Assignment 6

Giang Vu

4/6/2021

Homework 6

i.

The function poisLoglik is defined as follows. I tested wih $data = c(1, 0, 0, 1, 1)$ and $\lambda = 1$ and got -5, similar to when I did the calculation by hand.

```
#define poisLoglik
poisLoglik <- function(lambda, data){
  stopifnot(
    length(lambda)==1
  )
  res <- c(numeric(length(data)))
  for (i in 1:length(data)){
    res[i] <- log((lambda^data[i])*exp(-lambda)/factorial(data[i]))
  }
  return(
    sum(res)
  )
}

#test
poisLoglik(1, c(1,0,0,1,1))
```

```
## [1] -5
```

ii.

The function count_new_genres is defined below. Testing it with values 1803 and 1850, I got the results 0 and 3, which agrees with the data set.

```
#read data
moretti <- read.csv("moretti.csv",header = T,sep = ",")

#define count_new_genres
count_new_genres <- function(year){
```

```
  res <- c(numeric(length(year)))
  for (i in 1:length(year)){
    res[i] <- nrow(moretti[moretti$Begin==year[i],])
  }
  return(res)
}

#test
count_new_genres(1803)
```

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

```
count_new_genres(1850)
```

```
## [1] 3
```

iii.

Vector new_genres is created by applying the function in part ii on a vector of consecutive numbers starting from 1740 to 1900.
The positions of the result for a specific year is the relative position of that year numeric value to the starting year 1740.
For example, the position of value for 1803 would be the $1803 - 1740 + 1 = 64$th number in the result vector.
Similarly the value for year 1850 would be the 111st number in the vector.
I checked and those values are the same with what we got in part ii, which is 0 new genre for year 1803 and 3 new genres for year 1850.

```
#create vector
new_genres <- count_new_genres(1740:1900)

#test
new_genres[64]
```

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

```
new_genres[111]
```

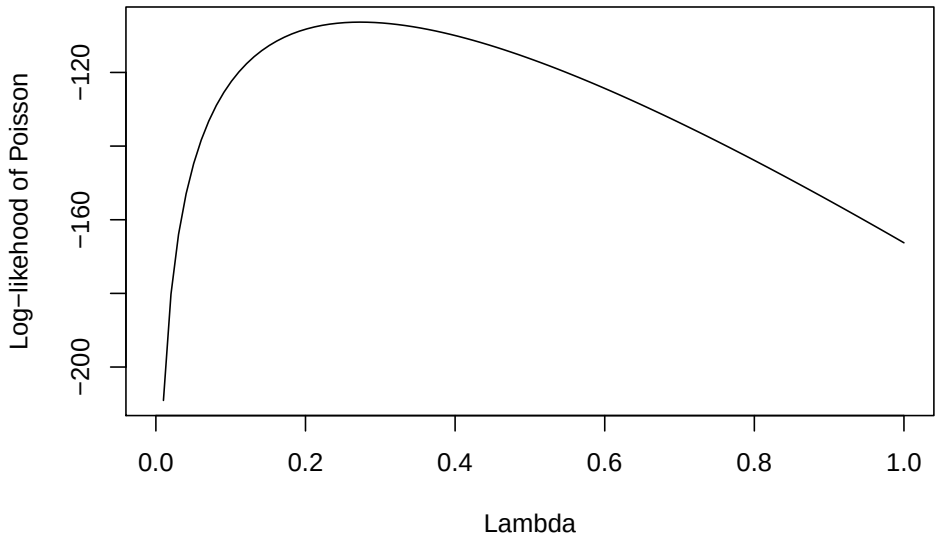
```
## [1] 3
```

iv.

Below is the plot of posLoglik as a function of λ on the new_genres data.
By looking at the plot and actually checking, the maximum point of this function is achieved when $\lambda = 0.273$.

```r
#create df with 2 cols, one is different lambda values, the other is log likelihood result
lambda <- seq(0,1,length=100)
df <- data.frame(lambda=lambda)
df$ll <- 0
for (i in 1:nrow(df)) {
  df$ll[i] <- poisLoglik(df$lambda[i],data = new_genres)
}

#plot
plot(x=df$lambda,y=df$ll,type = "l",
     ylab = "Log-likehood of Poisson", xlab = "Lambda")
```




```r
#check maximum point
df$lambda[which.max(df$ll)]
```

```
## [1] 0.2727273
```

v.

I defined a new function, which is the negative log likelihood function for the Poisson distribution.
Minimizing this negative log likelihood function with nlm() is equivalent to maximizing the original log likelihood function in part (i.).
The estimated result agrees with what we found earlier, the value $\lambda = 0.273$ is the value that maximizes our log likelihood function.

```r
#define negative log likelihood function
negpoisll <- function(lambda, data){
```

```
  stopifnot(
    length(lambda)==1
  )
  res <- c(numeric(length(data)))
  for (i in 1:length(data)){
    res[i] <- log((lambda^data[i])*exp(-lambda)/factorial(data[i]))
  }
  return(
    -sum(res)
  )
}

#minimize negative log likelihood
#equivalent to maximize log likelihood
nlm(negpoisll,0.1,data=new_genres)
```

```
## $minimum
## [1] 106.3349
##
## $estimate
## [1] 0.2732914
##
## $gradient
## [1] 3.893774e-06
##
## $code
## [1] 1
##
## $iterations
## [1] 7
```

vi.

