Prof. Dr. E. Cramer, Prof. Dr. M. Kateri,
L. Kaufmann, M.Sc., T. van Bentum, M.Sc. ,
A. Müllenmeister

RWTH Aachen, SS 2021
Release date: May 4th, 2021
Solution: May 11 th, 2021

Applied Data Analysis

R-Laboratory 4

Model Parsimony – Testing in Linear Models – Analysis of Variance

Useful packages and functions:

- `set.seed()`
- `runif()`
- `lm()`
- `shapiro.test()`
- `rgl`
- `rgl::plot3d()`
- `rgl::planes3d()`
- `summary()`
- `tapply()`
- `car`
- `car::leveneTest()`
- `aov()`
- `TukeyHSD()`
- `par()`
- `cooks.distance()`
- `anova()`

Task 13

For values of x in $[0, 100]$, suppose the linear model $(Y \mid X = x) \sim \mathcal{N}(\mu(x), \sigma^2)$ holds with

$$\mathrm{E}(Y \mid X = x) = \mu(x) = 45 + 0.1x + 5 \cdot 10^{-4}x^2 + 5 \cdot 10^{-7}x^3 + 5 \cdot 10^{-11}x^4 + 5 \cdot 10^{-13}x^5$$

and $\sigma = 10$.

(a) Set a seed to 2020 and initialize two numeric vectors `vec.delta.simple` and `vec.delta.correct` of length 100.

(b) Repeat the following procedure 100 times using a loop.

(i) Generate 25 observations from the model with X uniformly distributed on $[0, 100]$.

(ii) Fit a "simple" model with $\mu^{(0)}(x) = \beta_0 + \beta_1 x$ and afterwards the "correct" model with $\mu^{(1)}(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \beta_5 x^5$.

(iii) For the first five iterations construct plots showing the data, the true relationship, and both model fits.

(iv) For each model, summarize the quality of the model fit by the mean

$$\Delta_j = \frac{1}{25} \sum_{i=1}^{25} |\hat{\mu}^{(j)}(x_i) - \mu(x_i)|, \qquad j \in \{0, 1\}. \qquad (+)$$

Store the values of Δ_j, $j = 0, 1$, in (+) in the two numeric vectors `vec.delta.simple` and `vec.delta.correct`.

(c) By comparing the values of Δ_j, $j = 0, 1$, in the vectors vectors `vec.delta.simple` and `vec.delta.correct` and the plots of (iii), which model do you prefer? Explain what this Task illustrates about model parsimony.

Task 14

(a) Load the `.RData` file of the pre-processed data of *Survey1* (Task 7) into the R workspace.

(b) Create a regression model with the approach

$$\texttt{DimSelf} = d + a\ \texttt{DimEmotion} + b\ \texttt{DimBody} \qquad (++)$$

for parameters $a, b, d \in \mathbb{R}$.

(c) Analyze, if the model assumptions are sufficiently satisfied:

(i) Create the following plots (you can directly apply the function `plot` to the fitted model for the first four plots) and interpret them:

i. Residuals versus fitted values
ii. $\sqrt{|\text{Standardized residuals}|}$ versus fitted values
iii. Quantiles of the standardized residuals versus the expected quantiles of the standard normal distribution (called QQ-Plot)
iv. Standardized residuals versus leverage
v. Cook's distance.

(ii) Test on level $\alpha = 0.05$, if there is evidence against the assumption of normally distributed residuals.

Hint: Use a Shapiro-Wilk-Test with the R function `shapiro.test`.

(d) Create a 3d scatterplot with `DimEmotion` on the x-axis, `DimBody` on the y-axis and `DimSelf` on the z-axis. Add the regression surface of the model (++) to the plot.

Hint: Use the functions `plot3d` and `planes3d` from the package `rgl`.

(e) In the model (++) test the hypotheses

$$H_0 : b = 0 \quad \text{versus} \quad H_1 : b \neq 0$$

on the significance level $\alpha = 0.05$. Is the null hypothesis rejected?

Task 15

(a) Load the `.RData` file of the pre-processed data of *Solar* (Task 10) into the R workspace.

(b) Create four boxplots for the attribute `Pmax`, one for each batch.

(c) Carry out an analysis of variance for the attribute `Pmax` regarding the factor `batch`.

(d) Analyze the model assumptions as in Task 14 (c). Additionally, test on level $\alpha = 0.05$, if there is evidence against the assumption of equal variance in each group of factor level.

Hint: Use a Levene-Test with the R function `leveneTest` from the package `car` to test the equality of the variances.

(e) Test on the significance level $\alpha = 0.05$, if the null hypothesis of equal means of `Pmax` for the four batches is rejected.

Hint: You may call the function `summary` with the fitted model as argument.

(f) Carry out a pairwise comparison of the batches regarding `Pmax` using a Tukey-Test on significance level $\alpha = 0.1$. Create a plot of the computed confidence intervals.

Task 16

(a) Consider the data set `ToothGrowth` from the package `datasets` and transform the attribute `dose` to type `factor`.

(b) Create a boxplot for the value of `len`, separated on all factor combinations of `supp` and `dose`. What is your impression?

