# Assignment-1

Andaleeb Hassan

## Binomial Distribution

If $X_1, X_2, \cdots, X_n \overset{\text{iid}}{\sim} B(k,p)$, then the probability mass function:

$$\mathbb{P}(X_i = x) = \binom{k}{x} p^x (1-p)^{k-x} \qquad\qquad p \in [0,1] \; and \; 0 \leq x \leq k$$

.

### likelyhood function:

$$\mathbb{L}(p \mid X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n) = \prod_{i=1}^{n} \binom{k}{x_i} p^{x_i} (1-p)^{k-x_i}$$

### log-likelyhood function:

If

$$l(p \mid X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n) = \ln\left(\prod_{i=1}^{n} \binom{k}{x_i}\right) + \ln p \; \Sigma_{i=1}^{n} x_i + \ln(1-p) \; \Sigma_{i=1}^{n}(k - x_i)$$

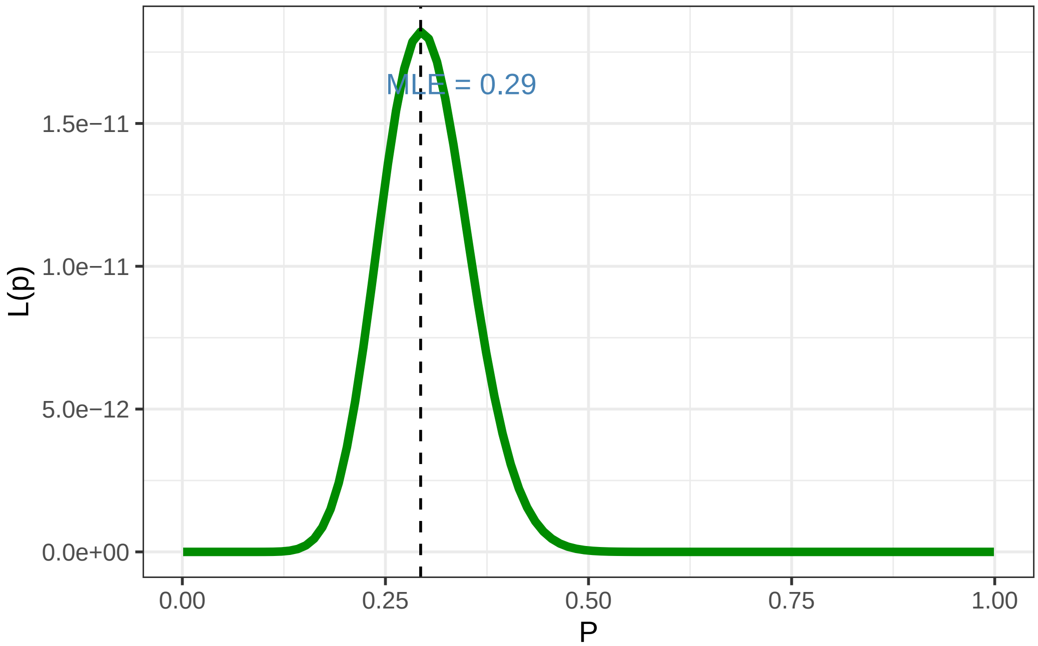
$$l(p \mid X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n) = M + \ln p \; \Sigma_{i=1}^{n} x_i + \ln(1-p) \; \Sigma_{i=1}^{n}(k - x_i) \qquad \text{[M is a constant]}$$

### plotting

```
Bino_like <- function(p, k, x) {
              if(p<= 0| p>=1){return(0)
              }else{
                  prod(choose(k,x)*p^x*(1-p)^(k-x))  }
  }

x <- c(rep(0,5),rep(1,7),rep(2,3),rep(3,1),rep(4,1))
p <- seq(0.001,.999,length.out = 100)
k <- 4
l <- c()

for(i in 1:100){
  l[i] = Bino_like(p[i],k,x)
}


MLE_p <- p[which.max(l)]
```

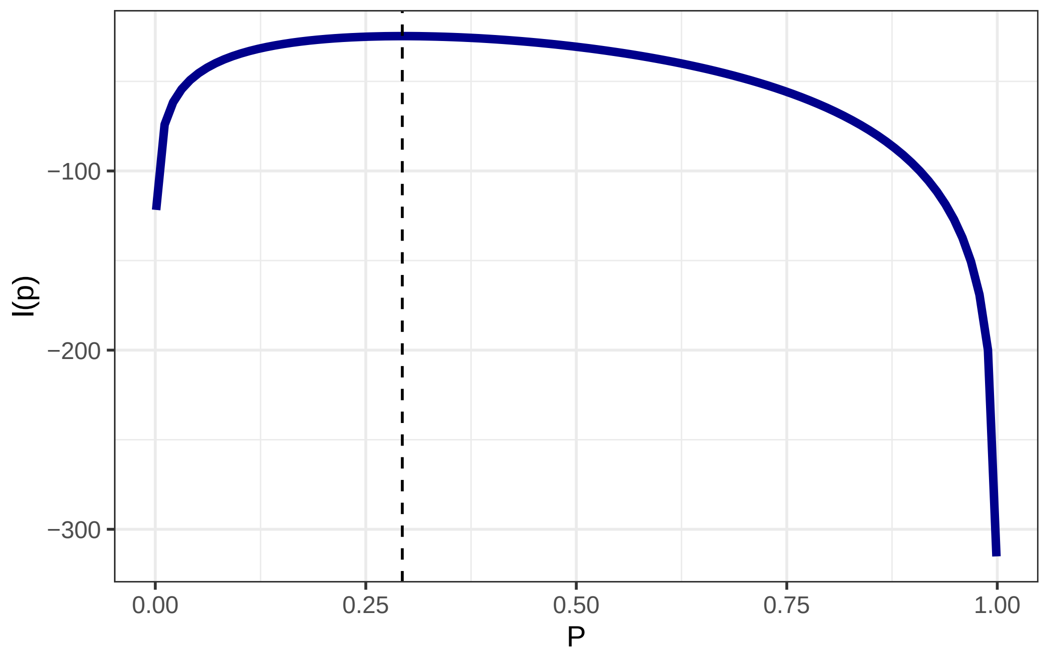
#### likelyhood

```
ggplot(mapping =aes(p,l))+
  geom_line(col = "green4",linewidth =1.5)+
  geom_vline(xintercept = p[which.max(l)],linetype = "dashed")+
  annotate("text", x = MLE_p + 0.05, y = max(l)*0.9,
label = paste0("MLE = ", round(MLE_p, 2)), color = "steelblue")+
  xlab("P")+
  ylab("L(p)")+
  theme_bw()
```




**log-likelyhood**

```
ggplot(mapping =aes(p,log(l)))+
  geom_line(col = "blue4",linewidth =1.5)+
  geom_vline(xintercept = p[which.max(l)],linetype = "dashed")+
  xlab("P")+
  ylab("l(p)")+
  theme_bw()
```




### the MLE

```
MLE_p <- p[which.max(l)]

round(MLE_p,3)
```

```
[1] 0.293
```

## Poisson Distribution

If $X_1, X_2, \cdots, X_n \overset{\text{iid}}{\sim} Poisson(\lambda)$, then the probability mass function:

$$\mathbb{P}(X_i = x) = \frac{\lambda^x e^{-\lambda}}{x!} \qquad\qquad \lambda \in (0, \infty) \; and x \in \mathbb{N} \; \cup \{0\}$$

.

### likelyhood function:

$$
\begin{aligned}
\mathbb{L}(\lambda \mid X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n) &= \prod_{i=1}^{n} \lambda^{x_i} e^{-\lambda} (x_i!)^{-1} \\
\mathbb{L}(\lambda \mid X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n) &= \lambda^{\Sigma_{i=1}^{n} x_i} e^{-n\lambda} (\prod_{i=1}^{n} (x_i!)^{-1}) \\
\mathbb{L}(\lambda \mid X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n) &= \lambda^{\Sigma_{i=1}^{n} x_i} e^{-n\lambda} M \qquad \text{[M is a constant]}
\end{aligned}
$$

### log-likelyhood function:

$$
l(\lambda \mid X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n) = ln(M) + \ln\lambda \ \Sigma_{i=1}^{n} x_i - n\lambda \qquad \text{[M is a constant]}
$$

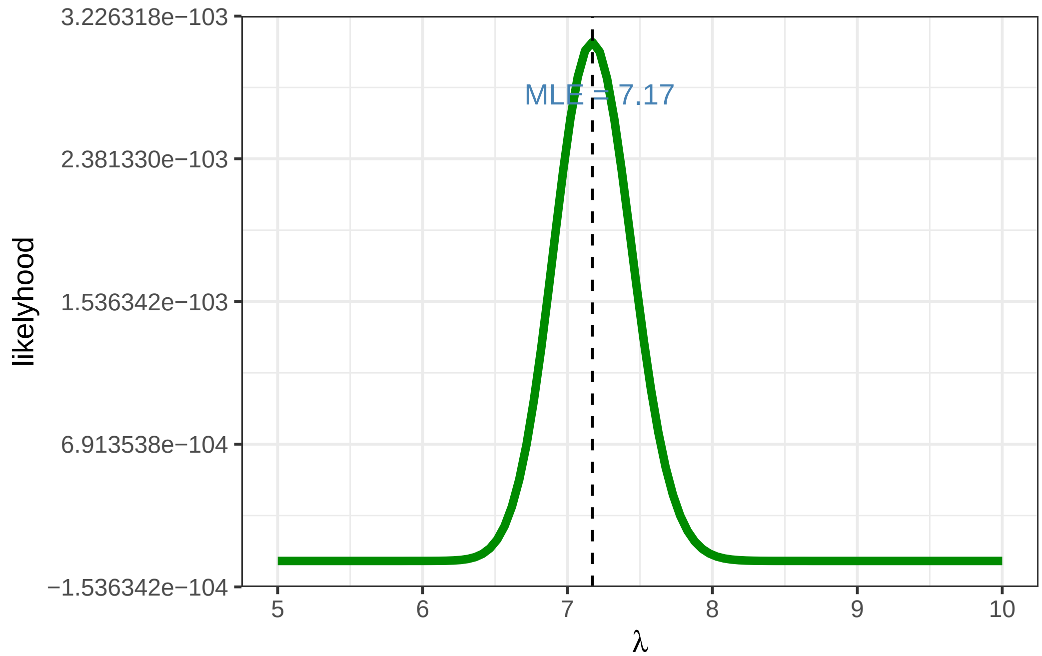
### plotting

```
Pois_like <- function(lambda,x){
  if(lambda <= 0){return(0)}
  else{    prod(exp(-lambda)*lambda^(x)*(factorial(x))^-1)}
}

set.seed(00007)


x <- rpois(100,lambda = 7)
lambda <- seq(5,10,length.out=100)
l <- c()
for(i in 1:100){
  l[i] <- Pois_like(lambda[i], x)
}
MLE_lambda <- lambda[which.max(l)]
```

#### likelyhood

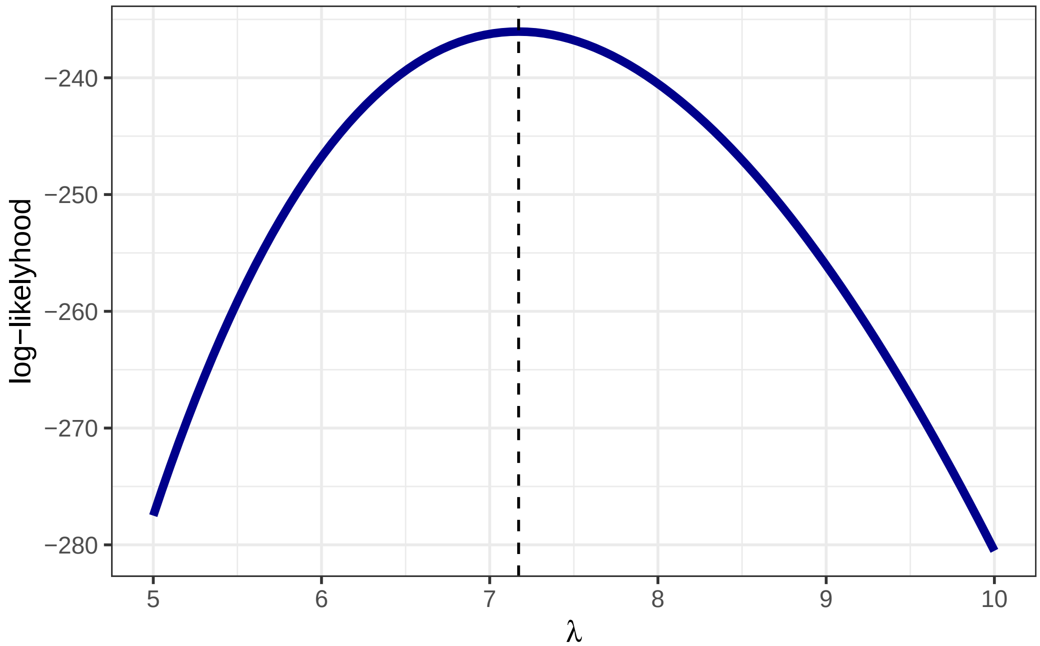
```
ggplot(mapping = aes(lambda,l))+
  geom_line(linewidth = 1.5, col = "green4")+
  labs(x = expression(lambda),y =  "likelyhood")+
  geom_vline(xintercept =lambda[which.max(l)],linetype ="dashed")+
  annotate("text", x = MLE_lambda + 0.05, y = max(l)*0.9,
```

```
label = paste0("MLE = ", round(MLE_lambda, 2)), color = "steelblue")+
  theme_bw()
```




### log-likelyhood

```
ggplot(mapping = aes(lambda,log(l)))+
  geom_line(linewidth = 1.5, col = "blue4")+
  labs(x = expression(lambda),y =  "log-likelyhood")+
  geom_vline(xintercept =lambda[which.max(l)],linetype ="dashed")+
  theme_bw()
```




### MLE

```
MLE_lambda <- lambda[which.max(l)]
round(MLE_lambda,2)
```

```
[1] 7.17
```

## Exponential Distribution

If $X_1, X_2, \cdots, X_n \overset{\text{iid}}{\sim} Exp(\lambda)$, then the probability density function:

$$f_X(x,\lambda) = \begin{cases} \lambda e^{-\lambda x} & ; x \geq 0 \\ 0 & ; x < 0 \end{cases} \qquad \text{[Where } \lambda \text{ is the rate parameter ]}$$

### likelyhood function:

$$\begin{aligned} \mathbb{L}(\lambda |\ X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n) = & \prod_{i=1}^{n} \lambda e^{-\lambda x_i}\ \mathbb{I}(xi \geq 0) \\ = & \lambda^n e^{-\lambda \Sigma_{i=1}^{n} x_i} \mathbb{I}(xi \geq 0) \end{aligned}$$

### log-likelyhood function:

$$l(\lambda |\ X_1 = x_1, X_2 = x_2, \cdots, X_n = x_n) = nln(\lambda) - \lambda \sum_{i=1}^{n} x_i\ \mathbb{I}(xi \geq 0)$$

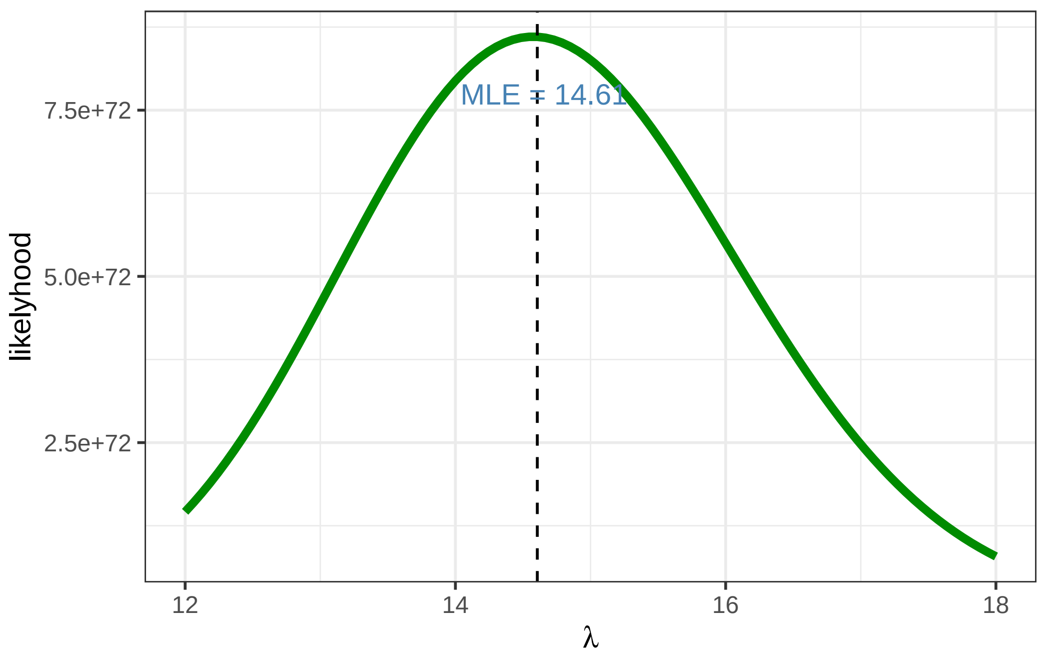
### plotting

```
 exp_like <- function(lambda,x){
   if(lambda <= 0){
     return(0)
   }
   else{
     prod(lambda*exp(-lambda*x))
   }

 }

set.seed(047)
x <- rexp(100,15)
lambda <- seq(12,18,length.out =100)
l <- c()
```

```
for(i in 1:100){
  l[i]  <- exp_like(lambda[i],x)
}

MLE_lambda <- lambda[which.max(l)]
```

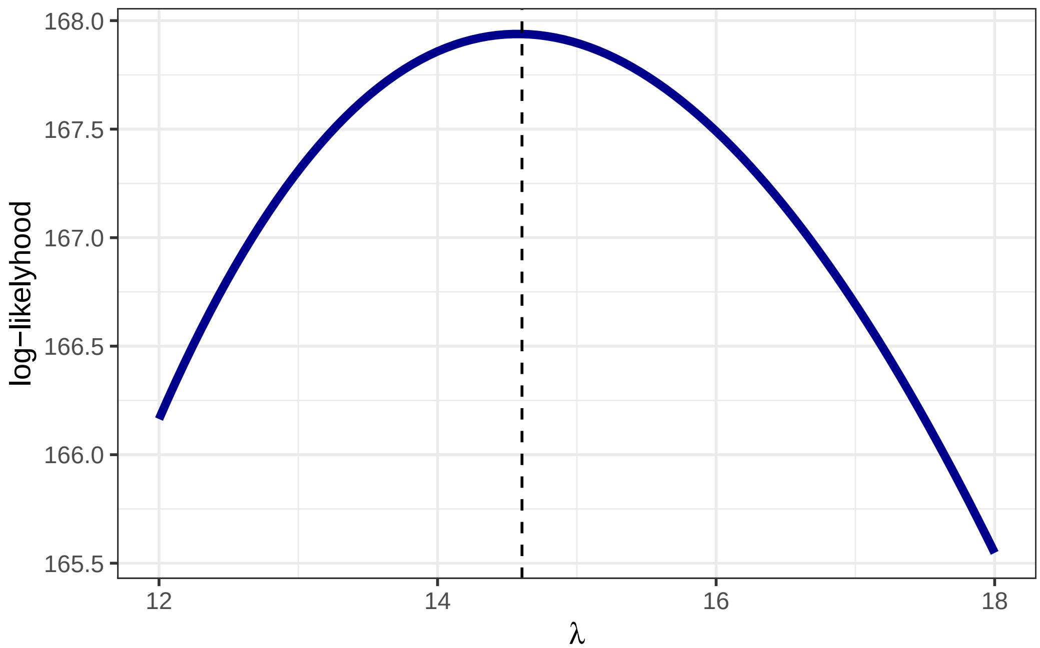
#### likelyhood

```
ggplot(mapping = aes(lambda,l))+
  geom_line(linewidth = 1.5, col = "green4")+
  labs(x = expression(lambda),y =  "likelyhood")+
  geom_vline(xintercept =lambda[which.max(l)],linetype
             ="dashed")+
  annotate("text", x = MLE_lambda + 0.05, y = max(l)*0.9,
label = paste0("MLE = ", round(MLE_lambda, 2)), color = "steelblue")+
  theme_bw()
```



#### log-likelyhood

```
ggplot(mapping = aes(lambda,log(l)))+
  geom_line(linewidth = 1.5, col = "blue4")+
  labs(x = expression(lambda),y =  "log-likelyhood")+
  geom_vline(xintercept =lambda[which.max(l)],linetype
             ="dashed")+
  theme_bw()
```




### MLE

```
MLE_lambda <- lambda[which.max(l)]
round(MLE_lambda,2)
```

```
[1] 14.61
```

## Normal Distribution

If $Y_1, Y_2, \cdots, Y_n \overset{\text{iid}}{\sim} N(\mu, \sigma^2)$, then the probability density function:

$$f_Y(y;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \qquad\qquad y \ \in \mathbb{R}, \mu \in \mathbb{R} \ and \ \sigma \in \mathbb{R}^+$$

### likelyhood function:

$$\begin{aligned}\mathbb{L}(\mu,\sigma^2|\ Y_1 = y_1, Y_2 = y_2, \cdots, Y_n = y_n) =& \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i-\mu)^2}{2\sigma^2}} \\ =& \frac{1}{(2\pi\sigma^2)^{\frac{n}{z}}} e^{-\frac{\Sigma_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}}\end{aligned}$$

### log-likelyhood function:

$$l(\mu,\sigma^2|\ Y_1 = y_1, Y_2 = y_2, \cdots, Y_n = y_n) = -\frac{n}{2} ln(2\pi\sigma^2) - \frac{\Sigma_{i=1}^{n}(y_i-\mu)^2}{2\sigma^2}$$

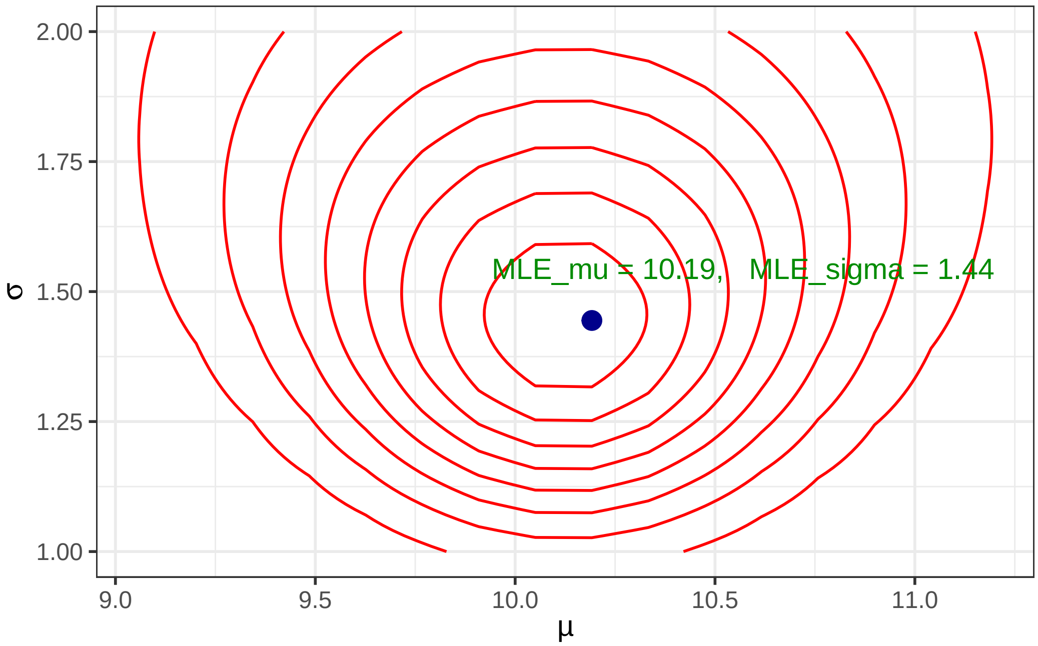
### plotting

```
normal_like <- function(mu,sigma,x){
  if(sigma <= 0){return(0)}
  else{
   prod((2*pi*sigma**2)^(-1/2)*exp(-(x-mu)**2*(2*sigma**2)^-1))
  }
}


set.seed(007)
x <- rnorm(10,mean = 10,sd = 1.2)
mu <- seq(1,15,length.out=100)
sigma <- seq(1,2,length.out = 100 )

l <- matrix(rep(0,100*100),nrow = 100,ncol =100)

for(i in 1:100){
```

```
  for(j in 1:100){
    l[i,j] <- normal_like(mu[i],sigma[j],x)
  }
}

position <- which( l == max(l), arr.ind = TRUE )

MLE_mu <- mu[position[1]]
MLE_sigma <- sigma[position[2]]
```

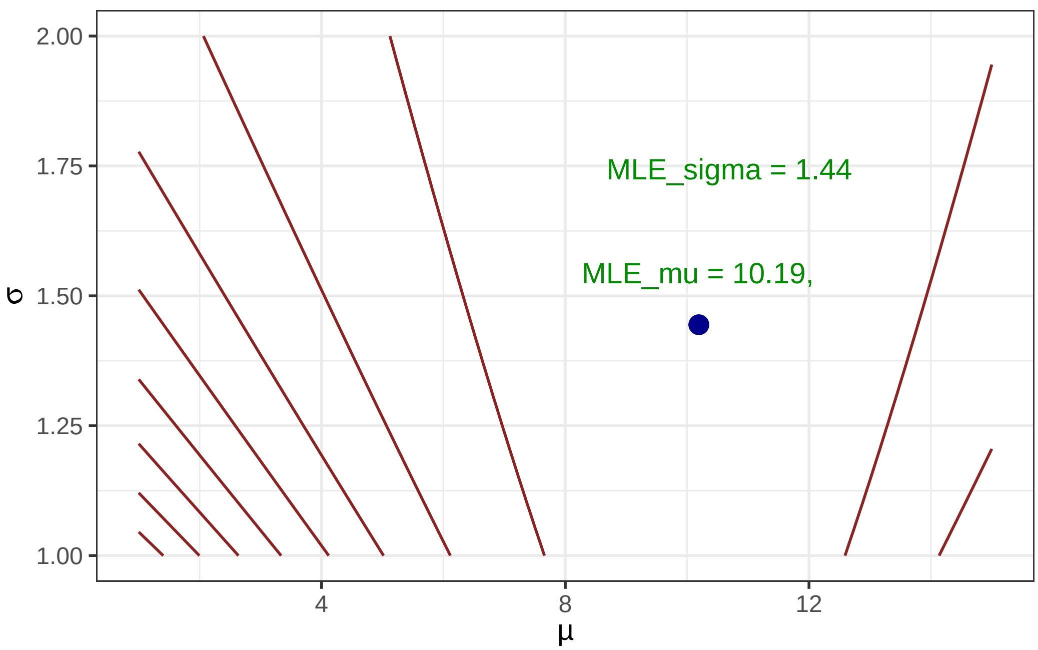
### likelyhood

```
data <- tibble( mu = rep(mu,times = 100),
                sigma = rep(sigma, each =100),
                l = c(l))
ggplot(data =data, mapping = aes(x=mu,y= sigma,z=l))+
  geom_contour(col = "red")+
  annotate(geom = "point",x= MLE_mu,y =MLE_sigma,size =3,col = "blue4")+
  labs(x=expression("\u03BC"),y=expression(sigma))+
  annotate("text", x = MLE_mu+ 0.05, y = MLE_sigma+.1,
label = paste0("MLE_mu = ", round(MLE_mu, 2),sep = ", "), color = "green4" )+
  annotate("text", x = MLE_mu+ .7, y = MLE_sigma+.1,
label = paste0("MLE_sigma = ", round(MLE_sigma, 2)), color = "green4")+
  theme_bw()
```




### log-likelyhood

```
ggplot(data =data, mapping = aes(x=mu,y= sigma,z=log(l)))+
  geom_contour(col = "brown4")+
  annotate(geom = "point",x= MLE_mu,y =MLE_sigma,size =3,col = "blue4")+
  labs(x=expression("\u03BC"),y=expression(sigma))+
  annotate("text", x = MLE_mu+ 0.05, y = MLE_sigma+.1,
label = paste0("MLE_mu = ", round(MLE_mu, 2),sep = ", "), color = "green4" )+
  annotate("text", x = MLE_mu+ .5, y = MLE_sigma+.3,
label = paste0("MLE_sigma = ", round(MLE_sigma, 2)), color = "green4")+
  theme_bw()
```




### MLE

```
position <- which( l == max(l), arr.ind = TRUE )

MLE_mu <- mu[position[1]]
MLE_sigma <- sigma[position[2]]

MLE_mu
```

```
[1] 10.19192
```

```
MLE_sigma
```

```
[1] 1.444444
```

## Gamma Distribution

If $Y_1, Y_2, \cdots, Y_n \overset{\text{iid}}{\sim} \Gamma(\alpha, \beta)$, then the probability density function:

$$f_Y(y\ |\alpha, \beta) = \frac{y^{\alpha-1}\ e^{-\frac{y}{\beta}}}{\beta^{\alpha}\ \Gamma(\alpha)} \qquad\qquad y > 0\ and\ \alpha, \beta > 0$$

### likelyhood function:

$$\begin{aligned}\mathbb{L}(\alpha, \beta|\ Y_1 = y_1, Y_2 = y_2, \cdots, Y_n = y_n) = &\prod_{i=1}^{n} \frac{y_i^{\alpha-1}\ e^{-\frac{y_i}{\beta}}}{\beta^{\alpha}\ \Gamma(\alpha)} \\ =&(\prod_{i=1}^{n} y_i^{\alpha-1})\ e^{-\sum_{i=1}^{n} \frac{y_i}{\beta}} \beta^{-n\alpha}\ (\Gamma(\alpha))^{-n}\end{aligned}$$

### log-likelyhood function:

$$l(\alpha, \beta|\ Y_1 = y_1, Y_2 = y_2, \cdots, Y_n = y_n) = (\alpha - 1)\sum_{i=1}^{n} \ln(y_i) - \beta^{-1}\sum_{i=1}^{n} y_i - n\alpha\ \ln(\beta) - n\ln(\Gamma(\alpha))$$

### plotting

```
gamma_like <- function(alpha,beta,x){

   prod(x^(alpha-1)*exp(-x/beta)*beta^(-alpha)*(gamma(alpha))^-1)

}

set.seed(007)

x     <- rgamma(50,10,scale=5)

alpha <- seq(5,15,length.out=100)
beta  <- seq(.5,7,length.out = 100)

l <- matrix(rep(0,100*100),nrow = 100,ncol =100)
```

```r
for(i in 1:100){
  for(j in 1:100){
    l[i,j] <- gamma_like(alpha[i],beta[j],x)
  }
}

position <- which( l == max(l), arr.ind = TRUE )

MLE_alpha <- alpha[position[1]]
MLE_beta <- beta[position[2]]

MLE_alpha
```

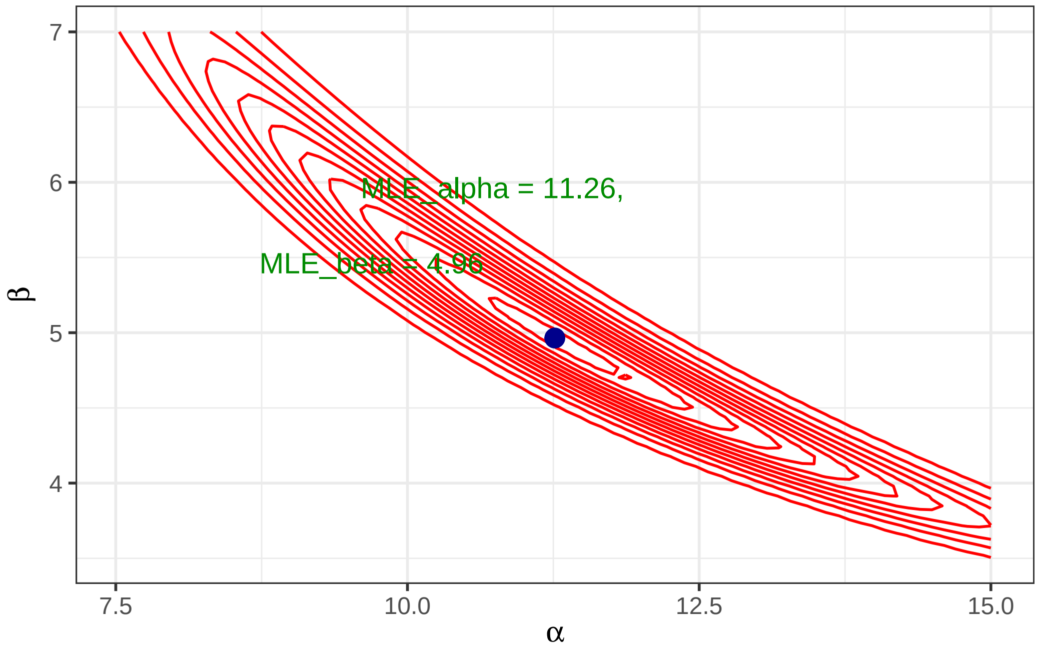
```
[1] 11.26263
```

```r
MLE_beta
```

```
[1] 4.964646
```

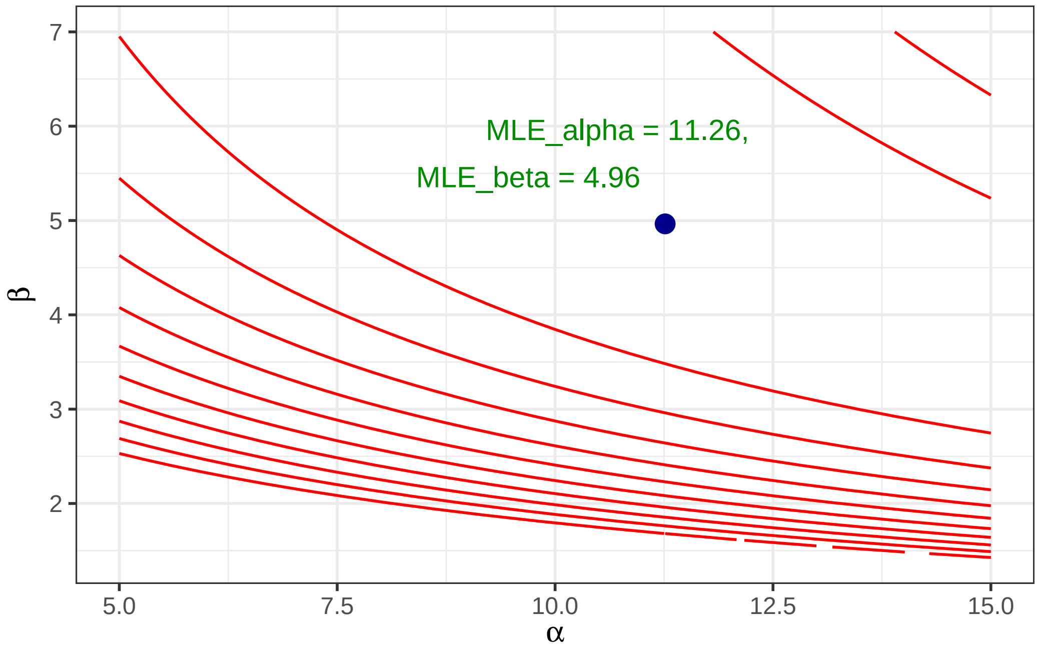
#### likelyhood

```r
data <- tibble( alpha = rep(alpha,times = 100),
                beta = rep(beta, each =100),
                l = c(l))
ggplot(data =data, mapping = aes(x=alpha,y= beta,z=l))+
  geom_contour(col = "red")+
  annotate(geom = "point",x= MLE_alpha,y =MLE_beta,size =3,col = "blue4")+
  labs(x=expression(alpha),y=expression(beta))+
  annotate("text", x = MLE_alpha- 0.5, y = MLE_beta+1,
label = paste0("MLE_alpha = ", round(MLE_alpha, 2),sep = ", "), color = "green4" )+
  annotate("text", x = MLE_mu -.5, y = MLE_beta+.5,
label = paste0("MLE_beta = ", round(MLE_beta, 2)), color = "green4")+
  theme_bw()
```




**log-likelyhood**

```
data <- tibble( alpha = rep(alpha,times = 100),
                beta = rep(beta, each =100),
                l =log( c(l)))
ggplot(data =data, mapping = aes(x=alpha,y= beta,z=l))+
  geom_contour(col = "red")+
  annotate(geom = "point",x= MLE_alpha,y =MLE_beta,size =3,col = "blue4")+
  labs(x=expression(alpha),y=expression(beta))+
  annotate("text", x = MLE_alpha- 0.5, y = MLE_beta+1,
label = paste0("MLE_alpha = ", round(MLE_alpha, 2),sep = ", "), color = "green4" )+
  annotate("text", x = MLE_mu -.5, y = MLE_beta+.5,
label = paste0("MLE_beta = ", round(MLE_beta, 2)), color = "green4")+
  theme_bw()
```




### MLE

```
position <- which( l == max(l), arr.ind = TRUE )

MLE_alpha <- alpha[position[1]]
MLE_beta <- beta[position[2]]

MLE_alpha
```

```
[1] 11.26263
```

```
MLE_beta
```

```
[1] 4.964646
```

## Beta Distribution

If $Y_1, Y_2, \cdots, Y_n \overset{\text{iid}}{\sim} Beta(\alpha, \beta)$, then the probability density function:

$$f_Y(y\mid\alpha,\beta) = \frac{y^{\alpha-1}(1-y)^{\beta-1}}{\mathbb{B}(\alpha,\beta)} \qquad\qquad y>0 \; and \; \alpha,\beta>0$$

**likelyhood function:**

$$\begin{aligned}\mathbb{L}(\alpha,\beta\mid Y_1=y_1,Y_2=y_2,\cdots,Y_n=y_n) &= \prod_{i=1}^{n}\frac{y_i^{\alpha-1}(1-y_i)^{\beta-1}}{\mathbb{B}(\alpha,\beta)}\\ &=(\prod_{i=1}^{n} y_i^{\alpha-1}(1-y_i)^{\beta-1})\mathbb{B}(\alpha,\beta)^{-n}\end{aligned}$$

**log-likelyhood function:**

$$l(\alpha,\beta\mid Y_1=y_1,Y_2=y_2,\cdots,Y_n=y_n) = (\alpha-1)\sum_{i=1}^{n}\ln(y_i)+(\beta-1)\sum_{i=1}^{n}\ln(1-y_i)-n\ln(\mathbb{B}(\alpha,\beta))$$

**plotting**

```
 beta_like <- function(alpha,beta,x){

   if(alpha <= 0 || beta <= 0 ){

   }else{prod(x^(alpha-1)*(1-x)^(beta-1)*beta(alpha,beta)^-1)

   }


 }

set.seed(07)

x <- rbeta(100,10,15)


alpha <- seq(5,15,length.out=100)
beta  <- seq(10,20,length.out = 100)

l <- matrix(rep(0,100*100),nrow = 100,ncol =100)
```

```
for(i in 1:100){
  for(j in 1:100){
    l[i,j] <- beta_like(alpha[i],beta[j],x)
  }
}

position <- which( l == max(l), arr.ind = TRUE )

MLE_alpha <- alpha[position[1]]
MLE_beta <- beta[position[2]]

MLE_alpha
```

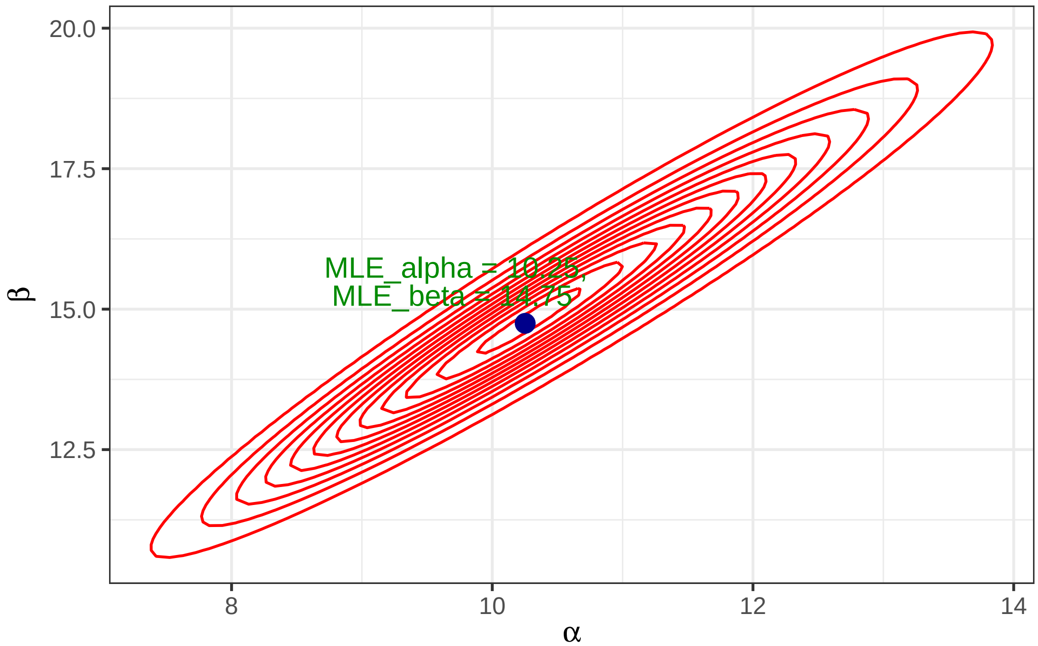
```
[1] 10.25253
```

```
MLE_beta
```

```
[1] 14.74747
```

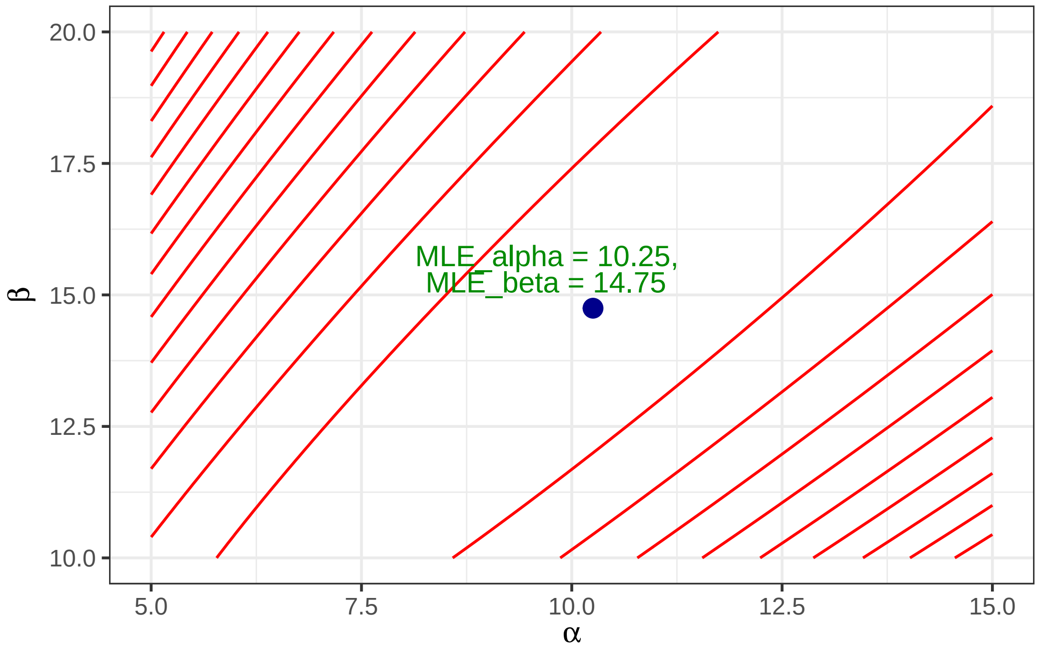
#### likelyhood

```
data <- tibble( alpha = rep(alpha,times = 100),
                beta = rep(beta, each =100),
                l = c(l))
ggplot(data =data, mapping = aes(x=alpha,y= beta,z=l))+
  geom_contour(col = "red")+
  annotate(geom = "point",x= MLE_alpha,y =MLE_beta,size =3,col = "blue4")+
  labs(x=expression(alpha),y=expression(beta))+
  annotate("text", x = MLE_alpha- 0.5, y = MLE_beta+1,
label = paste0("MLE_alpha = ", round(MLE_alpha, 2),sep = ", "), color = "green4" )+
  annotate("text", x = MLE_mu -.5, y = MLE_beta+.5,
label = paste0("MLE_beta = ", round(MLE_beta, 2)), color = "green4")+
  theme_bw()
```




#### log-likelyhood

```
data <- tibble( alpha = rep(alpha,times = 100),
                beta = rep(beta, each =100),
                l =log( c(l)))
ggplot(data =data, mapping = aes(x=alpha,y= beta,z=l))+
  geom_contour(col = "red")+
  annotate(geom = "point",x= MLE_alpha,y =MLE_beta,size =3,col = "blue4")+
  labs(x=expression(alpha),y=expression(beta))+
  annotate("text", x = MLE_alpha- 0.5, y = MLE_beta+1,
label = paste0("MLE_alpha = ", round(MLE_alpha, 2),sep = ", "), color = "green4" )+
  annotate("text", x = MLE_mu -.5, y = MLE_beta+.5,
label = paste0("MLE_beta = ", round(MLE_beta, 2)), color = "green4")+
  theme_bw()
```




### MLE

```
position <- which( l == max(l), arr.ind = TRUE )

MLE_alpha <- alpha[position[1]]
MLE_beta <- beta[position[2]]

MLE_alpha
```

```
[1] 10.25253
```

```
MLE_beta
```

```
[1] 14.74747
```