Emerging Markets' Appetite for Sovereign Risk: Exploring Relationship between EM GDP Output, Commodity Prices, and Sovereign Risk

Introduction

Emerging Markets' GDPs, measured for Economic Output, are affected by fluctuations not just within commodity prices, but also by country risk, and sovereign risk factors. Historically, commodities have been key to the rise of Emerging Markets. However, for some Emerging Markets, commodities have also been a catalyst for economic recessions within such economies. Emerging Markets are also subjected to a variety of political, economic, and financial risks, that all affect their economic output and their production of commodities, as well as commodity prices. This impacts EMs' ability to service their own consumers and consumers within industrialized countries.

To explore such relationship, a series of multivariate regression analyses were used. The dependent variable was the GDP of EM countries based on current US dollar prices, while the independent variables were:Commodity Prices, Political Risk, Sovereign Risk, Country Risk, Exports of Commodities by the selected EM countries, Imports of Commodities by the selected EM countries, and Imports of Commodities by the selected Industrialized Economies from 1985-2016. Such exports and imports were chosen as other explanatory variables that affect greatly affect commodity prices, hence their inclusion were important to consider when assessing Political, Country, and Sovereign Risks' ability to affect Commodity Prices and GDP. The list of Emerging Market countries examined were based off the MSCI index, while the list of Industrialized Economies came from the World Population Review. While the original MSCI index contains 26 EM countries, this project only involved 20 EM economies. For one, a few countries in the MSCI were also considered developed economies according to the World Population Review, in which those that were considered Developed Economies were removed from the EM list in this project. In addition, a few of the EM countries in the MSCI list did not contain enough data involving the aforementioned risk variables, hence those EMs were removed from consideration and thus the project only deals with 20 EM countries.

The risk variables explored came from data collected from The International Country Risk Guide(ICRG) because of it's comprehensive historical data of various financial, economic, and political risk data dating back to 1985, thus resulting in all other variables used in this model to have data stemming from 1985. Country Risk, as defined per https://www.investopedia.com/terms/c/countryrisk.asp (https://www.investopedia.com/terms/c/countryrisk.asp), includes the combination of political and economic risk. In order to accurately reflect country risk variable, the methodology used to calculate ICRG's composite rating which includes all three financial, political, and economic risk variables was used to calculate country risk. To sum, the three risk indicators that will be featured in these analyses are: Political Risk, Country Risk, and Sovereign Risk.

Sources

Data for GDP=https://www.imf.org/external/pubs/ft/weo/2019/01/weodata/download.aspx (https://www.imf.org/external/pubs/ft/weo/2019/01/weodata/download.aspx)

Data for Political, Economic, and Sovereign Risk data,ICRG Methodology and definition of variables used=https://www.prsgroup.com/wp-content/uploads/2012/11/icrgmethodology.pdf (https://www.prsgroup.com/wp-content/uploads/2012/11/icrgmethodology.pdf)

Data for Commodity Prices=https://unctadstat.unctad.org/wds/TableViewer/tableView.aspx?ReportId=30728 (https://unctadstat.unctad.org/wds/TableViewer/tableView.aspx?ReportId=30728)

Imports and Exports of Commodities=http://data.imf.org/?sk=2CDDCCB8-0B59-43E9-B6A0-59210D5605D2 (http://data.imf.org/?sk=2CDDCCB8-0B59-43E9-B6A0-59210D5605D2)

List for Industrialized Countries=http://worldpopulationreview.com/countries/developed-countries/ (http://worldpopulationreview.com/countries/developed-countries/)

Context for why imports of commodities by industrialized countries and exports of commodities by emerging markets were selected for this model= http://www.carmenreinhart.com/user_uploads/BKRW_OP_1994.pdf (http://www.carmenreinhart.com/user_uploads/BKRW_OP_1994.pdf),

http://www.carmenreinhart.com/user_uploads/BR_IMFSP_1994.pdf (http://www.carmenreinhart.com/user_uploads/BR_IMFSP_1994.pdf)

List for EM countries used=https://www.investopedia.com/terms/e/emergingmarketsindex.asp (https://www.investopedia.com/terms/e/emergingmarketsindex.asp)

Appendix

GDP= GDP dataset imported

EMCountryList=all EMs that will be used in this model

EMGDP=GDP output of Emerging Markets

CMPrices=dataset imported

CMPrices=list of value needed [Price Index all commodity groups]

CMdata1...CMdata2= separate data frames later to be concated.

CMPriceData=All commodity yearly price data from 1984-2016

FR=Financial Rating from ICRG dataset

SovereignRisk= FR data in ICRG Dataset

ER=Economic Risk from ICRG dataset

PR=Political RIsk from ICRG Dataset

Country Risk=[ER+PR]*0.5

RiskIndicators=Political Risk, Sovereign Risk, and Country Risk

GDPvPrices=Dataframe used to plot GDP vs Price data and correlation

PriceRisk=Dataframe used to plot various price vs risk indicators data

IndustrializedNations=list of industrialized Economies used in this model

IMFCountryList=all of the countries in the EMCountryList

IEComImports=Imports of Commodities by Industrialized Economies

EMComExports= Exports of Commodities by Emerging Markets

EMComImports= Imports of Commodities by Emerging Markets

PriceIMEX=Commodity price data and import and export data

GDPRisk=EM GDP data and Risk Indicator Data

GDPvPrices= GDP Data and Commodity Prices, dataframe used to reflect change in all relevant variables when taking switching China and BRICS in and out of the equation

PriceRiskXIM=dataframe used to reflect change in all relevant variables when taking switching China and BRICS in and out of the equation

INDGDPPriceRisk= variables related to India analysis

ARGGDPPriceRisk= variables related to Argentina analysis

CHINAGDPPriceRisk=variables related to China analysis.

ChinaML1= China Machine Learning data frame looking at how Economic Risk impacts GDP

ChinaML2= China Machine Learning dataframe looking at how Economic Risk impacts Sovereign Risk

IndiaML= India Machine Learning data frame looking at how Country Risk impacts GDP

ArgentinaML= Argentina Machine Learning data frame looking at how Sovereign Risk impacts GDP

EMGDPML= Emerging Markets Machine Learning data frame looking at how Sovereign Risk impacts EM's GDP

EMML= Emerging Markets Machine Learning data frame looking at how Sovereign Risk impacts Commodity Prices

Data Import and Cleaning

Data was imported and cleaned in order to create new dataframes. Once the Data Frames were created, simpoe plot graphs were created among the independent and dependent variables examined.

In [623]:
```
# Data imported for the following EMs, looked at.

import pandas as pd
df1=pd.read_excel (r"C:\New folder\WEO data.xlsx")
import numpy as np

df1

EMCountryList=["Argentina",'Brazil','Chile','China','Colombia','Egypt','Hungar
y','India','Indonesia','Malaysia','Mexico','Pakistan','Peru','Philippines','Po
land','Qatar','Russia','Saudi Arabia','South Africa','Thailand']

GDP=df1.loc[df1["Country"].isin(EMCountryList)]

GDP=GDP.loc[GDP['Units'].str.contains('U.S. dollars')]

GDP=GDP.loc[GDP['Subject Descriptor'].str.contains('Gross domestic product, cu
rrent prices')]

GDP
```

Out[623]:

	WEO Country Code	ISO	WEO Subject Code	Country	Subject Descriptor	Subject Notes	Units	Scale	Country/Series-specific Notes
228	213	ARG	NGDPD	Argentina	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
1038	223	BRA	NGDPD	Brazil	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
1533	228	CHL	NGDPD	Chile	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
1578	924	CHN	NGDPD	China	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
1623	233	COL	NGDPD	Colombia	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
2253	469	EGY	NGDPD	Egypt	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
3333	944	HUN	NGDPD	Hungary	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
3423	534	IND	NGDPD	India	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
3468	536	IDN	NGDPD	Indonesia	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...

	WEO Country Code	ISO	WEO Subject Code	Country	Subject Descriptor	Subject Notes	Units	Scale	Country/Series-specific Notes
4683	548	MYS	NGDPD	Malaysia	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
4998	273	MEX	NGDPD	Mexico	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
5853	564	PAK	NGDPD	Pakistan	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
6078	293	PER	NGDPD	Peru	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
6123	566	PHL	NGDPD	Philippines	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
6168	964	POL	NGDPD	Poland	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
6303	453	QAT	NGDPD	Qatar	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
6393	922	RUS	NGDPD	Russia	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
6618	456	SAU	NGDPD	Saudi Arabia	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...

	WEO Country Code	ISO	WEO Subject Code	Country	Subject Descriptor	Subject Notes	Units	Scale	Country/Series-specific Notes
7068	199	ZAF	NGDPD	South Africa	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...
7743	578	THA	NGDPD	Thailand	Gross domestic product, current prices	Values are based upon GDP in national currency...	U.S. dollars	Billions	See notes for: Gross domestic product, curren...

20 rows × 55 columns

In [624]:
```
#GDP data was then merged to create an average among the EMs examined per each
year between 1985 to 2016.

data1={'Year':['1985','1986','1987','1988','1989'],'GDP':[GDP.loc[:,1985].mean
().round(2),GDP.loc[:,1986].mean().round(2),GDP.loc[:,1987].mean().round(2),GD
P.loc[:,1988].mean().round(2),GDP.loc[:,1989].mean().round(2)]}
EMGDP1=pd.DataFrame(data1)

EMGDP1

data2={'Year':['1990','1991','1992','1993','1994','1995'],'GDP':[GDP.loc[:,199
0].mean().round(2),GDP.loc[:,1991].mean().round(2),GDP.loc[:,1992].mean().roun
d(2),GDP.loc[:,1993].mean().round(2),GDP.loc[:,1994].mean().round(2),GDP.loc
[:,1995].mean().round(2)]}
EMGDP2=pd.DataFrame(data2)
EMGDP2

data3={'Year':['1996','1997','1998','1999','2000','2001'],'GDP':[GDP.loc[:,199
6].mean().round(2),GDP.loc[:,1997].mean().round(2),GDP.loc[:,1998].mean().roun
d(2),GDP.loc[:,1999].mean().round(2),GDP.loc[:,2000].mean().round(2),GDP.loc
[:,2001].mean().round(2)]}
EMGDP3=pd.DataFrame(data3)
EMGDP3

data4={'Year':['2002','2003','2004','2005','2006','2007'],'GDP':[GDP.loc[:,200
2].mean().round(2),GDP.loc[:,2003].mean().round(2),GDP.loc[:,2004].mean().roun
d(2),GDP.loc[:,2005].mean().round(2),GDP.loc[:,2006].mean().round(2),GDP.loc
[:,2007].mean().round(2)]}
EMGDP4=pd.DataFrame(data4)
EMGDP4

data5={'Year':['2008','2009','2010','2011','2012','2013'],'GDP':[GDP.loc[:,200
8].mean().round(2),GDP.loc[:,2009].mean().round(2),GDP.loc[:,2010].mean().roun
d(2),GDP.loc[:,2011].mean().round(2),GDP.loc[:,2012].mean().round(2),GDP.loc
[:,2013].mean().round(2)]}
EMGDP5=pd.DataFrame(data5)
EMGDP5

data6={'Year':['2014','2015','2016'],'GDP':[GDP.loc[:,2014].mean().round(2),GD
P.loc[:,2015].mean().round(2),GDP.loc[:,2016].mean().round(2)]}
EMGDP6=pd.DataFrame(data6)
EMGDP6

EMGDP=pd.concat([EMGDP1,EMGDP2,EMGDP3,EMGDP4,EMGDP5,EMGDP6])


EMGDP
```

Out[624]:

	Year	GDP
0	1985	91.49
1	1986	91.53
2	1987	100.36
3	1988	113.91
4	1989	126.86
0	1990	134.15
1	1991	137.78
2	1992	149.26
3	1993	171.26
4	1994	190.53
5	1995	216.39
0	1996	241.07
1	1997	254.93
2	1998	239.29
3	1999	233.89
4	2000	255.18
5	2001	260.65
0	2002	263.93
1	2003	295.30
2	2004	348.16
3	2005	414.25
4	2006	491.49
5	2007	607.78
0	2008	726.82
1	2009	711.14
2	2010	867.39
3	2011	1032.07
4	2012	1100.41
5	2013	1171.20
0	2014	1213.92
1	2015	1161.19
2	2016	1160.00

In [625]:
```
CMPrices=pd.read_excel (r"C:\New folder\us_annualcommoditypriceindicesaverages
_72693818609729.xls")
CMPrices
```

Out[625]:

	Free market commodity price indices, annual, 1960 - 2016 (Discontinued)	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnam
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
2	MEASURE	PRICE INDICES 2000=100	NaN	NaN	NaN	NaN	NaN	N
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
4	PERIOD	1960	1961	1962	1963	1964	1965	19
5	PRODUCT	NaN	NaN	NaN	NaN	NaN	NaN	N
6	Price index - All groups (in terms of current ...	44.3333	43.3833	43.8167	50.2333	50.6667	49.2417	51.7
7	Price Index - All groups (in terms of constant...	182.383	174.192	174.525	200.033	198.733	188.542	193.
8	Price index - All groups (in terms of SDRs)	58.6167	57.3833	57.9667	66.425	66.9917	65.1	6
9	ALL FOOD	46.7417	47.1167	48.5417	59.325	56.2083	50.1917	52.5
10	- FOOD AND TROPICAL BEVERAGES	45.0417	44.775	47.275	58.9917	55.1167	47.3917	50.6
11	- - FOOD	44.15	44.3833	47.275	60.0167	54.775	46.7333	50.1
12	- - TROPICAL BEVERAGES	52.7417	48.2083	47.275	50.05	58.3	52.9083	54.5
13	- VEGETABLE OILSEEDS AND OILS	61.0583	66.5667	59.1583	62.0583	65.1833	73.6833	68.8
14	AGRICULTURAL RAW MATERIALS	52.9667	46.775	44.25	43.325	43.1667	44.275	43.6
15	MINERALS, ORES AND METALS	34.3667	33.15	32.825	33.0333	41.9167	49.5583	53.9
16	Crude petroleum, average of UK Brent (light), ...	7.08333	7.08333	7.5	7.5	7.5	7.5	
17	Crude petroleum, UK Brent, light blend API 38°...	..	..	..	..	..	..	
18	Crude petroleum, Dubai, medium, Fateh API 32°,...	..	..	..	..	..	..	

	Free market commodity price indices, annual, 1960 - 2016 (Discontinued)	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnam
19	MEMO ITEM	_	_	_	_	_	_	
20	Unit value index of manufactured goods exports...	24.3	24.9	25.1	25.1	25.5	26.1	2

21 rows × 58 columns

In [626]:
```
# Commodity Prices were then imported and cleaned.

CMPrices=pd.read_excel (r"C:\New folder\us_annualcommoditypriceindicesaverages
_72693818609729.xls")

CMPrices

CMPrices=CMPrices.loc[[6]]

CMPrices
# unnamed are used as references in order to get the right years from 1985-201
6.
CMPrices=CMPrices[['Unnamed: 35', 'Unnamed: 36', 'Unnamed: 37', 'Unnamed: 38',
       'Unnamed: 39', 'Unnamed: 40', 'Unnamed: 41', 'Unnamed: 42',
       'Unnamed: 43', 'Unnamed: 44', 'Unnamed: 45', 'Unnamed: 46',
       'Unnamed: 47', 'Unnamed: 48', 'Unnamed: 49', 'Unnamed: 50',
       'Unnamed: 51', 'Unnamed: 52', 'Unnamed: 53', 'Unnamed: 54',
       'Unnamed: 55', 'Unnamed: 56', 'Unnamed: 57']]

CMPrices




#Data was then merged to create the following dataframe.
data7={'Year':['1985','1986','1987','1988','1989','1990','1991','1992','1993'
], 'Price':[96.2167,96.675,102.208,131.117,132.325,124.025, 117.042,113.858,10
8.258
]}

CMdata1=pd.DataFrame(data7)
CMdata1



data8={'Year':['1994','1995','1996','1997','1998','1999','2000','2001','2002',
'2003'], 'Price':[126.508,137.575,134.35,131.667,114.4,98.4833,99.9917,96.375,
97.3167,104.858]}


CMdata2=pd.DataFrame(data8)
CMdata2

data9={'Year':['2004','2005','2006','2007','2008','2009','2010','2011','2012',
'2013'],'Price':[125.783,140.392,182.825,206.525,256.033,212.742,256.042,302,2
76.783,258.183]}

CMdata3=pd.DataFrame(data9)
CMdata3

data10={'Year':['2014','2015','2016'],'Price':[242.508,201.575,200.083] }
CMdata4=pd.DataFrame(data10)
CMdata4

CMPriceData=pd.concat([CMdata1,CMdata2,CMdata3,CMdata4])
CMPriceData
```

Out[626]:

	Year	Price
0	1985	96.2167
1	1986	96.6750
2	1987	102.2080
3	1988	131.1170
4	1989	132.3250
5	1990	124.0250
6	1991	117.0420
7	1992	113.8580
8	1993	108.2580
0	1994	126.5080
1	1995	137.5750
2	1996	134.3500
3	1997	131.6670
4	1998	114.4000
5	1999	98.4833
6	2000	99.9917
7	2001	96.3750
8	2002	97.3167
9	2003	104.8580
0	2004	125.7830
1	2005	140.3920
2	2006	182.8250
3	2007	206.5250
4	2008	256.0330
5	2009	212.7420
6	2010	256.0420
7	2011	302.0000
8	2012	276.7830
9	2013	258.1830
0	2014	242.5080
1	2015	201.5750
2	2016	200.0830

In [627]:
```
#Next involves importing and cleaning the various risk indicators into new dat
aframes later to be merged. All countries have the same index number in each r
isk dataset.
#Data Cleaning for Political Risk dataset.
PR=pd.read_excel (r"C:\New folder\CountryData (2).xlsx")
PR=PR.loc[PR['Country'].isin(EMCountryList)]
PR
```

Out[627]:

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/1984
3	Argentina	Political Risk Rating	50.0	52.0	51.0	52.0	56.0	52.0	54.0	55.0
15	Brazil	Political Risk Rating	55.0	55.0	55.0	57.0	56.0	57.0	58.0	61.0
21	Chile	Political Risk Rating	49.0	50.0	48.0	47.0	49.0	46.0	45.0	45.0
22	China	Political Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
23	Colombia	Political Risk Rating	62.0	62.0	61.0	61.0	63.0	62.0	68.0	68.0
36	Egypt	Political Risk Rating	50.0	50.0	50.0	48.0	51.0	52.0	51.0	52.0
56	Hungary	Political Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	77.0
58	India	Political Risk Rating	60.0	60.0	58.0	58.0	56.0	49.0	46.0	47.0
59	Indonesia	Political Risk Rating	46.0	46.0	46.0	46.0	47.0	47.0	49.0	49.0
81	Malaysia	Political Risk Rating	77.0	76.0	76.0	76.0	75.0	74.0	73.0	73.0
84	Mexico	Political Risk Rating	66.0	69.0	70.0	70.0	70.0	69.0	68.0	68.0
99	Pakistan	Political Risk Rating	42.0	42.0	41.0	41.0	41.0	39.0	38.0	36.0
103	Peru	Political Risk Rating	45.0	45.0	45.0	41.0	45.0	45.0	44.0	44.0
104	Philippines	Political Risk Rating	38.0	38.0	38.0	38.0	38.0	39.0	40.0	39.0
105	Poland	Political Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
107	Qatar	Political Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	51.0
109	Russia	Political Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/1984
110	Saudi Arabia	Political Risk Rating	52.0	52.0	52.0	54.0	55.0	54.0	54.0	54.0
119	South Africa	Political Risk Rating	62.0	62.0	63.0	65.0	66.0	66.0	67.0	67.0
129	Thailand	Political Risk Rating	60.0	60.0	60.0	60.0	60.0	60.0	61.0	61.0

20 rows × 461 columns

In [628]:
```
FR=pd.read_excel (r"C:\New folder\CountryData.xlsx")
FR
```

Out[628]:

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/198
0	Albania	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
1	Algeria	Financial Risk Rating	27.5	27.5	28.0	28.0	27.0	25.0	25.0	25.
2	Angola	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
3	Argentina	Financial Risk Rating	15.0	14.5	14.0	14.0	13.0	9.0	9.0	9.
4	Armenia	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
5	Australia	Financial Risk Rating	41.5	41.5	41.0	41.0	41.0	42.0	42.0	42.
6	Austria	Financial Risk Rating	45.0	45.0	44.0	44.0	44.5	44.5	44.0	44.
7	Azerbaijan	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
8	Bahamas	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
9	Bahrain	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
10	Bangladesh	Financial Risk Rating	20.0	20.0	20.0	20.0	19.5	20.0	20.0	20.
11	Belarus	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
12	Belgium	Financial Risk Rating	42.5	42.5	41.0	41.0	41.5	41.5	41.0	41.
13	Bolivia	Financial Risk Rating	14.5	14.5	14.0	14.0	13.5	7.0	6.0	6.
14	Botswana	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
15	Brazil	Financial Risk Rating	20.0	20.0	20.0	20.0	20.5	21.5	22.0	23.
16	Brunei	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/198
17	Bulgaria	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
18	Burkina Faso	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
19	Cameroon	Financial Risk Rating	28.0	28.0	28.0	27.5	28.0	28.5	28.5	28.
20	Canada	Financial Risk Rating	44.0	44.0	43.0	43.0	43.0	43.0	42.5	43.
21	Chile	Financial Risk Rating	23.5	22.5	24.0	23.0	24.0	23.0	23.0	24.
22	China	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
23	Colombia	Financial Risk Rating	30.0	30.0	30.0	30.0	30.5	29.0	29.0	29.
24	Congo	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
25	Congo, DR	Financial Risk Rating	16.0	16.0	15.0	15.0	15.0	15.5	15.5	15.
26	Costa Rica	Financial Risk Rating	20.0	20.0	20.0	20.0	21.0	21.5	21.0	22.
27	Côte d’Ivoire	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
28	Croatia	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
29	Cuba	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
...	...	...	...	...	...	...	...	...	...	.
118	Somalia	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
119	South Africa	Financial Risk Rating	34.5	34.5	35.5	36.0	35.0	35.5	35.5	35.
120	Spain	Financial Risk Rating	37.0	37.0	37.0	36.5	36.5	37.5	37.5	36.
121	Sri Lanka	Financial Risk Rating	27.5	27.5	26.5	26.0	24.5	24.0	24.0	24.

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/198
122	Sudan	Financial Risk Rating	18.0	18.0	18.5	18.5	18.0	18.0	18.0	18.
123	Suriname	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
124	Sweden	Financial Risk Rating	44.0	44.0	43.0	43.0	43.0	42.0	42.0	42.
125	Switzerland	Financial Risk Rating	47.5	47.5	47.5	47.5	47.5	48.0	50.0	50.
126	Syria	Financial Risk Rating	16.5	16.5	15.5	14.5	15.0	14.5	14.5	16.
127	Taiwan	Financial Risk Rating	39.0	39.0	38.0	38.0	38.5	39.0	39.0	39.
128	Tanzania	Financial Risk Rating	22.0	22.0	23.0	23.0	23.0	23.0	23.0	23.
129	Thailand	Financial Risk Rating	31.0	31.0	31.0	31.0	31.0	30.5	30.5	30.
130	Togo	Financial Risk Rating	24.5	24.5	23.5	23.5	23.5	25.5	25.5	25.
131	Trinidad & Tobago	Financial Risk Rating	27.5	27.5	28.5	28.5	29.0	29.0	29.0	29.
132	Tunisia	Financial Risk Rating	22.0	22.0	22.0	22.0	21.5	23.5	23.5	23.
133	Turkey	Financial Risk Rating	25.5	25.5	24.5	25.0	24.5	24.5	24.5	23.
134	UAE	Financial Risk Rating	31.5	31.5	33.0	33.0	31.5	31.0	31.0	30.
135	Uganda	Financial Risk Rating	21.5	21.5	20.0	20.0	20.0	17.0	17.0	17.
136	Ukraine	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
137	United Kingdom	Financial Risk Rating	46.0	46.0	46.0	46.0	46.0	45.5	45.0	45.
138	United States	Financial Risk Rating	47.5	47.5	47.5	47.0	47.0	47.5	47.5	48.

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/198
139	Uruguay	Financial Risk Rating	26.5	26.5	26.5	26.5	27.5	28.0	28.0	27.
140	USSR	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
141	Venezuela	Financial Risk Rating	23.0	22.5	22.5	21.5	23.0	22.0	23.0	24.
142	Vietnam	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
143	Yemen	Financial Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
144	Zambia	Financial Risk Rating	21.0	21.0	19.5	19.5	19.5	19.5	19.5	19.
145	Zimbabwe	Financial Risk Rating	23.5	23.5	23.5	23.5	23.5	25.0	25.0	25.
146	The following requested columns had no data to...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
147	Copyright The PRS Group, Inc., 1979-2019, East...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

148 rows × 431 columns

In [629]:

```
#Data importing and cleaning for Sovereign Risk data.  ICRG calls sovereign ri
sk Financial Risk. This project refers to FR as Sovereign Risk.
FR=pd.read_excel (r"C:\New folder\CountryData.xlsx")

FR=FR.loc[FR['Country'].isin(EMCountryList)]
FR

FR=FR[[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1
999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014
,2015,2016]]
FR
```

Out[629]:

	1985	1986	1987	1988	1989	1990	1991	1992	
3	9.000000	10.750000	14.750000	17.833333	17.750000	21.000000	28.166667	36.000000	37
15	23.666667	23.750000	24.416667	29.000000	28.333333	29.333333	33.416667	37.750000	35
21	24.083333	25.166667	29.583333	36.666667	37.166667	38.666667	41.666667	42.000000	42
22	35.250000	32.916667	30.083333	29.916667	27.000000	24.583333	24.833333	32.166667	40
23	29.000000	29.833333	30.250000	30.416667	28.000000	31.166667	40.000000	41.583333	41
36	20.333333	18.000000	19.916667	20.833333	20.250000	21.666667	28.333333	38.916667	39
56	33.333333	31.916667	29.083333	27.583333	30.000000	30.416667	32.000000	35.916667	39
58	28.000000	29.000000	28.666667	26.833333	26.500000	25.250000	24.666667	32.500000	35
59	25.416667	22.583333	21.000000	19.833333	25.083333	36.916667	44.000000	43.666667	41
81	33.500000	28.666667	26.083333	27.833333	35.500000	42.250000	45.000000	45.000000	45
84	24.500000	20.333333	25.250000	27.583333	29.166667	33.416667	39.166667	42.333333	41
99	22.166667	22.500000	23.000000	22.166667	21.500000	21.083333	20.916667	28.000000	30
103	13.916667	12.500000	11.916667	16.250000	18.000000	20.833333	27.916667	27.333333	29
104	20.833333	20.833333	20.000000	20.666667	25.750000	23.583333	21.916667	29.500000	34
105	21.416667	18.000000	18.000000	21.333333	22.500000	24.750000	28.416667	33.083333	35
107	26.000000	26.000000	26.000000	26.000000	26.916667	30.333333	32.500000	40.166667	42
109	NaN	NaN	NaN	NaN	NaN	NaN	NaN	28.888889	27
110	27.000000	24.333333	24.416667	26.083333	28.833333	28.833333	30.666667	41.916667	43
119	27.750000	23.083333	27.083333	30.000000	30.166667	30.166667	30.000000	32.166667	36
129	29.000000	29.000000	29.000000	34.250000	40.500000	43.083333	42.000000	42.916667	42

20 rows × 32 columns

In [630]:
```
datafr={'Year':['1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992'
, '1993', '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '200
2', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2
012', '2013', '2014', '2015', '2016'],
       'Sovereign Risk': [FR.loc[:,1985].mean(),FR.loc[:,1986].mean(),FR.loc
[:,1987].mean(),FR.loc[:,1988].mean(),FR.loc[:,1989].mean(),FR.loc[:,1990].mea
n(),FR.loc[:,1991].mean(),FR.loc[:,1992].mean(),FR.loc[:,1993].mean(),FR.loc
[:,1994].mean(),FR.loc[:,1995].mean(),FR.loc[:,1996].mean(),FR.loc[:,1997].mea
n(),FR.loc[:,1998].mean(),FR.loc[:,1999].mean(),FR.loc[:,2000].mean(),FR.loc
[:,2001].mean(),FR.loc[:,2002].mean(),FR.loc[:,2003].mean(),FR.loc[:,2004].mea
n(),FR.loc[:,2005].mean(),FR.loc[:,2006].mean(),FR.loc[:,2007].mean(),FR.loc
[:,2008].mean(),FR.loc[:,2009].mean(),FR.loc[:,2010].mean(),FR.loc[:,2011].mea
n(),FR.loc[:,2012].mean(),FR.loc[:,2013].mean(),FR.loc[:,2014].mean(),FR.loc
[:,2015].mean(),FR.loc[:,2016].mean()]}
SovereignRisk=pd.DataFrame(datafr)
SovereignRisk
```

Out[630]:

	Year	Sovereign Risk
0	1985	24.956140
1	1986	23.640351
2	1987	24.131579
3	1988	25.846491
4	1989	27.311404
5	1990	29.333333
6	1991	32.399123
7	1992	36.590278
8	1993	37.912500
9	1994	38.233333
10	1995	38.066667
11	1996	38.729167
12	1997	37.633333
13	1998	34.797917
14	1999	35.470833
15	2000	36.064583
16	2001	36.518750
17	2002	36.704167
18	2003	37.404167
19	2004	38.854167
20	2005	39.608333
21	2006	40.583333
22	2007	40.716667
23	2008	39.827083
24	2009	39.002083
25	2010	41.262500
26	2011	41.337500
27	2012	41.083333
28	2013	41.072917
29	2014	40.608333
30	2015	39.339583
31	2016	39.550000

In [631]:
```
ER=pd.read_excel (r"C:\New folder\CountryData (1).xlsx")
ER
```

Out[631]:

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/19
0	Albania	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	Algeria	Economic Risk Rating	27.0	26.5	27.5	27.0	27.0	27.0	26.0	26
2	Angola	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	Argentina	Economic Risk Rating	15.0	14.0	13.5	13.0	12.5	10.5	10.5	10
4	Armenia	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
5	Australia	Economic Risk Rating	37.5	37.0	37.5	38.0	38.0	38.0	38.0	38
6	Austria	Economic Risk Rating	41.5	41.5	41.5	41.5	41.5	41.5	41.5	41
7	Azerbaijan	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
8	Bahamas	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
9	Bahrain	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
10	Bangladesh	Economic Risk Rating	25.0	25.0	25.0	25.5	25.5	25.5	26.0	25
11	Belarus	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
12	Belgium	Economic Risk Rating	36.5	36.5	37.0	36.5	36.5	36.5	37.0	37
13	Bolivia	Economic Risk Rating	13.0	13.0	14.5	14.5	14.5	14.0	13.5	13
14	Botswana	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
15	Brazil	Economic Risk Rating	16.0	16.0	18.0	19.0	19.5	19.5	20.0	20
16	Brunei	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/19
17	Bulgaria	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
18	Burkina Faso	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
19	Cameroon	Economic Risk Rating	30.0	30.0	31.5	30.0	30.0	30.0	31.5	31
20	Canada	Economic Risk Rating	40.0	39.5	39.5	40.0	39.5	39.5	39.5	39
21	Chile	Economic Risk Rating	23.5	23.0	25.5	25.5	25.0	25.0	22.5	22
22	China	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
23	Colombia	Economic Risk Rating	21.5	20.5	23.0	23.5	23.0	23.5	25.5	25
24	Congo	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
25	Congo, DR	Economic Risk Rating	11.0	11.0	18.0	19.0	18.5	16.5	15.5	15
26	Costa Rica	Economic Risk Rating	20.0	21.0	26.0	26.0	26.0	26.0	25.5	25
27	Côte d'Ivoire	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
28	Croatia	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
29	Cuba	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	...	
118	Somalia	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
119	South Africa	Economic Risk Rating	35.5	35.0	34.5	34.0	34.0	34.0	33.5	34
120	Spain	Economic Risk Rating	33.5	33.0	33.0	33.0	33.5	33.5	34.0	34
121	Sri Lanka	Economic Risk Rating	25.5	25.5	26.5	26.5	25.5	25.5	25.5	25

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/19
122	Sudan	Economic Risk Rating	22.5	22.5	28.0	28.0	28.0	28.0	28.0	28
123	Suriname	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
124	Sweden	Economic Risk Rating	38.5	38.5	38.5	38.5	38.5	38.5	38.5	39
125	Switzerland	Economic Risk Rating	44.5	44.5	44.5	44.5	44.5	44.5	44.5	44
126	Syria	Economic Risk Rating	24.5	25.0	25.0	24.0	24.0	24.0	24.0	24
127	Taiwan	Economic Risk Rating	41.0	40.0	40.5	41.0	41.0	41.0	40.5	40
128	Tanzania	Economic Risk Rating	19.5	19.5	19.5	20.0	20.0	20.0	20.0	20
129	Thailand	Economic Risk Rating	35.5	34.5	35.0	35.0	35.5	35.5	35.0	35
130	Togo	Economic Risk Rating	20.0	20.0	20.0	20.0	27.0	28.5	29.0	29
131	Trinidad & Tobago	Economic Risk Rating	35.0	36.0	35.0	34.0	34.0	34.0	33.0	33
132	Tunisia	Economic Risk Rating	32.0	30.5	31.5	30.5	31.0	30.0	29.5	30
133	Turkey	Economic Risk Rating	28.5	29.5	29.5	29.5	29.5	29.5	29.5	29
134	UAE	Economic Risk Rating	39.5	39.0	39.0	39.0	39.0	40.0	40.0	40
135	Uganda	Economic Risk Rating	12.0	12.0	14.5	14.5	14.5	14.5	14.5	14
136	Ukraine	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
137	United Kingdom	Economic Risk Rating	35.0	35.5	35.5	36.0	35.5	35.5	35.5	35
138	United States	Economic Risk Rating	40.0	39.5	40.0	39.5	38.0	38.0	38.0	38

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/19
139	Uruguay	Economic Risk Rating	28.5	28.0	30.0	31.5	30.0	30.0	32.0	32
140	USSR	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
141	Venezuela	Economic Risk Rating	32.0	32.0	30.0	30.5	30.0	34.5	35.0	34
142	Vietnam	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
143	Yemen	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
144	Zambia	Economic Risk Rating	21.5	21.5	17.0	19.5	19.5	19.5	20.0	20
145	Zimbabwe	Economic Risk Rating	23.0	27.0	27.5	27.5	27.5	26.0	26.0	26
146	The following requested columns had no data to...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
147	Copyright The PRS Group, Inc., 1979-2019, East...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

148 rows × 431 columns

In [632]:
```
#Data importing and cleaning for economic risk.

ER=pd.read_excel (r"C:\New folder\CountryData (1).xlsx")
ER=ER.loc[ER['Country'].isin(EMCountryList)]
ER
```

Out[632]:

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/198
3	Argentina	Economic Risk Rating	15.0	14.0	13.5	13.0	12.5	10.5	10.5	10.
15	Brazil	Economic Risk Rating	16.0	16.0	18.0	19.0	19.5	19.5	20.0	20.
21	Chile	Economic Risk Rating	23.5	23.0	25.5	25.5	25.0	25.0	22.5	22.
22	China	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
23	Colombia	Economic Risk Rating	21.5	20.5	23.0	23.5	23.0	23.5	25.5	25.
36	Egypt	Economic Risk Rating	29.0	28.5	29.5	29.5	28.0	27.0	26.5	26.
56	Hungary	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	32.
58	India	Economic Risk Rating	26.5	26.5	27.5	28.5	28.0	27.0	26.0	26.
59	Indonesia	Economic Risk Rating	28.5	28.0	28.5	29.0	30.0	30.0	30.0	30.
81	Malaysia	Economic Risk Rating	38.0	38.5	40.5	38.0	38.0	37.5	37.5	36.
84	Mexico	Economic Risk Rating	24.0	24.5	27.0	29.5	28.0	26.5	26.5	28.
99	Pakistan	Economic Risk Rating	32.5	31.0	31.5	32.5	33.0	31.5	32.5	33.
103	Peru	Economic Risk Rating	21.0	21.5	20.5	21.0	20.5	21.0	21.0	21.
104	Philippines	Economic Risk Rating	24.0	23.5	24.5	24.5	24.5	24.5	24.5	25.
105	Poland	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal
107	Qatar	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	36.
109	Russia	Economic Risk Rating	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Nal

	Country	Variable	01/1984	02/1984	03/1984	04/1984	05/1984	06/1984	07/1984	08/198
110	Saudi Arabia	Economic Risk Rating	37.0	37.5	38.0	38.0	37.5	37.5	37.5	37.
119	South Africa	Economic Risk Rating	35.5	35.0	34.5	34.0	34.0	34.0	33.5	34.
129	Thailand	Economic Risk Rating	35.5	34.5	35.0	35.0	35.5	35.5	35.0	35.

20 rows × 431 columns

In [633]:

```
ER=ER[[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1
999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014
,2015,2016]]
ER
```

Out[633]:

	1985	1986	1987	1988	1989	1990	1991	1992	
3	14.576389	16.000000	15.833333	17.791667	14.708333	22.166667	22.375000	26.416667	25
15	19.680556	21.708333	22.083333	20.291667	20.666667	20.875000	23.291667	25.958333	26
21	21.937500	22.208333	27.416667	30.166667	31.041667	29.833333	31.208333	36.125000	38
22	36.583333	34.125000	35.541667	36.708333	33.583333	34.625000	38.291667	41.083333	35
23	27.055556	29.875000	32.250000	31.875000	30.250000	30.458333	33.833333	35.958333	37
36	29.881944	25.208333	19.916667	24.458333	23.708333	24.375000	28.083333	29.791667	35
56	30.325000	29.833333	28.250000	27.000000	25.791667	25.083333	22.708333	30.083333	34
58	30.336806	33.250000	32.125000	29.625000	30.750000	29.833333	27.333333	27.125000	29
59	32.635417	33.416667	31.333333	33.500000	33.875000	35.333333	36.458333	35.083333	37
81	37.420139	37.875000	37.875000	40.333333	39.375000	40.166667	39.000000	38.208333	42
84	28.243056	24.500000	24.291667	25.833333	27.000000	28.583333	29.291667	30.375000	30
99	33.246528	33.041667	33.625000	32.000000	31.541667	31.375000	32.000000	31.875000	31
103	22.947917	25.583333	22.875000	19.500000	20.833333	20.291667	23.666667	25.125000	22
104	26.770833	26.083333	30.833333	30.875000	28.166667	28.708333	28.541667	31.125000	34
105	21.416667	19.875000	19.625000	18.375000	22.625000	26.125000	31.500000	34.375000	36
107	35.141667	32.333333	34.583333	37.333333	36.500000	35.916667	41.666667	41.125000	41
109	NaN	NaN	NaN	NaN	NaN	NaN	NaN	25.777778	32
110	34.750000	33.041667	37.875000	37.458333	38.916667	38.583333	37.958333	40.500000	39
119	31.826389	32.750000	31.833333	33.083333	33.083333	31.958333	32.750000	34.666667	37
129	34.392361	34.333333	36.416667	35.916667	35.791667	37.541667	36.750000	37.458333	39

20 rows × 32 columns

In [634]:
```
dataer={'Year':['1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992'
, '1993', '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '200
2', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2
012', '2013', '2014', '2015', '2016'],
       'Economic Risk': [ER.loc[:,1985].mean(),ER.loc[:,1986].mean(),ER.loc[:,
1987].mean(),ER.loc[:,1988].mean(),ER.loc[:,1989].mean(),ER.loc[:,1990].mean
(),ER.loc[:,1991].mean(),ER.loc[:,1992].mean(),ER.loc[:,1993].mean(),ER.loc[:,
1994].mean(),ER.loc[:,1995].mean(),ER.loc[:,1996].mean(),ER.loc[:,1997].mean
(),ER.loc[:,1998].mean(),ER.loc[:,1999].mean(),ER.loc[:,2000].mean(),ER.loc[:,
2001].mean(),ER.loc[:,2002].mean(),ER.loc[:,2003].mean(),ER.loc[:,2004].mean
(),ER.loc[:,2005].mean(),ER.loc[:,2006].mean(),ER.loc[:,2007].mean(),ER.loc[:,
2008].mean(),ER.loc[:,2009].mean(),ER.loc[:,2010].mean(),ER.loc[:,2011].mean
(),ER.loc[:,2012].mean(),ER.loc[:,2013].mean(),ER.loc[:,2014].mean(),ER.loc[:,
2015].mean(),ER.loc[:,2016].mean()]}
EconomicRisk=pd.DataFrame(dataer)
EconomicRisk
```

Out[634]:

	Year	Economic Risk
0	1985	28.903582
1	1986	28.686404
2	1987	29.188596
3	1988	29.585526
4	1989	29.379386
5	1990	30.096491
6	1991	31.405702
7	1992	32.911806
8	1993	34.397917
9	1994	35.670833
10	1995	35.670833
11	1996	36.485417
12	1997	36.202083
13	1998	33.412500
14	1999	32.143250
15	2000	36.362500
16	2001	36.343750
17	2002	35.825000
18	2003	36.920833
19	2004	38.441667
20	2005	38.593750
21	2006	39.152083
22	2007	39.710417
23	2008	38.762500
24	2009	34.235417
25	2010	36.756250
26	2011	37.720833
27	2012	37.506250
28	2013	37.533333
29	2014	37.070833
30	2015	36.420833
31	2016	35.627083

In [ ]:

```

```

In [635]:
```
PR=PR[[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1
999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014
,2015,2016]]
PR
```

Out[635]:

	1985	1986	1987	1988	1989	1990	1991	1992	
3	54.666667	56.166667	57.750000	56.416667	58.666667	61.166667	65.000000	67.666667	70
15	63.166667	65.583333	65.083333	67.000000	66.000000	67.333333	65.083333	66.583333	65
21	43.333333	44.083333	49.583333	56.083333	57.666667	64.333333	65.833333	66.583333	69
22	67.833333	64.583333	61.333333	63.416667	59.833333	56.916667	57.916667	67.750000	71
23	60.250000	58.750000	57.833333	50.916667	52.166667	55.750000	59.166667	60.083333	59
36	51.500000	42.000000	43.250000	42.833333	45.750000	49.583333	50.750000	58.083333	63
56	77.833333	74.916667	72.083333	71.250000	72.916667	71.000000	68.916667	71.166667	72
58	47.833333	51.583333	49.083333	46.166667	42.583333	39.083333	34.750000	47.750000	55
59	47.666667	43.000000	41.000000	39.833333	40.916667	48.333333	56.583333	56.916667	60
81	70.166667	65.000000	63.166667	59.166667	58.750000	65.833333	70.583333	68.833333	71
84	65.000000	61.500000	62.333333	66.083333	67.166667	69.750000	71.083333	70.333333	70
99	40.250000	41.833333	40.333333	37.500000	35.916667	30.750000	31.666667	37.333333	42
103	38.250000	38.833333	39.250000	40.166667	40.000000	42.416667	44.083333	39.916667	45
104	41.000000	43.583333	44.750000	41.166667	41.666667	35.916667	38.833333	49.166667	56
105	49.833333	50.833333	53.250000	55.333333	59.000000	62.250000	62.333333	64.750000	72
107	50.583333	50.250000	49.166667	47.583333	48.583333	53.083333	54.916667	63.000000	64
109	NaN	NaN	NaN	NaN	NaN	NaN	NaN	52.111111	51
110	49.500000	49.250000	50.833333	50.333333	55.416667	55.083333	57.750000	68.833333	70
119	55.583333	50.333333	52.416667	58.000000	59.333333	57.916667	56.833333	65.250000	65
129	57.666667	54.500000	55.000000	59.000000	58.750000	60.416667	56.250000	60.583333	65

20 rows × 32 columns

In [636]:
```
datapr={'Year':['1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992'
, '1993', '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '200
2', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2
012', '2013', '2014', '2015', '2016'],
       'Political Risk': [PR.loc[:,1985].mean(),PR.loc[:,1986].mean(),PR.loc
[:,1987].mean(),PR.loc[:,1988].mean(),PR.loc[:,1989].mean(),PR.loc[:,1990].mea
n(),PR.loc[:,1991].mean(),PR.loc[:,1992].mean(),PR.loc[:,1993].mean(),PR.loc
[:,1994].mean(),PR.loc[:,1995].mean(),PR.loc[:,1996].mean(),PR.loc[:,1997].mea
n(),PR.loc[:,1998].mean(),PR.loc[:,1999].mean(),PR.loc[:,2000].mean(),PR.loc
[:,2001].mean(),PR.loc[:,2002].mean(),PR.loc[:,2003].mean(),PR.loc[:,2004].mea
n(),PR.loc[:,2005].mean(),PR.loc[:,2006].mean(),PR.loc[:,2007].mean(),PR.loc
[:,2008].mean(),PR.loc[:,2009].mean(),PR.loc[:,2010].mean(),PR.loc[:,2011].mea
n(),PR.loc[:,2012].mean(),PR.loc[:,2013].mean(),PR.loc[:,2014].mean(),PR.loc
[:,2015].mean(),PR.loc[:,2016].mean()]}
PoliticalRisk=pd.DataFrame(datapr)
PoliticalRisk
```

Out[636]:

	Year	Political Risk
0	1985	54.311404
1	1986	52.978070
2	1987	53.026316
3	1988	53.065789
4	1989	53.741228
5	1990	55.100877
6	1991	56.228070
7	1992	60.134722
8	1993	63.200000
9	1994	66.479167
10	1995	66.683333
11	1996	67.495833
12	1997	69.587500
13	1998	67.825000
14	1999	64.558333
15	2000	63.683333
16	2001	66.133333
17	2002	65.300000
18	2003	66.183333
19	2004	67.306250
20	2005	67.189583
21	2006	67.225000
22	2007	66.777083
23	2008	66.199583
24	2009	66.125000
25	2010	65.666667
26	2011	64.593750
27	2012	63.718750
28	2013	63.427083
29	2014	63.027083
30	2015	62.668750
31	2016	63.327083

In [637]:
```
#Merging all risk dataframes into one.

dataRI={ 'Year':['1985', '1986', '1987', '1988', '1989', '1990', '1991', '199
2', '1993', '1994', '1995', '1996', '1997', '1998', '1999', '2000', '2001', '2
002', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011',
'2012', '2013', '2014', '2015', '2016'],
    'Political Risk':[PR.loc[:,1985].mean(),PR.loc[:,1986].mean(),PR.loc[:,198
7].mean(),PR.loc[:,1988].mean(),PR.loc[:,1989].mean(),PR.loc[:,1990].mean(),PR
.loc[:,1991].mean(),PR.loc[:,1992].mean(),PR.loc[:,1993].mean(),PR.loc[:,1994]
.mean(),PR.loc[:,1995].mean(),PR.loc[:,1996].mean(),PR.loc[:,1997].mean(),PR.l
oc[:,1998].mean(),PR.loc[:,1999].mean(),PR.loc[:,2000].mean(),PR.loc[:,2001].m
ean(),PR.loc[:,2002].mean(),PR.loc[:,2003].mean(),PR.loc[:,2004].mean(),PR.loc
[:,2005].mean(),PR.loc[:,2006].mean(),PR.loc[:,2007].mean(),PR.loc[:,2008].mea
n(),PR.loc[:,2009].mean(),PR.loc[:,2010].mean(),PR.loc[:,2011].mean(),PR.loc
[:,2012].mean(),PR.loc[:,2013].mean(),PR.loc[:,2014].mean(),PR.loc[:,2015].mea
n(),PR.loc[:,2016].mean()],
    'Economic Risk': [ER.loc[:,1985].mean(),ER.loc[:,1986].mean(),ER.loc[:,198
7].mean(),ER.loc[:,1988].mean(),ER.loc[:,1989].mean(),ER.loc[:,1990].mean(),ER
.loc[:,1991].mean(),ER.loc[:,1992].mean(),ER.loc[:,1993].mean(),ER.loc[:,1994]
.mean(),ER.loc[:,1995].mean(),ER.loc[:,1996].mean(),ER.loc[:,1997].mean(),ER.l
oc[:,1998].mean(),ER.loc[:,1999].mean(),ER.loc[:,2000].mean(),ER.loc[:,2001].m
ean(),ER.loc[:,2002].mean(),ER.loc[:,2003].mean(),ER.loc[:,2004].mean(),ER.loc
[:,2005].mean(),ER.loc[:,2006].mean(),ER.loc[:,2007].mean(),ER.loc[:,2008].mea
n(),ER.loc[:,2009].mean(),ER.loc[:,2010].mean(),ER.loc[:,2011].mean(),ER.loc
[:,2012].mean(),ER.loc[:,2013].mean(),ER.loc[:,2014].mean(),ER.loc[:,2015].mea
n(),ER.loc[:,2016].mean()],
    'Sovereign Risk':[FR.loc[:,1985].mean(),FR.loc[:,1986].mean(),FR.loc[:,198
7].mean(),FR.loc[:,1988].mean(),FR.loc[:,1989].mean(),FR.loc[:,1990].mean(),FR
.loc[:,1991].mean(),FR.loc[:,1992].mean(),FR.loc[:,1993].mean(),FR.loc[:,1994]
.mean(),FR.loc[:,1995].mean(),FR.loc[:,1996].mean(),FR.loc[:,1997].mean(),FR.l
oc[:,1998].mean(),FR.loc[:,1999].mean(),FR.loc[:,2000].mean(),FR.loc[:,2001].m
ean(),FR.loc[:,2002].mean(),FR.loc[:,2003].mean(),FR.loc[:,2004].mean(),FR.loc
[:,2005].mean(),FR.loc[:,2006].mean(),FR.loc[:,2007].mean(),FR.loc[:,2008].mea
n(),FR.loc[:,2009].mean(),FR.loc[:,2010].mean(),FR.loc[:,2011].mean(),FR.loc
[:,2012].mean(),FR.loc[:,2013].mean(),FR.loc[:,2014].mean(),FR.loc[:,2015].mea
n(),FR.loc[:,2016].mean()]}
RiskIndicators=pd.DataFrame(dataRI)
RiskIndicators
```

Out[637]:

	Year	Political Risk	Economic Risk	Sovereign Risk
0	1985	54.311404	28.903582	24.956140
1	1986	52.978070	28.686404	23.640351
2	1987	53.026316	29.188596	24.131579
3	1988	53.065789	29.585526	25.846491
4	1989	53.741228	29.379386	27.311404
5	1990	55.100877	30.096491	29.333333
6	1991	56.228070	31.405702	32.399123
7	1992	60.134722	32.911806	36.590278
8	1993	63.200000	34.397917	37.912500
9	1994	66.479167	35.670833	38.233333
10	1995	66.683333	35.670833	38.066667
11	1996	67.495833	36.485417	38.729167
12	1997	69.587500	36.202083	37.633333
13	1998	67.825000	33.412500	34.797917
14	1999	64.558333	32.143250	35.470833
15	2000	63.683333	36.362500	36.064583
16	2001	66.133333	36.343750	36.518750
17	2002	65.300000	35.825000	36.704167
18	2003	66.183333	36.920833	37.404167
19	2004	67.306250	38.441667	38.854167
20	2005	67.189583	38.593750	39.608333
21	2006	67.225000	39.152083	40.583333
22	2007	66.777083	39.710417	40.716667
23	2008	66.199583	38.762500	39.827083
24	2009	66.125000	34.235417	39.002083
25	2010	65.666667	36.756250	41.262500
26	2011	64.593750	37.720833	41.337500
27	2012	63.718750	37.506250	41.083333
28	2013	63.427083	37.533333	41.072917
29	2014	63.027083	37.070833	40.608333
30	2015	62.668750	36.420833	39.339583
31	2016	63.327083	35.627083	39.550000

In [638]:
```
#As mentioned before, Country Risk is calculated by the afformentioned methodology to calculate the ICRG composite rating.
RiskIndicators['Country Risk']=(RiskIndicators['Political Risk']+RiskIndicators['Economic Risk'])*0.5
RiskIndicators
```

Out[638]:

	Year	Political Risk	Economic Risk	Sovereign Risk	Country Risk
0	1985	54.311404	28.903582	24.956140	41.607493
1	1986	52.978070	28.686404	23.640351	40.832237
2	1987	53.026316	29.188596	24.131579	41.107456
3	1988	53.065789	29.585526	25.846491	41.325658
4	1989	53.741228	29.379386	27.311404	41.560307
5	1990	55.100877	30.096491	29.333333	42.598684
6	1991	56.228070	31.405702	32.399123	43.816886
7	1992	60.134722	32.911806	36.590278	46.523264
8	1993	63.200000	34.397917	37.912500	48.798958
9	1994	66.479167	35.670833	38.233333	51.075000
10	1995	66.683333	35.670833	38.066667	51.177083
11	1996	67.495833	36.485417	38.729167	51.990625
12	1997	69.587500	36.202083	37.633333	52.894792
13	1998	67.825000	33.412500	34.797917	50.618750
14	1999	64.558333	32.143250	35.470833	48.350792
15	2000	63.683333	36.362500	36.064583	50.022917
16	2001	66.133333	36.343750	36.518750	51.238542
17	2002	65.300000	35.825000	36.704167	50.562500
18	2003	66.183333	36.920833	37.404167	51.552083
19	2004	67.306250	38.441667	38.854167	52.873958
20	2005	67.189583	38.593750	39.608333	52.891667
21	2006	67.225000	39.152083	40.583333	53.188542
22	2007	66.777083	39.710417	40.716667	53.243750
23	2008	66.199583	38.762500	39.827083	52.481042
24	2009	66.125000	34.235417	39.002083	50.180208
25	2010	65.666667	36.756250	41.262500	51.211458
26	2011	64.593750	37.720833	41.337500	51.157292
27	2012	63.718750	37.506250	41.083333	50.612500
28	2013	63.427083	37.533333	41.072917	50.480208
29	2014	63.027083	37.070833	40.608333	50.048958
30	2015	62.668750	36.420833	39.339583	49.544792
31	2016	63.327083	35.627083	39.550000	49.477083

In [639]:
```
#List of Industrialized nations used in this model.
IndustrializedNations=['Australia', 'Austria', 'Belgium','Canada', 'Cyprus',
'Czech Republic','Denmark','Estonia','Finland','France','Germany','Greece','Ic
eland','Ireland','Israel','Italy','Japan','Luxembourg','Netherlands','New Zeal
and','Norway','Portugal','Slovenia','South Korea','Spain','Sweden','Singapore'
,'Switzerland','Turkey','United Kingdom','United States']
```

In [640]:
```
EMComExports=pd.read_csv(r"C:\New folder\PCTOT_06-10-2019 21-29-55-29_timeSeri
es.csv")
EMComExports
```

Out[640]:

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
0	Greece	174	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	32.768295	32.
1	Greece	174	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	176.011230	176.
2	Greece	174	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	97.688744	97.
3	Greece	174	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	93.098709	93.
4	Greece	174	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	104.930275	104.
5	Grenada	328	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	49.625645	52.
6	Grenada	328	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	183.068268	178.
7	Grenada	328	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	100.650032	100.
8	Grenada	328	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	94.332726	94.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
9	Grenada	328	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	106.696869	106.
10	Guatemala	258	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	96.737930	113.
11	Guatemala	258	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	26.792646	26.
12	Guatemala	258	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	182.292160	199.
13	Guatemala	258	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	99.996277	101.
14	Guatemala	258	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	92.180054	92.
15	Guatemala	258	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	108.479294	109.
16	Guyana	336	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	128.371109	185.
17	Guyana	336	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	212.733810	270.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
18	Guyana	336	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	106.982338	119.
19	Guyana	336	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	74.881348	74.
20	Guyana	336	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	142.869202	159.
21	China, P.R.: Hong Kong	532	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	105.606949	95.
22	China, P.R.: Hong Kong	532	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	58.939880	55.
23	China, P.R.: Hong Kong	532	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	150.703522	153.
24	China, P.R.: Hong Kong	532	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	100.112991	99.
25	China, P.R.: Hong Kong	532	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	94.433228	93.
26	China, P.R.: Hong Kong	532	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	106.014587	106.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
27	Honduras	268	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	114.992874	116.
28	Honduras	268	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	24.311348	24.
29	Honduras	268	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	192.343292	194.
...	...	...	...	...	...	...	...	...	
2154	Vanuatu	846	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	100.022644	104.
2155	Yemen, Republic of	474	Commodity Export Price Index, Individual Commo...	x	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2156	Yemen, Republic of	474	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2157	Yemen, Republic of	474	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2158	South Africa	199	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	65.903465	63.
2159	South Africa	199	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	96.383186	96.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
2160	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	31.486126	31.
2161	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2162	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2163	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2164	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2165	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	94.496376	94.
2166	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	47.010696	47.
2167	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	36.628090	36.
2168	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
2169	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	47.798470	48.
2170	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2171	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x	Historical, Rolling Weights, Index	H_RW_IX	Value	19.158808	20.
2172	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	99.478600	99.
2173	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	33.591061	33.
2174	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2175	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2176	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2177	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
2178	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	33.586281	33.
2179	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm	Historical, Rolling Weights, Index	H_RW_IX	Value	25.090509	24.
2180	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	47.428940	52.
2181	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2182	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2183	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	45.167797	45.

2184 rows × 65 columns

In [641]:
```
#Exports of Commodities data being imported and cleanted to create a new dataf
rame.
EMComExports=pd.read_csv(r"C:\New folder\PCTOT_06-10-2019 21-29-55-29_timeSeri
es.csv")

Exports=['Commodity Export Price Index, Individual Commodites Weighted by Rati
o of Exports to GDP']

# Because the IMF database goes by different names for a couple of EM in the o
riginal EM list,
#this list is created to reflects China's and Russia's name in this dataset.
IMFCountryList=["Argentina",'Brazil','Chile','China, P.R.: Mainland','Colombi
a','Egypt','Hungary','India','Indonesia','Malaysia','Mexico','Pakistan','Peru'
,'Philippines','Poland','Qatar','Russian Federation','Saudi Arabia','South Afr
ica','Thailand']

EMComExports

EMComExports=EMComExports.loc[EMComExports['Indicator Name'].isin(Exports)]

EMComExports

EMComExports.drop_duplicates(['Country Code'])

EMComExports=EMComExports.drop_duplicates(['Country Code'])

EMComExports=EMComExports.loc[EMComExports['Country Name'].isin(IMFCountryList
)]

EMComExports

data13={'Year':['1985','1986','1987','1988','1989','1990','1991','1992','1993'
,'1994','1995','1996','1997','1998','1999','2000','2001','2002','2003','2004',
'2005','2006','2007','2008','2009','2010','2011','2012','2013','2014','2015',
'2016'], 'Exports':[EMComExports.loc[:,'1985'].mean(),EMComExports.loc[:,'198
6'].mean(),EMComExports.loc[:,'1987'].mean(),EMComExports.loc[:,'1988'].mean
(),EMComExports.loc[:,'1989'].mean(),EMComExports.loc[:,'1990'].mean(),EMComEx
ports.loc[:,'1991'].mean(),EMComExports.loc[:,'1992'].mean(),EMComExports.loc
[:,'1993'].mean(),EMComExports.loc[:,'1994'].mean(),EMComExports.loc[:,'1995']
.mean(),EMComExports.loc[:,'1996'].mean(),EMComExports.loc[:,'1997'].mean(),EM
ComExports.loc[:,'1998'].mean(),EMComExports.loc[:,'1999'].mean(),EMComExports
.loc[:,'2000'].mean(),EMComExports.loc[:,'2001'].mean(),EMComExports.loc[:,'20
02'].mean(),EMComExports.loc[:,'2003'].mean(),EMComExports.loc[:,'2004'].mean
(),EMComExports.loc[:,'2005'].mean(),EMComExports.loc[:,'2006'].mean(),EMComEx
ports.loc[:,'2007'].mean(),EMComExports.loc[:,'2008'].mean(),EMComExports.loc
[:,'2009'].mean(),EMComExports.loc[:,'2010'].mean(),EMComExports.loc[:,'2011']
.mean(),EMComExports.loc[:,'2012'].mean(),EMComExports.loc[:,'2013'].mean(),EM
ComExports.loc[:,'2014'].mean(),EMComExports.loc[:,'2015'].mean(),EMComExports
.loc[:,'2016'].mean()]}

ExportsEM=pd.DataFrame(data13)

ExportsEM
```

Out[641]:

	Year	Exports
0	1985	94.052679
1	1986	90.762845
2	1987	91.118964
3	1988	90.369709
4	1989	90.787319
5	1990	91.382398
6	1991	90.197295
7	1992	89.548287
8	1993	89.021850
9	1994	89.203669
10	1995	89.646691
11	1996	90.015375
12	1997	89.981229
13	1998	88.157265
14	1999	88.973939
15	2000	91.409462
16	2001	90.751608
17	2002	90.909507
18	2003	92.017528
19	2004	93.628389
20	2005	95.733289
21	2006	97.471658
22	2007	98.226824
23	2008	100.323182
24	2009	96.560020
25	2010	98.762377
26	2011	100.776637
27	2012	100.000000
28	2013	100.132317
29	2014	99.542036
30	2015	95.251063
31	2016	94.616478

In [642]:
```
EMComImports=pd.read_csv(r"C:\New folder\PCTOT_06-10-2019 21-29-55-29_timeSeri
es.csv")
EMComImports
```

Out[642]:

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
0	Greece	174	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	32.768295	32.
1	Greece	174	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	176.011230	176.
2	Greece	174	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	97.688744	97.
3	Greece	174	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	93.098709	93.
4	Greece	174	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	104.930275	104.
5	Grenada	328	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	49.625645	52.
6	Grenada	328	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	183.068268	178.
7	Grenada	328	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	100.650032	100.
8	Grenada	328	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	94.332726	94.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
9	Grenada	328	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	106.696869	106.
10	Guatemala	258	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	96.737930	113.
11	Guatemala	258	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	26.792646	26.
12	Guatemala	258	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	182.292160	199.
13	Guatemala	258	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	99.996277	101.
14	Guatemala	258	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	92.180054	92.
15	Guatemala	258	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	108.479294	109.
16	Guyana	336	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	128.371109	185.
17	Guyana	336	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	212.733810	270.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
18	Guyana	336	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	106.982338	119.
19	Guyana	336	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	74.881348	74.
20	Guyana	336	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	142.869202	159.
21	China, P.R.: Hong Kong	532	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	105.606949	95.
22	China, P.R.: Hong Kong	532	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	58.939880	55.
23	China, P.R.: Hong Kong	532	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	150.703522	153.
24	China, P.R.: Hong Kong	532	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	100.112991	99.
25	China, P.R.: Hong Kong	532	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	94.433228	93.
26	China, P.R.: Hong Kong	532	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	106.014587	106.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
27	Honduras	268	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	114.992874	116.
28	Honduras	268	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	24.311348	24.
29	Honduras	268	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	192.343292	194.
...	...	...	...	...	...	...	...	...	
2154	Vanuatu	846	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	100.022644	104.
2155	Yemen, Republic of	474	Commodity Export Price Index, Individual Commo...	x	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2156	Yemen, Republic of	474	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2157	Yemen, Republic of	474	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2158	South Africa	199	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	65.903465	63.
2159	South Africa	199	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	96.383186	96.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
2160	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	31.486126	31.
2161	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2162	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2163	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2164	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2165	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	94.496376	94.
2166	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	47.010696	47.
2167	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	36.628090	36.
2168	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
2169	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	47.798470	48.
2170	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2171	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x	Historical, Rolling Weights, Index	H_RW_IX	Value	19.158808	20.
2172	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	99.478600	99.
2173	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	33.591061	33.
2174	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2175	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2176	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2177	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
2178	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	33.586281	33.
2179	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm	Historical, Rolling Weights, Index	H_RW_IX	Value	25.090509	24.
2180	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	47.428940	52.
2181	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2182	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2183	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	45.167797	45.

2184 rows × 65 columns

In [643]:
```
#Data imported and cleaned for Imports of Commodities by Emerging Markets.
EMComImports=pd.read_csv(r"C:\New folder\PCTOT_06-10-2019 21-29-55-29_timeSeries.csv")
EMComImports=EMComImports.loc[EMComImports['Country Name'].isin(IMFCountryList)]
EMComImports=EMComImports.drop_duplicates(['Country Code'])
EMComImports
data15={"Year":['1985','1986','1987','1988','1989','1990','1991','1992','1993','1994','1995','1996','1997','1998','1999','2000','2001','2002','2003','2004','2005','2006','2007','2008','2009','2010','2011','2012','2013','2014','2015','2016'], 'Imports':[EMComImports.loc[:,'1985'].mean(),EMComImports.loc[:,'1986'].mean(),EMComImports.loc[:,'1987'].mean(),EMComImports.loc[:,'1988'].mean(),EMComImports.loc[:,'1989'].mean(),EMComImports.loc[:,'1990'].mean(),EMComImports.loc[:,'1991'].mean(),EMComImports.loc[:,'1992'].mean(),EMComImports.loc[:,'1993'].mean(),EMComImports.loc[:,'1994'].mean(),EMComImports.loc[:,'1995'].mean(),EMComImports.loc[:,'1996'].mean(),EMComImports.loc[:,'1997'].mean(),EMComImports.loc[:,'1998'].mean(),EMComImports.loc[:,'1999'].mean(),EMComImports.loc[:,'2000'].mean(),EMComImports.loc[:,'2001'].mean(),EMComImports.loc[:,'2002'].mean(),EMComImports.loc[:,'2003'].mean(),EMComImports.loc[:,'2004'].mean(),EMComImports.loc[:,'2005'].mean(),EMComImports.loc[:,'2006'].mean(),EMComImports.loc[:,'2007'].mean(),EMComImports.loc[:,'2008'].mean(),EMComImports.loc[:,'2009'].mean(),EMComImports.loc[:,'2010'].mean(),EMComImports.loc[:,'2011'].mean(),EMComImports.loc[:,'2012'].mean(),EMComImports.loc[:,'2013'].mean(),EMComImports.loc[:,'2014'].mean(),EMComImports.loc[:,'2015'].mean(),EMComImports.loc[:,'2016'].mean()]}

ImportsEM=pd.DataFrame(data15)
```

In [644]:
```
ImportsEM
```

Out[644]:

	Year	Imports
0	1985	63.212399
1	1986	50.174236
2	1987	51.169544
3	1988	50.211320
4	1989	51.398915
5	1990	52.149496
6	1991	46.496687
7	1992	45.035070
8	1993	42.923114
9	1994	45.238428
10	1995	47.524445
11	1996	47.573090
12	1997	47.774487
13	1998	41.052222
14	1999	41.741128
15	2000	49.110493
16	2001	46.652244
17	2002	47.373545
18	2003	51.508441
19	2004	59.232385
20	2005	68.468254
21	2006	80.358308
22	2007	85.772858
23	2008	100.881672
24	2009	77.563004
25	2010	94.301768
26	2011	109.652496
27	2012	100.000000
28	2013	101.387097
29	2014	96.839721
30	2015	71.624776
31	2016	70.378378

In [645]:
```
IEComImports=pd.read_csv(r"C:\New folder\PCTOT_06-10-2019 21-29-55-29_timeSeri
es.csv")
IEComImports
```

Out[645]:

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
0	Greece	174	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	32.768295	32.
1	Greece	174	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	176.011230	176.
2	Greece	174	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	97.688744	97.
3	Greece	174	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	93.098709	93.
4	Greece	174	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	104.930275	104.
5	Grenada	328	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	49.625645	52.
6	Grenada	328	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	183.068268	178.
7	Grenada	328	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	100.650032	100.
8	Grenada	328	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	94.332726	94.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
9	Grenada	328	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	106.696869	106.
10	Guatemala	258	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	96.737930	113.
11	Guatemala	258	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	26.792646	26.
12	Guatemala	258	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	182.292160	199.
13	Guatemala	258	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	99.996277	101.
14	Guatemala	258	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	92.180054	92.
15	Guatemala	258	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	108.479294	109.
16	Guyana	336	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	128.371109	185.
17	Guyana	336	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	212.733810	270.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
18	Guyana	336	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	106.982338	119.
19	Guyana	336	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	74.881348	74.
20	Guyana	336	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	142.869202	159.
21	China, P.R.: Hong Kong	532	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	105.606949	95.
22	China, P.R.: Hong Kong	532	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	58.939880	55.
23	China, P.R.: Hong Kong	532	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	150.703522	153.
24	China, P.R.: Hong Kong	532	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	100.112991	99.
25	China, P.R.: Hong Kong	532	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	94.433228	93.
26	China, P.R.: Hong Kong	532	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	106.014587	106.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
27	Honduras	268	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	114.992874	116.
28	Honduras	268	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	24.311348	24.
29	Honduras	268	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	192.343292	194.
...	...	...	...	...	...	...	...	...	
2154	Vanuatu	846	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	100.022644	104.
2155	Yemen, Republic of	474	Commodity Export Price Index, Individual Commo...	x	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2156	Yemen, Republic of	474	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2157	Yemen, Republic of	474	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2158	South Africa	199	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	65.903465	63.
2159	South Africa	199	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	96.383186	96.

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
2160	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	31.486126	31.
2161	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2162	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2163	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2164	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2165	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	94.496376	94.
2166	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m	Historical, Fixed Weights, Index	H_FW_IX	Value	47.010696	47.
2167	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm	Historical, Fixed Weights, Index	H_FW_IX	Value	36.628090	36.
2168	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
2169	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	47.798470	48.
2170	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x	Historical, Fixed Weights, Index	H_FW_IX	Value	NaN	
2171	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x	Historical, Rolling Weights, Index	H_RW_IX	Value	19.158808	20.
2172	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	99.478600	99.
2173	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	33.591061	33.
2174	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2175	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2176	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2177	Uzbekistan	927	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	

	Country Name	Country Code	Indicator Name	Indicator Code	Type Name	Type Code	Attribute	1962	
2178	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	33.586281	33.
2179	Equatorial Guinea	642	Commodity Net Export Price Index, Individual C...	xm	Historical, Rolling Weights, Index	H_RW_IX	Value	25.090509	24.
2180	Equatorial Guinea	642	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	47.428940	52.
2181	Uzbekistan	927	Commodity Net Export Price Index, Individual C...	xm_gdp	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2182	Uzbekistan	927	Commodity Import Price Index, Individual Commo...	m	Historical, Rolling Weights, Index	H_RW_IX	Value	NaN	
2183	Equatorial Guinea	642	Commodity Export Price Index, Individual Commo...	x_gdp	Historical, Fixed Weights, Index	H_FW_IX	Value	45.167797	45.

2184 rows × 65 columns

In [646]:
```
#Data Importing and cleaning for Imports of Commodities by Industrialized coun
tries.
IEComImports=pd.read_csv(r"C:\New folder\PCTOT_06-10-2019 21-29-55-29_timeSeri
es.csv")

IEComImports

IndustrializedNations=['Australia', 'Austria', 'Belgium','Canada', 'Cyprus',
'Czech Republic','Denmark','Estonia','Finland','France','Germany','Greece','Ic
eland','Ireland','Israel','Italy','Japan','Luxembourg','Netherlands','New Zeal
and','Norway','Portugal','Slovenia','South Korea','Spain','Sweden','Singapore'
,'Switzerland','Turkey','United Kingdom','United States']

Imports=['Commodity Import Price Index, Individual Commodites Weighted by Rati
o of Imports to GDP']

IEComImports=IEComImports.loc[IEComImports['Indicator Name'].isin(Imports)]

IEComImports=IEComImports.loc[IEComImports['Country Name'].isin(Industrialized
Nations)]

IEComImports=IEComImports.drop_duplicates(['Country Code'])


data14={"Year":['1985','1986','1987','1988','1989','1990','1991','1992','1993'
,'1994','1995','1996','1997','1998','1999','2000','2001','2002','2003','2004',
'2005','2006','2007','2008','2009','2010','2011','2012','2013','2014','2015',
'2016'], 'Imports':[IEComImports.loc[:,'1985'].mean(),IEComImports.loc[:,'198
6'].mean(),IEComImports.loc[:,'1987'].mean(),IEComImports.loc[:,'1988'].mean
(),IEComImports.loc[:,'1989'].mean(),IEComImports.loc[:,'1990'].mean(),IEComIm
ports.loc[:,'1991'].mean(),IEComImports.loc[:,'1992'].mean(),IEComImports.loc
[:,'1993'].mean(),IEComImports.loc[:,'1994'].mean(),IEComImports.loc[:,'1995']
.mean(),IEComImports.loc[:,'1996'].mean(),IEComImports.loc[:,'1997'].mean(),IE
ComImports.loc[:,'1998'].mean(),IEComImports.loc[:,'1999'].mean(),IEComImports
.loc[:,'2000'].mean(),IEComImports.loc[:,'2001'].mean(),IEComImports.loc[:,'20
02'].mean(),IEComImports.loc[:,'2003'].mean(),IEComImports.loc[:,'2004'].mean
(),IEComImports.loc[:,'2005'].mean(),IEComImports.loc[:,'2006'].mean(),IEComIm
ports.loc[:,'2007'].mean(),IEComImports.loc[:,'2008'].mean(),IEComImports.loc
[:,'2009'].mean(),IEComImports.loc[:,'2010'].mean(),IEComImports.loc[:,'2011']
.mean(),IEComImports.loc[:,'2012'].mean(),IEComImports.loc[:,'2013'].mean(),IE
ComImports.loc[:,'2014'].mean(),IEComImports.loc[:,'2015'].mean(),IEComImports
.loc[:,'2016'].mean()]}

ImportsIE=pd.DataFrame(data14)
```

In [647]:
```
ImportsIE
```

Out[647]:

	Year	Imports
0	1985	96.557585
1	1986	94.154582
2	1987	94.529721
3	1988	93.760895
4	1989	94.155338
5	1990	94.759359
6	1991	93.871376
7	1992	93.433895
8	1993	93.036918
9	1994	93.048641
10	1995	93.216708
11	1996	93.612508
12	1997	93.581168
13	1998	92.201875
14	1999	92.696455
15	2000	94.531046
16	2001	94.091867
17	2002	94.238607
18	2003	94.881072
19	2004	95.907690
20	2005	97.302934
21	2006	98.363765
22	2007	98.810095
23	2008	100.078963
24	2009	97.786445
25	2010	99.250814
26	2011	100.507312
27	2012	100.000000
28	2013	100.144213
29	2014	99.781789
30	2015	96.908815
31	2016	96.528836

Step by Step

The First step will be to plot simple graphs of the relationship between the various independent and dependent variables. The following step will involve a basic correlation and bi-linear regression dates between the variables involved in the data visualization to demonstrate how the strong the risk variables explored in this model are with regards to impacting their correspondent dependent variables. After basic exploration, the next step will be to conduct various multivariate regression analyses. Because China, who has the highest GDP among it's EM peers in this model, and it's other BRICS (Brazil, Russia, India, and South Africa) also pull such similar weight thus affecting the independent variables in question, four versions of the multivariate model is portrayed below: one with all the EMs listed in this model, one without China, one without BRICS, and one only including BRICS.

Data Visualization

In [648]:
```
import numpy as np
import matplotlib.pyplot as plt
from pylab import figure
%matplotlib inline
```

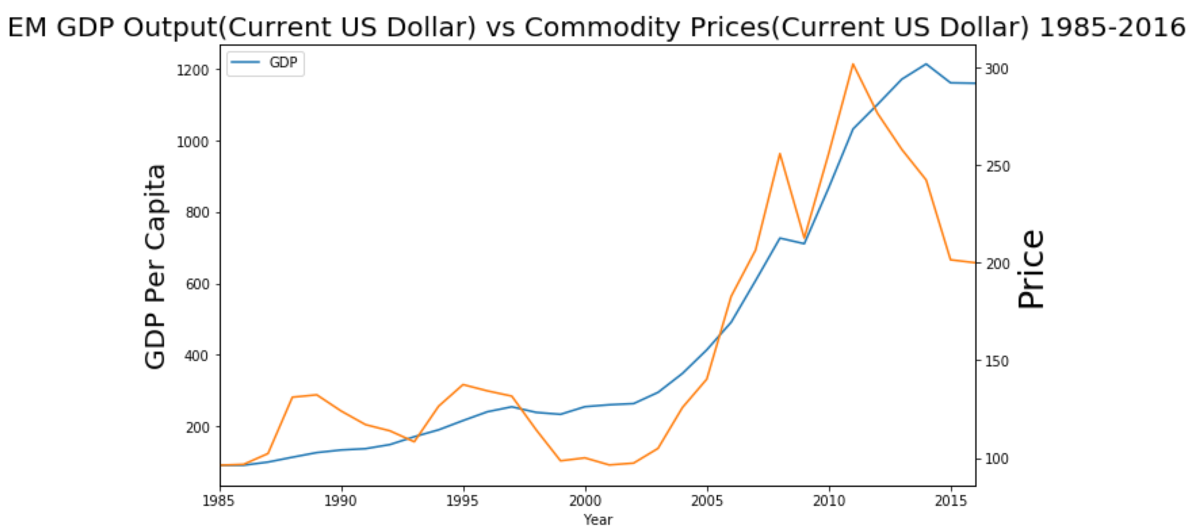
In [649]:
```
#to easily graph and conduct analysis, new dataframes have been created
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]

GDP=[91.49, 91.53, 100.36, 113.91, 126.86, 134.15, 137.78, 149.26, 171.26, 19
0.53, 216.39, 241.07, 254.93, 239.29, 233.89, 255.18, 260.65, 263.93, 295.3, 3
48.16, 414.25, 491.49, 607.78, 726.82, 711.14, 867.39, 1032.07, 1100.41, 1171.
2, 1213.92, 1161.19, 1160.0]

prices=[96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858,
108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375, 9
7.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.042
, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]

GDPvPrices=pd.DataFrame({'GDP':GDP, 'Commodity Prices':prices}, index=year)
```

In [650]:
```
GDPvPrices
```

Out[650]:

	GDP	Commodity Prices
1985	91.49	96.2167
1986	91.53	96.6750
1987	100.36	102.2080
1988	113.91	131.1170
1989	126.86	132.3250
1990	134.15	124.0250
1991	137.78	117.0420
1992	149.26	113.8580
1993	171.26	108.2580
1994	190.53	126.5080
1995	216.39	137.5750
1996	241.07	134.3500
1997	254.93	131.6670
1998	239.29	114.4000
1999	233.89	98.4833
2000	255.18	99.9917
2001	260.65	96.3750
2002	263.93	97.3167
2003	295.30	104.8580
2004	348.16	125.7830
2005	414.25	140.3920
2006	491.49	182.8250
2007	607.78	206.5250
2008	726.82	256.0330
2009	711.14	212.7420
2010	867.39	256.0420
2011	1032.07	302.0000
2012	1100.41	276.7830
2013	1171.20	258.1830
2014	1213.92	242.5080
2015	1161.19	201.5750
2016	1160.00	200.0830

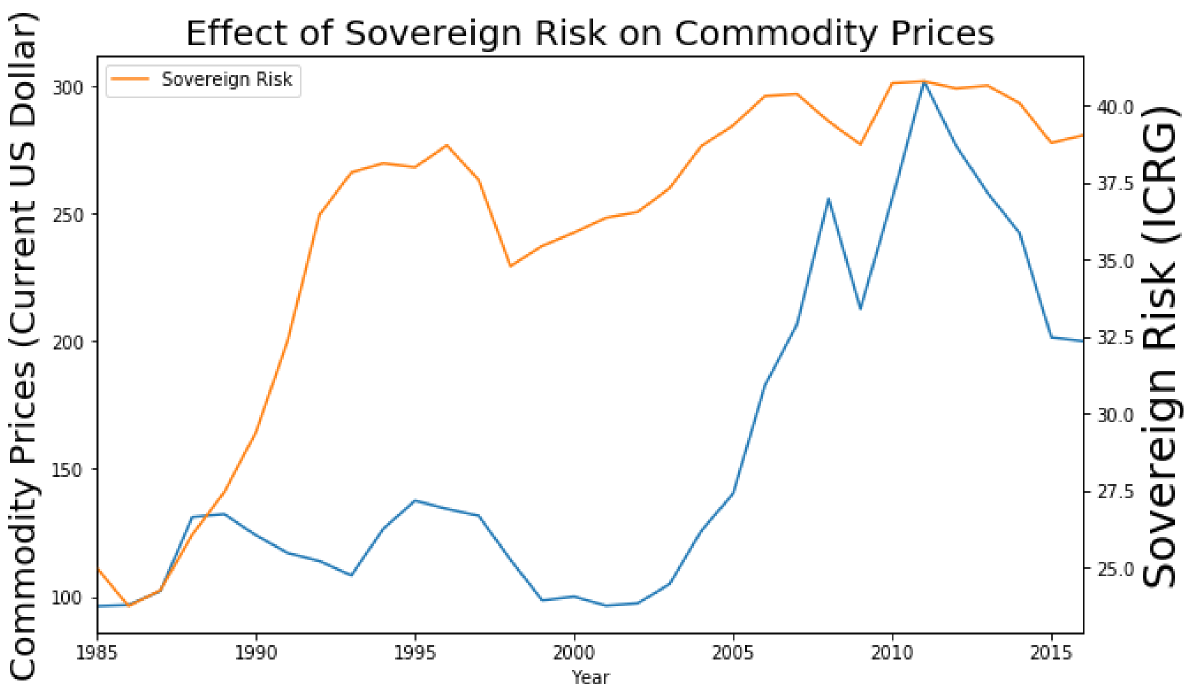
In [651]:
```
ax=GDPvPrices['GDP'].plot(label='GDP',figsize=(10,6))
ax.set_ylabel('GDP Per Capita', fontsize=20)
ax.set_xlabel('Year')
lns1=ax.plot(label='GDP')


ax.legend(['GDP'])

ax2=GDPvPrices['Commodity Prices'].plot(secondary_y=True,label='Commodity Prices (Current $US Dollars)', figsize=(10,6))
ax2.set_ylabel('Price',fontsize=25)
lns2=ax2.plot(label='Commodity Prices')




plt.title('EM GDP Output(Current US Dollar) vs Commodity Prices(Current US Dollar) 1985-2016',fontsize=20)

plt.figure(figsize=(10,10))

plt.show()
```




```
<Figure size 720x720 with 0 Axes>
```

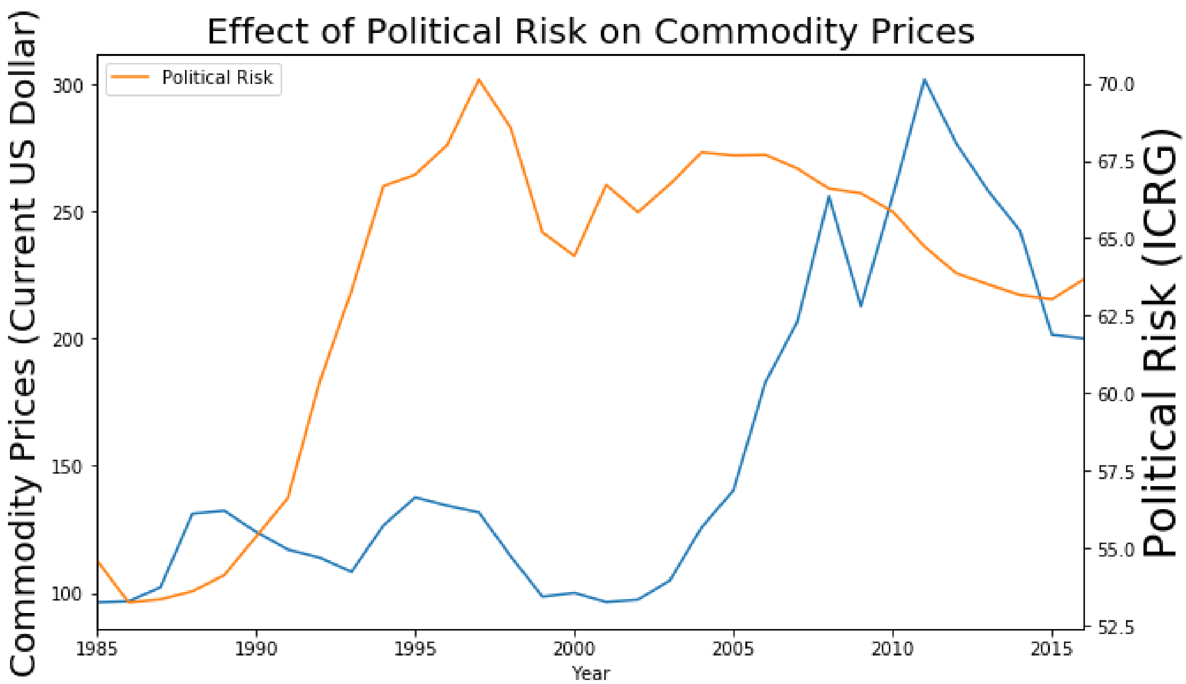
In [652]:
```
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]
prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858
, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375,
97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.04
2, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]
SovereignRisk=[ 25.0, 23.76, 24.28, 26.08, 27.45, 29.4, 32.38, 36.46, 37.83, 3
8.13, 38.0, 38.71, 37.59, 34.79, 35.45, 35.88, 36.36, 36.55, 37.32, 38.69, 39.
36, 40.31, 40.38, 39.49, 38.74, 40.73, 40.79, 40.56, 40.65, 40.08, 38.79, 39.0
4]
PoliticalRisk=[ 54.61, 53.25, 53.35, 53.61, 54.13, 55.37, 56.62, 60.37, 63.31,
66.68, 67.05, 68.0, 70.12, 68.56, 65.19, 64.43, 66.72, 65.84, 66.74, 67.77, 6
7.67, 67.69, 67.25, 66.6, 66.45, 65.86, 64.73, 63.87, 63.51, 63.17, 63.03, 63.
67]
CountryRisk=[ 41.81, 40.98, 41.29, 41.6, 41.82, 42.73, 43.96, 46.59, 48.84, 5
1.15, 51.37, 52.23, 53.15, 51.06, 48.8, 50.46, 51.62, 50.91, 51.85, 53.08, 53.
08, 53.36, 53.4, 52.61, 50.27, 51.08, 50.96, 50.43, 50.36, 50.04, 49.74, 49.62
]

PriceRisk=pd.DataFrame({'Commodity Prices':prices,'Sovereign Risk':SovereignRi
sk,'Political Risk':PoliticalRisk,'Country Risk':CountryRisk},index=year)
```

In [653]:
```
ax=PriceRisk['Commodity Prices'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('Commodity Prices (Current US Dollar)',fontsize=20)
ax.set_xlabel('Year')


ax2=PriceRisk['Sovereign Risk'].plot(secondary_y=True,label='Sovereign Risk')
ax2.set_ylabel('Sovereign Risk (ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of Sovereign Risk on Commodity Prices',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()

plt.savefig('figure3.png')
```



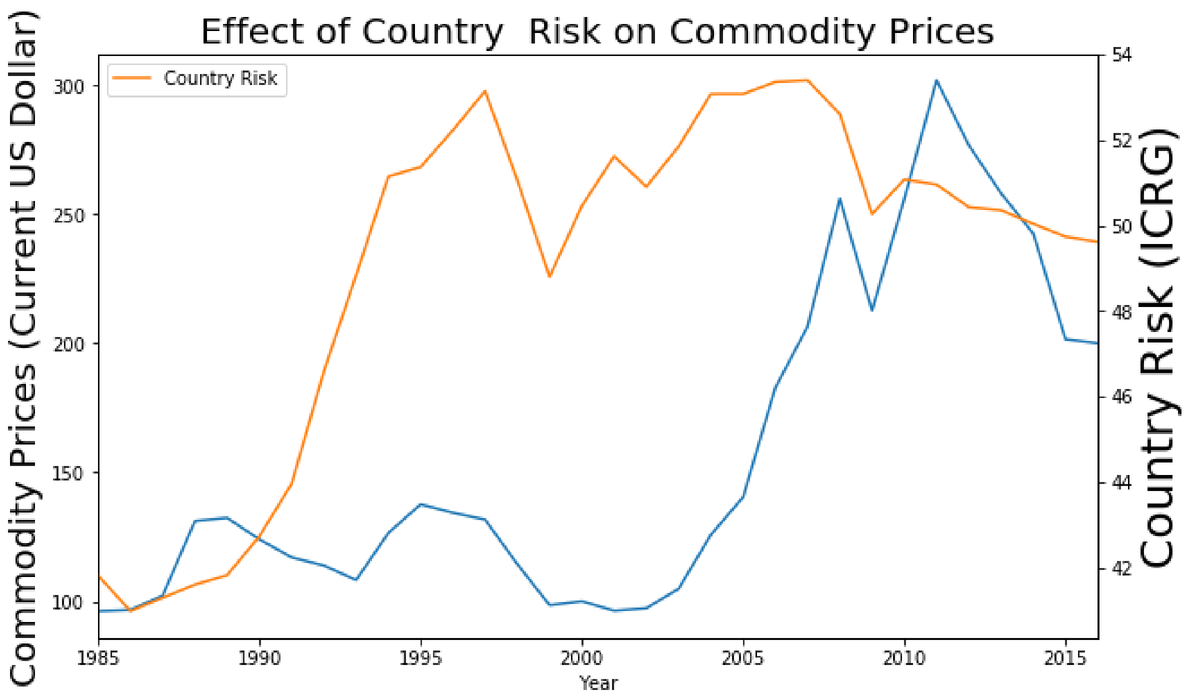

```
<Figure size 720x720 with 0 Axes>

<Figure size 432x288 with 0 Axes>
```

In [654]:
```
ax=PriceRisk['Commodity Prices'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('Commodity Prices (Current US Dollar)',fontsize=20)
ax.set_xlabel('Year')


ax2=PriceRisk['Political Risk'].plot(secondary_y=True,label='Political Risk')
ax2.set_ylabel('Political Risk (ICRG)',fontsize=25)

plt.legend(loc='upper left')


plt.title('Effect of Political Risk on Commodity Prices',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```



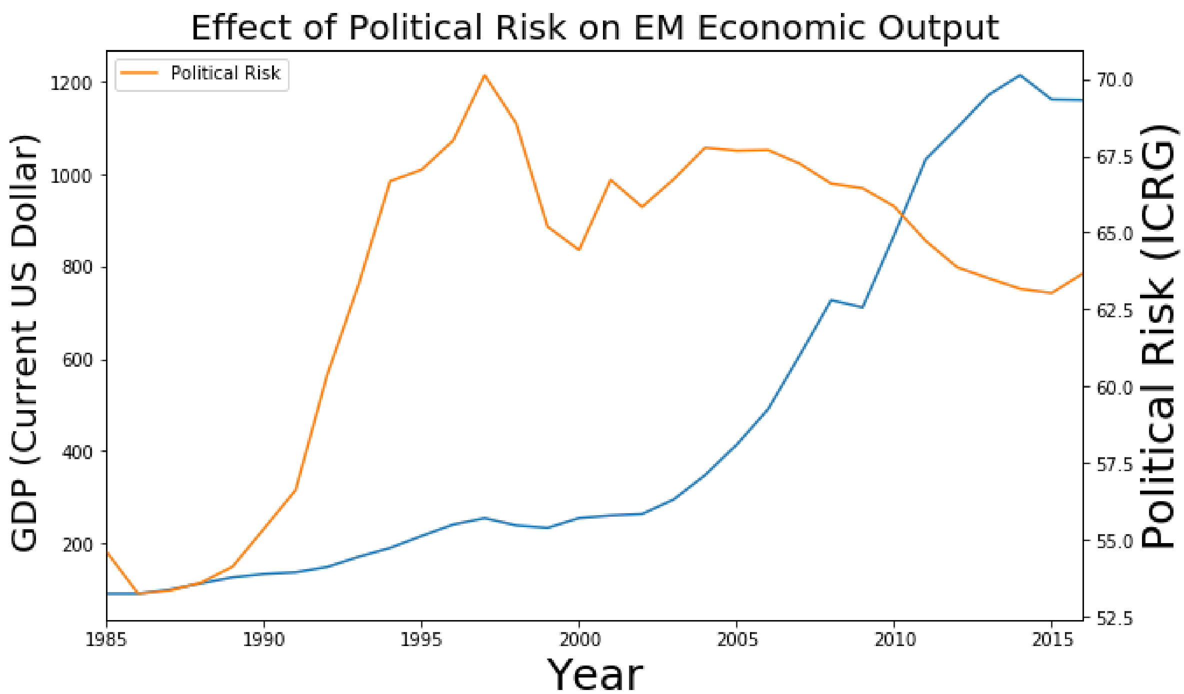

```
<Figure size 720x720 with 0 Axes>
```

In [655]:
```
ax=PriceRisk['Commodity Prices'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('Commodity Prices (Current US Dollar)',fontsize=20)
ax.set_xlabel('Year')


ax2=PriceRisk['Country Risk'].plot(secondary_y=True,label='Country Risk')
ax2.set_ylabel('Country Risk (ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of Country  Risk on Commodity Prices',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```



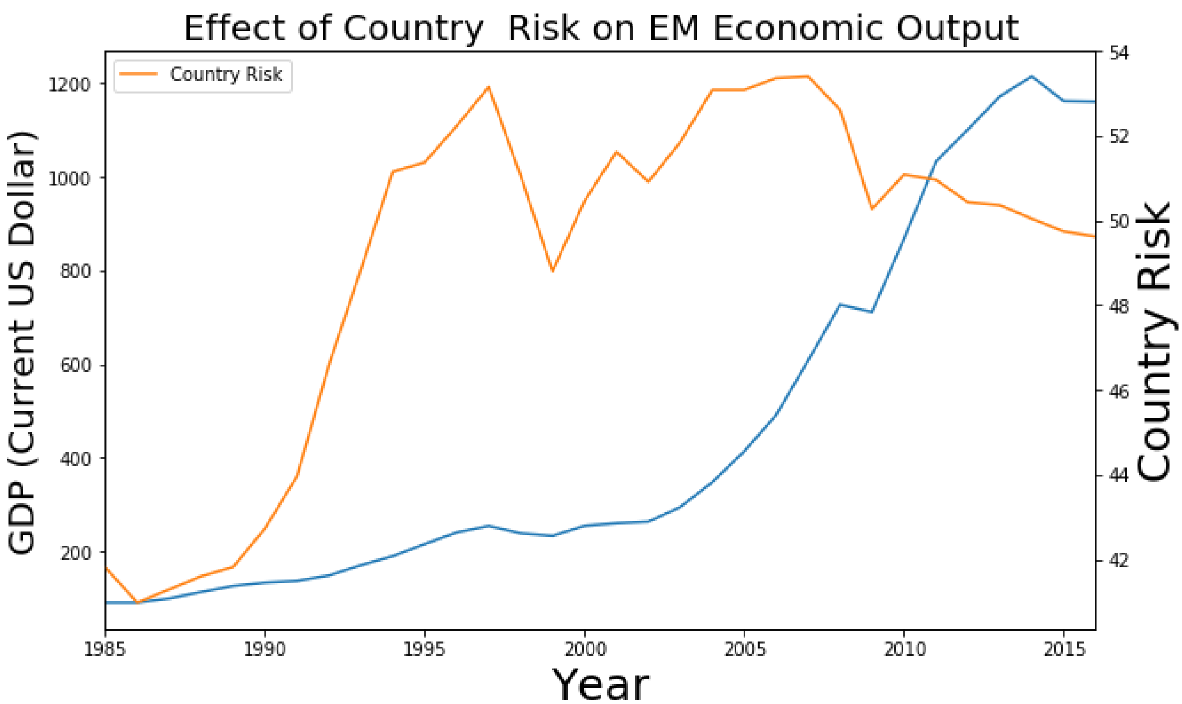

```
<Figure size 720x720 with 0 Axes>
```

In [656]:
```
ax=GDPvPrices['GDP'].plot(label='GDP',figsize=(10,6))
ax.set_ylabel('GDP (Current US Dollar) ',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=PriceRisk['Political Risk'].plot(secondary_y=True,label='Political Risk')
ax2.set_ylabel('Political Risk (ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of Political Risk on EM Economic Output',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```



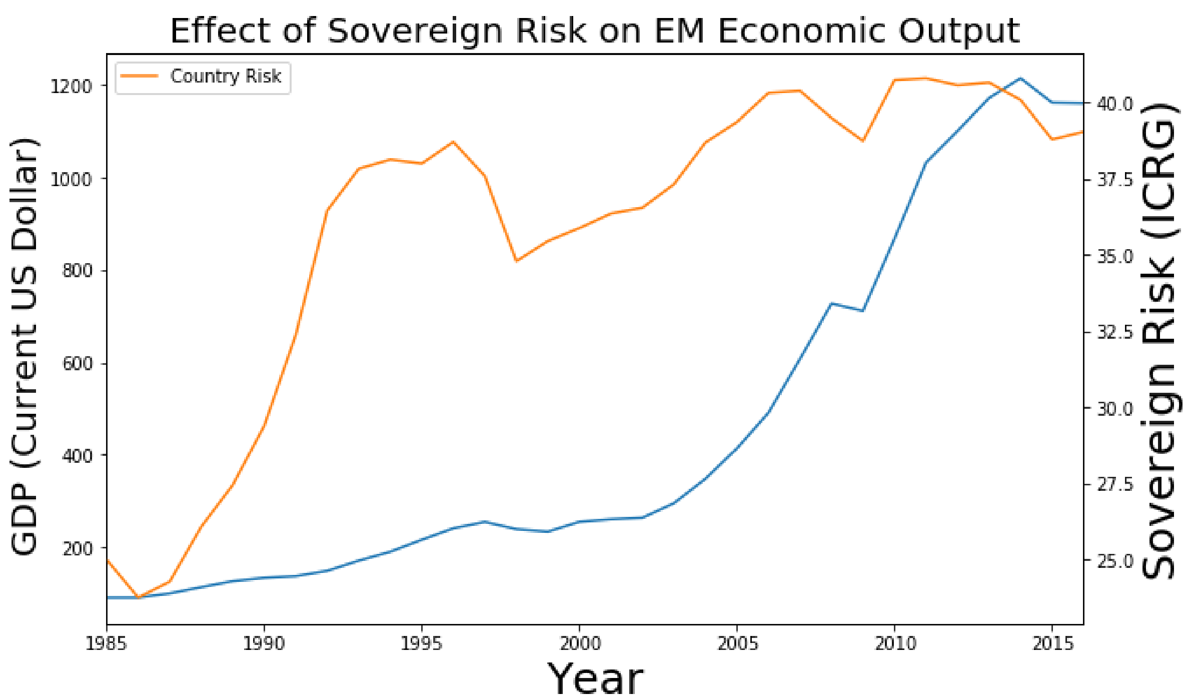

```
<Figure size 720x720 with 0 Axes>
```

In [657]:
```
ax=GDPvPrices['GDP'].plot(label='GDP',figsize=(10,6))
ax.set_ylabel('GDP (Current US Dollar)',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=PriceRisk['Country Risk'].plot(secondary_y=True,label='Country Risk')
ax2.set_ylabel('Country Risk',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of Country  Risk on EM Economic Output',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```




```
<Figure size 720x720 with 0 Axes>
```

In [658]:
```
ax=GDPvPrices['GDP'].plot(label='GDP',figsize=(10,6))
ax.set_ylabel('GDP (Current US Dollar) ',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=PriceRisk['Sovereign Risk'].plot(secondary_y=True,label='Country Risk')
ax2.set_ylabel('Sovereign Risk (ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of Sovereign Risk on EM Economic Output',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```




```
<Figure size 720x720 with 0 Axes>
```

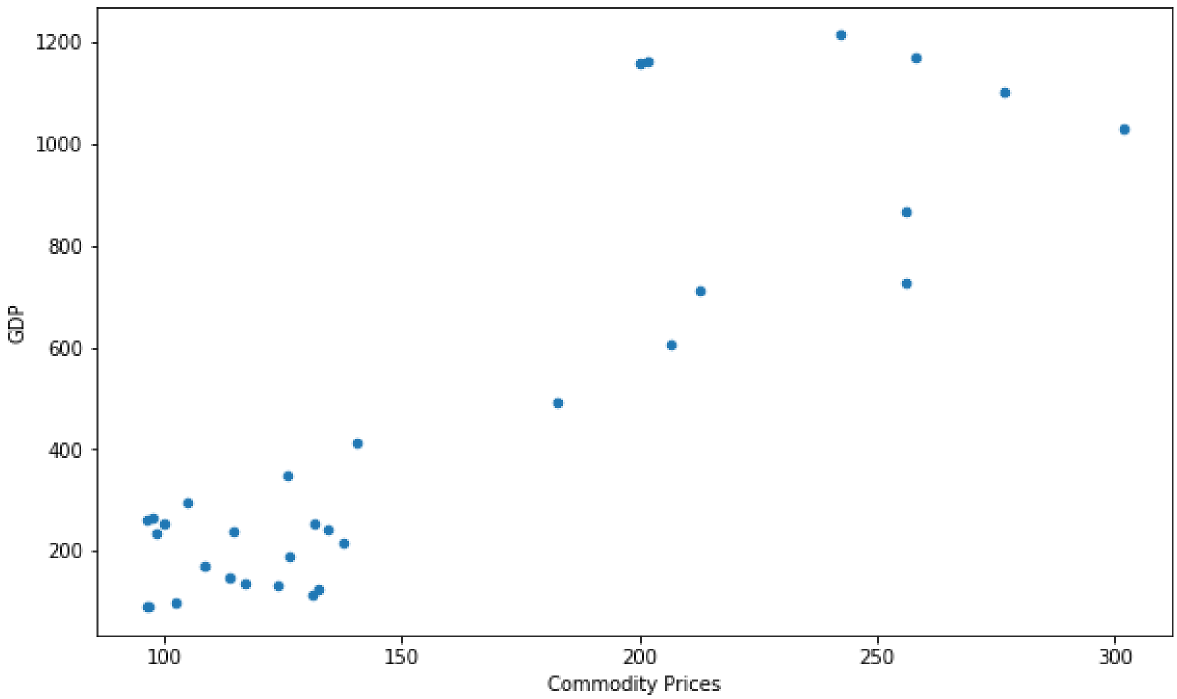
The following step will involve a basic correlation and bi-linear regression analysis between the variables involved in the data visualization to demonstrate how the strong the risk variables explored in this model are with regards to impacting their correspondent dependent variables.

Various multivariate regression analyses will be conducted afterwards. One model will consist of all of the EMs listed. Given China and their BRICS (Brazil, Russia, India, and South Africa), respectively have the highest GDP among the EM countries listed in this project, separate multivariate regression models were used, one without China, one without BRICS, and one with only BRICs. Doing so helped balance the influence the BRICS had affecting various independent variables in question and helped fathoming the impact Sovereign, Country, and Political risk has had on Commodity Prices and EMs' GDP output.

Exploring Relationships between Key Variables

In [659]:
```
import statsmodels.formula.api as smf
import statsmodels.api as sm

GDPvPrices.plot.scatter(x='Commodity Prices', y='GDP', figsize=(10,6))
GDPvPrices['Commodity Prices'].corr(GDPvPrices['GDP'])
```

Out[659]: 0.8916691310998767



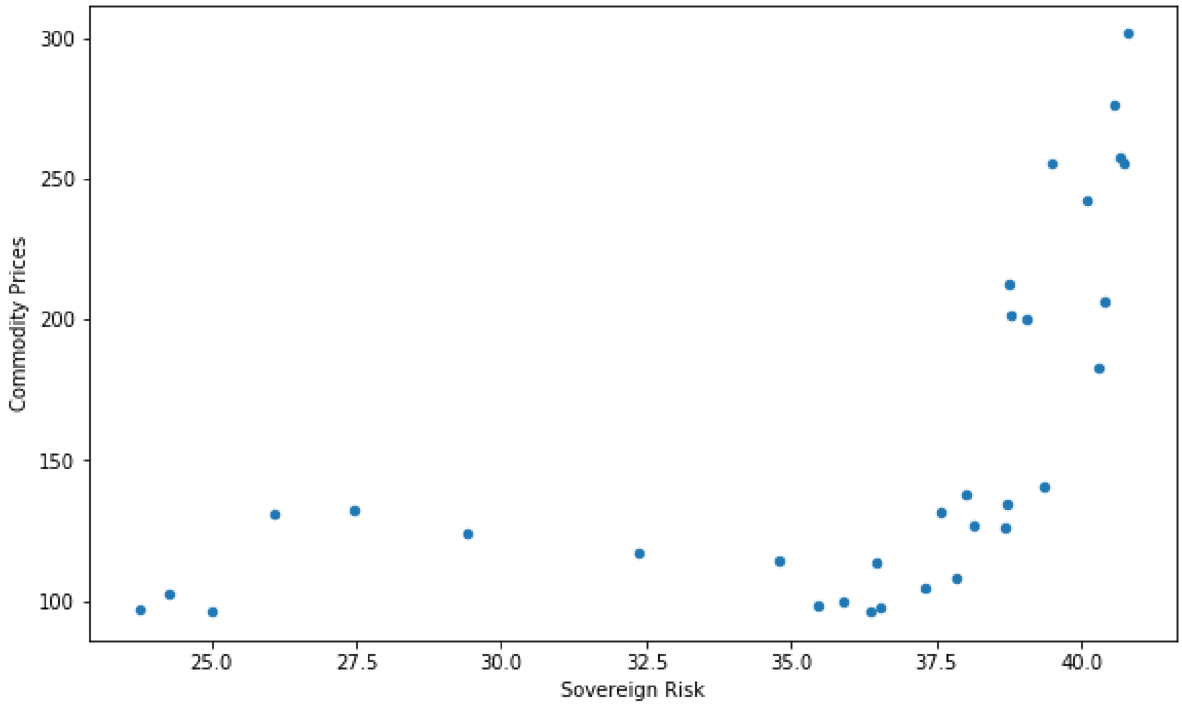
In [660]:
```
x=GDPvPrices['Commodity Prices']
y=GDPvPrices['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.79
5
Model:                            OLS   Adj. R-squared:                  0.78
8
Method:                 Least Squares   F-statistic:                     116.
4
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           7.59e-1
2
Time:                        22:41:29   Log-Likelihood:                -210.1
3
No. Observations:                  32   AIC:                             424.
3
Df Residuals:                      30   BIC:                             427.
2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=======
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const             -401.6992     85.430     -4.702      0.000    -576.170    -
227.228
Commodity Prices     5.4586      0.506     10.789      0.000       4.425
6.492
==============================================================================
=
Omnibus:                        9.766   Durbin-Watson:                   0.30
0
Prob(Omnibus):                  0.008   Jarque-Bera (JB):                8.40
8
Skew:                           1.123   Prob(JB):                       0.014
9
Kurtosis:                       4.123   Cond. No.                         45
9.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

```
C:\Users\ishaa\Anaconda3NEW1\lib\site-packages\numpy\core\fromnumeric.py:238
9: FutureWarning: Method .ptp is deprecated and will be removed in a future v
ersion. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

In [661]:
```
PriceRisk.plot.scatter(x='Sovereign Risk',y='Commodity Prices',figsize=(10,6))
PriceRisk['Sovereign Risk'].corr(PriceRisk['Commodity Prices'])
```

Out[661]: 0.5831224067321231

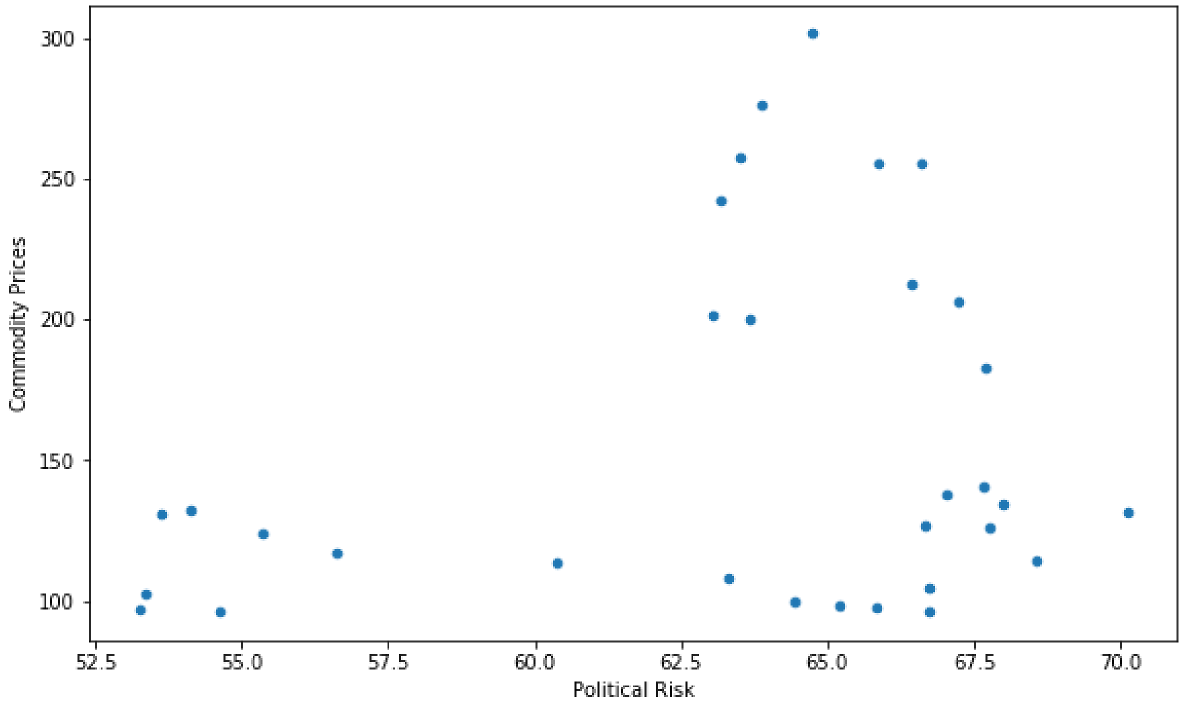


In [662]:
```
x=PriceRisk['Sovereign Risk']
y=PriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                             OLS Regression Results
=============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.34
0
Model:                            OLS   Adj. R-squared:                  0.31
8
Method:                 Least Squares   F-statistic:                     15.4
6
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           0.00046
1
Time:                        22:41:30   Log-Likelihood:                -170.8
6
No. Observations:                  32   AIC:                             345.
7
Df Residuals:                      30   BIC:                             348.
7
Df Model:                           1
Covariance Type:            nonrobust
=============================================================================
=====
                     coef    std err          t      P>|t|      [0.025
0.975]
-----------------------------------------------------------------------------
-----
const            -93.7816     64.454     -1.455      0.156    -225.415      3
7.852
Sovereign Risk     6.9848      1.777      3.932      0.000       3.356      1
0.613
=============================================================================
=
Omnibus:                        4.028   Durbin-Watson:                   0.19
1
Prob(Omnibus):                  0.133   Jarque-Bera (JB):                2.11
4
Skew:                           0.358   Prob(JB):                        0.34
8
Kurtosis:                       1.965   Cond. No.                         25
4.
=============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [663]:
```
PriceRisk.plot.scatter(x='Political Risk',y='Commodity Prices',figsize=(10,6))
PriceRisk['Political Risk'].corr(PriceRisk['Commodity Prices'])
```

Out[663]: 0.25313536002492953

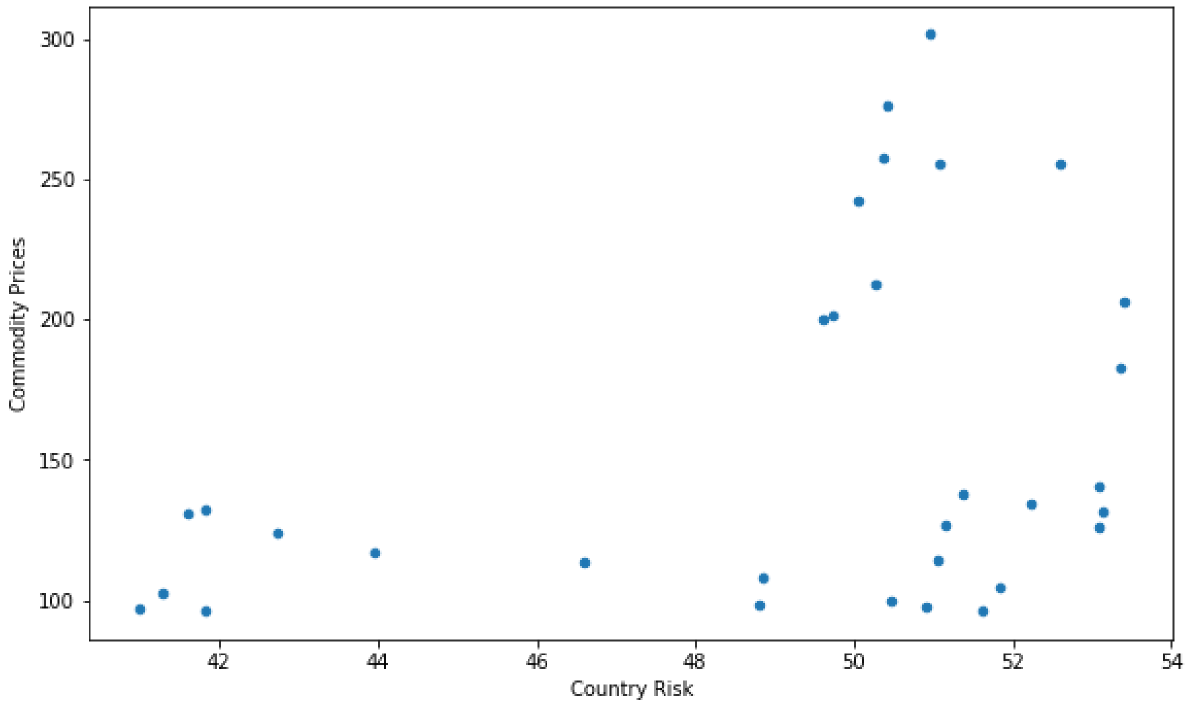


In [664]:
```
x=PriceRisk['Political Risk']
y=PriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.06
4
Model:                            OLS   Adj. R-squared:                  0.03
3
Method:                 Least Squares   F-statistic:                     2.05
4
Date:                Sat, 12 Oct 2019   Prob (F-statistic):              0.16
2
Time:                        22:41:30   Log-Likelihood:                -176.4
5
No. Observations:                  32   AIC:                             356.
9
Df Residuals:                      30   BIC:                             359.
8
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
-----
const            -38.2247    136.676     -0.280      0.782    -317.355     24
0.906
Political Risk     3.0851      2.153      1.433      0.162      -1.311       
7.481
==============================================================================
=
Omnibus:                        4.280   Durbin-Watson:                   0.14
5
Prob(Omnibus):                  0.118   Jarque-Bera (JB):                3.92
1
Skew:                           0.810   Prob(JB):                        0.14
1
Kurtosis:                       2.437   Cond. No.                         79
2.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [665]:
```
PriceRisk.plot.scatter(x='Country Risk',y='Commodity Prices',figsize=(10,6))
PriceRisk['Country Risk'].corr(PriceRisk['Commodity Prices'])
```

Out[665]: 0.36031849863290183



In [666]:
```
x=PriceRisk['Country Risk']
y=PriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.13
0
Model:                            OLS   Adj. R-squared:                  0.10
1
Method:                 Least Squares   F-statistic:                     4.47
6
Date:                Sat, 12 Oct 2019   Prob (F-statistic):             0.042
8
Time:                        22:41:31   Log-Likelihood:                -175.2
9
No. Observations:                  32   AIC:                             354.
6
Df Residuals:                      30   BIC:                             357.
5
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
===
                   coef    std err          t      P>|t|      [0.025      0.9
75]
------------------------------------------------------------------------------
---
const         -116.4289    129.683     -0.898      0.376    -381.277     148.
419
Country Risk     5.5726      2.634      2.116      0.043       0.193      10.
952
==============================================================================
=
Omnibus:                        3.435   Durbin-Watson:                   0.14
0
Prob(Omnibus):                  0.180   Jarque-Bera (JB):                3.07
3
Skew:                           0.693   Prob(JB):                        0.21
5
Kurtosis:                       2.381   Cond. No.                         60
4.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

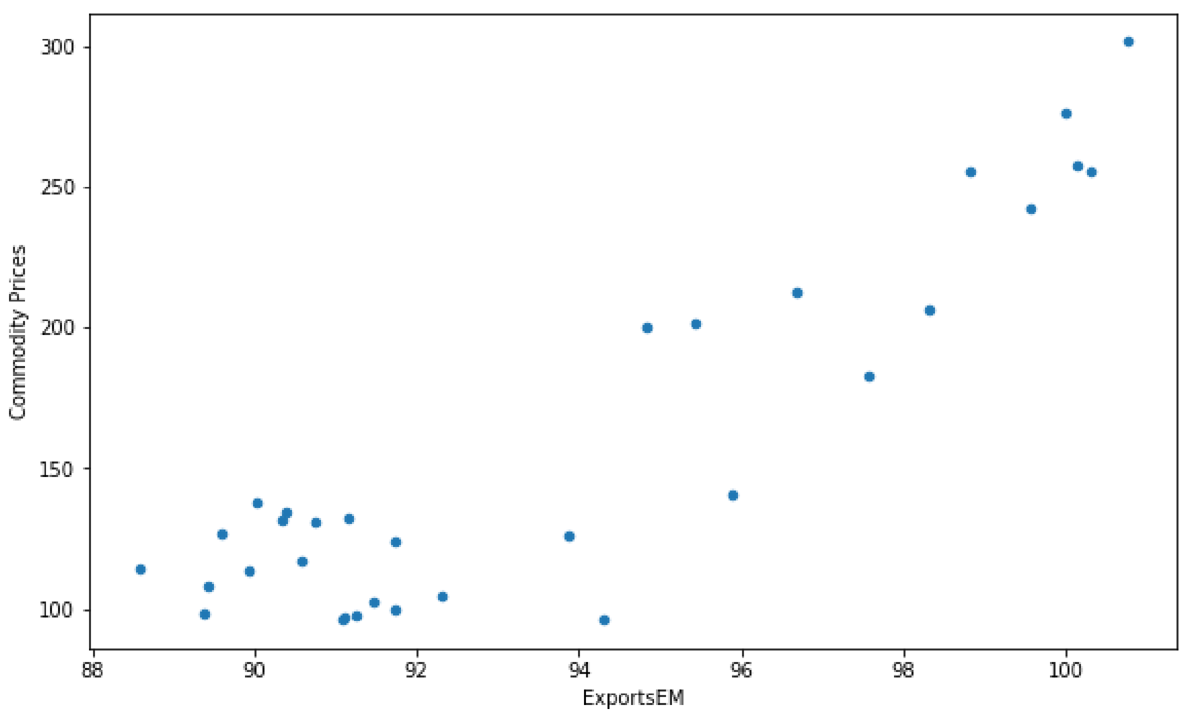
In [667]:
```
ImportsEM['Imports'].values.round(2)
```

Out[667]:
```
array([ 63.21,  50.17,  51.17,  50.21,  51.4 ,  52.15,  46.5 ,  45.04,
        42.92,  45.24,  47.52,  47.57,  47.77,  41.05,  41.74,  49.11,
        46.65,  47.37,  51.51,  59.23,  68.47,  80.36,  85.77, 100.88,
        77.56,  94.3 , 109.65, 100.  , 101.39,  96.84,  71.62,  70.38])
```

In [668]:
```
prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858
, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375,
97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.04
2, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]
ExportsEM=[ 94.29, 91.11, 91.47, 90.74, 91.15, 91.72, 90.57, 89.93, 89.42, 89.
6, 90.03, 90.38, 90.35, 88.58, 89.38, 91.73, 91.09, 91.25, 92.31, 93.86, 95.89
, 97.57, 98.3, 100.31, 96.68, 98.81, 100.75, 100.0, 100.13, 99.56, 95.42, 94.8
2]
ImportsIE=[ 96.56, 94.15, 94.53, 93.76, 94.16, 94.76, 93.87, 93.43, 93.04, 93.
05, 93.22, 93.61, 93.58, 92.2, 92.7, 94.53, 94.09, 94.24, 94.88, 95.91, 97.3,
98.36, 98.81, 100.08, 97.79, 99.25, 100.51, 100.0, 100.14, 99.78, 96.91, 96.53
]
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]
ImportsEM=[ 63.21,  50.17,  51.17,  50.21,  51.4 ,  52.15,  46.5 ,  45.04,
        42.92,  45.24,  47.52,  47.57,  47.77,  41.05,  41.74,  49.11,
        46.65,  47.37,  51.51,  59.23,  68.47,  80.36,  85.77, 100.88,
        77.56,  94.3 , 109.65, 100.  , 101.39,  96.84,  71.62,  70.38]
PriceIMEX=pd.DataFrame({'Commodity Prices':prices,'ExportsEM':ExportsEM,'Impor
tsIE':ImportsIE,'ImportsEM':ImportsEM},index=year)
```

In [669]:
```
PriceIMEX.plot.scatter(x='ExportsEM',y='Commodity Prices', figsize =(10,6))
PriceIMEX['ExportsEM'].corr(PriceIMEX['Commodity Prices'])
```

Out[669]:
```
0.901852225048027
```

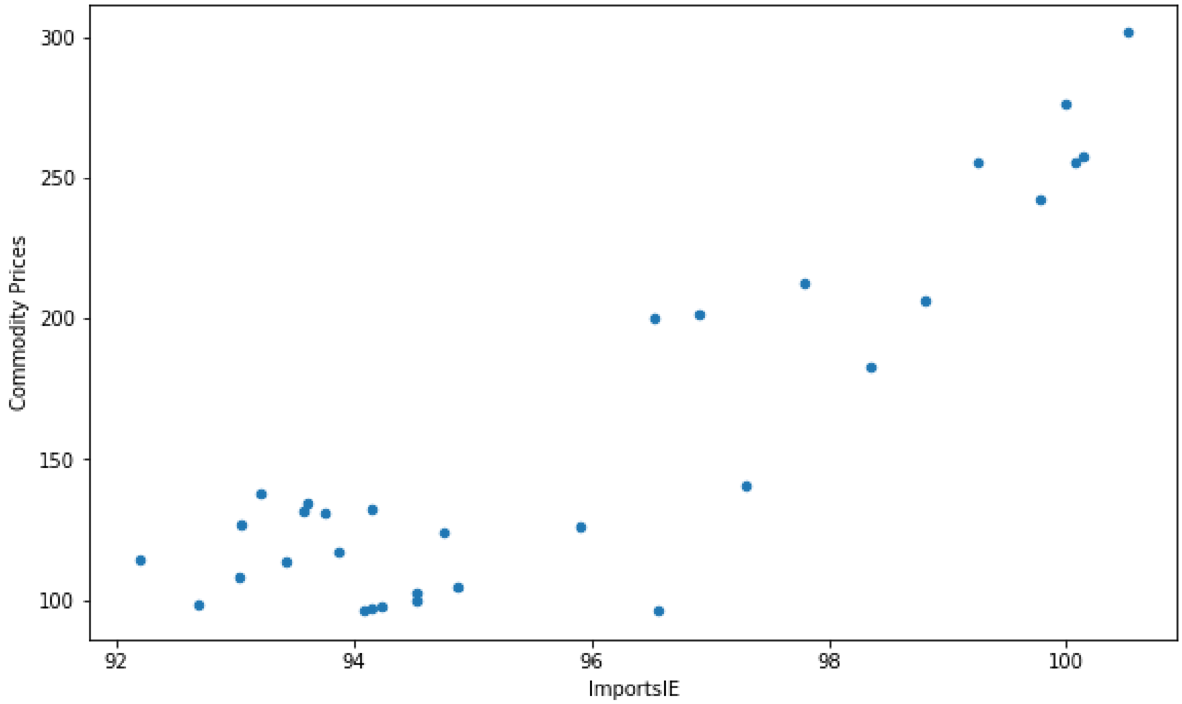


In [670]:
```
x=PriceIMEX['ExportsEM']
y=PriceIMEX['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.81
3
Model:                            OLS   Adj. R-squared:                  0.80
7
Method:                 Least Squares   F-statistic:                     130.
7
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           1.85e-1
2
Time:                        22:41:33   Log-Likelihood:                -150.6
6
No. Observations:                  32   AIC:                             305.
3
Df Residuals:                      30   BIC:                             308.
2
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const      -1198.9938    118.705    -10.101      0.000   -1441.421    -956.56
7
ExportsEM     14.4777      1.266     11.433      0.000      11.892      17.06
4
==============================================================================
=
Omnibus:                        2.494   Durbin-Watson:                   0.40
2
Prob(Omnibus):                  0.287   Jarque-Bera (JB):                2.20
9
Skew:                          -0.615   Prob(JB):                        0.33
1
Kurtosis:                       2.621   Cond. No.                     2.27e+0
3
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 2.27e+03. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

In [671]:
```
PriceIMEX.plot.scatter(x='ImportsIE',y='Commodity Prices', figsize =(10,6))
PriceIMEX['ImportsIE'].corr(PriceIMEX['Commodity Prices'])
```

Out[671]: 0.8920714851374041

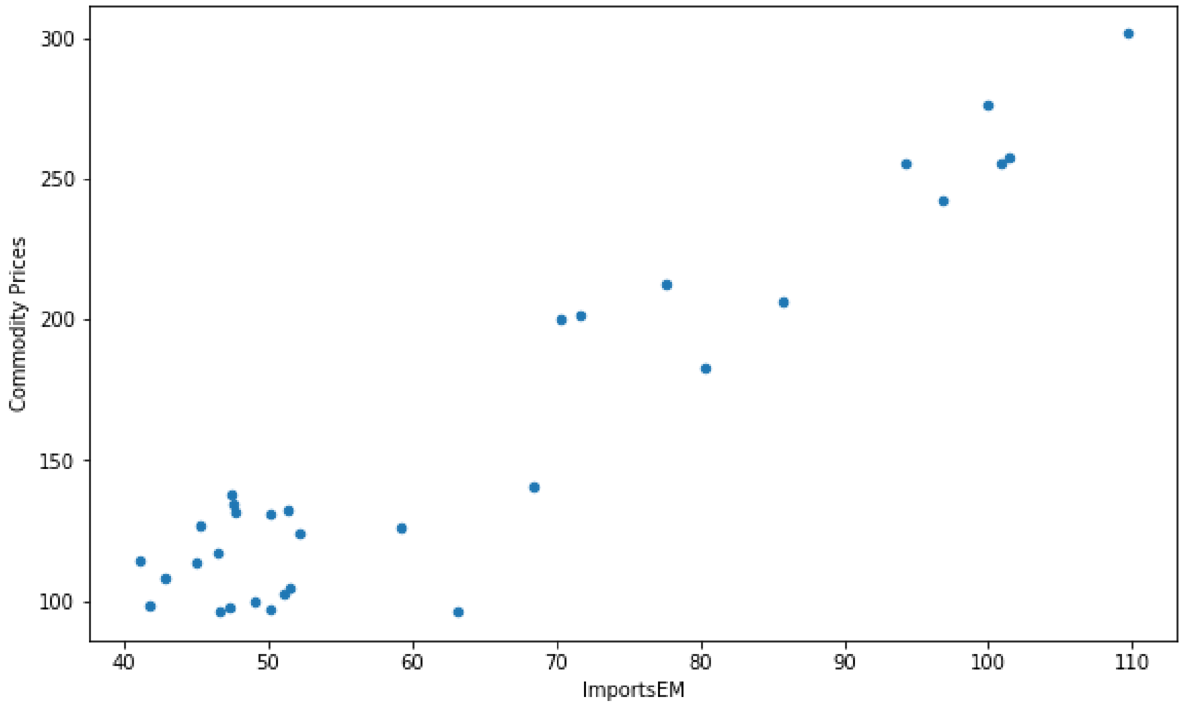


In [672]:
```
x=PriceIMEX['ImportsIE']
y=PriceIMEX['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.79
6
Model:                            OLS   Adj. R-squared:                  0.78
9
Method:                 Least Squares   F-statistic:                     116.
9
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           7.20e-1
2
Time:                        22:41:33   Log-Likelihood:                -152.0
9
No. Observations:                  32   AIC:                             308.
2
Df Residuals:                      30   BIC:                             311.
1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const      -1901.8706    190.488     -9.984      0.000   -2290.899   -1512.84
2
ImportsIE     21.4907      1.988     10.812      0.000      17.431      25.55
0
==============================================================================
=
Omnibus:                        2.859   Durbin-Watson:                   0.45
5
Prob(Omnibus):                  0.239   Jarque-Bera (JB):                2.29
5
Skew:                          -0.652   Prob(JB):                        0.31
7
Kurtosis:                       2.866   Cond. No.                     3.57e+0
3
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 3.57e+03. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

In [673]:
```
PriceIMEX.plot.scatter(x='ImportsEM',y='Commodity Prices', figsize =(10,6))
PriceIMEX['ImportsEM'].corr(PriceIMEX['Commodity Prices'])
```

Out[673]: 0.9485363204031945



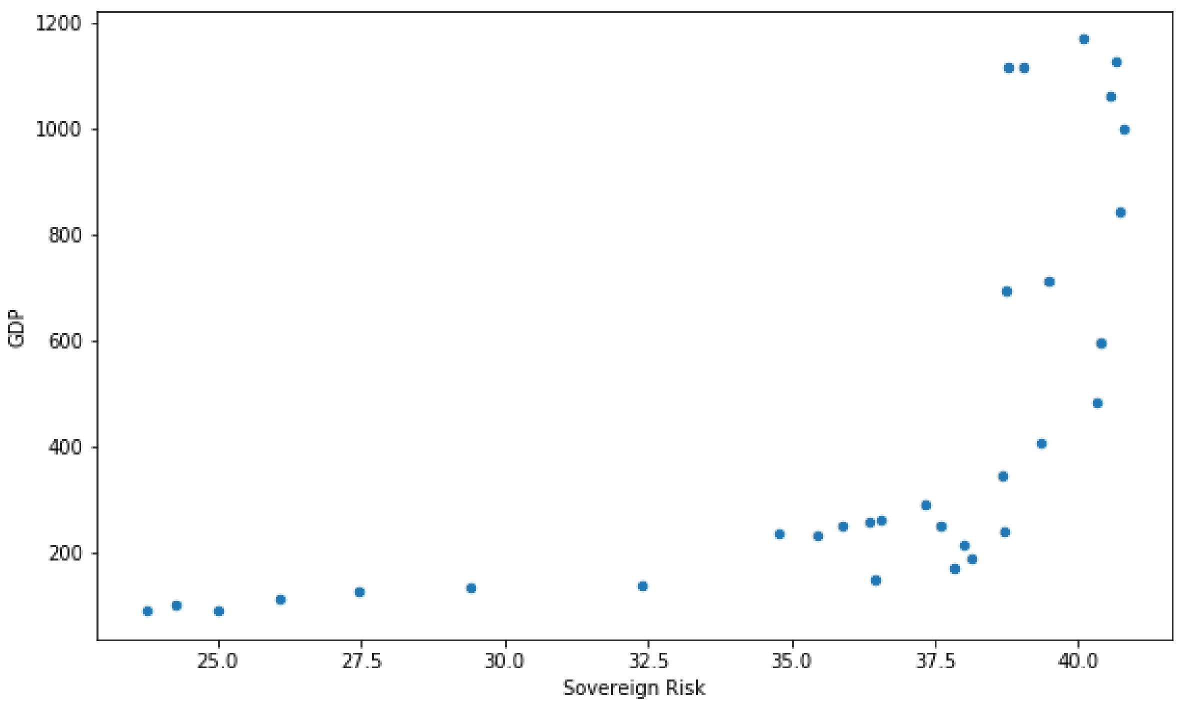
In [674]:
```
x=PriceIMEX['ImportsEM']
y=PriceIMEX['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.90
0
Model:                            OLS   Adj. R-squared:                  0.89
6
Method:                 Least Squares   F-statistic:                     269.
2
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           1.58e-1
6
Time:                        22:41:34   Log-Likelihood:                -140.7
2
No. Observations:                  32   AIC:                             285.
4
Df Residuals:                      30   BIC:                             288.
4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const        -20.1673     11.381     -1.772      0.087     -43.410       3.07
5
ImportsEM      2.7866      0.170     16.406      0.000       2.440       3.13
4
==============================================================================
=
Omnibus:                        5.349   Durbin-Watson:                   0.45
7
Prob(Omnibus):                  0.069   Jarque-Bera (JB):                3.91
2
Skew:                          -0.820   Prob(JB):                        0.14
1
Kurtosis:                       3.493   Cond. No.                         21
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [675]:
```
GDP=[ 89.31, 89.78, 98.62, 112.04, 124.48, 132.34, 136.18, 147.69, 168.29, 18
7.02, 212.6, 236.54, 249.62, 234.79, 229.87, 249.32, 254.73, 258.71, 290.87, 3
43.04, 406.34, 481.11, 594.03, 709.16, 693.03, 840.37, 996.64, 1059.71, 1126.8
5, 1167.42, 1115.26, 1114.06]
SovereignRisk=[ 25.0, 23.76, 24.28, 26.08, 27.45, 29.4, 32.38, 36.46, 37.83, 3
8.13, 38.0, 38.71, 37.59, 34.79, 35.45, 35.88, 36.36, 36.55, 37.32, 38.69, 39.
36, 40.31, 40.38, 39.49, 38.74, 40.73, 40.79, 40.56, 40.65, 40.08, 38.79, 39.0
4]
PoliticalRisk=[ 54.61, 53.25, 53.35, 53.61, 54.13, 55.37, 56.62, 60.37, 63.31,
66.68, 67.05, 68.0, 70.12, 68.56, 65.19, 64.43, 66.72, 65.84, 66.74, 67.77, 6
7.67, 67.69, 67.25, 66.6, 66.45, 65.86, 64.73, 63.87, 63.51, 63.17, 63.03, 63.
67]
CountryRisk=[ 41.81, 40.98, 41.29, 41.6, 41.82, 42.73, 43.96, 46.59, 48.84, 5
1.15, 51.37, 52.23, 53.15, 51.06, 48.8, 50.46, 51.62, 50.91, 51.85, 53.08, 53.
08, 53.36, 53.4, 52.61, 50.27, 51.08, 50.96, 50.43, 50.36, 50.04, 49.74, 49.62
]
GDPRisk=pd.DataFrame({'GDP':GDP,'Sovereign Risk':SovereignRisk,'Political Ris
k':PoliticalRisk,'Country Risk':CountryRisk})
```

In [676]:
```
GDPRisk.plot.scatter(x='Sovereign Risk',y='GDP', figsize =(10,6))
GDPRisk['Sovereign Risk'].corr(GDPRisk['GDP'])
```

Out[676]: 0.6342345613672907



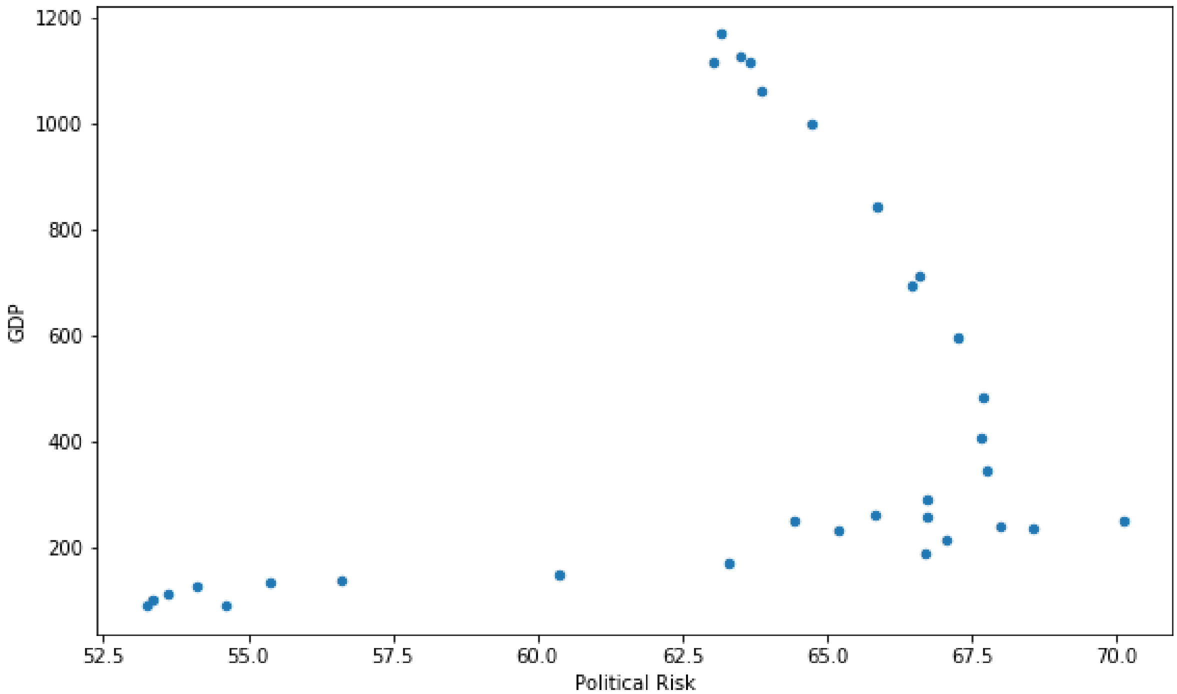
In [677]:

```
x=PriceRisk['Sovereign Risk']
y=GDPvPrices['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.39
8
Model:                            OLS   Adj. R-squared:                  0.37
7
Method:                 Least Squares   F-statistic:                     19.8
0
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           0.00011
0
Time:                        22:41:34   Log-Likelihood:                -227.3
8
No. Observations:                  32   AIC:                             458.
8
Df Residuals:                      30   BIC:                             461.
7
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
-----
const          -1204.7459    376.987     -3.196      0.003   -1974.656     -43
4.836
Sovereign Risk    46.2350     10.391      4.449      0.000      25.013       6
7.457
==============================================================================
=
Omnibus:                        3.747   Durbin-Watson:                   0.06
3
Prob(Omnibus):                  0.154   Jarque-Bera (JB):                2.89
6
Skew:                           0.604   Prob(JB):                        0.23
5
Kurtosis:                       2.157   Cond. No.                         25
4.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [678]:
```
GDPRisk.plot.scatter(x='Political Risk',y='GDP', figsize =(10,6))
GDPRisk['Political Risk'].corr(GDPRisk['GDP'])
```

Out[678]: 0.3074780820557676

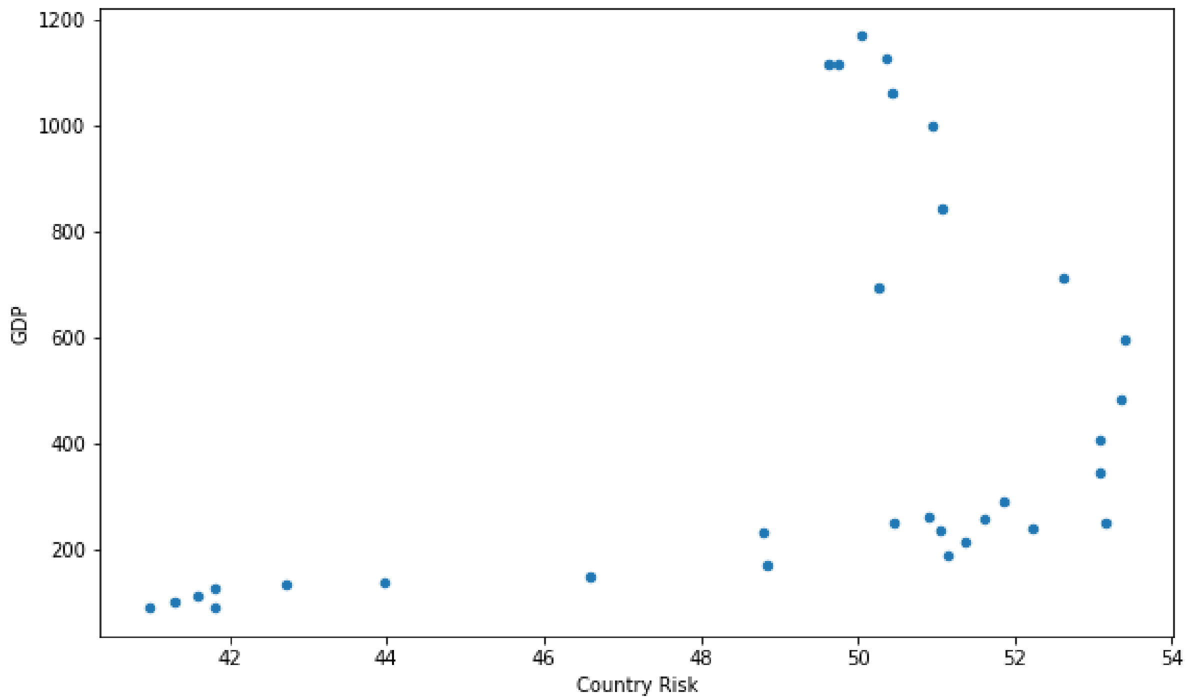


In [679]:
```
x=PriceRisk['Political Risk']
y=GDPvPrices['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
=============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.09
2
Model:                            OLS   Adj. R-squared:                  0.06
1
Method:                 Least Squares   F-statistic:                     3.02
9
Date:                Sat, 12 Oct 2019   Prob (F-statistic):             0.092
0
Time:                        22:41:35   Log-Likelihood:                -233.9
5
No. Observations:                  32   AIC:                             471.
9
Df Residuals:                      30   BIC:                             474.
8
Df Model:                           1
Covariance Type:            nonrobust
=============================================================================
=====
                     coef    std err          t      P>|t|      [0.025
0.975]
-----------------------------------------------------------------------------
-----
const           -974.6014    824.253     -1.182      0.246   -2657.951      70
8.748
Political Risk    22.5955     12.982      1.741      0.092      -3.917       4
9.108
=============================================================================
=
Omnibus:                        6.520   Durbin-Watson:                   0.03
7
Prob(Omnibus):                  0.038   Jarque-Bera (JB):                6.32
6
Skew:                           1.073   Prob(JB):                       0.042
3
Kurtosis:                       2.627   Cond. No.                         79
2.
=============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [680]:
```
GDPRisk.plot.scatter(x='Country Risk',y='GDP', figsize =(10,6))
GDPRisk['Country Risk'].corr(GDPRisk['GDP'])
```

Out[680]: 0.4150632125706353



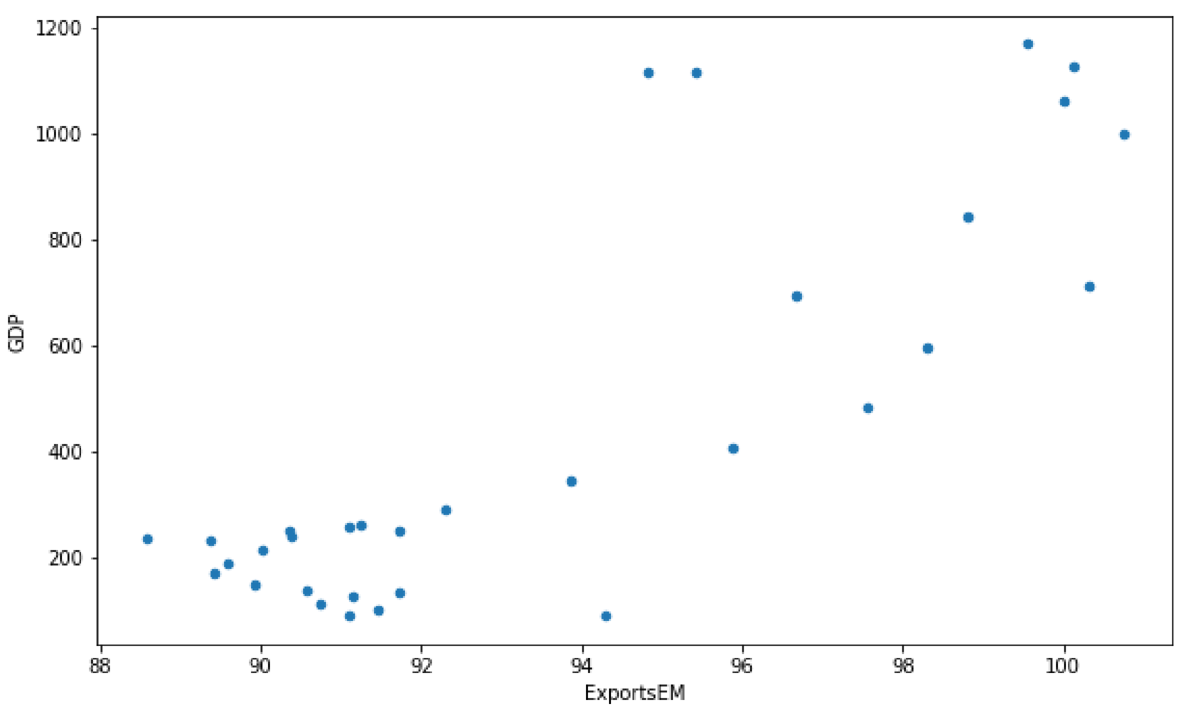
In [681]:
```
x=PriceRisk['Country Risk']
y=GDPvPrices['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.16
8
Model:                            OLS   Adj. R-squared:                  0.14
1
Method:                 Least Squares   F-statistic:                     6.07
5
Date:                Sat, 12 Oct 2019   Prob (F-statistic):             0.019
7
Time:                        22:41:35   Log-Likelihood:                -232.5
4
No. Observations:                  32   AIC:                             469.
1
Df Residuals:                      30   BIC:                             472.
0
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
===
                   coef    std err          t      P>|t|      [0.025      0.9
75]
------------------------------------------------------------------------------
---
const        -1451.1402    776.093     -1.870      0.071   -3036.134     133.
854
Country Risk    38.8537     15.763      2.465      0.020       6.661      71.
047
==============================================================================
=
Omnibus:                        5.955   Durbin-Watson:                   0.03
9
Prob(Omnibus):                  0.051   Jarque-Bera (JB):                5.69
1
Skew:                           1.014   Prob(JB):                       0.058
1
Kurtosis:                       2.601   Cond. No.                         60
4.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [682]:
```
GDP=[ 89.31, 89.78, 98.62, 112.04, 124.48, 132.34, 136.18, 147.69, 168.29, 18
7.02, 212.6, 236.54, 249.62, 234.79, 229.87, 249.32, 254.73, 258.71, 290.87, 3
43.04, 406.34, 481.11, 594.03, 709.16, 693.03, 840.37, 996.64, 1059.71, 1126.8
5, 1167.42, 1115.26, 1114.06]
ExportsEM=[ 94.29, 91.11, 91.47, 90.74, 91.15, 91.72, 90.57, 89.93, 89.42, 89.
6, 90.03, 90.38, 90.35, 88.58, 89.38, 91.73, 91.09, 91.25, 92.31, 93.86, 95.89
, 97.57, 98.3, 100.31, 96.68, 98.81, 100.75, 100.0, 100.13, 99.56, 95.42, 94.8
2]
ImportsIE=[ 96.56, 94.15, 94.53, 93.76, 94.16, 94.76, 93.87, 93.43, 93.04, 93.
05, 93.22, 93.61, 93.58, 92.2, 92.7, 94.53, 94.09, 94.24, 94.88, 95.91, 97.3,
98.36, 98.81, 100.08, 97.79, 99.25, 100.51, 100.0, 100.14, 99.78, 96.91, 96.53
]
ImportsEM=[ 63.21,  50.17,  51.17,  50.21,  51.4 ,  52.15,  46.5 ,  45.04,
        42.92,  45.24,  47.52,  47.57,  47.77,  41.05,  41.74,  49.11,
        46.65,  47.37,  51.51,  59.23,  68.47,  80.36,  85.77, 100.88,
        77.56,  94.3 , 109.65, 100.  , 101.39,  96.84,  71.62,  70.38]
GDPXIM=pd.DataFrame({'GDP':GDP, 'ExportsEM':ExportsEM,'ImportsIE':ImportsIE,'I
mportsEM':ImportsEM})
```

In [683]:
```
GDPXIM.plot.scatter(x='ExportsEM',y='GDP', figsize =(10,6))
GDPXIM['ExportsEM'].corr(GDPXIM['GDP'])
```

Out[683]: 0.8307601227265224

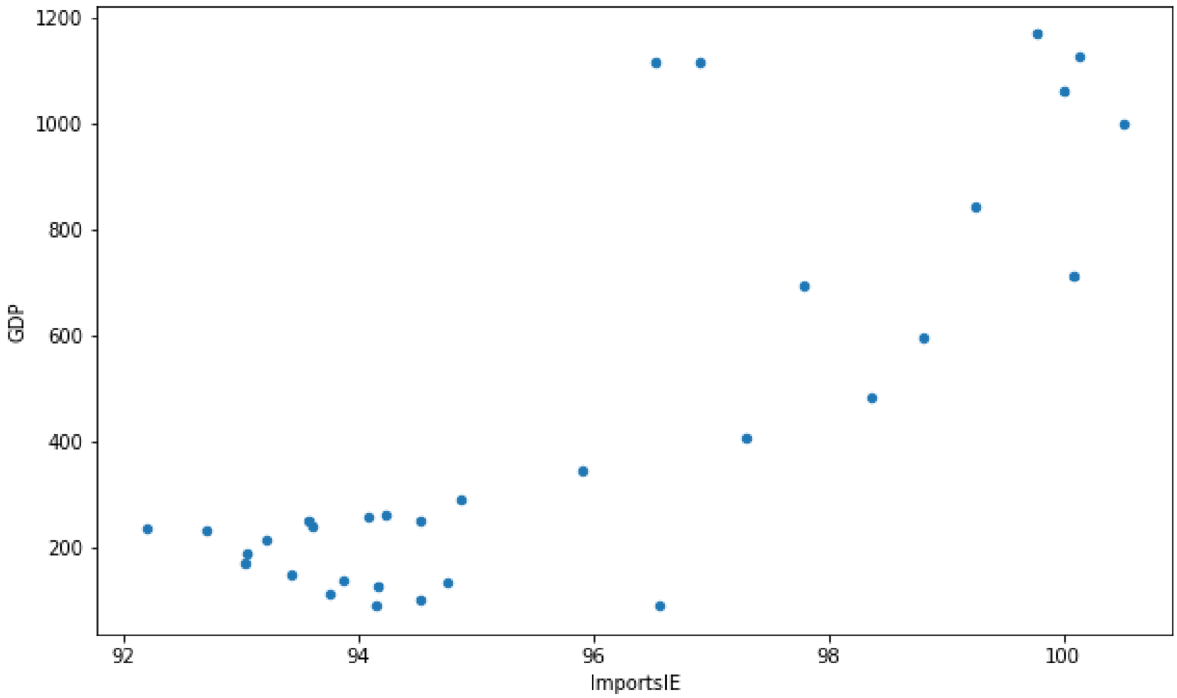


In [684]:
```
x=PriceIMEX['ExportsEM']
y=GDPvPrices['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.68
6
Model:                            OLS   Adj. R-squared:                  0.67
6
Method:                 Least Squares   F-statistic:                     65.5
7
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           4.87e-0
9
Time:                        22:41:36   Log-Likelihood:                -216.9
5
No. Observations:                  32   AIC:                             437.
9
Df Residuals:                      30   BIC:                             440.
8
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const      -7168.8330    942.365     -7.607      0.000   -9093.398   -5244.26
8
ExportsEM     81.4014     10.053      8.097      0.000      60.871     101.93
2
==============================================================================
=
Omnibus:                        9.095   Durbin-Watson:                   0.25
1
Prob(Omnibus):                  0.011   Jarque-Bera (JB):                7.87
7
Skew:                           0.946   Prob(JB):                       0.019
5
Kurtosis:                       4.527   Cond. No.                     2.27e+0
3
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 2.27e+03. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

In [685]:
```
GDPXIM.plot.scatter(x='ImportsIE',y='GDP', figsize =(10,6))
GDPXIM['ImportsIE'].corr(GDPXIM['GDP'])
```

Out[685]: 0.8207145722211843

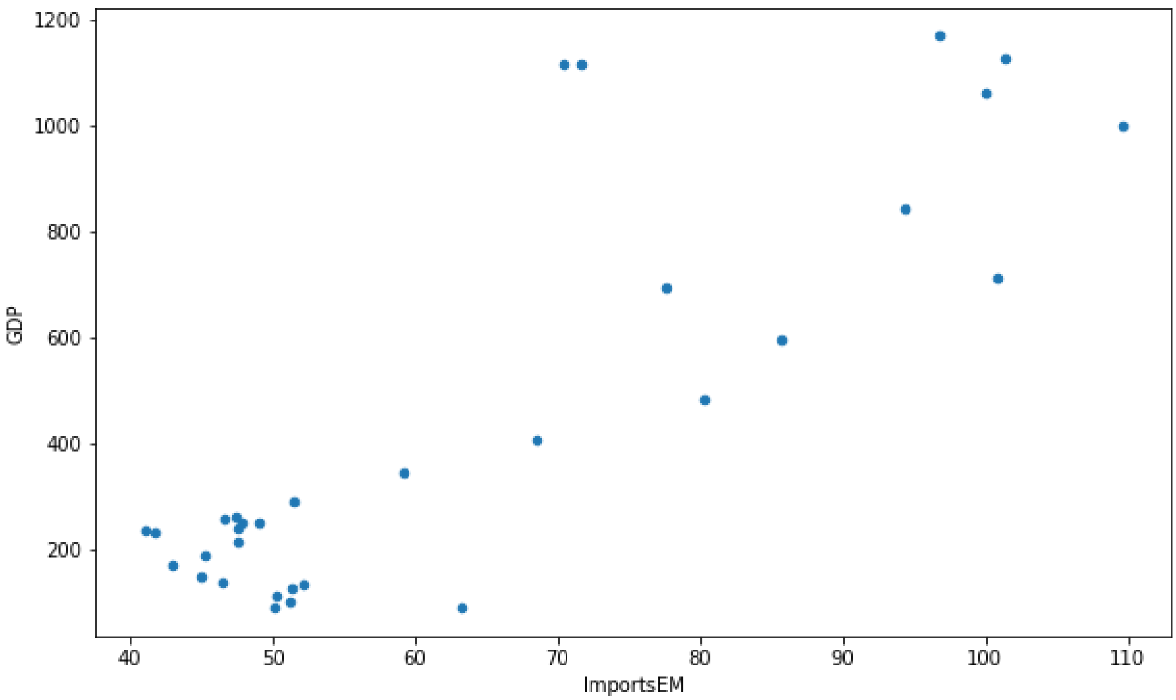


In [686]:
```
x=PriceIMEX['ImportsIE']
y=GDPvPrices['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.67
0
Model:                            OLS   Adj. R-squared:                  0.65
9
Method:                 Least Squares   F-statistic:                     60.8
4
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           1.05e-0
8
Time:                        22:41:37   Log-Likelihood:                -217.7
6
No. Observations:                  32   AIC:                             439.
5
Df Residuals:                      30   BIC:                             442.
5
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const      -1.111e+04   1482.936     -7.490      0.000   -1.41e+04   -8079.04
1
ImportsIE    120.6945     15.473      7.800      0.000      89.094     152.29
5
==============================================================================
=
Omnibus:                        8.306   Durbin-Watson:                   0.26
5
Prob(Omnibus):                  0.016   Jarque-Bera (JB):                6.94
7
Skew:                           0.882   Prob(JB):                       0.031
0
Kurtosis:                       4.448   Cond. No.                     3.57e+0
3
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 3.57e+03. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

In [687]:
```
GDPXIM.plot.scatter(x='ImportsEM',y='GDP', figsize =(10,6))
GDPXIM['ImportsEM'].corr(GDPXIM['GDP'])
```

Out[687]: 0.85165145111072



In [688]:
```
x=PriceIMEX['ImportsEM']
y=GDPvPrices['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.72
2
Model:                            OLS   Adj. R-squared:                  0.71
3
Method:                 Least Squares   F-statistic:                     77.8
4
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           7.78e-1
0
Time:                        22:41:37   Log-Likelihood:                -215.0
2
No. Observations:                  32   AIC:                             434.
0
Df Residuals:                      30   BIC:                             437.
0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=
                 coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
-
const       -516.1371    116.045     -4.448      0.000    -753.132    -279.14
2
ImportsEM     15.2795      1.732      8.822      0.000      11.743      18.81
6
==============================================================================
=
Omnibus:                       14.624   Durbin-Watson:                   0.31
9
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               16.63
3
Skew:                           1.270   Prob(JB):                     0.00024
5
Kurtosis:                       5.454   Cond. No.                         21
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

NOTES:

-As expected earlier, imports of commodities by industrialized countries and emerging markets, and exports of commodities by emerging markets have a high correlation and r square among commodity prices.

-Among the risk variables explored Commodity Prices as the dependent variable, we see that Sovereign Risk seems to have more of an impact on Commodity prices than Political and Country RIsk.

-It is also the same scenario with GDP as well as Sovereign Risk is the strongest risk indicator factor with regards to influencing EM GDP output.

-Previously explained, Exports of Commodities by EM and Imports of Commodities by Industrialized

Multivariate Regression Analysis with China and other BRICS Economies

It is worth noting that Sovereign Risk seems to be the biggest risk factor with regards to fluctuations commodity prices and EMs' economic output. Afterall, commodity prices are greatly affected by global liquidity levels and various financial indicators, many of which were calculated in the Sovereign Risk rating presented by the ICRG. Perhaps Sovereign Risk is a bigger risk factor to worry about when taking in fluctuations of EM's GDP output than other political and economic factors?

In [689]:
```
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]
prices=[96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858,
108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375, 9
7.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.042
, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]
SovereignRisk=[ 24.96, 23.64, 24.13, 25.85, 27.31, 29.33, 32.4, 36.59, 37.91,
38.23, 38.07, 38.73, 37.63, 34.8, 35.47, 36.06, 36.52, 36.7, 37.4, 38.85, 39.6
1, 40.58, 40.72, 39.83, 39.0, 41.26, 41.34, 41.08, 41.07, 40.61, 39.34, 39.55]
PoliticalRisk=[ 54.31, 52.98, 53.03, 53.07, 53.74, 55.1, 56.23, 60.13, 63.2, 6
6.48, 66.68, 67.5, 69.59, 67.82, 64.56, 63.68, 66.13, 65.3, 66.18, 67.31, 67.1
9, 67.22, 66.78, 66.2, 66.12, 65.67, 64.59, 63.72, 63.43, 63.03, 62.67, 63.33]
CountryRisk=[ 41.61, 40.83, 41.11, 41.33, 41.56, 42.6, 43.82, 46.52, 48.8, 51.
08, 51.18, 51.99, 52.89, 50.62, 48.35, 50.02, 51.24, 50.56, 51.55, 52.87, 52.8
9, 53.19, 53.24, 52.48, 50.18, 51.21, 51.16, 50.61, 50.48, 50.05, 49.54, 49.48
]
ExportsEM=[ 94.05, 90.76, 91.12, 90.37, 90.79, 91.38, 90.2, 89.55, 89.02, 89.2
, 89.65, 90.02, 89.98, 88.16, 88.97, 91.41, 90.75, 90.91, 92.02, 93.63, 95.73,
97.47, 98.23, 100.32, 96.56, 98.76, 100.78, 100.0, 100.13, 99.54, 95.25, 94.62
]
ImportsIE=[ 96.56, 94.15, 94.53, 93.76, 94.16, 94.76, 93.87, 93.43, 93.04, 93.
05, 93.22, 93.61, 93.58, 92.2, 92.7, 94.53, 94.09, 94.24, 94.88, 95.91, 97.3,
98.36, 98.81, 100.08, 97.79, 99.25, 100.51, 100.0, 100.14, 99.78, 96.91, 96.53
]
ImportsEM=[63.21, 50.17, 51.17, 50.21, 51.4, 52.15, 46.5, 45.04, 42.92, 45.24,
47.52, 47.57, 47.77, 41.05, 41.74, 49.11, 46.65, 47.37, 51.51, 59.23, 68.47, 8
0.36, 85.77, 100.88, 77.56, 94.3, 109.65, 100.0, 101.39, 96.84, 71.62, 70.38]
PriceRiskXIM1=pd.DataFrame({'Commodity Prices':prices,'Sovereign Risk':Soverei
gnRisk,'Political Risk':PoliticalRisk,'Country Risk':CountryRisk,'ImportsIE':I
mportsIE,'ExportsEM':ExportsEM,'ImportsEM':ImportsEM},index=year)
```

In [945]:
```
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]

