Analysis of Covariance (ANCOVA) in R (draft)

Francis Huang

August 13th, 2014

Introduction

This short guide shows how to use our SPSS class example and get the same results in R. We introduce the new variable– the covariate or the concomitant variable. We would like to control or account for this third variable (a continuous variable) and if all goes well, we get better results. We'll need to install a few packages, namely: `rio`,`car`, `multcomp`, `effects`, `psych`. NOTE: don't need to reinstall if you've already installed. You must call them using the `library` function.

Let's first get our data[1], in R. You'll need the `rio` package to read the SPSS file. After importing our data, convert group into a factor with the appropriate labels.

[1] `http://web.missouri.edu/~huangf/data/quantf/ch14stateg.sav`

```
options(digits = 3)
library(rio)
x <- import("http://web.missouri.edu/~huangf/data/quantf/ch14stateg.sav")
str(x)
```

```
## 'data.frame':    12 obs. of  3 variables:
##  $ group   :Class 'labelled'  atomic [1:12] 1 1 1 1 1 1 2 2 2 2 ...
##   .. ..- attr(*, "labels")= Named num [1:2] 1 2
##   .. .. ..- attr(*, "names")= chr [1:2] "Trad" "New Method"
##  $ quiz    : num  1 2 3 4 5 6 1 2 4 5 ...
##  $ aptitude: num  4 3 5 6 7 9 1 3 2 4 ...
```

```
x$group <- factor(x$group, label = c("Trad", "New Method"))
str(x)
```

```
## 'data.frame':    12 obs. of  3 variables:
##  $ group   : Factor w/ 2 levels "Trad","New Method": 1 1 1 1 1 1 2 2 2 2 ...
##  $ quiz    : num  1 2 3 4 5 6 1 2 4 5 ...
##  $ aptitude: num  4 3 5 6 7 9 1 3 2 4 ...
```

```
table(x$group)
```

```
##
##       Trad New Method
##          6          6
```

Assumption checking

Now we want to compare some assumptions (see the textbook).

Assumption 1: equality of slopes–interaction is not significiant, testing the equality of slopes that the covariate is associated with the outcome the same way between groups we are just interested in the interaction term here.

```
res1 <- aov(quiz ~ group + aptitude + group:aptitude,
    data = x)
summary(res1)
```

```
##                Df Sum Sq Mean Sq F value
## group           1   0.75    0.75     0.7
## aptitude        1  30.94   30.94    28.9
## group:aptitude  1   0.00    0.00     0.0
## Residuals       8   8.56    1.07
##                 Pr(>F)
## group          0.42670
## aptitude       0.00066 ***
## group:aptitude 1.00000
## Residuals
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

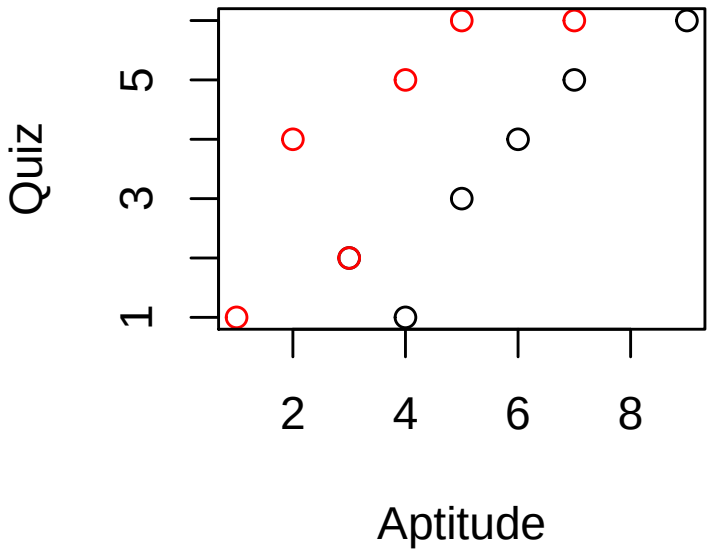
Assumption 2: linearity of slopes. Here, we are eyeballing the slopes or the trend lines between the two groups. They look roughly equal (not, we can have more formal tests for this)

```
plot(x$aptitude, x$quiz, col = x$group, xlab = "Aptitude",
    ylab = "Quiz")
```

Assumption 3: Equality of the two groups on the covariate. Just run a t-test and show that two groups are not different on the covariate. If more than two groups, of course you can run an ANOVA. Results below show no statistically significant difference.

```
t.test(aptitude ~ group, data = x)
```

```
##
##  Welch Two Sample t-test
##
## data:  aptitude by group
## t = 2, df = 10, p-value = 0.1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```


Quiz
1
3
5
2
4
6
8
Aptitude

```
##  -0.779  4.779
## sample estimates:
##       mean in group Trad
##                     5.67
## mean in group New Method
##                     3.67
```

Assumption 4: Homogeneity of variance. We've already discussed this before. To get this, run[2]:

[2] Install the `car` package first to access the `levene.test` function. Including the `center=mean` option gives the same result as SPSS

```
library(car)
leveneTest(quiz ~ group, center = mean, data = x)
```

```
## Levene's Test for Homogeneity of Variance (center = mean)
##       Df F value Pr(>F)
## group  1    0.09   0.77
##       10
```

HOV is supported.

Running the actual ANCOVA

When running an ANCOVA, order matters. You want to remove the effect of the covariate first– that is, you want to control for it– prior to entering your main variable or interest. [3]

[3] If you do not do this in order, you will get different results!

```
res1 <- aov(quiz ~ aptitude + group, data = x)
# NOTE: covariate goes first!! NOTE: there is
# no interaction
summary(res1)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## aptitude     1  20.88   20.88    22.0 0.0011
## group        1  10.81   10.81    11.4 0.0082
## Residuals    9   8.56    0.95
##
## aptitude    **
## group       **
## Residuals
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

```
TukeyHSD(res1)  #does not produce correct differences!
```

```
## Warning in replications(paste("~", xx), data
## = mf): non-factors ignored: aptitude
```

```
## Warning in TukeyHSD.aov(res1): 'which'
## specified some non-factors which will be
## dropped
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = quiz ~ aptitude + group, data = x)
##
## $group
##                 diff  lwr  upr p adj
## New Method-Trad 1.69 0.42 2.97 0.015
```

```
# You can check for normality here because we
# need the residuals of the actual model. To
# get the residuals, just use the resid
# function
```

```
residuals <- resid(res1)
residuals
```

```
##       1       2       3       4       5
## -1.1429  0.6714  0.0429  0.2286  0.4143
##       6       7       8       9      10
## -0.2143 -0.8286 -1.4571  1.3571  0.7286
##      11      12
##  0.9143 -0.7143
```

```
shapiro.test(residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 1, p-value = 0.9
```

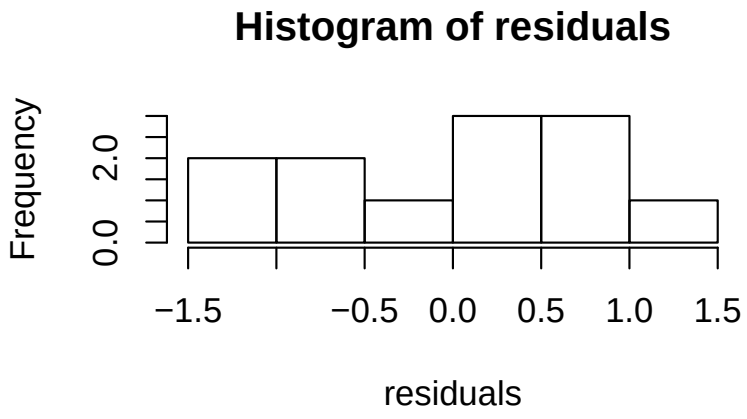
```
# library(e1071) #this contains the skew
# function skew(residuals) skew(x$quiz)
hist(residuals)
```



```
# all points to normality being met
```

```
# to get the 'real' p value and difference,
library(multcomp)
```

```
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
```




```
# uses the general linear hypothesis function
posthoc <- glht(res1, linfct = mcp(group = "Tukey"))
# looks odd-- uses the glht function. Just
# replace the output object (res1), the
# grouping variable (group)
summary(posthoc)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = quiz ~ aptitude + group, data = x)
##
## Linear Hypotheses:
##                         Estimate Std. Error
## New Method - Trad == 0    2.129      0.631
##                         t value Pr(>|t|)
## New Method - Trad == 0    3.37   0.0082 **
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
confint(posthoc)
```

```
##
##   Simultaneous Confidence Intervals
##
```

```
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = quiz ~ aptitude + group, data = x)
##
## Quantile = 2.26
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##                         Estimate lwr   upr
## New Method - Trad == 0 2.129    0.701 3.556
```

NOTE: you want to see the means by group (unadjusted):

```
library(psych)  #adds describBy
```

```
##
## Attaching package: 'psych'
##
## The following object is masked from 'package:car':
##
##     logit
```

```
# if you get an error, that means you have to
# install the package first
# install.packages('psych')
describeBy(x$quiz, x$group)
```

```
## group: Trad
##   vars n mean   sd median trimmed  mad min
## 1    1 6  3.5 1.87    3.5     3.5 2.22   1
##   max range skew kurtosis   se
## 1   6     5    0     -1.8 0.76
## ------------------------------
## group: New Method
##   vars n mean  sd median trimmed  mad min
## 1    1 6    4 2.1    4.5       4 2.22   1
##   max range  skew kurtosis   se
## 1   6     5 -0.33    -1.88 0.86
```

To get the *adjusted* group means– adjusted for the covariate, need to use the `effects` package.

```
# install.packages('effects')
library(effects)
```

```
##
## Attaching package: 'effects'
##
## The following object is masked from 'package:car':
##
##     Prestige
```

```
effect("group", res1)
```

```
##
##  group effect
## group
##       Trad New Method
##       2.69       4.81
```

Note that the means differences (2.69 vs. 4.81) is much bigger compared to the original mean differences of 3.5 vs. 4. That's why we have statistically significant results too. NOTE: If you don't follow the ordering advice:

```
res2 <- aov(quiz ~ group + aptitude, data = x)
# In ANOVA, above, the effect of group is
# removed first, then aptitude
summary(res2)  #not significant here
```

```
##             Df Sum Sq Mean Sq F value
## group        1   0.75    0.75    0.79
## aptitude     1  30.94   30.94   32.54
## Residuals    9   8.56    0.95
##              Pr(>F)
## group       0.39757
## aptitude    0.00029 ***
## Residuals
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1  ' ' 1
```

Your results will be different! It will look like you ran your model without the covariate. NOTE, the results below are the same for group because they were both entered first.

```
res3 <- aov(quiz ~ group, data = x)
# When there was no covariate, SS was .75--
# not significant
summary(res3)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## group        1    0.7    0.75    0.19   0.67
## Residuals   10   39.5    3.95
```