Ancova Field

Lets load a bunch of packages

```
library(multcomp)
```

```
## Loading required package: mvtnorm
## Loading required package: survival
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'TH.data'
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
library(compute.es)
library(effects)
```

```
## Loading required package: carData
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
library(ggplot2)
library(pastecs)
library(WRS2)
library(psych)
```

```
##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(pander)
library(car)
```

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

Import the data

```
viagraData <- read.csv("ViagraCovariate.csv",sep="\t")
viagraData$dose <- factor(viagraData$dose,
                          labels = c("Placebo","Low Dose","Hight Dose"))
```

```r
head(viagraData)
```

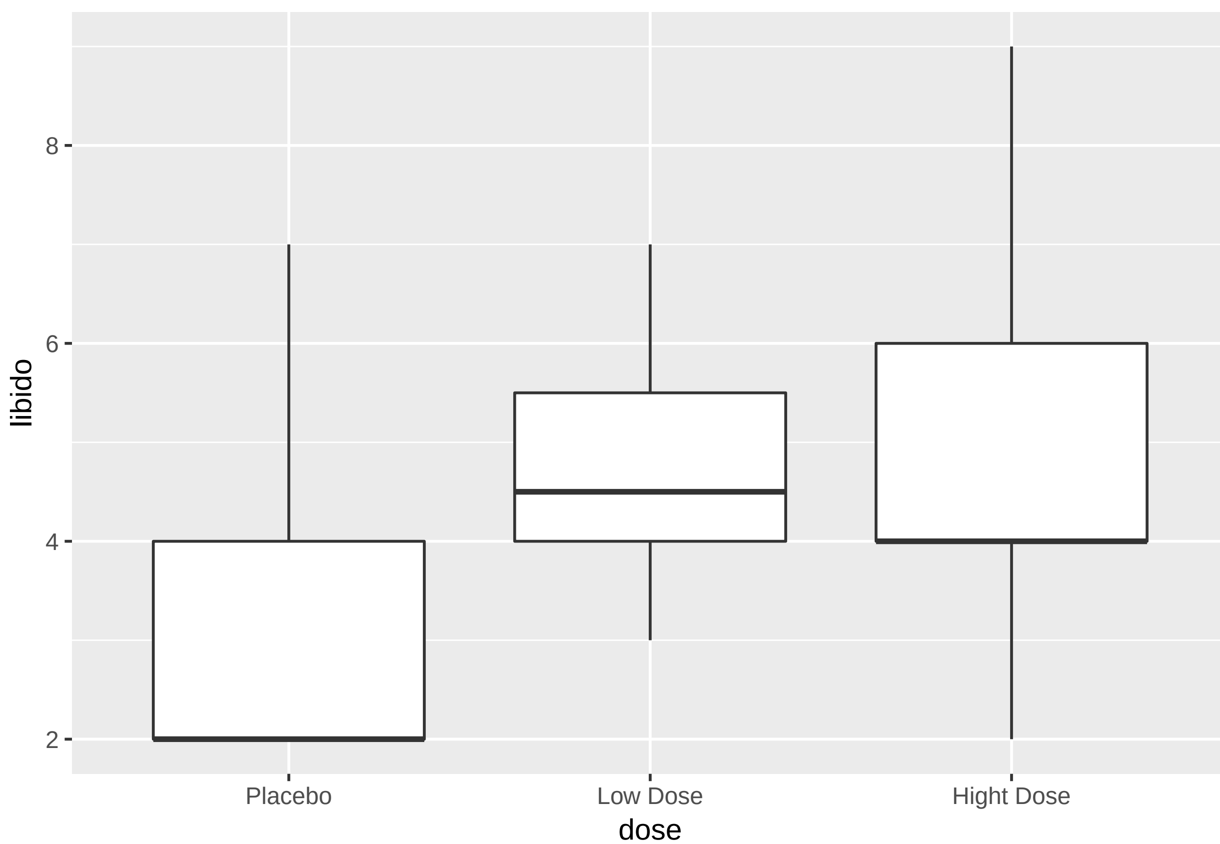
```
##      dose libido partnerLibido
## 1 Placebo      3             4
## 2 Placebo      2             1
## 3 Placebo      5             5
## 4 Placebo      2             1
## 5 Placebo      2             2
## 6 Placebo      2             2
```

```r
tail(viagraData)
```

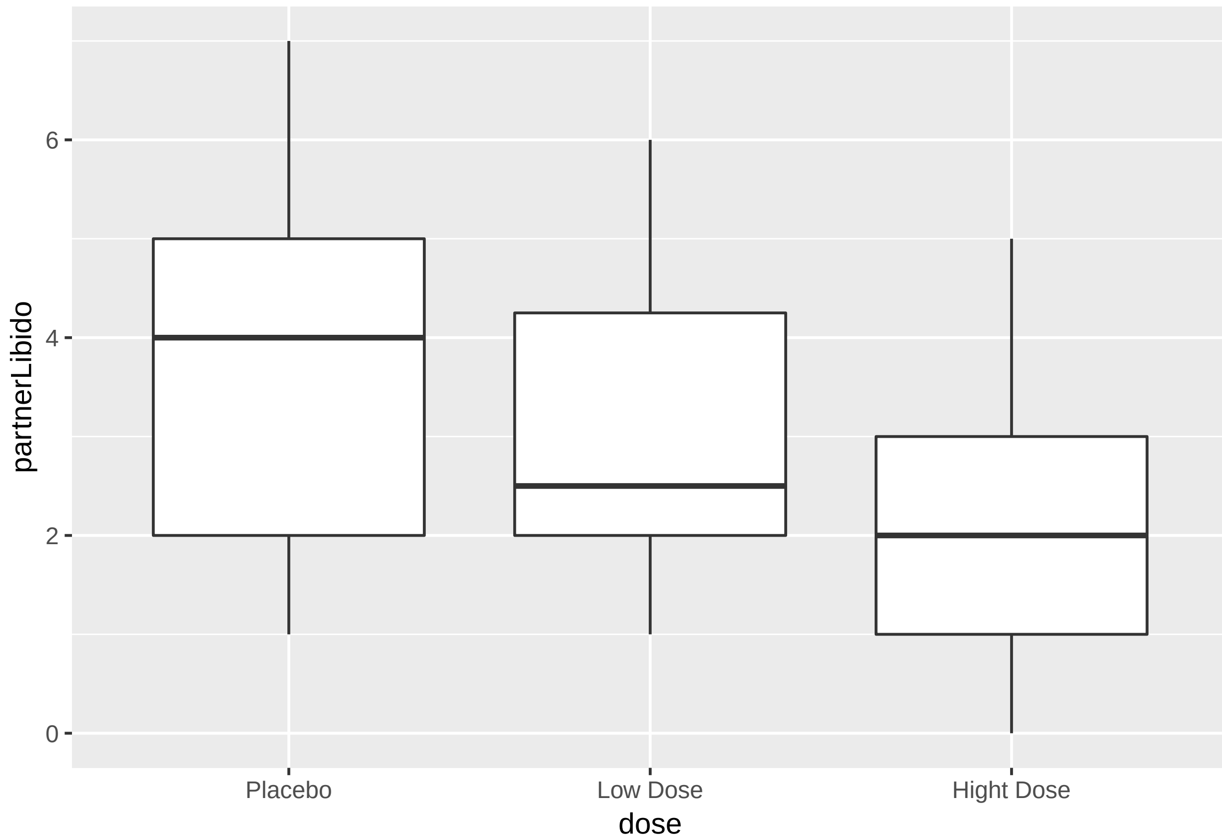
```
##          dose libido partnerLibido
## 25 Hight Dose      6             0
## 26 Hight Dose      4             1
## 27 Hight Dose      6             3
## 28 Hight Dose      2             0
## 29 Hight Dose      8             1
## 30 Hight Dose      5             0
```

Some illustrative plots

```r
ggplot(viagraData,aes(dose,libido))+
geom_boxplot()
```



```r
ggplot(viagraData,aes(dose,partnerLibido))+
  geom_boxplot()
```



Some Descriptive statistics

```r
by(viagraData$libido, viagraData$dose, stat.desc)
```

```
## viagraData$dose: Placebo
##      nbr.val     nbr.null       nbr.na          min          max
##    9.0000000    0.0000000    0.0000000    2.0000000    7.0000000
##        range          sum       median         mean      SE.mean
##    5.0000000   29.0000000    2.0000000    3.2222222    0.5957670
## CI.mean.0.95          var      std.dev     coef.var
##    1.3738411    3.1944444    1.7873009    0.5546796
## --------------------------------------------------------
## viagraData$dose: Low Dose
##      nbr.val     nbr.null       nbr.na          min          max
##    8.0000000    0.0000000    0.0000000    3.0000000    7.0000000
##        range          sum       median         mean      SE.mean
##    4.0000000   39.0000000    4.5000000    4.8750000    0.5153882
## CI.mean.0.95          var      std.dev     coef.var
##    1.2186994    2.1250000    1.4577380    0.2990232
## --------------------------------------------------------
## viagraData$dose: Hight Dose
##      nbr.val     nbr.null       nbr.na          min          max
##   13.0000000    0.0000000    0.0000000    2.0000000    9.0000000
##        range          sum       median         mean      SE.mean
```

```
##    7.0000000   63.0000000    4.0000000    4.8461538    0.5866698
## CI.mean.0.95          var      std.dev     coef.var
##    1.2782437    4.4743590    2.1152681    0.4364839
```

```
by(viagraData$partnerLibido, viagraData$dose, stat.desc)
```

```
## viagraData$dose: Placebo
##      nbr.val     nbr.null       nbr.na          min          max
##    9.0000000    0.0000000    0.0000000    1.0000000    7.0000000
##        range          sum       median         mean      SE.mean
##    6.0000000   31.0000000    4.0000000    3.4444444    0.6894263
## CI.mean.0.95          var      std.dev     coef.var
##    1.5898199    4.2777778    2.0682789    0.6004681
## --------------------------------------------------------
## viagraData$dose: Low Dose
##      nbr.val     nbr.null       nbr.na          min          max
##    8.0000000    0.0000000    0.0000000    1.0000000    6.0000000
##        range          sum       median         mean      SE.mean
##    5.0000000   25.0000000    2.5000000    3.1250000    0.6105472
## CI.mean.0.95          var      std.dev     coef.var
##    1.4437147    2.9821429    1.7268882    0.5526042
## --------------------------------------------------------
## viagraData$dose: Hight Dose
##      nbr.val     nbr.null       nbr.na          min          max
##   13.0000000    3.0000000    0.0000000    0.0000000    5.0000000
##        range          sum       median         mean      SE.mean
##    5.0000000   26.0000000    2.0000000    2.0000000    0.4529108
## CI.mean.0.95          var      std.dev     coef.var
##    0.9868079    2.6666667    1.6329932    0.8164966
```

Compute Levene Test

```
leveneTest(viagraData$libido, viagraData$dose, center = median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2  0.3256 0.7249
##       27
```

Are the predictor variable and covariate independent?

In this case, the proposed covariate is partner's libido, and we need to check that this variable was roughly equal across levels of our independent variable. In other words, is the mean level of partner's libido roughly equal across our three Viagra groups? We can test this by running an ANOVA with partnerLibido as the outcome and dose as the predictor.

Conduct an ANOVA to test whether partner's libido (our covariate) is independent of the dose of Viagra (our independent variable)

```
lm_partner <- lm(partnerLibido~dose,data=viagraData)
lm_partner
```

```
##
```

```
## Call:
## lm(formula = partnerLibido ~ dose, data = viagraData)
##
## Coefficients:
##    (Intercept)    doseLow Dose  doseHight Dose
##         3.4444         -0.3194         -1.4444
```

```
aov_partner <- aov(lm_partner)
anova(aov_partner)
```

```
## Analysis of Variance Table
##
## Response: partnerLibido
##           Df Sum Sq Mean Sq F value Pr(>F)
## dose       2 12.769  6.3847  1.9793 0.1577
## Residuals 27 87.097  3.2258
```

Since there was no significance in the anova we can fit the ANCOVA modelas follows

```
viagraModel<-aov(libido ~ dose - partnerLibido, data = viagraData)
Anova(viagraModel,type="3")
```

```
## Anova Table (Type III tests)
##
## Response: libido
##             Sum Sq Df F value    Pr(>F)
## (Intercept) 93.444  1 26.8054 1.891e-05 ***
## dose        16.844  2  2.4159    0.1083
## Residuals   94.123 27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(viagraModel,type="III")
```

```
## Anova Table (Type III tests)
##
## Response: libido
##             Sum Sq Df F value    Pr(>F)
## (Intercept) 93.444  1 26.8054 1.891e-05 ***
## dose        16.844  2  2.4159    0.1083
## Residuals   94.123 27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that we have simply added '+ partnerLibido' to the list of predictors. In essence, this is all there is to it. We could simply execute this command, sit back, crack open a cool drink and admire our handiwork. However, just as we were starting to enjoy a wave of smugness at having conducted an ANCOVA, the sinister shadow of humility would slap us on the face and point out that we need to think about the order of our predictors. If we usethe aov() function alone then we'll get different results if we specify our model as 'libido ~ dose + partnerLibido' than if we specify 'libido ~ partnerLibido + dose' (note the order of predictors). This is curious and is something to which we need to give some thought

```
contrasts(viagraData$dose)<-cbind(c(-2,1,1), c(0,-1,1))
viagraModel<-aov(libido ~ partnerLibido + dose, data = viagraData)
Anova(viagraModel, type="III")
```

```
## Anova Table (Type III tests)
##
```

```
## Response: libido
##               Sum Sq Df F value    Pr(>F)
## (Intercept)   76.069  1 25.0205 3.342e-05 ***
## partnerLibido 15.076  1  4.9587   0.03483 *
## dose          25.185  2  4.1419   0.02745 *
## Residuals     79.047 26
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now if we want to explore group effects first we need to do adjustment of the means with de effects function

```r
adjustedMeans<-effect("dose", viagraModel, se=TRUE)
summary(adjustedMeans)
```

```
##
##  dose effect
## dose
##    Placebo   Low Dose Hight Dose
##   2.926370   4.712050   5.151251
##
##  Lower 95 Percent Confidence Limits
## dose
##    Placebo   Low Dose Hight Dose
##   1.700854   3.435984   4.118076
##
##  Upper 95 Percent Confidence Limits
## dose
##    Placebo   Low Dose Hight Dose
##   4.151886   5.988117   6.184427
```

```r
adjustedMeans$se
```

```
## [1] 0.5962045 0.6207971 0.5026323
```

The overall ANCOVA does not tell us which means differ, so to break down the overall effect of dose we need to look at the contrasts that we specified before we created the ANCOVA model. To see these contrasts we can use the summary.lm() function on the ANCOVA model (viagraModel):

```r
summary.lm(viagraModel)
```

```
##
## Call:
## aov(formula = libido ~ partnerLibido + dose, data = viagraData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2622 -0.7899 -0.3230  0.8811  4.5699
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.1260     0.6250   5.002 3.34e-05 ***
## partnerLibido   0.4160     0.1868   2.227  0.03483 *
## dose1           0.6684     0.2400   2.785  0.00985 **
## dose2           0.2196     0.4056   0.541  0.59284
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.744 on 26 degrees of freedom
## Multiple R-squared:  0.2876, Adjusted R-squared:  0.2055
## F-statistic:   3.5 on 3 and 26 DF,  p-value: 0.02954
```

It is also possible to obtain post hoc tests as we did for ANOVA (see section 10.6.8). However, because we want to test differences between the adjusted means, we can use only the glht() function; the pairwise.t.test() function will not test the adjusted means. As such, we are limited to using Tukey or Dunnett's post hoc tests. Remember from Chapter 10 that to use this function we enter our model (in this case the ANCOVA model) into it and then use the summary() and confint() functions to see the post hoc tests in the console. For the viagraModel, we could therefore execute:

```
postHocs<-glht(viagraModel, linfct = mcp(dose = "Tukey"))
summary(postHocs)
```

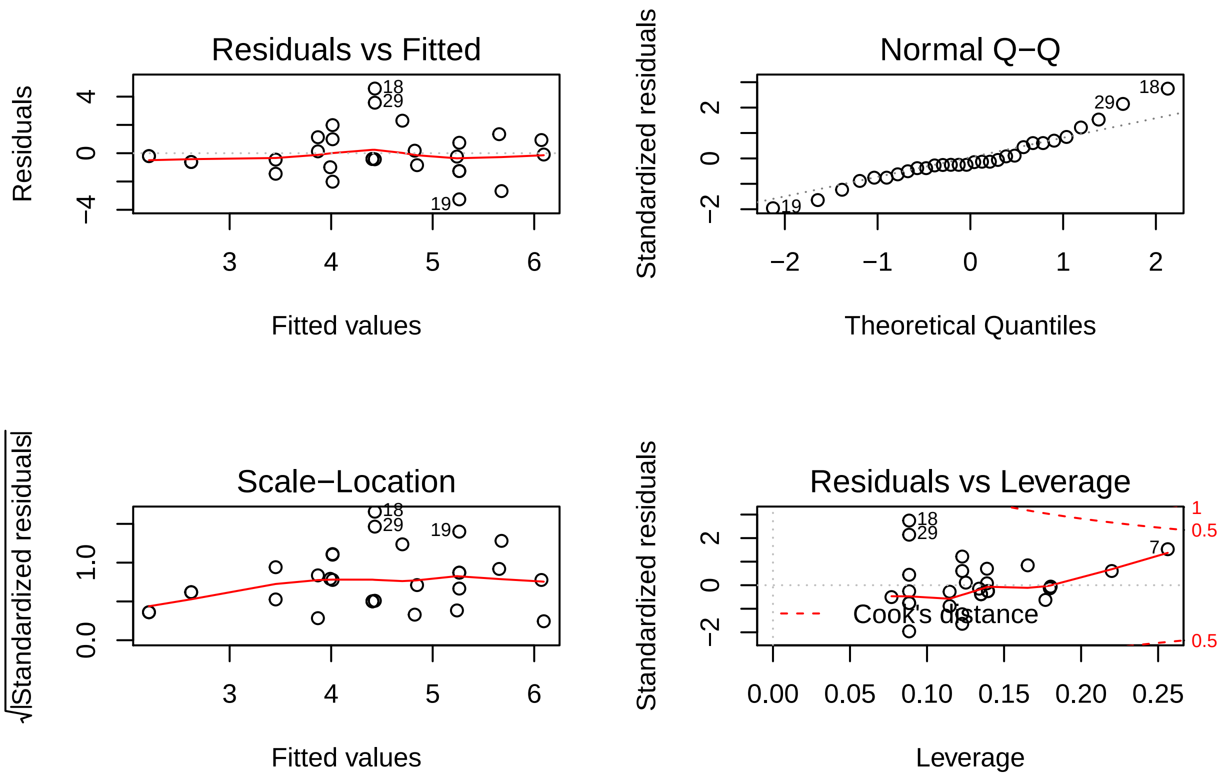
```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = libido ~ partnerLibido + dose, data = viagraData)
##
## Linear Hypotheses:
##                             Estimate Std. Error t value Pr(>|t|)
## Low Dose - Placebo == 0       1.7857     0.8494   2.102   0.1088
## Hight Dose - Placebo == 0     2.2249     0.8028   2.771   0.0266 *
## Hight Dose - Low Dose == 0    0.4392     0.8112   0.541   0.8516
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
confint(postHocs)
```

```
##
##   Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = libido ~ partnerLibido + dose, data = viagraData)
##
## Quantile = 2.4832
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##                             Estimate lwr     upr
## Low Dose - Placebo == 0      1.7857  -0.3235  3.8948
## Hight Dose - Placebo == 0    2.2249   0.2313  4.2185
## Hight Dose - Low Dose == 0   0.4392  -1.5753  2.4537
```

plots for the ANCOVA model

```r
par(mfrow=c(2,2))
plot(viagraModel)
```



Test the asumptions of ANCOVA To test the assumption of homogeneity of regression slopes we need to run the ANCOVA again, but include the interaction between the covariate and predictor variable. We can do this in three ways. The first is to re-specify the whole model from scratch. We can include interaction terms by linking variable names with a colon. For example, the interaction of partnerLibido and dose would be written in R as partnerLibido:dose (or indeed dose:partnerLibido, it doesn't matter). Therefore, to include this interaction in an ANCOVA model we could execute: