R commands for oneway ANOVA

1. Doing oneway ANOVA on the data in the textbook, page 287.

```
> #####
> # Import data set. This is formated two columns/variables
> PF.data <- read.table(file="pulmonary function.txt", header=TRUE, na.strings =
".")
> # Display a few lines
> head(PF.data)
  center fev1
1      1 3.23
2      1 3.47
3      1 1.86
4      1 2.47
5      1 3.01
6      1 1.69
>
> # center is a categorical (factor) variable
> #We need to do this for the grouping variable
> PF.data$center<-as.factor(PF.data$center)
>
> # Fit by aov(), then prodcue the ANOVA table
> PF.fit <- aov(fev1~center, data=PF.data)
> summary(PF.fit)
            Df Sum Sq Mean Sq F value Pr(>F)
center       2  1.583  0.7914   3.115  0.052 .
Residuals   57 14.480  0.2540
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
3 observations deleted due to missingness
```

2. Contrasts.

We can calculate the contrast estimate and the corresponding t-test as following.

```
> #Contrast of John Hopkins (center 1) versus Rancho Los Amigos (2)
> contr1<-c(1, -1, 0)
> #Find groups means and sds by center
> require(psych)
> stat.fev1<-describeBy(PF.data$fev1,PF.data$center)
> require(data.table)
> stat.fev2<-data.frame(rbindlist(stat.fev1))[,c('n','mean','sd')]
> stat.fev2 #display
   n     mean        sd
1 21 2.626190 0.4961701
2 16 3.032500 0.5232399
3 23 2.878696 0.4977157
> # Estimate of contrast
> contr1.est<-sum(contr1*stat.fev2[,'mean'])
> # Find the standard error of the contrast estimate
> # By hand we would use
> MSE<-sum((stat.fev2[,'n']-1)*stat.fev2[,'sd']^2)/sum(stat.fev2[,'n']-1)
> # But we really like to get MSE directly from the ANOVA fit
> MSE<-summary(PF.fit)[[1]][2,'Mean Sq'] #extract MSE
> contr1.se<- sqrt(sum(contr1^2/stat.fev2[,'n'])*MSE)
> contr1.t<- contr1.est/contr1.se
> k<-length(levels(PF.data$center)) #number of groups
> contr1.p<-2*pt(-abs(contr1.t),df=sum(stat.fev2[,'n'])-k)

> #Display estimate, se, t-statistic, p-value
> c(contr1.est, contr1.se, contr1.t, contr1.p)
[1] -0.40630952  0.16725608 -2.42926607  0.01830437
```

Alternatively we can do this in R through the design matrix, which will need (k-1) non-linear-dependent contrasts. Here we have k=3 groups, so that means we can do 2 contrasts at a time.

```
> ### Use the contrast design matrix to do this
> contr2<-c(0,1,-1)     #compare the last two centers
> contr3<-c(2,-1,-1)    #compare the first center with the last two together.
> 
> options('contrasts')
$contrasts
        unordered           ordered 
"contr.treatment"      "contr.poly" 
> contr.mat <- rbind(rep(1/3, 3), contr1, contr2)
> my.contr <- solve(contr.mat)[,-1] ## Get the inverse matrix, put into the contra
sts in lm()
> contrasts(PF.data$center)<-my.contr
> summary(lm(fev1~center, data=PF.data))
Call:
lm(formula = fev1 ~ center, data = PF.data)
Residuals:
     Min       1Q   Median       3Q      Max 
-1.32250 -0.32250 -0.02244  0.32630  1.18130 
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.84580    0.06584  43.220   <2e-16 ***
centercontr1 -0.40631    0.16726  -2.429   0.0183 *  
centercontr2  0.15380    0.16408   0.937   0.3525    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.504 on 57 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.09854,	Adjusted R-squared:  0.06691 
F-statistic: 3.115 on 2 and 57 DF,  p-value: 0.052
```

Compare with the results we did with the formulas above for the first contrast, we can see the results are the same.

```
> c(contr1.est, contr1.se, contr1.t, contr1.p)
[1] -0.40630952  0.16725608 -2.42926607  0.01830437
```

To get the third contrast, use a new design matrix

```
> contr.mat <- rbind(rep(1/3, 3), contr2, contr3) #contrasts 2 and 3
> my.contr <- solve(contr.mat)[,-1] ## Get the inverse matrix, put into the contra
sts in lm()
> contrasts(PF.data$center)<-my.contr
> summary(lm(fev1~center, data=PF.data))
Call:
lm(formula = fev1 ~ center, data = PF.data)
Residuals:
     Min       1Q   Median       3Q      Max 
-1.32250 -0.32250 -0.02244  0.32630  1.18130 
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.84580    0.06584  43.220   <2e-16 ***
centercontr2  0.15380    0.16408   0.937   0.3525    
centercontr3 -0.65881    0.27443  -2.401   0.0197 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.504 on 57 degrees of freedom
  (3 observations deleted due to missingness)
Multiple R-squared:  0.09854,	Adjusted R-squared:  0.06691 
F-statistic: 3.115 on 2 and 57 DF,  p-value: 0.052
```

3. Adjustments for multiple testing

For planned contrasts, say these three contrasts here. Check the output

```
             Estimate Std. Error t value Pr(>|t|)
centercontr1 -0.40631    0.16726  -2.429   0.0183 *
centercontr2  0.15380    0.16408   0.937   0.3525
centercontr3 -0.65881    0.27443  -2.401   0.0197 *
```

For Bonferroni:
m=3 contrasts, so compare p-values with α/m instead. At α=0.05 level, we need to compare with 0.05/3=0.0167.
Now all 3 contrasts p-values are bigger than 0.0167, thus none is significant.

For Scheffe:
Calculate the cutoff (α=0.05) as

```
> sqrt((k-1)*qf(0.95,df1=k-1,df2=n-k)) #Scheffe coeficient at alpha=0.05
[1] 2.513501
```

None of the t statistics exceed 2.51, thus none is significant. The answer is same as that using Bonferroni adjustment.

For Tukey:
The first two contrasts are in fact pairwise comparisons and can use the Tukey's correction. So calculate the cutoff (α=0.05) as

```
> qtukey(0.95,nmeans=k, df=n-k)*contr1.se/sqrt(2)
[1] 0.4024881
```

Since the difference is 0.40631, bigger than the cutoff, the first contrast is significant. This differs from the conclusion from the Bonferroni and Scheffe's adjustment above. Notice that the Tukey's adjustment assume that only pairwise comparisons are considered, thus excluding the third contrast above.

4. Adjustments for all pairwise comparisons (post-hoc tests)

For comparing all pairs of group means. We can use pairwise.t.test() and TukeyHSD().

```
> ## Pairwise comparisons
> # No adjustment or LSD
> pairwise.t.test(PF.data$fev1, g=PF.data$center, p.adjust.method = 'none')
        Pairwise comparisons using t tests with pooled SD
data:  PF.data$fev1 and PF.data$center
  1     2
2 0.018 -
3 0.102 0.353
P value adjustment method: none
> # Bonferroni for the k(k-1)/2 pairs
> pairwise.t.test(PF.data$fev1, g=PF.data$center, p.adjust.method = 'bonferroni')
        Pairwise comparisons using t tests with pooled SD

data:  PF.data$fev1 and PF.data$center

  1     2
2 0.055 -
3 0.307 1.000

P value adjustment method: bonferroni

> # FDR or BH
> pairwise.t.test(PF.data$fev1, g=PF.data$center, p.adjust.method = 'fdr')
        Pairwise comparisons using t tests with pooled SD

data:  PF.data$fev1 and PF.data$center

  1     2
2 0.055 -
3 0.154 0.353
P value adjustment method: fdr

> # Tukey's HSD
> TukeyHSD(PF.fit, confidence.level=0.95)
  Tukey multiple comparisons of means
    95% family-wise confidence level
Fit: aov(formula = fev1 ~ center, data = PF.data)

$center
          diff          lwr       upr     p adj
2-1  0.4063095  0.003821453 0.8087976 0.0473852
3-1  0.2525052 -0.113573610 0.6185840 0.2294901
3-2 -0.1538043 -0.548652567 0.2410439 0.6191128
```

At α=0.05 level, the difference between first and second groups (centers here) is significant under the Tukey's adjustment, but not significant under Bonferroni or fdr adjustment.
Other pairwise comparisons are all not significant.

==========Some notes on R commands ======================

We extracted the MSE using

```
MSE<-summary(PF.fit)[[1]][2,'Mean Sq'] #extract MSE
```

How do I know to extract it this way? Look at what is contained in `summary(PF.fit)`

```
> sum1<-summary(PF.fit)
> str(sum1)
List of 1
 $ :Classes ‘anova’ and 'data.frame': 2 obs. of  5 variables:
  ..$ Df     : num [1:2] 2 57
  ..$ Sum Sq : num [1:2] 1.58 14.48
  ..$ Mean Sq: num [1:2] 0.791 0.254
  ..$ F value: num [1:2] 3.12 NA
  ..$ Pr(>F) : num [1:2] 0.052 NA
 - attr(*, "class")= chr [1:2] "summary.aov" "listof"
 - attr(*, "na.action")=Class 'omit'  Named int [1:3] 12 30 37
  .. ..- attr(*, "names")= chr [1:3] "12" "30" "37"
```

It is a list, the first element in a list is indexed as [[1]], which is a data.frame here:

```
> sum1[[1]]
            Df  Sum Sq Mean Sq F value Pr(>F)
center       2  1.5828 0.79142  3.1153  0.052 .
Residuals   57 14.4803 0.25404
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The MSE is in the second row (Residuals) and in the third column, so we can extract it using either of the two ways below:

```
> sum1[[1]][2,'Mean Sq']
[1] 0.2540396
> sum1[[1]]['Residuals','Mean Sq']
[1] 0.2540396
```