Math 423
Linear Regression

Homework IV

Frédéric Boileau

Prof. David A. Stephens

2nd December 2016

1

a

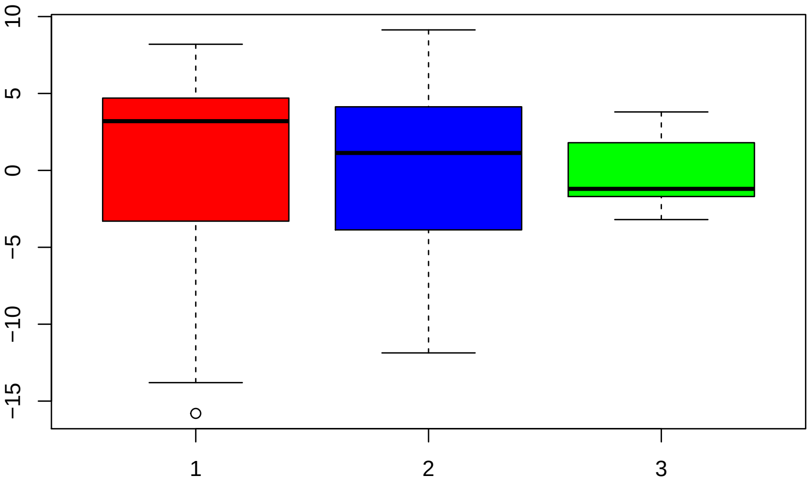
```
setwd(dirname(rstudioapi::getActiveDocumentContext()$path))

## Error:  RStudio not running

testScores = read.csv("TestScores.csv",header=TRUE)
testScores$Faculty = as.factor(testScores$Faculty)
yT = testScores$Score
faculty = testScores$Faculty
```

Now that we have converted the qualitative data's class into factors we can fit the model using lm. This is a model with only one predictor with three levels. Moreover we directly sum up the results in a boxplot to get an idea of the fit by quick inspection.

```
fit.yT = lm(yT ~ faculty)
clrs = c("red", "blue", "green")
boxplot(residuals(fit.yT) ~ faculty, vertical = TRUE, col = clrs)
```



From the onset we see that there is a wide disparity of variance between the first two faculties and the third. This is especially true when compared with the means. To take at closer look at the situation let's run an anova

```
anova(fit.yT)

## Analysis of Variance Table
##
## Response: yT
##           Df Sum Sq Mean Sq F value    Pr(>F)
## faculty    2 1529.4  764.69  20.016 7.843e-07 ***
## Residuals 42 1604.5   38.20
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We get a very small p-value for the regression so we conclude there is a statistically significant difference between test score for students from the 3 different faculties.

```
library(lsmeans)

## Warning:  package ’lsmeans’ was built under R version 3.3.2
## Loading required package:  estimability
## Warning:  package ’estimability’ was built under R version 3.3.2

means.yT = lsmeans(fit.yT, ~ faculty)
print(means.yT)

##  faculty   lsmean       SE df lower.CL upper.CL
##  1       35.80000 1.595894 42 32.57936 39.02064
##  2       35.86667 1.595894 42 32.64602 39.08731
##  3       48.20000 1.595894 42 44.97936 51.42064
##
## Confidence level used: 0.95
```

The means of the test scores of students in differentiated by faculty and their respective standard errors are displayed above. The means with their standard errors beeing in the second and third columns respectively.

2

a

As for the preceding section the first thing we do is import the data and format it appropriately

```
filter = read.csv("Filter.csv")
filter$carsize = as.factor(filter$carsize)
filter$type = as.factor(filter$type)
yF = filter$noise
type = filter$type
carsize = filter$carsize
```

Now we have 2 predictors and we want to fit the five possible models as liste in page 1. We have to type of predictors; factor predictors and interactions of factor predictors. Consequently interactions are counted as predictors in their own right. So a model with two predictors and their interaction has 4 parameters including the intercept. We now fit the 5 possible models

```
fit.nothing = lm(yF ~ 1)
fit.carsize = lm(yF ~ carsize, data = filter)
fit.type = lm(yF ~ type)
fit.both = lm(yF ~ carsize + type)
fit.interaction = lm(yF ~ carsize + type +carsize:type)
```

To have a table displaying the residual sum of squares of the 5 different models and the number of parameters for each model we can simply call the anova function with the models as arguments to get in one function call all the SSres.

```
a.1 = anova(fit.nothing,fit.type, fit.carsize, fit.both, fit.interaction)
print(a.1)
```

```
## Analysis of Variance Table
##
## Model 1: yF ~ 1
## Model 2: yF ~ type
## Model 3: yF ~ carsize
## Model 4: yF ~ carsize + type
## Model 5: yF ~ carsize + type + carsize:type
##   Res.Df     RSS Df Sum of Sq        F    Pr(>F)
## 1     35 29874.3
## 2     34 28818.1  1    1056.2  16.1465 0.0003631 ***
## 3     33  3822.9  1   24995.1 382.0913 < 2.2e-16 ***
## 4     32  2766.7  1    1056.2  16.1465 0.0003631 ***
## 5     30  1962.5  2     804.2   6.1465 0.0057915 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model	SS_{res}	p
1	2.9874306×10^4	1
1 + type	2.8818056×10^4	2
1 + carsize	3822.9166667	3
1 + type + carsize	2766.6666667	4
1 + type + carsize + type*carsize	1962.5	6

b

We now compare the "reduced" model which only considers the predictor "carsize" to the one that also includes the "type" predictor. When we include the main effect of a predictor we include all possible interactions as standard practice as to not make arbitrarily determined levels relevant (also vice-versa).

```
a.2 = anova(fit.carsize,fit.interaction)
print(a.2)

## Analysis of Variance Table
##
## Model 1: yF ~ carsize
## Model 2: yF ~ carsize + type + carsize:type
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     33 3822.9
## 2     30 1962.5  3    1860.4 9.4798 0.0001461 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now to compute manually the p value of the partial F test we need to determine the degrees of freedom of the numerator (r) and the denominator(n-p). For the full model we have to two type of parameters; $p_1 = M_{carsize} + M_{type} - 1 = 3 + 2 - 1 = 4$ for the main effects and $p_2 = (M_{carsize} - 1)(M_{type} - 1) = 2$ for the interections. Adding up we get $p = p_1 + p_2 = 6$. The reduced model has one factor variable with 3 levels so 3 parameters. The difference in both is 3 so $r = 3$ and

```
r = 3
n = 36
p = 6
df1 =r
df2 = n-p
num1 = sum(residuals(fit.carsize)^2)/ df1
num2 = sum(residuals(fit.interaction)^2) / df1
num = num1 - num2
den=  (sum(residuals(fit.interaction)^2)/(df2))
f_stat = num/den
p_value = 1 - pf(f_stat,df1,df2)
print(f_stat)

## [1] 9.47983

print(p_value)

## [1] 0.0001460971
```

3

To evaluate the effect of having surgery on patient satisfaction we have to also consider the rest of the data. We want to build an adequate model for patient satisfaction in a hospital. We have four possible predictors, all of them continuous except one; a factor predictor. The factor predictor has two levels indicating if the patient has had surgery or not. We start by looking at a full additive model

```
library(car)
pat = read.csv("PatSat.csv", header = TRUE)
pat$Surgery = as.factor(pat$Surgery)
y = pat$Satisfaction
age = pat$Age
sev = pat$Severity
sur = pat$Surgery
anx = pat$Anxiety
fit.add = lm(y ~ age + sev + sur + anx)
vif(fit.add)

##      age      sev      sur      anx
## 1.939128 1.441055 1.072782 1.689768
```

```
summary(fit.add)

##
## Call:
## lm(formula = y ~ age + sev + sur + anx)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.506  -5.096   1.306   4.738  28.722
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 140.1689     8.3191  16.849 2.77e-13 ***
## age          -1.1428     0.1904  -6.002 7.22e-06 ***
## sev          -0.4699     0.1866  -2.518   0.0204 *
## surYes        2.2259     4.1402   0.538   0.5968
## anx           1.2673     1.4922   0.849   0.4058
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.921 on 20 degrees of freedom
## Multiple R-squared:  0.8183,Adjusted R-squared:  0.7819
## F-statistic: 22.51 on 4 and 20 DF,  p-value: 3.611e-07
```

We achieve with this model a reasonable although not very high adjusted R-squared value. Now we want to scale back; the main effects of the variables Surgery and Anxiety appear to not be significant in this model before considering interactions. Moreoever we are lucky in that the predictor $s$urgery has a very low VIF so multicollinearity will not be a major problem in evaluating if surgery affects patient satisfaction. We decide to start with a model containing only the main effects of age and severity.

```
fit.ageSeverity = lm(y ~ age + sev)
summary(fit.ageSeverity)

##
## Call:
## lm(formula = y ~ age + sev)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.3691  -5.9535   0.2975   4.0462  29.3439
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 139.9233     8.1002  17.274 2.78e-14 ***
## age          -1.0462     0.1573  -6.652 1.09e-06 ***
## sev          -0.4359     0.1788  -2.439   0.0233 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.682 on 22 degrees of freedom
## Multiple R-squared:  0.8096,Adjusted R-squared:  0.7923
## F-statistic: 46.77 on 2 and 22 DF,  p-value: 1.193e-08
```

First we notice that our R-squared values has augmented though very midly while the standard R squared value is almost the same. We look for further simplification:

```
drop1(fit.ageSeverity, test = "F")

## Single term deletions
##
## Model:
## y ~ age + sev
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>              2062.3 116.32
## age     1    4148.3 6210.6 141.88 44.2528 1.092e-06 ***
## sev     1     557.4 2619.7 120.30  5.9467   0.02328 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-test indicates we cannot simplify this model further according to our choice of Fout (4) We now look at possible interactions between the two predictors. We again follow the rule that all main effects have to be included whenever we consider interactions.

```
fit.ageSeverity_int = lm(y ~ age + sev + age:sev)
anova(fit.ageSeverity, fit.ageSeverity_int)

## Analysis of Variance Table
##
## Model 1: y ~ age + sev
## Model 2: y ~ age + sev + age:sev
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     22 2062.3
## 2     21 2032.7  1    29.549 0.3053 0.5864
```

We don’t need any further analysis to see that the added interaction term doesn’t add any statistical significane to the model.Nevertheless we want to look at all possible interactions of the second order and compare them. To the model that only considers the main effects of age and severity (the ”best” one so far).

```
fit.allSecond = lm(y ~ age*(sev + anx + sur))
summary(fit.allSecond)

##
## Call:
## lm(formula = y ~ age * (sev + anx + sur))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.804  -5.544  -1.247   3.747  29.387
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 138.754434  44.705750   3.104  0.00645 **
## age          -1.119644   0.840185  -1.333  0.20025
## sev          -0.122382   1.318898  -0.093  0.92715
## anx          -3.473254  13.038771  -0.266  0.79315
## surYes        9.850133  20.980005   0.470  0.64468
## age:sev      -0.005473   0.021553  -0.254  0.80261
## age:anx       0.076379   0.210855   0.362  0.72164
## age:surYes   -0.148719   0.393749  -0.378  0.71033
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.65 on 17 degrees of freedom
## Multiple R-squared:  0.8221,Adjusted R-squared:  0.7488
## F-statistic: 11.22 on 7 and 17 DF,  p-value: 2.768e-05
```

```
fit.foward = update(fit.ageSeverity, ~. + sur)
summary(fit.foward)

##
## Call:
## lm(formula = y ~ age + sev + sur)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.618  -5.394   1.136   5.061  28.649
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 139.7733     8.2508  16.941 1.01e-13 ***
## age          -1.0605     0.1628  -6.515 1.87e-06 ***
## sev          -0.4410     0.1823  -2.420   0.0247 *
## surYes        1.9865     4.1031   0.484   0.6333
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.855 on 21 degrees of freedom
## Multiple R-squared:  0.8117,Adjusted R-squared:  0.7848
## F-statistic: 30.17 on 3 and 21 DF,  p-value: 8.377e-08
```

The last p-value strongly discourages us from rejecting the null hypothesis. We thus conclude that from the data available having had surgery or not does not seem to significantly affect a patient's satisfaction. At the very least not in way that could be detected through a mean model with all the regular assumptions underlying linear regression. (PS: an anova call wasn't necessary as the test is included in the summary). Also a backwards elimination at the next optional section gives us more confidence in our results.

More careful Version

We start by looking at a full model and looking for simplifications. We choose a Fout of 4 and start carefully by removing interactions.

```
library(car)
pat = read.csv("PatSat.csv", header = TRUE)
pat$Surgery = as.factor(pat$Surgery)
y = pat$Satisfaction
age = pat$Age
sev = pat$Severity
sur = pat$Surgery
anx = pat$Anxiety
fit.full = lm(y ~ age*sev*sur*anx)
drop1(fit.full, test = "F")

## Single term deletions
##
## Model:
## y ~ age * sev * sur * anx
##                  Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>                        1146.8 127.65
## age:sev:sur:anx  1    62.847 1209.6 126.98  0.4932 0.5002

fit.back1 = update(fit.full, ~. -anx:sev:age:sur)
drop1(fit.back1, test = "F")

## Single term deletions
##
## Model:
## y ~ age + sev + sur + anx + age:sev + age:sur + sev:sur + age:anx +
##     sev:anx + sur:anx + age:sev:sur + age:sev:anx + age:sur:anx +
##     sev:sur:anx
##              Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>                    1209.6 126.98
## age:sev:sur  1     4.707 1214.3 125.08  0.0389 0.8476
## age:sev:anx  1   112.288 1321.9 127.20  0.9283 0.3580
## age:sur:anx  1   155.390 1365.0 128.00  1.2846 0.2835
## sev:sur:anx  1    49.023 1258.6 125.97  0.4053 0.5387
```

```
fit.back2 = update(fit.back1, ~. - age:sev:sur - age:sev:anx - age:sur:anx -
    sev:sur:anx)
drop1(fit.back2, test = "F")

## Single term deletions
##
## Model:
## y ~ age + sev + sur + anx + age:sev + age:sur + sev:sur + age:anx +
##     sev:anx + sur:anx
##          Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>                1520.1 124.69
## age:sev  1    14.310 1534.4 122.92  0.1318 0.7220
## age:sur  1   193.230 1713.3 125.68  1.7797 0.2035
## sev:sur  1   215.167 1735.2 126.00  1.9817 0.1810
## age:anx  1     0.070 1520.1 122.69  0.0006 0.9802
## sev:anx  1     5.003 1525.1 122.77  0.0461 0.8331
## sur:anx  1   118.688 1638.7 124.57  1.0931 0.3135
```

```
fit.back3 = update(fit.back2, ~. - age:sev - age:sur - sev:sur - age:anx - sev:anx -
    sur:anx)
drop1(fit.back3, test = "F")

## Single term deletions
##
## Model:
## y ~ age + sev + sur + anx
##        Df Sum of Sq    RSS    AIC F value   Pr(>F)
## <none>              1968.5 119.15
## age     1    3545.1 5513.7 142.90 36.0182 7.22e-06 ***
## sev     1     624.1 2592.6 124.04  6.3408  0.02043 *
## sur     1      28.4 1997.0 117.51  0.2890  0.59677
## anx     1      71.0 2039.5 118.04  0.7212  0.40579
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(fit.full,test = "F")

##
## Call:
## lm(formula = y ~ age * sev * sur * anx)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6595 -4.9004  0.0835  2.0335 23.1272
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -89.68718  516.56705  -0.174    0.866
## age                  3.84695    9.22412   0.417    0.686
## sev                  4.21799   15.02086   0.281    0.785
## surYes             490.96709  585.80620   0.838    0.424
## anx                 85.83325  164.31169   0.522    0.614
## age:sev             -0.09728    0.25905  -0.376    0.716
## age:surYes         -10.80813   10.76625  -1.004    0.342
## sev:surYes          -9.56630   16.21942  -0.590    0.570
## age:anx             -1.50493    2.78524  -0.540    0.602
## sev:anx             -2.00275    4.75125  -0.422    0.683
## surYes:anx        -141.50388  176.53369  -0.802    0.443
## age:sev:surYes       0.20980    0.28866   0.727    0.486
## age:sev:anx          0.03460    0.07944   0.436    0.673
## age:surYes:anx       2.71251    2.97764   0.911    0.386
## sev:surYes:anx       3.07306    4.90578   0.626    0.547
## age:sev:surYes:anx  -0.05776    0.08224  -0.702    0.500
##
## Residual standard error: 11.29 on 9 degrees of freedom
## Multiple R-squared:  0.8941,Adjusted R-squared:  0.7177
## F-statistic: 5.067 on 15 and 9 DF,  p-value: 0.009299
```