Assignments Q&A

Please use this document to ask questions about the assignments. I will be monitoring the page answering questions but feel free to answer things yourself if you know the solution!

When asking questions, clearly indicate which assignment questions/sub questions they concern. Feel free to include pictures and diagrams if it helps you explain your point.

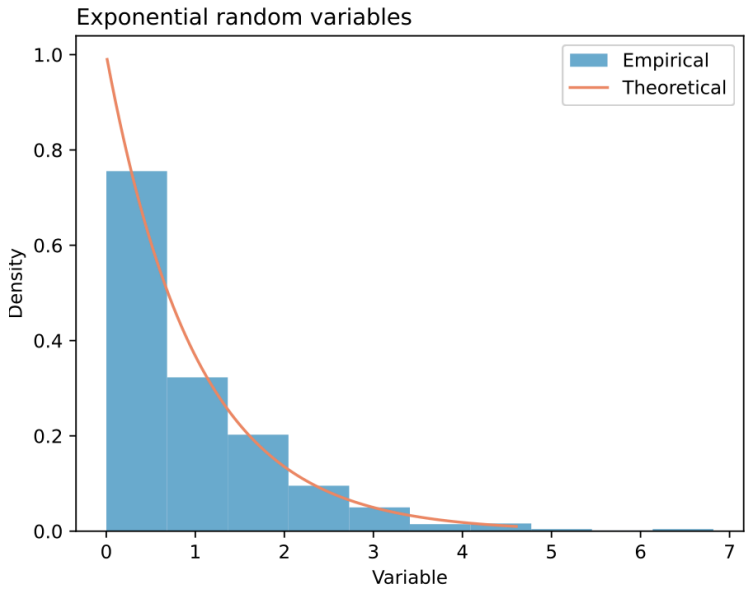
Week 1: Introduction to Python

Question 1: For the scipy/numpy documentation of the exponential function, please explain the theory behind the arguments of this function and why we do not use the expon.pdf to create the start/end points: `x = np.linspace(expon.ppf(0.01), expon.ppf(0.99), 100)`

Answer: The start point and end point arguments are chosen to match those from the empirical data generated in exp_rvs = stats.expon.rvs(scale = 1, size = 1000). The .min() and .max() attributes will select the minimum and maximum values from the generated data. This ensures that the 'Empirical' and 'Theoretical' plots are on the exact same scale.

You alternatively could use the stats.expon.ppf(0.01) and stats.expon.ppf(0.99) function which will give you fixed start and end points at the 1% and 99% cutoff points of the CDF. However, under this method the end points will not be the same as the empirical data for two reasons. Firstly, the data generation will give random start and end points whereas those from the ppf function are fixed. Secondly, we generated 1000 data points in the empirical case, thus the boundaries are highly likely to be significantly wider than stats.expon.ppf(). The discrepancy will be small at the lower boundary but sizable at the upper one. You can see that when running the

alternative code the theoretical line stops well short of the empirical histogram (see below). x = np.linspace(expon.ppf(0.001), expon.ppf(0.999), 100) would give the appropriate start and end points but the randomness point still stands.



Question 2: In this question, "Create another histogram, this time ensure that there is only a single age per column. On top of the histogram draw vertical lines representing the mean age and plus/minus two standard deviations. What do you notice?", one of the first lines of the answer, is the below. Why is 2 added to df.age.min() +2?

```
bin_breaks = np.linspace(start = df.age.min() - 0.5,
stop = df.age.max() + 0.5,
num = df.age.max() - df.age.min() + 2)
```

Answer: First, if the youngest person is 17 (min) and the oldest is 90 (max), the difference is 73 years (90-17 = 73). However, you need to include the end point, since there are actually 74 people. This can be seen very simply for smaller numbers. If you have four people aged 2, 3, 4, 5 the range is 3 but there are in fact 4 people since we need to count the end point.

Second, we extended the start and stop points by 0.5 at either end to ensure that the bin breaks are at decimals. i.e. 16.5-17.5, 17.5-18.5…etc this means that we have certainty that the person who is '18' falls into one. If we didn't do this, the bins would be 17-18, 18-19…etc. It is a bit unclear which the 18-year old falls into. Given we have extended the start and stop points by

0.5 each, we then need to add another 1 to the number of items in our linspace that we are generating.

Putting these two points together we have to add 2 to the number of items generated in the array so that it extends from 16.5 to 90.5 with cutoffs every 1.0.

Question 3: For the circle/rectangle question for this line, are the x and y limits: ax.set_xlim(-1,1) arbitrary (ie. they could be set to (-2,2)) or do they need to be limited to (-1,1) for a particular reason?

Answer: Purely arbitrary. If you adjusted all of the relevant parts then you could absolutely just change the limits and get an equivalent estimation. You would have to make sure to adjust the square, mask and circle parts carefully.

Week 2: Probability

If asking a question about the Python notebooks, please clearly specify which notebook and question you are referring to.

Question 1: The Monte Carlo Approach question

What does the '_' mean in the line below (eg. instead of using 'i', for example)?

counts = np.array([random_rolls_until_six() for _ in range(1000)])

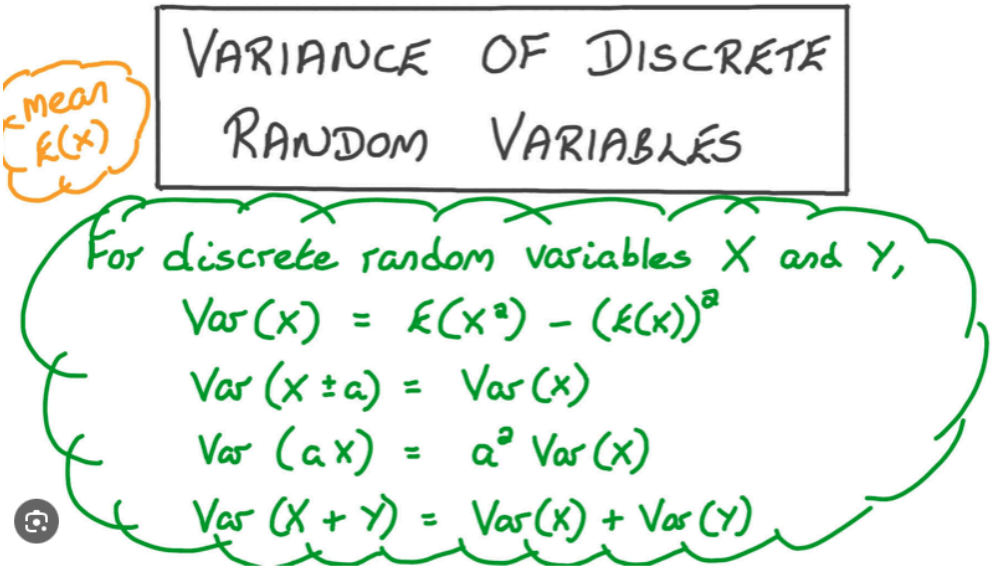
Answer: '_' is just an alternative to 'i'. If you're writing really clean code then it's best to give it a proper name. For example 'row' in 'rows' or 'roll' in 'range()'. But if you're just being quick then it's common to use 'i' or '_'.

Week 2: Problem Sheet 1 (Probability, CLT)

Probability problem set questions 3+4: Which formula should we be using to calculate the standard deviation? The methods for calculating standard deviation in questions three and four are different, so I was wondering if you could briefly go over each solution and the use cases for

each formula? Is question 4 different because we are calculating the standard deviation of an average?

- First is just the standard deviation formula.
- Correct, second is looking at the standard deviation of the **average** rather than the standard deviation of the second data. Remember our rules for variance when we have independence. Var(X+Y) = Var(X) + Var(Y) assuming independence.



Question: Please can you explain the solution for Q7 in the problem set.

Answer: Consider the probability of all outcomes (0, 1, 2,..., 10). Which is the most likely to occur? In this case the probability of 0 is highest. The fact that it is over 0.5 means that every other probability MUST be below 0.5 so we actually know it is 0 just from calculating the probability of 0.

Question: Can you explain how P(A) =⅓ was calculated in 12. b) in the problem set and the logic/property behind 12.c)

Answer: There is an error in the question - will send a message round. It should be over the same timeframe. Good spot.

If you have independence, the distribution of the difference between two normal variables is also normal. Really nice property of normal distributions. Define a new RV as the difference in

the distributions then work out the distribution (normal), mean and variance of this new RV. Then it is just a case of doing standard calculations.

Question: For 10.b., shouldn't the 0.87535 be subtracted from 1 to get the final answer? $1-[P(x=0)-P(x=1)-P(x=2))=1-0.87535$?

