Technical University of Munich
Department of Mathematics

MA4401 Applied Regression
Homework problem 3

Prof. Donna Ankerst, Stephan Haug

November 28, 2017

Grading: For each ✓ you should give 2, for each ✓ 1 and for each ✓ 0.5 points. For each part there exist also other approaches to solve the problem. Hence you have to check if solutions, which are different than the proposed one, are also correct.

Remark: If in the end the total number of points is not an integer (moodle only knows integers), you have to round up. So e.g. a result of 8.5 will be evaluated as 9 in moodle.

Problem H.3

Data on last year's sales (`Y`), in 100,000s of dollars, in 15 sales districts are given in the file `sales.csv`. This file also contains promotional expenditures (`X1`), in thousands of dollars, the number of active accounts (`X2`), the number of competing brands (`X3`), and the district potential (`X4`), coded, for each of the districts.

a) Produce a pairs plot of the data using `GGally::ggpairs()`. Describe what you see concerning the relation between the four predictors and the response `Y`.

b) Fit a model, containing all predictor variables, to the data. Do a graphical residual analysis by using `autoplot()` (depends on the `ggfortify` package). Are there strong violations of the model assumptions?

c) Test the following hypotheses:

   (i) $\beta_4 = 0$

   (ii) $\beta_3 = \beta_4 = 0$

   (iii) $\beta_2 = \beta_3$

   (iv) $\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$

   at the 5% level. *Hint:* See Lecture 2b, p. 28ff

d) Write a function `test_linht()`, which computes the test statistic and p-value for test as given in part c). The input of the function should be the full and reduced model, the degrees of freedom for the additional sum of squares and the degrees of freedom for the residual sum of squares of the full model. The output of the function should be like this

```
test_linht(model_red, model_full, df_add = 1, df_full = 10)

## $test_statistic
## [1] 0.4074726
##
## $p_value
## [1] 0.5375986
```

   Apply your function to the test problem from part c) (ii).

   Hint: Use a `list` as output of your function. The test statistic and the p-value can then be different elements of the list.

e) Consider the reduced model with $\beta_4 = 0$. Estimate the regression coefficients in this reduced model. Give an interpretation of the estimated coefficients.

f) Using the model in e), obtain a prediction for the sales in a district where `X1=3`, `X2=45`, and `X3=10`. Obtain the corresponding 95% prediction interval.

