

Python Project

Generated by Doxygen 1.9.6

1 Python Project	1
1.1 Description	1
1.2 Libraries/Modules	1
1.3 Author	1
2 Namespace Index	3
2.1 Namespace List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 projekt Namespace Reference	7
4.1.1 Function Documentation	8
4.1.1.1 boxplots()	8
4.1.1.2 gdp_time()	8
4.1.1.3 model_evaluation()	9
4.1.1.4 model_summary()	9
4.1.1.5 regression_model()	10
4.1.2 Variable Documentation	10
4.1.2.1 btn_boxplots	10
4.1.2.2 btn_gdp_over_time	10
4.1.2.3 btn_model_evaluation	10
4.1.2.4 btn_model_summary	10
4.1.2.5 btn_regression_model	11
4.1.2.6 C	11
4.1.2.7 consumption_data	11
4.1.2.8 country_list	11
4.1.2.9 country_menu	11
4.1.2.10 data	11
4.1.2.11 gdp_data	11
4.1.2.12 L	12
4.1.2.13 M	12
4.1.2.14 model	12
4.1.2.15 root	12
4.1.2.16 selected_country	12
4.1.2.17 summary	12
4.1.2.18 width	13
4.1.2.19 X	13
4.1.2.20 x	13
4.1.2.21 y	13
5 File Documentation	15
5.1 projekt.py File Reference	15

Chapter 1

Python Project

1.1 Description

Projekt zaliczeniowy z przedmiotu Języki Skryptowe. Program polega na stworzeniu modelu regresji liniowej zależności między PKB i konsumpcją. Stworzeniu jego podsumowania na podstawie, którego zostały wyciągnięte wnioski i został on oceniony. Stworzone również zostały dodatkowe wykresy: pudełkowe i wykresy zależności PKB od czasu dla każdego kraju.

Program pobiera dane o PKB i konsumpcji (źródło: <https://data.oecd.org/gdp/gross-domestic-product-gdp.htm>, <https://data.oecd.org/hha/household-spending.htm>) z plików csv(`consumption_`↵`data.csv`, `gdp_data.csv`). Są one wykorzystane do stworzenia modelu i wykresów za pomocą odpowiednich bibliotek. Następnie w GUI użytkownik może wyświetlić model regresji, jego podsumowanie i ocenę oraz wykresy pudełkowe, wykresy zależności PKB od czasu dla konkretnego kraju, który użytkownik wybierze.

1.2 Libraries/Modules

```
- pandas
- statsmodels.api
- matplotlib.pyplot
- matplotlib.backends.backend_tkagg
- tkinter
```

1.3 Author

- Justyna Sarkowicz gr 1 Informatyka i Ekonometria I rok, stacjonarnie

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

projekt	7
-----------------------------------	---

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

projekt.py	15
----------------------------	-------	----

Chapter 4

Namespace Documentation

4.1 projekt Namespace Reference

Functions

- def `regression_model` ()
Creates a linear regression model, by using matplotlib.pyplot and display it on GUI.
- def `boxplots` ()
Creates a boxplots of consumption and GDP.
- def `gdp_time` (country_code)
Create charts of GDP over time for country that user chose.
- def `model_evaluation` ()
Display an interpretation of regression model on GUI.
- def `model_summary` ()
Display the summary of regression model.

Variables

- Users `C` :\\Users\\Justyna\\Desktop\\Projekt_Python>doxypy -a -c "C:/Users/Justyna/Desktop/Projekt_Python/projekt.py"
- `pd consumption_data` = `pd.read_csv('consumption_data.csv')`
- `pd gdp_data` = `pd.read_csv('gdp_data.csv')`
- `pd data` = `pd.merge(consumption_data, gdp_data, on='LOCATION', suffixes=('_consumption', '_gdp'))`
Merging columns by LOCATION.
- `pd X` = `data[['Value_consumption']]`
- `pd y` = `data[['Value_gdp']]`
- `sm model` = `sm.OLS(y, X).fit()`
Using OLS estimator.
- `sm summary` = `model.summary()`
Summary of regression model.
- `Tk root` = `Tk()`
GUI setup.
- `Message M` = `Message(root, padx=10, pady=10, justify='center', font='Helvetica 8')`
Configuration of text.
- `Label L` = `Label(root, font='bold')`

Configuration of label.

