Simulation

Assignment 2

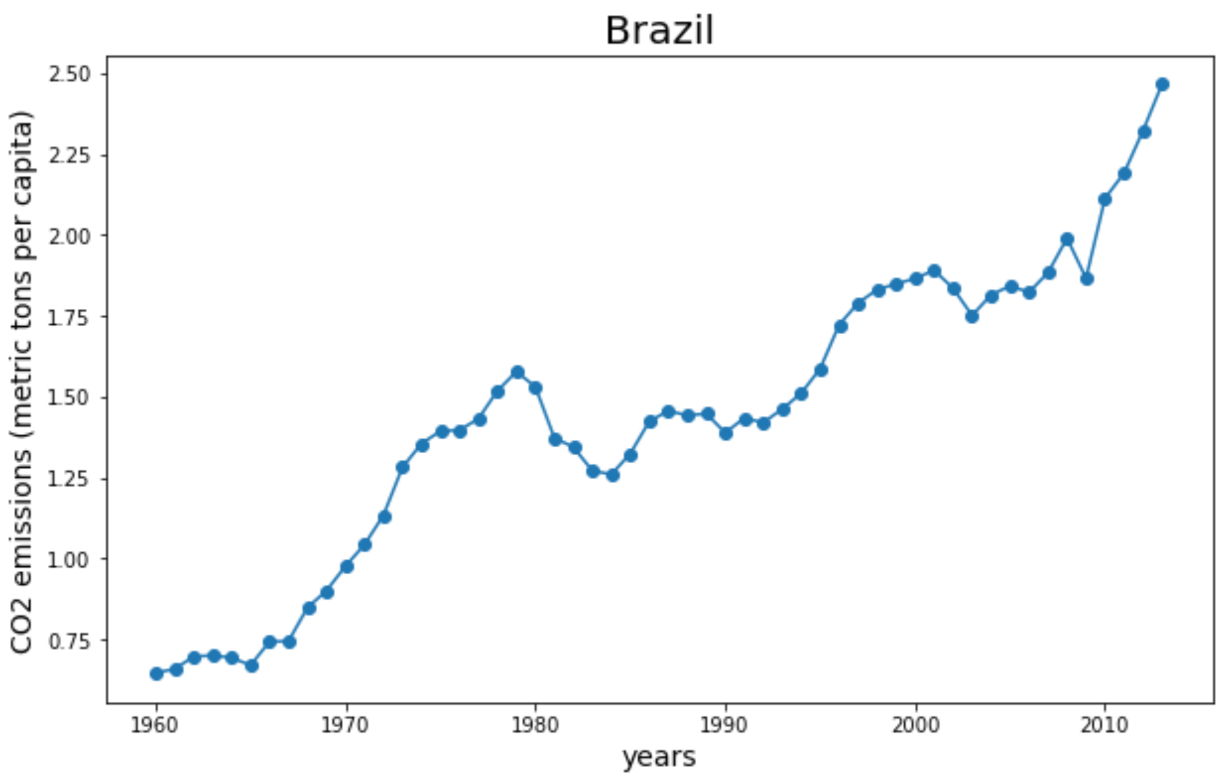
Name: Viviana Márquez
Code: 614132005
Date: Saturday, February 18th, 2016

In [1]:

```
it=0
import csv
with open('Data.csv') as f:
    reader = csv.reader(f)
    for row in reader:
        if it==4:
            fieldnames = row
        if it>4:
            if row[1]=='BRA': #Change here country
                usa = row
            if row[1]=='RUS':
                rus = row
        it = it+1

fieldnames_rus=fieldnames[36:]
fieldnames_rus=fieldnames[:-4]
fieldnames_rus=fieldnames_rus[36:]
rus=rus[36:]
rus=rus[:-4]
usa=usa[4:]
usa=usa[:-4]
fieldnames=fieldnames[4:]
fieldnames=fieldnames[:-4]
```

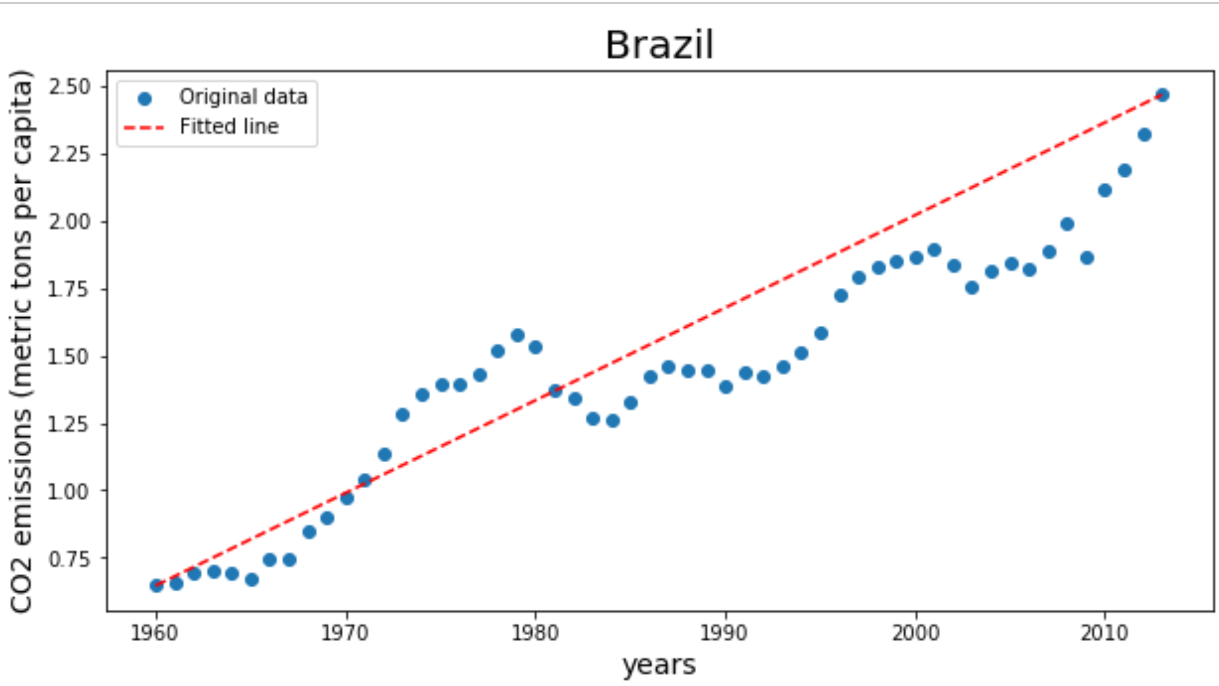
```
In [2]: %matplotlib inline
        import matplotlib as mpl
        import matplotlib.pyplot as plt
        mpl.rc("figure", figsize=(10, 6))
        plt.plot(fieldnames,usa,'o-')
        #plt.plot(fieldnames_rus,rus)
        plt.title('Brazil', fontsize=20)
        plt.xlabel('years', fontsize=14)
        plt.ylabel('CO2 emissions (metric tons per capita)', fontsize=14)
        plt.show()
```



1. Data

1. (5 Points) Fit a straight line passing through the endpoints;

