Project: Investigate Children Out of School

Table of Contents

Introduction

> **Key notes**: "Gapminder has collected a lot of information about how people live their lives in different countries, tracked across the years, and on a number of different indicators.
>
> **Questions to explore**:
>
> - 1. Research Question 1: What is the total numbers of children out of primary school over years, indicate the male and female numbers as well?
> - 2. Research Question 2: What is distribution of female children who was out of primary school from 1980 to 1995?
> - 3. Research Question 3: What are numbers of children out of school in total, by male and female in China, 1985?
> - 4. What are relationship of children out of school of female in China in russian and usa over time? Which has a better trend?
> - 5. Research Question 5: What is the overall trend for children out of primary school over the years?

In [1]:

```
# Set up import statements for all of the packages that are planed to use;
# Include a 'magic word' so that visualizations are plotted;
# call on dataframe to display the first 5 rows.

import pandas as pd
import numpy as np
import datetime
from statistics import mode
% matplotlib inline
import matplotlib.pyplot as plt
%config InlineBackend.figure_format = 'retina'
import seaborn as sns
sns.set_style('darkgrid')
```

In [2]:

```
# Reading an Excel file in python using pandas
# call on dataframe to display the first 5 rows

xl = pd.ExcelFile('Child out of school primary.xlsx')

xl.sheet_names
[u'Data']

df_tot = xl.parse("Data")
df_tot.head()
```

Out[2]:

	Children out of school, primary	1970	1971	1972	1973	1974	1975	1976	1977	
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	Afghanistan	NaN	NaN	NaN	NaN	1559835.0	NaN	NaN	NaN	Na
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	Albania	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	Algeria	NaN	NaN	NaN	806895.0	731571.0	696223.0	671496.0	646398.0	62

5 rows × 43 columns

In [3]:

```
x2 = pd.ExcelFile('Child out of school primiary female.xlsx')

x2.sheet_names
[u'Data']

df_f = x2.parse("Data")
df_f.head()
```

Out[3]:

	Children out of school, primary, female	1970	1971	1972	1973	1974	1975	1976	1977	
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	Afghanistan	NaN	NaN	NaN	NaN	923588.0	NaN	NaN	NaN	NaN
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	Albania	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Algeria	NaN	NaN	NaN	564297.0	538928.0	523330.0	509066.0	491870.0	476

5 rows × 43 columns

In [4]:

```
x3 = pd.ExcelFile('Child out of school primiary male.xlsx')

x3.sheet_names
[u'Data']

df_m = x3.parse("Data")
df_m.head()
```

Out[4]:

	Children out of school, primary, male	1970	1971	1972	1973	1974	1975	1976	1977	
0	Abkhazia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	Afghanistan	NaN	NaN	NaN	NaN	636247.0	NaN	NaN	NaN	Na
2	Akrotiri and Dhekelia	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	Albania	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	Algeria	NaN	NaN	NaN	242598.0	192643.0	172893.0	162430.0	154528.0	150

5 rows × 43 columns

In [5]:

```
# Check if the three dataframe have the same shape

df_tot.shape, df_m.shape, df_f.shape
```

Out[5]:

```
((275, 43), (275, 43), (275, 43))
```

In [6]:

```
# Check if the first columns from the 3 dataframe are exactly the same
assert (df_tot['Children out of school, primary'].tolist() == df_m['Children out
of school, primary, male'].tolist()\
       == df_f['Children out of school, primary, female'].tolist())
```

In [7]:

```
# Merge the 3 dataframe

df1 = df_tot.merge(df_f, how='outer', left_index = True, right_index = True)

df1 = df1.merge(df_m, how='outer', left_index = True, right_index = True)

# Confirm changes

df1.shape
```

Out[7]:

```
(275, 129)
```

Data Wrangling

> **Key notes**: In this section of the report, the following work will be done: load the data; check for cleanliness; trim and clean dataset for analysis.

General Properties

In [8]:

```
# return the datatypes of the columns.

df1.dtypes
```

Out[8]:

```
Children out of school, primary     object
1970_x                             float64
1971_x                             float64
1972_x                             float64
1973_x                             float64
1974_x                             float64
1975_x                             float64
1976_x                             float64
1977_x                             float64
1978_x                             float64
1979_x                             float64
1980_x                             float64
1981_x                             float64
1982_x                             float64
1983_x                             float64
1984_x                             float64
1985_x                             float64
1986_x                             float64
```

```
1987_x                             float64
1988_x                             float64
1989_x                             float64
1990_x                             float64
1991_x                             float64
1992_x                             float64
1993_x                             float64
1994_x                             float64
1995_x                             float64
1996_x                             float64
1997_x                             float64
1998_x                             float64
                                    ...
1982                               float64
1983                               float64
1984                               float64
1985                               float64
1986                               float64
1987                               float64
1988                               float64
1989                               float64
1990                               float64
1991                               float64
1992                               float64
1993                               float64
1994                               float64
1995                               float64
1996                               float64
1997                               float64
1998                               float64
1999                               float64
2000                               float64
2001                               float64
2002                               float64
2003                               float64
2004                               float64
2005                               float64
2006                               float64
2007                               float64
2008                               float64
2009                               float64
2010                               float64
2011                               float64
Length: 129, dtype: object
```

In [9]:

```
# check for duplicates in the data.

sum(df1.duplicated())
```

Out[9]:

```
0
```

In [10]:

```
#  check if any value is NaN in DataFrame and in how many columns

df1.isnull().any().any(), sum(df1.isnull().any())
```

Out[10]:

```
(True, 126)
```

In [11]:

```
# Generates descriptive statistics, excluding NaN values.

df1.describe()
```

Out[11]:

	1970_x	**1971_x**	**1972_x**	**1973_x**	**1974_x**	
count	1.700000e+01	4.900000e+01	5.300000e+01	5.300000e+01	4.800000e+01	4.5000
mean	6.843502e+05	1.022907e+06	3.314347e+05	3.489581e+05	4.035501e+05	4.230
std	1.321416e+06	4.052435e+06	8.888364e+05	6.248331e+05	9.659810e+05	1.0987
min	4.610000e+02	0.000000e+00	1.800000e+01	1.160000e+03	1.860000e+02	0.0000
25%	6.148100e+04	2.506400e+04	2.677500e+04	3.293600e+04	2.891925e+04	2.4000
50%	1.633390e+05	9.488500e+04	7.827400e+04	9.425400e+04	8.502600e+04	7.1392
75%	4.429960e+05	4.115620e+05	1.959640e+05	4.260080e+05	3.638538e+05	2.3386
max	4.782696e+06	2.799442e+07	5.911106e+06	3.151254e+06	6.276900e+06	5.9376

8 rows × 126 columns

Data Cleaning

In [12]:

```
# Locate the columns whose NaN values needs to be treated

col = df1.drop(['Children out of school, primary', 'Children out of school, primary, female'\
               , 'Children out of school, primary, male'], axis=1)

# Replace NaN with mean

for c in col:
    c_mean = df1[c].mean()
    df1[c].fillna(c_mean, inplace = True)

# Confirm changes

df1.isnull().any().any()
```

Out[12]:

```
False
```

In [13]:

```
# Rename column for simplification

df1.rename(columns = {'Children out of school, primary':'country'}, inplace = True)

# check the new dataframe

df1.head()
```

Out[13]:

	country	1970_x	1971_x	1972_x	1973_x	1974_
0	Abkhazia	684350.176471	1.022907e+06	331434.679245	348958.09434	403550.125
1	Afghanistan	684350.176471	1.022907e+06	331434.679245	348958.09434	1559835.00
2	Akrotiri and Dhekelia	684350.176471	1.022907e+06	331434.679245	348958.09434	403550.125
3	Albania	684350.176471	1.022907e+06	331434.679245	348958.09434	403550.125
4	Algeria	684350.176471	1.022907e+06	331434.679245	806895.00000	731571.00

