

Univariate Review I

Mike Strube

September 3, 2018

1 Overview

The primary goal of inferential statistics is to provide a basis for confidently claiming that a theory-predicted "signal" is present in the data. Statistical tests can be thought of as "signal-to-noise" ratios:

$$\text{Test} = \frac{\text{Signal} + \text{Noise}}{\text{Noise}}$$

When the signal is "large enough" relative to the noise, we may claim it to be a "significant" signal.

Another way to frame the basic inferential test is as an "effect" relative to its standard error.

$$\text{Test} = \frac{\text{Signal} + \text{Noise}}{\text{Noise}} = \frac{\text{Effect}}{SE_{\text{Effect}}}$$

The standard error (SE), or standard deviation of the effect, tells us how much the effect can be expected to vary under the null hypothesis that there is no signal. If the effect is unusual in this context, we can claim there is probably a signal present (i.e., not just noise).

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(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(lsmmeans)

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

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

library(ggplot2)

```

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

```
# Get the data from the working directory.
setwd("C:\\Courses\\Psychology 516\\PowerPoint\\2018")
Data_1 <- read.table("univariate_t_test_example.csv", sep = ",", header = TRUE)
Data_1 <- as.data.frame(Data_1)

Data_2 <- read.table("Set_1.csv", sep = ",", header = TRUE)
Data_2 <- as.data.frame(Data_2)
```

3 Simple Two-Group Example

The basic idea of statistical tests as signal detection can be illustrated with a two-group example.

3.1 Means and Variability

```
Means <- aggregate(Data_1$DV, by = list(Data_1$Group), FUN = mean)
SDs <- aggregate(Data_1$DV, by = list(Data_1$Group), FUN = sd)
```

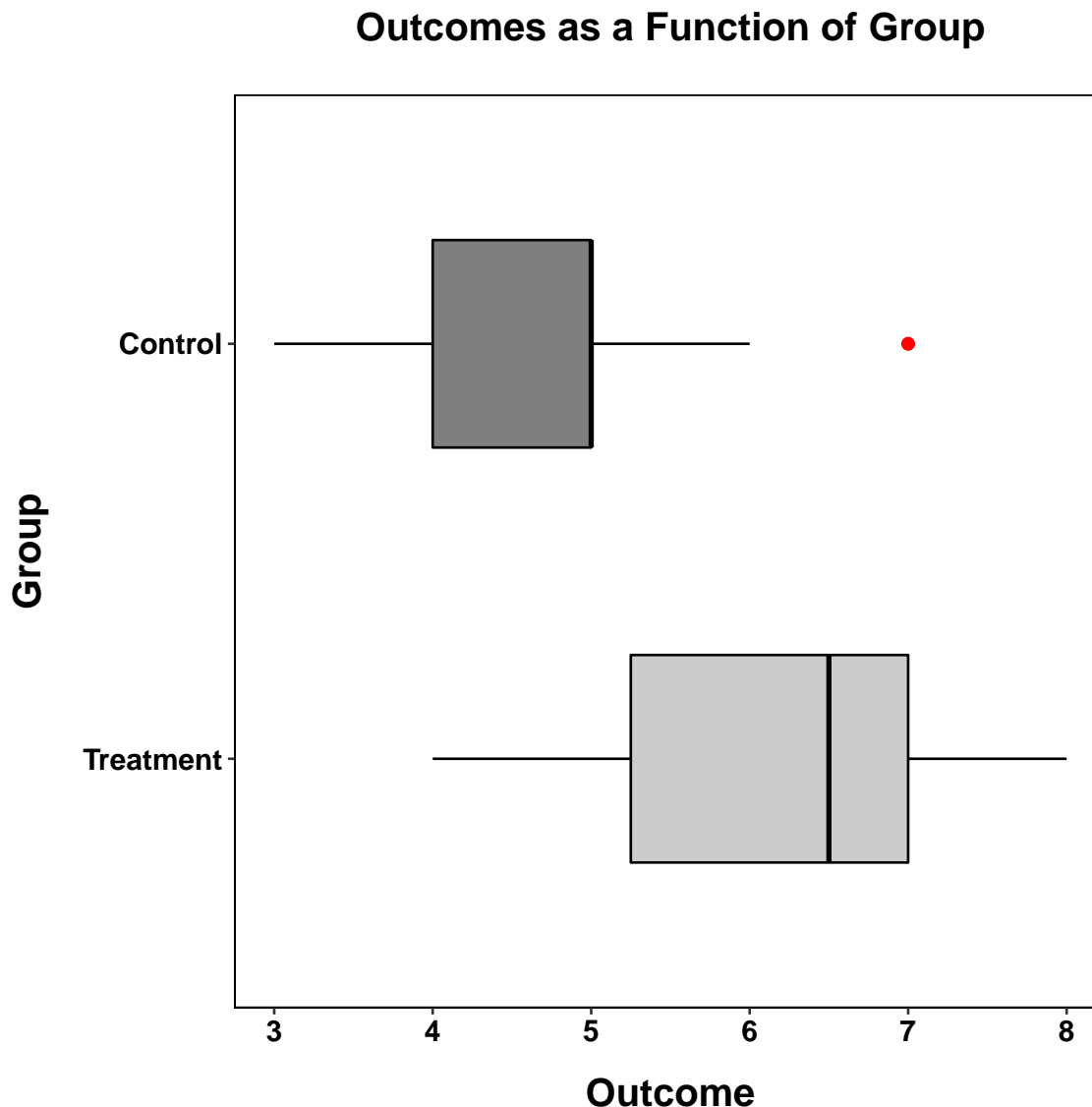
Two-Group Example		
Group	Mean	SE
Treatment	6.3	0.4
Control	4.7	0.4

The effect or signal of interest is the mean difference: $6.3 - 4.7 = 1.6$ Is this big or "significant?" We need to consider variability under the null to know this. This can partly be gauged by examining boxplots, which illustrate separation of the groups (signal + noise) relative to variability within the groups (noise).

```
# Boxplots by group
Data_1$Group_F <- factor(Data_1$Group, levels = c(1, 2), labels = c("Treatment",
  "Control"))

ggplot(Data_1, aes(y = DV, fill = Group_F, x = Group_F)) + geom_boxplot(aes(y = DV,
  fill = Group_F, x = Group_F), fill = c("grey80", "grey50"), color = "black",
  size = 0.5, width = 0.5, outlier.colour = "red", outlier.shape = 19,
  outlier.size = 2, notch = FALSE) + coord_flip() + ylab("Outcome") +
  xlab("Group") + theme(text = element_text(size = 14, family = "sans",
  color = "black", face = "bold"), axis.text.x = element_text(colour = "black",
  size = 12, face = "bold"), axis.text.y = element_text(colour = "black",
  size = 12, face = "bold"), 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")) + ggtitle("Outcomes as a Function of Group")
```



3.2 Sampling Distribution for the Mean Difference

The simple t-test can be thought of as a signal detection test. It takes the mean difference and scales it by the standard error of the mean difference. This produces a t-value, which can be compared to the t-distribution. The t-distribution is assumed to describe the sampling distribution of mean differences under the null hypothesis.

```
t_output <- t.test(DV ~ Group, data = Data_1, var.equal = TRUE)
t_output
```

```
##
## Two Sample t-test
##
## data: DV by Group
## t = 2.8, df = 18, p-value = 0.01
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 0.383 2.817
## sample estimates:
## mean in group 1 mean in group 2
## 6.3 4.7

t_output[[1]]

## t
## 2.762

t_output[[2]]

## df
## 18
```

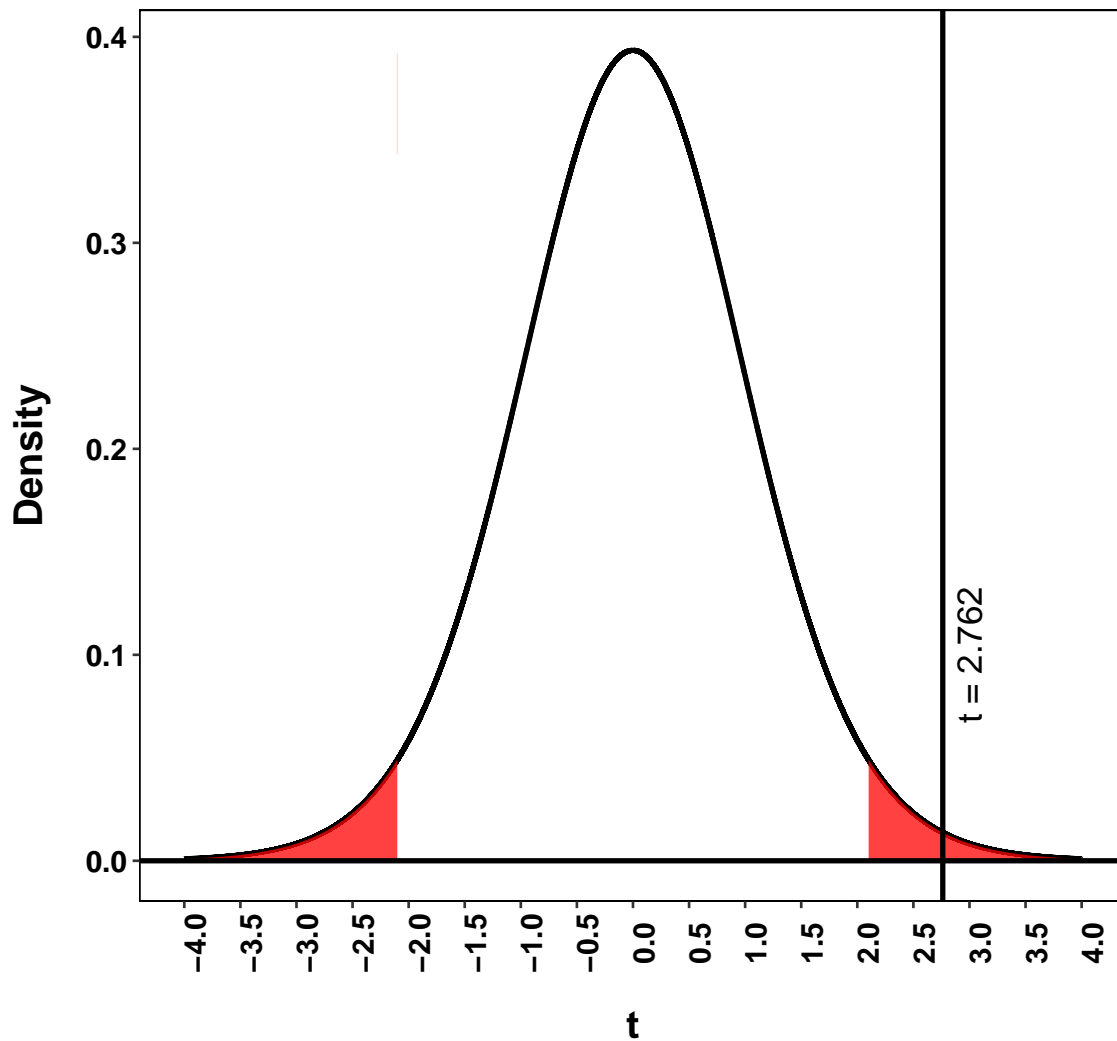
The results of the t-test can be illustrated using the assumed sampling distribution of the mean differences for degrees of freedom equal to 18 (N - 2). In this example, the mean difference is a rare event in the context of the null hypothesis.

```
set.seed(1)
x <- seq(-4, 4, 0.0001)
y <- dt(x, df = 18)

D <- cbind(x, y)
D <- as.data.frame(D)

ggplot(data = D, aes(x = x, y = y)) + geom_line(size = 1) + geom_area(aes(x = ifelse(x >=
  2.101, x, 0)), fill = "red", alpha = 0.75) + geom_area(aes(x = ifelse(x <=
  -2.101, x, -2.101)), fill = "red", alpha = 0.75) + scale_y_continuous(limits = c(0,
  max(D$y))) + scale_x_continuous(breaks = seq(-4, 4, 0.5)) + coord_cartesian(xlim = c(-4,
  4), ylim = c(0, max(D$y))) + xlab("t") + ylab("Density") + theme(text = element_text(size = 14,
  family = "sans", color = "black", face = "bold"), axis.text.x = element_text(colour = "black",
  size = 12, face = "bold", angle = 90), axis.text.y = element_text(colour = "black",
  size = 12, face = "bold"), 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")) + geom_hline(yintercept = 0,
  size = 1, linetype = 1, color = "black") + geom_vline(xintercept = 2.762,
  size = 1, linetype = 1, color = "black") + annotate("text", x = 3,
  y = 0.1, label = "t = 2.762", angle = 90, size = 5) + ggtitle("The t Distribution (df=18, p = .05)")
```

The t Distribution (df=18, p = .05)



4 More Than Two Groups

With more than two groups, an ANOVA is used, but the basic principles remain the same. If, for example, we have five groups rather than two, we are still interested in the variability of the means relative to the variability within the groups.

Group ₁	Group ₂	Group ₃	Group ₄	Group ₅
Y ₁₁	Y ₁₂	Y ₁₃	Y ₁₄	Y ₁₅
Y ₂₁	Y ₂₂	Y ₂₃	Y ₂₄	Y ₂₅
Y ₃₁	Y ₃₂	Y ₃₃	Y ₃₄	Y ₃₅
Y ₄₁	Y ₄₂	Y ₄₃	Y ₄₄	Y ₄₅
Y ₅₁	Y ₅₂	Y ₅₃	Y ₅₄	Y ₅₅
.
.
.
$\bar{Y}_{.1}$	$\bar{Y}_{.2}$	$\bar{Y}_{.3}$	$\bar{Y}_{.4}$	$\bar{Y}_{.5}$

We are interested in the variability of the means as evidence for a treatment-induced signal.

Group ₁	Group ₂	Group ₃	Group ₄	Group ₅
M ₁	M ₂	M ₃	M ₄	M ₅

We get an estimate of the expected means variability in the absence of a signal by examining the variability within the groups. The variability within the groups is assumed to be homogeneous; each provides an estimate of expected variability under the null hypothesis.

Group ₁	Group ₂	Group ₃	Group ₄	Group ₅
M ₁	M ₂	M ₃	M ₄	M ₅
$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	$\hat{\sigma}_3^2$	$\hat{\sigma}_4^2$	$\hat{\sigma}_5^2$

The variance of the means when the null hypothesis is true can be estimated from the variance of scores.

$$\sigma_Y^2 = \frac{\sigma_Y^2}{\sqrt{n}}$$

The ratio of two estimates of score variability, one based on means and one based on within-groups variances, is F distributed under the null hypothesis. As the ratio departs from 1, there is increasing evidence for a signal in the data.

$$F \approx \frac{n\sigma_Y^2}{\bar{\sigma}_Y^2}$$

For our example we will use a data set with three groups and three variables. At the moment, we will treat each variable separately. Later, we will examine linear combinations of the variables.

4.1 Means and Variability

```
# Create a unique dummy code for each of the three groups.
Data_2$G1 <- ifelse(Data_2$Group == 1, 1, 0)
Data_2$G2 <- ifelse(Data_2$Group == 2, 1, 0)
```

```

Data_2$G3 <- ifelse(Data_2$Group == 3, 1, 0)

describeBy(Data_2$DV1, group = Data_2$Group)

##
## Descriptive statistics by group
## group: 1
##   vars  n mean   sd median trimmed  mad min max range skew
## X1    1 25 5.04 0.98      5    5.05 1.48   3  7    4 -0.08
##   kurtosis se
## X1    -0.94 0.2
## -----
## group: 2
##   vars  n mean   sd median trimmed  mad min max range skew
## X1    1 25 4.88 1.01      5      5 1.48   2  6    4 -0.93
##   kurtosis se
## X1     0.62 0.2
## -----
## group: 3
##   vars  n mean   sd median trimmed  mad min max range skew
## X1    1 25 4.88 1.13      5    4.81 1.48   3  8    5 0.72
##   kurtosis se
## X1     0.65 0.23

describeBy(Data_2$DV2, group = Data_2$Group)

##
## Descriptive statistics by group
## group: 1
##   vars  n mean   sd median trimmed  mad min max range skew
## X1    1 25 4.88 1.09      5    4.9 1.48   3  7    4 -0.33
##   kurtosis se
## X1    -0.71 0.22
## -----
## group: 2
##   vars  n mean   sd median trimmed  mad min max range skew
## X1    1 25 6.96 1.06      7      7 1.48   5  9    4 -0.33
##   kurtosis se
## X1    -0.68 0.21
## -----
## group: 3
##   vars  n mean   sd median trimmed  mad min max range skew
## X1    1 25 6.96 1.06      7    6.95 1.48   5  9    4 0.08
##   kurtosis se
## X1    -0.62 0.21

describeBy(Data_2$DV3, group = Data_2$Group)

##
## Descriptive statistics by group
## group: 1
##   vars  n mean   sd median trimmed  mad min max range skew
## X1    1 25 5.16 1.18      5    5.1 1.48   3  8    5 0.58
##   kurtosis se
## X1    -0.23 0.24

```

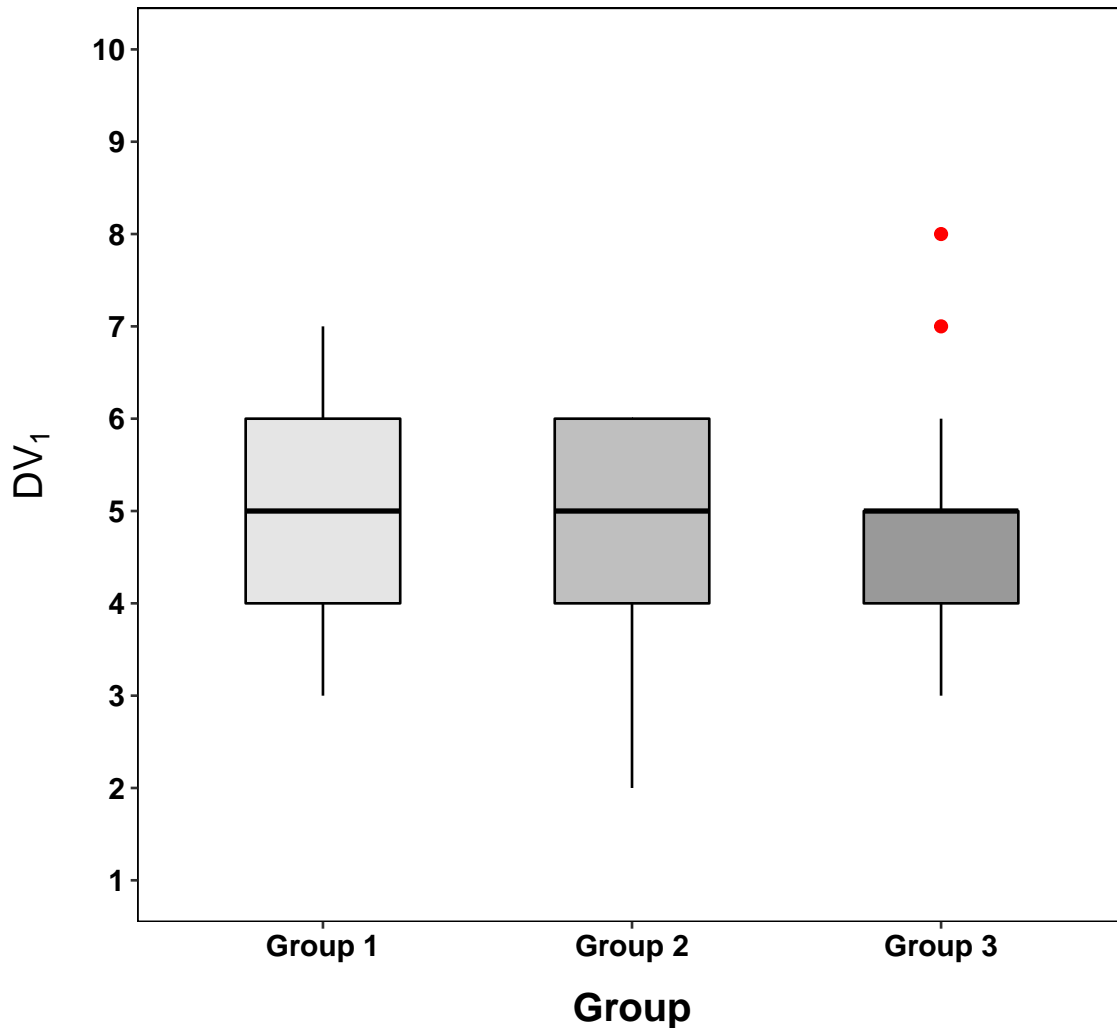


```
## -----
## group: 2
##      vars  n mean    sd median trimmed  mad min max range skew
## X1      1 25 6.68 0.95      6    6.62 1.48   5   9    4 0.63
##      kurtosis  se
## X1      -0.41 0.19
## -----
## group: 3
##      vars  n mean    sd median trimmed  mad min max range skew
## X1      1 25 4.84 1.14      5    4.86 1.48   3   7    4 -0.18
##      kurtosis  se
## X1      -1.06 0.23
```

