Log 708 - Chapter 7 Solutions

Halvard Arntzen

7.1

a.

We'll read the data, and compute dummy variables as follows.

```
#read data
flatprices <- read.csv("M:/Undervisning/Undervisningh21/Data/flat_prices.csv")
head(flatprices)
```

```
##   price area rooms standard situated town distcen age rent
## 1  1031  100     3        2        6    1       5  15 2051
## 2  1129  116     3        1        5    1       4  42 2834
## 3  1123  110     3        2        5    1       3  25 2468
## 4   607   59     2        3        5    1       6  25 1940
## 5   858   72     2        3        4    1       1  17 1611
## 6   679   64     2        2        3    1       3  17 2039
```

```
#compute dummies as in com
flatprices$DMOL = as.integer(flatprices$town == 1)
#encode DKSU = 1 for Kristiansund
flatprices$DKSU = as.integer(flatprices$town == 2)
#encode DASU = 1 for Ålesund
flatprices$DASU = as.integer(flatprices$town == 3)
```

The following is not a part of the question, but shows a way to print a sample of the dataframe to check that we have calculated correctly:

```
#we can show a random sample of flats: (Recall chapter 3? :-) )
set.seed(1212)
n <- nrow(flatprices)
#sample 8 row numbers from 1 to n

S <- sample(1:n, size = 8,  replace = FALSE)
#show rows corresponding to S and selected columns
DF <- flatprices[S, ]
subset(DF, select = c(price, area, town, DMOL, DKSU, DASU))
```

```
##     price area town DMOL DKSU DASU
## 137  1698  159    3    0    0    1
## 108   871   94    2    0    1    0
## 126   738   74    3    0    0    1
## 64   1031  119    2    0    1    0
## 44    766   83    2    0    1    0
## 55    874  100    2    0    1    0
## 6     679   64    1    1    0    0
## 131   958   87    3    0    0    1
```

b.

We could probably figure out how the model looks from the compendium, but here we try to analyze the situation directly. Let's check what variables we have

```
names(flatprices)
```

```
##  [1] "price"    "area"     "rooms"    "standard" "situated" "town"
##  [7] "distcen"  "age"      "rent"     "DMOL"     "DKSU"     "DASU"
```

So, we should try first all except those related to town, then strip away non-significant variables.

```
modelA <- lm(price ~ area + rooms + standard + situated +
                      distcen + age + rent, data = flatprices)

summary(modelA)
```

```
##
## Call:
## lm(formula = price ~ area + rooms + standard + situated + distcen +
##     age + rent, data = flatprices)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -83.30 -55.45  25.03  49.71  89.95
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 213.361798  32.609031   6.543 1.02e-09 ***
## area         10.404197   0.418629  24.853  < 2e-16 ***
## rooms       -14.377947  10.057634  -1.430    0.155
## standard     -1.519766   5.169732  -0.294    0.769
## situated     18.201727   2.801950   6.496 1.30e-09 ***
## distcen     -24.106278   2.533768  -9.514  < 2e-16 ***
## age          -0.413237   0.613917  -0.673    0.502
## rent         -0.096044   0.008879 -10.817  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.08 on 142 degrees of freedom
## Multiple R-squared:  0.964,  Adjusted R-squared:  0.9622
## F-statistic: 543.2 on 7 and 142 DF,  p-value: < 2.2e-16
```

Three variables are non-significant, we can take out the one with max P-value: take out `standard`:

```
modelB <- update(modelA, . ~ . - standard)
summary(modelB)
```

```
##
## Call:
## lm(formula = price ~ area + rooms + situated + distcen + age +
##     rent, data = flatprices)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -83.02 -55.78  26.06  49.74  88.25
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 210.911441  31.424868   6.712 4.18e-10 ***
## area         10.406967   0.417184  24.946  < 2e-16 ***
## rooms       -14.448710  10.022583  -1.442    0.152
## situated     18.081878   2.763262   6.544 9.98e-10 ***
## distcen     -24.060495   2.520886  -9.544  < 2e-16 ***
## age          -0.420555   0.611449  -0.688    0.493
## rent         -0.096118   0.008847 -10.864  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.91 on 143 degrees of freedom
## Multiple R-squared:  0.964,  Adjusted R-squared:  0.9625
## F-statistic: 637.8 on 6 and 143 DF,  p-value: < 2.2e-16
```

