Technichal University of Munich
Department of Mathematics

MA4401 Applied Regression, Homework problem 5

Prof. Donna Ankerst, Stephan Haug (January 09, 2018)

Problem H.5

In this problem we will examine data from 27 coral reef heads, Porites lobata (), located in the Great Barrier Reef. Risk and Sammarco (1991) () found that the density of the coral skeletons increases with distance from the Australian shore, due to differences in inshore and offshore environments. The data is contained in the file `coral_reefs.csv` , which can be found on moodle.

a) Produce a scatterplot of the data using `ggplot2` functions. Choose different colours for the different reefs. *(1 point)*

b) Fit polynomial models of increasing degree to the data. In each step test for a lack of fit of the current model. Stop if you can't reject the null hypothesis anymore. Store all fitted models in a list object. *(3 points)*

c) Perform a residual analysis for the last model (the one, which didn't show a lack of fit) from part b). Are there any outliers due to the Cook's D statistic?
Use the function `powerTransform()` from the `car` package to estimate the parameter λ of the Box-Cox transformation for this model. Find an appropriate R function to perform a Likelihood-Ratio test of the testproblem

$$H_0 : \lambda = 1 \qquad \text{vs.} \qquad H_1 : \lambda \neq 1 \, .$$

(*Hint:* `help(car::powerTransform)` will be helpful). Based on the test result, decide if a Box-Cox transformation should be applied. *(4 points)*

Remark: Use the function `autoplot()` for the residual analysis.

d) Compute the adjusted R^2 and the AIC for all models stored in the list from part b). For each criteria use the function `sapply()` to compute it for all models simultaneously. Do both criteria also favour the model analysed in part c)? *(2 points)*

Hint: Remember that the adjusted R^2 is contained in the summary of a linear model.

General grading instructions: 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.

