Programming for Data Analysis Assignment 2018:

For this project you must create a data set by simulating a real-world phenomenon of your choosing. You may pick any phenomenon you wish – you might pick one that is of interest to you in your personal or professional life. Then, rather than collect data related to the phenomenon, you should model and synthesise such data using Python. We suggest you use the numpy.random package for this purpose

Choose a real-world phenomenon that can be measured and for which you could collect at least one-hundred data points across at least four different variables.

- Investigate the types of variables involved, their likely distributions, and their relationships with each other.
- Synthesise/simulate a data set as closely matching their properties as possible.
- Detail your research and implement the simulation in a Jupyter notebook – the data set itself can simply be displayed in an output cell within the notebook.

Note that this project is about simulation – you must synthesise a data set. Some students may already have some real-world data sets in their own files. It is okay to base your synthesised data set on these should you wish (please reference it if you do), but the main task in this project is to create a synthesised data set. The next section gives an example project idea. Page 1 of 4

Example project idea As a lecturer I might pick the real-world phenomenon of the performance of students studying a ten-credit module. After some research, I decide that the most interesting variable related to this is the mark a student receives in the module - this is going to be one of my variables (grade).

Upon investigation of the problem, I find that the number of hours on average a student studies per week (hours), the number of times they log onto Moodle in the first three weeks of term (logins), and their previous level of degree qualification (qual) are closely related to grade . The hours and grade variables will be non-negative real number with two decimal places, logins will be a non-zero integer and qual will be a categorical variable with four possible values: none , bachelors, masters, or phd.

After some online research, I find that full-time post-graduate students study on av- erage four hours per week with a standard deviation of a quarter of an hour and that a normal distribution is an acceptable model of such a variable. Likewise, I investigate the other four variables, and I also look at the relationships between the variables. I devise an algorithm (or method) to generate such a data set, simulating values of the four variables for two-hundred students. I detail all this work in my notebook, and then I add some code in to generate a data set with those properties.

To be able to model and therefore predict the distribution of grades for graduates in a given year based on:

Number of expected graduates:

LC points range at point of entry College/Sector Type of School (DEIS/Fee-paying) Field of Study (Broad) Gender

Entry-route: Lc/non-LC.

Restricted to Honours degree Graduates only

Potential real-life applications

Estimating the work readiness of graduates and the flow into employment (2:1) and post-graduate study (1st)

For non-completion and low attainment (lower that 2:1, bare pass or non-completion), the provision of adequate post-entry supports; revision of entry requirements (on a course by course basis).

Limits of the study: for a like for like comparison, Level 8 honours degree students only. L6/L7 would require a seperate study.

This does not taken into account students cultual or social economic background, or if the student had a disability (includ. It is suggested that a more appropriate approach would be to do a macro study of all students and a seperate study of students from minority or under-represented groups and compare paterns of entry, progression/completion and attainment against this of the various sub-groups and if there are any deviations, disgnose what interventions need to be made on their behalf (if any)

Initial research:

(1) SRS Query - Graduates

Student id Academic Year Gender Long Non Standard Award Desc Institute Type Institute Isced Broad Desc Grade Desc

(2) SRS Query - Entrants

Student id Number of Students Academic Year = 2012/13 and 2014/15 Gender Long LC Points Range 1 LC Points Range 2 DEIS_OR_FEE_PAYING_DESC High Qual Desc Age Group Institute Institute Type Progtype Desc Isced Broad Desc

(3) in Excel, crossrefrenced

The aim of this project will be to investigate a dataset of students entering in the years 2012 and 2013 and graduating in 2016 or 2017 (double check correct years). Specifically, for Honours degrees, I want to investagate the distribution of final grades and see if there is a link between final grade achived and other factors, for example, gender, previous school attendedm, age on entry and LC points. The idea is to see if the dgrade distributions as a whol or for sunbgroups of entrants can be modelled by one of the more commonly used distributions used in the social sciences (e.g. pareto, do only 20% of students get a first, etc).

The second part of the exercise will attempt to model the future distribution of grades for 2018 or 2019, based on an estimate of students predicted to graduate in these years.

learnings

Pandas and excel

more about distributions.

Please note that as of 06/12/2018.

