Analysis of variance

S520

November 10, 2016

Reference: Trosset chapter 12

Motivation: Special case (equal sample sizes)

We've done one- and two-sample tests; now let's generalize to tests where we have k independent samples. (Of course, if k is 1 or 2, we'll be better off using the methods we've already learned.) The null hypothesis we'll test will be:

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k$$

The alternative will be that at least one of the population means is different.

Let's generate some fictitious data: three independent samples, each of size 50, each from the same Normal$(50, 15^2)$ distribution. But before we do that, let me set a random seed to make sure I get the same answers every time I run this:

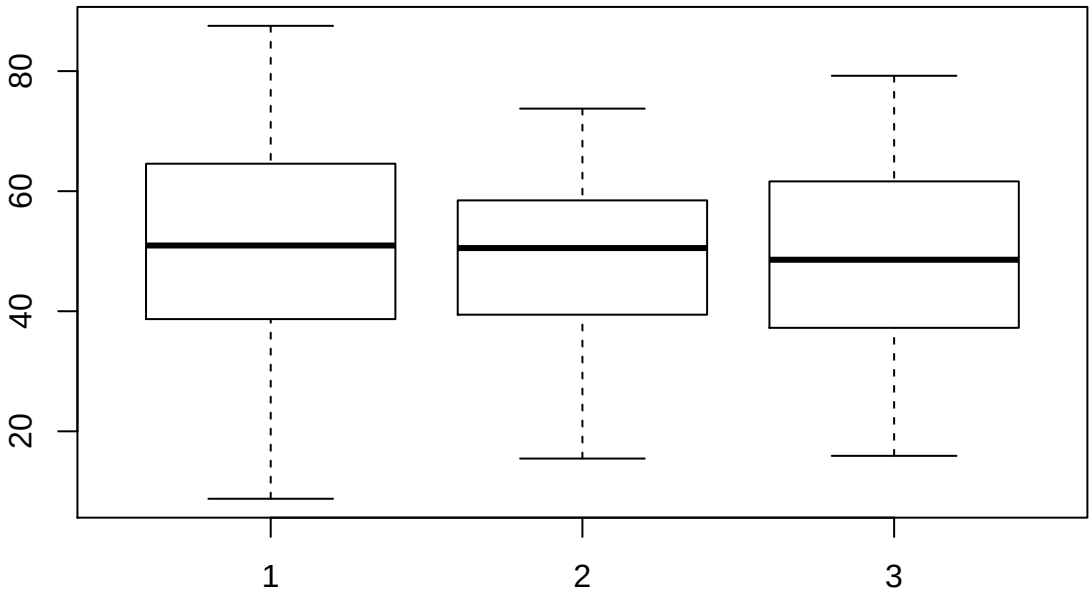
```
set.seed(123456)
```

Okay, now let's generate three random sample using `rnorm()`.

```
x1 = rnorm(50, mean=50, sd=15)
x2 = rnorm(50, mean=50, sd=15)
x3 = rnorm(50, mean=50, sd=15)
```

Now let's pretend we didn't know how the data was generated. Just by looking at the data, does it look like all the population means could be equal?

```
boxplot(x1, x2, x3)
```



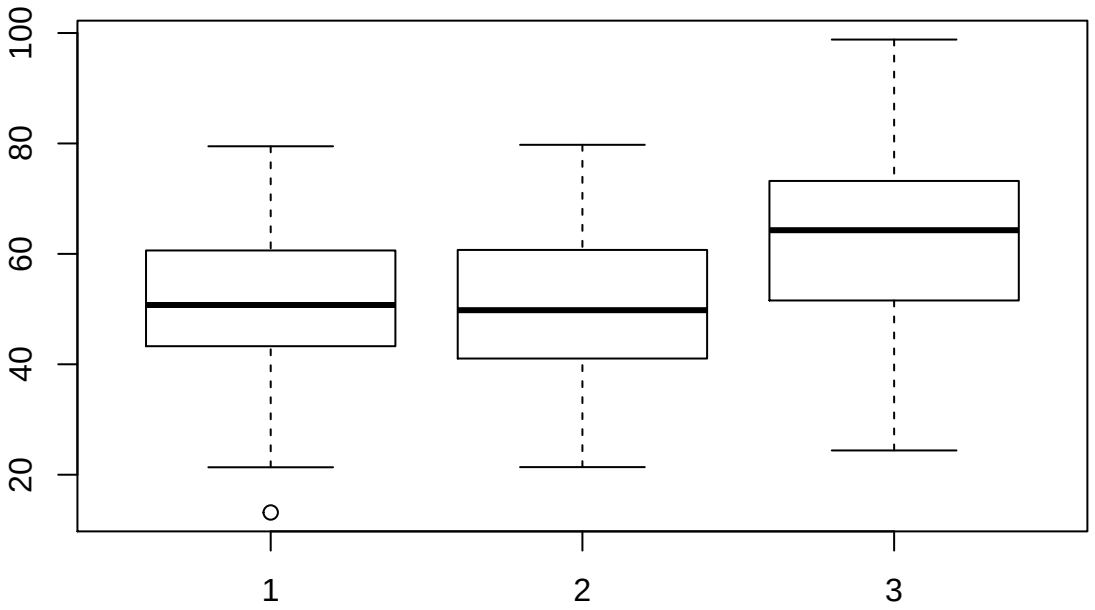
It's quite plausible.

Here's a second fictitious data set.

```
y1 = rnorm(50, mean=50, sd=15)
y2 = rnorm(50, mean=50, sd=15)
y3 = rnorm(50, mean=60, sd=15)
```

Now forget how the data was generated. Do all three samples come from populations with the same mean?

```
boxplot(y1, y2, y3)
```



It doesn't look like it. But we need a more rigorous way to test the hypothesis of equal population means when we have samples from three or more populations. However, unlike the tow sample case where there was an obvious parameter (the difference in means) to study, it's not so obvious what to do here. Let's ask the father of statistis, Sir Ronald A. Fisher!

FISHER: Okay, I've got an idea. Let's start by making some assumptions:

- The observations are independent;
- All of the populations are normal;
- Homoscedasticity.

YOU: Independence I'm okay with. I'm not absolutely sure about normality.

FISHER: Well, you can check normality holds at least approximately by drawing QQ plots. I'll leave that as an exercise.

YOU: Okay, but what does homoscedasticity mean?

FISHER: We assume all the populations have equal variance.

YOU: Well, why didn't you just say that?

FISHER: It's a perfectly cromulent word. Anyway, what we do next depends on whether or not we know the population variances.

YOU: Do we ever know the population variances?

FISHER: No, so let's ignore that case. Instead, just look at the sample SDs, and as long as they're not way different, we should be okay.

```r
sd(x1)
```

```
## [1] 15.62226
```

```r
sd(x2)
```

```
## [1] 14.14227
```

```r
sd(x3)
```

```
## [1] 15.01794
```

YOU: Is that close enough?

FISHER: Sure. Remember sample SDs will always be a bit different, because they're random. Do you agree that it's quite plausible that all of the population SDs are around 15?

YOU: Sounds about right.

FISHER: Since the sample sizes are all equal, we can get a rigorous "within-sample" estimate of σ^2 by averaging:

```r
(var(x1) + var(x2) + var(x3)) / 3
```

```
## [1] 223.1992
```

YOU: But how does that help us?

FISHER: Aha! Suppose the null hypothesis is true: the population means are equal. What would that suggest about the sample means?

YOU: They should be close to whatever the true population mean is? And close to each other?

FISHER: Right! And to measure how far apart the sample means are from each other, we can take the variance of the sample means.

YOU: Let's see. If I remember chapter 8 correctly, if the population variances are all σ^2, then the variances of the individual sample means should be σ^2/n.

FISHER: So to get a "between-sample" estimate of σ^2, take the (sample) variance of the sample means, and multiply by n:

```r
var(c(mean(x1), mean(x2), mean(x3))) * 50
```

```
## [1] 154.7214
```

YOU: That's a little less than the within-sample estimate.

