Medical Data Visualizer

You will be working on this project with our Gitpod starter code.

We are still developing the interactive instructional part of the Python curriculum. For now, here are some videos on the freeCodeCamp.org YouTube channel that will teach you everything you need to know to complete this project:

- Python for Everybody Video Course (14 hours)
- How to Analyze Data with Python Pandas (10 hours)

In this project, you will visualize and make calculations from medical examination data using `matplotlib`, `seaborn`, and `pandas`. The dataset values were collected during medical examinations.

Data description

The rows in the dataset represent patients and the columns represent information like body measurements, results from various blood tests, and lifestyle choices. You will use the dataset to explore the relationship between cardiac disease, body measurements, blood markers, and lifestyle choices.

File name: medical_examination.csv

Feature	Variable Type	Variable	Value Type
Age	Objective Feature	`age`	int (days)
Height	Objective Feature	`height`	int (cm)
Weight	Objective Feature	`weight`	float (kg)
Gender	Objective Feature	`gender`	categorical code
Systolic blood pressure	Examination Feature	`ap_hi`	int
Diastolic blood pressure	Examination Feature	`ap_lo`	int
Cholesterol	Examination Feature	`cholesterol`	1: normal, 2: above normal, 3: well above normal
Glucose	Examination Feature	`gluc`	1: normal, 2: above normal, 3: well above normal
Smoking	Subjective Feature	`smoke`	binary
Alcohol intake	Subjective Feature	`alco`	binary
Physical activity	Subjective Feature	`active`	binary
Presence or absence of cardiovascular disease	Target Variable	`cardio`	binary

Instructions

Create a chart similar to `examples/Figure_1.png`, where we show the counts of good and bad outcomes for the `cholesterol`, `gluc`, `alco`, `active`, and `smoke` variables for patients with `cardio=1` and `cardio=0` in different panels.

By each number in the `medical_data_visualizer.py` file, add the code from the associated instruction number below.

1. Import the data from `medical_examination.csv` and assign it to the `df` variable.

kilograms by the square of their height in meters. If that value is > 25 then the person is overweight. Use the value `0` for NOT overweight and the value `1` for overweight.

3. Normalize data by making `0` always good and `1` always bad. If the value of `cholesterol` or `gluc` is 1, set the value to `0`. If the value is more than `1`, set the value to `1`.
4. Draw the Categorical Plot in the `draw_cat_plot` function.
5. Create a DataFrame for the cat plot using `pd.melt` with values from `cholesterol`, `gluc`, `smoke`, `alco`, `active`, and `overweight` in the `df_cat` variable.
6. Group and reformat the data in `df_cat` to split it by `cardio`. Show the counts of each feature. You will have to rename one of the columns for the `catplot` to work correctly.
7. Convert the data into `long` format and create a chart that shows the value counts of the categorical features using the following method provided by the seaborn library import: `sns.catplot()`.
8. Get the figure for the output and store it in the `fig` variable.
9. Do not modify the next two lines.
10. Draw the Heat Map in the `draw_heat_map` function.
11. Clean the data in the `df_heat` variable by filtering out the following patient segments that represent incorrect data:
    - diastolic pressure is higher than systolic (Keep the correct data with `(df['ap_lo'] <= df['ap_hi'])`)
    - height is less than the 2.5th percentile (Keep the correct data with `(df['height'] >= df['height'].quantile(0.025))`)
    - height is more than the 97.5th percentile
    - weight is less than the 2.5th percentile
    - weight is more than the 97.5th percentile
12. Calculate the correlation matrix and store it in the `corr` variable.
13. Generate a mask for the upper triangle and store it in the `mask` variable.
14. Set up the `matplotlib` figure.
15. Plot the correlation matrix using the method provided by the `seaborn` library import: `sns.heatmap()`.
16. Do not modify the next two lines.

Development

Write your code in `medical_data_visualizer.py`. For development, you can use `main.py` to test your code.

Testing

The unit tests for this project are in `test_module.py`. We imported the tests from `test_module.py` to `main.py` for your convenience.

Submitting

Copy your project's URL and submit it to freeCodeCamp.

Solution Link

ex: https://replit.com/@camperbot/hello

I've completed this challenge

Get a Hint

Ask for Help