EdgeAnglesV4

June 23, 2021

1 Edge angle experiments

1.1 Imports

[1]:
```
import numpy as np
import pymc3 as pm
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
from scipy.stats import norm
from sklearn.preprocessing import LabelEncoder
from theano import tensor as tt
from sys import exit
import pickle
import arviz
import re
```

```
WARNING (theano.tensor.blas): Using NumPy C-API based implementation for BLAS
functions.
/home/konstantin/.local/lib/python3.6/site-packages/numba/core/errors.py:144:
UserWarning: Insufficiently recent colorama version found. Numba requires
colorama >= 0.3.9
  warnings.warn(msg)
```

[2]:
```
writeOut = False
```

[3]:
```
path = './Plots_New/'
```

1.2 Plot settings

[4]:
```
widthMM = 190
widthInch = widthMM / 25.4
ratio = 0.66
heigthInch = ratio*widthInch
aspect = widthInch / heigthInch
```

```
[5]: SMALL_SIZE = 8
MEDIUM_SIZE = 10
BIGGER_SIZE = 12

plt.rc('font', size=SMALL_SIZE)          # controls default text sizes
plt.rc('axes', titlesize=SMALL_SIZE)     # fontsize of the axes title
plt.rc('axes', labelsize=MEDIUM_SIZE)    # fontsize of the x and y labels
plt.rc('xtick', labelsize=SMALL_SIZE)    # fontsize of the tick labels
plt.rc('ytick', labelsize=SMALL_SIZE)    # fontsize of the tick labels
plt.rc('legend', fontsize=SMALL_SIZE)    # legend fontsize
plt.rc('figure', titlesize=BIGGER_SIZE)  # fontsize of the figure title
sns.set_style("ticks")
```

1.3 Load data

Read in all excel sheets at once into one big table.

```
[6]: df = pd.concat([pd.read_excel("data/SEC-NUM-10_SEC-R-15_EAP-flake_1_E1.xlsx"),␣
 ↪pd.read_excel("data/SEC-NUM-10_SEC-R-15_WEM-60_1_E1.xlsx"),pd.
 ↪read_excel("data/SEC-NUM-10_SEC-R-15_BU-072_1_E1.xlsx")], axis=0)
df.reset_index().dropna("columns").drop('index', 1)
```

```
[6]:                          section  angle_number  steps  dist_intersection  \
0     EAP-flake_1_E1_RE_SEC-01_local             1    0.2                0.2
1     EAP-flake_1_E1_RE_SEC-01_local             2    0.2                0.4
2     EAP-flake_1_E1_RE_SEC-01_local             3    0.2                0.6
3     EAP-flake_1_E1_RE_SEC-01_local             4    0.2                0.8
4     EAP-flake_1_E1_RE_SEC-01_local             5    0.2                1.0
…                                …            …      …                  …
4302     BU-072_1_E1_RE_SEC-10_local            10    1.0               10.0
4303     BU-072_1_E1_RE_SEC-10_local            11    1.0               11.0
4304     BU-072_1_E1_RE_SEC-10_local            12    1.0               12.0
4305     BU-072_1_E1_RE_SEC-10_local            13    1.0               13.0
4306     BU-072_1_E1_RE_SEC-10_local            14    1.0               14.0

      segment_length  3points  2lines  best_fit
0                0.5    116.6   111.3     107.4
1                0.5    106.0    95.6      98.9
2                0.5    105.4    80.1      79.6
3                0.5     93.0    53.4      55.9
4                0.5     84.6    52.3      50.8
…                …      …       …
4302             1.0     61.6    27.0      27.7
4303             1.0     57.1     9.6       9.9
4304             1.0     53.1    13.7      13.5
4305             1.0     49.4     8.6       9.8
```

```
4306              1.0     46.3    24.7     150.4

[4307 rows x 8 columns]
```

Add extra columns for location and object.

[7]:
```
df['location'] = df.apply(lambda r: int(re.findall(r'\d+', r.
 ↪section)[-1]),axis=1)
df['object'] = df.apply(lambda r: r.section.split('-')[0],axis=1)
```

[8]:
```
df
```

[8]:
```
                             section  angle_number  steps  dist_intersection  \
0     EAP-flake_1_E1_RE_SEC-01_local             1    0.2                0.2
1     EAP-flake_1_E1_RE_SEC-01_local             2    0.2                0.4
2     EAP-flake_1_E1_RE_SEC-01_local             3    0.2                0.6
3     EAP-flake_1_E1_RE_SEC-01_local             4    0.2                0.8
4     EAP-flake_1_E1_RE_SEC-01_local             5    0.2                1.0
…                                …            …      …                  …
1411     BU-072_1_E1_RE_SEC-10_local            10    1.0               10.0
1412     BU-072_1_E1_RE_SEC-10_local            11    1.0               11.0
1413     BU-072_1_E1_RE_SEC-10_local            12    1.0               12.0
1414     BU-072_1_E1_RE_SEC-10_local            13    1.0               13.0
1415     BU-072_1_E1_RE_SEC-10_local            14    1.0               14.0

      segment_length  3points  2lines  best_fit  location object
0                0.5    116.6   111.3     107.4         1    EAP
1                0.5    106.0    95.6      98.9         1    EAP
2                0.5    105.4    80.1      79.6         1    EAP
3                0.5     93.0    53.4      55.9         1    EAP
4                0.5     84.6    52.3      50.8         1    EAP
…                  …       …       …        …       …
1411             1.0     61.6    27.0      27.7        10     BU
1412             1.0     57.1     9.6       9.9        10     BU
1413             1.0     53.1    13.7      13.5        10     BU
1414             1.0     49.4     8.6       9.8        10     BU
1415             1.0     46.3    24.7     150.4        10     BU

[4307 rows x 10 columns]
```

Convert three method column to columns 'edge_angle' and 'method'.

[9]:
```
otherCols =␣
 ↪['section','angle_number','steps','dist_intersection','segment_length','location','object']
df = pd.melt(df, value_vars=['3points', '2lines','best_fit'],␣
 ↪id_vars=otherCols, var_name='method',value_name='edge_angle')
```

[10]:
```python
df
```

[10]:
```
                              section  angle_number  steps  dist_intersection  \
0      EAP-flake_1_E1_RE_SEC-01_local             1    0.2                0.2
1      EAP-flake_1_E1_RE_SEC-01_local             2    0.2                0.4
2      EAP-flake_1_E1_RE_SEC-01_local             3    0.2                0.6
3      EAP-flake_1_E1_RE_SEC-01_local             4    0.2                0.8
4      EAP-flake_1_E1_RE_SEC-01_local             5    0.2                1.0
…                                 …           …    …                 …
12916     BU-072_1_E1_RE_SEC-10_local            10    1.0               10.0
12917     BU-072_1_E1_RE_SEC-10_local            11    1.0               11.0
12918     BU-072_1_E1_RE_SEC-10_local            12    1.0               12.0
12919     BU-072_1_E1_RE_SEC-10_local            13    1.0               13.0
12920     BU-072_1_E1_RE_SEC-10_local            14    1.0               14.0

       segment_length  location object    method  edge_angle
0                 0.5         1    EAP   3points       116.6
1                 0.5         1    EAP   3points       106.0
2                 0.5         1    EAP   3points       105.4
3                 0.5         1    EAP   3points        93.0
4                 0.5         1    EAP   3points        84.6
…                 …       …    …       …         …
12916             1.0        10     BU  best_fit        27.7
12917             1.0        10     BU  best_fit         9.9
12918             1.0        10     BU  best_fit        13.5
12919             1.0        10     BU  best_fit         9.8
12920             1.0        10     BU  best_fit       150.4

[12921 rows x 9 columns]
```

Use the filter criterion dist_intersection' only 2.0mm, 5.0mm und 10.0mm and 0.5mm 'segment_length' each.

