R_Coursework_2022

Leo Jones

2022-12-08

Q1. Reading the data into R:

```
dat <- read.table("prob.txt")
```

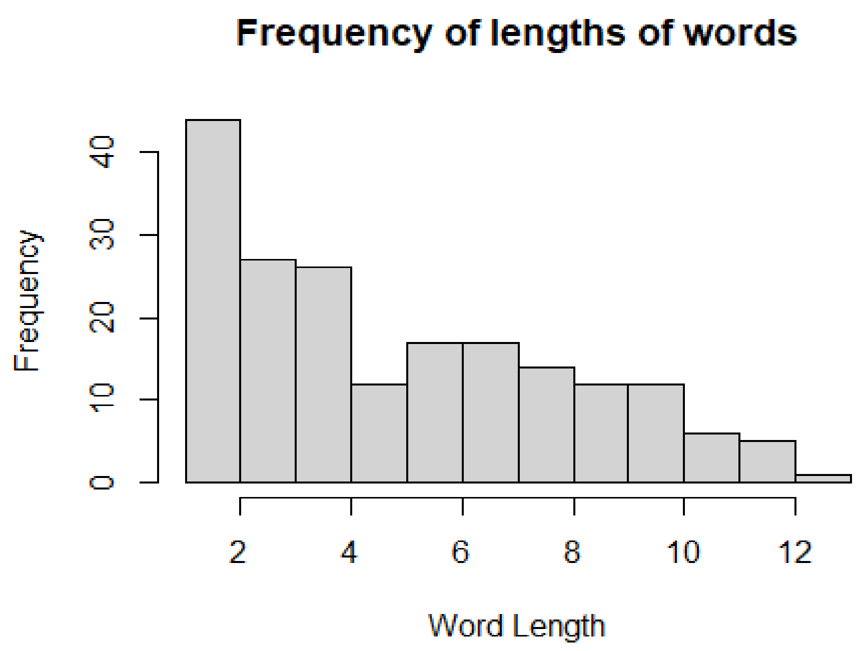
Q2. Formatting table:

```
dat <- paste(dat,collapse =' ')
dat <- gsub("[[:punct:]]", "", gsub("-"," ",dat)) #removing punctuation
dat<- data.frame(strsplit(dat, "[[:space:]]")) #splitting into words
colnames(dat)<- ("Words")
head(dat)
```

```
##         Words
## 1          We
## 2        will
## 3       begin
## 4          by
## 5 developing
## 6           a
```

Q3. Creating a letter count:

```
dat$Count <- nchar(dat$Words)
hist(dat$Count, xlab='Word Length', main='Frequency of lengths of words')
```



Q4. New column for word length:

```r
dat$State <- ifelse(dat$Count<4, '0', '1')
```

Q5. Summing over this column:

```r
paste(sum(strtoi(dat$State)), ' long words out of ', nrow(dat), '\n',
nrow(dat)- sum(strtoi(dat$State)), ' short words out of ', nrow(dat))

## "122  long words out of  193"

## "71  short words out of  193"
```

Q6. Counting pairs:

```r
dat$Pair <- mapply(dat$State, shift(dat$State, 1, 2, 'lead'), FUN =
function(a,b){paste(a,b)} ) #this adds a column shifted by one value and then
compares the two columns
n11 <- nrow(subset(dat, dat$Pair=='1 1'))
n10 <- nrow(subset(dat, dat$Pair=='1 0'))
n01 <- nrow(subset(dat, dat$Pair=='0 1'))
n00 <- nrow(subset(dat, dat$Pair=='0 0'))
```

Q7. Estimating proportions based on the values obtained in (5):

