Testing Multiple Dependent Variables, Hotellings T2, the MANOVA Procedure, and Canonical Correlation Analysis

Shane T. Mueller shanem@mtu.edu

2019-04-23

Suppose you measured two groups or conditions and have a lot of relatively weak measures of difference. You might find that each of the measures are correlated, and measure related constructs. Whether or not the individual measures show significant differences between groups, it might be nice to construct a composite of the DVs. But what is the right way to do that?

There are some problems to think about. Different measures might be negatively correlated, so averaging them would wipe out any effect. Or they may be of different magnitudes (e.g., completion time in seconds for one task, and accuracy for another). If you averaged these, one would swamp the other. Even if you had the same scale (two likert-scale responses), differences in variance could make averaging together problematic.

Keeping them separate is also not always ideal. This produces multiple tests, and so you would have a greater chance of a false alarm. On the other hand, you may have several weak signals–each of which are noisy and none of which are significantly different, even if a composite score might be.

The MANOVA is an extension of the ANOVA that attempts to run an ANOVA on a number of dependent measures, combining them optimally.

We will start with an example data set in which three outcome variables are each weakly related to a group membership:

library(Hotelling)
library(ggplot2)
library(ICSNP)
library(reshape2)
set.seed(1000)
n <- 350
group <- rep(1:2, each = n)
xs <- rnorm(n \* 2, mean = group \* 0.2, sd = 1.4)
ys <- rnorm(n \* 2, mean = 4 + group \* 1.5, sd = 15)
zs <- rnorm(n \* 2, mean = 3 - group \* 0.2, sd = 0.9)

ggplot(data.frame(group, xs, ys), aes(x = xs, y = ys, color = group)) + geom\_point() +
 labs(title = "X versus Y")



ggplot(data.frame(group, xs, zs), aes(x = xs, y = zs, color = group)) + geom\_point() +
 labs(title = "X versus Z")



long <- melt(data.frame(group, xs, ys, zs), id.vars = c("group"))
long$group <- as.factor(long$group)
ggplot(data = long, aes(x = (group), y = value)) + geom\_violin(aes(fill = group),
 colour = "grey20", scale = "width") + facet\_wrap(~variable) + theme\_minimal() +
 scale\_fill\_manual(values = c("gold", "cadetblue3"))



Notice that the three values we measured are on different scales, and they differ in different ways across groups.

t.test(xs ~ group)

 Welch Two Sample t-test

data: xs by group
t = -1.3874, df = 697.9, p-value = 0.1658
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.33887615 0.05825459
sample estimates:
mean in group 1 mean in group 2
 0.2250053 0.3653161

t.test(ys ~ group)

 Welch Two Sample t-test

data: ys by group
t = -1.7294, df = 696.93, p-value = 0.08419
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -4.2298935 0.2680513
sample estimates:
mean in group 1 mean in group 2
 5.006313 6.987234

t.test(zs ~ group)

 Welch Two Sample t-test

data: zs by group
t = 1.7669, df = 697.25, p-value = 0.07768
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.01332771 0.25302765
sample estimates:
mean in group 1 mean in group 2
 2.75949 2.63964

None of these three measures differ significantly at a p<.05 level. Furthermore, the values are on different scales, and for some of them group 1 is higher, while for others group 1 is lower. If we had a high ICC, we might consider making a composite score-averaging these together, and trying to look at the measure that way. In this case, this would not be possible unless we either had some theory (expectations about positive/negative difference and scaling), or did some normalization to bring the measures all onto the same scale. So in an ideal world, maybe we would know something about the data and do this:

x2 <- xs/0.2
y2 <- (ys - 4)/1.5
z2 <- -(zs - 3)/(0.2)

tmp <- data.frame(group = as.factor(rep(group, 3)), val = c(x2, y2, z2), xyz = rep(c("X",
 "Y", "Z"), each = n))

ggplot(tmp, aes(x = group, y = val, col = group)) + geom\_violin(aes(fill = group),
 colour = "grey20", scale = "width") + geom\_jitter(color = "grey30", width = 0.05,
 height = 0) + facet\_wrap(~xyz) + theme\_minimal() + scale\_fill\_manual(values = c("gold",
 "cadetblue3"))



composite <- rowMeans(cbind(x2, y2, z2))
t.test(tmp$val ~ as.factor(tmp$group))

 Welch Two Sample t-test

data: tmp$val by as.factor(tmp$group)
t = -2.6858, df = 2096.8, p-value = 0.007292
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.511823 -0.235789
sample estimates:
mean in group 1 mean in group 2
 0.9994848 1.8732907

If we get lucky and have a good way of combining multiple values, we can still do a simple t-test. An alternative is the Hotelling $T^{2}$ test. This tests whether there is a difference between the groups on the three measures altogether. Hotelling’s $T^{2}$ is available from several packages. But the way you call it differs across packages, so you need to be careful:

## Hotelling test from ICSNP

HotellingsT2(cbind(xs, ys, zs) ~ group)

 Hotelling's two sample T2-test

data: cbind(xs, ys, zs) by group
T.2 = 2.7408, df1 = 3, df2 = 696, p-value = 0.04244
alternative hypothesis: true location difference is not equal to c(0,0,0)

## hotelling.test from Hotelling package

Previously, the Hotelling package had a function called hotel.test that worked differently from HotellingsT2. In recent versions this has been replaced by hotelling.test, that permits specifying the dependent measures either as a matrix or with $+$ syntax on the left of the $ $ sign:

## From Hotelling package
print(hotelling.test(cbind(xs, ys, zs) ~ group)) ##This won't work in earlier versions of hotel.test.

Test stat: 2.7408
Numerator df: 3
Denominator df: 696
P-value: 0.04244

print(hotelling.test(xs + ys + zs ~ group)) ##correct; value is the same as ICSNP

Test stat: 2.7408
Numerator df: 3
Denominator df: 696
P-value: 0.04244

Notice that the sign and scale of the components don’t matter– z is negatively related to the others, but if we reverse code it, the results are the same: ##Negatively code z

z2 <- 5 - zs

print(hotelling.test(xs + ys + z2 ~ group))

Test stat: 2.7408
Numerator df: 3
Denominator df: 696
P-value: 0.04244

print(HotellingsT2(cbind(xs, ys, z2) ~ group))

 Hotelling's two sample T2-test

data: cbind(xs, ys, z2) by group
T.2 = 2.7408, df1 = 3, df2 = 696, p-value = 0.04244
alternative hypothesis: true location difference is not equal to c(0,0,0)

Similarly, if we reduce scale of one of the components (y), everything stays the same:

y2 <- ys/10

print(hotelling.test(xs + y2 + z2 ~ group))

Test stat: 2.7408
Numerator df: 3
Denominator df: 696
P-value: 0.04244

print(HotellingsT2(cbind(xs, y2, z2) ~ group))

 Hotelling's two sample T2-test

data: cbind(xs, y2, z2) by group
T.2 = 2.7408, df1 = 3, df2 = 696, p-value = 0.04244
alternative hypothesis: true location difference is not equal to c(0,0,0)

So, Hotelling’s $T^{2}$ provides a way of creating sort of a composite score, without worrying about the scale or coding direction of the components. Notice that each individual component did not differ significantly across groups, but the set of three did. This also protects against multiple testing–if we’d measured 10 things and expected them all to differ a little but none did, we’d still have a high chance of a type-II false alarm error. Here, we do just one test. However, notice that the $T^{2}$ was not as strong as our “theory-based” composite score above, so we could do better if we have a principled method for combining values (and we were right).

## Extending to Multiple Groups

Just as we use one-way and multi-way ANOVA to generalize the t-test, we use the MANOVA (‘multivariate analysis of variance’) to generalize $T^{2}$. Sometimes people refer to the Wilks’s Lambda as the alternative to MANOVA corresponding to the one-way ANOVA, but this is mostly nomenclature. R will sometimes do this automatically if given the right inputs. The simplest version is of course the two-group version, and we can see that the car library Anova will produce the same F test as Hotelling. The Manova() function is basically a specialized name for the same analysis.

library(car)
lm(cbind(xs, ys, zs) ~ group)

Call:
lm(formula = cbind(xs, ys, zs) ~ group)

Coefficients:
 xs ys zs
(Intercept) 0.08469 3.02539 2.87934
group 0.14031 1.98092 -0.11985

Anova(lm(cbind(xs, ys, zs) ~ group))

Type II MANOVA Tests: Pillai test statistic
 Df test stat approx F num Df den Df Pr(>F)
group 1 0.011676 2.7408 3 696 0.04244 \*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Manova(lm(cbind(xs, y2, z2) ~ group))

Type II MANOVA Tests: Pillai test statistic
 Df test stat approx F num Df den Df Pr(>F)
group 1 0.011676 2.7408 3 696 0.04244 \*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Again, it doesn’t matter the scale or direction of the individual DVs.

what happens when we have three groups? We can actually give a multiple-column dependent measure to lm, but it does not really do what we might expect:

set.seed(1001)
val <- rnorm(500)
group <- as.factor(sample(c("A", "B", "C"), 500, replace = T))
out1 <- 30 + val \* 0.3 - as.numeric(group) \* 0.25 + rnorm(500) \* 4
out2 <- 40 - val \* 0.4 + as.numeric(group) \* 0.3 + rnorm(500) \* 4

dat1 <- data.frame(val, group, out1, out2)
dat2 <- data.frame(val, group, out = c(out1, out2), measure = rep(c("out1",
 "out2"), each = 500))

ggplot(dat2, aes(x = val, y = out, color = group)) + geom\_point() + facet\_wrap(~measure)



## this just does things 'in parallel'
lm1 <- lm(cbind(out1, out2) ~ val + group)
summary(lm1)

Response out1 :

Call:
lm(formula = out1 ~ val + group)

Residuals:
 Min 1Q Median 3Q Max
-11.2953 -2.6617 -0.0582 2.6439 11.4131

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 30.2180 0.2995 100.905 <2e-16 \*\*\*
val 0.3136 0.1742 1.800 0.0725 .
groupB -0.7703 0.4207 -1.831 0.0677 .
groupC -0.9218 0.4233 -2.177 0.0299 \*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.843 on 496 degrees of freedom
Multiple R-squared: 0.01653, Adjusted R-squared: 0.01058
F-statistic: 2.778 on 3 and 496 DF, p-value: 0.04069

Response out2 :

Call:
lm(formula = out2 ~ val + group)

Residuals:
 Min 1Q Median 3Q Max
-10.633 -2.857 -0.175 2.714 13.164

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 40.1447 0.3226 124.459 <2e-16 \*\*\*
val -0.3055 0.1877 -1.628 0.1041
groupB 0.5854 0.4531 1.292 0.1970
groupC 0.9849 0.4560 2.160 0.0312 \*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.14 on 496 degrees of freedom
Multiple R-squared: 0.01418, Adjusted R-squared: 0.00822
F-statistic: 2.379 on 3 and 496 DF, p-value: 0.06902

Here, it really just runs a regression twice, once on each outcome variable. But if we give this lm to anova, it will do a MANOVA

library(car)
anova(lm1)

Analysis of Variance Table

 Df Pillai approx F num Df den Df Pr(>F)
(Intercept) 1 0.99384 39930 2 495 < 2e-16 \*\*\*
val 1 0.01029 3 2 495 0.07733 .
group 2 0.01965 2 4 992 0.04381 \*
Residuals 496
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

However, this uses a “Type-I” MANOVA. Because we have multiple predictors, maybe we want a type-II MANOVA, which car::Anova and car::Manova do.

## these two produce the same results:
Anova(lm1)

Type II MANOVA Tests: Pillai test statistic
 Df test stat approx F num Df den Df Pr(>F)
val 1 0.011447 2.8659 2 495 0.05788 .
group 2 0.019655 2.4614 4 992 0.04381 \*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Manova(lm1)

Type II MANOVA Tests: Pillai test statistic
 Df test stat approx F num Df den Df Pr(>F)
val 1 0.011447 2.8659 2 495 0.05788 .
group 2 0.019655 2.4614 4 992 0.04381 \*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Interpreting the MANOVA Statistics

When computing a standard ANOVA, we are dividing up our variance into pools accounted for by different predictors and combinations of predictors. In MANOVA, instead of variance, we have a covariance matrix that we are trying to do the same thing with. For $N$ outcomes, this involves $N$ variance terms, plus $N\*(N−1)/2$ covariance (correlation) terms. The test statistics work by computing eigenvalues of this covariance matrix, and denote $λ\_{i}$ as the $ith$ eigenvalue. Most test statistics are function of these $λ$ values, and there are a number of ways to combine these, and each has its own null-hypothesis distribution that is generally related to the F test.

The following is from: <https://en.wikiversity.org/wiki/Advanced_ANOVA/MANOVA>

Choose from among these multivariate test statistics to assess whether there are statistically significant differences across the levels of the IV(s) for a linear combination of DVs. In general Wilks’ $λ$ is recommended unless there are problems with small N, unequal ns, violations of assumptions, etc. in which case Pillai’s trace is more robust:

## Roy’s greatest characteristic root

* Tests for differences on only the first discriminant function
* Most appropriate when DVs are strongly interrelated on a single dimension
* Highly sensitive to violation of assumptions - most powerful when all assumptions are met

## Wilks’ lambda ($Λ$)

* Most commonly used statistic for overall significance
* Considers differences over all the characteristic roots
* The smaller the value of Wilks’ lambda, the larger the between-groups dispersion

## Hotelling’s trace

* Considers differences over all the characteristic roots

## Pillai’s criterion

* Considers differences over all the characteristic roots
* More robust than Wilks’; should be used when sample size decreases, unequal cell sizes or homogeneity of covariances is violated

Typically, packages will report Pillai’s criterion or Pillai’s trace. It is the sum of $(λ)/(1−λ)$, which should remind you of the effect size $f^{2}$ used in calculating power, since the eigenvalue is a measure of the proportion of variance. Hotelling’s Trace (or Hotelling-Lawley’s Trace) is the sum of the eigenvalues; Wilks $λ$ is the sum of $1/(1+λ\_{i})$. Roy’s greatest root is the largest $λ$; essentially considering just the first principle component.

Each of these tests are available for specifying in Anova, but “Pillai” is the default:

Manova(lm1, test.statistic = "Wilks")

Type II MANOVA Tests: Wilks test statistic
 Df test stat approx F num Df den Df Pr(>F)
val 1 0.98855 2.8659 2 495 0.05788 .
group 2 0.98035 2.4678 4 990 0.04335 \*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Manova(lm1, test.statistic = "Hotelling-Lawley")

Type II MANOVA Tests: Hotelling-Lawley test statistic
 Df test stat approx F num Df den Df Pr(>F)
val 1 0.011579 2.8659 2 495 0.05788 .
group 2 0.020034 2.4743 4 988 0.04289 \*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Manova(lm1, test.statistic = "Roy")

Type II MANOVA Tests: Roy test statistic
 Df test stat approx F num Df den Df Pr(>F)
val 1 0.011579 2.8659 2 495 0.057875 .
group 2 0.019673 4.8788 2 496 0.007975 \*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The different statistics give different interpretations. Roy’s greatest root eessentially asks whether there is a difference on the primary dimension of shared variance of the dependent variables. The others consider all the dimensions of shared variance, but combine these differently.

Ultimately, the MANOVA often hides more than it reveals. Many statisticians have recommended you avoid using it unless you really need to, but it can come in handy, and understanding is important for evaluating other research.

## Conceptual relationship with PCA.

Conceptually, MANOVA is sort of like doing something like the following, which performs a regression on dimensions of a principal components analysis. If you are trying to do something like this, it would usually be better to use the MANOVA directly rather than trying to figure out something ad hoc like predicting eigen vectors ‘by hand’:

E <- eigen(cor((dat1[, 3:4])))
print(E)

eigen() decomposition
$values
[1] 1.0402318 0.9597682

$vectors
 [,1] [,2]
[1,] -0.7071068 -0.7071068
[2,] 0.7071068 -0.7071068

p1 <- as.matrix(dat1[, 3:4]) %\*% as.matrix(E$vectors[, 1])
p2 <- as.matrix(dat1[, 3:4]) %\*% as.matrix(E$vectors[, 2])

m1 <- (lm(p1 ~ val + group))
summary(m1)

Call:
lm(formula = p1 ~ val + group)

Residuals:
 Min 1Q Median 3Q Max
-12.7285 -2.6673 -0.0839 2.6256 12.3623

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 7.0193 0.3152 22.270 < 2e-16 \*\*\*
val -0.4378 0.1834 -2.387 0.01734 \*
groupB 0.9586 0.4428 2.165 0.03088 \*
groupC 1.3482 0.4455 3.026 0.00261 \*\*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.045 on 496 degrees of freedom
Multiple R-squared: 0.02903, Adjusted R-squared: 0.02315
F-statistic: 4.942 on 3 and 496 DF, p-value: 0.002168

Anova(m1)

Anova Table (Type II tests)

Response: p1
 Sum Sq Df F value Pr(>F)
val 93.3 1 5.6996 0.017343 \*
group 158.8 2 4.8521 0.008187 \*\*
Residuals 8115.3 496
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

m2 <- lm(p2 ~ val + group)
summary(m2)

Call:
lm(formula = p2 ~ val + group)

Residuals:
 Min 1Q Median 3Q Max
-13.2896 -2.6570 0.1609 2.7184 10.7235

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -49.753962 0.307217 -161.951 <2e-16 \*\*\*
val -0.005704 0.178737 -0.032 0.975
groupB 0.130745 0.431599 0.303 0.762
groupC -0.044651 0.434274 -0.103 0.918
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.943 on 496 degrees of freedom
Multiple R-squared: 0.000362, Adjusted R-squared: -0.005684
F-statistic: 0.05987 on 3 and 496 DF, p-value: 0.9808

Anova(m2)

Anova Table (Type II tests)

Response: p2
 Sum Sq Df F value Pr(>F)
val 0.0 1 0.0010 0.9746
group 2.8 2 0.0897 0.9142
Residuals 7710.3 496

Notice that for two groups, the eigenvectors are essentially the mean and the difference between the observations. As the number of outcomes increases, these would be less constrained. If you look at the F tests for each predictor, you will see that the values are very similar to those produced in the PCA version (either on average, or max values, etc.). But this means that, depending on the test statistic, a MANOVA might also be considering strange and non-intuitive combinations of variables. Importantly, the group of dependent measures do not need to be correlated. Consider the following dependent variables. Here, out1 and out2 are independent, but group is related to the difference between them. Group is thus also indirectly related to each one, because the difference is larger when one is large or when the other is small (or both). Group2 and out3 are unrelated to the others.

set.seed(150)
yout1 <- rnorm(150)
yout2 <- rnorm(150)
yout3 <- rnorm(150)
delta <- ((yout1 - mean(yout1)) - (yout2 - mean(yout2))) + rnorm(150) \* 5
group1 <- floor(rank(delta)/50) + 1
group2 <- sample(1:5, replace = T, size = 150)
out <- cbind(yout1, yout2, yout3)
summary(lm(out ~ group1))

