second project

Please follow the following for the second project:

**1)** Select a dataset of your own choice (simple, complicated, whatever your

level of comfort is)

The dataset consists of an excel csv file. The data set has 10 parameters and 14,999 rows. The data is simulated (they really needed 0ne more simulated employee for an even 15,000. It is a decent sized data set. In looking through the data it is clear there will be some outlier data. Unfortunately, I could not find an ‘original' data set on kaggle or git hub with a readme that would provide additional information on each of the 10 parameters:

1. Satsatisfaction\_level: These numbers range from 9% to 100%. It might be interesting to see if there is a correlation between job satisfaction & if the person has left the company.
2. last\_evaluation: These numbers range from 33% to 100%. They appear to be the “grade” on the last evaluation. It might be interesting to see if there is a correlation between job satisfaction & evaluation.
3. number\_project: Numer of projects ranges from 2 to 7. It isn’t clear what constitutes a project or the time frame is (it per week, month, year or perhaps total projects completed over work life.
4. average\_montly\_hours: Average monthly hours. These values vary widely. The low is 96 (probably part-time employees) to a high of 310 (which has to be some substantial overtime. Typically a work month ranges from 20-23 working days (excludes holidays and weekends). Based on my experience in paying out a terminated employee the average is 21 work days per month. A 96 hour month is approx. 4.5 hours a day & 310 hour month is a 14 hour day. We can’t be sure if it is actual hours worked (most likely). Or if overtime hours are prorated at 1.5 or 2 (depending on Overtime hour policies)
5. time\_spend\_company: How many years at the company. These values range from 2 to 10. Note it is odd that there are no 1’s or 0’s (less than 1 year). This could be an indicator of low turnover.
6. Work\_accident: So if there has been an on-the job accident. Is either 1 or 0 (on/off). 1 for accident, 0 for no accidenother
7. Left: Meaning left the company. Is either 1 or 0 (on/off). 1 for left, 0 for not left.
8. promotion\_last\_5years: Is either 1 or 0 (on/off). 1 for promotion, 0 for no promotion.
9. sales: is really the occupational category. these include sales, accounting, R&D, IT, HR, support, etc
10. salary: has 3 parameters low, medium & high.

2) Chop the dataset using pandas to fit whatever visualization you have in

mind.

**Line 1:** bring in modules. I’ll be using numpy, pandas & matplotlib

**Line 2:** read data from the csv file & save it into a variable/dataframe called data.

**Line 3**: data shape we have a database of 14999 rows and 10 columns. This allows us to see the size of the data.

**Line 4 & 5:** Verify all data has loaded; use data.head(020 ) to look at the first 20 rows of the data & tail.data() to look at the last 5 rows. The empty parenthesis returns 5 rows if we wanted to look at more rows (as we did with the head) we would specify the number. All our data is loaded even though the last row in tail is numbered 14998 (we have 14999 rows) we have to keep in mind the first row is row 0.

**Line 6:** Verify that we are working with a data frame. We ‘know’ we have a data frame, type(data) verifies that the data was read in as a data frame .

**Line 7**: See if we have any missing data. We don’t have any missing data. We are asking if there are any null values. If we had any null values we would get ‘True” but since everything is ‘False’ we don’t have any null values.

**Line 8**: Use data.describe to see an overview of the data means, median, max, min, etc for each column. This gives us information on the columns with data, but the columns with word descriptors (such as salary with low, medium & high are not included). This is an interesting feature that gives us some statistical information about our data.

**Line9**: Now that we have an overview of what is in our data we can think about what information we will need. Since we want to see if there is a correlation between job satisfaction & leaving the company, as well as a correlation between job satisfaction & evaluation. We really only need these three columns. However, we may not want to get rid of all the other columns. Depending on what we see in our data, we may want to explore something else. We’ll drop the columns for: number\_project, Work\_accident, average\_montly\_hours, Work\_accident, and sales. Note, the “axis=1” indicates to the drop that this is for columns not rows. We’ll save these into a variable called df\_HR1; that way if we find there is something more we want to explore, we’ll go back to our whole data set.

3) Clean the dataset and drop missing values - if needed

There are no mission values in the data set.

4) Create two different types of visualization - minimum. Your choice, just

make sure they are different kinds - (for example a bar chart and a pie, but

not two pies).

Ok so I think it might be interesting to see if there is a correlation between employee satisfaction and their last evaluation.

**Line10**: We are ready to look at some graphs. First let’s use a pie chart to see the difference between the number of people that have left and stayed (ie not left) with the company. To create the pie chart we want to look at the distribution of the number of people that left vs the number that stayed. A 1 means the person left/quit so 3,571 quit (24%). A 0 indicates someone still with the company 11,428 (76%).

**Line11**: So here is our code for the pie chart. We assign labels that correspond to our numbers. The autopct (autopercent) formats how the percentages show in our pie chart. I don’t want decimals. The startangle tells us “where” to start the first line. The first “slice is at a 90 degree angle starting at the top. I’ve added plt.axis("equal") to make our pie chart appear round. There is a correlation between the monitor resolution and the pixels used for the graph. This causes the pie to appear oval instead of round. To force the pie to be round, we use the plt.axis("equal") . This is much more appealing to our eyes. “disconnect between the makes our pie chart look round. I’ve increased the size of the pie chart using the first 2 lines fig = plt.figure(figsize=(10,6)) and ax = fig.add\_axes((0,0,.5,1)).

**What our graph shows**: Now that we have a graph one of our initial assumptions (that because people stayed for a minimum of 2 years) that turnover was low is found to be incorrect. A 25% turnover is pretty high. Typically, you may have a department or two with high turnover, but is should be greatly reduced when looking at the entire organization.

**Line 12:** Now let’s look to see if there is a correlation between those that are leaving and job satisfaction. I’d like to have a bar graph (but I was finding it hard to make). I was able to make a graph of the total satisfaction levels among all employees. This Histogram gives an idea of the data distribution.

**Line13**: In order to make a bar graph we have to first group or data so that matplotlib can make the bar graph. First, we’ll divide into 2 groups for left/not left.

Line 14: We have multiple vales for job satisfaction that range from 9% to 100%. We will need to group the percentages together. We’ll use a “typical” 10% grading scale:

91-100 =A, greater than 91% Extremely Satisfied

81-90 =B, greater than 81% but less than 91% Very Satisfied

71-80=C, greater than 71% but less than 81% Satisfied

61-70=D, greater than 61% but less than 71% Dissatisfied

0-60 =F less than 61% Very Dissatisfied

This scale is somewhat arbitrary, but we need to start somewhere. If job satisfaction is only 60% ,it is a failing grade.

Through some searching & trail & error I figured out how to group these into my 5 categories.

**Line 15**: Figuring out how to get the counts for these proved even trickier. I tried in vain to use value\_counts() in an amazing number of configurations before I stumbled onto just using count(). (An entire evening of stumbling). I was unsuccessful in just getting only the ‘left’ column to show though. This did give me the numbers I needed to be able to make my bar plot. The numbers added to the 3571 employees that left so the numbers check out. You can look at my “ugly” Jupiter notebook if you want to see some of the codes I tried. (I didn’t start keeping track at the start).

**Line 16**: The bar plot showing satisfaction level for employees that left (Success at making a bar graph in matplotlib)

**Lines 17 & 18** The data for the bar plot for employees that stayed. My total came to 11428 so my numbers match.

**Line 19**: The bar plot showing satisfaction level for employees that stayed with the company.

**Line 20**: puts these two bar graphs together. Let’s take a moment to look at this graph. Now Very dissatisfied is our largest bar for both those who left & those who are still employed. This is partly because it contains all satisfaction scores from 0 to .61, while the other groupings only represent a 10% score distribution it still represents a “failing grade.” Of the 3571 employees who left group 2606 (73% were very dissatisfied. The still employed dissatisfaction level also looks high on the bar card, however it constitutes 4331 of the 11428 still employed (38%). This still seems to be a large group that is dissatisfied.

**Line 21:** Now let’s look at job satisfaction vs evaluation to see if there is a correlation between a good evaluation & satisfaction and a poor evaluation & dissatisfaction. We’ll start with a scatter plot, expecting to see outliers (based on the wide range of vales in data).

What the graph shows. There are so many data points they basically fill the whole graph. Points are grouped on the upper right corner (high satisfaction & high evaluation) like we would expect. However we did not expect that the high satisfaction would be also predominately grouped with lower evaluations as well. In addition, there is a significant grouping of employees with high performance ratings but low job satisfaction.

5) You can have more visualizations if you want

6) Make sure your readme file is perfect!

7) Submit on github and kaggle (if using kaggle datasets), just github if you

are using data from other sources like drivedata.org or other census data.