Module 2: MULTIPLE LINEAR REGRESSION, solution IL week 1

TMA4315 Generalized linear models H2018

Contents

R packages

```
install.packages(c("formatR", "gamlss.data", "leaps"))
```

Problem 1: Theory

1.

Fix covariates X. *Collect Y, create CI using $\hat{\beta}$ and $\hat{\sigma}$*, repeat from * to * many times. 95 % of the times the CI contains the true β. Collect Y means simulate it with the true β as parameter(s). The following R-code illustrates this:

```
# CI for beta_j

true_beta <- c(3.14, 10, 0.8)  # choosing true betas
true_sd <- 10  # choosing true sd
```

```r
set.seed(345)
X <- matrix(c(rep(1, 100), runif(100, 2, 5), sample(1:100, 100, replace = TRUE)),
    nrow = 100, ncol = 3)  # fixing X. set.seed() is used to produce same X every time this code is use

# simulating and fitting models many times
ci_int <- ci_x1 <- ci_x2 <- 0
nsim <- 1000
for (i in 1:nsim) {
    y <- rnorm(n = 100, mean = X %*% true_beta, sd = rep(true_sd, 100))
    mod <- lm(y ~ x1 + x2, data = data.frame(y = y, x1 = X[, 2], x2 = X[,
        3]))
    ci <- confint(mod)
    ci_int[i] <- ifelse(true_beta[1] >= ci[1, 1] && true_beta[1] <= ci[1,
        2], 1, 0)
    ci_x1[i] <- ifelse(true_beta[2] >= ci[2, 1] && true_beta[2] <= ci[2,
        2], 1, 0)
    ci_x2[i] <- ifelse(true_beta[3] >= ci[3, 1] && true_beta[3] <= ci[3,
        2], 1, 0)
}

c(mean(ci_int), mean(ci_x1), mean(ci_x2))
```

```
## [1] 0.952 0.944 0.945
```

2.

No solution.

3.

No solution.

4.

They use two different tests: `summary` tests if a given cofficient can be 0 while the others are present, `anova` tests if the coefficient reduces the SSE enough to be allowed in the model sequentially.

5.

SSE(small) $\geq$ SSE(large) since SSE will be smaller with more covariates explaining variation (and for a covariate that is completly unrelated to the data it might not be a large change, but the SSE will not increase). R^2 directly related to SSE: R^2 = 1 - SSE/SST, and SST does not change when the model changes.

6.

The deviance of model A is given by:

$-2(\ln L(A) - \ln L(\text{saturated model}))$

This is the same as the likelihood ratio test statistic of the saturated model and model A (model A is the smaller model).

The *saturated model* is a model where the deviance (per def) is 0. We have so many covariates that the $\hat{Y}$ are all correct, and we have no degrees of freedom left.

The deviance can never be negative (the log-likelihood is always larger for a better model fit, i.e., for a model with more covariates), so the deviance can not become smaller than 0. The saturated model has a deviance of 0.

Problem 2: Dummy vs. effect coding in MLR (continued)

1.

No solution, see module pages and solution from last week.

2.

```r
income <- c(300, 350, 370, 360, 400, 370, 420, 390, 400, 430, 420, 410,
    300, 320, 310, 305, 350, 370, 340, 355, 370, 380, 360, 365)
gender <- c(rep("Male", 12), rep("Female", 12))
place <- rep(c(rep("A", 4), rep("B", 4), rep("C", 4)), 2)
data <- data.frame(income, gender = factor(gender, levels = c("Female",
    "Male")), place = factor(place, levels = c("A", "B", "C")))

model <- lm(income ~ place - 1, data = data, x = TRUE)
model$x
```

```
##    placeA placeB placeC
## 1       1      0      0
## 2       1      0      0
## 3       1      0      0
## 4       1      0      0
## 5       0      1      0
## 6       0      1      0
## 7       0      1      0
## 8       0      1      0
## 9       0      0      1
## 10      0      0      1
## 11      0      0      1
## 12      0      0      1
## 13      1      0      0
## 14      1      0      0
## 15      1      0      0
## 16      1      0      0
## 17      0      1      0
## 18      0      1      0
## 19      0      1      0
## 20      0      1      0
## 21      0      0      1
## 22      0      0      1
## 23      0      0      1
```

```
## 24      0      0      1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$place
## [1] "contr.treatment"
```

```r
summary(model)
```

```
##
## Call:
## lm(formula = income ~ place - 1, data = data, x = TRUE)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.375 -22.500  -5.625  23.750  45.625
##
## Coefficients:
##        Estimate Std. Error t value Pr(>|t|)
## placeA  326.875      9.733   33.58   <2e-16 ***
## placeB  374.375      9.733   38.46   <2e-16 ***
## placeC  391.875      9.733   40.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.53 on 21 degrees of freedom
## Multiple R-squared:  0.9951, Adjusted R-squared:  0.9944
## F-statistic:  1409 on 3 and 21 DF,  p-value: < 2.2e-16
```

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: income
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## place      3 3204559 1068186  1409.4 < 2.2e-16 ***
## Residuals 21   15916     758
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Dummy coding is used.

You include either place A, B or C by calculating $\beta_A x_A + \beta_B x_B + \beta_C x_C$ where just one of x_A, x_B and x_C is 1, and the two others 0. See that the income is lowest at location A, and highest at location C.

H_0 is that the model does not contain the covariate `place`, i.e., the model is just `y ~ 0` as we have no intercept. This is very, very unlikely and we keep the covariate.

3.

```r
model1 <- lm(income ~ place, data = data, x = TRUE, contrasts = list(place = "contr.treatment"))
head(model1$x)
```

```
##   (Intercept) placeB placeC
## 1           1      0      0
## 2           1      0      0
```

```
## 3            1      0      0
## 4            1      0      0
## 5            1      1      0
## 6            1      1      0
```

```r
summary(model1)
```

```
##
## Call:
## lm(formula = income ~ place, data = data, x = TRUE, contrasts = list(place = "contr.treatment"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.375 -22.500  -5.625  23.750  45.625
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  326.875      9.733  33.583  < 2e-16 ***
## placeB        47.500     13.765   3.451 0.002394 **
## placeC        65.000     13.765   4.722 0.000116 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.53 on 21 degrees of freedom
## Multiple R-squared:  0.5321, Adjusted R-squared:  0.4875
## F-statistic: 11.94 on 2 and 21 DF,  p-value: 0.000344
```

```r
model2 <- lm(income ~ place, data = data, x = TRUE, contrasts = list(place = "contr.sum"))
head(model2$x)
```

```
##   (Intercept) place1 place2
## 1           1      1      0
## 2           1      1      0
## 3           1      1      0
## 4           1      1      0
## 5           1      0      1
## 6           1      0      1
```

```r
summary(model2)
```

```
##
## Call:
## lm(formula = income ~ place, data = data, x = TRUE, contrasts = list(place = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.375 -22.500  -5.625  23.750  45.625
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  364.375      5.619  64.841  < 2e-16 ***
## place1       -37.500      7.947  -4.719 0.000117 ***
## place2        10.000      7.947   1.258 0.222090
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 27.53 on 21 degrees of freedom
## Multiple R-squared:  0.5321, Adjusted R-squared:  0.4875
## F-statistic: 11.94 on 2 and 21 DF,  p-value: 0.000344
```

`model1` is dummy, `model2` is effect.

model1: `intercept` is the income at place A, `intercept` plus `placeB` is the income at place B, and `intercept` plus `placeC` is the income at place C.

model2: `intercept` plus `place1` is the income at place A, `intercept` plus `place2` is the income at place B, and `intercept` minus `place1` minus `place2` is the income at place C.

This we can see from the design matrix X printed using `model1$x`. The design matrices for the two models differ, and thus the interpretation of the parameters also differ.

4.

```r
# have no covariates, so dummy or effect coding does not matter
model0 <- lm(income ~ 1, data = data)

anova(model0, model1)
```

```
## Analysis of Variance Table
##
## Model 1: income ~ 1
## Model 2: income ~ place
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1     23 34016
## 2     21 15916  2     18100 11.941 0.000344 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(model0, model2)
```

```
## Analysis of Variance Table
##
## Model 1: income ~ 1
## Model 2: income ~ place
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1     23 34016
## 2     21 15916  2     18100 11.941 0.000344 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The results are the same, since we test for the whole variable at once and not only one of the levels (which is done in summary). Conclusion is to keep the covariate.

5.

```r
model3 <- lm(income ~ place + gender, data = data, x = TRUE, contrasts = list(place = "contr.treatment"
    gender = "contr.treatment"))
summary(model3)
```

```
##
## Call:
```

```
## lm(formula = income ~ place + gender, data = data, x = TRUE,
##     contrasts = list(place = "contr.treatment", gender = "contr.treatment"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.500  -6.250   0.000   9.687  25.000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  306.250      6.896  44.411  < 2e-16 ***
## placeB        47.500      8.446   5.624 1.67e-05 ***
## placeC        65.000      8.446   7.696 2.11e-07 ***
## genderMale    41.250      6.896   5.982 7.54e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.89 on 20 degrees of freedom
## Multiple R-squared:  0.8322, Adjusted R-squared:  0.8071
## F-statistic: 33.07 on 3 and 20 DF,  p-value: 6.012e-08
```

```r
anova(model3)
```

```
## Analysis of Variance Table
##
## Response: income
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## place      2 18100.0  9050.0  31.720 6.260e-07 ***
## gender     1 10209.4 10209.4  35.783 7.537e-06 ***
## Residuals 20  5706.2   285.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model4 <- lm(income ~ place + gender, data = data, x = TRUE, contrasts = list(place = "contr.sum",
    gender = "contr.sum"))
summary(model4)
```

```
##
## Call:
## lm(formula = income ~ place + gender, data = data, x = TRUE,
##     contrasts = list(place = "contr.sum", gender = "contr.sum"))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.500  -6.250   0.000   9.687  25.000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  364.375      3.448 105.680  < 2e-16 ***
## place1       -37.500      4.876  -7.691 2.13e-07 ***
## place2        10.000      4.876   2.051   0.0536 .
## gender1      -20.625      3.448  -5.982 7.54e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.89 on 20 degrees of freedom
```

```
## Multiple R-squared:  0.8322, Adjusted R-squared:  0.8071
## F-statistic: 33.07 on 3 and 20 DF,  p-value: 6.012e-08
```

```r
anova(model4)
```

```
## Analysis of Variance Table
##
## Response: income
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## place      2 18100.0  9050.0  31.720 6.260e-07 ***
## gender     1 10209.4 10209.4  35.783 7.537e-06 ***
## Residuals 20  5706.2   285.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In `model3` dummy coding is used, and in `model4` effect coding is used.

`model3`: same as for `model1`, but now also adding `genderMale` if the person of interest is male (and nothing if the person is female).

`model4`: same as for `model2`, but now adding `gender1` if the person of interest is female, and subtracting `gender1` if the person is male.

The anova tables are equal for the two models since we test for the change when the whole covariate (not only the level) is included/excluded.

6.

```r
model5 <- lm(income ~ place + gender + place:gender)  # or lm(income ~ place*gender)
summary(model5)
```

```
##
## Call:
## lm(formula = income ~ place + gender + place:gender)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.000  -5.938   1.250  11.250  25.000
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        308.750      8.824  34.989  < 2e-16 ***
## placeB              45.000     12.479   3.606 0.002020 **
## placeC              60.000     12.479   4.808 0.000141 ***
## genderMale          36.250     12.479   2.905 0.009446 **
## placeB:genderMale    5.000     17.648   0.283 0.780168
## placeC:genderMale   10.000     17.648   0.567 0.577963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.65 on 18 degrees of freedom
## Multiple R-squared:  0.8352, Adjusted R-squared:  0.7894
## F-statistic: 18.24 on 5 and 18 DF,  p-value: 1.74e-06
```

```r
anova(model5)
```

```
## Analysis of Variance Table
##
## Response: income
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## place         2 18100.0  9050.0 29.0569 2.314e-06 ***
## gender        1 10209.4 10209.4 32.7793 1.988e-05 ***
## place:gender  2   100.0    50.0  0.1605    0.8529
## Residuals    18  5606.2   311.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Both summary and anova says the interaction is not significant (which is what we suspected last week).

Problem 3: Compulsory exercise 1

No solution.

Problem 4: Munich Rent index (optional)

```r
library(gamlss.data)
library(dplyr)
data("rent99")
```

1.

```r
formula <- rent ~ area + location + bath + kitchen + cheating
rent1 <- lm(formula, data = rent99)  #, contrasts = list(location = 'contr.sum'))

rent99 <- rent99 %>% mutate(yearc.cat = cut(yearc, breaks = c(-Inf, seq(1920,
    2000, 10)), labels = 10 * 1:9))

formula <- rent ~ area + location + bath + kitchen + cheating + yearc.cat
rent2 <- lm(formula, data = rent99)  #, contrasts = list(location = 'contr.sum'))

rent99 <- rent99 %>% mutate(yearc.cat2 = cut(yearc, breaks = c(-Inf,
    seq(1920, 2000, 20)), labels = c(20, 40, 60, 80, 0)))

formula <- rent ~ area + location + bath + kitchen + cheating + yearc.cat2
rent3 <- lm(formula, data = rent99)  #, contrasts = list(location = 'contr.sum'))
```

2.

```r
library(MASS)
library(leaps)
res1 <- regsubsets(model.matrix(rent3)[, -1], y = rent99$rent)
summary(res1)$bic
```

```
## [1] -1272.388 -1646.402 -1918.488 -1953.952 -2003.720 -2046.262 -2082.760
## [8] -2099.847
```

3.

```r
res2 <- stepAIC(rent3)
```

```
## Start:  AIC=30369.92
## rent ~ area + location + bath + kitchen + cheating + yearc.cat2
##
##              Df Sum of Sq      RSS   AIC
## <none>                    58244480 30370
## - bath        1    476281 58720761 30393
## - kitchen     1    806944 59051424 30410
## - location    2   2346788 60591268 30488
## - cheating    1   4158813 62403293 30580
## - yearc.cat2  4   6575067 64819547 30692
## - area        1  34256726 92501206 31794
```

```r
step(res2)
```

```
## Start:  AIC=30369.92
## rent ~ area + location + bath + kitchen + cheating + yearc.cat2
##
##              Df Sum of Sq      RSS   AIC
## <none>                    58244480 30370
## - bath        1    476281 58720761 30393
## - kitchen     1    806944 59051424 30410
## - location    2   2346788 60591268 30488
## - cheating    1   4158813 62403293 30580
## - yearc.cat2  4   6575067 64819547 30692
## - area        1  34256726 92501206 31794

##
## Call:
## lm(formula = rent ~ area + location + bath + kitchen + cheating +
##     yearc.cat2, data = rent99)
##
## Coefficients:
##  (Intercept)          area     location2     location3         bath1
##      -38.376         4.785        47.270       124.028        53.586
##     kitchen1     cheating1  yearc.cat240  yearc.cat260  yearc.cat280
##       81.910       130.696       -13.010        -1.602        37.596
##  yearc.cat20
##      139.097
```