Response yout1 :

Call:
lm(formula = yout1 ~ group1)

Residuals:
 Min 1Q Median 3Q Max
-2.41516 -0.71119 0.00642 0.75209 2.45372

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.21468 0.21687 -0.99 0.324
group1 0.14901 0.09933 1.50 0.136

Residual standard error: 1.008 on 148 degrees of freedom
Multiple R-squared: 0.01498, Adjusted R-squared: 0.008322
F-statistic: 2.25 on 1 and 148 DF, p-value: 0.1357

Response yout2 :

Call:
lm(formula = yout2 ~ group1)

Residuals:
 Min 1Q Median 3Q Max
-2.80585 -0.52989 -0.04297 0.60846 2.24696

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.40930 0.20202 2.026 0.0446 \*
group1 -0.23560 0.09253 -2.546 0.0119 \*
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9388 on 148 degrees of freedom
Multiple R-squared: 0.04197, Adjusted R-squared: 0.0355
F-statistic: 6.484 on 1 and 148 DF, p-value: 0.01191

Response yout3 :

Call:
lm(formula = yout3 ~ group1)

Residuals:
 Min 1Q Median 3Q Max
-2.99510 -0.70386 -0.03446 0.70520 2.32555

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.05673 0.21535 0.263 0.793
group1 -0.03119 0.09864 -0.316 0.752

Residual standard error: 1.001 on 148 degrees of freedom
Multiple R-squared: 0.0006752, Adjusted R-squared: -0.006077
F-statistic: 0.09999 on 1 and 148 DF, p-value: 0.7523

Manova(lm(out ~ group1 + group2), test.statistic = "Roy")

Type II MANOVA Tests: Roy test statistic
 Df test stat approx F num Df den Df Pr(>F)
group1 1 0.057303 2.7696 3 145 0.04385 \*
group2 1 0.021894 1.0582 3 145 0.36890
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Manova(lm(out ~ group1 + group2), test.statistic = "Pillai")

Type II MANOVA Tests: Pillai test statistic
 Df test stat approx F num Df den Df Pr(>F)
group1 1 0.054197 2.7696 3 145 0.04385 \*
group2 1 0.021425 1.0582 3 145 0.36890
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Manova(lm(out ~ group1 + group2), test.statistic = "Wilks")

Type II MANOVA Tests: Wilks test statistic
 Df test stat approx F num Df den Df Pr(>F)
group1 1 0.94580 2.7696 3 145 0.04385 \*
group2 1 0.97858 1.0582 3 145 0.36890
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Relation to Repeated Measures ANOVA

This all seems like it is a bit like a repeated-measures ANOVA. By putting the data in ‘long’ format, we can just combine out1 and out2 into a single DV, then predict the mean difference between the two with a categorical value:

dat2 <- data.frame(val, group, out = c(out1, out2), measure = rep(c("out1",
 "out2"), 500))

model.rm <- lm(out ~ val + group + measure, data = dat2)
summary(model.rm)

Call:
lm(formula = out ~ val + group + measure, data = dat2)

Residuals:
 Min 1Q Median 3Q Max
-17.3246 -5.5151 -0.1335 5.4693 18.8956

Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) 35.100770 0.449277 78.127 <2e-16 \*\*\*
val 0.001362 0.219079 0.006 0.995
groupB -0.074609 0.531435 -0.140 0.888
groupC 0.041278 0.532747 0.077 0.938
measureout2 0.142642 0.434432 0.328 0.743
---
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.83 on 995 degrees of freedom
Multiple R-squared: 0.0001685, Adjusted R-squared: -0.003851
F-statistic: 0.04192 on 4 and 995 DF, p-value: 0.9967

This might be OK is some situations. But this is assuming that the two measures differ just by a constant. What if they have different scales, different variances, different directional relationships? Just like the t-test above, if we had a theory that could be used to combine these, it might be a better approach, but in this case, we are restricted in the inferences we can make. Notice that for each sub-model, the effects are in different directions, and so cancel out one another if we do it this way. We could sort of do this by predicting all the interactions too, but this begins to get very complicated such that a MANOVA might be preferred.

## Relationship to Logistic Regression, PCA, LDA, and Classification

As we can see, although MANOVA seems like it is just a simple extension of ANOVA, it relates to many multi-variate concepts. For Hotelling’s $T^{2}$ version, we are attempting to find whether there is a difference between two groups on multiple measures. We could easily turn this around; predicting group membership based on those multiple `predictors’. This is what methods such as logistic regression and linear discriminant analysis do. As we extend to multiple categories in MANOVA, this becomes similar to what some machine classification schemes permit–classifying categories based on multiple observed ‘features’. Finally, we can understand MANOVA as sort of doing regressions on each principle component, and determining a reasonable means of combining those regressions at the end of the day.

# Canonical Correlation Analysis

MANOVA extends ANOVA/regression and allows multiple predictors and multiple outcome variables. A lesser-known alternative is Canonical Correlation Analysis (CCA), which tries to establish the cross-correlation between two sets of variables, and does so by establishing a dimensionality of the relationship. The dimensionality is essentially a PCA on the cross-correlation matrix, and then an regression for each group of variables onto the PCA dimensions.

## Packages and resources

* <https://stats.idre.ucla.edu/r/dae/canonical-correlation-analysis/>
* cancor function in stats package
* CCA package
* candisc package

library(CCA)
matcor(out, cbind(group1, group2))

$Xcor
 yout1 yout2 yout3
yout1 1.00000000 -0.06856134 -0.02984807
yout2 -0.06856134 1.00000000 -0.03009671
yout3 -0.02984807 -0.03009671 1.00000000

$Ycor
 group1 group2
group1 1.00000000 0.02128681
group2 0.02128681 1.00000000

$XYcor
 yout1 yout2 yout3 group1 group2
yout1 1.00000000 -0.06856134 -0.02984807 0.12238444 0.07803933
yout2 -0.06856134 1.00000000 -0.03009671 -0.20486423 -0.10699825
yout3 -0.02984807 -0.03009671 1.00000000 -0.02598397 0.07406789
group1 0.12238444 -0.20486423 -0.02598397 1.00000000 0.02128681
group2 0.07803933 -0.10699825 0.07406789 0.02128681 1.00000000

img.matcor(matcor(out, cbind(group1, group2)))



img.matcor(matcor(out, cbind(group1, group2)), type = 2)



cc(out, cbind(group1, group2))

$cor
[1] 0.26263241 0.07937333

$names
$names$Xnames
[1] "yout1" "yout2" "yout3"

$names$Ynames
[1] "group1" "group2"

$names$ind.names
NULL

$xcoef
 [,1] [,2]
yout1 -0.48460186 -0.1331493
yout2 0.87578518 -0.1470000
yout3 -0.03886284 -0.9927121

$ycoef
 [,1] [,2]
group1 -1.043720 0.5989572
group2 -0.337832 -0.6188467

$scores
$scores$xscores
 [,1] [,2]
 [1,] 1.40401546 -0.6560067344
 [2,] -0.39846124 0.5756319377
 [3,] -0.56810871 0.9189118013
 [4,] -0.07026177 -0.7831775998
 [5,] -0.04980326 1.3627982515
 [6,] -0.67802870 1.1111871177
 [7,] -0.14164189 -0.4383861289
 [8,] 1.66824470 -0.0246411422
 [9,] -0.12530280 -0.6128865008
 [10,] -0.29835922 -0.3348299896
 [11,] 0.46091832 -0.1758593157
 [12,] 0.80330963 0.3810364259
 [13,] 1.45985150 -1.5112320423
 [14,] 0.39275927 1.9624447423
 [15,] -0.29570131 0.2910248132
 [16,] -1.31224844 2.1406132258
 [17,] 0.35552775 1.2307228392
 [18,] -1.23657987 -1.3328103715
 [19,] -1.47315475 0.4556874421
 [20,] 1.04670887 -0.3184751907
 [21,] 0.41765841 -2.0466490483
 [22,] -0.92246469 -0.4598266576
 [23,] -0.43184247 -0.4478789441
 [24,] 0.53106491 -0.1757817904
 [25,] -1.50927235 -0.7759573649
 [26,] 0.05420050 -0.0279553714
 [27,] -0.44267714 -1.4419239979
 [28,] 0.66196527 1.3395395720
 [29,] -0.88008075 -1.7262296778
 [30,] -0.11863736 1.2358098517
 [31,] -1.74129112 0.3060767911
 [32,] 0.88694218 0.0132243314
 [33,] -0.60949351 -0.8203385382
 [34,] -0.28210436 -0.0905208579
 [35,] -0.40356328 0.4048876953
 [36,] 0.83245610 -1.6660496589
 [37,] 0.78948935 1.6584631433
 [ reached getOption("max.print") -- omitted 113 rows ]

$scores$yscores
 [,1] [,2]
 [1,] 1.09387324 -0.55730296
 [2,] 0.75604121 -1.17614966
 [3,] -0.99356688 0.64061144
 [4,] -0.65573486 1.25945813
 [5,] -1.33139891 0.02176474
 [6,] -0.28767885 -0.57719246
 [7,] -0.31790283 1.87830483
 [8,] -0.28767885 -0.57719246
 [9,] -1.33139891 0.02176474
 [10,] 1.43170526 0.06154373
 [11,] -0.62551087 -1.19603915
 [12,] -0.28767885 -0.57719246
 [13,] 0.75604121 -1.17614966
 [14,] 0.75604121 -1.17614966
 [15,] 1.43170526 0.06154373
 [16,] -0.65573486 1.25945813
 [17,] -0.31790283 1.87830483
 [18,] -1.33139891 0.02176474
 [19,] -0.65573486 1.25945813
 [20,] 0.05015318 0.04165424
 [21,] 0.38798520 0.66050093
 [22,] -1.66923093 -0.59708196
 [23,] 1.43170526 0.06154373
 [24,] 1.76953729 0.68039043
 [25,] -1.69945492 1.85841533
 [26,] -0.62551087 -1.19603915
 [27,] 1.09387324 -0.55730296
 [28,] -0.31790283 1.87830483
 [29,] 0.72581723 1.27934763
 [30,] -0.99356688 0.64061144
 [31,] 0.75604121 -1.17614966
 [32,] 1.09387324 -0.55730296
 [33,] 0.05015318 0.04165424
 [34,] -0.65573486 1.25945813
 [35,] 1.76953729 0.68039043
 [36,] -0.62551087 -1.19603915
 [37,] 0.72581723 1.27934763
 [ reached getOption("max.print") -- omitted 113 rows ]

$scores$corr.X.xscores
 [,1] [,2]
yout1 -0.54667055 -0.09555281
yout2 0.87196346 -0.10147007
yout3 -0.04933294 -0.98222260

$scores$corr.Y.xscores
 [,1] [,2]
group1 -0.2305201 0.03803232
group2 -0.1307208 -0.06884290

$scores$corr.X.yscores
 [,1] [,2]
yout1 -0.14357340 -0.007584344
yout2 0.22900586 -0.008054017
yout3 -0.01295643 -0.077962274

$scores$corr.Y.yscores
 [,1] [,2]
group1 -0.8777290 0.4791574
group2 -0.4977329 -0.8673304

Here, we see the canonical correlation suggests two dimensions. The first dimension correlates negatively with out1, positively with out2, and not with out3. The second dimension correlates with out3 only. On the other side, group 1 and group 2 correlate (negatively) with dimension 1, and group1 is positively related to dimension 2 but group2 is negatively related to dimension 2.