

TMA4267 – Linear Statistical Models

Solutions to Exercise 4 – V2014

Problem 1: F test and partial F test in multiple linear regression

a) We already know from the theory that $I - H$ and H_0 are idempotent. $(H - H_0)^2 = H^2 - HH_0 - H_0H + H_0^2 = H - HH_0 - H_0H + H_0$. Since H_0 is a projection onto the column space of X_0 , a further projection by H to the column space of X will do nothing further, and $HH_0 = H_0$. Also, $I - H$ projects onto a subspace orthogonal to the column space of X , so a further projection by H_0 gives $H_0(I - H) = O$, and $H_0H = H_0$. So $(H - H_0)^2 = H - H_0$.

b) The first identity is true in general since $(I - H)X = O$. For the second identity, under the null hypothesis $X\beta$ is in the column space of X_0 , so that $HX\beta = H_0X\beta = X\beta$, and $(H - H_0)X\beta = O$.

c) We know from the theory that $\text{rank}(I - H) = n - p$, $\text{rank } H = p$ and that $\text{rank } H_0 = r$. Then $\text{rank}(H - H_0) = \text{tr}(H - H_0) = \text{tr } H - \text{tr } H_0 = p - r$. The last statement follows from Theorem 3.27 of B/F and the result about the ratio of two independent chi-square distributed variables divided by their number of degrees of freedom.

d) Code from <http://www.math.ntnu.no/~bakke/TMA4267/2015V/acidrain-exerc4.r>:

```
## Exercise set 4 2015V:
# How does pH in Nowegian lakes depend on sulfate, nitrate, calcium, aluminium
# and organic content (x1, ... x5), area of lake (x6) and location (x7 = 0,
# Telemark, or x7 = 1, Trøndelag)? Data from Statens forurensningstilsyn
# (1986). Here 26 random lakes from Telemark and Trøndelag out of 1005 lakes
# have been drawn.

acidrain <-
  read.table("http://www.math.ntnu.no/~mettela/TMA4267/Data/acidrain.txt",
    header=TRUE)

attach(acidrain)

n <- length(y)

x <- cbind(rep(1,n),acidrain[,2:8]) # we want intercept in model
names(x)[1] <- 1
```

```

x <- as.matrix(x)
p <- dim(x)[2]

i <- diag(n)
h <- x%%solve(t(x)%%x)%%t(x)

# Test of submodel where coefficients of x2, x4, x5, x6, x7 are zero:
r <- 3
x0 <- x[,c(1,2,4)]
h0 <- x0%%solve(t(x0)%%x0)%%t(x0)
f <- t(y)%%(h-h0)%%y/(p-r)/(t(y)%%(i-h)%%y/(n-p))
f
pf(f,p-r,n-p,lower.tail=FALSE) # p-value - cannot reject null hypothesis
# or by R functions (1 is added by R as a covariate):
fit <- lm(y~x1+x2+x3+x4+x5+x6+x7)
fit0 <- lm(y~x1+x3)
anova(fit0,fit)

# Test of whether all coefficients except the intercept are zero
r <- 1
x0 <- x[,1]
h0 <- x0%%solve(t(x0)%%x0)%%t(x0)
f <- t(y)%%(h-h0)%%y/(p-r)/(t(y)%%(i-h)%%y/(n-p))
f
pf(f,p-r,n-p,lower.tail=FALSE) # p-value - cannot reject null hypothesis
# or by R functions:
fit0 <- lm(y~1)
anova(fit0,fit)
# or:
fit2 <- lm(y~x)
anova(fit2)

```

e) In this case $X_0 = \mathbf{1}$, so that $H_0 = X_0(X_0^T X_0)^{-1} X_0^T = \mathbf{1}(\mathbf{1}^T \mathbf{1})^{-1} \mathbf{1}^T = \mathbf{1}n^{-1} \mathbf{1}^T = \frac{1}{n} \mathbf{1} \mathbf{1}^T$. R code is given above.

Problem 2: Inference about a new observation in multiple linear regression

a) $\hat{\beta} \sim N(\beta, \sigma^2(X^T X)^{-1})$, so that the linear transformation $\mathbf{X}_0 \hat{\beta}$ of $\hat{\beta}$ has the distribution $N(\mathbf{X}_0^T \beta, \sigma^2 \mathbf{X}_0 (X^T X)^{-1} \mathbf{X}_0^T)$, which is univariate. So its expected value is the EY_0 specified by the model, and is thus unbiased.

b) $\text{SSE}/\sigma^2 \sim \chi^2(n-p)$, where n is the length of $\mathbf{Y}$ and p is the length of β , and it is independent of $\hat{\beta}$ and thus of $\mathbf{X}_0^T \hat{\beta}$. Standardize $\mathbf{X}_0 \hat{\beta}$ and divide it by $\sqrt{\text{SSE}/((n-p)\sigma^2)}$ to get

$$T = \frac{\mathbf{X}_0^T \hat{\beta} - EY_0}{\sqrt{\mathbf{X}_0 (X^T X)^{-1} \mathbf{X}_0^T \text{SSE}/(n-p)}} \sim t(n-p).$$

Solve the double inequality $-t_{n-p} \leq T \leq t_{n-p}$, where $-t_{\alpha/2}$ is the $\alpha/2$ -quantile of a $t(n-p)$ variable, to get a confidence interval having bounds

$$\mathbf{X}_0^T \hat{\beta} \pm \sqrt{\mathbf{X}_0 (X^T X)^{-1} \mathbf{X}_0^T \text{SSE} / (n-p)}.$$

c) Since Y_0 is independent of ϵ and thus of $\mathbf{X}_0^T \hat{\beta}$, $Y_0 - \mathbf{X}_0^T \hat{\beta} \sim N(0, \sigma^2(1 + \mathbf{X}_0 (X^T X)^{-1} \mathbf{X}_0^T))$. Standardize $Y_0 - \mathbf{X}_0^T \hat{\beta}$ and proceed as in (b) to get a prediction interval having bounds

$$\mathbf{X}_0^T \hat{\beta} \pm \sqrt{(1 + \mathbf{X}_0 (X^T X)^{-1} \mathbf{X}_0^T) \text{SSE} / (n-p)}.$$

d) Last part of <http://www.math.ntnu.no/~bakke/TMA4267/2015V/acidrain-exerc4.r>:

```
# Confidence interval for EY0 and prediction interval for EY0 with new
# covariate vector
x0 <- c(1,3,50,1,50,2,1,0) # remember intercept (the first 1)
sse <- y%*(i-h)%*y # R automatically transposes vector when necessary
betahat <- solve(t(x)%*x)%*t(x)%*y
# Confidence interval:
halflength <-
  qt(.025,n-p,lower.tail=FALSE)*sqrt(x0%*solve(t(x)%*x)%*x0*sse/(n-p))
sum(x0*betahat)-halflength
sum(x0*betahat)+halflength
# Prediction interval:
halflength2 <-
  qt(.025,n-p,lower.tail=FALSE)*sqrt((1+x0%*solve(t(x)%*x)%*x0*sse/(n-p))
sum(x0*betahat)-halflength2
sum(x0*betahat)+halflength2

# By R functions instead:
newdata <- data.frame(x1=3,x2=50,x3=1,x4=50,x5=2,x6=1,x7=0)
# or the following two lines
newdata <- data.frame(3,50,1,50,2,1,0)
names(newdata) <- names(coefficients(fit)[-1])
# In either case:
predict(fit,newdata,level=.95,interval="confidence")
predict(fit,newdata,level=.95,interval="prediction")
```

e) The design matrix in this case is

$$X = \begin{pmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_n \end{pmatrix}^T,$$

and you can verify that

$$\mathbf{X}_0 (X^T X)^{-1} \mathbf{X}_0^T = \frac{\sum_i (x_i - x_0)^2}{n \sum_i (x_i - \bar{x})^2}.$$

Next, write $x_i - x_0 = (x_i - \bar{x}) + (\bar{x} - x_0)$ to get $\sum_i (x_i - x_0)^2 = \sum_i (x_i - \bar{x})^2 + n(\bar{x} - x_0)^2$ (crossterms vanish). The bounds given follow.

Problem 3: The square root matrix and the Mahalanobis transform

- a) Assume Σ is symmetric and positive definite, and $(\lambda_i, \mathbf{e}_i)$, $i = 1, \dots, p$ are the eigenvalues and eigenvectors of Σ .

