

matplotlib_line_graphs_workbook_answers

September 21, 2020

1 MATPLOTLIB LINE GRAPS WORKBOOK

Remember, there are different ways to write code to get the same answer, so your answer can be correct and different to the answer example!

If you feel stuck and want some in person help, then have a look at the events page to join in a workshop <https://swamphen.co.uk/events>.

```
In [ ]: # import salaries data
        from google.colab import files
        uploaded = files.upload()

In [ ]: # read in salaries data to colab
        import csv
        with open('Salaries.csv') as data:
            reader = csv.reader(data, delimiter = ',')
            for i in reader:
                my_list = list(reader)

        print(my_list)

In [ ]: # extract the data to use
        # change rank to AssocProf = 0, AsstProf = 1, Prof = 2
        # change discipline to A = 0, B = 1
        # extract years since phd
        # extract years of service
        # change sex to Female = 0, Male = 1
        # extract salary ($)

        rank = []
        discipline = []
        years_phd = []
        years = []
        sex = []
        salary = []

        for row in my_list:
            if row[0] == 'AssocProf':
```

```

        rank.append(0)
    elif row[0] == 'AsstProf':
        rank.append(1)
    else:
        rank.append(2)
    if row[1] == 'A':
        discipline.append(0)
    else:
        discipline.append(1)
    years_phd.append(int(row[2]))
    years.append(int(row[3]))
    if row[4] == 'Female':
        sex.append(0)
    else:
        sex.append(1)
    salary.append(int(row[5]))

print(rank, discipline, years, sex, salary)

```

```

In [ ]: # import matplotlib
import matplotlib.pyplot as plt

```

```

In [ ]: # plot a line graph of salary in green with a solid line
# add in axes labels
plt.plot(salary, 'g-')
plt.xlabel('number')
plt.ylabel('salary ($)')

```

```

In [ ]: # what does this tell you about the range of salaries?

# band where most people fall with a few outliers

```

```

In [ ]: # plot years' service in red with a dotted line
# add in a graph title
plt.plot(years, 'r:')
plt.title('years service in academia')

```

```

In [ ]: # what does this tell you about years' service?

# full range of years represented
# less between 40-50 years

```

```

In [ ]: # plot years' service against salary in black with a dashed line and a
# dot at the points
plt.plot(years, salary, 'k--.')

```

```

In [ ]: # how could you change this horrible mess to a sensible line graph?

# sort on years service

```

```

In [ ]: # we will come back to plotting this new line graph with data sorted
        # in numerical order as it is not trivial in native Python, but is
        # in numpy and pandas!

In [ ]: # what graph should you use to show the data in this form better?

        # plot it!

        plt.plot(years, salary, 'k.')

In [ ]: # does this show any correlation between salary and years service?

        # weak correlation, but low paid people at all ages and reasonable paid people
        # at all ages

In [ ]: # plot a line graph of sex in yellow dot dash line add axes labels
        plt.plot(sex, 'y-.')
        plt.xlabel('number')
        plt.ylabel('sex f = 0, m = 1')

In [ ]: # what does this tell you about the number of men and women

        # a lot more men than women

In [ ]: # plot salary as a function of gender in green solid line with a star as a marker
        plt.plot(salary, sex, 'g*-')

In [ ]: # horrible looking graph again!
        # change your plot statement so there is no line plotted to make it look better
        plt.plot(salary, sex, ('*g'))

In [ ]: # what does that tell you about the split in wages between men and women

        # no high paid women there
        # women congregated in lower pay scales

In [ ]: # plot years_phd and years on the same graph
        plt.plot(years_phd)
        plt.plot(years)

In [ ]: # choose lines so you can see both data sets
        plt.plot(years_phd, 'k.-')
        plt.plot(years, 'r.:')

In [ ]: # what would be the best graph to choose to show this data?
        plt.hist([years_phd, years])

```