HW2P1

September 21, 2016

0.0.1 Homework 2: More Exploratory Data Analysis

0.1 Gene Expression Data and Election Polls

Due: Thursday, September 29, 2016 11:59 PM

Submission Instructions To submit your homework, create a folder named **lastname_firstinitial_hw#** and place your IPython notebooks, data files, and any other files in this folder. Your IPython Notebooks should be completely executed with the results visible in the notebook. We should not have to run any code. Make sure to share the private repo with my github account (mdog) and submit the repo path through blackboard.

0.2 Introduction

John Tukey wrote in Exploratory Data Analysis, 1977: "The greatest value of a picture is when it forces us to notice what we never expected to see." In this assignment we will continue using our exploratory data analysis tools, but apply it to new sets of data: gene expression and polls from the 2012 Presidental Election and from the 2014 Senate Midterm Elections.

First: You will use exploratory data analysis and apply the singular value decomposition (SVD) to a gene expression data matrix to determine if the the date that the gene expression samples are processed has large effect on the variability seen in the data.

Second: You will use the polls from the 2012 Presidential Elections to determine (1) Is there a pollster bias in presidential election polls? and (2) Is the average of polls better than just one poll?

Finally: You will use the HuffPost Pollster API to extract the polls for the current 2014 Senate Midterm Elections and provide a preliminary prediction of the result of each state.

Data We will use the following data sets:

1. A gene expression data set called `exprs_GSE5859.csv` and sample annotation table called `sampleinfo_GSE5859.csv` which are both available on Github in the 2014_data repository: expression data set and sample annotation table.
2. Polls from the 2012 Presidential Election: Barack Obama vs Mitt Romney. The polls we will use are from the Huffington Post Pollster.
3. Polls from the 2014 Senate Midterm Elections from the HuffPost Pollster API.

0.3 Load Python modules

```
In [1]: # special IPython command to prepare the notebook for matplotlib
        %matplotlib inline

        import requests
        from io import StringIO
        import numpy as np
        import pandas as pd # pandas
        import matplotlib.pyplot as plt # module for plotting
        import datetime as dt # module for manipulating dates and times
        import numpy.linalg as lin # module for performing linear algebra operatio
```

0.4 Problem 1

In this problem we will be using a gene expression data set obtained from a microarray experiement Read more about the specific experiment here. There are two data sets we will use:

1. The gene expression intensities where the rows represent the features on the microarray (e.g. genes) and the columsns represent the different microarray samples.
2. A table that contains the information about each of the samples (columns in the gene expression data set) such as the sex, the age, the treatment status, the date the samples were processed. Each row represents one sample.

Problem 1(a) Read in the two files from Github: exprs_GSE5859.csv and sampleinfo_GSE5859.csv as pandas DataFrames called `exprs` and `sampleinfo`. Use the gene names as the index of the `exprs` DataFrame.

```
In [2]: #your code here
```

Make sure the order of the columns in the gene expression DataFrame match the order of file names in the sample annotation DataFrame. If the order of the columns the `exprs` DataFrame do not match the order of the file names in the `sampleinfo` DataFrame, reorder the columns in the `exprs` DataFrame.

Note: The column names of the gene expression DataFrame are the filenames of the orignal files from which these data were obtained.

Hint: The method `list.index(x)` [read here] can be used to return the index in the list of the first item whose value is x. It is an error if there is no such item. To check if the order of the columns in `exprs` matches the order of the rows in `sampleinfo`, you can check using the method `.all()` on a Boolean or list of Booleans:

Example code: `(exprs.columns == sampleinfo.filename).all()`

```
In [3]: #your code here
```

Show the head of the two tables: `exprs` and `sampleinfo`.

```
In [ ]: #your code here
```

Problem 1(b) Extract the year and month as integers from the `sampleinfo` table.

Hint: To convert a Series or a column of a pandas DataFrame that contains a date-like object, you can use the `to_datetime` function [read here]. This will create a `DatetimeIndex` which can be used to extract the month and year for each row in the DataFrame.

```
In [4]: #your code here
```

Problem 1(c) Convert the dates in the `date` column from the `sampleinfo` table into days since October 31, 2002. Add a column to the `sampleinfo` DataFrame titled `elapsedInDays` containing the days since October 31, 2002. Show the head of the `sampleinfo` DataFrame which includes the new column.

Hint: Use the `datetime` module to create a new `datetime` object for the specific date October 31, 2002. Then, subtract the October 31, 2002 date from each date from the `date` column in the `sampleinfo` DataFrame.

```
In [5]: #your code here
```

Problem 1(d) Use exploratory analysis and the singular value decomposition (SVD) of the gene expression data matrix to determine if the date the samples were processed has large effect on the variability seen in the data or if it is just ethnicity (which is confounded with date).

Hint: See the end of the lecture from 9/23/2014 for help with SVD First subset the the `sampleinfo` DataFrame to include only the CEU ethnicity. Call this new subsetted DataFrame `sampleinfoCEU`. Show the head of `sampleinfoCEU` DataFrame.

```
In [6]: #your code here
```

Next, subset the `exprs` DataFrame to only include the samples with the CEU ethnicity. Name this new subsetted DataFrame `exprsCEU`. Show the head of the `exprsCEU` DataFrame.

```
In [7]: #your code here
```

Check to make sure the order of the columns in the `exprsCEU` DataFrame matches the rows in the `sampleinfoCEU` DataFrame.

```
In [8]: #your code here
```

Compute the average gene expression intensity in the `exprsCEU` DataFrame across all the samples. For each sample in the `exprsCEU` DataFrame, subtract the average gene expression intensity from each of the samples. Show the head of the mean normalized gene expression data.

```
In [9]: #your code here
```

Using this mean normalized gene expression data, compute the projection to the first Principal Component (PC1).

Hint: Use the `numpy.linalg.svd()` function in the `numpy.linalg` module (or the `scipy.linalg.svd()` function in the `scipy.linalg` module) to apply an singular value decomposition to a matrix.

```
In [10]: #your code here
```

Create a histogram using the values from PC1. Use a bin size of 25.

```
In [11]: #your code here
```

Create a scatter plot with the days since October 31, 2002 on the x-axis and PC1 on the y-axis.

```
In [12]: #your code here
```

Around what day do you notice a difference in the way the samples were processed?

```
In [13]: #your code here
```

Answer:

0.5 Discussion for Problem 1

Write a brief discussion of your conclusions to the questions and tasks above in 100 words or less.

---

1 Submission Instructions

To submit your homework, create a folder named **lastname_firstinitial_hw#** and place your IPython notebooks, data files, and any other files in this folder. Your IPython Notebooks should be completely executed with the results visible in the notebook. We should not have to run any code. Make sure to share the private repo with my github account (mdog) and submit the repo path through blackboard.

```
In [ ]:
```