Assignment 5

Instructions: Make an R Notebook and in it answer the questions below. When you are done, hand in on Quercus the *output* from Previewing (or Knitting) your Notebook, probably an `html` or `pdf` file. An `html` file is easier for the grader to deal with. Do *not* hand in the Notebook itself. You want to show that you can (i) write code that will answer the questions, (ii) run that code and get some sensible output, (iii) write some words that show you know what is going on and that reflect your conclusions about the data. Your goal is to convince the grader that you *understand* what you are doing: not only doing the right thing, but making it clear that you know *why* it's the right thing.

Do *not* expect to get help on this assignment. The purpose of the assignments is for you to see how much *you* have understood. You will find that you also learn something from grappling with the assignments. The time to get help is after you watch the lectures and work through the problems from PASIAS, via tutorial and the discussion board, that is *before* you start work on the assignment. The only reasons to contact the instructor while working on an assignment are to report (i) something missing like a data file that cannot possibly be read, (ii) something *beyond your control* that makes it impossible to finish the assignment in time after you have started it.

There is a time limit on this assignment (you will see Quercus counting down the time remaining).

1. According to the Mayo Clinic, dandruff is "a common condition that causes the skin on the scalp to flake. It isn't contagious or serious. But it can be embarrassing and difficult to treat." Shampoos often claim to be effective in treating dandruff. In a study, four shampoos were compared:

   - `PyrI`: 1% pyrithione zinc shampoo
   - `PyrII`: the same as `PyrI` but with instructions to shampoo two times at each wash. The labels for these are `Pyr` with a Roman numeral I or II attached.
   - `Keto`: 2% ketoconazole shampoo
   - `Placebo`: a placebo shampoo

   Each subject was randomly assigned to a shampoo. After six weeks of treatment, eight sections of the scalp were examined for each subject. Each section of the scalp was given a score that measured the amount of flaking on a scale of 0-10, less flaking being better. The response variable, called `Flaking`, was the sum of these eight scores, and is a whole number for each subject.

   The data are in http://ritsokiguess.site/STAC32/dandruff.txt, with the data values separated by *tabs*.

   Your task is to write a report on your analysis of this data set, and to make a recommendation for the best shampoo(s) out of the four studied here. The target audience for your report is the principal investigator of the study described above, who knows a lot about shampoo, but not so much about statistics. (They took a course some time ago that covered the material you've seen in this course so far, at about the level of STAB22 or STA 220.) Some things you might want to consider, in no particular order (you need to think about where and if to include these things):

   - an Introduction, written in your own words as much as possible
   - a Conclusion that summarizes what you found
   - a suitable and complete piece of statistical inference
   - a numerical summary of the data
   - graph(s) of the data
   - an assessment of the assumptions of your analysis
   - citation of external sources

- anything else that you can make the case for including

In R Markdown (the text of an R Notebook), you can use `##` to make a heading (you can experiment with more or fewer `#` symbols).

Your aim is to produce a report, suitable for the intended audience, with all the important elements and no irrelevant ones, that is well-written and easy to follow. There is credit for good writing. For this report, you should include your code in with your report. (In a real report, you would probably show the output and not the code, but we are interested in your code here as well.)

Solution:

My report follows. Extra comments are in footnotes (at the end) or the multitudinous Extras after.

A comparison of four shampoos in treating dandruff

Introduction

Shampoos are often claimed to be effective at treating dandruff. In a study, the dandruff-treating properties of four shampoos were compared. These four shampoos were, as referred to in the dataset:

- `PyrI`: 1% pyrithione zinc shampoo
- `PyrII`: as `PyrI` but with instructions to shampoo two times at each wash.[1]
- `Keto`: 2% ketoconazole shampoo
- `Placebo`: a placebo shampoo

Each of the experimental subjects was randomly given one of the shampoos. After using their shampoo for six weeks, eight sections of the subject's scalp were examined for each subject. Each section of the scalp was given a score that measured the amount of flaking on a scale of 0-10. The response variable, called `Flaking`, was the sum of these eight scores, and is a whole number for each subject. A smaller value of `Flaking` indicates less dandruff.[2]

Our aim is to see which shampoo or shampoos are most effective at treating dandruff, that is, have the smallest value of `Flaking` on average.

Exploratory analysis

We begin by reading in the data:

```
my_url <- "http://ritsokiguess.site/STAC32/dandruff.txt"
dandruff <- read_tsv(my_url)
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   OBS = col_double(),
##   Treatment = col_character(),
##   GroupNum = col_double(),
##   Flaking = col_double()
## )
```

```
dandruff
```

```
## # A tibble: 355 x 4
##      OBS Treatment GroupNum Flaking
##    <dbl> <chr>        <dbl>   <dbl>
```

```
##  1     1 PyrI             1      17
##  2     2 PyrI             1      16
##  3     3 PyrI             1      18
##  4     4 PyrI             1      17
##  5     5 PyrI             1      18
##  6     6 PyrI             1      16
##  7     7 PyrI             1      17
##  8     8 PyrI             1      20
##  9     9 PyrI             1      17
## 10    10 PyrI             1      17
## # ... with 345 more rows
```

355 subjects took part in the study altogether. The shampoo used is indicated in the `Treatment` column. The remaining columns `OBS` and `GroupNum` will not be used in this analysis.

Numerical summaries of the data are as shown:

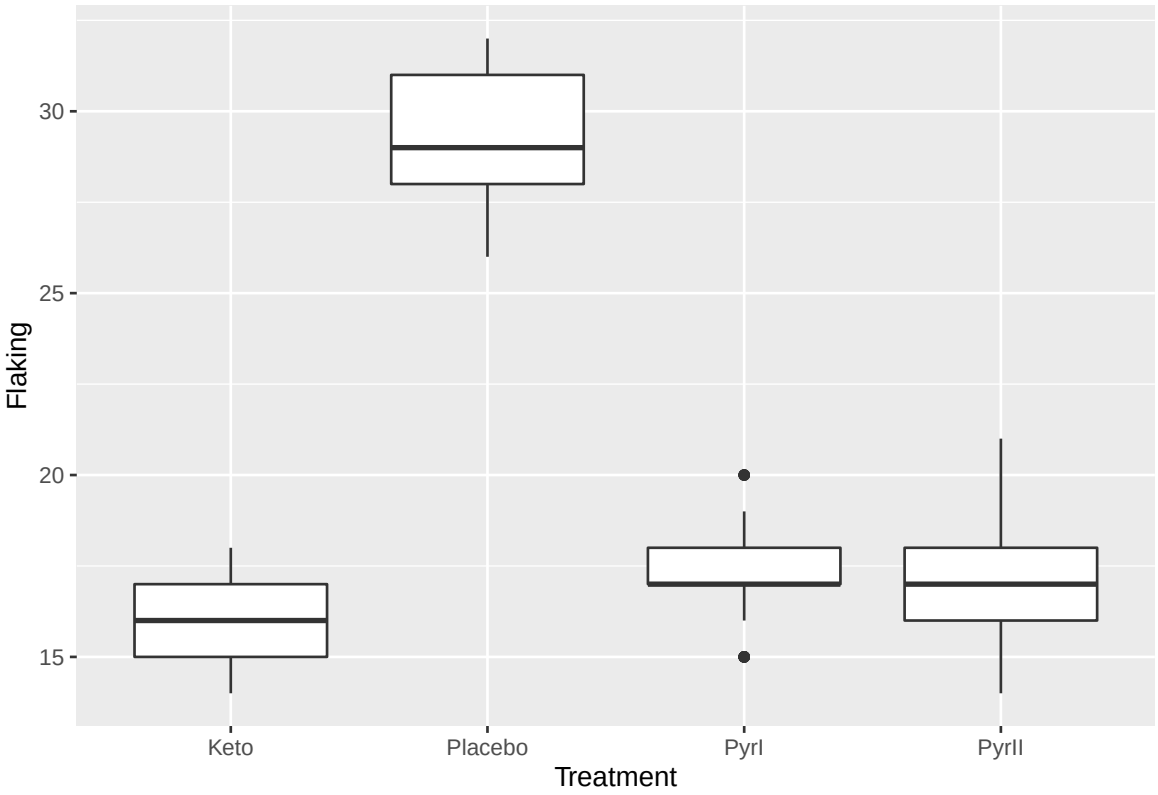
```
dandruff %>% group_by(Treatment) %>%
  summarise(n=n(), fl_mean=mean(Flaking), fl_sd=sd(Flaking))
```

```
## # A tibble: 4 x 4
##   Treatment     n fl_mean fl_sd
## * <chr>     <int>   <dbl> <dbl>
## 1 Keto        106    16.0 0.931
## 2 Placebo      28    29.4 1.59
## 3 PyrI        112    17.4 1.14
## 4 PyrII       109    17.2 1.35
```

There are approximately 100 observations in each group, apart from the Placebo group, which had only 28. The mean number of flakes is much higher for the Placebo group than for the others, which seem similar. The group standard deviations are fairly similar.

With a categorical `Treatment` and a quantitative `Flakes`, a suitable graph is a side-by-side boxplot:

```
ggplot(dandruff, aes(x=Treatment, y=Flaking)) + geom_boxplot()
```



Once again, we see that the flaking for the Placebo shampoo is much higher than for the others. There are outliers in the `PyrI` group, but given that the data values are all whole numbers, they are not far different from the rest of the data. Considering these outliers, the spreads of the groups all look fairly similar and the distributions appear more or less symmetric.[3]

Analysis of Variance

For comparing four groups, we need some kind of analysis of variance. Having seen that the `Flaking` values within the four groups are more or less normal with more or less equal spreads, we run a standard ANOVA:

```
dandruff.1 <- aov(Flaking~Treatment, data=dandruff)
summary(dandruff.1)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Treatment     3   4151  1383.8   967.8 <2e-16 ***
## Residuals   351    502     1.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With an extremely small P-value, we conclude that the four shampoos do not all have the same mean value of `Flaking`.

To find out which ones are different from which, we use Tukey's method:

```
TukeyHSD(dandruff.1)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
```

```
##
## Fit: aov(formula = Flaking ~ Treatment, data = dandruff)
##
## $Treatment
##                       diff         lwr         upr     p adj
## Placebo-Keto   13.3645553  12.7086918  14.0204187 0.0000000
## PyrI-Keto       1.3645553   0.9462828   1.7828278 0.0000000
## PyrII-Keto      1.1735330   0.7524710   1.5945950 0.0000000
## PyrI-Placebo  -12.0000000 -12.6521823 -11.3478177 0.0000000
## PyrII-Placebo -12.1910223 -12.8449971 -11.5370475 0.0000000
## PyrII-PyrI     -0.1910223  -0.6063270   0.2242825 0.6352706
```

All of the shampoo treatments are significantly different from each other except for the two pyrithione ones. To see which shampoos are best and worst, we remind ourselves of the treatment means:

```
dandruff %>% group_by(Treatment) %>%
  summarise(n=n(), fl_mean=mean(Flaking), fl_sd=sd(Flaking))
```

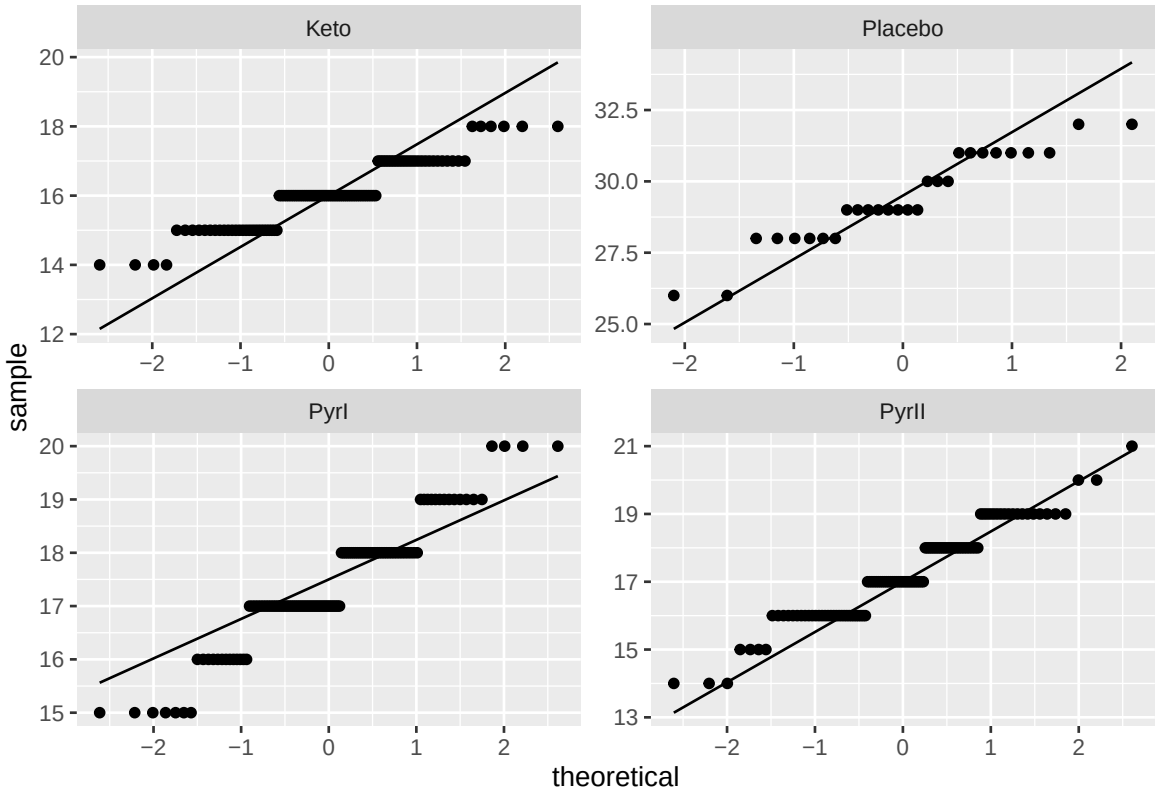
```
## # A tibble: 4 x 4
##   Treatment     n fl_mean fl_sd
## * <chr>     <int>   <dbl> <dbl>
## 1 Keto        106    16.0 0.931
## 2 Placebo      28    29.4 1.59
## 3 PyrI        112    17.4 1.14
## 4 PyrII       109    17.2 1.35
```

The placebo shampoo has significantly more flaking than all the others, and the ketoconazole[4] shampoo has significantly less flaking than all the others. From this analysis, therefore, we would recommend the ketoconazole shampoo over all the others.

Assessment of Assumptions

The analysis of variance done above requires that the observations within each treatment group (shampoo) be approximately normally distributed, given the sample sizes, with approximately equal spreads. To assess this, we look at normal quantile plots[5] for each shampoo:[6]

```
ggplot(dandruff, aes(sample=Flaking)) + stat_qq() + stat_qq_line() +
  facet_wrap(~Treatment, scales = "free")
```



Given that the data are whole numbers, the distributions each appear close to the lines, indicating that the distributions are close to normal in shape.[7] The distribution of the `PyrI` values is slightly long-tailed, but with over 100 observations in that group, this shape is not enough to invalidate the normality assumption.[8]

Having concluded that the normality is sufficient, we need to assess the equality of spreads. Referring back to our summary table:

```
dandruff %>% group_by(Treatment) %>%
  summarise(n=n(), fl_mean=mean(Flaking), fl_sd=sd(Flaking))
```

```
## # A tibble: 4 x 4
##   Treatment     n fl_mean fl_sd
## * <chr>     <int>   <dbl> <dbl>
## 1 Keto        106    16.0 0.931
## 2 Placebo      28    29.4 1.59
## 3 PyrI        112    17.4 1.14
## 4 PyrII       109    17.2 1.35
```

we note that the spreads are not greatly different, and so the equal-spread assumption appears to be satisfied.[9]

In summary, the assumptions for the analysis of variance we did seem to be reasonably well satisfied, and we can have some confidence in our conclusions.

Conclusions

We found that the ketoconazole shampoo produced the smallest mean flaking, and its mean was significantly smaller than that of all the other shampoos. This shampoo can be recommended over the others. There was no significant difference between the two pyrithione treatments; shampooing twice had no benefit over shampooing once.

The difference in means between the ketoconazole and the two pyrithione shampoos was only about 1.2. This difference was significant because of the large sample sizes, but it is a separate question as to whether a difference of this size is of practical importance. If it is not, any of the shampoos except for the placebo can be recommended.

End

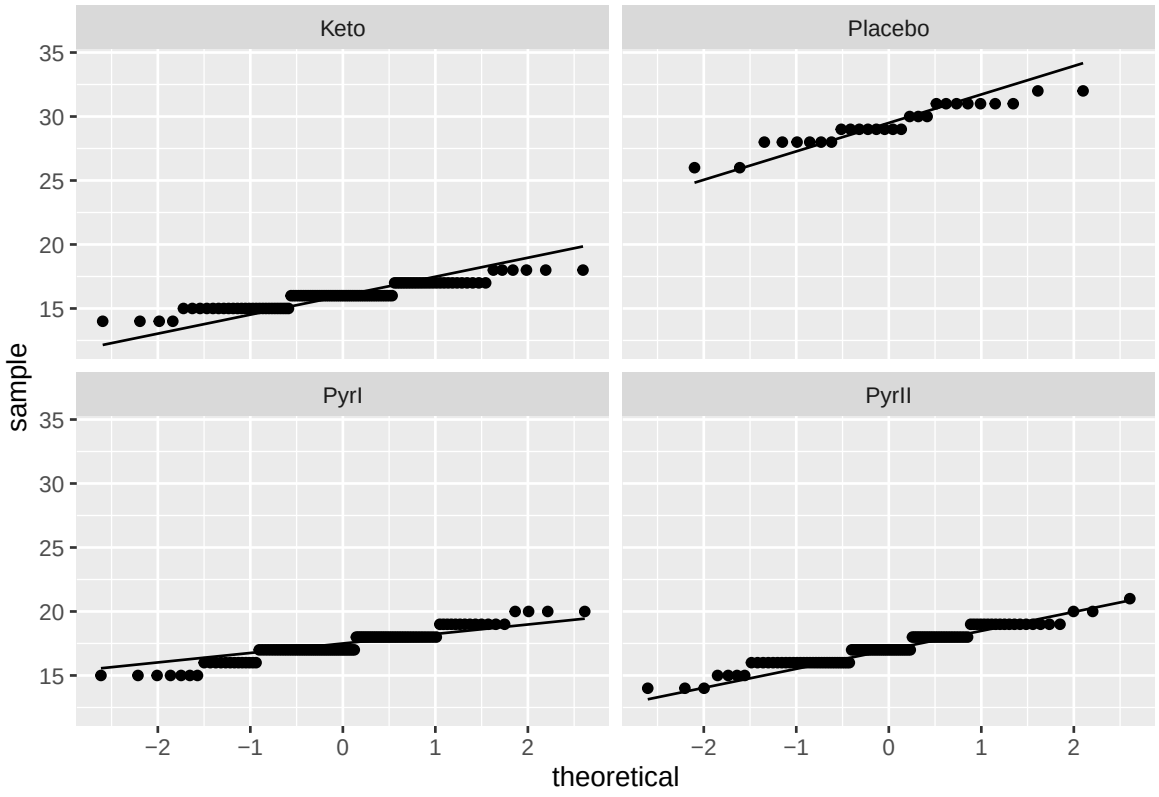
That is the end of my report. You, of course, don't need the word "end" or any of the footnotes I had. These were to draw your attention to other things that don't necessarily belong in the report, but I would like you to be aware of them. When you reach the end of your report, you can just stop.[10]

Some extra, bigger, thoughts (there are quite a few of these. I hope I don't repeat things I also say in the footnotes):

Extra 1: The placebo group is much smaller than the others, but all the groups are pretty big by ANOVA standards. Apparently what happened is that originally the three "real" treatments had 112 subjects each, but the placebo had 28 (ie., a quarter of the subjects that the other groups had), and a few subjects dropped out. There's no problem, in one-way ANOVAs of this kind, with having groups of unequal sizes; the F-test is fine, and as long as you use a suitable extension of Tukey that deals with unequal sample sizes, you are OK there too. `TukeyHSD`, according to the help file, "incorporates an adjustment for sample size that produces sensible intervals for mildly unbalanced designs". In this case, we might be holding our breath a bit, depending on what "mildly unbalanced" actually means. Usually in this kind of study, you have the groups about the same size, because proving that the smallest group differs from any of the others is more difficult. I guess these researchers were pretty confident that the placebo shampoo would be clearly worse than the others! (Additional: `TukeyHSD` uses the so-called Tukey-Kramer test when sample sizes within groups are unequal. My understanding is that this is good no matter what the sample sizes are.)

Extra 2: The mean for `Placebo` is quite a lot bigger than for the other groups, so a plot with different scales for each facet is best. Otherwise you get this kind of thing, which is much harder to read:

```
ggplot(dandruff, aes(sample=Flaking)) + stat_qq() + stat_qq_line() +
  facet_wrap(~Treatment)
```

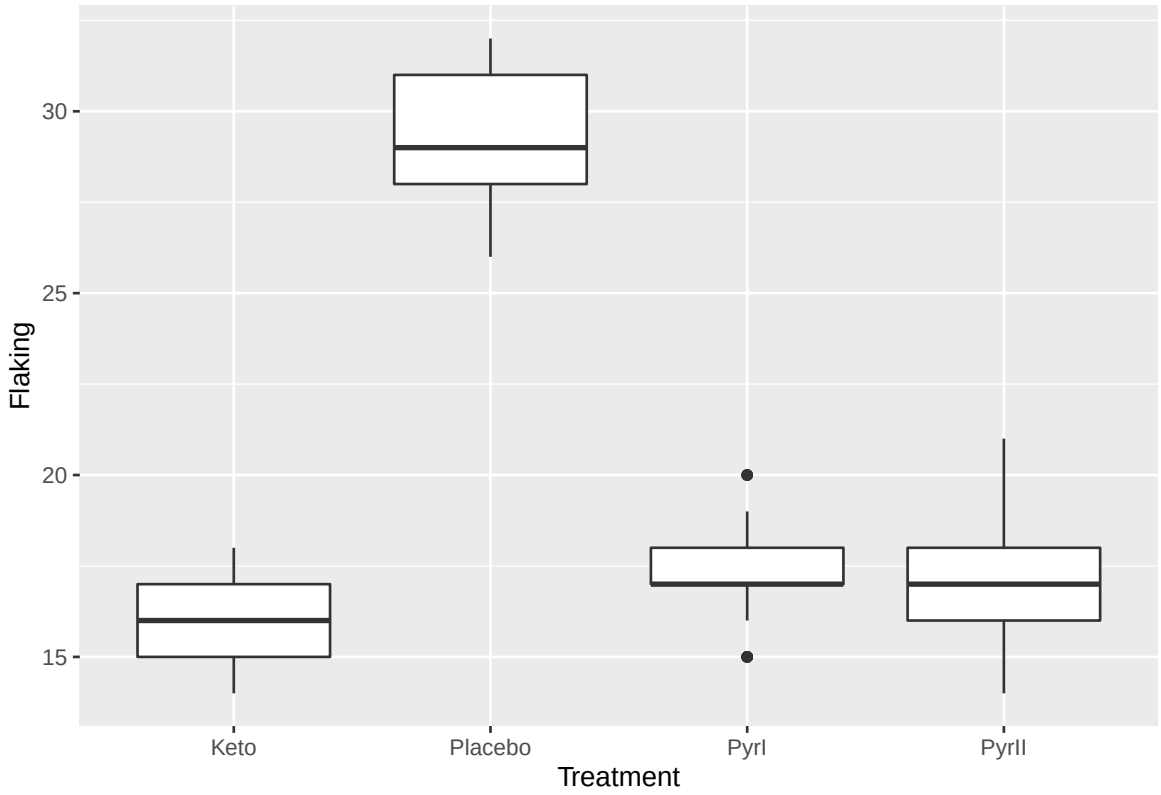


The points fill less than half their facets, which makes the plots harder to understand. This also makes it look as if the distributions are more normal, because the vertical scale has been compressed. Having a better basis for assessing the normality is a good idea, given that the purpose of the plot *is* assessing the normality! Hence, using `scales = "free"` is best.

Extra 3: You might have been wondering why the boxplots, which are the usual thing in these circumstances, look worse than the normal quantile plots.

Let's revisit them and see what happened:

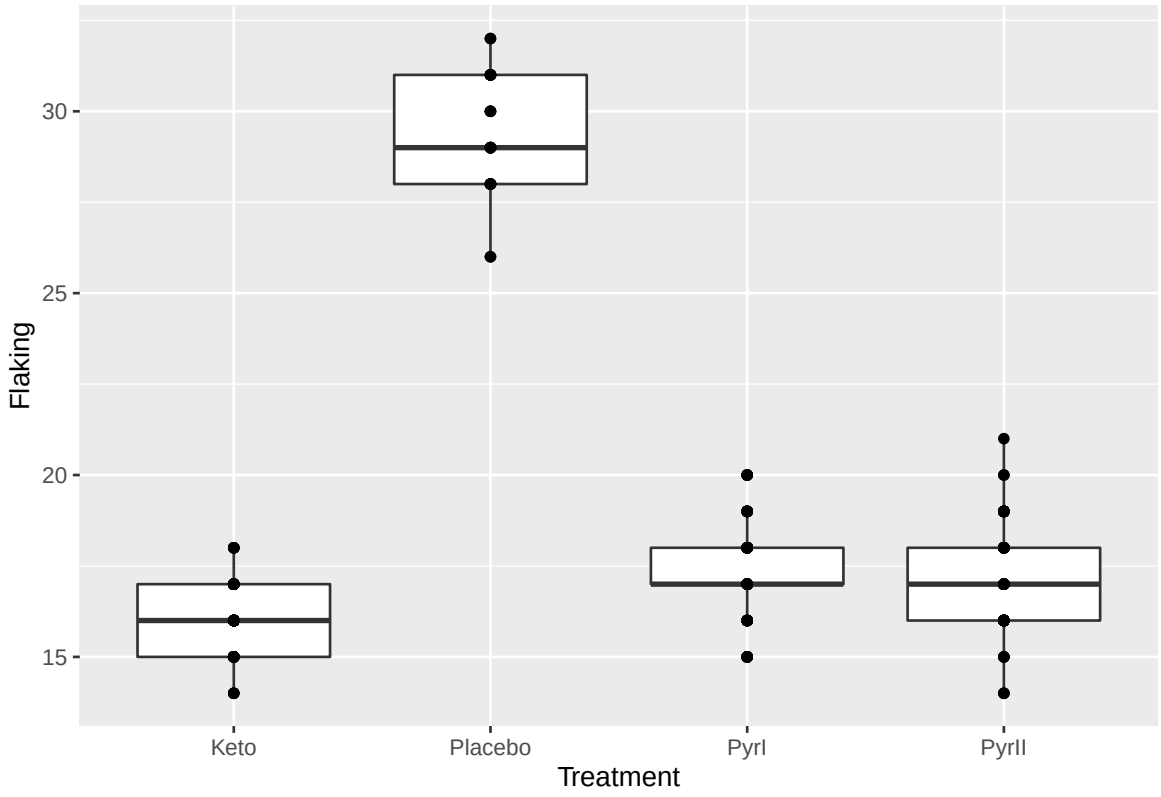
```
ggplot(dandruff, aes(x=Treatment, y=Flaking)) + geom_boxplot()
```



The `Placebo` group has the largest IQR, and the `PyrI` group appears to have two outliers. We need to bear in mind, though, that the data values are whole numbers and there might be repeats; also, what looks like an outlier here might not look quite so much like one when we see all the data.

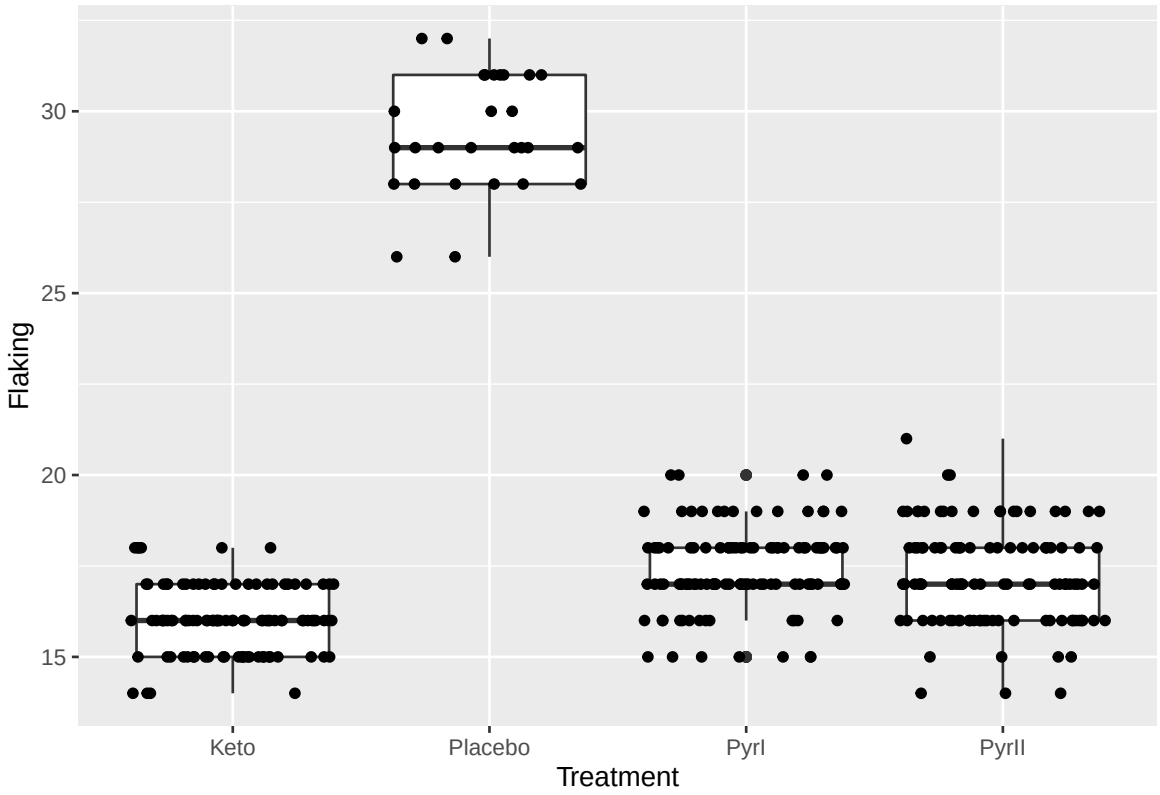
What you can do is to add a `geom_point` on to the plot to plot the observations as points:

```
ggplot(dandruff, aes(x=Treatment, y=Flaking)) +
  geom_boxplot() + geom_point()
```



But there are a lot more data points than this! What happened is that a point at, say, 20 is *all* the observations in that group that were 20, of which there might be a lot, but we cannot see how many, because they are printed on top of each other. To see all the observations, we can **jitter** them: that is, plot them all *not* in the same place. In this case, we have the whole width of the boxplot boxes to use; we could also jitter vertically, but I decided not to do that here. There is a `geom_jitter` that does exactly this:[11]

```
ggplot(dandruff, aes(x=Treatment, y=Flaking)) + geom_boxplot() +
  geom_jitter(height = 0)
```

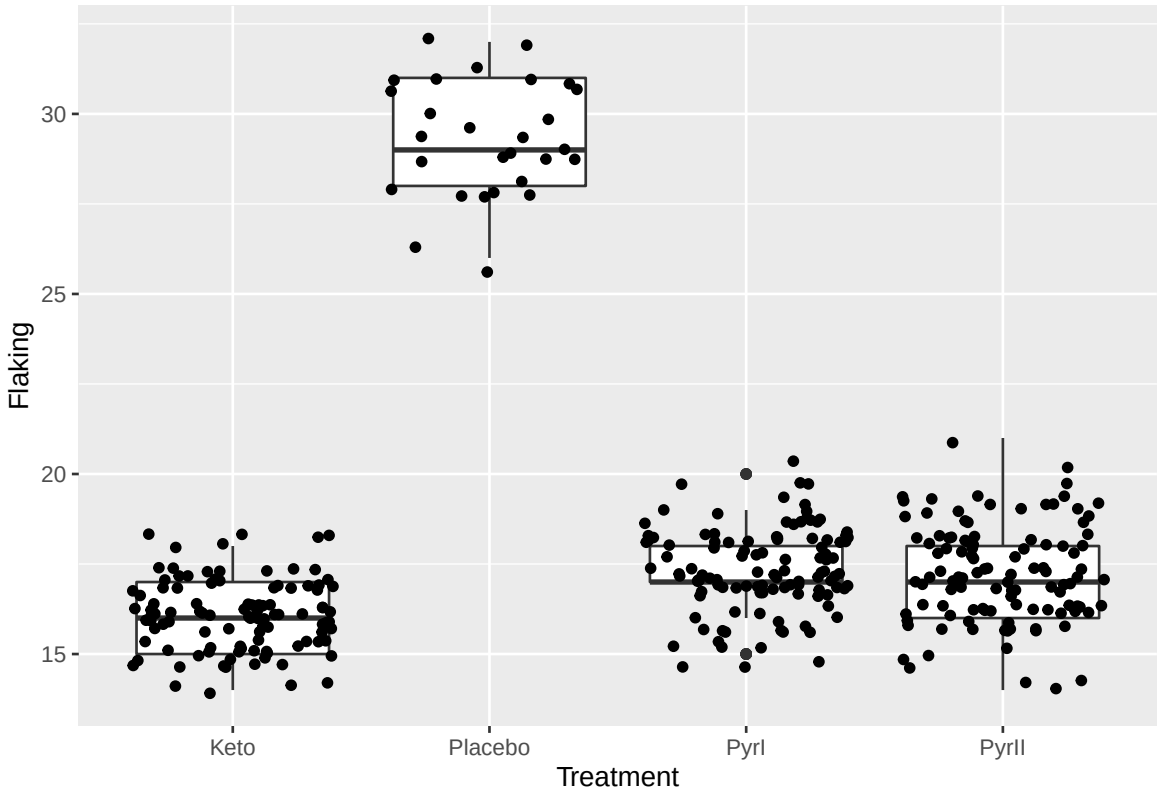


The plot is rather messy,[12] but now you see everything. The `height=0` means not to do any vertical jittering: just spread the points left and right.[13] Where the points are exactly on the x-scale is now irrelevant; this is just a device to spread the points out so that you can see them all.

I left the vertical alone so that you can still see the actual data values. Even though the highest and lowest values in `PyrI` were shown as outliers on the original boxplot, you can see that they are really not. When the data values are discrete (separated) like this, an apparent outlier may be only one bigger or smaller than the next value, and thus not really an outlier at all.

To try the vertical jittering too, use the defaults on `geom_jitter`:

```
ggplot(dandruff, aes(x=Treatment, y=Flaking)) + geom_boxplot() + geom_jitter()
```

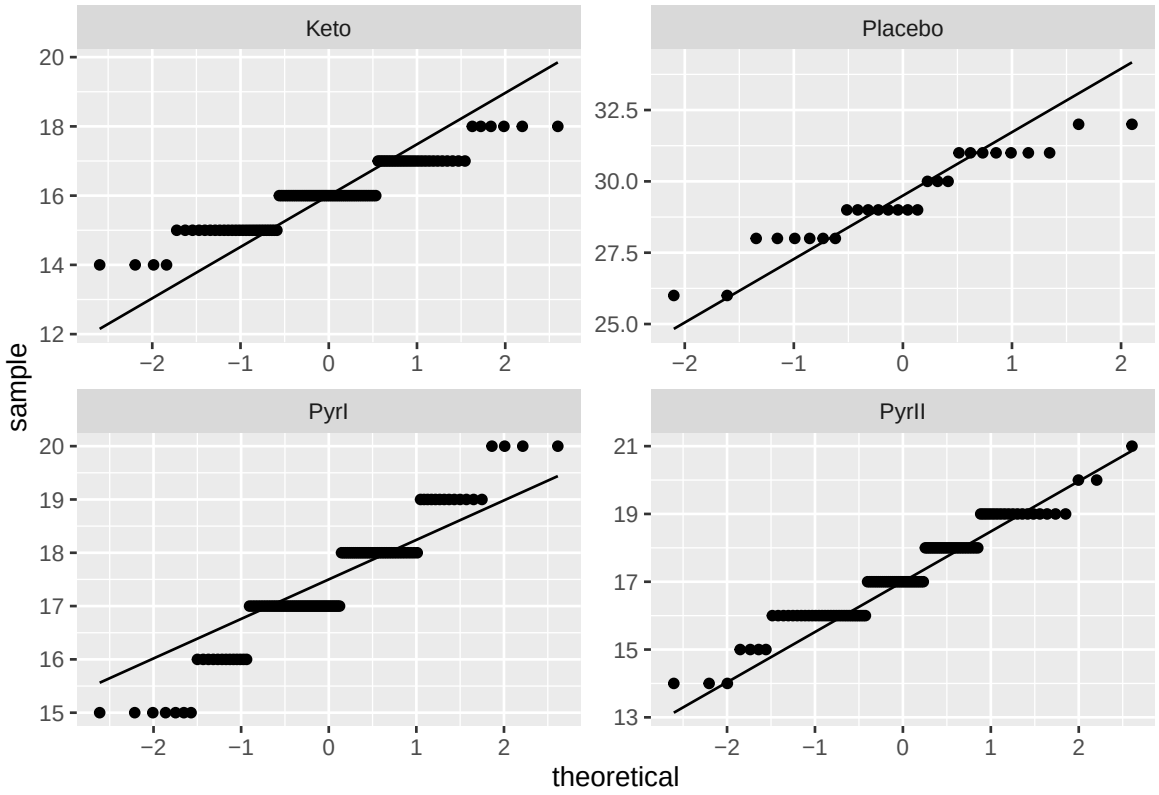


This maybe spreads the points out better, so you can be more certain that you're seeing them all, but you lose the clear picture of the data values being whole numbers.

When Tukey popularized the boxplot, his idea was that it would be drawn by hand with relatively small samples, and when you draw boxplots for large samples, you can get an apparently large number of outliers, that are not in retrospect quite as extreme as they may look at first. This may also have happened here. Tukey, however, did not invent the boxplot; credit for that goes to Mary Eleanor Spear with her "range plot".[14]

Extra 4: More discussion of normality. The assumptions for a standard ANOVA (that is to say, not a Welch ANOVA) are normally-distributed data within each treatment group, with equal spreads. What that means in practice is that you want normal *enough* data given the sample sizes, and approximately equal spreads. My normal quantile plots are a long way back, so let's get them again:

