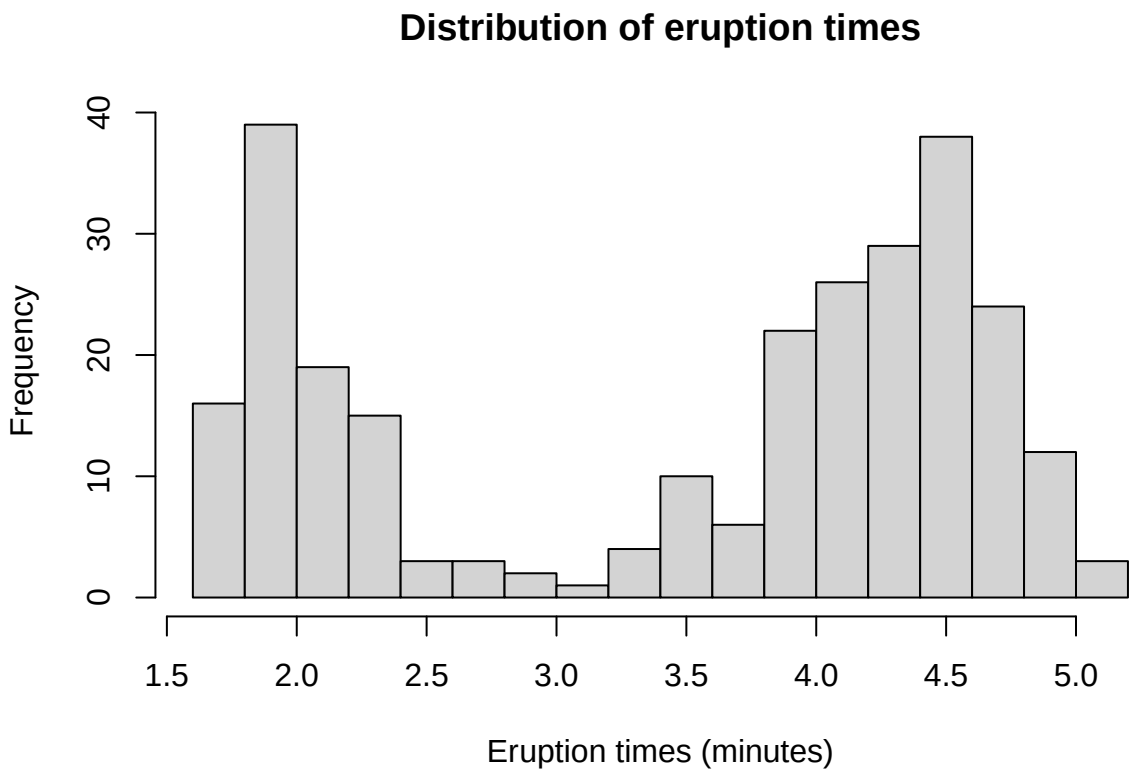
Tristan Peroy - Probability for Statistics coursework

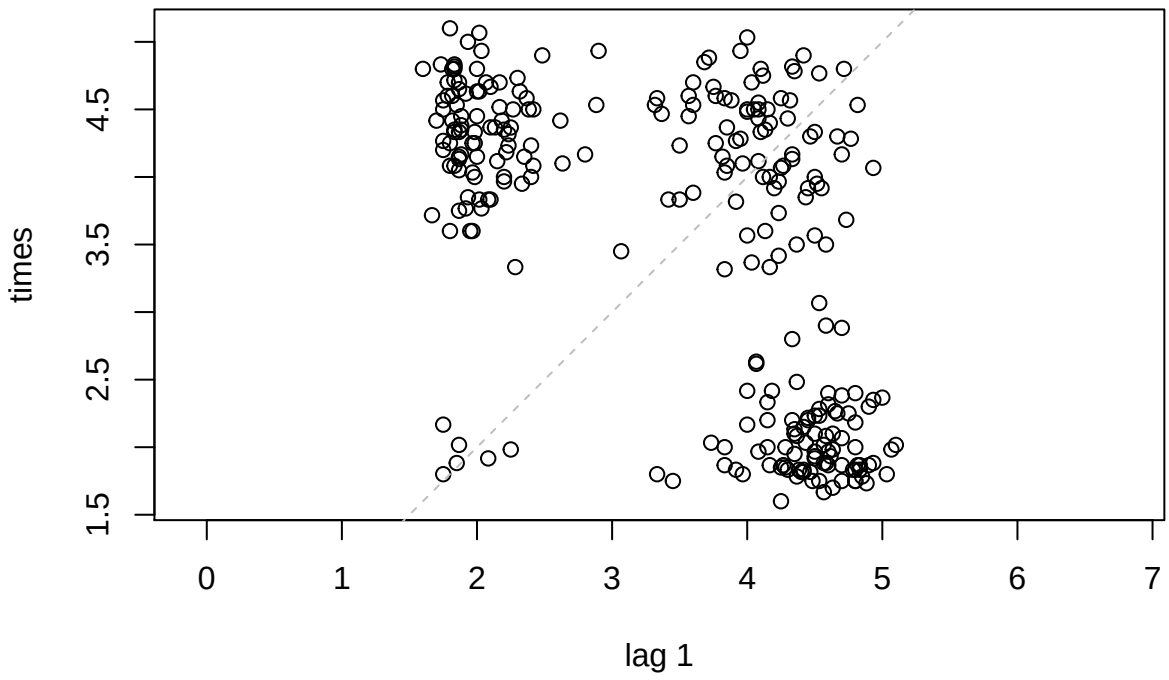
Loading and Exploration

Q1-6

```
#First, I read the data into R.
oldfaithful<-read.table("oldfaithful.txt",header=T)
#Then, I extract the eruption times column.
times<-oldfaithful$time
#I plot a histogram of the distribution with a large number of breaks for definition
histogram=hist(times,main="Distribution of eruption times",ylab="Frequency",
               xlab="Eruption times (minutes)",breaks=20)
```




```
lag.plot(times) #plot successive times against each other
```




```
state=c(times>3)+0 #convert each time to 0 if it is short(<=3) or 1 if long

string_state=paste(state,collapse="") #convert states to string
library(stringr)
count=str_count(string_state,c("0","1")) #get count of 0s and 1s in string
proportions=count/length(state) #divide by the total number of states to get proportion
short_prop=proportions[1]; long_prop=proportions[2]
short_prop;long_prop #hence get fractions of short and long eruptions
```

```
## [1] 0.3566176
```

```
## [1] 0.6433824
```

```
states=str_count(string_state,paste0("(?=",c("00","01","10","11"),")"))
states #get changes of states (0 to 0, 0 to 1, 1 to 0 and 1 to 1)
```

```
## [1]  6 91 91 83
```

Plotting a histogram of the distribution, we see that the distribution has 2 peaks with frequency peaking at eruption times around 2 and 4.5, and with a trough near 3. The histogram suggests roughly two groups, long and short. We then make a plot of successive times i.e. (t_i,t_i+1). The graph contains many points near the y=x line which correspond to when t_i and t_i+1 are similar as successive eruptions are both long or both short (short to short is the group nearer the origin and long to long further away). Meanwhile, points above or below the y=x line are for when successive eruptions are long then short or vice versa. The mean

squared error seems high overall, so correlation between successive times seems weak, hence eruption times seem not to have a relationship between each other. From the histogram, let an eruption be short less than or equal to 3 minutes and long above 3 minutes, as this separates the graph into 2 slopes.

Evaluating an independence model

Q7 There is roughly 1/3 of short eruptions and 2/3 of long eruptions in the data. Given independence of successive states, probability of (0,0) is equal to probability of 0*probability of 0, etc.Hence proportion of (0,0) is 1/3x1/3=1/9, proportion of (1,0) and of (0,1) is 1/3x2/3 i.e. 2/9 and proportion of (1,1) is 2/3x2/3=4/9. This makes sense intuitively as by independence we assume you have 2/3 chance of ending up at long in the next state and also 2/3 chance of being at long now, and multiply these probabilities by independence. P(A and B)=P(A)xP(B) i.e.P(01)=P(0)P(1). 1/9+2/9+2/9+4/9=1 as required.

Q8 I work out the log likelihood under the multinomial model. I find p_hatij=n_ij/(n-1) in each case.

```
n=length(state)
n00=6;n01=91;n10=91;n11=83;
p_hat00=n00/(n-1) #estimate p_hats from the data directly for multinomial model
p_hat01=n01/(n-1) #do this for each transition state
p_hat10=n10/(n-1)
p_hat11=n11/(n-1)
L_p_hat=p_hat00**n00*p_hat10**n10*p_hat01**n01*p_hat11**n11
#calculate the likelihood for the multinomial model
n0= 0.3566176;n1= 0.6433824; #marginal proportions of 0 and 1
q00=n0*n0;q10=n1*n0;q01=n0*n1;q11=n1*n1 #by independence
#proportions of transition states in null model
L_q_hat=(q00)**n00*(q01)**n10*(q10)**n01*(q11)**n11
#likelihood under null model
L=log(L_p_hat/L_q_hat)
#log likelihood ratio statistic L is difference of log likelihoods
L
```

```
## [1] 33.82183
```

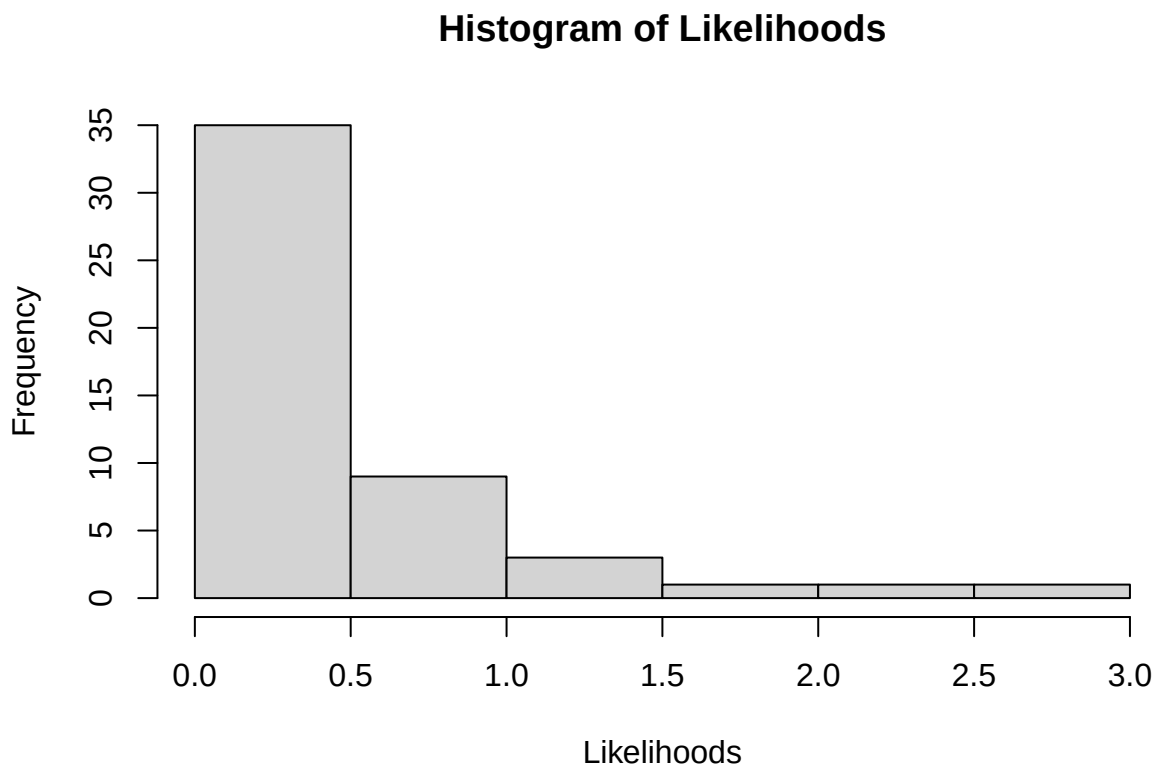
Q9

```
N=50 #number of random permutations we will make
Likelihoods=c()
for (i in 1:N){
  n=length(state)
  sample=sample(state,n,replace=F)#randomly permute the null distribution
  string_state=paste(sample,collapse="") #work out likelihoods ratio for each i
  library(stringr)
  count=str_count(string_state,c("0","1"))
  proportions=count/n
  n0=proportions[1]; n1=proportions[2]
  states=str_count(string_state,paste0("(?=",c("00","01","10","11"),")"))
  n00=states[1];n01=states[2];n10=states[3];n11=states[4]
  p_hat00=n00/(n-1);p_hat01=n01/(n-1);p_hat10=n10/(n-1);p_hat11=n11/(n-1)
  L_p_hat=p_hat00**n00*p_hat10**n10*p_hat01**n01*p_hat11**n11
  q00=n0*n0;q10=n1*n0;q01=n0*n1;q11=n1*n1
  L_q_hat=(q00)**n00*(q01)**n10*(q10)**n01*(q11)**n11
  L=log(L_p_hat/L_q_hat)#calculate each log likelihood as before
```

```
  Likelihoods=append(Likelihoods,L) #collect the simulated likelihoods in a list
}
Likelihoods
```

```
##  [1] 0.018638978 0.004790202 1.447138754 0.247536177 0.018638978 0.968443788
##  [7] 0.018638978 0.008946781 0.069602695 1.110576523 0.466135764 0.004790202
## [13] 0.008946781 0.183004070 0.578187288 0.637826310 0.018638978 0.059187447
## [19] 0.201074729 0.968443788 0.105695155 0.247536177 0.578187288 0.008946781
## [25] 0.247536177 0.008946781 0.008946781 0.059187447 0.069602695 1.942443633
## [31] 1.110576523 0.375957931 0.201074729 0.183004070 0.788488455 2.124653499
## [37] 0.968443788 0.320764045 0.059187447 0.059187447 0.041337426 0.149356633
## [43] 0.201074729 0.564934392 0.201074729 0.201074729 2.719209035 0.637826310
## [49] 0.059187447 0.020156427
```

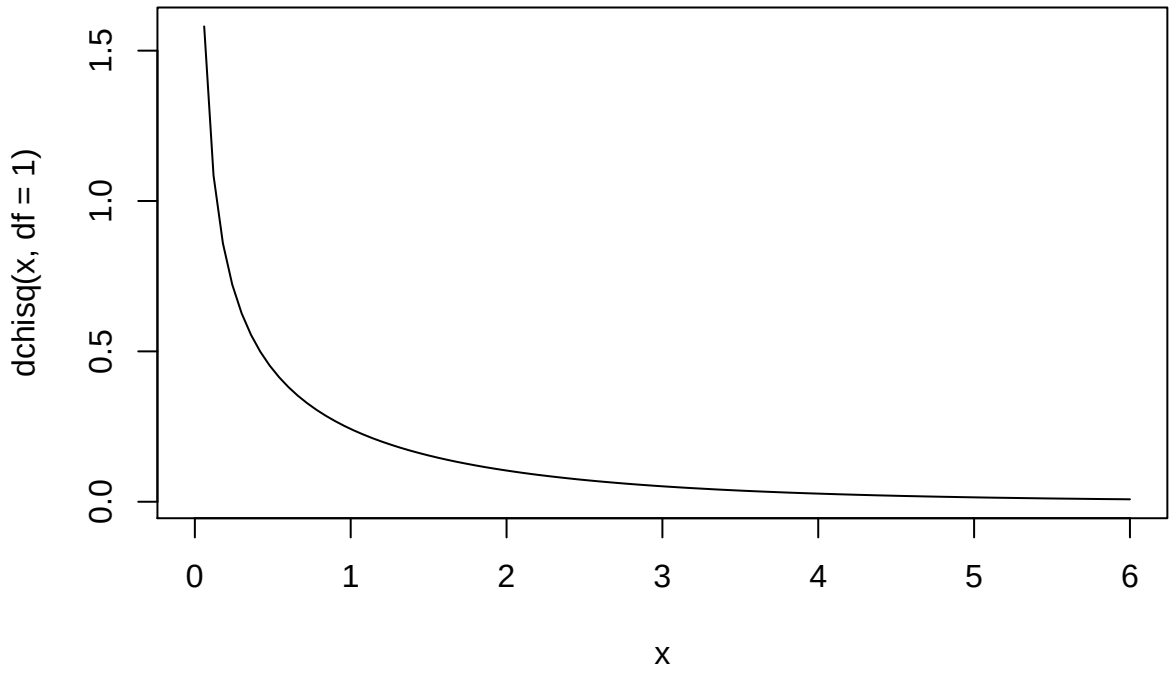
```
hist(Likelihoods)#make a histogram to see distribution in likelihoods
```



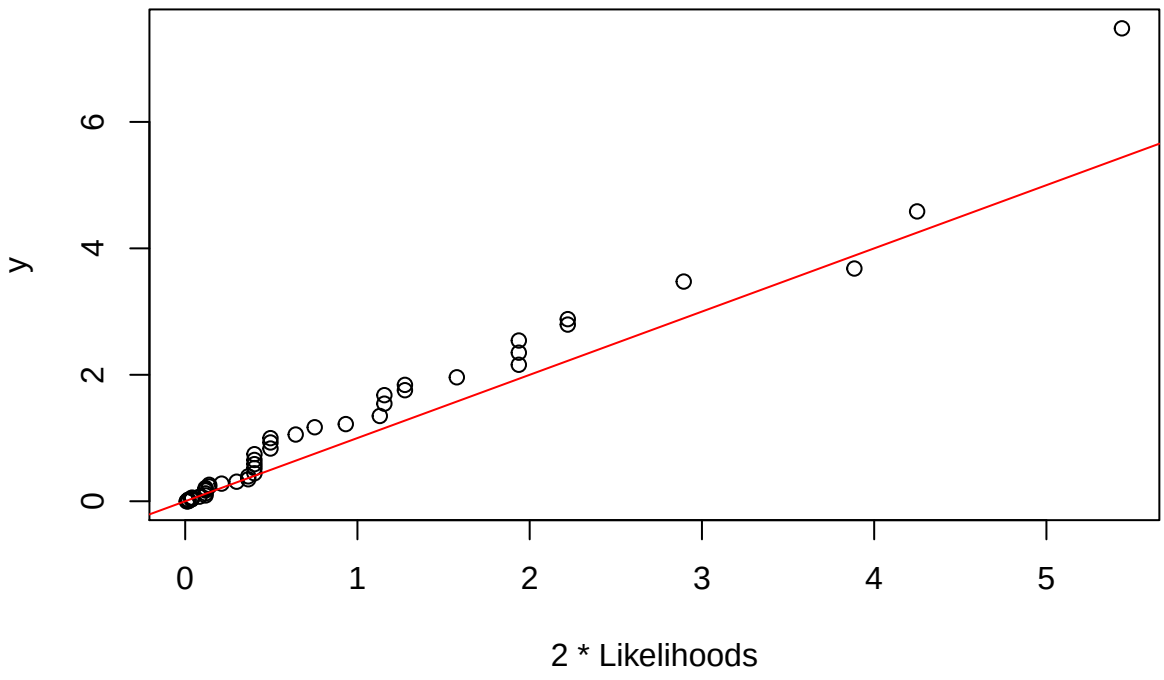
Q10 The test statistic for the data is about 34 which is much larger than the results obtained in the null distribution which are mostly between 0 and 2 and with low frequency going up to 6. The likelihood ratio statistic is often greater than one: data simulated under the null model will actually have higher probability under the alternative model. However, the likelihood ratio test statistic is much greater than we would expect for data simulated under the null model, so the difference between the null and alternative models is much smaller than the difference between the data and the alternative model (5 times the greatest in the simulation hence very unlikely), so we have evidence against the null hypothesis. We can hence reject the null hypothesis that the data follows the null distribution (independence model). Since we have simulated many data sets, the fact that the alternative model has more free parameters is not important.

Q11 First in order to compare our distribution to a chi-squared we must estimate the degree r of chi-squared where r is the difference between the number of free parameters estimated under the alternative hypothesis and the number of free parameters estimated under the null hypothesis. The free parameters under the alternative hypothesis are p00 and p10, and p01=1-p00, p11=1-p10 so only the first 2 are free parameters of the log likelihood (knowing these proportions is equivalent to knowing n and the nijs) The free parameters estimated under the null hypothesis is just p0, as p1=1-p0 so only p0 is free. So r=2-1=1. We can see that for r=1, the distribution of the chi-squared is roughly twice the distribution of likelihood ratio statistic, and we can see they are correlated.

```
curve(dchisq(x,df=1),from = 0, to = 6) #plot a chi-squared distribution
```




```
y=rchisq(n,df=1) #produce sample of n chi-squared distributed values
qqplot(2*Likelihoods,y) #compare quartiles of chi-squared against the likelihoods
abline(0, 1, col="red")
```



A Two State Markov Model

Q12 The proportion of 0s that convert to 0 and that convert to 1 sums to 1. Hence we take the ratio of each in the data. In the transition matrix, p00+p01=1 and p10+p01=1. p00=n00/(n01+n00)=6/(6+91)=0.06. p01=1-0.06=0.94. p10=n10/(n10+n11)=91/(91+83)=0.52, 011=1-0.52=0.48.

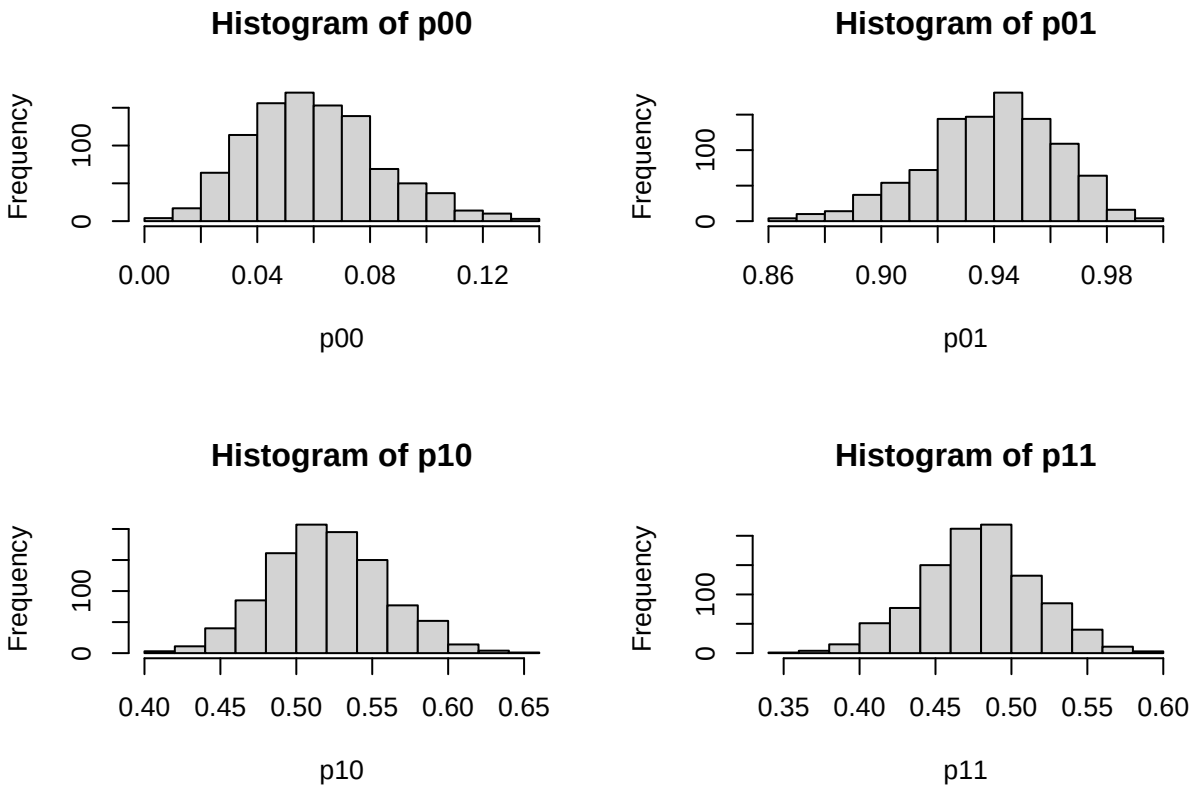
```
P = matrix(c(0.06,0.94,0.52,0.48),
           nrow=2,
           byrow=TRUE )
```

Q13

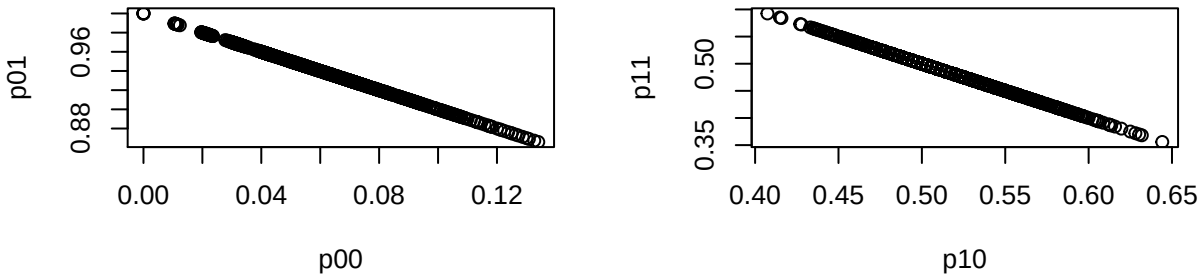
```
initial=c(1/3,2/3) #set the initial distribution
M<-272 # length of chain to be simulated
markovchain <- function(initial, P, M) {#function simulating draw of length M
  x<-vector(length=M)
  x[1]<-sample(x=2,size=1,prob=initial)#take first sample element
  for(i in 2:M){ #iteratively compute other elements
    x[i]<-sample(x=2,
                 size=1,
                 prob=P[x[i-1],],replace=T)
  }
  return (x-1) #return the draw
}                        #given initial distribution and transition probabilities
markovchain(initial,P,M) #create a chain of 0s and 1s
```

```
##   [1] 1 0 1 1 0 1 1 0 1 0 1 0 1 0 1 1 1 1 0 1 0 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1 0
##  [38] 1 1 1 1 0 1 1 0 1 1 1 0 1 0 1 0 1 1 0 1 1 0 1 0 1 1 1 1 0 1 1 1 0 1 1 0 1
##  [75] 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1
## [112] 1 0 1 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 0
## [149] 1 1 1 0 0 1 1 0 0 1 0 1 0 1 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 1 1 1 1 0 1
## [186] 1 1 1 1 1 1 1 0 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 1 1 0 1 0 1 0
## [223] 1 0 1 0 1 0 1 1 1 0 1 1 0 1 0 1 1 0 1 1 0 1 0 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1
## [260] 1 1 0 1 0 1 0 1 0 1 1 0 1
```

```
p00=c();p01=c();p10=c();p11=c()
for (i in 1:1000){ #take 20 samples of markov chains
  chain=markovchain(initial,P,M)
  library(stringr)
  string_state=paste(chain,collapse="")
  states=str_count(string_state,paste0("(?=",c("00","01","10","11"),")"))
  n00=states[1];n01=states[2];n10=states[3];n11=states[4]
  prob00=n00/(n00+n01);prob01=1-prob00;prob11=n11/(n11+n10);prob10=1-prob11
  #work out the transition probabilities for each sample
  p00=append(p00,prob00);p01=append(p01,prob01)#collect into vectors
  p10=append(p10,prob10);p11=append(p11,prob11)
}
par(mfrow=c(2,2))
hist(p00);hist(p01);hist(p10);hist(p11)
```