[11]:
```python
df = df[(df.segment_length == 0.5) & ((df.dist_intersection == 2.0)  | (df.
↪dist_intersection == 5.0) | (df.dist_intersection == 10.0) ) ]
```

Furthermore I renamed 'location' to 'sectionNumber'.

[12]:
```python
df = df.rename(columns={"location": "sectionNumber"})
```

[13]:
```python
df
```

[13]:
```
                              section  angle_number  steps  dist_intersection  \
9      EAP-flake_1_E1_RE_SEC-01_local            10    0.2                2.0
24     EAP-flake_1_E1_RE_SEC-01_local            25    0.2                5.0
49     EAP-flake_1_E1_RE_SEC-01_local            50    0.2               10.0
55     EAP-flake_1_E1_RE_SEC-01_local             4    0.5                2.0
61     EAP-flake_1_E1_RE_SEC-01_local            10    0.5                5.0
```

```
…                                     …           …    …            …
12858     BU-072_1_E1_RE_SEC-10_local             10    0.5                5.0
12868     BU-072_1_E1_RE_SEC-10_local             20    0.5               10.0
12894     BU-072_1_E1_RE_SEC-10_local              2    1.0                2.0
12897     BU-072_1_E1_RE_SEC-10_local              5    1.0                5.0
12902     BU-072_1_E1_RE_SEC-10_local             10    1.0               10.0

       segment_length  sectionNumber object   method  edge_angle
9                 0.5              1    EAP  3points        63.7
24                0.5              1    EAP  3points        37.7
49                0.5              1    EAP  3points         8.1
55                0.5              1    EAP  3points        63.7
61                0.5              1    EAP  3points        37.7
…                 …            …    …      …      …
12858             0.5             10     BU  best_fit        72.5
12868             0.5             10     BU  best_fit        33.6
12894             0.5             10     BU  best_fit        81.5
12897             0.5             10     BU  best_fit        72.5
12902             0.5             10     BU  best_fit        33.6

[801 rows x 9 columns]
```

1.3.1 Check for triple values

Read in the raw data again.

[14]:
```python
raw = pd.read_excel("data/SEC-NUM-10_SEC-R-15_BU-072_1_E1.xlsx")
raw
```

[14]:
```
                          section  angle_number  steps  dist_intersection  \
0     BU-072_1_E1_RE_SEC-01_local             1    0.2                0.2
1     BU-072_1_E1_RE_SEC-01_local             2    0.2                0.4
2     BU-072_1_E1_RE_SEC-01_local             3    0.2                0.6
3     BU-072_1_E1_RE_SEC-01_local             4    0.2                0.8
4     BU-072_1_E1_RE_SEC-01_local             5    0.2                1.0
…                               …           …     …                 …
1411  BU-072_1_E1_RE_SEC-10_local            10    1.0               10.0
1412  BU-072_1_E1_RE_SEC-10_local            11    1.0               11.0
1413  BU-072_1_E1_RE_SEC-10_local            12    1.0               12.0
1414  BU-072_1_E1_RE_SEC-10_local            13    1.0               13.0
1415  BU-072_1_E1_RE_SEC-10_local            14    1.0               14.0

      segment_length  3points  2lines  best_fit
0                0.5     76.3    68.6      95.9
1                0.5     63.6    46.2      46.1
2                0.5     55.6    39.9      39.3
```

```
3                 0.5     50.9    35.5      37.1
4                 0.5     47.3    34.3      33.2
…               …      …      …       …
1411              1.0     61.6    27.0      27.7
1412              1.0     57.1     9.6       9.9
1413              1.0     53.1    13.7      13.5
1414              1.0     49.4     8.6       9.8
1415              1.0     46.3    24.7     150.4

[1416 rows x 8 columns]
```

The values 121.4 and 26.8 are indeed measured several times, however always with different settings.

[15]:
```
raw[(raw.best_fit == 121.4) | (raw.best_fit == 26.8) ]
```

[15]:
```
                         section  angle_number  steps  dist_intersection  \
139  BU-072_1_E1_RE_SEC-01_local             5    1.0                5.0
503  BU-072_1_E1_RE_SEC-04_local            61    0.2               12.2

     segment_length  3points  2lines  best_fit
139             1.0     34.2    25.9      26.8
503             0.5     38.2    29.9      26.8
```

An explicit query for duplicate lines (i.e. all entries the same) yields nothing.

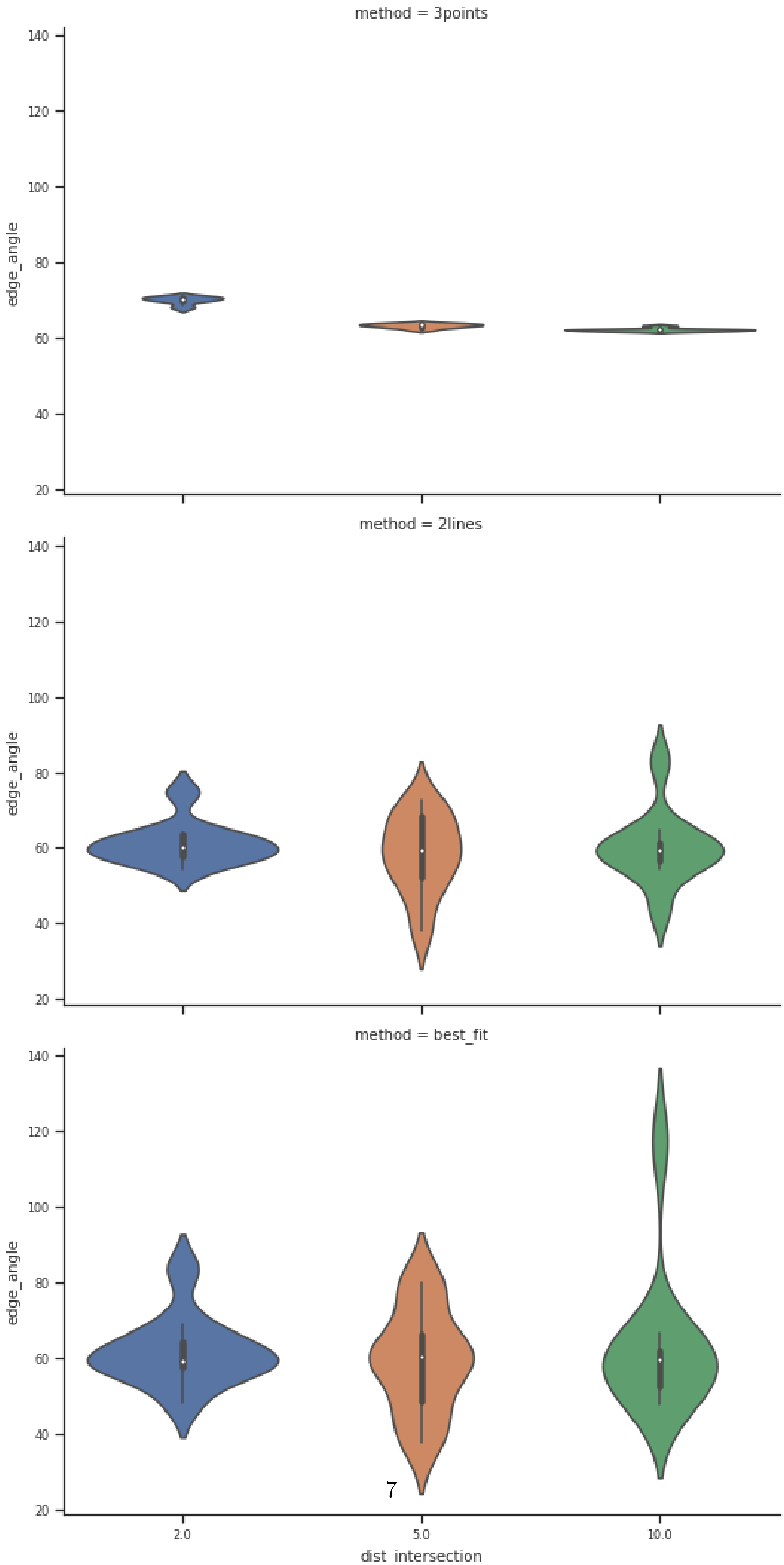
[16]:
```
df[df.duplicated(keep=False)]
```

[16]:
```
Empty DataFrame
Columns: [section, angle_number, steps, dist_intersection, segment_length,
sectionNumber, object, method, edge_angle]
Index: []
```

1.4 Check that the angle is equal to 60° for WEM

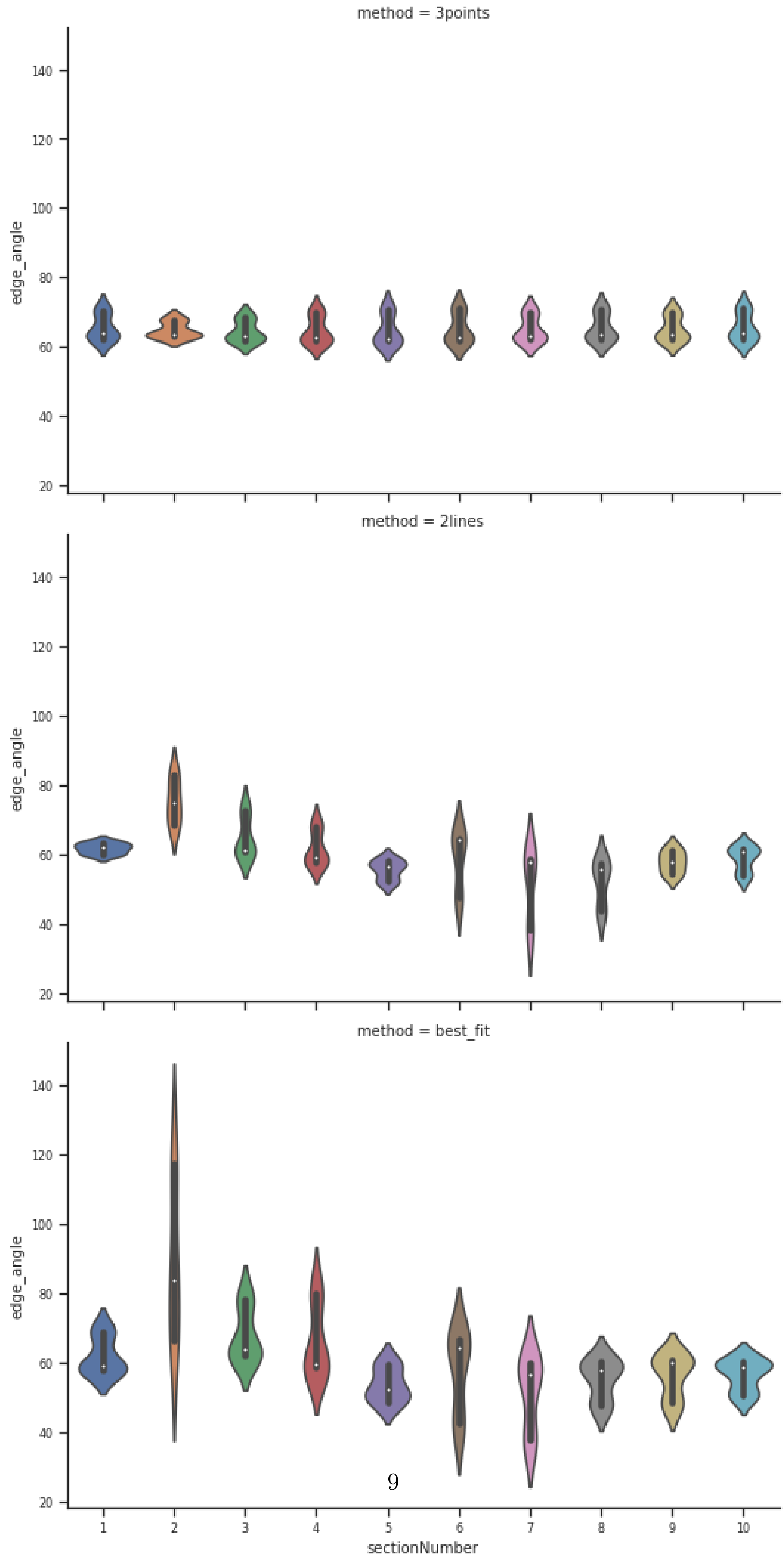
First, the section is averaged over, then the dist_intersection.

[17]:
```
sns.catplot(data=df[df.object ==␣
 ↪'WEM'],x='dist_intersection',y='edge_angle',row='method',kind='violin',height=heigthInch,as
plt.savefig(path + "Check_WEM_dist.pdf", bbox_inches='tight',dpi= 300)
```


method = 3points
method = 2lines
method = best_fit
edge_angle
20
40
60
80
100
120
140
2.0
5.0
10.0
dist_intersection

[18]:

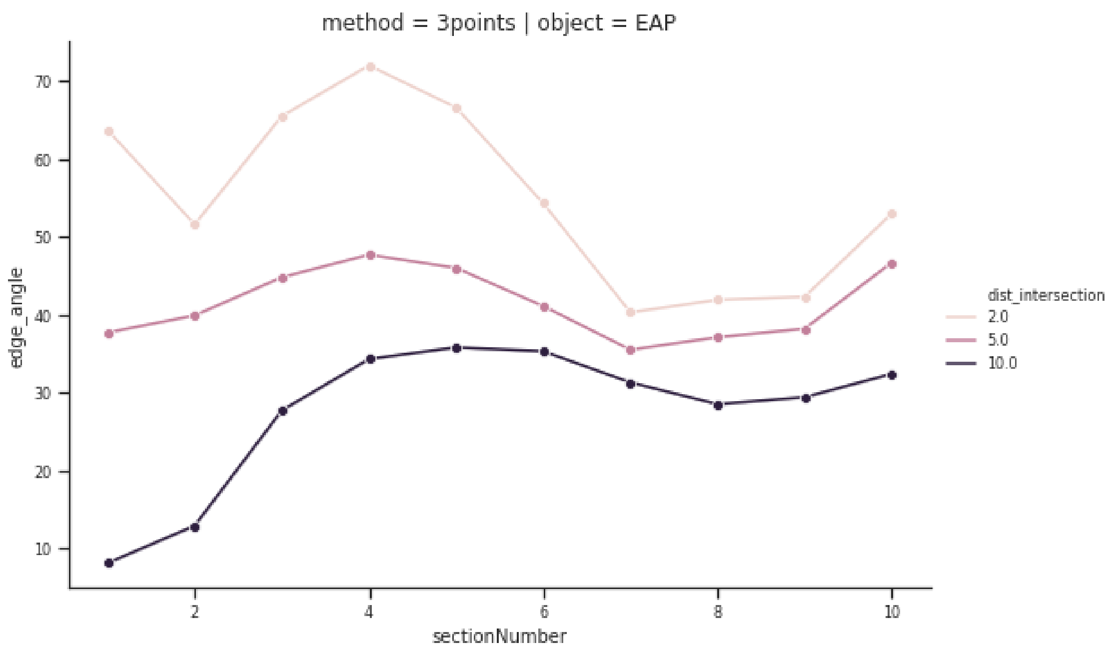
```
sns.catplot(data=df[df.object ==␣
 ↪'WEM'],x='sectionNumber',y='edge_angle',row='method',kind='violin',height=heigthInch,aspect
plt.savefig(path + "Check_WEM_sec.pdf", bbox_inches='tight',dpi= 300)
```

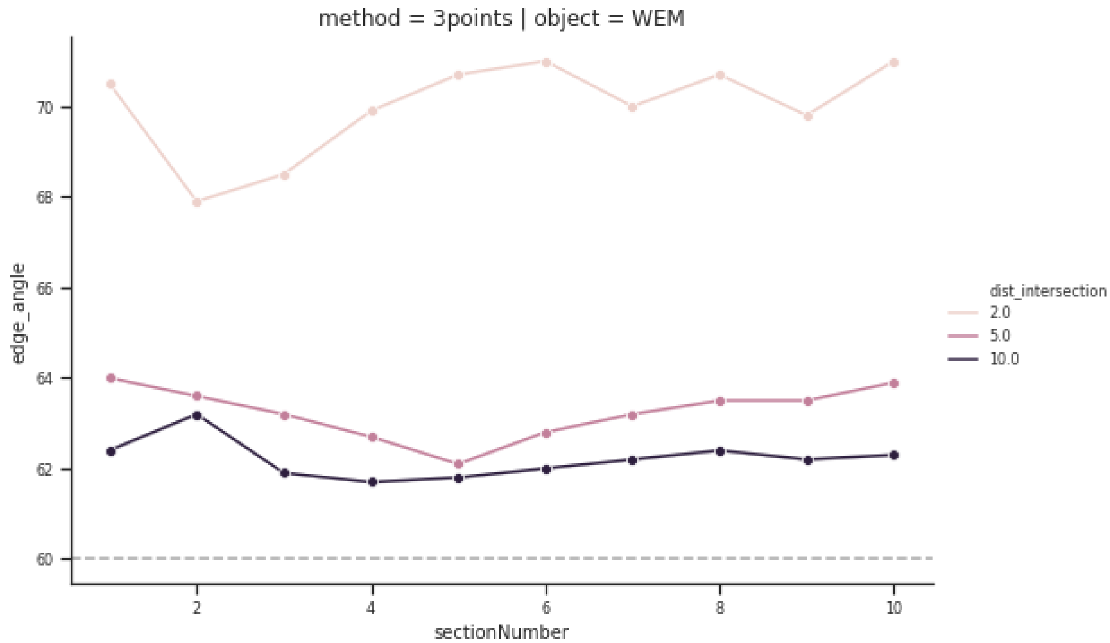

method = 3points
method = 2lines
method = best_fit
edge_angle
sectionNumber
20
40
60
80
100
120
140
1
2
3
4
5
6
7
8
9
10

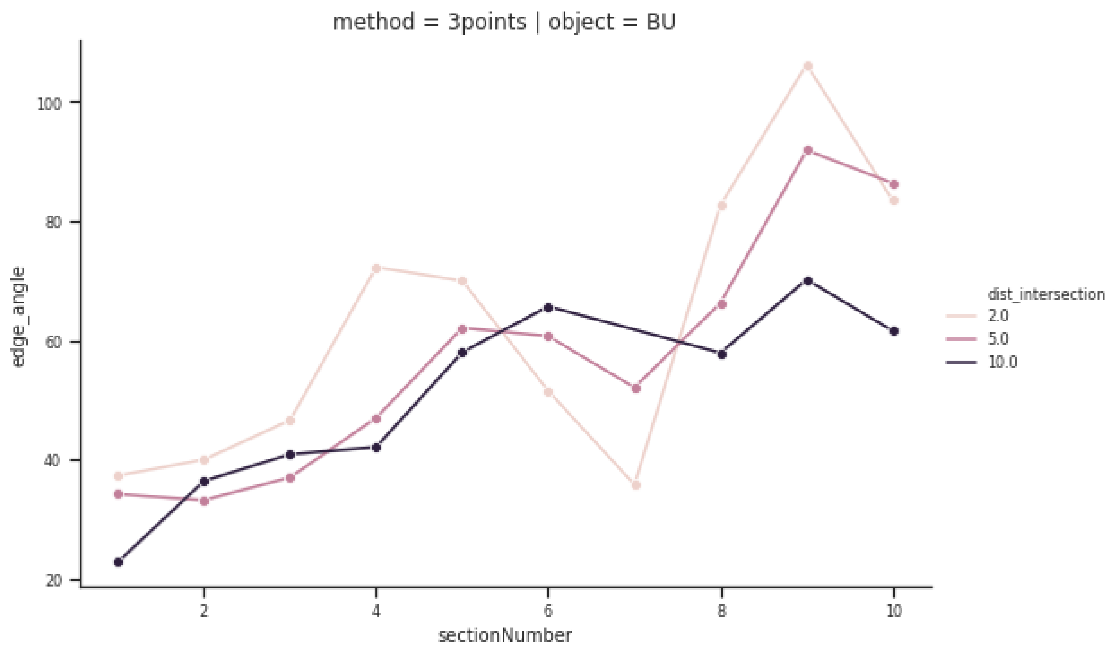
1.5 Show the variance of the angles along the edge, i. e. for the different locations

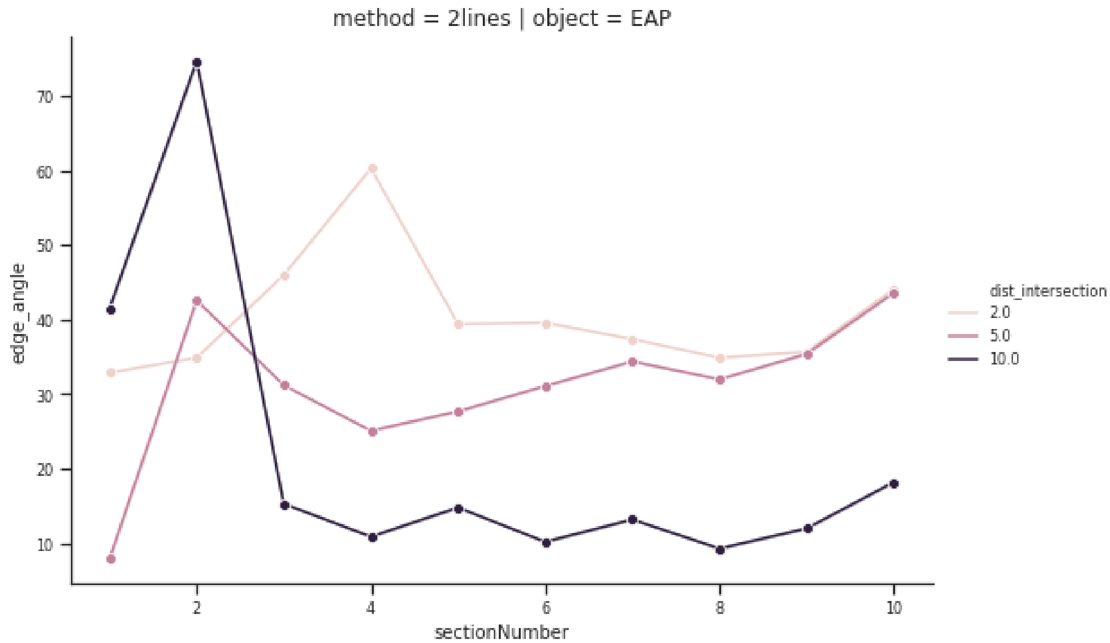
We look at the angle for all objects, all locations and all methods. The depth is averaged over.

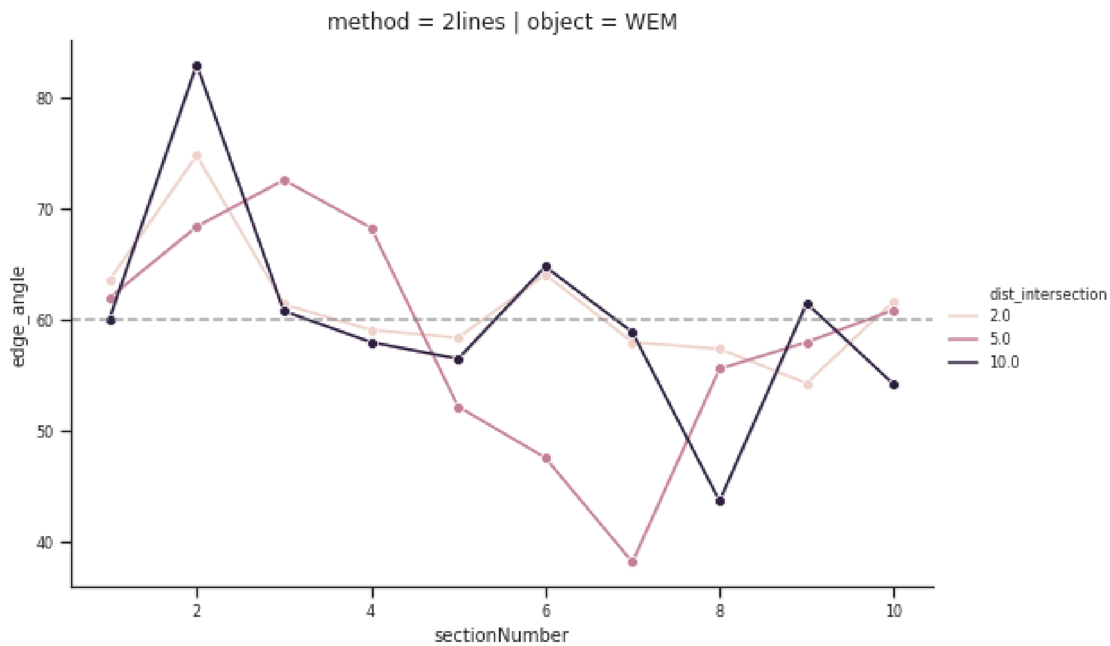
[19]:
```
for method in df.method.unique():
    for objectName in df.object.unique():
        data = df[(df.method == method) & (df.object == objectName)]
        ax = sns.
 ↪relplot(data=data,x='sectionNumber',y='edge_angle',kind='line',hue='dist_intersection',lege
 ↪marker='o')
        plt.title("method = {} | object = {}".format(method,objectName),␣
 ↪fontsize=BIGGER_SIZE)
        if objectName == "WEM":
            plt.axhline(y=60,color='gray',alpha=0.7,ls='--')
        plt.savefig(path + "Along_edge_{}_{}.pdf".format(method,objectName),␣
 ↪bbox_inches='tight',dpi= 300)
```

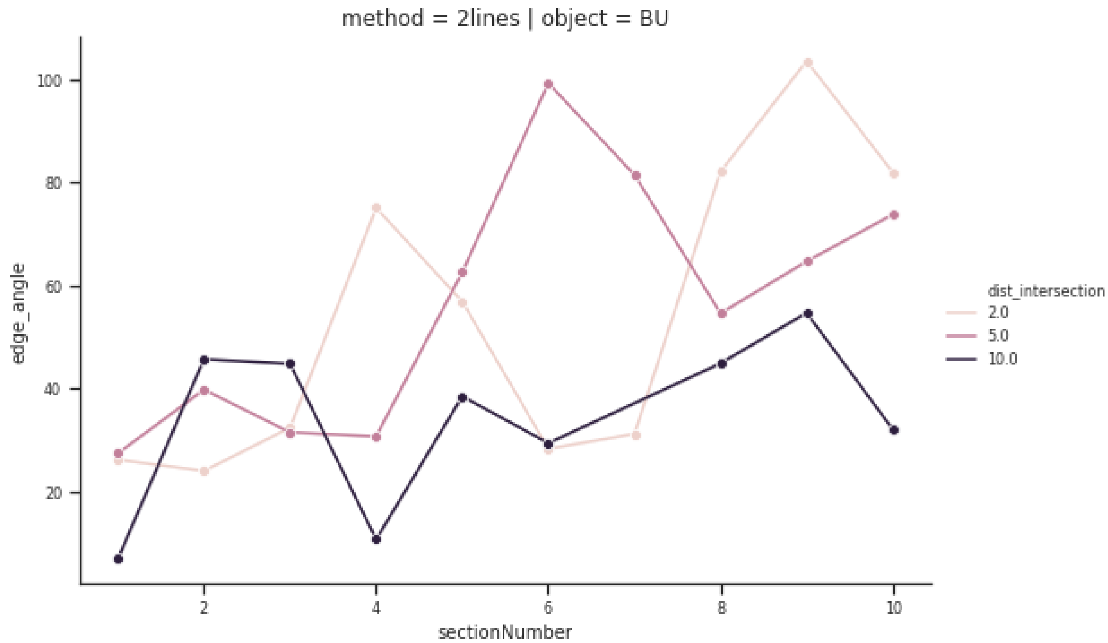


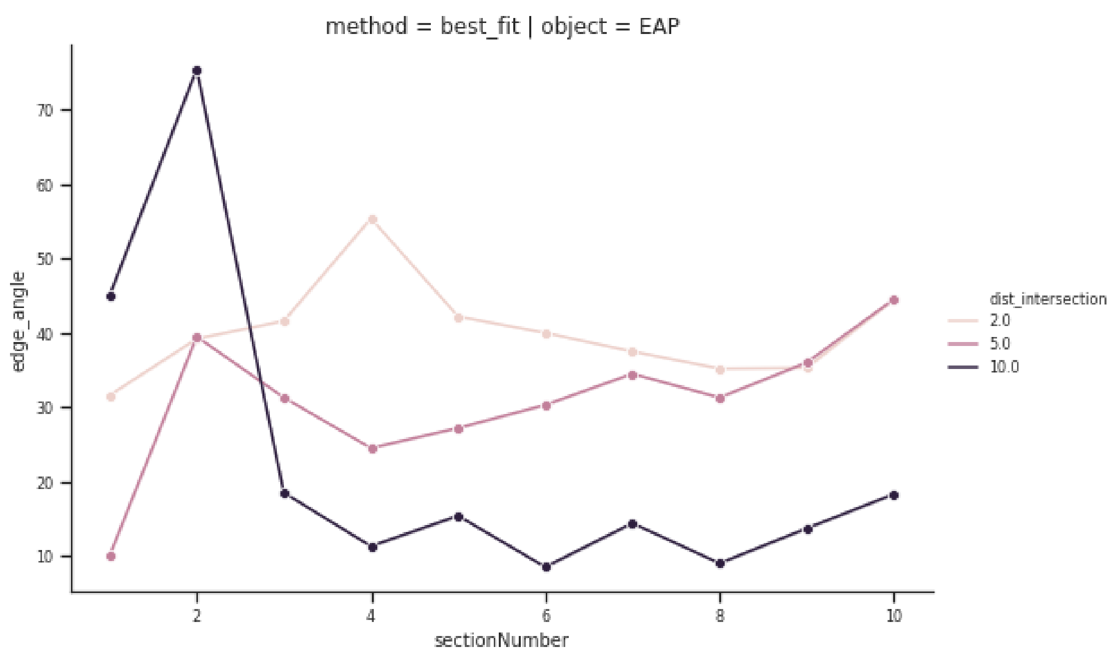

method = 3points | object = WEM
edge_angle
sectionNumber
dist_intersection
2.0
5.0
10.0

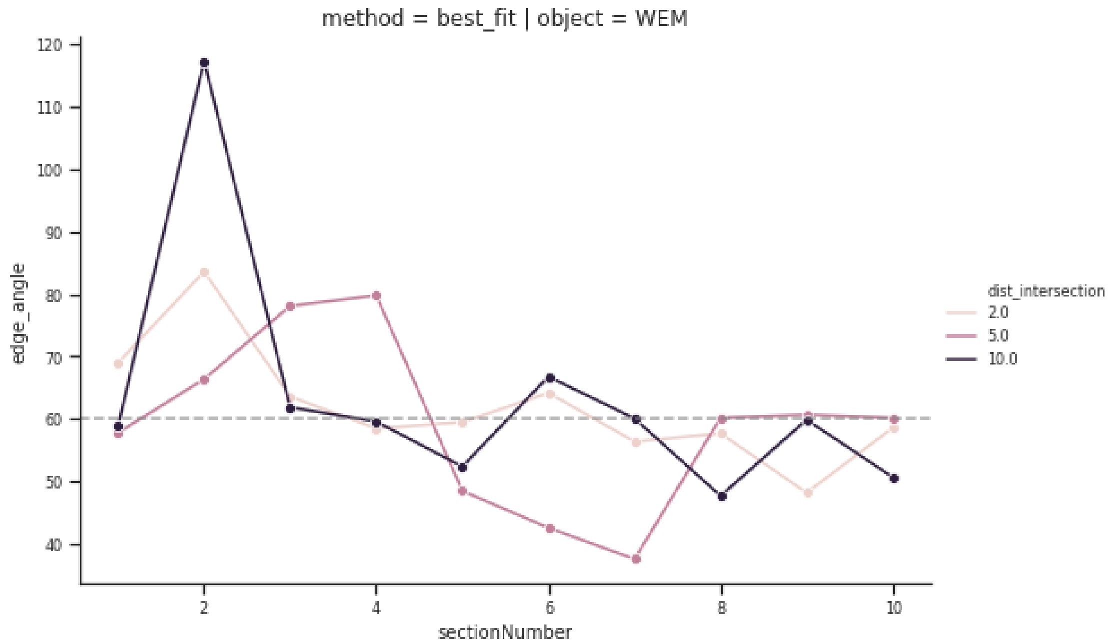

method = 3points | object = BU
edge_angle
sectionNumber
dist_intersection
2.0
5.0
10.0

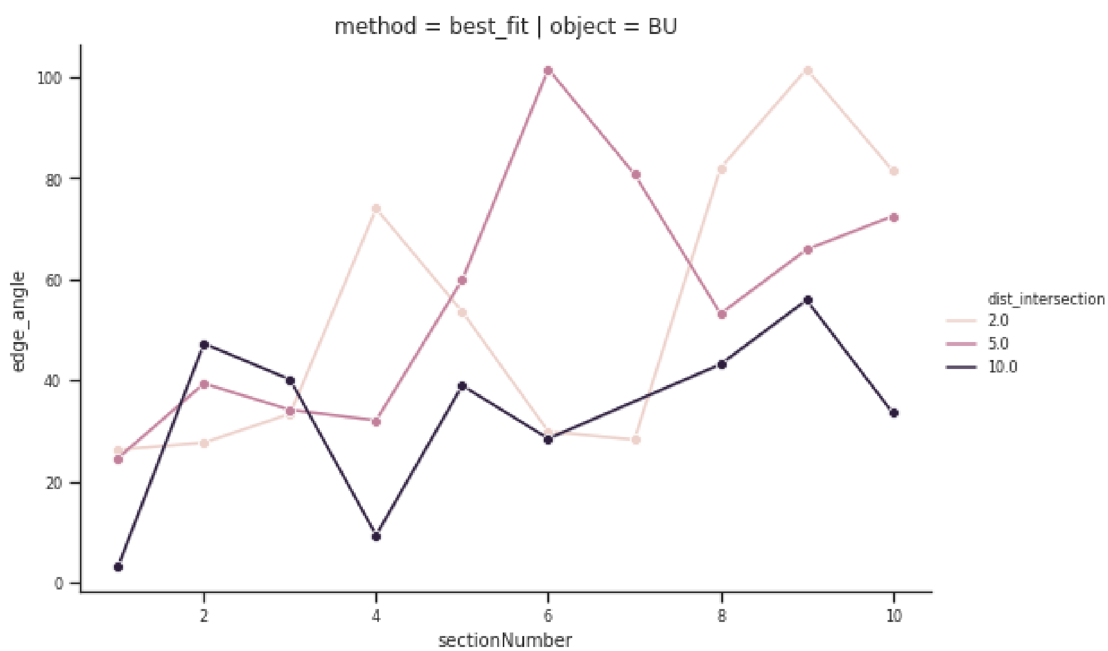

method = 2lines | object = EAP
edge_angle
sectionNumber
dist_intersection
2.0
5.0
10.0
10
20
30
40
50
60
70
2
4
6
8
10


method = 2lines | object = WEM
edge_angle
sectionNumber
dist_intersection
2.0
5.0
10.0
40
50
60
70
80
2
4
6
8
10


method = 2lines | object = BU
edge_angle
sectionNumber
dist_intersection
2.0
5.0
10.0
20
40
60
80
100
2
4
6
8
10


method = best_fit | object = EAP
edge_angle
sectionNumber
dist_intersection
2.0
5.0
10.0
10
20
30
40
50
60
70
2
4
6
8
10

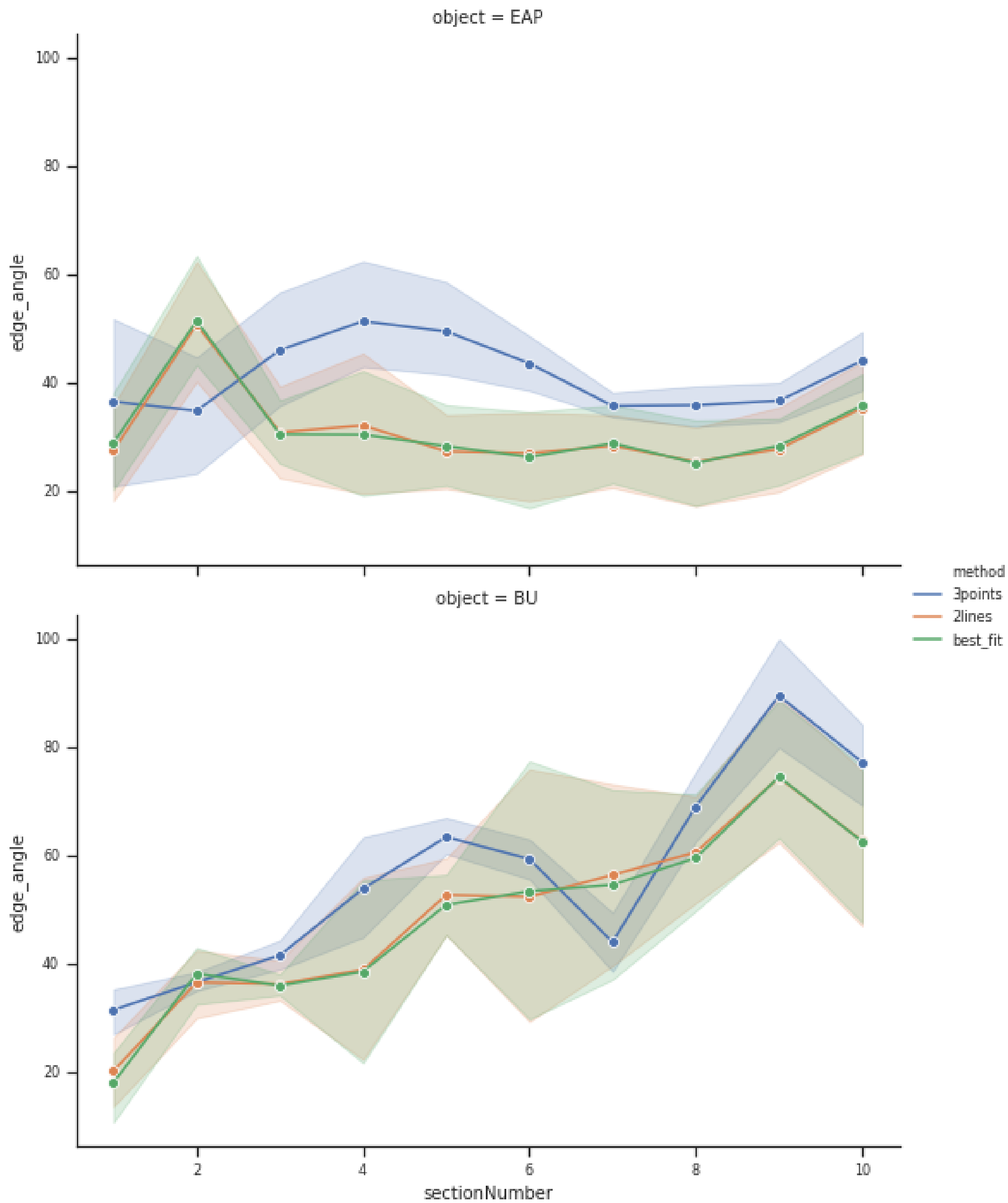




1.6 Show differences between methods on the two other samples

We look at the angle on the two other objects by all methods for all locations and depths.

[20]:
```
sns.relplot(data=df[ ~(df.object == 'WEM')␣
 ↪],x='sectionNumber',y='edge_angle',hue='method',row='object',height=heigthInch,aspect=aspect
 ↪marker='o')
plt.savefig(path + "EAP_BU.pdf", bbox_inches='tight',dpi= 300)
```



1.7 Overall goal is to access which method is most suitable

1.8 Method

In statistics, a quality measure that is often used is the mean squared error.

We can evaluate the mean squared error for the WEM object under the assumption that the true angle for all sections is 60°.

[21]:
```
dfMS = df[df.object == 'WEM']
trueAngle = 60.0
dfMS = dfMS.assign(squaredError=dfMS.edge_angle.apply( lambda x: np.power(x -␣
↪trueAngle,2)))
dfMS.groupby(['method','dist_intersection']).mean()['squaredError']
```

[21]:
```
method    dist_intersection
2lines    2.0                   30.028
          5.0                  101.622
          10.0                  87.173
3points   2.0                  100.994
          5.0                   10.869
          10.0                   5.047
best_fit  2.0                   83.051
          5.0                  170.279
          10.0                 363.007
Name: squaredError, dtype: float64
```

1.9 Result

We see that the minimum squared error is realized by the 3points method at dist_intersection = 10.0 and is thus the recommended method.

As a sanity check I look at the number of data points and manually inspect the results:

[22]:
```
dfMS.groupby(['method','dist_intersection']).count()['squaredError']
```

[22]:
```
method    dist_intersection
2lines    2.0                  30
          5.0                  30
          10.0                 30
3points   2.0                  30
          5.0                  30
          10.0                 30
best_fit  2.0                  30
          5.0                  30
          10.0                 30
Name: squaredError, dtype: int64
```

```
[23]: df[(df.object == 'WEM') & (df.method == '3points') & (df.dist_intersection ==
      10.0)]
```

```
[23]:                         section  angle_number  steps  dist_intersection  \
1447  WEM-60_1_E1_RE_SEC-01_local            50    0.2               10.0
1492  WEM-60_1_E1_RE_SEC-01_local            20    0.5               10.0
1527  WEM-60_1_E1_RE_SEC-01_local            10    1.0               10.0
1596  WEM-60_1_E1_RE_SEC-02_local            50    0.2               10.0
1641  WEM-60_1_E1_RE_SEC-02_local            20    0.5               10.0
1676  WEM-60_1_E1_RE_SEC-02_local            10    1.0               10.0
1745  WEM-60_1_E1_RE_SEC-03_local            50    0.2               10.0
1790  WEM-60_1_E1_RE_SEC-03_local            20    0.5               10.0
1825  WEM-60_1_E1_RE_SEC-03_local            10    1.0               10.0
1894  WEM-60_1_E1_RE_SEC-04_local            50    0.2               10.0
1939  WEM-60_1_E1_RE_SEC-04_local            20    0.5               10.0
1974  WEM-60_1_E1_RE_SEC-04_local            10    1.0               10.0
2043  WEM-60_1_E1_RE_SEC-05_local            50    0.2               10.0
2088  WEM-60_1_E1_RE_SEC-05_local            20    0.5               10.0
2123  WEM-60_1_E1_RE_SEC-05_local            10    1.0               10.0
2192  WEM-60_1_E1_RE_SEC-06_local            50    0.2               10.0
2237  WEM-60_1_E1_RE_SEC-06_local            20    0.5               10.0
2272  WEM-60_1_E1_RE_SEC-06_local            10    1.0               10.0
2341  WEM-60_1_E1_RE_SEC-07_local            50    0.2               10.0
2386  WEM-60_1_E1_RE_SEC-07_local            20    0.5               10.0
2421  WEM-60_1_E1_RE_SEC-07_local            10    1.0               10.0
2490  WEM-60_1_E1_RE_SEC-08_local            50    0.2               10.0
2535  WEM-60_1_E1_RE_SEC-08_local            20    0.5               10.0
2570  WEM-60_1_E1_RE_SEC-08_local            10    1.0               10.0
2639  WEM-60_1_E1_RE_SEC-09_local            50    0.2               10.0
2685  WEM-60_1_E1_RE_SEC-09_local            20    0.5               10.0
2720  WEM-60_1_E1_RE_SEC-09_local            10    1.0               10.0
2789  WEM-60_1_E1_RE_SEC-10_local            50    0.2               10.0
2835  WEM-60_1_E1_RE_SEC-10_local            20    0.5               10.0
2870  WEM-60_1_E1_RE_SEC-10_local            10    1.0               10.0

      segment_length  sectionNumber object   method  edge_angle
1447             0.5              1    WEM  3points        62.4
1492             0.5              1    WEM  3points        62.4
1527             0.5              1    WEM  3points        62.4
1596             0.5              2    WEM  3points        63.2
1641             0.5              2    WEM  3points        63.2
1676             0.5              2    WEM  3points        63.2
1745             0.5              3    WEM  3points        61.9
1790             0.5              3    WEM  3points        61.9
1825             0.5              3    WEM  3points        61.9
1894             0.5              4    WEM  3points        61.7
1939             0.5              4    WEM  3points        61.7
```

1974	0.5	4	WEM	3points	61.7
2043	0.5	5	WEM	3points	61.8
2088	0.5	5	WEM	3points	61.8
2123	0.5	5	WEM	3points	61.8
2192	0.5	6	WEM	3points	62.0
2237	0.5	6	WEM	3points	62.0
2272	0.5	6	WEM	3points	62.0
2341	0.5	7	WEM	3points	62.2
2386	0.5	7	WEM	3points	62.2
2421	0.5	7	WEM	3points	62.2
2490	0.5	8	WEM	3points	62.4
2535	0.5	8	WEM	3points	62.4
2570	0.5	8	WEM	3points	62.4
2639	0.5	9	WEM	3points	62.2
2685	0.5	9	WEM	3points	62.2
2720	0.5	9	WEM	3points	62.2
2789	0.5	10	WEM	3points	62.3
2835	0.5	10	WEM	3points	62.3
2870	0.5	10	WEM	3points	62.3

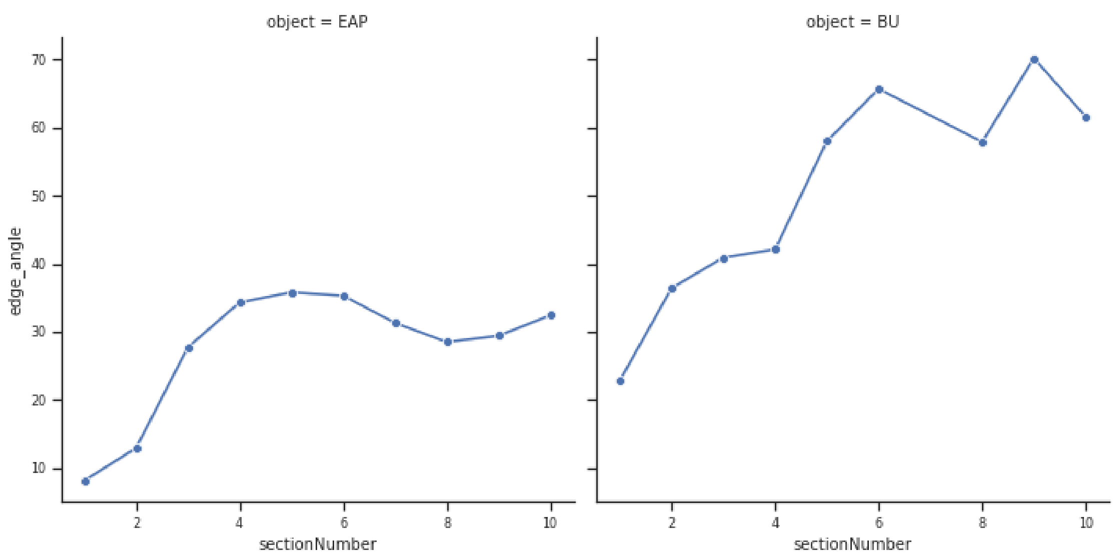
We see that the values occur repeatedly, but there still seven distinct values.

1.10 Prediction

Having choosen the method, the results on the other two objects now look as follows:

[24]:
```
sns.relplot(data=df[ ~(df.object == 'WEM') & (df.method == '3points') & (df.
↪dist_intersection == 10.
↪0)],x='sectionNumber',y='edge_angle',col='object',kind='line', marker='o')
```

[24]:
```
<seaborn.axisgrid.FacetGrid at 0x7f536c773da0>
```



1.11 Write out

[25]:
```
!jupyter nbconvert --to html EdgeAnglesV4.ipynb
```

```
[NbConvertApp] Converting notebook EdgeAnglesV4.ipynb to html
[NbConvertApp] Writing 1067356 bytes to EdgeAnglesV4.html
```

[26]:
```
!jupyter nbconvert --to markdown EdgeAnglesV4.ipynb
```

```
[NbConvertApp] Converting notebook EdgeAnglesV4.ipynb to markdown
[NbConvertApp] Support files will be in EdgeAnglesV4_files/
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Making directory EdgeAnglesV4_files
[NbConvertApp] Writing 33407 bytes to EdgeAnglesV4.md
```

[ ]:
```

```