In [32]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

Import data and filter rows with 500+ pitches thrown in consecutive years

Caveats with data

- Haven't done work on adjusting for pitches against LHB/RHB. Or taking into account categorical nature of pitch types
- Stuff+,location+,pitching+ data only goes back to 2020

Main Questions

- Does change in arsenal (quantified by EMD/other metrics) have a significant impact on performance?
- Conditional on stuff+ increasing/decreasing, is change a significant factor
- Is there a relationship between large changes and other variables? Location, stuff, innings pitched?

In [50]:
```
df = pd.read_csv('pitcher_year_to_year_emd_with_siera_and_stuffplus_teamfilled.csv')
df = df.loc[(df['n_pitches_year1'] > 500) & (df['n_pitches_year2'] > 500)]
print(df.shape)
df.head()
```

```
(2163, 132)
```

Out[50]:

	pitcher	start_year	end_year	emd_whitened_sliced	n_pitches_year1	n_pitches_year2	y1_velo_CH	y1_hb_CH	y1_vb_CH	y2_velo_CH	.
1	A.J. Cole	2017	2018	0.475684	944	871	86.090476	11.076190	9.465079	86.460870	.
6	A.J. Minter	2018	2019	0.459521	1003	588	86.231250	-13.125000	-0.409375	86.060000	.
9	A.J. Minter	2021	2022	0.311691	876	1111	87.205983	-16.588034	3.091453	87.585075	.
10	A.J. Minter	2022	2023	0.284961	1111	1060	87.585075	-16.197512	3.389055	86.647973	.
11	A.J. Minter	2023	2024	0.467814	1060	522	86.647973	-15.368919	5.068243	86.710753	.

5 rows × 132 columns

Example of Small Change - Colin Rea - 2023 to 2024

Year	Pitch Type	#	# RHB	# LHB	%	MPH
2025	Four Seamer	1,052	417	635	41.5	93.9
2025	Split Finger	305	28	277	12.0	87.3
2025	Sinker	267	222	45	10.5	93.0
2025	Slider	256	200	56	10.1	85.2
2025	Sweeper	234	223	11	9.2	82.8
2025	Curveball	231	13	218	9.1	80.3
2025	Cutter	188	74	114	7.4	88.2
2024	Sinker	818	499	319	30.9	92.3
2024	Four Seamer	516	232	284	19.5	93.0
2024	Cutter	515	193	322	19.4	87.6
2024	Sweeper	446	279	167	16.8	82.0
2024	Split Finger	235	48	187	8.9	86.6
2024	Curveball	120	59	61	4.5	78.9
2023	Sinker	606	407	199	30.1	92.6
2023	Cutter	533	266	267	26.4	86.7
2023	Four Seamer	385	151	234	19.1	93.2
2023	Sweeper	221	151	70	11.0	83.2
2023	Curveball	159	75	84	7.9	78.8
2023	Split Finger	112	8	104	5.6	86.0

Key Points

- Sinker/Cutter usage and velocity practically unchanged
- Main Shift is less cutter usage in 2024 that equates to more sweeper usage.

- Everything is a similar speed. His velo difference between the cutter and sweeper are not very different, so EMD does not see that change as very large.

In [30]:
```
df.sort_values('emd_whitened_sliced', ascending=True).head(3)
```

Out[30]:

	pitcher	start_year	end_year	emd_whitened_sliced	n_pitches_year1	n_pitches_year2	y1_velo_CH	y1_hb_CH	y1_vb_CH	y2_velo_CH
992	Colin Rea	2023	2024	0.070093	2016	2650	NaN	NaN	NaN	NaN
3880	Sandy Alcantara	2022	2023	0.073045	3261	2721	91.773154	16.772931	3.733333	91.129704
2895	Luis Castillo	2020	2021	0.074374	1153	3164	88.210405	16.459827	0.964162	88.333817

3 rows × 132 columns

Example of Large Change - Neil Ramirez - 2017 to 2018

Year	Pitch	Team	Hand	#	MPH	Vertical Drop	vs. Comparable	Horizontal Break
2019	Curveball	TOR	R	61	79.4	52.6	-1.7	8.6 GLV
2019	Slider	TOR	R	143	84.6	38.6	2.0	6.3 GLV
2019	4-Seam Fastball	TOR	R	257	94.4	11.8	3.2	6.4 ARM
2018	Slider	CLE	R	319	85.9	36.8	2.1	6.0 GLV
2018	4-Seam Fastball	CLE	R	252	95.2	12.1	2.7	5.8 ARM
2018	Sinker	CLE	R	175	95.4	12.5	-7.2	8.5 ARM
2017	Slider	NYM	R	200	84.7	34.0	-0.9	5.5 GLV
2017	4-Seam Fastball	NYM	R	119	93.1	9.3	4.8	2.2 ARM
2017	Sinker	NYM	R	191	93.2	12.8	-6.7	9.0 ARM
2017	Changeup	NYM	R	5	86.9	18.3	-8.6	13.5 ARM
2017	Curveball	NYM	R	125	78.4	50.4	-3.0	4.0 GLV

Key Points

- Helps to look at movement differences to see how large of a change this is.
- Completely throws away the curveball (20% usage!) which has a very different shape than the rest of the pitches
- Slider: +10% usage, thrown 1.2mph harder, with more movement
- 4S Fastball: +15% usage and +2 mph! Verty fastball
- Throwing sinker harder with similar movement

In [31]:
```
df.sort_values('emd_whitened_sliced', ascending=False).head(3)
```

Out[31]:

	pitcher	start_year	end_year	emd_whitened_sliced	n_pitches_year1	n_pitches_year2	y1_velo_CH	y1_hb_CH	y1_vb_CH	y2_velo_CH
3328	Neil Ramírez	2017	2018	1.013255	640	749	86.900000	13.500000	17.880000	NaN
2331	Jordan Hicks	2023	2024	0.981314	1113	1958	NaN	NaN	NaN	NaN
1803	Ian Kennedy	2018	2019	0.922660	2054	1053	85.047867	12.822275	11.974408	87.947368

3 rows × 132 columns

Breakdown over a career - Charlie Morton

- Largest changes are when he switches teams. We see stuff gets worse, but performance still improves.

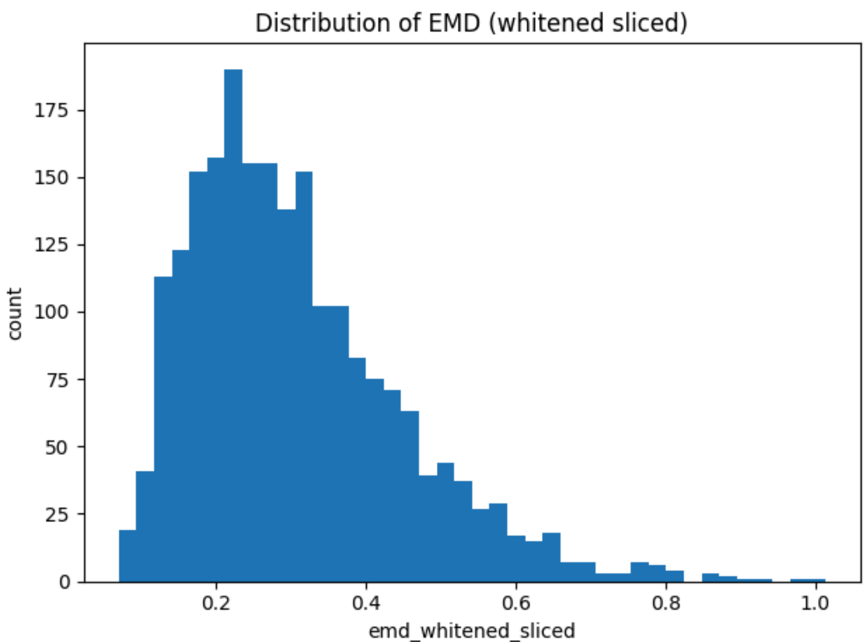
In [24]:
```
df[['pitcher', 'start_year', 'end_year', 'emd_whitened_sliced', 'team_y1', 'team_y2', 'diff_siera', 'diff_stuff+']].loc[df['pi
```

Out[24]:

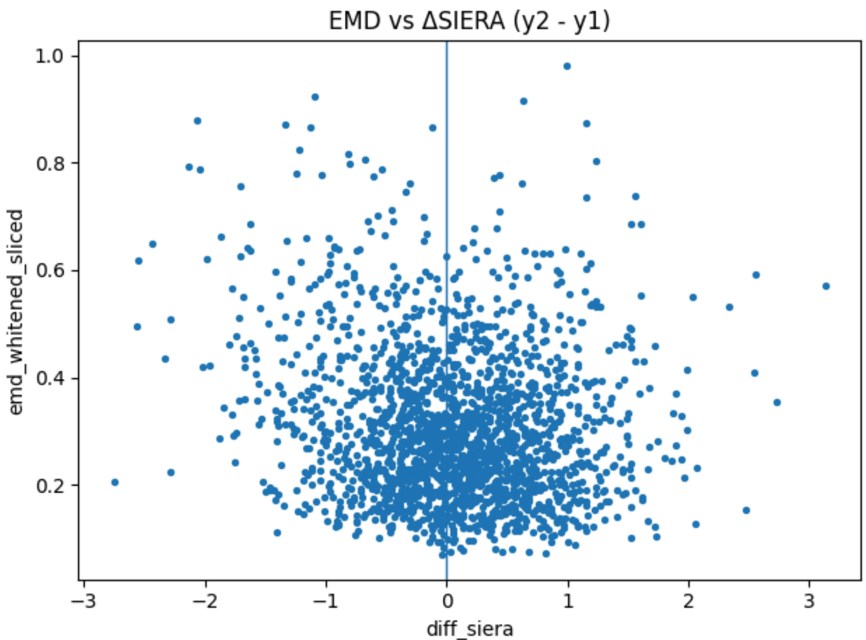
	pitcher	start_year	end_year	emd_whitened_sliced	team_y1	team_y2	diff_siera	diff_stuff+
799	Charlie Morton	2017	2018	0.293683	HOU	HOU	-0.18	NaN
800	Charlie Morton	2018	2019	0.461789	HOU	TBR	0.02	NaN
801	Charlie Morton	2019	2020	0.335828	TBR	TBR	0.43	NaN
802	Charlie Morton	2020	2021	0.607121	TBR	ATL	-0.44	-4.765562
803	Charlie Morton	2021	2022	0.115545	ATL	ATL	-0.05	-4.406869
804	Charlie Morton	2022	2023	0.134347	ATL	ATL	0.96	-4.279511
805	Charlie Morton	2023	2024	0.162618	ATL	ATL	-0.38	-1.133465
806	Charlie Morton	2024	2025	0.215717	ATL	- - -	0.33	2.872835

Plots and Charts and Graphs

In [34]:
```
plt.figure()
plt.hist(df["emd_whitened_sliced"].dropna(), bins=40)
plt.title("Distribution of EMD (whitened sliced)")
plt.xlabel("emd_whitened_sliced")
plt.ylabel("count")
plt.tight_layout()
```

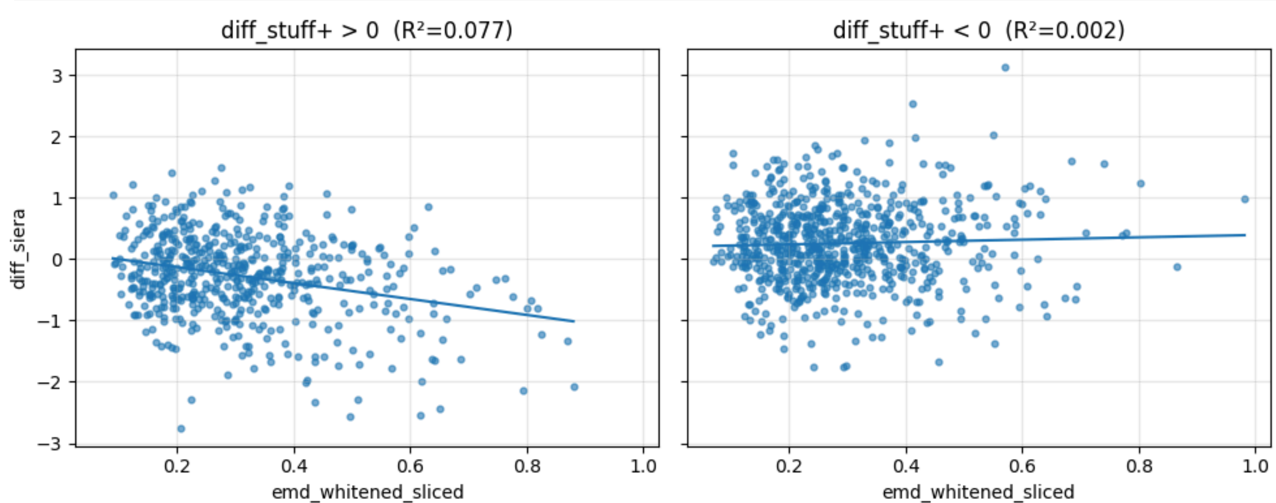


In [35]:
```
plt.figure()
plt.scatter(df["diff_siera"], df["emd_whitened_sliced"], s=8)
plt.title("EMD vs ΔSIERA (y2 - y1)")
plt.xlabel("diff_siera")
plt.ylabel("emd_whitened_sliced")
plt.axvline(0, linewidth=1)
plt.tight_layout()
```



In [ ]:

```python
pos = df["diff_stuff+"] > 0
neg = df["diff_stuff+"] < 0

fig, ax = plt.subplots(1, 2, figsize=(10, 4), sharex=True, sharey=True)

def panel(a, msk, title):
    x = df.loc[msk, "emd_whitened_sliced"].to_numpy()
    y = df.loc[msk, "diff_siera"].to_numpy()
    ok = np.isfinite(x) & np.isfinite(y); x, y = x[ok], y[ok]

    b1, b0 = np.polyfit(x, y, 1)
    yhat = b1*x + b0
    r2 = 1 - ((y-yhat)**2).sum() / ((y-y.mean())**2).sum()

    a.scatter(x, y, s=12, alpha=.6)
    xx = np.linspace(x.min(), x.max(), 100)
    a.plot(xx, b1*xx + b0)
    a.set_title(f"{title}  (R²={r2:.3f})")
    a.set_xlabel("emd_whitened_sliced"); a.grid(alpha=.3)

panel(ax[0], pos, "diff_stuff+ > 0")
ax[0].set_ylabel("diff_siera")

panel(ax[1], neg, "diff_stuff+ < 0")

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



In [46]:

```python
emd_col = "emd_whitened_sliced"
```

```
m = df[df["team_y1"].ne(df["team_y2"]) & df[emd_col].notna()].copy()

# ---- table: destination org (yr2) average EMD when pitcher changed teams
tbl = (m.groupby("team_y2")[emd_col]
         .agg(mean_emd="mean", median_emd="median", n="size")
         .sort_values(["mean_emd","n"], ascending=[False, False]))
tbl
```

Out[46]:

team_y2	mean_emd	median_emd	n
SFG	0.416062	0.383295	13
MIL	0.385921	0.390318	14
HOU	0.385325	0.347867	13
MIA	0.381238	0.388965	9
MIN	0.363012	0.327107	8
TBR	0.355825	0.304438	18
COL	0.348893	0.352520	8
SEA	0.336470	0.342771	10
ATL	0.332129	0.317271	13
OAK	0.323641	0.329407	15
NYY	0.321883	0.312938	18
TEX	0.315342	0.322075	27
BAL	0.312124	0.261795	12
BOS	0.310805	0.278047	12
ARI	0.307805	0.288094	18
LAD	0.307728	0.285108	14
PHI	0.307236	0.262116	17
KCR	0.306945	0.286128	17
CLE	0.305865	0.281952	14
LAA	0.305468	0.283100	15
NYM	0.301525	0.265110	21
PIT	0.298519	0.280798	17
CHC	0.296807	0.269047	19
SDP	0.296390	0.309538	19
CIN	0.292626	0.272254	16
TOR	0.286477	0.243541	18
CHW	0.285308	0.246737	17
DET	0.267540	0.288215	11
WSN	0.253327	0.242122	10
ATH	0.252581	0.195247	7
STL	0.222022	0.148444	7

In [49]:

```
emd_col = "emd_whitened_sliced"

d = df[df[emd_col].notna()].copy()
d["team_change"] = d["team_y1"].ne(d["team_y2"]).map({True: "Team change", False: "No team change"})

# summary table
tbl = (d.groupby("team_change")[emd_col]
        .agg(mean="mean", median="median", n="size", std="std")
        .reindex(["No team change", "Team change"]))
tbl
```

Out[49]:

	mean	median	n	std
team_change				
No team change	0.303201	0.277157	1219	0.139652
Team change	0.310918	0.280812	944	0.146202