```
# Boxplots by group
Data_2$Group_F <- factor(Data_2$Group, levels = c(1, 2, 3), labels = c("Group 1",
  "Group 2", "Group 3"))

ggplot(Data_2, aes(y = DV1, fill = Group_F, x = Group_F)) + geom_boxplot(fill = c("grey90",
  "grey75", "grey60"), color = "black", size = 0.5, width = 0.5,
  outlier.colour = "red", outlier.shape = 19, outlier.size = 2,
  notch = FALSE) + scale_y_continuous(breaks = seq(1, 10, 1)) +
  coord_cartesian(xlim = c(1, 3), ylim = c(1, 10)) + ylab(expression(paste(DV[1]))) +
  xlab("Group") + theme(text = element_text(size = 14, family = "sans",
  color = "black", face = "bold"), axis.text.x = element_text(colour = "black",
  size = 12, face = "bold"), axis.text.y = element_text(colour = "black",
  size = 12, face = "bold"), 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")) + ggtitle(expression(paste(DV[1],
  " as a Function of Group")))
```

DV₁ as a Function of Group

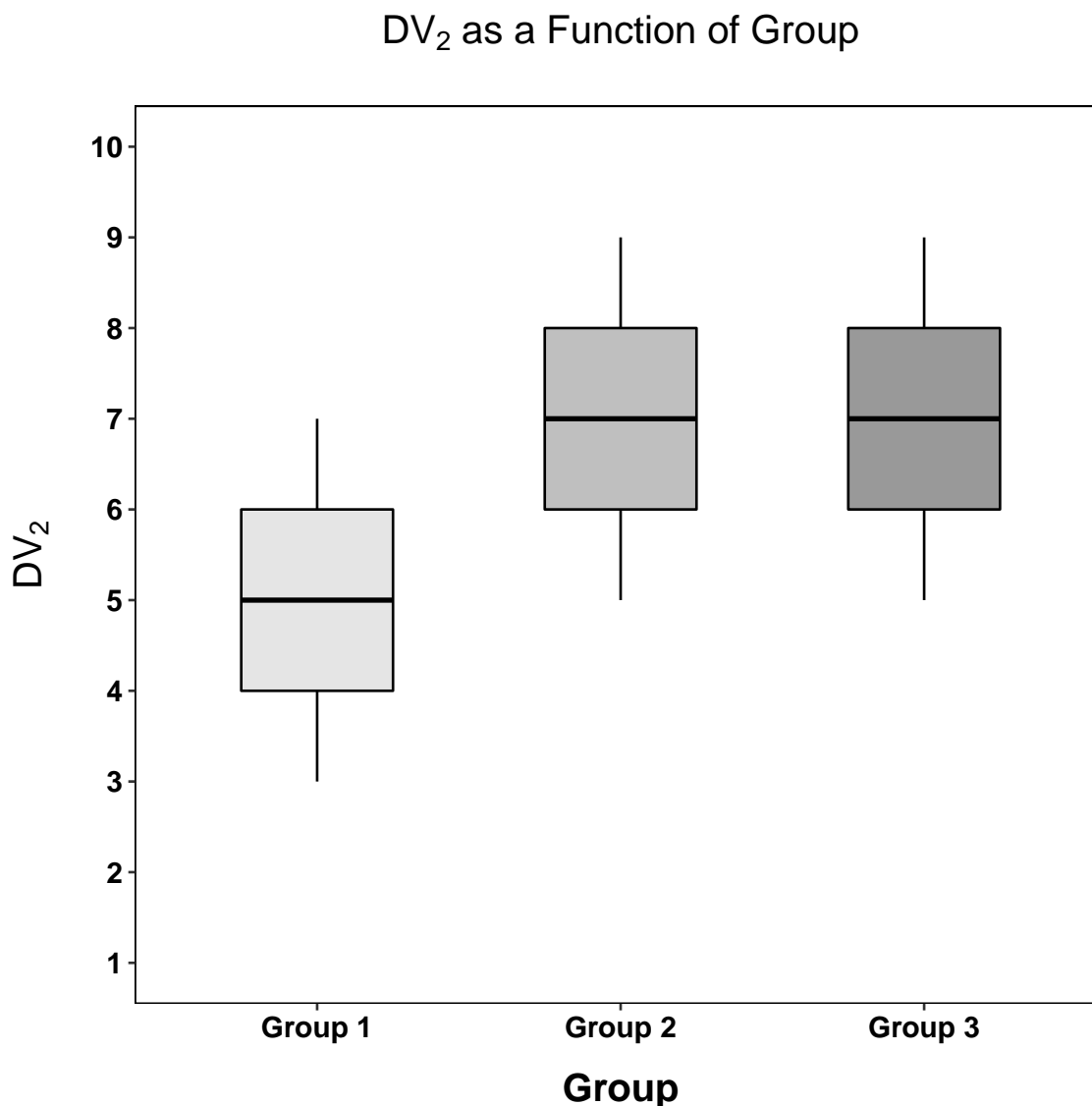


```
# Boxplots by group
ggplot(Data_2, aes(y = DV2, fill = Group_F, x = Group_F)) + geom_boxplot(fill = c("grey90",
"grey75", "grey60"), color = "black", size = 0.5, width = 0.5,
outlier.colour = "red", outlier.shape = 19, outlier.size = 2,
notch = FALSE) + scale_y_continuous(breaks = seq(1, 10, 1)) +
coord_cartesian(xlim = c(1, 3), ylim = c(1, 10)) + ylab(expression(paste(DV[2]))) +
xlab("Group") + theme(text = element_text(size = 14, family = "sans",
color = "black", face = "bold"), axis.text.x = element_text(colour = "black",
size = 12, face = "bold"), axis.text.y = element_text(colour = "black",
size = 12, face = "bold"), 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")) + ggtitle(expression(paste(DV[2],
" as a Function of Group")))

```



```

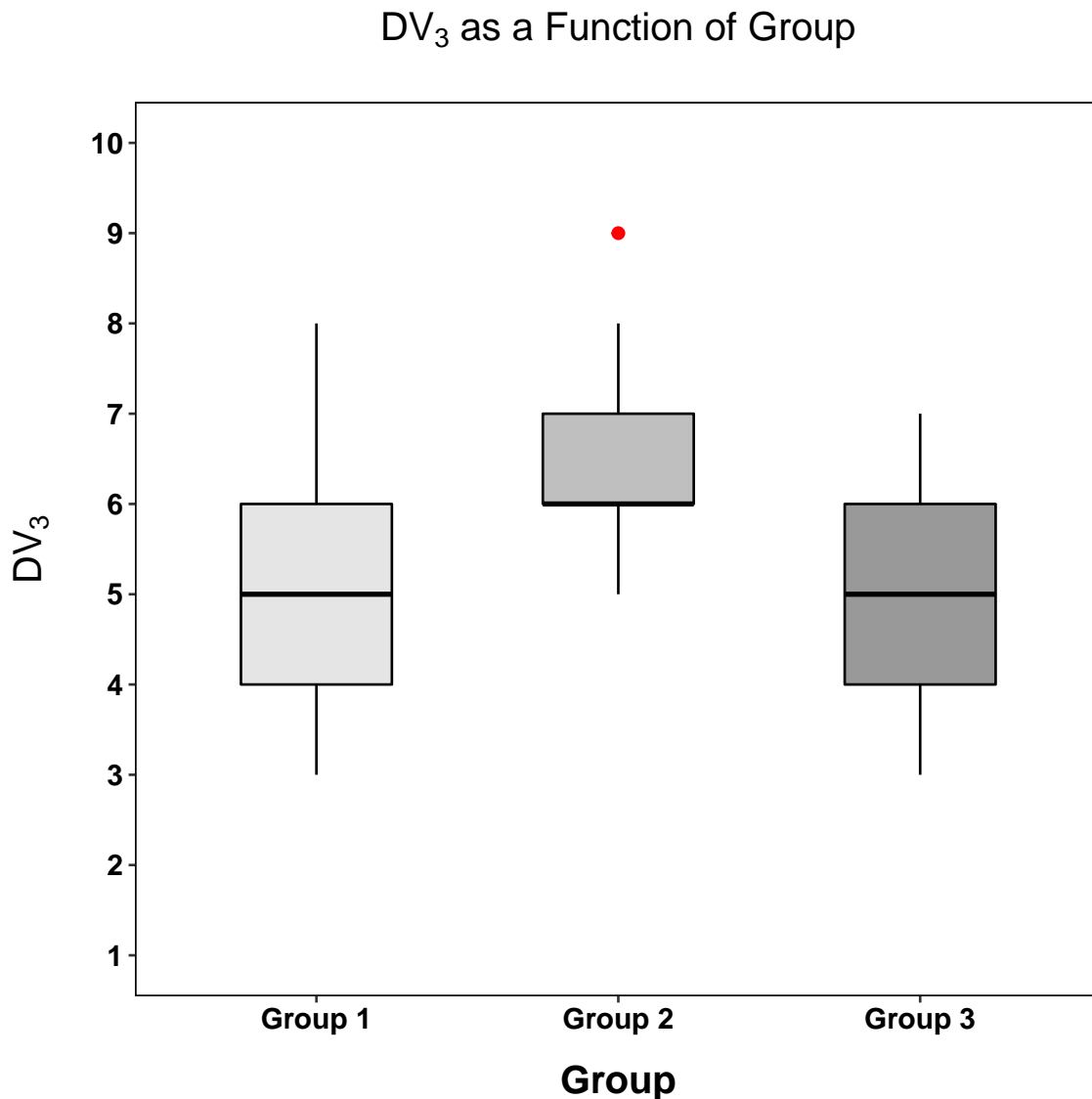
# Boxplots by group
ggplot(Data_2, aes(y = DV3, fill = Group_F, x = Group_F)) + geom_boxplot(fill = c("grey90",
"grey75", "grey60"), color = "black", size = 0.5, width = 0.5,
outlier.colour = "red", outlier.shape = 19, outlier.size = 2,
notch = FALSE) + scale_y_continuous(breaks = seq(1, 10, 1)) +
coord_cartesian(xlim = c(1, 3), ylim = c(1, 10)) + ylab(expression(paste(DV[3]))) +
xlab("Group") + theme(text = element_text(size = 14, family = "sans",
color = "black", face = "bold"), axis.text.x = element_text(colour = "black",
size = 12, face = "bold"), axis.text.y = element_text(colour = "black",

```

```

size = 12, face = "bold"), 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")) + ggtitle(expression(paste(DV[3],
" as a Function of Group")))

```



4.2 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_2$DV1 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
DV2_LM <- lm(Data_2$DV2 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
DV3_LM <- lm(Data_2$DV3 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
# Correlations among original variables.
cor(Data_2[, 2:4])

##          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
```

4.3 ANOVA Results

The analysis of variance produces an F ratio, which is a ratio of two estimates of variance of the scores within-groups under the null hypothesis. One of these is based on the variability of the means. Provided there are no systematic differences among the groups (i.e., no signal), then the variability among the means (when scaled appropriately) will be consistent with the variability within the groups.

```
summary(aov(Data_2$DV1 ~ as.factor(Data_2$Group)))

##                Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Data_2$Group)  2    0.4   0.213    0.2  0.82
## Residuals              72   78.2   1.087

summary(aov(Data_2$DV2 ~ as.factor(Data_2$Group)))

##                Df Sum Sq Mean Sq F value  Pr(>F)
## as.factor(Data_2$Group)  2   72.1    36.1   31.4 1.5e-10
## Residuals              72   82.6     1.1

summary(aov(Data_2$DV3 ~ as.factor(Data_2$Group)))

##                Df Sum Sq Mean Sq F value      Pr(>F)
## as.factor(Data_2$Group)  2   48.3    24.2   20.2 0.00000011
## Residuals              72   86.2     1.2
```

The results of the F-tests can be illustrated using the assumed F sampling distribution for degrees of freedom equal to 2 and 72.

```

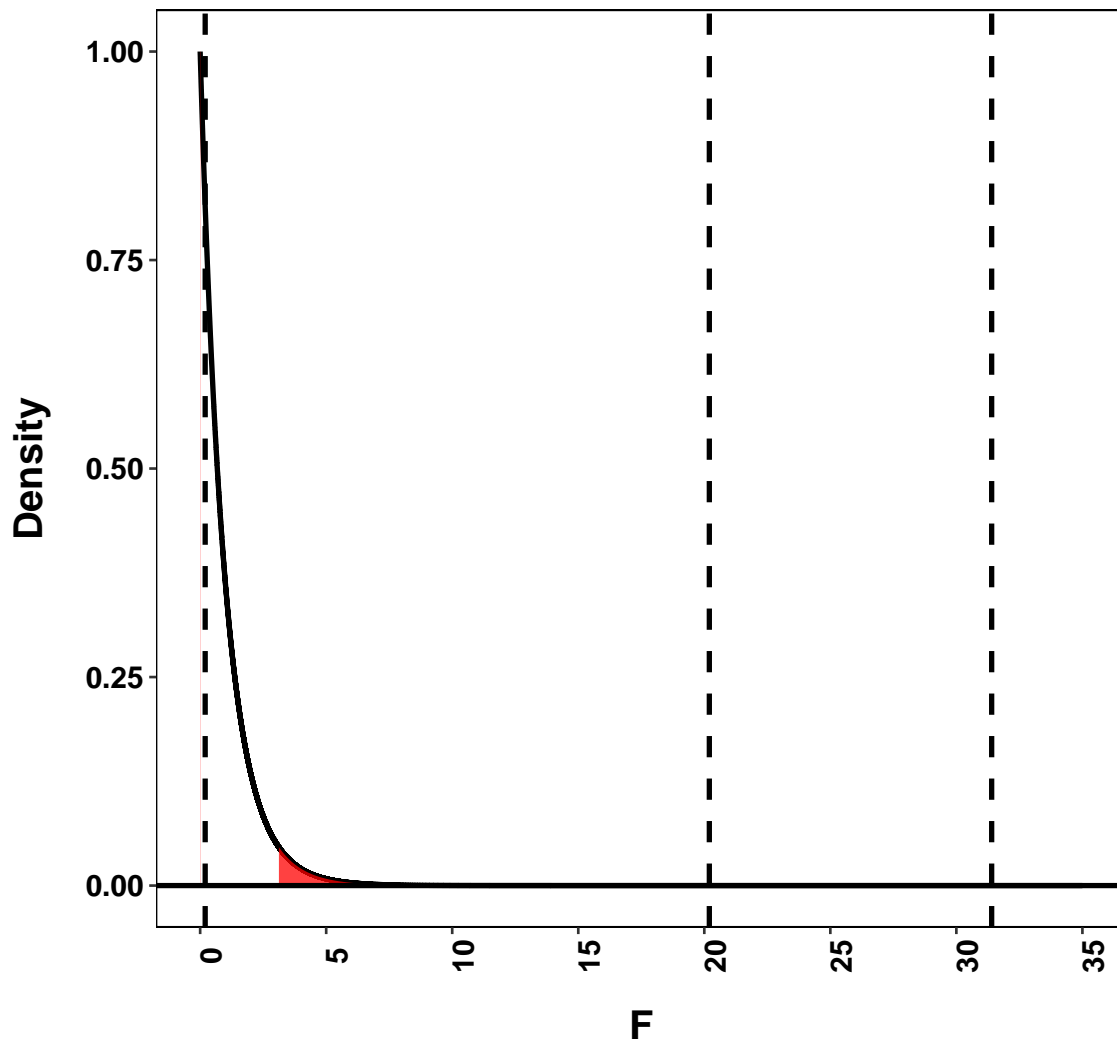
set.seed(1)
x <- seq(0, 35, 0.0001)
y <- df(x, df1 = 2, df2 = 72)

D <- cbind(x, y)
D <- as.data.frame(D)

ggplot(data = D, aes(x = x, y = y)) + geom_line(size = 1) + geom_area(aes(x = ifelse(x >=
  3.124, x, 0)), fill = "red", alpha = 0.75) + scale_y_continuous(limits = c(0,
  max(D$y))) + scale_x_continuous(breaks = seq(0, 35, 5)) + coord_cartesian(xlim = c(0,
  35), ylim = c(0, max(D$y))) + xlab("F") + ylab("Density") + theme(text = element_text(size = 14,
  family = "sans", color = "black", face = "bold"), axis.text.x = element_text(colour = "black",
  size = 12, face = "bold", angle = 90), axis.text.y = element_text(colour = "black",
  size = 12, face = "bold"), 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")) + geom_hline(yintercept = 0,
  size = 1, linetype = 1, color = "black") + geom_vline(xintercept = 0.2,
  size = 1, linetype = 2, color = "black") + geom_vline(xintercept = 31.4,
  size = 1, linetype = 2, color = "black") + geom_vline(xintercept = 20.2,
  size = 1, linetype = 2, color = "black") + ggtitle("The F Distribution (df = 2, 72, p = .05)")

```

The F Distribution (df = 2, 72, p = .05)



4.4 Homogeneity of Variance

The ANOVA uses a single estimate of the within-group variability, called the residual mean square. This is a pooling of the separate within-group variances. This pooling is appropriate provided the separate estimates come from the same population; this is the homogeneity of variances assumption.

```
leveneTest(Data_2$DV1 ~ as.factor(Data_2$Group), data = Data_2)

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

leveneTest(Data_2$DV2 ~ as.factor(Data_2$Group), data = Data_2)
```

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

leveneTest(Data_2$DV3 ~ as.factor(Data_2$Group), data = Data_2)

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

4.5 Pairwise Comparisons

In the presence of a significant effect, we likely will want to conduct follow-up comparisons to better understand its nature. Pairwise comparisons are common, but need to be conducted using adjusted p-values to keep the family-wise error rate under control.

```
pairwise.t.test(Data_2$DV2, as.factor(Data_2$Group), p.adj = "holm")

##
## Pairwise comparisons using t tests with pooled SD
##
## data: Data_2$DV2 and as.factor(Data_2$Group)
##
##      1      2
## 2 6e-09 -
## 3 6e-09 1
##
## P value adjustment method: holm

TukeyHSD(aov(Data_2$DV2 ~ as.factor(Data_2$Group)))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = Data_2$DV2 ~ as.factor(Data_2$Group))
##
## $`as.factor(Data_2$Group)`
##      diff      lwr      upr p adj
## 2-1  2.080e+00  1.3552  2.8048    0
## 3-1  2.080e+00  1.3552  2.8048    0
## 3-2 -8.882e-16 -0.7248  0.7248    1
```

4.6 L Matrix Approach

Maximum flexibility comes from using the L matrix approach to tailor group comparisons. To use this approach, we specify a "no intercept" model, which produces group means as coefficients. We can then form linear combinations of these coefficients.

```
# Now run no-intercept models, saving the results in an object.
DV1_LM <- lm(Data_2$DV1 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
DV2_LM <- lm(Data_2$DV2 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
```



```

DV3_LM <- lm(Data_2$DV3 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
# The coefficients will now be the group means.
summary(DV1_LM)

##
## Call:
## lm(formula = Data_2$DV1 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.88  -0.88   0.12   0.96   3.12
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## Data_2$G1      5.040      0.208    24.2  <2e-16
## Data_2$G2      4.880      0.208    23.4  <2e-16
## Data_2$G3      4.880      0.208    23.4  <2e-16
##
## Residual standard error: 1.04 on 72 degrees of freedom
## Multiple R-squared:  0.959, Adjusted R-squared:  0.957
## F-statistic: 560 on 3 and 72 DF,  p-value: <2e-16

summary(DV2_LM)

##
## Call:
## lm(formula = Data_2$DV2 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.96  -0.92   0.04   1.04   2.12
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## Data_2$G1      4.880      0.214    22.8  <2e-16
## Data_2$G2      6.960      0.214    32.5  <2e-16
## Data_2$G3      6.960      0.214    32.5  <2e-16
##
## Residual standard error: 1.07 on 72 degrees of freedom
## Multiple R-squared:  0.973, Adjusted R-squared:  0.972
## F-statistic: 877 on 3 and 72 DF,  p-value: <2e-16

summary(DV3_LM)

##
## Call:
## lm(formula = Data_2$DV3 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.16  -0.68  -0.16   0.84   2.84
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## Data_2$G1      5.160      0.219    23.6  <2e-16

```

```
## Data_2$G2      6.680      0.219      30.5 <2e-16
## Data_2$G3      4.840      0.219      22.1 <2e-16
##
## Residual standard error: 1.09 on 72 degrees of freedom
## Multiple R-squared:  0.965, Adjusted R-squared:  0.963
## F-statistic: 659 on 3 and 72 DF,  p-value: <2e-16
```

The multcomp package can be used to specify the weighted combinations to test.

```
L_Matrix = matrix(c(1, -1, 0, 1, 0, -1, 0, 1, -1, 1, 1, -2, 2, -1,
                    -1, 1, -2, 1), nrow = 6, ncol = 3, byrow = TRUE)
rownames(L_Matrix) <- c("G1 vs G2", "G1 vs G3", "G2 vs G3", "G1 and G2 vs G3",
                        "G1 vs G2 and G3", "G2 vs G1 and G3")
L_Matrix

##           [,1] [,2] [,3]
## G1 vs G2      1  -1   0
## G1 vs G3      1   0  -1
## G2 vs G3      0   1  -1
## G1 and G2 vs G3  1   1  -2
## G1 vs G2 and G3  2  -1  -1
## G2 vs G1 and G3  1  -2   1

glht_L_Matrix_1 <- glht(DV1_LM, linfct = L_Matrix, alternative = "two.sided",
                        rhs = 0)
summary(glht_L_Matrix_1, adjusted("holm"))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = Data_2$DV1 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
##
## Linear Hypotheses:
##
##           Estimate Std. Error t value Pr(>|t|)
## G1 vs G2 == 0      0.160      0.295    0.54      1
## G1 vs G3 == 0      0.160      0.295    0.54      1
## G2 vs G3 == 0      0.000      0.295    0.00      1
## G1 and G2 vs G3 == 0 0.160      0.511    0.31      1
## G1 vs G2 and G3 == 0 0.320      0.511    0.63      1
## G2 vs G1 and G3 == 0 0.160      0.511    0.31      1
## (Adjusted p values reported -- holm method)

glht_L_Matrix_2 <- glht(DV2_LM, linfct = L_Matrix, alternative = "two.sided",
                        rhs = 0)
summary(glht_L_Matrix_2, adjusted("holm"))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = Data_2$DV2 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
##
## Linear Hypotheses:
##
##           Estimate Std. Error t value Pr(>|t|)
## G1 vs G2 == 0     -2.080      0.303   -6.87 9.7e-09
## G1 vs G3 == 0     -2.080      0.303   -6.87 9.7e-09
```

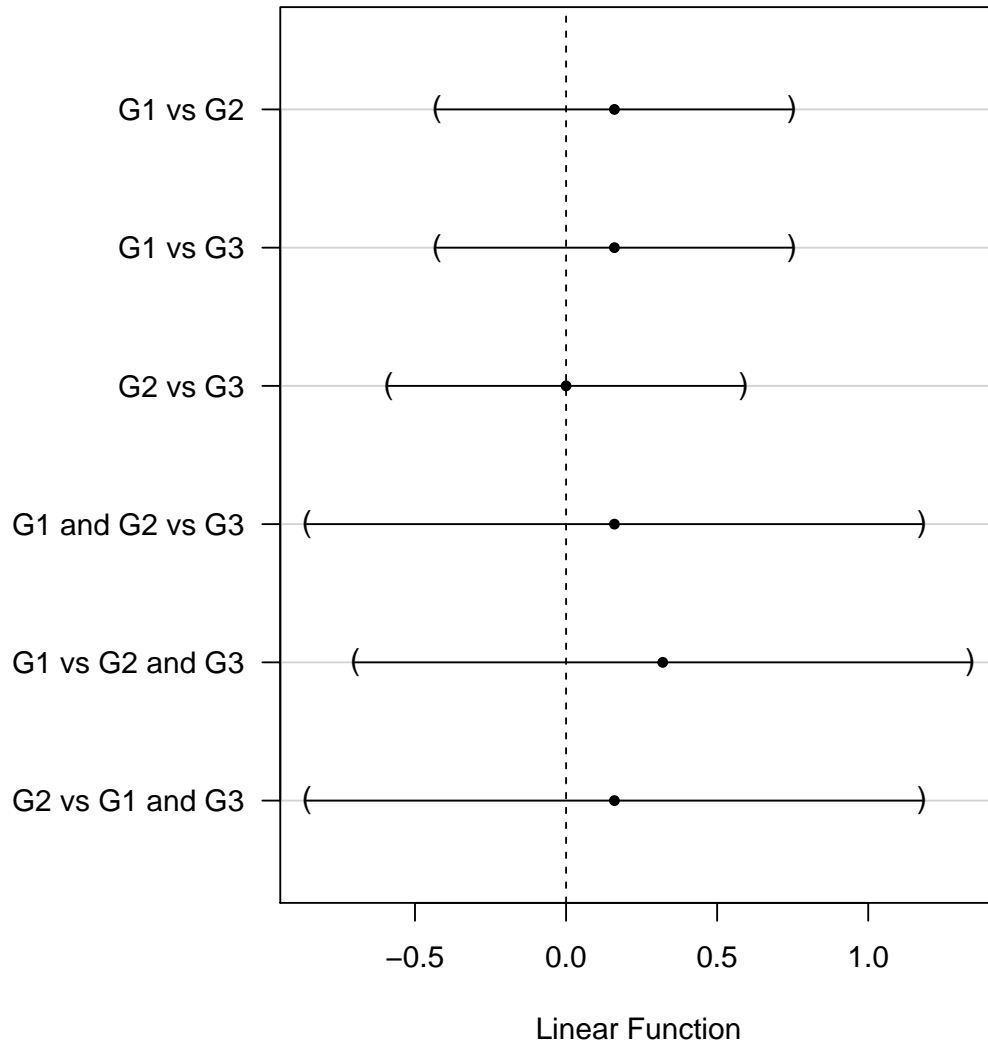
```
## G2 vs G3 == 0          0.000      0.303    0.00  1.00000
## G1 and G2 vs G3 == 0  -2.080      0.525   -3.96  0.00051
## G1 vs G2 and G3 == 0  -4.160      0.525   -7.93  1.2e-10
## G2 vs G1 and G3 == 0  -2.080      0.525   -3.96  0.00051
## (Adjusted p values reported -- holm method)

glht_L_Matrix_3 <- glht(DV3_LM, linfct = L_Matrix, alternative = "two.sided",
  rhs = 0)
summary(glht_L_Matrix_3, adjusted("holm"))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = Data_2$DV3 ~ -1 + Data_2$G1 + Data_2$G2 + Data_2$G3)
##
## Linear Hypotheses:
##              Estimate Std. Error t value    Pr(>|t|)
## G1 vs G2 == 0      -1.520      0.309   -4.91 0.00002179
## G1 vs G3 == 0       0.320      0.309    1.03  0.30449
## G2 vs G3 == 0       1.840      0.309    5.95 0.00000045
## G1 and G2 vs G3 == 0  2.160      0.536    4.03  0.00041
## G1 vs G2 and G3 == 0  -1.200      0.536   -2.24  0.05647
## G2 vs G1 and G3 == 0  -3.360      0.536   -6.27 0.00000014
## (Adjusted p values reported -- holm method)

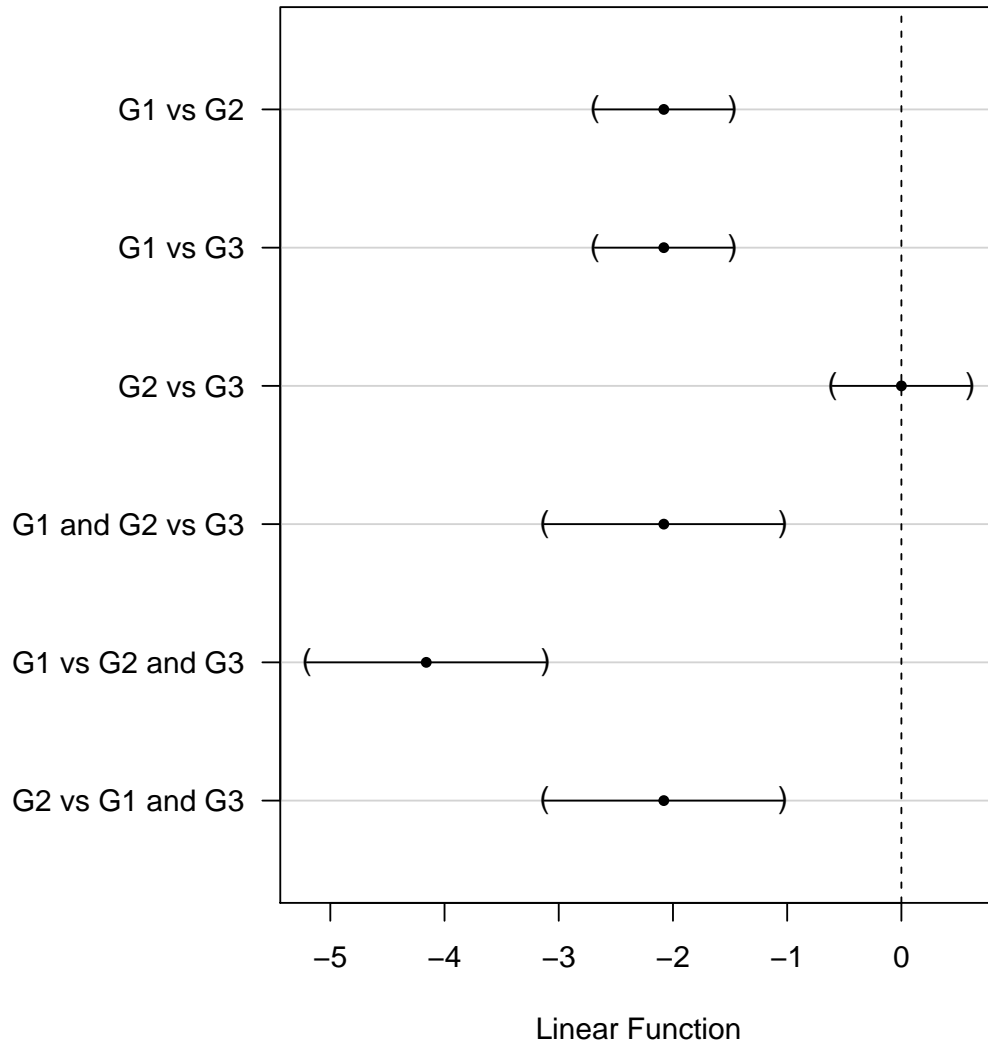
par(mai = c(1, 2, 1, 1))
plot(confint(glht_L_Matrix_1, calpha = univariate_calpha()), main = "95% Confidence Interval: DV 1")
```

95% Confidence Interval: DV 1



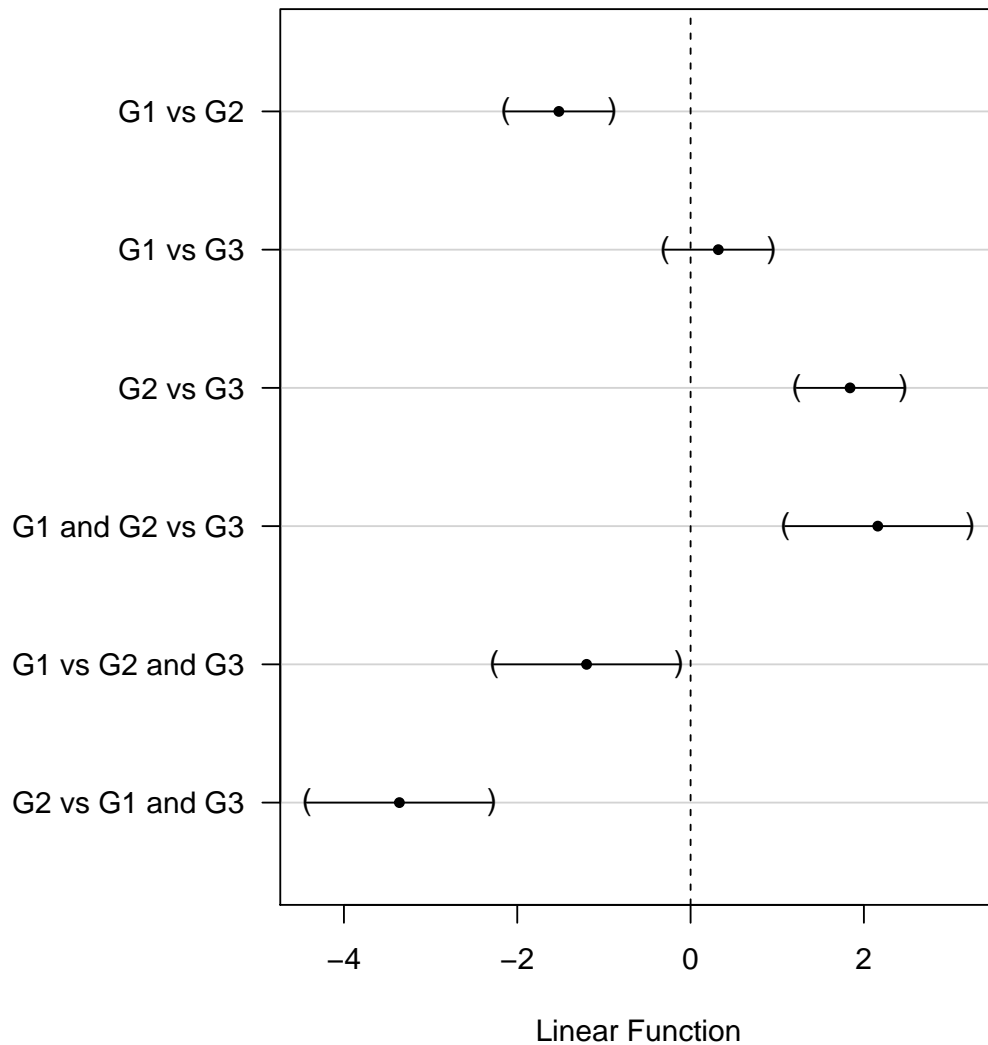
```
plot(confint(glht_L_Matrix_2, calpha = univariate_calpha()), main = "95% Confidence Interval: DV 2")
```

95% Confidence Interval: DV 2



```
plot(confint(glht_L_Matrix_3, calpha = univariate_calpha()), main = "95% Confidence Interval: DV 3")
```

95% Confidence Interval: DV 3



When the L matrix or vector is applied to the variance-covariance matrix (V) of the elements using $L'VL$, the result is the variance-covariance matrix for the linear combination(s). The following uses the variance-covariance matrix from the linear model of DV2 and the first comparison. The result is the standard error for that comparison (see summary output).

```
V <- as.matrix(vcov(DV2_LM))
L1 <- as.matrix(L_Matrix[1, ])
V
```

	Data_2\$G1	Data_2\$G2	Data_2\$G3
## Data_2\$G1	0.04587	0.00000	0.00000
## Data_2\$G2	0.00000	0.04587	0.00000
## Data_2\$G3	0.00000	0.00000	0.04587

```
L1
```

```
##          [,1]
## [1,]      1
## [2,]     -1
## [3,]      0

t(L1) %*% V %*% L1

##          [,1]
## [1,] 0.09173

sqrt(t(L1) %*% V %*% L1)

##          [,1]
## [1,] 0.3029
```