Assignment 4

Before working on this assignment please read these instructions fully. In the submission area, you will notice that you can click the link to **Preview the Grading** for each step of the assignment. This is the criteria that will be used for peer grading. Please familiarize yourself with the criteria before beginning the assignment.

This assignment requires that you to find **at least** two datasets on the web which are related, and that you visualize these datasets to answer a question with the broad topic of **economic activity or measures** (see below) for the region of **None, None, Singapore**, or **Singapore** more broadly.

You can merge these datasets with data from different regions if you like! For instance, you might want to compare **None, None, Singapore** to Ann Arbor, USA. In that case at least one source file must be about **None, None, Singapore**.

You are welcome to choose datasets at your discretion, but keep in mind **they will be shared with your peers**, so choose appropriate datasets. Sensitive, confidential, illicit, and proprietary materials are not good choices for datasets for this assignment. You are welcome to upload datasets of your own as well, and link to them using a third party repository such as github, bitbucket, pastebin, etc. Please be aware of the Coursera terms of service with respect to intellectual property.

Also, you are welcome to preserve data in its original language, but for the purposes of grading you should provide english translations. You are welcome to provide multiple visuals in different languages if you would like!

As this assignment is for the whole course, you must incorporate principles discussed in the first week, such as having as high data-ink ratio (Tufte) and aligning with Cairo's principles of truth, beauty, function, and insight.

Here are the assignment instructions:

- State the region and the domain category that your data sets are about (e.g., **None, None, Singapore** and **economic activity or measures**).
- You must state a question about the domain category and region that you identified as being interesting.
- You must provide at least two links to available datasets. These could be links to files such as CSV or Excel files, or links to websites which might have data in tabular form, such as Wikipedia pages.
- You must upload an image which addresses the research question you stated. In addition to addressing the question, this visual should follow Cairo's principles of truthfulness, functionality, beauty, and insightfulness.
- You must contribute a short (1-2 paragraph) written justification of how your visualization addresses your stated research question.

What do we mean by **economic activity or measures**? For this category you might look at the inputs or outputs to the given economy, or major changes in the economy compared to other regions.

Tips

- Wikipedia is an excellent source of data, and I strongly encourage you to explore it for new data sources.
- Many governments run open data initiatives at the city, region, and country levels, and these are wonderful resources for localized data sources.
- Several international agencies, such as the United Nations (http://data.un.org/), the World Bank (http://data.worldbank.org/), the Global Open Data Index (http://index.okfn.org/place/) are other great

places to look for data.
- This assignment requires you to convert and clean datafiles. Check out the discussion forums for tips on how to do this from various sources, and share your successes with your fellow students!

Example

Looking for an example? Here's what our course assistant put together for the **Ann Arbor, MI, USA** area using **sports and athletics** as the topic. Example Solution File (./readonly/Assignment4_example.pdf)

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt

%matplotlib notebook
plt.style.use('seaborn-colorblind')
```

In [2]:

```
pd.read_csv('balance.csv').head()
```

Out[2]:

1987 1Q	1987 2Q	1987 3Q	1987 4Q	1988 1Q	...	2014 3Q	2014 4Q	2015 1Q
01.7	529.8	671.8	925.2	1,057.60	...	3,405.20	110.5	-1,310.90
659.2	261.5	110	70.7	628.9	...	23,731.80	21,976.70	19,624.30
1,041.10	-315	-403.7	-650.1	-117.6	...	29,017.60	28,281.50	30,422.80
2,591.40	14,832.40	16,324.90	17,454.90	17,658.80	...	141,828.20	138,277.40	130,180.50
3,632.50	15,147.40	16,728.60	18,105	17,776.40	...	112,810.60	109,995.90	99,757.70

In [3]:

```
list(pd.read_csv('balance.csv').columns.values)
```

```
 ' 1986 4Q ',
 ' 1987 1Q ',
 ' 1987 2Q ',
 ' 1987 3Q ',
 ' 1987 4Q ',
 ' 1988 1Q ',
 ' 1988 2Q ',
 ' 1988 3Q ',
 ' 1988 4Q ',
 ' 1989 1Q ',
 ' 1989 2Q ',
 ' 1989 3Q ',
 ' 1989 4Q ',
 ' 1990 1Q ',
 ' 1990 2Q ',
 ' 1990 3Q ',
 ' 1990 4Q ',
 ' 1991 1Q ',
 ' 1991 2Q ',
 ' 1991 3Q ',
```

In [4]:

```
pd.read_csv('balance.csv', index_col=' Variables ').transpose().columns.values
```

Out[4]:

```
array([' D Overall Balance (A-B+C) ', '     A Current Account Balance
 ',
       '         Goods Balance ', '             Exports Of Goods ',
       '             Imports Of Goods ', '         Services Balance ',
       '             Exports Of Services ',
       '                 Maintenance And Repair Services ',
       '                 Transport ', '                 Travel ',
       '                 Insurance ',
       '                 Government Goods And Services ',
       '                 Construction ', '                 Financial
 ',
       '                 Telecommunications, Computer & Information ',
       '                 Charges For The Use Of Intellectual Property
 ',
       '                 Personal, Cultural And Recreational ',
       '                 Other Business Services ',
       '             Imports Of Services ',
       '                 Maintenance And Repair Services ',
       '                 Transport ', '                 Travel ',
       '                 Insurance ',
       '                 Government Goods And Services ',
       '                 Construction ', '                 Financial
 ',
       '                 Telecommunications, Computer & Information ',
       '                 Charges For The Use Of Intellectual Property
 ',
       '                 Personal, Cultural And Recreational ',
       '                 Other Business Services ',
       '         Primary Income Balance ',
       '             Primary Income Receipts ',
       '             Primary Income Payments ',
       '         Secondary Income Balance ',
       '             General Government (Net) ',
       '             Other Sectors (Net) ',
       '     B Capital & Financial Account Balance ',
       '         Financial Account (Net) ',
       '             Direct Investment ', '                 Assets ',
       '                 Liabilities ',
       '             Portfolio Investment ', '                 Assets
 ',
       '                     Deposit-taking Corporations, Except The C
entral Bank ',
       '                     Official ', '                     Others
 ',
       '                 Liabilities ',
       '                     Deposit-taking Corporations, Except The C
entral Bank ',
       '                     Others ',
       '             Financial Derivatives ', '                 Assets
',
       '                 Liabilities ', '             Other Investment
',
       '                 Assets ',
       '                     Deposit-taking Corporations, Except The C
entral Bank ',
       '                     Official ', '                     Others
```

```
                                Official ',                                Others
 ',
       '                 Liabilities ',
       '                     Deposit-taking Corporations, Except The C
entral Bank ',
       '                     Others ', '     C Net Errors & Omissions
 ',
       ' E Reserve Assets ', '     Special Drawing Rights ',
       '     Reserves Position In The IMF ',
       '     Foreign Exchange Assets '], dtype=object)
```

In [5]:

```
import locale
from locale import atof

locale.setlocale(locale.LC_NUMERIC, '')
```

Out[5]:

```
'en_US.UTF-8'
```

In [6]:

```
df_balance = pd.read_csv('balance.csv', index_col=' Variables ', thousands=',').tra
df_balance = df_balance[[' D Overall Balance (A-B+C) ', '         Goods Balance ',
df_balance.head()
```

Out[6]:

Variables	D Overall Balance (A-B+C)	Goods Balance	Services Balance
1986 1Q	33.9	-1044.30	526.2
1986 2Q	508	-7.6	880
1986 3Q	206.6	-442.9	872.6
1986 4Q	460.1	-554.2	847.9
1987 1Q	201.7	-1041.10	698.9

In [7]:

```
df_balance = df_balance.astype(float)
```

In [8]:

```
pd.read_csv('gdp.csv').head()
```

Out[8]:

	Variables	1975 1Q	1975 2Q	1975 3Q	1975 4Q	1976 1Q	1976 2Q	1976 3Q	1976
0	Gross Domestic Product At 2010 Market Prices	7,052.10	7,154.80	7,321.10	7,435.20	7,632.50	7,683.90	7,846.40	7,94
1	Goods Producing Industries	2,017.50	2,034.90	2,202.60	2,260.20	2,334.50	2,336.80	2,407.30	2,38
2	Manufacturing	1,308.90	1,303.30	1,441.80	1,462.90	1,520.50	1,504.40	1,565.60	1,58
3	Construction	508.9	533	559	591.8	605	622.4	626.6	576.
4	Utilities	130.2	131.3	135.7	138.3	139.6	142.7	146.1	149.

5 rows × 169 columns

In [9]:

```
pd.read_csv('gdp.csv', index_col=' Variables ').transpose().columns.values
```

Out[9]:

```
array([' Gross Domestic Product At 2010 Market Prices ',
       '     Goods Producing Industries ', '         Manufacturing ',
       '         Construction ', '         Utilities ',
       '         Other Goods Industries ',
       '     Services Producing Industries ',
       '         Wholesale & Retail Trade ',
       '         Transportation & Storage ',
       '         Accommodation & Food Services ',
       '         Information & Communications ',
       '         Finance & Insurance ', '         Business Services ',
       '         Other Services Industries ',
       '     Ownership Of Dwellings ',
       '     Gross Value Added At Basic Prices ',
       '     Add: Taxes On Products '], dtype=object)
```

In [10]:

