Exercises l5-l6

#statistics

Example 5.1

$1 = \sum_{x=1}^{\infty} c_\theta \theta^x = c_\theta(\frac{1}{1-\theta} - 1)$
and solving the equation we get $f(x;\theta) = (1-\theta)\theta^{x-1}$ a geometric rv with parameter $1-\theta$.

and solving the equation we get $f(x;\theta) = (1-\theta)\theta^{x-1}$ a geometric rv with parameter $1-\theta$.
(a) Wald confidence interval for θ of level $1-\alpha$, $\alpha = 0.9$ $\overline{x} = 10$ and $n = 30$.
$|W| = |\frac{\hat{\theta}-\theta_0}{\widehat{se}}|$

$L(\theta) = \frac{(1-\theta)^n}{\theta^n}\theta^{\sum_{i=1}^{n} x_i} = (1-\theta)^n \theta^{n(\overline{x}-1)}$
$\ell(\theta) = n\log(1-\theta) + n(\overline{x}-1)\log\theta$
$J(\theta) = \frac{n(\overline{x}-1)}{\theta^2} + \frac{n}{(1-\theta)^2}$
$\hat{\theta} = \frac{\overline{x}-1}{\overline{x}} = 0.9$
$J(\hat{\theta}) = 3333.3333$
$\widehat{se} = 1/\sqrt{\hat{J}} = 0.01732$

$R = \{\boldsymbol{X} : |W| \geq z_{1-\frac{\alpha}{2}}\} = \{\boldsymbol{X} : \theta_0 \leq \hat{\theta} - z_{1-\frac{\alpha}{2}}\widehat{se}\} \vee \theta_0 \geq \hat{\theta} + z_{1-\frac{\alpha}{2}}\widehat{se}\}$
$R^{\mathsf{c}} = \{\boldsymbol{X} : \hat{\theta} - z_{1-\frac{\alpha}{2}}\widehat{se} \leq \theta_0 \leq \hat{\theta} + z_{1-\frac{\alpha}{2}}\widehat{se}\}$

$z_{1-\frac{0.9}{2}} = 0.1256613$
confidence interval of level $1-\alpha = 0.1$: $CI = [0.8978, 0.9022]$
(b) I would reject H_0 since $\theta_0 = 0.5$ is outside the confidence interval.

(c) The likelihood ratio test statistic is $\lambda(\boldsymbol{x}) = \frac{L(\theta_0)}{L(\hat{\theta})} = \frac{L(\theta_0)}{4.42 \cdot 10^{-43}}$
$R_\alpha(\theta_0) = \{\boldsymbol{x} : -2\log(\lambda(\boldsymbol{x})) > \chi^2_{1,1-\alpha}\}$
We can use this region because the shipment is made of a very large number of items (asymptotic approximation)
We can compute $-2\log(\lambda(\boldsymbol{x}))$ and $\chi^2_{1,1-\alpha}$ and plot them, then we take the values of θ for which the sample is in the acceptance region, this is done with the following R script:

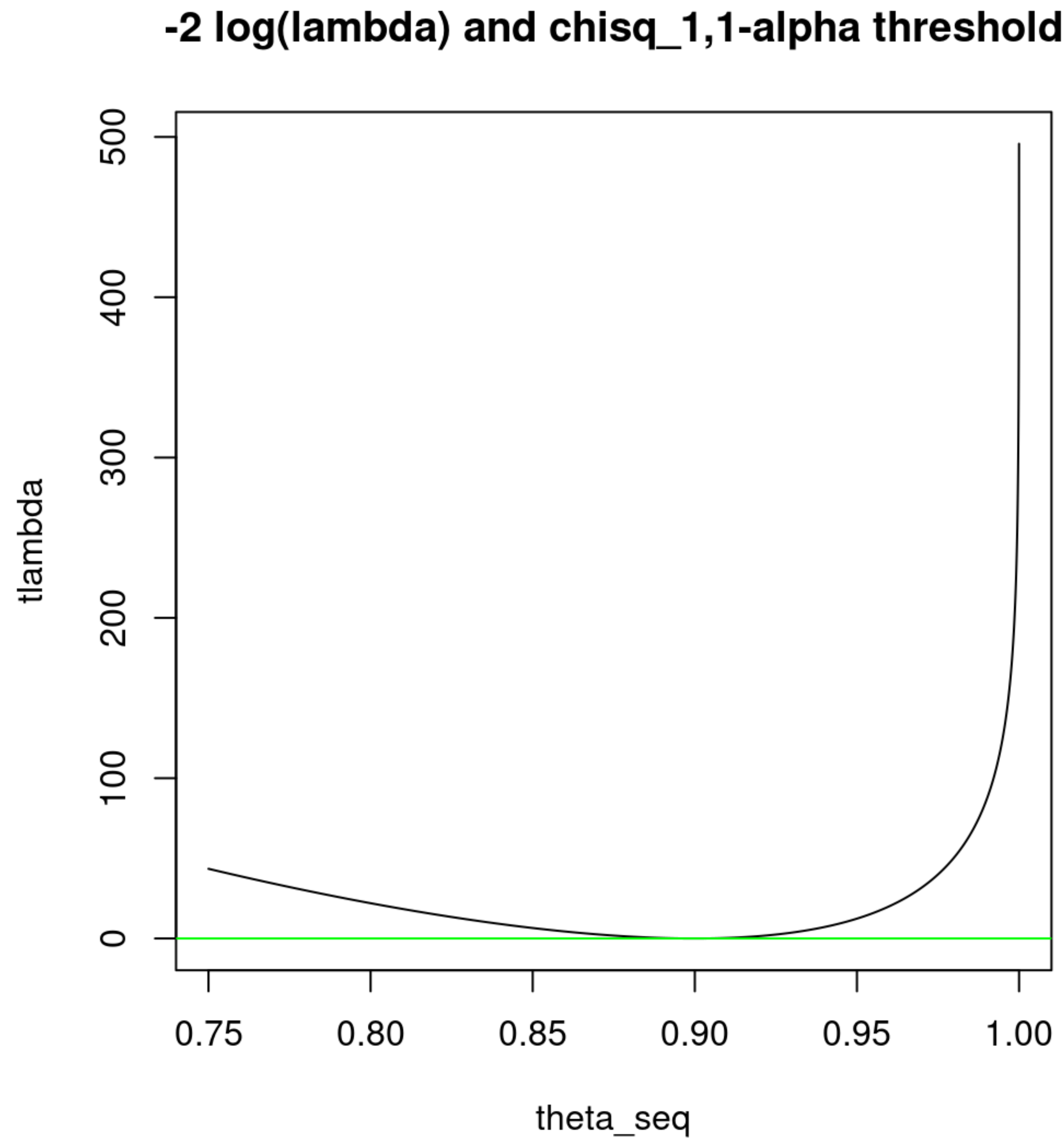
```
ell_f <-function(theta){
  n=30
  bar_x = 10
  n*log(1-theta)+n*(bar_x-1)*log(theta)
}
theta_seq = seq(0.75,1,0.00001)

alpha = 0.9
threshold = qchisq(p = 1-alpha,df = 1)
MLE = 0.9
tlambda = -2*ell_f(theta_seq)+2*ell_f(MLE)
plot(theta_seq,tlambda,main = '-2 log(lambda) and chisq_1,1-alpha threshold',lwd=1,lty=1,type
```

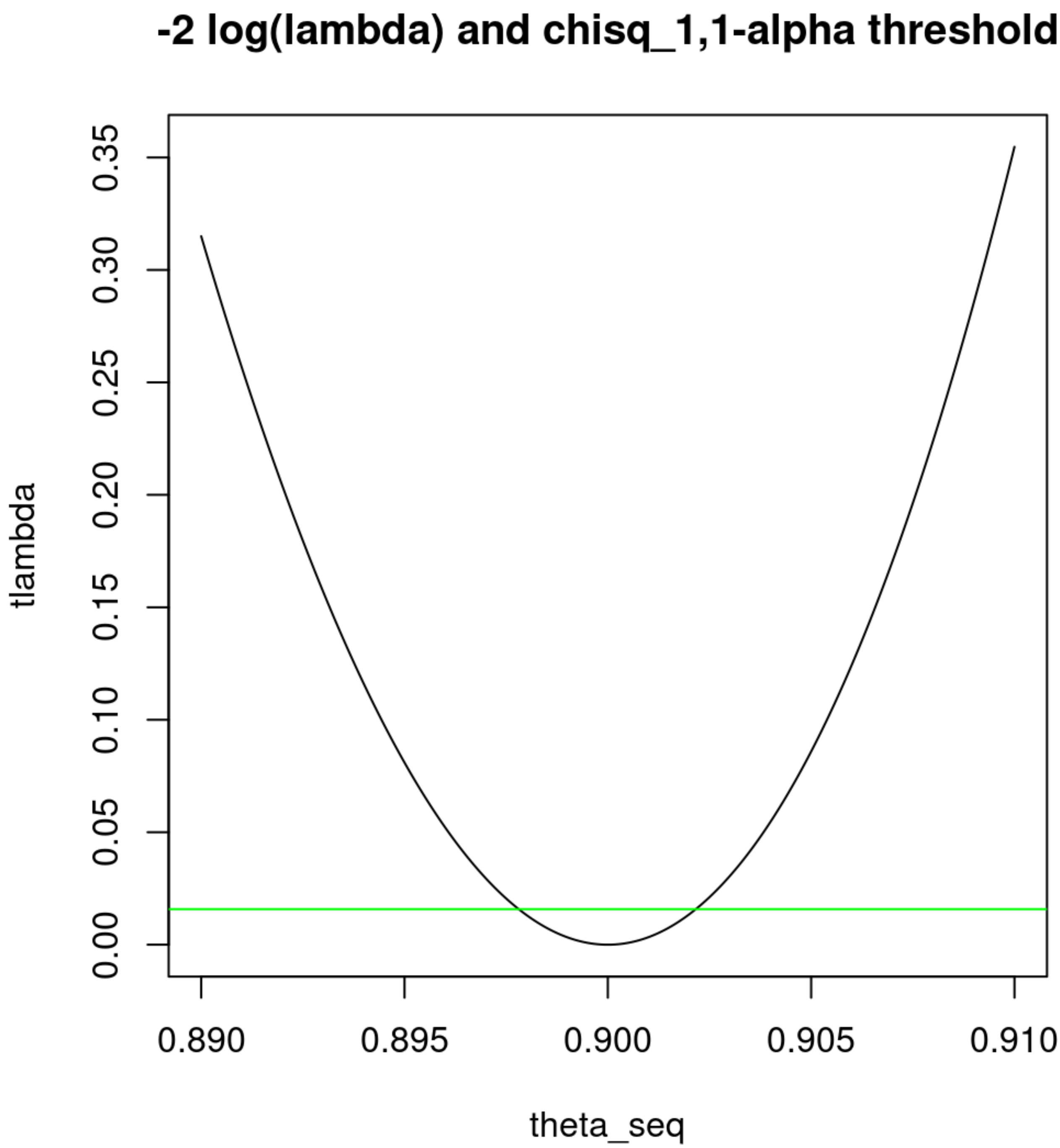
```
= 'l')
abline(h=threshold,col='green')

confidence_interval = theta_seq[tlambda<threshold]
sprintf('[ %f , %f ]',confidence_interval[1],confidence_interval[length(confidence_interval)])
```

that outputs:

```
"[ 0.897810 , 0.902160 ]"
```

which is the confidence interval for $\alpha = 0.9$.





(d) p-value: probably this point is wrong.

$\text{p-value} = P_\theta(-2\log(\lambda) \geq -2\log(\lambda_{obs}))$

$\lambda_{obs} = 1$

Take the log on both sides and multiply by -2.

$\text{p-value} = P_\theta(\chi_1^2 \geq 0) = 1.$

Example 5.2

Let $X_i \sim Poi(\theta_1), i = 1, \ldots, m$ and $Y_j \sim Poi(\theta_2), j = 1, \ldots, n$, with X_i, Y_jbeing independent for all i, j.

(a) Log-likelihood ratio test for $H_0 : \theta_1 = \theta_2$ against $H_1 : \theta_1 \neq \theta_2$ at the level α. $\overline{y} = 6, \overline{x} = 2, m = 15, n = 10$, compute the test and get the p-value.

The joint pdf is $f_{joint}(x_1, \ldots, x_m, y_1, \ldots, y_n) = \frac{e^{-m\theta_1}\theta_1^{\sum_{i=1}^{m} x_i}}{\prod_{i=1}^{m} x_i!} \frac{e^{-n\theta_2}\theta_2^{\sum_{i=1}^{n} y_i}}{\prod_{i=1}^{n} y_i!}$

if $\theta_1 = \theta_2$ (null hypothesis) the likelihood is $\frac{e^{-\theta(m+n)}\theta^{\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i}}{\prod_{i=1}^{m} x_i! \prod_{i=1}^{n} y_i!}$

and the log likelihood is $\ell(\theta) = -\theta(m+n) + (\log\theta)(\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i) - \log(\prod_{i=1}^{m} x_i! \prod_{i=1}^{n} y_i!)$

we solve the likelihood equation ($\frac{\partial \ell(\theta)}{\partial \theta} = 0$) and get $\hat{\theta} = \frac{\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i}{m+n}$ under H_0

to find the denominator of $\lambda(\cdot)$ we can use the equivariance principle of the MLE and $\hat{\theta}_1 = \overline{x}$ and $\hat{\theta}_2 = \overline{y}$.

We have that

$$\log(\lambda(\cdot)) = -(\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i) + \log(\frac{\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i}{m+n})(\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i) - \log(\prod_{i=1}^{m} x_i! \prod_{i=1}^{n} y_i!) - (-m\overline{x} - n\overline{y} + (\sum_{i=1}^{m} x_i)\log(\overline{x}) + (\sum_{i=1}^{n} y_i)\log(\overline{y}) \cdot$$

$$= \log(\frac{\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i}{m+n})(\sum_{i=1}^{m} x_i + \sum_{i=1}^{n} y_i) - (\sum_{i=1}^{m} x_i)\log(\overline{x}) - (\sum_{i=1}^{n} y_i)\log(\overline{y}) =$$

$= \log(\frac{m\overline{x}+n\overline{y}}{m+n})(m\overline{x} + n\overline{y}) - m\overline{x}\log(\overline{x}) - n\overline{y}\log(\overline{y})$ we call this $\log(\lambda(\boldsymbol{x}, \boldsymbol{y}))$

And we have that the rejection region is $\{\boldsymbol{x}, \boldsymbol{y} : -2\log(\lambda(\boldsymbol{x}, \boldsymbol{y})) > \chi^2_{1,1-\alpha}\}$.

using the observed samples we get

```
> 1-pchisq(26.034,df=1)
[1] 3.354574e-07
```

$\{\boldsymbol{x}, \boldsymbol{y} : 26.03 > \chi^2_{2,1-\alpha}\}$.
and we get that $p - value \approx 2.222 \cdot 10^{-6}$

(b) If we define $\delta \triangleq \theta_1 - \theta_2$ we have that $\hat{\delta} = \hat{\theta}_1 - \hat{\theta}_2$ for the principle of equivariance.
$\hat{\delta} = \overline{x} - \overline{y} = -4$
and the confidence interval of approximately confidence level $1 - \alpha$ is
$[\hat{\delta} - z_{1-\frac{\alpha}{2}}\widehat{se}, \hat{\delta} + z_{1-\frac{\alpha}{2}}\widehat{se}]$
we just need to compute $\widehat{se}$, we have that asymptotically MLE is normal, $\hat{\theta}_1 \dot{\sim} \mathcal{N}(\theta_1, \mathrm{var}(\hat{\theta}_1))$, $\hat{\theta}_2 \dot{\sim} \mathcal{N}(\theta_2, \mathrm{var}(\hat{\theta}_2))$
and since $\hat{\theta}_1 = \overline{X}$ and $\hat{\theta}_2 = \overline{Y}$ they are independent because they are functions of independent rvs. This means that:
$\hat{\delta} \dot{\sim} \mathcal{N}(\overline{X} - \overline{Y}, \mathrm{var}(\overline{X}) + \mathrm{var}(\overline{Y}))$
and using the formula for the variance of the sample average and the variance of Poisson rvs we get:
$\hat{\delta} \dot{\sim} \mathcal{N}(\overline{X} - \overline{Y}, \frac{\theta_1}{m} + \frac{\theta_2}{n})$
$se = \sqrt{\frac{\theta_1}{m} + \frac{\theta_2}{n}}$
$\widehat{se} = \sqrt{\frac{\overline{X}}{m} + \frac{\overline{Y}}{n}}$
using the observed values we get:
$\widehat{se} = 0.8563$
$\mathrm{CI}_{1-\alpha} = [-4 - z_{1-\frac{\alpha}{2}}0.8563, -4 + z_{1-\frac{\alpha}{2}}0.8563]$
and for example for $1 - \alpha = 99\%$ we can compute with R:

```
> alpha = 1 - 0.99
> z = qnorm(1-alpha/2)
> c(-4-z*0.8563,-4+z*0.8536)
[1] -6.205683 -1.801272
```

which means that $\hat{\delta} = 0$ is outside of our confidence interval of confidence 99%

if we take $\alpha = 10^{-6}$ we get the confidence interval $[-8.1887100, 0.1755026]$ which includes 0.

Example 5.3

$X_i \sim \mathcal{N}(\mu, \sigma^2), i = 1, \ldots, n$ both parameters are unknown
(a) $H_0 : \mu = 1$ against $H_1 : \mu \neq 1$ and observed sample with $\overline{x} = 2.1$ and $s^2 = 1.2$ determine n needed for $\beta(2) = 0.01$, (since $\mu = 2$ we are under H_1)

$\beta(2) = P_{\mu=2}(\boldsymbol{x} \notin R)$
if R is defined with a LRT we are in the case of a t-test,
$R = \{\boldsymbol{x} : |\frac{\sqrt{n}(\overline{x}-\mu_0)}{s}| \geq t_{n-1,1-\frac{\alpha}{2}}\}$
$\beta(2) = 1 - P_{\mu=2}(\text{reject } H_0) = P_{\mu=2}(\boldsymbol{x} \notin R) = 0.01$
$\overline{x} \sim \mathcal{N}(\mu, \frac{\sigma^2}{n})$
$T_n \sim \mathcal{N}(\frac{\sqrt{n}}{s}, \frac{\sigma^2}{s^2})$
$\beta(2) = 1 - 2\Phi\left(\left(t_{n-1,1-\frac{\alpha}{2}} - \frac{\sqrt{n}}{s}\right)\frac{s}{\sigma}\right)$
how do I continue?

Example 5.4

$X_1, \ldots, X_n$ iid random sample from $\text{Unif}(0, \theta)$, $\theta > 0$. Construct a $1-\alpha$ confidence interval for θ.

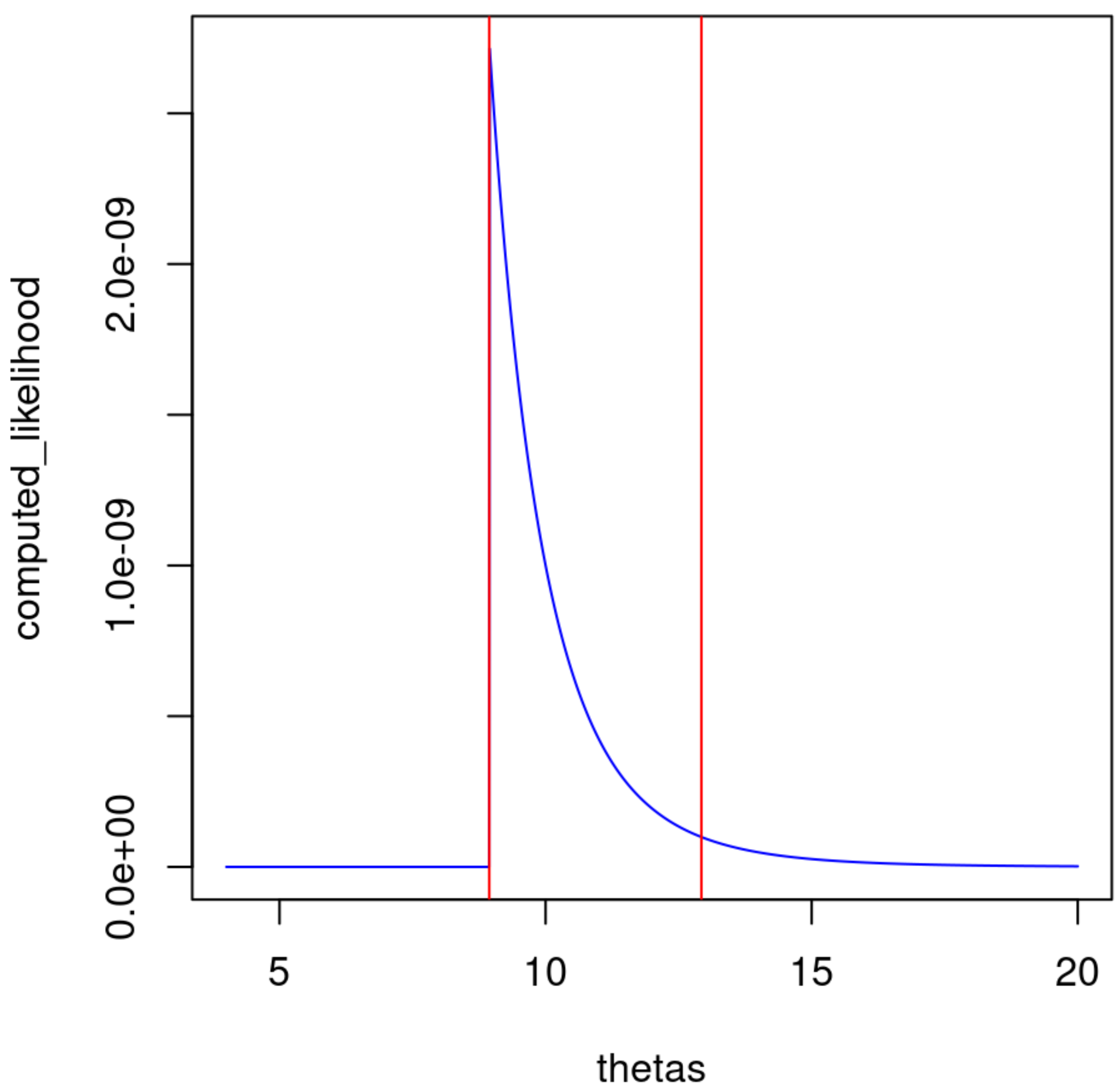
We use a LRT, we have that $L(\theta) = \frac{1}{\theta^n}$ if $x_{(n)} \leq \theta$ and 0 otherwise.
$R_\alpha(\theta_0) = \{\boldsymbol{x} : -2\log(\lambda(\boldsymbol{x})) > \chi^2_{1,1-\alpha}\}$
$\hat{\theta} = x_{(n)}$
$\lambda(\boldsymbol{x}) = x^n_{(n)} \cdot \frac{1}{\theta^n}$ for $\theta \geq x_{(n)}$
$\lambda(\boldsymbol{x}) = 0$ for $\theta < x_{(n)}$ (this means that we always reject independently of the sample)
$R_\alpha(\theta_0) = \{\boldsymbol{x} : e^{-2\log(\lambda(\boldsymbol{x}))} > e^{\chi^2_{1,1-\alpha}}\}$

