University of Toronto Scarborough
Department of Computer and Mathematical Sciences
STAC32 (K. Butler), Final Exam
December 14, 2018 19:00-22:00

IT IS ASSUMED THAT YOU HAVE READ THE BOX BELOW.

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 70 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.

Code for R graphs should be in `ggplot` style, as in lecture. There may be partial credit for "base" graphs.

For any questions below involving R code, you may assume that this code has already been run:

```
library(tidyverse)
library(broom)
library(ggrepel)
```

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	3	
2	2	
3	3	
5	3	
9	2	
10	3	
12	3	
15	4	
17	3	
18	2	
19	4	
25	5	
26	4	
27	3	
29	4	
31	2	
32	4	
42	9	
43	2	
47	3	
49	4	

1. The data in Figure 1 (in the booklet of code and output) show grain yields in modern and historic times, measured in kilograms per hectare, for 16 grain-producing regions of the world. The data have been uploaded to your SAS Studio account, in a file called `grain-yields.txt`.

   (a) (3 marks) Give SAS code for reading the data into a SAS data set called `grain`.

Solution: Something like this. My username is `ken` (you need to use yours):

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

That gives this output:

```
proc print;
```

```
              Obs          Region          Modern         Historic

                1               1            1610             1590
                2               2            2230             2360
                3               3            5270             5161
                4               4            6990             7170
                5               5            2010             1920
                6               6            4560             4760
                7               7             780              660
                8               8            6510             6320
                9               9            2850             2920
               10              10            3550             2440
               11              11            1710             1340
               12              12            2050             2180
               13              13            2750             3110
               14              14            2550             2070
               15              15            6750             7330
               16              16            3670             2980
```

You don't need the `proc print`, though I'm good with it if it's there (on the basis that normally, when reading data in, you'd take a look at it to be sure that it was read in correctly).

Grading note: minus one per error, down to a minimum of one if you have *something* right. There should be something that looks like a username after `/home`, ie. not `ken` for you (that's *my* username). I'm not overly picky otherwise; the point is to put in something that looks something like your username.

This is the same grading scheme that is used for coding throughout. If you want full marks, you have to be 100% right, right down to semicolons in SAS. (You will probably get away with missing semicolons later, but not in the first question.)

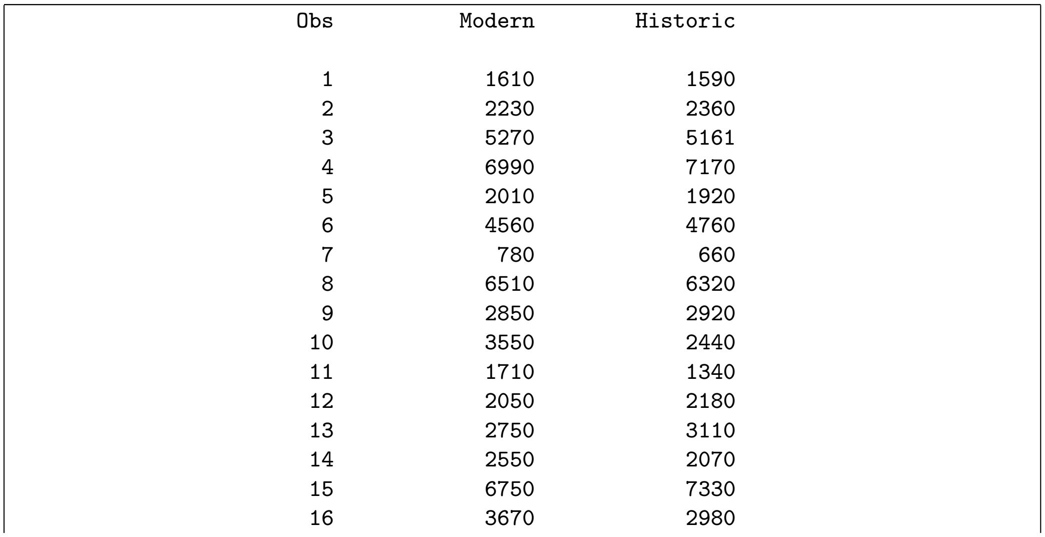
If *your* actual name is Ken, your username might be `kenw` or something else after the `ken`, so add *something* to `ken` to make it look like the kind of username you have. I'm not checking; all I know is that your username *isn't* `ken`! Thus, if that's what you write, you are copying what *I* wrote *without thinking*.

(b) (2 marks) Give SAS code to display only the two columns of your data set called `Modern` and `Historic`.

Solution: Here and throughout, SAS is not case-sensitive, so if you want to refer to your variables as `modern` and `historic`, feel free to do so.

The easiest way is to add a `var` line to your `proc print`:

```
proc print;
  var Modern Historic;
```



```
Obs      Modern      Historic

  1        1610          1590
  2        2230          2360
  3        5270          5161
  4        6990          7170
  5        2010          1920
  6        4560          4760
  7         780           660
  8        6510          6320
  9        2850          2920
 10        3550          2440
 11        1710          1340
 12        2050          2180
 13        2750          3110
 14        2550          2070
 15        6750          7330
 16        3670          2980
```

This is also a fair guess if you haven't seen it yourself in this context; `var` seems to be a pretty standard way to specify variables to do something with in the SAS world.

If you *must*, create a new data set without `Region` and `print` that, but this is a lot of work for two points:

```
data grain2;
  set grain;
  drop Region;

proc print;
```

```
                              Obs          Modern        Historic

                                1            1610            1590
                                2            2230            2360
                                3            5270            5161
                                4            6990            7170
                                5            2010            1920
                                6            4560            4760
                                7             780             660
                                8            6510            6320
                                9            2850            2920
                               10            3550            2440
                               11            1710            1340
                               12            2050            2180
                               13            2750            3110
                               14            2550            2070
                               15            6750            7330
                               16            3670            2980
```

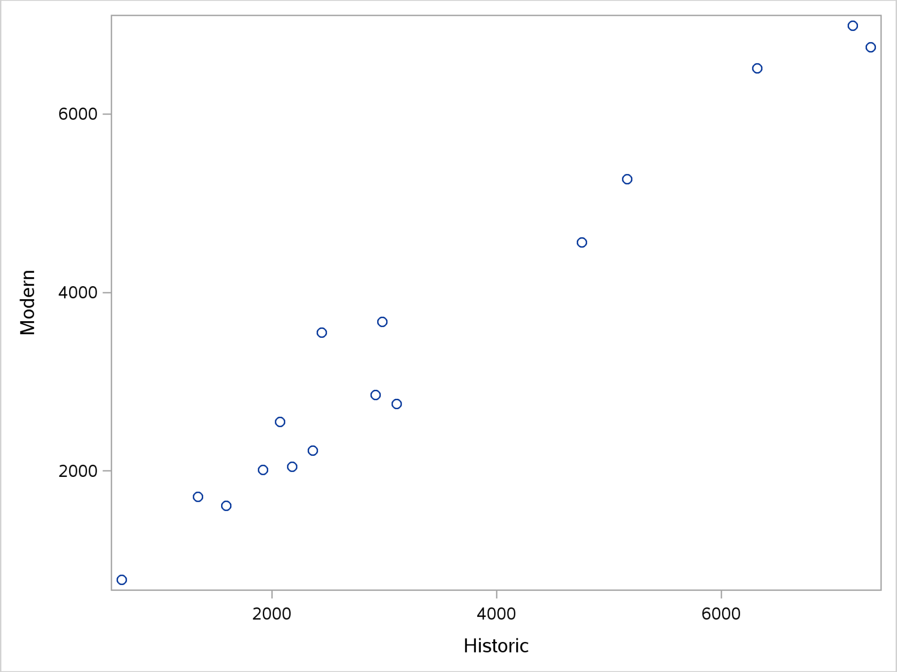
If you can do that and get it right, full credit, but there are a lot more places to go wrong (and thus lose points).

A simple `proc print` is one point, since it displays the region as well, which I didn't want.

(c) (3 marks) Give SAS code to produce an appropriate graph of this data set, bearing in mind that the column `Region` simply labels the regions and is not a variable as such.

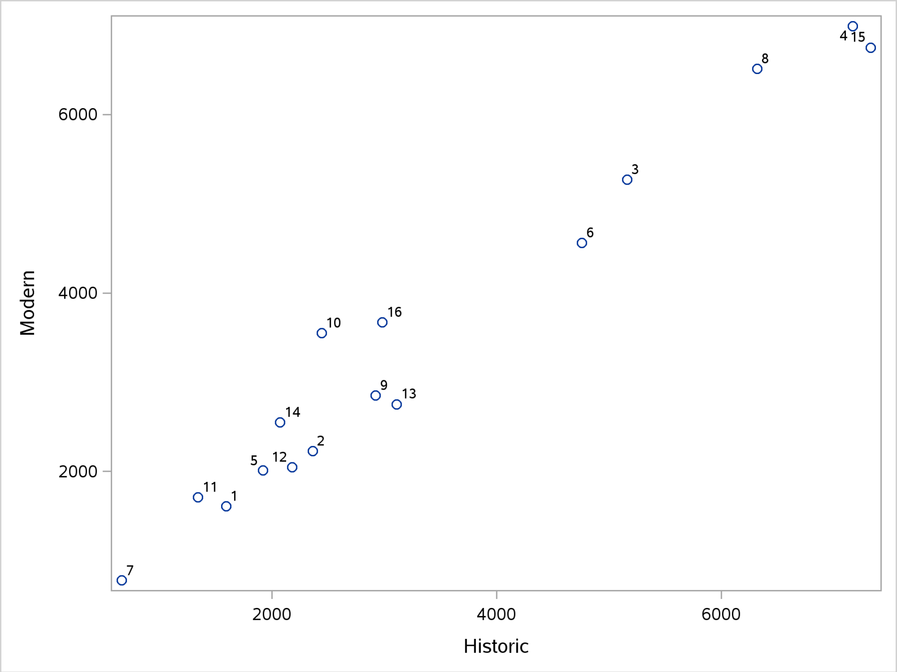
Solution: The last part of the question says "you don't need to include `Region` on your graph". There remain two quantitative variables, `Modern` and `Historic`, so a scatterplot is called for. I don't think there's any obvious response variable, so I would accept either variable on the y-axis. My *own* preference is to think of `Modern` as kind of a response to `Historic` for each region, so I would do it the way below, but if you have the variables the other way around, I'm fine with that too:

```
proc sgplot;
  scatter x=Historic y=Modern;
```


Modern
6000
4000
2000
2000
4000
6000
Historic

If you really wanted to include the Regions on the plot, the best way to do it would be as a `datalabel`, which goes like this:

```
proc sgplot;
  scatter x=Historic y=Modern / datalabel=Region;
```



If you can come up with this on an exam, I'm very impressed! (It's the same idea as on my solutions to the ecological footprint problem on Assignment 7, and if you've seen the Denali problem (wolf and caribou populations), it's there as well. In 2018, it was in the last lecture.)

Trying for the **datalabel** idea and getting it wrong is going to cost you a mark, because it won't then work. Thus, if you're not sure, you'll have make a choice between not plotting the labels at all (and possibly missing something I wanted), and trying to plot the labels (and possibly getting it wrong). The clue here is that this is the first question on the exam, which I prefer to have as something of a warmup, so the simpler solution that works is likely to be the one I'm after.

(d) (3 marks) Give SAS code to calculate the median and interquartile range of the modern and the historic grain production.

Solution: A custom `proc means`:

```
proc means median qrange;
  var Modern Historic;
```

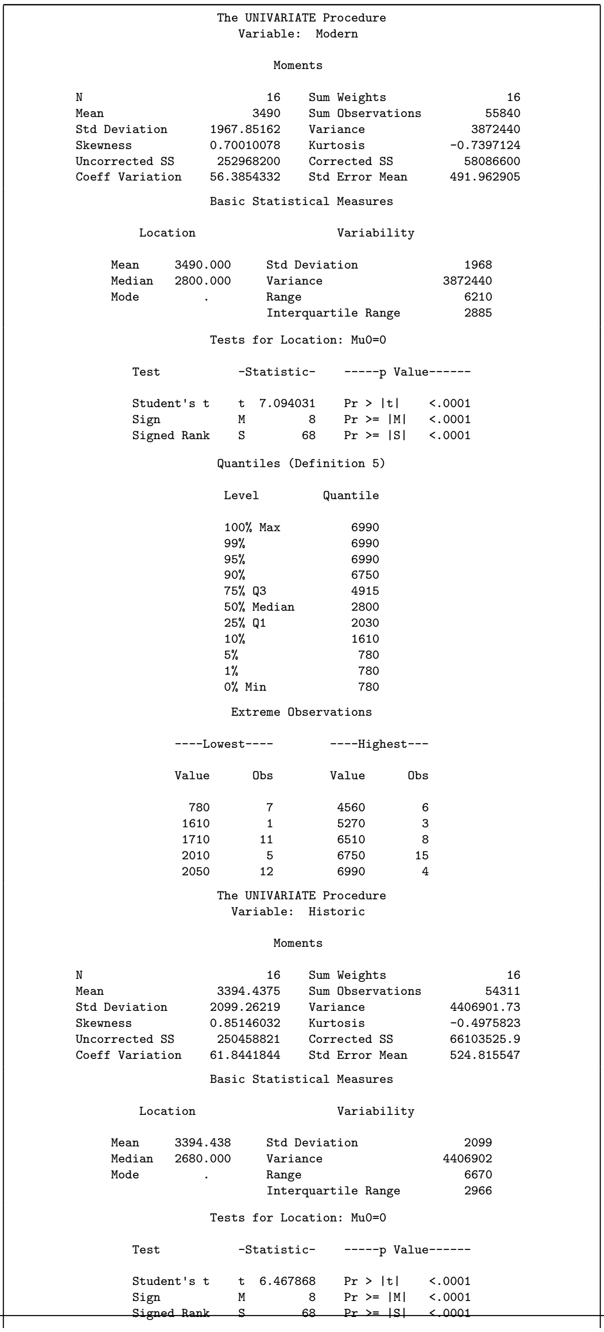
```
                         The MEANS Procedure

                                                Quartile
             Variable          Median              Range
             ----------------------------------------
             Modern           2800.00            2885.00
             Historic         2680.00            2965.50
             ----------------------------------------
```

You need the two statistics to calculate on the `proc means` line, and the two variables to calculate them for on the `var` line. The other variable `Region` is also numeric (though the numbers don't mean anything as numbers, but SAS doesn't know that), so without the `var` line you will incorrectly get the median and IQR of `Region` as well.

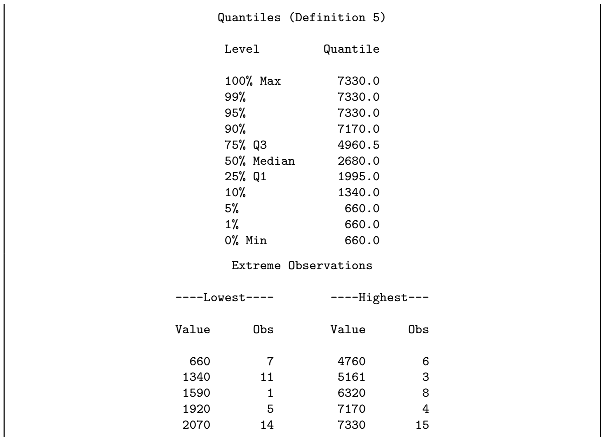
There is no categorical variable here, so there is no `class` anywhere here.

If you think `proc univariate` will do it, you will need to be precise because there is a lot of output:

```
proc univariate;
  var Modern Historic;
```



```
                         The UNIVARIATE Procedure
                            Variable:  Modern

                                 Moments

        N                          16    Sum Weights                 16
        Mean                     3490    Sum Observations         55840
        Std Deviation      1967.85162    Variance               3872440
        Skewness           0.70010078    Kurtosis            -0.7397124
        Uncorrected SS      252968200    Corrected SS          58086600
        Coeff Variation    56.3854332    Std Error Mean      491.962905
                       Basic Statistical Measures

             Location                    Variability

         Mean     3490.000     Std Deviation             1968
         Median   2800.000     Variance               3872440
         Mode         .        Range                     6210
                               Interquartile Range       2885
                       Tests for Location: Mu0=0

            Test           -Statistic-    -----p Value------

            Student's t    t  7.094031    Pr > |t|    <.0001
            Sign           M         8    Pr >= |M|   <.0001
            Signed Rank    S        68    Pr >= |S|   <.0001
                        Quantiles (Definition 5)

                         Level         Quantile

                         100% Max          6990
                         99%               6990
                         95%               6990
                         90%               6750
                         75% Q3            4915
                         50% Median        2800
                         25% Q1            2030
                         10%               1610
                         5%                 780
                         1%                 780
                         0% Min             780
                          Extreme Observations

                    ----Lowest----        ----Highest---

                    Value      Obs        Value      Obs

                      780        7         4560        6
                     1610        1         5270        3
                     1710       11         6510        8
                     2010        5         6750       15
                     2050       12         6990        4
                         The UNIVARIATE Procedure
                           Variable:  Historic

                                 Moments

        N                          16    Sum Weights                 16
        Mean                3394.4375    Sum Observations         54311
        Std Deviation      2099.26219    Variance            4406901.73
        Skewness           0.85146032    Kurtosis            -0.4975823
        Uncorrected SS      250458821    Corrected SS        66103525.9
        Coeff Variation    61.8441844    Std Error Mean      524.815547
                       Basic Statistical Measures

             Location                    Variability

         Mean     3394.438     Std Deviation             2099
         Median   2680.000     Variance               4406902
         Mode         .        Range                     6670
                               Interquartile Range       2966
                       Tests for Location: Mu0=0

            Test           -Statistic-    -----p Value------

            Student's t    t  6.467868    Pr > |t|    <.0001
            Sign           M         8    Pr >= |M|   <.0001
            Signed Rank    S        68    Pr >= |S|   <.0001
```



```
            Quantiles (Definition 5)

             Level         Quantile

             100% Max        7330.0
             99%             7330.0
             95%             7330.0
             90%             7170.0
             75% Q3          4960.5
             50% Median      2680.0
             25% Q1          1995.0
             10%             1340.0
             5%               660.0
             1%               660.0
             0% Min           660.0

              Extreme Observations

      ----Lowest----        ----Highest---

      Value      Obs        Value      Obs

        660        7         4760        6
       1340       11         5161        3
       1590        1         6320        8
       1920        5         7170        4
       2070       14         7330       15
```

If you are going to go this way, you have to say where I would need to look in the output. It's in the Basic Statistical Measures. Saying this is enough, but you need to have an example `proc univariate` output with you to get this from. One mark each for a `proc univariate` correctly coded for each variable (either separately or together), and one mark for some reasonably precise description of where in the voluminous output the grader should look.

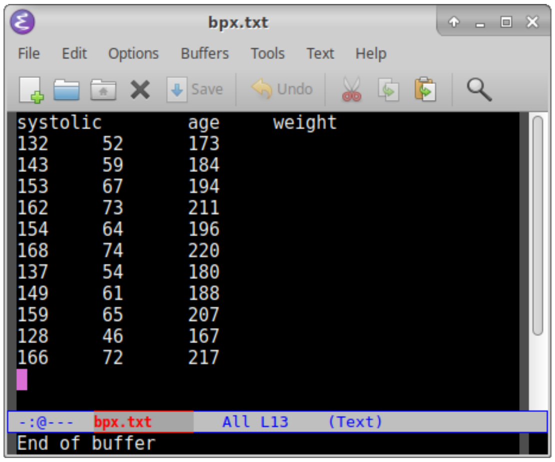
2. Blood pressure is measured with two numbers: the "systolic" (larger) number, and the "diastolic" (smaller) number. The systolic blood pressure is thought to depend on age and weight. Data are available for a sample of 11 men. The data are shown in Figure 2.

   (a) (2 marks) Give code to read the data *into an R data frame* called `bp`, as we did it in class. Assume that the data are in a file `bp.txt` in the same folder as your current project in R Studio.

Solution: "As in class" implies one of the `read_` functions, not something like `read.table`. These data values are separated by more than one space and aligned in columns (with at least one space in between all the way down), so `read_table` is the thing:

```
bp=read_table("bp.txt")

## Parsed with column specification:
## cols(
##   systolic = col_integer(),
##   age = col_integer(),
##   weight = col_integer()
## )

bp

## # A tibble: 11 x 3
##     systolic   age weight
##        <int> <int>  <int>
##  1       132    52    173
##  2       143    59    184
##  3       153    67    194
##  4       162    73    211
##  5       154    64    196
##  6       168    74    220
##  7       137    54    180
##  8       149    61    188
##  9       159    65    207
## 10       128    46    167
## 11       166    72    217
```

`read_tsv` is not quite the right thing here: if the data were separated by tabs, the numbers for eg. `age` would be aligned with the "a" of `age` and the "w" of `weight`. Or, even, this:



```
systolic        age     weight
132     52      173
143     59      184
153     67      194
162     73      211
154     64      196
168     74      220
137     54      180
149     61      188
159     65      207
128     46      167
166     72      217

-:@---  bpx.txt       All L13     (Text)
End of buffer
```

I tried it to see what would actually happen. The word "systolic" is actually 8 characters long, so pressing "tab" after it moves the heading `age` across 8 characters, on top of the *weight* values, and the heading `weight` is eight more characters further across. So as it happens the columns aren't even lined up.

(In case you are wondering, the screenshot is Emacs, but any text editor should give similar results, depending on how many spaces a tab is considered equivalent to, 8 in this case.)

Two points for `read_table`, done properly. One point for `read_tsv` or for `read_delim` with more than one space (neither of these will work, but they are sensible ideas). Nothing for anything else, including `read.table` (that will work, but it is not as we did it in class).

(b) (3 marks) Give R code to create a new column in your data frame called `weight_class` which has the values `heavy` if each man's weight is above the mean weight, and `light` otherwise. Use `tidyverse` ideas. (You do not need to save the new data frame.)

Solution: This uses `mutate`. `ifelse` is the cleanest way to go:

```
bp %>% mutate(weight_class=ifelse(weight>mean(weight),"heavy","light"))
```

```
## # A tibble: 11 x 4
##    systolic   age weight weight_class
##       <int> <int>  <int> <chr>
##  1      132    52    173 light
##  2      143    59    184 light
##  3      153    67    194 light
##  4      162    73    211 heavy
##  5      154    64    196 heavy
##  6      168    74    220 heavy
##  7      137    54    180 light
##  8      149    61    188 light
##  9      159    65    207 heavy
## 10      128    46    167 light
## 11      166    72    217 heavy
```

If you like, work out the mean weight first and save it:

```
m=mean(bp$weight)
bp %>% mutate(weight_class=ifelse(weight>m,"heavy","light"))
```

```
## # A tibble: 11 x 4
##    systolic   age weight weight_class
##       <int> <int>  <int> <chr>
##  1      132    52    173 light
##  2      143    59    184 light
##  3      153    67    194 light
##  4      162    73    211 heavy
##  5      154    64    196 heavy
##  6      168    74    220 heavy
##  7      137    54    180 light
##  8      149    61    188 light
##  9      159    65    207 heavy
## 10      128    46    167 light
## 11      166    72    217 heavy
```

Or use `case_when`, which goes this way:

```
bp %>% mutate(weight_class=case_when( weight>mean(weight) ~ "heavy",
                                      TRUE                ~ "light"))
```

```
## # A tibble: 11 x 4
##    systolic   age weight weight_class
##       <int> <int>  <int> <chr>
##  1      132    52    173 light
##  2      143    59    184 light
##  3      153    67    194 light
##  4      162    73    211 heavy
##  5      154    64    196 heavy
##  6      168    74    220 heavy
##  7      137    54    180 light
##  8      149    61    188 light
##  9      159    65    207 heavy
## 10      128    46    167 light
## 11      166    72    217 heavy
```

Indentation and alignment is up to you.

If you can't get either of those, this is second best:

```
bp %>% mutate(weight_class2=(weight>mean(weight)))
```

```
## # A tibble: 11 x 4
##    systolic   age weight weight_class2
##       <int> <int>  <int> <lgl>
##  1      132    52    173 FALSE
##  2      143    59    184 FALSE
##  3      153    67    194 FALSE
##  4      162    73    211 TRUE
##  5      154    64    196 TRUE
##  6      168    74    220 TRUE
##  7      137    54    180 FALSE
##  8      149    61    188 FALSE
##  9      159    65    207 TRUE
## 10      128    46    167 FALSE
## 11      166    72    217 TRUE
```

This, as you see, provides true and false values instead of `heavy` and `light`, so it doesn't quite do the right thing. 2 points for that, if you code it correctly.

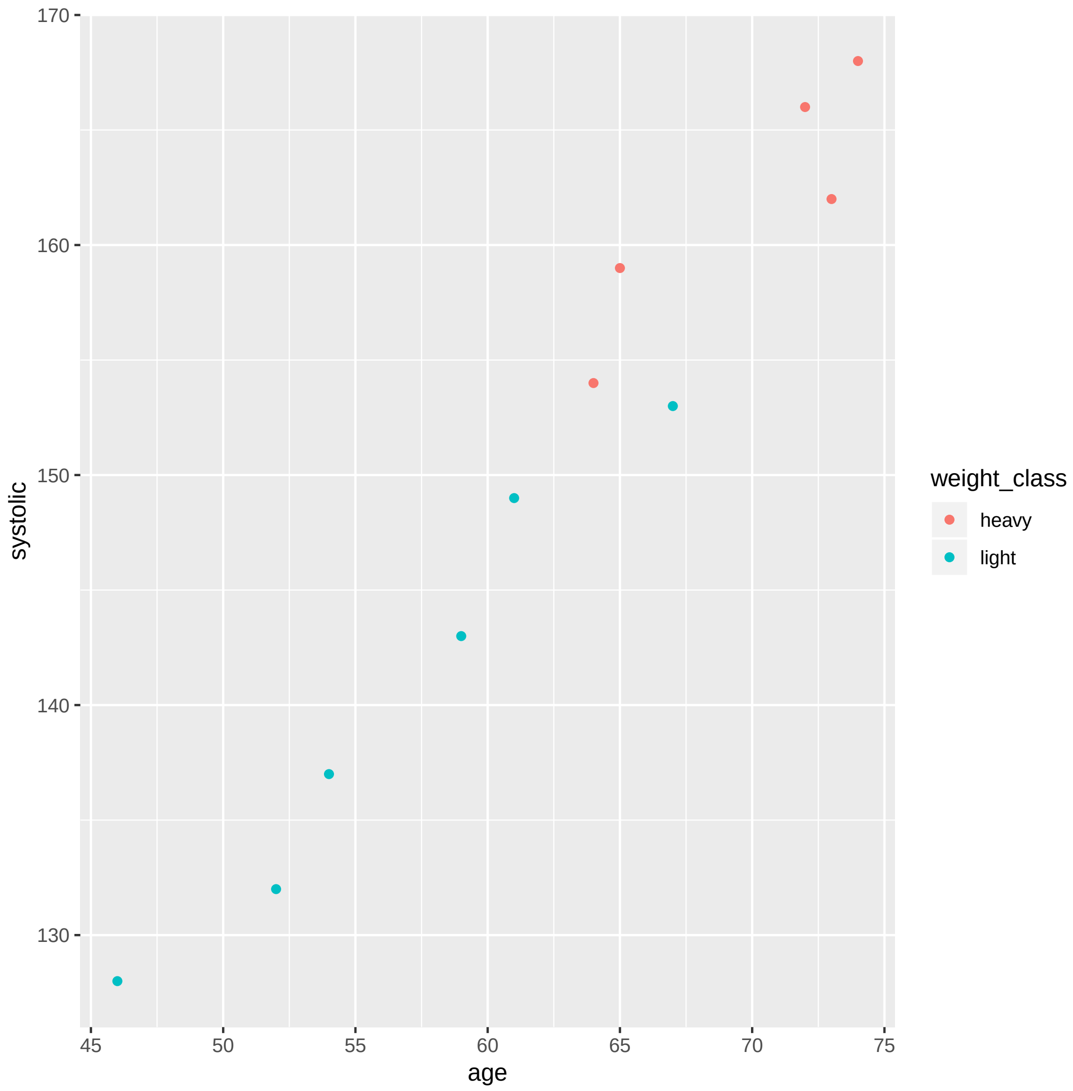
Save the new data frame if you want, or not. I have no preferences either way.

The usual minus one per error, with a minimum of one if you have something of value correct.

(c) (3 marks) Suppose that the data frame you created in the previous part is saved in data frame `bp2`. Give R code that uses this data frame to plot systolic blood pressure against age, distinguishing the points by whether each man is lighter or heavier than the mean.

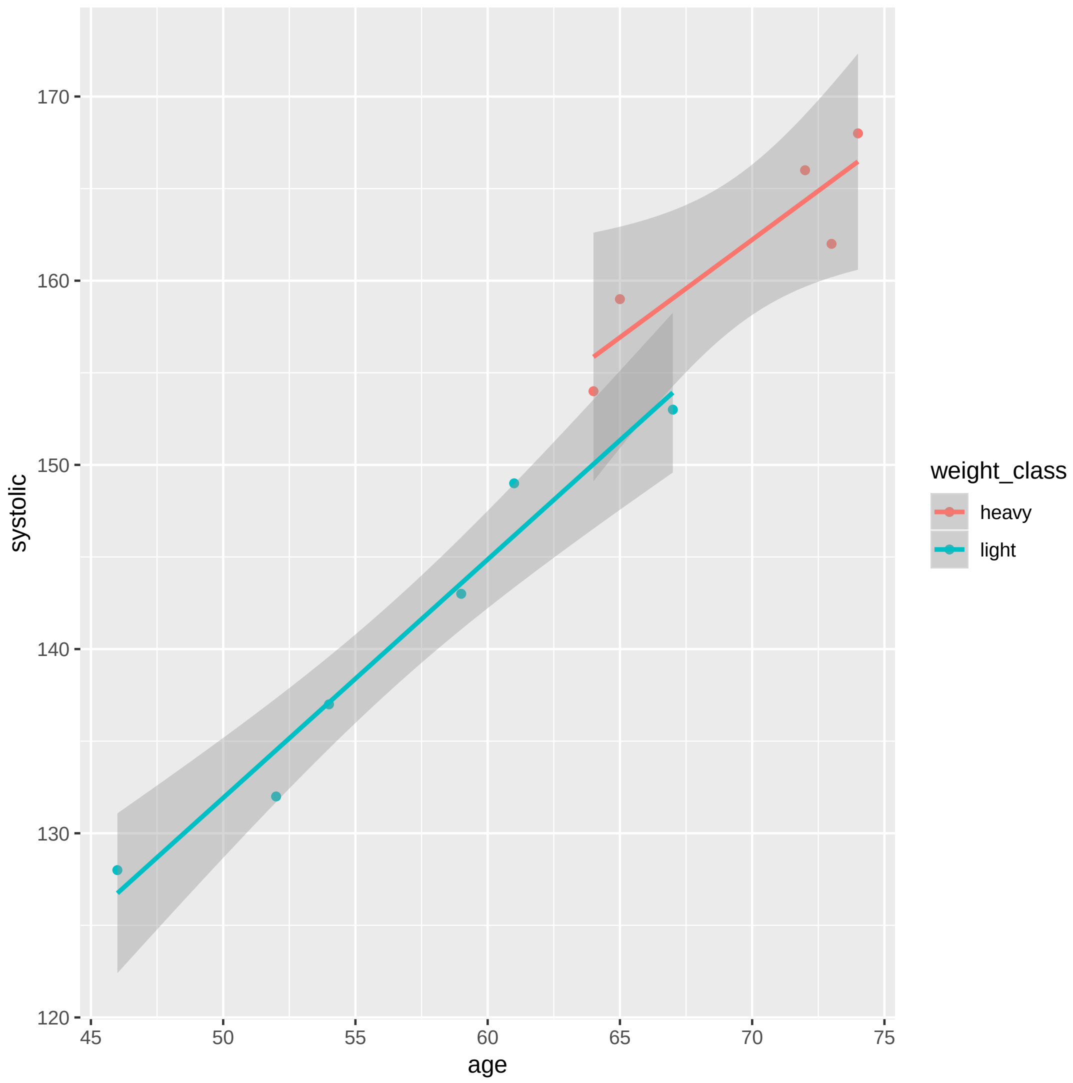
Solution: Systolic blood pressure and age are both quantitative, so this calls for a scatter plot with the other, categorical, value (the one we called `weight_class`) being distinguished by colour (easiest):

```
ggplot(bp2, aes(x=age, y=systolic, colour=weight_class))+geom_point()
```



Feel free to add a regression line or a smooth trend. I don't need it, but if you add it correctly I'm good with it. (In this case, you'll get a separate regression line or smooth curve for *each* weight class.) For example:

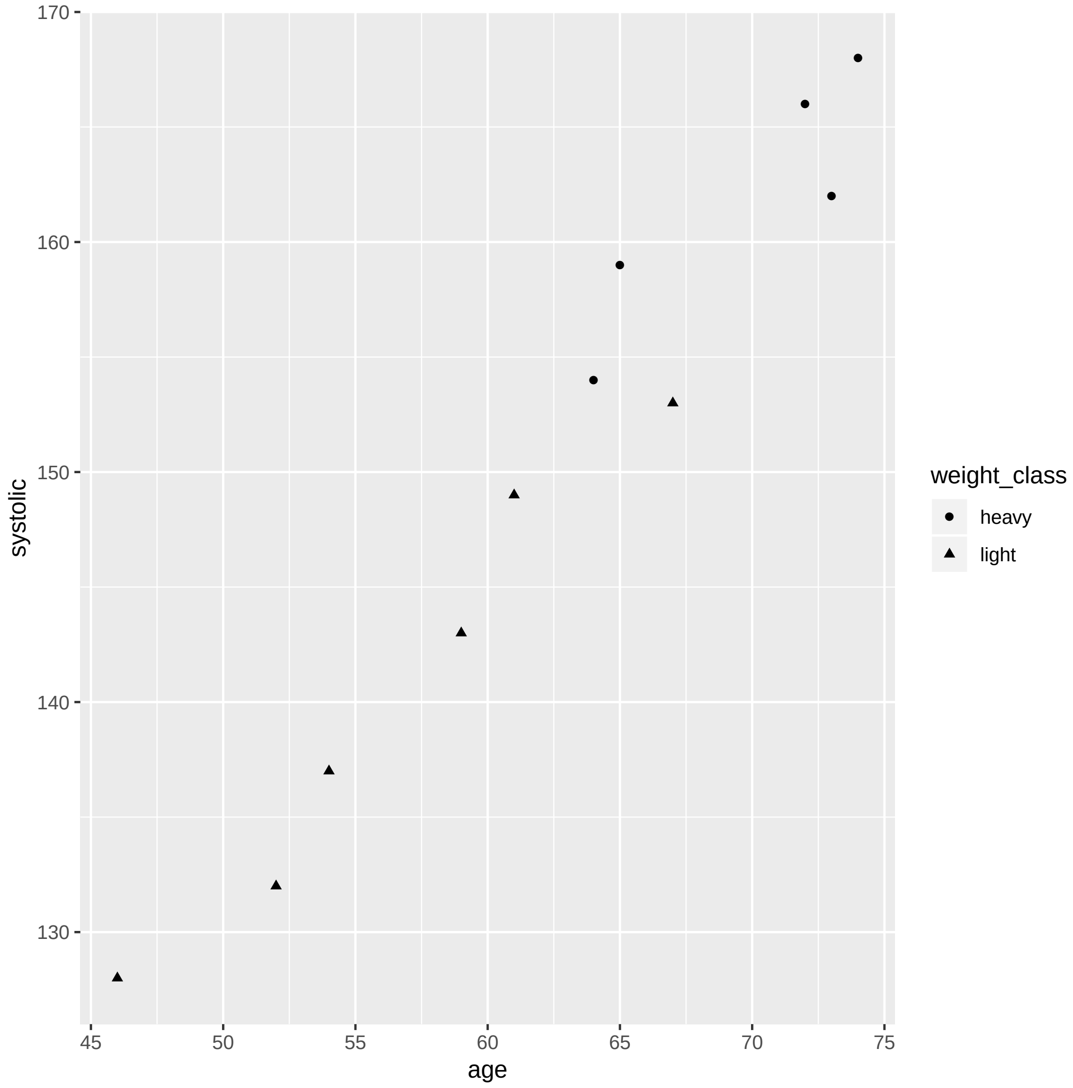
```
ggplot(bp2, aes(x=age, y=systolic, colour=weight_class))+geom_point()+
    geom_smooth(method="lm")
```



(The heavier men are mostly older and have higher blood pressure.)

Distinguishing the weight classes this way also works:

```
ggplot(bp2, aes(x=age, y=systolic, shape=weight_class))+geom_point()
```



Not, to my mind, as aesthetic, but it works, so it gets the points.

Exam technique: if you couldn't make the column `weight_class` in the previous part, for *this* part, you should assume that you *could* do it, and therefore for this part you *do* have a column called `weight_class` in a data frame `bp2` ready to use.

(d) (4 marks) Give R code to run a multiple regression predicting systolic blood pressure from age and weight, and to display the results, including output like Figure 3. If output like what you see in the Figure would be produced by your code, I don't mind what other output would be produced. You should use the (quantitative) weight as read in from the file, not the categorical `weight_class` that you created earlier.

Solution: These are the two lines I used:

```
bp.1=lm(systolic~age+weight, data=bp)
summary(bp.1)
```

```
##
## Call:
## lm(formula = systolic ~ age + weight, data = bp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0708 -1.1806 -0.2567  0.9251  2.7027
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.5126     9.8476   2.286  0.05158 .
## age           0.5795     0.2290   2.531  0.03522 *
## weight        0.4704     0.1173   4.011  0.00389 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.803 on 8 degrees of freedom
## Multiple R-squared:  0.986,Adjusted R-squared:  0.9825
## F-statistic: 281.7 on 2 and 8 DF,  p-value: 3.843e-08
```

You can call the fitted model object whatever you like.

The key is to recognize that the Figure contains part of what you would see from `summary`, so that `summary(bp.1)` would get that and other things, which I said is OK. The reason for doing it this way (and not showing you the whole output from `summary`) is that the full output contains a line `Call` which gives away the answer.

This also works, `tidy` being from package `broom` which I said (on the front page of the exam) that you could assume had been loaded:

```
tidy(bp.1)
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)   22.5       9.85       2.29 0.0516
## 2 age            0.580     0.229      2.53 0.0352
## 3 weight         0.470     0.117      4.01 0.00389
```

This looks a little different, but it contains all the same information, so I am good with this too.

Grading: 3 points for the `lm` line, minus one per error down to a minimum of 1 if you have something that looks like a regression. One point for the summary line (`summary` of whatever you called the fitted model object, my `bp.1`), or `tidy`. If you have a summary of something *else*, and not this table, no points.

(e) (3 marks) Part of the output from your regression is shown in Figure 3. Imagine that you have two men of the same (unknown) age, but the second man is 10 pounds heavier than the first man. What precisely does the regression output tell you about how their systolic blood pressures will compare on average? Explain briefly.

Solution: This is backwards from the way I usually ask these. Often I ask for the meaning of, say, the value 0.4704, and you have to say "if the weight of a man increases by one pound, all else equal, the systolic blood pressure is predicted to increase by 0.47".

To answer this question, you need to understand that and how it helps you to answer this question: the ages are the same, but to get the weight of the second man, you take the weight of the first and add 10 pounds to it, so their predicted systolic blood pressures will differ by 10 times the slope, that is, 4.7. This will be true no matter what the age of the two men is, and no matter what actual weights they are, because the relationship is assumed linear (so that the slopes are constant).

If you don't see this, pick an age and a weight (any ones will do), predict `systolic` for it, increase the weight by 10, and predict again. It doesn't matter what age and weight you pick (so you ought to pick more than one or explain why it doesn't matter). Use your calculator for this. I'll do a couple, in R because I am lazy:

```
new=crossing(weight=c(180,190), age=c(50,55))
new
```

```
## # A tibble: 4 x 2
##   weight   age
##    <dbl> <dbl>
## 1    180    50
## 2    180    55
## 3    190    50
## 4    190    55
```

and then

```
pred=predict(bp.1,new)
bind_cols(new,pred=pred)
```

```
## # A tibble: 4 x 3
##   weight   age  pred
##    <dbl> <dbl> <dbl>
## 1    180    50 136.2
## 2    180    55 139.1
## 3    190    50 140.9
## 4    190    55 143.8
```

If you increase the weight by 10 while leaving age fixed (at either 50 or 55, it doesn't matter), you increase the predicted systolic blood pressure by $140.9 - 136.2 = 143.8 - 139.1 = 4.7$.

If you do that calculation, and see the number 4.7 come out again, you might recognize where it came from.

Grading: a point for recognizing that it has something to do with the 0.47. A point for recognizing that you have to multiply something by 10. A point for recognizing that the heavier (second) man will have the higher blood pressure (on average). If you get to the final answer correctly, it implies that you have all of these, so full marks. If you do predictions, two points if you correctly pick one age and predict for two weights ten pounds apart; three points if you in addition make some kind of assertion that it doesn't matter what age you pick, or if you do a prediction for two or more ages (the implication being that if it works for two or more ages, it is likely to work for all ages).

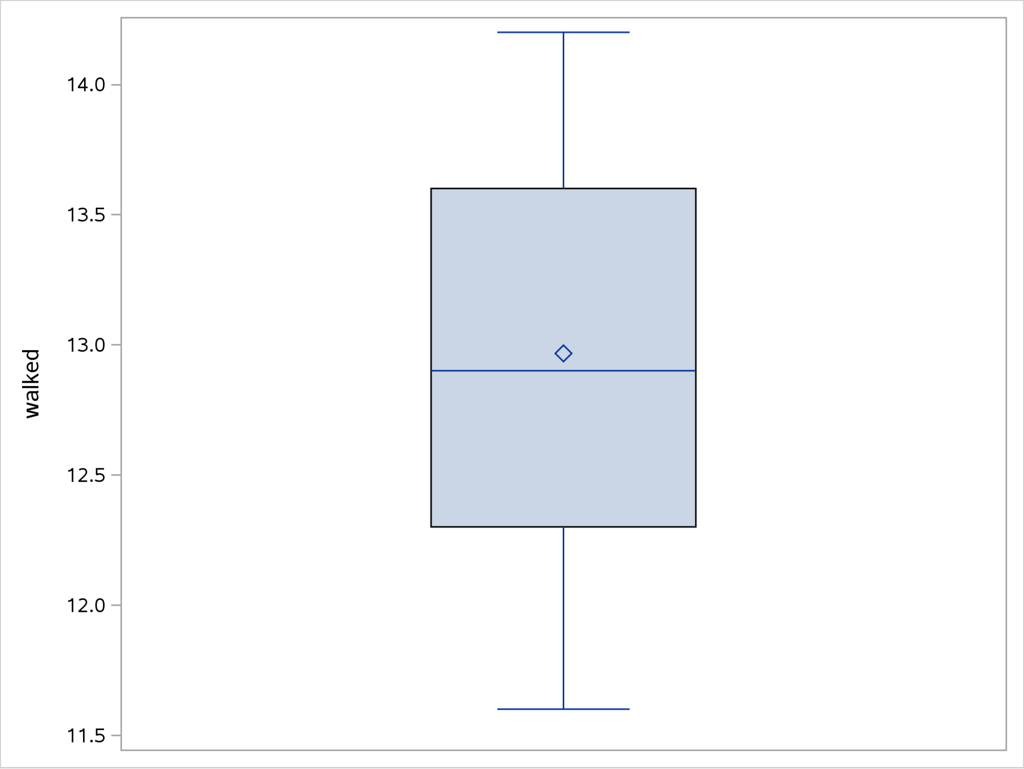
As I conceived this question, it had nothing to do with the P-values, but I realized that it was reasonable to check the P-values for significance to see whether `weight` had any effect at all. Thus an answer that made the point that the blood pressure was going to be different (on average) between the two men seemed to be worth something, and it got one point. If you wanted all three, you had to tell me *how much* different.

3. A child psychologist reads in the literature that children typically start walking at 12.5 months. The child psychologist believes this is incorrect (as an average for all children), so she takes a sample of 18 children and measures the age (in months) at which each child starts walking. The data are shown in Figure 4. This is a SAS data set called `walking` with the column names as shown. Assume that it is the most recently created data set.

   (a) (2 marks) Give SAS code to produce the graph in Figure 5.

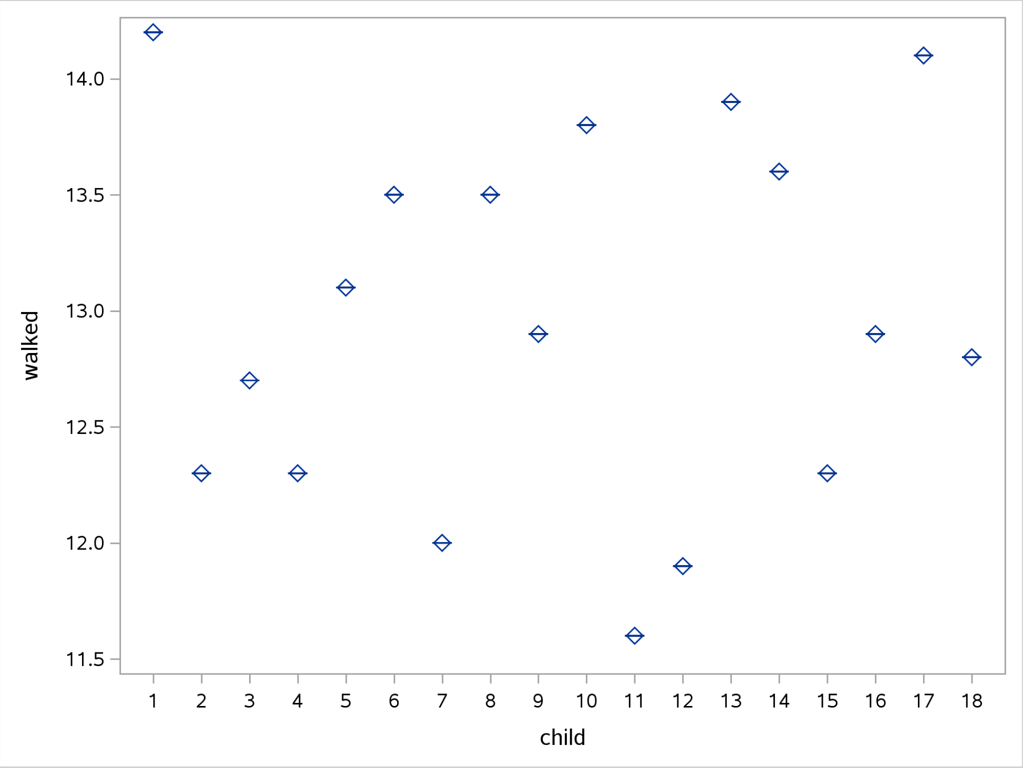
Solution: This is a one-group vertical boxplot. (I might have drawn a histogram instead, but I didn't.) My code was this:

```
proc sgplot;
  vbox walked;
```



Two marks for that, one for something like that with one or two errors, zero for anything else. (Missing semicolons are errors. The grader might forgive you one, but not two, missing semicolons.) Because there is no categorical grouping variable, there is no `category=`, so to have one is an error. For example, if you try `vbox walked / category=child`, using the only other variable in the data set, you'll get eighteen miniature "boxplots", one for each child:

```
proc sgplot;
  vbox walked / category=child;
```



Not what I showed you at all.

(b) (4 marks) Use Figure 5 along with the output in Figure 6 to assess the evidence for the child psychologist's belief. What do you conclude, in the context of the data? Make it clear which parts of Figure 6 you are using.

Solution: There are two (actually three, but we only use two) tests in the `Tests for Location` section of the `proc univariate` output. To decide whether to use the sign test or the t-test, look at the boxplot. This is very much symmetric with no outliers, so there is no reason to look further than the t-test for the mean. (The difference between the mean and median is inconsequential. Figure 6 shows that the mean is 12.97 and the median is 12.9.)

Now look in Figure 6 for the Tests for Location. The P-value for the t-test is 0.0234. This is two-sided, which is what we want because the child psychologist is looking for *any* difference from 12.5. Thus we reject the null hypothesis (that the mean walking age is 12.5 months) in favour of a two-sided alternative (that the mean walking age differs from 12.5 months).

Grading: One point for making a reasoned choice of t or sign from the boxplot (more generally, for preferring to use the mean or median). One point for picking the corresponding test from Figure 6 and giving its P-value. One point for making a decision to accept or reject the null hypothesis on the basis of that P-value (a pretty easy one). One point for stating a conclusion about mean (or median) ages at which children learn to walk.

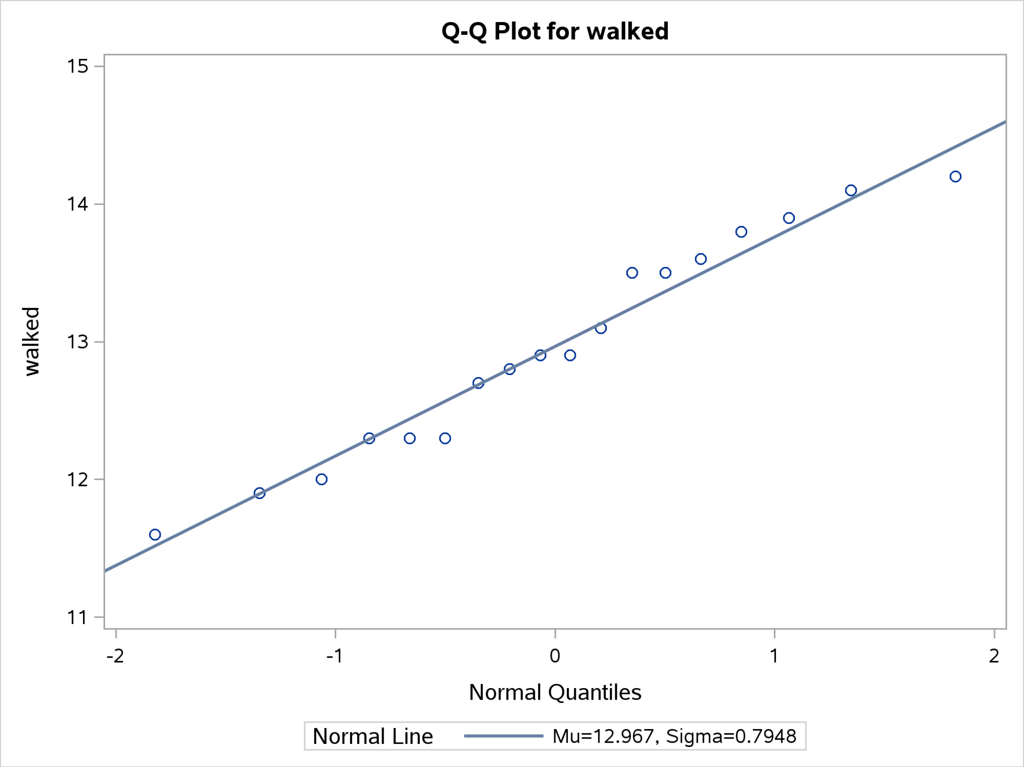
For example, if you choose to use the sign test, you will lose a mark for that choice (I don't think you can make a case for using it here), but if you follow through, stating your conclusion like "there is no evidence that the median walking age differs from 12.5 months", you can get 3 points.

You need to do a hypothesis test here, because I said "for all children" in the question. This implies inference from this sample to a larger population. Looking at the sample mean (or median) and comparing *that* to 12.5 doesn't answer the question, because this sample could have come out above 12.5 by chance, and only the test will tell you whether 12.97 was high enough to rule out chance as an explanation. I think, if you make a reasoned choice of mean or median from the boxplot and compare that to 12.5, that's one mark altogether.

Extra: you might have been wondering why the t-test and the sign test came out so different. My guess is that the data are very much normal in shape, so the t-test will be more powerful than the sign test (because of the way the t-test uses the data). If you check the `Quantiles` at the bottom of Figure 6, you can see that even the most extreme data values are well within 2 standard deviations of the mean, so that if anything the data is *short*-tailed compared with the normal:

```
proc univariate noprint;
  qqplot walked / normal(mu=est sigma=est);
```

I put a comma in there again, the first time, and of course it didn't work.


Q-Q Plot for walked
walked
15
14
13
12
11
-2
-1
0
1
2
Normal Quantiles
Normal Line
Mu=12.967, Sigma=0.7948

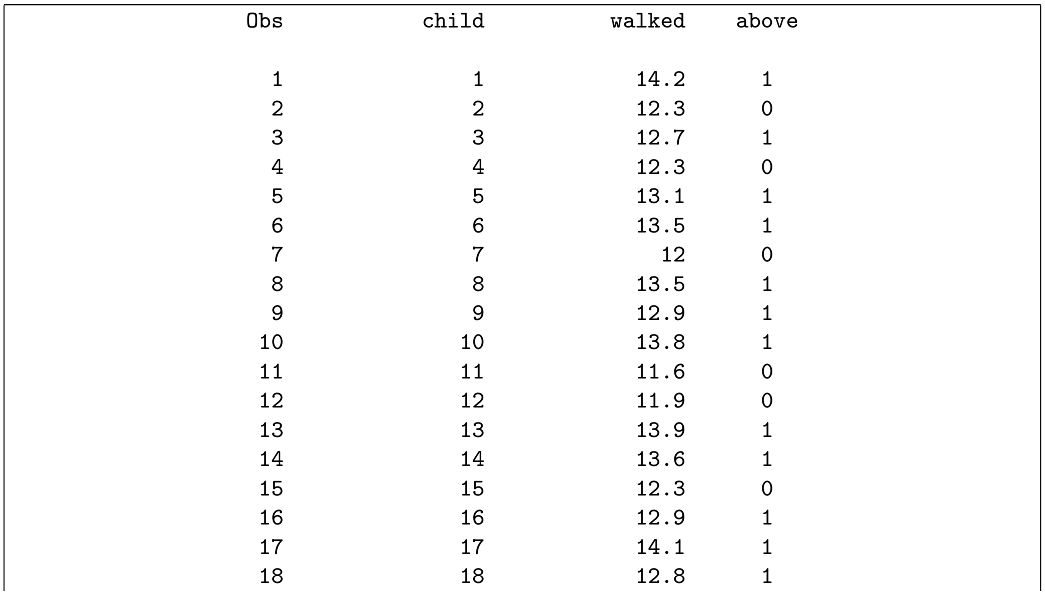
Very much normal. If you estimate `sigma` using the IQR, you get

```
1.3/1.35

## [1] 0.962963
```

which is close to the sample SD, actually a little bigger.

The sign test uses a count of data values above and below the null median, 12.5:

```
data walking2;
  set walking;
  above=(walked>12.5);

proc print;
```



```
                     Obs          child          walked    above

                       1              1            14.2      1
                       2              2            12.3      0
                       3              3            12.7      1
                       4              4            12.3      0
                       5              5            13.1      1
                       6              6            13.5      1
                       7              7              12      0
                       8              8            13.5      1
                       9              9            12.9      1
                      10             10            13.8      1
                      11             11            11.6      0
                      12             12            11.9      0
                      13             13            13.9      1
                      14             14            13.6      1
                      15             15            12.3      0
                      16             16            12.9      1
                      17             17            14.1      1
                      18             18            12.8      1
```

(above is 1 if the data value is above 12.5 and 0 otherwise) and now we count those:

```
proc freq;
  tables above;
```

```
                            The FREQ Procedure

                                                Cumulative    Cumulative
           above    Frequency     Percent     Frequency      Percent
           ---------------------------------------------------------
               0           6       33.33             6        33.33
               1          12       66.67            18       100.00
```

How unlikely a split is 12–6? SAS's P-value says it's not very unlikely, even if the median were 12.5. In R we would find the probability of 6 or less (or of 12 or more) and double it because the test is two-sided:

```
2*sum(dbinom(0:6,18,0.5))

## [1] 0.2378845
```

or

```
2*sum(dbinom(12:18,18,0.5))

## [1] 0.2378845
```

Those are the P-value that SAS got. With 18 observations, we need a more uneven split than this to get significance:

```
2*sum(dbinom(0:5,18,0.5))

## [1] 0.09625244
```

A 13–5 split doesn't quite do it, but a 14–4 split would:

```
2*sum(dbinom(0:4,18,0.5))

## [1] 0.03088379
```

What seems to be happening here is that this is a case where the t-test is the best one (the data are definitely normal enough), and so the t-test is more powerful than the sign test is. If you used the sign test, you would fail to prove that the median differs from 12.5, when you could have proved that the (almost identical) mean *does* differ from 12.5.

I didn't know this one would work this way when I first looked at the data, but I'm actually quite happy that it did. When you're comparing the pooled and Welch-Satterthwaite t-tests, the latter is often a "safe" choice no matter what (see the fox rabies question later), but when your choice is between one-sample t and sign, there is no safe choice: you have to look at the plot and pick the best one, otherwise you could lose out.

4. Rabies is a deadly virus that spreads to people from the saliva of infected animals. The rabies virus is usually transmitted through a bite. Animals most likely to transmit rabies include bats, coyotes, foxes, raccoons and skunks.

   A study was made of rabies cases in foxes in southern Germany. 31 locations were sampled altogether in two regions, labelled A and B here. The data, in a SAS data set called `foxrabies`, are shown in Figure 7. The values in `cases` are counts of the number of cases of rabies at that location. Some analysis is shown in Figure 8.

   (a) (2 marks) Suppose that one interest of the researchers who collected the data was to see whether the mean number of cases per location exceeded 3, regardless of region. Explain briefly why Figure 8 *does not* enable you to assess the evidence for this.

   Solution: Figure 8 is a two-sample t-test, for comparing the two regions *with each other*, not comparing all the data with some external mean like 3. To do that, we need a *one-sample* t-test.

   Two points: one for "this is a two-sample test", one for "but we need a one-sample test". Or anything logically equivalent. If you say "we need a one-sample test and the output is not that", I'm OK with that, but only one point for "the output is for the wrong test" with no further comment. Some people gave a very nice explanation of this being a two-sample test, but I could only give it one point without something to convince me of why we needed something else.

   The issue here is not one of one-*sided* vs. two-*sided*, or of whether the data have a sufficiently normal distribution. The immediate issue is that this is the *wrong test* for the situation, so that's what you need to grab first.

   I was happy with a consideration of means rather than of test, since this would enable us to get to the right place: the Figure gives the means separated by region, and we need the mean of all the data together.

   Somebody said "we need to regardless the regions". I've never seen "regardless" as a verb before, but I was happy with it since it was perfectly clear what it meant.

   (b) (3 marks) Give SAS code that will produce output to allow the researchers to assess their interest of the previous part. If your code will produce output that almost addresses the researchers' interest but does not quite do so, explain briefly what you will need to do with the output from your code. You may assume that the distribution of the number of cases is approximately normal.

   Solution: Thus, a one-sample t-test of all the `count` values, testing the null hypothesis that the mean is 3 against the one-sided alternative that it is greater. That translates into this code:

```
proc ttest h0=3 sides=U;
  var cases;
```

   `side` or `sides` is equally good. I've used both in my notes. The output turns out to be this:

```
          N        Mean     Std Dev     Std Err     Minimum     Maximum

         31      4.3548      2.6274      0.4719      1.0000      9.0000
              Mean       95% CL Mean        Std Dev      95% CL Std Dev

            4.3548      3.5539  Infty        2.6274      2.0996   3.5120
                                      DF    t Value    Pr > t

                                      30       2.87    0.0037
```

It so happens that the mean number of cases per location *does* significantly exceed 3, the P-value being 0.0037. This is the kind of test that might be done if the historical cases per location was 3, and the researchers wanted to see whether there were more cases now (as a result of changes in farming practices, climate change or whatever).

The extra wording in the question was because you might have thought to use `proc univariate` instead, and take the t-test from there:

```
proc univariate mu0=3;
  var cases;
```

Any other variant that produce a null-hypothesis mean to test, like `location`, is equally good. With `proc univariate`, `h0` *does not work*, so you will lose a point if you wrote that.

The relevant piece of the output is this:

```
                    The UNIVARIATE Procedure
                        Variable:  cases

                   Tests for Location: Mu0=3

         Test           -Statistic-    -----p Value------

         Student's t    t  2.871061    Pr > |t|    0.0074
         Sign           M       4.5    Pr >= |M|   0.1221
         Signed Rank    S      97.5    Pr >= |S|   0.0150
```

Two marks for that code if you do it right. The third mark is for realizing that this P-value will be two-sided. Strictly, you should check that you are on the correct side, but I would accept simply saying that you have to halve the P-value for the last point.

Two marks also for using `proc ttest` otherwise correctly without the `sides`. To get the third, you likewise need to halve the (two-sided) P-value that comes out. If you do that, you're good. (The principle is that if you find some way to get a 1-sided P-value, I'm happy.)

You could read this as simply requiring the mean of all the locations, if you said in (a) that this was what we needed to look at. But there was an implication that "exceeded 3" was an inference to a larger population ("all possible locations"), requiring a test or confidence interval, rather than a simple calculation. I would have used a word like "calculate" here for something like obtaining an overall mean; "assess" implies that more work than this needs to be done. I could have been more picky in (a) about this; yes, the Figure does show the mean number of cases per location by region, but the reason for obtaining output like that is to do a (two-sample) t-test, otherwise you might as well have just run `proc means`. To be consistent with that, the best answer here is a t-test as well. I think a `proc means` here is only one point.

(c) (4 marks) What is the main conclusion that you *can* draw from Figure 8? Explain briefly. Assume that anything needing to be approximately normal *is* approximately normal; if you need to make any other assumptions, make them and state what additional assumptions you make.

Solution: I am trying to give you as few clues as possible, because I want to see what you can deduce.

I think the main conclusion is in the P-value of the two-sample t-test. This is testing the null hypothesis that the two regions have the same mean, against the alternative that they have different means. We know this is a two-sided test because we have “proper” confidence intervals that do not go off to infinity. Another clue is that there is nothing in the question about, say, Region A possibly having more cases than Region B. The last thing is about whether we should do a pooled test or Satterthwaite-Welch. You can make a call about this by looking at the SDs of the two regions in the top two lines of the output. I would say these are very similar, something that is supported by the test at the very bottom not being significant. Any of these pieces of reasoning would satisfy me:

- the sample SDs are very similar (or not significantly different), so go with the pooled test and get a P-value of 0.3962
- Look at the “folded F” test at the bottom of the output first; since this fails to reject equal variances, go with the pooled test, P-value 0.3962
- the Satterthwaite-Welch test is at least pretty good even if the population SDs are the same, so it is safe to go with Satterthwaite and get a P-value of 0.3940
- the two P-values, 0.3962 and 0.3940, are very similar, so it doesn’t matter which one we use.
- Add “equal variances” as an extra assumption, and go with the pooled test, P-value 0.3962. (This is fine because in practice this can be checked, and if it turns out to be wrong, you can revisit which test you look at.)

You will notice that the “safety” argument will work for any two-sample t-test, at least with me, which means that you don’t even have to compare the spreads!

Whichever P-value you choose, it is not by any means small, so there is no evidence of a difference in the mean number of cases per location between the two regions.

Grading: 4 points for deducing a two-sided test, making a reasoned choice about the appropriate test, deciding on rejection or not, and stating a conclusion in terms of the data (one for each of those is the guideline). I used the word “conclusion” because I wanted you to say something about whether the regions were different or not.

If you work with the appropriate confidence interval instead, making a reasoned choice of which one to use and saying something like “the difference in mean number of cases between region A and region B is between -1.12 (-1.11) and 2.76 (2.75) with 95% confidence”, your maximum is 3 *unless* you also note that the confidence interval contains zero and therefore there is no evidence of a difference (or, the difference could go either way). This last would be “drawing a conclusion”.

I have a feeling I will be grading this one myself! (I did.)

(If you go the CI way, it’s easier to get 3 points, but it’s harder to get 4.)

If you wrote something here, I tried very hard to find you one point even if you hit none of the other items above. This is even true if you erroneously thought that *this* test had something to do with the means being 3.

(d) (3 marks) Look at Figure 9. What do you conclude about the appropriateness of a t procedure here? Explain briefly.

Solution: For a two-sample t-test such as this, *both* distributions need to be approximately normal in shape (but see below).

Analysis:

for group A, you could make these points:

- declare this to be normal enough, with the points being close enough to the line. (This was not my initial reaction, but the way the Figure came out, I couldn't contest this.)
- declare it to be non-normal, because of a S-curve, or because of "outliers". But you need to be very careful about calling the points off the line "outliers": these ones are *not extreme enough* compared to the normal, so the distribution has *short* tails compared to the normal. Real outliers, the kind that would damage a t-test, would go with points that are too extreme, too high or too low. So I don't think you can declare group A to be non-normal in a way that would cause problems with a t-test.
- Group A is *not* skewed since for that you require short tail at one end and a long tail at the other (too bunched up at one end, too spread out at the other).

For group B:

- declare this also to be normal enough, with the points close enough to the line.
- declare the distribution to be skewed to the right, with a curve on the normal quantile plot
- say that you have outliers at the top end (one or two).

Then you need an overall conclusion. If you were happy enough with *both* groups A and B, then you should be happy enough with the t-test. If *either one* of A and B are non-normal for reasons that cause problems, then the t-test is no good.

My opinion: I would say Region B is close enough to normal (the points are close to the line overall), but Region A is not: the distribution is too bunched up at the top and bottom. Thus you could say that the t-test we just did is inappropriate, because we need both distributions to be approximately normal and they are not. That would be a two-point answer.

To get the third point, look more closely at how region A is not normal: it has *short* tails. This says that the mean would be *perfectly good* as a measure of location, and therefore the two-sample t would be fine. Said differently, region A does not exhibit outliers or skewness, and *those* are the things that will damage a t-test. So neither group has any problems.

You could argue for the third point by saying that B shows a curved pattern, with the values at the bottom too bunched up and the ones at the top being too spread out. Or argue for two outliers at the top. I don't think the pattern is curved enough or the outliers are outlying enough to support this, but if you say that the t-test should not be done on this basis, that's a reasonable argument (if not, to my mind, a reasonable conclusion).

Another argument in favour of keeping the t-test is the combo of (i) the departure from normality is not too bad, or at least not too damaging (those short tails) and (ii) the P-value it gave is a long way from significance. It would take a gross failure of normality to make something like a Mood's median test give significance here, and we don't have *that.* That is to say, we might not trust the P-value, but the conclusion (don't reject) is sound. This argument, properly made, is also three points.

I gave a lot of twos to people who found problems but didn't articulate clearly enough why those problems would be damaging to the t-test, or who didn't state clearly enough whether they would trust the t-test or not, and why.

5. As the world population increases, it is important to grow enough grain to feed everyone. Sixteen large grain-producing regions were sampled. The production of grain (in kilograms per hectare) was recorded for each of these regions, both modern production and historical production. The historical production figures come from the mid-20th century. Do we have evidence that modern production is higher than historical production? The data are shown in Figure 1. We previously used this data set for another question.

   You may assume that you have a SAS data set called `grain` that contains the data from Figure 1, and that this is the most recently-created data set.

   (a) (2 marks) To compare modern and historic grain production, why would a matched pairs test be better than a two-sample test? Explain briefly.

   Solution: We are comparing modern and historic grain production in the *same* regions. Or, each region produces *two* grain production measurements, a modern one and a historic one.

   Two independent samples would be appropriate if we had different regions measured at different time periods. That wouldn't make much sense here, because it is much easier to measure changes by looking at the same region across times.

   Two marks or zero, probably.

   (b) (2 marks) Give SAS code to carry out a suitable matched-pairs t-test for these data.

   Solution:

   This kind of thing. You don't need the `data=grain`, but no harm if it is there:

```
proc ttest data=grain;
  paired Modern*Historic;
```

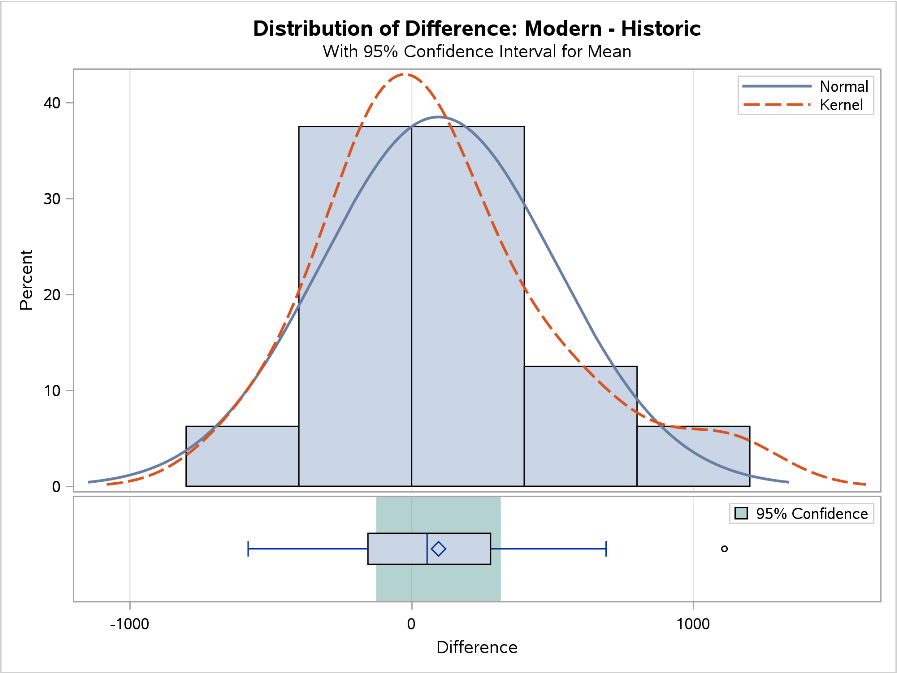
   Having the two variables the other way around is fine (the comparison will then be historic minus modern, which we would expect to be mostly negative).

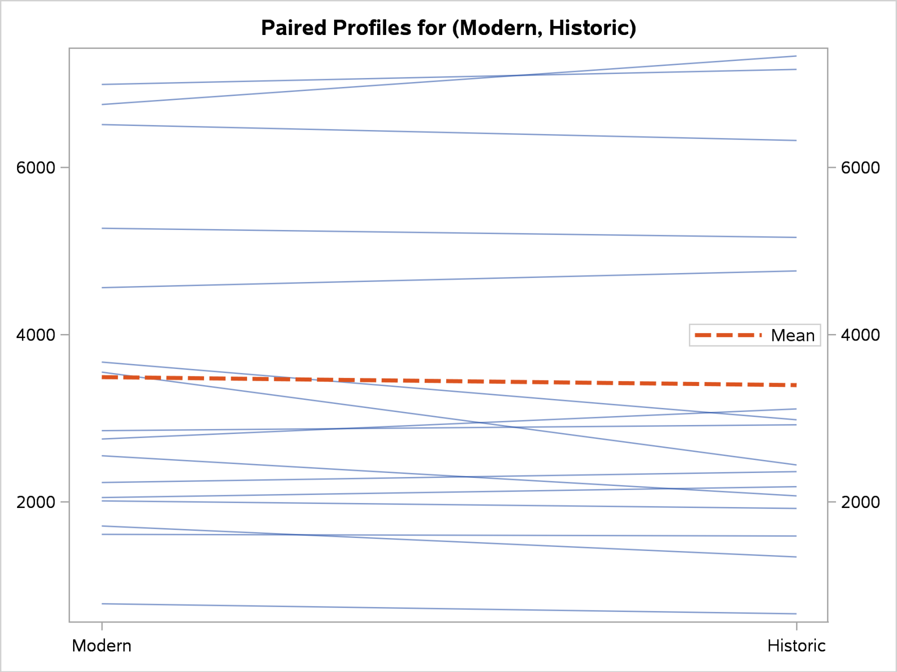
   Two points for this, one for this but with a definite error.

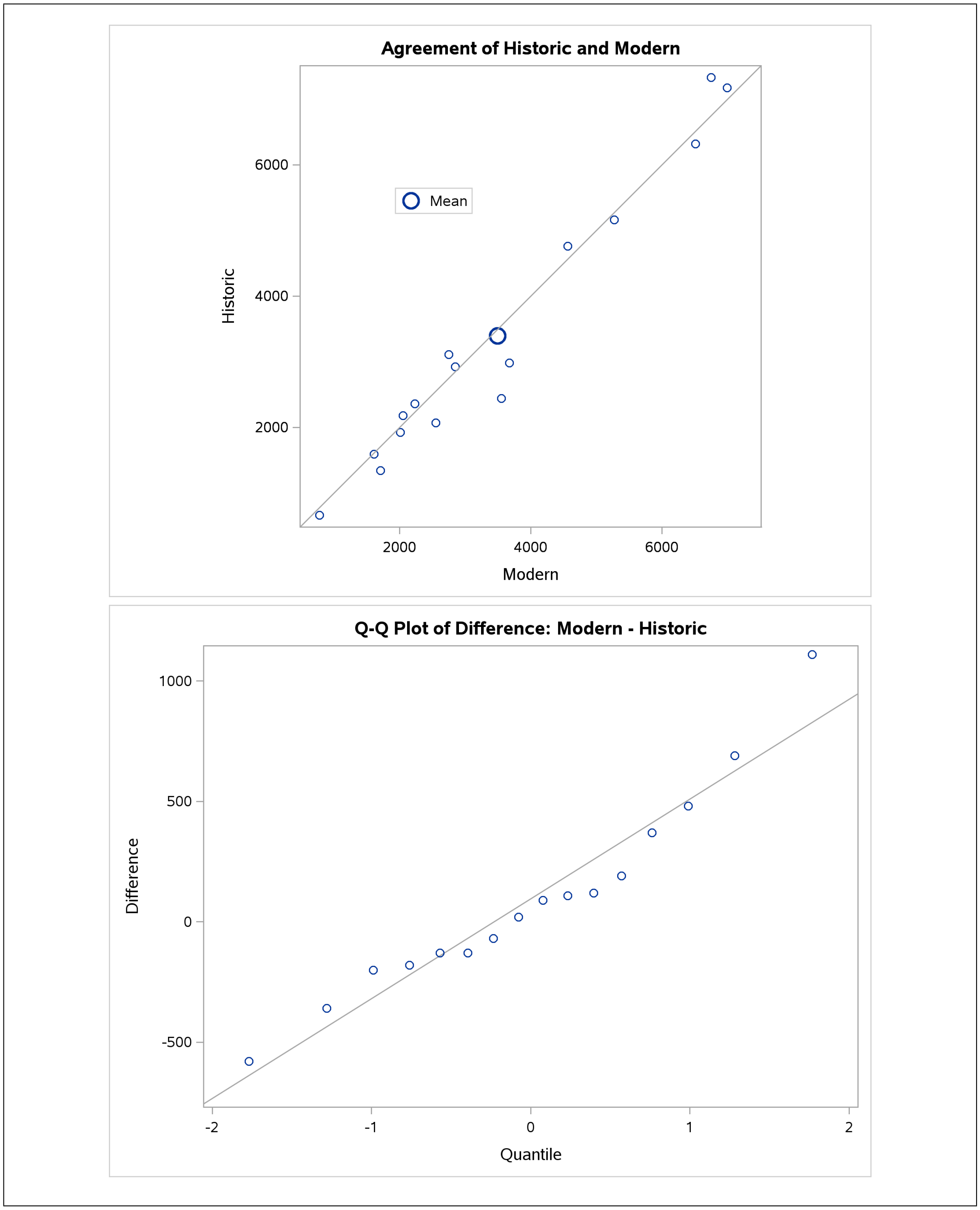
   My output is this:

```
          N        Mean     Std Dev     Std Err     Minimum     Maximum

         16     95.5625       414.5       103.6      -580.0      1110.0
            Mean       95% CL Mean        Std Dev      95% CL Std Dev

          95.5625      -125.3    316.4       414.5       306.2    641.5
                             DF    t Value    Pr > |t|

                             15       0.92      0.3710
```

along with these graphs:







(c) (2 marks) What key assumption is made in order to be able to trust the paired t-test?

Solution: We need the *differences* between the two measurements for each region to be (approximately) normally distributed.

It is irrelevant whether the historic or the modern grain production figures are themselves normally distributed; as long as the difference is, all is good. If both measurements *do* happen to be normally distributed, then their difference will certainly be normal, but this is (as mathematicians put it) sufficient but not necessary.

I suspect (before grading) that the grading here is going to be either 2 or 0. Saying that something else is approximately normal, or saying just "approximately normal" without saying *what* has to be, rather misses the point of the question. I did give (after grading) a few 1s, mostly for people who got the right answer and then also added something that was definitely wrong.

(d) (4 marks) Give SAS code to produce a graph that will enable you to assess the assumption you named in the previous part. If you need to make any design decisions for your graph, it is up to you to decide on something sensible.

Solution: The four marks is a clue that you have two fairly substantial things to do. They are: (i) calculate the differences, and (ii) make a graph of them.

I just checked that I have the data set handy, and it's called `grain`.

For (i), you need to create a new data set with the calculated differences in it, like this:

```
data grain2;
  set grain;
  diff=Modern-Historic;
```

Call the new data set, and the difference variable, whatever you like; also, there is no problem in taking the differences the other way around.

To convince you that it worked, `grain2` is now the most recently created data set, so:

```
proc print;
```

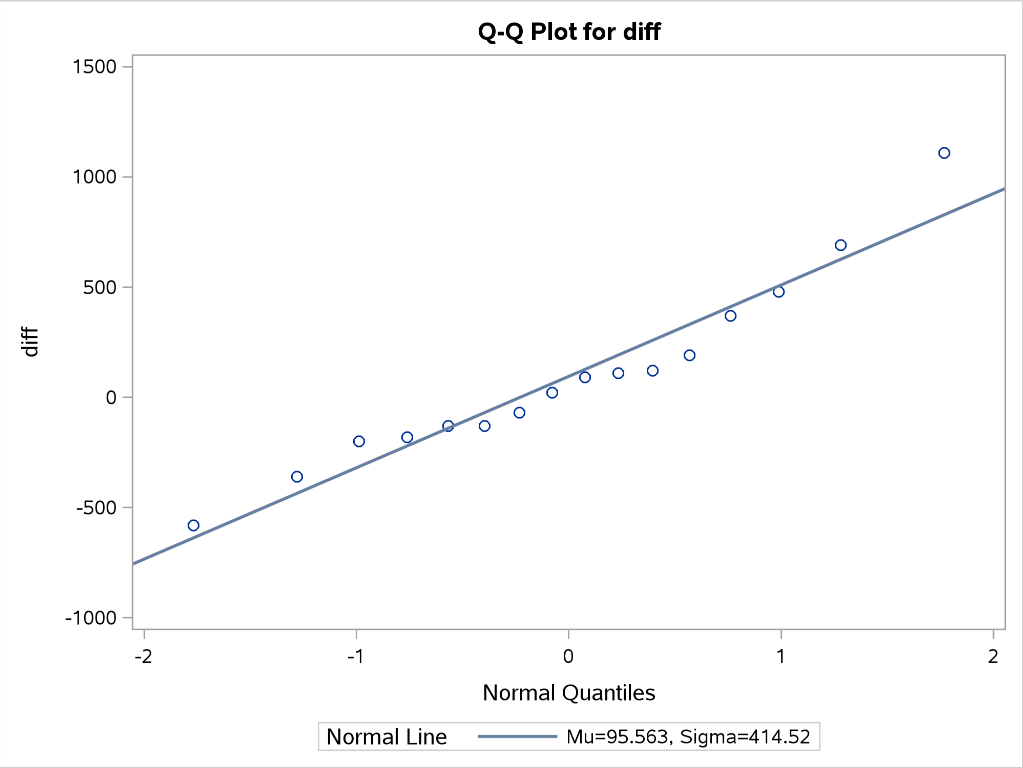
```
          Obs          Region          Modern         Historic     diff

            1               1            1610             1590       20
            2               2            2230             2360     -130
            3               3            5270             5161      109
            4               4            6990             7170     -180
            5               5            2010             1920       90
            6               6            4560             4760     -200
            7               7             780              660      120
            8               8            6510             6320      190
            9               9            2850             2920      -70
           10              10            3550             2440     1110
           11              11            1710             1340      370
           12              12            2050             2180     -130
           13              13            2750             3110     -360
           14              14            2550             2070      480
           15              15            6750             7330     -580
           16              16            3670             2980      690
```

For (ii), you need to make a suitable graph of the differences. I think the best graph is a normal quantile plot, something like this:

```
proc univariate noprint;
  qqplot diff / normal(mu=est sigma=est);
```

You need a line on the plot to assess normality with, so you need the bit after the slash on the `qqplot` line as well.

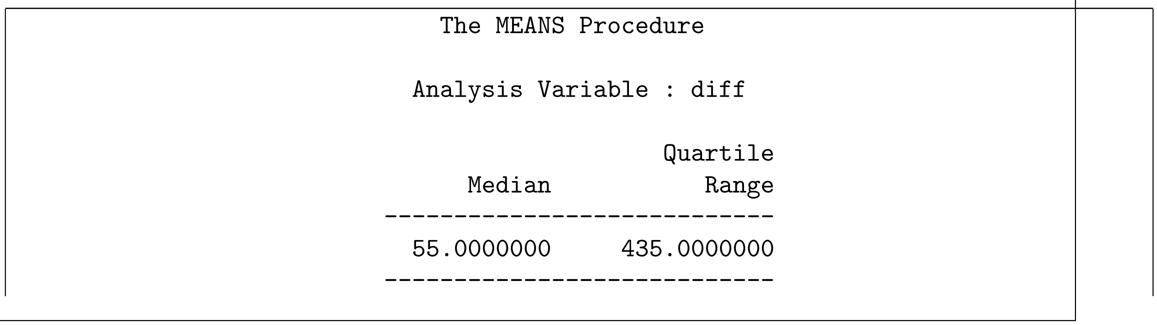
No comma inside the brackets, which often gets me as well. The `noprint` is optional; in practice it would save you getting a lot of printed output that you don't really want and have to page through, but as long as the normal quantile plot is there somewhere, I'm good.



For this, I'd say it's perfectly all right to estimate `mu` and `sigma` using the sample mean and SD (I was mainly interested in whether you could get as far as producing the right kind of graph), but if you would prefer to estimate them using the IQR and the median, feel free to say that. You don't need to go into details (and besides, you don't have any numbers to work with). This is what I meant by "design decision". I didn't want a lot of you asking about this during the exam; whichever way you go here is fine. Make up some numbers and put them in, or say what goes in in place of the `est`, whatever you like.

Extra: for my curiosity, using IQR and median, and a little surreptitious R:

```
proc means median qrange;
  var diff;
```

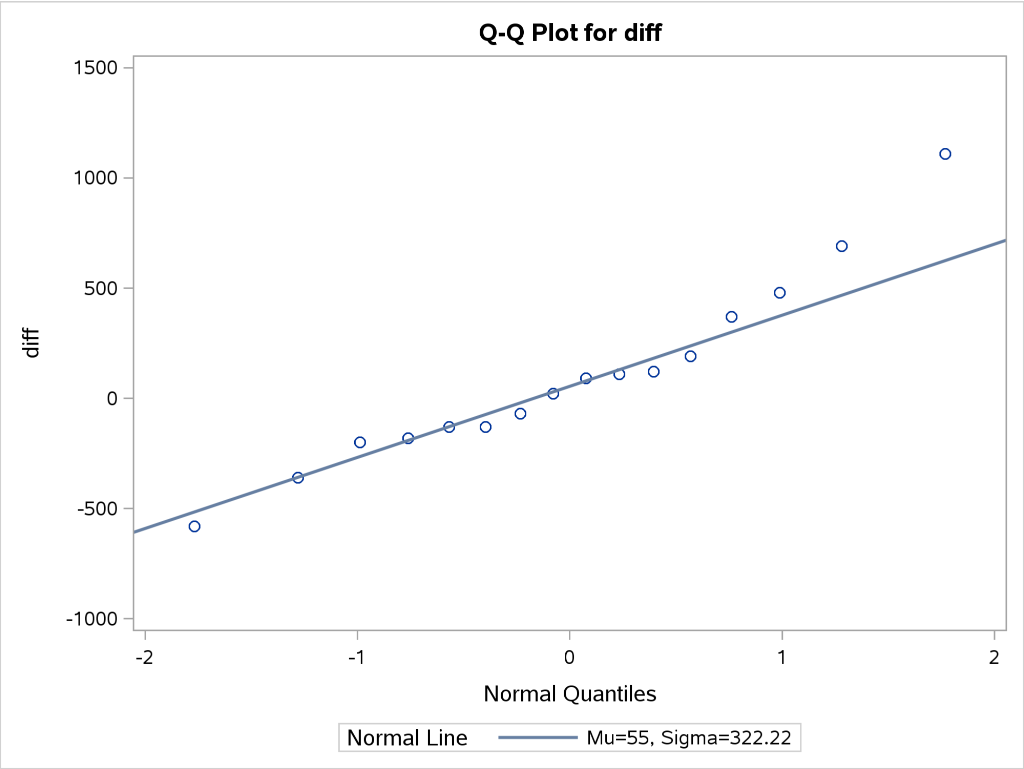


```
                          The MEANS Procedure

                       Analysis Variable : diff

                                      Quartile
                        Median           Range
                  ----------------------------
                    55.0000000     435.0000000
                  ----------------------------
```

```
435/1.35

## [1] 322.2222
```

and then

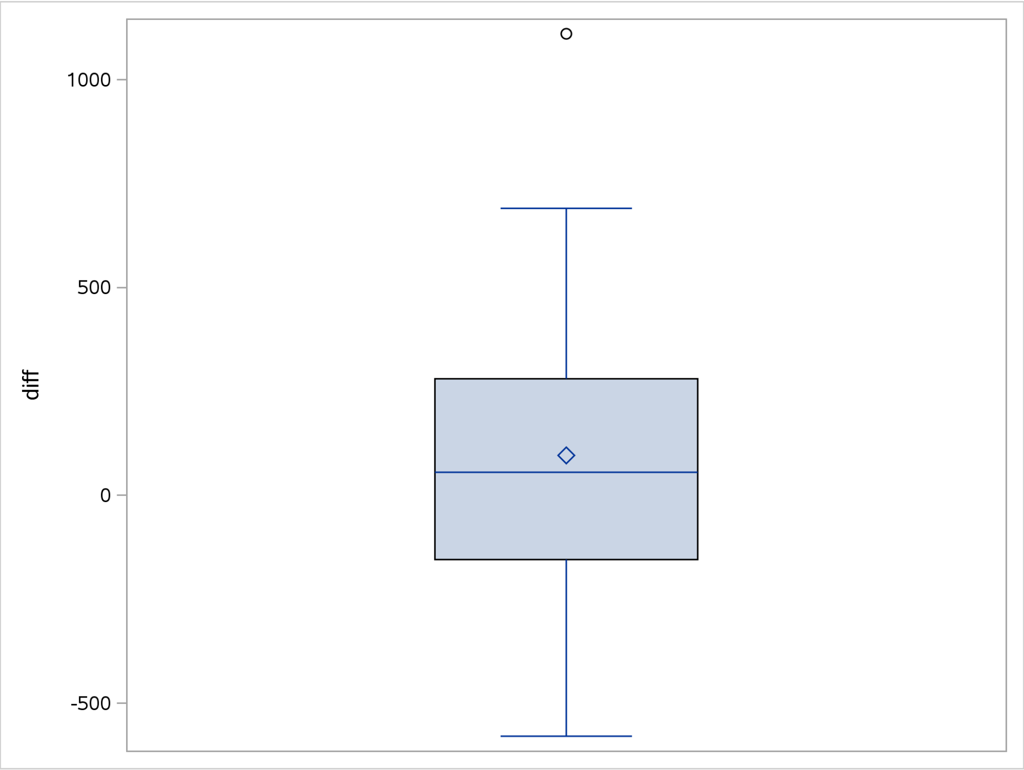
```
      proc univariate noprint;
        qqplot diff / normal(mu=55 sigma=322.22);
```



This looks a bit different from the first one: here, the line passes *above* the first point, and the last four points are all above the line, with the last one especially looking off the line. If you compare the values of `mu` and `sigma` below the graphs, you'll see that there is a substantial difference between them, with the second graph having a much smaller value of `sigma` and thus a flatter line. My take is that the largest difference might be an outlier, and if so it would be inflating the SD. (It doesn't look at all an outlier on the first graph, but it might be one on the second.)

