                       family    more
##   Health & Fitness      Humor Personal Growth Philanthropic
## 1              eat resolution            stop         tweet
## 2              gym         קג          person          plan
## 3              fit          ק      resolution         daily
##   Recreation & Leisure Time Management/Organization
## 1                watch                         time
## 2                 more                   resolution
## 3                 game                           im
```

For most of the categories the resaults make sence expect of: Education/ training which we have the word “ass” the word is inappropriate for aducation but make sence for training.

f

```
ggplot(count_cat, aes(x=Resolution_Category, y=number))+ theme_light() +
  geom_bar(color='red', fill='yellow',stat = "identity")+ theme(axis.text.x = element_text(angle = 45, hjust = 1)) +xlab("")
+ ylab("Number of appearances")+ labs(title="Number of tweets for each of the categories.")
```




```
table(Resolution_Category,tweet_region)
```

```
##                               tweet_region
## Resolution_Category            Midwest Northeast South West
##   Career                            24        30    44   28
##   Education/Training                14        19    33   23
##   Family/Friends/Relationships      72        79    97  103
##   Finance                           38        32    65   41
##   Health & Fitness                 191       170   284  195
##   Humor                            201       226   300  283
##   Personal Growth                  352       375   592  462
##   Philanthropic                     12        19    34   19
##   Recreation & Leisure              96       101   146  124
##   Time Management/Organization      20        22    26   19
```

g

```
chisq_region <-chisq.test(table(Resolution_Category,tweet_region))
chisq_region
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(Resolution_Category, tweet_region)
## X-squared = 26.366, df = 27, p-value = 0.4984
```

Pval > alpha=0.05 => the Pval is bigger than 0.05 so I will not reject the null hypothesis.

```
table(Resolution_Category,gender)
```

```
##                               gender
## Resolution_Category            female male
##   Career                           46   80
##   Education/Training               44   45
##   Family/Friends/Relationships    188  163
##   Finance                          96   80
##   Health & Fitness                467  373
##   Humor                           369  641
##   Personal Growth                 975  806
##   Philanthropic                    42   42
##   Recreation & Leisure            216  251
##   Time Management/Organization     50   37
```

```
chisq_gender <- chisq.test(Resolution_Category,gender)
chisq_gender
```

```
##
##  Pearson's Chi-squared test
##
## data:  Resolution_Category and gender
## X-squared = 116.67, df = 9, p-value < 0.00000000000000022
```

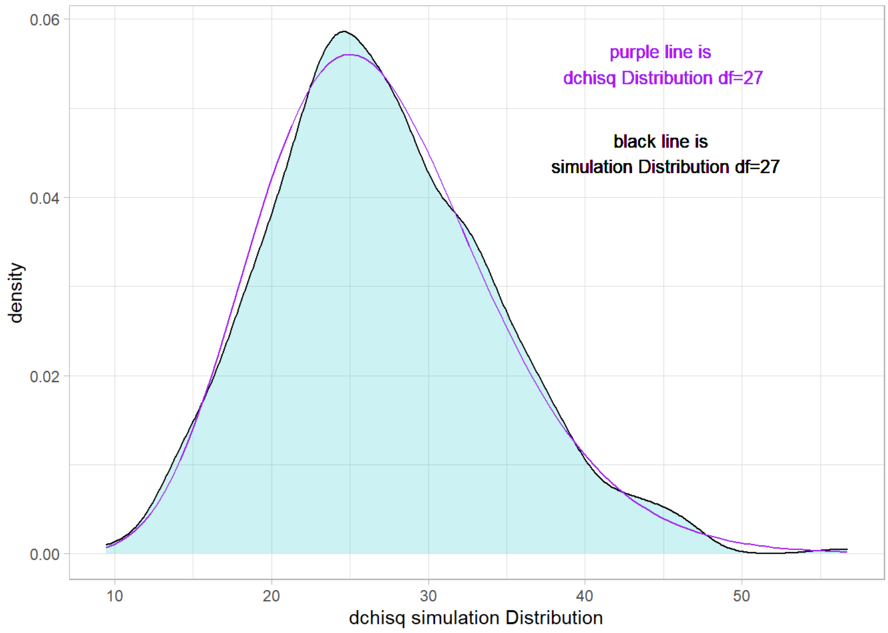
Pval < 0.05 = alpha => the Pval is a lot smaller than alpha so I will reject the null hypothesis. From that reason I can understand that there is a significant difference in the distribution of categories between males and females.

h

```
randomize <- function(){
  regions <- tweet_region
  copy <- data_new_year
  for(i in seq(length(rownames(data_new_year)))){
    r <- rdunif(1,1,length(regions))
    copy[i,15] <- regions[r]
    regions <- regions[-r]
  }
  t_random <- table(copy$Resolution_Category,copy$tweet_region)
  return(chisq.test(t_random)$statistic)
}
```

```
sims <- c()
for(j in seq(1000)){
  sims[j] <- randomize()
}
```

```
ggplot(data = as.data.frame(sims),aes(x=sims))+
  geom_density(fill="#00Bfc4",alpha=0.2)+ geom_text(y=.055, x=45, label="purple line is \n dchisq Distribution df=27 ",size
 = 3.5, color = "purple") + geom_text(x=45, y=0.045, label="black line is \n simulation Distribution df=27",size = 3.5, colo
r = "black") + theme_light() +stat_function(fun = dchisq, args = list(df = 27),col='purple') + xlab("dchisq simulation Distr
ibution")
```



As seen from the graph the distributions are very similar and if I'll run the simulation more times I am sure eventully it will be the same.

```
p_val <- length(sims[sims>=chisq_region$statistic])/1000
p_val
```

```
## [1] 0.499
```

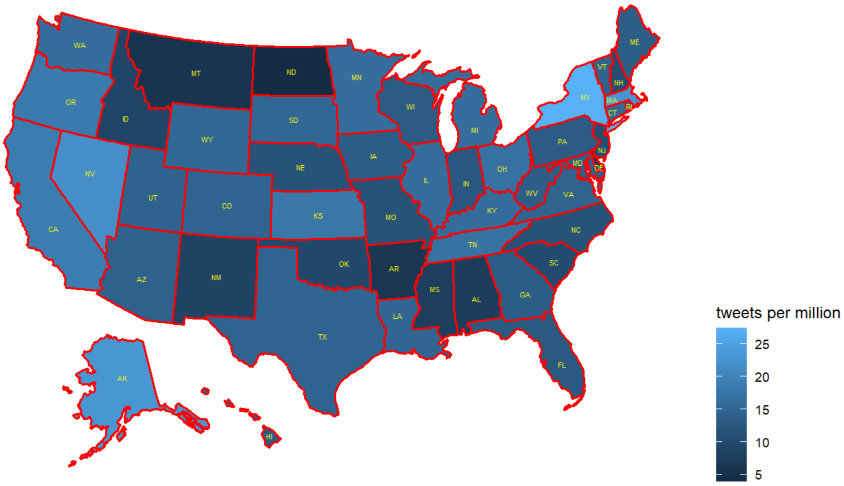
```
diff <- p_val - chisq_region$p.value
diff
```

```
## [1] 0.0006139286
```

I accpect the null hypothesis for both of the cases, pval is lower than the chisq test but have a small difference.

i

```
states <- data_new_year %>% count(tweet_state)
statepop$per_million <- statepop$pop_2015/1000000
colnames(states) <- c("abbr", "n")
ready_1 <- full_join(states, statepop, by = "abbr")
ready_1$tweet_per_mil <- ready_1$n/ready_1$per_million
ready_1 <- ready_1[-8,]
lit_map <- plot_usmap(data = ready_1,values = "tweet_per_mil",label_color = "yellow", exclude = "DC", color = "red", labels
 = TRUE, size = 0.6)+ scale_fill_continuous(name = "tweets per million") + theme(legend.position = "right")
lit_map$layers[[2]]$aes_params$size <- 1.5
print(lit_map)
```




```
knitr::kable(top_n(ready_1,3, tweet_per_mil), caption = "Top 3 states")
```

Top 3 states

abbr	n	fips	full	pop_2015	per_million	tweet_per_mil
AK	17	02	Alaska	738432	0.738432	23.02175
MA	156	25	Massachusetts	6794422	6.794422	22.96001
NY	543	36	New York	19795791	19.795791	27.43007

```
min <- tail(ready_1[order(ready_1$tweet_per_mil,decreasing = TRUE),],3)
knitr::kable(min, caption = "Min 3 states")
```

Min 3 states

	abbr	n	fips	full	pop_2015	per_million	tweet_per_mil
9	DE	6	10	Delaware	945934	0.945934	6.342937
27	MT	6	30	Montana	1032949	1.032949	5.808612
29	ND	3	38	North Dakota	756927	0.756927	3.963394