Sardana_Module5-HW_files

2023-02-13

Problem 1 Maximum Likelihood estimation

1 a) Numerical Optimization given by Numerical_me 1 b) Analytical estimation given by Analytic_me

```
# Maximum likelihood function
likelihood <- function(lam) prod(dexp(c(1.433, 0.524, 0.384, 4.515, 1.852, 0.429), rate = lam))

# Negative log-likelihood function
neglik <- function(lam)-sum(log(dexp(c(1.433, 0.524, 0.384, 4.515, 1.852, 0.429), rate = lam)))

# Maximum likelihood estimate numerical
Numerical_me <- optim(par=1,neglik)
```

```
## Warning in optim(par = 1, neglik): one-dimensional optimization by Nelder-Mead is unreliable:
## use "Brent" or optimize() directly
```

```
Numerical_me$par
```

```
## [1] 0.6566406
```

```
# Analytical maximum likelihood estimate
observations <- c(1.433, 0.524, 0.384, 4.515, 1.852, 0.429)
Analytic_me = 1/mean(observations)
Analytic_me
```

```
## [1] 0.6566707
```

Problem 2

a) According to chi square distribution If the population mean is m for a random sample X1,X2,…Xn, point estimator of m is 98.6
b) One sided 90% confidence interval Chi square distribution

```
# sample size = 95
n <- 75
# degree of freedom = n-1
df = n-1
# at low confidence interval 1-alpha = 0.90 , alpha = 0.01 and at alpha/2 = 0.05
# calculating chisquare at low confidence interval
qchisq(c(.05),df=74, lower.tail=FALSE)
```

```
## [1] 95.08147
```

Problem3

a)

```
# Load multtest package
library(multtest)
```

```
## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
data(golub)
grep("Zyxin",golub.gnames[,2])
```

```
## [1] 2124
```

```
# Extracting the Zyxin gene data from golub at 2124 row
Zyxin <- golub[2124,]
# split the gene factor into 2 ALL and AML group
gene_factor <- factor(golub.cl,levels = 0:1,labels = c("ALL","AML"))

All_exp <- Zyxin[gene_factor == "ALL"]
AML_exp <- Zyxin[gene_factor == "AML"]

# To calculate the the bootstrap 95% CIs for the mean and variance of the ALL group
n <- length(All_exp)
nboot <- 1000
boot.xbar <- rep(NA,nboot)
boot.var <- rep(NA,nboot)
```

```r
for (i in 1:nboot){
  data.star <- All_exp[sample(1:n,replace = TRUE)]
  boot.xbar[i] <- mean(data.star)
  boot.var[i]<- var(data.star)
}

Mean_allCI <- quantile(boot.xbar, c(0.025, 0.975))
Var_allCI <- quantile(boot.var, c(0.025, 0.975))

# Print the mean and variance of ALL group
print(sprintf("At bootstrap 95%% CIs , the mean of the ALL group: [%.3f, %.3f]", Mean_allCI[1], Mean_al
```

```
## [1] "At bootstrap 95% CIs , the mean of the ALL group: [-0.575, -0.028]"
```

```r
print(sprintf("At bootstrap 95%% CIs , the variance of the ALL group: [%.3f, %.3f]", Var_allCI[1], Var_a
```

```
## [1] "At bootstrap 95% CIs , the variance of the ALL group: [0.344, 0.642]"
```

```r
# To calculate the the bootstrap 95% CIs for the mean and variance of the AML group

n <- length(AML_exp)
nboot <- 1000
boot.xbar <- rep(NA, nboot)
boot.var <- rep(NA, nboot)

for (i in 1:nboot) {
  data.star <- AML_exp[sample(1:n, replace=TRUE)]
  boot.xbar[i] <- mean(data.star)
  boot.var[i] <- var(data.star)
}

Mean_amlCI <- quantile(boot.xbar, c(0.025, 0.975))
Var_amlCI <- quantile(boot.var, c(0.025, 0.975))

# Print the mean and variance bootstrap CIs for the AML group
print(sprintf("At bootstrap 95%% CIs , the mean of the AML group: [%.3f, %.3f]", Mean_amlCI[1], Mean_am
```

```
## [1] "At bootstrap 95% CIs , the mean of the AML group: [1.376, 1.789]"
```

```r
print(sprintf("At bootstrap 95%% CIs , the variance of the AML group: [%.3f,%.3f]", Var_amlCI[1], Var_am
```

```
## [1] "At bootstrap 95% CIs , the variance of the AML group: [0.049,0.203]"
```

Problem set 3

(b) to find the parametric 95% CIs for the mean and for the variance of the gene # For AML group

```r
mean_aml <- mean(AML_exp)
mean_aml_SE <- sd(AML_exp) / sqrt(length(AML_exp))
mean_aml_CI <- mean_aml + qnorm(c(0.025, 0.975)) * mean_aml_SE

# For the ALL group
mean_all <- mean(All_exp)
var_all <- var(All_exp)
df_all <- length(All_exp) - 1
var_all_CI <- qchisq(c(0.025, 0.975), df = df_all) * var_all / df_all

# For the AML group
var_aml <- var(AML_exp)
df_aml <- length(AML_exp) - 1
var_aml_CI <- qchisq(c(0.025, 0.975), df = df_aml) * var_aml / df_aml

# Results calculating the mean and variance of all and aml groups

#cat("ALL group:")
#cat("Mean:", mean_all, "CI:", mean_all_CI, "\n")
#cat("Variance:", var_all, "CI:", var_all_CI, "\n")

#cat("AML group:")
#cat("Mean:", mean_aml, "CI:", mean_aml_CI, "\n")
#cat("Variance:", var_aml, "CI:", var_aml_CI, "\n")
```

Problem 3 set

(c) To Find the bootstrap 95% CI for the median gene expression in aml and all groups separately.

```r
# Bootstrap median for ALL group
Zyxin<-golub[2124,]
n<-length(Zyxin)
nboot <- 1000
median_all <- numeric(nboot)
for (i in 1:nboot) {
  boot_all <- All_exp[sample(1:length(All_exp), replace = TRUE)]
  median_all[i] <- median(boot_all)
}

median_all_CI <- quantile(median_all, c(0.025, 0.975))

# Bootstrap median for AML group

median_aml <- numeric(nboot)
for (i in 1:nboot) {
  boot_aml <- AML_exp[sample(1:length(AML_exp), replace = TRUE)]
  median_aml[i] <- median(boot_aml)
}

median_aml_CI <- quantile(median_aml, c(0.025, 0.975))

cat(" The median gene expression for ALL group is:", median_all_CI, "\n")
```

```
##  The median gene expression for ALL group is: -0.73507 0.31432
```

```
cat("The median gene expression for AML group is:", median_aml_CI, "\n")
```

```
## The median gene expression for AML group is: 1.22814 1.82829
```

Problem set 4

(a) and (b)

```
nsim<-1000
MCsim<- function(nsim, lambda){
  cov1<-cov2<-rep(NA,nsim)
  for (i in 1:nsim){
    x = rpois(50,lambda)
    xbar = mean(x)
    Xsd = sd(x)
    CI1<-c(xbar+(qt(0.05, 49)*sqrt(xbar/50)), xbar+qt(0.95, 49)*sqrt(xbar/50))
    CI2<-c((49*(Xsd^2)/qchisq(0.95, 49)), 49*(Xsd^2)/qchisq(0.05, 49))
    cov1[i]<-(CI1[1]<lambda)&(lambda<CI1[2])
    cov2[i]<-(CI2[1]<lambda)&(lambda<CI2[2])
    }
  print(paste("When lambda=", lambda, ": coverage for first CI is", mean(cov1), ", coverage for second C
  }

# Used monte carlo simulations for nsim = 1000 runs at different parameter values
MCsim(nsim = 1000, lambda = 0.1)
```

```
## [1] "When lambda= 0.1 : coverage for first CI is 0.898 , coverage for second CI is 0.57 ."
```

```
MCsim(nsim = 1000, lambda = 1)
```

```
## [1] "When lambda= 1 : coverage for first CI is 0.904 , coverage for second CI is 0.823 ."
```

```
MCsim(nsim = 1000, lambda = 10)
```

```
## [1] "When lambda= 10 : coverage for first CI is 0.918 , coverage for second CI is 0.883 ."
```