Problem Set 2: Analysis of racial disparities in felony sentencing, Part 2

Part 2: Investigating Black vs. White sentencing disparities

We're going to investigate different types of disparities in sentencing between Black defendants and White defendants. We're focusing on these groups for the purpose of the problem set, but the analysis could be extended to study Hispanic defendants or, in a different jurisdiction, Asian and other minoritized groups.

Details if interested in digging deeper: If interested (optional), you can read more technical coverage of how we might (1) measure disparities, and (2) what factors you want to adjust for when deciding whether two defendants are 'similarly situated' but for their race in the following sources:

- Review of sentencing disparities research
- Discussion of causal model/blinding race at charging stage of the prosecutorial process
- Discussion of measuring discrimination in policing that can generalize to the sentencing case
- General discussion of causal challenges in measuring between-group disparities

One major caveat: when investigating whether two similar defendants received different sentences, we're missing one important attribute that influences sentencing: the defendant's criminal history. This influences sentencing both through sentencing guidelines, which can prescribe longer sentences for those who have certain types of prior convictions, and through judicial discretion if judges are more lenient with first-time defendants. The above sources discuss how much we want to "control away" for this prior history, since if we think there are racial biases in which defendants, conditional on *committing* a crime, are arrested and charged, we may not want to adjust for that factor. More discussion in this article

1.0: (0 points)

First, read in the following dataset (regardless of progress on part one): `sentencing_cleaned.pkl` (if you can't read in the pkl you can read in the .csv format but may need to recast some of the datetime columns)

Note: don't worry if there are slight differences in your output from Part One and this dataset/it's not a good use of time to try to reverse engineer Part One answers from this cleaned data.

In [38]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re


df = pd.read_pickle('pset2_inputdata/sentencing_cleaned.pkl')
```

```
c:\Users\sharp\miniforge3\envs\qss20\Lib\site-packages\pandas\compat\pickle_compat.py:35: VisibleDeprecationWarning: dtype(): align should be passed as Python or NumPy boolean but got `align=0`. Did you mean to pass a tuple to create a subarray type? (Deprecated NumPy 2.4)
  stack[-1] = func(*args)
c:\Users\sharp\miniforge3\envs\qss20\Lib\site-packages\pandas\compat\pickle_compat.py:35: VisibleDeprecationWarning: dtype(): align should be passed as Python or NumPy boolean but got `align=0`. Did you mean to pass a tuple to create a subarray type? (Deprecated NumPy 2.4)
  stack[-1] = func(*args)
c:\Users\sharp\miniforge3\envs\qss20\Lib\site-packages\pandas\compat\pickle_compat.py:35: VisibleDeprecationWarning: dtype(): align should be passed as Python or NumPy boolean but got `align=0`. Did you mean to pass a tuple to create a subarray type? (Deprecated NumPy 2.4)
  stack[-1] = func(*args)
```

In [39]:

```
df.columns

print(df.shape)
print(df["CASE_PARTICIPANT_ID"].nunique())
```

```
(135165, 52)
135165
```

1.1: Investigating one type of between-group difference: who reaches the sentencing stage? (5 points)

Tabulate and visualize the proportion of defendants, out of all defendants sentenced in a given month/year, who are Black and who are White (separate proportions)

- Denominator is number of unique cases that month
- Numerator for black defendants is count of is_black_derived
- Numerator for white defendants is count of is_white_derived
- Fraction of each is numerator/denominator
- Print the table
- Create a graph with two lines--- one for Black defendants as fraction of total; another for White defendants. Make sure it includes a legend summarizing which color is for which group, and clean the legend so that it has informative names (e.g., Black or White rather than prop_black or prop_white)
- Use mathematical notation to write out each of the proportions using summation notation in a 1-2 sentence writeup describing trends. What seems to be going on in April and May 2020?

Optional challenge: improve the viz by shading the background of the visualization for months with fewer than 100 cases

Optional challenge: improve the viz by adding a vertical line for 12-01-2016, the month that new State's Attorney Foxx took office

In [40]:

```
# Group by month/year and calculate proportions
monthly_stats = df.groupby("sentenceym_derived").agg(
    total_cases=("CASE_ID", "nunique"),
    black_count=("is_black_derived", "sum"),
    white_count=("is_white_derived", "sum")
).reset_index()
```

```
# Calculate proportions
monthly_stats["prop_black"] = monthly_stats["black_count"] / monthly_stats["total_cases"]
monthly_stats["prop_white"] = monthly_stats["white_count"] / monthly_stats["total_cases"]

# Print the table
print("Monthly Proportions of Black and White Defendants:")
print(monthly_stats.to_string())
```

Monthly Proportions of Black and White Defendants:

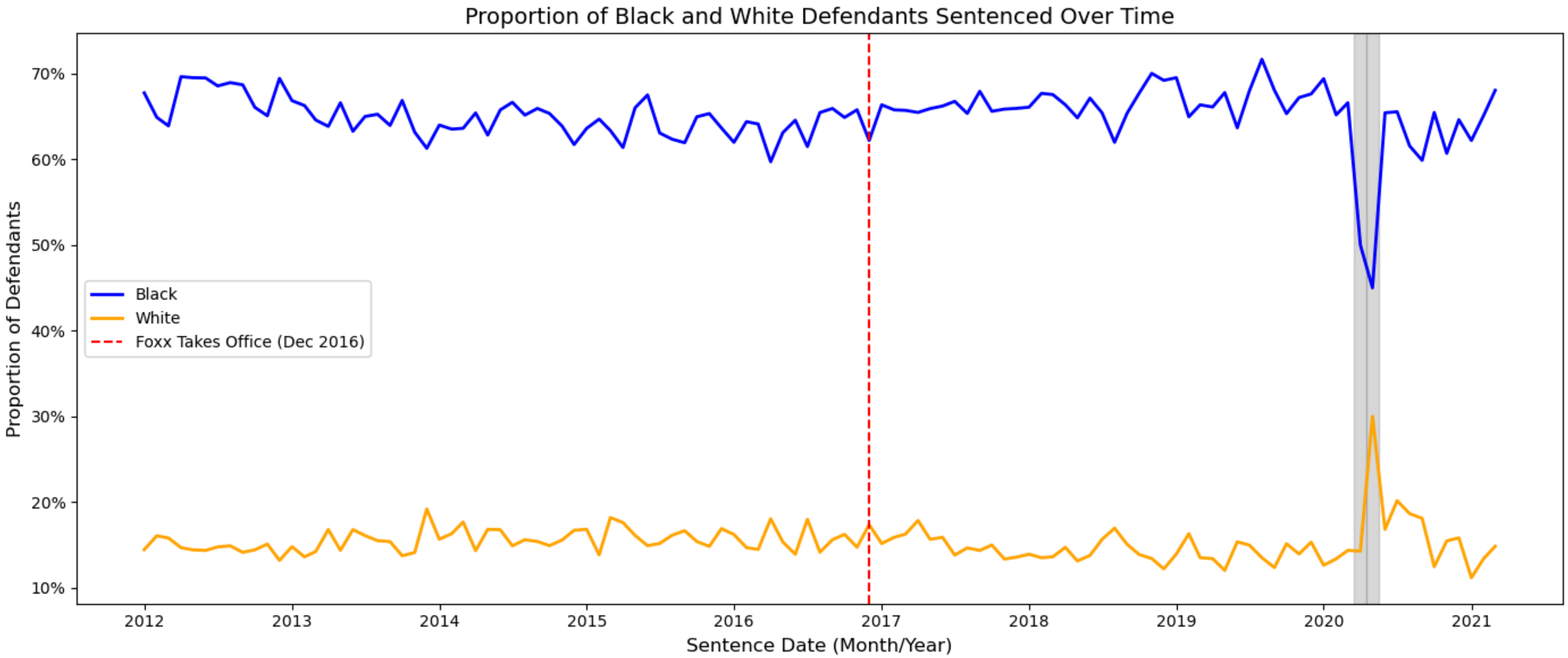
	sentenceym_derived	total_cases	black_count	white_count	prop_black	prop_white
0	2012-01-01	1674	1134	242	0.677419	0.144564
1	2012-02-01	1450	941	233	0.648966	0.160690
2	2012-03-01	1612	1030	255	0.638958	0.158189
3	2012-04-01	1436	1000	211	0.696379	0.146936
4	2012-05-01	1545	1074	223	0.695146	0.144337
5	2012-06-01	1747	1214	251	0.694906	0.143675
6	2012-07-01	1746	1197	258	0.685567	0.147766
7	2012-08-01	1684	1161	251	0.689430	0.149050
8	2012-09-01	1485	1020	210	0.686869	0.141414
9	2012-10-01	1815	1199	262	0.660606	0.144353
10	2012-11-01	1529	995	231	0.650752	0.151079
11	2012-12-01	1348	936	178	0.694362	0.132047
12	2013-01-01	1743	1165	258	0.668388	0.148021
13	2013-02-01	1263	837	172	0.662708	0.136184
14	2013-03-01	1530	988	218	0.645752	0.142484
15	2013-04-01	1684	1075	283	0.638361	0.168052
16	2013-05-01	1739	1158	250	0.665900	0.143761
17	2013-06-01	1500	949	252	0.632667	0.168000
18	2013-07-01	1634	1062	263	0.649939	0.160955
19	2013-08-01	1554	1014	241	0.652510	0.155084
20	2013-09-01	1520	972	234	0.639474	0.153947
21	2013-10-01	1862	1245	256	0.668636	0.137487
22	2013-11-01	1466	926	207	0.631651	0.141201
23	2013-12-01	1547	948	297	0.612799	0.191984
24	2014-01-01	1519	972	238	0.639895	0.156682
25	2014-02-01	1598	1015	261	0.635169	0.163329
26	2014-03-01	1561	993	276	0.636131	0.176810
27	2014-04-01	1639	1072	235	0.654057	0.143380
28	2014-05-01	1563	982	263	0.628279	0.168266
29	2014-06-01	1609	1058	270	0.657551	0.167806
30	2014-07-01	1664	1109	248	0.666466	0.149038
31	2014-08-01	1409	918	220	0.651526	0.156139
32	2014-09-01	1629	1074	251	0.659300	0.154082
33	2014-10-01	1662	1086	248	0.653430	0.149218
34	2014-11-01	1329	849	207	0.638826	0.155756
35	2014-12-01	1442	890	241	0.617198	0.167129
36	2015-01-01	1610	1024	271	0.636025	0.168323
37	2015-02-01	1314	850	182	0.646880	0.138508
38	2015-03-01	1544	978	281	0.633420	0.181995
39	2015-04-01	1561	958	275	0.613709	0.176169
40	2015-05-01	1303	860	210	0.660015	0.161167
41	2015-06-01	1508	1018	225	0.675066	0.149204
42	2015-07-01	1489	939	226	0.630625	0.151780
43	2015-08-01	1301	811	210	0.623367	0.161414
44	2015-09-01	1500	929	250	0.619333	0.166667
45	2015-10-01	1441	936	222	0.649549	0.154060
46	2015-11-01	1220	797	181	0.653279	0.148361
47	2015-12-01	1348	858	228	0.636499	0.169139
48	2016-01-01	1344	833	218	0.619792	0.162202
49	2016-02-01	1286	828	189	0.643857	0.146967

50	2016-03-01	1360	872	197	0.641176	0.144853
51	2016-04-01	1246	744	225	0.597111	0.180578
52	2016-05-01	1309	826	201	0.631016	0.153552
53	2016-06-01	1487	960	207	0.645595	0.139206
54	2016-07-01	1189	731	214	0.614802	0.179983
55	2016-08-01	1427	934	202	0.654520	0.141556
56	2016-09-01	1306	861	204	0.659265	0.156202
57	2016-10-01	1324	859	215	0.648792	0.162387
58	2016-11-01	1268	834	187	0.657729	0.147476
59	2016-12-01	1370	852	238	0.621898	0.173723
60	2017-01-01	1391	923	211	0.663551	0.151689
61	2017-02-01	1145	753	182	0.657642	0.158952
62	2017-03-01	1248	820	203	0.657051	0.162660
63	2017-04-01	1109	726	198	0.654644	0.178539
64	2017-05-01	1232	812	193	0.659091	0.156656
65	2017-06-01	1246	825	198	0.662119	0.158909
66	2017-07-01	1107	739	153	0.667570	0.138211
67	2017-08-01	1316	860	193	0.653495	0.146657
68	2017-09-01	1135	771	163	0.679295	0.143612
69	2017-10-01	1247	818	187	0.655974	0.149960
70	2017-11-01	1145	754	153	0.658515	0.133624
71	2017-12-01	913	602	124	0.659365	0.135816
72	2018-01-01	1185	783	165	0.660759	0.139241
73	2018-02-01	814	551	110	0.676904	0.135135
74	2018-03-01	1020	689	139	0.675490	0.136275
75	2018-04-01	978	649	144	0.663599	0.147239
76	2018-05-01	1058	686	139	0.648393	0.131380
77	2018-06-01	1074	721	148	0.671322	0.137803
78	2018-07-01	1041	681	163	0.654179	0.156580
79	2018-08-01	1173	727	199	0.619778	0.169650
80	2018-09-01	1028	672	155	0.653696	0.150778
81	2018-10-01	1267	858	176	0.677190	0.138911
82	2018-11-01	924	647	124	0.700216	0.134199
83	2018-12-01	802	555	98	0.692020	0.122195
84	2019-01-01	1040	723	145	0.695192	0.139423
85	2019-02-01	993	645	162	0.649547	0.163142
86	2019-03-01	954	633	129	0.663522	0.135220
87	2019-04-01	1059	700	142	0.661001	0.134089
88	2019-05-01	1080	732	130	0.677778	0.120370
89	2019-06-01	1002	638	154	0.636727	0.153693
90	2019-07-01	968	658	145	0.679752	0.149793
91	2019-08-01	1006	721	136	0.716700	0.135189
92	2019-09-01	1083	737	134	0.680517	0.123730
93	2019-10-01	1116	729	169	0.653226	0.151434
94	2019-11-01	817	549	114	0.671971	0.139535
95	2019-12-01	874	591	134	0.676201	0.153318
96	2020-01-01	1147	796	145	0.693984	0.126417
97	2020-02-01	807	526	108	0.651797	0.133829
98	2020-03-01	473	315	68	0.665962	0.143763
99	2020-04-01	14	7	2	0.500000	0.142857
100	2020-05-01	20	9	6	0.450000	0.300000
101	2020-06-01	107	70	18	0.654206	0.168224

102	2020-07-01	238	156	48	0.655462	0.201681
103	2020-08-01	359	221	67	0.615599	0.186630
104	2020-09-01	486	291	88	0.598765	0.181070
105	2020-10-01	634	415	79	0.654574	0.124606
106	2020-11-01	491	298	76	0.606925	0.154786
107	2020-12-01	537	347	85	0.646182	0.158287
108	2021-01-01	447	278	50	0.621924	0.111857
109	2021-02-01	491	320	66	0.651731	0.134420
110	2021-03-01	532	362	79	0.680451	0.148496

In [41]:

```python
# Create the visualization
fig, ax = plt.subplots(figsize=(14, 6))

# Plot the two lines
ax.plot(monthly_stats["sentenceym_derived"], monthly_stats["prop_black"], label="Black", color="blue", linewidth=2)
ax.plot(monthly_stats["sentenceym_derived"], monthly_stats["prop_white"], label="White", color="orange", linewidth=2)

## Optional Challege

# Shade background for months with fewer than 100 cases
monthly_stats["is_low_case"] = monthly_stats["total_cases"] < 100

for date in monthly_stats.loc[monthly_stats["is_low_case"], "sentenceym_derived"]:
    ax.axvspan(date - pd.DateOffset(days=15), date + pd.DateOffset(days=15), alpha=0.3, color="gray")

# Add vertical line for 12-01-2016 (Foxx took office)
ax.axvline(x=pd.Timestamp("2016-12-01"), color="red", linestyle="--",
           linewidth=1.5, label="Foxx Takes Office (Dec 2016)")

# Labels and legend
ax.set_xlabel("Sentence Date (Month/Year)", fontsize=12)
ax.set_ylabel("Proportion of Defendants", fontsize=12)
ax.yaxis.set_major_formatter(plt.FuncFormatter(lambda y, _: f'{y:.0%}'))
ax.set_title("Proportion of Black and White Defendants Sentenced Over Time", fontsize=14)
ax.legend(loc="best")

plt.tight_layout()
plt.show()
```



In [42]:
```
# Filter for April and May 2020 to examine COVID-19 period
covid_months = monthly_stats[
    (monthly_stats["sentenceym_derived"] >= "2020-04-01") &
    (monthly_stats["sentenceym_derived"] <= "2020-05-31")
]

print("April and May 2020: COVID-19 Impact on Sentencing\n")
print(covid_months[["sentenceym_derived", "total_cases", "black_count", "white_count", "prop_black", "prop_white"]].to_string(index=False))

print(f"\nAverage total cases per month (Apr-May 2020): {covid_months['total_cases'].mean():.1f}")
print(f"Average proportion Black defendants: {covid_months['prop_black'].mean():.1%}")
print(f"Average proportion White defendants: {covid_months['prop_white'].mean():.1%}")

print("\nComparison to Overall Dataset:")
print(f"Overall average cases per month: {monthly_stats['total_cases'].mean():.1f}")
print(f"Overall average proportion Black: {monthly_stats['prop_black'].mean():.1%}")
print(f"Overall average proportion White: {monthly_stats['prop_white'].mean():.1%}")
```

```
April and May 2020: COVID-19 Impact on Sentencing

sentenceym_derived  total_cases  black_count  white_count  prop_black  prop_white
        2020-04-01           14            7            2        0.50    0.142857
        2020-05-01           20            9            6        0.45    0.300000

Average total cases per month (Apr-May 2020): 17.0
Average proportion Black defendants: 47.5%
Average proportion White defendants: 22.1%

Comparison to Overall Dataset:
Overall average cases per month: 1217.7
Overall average proportion Black: 65.1%
Overall average proportion White: 15.3%
```

Mathematical Notation and Interpretation

For each month t, we calculate:

$$\text{Proportion Black}_t = \frac{\sum_{i=1}^{N_t} \mathbf{1}[\text{is\_black\_derived}_i = 1]}{N_t}$$

$$\text{Proportion White}_t = \frac{\sum_{i=1}^{N_t} \mathbf{1}[\text{is\_white\_derived}_i = 1]}{N_t}$$

where N_t is the number of unique cases sentenced in month t, and $\mathbf{1}[\text{is\_racetype\_derived}_i = 1]$ is the indicator function.

Trends: Black defendants consistently represent a larger proportion of those sentenced (~60-70%) compared to White defendants (~15-20%) throughout the period. In April and May 2020, we see unusual patterns likely due to COVID-19 pandemic disruptions. Court operations were most likely reduced, leading to very few total cases being processed and potentially affecting which types of cases and defendants were prioritized for sentencing. The gray shaded regions indicate months with fewer than 100 cases, highlighting this disruption.

1.2: Investigating the first type of disparity: probation versus incaceration (10 points)

One type of disparity beyond who arrives at the sentencing stage is whether the defendant receives probation or incaceration.

According to the codebook, incarceration is indicated by `COMMITMENT_TYPE` == "Illinois Department of Corrections"

Recreate the previous plot but where the y axis represents the difference between the following proportions (can be either Black - White or White - Black but make sure to label), adding a smoothed line:

- Percent of black defendants who are incarcerated out of all black defendants that month/year
- Percent of white defendants who are incarcerated out of all white defendants that month/year

In a markdown cell after, write 1-2 sentences on your observations of trends over time. Do gaps seem to be widening or increasing?

In [43]:
```
# Create indicator for incarceration
df["is_incarcerated"] = (df["COMMITMENT_TYPE"] == "Illinois Department of Corrections").astype(int)

# Filter to Black and White defendants only
black_df = df[df["is_black_derived"] == 1]
white_df = df[df["is_white_derived"] == 1]

# Calculate monthly incarceration rates for Black defendants
black_monthly = black_df.groupby("sentenceym_derived").agg(
    black_total=("CASE_ID", "nunique"),
    black_incarcerated=("is_incarcerated", "sum")
).reset_index()
black_monthly["black_incarceration_rate"] = black_monthly["black_incarcerated"] / black_monthly["black_total"]
black_monthly["black_incarceration_rate_smooth"] = black_monthly["black_incarceration_rate"].rolling(window=6, center=True, min_periods=1).mean()

# Calculate monthly incarceration rates for White defendants
white_monthly = white_df.groupby("sentenceym_derived").agg(
    white_total=("CASE_ID", "nunique"),
    white_incarcerated=("is_incarcerated", "sum")
).reset_index()
white_monthly["white_incarceration_rate"] = white_monthly["white_incarcerated"] / white_monthly["white_total"]
white_monthly["white_incarceration_rate_smooth"] = white_monthly["white_incarceration_rate"].rolling(window=6, center=True, min_periods=1).mean()

# Merge the two dataframes
disparity_df = black_monthly.merge(white_monthly, on="sentenceym_derived", how="outer")

# Calculate the disparity (Black - White)
disparity_df["incarceration_gap"] = disparity_df["black_incarceration_rate"] - disparity_df["white_incarceration_rate"]
disparity_df["smoothed_gap"] = disparity_df["incarceration_gap"].rolling(window=6, center=True, min_periods=1).mean()

# Print the table
print("Monthly Incarceration Rates and Black-White Disparity:")
print(disparity_df[["sentenceym_derived", "black_incarceration_rate", "white_incarceration_rate", "incarceration_gap"]].to_string())
```

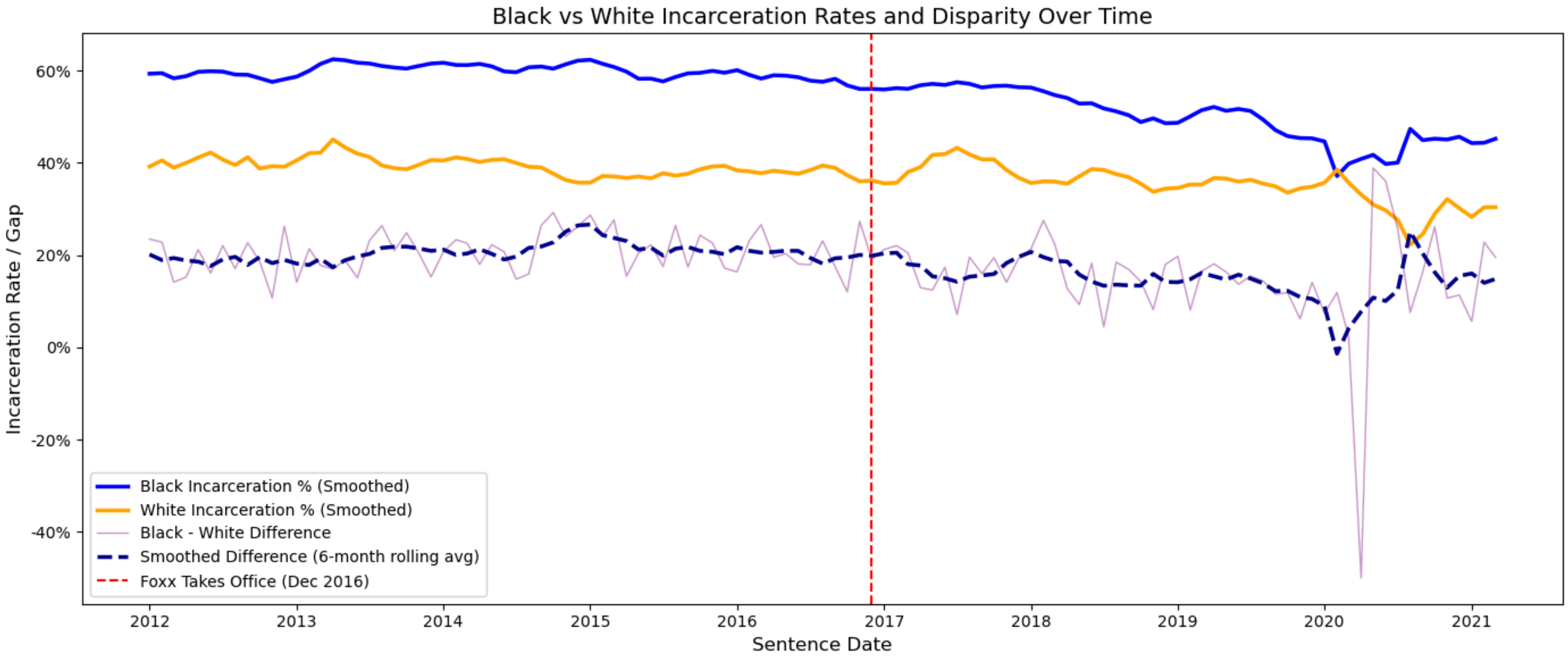
```
Monthly Incarceration Rates and Black-White Disparity:
   sentenceym_derived  black_incarceration_rate  white_incarceration_rate  incarceration_gap
0          2012-01-01                  0.594356                  0.359504           0.234852
1          2012-02-01                  0.601488                  0.373391           0.228097
2          2012-03-01                  0.584466                  0.443137           0.141329
3          2012-04-01                  0.598000                  0.445498           0.152502
4          2012-05-01                  0.539106                  0.327354           0.211752
5          2012-06-01                  0.611203                  0.450199           0.161003
6          2012-07-01                  0.650794                  0.430233           0.220561
7          2012-08-01                  0.608958                  0.438247           0.170711
8          2012-09-01                  0.579412                  0.352381           0.227031
9          2012-10-01                  0.560467                  0.374046           0.186421
10         2012-11-01                  0.535678                  0.428571           0.107107
11         2012-12-01                  0.566239                  0.303371           0.262869
12         2013-01-01                  0.602575                  0.461240           0.141335
13         2013-02-01                  0.643967                  0.430233           0.213734
14         2013-03-01                  0.613360                  0.435780           0.177581
15         2013-04-01                  0.636279                  0.466431           0.169848
16         2013-05-01                  0.625216                  0.436000           0.189216
17         2013-06-01                  0.626976                  0.476190           0.150785
18         2013-07-01                  0.589454                  0.357414           0.232039
19         2013-08-01                  0.612426                  0.348548           0.263878
20         2013-09-01                  0.602881                  0.393162           0.209718
21         2013-10-01                  0.604016                  0.355469           0.248547
22         2013-11-01                  0.604752                  0.400966           0.203785
23         2013-12-01                  0.613924                  0.461279           0.152645
24         2014-01-01                  0.623457                  0.415966           0.207490
25         2014-02-01                  0.643350                  0.409962           0.233388
26         2014-03-01                  0.613293                  0.387681           0.225612
27         2014-04-01                  0.575560                  0.395745           0.179815
28         2014-05-01                  0.602851                  0.380228           0.222623
29         2014-06-01                  0.629490                  0.422222           0.207267
30         2014-07-01                  0.591524                  0.443548           0.147976
31         2014-08-01                  0.577342                  0.418182           0.159160
32         2014-09-01                  0.603352                  0.338645           0.264707
33         2014-10-01                  0.639042                  0.346774           0.292268
34         2014-11-01                  0.612485                  0.371981           0.240505
35         2014-12-01                  0.602247                  0.340249           0.261998
36         2015-01-01                  0.648438                  0.361624           0.286814
37         2015-02-01                  0.624706                  0.384615           0.240090
38         2015-03-01                  0.614519                  0.338078           0.276441
39         2015-04-01                  0.586639                  0.432727           0.153912
40         2015-05-01                  0.570930                  0.366667           0.204264
41         2015-06-01                  0.542240                  0.320000           0.222240
42         2015-07-01                  0.555911                  0.380531           0.175380
43         2015-08-01                  0.626387                  0.361905           0.264482
44         2015-09-01                  0.578041                  0.404000           0.174041
45         2015-10-01                  0.644231                  0.400901           0.243330
46         2015-11-01                  0.618570                  0.392265           0.226304
47         2015-12-01                  0.548951                  0.377193           0.171758
48         2016-01-01                  0.581032                  0.417431           0.163601
49         2016-02-01                  0.602657                  0.370370           0.232287
```

50	2016-03-01	0.611239	0.345178	0.266061
51	2016-04-01	0.581989	0.386667	0.195323
52	2016-05-01	0.571429	0.368159	0.203269
53	2016-06-01	0.591667	0.410628	0.181039
54	2016-07-01	0.575923	0.397196	0.178727
55	2016-08-01	0.582441	0.351485	0.230956
56	2016-09-01	0.566783	0.392157	0.174626
57	2016-10-01	0.566938	0.446512	0.120427
58	2016-11-01	0.610312	0.336898	0.273413
59	2016-12-01	0.505869	0.315126	0.190742
60	2017-01-01	0.529794	0.317536	0.212259
61	2017-02-01	0.583001	0.362637	0.220364
62	2017-03-01	0.559756	0.354680	0.205076
63	2017-04-01	0.584022	0.454545	0.129477
64	2017-05-01	0.600985	0.476684	0.124301
65	2017-06-01	0.552727	0.378788	0.173939
66	2017-07-01	0.548038	0.477124	0.070914
67	2017-08-01	0.568605	0.373057	0.195548
68	2017-09-01	0.595331	0.435583	0.159748
69	2017-10-01	0.563570	0.368984	0.194586
70	2017-11-01	0.553050	0.411765	0.141286
71	2017-12-01	0.571429	0.379032	0.192396
72	2018-01-01	0.553001	0.339394	0.213607
73	2018-02-01	0.548094	0.272727	0.275367
74	2018-03-01	0.590711	0.366906	0.223805
75	2018-04-01	0.516179	0.388889	0.127290
76	2018-05-01	0.502915	0.410072	0.092844
77	2018-06-01	0.533981	0.351351	0.182629
78	2018-07-01	0.480176	0.435583	0.044593
79	2018-08-01	0.551582	0.366834	0.184748
80	2018-09-01	0.523810	0.354839	0.168971
81	2018-10-01	0.477855	0.335227	0.142628
82	2018-11-01	0.452859	0.370968	0.081892
83	2018-12-01	0.445045	0.265306	0.179739
84	2019-01-01	0.528354	0.331034	0.197320
85	2019-02-01	0.488372	0.407407	0.080965
86	2019-03-01	0.529226	0.364341	0.164885
87	2019-04-01	0.561429	0.380282	0.181147
88	2019-05-01	0.532787	0.369231	0.163556
89	2019-06-01	0.487461	0.350649	0.136811
90	2019-07-01	0.478723	0.324138	0.154585
91	2019-08-01	0.511789	0.367647	0.144142
92	2019-09-01	0.503392	0.388060	0.115332
93	2019-10-01	0.449931	0.331361	0.118570
94	2019-11-01	0.395264	0.333333	0.061931
95	2019-12-01	0.409475	0.268657	0.140819
96	2020-01-01	0.453518	0.379310	0.074207
97	2020-02-01	0.507605	0.388889	0.118716
98	2020-03-01	0.463492	0.441176	0.022316
99	2020-04-01	0.000000	0.500000	-0.500000
100	2020-05-01	0.555556	0.166667	0.388889
101	2020-06-01	0.471429	0.111111	0.360317

102	2020-07-01	0.506410	0.250000	0.256410
103	2020-08-01	0.389140	0.313433	0.075707
104	2020-09-01	0.481100	0.318182	0.162918
105	2020-10-01	0.438554	0.177215	0.261339
106	2020-11-01	0.409396	0.302632	0.106764
107	2020-12-01	0.489914	0.376471	0.113443
108	2021-01-01	0.496403	0.440000	0.056403
109	2021-02-01	0.425000	0.196970	0.228030
110	2021-03-01	0.397790	0.202532	0.195258

In [44]:

```
# Create the graphs
fig, ax = plt.subplots(figsize=(14, 6))

# Plot the smoothed lines
ax.plot(disparity_df["sentenceym_derived"], disparity_df["black_incarceration_rate_smooth"],
        label="Black Incarceration % (Smoothed)", color="blue", linewidth=2.5)
ax.plot(disparity_df["sentenceym_derived"], disparity_df["white_incarceration_rate_smooth"],
        label="White Incarceration % (Smoothed)", color="orange", linewidth=2.5)

# Plot the disparity line and its smoothed version
ax.plot(disparity_df["sentenceym_derived"], disparity_df["incarceration_gap"],
        label="Black - White Difference", color="purple", linewidth=1, alpha=0.4)
ax.plot(disparity_df["sentenceym_derived"], disparity_df["smoothed_gap"],
        label="Smoothed Difference (6-month rolling avg)", color="darkblue", linewidth=2.5, linestyle="--")

# Add vertical line for Foxx taking office
ax.axvline(x=pd.Timestamp("2016-12-01"), color="red", linestyle="--",
           linewidth=1.5, label="Foxx Takes Office (Dec 2016)")

# Labels and legend
ax.set_xlabel("Sentence Date", fontsize=12)
ax.set_ylabel("Incarceration Rate / Gap", fontsize=12)
ax.yaxis.set_major_formatter(plt.FuncFormatter(lambda y, _: f'{y:.0%}'))
ax.set_title("Black vs White Incarceration Rates and Disparity Over Time", fontsize=14)
ax.legend(loc="best")

plt.tight_layout()
plt.show()
```



Observations on Incarceration Disparity Trends

The graph shows that Black defendants consistently have higher incarceration rates than White defendants throughout the period, with the gap (Black - White) remaining positive. The disparity appears relatively stable over time, with a slight down trend after State's Attorney Foxx took office in December 2016, but no clear evidence of the gap systematically widening or narrowing. The smoothed trend, by a 6 month rolling avg line, helps visualize that while there is month-to-month variation, the overall disparity persists without dramatic changes in magnitude.

1.3: Investigating mechanisms: incaceration rates by charge

Your colleague sees the previous graph and is worried that the gap could be different---either wider or smaller---if you adjust for the fact that prosecutors have discretion in what crimes to charge defendants with. If white defendants are charged with crimes that tend to receive probation rather than incarceration, that could explain some of the gaps.

In the next questions, you'll begin to investigate this.

In [45]:
```
# Distribution of simplified_offense_derived
print("Overview of offense types in the data:")
print(f"Total unique offense types: {df['simplified_offense_derived'].nunique()}")
print(f"\nMost common offenses overall:")
print(df['simplified_offense_derived'].value_counts().head(15))
```

```
Overview of offense types in the data:
Total unique offense types: 64

Most common offenses overall:
simplified_offense_derived
Narcotics                                   38780
DUI                                         12952
UUW - Unlawful Use of Weapon                11885
Retail Theft                                10546
Driving With Suspended Or Revoked License    8125
Battery                                      7432
Burglary                                     6431
Theft                                        4386
Residential Burglary                         3688
Vehicle-related                              3354
Robbery                                      2997
Escape - Failure to Return                   2637
Failure to Register as a Sex Offender        2301
Other Offense                                2225
Armed Robbery                                2147
Name: count, dtype: int64
```

1.3.1: Find the most common offenses (3 points)

First, create a set of 'frequent offenses' that represent (over the entire period) the union of the 10 offenses Black defendant are most likely to be charged with and the 10 offenses white defendants are most likely to be charged with (might be far less than 20 total if there's a lot of overlap in common charges)

Use the `simplified_offense_derived` for this

In [46]:
```
# Top 10 offenses for Black defendants
top_black_offenses = (black_df.groupby("simplified_offense_derived")["CASE_ID"]
                      .nunique()
                      .sort_values(ascending=False)
                      .head(10)
                      .index.tolist())

# Top 10 offenses for White defendants
top_white_offenses = (white_df.groupby("simplified_offense_derived")["CASE_ID"]
                      .nunique()
                      .sort_values(ascending=False)
                      .head(10)
                      .index.tolist())

# Union of both sets
frequent_offenses = set(top_black_offenses).union(set(top_white_offenses))

print("Top 10 offenses for Black defendants:")
for i, offense in enumerate(top_black_offenses, 1):
    count = black_df[black_df["simplified_offense_derived"] == offense]["CASE_ID"].nunique()
    print(f"  {i}. {offense}: {count} cases")
```

```python
print("\nTop 10 offenses for White defendants:")
for i, offense in enumerate(top_white_offenses, 1):
    count = white_df[white_df["simplified_offense_derived"] == offense]["CASE_ID"].nunique()
    print(f"  {i}. {offense}: {count} cases")

print(f"\nUnion of frequent offenses ({len(frequent_offenses)} unique offenses):")
for i, offense in enumerate(sorted(frequent_offenses), 1):
    print(f"  {i}. {offense}")
```

```
Top 10 offenses for Black defendants:
  1. Narcotics: 29451 cases
  2. UUW - Unlawful Use of Weapon: 9608 cases
  3. Retail Theft: 6812 cases
  4. Battery: 4650 cases
  5. Driving With Suspended Or Revoked License: 4459 cases
  6. DUI: 3842 cases
  7. Burglary: 3798 cases
  8. Theft: 2623 cases
  9. Vehicle-related: 2400 cases
  10. Robbery: 2348 cases

Top 10 offenses for White defendants:
  1. Narcotics: 4923 cases
  2. Retail Theft: 2509 cases
  3. DUI: 1995 cases
  4. Burglary: 1454 cases
  5. Battery: 1448 cases
  6. Driving With Suspended Or Revoked License: 1160 cases
  7. Theft: 1095 cases
  8. Residential Burglary: 717 cases
  9. Other Offense: 569 cases
  10. UUW - Unlawful Use of Weapon: 527 cases

Union of frequent offenses (12 unique offenses):
  1. Battery
  2. Burglary
  3. DUI
  4. Driving With Suspended Or Revoked License
  5. Narcotics
  6. Other Offense
  7. Residential Burglary
  8. Retail Theft
  9. Robbery
  10. Theft
  11. UUW - Unlawful Use of Weapon
  12. Vehicle-related
```

1.3.2: Look at incarceration rates (again just whether incarcerated) by race and offense type for these top offenses (3 points)

Print a wide-format version of the resulting table (so each row is an offense type, one col is black incarceration rate for that offense type; another is the white incarceration rate) and interpret. Which offenses show the largest disparities in judges being less likely to sentence White defendants to incarceration/more likely to offer those defendants probation?

In [47]:
```
# Filter to only frequent offenses
black_frequent = black_df[black_df["simplified_offense_derived"].isin(frequent_offenses)]
white_frequent = white_df[white_df["simplified_offense_derived"].isin(frequent_offenses)]

# Calculate incarceration rates for Black defendants by offense
black_offense_rates = black_frequent.groupby("simplified_offense_derived").agg(
    black_total=("CASE_ID", "nunique"),
    black_incarcerated=("is_incarcerated", "sum")
).reset_index()
black_offense_rates["black_incarceration_rate"] = (black_offense_rates["black_incarcerated"] / black_offense_rates["black_total"])

# Calculate incarceration rates for White defendants by offense
white_offense_rates = white_frequent.groupby("simplified_offense_derived").agg(
    white_total=("CASE_ID", "nunique"),
    white_incarcerated=("is_incarcerated", "sum")
).reset_index()
white_offense_rates["white_incarceration_rate"] = (white_offense_rates["white_incarcerated"] / white_offense_rates["white_total"])

# Merge to create wide-format table
offense_disparity = black_offense_rates.merge(white_offense_rates, on="simplified_offense_derived", how="outer")

# Calculate the disparity (Black - White)
offense_disparity["disparity"] = (offense_disparity["black_incarceration_rate"] - offense_disparity["white_incarceration_rate"])

# Sort by disparity to see largest gaps
offense_disparity_sorted = offense_disparity.sort_values("disparity", ascending=False)

# Wide format table for display
wide_table = offense_disparity_sorted[[
    "simplified_offense_derived",
    "black_incarceration_rate",
    "white_incarceration_rate",
    "disparity",
    "black_total",
    "white_total"
]].copy()

wide_table.columns = ["Offense Type", "Black Incarceration Rate", "White Incarceration Rate", "Disparity (B-W)", "Black N", "White N"]

# Convert certain columns to percentage for legibility
wide_table["Black Incarceration Rate"] = wide_table["Black Incarceration Rate"].apply(lambda x: f"{x:.1%}")
wide_table["White Incarceration Rate"] = wide_table["White Incarceration Rate"].apply(lambda x: f"{x:.1%}")
wide_table["Disparity (B-W)"] = wide_table["Disparity (B-W)"].apply(lambda x: f"{x:.1%}")

# Table output
print("Incarceration Rates by Race and Offense Type (in Wide Format)")
print(wide_table.to_string(index=False))

# Double line spaces into interpretation
print("\n\nInterpretation:")
```

```python
print("-" * 100)
print("Offenses with LARGEST disparities (Black more likely to be incarcerated than White):")
top_disparities = offense_disparity_sorted.head(3)

# Format for output
# Output should look like this: - Offense Type: X.X% gap (Black: Y.Y%, White: Z.Z%)
for _, row in top_disparities.iterrows():
    print(f"  - {row['simplified_offense_derived']}: {row['disparity']:.1%} gap "
          f"(Black: {row['black_incarceration_rate']:.1%}, White: {row['white_incarceration_rate']:.1%})")

print("\nOffenses with SMALLEST or NEGATIVE disparities (White similar or more likely incarcerated):")
bottom_disparities = offense_disparity_sorted.tail(3)
for _, row in bottom_disparities.iterrows():
    print(f"  - {row['simplified_offense_derived']}: {row['disparity']:.1%} gap "
          f"(Black: {row['black_incarceration_rate']:.1%}, White: {row['white_incarceration_rate']:.1%})")
```

```
Incarceration Rates by Race and Offense Type (in Wide Format)
                         Offense Type Black Incarceration Rate White Incarceration Rate Disparity (B-W)  Black N  White N
                            Narcotics                    53.8%                    24.3%           29.5%    29451     4923
                              Battery                    50.9%                    27.6%           23.3%     4650     1448
             UUW - Unlawful Use of Weapon                69.6%                    48.2%           21.4%     9608      527
                         Retail Theft                    59.6%                    45.0%           14.6%     6812     2509
                             Burglary                    66.2%                    53.6%           12.6%     3798     1454
                        Other Offense                    34.1%                    23.0%           11.0%     1242      569
Driving With Suspended Or Revoked License                58.2%                    49.7%            8.6%     4459     1160
                                Theft                    38.7%                    31.9%            6.8%     2623     1095
                  Residential Burglary                   71.8%                    65.4%            6.4%     2282      717
                                  DUI                    40.5%                    34.2%            6.3%     3842     1995
                              Robbery                    65.0%                    58.9%            6.1%     2348      224
                      Vehicle-related                    55.4%                    58.6%           -3.2%     2400      362


Interpretation:
----------------------------------------------------------------------------------------------------
Offenses with LARGEST disparities (Black more likely to be incarcerated than White):
  - Narcotics: 29.5% gap (Black: 53.8%, White: 24.3%)
  - Battery: 23.3% gap (Black: 50.9%, White: 27.6%)
  - UUW - Unlawful Use of Weapon: 21.4% gap (Black: 69.6%, White: 48.2%)

Offenses with SMALLEST or NEGATIVE disparities (White similar or more likely incarcerated):
  - DUI: 6.3% gap (Black: 40.5%, White: 34.2%)
  - Robbery: 6.1% gap (Black: 65.0%, White: 58.9%)
  - Vehicle-related: -3.2% gap (Black: 55.4%, White: 58.6%)
```

1.3.3: Examine whether this changes pre and post change to charging threshold for retail theft (13 points)

One important question is not only whether there are disparities by offense type but also whether these disparities are changing over time.

The SAO, for instance, announced in December of 2016 that they would no longer default to charging retail thefts of under $1,000 as felonies. This change might have (1) decreased disparities or (2) increased disparities, depending on the correlation between race/ethnicity and magnitude of goods stolen: news coverage.

Focusing on `simplified_offense_derived` == "Retail theft." Using a function and/or loop (Dec. 2016 is always excluded as a transition month):

- Compare Black-White disparities before and after the change using a two-month bandwidth (so pre is October and November 2016; post is January and February 2017)
- Compare Black-White disparities before and after the change using a four-month bandwidth (so pre is August- November 2016; post is January - April 2017)
- Compare Black-White disparities using an eight-month bandwidth
- Compare Black-White disparities using a twelve-month bandwidth

---

- Print a table with the results (any organization is fine as long as it's clear)
- Create a bar chart where the x axis represents different bandwidths (2, 4, etc); the y axis the size of the Black-White gap in whether the defendant receives incarceration, and for each of the x axis points, you have one shaded bar representing "before" the change, another representing "after" the change (make sure that before is ordered before after and the bandwidths are from smallest to largest)

Note: for each of the bandwidths include dates the entire month (e.g., for the first, include not only 02-01-2017 but everything up through 02-28-2017; easiest way is for the subsetting to use the rounded `sentenceym_derived`). Also make sure to only include white or black defendants.

Extra credit: because the bandwidths have different sample sizes, a better viz incorporates measures of uncertainty. Add standard errors to the estimates using the formula: $(\frac{p(1-p)}{n})^{0.5}$ where p is the gap and N is the number of cases in each bandwidth period

In [48]:
```python
def calculate_disparity(data, pre_start, pre_end, post_start, post_end):
    # Pre-period data
    pre_data = data[(data["sentenceym_derived"] >= pre_start) &
                    (data["sentenceym_derived"] <= pre_end)]

    # Post-period data
    post_data = data[(data["sentenceym_derived"] >= post_start) &
                     (data["sentenceym_derived"] <= post_end)]

    results = {}

    for period_name, period_data in [("pre", pre_data), ("post", post_data)]:
        # Black defendants
        black_data = period_data[period_data["is_black_derived"] == 1]
        black_n = black_data["CASE_ID"].nunique()
        black_incarcerated = black_data["is_incarcerated"].sum()
        black_rate = black_incarcerated / black_n if black_n > 0 else np.nan

        # White defendants
        white_data = period_data[period_data["is_white_derived"] == 1]
        white_n = white_data["CASE_ID"].nunique()
        white_incarcerated = white_data["is_incarcerated"].sum()
        white_rate = white_incarcerated / white_n if white_n > 0 else np.nan
```

```
        # Calculate gap and standard error
        gap = black_rate - white_rate
        total_n = black_n + white_n

        # Standard error formula = sqrt(p(1-p)/n) where p is the gap (sqrt same as **0.5)
        se = np.sqrt((gap * (1 - gap)) / total_n) if total_n > 0 else np.nan

        results[period_name] = {
            "black_rate": black_rate,
            "white_rate": white_rate,
            "gap": gap,
            "se": se,
            "black_n": black_n,
            "white_n": white_n,
            "total_n": total_n
        }

    return results
```

In [49]:

```
# Filter to retail theft cases and Black/White defendants only
retail_df = df[(df["simplified_offense_derived"] == "Retail Theft") &
               ((df["is_black_derived"] == 1) | (df["is_white_derived"] == 1))].copy()

# Define the policy change date (December 2016 is excluded as a transition month)
policy_date = pd.Timestamp("2016-12-01")

# Define bandwidths and their date ranges
bandwidths = {
    2: {
        "pre_start": pd.Timestamp("2016-10-01"),
        "pre_end": pd.Timestamp("2016-11-01"),
        "post_start": pd.Timestamp("2017-01-01"),
        "post_end": pd.Timestamp("2017-02-01")
    },
    4: {
        "pre_start": pd.Timestamp("2016-08-01"),
        "pre_end": pd.Timestamp("2016-11-01"),
        "post_start": pd.Timestamp("2017-01-01"),
        "post_end": pd.Timestamp("2017-04-01")
    },
    8: {
        "pre_start": pd.Timestamp("2016-04-01"),
        "pre_end": pd.Timestamp("2016-11-01"),
        "post_start": pd.Timestamp("2017-01-01"),
        "post_end": pd.Timestamp("2017-08-01")
    },
    12: {
        "pre_start": pd.Timestamp("2015-12-01"),
        "pre_end": pd.Timestamp("2016-11-01"),
        "post_start": pd.Timestamp("2017-01-01"),
        "post_end": pd.Timestamp("2017-12-01")
    }
```

```
}

# Calculate disparities for each bandwidth
results_list = []
for bw, dates in bandwidths.items():
    result = calculate_disparity(retail_df, dates["pre_start"], dates["pre_end"],dates["post_start"], dates["post_end"])
    results_list.append({
        "Bandwidth": bw,
        "Period": "Pre",
        "Black Rate": result["pre"]["black_rate"],
        "White Rate": result["pre"]["white_rate"],
        "Gap (B-W)": result["pre"]["gap"],
        "SE": result["pre"]["se"],
        "Black N": result["pre"]["black_n"],
        "White N": result["pre"]["white_n"],
        "Total N": result["pre"]["total_n"]
    })
    results_list.append({
        "Bandwidth": bw,
        "Period": "Post",
        "Black Rate": result["post"]["black_rate"],
        "White Rate": result["post"]["white_rate"],
        "Gap (B-W)": result["post"]["gap"],
        "SE": result["post"]["se"],
        "Black N": result["post"]["black_n"],
        "White N": result["post"]["white_n"],
        "Total N": result["post"]["total_n"]
    })

results_df = pd.DataFrame(results_list)

# Print the results table
print("BLACK-WHITE INCARCERATION DISPARITY FOR RETAIL THEFT: PRE vs POST POLICY CHANGE")
print(f"\nPolicy Change: December 2016 (raising felony threshold for retail theft to $1,000)")
print("Note: December 2016 excluded as transition month\n")

# Format table
display_df = results_df.copy()
display_df["Black Rate"] = display_df["Black Rate"].apply(lambda x: f"{x:.1%}")
display_df["White Rate"] = display_df["White Rate"].apply(lambda x: f"{x:.1%}")
display_df["Gap (B-W)"] = display_df["Gap (B-W)"].apply(lambda x: f"{x:.1%}")
display_df["SE"] = display_df["SE"].apply(lambda x: f"{x:.3f}")

print(display_df.to_string(index=False))

# Summary by bandwidth
print("\n" + "-" * 100)
print("SUMMARY: Gap Change by Bandwidth")
for bw in [2, 4, 8, 12]:
    pre_gap = results_df[(results_df["Bandwidth"] == bw) & (results_df["Period"] == "Pre")]["Gap (B-W)"].values[0]
    post_gap = results_df[(results_df["Bandwidth"] == bw) & (results_df["Period"] == "Post")]["Gap (B-W)"].values[0]
    change = post_gap - pre_gap
```

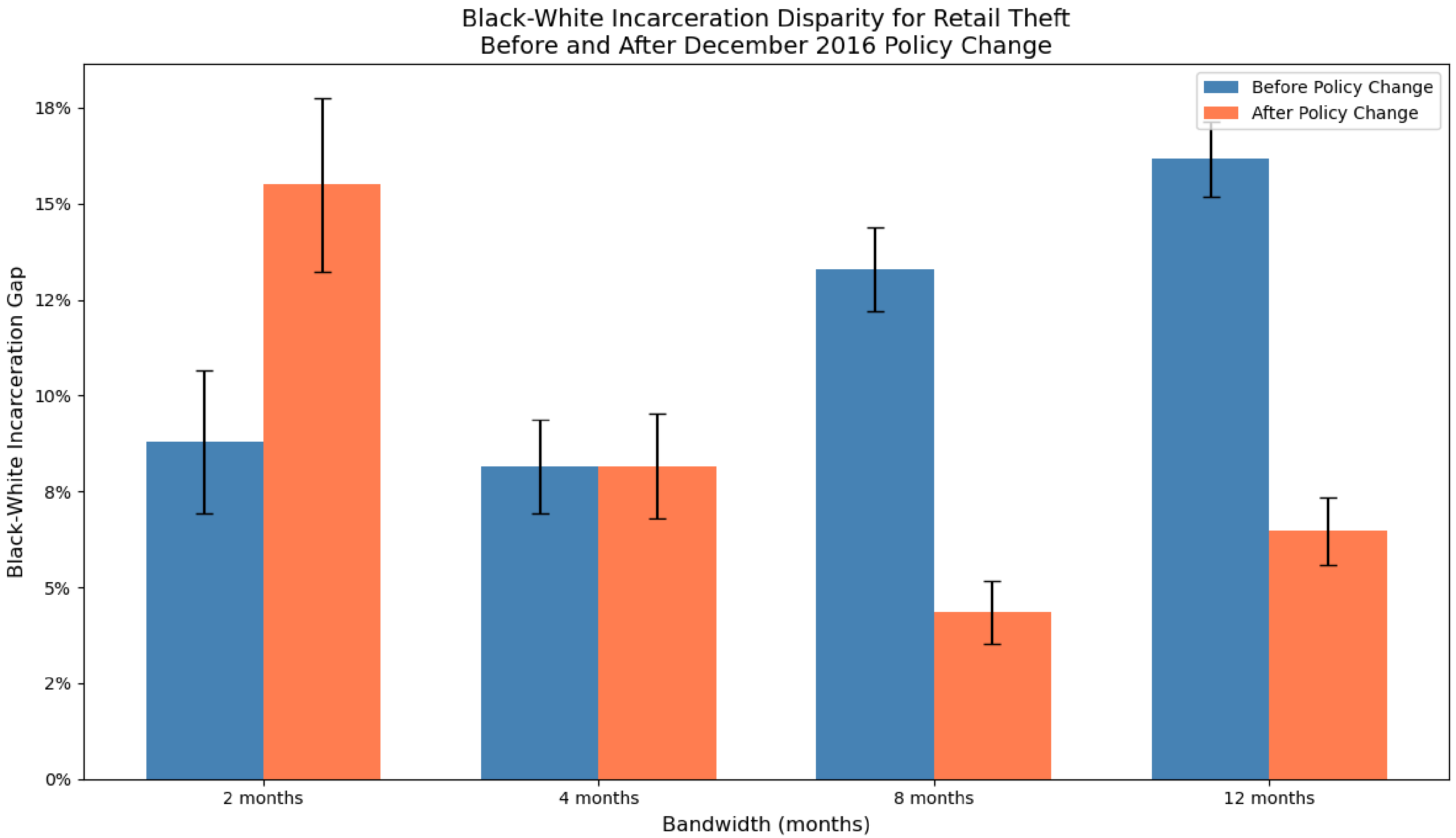
```
    direction = "narrowed" if change < 0 else "widened"
    print(f"  {bw}-month bandwidth: Pre gap = {pre_gap:.1%}, Post gap = {post_gap:.1%}, "
          f"Change = {change:+.1%} ({direction})")
```

```
BLACK-WHITE INCARCERATION DISPARITY FOR RETAIL THEFT: PRE vs POST POLICY CHANGE

Policy Change: December 2016 (raising felony threshold for retail theft to $1,000)
Note: December 2016 excluded as transition month

 Bandwidth Period Black Rate White Rate Gap (B-W)    SE  Black N  White N  Total N
         2    Pre      62.6%      53.8%      8.8% 0.019      182       52      234
         2   Post      37.2%      21.7%     15.5% 0.023      188       69      257
         4    Pre      55.9%      47.7%      8.2% 0.012      376      130      506
         4   Post      41.2%      33.0%      8.2% 0.014      284      112      396
         8    Pre      56.9%      43.6%     13.3% 0.011      703      266      969
         8   Post      49.3%      45.0%      4.3% 0.008      458      180      638
        12    Pre      58.1%      41.9%     16.2% 0.010     1057      384     1441
        12   Post      51.1%      44.6%      6.5% 0.009      560      213      773

----------------------------------------------------------------------------------------------------
SUMMARY: Gap Change by Bandwidth
  2-month bandwidth: Pre gap = 8.8%, Post gap = 15.5%, Change = +6.7% (widened)
  4-month bandwidth: Pre gap = 8.2%, Post gap = 8.2%, Change = +0.0% (widened)
  8-month bandwidth: Pre gap = 13.3%, Post gap = 4.3%, Change = -8.9% (narrowed)
  12-month bandwidth: Pre gap = 16.2%, Post gap = 6.5%, Change = -9.7% (narrowed)
```

In [50]:

```
# Create the bar chart with error bars (extra credit)
fig, ax = plt.subplots(figsize=(12, 7))

# Prepare data for plotting
bandwidths_list = [2, 4, 8, 12]
x = np.arange(len(bandwidths_list))
width = 0.35

pre_gaps = []
post_gaps = []
pre_ses = []
post_ses = []

for bw in bandwidths_list:
    pre_row = results_df[(results_df["Bandwidth"] == bw) & (results_df["Period"] == "Pre")]
    post_row = results_df[(results_df["Bandwidth"] == bw) & (results_df["Period"] == "Post")]
    pre_gaps.append(pre_row["Gap (B-W)"].values[0])
    post_gaps.append(post_row["Gap (B-W)"].values[0])
    pre_ses.append(pre_row["SE"].values[0])
    post_ses.append(post_row["SE"].values[0])

# Create bars with error bars
bars1 = ax.bar(x - width/2, pre_gaps, width, label="Before Policy Change",
               color="steelblue", yerr=pre_ses, capsize=5, error_kw={"linewidth": 1.5})
bars2 = ax.bar(x + width/2, post_gaps, width, label="After Policy Change",
               color="coral", yerr=post_ses, capsize=5, error_kw={"linewidth": 1.5})
```

```python
# Customize the chart
ax.set_xlabel("Bandwidth (months)", fontsize=12)
ax.set_ylabel("Black-White Incarceration Gap", fontsize=12)
ax.set_title("Black-White Incarceration Disparity for Retail Theft\nBefore and After December 2016 Policy Change",
             fontsize=14)
ax.set_xticks(x)
ax.set_xticklabels([f"{bw} months" for bw in bandwidths_list])
ax.legend(loc="upper right")

# Add a horizontal line at y=0 for reference
ax.axhline(y=0, color="black", linestyle="-", linewidth=0.5)

# Format y-axis as percentage
ax.yaxis.set_major_formatter(plt.FuncFormatter(lambda y, _: f"{y:.0%}"))

plt.tight_layout()
plt.show()
```



1.3.3.1: Interpret the results (2 points)

Write a two-sentence interpretation of the results. What might this show about how people on both sides of the issue---those who argued that the retail theft policy change would narrow disparities; those who argued that the change may widen disparities--could support their claims?

Answer

Across bandwidths, the estimated Black–White incarceration gap after the policy change is higher in the very short window, about the same at 4 months, but much lower in longer windows. This pattern lets narrows disparities advocates point to the 8–12 month results as evidence of a meaningful post-policy reduction, while the widen disparities advocates can cite the 2-month spike to argue the change initially coincided with larger disparities or that the conclusion depends heavily on the chosen time.