CGS698C, Assignment 02 solutions

Himanshu Yadav

2024-01-25

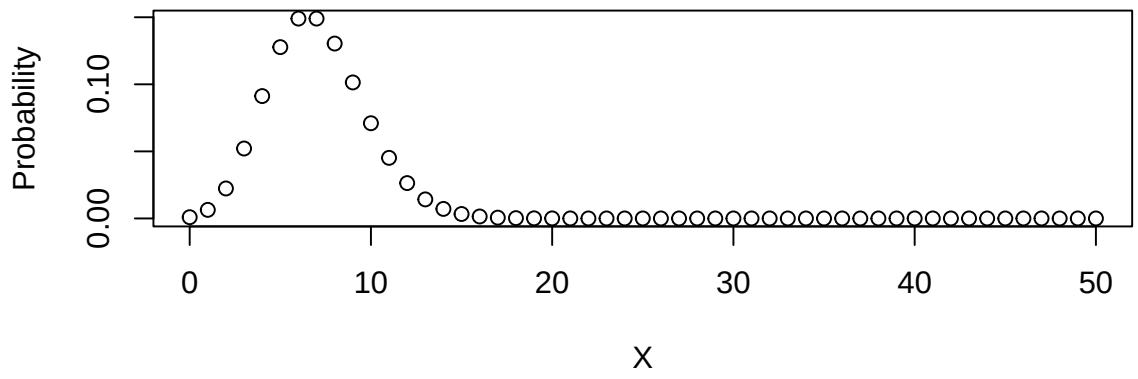
Part 1: Discrete random variables

1.1(a) $P(X = 10) = f(10) = \frac{7^{10}e^{-7}}{10!} = 0.0709$

(b) $P(7 < X < 10) = P(X = 8) + P(x = 9) = \frac{7^{8}e^{-7}}{8!} + \frac{7^{9}e^{-7}}{9!} = 0.2317$

(c)

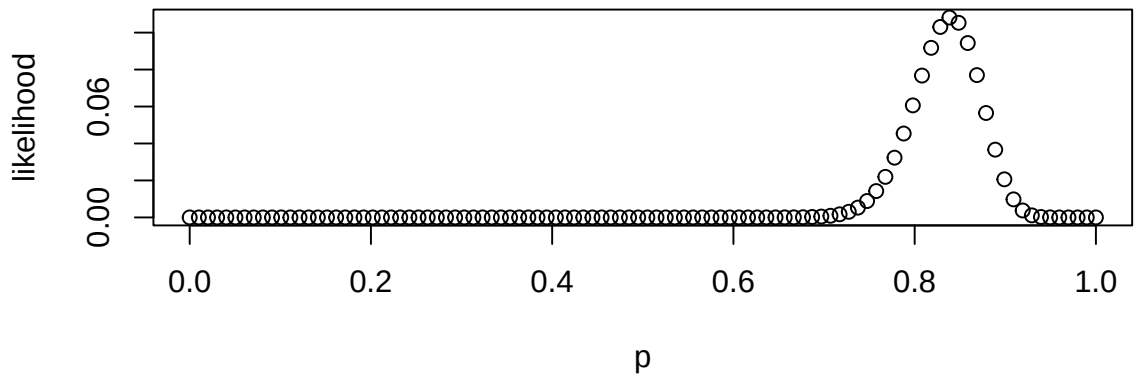
```
X <- 0:50
Probability <- rep(NA,51)
for(i in 1:51){
  Probability[i] <- 7^(X[i])*exp(-7)/factorial(X[i])
}
plot(X,Probability)
```



1.2

```
# (a)
p <- seq(from=0,to=1,length=100)

# (b)
likelihood <- dbinom(x=84,size=100,prob=p)
plot(p,likelihood)
```




```r
# (c)
p[which(likelihood==max(likelihood))]
```

```
## [1] 0.8383838
```

Part 2: Continuous random variables

2.1

```r
normal_density <- function(x,mu,sigma){
  density <- (1/(sigma*sqrt(2*pi)))*exp((-1*((x-mu)^2))/(2*(sigma^2)))
  return(density)
}

# (a)
normal_density(x=0,mu=1,sigma=1)
```

```
## [1] 0.2419707
```

```r
# (b)
normal_density(x=1,mu=0,sigma=1)
```

```
## [1] 0.2419707
```

(c)
Given: $P(x_1 < X < x_2) = 0.3$
$P(x_1 < X < x_3) = 0.45$
If $x_1 < x_2 < x_3$,
From the third axiom, we can write
$P(x_1 < X < x_3) = P(x_1 < X < x_2) + P(x_2 < X < x_3)$

$P(x_2 < X < x_3) = P(x_1 < X < x_3) - P(x_2 < X < x_3) = 0.45 - 0.30 = 0.15$
However, if $x_2 < x_1 < x_3$,
From the third axiom, we can write
$P(x_2 < X < x_3) = P(x_2 < X < x_1) + P(x_1 < X < x_3)$
$P(x_2 < X < x_3) = 0.45 + 0.30 = 0.75$

2.2

```
dnorm(x=550,mean=650,sd=125)
```

```
## [1] 0.002317532
```

2.3

```
pnorm(700,mean=500,sd=100)
```

```
## [1] 0.9772499
```

```
1-pnorm(900,mean=500,sd=100)
```

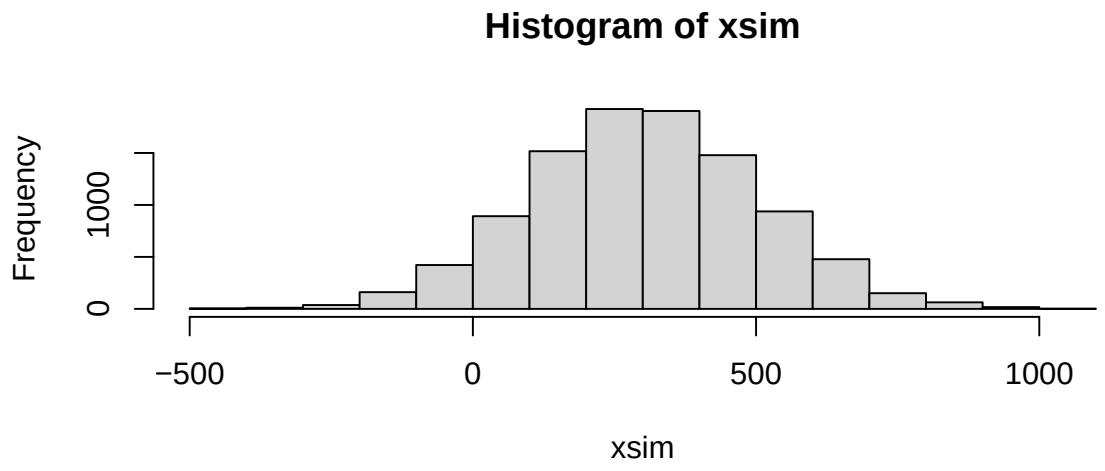
```
## [1] 3.167124e-05
```

```
pnorm(800,mean=500,sd=100)-pnorm(200,mean=500,sd=100)
```

```
## [1] 0.9973002
```

