HW2: MLE and Method of moments

Arjun Bhan

3/6/2021

Exercise 1:

```
library("maxLik")
```

```
## Loading required package: miscTools
```

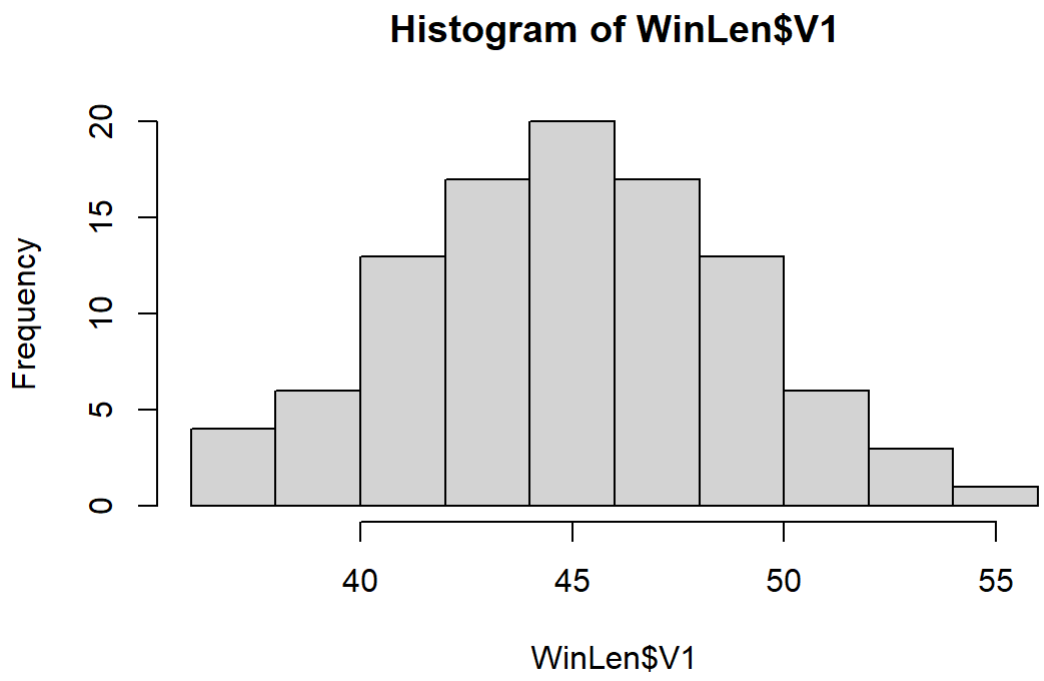
```
##
## Please cite the 'maxLik' package as:
## Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood estimation
in R. Computational Statistics 26(3), 443-458. DOI 10.1007/s00180-010-0217-1.
##
## If you have questions, suggestions, or comments regarding the 'maxLik' package, please use a
forum or 'tracker' at maxLik's R-Forge site:
## https://r-forge.r-project.org/projects/maxlik/
```

```
#a
WinLen<-read.table("https://foxweb.marist.edu/users/duy.nguyen2/s057.txt", header=F)
#b
hist(WinLen$V1)
qnorm(WinLen$V1)
```

```
## Warning in qnorm(WinLen$V1): NaNs produced
```

```
##   [1] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
##  [19] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
##  [37] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
##  [55] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
##  [73] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
##  [91] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
```

```
qqline(WinLen$V1)
```




```
#By analyzing the graph the data seem to be fairly normally distributed.
#c
logLikeFun=function(param){
  mu=param[1]
  sigma=param[2]
  sum(dnorm(WinLen$V1, mean = mu, sd=sigma,log=TRUE))
}
mle=maxLik(logLik = logLikeFun, start=c(mu=0, sigma=1))
```

```
## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced
```

```
## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced
```

```
## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced

## Warning in dnorm(WinLen$V1, mean = mu, sd = sigma, log = TRUE): NaNs produced
```

```
summary(mle)
```

```
## --------------------------------------------
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 16 iterations
## Return code 8: successive function values within relative tolerance limit (reltol)
## Log-Likelihood: -277.9915
## 2  free parameters
## Estimates:
##       Estimate Std. error t value Pr(> t)
## mu     45.5000     0.3911  116.33  <2e-16 ***
## sigma   3.9000     0.2754   14.16  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## --------------------------------------------
```

```
# mean=45.5000 and standard deviation=3.9000
#D
1-pnorm(50,mean=45.5000,sd=3.9000)
```

```
## [1] 0.1242816
```

```
#The probability P(Wing length>50) is 0.1242816
```

Exercise 2:

```
#A
# The expected value is p
#B
#We have only one unknown which is the P value. In order to find the unknown we set the theoreti
cal mean to be equal to the empirical mean. The theoretical mean would be P and the empirical me
an would be the average of all the values.
#xbar =(1/n)(x1,x2....+xn)
#P=xbar
#C
set.seed(1)
N<-10000
x=rbinom(N, 1, p=0.5)
p=mean(x)
p
```

```
## [1] 0.4953
```

Exercise 3:

```
#a
#The expected value is 1/lambda

#b
#1/lambda = (1/n)(x1,x2....+xn)
#1/lambda=xbar
#lambda=1/xbar


#c
set.seed(8)
n=10000
x=rexp(n, rate=4)
lambda=1/mean(x)
lambda
```

```
## [1] 3.963788
```

Exercise 4:

```
#a
#The expected value is 1/p
#b
# 1/p=(X1+X2+...+Xn)/n
# 1/p=xbar
#p=1/xbar
#c
set.seed(123)
n=10000
x=rgeom(n, .5)
p=1/mean(x)
p
```

```
## [1] 1.012556
```

Exercise 5:

```
#a
Clust <-c(1,2,3,4,5,6,7,8)
Freq<-c(132,52,34,9,7,5,5,6)
xbar<-sum(Clust*Freq)/sum(Freq)
xbar
```

```
## [1] 2.088
```

```
# The answer is 2.088

#B

#1/p=xbar
p=1/xbar
p
```

```
## [1] 0.4789272
```

```
#0.4789272

#C

#250*f(i/p)
#=250(1-p)^(i-1) * p :1,2,...,8
250*(1-0.4789272)^(1-1)*(0.4789272)
```

```
## [1] 119.7318
```

```
#119.7318
250*(1-0.4789272)^(2-1)*(0.4789272)
```

```
## [1] 62.38898
```

```
#62.38898
250*(1-0.4789272)^(3-1)*(0.4789272)
```

```
## [1] 32.5092
```

```
#32.5092
250*(1-0.4789272)^(4-1)*(0.4789272)
```

```
## [1] 16.93966
```

```
#16.93966
250*(1-0.4789272)^(5-1)*(0.4789272)
```

```
## [1] 8.826797
```

```
#8.826797
250*(1-0.4789272)^(6-1)*(0.4789272)
```

```
## [1] 4.599404
```

```
#4.599404
250*(1-0.4789272)^(7-1)*(0.4789272)
```

```
## [1] 2.396624
```

```
#2.396624
250*(1-0.4789272)^(8-1)*(0.4789272)
```

```
## [1] 1.248816
```

```
#1.248816

#D
```

```
# The geometric model sort of describes the distribution. The values are close but, not exact.
```