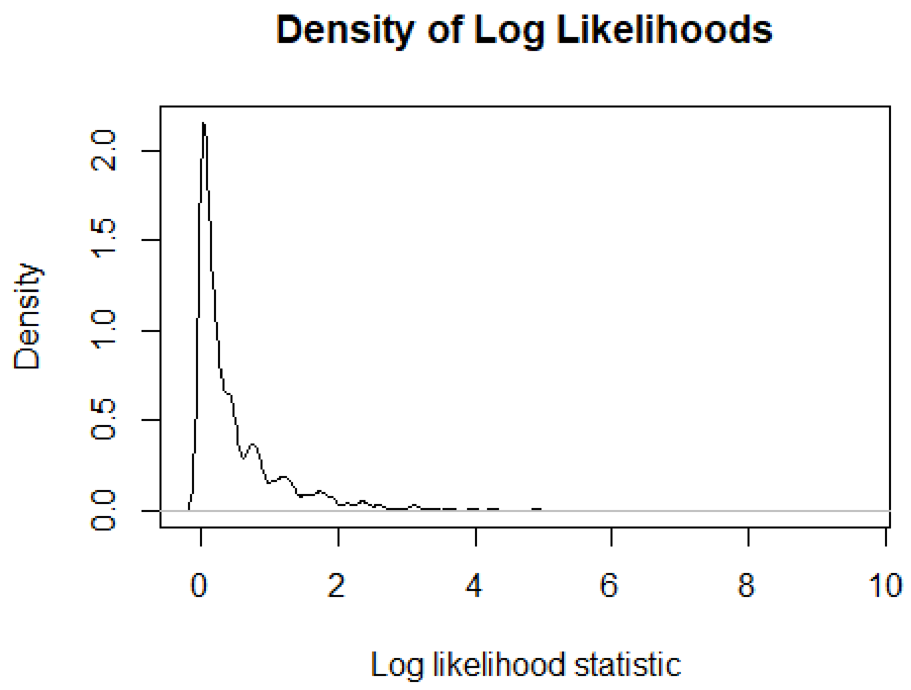
```r
paste('The expected proportion of 11 is', signif((122/193)^2,4),'\n The
expected proportion of 10 is ', signif(122*71/193^2, 4),'\n The expected
proportion of 01 is ', signif(122*71/193^2,4),'\n The expected proportion of
00 is ', signif((71/193)^2,4))

## "The expected proportion of 11 is 0.3996"

## "The expected proportion of 10 is  0.2325"

## "The expected proportion of 01 is  0.2325"

## "The expected proportion of 00 is  0.1353"
```

Q8. Log likelihood statistic: To avoid overflow errors, I calculated the likelihoods already under logarithms. Due to this we can ignore any constant term (from likelihood being directly proportional to the sum of p/q's) as it will cancel out when you subtract the q statistic from the p statistic. I've defined the log likelihood for p and q as functions as this will be useful later.

```r
loglq <- function(a,b,c,d){
  return( ((2*a)+b+c)*log(122) + ((2*d)+b+c)*log(71) - 384*log(193))
}
#likeq <- 132*log(122) + 111*log(122*71) + 30*log(71) - 384*log(193)
loglp <- function(a,b,c,d){
  return(a*log(a) + b*log(b) + c*log(c) + d*log(d) - 192*log(192))
}
likepq <- loglp(n11,n10,n01,n00) - loglq(n11,n10,n01,n00)
```

Q9. Log likelihood for random permutations: I created a function that generates random permutations and returns the log likelihood statistic. I then displayed this using a density plot.

```
randperm <- function(){
  perm <- sample(dat$State, size=193, replace=FALSE) #smaple function
randomizes permutation
  pairs <- paste(perm , shift(perm, 1, 2, 'lead')) #shifting and
concatenating
  a=sum(pairs=='1 1')
  b=sum(pairs=='1 0')
  c=sum(pairs=='0 1')
  d=sum(pairs=='0 0')
  return(loglp(a,b,c,d)-loglq(a,b,c,d)) #returning log likelihood statistic

}

s=replicate(10000,(randperm()))
plot(density(c(s)), main='Density of Log Likelihoods', xlab= 'Log likelihood
statistic') #replicate runs this function 10000 times
```

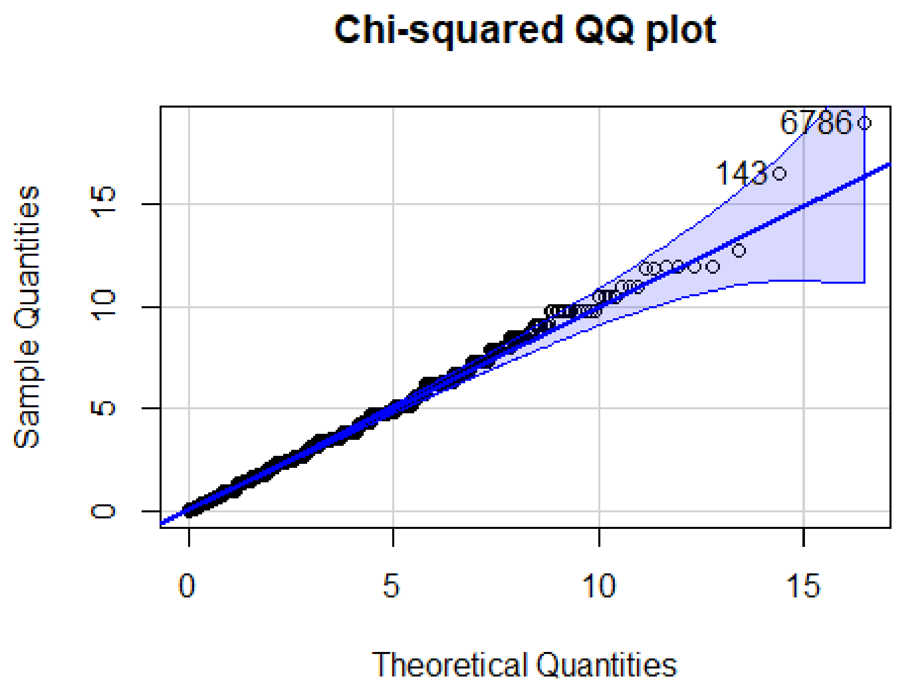


```
paste('Mean: ',round(mean(s),4),'\n Standard deviation: ',round(sd(s),4))

## "Mean:  0.5081"

## "Standard deviation:  0.7176"
```

We get graph highly concentrated around one value, with a mean of ~0.5 and sd ~ 0.7. Most of our values lie between 0 and 1 with all of them positive.

Q10. QQ-plot with twice our log likelihoods to a chi- squared distribution We have one degree of freedom as we have two free parameters estimated under the alternative hypothesis and three under the null hypothesis.

```
qqPlot(c(s)*2, distribution ='chisq', df=1, ylab='Sample Quantities',
xlab='Theoretical Quantities', main='Chi-squared QQ plot')
```



```
## [1] 6786  143

#plotting qqplot with chi-squared 1 degree of freedom
```

Q11. Comparing this to our first log likelihood:

```
paste(quantile(c(replicate(10000,(randperm())))*2, probs=(0.95))) #95th
percentile of the doubled log likelihood

## [1] "3.83162539448676"

paste(qchisq(0.95,1)) #95th percentile of the chi-squared distribution for
comparison

## [1] "3.84145882069412"
```

We need to consider half of this value ~1.9 as we doubled our log likelihood. At 5% significance, 1.9<6.0 (from Q8) and we therefore reject the null hypothesis.

Q12. Creating the matrix of transition probabilities using proportions in (6):

```
tmatrix <- matrix(c(15/71,56/71,55/121,66/121), nrow=2) #these are eaily
figured out from a diagram
print(tmatrix)

##           [,1]      [,2]
## [1,] 0.2112676 0.4545455
## [2,] 0.7887324 0.5454545
```

Q13. Function that simulates a draw of length m:

```
sim <- function(m,x_0){
  c <- list()
  if(m>1){
    x_1=ifelse(x_0==0, ifelse(sample(1:71,1)>15,1,0),
ifelse(sample(1:121,1)>55,0,1))
    #using nested ifs to calculate x_n+1 from x_n
    c=sim(m-1,x_1)} #generating the draw recursively
  return(append(c,x_0,after=0))
}

paste(sim(10,0)) #a random draw of length 10, starting at 0

##  [1] "0" "1" "1" "1" "1" "0" "1" "0" "1" "1"
```

Q14. Simulating n=100 realizations and using these to estimate the transition probabilities:

```
real <- function(m,x_0){
  return(lapply(str_count(stri_join_list(sim(m,x_0),collapse =
''),paste0("(?=",c("00","01","10","11"),")")), FUN= function(a){round(a/(m-
1),4)}))
#this function runs the simulation once and returns the estimated transition
probabilities, to 4sf
}

estimates <- data.frame(t(replicate(100,real(100,0)))) #storing the results
of 100 estimations into a data frame
head(estimates)

##       X1     X2     X3     X4
## 1 0.1212 0.3333 0.3333 0.2121
## 2 0.0707 0.3333 0.3333 0.2626
## 3 0.1111 0.3434 0.3333 0.2121
## 4 0.0808 0.3232 0.3131 0.2828
## 5 0.0505 0.3131  0.303 0.3333
## 6 0.0505 0.3636 0.3636 0.2222

X1=sum(as.numeric(estimates$X1))/100
X2=sum(as.numeric(estimates$X2))/100
X3=sum(as.numeric(estimates$X3))/100
X4=sum(as.numeric(estimates$X4))/100 #I know I can do this more efficiently
```

```
using apply but it wasn't working :(
paste(X1,',',X2,',',X3,',',X4/100) #means of the columns

## [1] "0.089587 , 0.325624 , 0.319665 , 0.00265024"

#the pairwise correlations are as follows
paste(cor(as.numeric(estimates$X1),as.numeric(estimates$X2)))

## [1] "-0.297372030199297"

paste(cor(as.numeric(estimates$X1),as.numeric(estimates$X3)))

## [1] "-0.264804953980401"

paste(cor(as.numeric(estimates$X1),as.numeric(estimates$X4)))

## [1] "-0.402628418378692"

paste(cor(as.numeric(estimates$X2),as.numeric(estimates$X3)))

## [1] "0.980480472428793"

paste(cor(as.numeric(estimates$X4),as.numeric(estimates$X2)))

## [1] "-0.749617823761829"

paste(cor(as.numeric(estimates$X3),as.numeric(estimates$X4)))

## [1] "-0.771615084472036"
```

The means for my random realizations are similar to that of the transition matrix, implying that the estimators are unbiased.

Q15. The model could be improved by considering the Markov chain to be non time-homogenous, as longer words may be used more frequently as you go further into the text. You could also analyze other pieces of text to get a more general model, and test this model's accuracy on completely new samples.