CS37300 Homework	— Homework 2—	Last update: September 19, 2017
https://www.cs.purdue.edu/homes/ribeirob/courses/Fall2017		Due: Sept 22, 2017 (11:59pm)

Instructions and Policy: Each student should write up their own solutions independently. You need to indicate the names of the people you discussed a problem with; ideally you should discuss with no more than two other people.

- YOU MUST INCLUDE YOUR NAME IN THE HOMEWORK
- The answers (without the python scripts) MUST be in submitted in a single PDF file via Blackboard.
- The python scripts will be submitted separately via turnin at data.cs.purdue.edu.
- Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary.
- Theoretical questions MUST include the intermediate steps to the final answer.
- Zero points in any question where the python code answer doesn't match the answer in the PDF.
- Python code answers MUST adhere to the format described below.

There are TWO (2) questions in this homework.

Python code guidelines

Turn in each question of your homework in a separate python file named `hw2-X.py`, where X is the question number.

1. If the question has n items, the last n "print" statements should be just the output of those answers without any extra characters or empty lines. For instance, if the question is about the empirical average and variance, the output should be:

   7.64
   6.30

   And not:
   Average: 7.64
   Mode: 6.3

2. If the answer is a plot, it should be added to the PDF and, in the code, it should always be saved as an file (image or PDF), and not using plt.show()

Your code is REQUIRED to run on either Python 2 or Python 3 at scholar.rcac.purdue.edu. Preferably use Python 3 (Python 2 will also be accepted). The TA's will help you with the use of the scholar cluster. If the name of the executable is incorrect, it won't be graded. Please make sure you didn't use any library/source explicitly forbidden to use. If such library/source code is used, you will get 0 pt for the coding part of the assignment. If your code doesn't run on scholar.rcac.purdue.edu, then even if it compiles in another computer, your code will still be considered not-running and the respective part of the assignment will receive 0 pt.

Q1 (5 pts): Testing Hypotheses with Bayesian Monte Carlo Simulations.

Submission: You're required to create a single python script, `hw2-1.py`. DO NOT include the source code in your submitted PDF. You also need to answer the questions asked above (except for the graph) in the PDF you submit on Blackboard.
You're only allowed to use the following libraries: csv, numpy, matplotlib, random, datetime

In this question we will revisit the hypothesis test of our Wine experiment (anchoring effect). Execute the python code of the hypothesis test we saw in class (Lecture 4).

```
import numpy as np

X1000 = np.array([50,10,37,650,400,80,130])
X10 = np.array([20,30,60,10,100,40])

sum_X1000 = np.sum(X1000)
n1000 = X1000.shape[0]
sum_X10 = np.sum(X10)
n10 = X10.shape[0]

N = 500
alpha = 10
beta = 1

sampled_mu1000 = np.random.gamma(shape=(alpha + sum_X1000), scale=1./(beta + n1000), size=N)
sampled_mu10 = np.random.gamma(shape=(alpha + sum_X10), scale=1./(beta + n10), size=N)

total_1000_ge_10 = np.sum(sampled_mu1000 > sampled_mu10)

print("Empirical probability P[mu_1000 > mu_10 | Data] =", float(total_1000_ge_10)/N)
```

Statistically, the code assumes the data comes from

$$X_{\$1000} \sim \text{Poisson}(\lambda_{\$1000}),$$

from the students that saw the value \$1000 in the wine question and

$$X_{\$10} \sim \text{Poisson}(\lambda_{\$10}),$$

from the students that saw the value \$10 in the wine question. To be able to sample $\lambda_{\$1000}$ and $\lambda_{\$10}$ given the data, we need to define priors for $\lambda_{\$1000}$ and $\lambda_{\$10}$.

(a) (2pt) Posterior and variance.

(**1pt**) (1) Compute the empirical standard deviation of the posterior samples: `sampled_mu1000` and `sampled_mu10`. Compute the empirical standard deviation of the original data: `X1000` and `X10`. Report the results in the PDF and output these values in your python code.

(**1pt**) (2) How should we compare the variance of the posterior samples to the variance we see in the data? Describe your reasoning mathematically.
Hint: The posterior samples are the parameters of the Poisson. If the parameter of the Poisson is λ, then the variance is also λ. The data `X1000` and `X10` are assumed to be samples of Poissons with their respective parameters.

(b) (1pt) Now assume $X_{\$1000} \sim \text{Normal}(\mu_{\$1000}, \sigma)$ and $X_{\$10} \sim \text{Normal}(\mu_{\$10}, \sigma)$ (we are using the numpy convention for the Normal parameters, the statistical convention uses σ^2), where σ is a standard deviation parameter. Further assume the priors $\mu_{\$10} \sim \text{Normal}(10, 1)$ and $\mu_{\$1000} \sim \text{Normal}(10, 1)$. Let $X_{\$1000,1}, \ldots, X_{\$1000,7}$ be the values give by the seven students that saw \$1000 as the amount, and $X_{\$10,1}, \ldots, X_{\$10,6}$ be the values given by seven students that saw \$10 as the amount. Give the distribution of the posterior

$$P[\mu_{\$10} | X_{\$10,1}, \ldots, X_{\$10,6}]$$

and

$$P[\mu_{\$1000} | X_{\$1000,1}, \ldots, X_{\$1000,7}].$$

Hint: The posterior will be a Normal distribution. To find the parameters, check the equations in conjugate.pdf (see course website/schedule) under "Normal Likelihood (unknown mean, known variance)/ Normal Prior".

(c) (2pt) Q1(a) shows that the empirical variance in the data is much larger than that of our sampled posteriors. Rewrite the above python code to use the new Normal assumptions in Q1(b) with different values of $\sigma \in \{1, 10, 100\}$. (1) Describe how the new algorithm works in pseudo-code (in the PDF). (2) Give the empirical probability

$$P[\mu_{\$1000} > \mu_{\$10} | X_{\$10,1}, \ldots, X_{\$10,6}, X_{\$1000,1}, \ldots, X_{\$1000,7}]$$

for the different values of σ (in the PDF and print statements in the python source); (3) explain why σ impacts the results (in the PDF).

Q2 (5 pts): Data Visualization.

Submission: You're required to create a single python script, `hw2-2.py`. DO NOT include the source code in your submitted PDF. You also need to answer the questions asked above (except for the graph) in the PDF you submit on Blackboard.
You're only allowed to use the following libraries: csv, numpy, matplotlib, random, datetime

For this question we will be using a study of of air pollution in 41 cities across the US (the file access is provided below). Each row is a city and the columns are:

city: Name of the city;
SO2: SO2 content of air in micrograms per cubic metre;
temp: Average annual temperature in degrees Fahrenheit;
manu: Number of manufacturing enterprises employing 20 or more workers;
popul: Population size in thousands;
wind: Average annual wind speed in miles per hour;
precip: Average annual precipitation in inches;
predays: Average number of days with precipitation per year.

What might be a question of interest about these data? Probably "How is pollution level as measured by sulphur dioxide concentration related to the six other variables?" For now let's focus on the question, how many(if not all) of these six variables are most informative about the data. We'll perform a Principal Component Analysis for this. Please note, we have scaled the data for your use.

Create a 'dataset' matrix $\mathbf{X}$ using only the last 6 columns of the data, such that the resulting matrix is a 41×6 matrix,

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{41,6} \\ x_{2,1} & x_{2,2} & \cdots & x_{41,6} \\ \vdots & \vdots & \ddots & \vdots \\ x_{41,1} & x_{41,2} & \cdots & x_{41,6} \end{pmatrix}.$$

Denote the i-th row of $\mathbf{X}$ as a 6-dimensional column vector $\mathbf{x}_j = \begin{pmatrix} x_{1,j} \\ x_{2,j} \\ \vdots \\ x_{6,j} \end{pmatrix}$.

Let $\mathbf{X}'$ be the normalized version of $\mathbf{X}$, as

$$\mathbf{X}'_{i,j} = (\mathbf{X}_{i,j} - \mathbf{m}_j)/\hat{\sigma}_j,$$

where $\mathbf{m}$ is a vector with the column averages, whose j-th component is

$$\mathbf{m}_j = \frac{1}{41}\sum_{i=1}^{41} x_{i,j},$$

and

$$\hat{\sigma}_j = \sqrt{\frac{1}{41}\sum_{i=1}^{41}(x_{i,j} - \mathbf{m}_j)^2}.$$

Create a 6×6 matrix

$$\mathbf{S} = \sum_{j=1}^{41} \mathbf{x}_j \mathbf{x}_j^T.$$

To read the file, please use the following piece of code:

```
import urllib.request

my_url = "https://www.cs.purdue.edu/homes/ribeirob/courses/Fall2017/data/airpollution.csv"

local_filename, headers = urllib.request.urlretrieve(my_url)

with open(local_filename) as in_file:
    for line in in_file.readlines():
        # REMEMBER TO SKIP THE FIRST HEADER LINE
        # process the string line into the matrix
```

(a) (1pt) Compute the eigenvalues and eigenvectors of $\mathbf{S}$. Use `linalg.eig` in `numpy` to do this. Print the eigenvalues ordered from the largest **absolute** value to the smallest **absolute** value (one eigenvalue per line). Report these values in the PDF.
Note: Eigenvalues can be positive or negative, but in this problem they should all have the same sign.

(b) (2pt) An key aspect of dimensionality reduction is determining the correct number of dimensions k. Plot the eigenvalues you calculated in the last step from the largest **absolute** value to the smallest **absolute** value (report in PDF; in the source code, write the plot to a file). Looking at the plot, argue which value of k should be selected (in the PDF).
Hint: A good argument is that your choice captures most of the variance in the data. Compute the variance of your data and compare against the sum of the eigenvalues of the k eigenvector choice.

(c) (2pt) Consider now $k = 2$ (only two dimensions in the PCA). Create a new 6×2 eigenvector matrix $\mathbf{U}$, which uses only two eigenvectors associated with the largest two eigenvalues. The final step now is to transform your sample into the new subspace using the formula:

$$\mathbf{X_{new}} = \mathbf{X}' \cdot \mathbf{U}$$

Create a scatter plot with the 41 points in $\mathbf{X_{new}}$ save it in a file. Describe what this procedure is doing to our original data points (in the PDF) and report the plot in the PDF.