Using diff(), I was able to create the vector intergenre_intervals.
The time intervals between genre appearances have mean 3.442, standard deviation 3.71 and coefficient of variation 1.077.

```
#use diff with lag 1
intergenre_intervals <- diff(moretti$Begin)
intergenre_intervals
```

```
##  [1]  8 11  7  2  2  3 16  1  1  9  4  4  6  8  3  1  2  2  0  2  6  1  7  0  1
## [26]  1  1  1  0  0  1  6 11  3  1  0  1  3  8  1  0  3  0
```

```
#mean
mean(intergenre_intervals)
```

```
## [1] 3.44186
```

```r
#standard deviation
sd(intergenre_intervals)
```

```
## [1] 3.705224
```

```r
#coefficient of variation
moretti.coef <- sd(intergenre_intervals)/mean(intergenre_intervals)
moretti.coef
```

```
## [1] 1.076518
```

vii.

a.

A function intergenre_fcn is designed so that when it takes in a vector like new_genres, it will assign year index starting from 1 to the number of years in its input vector.
The function then takes out only year with 1 or more new genres created, and creates a new vector that is the same form as the original "Begin" column in the dataset, where each year index is repeated by the number of new genres created that year.
The function finally returns a veetor of time intervals by applying diff() on the vector from previous step.
I tested with input new_genres and I got the result that is exactly the same as intergenre_intervals.

```r
#define function
intergenre_fcn <- function(ng){
  names(ng) <- 1:length(ng) #mark year index
  ng1 <- ng[ng!=0] #filter only year with new genre
  ng2 <- rep(names(ng1),ng1) #create a vector like original 'begin' column of data, year index is repea
  return(diff(sort(as.numeric(ng2))))
}

#test
intergenre_fcn(new_genres)
```

```
##  [1]  8 11  7  2  2  3 16  1  1  9  4  4  6  8  3  1  2  2  0  2  6  1  7  0  1
## [26]  1  1  1  0  0  1  6 11  3  1  0  1  3  8  1  0  3  0
```

b.

A function is defined below that draws a sample from Poisson, apply the function intergenre_fcn from part a. to that sample to get the interval vector, and then make a list of the vector and the coefficient of variation.
Under seed 6, I tested with 161 years and mean of 0.273, and the mean of the intervals is 3.12, which is between 3 and 4.

```r
#define function
pois.sim <- function(no.year, mean.no.genres){
  pois <- rpois(n=no.year, lambda = mean.no.genres) #draw from poisson
  pois_interval <- intergenre_fcn(pois) #apply function from previous part for the draw
```

```
  res <- list(`inter_appearance_intervals`=pois_interval,
              `coeff_of_variation`=sd(pois_interval)/mean(pois_interval)) #make result list
  return(res)
}

#test with 161 years and mean 0.273
set.seed(6)
test <- pois.sim(161,0.273)
test
```

```
## $inter_appearance_intervals
##  [1]  3  1  0  1  1  4  3  6  1  6  5  3  3  0  1  4  1  0  1  4  0  4  1  0  1
## [26]  1  5  6  0  1  6  3  6 11  7  5  1  3  5 14  8  0 12  4  1  1  1  1  1  2
## [51]  0
##
## $coeff_of_variation
## [1] 1.027711
```

```
mean(test$inter_appearance_intervals)
```

```
## [1] 3.117647
```

viii.

The simulation is run 10,000 times and the coefficient of variation from each time is collected in a vector.

```
#run simulation under seed 6
set.seed(6)
sim.coef.var <- numeric(10000)
for (i in 1:10000) {
  sim <- pois.sim(161,0.273)
  sim.coef.var[i] <- sim$coeff_of_variation
}
head(sim.coef.var,20)
```

```
##  [1] 1.0277114 1.1313917 0.8328451 1.0846123 0.9115654 0.9166891 0.8846134
##  [8] 1.1638127 1.0866804 0.9052300 0.8676576 1.0269144 1.1334656 0.8619138
## [15] 0.8854351 1.4038169 0.8609797 0.7398047 1.1306102 1.1251608
```

About 23% of the simulation runs have higher coefficient of variation than Moretti's data.

```
#fraction of coef.var that's higher than original data's
sum(sim.coef.var > moretti.coef)/10000
```

```
## [1] 0.2303
```

ix.

We look at the coefficient of variation as a measurement for how much genres tend to appear together in burst.
The result from the previous simulations tells us that if new genres' appearance follow a Poisson distribution and don't appear in bursts, then about 23% of the time we will get results as extreme as Moretti's data.
So it is not uncommon to get such results assuming that the appearance doesn't happen in bursts.
Therefore we fail reject the null hypothesis that the genres' appearance are truly randomly distributed with Poisson distribution and are not in bursts.
In other words, we conclude that the genres don't appear in bursts.