

In []:

1

executed in 11.6s, finished 16:21:01 2020-10-07

In [1]:

```

1  from GlucoCheck.glucoCheck import glucoCheckOps
2  import pandas as pd
3  import random
4  import numpy as np
5  from tqdm.auto import tqdm
6
7  from scipy import stats
8
9  import random
10 import re
11 from dateutil.parser import parse
12
13 import warnings
14 warnings.filterwarnings('ignore')
15
16 import os
17

```

executed in 12.8s, finished 12:22:58 2020-10-28

Using TensorFlow backend.

In [2]:

```

1  ▼ def createGap(df,start,end):
2      """
3      Creating a Gap
4  ▼    input:
5          start: seed
6          end: seed + gap
7  ▼    output:
8          df: dataframe with index => DisplayTime value => GlucoseValue
9      """
10
11     #df = readData()
12     l = len(df.index)
13  ▼    if end>l:
14         end = l
15
16  ▼    for i in range(start,end):
17         df['GlucoseValue'][i]=float("NaN")
18
19     return df

```

executed in 43ms, finished 12:22:58 2020-10-28

```
In [3]: 1 ▾ #Extract Data
2 data = pd.read_csv("~/Desktop/NCSA_genomics/Python - notebooks/Data/O
3 data = data[data['subjectId']=='OD552']
4 data = data.reset_index(drop=True)
5 data
```

executed in 524ms, finished 12:22:58 2020-10-28

Out[3]:

	subjectId	Display Time	GlucoseValue
0	OD552	4/16/25 11:17	95
1	OD552	4/16/25 11:22	86
2	OD552	4/16/25 11:27	81
3	OD552	4/16/25 11:32	81
4	OD552	4/16/25 11:37	82
...
11439	OD552	6/7/25 16:49	238
11440	OD552	6/7/25 16:54	233
11441	OD552	6/7/25 16:59	229
11442	OD552	6/7/25 17:04	224
11443	OD552	6/7/25 17:09	215

11444 rows × 3 columns

In []:

1

executed in 50ms, finished 15:27:50 2020-10-16

In [85]:

```
1 ▾ #1 week after : 1890, 1974, 2003, 2196, 2378, 2581, 2751, 3190, 3223,
2 #2 weeks after: 3600, 3797, 3828, 3939, 4210, 4353, 4567, 4890, 5102,
3 #3 weeks after: 5500, 5681, 5727, 5893, 5919, 6060, 6143, 6250, 6492,
4 #4 weeks after: 7000, 7296, 7384, 7557, 7572, 7698, 7839, 7934, 8190,
5 #5 weeks after: 8500, 8670, 8792, 8810, 8972, 9009, 9380, 9592, 9715,
6
7 seed_points = [8500, 8670, 8792, 8810, 8972, 9009, 9380, 9592, 9715,
8
9 #
```

executed in 5ms, finished 14:38:47 2020-10-28

In [86]:

```
1 ▾ # obj = glucoCheckOps()
```

executed in 14ms, finished 14:38:48 2020-10-28

In []:

1

executed in 5ms, finished 00:11:30 2020-10-23

```

In [87]: 1
          2 #for gap size 50
          3 ioa_gap50 = list()
          4
          5
          6 ▼ for seed in tqdm(seed_points):
          7     start = seed
          8     end = seed+49
          9
          10     dataWithMissing = data.copy()
          11     dataWithMissing = createGap(dataWithMissing,start,end)
          12
          13     dataBeforeGap = dataWithMissing[:seed]
          14
          15     obj = glucoCheckOps()
          16     # obj.train(dataBeforeGap);
          17     imputed_data = obj.impute(dataWithMissing)
          18
          19     ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue'])
          20
          21     del obj
          22
          23     ioa_gap50.append(ioa)
          24
          25 ioa_gap50

```

executed in 21m 16s, finished 15:00:04 2020-10-28

100%

10/10 [21:15<00:00, 127.60s/it]

Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!
Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!
Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!
Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!
Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!

```
Gap < 50; We use LSTM imputations  
Training Model...
```

```
Model trained successfully!  
Gap < 50; We use LSTM imputations  
Training Model...
```

```
Model trained successfully!  
Gap < 50; We use LSTM imputations  
Training Model...
```

```
Model trained successfully!  
Gap < 50; We use LSTM imputations  
Training Model...
```

```
Model trained successfully!  
Gap < 50; We use LSTM imputations  
Training Model...
```

```
Model trained successfully!
```

```
Out[87]: [0.4525282876143849,  
          0.43283675717411263,  
          0.04536232276409058,  
          0.2659457270564931,  
          0.40656865225495553,  
          0.11171028282332462,  
          0.45395845199918616,  
          0.36090061897957715,  
          0.4812531510872171,  
          0.40038168081770564]
```

```

In [88]: #for gap size 30
ioa_gap30 = list()
3
for seed in tqdm(seed_points):
4     start = seed
5     end = start+29
6
7     dataWithMissing = data.copy()
8     dataWithMissing = createGap(dataWithMissing,start,end)
9
10
11 dataBeforeGap = dataWithMissing[:seed]
12
13 obj = glucoCheckOps()
14 obj.train(dataBeforeGap);
15 imputed_data = obj.impute(dataWithMissing)
16
17 ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue'])[start:
18
19 del obj
20
21 ioa_gap30.append(ioa)
22
23 ioa_gap30
24

```

executed in 21m 14s, finished 15:21:18 2020-10-28

100%

10/10 [21:14<00:00, 127.41s/it]

Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!
Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!
Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!
Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!
Gap < 50; We use LSTM imputations
Training Model...

Model trained successfully!
Gap < 50; We use LSTM imputations

```
Training Model...
```

```
Model trained successfully!  
Gap < 50; We use LSTM imputations  
Training Model...
```

```
Model trained successfully!  
Gap < 50; We use LSTM imputations  
Training Model...
```

```
Model trained successfully!  
Gap < 50; We use LSTM imputations  
Training Model...
```

```
Model trained successfully!  
Gap < 50; We use LSTM imputations  
Training Model...
```

```
Model trained successfully!
```

```
Out[88]: [0.16023330664544122,  
          0.476528311473848,  
          0.4022534172757467,  
          0.4416303261034009,  
          0.3522120856027421,  
          0.24223249775840583,  
          0.37174923725119835,  
          0.11317280565192311,  
          0.9811754685947431,  
          0.4029590948650995]
```

```

In [89]: gap_size 12
gap15 = list()
3
4
seed in tqdm(seed_points):
start = seed
end = start+12
8
dataWithMissing = data.copy()
dataWithMissing = createGap(dataWithMissing,start,end)
11
dataBeforeGap = dataWithMissing[:seed]
13
obj = glucoCheckOps()
15obj.train(dataBeforeGap);
16
imputed_data = obj.impute(dataWithMissing)
17
ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue'])[start:end-
19
del obj
21
ioa_gap15.append(ioa)
23
gap15

```

executed in 38.8s, finished 15:21:57 2020-10-28

100%

10/10 [00:38<00:00, 3.88s/it]

Gap < 15; We use the spline imputations
 Gap < 15; We use the spline imputations
 Gap < 15; We use the spline imputations
 Gap < 15; We use the spline imputations
 Gap < 15; We use the spline imputations
 Gap < 15; We use the spline imputations
 Gap < 15; We use the spline imputations
 Gap < 15; We use the spline imputations
 Gap < 15; We use the spline imputations
 Gap < 15; We use the spline imputations

```

Out[89]: [0.44936624333695474,
0.2787600108413275,
0.8266039102629159,
0.6292373627522039,
0.7897450898959633,
0.9633021796302094,
0.6813647805613532,
0.4464093609910734,
0.9665086427570805,
0.8353953672906045]

```

```

In [90]: size 100
002 = list()
      3
      in todm(seed_points):
      5 seed
      = seed+99
      7
      WithMissing = data.copy()
      WithMissing = createGap(dataWithMissing,start,end)
      10
      BeforeGap = dataWithMissing[:seed]
      12
      = glucoCheckOps()
      14 train(dataBeforeGap);
      15 imputed_data = obj.impute(dataWithMissing)
      16
      sinstance(imputed_data, pd.DataFrame):
      18 ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue'])[start:end-
      19 ioa_gap100.append(ioa)
      : 20 ▾
      ioa_gap100.append(0)
      22
      23
      24
      25
      26
      27

```

executed in 123ms, finished 15:21:57 2020-10-28

100%

10/10 [00:00<00:00, 54.50it/s]

We cannot impute this data
 We cannot impute this data
 We cannot impute this data
 We cannot impute this data
 We cannot impute this data
 We cannot impute this data
 We cannot impute this data
 We cannot impute this data
 We cannot impute this data
 We cannot impute this data

Out[90]: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]


```

In [91]: 1
2
3
4 #for gap size 5
5 ioa_gap5 = list()
6 # fb_gap5 = list()
7 # mad_gap5 = list()
8 # rmse_gap5 = list()
9 # mape_gap5 = list()
10
11 ▼ for seed in tqdm(seed_points):
12     start = seed
13     end = start+4
14
15     dataWithMissing = data.copy()
16     dataWithMissing = createGap(dataWithMissing,start,end)
17
18     dataBeforeGap = dataWithMissing[:seed]
19
20     obj = glucoCheckOps()
21     # obj.train(dataBeforeGap);
22     imputed_data = obj.impute(dataWithMissing)
23
24     ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue']))
25
26     del obj
27
28     ioa_gap5.append(ioa)
29
30
31
32
33 ioa_gap5

```

executed in 148ms, finished 15:21:57 2020-10-28

100%

10/10 [00:00<00:00, 22.90it/s]

Gap < 5; We use the linear imputations
 Gap < 5; We use the linear imputations
 Gap < 5; We use the linear imputations
 Gap < 5; We use the linear imputations
 Gap < 5; We use the linear imputations
 Gap < 5; We use the linear imputations
 Gap < 5; We use the linear imputations
 Gap < 5; We use the linear imputations
 Gap < 5; We use the linear imputations
 Gap < 5; We use the linear imputations

```

Out[91]: [0.7713873077903328,
0.0,
0.4636841151965263,
0.5669934385861005,

```

```
0.7776393427207972,
0.8800968914008883,
0.8042895442359249,
0.6666666666666667,
0.9527091050047378,
0.7858390417322154]
```

In []:

1

In [92]:

```
OA1= pd.DataFrame({'Seeds':seed_points, 'Gap:5':ioa_gap5, 'Gap:12':ioa_gap12,
OA2
3
```

executed in 19ms, finished 15:21:58 2020-10-28

Out[92]:

	Seeds	Gap:5	Gap:12	Gap:30	Gap:50	Gap:100
0	8500	0.771387	0.449366	0.160233	0.452528	0
1	8670	0.000000	0.278760	0.476528	0.432837	0
2	8792	0.463684	0.826604	0.402253	0.045362	0
3	8810	0.566993	0.629237	0.441630	0.265946	0
4	8972	0.777639	0.789745	0.352212	0.406569	0
5	9009	0.880097	0.963302	0.242232	0.111710	0
6	9380	0.804290	0.681365	0.371749	0.453958	0
7	9592	0.666667	0.446409	0.113173	0.360901	0
8	9715	0.952709	0.966509	0.981175	0.481253	0
9	9800	0.785839	0.835395	0.402959	0.400382	0

In []:

1

In [93]:

```
1 IOA.to_csv("~/Desktop/5week.csv")
```

executed in 10ms, finished 15:21:58 2020-10-28

In []:

1

In []:

1

In [94]:

```
1 import matplotlib.pyplot as plt
2 import matplotlib.ticker as ticker
3 import seaborn as sns
```

executed in 7ms, finished 15:21:58 2020-10-28

In [95]:

```
1 ▼ # IOA
```

executed in 5ms, finished 15:21:58 2020-10-28

```
In [96]: 1 gaps = [5,15,30,50,100]
2         ioa = []
3         ioa.append(IOA[ 'Gap:5' ].mean())
4         ioa.append(IOA[ 'Gap:12' ].mean())
5         ioa.append(IOA[ 'Gap:30' ].mean())
6         ioa.append(IOA[ 'Gap:50' ].mean())
7         ioa.append(IOA[ 'Gap:100' ].mean())
8         ioa
```

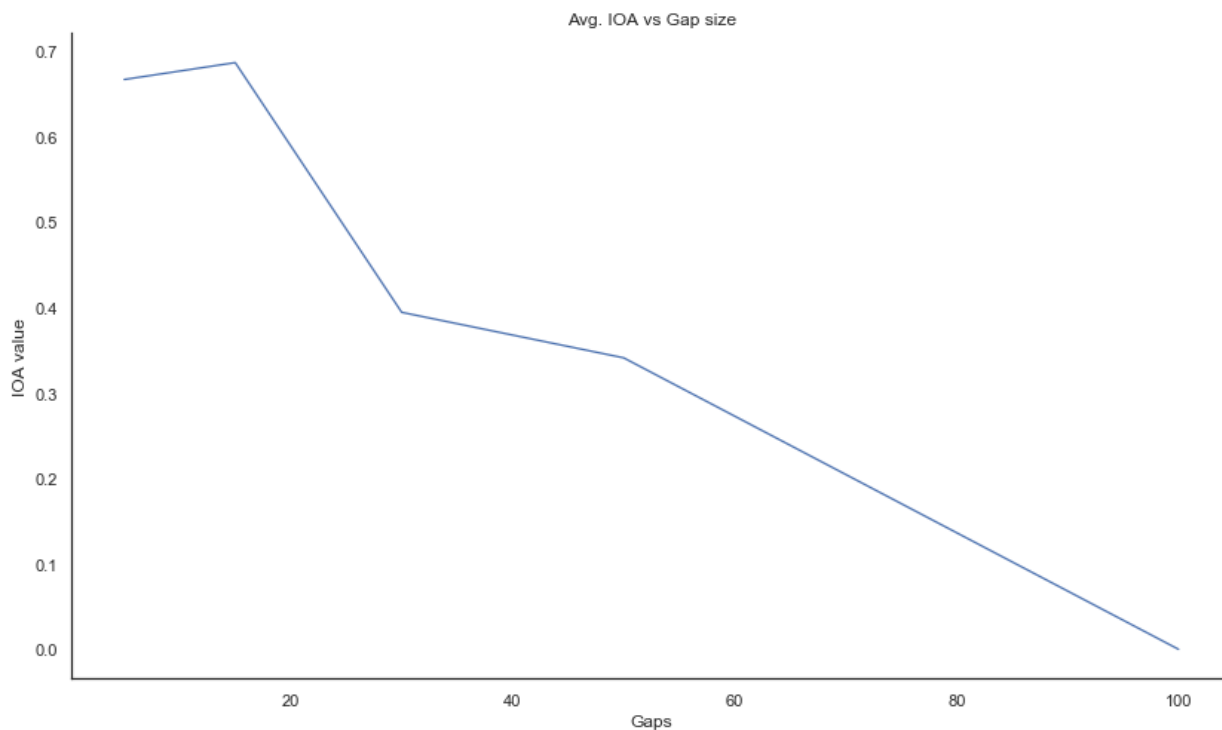
executed in 11ms, finished 15:21:58 2020-10-28

```
Out[96]: [0.666930545333419,
0.6866692948319686,
0.3944146551222549,
0.3411445932571048,
0.0]
```

```
In [97]: 1 plt.figure(figsize=(14,8))
2         plt.title("Avg. IOA vs Gap size")
3         sns.set(style="white")
4         fig = sns.lineplot(x = gaps, y = ioa, palette="tab10", linewidth=1.25)
5         sns.despine()
6
7         fig.set_xlabel('Gaps')
8         fig.set_ylabel('IOA value')
```

executed in 389ms, finished 15:21:58 2020-10-28

```
Out[97]: Text(0, 0.5, 'IOA value')
```



```
In [ ]: 1
```

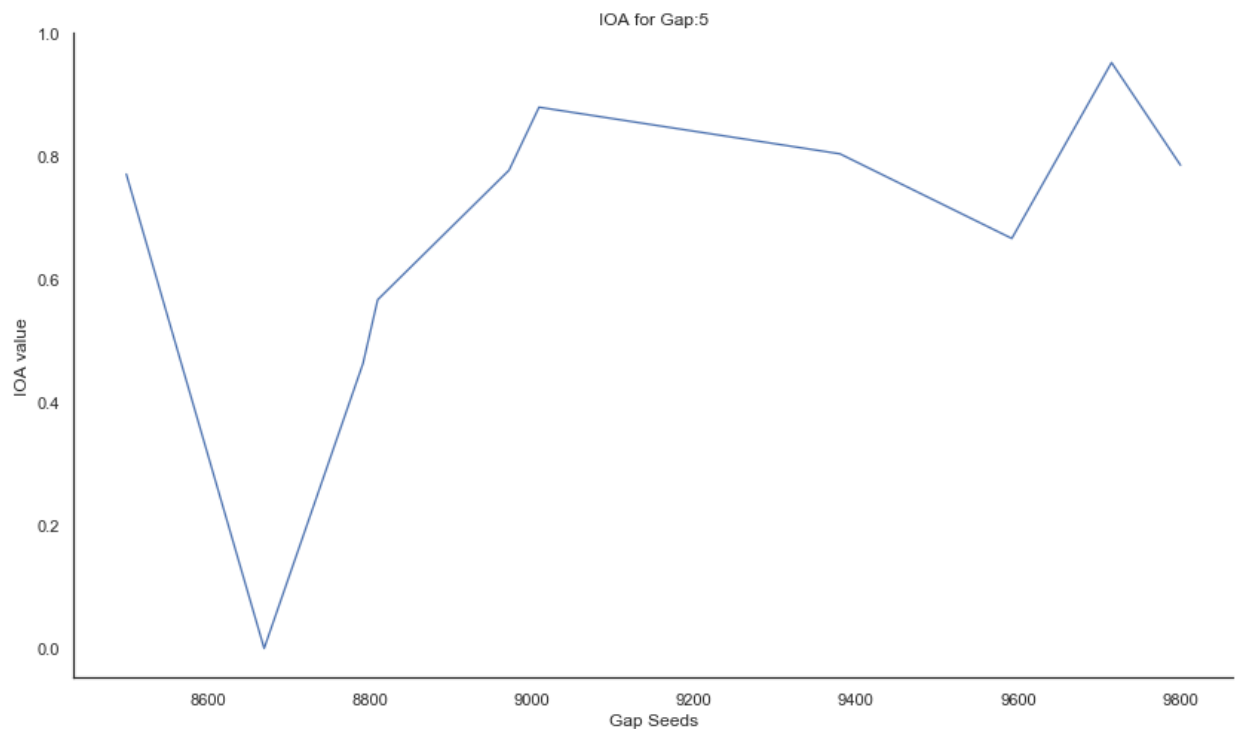
```
In [ ]: 1
```

executed in 8ms, finished 14:05:45 2020-10-14

```
In [98]: 1
2 plt.figure(figsize=(14,8))
3 plt.title("IOA for Gap:5")
4 sns.set(style="white")
5 fig = sns.lineplot(x = seed_points, y = IOA['Gap:5'], data = IOA, pal
6 sns.despine()
7
8 fig.set_xlabel('Gap Seeds')
9 fig.set_ylabel('IOA value')
```

executed in 416ms, finished 15:21:58 2020-10-28

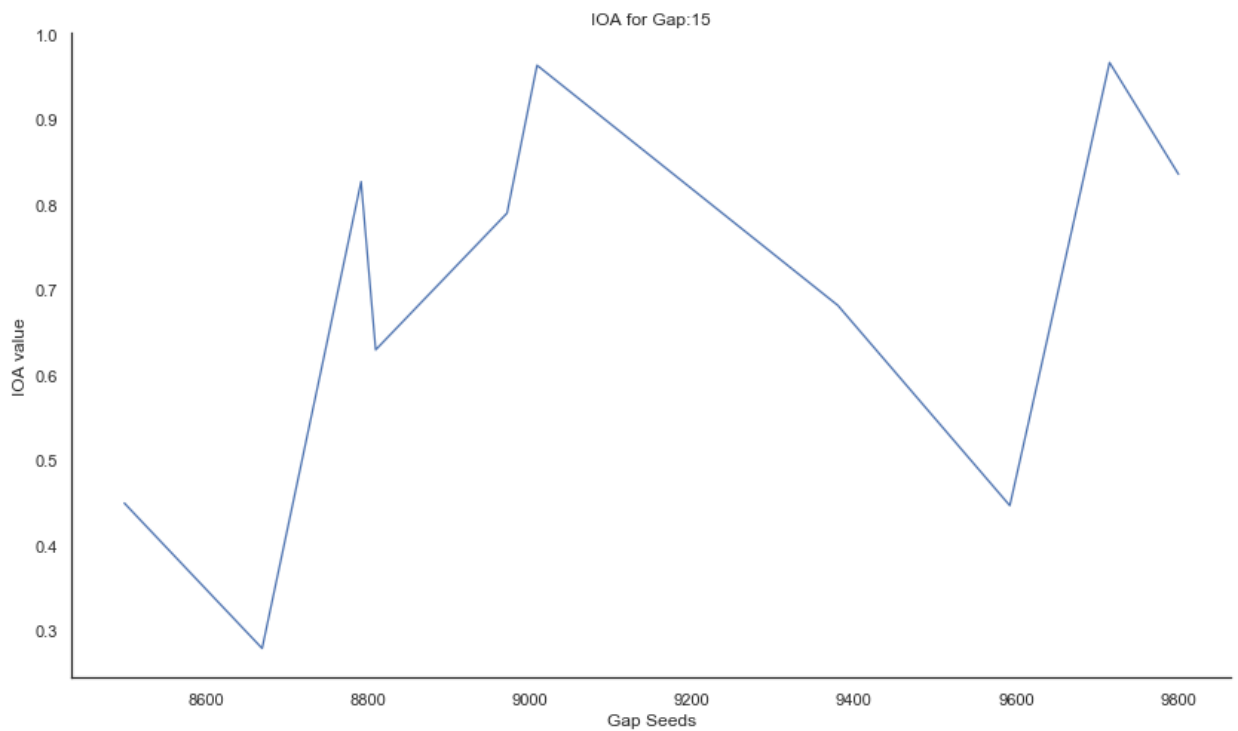
Out[98]: Text(0, 0.5, 'IOA value')



```
In [99]: 1
2 plt.figure(figsize=(14,8))
3 plt.title("IOA for Gap:15")
4 sns.set(style="white")
5 fig = sns.lineplot(x = seed_points, y = IOA['Gap:12'], data = IOA, pa
6 sns.despine()
7
8 fig.set_xlabel('Gap Seeds')
9 fig.set_ylabel('IOA value')
```

executed in 416ms, finished 15:21:59 2020-10-28

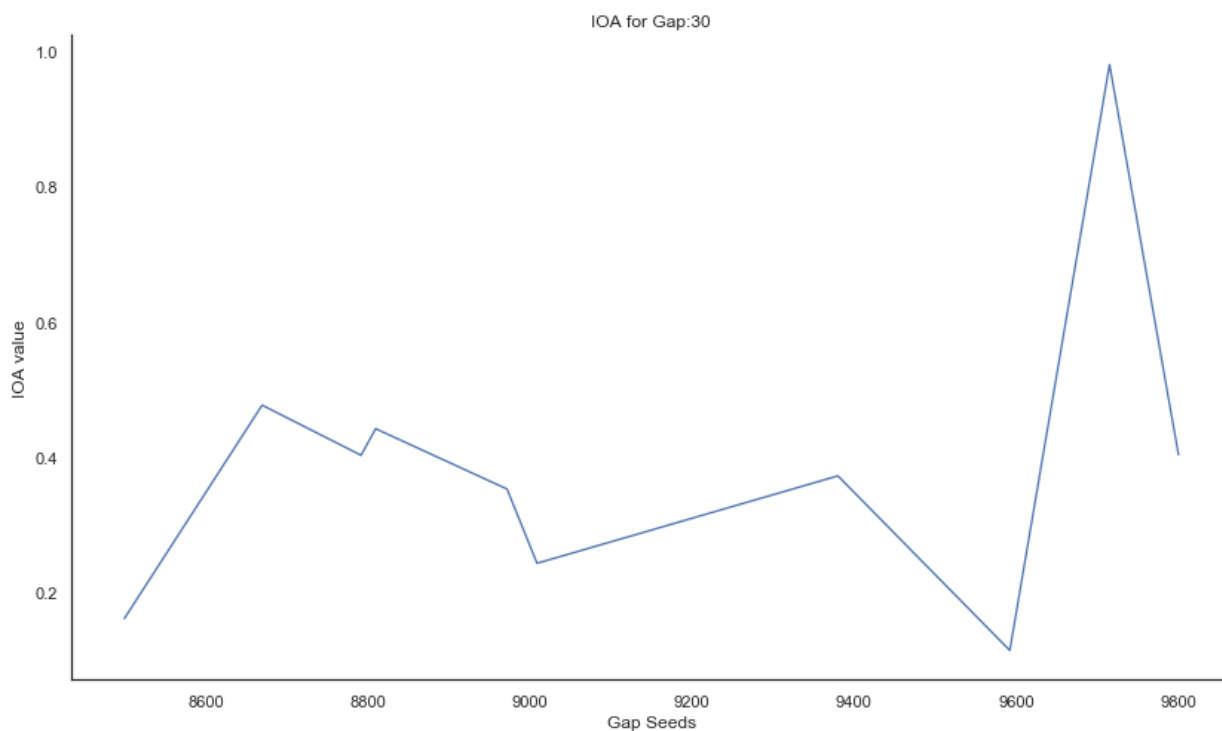
Out[99]: Text(0, 0.5, 'IOA value')



```
In [100]: 1
          2 plt.figure(figsize=(14,8))
          3 plt.title("IOA for Gap:30")
          4 sns.set(style="white")
          5 fig = sns.lineplot(x = seed_points, y = IOA['Gap:30'], data = IOA, pa
          6 sns.despine()
          7
          8 fig.set_xlabel('Gap Seeds')
          9 fig.set_ylabel('IOA value')
```

executed in 427ms, finished 15:21:59 2020-10-28

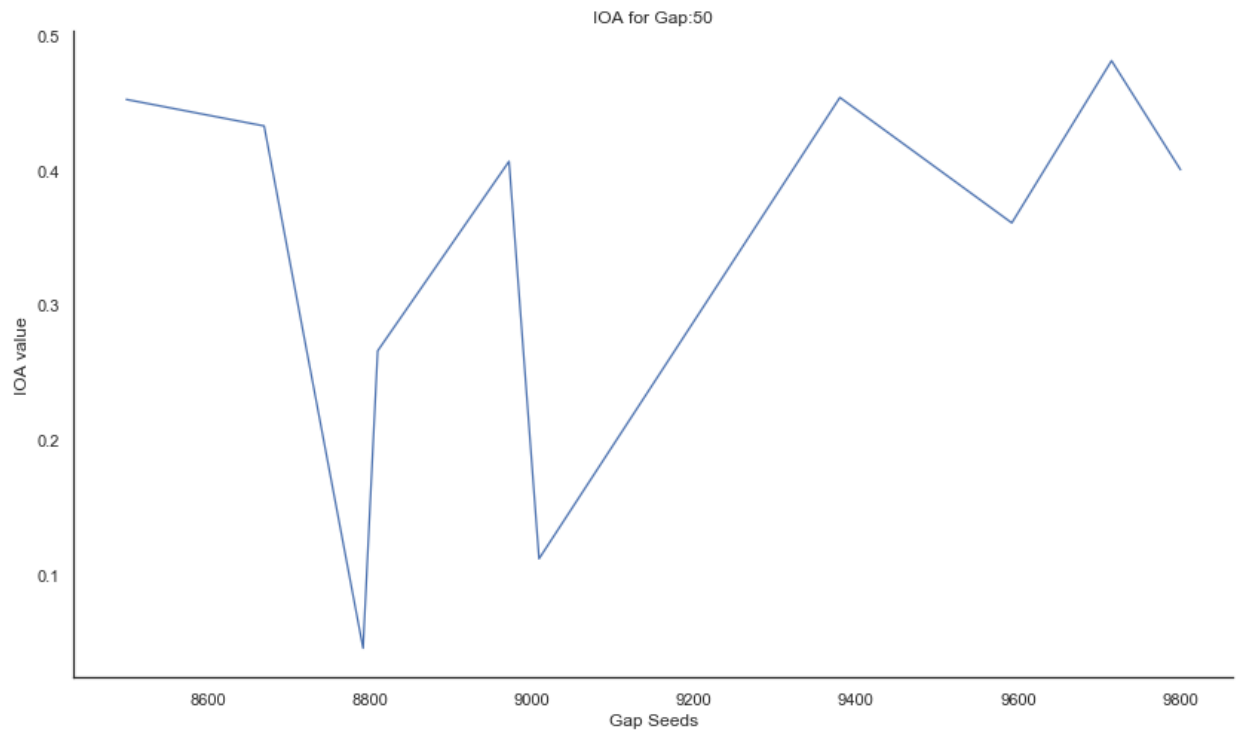
Out[100]: Text(0, 0.5, 'IOA value')



```
In [101]: 1
          2 plt.figure(figsize=(14,8))
          3 plt.title("IOA for Gap:50")
          4 sns.set(style="white")
          5 fig = sns.lineplot(x = seed_points, y = IOA['Gap:50'], data = IOA, pa
          6 sns.despine()
          7
          8 fig.set_xlabel('Gap Seeds')
          9 fig.set_ylabel('IOA value')
```

executed in 388ms, finished 15:22:00 2020-10-28

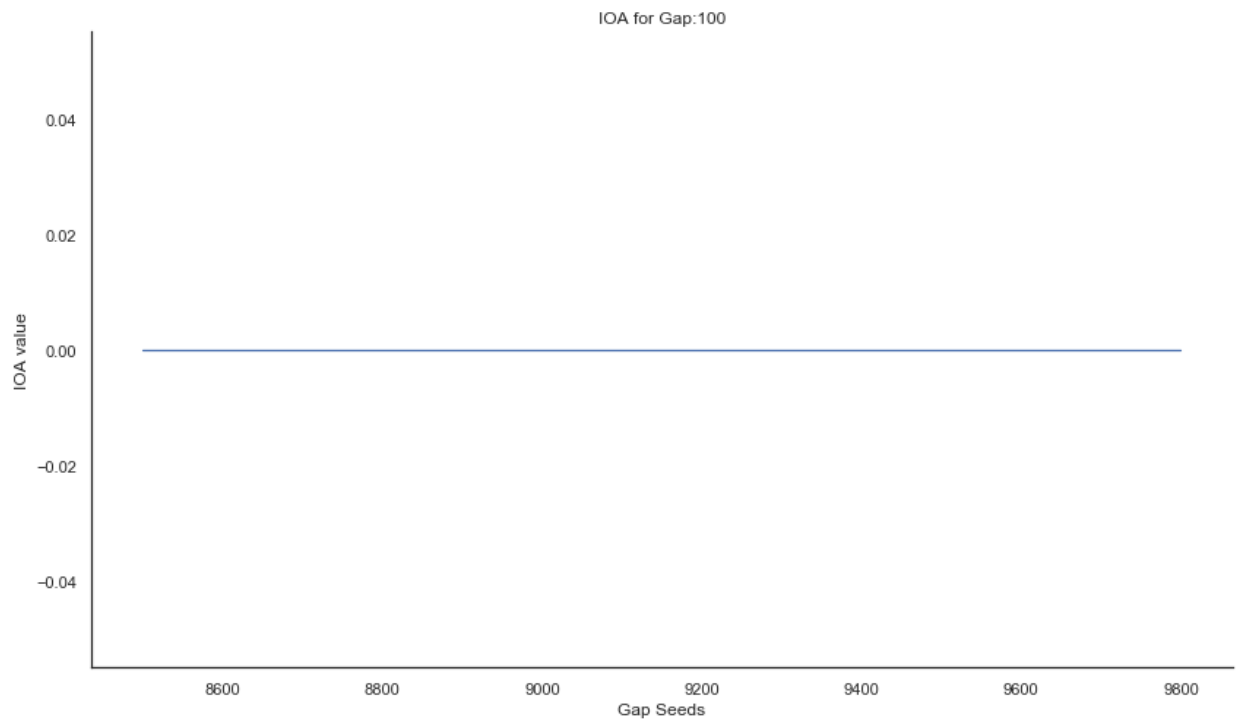
Out[101]: Text(0, 0.5, 'IOA value')



```
In [102]: 1
          2 plt.figure(figsize=(14,8))
          3 plt.title("IOA for Gap:100")
          4 sns.set(style="white")
          5 fig = sns.lineplot(x = seed_points, y = IOA['Gap:100'], data = IOA, p
          6 sns.despine()
          7 fig.set_xlabel('Gap Seeds')
          8 fig.set_ylabel('IOA value')
```

executed in 388ms, finished 15:22:00 2020-10-28

Out[102]: Text(0, 0.5, 'IOA value')



```
1 #####
```

In []:

```
1
```


In [22]:	<div>1 ▼ <code># MAD = pd.DataFrame({'Gap:5':mad_gap5, 'Gap:15':mad_gap15, 'Gap:30':m</code></div> <div>2 <code># MAD</code></div>
executed in 7ms, finished 12:36:57 2020-10-28	
In [23]:	<div>1 ▼ <code># FB = pd.DataFrame({'Gap:5':fb_gap5, 'Gap:15':fb_gap15, 'Gap:30':fb_</code></div> <div>2 <code># FB</code></div>
executed in 7ms, finished 12:36:57 2020-10-28	
In [24]:	<div>1 ▼ <code># RMSE = pd.DataFrame({'Gap:5':rmse_gap5, 'Gap:15':rmse_gap15, 'Gap:3</code></div> <div>2 <code># RMSE</code></div>
executed in 8ms, finished 12:36:57 2020-10-28	
In [25]:	<div>1 ▼ <code># MAPE = pd.DataFrame({'Gap:5':mape_gap5, 'Gap:15':mape_gap15, 'Gap:3</code></div> <div>2 <code># MAPE</code></div>
executed in 55ms, finished 12:36:58 2020-10-28	
In [26]:	<div>1 ▼ <code># IOA.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/M</code></div> <div>2 <code># FB.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/Me</code></div> <div>3 <code># RMSE.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/</code></div> <div>4 <code># MAPE.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/</code></div> <div>5 <code># MAD.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/M</code></div>
executed in 11ms, finished 12:36:58 2020-10-28	
In []:	<div>1</div>