Research Methods

Robert E Simpson

Singapore University of Technology & Design

robert_simpson@sutd.edu.sg

October 16, 2018

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Objectives

You should be able to:

- Perform t-tests (2 sample difference, paired difference) using python
- Perform χ^2 test using python
- Perform a χ^2 goodness of fit test using python
- Draw flow diagrams to show how to computationally perform maximum likelihood estimation
- Write python program to compute the most likely value of of parameters in a distribution

There are four case problems in this class.

1 sample T-test

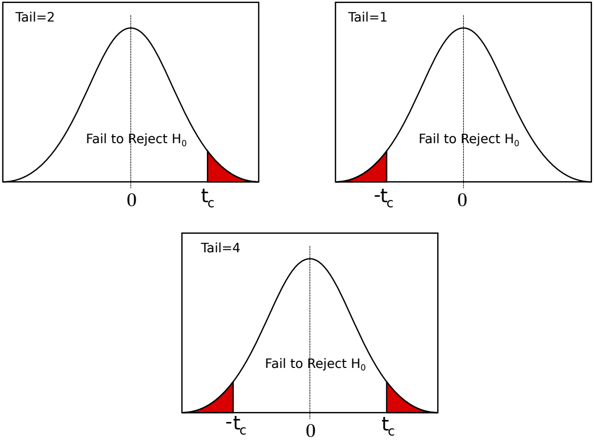
Used to test whether the mean of a small number of samples is signifficantly different from a specific value. The function will give the t-statistic (similar to the standard deviation for a normal distribution) and the p-value relative to the specified mean.
The scipy.stats t-test performs a two-tailed test and you must divide the p-value in half for a one tailed test.

```
stats.ttest_1samp(data, mean)
```

`data` a list of random variables

`mean` is the hypothesised mean against which we are comparing the sample distribution

One tail or two?



Two independent samples: difference test on means

```
scipy.stats.ttest_ind(a, b, axis=0, equal_var=True,
    nan_policy='propagate')
```

The difference in the means of the two waves is calculated according to:

$$t = \frac{(\bar{x}_1 - \bar{x}_2) - (\bar{\mu}_1 - \bar{\mu}_2)}{\sqrt{\left(\frac{s_1^2}{n1} + \frac{s_2^2}{n2}\right)}} \tag{1}$$

Paired difference test

```
scipy.stats.ttest_rel(a, b)
```

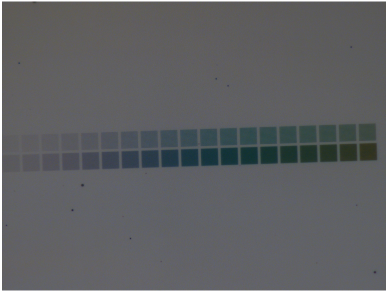
The difference is taken between the individual measurements. Thus systematic variations can be negated.

$$t = \frac{\bar{d}}{\sigma} \tag{2}$$

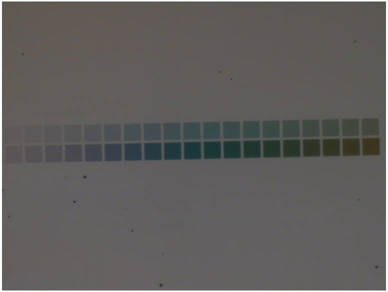
$$\text{where } \sigma = \frac{s_d}{\sqrt{n}} \tag{3}$$

Settling Arguments with PhD Advisor

We were developing a new display technology. We deposited a film of Si3N4 on top of the display pixels. Does the Si_3N_4 influence the RGB ratios of the display?



Before Si_3N_4



After Si_3N_4

Weiling Wins Argument

The null hypothesis was that the samples are the same. We set α=0.05

n = 19
degrees of freedom = 18
t_{Value} = -0.268355
t(α = 0.05) = 2.1
P = 0.79

$P > \alpha$, therefore we cannot reject H_0. We conclude that statistically the Blue to Green colour ratio is the same at a 5% level of significance.

Tests for distributions

χ^2 statistic applies to discrete data:

- Used to test the hypothesis that a set of sample data does not differ significantly from some theoretical distribution.
- it is a goodness-of-fit test

$$\chi^2 = \sum_k \frac{(f_{obs} - f_{exp})^2}{f_{exp}} \tag{4}$$

f_{obs} is the observed frequency
f_{exp} is the expected/modelled frequency
k is the number of 'bins'

If the observed data agrees exactly with the model, then $\chi^2 = 0$

Restrictions for $\chi^2 = \sum_k \frac{(f_{obs} - f_{exp})^2}{f_{exp}}$

To use the χ^2 test properly, the following restrictions apply

- Sample size should be greater than 30
- None of the expected frequencies should be less than 5.

In python the χ^2 test is invoked by:
`scipy.stats.chisquare(measured,modelled)`

Maximum Likelihood Estimation

The objective is to find the parameters of a function that most likely will lead to an observed set of outcomes.

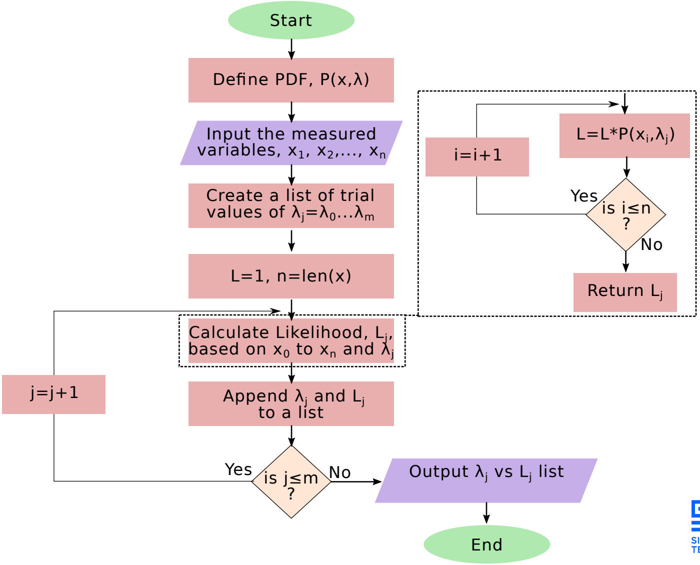
Suppose a sample (x_1, x_2, ... x_n) is drawn from a population with a probability function p.
If the sample is random, then $p = p(x_1).p(x_2)...p(x_n)$, then the likelihood is thus:

$$L = \Pi_{i=1}^{n} p(x_i) \tag{5}$$

Maximum Likelihood Estimation

- It can handle any type of error distribution.
- Straight Forward and easily solved by computers
- Can show a range of plausible values and for deducing confidence limits
- Can be used when there is no knowledge of the underlying distribution but needs an analysts input.

MLE Procedure



Case problem 1

Luminosity

A supplier gives an average rating for a particular lamp at 2650 lumens. A random sample of 10 lamps provides the luminosity measurements below.

2661	2624
2641	2670
2650	2638
2645	2652
2628	2675

a) Assuming we would like to compare the mean luminosity, what are the possible hypotheses that we could test?

b) Is the average luminosity $<$ my requirement of 2650?

CP6.6 Soln: Possible Hypotheses

$$
\begin{aligned}
H_0 &\quad \mu = 2650 \\
H_{a1} &\quad \mu > 2650 \\
H_{a2} &\quad \mu < 2650 \\
H_{a3} &\quad \mu \neq 2650
\end{aligned}
$$

Hypothesis

"Is the average luminosity < my requirement of 2650?" The hypothesis can be stated succinctly as...

$$\text{The Null hypothesis, } H_0 \quad \mu = 2650 \tag{6}$$
$$\text{The alternative hypothesis, } H_a \quad \mu < 2650 \tag{7}$$

2661	2624
2641	2670
2650	2638
2645	2652
2628	2675

Set Acceptable Risks

Set $\alpha = 0.05$ (The risk of rejecting H_0, given that it is true)

Therefore, first find the critical t-value for a signiggicance of 0.05 and 9 degrees of freedom. You can look up this in the T-tables, or simply run the following python command.

```
print ss.t.ppf(q=0.05, df=9)
-1.83311293265
```

Next, we do a one sample t-test on the mean luminosity to check whether it is statistically less than 2650. Again, this can be done with python:

```
lum=[2661., 2624., 2641., 2670., 2650., 2638., 2645.,
    2652., 2628., 2675.]
print ss.ttest_1samp(lum, 2650)
Ttest_1sampResult(statistic=-0.30151134457774642,
    pvalue=0.76987499989214936)
```

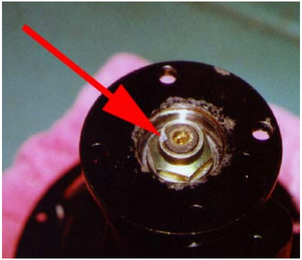
The T-statistic=-0.3015. This is substantially smaller than the critical t-value=-1.833. Therefore we cannot reject H_0 at the 0.05 signifficance level. Note, the pvalue=0.76987 corresponds to a two-tail test but we are doing a one-tail test, therefore $p = \frac{pvalue}{2} = 0.384935$. Again $P = 0.384935 > \alpha$. This also shows that we cannot reject the null at the 0.05 significance level.

- From the available statistical evidence we infer that the population mean from which the sample came is not significantly smaller than 2650 lumens.
- Does this prove that the null hypothesis is true (that the true population mean = 2650)?
  - Unfortunately not...
- We have evidence to "not reject" the null, but not to "prove" it.
  - Statistically, all we can ever do is **not reject**.
  - We can never **prove** nor **accept**.
- However, we usually operate under the assumption that not rejecting is practically the same as accepting.

Case problem 2: Effect of Staking on Axle Vibrations

- Axial movement, in microns, was measured on 10 motors selected at random.
- Each of these 10 motors then had their hub nuts staked[1] and the axial movement was remeasured.

Is the average axial play affected by staking or not?



[1]Staking is when you use a chisel to indent the nut into the shaft. This may prevent the nut from backing off the axle shaft

Case Problem 2: Effect of Staking on Axle Vibrations

Motor	Axial Play with Staking	Axial Play W/O Staking
1	19.0	19.4
2	19.4	19.2
3	16.9	17.3
4	16.4	16.9
5	18.0	18.3
6	16.4	16.6
7	17.0	16.9
8	18.2	18.5
9	18.6	18.5
10	19.0	19.4

Case Problem 6.7

Effect of Staking on Axle Vibrations

Tasks:

i Formulate the hypothesis: the null (no difference), and the alternate (there is a difference).

ii Set a threshold probability significance level α (maybe 5%).

iii Determine the probability (p-value) that the *null hypothesis* can be true

iv Rule out the null hypothesis if the p-value $\leq \alpha$ and accept the *alternative hypothesis* ("If P is low, H-oh must go!")

Think carefully about which test you will use for this problem.

CP 6.7 solution

i $H_0 : \mu_{before} = \mu_{after}$, $H_a : \mu_{before} \neq \mu_{after}$

ii Used a two tailed (because we are looking for differences) at $\alpha = 0.05$

iii For two tailed test we have 2.5% in each tail. Therefore we use: `print ss.t.ppf(q=0.025, df=9)`. Which gives a critical t-value of: `-2.262`

iv Running the paired difference test: `print ss.ttest_rel(staked, notstaked)`, gives:
`Ttest_relResult(statistic=-2.6410973913470062, pvalue=0.026865404794983259)`

The t-statistic is -2.641, and the critical value is -2.262. Since the t-statistic is more negative than the critical value, we must reject the null hypothesis.
Alternatively, by looking at the p-value, the computed p-value is 0.027, which is smaller than our significance level of $\alpha = 0.05$. Therefore we reject the H_0 and conclude that 'staking' did have an impact on the mean.

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Case problem 6.8

Defects in the manufacture of an IC

Analyse the following data for the number of defects per a chip.

Number of Defects	0	1	2	3	4
Number of Chips	242	94	38	4	2

Test to see whether the samples are drawn from a Poisson distribution.

Recommended procedure:

i Plot the histogram

ii Choose an appropriate distribution

iii Set the null hypothesis that the sample is drawn from the distribution specified in (ii)

iv write an igor function to calculate the modelled/expected distribtuon

v Perform a hypothesis test with $\alpha = 0.05$

vi What do you conclude

CP 6.8 Solution

For x$\geq$ 3 the expected frequency is less than five, therefore combine these into a category 3 or more.

Table: Expected values according to Possion PDF with $\mu = 0.5$

x	**P(x)**	**Freq**
0	0.606531	230.482
1	0.303265	115.241
2	0.0758163	28.8102
$\geq$3	0.014373561	5.4619343

CP 6.8 Solution

```
def computemean(count, number): #computes the mean number of defects
    mysum=0
    for idx, line in enumerate(count):
        mysum=mysum+(count[idx]*number[idx])
        total=sum(Freq)
    return float(mysum)/total

Defects=[0, 1, 2, 3]
Freq=[242, 94, 38, 6]  #combined the last two categories to make more than 5
     occurrences

TotalCount=sum(Freq)
Exp=ss.poisson.pmf(Defects,0.5)*TotalCount  #expected/modelled value

plt.plot(Defects,Exp)
plt.plot(Defects,Freq)

print "The␣chi^2␣statistic␣is", ss.chisquare(Freq,Exp)[0]
print "The␣chi^2␣P␣value␣is", ss.chisquare(Freq,Exp)[1]
```

CP 6.8 Solution

The chi^2 statistic is 7.72104992333
The χ^2 P-value is 0.0521426550568
"If P is low, reject H_0". P is higher than 0.05, therefore there is no statistical difference at the 95% confidence level between the measured and modelled distributions. This conclusion is marginal since the P value is 0.052. Therefore it is advisable to collect more data to gain a stronger confidence in the conclusion.

Case Problem 6.9

The lifetime of an appliance can be described by the exponential distribution:

$$E(x;\lambda) = \lambda e^{-\lambda x} \quad \text{if } x > 0$$
$$= 0 \quad \textit{otherwise}$$

16 appliances have been tested and the lifetimes in hours are given below. Numerically estimate the most likely value of the parameter λ.

2100	2412	2738	2107
2435	2985	2128	2438
2996	2138	2456	3369
2167	2596	2374	2692

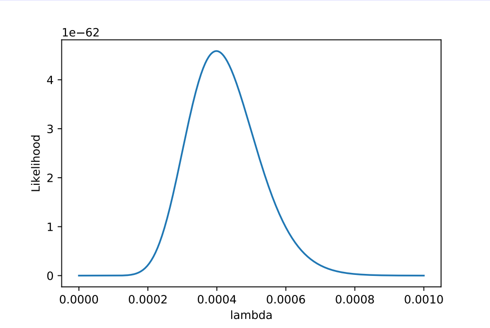
CP 6.9 Solution

```
#week 6 cp 4
def lifetime(lam, mean):   #the exponensial
    return lam*np.exp(-lam*mean)

def likelihood(trials, data):
    meanlife=np.mean(data)
    print meanlife
    mydist=[]
    for line in trials:  #  loop over the different trial lambdas
        prob=1.
        for l in data:   #  Sub-loop tp compute the likelihood for the lifetime
             sample
            prob=prob*lifetime(line, l)
        mydist.append(prob) # make a list of the likelihood for each trial
             lambda
    return mydist

life=[2100, 2412, 2738, 2107, 2435, 2985, 2128, 2438, 2996, 2138, 2456, 3369,
     2167, 2596, 2374, 2692]
lamtrials=np.linspace(0, 1e-3,1001)  # create a list of lambda to try.
likely= likelihood(lamtrials, life)
plt.figure()
plt.plot(lamtrials, likely)
plt.xlabel('lambda')
plt.ylabel('Likelihood')
plt.savefig('w6cp4.pdf')
```

CP 6.9 solution



Thus the mean rate that gives rise to the sample selected is:
$\lambda = 0.4 \times 10^{-3}$

CP 6.9 Solution

For this exponential lifetime problem the likelihood can also be simplified, which might make the program simpler. In my example, I didn't use this simplification.

$$L = \Pi_{i=1}^{n} p(x_i) = (\lambda e^{-\lambda x_1})(\lambda e^{-\lambda x_2})(\lambda e^{-\lambda x_3})\ldots \tag{8}$$

$$L = \lambda^n e^{-\lambda \sum x_i} \tag{9}$$

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN