University of Toronto Scarborough
Department of Computer and Mathematical Sciences
STAC32 (K. Butler), Midterm Exam
October 24, 2016

Aids allowed:

- My lecture slides
- Any notes that you have taken in this course
- Your assignments and feedback on them
- My assignment solutions
- The course R text
- The course SAS text
- Non-programmable, non-communicating calculator

Before you begin, complete the signature sheet, but sign it only when the invigilator collects it. The signature sheet shows that you were present at the exam.

This exam has 43 numbered pages of questions. Please check to see that you have all the pages.

In addition, you should have an additional booklet of output to refer to during the exam. Contact an invigilator if you do not have this.

Answer each question in the space provided (under the question). If you need more space, use the backs of the pages, but be sure to draw the marker's attention to where the rest of the answer may be found.

The maximum marks available for each part of each question are shown next to the question part. In addition, the total marks available for each page are shown at the bottom of the page, and also in the table on the next page.

When giving SAS code, you can provide code that runs either on the online version of SAS Studio, or on the version that runs on a virtual machine. Either version is acceptable.

The University of Toronto's Code of Behaviour on Academic Matters applies to all University of Toronto Scarborough students. The Code prohibits all forms of academic dishonesty including, but not limited to, cheating, plagiarism, and the use of unauthorized aids. Students violating the Code may be subject to penalties up to and including suspension or expulsion from the University.

Last name: _____

First name: _____

Student number: _____

For marker's use only:

Page	Points	Score
1	2	
2	3	
5	2	
7	6	
8	3	
14	5	
15	4	
17	5	
19	2	
21	4	
22	2	
24	8	
26	7	
27	2	
28	6	
31	3	
32	5	
34	3	
36	2	
37	3	
39	2	
40	2	
Total:	81	

You may assume that this has already been run:

```
library(tidyverse)
```

1. Coffee travel mugs are designed to keep coffee warm. A student decided to compare four different brands of travel mug. Each time, the student heated water to 90° C, poured it into the mug and sealed it. After 30 minutes, the student opened up the mug, measured the temperature of the water, and recorded the difference between the original temperature and the temperature after 30 minutes. Some of the data are shown in Figure 1 in the booklet of code and output.

   These data have been stored in a file `coffee.txt` in your current working folder in R Studio and also in SAS Studio.

   In later parts of this question, you do not need to repeat the code that you used in earlier parts. (For example, in doing computations on the data set, you may assume that the data set was already successfully read in.)

   (a) (2 marks) Give code to read the data into a data frame in R, and to display the whole data frame.

Solution: 2017 solution: recognize that the columns are separated by one space, including the headers, so what you need is `read_delim`:

```
coffee=read_delim("coffee.txt"," ")
```

```
## Parsed with column specification:
## cols(
##   cup = col_character(),
##   tempdiff = col_double()
## )
```

```
coffee
```

```
## # A tibble: 32 x 2
##          cup tempdiff
##        <chr>    <dbl>
##  1      SIGG     12.0
##  2      SIGG     16.0
##  3      SIGG      9.0
##  4      SIGG     23.0
##  5      SIGG     11.0
##  6      SIGG     20.5
##  7      SIGG     12.5
##  8      SIGG     20.5
##  9      SIGG     24.5
## 10 Starbucks     13.0
## # ... with 22 more rows
```

In 2016, this was also a nice gentle warmup:

```
coffee=read.table("coffee.txt",header=T)
coffee
```

```
##          cup tempdiff
## 1       SIGG     12.0
## 2       SIGG     16.0
## 3       SIGG      9.0
## 4       SIGG     23.0
## 5       SIGG     11.0
## 6       SIGG     20.5
## 7       SIGG     12.5
## 8       SIGG     20.5
## 9       SIGG     24.5
## 10 Starbucks     13.0
## 11 Starbucks      7.0
## 12 Starbucks      7.0
## 13 Starbucks     17.5
## 14 Starbucks     10.0
## 15 Starbucks     15.5
## 16 Starbucks      6.0
## 17 Starbucks      6.0
## 18     CUPPS      6.0
## 19     CUPPS      6.0
## 20     CUPPS     18.5
## 21     CUPPS     10.0
## 22     CUPPS     17.5
## 23     CUPPS     11.0
## 24     CUPPS      6.5
## 25    Nissan      2.0
## 26    Nissan      1.5
## 27    Nissan      2.0
## 28    Nissan      3.0
## 29    Nissan      0.0
## 30    Nissan      7.0
## 31    Nissan      0.5
## 32    Nissan      6.0
```

In 2017, you won't get full marks for `read.table`, since that is not what we did in class.

(b) (3 marks) Give code to read the data into a SAS data set, and to display the whole data set.

Solution: 2016: Pretty straightforward, but a couple of things to get right:

```
data coffee;
  infile '/home/ken/coffee.txt' firstobs=2;
  input cup $ tempdiff;
```

The variable names don't matter, as long as you use them below. Anything that looks like a username is OK (I'm not checking that), or the `/folders/myfolders/` thing. Did you remember `firstobs=2` to skip the header line in the data file? (Or you can say that the header line was removed first, but you have to say something like that if you have no `firstobs`.)

```
proc print;
```

Obs	cup	tempdiff
1	SIGG	12.0
2	SIGG	16.0
3	SIGG	9.0
4	SIGG	23.0
5	SIGG	11.0
6	SIGG	20.5
7	SIGG	12.5
8	SIGG	20.5
9	SIGG	24.5
10	Starbuck	13.0
11	Starbuck	7.0
12	Starbuck	7.0
13	Starbuck	17.5
14	Starbuck	10.0
15	Starbuck	15.5
16	Starbuck	6.0
17	Starbuck	6.0
18	CUPPS	6.0
19	CUPPS	6.0
20	CUPPS	18.5
21	CUPPS	10.0
22	CUPPS	17.5
23	CUPPS	11.0
24	CUPPS	6.5
25	Nissan	2.0
26	Nissan	1.5
27	Nissan	2.0
28	Nissan	3.0
29	Nissan	0.0
30	Nissan	7.0
31	Nissan	0.5
32	Nissan	6.0

2017 solution is actually easier than the 2016 one, since this is a standard `proc import`:

```
proc import
  datafile='/home/ken/coffee.txt'
  out=coffee
  dbms=dlm
  replace;
  getnames=yes;
  delimiter=' ';

proc print;
```

Obs	cup	tempdiff
1	SIGG	12
2	SIGG	16
3	SIGG	9
4	SIGG	23
5	SIGG	11
6	SIGG	20.5
7	SIGG	12.5
8	SIGG	20.5
9	SIGG	24.5
10	Starbucks	13
11	Starbucks	7
12	Starbucks	7
13	Starbucks	17.5
14	Starbucks	10
15	Starbucks	15.5
16	Starbucks	6
17	Starbucks	6
18	CUPPS	6
19	CUPPS	6
20	CUPPS	18.5
21	CUPPS	10
22	CUPPS	17.5
23	CUPPS	11
24	CUPPS	6.5
25	Nissan	2
26	Nissan	1.5
27	Nissan	2
28	Nissan	3
29	Nissan	0
30	Nissan	7
31	Nissan	0.5
32	Nissan	6

You might think I'm being picky about some of the things I'm deducting marks for, but if you are going to be dealing with data, you need to have enough attention to detail to give me code that will work. In the real world, code that doesn't work is no good, but if it's almost right it's easy to fix.

(c) (2 marks) Give SAS code to calculate the mean temperature difference for each brand of cup.

Solution:

```
proc means;
  var tempdiff;
  class cup;
```

with output

The MEANS Procedure

Analysis Variable : tempdiff

cup	N Obs	N	Mean	Std Dev	Minimum	Maximum
CUPPS	7	7	10.7857143	5.3139529	6.0000000	18.5000000
Nissan	8	8	2.7500000	2.5071327	0	7.0000000
SIGG	9	9	16.5555556	5.7142609	9.0000000	24.5000000
Starbucks	8	8	10.2500000	4.5512949	6.0000000	17.5000000

(d) (3 marks) Give R code to calculate the mean temperature difference for each brand of cup.

Solution: 2016: This one is most easily `aggregate`, making sure you get the inputs in the right order:

```
aggregate(tempdiff~cup,coffee,mean)
```

```
##         cup tempdiff
## 1     CUPPS 10.78571
## 2    Nissan  2.75000
## 3      SIGG 16.55556
## 4 Starbucks 10.25000
```

Or the 2017 way (which was also good in 2016):

```
coffee %>% group_by(cup) %>%
  summarize(m=mean(tempdiff))
```

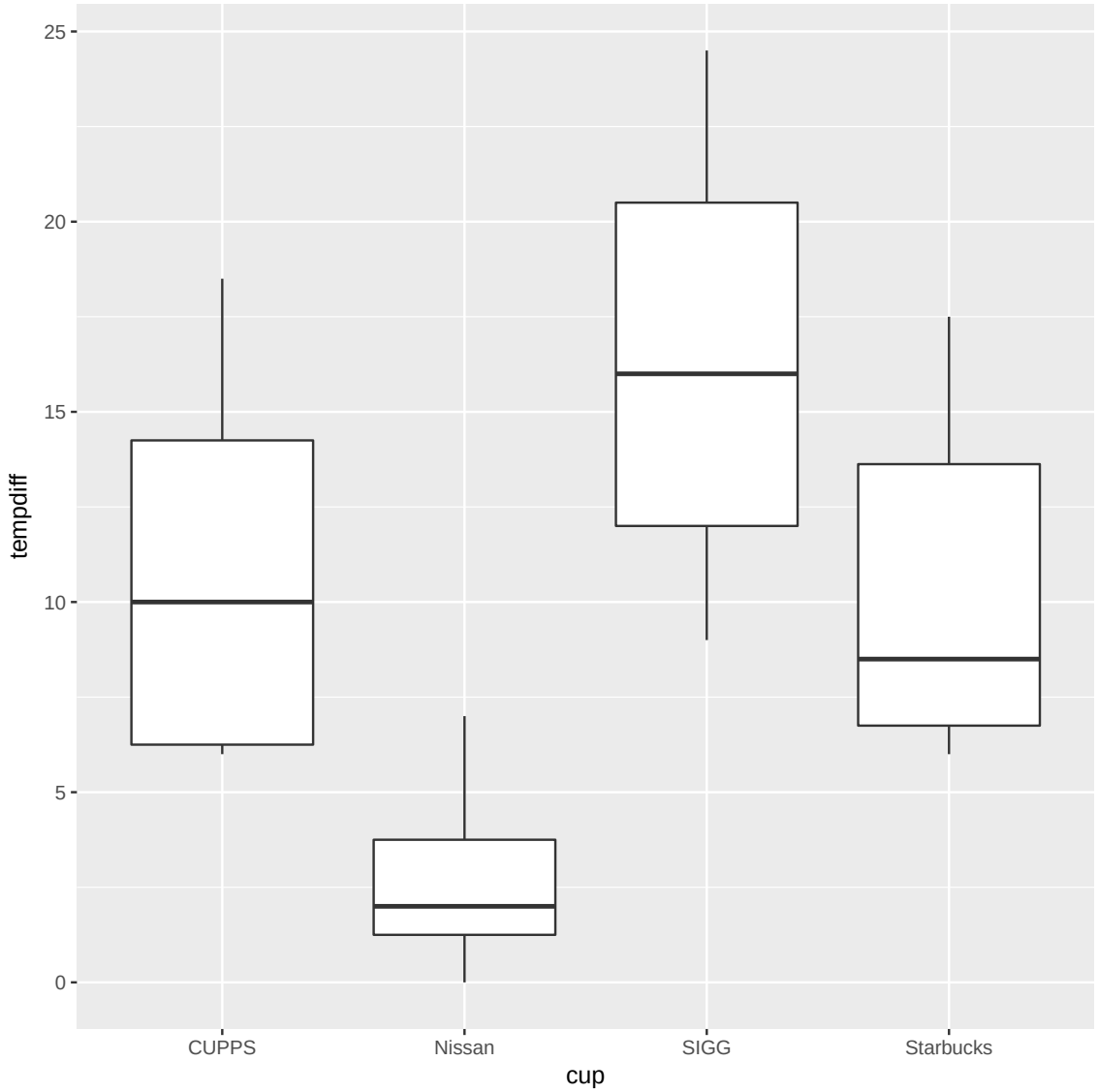
```
## # A tibble: 4 x 2
##         cup        m
##      <fctr>    <dbl>
## 1     CUPPS 10.78571
## 2    Nissan  2.75000
## 3      SIGG 16.55556
## 4 Starbucks 10.25000
```

If you go this way, you need *both* an appropriate `group_by` and a `summarize`. You *don't* need the `library(tidyverse)` first, since I said that this has already been loaded.

(e) (3 marks) Give R code to make `ggplot`-style side-by-side boxplots of temperature difference for each cup. (You may assume that `ggplot2` has already been loaded with `library(ggplot2)`.)

Solution: `ggplot` with an `aes`, then `geom_boxplot` with brackets:

```
ggplot(coffee,aes(x=cup,y=tempdiff))+geom_boxplot()
```



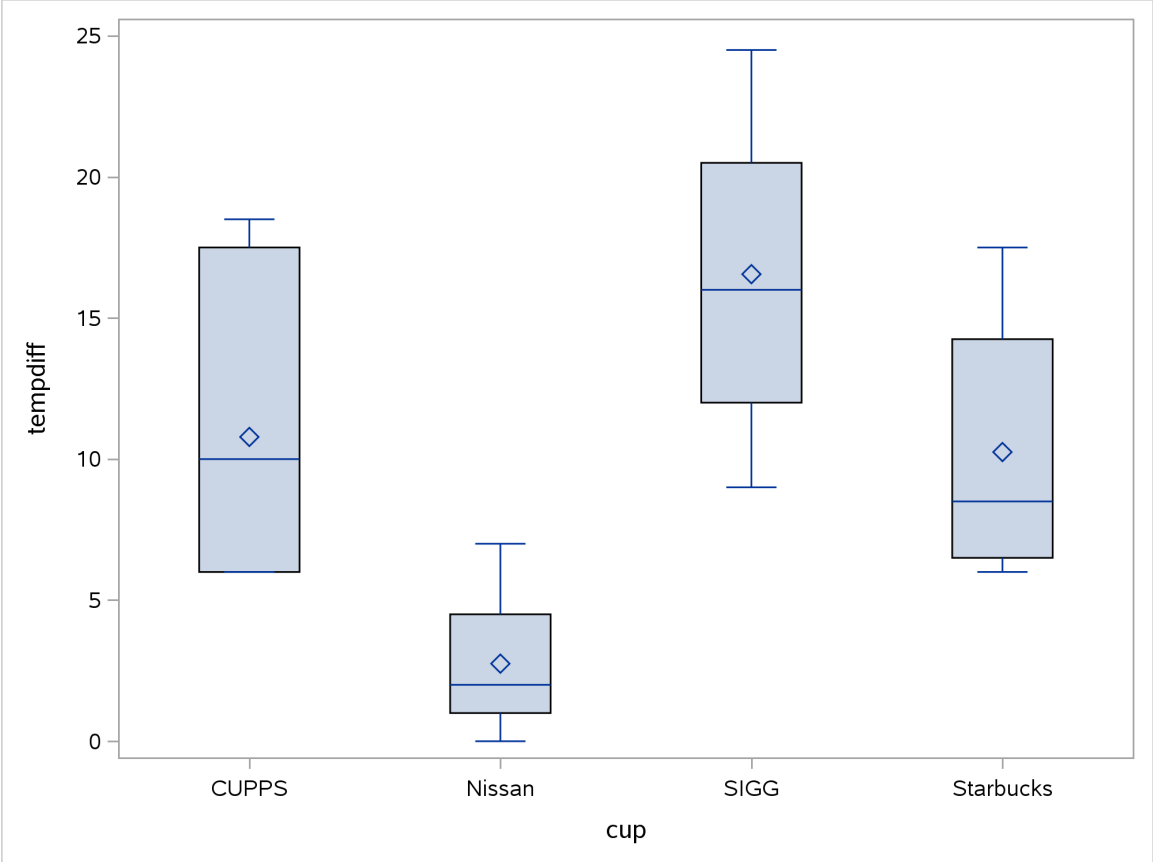
Notes: `cup` is treated as a factor (it was read in as text) so you don't need to convert it into one with `factor(cup)`. Doing so won't do any harm (so it won't cost you any points), but it's not needed. However, you *do* need the open-close brackets after `geom_boxplot()`; if you omit them, it won't work (and will give you a non-helpful message about "closures": I know, I've been there), so leaving them off will cost you a point. (This comes back to the attention-to-detail thing.)

I specifically asked for a `ggplot`-style boxplot, so a base-graphics `boxplot(tempdiff~cup)` does not answer the question. However, since it *does* do something useful (if you get it right), you get one point for a correct boxplot in this style.

(f) (3 marks) Give SAS code to make side-by-side boxplots of temperature difference for each cup.

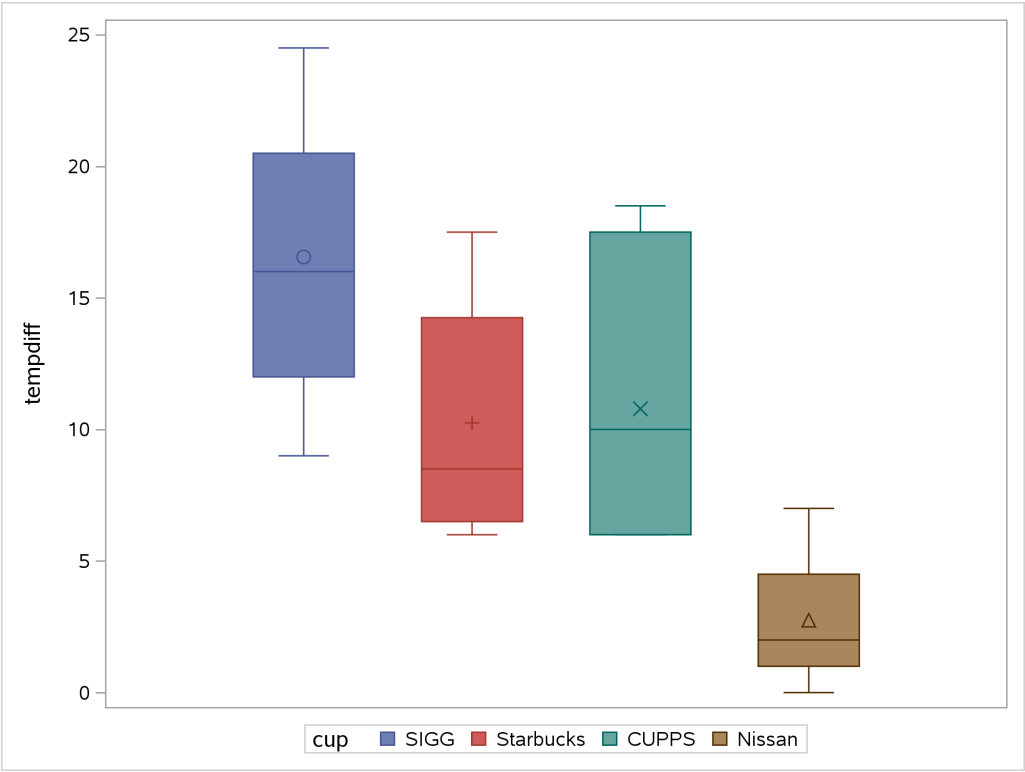
Solution: The way we did in class was `proc sgplot`:

```
proc sgplot;
  vbox tempdiff / category=cup;
```


25
20
15
10
5
0
tempdiff
CUPPS
Nissan
SIGG
Starbucks
cup

The thing after the slash has to be `category` or `group`. `class` doesn't work. `group` does something different:

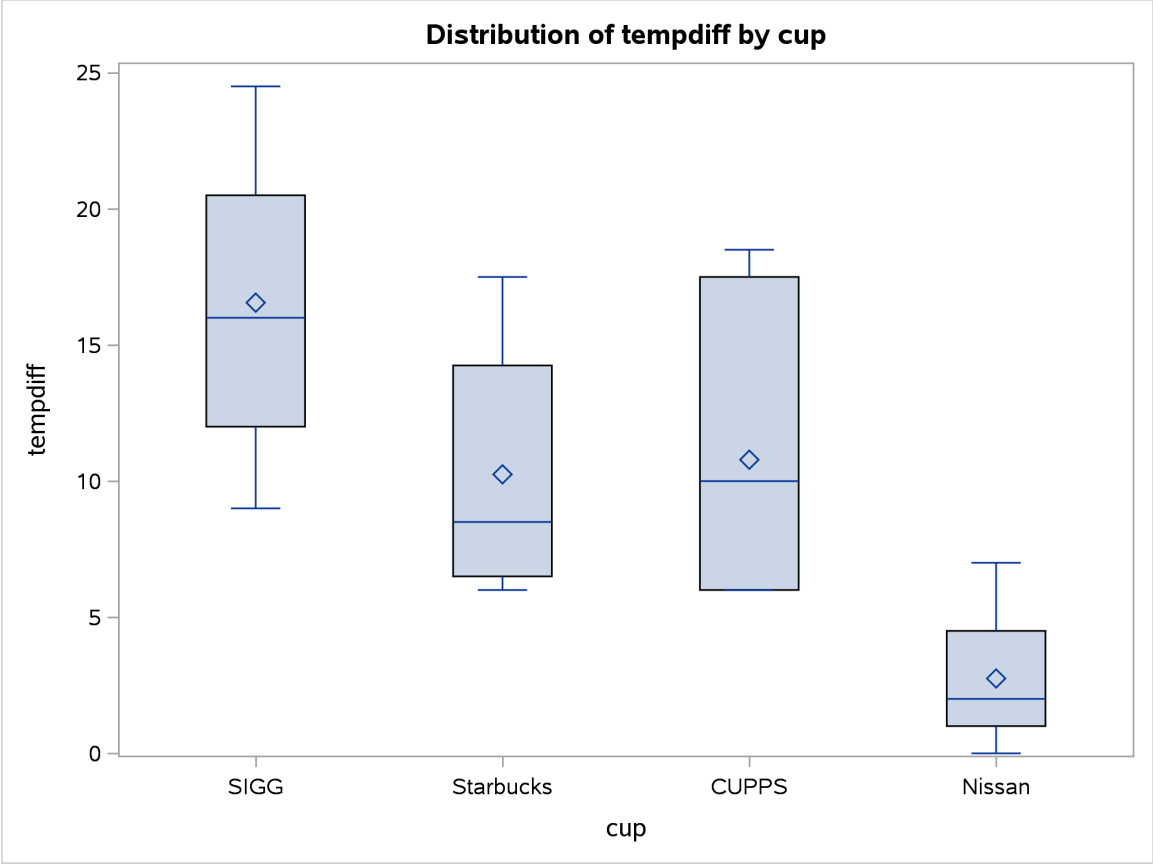
```
proc sgplot;
  vbox tempdiff / group=cup;
```


25
20
15
10
5
0
tempdiff
cup
SIGG
Starbucks
CUPPS
Nissan

This is good (you might even like it better).

If you also figured out (on your own) that you could use `proc boxplot`, that's good too, but you have to get it right:

```
proc boxplot;
  plot tempdiff*cup;
```



This one seems to list the cups in the order that they are in the data, rather than alphabetically.

2. A car magazine obtained data on a large number of models of American and Japanese cars. The magazine was interested in testing whether the gas mileage, measured in miles per US gallon of gasoline, was on average better for Japanese cars than American ones. The structure of the data is shown in Figure 2 in the booklet of code and output.

   (a) (3 marks) Give R code to carry out a suitable t-test that will address the magazine's interest. Briefly justify your choice of alternative hypothesis. You may assume that the data have already been read into a data frame called `carmpg`.

Solution: I need to read in the data first (you don't):

```
carmpg=read_delim("carmpg.txt"," ")
```

```
## Parsed with column specification:
## cols(
##   row_number = col_integer(),
##   country = col_character(),
##   mpg = col_integer()
## )
```

```
carmpg
```

```
## # A tibble: 328 x 3
##    row_number country   mpg
##         <int>   <chr> <int>
##  1          1      us    18
##  2          2      us    15
##  3          3      us    18
##  4          4      us    16
##  5          5      us    17
##  6          6      us    15
##  7          7      us    14
##  8          8      us    14
##  9          9      us    14
## 10         10      us    15
## # ... with 318 more rows
```

and now I can do my t-test. It's a two-sample t-test (since there is no pairing-up of the cars, and in any case there are a different number of each). We have to get the alternative hypothesis right: the two countries are listed as `japan` and `us` in that order, and a *high* gas mileage on this scale is better, so Japan has to be greater than the US:

```
t.test(mpg~country,data=carmpg,alternative="greater")
```

This is the actual code I ran to get Figure 3.

The word in the `alternative` has to be `"greater"`. "More" or "higher" or anything like that will not work.

   (b) (2 marks) The results of my t-test are shown in Figure 3. What do you conclude, in the context of the data?

Solution: The null hypothesis says that the mean gas mileage for cars from the two countries is the same. The P-value is extremely small (less than 0.0000000000000002, ie. basically zero), so we reject the null hypothesis, in favour of the alternative that the mean gas mileage for Japanese cars is better

(greater) than for American cars. In other words, the magazine's supposition was proved correct.

I think you know by now that "reject H_0" is a long way from a complete answer. I need to see something about gas mileages of cars: that Japanese cars have better mean gas mileage than American ones. (If you decided that your test should be two-sided above, and you followed through with a conclusion like "Japanese and American cars differ in mean gas mileage" that is correctly two-sided, you'll get full marks for this part.)

Your conclusion should be based on what your alternative hypothesis says. The sample mean gas mileages (from the data) could be used to *support* the non-surprisingness of the conclusion, but not to get the conclusion in the first place. That should come from the small P-value and alternative hypothesis.

(c) (3 marks) Figure 4 shows side-by-side boxplots of the gas mileages for the cars from each country. Explain briefly why we might initially have concerns about the validity of the t-test, but explain why these concerns are not serious here.

Solution: The boxplots appear to have upper outliers and are slightly right-skewed. This would usually indicate some (mild) concern with the t-test, which assumes normal data, but if you look back at Figure 2, you'll see that we have a total of over 300 observations (almost 250 US cars and almost 80 Japanese ones), so that the sample sizes are very large. Thus this t-test can actually handle a *lot* of non-normality in the data, more than is shown here.

I wanted to see three things: t-test assumes normality, normality fails because of outliers/skewness, and large sample size so that normality is not really important. (If we had been doing Mood's median test, the outliers would not have mattered at all. It was the fact that the t-test assumes normality that was the reason we were looking for outliers in the first place.)

I think "sample sizes are large" is a more insightful answer than "skewness is not bad", because this skewness might be serious enough to cause trouble if we had, say, only 10 cars from each country. You could also sell me on the P-value being so small (and thus the conclusion being so clear) that even if the t-test's P-value was off from the truth by a bit, the conclusion wouldn't change. There were one or two other things I could be swayed by.

There were some really good answers here.

(d) (1 mark) How would you tell *how much* better the gas mileages are for Japanese cars than American ones?

Solution: A very brief answer here is fine: calculate a confidence interval. That's all I need. (The implication is "all Japanese and American cars", not just the ones that happened to appear in the sample. If you said "about 10 mpg better" by comparing the sample means, this was the kind of direction you were going, but you needed to summarize the uncertainty as well, which a CI would do.)

The actual process of calculating a confidence interval when you have done a one-sided test is addressed in the question about the police trainees, so I didn't need to ask it here. The answer to that here is the same as there: do your test again, but do it two-sided, and ignore the P-value:

```
t.test(mpg~country,data=carmpg)

##
##  Welch Two Sample t-test
##
## data:  mpg by country
## t = 12.946, df = 136.87, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   8.757621 11.915248
## sample estimates:
## mean in group japan    mean in group us
##            30.48101            20.14458
```

Between 8.76 and 11.92 miles per gallon. This is a short interval because we have a lot of data. This confidence interval being so far from zero is the reason the P-value came out so small.

I'm not so sure about the interpretation of a one-sided CI (for example, the one in Figure 3). I think you have to say something like "the difference is at least 9.01". This puts all the missing 5% of the confidence at one end (the bottom end), so that it goes up further but not down as far. I'm not a fan of these one-sided intervals; to me, a confidence interval is inherently a two-sided thing, so I would get one by running a two-sided test.

If you really wanted a one-sided interval, for example for the population median (based on the sign test), you would run a one-sided sign test repeatedly for various different null medians, and take the values not rejected by your one-sided test as your interval (which would go all the way off to infinity at one end).

3. Police trainees sit in a darkened room facing a projector screen. Ten different licence plates are projected on the screen, one at a time, for 5 seconds each, separated by 15-second intervals. After the last 15-second interval, the lights are turned on, and the trainees are asked to write down as many of the 10 licence plate numbers as they can (in any order at all). The number of licence plates correctly recalled is recorded.

