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]:
```
#Extract Data
data = pd.read_csv("~/Desktop/NCSA_genomics/Python - notebooks/Data/O
data = data[data['subjectId']=='OD552']
data = data.reset_index(drop=True)
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 [49]:
```
#1 week after : 1890, 1974, 2003, 2196, 2378, 2581, 2751, 3190, 3223,
#2 weeks after: 3600, 3797, 3828, 3939, 4210, 4353, 4567, 4890, 5102,
#3 weeks after: 5500, 5681, 5727, 5893, 5919, 6060, 6143, 6250, 6492,

seed_points = [5500, 5681, 5727, 5893, 5919, 6060, 6143, 6250, 6492,

#
```

executed in 9ms, finished 13:21:03 2020-10-28

In [50]:
```
# obj = glucoCheckOps()
```

executed in 5ms, finished 13:21:03 2020-10-28

In [ ]:
```
1
```

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

In [51]:

```
#for gap size 50
ioa_gap50 = list()


for seed in tqdm(seed_points):
    start = seed
    end = seed+49

    dataWithMissing = data.copy()
    dataWithMissing = createGap(dataWithMissing,start,end)

    dataBeforeGap = dataWithMissing[:seed]

    obj = glucoCheckOps()
#     obj.train(dataBeforeGap);
    imputed_data = obj.impute(dataWithMissing)

    ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue']

    del obj

    ioa_gap50.append(ioa)

ioa_gap50
```

executed in 14m 24s, finished 13:35:28 2020-10-28

```
100%                                          10/10 [18:19<00:00, 109.99s/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[51]:
```
[0.42895962642412966,
 0.052370633845662984,
 0.2264453211131453,
 0.11252241781363614,
 0.40012795905310305,
 0.1169416401602813,
 0.21116392058170463,
 0.6431638576814509,
 0.7510333268276689,
 0.468311331504886]
```

In [52]:
```
#for gap size 30
ioa_gap30 = list()

for seed in tqdm(seed_points):
start = seed
end = start+29

dataWithMissing = data.copy()
dataWithMissing = createGap(dataWithMissing,start,end)

dataBeforeGap = dataWithMissing[:seed]

obj = glucoCheckOps()
#   obj.train(dataBeforeGap);
imputed_data = obj.impute(dataWithMissing)

ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue'][start:

del obj

ioa_gap30.append(ioa)

ioa_gap30

```

executed in 14m 2s, finished 13:49:30 2020-10-28

```
100%                                                10/10 [14:01<00:00, 84.19s/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[52]:
```
[0.2617663417755447,
 0.7701816702395669,
 0.6815959674235681,
 0.059855718570943406,
 0.27156680426982616,
 0.12865067602738578,
 0.14233508315236076,
 0.4145251335373411,
 0.4742003459031682,
 0.4452977760154444]
```

In [53]:
```
gap_size 12
gap15 = list()


for seed in tqdm(seed_points):
start = seed
end = start+12

dataWithMissing = data.copy()
dataWithMissing = createGap(dataWithMissing,start,end)

dataBeforeGap = dataWithMissing[:seed]

obj = glucoCheckOps()
obj.train(dataBeforeGap);
imputed_data = obj.impute(dataWithMissing)

ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue'][start:end-

del obj

ioa_gap15.append(ioa)

gap15
```

executed in 43.3s, finished 13:50:13 2020-10-28

100% 10/10 [00:43<00:00, 4.33s/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[53]:
```
[0.7177072977733607,
 0.4826115474008241,
 0.9596194714944348,
 0.6971086923247779,
 0.7037980232831377,
 0.13486370271476222,
 0.4447497951243854,
 0.5649919770947314,
 0.7752759763091011,
 0.17160664954152205]
```

In [54]:
```
size = 100
ioa = list()

for seed in tqdm(seed_points):
    start = seed
    end = seed+99

    dataWithMissing = data.copy()
    dataWithMissing = createGap(dataWithMissing,start,end)

    dataBeforeGap = dataWithMissing[:seed]

    obj = GlucoCheckOps()
    obj.train(dataBeforeGap);
    imputed_data = obj.impute(dataWithMissing)

    if isinstance(imputed_data, pd.DataFrame):
        ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue'][start:end-
        ioa_gap100.append(ioa)
    else:
        ioa_gap100.append(0)

obj
```

executed in 250ms, finished 13:50:14 2020-10-28

```
100%                                                10/10 [00:00<00:00, 25.97it/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[54]:
```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

In [55]:

```
#for gap size 5
ioa_gap5 = list()
# fb_gap5 = list()
# mad_gap5 = list()
# rmse_gap5 = list()
# mape_gap5 = list()

for seed in tqdm(seed_points):
    start = seed
    end = start+4

    dataWithMissing = data.copy()
    dataWithMissing = createGap(dataWithMissing,start,end)

    dataBeforeGap = dataWithMissing[:seed]

    obj = glucoCheckOps()
#     obj.train(dataBeforeGap);
    imputed_data = obj.impute(dataWithMissing)

    ioa = obj.index_agreement(np.asarray(imputed_data['GlucoseValue']

    del obj

    ioa_gap5.append(ioa)



ioa_gap5
```

executed in 262ms, finished 13:50:14 2020-10-28

```
100%                                    10/10 [00:01<00:00, 9.98it/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[55]:

```
[0.6797939434788887,
 0.8450704225352111,
 0.6704761904761904,
 0.0,
```

```
0.3230691569914136,
0.34557235421166177,
0.22249690976514302,
0.4457369328563716,
0.5482592291895501,
0.2870813397129156]
```

In [ ]:
```
1
```

In [56]:
```
OA1= pd.DataFrame({'Seeds':seed_points, 'Gap:5':ioa_gap5, 'Gap:12':ioa_gap15
OA2
  3
```

executed in 36ms, finished 13:50:14 2020-10-28

Out[56]:

	Seeds	Gap:5	Gap:12	Gap:30	Gap:50	Gap:100
0	5500	0.679794	0.717707	0.261766	0.428960	0
1	5681	0.845070	0.482612	0.770182	0.052371	0
2	5727	0.670476	0.959619	0.681596	0.226445	0
3	5893	0.000000	0.697109	0.059856	0.112522	0
4	5919	0.323069	0.703798	0.271567	0.400128	0
5	6060	0.345572	0.134864	0.128651	0.116942	0
6	6143	0.222497	0.444750	0.142335	0.211164	0
7	6250	0.445737	0.564992	0.414525	0.643164	0
8	6492	0.548259	0.775276	0.474200	0.751033	0
9	6600	0.287081	0.171607	0.445298	0.468311	0

In [ ]:
```
1
```

In [57]:
```
1 IOA.to_csv("~/Desktop/3week.csv")
```

executed in 12ms, finished 13:50:14 2020-10-28

In [ ]:
```
1
```

In [ ]:
```
1
```

In [58]:
```
1 import matplotlib.pyplot as plt
2 import matplotlib.ticker as ticker
3 import seaborn as sns
```

executed in 6ms, finished 13:50:14 2020-10-28

In [59]:
```
1 ▾ # IOA
```

executed in 8ms, finished 13:50:14 2020-10-28

In [60]:
```
gaps = [5,15,30,50,100]
ioa = []
ioa.append(IOA['Gap:5'].mean())
ioa.append(IOA['Gap:12'].mean())
ioa.append(IOA['Gap:30'].mean())
ioa.append(IOA['Gap:50'].mean())
ioa.append(IOA['Gap:100'].mean())
ioa
```

executed in 23ms, finished 13:50:14 2020-10-28

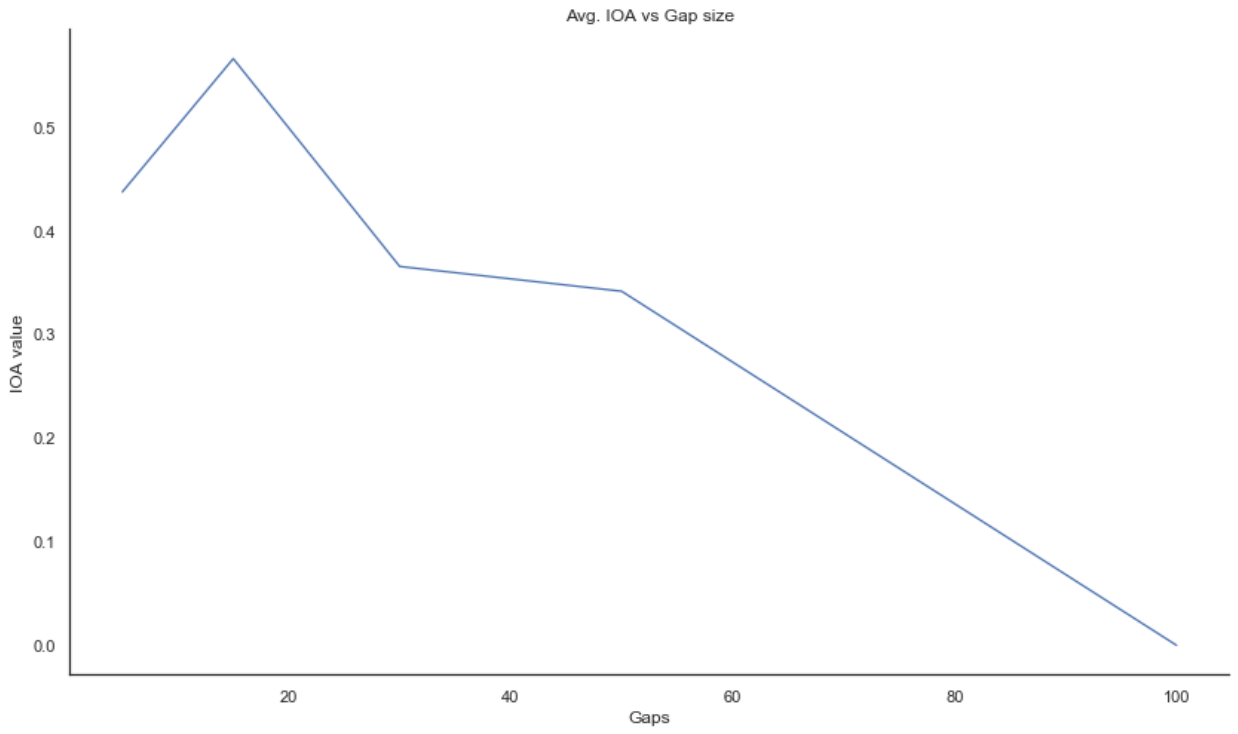
Out[60]:
```
[0.43675564792173455,
 0.5652333133061036,
 0.36499755169151493,
 0.3411040035005669,
 0.0]
```

In [61]:
```
plt.figure(figsize=(14,8))
plt.title("Avg. IOA vs Gap size")
sns.set(style="white")
fig = sns.lineplot(x = gaps, y = ioa, palette="tab10", linewidth=1.25
sns.despine()

fig.set_xlabel('Gaps')
fig.set_ylabel('IOA value')
```

executed in 766ms, finished 13:50:15 2020-10-28

Out[61]: Text(0, 0.5, 'IOA value')



In [ ]:
```
1
```

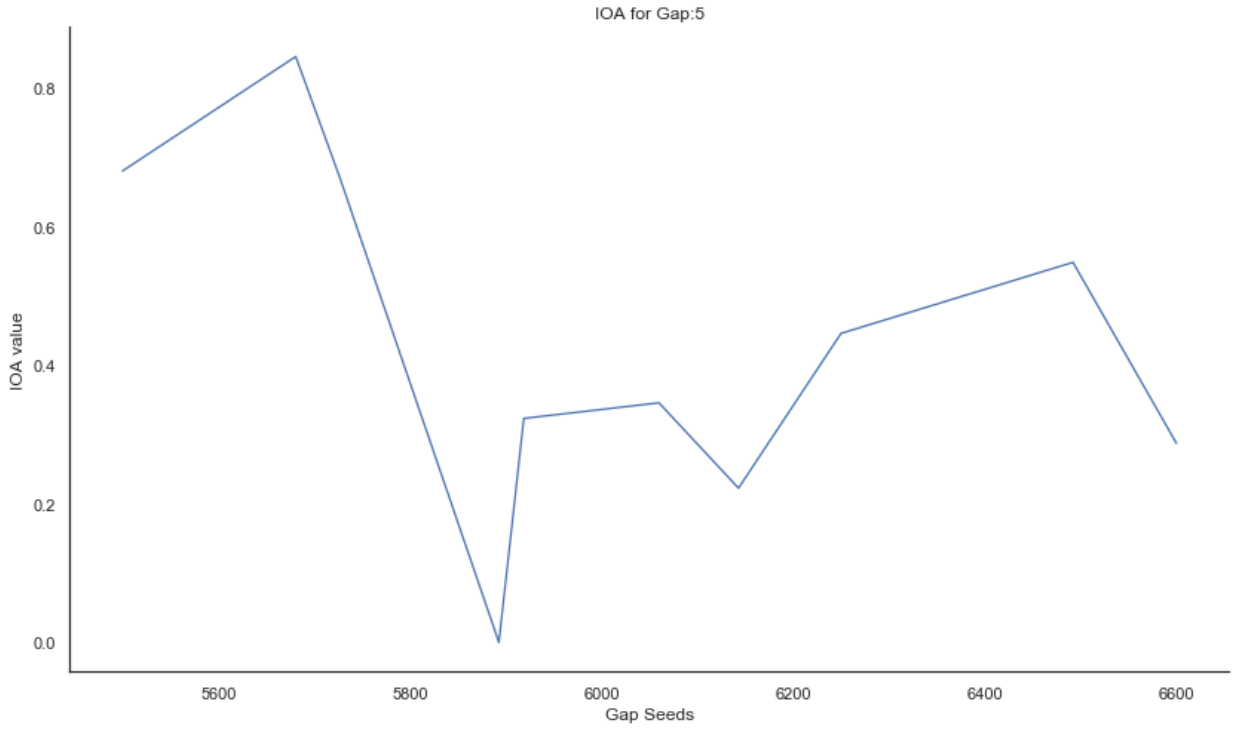
In [ ]:
```
1
```

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

In [62]:
```
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 716ms, finished 13:50:15 2020-10-28

Out[62]: Text(0, 0.5, 'IOA value')

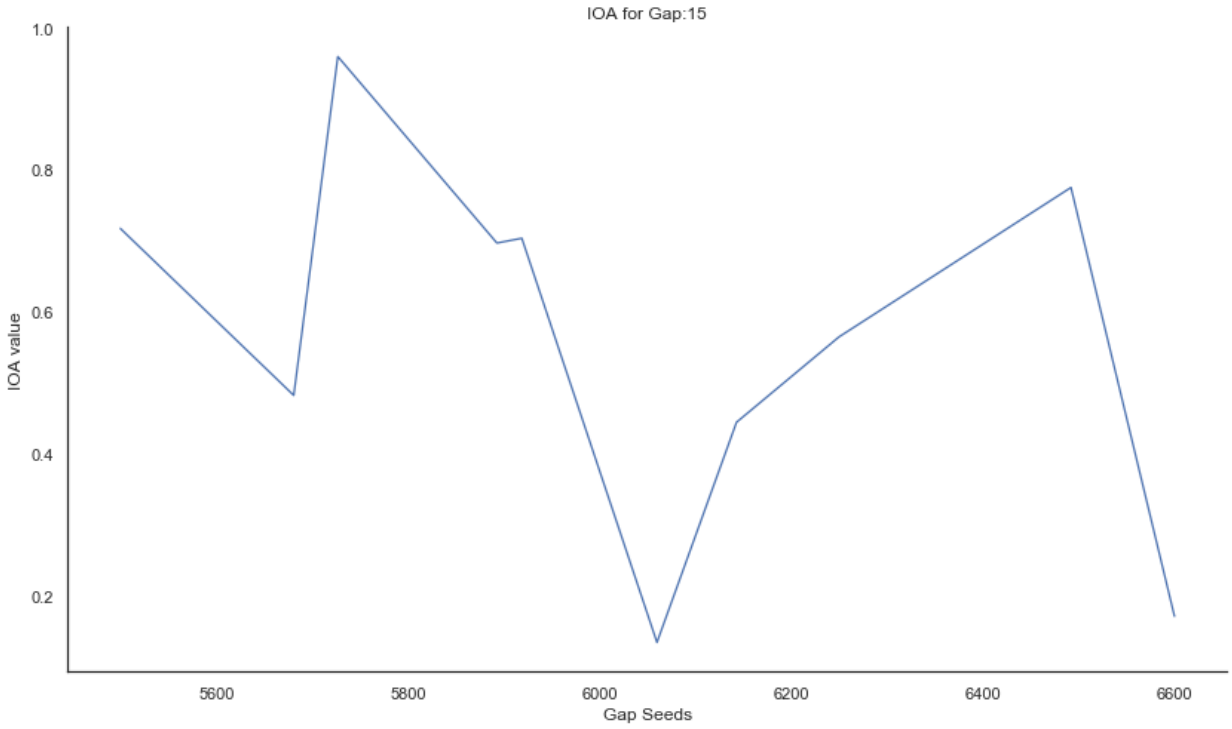


In [63]:

```
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 860ms, finished 13:50:16 2020-10-28

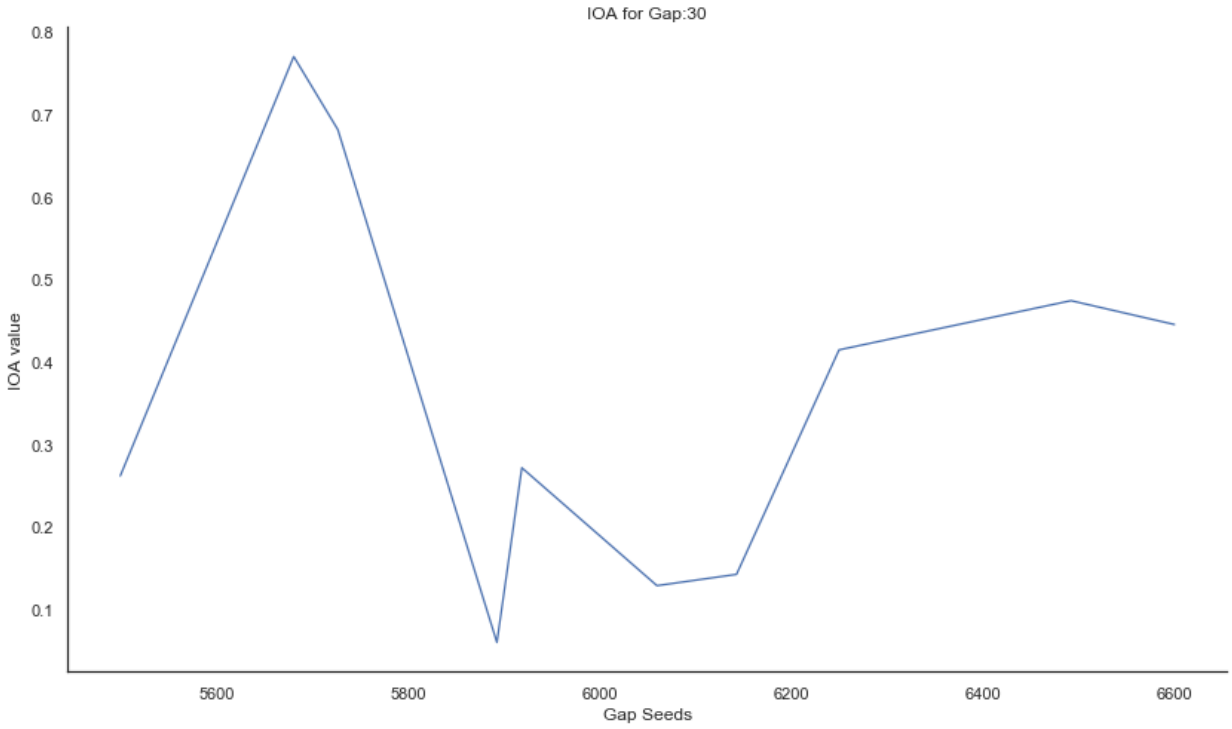
Out[63]: Text(0, 0.5, 'IOA value')



In [64]:
```
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 1.01s, finished 13:50:17 2020-10-28

Out[64]: Text(0, 0.5, 'IOA value')

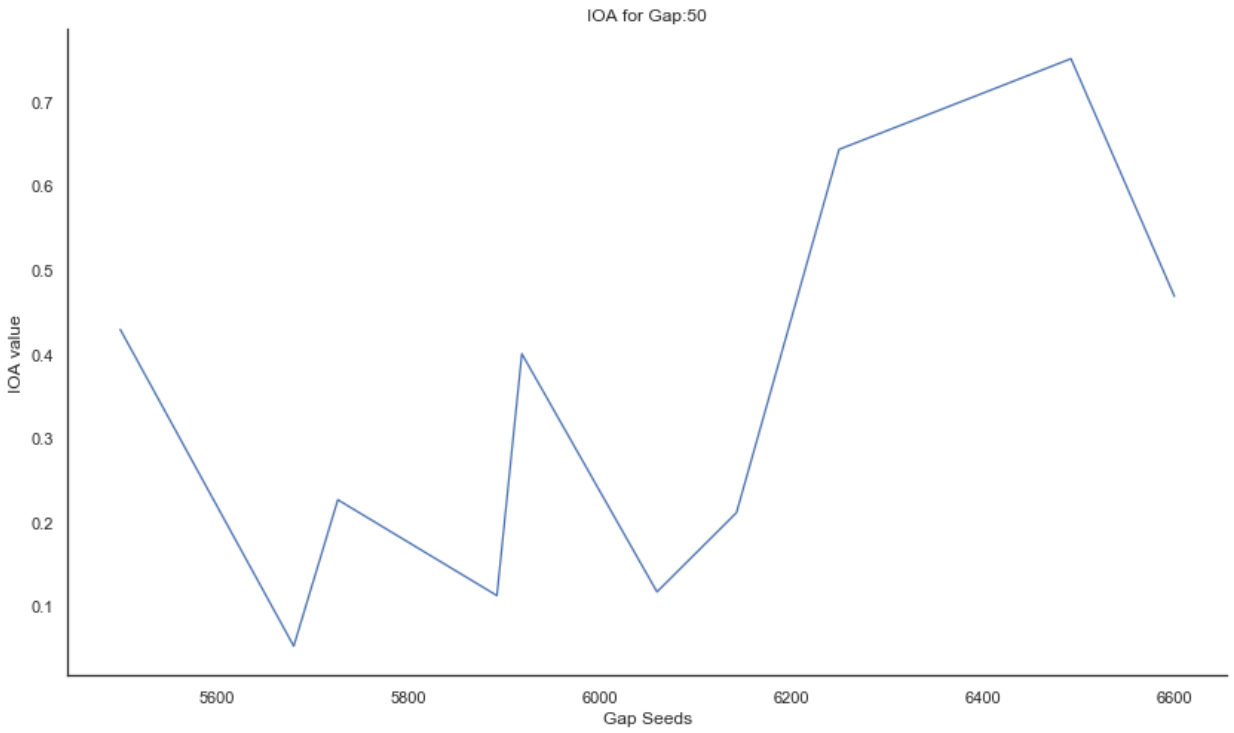


In [65]:

```
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 900ms, finished 13:50:18 2020-10-28

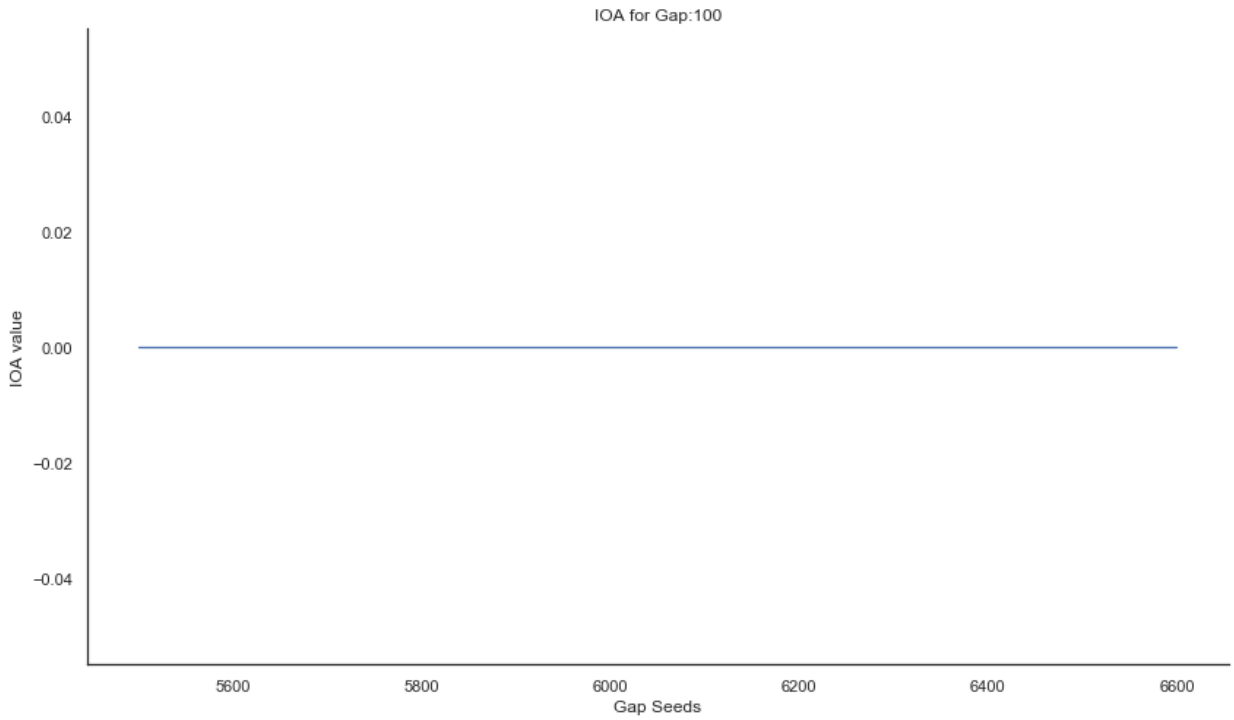
Out[65]: Text(0, 0.5, 'IOA value')



In [66]:
```
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 689ms, finished 13:50:19 2020-10-28

Out[66]: Text(0, 0.5, 'IOA value')



In [ ]:
```
1
```

In [ ]:
```
1
```

In [22]:
```
# MAD = pd.DataFrame({'Gap:5':mad_gap5, 'Gap:15':mad_gap15, 'Gap:30':m
# MAD
```
executed in 7ms, finished 12:36:57 2020-10-28

In [23]:
```
# FB = pd.DataFrame({'Gap:5':fb_gap5, 'Gap:15':fb_gap15, 'Gap:30':fb_
# FB
```
executed in 7ms, finished 12:36:57 2020-10-28

In [24]:
```
# RMSE = pd.DataFrame({'Gap:5':rmse_gap5, 'Gap:15':rmse_gap15, 'Gap:3
# RMSE
```
executed in 8ms, finished 12:36:57 2020-10-28

In [25]:
```
# MAPE = pd.DataFrame({'Gap:5':mape_gap5, 'Gap:15':mape_gap15, 'Gap:3
# MAPE
```
executed in 55ms, finished 12:36:58 2020-10-28

In [26]:
```
# IOA.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/M
# FB.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/Me
# RMSE.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/
# MAPE.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/
# MAD.to_csv("~/Desktop/NCSA_genomics/Python - notebooks/GlucoCheck/M
```
executed in 11ms, finished 12:36:58 2020-10-28

In [ ]:
```

```