* Open colab
* Add two files to colab
* udemy\_courses.csv
* edx\_courses.csv

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import sqlite3

from pathlib import Path as Path

edx\_course\_descriptions = pd.read\_csv('edx\_courses.csv')

edx\_course\_descriptions.tail(3)

edx\_user\_data = pd.read\_csv('udemy\_courses.csv')

edx\_user\_data.head(3)

edx\_course\_descriptions.dtypes

edx\_course\_descriptions['n\_enrolled'] = edx\_course\_descriptions['n\_enrolled'].str.replace(',','')

edx\_course\_descriptions['price'] = edx\_course\_descriptions['price'].str.replace('FREE-Add a Verified Certificate for', '')

edx\_course\_descriptions['price'] = edx\_course\_descriptions['price'].str.replace('$', '')

edx\_course\_descriptions['price'] = edx\_course\_descriptions['price'].str.replace('USD', '')

edx\_course\_descriptions['course\_length'] = edx\_course\_descriptions['course\_length'].str.replace('Weeks', '')

edx\_course\_descriptions['n\_enrolled'] = edx\_course\_descriptions['n\_enrolled'].fillna(0)

edx\_course\_descriptions['n\_enrolled'] = edx\_course\_descriptions['n\_enrolled'].astype(int)

edx\_course\_descriptions['price'] = edx\_course\_descriptions['price'].astype(float)

edx\_course\_descriptions['course\_length'] = edx\_course\_descriptions['course\_length'].astype(int)

edx\_course\_descriptions.isna().sum()

edx\_course\_descriptions['summary'] = edx\_course\_descriptions['summary'].fillna('Unlisted')

edx\_course\_descriptions['instructors'] = edx\_course\_descriptions['instructors'].fillna('Unlisted')

edx\_course\_descriptions['subtitles'] = edx\_course\_descriptions['subtitles'].fillna('English')

edx\_course\_descriptions['course\_description'] = edx\_course\_descriptions['course\_description'].fillna('Unlisted')

edx\_course\_descriptions['course\_syllabus'] = edx\_course\_descriptions['course\_syllabus'].fillna('Unlisted')

edx\_course\_descriptions.tail(3)

del edx\_course\_descriptions['course\_url']

edx\_course\_descriptions.tail(3)

edx\_course\_descriptions['Level'] = edx\_course\_descriptions['Level'].astype('category')

edx\_course\_descriptions['Level'] = edx\_course\_descriptions['Level'].cat.codes

edx\_course\_descriptions['course\_type'] = edx\_course\_descriptions['course\_type'].astype('category')

edx\_course\_descriptions['course\_type'] = edx\_course\_descriptions['course\_type'].cat.codes

edx\_course\_descriptions.dtypes

edx\_user\_data['Launch Date'] = pd.to\_datetime(edx\_user\_data['Launch Date'])

edx\_course\_descriptions.describe()

edx\_course\_descriptions['subject'].value\_counts()

edx\_course\_description\_csv = edx\_course\_descriptions.to\_csv("edx\_course\_description\_for\_bq.csv")

Path('edx\_course\_descriptions.db').touch()

edx\_conn = sqlite3.connect('edx\_course\_descriptions.db')

edx\_cursor = edx\_conn.cursor()

edx\_course\_descriptions.to\_sql('edx\_course\_descriptions', edx\_conn, if\_exists='append', index=False)

edx\_description\_init\_query = pd.read\_sql(''' SELECT \* FROM edx\_course\_descriptions ''', edx\_conn)

edx\_description\_init\_query

data\_analysis\_query = pd.read\_sql(''' SELECT title, price, course\_effort, n\_enrolled, institution

                                      FROM edx\_course\_descriptions

                                      WHERE subject == 'Data Analysis & Statistics'

                                      ORDER BY price DESC LIMIT 15 ''', edx\_conn)

data\_analysis\_query

sns.barplot(x = 'price', y = 'title', data = data\_analysis\_query, color = 'blue', edgecolor='black')

sns.barplot(x = 'n\_enrolled', y = 'title', data = data\_analysis\_query, color = 'blue', edgecolor='black')

sns**.**barplot(x **=** 'price', y **=** 'n\_enrolled', data **=** data\_analysis\_query, color **=** 'blue', edgecolor **=** 'black', ci**=False**)

data\_analysis\_query.mean()

data\_analysis\_python = pd.read\_sql(''' SELECT title, price, course\_effort, n\_enrolled, institution