Continuing, we take out `age`:

```
modelC <- update(modelB, . ~ . - age)
summary(modelC)
```

```
##
## Call:
## lm(formula = price ~ area + rooms + situated + distcen + rent,
##     data = flatprices)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -82.83 -56.31  28.38  48.76  89.01
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 199.503272  26.642177   7.488 6.42e-12 ***
## area         10.423401   0.415736  25.072  < 2e-16 ***
## rooms       -14.844367   9.987737  -1.486    0.139
## situated     18.073384   2.758175   6.553 9.37e-10 ***
## distcen     -24.061507   2.516269  -9.562  < 2e-16 ***
## rent         -0.095765   0.008816 -10.862  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.81 on 144 degrees of freedom
## Multiple R-squared:  0.9639, Adjusted R-squared:  0.9626
## F-statistic: 768.1 on 5 and 144 DF,  p-value: < 2.2e-16
```

Finally, remove `rooms`

```
modelD <- update(modelC, . ~ . - rooms)
summary(modelD)
```

```
##
## Call:
## lm(formula = price ~ area + situated + distcen + rent, data = flatprices)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -84.54 -56.62  29.45  51.12  87.32
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 213.63978   24.99003   8.549 1.59e-14 ***
## area          9.85955    0.17074  57.746  < 2e-16 ***
## situated     17.87278    2.76633   6.461 1.48e-09 ***
## distcen     -23.93052    2.52519  -9.477  < 2e-16 ***
## rent         -0.09745    0.00878 -11.099  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.03 on 145 degrees of freedom
## Multiple R-squared:  0.9633, Adjusted R-squared:  0.9623
## F-statistic: 951.7 on 4 and 145 DF,  p-value: < 2.2e-16
```

So, we end with the same model as in the compendium, with predictor variables `area, situated, distcen, rent`. Note that the R^2 is essentially the same for modelA and modelD.

c + d + e.

So, we should leave out `DKSU` and include the two others. We continue to use `update` as follows. We can call the model objects `modelPRK` and `modelPRM`

```
library(stargazer)
```

```
## Warning: package 'stargazer' was built under R version 4.0.3
```

```
##
## Please cite as:
```

```
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Ta
```

```
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
modelPRM <- update(modelD, . ~ . + DKSU + DASU)
modelPRK <- update(modelD, . ~ . + DMOL + DASU)

stargazer(modelD, modelPRM, modelPRK,
          type = "text",
          keep.stat = c("n", "rsq"))
```

```
##
## ================================================
##                        Dependent variable:
##                    --------------------------------
##                                price
##                      (1)          (2)         (3)
## ------------------------------------------------
## area              9.860***    9.978***    9.978***
##                   (0.171)      (0.033)     (0.033)
##
## situated         17.873***    15.779***   15.779***
##                   (2.766)      (0.536)     (0.536)
##
## distcen          -23.931*** -21.160***   -21.160***
##                   (2.525)      (0.496)     (0.496)
##
## rent             -0.097***    -0.098***   -0.098***
##                   (0.009)      (0.002)     (0.002)
##
## DKSU                         -106.013***
##                               (2.095)
##
```

```
## DMOL                                  106.013***
##                                        (2.095)
##
## DASU                        2.665      108.677***
##                            (2.419)      (2.163)
##
## Constant      213.640*** 254.281***   148.268***
##                (24.990)    (5.056)     (4.941)
##
## ---------------------------------------------
## Observations     150         150         150
## R2              0.963       0.999       0.999
## =============================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

The PRK model is in column (3), the R^2 is the same as for PRM, so the explanatory power is the same. Further, the coefficients common to the two models (except DASU) are also identical. We get the suspicion that the two models are in some sense identical, only differing in the chosen reference town. For PRM and PRK respectively, the constant reflects the level of the reference town, being Molde and Kristiansund. As expected these are different.

In fact, the estimated coefficient of DMOL in the PRK model represents the difference between Kristiansund and Molde. This is identical to the difference in constant terms from PRK to PRM.

f. Here we need to make a dataframe with the given information. We need to include values for all dummy variables, since these are part of the models. This dataframe will have only one row, if we don't want to try some variations.

