MA26620: Practical 6

Practical 6: ♫ All I Want For Christmas Is... a solid understanding of two sample t tests and the ability to use R to conduct t tests ♫

Good morning. Today's practical is loosely Christmas themed, as it's the last one before Christmas. This may seem ever-so-slightly jarring when you're re-reading this as part of your May exam revision, but at the time of writing, I'm even wearing a Christmassy jumper, so it's fine. We'll begin with an example of conducting a two sample t-test with pen, paper and tables (you won't need R for this although the resources on the module webpages may be useful) and then we'll move on to see how to do such tests in R.

1 Two Sample t-test (without using R)

(See your lecture notes and/or the content on the module webpages and/or ask us for help during the practical) The amount spent on Christmas presents for friends and family is recorded for 10 residents of Aberystwyth and 7 residents of Machynlleth:

	Amount spent (£)										**Mean (£)**	**Var (£^{2})**
Aber	165.80	141.10	103.40	196.84	177.66	190.00	166.00	113.23	114.96	161.84	**153.08**	**1104.494**
Mach	50.55	100.92	176.89	18.63	165.27	193.41	230.19				**133.69**	**6161.200**

Clearly stating any assumptions you make (though do not assume equal variances) and explaining any notation you introduce, assess whether Aberystwyth residents spend more on Christmas presents than Machynlleth residents.

2 Binomial

1. Given the increasing world population, his increasing years and his propensity to double-check any lists, Father Christmas ($\in \mathbb{R}$) has recently found that assessing whether people are *naughty* or *nice* has made his workload almost unbearable. From his many years of experience, he knows that typically 80% of the Aberystwyth townsfolk are nice, while 20% are naughty (he deems it impossible to be simultaneously naughty *and* nice).

   In an attempt to speed up the classification process, Father Christmas decides to trial a fiendishly complicated new Python script expertly programmed for him by a keen second year mathematics student which claims to be able to speedily conduct the classification process by eliminating the usual copious paperwork. However, he is concerned that the new script may be harsher than his traditional methods, and thus may lead to fewer people being classified as nice.

   Should Father Christmas be concerned if the Python script assesses that:

   (i) 7 out of 10 Aberystwyth townsfolk are nice?

   (ii) 14 out of 20 Aberystwyth townsfolk are nice?

   (iii) 98 out of 140 Aberystwyth townsfolk are nice?

3 Using R to conduct hypothesis tests and to compute confidence intervals

3.1 Two samples

After Section 1 of today's practical, the two sample t-test is fresh in our mind, so let's start by learning how to conduct a two sample t-test in R before then looking at one-sample t tests.

Enter the data from Section 1 into R into two variables using

```
aber <- c(165.80,...,161.84)
mach <- c(50.55,230.19)
```

We met the F-test in lectures briefly as a way to determine whether an assumption of equal variances. We can easily run such a test by running the command `var.test(aber,mach)`. You'll see that the resulting p-value is about 0.023, certainly less than 0.05, so we'd reject the null hypothesis of equal variances here. This justifies why I told you not to assume equal variances earlier.

Next, combine the spends into a single variable: `spending <- c(aber,mach)`. Then create another variable, `townName`,

```
townName<-rep(c("Aberystwyth","Machynlleth"),c(10,7))
```

This just makes a vector consisting of Aberystwyth written ten times, then Machynlleth written seven times.

We will now use the command `t.test` using R's modelling notation, `y~x`. We have previously encountered this ~ notation in connection with `ggplot` and `lm` in earlier practicals. Try

```
t.test(spending~townName, alternative="greater", mu=0, var.equal=F,
                                                conf.level=0.95)
```

R throws out quite a few lines of useful output. You can type `?t.test` to see all the input options you have.

The line `t = 0.61602, df = 7.5214, p-value = 0.278` gives you the t-statistic, degrees of freedom and p-value. Here the p-value is high, so we have no evidence to doubt $H_0 : \mu = 0$ where μ denotes the difference in mean spend $(\mu_{\text{Aber}} - \mu_{\text{Mach}})$.

Also among the output is a 95% confidence interval for μ and the values of the means of each sample.

3.2 One sample

Suppose we instead only had one sample from a population with unknown variance and wanted to test something about the population mean under the usual t assumptions. For instance, would we believe that the population mean Christmas spend of people in Aberystwyth is in fact £180? Re-enter just the Aberystwyth data, perhaps calling it `aber` and then run `t.test(aber, mu=180, alternative="two.sided")` What would you conclude from this output?

Note also that the output includes a 95% confidence interval for the population mean μ. Suppose you wanted a 99% confidence level instead - can you get R to tell you this?

4 Paired t tests

In our earlier two sample example, our samples were independent of each other (the first Aberystwyth family had nothing to do with the first Machynlleth family). We now consider a case where the two samples are not independent of each other but are instead *paired*.

Download `spider.csv`. This contains anxiety scores for 12 arachnophobes on being shown (i) a video of a spider emerging from the branches of a Christmas tree in close-up and (ii) a real spider emerging from a real Christmas tree.

Read the data into a dataframe which you should inspect and attach. The data are paired since each individual was tested twice, once with a real spider and once with a close-up picture. This dataframe is said to be in 'wide format' with data from different groups in columns side by side and each row representing one individual's scores. This is different to the Aberystwyth/Machynlleth data which would be said to be in 'long format'.

Check whether the data is reasonably close to Normal.

```
qqnorm(Real-Picture, main="Differences in Anxiety Scores")
qqline(Real-Picture)
```

A paired t-test can be implemented using the `t.test()` command. Since we have no particular reason to assume that the real spider will be scarier than the picture, or vice versa, we use a two sided alternative.

```
t.test(Picture, Real, alternative="two.sided", mu=0,paired =T)
```

Note the different form of t.test for wide format.

What hypotheses are being tested here? Should you reject the null hypothesis and, if so, at what level? What conclusion do you come to about the relative impact of real and pictured spiders on these subjects? Copy the output to a Word document and write a paragraph describing what you have concluded and why.

What happens to the output if you swap the order of the two variables, Picture and Real? Make sure you understand why. What happens if you change the setting of `alternative`? Try greater and less.

Repeat the t test with the setting `paired=F`. There are several striking differences between this output and the `paired=T` case. What are they?