In [94]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot  as plt
%matplotlib inline
import seaborn as sns
import math as m
import sympy as sp
from sympy import *
import pandas as pd
import
pd.set_option('display.max_columns', 100)

pd.set_option('display.max_rows', 100)
pd.set_option('display.width', 100)
```

In [95]:

```
y1, y2, y3, z1, z2, z3 = symbols('y1, y2, y3, z1, z2, z3')
```

In [3]:

```
ks = []
#vs = [0.9878042228*v[0] + 0.6802782372*v[-3], 0.9923751319*v[0] + 0.4964510255*v[-3],
0.9874137376*v[0] + 0.6948766460*v[-3],0.3003651625*v[-3] - 0.00716082355*v[0], 0.49069
83058*v[-3] - 0.00522580027*v[0],   0.2852226010*v[-3]- 0.00731449101*v[0]]

vs = []
#vs =[1.782741376,1.728868443,1.829729471,-0.01598169399,0.2100827025,-0.007342470031]

#vs = [1.66808246, 1.4888261574, 1.6822903836, 0.29320433895, 0.48547250553, 0.27790810
999]

def linear_equation(c1, c2, c3, c4, c5, c6):

    eq1 =sp.Eq(-0.2949582188e-1*y3+1.147055149*z3+0.51594241e-3*y2-y2+2.427688399*z2-0.
2148538728e-2*y1+.4770180184*z1 , -c1)
    eq2 =sp.Eq(-0.5160018172e-2*y3-y3+1.186217417*z3+0.6192921461e-3*y2+0.6680655564e-1
*z2-0.1586011615e-3*y1+0.3756480678e-1*z1 ,-c2)

    eq3 =sp.Eq(-0.3017149629e-1*y3+1.085604910*z3-0.1181792169e-1*y2+2.903389798*z2-0.7
424319581e-2*y1-y1+1.206128646*z1 , -c3)

    eq4 =sp.Eq(-0.1207426474e-1*y3-1.188625236*z3-0.2555461473e-1*y2-2.452727072*z2-z2-
0.5021242299e-2*y1-.4841877994*z1 , -c4)

    eq5 =sp.Eq(-0.1248649912e-1*y3-1.203863934*z3-z3-0.7032269015e-3*y2-0.6689049039e-1
*z2-0.3954190186e-3*y1-0.3811882695e-1*z1 ,-c5)

    eq6 =sp.Eq(-0.1142742010e-1*y3-1.127203826*z3-0.3056199788e-1*y2-2.945769718*z2-0.1
269609101e-1*y1-1.226249056*z1-z1 ,  -c6)


    ans = sp.solve((eq1, eq2, eq3, eq4, eq5, eq6), (y1,y3, y2, z1,z3,z2))
    for k, v in ans.items():
        v = N((v),10)
        vs.append(v)
        ks.append(k)


linear_equation(1.66808246, 1.4888261574, 1.6822903836, 0.29320433895, 0.48547250553,
0.27790810999)
vs
```

Out[3]:

```
[1.782741376,
 1.728868443,
 1.829729471,
 -0.01598169399,
 0.2100827025,
 -0.007342470031]
```

In [ ]:

In [24]:

```
c = 50000
while c > 0:
    print(linear_equation(
            0.9878042228*vs[-6] + 0.6802782372*vs[-3],
            0.9923751319*vs[-6] + 0.4964510255*vs[-3],
            0.9874137376*vs[-6] + 0.6948766460*vs[-3],
            0.3003651625*vs[-3] - 0.00716082355*vs[-6],
            0.4906983058*vs[-3] - 0.00522580027*vs[-6],
            0.2852226010*vs[-3]- 0.00731449101*vs[-6]))
    c =c - 2


print('end')
```

```
end
Wall time: 5min 16s
Parser   : 105 ms
```

In [96]:

```
df = pd.DataFrame({
    'keys': ks,
    'values': vs
})
df
```

Out[96]:

	keys	values
0	y1	1.782741376
1	y3	1.728868443
2	y2	1.829729471
3	z1	-0.01598169399
4	z3	0.2100827025
5	z2	-0.007342470031
6	y1	1.577315539
7	y3	1.730416348
8	y2	1.626952926
9	z1	-0.01659565151
10	z3	-0.01758599283
11	z2	-0.01709298657
12	y1	1.393375576
13	y3	1.529174650
14	y2	1.437246638
15	z1	-0.01466709019
16	z3	-0.01609484259
17	z2	-0.01512882153
18	y1	1.230879913
19	y3	1.350843601
20	y2	1.269634787
21	z1	-0.01295663061
22	z3	-0.01421940155
23	z2	-0.01336457646
24	y1	1.087334502
25	y3	1.193308009
26	y2	1.121569777
27	z1	-0.01144562634
28	z3	-0.01256113692
29	z2	-0.01180599765
30	y1	0.9605293794
31	y3	1.054144240
32	y2	0.9907721308
33	z1	-0.01011083557
34	z3	-0.01109625515
35	z2	-0.01042918033
36	y1	0.8485122904

	keys	values
37	y3	0.9312097706
38	y2	0.8752281273
39	z1	-0.008931708320
40	z3	-0.009802208109
41	z2	-0.009212927660
42	y1	0.7495586521
43	y3	0.8226119390
44	y2	0.7731588839
45	z1	-0.007890091075
46	z3	-0.008659073040
47	z2	-0.008138514570
48	y1	0.6621450028
49	y3	0.7266788038
...	...	...
27472	z3	-5.992691931e-249
27473	z2	-5.632428595e-249
27474	y1	4.582512467e-247
27475	y3	5.029132084e-247
27476	y2	4.726795181e-247
27477	z1	-4.823697334e-249
27478	z3	-5.293823245e-249
27479	z2	-4.975573876e-249
27480	y1	4.048099134e-247
27481	y3	4.442633900e-247
27482	y2	4.175555575e-247
27483	z1	-4.261156983e-249
27484	z3	-4.676456736e-249
27485	z2	-4.395321659e-249
27486	y1	3.576009169e-247
27487	y3	3.924533228e-247
27488	y2	3.688601619e-247
27489	z1	-3.764220178e-249
27490	z3	-4.131087607e-249
27491	z2	-3.882738548e-249
27492	y1	3.158974411e-247
27493	y3	3.466853539e-247
27494	y2	3.258436312e-247
27495	z1	-3.325236222e-249
27496	z3	-3.649319513e-249

	keys	values
27497	z2	-3.429932961e-249
27498	y1	2.790574313e-247
27499	y3	3.062548527e-247
27500	y2	2.878436951e-247
27501	z1	-2.937446645e-249
27502	z3	-3.223735291e-249
27503	z2	-3.029933633e-249
27504	y1	2.465137092e-247
27505	y3	2.705393631e-247
27506	y2	2.542753175e-247
27507	z1	-2.594881150e-249
27508	z3	-2.847782769e-249
27509	z2	-2.676582291e-249
27510	y1	2.177652412e-247
27511	y3	2.389890195e-247
27512	y2	2.246216896e-247
27513	z1	-2.292265697e-249
27514	z3	-2.515673889e-249
27515	z2	-2.364438839e-249
27516	y1	1.923694242e-247
27517	y3	2.111180821e-247
27518	y2	1.984262724e-247
27519	z1	-2.024941307e-249
27520	z3	-2.222295600e-249
27521	z2	-2.088697605e-249

27522 rows × 2 columns

In [ ]:

```
import pa
```

In [69]:

```
df['values2'] = pd.Series(df['values'])
```

In [1]:

```
df
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-1-00cf07b74dcd> in <module>
----> 1 df

NameError: name 'df' is not defined
```

In [64]:

```

```

Out[64]:

keys	values

In [78]:

```
df.astype('int')
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-78-d2a2db5e8de2> in <module>
----> 1 df.astype('int')

~\Anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args,
**kwargs)
    176                 else:
    177                     kwargs[new_arg_name] = new_arg_value
--> 178             return func(*args, **kwargs)
    179         return wrapper
    180     return _deprecate_kwarg

~\Anaconda3\lib\site-packages\pandas\core\generic.py in astype(self, dtyp
e, copy, errors, **kwargs)
   4999             # else, only a single dtype is given
   5000             new_data = self._data.astype(dtype=dtype, copy=copy, e
rrors=errors,
-> 5001                                          **kwargs)
   5002             return self._constructor(new_data).__finalize__(self)
   5003

~\Anaconda3\lib\site-packages\pandas\core\internals.py in astype(self, dty
pe, **kwargs)
   3712
   3713     def astype(self, dtype, **kwargs):
-> 3714         return self.apply('astype', dtype=dtype, **kwargs)
   3715
   3716     def convert(self, **kwargs):

~\Anaconda3\lib\site-packages\pandas\core\internals.py in apply(self, f, a
xes, filter, do_integrity_check, consolidate, **kwargs)
   3579
   3580             kwargs['mgr'] = self
-> 3581             applied = getattr(b, f)(**kwargs)
   3582             result_blocks = _extend_blocks(applied, result_blocks)
   3583

~\Anaconda3\lib\site-packages\pandas\core\internals.py in astype(self, dty
pe, copy, errors, values, **kwargs)
    573     def astype(self, dtype, copy=False, errors='raise', values=Non
e, **kwargs):
    574         return self._astype(dtype, copy=copy, errors=errors, value
s=values,
--> 575                             **kwargs)
    576
    577     def _astype(self, dtype, copy=False, errors='raise', values=No
ne,

~\Anaconda3\lib\site-packages\pandas\core\internals.py in _astype(self, dt
ype, copy, errors, values, klass, mgr, **kwargs)
    662
    663                 # _astype_nansafe works fine with 1-d only
--> 664                 values = astype_nansafe(values.ravel(), dtype, cop
y=True)
    665                 values = values.reshape(self.shape)
    666

~\Anaconda3\lib\site-packages\pandas\core\dtypes\cast.py in astype_nansafe
```

```
(arr, dtype, copy)
    707         # work around NumPy brokenness, #1987
    708         if np.issubdtype(dtype.type, np.integer):
--> 709             return lib.astype_intsafe(arr.ravel(), dtype).reshape(
arr.shape)
    710
    711         # if we have a datetime/timedelta array of objects

pandas\_libs\lib.pyx in pandas._libs.lib.astype_intsafe()

pandas/_libs/src\util.pxd in util.set_value_at_unsafe()

~\Anaconda3\lib\site-packages\sympy\core\expr.py in __int__(self)
    222         from sympy import Dummy
    223         if not self.is_number:
--> 224             raise TypeError("can't convert symbols to int")
    225         r = self.round(2)
    226         if not r.is_Number:

TypeError: can't convert symbols to int
```

In [3]:

```
df
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-3-00cf07b74dcd> in <module>
----> 1 df

NameError: name 'df' is not defined
```

In [ ]:

```

```

In [63]:

```

```

In [97]:

```

```

In [ ]:

```

```

In [93]:

```

```

Out[93]:

```
9174
```

In [113]:

```
```

In [ ]:

```
```

In [115]:

```
```

In [2]:

```
z1
```

```
---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-2-8fdf29504afa> in <module>
----> 1 z1

NameError: name 'z1' is not defined
```

In [137]:

```
x = pd.concat([y1, z1], axis = 1)
```

In [146]:

```
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
```