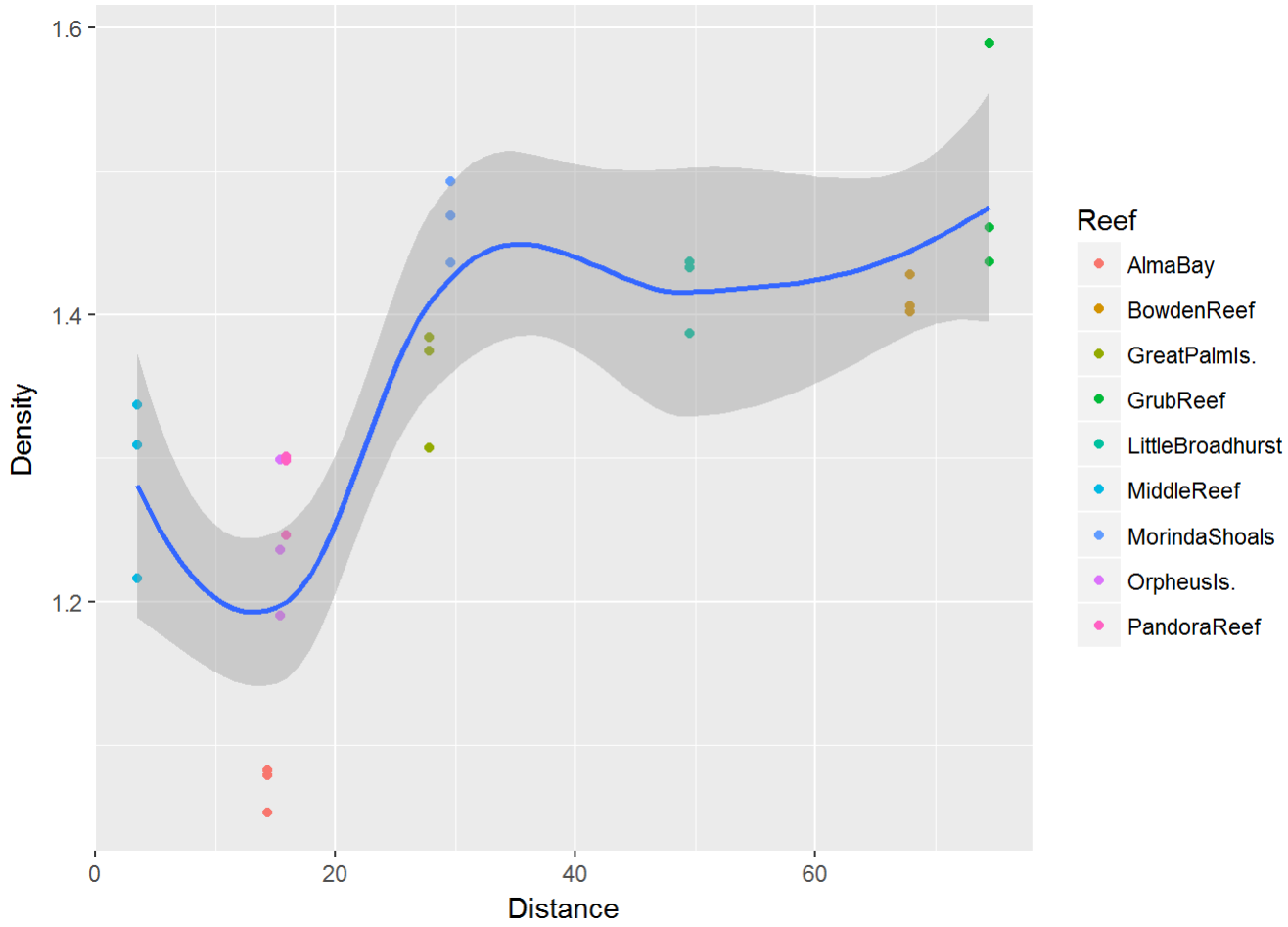
Solution

We start by reading in the data

```
library(tidyverse)
coral <- read_csv("coral_reefs.csv")
```

a) The first step is to decide which variable will be our response variable. The problem says that the density increases with the distance. Hence, distance will be the independent/predictor variable "influencing" the response variable density. Next we produce a scatterplot of the response `Density` against the predictor `Distance` and use `Reef` to colour the data points.

```
ggplot(coral, aes(x = Distance, y = Density)) + geom_point(aes(colour = Reef)) + geom
_smooth()
```



Grading: A scatterplot without colours gives **0.5 points**. *Colouring the points with respect to* `Reef` *gives another* **0.5 points**. *Using* `geom_smooth()` *was additionally and is therefore not needed.*

b) The aim is to fit polynomial models of increasing degree until we can't show that there is a lack of fit. We start with a model of degree one, so a simple linear regression model. The problem says, that we should store all models in a list. We initialise the list with the first model and then add further list elements.

1.

```
list_of_models <- list(lm(Density ~ Distance, data = coral))
```

Now we have to test if there is a lack of fit (there will be, if we look at the plot in part a)). From Problem C.12 we know that the test can be done with the `anova()` function. All we need is a model to compare with. The model needs to fit an unrestricted mean value for each distance group. Since each group is identified by the variable `Reef`, we can use the model

```
model_reef <- lm(Density ~ Reef, data = coral)
```

and obtain

```
anova(model_reef, list_of_models[[1]])
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Reef
## Model 2: Density ~ Distance
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     18 0.036908
## 2     25 0.240736 -7  -0.20383 14.201 3.665e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small, so we have to reject the null hypothesis $H_0 : \mu_i = \beta_0 + \beta_1 x_i$.

2. Since we rejected the null hypothesis in the last step, we fit now a polynomial degree two and compare it with model `model_reef`.

```
list_of_models[[2]] <- lm(Density ~ Distance + I(Distance^2), data = coral)

anova(model_reef, list_of_models[[2]])
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Reef
## Model 2: Density ~ Distance + I(Distance^2)
##   Res.Df      RSS Df Sum of Sq      F   Pr(>F)
## 1     18 0.036908
## 2     24 0.230964 -6  -0.19406 15.774 2.74e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small, so we have to reject the null hypothesis $H_0 : \mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$.

3. Since we rejected the null hypothesis in the last step, we fit now a polynomial degree three and compare it with model `model_reef`.

```
list_of_models[[3]] <- lm(Density ~ Distance + I(Distance^2)+ I(Distance^3), data = c
oral)

anova(model_reef, list_of_models[[3]])
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Reef
## Model 2: Density ~ Distance + I(Distance^2) + I(Distance^3)
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     18 0.036908
## 2     23 0.223814 -5  -0.18691 18.231 1.748e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small, so we have to reject the null hypothesis $H_0 : \mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3$.

4. Since we rejected the null hypothesis in the last step, we fit now a polynomial degree four and compare it with model `model_reef`.

```
list_of_models[[4]] <- lm(Density ~ Distance + I(Distance^2)+ I(Distance^3) +
                            I(Distance^4), data = coral)

anova(model_reef, list_of_models[[4]])
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Reef
## Model 2: Density ~ Distance + I(Distance^2) + I(Distance^3) + I(Distance^4)
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     18 0.036908
## 2     22 0.125908 -4    -0.089 10.851 0.0001176 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small, so we have to reject the null hypothesis
$H_0 : \mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4$.

5. Since we rejected the null hypothesis in the last step, we fit now a polynomial degree five and compare it with model `model_reef`.

```
list_of_models[[5]] <- lm(Density ~ Distance + I(Distance^2)+ I(Distance^3) +
                            I(Distance^4) + I(Distance^5), data = coral)

anova(model_reef, list_of_models[[5]])
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Reef
## Model 2: Density ~ Distance + I(Distance^2) + I(Distance^3) + I(Distance^4) +
##     I(Distance^5)
##   Res.Df      RSS Df Sum of Sq      F   Pr(>F)
## 1     18 0.036908
## 2     21 0.109582 -3 -0.072674 11.814 0.000164 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small, so we have to reject the null hypothesis
$H_0 : \mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \beta_5 x_i^5$.

6. Since we rejected the null hypothesis in the last step, we fit now a polynomial degree six and compare it with model `model_reef`.

```
list_of_models[[6]] <- lm(Density ~ Distance + I(Distance^2)+ I(Distance^3) +
                            I(Distance^4) + I(Distance^5) + I(Distance^6), data = cor
al)

anova(model_reef, list_of_models[[6]])
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Reef
## Model 2: Density ~ Distance + I(Distance^2) + I(Distance^3) + I(Distance^4) +
##     I(Distance^5) + I(Distance^6)
##   Res.Df      RSS Df Sum of Sq      F    Pr(>F)
## 1     18 0.036908
## 2     20 0.095052 -2 -0.058144 14.178 0.0002006 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is very small, so we have to reject the null hypothesis $H_0 : \mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \beta_5 x_i^5 + \beta_6 x_i^6$.

7. Since we rejected the null hypothesis in the last step, we fit now a polynomial degree seven and compare it with model `model_reef` .

```
list_of_models[[7]] <- lm(Density ~ Distance + I(Distance^2)+ I(Distance^3) +
                            I(Distance^4) + I(Distance^5) + I(Distance^6) +
                            I(Distance^7), data = coral)

anova(model_reef, list_of_models[[7]])
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Reef
## Model 2: Density ~ Distance + I(Distance^2) + I(Distance^3) + I(Distance^4) +
##     I(Distance^5) + I(Distance^6) + I(Distance^7)
##   Res.Df      RSS Df   Sum of Sq      F Pr(>F)
## 1     18 0.036908
## 2     19 0.037347 -1 -0.00043934 0.2143  0.649
```

Now we observe a p-value of 0.649 and hence, can't reject the null hypothesis

$$H_0 : \mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \beta_5 x_i^5 + \beta_6 x_i^6 + \beta_7 x_i^7$$

So, this model seems to show no lack of fit.

```
summary(list_of_models[[7]])
```