   Fifteen of the trainees (randomly chosen) went through the above procedure, and were then given a week-long memory training course. After that, they were re-tested. The number of licence plates correctly recalled before and after training is shown in Figure 5.

   (a) (2 marks) Are these data matched pairs or two separate samples? Explain briefly.

Solution: There are 15 trainees, each of whom has a before measurement and an after measurement, paired up, because they are the same trainees. So this is matched pairs. (In fact, before-and-after is one of the classic situations where a matched-pairs analysis is done.)

The word I was looking for in your answer was "same". If that was there, your answer was probably right.

   (b) (3 marks) The data have been read into a SAS data set with two variables, called `before` and `after`. Give code to run a suitable t-test to determine whether the training program is *helpful*.

Solution: First, I need to read in the data (which you don't):

```
proc import
  datafile='/home/ken/police.txt'
  out=police
  dbms=dlm
  replace;
  getnames=yes;
  delimiter=' ';
```

Stop and think before you write any code! What does "helpful" mean here? A higher score is better, so the mean "after" score needs to be *higher* than the mean "before" score. This has definitely got to be a one-sided test, not the two-sided one that many people wrote down.

Let's phrase everything with "after" first and "before" second (since that's alphabetical order even if it isn't the order that makes sense):

```
proc ttest sides=U;
  paired after*before;
```

```
           N        Mean     Std Dev     Std Err     Minimum     Maximum

          15      1.5333      2.1996      0.5679     -3.0000      5.0000
    Mean       95% CL Mean        Std Dev      95% CL Std Dev

  1.5333      0.5330  Infty        2.1996      1.6104   3.4689
                       DF    t Value    Pr > t

                       14       2.70    0.0086
```

Any of `sides`, `side` and `sided` work and are equally acceptable.

Look back at the data in Figure 5, and you'll see that the "after" scores are typically higher than the "before" scores. We are trying to prove that the (population) mean `after` is higher than `before`, so the P-value should at least be fairly small, which it is.

The way that makes logical sense is to put "before" first and "after" second. In the determination of what alternative hypothesis SAS uses, does it use the order in `paired`, or alphabetical order? Let's find out:

```
proc ttest sides=U;
  paired before*after;
```

```
          N        Mean     Std Dev     Std Err     Minimum     Maximum

         15     -1.5333      2.1996      0.5679     -5.0000      3.0000
            Mean        95% CL Mean         Std Dev       95% CL Std Dev

          -1.5333     -2.5336  Infty         2.1996       1.6104   3.4689
                                DF    t Value    Pr > t

                                14      -2.70    0.9914
```

```
proc ttest sides=L;
  paired before*after;
```

N	Mean	Std Dev	Std Err	Minimum	Maximum
15	-1.5333	2.1996	0.5679	-5.0000	3.0000

Mean	95% CL Mean		Std Dev	95% CL Std Dev	
-1.5333	-Infty	-0.5330	2.1996	1.6104	3.4689

DF	t Value	Pr < t
14	-2.70	0.0086

The second one is the result that makes sense, so *alphabetical* order is what SAS must use. So if you put `before` first and `after` second on the `paired` line, your alternative *still* has to work with the variables in alphabetical order. That is to say, the order on `paired` is irrelevant.

I think my first way is safest, so it is your best strategy. If you are not sure, put everything in alphabetical (rather than logical) order. Or, *explaining your confusion*, rather than just guessing, is the way to get my sympathy (and maybe an extra mark if there are any to be had). You might even convince me to give full marks here by something like giving both of my second and third alternatives and saying something like "I expect the P-value to be small, so I would try both and take the one that gives the smaller P-value".

You can also try to work with differences, after minus before. The right way to calculate them is on a data step (though I was prepared to cut you some slack on this), like this:

```
data police2;
  set police;
  diff=after-before;
```

Then you do a one-sample t-test on the differences, using a one-sided *upper*-tail test (this would be `alternative="greater"` in R):

```
proc ttest sides=U;
  var diff;
```

with output

N	Mean	Std Dev	Std Err	Minimum	Maximum
15	1.5333	2.1996	0.5679	-3.0000	5.0000

Mean	95% CL Mean		Std Dev	95% CL Std Dev	
1.5333	0.5330	Infty	2.1996	1.6104	3.4689

DF	t Value	Pr > t
14	2.70	0.0086

As you see, the answer is identical to the others.

(c) (2 marks) I ran your code from above and obtained the results shown in Figure 6. What do you

conclude from this analysis, in terms that would be helpful to the people running the police training program?

Solution: The null hypothesis here is that mean recall is the same before and after the memory training, with the alternative being that recall is *better* after training (one-sided). We reject the null in favour of the alternative (small P-value), so we conclude that the training *is* helpful in recalling licence plates. (Thus, if we had to make a recommendation, we would say that memory training should be used for all trainees.)

Now, this was actually rather a pain to mark, because the right answer to this part depends on whether you did a one-sided or two-sided test in the previous part. If your test was two-sided, you are only entitled to conclude that the memory training makes a difference, which could be a positive or a negative one: the training could be helpful *or* harmful, and the two-sided test doesn't tell you which. If you want to argue that the training is helpful, you have to do something more: for example, looking at the confidence interval for the mean difference and noting it contains positive values. Or you can observe that the mean difference is positive, so that's the way the data are pointing. So if you lost a mark here, it may well be because your test was two-sided, and you unwarrantedly drew a one-sided conclusion.

There were an awful lot of people who noted that the P-value was small and then *failed* to reject the null hypothesis (and thus concluded that the training made no difference). You are in an upper-level Statistics class. This is a mistake you *should not* be making.

(d) (1 mark) The confidence interval shown in Figure 6 makes no sense. How would you change your code to obtain a sensible 95% confidence interval?

Solution: The confidence interval given makes no sense because it is one-sided, like the test. So we need to run a two-sided test (as well): that is, take out the `sides` statement:

```
proc ttest;
  paired before*after;
```

N	Mean	Std Dev	Std Err	Minimum	Maximum
15	-1.5333	2.1996	0.5679	-5.0000	3.0000

Mean	95% CL Mean		Std Dev	95% CL Std Dev	
-1.5333	-2.7514	-0.3153	2.1996	1.6104	3.4689

DF	t Value	Pr > |t|
14	-2.70	0.0173

This time we *don't* look at the P-value, but we *do* look at the confidence interval: with 95% confidence, trainees get between 0.3 and 2.75 *more* identifications correct after the memory training than before. This seems like potentially a worthwhile gain.

The test and CI are telling us different but complementary things: the test is saying that average correct identifications *are greater* after the memory training than before, and the confidence interval is saying something about *how much greater* they are. As ever, to get a more precise (shorter) interval, we need a larger sample size: here, we need to give more trainees the memory training. We could think of the investigation that was actually done as a "pilot experiment", and now that we have evidence that the memory training helps, we could do a larger study to find out how much it helps.

Now, if you thought (b) was two-sided, I was left in a bind about how to mark this one. I decided that doing something different from (b) that would work was OK: for example, calculating the differences, and doing a one-sample t-test that their mean was zero (two-sided), or adding `sides=2` to your code from (b). (They don't actually make any difference, but show the right thought process, so I was happy with those.) I decided that just adding `alpha=0.05` to your code from (b) didn't show quite enough insight, so that didn't get the mark. Also, if working with differences, you needed to tell me what `diff` actually was in this case. Strictly, as above, differences should be calculated in a `data` step, but as above, I was willing to be relaxed about this.

(e) (3 marks) Look at Figures 7, 8, and 9. For *each* Figure, choose *one* of these two things: (i) if it is relevant to the trustworthiness of the test, describe what it tells you; (ii) if it is not relevant, explain (very) briefly why not.

Solution: This is a matched-pairs t-test, so the thing that needs to be normal is the *differences* between before and after. Thus Figures 7 and 8 are irrelevant. We only need to look at Figure 9. This is a normal quantile plot. The assumption we need to make is that the differences are approximately normal, that is, that they more or less follow the line. I think they do; I certainly don't see any obvious way that they don't. So I think my matched-pairs t-test can be trusted (and therefore that our conclusion, that the memory training is helpful, is a solid one).

I think a fair few people misunderstood the language. The "relevant" part means "do we need to look at this figure to make a decision about whether to believe the t-test?" Whether Fig 7 or Fig 8 are normal

matters not at all for this test, because the only thing that needs to be normal is the differences in Figure 9. Also, you are meant to be applied statisticians: it is not enough to say that the Figures enable you to assess normality. You actually have to do it and make a call! I think that Figure 9 is normal enough, but you are free to disagree (if you have a reason to), from which it follows that the matched pairs t-test should *not* be trusted.

You might be able to make the case that Figs 7 and 8 are normal, and 9 is the difference between two (approx) normals, and so it should also be (approx) normal, without even having to look at it.

(f) (2 marks) A police officer who had some training in statistics a long time ago says "but these are numbers of licence plates identified out of 10. They must be binomial, and cannot be normal. Therefore you have no business doing a t-test." How would you respond to this?

Solution: My response would probably begin with the words "With respect, officer". (When talking to a public official, "with respect" means about the same as, when you are talking to your friends, "I am about to tell you that you are an idiot", except that telling a police officer "you are an idiot" is a very fast way to get yourself an appearance in court. In other words, "with respect" actually means "without respect"!)

The issue here is that we do not need *exact* normality to be justified in running a t-test. Approximate normality is good enough. Even if our data are binomial, the differences can be (and here are) approximately normal. A difference could be anything from -10 to $+10$, and the differences we have span a fair fraction of that interval, so I think this is all right. And, indeed, the normal quantile plot said that they were approximately normal.

If the trainees tended to get almost all of the identifications right, or almost all of them wrong, the before and after figures would both be near the extremes 0 or 10, and the differences would all be close to 0. That would be one of the cases where the binomial is not normal-like. Think of the normal approximation to the binomial: this works best, for any given n, if p is somewhere near 0.5, as appears to be the case here. When p is close to 0 or 1 (as would be the case if the trainees tended to identify none or all of the licence plates), you will tend to get almost all failures or almost all successes, and the distribution will be very discrete. As I said, though, that didn't happen here.

The discreteness shows up on Figure 9 as those horizontal "streaks" of points (you see they correspond to trainees who had the same (integer) difference). But the "streaks" follow the line about as well as you could expect, so I'm good.

Any kind of sensible discussion here is good with me. There are a lot of ways you could say what I said, or something of equal value. Hiding in many of them are the idea that "binomial can be approximately normal". If you don't like the idea of doing the t-test here, you can compare the number of licence plates correctly identified before and after by a sign test (and ask if the number of trainees that identified more plates correctly after the training was higher than you'd expect, if the training were actually ineffective), or also a randomization test, based on swapping the before and after results for a randomly chosen set of trainees, repeatedly. Or you can contest the binomial assertion. What I meant is that each licence plate is either identified or not, but some of them might be harder to remember than others (which would mean that the total number of plates identified by a trainee is not binomial).

The central limit theorem (normal approximation to binomial) is a bit shaky here, because there were only 10 licence plates to identify (or not) before and after training. But things are strengthened a fair bit by taking the difference between before and after (the difference between two near-normals is nearer-normal, informally).

So, you see that you have a lot of possibilities. I also note that at least some of you know how to talk

to a police officer, judging by the presence of the word "sir" in your answers!

4. The IRS (Internal Revenue Service) is an American government organization whose purpose is to make sure people pay the tax they owe. The IRS says it should take the average taxpayer 150 minutes to fill out their tax forms. A journalist has been hearing about people taking much longer to complete their tax forms, and so arranges a study in which 30 randomly-chosen taxpayers record the number of minutes it takes them to complete their tax forms. The journalist's aim is to show that it actually takes longer than 150 minutes on average to complete the forms. The data are shown in Figure 10.

   (a) (2 marks) A histogram of the data is shown in Figure 11. Note that the `bins=10` in the code means "choose the bin width so that are ten (possibly empty) classes of equal width that the values are classified into". Why is the median a the better measure of centre (location) than the mean for these data? Explain briefly.

Solution: The data are skewed to the right. (The peak is to the left and there is a long right tail.) Therefore the mean will be pulled upward by the skew, and so the *median* would be a better measure of centre (it will not get affected by the small number of high values).

I was after two things: skewness (or outliers or both), and the fact that these will affect the mean but not the median. Another way to attack this is to say that the mean applies to (approximately) normal data, which these are not, and therefore we should use the median instead.

It was also possible to convince me to give you two points by arguing eloquently that the median is better for skewed data (but it needed to be eloquent, because saying what would happen to the mean is better).

Times to complete some task have a lower limit of zero, and distributions tend to be skewed away from limits, so it's so surprise that these data are skewed to the right.

   (b) (2 marks) Explain briefly why a sign test would be better than a t-test to assess the journalist's claim. (There may be some overlap with the previous part. If you have made the relevant points somewhere within these two parts, I am happy.)

Solution: The sign test is a test of the median (which we said we wanted to use above), while the t-test is a test of the mean. Or, you can say that the t-test requires approximately normal data, which we do not have here. Either of these observations will do; the previous part was meant to be a hint towards the first one.

If you already said something like this in (a), you can use it again for (b), or draw my attention to what you said in (a). That's what I meant by "overlap".

   (c) (2 marks) Write down the null and alternative hypotheses for your sign test, defining any symbols you use.

Solution: I like to let M stand for the population median (completion time in minutes), but you can use any symbol you like. The IRS says that the population median should be 150 minutes (null hypothesis), while the journalist says it is longer. Thus we have $H_0 : M = 150$ and $H_a : M > 150$.

The words "population median" ought to be stated or implied. You can say something like "time taken by the average taxpayer" to imply the whole population of taxpayers.

If it pleases you logically, the null could be $H_0 : M \leq 150$, but the null and alternative definitely need to be the right way around (the null always has an "equals" in it, since it has to give you a value to work from).

   (d) (2 marks) What does the table in Figure 12 tell us? Explain briefly.

Solution: This is a count of how many of the data values were above 150 (16, under `TRUE` and how many were 150 or below (14, under `FALSE`). These values will be used in a moment to get a P-value for the sign test.

What you ought to be suspecting at this point is that these values are a pretty even split, above and below 150, and so we will have a hard time rejecting the null hypothesis (which is likely to require a much more uneven split than this). This table is exactly the kind of thing you would expect to see if the population median really is 150 minutes.

This is one of those questions where you can do yourself in by overthinking. Just tell me what that 16 is and what that 14 is. I can stand a little interpretation (something like "there were an almost equal number of times above and below 150 minutes" is OK), but be careful not to go so far that you forget to tell me what those two numbers actually are. If I wanted interpretation, I would say something like "what do you learn from the table" or "what do you conclude from the table". This one was meant to be "tell me what you see", as simple as that.

(e) (2 marks) Figure 13 is a table of the binomial distribution with $n = 30$ and $p = 0.5$. The number in the `prob` column is the probability of the number of successes in the `success` column *or fewer*. Thus, for example, the probability of 5 successes or fewer is 0.00016 (1.6×10^{-4}).

Use this table to obtain a P-value for your sign test. *Do not* express your answer in scientific notation; you should give an ordinary decimal number.

Solution: Our test was one-sided so we only need one tail of the distribution (no multiplying by two). If the alternative hypothesis is correct, and the median is greater than 150, then the majority of the observed values should be greater than 150 and only a few should be less. In our data, 14 values were less (and 16 were more). So our P-value is the probability of 14 successes or less, which is 0.428. You need to "translate" this from `4.28e-01`, which is in scientific notation, and you should round it to a sensible number of decimals like 3.

If you prefer, you can also find the P-value as the probability of 16 successes or more. This will get you to the same answer, but you have to think carefully: the probability of 16 successes or more is one minus the probability of *15* successes or less, because the binomial distribution is discrete. Thus, this way, the P-value is $1 - 0.572 = 0.428$ as before.

You ought to do as I did for this one-sided test and check that our result was "on the correct side". If we had observed 16 values below 150 and 14 above (the values we had but flipped around), we would have had *no* evidence in favour of the alternative at all, and we would have stopped there.

A lot of people found the probability of *16* or less. This is the wrong tail (as explained above, you need 16 *or more*). Also, the wording of the question ought to make it clear to you that you need a *number*, not just something with a `pbinom` in it. (That would show that you know how to copy something from your notes, but *not* how to make use of it.)

There is an exam strategy here and in the next part, that you should make sure you know about: if you can't get an answer for some reason, you should *explain your process* by which you would get an answer. There are usually some marks to be had for showing that you know what to do, even if you can't do it. (In the next part, where you have to draw a conclusion based on your P-value, you can get full marks by explaining how to get from a P-value to a decision about whether or not the journalist has a story, *even if you don't have a P-value*. I use notation like "given (a)" when marking something like this. Going with this: if you just give an answer (and that's all I wanted, as here), you get full marks if the answer is right and zero if it is wrong, but if you have some explanation or process, you can get part marks even if your answer is wrong. Here, you could say that you need the probability of 14 or less, and even if you mess up reading the value from the table, you've got yourself one point.

(f) (2 marks) Does the journalist "have a story": that is, is there evidence in favour of his claim? Explain briefly.

Solution: We have not rejected the null hypothesis, since the P-value was way too high. The journalist's claim was the alternative hypothesis, so there is no evidence in favour of his claim at all. The journalist does not have a story. (The IRS's claim was not "proved correct" by this, but the data observed were entirely consistent with the median time being 150 minutes.)

The right answer *for you* is to start from the P-value you got, decide if it's less than 0.05, and then say something about whether the journalist's claim was justified. (If you didn't have a P-value, describe what you would do.) If you got your hypotheses the wrong way around in (c), I was less sympathetic, because there needs to be *evidence in favour* rather than a lack of evidence against, which is what failing to reject a null would be.

(g) (3 marks) Figure 14 shows a function to obtain a two-sided P-value for a sign test with null median

`med` and data `z`. (This previously appeared in one of the assignment solutions). This function is used in Figure 15. Use the results shown there to obtain a 95% confidence interval for the median time spent filling out the IRS forms, to the accuracy that the output permits.

Solution: Look for the two places where the P-values cross 0.05, first on the way up, and second on the way down. These are: between 110 and 115, and then between 235 and 240. Thus I would say the 95% confidence interval is between 115 and 235 (taking the values that I know to be inside).

If you want to try to interpolate, that is not at all necessary, but I won't penalize you for trying. That would give something like 112.5 to 239. In the end, as long as the lower end of your interval is an x with $110 < x \leq 115$ and the upper end is a y with $235 \leq y < 240$, I'm happy. (We know 110 and 240 to be outside the interval, so don't try to include them in the interval.)

In practice, you would probably have a second round to refine the interval: you'd pick some null medians between 110 and 115, and do the `sapply` thing again, and also between 235 and 240 to get the top end more accurately. What happens is that the P-value changes at a data value, and then you get into that issue about "what if the null median is exactly equal to a data value", and you have to throw away values exactly equal. I didn't want to get into that here.

This confidence interval is very wide, and, as it should, contains the null value of 150. (We did a one-sided test, so the correspondence is no longer exact, but our two-sided P-value would have been twice what we actually had, which would have been way greater than 0.05.) The sign test doesn't use the data very efficiently, since it only counts above and below (and not how far above and below), so you tend to need a large sample to get confidence intervals for the median that are at all respectable (by which I mean "short enough to be useful"). This is the same issue as for opinion polls on questions with yes-no answers: there, each respondent doesn't give you much information, so you need a lot of them, like hundreds, to estimate the proportion in favour with any accuracy.

(h) (2 marks) The journalist also did a t-test that the mean was 150 minutes, against the alternative that it was greater, and obtained a P-value of 0.06. What do you think it is about the data that would make the t-test come out almost significant, while the sign test is nowhere near significant? Explain briefly.

Solution: We said when we looked at the histogram that the data were skewed to the right. There are a few large values that would have a disproportionate (increasing) effect on the mean (and would have a much smaller effect on the median). In fact, the sample mean is 183.7 and the sample median is only 164. So there would be more evidence that the *mean* is greater than 150 than there is that the *median* is greater than 150.

It is enough to observe that (i) there are outliers at the upper end, (ii) these are going to make the mean bigger (than it ought to be). Observing that there are outliers is not enough, and observing that they are going to affect the mean is not quite enough, because what makes this work is that the t-statistic is bigger (more positive) than it would be without the outliers, and thus the P-value is smaller than it ought to be.

Possibly also relevant is the fact that the sign test is typically not as powerful as the sign test (it has a harder job rejecting the null than the t-test would, in a situation where the t-test is reasonably applicable). You could argue that this applies here, since the skewness is not *so* severe, and the sample size is moderately large (so that the sampling distribution of the sample mean is approximately normal). But I find this to be less insightful as an explanation as the thing about the few large values affecting the mean. If you want to go this route, I think you have to make all the points I made.

5. Suppose you have data

$$16, 17, 18, 20, 20, 23, 25, 29, 34$$

and you want to test whether the population *third quartile* could be 21 (as the null hypothesis) against the alternative that the population third quartile is *less* than 21. You want to use the same idea as the sign test for the median.

(a) (2 marks) What would you use as the *test statistic* for your test (that is, the value you would look up in a table to get a P-value for)?

Solution: Observe whether each value is above or below the hypothesized third quartile, as we do for the sign test, and count the number of values below 21, which in this case is 5 (out of 9). (You could equally well count the number of values above, 4 out of 9.)

In R code:

```
x=c(16,17,18,20,20,23,25,29,34)
table(x<21)
```

```
##
## FALSE  TRUE
##     4     5
```

(b) (4 marks) How would you obtain a P-value for testing that the third quartile of the population from which these data came is 21? Explain in words what you would do. Give R code if you think it makes your explanation clearer. (I should be able to reproduce what you say and get the right answer for the right reason.)

Solution: First off, copying my sign test function without explanation is not displaying any insight. You can use it, but you'll need to make some changes (as described below) and *explain what you're doing*.

If the null hypothesis is correct and the third quartile really is 21, then the number of observations below 21 should have a binomial distribution with $n = 9$ (sample size of 9) and $p = 0.75$, since three-quarters of the observations should be below Q3 and the other quarter should be above. The mean of this binomial is $9(0.75) = 6.75$. I observed five values under 21, which is less than the mean, so the P-value is the probability of observing 5 successes or less in a binomial distribution with $n = 9, p = 0.75$. In R code, that would be

```
pbinom(5,9,0.75)
```

```
## [1] 0.1657257
```

Looking at "above" requires more thought here than it does for the sign test, because the number of values above 21 is binomial with $n = 9$ *and* $p = 0.25$. This has a mean of $9(0.25) = 2.25$. We observed 4 values above 21, well above the mean, so that the P-value is the probability of 4 *or more* successes in this binomial distribution, which is one minus the probability of 3 or fewer:

```
1-pbinom(3,9,0.25)
```

```
## [1] 0.1657257
```

Here, we would not reject a third quartile of 21. This is not very surprising, since the proportion of data values less than 21 was $5/9 = 0.56$, not all that far from 0.75.

Notice that what I had to do, to figure out whether I was in the upper or lower tail, was to compare the number of values I observed on the appropriate side with *the mean* of the appropriate binomial distribution. I couldn't just take the smaller of the two values, like I did with the sign test, where there was a 50-50 chance of being on either side of the median, and therefore the smaller frequency, above or below, was the one in the lower tail. Here, it is not like that: I'm *expecting* more of the data values to be below the hypothesized third quartile, and fewer above. So simply finding the probability of less than 4, even in the right binomial distribution, would have been wrong, because it is in the wrong tail. I had to compare 4 with the *right* binomial distribution, which was the one with $p = 0.25$, and for that, it was in the *upper* tail.

I was somewhat relaxed about this: if you appeared to be doing about the right thing for the right reason, I was happy.

If you have discovered `binom.test`, you may be able to make that work:

```
binom.test(5,9,p=0.75,alternative="less")
```

```
##
##  Exact binomial test
##
## data:  5 and 9
## number of successes = 5, number of trials = 9, p-value = 0.1657
## alternative hypothesis: true probability of success is less than 0.75
## 95 percent confidence interval:
##  0.0000000 0.8312495
## sample estimates:
## probability of success
##              0.5555556
```

```
binom.test(4,9,p=0.25,alternative="greater")
```

```
##
##  Exact binomial test
##
## data:  4 and 9
## number of successes = 4, number of trials = 9, p-value = 0.1657
## alternative hypothesis: true probability of success is greater than 0.25
## 95 percent confidence interval:
##  0.1687505 1.0000000
## sample estimates:
## probability of success
##              0.4444444
```

giving the same answer, as long as you use the appropriate tail.

Another approach is to try to make this a chi-squared test, since we have frequencies of observations in different categories. This is not a chi-squared test of association (as Mood's median test is), but a chi-squared test of fit: three-quarters of the observations should be below 21 and the other quarter above. `chisq.test` handles this. The first input is the observed frequencies for below and above 21, and the second one, labelled `p`, is the proportion of values that should be in each category:

```
chisq.test(c(5,4),p=c(0.75,0.25))
```

```
## Warning in chisq.test(c(5, 4), p = c(0.75, 0.25)): Chi-squared approximation may
be incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  c(5, 4)
## X-squared = 1.8148, df = 1, p-value = 0.1779
```

The P-value is not the same. This is because this chi-squared test does something equivalent to the normal approximation to the binomial. Here, $np = 9(0.75)$ and $n(1-p) = 9(0.25)$ are both smaller than 10, so the normal approximation to the binomial will not be much good. If you can make this work, I'm good with it, though you won't have a test statistic to give for (a) (I'll make sure you get a fair number of marks overall). This test is by its very nature *two*-sided, and we want our test to be one-sided, so the P-value that comes out of here ought to be divided by 2 (which means that the correspondence between this approximate P-value and the exact one from the binomial is really not very good).

What I find rather interesting is that the sign test does not use the *sample* median at all, and this test does not use the sample Q3 at all. This probably seems rather odd, since you are used to the t-test using the sample mean (and the z-test for proportions using the sample proportion). But it goes to show that you can obtain tests in different ways. (The analysis of variance is a bit like this one in that you compare means of several groups not by feeding the group means into some big formula, or at least not directly, but instead measure variability *around* the group means and the overall mean and then doing something with that.)

6. A psychologist is studying pattern-recognition skills in grade 4 children. Each child is given a test with 10 patterns to identify. Each child does the test under one of four different conditions:

   Praise: child given praise for right answers and no comment about wrong answers

   Criticism: child given criticism for wrong answers and no comment for right answers

   Interest: child given no praise or criticism, but observer expresses interest in what the child is doing

   Silence: observer remains silent while watching the child

   The response variable for the study is the number of patterns out of 10 that each child correctly identifies. The research question is whether the different conditions have any kind of impact on pattern-recognition ability. The data are shown in Figure 16.

   (a) (3 marks) The mean and SD of the number of patterns identified for each setting, along with boxplots, are shown in Figures 17 and 18. What *two* assumptions do we need to assess for analysis of variance? Explain briefly why those assumptions are difficult to assess here. You do not need to go further in assessing the assumptions. (We will assume in the rest of the question that those assumptions are sufficiently satisfied.)

Solution: The two things you need to address are (i) approximate normality within each group, and (ii) similar spreads between the groups. ("Independence of observations" is an assumption too, but it's common to most of our tests, so it's not really special to analysis of variance.)

They are difficult to assess here because the sample sizes in each group are so small. (Or, that the data are rather obviously discrete, which also makes it hard to judge whether they are "normal enough".)

That's all I wanted, but note that the small sample sizes mean that the data within each group may look non-normal and the spreads may look different, just by chance (even if the populations within each group really are normal, all with the same spread). The boxplots for "criticism" and "interest" look symmetric, albeit with a larger spread for "interest"; the boxplots for "praise" and "silence" look skewed left, with "praise" having an especially small spread (all the numbers of patterns solved in the "praise" group are high, near to the maximum of 10, and as usual when the values are near to a limit, they tend to be skewed away from that limit).

So it's not at all clear that the assumptions are satisfied here, but with such small samples it's very difficult to judge, and so we will press on anyway.

Note some things (seen while marking):

- I didn't ask you to assess the assumptions, so you don't need to try. (As I explained above, you could get data like these with these small sample sizes even if the normality and equal-variance assumptions were both good.)
- Different numbers of subjects in each group is not a problem. Provided our assumptions are OK, the ANOVA will be fine.
- We don't *need* large samples. If we are in the fortunate position of having them, then the normality matters less, but small reasonably-normal samples of reasonably-equal spreads are fine.
- Make sure you are clear what an assumption *is*, and that you know the difference between *stating* an assumption and *assessing* it. An assumption is something that has to be true for an analysis to be valid (reliable, trustworthy). Like, for example, normally distributed data (for the t-tests). (You can throw in the word "approximately" if you like, because the t-tests will still work all right if the data are not *exactly* normal, but the theory by which the t-tests are obtained says "if the data are from (exactly) a normal distribution, then the test statistic has

a t distribution if the null hypothesis is true." I don't think "null hypothesis true" is part of the assumptions of the test; if it isn't true, the test will still properly reject most of the time as it should.

So, a statement of an assumption is something like "the data within each group are normally distributed". An assumption is a statement about data that may or may not be true.

Assessing an assumption is another matter. Then you *do* have to look at pictures of your data, and decide if your data look normal enough (for example). I didn't need any of that in this question.

- While marking this question, if you made the relevant points in some reasonably clear fashion, you got the marks, even if I wasn't sure if you knew which ones were assumptions and which one was an explanation of why the assumptions were difficult to assess. I might not be so generous in the future.

(b) (2 marks) An analysis of variance is shown in Figure 19. What precisely do you conclude from this, in the context of the data?

Solution: The key here is knowing where to stop! The null hypothesis is that all the settings have the same mean number of patterns recognized, and this is rejected, so the mean numbers of patters recognized are *not all the same* (or something equivalent to that).

"Conditions have an impact on pattern recognition" or something similar is another way to say the same thing, since two of the conditions could have the *same* impact. What you need to avoid is anything that looks like "the means are *all* different", because that is not what the alternative hypothesis is. ("Three of the means are the same but the other one is different" is part of the *alternative*, not the null.)

If you want to include the wording of the null hypothesis in your answer, that works, but you have to be careful of your English: "we reject the null hypothesis that all the means are equal, in favour of the alternative that not all the means are equal" is OK, but "we reject the null that not all the means are equal" is confused: is the bit after "that" supposed to be a statement of the null hypothesis or of your conclusion? It looks like a statement of the null, but it isn't. "We reject the null, and conclude that not all the means are equal" is clear (and correct, though I would prefer something about "conditions" and "patterns recognized" in your answer).

Saying "reject the null" and stopping there is a fast way to have me take off a mark. "Reject the null" is *never* a complete answer.

If you want to look ahead to the Tukey, you can (though you don't need to). It might be useful as a way to convince me that you understand what the alternative hypothesis in an ANOVA is: "there are differences but we don't know where they are".[1]

(c) (3 marks) The results from Tukey's method are shown in Figure 20. Was it a good idea to obtain these results? Explain briefly, in the context of the data. If you think it was a good idea, what do you conclude from the results?

Solution: The analysis of variance F-test was significant, so that we know that some settings produce different mean numbers of patterns recognized than others, but not (yet) which ones. This is what Tukey's test will do.

I'm trying to move you beyond "F significant, therefore do Tukey" to an understanding of *why* doing Tukey is a good idea.

This SAS Tukey output is more like R's, in that each pair of groups is listed, along with an indication of whether the group means are significantly different. This is because the numbers of observations in each group are not all equal, and so the "lines" thing doesn't apply. But don't worry about that: just interpret what you see.

There is only one pair of groups that are significantly different: `praise` has a significantly higher mean (number of patterns recognized) than `criticism`. This is not terribly surprising from a practical point of view. No other significant differences were found; for example, `silence` and `interest` have no special effect on how many patterns were recognized.

The purpose of Tukey is to say which groups (conditions) differ significantly from which in terms of pattern recognition. It is not for ranking: if you want to do that, you simply need to put the sample means in order. But your ranking of, say, silence and interest relative to each other is no more than random, because there is no significant difference between these.

I could be swayed in this part by a demonstration of a clear understanding of what was going on: that the ANOVA only says "there exist differences, at least one", but not where they are, Tukey says "these particular conditions are significantly different from each other", and finally using the Tukey output to say "praise and criticism differ significantly from each other with praise being higher, and no other significant differences exist" (or, "only praise and criticism differ significantly").

If you for some reason thought that all the means were actually equal, you got a free ride in this part: "there are no differences to find, so no point doing Tukey" was a very fast three marks (but of course you lost some in the previous part). I would also take a well-argued case along the lines of "the data aren't close to normal, so we shouldn't be doing ANOVA, never mind Tukey". My implication in the question, however, was that I thought things were not bad enough to stop me doing the ANOVA (after all, I did it in (b)), and that therefore, for the purposes of the question, you would humour my opinion.

Anyway, with these tiny sample sizes, it's hard to prove anything beyond the obvious. So our advice to the psychologist should be "collect more data"!

7. Suppose we are trying to organize a study that estimates the number of hours of TV people watch per week. In particular, a study carried out ten years ago found that people watched a mean of 20 hours of TV per week, with a standard deviation of 8 hours per week. We are trying to see if the mean has changed since then (for example, it may have gone down because people are spending more time online, for example on Facebook, rather than watching TV as they used to do).

   (a) (3 marks) We have decided that a change in mean of 2 hours per week is of interest to us. We have a budget to survey 40 people. We want to know the power of our test to detect this change. Write SAS code to find this out (under the assumption that the number of hours of TV watched has approximately a normal distribution).

Solution: This is a one-sample t-test, two-sided. Our best guess at the population standard deviation is the value from the study ten years ago (which may not be very good, but it's the best we have). The null mean is 20, and the actual mean is 2 away from 20 (so either 18 or 22 will work and will give the same answer):

```
proc power;
  onesamplemeans
  test=t
  nullmean=20
  mean=18
  stddev=8
  ntotal=40
  power=.;
```

The POWER Procedure
One-Sample t Test for Mean

Fixed Scenario Elements

Distribution	Normal
Method	Exact
Null Mean	20
Mean	18
Standard Deviation	8
Total Sample Size	40
Number of Sides	2
Alpha	0.05

Computed Power

Power
0.338

or

```
proc power;
  onesamplemeans
  test=t
  nullmean=20
  mean=22
  stddev=8
  ntotal=40
  power=.;
```

The POWER Procedure
One-Sample t Test for Mean

Fixed Scenario Elements

Distribution	Normal
Method	Exact
Null Mean	20
Mean	22
Standard Deviation	8
Total Sample Size	40
Number of Sides	2
Alpha	0.05

Computed Power

Power
0.338

What also works is to think of the null mean as being zero (in which case you don't need to specify it) and to take `mean` to be 2, the difference from the null (or -2, which will give the same answer):

```
proc power;
  onesamplemeans
  test=t
  mean=2
  stddev=8
  ntotal=40
  power=.;
```

The POWER Procedure
One-Sample t Test for Mean

Fixed Scenario Elements

Distribution	Normal
Method	Exact
Mean	2
Standard Deviation	8
Total Sample Size	40
Number of Sides	2
Null Mean	0
Alpha	0.05

Computed Power

Power
0.338

The power, it turns out, is only 34%.

In fact, any mean and null mean that are 2 apart will work. *Logically*, a null mean of 20 and a mean of 18 or 22 are the only things that make sense, but if you switch `mean` and `nullmean` around, you'll still get the same answer, so that (perhaps fortunately) gets full marks too.

There is only one group of people watching TV. If it had been, say, two different age groups of people that we were comparing with each other, then `twosamplemeans` and `diff_satt` would have been correct. But that was not the case. (I suspect that most of you who gave that answer copied my lines of code from the notes without really understanding what you were doing.)

If you don't know anything else, write down `proc power`. This might get you a point, especially if you weren't going to get many other points on the page.

(b) (2 marks) Our supervisor has said that she wants to achieve power 0.80, and will provide resources to collect a sample that is large enough. To find out how large a sample will be needed, what changes would you make to the code of the previous part?

Solution: Two changes: put `.` (missing) for `ntotal`, and put in the desired power instead of the `.` you previously had next to `power=`:

```
proc power;
```

```
onesamplemeans
test=t
nullmean=20
mean=18
stddev=8
ntotal=.
power=0.80;
```

Since we are getting close to the end of the exam, I was fairly relaxed about this one: as long as you showed that you knew what you were doing, you could say it more or less any way you liked. But you *did* have to name *two* things.

In the light of my calculations in the previous part, we would expect to need a much bigger sample size in order to increase the power from 0.34 to 0.80:

```
                    The POWER Procedure
                 One-Sample t Test for Mean

                  Fixed Scenario Elements

Distribution                      Normal
Method                             Exact
Null Mean                             20
Mean                                  18
Standard Deviation                     8
Nominal Power                        0.8
Number of Sides                        2
Alpha                                0.05
                  Computed N Total

                  Actual         N
                   Power     Total

                   0.802       128
```

and so we do: the sample size goes up from 40 to 128. Let us hope that our supervisor has deep pockets!

(c) (3 marks) Another way to obtain the power in this situation is by simulation. The assumption used by SAS in the previous parts is that the population distribution is normal. Recall that the null mean is 20 and that the true mean differs from the null mean by 2 (the direction does not matter). Write an R function, with no input, that draws a random sample of size 40 from the true normal distribution, tests whether its mean is significantly different from the null mean, against a two-sided alternative, and returns the P-value of the test.

Solution: Generating a random sample from a normal distribution uses `rnorm`, which requires three things: the number of random values to generate, the true mean of the normal distribution to generate them from, and the SD of that normal distribution. Thus this will work for the random sample part:

```
set.seed(457299)
```

```
x=rnorm(40,18,8)
x
```

```
##  [1] 30.974939 12.029221 15.848554 12.403719 19.705903 23.671748  9.373368
##  [8] 24.330483 18.032376 26.767037  4.756199  8.345006 28.149993 24.707146
## [15] 12.031149 18.422029 30.119003 17.101530 20.132282  4.236974 27.286229
## [22] 21.775012 25.556864 11.541939 20.233511 23.491555 22.861133 18.570753
## [29] 16.764112  8.579104 18.098288 12.846144 29.221960 11.072184 28.598037
## [36] 29.749145 27.570900 29.477170 13.366231 21.027655
```

Or you can use 22 instead of 18, equally good (by symmetry).

The testing part uses `t.test` and the null mean:

```
tt=t.test(x,mu=20)
tt
```

```
##
##  One Sample t-test
##
## data:  x
## t = -0.44299, df = 39, p-value = 0.6602
## alternative hypothesis: true mean is not equal to 20
## 95 percent confidence interval:
##  17.05370 21.88763
## sample estimates:
## mean of x
##  19.47066
```

In this case, the sample mean happened to be close to 20 (even though the population mean was 18). We just want the P-value:

```
tt$p.value
```

```
## [1] 0.6602238
```

(These ideas come almost directly from the solutions to assignment 5.)

So the function will look something like this, with no input:

```
sim.t.test=function() {
  x=rnorm(40,18,8)
  tt=t.test(x,mu=20)
  return(tt$p.value)
}
```

and I can run it a few times for checking:

```
replicate(10,sim.t.test())
```

```
##  [1] 0.0090626904 0.1071003049 0.0109193404 0.0214205884 0.5544731116
##  [6] 0.0001201187 0.0150063587 0.1814994358 0.7197975782 0.4378382423
```

Sometimes it rejects and sometimes not. In my case, there were 5 rejections out of 10.

I was likewise fairly relaxed about this. If you wrote something that looked like your own function (even if it did nothing), you probably got one point; if you got most of the way to an answer, you got 2, and if you got almost all of the way, I gave you 3 even if it wasn't perfect.

What I do *not* like is you copying one of my functions with no changes or minimal changes. This shows a lack of understanding about what is needed here, because the functions I saw copied did other things, like a sign test or sampling from a different distribution, that could not possibly answer this question. That is a fast way to a zero. Also, I said "simulate", and `power.t.test` *calculates* the power under the specific situation of normally-distributed data (as `proc power` does), so that wasn't what was needed here. The point of the last three parts of this question is to be able to handle *any* data distribution (but I get ahead of myself).

(d) (2 marks) Give R code to run your function 1000 times and to count how many P-values are less than 0.05. (This number, divided by 1000, is the simulated power.)

Solution: `replicate` is the key here, and you see that the hard part was actually writing the function above:

```
ans=replicate(1000,sim.t.test())
table(ans<0.05)
```

```
##
## FALSE  TRUE
##   670   330
```

and the simulated power is the number under `TRUE` divided by 1000.

This is the easiest way. Or you can take advantage of `TRUE` counting as 1 and `FALSE` as zero:

```
sum(ans<0.05)
```

```
## [1] 330
```

If you insist, you can write a loop and count the number of rejections as you go. This is a very Python-like way of doing it:

```
count=0
for (i in 1:length(ans)) {
   if (ans[i]<0.05) count=count+1
}
count
```

```
## [1] 330
```

On the principle of "get the job done", a loop, correctly written, is as good as either of the more R-like ways. I think `table` is much the easiest and most elegant solution, but I am happy to accept inelegant solutions. If they work!

In any of these cases, my simulation gave a power of 0.33, which is very close to (but not equal to) what SAS calculated (without doing any simulation).

You needed to do two things: use `replicate` (or, if you insist, a loop) to run your function 1000 times, and then something like `table(pval<0.05)` to count how many of them were small enough to reject with. This really ought to be two lines; if you try to do it in one, it would have to look like this, with the brackets in the right places:

```
table(replicate(1000,sim.t.test())<0.05)
```

```
##
## FALSE  TRUE
##   670   330
```

This one ought to be different from before, because I just ran another simulation. (It seems that it happened to come out the same.)

If your function in (c) returns something that is already true or false according to whether or not you reject (like the `is.reject` in one of my functions), then you can say

```
r=replicate(1000,my.function)
table(r)
```

because this `r` already *is* true or false. But otherwise, you have to compare your long list of P-values with 0.05 before you make the table.

Here is a way to get two points here, *even if you have no idea how to do the rest of the question.* You can assume that you wrote a function called `f`, say, in (c) (even though you actually didn't) that takes no input, and that when you run it, it returns you a P-value.[2] So you write "assume the function required in (c) was called `f`", and then these two lines of code:

```
pvals=replicate(1000,f())
table(pvals<0.05)
```

and that will get you two points!

The logic to this is a marking principle called "follow-through", and the idea is that just because you couldn't do one part of a question, that shouldn't handicap you in later parts. If your answer to a later part follows logically from what would have come from an earlier part, had you been able to do it, then you can get full marks for the later part.[3] (This is why explaining what you would do, even if you can't do it because you couldn't do the earlier part, or you got the earlier part wrong, is often worthwhile.)

(e) (2 marks) What would be an advantage to using the simulation approach to assessing power over the `proc power` approach, in general? Explain briefly, giving an example if you wish.

Solution: The principal advantage of the simulation approach is that we are not restricted to those particular situations that are coded for in `proc power` (mainly, normally-distributed data and a certain list of tests).

Anything that says approximately this (such as "we can deal with non-normal data") is good; anything that I judged to be tangentially relevant got a point.

By way of example: the number of hours of TV watched may have some kind of right-skewed distribution, since it has a lower limit of zero. One possibility might be a Poisson distribution (since the number of hours is recorded as a whole number, and this is a distribution you might have heard of). To simulate data from a Poisson distribution, you replace `rnorm` in the function above by `rpois`.

The point is that R offers you a lot more flexibility, if you are willing to work to take advantage of it, and if you are willing to specify what distribution the data (might) have, at least as a what-if.

To see how the Poisson thing plays out, we copy-and-paste our function, and we replace `rnorm(40,18,8)` by `rpois(40,18)`. `rpois` only has two inputs: the sample size and the true mean (since the variance/SD of the Poisson distribution is determined by the mean):

```
sim.t.test.poisson=function() {
  x=rpois(40,18)
  tt=t.test(x,mu=20)
  return(tt$p.value)
}
```

Try that a few times:

```
replicate(5,sim.t.test.poisson())
```

```
## [1] 1.338717e-05 1.089498e-06 2.884401e-01 6.226055e-02 6.186020e-03
```

These P-values are all pretty small, though one of them (the third one) is bigger than 0.05. The Poisson distribution has variance the same as the mean (18), so its SD is $\sqrt{18}$:

```
sqrt(18)
```

```
## [1] 4.242641
```

only about half the size of the SD of the normal we had. Thus 18 and 20 are farther apart relative to the SD in this Poisson case than the normal above. Thus we'd expect the power of the test for Poisson data to be higher. Some people phrase this in terms of "effect sizes": scale the difference in means by the appropriate SD. Here, for our actual question with normal data (SD 8), the effect size is

```
(20-18)/8
```

```
## [1] 0.25
```

but for the notional Poisson data of this part, with variance 18, the effect size is

```
(20-18)/sqrt(18)
```

```
## [1] 0.4714045
```

A bigger effect size is easier to detect, and so the power should be bigger. Is it?

```
sim=replicate(1000,sim.t.test.poisson())
table(sim<0.05)
```

```
##
## FALSE  TRUE
##   182   818
```

Oh yes, 0.816 compared to 0.330.

The issue here is not that the t test is more powerful when the data are Poisson than when the data are normal. To do *that*, we'd have to generate normal data with *the same SD* as the Poisson. What we have actually concluded here is only that bigger differences from the null are easier to correctly reject for: the subtlety here is that the difference in *means* is the same, but the Poisson SD is smaller, so that *relative to the SD*, the Poisson data is farther away from the null.

I don't care what precise example you come up with. You might not have heard of the Poisson distribution, so pick one that you *have* heard of. I think the easiest way of answering the question is to think about the data being from some non-normal distribution, which you can generate in R via `rbinom` or `rpois` or whatever, but which takes you outside what `proc power` can handle. (Changing the data distribution is just a matter of changing one line in the function, as I did. Once you've written the function, you can use the same idea for any data distribution.)

The downside of estimating power using simulation is that it doesn't give you a direct way of getting the sample size required to achieve a certain power, since you have to specify a sample size in your generation of the random samples from the true distribution. The only way I see around that is to do *repeated* simulations. For example, you might find that your first guess at the sample size doesn't give you as much power as you want, so you increase the sample size (guessing how much to increase it by) and simulate the power again. Then you repeat this process, adjusting the sample size up or down until you get (roughly) the power that you want. To illustrate, let's take our first function:

```
sim.t.test=function() {
  x=rnorm(40,18,8)
  tt=t.test(x,mu=20)
  return(tt$p.value)
}
```

This used a sample size of 40 and got a power of 0.330. Let's suppose that we are aiming for a power of 0.5 (not very ambitious, but then this small difference is a hard one to detect). How much power would a sample size of 80 give? We can either edit this function to have the line read `x=rnorm(80,18,8)`, or, since we'll be calling the function several times, we can let sample size `n` be an *input* to it:

```
sim.t.test=function(n) {
  x=rnorm(n,18,8)
  tt=t.test(x,mu=20)
  return(tt$p.value)
}
```

I've made two changes: the `n` in the header line, and the replacement of 40 with the input `n`.

So now I estimate the power for a sample of size 80 thus:

```
ans=replicate(1000,sim.t.test(80))
table(ans<0.05)
```

```
##
## FALSE  TRUE
##   410   590
```

This gave a power 0.595, a bit bigger than our target, so our guess at a sample size of 80 was a bit too big. So we need to guess again. In the absence of any better ideas, a sample size of 60 (halfway between 40 and 80) ought to give a power something like halfway between 0.330 and 0.595, which is just under 0.5. So maybe `n` ought to be a little bigger than 60. How about 65?

```
ans=replicate(1000,sim.t.test(65))
table(ans<0.05)
```

```
##
## FALSE  TRUE
##   466   534
```

Maybe a smidgen smaller than 65 would be OK, but I'm going to call this good. If you want to improve it, you can try again, maybe 60 or 62 or something like that.

Notes

[1] "There are differences among the means" also works as an answer here, because it says that at least one pair of means differs without implying that they all do. In mathematical terms, it's the difference between $\exists$ and $\forall$.

[2] The function `f` is *literally* a black box!

[3] There are some caveats about how you can't make things easier for yourself by getting an earlier part wrong, but this is the basic idea.