$R^\complement_\alpha(\theta_0) = \{\boldsymbol{x} : e^{-2\log(\lambda(\boldsymbol{x}))} \leq e^{\chi^2_{1,1-\alpha}}\}$
$R^\complement_\alpha(\theta_0) = \{\boldsymbol{x} : (\lambda(\boldsymbol{x}))^{-2} \leq e^{\chi^2_{1,1-\alpha}}\}$
$R^\complement_\alpha(\theta_0) = \{\boldsymbol{x} : \frac{\theta_0^{2n}}{x^{2n}_{(n)}} \leq e^{\chi^2_{1,1-\alpha}}\}$
$R^\complement_\alpha(\theta_0) = \{\boldsymbol{x} : \frac{\theta_0}{x_{(n)}} \leq e^{\frac{\chi^2_{1,1-\alpha}}{2n}}\}$
$R^\complement_\alpha(\theta_0) = \{\boldsymbol{x} : \theta_0 \leq x_{(n)} e^{\frac{\chi^2_{1,1-\alpha}}{2n}}\}$
also we need to remember that $\theta_0 \geq x_{(n)}$.
this means that the $\text{CI}_{1-\alpha} = [x_{(n)}, x_{(n)} e^{\frac{\chi^2_{1,1-\alpha}}{2n}}]$

an example in R is:

```
likelihood_f <- function(theta, sample_max, n) {
  if (theta < sample_max) {
    return(0)
  } else{
    return(1 / theta ^ n)
  }
}
true_theta <- 10
n <- 9
alpha <- 0.01
obs <- runif(n = n, min = 0, max = true_theta)
sample_max <- max(obs)
thetas <- seq(4, 20, 0.01)
computed_likelihood <-
  sapply(thetas, likelihood_f, sample_max = sample_max, n = n)
plot(thetas,
     computed_likelihood,
     type = 'l',
     col = 'blue')
abline(v = sample_max, col = 'red')
right_CI = sample_max * exp(qchisq(p = 1 - alpha, df = 1) / (2 * n))
abline(v = right_CI, col = 'red')
sprintf("[%f,%f]", sample_max, right_CI)
```

and the CI is `[8.941970,12.927558]` using $\alpha = 0.01$,



Example 5.5

We can use the following R script:

```
N <- 1e4
mu <- 0
sigma_sq_vals <- c(0.5,1,2)
```

```
n_vals <- c(10, 30)
confidence <- 0.95#1-alpha
alpha <- 1 - confidence
z_wald <- qnorm(1 - alpha / 2)

n_val <- n_vals[2]
for (n_val in n_vals) {
  for (sigma_sq in sigma_sq_vals) {
    print('------------------------------------------')
    print(sprintf("sigma^2 = %f, n of obs = %d, mu = %f", sigma_sq, n_val, mu))
    expect_len_wald <- 0
    coverage_prob_wald <- 0
    expect_len_lrtq <- 0
    coverage_prob_lrtq <- 0
    for (i in 1:N) {
      obs <- rnorm(n_val, mean = 0, sd = sqrt(sigma_sq))
      sample_average <- mean(obs)
      sigma_hat_sq <- mean((obs - sample_average) ^ 2)
      {
        #LRT with quantiles
        chi_left <- qchisq(p = 1 - alpha / 2, df = n_val - 1)
        chi_right <- qchisq(p = alpha / 2, df = n_val - 1)
        CI_lrtq_left <- n_val * sigma_hat_sq / chi_left
        CI_lrtq_right <- n_val * sigma_hat_sq / chi_right
        length_lrtq <- CI_lrtq_right - CI_lrtq_left
        expect_len_lrtq <- expect_len_lrtq + length_lrtq
        if (CI_lrtq_left <= sigma_sq & CI_lrtq_right >= sigma_sq) {
          coverage_prob_lrtq <- coverage_prob_lrtq + 1
        }
      }
      {
        #wald confidence interval
        #asymptotically mle is N(sigma^2,var(mle)) so,
        #se^2=var(mle)=2sigma^4*(n-1)/n^2 and we have:
        se_hat = sqrt(2 * (n_val - 1)) * sigma_hat_sq / n_val
        CI_wald_left <- sigma_hat_sq - z_wald * se_hat
        CI_wald_right <- sigma_hat_sq + z_wald * se_hat
        length_wald <- CI_wald_right - CI_wald_left
        expect_len_wald <- expect_len_wald + length_wald
        if (CI_wald_left <= sigma_sq & CI_wald_right >= sigma_sq) {
          coverage_prob_wald <- coverage_prob_wald + 1
        }
      }
    }
    expect_len_wald <- expect_len_wald / N
    coverage_prob_wald <- coverage_prob_wald / N
    print(
      sprintf(
        "Wald CI: coverage prob=%f, expected length = %f",
        coverage_prob_wald,
```

```
        expect_len_wald
      )
    )
    expect_len_lrtq <- expect_len_lrtq / N
    coverage_prob_lrtq <- coverage_prob_lrtq / N
    print(
      sprintf(
        "LRT with quantiles CI: coverage prob=%f, expected length = %f",
        coverage_prob_lrtq,
        expect_len_lrtq
      )
    )
  }
}
```