Answer: No, because the answer to $1-[P(x=0)+P(x=1)+P(x=2)]$ is 0.87535. The answers aren't great at showing their working but remind yourself how to calculate the prob of the poisson. See the probability mass function (PMF) https://en.wikipedia.org/wiki/Poisson_distribution. Also intuitively, if the mean is 5 then we would expect an above 0.5 probability of getting 3 or more -> good sense check of an answer. The solutions here are very vague though so it's not particularly helpful.

Question: Why do you multiply by 2 in question 12.c.i? Why are the z scores different in ii?

Question for 9: I had to look up the derivation of the mean of the poisson distribution. Can you explain the part with the taylor series expansion?

Answer: Apologies for the delay! Yes, the answers are not particularly informative on this one.

(1) Start with the fact that the mean is the expected value E[X], and we know the formula for expected value in discrete distributions is the sum of (x*P(X=x)) for all possible values (in this case 0 to infinite).
(2) Substitute in the probability mass function (PMF) for P(X=x). You can think of this as the definition of the distribution. Perhaps this should be given in the question (!)
(3) a. Factor out the fixed terms and one of the lambda as in the solutions, and b. Realise that when x=0, the expected value is 0, so we can just start the sum at 1.
(4) We're then going to define j as i-1 to make things a bit simpler, and we now realise that this is the expansion of e^lambda!
(5) Terms cancel.

The Taylor Series expansion of e^x is one of those results which you would only spot if you knew the result. You wouldn't be able to reason about this without knowing the result in advance. But you can derive that it is this infinite sum relatively easily using the Taylor Series formula. I'll leave the proof of the Taylor Series to further reading!

Question: For Q12 (a), why is it assumed that the time limit criterion encompasses the subset of the editing required set? In other words, why can't we calculate the editing-required speeches separately and then add them to those that defy the time limit?

Question: Could you explain Question 12D in a bit more depth?

Question: Why is the z distribution and not the t distribution used in 14a? (I'm assuming constant standard deviation doesn't mean sample sd = population sd)

Question: For question 11, why do we use e^{-T/2} > 0.9 instead of 0.1 > e^{-T/2}?

Question: For question 13b: shouldn't it be 999 degrees of freedom? And is it okay to calculate the exact t-value with a calculator like TI-84?

Week 3: Regression

Code Assignment

Question: For this formula, shouldn't the denominator include squared values?

```
corr_coef = np.sum((x - x_bar) * (y - y_bar)) / (np.sqrt(np.sum((x - x_bar) * (x - x_bar))) * np.sqrt(np.sum((y - y_bar) * (y - y_bar))))
```

Yes, (x - x_bar) * (x - x_bar) means (x - x_bar)^2 so it should work.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Where,

r = Pearson Correlation Coefficient

x_i = x variable samples y_i = y variable sample

$\bar{x}$ = mean of values in x variable $\bar{y}$ =mean of values in y variable

Question: How is the standard error term while calculating the confidence intervals equal to these formulas? (Unsure if these formulas are in our notes, just wanted to confirm when/how they can be used instead of the longer formula in our notes)
np.sqrt(var_alpha_hat)/np.sqrt(var_beta_hat)?

- Could you provide a bit more info about the question and which formula are in your notes vs the notebook.
- Could be because it is the t-distribution vs the normal dist?
- Could be because we have used sub-parts in the python notebook like s_x and s_xx

Question: In Q6, why is n=392 (equal to the number of rows/observations) but in Q7 onwards, n=5 (the number of draws)?

Answer:

- Suboptimal notation.
- In general ‘n’ is used far more that it should be
- Agree that this isn’t the best but we just implicitly defined n differently in both questions.

$$\text{Average S.D.} = \sqrt{(s_1^2 + s_2^2 + \ldots + s_k^2) / k}$$

Question: Is this the formula we should be using for Q4: -It isn't producing the correct response but I'm not sure what the alternative is- this is answered above, but I'm still not sure why this formula isn't right

This formula is not correct for our use. **We're not looking for the average standard deviation but the standard deviation of the average! A subtle difference.** Start from first principles: First define the average ½(x+y), then take the variance of the average Var(½(x+y)). Use the variance rules above to manipulate it and then square root it to get the SD. You can show that it is different from your formula above.

