Assignment 4

Due Tuesday Feb 11 at 11:59pm on Quercus

As before, the questions without solutions are an assignment: you need to do these questions yourself and hand them in (instructions below). The assignment is due on the date shown above. An assignment handed in after the deadline is late, and may or may not be accepted (see course outline). My solutions to the assignment questions will be available when everyone has handed in their assignment.

You are reminded that work handed in with your name on it must be *entirely your own work*.

Assignments are to be handed in on Quercus. See `https://www.utsc.utoronto.ca/~butler/c32/quercus1.nb.html` for instructions on handing in assignments in Quercus. Markers' comments and grades will be available there as well.

As ever, begin with this:

```
library(tidyverse)
```

1. Work through problems 9.1 and 9.2 in PASIAS. If you like, also work through problem 9.3. (This last problem is not immediately relevant to this assignment, but you have all the background to make sense of it, and doing so will give you some context for what is going on.)

   Hand the next one in.

2. The spreadsheet at `https://github.com/nxskok/datasets/blob/master/goodman%20-%20modern%20statistics/Canadian_Equity_Funds.xls` contains a random sample of equity funds published in 2005. We are interested in the asset value of each fund, shown in column C of the spreadsheet. These are in millions of dollars.

   (a) (3 marks) Read in and display (some of) this spreadsheet. To do this, you will need to follow a number of steps:

   - Go to the URL given above and click Download. This will download the spreadsheet to your computer.
   - If you are using `rstudio.cloud`, make sure you have it open in some project. In the file pane bottom right, click Upload and find the spreadsheet you downloaded. (For me, it goes into the Downloads folder by default.)
   - If you are running R Studio on your own computer, find where the file went (eg. by using `file.choose`).
   - Read the spreadsheet in directly. There is only one worksheet.

   Solution:

   After you have run steps 1 and whichever of 2 or 3 is appropriate for you, you'll need something like this:

```
f <- "Canadian_Equity_Funds.xls"
library(readxl)
funds0 <- read_excel(f, sheet=1)
funds0
```

```
## # A tibble: 60 x 10
##    Rating Name     Assets NAVPS   MER LoadFees Ret_1Mth Ret_3Mth Ret_1Yr Ret_3Yr
##     <dbl> <chr>     <dbl> <dbl> <dbl> <chr>       <dbl>    <dbl>   <dbl>   <dbl>
##  1      5 ABC Fun~  536.   18.6  2    N            -3.4      2.6    15.2    12.8
##  2      1 AIC Adv~ 1235.   68.9  2.45 O            -2.9     -1.9     2.1    -0.3
##  3      4 AIC PPC~   50.9  13.3  2.8  O            -1.6      1.5    15.7    NA
##  4      4 Assante~   59.4  14.4  0.43 O            -3.4      1.6    13.1     7.1
##  5      3 Assumpt~   NA    19.9  4.28 N            -3        1.5    14.8     5.3
##  6      3 BMO Spe~  400.   22.9  2.52 N            -3.1     -1.8    11.3     8.7
##  7      1 Canada ~    9.2  14.4  3.54 R            -1.6      0.6     2       1.4
##  8      3 CDA Com~   42.8  48.3  0.97 N            -1.7      2.3    10.7     6.1
##  9      2 CIBC Ca~  105.   19.6  2.62 N            -5.4     -0.5     8.8     4
## 10      3 CI Sign~    4.7  12.3  3.78 O            -2        0.7    14.4     5.6
## # ... with 50 more rows
```

The `sheet=1` is optional.

If you get stuck, open up the spreadsheet you downloaded in Excel (or whatever), save it as a `.csv`, and upload *that* to `rstudio.cloud`. You won't get full credit (in this part) for doing it this way, but you *will* be able to do the rest of the question, which is better than not being able to do it at all.

(b) (2 marks) Create a new data frame that is the old one without any rows containing missing values in the `Assets` column. (Hint: `drop_na`, and save the result.)

Solution:

Easier to do than to describe, really:

```
funds0 %>% drop_na(Assets) -> funds
```

You can save it back into the same data frame if you prefer. Don't forget the uppercase A on `Assets`!

I didn't tell you that you can supply a column name to `drop_na`, but you can soon discover this by searching. If you don't put *anything* inside `drop_na`, you'll drop all the rows with missing values on anything. This sometimes is what you want, but here we only care about missing values in `Assets`. If you drop *all* the missing values, you'll get rid of some perfectly good values in `Assets` that happen to be missing on something else that we don't care about:

```
funds0 %>% drop_na()
```

```
## # A tibble: 36 x 10
##    Rating Name     Assets NAVPS   MER LoadFees Ret_1Mth Ret_3Mth Ret_1Yr Ret_3Yr
##     <dbl> <chr>     <dbl> <dbl> <dbl> <chr>       <dbl>    <dbl>   <dbl>   <dbl>
##  1      5 ABC Fun~  536.   18.6  2    N            -3.4      2.6    15.2    12.8
##  2      1 AIC Adv~ 1235.   68.9  2.45 O            -2.9     -1.9     2.1    -0.3
##  3      4 Assante~   59.4  14.4  0.43 O            -3.4      1.6    13.1     7.1
##  4      3 BMO Spe~  400.   22.9  2.52 N            -3.1     -1.8    11.3     8.7
##  5      1 Canada ~    9.2  14.4  3.54 R            -1.6      0.6     2       1.4
##  6      3 CDA Com~   42.8  48.3  0.97 N            -1.7      2.3    10.7     6.1
##  7      2 CIBC Ca~  105.   19.6  2.62 N            -5.4     -0.5     8.8     4
##  8      3 CI Sign~    4.7  12.3  3.78 O            -2        0.7    14.4     5.6
##  9      4 Clarica~   53.9  15.0  3.69 N            -1.5      4.1    19.3     6
## 10      4 Clarica~  104.   15.2  3.35 N            -1.5      4.2    20       6.5
## # ... with 26 more rows
```

There are only 36 rows left now. We started with 60 and should have 54 (see below).

Extra: let's check for missing values before and after:

```
summary(funds0)
```

```
##      Rating          Name              Assets            NAVPS
##  Min.   :1.000   Length:60          Min.   :   0.30   Min.   :  4.15
##  1st Qu.:2.000   Class :character   1st Qu.:   8.35   1st Qu.: 11.40
##  Median :3.000   Mode  :character   Median :  21.70   Median : 13.56
##  Mean   :3.021                      Mean   : 120.42   Mean   : 30.69
##  3rd Qu.:4.000                      3rd Qu.:  65.22   3rd Qu.: 19.66
##  Max.   :5.000                      Max.   :1235.10   Max.   :303.19
##  NA's   :13                         NA's   :6
##       MER          LoadFees           Ret_1Mth         Ret_3Mth
##  Min.   :0.430   Length:60          Min.   :-5.600   Min.   :-5.300
##  1st Qu.:2.465   Class :character   1st Qu.:-2.925   1st Qu.: 0.325
##  Median :2.890   Mode  :character   Median :-1.700   Median : 1.500
##  Mean   :2.789                      Mean   :-1.968   Mean   : 1.257
##  3rd Qu.:3.277                      3rd Qu.:-1.150   3rd Qu.: 2.600
##  Max.   :5.630                      Max.   : 0.500   Max.   : 4.300
##
##     Ret_1Yr         Ret_3Yr
##  Min.   :-3.40   Min.   :-1.200
##  1st Qu.: 7.25   1st Qu.: 3.900
##  Median :11.90   Median : 6.000
##  Mean   :11.30   Mean   : 5.949
##  3rd Qu.:15.20   3rd Qu.: 7.900
##  Max.   :23.80   Max.   :18.100
##  NA's   :5       NA's   :19
```

There were six missing values in the `Assets` column (and more missing values elsewhere, but we don't care about those here.)

```
summary(funds)
```

```
##      Rating       Name               Assets            NAVPS
##  Min.   :1    Length:54          Min.   :   0.30   Min.   :  4.15
##  1st Qu.:2    Class :character   1st Qu.:   8.35   1st Qu.: 11.28
##  Median :3    Mode  :character   Median :  21.70   Median : 13.56
##  Mean   :3                       Mean   : 120.42   Mean   : 32.47
##  3rd Qu.:4                       3rd Qu.:  65.22   3rd Qu.: 19.79
##  Max.   :5                       Max.   :1235.10   Max.   :303.19
##  NA's   :12
##       MER           LoadFees            Ret_1Mth         Ret_3Mth
##  Min.   :0.430   Length:54          Min.   :-5.600   Min.   :-5.300
##  1st Qu.:2.435   Class :character   1st Qu.:-2.800   1st Qu.:-0.050
##  Median :2.760   Mode  :character   Median :-1.700   Median : 1.400
##  Mean   :2.736                      Mean   :-1.948   Mean   : 1.130
##  3rd Qu.:3.292                      3rd Qu.:-1.050   3rd Qu.: 2.575
##  Max.   :5.630                      Max.   : 0.500   Max.   : 4.200
##
##     Ret_1Yr         Ret_3Yr
##  Min.   :-3.40   Min.   :-1.200
##  1st Qu.: 7.00   1st Qu.: 3.250
##  Median :11.30   Median : 6.100
##  Mean   :11.11   Mean   : 6.031
##  3rd Qu.:15.20   3rd Qu.: 8.100
##  Max.   :23.80   Max.   :18.100
##  NA's   :5       NA's   :18
```

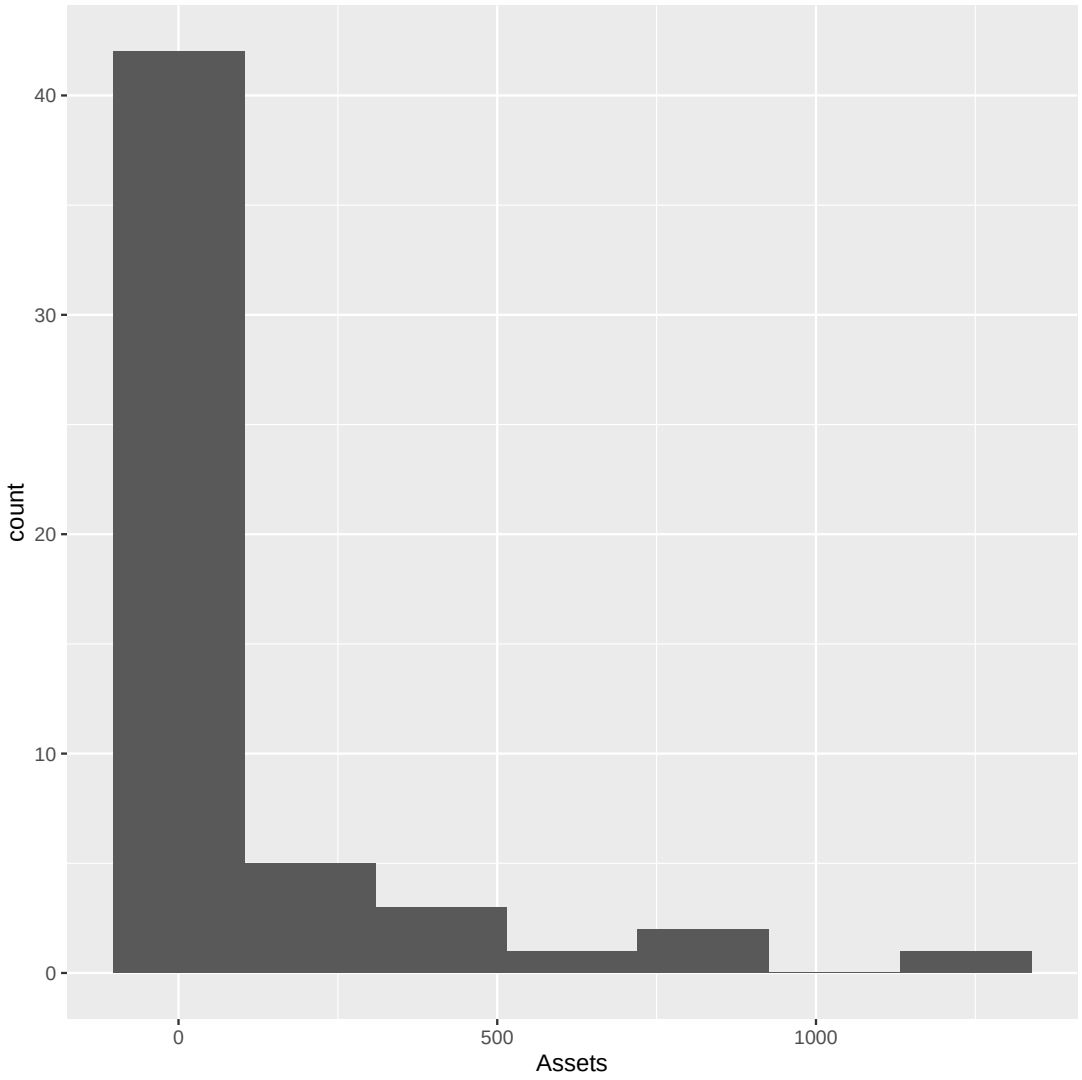
`Assets` no longer has any missing values, and the data frame has six rows fewer. Check.

(c) (3 marks) Make a suitable plot of the `Assets` values (ignoring the other columns). Why would you prefer a sign test for the median to a t-test for the mean? Explain briefly.

Solution:

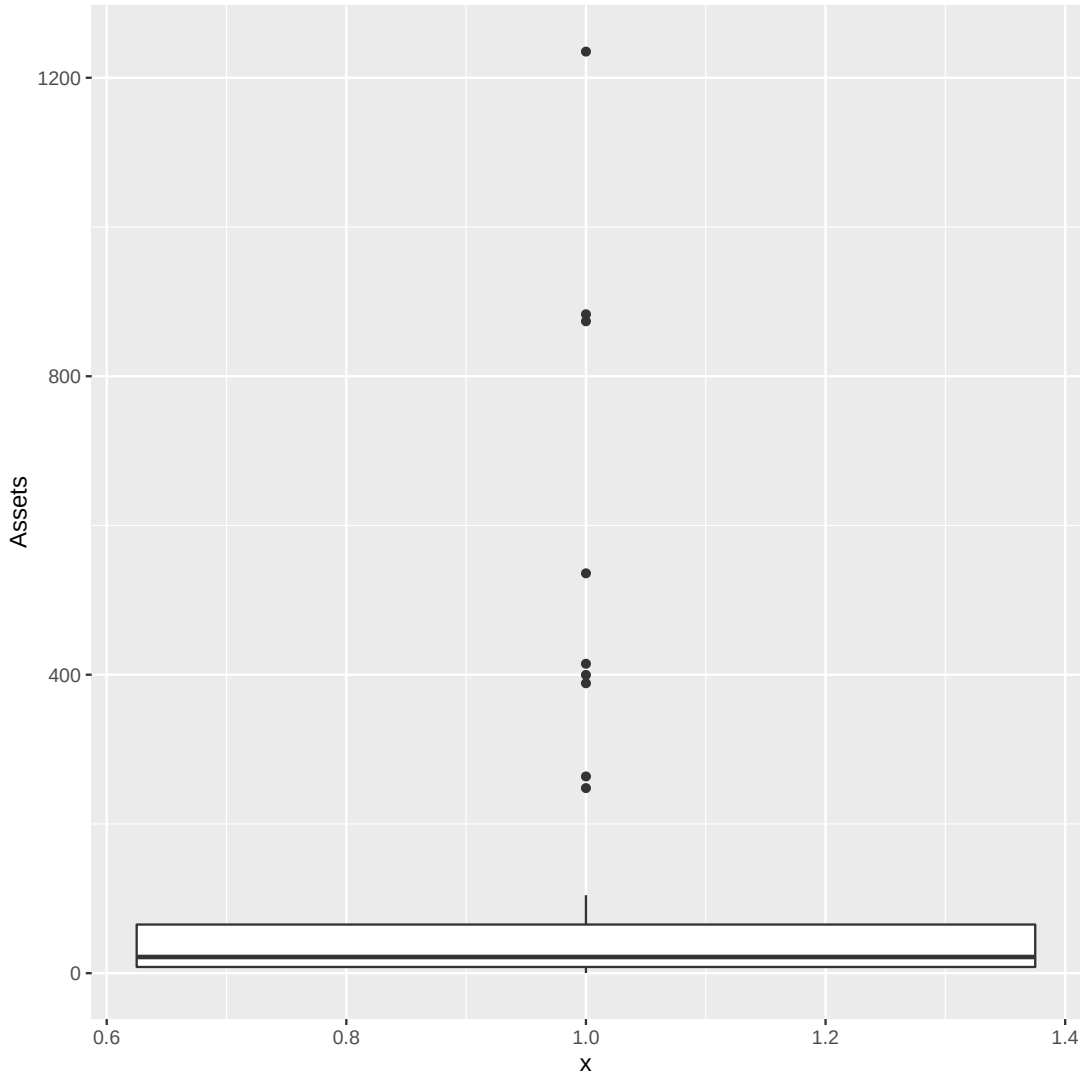
The usual for one variable: a histogram with a sensible number of bins, or a one-group boxplot, or even (given why we are looking at the plot) a normal quantile plot. Thus, one of these:

```
ggplot(funds, aes(x=Assets)) + geom_histogram(bins=7)
```



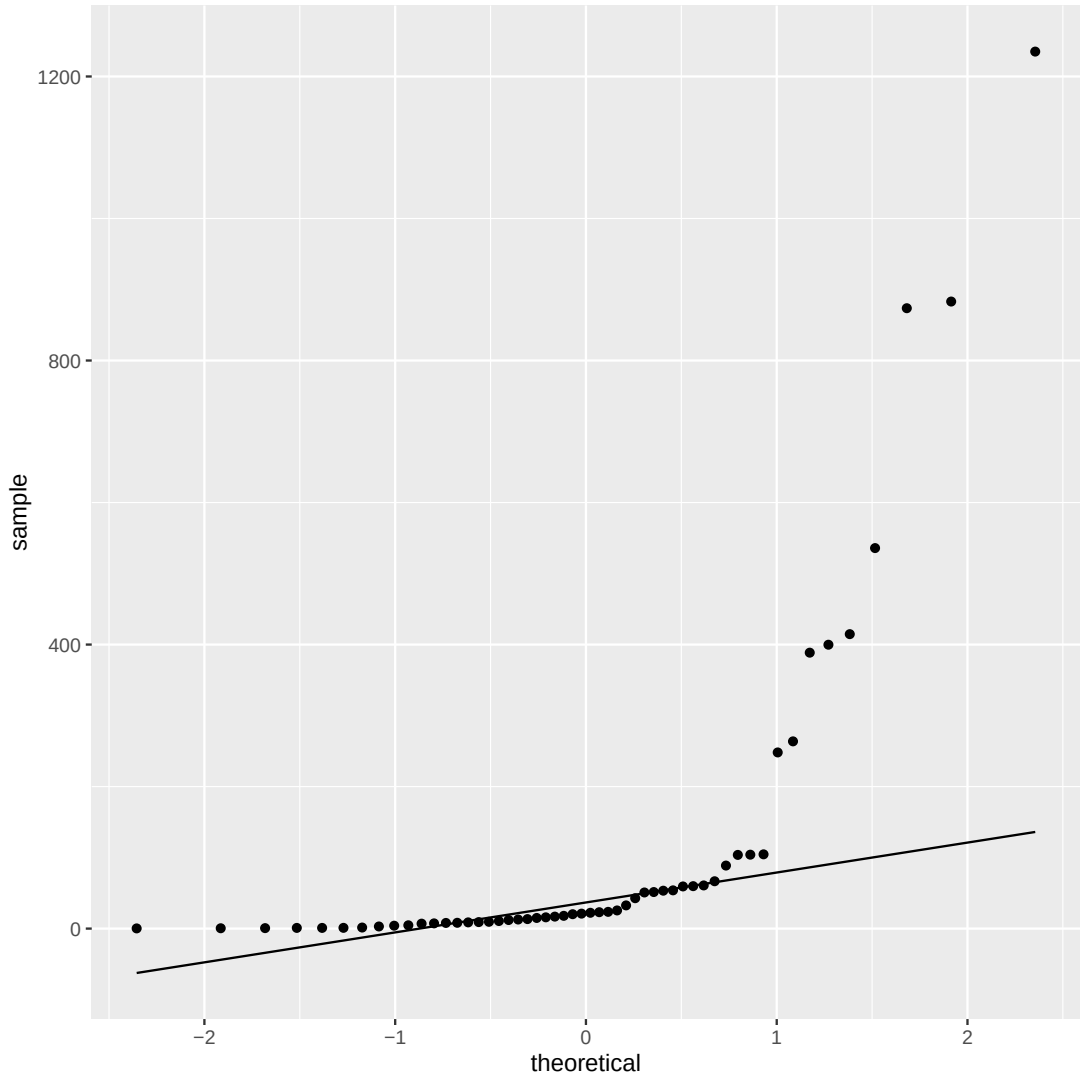
or

```
ggplot(funds, aes(x=1, y=Assets)) + geom_boxplot()
```



(outliers!) or

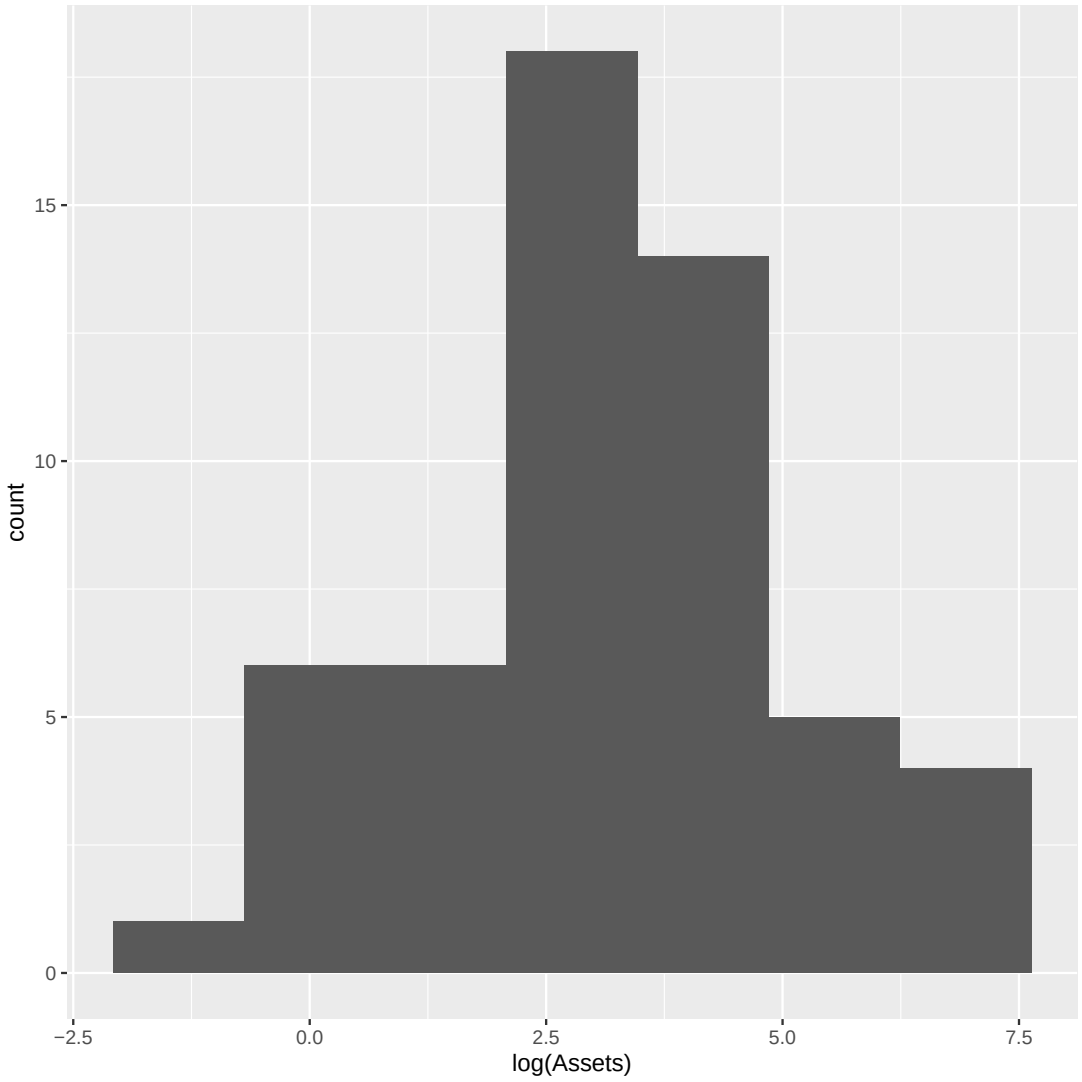
```
ggplot(funds, aes(sample=Assets)) + stat_qq() + stat_qq_line()
```



All of these are pointing towards a strongly right-skewed distribution. With this much skewness, even a sample size of 54 is not going to be large enough for the Central Limit Theorem to help us. Or, point out the many outliers, if you drew a boxplot. So, we should be looking at the median rather than the mean, and so a sign test should be better than a t-test.

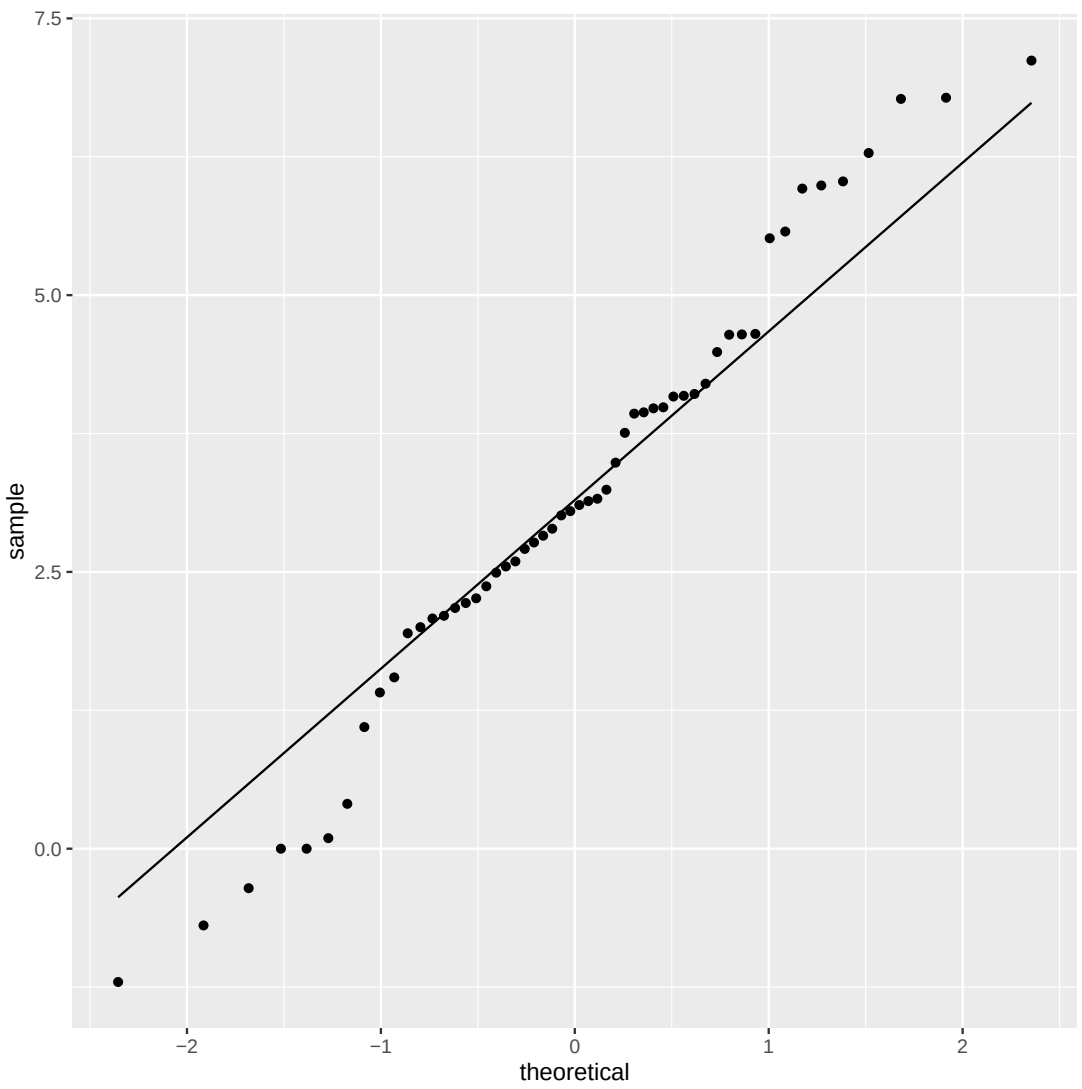
Extra: I would also consider taking logs of the `Assets` values, with a distribution like this. Does that look more normal?

```
ggplot(funds, aes(x=log(Assets))) + geom_histogram(bins=7)
```



It does, but I'm concerned that this is a little "peaky":

```
ggplot(funds, aes(sample=log(Assets))) + stat_qq() + stat_qq_line()
```



The tails are a little bit too long for a normal (the high values are too high and the low ones too low): that is, there is too much of a peak and too much in the tails, compared to a normal. I think, with this kind of sample size, I could live with this being "approximately normal".

But the asset values themselves are what we need to work with.

(d) (3 marks) A fund manager thinks that the median asset value (of the funds of which these are a sample) should be 55 (million dollars). Count (using R) the number of funds that have `Assets` above and below this value. Would you expect a sign test to reject a null median of 55 (million dollars), in favour of a lower median? Explain briefly.

Solution:

The simplest way is

```
funds %>% count(Assets>55)
```

```
## # A tibble: 2 x 2
##   `Assets > 55`     n
##   <lgl>         <int>
## 1 FALSE            37
## 2 TRUE             17
```

Or define a column saying whether the assets are bigger than 55 and count that:

```
funds %>% mutate(big=(Assets>55)) %>%
  count(big)
```

```
## # A tibble: 2 x 2
##   big       n
##   <lgl> <int>
## 1 FALSE    37
## 2 TRUE     17
```

or more aesthetically:

```
funds %>% mutate(comparison=ifelse(Assets>55, "big", "small")) %>%
  count(comparison)
```

```
## # A tibble: 2 x 2
##   comparison     n
##   <chr>      <int>
## 1 big           17
## 2 small         37
```

Or, for any of these, count the number of values *smaller* than 55. There are no values, as far as I can tell, exactly equal to 55; there are 17 above and 37 below.

To guess whether the sign test is going to reject a null median of 55 in favour of a lower one: well, if the median were something like 30, you'd expect more values below 55 and fewer above, compared to a 50–50 split. This is the kind of thing we observed. Make a call about whether you think this is an uneven enough split or not (you could successfully argue it either way, as long as your logic is sound). The key is that if the median is actually less than 55, you will observe *more* values below 55. (This may not be obvious to you, but try it with a true median and see what happens.)

My guess, for what it's worth, is that this is more unbalanced (in the right direction) than a 50–50 split, with this sample size.

(e) (3 marks) Run a sign test, using `smmr`, to see whether there is any evidence that that fund manager is wrong and that the median is actually lower than 55. What do you conclude, in the context of the data?

Solution:

This one is one-sided, so grab the right P-value:

```
library(smmr)
sign_test(funds, Assets, 55)
```

```
## $above_below
## below above
##    37    17
##
## $p_values
##   alternative    p_value
## 1       lower 0.00453667
## 2       upper 0.99808087
## 3   two-sided 0.00907334
```

If this gives you an error, make sure you have installed `smmr` first, *following the instructions in class* (and not via `install.packages`, which will not work).

The appropriate P-value is the first one, 0.0045, for the lower tail. This is smaller than 0.05, so reject the null median in favour of the alternative that it is less. We have evidence that the median assets of all funds (of which these are a sample) is less than 55 (million dollars).

Make sure you give the correct one-sided P-value. If you just say something like "the P-value is less than 0.05", you are *wrong*, because there are two P-values less than 0.05 and you didn't say which one is the appropriate one.

(f) (3 marks) Obtain a 95% confidence interval for the population median. Is the value 55 inside or outside your interval? Does it make sense to compare this with your hypothesis test? Explain briefly.

Solution:

```
ci_median(funds, Assets)
```

```
## [1] 12.80234 51.50594
```

The 95% confidence interval goes from 12.8 to 51.5 (million dollars), which you ought to state (with appropriate rounding off).

The null median 55 is outside this interval. As to whether it makes sense to compare this with your hypothesis test, there are a couple of ways you can go:

- We did a one-sided test, and a confidence interval is two-sided, so it doesn't make sense to think about whether 55 is inside the interval or not.
- To make a valid comparison, compare the CI with the *two-sided* test. This has a P-value 0.0091, which is also less than 0.05. So the two-sided test would also reject a null median of 55 (in favour of the alternative that the median is different from 55), and this is *consistent* with 55 being outside the 95% confidence interval.

Don't be sloppy here: if you are going to say that 55 being outside the interval is consistent with something, you have to say that it is consistent with the *two-sided* test (and say how). Comparing the one-sided test is only valid if you happen to have a one-sided confidence interval.

Extra: I said that the log-assets were approximately normal, so it would make sense to do a t-test for the log-assets (comparing the *mean* log-assets with the log of 55):

```
with(funds, t.test(log(Assets), mu=log(55), alternative="less"))
```

```
##
##  One Sample t-test
##
## data:  log(Assets)
## t = -3.1369, df = 53, p-value = 0.001393
## alternative hypothesis: true mean is less than 4.007333
## 95 percent confidence interval:
##      -Inf 3.615454
## sample estimates:
## mean of x
##   3.16696
```

This gives a one-sided test and one-sided interval. The (natural) log of 55 is almost exactly 4 (a coincidence):

```
log(55)
```

```
## [1] 4.007333
```

To get a proper confidence interval, do the two-sided test:

```
with(funds, t.test(log(Assets), mu=log(55)))
```

```
##
##  One Sample t-test
##
## data:  log(Assets)
## t = -3.1369, df = 53, p-value = 0.002786
## alternative hypothesis: true mean is not equal to 4.007333
## 95 percent confidence interval:
##  2.629624 3.704297
## sample estimates:
## mean of x
##   3.16696
```

In both cases, the log of 55 is outside the comparable interval (the one-sided one for the one-sided test, the two-sided one for the two-sided test). To get the last interval back onto the original scale, we can anti-log:

```
c(exp(2.63), exp(3.70))
```

```
## [1] 13.87377 40.44730
```

which is shorter than, but in the same ballpark as, the interval for the median. (Note that the P-values for the t-test on the logged data are a bit smaller than the ones for the sign test.)