

# Univariate Review II

Mike Strube

September 5, 2018

## 1 Overview

*The simple ANOVA design with one outcome can be extended to include multiple outcomes (repeated measures). This can include the same outcome collected at different times, or, it can include the same outcome collected under different experimental conditions. These are treated equivalently from a statistical standpoint ("the numbers don't know where they came from") but the two types of designs can produce numbers that may not satisfy assumptions equally well.*

## 2 Preliminaries

*In this section, the RStudio workspace and console panes are cleared of old output, variables, and other miscellaneous debris. Packages are loaded and any required data files are retrieved.*

```
options(replace.assign = TRUE, width = 65, digits = 4, scipen = 4, fig.width = 4,
        fig.height = 4)
# Clear the workspace and console.
rm(list = ls(all = TRUE))
cat("\f")
```

```
# Turn off showing of significance asterisks.
options(show.signif.stars = F)
# Set the contrast option; important for ANOVAs.
options(contrasts = c("contr.sum", "contr.poly"))
how_long <- Sys.time()
set.seed(123)
library(knitr)
```

```
library(arm)

## Loading required package: MASS
## Loading required package: Matrix
## Loading required package: lme4
##
## arm (Version 1.10-1, built: 2018-4-12)
## Working directory is C:/Courses/Psychology 516/PowerPoint/2018
library(psych)
```

```

##
## Attaching package: 'psych'
## The following objects are masked from 'package:arm':
##
##   logit, rescale, sim

library(lme4)
library(lmtest)

## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric

library(car)

## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:psych':
##
##   logit
## The following object is masked from 'package:arm':
##
##   logit

library(emmeans)

## Warning: package 'emmeans' was built under R version 3.5.1
## NOTE: As of emmeans versions > 1.2.3,
##       The 'cld' function will be deprecated in favor of 'CLD'.
##       You may use 'cld' only if you have package:multcomp attached.

library(multcomp)

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

library(ggplot2)

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

```

```

library(ez)
library(biotools)

## Loading required package: rpanel
## Loading required package: tcltk
## Package 'rpanel', version 1.1-4: type help(rpanel) for summary information
## Loading required package: tkrplot
## Loading required package: lattice
## Loading required package: SpatialEpi
## Loading required package: sp

## ---
## biotools version 3.1

##

library(MVN)

## sROC 0.1-2 loaded

```

## 2.1 Data Files

*Data in two different formats (wide and long) will be used to illustrate different approaches to analysis of variance with repeated measures. Data in wide format have all information for each case in a single row; multiple columns contain the repeated measures. Data in long format have each repeated measure in a separate row. A single column contains all of the repeated measures values. A categorical variable is used to indicate the particular time period or repeated measure. Dummy variables can be useful in some analyses to produce no-intercept models that estimate means for further comparison.*

```

# Get the data from the working directory.
setwd("C:\\Courses\\Psychology 516\\PowerPoint\\2018")
Data_Wide <- read.table("Data_2.csv", sep = ",", header = TRUE)
Data_Wide <- as.data.frame(Data_Wide)
head(Data_Wide, n = 12)

##      Subject Group DV1 DV2 DV3 G1 G2 G3
## 1         1      1   5   6   5  1  0  0
## 2         2      1   4   5   7  1  0  0
## 3         3      1   5   3   4  1  0  0
## 4         4      1   4   5   6  1  0  0
## 5         5      1   3   6   5  1  0  0
## 6         6      1   5   4   4  1  0  0
## 7         7      1   5   4   7  1  0  0
## 8         8      1   6   5   5  1  0  0
## 9         9      1   7   7   8  1  0  0
## 10        10      1   6   4   7  1  0  0
## 11        11      1   6   5   5  1  0  0
## 12        12      1   6   3   5  1  0  0

Data_Long <- read.table("Set_2.csv", sep = ",", header = TRUE)
Data_Long <- as.data.frame(Data_Long)

# Create dummy variables to indicate the particular time period of

```

```

# measurement.
Data_Long$T1 <- ifelse(Data_Long$Time == 1, 1, 0)
Data_Long$T2 <- ifelse(Data_Long$Time == 2, 1, 0)
Data_Long$T3 <- ifelse(Data_Long$Time == 3, 1, 0)
head(Data_Long, n = 12)

##      Subject Group G1 G2 G3 Time DV T1 T2 T3
## 1         1      1  1  0  0     1  5  1  0  0
## 2         1      1  1  0  0     2  6  0  1  0
## 3         1      1  1  0  0     3  5  0  0  1
## 4         2      1  1  0  0     1  4  1  0  0
## 5         2      1  1  0  0     2  5  0  1  0
## 6         2      1  1  0  0     3  7  0  0  1
## 7         3      1  1  0  0     1  5  1  0  0
## 8         3      1  1  0  0     2  3  0  1  0
## 9         3      1  1  0  0     3  4  0  0  1
## 10        4      1  1  0  0     1  4  1  0  0
## 11        4      1  1  0  0     2  5  0  1  0
## 12        4      1  1  0  0     3  6  0  0  1

# Create factor versions of variables.
Data_Wide$Group_F <- factor(Data_Wide$Group, levels = c(1, 2, 3),
  labels = c("Group 1", "Group 2", "Group 3"))
Data_Long$Group_F <- factor(Data_Long$Group, levels = c(1, 2, 3),
  labels = c("Group 1", "Group 2", "Group 3"))
Data_Long$Time_F <- factor(Data_Long$Time, levels = c(1, 2, 3), labels = c("Time 1",
  "Time 2", "Time 3"))
Data_Long$Subject_F <- factor(Data_Long$Subject)

```

### 3 Traditional ANOVA Model

*We will treat the data as coming from a 3 x 3 (Group x Time) design. There are a variety of ways to get the ANOVA model with repeated measures in R. The traditional approach using the `aov()` function requires the data in long format. Note that Group, Time, and Subject must be specified as factors if they are not designated as such in the data frame. In a repeated measures design, the repeated measure is considered to be nested within the subject factor (each case has its own profile of scores). The `Error()` designation is needed to define the error terms correctly.*

*Note that the standard ANOVA model assumes the factors are fixed. This becomes an important distinction when repeated measure designs are tested in a hierarchical linear model framework. In the HLM framework, nested variables can be specified as random or fixed. The `aov()` function also produces Type I sums of squares in which variance accounted for by the effects is based on order of entry. Other approaches (Type II and Type III sums of squares) can be obtained with the `ezANOVA()` function from the `ez` package. When the design is balanced (as is true here), all of these approach produce identical results.*

```
ANOVA_1 <- aov(Data_Long$DV ~ Time_F + Group_F + Time_F:Group_F +  
  Error(Subject_F/Time_F), data = Data_Long)  
summary(ANOVA_1)  
  
##  
## Error: Subject_F  
##           Df Sum Sq Mean Sq F value    Pr(>F)  
## Group_F    2   49.4    24.69    13.2 0.000013  
## Residuals 72  134.5     1.87  
##  
## Error: Subject_F:Time_F  
##           Df Sum Sq Mean Sq F value    Pr(>F)  
## Time_F      2   66.7    33.4    42.7 2.7e-15  
## Time_F:Group_F  4   71.5    17.9    22.9 1.2e-14  
## Residuals    144  112.5     0.8
```

#### 3.1 Means, Standard Errors, and Confidence Intervals

*The function, `emmeans()`, from the `emmeans` package provides marginal and cell means, standard errors, and confidence limits. It requires a reference grid that defines the nature of the design. That information is contained in the fit object from the ANOVA.*

```
model_rg <- ref_grid(ANOVA_1)  
  
Group_emm <- emmeans(model_rg, "Group_F")  
  
## NOTE: Results may be misleading due to involvement in interactions  
  
Time_emm <- emmeans(model_rg, "Time_F")  
  
## NOTE: Results may be misleading due to involvement in interactions  
  
Group_x_Time_emm <- emmeans(model_rg, c("Group_F", "Time_F"))  
  
Group_emm
```

```
## Group_F emmean      SE df lower.CL upper.CL
## Group 1  5.027 0.1578 72    4.712    5.341
## Group 2  6.173 0.1578 72    5.859    6.488
## Group 3  5.560 0.1578 72    5.245    5.875
##
## Results are averaged over the levels of: Time_F
## Confidence level used: 0.95

Time_emm

## Time_F emmean      SE      df lower.CL upper.CL
## Time 1  4.933 0.1235 179.9    4.690    5.177
## Time 2  6.267 0.1235 179.9    6.023    6.510
## Time 3  5.560 0.1235 179.9    5.316    5.804
##
## Results are averaged over the levels of: Group_F
## Confidence level used: 0.95

Group_x_Time_emm

## Group_F Time_F emmean      SE      df lower.CL upper.CL
## Group 1 Time 1  5.04 0.2139 179.9    4.618    5.462
## Group 2 Time 1  4.88 0.2139 179.9    4.458    5.302
## Group 3 Time 1  4.88 0.2139 179.9    4.458    5.302
## Group 1 Time 2  4.88 0.2139 179.9    4.458    5.302
## Group 2 Time 2  6.96 0.2139 179.9    6.538    7.382
## Group 3 Time 2  6.96 0.2139 179.9    6.538    7.382
## Group 1 Time 3  5.16 0.2139 179.9    4.738    5.582
## Group 2 Time 3  6.68 0.2139 179.9    6.258    7.102
## Group 3 Time 3  4.84 0.2139 179.9    4.418    5.262
##
## Confidence level used: 0.95
```

### 3.2 Bar Graphs with 95% Confidence Intervals

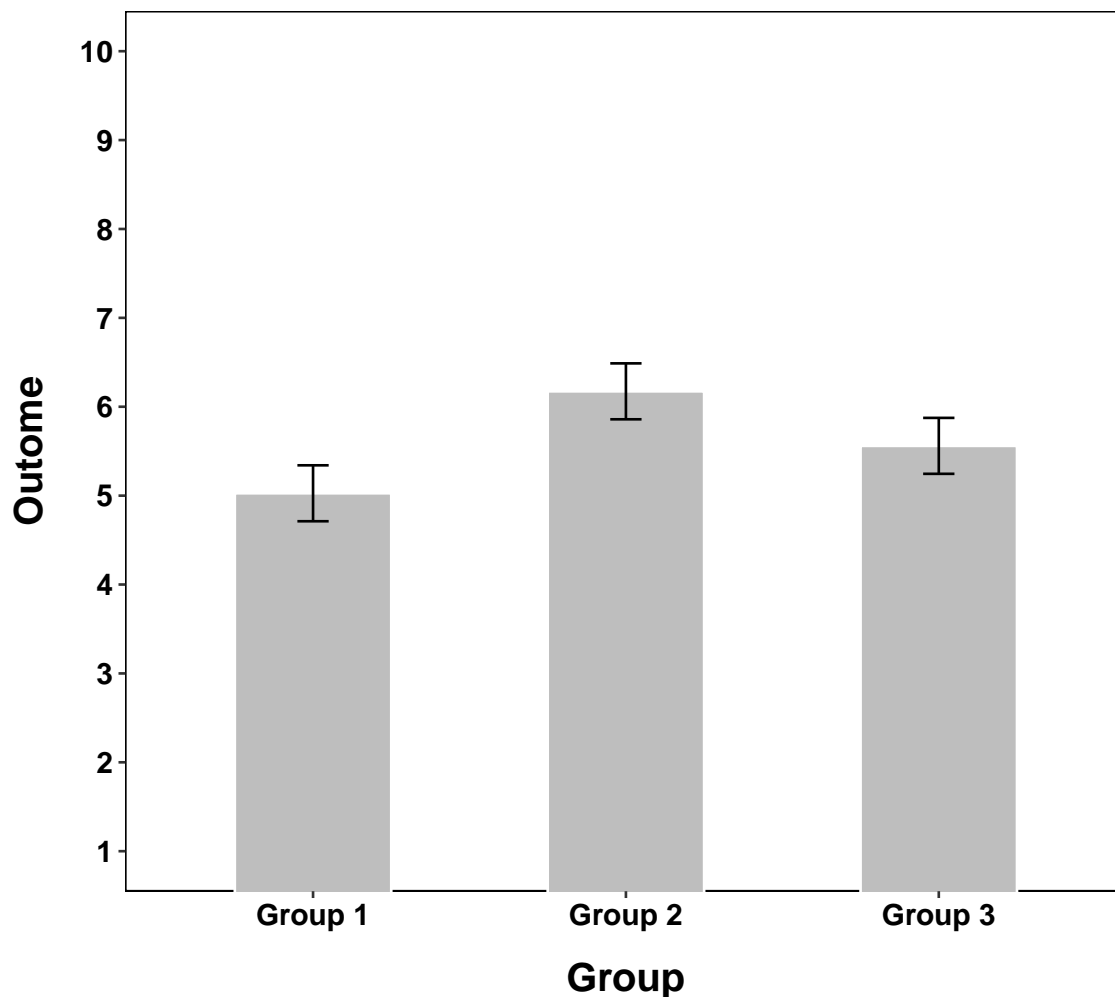
*The following figures provide a better illustration of the ANOVA effects. They provide the condition means and 95% confidence intervals.*

```
plot_data <- cbind(summary(Group_emm)[1], summary(Group_emm)[2], summary(Group_emm)[3],
  summary(Group_emm)[5], summary(Group_emm)[6])
plot_data <- as.data.frame(plot_data)

ggplot(plot_data, aes(x = Group_F, y = emmean)) + geom_bar(position = position_dodge(),
  stat = "identity", color = "white", width = 0.5, fill = "grey") +
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL), width = 0.1,
    position = position_dodge(0.5)) + coord_cartesian(xlim = c(1,
  3), ylim = c(1, 10)) + scale_y_continuous(breaks = seq(1, 10,
  1)) + xlab("Group") + ylab("Outcome") + theme(text = element_text(size = 14,
  family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
  size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
  size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
  0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
  15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
  plot.title = element_text(size = 16, face = "bold", margin = margin(0,
```

```
0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
  linetype = 1, color = "black"), panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
legend.title = element_blank()) + ggtitle("Group Differences Collapsed Across Time \n(95% Confidence
```

## Group Differences Collapsed Across Time (95% Confidence Intervals)



```
plot_data <- cbind(summary(Time_emm)[1], summary(Time_emm)[2], summary(Time_emm)[3],
  summary(Time_emm)[5], summary(Time_emm)[6])
plot_data <- as.data.frame(plot_data)

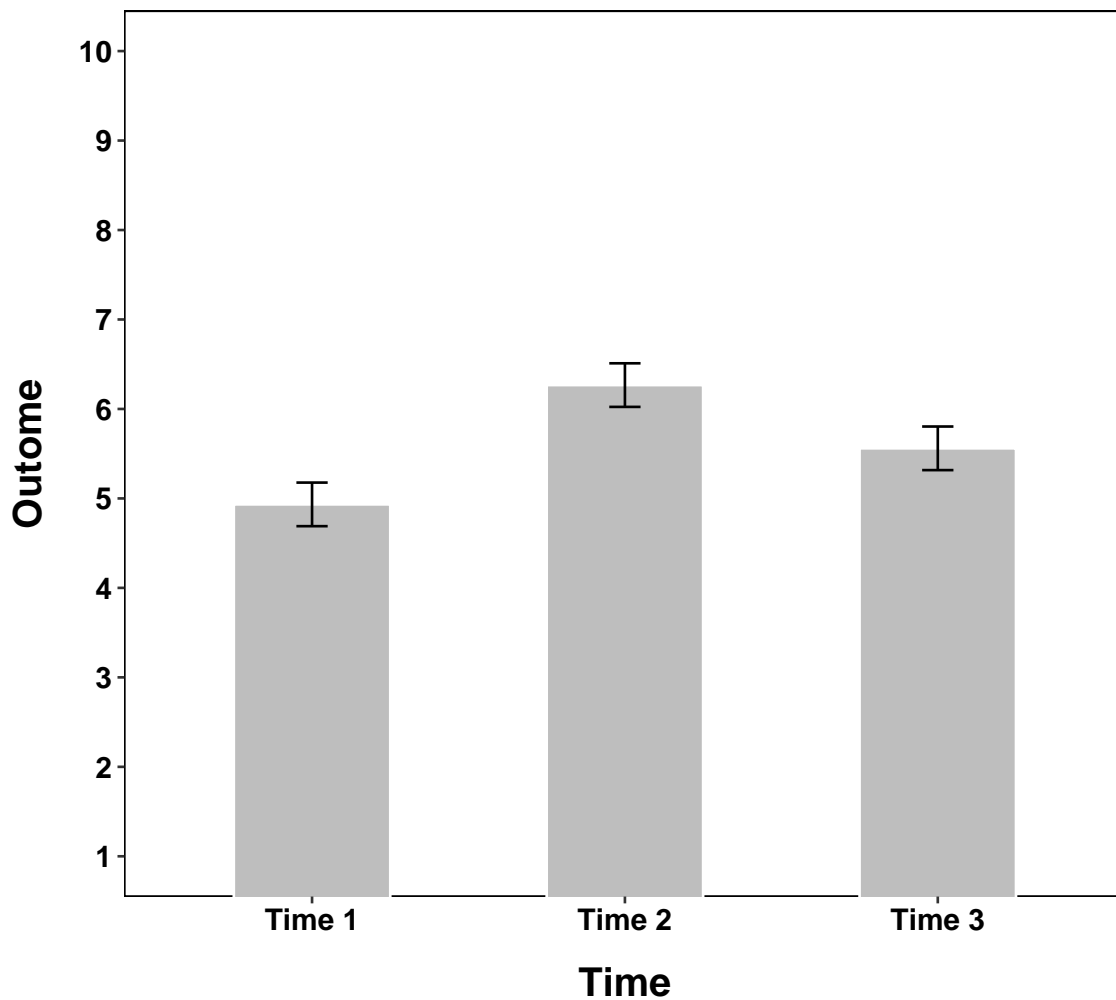
ggplot(plot_data, aes(x = Time_F, y = emmean)) + geom_bar(position = position_dodge(),
  stat = "identity", color = "white", width = 0.5, fill = "grey") +
  geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL), width = 0.1,
```

```

    position = position_dodge(0.5)) + coord_cartesian(xlim = c(1,
3), ylim = c(1, 10)) + scale_y_continuous(breaks = seq(1, 10,
1)) + xlab("Time") + ylab("Outcome") + theme(text = element_text(size = 14,
family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
plot.title = element_text(size = 16, face = "bold", margin = margin(0,
0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
linetype = 1, color = "black"), panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
legend.title = element_blank()) + ggtitle("Time Differences Collapsed Across Group \n(95% Confidence

