City Council Voter Turnout Analysis

In [13]:

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

#loading datasets into dataframes
cc2011 = pd.read_csv("2011_CityCouncil_Results_Race_Turnout.csv")
cc2013 = pd.read_csv("2013_CityCouncil_Race_Turnout_Results.csv")
cc2015 = pd.read_csv("2015_city_council.csv")
cc2017 = pd.read_csv("2017_CityCouncil_AtLarge_Turnout_Race.csv")
cc2019 = pd.read_csv("2019_CityCouncil_Race Turnout.csv")
```

In [2]:

```python
#checking that these are all the same length
print("Shape of cc2011:", cc2011.shape)
print("Shape of cc2013:", cc2013.shape)
print("Shape of cc2015:", cc2015.shape)
print("Shape of cc2017:", cc2017.shape)
print("Shape of cc2019:", cc2019.shape)
```

```
Shape of cc2011: (254, 48)
Shape of cc2013: (254, 46)
Shape of cc2015: (254, 47)
Shape of cc2017: (254, 54)
Shape of cc2019: (257, 51)
```

In [3]:

```python
cc2011 = cc2011.drop([253])
cc2011
```

Out[3]:

	Unnamed: 0	WILL DORCENA	AYANNA PRESSLEY	FELIX G ARROYO	JOHN R CONNOLLY	MICHAEL F FLAHERTY	STEPHEN J MURPHY	SEAN H RYAN	WILL FEEG Writ
0	0	16	113	111	117	100	99	24	
1	1	9	44	55	42	70	52	10	
2	2	40	133	155	123	101	109	34	
3	3	5	29	47	40	45	43	8	
4	4	12	54	67	63	69	63	11	
...	...	...	...	...	...	...	...	...	
248	248	16	71	88	90	75	72	18	
249	249	18	75	86	75	60	55	17	
250	250	32	123	126	112	99	119	30	
251	251	11	66	74	84	52	77	12	
252	252	22	60	79	79	86	75	23	

253 rows × 48 columns

In [4]:

```
cc2013 = cc2013.drop([253])
cc2015 = cc2015.drop([253])
cc2017 = cc2017.drop([253])
```

2019 is slightly longer than the others for some reason. We will discard 2019 for now because the data needs to be manually standarized to match the other datasets. We will also disregard WP 2213 from each of the datasets since this data is incomplete.

In [5]:

```
#beginning with 2011, add turnout column
cc2011["Turnout2011"] = cc2011["BALLOTS CAST"]/cc2011["Registered Voters"]
cc2011["Turnout2011"]
```

Out[5]:

```
0      0.186715
1      0.145342
2      0.156431
3      0.175182
4      0.140314
         ...
248    0.116996
249    0.096774
250    0.136968
251    0.127363
252    0.155075
Name: Turnout2011, Length: 253, dtype: float64
```

In [6]:

```
# adding turnout column to 2013
cc2013["Turnout2013"] = cc2013["BALLOTS CAST"]/cc2013["Registered Voters"]
cc2013["Turnout2013"]
```

Out[6]:

```
0      0.520227
1      0.453782
2      0.439695
3      0.441221
4      0.425512
         ...
248    0.318999
249    0.265783
250    0.355091
251    0.338967
252    0.323427
Name: Turnout2013, Length: 253, dtype: float64
```

In [7]:

```
# adding turnout column to 2015
cc2015["Turnout2015"] = cc2015["BALLOTS CAST"]/cc2015["Registered Voters"]
cc2015["Turnout2015"]
```

Out[7]:

```
0      0.135303
1      0.112750
2      0.134523
3      0.089686
4      0.077748
         ...
248    0.091053
249    0.075243
250    0.113333
251    0.108798
```

```
252    0.118353
Name: Turnout2015, Length: 253, dtype: float64
```

In [8]:

```python
# 2017 already has a turnout column for some reason
cc2017 = cc2017.rename(columns= {"Turnout":"Turnout2017"})
cc2017["Turnout2017"]
```

Out[8]:

```
0      0.354331
1      0.287923
2      0.298865
3      0.288221
4      0.282869
         ...
248    0.230109
249    0.213855
250    0.268729
251    0.254563
252    0.236667
Name: Turnout2017, Length: 253, dtype: float64
```

In [10]:

```python
# creating a new dataframe with turnout data
temp1 = cc2011[["WP", "Turnout2011"]]
turnouts = temp1.join(cc2013[["Turnout2013"]]).join(cc2015[["Turnout2015"]]).joi
turnouts
```

Out[10]:

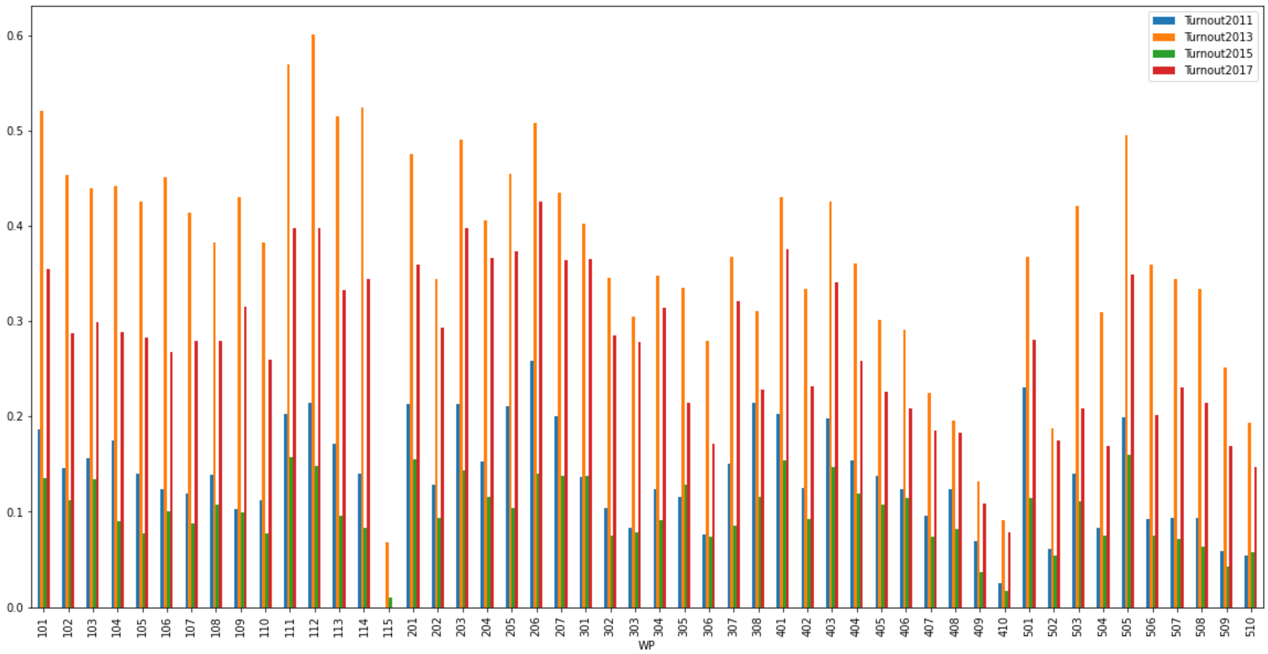
	WP	Turnout2011	Turnout2013	Turnout2015	Turnout2017
0	101	0.186715	0.520227	0.135303	0.354331
1	102	0.145342	0.453782	0.112750	0.287923
2	103	0.156431	0.439695	0.134523	0.298865
3	104	0.175182	0.441221	0.089686	0.288221
4	105	0.140314	0.425512	0.077748	0.282869
...	...	...	...	...	...
248	2208	0.116996	0.318999	0.091053	0.230109
249	2209	0.096774	0.265783	0.075243	0.213855
250	2210	0.136968	0.355091	0.113333	0.268729
251	2211	0.127363	0.338967	0.108798	0.254563
252	2212	0.155075	0.323427	0.118353	0.236667

253 rows × 5 columns

In [27]:

```python
# visualizing our datasets 50 WPs at a time
turnouts[:50].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015", "Turn
```

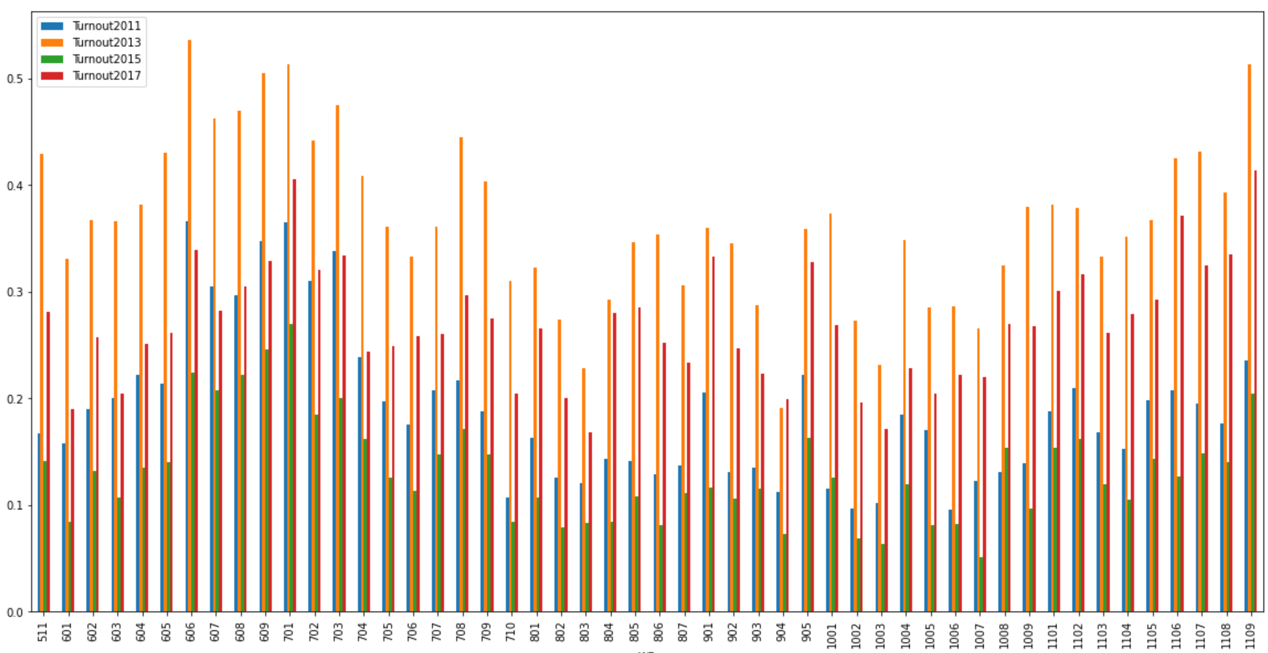
Out[27]: <AxesSubplot:xlabel='WP'>



In [28]:

```
turnouts[50:100].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015", "T
```

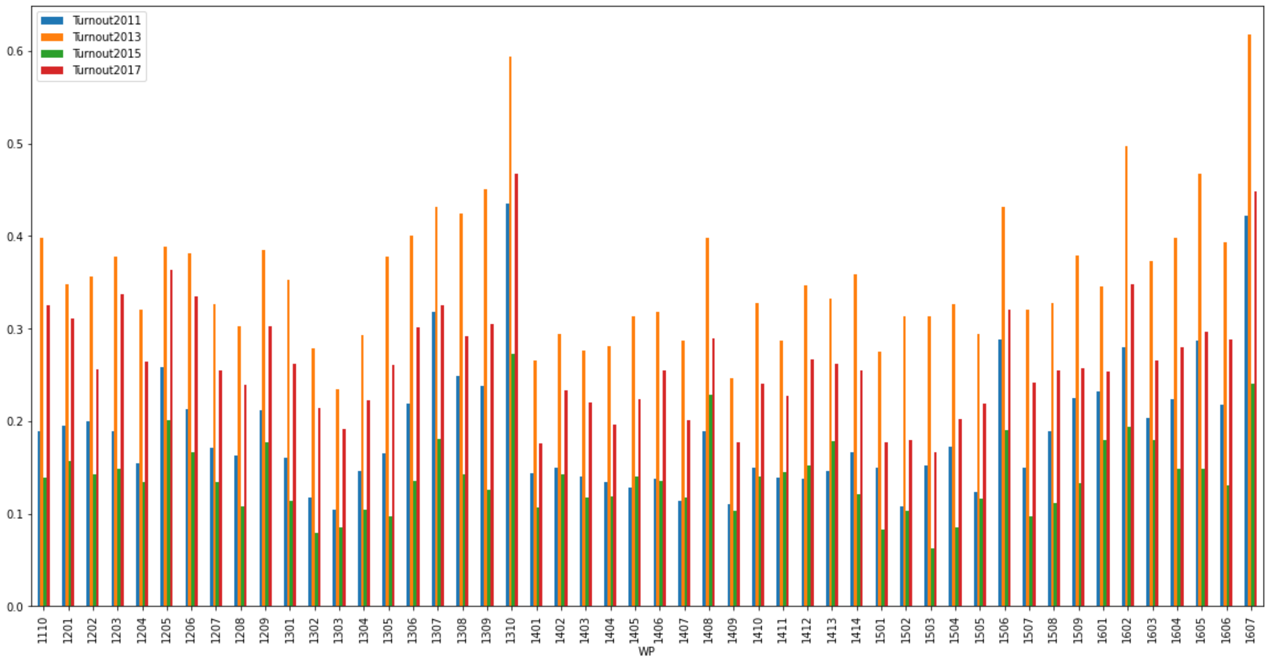
Out[28]: <AxesSubplot:xlabel='WP'>



In [29]:

```
turnouts[100:150].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015", "
```

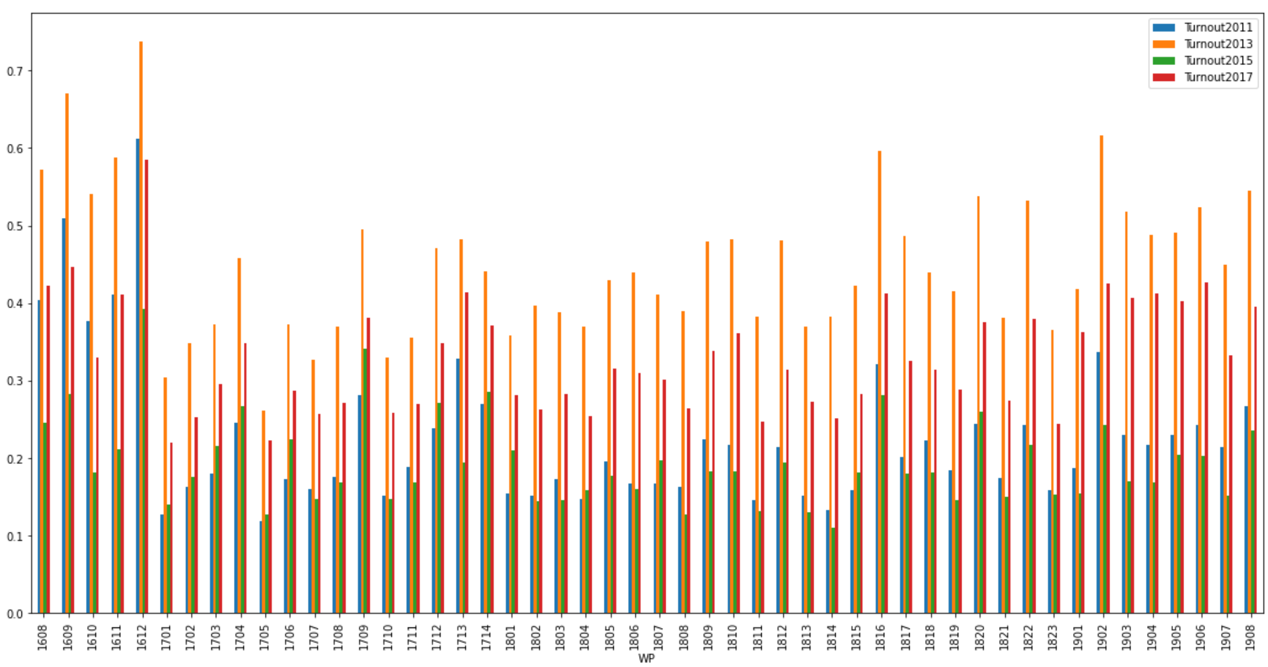
Out[29]: <AxesSubplot:xlabel='WP'>



In [30]:

```
turnouts[150:200].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015", "
```

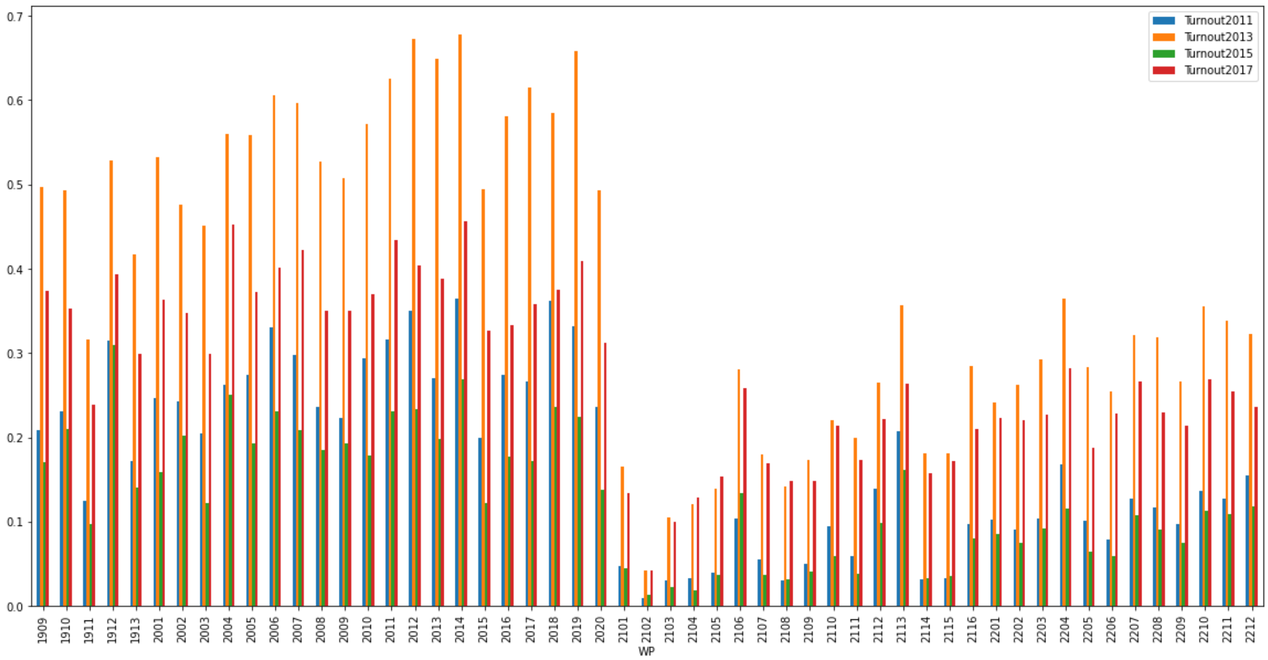
Out[30]: <AxesSubplot:xlabel='WP'>



In [31]:

```
turnouts[200:].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015", "Tur
```

Out[31]: <AxesSubplot:xlabel='WP'>



The 2013 race seems to have the highest voter turnout. Why is this?

Calculating average change over time

In [37]:

```python
import math
turnouts["Diff11_13"] = turnouts["Turnout2011"] - turnouts["Turnout2013"]
turnouts["Diff11_13"] = turnouts["Diff11_13"].abs()
turnouts["Diff13_15"] = turnouts["Turnout2013"] - turnouts["Turnout2015"]
turnouts["Diff13_15"] = turnouts["Diff13_15"].abs()
turnouts["Diff15_17"] = turnouts["Turnout2015"] - turnouts["Turnout2017"]
turnouts["Diff15_17"] = turnouts["Diff15_17"].abs()
turnouts["SumChange"] = turnouts["Diff11_13"] + turnouts["Diff13_15"] + turnouts
turnouts["AvgChange"] = turnouts["SumChange"] / 3.0
turnouts
```

Out[37]:

	WP	Turnout2011	Turnout2013	Turnout2015	Turnout2017	Diff11_13	Diff13_15	Diff15_17
0	101	0.186715	0.520227	0.135303	0.354331	0.333512	0.384923	0.219027
1	102	0.145342	0.453782	0.112750	0.287923	0.308440	0.341031	0.175172
2	103	0.156431	0.439695	0.134523	0.298865	0.283264	0.305172	0.164342
3	104	0.175182	0.441221	0.089686	0.288221	0.266039	0.351535	0.198534
4	105	0.140314	0.425512	0.077748	0.282869	0.285198	0.347764	0.205121
...	...	...	...	...	...	...	...	...
248	2208	0.116996	0.318999	0.091053	0.230109	0.202003	0.227946	0.139056
249	2209	0.096774	0.265783	0.075243	0.213855	0.169009	0.190541	0.138613
250	2210	0.136968	0.355091	0.113333	0.268729	0.218123	0.241758	0.155395
251	2211	0.127363	0.338967	0.108798	0.254563	0.211604	0.230169	0.145764
252	2212	0.155075	0.323427	0.118353	0.236667	0.168351	0.205073	0.118313

253 rows × 10 columns

In [39]:

```
#finding the top 20 precincts with the greatest average change
top_change = turnouts.sort_values(by=['AvgChange'])
top_change
```

Out[39]:

	WP	Turnout2011	Turnout2013	Turnout2015	Turnout2017	Diff11_13	Diff13_15	Diff15_17
226	2102	0.009475	0.042741	0.013972	0.042870	0.033267	0.028769	0.028898
14	115	0.000000	0.067568	0.010526	0.000000	0.067568	0.057041	0.010526
39	410	0.025397	0.091667	0.017196	0.078105	0.066270	0.074471	0.060909
38	409	0.069479	0.131491	0.037133	0.108520	0.062012	0.094358	0.071387
227	2103	0.030808	0.105505	0.022889	0.099558	0.074697	0.082616	0.076669
...	...	...	...	...	...	...	...	...
12	113	0.170962	0.514525	0.095393	0.332463	0.343563	0.419132	0.237071
10	111	0.202640	0.569517	0.157428	0.397193	0.366877	0.412089	0.239765
217	2013	0.269737	0.648734	0.197959	0.388199	0.378997	0.450775	0.190240
13	114	0.139509	0.523711	0.083164	0.343750	0.384202	0.440547	0.260586
11	112	0.214850	0.600766	0.147806	0.397070	0.385916	0.452960	0.249264

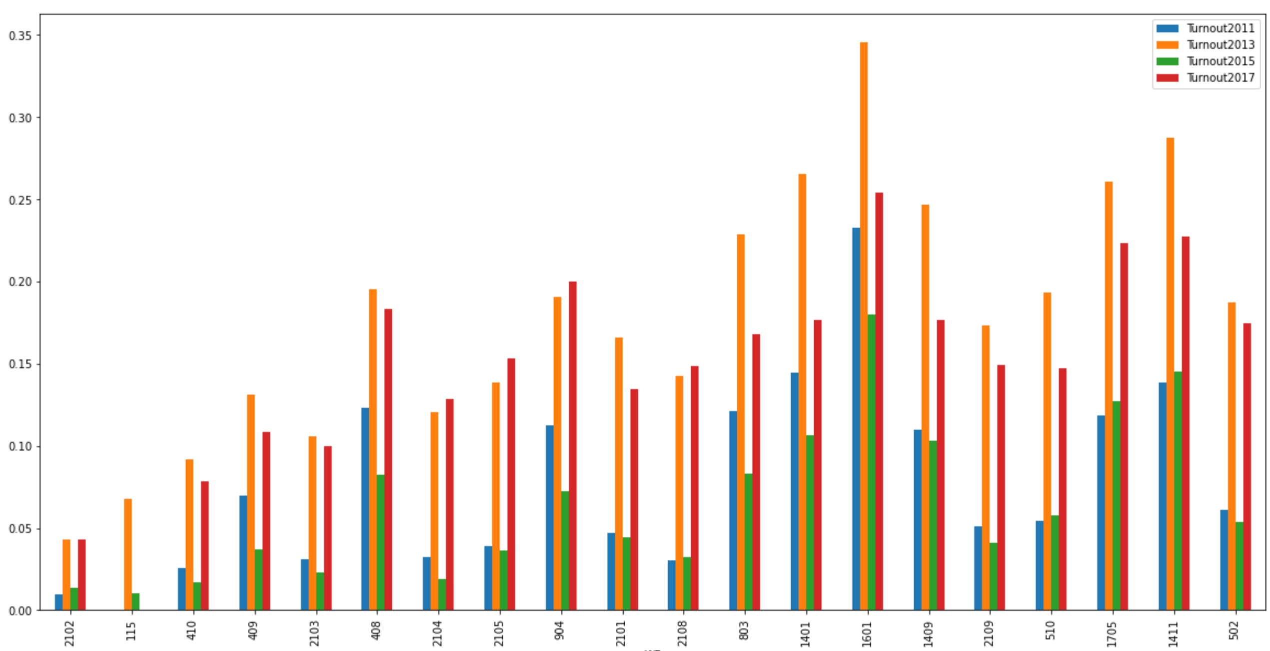
253 rows × 10 columns

In [40]:

```
#visualizing these top 20
top_change[:20].plot(x="WP", y=["Turnout2011", "Turnout2013", "Turnout2015", "Tu
```

Out[40]:

```
<AxesSubplot:xlabel='WP'>
```



The next steps would include demographic analysis of each of these precincts.