2.4

```
xsim <- rnorm(n=10000, 300, 200)
hist(xsim)
```



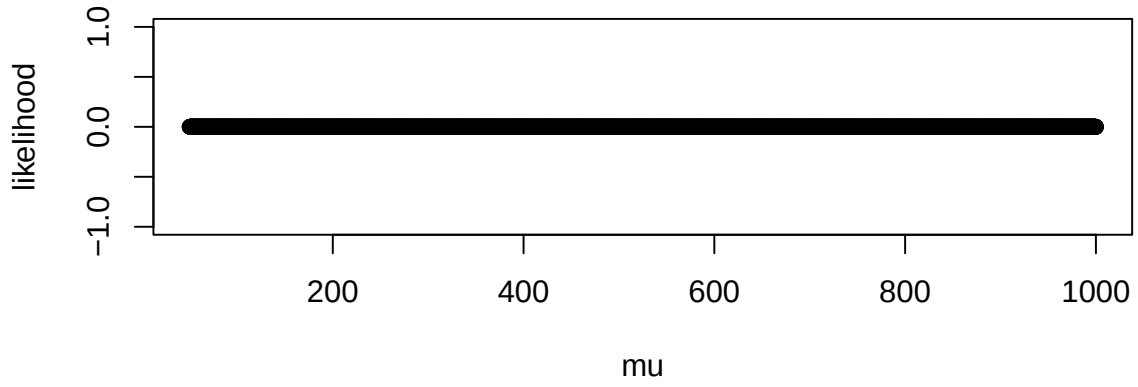

```
c(mean=mean(xsim),sd=sd(xsim),minimum=min(xsim),maximum=max(xsim))
```

```
##      mean        sd   minimum   maximum
##  303.6912  200.2866 -483.4975 1053.0781
```

Part 3

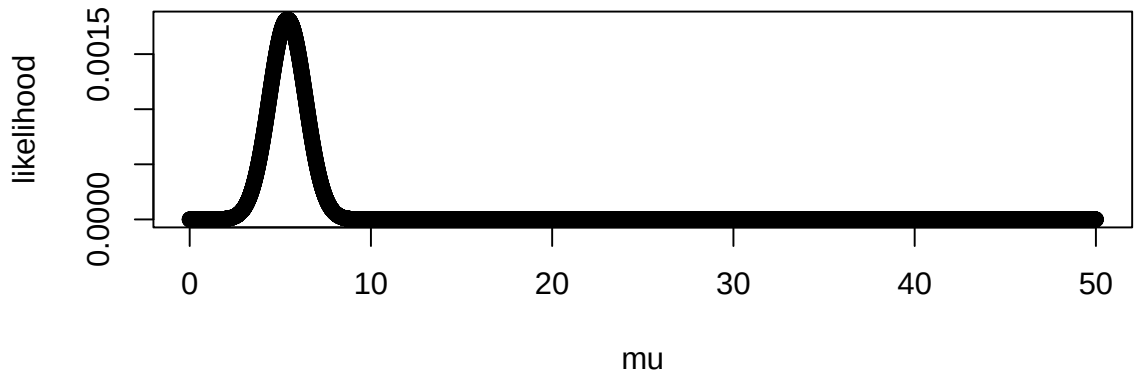
```
likelihood_function <- function(x,mu){
  likelihood <- (1/(x*sqrt(2*pi)))*exp((-1/2)*((log(x)-mu)^2))
  return(likelihood)
}

mu <- seq(from=50,to=1000,length=10000)
likelihood <- likelihood_function(x=220,mu=mu)
plot(mu,likelihood)
```



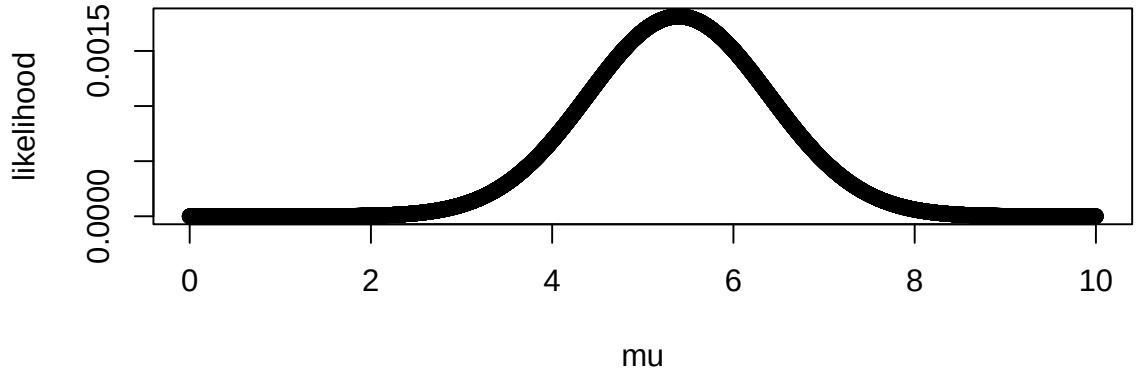
All the values of μ between 50 and 1000 have close to zero likelihood. It means we chose a wrong range here. Notice, the probability density function is not of a Normal distribution. This is some different distribution. Let's try smaller values of μ.

```
mu <- seq(from=0,to=50,length=10000)
likelihood <- likelihood_function(x=220,mu=mu)
plot(mu,likelihood)
```



