Homework 1

Satwik Singh satwiks2

Question 1

1. Maximum: 100. Minimum: 37
2. Q1, Q2, Q3 = [68, 77, 87]
    - Sort the data and then take the cutoff elements for the required quartiles.
3. Mean = 76.715
    - Sum of data / number of datapoints.
4. Mode = 77, count = 37
    - Go throught the list keeping a count of all observed values and then get the one that occurs most.
5. Variance = 173.106
    - $$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x - \bar{x})^2$$
    - Iterate over array and sum (element - mean)^2 then divide by N

```
import numpy as np
from scipy import stats

data_path = '/Users/satwik/classes/cs412/hw1/data.online.scores.txt'
table = np.full((1000,3), np.inf)

with open(data_path, 'r') as dat:
    i = 0
    for row in dat:
        cols = row.split()
        table[i,] = cols
        i+=1

_, max_midterm, max_final = np.amax(table, axis=0)
_, min_mid, min_final = np.min(table, axis=0)

print("problem 1a.\n", 'max_mid:', max_midterm, 'min_mid:', min_mid)

quantiles = np.quantile(table[:, 1], [0.25, 0.5, 0.75])
means = np.mean(table[:, 1])
modes = stats.mode(table[:,1])
variance = np.var(table[:,1], ddof=0)

print("problem 1b.\n", "quantiles: ", quantiles)
print("problem 1c.\n", "means: ",means)
print("problem 1d.\n", "modes: ",modes)
print("problem 1e.\n", "variance: ",variance)
```

Question 2

1. Var Midterm-original = 173.106, Var Midterm-normalized = 1
    1. Used regular variance from previous question for original
    2. Normalized the midterm scores by using the following formula on each element

$$zscore = \frac{x_i - \mu}{\sigma}$$

2. Using the formula above we get norm score = $\frac{90-\mu}{\sigma}$ = 1.01
3. Correlation coefficient midterm-original and finals-original = 0.544
    1. using the formula coeff = $\frac{\sum_{i=1}^{n}(x_i y_i) - n\bar{x}\bar{y}}{n\sigma_x\sigma_y}$
4. Correlation coefficient midterm-normalized and finals-original = 0.544
5. Covariance midterm-original, finals-original = 78.176
    1. using the formula covar = $\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n}$

```
import numpy as np
from scipy import stats

data_path = '/Users/satwik/classes/cs412/hw1/data.online.scores.txt'
table = np.full((1000,3), np.inf)

with open(data_path, 'r') as dat:
    i = 0
    for row in dat:
        cols = row.split()
        table[i,] = cols
        i+=1
variance = np.var(table[:,1], ddof=0)
norm_table = stats.zscore(table[:,1:], axis=0)
norm_variance = np.var(norm_table[:,1], ddof=0)
norm = (90 - np.mean(table[:, 1])) / np.sqrt(np.var(table[:,1], ddof=0))
pearsons = stats.pearsonr(table[:,1], table[:,2])[0]
pearsons_n = stats.pearsonr(norm_table[:,0], table[:,2])[0]
covar = np.cov(table[:,1], table[:,2], ddof=0)

print("problem 2a.\n", "norm var: ", norm_variance, "variance: ",variance)
print("problem 2b.\n", "norm: ",norm)
print("problem 2c.\n", "pearsons: ",pearsons)
print("problem 2d.\n", "pearsons_n: ",pearsons_n)
print("problem 2e.\n", "covar: ",covar)
```

Question 3

1. MD(CML, CBL) = $\sqrt[h]{|x_{cml_1} - x_{cbl_1}||^h + ... + ||x_{cml_p} - x_{cbl_p}||^h}$
    1. h = 1; MD = 6152
    2. h = 2; MD = 715.328
    3. h = ∞; MD = 170
2. Cosine similarity = $\frac{\mathbf{x}.\mathbf{y}}{||\mathbf{x}||||\mathbf{y}||}$ = 0.841
    1. $x.y = x_1y_1 + ... + x_py_p$
    2. $||k|| = \sqrt{k_1^2 + .. + k_p^2}$
3. KL divergence $D_{KL}(p_{cml}(x))||p_{cbl}(x)) = \sum_{x\epsilon X} p_{cml}(x) ln \frac{p_{cml}(x)}{p_{cbl}(x)}$ = 0.207
    1. Where the probability distribution is calculated using $p(x_i) = \frac{x_i}{\sum_{i=1}^{N} x_i}$

```
import numpy as np
from scipy import stats
from scipy.spatial import minkowski_distance
from scipy.special import kl_div
from scipy.stats import chi2_contingency

path = '/Users/satwik/classes/cs412/hw1/data.libraries.inventories.txt'
table = np.full((2,100), np.inf)

with open(path, 'r') as dat:
    i = 0
    for row in dat:
        if i == 0:
            i+=1
            continue
        cols = row.split()
        table[i-1,] = cols[1:]
        i+=1

h1 = minkowski_distance(table[0,], table[1,], 1)
h2 = minkowski_distance(table[0,], table[1,], 2)
h_inf = minkowski_distance(table[0,], table[1,], np.inf)


cosine_sim = np.dot(table[0,], table[1,]) / (np.linalg.norm(table[0,]) * np.linalg.norm
print("Q3 ---")
print("h1: " + str(h1))
print("h2: " + str(h2))
print("h_inf: " + str(h_inf))
print("cosine_sim: " + str(cosine_sim))

prob_distribution = table / table.sum(axis=1).reshape(2,1)
KL = np.sum(kl_div(prob_distribution[0,], prob_distribution[1,]))
print("KL: " + str(KL))
```

Question 4

1. distance assuming symmetric = 0.016
    1. Assuming symmetry $d(i,j) = \frac{r+s}{q+r+s+t}$
2. jaccard coefficient - 0.732
    1. using formula $sim(i,j) = \frac{q}{q+r+s} = 1 - d(i,j)_{asymmetric}$
3. Chi sq statistic = 2450.716
    1. Create expected values table using formula:

       expected val = (row total * column total) / grand total
    2. Then use the expected value table to calculate chi squared statistic using:

$$\chi^2 = \sum \frac{(observed - expected)^2}{expected}$$

4. Using code the p value is close to 0 so we reject the null hypothesis at significance level 0.05.
    1. If we want to do it by hand we get df = 1 and alpha = 0.05
    2. using lookup table we get critical value = 3.841
    3. since critical value < calculated chi square value we reject the null hypothesis.

```
print("\nQ4 ---")
contingency_dat =np.array([[150,40],[15,3300]])
distance = (15+40)/(np.sum(contingency_dat))
print("sym_dist: " + str(distance))
jaccard = 150/(150+15+40)
print("jaccard: " + str(jaccard))

stat,p,ddof,expected = chi2_contingency(contingency_dat)
print("chi_stat: " + str(stat))
print("ddof: " + str(ddof))
alpha = 0.05
print("p value is " + str(p))
if p <= alpha:
    print('Dependent (reject H0)')
else:
    print('Independent (H0 holds true)')
```