- `x`
- `StringVar selected_country = StringVar()`
- `list country_list = list(data.index.values.tolist())`
- `OptionMenu country_menu = OptionMenu(root, selected_country, *country_list)`

Create a drop-down menu to select the country.

- `width`
- `Button btn_gdp_over_time = Button(root, text="GDP over time", width=20, command=lambda↵ : gdp_time(selected_country.get()))`
- `Button btn_regression_model = Button(root, text="Regression model", width=20, command=regression_model)`
- `Button btn_boxplots = Button(root, text="Boxplots", width=20, command=boxplots)`
- `Button btn_model_evaluation = Button(root, text="Model evaluation ", width=20, command=model_evaluation)`
- `Button btn_model_summary = Button(root, text="Model summary", width=20, command=model_summary)`

4.1.1 Function Documentation

4.1.1.1 boxplots()

```
def projekt.boxplots ( )
```

Creates a boxplots of consumption and GDP.

```
00086 def boxplots():
00087     fig = plt.Figure(figsize=(7, 4), dpi=80)
00088     ax = fig.add_subplot(111)
00089     ax.boxplot([data['Value_consumption'], data['Value_gdp']])
00090     ax.set_xticklabels(['Consumption', 'GDP'])
00091     ax.set_ylabel('Dollars')
00092     ax.set_title('Boxplots of consumption and GDP in UE')
00093     canvas = FigureCanvasTkAgg(fig, master=root)
00094     canvas.draw()
00095     canvas.get_tk_widget().place(x=220, y=10)
00096
00097
```

4.1.1.2 gdp_time()

```
def projekt.gdp_time (
    country_code )
```

Create charts of GDP over time for country that user chose.

Parameters

<code>country_code</code>	The selected value of the country code list.
---------------------------	----------------------------------------------

Returns

GDP plot over the time for selected country. If country is not selected it returns nothing.

```
00103 def gdp_time(country_code):
00104     if country_code == 'Select a country':
```

```

00105         return
00106     data_t = pd.read_csv('gdp_data.csv')
00107     country_data = data_t[data_t['LOCATION'] == country_code] # comparing parametr with LOCATION
00108     dates = country_data['TIME'].tolist()
00109     gdp_values = country_data['Value'].tolist()
00110     fig = plt.Figure(figsize=(7, 4), dpi=80)
00111     ax = fig.add_subplot(111)
00112     ax.plot(dates, gdp_values)
00113     ax.set_xlabel('Time')
00114     ax.set_ylabel('GDP')
00115     ax.set_title('GDP in ' + country_code)
00116     canvas = FigureCanvasTkAgg(fig, master=root)
00117     canvas.draw()
00118     canvas.get_tk_widget().place(x=220, y=10)
00119
00120

```

4.1.1.3 model_evaluation()

```
def projekt.model_evaluation ( )
```

Display an interpretation of regression model on GUI.

```

00122 def model_evaluation():
00123     interpretation = "By normalizing the data using Min-Max scaling, the range of the data is
transformed" \
00124         "to a fixed range between 0 and 1. This ensures that no variable dominates the
model" \
00125         "simply because it has larger values than others. This normalization allows for a
fair" \
00126         "comparison between the two variables, Consumption and GDP. The results show that
the" \
00127         "model has an R-squared value of 0.993, which means that 99.3% of the variation
in GDP" \
00128         "can be explained by the variation in consumption. The coefficient of the
Value_consumption" \
00129         " variable is 0.9208, which means that for every unit increase in consumption,
there is" \
00130         " an expected increase of 0.9208 units in GDP. Additionally, the p-value of the"
00131         "
00132         "coefficient of the consumption variable is less than 0.05, indicating that it is
" \
00132         "statistically significant. Therefore, we can conclude that there is a
significant" \
00133         "positive relationship between consumption" \
00134         "and GDP for EU countries. As consumption increases, GDP is expected to increase
as well." \
00135         "However, it is important to note that correlation does not imply causation and
there may" \
00136         "be other factors that are contributing to the relationship observed in the
data."
00137     M.config(text=interpretation, justify='center', font='Helvetica 12')
00138     L.config(text='Evaluation of regression model', font='bold')
00139
00140

```

4.1.1.4 model_summary()

```
def projekt.model_summary ( )
```

Display the summary of regression model.

```

00142 def model_summary():
00143     M.config(text=summary, font='Helvetica 8')
00144     L.config(text='Regression model summary')
00145
00146

```

4.1.1.5 regression_model()

```
def projekt.regression_model ( )
```

Creates a linear regression model, by using matplotlib.pyplot and display it on GUI.

The data are normalized and grouped.

```
00071 def regression_model():
00072     fig = plt.Figure(figsize=(7, 4), dpi=80) # defining the chart size
00073     ax = fig.add_subplot(111) # return an object in the grid
00074     ax.scatter(X['Value_consumption'], y) # a scatter plot
00075     ax.plot(X['Value_consumption'], model.predict(X), color='pink') # creating regression model
00076     ax.set_xlabel('Consumption')
00077     ax.set_ylabel('GDP')
00078     ax.set_title('Linear regression model of GPP and consumption in UE countries')
00079     # creating a canvas object
00080     canvas = FigureCanvasTkAgg(fig, master=root)
00081     canvas.draw()
00082     canvas.get_tk_widget().place(x=220, y=10)
00083
00084
```

4.1.2 Variable Documentation

4.1.2.1 btn_boxplots

```
Button projekt.btn_boxplots = Button(root, text="Boxplots", width=20, command=boxplots)
```

4.1.2.2 btn_gdp_over_time

```
Button projekt.btn_gdp_over_time = Button(root, text="GDP over time", width=20, command=lambda↵
: gdp_time(selected_country.get()))
```

4.1.2.3 btn_model_evaluation

```
Button projekt.btn_model_evaluation = Button(root, text="Model evaluation ", width=20, command=model_evaluation)
```

4.1.2.4 btn_model_summary

```
Button projekt.btn_model_summary = Button(root, text="Model summary", width=20, command=model_summary)
```

4.1.2.5 btn_regression_model

```
Button projekt.btn_regression_model = Button(root, text="Regression model", width=20, command=regression_model)
```

4.1.2.6 C

```
Users projekt.C : \Users\Justyna\Desktop\Projekt_Python>doxypypy -a -c "C:/Users/Justyna/Desktop/Projekt_Python/projekt.py"
```

4.1.2.7 consumption_data

```
pd projekt.consumption_data = pd.read_csv('consumption_data.csv')
```

4.1.2.8 country_list

```
list projekt.country_list = list(data.index.values.tolist())
```

4.1.2.9 country_menu

```
OptionMenu projekt.country_menu = OptionMenu(root, selected_country, *country_list)
```

Create a drop-down menu to select the country.

4.1.2.10 data

```
pd projekt.data = pd.merge(consumption_data, gdp_data, on='LOCATION', suffixes=('_consumption', '_gdp'))
```

Merging columns by LOCATION.

4.1.2.11 gdp_data

```
pd projekt.gdp_data = pd.read_csv('gdp_data.csv')
```

4.1.2.12 L

```
Label projekt.L = Label(root, font='bold')
```

Configuration of label.

4.1.2.13 M

```
Message projekt.M = Message(root, padx=10, pady=10, justify='center', font='Helvetica 8')
```

Configuration of text.

4.1.2.14 model

```
sm projekt.model = sm.OLS(y, X).fit()
```

Using OLS estimator.

4.1.2.15 root

```
Tk projekt.root = Tk()
```

GUI setup.

4.1.2.16 selected_country

```
StringVar projekt.selected_country = StringVar()
```

4.1.2.17 summary

```
sm projekt.summary = model.summary()
```

Summary of regression model.

4.1.2.18 width

```
projekt.width
```

4.1.2.19 X

```
sm projekt.X = data[['Value_consumption']]
```

4.1.2.20 x

```
projekt.x
```

4.1.2.21 y

```
projekt.y = data['Value_gdp']
```


Chapter 5

File Documentation

5.1 projekt.py File Reference

Namespaces

- namespace `projekt`

Functions

- def `projekt.regression_model ()`
Creates a linear regression model, by using matplotlib.pyplot and display it on GUI.
- def `projekt.boxplots ()`
Creates a boxplots of consumption and GDP.
- def `projekt.gdp_time (country_code)`
Create charts of GDP over time for country that user chose.
- def `projekt.model_evaluation ()`
Display an interpretation of regression model on GUI.
- def `projekt.model_summary ()`
Display the summary of regression model.

Variables

- Users `projekt.C` :Users\Justyna\Desktop\Projekt_Python>doxypypy -a -c "C:/Users/Justyna/Desktop/Projekt_Python/projekt.py"
- pd `projekt.consumption_data` = pd.read_csv('consumption_data.csv')
- pd `projekt.gdp_data` = pd.read_csv('gdp_data.csv')
- pd `projekt.data` = pd.merge(consumption_data, gdp_data, on='LOCATION', suffixes=('_consumption', '_gdp'))
Merging columns by LOCATION.
- pd `projekt.X` = data[['Value_consumption']]
- pd `projekt.y` = data['Value_gdp']
- sm `projekt.model` = sm.OLS(y, X).fit()
Using OLS estimator.
- sm `projekt.summary` = model.summary()
Summary of regression model.
- Tk `projekt.root` = Tk()

GUI setup.

- Message `projekt.M` = Message(root, padx=10, pady=10, justify='center', font='Helvetica 8')

Configuration of text.

- Label `projekt.L` = Label(root, font='bold')

Configuration of label.

- `projekt.x`
- StringVar `projekt.selected_country` = StringVar()
- list `projekt.country_list` = list(data.index.values.tolist())
- OptionMenu `projekt.country_menu` = OptionMenu(root, selected_country, *country_list)

Create a drop-down menu to select the country.

- `projekt.width`
- Button `projekt.btn_gdp_over_time` = Button(root, text="GDP over time", width=20, command=lambda: gdp↵_time(selected_country.get()))
- Button `projekt.btn_regression_model` = Button(root, text="Regression model", width=20, command=regression↵_model)
- Button `projekt.btn_boxplots` = Button(root, text="Boxplots", width=20, command=boxplots)
- Button `projekt.btn_model_evaluation` = Button(root, text="Model evaluation ", width=20, command=model_↵evaluation)
- Button `projekt.btn_model_summary` = Button(root, text="Model summary", width=20, command=model_↵summary)

Index

- boxplots
 - projekt, [8](#)
- btn_boxplots
 - projekt, [10](#)
- btn_gdp_over_time
 - projekt, [10](#)
- btn_model_evaluation
 - projekt, [10](#)
- btn_model_summary
 - projekt, [10](#)
- btn_regression_model
 - projekt, [10](#)
- C
 - projekt, [11](#)
- consumption_data
 - projekt, [11](#)
- country_list
 - projekt, [11](#)
- country_menu
 - projekt, [11](#)
- data
 - projekt, [11](#)
- gdp_data
 - projekt, [11](#)
- gdp_time
 - projekt, [8](#)
- L
 - projekt, [11](#)
- M
 - projekt, [12](#)
- model
 - projekt, [12](#)
- model_evaluation
 - projekt, [9](#)
- model_summary
 - projekt, [9](#)
- projekt, [7](#)
 - boxplots, [8](#)
 - btn_boxplots, [10](#)
 - btn_gdp_over_time, [10](#)
 - btn_model_evaluation, [10](#)
 - btn_model_summary, [10](#)
 - btn_regression_model, [10](#)
 - C, [11](#)
 - consumption_data, [11](#)
 - country_list, [11](#)
 - country_menu, [11](#)
 - data, [11](#)
 - gdp_data, [11](#)
 - gdp_time, [8](#)
 - L, [11](#)
 - M, [12](#)
 - model, [12](#)
 - model_evaluation, [9](#)
 - model_summary, [9](#)
 - regression_model, [9](#)
 - root, [12](#)
 - selected_country, [12](#)
 - summary, [12](#)
 - width, [12](#)
 - X, [13](#)
 - x, [13](#)
 - y, [13](#)
- projekt.py, [15](#)
- regression_model
 - projekt, [9](#)
- root
 - projekt, [12](#)
- selected_country
 - projekt, [12](#)
- summary
 - projekt, [12](#)
- width
 - projekt, [12](#)
- X
 - projekt, [13](#)
- x
 - projekt, [13](#)
- y
 - projekt, [13](#)