In [41]:
```
import numpy as np
import pandas as pd
import seaborn as sns
```

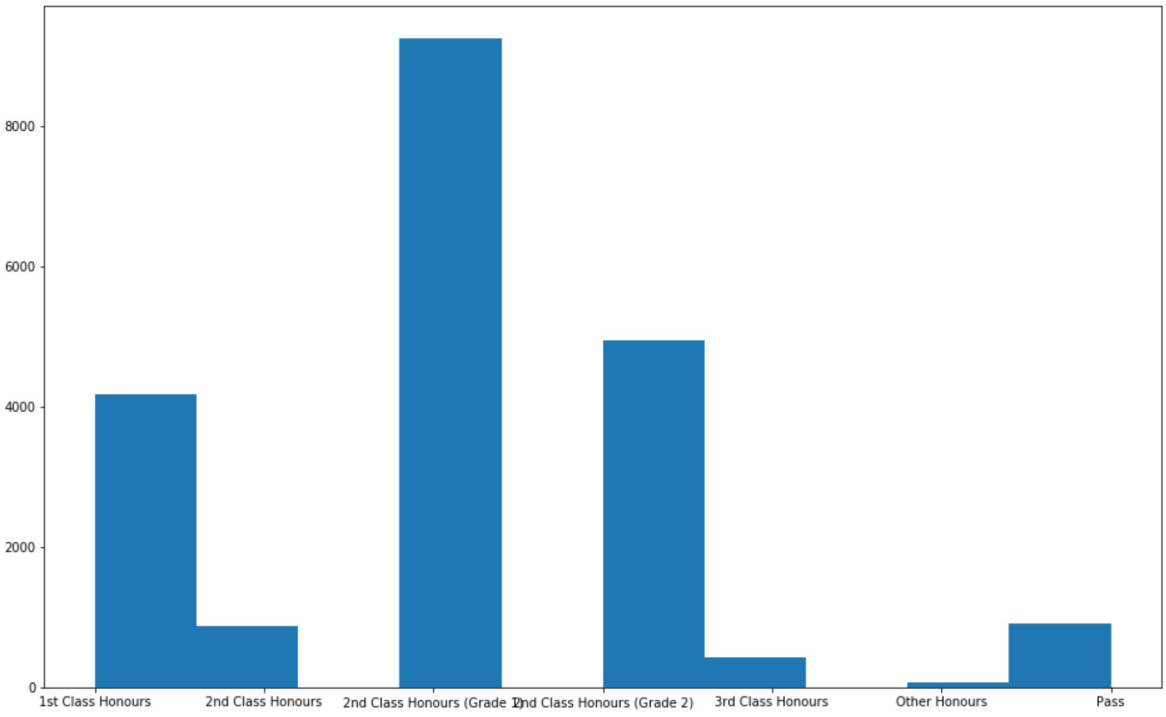
In [117]:
```
# Import as data_frame df, the dataset in the Graduates worksheet of the 'Graduates
# As per default setting of the read_excel method, the first line will be


df = pd.read_excel('Graduates2017 - Excel.xlsx', 'Graduates')
```

Out[117]:

	ALT Student id	Year of Entry	Gender Long	LC Points Range 1	LC Points Range 2	DEIS_OR_FEE_PAYING_DESC	High Qual Desc	Age Group	Institute	In
0	AD-12360856	2012/2013	Female	305 to <355	medium points	Neither	NaN	19	National College of Art and Design	C
1	AD-12375986	2012/2013	Female	305 to <355	medium points	Neither	NaN	18	National College of Art and Design	C
2	AD-12420552	2012/2013	Female	405 to <455	high points	Neither	NaN	19	National College of Art and Design	C
3	AD-12444362	2012/2013	Male	455 to <505	high points	Neither	NaN	19	National College of Art and Design	C
4	AD-12500127	2012/2013	Female	205 to <255	medium points	Neither	NaN	22	National College of Art and Design	C

In [102]:
```
plt.figure(figsize=(16, 10))
plt.hist(df['Grade Desc'])
```



In [29]:

```
Out[29]: ALT Student id                20643
         Year of Entry                 20643
         Gender Long                   20643
         LC Points Range 1             17630
         LC Points Range 2             13802
         DEIS_OR_FEE_PAYING_DESC       20643
         High Qual Desc                15292
         Age Group                     20643
         Institute                     20643
         Institute Type                20643
         Progtype Desc                 20643
         Year of Graduation            20643
         Non Standard Award Desc       20643
         Field of Study                20643
         Grade Desc                    20643
         Number of years               20643
         Institute Alternative Name    20643
         dtype: int64
```

In [50]:

```
Out[50]: 2nd Class Honours (Grade 1)    9243
         2nd Class Honours (Grade 2)    4940
         1st Class Honours              4170
         Pass                            914
         2nd Class Honours               877
         3rd Class Honours               429
         Other Honours                    70
         Name: Grade Desc, dtype: int64
```

