ch05

```r
######################## Example: Itching ########################
itch <- read.csv("http://statacumen.com/teach/ADA2/ADA2_notes_Ch05_itch.csv")

library(xtable)
xtab.out <- xtable(itch, digits=4)
print(xtab.out, floating=FALSE, math.style.negative=TRUE)
```

```
## % latex table generated in R 3.4.4 by xtable 1.8-2 package
## % Mon Feb 18 22:46:36 2019
## \begin{tabular}{rrrrrrrrr}
##   \hline
##  & Patient & Nodrug & Placebo & Papv & Morp & Amino & Pento & Tripel \\
##   \hline
## 1 &     1 &   174 &   263 &   105 &   199 &   141 &   108 &   141 \\
##   2 &     2 &   224 &   213 &   103 &   143 &   168 &   341 &   184 \\
##   3 &     3 &   260 &   231 &   145 &   113 &    78 &   159 &   125 \\
##   4 &     4 &   255 &   291 &   103 &   225 &   164 &   135 &   227 \\
##   5 &     5 &   165 &   168 &   144 &   176 &   127 &   239 &   194 \\
##   6 &     6 &   237 &   121 &    94 &   144 &   114 &   136 &   155 \\
##   7 &     7 &   191 &   137 &    35 &    87 &    96 &   140 &   121 \\
##   8 &     8 &   100 &   102 &   133 &   120 &   222 &   134 &   129 \\
##   9 &     9 &   115 &    89 &    83 &   100 &   165 &   185 &    79 \\
##   10 &    10 &   189 &   433 &   237 &   173 &   168 &   188 &   317 \\
##    \hline
## \end{tabular}
```

```r
library(reshape2)
itch.long <- melt(itch
                , id.vars       = "Patient"
                , variable.name = "Treatment"
                , value.name    = "Seconds"
              )
str(itch.long)
```

```
## 'data.frame':    70 obs. of  3 variables:
##  $ Patient  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Treatment: Factor w/ 7 levels "Nodrug","Placebo",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Seconds  : int  174 224 260 255 165 237 191 100 115 189 ...
```

```r
head(itch.long, 3)
```

```
##   Patient Treatment Seconds
## 1       1    Nodrug     174
## 2       2    Nodrug     224
## 3       3    Nodrug     260
```

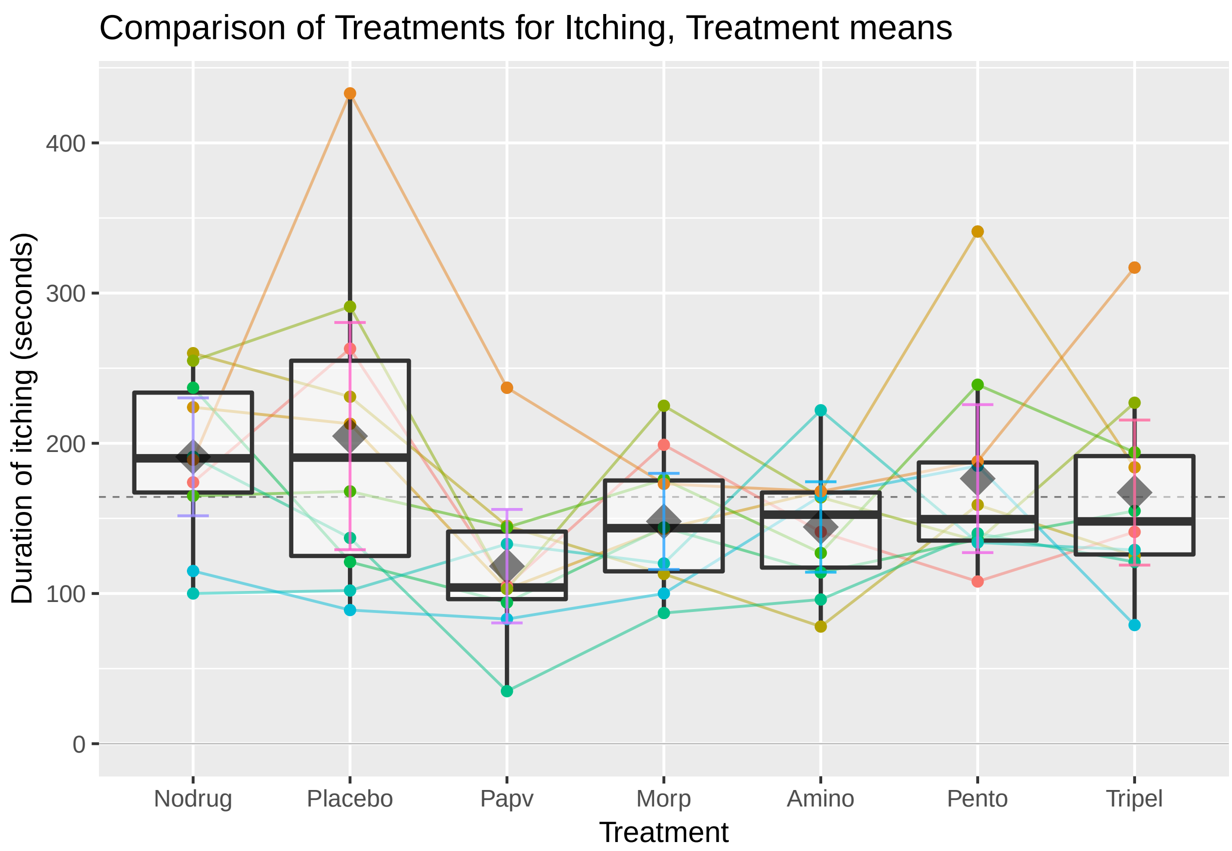
```r
tail(itch.long, 3)
```

```
##    Patient Treatment Seconds
## 68       8    Tripel     129
## 69       9    Tripel      79
## 70      10    Tripel     317
```

```r
# make Patient a factor variable
itch.long$Patient <- factor(itch.long$Patient)
str(itch.long)
```

```
## 'data.frame':    70 obs. of  3 variables:
##  $ Patient  : Factor w/ 10 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Treatment: Factor w/ 7 levels "Nodrug","Placebo",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Seconds  : int  174 224 260 255 165 237 191 100 115 189 ...
```

```r
# Plot the data using ggplot
library(ggplot2)
p <- ggplot(itch.long, aes(x = Treatment, y = Seconds))
# plot a reference line for the global mean (assuming no groups)
p <- p + geom_hline(aes(yintercept = 0),
                    colour = "black", linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_hline(aes(yintercept = mean(Seconds)),
                    colour = "black", linetype = "dashed", size = 0.3, alpha = 0.5)
# colored line for each patient
p <- p + geom_line(aes(group = Patient, colour = Patient), alpha = 0.5)
# boxplot, size=.75 to stand out behind CI
p <- p + geom_boxplot(size = 0.75, alpha = 0.5)
# points for observed data
p <- p + geom_point(aes(colour = Patient))
# diamond at mean for each group
p <- p + stat_summary(fun.y = mean, geom = "point", shape = 18, size = 6,
                      alpha = 0.5)
# confidence limits based on normal distribution
p <- p + stat_summary(fun.data = "mean_cl_normal", geom = "errorbar",
                      width = .2, aes(colour=Treatment), alpha = 0.8)
p <- p + labs(title = "Comparison of Treatments for Itching, Treatment means")
p <- p + ylab("Duration of itching (seconds)")
# removes legend
p <- p + theme(legend.position="none")
print(p)
```



```
lm.s.t.p <- lm(Seconds ~ Treatment + Patient, data = itch.long)
library(car)
```

```
## Loading required package: carData
```

```
Anova(lm.s.t.p, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: Seconds
##             Sum Sq Df F value    Pr(>F)
## (Intercept) 155100  1 50.1133 3.065e-09 ***
## Treatment    53013  6  2.8548  0.017303 *
## Patient     103280  9  3.7078  0.001124 **
## Residuals   167130 54
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm.s.t.p)
```

```
##
## Call:
## lm(formula = Seconds ~ Treatment + Patient, data = itch.long)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -81.286 -34.800  -8.393  30.900 148.914
##
## Coefficients:
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        188.286     26.598   7.079 3.07e-09 ***
## TreatmentPlacebo    13.800     24.880   0.555  0.58141
## TreatmentPapv      -72.800     24.880  -2.926  0.00501 **
## TreatmentMorp      -43.000     24.880  -1.728  0.08965 .
## TreatmentAmino     -46.700     24.880  -1.877  0.06592 .
## TreatmentPento     -14.500     24.880  -0.583  0.56245
## TreatmentTripel    -23.800     24.880  -0.957  0.34303
## Patient2            35.000     29.737   1.177  0.24436
## Patient3            -2.857     29.737  -0.096  0.92381
## Patient4            38.429     29.737   1.292  0.20176
## Patient5            11.714     29.737   0.394  0.69518
## Patient6           -18.571     29.737  -0.625  0.53491
## Patient7           -46.286     29.737  -1.557  0.12543
## Patient8           -27.286     29.737  -0.918  0.36292
## Patient9           -45.000     29.737  -1.513  0.13604
## Patient10           82.000     29.737   2.758  0.00793 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.63 on 54 degrees of freedom
## Multiple R-squared:  0.4832, Adjusted R-squared:  0.3397
## F-statistic: 3.367 on 15 and 54 DF,  p-value: 0.00052
```

```
# multcomp has functions for multiple comparisons
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
```

```
# Use the ANOVA object and run a "General Linear Hypothesis Test"
# specifying a linfct (linear function) to be tested.
# The mpc (multiple comparison) specifies the factor and method.
# Here: correcting over Treatment using Tukey contrast corrections.
glht.itch.t <- glht(aov(lm.s.t.p), linfct = mcp(Treatment = "Tukey"))
summary(glht.itch.t)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = lm.s.t.p)
##
## Linear Hypotheses:
```

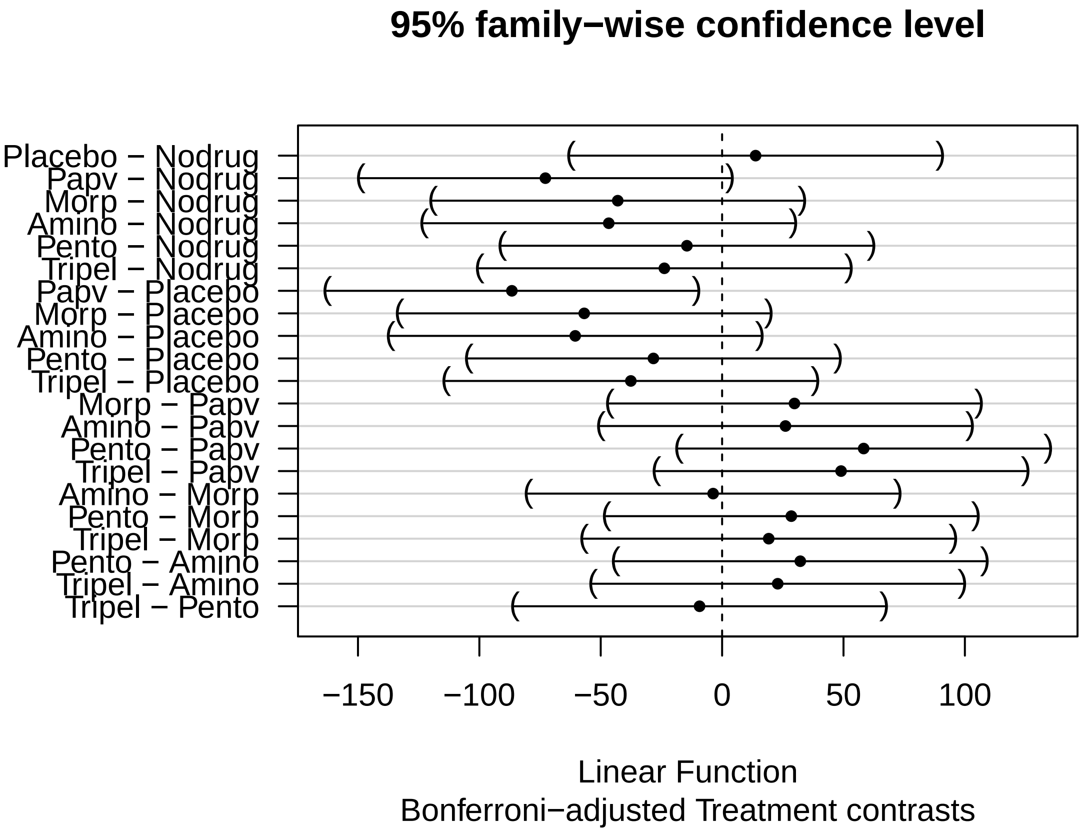
```
##                          Estimate Std. Error t value Pr(>|t|)
## Placebo - Nodrug == 0    13.80      24.88   0.555   0.9978
## Papv - Nodrug == 0      -72.80      24.88  -2.926   0.0702 .
## Morp - Nodrug == 0      -43.00      24.88  -1.728   0.6008
## Amino - Nodrug == 0     -46.70      24.88  -1.877   0.5040
## Pento - Nodrug == 0     -14.50      24.88  -0.583   0.9971
## Tripel - Nodrug == 0    -23.80      24.88  -0.957   0.9610
## Papv - Placebo == 0     -86.60      24.88  -3.481   0.0163 *
## Morp - Placebo == 0     -56.80      24.88  -2.283   0.2710
## Amino - Placebo == 0    -60.50      24.88  -2.432   0.2056
## Pento - Placebo == 0    -28.30      24.88  -1.137   0.9135
## Tripel - Placebo == 0   -37.60      24.88  -1.511   0.7369
## Morp - Papv == 0         29.80      24.88   1.198   0.8920
## Amino - Papv == 0        26.10      24.88   1.049   0.9398
## Pento - Papv == 0        58.30      24.88   2.343   0.2430
## Tripel - Papv == 0       49.00      24.88   1.969   0.4455
## Amino - Morp == 0        -3.70      24.88  -0.149   1.0000
## Pento - Morp == 0        28.50      24.88   1.146   0.9108
## Tripel - Morp == 0       19.20      24.88   0.772   0.9867
## Pento - Amino == 0       32.20      24.88   1.294   0.8516
## Tripel - Amino == 0      22.90      24.88   0.920   0.9677
## Tripel - Pento == 0      -9.30      24.88  -0.374   0.9998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```r
summary(glht.itch.t, test = adjusted("bonferroni"))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = lm.s.t.p)
##
## Linear Hypotheses:
##                          Estimate Std. Error t value Pr(>|t|)
## Placebo - Nodrug == 0    13.80      24.88   0.555    1.000
## Papv - Nodrug == 0      -72.80      24.88  -2.926    0.105
## Morp - Nodrug == 0      -43.00      24.88  -1.728    1.000
## Amino - Nodrug == 0     -46.70      24.88  -1.877    1.000
## Pento - Nodrug == 0     -14.50      24.88  -0.583    1.000
## Tripel - Nodrug == 0    -23.80      24.88  -0.957    1.000
## Papv - Placebo == 0     -86.60      24.88  -3.481    0.021 *
## Morp - Placebo == 0     -56.80      24.88  -2.283    0.554
## Amino - Placebo == 0    -60.50      24.88  -2.432    0.386
## Pento - Placebo == 0    -28.30      24.88  -1.137    1.000
## Tripel - Placebo == 0   -37.60      24.88  -1.511    1.000
## Morp - Papv == 0         29.80      24.88   1.198    1.000
## Amino - Papv == 0        26.10      24.88   1.049    1.000
## Pento - Papv == 0        58.30      24.88   2.343    0.479
## Tripel - Papv == 0       49.00      24.88   1.969    1.000
## Amino - Morp == 0        -3.70      24.88  -0.149    1.000
## Pento - Morp == 0        28.50      24.88   1.146    1.000
```

