MAST90104: A First Course in Statistical Learning

Week 7 Practical and Workshop - Solution

1 Practical questions

1. The data *teengamb* from the `faraway` package contains data from a survey in Britain to study teenage gambling. The variables are

   - **sex**: 0=male, 1=female
   - **status**: Socioeconomic status score based on parents' occupation income in pounds per week
   - **verbal**: verbal score in words out of 12 correctly defined
   - **gamble**: expenditure on gambling in pounds per year

   We can import the data by

   ```
   data(teengamb)
   ```

   We are interested in predicting **gamble** using the other variables. Implement the following variable selection methods to determine the "best" model.

   (Note that here we are ignoring the fact gender can be treated as factor)

   (a) Backward elimination

   Solution

   ```
   > install.packages('faraway')
   > library(faraway)
   > data(teengamb)
   > nullmodel <- lm(gamble ˜ 1 ,data = teengamb)
   > fullmodel <- lm(gamble˜., data = teengamb)
   > # backward selection
   > drop1(fullmodel,scope= ˜ ., test="F")
   Single term deletions

   Model:
   gamble ˜ sex + status + income + verbal
          Df Sum of Sq   RSS    AIC F value    Pr(>F)
   <none>              21624 298.18
   sex     1    3735.8 25360 303.67  7.2561   0.01011 *
   status  1      17.8 21642 296.21  0.0345   0.85349
   income  1   12056.2 33680 317.00 23.4169 1.792e-05 ***
   verbal  1     955.7 22580 298.21  1.8563   0.18031
   ---
   Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
   > model2 <- lm(gamble ˜ sex + income + verbal,data = teengamb)
   > drop1(model2, scope = ˜.,test = 'F')
   Single term deletions

   Model:
   gamble ˜ sex + income + verbal
          Df Sum of Sq   RSS    AIC F value    Pr(>F)
   <none>              21642 296.21
   sex     1    5787.9 27429 305.35 11.5001  0.001502 **
   income  1   13236.1 34878 316.64 26.2990 6.644e-06 ***
   verbal  1    1139.8 22781 296.63  2.2646  0.139667
   ```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> model3 <- lm(gamble ~ sex + income,data = teengamb)
> drop1(model3, scope = ~., test = 'F')
Single term deletions

Model:
gamble ~ sex + income
       Df Sum of Sq   RSS    AIC F value    Pr(>F)
<none>              22781 296.63
sex     1    5227.3 28009 304.34  10.096  0.002717 **
income  1   15309.8 38091 318.79  29.569 2.245e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> # stop
```

(b) Forward selection

Solution:

```
> # forward selection
> add1(nullmodel,scope = ~. +sex + status + income + verbal,test= 'F')
Single term additions

Model:
gamble ~ 1
       Df Sum of Sq   RSS    AIC F value    Pr(>F)
<none>              45689 325.34
sex     1    7598.4 38091 318.79  8.9766  0.004437 **
status  1     116.2 45573 327.22  0.1147  0.736438
income  1   17680.9 28009 304.34 28.4070 3.045e-06 ***
verbal  1    2212.5 43477 325.00  2.2900  0.137202
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> model2 <- lm(gamble~income,data = teengamb)
> add1(model2, scope = ~. +sex + status  + verbal,test= 'F')
Single term additions

Model:
gamble ~ income
       Df Sum of Sq   RSS    AIC F value   Pr(>F)
<none>              28009 304.34
sex     1    5227.3 22781 296.63 10.0960 0.002717 **
status  1     719.8 27289 305.11  1.1605 0.287228
verbal  1     579.1 27429 305.35  0.9290 0.340385
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> model3 <- lm(gamble ~ sex + income, data = teengamb)
> add1(model3, scope = ~.  + status  + verbal,test= 'F')
Single term additions

Model:
gamble ~ sex + income
       Df Sum of Sq   RSS    AIC F value Pr(>F)
<none>              22781 296.63
status  1    201.82 22580 298.21  0.3843 0.5386
verbal  1   1139.78 21642 296.21  2.2646 0.1397
```

(c) Stepwise selection

Solution

```
> # stepwise selection
> step(nullmodel, scope = ~. + sex + status + income + verbal)
Start:  AIC=325.34
gamble ~ 1

         Df Sum of Sq   RSS    AIC
+ income  1   17680.9 28009 304.34
+ sex     1    7598.4 38091 318.79
+ verbal  1    2212.5 43477 325.00
<none>                45689 325.34
+ status  1     116.2 45573 327.22

Step:  AIC=304.34
gamble ~ income

         Df Sum of Sq   RSS    AIC
+ sex     1    5227.3 22781 296.63
<none>                28009 304.34
+ status  1     719.8 27289 305.11
+ verbal  1     579.1 27429 305.35
- income  1   17680.9 45689 325.34

