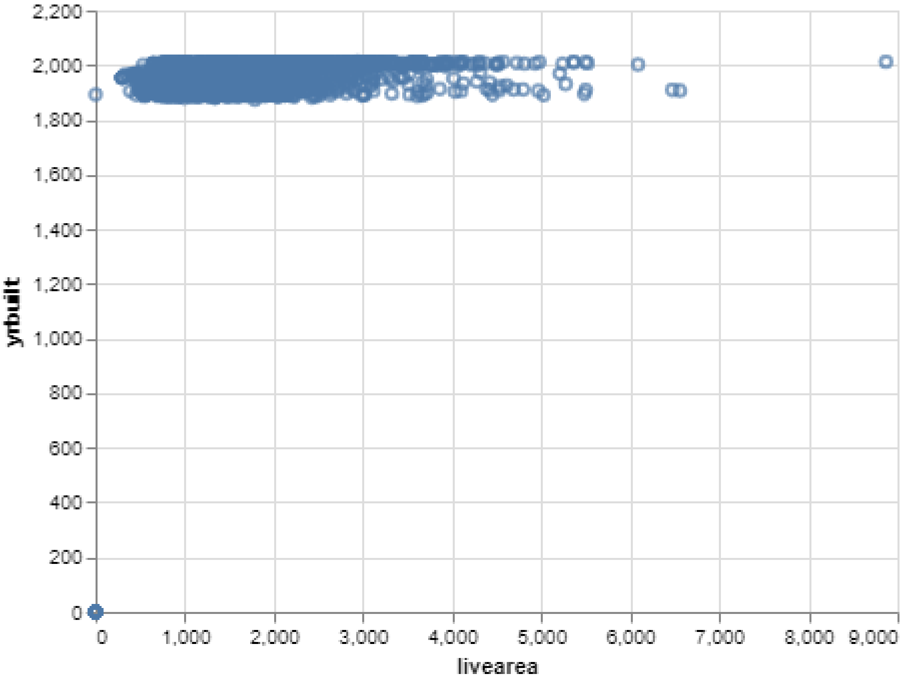
Finishing The Semester

Author: Gabriel Sanahuano

Coding Challenge

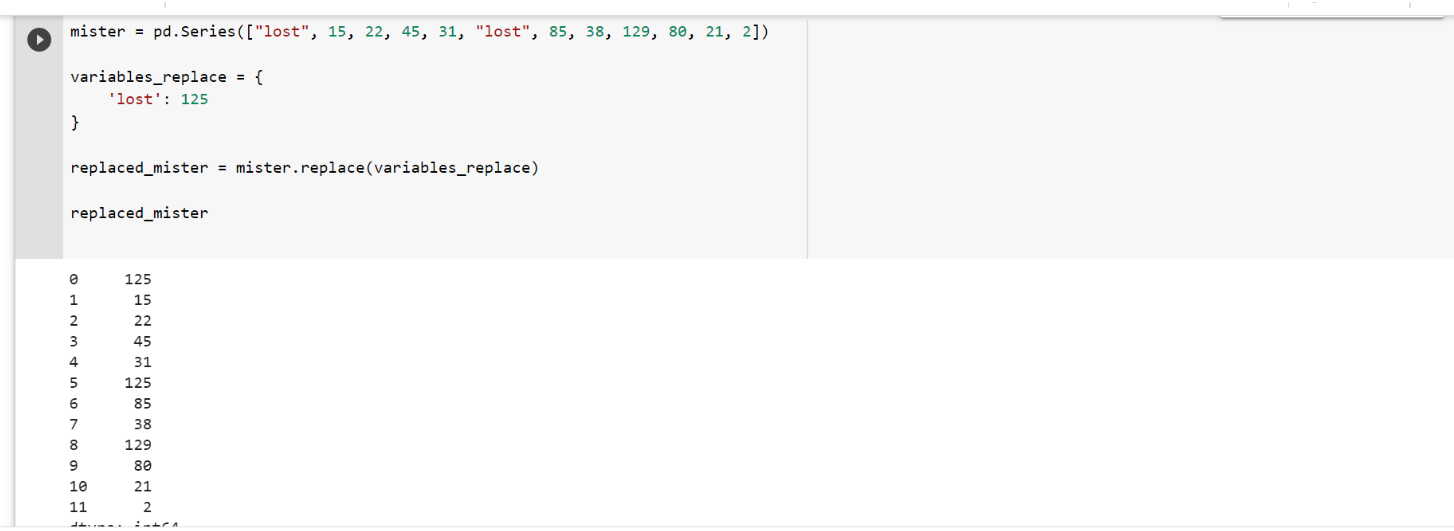
Challenge 1

Split Entry houses are a failed building experiment in the United States. Use the data from our Denver homes project, as shown below, to recreate the following graphic.



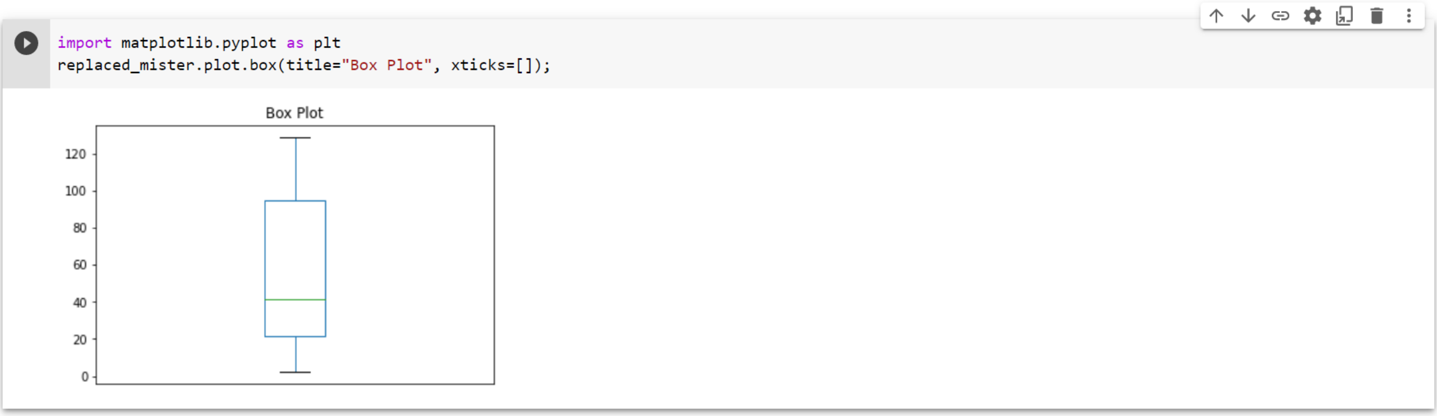
Challenge 2

Our computations can't be done with missing values. Programmatically replace all the lost values with 125 and make a box-plot.



```
mister = pd.Series(["lost", 15, 22, 45, 31, "lost", 85, 38, 129, 80, 21, 2])

variables_replace = {
    'lost': 125
}

replaced_mister = mister.replace(variables_replace)

replaced_mister
```

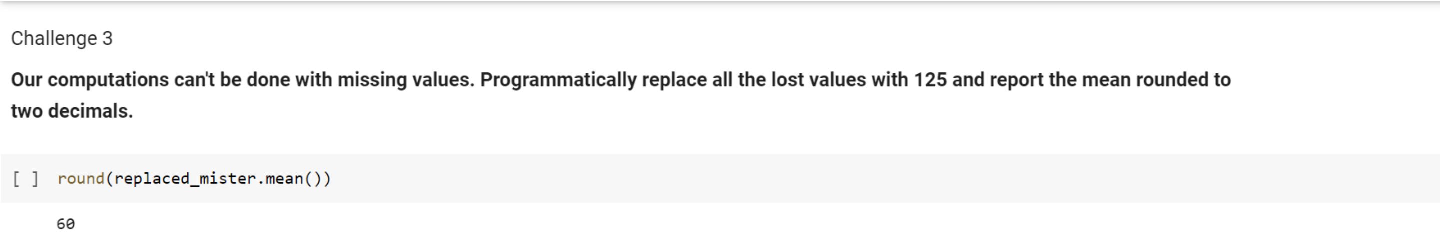
```
0     125
1      15
2      22
3      45
4      31
5     125
6      85
7      38
8     129
9      80
10     21
11      2
dtype: int64
```



```
import matplotlib.pyplot as plt
replaced_mister.plot.box(title="Box Plot", xticks=[]);
```

Challenge 3

Our computations can't be done with missing values. Programmatically replace all the lost values with 125 and report the mean rounded to two decimals.



Challenge 3

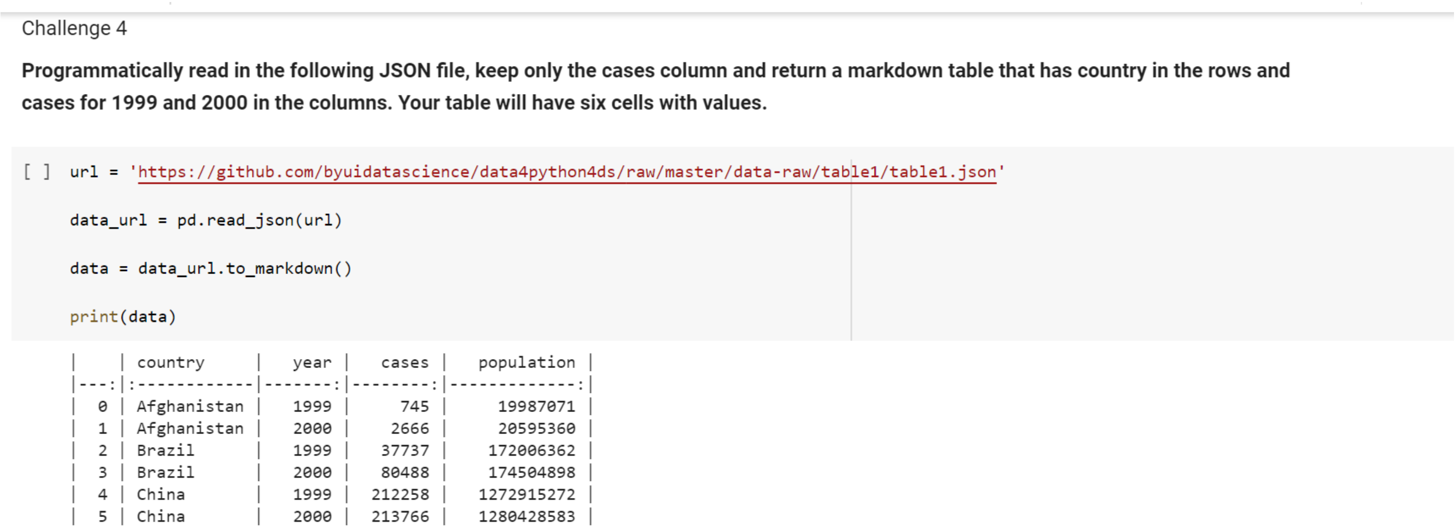
Our computations can't be done with missing values. Programmatically replace all the lost values with 125 and report the mean rounded to two decimals.

```
[ ] round(replaced_mister.mean())
```

```
60
```

Challenge 4

Programmatically read in the following JSON file, keep only the cases column and return a markdown table that has country in the rows and cases for 1999 and 2000 in the columns. Your table will have six cells with values.



Challenge 4

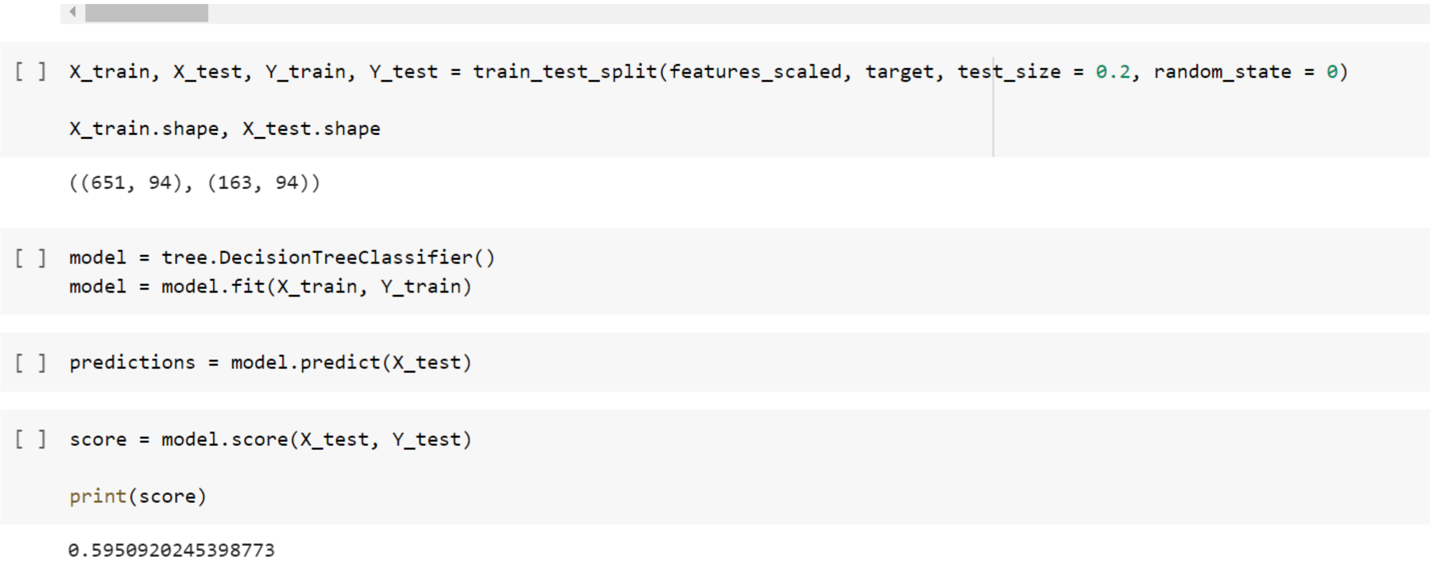
Programmatically read in the following JSON file, keep only the cases column and return a markdown table that has country in the rows and cases for 1999 and 2000 in the columns. Your table will have six cells with values.

```
[ ] url = 'https://github.com/byuidatascience/data4python4ds/raw/master/data-raw/table1/table1.json'

    data_url = pd.read_json(url)

    data = data_url.to_markdown()

    print(data)
```

```
|    | country     |   year |   cases |   population |
|---:|:------------|-------:|--------:|-------------:|
|  0 | Afghanistan |   1999 |     745 |     19987071 |
|  1 | Afghanistan |   2000 |    2666 |     20595360 |
|  2 | Brazil      |   1999 |   37737 |    172006362 |
|  3 | Brazil      |   2000 |   80488 |    174504898 |
|  4 | China       |   1999 |  212258 |   1272915272 |
|  5 | China       |   2000 |  213766 |   1280428583 |
```

Challenge 5

Use our cleaned example of the star wars data from project 6 to predict the gender of the respondent to the survey. Report your precision and a feature importance plot.



```
[ ] X_train, X_test, Y_train, Y_test = train_test_split(features_scaled, target, test_size = 0.2, random_state = 0)

    X_train.shape, X_test.shape
```

```
((651, 94), (163, 94))
```

```
[ ] model = tree.DecisionTreeClassifier()
    model = model.fit(X_train, Y_train)
```

```
[ ] predictions = model.predict(X_test)
```

```
[ ] score = model.score(X_test, Y_test)

    print(score)
```

```
0.5950920245398773
```

Google Colab

https://colab.research.google.com/gist/gabecastri/f223272bbdc5724151947bfe4a8e5a97/copy-of-welcome-to-colaboratory.ipynb

...

APPENDIX A (PYTHON SCRIPT)

```
import pandas as pd
import altair as alt
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn import tree
from sklearn.ensemble import GradientBoostingClassifier
from sklearn import metrics

from sklearn.naive_bayes import GaussianNB

dat_home = pd.read_csv(url).sample(n=4500, random_state=15)

chart = (alt.Chart(dat_home)
    .encode(
      alt.X('livearea'),
      alt.Y('yrbuilt')
      )
    .mark_point())

chart

mister = pd.Series(["lost", 15, 22, 45, 31, "lost", 85, 38, 129, 80, 21, 2])

variables_replace = {
    'lost': 125
}

replaced_mister = mister.replace(variables_replace)

replaced_mister

import matplotlib.pyplot as plt
replaced_mister.plot.box(title="Box Plot", xticks=[]);

round(replaced_mister.mean())

url = 'https://github.com/byuidatascience/data4python4ds/raw/master/data-raw/table1/table1.json'

data_url = pd.read_json(url)

data = data_url.to_markdown()

print(data)

url = "http://byuistats.github.io/CSE250-Course/data/clean_starwars.csv"
dat = pd.read_csv(url)

dat.head(3)

dat_home.isnull().sum()
```

```
features = dat.drop(columns = ['gender'])
target = dat.filter(['gender'])

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()

scaler.fit(features)
features_scaled = scaler.transform(features)
features_scaled = pd.DataFrame(features_scaled, columns=features.columns)

features_scaled.head()

X_train, X_test, Y_train, Y_test = train_test_split(features_scaled, target, test_size = 0.2, ra

X_train.shape, X_test.shape

model = tree.DecisionTreeClassifier()
model = model.fit(X_train, Y_train)

predictions = model.predict(X_test)

score = model.score(X_test, Y_test)

print(score)
```