Sardana_Module9-HW

Prachi Sardana

2023-03-20

Question 1

```
# Loaded golub data set
data(golub, package='multtest')
# a) GRO2 gene expression
GRO2_gene <- grep("GRO2", golub.gnames[, 2])
GRO2_gene
```

```
## [1] 2714
```

```
# GRO3 gene expression
GRO3_gene <- grep("GRO3", golub.gnames[,2])
GRO3_gene
```

```
## [1] 2715
```

```
x <- golub[2714,]
y <- golub[2715,]
# correlation of x and y
cor(x,y)
```

```
## [1] 0.7966283
```

```
# b)
# parametric 90% confident interval for the correlation with cor.test()
cor.test(x,y,
         alternative = "greater",
         method = "pearson",
         exact = NULL, conf.level = 0.90, continuity = FALSE)
```

```
##
##  Pearson's product-moment correlation
##
## data:  x and y
## t = 7.9074, df = 36, p-value = 1.101e-09
## alternative hypothesis: true correlation is greater than 0
## 90 percent confidence interval:
##  0.7027404 1.0000000
## sample estimates:
##       cor
## 0.7966283
```

```r
# c)

# bootstrap 90% confident interval for the correlation

nboot <- 2000 # resample 2000 times
boot.cor<-matrix(0,nrow=nboot,ncol=1) # vector to have resampled statistics
data<- cbind(x,y) # Data set with x and y two coloumns
for (i in 1:nboot){
  dat.star<-data[sample(1:nrow(data), replace=TRUE), ] # resample the pairs
  boot.cor[i,] <- cor(dat.star[,1], dat.star[,2]) # correlation on resampled data
}
# Find quantiles for resampled statistics
quantile(boot.cor[,1], c(0.05,0.90)) # bootstrap 90% interval
```

```
##        5%       90%
## 0.6004499 0.8779893
```

Question 2

```r
# a)
# total genes having correlation values less than negative 0.5
data(golub)
```

```
## Warning in data(golub): data set 'golub' not found
```

```r
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")'.
```

```r
# golub.gnames[2124,] zyxin gene expression

corr <- apply(golub,1,cor, as.numeric( golub[2124,] ))
num_corr <- sum(corr < -0.5)
print(paste("Number of genes with correlation less than -0.5:", num_corr))
```

```
## [1] "Number of genes with correlation less than -0.5: 85"
```

```r
# b )
# gene names for the top five genes that are most negatively correlated with Zyxin gene.


order_corr <- order(corr)
top_five_genes <- golub.gnames[order_corr,][1:5,2]
print(paste("Top five genes which are negatively correlated are",top_five_genes))
```

```
## [1] "Top five genes which are negatively correlated are Macmarcks"
## [2] "Top five genes which are negatively correlated are Inducible protein mRNA"
## [3] "Top five genes which are negatively correlated are C-myb gene extracted from Human (c-myb) gene
## [4] "Top five genes which are negatively correlated are Oncoprotein 18 (Op18) gene"
## [5] "Top five genes which are negatively correlated are 54 kDa protein mRNA"
```

```r
# c)

# Using the correlation test, total genes are negatively correlated with the Zyxin gene with fdr of 0.0

cor_zyxin <- apply(golub,1,function(x) cor.test(x, as.numeric(golub[2124,]),
alternative = "less")$p.value)
p_fdr <- p.adjust(cor_zyxin,method = "fdr")
sum(p_fdr < 0.05)
```

```
## [1] 142
```

Question 3

```r
# a)

data(golub)
library(multtest)

GRO2_gene <- golub[2714,]
GRO3_gene <- golub[2715,]

reg.fit <- lm(GRO3_gene ~ GRO2_gene)
summary(reg.fit)
```

```
##
## Call:
## lm(formula = GRO3_gene ~ GRO2_gene)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78038 -0.10639 -0.00553  0.14225  0.96298
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.84256    0.05941 -14.182 2.62e-16 ***
## GRO2_gene    0.35820    0.04530   7.907 2.20e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3201 on 36 degrees of freedom
## Multiple R-squared:  0.6346, Adjusted R-squared:  0.6245
## F-statistic: 62.53 on 1 and 36 DF,  p-value: 2.201e-09
```

```
cor.test(GRO3_gene,GRO2_gene)
```

```
##
##  Pearson's product-moment correlation
##
## data:  GRO3_gene and GRO2_gene
## t = 7.9074, df = 36, p-value = 2.201e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6399101 0.8897262
## sample estimates:
##       cor
## 0.7966283
```

```
# Since, the p value is 2.201e-09, the Pearson's product-moment correlation test indicates that there i

# b)
predict(reg.fit, newdata=data.frame(GRO2_gene=0), interval="prediction",level = 0.80)
```

```
##         fit       lwr        upr
## 1 -0.842559 -1.267563 -0.4175553
```

```
# c)
shapiro.test(resid(reg.fit))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(reg.fit)
## W = 0.94779, p-value = 0.07532
```

```
# Since the p value is greater than 0.05 , we can't reject the null hypothesis
```

Question 4

```r
# a
# Loaded the data stackloss
data("stackloss")

# Regress stack.loss on the other three variables.
lin.reg<-lm(stack.loss~Air.Flow+Water.Temp+Acid.Conc.,data = stackloss)
summary(lin.reg)
```

```
##
## Call:
## lm(formula = stack.loss ~ Air.Flow + Water.Temp + Acid.Conc.,
##     data = stackloss)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2377 -1.7117 -0.4551  2.3614  5.6978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -39.9197    11.8960  -3.356  0.00375 **
## Air.Flow      0.7156     0.1349   5.307  5.8e-05 ***
## Water.Temp    1.2953     0.3680   3.520  0.00263 **
## Acid.Conc.   -0.1521     0.1563  -0.973  0.34405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.243 on 17 degrees of freedom
## Multiple R-squared:  0.9136, Adjusted R-squared:  0.8983
## F-statistic:  59.9 on 3 and 17 DF,  p-value: 3.016e-09
```

```r
lm(formula = stack.loss~Air.Flow+Water.Temp+Acid.Conc.,data = stackloss)
```

```
##
## Call:
## lm(formula = stack.loss ~ Air.Flow + Water.Temp + Acid.Conc.,
##     data = stackloss)
##
## Coefficients:
## (Intercept)     Air.Flow   Water.Temp   Acid.Conc.
##    -39.9197       0.7156       1.2953      -0.1521
```

```r
# The fitted regression equation is as follows: Stack.Loss = -39.92 +0.72Air.FLow +1.30Water.Temp -0.15
```

Question 4 # b No, none of the factors have a statistically significant impact on stack loss.Although air flow and water temps do,yet,acid concentration does not. Together, the variables account for 90% of the overall variation in stack loss.

Question 4 # c

```r
# Loaded the data stackloss
data("stackloss")
```

```r
# Regress stack.loss on the other three variables
lin.reg<-lm(stack.loss~Air.Flow+Water.Temp+Acid.Conc.,data = stackloss)

# Find a 90% confidence interval
predict(lin.reg, data.frame(Air.Flow=60,Water.Temp=20,Acid.Conc.=90),interval= "confidence", level = 0.9
```

```
##        fit      lwr      upr
## 1 15.23343 13.50069 16.96617
```

```r
# Find 90% prediction interval for stack.loss
predict(lin.reg, data.frame(Air.Flow=60,Water.Temp=20,Acid.Conc.=90),interval= "prediction", level = 0.9
```

```
##        fit      lwr      upr
## 1 15.23343 9.331184 21.13568
```