                                       FROM edx\_course\_descriptions

                                       WHERE title LIKE '%Python'

                                       ORDER BY price DESC  ''', edx\_conn)

data\_analysis\_python

sns.barplot(x = 'price', y = 'n\_enrolled', data = data\_analysis\_python, color = 'orange', edgecolor='black', ci=False)

data\_analysis\_prog = pd.read\_sql(''' SELECT title, price, course\_effort, n\_enrolled, institution

                                       FROM edx\_course\_descriptions

                                       WHERE title LIKE '%Programming'

                                       ORDER BY price DESC  ''', edx\_conn)

data\_analysis\_prog

data\_analysis\_stat = pd.read\_sql(''' SELECT title, price, course\_effort, n\_enrolled, institution

                                     FROM edx\_course\_descriptions

                                     WHERE title LIKE '%Statistics'

                                     ORDER BY price DESC ''', edx\_conn)

data\_analysis\_stat

data\_analysis\_prob = pd.read\_sql(''' SELECT title, price, course\_effort, n\_enrolled, institution

                                     FROM edx\_course\_descriptions

                                     WHERE title LIKE '%Probability' AND title NOT LIKE '%MathTrackX: Probability'

                                     ORDER BY price DESC ''', edx\_conn)

data\_analysis\_prob

comp\_sci\_query = pd.read\_sql(''' SELECT title, price, course\_effort, n\_enrolled, institution

                                 FROM edx\_course\_descriptions WHERE subject == 'Computer Science'

                                 ORDER BY price DESC LIMIT 15 ''', edx\_conn)

comp\_sci\_query

sns.barplot(x = 'price', y = 'title', data = comp\_sci\_query, color = 'orange', edgecolor='black', ci=False)

sns.barplot(x = 'n\_enrolled', y = 'title', data = comp\_sci\_query, color = 'orange', edgecolor='black', ci=False)

data\_analysis\_tot = pd.read\_sql(''' SELECT SUM(n\_enrolled) AS total\_program\_enrollment, COUNT(title) AS num\_courses, AVG(price) AS avg\_course\_purchase, MIN(price) AS min\_course\_purchase, MAX(price) AS max\_course\_purchase

                                    FROM edx\_course\_descriptions

                                    WHERE subject == 'Data Analysis & Statistics'

                                    ORDER BY price DESC ''', edx\_conn)

data\_analysis\_tot

edx\_tot = pd.read\_sql(''' SELECT SUM(n\_enrolled) AS total\_edx\_enrollment, COUNT(title) AS total\_courses, AVG(price) AS total\_avg

                          FROM edx\_course\_descriptions

                          ORDER BY price DESC ''', edx\_conn)

edx\_tot

business\_tot = pd.read\_sql(''' SELECT SUM(n\_enrolled) AS total\_business\_enrollment, COUNT(title) AS total\_courses, AVG(price) AS avg\_course\_purchase, MIN(price) AS min\_course\_purchase, MAX(price) AS max\_course\_purchase

                               FROM edx\_course\_descriptions

                               WHERE subject == 'Business & Management'

                               ORDER BY price DESC ''', edx\_conn)

business\_tot

engineering\_tot = pd.read\_sql(''' SELECT SUM(n\_enrolled) AS total\_engineering\_enrollment, COUNT(title) AS total\_courses, AVG(price) AS avg\_course\_purchase, MIN(price) AS min\_course\_purchase, MAX(price) AS max\_course\_purchase

                               FROM edx\_course\_descriptions

                               WHERE subject == 'Engineering'

                               ORDER BY price DESC ''', edx\_conn)

engineering\_tot

ed\_tot = pd.read\_sql(''' SELECT SUM(n\_enrolled) AS total\_education\_enrollment, COUNT(title) AS total\_courses, AVG(price) AS avg\_course\_purchase, MIN(price) AS min\_course\_purchase, MAX(price) AS max\_course\_purchase

                               FROM edx\_course\_descriptions

                               WHERE subject == 'Education & Teacher Training'

                               ORDER BY price DESC ''', edx\_conn)

ed\_tot

law\_tot = pd.read\_sql(''' SELECT SUM(n\_enrolled) AS total\_law\_enrollment, COUNT(title) AS total\_courses, AVG(price) AS avg\_course\_purchase, MIN(price) AS min\_course\_purchase, MAX(price) AS max\_course\_purchase

                               FROM edx\_course\_descriptions

                               WHERE subject == 'Law'

                               ORDER BY price DESC ''', edx\_conn)

law\_tot

econ\_tot = pd.read\_sql(''' SELECT SUM(n\_enrolled) AS total\_economics\_enrollment, COUNT(title) AS total\_courses, AVG(price) AS avg\_course\_purchase, MIN(price) AS min\_course\_purchase, MAX(price) AS max\_course\_purchase

                               FROM edx\_course\_descriptions

                               WHERE subject == 'Economics & Finance'

                               ORDER BY price DESC ''', edx\_conn)

econ\_tot

by\_subject = pd.read\_sql(''' SELECT subject, COUNT(subject) AS subject\_count,

                             100 \* COUNT(subject) / (SELECT COUNT(\*) FROM edx\_course\_descriptions) AS subject\_pct

                             FROM edx\_course\_descriptions

                             GROUP BY subject

                             ORDER BY subject\_pct DESC LIMIT 10''', edx\_conn)

by\_subject

by\_subject\_csv = by\_subject.to\_csv('by\_subject\_csv.csv')

sns.barplot(x = 'subject\_pct', y = 'subject', data = by\_subject, color = 'red', edgecolor='black', ci=False)

edx\_by\_price = pd.read\_sql(''' SELECT price, COUNT(price) AS price\_count,

                           100 \* COUNT(price) / (SELECT COUNT(\*) FROM edx\_course\_descriptions) AS price\_pct

                           FROM edx\_course\_descriptions

                           GROUP BY price

                           ORDER BY price\_pct DESC LIMIT 10 ''', edx\_conn)

edx\_by\_price

sns.barplot(x = 'price', y = 'price\_pct', data = edx\_by\_price, color = 'red', edgecolor='black', ci=False)

by\_course\_effort = pd.read\_sql(''' SELECT course\_effort, COUNT(course\_effort) AS num\_hours,

                                   100 \* COUNT(course\_effort) / (SELECT COUNT(\*) FROM edx\_course\_descriptions) AS hours\_pct

                                   FROM edx\_course\_descriptions

                                   GROUP BY course\_effort

                                   ORDER BY hours\_pct DESC LIMIT 10 ''', edx\_conn)

by\_course\_effort

sns.barplot(x = 'hours\_pct', y = 'course\_effort', data = by\_course\_effort, color = 'red', edgecolor='black', ci=False)

by\_institution = pd.read\_sql(''' SELECT institution, COUNT(institution) AS institution\_count,

                                 100 \* COUNT(institution) / (SELECT COUNT(\*) FROM edx\_course\_descriptions) AS institution\_pct

                                 FROM edx\_course\_descriptions

                                 GROUP BY institution

                                 ORDER BY institution\_pct DESC LIMIT 10''', edx\_conn)

by\_institution

sns.barplot(x = 'institution\_pct', y = 'institution', data = by\_institution, color = 'red', edgecolor='black', ci=False)

by\_institution\_dsc = pd.read\_sql(''' SELECT institution, COUNT(institution) AS institution\_count,

                                 100 \* COUNT(institution) / (SELECT COUNT(\*) FROM edx\_course\_descriptions) AS institution\_pct

                                 FROM edx\_course\_descriptions

                                 WHERe subject == 'Data Analysis & Statistics'

                                 GROUP BY institution

                                 ORDER BY institution\_count DESC LIMIT 10''', edx\_conn)

by\_institution\_dsc

sns.barplot(x = 'institution\_count', y = 'institution', data = by\_institution\_dsc, color = 'red', edgecolor='black', ci=False)

by\_institution\_comp\_sci = pd.read\_sql(''' SELECT institution, COUNT(institution) AS institution\_count,

                                 100 \* COUNT(institution) / (SELECT COUNT(\*) FROM edx\_course\_descriptions) AS institution\_pct

                                 FROM edx\_course\_descriptions

                                 WHERe subject == 'Computer Science'

                                 GROUP BY institution

                                 ORDER BY institution\_count DESC LIMIT 10''', edx\_conn)

by\_institution\_comp\_sci

sns.barplot(x = 'institution\_count', y = 'institution', data = by\_institution\_comp\_sci, color = 'red', edgecolor='black', ci=False)

subject\_hundred = pd.read\_sql(''' SELECT subject, AVG(price) as avg\_price\_subject

                                  FROM edx\_course\_descriptions

                                  WHERE price > 100.0

                                  GROUP BY subject

                                  ORDER BY avg\_price\_subject DESC LIMIT 10''', edx\_conn)

subject\_hundred = subject\_hundred.round(2)

subject\_hundred

sns.barplot(x = 'avg\_price\_subject', y = 'subject', data = subject\_hundred, color = 'purple', edgecolor='black', ci=False)

by\_enroll = pd.read\_sql(''' SELECT subject, SUM(n\_enrolled) AS total\_enrollment

                          FROM edx\_course\_descriptions

                          GROUP BY subject

                          ORDER BY total\_enrollment DESC LIMIT 10 ''', edx\_conn)

by\_enroll

sns.barplot(x = 'total\_enrollment', y = 'subject', data = by\_enroll, color = 'purple', edgecolor='black', ci=False)

by\_avg\_enroll = pd.read\_sql(''' SELECT subject, AVG(n\_enrolled) AS avg\_enrollment

                          FROM edx\_course\_descriptions

                          GROUP BY subject

                          ORDER BY avg\_enrollment DESC LIMIT 10 ''', edx\_conn)

by\_avg\_enroll

sns.barplot(x = 'avg\_enrollment', y = 'subject', data = by\_avg\_enroll, color = 'purple', edgecolor='black', ci=False)

Path('edx\_behavior.db').touch()

edx\_behavior\_conn = sqlite3.connect('edx\_behavior.db')

edx\_behavior\_cursor = edx\_conn.cursor()

edx\_user\_data.to\_sql('edx\_behavior', edx\_behavior\_conn, if\_exists='append', index=False)

edx\_user\_data\_init\_query = pd.read\_sql(''' SELECT \* FROM edx\_behavior LIMIT 5 ''', edx\_behavior\_conn)

edx\_user\_data\_init\_query

by\_course\_subject = pd.read\_sql(''' SELECT `subject`, COUNT(`subject`) AS course\_subject\_count,

                                    100 \* COUNT(`subject`) / (SELECT COUNT(\*) FROM edx\_behavior) AS subject\_pct

                                    FROM edx\_behavior

                                    GROUP BY `subject`

                                    ORDER BY subject\_pct DESC ''', edx\_behavior\_conn)

by\_course\_subject

sns.barplot(x = 'subject\_pct', y = 'subject', data = by\_course\_subject, color = 'blue', edgecolor='black', ci=False)

udemy = pd.read\_csv('udemy\_courses.csv')

udemy

del udemy['course\_id']

del udemy['url']

udemy.dtypes

udemy['published\_timestamp'] = pd.to\_datetime(udemy['Launch Date'])

udemy['level'] = udemy['level'].astype('category')

udemy['level'] = udemy['level'].cat.codes

udemy['is\_paid'] = udemy['is\_paid'].astype('category')

udemy['is\_paid'] = udemy['is\_paid'].cat.codes

udemy

Path('udemy\_updated.db').touch()

udemy\_conn = sqlite3.connect('udemy\_updated.db')

udemy\_cursor = udemy\_conn.cursor()

udemy.to\_sql('udemy\_updated', udemy\_conn, if\_exists='append', index=False)

udemy\_init\_query = pd.read\_sql(''' SELECT \* FROM udemy\_updated ''', udemy\_conn)

udemy\_init\_query

udemy\_init\_query['subject'].value\_counts()

by\_subject = pd.read\_sql(''' SELECT subject, COUNT(subject) AS subject\_count,

                             100 \* COUNT(subject) / (SELECT COUNT(\*) FROM udemy\_updated) AS subject\_pct

                             FROM udemy\_updated

                             GROUP BY subject

                             ORDER BY subject\_pct DESC ''', udemy\_conn)

by\_subject

sns.barplot(x = 'subject\_pct', y = 'subject', data = by\_subject, color = 'blue', edgecolor = 'black', ci=False)

by\_price = pd.read\_sql(''' SELECT price, COUNT(price) AS subject\_count,

                             100 \* COUNT(price) / (SELECT COUNT(\*) FROM udemy\_updated) AS price\_pct

                             FROM udemy\_updated

                             GROUP BY price

                             ORDER BY price\_pct DESC LIMIT 10 ''', udemy\_conn)

by\_price

sns.barplot(x = 'price', y = 'price\_pct', data = by\_price, color = 'blue', edgecolor = 'black', ci=False)

by\_price['price'] = by\_price['price'].astype(str)

by\_price.dtypes

udemy\_by\_price\_csv = by\_price.to\_csv('udemy\_by\_price.csv')

udemy\_by\_price\_csv

by\_subs\_subject = pd.read\_sql(''' SELECT course\_title, subject, num\_subscribers FROM udemy\_updated

                          GROUP BY subject

                          ORDER BY num\_subscribers DESC ''', udemy\_conn)

by\_subs\_subject

udemy\_by\_subs = pd.read\_sql(''' SELECT course\_title, subject, num\_subscribers FROM udemy\_updated

                          ORDER BY num\_subscribers DESC LIMIT 10 ''', udemy\_conn)

udemy\_by\_subs

sns.barplot(x = 'num\_subscribers', y = 'course\_title', hue = 'subject', data = udemy\_by\_subs, color = 'blue', edgecolor='black', ci=False)

udemy\_by\_subs\_csv = udemy\_by\_subs.to\_csv('udemy\_by\_subs.csv')

by\_reviews = pd.read\_sql(''' SELECT course\_title, subject, num\_reviews

                             FROM udemy\_updated

                             ORDER BY num\_reviews DESC LIMIT 10''', udemy\_conn)

by\_reviews

sns.barplot(x = 'num\_reviews', y = 'course\_title', data = by\_reviews, color = 'red', edgecolor = 'black', ci=False)

reviews\_lec = pd.read\_sql(''' SELECT course\_title, subject, num\_lectures, num\_reviews

                             FROM udemy\_updated

                             ORDER BY num\_reviews DESC''', udemy\_conn)

reviews\_lec

reviews\_lec.dtypes

sns.scatterplot(x = 'num\_lectures', y = 'num\_reviews', data = reviews\_lec)

reviews\_lec.dtypes

reviews\_duration = pd.read\_sql(''' SELECT course\_title, subject, num\_lectures, num\_reviews, content\_duration, price

                             FROM udemy\_updated

                             ORDER BY num\_reviews DESC''', udemy\_conn)

reviews\_duration

sns.scatterplot(x = 'content\_duration', y = 'num\_reviews', data = reviews\_duration)

reviews\_duration\_csv = reviews\_duration.to\_csv('udemy\_reviews\_duration.csv')

total\_enrollment\_biz\_web = pd.read\_sql(''' SELECT subject, num\_subscribers, SUM(num\_subscribers) AS sum\_sub, AVG(price) AS avg\_course\_price

                                           FROM udemy\_updated

                                           GROUP BY subject

                                           ORDER BY sum\_sub DESC ''', udemy\_conn)

total\_enrollment\_biz\_web

tot\_price = pd.read\_sql(''' SELECT subject, num\_subscribers, SUM(num\_subscribers) AS sum\_sub, AVG(price) AS avg\_course\_price, SUM(price) AS total\_revenue

                                           FROM udemy\_updated

                                           GROUP BY subject

                                           ORDER BY sum\_sub DESC ''', udemy\_conn)

tot\_price

sns.barplot(x = 'subject', y = 'total\_revenue', data = tot\_price, color = 'blue', edgecolor = 'black', ci = False)

plt.xticks(rotation=90)

sns.barplot(x = 'subject', y = 'avg\_course\_price', data = tot\_price, color = 'blue', edgecolor = 'black', ci = False)

plt.xticks(rotation=90)

udemy\_tot\_price = tot\_price.to\_csv('udemy\_avg\_price.csv')