Step:  AIC=296.63
gamble ~ income + sex

         Df Sum of Sq   RSS    AIC
+ verbal  1    1139.8 21642 296.21
<none>                22781 296.63
+ status  1     201.8 22580 298.21
- sex     1    5227.3 28009 304.34
- income  1   15309.8 38091 318.79

Step:  AIC=296.21
gamble ~ income + sex + verbal

         Df Sum of Sq   RSS    AIC
<none>                21642 296.21
- verbal  1    1139.8 22781 296.63
+ status  1      17.8 21624 298.18
- sex     1    5787.9 27429 305.35
- income  1   13236.1 34878 316.64

Call:
lm(formula = gamble ~ income + sex + verbal, data = teengamb)

Coefficients:
(Intercept)       income          sex       verbal
     24.139        4.898      -22.960       -2.747
```

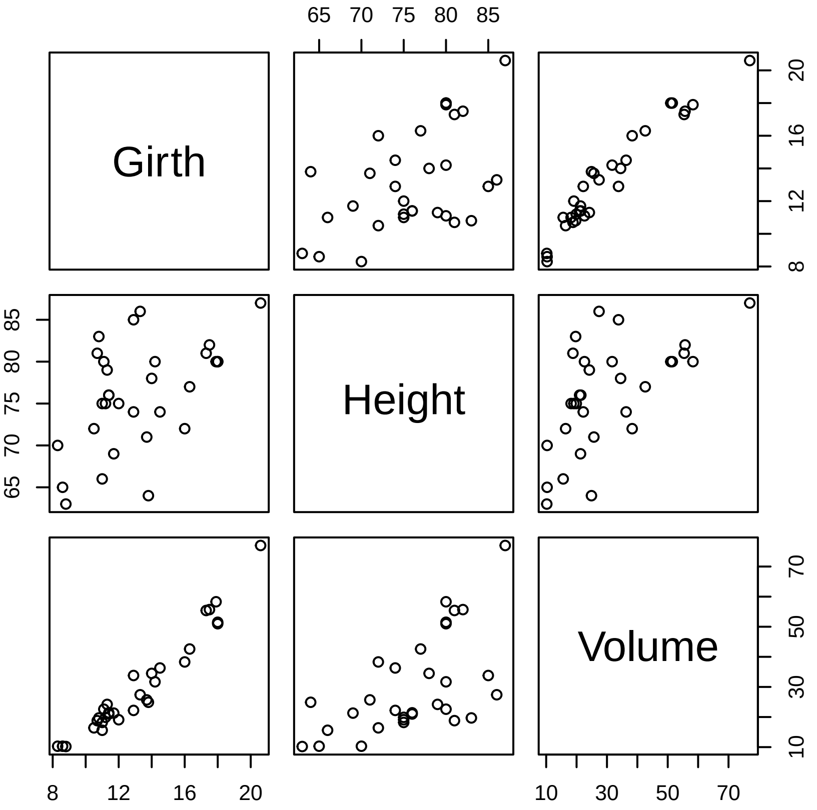
Comment on the AIC of the final model chosen by the 3 methods.

Solution: Backward and forward selection chose the same variables (`sex` and `income`) while stepwise selection final model has one more parameters, `verbal`. The AIC of the 2 models are very close to each other (296.63 vs 296.21).

2. Load and examine the dataset `trees` using

```
data(trees)
?trees
```

```
pairs(trees)
```



We will model the volume of a black cherry tree as a function of its girth and height.

(a) By calculating $R(\gamma_1|\gamma_2)$ and SS_{Res} from the data $\mathbf{y}$ and design matrix X, use an F test to determine if including the variable `Height` significantly improves the model fitted using only `Girth` (and an intercept).

Solution

```
> data(trees)
> # a.
> y <- trees$Volume
> n <- length(y)
> # model with both Girth and Height
> X <- cbind(1, trees$Girth, trees$Height)
> betahat <- solve(t(X) %*% X, t(X) %*% y)
> SS_res <- sum((y - X %*% betahat)^2)
[1] 421.9214
> SS_reg <- sum((X %*% betahat)^2)
> # model with only Girth + intercept
> X2 <- X[,-3]
> betahat2 <- solve(t(X2) %*% X2, t(X2) %*% y)
> SS_reg2 <- sum((X2 %*% betahat2)^2)
> R_g1g2 <- SS_reg - SS_reg2
[1] 102.3812
> Fstat <- (R_g1g2/1)/(SS_res/(n - 3))
[1] 6.79433
> pf(Fstat, 1, n - 3, lower.tail = F)
[1] 0.01449097
```

Repeat the test using the lm and anova commands, to see if you get the same numbers.

```
> model1 <- lm(Volume ~ Girth, data = trees)
> model2 <- lm(Volume ~ Girth + Height, data = trees)
> anova(model1, model2)
Analysis of Variance Table

Model 1: Volume ~ Girth
Model 2: Volume ~ Girth + Height
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     29 524.30
2     28 421.92  1    102.38 6.7943 0.01449 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(b) Add variables Girth squared and Girth squared times Height to the model, then use stepwise selection to simplify the model. (You can use step for this step.). Comment on the form of your final model.

Solution

```
> trees$GirthSq <- trees$Girth^2
> model <- lm(Volume ~ Girth + Height + GirthSq + GirthSq*Height, data = trees)
> model <- step(model, scope = ~ .)
Start:  AIC=64.36
Volume ~ Girth + Height + GirthSq + GirthSq * Height

                 Df Sum of Sq    RSS    AIC
- Girth           1    0.2288 179.27 62.402
- Height:GirthSq  1    6.9694 186.01 63.547
<none>                        179.04 64.363

Step:  AIC=62.4
Volume ~ Height + GirthSq + Height:GirthSq

                 Df Sum of Sq    RSS    AIC
<none>                        179.27 62.402
+ Girth           1     0.229 179.04 64.363
- Height:GirthSq  1    40.164 219.44 66.669
```

Note that R will not attempt to drop GirthSq and Height if the interaction term GirthSq*Height is still in the model.

(c) Use diagnostic plots to check the fit of your final model.

(d) What transformation might be indicated from the plot of residuals versus fitted values? Transform all variables with this transformation. What might the appropriate model be? Fit it and comment on the resulting residuals.

Solution

The plot of residuals versus fitted values show a fan-shape pattern, signalling that the errors do not have constant variance.We may want to transform the response variable. The Scale-Location plot show some increase in $\sqrt{|\text{Standardized residuals}}$ as fitted values increases, so a log transformation may be appropriate.

We consider 2 models: one with Height and Girth, and one where we also apply a log transformation to Height and Girth. The diagnostics plots of the 2 models (Figure 2 and Figure 3) suggest the log-log model is better. The residuals plot in model3 looks slightly better than the initial model, however there is still some heteroskedasticity shown in the plot.

```
> model2 <- lm(log(Volume) ~ Height + Girth, data = trees)
> model3 <- lm(log(Volume) ~ log(Height) + log(Girth), data = trees)
```

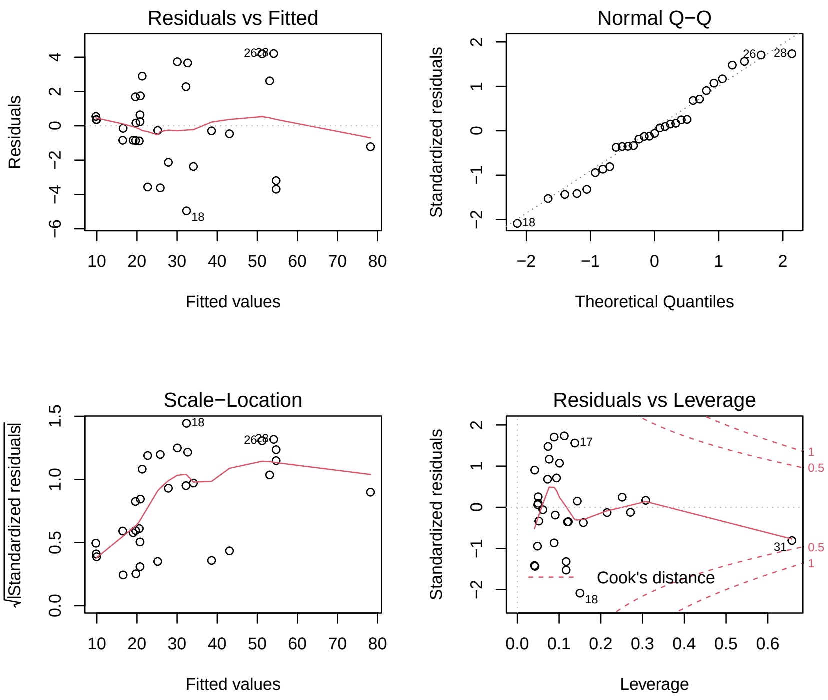


Figure 1: Diagnostic plots of the final model

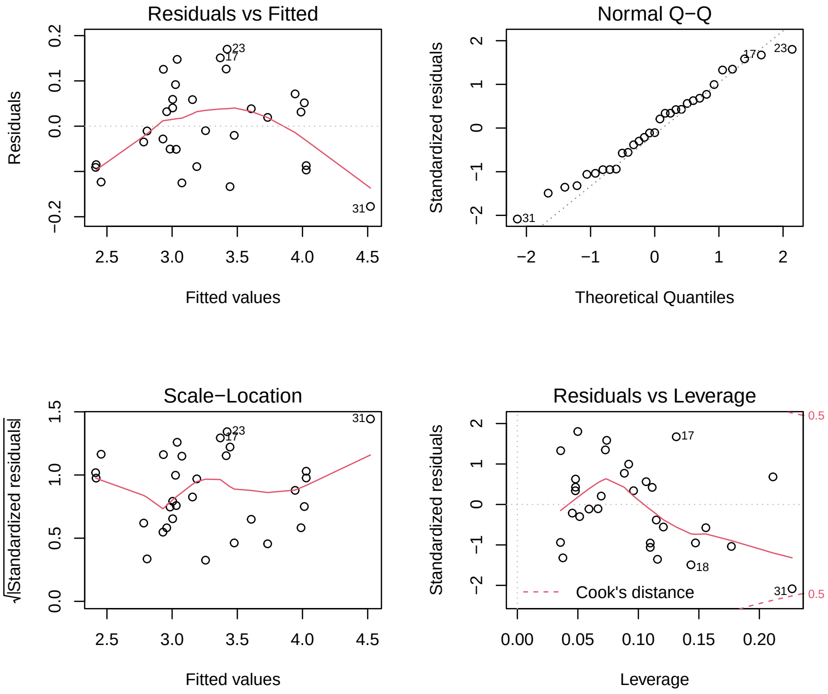


Figure 2: Diagnostic plots of `model2`

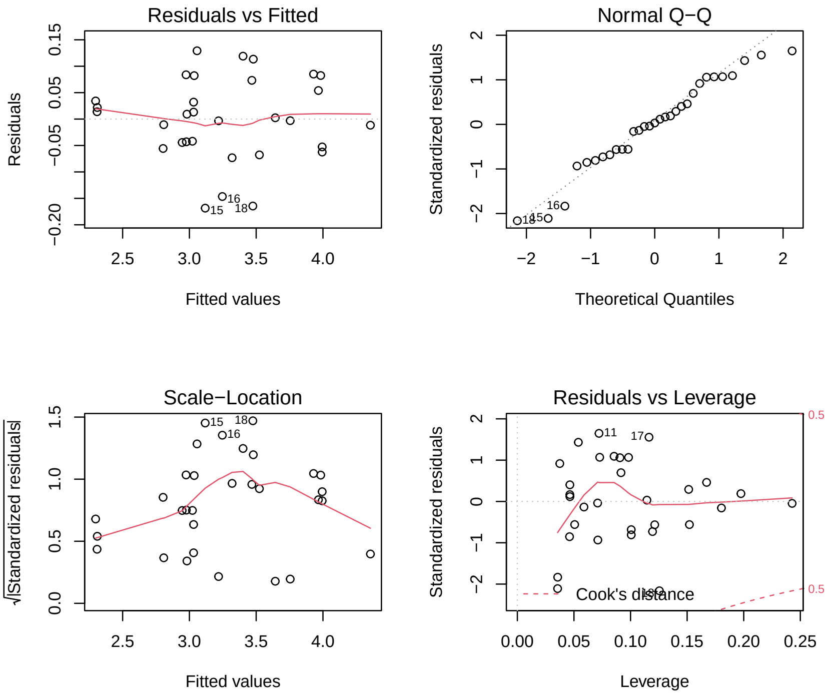


Figure 3: Diagnostic plots of `model3`

3. In a manufacturing plant, filters are used to remove pollutants. We are interested in comparing the lifespan of 5 different types of filters. Six filters of each type are tested, and the time to failure in hours is given in the dataset (on the website) `filters` (in `csv` format).

   (a) Use the `read.csv` function to read the data. Then convert the `type` component into a factor.
   Solution:

```
> filters <- read.csv("data sets/data/filters.csv")
> filters$type <- factor(filters$type)
```

   (b) Using only the `matrix` command, construct a **y** vector and a full rank X for this linear model, corresponding to `cont.treatment`
   Solution:

```
> y <- filters$life
> X.treatment <- matrix(0,30,5)
> X.treatment [,1] <- 1
> # first type is baseline
> for (i in 2:5) { X.treatment [filters$type==i,i] <- 1 }
```

   (c) Fit the models and compare with the `lm` output.
   Solution:

```
> XtXinv <- solve(t(X.treatment )%*%X.treatment )
> solve(t(X.treatment )%*%X.treatment )%*%t(X.treatment )%*%y
          [,1]
[1,] 249.16667
[2,] -61.66667
[3,] -83.16667
[4,] 108.16667
[5,] 112.16667
```

```
> model <- lm(y~type, data=filters)
> betahat <- model$coefficients
(Intercept)       type2       type3       type4       type5
  249.16667   -61.66667   -83.16667   108.16667   112.16667
```

Both methods give same the same estimates.

(d) Calculate s^2 using the residuals

Solution:

```
> s2 <- sum(lm(y~type, data=filters)$residuals^2)/25
[1] 15304.2
```