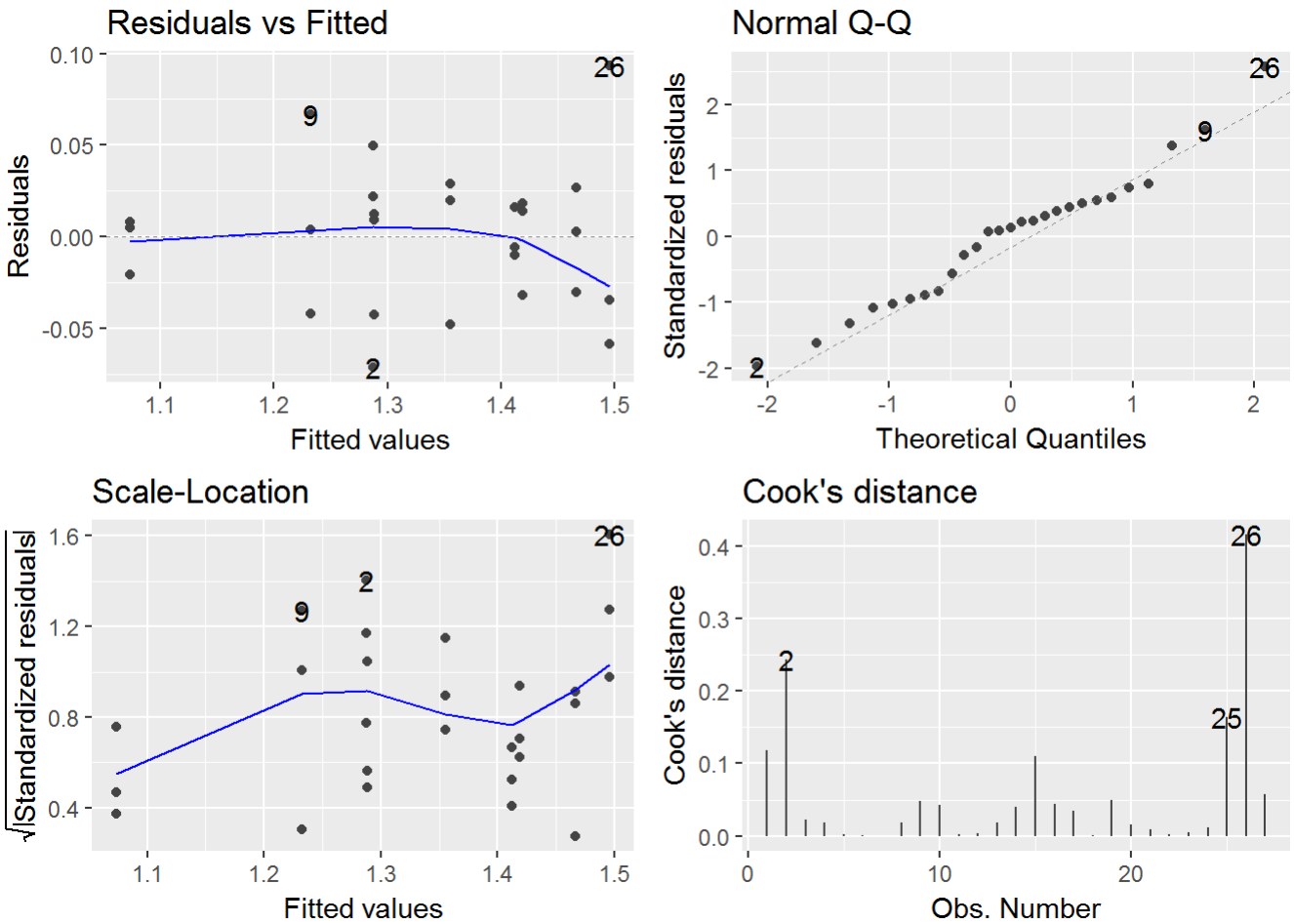
```
##
## Call:
## lm(formula = Density ~ Distance + I(Distance^2) + I(Distance^3) +
##     I(Distance^4) + I(Distance^5) + I(Distance^6) + I(Distance^7),
##     data = coral)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.071328 -0.031120  0.005066  0.018999  0.093334
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.871e+00  1.230e+00   7.213 7.54e-07 ***
## Distance      -3.597e+00  5.892e-01  -6.105 7.20e-06 ***
## I(Distance^2)  5.140e-01  8.620e-02   5.963 9.71e-06 ***
## I(Distance^3) -3.402e-02  5.850e-03  -5.815 1.33e-05 ***
## I(Distance^4)  1.187e-03  2.089e-04   5.683 1.77e-05 ***
## I(Distance^5) -2.237e-05  4.013e-06  -5.574 2.24e-05 ***
## I(Distance^6)  2.140e-07  3.901e-08   5.487 2.71e-05 ***
## I(Distance^7) -8.135e-10  1.501e-10  -5.418 3.15e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04434 on 19 degrees of freedom
## Multiple R-squared:  0.9181, Adjusted R-squared:  0.8879
## F-statistic: 30.43 on 7 and 19 DF,  p-value: 4.98e-09
```

The partial t-tests in the summary show that variables are significant.

Grading: If at least one polynomial model of degree greater than one was fitted in a correct way, one gets **1 point**. *If the lack-of-fit test was at least once performed in a correct way (this includes choosing the correct unrestricted reference model), one gets* **1 point**. *If the model with degree seven was identified as the first one showing no lack of fit (this includes giving a proper reason), one gets* **1 point**.

c) The residual analysis will be done using the output of `autoplot()`. To also show the Cook's D values, we will use the `which` argument (see e.g. the solution of C.9).

```
library(ggfortify)
autoplot(list_of_models[[7]], which = 1:4)
```



i. In the Residuals vs Fitted plot we do not see any obvious structure. The residuals spread around zero.
ii. The Normal Q-Q plot shows that empirical distribution of the residuals is close to normal distribution.
iii. In the Scale-Location plot we see a slightly inhomogeneous variation of the residuals. The variation tends to increase with the fitted values. So, the assumption of a constant variance might not be fulfilled.
iv. In the Cook's distance plot three observations (2, 25, 26) are marked, but none of them has value larger than 0.5. One might further analyse observation 26, since it seems to have a rather untypical value for the reef

```
coral$Reef[26]
```

```
## [1] "GrubReef"
```

But we won't do this.

Since the assumption of a constant variance seems not to be fulfilled, we will estimate the λ of a Box-Cox transformation in the next step.

```
library(car)
(pt_m7 <- powerTransform(list_of_models[[7]]))
```

```
## Estimated transformation parameters
##         Y1
## -0.4147315
```

We observe a negative value for the estimated lambda. But since we do not have that many observations, it is not clear if this value is significantly different from 1, which would imply that the response should not be transformed.

The hint said to look at the help file of `powerTransform()`. There one finds the function `testTransform()`, which can be used to test

$$H_0 : \lambda = 1 \qquad \text{vs.} \qquad H_1 : \lambda \neq 1 \, .$$

The input of the function is the object created by `powerTransform()` and the assumed value of lambda under the null hypothesis.

```
testTransform(pt_m7, 1)
```

```
##                             LRT df      pval
## LR test, lambda = (1) 1.014436  1 0.3138426
```

With a p-value of 0.3138426 we can't reject the null hypothesis. Hence, we won't transform the response, which is equivalent to applying the Box-Cox transformation with $\lambda = 1$. This is also implied by the rounded value

```
pt_m7$roundlam
```

```
## Y1
##  1
```

Remark: The test result is also obtained by just looking at the summary of `pt_m7`.

```
summary(pt_m7)
```

```
## bcPower Transformation to Normality
##    Est.Power Std.Err. Wald Lower Bound Wald Upper Bound
## Y1   -0.4147   1.3976           -3.154           2.3245
##
## Likelihood ratio tests about transformation parameters
##                               LRT df      pval
## LR test, lambda = (0) 0.08787662  1 0.7668941
## LR test, lambda = (1) 1.01443557  1 0.3138426
```

So this is an equivalent solution.

Grading: Discussing the four plots in a way similar to i.-iv. gives **2 points** *(so each argument 0.5 points). Estimating λ with `powerTransform()` gives* **1 point**. *Conducting the test in a correct way gives* **1 point**.

d) We start with computing the AIC for all seven models. The problem says to use `sapply()`, which allows us to apply a function (in this case `AIC()`) to all elements of a list. Since `AIC()` can be applied to `lm` object, which are the list elements, we just need to run

```
aic_values <- sapply(list_of_models, AIC)
```

Computing the adjusted R^2 in this way, is a bit more difficult, since there is no function to compute it. Hence, we have to define our own function. This can be done inside `sapply()` without creating a new object. The hint said, that the adjusted R^2 is contained in the summary. Therefore the function will compute the summary and from the output we then just extract the adjusted R^2.

```
adjR2_values <- sapply(list_of_models, function(x) summary(x)$adj.r.squared)
```

Now we take a look at the values

```
adjR2_values
```

```
## [1] 0.4509476 0.4512870 0.4451551 0.6736804 0.7024694 0.7290151 0.8879225
```

```
aic_values
```

```
## [1] -44.81436 -43.93324 -42.78230 -56.31441 -58.06417 -59.90476 -83.12724
```

and see that the adjusted R^2 increases with the polynomial degree, while the AIC decreases. This means that based on those two criteria, we also would have chosen the model with polynomial degree seven.

Grading: Computing the values for the adjusted R^2 and the AIC gives **1 point**. *If the values are computed one after the other without using* `sapply()` *one just gets 0.5 points instead of 1 point*. *Concluding that the model with seven degrees is favoured by both criteria gives* **1 point**.