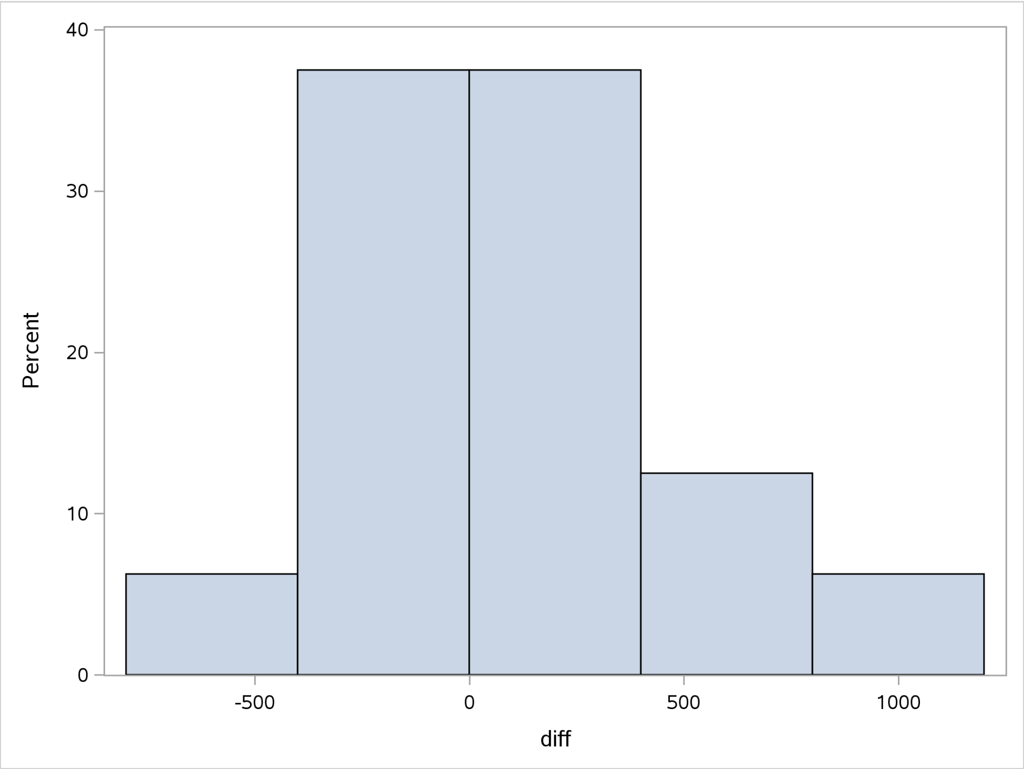
Other acceptable graphs would be a boxplot or a histogram, which would show up asymmetric shape or outliers well enough to tell us about approximate normality. In each case, the differences need to be used in the plot. Here's a boxplot:

```
proc sgplot;
  vbox diff;
```



and here's a histogram:

```
proc sgplot;
  histogram diff;
```

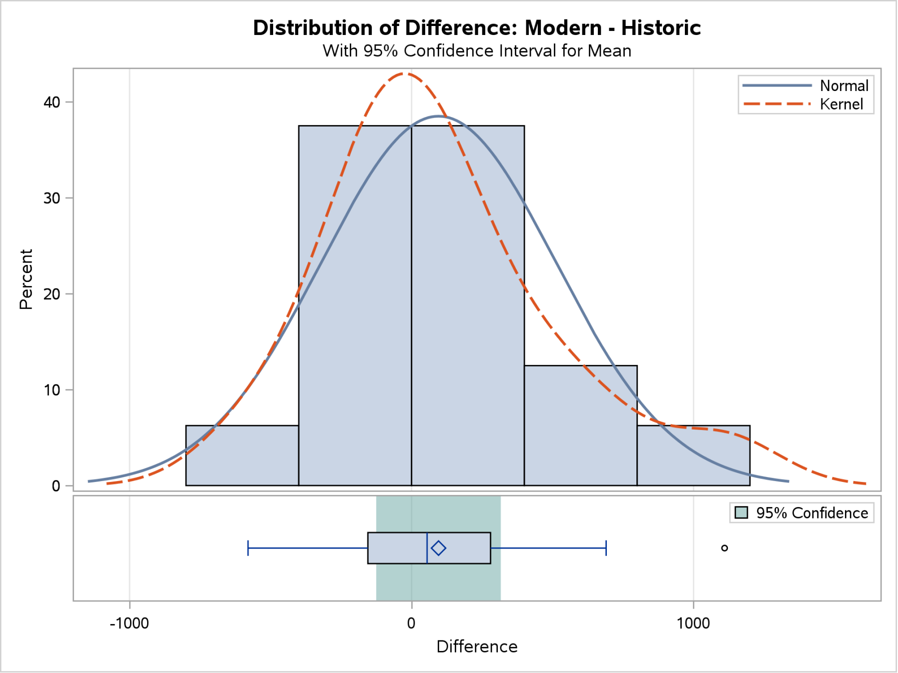


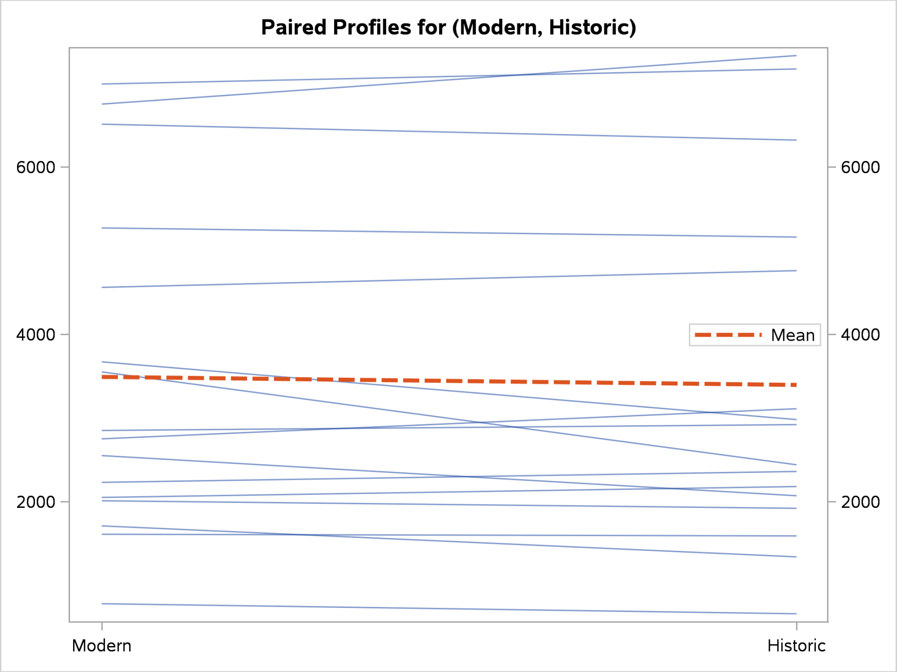
If you go this way, the second two points are pretty easy to get.

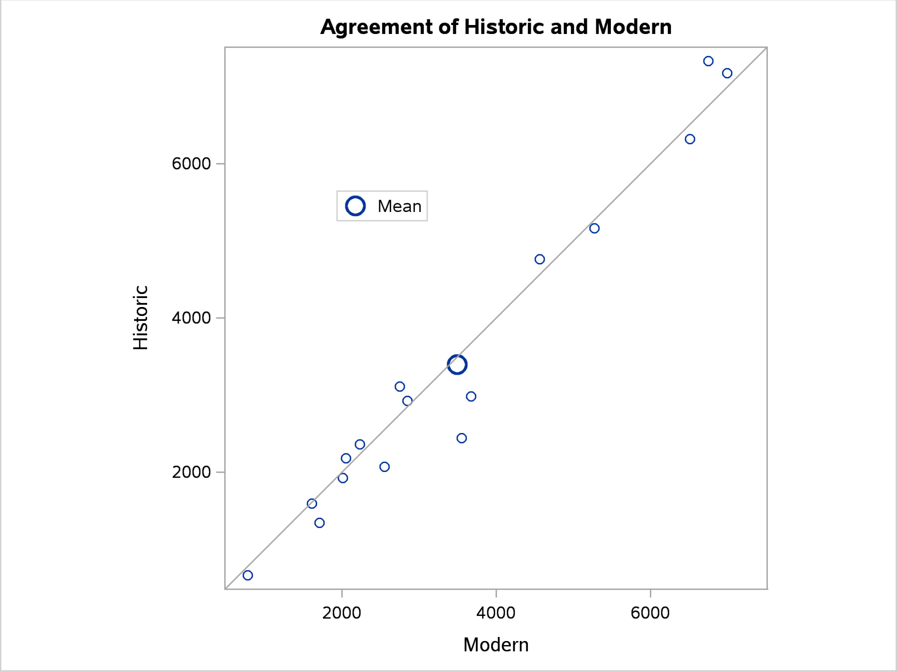
If you (incorrectly) thought something else was the key assumption in the previous part, then you can get marks here by providing code to produce a graph to assess whatever you thought the assumption was. If in so doing you make the coding easier than it would (correctly) be (which is likely), then you won't get full marks here, but you should get something. For example, if you thought both sets of measurements should be normal, then you should make a suitable plot to assess normality (boxplot or normal quantile plot, probably) for *each* of `Modern` and `Historic`. This is easier coding, since you don't have to calculate the differences, but if you do it right for *both* groups, you should get two marks (essentially, the second two marks below). Some people shot for something like side by side boxplots, having concluded that both groups needed to be normal; the data reorganization required for that would cover the first two points.

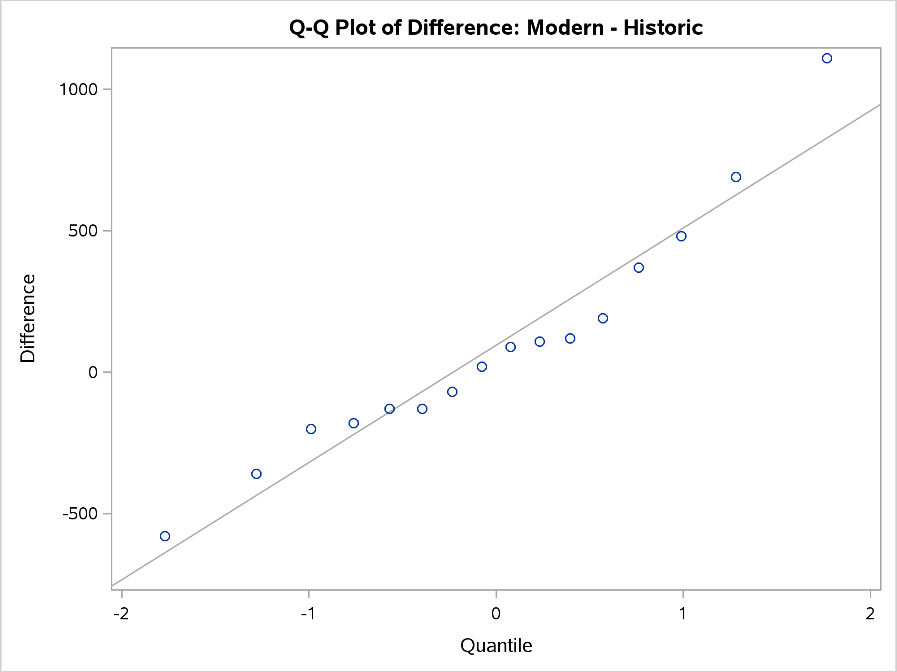
Grading: these are two independent two-point subquestions. For the calculation of the diffences, 2 points if you got it right or made an inconsequential error, 1 point if there was a definite error but you had the right idea, 0 otherwise. For the plot, 2 points if you made a sensible plot of the differences, possibly with an inconsequential error, 1 point for something that wouldn't work but used the differences and made some sense, 0 otherwise. Note that you can get the second two marks even if you could not calculate the differences; the way you handle this is to say something like "suppose the differences were calculated in a column called `diff`", and then go ahead and give code for your plot. For some people, a column called `diff` mysteriously appeared; of course, I couldn't give you any points for calculating it, but you could certainly get two for plotting it.

The cleverest answer I saw noted that the paired *t*-test output has some graphs attached to it, repeated here:




Paired Profiles for (Modern, Historic)
6000
4000
2000
6000
4000
2000
Mean
Modern
Historic


Agreement of Historic and Modern
Mean
Historic
6000
4000
2000
2000
4000
6000
Modern

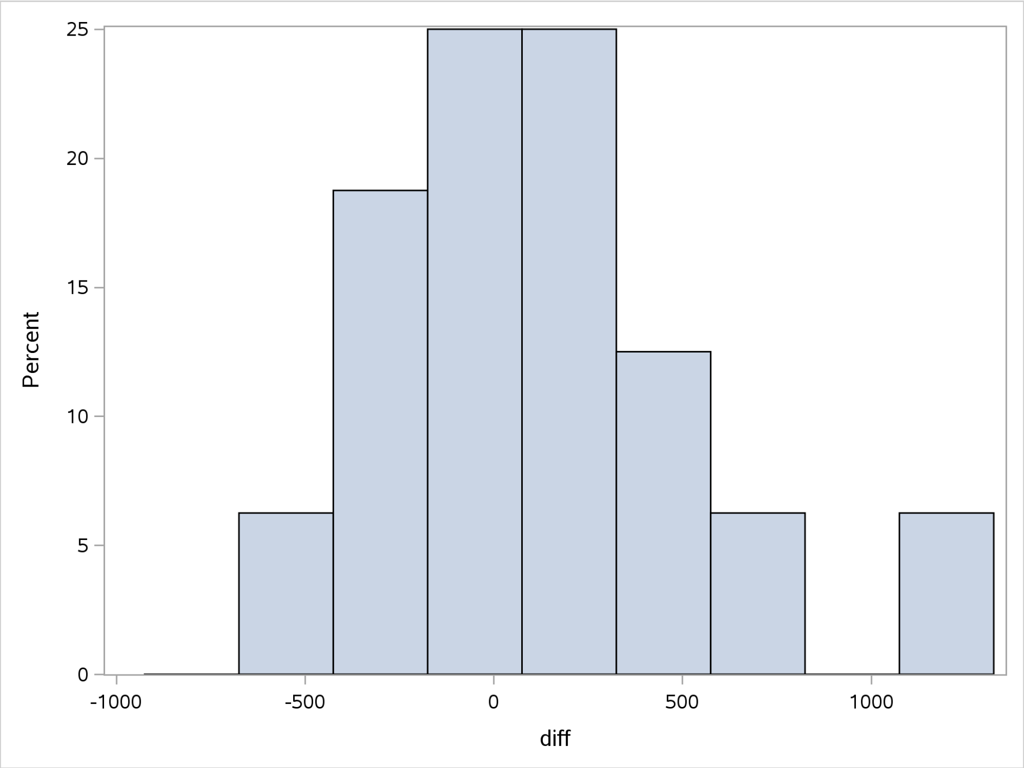

Q-Q Plot of Difference: Modern - Historic
Difference
1000
500
0
-500
Quantile
-2
-1
0
1
2

and that the last one of *those* is a normal quantile plot of the differences, so that all you have to do is to look at it. I couldn't very well give this answer less than four points.

Somebody else wanted a spaghetti plot. The SAS way to get it is to use this output (it's easier to build yourself in R). However, I don't see how such a plot helps you assess the appropriate normality.

Extra: The boxplot has an outlier, but the choice of bins mean that this does not show up on the histogram. I could possibly fake up the histogram to make it appear:

```
proc sgplot;
  histogram diff / binstart=-800 binwidth=250;
```



By "fake up" I mean that I know where the outlier was on the boxplot, and I deliberately chose the bins to make it show up here too. Not exactly honest data science.

6. A personnel officer developed four different aptitude tests and administered them to 25 new entry-level clerical employees. Each of these employees went through a "probationary period" in which they could get accustomed to the work involved. At the end of each employee's probationary period, they were assessed for proficiency on the job. The personnel officer would like to know whether any of the aptitude tests were able to predict job proficiency, and, if so, which aptitude tests. The data are shown in Figure 10. This display is of a SAS data set called `jobs`, which is the most recently created one.

   All the test and proficiency scores are numbers; they have no units.

   (a) (2 marks) Give the SAS code to run the regression whose output is shown in Figures 11 through 13.

   Solution: This is not meant to be difficult, though you might have to do some detective work to figure out the response and explanatory variables. Look in the regression output to see that the response variable is called `proficiency` and the four explanatory test scores are called `test1` through `test4` and you have:

```
      proc reg;
        model proficiency=test1 test2 test3 test4;
```

   Two points for that (or, if you must, the same `model` with `proc glm`). One for a `proc reg` with one or two errors in the `model` line, zero otherwise.

   Recall that SAS produces a lot of output (by default), so those two lines of code produce all three Figures' worth of output. There is no extra code needed to produce those graphs, and suggesting that extra code is needed is an error. (The hint is that this part is only two marks, so two lines of code is about right.)

   (b) (3 marks) In Figure 11, what is the parameter estimate for the aptitude test labelled `test2`? What does the value mean, in the context of the data?

   Solution: -0.19 (one mark that I deliberately made easy).

   As to what it means: if score on `test2` increases by 1, job proficiency is predicted to go *down* by 0.19 (one mark; not "goes up by -0.19" since that is hard to understand), all else equal (or, without changing the other explanatory variables, or keeping the other explanatory variables the same), the final mark.

   Don't make this harder than it is: the test scores are numbers, and there is no categorical variable called `test` with levels `test1` through `test4` or anything like this.

   I decided to give two marks if you had the parameter estimate value and something indicating a negative relationship between scores on `test2` and job proficiency. Thus two marks covers a wide range, but for three marks you had to get everything.

   (c) (4 marks) Look at Figures 12 and 13. We are looking for any problems with the regression. Assess: (i) plot of residuals against the fitted values, (ii) normal quantile plot of residuals, (iii) plots of residuals against explanatory variables. For each of these, indicate which plot(s) on the Figures you are looking at, and what you conclude. In addition, make an overall comment about the appropriateness of the regression.

   Solution:

   I see these:

- residuals against fitted values: the top left plot in Figure 12. I see a random pattern (good), or maybe you see a little fanning-in (getting narrower from left to right). Alternative: the second plot of the first row is *standardized* residuals against fitted values, showing the same pattern. I accept either of these.
- normal quantile plot: the first plot in the second row of Figure 12. This is as normal as you could wish for, the points being close to the line all the way up.
- residuals against explanatory: all the plots in Figure 13. These to my mind are all acceptably random. (There appears to be a low outlier among the `test4` scores that shows up on the left side of the plot, but there is no pattern *among the residuals*, so this is a (small) problem with the data rather than with the regression.)

So my overall conclusion is that I am happy with the regression.

Guideline: a point for each of these, that is, a point for identifying the graph that assesses what you want to assess and for making a sensible comment on it, for each of the three things. For the fourth point, if you found a problem anywhere, you should say that the regression is not appropriate; if you found no problems (as I didn't), you should declare that the regression is appropriate.

This is, in retrospect, a lot of comment for four points, but I didn't think I could reasonably make it out of about seven points (to allow two points for each of the graphs). With that in mind, a brief comment about each graph is enough; I don't need a detailed explanation. If it looks reasonably sensible, I am happy with it.

(In a previous version of this exam, I left this more wide-open, but I figured it would be impossible to grade, so I told you what to look at to make things easier for you and most likely for me as well.)

(d) (2 marks) What *one* thing would you do next to improve this regression? Explain briefly.

Solution: This depends on whether you found a problem previously.

If you found a problem, your task is to find something that would fix it. For example, if you thought the top left graph in Figure 12 showed fanning-in, you might suggest trying a transformation of the response variable (since the problem is in this case in the residual-vs-fitted-value plot). Suggest one, or suggest doing Box-Cox. If you found a problem elsewhere, for example in one of the plots of residuals vs. an x, you should suggest a re-expression of *that* x, since that is where the problem is.

If you didn't find any problems, then you should suggest a model-building move. The most obvious one is to remove `test3` from the regression, since it has the highest P-value out of the four aptitude tests in Figure 11. I would also accept removing `test4` on the grounds that it is not significant, though removing it and not `test3` is kind of odd. If you remove *both* the non-significant `test3` and `test4`, you *also* need to do a test to see whether removing both of those was a mistake (since removing one might have made the other significant). I think I talked about that in an assignment solution, but it's fine to talk about what you would do even if you don't have code to do it. (In the real world, you would probably consult Stack Overflow at this point!)

If you found a problem but you suggest removing an x, that's one point, because fixing your problem might change which test scores are important (particularly so if your suggestion is to make a transformation, which is a *nonlinear* change to the model). So you would do the transformation first, fit the model with the transformed variables, assess *the new model* for problems, and if you find none, *then* think about which x-variables to remove. (There is a certain amount of interaction between model building and model checking, but the likelihood is that getting the right kind of relationship between response and explanatory variables will have the biggest effect. If an explanatory variable really has nothing to say, then it will continue to have nothing to say after a transforamtion.)

If you mention more than one thing, you automatically lose a point, even if both of your ideas are sensible, because the aim is for you to name what you consider to be the *most important* thing to do next.

I gave you a point if you suggested the "wrong" thing given your previous conclusion (or whatever I could discern your previous conclusion to be): a transformation if you were happy with the regression, removing an x if you found problems, on the basis that these are not terrible things to try anyway.

I marked this part before the previous one (because I wanted to end the evening on something not too big), and so I had to scan the previous part anyway to determine whether you thought there were any problems (making me wonder whether I should have done that part first anyway).

Extra: I would be extra-cautious about removing more than one x here, because these are supposed to be four aptitude tests, and there might be a good deal of overlap in what they measure. That would play out in the test scores being possibly highly correlated amongst themselves, and taking out one might have a big impact on the P-values of the others:

```
proc reg;
  model proficiency=test1 test2 test4;
```

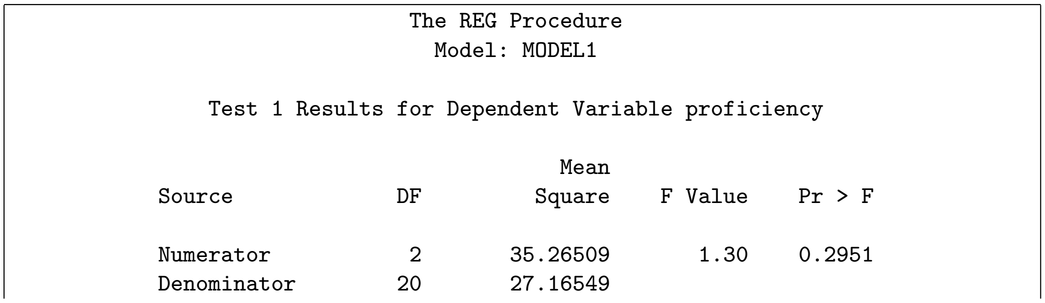
```
                              The REG Procedure
                                Model: MODEL1
                       Dependent Variable: proficiency

                   Number of Observations Read          25
                   Number of Observations Used          25
                             Analysis of Variance

                                  Sum of           Mean
Source                   DF      Squares         Square    F Value    Pr > F

Model                     3   2181.34117      727.11372      27.56    <.0001
Error                    21    554.09883       26.38566
Corrected Total          24   2735.44000
             Root MSE              5.13670    R-Square     0.7974
             Dependent Mean       94.68000    Adj R-Sq     0.7685
             Coeff Var             5.42532
                             Parameter Estimates

                         Parameter       Standard
    Variable       DF     Estimate          Error    t Value    Pr > |t|

    Intercept       1     98.49852       29.36114       3.35      0.0030
    test1           1      0.82339        0.20855       3.95      0.0007
    test2           1     -0.18616        0.08966      -2.08      0.0503
    test4           1     -0.60013        0.39884      -1.50      0.1473
```

As it happens, the other P-values didn't change much, and test2 just crossed the line and became non-significant. So now we are definitely justified in removing test4, and after that we can make a call about whether we should keep test2.

Extra 2: if you wanted to think about removing both test3 and test4, you add a test line to your regression (the one containing everything):

```
proc reg;
  model proficiency=test1 test2 test3 test4;
  test test3=0, test4=0;
```

You get all the output from the regression from before plus this:



```
                          The REG Procedure
                           Model: MODEL1

              Test 1 Results for Dependent Variable proficiency

                                      Mean
          Source             DF         Square    F Value    Pr > F

          Numerator           2       35.26509       1.30    0.2951
          Denominator        20       27.16549
```

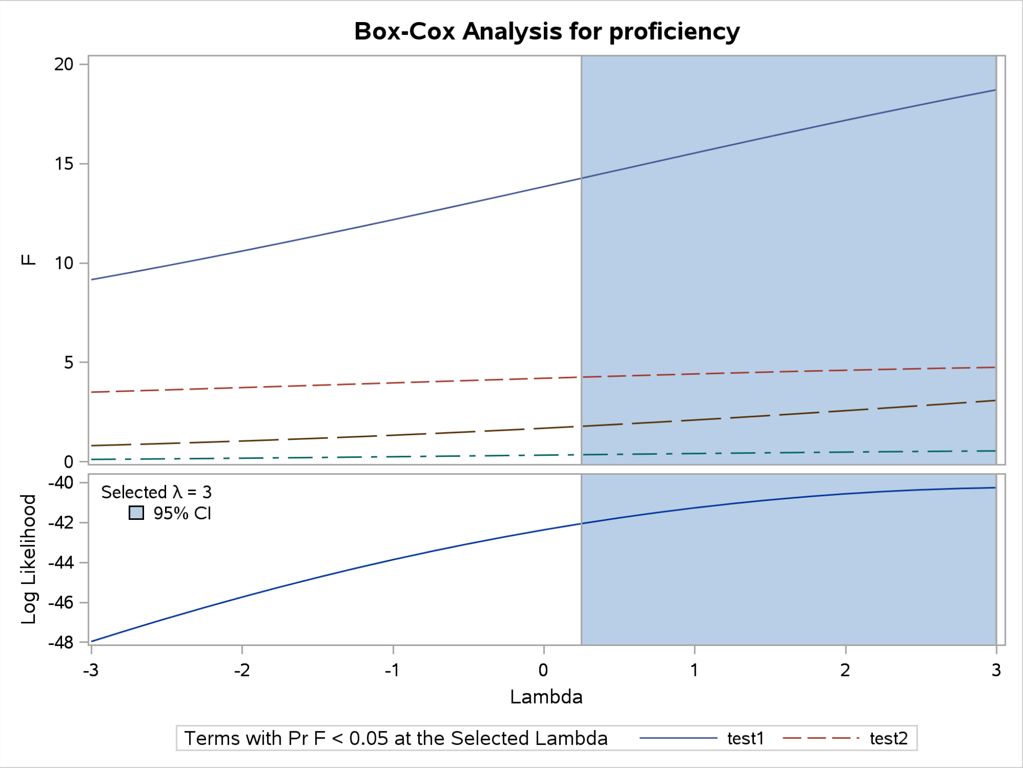
Taking both explanatory variables out is entirely justifiable. More precisely, the big model (with everything) and the small model (without `test3` and `test4`) show no significant difference in fit, so we go with the small model because it is simpler. You might say that the bigger model has to "prove its worth" by producing a small P-value in this test.

This is the same thing as `anova` for comparing two models in R. A "multiple-partial F-test", in the jargon.

Extra 3: Box-Cox would go like this:

```
proc transreg;
  model boxcox(proficiency)=identity(test1 test2 test3 test4);
```

with graphic output:



As it turns out, there is no justification for a transformation at all. The confidence interval for λ goes from 0.25 to up beyond 3 somewhere. (SAS stops looking at 3.) Fanning-in, in general, if that's what you saw, is a weird kind of thing to deal with at the best of times. Box-Cox doesn't always help with it. Fanning *out* often responds well to taking logs or square roots that will bring the higher values down a bit, but fanning-in not so much.

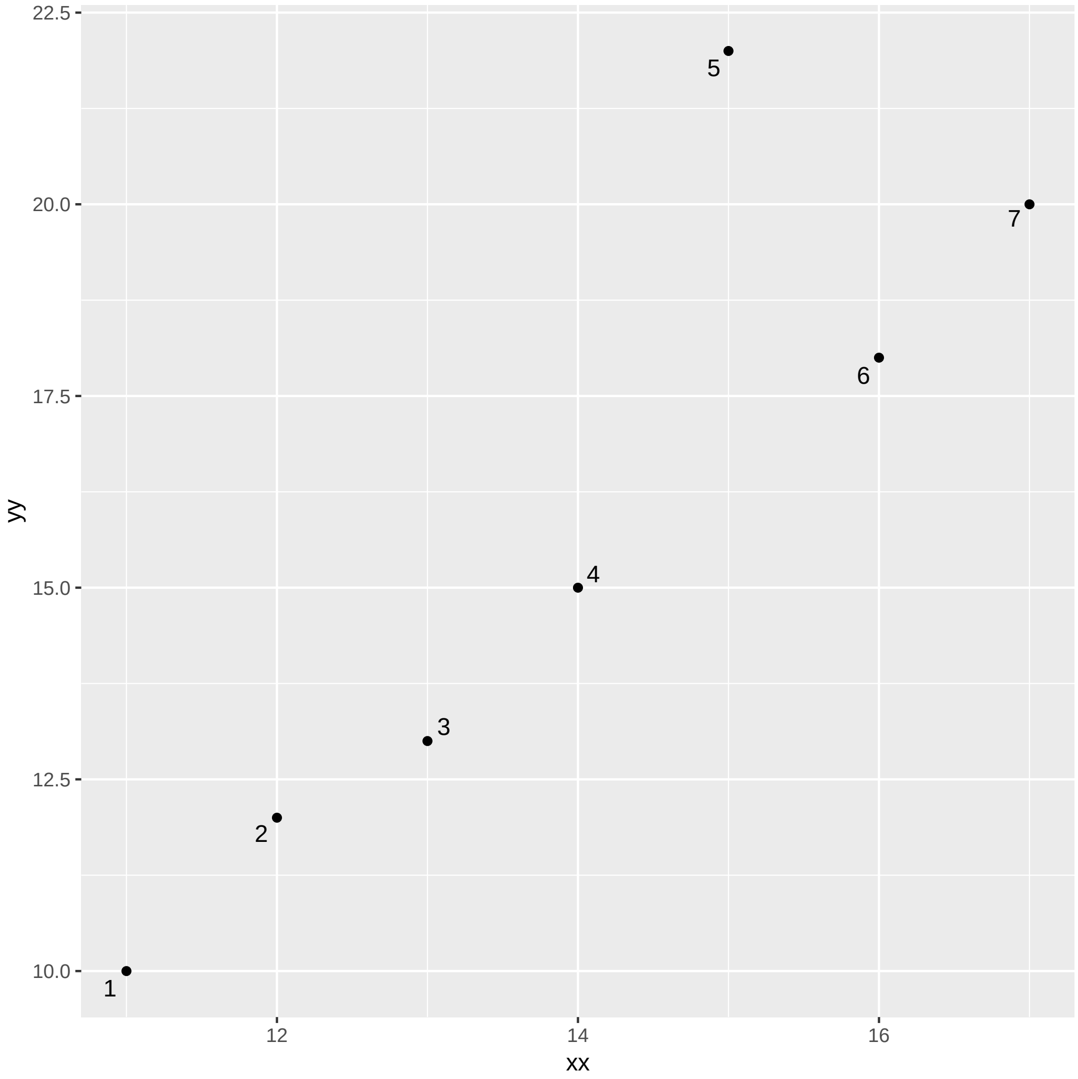
Even though Box-Cox turns out not to help much here, it's a sensible thing to suggest, so you get credit for suggesting it as a way to hunt for a transformation (if that's what you thought you needed).

7. Figure 14 shows an R data frame called `dd`, with a column called `row` of row numbers, and two data columns called `xx` and `yy`. A scatterplot of `yy` against `xx` is shown in Figure 15, with the data points labelled by which row of the data frame they come from.

   (a) (3 marks) Give the R code that was used to produce Figure 15.

Solution: This is a scatterplot with the points labelled by row number. You might guess that `geom_text_repel` was used to label the points, something that is supported by the labels being sometimes to the left and sometimes to the right of the points they label. Thus:

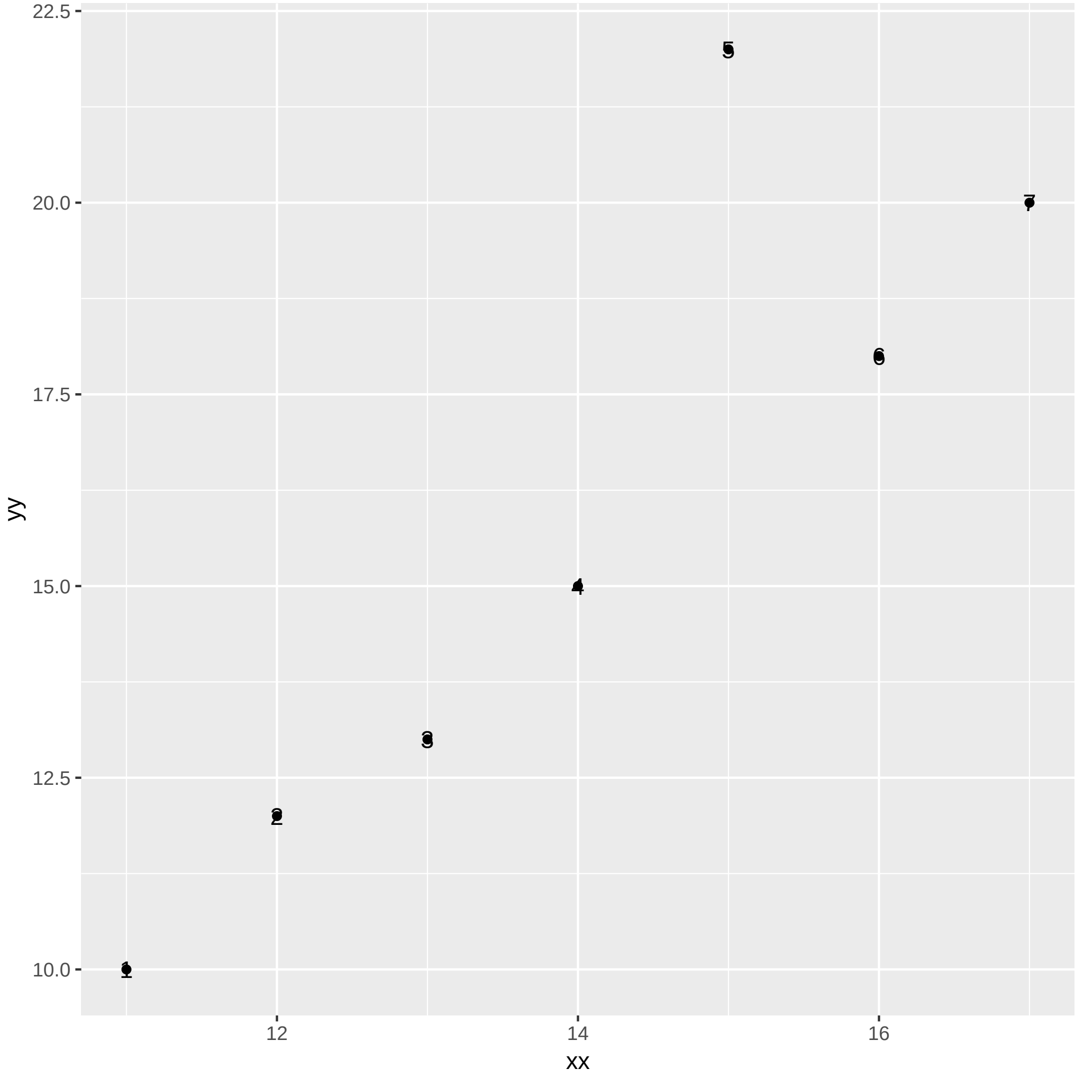
```
ggplot(dd, aes(x=xx, y=yy, label=row))+geom_point()+geom_text_repel()
```



The usual marking scheme: minus one per error, a minimum of one if you get something correct, here making a scatterplot without labelling the points. The guideline beyond that is one more for the `label` piece in the aesthetic, and one more for the `geom_text_repel`. I suppose that means that if you get both the components of labelling the points but you forget to plot them, you get 2, which I'm OK with here. It was really the labelling of the points that I wanted to see you do here. The scatterplot is really taken for granted at this stage. (That was the kind of thing that I would have asked about on the midterm.)

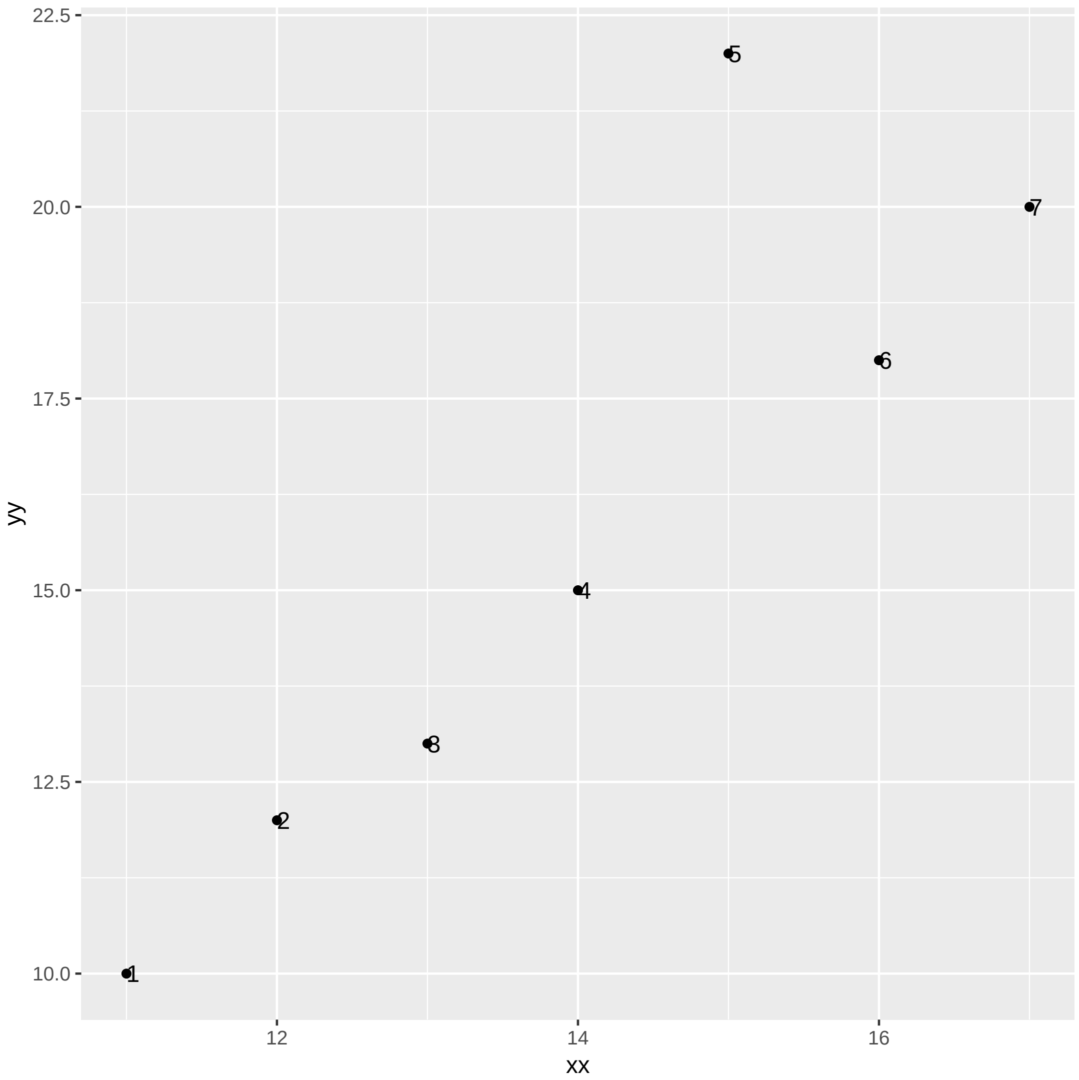
If you have a way of labelling the points properly with `geom_text`, that will be fine, but you'll have some more work to do. For example, this puts the labels *on top of* the points instead of beside them:

```
ggplot(dd, aes(x=xx, y=yy, label=row))+geom_point()+geom_text()
```



This is two marks. Getting full marks this way requires you to do something to get the labels away from the points, eg. by moving them a little to the right:

```
ggplot(dd, aes(x=xx, y=yy, label=row))+geom_point()+geom_text(hjust=0)
```



I stole this idea from my lecture notes (the pages I didn't show you in class). I would for myself move the labels a bit further away from the points, but since you don't get the chance to try it out, I would accept `geom_text` with any value for `hjust` (or even `vjust`), to move the label away from the point a little.

Or, of course, you can use `geom_text_repel`, which is designed for this job.

(b) (4 marks) Figure 16 shows a function `rsq` that takes as input any data frame `d` that has columns `xx` and `yy`, runs a regression predicting `yy` from `xx`, and outputs the R-squared from that regression. Figure 17 shows a function `omit1` that takes as input two things: (i) any data frame `d` that has columns `xx` and `yy`, and (ii) a row number `i`. The function *omits row i*, and then runs the regression of (the remaining) `yy` on (the remaining) `xx`, outputting the R-squared from that regression.

I will not be asking how these functions work. You may take it for granted that they do what I say they do, even if you don't see how they do it.

Your task is to add a new column called `rsquared` to data frame `dd`. In that column should be the value of R-squared for the regression of `yy` on `xx` using all the observations *except* the one in that row. For example, the first entry in the column `rsquared` should be the value of R-squared for the regression of `yy` on `xx` using all the `xx` and `yy` values except for the first `(xx, yy)` pair.

Give R code that will create this new column `rsquared`, using `tidyverse` ideas, and using the function `omit1` defined in Figure 17. You do not have to save your new data frame.

Solution:

This turns out to be alarmingly simple, but I'm awarding a lot of marks to give you some credit if you go wrong:

```
dd %>% mutate(rsquared=map_dbl(row,~omit1(dd,.)))

## # A tibble: 7 x 4
##     row    xx    yy rsquared
##   <dbl> <dbl> <dbl>    <dbl>
## 1     1    11    10    0.692
## 2     2    12    12    0.758
## 3     3    13    13    0.783
## 4     4    14    15    0.795
## 5     5    15    22    0.996
## 6     6    16    18    0.803
## 7     7    17    20    0.771
```

The usual marking scale. The baseline 1 point is if you get this far:

```
dd %>% mutate(rsquared=xxxx)
```

with anything at all in place of the `xxxx`. Four points if you get that plus:

- a `map_dbl` because my function `omit1` returns a decimal number
- getting the "for each" from `row`
- getting the squiggle and the `omit1`
- getting the right two inputs to `omit1`, `dd` and the for-each dot, in the right order.

This doesn't work (though you might be tempted to try it):

```
dd %>% mutate(rsquared=omit1(row))

## Error in mutate_impl(.data, dots): Evaluation error: no applicable method for
'slice_' applied to an object of class "c('double', 'numeric')".
```

This falls under the one-mark umbrella.

The problem is that the function `omit1` is not "vectorized", in that it doesn't work if the second input is a vector rather than a number. Often things like this work because vectorization is built in:

```
tibble(x=1:8) %>% mutate(y=sqrt(x))

## # A tibble: 8 x 2
##       x     y
##   <int> <dbl>
## 1     1  1
## 2     2  1.41
## 3     3  1.73
## 4     4  2
## 5     5  2.24
## 6     6  2.45
## 7     7  2.65
## 8     8  2.83
```

This works because `sqrt` works for vectors, but for `omit1` the second input has to be a number, otherwise `slice` won't work properly. So the only way is to use `map_dbl`, or, I suppose, a `for` loop, but the "using `tidyverse` ideas" in the question ruled that out as an answer.

If you manage to come up with all of this, displaying a data frame with a column called `rsquared` that would contain the right values:

```
rsquared=numeric(0)
for (i in 1:7) {
    rsquared[i]=omit1(dd,dd$row[i])
}
with(dd,tibble(row,xx,yy,rsquared))

## # A tibble: 7 x 4
##     row    xx    yy rsquared
##   <dbl> <dbl> <dbl>    <dbl>
## 1     1    11    10    0.692
## 2     2    12    12    0.758
## 3     3    13    13    0.783
## 4     4    14    15    0.795
## 5     5    15    22    0.996
## 6     6    16    18    0.803
## 7     7    17    20    0.771
```

I guess you deserve two points for it, minus one per mistake that you make. (It took me about four goes to get this right to show you; the `map` actually came out right first time, which probably tells you something about the relative ease of coding `map` and `for`, once you understand them.)

Another way of doing this that does not use my function `omit1` is to use the `subset` input to `lm`. This is about the easiest way I can see to do it like that:

```
dd %>% mutate(reg=map(row, ~lm(yy~xx,data=dd, subset=-.))) %>%
  mutate(regsum=map(reg,~summary(.))) %>%
  mutate(rsquared=map_dbl(regsum, "r.squared"))
```

```
## # A tibble: 7 x 6
##     row    xx    yy reg      regsum           rsquared
##   <dbl> <dbl> <dbl> <list>   <list>              <dbl>
## 1     1    11    10 <S3: lm> <S3: summary.lm>    0.692
## 2     2    12    12 <S3: lm> <S3: summary.lm>    0.758
## 3     3    13    13 <S3: lm> <S3: summary.lm>    0.783
## 4     4    14    15 <S3: lm> <S3: summary.lm>    0.795
## 5     5    15    22 <S3: lm> <S3: summary.lm>    0.996
## 6     6    16    18 <S3: lm> <S3: summary.lm>    0.803
## 7     7    17    20 <S3: lm> <S3: summary.lm>    0.771
```

As you see, the right answer, so full marks. The columns that I called `reg` and `regsum` here are list-columns: they hold all the information in respectively the fitted model and the summary output of the fitted model. The steps are:

- fit the regression model without the data in the row you're looking at (the dot means "it" and minus-dot means (here) using all the rows except the one you're looking at.
- for each of those regression models, run `summary` on it.
- for each of the summaries, extract the thing called `r.squared`, which is a decimal number (hence the `dbl`).

Another way you might think of is to use `glance` from `broom`:

```
dd %>% mutate(reg=map(row, ~lm(yy~xx,data=dd, subset=-.))) %>%
  mutate(regsum=map(reg,~glance(.))) %>% unnest(regsum)
```

```
## # A tibble: 7 x 15
##     row    xx    yy reg   r.squared adj.r.squared sigma statistic p.value
##   <dbl> <dbl> <dbl> <lis>     <dbl>         <dbl> <dbl>     <dbl>   <dbl>
## 1     1    11    10 <S3:~     0.692         0.615 2.47       9.00 4.00e-2
## 2     2    12    12 <S3:~     0.758         0.697 2.48      12.5  2.40e-2
## 3     3    13    13 <S3:~     0.783         0.729 2.43      14.5  1.90e-2
## 4     4    14    15 <S3:~     0.795         0.744 2.45      15.5  1.70e-2
## 5     5    15    22 <S3:~     0.996         0.995 0.261   1040.   5.51e-6
## 6     6    16    18 <S3:~     0.803         0.753 2.34      16.3  1.57e-2
## 7     7    17    20 <S3:~     0.771         0.714 2.34      13.5  2.13e-2
## # ... with 6 more variables: df <int>, logLik <dbl>, AIC <dbl>, BIC <dbl>,
## #   deviance <dbl>, df.residual <int>
```

which gets the R-squareds along with other things. Also full marks.

But you see that using my `omit1` is a lot easier, which is why I gave it to you. The aim of the question was to test whether you understood `map`, so I wanted to give you one where it came out fairly simply if you understood what you were doing.

This part was (deliberately) challenging. It was four marks not because you needed to produce a lot of code, but because there was a lot of clear thinking required. I also figured that a lot of people would go wrong somewhere, so I wanted to be able to give a lot of partial credit for sensible ideas.

(c) (2 marks) The output from running your code of the previous part is shown in Figure 18. Describe how the outlier influences the results and explain briefly why it has the effect it does.

Solution: Note that you can answer this part *even if you have no idea what is going on in the rest of the question.*

The key observation (one point) is that the R-squared is *much* higher for the regression where row 5 is omitted than for all the other regressions (which are fairly similar).

As to why that is: row 5 is an outlier, and if you take out the outlier, the other six points are on an almost perfect straight line, so the R-squared is going to be very high; with the outlier left in, the fit to the straight line is not going to be so good.

Some people failed to grasp that the R-squared values in the Figure were from the regression *leaving that point out.* Yes, the regression line will get pulled towards outliers, but even so, the error sum of squares will be larger, and thus the R-squared will be smaller, than it would be without the outlier.

I'm glad I asked this question, because it was a take on regression that you might not have seen before, and you needed some intuition about what happens in regression to figure it out. There were lots of ways to explain it that I liked, and the fact that I saw so many different ones suggests that you had to work out what to say, rather than repeating something that you had seen before. But the key is to note just how straight the regression is when you leave point #5 out.

Extra: I think this could get messed up by influential points (points that are off the trend *and* have an extreme x-value), but it suggests one way that you might look for outliers, one that would work in multiple regression as well: if leaving a point out of the regression results in a much higher R-squared than when that point is included, the point you left out is an outlier. (There's some theory that enables you to write down the matrix formulation of a regression with one observation removed, compared to the matrix formulation of the regression with all the observations, and this enables you to say something about how R-squared changes. This is within the grasp of those of you who have taken C67. It's in one of the textbooks on my office bookshelf. I don't remember exactly how it goes, but I know that it does go somehow. I think it's also tied in with Cook's distance.)

8. There is evidence that smiling can affect judgments of possible wrongdoing. This phenomenon, termed the “smile-leniency effect”, was the focus of a study in 1995.

   136 subjects were asked to judge a case of possible academic misconduct. The subjects each received a file of evidence that a student had cheated on an exam. This evidence was the same for all the subjects except for one thing: a photo of the student (who allegedly cheated) with one of four facial expressions: a “felt smile” (that is, a genuine smile), a false smile, a miserable smile and a neutral face. The facial expression was randomized to subjects, with each type of smile appearing the same number of times. (Yes, there is such a thing as a “miserable smile”, apparently.)

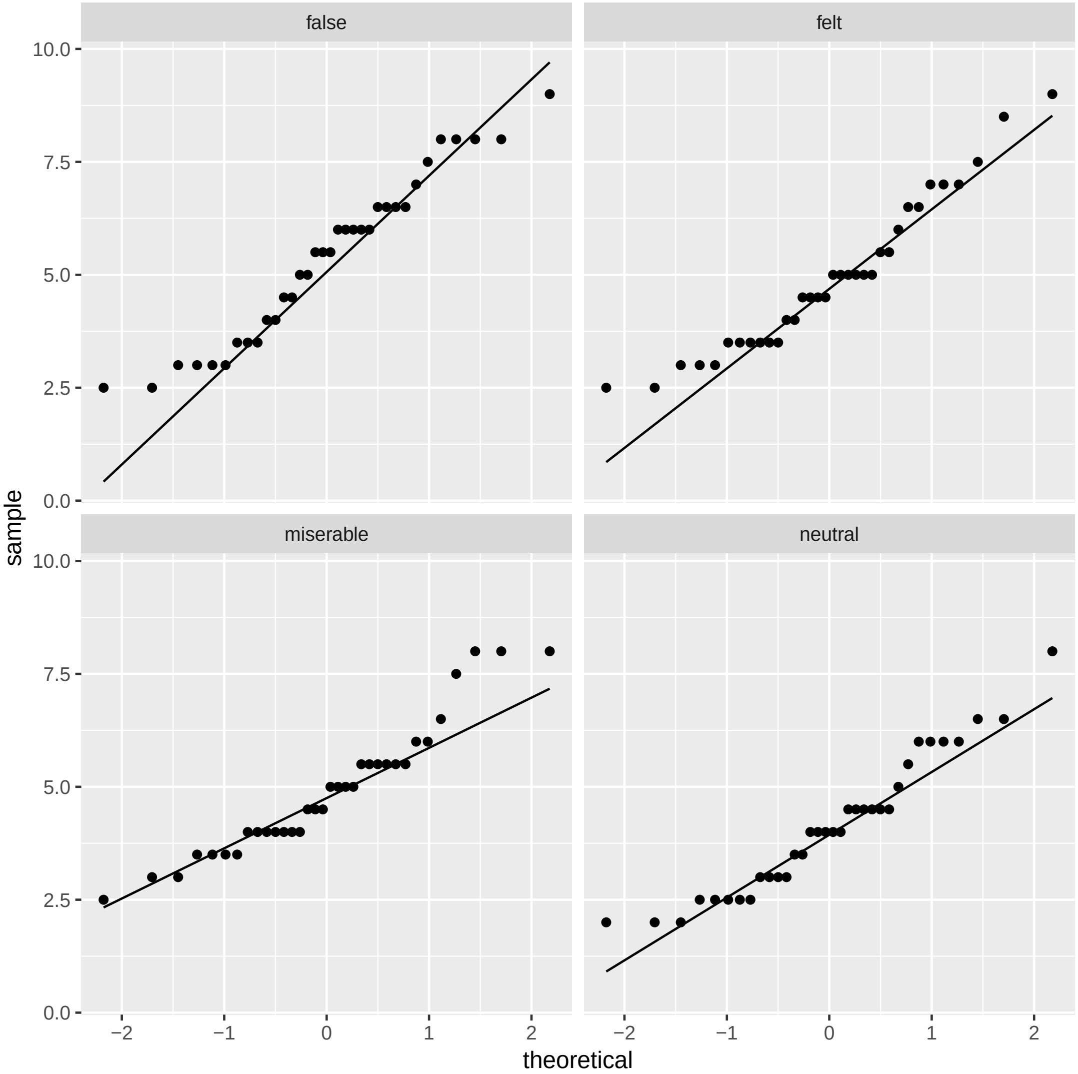
   After reviewing the evidence and looking at the photo they received, each subject had to respond to two questions on a 9-point scale. First, how they felt about the case as presented (the student should be given the benefit of the doubt, strongly disagree to strongly agree). Second, what they think an appropriate punishment should be (scale: cleared of all charges, up to given the maximum academic punishment). Each subject’s answers were combined into a “leniency score”, with a higher value meaning more lenient (the subject felt that the student should be punished less and/or that the student should be given the benefit of the doubt).

   Some of the data set is shown in Figure 19, and a table of sample sizes, means, and medians for each group (for all the data) is shown in Figure 20. The data frame is called `smile_leniency`.

   (a) (4 marks) Give the R code that was used to make the plots in **Figure 22**.

   Solution: These are facetted normal quantile plots, so you want the code for a normal quantile plot and for the facetting (by `smile`). This is what I used:

   ```
   ggplot(smile_leniency, aes(sample=leniency))+
       stat_qq()+stat_qq_line()+facet_wrap(~smile)
   ```



Remember that the `smile` doesn't come in until right at the end. What you imagine doing is making a normal quantile plot of *all* the `leniency` values together and then at the end make separate plots by `smile` group.

If you didn't notice the facetting part, you need to show how to make four separate normal quantile plots, one for each group. This means: using `filter` to pull out one of the groups, making a normal quantile plot of all the leniency values in the group you selected, and then repeating for the other groups. This will be very repetitive, so showing how to do one and describing the changes for the others is fine. It's enough to convince me that you know how to do it this way, without writing down every single thing. I guess it's also possible to do it using a loop, but you still have to pull out the data for the matching type of smile each time through the loop.

Marks: two for the normal quantile plot part, two for the facetting. Minus one per mistake, as per usual. Because the marks were split the way they were, if you wrote down the faceting part correctly, you guaranteed yourself two no matter what else you wrote down, or likewise making a normal quantile plot of `leniency` got you two no matter how much you messed up the facets. Making boxplots was Figure 21 (as well as being the kind of thing I would have asked on the midterm), and so was a fast route to zero.

My `facet_wrap` contained neither `ncol` nor `scales="free"`. A quartet of facetted plots comes out square (it's *three* of them that come out tall and skinny or short and squat), and my plots actually use the same scale. If you use the same scale on facetted normal quantile plots, you get to compare the centres (average height up the facet) and spreads (slope of the `qq_line`), but insisting on you seeing that here seemed unnecessarily picky, so you can freely include those or not in your `facet_wrap`.

The plot *is* a facetted plot (one graph containing four subgraphs), so if you produce four separate normal quantile plots, your maximum is 3 points. (It is also harder work to do it this way, and will take you longer to write down.)

(b) (2 marks) Data of this kind are often analyzed using analysis of variance. What are the two major assumptions required for an analysis of variance to be appropriate?

Solution: These two:

- Normally-distributed data within each group
- Equal spreads of data for all the groups ("equal spread between groups").

Feel free to add an "approximately" or a "sufficiently" to either or both of those.

One point for each of these. To get the first point, you need more than "normality": you have to say that *each group* is normally-distributed enough. Hint: when you find yourself writing "the data is" (or "are") in an answer, this is a warning that you can be more precise, and here is one of those places. What *about* the data has to be normal? The data *within each group* or *for each smile* or something like that. If you take all the data together, you are mixing up groups with possibly different means, so the data as a whole *won't* be normal even if the groups are (and regular ANOVA would be fine). See the Extra for more discussion of this.

Extra: since an ANOVA can also be viewed as a regression, you might be wondering how assessing each group for normality stacks up against looking at the normality of *all* the residuals together, as you would do in regression. For an ANOVA, the "predicted values" are the group means, so the residuals are the differences between each observation and its group mean. If all the observations are normal, possibly with different means but all with the same SD (our second assumption), then all the residuals taken together will have a normal distribution with mean zero and SD whatever that common SD is. So assessing all the residuals together for normality does make sense.

Note that this *does not* work for Welch ANOVA, because you are mixing together residuals with *different* SDs, so that the combined collection of residuals is *not* normal even if the residuals from each group are.

In this problem, "the residuals need to be normal" is part of the answer but not the full answer, because it fails if the group spreads are not equal. One point, therefore.

(c) (3 marks) Use any or all of Figures 20 through 22 to assess the assumptions you named in part (b). Do you conclude that the assumptions are satisfied, or not? Explain briefly.

Solution: Make a call about normality (one point) and equal spreads (one point), considering the sample size (one point). Any call that is supported by the information here is OK. The "explain briefly" is the process by which you get from the graphs to your conclusion, and is the most important part.

Figure 20 tells you that there are 34 observations in each group. These are largish groups, so we don't have to be all that picky about normality.

My take on the boxplots is that the values for `false` and `felt` are slightly right-skewed, and `miserable` and `neutral` have upper outliers. Given the sample sizes, I think I am OK with these, but feel free to make your own call here. As for equal spreads, the `miserable` group has a slightly smaller IQR than the other groups. Make a call about whether you think that matters.

On the normal quantile plots, I would say that all four distributions are actually not at all bad, given that the sample size is large. There is something odd happening at the top of the `miserable` group; you might say that those top three or four values are outliers. There is kind of an upper outlier in the `neutral` group (the one that showed up on the boxplot), but even that doesn't look too bad here. It's hard to judge equal spread on these (which is why I gave you the boxplots as well), but if you look carefully, the line for `miserable` is not quite as steep as the others, which is how a smaller spread shows up.

If you didn't get the same two assumptions in (b) as I did, assess the assumptions you *did* make, as best you can. The grading guideline is that you can get full marks in this part for doing so, as long as you haven't made it easier for yourself by doing so. (If you have made it easier, expect to lose something here.) If your assumptions were different from mine but still had something to do with normality, you should still consider the sample sizes.

You need to make a definite decision about normality and about equal spreads. I don't mind what that decision is (provided it's backed up by something), but you need to make one. This is because I want to be sure, in the next part, that you are choosing an appropriate analysis for an appropriate reason. If you don't make a definite decision here, you risk losing points here *and* in the next part too.

Extra: these are R normal quantile plots, so the line goes through the quartiles (theoretical and observed). So the line isn't getting pulled towards the outliers, as on a SAS `qqplot` with `mu=est sigma=est`. To my mind, the boxplots here look a lot less normal than the normal quantile plots do.

(d) (2 marks) Figures 23, 24 and 25 show three possible analyses of these data. Which analysis do you think is most appropriate? Explain briefly.

Solution: It depends what you thought about normality and equal spreads by looking at the plots:

- If you didn't trust normality, look at the Mood's median test, Figure 25.
- If you trusted normality but not equal variances, look at the Welch ANOVA, Figure 24.
- If you were happy with normality *and* equal variances, look at the regular ANOVA, Figure 23.

You see that any of these three answers are possible; what makes *your* answer right or wrong is its consistency with what you said in the previous part about the plots.

It was pretty easy to get two points here, as long as you avoided contradicting yourself. Sorry, double negative: as long as you were consistent with what you said before.

(e) (2 marks) Based on what you said in part (d), what do you conclude from your most appropriate analysis, in the context of these data?

Solution: The P-values from the regular ANOVA (0.0182) and Welch ANOVA (0.0217) are both less than 0.05, so if you picked one of those, you conclude that the mean leniency scores are not all equal, or that they depend on what kind of smile was in the photo. The P-value from the Mood's median test, 0.0676, is just *greater* than 0.05, so we do not quite have any evidence that any of the medians are different (or we conclude that all the medians are the same). (If you chose an α of 0.10, you will have a significant difference here as well.)

One mark for picking out the appropriate P-value (based on what you said before), and one for drawing an appropriate conclusion from it. I include the words "in the context of the data" in my questions for a reason: if you want the second point, you need to talk about smiles or leniency or both in your answer. Rejecting the null hypothesis, or failing to do so, is a stepping stone towards concluding what is actually important (in this case, that the type of smile does or does not make a difference in the amount of leniency shown). It is not a conclusion in itself, even if (in this case) you say that it demonstrates that the means (or medians) are (or are not) different from each other; the response to that is "means or medians *of what*?".

After that mini-rant, I should point out that there were some really good answers here that were very clear. But some, not so much.

I'm grading this and the next part separately according to what analysis you said was most appropriate, so I can tell more easily that you're getting your answers from the right place.

Extra: the regular and Welch ANOVA P-values are pretty close, so it didn't matter much whether you thought the spreads were similar or not. The non-significant Mood's median test is probably lacking power compared to the other two, since all it does is count values above and below the overall median leniency score. My take here is that the mean is a pretty good measure of centre for these data, not overly influenced by the single outliers or the skewness in any of the groups, so we might as well take advantage of the extra power of the other two tests.

(f) (3 marks) Figures 26, 27 and 28 show some further analysis. Which, if any, of these Figures should you look at? What do you conclude, in the context of the data? (If you are not entitled to look at any of these Figures, explain briefly why.)

Solution: One point for picking the correct followup for the analysis you looked at before. That is:

- if ANOVA (Figure 23), then the Tukey, Figure 28.
- if Welch (Figure 24), then the Games-Howell, Figure 27.
- if Mood's median test (Figure 25), then the pairwise median tests, Figure 26. Or see below.

Two points for drawing the appropriate conclusion from your followup analysis:

- for Tukey or Games-Howell, the only significant difference is between the neutral face and the false smile, with the latter being higher (look back at Figure 20). You ought to say or imply that none of the other differences are significant. (The word "only" is good for this.)
- For Mood's median test, the best answer is to say that because it is not significant, we should not look at *any* further analysis. However, if we do look at the pairwise median tests, none of them are significant at $\alpha = 0.05$ anyway, so we end up in the same place,

and I would take either thought process. (Usually what happens when the main analysis is not significant, none of the paired comparisons in the followup are either, but it's not a guarantee.) If you did this test with $\alpha = 0.10$, you'll come to the same kind of conclusion as above: only the false smile and the neutral face differ significantly in median leniency. Note that you need to look at the *adjusted* P-values here, not the unadjusted ones in the third column, since we are doing (here) six tests at once and we have to account for that.

I guess that means that if you pick the wrong followup, but you interpret correctly the one you picked, that's two points.

If you picked Mood's median test (before) and declined to look at any of the other Figures because the Mood test was not significant, that's three points. (If you want to mention that *normally* we'd follow up with the pairwise median tests, you can, but I don't insist on it.)

Extra: the experimenters were probably disappointed by this; even though they had a lot of data, there was a lot of variability and they didn't demonstrate much. I'm sure they would have liked to distinguish better among types of smiles.

Extra 2: I have to admit that I juggled the order of the followup tests to see whether you were paying attention.

Extra 3: I took a look at the original paper: the authors' advice at the end was "if guilty, smile", even a false smile. I'm not sure what that implies about the world.

9. A manufacturer of felt-tip markers wanted to know whether a new advertising display, featuring a picture of a physician, was more effective at selling the markers than the current display, featuring a picture of an athlete. The current display was located in the stationery area (of a drugstore chain), while the new display could be located in either the stationery area or at the checkouts. Fifteen stores took part in the study. In all of them, the current display (the athlete picture, located in the stationery area) was used for three weeks, and the sales recorded. Then, for the next three weeks, each store was randomly allocated one new display: either (i) the athlete display in the stationery area, as before (control), (ii) the physician display in the stationery area, or (iii) the physician display at the checkouts. Sales were then recorded for those next three weeks, in suitable units. The data are shown in Figure 29. `sales1` and `sales2` refer to the sales in the first and second three-week periods.

   (a) (2 marks) Why do you think the sales for the first three-week period were recorded, even though the new display was not being used?

Solution: Because some of the stores might sell more markers than others, and the manufacturer wanted to get an idea of "typical" sales: that is, to establish a baseline. Or, the manufacturer wanted to see whether sales *increased* from whatever they were before. Something like that. A relevant sentence with the word "compare" in it is probably good for two points.

The word "baseline" is better than "control" in this context, because the control group here is really the stores that had the athlete display in the *second* three-week period. But I was happy with people who said "control" here as long as they got at the idea that the first three-week period was to provide something to compare with.

Another insightful way to look at it is to think of it as being matched pairs: we have two sales measurements for each store (before and after), so we have a comparison *within* stores (and we can partition off the variability due to the stores being different from each other, or any other things that might affect sales at one store rather than another, because we are effectively comparing stores with *themselves*). Another way to analyze these data, rather than doing an analysis of covariance as I did, is to calculate the *differences* of sales after minus before, so that each store produces one "change in sales" number, and see how that depends on the (new) display using an ordinary one-way ANOVA. But we already had one of those on this exam, and so I wanted to give you something different.

We also have some machine-learning people around, since I saw "training" and "testing" a couple of times, which is another nice way to see it.

I was generally pretty relaxed about this one; if you said any of these things, you should have two points.

(b) (2 marks) A regression model was fitted, predicting `sales2` from `sales1` and `Treatment`. Output from the fit is shown in Figures 30 and 31. What do you conclude from Figure 30, in the context of the data? Explain briefly.

Solution: This shows whether either or both of the previous sales and the treatment (type of display) have any effect *on current sales*. Both P-values are small, so they both do.

I am looking for (i) the P-values are both small, and (ii) what that means in the context of the data. There was a somewhat fuzzy line between 1 and 2 points; if it looked as if you said something sensible about `sales2` being related to `sales1` and `treatment`, you'll have come out on the right side of that line. There were some really nice answers; it was clear that some people really understood clearly what was going on.

(c) (3 marks) In Figure 31, what precisely does the number 19.12547 tell you, in the context of the data?

Solution: This is a "slope" attached to a categorical variable, so it is a comparison with the baseline, which is the "control" treatment `athlete`. This says that, for stores that have the same sales in the first three weeks, the one having the physician display in the stationery area is predicted to have sales 19.12 bigger than the one that has the athlete display (in the stationery area).

Grading: 3 points if you get all of this. 2 points for something like "the physician display in the stationery area increases sales by 19 units", without saying what this increase is compared to. 1 point if you said anything of some relevance without getting to this point (like "increasing `physician-stationery` by 1", which you can't do because `treatment` is categorical: either the treatment is this or it's something else).

Extra: both the estimate values for the two physician displays are noticeably bigger than zero, so both of these displays produce greater sales than the athlete display, taking sales in the first three weeks into account.

Extra 2: in a previous version of the exam, I also asked you about the number 0.83474, which is an ordinary slope of a quantitative variable (if you increase `sales1` by 1, leaving `Treatment` fixed, how much does `sales2` increase). But I felt the exam was getting too long, so I removed this, and also a question about code to make Figure 34. Both of these were partial repeats of another question.

(d) (2 marks) The previous regression assumed that the rate at which sales in the second three-week period increased with sales in the first three-week period was the same for each treatment. Is that supported by the graph in Figure 34? Explain briefly. (Note that the graph is at the end of the Booklet of Code and Output.)

Solution: The issue here is whether those lines are close enough to parallel or not (that is, do they have the same slopes?). I'm guessing that you will say that the blue line is less steep than the red and green ones, so they are not parallel enough for that reason. Or you can say that the lines are not all that far from being parallel, especially given only five stores in each group, so it is believable that the rates of increase are the same. I'm not picky. Go with one and some kind of justification of it.

There is definitely something (probably 1) for saying that all three lines are going uphill, but I would like a little more than that.

Extra: should you wish to test for parallelism of the lines for each treatment, you do it by including an interaction, thus:

```
markers.2=lm(sales2~sales1*Treatment, data=markers)
drop1(markers.2, test="F")
```

```
## Single term deletions
##
## Model:
## sales2 ~ sales1 * Treatment
##                  Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>                        145.20 46.051
## sales1:Treatment  2    31.329 176.53 44.982  0.9709 0.4151
```

Coding-wise, the `*` means "both the main effects of and the interaction between the two things joined by a *". If you want to refer to just the interaction term itself, you use :. Thus, I could also have done this:

```
markers.3=update(markers.1,.~.+sales1:Treatment)
drop1(markers.3, test="F")
```

```
## Single term deletions
##
## Model:
## sales2 ~ sales1 + Treatment + sales1:Treatment
##                  Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>                        145.20 46.051
## sales1:Treatment  2    31.329 176.53 44.982  0.9709 0.4151
```

to add just the interaction term on top of what was there before.

In the `drop1` output, the interaction is "higher-order" than the terms for `sales1` or `Treatment` by themselves (the so-called "main effects"). So you have to make a decision about it first, and if you keep it, you also have to keep anything contained in it. Here, though, the interaction is nowhere near significant, so there is no value in keeping it: the slopes are "not significantly different".

It's kind of pointless, but just to show you — if you look at the output from `summary` for this model:

```
summary(markers.3)

##
## Call:
## lm(formula = sales2 ~ sales1 + Treatment + sales1:Treatment,
##     data = markers)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9092 -2.1073 -0.3539  2.4037  5.3963
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             -1.3506     9.8212  -0.138  0.89365
## sales1                                   0.7685     0.1375   5.588  0.00034
## Treatmentphysician-checkout              9.3656    13.6954   0.684  0.51129
## Treatmentphysician-stationery           51.6274    37.5361   1.375  0.20227
## sales1:Treatmentphysician-checkout       0.1823     0.1987   0.918  0.38270
## sales1:Treatmentphysician-stationery    -0.4239     0.4968  -0.853  0.41563
##
## (Intercept)
## sales1                                 ***
## Treatmentphysician-checkout
## Treatmentphysician-stationery
## sales1:Treatmentphysician-checkout
## sales1:Treatmentphysician-stationery
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.017 on 9 degrees of freedom
## Multiple R-squared:  0.9498,Adjusted R-squared:  0.9219
## F-statistic: 34.05 on 5 and 9 DF,  p-value: 1.384e-05
```

the terms at the bottom with a colon in them, like `sales1:Treatmentphysician-stationery`, express changes in *slope* from the baseline. This one is -0.4239, which says that the slope for the physician display in the stationery department is 0.42 less than the slope for the athlete display (baseline). This kind of difference in slopes is not nearly big enough to be significant.

I get into this a bit more in STAD29, where we talk about analysis of covariance (the fancy name for this) in a bit more detail.

10. You might recall the 68–95–99.7 rule from a previous course. It is sometimes called the "empirical rule". It says that 68% of the values in a normal distribution with mean μ and standard deviation σ lie within one standard deviation of the mean, 95% of them lie within two standard deviations of the mean, and 99.7% of them (that is, almost all) lie within three standard deviations of the mean.

For those who prefer this mathematically: if X has a normal distribution with mean μ and SD σ, then $P(\mu - \sigma < X \le \mu + \sigma) = 0.68$, $P(\mu - 2\sigma < X \le \mu + 2\sigma) = 0.95$, and $P(\mu - 3\sigma < X \le \mu + 3\sigma) = 0.997$.

This idea gives us an alternative way to estimate `sigma` for use in a SAS normal quantile plot, as we explore. You do not need any normal tables for this question.

(a) (4 marks) Using this rule, explain briefly why $\mu - \sigma$ and $\mu + \sigma$ are the 16th and 84th percentiles (respectively) of a normal distribution.

Solution: $\mu - \sigma$ and $\mu + \sigma$ are respectively one standard deviation below and above the mean, so 68% of the normal distribution is between them. (One point.) The remaining 32% is more than one SD away from the mean, and because the normal distribution is symmetric, half of it (16%) is below $\mu - \sigma$ and half of it is above $\mu + \sigma$. (The second point).

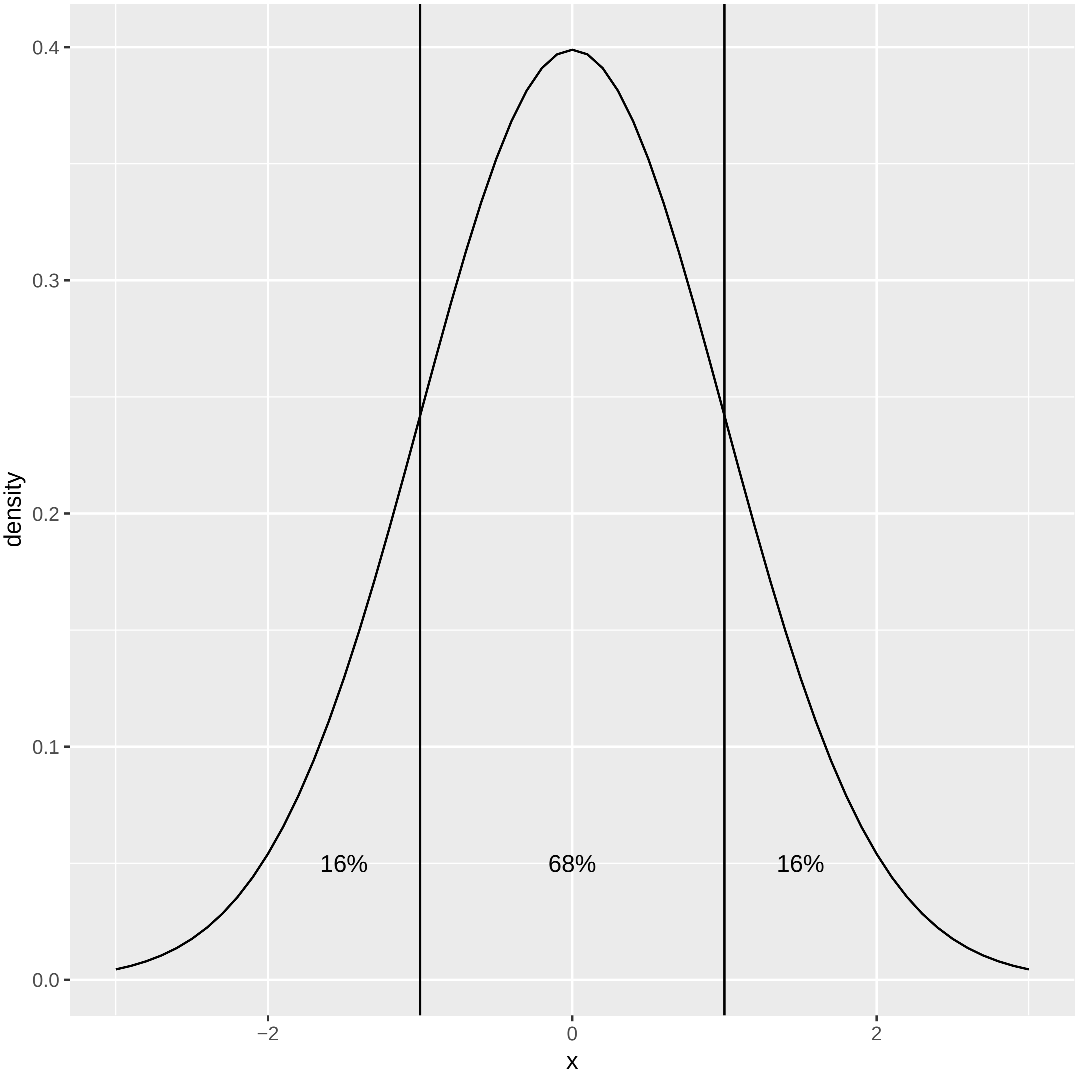
Since $\mu - \sigma$ has 16% of the normal distribution below it, it must be the 16th percentile of the normal distribution. (The third point.)

$\mu + \sigma$ has 16% of the normal distribution *above* it, so it must have $100 - 16 = 84$% of the normal distribution below it, and therefore is the 84th percentile. (The fourth point.)

If you did it a different way, or tried to, I attempted to grade you according to how far along a sensible path you seemed to get. If you said *something* sensible, you ought to get 1, and if your thought process was *almost* sound, you should have 3.

If it is clear from your explanation that you understand parts of this, you can get credit for them even if you don't explicitly say them. For example, you might draw a picture. This one is a standard normal (because it was easier to draw):

```
d=tibble(x=seq(-3,3,0.1))
w=tribble(
    ~x, ~y, ~text,
    -1.5, 0.05, "16%",
    0, 0.05, "68%",
    1.5, 0.05, "16%"
    )
d %>% mutate(density=dnorm(x)) %>%
    ggplot(aes(x=x, y=density))+geom_line()+geom_vline(xintercept=-1)+
           geom_vline(xintercept=1)+
    geom_text(data=w, aes(x=x, y=y, label=text))
```



and reason out the areas below $\mu - \sigma$, above $\mu + \sigma$ and between them, and get $\mu + \sigma$ as the 84th percentile by adding $16 + 68 = 84$. That would be a full-credit answer, properly justified.

Another cute way to argue it is to say that the mean is the median and has 50% of the distribution above and below, so if you halve the middle 68% you can go that much up and that much down from 50% to get the same result.

Starting with the answer needs care: yes, it is true that $84 - 16 = 68$, but I think you also need to mention that 16% and 84% are symmetrically placed so that they make the *middle* 68%, not just any old 68%. Something like three points for this approach.

Extra: technique for the graph above is to construct two data frames, one for the normal curve (a whole bunch of x-values and the normal density at each one), and the second of the places to put the text (68% and 16%) and the text to put there. Then, plot the normal curve with vertical lines at ±1. I did this first to work out where to put the numbers on the plot. `geom_text` puts text *at* points (as opposed to labelling points, where you want the text to be beside the points). Having a data frame of x and y coordinates and the text to put there makes it easier. Here, though, I have to say in the `geom_text` that I now want to use the second data frame I made, and the things I called `x` and `y` in that.

(b) (2 marks) Consider the difference d between the 84th percentile and the 16th percentile. In a normal distribution, what is d in terms of μ and σ? (d might depend on one or both of these.)

Solution: The 84th percentile is $\mu + \sigma$, and the 16th percentile is $\mu - \sigma$, so the difference between them is $(\mu + \sigma) - (\mu - \sigma) = 2\sigma$. (It winds up not depending on μ at all.)

(c) (2 marks) How, therefore, could you use the 16th and 84th percentiles of sample data to estimate σ? Explain briefly.

Solution: The difference between them in a normal distribution is 2σ, so take the difference between the sample percentiles and divide it by 2.

This is the same idea as using the interquartile range and dividing by 1.35 (I got the idea for this question from talking with a student about this very issue). The 16th and 84th percentiles are further out than the quartiles, so we have to divide by more than 1.35 to estimate σ "unbiasedly". (I'm being slightly loose in calling this "unbiased" because it's not really averaging over repeated samples; the difference between the 84th and 16th percentiles *is* 2σ in a normal distribution, in the sense that it's another way of looking at the same thing. I guess the sample 84th and 16th percentiles are reasonable estimates of the same things in the population; they may not be unbiased, because there are different ways to define them.)

This suggests that we could compare various percentiles symmetric around the median to see which pair estimates σ most accurately. (I feel a blog post coming on). Dividing by the right thing will make the estimator "unbiased" (see discussion above), and then the question becomes how close the estimates of σ are to the true value: that is, how the variances compare. This could be done by simulation: generate some random samples from normal (or other distribution), and for each one compute the percentile-based estimates of σ, then work out the variances (or standard deviations or IQRs) of the estimates. If we generate enough random samples, we might see that there is a pattern. I have a feeling that for normally-distributed data, you want to go further out, up to a limit maybe (this turns out to be the case), but for data from a distribution with long tails, you *don't* want to go so far out, because you could be tripped up by outliers (the reason we got into this in the first place).

Suggesting using the IQR here isn't going to profit you very much because you don't have the quartiles, you have those two percentiles.

If you messed up the previous part, an honest attempt to use whatever you found there should net you two points here.

Another (not quite the same) approach is note that the two quartiles are $\mu \pm \sigma$, estimate μ somehow (in the spirit of this, the median is the obvious thing, but the mean is not unreasonable and I accept this too), and then the difference between the estimate of μ and one of the percentiles is an estimate of σ. Use either, or both and average them. Some people noted that the two percentiles are the same distance either side of μ, so you could also estimate μ by averaging the two percentiles, and then estimating σ as I described just above. Any of these work for me.

This last way of estimating μ is in the spirit of what Tukey called the "midhinge" ("hinge" was Tukey's word for "quartile"): he suggested to estimate centre using the average of the first and third quartiles. If the distribution is right-skewed (say), this will be pulled up a bit by the skew, but not as much as the mean would be, because the quartiles have what Tukey called "high breakdown": you would need a *lot* of outliers before the quartiles would be damaged by them.

(d) (2 marks) Consider the data in Figure 32, whose 16th and 84th percentiles and median are shown in Figure 33, labelled `x16`, `x84` and `x50` respectively. Use these values to estimate `mu` and `sigma` for a SAS normal quantile plot.

Solution: Estimate `sigma` as described above: take the difference between the 84th and 16th percentiles and divide by 2, using your calculator:

```
(48-26)/2

## [1] 11
```

and estimate `mu` using the median 36.

If you somehow came up with some different relationship between the median and 16th and 84th percentiles and μ and σ, use that here (consistency). You'll lose marks earlier, but you can get full marks here in that case. I had some fun and games trying to figure out if you had done that; if I thought you had, you got two marks. (You can make it easier for me and thus yourself if you show me your calculation; doing so will help to convince me. If you didn't, and I wasn't sure, you might have only gotten one.)

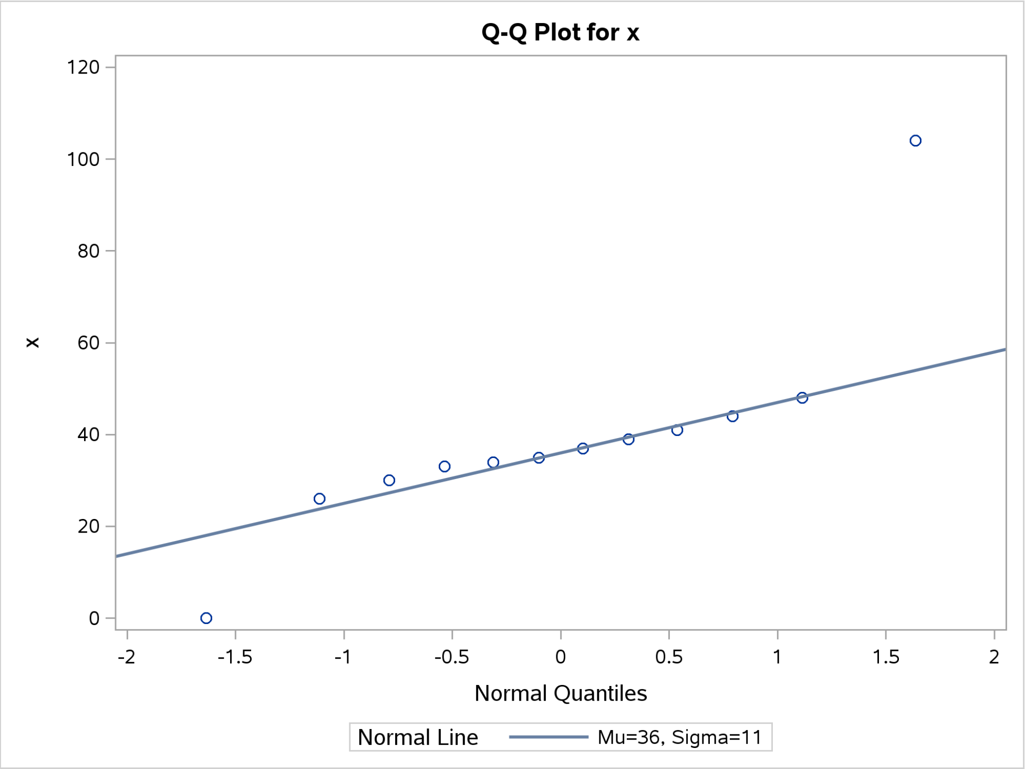
If you didn't have anything else to work from, you know from back in (a) that the 16th percentile is $\mu - \sigma$ and the 84th percentile is $\mu + \sigma$. Put in the data values for the percentiles and solve for μ and σ (two equations, two unknowns).

Extra: statisticians distinguish between "estimator" and "estimate". The former is a procedure such as a formula or a set of equations, while the latter is a number (obtained from a data set). Here I asked for an estimate, so what I want is a number (or, in this case, a pair of numbers).

(e) (3 marks) Give SAS code to draw a normal quantile plot of the data `x` that are shown in Figure 32, using your estimated μ and σ.

Solution:

```
proc univariate noprint;
  qqplot x / normal(mu=36 sigma=11);
```



The thing to make sure of is that you use your estimates for mu and sigma from the previous part, whatever they were. If you couldn't get any values, make some up, or show where you would insert the values you had, if you had any. If you can show that you know or would know how to do this part via some means, you can get full marks here even if you couldn't do the previous part. As long as you had some plausible looking numbers in there, I was good.

The noprint is optional. Minus one per error as usual.

Some things to note:

- The variable was called x, not diff as I had in class (and a lot of you seemed to have in your notes).
- The whole point of this question was to provide you with some values for mu and sigma, so mu=est sigma=est was a definite error here.
- There is supposed to be *no comma* between mu= and sigma=, but at this stage in the exam I wasn't going to be picky about that, so I forgave it if you put one in. (If you lost points it was for one of the other two things.)

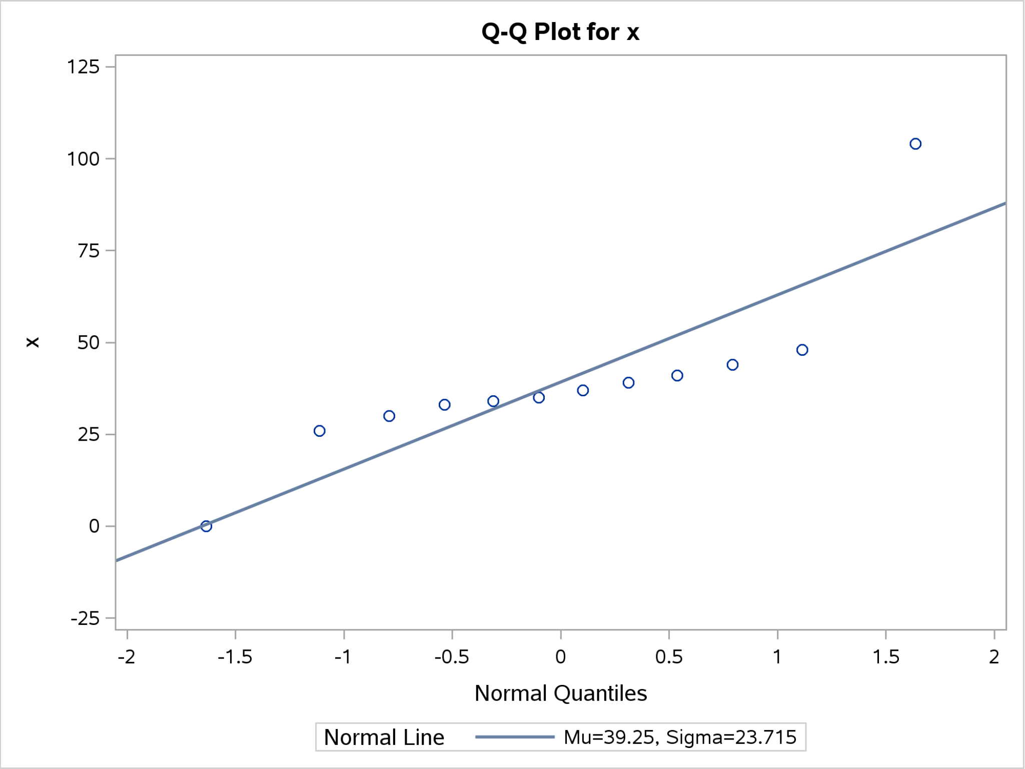
(f) (2 marks) What would be the advantage to estimating μ and σ as in this question, rather than the way SAS does it using mu=est sigma=est? Explain briefly.

Solution: For you, "the estimates would not be affected by outliers" is all I need.

SAS by default uses the sample mean and SD to estimate mu and sigma with. These are affected by outliers; for example if we have an upper outlier (as the plot above does), the mean will be pulled upwards and the SD will be increased because of the observation a long way from the mean. Using percentiles, like using quartiles, will remove the influence of outliers.

The effect on the plot is seen above: the line goes through the middle points and makes the outliers at the top (and maybe at the bottom) stand out. If mu and sigma had been estimated using the sample mean and SD (using mu=est sigma=est), the line would have been steeper and the outliers wouldn't have stood out so much:

```
proc univariate noprint;
  qqplot x / normal(mu=est sigma=est);
```



Here, nothing is very far from the line and you might be tempted to call this “approximately normal”. As the previous plot shows, that would be a mistake.