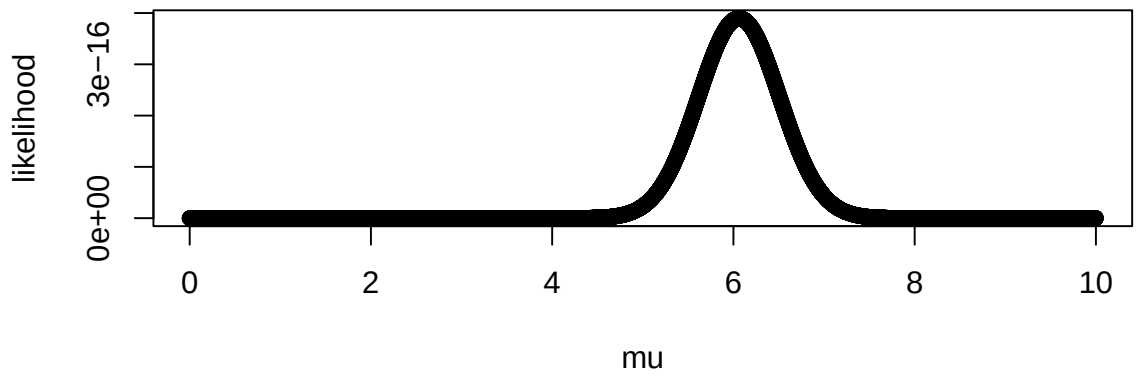

```r
# Even smaller

mu <- seq(from=0,to=10,length=10000)
likelihood <- likelihood_function(x=220,mu=mu)
plot(mu,likelihood)
```



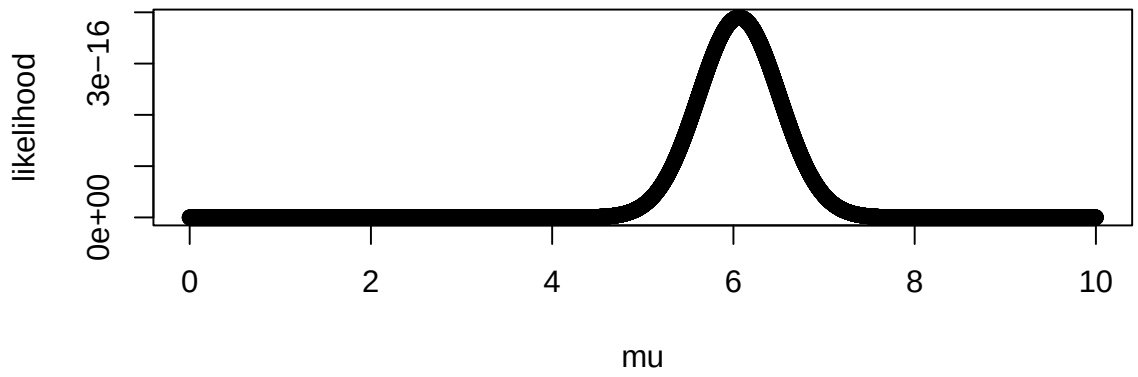

```r
# (b)
# When we have 5 observations
joint_likelihood_function <- function(x,mu){
  likelihood <- (1/(prod(x)*((sqrt(2*pi))^length(x))))*exp((-1/2)*sum((log(x)-mu)^2))
  return(likelihood)
}
```

```r
x <- c(303, 443, 220, 560, 880)
mu <- seq(from=0,to=10,length=10000)
likelihood <- rep(NA,10000)
for(i in 1:10000){
  likelihood[i] <- joint_likelihood_function(x=x,mu=mu[i])
}
plot(mu,likelihood)
```