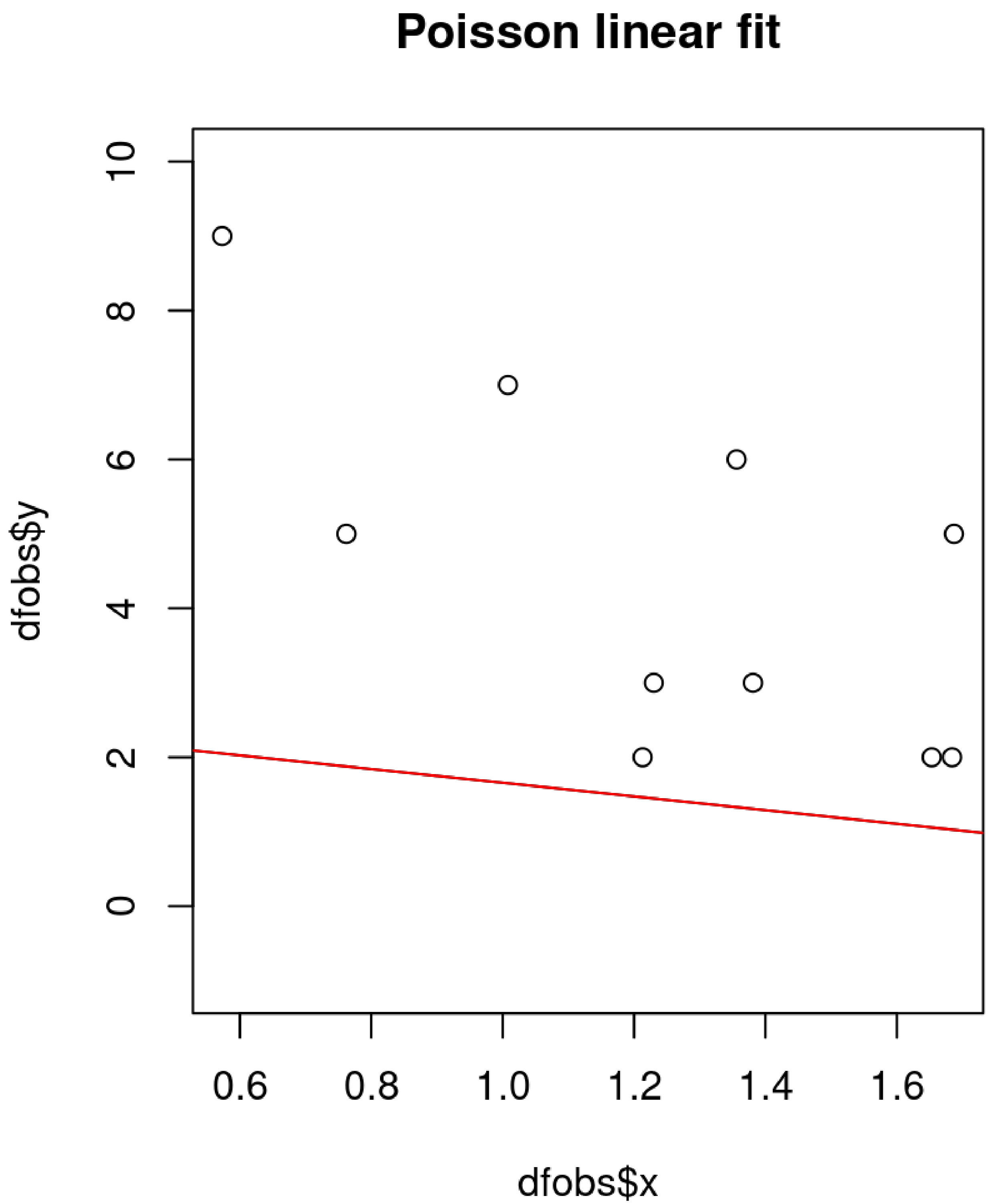
to compute an estimate of the coverage probability and the average length of the CI. We get the following results:

```
[1] "sigma^2 = 0.500000, n of obs = 10, mu = 0.000000"
[1] "Wald CI: coverage prob=0.787800, expected length = 0.744089"
[1] "LRT with quantiles CI: coverage prob=0.950600, expected length = 1.421654"
[1] "-------------------------------------------"
[1] "sigma^2 = 1.000000, n of obs = 10, mu = 0.000000"
[1] "Wald CI: coverage prob=0.795200, expected length = 1.500114"
[1] "LRT with quantiles CI: coverage prob=0.949300, expected length = 2.866113"
[1] "-------------------------------------------"
[1] "sigma^2 = 2.000000, n of obs = 10, mu = 0.000000"
[1] "Wald CI: coverage prob=0.793600, expected length = 2.996621"
[1] "LRT with quantiles CI: coverage prob=0.947600, expected length = 5.725333"
[1] "-------------------------------------------"
[1] "sigma^2 = 0.500000, n of obs = 30, mu = 0.000000"
[1] "Wald CI: coverage prob=0.897600, expected length = 0.483678"
[1] "LRT with quantiles CI: coverage prob=0.948800, expected length = 0.589762"
[1] "-------------------------------------------"
[1] "sigma^2 = 1.000000, n of obs = 30, mu = 0.000000"
[1] "Wald CI: coverage prob=0.892800, expected length = 0.962872"
[1] "LRT with quantiles CI: coverage prob=0.950400, expected length = 1.174057"
[1] "-------------------------------------------"
[1] "sigma^2 = 2.000000, n of obs = 30, mu = 0.000000"
[1] "Wald CI: coverage prob=0.888700, expected length = 1.919253"
[1] "LRT with quantiles CI: coverage prob=0.950300, expected length = 2.340199"
```

and from these results it seems that the LRT CI has an higher coverage probability but also a bigger expected length compared with the Wald CI.

Example 5.6

We can use `optim` defining a log likelihood function to optimize or use `glm`, we get the same result (in the figure both fits are plotted but they overlap). The value of $\hat{\beta}_0 = 2.57698$ and $\hat{\beta}_1 = -0.9196247$.



And we can compute the information matrix in $(\hat{\beta}_0, \hat{\beta}_1)$ and we get

$$\begin{bmatrix} 44.00000 & 49.52698 \\ 49.52698 & 62.12764 \end{bmatrix}$$

(b) For this point we can use `optim` again but this time fixing $\beta_0 = 0$

and we get $-2\log(\lambda(x, y)) = 5.300582$
and the p-value is 0.021318
the script used for points (a) and (b)

```
ll_function <- function(obs_x,obs_y,beta_0,beta_1) {
  an <- function(x,y){
    lambdas<- beta_0+beta_1*x
    return(lambdas*y-exp(lambdas))
  }
  aa <- sum(mapply(an,obs_x,obs_y))-log(prod(sapply(obs_y,factorial)))
  print(aa)
  return(aa)
}
information_function <- function(obs_x,beta_0,beta_1){
  m00 <- sum(exp(beta_0+beta_1*obs_x))
  m10 <- m01 <- sum(obs_x*exp(beta_0+beta_1*obs_x))
  m11 <- sum((obs_x^2)*exp(beta_0+beta_1*obs_x))
  return(rbind(c(m00,m01),c(m10,m11)))
}
obs <-
  rbind(
    c(5, 0.762),
    c(2, 1.213),
    c(5, 1.687),
    c(2, 1.684),
    c(2, 1.653),
    c(3, 1.381),
    c(9, 0.573),
    c(7, 1.008),
    c(6, 1.356),
    c(3, 1.23)
  )
dfobs <- data.frame(y = obs[, 1], x = obs[, 2])
model = glm(formula = y ~ x, family = poisson(), dfobs)
glm_intercept = model$coefficients[1]
glm_slope = model$coefficients[2]


plot(dfobs$x, dfobs$y, ylim = c(-1, 10), main = 'Poisson linear fit')
abline(a = glm_intercept, b = glm_slope)

result = optim(
  par = c(-1,2),
  fn = function(parameters){-ll_function(dfobs$x,dfobs$y, parameters[1], parameters[2])},
  method = "BFGS"
)

optim_intercept = result$par[1]
optim_slope = result$par[2]
```

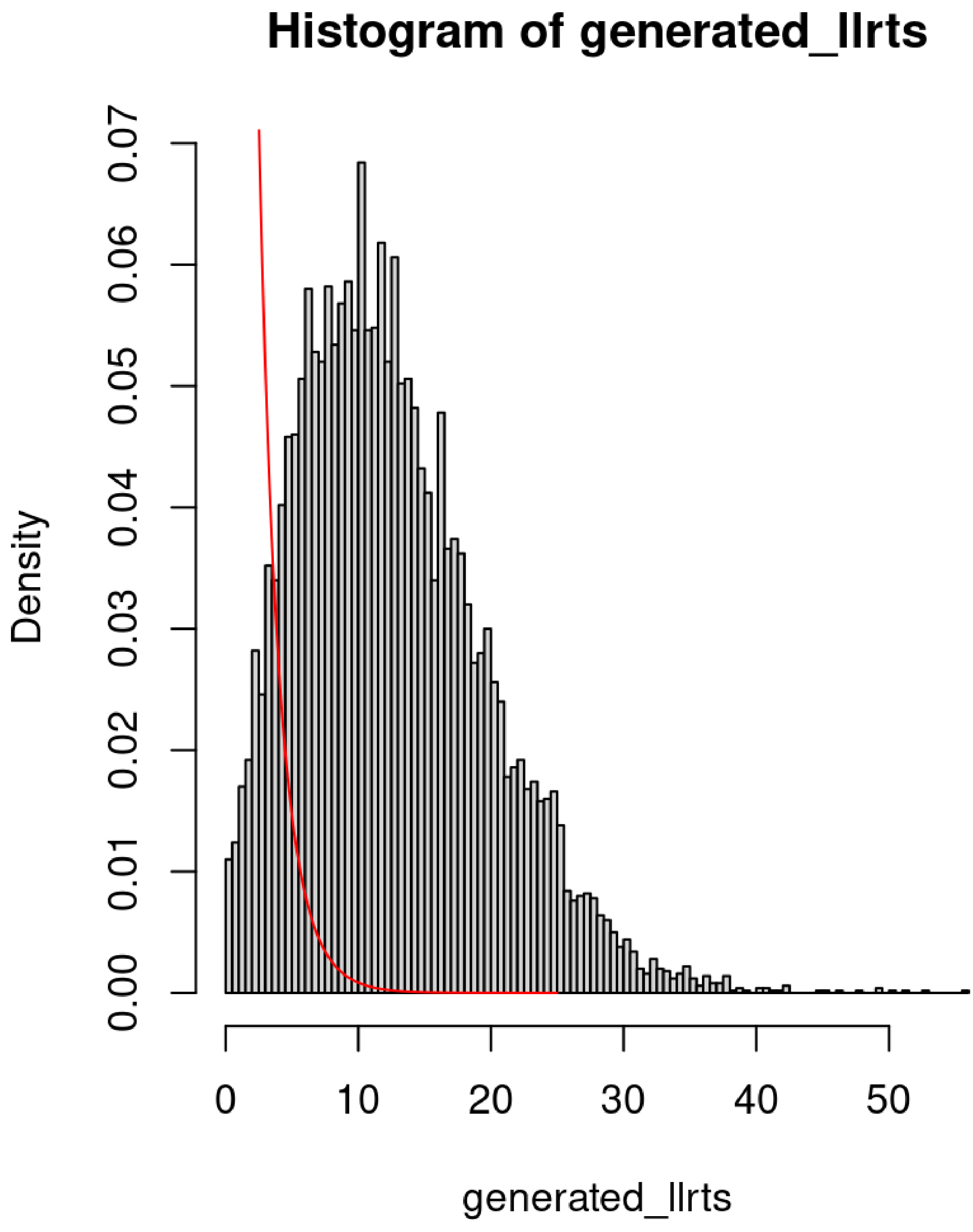
```
abline(a = optim_intercept ,b = optim_slope,col='red')
# the 2 ablines coincide!

print(inf_mat_hat<-information_function(obs_x = dfobs$x,beta_0 = optim_intercept, beta_1 =
optim_slope))
#eigen(-inf_mat_hat)#all eigenvalues are negative, local maximum

#Now we compute the likelihood ratio test for beta_1 = 0 with the observed data
fixed_beta_1 <- 0
result_under_theta0 <-  optim(
  par = c(2),
  fn = function(parameters){-ll_function(dfobs$x,dfobs$y,parameters[1],fixed_beta_1)},
  method = "BFGS"
)
loglrt_theta <- ll_function(dfobs$x,dfobs$y,optim_intercept,optim_slope)
loglrt_theta0 <- ll_function(dfobs$x,dfobs$y,result_under_theta0$par[1],fixed_beta_1)



xx <- -2*(loglrt_theta0-loglrt_theta)
example_alpha <- 1-0.99
threshold <- qchisq(p = 1-example_alpha, df = 1)
print(xx)
print(threshold)
print(sprintf('pvalue:%f',1-pchisq(xx,df=1)))
```