FISHER: That's good for the null. If the sample means are a bit closer than expected, that's more compatible with the hypothesis of equal population means than if they were far apart.

YOU: Let's see what happens if we do the same thing on the y-samples.

```r
# Within-sample
(var(y1) + var(y2) + var(y3)) / 3
```

```
## [1] 216.7147
```

```r
# Between-sample
var(c(mean(y1), mean(y2), mean(y3))) * 50
```

```
## [1] 2482.149
```

Okay, as I suspected, the between-sample estimate is much bigger. But how do I convince anyone this isn't just luck?

FISHER: Well, you make the between-square estimate divided by the within-square estimate your test statistic. Then you get a genius statistician to work out the distribution of the statistic under the null, which lets you find a P-value.

YOU: And you would be that genius statistician, I suppose?

FISHER: If you insist. Let's call the null distribution an F-distribution...

YOU: F for Fisher, I take it.

FISHER: With 2 and 147 degrees of freedom.

YOU: Wait, there are two different degrees of freedom?

FISHER: Yeah, you need one for the between and one for the within.

YOU: Handwavy explanation?

FISHER: You've got three groups for the between, and you lose one degree of freedom because you need the overall mean. You've 150 observations in total for the within, and you lose three DFs for the three group means. Now when you use this newfangled R program, you need to specify both degrees of freedom. Here's the code for the `x`-samples:

```
within.ms = (var(x1) + var(x2) + var(x3)) / 3
between.ms = var(c(mean(x1), mean(x2), mean(x3))) * 50
F = between.ms / within.ms
1 - pf(F, df1=2, df2=147)
```

```
## [1] 0.5016009
```

YOU: It's always `1 - pf()` because you always want to reject for big values of F?

FISHER: Yes, because it's the big values of F that are evidence against the null and for the alternative. In this case, it's not a small P-value, so the data is compatible with the null.

YOU: So for the y-samples,

```
within.ms = (var(y1) + var(y2) + var(y3)) / 3
between.ms = var(c(mean(y1), mean(y2), mean(y3))) * 50
F = between.ms / within.ms
1 - pf(F, df1=2, df2=147)
```

```
## [1] 2.383875e-05
```

The data isn't compatible with the null, so there's evidence the means are different. Great! Now I understand analysis of variance!

FISHER: Not yet.

YOU: Uh oh.

FISHER: We've only dealt with the easiest case, where all the sample sizes are equal. Now we have to work out what to do if the sample sizes aren't equal.

YOU: Ugh, I'm tired. Just write me down some algebra.

FISHER: Oh fine. Remember you can always look up a rigorous statistics textbook (such as Trosset pp. 309-312) for the full argument.

- Assume that we have k independent samples from normal populations that all have the same variance. (As usual, the larger the samples, the less strict we need to be about the assumptions.)
- Let the sample sizes be $n_1, n_2, \ldots, n_k$.
- Let X_{ij} denote observation j in sample i.
- Let $\bar{X}_{i\cdot}$ be the mean of sample i.

- Let $\bar{X}_{..}$ be the **grand mean**: that is, the mean of all the data put together.
- The **within-groups sum of squares** is

$$SS_W = \sum_{i=1}^{n}\sum_{j=1}^{n_j}(X_{ij} - \bar{X}_{i.})^2 = \sum_{i=1}^{k}(n_i - 1)s_i^2$$

  where s_i^2 is the variance of the ith sample.
- The **between-groups sum of squares** is

$$SS_B = \sum_{i=1}^{k} n_i(\bar{X}_{i.} - \bar{X}_{..})^2$$

- The **total sum of squares** is $SS_T = SS_B + SS_W$. If you check p. 310 of Trosset, there's actually a really elegant form for SS_T...

YOU: Don't care.

FISHER: Righto.

- The **between degrees of freedom** is $k - 1$.
- The **within degrees of freedom** is $N - k$. (Adding these up gives $N - 1$ total degrees of freedom.)
- The **between mean square** is

$$\frac{SS_B}{k-1}$$

- The **within mean square** is

$$\frac{SS_W}{N-k}$$

- The F-statistic is

$$F = \frac{\text{between mean square}}{\text{within mean square}}$$

- Finally, to get the P-value, find the probability that an F random variable with $k-1$ and $N-k$ degrees of freedom gives you a value at least as large as the F you just calculated:

```
1 - pf(F, df1=k-1, df2=N-k)
```

YOU: ...

FISHER: What?

YOU: Is this one of those times where you can get all this stuff in one line of R?

FISHER: It's acutally three lines.

```
x.all = c(x1, x2, x3)
group = factor(c(rep(1, 50), rep(2, 50), rep(3, 50)))
anova(lm(x.all ~ group))
```

```
## Analysis of Variance Table
##
## Response: x.all
##            Df Sum Sq Mean Sq F value Pr(>F)
## group       2    309  154.72  0.6932 0.5016
## Residuals 147  32810  223.20
```

That's an example of an **ANOVA table**, which is the standard way of showing ANOVA results. It wasn't one of my better ideas – side-by-side boxplots or dotplots, for example, are much more informative, but they hadn't been invented bak then. My bad.

YOU: As far as bad ideas go, ANOVA tables aren't are bad as when you argued smoking didn't cause cancer in the Fifties. Anyway, what's this `factor()` function in the R code?

FISHER: `factor()` makes sure you're treating the `group` variable as a set of categories. The labels 1, 2, and 3 are arbitrary, so we don't want to treat them as numbers.

YOU: And the `lm()` part?

FISHER: It'll make sense once you do regression.

YOU: In conclusion, the P-value isn't small, which means the data is compatible with the null hypothesis that all the populations have the same mean. Let's try it on the y-samples:

```
y.all = c(y1, y2, y3)
group = factor(c(rep(1, 50), rep(2, 50), rep(3, 50)))
anova(lm(y.all ~ group))
```

```
## Analysis of Variance Table
##
## Response: y.all
##            Df Sum Sq Mean Sq F value    Pr(>F)
## group       2   4964 2482.15  11.454 2.384e-05 ***
## Residuals 147  31857  216.71
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here the P-value is very small, so the data isn't consistent with equal population means. That is, the population means are different.

FISHER: Yes. (You could also have picked an α in advance and decided to reject or to not reject, but that's Jerzy Neyman's idea, not mine.)

YOU: So the population means are different. The end.

FISHER: Not even close.

YOU: OH COME ON.

FISHER: Remember, a small P-value might tell you the null isn't true, but it doesn't tell you what *is* true. In this case, the population means are different, great, but now you have to actually estimate what they are. You can use graphs, more tests, confidence intervals, whatever.

YOU: I think I've reached the limit of what I can learn from fake data, and am ready to study real data now. Thanks, Sir Ronald.

FISHER: You're welcome. Cigarette?

YOU: No thank you.

Example: Anorexia treatments

The raw data is at:

http://mypage.iu.edu/~mtrosset/StatInfeR/Data/anorexia.dat

Looking at the data, there are six columns. These are (Trosset p. 326):

- Weights of anorexia patients (in pounds) before cognitive behavioral treatment
- Weights of anorexia patients after cognitive behavioral treatment
- Weights of anorexia patients before standard treatment
- Weights of anorexia patients after standard treatment
- Weights of anorexia patients before family therapy

- Weights of anorexia patients after family therapy

We don't have equal numbers for each treatment, so we can't just read the data in as a data frame. I did some jiggling to get the data into a form we can easily read and posted the file as `anorexia.txt` on Canvas. After saving the file in your working directory:

```
anorexia = read.table("anorexia.txt", header=TRUE)
summary(anorexia)
```

```
##      Treatment      Before          After
##  Cognitive:29   Min.   :70.00   Min.   : 71.30
##  Family   :17   1st Qu.:79.60   1st Qu.: 79.33
##  Standard :26   Median :82.30   Median : 84.05
##                 Mean   :82.41   Mean   : 85.17
##                 3rd Qu.:86.00   3rd Qu.: 91.55
##                 Max.   :94.90   Max.   :103.60
```

Now, for each treatment, the "before" and "after" weights are for the same set of individuals, so we can't assume they're independent. As usual, when we have two measurements on each individual, we take differences: in this case, we find the *difference* (change) in weight for each anorexia patient.

```
all.diffs = anorexia$After - anorexia$Before
```

Now split the differences into the three samples:

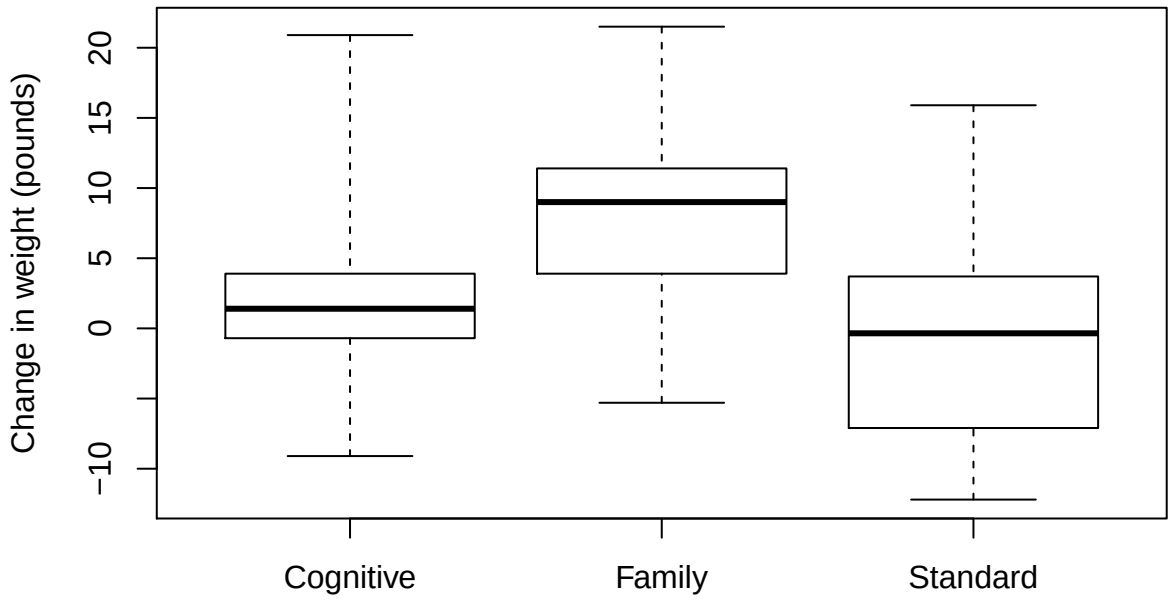
```
cog.diff = all.diffs[anorexia$Treatment=="Cognitive"]
fam.diff = all.diffs[anorexia$Treatment=="Family"]
std.diff = all.diffs[anorexia$Treatment=="Standard"]
```

Write down the sizes of each sample, plus the total sample size:

```
n1 = length(cog.diff)
n2 = length(fam.diff)
n3 = length(std.diff)
N = n1 + n2 + n3
```

Draw a well-labeled graph:

```
boxplot(cog.diff, fam.diff, std.diff, range=0,
    names=c("Cognitive","Family","Standard"),
    ylab="Change in weight (pounds)")
```



Is it plausible that the population standard deviations are equal? Let's check homoscedasticity:

```
sd(cog.diff)
```

```
## [1] 7.308504
```

```
sd(fam.diff)
```

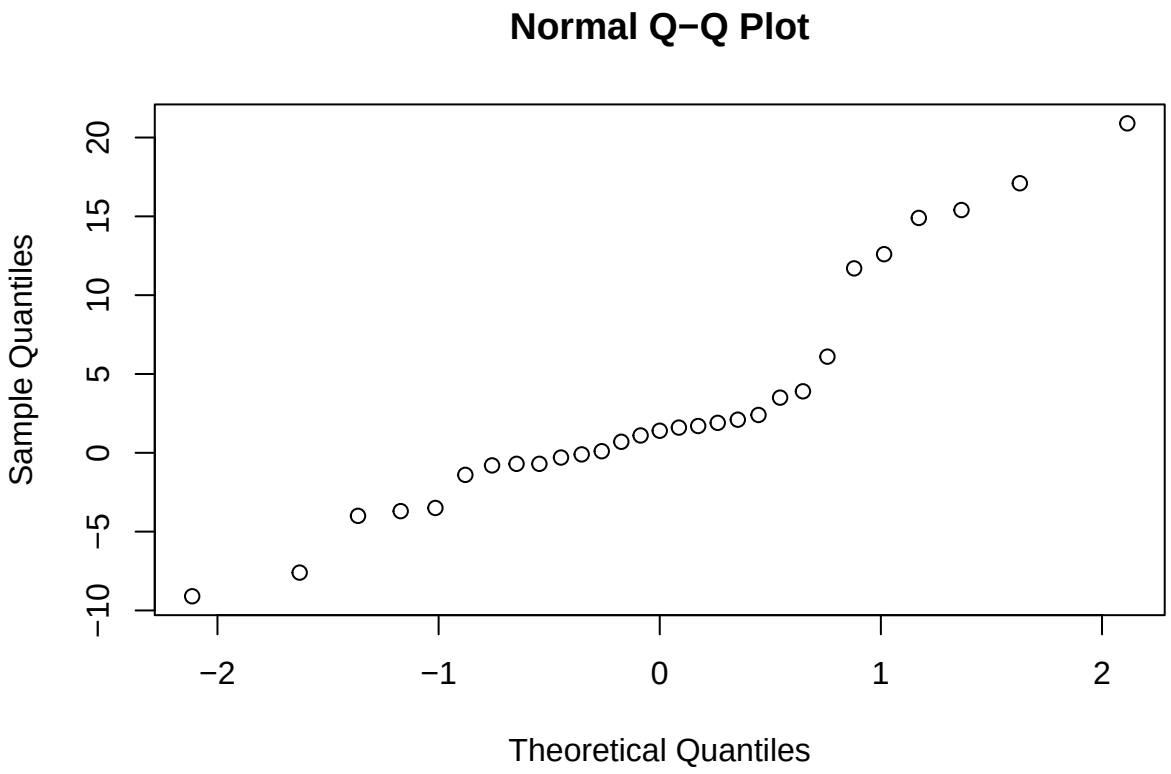
```
## [1] 7.157421
```

```
sd(std.diff)
```

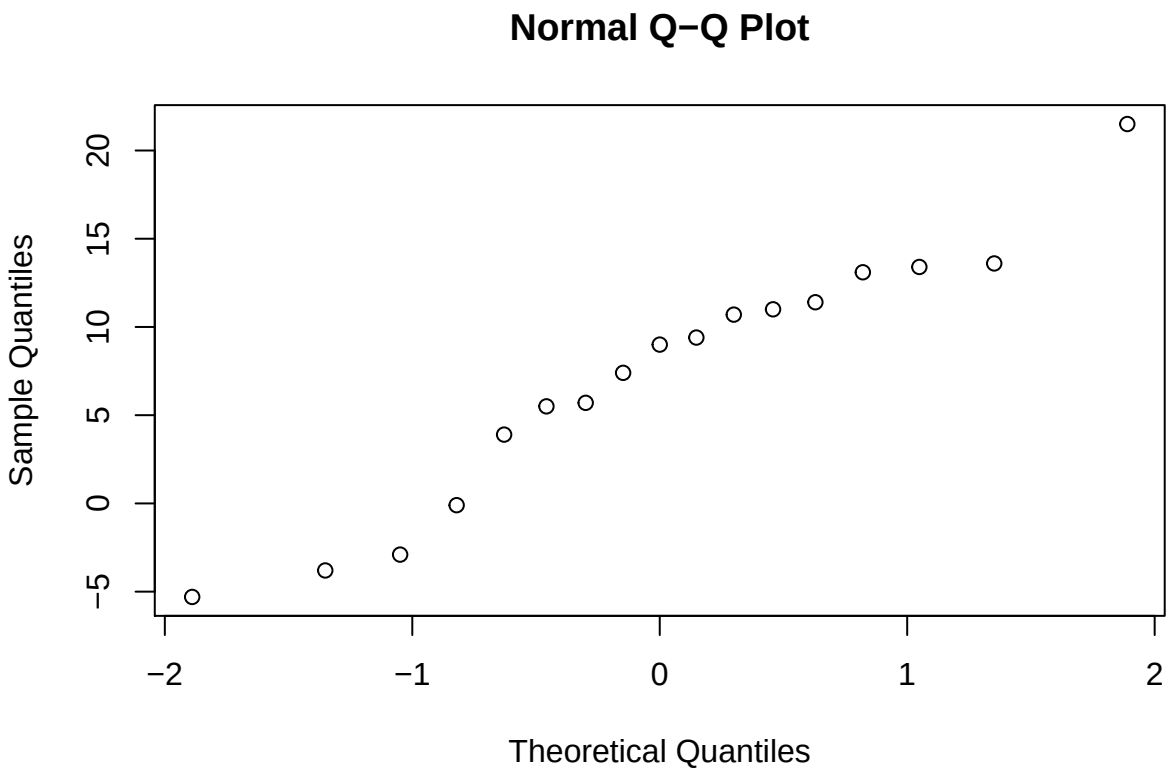
```
## [1] 7.988705
```

The sample SDs are all close, so that's not a problem. Now check normality:

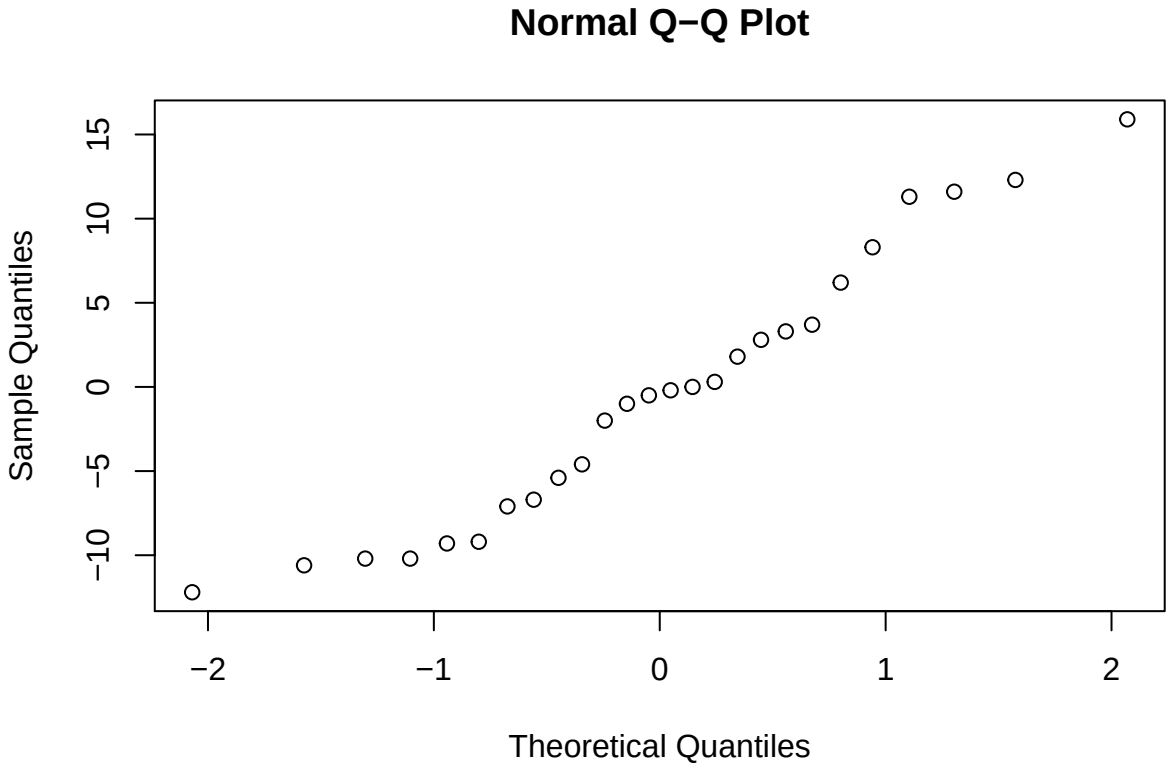
```
qqnorm(cog.diff)
```




```
qqnorm(fam.diff)
```




```
qqnorm(std.diff)
```



This is a bit more worrying – the QQ plot for the cognitive group, for example, doesn't look especially straight. It would be best to get more data, but in the meantime we'll carry on and hope nothing bad happens. Find the group means and the grand mean:

```
cog.mean = mean(cog.diff)
fam.mean = mean(fam.diff)
std.mean = mean(std.diff)
grand.mean = mean(all.diffs)
```

Following Trosset pp. 310-312, we find the total sum-of-squares and degrees of freedom:

```
SST = sum( (all.diffs-grand.mean)^2 )
total.df = N - 1
```

Now find the between sum-of-squares and mean-square:

```
SSB = n1*(cog.mean-grand.mean)^2 +
    n2*(fam.mean-grand.mean)^2 +
    n3*(std.mean-grand.mean)^2
between.df = 2
between.meansquare = SSB/2
```

And the within sum-of-squares and mean-square:

```
SSW = sum( (cog.diff-cog.mean)^2 ) +
    sum( (fam.diff-fam.mean)^2 ) +
    sum( (std.diff-std.mean)^2 )
SSW
```

```
## [1] 3910.742
```

```
within.df = N - 3
within.meansquare = SSW/within.df
```

Note that we could have found SS_W a different way:

```
SSW = (n1-1)*var(cog.diff) +
  (n2-1)*var(fam.diff) +
  (n3-1)*var(std.diff)
SSW
```

```
## [1] 3910.742
```

Check that the following are equal:

```
SST
```

```
## [1] 4525.386
```

```
SSB + SSW
```

```
## [1] 4525.386
```

Now, is the between mean-square close to the within mean-square? Is it much bigger?

```
between.meansquare
```

```
## [1] 307.3218
```

```
within.meansquare
```

```
## [1] 56.67743
```

To formalize this idea, we do the F-test.

```
F = between.meansquare/within.meansquare
1 - pf(F, df1=between.df, df2=within.df)
```

```
## [1] 0.006498653
```

Here's the shortcut:

```
anova(lm(all.diffs ~ anorexia$Treatment))
```

```
## Analysis of Variance Table
##
## Response: all.diffs
##                     Df Sum Sq Mean Sq F value   Pr(>F)
## anorexia$Treatment  2  614.6 307.322  5.4223 0.006499 **
## Residuals          69 3910.7  56.677
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The small P-value means we have evidence against the null hypothesis. So it appears (from the data we have) that the treatments do *not* have the same average effect.

This begs the question: How do the treatments differ? We'll return to this question in the next set of notes.

Extra for experts: If you didn't like the normality assumption

If you didn't like the normality assumption, we can find the P-value in a way that doesn't require knowing the form of the distribution:

1. Calculate the F-statistic from the observed data in the usual way, but don't find the P-value.
2. Shuffle all the data randomly. Then take the first 29 numbers of the shuffled data to be "cognitive," the next 17 to be "family," and the last 26 to be "standard."
3. Calculate the F-statistic for this random data.

4. Repeat steps 1 and 2 a hundred thousand times or so.
5. Your P-value will be the proportion of the random F-statistics that are at least as big as the observed F-statistic.

This is called a **permutation test**. If this seems like a good idea, take STAT-S 625, Nonparametric Statistics.

```
F.random = function(){
  random.diffs = sample(all.diffs)
  cog.random = random.diffs[1:29]
  fam.random = random.diffs[30:46]
  std.random = random.diffs[47:72]
  SSW.random = (n1-1)*var(cog.random) +
    (n2-1)*var(fam.random) +
    (n3-1)*var(std.random)
  SSB.random = SST - SSW.random
 return((SSB.random/between.df) / (SSW.random/within.df))
}
F.list = replicate(100000, F.random())
mean(F.list >= F)
```

```
## [1] 0.00695
```