A symmetric matrix has real eigenvalues. A positive definite matrix fulfills

$$\mathbf{x}^T \Sigma \mathbf{x} > 0 \text{ for all } \mathbf{x} \neq 0$$

Let $\mathbf{x} = \mathbf{e}_i$ be the i th eigenvector of Σ .

$$\begin{aligned} \mathbf{e}_i^T \Sigma \mathbf{e}_i &= \mathbf{e}_i^T P \Lambda P^T \mathbf{e}_i \\ &= \mathbf{e}_i^T \left(\sum_{i=1}^p \lambda_i \mathbf{e}_i \mathbf{e}_i^T \right) \mathbf{e}_i \\ &= \mathbf{e}_i^T (\lambda_i \mathbf{e}_i \mathbf{e}_i^T) \mathbf{e}_i \\ &= \lambda_i > 0 \end{aligned}$$

so all eigenvalue of a SPD must be positive.

Then the eigenvalues and -vectors of the inverse matrix of Σ ?

First, Σ^{-1} exists since Σ is SPD. Further, for eigenvector/value pair $(\lambda_i, \mathbf{e}_i)$ for Σ

$$\begin{aligned} \Sigma \mathbf{e}_i &= \lambda_i \mathbf{e}_i \\ \Sigma^{-1} \Sigma \mathbf{e}_i &= \lambda_i \Sigma^{-1} \mathbf{e}_i \\ I \mathbf{e}_i &= \lambda_i \Sigma^{-1} \mathbf{e}_i \\ \frac{1}{\lambda_i} \mathbf{e}_i &= \Sigma^{-1} \mathbf{e}_i \end{aligned}$$

so $(1/\lambda_i, \mathbf{e}_i)$ is an eigenvector/value pair for Σ^{-1} .

So, Σ and Σ^{-1} have the same eigenvectors. And, if λ_i is an eigenvalue for Σ then $1/\lambda_i$ is an eigenvalue for Σ^{-1} . All eigenvalues are positive.

- b) Show that $\Sigma^{\frac{1}{2}}$ and $\Sigma^{-\frac{1}{2}}$ are symmetric:

$$\begin{aligned} \Sigma^{\frac{1}{2}} &= P \Lambda^{\frac{1}{2}} P^T \\ (\Sigma^{\frac{1}{2}})^T &= (P \Lambda^{\frac{1}{2}} P^T)^T = P \Lambda^{\frac{1}{2}} P^T = \Sigma^{\frac{1}{2}} \end{aligned}$$

Since transposing a diagonal matrix leaves the matrix unchanged. To prove that $\Sigma^{-\frac{1}{2}}$ is symmetric: substitute $\Sigma^{\frac{1}{2}}$ with $\Sigma^{-\frac{1}{2}}$ in the above equations.

The the following identities:

$$\begin{aligned}
\Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}} &= P \Lambda^{\frac{1}{2}} P^T P \Lambda^{\frac{1}{2}} P^T \\
&= P \Lambda^{\frac{1}{2}} \Lambda^{\frac{1}{2}} P^T \\
&= P \Lambda P^T = \Sigma \\
\Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} &= P \Lambda^{-\frac{1}{2}} P^T P \Lambda^{-\frac{1}{2}} P^T \\
&= P \Lambda^{-\frac{1}{2}} \Lambda^{-\frac{1}{2}} P^T \\
&= P \Lambda^{-1} P^T = \Sigma^{-1} \\
\Sigma^{\frac{1}{2}} \Sigma^{-\frac{1}{2}} &= P \Lambda^{\frac{1}{2}} P^T P \Lambda^{-\frac{1}{2}} P^T \\
&= P \Lambda^{\frac{1}{2}} \Lambda^{-\frac{1}{2}} P^T \\
&= P I P^T = I
\end{aligned}$$

where I is the identity matrix.

c)

$$\begin{aligned}
Y &= \Sigma^{-\frac{1}{2}}(X - \mu) \\
E(Y) &= E(\Sigma^{-\frac{1}{2}}(X - \mu)) = \Sigma^{-\frac{1}{2}}(E(X) - \mu) = \mathbf{0} \\
\text{Cov}(Y) &= \text{Cov}(\Sigma^{-\frac{1}{2}}(X - \mu)) = \Sigma^{-\frac{1}{2}} \text{Cov}(X) (\Sigma^{-\frac{1}{2}})^T \\
&= \Sigma^{-\frac{1}{2}} \Sigma \Sigma^{-\frac{1}{2}} = P \Lambda^{-\frac{1}{2}} P^T P \Lambda P^T P \Lambda^{\frac{1}{2}} P^T \\
&= P \Lambda^{-\frac{1}{2}} \Lambda \Lambda^{\frac{1}{2}} P^T \\
&= P I P^T = I
\end{aligned}$$

d) Now $X \sim N_p(\mu, \Sigma)$ where Σ is SPD, and we are asked to find the distribution of

$$(X - \mu)^T \Sigma^{-1} (X - \mu).$$

There are two main ways of proceeding 1) directly - by looking at the Mahalanobis transform Y as a linear transform, and 2) using the spectral decomposition of Σ^{-1} . We do the first, since that is less time consuming (you might try the other).

$$\begin{aligned}
(X - \mu)^T \Sigma^{-1} (X - \mu) &= (X - \mu)^T \Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} (X - \mu) \\
&= Y^T Y = \sum_{i=1}^P Y_i^2
\end{aligned}$$

using $\Sigma^{-\frac{1}{2}}(X - \mu)$ from c).

We know that Y is multivariate normal since it is a linear transform of X , and we have calculated the mean and covariance in c), so that $Y \sim N_p(\mathbf{0}, I)$. We know that this means that the elements Y_i of Y are independent (covariance zero implies independence for the multivariate normal). Further, we know that Y_i^2 follows a univariate χ_1^2 -distribution. Since the Y_i 's are independent, then the Y_i^2 's are also independent, and $\sum_{i=1}^P Y_i^2 \sim \chi_p^2$. Thus,

$$(X - \mu)^T \Sigma^{-1} (X - \mu) \sim \chi_p^2$$

e) The pdf of $\mathbf{X}$ is given as

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\}$$

Contours of constant density for the p -dimensional normal distribution are ellipsoids defined by $\mathbf{x}$ such that

$$(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) = c^2$$

These ellipsoids are centered at $\boldsymbol{\mu}$ and have axes $\pm c\sqrt{\lambda_i}\mathbf{e}_i$, where $\Sigma\mathbf{e}_i = \lambda_i\mathbf{e}_i$, for $i = 1, \dots, p$.

Further, since $(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu})$ is distributed as χ_p^2 , the volume (or area in two dimensions) inside the ellipsoid of $\mathbf{x}$ values satisfying

$$(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) \leq \chi_p^2(\alpha)$$

has probability $1 - \alpha$.

```
f) > Sigma <- matrix(c(1,0.8,0.8,1),2,2)
> eigen(Sigma)
$values
[1] 1.8 0.2

$vectors
      [,1]      [,2]
[1,] 0.7071068 0.7071068
[2,] 0.7071068 -0.7071068

> sqrt(qchisq(0.1,2,lower.tail=F))
[1] 2.145966
```

Draw contours for which 90% of the probability mass lies within the contours:

This will be a drawing of an ellipse with axes in the direction of the eigenvectors. We see these axes are in the direction (1,1) (primary axis) and (1,-1) (secondary axis) (it is always like this when the variances of the binormal are equal). The primary axis is the one corresponding to the largest eigenvalue. The axis length for the primary axis will be $2.15 \cdot \sqrt{1.8} = 2.34$ and the axis length for the secondary axis will be $2.15 \cdot \sqrt{0.2} = 0.96$.

Problem 4: Sampling multivariate normal data

```
# TMA4267 Exercise 3, Problem 6: Sampling multivariate normal data
# maybe first
#install.packages("MASS")
# and the same for ellipse

library(MASS)
library(ellipse)
library(mvtnorm)
library(rgl)
```

```

# a
muX <- matrix(c(0,0),ncol=1)
SigmaX <- diag(2)
a <- 0.1
A <- matrix(c(1,a,a,1),ncol=2)
muY <- A%*%muX
SigmaY <- A%*%SigmaX%*%t(A)
CorrY <- cov2cor(SigmaY)
# in general SigmaY has elements
#  $1+a^2$  on the diagonal and  $2*a$  off-diagonal
# corr is off-diag  $2a/(1+a^2)$ 

# simulating withmvrnorm
meanvec <- muY
covarmat <- SigmaY
binorm.sample <- mvrnorm(1000,meanvec,covarmat)
x <- binorm.sample[,1]
y <- binorm.sample[,2]
plot(x,y)

avec <- c(-0.9,-1/4,0,1/4,0.9)

# make function with a as input to automate this
#(and you see how to make a function)

sim <- function(a,muX,SigmaX,n=1000)
{
  print(a)
  A <- matrix(c(1,a,a,1),ncol=2)
  muY <- A%*%muX
  SigmaY <- A%*%SigmaX%*%t(A)
  invsd <- diag(1/diag(SigmaY))
  CorrY <- invsd%*%SigmaY%*%invsd
  binorm.sample <- mvrnorm(n,muY,SigmaY)
  x <- binorm.sample[,1]
  y <- binorm.sample[,2]

  par(mfrow=c(3,2),oma=c(1,2,1,1))
  #outer margins top is 2
  plot(x,y)
  boxplot(binorm.sample)
  hist(x,nclass=10)
  hist(y,nclass=10)
  qqnorm(x); qqline(x);
  qqnorm(y); qqline(y)
  mtext(paste("a=",a),side=3,line=-1,outer=TRUE)
  # side=3 is top, use the defined outer area

  meanY <- mean(binorm.sample)

```

```

        covY <- cov(binorm.sample)
        corrY <- cor(binorm.sample)
        print("MEAN")
        print(muY)
        print(meanY)
        print("COV")
        print(SigmaY)
        print(covY)
        print("CORR")
        print(CorrY)
        print(corrY)
        return(list(meanY,covY,corrY))
    }

par(ask=TRUE)
for (a in avec) sim(a,muX,SigmaX)

# instead make pdf file

par(ask=FALSE)
pdf("E36b.pdf")
for (a in avec) sim(a,muX,SigmaX)
dev.off()

# c plotting with dmnorm
?dmnorm
# input is matrix with X values,
#mean vector and covariance matrix

# d Plotting with persp
par(mfrow=c(1,1))
x <- seq(-5,5,length=100)
y <- seq(-5,5,length=100)
xymat <- expand.grid(x,y)
z <- matrix(dmnorm(xymat,mean=meanvec,sigma=covarmat),ncol=100)
persp(x,y,z,theta=10,phi=30,col="lightblue",shade=0.3,expand=0.8)

zlim = range(z)
ztemp = z/zlim[2]*10
zlen = zlim[2] - zlim[1] + 1
colorlut = heat.colors(100) # height color lookup table
col = colorlut[ ztemp*10-zlim[1]+1 ]
open3d()
surface3d(x, y, ztemp, color=col,back="lines")
# grab to rotate

# e plotting with the ellipse} library (this has also
# been done on Exercise 1, so you may already know this - new thing -
# the connection between the contours and a probability distribution.)

```

```

# things will not look "right", that is 90deg
# between axis if the plot is not square,
# R scales things in many ways... also look at E1script.r from E1

par(pty="s") # makes the plotting region square,
#default is rectangular

binorm.sample <- mvrnorm(1000,meanvec,covarmat)
x <- binorm.sample[,1]
y <- binorm.sample[,2]
rangeplot <- range(c(range(x),range(y)))

plot(x,y,xlim=rangeplot,ylim=rangeplot)
# set range to same in x and y to really get same scale on x and
# y axis and then the possibility to get 90deg between axes of the ellipse
lines(ellipse(covarmat,centre=meanvec))
lines(ellipse(covarmat,centre=meanvec,level=0.95),col=2)
      #same as above, default level is 0.95

eig = eigen(covarmat)
const = sqrt(qchisq(0.95,2))
for (i in 1:2){
  lambda = eig$values[i]
  e = eig$vectors[,i]
  pkt = const*sqrt(lambda)*e
  points(c(-pkt[1],pkt[1]),c(-pkt[2],pkt[2]),col=3,pch=20)
  lines(c(-pkt[1],pkt[1]),c(-pkt[2],pkt[2]),lwd=2,col=3)
}

```