Problem 1 (30 pts)

Suppose we have data of movie ratings from Twitter stored in 3 files. The data was created from people who connected their IMDB profile with their Twitter accounts. Whenever they rated a movie on the IMDB website, an automated process generated a standard, well-structured tweet.

Run the following code to import the datasets.

In [ ]:
```
import pandas as pd
users = pd.read_csv("users.csv", index_col="user_id")
ratings = pd.read_csv("ratings.csv")
movies = pd.read_csv("movies.csv", index_col="movie_id", na_filter=False)
```

part a - 5 pts

In the DataFrame **movies**, the column "movie_title" contains both movie title and year. Based on this column, create two columns called "Title" and "Year" that contain only the title and year.

Hint:

- Each cell is a string whose last a few index always correspond to "year".
- Think about sequence index and list comprehension.
- Year should have type int, and title should have type string.

In [ ]:
```

```

part b - 5 pts

According to the column "Year" you create in part a, for year from 2010 to 2020 (inclusive), which year has the most number of movies and which year has the least number of movies?

Note: If you stuck on part a, the "Title" and "Year" columns are provided in file **title_year**. Run the code below to update your DataFrame. Keep in mind, you will lose the 5 pts in part a if you do this.

In [ ]:
```
title_year = pd.read_csv("title_year.csv", index_col="movie_id")
movies = pd.concat([movies, title_year], axis = 1)
```

In [ ]:
```

```

part c - 5 pts

In the DataFrame **movies**, the column "genres" contains the genres of a particular movie. If a single movie belongs to more than one genre, the genres are separated by pipe characters `"|"`. Based on the "genres" column, create a new column called "genres_list" that transform the cell in "genres" into a list of genres.

For example, for the first movie, the cell value for the "genres" column is `Action|Horror`. Then, it should be `["Action", "Horror"]` for the new created "Genres_list" column. Similarly, for second movie, it should be should be `["Comedy", "Fantasy", "Romance"]` for the new created "Genres_list" column.

In [ ]:

part d - 5 pts

Run the codes below to join three datasets and assign to variable name **ratings1**. Use **ratings1**, create a barplot that shows the frequency of different ratings for all the `Action` movies. In the barplot, the x-axis should be the different ratings, e.g. 5, 6, 7, 8 and etc; the y-axis should give the number of users that gave that ratings (If you decide to use a horizontal version, the values in x and y axis will be switched).

Note: If you stuck on part c, the "genres_list" column is provided in file **genres_list.csv**. Run the codes below to update your DataFrame **movies** before joining with other 2 datasets. Keep in mind, you will lose the 5 pts in part c if you do this.

In [ ]:
```
genres_list = pd.read_csv("genres_list.csv", index_col="movie_id")
movies = pd.concat([movies, genres_list], axis = 1)
```

In [ ]:
```
ratings1 = ratings.join(users, on = "user_id").join(movies, on = "movie_id")
```

In [ ]:

part e - 10 pts

write a query function named `query_movie` that takes the id of a particular movie as input and return a dictionary with keys: Title, Year, Genres, Number of Ratings, Average Ratings and their corresponding values as values.

- Generes should be in the list form.
- Number of Ratings is how many users have rated the movie.
- Average Ratings is average rating score of the movie.

In [ ]:

You can use the followings to test your function.

In [6]:
```
query_movie(903624)
```

Out[6]:
```
{'Name': 'The Hobbit: An Unexpected Journey',
 'Year': 2012,
 'Genres': "['Adventure', 'Family', 'Fantasy']",
 'Number of Ratings': 758,
 'Average Ratings': 7.881266490765172}
```

In [7]:
```
query_movie(1446192)
```

Out[7]:
```
{'Name': 'Rise of the Guardians',
 'Year': 2012,
 'Genres': "['Animation', 'Action', 'Adventure', 'Comedy', 'Family', 'Fantas
y']",
 'Number of Ratings': 239,
 'Average Ratings': 7.606694560669456}
```

In [8]:
```
query_movie(1757746)
```

Out[8]:
```
{'Name': 'Extracted',
 'Year': 2012,
 'Genres': "['Drama', 'Sci-Fi']",
 'Number of Ratings': 25,
 'Average Ratings': 7.08}
```

In [ ]:
```

```

Problem 2 (20 pts)

For each of the following two parts, you need to conduct a hypothesis test to test the claim. When conducting the test, please use the following procedures:

1. Formulate your null and alternative hypothesis.
2. Decide which test is appropriate.
3. Check if the assumption for the test you chose in step 2 is satisfied. If not, check the assumption.
4. Calculate the test statisitic (Z or t score) and the p-value.
5. Make a decision

part a - 10 pts

Suppose a study was designed to see if there is any difference in the social attitudes involving twenty sets of twins. To test te claim, one of the twins from each set was randomly assigned to live in a foreign country for 1 year, while the other stayed at home. The data can be found in the file **twins.csv**. Sample1 and Sample2 show the overall scores on social behavior from a questionnaire for the stay-home and live-abroad twins.

Conduct an appropriate test at the $\alpha = 0.01$ level of significance.

In [ ]:
```

```

part b - 10 pts

An experiment was taken to measure the effects of ozone. A group of 22 70-day-old rats were kept in an ozone environment for 7 days and their weight gains were recorded. Another group of 31 rats of a similar age were kept in an ozone-free environment for 7 days and their weight gains were recorded. The data is given in file **ratweight.csv**.

Is there a significant difference in the weight gain between group? Conduct a two-sided test to test at $\alpha = 0.1$ level of significance.

In [ ]:

Problem 3 (25 pts)

In the previous homework, we have seen the standard Cauchy distribution has the following pdf (f):

$$f(x) = \frac{1}{\pi(1+x^2)}, \text{ for } x \in \mathbb{R}.$$

The distribution function (F) of Cauchy distribution is given by:

$$F(x) = \frac{1}{2} + \frac{1}{\pi} arctan\,(x)\,.$$

part a - 5 pts

Using the distribution function given above, write down its inverse function $F^{-1}(u)$.

In [ ]:

part b - 10 pts

Using the inversion method we covered in the class, and assuming we can generate $U \sim Uniform(0,1)$ (use `random.uniform()` from numpy), write a function **random_cauchy** that generate random variable from the standard Cauchy distribution.

Note:

1. Your function does not have to have an input, but it needs to return a standard Cauchy random variable as output.
2. In your function, you should not use any form of random.cauchy function from other package. But you are welcome to use them to check if your function is correct.

In [ ]:

part c - 10 pts

Write a function **random_cauchy1** that generate random variable from the standard Cauchy distribution assuming the explicit form of F^{-1} is **not given**. In this case, you need to eveluate $F^{-1}(u)$ numerically. Again, assume we can generate $U \sim Uniform(0,1)$ (use `random.uniform()` from numpy)

Note:

1. Your function does not have to have an input, but it needs to return a Cauchy random variable as output.
2. In your function, you should not use any form of random.cauchy function from other package. But you are welcome to use them to check if your function is correct.

In [ ]:

Problem 4 (25 pts)

In the lecture, we discussed how to generate normal random variable using the Laplace distribution. In this problem, we will see how to generate normal random variable using the rejection method and the standard Cauchy distribution.

In this context, since we want to generate random variable from standard normal distribution, so we have the target density f

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, x \in \mathbb{R}.$$

The instrumental density function g is

$$g(x) = \frac{1}{\pi(1+x^2)}, x \in \mathbb{R}.$$

part a - 5 pts

To make the rejection method work, we need to find a constant c, such that

$$f(x) \leq cg(x) \text{ for all } x \in \mathbb{R}.$$

Also, we mentioned we want to specify c such that:

$$c = \sup_x \frac{f(x)}{g(x)}.$$

Verify that $c = \sqrt{\frac{2\pi}{e}} \approx 1.520347$.

Note: You can do this analytically or numerically.

In [ ]:

part b - 10 pts

Using c you find in part a and `randon_cauchy` you wrote in Problem 3, write a function to implement the rejection method to generate a standard normal random variable. Again, assume we can generate $U \sim Uniform(0, 1)$ (use `random.uniform()` from numpy)

Note:

1. Your function does not have to have an input, but it needs to return a standard nornal random variable as output.
2. In your function, you should not use any form of random.normal function from other package. But you are welcome to use them to check if your function is correct.
3. In case you stuck on Problem 3, you can use `random.standard_cauchy(size = 1)` function from numpy package to get a random variable from the standard Cauchy distribution.

In [ ]:

part c - 10 pts

Generate 5000 standard normal random variable using the function you defined in part b. Generate another 5000 standard normal random variable using `np.random.normal`. Make 2 histograms overlay with each other to see if they are close. In addition, generate a normal qq plot using the 5000 normal random variables from your function. Comment on the plots.

In [ ]: