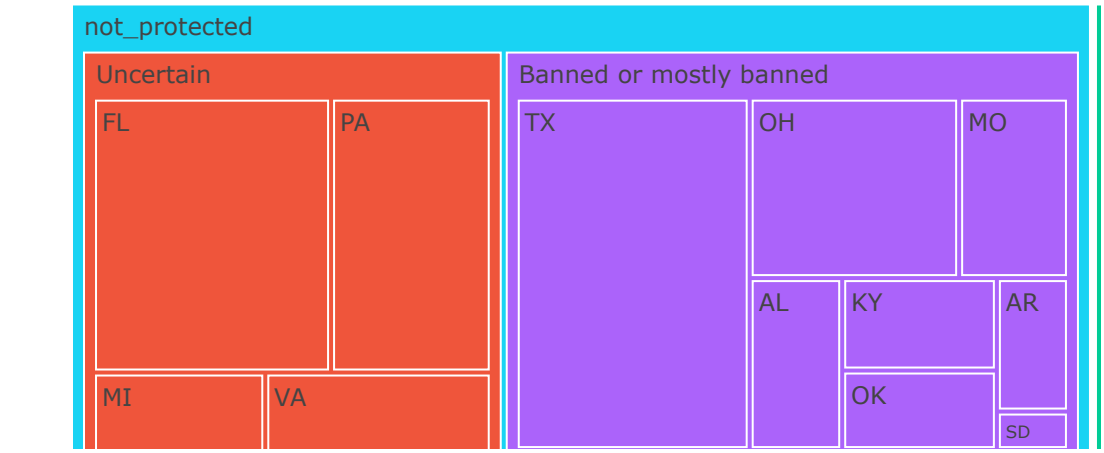
In [ ]:
```
import pandas as pd
import plotly.express as px
from IPython.display import display
from typing import List
px.set_mapbox_access_token(open(".mapbox_token").read())
pd.options.display.max_rows = 10

from helpers import  _load_synthetic_clinics, _load_at_risk_zip3, _k_closest_cl

draw_status_treemap()
```




In [ ]:
```
def build_distance_matrix(origin_zip3: List[str], clinics: pd.DataFrame, k:int
    #  this takes 2 min at 500 clinics / could consider optimizing
    """
        Iterates through each zip3 <> clinic permutation,
        and selects the min(distance) k clinic locations
    """

    dist_matrix = pd.concat(
```

```
        [_k_closest_clinics(zip3, clinics, k=k) for zip3 in origin_zip3]
    ).reset_index(drop=True)
    return dist_matrix

at_risk = _load_at_risk_zip3(adi_floor=20) # I could use alternate perspective
clinics = _load_synthetic_clinics(n=500) # see helpers.py for "Why Synthetic"

at_risk_distances = build_distance_matrix(at_risk['_zip3'], clinics)
at_risk_distances
```

```
572 zip3 Location{} at risk with ADI above 20
```

Out[ ]:

| | _state | _clinic_zip5 | _lat | _lng | _type | _origin_zip3 | _distance |
|---|---|---|---|---|---|---|---|
| **0** | IL | 62995 | 37.419275 | -88.879937 | synthetic_clinic | 301** | 293 |
| **1** | IL | 62967 | 37.584276 | -88.735808 | synthetic_clinic | 301** | 296 |
| **2** | IL | 62902 | 37.674381 | -89.112452 | synthetic_clinic | 301** | 315 |
| **3** | IL | 62833 | 38.340234 | -88.167646 | synthetic_clinic | 301** | 320 |
| **4** | IL | 62809 | 38.278964 | -88.337811 | synthetic_clinic | 301** | 321 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **5715** | CO | 80722 | 40.476059 | -103.200495 | synthetic_clinic | 828** | 347 |
| **5716** | CO | 80260 | 39.866988 | -105.001354 | synthetic_clinic | 828** | 348 |
| **5717** | CO | 80744 | 40.873626 | -102.391968 | synthetic_clinic | 828** | 349 |
| **5718** | CO | 80701 | 40.125042 | -103.817561 | synthetic_clinic | 828** | 353 |
| **5719** | CO | 80207 | 39.761385 | -104.916696 | synthetic_clinic | 828** | 356 |

5720 rows × 7 columns

In [ ]:

```
def _get_at_risk_stats(at_risk_distances: pd.DataFrame) -> pd.DataFrame:
    at_risk_stats = at_risk_distances.groupby(['_origin_zip3']).agg(
        k=("_distance","count"),
        distance_mean=("_distance","mean"), # mean distance to closest k clinic
        distance_min=("_distance","min"),
        distance_max=("_distance","max")
    ).reset_index()

    def _geocode_zip3(at_risk_stats):
        zip3_geo = _load_zip3_census()[['_zip3','_lat','_lng']]
        return at_risk_stats.rename(columns={"_origin_zip3":"_zip3"}).merge(_lo

    at_risk_stats = _geocode_zip3(at_risk_stats)
    return at_risk_stats
at_risk_stats = _get_at_risk_stats(at_risk_distances)
at_risk_stats # hmmm something messed up with NY and PA upstream
```

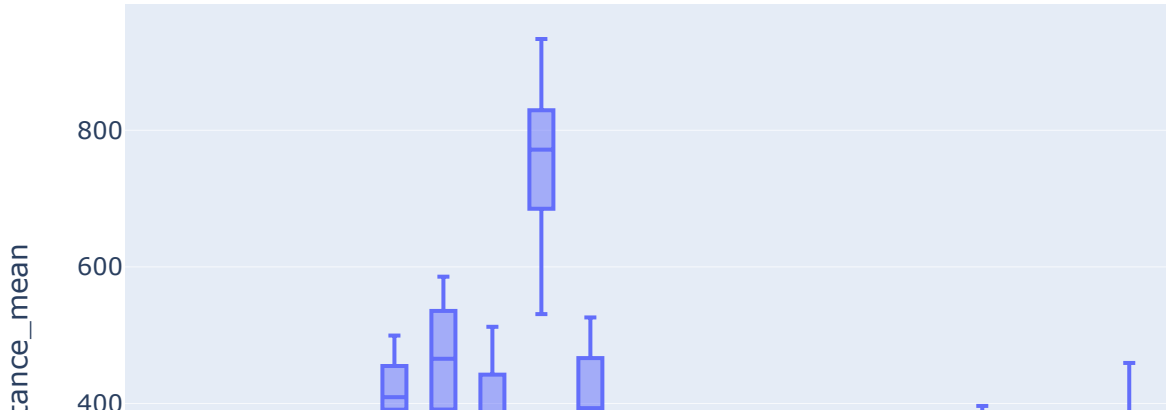
Out[ ]:

| | _zip3 | k | distance_mean | distance_min | distance_max | _state | _lat | _lng | _ |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 127** | 10 | 36.4 | 9 | 57 | NY | 41.695786 | -74.773225 | |
| **1** | 127** | 10 | 36.4 | 9 | 57 | PA | 41.485554 | -74.892326 | |
| **2** | 147** | 10 | 37.9 | 20 | 63 | NY | 42.211762 | -78.852733 | |
| **3** | 147** | 10 | 37.9 | 20 | 63 | PA | 42.046328 | -79.669830 | |
| **4** | 150** | 10 | 138.3 | 81 | 164 | PA | 40.457488 | -80.065227 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **638** | 979** | 10 | 218.4 | 127 | 257 | OR | 43.801827 | -117.543657 | |
| **639** | 990** | 10 | 125.9 | 67 | 180 | ID | 47.635828 | -117.094140 | |
| **640** | 990** | 10 | 125.9 | 67 | 180 | WA | 47.619646 | -117.441078 | |
| **641** | 991** | 10 | 125.5 | 68 | 196 | ID | 47.097949 | -117.081329 | |
| **642** | 991** | 10 | 125.5 | 68 | 196 | WA | 47.840974 | -118.012785 | |

643 rows × 10 columns

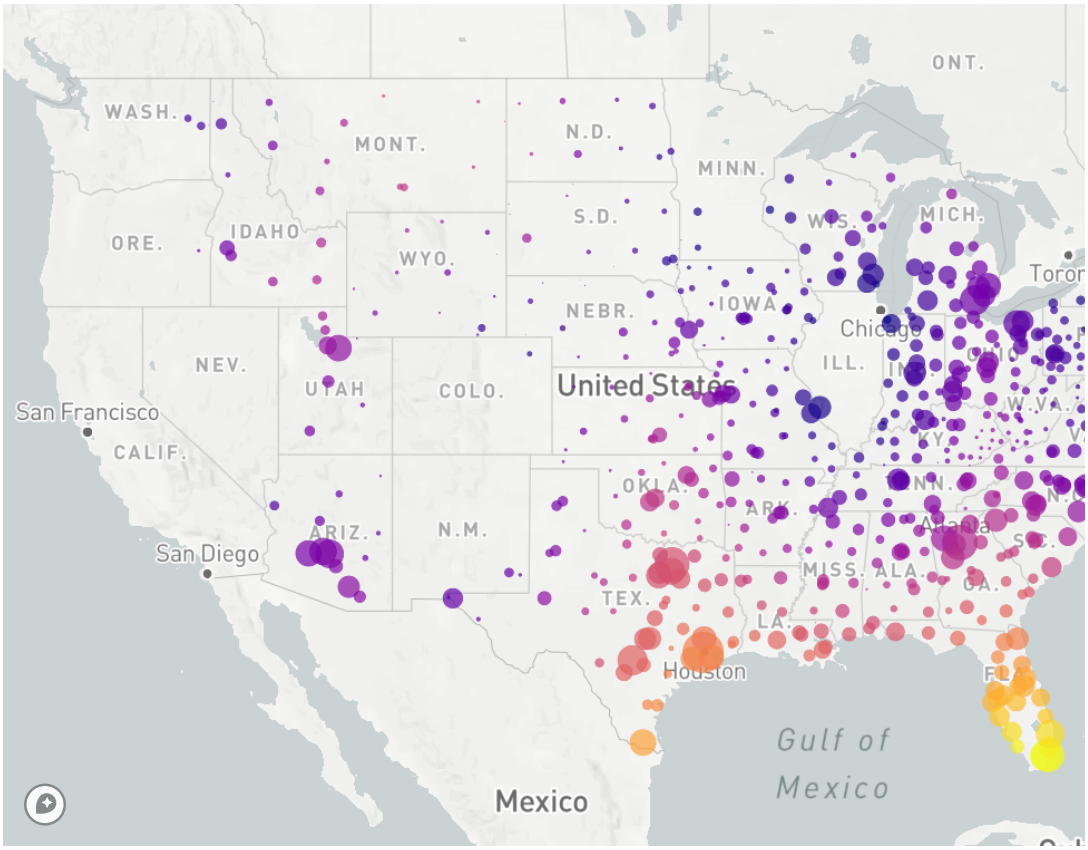
In [ ]:

```
px.box(
    at_risk_stats, x="_state", y="distance_mean",
    title="Range of Mean Distance to 10 Closest Clinics | by State"
).show(renderer="notebook")
```




In [ ]:
```
px.scatter_mapbox(
    at_risk_stats, lat="_lat", lon="_lng", size_max=15, hover_data=['_adi_mean'
    height=600, zoom=3, size='_census_total', color='distance_mean',
    title=f"Unprotected Origin Locations | n={len(at_risk_stats)} Zip3 | ADI Me
).show(renderer='notebook')
```

Unprotected Origin Locations | n=643 Zip3 | ADI Mean: 67.588




In [ ]:
```
px.histogram(
    at_risk_stats, x='distance_mean',
    title=f"On average, a protection seeker would need to travel {at_risk_stats
).show(renderer="notebook")
```


## NB: a good goal here is to find a way to strategically deploy resources to minimize harm

In [ ]:
```
def draw_at_risk_vs_clinic_locations_map() -> None:

    """
        Illustrative of Areas with Protections, vs those without
        Why syntethic clinics
    """
    at_risk = _load_at_risk_zip3()
    clinics = _load_synthetic_clinics(n=500)
    locations = pd.concat([clinics,at_risk]).fillna(at_risk['_census_total'].me
    px.scatter_mapbox(
        locations, lat="_lat", lon="_lng", size_max=15, mapbox_style="open-stre
        height=700, zoom=3, color='_state', hover_data=['_state','_zip3'],
        size='_census_total',
        title=f"At Risk Areas (3-Digit Zipcode) vs (Synthetic) Clinic Locations
    ).show(renderer='notebook')
    return locations

draw_at_risk_vs_clinic_locations_map()
```

```
498 zip3 Location{} at risk with ADI above 50
```

# At Risk Areas (3-Digit Zipcode) vs (Synthetic) Clinic Locations |

Out[ ]:

| | _state | _zip5 | _clinic_geo | _lat | _lng | _type | i |
|---|---|---|---|---|---|---|---|
| **0** | ME | 04276 | (44.5599203, -70.6252639) | 44.559920 | -70.625264 | synthetic_clinic | 275284.80 |
| **1** | WA | 99109 | (48.2924161, -117.6986398) | 48.292416 | -117.698640 | synthetic_clinic | 275284.80 |
| **2** | MN | 55129 | (44.8834738, -92.8927678) | 44.883474 | -92.892768 | synthetic_clinic | 275284.80 |
| **3** | CA | 93618 | (36.5221175, -119.3866282) | 36.522118 | -119.386628 | synthetic_clinic | 275284.80 |
| **4** | CO | 81653 | (40.8957476, -107.2422296) | 40.895748 | -107.242230 | synthetic_clinic | 275284.80 |
| **...** | ... | ... | ... | ... | ... | ... | |
| **493** | WV | 275284.803213 | 275284.803213 | 38.966814 | -78.995744 | at_risk | 955.00 |
| **494** | WY | 275284.803213 | 275284.803213 | 42.338740 | -104.575799 | at_risk | 961.00 |
| **495** | WY | 275284.803213 | 275284.803213 | 41.804052 | -106.980921 | at_risk | 962.00 |
| **496** | WY | 275284.803213 | 275284.803213 | 44.438020 | -108.408179 | at_risk | 963.00 |
| **497** | WY | 275284.803213 | 275284.803213 | 43.103802 | -108.847958 | at_risk | 964.00 |

998 rows × 10 columns

# _archive

below is archive and wip

# Simulation: Distance Traveled

Goal is to learn what the range of experiences will be as the experience of someone seeking care from an origin point in PA for example...

In [ ]:

```
def draw_closest_clinics_by_state(states: list):
    at_risk = _load_at_risk_zip3()
    _at_risk = at_risk[at_risk['_state'].isin(states)]
    locations = pd.concat([_at_risk, _load_synthetic_clinics()])

    px.scatter_mapbox(
        locations, lat="_lat", lon="_lng", size_max=15,
        height=600, zoom=3, color='_type',
        title=f"At Risk Areas vs (Simulated) 10 Closest Clinic Locations"
    ).show(renderer='notebook')

draw_closest_clinics_by_state(["PA",'LA'])
```

```
498 zip3 Location{} at risk with ADI above 50
```

# At Risk Areas vs (Simulated) 10 Closest Clinic Locations