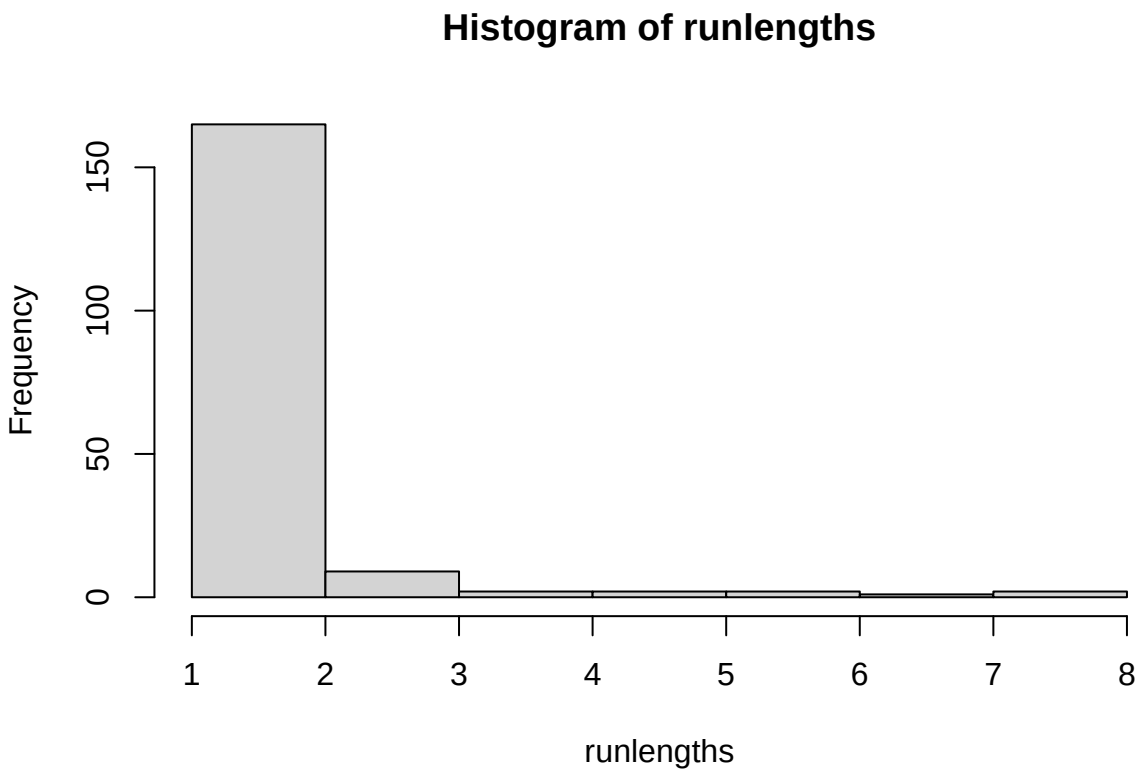
```
plot(p00,p01) #correlation plot between p00 and p01 etc.
plot(p10,p11)
```