GDP=[91.49, 91.53, 100.36, 113.91, 126.86, 134.15, 137.78, 149.26, 171.26, 19
0.53, 216.39, 241.07, 254.93, 239.29, 233.89, 255.18, 260.65, 263.93, 295.3, 3
48.16, 414.25, 491.49, 607.78, 726.82, 711.14, 867.39, 1032.07, 1100.41, 1171.
2, 1213.92, 1161.19, 1160.0]
prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858
, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375,
97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.04
2, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]

GDPvPrices1=pd.DataFrame({'GDP':GDP, 'Commodity Prices':prices}, index=year)
```

In [691]:
```
x=PriceRiskXIM1[['Commodity Prices','Sovereign Risk', 'Political Risk', 'Count
ry Risk','ImportsIE','ExportsEM','ImportsEM']]
y=GDPvPrices1['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.85
0
Model:                            OLS   Adj. R-squared:                  0.80
7
Method:                 Least Squares   F-statistic:                     19.4
6
Date:                Sat, 12 Oct 2019   Prob (F-statistic):          1.85e-0
8
Time:                        22:41:38   Log-Likelihood:                -205.1
2
No. Observations:                  32   AIC:                             426.
2
Df Residuals:                      24   BIC:                             438.
0
Df Model:                           7                                         
Covariance Type:            nonrobust                                         
==============================================================================
=======
                      coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
-------
const             2070.1115   1.51e+04      0.137      0.892   -2.91e+04    
3.33e+04
Commodity Prices     4.4420      3.140      1.415      0.170      -2.039      
10.923
Sovereign Risk      45.3520     23.307      1.946      0.063      -2.751      
93.455
Political Risk      10.0881     44.722      0.226      0.823     -82.213     
102.389
Country Risk       -62.3319     69.585     -0.896      0.379    -205.949      
81.285
ImportsIE         -303.4657    374.514     -0.810      0.426   -1076.424     
469.493
ExportsEM          309.0725    255.624      1.209      0.238    -218.510     
836.655
ImportsEM          -20.9769     20.790     -1.009      0.323     -63.886      
21.932
==============================================================================
=
Omnibus:                        4.464   Durbin-Watson:                   0.28
5
Prob(Omnibus):                  0.107   Jarque-Bera (JB):                3.51
6
Skew:                           0.811   Prob(JB):                        0.17
2
Kurtosis:                       3.096   Cond. No.                     1.20e+0
5
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

```
[2] The condition number is large, 1.2e+05. This might indicate that there ar
e
strong multicollinearity or other numerical problems.
```

Multivariate Regression Analysis excluding China

In [692]:

```
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]
prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858
, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375,
97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.04
2, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]
SovereignRisk=[ 24.38, 23.12, 23.8, 25.62, 27.33, 29.6, 32.82, 36.82, 37.76, 3
8.19, 38.02, 38.76, 37.52, 34.3, 34.95, 35.62, 36.06, 36.27, 36.99, 38.55, 39.
27, 40.23, 40.34, 39.4, 38.54, 40.91, 40.99, 40.75, 40.73, 40.25, 38.9, 39.19]
PoliticalRisk=[ 53.56, 52.33, 52.56, 52.49, 53.4, 55.0, 56.13, 59.73, 62.78, 6
6.46, 66.6, 67.4, 69.63, 67.91, 64.67, 63.75, 66.34, 65.25, 66.01, 67.16, 67.0
8, 67.17, 66.65, 66.1, 66.08, 65.73, 64.79, 63.86, 63.54, 63.34, 63.0, 63.77]
CountryRisk=[ 41.02, 40.36, 40.7, 40.84, 41.27, 42.42, 43.58, 46.11, 48.55, 5
1.19, 51.17, 51.89, 52.81, 50.52, 48.19, 49.98, 51.29, 50.46, 51.42, 52.79, 5
2.81, 53.12, 53.13, 52.36, 49.98, 51.15, 51.2, 50.62, 50.48, 50.13, 49.6, 49.5
5]
ExportsEM=[ 93.75, 90.3, 90.67, 89.89, 90.33, 90.95, 89.71, 89.05, 88.51, 88.7
, 89.16, 89.55, 89.51, 87.6, 88.46, 91.01, 90.32, 90.49, 91.65, 93.33, 95.54,
97.36, 98.14, 100.34, 96.4, 98.7, 100.81, 100.0, 100.14, 99.52, 95.02, 94.36]
ImportsIE=[ 96.56, 94.15, 94.53, 93.76, 94.16, 94.76, 93.87, 93.43, 93.04, 93.
05, 93.22, 93.61, 93.58, 92.2, 92.7, 94.53, 94.09, 94.24, 94.88, 95.91, 97.3,
98.36, 98.81, 100.08, 97.79, 99.25, 100.51, 100.0, 100.14, 99.78, 96.91, 96.53
]
ImportsEM=[ 62.71,  49.83,  50.81,  49.85,  51.05,  51.81,  46.26,  44.7 ,
        42.7 ,  45.05,  47.31,  47.31,  47.59,  40.94,  41.59,  48.94,
        46.43,  47.13,  51.32,  59.04,  68.48,  80.33,  85.57, 100.65,
        77.53,  94.3 , 109.57, 100.  , 101.23,  96.79,  71.52,  70.22]
PriceRiskXIM2=pd.DataFrame({'Commodity Prices':prices,'Sovereign Risk':Soverei
gnRisk,'Political Risk':PoliticalRisk,'Country Risk':CountryRisk,'ImportsIE':I
mportsIE,'ExportsEM':ExportsEM,'ImportsEM':ImportsEM},index=year)
```

In [693]:
```
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,2016]

GDP=[79.2, 79.77, 87.58, 97.36, 108.3, 119.46, 122.35, 131.02, 147.48, 170.75, 188.99, 208.11, 217.54, 197.54, 188.45, 204.67, 203.63, 200.06, 222.9, 262.99, 314.53, 371.34, 451.8, 522.74, 479.0, 593.76, 690.49, 707.26, 725.73, 723.36, 631.45, 630.43]

prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375, 97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.042, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]

GDPvPrices2=pd.DataFrame({'GDP':GDP, 'Commodity Prices':prices}, index=year)
```

In [694]:
```
x=PriceRiskXIM2[['Commodity Prices','Sovereign Risk', 'Political Risk', 'Count
ry Risk','ImportsIE','ExportsEM','ImportsEM']]
y=GDPvPrices2['GDP']
x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.91
2
Model:                            OLS   Adj. R-squared:                  0.88
7
Method:                 Least Squares   F-statistic:                     35.7
2
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           3.47e-1
1
Time:                        22:41:38   Log-Likelihood:                -178.8
0
No. Observations:                  32   AIC:                             373.
6
Df Residuals:                      24   BIC:                             385.
3
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
=======
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const              561.3684   7061.930      0.079      0.937    -1.4e+04
1.51e+04
Commodity Prices     2.4663      1.407      1.753      0.092      -0.438
5.371
Sovereign Risk      15.4113      9.801      1.572      0.129      -4.817
35.639
Political Risk      12.7721     18.411      0.694      0.495     -25.226
50.770
Country Risk       -27.3955     28.303     -0.968      0.343     -85.809
31.018
ImportsIE         -120.3129    165.813     -0.726      0.475    -462.535
221.909
ExportsEM          122.2230    108.138      1.130      0.270    -100.964
345.410
ImportsEM           -7.9589      9.281     -0.858      0.400     -27.114
11.196
==============================================================================
=
Omnibus:                        2.968   Durbin-Watson:                   0.30
1
Prob(Omnibus):                  0.227   Jarque-Bera (JB):                2.41
0
Skew:                           0.668   Prob(JB):                        0.30
0
Kurtosis:                       2.853   Cond. No.                     1.28e+0
5
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

```
[2] The condition number is large, 1.28e+05. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

Multivariate Regression Analysis without BRICS countries

In [695]:
```
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]
prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858
, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375,
97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.04
2, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]
SovereignRisk=[ 23.97, 22.69, 23.22, 25.02, 27.13, 29.87, 33.51, 37.89, 38.93,
39.33, 38.94, 39.72, 37.96, 34.17, 35.6, 35.77, 36.06, 36.34, 36.79, 38.16, 3
8.78, 39.77, 39.99, 39.18, 38.25, 40.31, 40.67, 40.69, 40.73, 40.26, 39.1, 39.
02]
PoliticalRisk=[ 53.17, 51.63, 51.97, 51.58, 52.89, 55.04, 56.92, 60.22, 63.63,
67.33, 67.37, 68.03, 69.97, 68.49, 66.14, 65.02, 67.47, 66.3, 66.45, 67.12, 6
7.16, 67.37, 66.91, 66.3, 66.2, 65.84, 65.09, 64.11, 63.89, 63.94, 63.61, 64.5
]
CountryRisk=[ 40.94, 39.92, 40.42, 40.54, 41.12, 42.67, 44.29, 46.9, 49.36, 5
1.65, 51.56, 52.27, 53.2, 51.05, 49.35, 50.85, 51.92, 51.0, 51.86, 52.96, 52.9
3, 53.24, 53.29, 52.58, 50.31, 51.42, 51.6, 50.96, 50.9, 50.76, 50.38, 50.35]
ExportsEM=[ 92.95, 88.92, 89.4, 88.43, 88.95, 89.69, 88.29, 88.19, 87.59, 87.8
3, 88.35, 88.72, 88.67, 86.63, 87.53, 90.23, 89.45, 89.68, 90.93, 92.75, 95.13
, 97.17, 98.03, 100.37, 96.06, 98.55, 100.85, 100.0, 100.15, 99.49, 94.6, 93.8
7]
ImportsIE=[ 96.56, 94.15, 94.53, 93.76, 94.16, 94.76, 93.87, 93.43, 93.04, 93.
05, 93.22, 93.61, 93.58, 92.2, 92.7, 94.53, 94.09, 94.24, 94.88, 95.91, 97.3,
98.36, 98.81, 100.08, 97.79, 99.25, 100.51, 100.0, 100.14, 99.78, 96.91, 96.53
]
ImportsEM=[ 60.67,  46.23,  47.71,  46.65,  48.01,  48.97,  42.97,  42.22,
        39.85,  42.51,  45.2 ,  44.93,  44.93,  37.81,  38.51,  46.17,
        43.48,  44.45,  48.82,  57.38,  67.16,  80.57,  86.3 , 101.76,
        76.23,  94.1 , 110.3 , 100.  , 101.18,  96.57,  70.16,  68.8 ]
PriceRiskXIM3=pd.DataFrame({'Commodity Prices':prices,'Sovereign Risk':Soverei
gnRisk,'Political Risk':PoliticalRisk,'Country Risk':CountryRisk,'ImportsIE':I
mportsIE,'ExportsEM':ExportsEM,'ImportsEM':ImportsEM},index=year)
```

In [696]:
```
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]

GDP=[ 60.14, 56.82, 61.18, 69.17, 74.06, 83.53, 93.63, 105.84, 117.18, 128.56,
129.78, 142.45, 149.31, 135.43, 144.51, 156.14, 157.67, 152.16, 161.53, 183.01
, 211.72, 244.55, 283.91, 329.84, 297.44, 357.15, 414.72, 436.45, 453.64, 456.
07, 427.51, 420.92]
prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858
, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375,
97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.04
2, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]

GDPvPrices3=pd.DataFrame({'GDP':GDP, 'Commodity Prices':prices}, index=year)
```

In [697]:
```
x=PriceRiskXIM3[['Commodity Prices','Sovereign Risk', 'Political Risk', 'Count
ry Risk','ImportsIE','ExportsEM','ImportsEM']]
y=GDPvPrices3['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.90
0
Model:                            OLS   Adj. R-squared:                  0.87
0
Method:                 Least Squares   F-statistic:                     30.7
1
Date:                Sat, 12 Oct 2019   Prob (F-statistic):          1.74e-1
0
Time:                        22:41:38   Log-Likelihood:                -164.7
8
No. Observations:                  32   AIC:                             345.
6
Df Residuals:                      24   BIC:                             357.
3
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
=======
                      coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const             780.6359   4195.131      0.186      0.854   -7877.690    9
438.961
Commodity Prices    2.5573      0.837      3.056      0.005       0.830
4.284
Sovereign Risk      4.5622      5.693      0.801      0.431      -7.188
16.313
Political Risk     -1.8468     12.340     -0.150      0.882     -27.315
23.622
Country Risk       -5.4141     19.204     -0.282      0.780     -45.048
34.220
ImportsIE        -123.1664     91.209     -1.350      0.189    -311.413
65.081
ExportsEM         128.8083     59.337      2.171      0.040       6.343
251.273
ImportsEM         -13.9411      5.774     -2.414      0.024     -25.858
-2.024
==============================================================================
=
Omnibus:                        1.992   Durbin-Watson:                   0.44
0
Prob(Omnibus):                  0.369   Jarque-Bera (JB):                1.62
7
Skew:                           0.541   Prob(JB):                        0.44
3
Kurtosis:                       2.782   Cond. No.                     1.17e+0
5
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

```
[2] The condition number is large, 1.17e+05. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

Multivariate Regression Analysis with only BRICS countries

In [698]:

```
SovereignRisk=[ 28.67, 27.19, 27.56, 28.94, 28.0, 27.33, 28.23, 32.69, 34.87,
34.95, 35.43, 35.75, 36.67, 36.69, 35.07, 36.96, 37.89, 37.8, 39.26, 40.92, 4
2.1, 43.03, 42.9, 41.76, 41.27, 44.12, 43.32, 42.26, 42.09, 41.66, 40.07, 41.1
3]
PoliticalRisk=[ 58.6, 58.02, 56.98, 58.65, 56.94, 55.31, 53.65, 59.89, 61.9, 6
3.93, 64.63, 65.9, 68.43, 65.82, 59.8, 59.68, 62.11, 62.3, 65.39, 67.87, 67.28
, 66.78, 66.37, 65.88, 65.91, 65.13, 63.09, 62.56, 62.03, 60.3, 59.85, 59.8]
CountryRisk=[ 44.11, 44.24, 43.69, 44.29, 43.23, 42.32, 42.03, 45.41, 47.11, 4
9.34, 50.04, 51.15, 51.97, 49.33, 45.37, 47.54, 49.18, 49.26, 50.62, 52.61, 5
2.78, 53.04, 53.09, 52.18, 49.8, 50.58, 49.84, 49.58, 49.21, 47.91, 47.03, 46.
86]
ExportsEM=[ 98.17, 97.66, 97.58, 97.64, 97.69, 97.72, 97.35, 93.61, 93.31, 93.
32, 93.54, 93.9, 93.9, 92.75, 93.29, 94.94, 94.66, 94.59, 95.28, 96.27, 97.54,
98.38, 98.81, 100.19, 98.07, 99.41, 100.55, 100.0, 100.09, 99.71, 97.2, 96.87]
ImportsIE=[ 96.56, 94.15, 94.53, 93.76, 94.16, 94.76, 93.87, 93.43, 93.04, 93.
05, 93.22, 93.61, 93.58, 92.2, 92.7, 94.53, 94.09, 94.24, 94.88, 95.91, 97.3,
98.36, 98.81, 100.08, 97.79, 99.25, 100.51, 100.0, 100.14, 99.78, 96.91, 96.53
]
ImportsEM=[ 72.75,  64.97,  64.13,  63.57,  64.09,  64.06,  59.72,  53.49,
        52.15,  53.44,  54.49,  55.5 ,  56.3 ,  50.78,  51.44,  57.95,
        56.16,  56.16,  59.57,  64.8 ,  72.4 ,  79.73,  84.19,  98.26,
        81.56,  94.91, 107.72, 100.  , 102.02,  97.64,  76.02,  75.1 ]
PriceRiskXIM4=pd.DataFrame({'Commodity Prices':prices,'Sovereign Risk':Soverei
gnRisk,'Political Risk':PoliticalRisk,'Country Risk':CountryRisk,'ImportsIE':I
mportsIE,'ExportsEM':ExportsEM,'ImportsEM':ImportsEM},index=year)
```

In [699]:

```
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]

GDP=[ 209.05, 221.7, 247.27, 281.69, 324.89, 323.98, 303.38, 279.49, 333.48, 3
76.46, 476.23, 536.93, 571.81, 550.87, 502.03, 552.3, 569.57, 599.26, 696.62,
843.59, 1021.82, 1232.32, 1579.39, 1917.75, 1952.22, 2398.11, 2884.13, 3092.28
, 3323.87, 3487.46, 3362.22, 3377.22]
prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858
, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375,
97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.04
2, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]

GDPvPrices4=pd.DataFrame({'GDP':GDP, 'Commodity Prices':prices}, index=year)
```

In [700]:
```
x=PriceRiskXIM4[['Commodity Prices','Sovereign Risk', 'Political Risk', 'Count
ry Risk','ImportsIE','ExportsEM','ImportsEM']]
y=GDPvPrices4['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.90
5
Model:                            OLS   Adj. R-squared:                  0.87
8
Method:                 Least Squares   F-statistic:                     32.8
2
Date:                Sat, 12 Oct 2019   Prob (F-statistic):          8.59e-1
1
Time:                        22:41:39   Log-Likelihood:                -232.4
5
No. Observations:                  32   AIC:                             480.
9
Df Residuals:                      24   BIC:                             492.
6
Df Model:                           7                                         
Covariance Type:            nonrobust                                         
==============================================================================
=======
                      coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
-------
const            9647.2087   1.84e+04      0.526      0.604   -2.82e+04    
4.75e+04
Commodity Prices    8.3080      4.964      1.673      0.107      -1.938      
18.554
Sovereign Risk    172.0263     48.368      3.557      0.002      72.199     
271.853
Political Risk    -40.6530     82.298     -0.494      0.626    -210.509     
129.202
Country Risk     -163.0242    119.333     -1.366      0.185    -409.316      
83.267
ImportsIE         -25.4742    239.769     -0.106      0.916    -520.332     
469.384
ExportsEM         -43.2015    130.317     -0.332      0.743    -312.163     
225.760
ImportsEM          13.4044     42.013      0.319      0.752     -73.307     
100.116
==============================================================================
=
Omnibus:                        2.330   Durbin-Watson:                   0.53
3
Prob(Omnibus):                  0.312   Jarque-Bera (JB):                1.71
8
Skew:                           0.567   Prob(JB):                        0.42
4
Kurtosis:                       2.954   Cond. No.                     6.27e+0
4
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

```
[2] The condition number is large, 6.27e+04. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

As seen throughout the various regression models, all models portray high R2 values, showing that the variables we put in the model are accurate. In every multivariate regression model, we see a high coefficient Sovereign Risk among its other correspondent independent variables. Given that we have seen this similar trend when conducting simple correlations and bi-linear regressions, perhaps let’s delve into one BRIC EM economy and one non EM economy to see the effect sovereign risk really has on GDP output and commodity prices. India, Argentina, and China will each be examined closely to demonstrate how Sovereign, Country, and Economic Risk each greatly affect commodity prices and each respective GDPs.

India vs Argentina

India Analysis

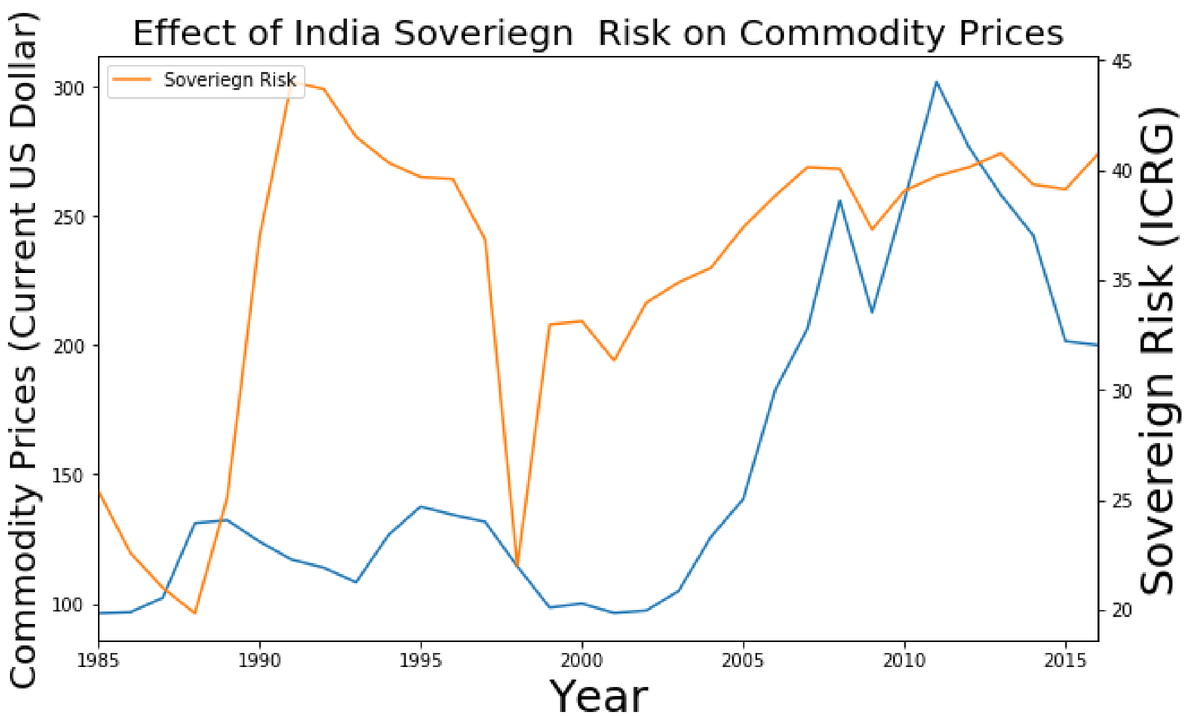
In [701]:
```
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]
prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858
, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375,
97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.04
2, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]
GDP=[237.618, 252.751, 283.75, 299.645, 300.187, 326.608,
        274.842, 293.262, 284.194, 333.014, 366.6, 399.791, 423.189,
        428.767, 466.841, 476.636, 493.934, 523.768, 618.369, 721.589,
        834.218, 949.118, 1238.7, 1224.1, 1365.37, 1708.46, 1823.05,
        1827.64, 1856.72, 2039.13, 2103.59, 2289.75]
SovereignRisk=[25.42,22.58,21.0,19.83,25.08,36.92,44.0,43.67,41.5,40.33,39.67,
39.58,36.83,22.0,32.96,33.12, 31.33, 33.96, 34.88, 35.54, 37.38, 38.83, 40.12,
40.04, 37.29, 39.04, 39.71, 40.12, 40.75, 39.33, 39.12, 40.71]
EconomicRisk=[ 30.34, 33.25, 32.12, 29.62, 30.75, 29.83, 27.33, 27.12,
        29.96, 34.62, 35.79, 36.29, 35.  , 33.04, 31.32, 33.58, 33.54,
        34.5 , 34.5 , 35.67, 35.5 , 35.54, 35.5 , 32.58, 33.5 , 32.92,
        33.38, 33.  , 32.54, 33.04, 35.17, 35.5 ]
PoliticalRisk=[ 47.83, 51.58, 49.08, 46.17, 42.58, 39.08, 34.75, 47.75,
        55.5 , 63.75, 64.08, 62.92, 65.5 , 60.58, 56.75, 54.92, 56.33,
        55.92, 58.46, 63.17, 63.71, 63.54, 62.46, 60.83, 63.17, 62.04,
        58.33, 57.62, 58.83, 60.46, 61.62, 62.88]
CountryRisk=[  39.085, 42.415, 40.6,   37.895, 36.665, 34.455, 31.04,  37.435,
42.73,
 49.185, 49.935, 49.605, 50.25,  46.81,  44.035, 44.25,  44.935, 45.21,  46.48
,
 49.42,  49.605, 49.54,  48.98,  46.705, 48.335, 47.48,  45.855, 45.31,  45.68
5,
 46.75,  48.395, 49.19 ]
Imports=[ 65.15,  52.63,  52.03,  48.17,  50.35,  51.41,  46.24,  44.91,
         43.18,  44.81,  45.69,  46.85,  49.94,  43.44,  44.42,  51.5 ,
         48.27,  47.4 ,  51.21,  55.66,  64.68,  71.98,  75.15,  93.23,
         79.42,  95.98, 109.79, 100.  , 103.85,  98.82,  74.29,  71.92]
Exports=[ 98.85,  98.19,  98.24,  98.01,  98.16,  98.29,  98.02,  97.95,
         97.83,  97.87,  97.93,  98.02,  98.11,  97.73,  97.88,  98.32,
         98.17,  98.16,  98.34,  98.59,  98.98,  99.26,  99.37,  99.86,
         99.4 ,  99.85, 100.19, 100.  , 100.07,  99.94,  99.18,  99.09]
IndGDPPriceRisk=pd.DataFrame({'CountryRisk':CountryRisk,'PoliticalRisk':Politi
calRisk,'EconomicRisk':EconomicRisk,'SovereignRisk':SovereignRisk,'GDP':GDP,'C
ommodity Prices':prices,'Imports':Imports,'Exports':Exports},index=year)
```

In [702]:
```
IndGDPPriceRisk
```

Out[702]:

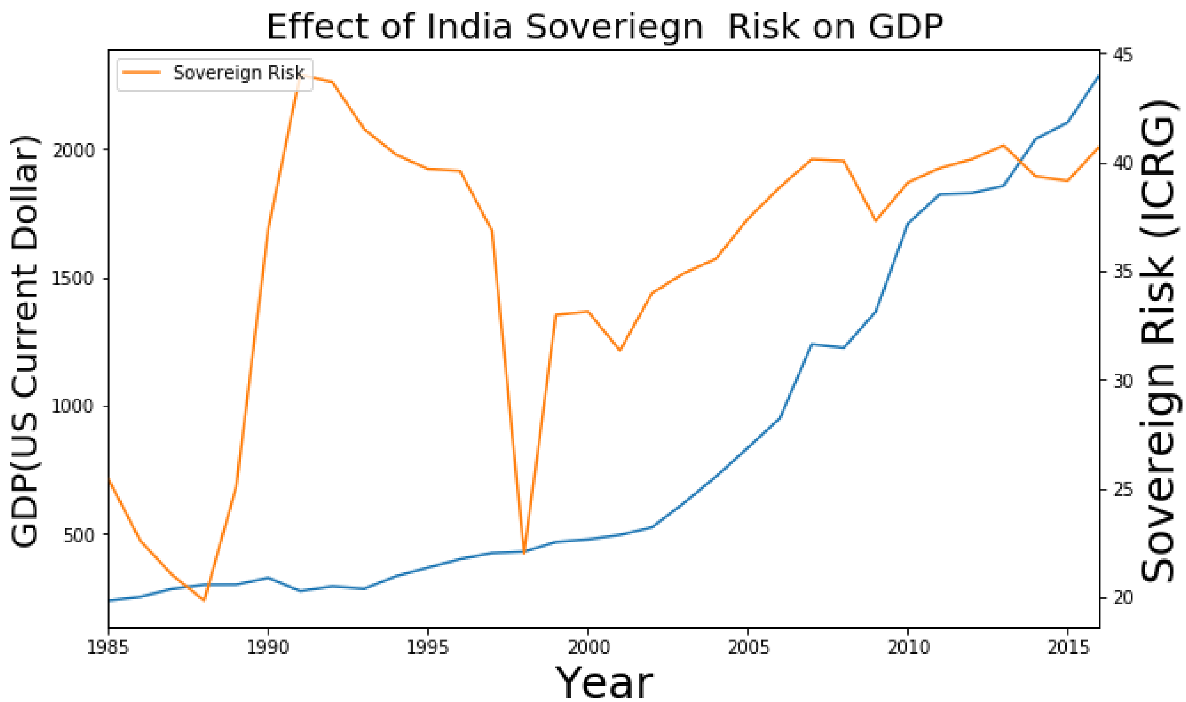
	CountryRisk	PoliticalRisk	EconomicRisk	SovereignRisk	GDP	Commodity Prices	Imports	I
1985	39.085	47.83	30.34	25.42	237.618	96.2167	65.15	
1986	42.415	51.58	33.25	22.58	252.751	96.6750	52.63	
1987	40.600	49.08	32.12	21.00	283.750	102.2080	52.03	
1988	37.895	46.17	29.62	19.83	299.645	131.1170	48.17	
1989	36.665	42.58	30.75	25.08	300.187	132.3250	50.35	
1990	34.455	39.08	29.83	36.92	326.608	124.0250	51.41	
1991	31.040	34.75	27.33	44.00	274.842	117.0420	46.24	
1992	37.435	47.75	27.12	43.67	293.262	113.8580	44.91	
1993	42.730	55.50	29.96	41.50	284.194	108.2580	43.18	
1994	49.185	63.75	34.62	40.33	333.014	126.5080	44.81	
1995	49.935	64.08	35.79	39.67	366.600	137.5750	45.69	
1996	49.605	62.92	36.29	39.58	399.791	134.3500	46.85	
1997	50.250	65.50	35.00	36.83	423.189	131.6670	49.94	
1998	46.810	60.58	33.04	22.00	428.767	114.4000	43.44	
1999	44.035	56.75	31.32	32.96	466.841	98.4833	44.42	
2000	44.250	54.92	33.58	33.12	476.636	99.9917	51.50	
2001	44.935	56.33	33.54	31.33	493.934	96.3750	48.27	
2002	45.210	55.92	34.50	33.96	523.768	97.3167	47.40	
2003	46.480	58.46	34.50	34.88	618.369	104.8580	51.21	
2004	49.420	63.17	35.67	35.54	721.589	125.7830	55.66	
2005	49.605	63.71	35.50	37.38	834.218	140.3920	64.68	
2006	49.540	63.54	35.54	38.83	949.118	182.8250	71.98	
2007	48.980	62.46	35.50	40.12	1238.700	206.5250	75.15	
2008	46.705	60.83	32.58	40.04	1224.100	256.0330	93.23	
2009	48.335	63.17	33.50	37.29	1365.370	212.7420	79.42	
2010	47.480	62.04	32.92	39.04	1708.460	256.0420	95.98	
2011	45.855	58.33	33.38	39.71	1823.050	302.0000	109.79	
2012	45.310	57.62	33.00	40.12	1827.640	276.7830	100.00	
2013	45.685	58.83	32.54	40.75	1856.720	258.1830	103.85	
2014	46.750	60.46	33.04	39.33	2039.130	242.5080	98.82	
2015	48.395	61.62	35.17	39.12	2103.590	201.5750	74.29	
2016	49.190	62.88	35.50	40.71	2289.750	200.0830	71.92	

In [703]:
```
ax=IndGDPPriceRisk['Commodity Prices'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('Commodity Prices (Current US Dollar)',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=IndGDPPriceRisk['SovereignRisk'].plot(secondary_y=True,label='Soveriegn Ri
sk')
ax2.set_ylabel('Sovereign Risk (ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of India Soveriegn  Risk on Commodity Prices',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```




```
<Figure size 720x720 with 0 Axes>
```

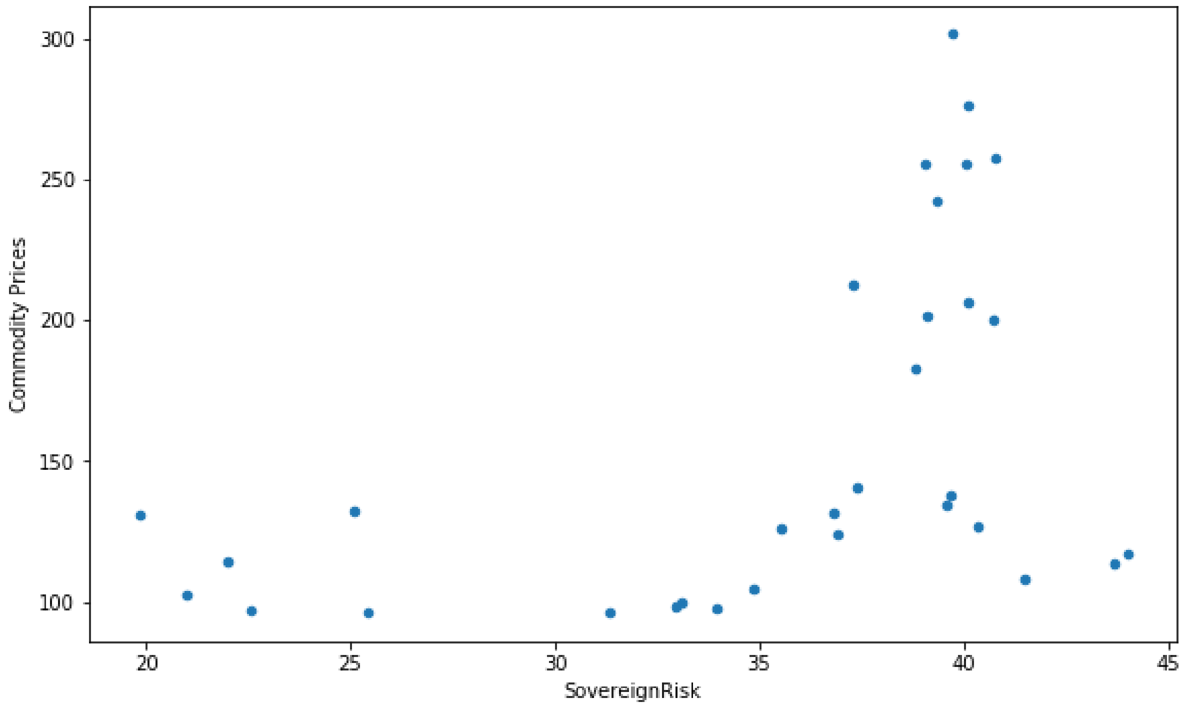
In [704]:
```
ax=IndGDPPriceRisk['GDP'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('GDP(US Current Dollar)',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=IndGDPPriceRisk['SovereignRisk'].plot(secondary_y=True,label='Sovereign Ri
sk')
ax2.set_ylabel('Sovereign Risk (ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of India Soveriegn  Risk on GDP',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```




```
<Figure size 720x720 with 0 Axes>
```

In [705]:
```
IndGDPPriceRisk.plot.scatter(x='SovereignRisk',y='Commodity Prices',figsize=(1
0,6))
IndGDPPriceRisk['SovereignRisk'].corr(IndGDPPriceRisk['Commodity Prices'])
```

Out[705]: 0.46613794096425415



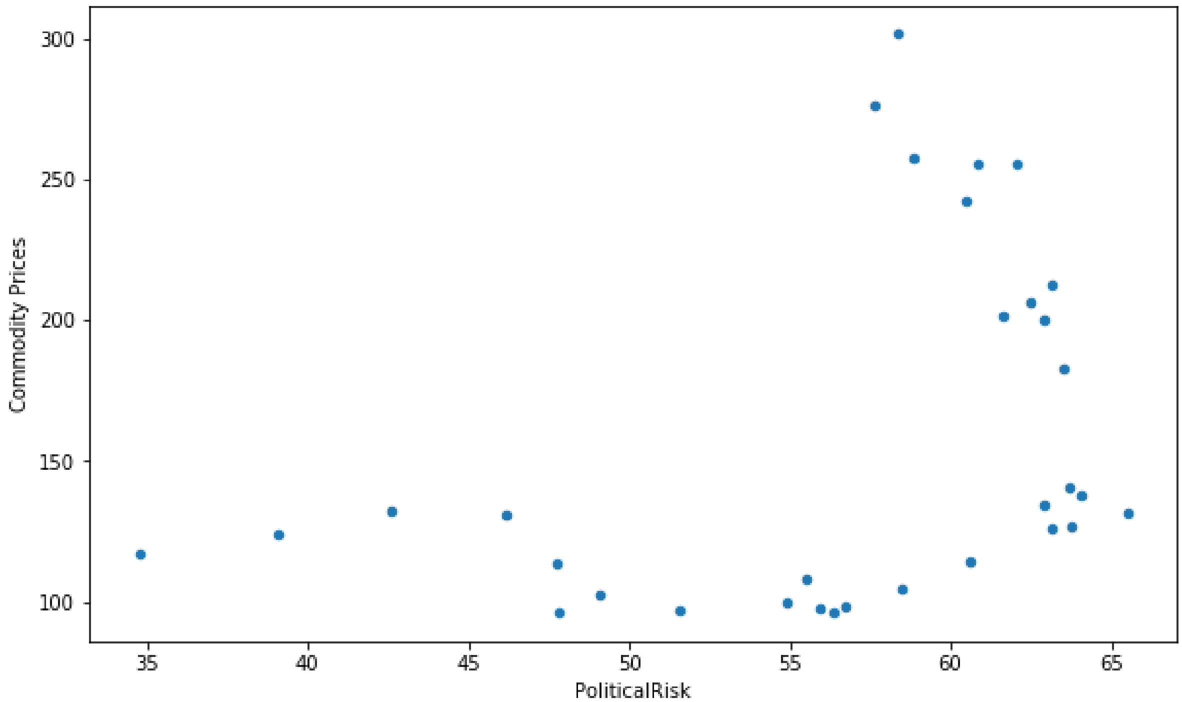
In [706]:
```
x=IndGDPPriceRisk[['SovereignRisk']]
y=IndGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:      Commodity Prices   R-squared:                       0.21
7
Model:                           OLS   Adj. R-squared:                  0.19
1
Method:                Least Squares   F-statistic:                     8.32
8
Date:               Sat, 12 Oct 2019   Prob (F-statistic):            0.0071
7
Time:                       22:41:40   Log-Likelihood:                -173.5
9
No. Observations:                 32   AIC:                             351.
2
Df Residuals:                     30   BIC:                             354.
1
Df Model:                          1
Covariance Type:           nonrobust
==============================================================================
====
                   coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const            6.2786     53.189      0.118      0.907    -102.348     11
4.905
SovereignRisk    4.2589      1.476      2.886      0.007       1.245
7.273
==============================================================================
=
Omnibus:                       2.613   Durbin-Watson:                   0.26
2
Prob(Omnibus):                 0.271   Jarque-Bera (JB):                2.20
8
Skew:                          0.531   Prob(JB):                        0.33
1
Kurtosis:                      2.273   Cond. No.                         19
1.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

```
C:\Users\ishaa\Anaconda3NEW1\lib\site-packages\numpy\core\fromnumeric.py:238
9: FutureWarning: Method .ptp is deprecated and will be removed in a future v
ersion. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

In [707]:
```
IndGDPPriceRisk.plot.scatter(x='PoliticalRisk',y='Commodity Prices',figsize=(10,6))
IndGDPPriceRisk['PoliticalRisk'].corr(IndGDPPriceRisk['Commodity Prices'])
```

Out[707]: 0.3750372018894608

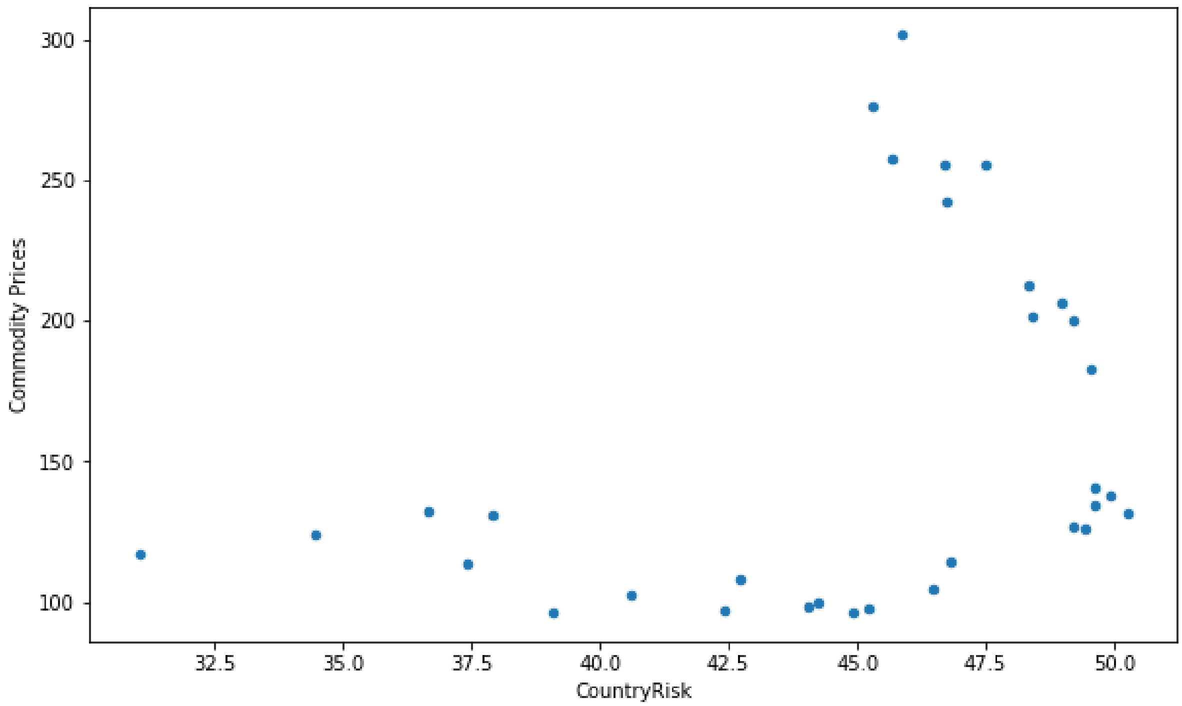


In [708]:
```
x=IndGDPPriceRisk[['PoliticalRisk']]
y=IndGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
=============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.14
1
Model:                            OLS   Adj. R-squared:                  0.11
2
Method:                 Least Squares   F-statistic:                     4.91
0
Date:                Sat, 12 Oct 2019   Prob (F-statistic):             0.034
4
Time:                        22:41:41   Log-Likelihood:                -175.0
9
No. Observations:                  32   AIC:                             354.
2
Df Residuals:                      30   BIC:                             357.
1
Df Model:                           1
Covariance Type:            nonrobust
=============================================================================
====
                   coef    std err          t      P>|t|      [0.025      0.
975]
-----------------------------------------------------------------------------
----
const          -12.6404     77.284     -0.164      0.871    -170.474     14
5.194
PoliticalRisk    2.9959      1.352      2.216      0.034       0.235
5.757
=============================================================================
=
Omnibus:                        4.178   Durbin-Watson:                   0.20
8
Prob(Omnibus):                  0.124   Jarque-Bera (JB):                3.83
3
Skew:                           0.815   Prob(JB):                        0.14
7
Kurtosis:                       2.535   Cond. No.                         42
1.
=============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [709]:
```
IndGDPPriceRisk.plot.scatter(x='CountryRisk',y='Commodity Prices',figsize=(10,
6))
IndGDPPriceRisk['CountryRisk'].corr(IndGDPPriceRisk['Commodity Prices'])
```

Out[709]: 0.34499524223578715

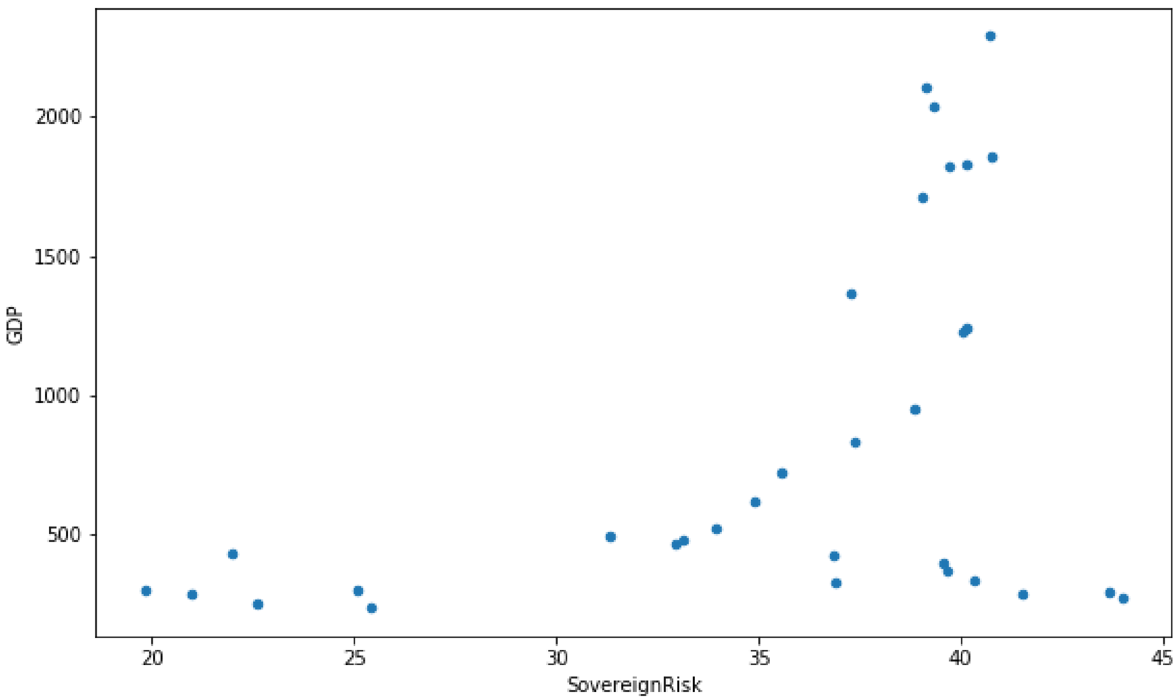


In [710]:
```
x=IndGDPPriceRisk[['CountryRisk']]
y=IndGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.11
9
Model:                            OLS   Adj. R-squared:                  0.09
0
Method:                 Least Squares   F-statistic:                     4.05
3
Date:                Sat, 12 Oct 2019   Prob (F-statistic):             0.053
1
Time:                        22:41:41   Log-Likelihood:                -175.4
9
No. Observations:                  32   AIC:                             355.
0
Df Residuals:                      30   BIC:                             357.
9
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==
                  coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
const         -37.7834     97.346     -0.388      0.701    -236.591     161.0
24
CountryRisk     4.3463      2.159      2.013      0.053      -0.063       8.7
55
==============================================================================
=
Omnibus:                        4.425   Durbin-Watson:                   0.20
0
Prob(Omnibus):                  0.109   Jarque-Bera (JB):                4.07
5
Skew:                           0.845   Prob(JB):                        0.13
0
Kurtosis:                       2.556   Cond. No.                         41
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [711]:
```
IndGDPPriceRisk.plot.scatter(x='SovereignRisk',y='GDP',figsize=(10,6))
IndGDPPriceRisk['SovereignRisk'].corr(IndGDPPriceRisk['GDP'])
```

Out[711]: 0.4469772245049541

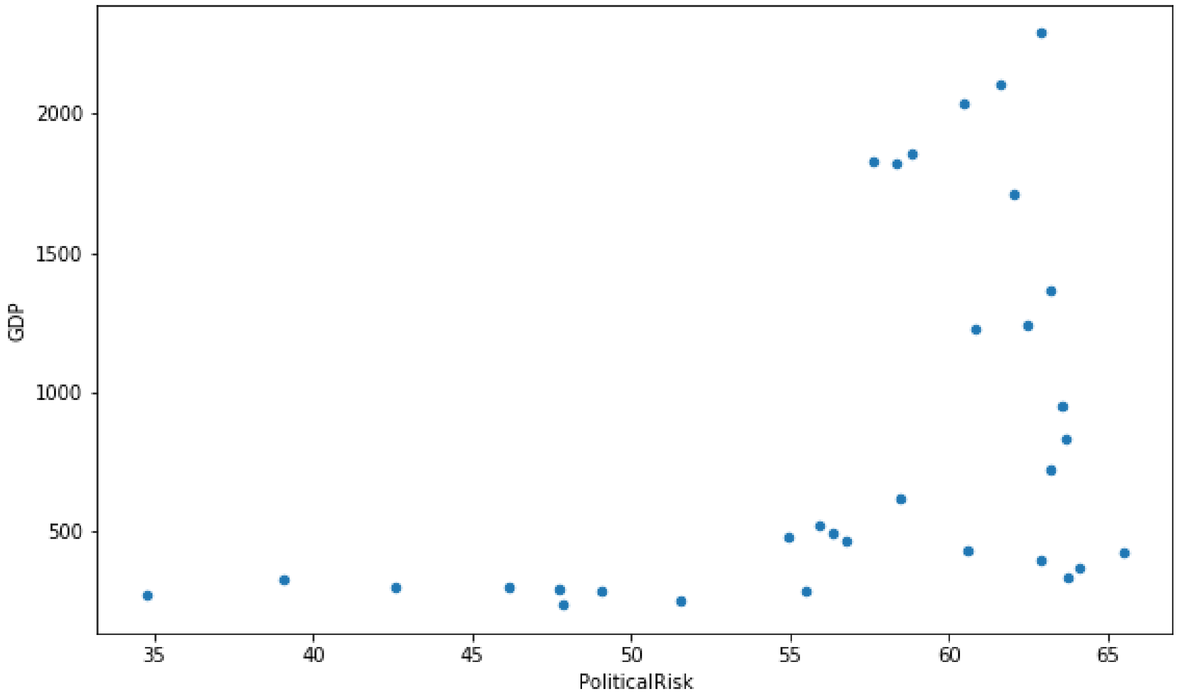


In [712]:
```
x=IndGDPPriceRisk[['SovereignRisk']]
y=IndGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.20
0
Model:                            OLS   Adj. R-squared:                  0.17
3
Method:                 Least Squares   F-statistic:                     7.49
0
Date:                Sat, 12 Oct 2019   Prob (F-statistic):             0.010
3
Time:                        22:41:42   Log-Likelihood:                -249.4
6
No. Observations:                  32   AIC:                             502.
9
Df Residuals:                      30   BIC:                             505.
8
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
====
                    coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const          -684.6764    569.425     -1.202      0.239   -1847.596     47
8.244
SovereignRisk    43.2396     15.799      2.737      0.010      10.973      7
5.506
==============================================================================
=
Omnibus:                        1.624   Durbin-Watson:                   0.12
7
Prob(Omnibus):                  0.444   Jarque-Bera (JB):                1.45
8
Skew:                           0.399   Prob(JB):                        0.48
2
Kurtosis:                       2.325   Cond. No.                         19
1.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [713]:
```
IndGDPPriceRisk.plot.scatter(x='PoliticalRisk',y='GDP',figsize=(10,6))
IndGDPPriceRisk['PoliticalRisk'].corr(IndGDPPriceRisk['GDP'])
```

Out[713]: 0.45701662953741795

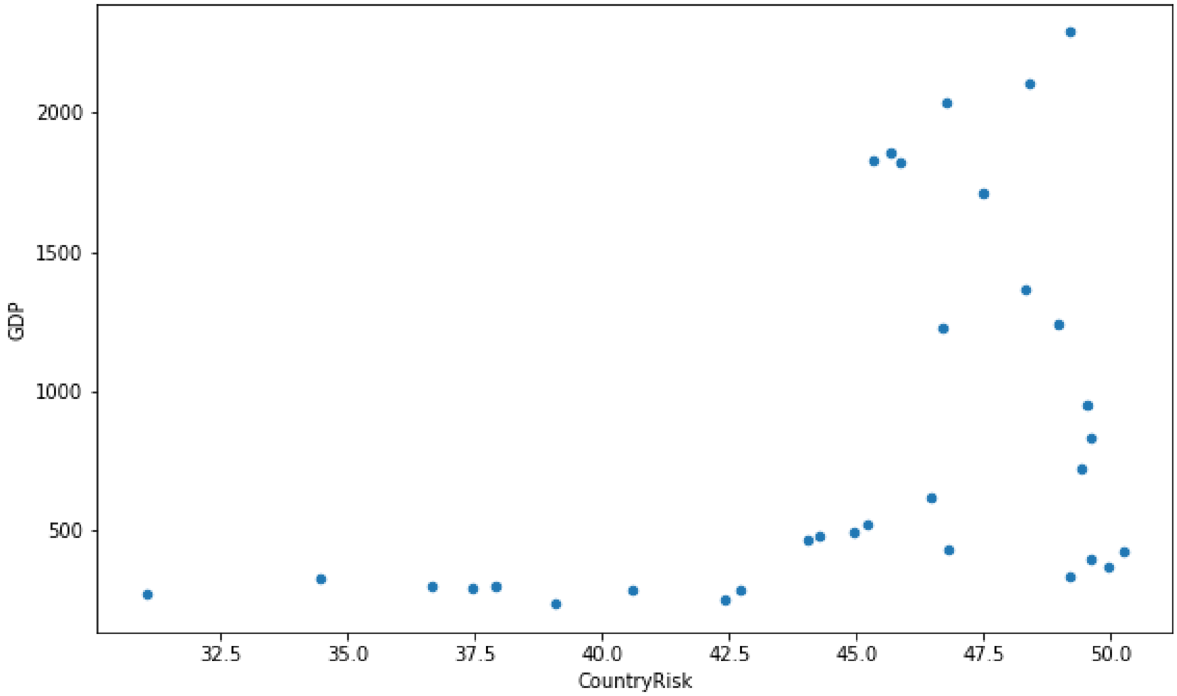


In [714]:
```
x=IndGDPPriceRisk[['PoliticalRisk']]
y=IndGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.20
9
Model:                            OLS   Adj. R-squared:                  0.18
2
Method:                 Least Squares   F-statistic:                     7.92
0
Date:                Sat, 12 Oct 2019   Prob (F-statistic):            0.0085
5
Time:                        22:41:42   Log-Likelihood:                -249.2
7
No. Observations:                  32   AIC:                             502.
5
Df Residuals:                      30   BIC:                             505.
5
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
====
                    coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const         -1343.2683    785.127     -1.711      0.097   -2946.712      26
0.176
PoliticalRisk    38.6548     13.735      2.814      0.009      10.604       6
6.706
==============================================================================
=
Omnibus:                        3.673   Durbin-Watson:                   0.09
2
Prob(Omnibus):                  0.159   Jarque-Bera (JB):                3.19
2
Skew:                           0.689   Prob(JB):                        0.20
3
Kurtosis:                       2.298   Cond. No.                         42
1.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [715]:
```
IndGDPPriceRisk.plot.scatter(x='CountryRisk',y='GDP',figsize=(10,6))
IndGDPPriceRisk['CountryRisk'].corr(IndGDPPriceRisk['GDP'])
```

Out[715]: 0.4423876671089368



In [716]:
```
x=IndGDPPriceRisk[['CountryRisk']]
y=IndGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.19
6
Model:                            OLS   Adj. R-squared:                  0.16
9
Method:                 Least Squares   F-statistic:                     7.30
0
Date:                Sat, 12 Oct 2019   Prob (F-statistic):             0.011
2
Time:                        22:41:43   Log-Likelihood:                -249.5
4
No. Observations:                  32   AIC:                             503.
1
Df Residuals:                      30   BIC:                             506.
0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==
                  coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
const       -1799.0820    984.819     -1.827      0.078   -3810.350     212.1
86
CountryRisk    59.0099     21.841      2.702      0.011      14.405     103.6
15
==============================================================================
=
Omnibus:                        3.900   Durbin-Watson:                   0.08
6
Prob(Omnibus):                  0.142   Jarque-Bera (JB):                3.35
0
Skew:                           0.704   Prob(JB):                        0.18
7
Kurtosis:                       2.270   Cond. No.                         41
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [717]:
```
x=IndGDPPriceRisk[['SovereignRisk','PoliticalRisk','CountryRisk','Commodity Prices','Imports','Exports']]
y=IndGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.81
4
Model:                            OLS   Adj. R-squared:                  0.76
9
Method:                 Least Squares   F-statistic:                     18.2
2
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           5.00e-0
8
Time:                        22:41:43   Log-Likelihood:                -226.1
2
No. Observations:                  32   AIC:                             466.
2
Df Residuals:                      25   BIC:                             476.
5
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
=======
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const            -2.962e+04   5.47e+04     -0.541      0.593   -1.42e+05
8.31e+04
SovereignRisk        6.8050     11.096      0.613      0.545     -16.047
29.657
PoliticalRisk      -33.8506     64.649     -0.524      0.605    -166.997
99.296
CountryRisk         72.2978    101.697      0.711      0.484    -137.151
281.746
Commodity Prices     4.6188      3.580      1.290      0.209      -2.755
11.993
Imports              1.9449     27.080      0.072      0.943     -53.828
57.718
Exports            284.2573    574.762      0.495      0.625    -899.488    1
468.002
==============================================================================
=
Omnibus:                       20.850   Durbin-Watson:                   0.44
9
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               31.29
2
Skew:                           1.617   Prob(JB):                     1.60e-0
7
Kurtosis:                       6.606   Cond. No.                     2.11e+0
5
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 2.11e+05. This might indicate that there a
```

```
re
strong multicollinearity or other numerical problems.
```

Argentina Analysis

In [718]:

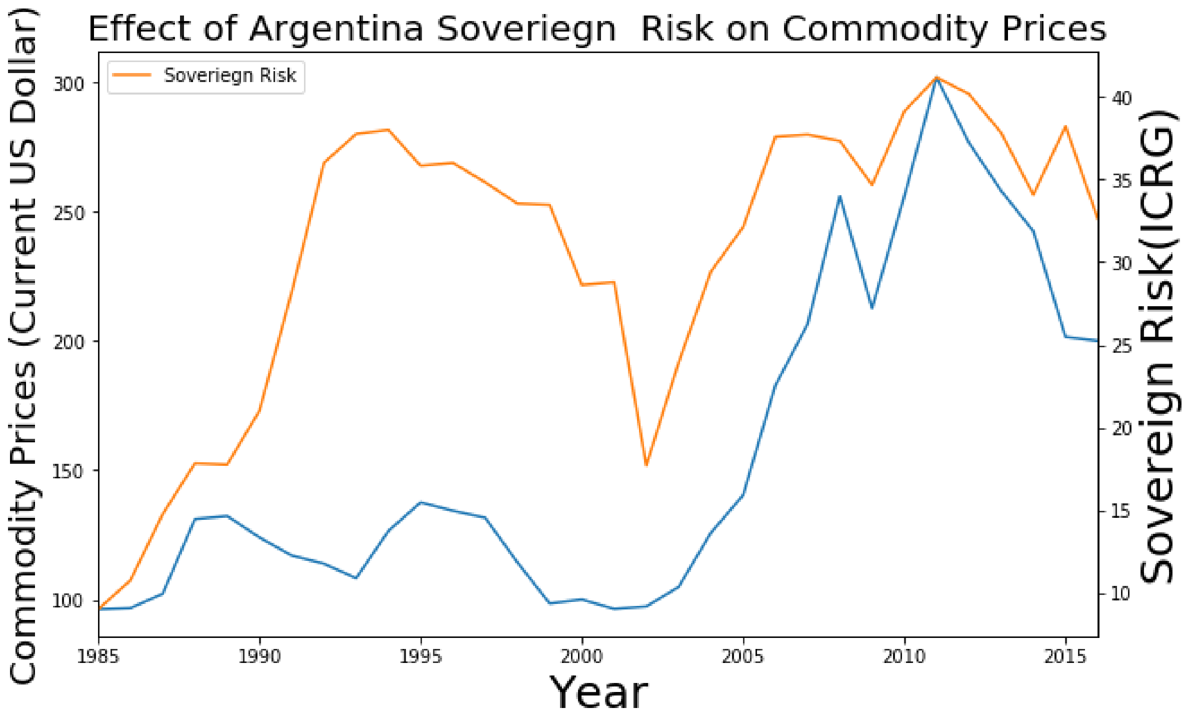
```
GDP=[95.593,114.949,117.854,138.044,88.567,153.205,205.515,247.987,256.365,27
9.15,280.08,295.12,317.549,324.242,307.673,308.491,291.738,108.731,138.151,16
4.922,199.273,232.892,287.921,363.545,334.633,424.728,527.644,579.666,611.471,
563.614,642.464,556.774]
year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]
SovereignRisk=[ 9.0, 10.75, 14.75, 17.83, 17.75, 21.0, 28.17, 36.0, 37.75, 38.
0, 35.83, 36.0, 34.83, 33.54, 33.46, 28.62, 28.79, 17.71, 23.92, 29.42, 32.12,
37.58, 37.71, 37.33, 34.67, 39.12, 41.17, 40.17, 37.83, 34.08, 38.21, 32.62]
prices=[ 96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858
, 108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375,
97.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.04
2, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]
EconomicRisk=[ 14.58, 16.  , 15.83, 17.79, 14.71, 22.17, 22.38, 26.42,
        25.83, 31.83, 31.67, 35.04, 39.21, 40.71, 37.7 , 38.5 , 35.88,
        27.  , 33.25, 40.46, 39.08, 40.62, 41.  , 38.83, 33.25, 37.  ,
        40.12, 36.62, 35.33, 31.17, 29.92, 28.33]

PoliticalRisk=[ 54.67, 56.17, 57.75, 56.42, 58.67, 61.17, 65.  , 67.67,
        70.67, 75.42, 74.25, 75.42, 73.83, 75.83, 74.17, 72.58, 72.71,
        59.17, 60.79, 64.92, 68.96, 71.  , 70.67, 67.  , 64.88, 64.  ,
        66.46, 65.21, 61.92, 62.08, 64.17, 66.79]
CountryRisk=[  34.625, 36.085, 36.79,  37.105, 36.69,  41.67,  43.69,  47.045,
48.25,
 53.625, 52.96,  55.23,  56.52,  58.27,  55.935, 55.54,  54.295, 43.085, 47.02
,
 52.69,  54.02,  55.81,  55.835, 52.915, 49.065, 50.5,   53.29,  50.915, 48.62
5,
 46.625, 47.045, 47.56 ]
Exports=[ 97.72,  96.43,  96.3 ,  96.86,  96.68,  96.43,  95.76,  95.77,
         95.91,  95.98,  95.94,  96.48,  96.4 ,  95.33,  94.92,  95.68,
         95.67,  96.13,  96.66,  97.18,  96.98,  97.37,  98.41,  99.74,
         98.2 ,  98.86, 100.03, 100.  , 100.07,  99.62,  98.05,  98.  ]
Imports=[ 70.9 ,  58.26,  56.98,  62.77,  60.97,  58.15,  52.15,  52.24,
         53.48,  54.22,  53.81,  58.59,  57.65,  48.62,  45.12,  50.59,
         50.78,  54.99,  59.55,  64.22,  61.97,  65.75,  77.56,  95.54,
         75.18,  83.65, 100.94, 100.  , 101.21,  94.42,  74.32,  73.94]
ArgGDPPriceRisk=pd.DataFrame({'EconomicRisk':EconomicRisk,'SovereignRisk':Sove
reignRisk,'GDP':GDP,'Commodity Prices':prices,'PoliticalRisk':PoliticalRisk,'C
ountryRisk':CountryRisk,'Imports':Imports,'Exports':Exports},index=year)
```

In [719]:
```
ArgGDPPriceRisk
```

Out[719]:

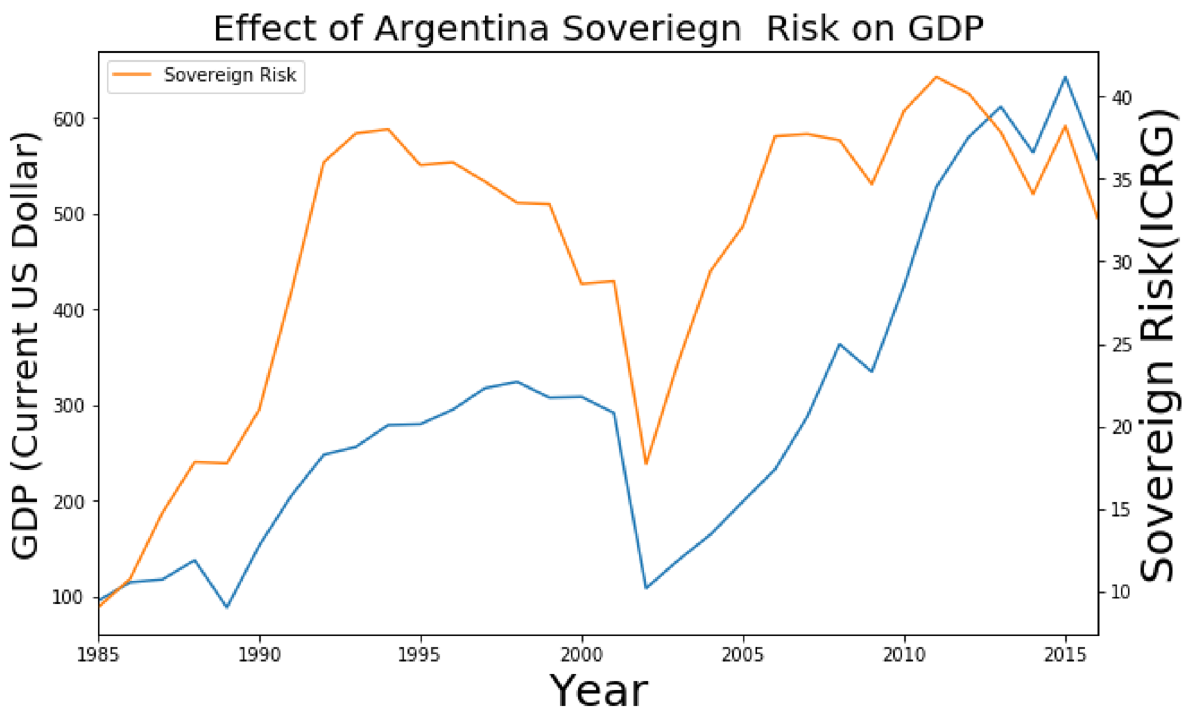
	EconomicRisk	SovereignRisk	GDP	Commodity Prices	PoliticalRisk	CountryRisk	Imports	E
1985	14.58	9.00	95.593	96.2167	54.67	34.625	70.90	
1986	16.00	10.75	114.949	96.6750	56.17	36.085	58.26	
1987	15.83	14.75	117.854	102.2080	57.75	36.790	56.98	
1988	17.79	17.83	138.044	131.1170	56.42	37.105	62.77	
1989	14.71	17.75	88.567	132.3250	58.67	36.690	60.97	
1990	22.17	21.00	153.205	124.0250	61.17	41.670	58.15	
1991	22.38	28.17	205.515	117.0420	65.00	43.690	52.15	
1992	26.42	36.00	247.987	113.8580	67.67	47.045	52.24	
1993	25.83	37.75	256.365	108.2580	70.67	48.250	53.48	
1994	31.83	38.00	279.150	126.5080	75.42	53.625	54.22	
1995	31.67	35.83	280.080	137.5750	74.25	52.960	53.81	
1996	35.04	36.00	295.120	134.3500	75.42	55.230	58.59	
1997	39.21	34.83	317.549	131.6670	73.83	56.520	57.65	
1998	40.71	33.54	324.242	114.4000	75.83	58.270	48.62	
1999	37.70	33.46	307.673	98.4833	74.17	55.935	45.12	
2000	38.50	28.62	308.491	99.9917	72.58	55.540	50.59	
2001	35.88	28.79	291.738	96.3750	72.71	54.295	50.78	
2002	27.00	17.71	108.731	97.3167	59.17	43.085	54.99	
2003	33.25	23.92	138.151	104.8580	60.79	47.020	59.55	
2004	40.46	29.42	164.922	125.7830	64.92	52.690	64.22	
2005	39.08	32.12	199.273	140.3920	68.96	54.020	61.97	
2006	40.62	37.58	232.892	182.8250	71.00	55.810	65.75	
2007	41.00	37.71	287.921	206.5250	70.67	55.835	77.56	
2008	38.83	37.33	363.545	256.0330	67.00	52.915	95.54	
2009	33.25	34.67	334.633	212.7420	64.88	49.065	75.18	
2010	37.00	39.12	424.728	256.0420	64.00	50.500	83.65	
2011	40.12	41.17	527.644	302.0000	66.46	53.290	100.94	
2012	36.62	40.17	579.666	276.7830	65.21	50.915	100.00	
2013	35.33	37.83	611.471	258.1830	61.92	48.625	101.21	
2014	31.17	34.08	563.614	242.5080	62.08	46.625	94.42	
2015	29.92	38.21	642.464	201.5750	64.17	47.045	74.32	
2016	28.33	32.62	556.774	200.0830	66.79	47.560	73.94	

In [720]:
```
ax=ArgGDPPriceRisk['Commodity Prices'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('Commodity Prices (Current US Dollar)',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=ArgGDPPriceRisk['SovereignRisk'].plot(secondary_y=True,label='Soveriegn Ri
sk')
ax2.set_ylabel('Sovereign Risk(ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of Argentina Soveriegn  Risk on Commodity Prices',fontsize=2
0)

plt.figure(figsize=(10,10))
plt.show()
```




```
<Figure size 720x720 with 0 Axes>
```

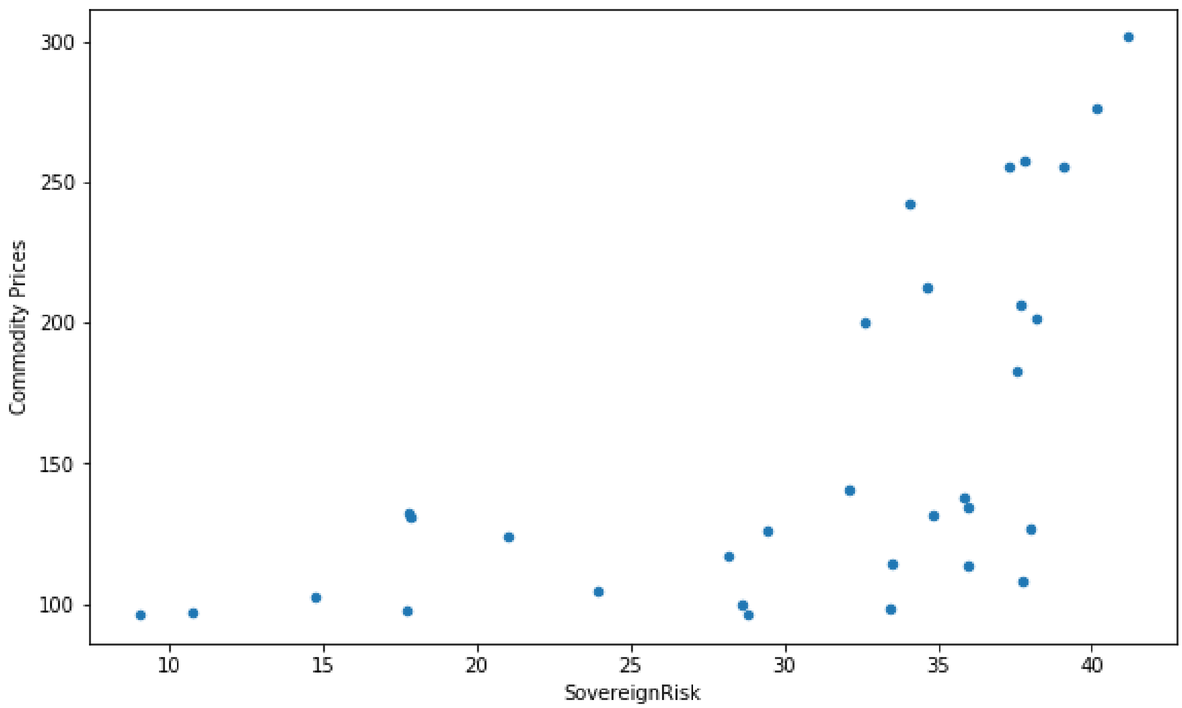
In [721]:
```
ax=ArgGDPPriceRisk['GDP'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('GDP (Current US Dollar)',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=ArgGDPPriceRisk['SovereignRisk'].plot(secondary_y=True,label='Sovereign Ri
sk')
ax2.set_ylabel('Sovereign Risk(ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of Argentina Soveriegn  Risk on GDP',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```




```
<Figure size 720x720 with 0 Axes>
```

In [722]:
```
ArgGDPPriceRisk.plot.scatter(x='SovereignRisk',y='Commodity Prices',figsize=(10,6))
ArgGDPPriceRisk['SovereignRisk'].corr(ArgGDPPriceRisk['Commodity Prices'])
```

Out[722]: 0.6072186505589854



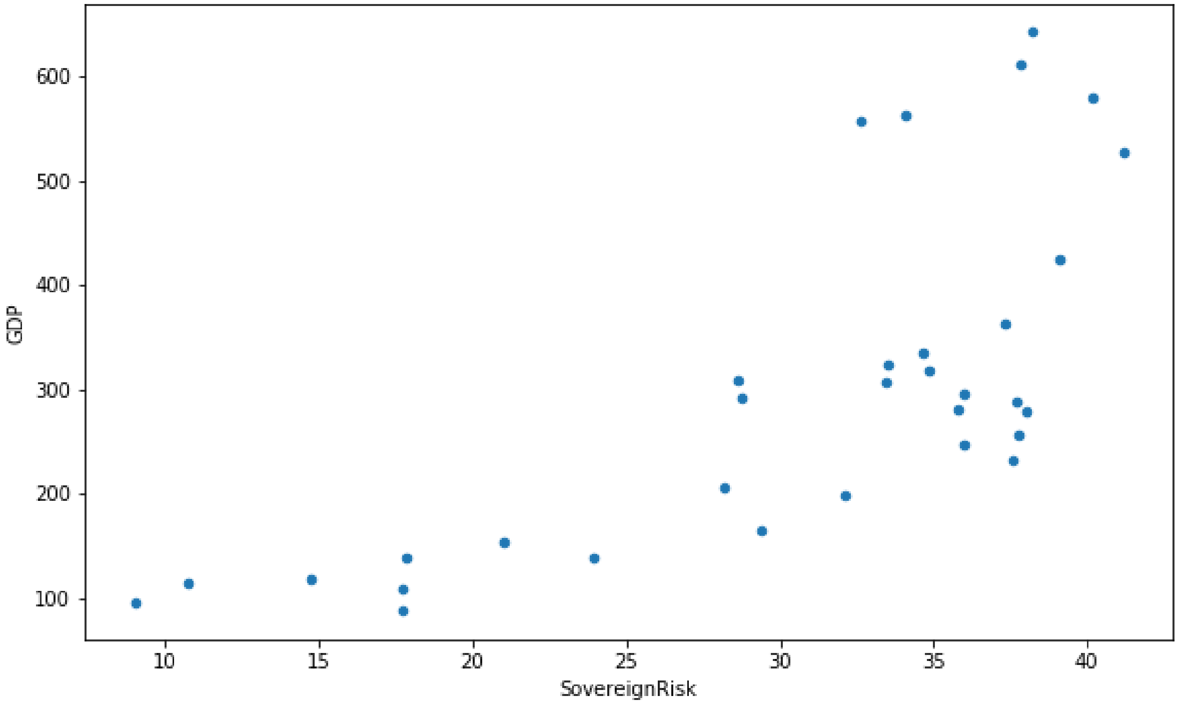
In [723]:
```
x=ArgGDPPriceRisk[['SovereignRisk']]
y=ArgGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                             OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.36
9
Model:                            OLS   Adj. R-squared:                  0.34
8
Method:                 Least Squares   F-statistic:                     17.5
2
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           0.00022
9
Time:                        22:41:45   Log-Likelihood:                -170.1
5
No. Observations:                  32   AIC:                             344.
3
Df Residuals:                      30   BIC:                             347.
2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
====
                   coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const           28.2504     32.054      0.881      0.385     -37.212      9
3.713
SovereignRisk    4.2232      1.009      4.186      0.000       2.163
6.284
==============================================================================
=
Omnibus:                        3.653   Durbin-Watson:                   0.24
4
Prob(Omnibus):                  0.161   Jarque-Bera (JB):                1.70
3
Skew:                           0.207   Prob(JB):                        0.42
7
Kurtosis:                       1.949   Cond. No.                         11
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

```
C:\Users\ishaa\Anaconda3NEW1\lib\site-packages\numpy\core\fromnumeric.py:238
9: FutureWarning: Method .ptp is deprecated and will be removed in a future v
ersion. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

In [724]:
```
ArgGDPPriceRisk.plot.scatter(x='SovereignRisk',y='GDP',figsize=(10,6))
ArgGDPPriceRisk['SovereignRisk'].corr(ArgGDPPriceRisk['GDP'])
```

Out[724]: 0.7178656439312139

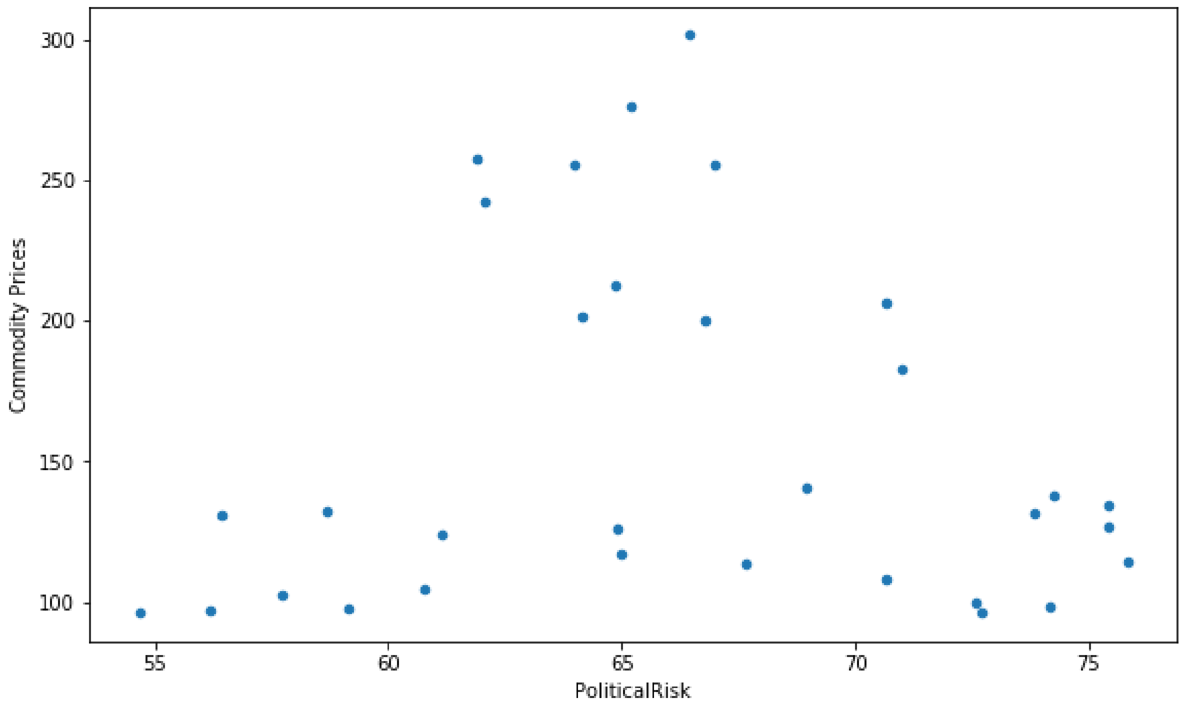


In [725]:
```
x=ArgGDPPriceRisk[['SovereignRisk']]
y=ArgGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.51
5
Model:                            OLS   Adj. R-squared:                  0.49
9
Method:                 Least Squares   F-statistic:                     31.9
0
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           3.74e-0
6
Time:                        22:41:45   Log-Likelihood:                -196.1
1
No. Observations:                  32   AIC:                             396.
2
Df Residuals:                      30   BIC:                             399.
2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
====
                    coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const           -92.3257     72.141     -1.280      0.210    -239.656      5
5.005
SovereignRisk    12.8242      2.271      5.648      0.000       8.187      1
7.461
==============================================================================
=
Omnibus:                        4.448   Durbin-Watson:                   0.13
0
Prob(Omnibus):                  0.108   Jarque-Bera (JB):                3.86
2
Skew:                           0.847   Prob(JB):                        0.14
5
Kurtosis:                       2.831   Cond. No.                         11
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [726]:
```
ArgGDPPriceRisk.plot.scatter(x='PoliticalRisk',y='Commodity Prices',figsize=(10,6))
ArgGDPPriceRisk['PoliticalRisk'].corr(ArgGDPPriceRisk['Commodity Prices'])
```

Out[726]: -0.03564897653142644

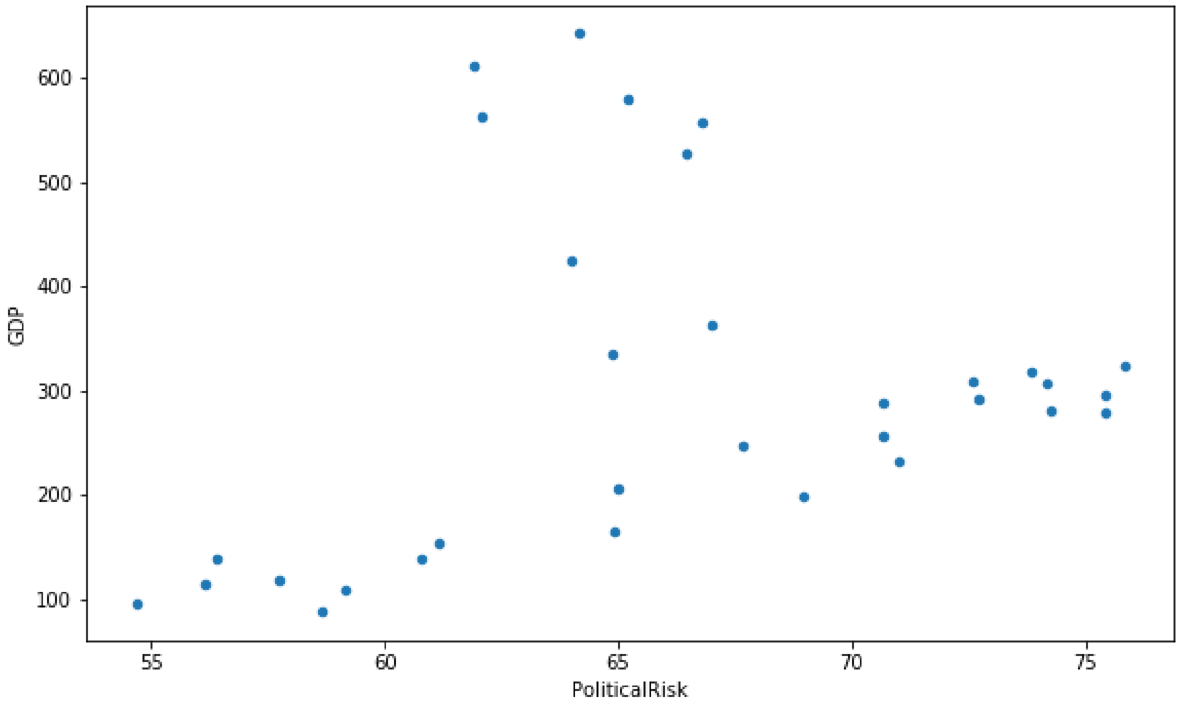


In [727]:
```
x=ArgGDPPriceRisk[['PoliticalRisk']]
y=ArgGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.00
1
Model:                            OLS   Adj. R-squared:                 -0.03
2
Method:                 Least Squares   F-statistic:                   0.0381
7
Date:                Sat, 12 Oct 2019   Prob (F-statistic):              0.84
6
Time:                        22:41:46   Log-Likelihood:                -177.4
9
No. Observations:                  32   AIC:                             359.
0
Df Residuals:                      30   BIC:                             361.
9
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
====
                    coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const           180.7956    122.201      1.479      0.149     -68.771     43
0.362
PoliticalRisk    -0.3588      1.836     -0.195      0.846      -4.109
3.391
==============================================================================
=
Omnibus:                        4.910   Durbin-Watson:                   0.13
4
Prob(Omnibus):                  0.086   Jarque-Bera (JB):                4.50
7
Skew:                           0.865   Prob(JB):                        0.10
5
Kurtosis:                       2.375   Cond. No.                         71
8.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [728]:
```
ArgGDPPriceRisk.plot.scatter(x='PoliticalRisk',y='GDP',figsize=(10,6))
ArgGDPPriceRisk['PoliticalRisk'].corr(ArgGDPPriceRisk['GDP'])
```

Out[728]: 0.24885116711382088

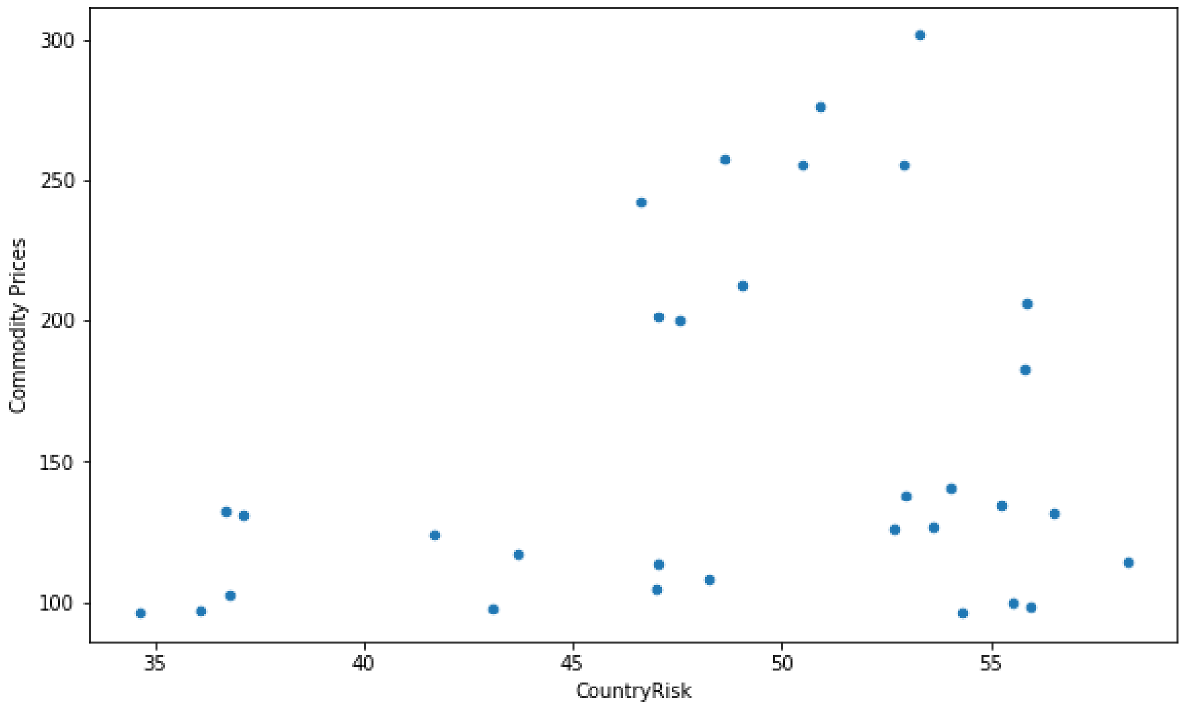


In [729]:
```
x=ArgGDPPriceRisk[['PoliticalRisk']]
y=ArgGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
=============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.06
2
Model:                            OLS   Adj. R-squared:                  0.03
1
Method:                 Least Squares   F-statistic:                     1.98
0
Date:                Sat, 12 Oct 2019   Prob (F-statistic):              0.17
0
Time:                        22:41:46   Log-Likelihood:                -206.6
8
No. Observations:                  32   AIC:                             417.
4
Df Residuals:                      30   BIC:                             420.
3
Df Model:                           1
Covariance Type:            nonrobust
=============================================================================
====
                    coef    std err          t      P>|t|      [0.025      0.
975]
-----------------------------------------------------------------------------
----
const          -127.5473    304.199     -0.419      0.678    -748.805     49
3.710
PoliticalRisk     6.4327      4.571      1.407      0.170      -2.903      1
5.768
=============================================================================
=
Omnibus:                        7.950   Durbin-Watson:                   0.09
8
Prob(Omnibus):                  0.019   Jarque-Bera (JB):                7.61
8
Skew:                           1.195   Prob(JB):                       0.022
2
Kurtosis:                       2.978   Cond. No.                         71
8.
=============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [730]:
```
ArgGDPPriceRisk.plot.scatter(x='CountryRisk',y='Commodity Prices',figsize=(10,
6))
ArgGDPPriceRisk['CountryRisk'].corr(ArgGDPPriceRisk['Commodity Prices'])
```

Out[730]: 0.24138762175549172

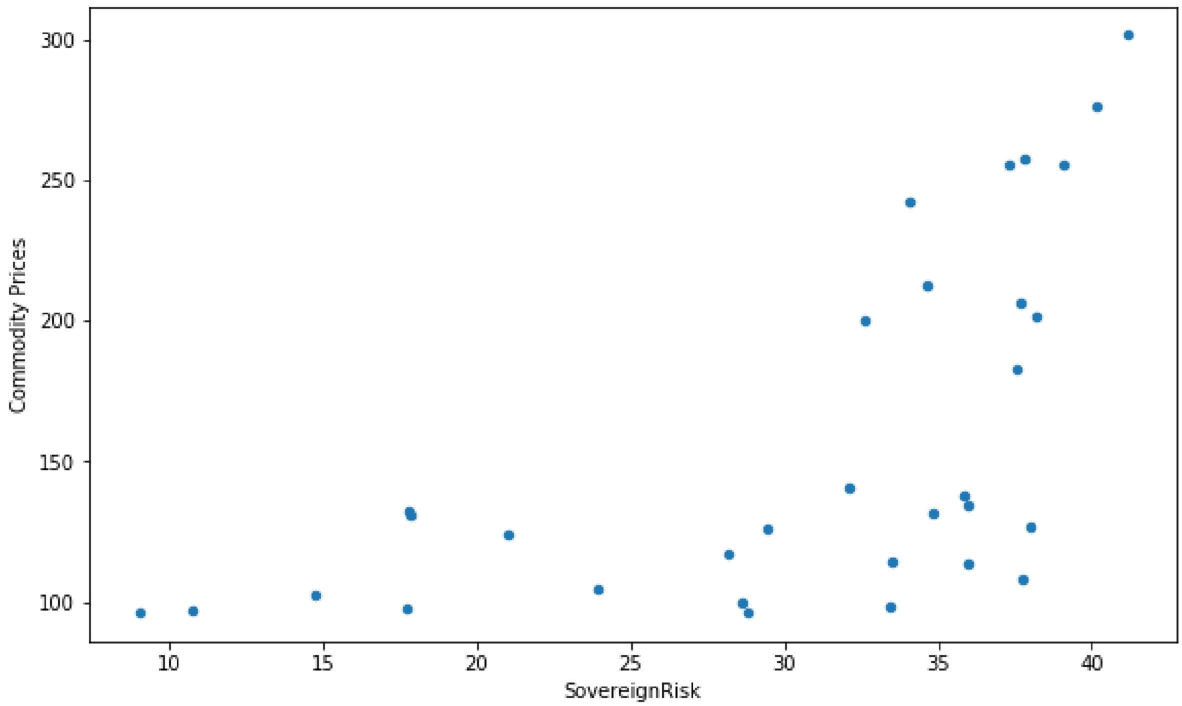


In [731]:
```
x=ArgGDPPriceRisk[['CountryRisk']]
y=ArgGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.05
8
Model:                            OLS   Adj. R-squared:                  0.02
7
Method:                 Least Squares   F-statistic:                     1.85
6
Date:                Sat, 12 Oct 2019   Prob (F-statistic):              0.18
3
Time:                        22:41:47   Log-Likelihood:                -176.5
5
No. Observations:                  32   AIC:                             357.
1
Df Residuals:                      30   BIC:                             360.
0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==
                  coef    std err          t      P>|t|      [0.025      0.97
5]
------------------------------------------------------------------------------
--
const          48.8865     80.129      0.610      0.546    -114.758     212.5
31
CountryRisk     2.2191      1.629      1.362      0.183      -1.107       5.5
46
==============================================================================
=
Omnibus:                        4.366   Durbin-Watson:                   0.13
2
Prob(Omnibus):                  0.113   Jarque-Bera (JB):                3.89
5
Skew:                           0.785   Prob(JB):                        0.14
3
Kurtosis:                       2.325   Cond. No.                         35
9.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [732]:
```
ArgGDPPriceRisk.plot.scatter(x='SovereignRisk',y='Commodity Prices',figsize=(1
0,6))
ArgGDPPriceRisk['SovereignRisk'].corr(ArgGDPPriceRisk['Commodity Prices'])
```

Out[732]: 0.6072186505589854



In [733]:
```
x=ArgGDPPriceRisk[['SovereignRisk']]
y=ArgGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:      Commodity Prices   R-squared:                       0.36
9
Model:                           OLS   Adj. R-squared:                  0.34
8
Method:                Least Squares   F-statistic:                     17.5
2
Date:               Sat, 12 Oct 2019   Prob (F-statistic):           0.00022
9
Time:                       22:41:47   Log-Likelihood:                -170.1
5
No. Observations:                 32   AIC:                             344.
3
Df Residuals:                     30   BIC:                             347.
2
Df Model:                          1
Covariance Type:           nonrobust
==============================================================================
====
                    coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const            28.2504     32.054      0.881      0.385     -37.212       9
3.713
SovereignRisk     4.2232      1.009      4.186      0.000       2.163
6.284
==============================================================================
=
Omnibus:                       3.653   Durbin-Watson:                   0.24
4
Prob(Omnibus):                 0.161   Jarque-Bera (JB):                1.70
3
Skew:                          0.207   Prob(JB):                        0.42
7
Kurtosis:                      1.949   Cond. No.                          11
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [734]:
```
x=ArgGDPPriceRisk[['SovereignRisk','PoliticalRisk','CountryRisk','Commodity Pr
ices','Imports','Exports']]
y=ArgGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.74
0
Model:                            OLS   Adj. R-squared:                  0.67
7
Method:                 Least Squares   F-statistic:                     11.8
3
Date:                Sat, 12 Oct 2019   Prob (F-statistic):          2.81e-0
6
Time:                        22:41:47   Log-Likelihood:                -186.1
8
No. Observations:                  32   AIC:                             386.
4
Df Residuals:                      25   BIC:                             396.
6
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
=======
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const             1.358e+04   1.05e+04      1.294      0.207   -8031.759
3.52e+04
SovereignRisk        8.7687      5.515      1.590      0.124      -2.589
20.126
PoliticalRisk        4.5365     10.173      0.446      0.659     -16.415
25.489
CountryRisk         -5.1077      7.466     -0.684      0.500     -20.484
10.269
Commodity Prices     0.5183      1.209      0.429      0.672      -1.972
3.008
Imports             17.2411     10.256      1.681      0.105      -3.881
38.364
Exports           -152.5607    113.372     -1.346      0.190    -386.055
80.933
==============================================================================
=
Omnibus:                       13.176   Durbin-Watson:                   0.42
7
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               13.24
8
Skew:                           1.272   Prob(JB):                      0.0013
3
Kurtosis:                       4.860   Cond. No.                     1.43e+0
5
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 1.43e+05. This might indicate that there a
```

```
re
strong multicollinearity or other numerical problems.
```

Looking at India, Country Risk is the biggest risk factor affecting GDP and commodity prices. While Sovereign Risk is still a big factor that impacts GDP and commodity prices, the results shows that it is important to consider Country Risk alongside Sovereign Risk when looking at various political, economic, and financial variables affecting commodity prices and economic output.

As for Argentina, it’s low Sovereign Risk rating shows how even with strong Political and Economic risk numbers, Sovereign Risk is relatively the biggest risk impacting fluctuations in commodity prices and EM GDP output.

It is thus important to consider that just because one risk indicator is significantly stronger than another risk indicator or two does not mean to rely solely on risk factor to accurately make forecasts for GDP and Commodity Prices.

China

In [735]:
```
EconomicRisk=[36.58, 34.12, 35.54, 36.71, 33.58, 34.62, 38.29, 41.08, 35.92,
        30.79, 34.42, 38.71, 40.04, 38.92, 40.33, 39.33, 38.42, 38.58,
        38.58, 38.58, 39.46, 40.83, 41.71, 41.5 , 40.79, 40.33, 39.67,
        39.79, 39.83, 39.79, 40.75, 41.  ]

PoliticalRisk=[67.83, 64.58, 61.33, 63.42, 59.83, 56.92, 57.92, 67.75,
        71.25, 66.83, 68.33, 69.25, 68.83, 66.25, 62.42, 62.42, 62.25,
        66.29, 69.5 , 70.17, 69.25, 68.33, 69.12, 68.08, 67., 64.46,
        60.92, 61.12, 61.29, 57.04, 56.38, 55]

year=[1985,1986,1987,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,19
99,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,
2015,2016]

GDP=[312.616, 303.34, 330.303, 411.923, 461.066, 398.623, 415.604,
        495.671, 623.054, 566.471, 736.87, 867.224, 965.338, 1032.57,
        1097.14, 1214.92, 1344.08, 1477.5, 1671.07, 1966.24, 2308.8,
        2774.29, 3571.45, 4604.29, 5121.68, 6066.35, 7522.1, 8570.35,
     9635.03, 10534.53, 11226.19, 11221.84]

prices=[96.2167, 96.675, 102.208, 131.117, 132.325, 124.025, 117.042, 113.858,
108.258, 126.508, 137.575, 134.35, 131.667, 114.4, 98.4833, 99.9917, 96.375, 9
7.3167, 104.858, 125.783, 140.392, 182.825, 206.525, 256.033, 212.742, 256.042
, 302.0, 276.783, 258.183, 242.508, 201.575, 200.083]
SovereignRisk=[35.25, 32.92, 30.08, 29.92, 27.  , 24.58, 24.83, 32.17, 40.75,
        39.  , 39.  , 38.08, 39.83, 44.17, 45.29, 44.42, 45.29, 45.  ,
        45.21, 44.67, 46.08, 47.33, 47.92, 47.92, 47.79, 48.  , 47.96,
        47.5 , 47.5 , 47.46, 47.71, 46.46]

Exports=[ 99.48,  99.09,  99.14,  99.06,  99.09,  99.16,  98.95,  98.93,
         98.82,  98.85,  98.91,  98.93,  98.94,  98.69,  98.79,  99.05,
         98.98,  98.97,  99.08,  99.31,  99.46,  99.67,  99.79, 100.04,
         99.64,  99.94, 100.17, 100.  , 100.07,  99.99,  99.56,  99.58]

Imports=[ 72.24,  56.34,  57.58,  56.67,  57.65,  58.3 ,  50.71,  51.4 ,
         47.18,  48.74,  51.65,  52.48,  51.2 ,  43.18,  44.71,  52.43,
         50.95,  52.07,  55.03,  62.94,  68.32,  80.87,  89.56, 105.21,
         78.26,  94.35, 111.25, 100.  , 104.41,  97.76,  73.63,  73.45]

ChinaGDPPriceRisk=pd.DataFrame({'Economic Risk':EconomicRisk,'Political Risk':
PoliticalRisk,'Sovereign Risk':SovereignRisk,'GDP':GDP,'Commodity Prices':pric
es,'Exports':Exports,'Imports':Imports},index=year)
ChinaGDPPriceRisk
```

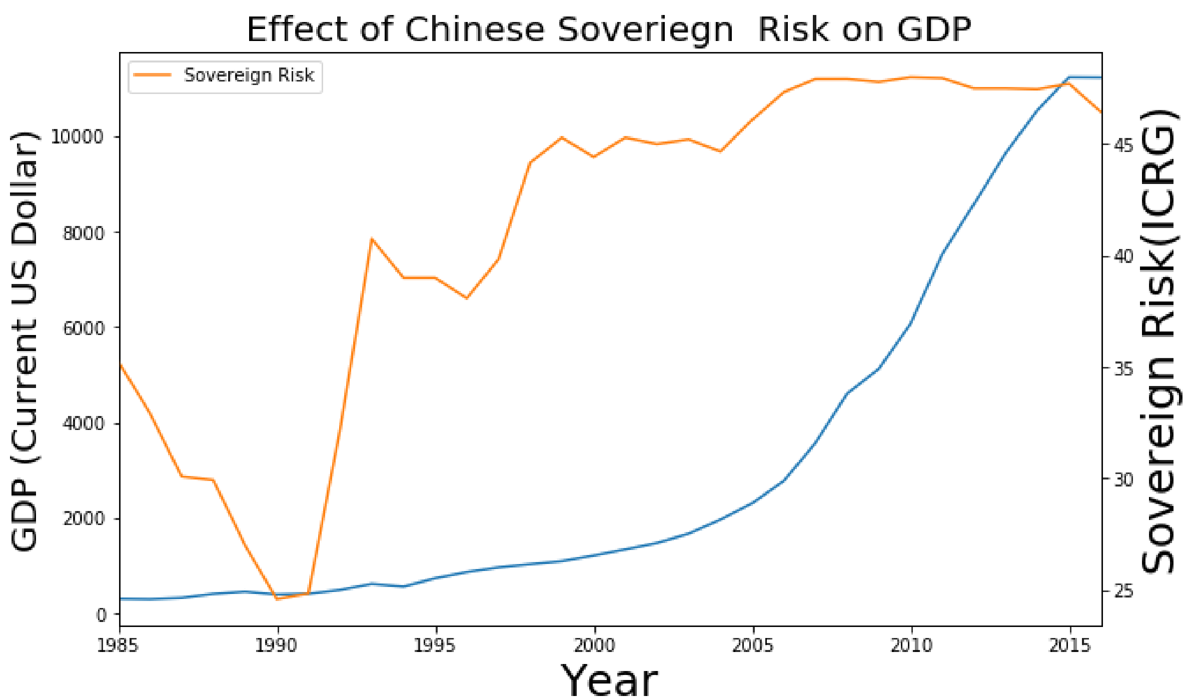
Out[735]:

	Economic Risk	Political Risk	Sovereign Risk	GDP	Commodity Prices	Exports	Imports
1985	36.58	67.83	35.25	312.616	96.2167	99.48	72.24
1986	34.12	64.58	32.92	303.340	96.6750	99.09	56.34
1987	35.54	61.33	30.08	330.303	102.2080	99.14	57.58
1988	36.71	63.42	29.92	411.923	131.1170	99.06	56.67
1989	33.58	59.83	27.00	461.066	132.3250	99.09	57.65
1990	34.62	56.92	24.58	398.623	124.0250	99.16	58.30
1991	38.29	57.92	24.83	415.604	117.0420	98.95	50.71
1992	41.08	67.75	32.17	495.671	113.8580	98.93	51.40
1993	35.92	71.25	40.75	623.054	108.2580	98.82	47.18
1994	30.79	66.83	39.00	566.471	126.5080	98.85	48.74
1995	34.42	68.33	39.00	736.870	137.5750	98.91	51.65
1996	38.71	69.25	38.08	867.224	134.3500	98.93	52.48
1997	40.04	68.83	39.83	965.338	131.6670	98.94	51.20
1998	38.92	66.25	44.17	1032.570	114.4000	98.69	43.18
1999	40.33	62.42	45.29	1097.140	98.4833	98.79	44.71
2000	39.33	62.42	44.42	1214.920	99.9917	99.05	52.43
2001	38.42	62.25	45.29	1344.080	96.3750	98.98	50.95
2002	38.58	66.29	45.00	1477.500	97.3167	98.97	52.07
2003	38.58	69.50	45.21	1671.070	104.8580	99.08	55.03
2004	38.58	70.17	44.67	1966.240	125.7830	99.31	62.94
2005	39.46	69.25	46.08	2308.800	140.3920	99.46	68.32
2006	40.83	68.33	47.33	2774.290	182.8250	99.67	80.87
2007	41.71	69.12	47.92	3571.450	206.5250	99.79	89.56
2008	41.50	68.08	47.92	4604.290	256.0330	100.04	105.21
2009	40.79	67.00	47.79	5121.680	212.7420	99.64	78.26
2010	40.33	64.46	48.00	6066.350	256.0420	99.94	94.35
2011	39.67	60.92	47.96	7522.100	302.0000	100.17	111.25
2012	39.79	61.12	47.50	8570.350	276.7830	100.00	100.00
2013	39.83	61.29	47.50	9635.030	258.1830	100.07	104.41
2014	39.79	57.04	47.46	10534.530	242.5080	99.99	97.76
2015	40.75	56.38	47.71	11226.190	201.5750	99.56	73.63
2016	41.00	55.00	46.46	11221.840	200.0830	99.58	73.45

In [736]:
```
ax=ChinaGDPPriceRisk['Commodity Prices'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('Commodity Prices (Current US Dollar)',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=ChinaGDPPriceRisk['Economic Risk'].plot(secondary_y=True,label='Soveriegn
 Risk')
ax2.set_ylabel('Sovereign Risk(ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of China Economic Risk on Commodity Prices',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```


```
<Figure size 720x720 with 0 Axes>
```

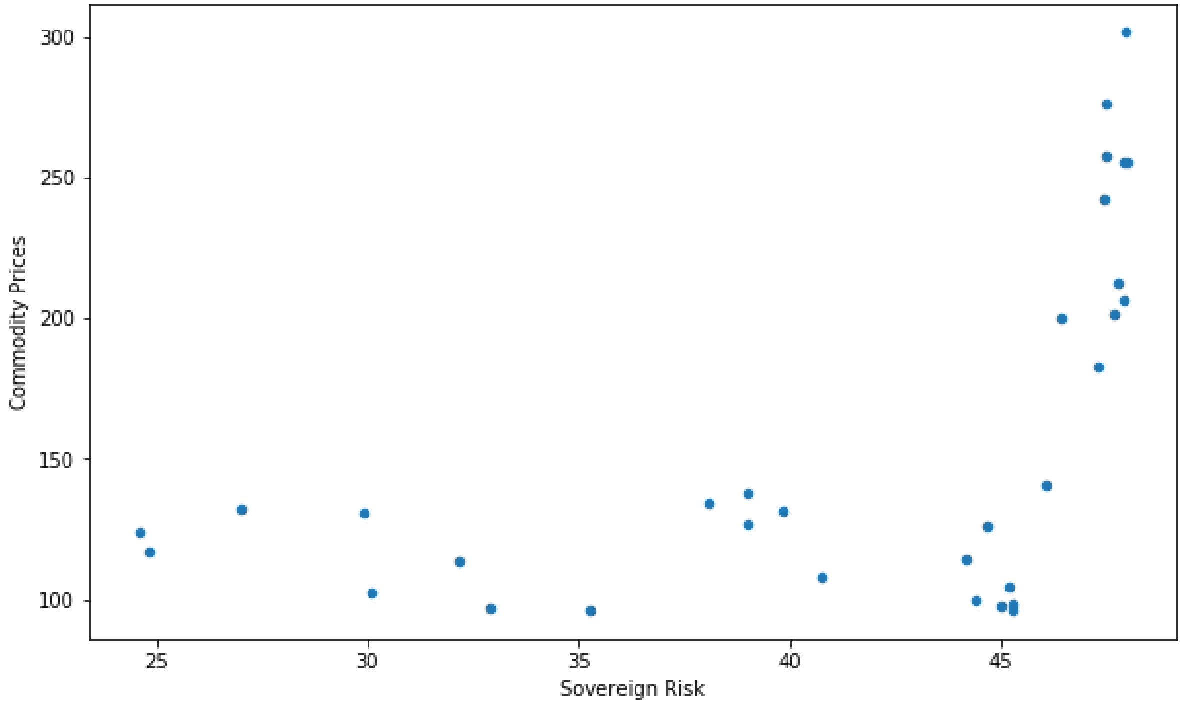
In [737]:
```
ax=ChinaGDPPriceRisk['GDP'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('GDP (Current US Dollar)',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=ChinaGDPPriceRisk['Sovereign Risk'].plot(secondary_y=True,label='Sovereign
Risk')
ax2.set_ylabel('Sovereign Risk(ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of Chinese Soveriegn  Risk on GDP',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```




```
<Figure size 720x720 with 0 Axes>
```

In [738]:
```
ChinaGDPPriceRisk.plot.scatter(x='Sovereign Risk',y='Commodity Prices',figsize
=(10,6))
ChinaGDPPriceRisk['Sovereign Risk'].corr(ChinaGDPPriceRisk['Commodity Prices'
])
```

Out[738]: 0.5503472794650955



In [739]:
```
ax=ChinaGDPPriceRisk['Sovereign Risk'].plot(label='Prices',figsize=(10,6))
ax.set_ylabel('Economic Risk (ICRG)',fontsize=20)
ax.set_xlabel('Year',fontsize=25)


ax2=ChinaGDPPriceRisk['Economic Risk'].plot(secondary_y=True,label='Sovereign
 Risk')
ax2.set_ylabel('Economic Risk(ICRG)',fontsize=25)

plt.legend(loc='upper left')

plt.title('Effect of Chinese Economic  Risk on Sovereign Risk',fontsize=20)

plt.figure(figsize=(10,10))
plt.show()
```


```
<Figure size 720x720 with 0 Axes>
```

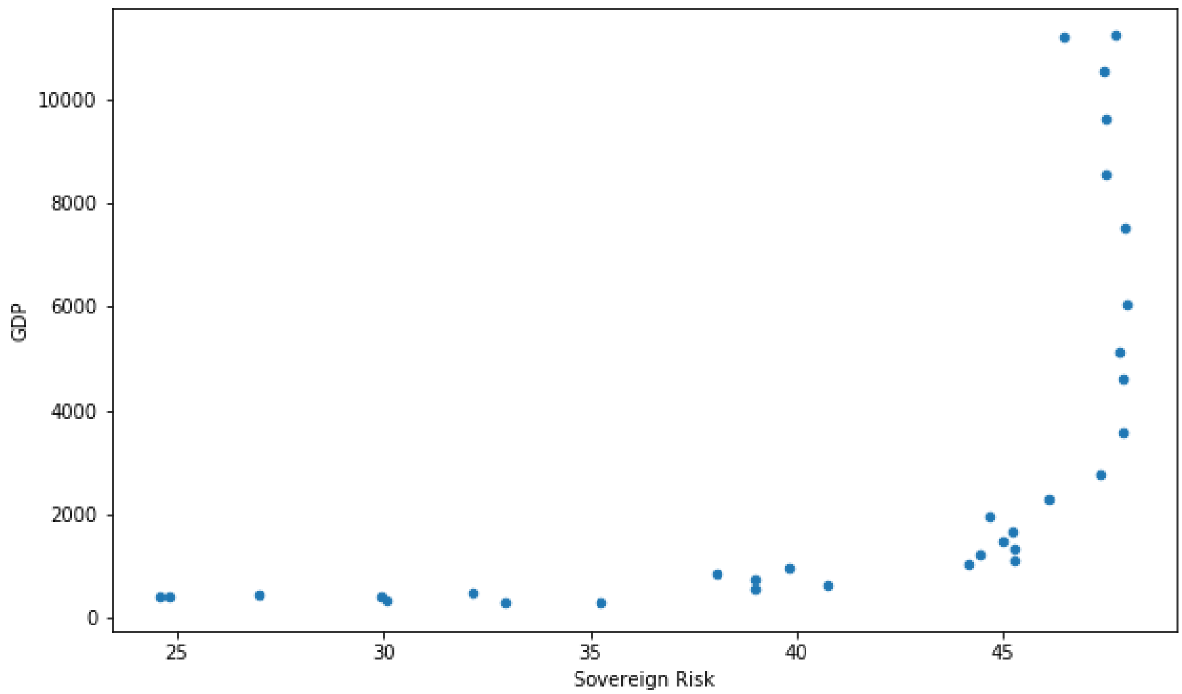
In [740]:
```
x=ChinaGDPPriceRisk[['Sovereign Risk']]
y=ChinaGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.30
3
Model:                            OLS   Adj. R-squared:                  0.28
0
Method:                 Least Squares   F-statistic:                     13.0
3
Date:                Sat, 12 Oct 2019   Prob (F-statistic):            0.0011
0
Time:                        22:41:50   Log-Likelihood:                -171.7
4
No. Observations:                  32   AIC:                             347.
5
Df Residuals:                      30   BIC:                             350.
4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
const            -31.7945     53.149     -0.598      0.554    -140.338      7
6.749
Sovereign Risk     4.5875      1.271      3.610      0.001       1.992
7.183
==============================================================================
=
Omnibus:                        1.326   Durbin-Watson:                   0.24
9
Prob(Omnibus):                  0.515   Jarque-Bera (JB):                1.06
8
Skew:                           0.215   Prob(JB):                        0.58
6
Kurtosis:                       2.215   Cond. No.                         23
5.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

```
C:\Users\ishaa\Anaconda3NEW1\lib\site-packages\numpy\core\fromnumeric.py:238
9: FutureWarning: Method .ptp is deprecated and will be removed in a future v
ersion. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

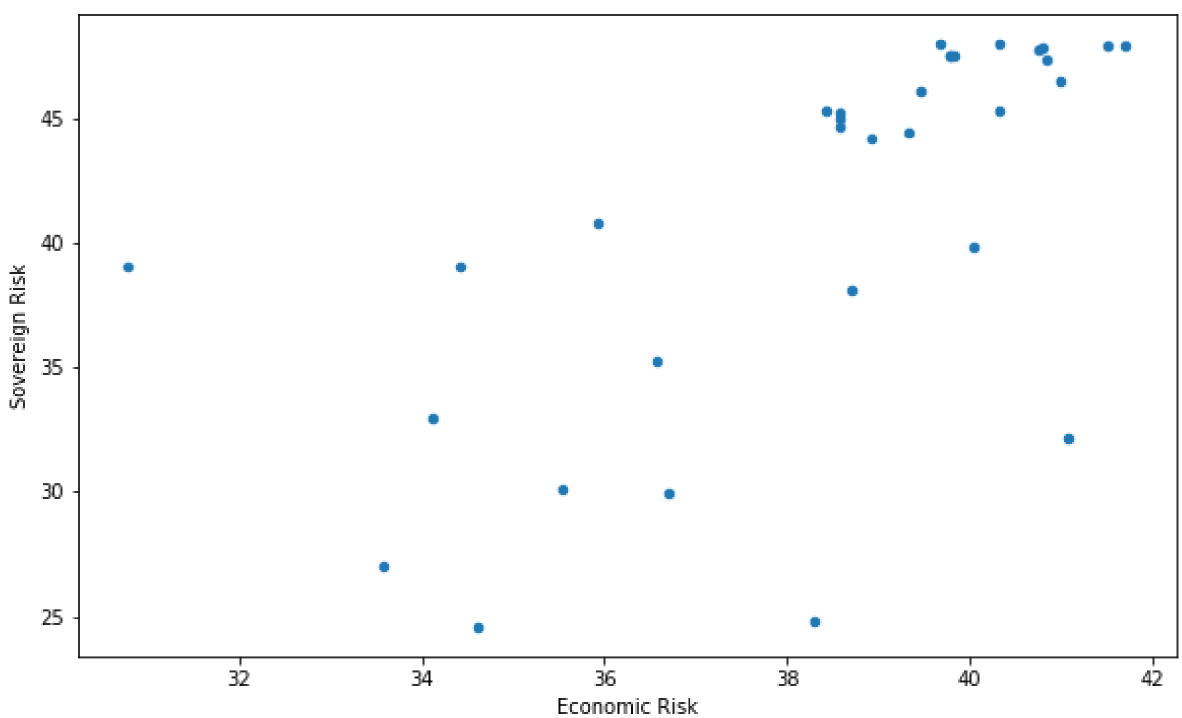
In [741]:
```
ChinaGDPPriceRisk.plot.scatter(x='Sovereign Risk',y='GDP',figsize=(10,6))
ChinaGDPPriceRisk['Sovereign Risk'].corr(ChinaGDPPriceRisk['GDP'])
```

Out[741]: 0.6122787737750908



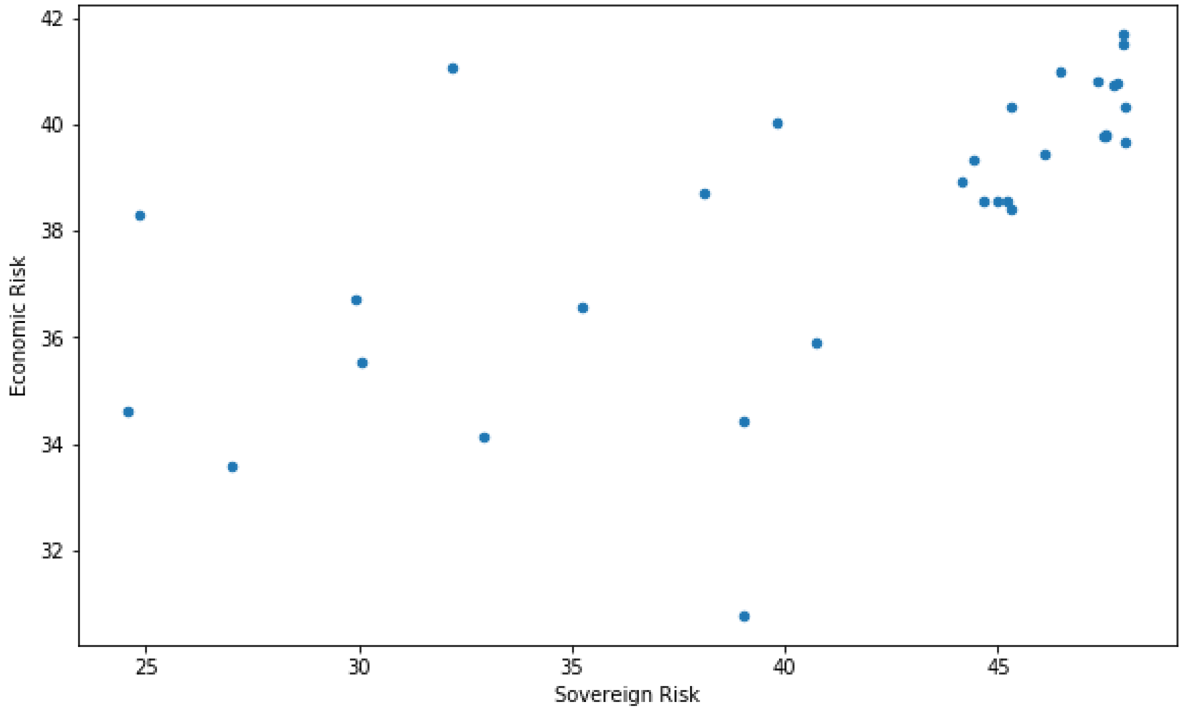
In [742]:
```
ChinaGDPPriceRisk.plot.scatter(x='Economic Risk',y='Sovereign Risk',figsize=(1
0,6))
ChinaGDPPriceRisk['Economic Risk'].corr(ChinaGDPPriceRisk['Sovereign Risk'])
```

Out[742]: 0.6374593698045233



In [743]:
```
ChinaGDPPriceRisk.plot.scatter(x='Sovereign Risk',y='Economic Risk',figsize=(10,6))
ChinaGDPPriceRisk['Sovereign Risk'].corr(ChinaGDPPriceRisk['Economic Risk'])
```

Out[743]: 0.6374593698045232



In [744]:
```
x=ChinaGDPPriceRisk['Economic Risk']
y=ChinaGDPPriceRisk['Sovereign Risk']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:         Sovereign Risk   R-squared:                       0.40
6
Model:                            OLS   Adj. R-squared:                  0.38
7
Method:                 Least Squares   F-statistic:                     20.5
4
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           8.71e-0
5
Time:                        22:41:51   Log-Likelihood:                -101.3
1
No. Observations:                  32   AIC:                             206.
6
Df Residuals:                      30   BIC:                             209.
6
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
====
                    coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const           -28.0581     15.310     -1.833      0.077     -59.326
3.210
Economic Risk     1.8028      0.398      4.532      0.000       0.990
2.615
==============================================================================
=
Omnibus:                        9.307   Durbin-Watson:                   0.63
3
Prob(Omnibus):                  0.010   Jarque-Bera (JB):                7.86
6
Skew:                          -1.087   Prob(JB):                       0.019
6
Kurtosis:                       4.084   Cond. No.                         56
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [745]:
```
x=ChinaGDPPriceRisk[['Sovereign Risk']]
y=ChinaGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

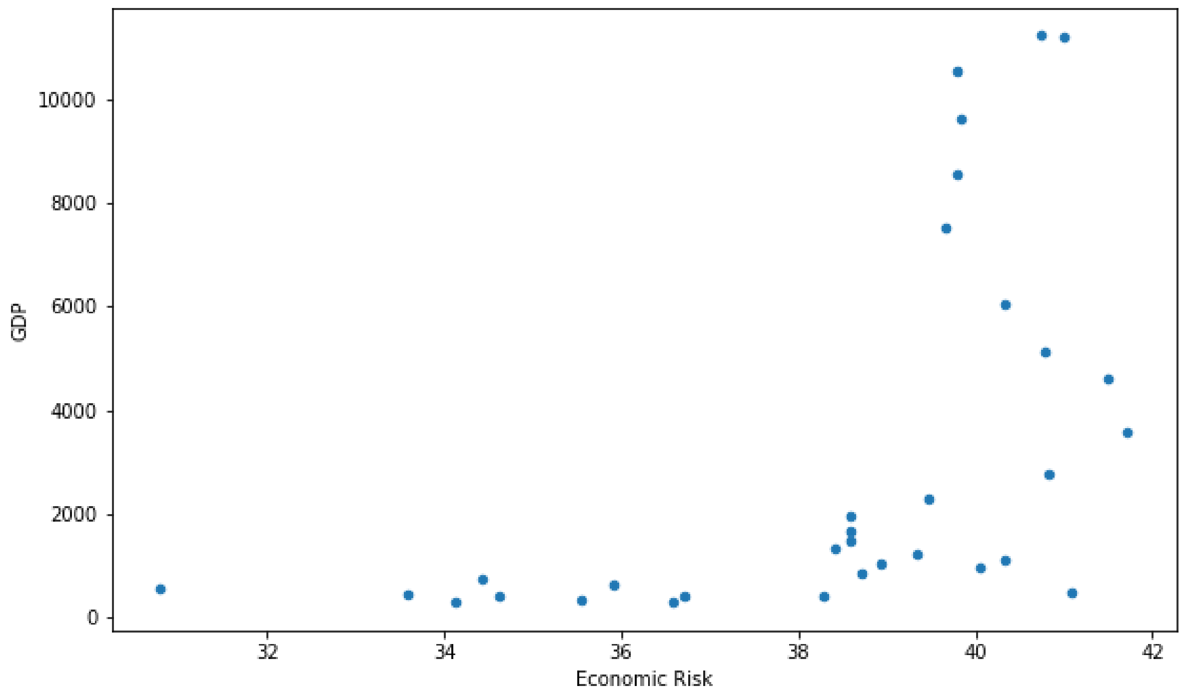
```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.37
5
Model:                            OLS   Adj. R-squared:                  0.35
4
Method:                 Least Squares   F-statistic:                     17.9
9
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           0.00019
6
Time:                        22:41:51   Log-Likelihood:                -299.4
6
No. Observations:                  32   AIC:                             602.
9
Df Residuals:                      30   BIC:                             605.
9
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
-----
const          -8886.8558   2876.759     -3.089      0.004   -1.48e+04   -301
1.731
Sovereign Risk   291.7249     68.777      4.242      0.000     151.263     43
2.186
==============================================================================
=
Omnibus:                        5.214   Durbin-Watson:                   0.09
3
Prob(Omnibus):                  0.074   Jarque-Bera (JB):                4.67
4
Skew:                           0.932   Prob(JB):                       0.096
6
Kurtosis:                       2.819   Cond. No.                         23
5.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [746]:
```
x=ChinaGDPPriceRisk[['Economic Risk']]
y=ChinaGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.27
6
Model:                            OLS   Adj. R-squared:                  0.25
2
Method:                 Least Squares   F-statistic:                     11.4
6
Date:                Sat, 12 Oct 2019   Prob (F-statistic):            0.0020
0
Time:                        22:41:51   Log-Likelihood:                -301.8
0
No. Observations:                  32   AIC:                             607.
6
Df Residuals:                      30   BIC:                             610.
5
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
====
                   coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const         -2.408e+04   8053.679     -2.990      0.006   -4.05e+04    -763
0.859
Economic Risk   708.4266    209.275      3.385      0.002     281.029     113
5.824
==============================================================================
=
Omnibus:                        5.219   Durbin-Watson:                   0.26
3
Prob(Omnibus):                  0.074   Jarque-Bera (JB):                4.71
9
Skew:                           0.935   Prob(JB):                       0.094
5
Kurtosis:                       2.787   Cond. No.                         56
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

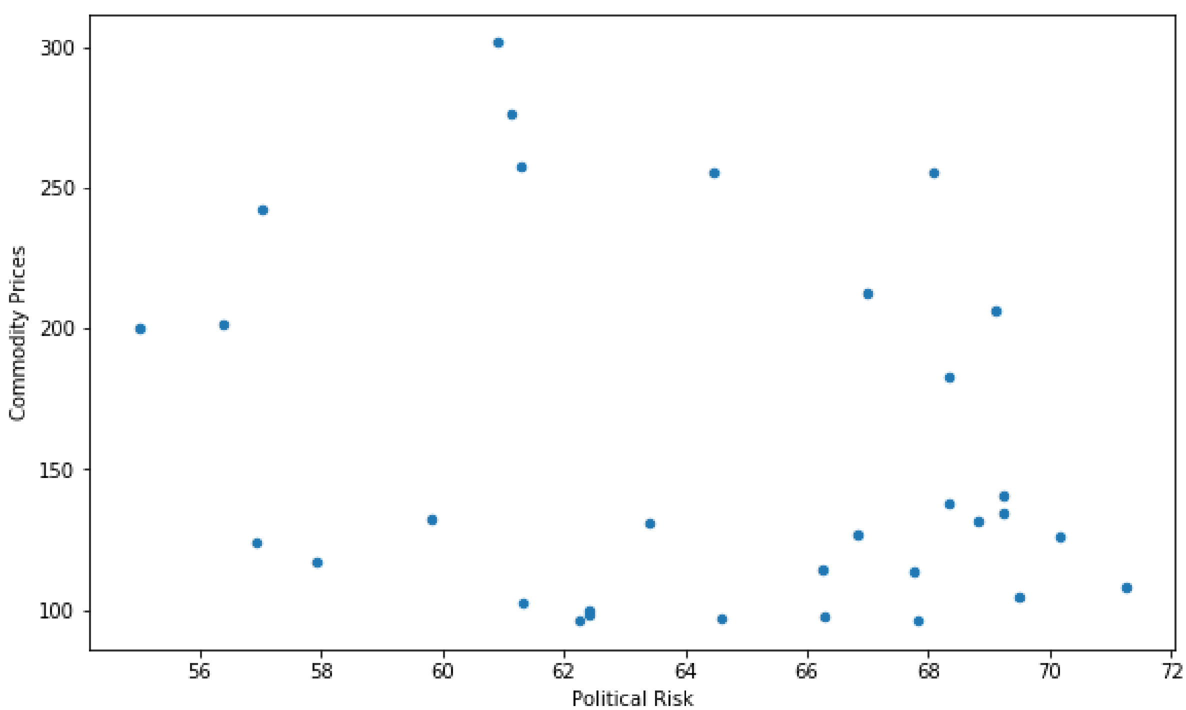
In [747]:
```
ChinaGDPPriceRisk.plot.scatter(x='Economic Risk',y='GDP',figsize=(10,6))
ChinaGDPPriceRisk['Economic Risk'].corr(ChinaGDPPriceRisk['GDP'])
```

Out[747]: 0.5257341795568661



In [748]:
```
ChinaGDPPriceRisk.plot.scatter(x='Political Risk',y='Commodity Prices',figsize
=(10,6))
ChinaGDPPriceRisk['Political Risk'].corr(ChinaGDPPriceRisk['Commodity Prices'
])
```

Out[748]: -0.2621544049133064

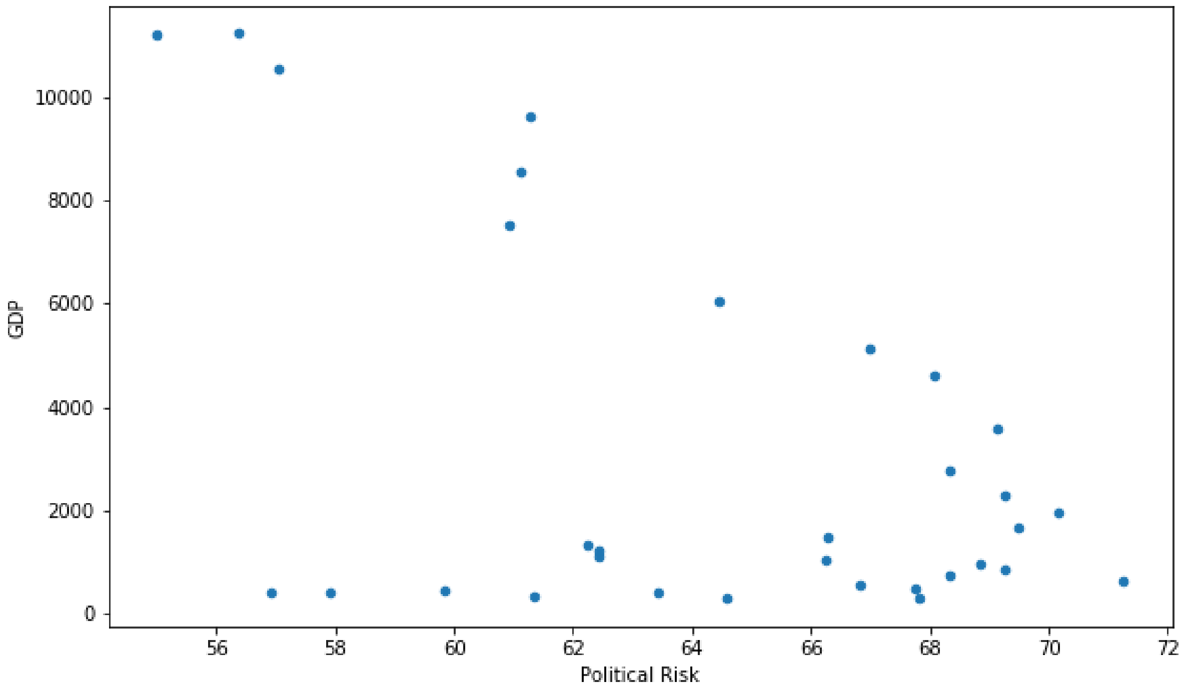


In [749]:
```
x=ChinaGDPPriceRisk[['Political Risk']]
y=ChinaGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.06
9
Model:                            OLS   Adj. R-squared:                  0.03
8
Method:                 Least Squares   F-statistic:                     2.21
4
Date:                Sat, 12 Oct 2019   Prob (F-statistic):              0.14
7
Time:                        22:41:52   Log-Likelihood:                -176.3
7
No. Observations:                  32   AIC:                             356.
7
Df Residuals:                      30   BIC:                             359.
7
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
const            387.2608    155.125      2.496      0.018      70.454     70
4.067
Political Risk    -3.5742      2.402     -1.488      0.147      -8.480
1.332
==============================================================================
=
Omnibus:                        4.142   Durbin-Watson:                   0.16
4
Prob(Omnibus):                  0.126   Jarque-Bera (JB):                3.66
7
Skew:                           0.756   Prob(JB):                        0.16
0
Kurtosis:                       2.319   Cond. No.                         91
6.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [750]:
```
ChinaGDPPriceRisk.plot.scatter(x='Political Risk',y='GDP',figsize=(10,6))
ChinaGDPPriceRisk['Political Risk'].corr(ChinaGDPPriceRisk['GDP'])
```

Out[750]: -0.5063602290920803

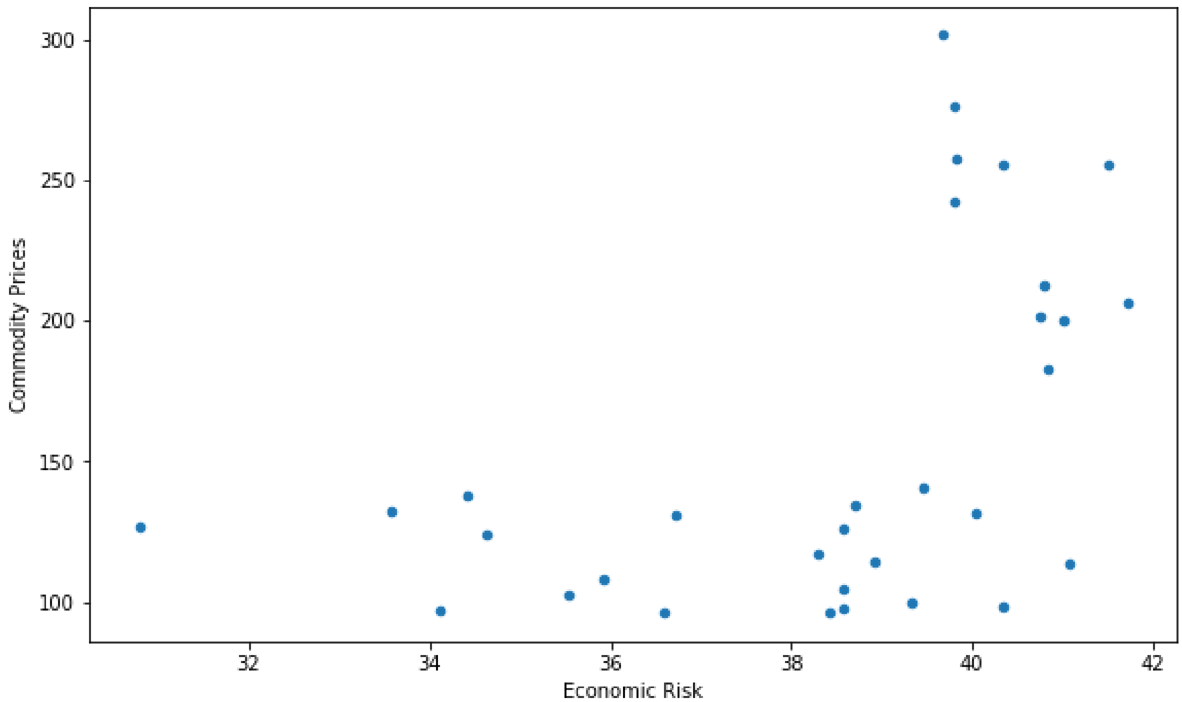


In [751]:
```
x=ChinaGDPPriceRisk[['Political Risk']]
y=ChinaGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.25
6
Model:                            OLS   Adj. R-squared:                  0.23
2
Method:                 Least Squares   F-statistic:                     10.3
4
Date:                Sat, 12 Oct 2019   Prob (F-statistic):            0.0031
1
Time:                        22:41:52   Log-Likelihood:                -302.2
4
No. Observations:                  32   AIC:                             608.
5
Df Residuals:                      30   BIC:                             611.
4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
const           2.854e+04   7923.122      3.602      0.001    1.24e+04    4.4
7e+04
Political Risk  -394.6037    122.690     -3.216      0.003    -645.171    -14
4.037
==============================================================================
=
Omnibus:                        2.759   Durbin-Watson:                   0.14
5
Prob(Omnibus):                  0.252   Jarque-Bera (JB):                1.34
8
Skew:                           0.033   Prob(JB):                        0.51
0
Kurtosis:                       1.997   Cond. No.                         91
6.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [752]:
```
ChinaGDPPriceRisk.plot.scatter(x='Economic Risk',y='Commodity Prices',figsize=
(10,6))
ChinaGDPPriceRisk['Economic Risk'].corr(ChinaGDPPriceRisk['Commodity Prices'])
```

Out[752]: 0.48556160926967085

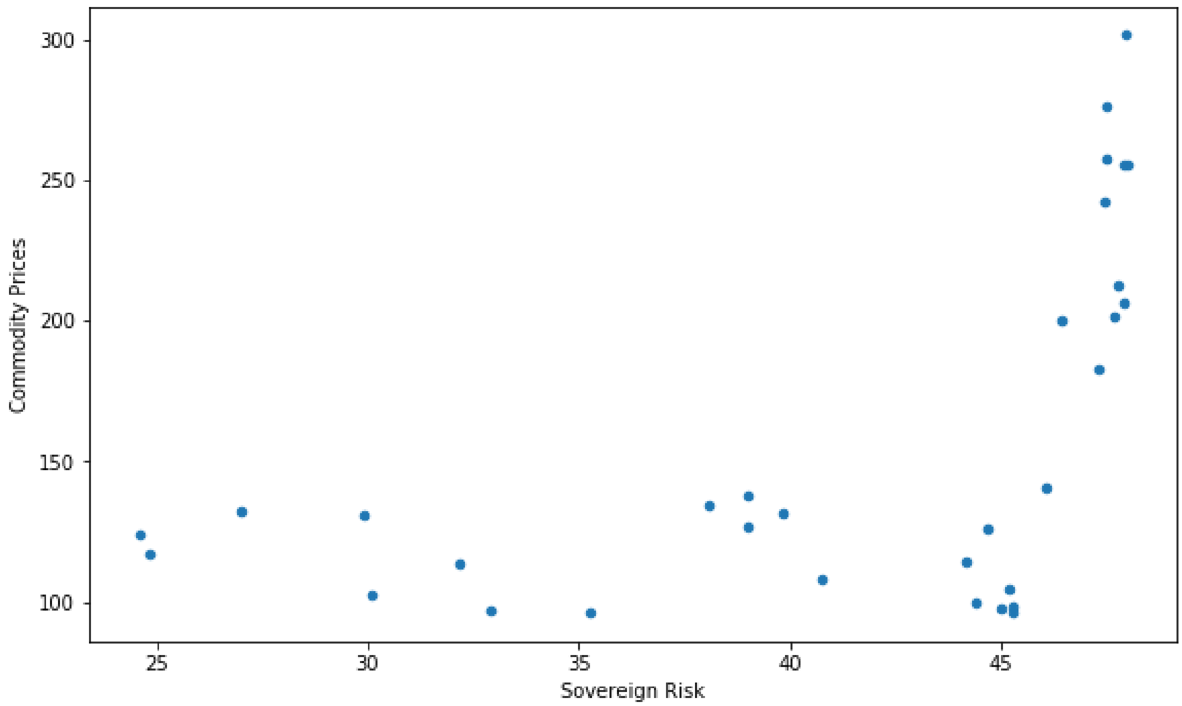


In [753]:
```
x=ChinaGDPPriceRisk[['Economic Risk']]
y=ChinaGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.23
6
Model:                            OLS   Adj. R-squared:                  0.21
0
Method:                 Least Squares   F-statistic:                     9.25
5
Date:                Sat, 12 Oct 2019   Prob (F-statistic):            0.0048
5
Time:                        22:41:53   Log-Likelihood:                -173.2
1
No. Observations:                  32   AIC:                             350.
4
Df Residuals:                      30   BIC:                             353.
4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
====
                   coef    std err          t      P>|t|      [0.025      0.
975]
------------------------------------------------------------------------------
----
const         -282.4631    144.801     -1.951      0.060    -578.186       1
3.259
Economic Risk   11.4469      3.763      3.042      0.005       3.763       1
9.131
==============================================================================
=
Omnibus:                        2.347   Durbin-Watson:                   0.37
4
Prob(Omnibus):                  0.309   Jarque-Bera (JB):                2.13
2
Skew:                           0.578   Prob(JB):                        0.34
4
Kurtosis:                       2.489   Cond. No.                         56
3.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [754]:
```
ChinaGDPPriceRisk.plot.scatter(x='Sovereign Risk',y='Commodity Prices',figsize
=(10,6))
ChinaGDPPriceRisk['Sovereign Risk'].corr(ChinaGDPPriceRisk['Commodity Prices'
])
```

Out[754]: 0.5503472794650955



In [755]:
```
x=ChinaGDPPriceRisk[['Sovereign Risk']]
y=ChinaGDPPriceRisk['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.30
3
Model:                            OLS   Adj. R-squared:                  0.28
0
Method:                 Least Squares   F-statistic:                     13.0
3
Date:                Sat, 12 Oct 2019   Prob (F-statistic):            0.0011
0
Time:                        22:41:53   Log-Likelihood:                -171.7
4
No. Observations:                  32   AIC:                             347.
5
Df Residuals:                      30   BIC:                             350.
4
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
const            -31.7945     53.149     -0.598      0.554    -140.338      7
6.749
Sovereign Risk     4.5875      1.271      3.610      0.001       1.992
7.183
==============================================================================
=
Omnibus:                        1.326   Durbin-Watson:                   0.24
9
Prob(Omnibus):                  0.515   Jarque-Bera (JB):                1.06
8
Skew:                           0.215   Prob(JB):                        0.58
6
Kurtosis:                       2.215   Cond. No.                         23
5.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [756]:
```
x=ChinaGDPPriceRisk[['Sovereign Risk','Political Risk','Economic Risk','Commodity Prices','Exports','Imports']]
y=ChinaGDPPriceRisk['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.90
1
Model:                            OLS   Adj. R-squared:                  0.87
7
Method:                 Least Squares   F-statistic:                     37.9
0
Date:                Sat, 12 Oct 2019   Prob (F-statistic):          2.26e-1
1
Time:                        22:41:53   Log-Likelihood:                -269.9
8
No. Observations:                  32   AIC:                             554.
0
Df Residuals:                      25   BIC:                             564.
2
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
=======
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const             -8.676e+05   4.09e+05     -2.121      0.044   -1.71e+06   -
2.52e+04
Sovereign Risk      177.1411     46.330      3.823      0.001      81.723
272.559
Political Risk     -306.3607     61.249     -5.002      0.000    -432.506    -
180.215
Economic Risk         8.9299    114.772      0.078      0.939    -227.447
245.306
Commodity Prices     36.9303     12.066      3.061      0.005      12.080
61.780
Exports            8977.5925   4174.254      2.151      0.041     380.555
1.76e+04
Imports            -217.6812    106.716     -2.040      0.052    -437.468
2.105
==============================================================================
=
Omnibus:                        3.050   Durbin-Watson:                   0.66
0
Prob(Omnibus):                  0.218   Jarque-Bera (JB):                1.59
5
Skew:                           0.223   Prob(JB):                        0.45
0
Kurtosis:                       2.002   Cond. No.                    4.06e+0
5
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 4.06e+05. This might indicate that there a
```

```
re
strong multicollinearity or other numerical problems.
```

It is also important to note that looking at the overall economic structure and how it has been performing relatively to commodity prices and GDP output is also important to consider. While previous models focused primarily on Sovereign, Political, and Country Risk variables, it is also worth mentioning about it is important to look at how the EM's overall Economy's health is. As seen with the results with China, Economic Risk is certainly a huge influence among commodity prices and economic output. Evenmore, Economic Risk and Sovereign Risk appear to have a strong correlation between each other as this implies the importance to measure an EM's financial health with its economic health outside any political influence before aggregating both political and economic variables when determining which among Sovereign, Political, and Country Risk has the biggest influence among Commodity Prices and EMs' GDP output.

Conclusion: Using Machine Learning Classification to predict likelidhood of Sovereign Risk making a High Impact on GDP

In [757]:
```
x=PriceRiskXIM1[['Sovereign Risk', 'Political Risk', 'Country Risk']]
y=GDPvPrices['GDP']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.65
7
Model:                            OLS   Adj. R-squared:                  0.62
1
Method:                 Least Squares   F-statistic:                     17.9
1
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           1.10e-0
6
Time:                        22:41:53   Log-Likelihood:                -218.3
5
No. Observations:                  32   AIC:                             444.
7
Df Residuals:                      28   BIC:                             450.
6
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
const            973.7285    705.695      1.380      0.179    -471.823    241
9.280
Sovereign Risk    93.3693     21.906      4.262      0.000      48.497     13
8.241
Political Risk  -102.2460     43.744     -2.337      0.027    -191.851     -1
2.641
Country Risk      52.0312     69.988      0.743      0.463     -91.332     19
5.395
==============================================================================
=
Omnibus:                        0.617   Durbin-Watson:                   0.33
9
Prob(Omnibus):                  0.734   Jarque-Bera (JB):                0.71
0
Skew:                          -0.271   Prob(JB):                        0.70
1
Kurtosis:                       2.511   Cond. No.                     1.48e+0
3
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 1.48e+03. This might indicate that there a
re
strong multicollinearity or other numerical problems.
```

While noting the apparent inverse relationship between Country Risk and Sovereign Risk, one should still keep in mind that regardless, Sovereign Risk is still a variable that is to not be undermined and still taken into account when accurately making forecasts of commodity prices ability to positively impact EM's GDP output.

Before concluding, Machine Learning Classification will be used to see if the models used for all of the Emerging Markets, China, India, and Argentina is telling enough to predict whether their respective risk variable in examination will highly impact GDP. Machine Learning Classification will also be used to examine whether the data presented on Commodity Prices and Sovereign Risk is telling enough to predict whether Sovereign Risk will highly impact Commodity Prices. The various Machine Learning Classification algorithms used include: Logistic Regression, K Nearest Neighbors (KNN), and Random Forest Classifier. For each model, the algorithms will be used initially to obtain a score. Whichever algorithm produces the highest score will be used to predict whether the data is telling enough to predict a high impact. The variable to be predicted, y-hat, in the models involving Emerging Markets, and the individual countries examined is 'Impact_High' and 'GDPImpact_High'. For the model involving commodity prices and sovereign risk, the y-hat is 'Impact_High'.

For each model, analysis was done on Excel before being imported into Jupyter on the y-hat. For GDP and Commodity prices, the dependent variables, every instance per year was averaged to calculate the distance a particular instance was from the mean. From there, using the data from distance from mean a threshold was created among for each model using one of three measures, interquartile range, midrange,and standard deviation, to go through the dependent variable in each model to determine if it was highly or lowly impacted. As for the independent variables, the risk variables from the ICRG, the ICRG database provided a list of Risk Categories for certain ranges of scores, which are Very Low Risk, Low Risk, Moderate Risk, High Risk, and Very High Risk.

Once the data was imported into Jupyter, ICRG Risk Category and Impact Category variables were converted to categorical values using label encoding in order for the data to be used to run the classification algorithms.

Machine Learning Classification

China Machine Learning Classification: Economic Risk's impact on GDP

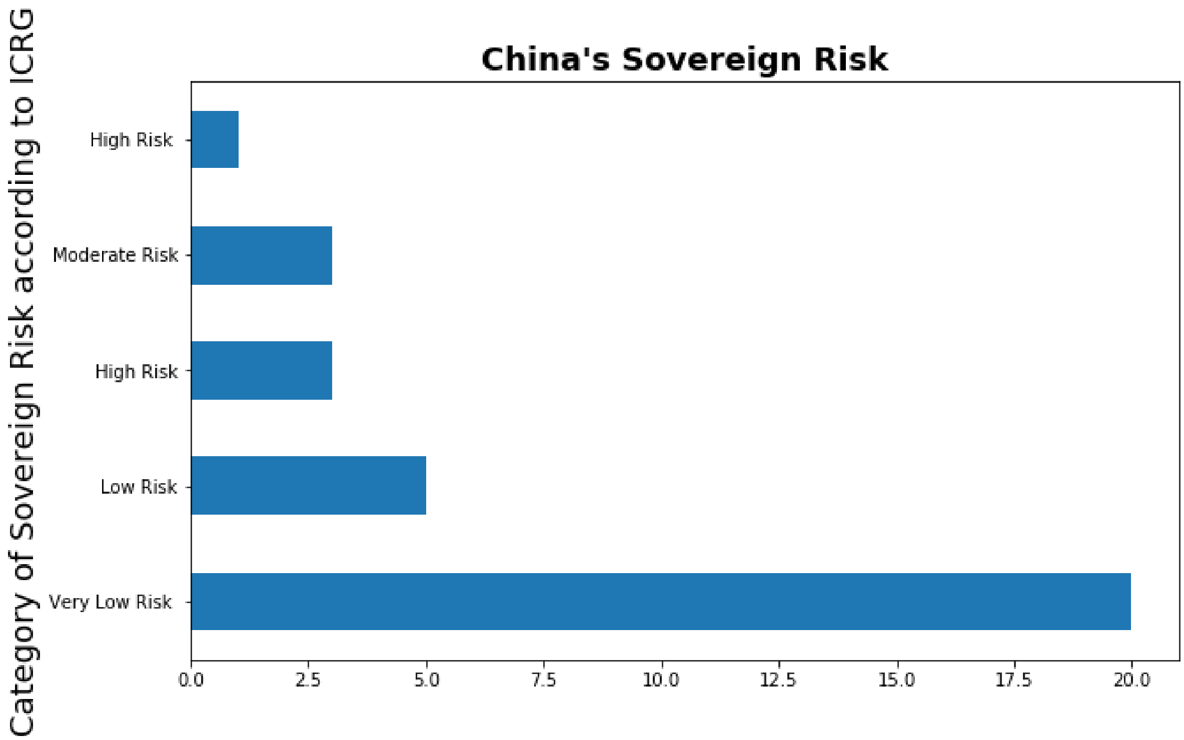
In [758]:
```
ChinaML1=pd.read_excel (r"C:\New folder\China Econ Risk Data .xlsx")
ChinaML1=ChinaML1.drop(0)
ChinaML1=ChinaML1.rename(columns={'Unnamed: 0':'Year','Unnamed: 1':'Category o
f Risk according to ICRG','Unnamed: 2':'Economic Risk','Unnamed: 3':'GDP','Unn
amed: 4':'Impact'})
ChinaML1
```

Out[758]:

	Year	Category of Risk according to ICRG	Economic Risk	GDP	Impact
1	1985	Low Risk	36.58	312.616	High
2	1986	Moderate Risk	34.12	303.34	High
3	1987	Moderate Risk	35.54	330.303	High
4	1988	High Risk	36.71	411.923	High
5	1989	High Risk	33.58	461.066	High
6	1990	High Risk	34.62	398.623	High
7	1991	High Risk	38.29	415.604	High
8	1992	Moderate Risk	41.08	495.671	High
9	1993	Very Low Risk	35.92	623.054	High
10	1994	Low Risk	30.79	566.471	High
11	1995	Low Risk	34.42	736.87	High
12	1996	Low Risk	38.71	867.224	High
13	1997	Low Risk	40.04	965.338	High
14	1998	Very Low Risk	38.92	1032.57	High
15	1999	Very Low Risk	40.33	1097.14	High
16	2000	Very Low Risk	39.33	1214.92	High
17	2001	Very Low Risk	38.42	1344.08	High
18	2002	Very Low Risk	38.58	1477.5	High
19	2003	Very Low Risk	38.58	1671.07	High
20	2004	Very Low Risk	38.58	1966.24	High
21	2005	Very Low Risk	39.46	2308.8	Low
22	2006	Very Low Risk	40.83	2774.29	Low
23	2007	Very Low Risk	41.71	3571.45	Low
24	2008	Very Low Risk	41.5	4604.29	High
25	2009	Very Low Risk	40.79	5121.68	High
26	2010	Very Low Risk	40.33	6066.35	High
27	2011	Very Low Risk	39.67	7522.1	High
28	2012	Very Low Risk	39.79	8570.35	High
29	2013	Very Low Risk	39.83	9635.03	High
30	2014	Very Low Risk	39.79	10534.5	High
31	2015	Very Low Risk	40.75	11226.2	High
32	2016	Very Low Risk	41	11221.8	High

In [759]:
```
fig, axe = plt.subplots()
ChinaML1['Category of Risk according to ICRG'].value_counts().plot.barh(figsiz
e =(10,6))

axe.set_ylabel("Category of Sovereign Risk according to ICRG", fontsize=18)

axe.set_title("China's Sovereign Risk", loc='center', fontsize=18, fontweight
= "bold" )
```

Out[759]: Text(0.5, 1.0, "China's Sovereign Risk")



In [760]:
```
ChinaML1.dtypes
```

Out[760]:
```
Year                                    object
Category of Risk according to ICRG      object
Economic Risk                           object
GDP                                     object
Impact                                  object
dtype: object
```

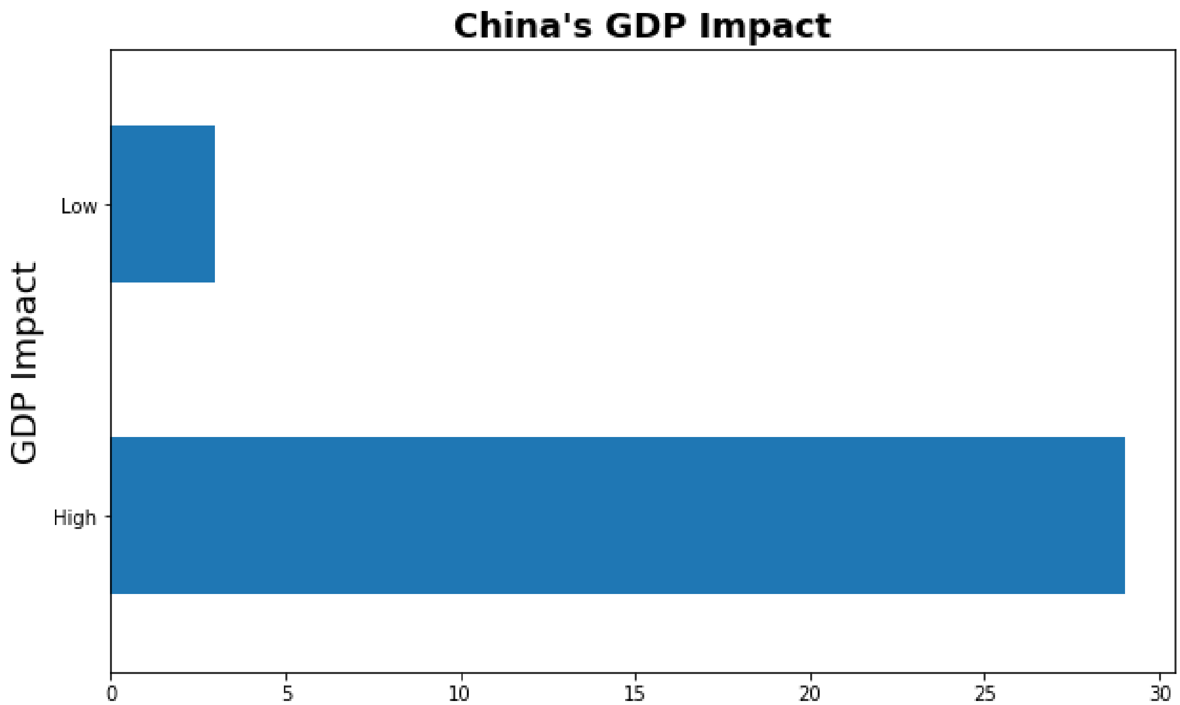
In [761]:
```
ChinaML1['Category of Risk according to ICRG']=ChinaML1['Category of Risk acco
rding to ICRG'].astype('category')
ChinaML1.dtypes
```

Out[761]:
```
Year                                    object
Category of Risk according to ICRG    category
Economic Risk                           object
GDP                                     object
Impact                                  object
dtype: object
```

In [762]:
```
ChinaML1['Category of Risk according to ICRG']=ChinaML1['Category of Risk acco
rding to ICRG'].cat.codes
```

In [763]:
```
fig, axe = plt.subplots()
ChinaML1['Impact'].value_counts().plot.barh(figsize =(10,6))

axe.set_ylabel("GDP Impact", fontsize=18)

axe.set_title("China's GDP Impact", loc='center', fontsize=18, fontweight = "b
old" )
```

Out[763]: Text(0.5, 1.0, "China's GDP Impact")



In [764]:
```
ChinaML1=pd.get_dummies(ChinaML1, columns=['Impact'])
```

In [765]:
```
ChinaML1
```

Out[765]:

	Year	Category of Risk according to ICRG	Economic Risk	GDP	Impact_High	Impact_Low
1	1985	2	36.58	312.616	1	0
2	1986	3	34.12	303.34	1	0
3	1987	3	35.54	330.303	1	0
4	1988	0	36.71	411.923	1	0
5	1989	0	33.58	461.066	1	0
6	1990	0	34.62	398.623	1	0
7	1991	1	38.29	415.604	1	0
8	1992	3	41.08	495.671	1	0
9	1993	4	35.92	623.054	1	0
10	1994	2	30.79	566.471	1	0
11	1995	2	34.42	736.87	1	0
12	1996	2	38.71	867.224	1	0
13	1997	2	40.04	965.338	1	0
14	1998	4	38.92	1032.57	1	0
15	1999	4	40.33	1097.14	1	0
16	2000	4	39.33	1214.92	1	0
17	2001	4	38.42	1344.08	1	0
18	2002	4	38.58	1477.5	1	0
19	2003	4	38.58	1671.07	1	0
20	2004	4	38.58	1966.24	1	0
21	2005	4	39.46	2308.8	0	1
22	2006	4	40.83	2774.29	0	1
23	2007	4	41.71	3571.45	0	1
24	2008	4	41.5	4604.29	1	0
25	2009	4	40.79	5121.68	1	0
26	2010	4	40.33	6066.35	1	0
27	2011	4	39.67	7522.1	1	0
28	2012	4	39.79	8570.35	1	0
29	2013	4	39.83	9635.03	1	0
30	2014	4	39.79	10534.5	1	0
31	2015	4	40.75	11226.2	1	0
32	2016	4	41	11221.8	1	0

In [766]:

```
ChinaML1=ChinaML1.rename(columns={'Category of Risk according to ICRG':'Risk_CategoryICRG'})
ChinaML1
```

Out[766]:

	Year	Risk_CategoryICRG	Economic Risk	GDP	Impact_High	Impact_Low
1	1985	2	36.58	312.616	1	0
2	1986	3	34.12	303.34	1	0
3	1987	3	35.54	330.303	1	0
4	1988	0	36.71	411.923	1	0
5	1989	0	33.58	461.066	1	0
6	1990	0	34.62	398.623	1	0
7	1991	1	38.29	415.604	1	0
8	1992	3	41.08	495.671	1	0
9	1993	4	35.92	623.054	1	0
10	1994	2	30.79	566.471	1	0
11	1995	2	34.42	736.87	1	0
12	1996	2	38.71	867.224	1	0
13	1997	2	40.04	965.338	1	0
14	1998	4	38.92	1032.57	1	0
15	1999	4	40.33	1097.14	1	0
16	2000	4	39.33	1214.92	1	0
17	2001	4	38.42	1344.08	1	0
18	2002	4	38.58	1477.5	1	0
19	2003	4	38.58	1671.07	1	0
20	2004	4	38.58	1966.24	1	0
21	2005	4	39.46	2308.8	0	1
22	2006	4	40.83	2774.29	0	1
23	2007	4	41.71	3571.45	0	1
24	2008	4	41.5	4604.29	1	0
25	2009	4	40.79	5121.68	1	0
26	2010	4	40.33	6066.35	1	0
27	2011	4	39.67	7522.1	1	0
28	2012	4	39.79	8570.35	1	0
29	2013	4	39.83	9635.03	1	0
30	2014	4	39.79	10534.5	1	0
31	2015	4	40.75	11226.2	1	0
32	2016	4	41	11221.8	1	0

In [767]:
```
ChinaML1.columns
```

Out[767]:
```
Index(['Year', 'Risk_CategoryICRG', 'Economic Risk', 'GDP', 'Impact_High',
       'Impact_Low'],
      dtype='object')
```

In [768]:
```
ChinaML1.Impact_High.value_counts()
```

Out[768]:
```
1    29
0     3
Name: Impact_High, dtype: int64
```

In [769]:
```
from patsy import dmatrices
y,X = dmatrices('Impact_High ~ Risk_CategoryICRG',data=ChinaML1)
```

In [770]:
```
y=np.ravel(y)
```

In [771]:
```
from sklearn.linear_model import LogisticRegression as logit
logit(solver='lbfgs').fit(X,y).score(X,y)
```

Out[771]:
```
array(0.90625)
```

In [772]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(logit(solver='lbfgs'),X,y,cv=5,scoring='accuracy').mean()
```

```
C:\Users\ishaa\Anaconda3NEW1\lib\site-packages\sklearn\model_selection\_spli
t.py:652: Warning: The least populated class in y has only 3 members, which i
s too few. The minimum number of members in any class cannot be less than n_s
plits=5.
  % (min_groups, self.n_splits)), Warning)
```

Out[772]:
```
0.9142857142857143
```

In [773]:
```
cross_val_score(logit(solver='lbfgs'),X,y,cv=3,scoring='roc_auc').mean()
```

Out[773]:
```
0.7000000000000001
```

In [774]:
```
from sklearn.neighbors import KNeighborsClassifier as knn
cross_val_score(knn(),X,y,cv=3,scoring='accuracy').mean()
```

Out[774]:
```
0.9060606060606061
```

In [775]:
```
from sklearn.ensemble import RandomForestClassifier as rf
cross_val_score(rf(n_estimators=100, max_depth=2,max_features=2),X,y,cv=2,scor
ing='accuracy').mean()
```

Out[775]:
```
0.907843137254902
```

In [776]:
```
for i in range(1,10):
    for j in range(1,3):
        print(i,j,cross_val_score(rf(n_estimators=100,max_depth=i,max_features
=j),X,y,cv=3,scoring='accuracy').mean())
```

```
1 1 0.9060606060606061
1 2 0.9060606060606061
2 1 0.9060606060606061
2 2 0.9060606060606061
3 1 0.9060606060606061
3 2 0.9060606060606061
4 1 0.9060606060606061
4 2 0.9060606060606061
5 1 0.9060606060606061
5 2 0.9060606060606061
6 1 0.9060606060606061
6 2 0.9060606060606061
7 1 0.9060606060606061
7 2 0.9060606060606061
8 1 0.9060606060606061
8 2 0.9060606060606061
9 1 0.9060606060606061
9 2 0.9060606060606061
```

In [777]:
```
ChinaML1.columns
```

Out[777]:
```
Index(['Year', 'Risk_CategoryICRG', 'Economic Risk', 'GDP', 'Impact_High',
       'Impact_Low'],
      dtype='object')
```

In [778]:
```
from sklearn.linear_model import LogisticRegression as reg
from sklearn.model_selection import train_test_split
```

In [779]:
```
x=ChinaML1.drop(['Impact_High'],axis=1)
y=ChinaML1['Impact_High']
```

In [780]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.2)
```

In [781]:
```
result = reg(solver='liblinear').fit(x_train,y_train).predict(x_test)
```

In [782]:
```
pd.Series(result).value_counts()
```

Out[782]:
```
1    7
dtype: int64
```

This model shows that in the next seven years from 2017 to 2023, Economic Risk will have a 100 % chance of highly impacting GDP.

China Machine Learning Classification: Economic Risk's Impact on Sovereign Risk

In [783]:
```
ChinaML2=pd.read_excel (r"C:\New folder\China Econ Sovereign Risk .xlsx")
ChinaML2=ChinaML2.drop(0)
ChinaML2=ChinaML2.rename(columns={'Unnamed: 0':'Year','Unnamed: 1':'Category o
f Economic Risk according to ICRG','Unnamed: 2':'Economic Risk','Unnamed: 3':
'Category of Sovereign Risk according to ICRG','Unnamed: 4':'Sovereign Risk',
'Unnamed: 5':'Impact'})
ChinaML2
```

Out[783]:

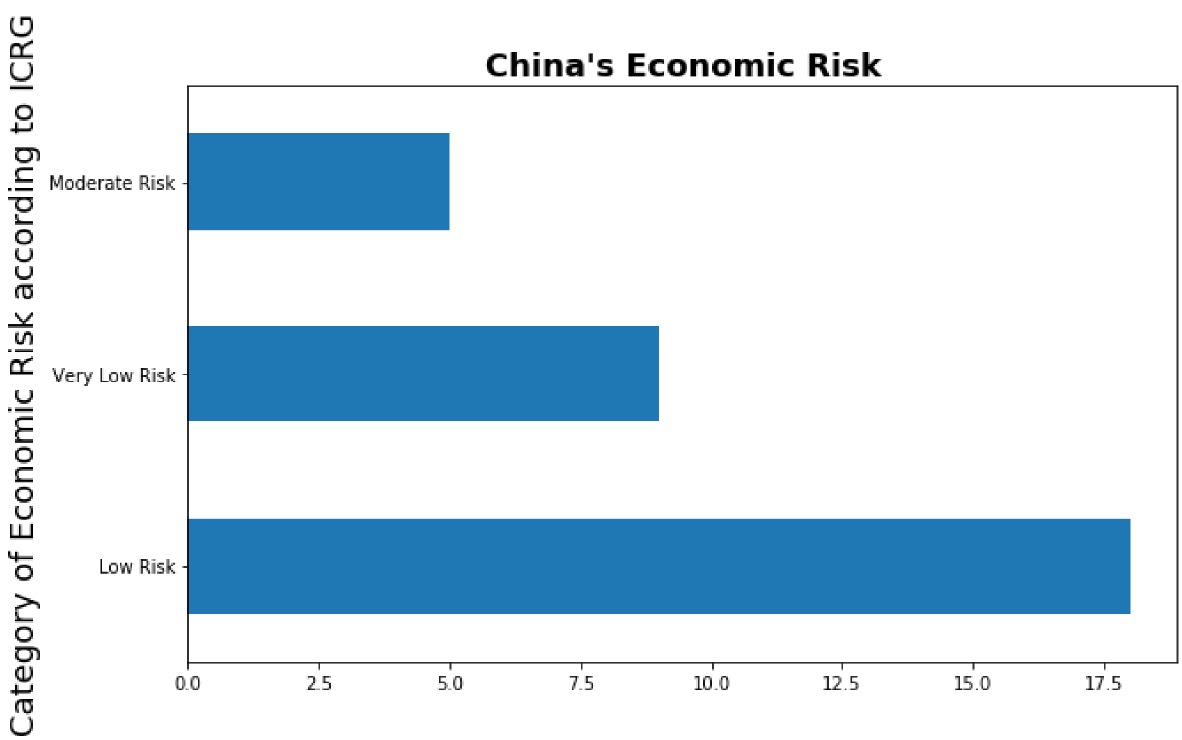
	Year	Category of Economic Risk according to ICRG	Economic Risk	Category of Sovereign Risk according to ICRG	Sovereign Risk	Impact
1	1985	Low Risk	36.58	Low Risk	35.25	High
2	1986	Moderate Risk	34.12	Moderate Risk	32.92	High
3	1987	Low Risk	35.54	Moderate Risk	30.08	High
4	1988	Low Risk	36.71	High Risk	29.92	High
5	1989	Moderate Risk	33.58	High Risk	27	High
6	1990	Moderate Risk	34.62	High Risk	24.58	High
7	1991	Low Risk	38.29	High Risk	24.83	High
8	1992	Very Low Risk	41.08	Moderate Risk	32.17	High
9	1993	Low Risk	35.92	Very Low Risk	40.75	Low
10	1994	Moderate Risk	30.79	Low Risk	39	Low
11	1995	Low Risk	34.42	Low Risk	39	Low
12	1996	Moderate Risk	38.71	Low Risk	38.08	Low
13	1997	Very Low Risk	40.04	Low Risk	39.83	Low
14	1998	Low Risk	38.92	Very Low Risk	44.17	Low
15	1999	Very Low Risk	40.33	Very Low Risk	45.29	High
16	2000	Low Risk	39.33	Very Low Risk	44.42	High
17	2001	Low Risk	38.42	Very Low Risk	45.29	High
18	2002	Low Risk	38.58	Very Low Risk	45	High
19	2003	Low Risk	38.58	Very Low Risk	45.21	High
20	2004	Low Risk	38.58	Very Low Risk	44.67	High
21	2005	Low Risk	39.46	Very Low Risk	46.08	High
22	2006	Very Low Risk	40.83	Very Low Risk	47.33	High
23	2007	Very Low Risk	41.71	Very Low Risk	47.92	High
24	2008	Very Low Risk	41.5	Very Low Risk	47.92	High
25	2009	Very Low Risk	40.79	Very Low Risk	47.79	High
26	2010	Low Risk	40.33	Very Low Risk	48	High
27	2011	Low Risk	39.67	Very Low Risk	47.96	High
28	2012	Low Risk	39.79	Very Low Risk	47.5	High
29	2013	Low Risk	39.83	Very Low Risk	47.5	High
30	2014	Low Risk	39.79	Very Low Risk	47.46	High
31	2015	Very Low Risk	40.75	Very Low Risk	47.71	High
32	2016	Very Low Risk	41	Very Low Risk	46.46	High

In [784]:
```
ChinaML2['Category of Sovereign Risk according to ICRG']=ChinaML2['Category of Sovereign Risk according to ICRG'].astype('category')
ChinaML2['Category of Economic Risk according to ICRG']=ChinaML2['Category of Economic Risk according to ICRG'].astype('category')

ChinaML2.dtypes
```

Out[784]:
```
Year                                              object
Category of Economic Risk according to ICRG     category
Economic Risk                                     object
Category of Sovereign Risk according to ICRG    category
Sovereign Risk                                    object
Impact                                            object
dtype: object
```

In [785]:
```
fig, axe = plt.subplots()
ChinaML2['Category of Economic Risk according to ICRG'].value_counts().plot.barh(figsize =(10,6))

axe.set_ylabel("Category of Economic Risk according to ICRG", fontsize=18)

axe.set_title("China's Economic Risk", loc='center', fontsize=18, fontweight = "bold" )
```

Out[785]: Text(0.5, 1.0, "China's Economic Risk")



In [786]:
```
ChinaML2=pd.get_dummies(ChinaML2, columns=['Impact'])
```

In [787]:
```
ChinaML2['Category of Economic Risk according to ICRG']=ChinaML2['Category of Economic Risk according to ICRG'].cat.codes
```

In [788]:
```
ChinaML2['Category of Sovereign Risk according to ICRG']=ChinaML2['Category of Sovereign Risk according to ICRG'].cat.codes
```

In [789]:
```
ChinaML2
```

Out[789]:

	Year	Category of Economic Risk according to ICRG	Economic Risk	Category of Sovereign Risk according to ICRG	Sovereign Risk	Impact_High	Impact_Low
1	1985	0	36.58	2	35.25	1	0
2	1986	1	34.12	3	32.92	1	0
3	1987	0	35.54	3	30.08	1	0
4	1988	0	36.71	0	29.92	1	0
5	1989	1	33.58	0	27	1	0
6	1990	1	34.62	0	24.58	1	0
7	1991	0	38.29	1	24.83	1	0
8	1992	2	41.08	3	32.17	1	0
9	1993	0	35.92	4	40.75	0	1
10	1994	1	30.79	2	39	0	1
11	1995	0	34.42	2	39	0	1
12	1996	1	38.71	2	38.08	0	1
13	1997	2	40.04	2	39.83	0	1
14	1998	0	38.92	4	44.17	0	1
15	1999	2	40.33	4	45.29	1	0
16	2000	0	39.33	4	44.42	1	0
17	2001	0	38.42	4	45.29	1	0
18	2002	0	38.58	4	45	1	0
19	2003	0	38.58	4	45.21	1	0
20	2004	0	38.58	4	44.67	1	0
21	2005	0	39.46	4	46.08	1	0
22	2006	2	40.83	4	47.33	1	0
23	2007	2	41.71	4	47.92	1	0
24	2008	2	41.5	4	47.92	1	0
25	2009	2	40.79	4	47.79	1	0
26	2010	0	40.33	4	48	1	0
27	2011	0	39.67	4	47.96	1	0
28	2012	0	39.79	4	47.5	1	0
29	2013	0	39.83	4	47.5	1	0
30	2014	0	39.79	4	47.46	1	0
31	2015	2	40.75	4	47.71	1	0
32	2016	2	41	4	46.46	1	0

In [790]:
```
ChinaML2=ChinaML2.rename(columns={'Category of Economic Risk according to ICR
G':'EconRisk_CategoryICRG'})
```

In [791]:
```
ChinaML2.columns
```

Out[791]:
```
Index(['Year', 'EconRisk_CategoryICRG', 'Economic Risk',
       'Category of Sovereign Risk according to ICRG', 'Sovereign Risk',
       'Impact_High', 'Impact_Low'],
      dtype='object')
```

In [792]:
```
ChinaML2.Impact_High.value_counts()
```

Out[792]:
```
1    26
0     6
Name: Impact_High, dtype: int64
```

In [793]:
```
from patsy import dmatrices
y,X = dmatrices('Impact_High ~ EconRisk_CategoryICRG',data=ChinaML2)
```

In [794]:
```
y=np.ravel(y)
```

In [795]:
```
from sklearn.linear_model import LogisticRegression as logit
logit(solver='lbfgs').fit(X,y).score(X,y)
```

Out[795]:
```
array(0.8125)
```

In [796]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(logit(solver='lbfgs'),X,y,cv=5,scoring='accuracy').mean()
```

Out[796]:
```
0.8166666666666668
```

In [797]:
```
cross_val_score(logit(solver='lbfgs'),X,y,cv=3,scoring='roc_auc').mean()
```

Out[797]:
```
0.4513888888888889
```

In [798]:
```
from sklearn.neighbors import KNeighborsClassifier as knn
cross_val_score(knn(),X,y,cv=3,scoring='accuracy').mean()
```

Out[798]:
```
0.7515151515151516
```

In [799]:
```
from sklearn.ensemble import RandomForestClassifier as rf
cross_val_score(rf(n_estimators=100, max_depth=2,max_features=2),X,y,cv=2,scor
ing='accuracy').mean()
```

Out[799]:
```
0.75
```

In [800]:
```
for i in range(1,10):
    for j in range(1,3):
        print(i,j,cross_val_score(rf(n_estimators=100,max_depth=i,max_features
=j),X,y,cv=3,scoring='accuracy').mean())
```

```
1 1 0.8121212121212121
1 2 0.8121212121212121
2 1 0.7515151515151516
2 2 0.7515151515151516
3 1 0.7515151515151516
3 2 0.7515151515151516
4 1 0.7515151515151516
4 2 0.7515151515151516
5 1 0.7515151515151516
5 2 0.7515151515151516
6 1 0.7515151515151516
6 2 0.7515151515151516
7 1 0.7515151515151516
7 2 0.7515151515151516
8 1 0.7515151515151516
8 2 0.7515151515151516
9 1 0.7515151515151516
9 2 0.7515151515151516
```

In [801]:
```
from sklearn.linear_model import LogisticRegression as reg
from sklearn.model_selection import train_test_split
```

In [802]:
```
x=ChinaML2.drop(['Impact_High'],axis=1)
y=ChinaML2['Impact_High']
```

In [803]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.2)
```

In [804]:
```
result = reg(solver='lbfgs').fit(x_train,y_train).predict(x_test)
```

In [805]:
```
pd.Series(result).value_counts()
```

Out[805]:
```
1    7
dtype: int64
```

This model shows that in the next seven years, from 2017 to 2023, Economic Risk will have a 100% likelihood of highly impacting Sovereign Risk.

India Machine Learning Classification: Country Risk's Impact on GDP

In [806]:
```
IndiaML=pd.read_excel (r"C:\New folder\India Country Risk Data .xlsx")
IndiaML=IndiaML.drop(0)
IndiaML=IndiaML.drop(1)
IndiaML=IndiaML.rename(columns={'Unnamed: 0':'Year','Unnamed: 1':'Category of Risk according to ICRG','Unnamed: 2':'Country Risk','Unnamed: 3':'GDP','Unnamed: 4':'Impact'})
IndiaML
```

Out[806]:

	Year	Category of Risk according to ICRG	Country Risk	GDP	Impact
2	1985	Low Risk	39.085	237.618	High
3	1986	Very Low Risk	42.415	252.751	High
4	1987	Very Low Risk	40.6	283.75	High
5	1988	Low Risk	37.895	299.645	High
6	1989	Low Risk	36.665	300.187	High
7	1990	Moderate Risk	34.455	326.608	High
8	1991	Moderate Risk	31.04	274.842	High
9	1992	Low Risk	37.435	293.262	High
10	1993	Very Low Risk	42.73	284.194	High
11	1994	Very Low Risk	49.185	333.014	High
12	1995	Very Low Risk	49.935	366.6	High
13	1996	Very Low Risk	49.605	399.791	High
14	1997	Very Low Risk	50.25	423.189	High
15	1998	Very Low Risk	46.81	428.767	High
16	1999	Very Low Risk	44.035	466.841	High
17	2000	Very Low Risk	44.25	476.636	High
18	2001	Very Low Risk	44.935	493.934	High
19	2002	Very Low Risk	45.21	523.768	High
20	2003	Very Low Risk	46.48	618.369	Low
21	2004	Very Low Risk	49.42	721.589	Low
22	2005	Very Low Risk	49.605	834.218	Low
23	2006	Very Low Risk	49.54	949.118	Low
24	2007	Very Low Risk	48.98	1238.7	High
25	2008	Very Low Risk	46.705	1224.1	High
26	2009	Very Low Risk	48.335	1365.37	High
27	2010	Very Low Risk	47.48	1708.46	High
28	2011	Very Low Risk	45.855	1823.05	High
29	2012	Very Low Risk	45.31	1827.64	High
30	2013	Very Low Risk	45.685	1856.72	High
31	2014	Very Low Risk	46.75	2039.13	High
32	2015	Very Low Risk	48.395	2103.59	High
33	2016	Very Low Risk	49.19	2289.75	High

In [807]:
```
IndiaML.dtypes
```

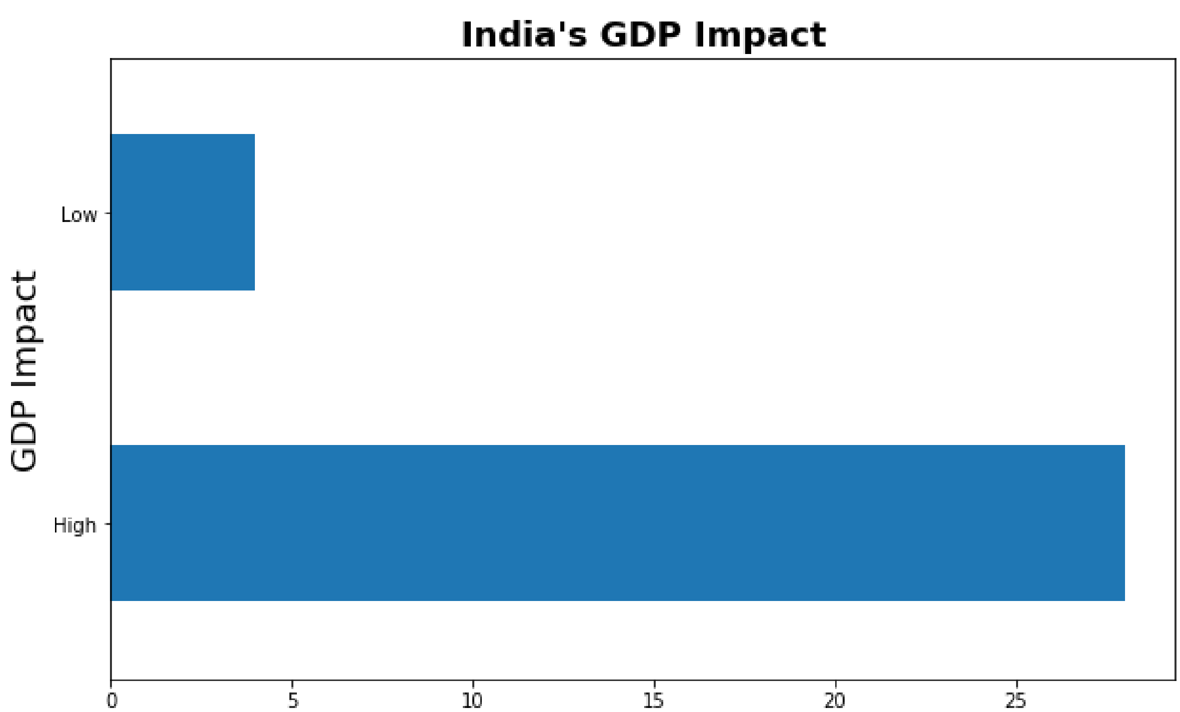
Out[807]:
```
Year                                    object
Category of Risk according to ICRG    object
Country Risk                            object
GDP                                     object
Impact                                  object
dtype: object
```

In [808]:
```
IndiaML['Category of Risk according to ICRG']=IndiaML['Category of Risk according to ICRG'].astype('category')
IndiaML.dtypes
```

Out[808]:
```
Year                                      object
Category of Risk according to ICRG    category
Country Risk                              object
GDP                                       object
Impact                                    object
dtype: object
```

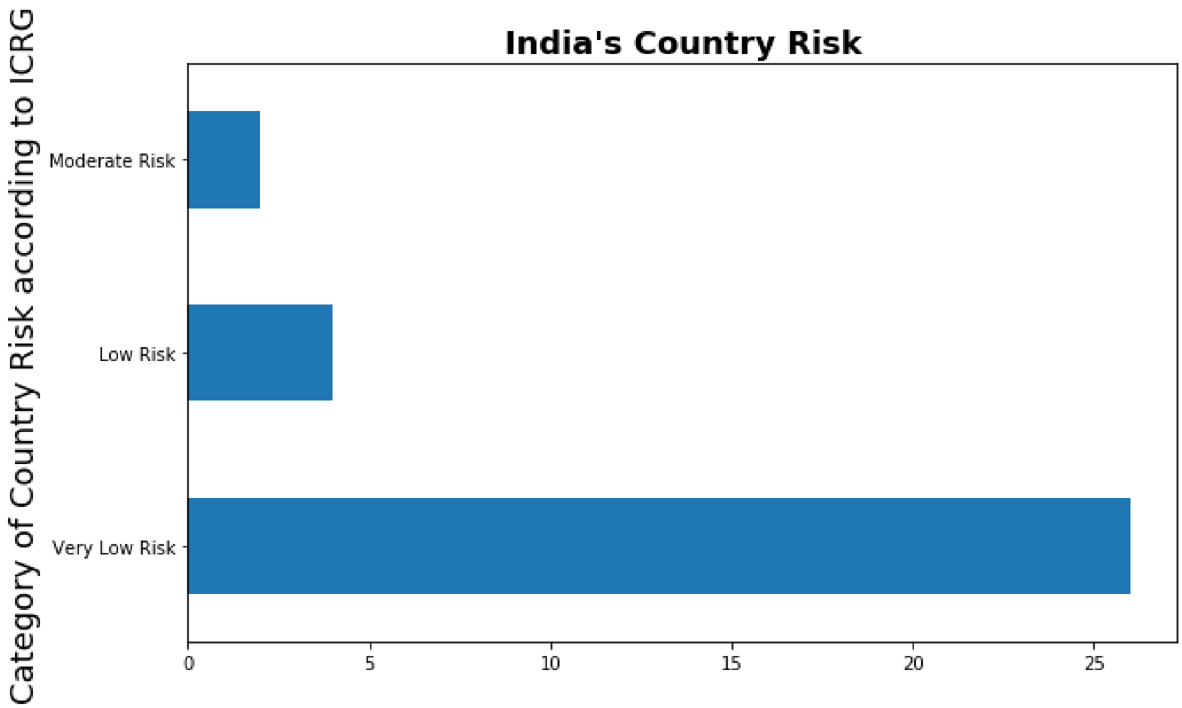
In [809]:
```
fig, axe = plt.subplots()
IndiaML['Impact'].value_counts().plot.barh(figsize =(10,6))

axe.set_ylabel("GDP Impact", fontsize=18)

axe.set_title("India's GDP Impact", loc='center', fontsize=18, fontweight = "bold" )
```

Out[809]:
```
Text(0.5, 1.0, "India's GDP Impact")
```



In [810]:
```
fig, axe = plt.subplots()
IndiaML['Category of Risk according to ICRG'].value_counts().plot.barh(figsize
=(10,6))

axe.set_ylabel("Category of Country Risk according to ICRG", fontsize=18)

axe.set_title("India's Country Risk", loc='center', fontsize=18, fontweight =
"bold" )
```

Out[810]: Text(0.5, 1.0, "India's Country Risk")



In [811]:
```
IndiaML['Category of Risk according to ICRG']=IndiaML['Category of Risk accord
ing to ICRG'].cat.codes
```

In [812]:
```
IndiaML=IndiaML.rename(columns={'Category of Risk according to ICRG':'CountryRisk_CategoryICRG'})
IndiaML
```

Out[812]:

	Year	CountryRisk_CategoryICRG	Country Risk	GDP	Impact
2	1985	0	39.085	237.618	High
3	1986	2	42.415	252.751	High
4	1987	2	40.6	283.75	High
5	1988	0	37.895	299.645	High
6	1989	0	36.665	300.187	High
7	1990	1	34.455	326.608	High
8	1991	1	31.04	274.842	High
9	1992	0	37.435	293.262	High
10	1993	2	42.73	284.194	High
11	1994	2	49.185	333.014	High
12	1995	2	49.935	366.6	High
13	1996	2	49.605	399.791	High
14	1997	2	50.25	423.189	High
15	1998	2	46.81	428.767	High
16	1999	2	44.035	466.841	High
17	2000	2	44.25	476.636	High
18	2001	2	44.935	493.934	High
19	2002	2	45.21	523.768	High
20	2003	2	46.48	618.369	Low
21	2004	2	49.42	721.589	Low
22	2005	2	49.605	834.218	Low
23	2006	2	49.54	949.118	Low
24	2007	2	48.98	1238.7	High
25	2008	2	46.705	1224.1	High
26	2009	2	48.335	1365.37	High
27	2010	2	47.48	1708.46	High
28	2011	2	45.855	1823.05	High
29	2012	2	45.31	1827.64	High
30	2013	2	45.685	1856.72	High
31	2014	2	46.75	2039.13	High
32	2015	2	48.395	2103.59	High
33	2016	2	49.19	2289.75	High

In [813]:
```
IndiaML=pd.get_dummies(IndiaML, columns=['Impact'])
```

In [814]:
```
IndiaML
```

Out[814]:

	Year	CountryRisk_CategoryICRG	Country Risk	GDP	Impact_High	Impact_Low
2	1985	0	39.085	237.618	1	0
3	1986	2	42.415	252.751	1	0
4	1987	2	40.6	283.75	1	0
5	1988	0	37.895	299.645	1	0
6	1989	0	36.665	300.187	1	0
7	1990	1	34.455	326.608	1	0
8	1991	1	31.04	274.842	1	0
9	1992	0	37.435	293.262	1	0
10	1993	2	42.73	284.194	1	0
11	1994	2	49.185	333.014	1	0
12	1995	2	49.935	366.6	1	0
13	1996	2	49.605	399.791	1	0
14	1997	2	50.25	423.189	1	0
15	1998	2	46.81	428.767	1	0
16	1999	2	44.035	466.841	1	0
17	2000	2	44.25	476.636	1	0
18	2001	2	44.935	493.934	1	0
19	2002	2	45.21	523.768	1	0
20	2003	2	46.48	618.369	0	1
21	2004	2	49.42	721.589	0	1
22	2005	2	49.605	834.218	0	1
23	2006	2	49.54	949.118	0	1
24	2007	2	48.98	1238.7	1	0
25	2008	2	46.705	1224.1	1	0
26	2009	2	48.335	1365.37	1	0
27	2010	2	47.48	1708.46	1	0
28	2011	2	45.855	1823.05	1	0
29	2012	2	45.31	1827.64	1	0
30	2013	2	45.685	1856.72	1	0
31	2014	2	46.75	2039.13	1	0
32	2015	2	48.395	2103.59	1	0
33	2016	2	49.19	2289.75	1	0

In [815]:
```
IndiaML.Impact_High.value_counts()
```

Out[815]:
```
1    28
0     4
Name: Impact_High, dtype: int64
```

In [816]:
```
from patsy import dmatrices
y,X = dmatrices('Impact_High ~ CountryRisk_CategoryICRG',data=IndiaML)
```

In [817]:
```
y=np.ravel(y)
```

In [818]:
```
from sklearn.linear_model import LogisticRegression as logit
logit(solver='lbfgs').fit(X,y).score(X,y)
```

Out[818]:
```
array(0.875)
```

In [819]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(logit(solver='lbfgs'),X,y,cv=5,scoring='accuracy').mean()
```

```
C:\Users\ishaa\Anaconda3NEW1\lib\site-packages\sklearn\model_selection\_spli
t.py:652: Warning: The least populated class in y has only 4 members, which i
s too few. The minimum number of members in any class cannot be less than n_s
plits=5.
  % (min_groups, self.n_splits)), Warning)
```

Out[819]:
```
0.880952380952381
```

In [820]:
```
cross_val_score(logit(solver='lbfgs'),X,y,cv=3,scoring='roc_auc').mean()
```

Out[820]:
```
0.6
```

In [821]:
```
from sklearn.neighbors import KNeighborsClassifier as knn
cross_val_score(knn(),X,y,cv=3,scoring='accuracy').mean()
```

Out[821]:
```
0.8777777777777778
```

In [822]:
```
from sklearn.ensemble import RandomForestClassifier as rf
cross_val_score(rf(n_estimators=100, max_depth=2,max_features=2),X,y,cv=2,scor
ing='accuracy').mean()
```

Out[822]:
```
0.875
```

In [823]:
```
for i in range(1,10):
    for j in range(1,3):
        print(i,j,cross_val_score(rf(n_estimators=100,max_depth=i,max_features
=j),X,y,cv=3,scoring='accuracy').mean())
```

```
1 1 0.8777777777777778
1 2 0.8777777777777778
2 1 0.8777777777777778
2 2 0.8777777777777778
3 1 0.8777777777777778
3 2 0.8777777777777778
4 1 0.8777777777777778
4 2 0.8777777777777778
5 1 0.8777777777777778
5 2 0.8777777777777778
6 1 0.8777777777777778
6 2 0.8777777777777778
7 1 0.8777777777777778
7 2 0.8777777777777778
8 1 0.8777777777777778
8 2 0.8777777777777778
9 1 0.8777777777777778
9 2 0.8777777777777778
```

In [824]:
```
from sklearn.linear_model import LogisticRegression as reg
from sklearn.model_selection import train_test_split
```

In [825]:
```
x=IndiaML.drop(['Impact_High'],axis=1)
y=IndiaML['Impact_High']
```

In [826]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.2)
```

In [827]:
```
result = reg(solver='liblinear').fit(x_train,y_train).predict(x_test)
```

In [828]:
```
pd.Series(result).value_counts()
```

Out[828]:
```
1    7
dtype: int64
```

This Model shows that in the next seven years from 2017 to 2023 Country Risk will have a 100% chance of highly impacting GDP.

Argentina Machine Learning Classification: Sovereign Risk's impact on GDP

In [829]:
```
ArgentinaML=pd.read_excel (r"C:\New folder\Argentina Sov Risk Data.xlsx")
ArgentinaML=ArgentinaML.drop(0)
ArgentinaML=ArgentinaML.drop(33)
ArgentinaML=ArgentinaML.rename(columns={'Unnamed: 0':'Year','Unnamed: 1':'Category of Risk according to ICRG','Unnamed: 2':'Sovereign Risk','Unnamed: 3':'GDP','Unnamed: 4':'GDPImpact'})
ArgentinaML
```

Out[829]:

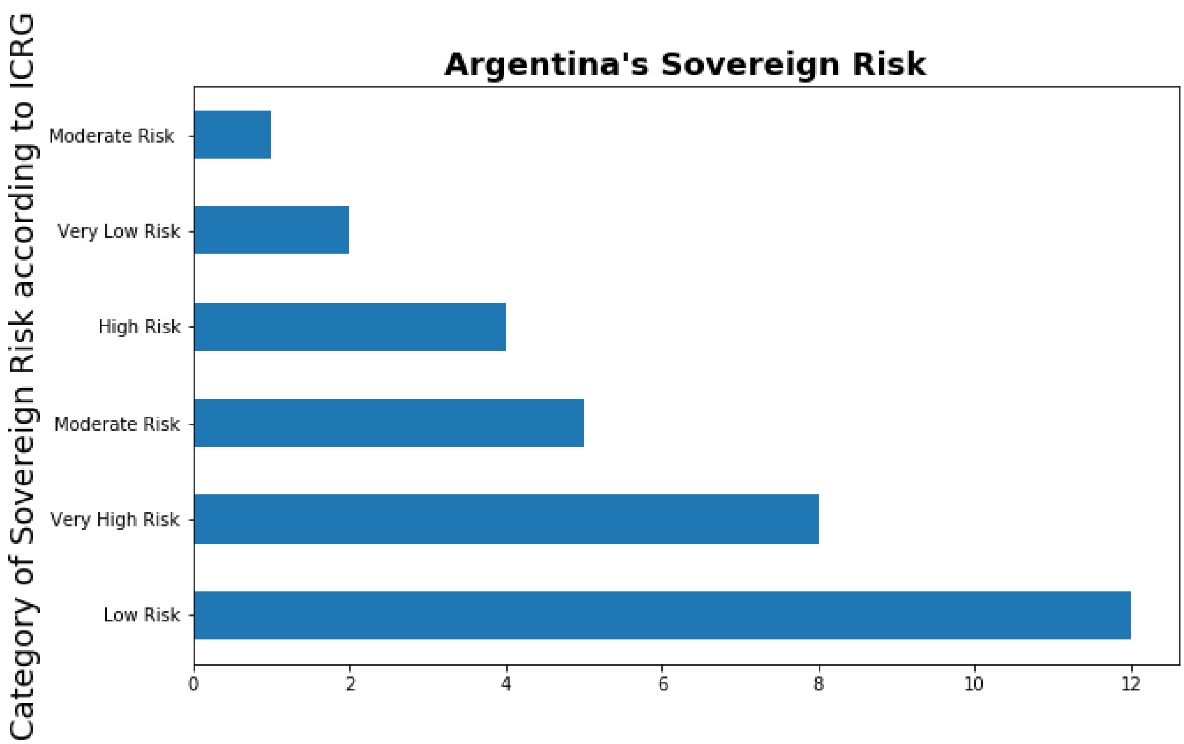
	Year	Category of Risk according to ICRG	Sovereign Risk	GDP	GDPImpact
1	1985	Very High Risk	9	95.593	High
2	1986	Very High Risk	10.75	114.949	High
3	1987	Very High Risk	14.75	117.854	High
4	1988	Very High Risk	17.83	138.044	High
5	1989	Very High Risk	17.75	88.567	High
6	1990	Very High Risk	21	153.205	High
7	1991	High Risk	28.17	205.515	Low
8	1992	Low Risk	36	247.987	Low
9	1993	Low Risk	37.75	256.365	Low
10	1994	Low Risk	38	279.15	Low
11	1995	Low Risk	35.83	280.08	Low
12	1996	Low Risk	36	295.12	Low
13	1997	Low Risk	34.83	317.549	Low
14	1998	Moderate Risk	33.54	324.242	Low
15	1999	Moderate Risk	33.46	307.673	Low
16	2000	High Risk	28.62	308.491	Low
17	2001	High Risk	28.79	291.738	Low
18	2002	Very High Risk	17.71	108.731	High
19	2003	Very High Risk	23.92	138.151	High
20	2004	High Risk	29.42	164.922	High
21	2005	Moderate Risk	32.12	199.273	Low
22	2006	Low Risk	37.58	232.892	Low
23	2007	Low Risk	37.71	287.921	Low
24	2008	Low Risk	37.33	363.545	Low
25	2009	Moderate Risk	34.67	334.633	Low
26	2010	Low Risk	39.12	424.728	High
27	2011	Very Low Risk	41.17	527.644	High
28	2012	Very Low Risk	40.17	579.666	High
29	2013	Low Risk	37.83	611.471	High
30	2014	Moderate Risk	34.08	563.614	High
31	2015	Low Risk	38.21	642.464	High
32	2016	Moderate Risk	32.62	556.774	High

In [830]:
```
ArgentinaML['Category of Risk according to ICRG']=ArgentinaML['Category of Ris
k according to ICRG'].astype('category')
ArgentinaML.dtypes
```

Out[830]:
```
Year                                      object
Category of Risk according to ICRG      category
Sovereign Risk                            object
GDP                                       object
GDPImpact                                 object
dtype: object
```

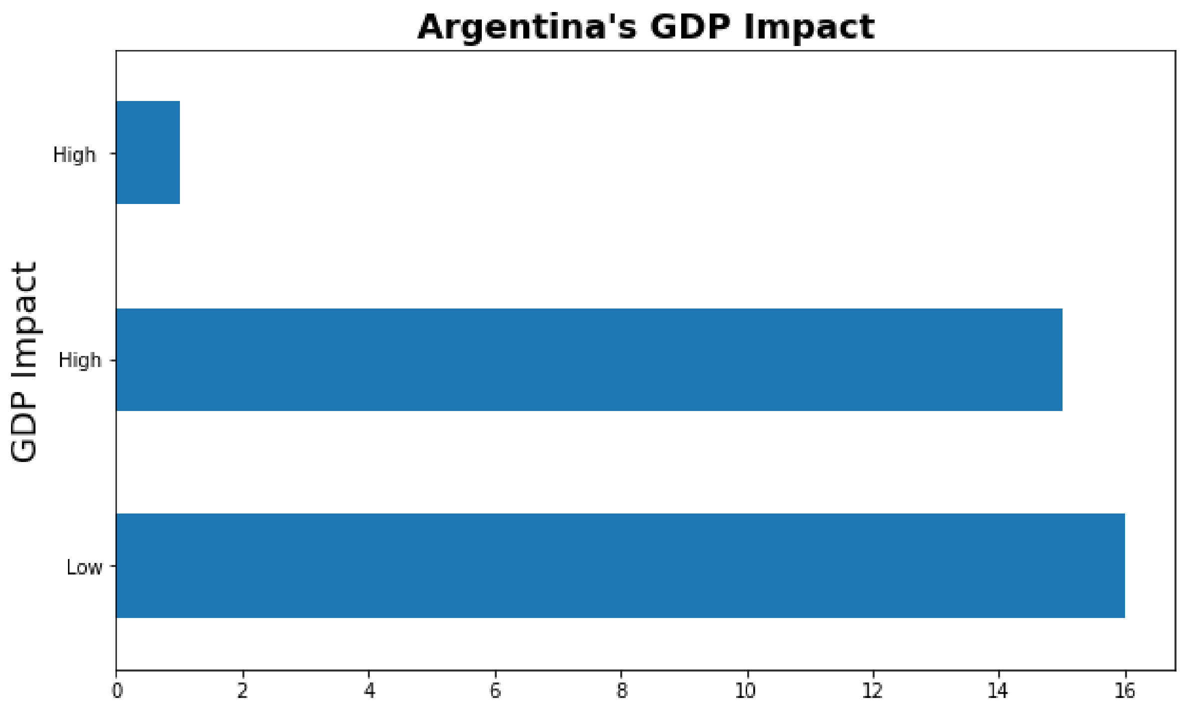
In [831]:
```
fig, axe = plt.subplots()
ArgentinaML['Category of Risk according to ICRG'].value_counts().plot.barh(fig
size =(10,6))

axe.set_ylabel("Category of Sovereign Risk according to ICRG", fontsize=18)

axe.set_title("Argentina's Sovereign Risk", loc='center', fontsize=18, fontwei
ght = "bold" )
```

Out[831]:
```
Text(0.5, 1.0, "Argentina's Sovereign Risk")
```



In [832]:
```
fig, axe = plt.subplots()
ArgentinaML['GDPImpact'].value_counts().plot.barh(figsize =(10,6))

axe.set_ylabel("GDP Impact", fontsize=18)

axe.set_title("Argentina's GDP Impact", loc='center', fontsize=18, fontweight
= "bold" )
```

Out[832]: Text(0.5, 1.0, "Argentina's GDP Impact")



In [833]:
```
ArgentinaML['Category of Risk according to ICRG']=ArgentinaML['Category of Ris
k according to ICRG'].cat.codes
```

In [834]:
```
ArgentinaML=pd.get_dummies(ArgentinaML, columns=['GDPImpact'])
```

In [835]:
```
ArgentinaML=ArgentinaML.rename(columns={'Category of Risk according to ICRG':
'Risk_CategoryICRG'})
ArgentinaML
```

Out[835]:

	Year	Risk_CategoryICRG	Sovereign Risk	GDP	GDPImpact_High	GDPImpact_High	GDPImpa
1	1985	4	9	95.593	1	0	
2	1986	4	10.75	114.949	1	0	
3	1987	4	14.75	117.854	1	0	
4	1988	4	17.83	138.044	1	0	
5	1989	4	17.75	88.567	1	0	
6	1990	4	21	153.205	1	0	
7	1991	0	28.17	205.515	0	0	
8	1992	1	36	247.987	0	0	
9	1993	1	37.75	256.365	0	0	
10	1994	1	38	279.15	0	0	
11	1995	1	35.83	280.08	0	0	
12	1996	1	36	295.12	0	0	
13	1997	1	34.83	317.549	0	0	
14	1998	2	33.54	324.242	0	0	
15	1999	2	33.46	307.673	0	0	
16	2000	0	28.62	308.491	0	0	
17	2001	0	28.79	291.738	0	0	
18	2002	4	17.71	108.731	1	0	
19	2003	4	23.92	138.151	1	0	
20	2004	0	29.42	164.922	1	0	
21	2005	2	32.12	199.273	0	0	
22	2006	1	37.58	232.892	0	0	
23	2007	1	37.71	287.921	0	0	
24	2008	1	37.33	363.545	0	0	
25	2009	2	34.67	334.633	0	0	
26	2010	1	39.12	424.728	1	0	
27	2011	5	41.17	527.644	0	1	
28	2012	5	40.17	579.666	1	0	
29	2013	1	37.83	611.471	1	0	
30	2014	2	34.08	563.614	1	0	
31	2015	1	38.21	642.464	1	0	
32	2016	3	32.62	556.774	1	0	

In [836]:
```
ArgentinaML.GDPImpact_High.value_counts()
```
Out[836]:
```
0    17
1    15
Name: GDPImpact_High, dtype: int64
```

In [837]:
```
ArgentinaML.columns
```
Out[837]:
```
Index(['Year', 'Risk_CategoryICRG', 'Sovereign Risk', 'GDP', 'GDPImpact_Hig
h',
       'GDPImpact_High ', 'GDPImpact_Low'],
      dtype='object')
```

In [838]:
```
from patsy import dmatrices
y,X = dmatrices('GDPImpact_High ~ Risk_CategoryICRG',data=ArgentinaML)
```

In [839]:
```
y=np.ravel(y)
```

In [840]:
```
from sklearn.linear_model import LogisticRegression as logit
logit(solver='lbfgs').fit(X,y).score(X,y)
```
Out[840]:
```
array(0.8125)
```

In [841]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(logit(solver='lbfgs'),X,y,cv=5,scoring='accuracy').mean()
```
Out[841]:
```
0.8
```

In [842]:
```
cross_val_score(logit(solver='lbfgs'),X,y,cv=3,scoring='roc_auc').mean()
```
Out[842]:
```
0.7722222222222223
```

In [843]:
```
from sklearn.neighbors import KNeighborsClassifier as knn
cross_val_score(knn(),X,y,cv=3,scoring='accuracy').mean()
```
Out[843]:
```
0.8060606060606061
```

In [844]:
```
from sklearn.ensemble import RandomForestClassifier as rf
cross_val_score(rf(n_estimators=100, max_depth=2,max_features=2),X,y,cv=2,scor
ing='accuracy').mean()
```
Out[844]:
```
0.7666666666666666
```

In [845]:
```
for i in range(1,10):
    for j in range(1,3):
        print(i,j,cross_val_score(rf(n_estimators=100,max_depth=i,max_features
=j),X,y,cv=3,scoring='accuracy').mean())
```

```
1 1 0.7727272727272728
1 2 0.7727272727272728
2 1 0.6818181818181818
2 2 0.7727272727272728
3 1 0.7727272727272728
3 2 0.7727272727272728
4 1 0.7727272727272728
4 2 0.7727272727272728
5 1 0.7727272727272728
5 2 0.7727272727272728
6 1 0.7727272727272728
6 2 0.7727272727272728
7 1 0.7727272727272728
7 2 0.7727272727272728
8 1 0.6818181818181818
8 2 0.7727272727272728
9 1 0.7727272727272728
9 2 0.6818181818181818
```

In [846]:
```
from sklearn.linear_model import LogisticRegression as reg
from sklearn.model_selection import train_test_split
```

In [847]:
```
x=ArgentinaML.drop(['GDPImpact_High'],axis=1)
y=ArgentinaML['GDPImpact_High']
```

In [848]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.2)
```

In [849]:
```
result = reg(solver='liblinear').fit(x_train,y_train).predict(x_test)
```

In [850]:
```
pd.Series(result).value_counts()
```

Out[850]:
```
0    4
1    3
dtype: int64
```

This Model shows that in the next seven years from 2017 to 2023, Sovereign Risk will have a 42.7 % chance of highly impacting GDP.

EM GDP Machine Learning Classification: Sovereign Risk Impacting GDP

In [851]:
```
EMGDPML=pd.read_excel (r"C:\New folder\EM GDP Data.xlsx")
EMGDPML=EMGDPML.drop(0)
EMGDPML=EMGDPML.rename(columns={'Unnamed: 0':'Year','Unnamed: 1':'Category of Risk according to ICRG','Unnamed: 2':'Sovereign Risk','Unnamed: 3':'GDP','Unnamed: 4':'GDPImpact'})
EMGDPML
```

Out[851]:

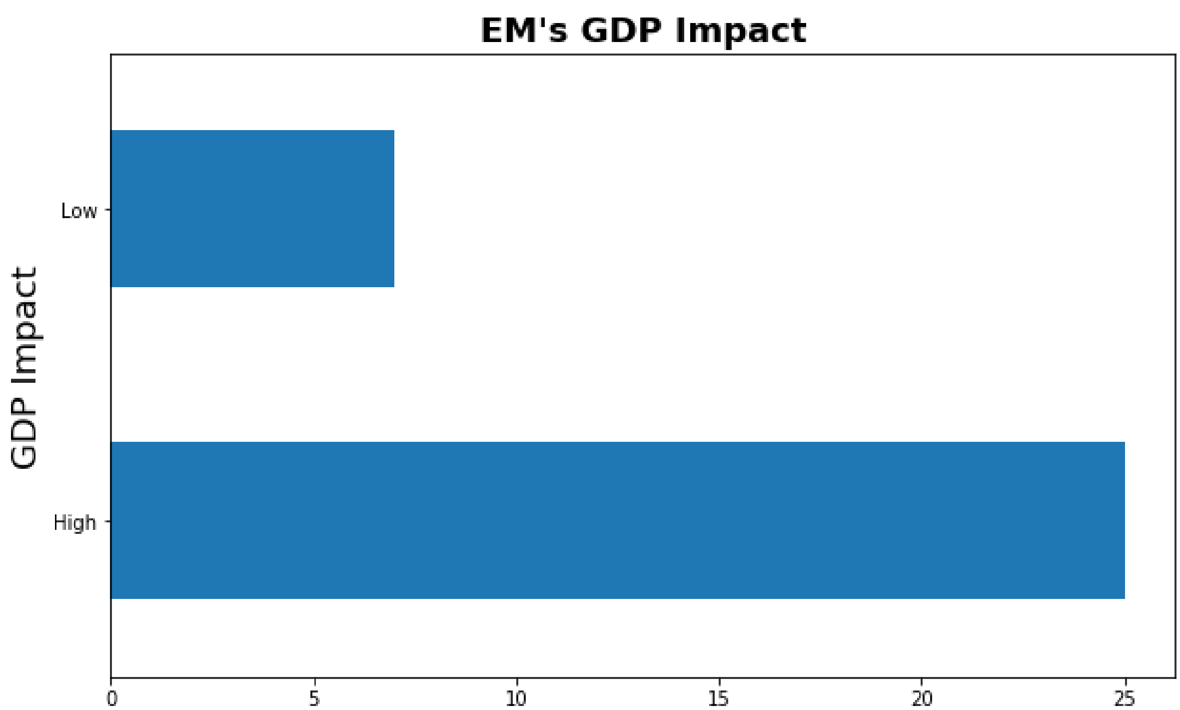
	Year	Category of Risk according to ICRG	Sovereign Risk	GDP	GDPImpact
1	1985	Very High Risk	24.96	91.49	High
2	1986	Very High Risk	23.64	91.53	High
3	1987	High Risk	24.13	100.36	High
4	1988	High Risk	25.85	113.91	High
5	1989	High Risk	27.31	126.86	High
6	1990	High Risk	29.33	134.15	High
7	1991	Moderate Risk	32.4	137.78	High
8	1992	Low Risk	36.59	149.26	High
9	1993	Low Risk	37.91	171.26	High
10	1994	Low Risk	38.23	190.53	High
11	1995	Low Risk	38.07	216.39	High
12	1996	Low Risk	38.73	241.07	High
13	1997	Low Risk	37.63	254.93	High
14	1998	Low Risk	34.8	239.29	High
15	1999	Low Risk	35.47	233.89	High
16	2000	Low Risk	36.06	255.18	High
17	2001	Low Risk	36.52	260.65	Low
18	2002	Low Risk	36.7	263.93	Low
19	2003	Low Risk	37.4	295.3	Low
20	2004	Low Risk	38.85	348.16	Low
21	2005	Low Risk	39.61	414.25	Low
22	2006	Very Low Risk	40.58	491.49	Low
23	2007	Very Low Risk	40.72	607.78	Low
24	2008	Very Low Risk	39.83	726.82	High
25	2009	Low Risk	39	711.14	High
26	2010	Very Low Risk	41.26	867.39	High
27	2011	Very Low Risk	41.34	1032.07	High
28	2012	Very Low Risk	41.08	1100.41	High
29	2013	Very Low Risk	41.07	1171.2	High
30	2014	Very Low Risk	40.61	1213.92	High
31	2015	Low Risk	39.34	1161.19	High
32	2016	Low Risk	39.55	1160	High

In [852]:
```
EMGDPML['Category of Risk according to ICRG']=EMGDPML['Category of Risk accord
ing to ICRG'].astype('category')
EMGDPML.dtypes
```

Out[852]:
```
Year                                     object
Category of Risk according to ICRG     category
Sovereign Risk                           object
GDP                                      object
GDPImpact                                object
dtype: object
```

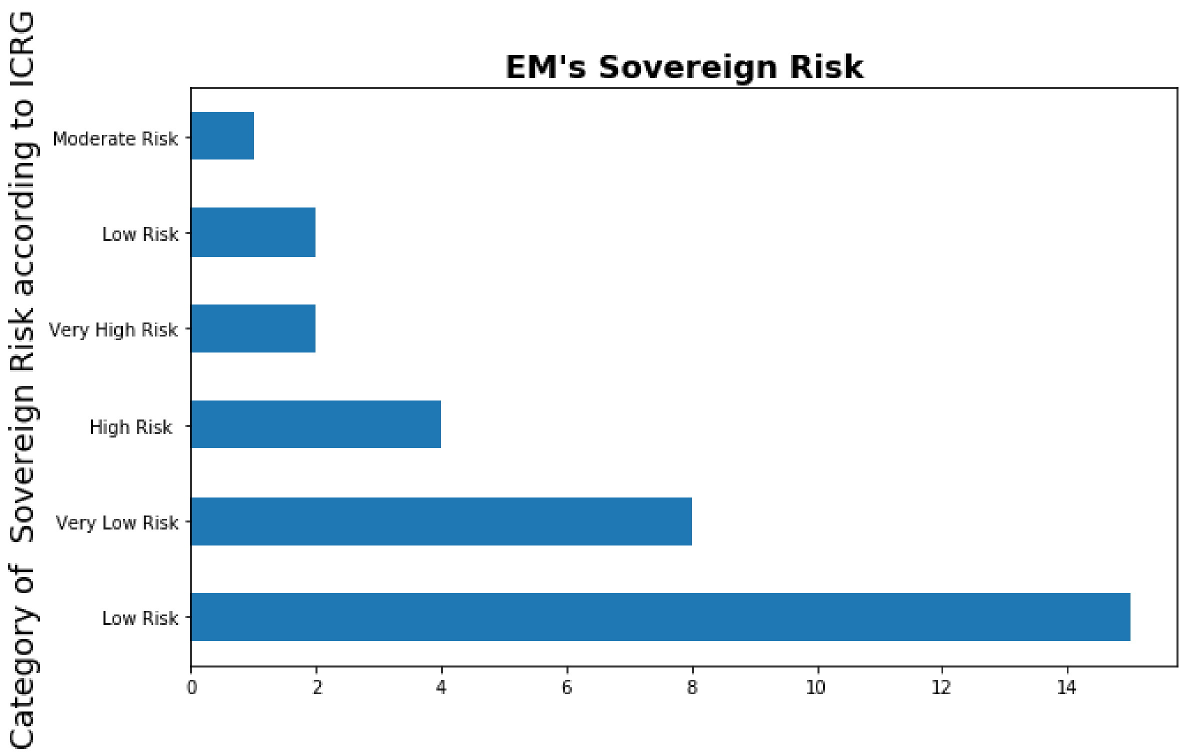
In [853]:
```
fig, axe = plt.subplots()
EMGDPML['GDPImpact'].value_counts().plot.barh(figsize =(10,6))

axe.set_ylabel("GDP Impact", fontsize=18)

axe.set_title("EM's GDP Impact", loc='center', fontsize=18, fontweight = "bol
d" )
```

Out[853]:
```
Text(0.5, 1.0, "EM's GDP Impact")
```



In [854]:
```
fig, axe = plt.subplots()


EMGDPML['Category of Risk according to ICRG'].value_counts().plot.barh(figsize
=(10,6))

axe.set_ylabel("Category of  Sovereign Risk according to ICRG", fontsize=18)

axe.set_title("EM's Sovereign Risk", loc='center', fontsize=18, fontweight =
"bold" )
```

Out[854]: Text(0.5, 1.0, "EM's Sovereign Risk")



In [855]:
```
EMGDPML['Category of Risk according to ICRG']=EMGDPML['Category of Risk accord
ing to ICRG'].cat.codes
```

In [856]:
```
EMGDPML=pd.get_dummies(EMGDPML, columns=['GDPImpact'])
```

In [857]:
```
EMGDPML=EMGDPML.rename(columns={'Category of Risk according to ICRG':'Risk_CategoryICRG'})
EMGDPML
```

Out[857]:

	Year	Risk_CategoryICRG	Sovereign Risk	GDP	GDPImpact_High	GDPImpact_Low
1	1985	4	24.96	91.49	1	0
2	1986	4	23.64	91.53	1	0
3	1987	1	24.13	100.36	1	0
4	1988	1	25.85	113.91	1	0
5	1989	1	27.31	126.86	1	0
6	1990	1	29.33	134.15	1	0
7	1991	3	32.4	137.78	1	0
8	1992	2	36.59	149.26	1	0
9	1993	2	37.91	171.26	1	0
10	1994	2	38.23	190.53	1	0
11	1995	2	38.07	216.39	1	0
12	1996	2	38.73	241.07	1	0
13	1997	2	37.63	254.93	1	0
14	1998	2	34.8	239.29	1	0
15	1999	2	35.47	233.89	1	0
16	2000	2	36.06	255.18	1	0
17	2001	2	36.52	260.65	0	1
18	2002	2	36.7	263.93	0	1
19	2003	2	37.4	295.3	0	1
20	2004	2	38.85	348.16	0	1
21	2005	2	39.61	414.25	0	1
22	2006	5	40.58	491.49	0	1
23	2007	5	40.72	607.78	0	1
24	2008	5	39.83	726.82	1	0
25	2009	0	39	711.14	1	0
26	2010	5	41.26	867.39	1	0
27	2011	5	41.34	1032.07	1	0
28	2012	5	41.08	1100.41	1	0
29	2013	5	41.07	1171.2	1	0
30	2014	5	40.61	1213.92	1	0
31	2015	0	39.34	1161.19	1	0
32	2016	2	39.55	1160	1	0

In [858]:
```
EMGDPML.GDPImpact_High.value_counts()
```

Out[858]:
```
1    25
0     7
Name: GDPImpact_High, dtype: int64
```

In [859]:
```
EMGDPML.columns
```

Out[859]:
```
Index(['Year', 'Risk_CategoryICRG', 'Sovereign Risk', 'GDP', 'GDPImpact_Hig
h',
       'GDPImpact_Low'],
      dtype='object')
```

In [860]:
```
from patsy import dmatrices
y,X = dmatrices('GDPImpact_High ~ Risk_CategoryICRG',data=EMGDPML)
```

In [861]:
```
y=np.ravel(y)
```

In [862]:
```
from sklearn.linear_model import LogisticRegression as logit
logit(solver='lbfgs').fit(X,y).score(X,y)
```

Out[862]:
```
array(0.78125)
```

In [863]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(logit(solver='lbfgs'),X,y,cv=5,scoring='accuracy').mean()
```

Out[863]:
```
0.7857142857142858
```

In [864]:
```
cross_val_score(logit(solver='lbfgs'),X,y,cv=3,scoring='roc_auc').mean()
```

Out[864]:
```
0.4351851851851852
```

In [865]:
```
from sklearn.neighbors import KNeighborsClassifier as knn
cross_val_score(knn(),X,y,cv=3,scoring='accuracy').mean()
```

Out[865]:
```
0.6166666666666667
```

In [866]:
```
from sklearn.ensemble import RandomForestClassifier as rf
cross_val_score(rf(n_estimators=100, max_depth=2,max_features=2),X,y,cv=2,scor
ing='accuracy').mean()
```

Out[866]:
```
0.7823529411764706
```

In [867]:
```
for i in range(1,10):
    for j in range(1,3):
        print(i,j,cross_val_score(rf(n_estimators=100,max_depth=i,max_features
=j),X,y,cv=3,scoring='accuracy').mean())
```

```
1 1 0.7833333333333333
1 2 0.7833333333333333
2 1 0.7833333333333333
2 2 0.7833333333333333
3 1 0.6166666666666667
3 2 0.7833333333333333
4 1 0.6166666666666667
4 2 0.7833333333333333
5 1 0.6166666666666667
5 2 0.6166666666666667
6 1 0.6166666666666667
6 2 0.6166666666666667
7 1 0.7833333333333333
7 2 0.6166666666666667
8 1 0.7833333333333333
8 2 0.6166666666666667
9 1 0.7833333333333333
9 2 0.7833333333333333
```

In [868]:
```
from sklearn.linear_model import LogisticRegression as reg
from sklearn.model_selection import train_test_split
```

In [869]:
```
x=EMGDPML.drop(['GDPImpact_High'],axis=1)
y=EMGDPML['GDPImpact_High']
```

In [870]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.2)
```

In [871]:
```
result = reg(solver='liblinear').fit(x_train,y_train).predict(x_test)
```

In [872]:
```
pd.Series(result).value_counts()
```

Out[872]:
```
1    6
0    1
dtype: int64
```

This Model shows that EM's Sovereign Risk will have a 85% chance to highly impact EM's GDP.

Commodity Price Sovereign Risk Machine Learning Classification: Sovereign Risk Impacting Commodity Prices

In [873]:
```
x=PriceRiskXIM1[['Sovereign Risk', 'Political Risk', 'Country Risk','ImportsIE','ExportsEM','ImportsEM']]
y=GDPvPrices['Commodity Prices']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.97
6
Model:                            OLS   Adj. R-squared:                  0.97
1
Method:                 Least Squares   F-statistic:                     171.
5
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           4.58e-1
9
Time:                        22:42:49   Log-Likelihood:                -117.6
5
No. Observations:                  32   AIC:                             249.
3
Df Residuals:                      25   BIC:                             259.
6
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
-----
const           3052.8480    744.430      4.101      0.000    1519.665    458
6.031
Sovereign Risk     4.5429      1.174      3.870      0.001       2.125       
6.961
Political Risk     1.0138      2.841      0.357      0.724      -4.838       
6.866
Country Risk      -5.7851      4.279     -1.352      0.188     -14.597       
3.027
ImportsIE        -47.3847     21.891     -2.165      0.040     -92.471      -
2.299
ExportsEM         14.4414     16.024      0.901      0.376     -18.560      4
7.443
ImportsEM          5.5045      0.736      7.480      0.000       3.989       
7.020
==============================================================================
=
Omnibus:                        0.112   Durbin-Watson:                   1.42
9
Prob(Omnibus):                  0.946   Jarque-Bera (JB):                0.22
6
Skew:                          -0.125   Prob(JB):                        0.89
3
Kurtosis:                       2.673   Cond. No.                     6.72e+0
4
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
[2] The condition number is large, 6.72e+04. This might indicate that there a
```

```
re
strong multicollinearity or other numerical problems.
```

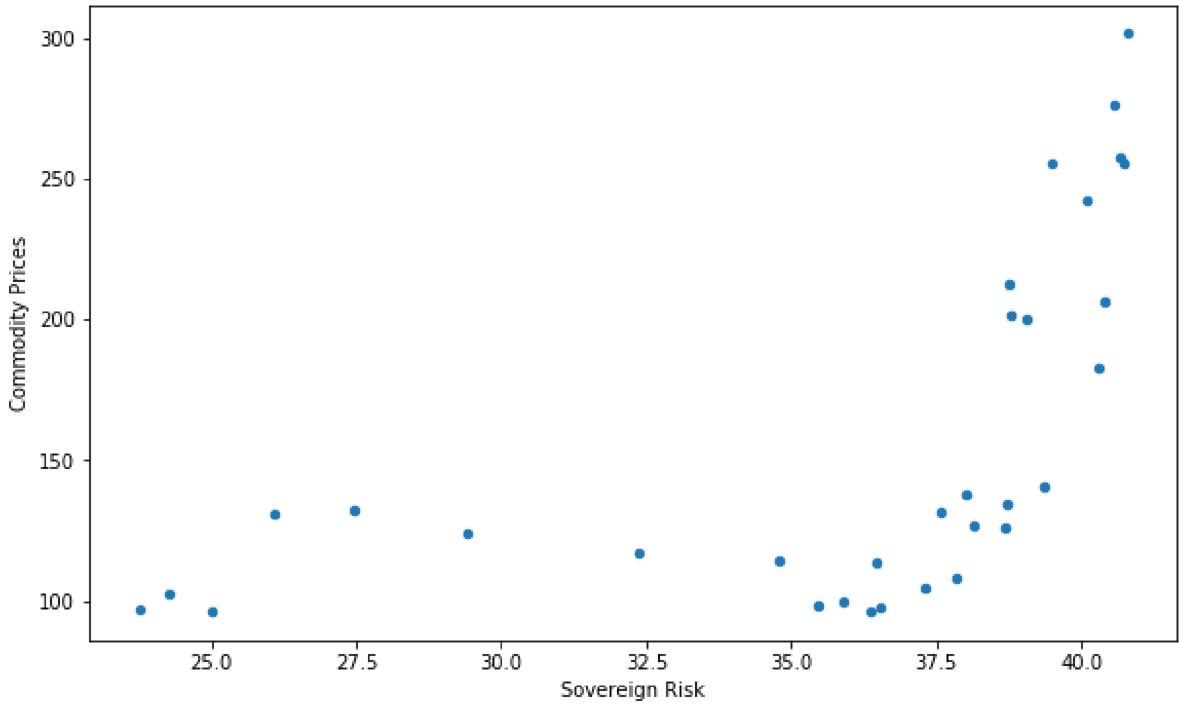
```
C:\Users\ishaa\Anaconda3NEW1\lib\site-packages\numpy\core\fromnumeric.py:238
9: FutureWarning: Method .ptp is deprecated and will be removed in a future v
ersion. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

In [874]:
```
y=GDPvPrices['Commodity Prices']
x=PriceRiskXIM1['Sovereign Risk']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:       Commodity Prices   R-squared:                       0.35
8
Model:                            OLS   Adj. R-squared:                  0.33
7
Method:                 Least Squares   F-statistic:                     16.7
3
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           0.00029
8
Time:                        22:42:49   Log-Likelihood:                -170.4
2
No. Observations:                  32   AIC:                             344.
8
Df Residuals:                      30   BIC:                             347.
8
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
=====
                     coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
-----
const            -92.9277     61.775     -1.504      0.143    -219.089      3
3.234
Sovereign Risk     6.9276      1.694      4.091      0.000       3.469      1
0.386
==============================================================================
=
Omnibus:                        4.312   Durbin-Watson:                   0.19
5
Prob(Omnibus):                  0.116   Jarque-Bera (JB):                2.11
5
Skew:                           0.337   Prob(JB):                        0.34
7
Kurtosis:                       1.936   Cond. No.                         24
8.
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [875]:
```
PriceRisk.plot.scatter(x='Sovereign Risk',y='Commodity Prices',figsize=(10,6))
PriceRisk['Sovereign Risk'].corr(PriceRisk['Commodity Prices'])
```

Out[875]: 0.5831224067321231



In [876]:
```
EMML=pd.read_excel (r"C:\New folder\EM Sovereign Risk CP Data.xlsx")
EMML=EMML.drop(0)
EMML=EMML.rename(columns={'Unnamed: 0':'Year','Unnamed: 1':'Category of Risk a
ccording to ICRG','Unnamed: 2':'Sovereign Risk','Unnamed: 3':'Commodity Prices
','Unnamed: 4':'Impact'})
EMML
```

Out[876]:

	Year	Category of Risk according to ICRG	Sovereign Risk	Commodity Prices	Impact
1	1985	Very High Risk	24.96	96.2167	High
2	1986	Very High Risk	23.64	96.675	High
3	1987	High Risk	24.13	102.208	High
4	1988	High Risk	25.85	131.117	Low
5	1989	High Risk	27.31	132.325	Low
6	1990	High Risk	29.33	124.025	High
7	1991	Moderate Risk	32.4	117.042	High
8	1992	Low Risk	36.59	113.858	High
9	1993	Low Risk	37.91	108.258	High
10	1994	Low Risk	38.23	126.508	Low
11	1995	Low Risk	38.07	137.575	Low
12	1996	Low Risk	38.73	134.35	Low
13	1997	Low Risk	37.63	131.667	Low
14	1998	Low Risk	34.8	114.4	High
15	1999	Low Risk	35.47	98.4833	High
16	2000	Low Risk	36.06	99.9917	High
17	2001	Low Risk	36.52	96.375	High
18	2002	Low Risk	36.7	97.3167	High
19	2003	Low Risk	37.4	104.858	High
20	2004	Low Risk	38.85	125.783	High
21	2005	Low Risk	39.61	140.392	Low
22	2006	Very Low Risk	40.58	182.825	Low
23	2007	Very Low Risk	40.72	206.525	High
24	2008	Very Low Risk	39.83	256.033	High
25	2009	Low Risk	39	212.742	High
26	2010	Very Low Risk	41.26	256.042	High
27	2011	Very Low Risk	41.34	302	High
28	2012	Very Low Risk	41.08	276.783	High
29	2013	Very Low Risk	41.07	258.183	High
30	2014	Very Low Risk	40.61	242.508	High
31	2015	Low Risk	39.34	201.575	High
32	2016	Low Risk	39.55	200.083	High

In [877]:
```
EMML.dtypes
```

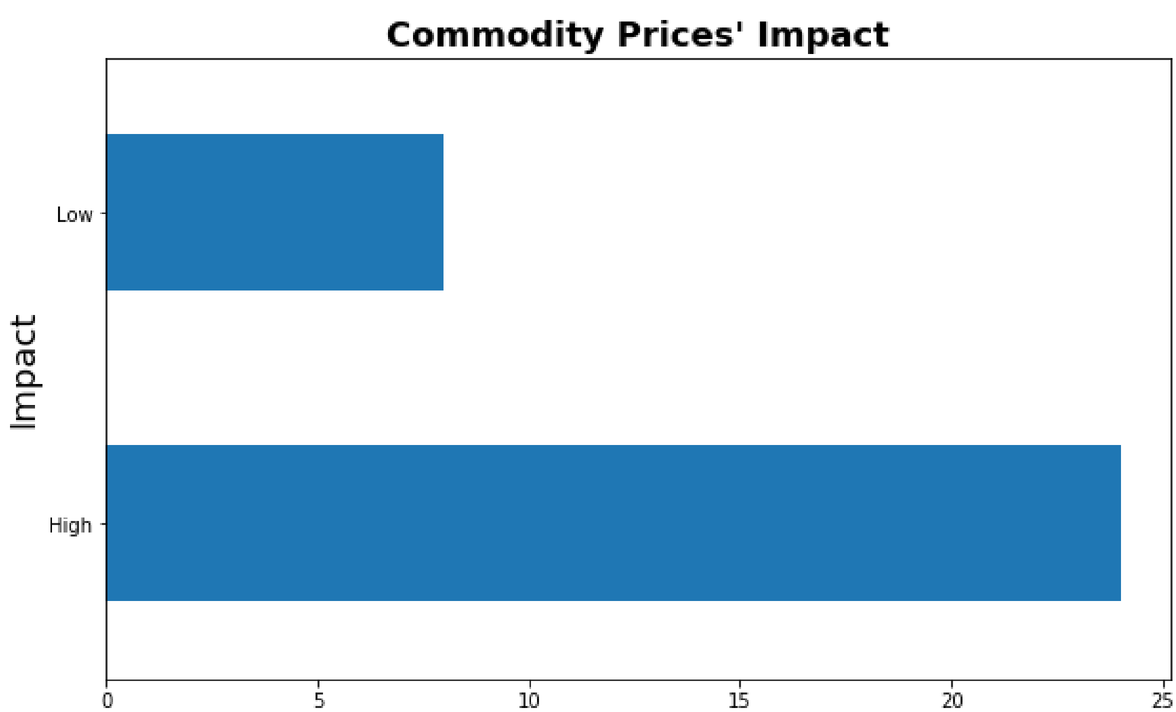
Out[877]:
```
Year                                   object
Category of Risk according to ICRG     object
Sovereign Risk                         object
Commodity Prices                       object
Impact                                 object
dtype: object
```

In [878]:
```
EMML['Category of Risk according to ICRG']=EMML['Category of Risk according to ICRG'].astype('category')
EMML.dtypes
```

Out[878]:
```
Year                                    object
Category of Risk according to ICRG    category
Sovereign Risk                          object
Commodity Prices                        object
Impact                                  object
dtype: object
```

In [879]:
```
fig, axe = plt.subplots()
EMML['Impact'].value_counts().plot.barh(figsize =(10,6))

axe.set_ylabel("Impact", fontsize=18)

axe.set_title("Commodity Prices' Impact", loc='center', fontsize=18, fontweight = "bold" )
```

Out[879]:
```
Text(0.5, 1.0, "Commodity Prices' Impact")
```



In [880]:
```
EMML['Category of Risk according to ICRG']=EMML['Category of Risk according to ICRG'].cat.codes
```

In [881]:
```
EMML=pd.get_dummies(EMML, columns=['Impact'])
```

In [882]:
```
EMML=EMML.rename(columns={'Category of Risk according to ICRG':'Risk_CategoryICRG'})
EMML
```

Out[882]:

	Year	Risk_CategoryICRG	Sovereign Risk	Commodity Prices	Impact_High	Impact_Low
1	1985	4	24.96	96.2167	1	0
2	1986	4	23.64	96.675	1	0
3	1987	1	24.13	102.208	1	0
4	1988	1	25.85	131.117	0	1
5	1989	1	27.31	132.325	0	1
6	1990	1	29.33	124.025	1	0
7	1991	3	32.4	117.042	1	0
8	1992	2	36.59	113.858	1	0
9	1993	2	37.91	108.258	1	0
10	1994	2	38.23	126.508	0	1
11	1995	2	38.07	137.575	0	1
12	1996	2	38.73	134.35	0	1
13	1997	2	37.63	131.667	0	1
14	1998	2	34.8	114.4	1	0
15	1999	2	35.47	98.4833	1	0
16	2000	2	36.06	99.9917	1	0
17	2001	2	36.52	96.375	1	0
18	2002	2	36.7	97.3167	1	0
19	2003	2	37.4	104.858	1	0
20	2004	2	38.85	125.783	1	0
21	2005	2	39.61	140.392	0	1
22	2006	5	40.58	182.825	0	1
23	2007	5	40.72	206.525	1	0
24	2008	5	39.83	256.033	1	0
25	2009	0	39	212.742	1	0
26	2010	5	41.26	256.042	1	0
27	2011	5	41.34	302	1	0
28	2012	5	41.08	276.783	1	0
29	2013	5	41.07	258.183	1	0
30	2014	5	40.61	242.508	1	0
31	2015	0	39.34	201.575	1	0
32	2016	2	39.55	200.083	1	0

In [883]:
```
EMML.Impact_High.value_counts()
```

Out[883]:
```
1    24
0     8
Name: Impact_High, dtype: int64
```

In [884]:
```
from patsy import dmatrices
y,X = dmatrices('Impact_High ~ Risk_CategoryICRG',data=EMML)
```

In [885]:
```
y=np.ravel(y)
```

In [886]:
```
from sklearn.linear_model import LogisticRegression as logit
logit(solver='lbfgs').fit(X,y).score(X,y)
```

Out[886]:
```
array(0.75)
```

In [887]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(logit(solver='lbfgs'),X,y,cv=5,scoring='accuracy').mean()
```

Out[887]:
```
0.7152380952380952
```

In [888]:
```
cross_val_score(logit(solver='lbfgs'),X,y,cv=3,scoring='roc_auc').mean()
```

Out[888]:
```
0.6319444444444445
```

In [889]:
```
from sklearn.neighbors import KNeighborsClassifier as knn
cross_val_score(knn(),X,y,cv=3,scoring='accuracy').mean()
```

Out[889]:
```
0.6606060606060606
```

In [890]:
```
from sklearn.ensemble import RandomForestClassifier as rf
cross_val_score(rf(n_estimators=100, max_depth=2,max_features=2),X,y,cv=2,scor
ing='accuracy').mean()
```

Out[890]:
```
0.6875
```

In [891]:
```
for i in range(1,10):
    for j in range(1,3):
        print(i,j,cross_val_score(rf(n_estimators=100,max_depth=i,max_features
=j),X,y,cv=3,scoring='accuracy').mean())
```

```
1 1 0.7515151515151516
1 2 0.7515151515151516
2 1 0.6848484848484849
2 2 0.7515151515151516
3 1 0.6848484848484849
3 2 0.7515151515151516
4 1 0.6848484848484849
4 2 0.7515151515151516
5 1 0.7515151515151516
5 2 0.7515151515151516
6 1 0.6848484848484849
6 2 0.6848484848484849
7 1 0.6848484848484849
7 2 0.6848484848484849
8 1 0.6848484848484849
8 2 0.6848484848484849
9 1 0.7515151515151516
9 2 0.7515151515151516
```

In [892]:
```
EMML['high_impact_rf'] = rf(n_estimators=100, max_depth=2,max_features=2).fit(
X,y).predict(X)
```

In [893]:
```
EMML.high_impact_rf.value_counts()
```

Out[893]:
```
1.0    32
Name: high_impact_rf, dtype: int64
```

In [894]:
```
from sklearn.linear_model import LogisticRegression as reg
from sklearn.model_selection import train_test_split
```

In [895]:
```
x=EMML.drop(['Impact_High'],axis=1)
y=EMML['Impact_High']
```

In [896]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.2)
```

In [897]:
```
result = reg(solver='liblinear').fit(x_train,y_train).predict(x_test)
```

In [898]:
```
pd.Series(result).value_counts()
```

Out[898]:
```
1    7
dtype: int64
```

This Model shows that Sovereign Risk among the EMs will have a 100% chance of highly impacting their GDPs from 2017-2023.

Lastly, for the following model below, analysis was done in Excel before being imported into Jupyter once again. Only this time, this model looks at how both Sovereign Risk's Impact on GDP and Commodity Prices' Impact on GDP affected Emerging Markets' Overall GDP. In order to do this, all of the following data was standardized. Then, for Sovereign Risk's Impact on GDP and Commodity Prices' Impact on GDP respectively, each standardized instance was multiplied by its corresponding dependent standardized instance per year to come up with the coefficient per year. From there, the correlation coefficient from all of the years were added up together, and used to find the correlation for each of the independent variables being examined. The respective correlation for its' corresponding independent variable was used as a threshold to determine if each standardized instance had a high impact or low impact on GDP. After doing these steps, Machine Learning Classification was then pursued by following the steps each of the previous models used.

Commodity Prices GDP Sovereign Risk Machine Learning Classification: Sovereign Risk and Commodity Prices impacting GDP using Standardized Data

In [899]:
```
CGDPSML=pd.read_excel (r"C:\New folder\SRCPDataStandarized.xlsx")
CGDPSML
```

Out[899]:

	Year	Sovereign Risk	Sovereign Risk Impact	Commodity Prices	Commodity Prices Impact	GDP	High or Low Impact GDP
0	1985	-0.349830	Low	-0.136560	Low	-0.646860	High
1	1986	-0.326120	Low	-0.138430	Low	-0.646770	High
2	1987	-0.324860	Low	-0.156390	Low	-0.627380	High
3	1988	-0.337940	Low	-0.259000	Low	-0.597630	High
4	1989	-0.344900	Low	-0.251050	Low	-0.569190	High
5	1990	-0.366170	Low	-0.214750	Low	-0.553190	High
6	1991	-0.407280	Low	-0.187410	Low	-0.545220	High
7	1992	-0.448840	Low	-0.168200	Low	-0.520010	High
8	1993	-0.424400	Low	-0.135750	Low	-0.471700	High
9	1994	-0.390140	Low	-0.173480	Low	-0.429390	High
10	1995	-0.336900	Low	-0.176800	Low	-0.372610	High
11	1996	-0.293730	Low	-0.144550	Low	-0.318420	High
12	1997	-0.256870	Low	-0.125810	Low	-0.287980	High
13	1998	-0.262220	Low	-0.105370	Low	-0.322320	High
14	1999	-0.278070	Low	-0.075370	Low	-0.334180	High
15	2000	-0.243880	Low	-0.067590	Low	-0.287430	High
16	2001	-0.237200	Low	-0.058420	Low	-0.275420	Low
17	2002	-0.232330	Low	-0.058500	Low	-0.268220	Low
18	2003	-0.176530	Low	-0.053050	Low	-0.199340	Low
19	2004	-0.077090	Low	-0.033260	Low	-0.083270	Low
20	2005	0.058557	Low	0.030455	Low	0.061844	Low
21	2006	0.225365	High	0.176518	Low	0.231445	Low
22	2007	0.475890	High	0.444736	High	0.486789	Low
23	2008	0.712966	High	0.919430	High	0.748171	High
24	2009	0.663737	High	0.680342	High	0.713742	High
25	2010	1.048985	High	1.298801	High	1.056829	High
26	2011	1.411043	High	2.158339	High	1.418425	High
27	2012	1.549017	High	2.134784	High	1.568483	High
28	2013	1.702048	High	2.142136	High	1.723921	High
29	2014	1.771486	High	2.077238	High	1.817723	High
30	2015	1.598742	High	1.501260	High	1.701941	High
31	2016	1.606178	High	1.482809	High	1.699328	High

In [900]:
```
CGDPSML.columns
```

Out[900]:
```
Index(['Year ', 'Sovereign Risk ', 'Sovereign Risk Impact  ',
       'Commodity Prices ', 'Commodity Prices Impact  ', 'GDP ',
       'High or Low Impact GDP '],
      dtype='object')
```

In [901]:
```
x=CGDPSML[['Commodity Prices ']]
y=CGDPSML['GDP ']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.93
6
Model:                            OLS   Adj. R-squared:                  0.93
4
Method:                 Least Squares   F-statistic:                     439.
3
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           1.81e-1
9
Time:                        22:43:03   Log-Likelihood:                 4.389
0
No. Observations:                  32   AIC:                            -4.77
8
Df Residuals:                      30   BIC:                            -1.84
6
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
========
                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
--------
const               -0.2272      0.043     -5.338      0.000      -0.314
-0.140
Commodity Prices     0.9849      0.047     20.960      0.000       0.889
1.081
==============================================================================
=
Omnibus:                        0.648   Durbin-Watson:                   0.51
1
Prob(Omnibus):                  0.723   Jarque-Bera (JB):                0.26
3
Skew:                           0.221   Prob(JB):                        0.87
7
Kurtosis:                       3.033   Cond. No.                         1.5
9
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

```
C:\Users\ishaa\Anaconda3NEW1\lib\site-packages\numpy\core\fromnumeric.py:238
9: FutureWarning: Method .ptp is deprecated and will be removed in a future v
ersion. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

In [902]:
```
x=CGDPSML[['Sovereign Risk ']]
y=CGDPSML['GDP ']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

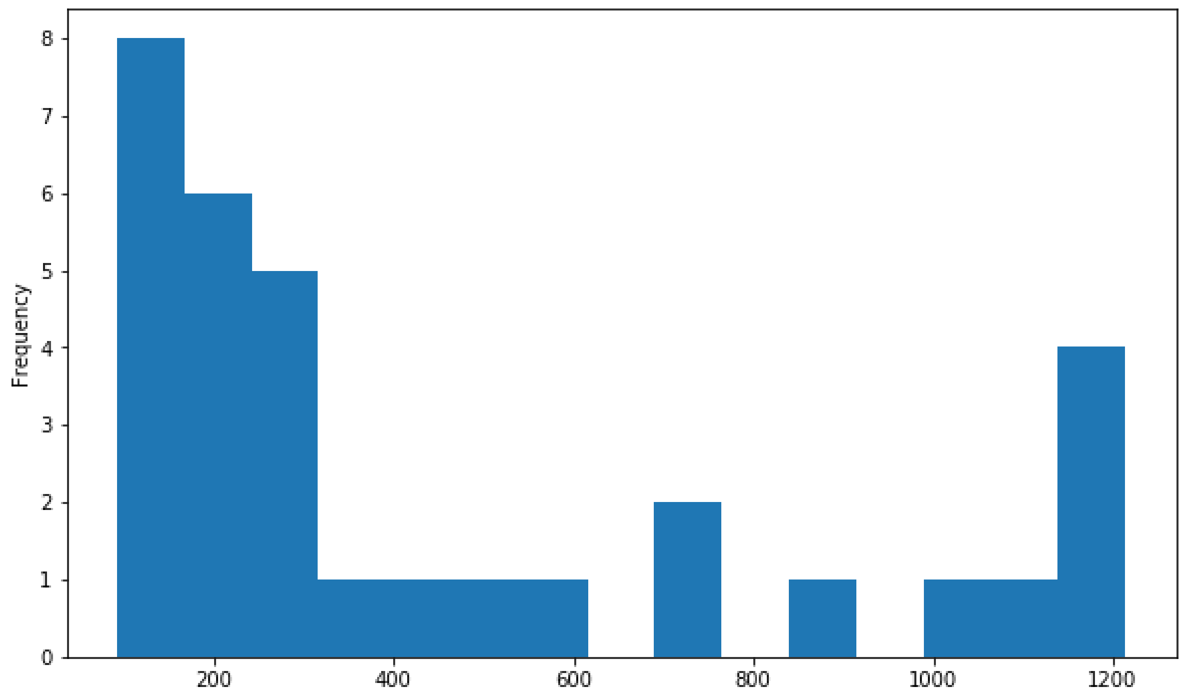
```
                            OLS Regression Results
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.98
9
Model:                            OLS   Adj. R-squared:                  0.98
9
Method:                 Least Squares   F-statistic:                     269
8.
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           6.04e-3
1
Time:                        22:43:03   Log-Likelihood:                 32.55
0
No. Observations:                  32   AIC:                            -61.1
0
Df Residuals:                      30   BIC:                            -58.1
7
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
======
                      coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
------
const              -0.0765      0.017     -4.617      0.000      -0.110
-0.043
Sovereign Risk      1.0912      0.021     51.944      0.000       1.048
1.134
==============================================================================
=
Omnibus:                        8.173   Durbin-Watson:                   0.11
9
Prob(Omnibus):                  0.017   Jarque-Bera (JB):                7.53
6
Skew:                          -1.185   Prob(JB):                       0.023
1
Kurtosis:                       3.179   Cond. No.                         1.4
4
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

In [903]:
```python
x=CGDPSML[['Sovereign Risk ','Commodity Prices ']]
y=CGDPSML['GDP ']

x=sm.add_constant(x)
est= sm.OLS(y,x).fit()
print(est.summary())
```

```
                            OLS Regression Results                            
==============================================================================
=
Dep. Variable:                    GDP   R-squared:                       0.99
0
Model:                            OLS   Adj. R-squared:                  0.98
9
Method:                 Least Squares   F-statistic:                     138
7.
Date:                Sat, 12 Oct 2019   Prob (F-statistic):           1.64e-2
9
Time:                        22:43:03   Log-Likelihood:                 33.52
3
No. Observations:                  32   AIC:                            -61.0
5
Df Residuals:                      29   BIC:                            -56.6
5
Df Model:                           2                                         
Covariance Type:            nonrobust                                         
==============================================================================
========
                       coef    std err          t      P>|t|      [0.025      
0.975]
------------------------------------------------------------------------------
--------
const               -0.0560      0.022     -2.509      0.018      -0.102      
-0.010
Sovereign Risk       1.2228      0.100     12.253      0.000       1.019      
1.427
Commodity Prices    -0.1248      0.093     -1.348      0.188      -0.314      
0.065
==============================================================================
=
Omnibus:                        9.174   Durbin-Watson:                   0.10
6
Prob(Omnibus):                  0.010   Jarque-Bera (JB):                8.44
5
Skew:                          -1.244   Prob(JB):                       0.014
7
Kurtosis:                       3.382   Cond. No.                         11.
2
==============================================================================
=

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correc
tly specified.
```

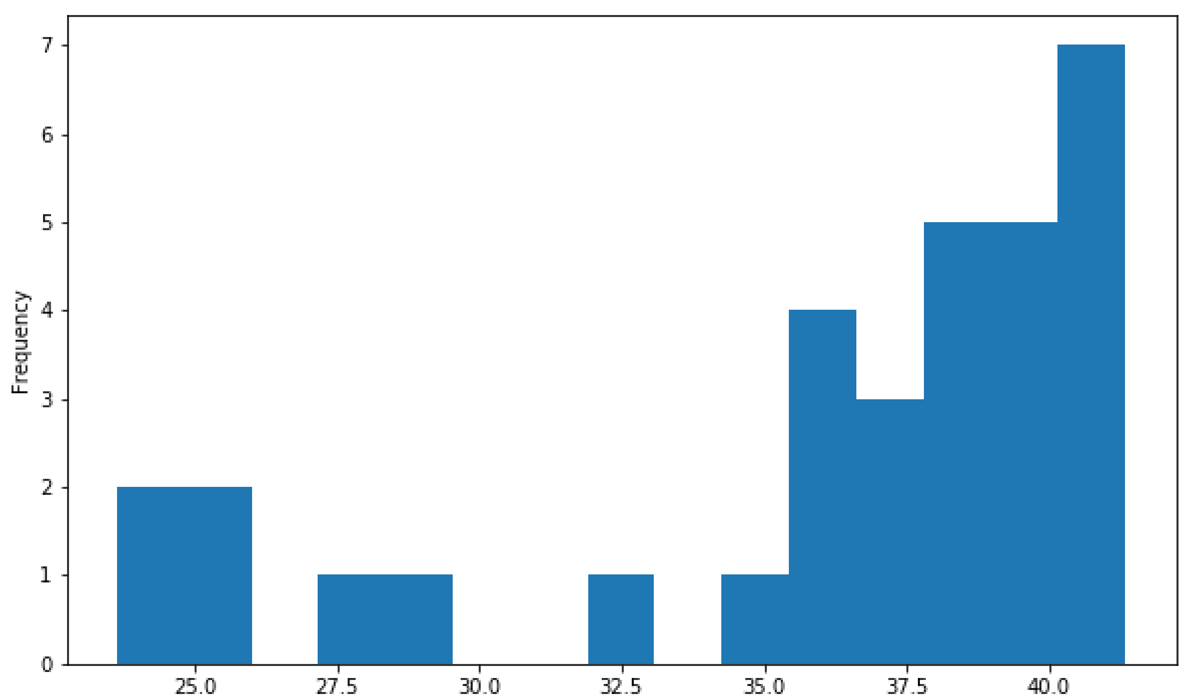
In [940]:
```
EMGDP['GDP'].plot(kind='hist',bins=15,figsize =(10,6))
```

Out[940]: <matplotlib.axes._subplots.AxesSubplot at 0x24c94aa6c88>



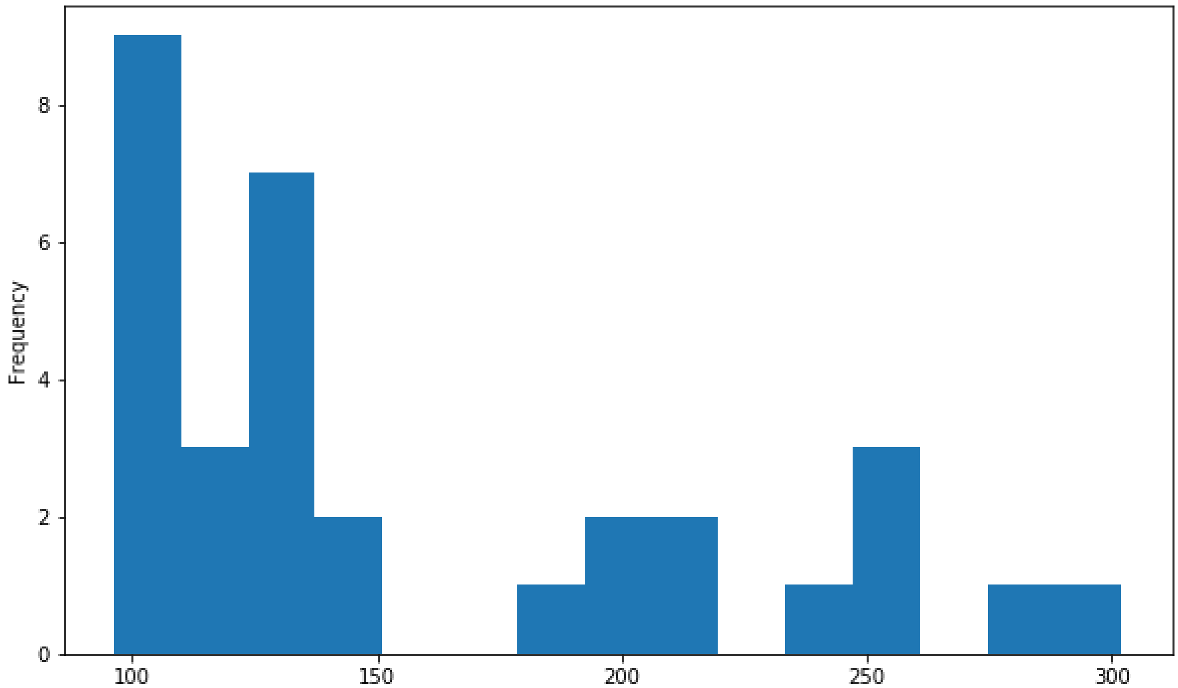
In [942]:
```
EMGDPML['Sovereign Risk'].plot(kind='hist',bins=15,figsize =(10,6))
```

Out[942]: <matplotlib.axes._subplots.AxesSubplot at 0x24c97294d68>



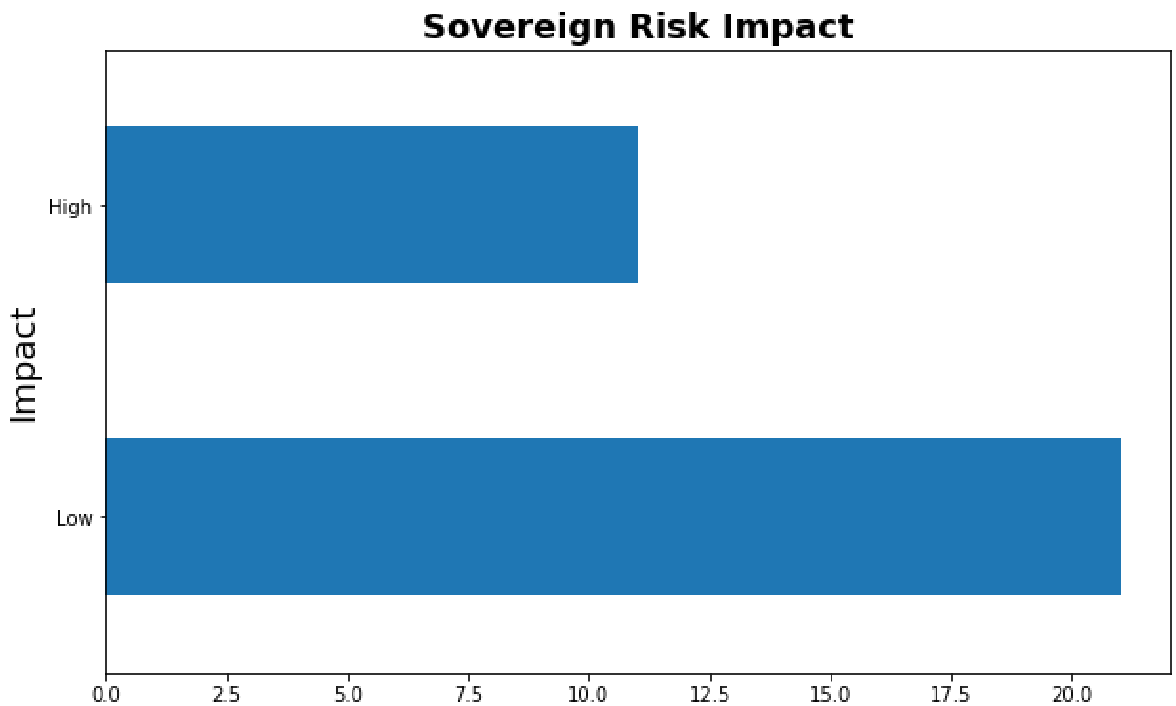
In [944]:
```
PriceRiskXIM3['Commodity Prices'].plot(kind='hist',bins=15,figsize =(10,6))
```

Out[944]: <matplotlib.axes._subplots.AxesSubplot at 0x24c98baf828>



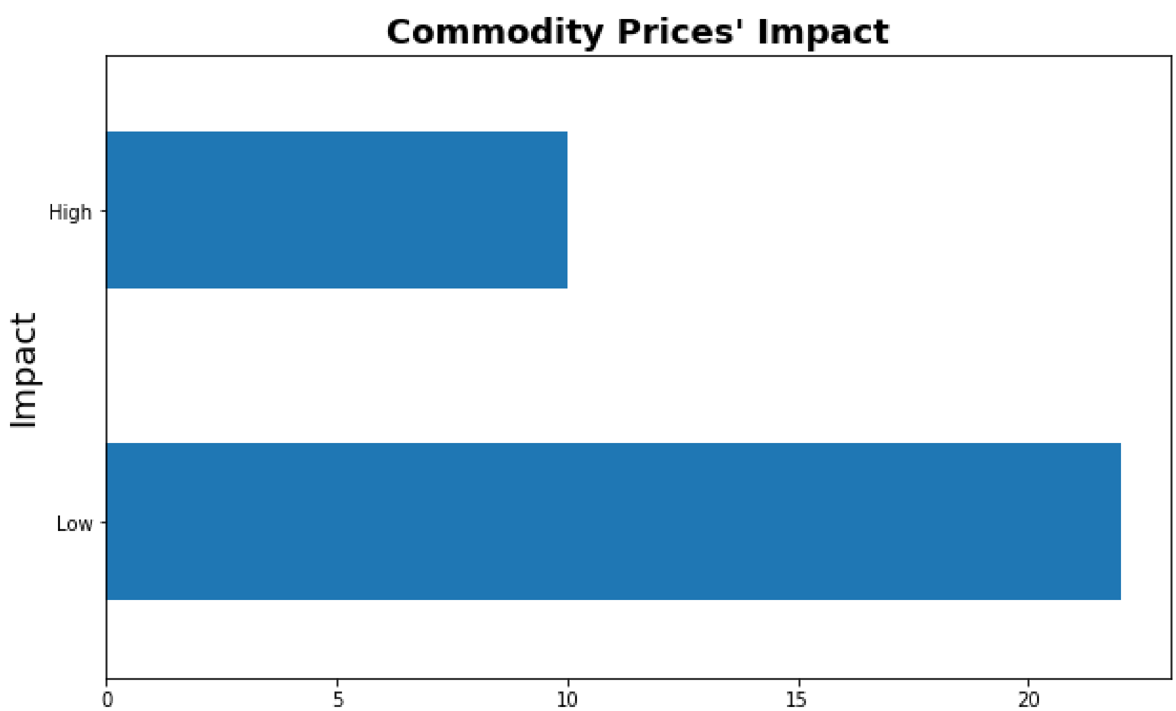
In [907]:
```
fig, axe = plt.subplots()
CGDPSML['Sovereign Risk Impact  '].value_counts().plot.barh(,figsize =(10,6))

axe.set_ylabel("Impact", fontsize=18)

axe.set_title("Sovereign Risk Impact", loc='center', fontsize=18, fontweight =
"bold" )
```

Out[907]:
```
Text(0.5, 1.0, 'Sovereign Risk Impact')
```



In [908]:
```
fig, axe = plt.subplots()
CGDPSML['Commodity Prices Impact  '].value_counts().plot.barh(figsize =(10,6))

axe.set_ylabel("Impact", fontsize=18)

axe.set_title("Commodity Prices' Impact", loc='center', fontsize=18, fontweigh
t = "bold" )
```

Out[908]: Text(0.5, 1.0, "Commodity Prices' Impact")



In [909]:
```
CGDPSML.dtypes
```

Out[909]:
```
Year                           int64
Sovereign Risk               float64
Sovereign Risk Impact         object
Commodity Prices             float64
Commodity Prices Impact       object
GDP                          float64
High or Low Impact GDP        object
dtype: object
```

In [910]:
```
CGDPSML['Commodity Prices Impact  ']=CGDPSML['Commodity Prices Impact  '].asty
pe('category')
CGDPSML['Sovereign Risk Impact  ']=CGDPSML['Sovereign Risk Impact  '].astype(
'category')
CGDPSML[  'High or Low Impact GDP ']=CGDPSML[  'High or Low Impact GDP '].asty
pe('category')
CGDPSML.dtypes
```

Out[910]:
```
Year                              int64
Sovereign Risk                  float64
Sovereign Risk Impact          category
Commodity Prices                float64
Commodity Prices Impact        category
GDP                             float64
High or Low Impact GDP         category
dtype: object
```

In [911]:
```
CGDPSML['Commodity Prices Impact  ']=CGDPSML['Commodity Prices Impact  '].cat.
codes
CGDPSML['Sovereign Risk Impact  ']=CGDPSML['Sovereign Risk Impact  '].cat.code
s
```

In [912]:
```
CGDPSML=pd.get_dummies(CGDPSML, columns=['High or Low Impact GDP '])
```

In [913]:
```
CGDPSML=CGDPSML.rename(columns={'High or Low Impact GDP _High':'Impact_High',
'High or Low Impact GDP _Low':'Impact_Low','Sovereign Risk Impact  ':'Sovereig
nRisk_Impact','Commodity Prices Impact  ':'CommodityPrices_Impact'})
CGDPSML
```

Out[913]:

	Year	Sovereign Risk	SovereignRisk_Impact	Commodity Prices	CommodityPrices_Impact	GDP	Im
0	1985	-0.349830	1	-0.136560	1	-0.646860	
1	1986	-0.326120	1	-0.138430	1	-0.646770	
2	1987	-0.324860	1	-0.156390	1	-0.627380	
3	1988	-0.337940	1	-0.259000	1	-0.597630	
4	1989	-0.344900	1	-0.251050	1	-0.569190	
5	1990	-0.366170	1	-0.214750	1	-0.553190	
6	1991	-0.407280	1	-0.187410	1	-0.545220	
7	1992	-0.448840	1	-0.168200	1	-0.520010	
8	1993	-0.424400	1	-0.135750	1	-0.471700	
9	1994	-0.390140	1	-0.173480	1	-0.429390	
10	1995	-0.336900	1	-0.176800	1	-0.372610	
11	1996	-0.293730	1	-0.144550	1	-0.318420	
12	1997	-0.256870	1	-0.125810	1	-0.287980	
13	1998	-0.262220	1	-0.105370	1	-0.322320	
14	1999	-0.278070	1	-0.075370	1	-0.334180	
15	2000	-0.243880	1	-0.067590	1	-0.287430	
16	2001	-0.237200	1	-0.058420	1	-0.275420	
17	2002	-0.232330	1	-0.058500	1	-0.268220	
18	2003	-0.176530	1	-0.053050	1	-0.199340	
19	2004	-0.077090	1	-0.033260	1	-0.083270	
20	2005	0.058557	1	0.030455	1	0.061844	
21	2006	0.225365	0	0.176518	1	0.231445	
22	2007	0.475890	0	0.444736	0	0.486789	
23	2008	0.712966	0	0.919430	0	0.748171	
24	2009	0.663737	0	0.680342	0	0.713742	
25	2010	1.048985	0	1.298801	0	1.056829	
26	2011	1.411043	0	2.158339	0	1.418425	
27	2012	1.549017	0	2.134784	0	1.568483	
28	2013	1.702048	0	2.142136	0	1.723921	
29	2014	1.771486	0	2.077238	0	1.817723	
30	2015	1.598742	0	1.501260	0	1.701941	
31	2016	1.606178	0	1.482809	0	1.699328	

In [914]:
```
CGDPSML.Impact_High.value_counts()
```
Out[914]:
```
1    25
0     7
Name: Impact_High, dtype: int64
```

In [915]:
```
CGDPSML.columns
```
Out[915]:
```
Index(['Year ', 'Sovereign Risk ', 'SovereignRisk_Impact', 'Commodity Prices
',
       'CommodityPrices_Impact', 'GDP ', 'Impact_High', 'Impact_Low'],
      dtype='object')
```

In [916]:
```
from patsy import dmatrices
y,X = dmatrices('Impact_High ~SovereignRisk_Impact+CommodityPrices_Impact',dat
a=CGDPSML)
```

In [917]:
```
y=np.ravel(y)
```

In [918]:
```
from sklearn.linear_model import LogisticRegression as logit
logit(solver='lbfgs').fit(X,y).score(X,y)
```
Out[918]:
```
array(0.78125)
```

In [919]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(logit(solver='lbfgs'),X,y,cv=5,scoring='accuracy').mean()
```
Out[919]:
```
0.7857142857142858
```

In [920]:
```
cross_val_score(logit(solver='lbfgs'),X,y,cv=3,scoring='roc_auc').mean()
```
Out[920]:
```
0.6041666666666666
```

In [921]:
```
from sklearn.neighbors import KNeighborsClassifier as knn
cross_val_score(knn(),X,y,cv=3,scoring='accuracy').mean()
```
Out[921]:
```
0.7833333333333333
```

In [922]:
```
from sklearn.ensemble import RandomForestClassifier as rf
cross_val_score(rf(n_estimators=100, max_depth=2,max_features=2),X,y,cv=2,scor
ing='accuracy').mean()
```
Out[922]:
```
0.7823529411764706
```

In [923]:
```
for i in range(1,10):
    for j in range(1,3):
        print(i,j,cross_val_score(rf(n_estimators=100,max_depth=i,max_features
=j),X,y,cv=3,scoring='accuracy').mean())
```

```
1 1 0.7833333333333333
1 2 0.7833333333333333
2 1 0.7833333333333333
2 2 0.7833333333333333
3 1 0.7833333333333333
3 2 0.7833333333333333
4 1 0.7833333333333333
4 2 0.7833333333333333
5 1 0.7833333333333333
5 2 0.7833333333333333
6 1 0.7833333333333333
6 2 0.7833333333333333
7 1 0.7833333333333333
7 2 0.7833333333333333
8 1 0.7833333333333333
8 2 0.7833333333333333
9 1 0.7833333333333333
9 2 0.7833333333333333
```

In [924]:
```
from sklearn.linear_model import LogisticRegression as reg
from sklearn.model_selection import train_test_split
```

In [925]:
```
x=CGDPSML.drop(['Impact_High'],axis=1)
y=CGDPSML['Impact_High']
```

In [926]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=.2)
```

In [927]:
```
result = reg(solver='liblinear').fit(x_train,y_train).predict(x_test)
```

In [928]:
```
pd.Series(result).value_counts()
```

Out[928]:
```
1    7
dtype: int64
```

This Model shows that both sovereign risk and commodity prices will have a 100% chance of highly impacting EM's GDP.

In [ ]:
```

```