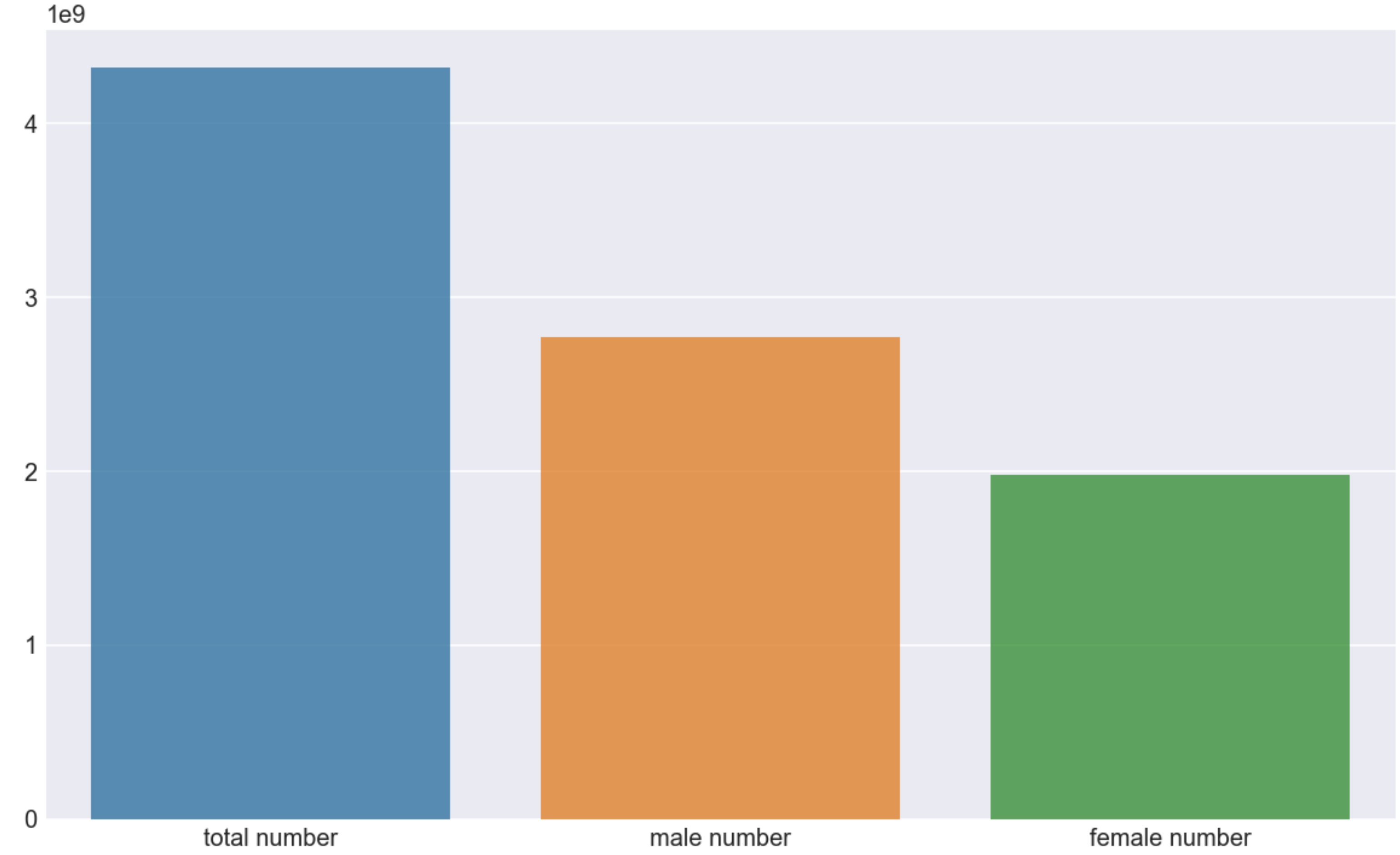
5 rows × 129 columns

Exploratory Data Analysis

Research Question 1: What is the total numbers of children out of primary school over years, indicate the male and female numbers as well?

In [14]:

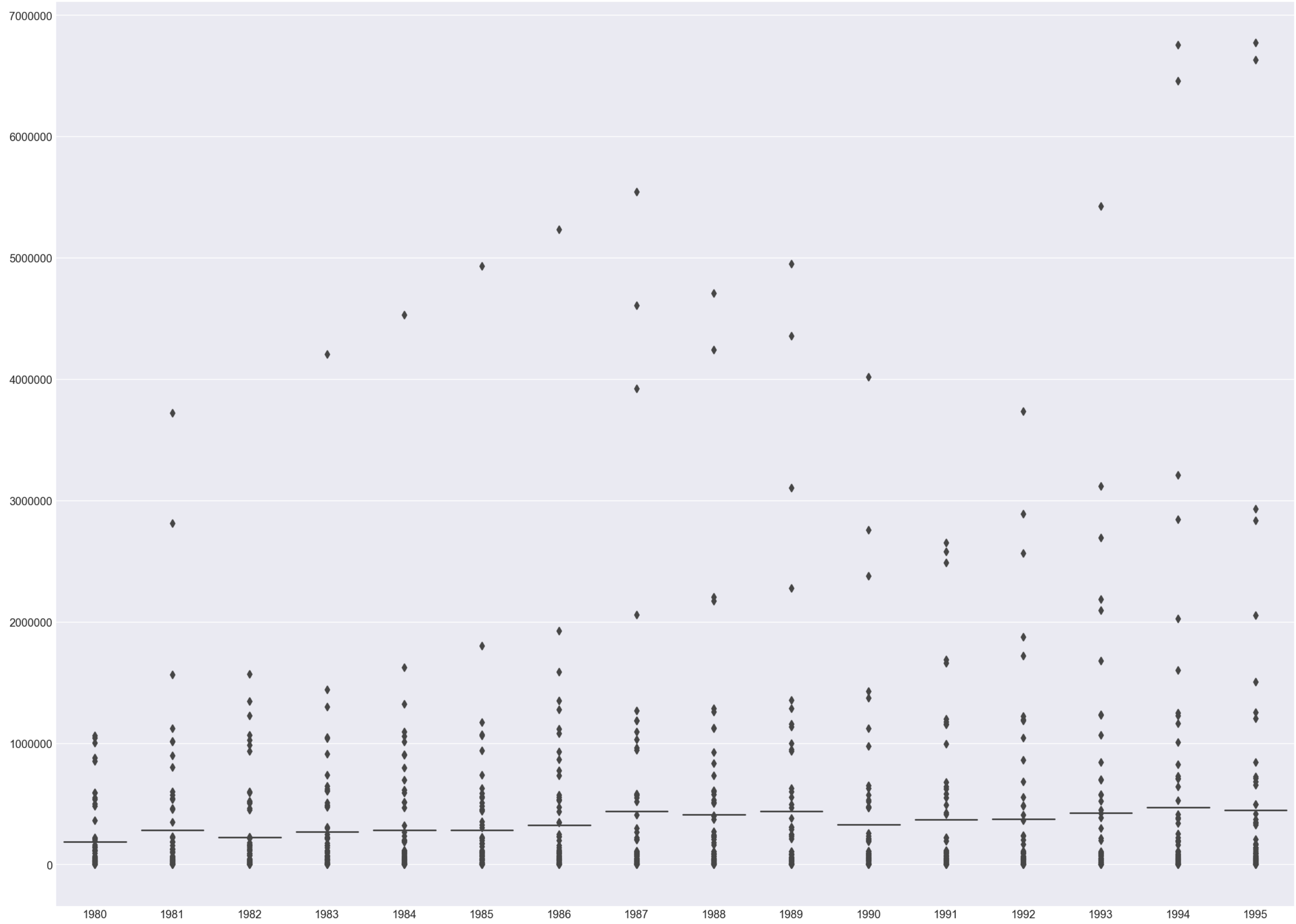
```
# Get the sum for each group

sum_tot = df1.iloc[:, 1:43]
m_tot = df1.iloc[:, 44:86]
f_tot = df1.iloc[:, 87:]



tot = []
for t in sum_tot.columns:
    tot.append(sum_tot[t].sum())



m = []
for ma in m_tot.columns:
    m.append(m_tot[ma].sum())



f = []
for fa in f_tot.columns:
    f.append(f_tot[fa].sum())

# Plot

x = ['total number', 'male number', 'female number']
y = [sum(tot), sum(m), sum(f)]

plt.subplots(figsize=(10,6))
sns.barplot(x,y, alpha = 0.8);
```



Research Question 2: What is distribution of female children who was out of primary school from 1980 to 1995?

In [15]:

```
# Target the year and plot

sum_tot1 = sum_tot.iloc[:, 10:26]
new_col = []
for ele in sum_tot1.columns:
    new_col.append(ele.split('_x')[0])

sum_tot1.columns = new_col

plt.figure(figsize=(20,15))
sns.boxplot(data = sum_tot1);
```



Research Question 3: What are numbers of children out of school in total, by male and female in China, 1985?

In [16]:

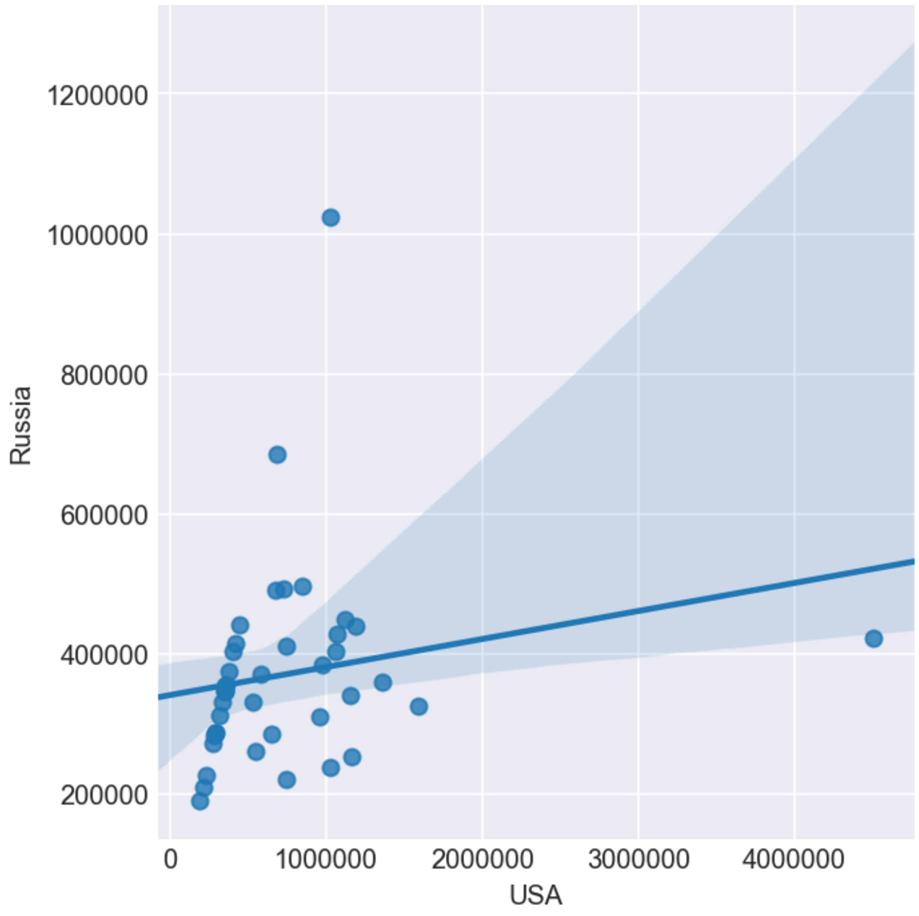
```
china = df1.copy()
china = china.set_index('country')
tot_chi = china.loc['China', '1985_x']
f_chi = china.loc['China', '1985_y']
m_chi = china.loc['China', '1985']

print('The numbers of children out of school in total, by male and female in Chi
na were \
{0:.0f}, {1:.0f} and {2:.0f} in 1985, respectively.'.format(tot_chi, f_chi, m_ch
i))
```

```
The numbers of children out of school in total, by male and female i
n China were 284172, 177367 and 122195 in 1985, respectively.
```

Research Question 4: What are relationship of children out of school of female in China in russian and usa over time? Which has a better trend?

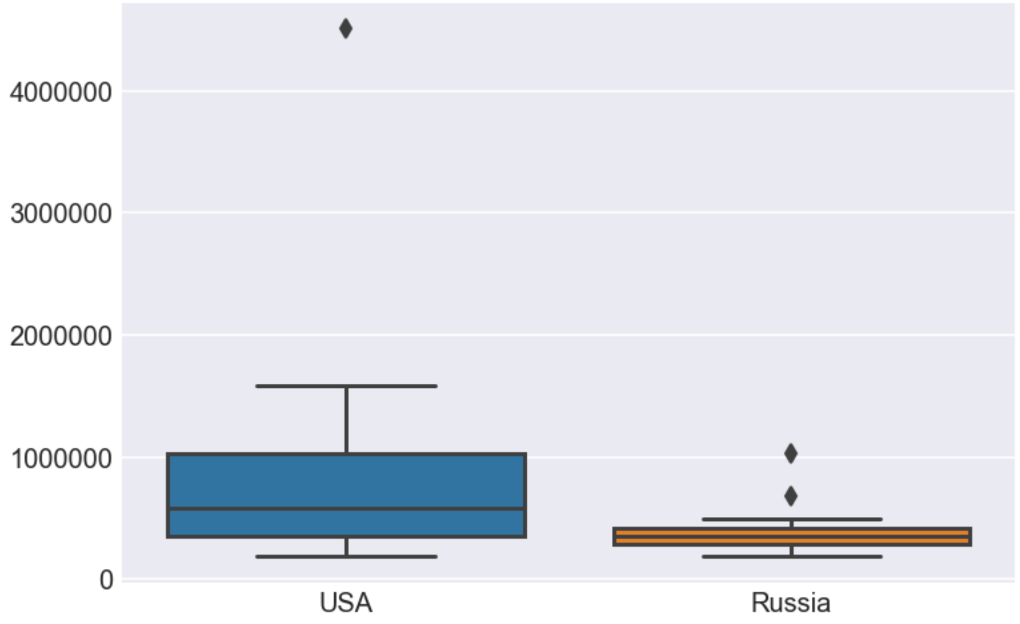
In [17]:

```
rus_us = df1.iloc[:, 0:42].copy()



new_col1 = []
for ele in rus_us:
    new_col1.append(ele.split('_x')[0])

rus_us.columns = new_col1

rus_us = rus_us.set_index('country')
rus_us_df = pd.DataFrame(columns=['USA','Russia'])
rus_us_df['USA'] = rus_us.loc['United States'].values
rus_us_df['Russia'] = rus_us.loc['Russia'].values



sns.lmplot(x = 'USA', y = 'Russia', data = rus_us_df);
```



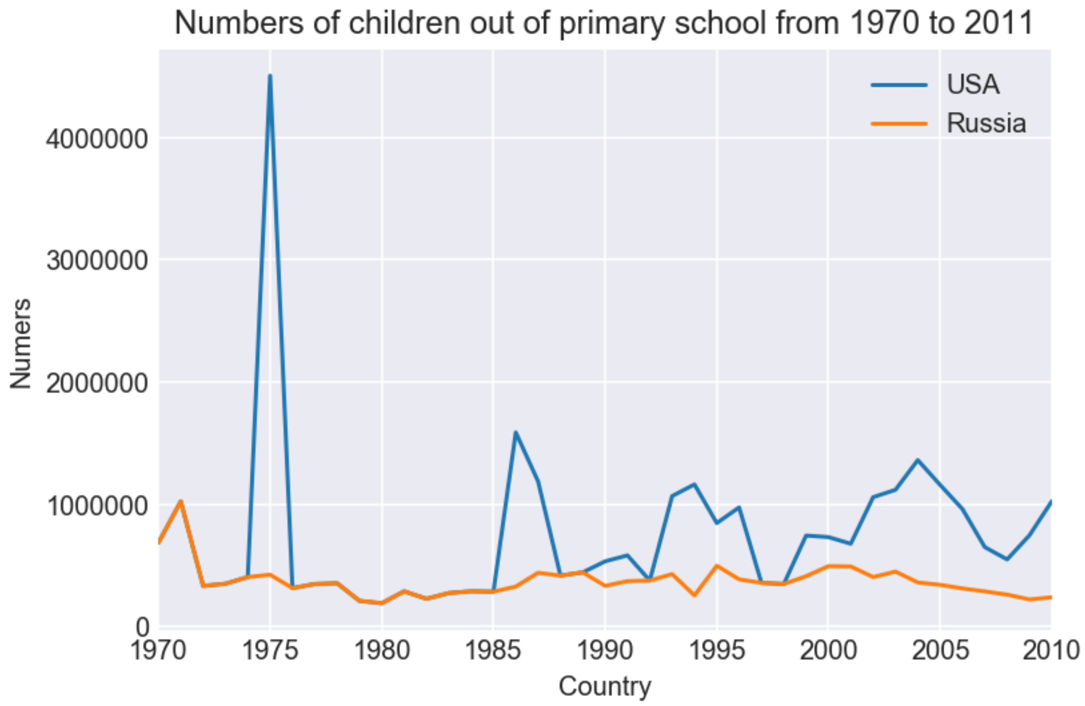
In [18]:

```
sns.boxplot(data=rus_us_df);
```



```
In [19]:
```

```
rus_us_df['year'] = rus_us.columns
rus_us_df.index = rus_us_df.year
rus_us_df.plot();
plt.ylabel('Numers')
plt.xlabel('Country')
plt.title('Numbers of children out of primary school from 1970 to 2011');
```

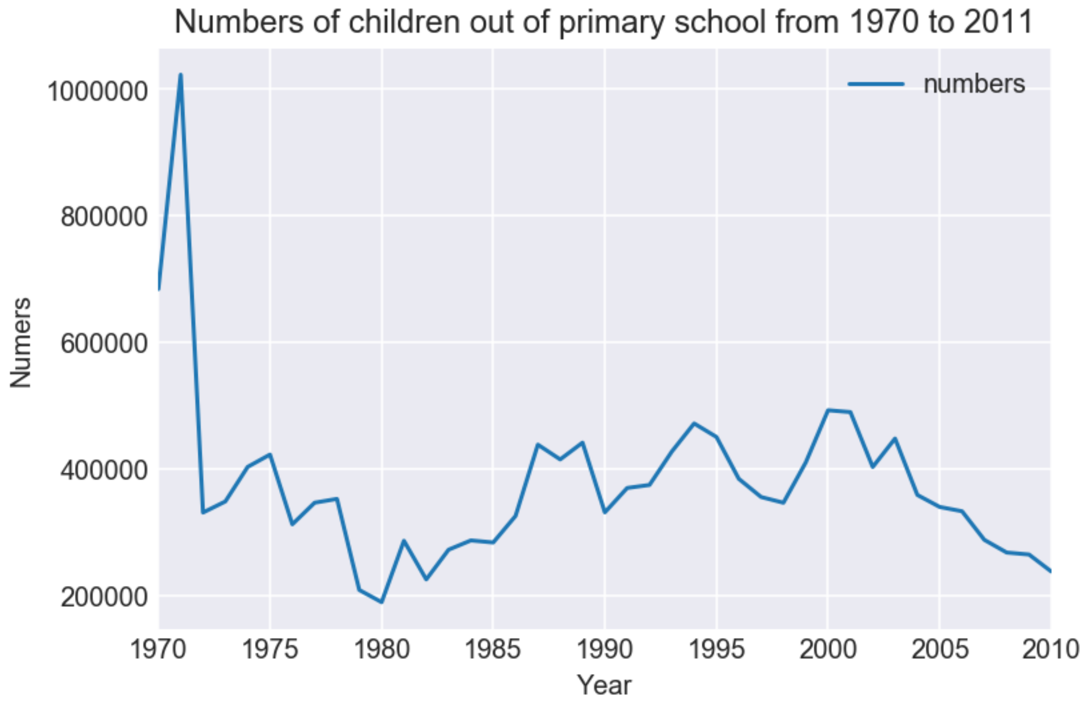


> There is a positive correlation between children droped out of primary school in Russia and USA. The estimated linear regression is shown as the blue line, the estimates varies in the light blue shade with 95% confident level. The trend of children out of school in USA is much higher than that of Russia over that past 40 years.

Research Question 5: What is the overall trend for children out of primary school over the years?

In [20]:

```
overall_df = pd.DataFrame(columns=['year','numbers'])
overall_df['year'] = rus_us.columns
n_list =[]

for n in rus_us.columns:
    n_list.append(rus_us[n].mean())


overall_df['numbers'] = np.array(n_list)
overall_df.index = overall_df.year

overall_df.plot();
plt.ylabel('Numers')
plt.xlabel('Year')
plt.title('Numbers of children out of primary school from 1970 to 2011');
```



From the analysis we can conclude that the overall trend of children out of primary school had been descreasing starting between 1970 and 1975 at which point of time the numbers fell down dramatically

Conclusions

> In current study, a good amount of profound analysis has been carried out. Prior to each step, deailed instructions was given and interpretions was also provided afterwards. The dataset across 41 years from 1970 to 2011.
>
> The limitations of current study was that the structure is only 275*42 in shape, thus the analysis would not be much reliable due to small scale samples.
>
> In addition, the parameters in the dataset is very simple, it only focus on the number of children out of school.

In [21]:

```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Children_Out_of_School_Primary.ipynb'])
```

Out[21]:

```
0
```