```
ggplot(dandruff, aes(sample=Flaking)) + stat_qq() + stat_qq_line() +
  facet_wrap(~Treatment, scales = "free")
```



The data values are all whole numbers, so we get those horizontal stripes of `Flaking` values that are all the same. As long as these more or less hug the line, we are all right. The `PyrII` values certainly do. In my top row, `Keto` and `Placebo` are not quite so good, but they have *short* tails compared to the normal, so there will be no problem using the means for these groups, as ANOVA does. The only one that is problematic at all is `PyrI`. That has slightly long tails compared to a normal. (You could, I suppose, call those highest and lowest values "outliers", but I don't think they are far enough away from the rest of the data to justify that.) Are these long tails a problem? That depends on how many observations we have:

```
dandruff %>% group_by(Treatment) %>%
  summarise(n=n(), mean_flaking=mean(Flaking), sd_flaking=sd(Flaking))
```

```
## # A tibble: 4 x 4
##   Treatment     n mean_flaking sd_flaking
## * <chr>     <int>        <dbl>      <dbl>
## 1 Keto        106         16.0      0.931
## 2 Placebo      28         29.4      1.59
## 3 PyrI        112         17.4      1.14
## 4 PyrII       109         17.2      1.35
```

There are 112 of them. Easily enough to overcome those long tails. So, to my mind, normality is no problem.

Aside: you might be wondering whether you can make nicer-looking tables in your reports. There are several ways. The `gt` package is the most comprehensive one I know, and has links to a large number of others (at the bottom of its webpage). The simplest one I know of is `kable` in the `knitr` package. You may well have that package already installed, but you'll need to load it, preferably at

the beginning of your report:

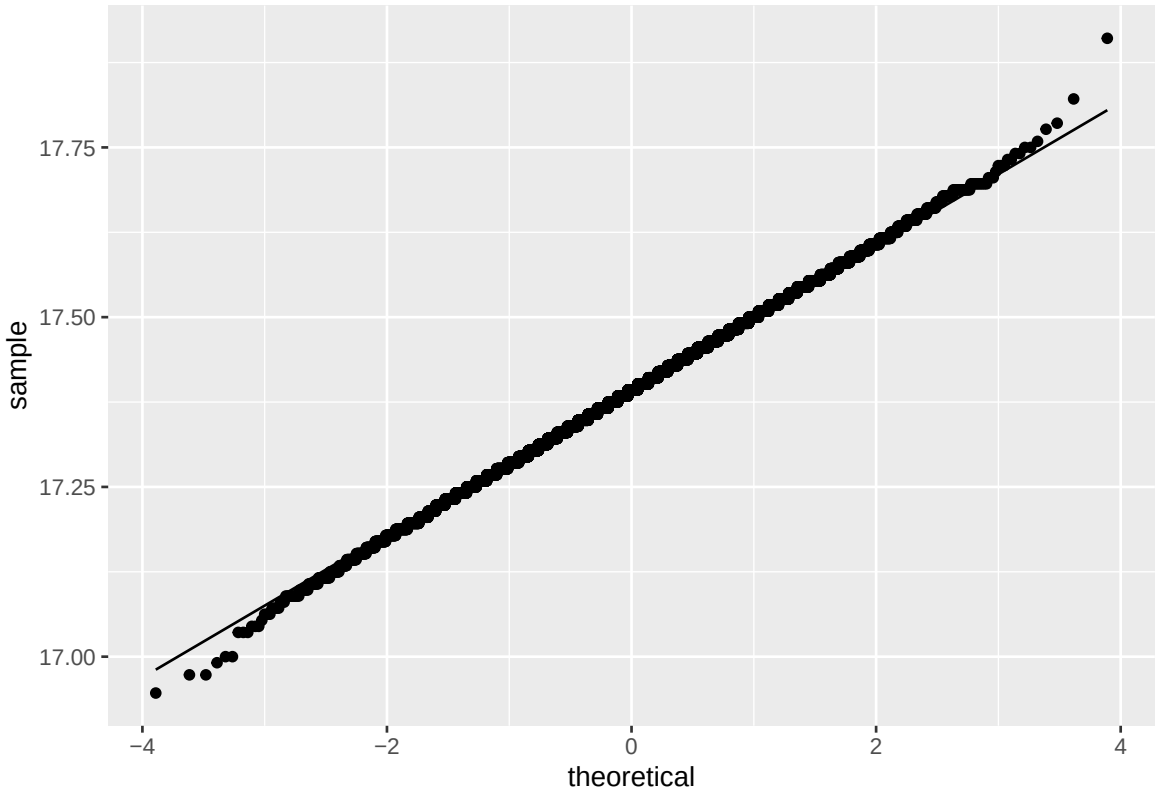
```
library(knitr)
dandruff %>% group_by(Treatment) %>%
  summarise(n=n(), mean_flaking=mean(Flaking), sd_flaking=sd(Flaking)) -> summary
kable(summary)
```

Treatment	n	mean_flaking	sd_flaking
Keto	106	16.02830	0.9305149
Placebo	28	29.39286	1.5948827
PyrI	112	17.39286	1.1418110
PyrII	109	17.20183	1.3524999

End of aside.

Before I got distracted, we were talking about whether the distribution of PyrI was normal enough, given the sample size. Another way of thinking about this is to look at the bootstrapped sampling distribution of the sample mean for this group:

```
dandruff %>% filter(Treatment == "PyrI") -> pyri
rerun(10000, sample(pyri$Flaking, replace = TRUE)) %>%
  map_dbl(~mean(.)) %>%
  enframe() %>%
  ggplot(aes(sample = value)) + stat_qq() + stat_qq_line()
```



Oh yes, no problem with the normality there. (The discreteness of the population implies that each sample mean is some number of one-hundred-and-twelfths, so that the sampling distribution is also discrete, just less discrete than the data distribution. This is the reason for the little stair-steps in the plot.) In addition, the fact that the least normal distribution is normal enough means that the other distributions must also be OK. If you wanted to be careful, you would assess the smallest Placebo group as well, though that if anything is short-tailed and so would not be a problem anyway.

The other question is whether those spreads are equal enough. The easiest way is to look back at your summary table (that I reproduced above), cast your eye down the SD column, and make a call about whether they are equal enough. The large sample sizes *don't* help here, although see the end of the question for more discussion. I would call these "not grossly unequal" and call standard ANOVA good, but you are also entitled to call them different enough, and then you need to say that in your opinion we should have done a Welch ANOVA. Or, if you got your normal quantile plots before you did your ANOVA, you could actually *do* a Welch ANOVA.

I am not a fan of doing one test to see whether you can do another test,[15] but if you really want

to, you can use something like Levene's test to test the null hypothesis that all the groups have the same variance.[16] Levene's test lives in the package `car` that you might have to install first:

```
library(car)
```

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

```
leveneTest(Flaking~Treatment, data = dandruff)
```

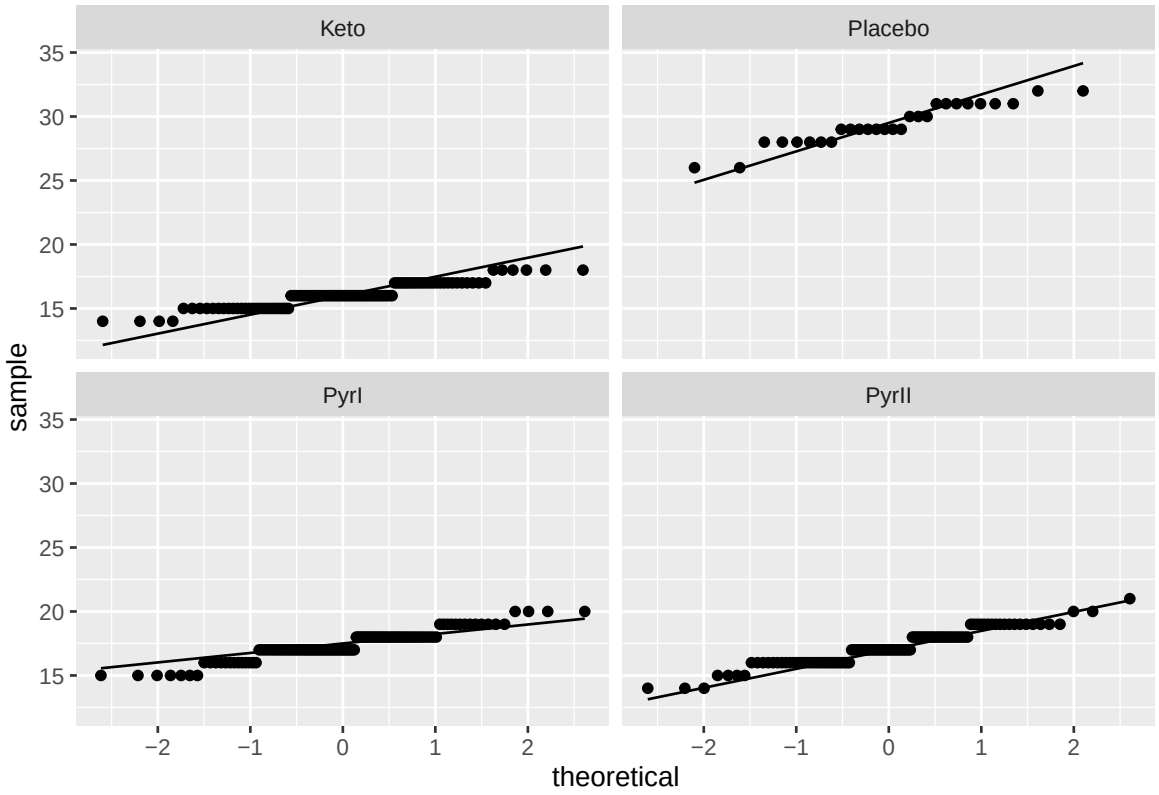
```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   3  6.2001 0.0004096 ***
##       351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Equal variances are resoundingly *rejected* here; the samples here have variances that are less equal than they would be if the populations all had the same variances. But that is really asking the wrong question: the one that matters is "does the inequality of variances that we saw here *matter* when it comes to doing the ANOVA?". With samples as big as we had, the variances could be declared unequal even if they were actually quite similar. This is another (different) angle on statistical significance (rather similar variances can be significantly different with large samples) vs. practical importance (does the fact that our sample variances are as different as they are matter to the ANOVA?). I do the Welch ANOVA in Extra 6, and you will see there whether it comes out much different than the regular ANOVA. See also Extra 7.

If your normal quantile plots looked like this:

```
ggplot(dandruff, aes(sample=Flaking)) + stat_qq() + stat_qq_line() +
  facet_wrap(~Treatment)
```



with the *same* scales, you can use the slopes of the lines to judge equal spreads: either equal enough, or the `Placebo` line is a bit steeper than the others. If you did `scales = "free"`, you *cannot* do this, because you have essentially standardized your data before making the normal quantile plots.

It is *hugely* important to distinguish the null hypothesis (all the means are the same) from the assumptions behind the test (how you know that the P-value obtained from testing your null hypothesis can be trusted). These are separate things, and getting them straight is a vital part of being a good statistician. You might say that this is part of somebody else knowing how they, as someone hiring a statistician, can trust *you*.

Extra 5: Several ways to say what you conclude from the ANOVA:

- The null hypothesis, which says that all the shampoos have the same mean amount of flaking, is rejected. (Or say it in two sentences: what the null hypothesis is, and then what you're doing with it.)
- Not all the shampoos have the same mean amount of flaking.
- There are shampoos that differ in mean amount of flaking.

Some wrong or incomplete ways to say it:

- We reject the null hypothesis. (Meaning what, about the data?)
- we reject the null hypothesis that the means are different. (You have confused the null with the conclusion, and come out with something that is backwards.)
- the mean flaking for the treatments is different (this says that they are *all* different, but you don't know that yet.)

Extra 6: You might have been wondering how Welch's ANOVA would have played out, given that the placebo group measurements looked more variable than the others. Wonder no more:

```
oneway.test(Flaking~Treatment, data=dandruff)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  Flaking and Treatment
## F = 595.03, num df = 3.00, denom df = 105.91, p-value < 2.2e-16
```

```
gamesHowellTest(Flaking~factor(Treatment), data=dandruff)
```

```
##
##  Pairwise comparisons using Games-Howell test
```

```
## data: Flaking by factor(Treatment)
```

```
##         Keto    Placebo PyrI
## Placebo 9.2e-14 -       -
## PyrI    8.7e-14 < 2e-16 -
## PyrII   2.1e-11 < 2e-16 0.67
```

```
##
## P value adjustment method: none
```

```
## alternative hypothesis: two.sided
```

The results are almost exactly the same: the conclusions are identical, and the P-values are even pretty much the same. The place where it would make a difference is when you are close to the boundary between rejecting and not. Here, our Tukey and Games-Howell P-values were all either close to 0 or about 0.6, whichever way we did it. So it didn't matter which one we did; you could justify using either. The regular ANOVA might have been a better choice for your report, though, because this is something your audience could reasonably be expected to have heard of. The Welch ANOVA deserves to be as well-used as the Welch two-sample t-test, but it doesn't often appear in Statistics courses. (This course is an exception, of course!)

Extra 7: the general principle when you are not sure of the choice between two tests is to run them both. If the conclusions agree, as they do here, then it doesn't matter which one you run. If they disagree, then it matters, and you need to think more carefully about which test is the more appropriate one. (Usually, this is the test with the fewer assumptions, but not always.)

Another way to go is to do a simulation (of the ordinary ANOVA). Generate some data that are like what you actually have, and then in your simulation see whether your α is near to 0.05. Since we are talking about α here, the simulated data needs to have the same *mean* in every group, so that the null hypothesis is true, but SDs and sample sizes like the ones in the data (and of a normal shape). Let me build up the process. Let's start by making a dataframe that contains the sample sizes, means and SDs for the data we want to generate. The treatment names don't matter:

```
sim_from <- tribble(
  ~trt, ~n, ~mean, ~sd,
  "A", 106, 0, 0.93,
  "B", 28, 0, 1.59,
  "C", 112, 0, 1.14,
  "D", 109, 0, 1.35
)
sim_from
```

```
## # A tibble: 4 x 4
```

```
##   trt       n  mean    sd
##   <chr> <dbl> <dbl> <dbl>
## 1 A       106     0  0.93
## 2 B        28     0  1.59
## 3 C       112     0  1.14
## 4 D       109     0  1.35
```

The way that comes to mind now is to make a list-column that contains a random sample from each group, and then use `unnest` to break this out into a dataframe that is the same shape as my `dandruff`. The list-column goes like this:

```
sim_from %>%
  mutate(sample = pmap(list(n, mean, sd), ~rnorm(..1, ..2, ..3)))
```

```
## # A tibble: 4 x 5
##   trt       n  mean    sd sample
##   <chr> <dbl> <dbl> <dbl> <list>
## 1 A       106     0  0.93 <dbl [106]>
## 2 B        28     0  1.59 <dbl [28]>
## 3 C       112     0  1.14 <dbl [112]>
## 4 D       109     0  1.35 <dbl [109]>
```

You have probably not seen `pmap` before. The problem is that the for-each has three things in parallel: the sample size, the mean, and the SD. There is a `map2`, but not a `map3` for three things, so what you do is to put the things to be for-eached over in a `list`, then use `pmap`, then refer to the things in your what-to-do (here `rnorm`) with two dots and a number, referring to the first, second, and third things in your `list` in that order. `rnorm` requires the number of random values to generate first, then the mean, then the SD.

OK, so let's unnest these:

```
sim_from %>%
  mutate(sample = pmap(list(n, mean, sd), ~rnorm(..1, ..2, ..3))) %>%
  unnest(sample)
```

```
## # A tibble: 355 x 5
##    trt       n  mean    sd sample
##    <chr> <dbl> <dbl> <dbl>  <dbl>
##  1 A       106     0  0.93 -0.791
##  2 A       106     0  0.93  0.565
##  3 A       106     0  0.93  0.920
##  4 A       106     0  0.93 -0.485
##  5 A       106     0  0.93 -0.144
##  6 A       106     0  0.93 -0.237
##  7 A       106     0  0.93  0.748
##  8 A       106     0  0.93 -0.134
##  9 A       106     0  0.93 -1.19
## 10 A       106     0  0.93 -0.446
## # ... with 345 more rows
```

and if you were to scroll (way!) down, you would see that we have representatives from each treatment group.

Next, we have to run the ANOVA and get the P-value from it (with the aim, in our simulation, of checking that about 5% of the simulated P-values are less than 0.05). I'll have some things to

explain after:

```
library(broom)
```

```
sim_from %>%
  mutate(sample = pmap(list(n, mean, sd),
                       ~rnorm(..1, ..2, ..3))) %>%
  unnest(sample) %>%
  aov(sample ~ trt, data = .) %>%
  tidy() %>%
  pluck("p.value", 1)
```

```
## [1] 0.6035003
```

All right:

- The output of `aov` is designed for human reading, not for computing with. Package `broom` produces "tidy" output in the form of a dataframe, that we can extract things from.
- The dataframe that `aov` is to be run on is the one coming out of the previous step of the pipeline, which has the special name "`.`". I read this as "data equals dot".
- `tidy` (from `broom`) produces the ANOVA table as a dataframe, with a column called `p.value`. There are actually two values in here: remember that when you do an ANOVA table by hand it looks like this in form (with made-up numbers):

```
Source   df Sum of Squares  Mean Square   F    P-value
Groups    3         21            7     3.5     0.01
Error   351        702            2
Total   354        723
```

R doesn't have a Total line (that goes back to the days of Fisher and his recommendations for laying out the hand calculation so that you get it right), but there *is* an Error line (called Residual), and if the dataframe is to have two rows, it must have two values in all the columns; the second one in the P-value column is missing.

- `pluck` pulls values out of dataframes. This one says "go to the column called `p.value` and get just the first value". The result is a simulated P-value from simulated data where the null hypothesis is true.

Now, we want to repeat that a bunch of times. `rerun` is the tool for this, but the thing we're rerunning is kind of complicated. So before we `rerun`, let's make the thing we're rerunning into a function:

```
sim1 <- function(sim_from) {
  sim_from %>%
  mutate(sample = pmap(list(n, mean, sd),
                       ~rnorm(..1, ..2, ..3))) %>%
  unnest(sample) %>%
  aov(sample ~ trt, data = .) %>%
  tidy() %>%
  pluck("p.value", 1)
}
```

```
sim1(sim_from)
```

```
## [1] 0.7673225
```

yep, that's plausible.

And then:

```
rerun(10000, sim1(sim_from)) %>%
  enframe() %>%
  unnest(value) %>%
  count(value <= 0.05)
```

```
## # A tibble: 2 x 2
##   `value <= 0.05`     n
## * <lgl>           <int>
## 1 FALSE            9079
## 2 TRUE              921
```

(this may take an appreciable amount of time).

That looks rather high. Is the proportion of times I am rejecting significantly different from 0.05? Testing null hypotheses about (single) proportions is done using `prop.test`. This uses the normal approximation to the binomial, with continuity correction:

```
prop.test(921, 10000, p = 0.05)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  921 out of 10000, null probability 0.05
## X-squared = 372.25, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.05
## 95 percent confidence interval:
##  0.08653939 0.09797664
## sample estimates:
##      p
## 0.0921
```

Ah, now, that's interesting. A supposed $\alpha = 0.05$ test is actually rejecting around 9% of the time, which is significantly different from 0.05. This surprises me. The confidence interval says that the test is rejecting *more* than 0.05 of the time. So the ANOVA is actually *not* all that accurate.[17]

So now let's do the same simulation for the Welch ANOVA to see whether it's better:

```
sim2 <- function(sim_from) {
  sim_from %>%
  mutate(sample = pmap(list(n, mean, sd),
                       ~rnorm(..1, ..2, ..3))) %>%
  unnest(sample) %>%
  oneway.test(sample ~ trt, data = .) %>%
  .$p.value
}
```

`tidy` doesn't know about `oneway.test`, so I had to pull out the thing called `p.value` myself.

Then:

```
rerun(10000, sim2(sim_from)) %>%
  enframe() %>%
  unnest(value) %>%
  count(value <= 0.05)
```

```
## # A tibble: 2 x 2
##   `value <= 0.05`     n
## * <lgl>           <int>
## 1 FALSE            9496
## 2 TRUE              504
```

```
prop.test(504, 10000, p = 0.05)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  504 out of 10000, null probability 0.05
## X-squared = 0.025789, df = 1, p-value = 0.8724
## alternative hypothesis: true p is not equal to 0.05
## 95 percent confidence interval:
##  0.04623423 0.05491508
## sample estimates:
##      p
## 0.0504
```

This one is right on the money. So this investigation says that the Welch ANOVA is much more trustworthy for data resembling what we observed.

Notes

1. The piece in the problem statement about why these two labels were used is clarification for you and doesn't belong in the report. If you leave it in, you need to at least paraphrase it; simply copying it without having a reason to do so shows that you are not thinking.

2. I'm using a fair few of my own words from the question. This is OK if you think they are clear, but the aim is to write a report that sounds like you rather than me.

3. Offer supported opinions of your own here, which don't need to be the same as mine. Alternatively, you can get the graph and numerical summaries first and comment on them both at once.

4. Use the full name of the shampoo if you are making a conclusion about it.

5. I've used scales = "free" to get the plots to fill their boxes, for the best assessment of normality. The downside of doing it this way is that you cannot use the slopes of the lines to compare spreads. I think this way is still better, though, because the mean for placebo is so much bigger than the others that if you use the same scale for each plot, you'll be wasting a lot of plot real estate that you could use to get a better picture of the normality.

6. It's also good, arguably clearer, to use this as your exploratory plot. This enables you to get to a discussion about normality earlier and you might decide in that case that you don't even need this discussion. You can do the assessment of assumptions first, and then do the corresponding analysis, or you can pick an apparently reasonable analysis and then critique it afterwards. Either way is logical here. In other cases it might be different; for example, in a regression, you might need to fit a model first and improve it after, since it may not be so clear what a good model might be off the top.

7. This is a way to write it if you suspect your reader won't remember what a normal quantile plot is, and by writing it this way you won't insult their intelligence if they *do* remember after all. The other side benefit of writing it this way is that it shows *your* understanding as well.

8. If you have any doubts about sufficient normality, you need to make sure you have also considered the relevant sample size, but if you are already happy with the normality, there is no need. The placebo group, for example, is the smallest, but its shape is if anything short-tailed, so its non-normality will be no problem no matter how small the sample is.

9. I'd rather assess equality of spreads by eyeballing them than by doing a test, but if you really want to, you could use Levene's test, illustrated in problem 12.4 in PASIAS and in Extra 4 for these data. It works for any number of groups, not just two.

10. I wanted to make it clear where my report ended and where the additional chat began.

11. I explain the *height=0* below the plot.

12. It looks to me as if the boxplot has been attacked by mosquitoes.

13. The default jittering is up to a maximum of not quite halfway to the next value. Here that means that each observation is nearest to the box it belongs with.

14. I learned this today.

15. Because the true alpha for the combined procedure in which the test you do second depends on the result of the first test is no longer 0.05; you need to think about what that true alpha is. It might not be too bad here, because regular ANOVA and Welch ANOVA tend to come out similar unless the sample variances are very different, in the same way that the Welch and pooled two-sample tests do. But it is not something to take for granted.

16. There are other tests you could use here. I like Levene's test because it works best when the samples are not normal, but the normality is OK here, so this is not so much of an issue. Of course, the time you want to be assessing equality of variances is when you have already seen sufficient normality, but that might have been because the samples were large rather than that they were especially normal in shape themselves.

17. When you get a simulation result that is not what you were expecting, there are two options to explore: either it really is different from your expectation, or there is something wrong with your code. I think my code is OK here, but do let me know if you see a problem with it.