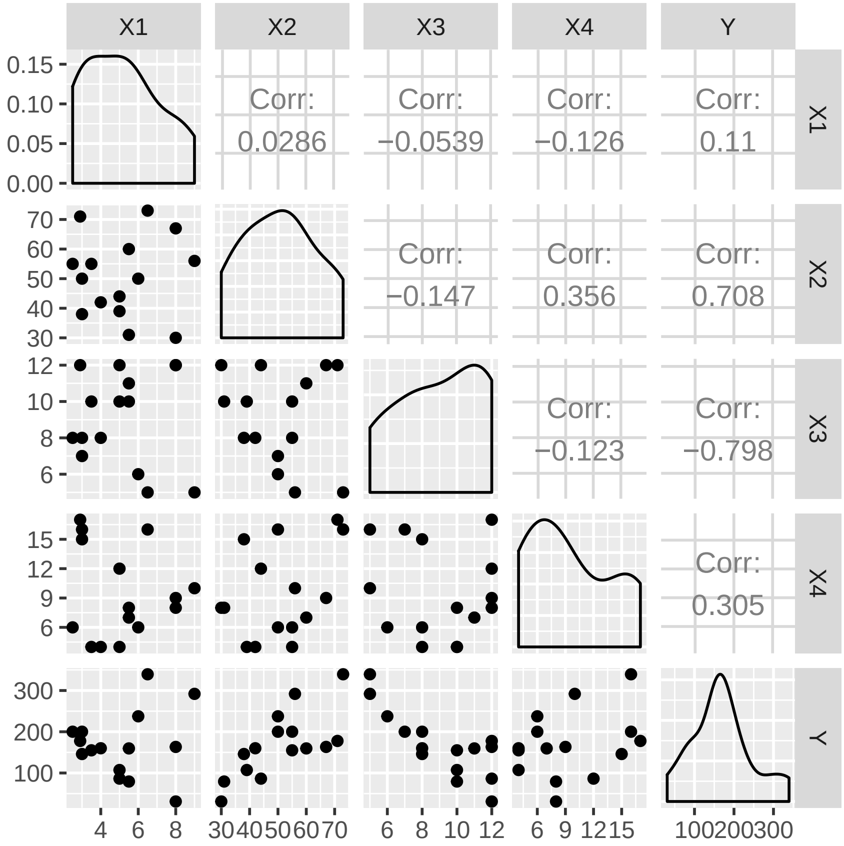
Solution

The first step is to read in the data

```r
library(tidyverse)
sales <- read_csv("sales.csv")
```

a) Now we produce a pairs plot of the data to get an idea about the relation between the response and the predictor variables.

```r
GGally::ggpairs(sales)
```



✓

We focus on the last line of plots, where we see the scatterplots of all predictors with the response.

`X1` We detect a weak positive relation between `X1` and the response. But it is doubtful if there is a linear dependence. On the other hand, the plot does not reflect any type of non-linear dependence.

`X2` There seems to be a strong positive linear relation between `X2` and the response. For large numbers of active accounts we see an increased variation.

X3 The number of competing brands has a strong negative linear relationship with the response. Again we have an increased variation for large values.

X4 The relation between X4 and the response is again positive, but much weaker than e.g. for X1. The assumption of a linear dependence could make sense in this case.

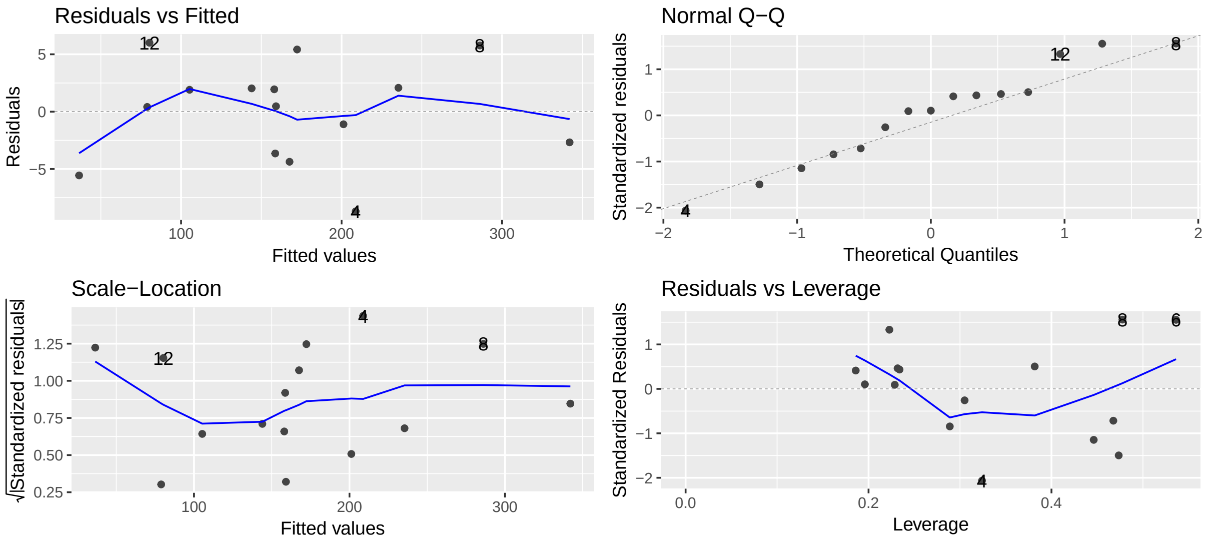
✓ (if just half of the conclusions are correct ✓ , if less zero points)

b) The next step is to fit the model using `lm()`. Since we want to fit a full model (using all predictors), we can use the . notation.

```
model_sales <- lm(Y ~ ., data = sales)
```