(c) Using the following script:

```
ll_function <- function(obs_x,obs_y,beta_0,beta_1) {
  an <- function(x,y){
    lambdas<- beta_0+beta_1*x
    #print(temp-exp(temp))
    return(lambdas*y-exp(lambdas))
  }
  aa <- sum(mapply(an,obs_x,obs_y))-log(prod(sapply(obs_y,factorial)))
  return(aa)
}

loglrt<- function(){
  beta_0 <- 3
  beta_1 <- -1.5
  obs_x <- c(0.762 ,1.213 ,1.687 ,1.684 ,1.653 ,1.381 ,0.573 ,1.008 ,1.356, 1.230)
  obs_y <- sapply(exp(beta_1 * obs_x+beta_0),FUN = rpois,n=1)
  #print(obs_y)

  result = optim(
    par = c(-1,2),
    fn = function(parameters){-ll_function(obs_x,obs_y, parameters[1], parameters[2])},
    method = "BFGS"
```

```
  )

  optim_intercept = result$par[1]
  optim_slope = result$par[2]

  fixed_beta_1 <- 0
  result_under_theta0 <-  optim(
    par = c(-1),
    fn = function(parameters){-ll_function(obs_x,obs_y,parameters[1],fixed_beta_1)},
    method = "BFGS"
  )
  loglrt_theta <- ll_function(obs_x,obs_y,optim_intercept,optim_slope)
  loglrt_theta0 <- ll_function(obs_x,obs_y,result_under_theta0$par[1],fixed_beta_1)
  xx <- -2*(loglrt_theta0-loglrt_theta)
}
#obs_x <- c(obs_x,obs_x,obs_x,obs_x)
N <- 1e4

generated_llrts <- vector(mode = "double",N)
for (i in 1:N){
  #print(i)
  generated_llrts[i] <- loglrt()
}

hist(generated_llrts,freq = FALSE,nclass = 100,)
ps <- seq(0,25,0.01)
lines(ps,dchisq(x = ps,df = 1),col='red')
```

we get



Example 5.7

(a) point estimate for μ

We can either use the sample average directly which is the MLE for the true mean or use that the MLE is equivariant and the mean of the gamma rv is $\frac{\alpha}{\beta}$.

If we use the first method we get: $\hat{\mu} = 10.233$ with the second method we get $\hat{\mu} = \frac{\hat{\alpha}}{\hat{\beta}} = 10.23299$, to compute MLE of α, β we can use the following R script:

```
boot_time <-
  c(14.87, 7.13, 6.46, 6.45, 9.41, 8.21, 11.18, 14.28, 6.36, 17.98)
ll_function <- function(x, alpha, beta) {
  return(sum(dgamma(
    x,
    shape = alpha,
    rate = beta,
    log = TRUE
  )))
}
result <- optim(
  par = c(0.15, 0.1),
  method = 'BFGS',
  fn = function(params) {
    -ll_function(x = boot_time,
                 alpha = params[1],
                 beta = params[2])
  }
)


mu_hat <- result$par[1] / result$par[2]
```

or using `MASS::fitdistr`:

```
result_MASS <- MASS::fitdistr(x = boot_time, densfun = 'gamma')
print(result$par)
```

and we get the same result in both ways.

(b) To compute the confidence interval for the mean we can use a Wald test.

$\hat{\mu} \dot{\sim} \mathcal{N}(\mu, \frac{\sigma^2}{n})$, $se = \sqrt{\frac{\sigma^2}{n}}$, $\widehat{se} = \sqrt{\frac{\widehat{\sigma}^2}{n}}$

and $W = |\frac{\hat{\mu}-\mu}{\widehat{se}}|$ and the $CI_{W,1-\alpha} = [\hat{\mu} - z_{1\frac{-\alpha}{2}}\widehat{se}, \hat{\mu} + z_{1\frac{-\alpha}{2}}\widehat{se}]$

and we get $CI_{W,0.95} = [7.779215, 12.686771]$, with the following snippet:

```
confidence <- 0.95
n <- length(boot_time)
se_hat <- sqrt(sigma_hat_sq/n)
z_quantile <- qnorm(1-(1-confidence)/2)
left_CI <- mu_hat - z_quantile*se_hat
```

```
right_CI <- mu_hat + z_quantile*se_hat
print(sprintf("wald CI for mu_hat=%f ,[%f,%f]",mu_hat,left_CI,right_CI))
```

(c)
$B \sim \text{Gamma}(\alpha, \beta)$
$P(\text{ boot time greater than 30s}) = P(B > 30) = 1 - P(B \leq 30) = 1 - F_B(30)$
so after we estimated α, β in point (a) we can compute this probability:

```
print(1 - pgamma(
  q = 30,
  shape = result$par[1],
  rate = result$par[2]
))
```

and get 0.0001367082.