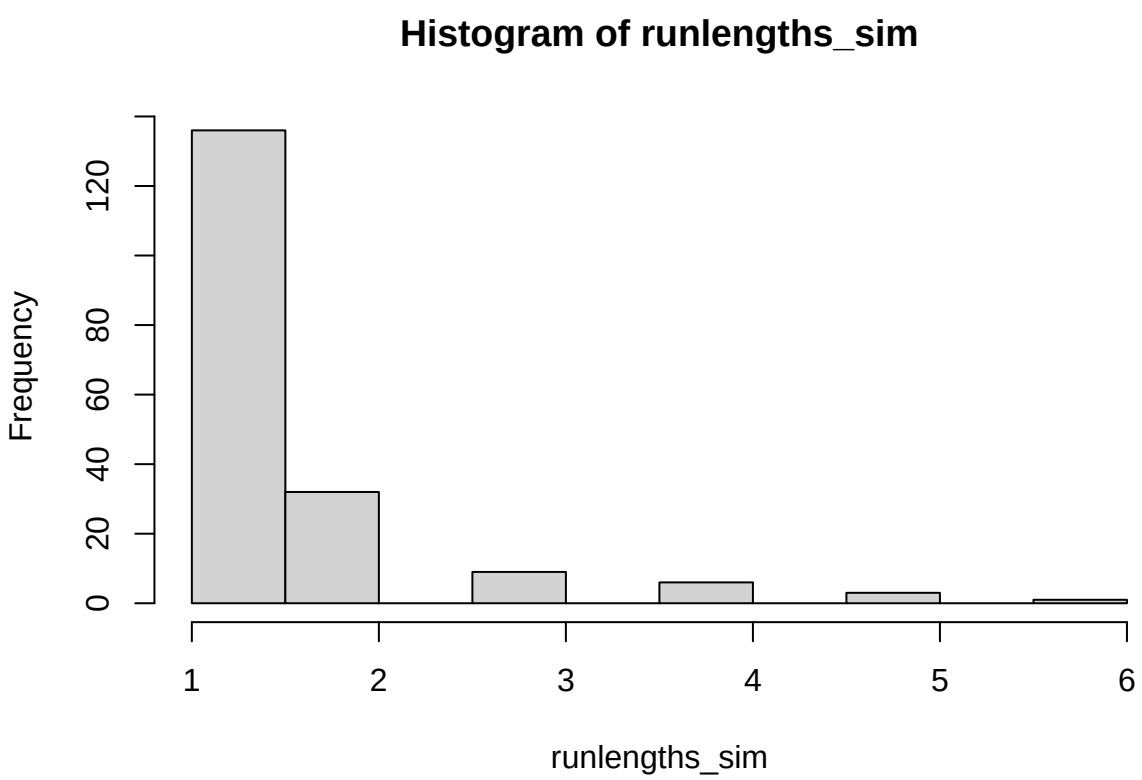
p00 and p01 are negatively correlated as are p10 and p10 (as expected as they should sum to 1). No correlation between the others eg p00 and p10.

Q14

```
runlengths=rle(unlist(state))$lengths #use run length encoding function to get run lengths
hist(runlengths)#show distribution of run lengths in the data
```




```
runlengths_sim=rle(unlist(markovchain(initial,P,M)))$lengths
hist(runlengths_sim)#show distribution of run lengths in simulation
```



The distribution of run lengths for the model and the distribution of run lengths for the data are similar, hence the model seems reasonable. We could also compare the model with the true data in other ways. For example, we could count the number of times the data changes state in both the model and the data. We could also calculate the proportion of 0s and 1s per quartile e.g. how many 1s are in the first quarter in both the simulation and the data. This would check for homogeneity.

Q15

As seen in the data, eruption times can roughly be divided into short of around 2 minutes and long of around 4.5 minutes which is 2 times more likely. We find the eruption time of an eruption isn't often similar to the next eruption time. When we assume that eruption times are independent of each other i.e. a short eruption doesn't influence the chance of the next one being short or long, we find that this model doesn't fit the data at all, so we can't make this assumption of independence.The independence model is bad. Meanwhile, a two-state time-homogeneous Markov model seems to reasonably model the data. In this model, successive states depend on each other based on probabilities. Given a short eruption, we have a 0.94 chance of a long eruption next, and given a long eruption, we have a 0.48 chance of a long eruption next. This can be used to try to predict future eruption distributions given past conditions. Time-homogeneous means these probabilities don't change over time.