Learnings from the Guided Modules

Learnings from the Guided Modules

#	Learning element	by when	Status	Tags	SUMMARY OF KEY LEARNINGS - beyond tutorials	PDF
1	LEARNING THROUGH PROJECTS		☐			
2	Guided project 1: Dr Semmelweiss and the discovery of Handwashing	@Mar 31, 2021	✓	categorical, data manipulation, descriptive statistics, importing & cleaning data, numerical, pandas, probability & statistics	The exercise helped me to apply and deepen my knowledge from the pandas tutorial. I now feel familiar with using the Jypiter Notebook, transforming and aggregating data. The elements that I had not yet learned in the tutorial at the time of the exercise, they the following: line plot: two lines for one category ax = clinic_1.plot(x="year", y="proportion_deaths",label="clinic_1") clinic_2.plot(x="year", y="proportion_deaths", label="clinic_2", ax=ax, ylabel="Proportion deaths") parse_dates=["date"] confidence_interval = pd.Series(boot_mean_diff).quantile([0.025, 0.975])	GuidedProject1_SemmelweissHandwashing.pdf
3	Guided project 2: A Visual History of Nobel Prize Winners	@Mar 31, 2021	✓	categorical, data manipulation, datetime, descriptive statistics, grouping data, importing & cleaning data, numerical, numpy, pandas, probability & statistics, regression line, seaborn	The exercise helped me to apply and deepen my knowledge from the pandas tutorial. I now feel familiar with aggregating data and have gained more routine in applying variables for grouped-over operations vs as selected criterion. The elements that I had not yet learned in the tutorial at the time of the exercise, they are the following: ax=sns.lineplot(x='decade', y='usa_born_winner', data=nobel, ci=None) len() np.floor() as_index=False from matplotlib.ticker import PercentFormatter ax.yaxis.set_major_formatter(PercentFormatter(1.0)) .nsmallest() .nlargest() filter: repeater = nobel.groupby('full_name').filter(lambda group:len(group)>=2) row argument: sns.lmplot(x ='year', y='age',row ='category', data=nobel,ci=None,lowess=True, aspect=2, line_kws={'color' : 'black'})	GuidedProject2_NobelPrizeWinners.pdf
4	Guided project 3: Do Left-handed People Really Die Young?	@Mar 31, 2021	✓	categorical, data manipulation, descriptive statistics, grouping data, importing & cleaning data, numerical, numpy, pandas, probability & statistics	The exercise helped me to apply and deepen my knowledge from the pandas tutorial. However, this exercise was a bit too focused on Bayes' theorem and the calculation of its different elements. The exercise has one mistake in it and some sub-tasks I could only solve by checking the hints for solution. The elements that I had not yet learned in the tutorial at the time of the exercise, they are the following: [-10:].mean() = mean of the last 10 elements [:10].mean() = mean of the first 10 elements when loading data use sep = '\t' and skiprows=[1] to account for the dataset's format, e.g. data=pd.read_csv(data_url_2,sep = '\t',skiprows=[1]) Bayes' theorem P(A|LH)=P(LH|A)*P(A) / P(LH) https://en.wikipedia.org/wiki/Bayes'_theorem mean(axis = 1) One can calculate the mean of two columns row-wise by using the .mean() function and setting the parameter axis = 1. : my_data["means"] = my_data[["column_1", "column_2"]].mean(axis = 1) One can plot two lists or arrays of the same length (one for x values and one for y values) like this: fig, ax = plt.subplots() ax.plot(x_list, y_list, label = "label for legend") np.nansum() https://docs.scipy.org/doc/numpy-1.10.4/reference/generated/numpy.nansum.html	GuidedProject3_LeftHanded.pdf