Question: I really don't understand the derivation in Q5

The proof of the derivation is not given in the solutions. There will be multiple examples of it online. The mean is relatively intuitive but the variance is a little harder. Don't worry too much about this proof.

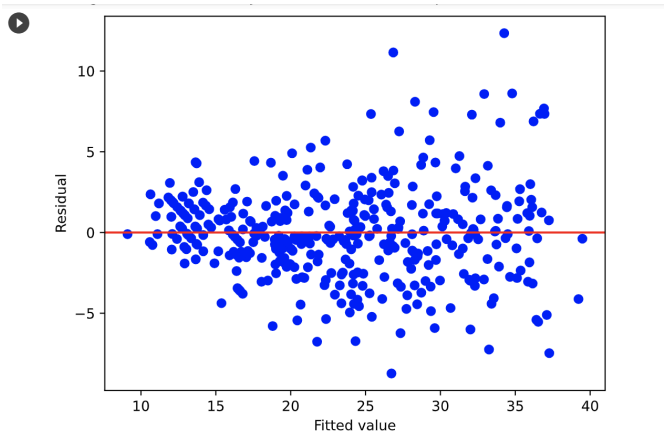
Week 4: Regression II

Question 7 of notebook:

- Python's output says "The condition number is large, 5.2e+03. This might indicate that there are strong multicollinearity".
    - **What is a condition number and what does it tell us?**

Answer: Have a look at the wikipedia for multicollinearity and for condition number. Condition number is a measure of the sensitivity of the regression results to changes in the data. Once researched then add a brief paragraph below for others:

- I'm confused about the transformations that were added to the model. In my notebook, I ended up removing ln_displacement because the coefficient for the transformation was not significant (and it did not improve the coefficient or significance for the displacement variable). **Ln_displacement also is not significant in the solutions, so is there a reason for keeping it in the model?** (note: I just saw question 8 lol, but I would still like some clarification about why ln_displacement is not necessary…is it just because of significance and why does it impact adjusted R^2?)
    - We also discussed in the tutorial that the untransformed variables should usually be kept in the model...did I understand that correctly? Is there a reason that the model in the solutions only includes the transformed variables? My final model kept all of the untransformed variables and only used transformations for horsepower and weight; is that also a valid approach? These are my residuals:



Answer: Log transformations can be done either to capture non-linear relationships or to help with skewed data. See section 2 of this https://kenbenoit.net/assets/courses/me104/logmodels2.pdf. You could keep the original term in but it would probably be overfitting to the data (in the same way that you could add lots of higher order terms to capture all of the intricacies of your data). This is actually quite an interesting question so do a bit of further research about when to include or exclude the linear

term. Another potential reason for dropping it is that it would complicate the interpretation of the coefficients a lot (not too difficult with higher order terms).

Watch the Ben Lambert video on log transformations for explanations about why they are done in the first place.

Question 9 of notebook:

- Could you give a little more context to the explanation in the solutions? The regression results don't say anything about America. How do we know that Eurpoe and Japan are improving in less time?

Answer: Good question - when you have a categorical variable with k variables then you include k-1 dummies in the regression. I.e. is it Japan (yes or no) and is it Europe (yes or no). The reason why we need only to include k-1 dummies is that if both are 'no' then the remainder is America. This means that the baseline here is America. So given that both of the coefficients on the interaction between origin and year are positive, they should be interpreted as positive **relative** to America. We therefore know that both are growing at a faster rate. Have a further look into categorical variables and dummy variables.

Week 4: Problem Sheet 2 (Regression)

Question: In Question 14 A, why does solving for μ (using the equation $z = \frac{x-\mu}{\frac{\sigma}{\sqrt{n}}}$) give us the minimized average as opposed to simply an estimate for the population average?

Answer: Since z is set as the smallest possible test statistic that we can see to accept the same then the associated mu is also the minimum. It all comes from selecting a z which is the most extreme value.

Question: Can you explain tails for t and z-scores? When would we use a 1 tailed approach or a 2 tailed approach?

Answer: See Ben Lambert video on one and two tailed hypothesis testing. Essentially it just depends on our hypothesis but there isn't a fundamental difference. If our hypothesis is the true mean is 'not equals' to the null hypothesis then we split the significance over two sides of the distribution (since this can be rejected by an extremely small and extremely large value). If our hypothesis is that the true mean is just less than the null hypothesis then all of the significance level is on one side of the distribution. Lots of resources online about this.

Question: In question 14 C, why is $\sqrt{2.5}$ in the equation $P(X >= 96.32) = 1 - \Phi((96.32-99.5)\sqrt{2.5})$? If the variance/standard deviation does not change, shouldn't we be dividing by $\sqrt{5}$?

Answer: Recall that the variance of the distribution of the mean is (s.d.^2)/n so in both cases it is (5^2)/10 i.e. 2.5. Recap resources can be found on the central limit theorem and t-distribution.

Question: In question 14 D, would increasing the significance level be a valid way to reduce the chance of a Type I error? I just figured (from looking at the equation for part B), that increasing alpha would mean subtracting a larger value from 1, resulting in a lower probability of falsely rejecting the null hypothesis.

Answer: The probability of a type I error is equal to the significance level. It is the probability of falsely rejecting the null hypothesis i.e. the probability that you get sufficiently extreme results by chance. See the Ben Lambert video on type I and II type errors.

Week 5: Logistic Regression I

Question: Hi Harry, for the week 5 Python notebook, could I just ask what your code is doing in Question 9 [Extension]? How does it determine the false positives? Thanks!

Answer: I've updated the answers on Canvas and the GitHub. The new code should be a bit better commented. Essentially, when running a test many times in a simulation, you would expect to see a false positive rate equal to the significance level (since this is the very definition of confidence level). However, when using small cell sizes you can show that the actual false positive rate is not equal to the significance level.

Week 6: Logistic Regression II

Week 7: Multilevel Modelling I

Question: In a regression model (I'm thinking OLS from the problem set, but this would be helpful to know in general), in what contexts do we consider X and Y as random variables? I have seen the OLS model written as both y_i = alpha + beta*x_i + error_i and Y_i = alpha + beta*X_i + error_i. I was thinking that y_i and x_i refer to individual values from the random samples X and Y, but it would be great if you could give some clarification/examples. (I was reading about the proofs for problem 6 online and they work under the assumption that X and Y are random variables, so I'm just trying to understand that).

Answer: Read about the Gauss Markov assumptions (I might go into them a little on Friday as well). There are two sets of assumptions, one of which treats the regressors as fixed and one as random. The only difference is the method by which we prove the results - assuming the regressors are random requires the law of iterated expectations (LIE) to be required. All of the proofs you'll see online will use the random variable assumption.

_i denotes the individual i. Generally, small letters (x and y) refer to a sample and big letters (X and Y) refer to the population. But this might not always be the case online. You will also see **X** refer to the vector of regressors.

See the first two paragraphs here: 4.1Assumptions

Question: For question 5: where do we take the formula used in the answer. The formulas on the internet for odds ratio don't produce the same results and I can't find the formula used in the codebook.

Answer: The logg odds formula? In binary logistic regression the log odds are just a linear model. So we just use the coefficients from the fitted regression. You should have this formula in your lecture notes. The coefficient estimates are from the regression we fitted a few lines earlier.

Question: Would you be able to explain your working on question 6?

```
[ ]  # At the change point the log odds will equal 0.
     # Sub in the coefficients and num_people = 2 to the log odds equation
     # Rearrange this for the income value.
     income_change_point = (2.23 - 0.189 * 2) / 0.0566

     # Generate an array of income values that are linearly spaced from 10 to 70
     income_vals = np.linspace(10, 70, 100)

     # Extract all of the coefficients and results from the fitted model
     theta_int = logistic_model.params.Intercept
     theta_income = logistic_model.params.income
     theta_num = logistic_model.params.num_people

     # Generate the probabilities of choosing car for each income using the standard probability function for logistic regression
     # See the original formula for inv_logit to make sense of this but it is just a sigmoid
     prob_car = inv_logit(theta_int + theta_income * income_vals + theta_num * 2)

     # Plot the graph of probailities against income and draw on the line where prob = 0.5
     plt.figure()
     plt.plot(income_vals, prob_car)
     plt.axhline(0.5, color='r')
     plt.axvline(income_change_point, color='r')
     plt.annotate("{x:.2f}".format(x=income_change_point), (income_change_point + 1, 0.2), color='r')
     plt.show()
```

Answer: Key point in at the change point the log odds will be 0. Use this to solve for the income at that point. The plot it all up.

Week 8: Multilevel Modelling II