team13_capstone_project

June 1, 2020

1 Team13: Capstone project of Python Bootcamp

This is the Capstone project for Team 13 of the Python Data Analysis Bootcamp. We are trying, more or less, to follow the structure of jupytemplate.

1.1 Purpose

State the purpose of the notebook.

1.2 Methology

Quickly describe assumptions and processing steps.

1.3 TODO / Improvements

- ☒ Find a dataset that has at least 2 CSV files
- ☐ Come up with 5 questions that you want to answer while exploring the dataset
- ☐ Perform EDA (Exploratoty Data Analysis) on your dataset with basic visualisations

1.4 Results

1.5 Setup

[16]:
```
# install system dependencies
import sys
import os

!conda install -c conda-forge --yes --prefix {sys.prefix} pandas jupyterthemes␣
↪seaborn jupyter_contrib_nbextensions pandoc
```

```
Collecting package metadata (current_repodata.json): done
Solving environment: done

# All requested packages already installed.
```

1.5.1 Library Import

[96]:
```
# load libraries and setup environment
# mandatory
import pandas as pd

%matplotlib inline
import matplotlib.pyplot as plt

# optional
import numpy as np
import seaborn as sns
from jupyterthemes import jtplot
from IPython.core.display import HTML
jtplot.style(theme='monokai', context='notebook', ticks=True, grid=False)
```

1.6 Parameter definition

We set all relevant parameters for our notebook. By convention, parameters are uppercase, while all the other variables follow Python's guidelines.

1.7 Data import

We retrieve all the required data for the analysis.

[165]:
```
cost_of_living = pd.read_csv('../data/andytran11996_cost-of-living/
 ↪datasets_73059_162758_cost-of-living-2018.csv')

# we are droppping the Rank column because it's entirely empty
cost_of_living = cost_of_living.drop(columns = 'Rank')

life_satisfaction = pd.read_csv('../data/roshansharma_europe-datasets/
 ↪datasets_231225_493692_life_satisfaction_2013.csv')
life_satisfaction = life_satisfaction.rename(columns = { "prct_life_satis_high":
 ↪ "People with highest life satisfaction [%]" })

print('successfully imported')
```