```
test <- data.frame(area = 90, rooms = 3, standard = 3,
                   distcen = 3, situated = 8, age = 25,
                   rent = 2500, DMOL = 1, DKSU = 0, DASU = 0)

predictPRM <- predict(modelPRM, test)
predictPRK <- predict(modelPRK, test)
#print both using the "cat" function (this is just for fancier output...not necessary

cat("PRM: ", predictPRM, "    PRK: ", predictPRK)
```

```
## PRM:  968.9268     PRK:  968.9268
```

Both models predict about 970 000 for the flat.

g.

For comparisons with Kristiansund, we look to the model PRK; here we find Molde at about 106 000+ and Ålesund at about 109 000+, and both estimates are significantly different from 0. Comparing Molde and Ålesund (and asserting the significance) is easiest with the PRM

model, where we find 2665 as the exact estimated difference, and this is not significantly different from 0, meaning that we have no strong evidence of a true level difference between Molde and Ålesund.

h.

The model needs to allow modification of the β_1 coefficient, dependent on the two dummy variables, so something like:

$$Y = \beta_0 + (\beta_1 + c_M DMOL + c_A DASU)X_1 + \beta_2 X_2 + \cdots + \beta_M DMOL + \beta_A DASU + \mathcal{E}.$$

i. We can do this in the "R" way, described in chapter 7.5 by redefining `town` as a factor.

```
library(dplyr)
flatprices$town <- as.character(flatprices$town)
#recode to names "Molde", "Krsund", "Alesund":
flatprices$town <- recode(flatprices$town, "1" = "Molde", "2" = "Krsund", "3" = "Alesund
#define town as "factor".
flatprices$town <- factor(flatprices$town,
                          levels = c("Molde", "Krsund", "Alesund"))

#ensure Kristiansund is reference level.
flatprices$town <- relevel(flatprices$town, ref = "Krsund")

#then estimate:
intermodelPRK <- lm(price ~ town*area + distcen + situated + rent, data = flatprices)
summary(intermodelPRK)
```

```
##
## Call:
## lm(formula = price ~ town * area + distcen + situated + rent,
##     data = flatprices)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.000  -7.712  -0.045   6.596  29.183
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      151.099094   5.582742  27.065   <2e-16 ***
## townMolde         96.506906   7.817812  12.344   <2e-16 ***
## townAlesund      105.495634   7.653326  13.784   <2e-16 ***
## area               9.950539   0.042747 232.779   <2e-16 ***
## distcen          -21.195336   0.500512 -42.347   <2e-16 ***
## situated          15.737518   0.539886  29.150   <2e-16 ***
## rent              -0.098339   0.001715 -57.336   <2e-16 ***
## townMolde:area     0.097831   0.077562   1.261    0.209
```

```
## townAlesund:area   0.031656   0.074543   0.425    0.672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.63 on 141 degrees of freedom
## Multiple R-squared:  0.9987, Adjusted R-squared:  0.9986
## F-statistic: 1.322e+04 on 8 and 141 DF,  p-value: < 2.2e-16
```

From this we see everything except the interaction terms is significant.

j. We could go for a solution using factor representation of the variable, rather than the suggested dummy variables. Since the interactions did not come out significant, we can simply update the PRK model, after some recoding.

```
flatprices$room <- as.character(flatprices$room)
#recode to groups
flatprices$room <- recode_factor(flatprices$room, "1" = "R1", "2" = "R2", "3" = "R3",
                                                  "4" = "R4", .default = "R5")




extendedmodel <- update(modelPRK, . ~ . + room)
summary(extendedmodel)
```

```
##
## Call:
## lm(formula = price ~ area + situated + distcen + rent + DMOL +
##     DASU + room, data = flatprices)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.5064  -7.4962   0.0229   5.7055  28.2811
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 150.969406   7.509109  20.105   <2e-16 ***
## area          9.888065   0.071733 137.846   <2e-16 ***
## situated     15.696140   0.552118  28.429   <2e-16 ***
## distcen     -21.106653   0.501243 -42.109   <2e-16 ***
## rent         -0.098856   0.001747 -56.592   <2e-16 ***
## DMOL        106.149539   2.115873  50.168   <2e-16 ***
## DASU        108.917450   2.205048  49.395   <2e-16 ***
## roomR2        4.200167   5.992772   0.701    0.485
## roomR3        7.605056   6.817079   1.116    0.267
## roomR4        9.777240   7.933441   1.232    0.220
```

```
## roomR5       12.569937   9.880324   1.272    0.205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.68 on 139 degrees of freedom
## Multiple R-squared:  0.9987, Adjusted R-squared:  0.9986
## F-statistic: 1.048e+04 on 10 and 139 DF,  p-value: < 2.2e-16
```

None of the levels of `room` appears to be significant, when controlling for other variables.

k.

Using dummy variables, say $R3$ for 3-room flats, and $DMOL$ for Molde, then

$$DM3 = DMOL \cdot R3$$

would be a dummy that is 1 only for 3-rooms flats in Molde. Using DM3 to correct β_1 in e.g. PRK would give the mentioned effect. For these data such modeling detail is hardly recommended, but for large data sets it could be just the thing we need to study some subtle phenomena!

7.2

Need to start reading data as usual.

```
library(stargazer)
wagedata <- read.csv("M:/Undervisning/Undervisningh21/Data/Wages.csv")
head(wagedata)
```

```
##   obs  wage female nonwhite union education exper age wind femalenonw
## 1   1 11.55      1        0     0        12    20  38    1          0
## 2   2  5.00      0        0     0         9     9  24    0          0
## 3   3 12.00      0        0     0        16    15  37    1          0
## 4   4  7.00      0        1     1        14    38  58    0          0
## 5   5 21.15      1        1     0        16    19  41    1          1
## 6   6  6.92      1        0     0        12     4  22    1          0
```

a. We'll run a regression with the suggested variables. We recode `female` into a factor, called `gender` with levels male, female. This is just to improve on the look of output. We can still keep the original variable to compare.

```
library(dplyr)
wagedata$gender <- as.character(wagedata$female)

wagedata$gender <- recode(wagedata$gender, "0" = "male", "1" = "female")
wagedata$gender <- factor(wagedata$gender, levels = c("male", "female"))
wagedata$gender <- relevel(wagedata$gender, ref = "male")
```

```
wagereg1 <- lm(wage ~ education + exper + female, data = wagedata)
wagereg2 <- lm(wage ~ education + exper + gender, data = wagedata)

stargazer(wagereg1, wagereg2, type = "text",
          keep.stat=c("rsq", "ser"),
          column.labels = c("model1", "model2"),
          model.numbers = FALSE)
```

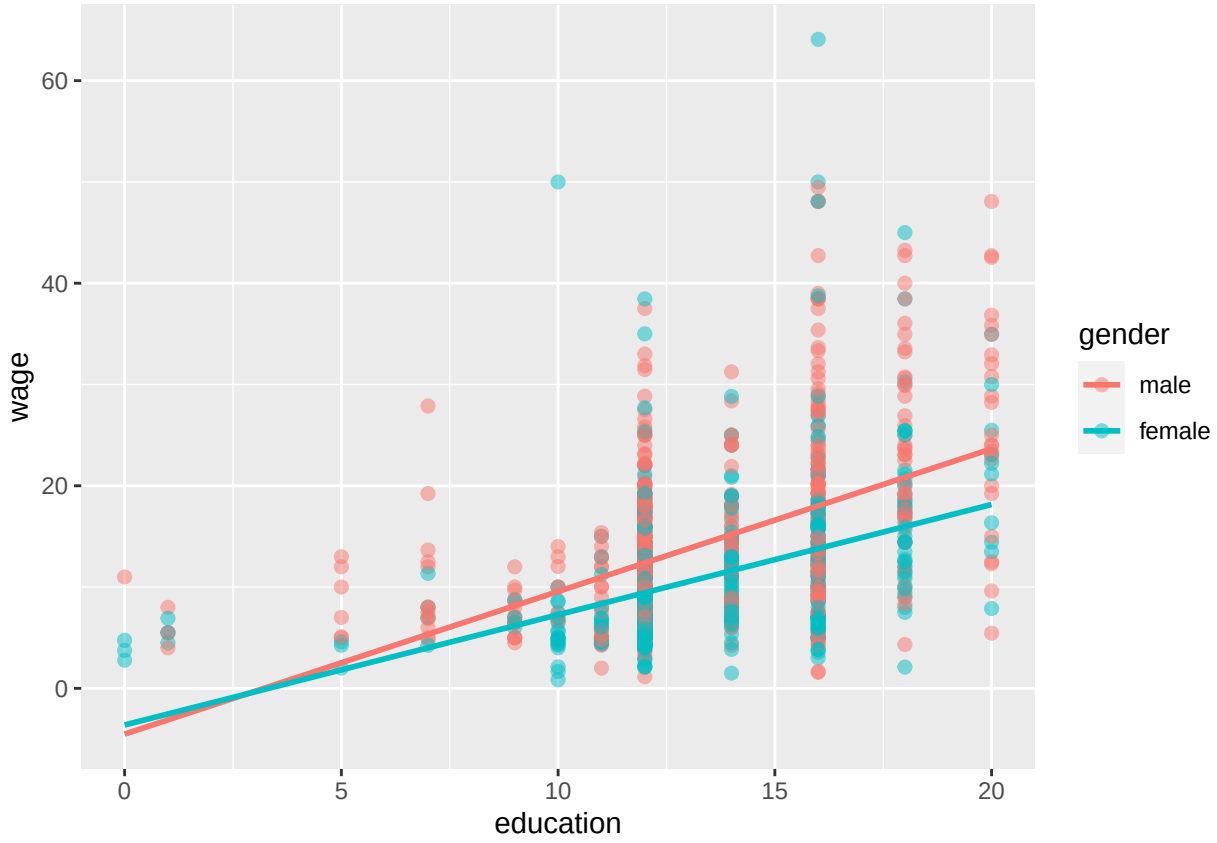
```
##
## ===============================================================
##                                    Dependent variable:
##                                ---------------------------
##                                            wage
##                                    model1        model2
## ---------------------------------------------------------------
## education                         1.394***      1.394***
##                                   (0.066)        (0.066)
##
## exper                             0.175***      0.175***
##                                   (0.016)        (0.016)
##
## female                           -3.186***
##                                   (0.364)
##
## genderfemale                                    -3.186***
##                                                  (0.364)
##
## Constant                         -7.653***      -7.653***
##                                   (1.006)        (1.006)
##
## ---------------------------------------------------------------
## R2                                 0.316          0.316
## Residual Std. Error (df = 1285)    6.536          6.536
## ===============================================================
## Note:                           *p<0.1; **p<0.05; ***p<0.01
```

The two models are of course identical.

b. Since we introduced the factor `gender` we can do like this:

```
library(ggplot2)
ggplot(wagedata, aes(x = education, y = wage, color = gender)) +
  geom_point(size = 2, alpha = 0.5) +
  geom_smooth(method = lm, se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



It seems that years of education pays off better for male workers.

c. Testing whether a significant interaction exist between `gender` and `education`.

```
wagereg3 <- lm(wage ~ education*gender + exper, data = wagedata)
stargazer(wagereg2, wagereg3, type = "text",
          keep.stat = c("rsq", "ser"),
          column.labels = c("model2", "model3"),
          model.numbers = FALSE)
```

```
##
## ===============================================================
##                                 Dependent variable:
##                             ----------------------------------
##                                         wage
##                                 model2            model3
## ---------------------------------------------------------------
## education                      1.394***          1.503***
##                                 (0.066)           (0.088)
##
## exper                          0.175***          0.173***
##                                 (0.016)           (0.016)
##
```

```
## education:genderfemale                          -0.243*
##                                                 (0.130)
##
## genderfemale                  -3.186***          0.008
##                                (0.364)          (1.747)
##
## Constant                      -7.653***        -9.074***
##                                (1.006)          (1.260)
##
## ---------------------------------------------------------
## R2                               0.316            0.318
## Residual Std. Error    6.536 (df = 1285) 6.530 (df = 1284)
## =========================================================
## Note:                              *p<0.1; **p<0.05; ***p<0.01
```

Comparing models 2 and 3 we find something interesting. In model 3 the overall gender difference is not significant, while we get an interaction effect as visualized above that is significant. Judging by model 3, on average, men gain 1.503$ per extra year of education, while women gain 0.243$ less, so about 1.26$ per extra year for women. So, it appears that the "flat" difference in model 2 is not correct. Using a more detailed model, we can see that the difference is caused by interaction.

d. We should make a dataframe for Jim and Maria, and compare the forecast wages using model3

```r
testworkers  <- data.frame(name = c("Maria", "Jim"),
                           education = c(12, 12),
                           exper = c(10, 10),
                           gender = c("female", "male"))

# alternatively, if we use the original "female" variable, we should have
# female = c(1, 0)  in defining the dataframe.

predictedwages <- predict(wagereg3, testworkers)

cbind(testworkers, round(predictedwages, 2))
```

```
##    name education exper gender round(predictedwages, 2)
## 1 Maria        12    10 female                     7.79
## 2   Jim        12    10   male                    10.70
```

The expected wages are 7.8, and 10.7 dollars respectively.

e. If the female and male workforce was homogeneous in all variables not included in the regression, we could say that the result strongly points to a large wage discrimination of women. There is however good reasons to expect that some of the estimated effects are biased from omitted variables. For example the model does not account for the

type of position (e.g. leader at some level vs ordinary worker) or the type of industry (e.g. mining vs healt care). It is likely that wage differences and gender differences relating to such factors lead to errors in the estimated interaction effect. If we could control for type of position and industry to some extent, we come closer to a comparison of men and women doing the *same job* which is where the question of "fair wages" can be adressed in a better way. (A different issue is of course whether the gender and wage differences based on position and industry type are fair. I.e. is it "fair" that more women work in health care and that health care workers generally earn less than miners?)

The data is somewhat old, and at the time probably more of "old fashioned" gender divisions were present in the USA than today. Newer data could give a different result, of course.

f. We are to test for an interaction effect as suggested.

```
wagereg4 <- lm(wage ~ education*gender + exper*gender, data = wagedata)
stargazer(wagereg3, wagereg4, type = "text",
          keep.stat = c("rsq", "ser"),
          column.labels = c("model2", "model3"),
          model.numbers = FALSE)
```

```
##
## ============================================================
##                                Dependent variable:
##                            ------------------------------
##                                        wage
##                               model2            model3
## ------------------------------------------------------------
## education                    1.503***          1.531***
##                               (0.088)           (0.088)
##
## genderfemale                  0.008             2.906
##                               (1.747)           (1.966)
##
## exper                        0.173***          0.224***
##                               (0.016)           (0.023)
##
## education:genderfemale        -0.243*           -0.320**
##                               (0.130)           (0.132)
##
## genderfemale:exper                             -0.100***
##                                                (0.032)
##
## Constant                     -9.074***         -10.418***
##                               (1.260)           (1.325)
##
```

```
## ------------------------------------------------------------
## R2                             0.318             0.324
## Residual Std. Error    6.530 (df = 1284) 6.507 (df = 1283)
## ============================================================
## Note:                            *p<0.1; **p<0.05; ***p<0.01
```

From model 4 we learn that also the `experience` variable has a significant interaction with `gender`. This effect can to some extent be caused by the way the "experience" variable was calculated (as outlined in exercise 6.1). Since more women spend some years at home with children the "experience" variable more often give the correct value for men, while for women it overestimates the professional work experience, and so it appears that experience pays off less for women.

As final words, this exercise exemplifies that wage modeling can be really challenging, and with the limited selection of variables available here, we are likely to have severe bias in our estimates. Interestingly, we could see that the "flat" wage difference was not real, an interaction effect gave a better picture of the situation. Further it is an example where regression models can be highly valuable even with relatively low R^2 values. The highest we got here was around 0.32.