TDDE01: Lab 1

A report by

Adam Nyberg
adany869

Assignment 1

This assignment is about classifying emails as spam or not spam by using a K-nearest neighbor classifier. I was given a dataset of 2740 emails with frequency of words and if they are spam or not. The data was split into 50% train and 50% test data.

1.1-2

A K-nearest neighbor classifier requires some kind of distance measure between data points and for this assignment I chose to use the proposed cosine similarity defined as:

$$c(X,Y) = \frac{X^T Y}{\sqrt{\sum_i X_i^2}\sqrt{\sum_i Y_i^2}}$$

$$d(X,Y) = 1 - c(X,Y)$$

With that I computed a matrix D with distances between all emails.

1.3

In this task I classified the test data by using $K = 5$ and the following classification rule.

$$\hat{Y} = 1 \; if \; p(Y = 1|X) > 0.5, otherwise \; \hat{Y} = 0$$

I then compared the classified $\hat{Y}$ to the real Y and got the result in the following table.

	Misclass rate train	Misclass rate test	Confusion matrix train	Confusion matrix test
	0.4474453	0.3175182	pred5Train 0 1 0 615 291 1 322 142	pred5 0 1 0 695 193 1 242 240

1.4

This task was identical to 1.3 but with $K = 1$ and that gave me the following results.

	Misclass rate train	Misclass rate test	Confusion matrix train	Confusion matrix test
	0.4321168	0.3474453	pred1Train 0 1 0 643 298 1 294 135	pred1 0 1 0 639 178 1 298 255

1.5

In this task I used the standard classifier kknn() with K=5 from package kknn and got a misclassification rate of 0.3437956. Below is the confusion matrix.

	0	1
0	647	181
1	290	252

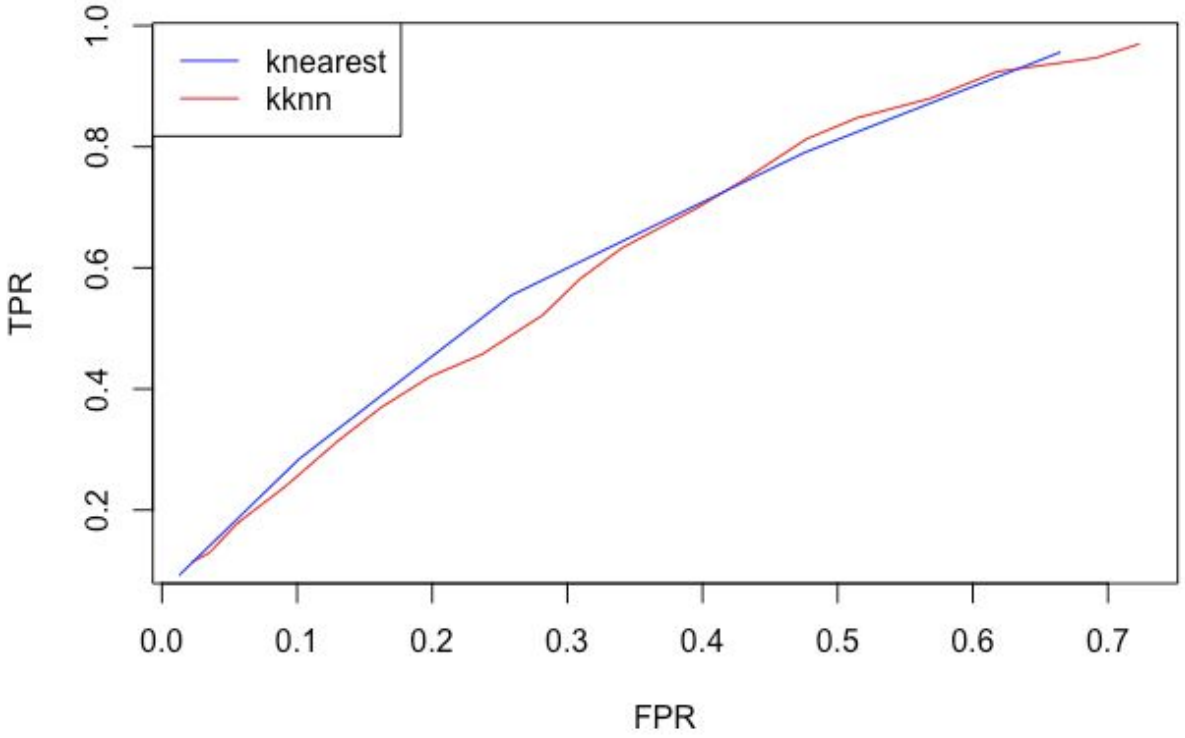
When comparing the results from 1.3-1.5 we can see that the lowest misclassification rate was from 1.3.

1.6

In this task I compared the results of using different calssification principles and then plotted the corresponding ROC Curves. $K = 5$ and both my classifier from 1.1 ($knearest()$) and the classifier from package kknn ($kknn()$) was used. The classification principles was the following.

$$\hat{Y} = 1 \; if \; p(Y = 1|X) > \pi, otherwise \; \hat{Y} = 0$$

Where $\pi = 0.05, 0.1, 0.15, \; ... \; 0.9, 0.95$. That gave me the following ROC Curve.



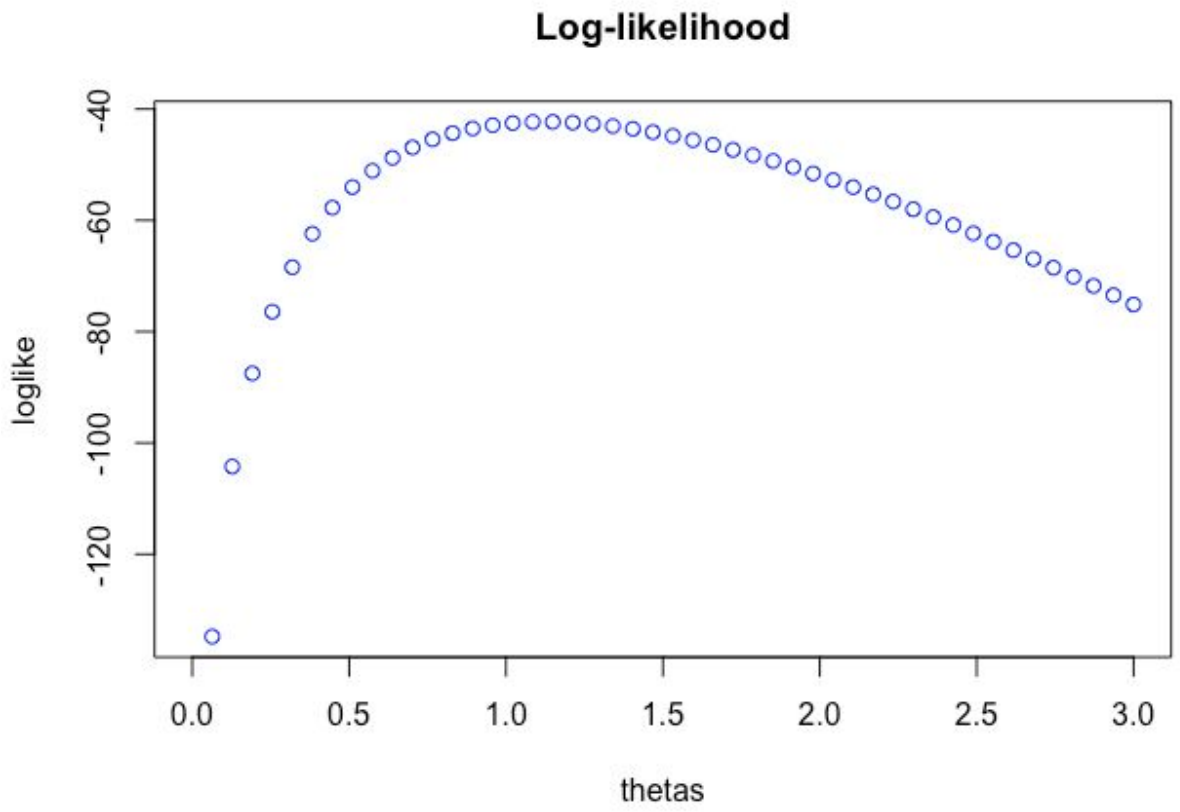
As we can see both knearest and kknn are pretty bad at classifying this data.

Assignment 2

In this assignment I worked with data describing the lifetime of of machines for the purpose of determine warranty time.

2.2

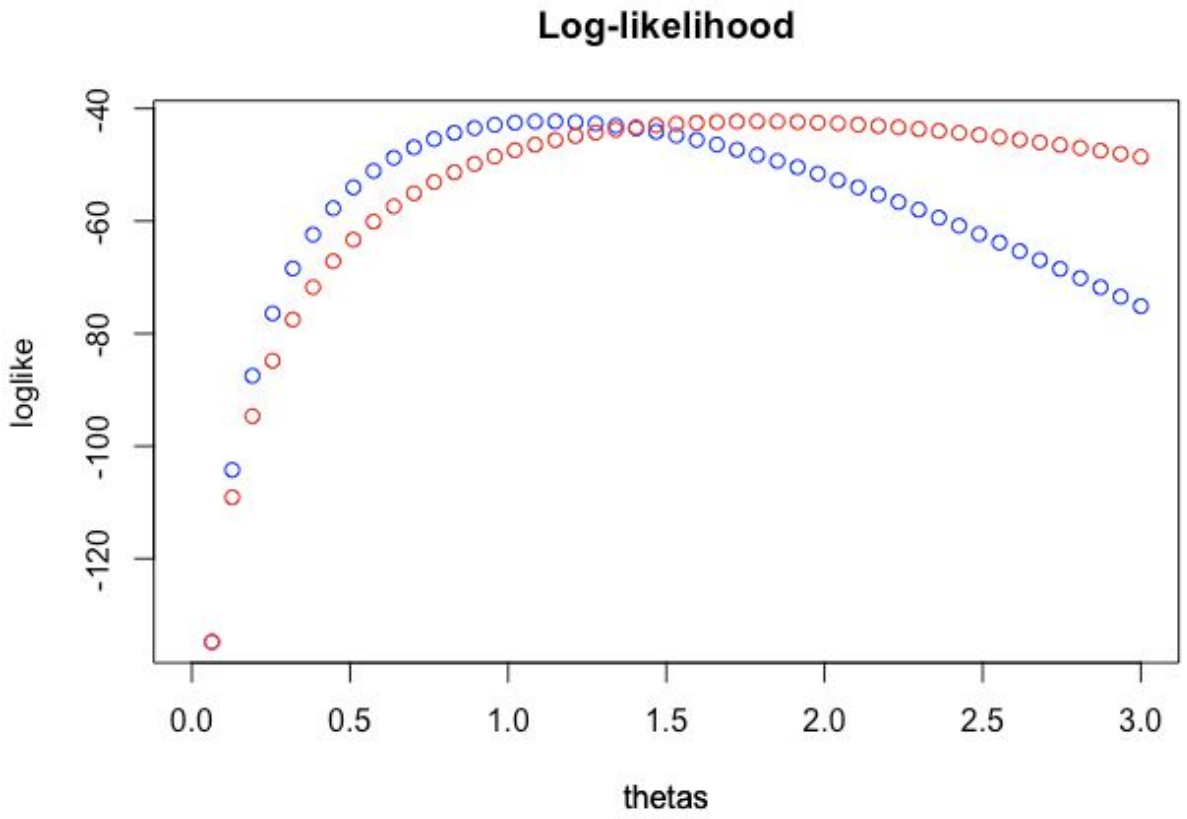
I was given the assumption of an exponential probability model $p(x|\theta) = \theta e^{-\theta x}$ which is an exponential distribution. The function for loglikelihood is $l = n * log(\theta) - \theta * sum(x)$. I the computed the log likelihood of p an made a plot showing the dependence on θ.



The calculated maximum likelihood value of θ was 1.148936.

2.3

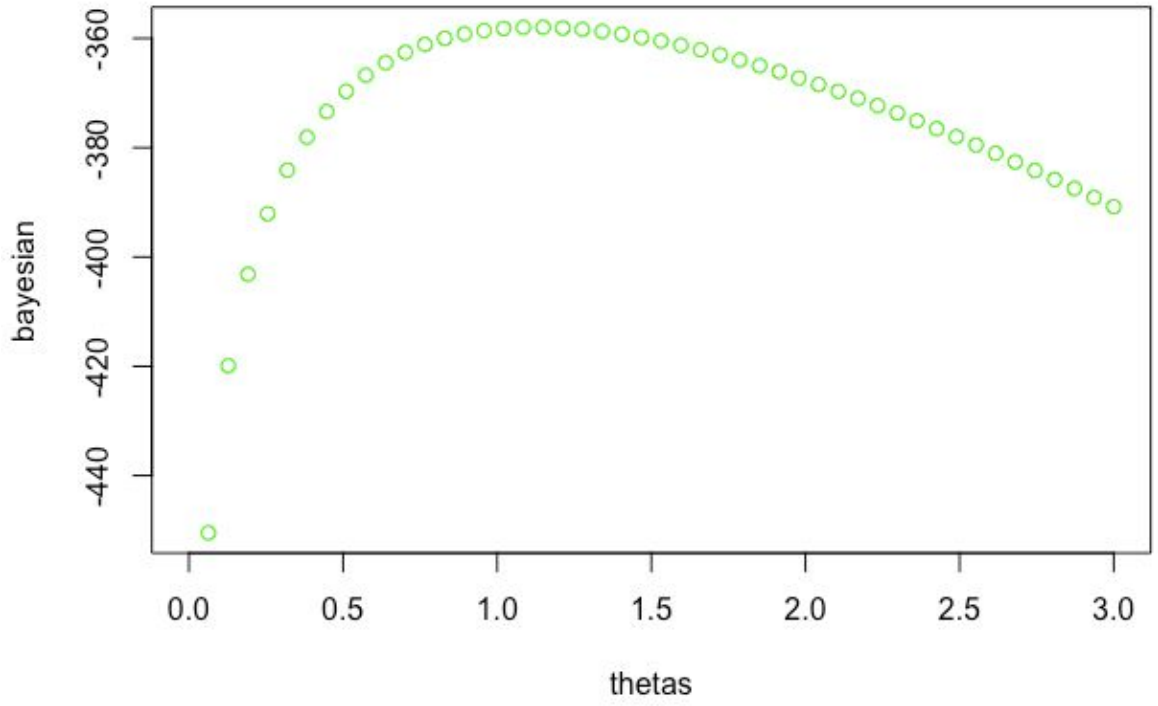
In this task I compared the results from the previous task with only using data from the six first observations about the lifetime. This resulted in the following plot, blue is when all data is used and red is for only six entries.



As we can see now the maximum likelihood value of θ was 1.787234 when only six entries was used. Reliability increases with the data size so in this case the blue data points should be more reliably than the red data points.

2.4

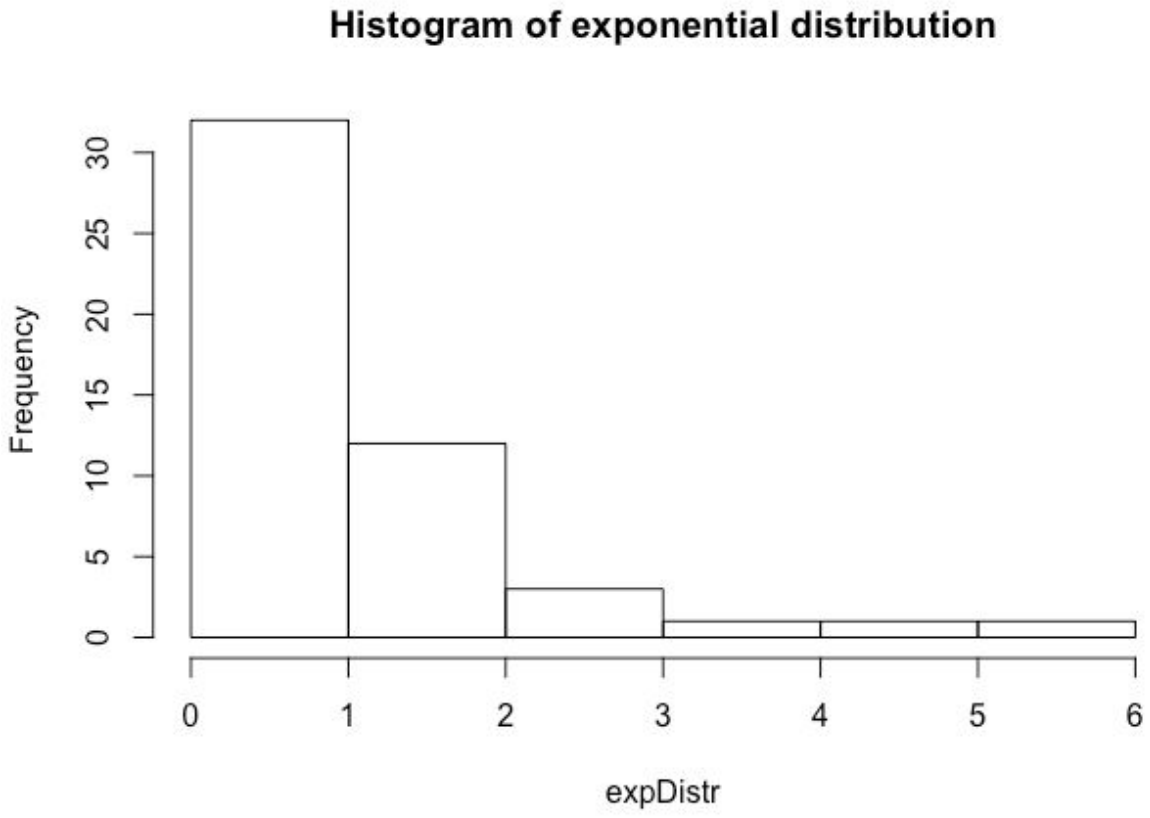
I now used a bayesian model $p(x|\theta) = \theta e^{-\theta x}$ with a prior $p(\theta) = \lambda e^{-\lambda\theta}$ where $\lambda = 10$. I programmed a function $l(\theta) = log(p(x|\theta)p(\theta))$ and plotted l s dependence on θ. That gave me the following plot.

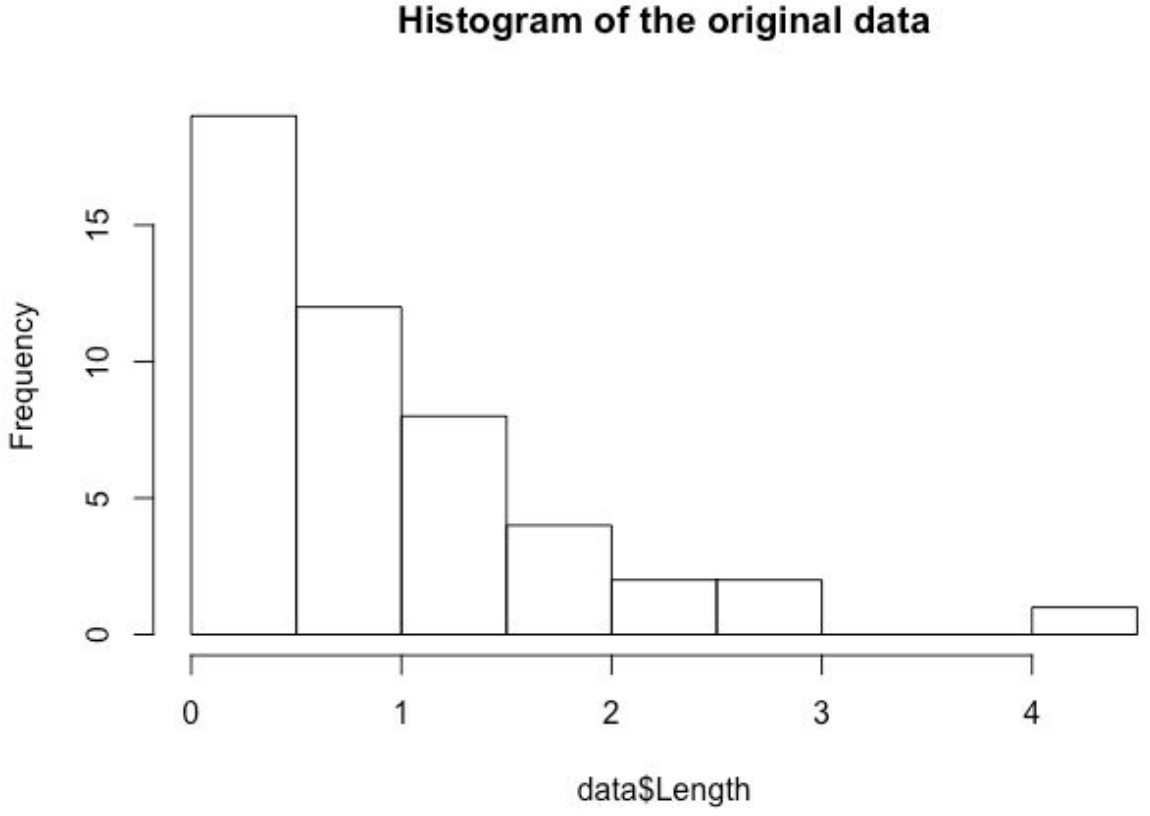


This is very reasonably because the bayesian model and the approach in 1.2 is two different ways to solve the same problem and both gave maximum likelihood value $\theta = 1.148936$.

2.5

In this task I created a histogram from my model in 1.2 and compared that to a histogram of the real data. The result is very identical which implies that the model used is a suitable model.


Histogram of exponential distribution
Frequency
0
5
10
15
20
25
30
0
1
2
3
4
5
6
expDistr


Histogram of the original data
Frequency
0
5
10
15
0
1
2
3
4
data$Length

```
library(kknn)
setwd("~/code/skola/tdde01/adam")

data = read.csv("lab1/spambase.csv")

n=dim(data)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.5))
train=data[id,]
test=data[-id,]

misclass = function(pred, actual) {
  return(mean(pred != actual))
}

classify = function(pred, const=0.5) {
  classified = c()
  for(i in 1:length(pred)) {
    classified[i] = if(pred[i] > const) 1 else 0
  }
  return(classified)
}

distance = function(X,Y) {
  xhat = as.matrix(X)/sqrt(rowSums(X^2))
  yhat = as.matrix(Y)/sqrt(rowSums(Y^2))
  return(1 - xhat%*%t(yhat))
}

knearest = function(data, K, newData, const=0.5) {
  dataSpam = data$Spam
  data$Spam = NULL
  newDataSpam = newData$Spam
  newData$Spam = NULL

  D = distance(data, newData)

  YpredList = c()

  for(i in 1:length(D[1,])) {
    closest = order(D[,i])[1:K]
    Ypred = 0

    for(j in closest) {
      if(dataSpam[j] == 1) Ypred = Ypred+1
    }
```

```
    YpredList[i] = if((Ypred/K) > const) 1 else 0
  }
  return(YpredList)
}



# 1.3-4
pred1 = knearest(train, 1, test)
pred5 = knearest(train, 5, test)

Conf1 = table(pred1, test$Spam)
Mis1 = misclass(pred1, test$Spam)

Conf5 = table(pred5, test$Spam)
Mis5 = misclass(pred5, test$Spam)

# 1.5
kknnPred5 = kknn(Spam~., train, test, 5)
kknnConf5 = table(classify(kknnPred5$fitted.values), test$Spam)
kknnMis5 = misclass(classify(kknnPred5$fitted.values), test$Spam)

# 1.6
tpr=c()
fpr=c()
kktpr=c()
kkfpr=c()

for(bias in seq(0.05, 0.95, 0.05)){
  i = bias*20

  nConf = table(knearest(train, 5, test, bias), test$Spam)
  tpr[i] = nConf[2,2]/sum(nConf[,2])
  fpr[i] = nConf[2,1]/sum(nConf[,1])

  kknnConf = table(classify(kknn(Spam~., train, test, 5)$fitted.values, bias), test$Spam)
  kktpr[i] = kknnConf[2,2]/sum(kknnConf[,2])
  kkfpr[i] = kknnConf[2,1]/sum(kknnConf[,1])
}

plot(kkfpr, kktpr, type="l", col="red", xlab = "FPR", ylab = "TPR")
lines(fpr, tpr, col="blue")
legend("topleft", lty=c(1,1), col=c("blue", "red"), legend = c("knearest", "kknn"))
```

```
setwd("~/code/skola/tdde01/adam")

data = read.csv("lab1/machines.csv")
set.seed(12345)

# 2.2
# the distrubation type is exponatial
L = function(theta, X) {
  return(apply(theta*exp(-theta%*%t(X)), 1, prod))
}

thetas = seq(0,3, length.out=length(data$Length))

loglike = log(L(thetas, data$Length))
plot(thetas, loglike, col='blue', main="Log-likelihood")
thetasMax = thetas[which(loglike==max(loglike))]

# 2.3
loglike6 = log(L(thetas, data$Length[1:6]))
plot(thetas, loglike, col='blue', main="Log-likelihood")
par(new = TRUE)
plot(thetas, loglike6, col='red', ylab = "", axes = FALSE)
thetasMax6 = thetas[which(loglike6==max(loglike6))]

# 2.4
bl = function(theta, X) {
  return(L(theta, X)*as.vector(L(10, X)))
}

bayesian = log(bl(thetas, data$Length))
plot(thetas, bayesian, col='green')
thetaMaxB = thetas[which(bayesian==max(bayesian))]

# 2.5

expDistr = rexp(50, thetasMax)
hist(expDistr, main="Histogram of exponential distribution")
hist(data$Length, main="Histogram of the original data")
```