Module 3: Recommended Exercises

TMA4268 Statistical Learning V2022

Emma Skarstein, Daesoo Lee, Stefanie Muff
Department of Mathematical Sciences, NTNU

January 24, 2022

We strongly recommend you to work through the Section 3.6 in the course book (Lab on linear regression)

Problem 1 (Extension from Book Ex. 9)

This question involves the use of multiple linear regression on the `Auto` data set from `ISLR` package (you may use `?Auto` to see a description of the data). First we exclude from our analysis the variable `name` and look at the data summary and structure of the dataset.

```
library(ISLR)
Auto = subset(Auto, select = -name)
# Auto$origin = factor(Auto$origin)
summary(Auto)
```

```
##       mpg          cylinders      displacement     horsepower        weight    
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613  
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225  
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804  
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978  
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615  
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140  
##   acceleration        year           origin     
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000  
##  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000  
##  Median :15.50   Median :76.00   Median :1.000  
##  Mean   :15.54   Mean   :75.98   Mean   :1.577  
##  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000  
##  Max.   :24.80   Max.   :82.00   Max.   :3.000  
```

```
str(Auto)
```

```
## 'data.frame':    392 obs. of  8 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
```

```
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num  1 1 1 1 1 1 1 1 1 1 ...
```

We obtain a summary and see that all variables are numerical (continuous). However, when we check the description of the data (again with `?Auto`) we immediately see that `origin` is actually encoding for either American (origin=1), European (origin=2) or Janapense (origin=3) origin of the car, thus the values 1, 2 and 3 do not have any actual numerical meaning. We therefore need to first change the data type of that variable to let R know that we are dealing with a qualitative (categorical) variable, instead of a continuous one (otherwise we will obtain wrong model fits). In R such variables are called *factor variables*, and before we continue to do any analyses we first need to convert `origin` into a factor variable (a synonymous for "qualitative predictor"):

```
Auto$origin = factor(Auto$origin)
```

a)

Use the function `ggpairs()` from `GGally` package to produce a scatterplot matrix which includes all of the variables in the data set.

b)

Compute the correlation matrix between the variables. You will need to remove the factor covariate `origin`, because this is no longer a continuous variable.

c)

Use the `lm()` function to perform a multiple linear regression with `mpg` (miles per gallon, a measure for fuel consumption) as the response and all other variables (except `name`) as the predictors. Use the `summary()` function to print the results. Comment on the output. In particular:

i. Is there a relationship between the predictors and the response?
ii. Is there evidence that the weight of a car influences `mpg`? Interpret the regression coefficient β_{weight} (what happens if a car weights 1000kg more, for example?).
iii. What does the coefficient for the year variable suggest?

d)

Look again at the regression output from question c). Now we want to test whether the `origin` variable is important. How does this work for a factor variable with more than only two levels?

e)

Use the `autoplot()` function from the `ggfortify` package to produce diagnostic plots of the linear regression fit by setting `smooth.colour = NA`, as sometimes the smoothed line can be misleading. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

f)

For beginners, it can be difficult to decide whether a certain QQ plot looks "good" or "bad", because we only look at it and do not test anything. A way to get a feeling for how "bad" a QQ plot may look, even when the normality assumption is perfectly ok, we can use simulations: We can simply draw from the normal distribution and plot the QQ plot. Use the following code to repeat this six times:

```
set.seed(2332)
n = 100

par(mfrow = c(2, 3))
for (i in 1:6) {
    sim = rnorm(n)
    qqnorm(sim, pch = 1, frame = FALSE)
    qqline(sim, col = "blue", lwd = 1)
}
```

g)

Let us look at interactions. These can be included via the `*` or `:` symbols in the linear predictor of the regression function (see Section 3.6.4 in the course book).

Fit another model for `mpg`, including only `displacement`, `weight`, `year` and `origin` as predictors, plus an interaction between `year` and `origin` (interactions can be included as `year*origin`; this adds the main effects and the interaction at once). Is there evidence that the interactions term is relevant? Give an interpretation of the result.

h)

Try a few different transformations of the variables, such as $\log(X)$, $\sqrt{X}$, X^2. See Section 3.6.5 in the course book for how to do this. Perhaps you manage to improve the residual plots that you got in e)? Comment on your findings.

Problem 2

a)

A core finding for the least-squares estimator $\hat{\boldsymbol\beta}$ of linear regression models is

$$\hat{\boldsymbol\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \ ,$$

with $\hat{\boldsymbol\beta} \sim N_p(\boldsymbol\beta, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$.

- Show that $\hat{\boldsymbol\beta}$ has this distribution with the given mean and covariance matrix.
- What do you need to assume to get to this result?
- What does this imply for the distribution of the jth element of $\hat{\boldsymbol\beta}$?
- In particular, how can we calculate the variance of $\hat{\beta}_j$?

b)

What is the interpretation of a 95% confidence interval? Hint: repeat experiment (on Y), on average how many CIs cover the true β_j? The following code shows an interpentation of a 95% confidence interval. Study and fill in the code where is needed

- Model: $Y = 1 + 3X + \varepsilon$, with $\varepsilon \sim \mathsf{N}(0, 1)$.

```
beta0 = ...
beta1 = ...
true_beta = c(beta0, beta1)  # vector of model coefficients
true_sd = 1  # choosing true sd
X = runif(100, 0, 1)  # simulate the predictor variable X
Xmat = model.matrix(~X, data = data.frame(X))  # create design matrix


ci_int = ci_x = 0  # Counts how many times the true value is within the confidence interval
nsim = 1000
for (i in 1:nsim) {
    y = rnorm(n = 100, mean = Xmat %*% true_beta, sd = rep(true_sd, 100))
    mod = lm(y ~ x, data = data.frame(y = y, x = X))
    ci = confint(mod)
    ci_int[i] = ifelse(..., 1, 0)  # if true value of beta0 is within the CI then 1 else 0
    ci_x[i] = ifelse(..., 1, 0)  # if true value of beta_1 is within the CI then 1 else 0
}

c(mean(ci_int), mean(ci_x))
```

c)

What is the interpretation of a 95% prediction interval? Hint: repeat experiment (on Y) for a given $\boldsymbol{x}_0$. Write R code that shows the interprentation of a 95% PI. Hint: In order to produce the PIs use the data point $x_0 = 0.4$. Furthermore you may use a similar code structure as in b).

d)

Construct a 95% CI for $\boldsymbol{x}_0^T\beta$. Explain what is the connections between a CI for β_j, a CI for $\boldsymbol{x}_0^T\beta$ and a PI for Y at $\boldsymbol{x}_0$.

e)

Explain the difference between *error* and *residual*. What are the properties of the raw residuals? Why don't we want to use the raw residuals for model check? What is our solution to this?