To use the `autoplot()` function, we have to load the `ggfortify` package. The package is used to transform the information contained in the `lm` object `model_sales` into a data frame (remember that all `ggplot2` function - and `autoplot()` is a `ggplot2` function - require a data frame as input).

```
library(ggfortify)
autoplot(model_sales)
```



✓

There seems to be no structural dependence between the residuals and the fitted values. The deviations for small and large fitted values are just due to the fact, that there is just one small and one rather large fitted value. For just 15 observations, the Normal QQ plot also looks fine. Of course, the (almost) ties in the residuals contradict the normality assumption, but the data was measured on a rather rough scale. ✓ Hence, we checked the model assumptions

2.) The mean of Y is linear in X. (first plot)

4.) For each X, the distribution of Y is normal. (second plot)

To check

3.) For each X, the distribution of Y has the same variance.

we take a look at the third plot (preferable, but also the first one could be used). The variation of the standardized residuals (which have an empirical standard deviation of 1) seems not to change for different fitted values. ✓

Thus, we do not see any strong violations of the above model assumptions. The assumption on independence can't be checked with these plots. We learn later how to do this.

Remark: Our findings are also confirmed by the following output

```
library(gvlma)
gvlma(model_sales)

##
## Call:
## lm(formula = Y ~ ., data = sales)
##
## Coefficients:
## (Intercept)           X1           X2           X3           X4
##    177.2286       2.1702       3.5380     -22.1583       0.2035
##
##
## ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
## USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
## Level of Significance =  0.05
##
## Call:
##  gvlma(x = model_sales)
##
##                      Value p-value                Decision
## Global Stat        1.0947  0.8951 Assumptions acceptable.
## Skewness           0.3018  0.5828 Assumptions acceptable.
## Kurtosis           0.2973  0.5856 Assumptions acceptable.
## Link Function      0.2418  0.6229 Assumptions acceptable.
## Heteroscedasticity 0.2539  0.6144 Assumptions acceptable.
```

See Pena, E.A. and Slate, E.H., Global Validation of Linear Model Assumptions, *J. American Statistical Association*, 101(473):341-354, 2006. for details on this assessment of the linear model assumptions.

c) (i) To test the first hypothesis, we just take a look at the summary.

```
summary(model_sales)

##
## Call:
## lm(formula = Y ~ ., data = sales)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6881 -3.1604  0.4714  2.0541  6.0053
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 177.2286     8.7874  20.169 1.98e-09 ***
## X1            2.1702     0.6737   3.221  0.00915 **
## X2            3.5380     0.1092  32.414 1.84e-11 ***
## X3          -22.1583     0.5454 -40.630 1.95e-12 ***
## X4            0.2035     0.3189   0.638  0.53760
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.119 on 10 degrees of freedom
## Multiple R-squared:  0.9971,Adjusted R-squared:  0.9959
## F-statistic: 851.7 on 4 and 10 DF,  p-value: 1.285e-12
```

We see, that given the remaining predictors, the district potential seems to have no significant effect on the sales, since we have a p-value of 0.5375986.

✓

(ii) To test the hypothesis $\beta_3 = \beta_4 = 0$, we need the residual sum of squares $SS(\widehat{\beta})$ of the full model and $SS(\widehat{\beta}_A)$, the residual sum of squares of the reduced model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon.$$

Therefore we fit the reduced model

```
model_sales_12 <- lm(Y ~ X1 + X2, data = sales)
```

The F-ratio is then equal to

```
additional_SS <- anova(model_sales_12)[3,2] - anova(model_sales)[5,2]
(F <- additional_SS / 2 / (anova(model_sales)[5,2] / (15 - 4 - 1)))

## [1] 833.8493
```

(see Problem C.5 for the use of `anova()`) which yields a p-value of

```
pf(F, df1 = 2, df2 = 15 - 4 - 1, lower.tail = FALSE)

## [1] 7.523679e-12
```

Hence, we reject the null hypothesis that the number of competing brands and the potential of the district jointly have no influence on the response.