(c) Fit a linear model for `len`, where `supp` and `dose` (inclusive interaction) are explanatory variables.

(d) Analyze the model assumptions as in Task 14 (c). Additionally, test on level $\alpha = 0.05$, if there is evidence against the assumption of equal variance in each group of factor combination.

(e) If the model assumptions are sufficiently satisfied, test on level $\alpha = 0.05$, if there is any influence of the explanatory variables on the value of `len`.

(f) If there is any influence of explanatory variables, analyze it concretely: Test on overall $\alpha = 0.05$, if the interaction of `supp` and `dose` and possibly their main effect has an influence on the value of `len`.

Hint: Use a Bonferroni correction to ensure that all tests together satisfy the significance level α. This means that you can compare each p-value with an adapted significance level $\tilde{\alpha} = \frac{\alpha}{k}$, where $k \in \mathbb{N}$ is the number of tests.

(g) If some effects are not significantly different from zero, fit a new model with the removed terms to get the final model for interpretation.

```
#############
#
# Task 13
#
#############

# (a)
set.seed(2020)

# (b)
N=25
vec.delta.simple=rep(0,100)
vec.delta.correct=rep(0,100)
vec.delta.correct.poly=rep(0,100)
for(i in 1:100){
  x=runif(N,0,100)

  mu=45+0.1*x+0.0005*x^2+5e-7*x^3+5e-11*x^4+5e-13*x^5
  y=mu+rnorm(N,sd=10)
  model.correct=lm(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5))

  # better use poly
  model.correct.poly=lm(y~poly(x,degree=5))
  fitted.vals.poly=predict(model.correct.poly,newdata=data.frame(x=x))

  model.simple=lm(y~x)

  if(i<6){
    # data
    plot(x,y)
    param1=model.correct$coefficients
    param2=model.simple$coefficients
    # true relationship
    curve(45+0.1*x+0.0005*x^2+5e-7*x^3+5e-11*x^4+5e-13*x^5,add=TRUE,col="blue")
    # correct model
    curve(param1[1]+param1[2]*x+param1[3]*x^2+param1[4]*x^3+param1[5]*x^4+param1[6]*x^5,add=TRUE,col="red")
    # simple model
    curve(param2[1]+param2[2]*x,add=TRUE,col="green")
    # correct model fitted with poly
    x.grid=seq(min(x),max(x),length.out = 100)
    y.pred=predict(model.correct.poly,newdata=data.frame(x=x.grid))
    lines(x.grid,y.pred, col="orange")
  }
  vec.delta.correct[i]=mean(abs(model.correct$fitted.values-mu))
  vec.delta.correct.poly[i]=mean(abs(fitted.vals.poly-mu))
  vec.delta.simple[i]=mean(abs(model.simple$fitted.values-mu))
}
```

```r
#############
#
# Task 14
#
#############
library(rgl)

# (a)
load("Survey1.RData")

# (b)
model.survey=lm(DimSelf~DimEmotion+DimBody, data = data.survey)

# (c)
# check the fit of the model
par(mfrow=c(2,2))
plot(model.survey)
par(mfrow=c(1,1))
plot(cooks.distance(model.survey))

# test of normality
shapiro.test(model.survey$residuals)

# (d)
plot3d(x=data.survey$DimEmotion,y=data.survey$DimBody,z=data.survey$DimSelf,xlab="DimEmotion",ylab="DimBody",zlab="DimSelf")
planes3d(a=model.survey$coefficient[2],b=model.survey$coefficient[3],c=-1,d=model.survey$coefficients[1],alpha=0.5)

# (e)
summary(model.survey)
```

```
boxplot(solar$Pmax ~ solar$batch)

# (c)
# analysis of variance
solar.aov = aov(solar$Pmax ~ solar$batch)

# (d)
# check the fit of the model
par(mfrow=c(2,2))
plot(solar.aov)
par(mfrow=c(1,1))
plot(cooks.distance(solar.aov))

#test of normality
shapiro.test(residuals(solar.aov))

#Levene-Test of equal variances
leveneTest(solar.aov)

# alternative: Levene-Test "by hand"
Median.Grp = tapply(solar$Pmax,solar$batch,median)
Z = abs(solar$Pmax - Median.Grp[solar$batch])
summary(aov(Z ~ solar$batch))

# (e)
summary(solar.aov)

# (f)
# pairwise comparison with Tukey-Test
# since the model is balanced (each group of a factor level has the same length)
solar.Tuk = TukeyHSD(solar.aov,conf.level=0.9)
solar.Tuk

# plot of the computed confidence intervals
plot(solar.Tuk)

#############
#
# Task 16
#
#############

# (a)
ToothGrowth
len=ToothGrowth$len
supp=ToothGrowth$supp
dose=as.factor(ToothGrowth$dose)

# (b)
boxplot(len~supp*dose)

# (c)
model=lm(len~supp*dose)

# (d)
# check the fit of the model
par(mfrow=c(2,2))
plot(model)
par(mfrow=c(1,1))
plot(cooks.distance(model))

shapiro.test(residuals(model))

leveneTest(model)

# (e)
summary(model)

# (f)
anova(model)

# (g)
model.new=lm(len~dose+supp)
```