```

### Time Differences Collapsed Across Group (95% Confidence Intervals)





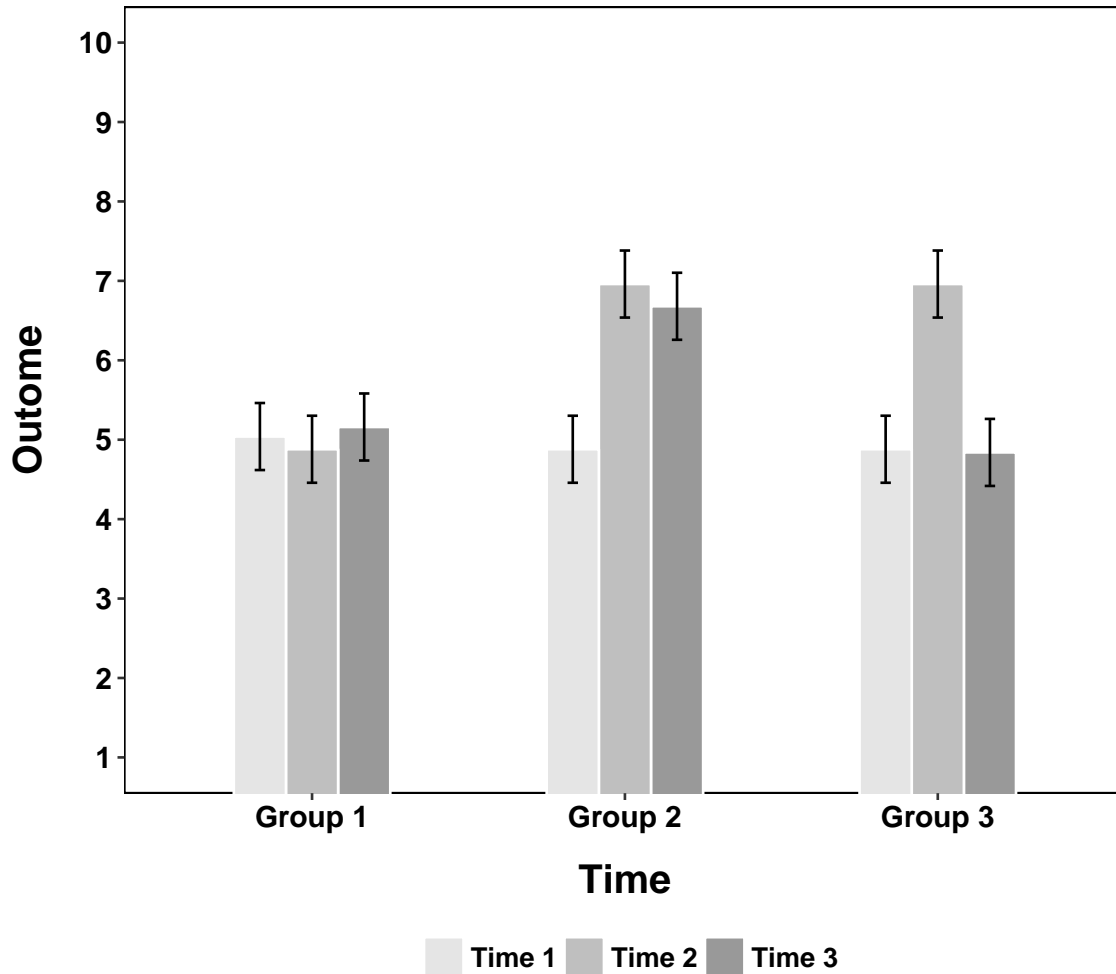
```

plot_data <- cbind(summary(Group_x_Time_emm)[1], summary(Group_x_Time_emm)[2],
  summary(Group_x_Time_emm)[3], summary(Group_x_Time_emm)[6], summary(Group_x_Time_emm)[7])
plot_data <- as.data.frame(plot_data)

ggplot(plot_data, aes(x = Group_F, y = emmean, fill = Time_F)) + geom_bar(position = position_dodge(),
  stat = "identity", color = "white", width = 0.5) + scale_fill_manual(values = c("gray90",
  "gray75", "gray60")) + geom_errorbar(aes(ymin = lower.CL, ymax = upper.CL),
  width = 0.1, position = position_dodge(0.5)) + coord_cartesian(xlim = c(1,
  3), ylim = c(1, 10)) + scale_y_continuous(breaks = seq(1, 10,
  1)) + xlab("Time") + ylab("Outcome") + theme(text = element_text(size = 14,
  family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
  size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
  size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
  0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
  15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
  plot.title = element_text(size = 16, face = "bold", margin = margin(0,
  0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
  linetype = 1, color = "black"), panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
  plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
  legend.title = element_blank()) + ggtitle("Outcome as a Function of Group and Time \n(95% Confidence

```

### Outcome as a Function of Group and Time (95% Confidence Intervals)



### 3.3 Comparisons Among Means

*The CLD( ) function can be used to obtain pairwise comparisons. The "adjust" option allows specifying the particular form of Type I error control (Holm procedure is used here). The pairwise comparisons can be made at any level in the design. For interactions, the pair-wise comparisons are conducted using a simple main effects approach. That means that pairs of means for one of the variables involved in the interaction are compared within each level of the other variable involved in the interaction (or each combination of levels if the interaction is three-way).*

*The output that has "group" as the right-most column provides a convenient summary display. Rows that do not share any numbers in the group column represent conditions that are significantly different.*

```
CLD(Group_emm, alpha = 0.05, adjust = "holm", details = TRUE)
```

```
## $emmeans
##   Group_F emmean      SE df lower.CL upper.CL .group
##   Group 1  5.027 0.1578 72    4.640    5.414    1
##   Group 3  5.560 0.1578 72    5.173    5.947    2
##   Group 2  6.173 0.1578 72    5.786    6.560    3
##
## Results are averaged over the levels of: Time_F
## Confidence level used: 0.95
## Conf-level adjustment: bonferroni method for 3 estimates
## P value adjustment: holm method for 3 tests
## significance level used: alpha = 0.05
##
## $comparisons
##   contrast      estimate      SE df t.ratio p.value
##   Group 3 - Group 1    0.5333 0.2232 72    2.390  0.0195
##   Group 2 - Group 1    1.1467 0.2232 72    5.137 <.0001
##   Group 2 - Group 3    0.6133 0.2232 72    2.748  0.0151
##
## Results are averaged over the levels of: Time_F
## P value adjustment: holm method for 3 tests
```

```
CLD(Time_emm, alpha = 0.05, adjust = "holm", details = TRUE)
```

```
## $emmeans
##   Time_F emmean      SE   df lower.CL upper.CL .group
##   Time 1  4.933 0.1235 179.9    4.635    5.232    1
##   Time 3  5.560 0.1235 179.9    5.262    5.858    2
##   Time 2  6.267 0.1235 179.9    5.968    6.565    3
##
## Results are averaged over the levels of: Group_F
## Confidence level used: 0.95
## Conf-level adjustment: bonferroni method for 3 estimates
## P value adjustment: holm method for 3 tests
## significance level used: alpha = 0.05
##
## $comparisons
##   contrast      estimate      SE   df t.ratio p.value
##   Time 3 - Time 1    0.6267 0.1443 144    4.343 <.0001
##   Time 2 - Time 1    1.3333 0.1443 144    9.240 <.0001
##   Time 2 - Time 3    0.7067 0.1443 144    4.897 <.0001
##
## Results are averaged over the levels of: Group_F
## P value adjustment: holm method for 3 tests
```

```
CLD(Group_x_Time_emm, by = "Group_F", alpha = 0.05, adjust = "holm",
     details = TRUE)
```

```
## $emmeans
## Group_F = Group 1:
##   Time_F emmean      SE   df lower.CL upper.CL .group
```

```

## Time 2 4.88 0.2139 179.9 4.28 5.48 1
## Time 1 5.04 0.2139 179.9 4.44 5.64 1
## Time 3 5.16 0.2139 179.9 4.56 5.76 1
##
## Group_F = Group 2:
## Time_F emmean SE df lower.CL upper.CL .group
## Time 1 4.88 0.2139 179.9 4.28 5.48 1
## Time 3 6.68 0.2139 179.9 6.08 7.28 2
## Time 2 6.96 0.2139 179.9 6.36 7.56 2
##
## Group_F = Group 3:
## Time_F emmean SE df lower.CL upper.CL .group
## Time 3 4.84 0.2139 179.9 4.24 5.44 1
## Time 1 4.88 0.2139 179.9 4.28 5.48 1
## Time 2 6.96 0.2139 179.9 6.36 7.56 2
##
## Confidence level used: 0.95
## Conf-level adjustment: bonferroni method for 9 estimates
## P value adjustment: holm method for 3 tests
## significance level used: alpha = 0.05
##
## $comparisons
## Group_F = Group 1:
## contrast estimate SE df t.ratio p.value
## Time 1 - Time 2 0.16 0.2499 144 0.640 1.0000
## Time 3 - Time 2 0.28 0.2499 144 1.120 0.7934
## Time 3 - Time 1 0.12 0.2499 144 0.480 1.0000
##
## Group_F = Group 2:
## contrast estimate SE df t.ratio p.value
## Time 1 - Time 2 1.80 0.2499 144 7.201 <.0001
## Time 3 - Time 2 2.08 0.2499 144 8.322 <.0001
## Time 3 - Time 1 0.28 0.2499 144 1.120 0.2645
##
## Group_F = Group 3:
## contrast estimate SE df t.ratio p.value
## Time 1 - Time 2 0.04 0.2499 144 0.160 0.8731
## Time 3 - Time 2 2.12 0.2499 144 8.482 <.0001
## Time 3 - Time 1 2.08 0.2499 144 8.322 <.0001
##
## P value adjustment: holm method for 3 tests

```

```

CLD(Group_x_Time_emm, by = "Time_F", alpha = 0.05, adjust = "holm",
     details = TRUE)

```

```

## $emmeans
## Time_F = Time 1:
## Group_F emmean SE df lower.CL upper.CL .group
## Group 2 4.88 0.2139 179.9 4.28 5.48 1
## Group 3 4.88 0.2139 179.9 4.28 5.48 1
## Group 1 5.04 0.2139 179.9 4.44 5.64 1
##
## Time_F = Time 2:

```

```

##   Group_F emmean      SE    df lower.CL upper.CL .group
##   Group 1   4.88 0.2139 179.9    4.28    5.48    1
##   Group 2   6.96 0.2139 179.9    6.36    7.56    2
##   Group 3   6.96 0.2139 179.9    6.36    7.56    2
##
## Time_F = Time 3:
##   Group_F emmean      SE    df lower.CL upper.CL .group
##   Group 3   4.84 0.2139 179.9    4.24    5.44    1
##   Group 1   5.16 0.2139 179.9    4.56    5.76    1
##   Group 2   6.68 0.2139 179.9    6.08    7.28    2
##
## Confidence level used: 0.95
## Conf-level adjustment: bonferroni method for 9 estimates
## P value adjustment: holm method for 3 tests
## significance level used: alpha = 0.05
##
## $comparisons
## Time_F = Time 1:
##   contrast      estimate      SE    df t.ratio p.value
##   Group 3 - Group 2 -7.772e-16 0.3024 179.9    0.000  1.0000
##   Group 1 - Group 2  1.600e-01 0.3024 179.9    0.529  1.0000
##   Group 1 - Group 3  1.600e-01 0.3024 179.9    0.529  1.0000
##
## Time_F = Time 2:
##   contrast      estimate      SE    df t.ratio p.value
##   Group 3 - Group 2  2.080e+00 0.3024 179.9    6.878  <.0001
##   Group 1 - Group 2  2.080e+00 0.3024 179.9    6.878  <.0001
##   Group 1 - Group 3  2.720e-15 0.3024 179.9    0.000  1.0000
##
## Time_F = Time 3:
##   contrast      estimate      SE    df t.ratio p.value
##   Group 3 - Group 2  3.200e-01 0.3024 179.9    1.058  0.2914
##   Group 1 - Group 2  1.840e+00 0.3024 179.9    6.084  <.0001
##   Group 1 - Group 3  1.520e+00 0.3024 179.9    5.026  <.0001
##
## P value adjustment: holm method for 3 tests

```

### 3.4 Assumption Checks

*The analysis of variance makes several assumptions. The residuals are assumed to be normally distributed. The variance-covariance matrices for the repeated measures are assumed to be homogeneous across between-subjects conditions (Group in this case). The repeated measures are assumed to satisfy the sphericity assumption.*

#### 3.4.1 Normality

*The validity of the normality assumption is examined via the residuals from the model so that systematic variability does not contaminate the evaluation. Because there are multiple error terms (i.e., multiple linear combinations), there are multiple sets of residuals that need to be checked for normality.*

**3.4.1.1 Shapiro-Wilk Test** *The Shapiro-Wilk test can be applied to the distribution of residuals to determine if the normal distribution is a reasonable approximation. Mild*

*violations are not a concern. As sample size grows, even trivial violations can be statistically significant. Graphical methods (presented later) provide another evaluation tool.*

```
# Save the residuals from the model.
resid_data_b <- residuals(ANOVA_1$Subject_F)
resid_data_b <- as.data.frame(resid_data_b)
names(resid_data_b) <- c("R")

resid_data_w_Time <- residuals(ANOVA_1$`Subject_F:Time_F`)
resid_data_w_Time <- as.data.frame(resid_data_w_Time)
names(resid_data_w_Time) <- c("R")

# Shapiro-Wilk test.
shapiro.test(resid_data_b$R)

##
##  Shapiro-Wilk normality test
##
## data:  resid_data_b$R
## W = 0.99, p-value = 0.8

shapiro.test(resid_data_w_Time$R)

##
##  Shapiro-Wilk normality test
##
## data:  resid_data_w_Time$R
## W = 0.99, p-value = 0.1
```

*The normality tests suggest no violations of normality.*

**3.4.1.2 Residual Plots** *The residuals can be plotted for visual evaluation of normality. In the following QQ-plots, the data are plotted against the expected distribution if normality holds. Deviations from normality are indicated by data points that fall greater distances from the diagonal (normal fit) line.*

```
# The data to be plotted are standardized and saved.
resid_data_b$RS <- scale(resid_data_b$R)

# ggplot2 does produce an 'expected' line in its QQ plot. It can
# be added by finding the location of the interquartile range (1st
# and 3rd quartiles) for the data to be plotted and then finding
# the corresponding normal values. The X-Y coordinates can then be
# used to define the slope of the line and the intercept.

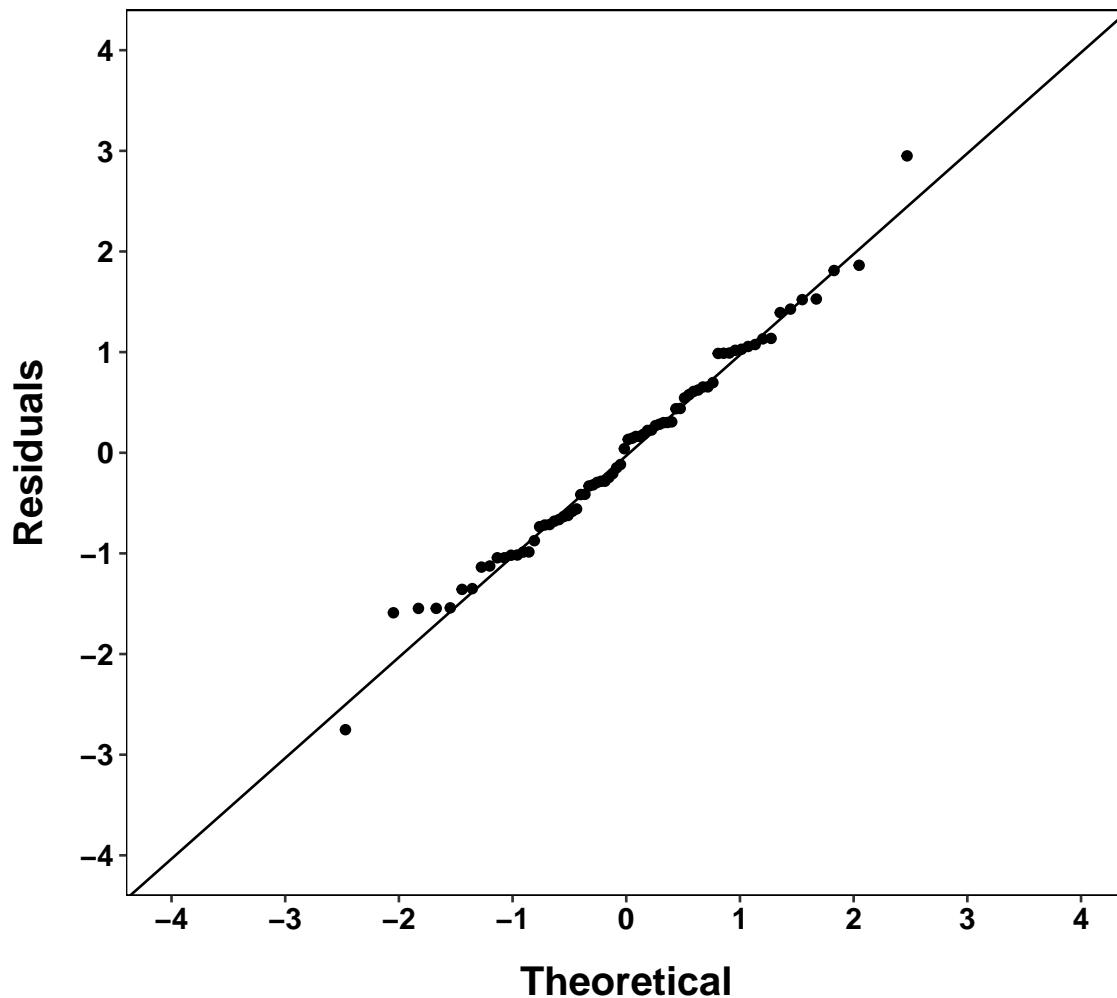
# The Y (obtained sample) values for the 1st and 3rd quartiles.
Y <- quantile(resid_data_b$RS, c(0.25, 0.75), na.rm = TRUE)
# The standard normal values corresponding to the Y values (i.e.,
# the X coordinates)
X <- qnorm(c(0.25, 0.75))
# The slope of the line defined by the pairs of X and Y values.
Slope <- diff(Y)/diff(X)
# And the intercept.
Intercept <- Y[1] - Slope * X[1]
```

```

ggplot(resid_data_b, aes(sample = RS)) + stat_qq(distribution = qnorm) +
  geom_abline(intercept = Intercept, slope = Slope) + scale_y_continuous(breaks = c(-4,
-3, -2, -1, 0, 1, 2, 3, 4)) + scale_x_continuous(breaks = c(-4,
-3, -2, -1, 0, 1, 2, 3, 4)) + coord_cartesian(xlim = c(-4, 4),
ylim = c(-4, 4)) + xlab("Theoretical") + ylab("Residuals") + theme(text = element_text(size = 14,
family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
plot.title = element_text(size = 16, face = "bold", margin = margin(0,
0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
linetype = 1, color = "black"), panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
legend.title = element_blank()) + ggtitle("Q-Q Plot for \n Between-Subject Residuals")