✓

(iii) To test the null hypothesis $\beta_2 = \beta_3$, we need to fit a model with predictors `X1`, `X2+X3` and `X4`. The transformation `X2+X3` can either be done before the model fitting or inside the formula by using the function `I()` (needed since summation is a formula operation, see `?I`).

```
(model_sales_23 <- lm(Y ~ X1 + I(X2 + X3) + X4, data = sales))

##
## Call:
## lm(formula = Y ~ X1 + I(X2 + X3) + X4, data = sales)
##
## Coefficients:
## (Intercept)           X1   I(X2 + X3)           X4
##     -61.389        4.613        3.051        2.541
```

The rest is now analogous to part (ii). The F-ratio is equal to

```
additional_SS <- anova(model_sales_23)[4,2] - anova(model_sales)[5,2]
(F <- additional_SS / (anova(model_sales)[5,2] / (15 - 4 - 1)))

## [1] 2225.067
```

which yields a p-value of

```
pf(F, df1 = 1, df2 = 15 - 4 - 1, lower.tail = FALSE)

## [1] 4.420375e-13
```

Hence, we reject the null hypothesis that the slope of the number of active accounts is equal to the slope of the number of competing brands. This makes sense, since we detected a positive relation for the number of active accounts and a negative one for the number of competing brands. ✓

(iv) For the last test we take again a look at the summary of the full model. The test we need to perform is just the F-test to check the usefulness of the regression. We observe a test statistic value of 851.7197644, which yields a p-value of $1.2851172 \times 10^{-12}$. Hence, we also reject this hypothesis.

✓

d) To write a function, which performs the tests from part c), we just need to include all steps in part c) in the body of the function. Since we are allowed to use the degrees of freedom as input (wouldn't be necessary, since we have the fitted models as input), we just need to determine the index (number of estimated parameters) in the ANOVA table to extract the residual sum of squares.

```
test_linht <- function(model_red, model_full, df_add = 1, df_full = 10){
  # number of estimated parameters
  index_red <- length(coef(model_red))
  index_full <- length(coef(model_full))
  # in the second column and (number of estimated parameters) row of the anova
  # table, we find ss_res
  additional_SS <- anova(model_red)[index_red,2] - anova(model_full)[index_full,2]
  # test statistic
  F <- additional_SS/df_add / (anova(model_full)[index_full,2] / df_full)
  # using lower.tail = FALSE is equivalent to compute 1 - pf()
  p_value <- pf(F, df1 = df_add, df2 = df_full, lower.tail = FALSE)
  # return is not needed; by default the object in the last line is returned,
  # but by using return() it becomes more clear
  return(list(test_statistic = F, p_value = p_value))

}
```

✓ (if the function works correctly, see application below)

To check if the function works correctly, we consider again the test problem from part c) (ii)

```
test_linht(model_sales_12, model_sales, df_add = 2, df_full = 10)

## $test_statistic
## [1] 833.8493
##
```

```
## $p_value
## [1] 7.523679e-12
```

We get the same result as before.

✓

e) We fit the reduced model by using `update()`

```
model_sales_123 <- update(model_sales, ~ . - X4)
# equivalent to lm(Y ~ X1 + X2 + X2, data = sales)
```

and take a look at the estimated coefficients

```
coef(model_sales_123)

## (Intercept)          X1          X2          X3
##  178.520620    2.105547    3.562403  -22.187992
```

The fitted model says that the sales will increase by 2.1055467×10^5 dollars if the promotional expenditures increase by 1000 dollars, while `X2` and `X3` remain fixed.✓ They sales will increase by even 3.5624025×10^5 dollars, if the number of active accounts is increased by one, while `X1` and `X3` remain fixed. ✓ On the other hand, the sales will decrease by -2.2187992×10^6 dollars, if the number of competing brands is increased by one, while `X2` and `X3` remain fixed.✓

f) The prediction, together with the corresponding prediction interval, can be computed by using `predict()` (cf. the solution of Problem C.4). Hence, we just have to specify the new observation in a correct way. This has to be done using a data frame.

```
predict(model_sales_123, newdata = data.frame(X1 = 3, X2 = 45, X3 = 10),
        interval = "prediction")

##        fit      lwr      upr
## 1 123.2655 111.4556 135.0753
```

✓