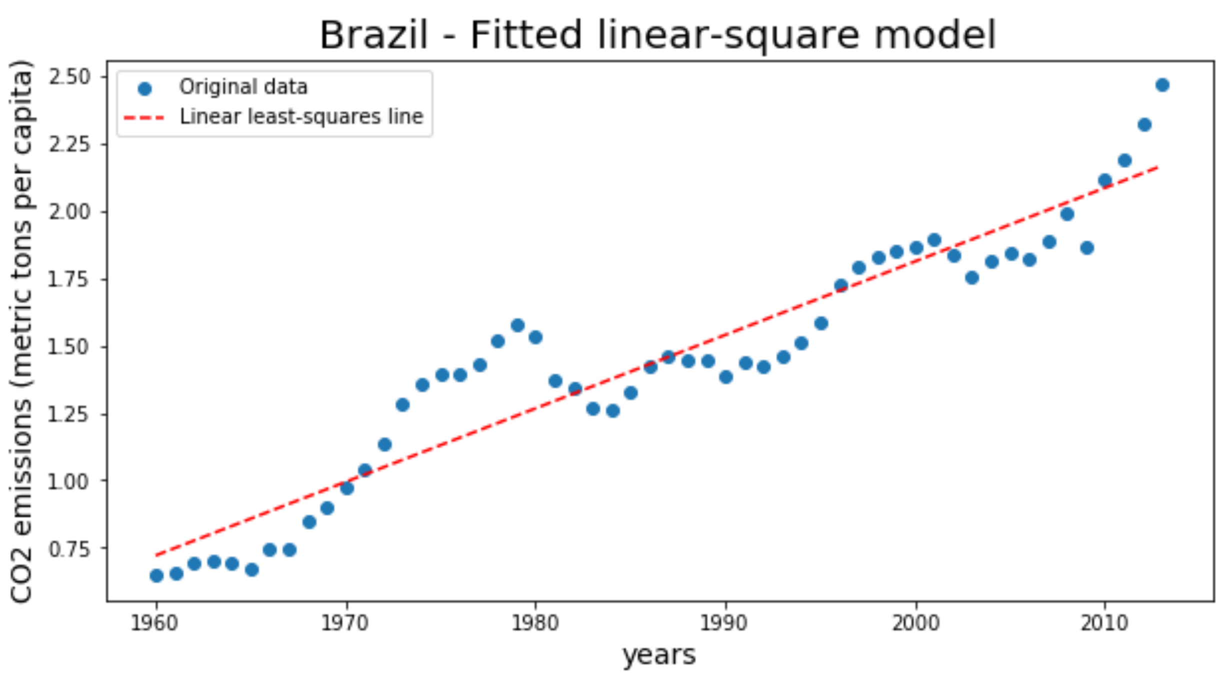
```
In [3]: mpl.rc("figure", figsize=(10, 5))
        plt.plot(fieldnames,usa, 'o',label='Original data')
        plt.plot([fieldnames[0],fieldnames[len(fieldnames)-1]],[usa[0],usa[len
        (usa)-1]],'--', color='r',label='Fitted line')
        plt.title('Brazil', fontsize=20)
        plt.xlabel('years', fontsize=14)
        plt.ylabel('CO2 emissions (metric tons per capita)', fontsize=14)
        plt.legend()
        plt.show()
```



2.- (15 Points) Fit a linear least-squares model. Find the maximum error in magnitude in the first case. Predict the year in which the CO2 level may exceed 3 metric tons per capita.

In [4]:
```
import numpy as np
x = np.asarray(fieldnames, dtype=float)
y = np.asarray(usa, dtype=float)
A = np.vstack([x, np.ones(len(x))]).T
m, c = np.linalg.lstsq(A, y)[0]
```

In [5]:
```
plt.plot(fieldnames,usa, 'o',label='Original data')
plt.plot(x, m*x + c, 'r--', label='Linear least-squares line')
y_2 = m*x + c
plt.title('Brazil - Fitted linear-square model', fontsize=20)
plt.xlabel('years', fontsize=14)
plt.ylabel('CO2 emissions (metric tons per capita)', fontsize=14)
plt.legend()
plt.show()
```



Using a linear least-square model, the maximum error in magnitude is 0.33881662634055765, and it

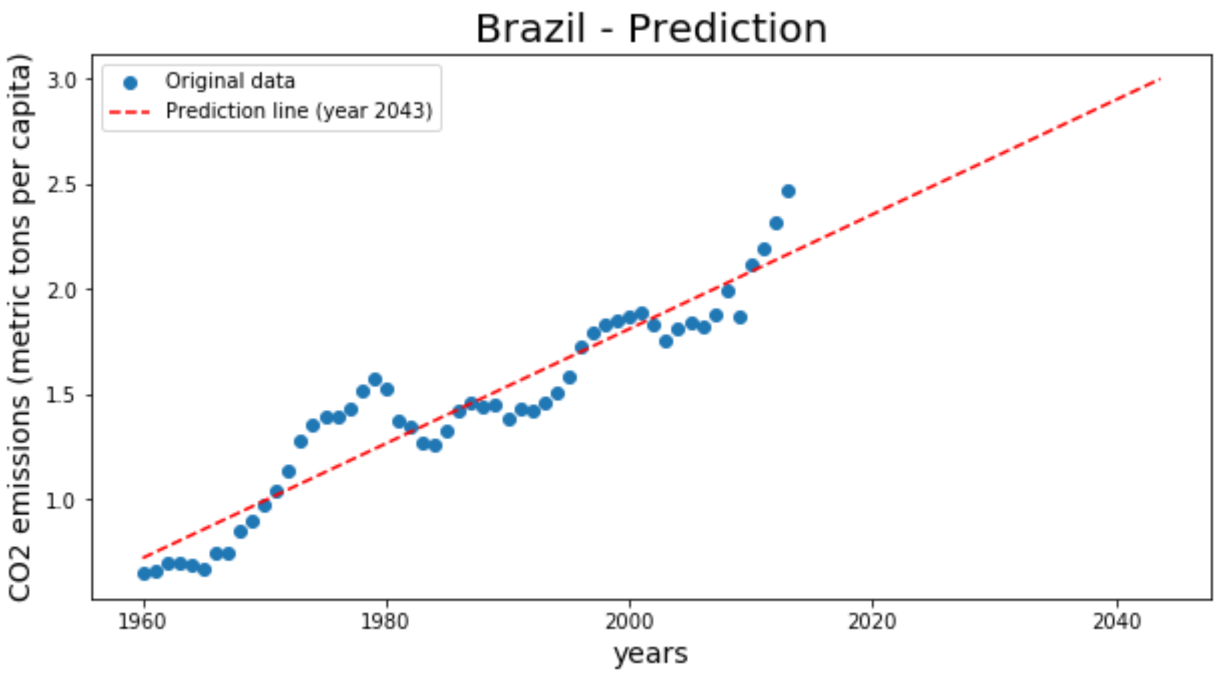
In [6]:
```
error = np.absolute(y-y_2)
print(max(error))
print(fieldnames[np.argmax(error)])
```

```
0.338816626341
1979
```

In [7]:
```
slope = (y_2[len(y_2)-1]-y_2[0])/(x[len(x)-1]-x[0])
var = ((3-y_2[0])/(slope))+x[0]
var
```

Out[7]: 2043.6336215957022

In [8]:
```
plt.plot(fieldnames,usa, 'o',label='Original data')
plt.plot([x[0],var],[y_2[0],3],'--', color='r',label='Prediction line
(year 2043)')
plt.title('Brazil - Prediction', fontsize=20)
plt.xlabel('years', fontsize=14)
plt.ylabel('CO2 emissions (metric tons per capita)', fontsize=14)
plt.legend()
plt.show()
```



The year in which the CO2 level may exceed 3 metric tons per capita in Brazil is 2043.

Simulation

Assignment 2

Name: Viviana Márquez
Code: 614132005
Date: Saturday, February 18th, 2016

---

2. Mathematical Preliminaries

• (5 Points) Is the following time-varying function positive definite?

$$V(t, x_1, x_2) = t(x_1^2 + x_2^2) + 4x_1 x_2 \sin t$$

Solution:

A function $V : \mathbb{R}^n \to \mathbb{R}$ is positive definite if:
• $V(z) = 0$ iff $z = 0$
• $V(z) > 0 \; \forall z, z \neq 0$

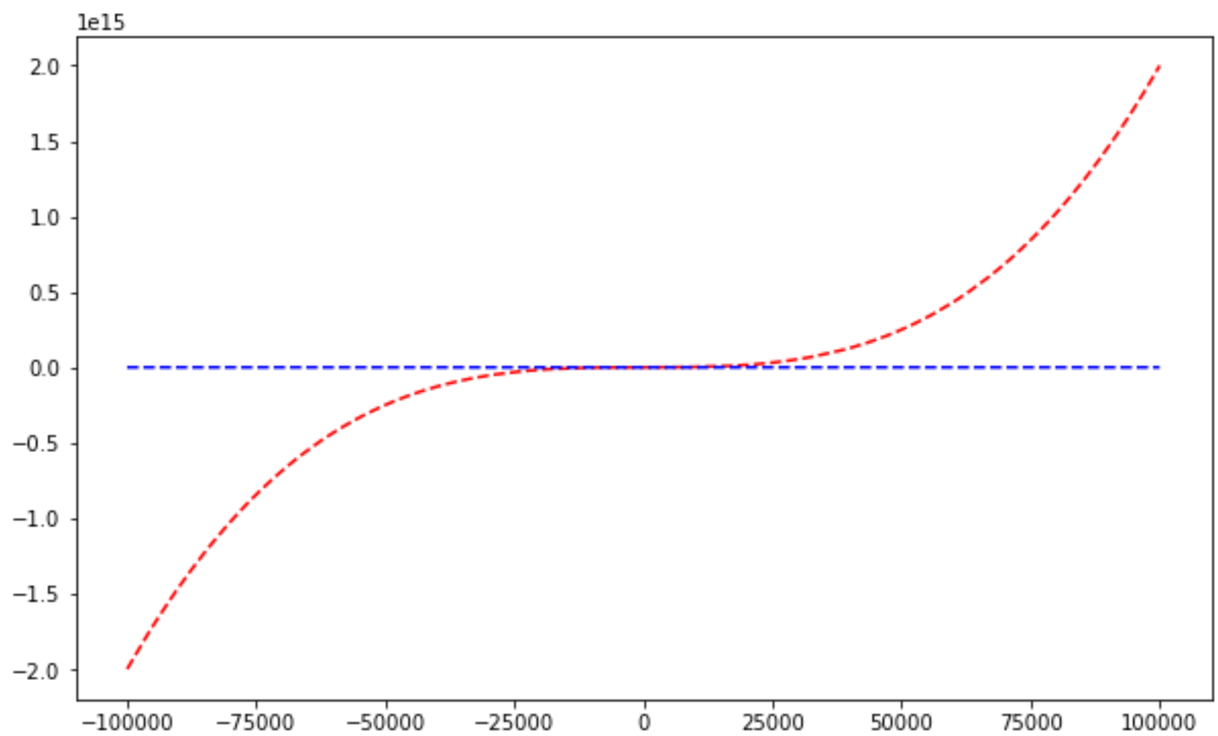
1. $V(t, 0, 0) = 0$ ✓
2. $V(t, x_1, x_2) \geq V(x_1, x_2) \; \forall x \neq 0$. From the numerical approach below, we can conclude that there is no such function.

   Therefore $V(t, x, y)$ is positive semidefinite.

In [1]:
```
import numpy as np
x = np.arange(-99999,99999)
y = np.arange(-99999,99999)
w = np.arange(-99999,99999)
```

In [2]:
```
z = w*(x**2 + y**2) + 4*x*y*np.sin(w)
z2 = (x**2 + y**2) + 4*x*y
```

In [3]:
```
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc("figure", figsize=(10, 6))
plt.plot(x,z,'--', color='r')
plt.plot(x,z2,'--', color='b')
plt.show()
```



• (10 Points) Test the stability of the zero solution for the following system:

$$x_1' = -x_1^3 + x_1^4$$
$$x_2' = x_1^4 + x_2^3$$

Solution:
Let us solve this problem using Krasovskii's method:

$$J(X) = \begin{pmatrix} -3x_1^2 + 4x_1^3 & 0 \\ 4x_1^3 & 3x_2^2 \end{pmatrix}$$

$$J^T(X) = \begin{pmatrix} -3x_1^2 + 4x_1^3 & 4x_1^3 \\ 0 & 3x_2^2 \end{pmatrix}$$

$$M(x_1, x_2) = J(X) + J^T(X) = \begin{pmatrix} -6x_1^2 + 8x_1^3 & 4x_1^3 \\ 4x_1^3 & 6x_2^2 \end{pmatrix}$$

$M(x_1, x_2)$ is negative definite, therefore $V^*(x)$ is negative as well. By theorem shown in class, then we can affirm the zero solution is asymptocally stable.

- (15 points) Test the linear stability of the zero solution $x_1(t) = 0, x_2(t) = 0$ in the Lotka-Volterra population model, i.e.,

$$\begin{aligned} x_1' &= ax_1 + x_1x_2 \\ x_2' &= -bx_2 + x_1x_2 \,, \end{aligned}$$

for your favorite pair of integers (a, b).

Solution:
Let us choose $a = 7$ and $b = 13$.

Now, let us set the following equations in order to be able to obtain the critical points:
$x_1' = 7x_1 + x_1x_2 = 0$
$x_2' = -13x_2 + x_1x_2 = 0$

In [4]:
```
def f(x,y):
    return 7*x + x*y
def g(x,y):
    return - 13*y + x*y

cp = []

def critical_points(r):
    for x in range(r):
        for y in range(r):
            if ((f(x,y) == 0) and (g(x,y) == 0)):
                cp.append((x,y))
                print('Critical point in %s,%s'% (x,y))
    return cp

critical_points(1000)
```

```
Critical point in 0,0
```

Out[4]: `[(0, 0)]`

Let us find the linearized system about $(0, 0)$

$x' = Ax$, where $A = \begin{pmatrix} 7 & 0 \\ 0 & -13 \end{pmatrix}$

In [5]:
```
from sympy import *
init_printing(use_unicode=True)
M = Matrix([[7,0], [0,-13]])
M.eigenvals()
```

Out[5]: $\{-13 : 1, \quad 7 : 1\}$

Therefore $(0, 0)$ is a saddle point, therefore unestable.