

```

import pandas as pd
import numpy as np
from scipy.stats import ttest_ind, kruskal
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset
file_path = "/content/hard_fix_september_11_2024.csv"
data = pd.read_csv(file_path)

# Convert year and month into a single date column for better analysis
data['Date'] = pd.to_datetime(data['Year'].astype(str) + '-' + data['Month'].astype(str).replace('nan', '01') + '-01', errors='coerce')

# Filter rows where Total Disconnections is non-null and greater than zero
data = data.dropna(subset=['Total Disconnections'])
data = data[data['Total Disconnections'] > 0]

# Create period columns: Pre-Covid, Covid, Post-Covid
data['Period'] = pd.cut(
    data['Year'],
    bins=[2017, 2019, 2020, 2023],
    labels=['Pre-Covid', 'Covid', 'Post-Covid']
)

# National Analysis: Pre vs Post Covid using T-Test
pre_covid = data[data['Period'] == 'Pre-Covid']['Disconnection Rate']
post_covid = data[data['Period'] == 'Post-Covid']['Disconnection Rate']

t_stat, p_value = ttest_ind(pre_covid, post_covid, nan_policy='omit')
print(f"T-Test Results: T-Stat = {t_stat}, P-Value = {p_value}")

➡ T-Test Results: T-Stat = 8.878595462120114, P-Value = 8.203098558914055e-19

# Kruskal-Wallis Test for all periods
kruskal_stat, kruskal_p_value = kruskal(
    data[data['Period'] == 'Pre-Covid']['Disconnection Rate'],
    data[data['Period'] == 'Covid']['Disconnection Rate'],
    data[data['Period'] == 'Post-Covid']['Disconnection Rate']
)
print(f"Kruskal-Wallis Test Results: Stat = {kruskal_stat}, P-Value = {kruskal_p_value}")

➡ Kruskal-Wallis Test Results: Stat = nan, P-Value = nan

# State-wise Analysis
state_summary = data.groupby(['State', 'Period'])['Disconnection Rate'].mean().unstack()
state_summary['Percent Change'] = ((state_summary['Post-Covid'] - state_summary['Pre-Covid']) /
                                   state_summary['Pre-Covid']) * 100

print("State-wise Summary:")
print(state_summary)

```

```

➡ State-wise Summary:

```

Period State	Pre-Covid	Covid	Post-Covid	Percent Change
Alabama	0.075043	0.056960	0.052646	-29.844886
Arizona	NaN	0.000534	0.003184	NaN
California	0.006716	0.006495	0.001158	-82.754065
Colorado	0.005095	0.002705	0.002250	-55.845360
Connecticut	0.009787	0.007945	0.005618	-42.599388
Florida	0.022871	0.013661	0.015343	-32.912364
Georgia	NaN	0.007258	0.007939	NaN
Hawaii	0.001057	NaN	0.001356	28.251385
Idaho	0.019136	0.008729	0.014007	-26.801617
Illinois	0.002331	0.002845	0.004215	80.835071
Indiana	0.006931	0.004788	0.005378	-22.408848
Iowa	0.004032	0.001639	0.003098	-23.162861
Kansas	NaN	0.000166	0.001509	NaN
Kentucky	0.007227	0.005168	0.005218	-27.795333
Louisiana	NaN	NaN	NaN	NaN
Maine	0.003413	0.000236	0.003572	4.663478
Maryland	0.004489	0.001749	0.003422	-23.767662

Massachusetts	0.003006	0.001128	0.002476	-17.639074
Michigan	0.012789	0.003701	0.004753	-62.835905
Minnesota	0.004700	0.001575	0.003781	-19.553329
Missouri	0.013653	0.009996	0.011747	-13.961000
Nebraska	NaN	0.003538	0.004847	NaN
New Hampshire	0.002455	0.000294	0.002512	2.323934
New Jersey	0.022273	0.017436	0.000905	-95.936964
New Mexico	NaN	NaN	0.003374	NaN
New York	0.002063	0.001013	0.002127	3.112159
North Carolina	0.004984	0.009771	0.006151	23.417862
North Dakota	0.000222	0.000786	0.002357	963.894319
Ohio	0.005433	0.003436	0.003741	-31.155385
Oklahoma	0.007906	0.007024	0.029542	273.642934
Oregon	0.002665	0.002096	0.001512	-43.253128
Pennsylvania	0.004457	0.003571	0.007311	64.011515
Rhode Island	NaN	0.000060	0.002521	NaN
South Carolina	0.008493	0.008732	0.007387	-13.023669
South Dakota	NaN	0.002645	0.001520	NaN
Texas	0.018449	0.016204	0.019573	6.094998
Utah	0.000234	0.003925	0.001912	717.995975
Vermont	0.003893	0.006150	0.003046	-21.747561
Virginia	0.006656	0.002958	0.006719	0.960167
Washington	0.006119	0.002630	0.001325	-78.341198
Washington, D.C.	0.002680	0.001392	0.002428	-9.416241
Wisconsin	0.003047	0.000067	0.003822	25.423143

```
<ipython-input-10-97a8219487fb>:2: FutureWarning: The default of observed=False is deprecated and will be changed to True
state_summary = data.groupby(['State', 'Period'])['Disconnection Rate'].mean().unstack()
```

```
# Plot State-wise changes
```

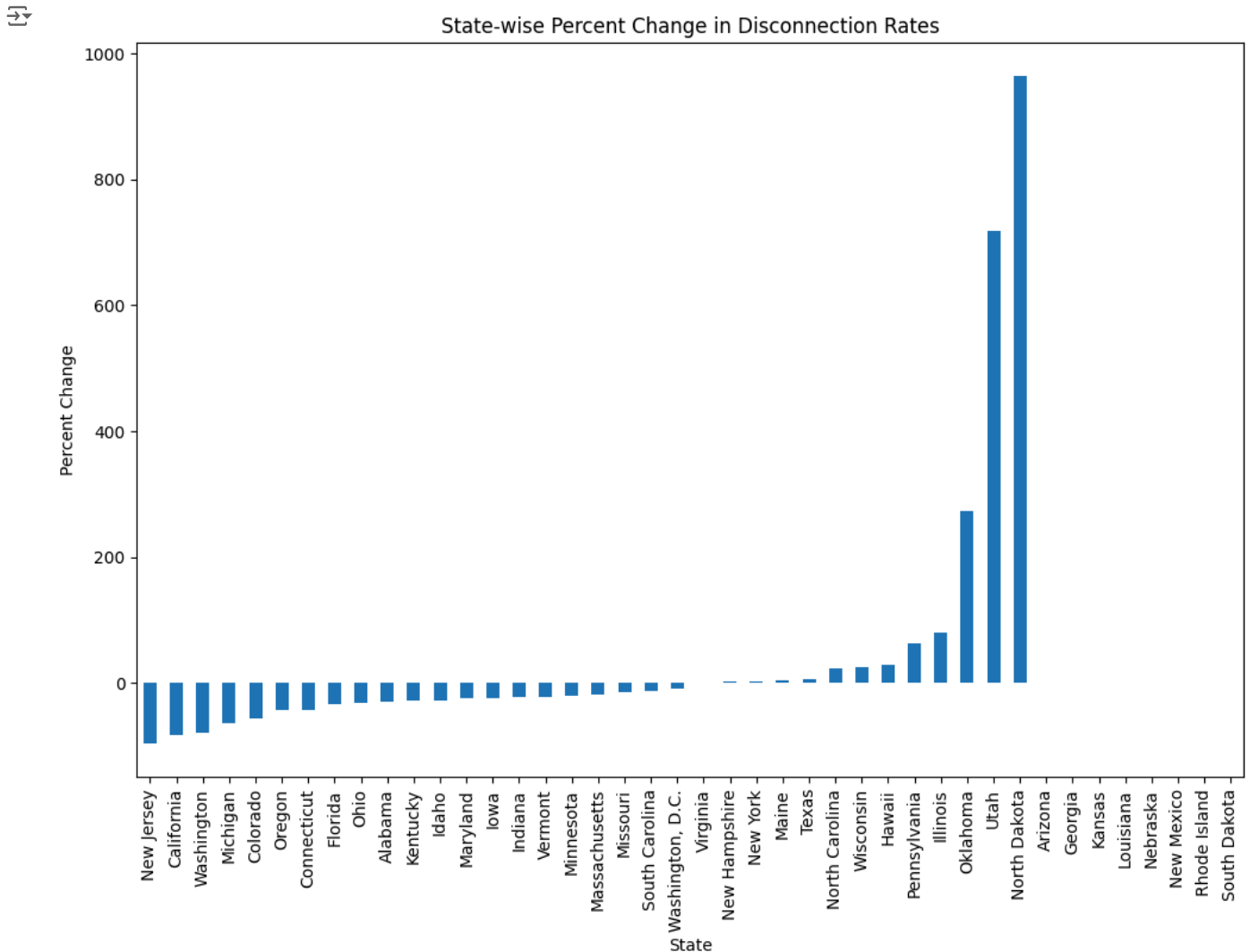
```
plt.figure(figsize=(12, 8))
```

```
state_summary['Percent Change'].sort_values().plot(kind='bar', title='State-wise Percent Change in Disconnection Rates')
```

```
plt.ylabel('Percent Change')
```

```
plt.xlabel('State')
```

```
plt.show()
```



```
# Provider Type Analysis
```

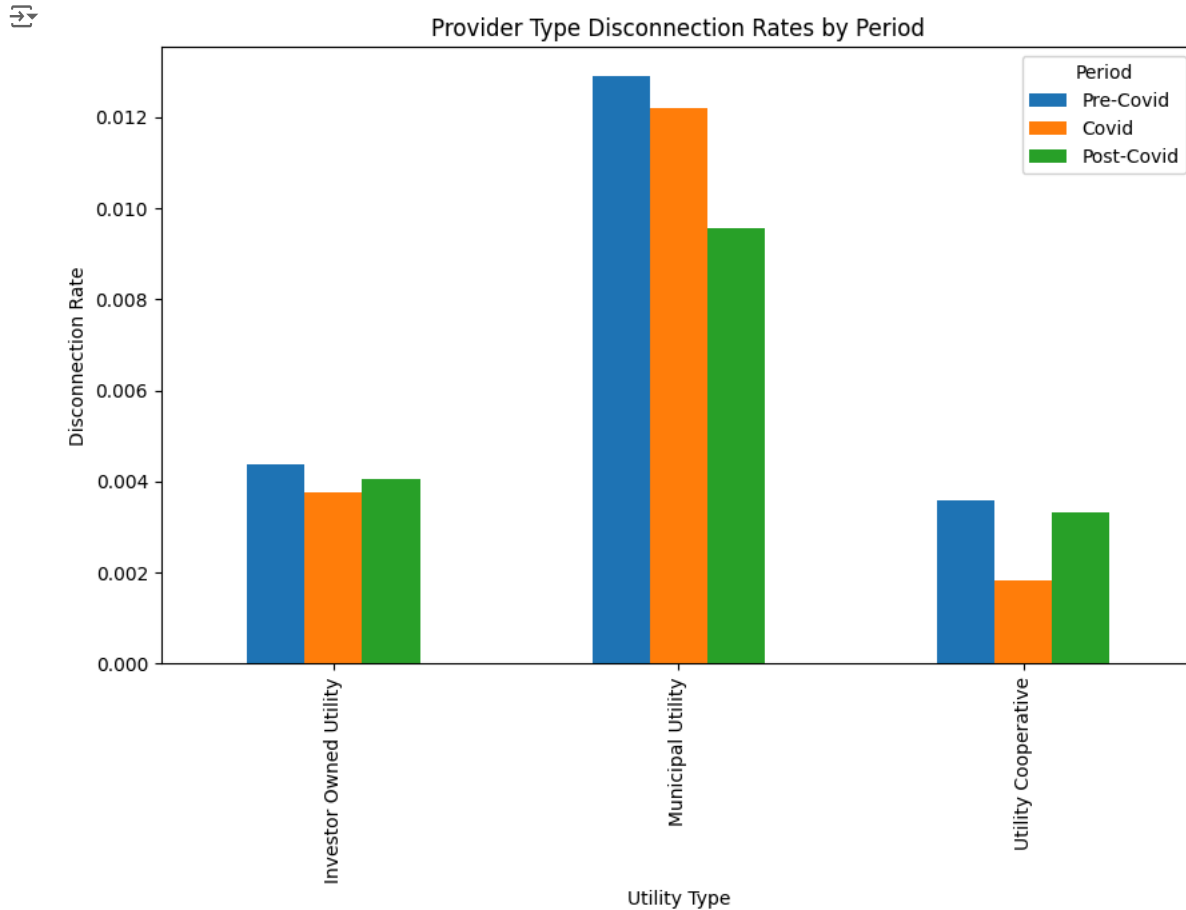
```
provider_summary = data.groupby(['Utility Type', 'Period'])['Disconnection Rate'].mean().unstack()
```

```
print("Provider Type Analysis:")
```

```
print(provider_summary)
```

```
↗ Provider Type Analysis:
Period      Pre-Covid      Covid      Post-Covid
Utility Type
Investor Owned Utility  0.004386  0.003763  0.004044
Municipal Utility      0.012896  0.012207  0.009557
Utility Cooperative     0.003596  0.001820  0.003317
<ipython-input-12-da159b1988c8>:2: FutureWarning: The default of observed=False is deprecated and will be changed to True
provider_summary = data.groupby(['Utility Type', 'Period'])['Disconnection Rate'].mean().unstack()
```

```
# Plot Provider Type Analysis
provider_summary.plot(kind='bar', figsize=(10, 6), title='Provider Type Disconnection Rates by Period')
plt.ylabel('Disconnection Rate')
plt.xlabel('Utility Type')
plt.legend(title='Period')
plt.show()
```



Start coding or [generate](#) with AI.