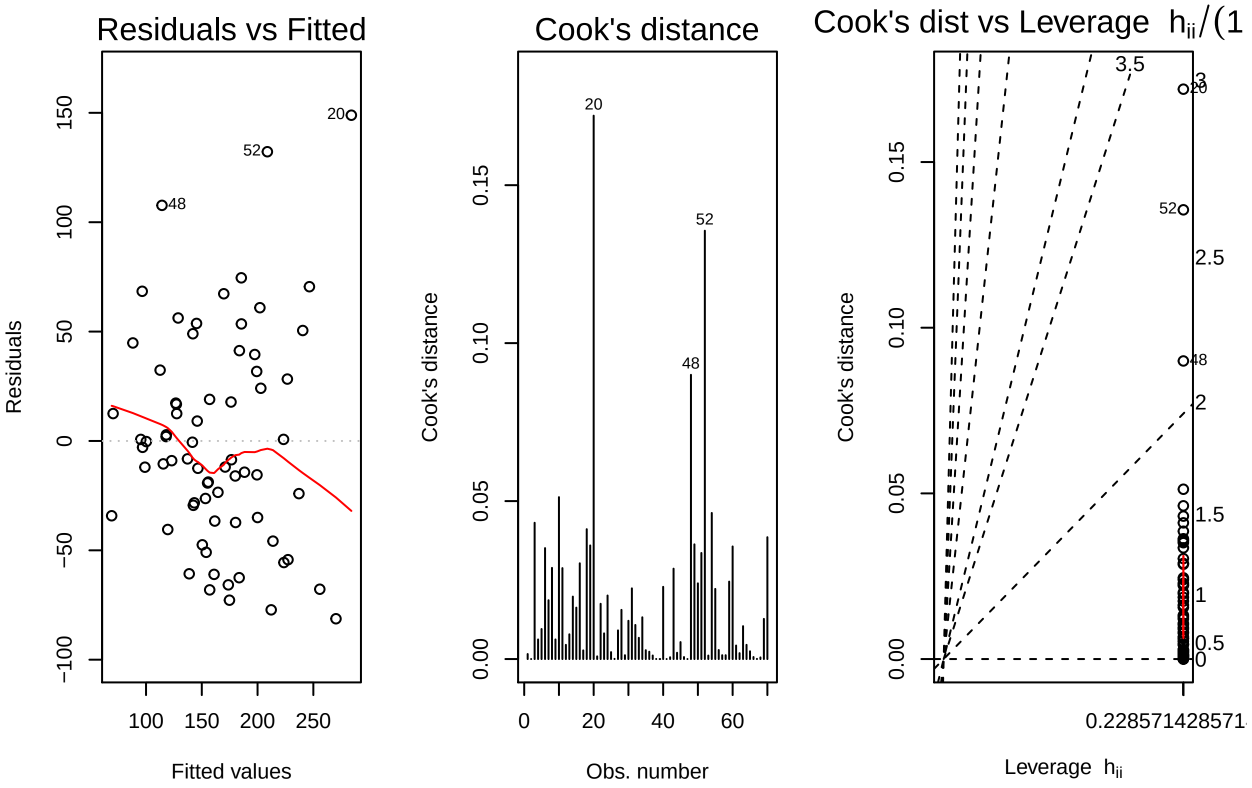
```
## Tripel - Morp == 0       19.20      24.88   0.772    1.000
## Pento - Amino == 0       32.20      24.88   1.294    1.000
## Tripel - Amino == 0      22.90      24.88   0.920    1.000
## Tripel - Pento == 0      -9.30      24.88  -0.374    1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)
```

```r
# plot the summary
  op <- par(no.readonly = TRUE) # the whole list of settable par's.
  # make wider left margin to fit contrast labels
  par(mar = c(5, 10, 4, 2) + 0.1)  # order is c(bottom, left, top, right)
# plot bonferroni-corrected difference intervals
plot(summary(glht.itch.t, test = adjusted("bonferroni"))
    , sub="Bonferroni-adjusted Treatment contrasts")
```



```r
  par(op) # reset plotting options




### Code for the less interesting contrasts.
### Testing multiple factors may be of interest in other problems.
### Note that the first block of code below corrects the p-values
###   for all the tests done for both factors together,
###   that is, the Bonferroni-corrected significance level is (alpha / (t + p))
###   where t = number of treatment comparisons
###   and   p = number of patient   comparisons.
```

```
# # correcting over Treatment and Patient
# glht.itch.tp <- glht(aov(lm.s.t.p), linfct = mcp(Treatment = "Tukey"
#                                                , Patient   = "Tukey"))
# summary(glht.itch.tp, test = adjusted("bonferroni"))
# plot(summary(glht.itch.tp, test = adjusted("bonferroni")))

# # correcting over Patient, only
# glht.itch.p <- glht(aov(lm.s.t.p), linfct = mcp(Patient = "Tukey"))
# summary(glht.itch.p, test = adjusted("bonferroni"))
# plot(summary(glht.itch.p, test = adjusted("bonferroni")))



# plot diagnostics
par(mfrow=c(1,3))
plot(lm.s.t.p, which = c(1,4,6))
```



```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05plot1.jpg")
```

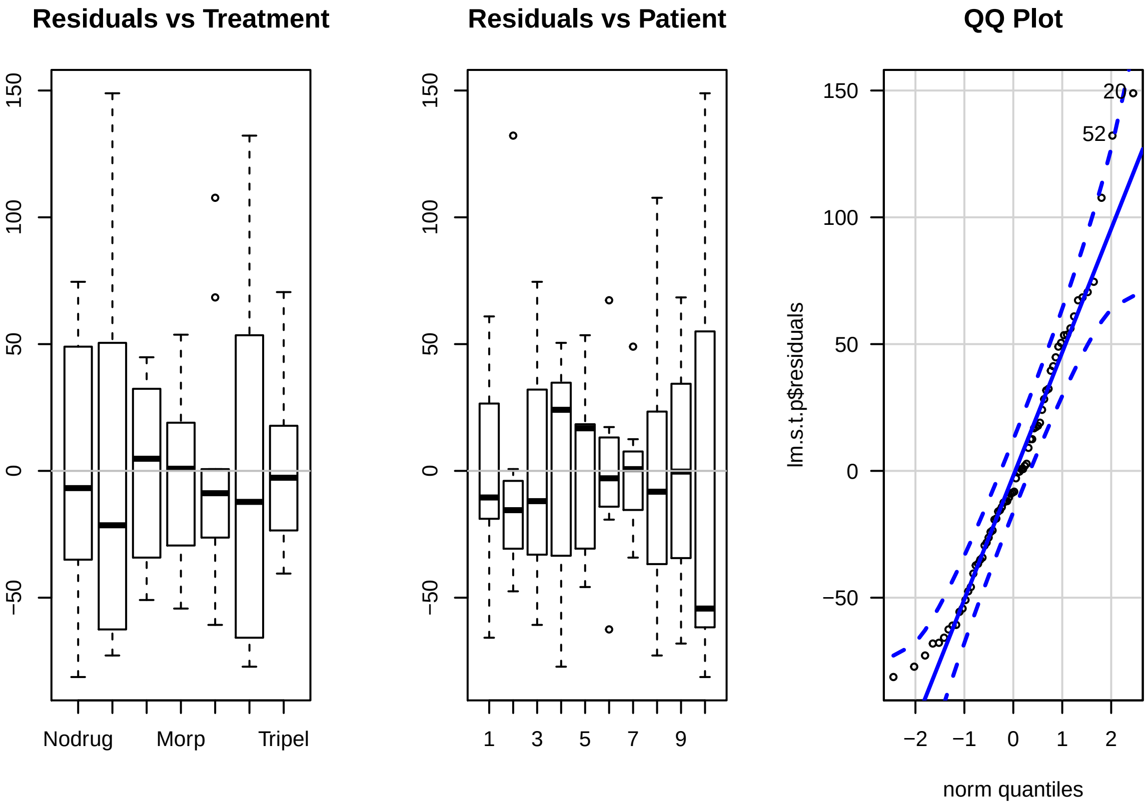
```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
par(mfrow=c(1,3))
plot(itch.long$Treatment, lm.s.t.p$residuals, main="Residuals vs Treatment")
  # horizontal line at zero
```

```
  abline(h = 0, col = "gray75")

plot(itch.long$Patient, lm.s.t.p$residuals, main="Residuals vs Patient")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

# Normality of Residuals
library(car)
qqPlot(lm.s.t.p$residuals, las = 1, main="QQ Plot")
```



```
## [1] 20 52
```

```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05plot2.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
## residuals vs order of data
#plot(lm.s.t.p$residuals, main="Residuals vs Order of data")
#  # horizontal line at zero
#  abline(h = 0, col = "gray75")



# Friedman test for differences between groups conditional on blocks.
```

```
#   The formula is of the form a ~ b | c,
#     where a, b and c give the data values (a)
#     and corresponding groups (b) and blocks (c), respectively.
friedman.test(Seconds ~ Treatment | Patient, data = itch.long)
```

```
##
##  Friedman rank sum test
##
## data:  Seconds and Treatment and Patient
## Friedman chi-squared = 14.887, df = 6, p-value = 0.02115
```

```
# Quade test is very similar to the Friedman test (compare the help pages).
quade.test(Seconds ~ Treatment | Patient, data = itch.long)
```

```
##
##  Quade test
##
## data:  Seconds and Treatment and Patient
## Quade F = 3.7321, num df = 6, denom df = 54, p-value = 0.003542
```

```
############################ Example: Beetles  ################################################
beetles <- read.table("http://statacumen.com/teach/ADA2/ADA2_notes_Ch05_beetles.dat"
                    , header = TRUE)
beetles
```

```
##    dose insecticide   t1   t2   t3   t4
## 1     1           A 0.31 0.45 0.46 0.43
## 2     1           B 0.82 1.10 0.88 0.72
## 3     1           C 0.43 0.45 0.63 0.76
## 4     1           D 0.45 0.71 0.66 0.62
## 5     2           A 0.36 0.29 0.40 0.23
## 6     2           B 0.92 0.61 0.49 1.24
## 7     2           C 0.44 0.35 0.31 0.40
## 8     2           D 0.56 1.02 0.71 0.38
## 9     3           A 0.22 0.21 0.18 0.23
## 10    3           B 0.30 0.37 0.38 0.29
## 11    3           C 0.23 0.25 0.24 0.22
## 12    3           D 0.30 0.36 0.31 0.33
```

```
# make dose a factor variable and label the levels
beetles$dose <- factor(beetles$dose, labels = c("low","medium","high"))
beetles$dose
```

```
##  [1] low    low    low    low    medium medium medium medium high   high
## [11] high   high
## Levels: low medium high
```

```
library(reshape2)
beetles.long <- melt(beetles
                , id.vars       = c("dose", "insecticide")
                , variable.name = "number"
                , value.name    = "hours10"
             )
str(beetles.long)
```

```
## 'data.frame':    48 obs. of  4 variables:
##  $ dose       : Factor w/ 3 levels "low","medium",..: 1 1 1 1 2 2 2 2 3 3 ...
```

```
##  $ insecticide: Factor w/ 4 levels "A","B","C","D": 1 2 3 4 1 2 3 4 1 2 ...
##  $ number     : Factor w/ 4 levels "t1","t2","t3",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ hours10    : num  0.31 0.82 0.43 0.45 0.36 0.92 0.44 0.56 0.22 0.3 ...
```

```r
head(beetles.long)
```

```
##     dose insecticide number hours10
## 1    low           A     t1    0.31
## 2    low           B     t1    0.82
## 3    low           C     t1    0.43
## 4    low           D     t1    0.45
## 5 medium           A     t1    0.36
## 6 medium           B     t1    0.92
```

```r
library(plyr)
# Calculate the cell means for each (dose, insecticide) combination
mean(beetles.long[, "hours10"])
```

```
## [1] 0.479375
```

```r
beetles.mean    <- ddply(beetles.long, .(), summarise, m = mean(hours10))
beetles.mean
```

```
##    .id        m
## 1 <NA> 0.479375
```

```r
beetles.mean.d  <- ddply(beetles.long, .(dose), summarise, m = mean(hours10))
beetles.mean.d
```

```
##     dose        m
## 1    low 0.617500
## 2 medium 0.544375
## 3   high 0.276250
```

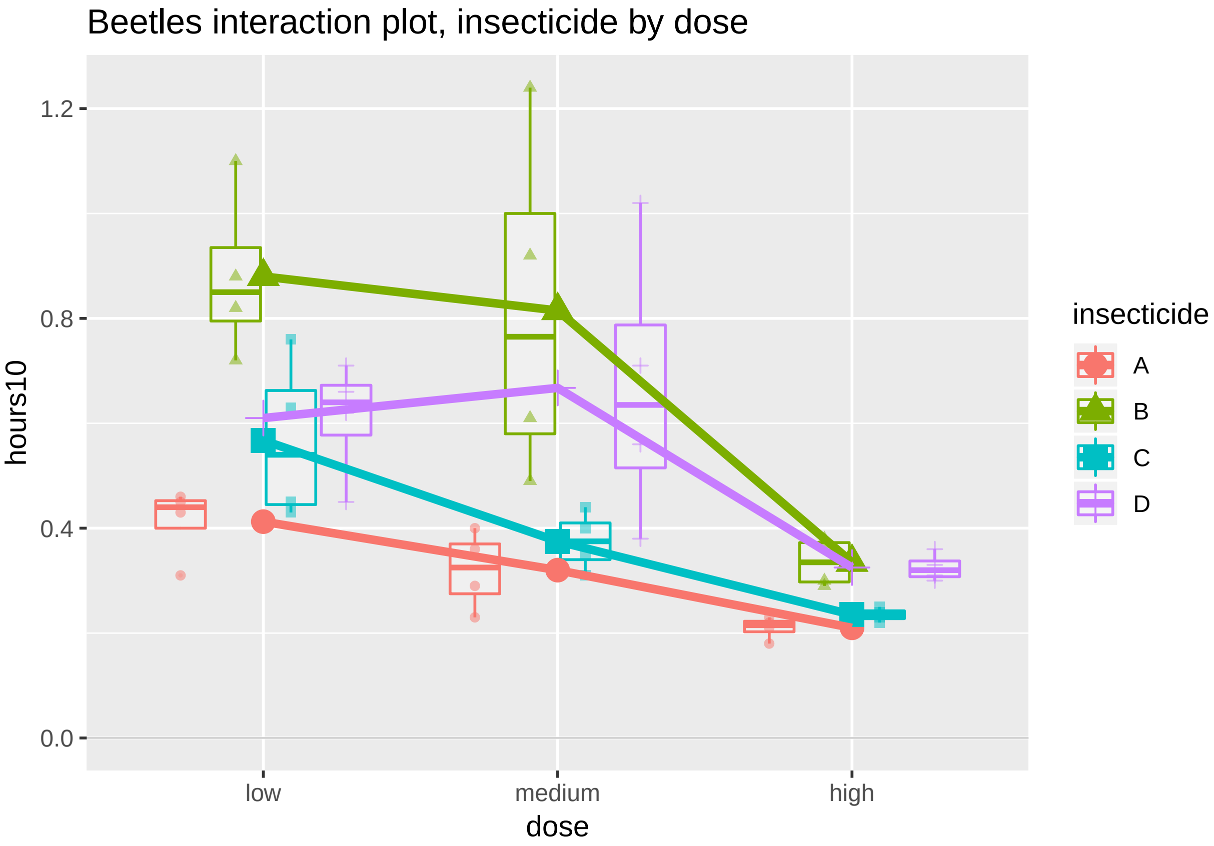
```r
beetles.mean.i  <- ddply(beetles.long, .(insecticide), summarise, m = mean(hours10))
beetles.mean.i
```

```
##   insecticide         m
## 1           A 0.3141667
## 2           B 0.6766667
## 3           C 0.3925000
## 4           D 0.5341667
```

```r
beetles.mean.di <- ddply(beetles.long, .(dose,insecticide), summarise, m = mean(hours10))
beetles.mean.di
```

```
##      dose insecticide      m
## 1     low           A 0.4125
## 2     low           B 0.8800
## 3     low           C 0.5675
## 4     low           D 0.6100
## 5  medium           A 0.3200
## 6  medium           B 0.8150
## 7  medium           C 0.3750
## 8  medium           D 0.6675
## 9    high           A 0.2100
## 10   high           B 0.3350
## 11   high           C 0.2350
## 12   high           D 0.3250
```

```
# Interaction plots, ggplot
library(ggplot2)
p <- ggplot(beetles.long, aes(x = dose, y = hours10, colour = insecticide, shape = insecticide))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = beetles.mean.di, aes(y = m), size = 4)
p <- p + geom_line(data = beetles.mean.di, aes(y = m, group = insecticide), size = 1.5)
p <- p + labs(title = "Beetles interaction plot, insecticide by dose")
print(p)
```



```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05plot3.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
p <- ggplot(beetles.long, aes(x = insecticide, y = hours10, colour = dose, shape = dose))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = beetles.mean.di, aes(y = m), size = 4)
p <- p + geom_line(data = beetles.mean.di, aes(y = m, group = dose), size = 1.5)
```

```
p <- p + labs(title = "Beetles interaction plot, dose by insecticide")
print(p)
```

```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05plot4.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
# Interaction plots, base graphics

interaction.plot(beetles.long$dose, beetles.long$insecticide, beetles.long$hours10
  , main = "Beetles interaction plot, insecticide by dose")
interaction.plot(beetles.long$insecticide, beetles.long$dose, beetles.long$hours10
  , main = "Beetles interaction plot, dose by insecticide")



lm.h.d.i.di <- lm(hours10 ~ dose + insecticide + dose:insecticide
                , data = beetles.long)
#or you can simply use lm(hours10 ~ dose:insecticide, data = beetles.long)
# lm.h.d.i.di <- lm(hours10 ~ dose*insecticide, data = beetles.long) # equivalent
library(car)
Anova(lm.h.d.i.di, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: hours10
##                    Sum Sq Df F value    Pr(>F)
## (Intercept)       0.68063  1 30.6004 2.937e-06 ***
## dose              0.08222  2  1.8482 0.1721570
## insecticide       0.45395  3  6.8031 0.0009469 ***
## dose:insecticide  0.25014  6  1.8743 0.1122506
## Residuals         0.80072 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(lm.h.d.i.di)
```

```
##
## Call:
## lm(formula = hours10 ~ dose + insecticide + dose:insecticide,
##     data = beetles.long)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32500 -0.04875  0.00500  0.04312  0.42500
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.41250    0.07457   5.532 2.94e-06 ***
## dosemedium               -0.09250    0.10546  -0.877   0.3862
## dosehigh                 -0.20250    0.10546  -1.920   0.0628 .
## insecticideB              0.46750    0.10546   4.433 8.37e-05 ***
## insecticideC              0.15500    0.10546   1.470   0.1503
## insecticideD              0.19750    0.10546   1.873   0.0692 .
## dosemedium:insecticideB   0.02750    0.14914   0.184   0.8547
## dosehigh:insecticideB    -0.34250    0.14914  -2.297   0.0276 *
## dosemedium:insecticideC  -0.10000    0.14914  -0.671   0.5068
## dosehigh:insecticideC    -0.13000    0.14914  -0.872   0.3892
## dosemedium:insecticideD   0.15000    0.14914   1.006   0.3212
## dosehigh:insecticideD    -0.08250    0.14914  -0.553   0.5836
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1491 on 36 degrees of freedom
## Multiple R-squared:  0.7335, Adjusted R-squared:  0.6521
## F-statistic:  9.01 on 11 and 36 DF,  p-value: 1.986e-07
```

```r
lm.h.d.i <- update(lm.h.d.i.di, ~ . - dose:insecticide )
library(car)
Anova(lm.h.d.i, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: hours10
##              Sum Sq Df F value    Pr(>F)
## (Intercept) 1.63654  1  65.408 4.224e-10 ***
## dose        1.03301  2  20.643 5.704e-07 ***
## insecticide 0.92121  3  12.273 6.697e-06 ***
```

```
## Residuals   1.05086 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm.h.d.i)
```

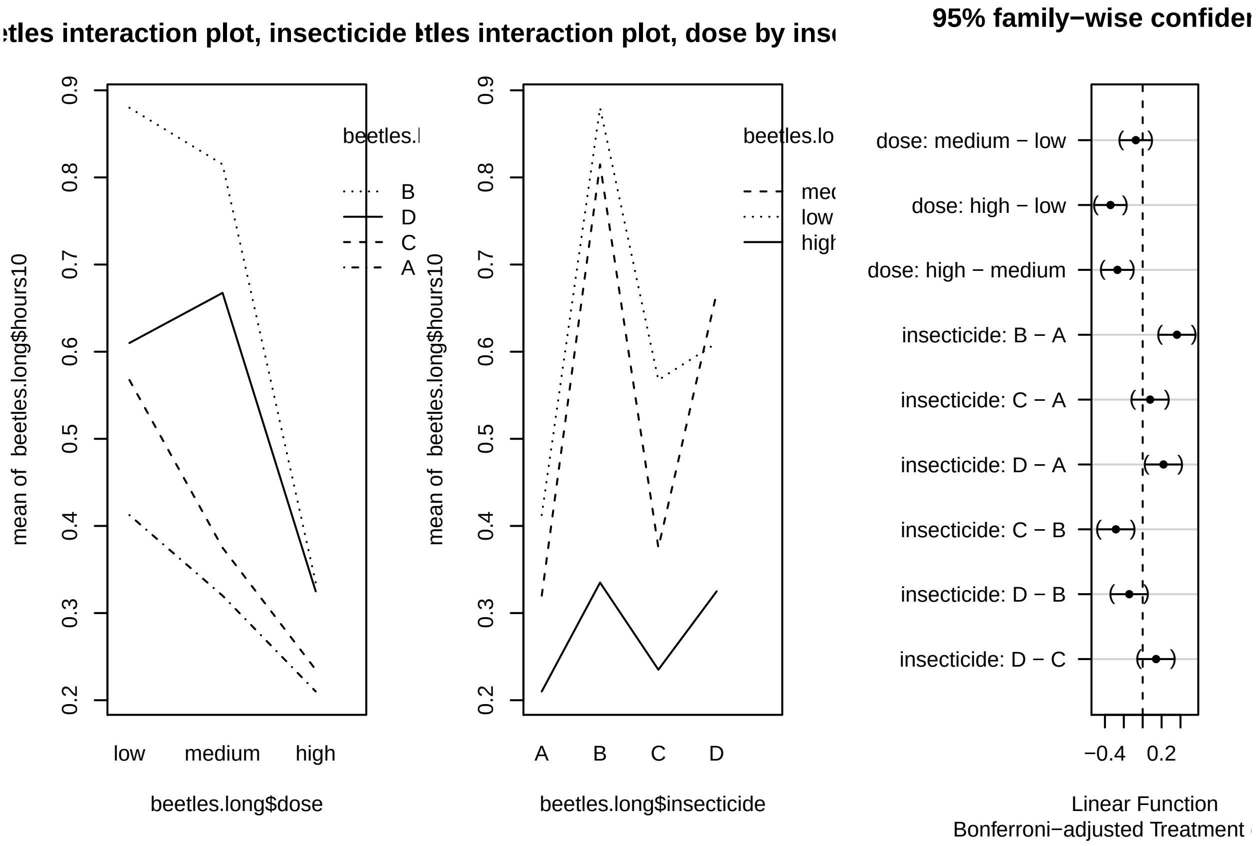
```
##
## Call:
## lm(formula = hours10 ~ dose + insecticide, data = beetles.long)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25167 -0.09625 -0.01490  0.06177  0.49833
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.45229    0.05592   8.088 4.22e-10 ***
## dosemedium    -0.07313    0.05592  -1.308  0.19813
## dosehigh      -0.34125    0.05592  -6.102 2.83e-07 ***
## insecticideB  0.36250    0.06458   5.614 1.43e-06 ***
## insecticideC  0.07833    0.06458   1.213  0.23189
## insecticideD  0.22000    0.06458   3.407  0.00146 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1582 on 42 degrees of freedom
## Multiple R-squared:  0.6503, Adjusted R-squared:  0.6087
## F-statistic: 15.62 on 5 and 42 DF,  p-value: 1.123e-08
```

```
# Testing multiple factors is of interest here.
# Note that the code below corrects the p-values
#   for all the tests done for both factors together,
#   that is, the Bonferroni-corrected significance level is (alpha / (d + i))
#   where d = number of dose        comparisons
#   and   i = number of insecticide comparisons.

# correcting over dose and insecticide
library(multcomp)
glht.beetle.di <- glht(aov(lm.h.d.i), linfct = mcp(dose = "Tukey"
                                                  , insecticide = "Tukey"))
summary(glht.beetle.di, test = adjusted("bonferroni"))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = lm.h.d.i)
##
## Linear Hypotheses:
##                            Estimate Std. Error t value Pr(>|t|)
## dose: medium - low == 0    -0.07313    0.05592  -1.308 1.000000
## dose: high - low == 0      -0.34125    0.05592  -6.102 2.55e-06 ***
## dose: high - medium == 0 -0.26812    0.05592  -4.794 0.000186 ***
```

```
## insecticide: B - A == 0   0.36250    0.06458   5.614 1.28e-05 ***
## insecticide: C - A == 0   0.07833    0.06458   1.213 1.000000
## insecticide: D - A == 0   0.22000    0.06458   3.407 0.013134 *
## insecticide: C - B == 0  -0.28417    0.06458  -4.400 0.000653 ***
## insecticide: D - B == 0  -0.14250    0.06458  -2.207 0.295702
## insecticide: D - C == 0   0.14167    0.06458   2.194 0.304527
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)
```

```
# plot the summary
  op <- par(no.readonly = TRUE) # the whole list of settable par's.
  # make wider left margin to fit contrast labels
  par(mar = c(5, 10, 4, 2) + 0.1)  # order is c(bottom, left, top, right)
# plot bonferroni-corrected difference intervals
plot(summary(glht.beetle.di, test = adjusted("bonferroni"))
    , sub="Bonferroni-adjusted Treatment contrasts")
```



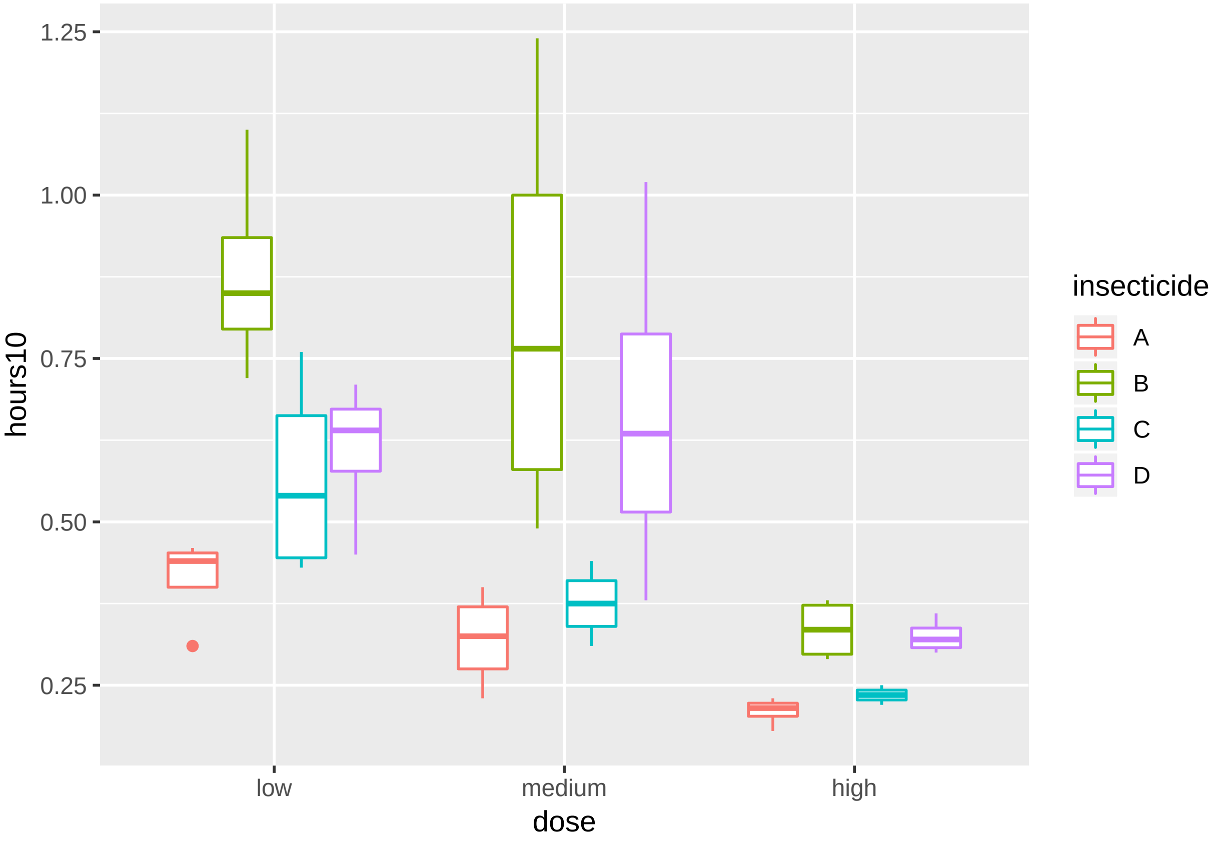
```
  par(op) # reset plotting options
  dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05multicomp.jpg")
```

```
## jpeg
##    3
```

```
  dev.off()
```

```
## pdf
##   2
```

```
#### Example: Beetles, checking assumptions
# boxplots, ggplot
library(ggplot2)
p <- ggplot(beetles.long, aes(x = dose, y = hours10, colour = insecticide))
p <- p + geom_boxplot()
print(p)
```



```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05boxplot.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

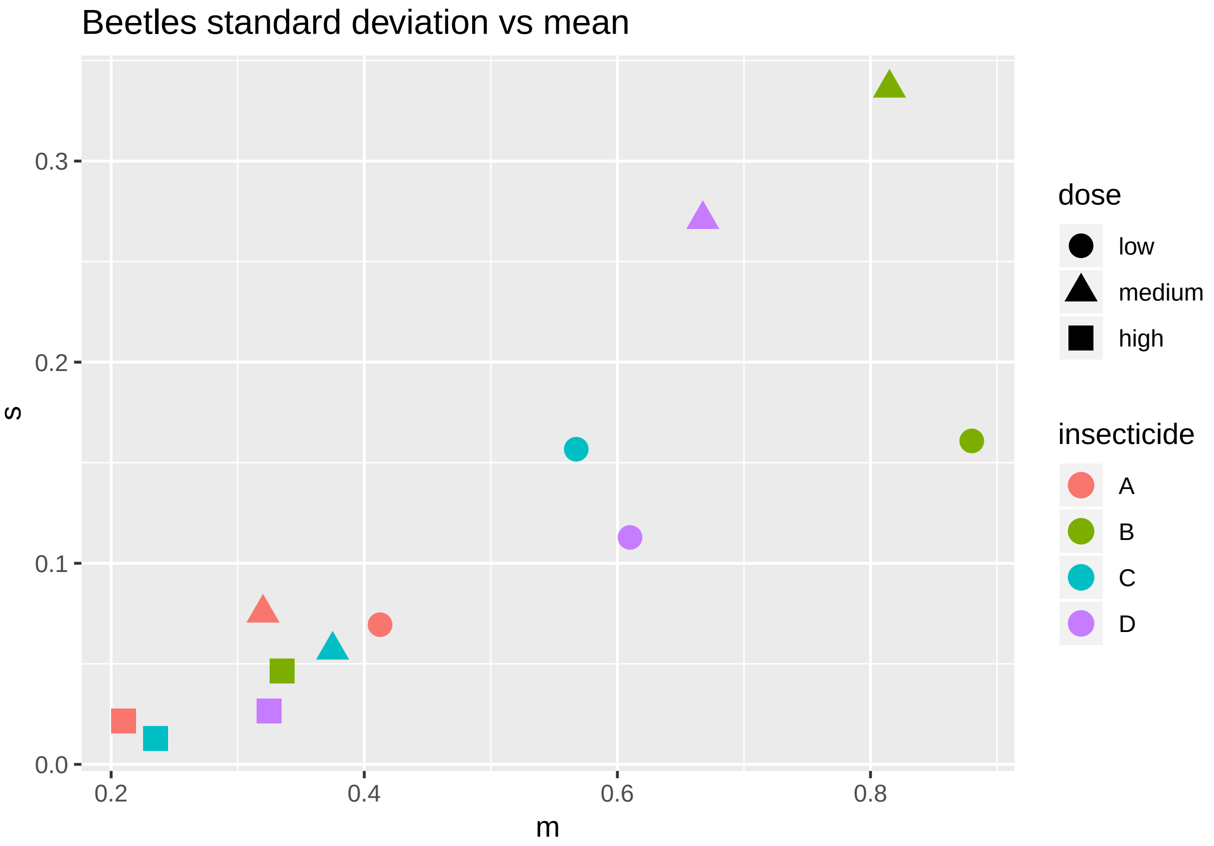
```
## pdf
##   2
```

