Math650 Homework 9

Yu Huang

2006-10-26

Abstract

Question 14, 15 of chapter 12.

1 confirming $F = T^2$

1.1 code

The reduced model is without WEIGHT. Compare the F-statistic and T-statistic.

```
data1 = read.csv("/usr/local/doc/statistical_sleuth/ASCII/case1102.csv")
data1$SEX = as.factor(data1$SEX)
LOG_AB_RATIO = log(data1$BRAIN/data1$LIVER)
data = cbind(data1, LOG_AB_RATIO)


lm_full = lm(LOG_AB_RATIO~DAYS+TUMOR+LOSS+WEIGHT+SEX, data=data)
lm_reduced =  lm(LOG_AB_RATIO~DAYS+TUMOR+LOSS+SEX, data=data)
rss_full = sum(lm_full$residuals^2)
rss_reduced = sum(lm_reduced$residuals^2)
F_stat = (rss_reduced-rss_full)/(rss_full/lm_full$df.residual)
cat("F_stat=", F_stat, "\n")

design_matrix = cbind(data1$DAYS, data1$TUMOR, data1$LOSS, data1$WEIGHT, data1$SEX)
design_matrix = as.matrix(design_matrix)
coeff_weight = lm_full$coefficients[5] #5 because 1 is Intercept
S = sqrt(rss_full/lm_full$df.residual)
t_stat = coeff_weight/(S*(ginv(t(design_matrix) %*% design_matrix)[4,4])^(1/2))
cat("t_stat^2 = ", t_stat^2, "\n")
```

1.2 Result

F-statistic	$T - statistic^2$
1.379458	1.850074

There's difference, but i don't know why.

2 Manually do one-step backward elimination

To see whether the automatic R function, **step** or **stepAIC** works as we think.

2.1 code

Continue the code from above.

```
backward_regression_step = function(reduced_formula, data, lm_full)
{
lm_reduced =  lm(reduced_formula, data=data)
rss_full = sum(lm_full$residuals^2)
rss_reduced = sum(lm_reduced$residuals^2)
rss_delta = rss_reduced-rss_full
F_stat = rss_delta/(rss_full/lm_full$df.residual)
result = data.frame(rss_reduced=rss_reduced, rss_delta=rss_delta, F_stat=F_stat)
return(result)
}

full_formula = LOG_AB_RATIO~DAYS+TUMOR+LOSS+WEIGHT+SEX
reduced_formula1 = update.formula(full_formula, ~.-DAYS)
reduced_formula2 = update.formula(full_formula, ~.-TUMOR)
reduced_formula3 = update.formula(full_formula, ~.-LOSS)
reduced_formula4 = update.formula(full_formula, ~.-WEIGHT)
reduced_formula5 = update.formula(full_formula, ~.-SEX)

for (i in c(reduced_formula1, reduced_formula2, reduced_formula3, reduced_formula4, reduce
{
print(i)
print(backward_regression_step(i, data, lm_full))
}
```

2.2 Results

```
LOG_AB_RATIO ~ TUMOR + LOSS + WEIGHT + SEX
  rss_reduced rss_delta   F_stat
1    112.9698      31.8 10.96960
LOG_AB_RATIO ~ DAYS + LOSS + WEIGHT + SEX
  rss_reduced  rss_delta      F_stat
1    81.18844 0.01862203 0.006423778
LOG_AB_RATIO ~ DAYS + TUMOR + WEIGHT + SEX
  rss_reduced   rss_delta       F_stat
1    81.17224 0.002420654 0.0008350186
LOG_AB_RATIO ~ DAYS + TUMOR + LOSS + SEX
  rss_reduced rss_delta   F_stat
1    85.16876  3.998941 1.379458
LOG_AB_RATIO ~ DAYS + TUMOR + LOSS + WEIGHT
  rss_reduced rss_delta F_stat
1    101.0353  19.86544 6.8527
```

LOSS has lowest F-stat, below threshold 4 and should be removed.

2.3 Compare it with stepAIC()

Run code

```
library(MASS)
stepAIC(lm_full, steps=1)
```

Result:

```
Start:  AIC= 41.59
 LOG_AB_RATIO ~ DAYS + TUMOR + LOSS + WEIGHT + SEX

         Df Sum of Sq     RSS     AIC
- LOSS    1     0.002  81.172  39.587
- TUMOR   1     0.019  81.188  39.594
- WEIGHT  1     3.999  85.169  41.221
<none>                 81.170  41.586
- SEX     1    19.865 101.035  47.030
- DAYS    1    31.800 112.970  50.826

Step:  AIC= 39.59
 LOG_AB_RATIO ~ DAYS + TUMOR + WEIGHT + SEX


Call:
lm(formula = LOG_AB_RATIO ~ DAYS + TUMOR + WEIGHT + SEX, data = data)

Coefficients:
(Intercept)         DAYS        TUMOR       WEIGHT         SEXM
 -2.790e+01    2.193e+00    2.291e-04    1.623e-02    2.384e+00
```

The *Sum of Sq* and *RSS* match the manual results. LOSS is also the lowest one and removed.

2.4 Conclusion

stepAIC works in the right way.

3 Try R or Splus functions

To get the final fit model.

3.1 R

R has a function called, *step*, which is almost same as MASS's *stepAIC*. Both of them worked in backward direction, but failed in *forward* and *both* direction.

```
back_result = stepAIC(lm_full)
lm_mean = lm(LOG_AB_RATIO~1, data=data)
#forward and both seem not to work.
forward_result = stepAIC(lm_mean, direction="forward")
both_result = stepAIC(lm_mean, direction="both")
```

So based on backward, the final model is

```
Step:  AIC= 37.28
 LOG_AB_RATIO ~ DAYS + SEX

       Df Sum of Sq     RSS     AIC
<none>               85.316  37.280
- DAYS  1    29.069 114.385  45.249
- SEX   1    55.202 140.518  52.245
```

3.2 Splus

Splus has a function named, stepwise, which works fairly well. But stepwise doesn't use 4 as F-statistic cutoff to choose parameters. It just runs till the end.

3.2.1 code

```
#ssh almaak.usc.edu, Splus version 6.1.2.  7.0 doesn't work due to license problem.
#splus, no "_" in variable name in splus for '='.
#If '_', use '<-', i.e. LOG_AB_RATIO <- log(data1$BRAIN/data1$LIVER); print(LOG_AB_RATIO)
#directly typing 'LOG_AB_RATIO' outputs nothing
data1 = importData("./MySwork/case1102.csv", type="ASCII")
data1$SEX = as.factor(data1$SEX)
LOGABRATIO = log(data1$BRAIN/data1$LIVER)
data = cbind(data1, LOG_AB_RATIO)
dtrix = cbind(data1$DAYS, data1$TUMOR, data1$LOSS, data1$WEIGHT, data1$SEX)
stepwise(dtrix, LOGABRATIO, method="forward", f.crit=4.0)
stepwise(dtrix, LOGABRATIO, method="backward", f.crit=c(4.0,4.0))
stepwise(dtrix, LOGABRATIO, method="efroymson")
stepwise(dtrix, LOGABRATIO, method="exhaustive")
```

3.2.2 Forward Result

```
$rss:
[1] 114.38450  85.31586  81.18877  81.17224  81.16982

$size:
[1] 1 2 3 4 5

$which:
      X1 X2 X3 X4 X5
1(+5)  F  F  F  F  T
2(+1)  T  F  F  F  T
3(+4)  T  F  F  T  T
4(+2)  T  T  F  T  T
5(+3)  T  T  T  T  T

$f.stat:
[1] 1.466525e+01 1.056226e+01 1.524996e+00 5.906594e-03 8.350186e-04

$method:
[1] "forward
```

Here's a little explanation of the matrix, *which*.

```
Matrix, which, is a
logical matrix with as many rows as there are returned subsets. Each row is a
logical vector that can be used to select the columns of x in the subset. For the
forward method there are ncol(x) rows with subsets of size 1, ..., ncol(x). For the
backward method there are ncol(x) rows with subsets of size ncol(x), ..., 1. For
Efroymson's method there is a row for each step of the stepwise procedure. For the
exhaustive search, there are nbest subsets for each size (if available). The row
labels consist of the subset size with some additional information in parentheses.
For the stepwise methods the extra information is +n or -n to indicate that the n-th
variable has been added or dropped. For the exhaustive method, the extra information
is #i where i is the subset number.
```

If we use 4 as f.stat cutoff, we'll stop after two rounds with $LOGABRATIO \sim DAYS + SEX$(X1 is DAYS and X5 is SEX), which is same as the result from stepAIC of R.

3.2.3 Backward Result

```
$rss:
[1]  81.17224  81.18877  85.31586 114.38450 166.80568

$size:
[1] 4 3 2 1 0

$which:
      X1 X2 X3 X4 X5
4(-3)  T  T  F  T  T
3(-2)  T  F  F  T  T
2(-4)  T  F  F  F  T
1(-1)  F  F  F  F  T
0(-5)  F  F  F  F  F

$f.stat:
[1] 8.350186e-04 5.906594e-03 1.524996e+00 1.056226e+01 1.466525e+01

$method:
[1] "backward"
```

If we use 4 as f.stat cutoff, we'll stop after 3 round and end up with same model as the forward method.

3.2.4 efroymson Result

```
$rss:
[1] 114.38450  85.31586

$size:
[1] 1 2
```

```
$which:
      X1 X2 X3 X4 X5
1(+5)  F  F  F  F  T
2(+1)  T  F  F  F  T

$f.stat:
[1] 14.66525 10.56226

$method:
[1] "efroymson"
```

Not quite sure about what efroymson is, but this one stops at the same model.

3.2.5 Exhaustive(both) Result

The exhaustive method is same as the *both* of stepAIC().

```
$rss:
 [1] 114.38450 140.51795 141.61316  85.31586 104.10268 113.50181  81.18877
 [8]  85.17482  85.30437  81.17224  81.18844  85.16876  81.16982

$size:
 [1] 1 1 1 2 2 2 3 3 3 4 4 4 5

$which:
      X1 X2 X3 X4 X5
1(#1)  F  F  F  F  T
1(#2)  T  F  F  F  F
1(#3)  F  F  F  T  F
2(#1)  T  F  F  F  T
2(#2)  T  F  F  T  F
2(#3)  F  F  F  T  T
3(#1)  T  F  F  T  T
3(#2)  T  F  T  F  T
3(#3)  T  T  F  F  T
4(#1)  T  T  F  T  T
4(#2)  T  F  T  T  T
4(#3)  T  T  T  F  T
5(#1)  T  T  T  T  T

$method:
[1] "exhaustive"
```

This one lacks f.stat, no idea where it should be stopped. I don't why stepwise of Splus doesn't output f.stat for the *exhaustive* method.

3.3 Conclusion

Whether the stepwise direction is **backward** or **forward**(**both** is not sure yet), the final model is same. This demonstrates that the stepwise regression is pretty robust.