```
df_gdp = pd.read_csv('gdp.csv', index_col=' Variables ', thousands=',').transpose()
df_gdp.head()
```

Out[10]:

Variables	Gross Domestic Product At 2010 Market Prices	Goods Producing Industries	Manufacturing	Construction	Utilities	Other Goods Industries	Se Prod Indu
1975 1Q	7052.1	2017.5	1308.9	508.9	130.2	83.6	4133
1975 2Q	7154.8	2034.9	1303.3	533.0	131.3	85.8	4214
1975 3Q	7321.1	2202.6	1441.8	559.0	135.7	87.9	4198
1975 4Q	7435.2	2260.2	1462.9	591.8	138.3	93.1	4232
1976 1Q	7632.5	2334.5	1520.5	605.0	139.6	96.2	4349

In [11]:

```
df_gdp = df_gdp[[' Gross Domestic Product At 2010 Market Prices ', '     Goods Prod
df_gdp = df_gdp[df_gdp.index>' 1986']
df_gdp.head()
```

Out[11]:

Variables	Gross Domestic Product At 2010 Market Prices	Goods Producing Industries	Services Producing Industries
1986 1Q	14997.8	4006.1	9007.6
1986 2Q	15240.1	4149.5	9059.1
1986 3Q	15479.2	4121.0	9288.4
1986 4Q	15909.4	4207.7	9604.3
1987 1Q	16227.3	4268.2	9805.5

In [12]:

```
all_data = df_gdp.join(df_balance)
all_data.head()
```

Out[12]:

Variables	Gross Domestic Product At 2010 Market Prices	Goods Producing Industries	Services Producing Industries	D Overall Balance (A-B+C)	Goods Balance	Services Balance
1986 1Q	14997.8	4006.1	9007.6	33.9	-1044.3	526.2
1986 2Q	15240.1	4149.5	9059.1	508.0	-7.6	880.0
1986 3Q	15479.2	4121.0	9288.4	206.6	-442.9	872.6
1986 4Q	15909.4	4207.7	9604.3	460.1	-554.2	847.9
1987 1Q	16227.3	4268.2	9805.5	201.7	-1041.1	698.9

In [13]:

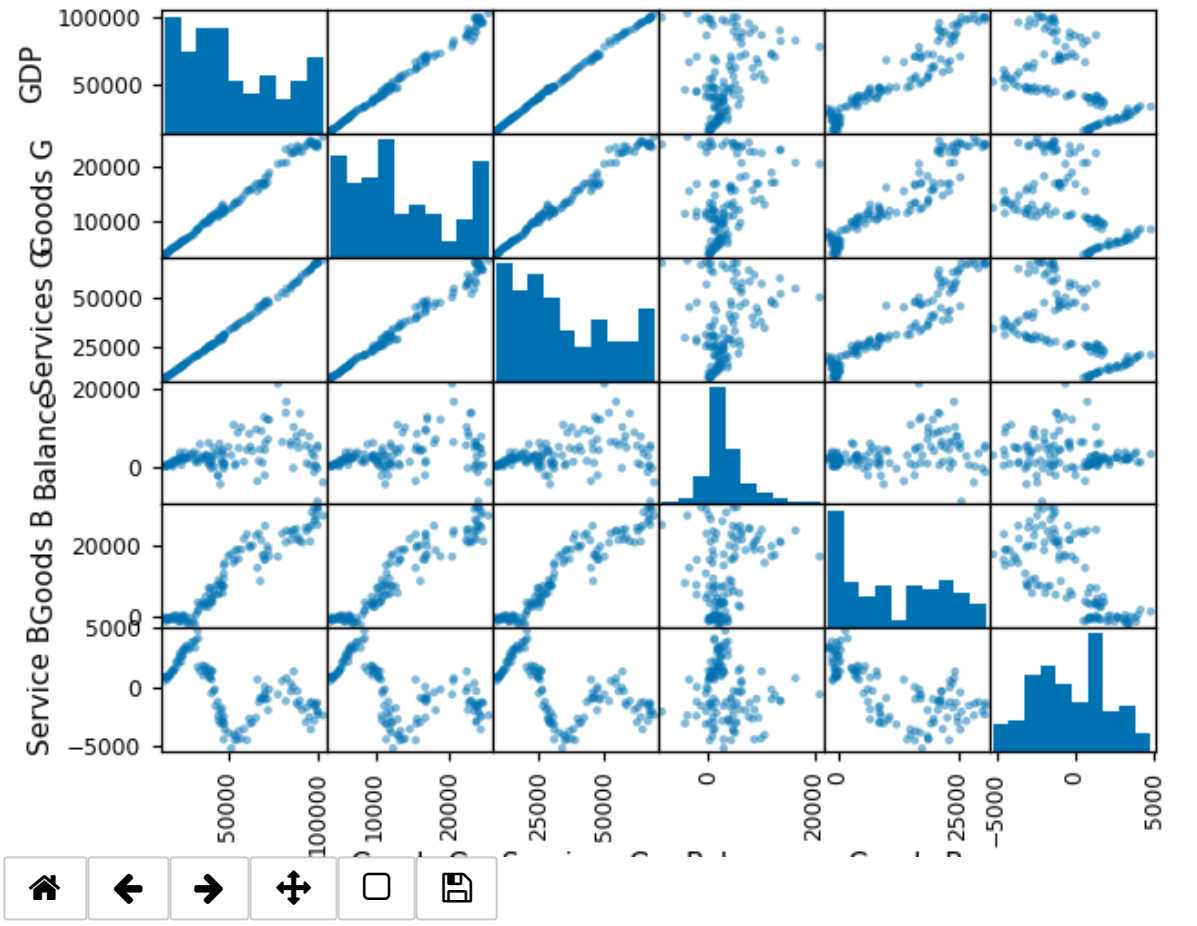
```
all_data.rename(columns = {' Gross Domestic Product At 2010 Market Prices ': 'GDP',
                           '     Goods Producing Industries ': 'Goods G',
                           '     Services Producing Industries ': 'Services G',
                           ' D Overall Balance (A-B+C) ': 'Balance',
                           '         Goods Balance ': 'Goods B',
                           '         Services Balance ': 'Service B'}, inplace = Tr
```

In [14]:

```
pd.tools.plotting.scatter_matrix(all_data)
```

```
/home/gokul/anaconda2/lib/python2.7/site-packages/ipykernel_launcher.p
y:1: FutureWarning: 'pandas.tools.plotting.scatter_matrix' is deprecat
ed, import 'pandas.plotting.scatter_matrix' instead.
  """Entry point for launching an IPython kernel.
```

Figure 1



Out[14]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fb9be6de45
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bb36b49
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bb26f65
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bb2ab2d
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bb1dbc9
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bb162d9
0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bb0d2d5
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bb058d9
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bafca49
```

```
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9baf5265
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9baeb8fd
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bae4a29
0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fb9baed235
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bad3675
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bacbd79
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bac2eb5
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9babb59d
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bab8be1
0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fb9bab1c7d
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9baa83e9
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9baa0af9
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba97da1
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba903d9
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba94369
0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba7fcad
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba78395
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba6f565
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba67c4d
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba660d5
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba5e9d9
0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba55d49
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba4e365
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba449fd
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba3dd2d
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba463cd
0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb9ba2c8f9
0>]], dtype=object)
```

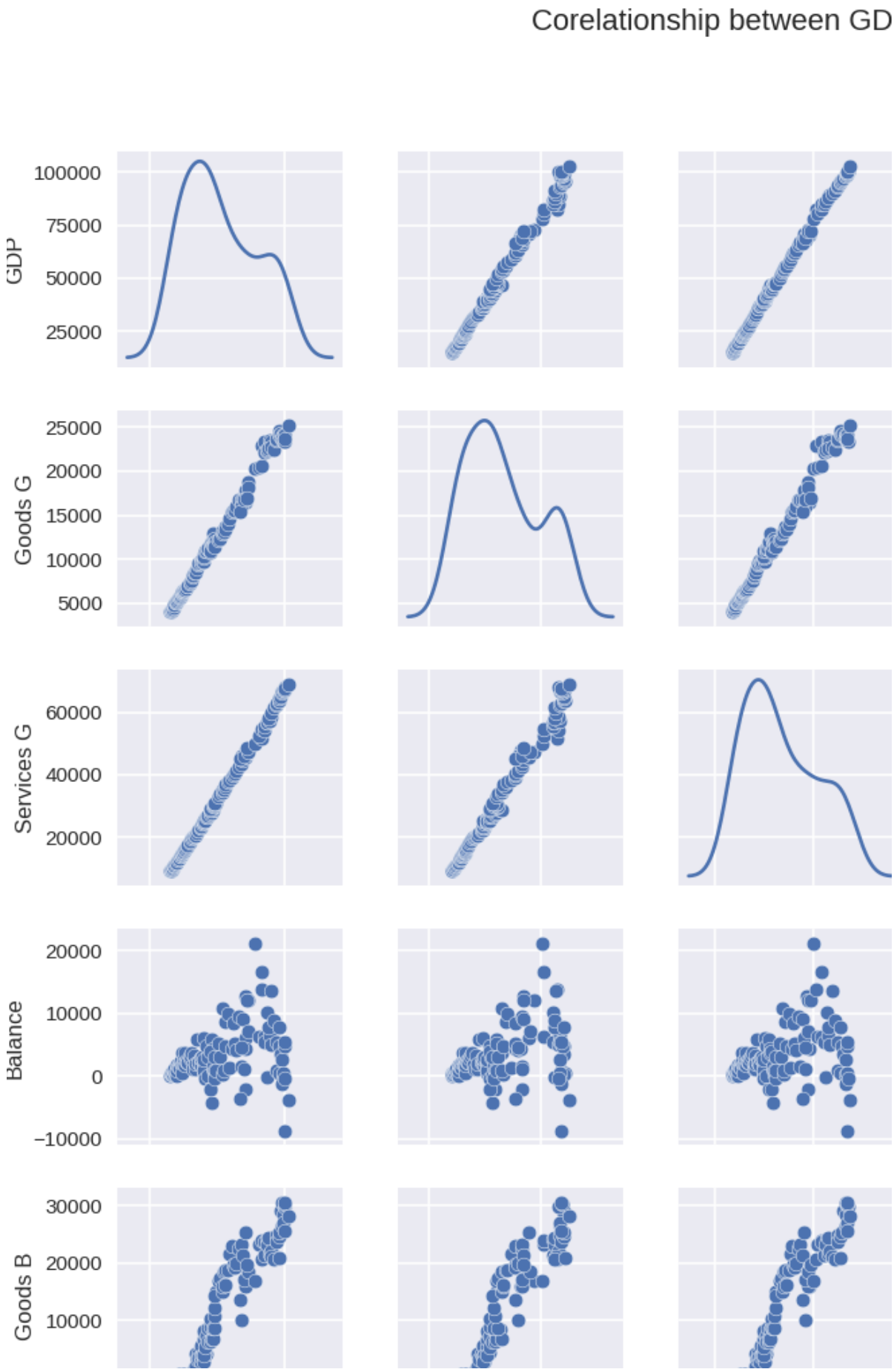
In [15]:

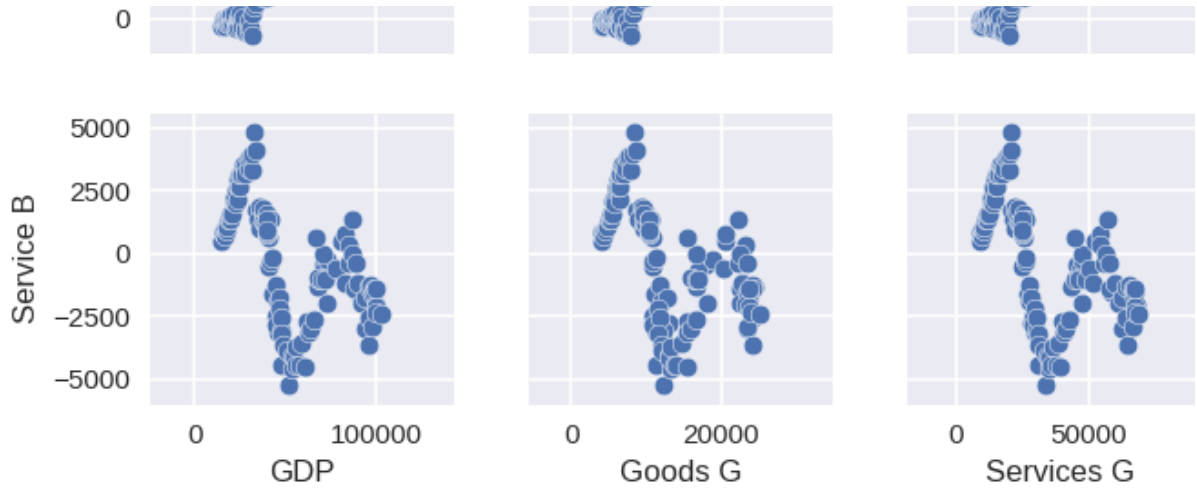
```
import seaborn as sns
```

In [16]:

```
g = sns.pairplot(all_data, diag_kind='kde', size=2);
```

Figure





In [17]:

```
plt.subplots_adjust(top=0.9)
g.fig.suptitle('Corelationship between GDP and Balance of Singapore')
```

Out[17]:

```
<matplotlib.text.Text at 0x7fb9a6234590>
```

In [ ]:

```

```