```
# mean vs sd plot
library(plyr)
# means and standard deviations for each dose/interaction cell
beetles.meansd.di <- ddply(beetles.long, .(dose,insecticide), summarise
                           , m = mean(hours10), s = sd(hours10))
beetles.meansd.di
```

```
##      dose insecticide      m          s
## 1     low           A 0.4125 0.06946222
## 2     low           B 0.8800 0.16083117
## 3     low           C 0.5675 0.15671099
## 4     low           D 0.6100 0.11284207
```

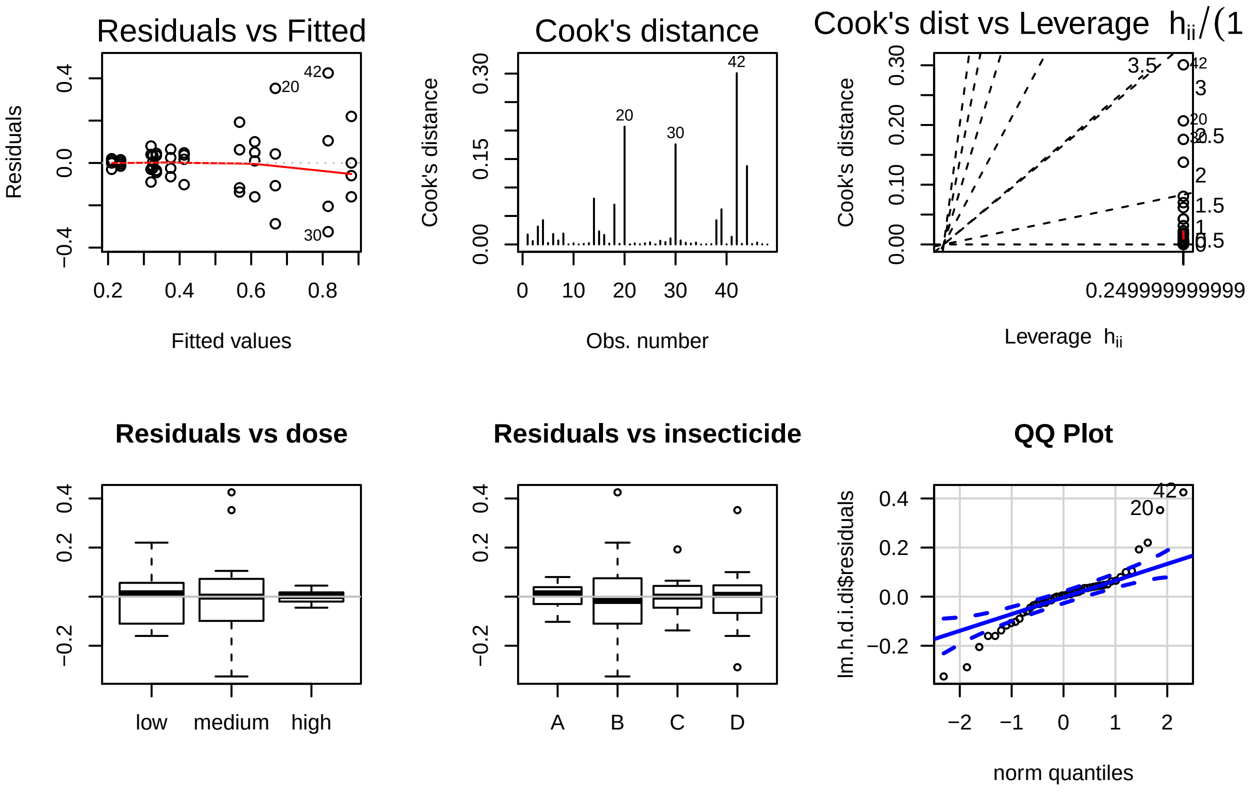
```
## 5  medium           A 0.3200 0.07527727
## 6  medium           B 0.8150 0.33630343
## 7  medium           C 0.3750 0.05686241
## 8  medium           D 0.6675 0.27097048
## 9    high           A 0.2100 0.02160247
## 10   high           B 0.3350 0.04654747
## 11   high           C 0.2350 0.01290994
## 12   high           D 0.3250 0.02645751
```

```
p <- ggplot(beetles.meansd.di, aes(x = m, y = s, shape = dose, colour = insecticide))
p <- p + geom_point(size=4)
p <- p + labs(title = "Beetles standard deviation vs mean")
print(p)
```



```
# interaction model
lm.h.d.i.di <- lm(hours10 ~ dose*insecticide, data = beetles.long)

# plot diagnistics
par(mfrow=c(2,3))
plot(lm.h.d.i.di, which = c(1,4,6))

plot(beetles.long$dose, lm.h.d.i.di$residuals, main="Residuals vs dose")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(beetles.long$insecticide, lm.h.d.i.di$residuals, main="Residuals vs insecticide")
# horizontal line at zero
abline(h = 0, col = "gray75")
```

```
# Normality of Residuals
library(car)
qqPlot(lm.h.d.i.di$residuals, las = 1, main="QQ Plot")
```



```
## [1] 42 20
```

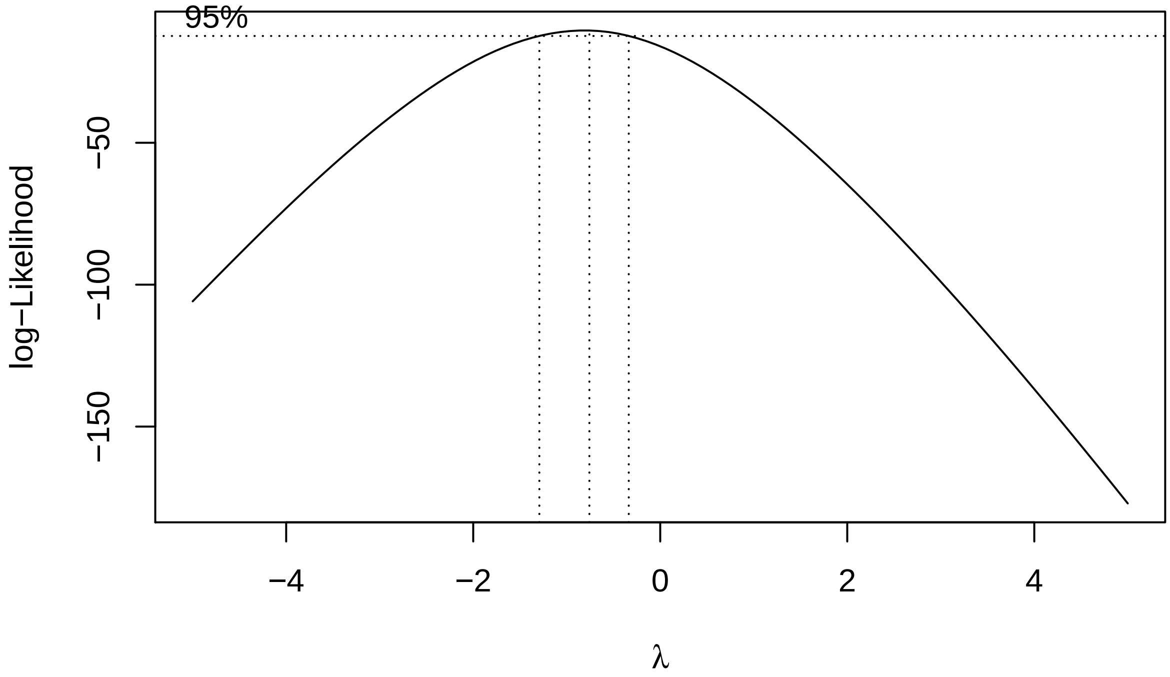
```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05diagfull.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
#transformation
par(mfrow=c(1,1))
library(MASS)
boxcox(lm.h.d.i.di, lambda = seq(-5, 5, length = 10),plotit = TRUE)
```



```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05boxcox.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

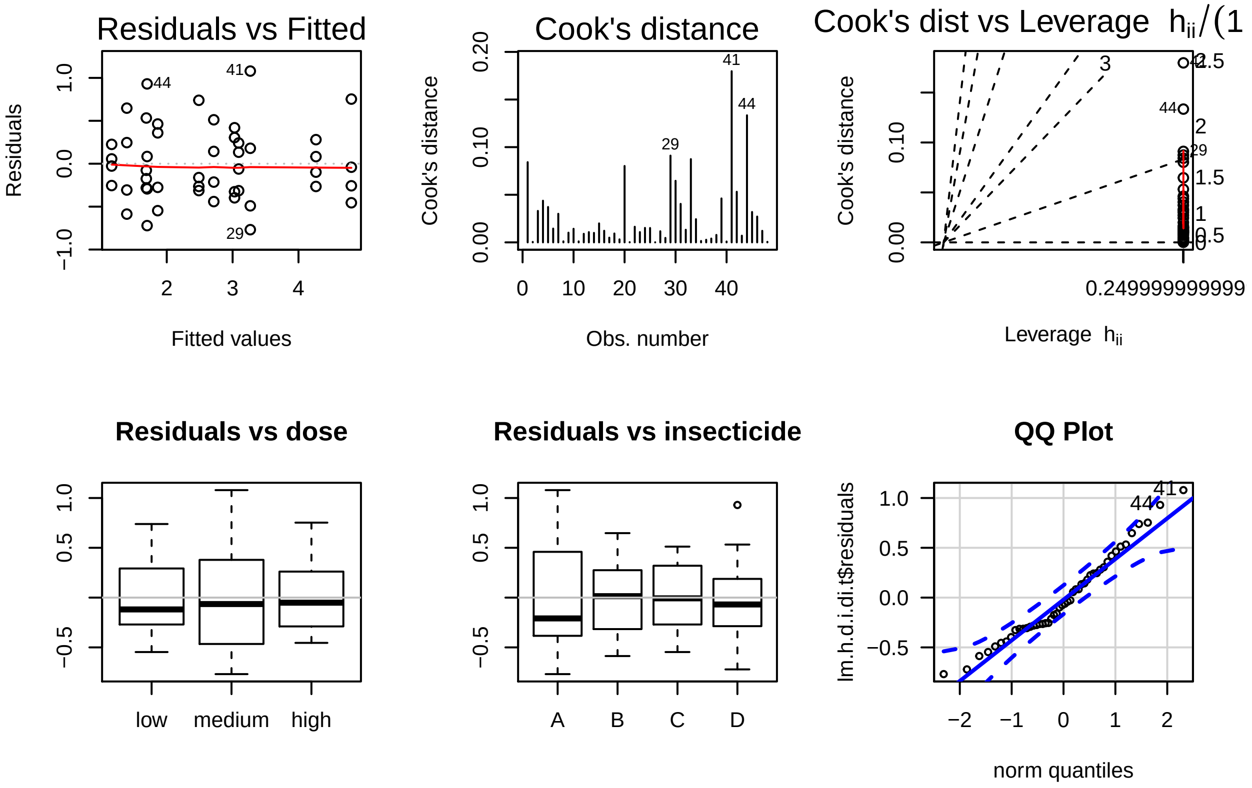
```
## pdf
##   2
```

```
##transformations on $y$

lm.h.d.i.di.t <- lm(1/hours10 ~ dose*insecticide, data = beetles.long)
Anova(lm.h.d.i.di.t, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: 1/hours10
##                    Sum Sq Df  F value    Pr(>F)
## (Intercept)       24.7383  1 103.0395 4.158e-12 ***
## dose              11.1035  2  23.1241 3.477e-07 ***
## insecticide        3.5723  3   4.9598  0.005535 **
## dose:insecticide   1.5708  6   1.0904  0.386733
## Residuals          8.6431 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# plot diagnistics
par(mfrow=c(2,3))
plot(lm.h.d.i.di.t, which = c(1,4,6))

plot(beetles.long$dose, lm.h.d.i.di.t$residuals, main="Residuals vs dose")
# horizontal line at zero
abline(h = 0, col = "gray75")
```

```
plot(beetles.long$insecticide, lm.h.d.i.di.t$residuals, main="Residuals vs insecticide")
# horizontal line at zero
abline(h = 0, col = "gray75")

# Normality of Residuals
library(car)
qqPlot(lm.h.d.i.di.t$residuals, las = 1, main="QQ Plot")
```



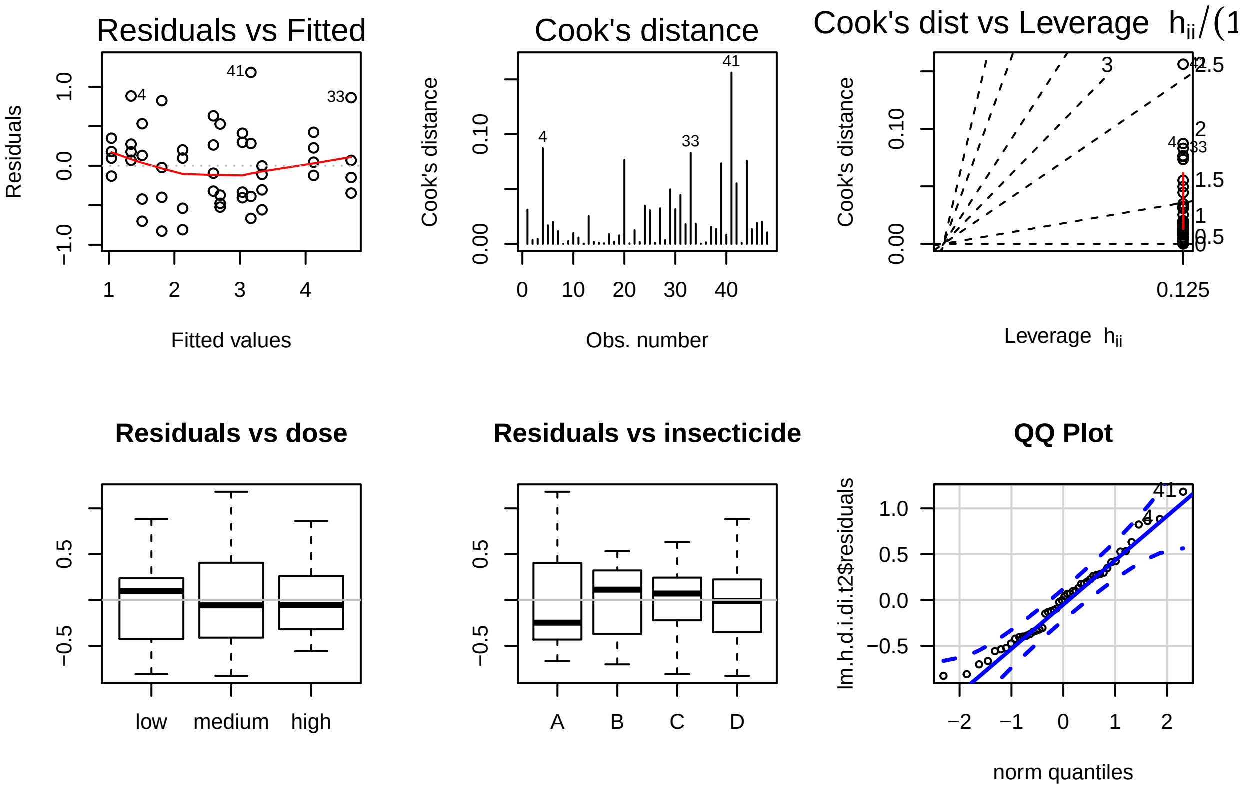
```
## [1] 41 44
```

```
##refit model without interaction term

  lm.h.d.i.di.t2 <- lm(1/hours10 ~ dose+insecticide, data = beetles.long)
  Anova(lm.h.d.i.di.t2, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: 1/hours10
##             Sum Sq Df F value    Pr(>F)
## (Intercept) 58.219  1 239.399 < 2.2e-16 ***
## dose        34.877  2  71.708 2.865e-14 ***
## insecticide 20.414  3  27.982 4.192e-10 ***
## Residuals   10.214 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# plot diagnostics
par(mfrow=c(2,3))
plot(lm.h.d.i.di.t2, which = c(1,4,6))
```

```
plot(beetles.long$dose, lm.h.d.i.di.t2$residuals, main="Residuals vs dose")
# horizontal line at zero
abline(h = 0, col = "gray75")

plot(beetles.long$insecticide, lm.h.d.i.di.t2$residuals, main="Residuals vs insecticide")
# horizontal line at zero
abline(h = 0, col = "gray75")

# Normality of Residuals
library(car)
qqPlot(lm.h.d.i.di.t2$residuals, las = 1, main="QQ Plot")
```



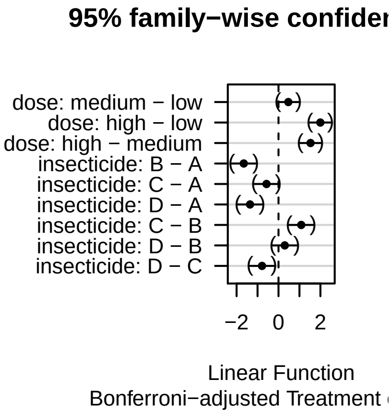
```
## [1] 41  4
```

```
#multiple comparisons
library(multcomp)
glht.beetle.di2 <- glht(aov(lm.h.d.i.di.t2), linfct = mcp(dose = "Tukey"
                                      , insecticide = "Tukey"))
summary(glht.beetle.di2, test = adjusted("bonferroni"))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = lm.h.d.i.di.t2)
##
## Linear Hypotheses:
```

```
##                             Estimate Std. Error t value Pr(>|t|)
## dose: medium - low == 0       0.4686     0.1744   2.688  0.09236 .
## dose: high - low == 0         1.9964     0.1744  11.451 1.52e-13 ***
## dose: high - medium == 0      1.5278     0.1744   8.763 4.46e-10 ***
## insecticide: B - A == 0      -1.6574     0.2013  -8.233 2.39e-09 ***
## insecticide: C - A == 0      -0.5721     0.2013  -2.842  0.06203 .
## insecticide: D - A == 0      -1.3583     0.2013  -6.747 3.01e-07 ***
## insecticide: C - B == 0       1.0853     0.2013   5.391 2.67e-05 ***
## insecticide: D - B == 0       0.2991     0.2013   1.485  1.00000
## insecticide: D - C == 0      -0.7862     0.2013  -3.905  0.00302 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)
```

```
# plot the summary
op <- par(no.readonly = TRUE) # the whole list of settable par's.
# make wider left margin to fit contrast labels
par(mar = c(5, 10, 4, 2) + 0.1)  # order is c(bottom, left, top, right)
# plot bonferroni-corrected difference intervals
plot(summary(glht.beetle.di2, test = adjusted("bonferroni"))
     , sub="Bonferroni-adjusted Treatment contrasts")
par(op) # reset plotting options
```



```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05multicomp2.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
#### Example: Output voltage for batteries
battery <- read.table("http://statacumen.com/teach/ADA2/ADA2_notes_Ch05_battery.dat"
                      , header = TRUE)
battery$material <- factor(battery$material)
battery$temp     <- factor(battery$temp)

library(reshape2)
battery.long <- melt(battery
                     , id.vars       = c("material", "temp")
                     , variable.name = "battery"
```

```
                        , value.name    = "maxvolt"
)
str(battery.long)
```

```
## 'data.frame':    36 obs. of  4 variables:
##  $ material: Factor w/ 3 levels "1","2","3": 1 1 1 2 2 2 3 3 3 1 ...
##  $ temp    : Factor w/ 3 levels "50","65","80": 1 2 3 1 2 3 1 2 3 1 ...
##  $ battery : Factor w/ 4 levels "v1","v2","v3",..: 1 1 1 1 1 1 1 1 1 2 ...
##  $ maxvolt : int  130 34 20 150 136 25 138 174 96 155 ...
```

```
table(battery.long$material,battery.long$temp)  #balanced design
```

```
##
##     50 65 80
##   1  4  4  4
##   2  4  4  4
##   3  4  4  4
```

```
lm.m.m.t.mt <- lm(maxvolt ~ material*temp, data = battery.long)
library(car)
Anova(lm.m.m.t.mt, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: maxvolt
##               Sum Sq Df  F value    Pr(>F)
## (Intercept)    72630  1 107.5664 6.456e-11 ***
## material         886  2   0.6562 0.5268904
## temp           15965  2  11.8223 0.0002052 ***
## material:temp   9614  4   3.5595 0.0186112 *
## Residuals      18231 27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(plyr)
# Calculate the cell means for each (material, temp) combination
battery.mean    <- ddply(battery.long, .(), summarise, m = mean(maxvolt))
battery.mean
```

```
##    .id        m
## 1 <NA> 105.5278
```

```
battery.mean.m  <- ddply(battery.long, .(material), summarise, m = mean(maxvolt))
battery.mean.m
```

```
##   material         m
## 1        1  83.16667
## 2        2 108.33333
## 3        3 125.08333
```

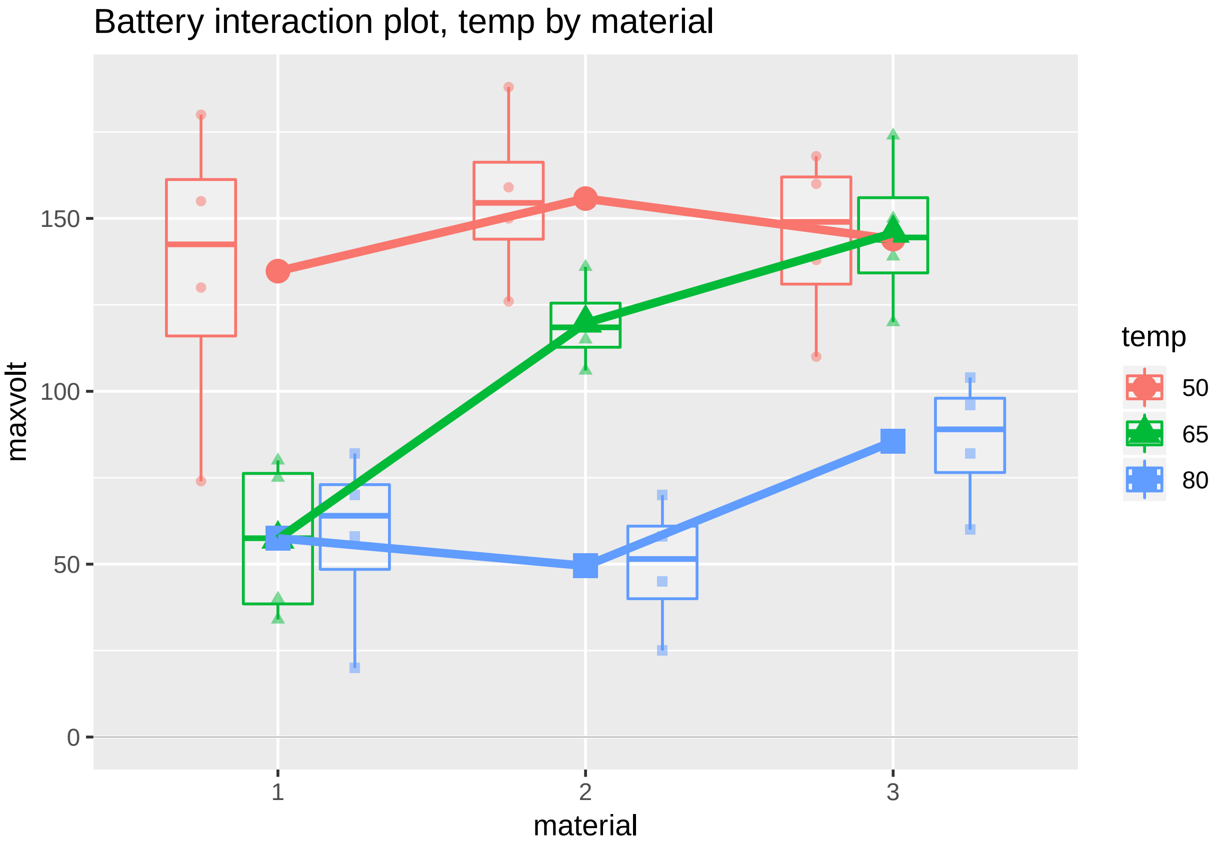
```
battery.mean.t  <- ddply(battery.long, .(temp), summarise, m = mean(maxvolt))
battery.mean.t
```

```
##   temp         m
## 1   50 144.83333
## 2   65 107.58333
## 3   80  64.16667
```

```
battery.mean.mt <- ddply(battery.long, .(material,temp), summarise, m = mean(maxvolt))
battery.mean.mt
```

```
##   material temp      m
## 1        1   50 134.75
## 2        1   65  57.25
## 3        1   80  57.50
## 4        2   50 155.75
## 5        2   65 119.75
## 6        2   80  49.50
## 7        3   50 144.00
## 8        3   65 145.75
## 9        3   80  85.50
```

```
# Interaction plots, ggplot
p <- ggplot(battery.long, aes(x = material, y = maxvolt, colour = temp, shape = temp))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = battery.mean.mt, aes(y = m), size = 4)
p <- p + geom_line(data = battery.mean.mt, aes(y = m, group = temp), size = 1.5)
p <- p + labs(title = "Battery interaction plot, temp by material")
print(p)
```



```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05interbat1.jpg")
```

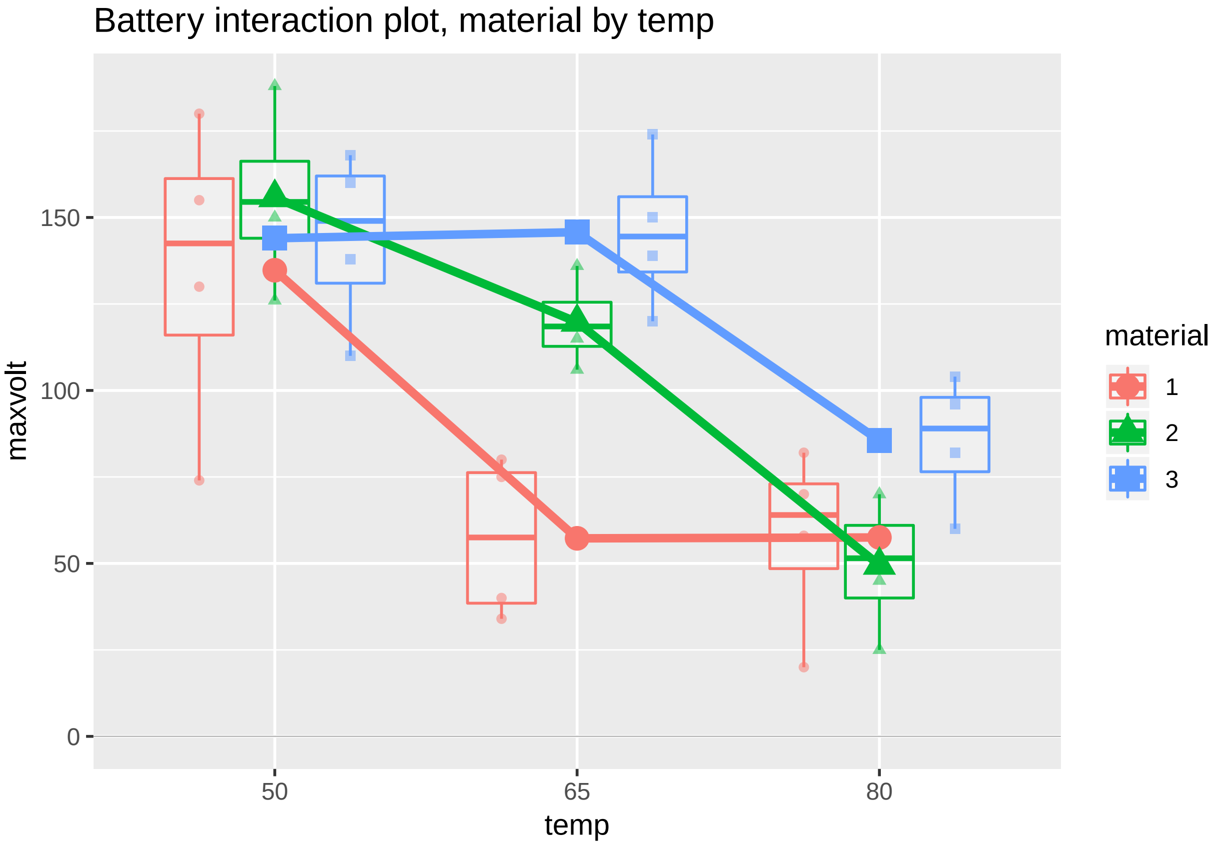
```
## jpeg
```

```
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
p <- ggplot(battery.long, aes(x = temp, y = maxvolt, colour = material, shape = material))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                    , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = battery.mean.mt, aes(y = m), size = 4)
p <- p + geom_line(data = battery.mean.mt, aes(y = m, group = material), size = 1.5)
p <- p + labs(title = "Battery interaction plot, material by temp")
print(p)
```



```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05interbat2.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
#### Multiple comparisons

### comparing means (interaction is significant, compare the different combinations of temp*materials
```

```r
par(mfrow=c(1,1))
  library(multcomp)
  battery.long$mt <- with(battery, interaction(material, temp))
  lm.mt <- lm(maxvolt ~ mt, data = battery.long)
  Anova(lm.mt, type=3)
```

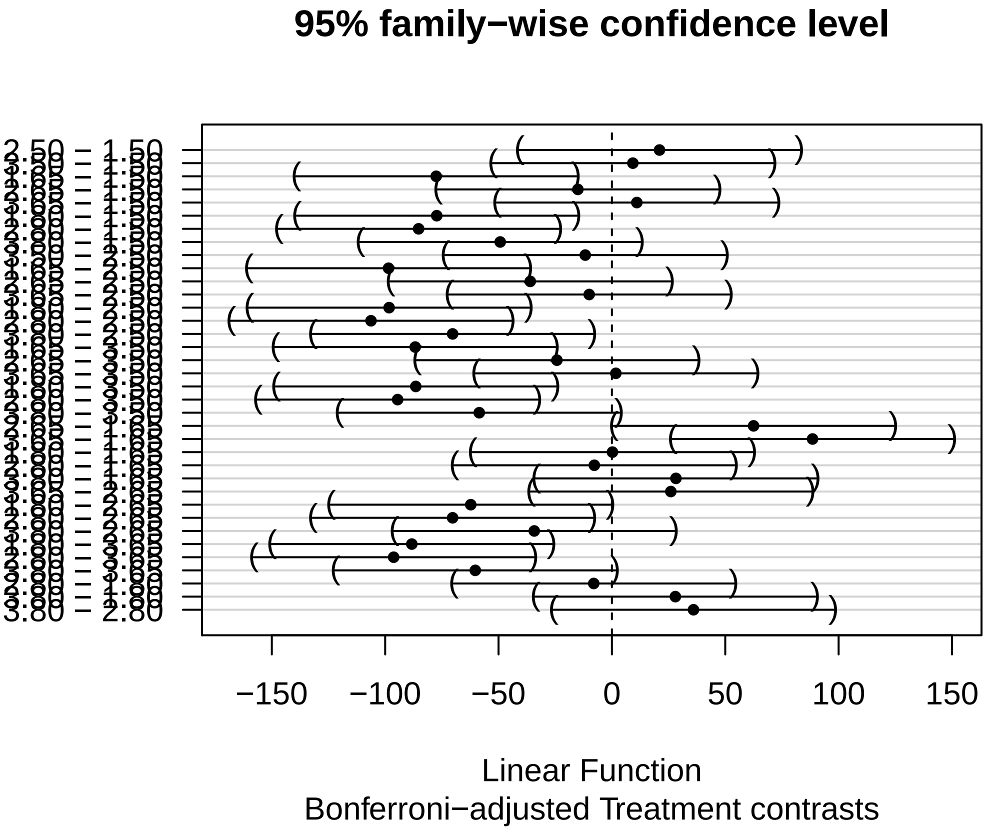
```
## Anova Table (Type III tests)
##
## Response: maxvolt
##             Sum Sq Df F value    Pr(>F)
## (Intercept)  72630  1 107.566 6.456e-11 ***
## mt           59416  8  10.999 9.426e-07 ***
## Residuals    18231 27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
  glht.battery <- glht(aov(lm.mt), linfct = mcp(mt = "Tukey"))
  summary(glht.battery, test = adjusted("bonferroni"))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: aov(formula = lm.mt)
##
## Linear Hypotheses:
##                   Estimate Std. Error t value Pr(>|t|)
## 2.50 - 1.50 == 0    21.00      18.37   1.143 1.000000
## 3.50 - 1.50 == 0     9.25      18.37   0.503 1.000000
## 1.65 - 1.50 == 0   -77.50      18.37  -4.218 0.008932 **
## 2.65 - 1.50 == 0   -15.00      18.37  -0.816 1.000000
## 3.65 - 1.50 == 0    11.00      18.37   0.599 1.000000
## 1.80 - 1.50 == 0   -77.25      18.37  -4.204 0.009262 **
## 2.80 - 1.50 == 0   -85.25      18.37  -4.640 0.002887 **
## 3.80 - 1.50 == 0   -49.25      18.37  -2.680 0.445639
## 3.50 - 2.50 == 0   -11.75      18.37  -0.639 1.000000
## 1.65 - 2.50 == 0   -98.50      18.37  -5.361 0.000416 ***
## 2.65 - 2.50 == 0   -36.00      18.37  -1.959 1.000000
## 3.65 - 2.50 == 0   -10.00      18.37  -0.544 1.000000
## 1.80 - 2.50 == 0   -98.25      18.37  -5.347 0.000432 ***
## 2.80 - 2.50 == 0  -106.25      18.37  -5.783 0.000135 ***
## 3.80 - 2.50 == 0   -70.25      18.37  -3.823 0.025379 *
## 1.65 - 3.50 == 0   -86.75      18.37  -4.721 0.002319 **
## 2.65 - 3.50 == 0   -24.25      18.37  -1.320 1.000000
## 3.65 - 3.50 == 0     1.75      18.37   0.095 1.000000
## 1.80 - 3.50 == 0   -86.50      18.37  -4.708 0.002405 **
## 2.80 - 3.50 == 0   -94.50      18.37  -5.143 0.000746 ***
## 3.80 - 3.50 == 0   -58.50      18.37  -3.184 0.131186
## 2.65 - 1.65 == 0    62.50      18.37   3.402 0.075676 .
## 3.65 - 1.65 == 0    88.50      18.37   4.817 0.001795 **
## 1.80 - 1.65 == 0     0.25      18.37   0.014 1.000000
## 2.80 - 1.65 == 0    -7.75      18.37  -0.422 1.000000
```

```
## 3.80 - 1.65 == 0    28.25      18.37   1.537 1.000000
## 3.65 - 2.65 == 0    26.00      18.37   1.415 1.000000
## 1.80 - 2.65 == 0   -62.25      18.37  -3.388 0.078348 .
## 2.80 - 2.65 == 0   -70.25      18.37  -3.823 0.025379 *
## 3.80 - 2.65 == 0   -34.25      18.37  -1.864 1.000000
## 1.80 - 3.65 == 0   -88.25      18.37  -4.803 0.001862 **
## 2.80 - 3.65 == 0   -96.25      18.37  -5.238 0.000578 ***
## 3.80 - 3.65 == 0   -60.25      18.37  -3.279 0.103263
## 2.80 - 1.80 == 0    -8.00      18.37  -0.435 1.000000
## 3.80 - 1.80 == 0    28.00      18.37   1.524 1.000000
## 3.80 - 2.80 == 0    36.00      18.37   1.959 1.000000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)
```

```r
# plot the summary
op <- par(no.readonly = TRUE) # the whole list of settable par's.
# make wider left margin to fit contrast labels
par(mar = c(5, 10, 4, 2) + 0.1)  # order is c(bottom, left, top, right)
# plot bonferroni-corrected difference intervals
plot(summary(glht.battery, test = adjusted("bonferroni"))
     , sub="Bonferroni-adjusted Treatment contrasts")
```



```r
par(op) # reset plotting options
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/ch05multicompbat.jpg")
```

```
## jpeg
##    3
```

```
  dev.off()
```

```
## pdf
##   2
```

```
###unbalanced desgin#########
#### Example: Rat insulin
rat <- read.table("http://statacumen.com/teach/ADA2/ADA2_notes_Ch05_ratinsulin.dat"
                 , header = TRUE)
# make time a factor variable and label the levels
rat$time<- factor(rat$time)
str(rat)
```

```
## 'data.frame':    48 obs. of  3 variables:
##  $ vein   : Factor w/ 2 levels "j","p": 1 1 1 1 1 1 1 1 1 1 ...
##  $ time   : Factor w/ 3 levels "0","30","60": 1 1 1 1 1 2 2 2 2 2 ...
##  $ insulin: int  18 36 12 24 43 61 116 63 132 68 ...
```

```
head(rat, 3)
```

```
##   vein time insulin
## 1    j    0      18
## 2    j    0      36
## 3    j    0      12
```

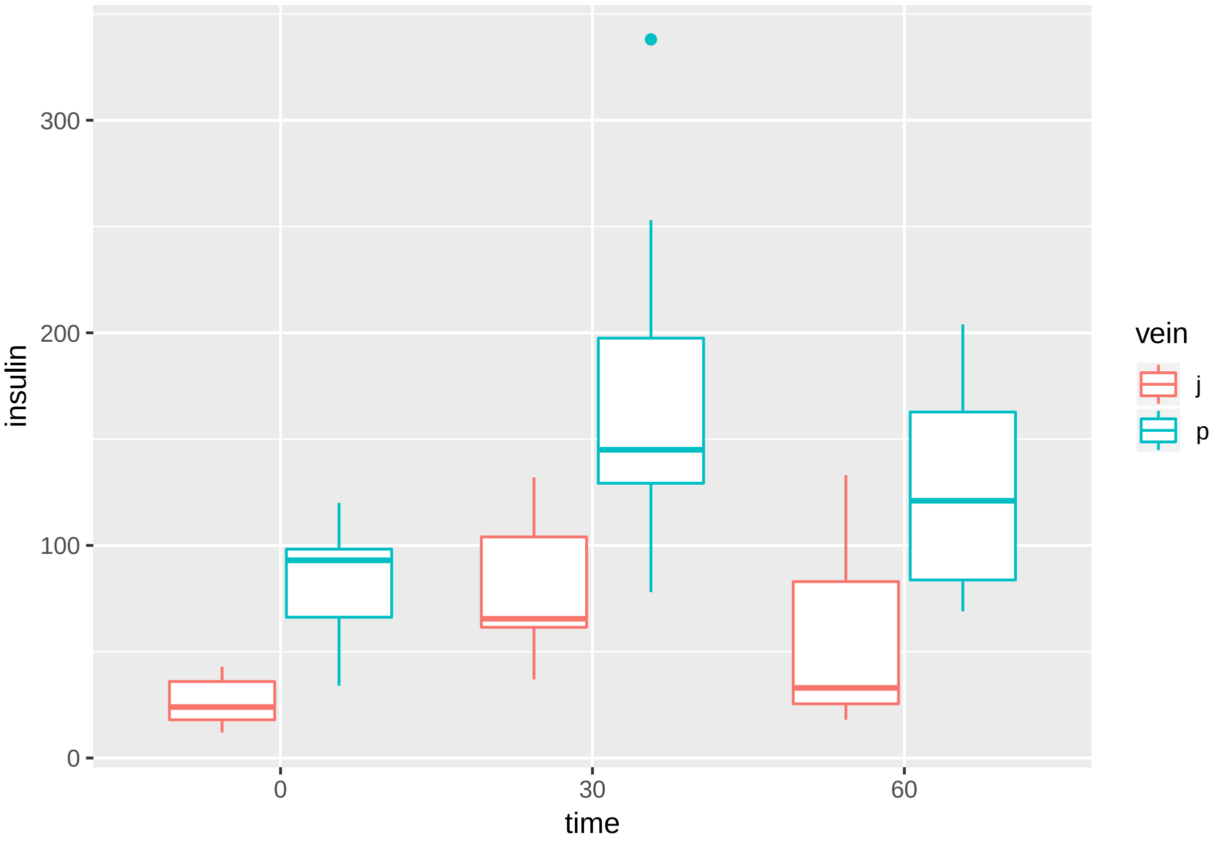
```
tail(rat, 3)
```

```
##    vein time insulin
## 46    p   60     105
## 47    p   60      71
## 48    p   60      83
```

```
table(rat$vein,rat$time)
```

```
##
##      0 30 60
##   j  5  6  3
##   p 12 10 12
```

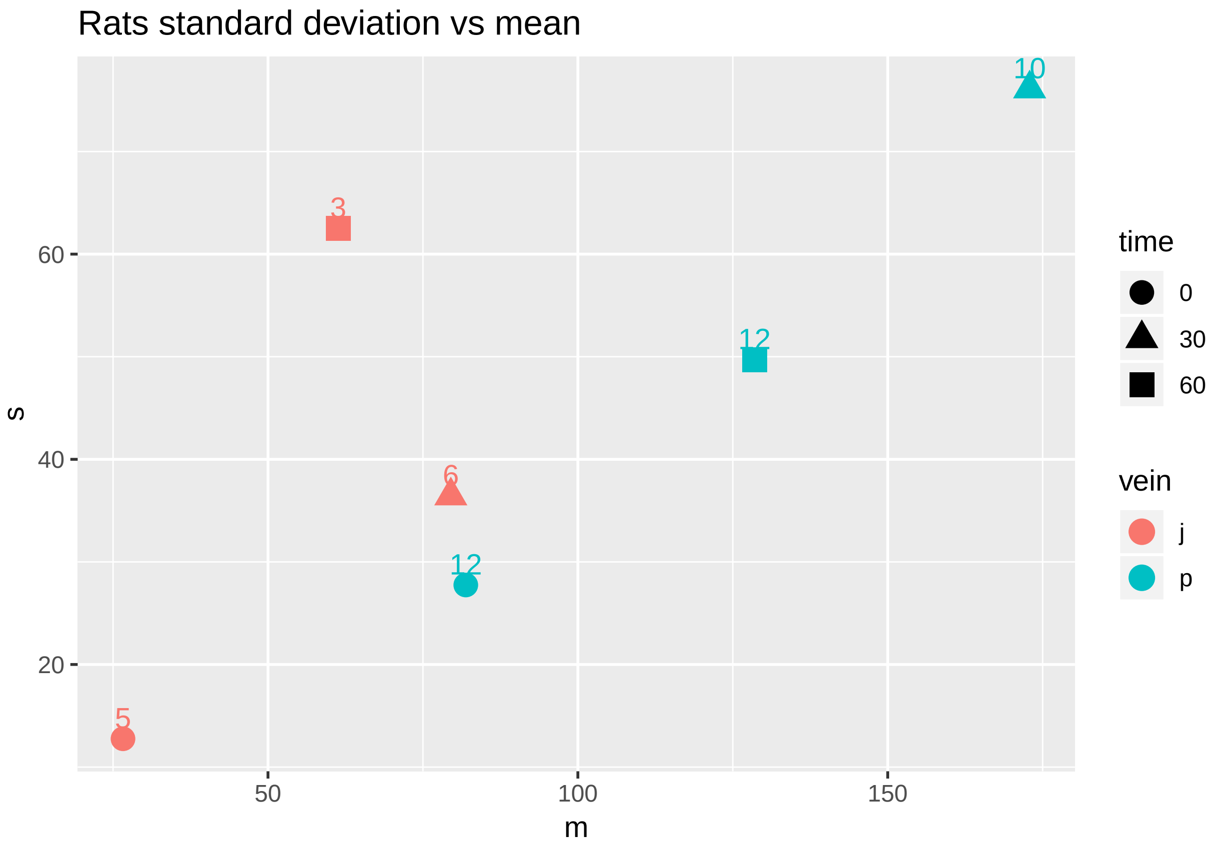
```
# boxplots, ggplot
p <- ggplot(rat, aes(x = time, y = insulin, colour = vein))
p <- p + geom_boxplot()
print(p)
```



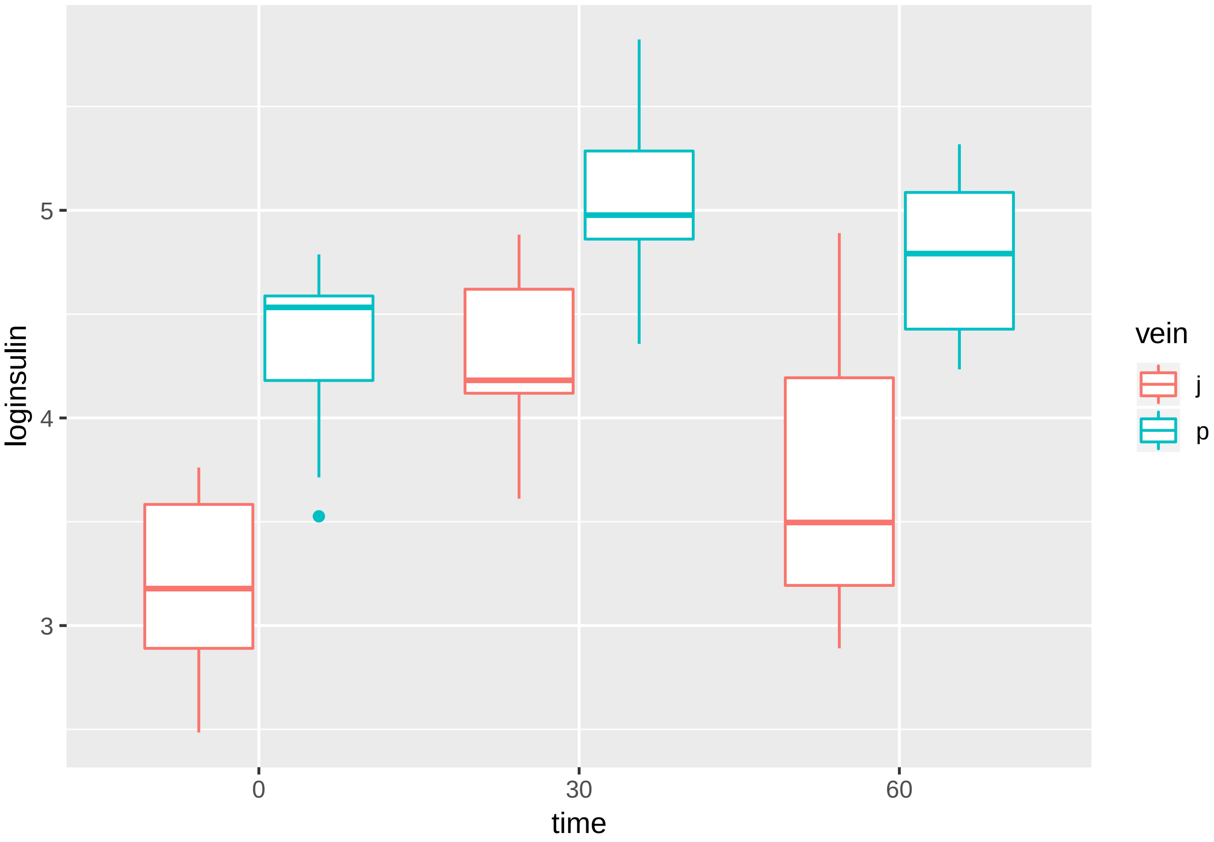
```
# mean vs sd plot
library(plyr)
# means and standard deviations for each time/interaction cell
rat.meansd.tv <- ddply(rat, .(time,vein), summarise
                      , m = mean(insulin), s = sd(insulin), n = length(insulin))
rat.meansd.tv
```

```
##   time vein         m        s  n
## 1    0    j  26.60000 12.75931  5
## 2    0    p  81.91667 27.74710 12
## 3   30    j  79.50000 36.44585  6
## 4   30    p 172.90000 76.11753 10
## 5   60    j  61.33333 62.51666  3
## 6   60    p 128.50000 49.71830 12
```

```
p <- ggplot(rat.meansd.tv, aes(x = m, y = s, shape = time, colour = vein, label=n))
p <- p + geom_point(size=4)
# labels are sample sizes
p <- p + geom_text(hjust = 0.5, vjust = -0.5)
p <- p + labs(title = "Rats standard deviation vs mean")
print(p)
```



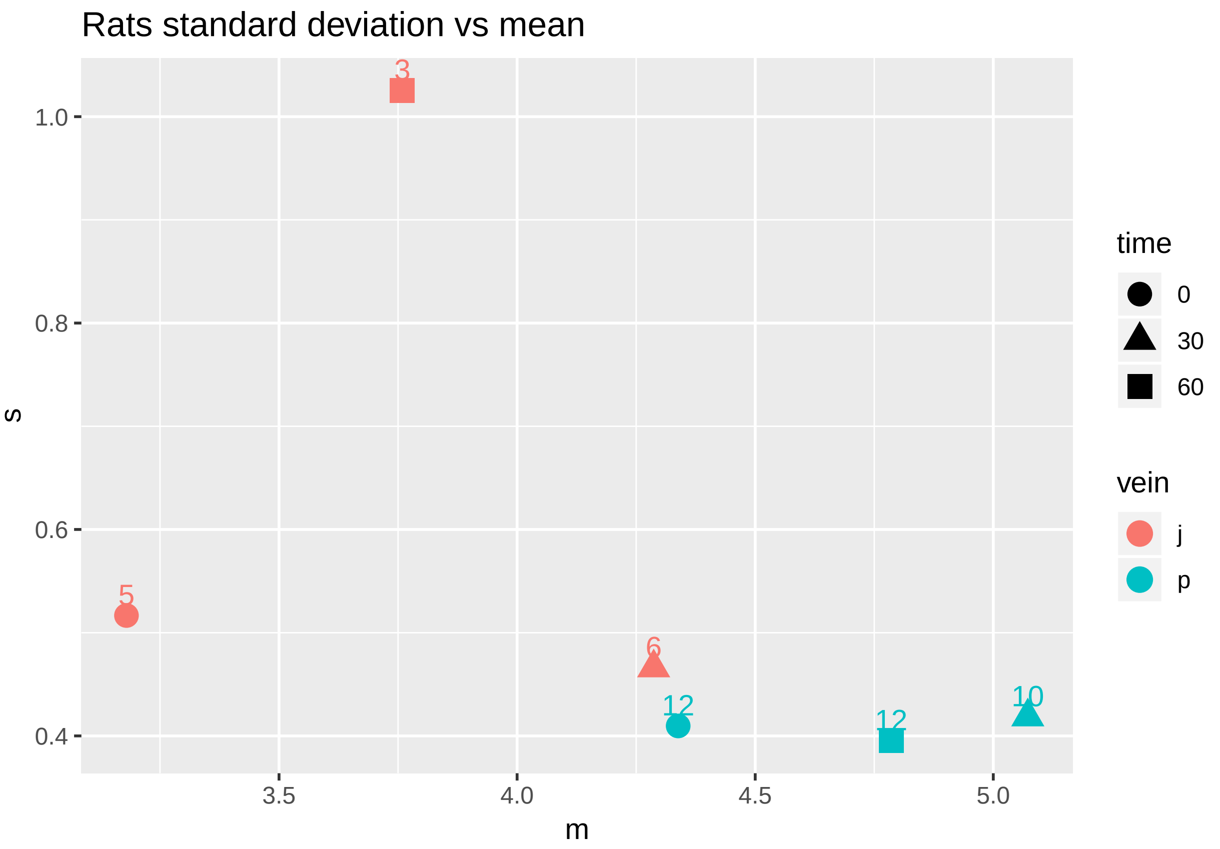
```
rat$loginsulin <- log(rat$insulin)



# boxplots, ggplot
p <- ggplot(rat, aes(x = time, y = loginsulin, colour = vein))
p <- p + geom_boxplot()
print(p)
```



```r
# mean vs sd plot
library(plyr)
# means and standard deviations for each time/interaction cell
rat.meansd.tv <- ddply(rat, .(time,vein), summarise
                        , m = mean(loginsulin)
                        , s = sd(loginsulin)
                        , n = length(loginsulin))
rat.meansd.tv
```

```
##   time vein        m         s  n
## 1    0    j 3.179610 0.5166390  5
## 2    0    p 4.338230 0.4096427 12
## 3   30    j 4.286804 0.4660571  6
## 4   30    p 5.072433 0.4185221 10
## 5   60    j 3.759076 1.0255165  3
## 6   60    p 4.785463 0.3953252 12
```

```r
p <- ggplot(rat.meansd.tv, aes(x = m, y = s, shape = time, colour = vein, label=n))
p <- p + geom_point(size=4)
# labels are sample sizes
p <- p + geom_text(hjust = 0.5, vjust = -0.5)
p <- p + labs(title = "Rats standard deviation vs mean")
print(p)
```



```
lm.i.t.v.tv <- lm(loginsulin ~ time*vein, data = rat
                , contrasts = list(time = contr.sum, vein = contr.sum))
## CRITICAL!!!  Unbalanced design warning.
##   The contrast statement above must be included identifying
##   each main effect with "contr.sum" in order for the correct
##   Type III SS to be computed.
##   See http://goanna.cs.rmit.edu.au/~fscholer/anova.php
library(car)
# type I   SS  (intercept SS not shown)
summary(aov(lm.i.t.v.tv))
```

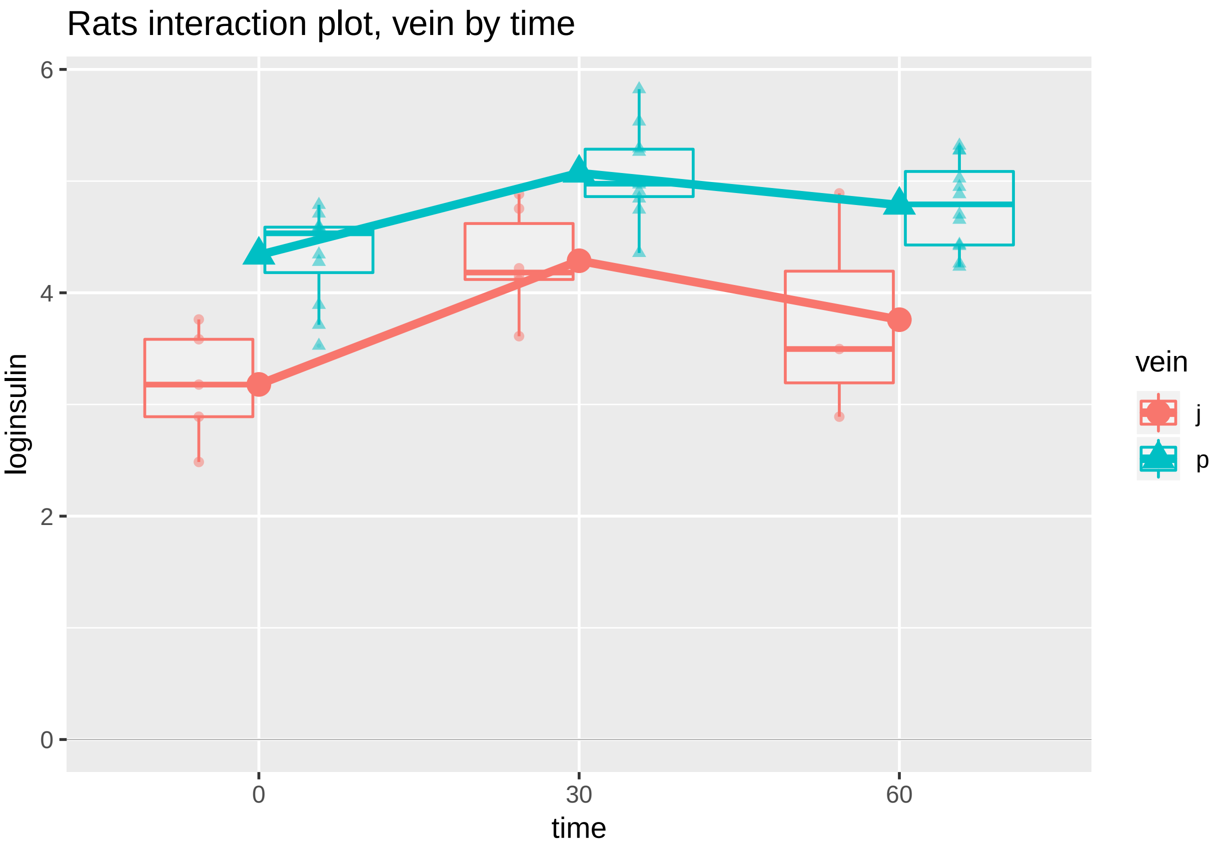
```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## time         2  5.450   2.725   12.18 6.74e-05 ***
## vein         1  9.321   9.321   41.66 8.82e-08 ***
## time:vein    2  0.259   0.130    0.58    0.565
## Residuals   42  9.399   0.224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# type III SS
Anova(lm.i.t.v.tv, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: loginsulin
##             Sum Sq Df   F value    Pr(>F)
## (Intercept) 668.54  1 2987.5842 < 2.2e-16 ***
## time          6.18  2   13.7996 2.475e-05 ***
```

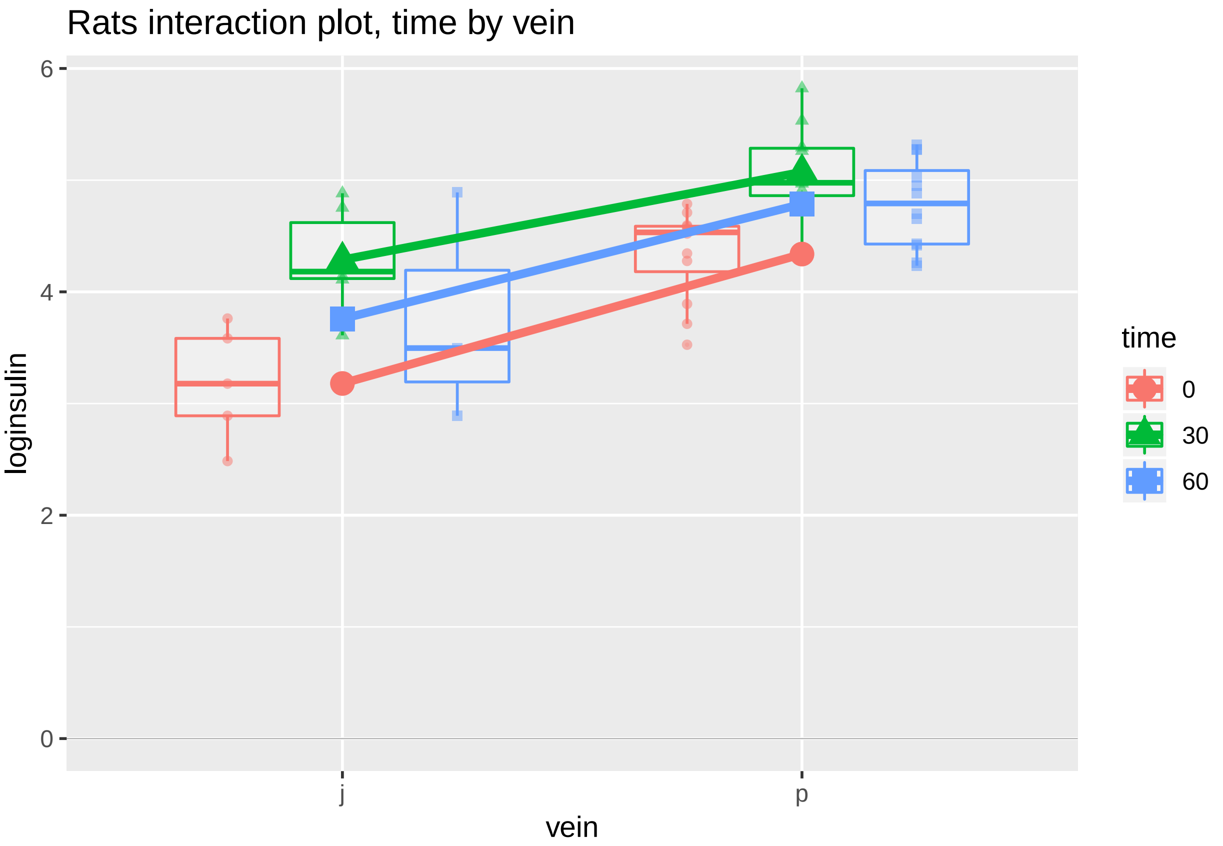
```
## vein          9.13  1   40.7955 1.101e-07 ***
## time:vein     0.26  2    0.5797    0.5645
## Residuals     9.40 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# calculate means for plot
library(plyr)
rat.mean.tv <- ddply(rat, .(time,vein), summarise, m = mean(loginsulin))

# Interaction plots, ggplot
p <- ggplot(rat, aes(x = time, y = loginsulin, colour = vein, shape = vein))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                  , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = rat.mean.tv, aes(y = m), size = 4)
p <- p + geom_line(data = rat.mean.tv, aes(y = m, group = vein), size = 1.5)
p <- p + labs(title = "Rats interaction plot, vein by time")
print(p)
```



```r
p <- ggplot(rat, aes(x = vein, y = loginsulin, colour = time, shape = time))
p <- p + geom_hline(aes(yintercept = 0), colour = "black"
                  , linetype = "solid", size = 0.2, alpha = 0.3)
p <- p + geom_boxplot(alpha = 0.25, outlier.size=0.1)
p <- p + geom_point(alpha = 0.5, position=position_dodge(width=0.75))
p <- p + geom_point(data = rat.mean.tv, aes(y = m), size = 4)
p <- p + geom_line(data = rat.mean.tv, aes(y = m, group = time), size = 1.5)
```

```
p <- p + labs(title = "Rats interaction plot, time by vein")
print(p)
```



```
#### lsmeans
library(plyr)
# unbalanced, don't match
rat.mean.t  <- ddply(rat, .(time), summarise, m = mean(loginsulin))
rat.mean.t
```

```
##   time        m
## 1    0 3.997460
## 2   30 4.777822
## 3   60 4.580186
```

```
library(lsmeans)
```

```
## The 'lsmeans' package is being deprecated.
## Users are encouraged to switch to 'emmeans'.
## See help('transition') for more information, including how
## to convert 'lsmeans' objects and scripts to work with 'emmeans'.
```

```
lsmeans(lm.i.t.v.tv, list(pairwise ~ time), adjust = "bonferroni")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## $`lsmeans of time`
##  time   lsmean        SE df lower.CL upper.CL
##  0    3.758920 0.1258994 42 3.504845 4.012996
##  30   4.679619 0.1221403 42 4.433130 4.926108
```

```
##  60   4.272270 0.1526754 42 3.964158 4.580381
##
## Results are averaged over the levels of: vein
## Confidence level used: 0.95
##
## $`pairwise differences of contrast`
##  contrast   estimate        SE df t.ratio p.value
##  0 - 30   -0.9206985 0.1754107 42  -5.249  <.0001
##  0 - 60   -0.5133494 0.1978900 42  -2.594  0.0390
##  30 - 60   0.4073491 0.1955199 42   2.083  0.1300
##
## Results are averaged over the levels of: vein
## P value adjustment: bonferroni method for 3 tests
```

```r
# unbalanced, don't match
rat.mean.v  <- ddply(rat, .(vein), summarise, m = mean(loginsulin))
rat.mean.v
```

```
##   vein        m
## 1    j 3.778293
## 2    p 4.712019
```

```r
  # compare jugular mean above (3.778) with the lsmeans average below (3.742)
(3.179610 + 4.286804 + 3.759076)/3
```

```
## [1] 3.74183
```

```r
lsmeans(lm.i.t.v.tv, list(pairwise ~ vein), adjust = "bonferroni")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## $`lsmeans of vein`
##  vein   lsmean         SE df lower.CL upper.CL
##  j    3.741830 0.13192664 42 3.475592 4.008069
##  p    4.732042 0.08142689 42 4.567716 4.896369
##
## Results are averaged over the levels of: time
## Confidence level used: 0.95
##
## $`pairwise differences of contrast`
##  contrast   estimate        SE df t.ratio p.value
##  j - p    -0.9902121 0.1550322 42  -6.387  <.0001
##
## Results are averaged over the levels of: time
```

```r
# unbalanced, but highest-order interaction cell means will match
rat.mean.tv <- ddply(rat, .(time,vein), summarise, m = mean(loginsulin))
rat.mean.tv
```

```
##   time vein        m
## 1    0    j 3.179610
## 2    0    p 4.338230
## 3   30    j 4.286804
## 4   30    p 5.072433
## 5   60    j 3.759076
## 6   60    p 4.785463
```

```r
lsmeans(lm.i.t.v.tv, list(pairwise ~ time | vein), adjust = "bonferroni")
```

```
## $`lsmeans of time | vein`
## vein = j:
##  time   lsmean        SE df lower.CL upper.CL
##  0    3.179610 0.2115533 42 2.752678 3.606542
##  30   4.286804 0.1931208 42 3.897071 4.676538
##  60   3.759076 0.2731141 42 3.207910 4.310243
##
## vein = p:
##  time   lsmean        SE df lower.CL upper.CL
##  0    4.338230 0.1365570 42 4.062647 4.613814
##  30   5.072433 0.1495908 42 4.770547 5.374320
##  60   4.785463 0.1365570 42 4.509880 5.061047
##
## Confidence level used: 0.95
##
## $`pairwise differences of contrast, vein | vein`
## vein = j:
##  contrast   estimate        SE df t.ratio p.value
##  0 - 30   -1.1071941 0.2864445 42  -3.865  0.0011
##  0 - 60   -0.5794659 0.3454650 42  -1.677  0.3027
##  30 - 60   0.5277282 0.3344951 42   1.578  0.3664
##
## vein = p:
##  contrast   estimate        SE df t.ratio p.value
##  0 - 30   -0.7342029 0.2025468 42  -3.625  0.0023
##  0 - 60   -0.4472330 0.1931208 42  -2.316  0.0766
##  30 - 60   0.2869699 0.2025468 42   1.417  0.4917
##
## P value adjustment: bonferroni method for 3 tests
```

```r
lsmeans(lm.i.t.v.tv, list(pairwise ~ vein | time), adjust = "bonferroni")
```

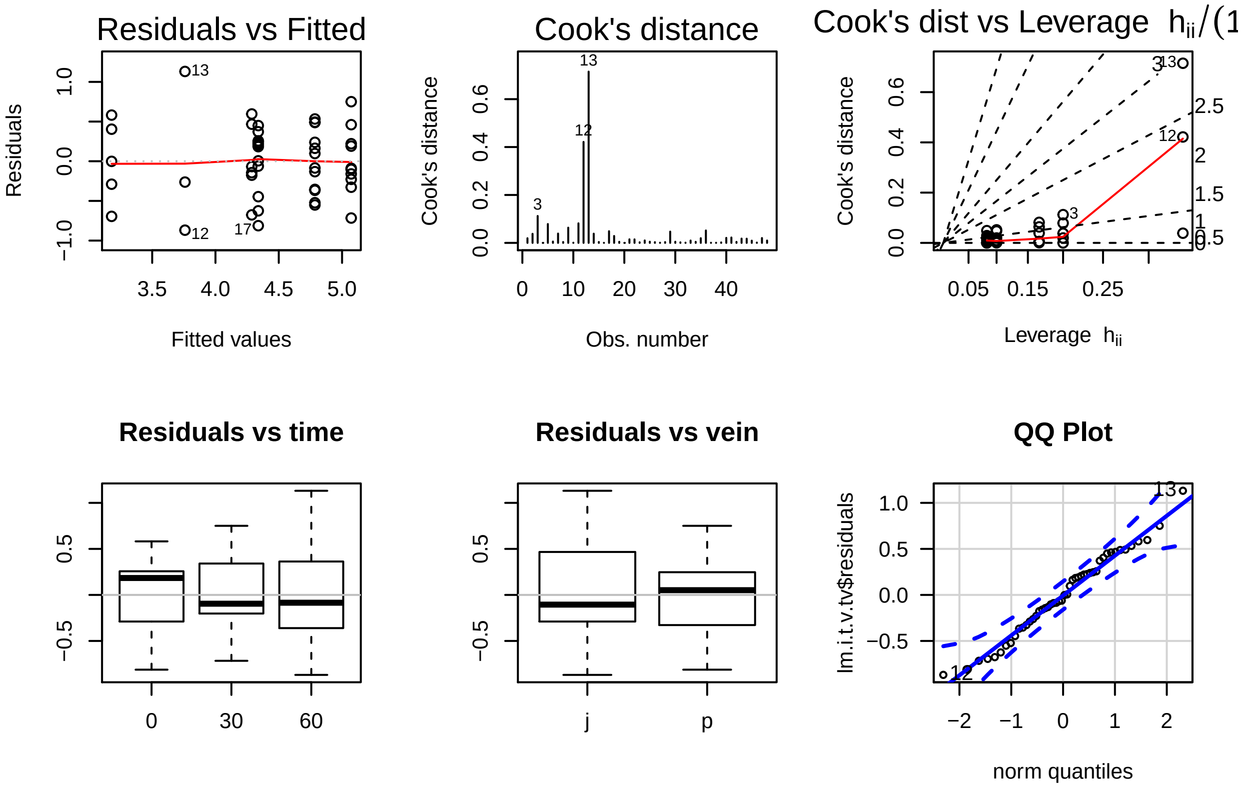
```
## $`lsmeans of vein | time`
## time = 0:
##  vein   lsmean        SE df lower.CL upper.CL
##  j    3.179610 0.2115533 42 2.752678 3.606542
##  p    4.338230 0.1365570 42 4.062647 4.613814
##
## time = 30:
##  vein   lsmean        SE df lower.CL upper.CL
##  j    4.286804 0.1931208 42 3.897071 4.676538
##  p    5.072433 0.1495908 42 4.770547 5.374320
##
## time = 60:
##  vein   lsmean        SE df lower.CL upper.CL
##  j    3.759076 0.2731141 42 3.207910 4.310243
##  p    4.785463 0.1365570 42 4.509880 5.061047
##
## Confidence level used: 0.95
##
## $`pairwise differences of contrast, time | time`
## time = 0:
```

```
##  contrast   estimate        SE df t.ratio p.value
##  j - p    -1.1586202 0.2517988 42  -4.601  <.0001
##
## time = 30:
##  contrast   estimate        SE df t.ratio p.value
##  j - p    -0.7856289 0.2442807 42  -3.216  0.0025
##
## time = 60:
##  contrast   estimate        SE df t.ratio p.value
##  j - p    -1.0263873 0.3053508 42  -3.361  0.0017
```

```r
# interaction model
lm.i.t.v.tv <- lm(loginsulin ~ time*vein, data = rat
                , contrasts = list(time = contr.sum, vein = contr.sum))
Anova(lm.i.t.v.tv, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: loginsulin
##              Sum Sq Df   F value    Pr(>F)
## (Intercept) 668.54  1 2987.5842 < 2.2e-16 ***
## time          6.18  2   13.7996 2.475e-05 ***
## vein          9.13  1   40.7955 1.101e-07 ***
## time:vein     0.26  2    0.5797    0.5645
## Residuals     9.40 42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# plot diagnistics
par(mfrow=c(2,3))
plot(lm.i.t.v.tv, which = c(1,4,6))

plot(rat$time, lm.i.t.v.tv$residuals, main="Residuals vs time")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

plot(rat$vein, lm.i.t.v.tv$residuals, main="Residuals vs vein")
  # horizontal line at zero
  abline(h = 0, col = "gray75")

# Normality of Residuals
library(car)
qqPlot(lm.i.t.v.tv$residuals, las = 1, main="QQ Plot")
```



```
## [1] 13 12
```

```
### comparing lsmeans (unbalanced)
library(lsmeans)
## compare levels of one factor at each level of another factor separately
# comapre different time levels
lsmeans(lm.i.t.v.tv, list(pairwise ~ time), adjust = "bonferroni")
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## $`lsmeans of time`
##  time   lsmean        SE df lower.CL upper.CL
##  0    3.758920 0.1258994 42 3.504845 4.012996
##  30   4.679619 0.1221403 42 4.433130 4.926108
##  60   4.272270 0.1526754 42 3.964158 4.580381
##
## Results are averaged over the levels of: vein
## Confidence level used: 0.95
##
## $`pairwise differences of contrast`
##  contrast   estimate        SE df t.ratio p.value
##  0 - 30   -0.9206985 0.1754107 42  -5.249  <.0001
##  0 - 60   -0.5133494 0.1978900 42  -2.594  0.0390
##  30 - 60   0.4073491 0.1955199 42   2.083  0.1300
##
## Results are averaged over the levels of: vein
## P value adjustment: bonferroni method for 3 tests
```

```
# comapre different vein levels
lsmeans(lm.i.t.v.tv, list(pairwise ~ vein), adjust = "bonferroni")
```

```
## NOTE: Results may be misleading due to involvement in interactions

## $`lsmeans of vein`
##  vein   lsmean         SE df lower.CL upper.CL
##  j    3.741830 0.13192664 42 3.475592 4.008069
##  p    4.732042 0.08142689 42 4.567716 4.896369
##
## Results are averaged over the levels of: time
## Confidence level used: 0.95
##
## $`pairwise differences of contrast`
##  contrast   estimate        SE df t.ratio p.value
##  j - p    -0.9902121 0.1550322 42  -6.387  <.0001
##
## Results are averaged over the levels of: time
```