Week5-Exercises-Solutions

Exercise solutions

Week 5

Exercise 1

The solutions are embedded in the notes.

Solution in R

Exercise 2.1 Reproduce the adjusted analysis of glucose carried out in p. 72. Make sure that you exclude diabetic patients

The objective is to reproduce the adjusted analysis of Vittinghof et al. (2012) for glucose in non-diabetic patients given p. 72.

First, we select the right dataset and then use *lm* with the proper model to get the results we seek:

```
hers <- read.csv("https://www.dropbox.com/scl/fi/ywlbb7duvez2nyk66ojp1/hersdata.csv?rlkey=tml
hers.nondiab <- hers[hers$diabetes=="no",]



#deletes two missing value for the variable drinkany
hers.nondiab$drinkany[hers.nondiab$drinkany==""] <- NA
hers.nondiab$drinkany <- droplevels(hers.nondiab$drinkany)


fit <- lm(glucose ~ exercise + age + drinkany + BMI, data = hers.nondiab)
summary(fit)
##
## Call:
## lm(formula = glucose ~ exercise + age + drinkany + BMI, data = hers.nondiab)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -47.560  -6.400  -0.886   5.496  32.060
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 78.96239    2.59284  30.454   <2e-16 ***
## exerciseyes -0.95044    0.42873  -2.217   0.0267 *
## age          0.06355    0.03139   2.024   0.0431 *
## drinkanyyes  0.68026    0.42196   1.612   0.1071
## BMI          0.48924    0.04155  11.774   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.389 on 2023 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.07197,    Adjusted R-squared:  0.07013
## F-statistic: 39.22 on 4 and 2023 DF,  p-value: < 2.2e-16
```

Exercise 2.2 Use matrix operations in Stata or R to create the X,Y,X X and X Y matrices and use these to obtain the LS estimates. [Caution: there are missing values in some of these covariates so delete first all observations with missing values before any matrix manipulation]

Matrix manipulations similar to the ones carried out in Exercise 1 gives us the LSE

```
# creates the reduced dataset
hers.nondiab1 <- hers.nondiab[, c("glucose",
                     "exercise", "age",
                     "drinkany", "BMI")]


hers.nondiab1 <-na.omit(hers.nondiab1)

# create the vector of responses and the design matrix
# do not forget the columns of ones for the intercept
# check the dimension of the objects that you created



Y <- hers.nondiab1$glucose
X <- model.matrix(glucose ~ exercise + age +  #gives the design matrix
```

```
                    drinkany + BMI,
                  data = hers.nondiab1)


X <- cbind(1, hers.nondiab1$exercise, hers.nondiab1$age, hers.nondiab1$drinkany, hers.nondiab



dim(X)
## [1] 2028    5
length(Y)
## [1] 2028

# calculate the LS estimate (called b) and print it
XTX<-t(X) %*% X
b=solve(XTX)%*%t(X)%*%Y
b
##              [,1]
## [1,] 79.23257076
## [2,] -0.95044096
## [3,]  0.06354948
## [4,]  0.68026413
## [5,]  0.48924198
```

Exercise 2.2 Optional: Use an explicit matrix calculation in Stata/R to obtain the variance-covariance matrix for b in the regression of glucose on the previous covariates. Calculate the standard errors and confirm your results by comparing with the regression output.

This is the same as what has been reported in the R output. You can get the SEs using the following code.

```
sigma2 <- sum((Y- X%*%b)^2)/(406-5) #var of the residuals
matvar <- sigma2*solve(XTX)     #Var-Covariance matrix
matvar
##             [,1]          [,2]          [,3]         [,4]          [,5]
## [1,] 39.2325167 -1.7605965826 -0.3698689805 -1.803669933 -0.3318138287
## [2,] -1.7605966  0.9272978402  0.0009627852 -0.006771813  0.0142594095
## [3,] -0.3698690  0.0009627852  0.0049712413  0.006216119  0.0009753379
## [4,] -1.8036699 -0.0067718131  0.0062161192  0.898230415  0.0037299508
## [5,] -0.3318138  0.0142594095  0.0009753379  0.003729951  0.0087106974
```

```r
# extract the diagonal and take the square root - to get the SEs
SE<-sqrt(diag(matvar))
SE
## [1] 6.26358657 0.96296305 0.07050703 0.94775019 0.09333112
```

Exercise 3.1 Using your favourite software compute the 95% CI for the mean glucose of a patient aged 65, who does not drink nor exercise and has BMI=29.

Solution in R

The predicted value and its 95% CI (for a particular patient profile) can be directly obtained in R via the command predict. Note that the option *confidence* must be used for the confidence interval and *prediction* for a (wider) prediction interval.

```r
fit <- lm(glucose ~ exercise + age + drinkany + BMI, data = hers.nondiab1)
summary(fit)
##
## Call:
## lm(formula = glucose ~ exercise + age + drinkany + BMI, data = hers.nondiab1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.560  -6.400  -0.886   5.496  32.060
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 78.96239    2.59284  30.454   <2e-16 ***
## exerciseyes -0.95044    0.42873  -2.217   0.0267 *
## age          0.06355    0.03139   2.024   0.0431 *
## drinkanyyes  0.68026    0.42196   1.612   0.1071
## BMI          0.48924    0.04155  11.774   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.389 on 2023 degrees of freedom
## Multiple R-squared:  0.07197,    Adjusted R-squared:  0.07013
## F-statistic: 39.22 on 4 and 2023 DF,  p-value: < 2.2e-16
new.data <- data.frame(exercise="no", age = 65, drinkany="no", BMI=29)
## 95% CI for the mean
pred.mean <- predict(fit, new.data, interval = "confidence")
pred.mean
```

```
##        fit      lwr      upr
## 1 97.28113 96.62155 97.94071
# 95% prediction interval
pred.forecast <- predict(fit, new.data, interval = "prediction")
pred.forecast
##        fit      lwr      upr
## 1 97.28113 78.85693 115.7053
```

Can you reproduce this result using matrix manipulations and the formula given above?

Algebraically it follows from (we assumed here that we still have the LSE estimates b and the fitted model. If you want to use you calculated estimate $\hat{\sigma}$ it should work just the same.

```
x    <- c(1,0,65,0,29)
pred.manual <- x%*%b

SE<- summary(fit)$sigma*sqrt(x%*%solve(t(X) %*% X)%*%cbind(x))
lower<-pred.manual-1.96*SE
upper<-pred.manual+1.96*SE
c(pred.manual,lower,upper)
## [1] 97.55130 95.82567 99.27694
```

The same values (up to rounding) are obtained. We used here 1.96 since the sample is large but the exact quantile from the appropriate t-distribution should be used in smaller samples.

Solution in Stata

NOTE: I will use the subset of the hers data as I am not able to make calculations with matrices larger than 800 rows (Stata version constrain!)

Exercise 2.1 Reproduce the adjusted analysis of glucose carried out in p. 72. Make sure that you exclude diabetic patients

The objective is to reproduce the adjusted analysis of Vittinghof et al. (2012) for glucose in non-diabetic patients given p. 72.

First, we select the right dataset and then use *regress* with the proper model to get the results we seek:

```
* Stata code

clear
import delimited "https://www.dropbox.com/scl/fi/9dtsid3cpziubhhuw9hhy/hers_subset.csv?rlkey=
encode  exercise, gen(exercise_r)
encode drinkany , gen(drinkany_r)

drop if diabetes == "yes"

* We will use only 20% of the data
* Due to the licence limitation of STATA
* Otherwise you will get an error message

sample 20


regress glucose i.exercise_r age i.drinkany_r bmi

## (encoding automatically selected: ISO-8859-1)
## (38 vars, 276 obs)
##
##
##
## (85 observations deleted)
##
## (153 observations deleted)
##
##
##       Source |       SS           df       MS      Number of obs   =        38
## -------------+----------------------------------   F(4, 33)        =      1.50
##        Model |  625.724716         4  156.431179   Prob > F        =    0.2252
##     Residual |  3445.11739        33  104.397497   R-squared       =    0.1537
## -------------+----------------------------------   Adj R-squared   =    0.0511
##        Total |  4070.84211        37   110.02276   Root MSE        =    10.218
##
## ------------------------------------------------------------------------------
##      glucose | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##   exercise_r |
##         yes  |  -.8196387   3.683401    -0.22   0.825    -8.313574    6.674297
##          age |  -.3129571   .3041805    -1.03   0.311    -.9318169    .3059027
##              |
```

```
##   drinkany_r |
##          yes  |  -6.013101   3.734437    -1.61   0.117    -13.61087    1.584667
##          bmi  |   .3334854   .3522123     0.95   0.351    -.3830959    1.050067
##        _cons  |   110.6162   24.42434     4.53   0.000     60.92448    160.3079
## ------------------------------------------------------------------------------
```

Matrix manipulations similar to the ones carried out in Exercise 1 gives us the LSE (after deleting the missing observations).

```
clear
import delimited "https://www.dropbox.com/scl/fi/ywlbb7duvez2nyk66ojp1/hersdata.csv?rlkey=tml
encode  exercise, gen(exercise_r)
encode drinkany , gen(drinkany_r)

drop if diabetes == "yes"

* remove diabetes patients

drop if bmi ==. | drinkany_r == .

* only missing observations in BMI and drinkany

gen cons=1

* We will use only 20% of the data
* Due to the licence limitation of STATA
* Otherwise you will get an error message

sample 20


mkmat cons exercise_r age drinkany_r bmi, matrix(X)
mkmat glucose, matrix(Y)
matrix XTX =X'*X
matrix invXTX = inv(XTX)
matrix b=invXTX*X'*Y
matrix list b
```

```
(encoding automatically selected: ISO-8859-1)
(37 vars, 2,763 obs)
```

```
(731 observations deleted)

(4 observations deleted)


(1,622 observations deleted)
```

```
b[5,1]
               glucose
      cons    72.41736
exercise_r  -2.1271677
       age   .08683906
drinkany_r   2.4667362
       bmi   .64236868
```

We get the same LS estimates as the ones reported in the Stata output. You can get the SEs using the following code (that computes first $\hat{\sigma}$)

```
clear
import delimited "https://www.dropbox.com/scl/fi/9dtsid3cpziubhhuw9hhy/hers_subset.csv?rlkey=
encode  exercise, gen(exercise_r)
encode drinkany , gen(drinkany_r)

drop if diabetes == "yes"

* remove diabetes patients
drop if bmi ==. | drinkany_r == .
* only missing observations in BMI and drinkany
gen cons=1

* We will use only 20% of the data
* Due to the licence limitation of STATA
* Otherwise you will get an error message

sample 20
```

```
mkmat cons exercise_r age drinkany_r bmi, matrix(X)
mkmat glucose, matrix(Y)
matrix XTX =X'*X
matrix invXTX = inv(XTX)
matrix b=invXTX*X'*Y
matrix list b


matrix res=Y-X*b
matrix sigma2=res'*res/(406-5)

* df= number of observations(times 20% of the sample) - number of parameters=406-5

matrix list sigma2

* it is necessary to transform the matrix (1x1) in a scalar to take the square root

scalar sigma = sqrt(sigma2[1,1])
display sigma
## (encoding automatically selected: ISO-8859-1)
## (38 vars, 276 obs)
##
##
##
## (85 observations deleted)
##
## (0 observations deleted)
##
##
## (153 observations deleted)
##
##
##
##
##
##
##
## b[5,1]
##                glucose
##       cons     85.5516
## exercise_r  -5.5828251
##        age   .30620033
```

```
## drinkany_r  -4.0001402
##        bmi   .13114028
##
##
##
##
## symmetric sigma2[1,1]
##            glucose
## glucose  5.7199346
##
##
## 2.3916385
```

This is the estimate reported by Stata as Root MSE. The SEs follow:

```
clear
import delimited "https://www.dropbox.com/scl/fi/ywlbb7duvez2nyk66ojp1/hersdata.csv?rlkey=tml
encode  exercise, gen(exercise_r)
encode drinkany , gen(drinkany_r)

drop if diabetes == "yes"

* remove diabetes patients
drop if bmi ==. | drinkany_r == .
* only missing observations in BMI and drinkany
gen cons=1

* We will use only 20% of the data
* Due to the licence limitation of STATA
* Otherwise you will get an error message

sample 20

mkmat cons exercise_r age drinkany_r bmi, matrix(X)
mkmat glucose, matrix(Y)
matrix XTX =X'*X
matrix invXTX = inv(XTX)
matrix b=invXTX*X'*Y
matrix list b

* Stata code
matrix res=Y-X*b
```

```
matrix sigma2=res'*res/(406*.2-5)
* df= number of observations - number of parameters=406-5=401
matrix list sigma2
* it is necessary to transform the matrix (1x1) in a scalar to take the square root
scalar sigma = sqrt(sigma2[1,1])
display sigma


* Inverse the matrix X'X and extract the diagonal
matrix D=vecdiag(invXTX)
matrix list D
* SE(b0) intercept
scalar SE0=sqrt(D[1,1])*sigma
display SE0
* SE(b1) exercise
scalar SE1=sqrt(D[1,2])*sigma
display SE1
* SE(b2) age
scalar SE2=sqrt(D[1,3])*sigma
display SE2
## (encoding automatically selected: ISO-8859-1)
## (37 vars, 2,763 obs)
##
##
##
## (731 observations deleted)
##
## (4 observations deleted)
##
##
## (1,622 observations deleted)
##
##
##
##
##
##
##
##
## b[5,1]
##                glucose
##       cons    77.33811
## exercise_r  -2.0892924
```

```
##        age   .11858697
## drinkany_r   .21326717
##        bmi   .51288159
##
##
##
##
## symmetric sigma2[1,1]
##          glucose
## glucose  475.952
##
##
## 21.816324
##
##
##
## D[1,5]
##           cons  exercise_r         age  drinkany_r         bmi
## r1   .44899007   .01034199    .0000568   .01019343   .00010571
##
##
## 14.618404
##
##
## 2.2186237
##
##
## .16442237
```

And so on for the other coefficients. They match the SEs reported in the Stata output.

Exercise 3

Solution in Stata

```
clear
import delimited "https://www.dropbox.com/scl/fi/ywlbb7duvez2nyk66ojp1/hersdata.csv?rlkey=tml
encode  exercise, gen(exercise_r)
encode drinkany , gen(drinkany_r)

drop if diabetes == "yes"
```

```
regress glucose exercise_r age drinkany_r bmi
* 95% CI for the mean glucoae for a patient
* aged 65, no exercise, no drinking, BMI=29
adjust exercise_r=0 age=65 drinkany_r=0 bmi=29, ci
## (encoding automatically selected: ISO-8859-1)
## (37 vars, 2,763 obs)
##
##
##
## (731 observations deleted)
##
##
##       Source |       SS           df       MS      Number of obs   =     2,028
## -------------+----------------------------------   F(4, 2023)      =     39.22
##        Model |  13828.8486         4  3457.21214   Prob > F        =    0.0000
##     Residual |  178319.973     2,023  88.1463042   R-squared       =    0.0720
## -------------+----------------------------------   Adj R-squared   =    0.0701
##        Total |  192148.822     2,027  94.7946828   Root MSE        =    9.3886
##
## ------------------------------------------------------------------------------
##      glucose | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
## -------------+----------------------------------------------------------------
##   exercise_r |   -.950441     .42873    -2.22   0.027    -1.791239   -.1096426
##          age |   .0635495   .0313911     2.02   0.043     .0019872    .1251118
##   drinkany_r |   .6802641   .4219569     1.61   0.107    -.1472514     1.50778
##          bmi |    .489242   .0415528    11.77   0.000     .4077512    .5707328
##        _cons |   79.23257   2.788671    28.41   0.000      73.7636    84.70154
## ------------------------------------------------------------------------------
##
##
## ------------------------------------------------------------------------------
##      Dependent variable: glucose     Command: regress
## Covariates set to value: exercise_r = 0, age = 65, drinkany_r = 0, bmi = 29
## ------------------------------------------------------------------------------
##
## ------------------------------------------
##       All |         xb          lb          ub
## ----------+-------------------------------
##           |    97.5513    [95.8247    99.2779]
## ------------------------------------------
##      Key:  xb         =  Linear Prediction
##            [lb , ub]  =  [95% Confidence Interval]
```

Algebraically it follows from the few lines below (assuming we kept the previous quantities and estimates):

```
clear
import delimited "https://www.dropbox.com/scl/fi/ywlbb7duvez2nyk66ojp1/hersdata.csv?rlkey=tml
encode  exercise, gen(exercise_r)
encode drinkany , gen(drinkany_r)

drop if diabetes == "yes"

* remove diabetes patients
drop if bmi ==. | drinkany_r == .
* only missing observations in BMI and drinkany
gen cons=1


* We will use only 20% of the data
* Due to the licence limitation of STATA
* Otherwise you will get an error message

sample 20


mkmat cons exercise_r age drinkany_r bmi, matrix(X)
mkmat glucose, matrix(Y)
matrix XTX =X'*X
matrix invXTX = inv(XTX)
matrix b=invXTX*X'*Y
matrix list b



* Stata code
matrix res=Y-X*b
matrix sigma2=res'*res/(406-5)
* df= number of observations - number of parameters=406-5=401
matrix list sigma2
* it is necessary to transform the matrix (1x1) in a scalar to take the square root
scalar sigma = sqrt(sigma2[1,1])
display sigma
```

```
* Inverse the matrix X'X and extract the diagonal
matrix D=vecdiag(invXTX)
matrix list D
*SE(b0) intercept
scalar SE0=sqrt(D[1,1])*sigma
display SE0
*SE(b1) exercise
scalar SE1=sqrt(D[1,2])*sigma
display SE1
*SE(b2) age
scalar SE2=sqrt(D[1,3])*sigma
display SE2



matrix profile=(1,0,65,0,29)
matrix E=profile*invXTX*profile'
scalar SE=sigma*sqrt(E[1,1])
display SE
matrix pred=profile*b
display pred[1,1]
matrix list pred
matrix lower=pred-1.96*SE
matrix upper=pred+1.96*SE
matrix list lower
matrix list upper
## (encoding automatically selected: ISO-8859-1)
## (37 vars, 2,763 obs)
##
##
##
## (731 observations deleted)
##
## (4 observations deleted)
##
##
## (1,622 observations deleted)
##
##
##
##
##
```

```
##
##
## b[5,1]
##                glucose
##       cons   80.554422
## exercise_r  -1.1625011
##        age    .0820942
## drinkany_r   1.4372651
##        bmi    .37887655
##
##
##
##
## symmetric sigma2[1,1]
##            glucose
## glucose  91.055561
##
##
## 9.5423038
##
##
##
## D[1,5]
##           cons  exercise_r         age  drinkany_r         bmi
## r1    .4530393   .01082784   .00005606   .01020371   .00010226
##
##
## 6.4227523
##
##
## .99294249
##
##
## .07144555
##
##
##
##
## 1.96577
##
##
## 96.877965
```

```
##
##
## symmetric pred[1,1]
##       glucose
## r1  96.877965
##
##
##
##
## symmetric lower[1,1]
##            c1
## r1  93.025056
##
##
## symmetric upper[1,1]
##            c1
## r1  100.73087
```

The same values (up to rounding) are obtained once again. We used here 1.96 since the sample is large but the exact quantile from the appropriate t-distribution should be used in smaller samples.