Assignment 5

Instructions (the same as for Assignment 1): Make an R Notebook and in it answer the two questions below (one Notebook for both questions). When you are done, hand in on Quercus the *output* from Previewing (or Knitting) your Notebook. 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. The only reason to contact the instructor while working on the assignments is to report something missing like a data file that cannot be read.

You have 3 hours to complete this assignment after you start it.

My solutions to this assignment, with extra discussion, will be available after everyone has handed in their assignment.

1. Earlier, we investigated some data on sex and handspan of students in a Statistics class. The data are at http://ritsokiguess.site/STAC32/handspan.txt, with handspans measured in inches. Previously, we did a two-sample t-test to see whether male students had a larger mean handspan than female students.

   (a) Read in and display (some of) the data.

Solution: Exactly as before, so copy what you did then:

```
my_url <- "http://ritsokiguess.site/STAC32/handspan.txt"
span <- read_delim(my_url, " ")
```

```
## Parsed with column specification:
## cols(
##   sex = col_character(),
##   handspan = col_double()
## )
```

```
span
```

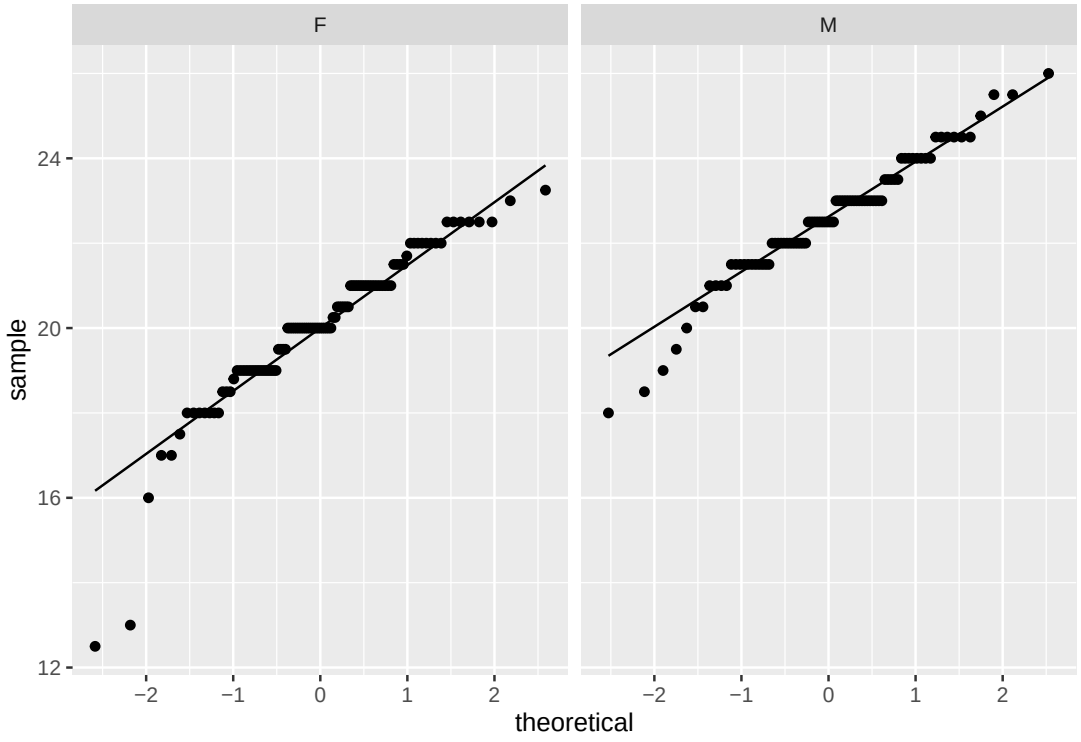
```
## # A tibble: 190 x 2
##    sex   handspan
##    <chr>    <dbl>
##  1 M         21.5
##  2 M         22.5
##  3 M         23.5
##  4 F         20
##  5 F         19
##  6 F         20.5
##  7 F         20.5
```

```
##  8 F         20.2
##  9 M         23
## 10 M         24.5
## # ... with 180 more rows
```

(b) Make a suitable normal quantile plot of the data. (Bear in mind what is supposed to have a normal distribution.)

Solution: Previously, we made boxplots, and found some low-end outliers. Here, we need *each* group to be approximately normal, so make normal quantile plots of handspan, facetted by sex:

```
ggplot(span, aes(sample=handspan)) + stat_qq() + stat_qq_line() +
    facet_wrap(~sex)
```



(c) Explain briefly why you might prefer to use Mood's median test to compare the handspans of the male and female students, compared to a two-sample t-test.

Solution: A two-sample t-test assumes that each of the two samples comes from a (approximately) normal distribution ("the data are normal" is not precise enough). The female values, on the left, definitely have some outliers at the low end (or a long lower tail), so these are definitely not normal. The male values (on the right) are slightly skewed to the left, or there are some mild outliers at the low end, or, if you prefer, these are approximately normal. (You need discussion of each of the males and females.) Because the males are not close enough to normal (or, because neither group is close enough to normal), we would prefer to use Mood's median test. (Say this.) You do yourself a favour by making it clear that you know that *both* groups have to be normal enough; if one is good but the other is not, that is not enough.

The other relevant issue is sample size. The best answer discusses that as well, even though you have a lot to think about already. This data set has 190 observations in it, so the samples must be pretty big:

```
span %>% count(sex)
```

```
## # A tibble: 2 x 2
##   sex       n
##   <chr> <int>
## 1 F       103
## 2 M        87
```

With these sample sizes, we can expect a lot of help from the central limit theorem. The apparent outliers in the males won't be a problem, and maybe we could even get away with those outliers in the females.

Extra: if you read the previous Extra on this data set, you'll recall that the normality was in fact good enough, given the sample sizes (as we assessed with a bootstrap).

(d) Run Mood's median test. What do you conclude from the test, in the context of the data?

Solution:

```
library(smmr)
median_test(span, handspan, sex)
```

```
## $table
##       above
## group above below
##     F    17    82
##     M    65    11
##
## $test
##        what       value
## 1 statistic 8.06725e+01
## 2        df 1.00000e+00
## 3   P-value 2.66404e-19
```

The P-value of 2.66×10^{-19} is extremely small, so we can conclude that males and females have different median handspans. Remember that we are now comparing medians, and that this test is *two*-sided.

You can stop here, or you can go on and note that most of the males have a handspan bigger than the median, and most of the females have a handspan smaller than the median, so that males have on average a larger handspan. But you have to make the case that males have a larger handspan; you cannot just assert this from the P-value.

A more formal way to do this is to make the same observation as above, then note that this is "on the correct side" (for males to have a larger handspan), and thus that you can halve the P-value, and conclude that males' handspans are indeed larger in terms of median.

Extra: you are probably expecting a confidence interval now for the difference in medians. I haven't talked about that in lecture, because the ideas are a bit trickier than they were for the confidence interval for the sign test. The sign test could be used for testing any median, so we could try a bunch of medians and see whether each one was rejected or not. The problem with

Mood's median test is that it only tests that the medians are *the same*. If you could easily test that the difference in medians was 3, say, you would know whether 3 was inside or outside the confidence interval for the difference in medians.

What were the actual sample medians, anyway?

```
span %>% group_by(sex) %>%
  summarize(med = median(handspan))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   sex     med
##   <chr> <dbl>
## 1 F      20
## 2 M      22.5
```

Here's an idea: if we shift all the female handspans up by 2.5 inches, the medians would be the same:

```
span %>% mutate(x = ifelse(sex=="F", handspan+2.5, handspan)) -> d
d
```

```
## # A tibble: 190 x 3
##    sex   handspan     x
##    <chr>    <dbl> <dbl>
##  1 M         21.5  21.5
##  2 M         22.5  22.5
##  3 M         23.5  23.5
##  4 F         20    22.5
##  5 F         19    21.5
##  6 F         20.5  23
##  7 F         20.5  23
##  8 F         20.2  22.8
##  9 M         23    23
## 10 M         24.5  24.5
## # ... with 180 more rows
```

Dataframe `d` has a new column `x` that is the handspan plus 2.5 inches for females, and the unchanged handspan for males. So the median of `x` should be the same for males and females:

```
d %>% group_by(sex) %>%
  summarize(med_x = median(x))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 2
##   sex   med_x
##   <chr> <dbl>
## 1 F      22.5
## 2 M      22.5
```

and also the medians of `x` cannot possibly be significantly different:

```
median_test(d, x, sex)
```

```
## $table
```

```
##      above
## group above below
##     F    46    36
##     M    41    35
##
## $test
##        what      value
## 1 statistic 0.07369901
## 2        df 1.00000000
## 3   P-value 0.78602526
```

Quite a lot of the values of `x` are exactly equal to the overall median (and are discarded), so the P-value is not exactly 1 as you would expect. But it is definitely not significant, and so a difference of 2.5 inches smaller for females is going to be in a confidence interval for the difference in medians.

The strategy now is to try shifting the female handspans by different amounts, run Mood's median test for each one, and see which shifts are not rejected. These are the ones for which that difference in medians would be in the confidence interval. Before we get to that, though, I want to simplify the procedure we have, so that it is easier to run it lots of times. First, let's get just the P-value out of the median test:

```
d.1 <- median_test(d, x, sex)
d.1 %>% pluck("test", "value", 3)
```

```
## [1] 0.7860253
```

That's the P-value. `pluck` pulls individual things out of bigger things. The variable I called `d.1` has two things in it. The one called `table` has the numbers of data values above and below the overall median; the one called `test` has the test statistic and P-value in it. `test` is a dataframe; inside *that* is a column called `what` and a column called `value` with the number we want in it, and we want the third thing in that (the other two are the chi-squared test statistic and its degrees of freedom). Hence the `pluck` statement got the right thing.

Let's think strategy: we want to shift the female handspans by a bunch of different amounts, run the test on each one, and get the P-value each time. When you're running a big for-each like this, you want the thing you do each time to be as simple as possible. So let's write a function that takes the shift as input, works out the new `x`, runs the test, and returns the P-value. We have all the ingredients, so it's a matter of putting them together:

```
shift_pval <- function(shift) {
  span %>% mutate(x = ifelse(sex == "F", handspan + shift, handspan)) -> d
  d.1 <- median_test(d, x, sex)
  d.1 %>% pluck("test", "value", 3)
}
```

In the function, the `shift` is input. The first line computes the handspans shifted by the input amount, whatever it is; the second line runs the median test on the shifted data; the last line pulls out, and returns, the P-value.

Let's test this on a shift of 2.5 inches, and on the original data (a shift of zero):

```
shift_pval(2.5)
```

```
## [1] 0.7860253
```

```
shift_pval(0)
```

```
## [1] 2.66404e-19
```

Those are the same P-values we got before, so good.

Now, let's get a bunch of shifts, say from 0 to 5 in steps of 0.5:

```
tibble(shift = seq(0, 5, 0.5))
```

```
## # A tibble: 11 x 1
##    shift
##    <dbl>
##  1   0
##  2   0.5
##  3   1
##  4   1.5
##  5   2
##  6   2.5
##  7   3
##  8   3.5
##  9   4
## 10   4.5
## 11   5
```

work out the P-value for each one (this is `map_dbl` because each P-value is a single number):

```
tibble(shift = seq(0, 5, 0.5)) %>%
  mutate(p_value = map_dbl(shift, ~shift_pval(.)))
```

```
## # A tibble: 11 x 2
##    shift  p_value
##    <dbl>    <dbl>
##  1   0   2.66e-19
##  2   0.5 7.38e-15
##  3   1   6.41e- 9
##  4   1.5 5.46e- 5
##  5   2   5.29e- 2
##  6   2.5 7.86e- 1
##  7   3   5.15e- 3
##  8   3.5 1.13e- 5
##  9   4   1.89e- 9
## 10   4.5 1.84e-14
## 11   5   1.61e-18
```

and finally decide whether each shift is inside or outside the CI (because I am too lazy to figure out the scientific notation):

```
tibble(shift = seq(0, 5, 0.5)) %>%
  mutate(p_value = map_dbl(shift, ~shift_pval(.))) %>%
  mutate(where = ifelse(p_value<0.05, "outside", "inside"))
```

```
## # A tibble: 11 x 3
##    shift  p_value where
##    <dbl>    <dbl> <chr>
```

```
##  1   0   2.66e-19 outside
##  2   0.5 7.38e-15 outside
##  3   1   6.41e- 9 outside
##  4   1.5 5.46e- 5 outside
##  5   2   5.29e- 2 inside
##  6   2.5 7.86e- 1 inside
##  7   3   5.15e- 3 outside
##  8   3.5 1.13e- 5 outside
##  9   4   1.89e- 9 outside
## 10   4.5 1.84e-14 outside
## 11   5   1.61e-18 outside
```

The confidence interval goes from 2 inches to 2.5 inches on this scale. I checked and it goes up to 3 really, except that 3 itself is outside the interval. So let's call it 2 to 3 inches. This means that the median female handspan is between 2 and 3 inches *smaller* than the median male handspan, because we had to shift the female handspans up by that much to make them not significantly different.

You, of course, would do just the last pipeline; I showed you the steps so you could see what was going on.

The final observation is that this interval is a long way from containing zero, because the P-value was so tiny. I had forgotten how the t-interval looked in comparison (two-sided now because we want the interval):

```
t.test(handspan~sex, data = span)
```

```
##
##  Welch Two Sample t-test
##
## data:  handspan by sex
## t = -10.871, df = 187.92, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.001496 -2.079466
## sample estimates:
## mean in group F mean in group M
##        20.01699        22.55747
```

Almost exactly the same (except that F is before M). So it made no difference at all whether we did a t-test or a Mood's median test.

2. Some people believe that poets, especially female poets, die younger than other types of writer. William Butler Yeats[1] wrote:

> She is the Gaelic[2] muse, for she gives inspiration to those she persecutes. The Gaelic poets die young, for she is restless, and will not let them remain long on earth.

A literature student wanted to investigate this, and so collected a sample of 123 female writers (of three different types), and noted the age at death of each writer.

The data are in http://ritsokiguess.site/STAC32/writers.csv.

(a) Read in and display (some of) the data.

Solution:

The usual:

```
my_url <- "http://ritsokiguess.site/STAC32/writers.csv"
writers <- read_csv(my_url)
```

```
## Parsed with column specification:
## cols(
##   Type1 = col_double(),
##   Type = col_character(),
##   Age = col_double()
## )
```

```
writers
```

```
## # A tibble: 123 x 3
##    Type1 Type     Age
##    <dbl> <chr>  <dbl>
##  1     1 Novels    57
##  2     1 Novels    90
##  3     1 Novels    67
##  4     1 Novels    56
##  5     1 Novels    90
##  6     1 Novels    72
##  7     1 Novels    56
##  8     1 Novels    90
##  9     1 Novels    80
## 10     1 Novels    74
## # ... with 113 more rows
```

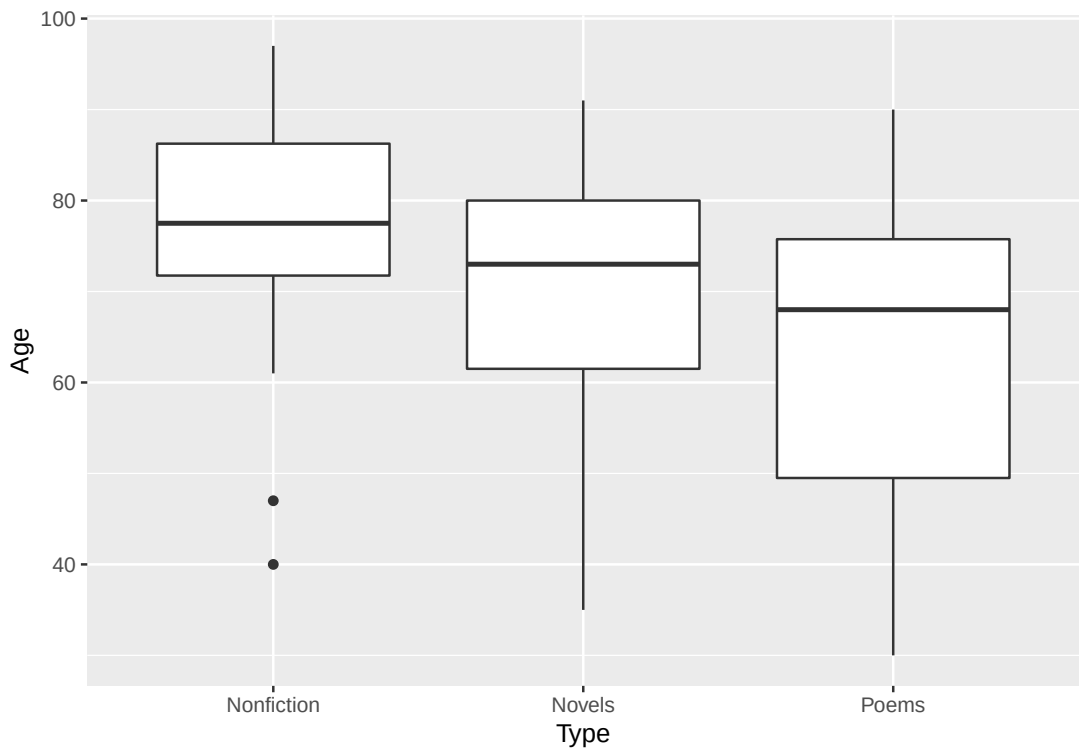
There are indeed 123 writers. The second column shows the principal type of writing each writer did, and the third column shows their age at death. The first column is a numerical code for the type of writing, which we ignore (since we can handle the text writing type).

(b) Make a suitable plot of the ages and types of writing.

Solution:

As usual, one quantitative and one categorical, so a boxplot:

```
ggplot(writers, aes(x=Type, y=Age)) + geom_boxplot()
```



At this point, a boxplot is best, since right now you are mostly after a general sense of what is going on, rather than assessing normality in particular (that will come later).

(c) Obtain a summary table showing, for each type of writing, the number of writers of that type, along with the mean, median and standard deviation of their ages at death.

Solution:

The customary `group_by` and `summarize`:

```
writers %>% group_by(Type) %>%
  summarize(n=n(), mean=mean(Age), med=median(Age), sd=sd(Age))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 5
##   Type           n  mean   med    sd
##   <chr>      <int> <dbl> <dbl> <dbl>
## 1 Nonfiction    24  76.9  77.5  14.1
## 2 Novels        67  71.4  73    13.1
## 3 Poems         32  63.2  68    17.3
```

(d) Run a complete analysis, starting with an ordinary (not Welch) analysis of variance, that ends with a conclusion in the context of the data and an assessment of assumptions.

Solution:

I've left this fairly open-ended, to see how well you know what needs to be included and what it means. There is a lot of room here for explanatory text to show that you know what you are doing. One output followed by another *without* any explanatory text suggests that you are just copying what I did without any idea about why you are doing it.

The place to start is the ordinary (not Welch) ANOVA. You may not think that this is the best thing to do (you'll have a chance to talk about that later), but I wanted to make sure that you practiced the procedure:

```
writers.1 <- aov(Age~Type, data=writers)
summary(writers.1)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Type          2   2744  1372.1   6.563 0.00197 **
## Residuals   120  25088   209.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This says that the mean ages at death of the three groups of writers are not all the same, or that there are differences among those writers (in terms of mean age at death). "The mean ages of the types of writer are different" is not accurate enough, because it comes too close to saying that *all three* groups are different, which is more than you can say right now.

The F-test is significant, meaning that there are some differences among[3] the means, and Tukey's method will enable us to see which ones differ:

```
TukeyHSD(writers.1)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Age ~ Type, data = writers)
##
## $Type
##                          diff       lwr       upr     p adj
## Novels-Nonfiction  -5.427239 -13.59016  2.735681 0.2591656
## Poems-Nonfiction  -13.687500 -22.95326 -4.421736 0.0018438
## Poems-Novels       -8.260261 -15.63375 -0.886772 0.0240459
```

There is a significant difference in mean age at death between the poets and both the other types of writer. The novelists and the nonfiction writers do not differ significantly in mean age at death.

We know from the boxplots (or the summary table) that this significant difference was because the poets died *younger* on average, which is exactly what the literature student was trying to find out. Thus, female poets really do die younger on average than female writers of other types. It is best to bring this point out, since this is the reason we (or the literature student) were doing this analysis in the first place. See Extra 1 for more.

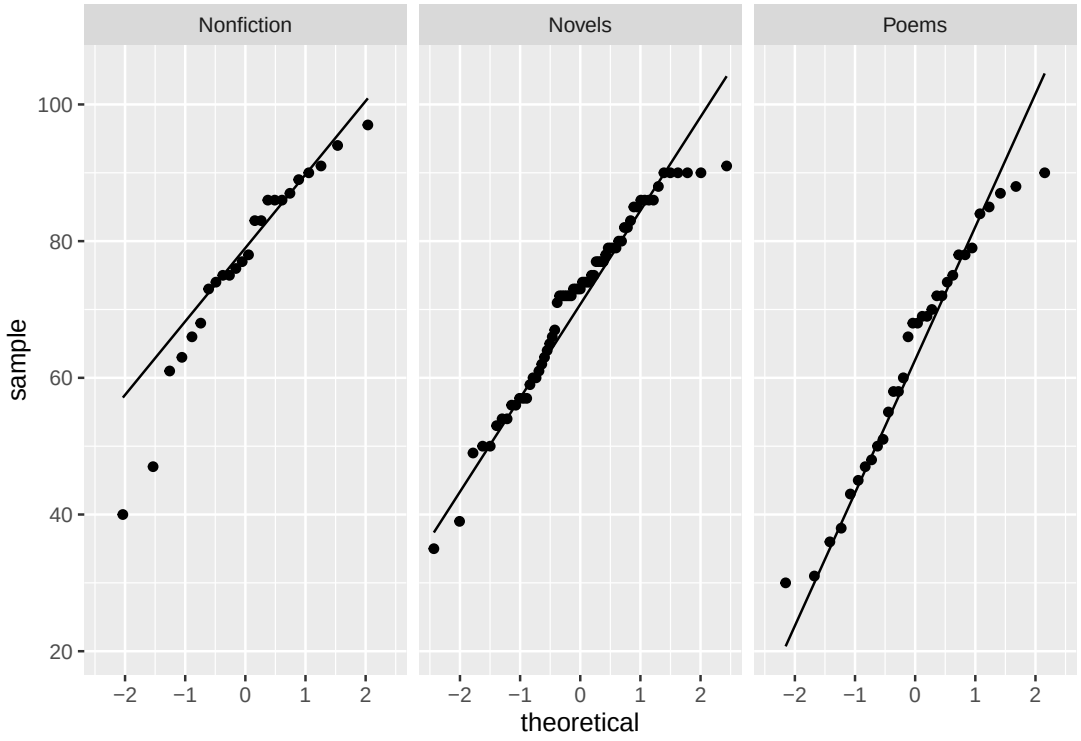
So now we need to assess the assumptions on which the ANOVA depends.

The assumption we made is that the ages at death of the authors of each different type had approximately a normal distribution (given the sample sizes) with approximately equal spread. The boxplots definitely look skewed to the left (well, not the poets so much, but the others

definitely). So now consider the sample sizes: 24, 67, and 32 for the three groups (respectively), and make a call about whether you think the normality is good enough. You are certainly entitled to declare the two outliers on the nonfiction writers to be too extreme given a sample size of only 24. Recall that once one sample fails normality, that's all you need.

Now, since you specifically want normality, you could reasonably look at normal quantile plots instead of the boxplots. Don't just get normal quantile plots, though; say something about why you want them instead of the boxplots you drew earlier:

```
ggplot(writers, aes(sample = Age)) +
  stat_qq() + stat_qq_line() +
  facet_wrap(~Type)
```



I see that the Nonfiction writers have two outliers at the low end (and are otherwise not bad); the writers of Novels don't go up high enough (it's almost as if there is some magic that stops them living beyond 90!); the writers of Poems have a short-tailed distribution. You'll remember that short tails are not a problem, since the mean is still descriptive of such a distribution; it's *long* tails or outliers or skewness that you need to be worried about. The outliers in the Nonfiction writers are the biggest concern.

Are you concerned that these outliers are a problem, given the sample size? There are only 24 nonfiction writers (from your table of means earlier), so the Central Limit Theorem will help a bit. Make a call about whether these outliers are a big enough problem. You can go either way on this, as long as you raise the relevant issues.

Another approach you might take is to look at the P-values. The one in the F-test is really small, and so is one of the ones in the Tukey. So even if you think the analysis is a bit off, those conclusions are not likely to change. The 0.02 P-value in the Tukey, however, is another story. This could become non-significant in actual fact if the P-value is not to be trusted.

Yet another approach (looking at the bootstrapped sampling distributions of the sample means) is in Extra 3. This gets more than a little hairy with three groups, especially doing it the way I do.

If you think that the normality is not good enough, it's a good idea to suggest that we might do a Mood's Median Test instead, and you could even do it (followed up with pairwise median tests). If you think that normality is all right, you might then look at the spreads. I think you ought to conclude that these are close enough to equal (the SDs from the summary table or the heights of the boxes on the boxplots), and so there is no need to do a Welch ANOVA. (Disagree if you like, but be prepared to make the case.)

I have several Extras:

Extra 1: having come to that tidy conclusion, we really ought to back off a bit. These writers were (we assume) a random sample of some population, but they were actually mostly Americans, with a few Canadian and Mexican writers. So this appears to be true at least for North American writers. But this is (or might be) a different thing to the Yeats quote about female Gaelic poets.

There is a more prosaic reason. It is harder (in most places, but especially North America) to get poetry published than it is to find a market for other types of writing. (A would-be novelist, say, can be a journalist or write for magazines to pay the bills while they try to find a publisher for their novel.) Thus a poet is living a more precarious existence, and that might bring about health problems.

Extra 2: with the non-normality in mind, maybe Mood's median test is the thing:

```
median_test(writers, Age, Type)
```

```
## $table
##             above
## group        above below
##   Nonfiction    17     6
##   Novels        33    30
##   Poems         10    22
##
## $test
##        what       value
## 1 statistic 9.872664561
## 2        df 2.000000000
## 3   P-value 0.007180888
```

The P-value here is a bit bigger than for the F-test, but it is still clearly significant. Hence, we do the pairwise median tests to find out which medians differ:

```
pairwise_median_test(writers, Age, Type)
```

```
## # A tibble: 3 x 4
##   g1         g2     p_value adj_p_value
##   <chr>      <chr>    <dbl>       <dbl>
## 1 Nonfiction Novels 0.0531      0.159
## 2 Nonfiction Poems  0.00119     0.00358
## 3 Novels     Poems  0.0142      0.0426
```

The conclusion here is exactly the same as for the ANOVA. The P-values have moved around a bit, though: the first one is a little closer to significance (remember, look at the last column

since we are doing three tests at once) and the last one is now only just significant.

```
writers %>% group_by(Type) %>%
  summarize(n=n(), mean=mean(Age), med=median(Age), sd=sd(Age))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 5
##   Type           n  mean   med    sd
##   <chr>      <int> <dbl> <dbl> <dbl>
## 1 Nonfiction    24  76.9  77.5  14.1
## 2 Novels        67  71.4  73    13.1
## 3 Poems         32  63.2  68    17.3
```

In both of these two cases (Nonfiction-Novels and Novels-Poems), the medians are closer together than the means are. That would explain why the Novels-Poems P-value would increase, but not why the Nonfiction-Novels one would decrease.

I would have no objection *in general* to your running a Mood's Median Test on these data, but the point of *this* problem was to give you practice with `aov`.

Extra 3: the other way to assess if the normality is OK given the sample sizes is to obtain bootstrap sampling distributions of the sample means for each `Type`. The sample size for the novelists is 67, so I would expect the skewness there to be fine, but the two outliers among the Nonfiction writers may be cause for concern, since there are only 24 of those altogether.

Let's see if we can do all three at once (I like living on the edge). I take things one step at a time, building up a pipeline as I go. Here's how it starts:

```
writers %>% nest(data=Age)
```

```
## # A tibble: 3 x 3
##   Type1 Type       data
##   <dbl> <chr>      <list>
## 1     1 Novels     <tibble [67 x 1]>
## 2     2 Poems      <tibble [32 x 1]>
## 3     3 Nonfiction <tibble [24 x 1]>
```

The thing `data` is a so-called list-column. I don't think you've seen one of these before, so let me take a moment to explain. The dataframes we have seen so far are like spreadsheets, in that each "cell" or "entry" in a dataframe has something like a number or a piece of text in it (or, occasionally, a thing that is True or False, or a date). Tibble-type dataframes are more flexible than that, however: each cell of a dataframe could contain *anything*.

In this one, the three things in the column `data` are each *dataframes*,[4] containing the column called `Age` from the original dataframe. These are the ages at death of the writers of that particular `Type`. These are the things we want bootstrap samples of.

I'm not at all sure how this is going to go, so let's shoot for just 5 bootstrap samples to start with. If we can get it working, we can scale up the number of samples later, but having a smaller number of samples is easier to look at:

```
writers %>% nest(data=Age) %>%
  mutate(samples=map(data, ~rerun(5, sample(.$Age, replace = TRUE))))
```

```
## # A tibble: 3 x 4
##   Type1 Type       data              samples
```

```
##   <dbl> <chr>      <list>            <list>
## 1     1 Novels     <tibble [67 x 1]> <list [5]>
## 2     2 Poems      <tibble [32 x 1]> <list [5]>
## 3     3 Nonfiction <tibble [24 x 1]> <list [5]>
```

This is the right kind of thing, but the problem with this kind of display is that we don't know what those two list-columns actually contain. To find out, piping the above into `str` ("structure") gives more detail:

```
writers %>% nest(data=Age) %>%
  mutate(samples=map(data, ~rerun(5, sample(.$Age, replace = T)))) %>% str()
```

```
## tibble [3 x 4] (S3: tbl_df/tbl/data.frame)
##  $ Type1  : num [1:3] 1 2 3
##  $ Type   : chr [1:3] "Novels" "Poems" "Nonfiction"
##  $ data   :List of 3
##   ..$ : tibble [67 x 1] (S3: tbl_df/tbl/data.frame)
##   .. ..$ Age: num [1:67] 57 90 67 56 90 72 56 90 80 74 ...
##   ..$ : tibble [32 x 1] (S3: tbl_df/tbl/data.frame)
##   .. ..$ Age: num [1:32] 88 69 78 68 72 60 50 47 74 36 ...
##   ..$ : tibble [24 x 1] (S3: tbl_df/tbl/data.frame)
##   .. ..$ Age: num [1:24] 74 86 87 68 76 73 63 78 83 86 ...
##  $ samples:List of 3
##   ..$ :List of 5
##   .. ..$ : num [1:67] 62 53 78 50 62 62 74 77 50 75 ...
##   .. ..$ : num [1:67] 73 67 67 85 90 73 67 73 90 75 ...
##   .. ..$ : num [1:67] 65 90 57 59 72 72 86 72 50 56 ...
##   .. ..$ : num [1:67] 71 50 75 74 60 85 72 63 90 82 ...
##   .. ..$ : num [1:67] 61 74 75 77 78 88 80 50 63 61 ...
##   ..$ :List of 5
##   .. ..$ : num [1:32] 69 45 31 84 75 79 66 31 47 85 ...
##   .. ..$ : num [1:32] 48 45 68 38 48 43 66 74 58 60 ...
##   .. ..$ : num [1:32] 36 85 58 90 72 66 69 58 30 75 ...
##   .. ..$ : num [1:32] 50 58 43 79 74 58 88 69 55 75 ...
##   .. ..$ : num [1:32] 38 88 78 43 50 74 38 78 88 75 ...
##   ..$ :List of 5
##   .. ..$ : num [1:24] 47 78 61 75 68 97 63 87 47 74 ...
##   .. ..$ : num [1:24] 63 97 74 89 75 86 66 83 75 75 ...
##   .. ..$ : num [1:24] 86 83 83 83 94 97 78 75 83 47 ...
##   .. ..$ : num [1:24] 89 97 91 89 75 40 47 83 75 76 ...
##   .. ..$ : num [1:24] 74 40 77 97 83 89 66 91 90 90 ...
```

Those do indeed look like five bootstrap samples of ages from each type of writer. So we are doing OK so far.

Next, we want the means of each sample. This is awkward, because we want the mean of *each* sample, and they are nested too deeply for comfort: not just in the `samples` column, but inside a list in that. Let's see if we can make them a bit more accessible:

```
writers %>% nest(data=Age) %>%
  mutate(samples=map(data, ~rerun(5, sample(.$Age, replace = T)))) %>%
  unnest(samples)
```

```
## # A tibble: 15 x 4
```

```
##    Type1 Type       data              samples
##    <dbl> <chr>      <list>            <list>
##  1     1 Novels     <tibble [67 x 1]> <dbl [67]>
##  2     1 Novels     <tibble [67 x 1]> <dbl [67]>
##  3     1 Novels     <tibble [67 x 1]> <dbl [67]>
##  4     1 Novels     <tibble [67 x 1]> <dbl [67]>
##  5     1 Novels     <tibble [67 x 1]> <dbl [67]>
##  6     2 Poems      <tibble [32 x 1]> <dbl [32]>
##  7     2 Poems      <tibble [32 x 1]> <dbl [32]>
##  8     2 Poems      <tibble [32 x 1]> <dbl [32]>
##  9     2 Poems      <tibble [32 x 1]> <dbl [32]>
## 10     2 Poems      <tibble [32 x 1]> <dbl [32]>
## 11     3 Nonfiction <tibble [24 x 1]> <dbl [24]>
## 12     3 Nonfiction <tibble [24 x 1]> <dbl [24]>
## 13     3 Nonfiction <tibble [24 x 1]> <dbl [24]>
## 14     3 Nonfiction <tibble [24 x 1]> <dbl [24]>
## 15     3 Nonfiction <tibble [24 x 1]> <dbl [24]>
```

The original dataframe had three rows, one for each `Type`, and all five bootstrap samples hidden in one row. This has split out each bootstrap sample into its own row, where it is easier to get at, with the result that there are now five rows where `Type` is `Novels`, one for each bootstrap sample, and $5 \times 3 = 15$ rows altogether.

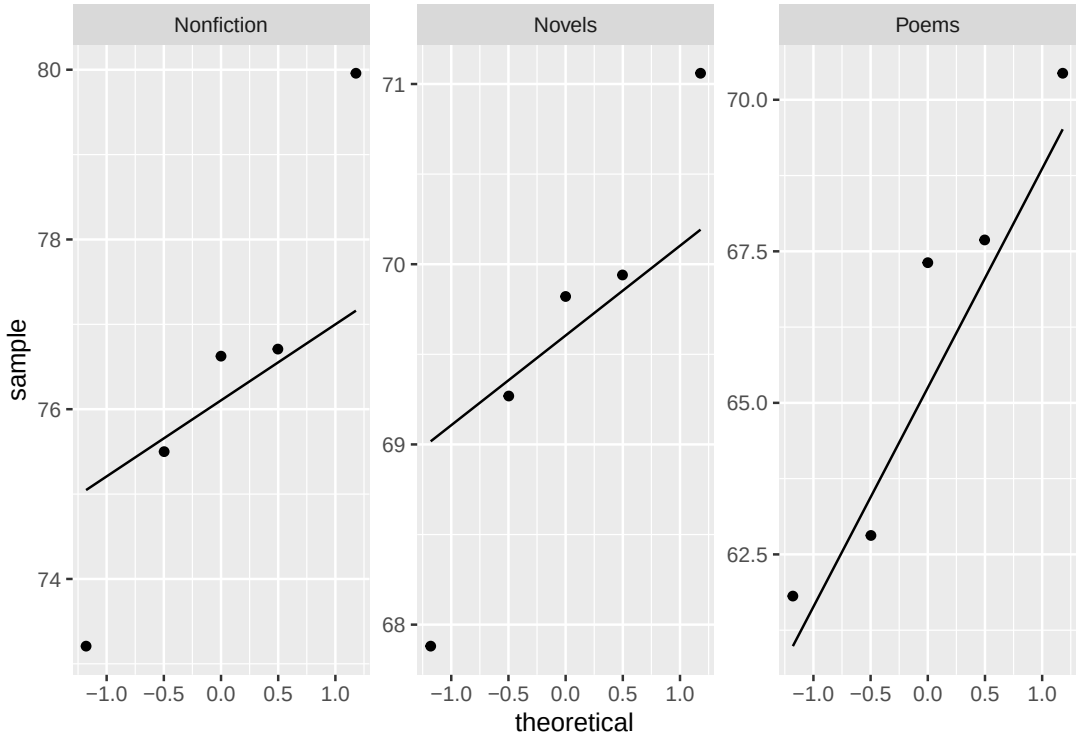
Now, for each of the bootstrap samples in `samples`, we want the mean of it, which might make you think of `map`, specifically `map_dbl`:

```
writers %>% nest(data=Age) %>%
  mutate(samples=map(data, ~rerun(5, sample(.$Age, replace = T)))) %>%
  unnest(samples) %>%
  mutate(mean=map_dbl(samples, ~mean(.)))
```

```
## # A tibble: 15 x 5
##    Type1 Type       data              samples     mean
##    <dbl> <chr>      <list>            <list>     <dbl>
##  1     1 Novels     <tibble [67 x 1]> <dbl [67]>  71.5
##  2     1 Novels     <tibble [67 x 1]> <dbl [67]>  74.7
##  3     1 Novels     <tibble [67 x 1]> <dbl [67]>  75.7
##  4     1 Novels     <tibble [67 x 1]> <dbl [67]>  71.9
##  5     1 Novels     <tibble [67 x 1]> <dbl [67]>  72.1
##  6     2 Poems      <tibble [32 x 1]> <dbl [32]>  59.6
##  7     2 Poems      <tibble [32 x 1]> <dbl [32]>  60.3
##  8     2 Poems      <tibble [32 x 1]> <dbl [32]>  63.1
##  9     2 Poems      <tibble [32 x 1]> <dbl [32]>  62.6
## 10     2 Poems      <tibble [32 x 1]> <dbl [32]>  58.8
## 11     3 Nonfiction <tibble [24 x 1]> <dbl [24]>  77.2
## 12     3 Nonfiction <tibble [24 x 1]> <dbl [24]>  77.7
## 13     3 Nonfiction <tibble [24 x 1]> <dbl [24]>  71.0
## 14     3 Nonfiction <tibble [24 x 1]> <dbl [24]>  72.2
## 15     3 Nonfiction <tibble [24 x 1]> <dbl [24]>  74.2
```

and then the last stage is to do facetted normal quantile plots of these, facetted by `Type`:

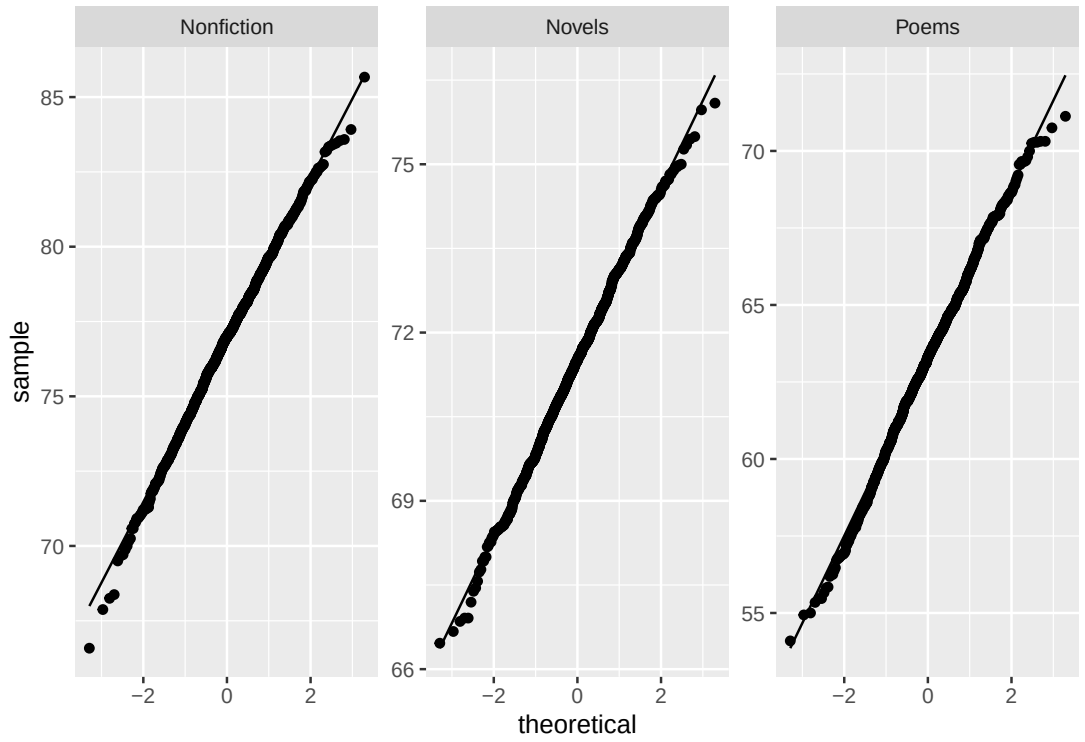
```
writers %>% nest(data=Age) %>%
  mutate(samples=map(data, ~rerun(5, sample(.$Age, replace = T)))) %>%
  unnest(samples) %>%
  mutate(mean=map_dbl(samples, ~mean(.))) %>%
  ggplot(aes(sample=mean)) + stat_qq() +
    stat_qq_line() + facet_wrap(~Type, scales="free")
```



You may breathe now. The hard work is done.

These plots, of course, are rather silly because they are means of only five bootstrap samples, instead of something like 1000. But our prototype works, so now all we have to do now is change 5 to 1000 and run it again:

```
writers %>% nest(data=Age) %>%
  mutate(samples=map(data, ~rerun(1000, sample(.$Age, replace = T)))) %>%
  unnest(samples) %>%
  mutate(mean=map_dbl(samples, ~mean(.))) %>%
  ggplot(aes(sample=mean)) + stat_qq() +
    stat_qq_line() + facet_wrap(~Type, scales="free")
```



Boom!

These three normal quantile plots are all acceptable, to my mind, although the Nonfiction one, with the two outliers and the smallest sample size, is still a tiny bit skewed to the left. Apart from that, the three sampling distributions of the sample means are close to normal, so our `aov` is much better than you might have thought from looking at the boxplots.

Notes

1. An Irish, that is to say, Gaelic, poet (see below), but a male one.

2. Gaelic is a language of Scotland and Ireland, and the culture of the people who speak it.

3. There might be differences *between* two things, but *among* three or more.

4. Like those Russian dolls.