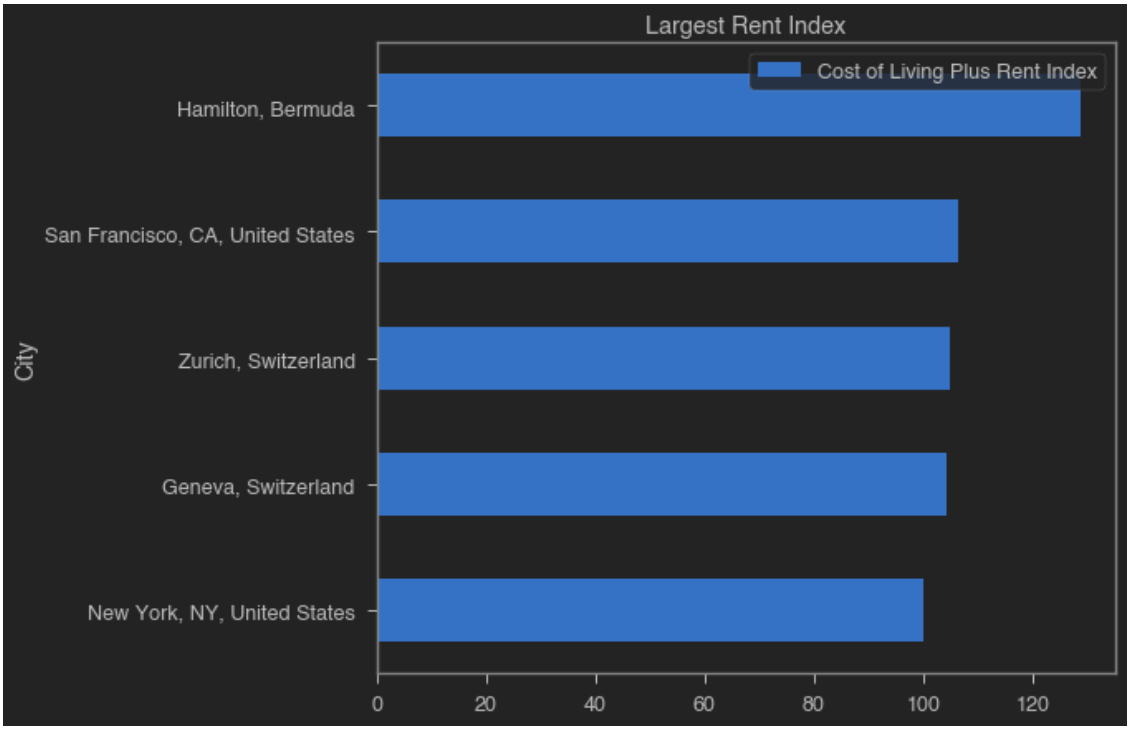
```
successfully imported
```

1.8 Data processing

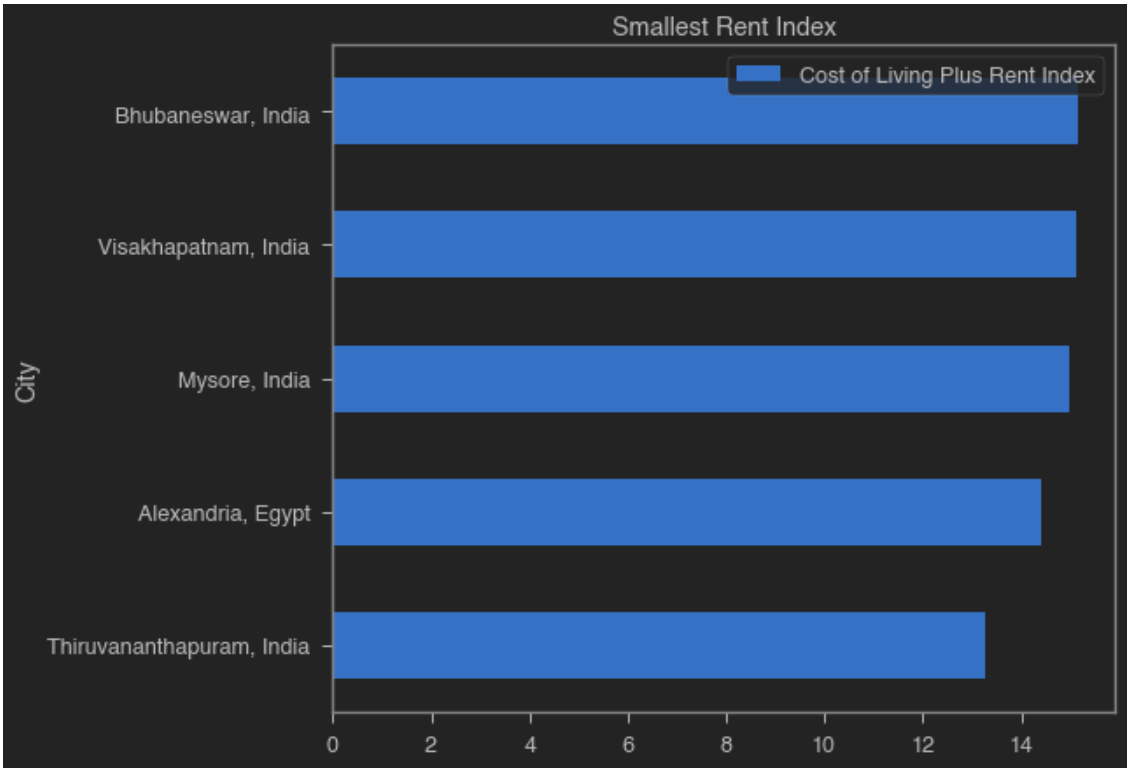
1.8.1 1. What are the five cities with the highest/lowest cost of living (incl. rent)?

[173]:
```
caption_column = 'City'
index_column = 'Cost of Living Plus Rent Index'

def display_cost_of_living(costs, title):
```

```
    filtered_costs = costs[[caption_column, index_column]].
 ↪sort_values(index_column, ascending = True)
    filtered_costs.plot.barh(title = title, x = caption_column, y =␣
 ↪index_column)
    plt.show()
    display(filtered_costs.sort_values(index_column, ascending = False).style.
 ↪hide_index())

# print the ten most expensive cities in the database in 2018
display_cost_of_living(cost_of_living.nlargest(5, index_column), 'Largest Rent␣
 ↪Index')
display_cost_of_living(cost_of_living.nsmallest(5, index_column), 'Smallest␣
 ↪Rent Index')
```



<pandas.io.formats.style.Styler at 0x7f3fe9d75210>




```
<pandas.io.formats.style.Styler at 0x7f3fd43c7b90>
```

1.9 2. What are the five happiest countries in Europe?

[174]:
```
index_column = "People with highest life satisfaction [%]"
caption_column = 'country'

life_satisfaction = life_satisfaction[[caption_column, index_column]]
life_satisfaction = life_satisfaction.nlargest(5, index_column)
life_satisfaction = life_satisfaction.sort_values(index_column, ascending =␣
 ↪True)
life_satisfaction.plot.barh(title = 'Percentage of satisfied people', x =␣
 ↪caption_column, y = index_column)
plt.show()
display(life_satisfaction.sort_values(index_column, ascending = False).style.
 ↪hide_index())
```

```
      ␣
↪---------------------------------------------------------------------------
```

```
KeyError                                  Traceback (most recent call␣
↪last)

<ipython-input-174-7c12f16eb9d6> in <module>
  2 caption_column = 'country'
  3
----> 4 life_satisfaction = life_satisfaction[[caption_column, index_column]]
  5 life_satisfaction = life_satisfaction.nlargest(5, index_column)
  6 life_satisfaction = life_satisfaction.sort_values(index_column,␣
↪ascending = True)


~/miniconda3/envs/team13/lib/python3.7/site-packages/pandas/core/frame.
↪py in __getitem__(self, key)
2804             if is_iterator(key):
2805                 key = list(key)
-> 2806             indexer = self.loc._get_listlike_indexer(key, axis=1,␣
↪raise_missing=True)[1]
2807
2808         # take() does not accept boolean indexers


~/miniconda3/envs/team13/lib/python3.7/site-packages/pandas/core/
↪indexing.py in _get_listlike_indexer(self, key, axis, raise_missing)
1551
1552         self._validate_read_indexer(
-> 1553             keyarr, indexer, o._get_axis_number(axis),␣
↪raise_missing=raise_missing
1554         )
1555         return keyarr, indexer


~/miniconda3/envs/team13/lib/python3.7/site-packages/pandas/core/
↪indexing.py in _validate_read_indexer(self, key, indexer, axis, raise_missing)
1644             if not (self.name == "loc" and not raise_missing):
1645                 not_found = list(set(key) - set(ax))
-> 1646                 raise KeyError(f"{not_found} not in index")
1647
1648             # we skip the warning on Categorical/Interval


KeyError: "['People with highest life satisfaction [%]'] not in index"
```

1.10 References

- data for the cost of living

- base data for countries of the world
- data for life expectancy from the WHO
- roshansharma_europe-datasets