```

## Q-Q Plot for Between-Subject Residuals



```
# The data to be plotted are standardized and saved.
resid_data_w_Time$RS <- scale(resid_data_w_Time$R)

# ggplot2 does produce an 'expected' line in its QQ plot. It can
# be added by finding the location of the interquartile range (1st
# and 3rd quartiles) for the data to be plotted and then finding
# the corresponding normal values. The X-Y coordinates can then be
# used to define the slope of the line and the intercept.

# The Y (obtained sample) values for the 1st and 3rd quartiles.
Y <- quantile(resid_data_w_Time$RS, c(0.25, 0.75), na.rm = TRUE)
# The standard normal values corresponding to the Y values (i.e.,
# the X coordinates)
X <- qnorm(c(0.25, 0.75))
```



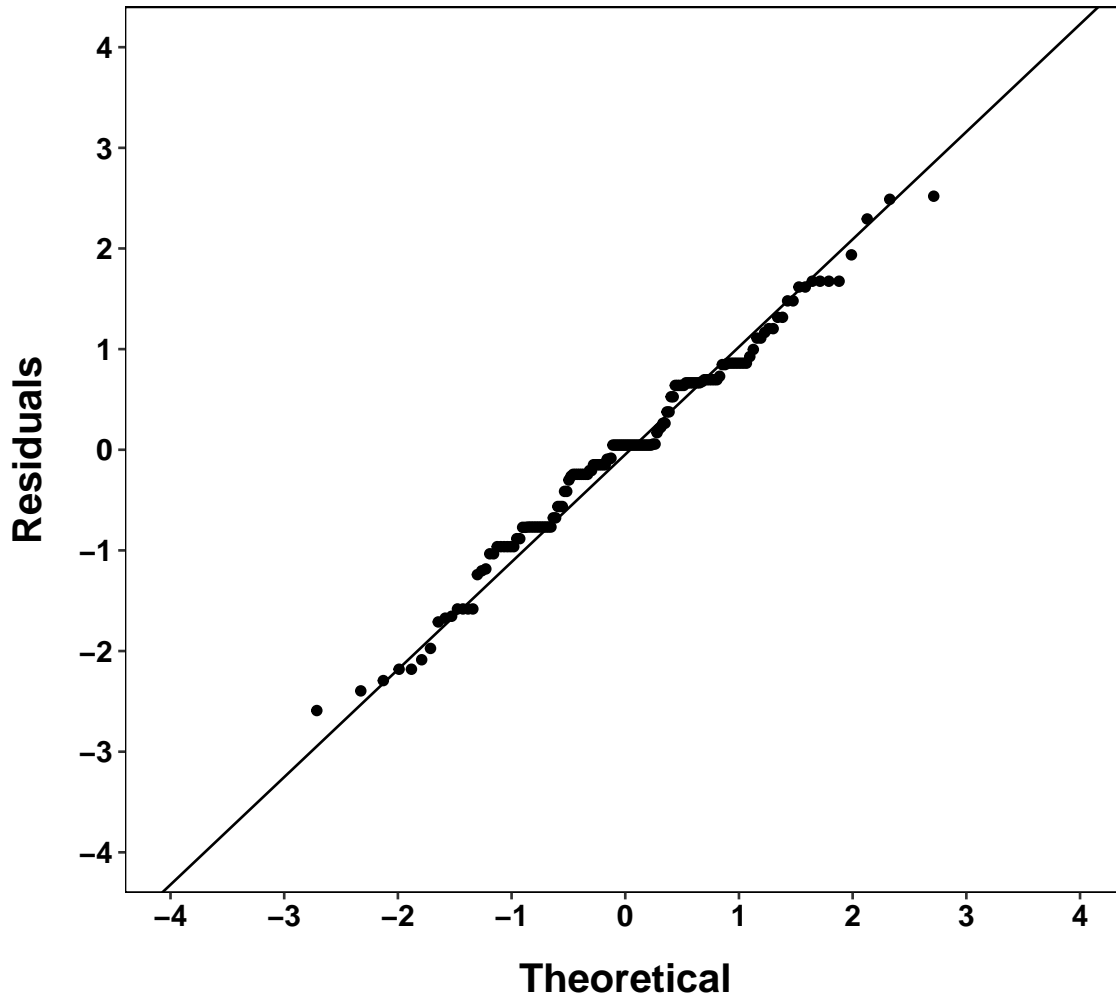
```

# The slope of the line defined by the pairs of X and Y values.
Slope <- diff(Y)/diff(X)
# And the intercept.
Intercept <- Y[1] - Slope * X[1]

ggplot(resid_data_w_Time, aes(sample = RS)) + stat_qq(distribution = qnorm) +
  geom_abline(intercept = Intercept, slope = Slope) + scale_y_continuous(breaks = c(-4,
-3, -2, -1, 0, 1, 2, 3, 4)) + scale_x_continuous(breaks = c(-4,
-3, -2, -1, 0, 1, 2, 3, 4)) + coord_cartesian(xlim = c(-4, 4),
ylim = c(-4, 4)) + xlab("Theoretical") + ylab("Residuals") + theme(text = element_text(size = 14,
family = "sans", color = "black", face = "bold"), axis.text.y = element_text(colour = "black",
size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
plot.title = element_text(size = 16, face = "bold", margin = margin(0,
0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
linetype = 1, color = "black"), panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
legend.title = element_blank()) + ggtitle("Q-Q Plot for \n Within-Subject Time Residuals")

```

## Q-Q Plot for Within-Subject Time Residuals



*As one additional check, the residuals are plotted in histograms, with the expected distribution if normality holds.*

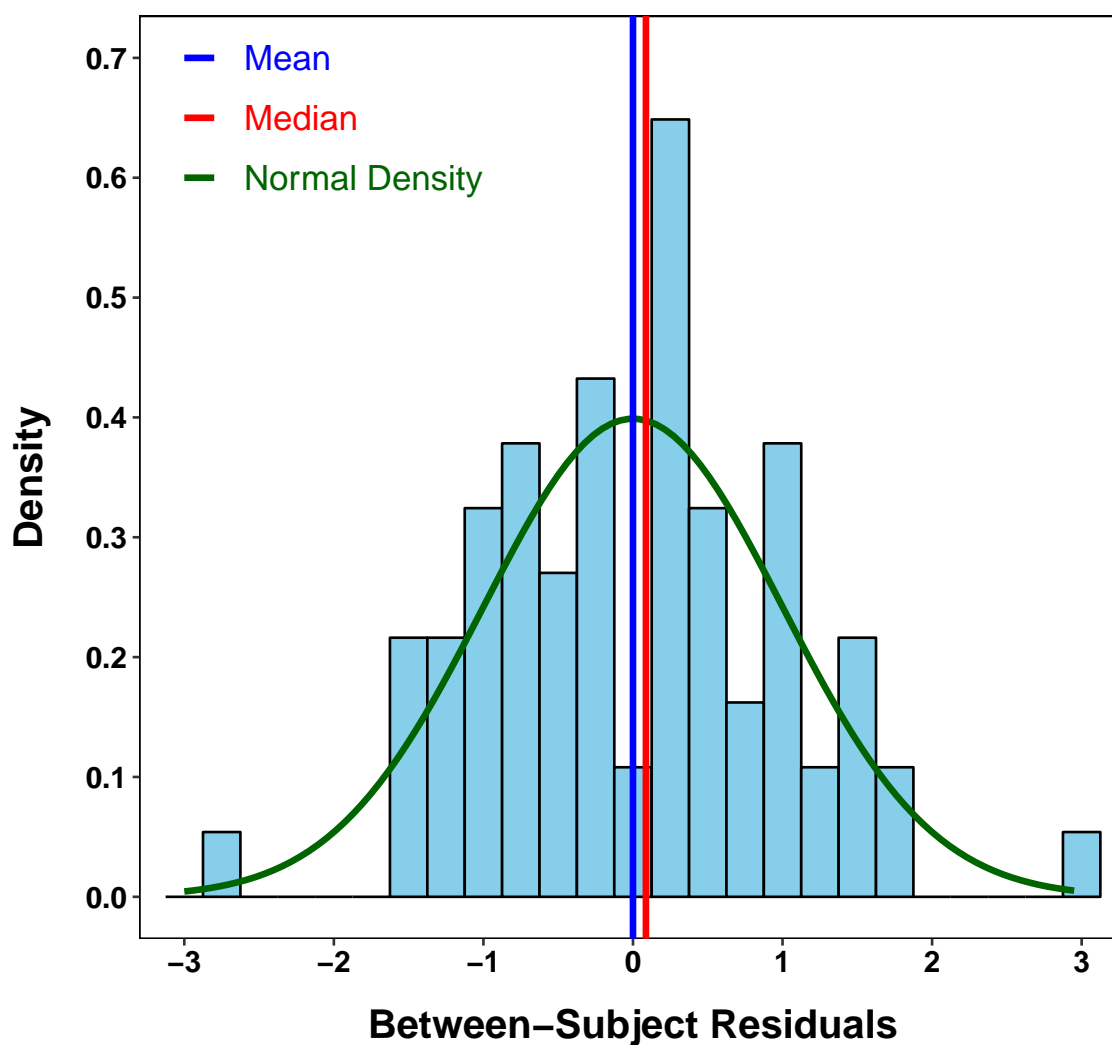
```
ggplot(resid_data_b, aes(x = RS)) + geom_histogram(aes(y = ..density..),
  binwidth = 0.25, color = "black", fill = "skyblue", size = 0.5,
  na.rm = TRUE) + stat_function(fun = dnorm, args = list(mean = mean(resid_data_b$RS,
  na.rm = TRUE), sd = sd(resid_data_b$RS, na.rm = TRUE)), size = 1.25,
  color = "darkgreen") + coord_cartesian(xlim = c(-3, 3), ylim = c(0,
  0.7)) + scale_x_continuous(breaks = c(seq(-3, 3, 1))) + scale_y_continuous(breaks = seq(0,
  0.7, 0.1)) + xlab("Between-Subject Residuals") + ylab("Density") +
  theme(text = element_text(size = 14, family = "sans", color = "black",
    face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
```

```

15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
plot.title = element_text(size = 16, face = "bold", margin = margin(0,
0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
linetype = 1, color = "black"), panel.grid.major = element_blank(),
panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
legend.title = element_blank()) + geom_vline(xintercept = mean(resid_data_b$RS),
size = 1.25, color = "blue") + geom_vline(xintercept = median(resid_data_b$RS),
size = 1.25, color = "red") + annotate("segment", x = -3, xend = -2.8,
y = 0.7, yend = 0.7, color = "blue", linetype = 1, size = 1.25) +
annotate("text", x = -2.6, y = 0.7, label = "Mean", color = "blue",
size = 5, hjust = 0) + annotate("segment", x = -3, xend = -2.8,
y = 0.65, yend = 0.65, color = "red", linetype = 1, size = 1.25) +
annotate("text", x = -2.6, y = 0.65, label = "Median", color = "red",
size = 5, hjust = 0) + annotate("segment", x = -3, xend = -2.8,
y = 0.6, yend = 0.6, color = "darkgreen", linetype = 1, size = 1.25) +
annotate("text", x = -2.6, y = 0.6, label = "Normal Density",
color = "darkgreen", size = 5, hjust = 0) + ggtitle("Distribution of Between-Subjects Residuals")

```

## Distribution of Between-Subjects Residuals



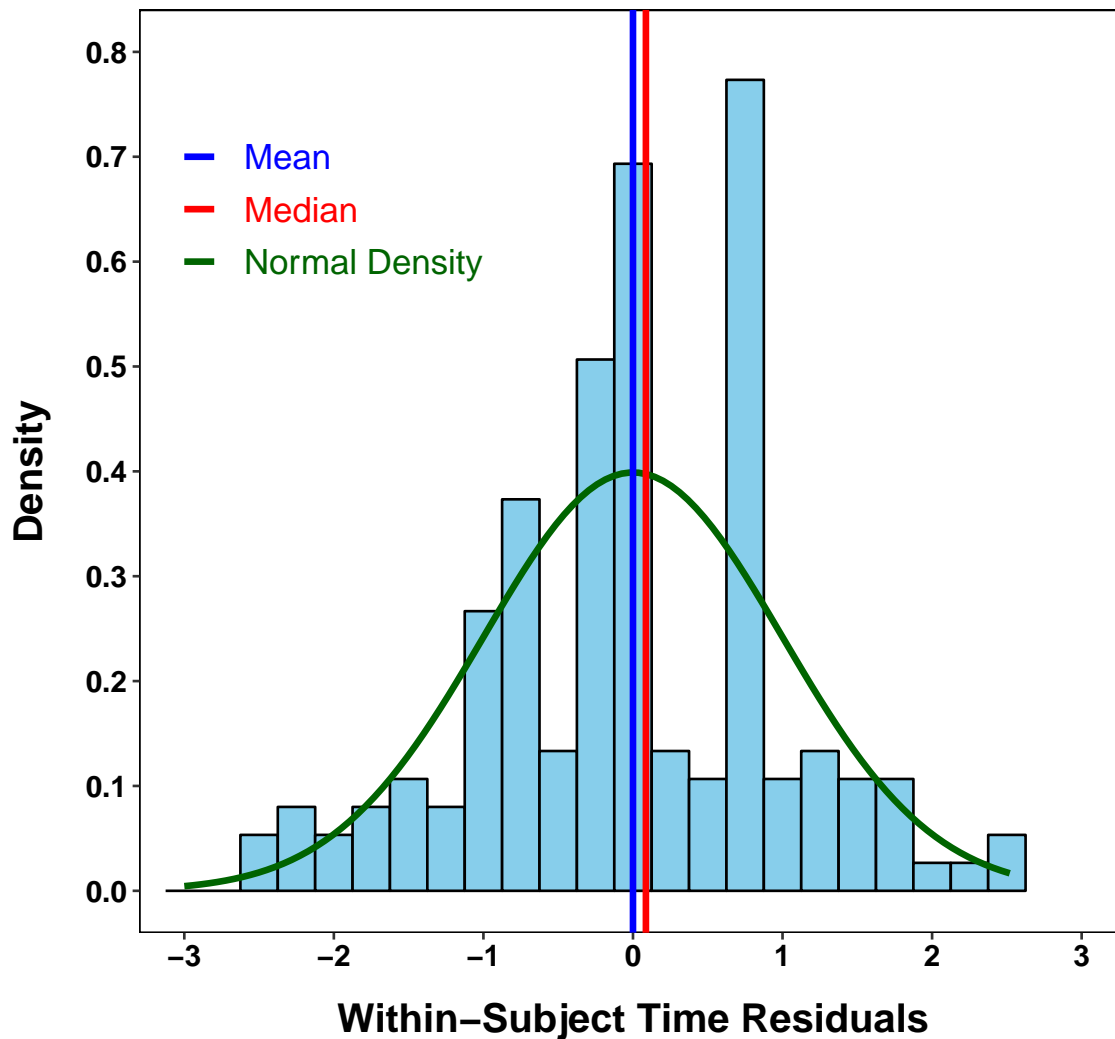
```
ggplot(resid_data_w_Time, aes(x = RS)) + geom_histogram(aes(y = ..density..),
  binwidth = 0.25, color = "black", fill = "skyblue", size = 0.5,
  na.rm = TRUE) + stat_function(fun = dnorm, args = list(mean = mean(resid_data_w_Time$RS,
  na.rm = TRUE), sd = sd(resid_data_w_Time$RS, na.rm = TRUE)), size = 1.25,
  color = "darkgreen") + coord_cartesian(xlim = c(-3, 3), ylim = c(0,
  0.8)) + scale_x_continuous(breaks = c(seq(-3, 3, 1))) + scale_y_continuous(breaks = seq(0,
  0.8, 0.1)) + xlab("Within-Subject Time Residuals") + ylab("Density") +
  theme(text = element_text(size = 14, family = "sans", color = "black",
    face = "bold"), axis.text.y = element_text(colour = "black",
    size = 12, face = "bold"), axis.text.x = element_text(colour = "black",
    size = 12, face = "bold", angle = 0), axis.title.x = element_text(margin = margin(15,
    0, 0, 0), size = 16), axis.title.y = element_text(margin = margin(0,
    15, 0, 0), size = 16), axis.line.x = element_blank(), axis.line.y = element_blank(),
  plot.title = element_text(size = 16, face = "bold", margin = margin(0,
```

```

    0, 20, 0), hjust = 0.5), panel.background = element_rect(fill = "white",
    linetype = 1, color = "black"), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), plot.background = element_rect(fill = "white"),
    plot.margin = unit(c(1, 1, 1, 1), "cm"), legend.position = "bottom",
    legend.title = element_blank()) + geom_vline(xintercept = mean(resid_data_b$RS),
    size = 1.25, color = "blue") + geom_vline(xintercept = median(resid_data_b$RS),
    size = 1.25, color = "red") + annotate("segment", x = -3, xend = -2.8,
    y = 0.7, yend = 0.7, color = "blue", linetype = 1, size = 1.25) +
    annotate("text", x = -2.6, y = 0.7, label = "Mean", color = "blue",
    size = 5, hjust = 0) + annotate("segment", x = -3, xend = -2.8,
    y = 0.65, yend = 0.65, color = "red", linetype = 1, size = 1.25) +
    annotate("text", x = -2.6, y = 0.65, label = "Median", color = "red",
    size = 5, hjust = 0) + annotate("segment", x = -3, xend = -2.8,
    y = 0.6, yend = 0.6, color = "darkgreen", linetype = 1, size = 1.25) +
    annotate("text", x = -2.6, y = 0.6, label = "Normal Density",
    color = "darkgreen", size = 5, hjust = 0) + ggtitle("Distribution of Within-Subject Time Residuals")

```

## Distribution of Within-Subject Time Residuals



*The normality assumption is largely met for these data. The violations are mild, the residuals are symmetric, and there is not evidence of extreme outliers.*

### 3.5 Homogeneity of Covariance Matrices

*The variance-covariance matrices for repeated measures are assumed to be homogeneous across between-subjects conditions. This is examined with Box's M test, but must be viewed with caution. The test is also sensitive to violations of normality.*

```
Box <- boxM(Data_Wide[3:5], Data_Wide$Group_F)

Box

##
## Box's M-test for Homogeneity of Covariance Matrices
```

```
##
## data: Data_Wide[3:5]
## Chi-Sq (approx.) = 4.2, df = 12, p-value = 1

Box$cov

## $`Group 1`
##      DV1    DV2    DV3
## DV1 0.9567 0.255 0.285
## DV2 0.2550 1.193 0.145
## DV3 0.2850 0.145 1.390
##
## $`Group 2`
##      DV1    DV2    DV3
## DV1 1.0267 0.4950 0.4183
## DV2 0.4950 1.1233 0.3617
## DV3 0.4183 0.3617 0.8933
##
## $`Group 3`
##      DV1    DV2    DV3
## DV1 1.2767 0.3700 0.5633
## DV2 0.3700 1.1233 0.3683
## DV3 0.5633 0.3683 1.3067

Box$pooled

##      DV1    DV2    DV3
## DV1 1.0867 0.3733 0.4222
## DV2 0.3733 1.1467 0.2917
## DV3 0.4222 0.2917 1.1967
```

*The variance homogeneity part of the assumption can be checked by Levene's test. It needs to be conducted separately for each of the repeated measures. As with other assumption tests based on significance, it becomes very sensitive as sample size grows. Mild violations are not a concern.*

```
leveneTest(Data_Wide[, 3], Data_Wide$Group_F)

## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 2      0.1   0.91
##      72

leveneTest(Data_Wide[, 4], Data_Wide$Group_F)

## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 2      0     1
##      72

leveneTest(Data_Wide[, 5], Data_Wide$Group_F)

## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 2     0.14  0.87
##      72
```

*There are no violations of the homogeneity assumptions.*

### 3.6 Sphericity

*Repeated measures designs also must satisfy the sphericity assumption. Sphericity is met when there is homogeneity of the variances for all possible difference scores. This will often be violated in any time-based design. Violations of sphericity are especially problematic because they can inflate the Type I error rate.*

*A test for sphericity is available in the ez package with the ezANOVA() function. This function also allows specifying the type of sums of squares. Type II sums of square are requested, but because we have a balanced design, the results will duplicate those found earlier with the aov() function (which uses Type I sums of squares).*

```
# Fit a standard analysis of variance.
ANOVA_2 <- ezANOVA(Data_Long, dv = DV, wid = Subject_F, within = .(Time_F),
  between = Group_F, detailed = TRUE, return_aov = TRUE, type = 2)
ANOVA_2$ANOVA

##           Effect DFn DFd      SSn  SSd      F      p p<.05
## 1      (Intercept)   1   72 7022.44 134.5 3759.04 6.924e-64    *
## 2          Group_F   2   72  49.39 134.5   13.22 1.289e-05    *
## 3          Time_F   2  144  66.75 112.5   42.74 2.690e-15    *
## 4 Group_F:Time_F   4  144  71.47 112.5   22.88 1.199e-14    *
##      ges
## 1 0.9660
## 2 0.1667
## 3 0.2128
## 4 0.2244

ANOVA_2$`Mauchly's Test for Sphericity`

##           Effect      W      p p<.05
## 3          Time_F 0.9836 0.5558
## 4 Group_F:Time_F 0.9836 0.5558

ANOVA_2$`Sphericity Corrections`

##           Effect   GGe   p[GG] p[GG]<.05   HFe   p[HF]
## 3          Time_F 0.9839 4.341e-15    * 1.011 2.690e-15
## 4 Group_F:Time_F 0.9839 1.901e-14    * 1.011 1.199e-14
##   p[HF]<.05
## 3      *
## 4      *
```

*The sphericity assumption is met for these data.*

### 3.7 Correlations

*With more than one variable, we will likely be interested in the correlations among the variables. These need to be obtained with care when there is an experimental group structure to the data. The imposed group differences can artificially inflate or deflate correlations. This artifact can be eliminated by correlating the residuals.*



```

# Run linear models to get residuals.
DV1_LM <- lm(Data_Wide$DV1 ~ -1 + Data_Wide$G1 + Data_Wide$G2 + Data_Wide$G3)
DV2_LM <- lm(Data_Wide$DV2 ~ -1 + Data_Wide$G1 + Data_Wide$G2 + Data_Wide$G3)
DV3_LM <- lm(Data_Wide$DV3 ~ -1 + Data_Wide$G1 + Data_Wide$G2 + Data_Wide$G3)
# Correlations among original variables.
cor(Data_Wide[, 3:5])

##          DV1      DV2      DV3
## DV1  1.0000  0.1934  0.2800
## DV2  0.1934  1.0000  0.2898
## DV3  0.2800  0.2898  1.0000

# Create a matrix of residuals.
R <- cbind(resid(DV1_LM), resid(DV2_LM), resid(DV3_LM))
# Correlations among residuals.
cor(R)

##          [,1]    [,2]    [,3]
## [1,]  1.0000  0.3344  0.3703
## [2,]  0.3344  1.0000  0.2490
## [3,]  0.3703  0.2490  1.0000

```

## 4 Multi-Level Model Approach

*An alternative is to use the `lmer()` function from the `lme4` package. This package is used for multilevel models, of which a repeated measures design is an example. In this case, the repeated measures are treated as nested within subjects. The dummy codes for time and group are used in a no-intercept model so that the coefficients are the cell means for all Group x Time combinations.*

*The `glht()` function from the `multcomp` package can be used to create very specific linear combinations of the means. Because all of the means from the 3 x 3 design are included in a single vector of coefficients, the information contained in the L and M matrices is combined in each vector of the matrix of coefficients.*

```

ANOVA_3 <- lmer(Data_Long$DV ~ -1 + G1:T1 + G1:T2 + G1:T3 + +G2:T1 +
  G2:T2 + G2:T3 + +G3:T1 + G3:T2 + G3:T3 + (1 | Subject), data = Data_Long)
summary(ANOVA_3)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## Data_Long$DV ~ -1 + G1:T1 + G1:T2 + G1:T3 + +G2:T1 + G2:T2 +
##      G2:T3 + +G3:T1 + G3:T2 + G3:T3 + (1 | Subject)
##      Data: Data_Long
##
## REML criterion at convergence: 651.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.2072 -0.5817  0.0304  0.6325  2.8018
##
## Random effects:
##      Groups      Name                Variance Std.Dev.

```

```
## Subject (Intercept) 0.362 0.602
## Residual 0.781 0.884
## Number of obs: 225, groups: Subject, 75
##
## Fixed effects:
## Estimate Std. Error t value
## G1:T1 5.040 0.214 23.6
## G1:T2 4.880 0.214 22.8
## G1:T3 5.160 0.214 24.1
## T1:G2 4.880 0.214 22.8
## T2:G2 6.960 0.214 32.5
## T3:G2 6.680 0.214 31.2
## T1:G3 4.880 0.214 22.8
## T2:G3 6.960 0.214 32.5
## T3:G3 4.840 0.214 22.6
##
## Correlation of Fixed Effects:
## G1:T1 G1:T2 G1:T3 T1:G2 T2:G2 T3:G2 T1:G3 T2:G3
## G1:T2 0.317
## G1:T3 0.317 0.317
## T1:G2 0.000 0.000 0.000
## T2:G2 0.000 0.000 0.000 0.317
## T3:G2 0.000 0.000 0.000 0.317 0.317
## T1:G3 0.000 0.000 0.000 0.000 0.000 0.000
## T2:G3 0.000 0.000 0.000 0.000 0.000 0.000 0.317
## T3:G3 0.000 0.000 0.000 0.000 0.000 0.000 0.317 0.317

LM_Matrix = matrix(c(1, 0, -1, 1, 0, -1, 1, 0, -1, 1, -2, 1, 1, -2,
1, 1, -2, 1, 0, 0, 0, 0, 0, 0, 1, 0, -1, 1, -2, 1, 0, 0, 0, 0,
0, 0, 1, 1, 1, -1, -1, -1, 0, 0, 0, 1, 0, -1, -1, 0, 1, 0, 0,
0, -1, 2, -1, 0, 0, 0, 1, -2, 1, 1, 0, 0, 1, 0, 0, -2, 0, 0),
nrow = 8, ncol = 9, byrow = TRUE)
rownames(LM_Matrix) <- c("Overall Linear", "Overall Quadratic", "G3:Linear",
"G1:Quadratic", "G1 vs G2", "Linear: G1 vs G2", "Quadratic: G1 vs G3",
"T1: G1 and G2 vs G3")
LM_Matrix

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## Overall Linear 1 0 -1 1 0 -1 1 0 -1
## Overall Quadratic 1 -2 1 1 -2 1 1 -2 1
## G3:Linear 0 0 0 0 0 0 1 0 -1
## G1:Quadratic 1 -2 1 0 0 0 0 0 0
## G1 vs G2 1 1 1 -1 -1 -1 0 0 0
## Linear: G1 vs G2 1 0 -1 -1 0 1 0 0 0
## Quadratic: G1 vs G3 -1 2 -1 0 0 0 1 -2 1
## T1: G1 and G2 vs G3 1 0 0 1 0 0 -2 0 0

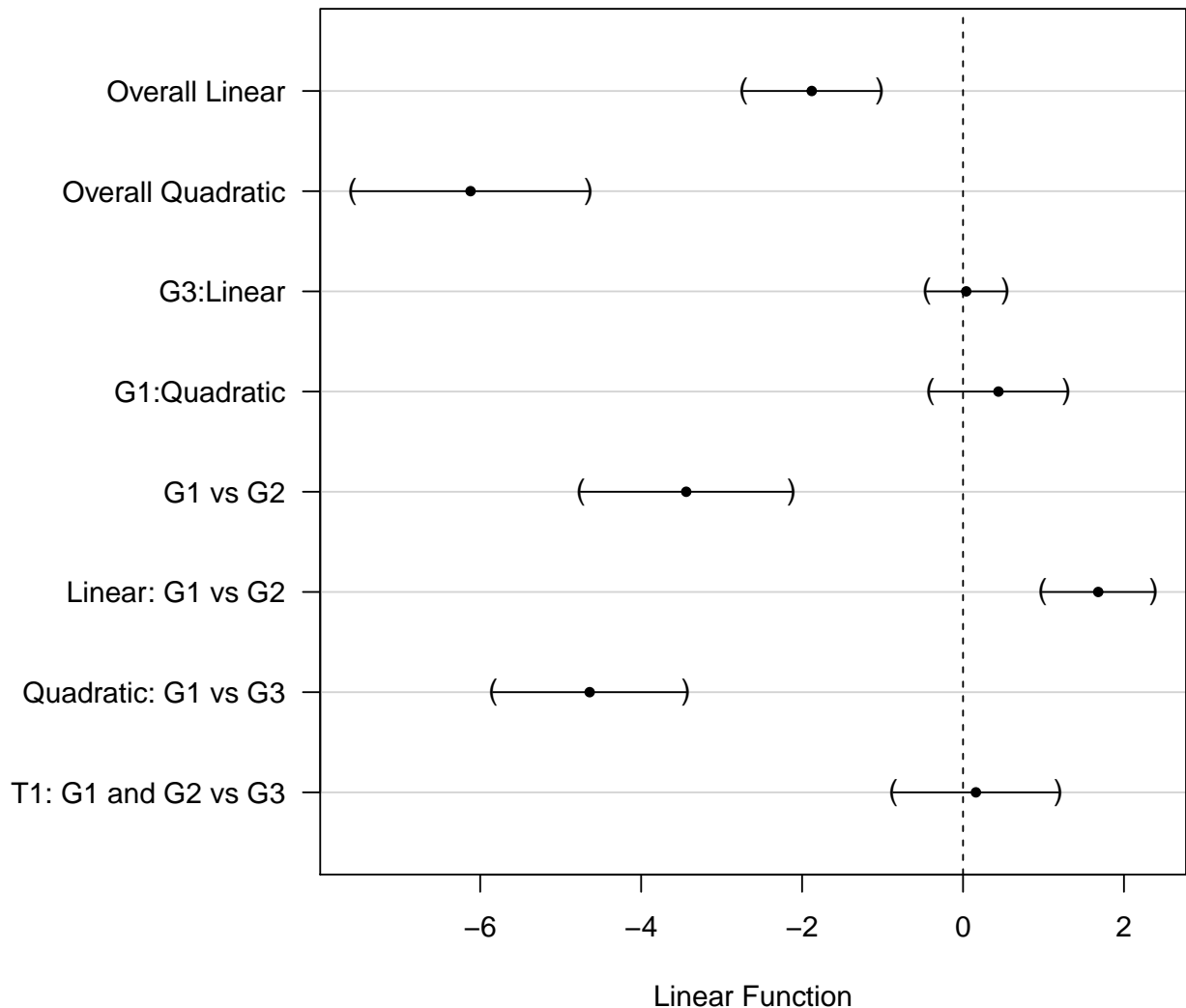
glht_LM_Matrix <- glht(ANOVA_3, linfct = LM_Matrix, alternative = "two.sided",
rhs = 0)
summary(glht_LM_Matrix, adjusted("holm"))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lmer(formula = Data_Long$DV ~ -1 + G1:T1 + G1:T2 + G1:T3 + +G2:T1 +
```

```
##      G2:T2 + G2:T3 + +G3:T1 + G3:T2 + G3:T3 + (1 | Subject), data = Data_Long)
##
## Linear Hypotheses:
##              Estimate Std. Error z value Pr(>|z|)
## Overall Linear == 0      -1.880      0.433   -4.34  5.6e-05
## Overall Quadratic == 0    -6.120      0.750   -8.16  1.8e-15
## G3:Linear == 0           0.040      0.250    0.16   1.00
## G1:Quadratic == 0         0.440      0.433    1.02   0.93
## G1 vs G2 == 0           -3.440      0.670   -5.14  1.7e-06
## Linear: G1 vs G2 == 0      1.680      0.353    4.75  1.0e-05
## Quadratic: G1 vs G3 == 0   -4.640      0.612   -7.58  2.4e-13
## T1: G1 and G2 vs G3 == 0    0.160      0.524    0.31   1.00
## (Adjusted p values reported -- holm method)

par(mai = c(1, 2, 1, 0))
plot(conftint(glht_LM_Matrix, calpha = univariate_calpha()), main = "95% Confidence Intervals")
```

## 95% Confidence Intervals



*If we specify a random intercept for each case, we assume compound symmetry among the repeated measures. This is a special case of sphericity. That this is being imposed on the data is apparent in the equivalence of the standard errors for each coefficient in the fixed effects (which are the means) and in the equivalent fixed effect correlations within each group. We can compare the F ratios from the two approaches by retrieving the ANOVA summary from a standard model specification with `lmer()`. They duplicate those obtained earlier with the `aov()` function.*

*The compound symmetry assumption can be relaxed. This can be done with the `lme()` function from the `nlme` package, which allows more complex error structures. The `gls()` function from the same package can also be used for more complex error structures.*

```
# Original F ratios
summary(ANOVA_1)
```

```
##
## Error: Subject_F
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Group_F    2   49.4   24.69    13.2 0.000013
## Residuals 72  134.5     1.87
##
## Error: Subject_F:Time_F
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Time_F      2   66.7   33.4    42.7 2.7e-15
## Time_F:Group_F  4   71.5   17.9    22.9 1.2e-14
## Residuals    144  112.5     0.8

ANOVA_4 <- lmer(Data_Long$DV ~ 1 + Group_F * Time_F + (1 | Subject),
  data = Data_Long)
summary(ANOVA_4)

## Linear mixed model fit by REML ['lmerMod']
## Formula: Data_Long$DV ~ 1 + Group_F * Time_F + (1 | Subject)
## Data: Data_Long
##
## REML criterion at convergence: 664.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.2072 -0.5817  0.0304  0.6325  2.8018
##
## Random effects:
## Groups Name Variance Std.Dev.
## Subject (Intercept) 0.362  0.602
## Residual          0.781  0.884
## Number of obs: 225, groups: Subject, 75
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    5.5867    0.0911   61.31
## Group_F1      -0.5600    0.1289   -4.35
## Group_F2       0.5867    0.1289    4.55
## Time_F1       -0.6533    0.0833   -7.84
## Time_F2        0.6800    0.0833    8.16
## Group_F1:Time_F1 0.6667    0.1178    5.66
## Group_F2:Time_F1 -0.6400    0.1178   -5.43
## Group_F1:Time_F2 -0.8267    0.1178   -7.02
## Group_F2:Time_F2  0.1067    0.1178    0.91
##
## Correlation of Fixed Effects:
##              (Intr) Grp_F1 Grp_F2 Tim_F1 Tim_F2 G_F1:T_F1
## Group_F1      0.000
## Group_F2      0.000 -0.500
## Time_F1       0.000  0.000  0.000
## Time_F2       0.000  0.000  0.000 -0.500
## Grp_F1:T_F1   0.000  0.000  0.000  0.000  0.000
## Grp_F2:T_F1   0.000  0.000  0.000  0.000  0.000 -0.500
## Grp_F1:T_F2   0.000  0.000  0.000  0.000  0.000 -0.500
## Grp_F2:T_F2   0.000  0.000  0.000  0.000  0.000  0.250
```

```
##          G_F2:T_F1 G_F1:T_F2
## Group_F1
## Group_F2
## Time_F1
## Time_F2
## Grp_F1:T_F1
## Grp_F2:T_F1
## Grp_F1:T_F2  0.250
## Grp_F2:T_F2 -0.500   -0.500

# F ratios from the mixed model approach:
anova(ANOVA_4)

## Analysis of Variance Table
##          Df Sum Sq Mean Sq F value
## Group_F      2   20.6    10.3    13.2
## Time_F        2   66.7    33.4    42.7
## Group_F:Time_F  4   71.5    17.9    22.9

# Original F ratios:
summary(ANOVA_1)

##
## Error: Subject_F
##          Df Sum Sq Mean Sq F value   Pr(>F)
## Group_F      2   49.4    24.69    13.2 0.000013
## Residuals 72  134.5     1.87
##
## Error: Subject_F:Time_F
##          Df Sum Sq Mean Sq F value   Pr(>F)
## Time_F      2   66.7    33.4    42.7 2.7e-15
## Time_F:Group_F  4   71.5    17.9    22.9 1.2e-14
## Residuals    144  112.5     0.8
```