Alternatively, I can use the `dlnorm` function in R. Because, the given PDF is the PDF of a lognormal distribution. You don't know this yet, but you will learn about it in the course eventually.

```r
x <- c(303, 443, 220, 560, 880)
mu <- seq(from=0,to=10,length=10000)
likelihood <- rep(NA,10000)
for(i in 1:10000){
  likelihood[i] <- prod(dlnorm(x=x,meanlog=mu[i],sdlog = 1))
}
plot(mu,likelihood)
```




```
mu[which(likelihood==max(likelihood))]
```

```
## [1] 6.061606
```

Part 4

4.1 Since, the three language corpora have equal number of word pairs. In order to know which language would have highest word pairs with distance 0, we need to know which language has the highest probability of occurrence of 0 distance between word pairs. English:

$P(X = 0) = f(0) = \frac{\lambda_1^0 e^{-\lambda_1}}{0!} = e^{-\lambda_1}$

German:

$P(X = 0) = f(0) = \frac{\lambda_2^0 e^{-\lambda_2}}{0!} = e^{-\lambda_2}$

Czech:

$P(X = 0) = f(0) = \frac{\lambda_3^0 e^{-\lambda_3}}{0!} = e^{-\lambda_3}$

We are given that $\lambda_3 < \lambda_1 < \lambda_2$

The language with λ_3 rate, i.e., Czech, will have the highest number of word pairs with linear distance of 0.

4.2 English: $P(X = 1) = \frac{\lambda_1^1 e^{-\lambda_1}}{1!} = 0.7 \times e^{-0.7} = 0.35$

German: $P(X = 1) = \frac{\lambda_2^1 e^{-\lambda_2}}{1!} = 0.8 \times e^{-0.8} = 0.36$

Czech: $P(X = 1) = \frac{\lambda_3^1 e^{-\lambda_3}}{1!} = 0.6 \times e^{-0.6} = 0.33$

4.3 You can do this manually and show the rough graph on paper. To do this in R, you can use the following code. You can of course write a better and more efficient code.

```
# Lambda for English, German, Czech and Artificial graphs
lambda <- c(0.7,0.8,0.6,2)

# Possible distances
distances <- 0:20
df.distribution <- data.frame(Linear_distance=distances)
df.distribution$English <- (lambda[1]^distances)*exp(-lambda[1])/factorial(distances)
df.distribution$German <- (lambda[2]^distances)*exp(-lambda[2])/factorial(distances)
df.distribution$Czech <- (lambda[3]^distances)*exp(-lambda[3])/factorial(distances)
df.distribution$Artificial <-
  (lambda[4]^distances)*exp(-lambda[4])/factorial(distances)


head(df.distribution)
```

```
##   Linear_distance      English      German        Czech Artificial
## 1               0 0.4965853038 0.449328964 0.5488116361 0.13533528
## 2               1 0.3476097127 0.359463171 0.3292869817 0.27067057
## 3               2 0.1216633994 0.143785269 0.0987860945 0.27067057
## 4               3 0.0283881265 0.038342738 0.0197572189 0.18044704
## 5               4 0.0049679221 0.007668548 0.0029635828 0.09022352
## 6               5 0.0006955091 0.001226968 0.0003556299 0.03608941
```

```
library(reshape2)
df.distribution.m <- melt(df.distribution,id="Linear_distance")

library(ggplot2)
ggplot(df.distribution.m,aes(x=Linear_distance,y=value,group=variable,
                             color=variable))+geom_point()+geom_line()
```