In [118]:

Out[118]:

					ALT Student id	Year of Entry	Gender Long	LC Points Range 1	DEIS_OR_FEE_PA
Institute Type	Institute	Field of Study	LC Points Range 2	Grade Desc					
Colleges	Mary Immaculate College, Limerick	Arts and humanities	high points	1st Class Honours	5	5	5	5	
				2nd Class Honours (Grade 1)	37	37	37	37	
				2nd Class Honours	33	33	33	33	

In [119]:

```
Out[119]: 20527
```

In [120]:
```
ids = df["ALT Student id"]
df[ids.isin(ids[ids.duplicated()])].count()
```

Out[120]:
```
ALT Student id                232
Year of Entry                 232
Gender Long                   232
LC Points Range 1             157
LC Points Range 2             133
DEIS_OR_FEE_PAYING_DESC       232
High Qual Desc                129
Age Group                     232
Institute                     232
Institute Type                232
Progtype Desc                 232
Year of Graduation            232
Non Standard Award Desc       232
Field of Study                232
Grade Desc                    232
Number of years               232
Institute Alternative Name    232
dtype: int64
```

In [121]:
```
# Remove duplicate
# sort dataset by alt id, then by  year
df.sort_values(['ALT Student id', 'Year of Entry'])
# make new dataframe with duplicate entries for the same students removed, keeping
df = df.drop_duplicates(subset='ALT Student id', keep='last', inplace=False)


df['ALT Student id'].count()
# Drop following columns from dataset:
# Institute - we will use anonymised institute instead
# Progtype Desc (i.e the programme they entered in, which may be different to an ho
# but with the removal of the duplicate IDs, no longer considered relevant)
# Alt Student ID - No longer relevant since duplicates removed - we are satisfied t
df = df.drop(columns=['ALT Student id', 'Institute', 'Progtype Desc'])
df.head()
```

Out[121]:

	Year of Entry	Gender Long	LC Points Range 1	LC Points Range 2	DEIS_OR_FEE_PAYING_DESC	High Qual Desc	Age Group	Institute Type	Year of Graduation	Non A
0	2012/2013	Female	305 to <355	medium points	Neither	NaN	19	Colleges	2017	Unc
1	2012/2013	Female	305 to <355	medium points	Neither	NaN	18	Colleges	2017	Unc
2	2012/2013	Female	405 to <455	high points	Neither	NaN	19	Colleges	2017	Unc
3	2012/2013	Male	455 to <505	high points	Neither	NaN	19	Colleges	2017	Unc
4	2012/2013	Female	205 to <255	medium points	Neither	NaN	22	Colleges	2017	Unc

In [125]:
```
df = df[df['Grade Desc'] != 'Other Honours'] # remove "other honours" - unclear hov
df = df[df['Grade Desc'] != '2nd Class Honours'] # remove undifferentiated second c
df.count()
```

Out[125]:
```
Year of Entry                 19581
Gender Long                   19581
LC Points Range 1             17265
LC Points Range 2             13557
DEIS_OR_FEE_PAYING_DESC       19581
High Qual Desc                14514
Age Group                     19581
Institute Type                19581
Year of Graduation            19581
Non Standard Award Desc       19581
Field of Study                19581
Grade Desc                    19581
Number of years               19581
Institute Alternative Name    19581
dtype: int64
```

In [126]:
```
# Replacing grades with average or median mark

df = df.replace('1st Class Honours',85)
df = df.replace('1st Class Honours',85)
df = df.replace('2nd Class Honours (Grade 1)',64.5)
df = df.replace('2nd Class Honours (Grade 2)',54.5)
df = df.replace('3rd Class Honours', 44.5)
df = df.replace('Pass', 44.5)

# is there a more efficient way of doing this in one command
```

Out[126]:

	Year of Entry	Gender Long	LC Points Range 1	LC Points Range 2	DEIS_OR_FEE_PAYING_DESC	High Qual Desc	Age Group	Institute Type	Grad
0	2012/2013	Female	305 to <355	medium points	Neither	NaN	19	Colleges	
1	2012/2013	Female	305 to <355	medium points	Neither	NaN	18	Colleges	
2	2012/2013	Female	405 to <455	high points	Neither	NaN	19	Colleges	
3	2012/2013	Male	455 to <505	high points	Neither	NaN	19	Colleges	
4	2012/2013	Female	205 to <255	medium points	Neither	NaN	22	Colleges	

In [127]:
```
# Save dataframe as new CSV file - this will become the working file for the next p
```

In [ ]: