Week 8 - Assessed Exercises

Data Programming with Python

It is often the case that we wish to decide which distribution is the best fit to a single variable. For example, we might want to see whether the residuals of a linear regression are approximately normally distributed. QQ-plots are one of the best ways to do this. They are often superior to drawing histograms as it's easier to assess whether the tails of the distribution fit.

In this assessed exercise we're going to create some QQ-plots. The steps to create a qqplot to compare a chosen probability distribution with a single variable x are:

1. Calculate the empirical cdf (ecdf) of x
2. Simulate a large number of observations from the chosen probability distribution
3. Find the quantiles of the distribution at the probabilities defined by the ecdf

If the two data sets match, a plot of the quantiles and the original data should fall on a straight line. For more detail, see e.g. `http://onlinestatbook.com/2/advanced_graphs/q-q_plots.html`

In this exercises we will use four data sets which come from four unknown probability distributions. One of them comes from a $N(0,1)$ distribution, another a t_5 distribution another a $Exp(1)$ distribution, and finally a *Chi-squared*(1) distribution. The files are labelled `qq1` to `qq4.txt` and are all of different lengths. We're going to use QQ-plots to find which data set matches to which probability distribution.
Include the import statements for all packages used within your code. Additionally, please include the package prefixes (`pd`, `np`, etc.) for functions/methods from these packages, even if the command runs in Canopy without the prefix. . You must submit the correct code, text answers and figures, and successfully complete the Brightspace quiz to receive full marks

1. For the first part of the task, we need to create the empirical cumulative distribution function (ecdf). This is defined as: Number of observations z less than or equal to z_i / Number of observations, for every z_i in z, i.e. $P(z < z_i)$, for $i = 1, \ldots, n$. Write a function called which takes a set of observations z and produces the empirical cdf If you are unfamiliar with empirical cdfs, you may want to read the following article: `https://towardsdatascience.com/what-why-and-how-to-read-empirical-cdf-123e2b922480`
   Plot each of the variables `qq1`, `qq2`, etc. versus their ecdf, as subplots in a single figure window. Save your figure and include it in your submission.

2. For the next part we need to create the quantiles of a chosen probability distribution Write a function which takes an ecdf created by your function in Q2 and simulates 10,000 observations from a normal(0,1) distribution. Then calculate the quantiles of these simulations at the probabilities defined by the ecdf.
   Create a scatter plot of the theoretical quantiles from your new function (x-axis) against `qq1` (y-axis). Repeat this for each dataset, creating each plot as a subplot on the same figure.

Save your figure and include it in your submission. If the two distributions match, the points should lie on a straight line - this is a QQ-plot. Which of the datasets is normally distributed variable?

3. Create a new function that takes two arguments. The first argument should be your data Series and the second argument should be a set of simulations from some probability distribution. It should use these samples to calculate the theoretical quantiles. This function should the computed theoretical quantiles

4. Run your function for each of the datasets, with

   - `d = Series(npr.randn(10000))` (normal distribution)
   - `d = Series(npr.exponential(1,10000))` (exponential distribution)
   - `d = Series(npr.standard_t(5,10000))` (t_5 distribution)
   - `d = Series(npr.chisquare(1,10000))` (chi-squared distribution)

   Plot empirical data versus the theoretical quantiles returned by `exercise3` to determine which data set matches to which probability distribution. Complete the quiz 'W8 - Assessed exercises Q4' to submit your answer for this question.

All of your code should be written into the *.py* template. Save your filled *.py* file with the following name structure *SurnameFirstname_Week7.py* (where *Surname* and *Firstname* should be replaced with your name) and upload it to Brightspace. You must also upload a PDF of your code and 2 figures (one for Q1 and one for Q2), and complete the Brightspace quiz associated with Q4.