

```

zips_texas_borderstates_centroids['ZCTA5'] =
    zips_texas_borderstates_centroids['ZCTA5'].astype(str).str.replace('.0',
    '', regex=False)

zips_withhospital_centroids =
    zips_texas_borderstates_centroids.merge(hospitals_per_zip, left_on =
    'ZCTA5', right_on = 'ZIP_CD', how = 'left').fillna(0)
zips_withhospital_centroids =
    zips_withhospital_centroids[zips_withhospital_centroids['hospital_count']
    >= 1]

print(len(zips_withhospital_centroids))
print(zips_withhospital_centroids.head())

```

448

	ZCTA5	centroid	ZIP_CD	hospital_count
8	70043	POINT (-89.96276 29.94804)	70043	1.0
27	70127	POINT (-89.97675 30.02501)	70127	1.0
31	70301	POINT (-90.74089 29.8141)	70301	1.0
38	70360	POINT (-90.81028 29.58819)	70360	2.0
45	70403	POINT (-90.48388 30.48002)	70403	2.0

/opt/anaconda3/lib/python3.11/site-packages/geopandas/geodataframe.py:1819:
SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:

https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

I'm using left merge.

Merge Variables: The variables used in the merge are ZCTA5 (from zips_texas_borderstates_centroids) and ZIP_CD (from hospitals_per_zip).

4. a.

```

subset_zips = zips_texas_centroids.sample(n=10, random_state=10)

start_time = time.time()

def cal_nearest_distance(row, df):
    nearest_geom = nearest_points(row['centroid'], df.unary_union)[1]
    return row['centroid'].distance(nearest_geom)

subset_zips['nearest_distance'] = subset_zips.apply(

```

```

    cal_nearest_distance,
    df=zips_withhospital_centroids,
    axis=1
)

end_time = time.time()

time_calculations = end_time - start_time
print(f"Time taken (10 zipcodes): {time_calculations} seconds")
time_all = time_calculations * (len(zips_texas_centroids)/10)
print(f"Time taken (all): {time_all} seconds")

Time taken (10 zipcodes): 0.009268999099731445 seconds
Time taken (all): 1.7935513257980347 seconds

/var/folders/29/92n8lbb16qsc1h27rdtp7j8c0000gn/T/ipykernel_11697/3923260464.py:6:
DeprecationWarning:

The 'unary_union' attribute is deprecated, use the 'union_all()' method instead.

b.
```

```

start_time_b = time.time()

zips_texas_centroids['nearest_distance'] = zips_texas_centroids.apply(
    cal_nearest_distance,
    df=zips_withhospital_centroids,
    axis=1
)

end_time_b = time.time()

time_calculations_b = end_time_b - start_time_b
print(f"Total time: {time_calculations_b} seconds")

/var/folders/29/92n8lbb16qsc1h27rdtp7j8c0000gn/T/ipykernel_11697/3923260464.py:6:
DeprecationWarning:

The 'unary_union' attribute is deprecated, use the 'union_all()' method instead.

Total time: 1.0758678913116455 seconds

/opt/anaconda3/lib/python3.11/site-packages/geopandas/geodataframe.py:1819:
SettingWithCopyWarning:
```

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation:

https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

The difference is significant, it actually runs much faster than I expected.

c.

UNIT["Degree",0.017453292519943295] indicates that the unit is degree (latitude and longitude coordinates), and the number 0.017453292519943295 is the factor that converts degree to radian. Transform the latitude and longitude into UTM using pyproj coordinates to accurately calculate the distance (in meters).

1m 0.000621371miles

```
zips_texas_centroids = zips_texas_centroids.to_crs(epsg=32614)
zips_withhospital_centroids = zips_withhospital_centroids.to_crs(epsg=32614)

zips_texas_centroids['nearest_distance_meters'] =
    ↪ zips_texas_centroids.apply(
        cal_nearest_distance,
        df=zips_withhospital_centroids,
        axis=1
    )

zips_texas_centroids['nearest_distance_miles'] =
    ↪ zips_texas_centroids['nearest_distance_meters'] * 0.000621371
print(zips_texas_centroids[['ZCTA5', 'nearest_distance_miles']].head())
```

/var/folders/29/92n8lbb16qsc1h27rdtp7j8c0000gn/T/ipykernel_11697/3923260464.py:6:
DeprecationWarning:

The 'unary_union' attribute is deprecated, use the 'union_all()' method instead.

	ZCTA5	nearest_distance_miles
9207	78624	0.000000
9208	78626	11.425191
9209	78628	7.579457
9210	78631	22.673291
9211	78632	15.131525

5. a. I started with a distance calculation using UTM (EPSG:32614), which gives the units of meters. The meters were then converted to miles in order to report the average distance from each ZIP code to the nearest hospital.

b.

```

average_distance_meters =
    ↵ zips_texas_centroids['nearest_distance_meters'].mean()
average_distance_miles = average_distance_meters * 0.000621371
print(f"Avg distance to the nearest hospital: {average_distance_miles:.2f}
    ↵ miles")

```

Avg distance to the nearest hospital: 13.43 miles

The average distance of 13.43 miles to the nearest hospital may seem like a reasonable result since Texas is a sparsely populated state.

In Texas, the distance between hospitals varies by region. According to the 2022 American Hospital Association report, the average Texas resident would have to travel approximately 8.5 miles to reach the nearest hospital. There is also data showing that the average rural Texas resident is about twice as far from a hospital as an urban resident, at about 12.5 miles, demonstrating the challenges of healthcare access in rural areas.

Reference:

<https://www.pewresearch.org/short-reads/2018/12/12/how-far-americans-live-from-the-closest-hospital-differs-by-community-type/>
<https://www.aha.org/news/headline/2018-12-14-study-rural-residents-travel-about-twice-far-hospital-average>

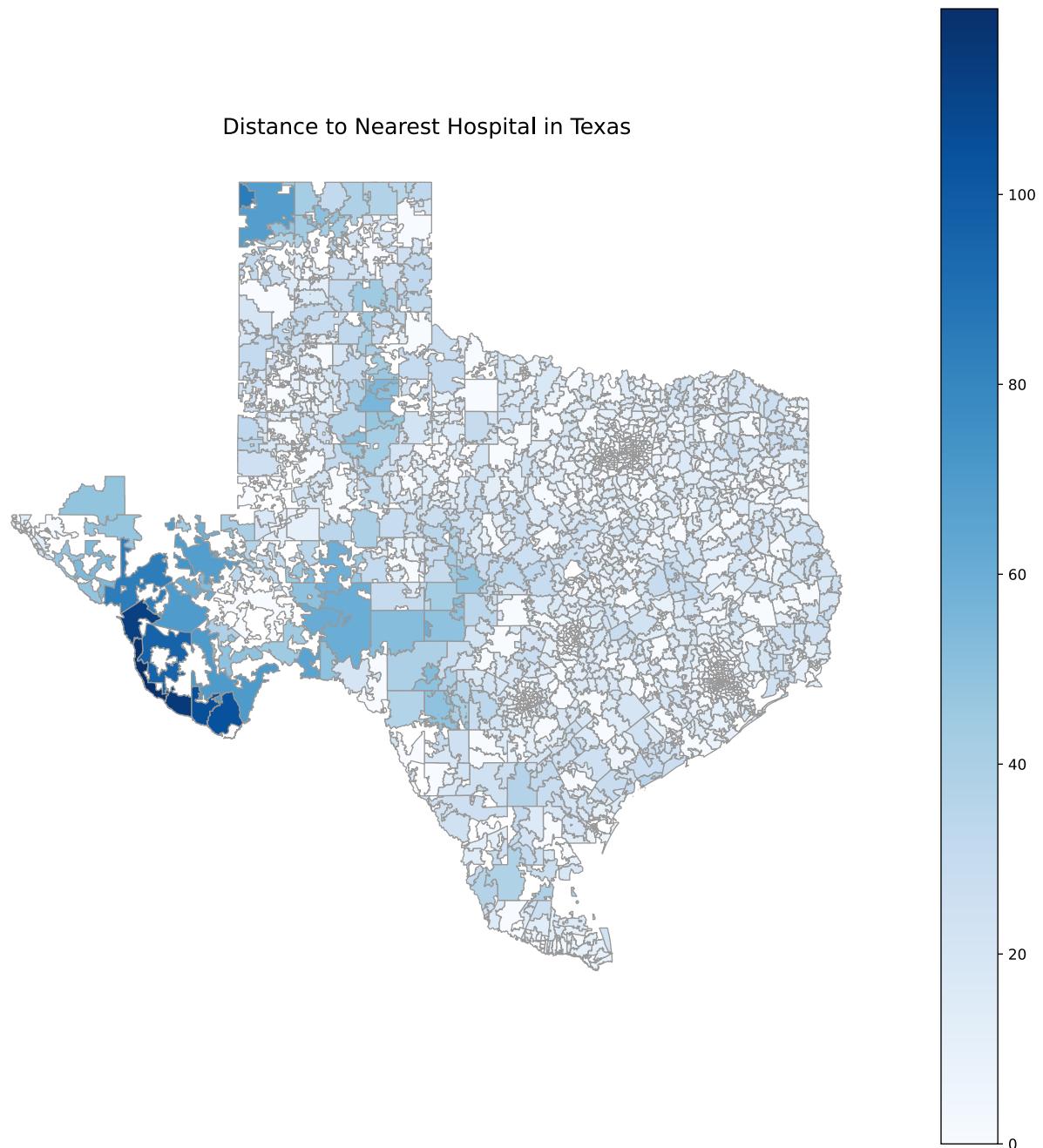
c.

```

zips_texas = zips_texas.merge(
    zips_texas_centroids[['ZCTA5', 'nearest_distance_miles']],
    on='ZCTA5',
    how='left'
)

fig, ax = plt.subplots(figsize=(14, 14))
zips_texas.plot(
    column='nearest_distance_miles',
    cmap='Blues',
    linewidth=0.8,
    ax=ax,
    edgecolor='0.6',
    legend=True
)
ax.set_title('Distance to Nearest Hospital in Texas', fontsize=15)
ax.set_axis_off()
plt.show()

```



Effects of closures on access in Texas (15 pts)

1.

```
texas_closures = valid_sorted_closed[
    valid_sorted_closed['ZIP_CD'].astype(str).str.startswith(texas_prefixes)
]
```

```

texas_closures_by_zip =
    ↵   texas_closures.groupby('ZIP_CD').size().reset_index(name='closure_count')

texas_closures_by_zip['ZIP_CD'] =
    ↵   texas_closures_by_zip['ZIP_CD'].astype(str).str.split('.').str[0]

print(texas_closures_by_zip)

ZIP_CD  closure_count
0      75051          1
1      75087          1
2      75140          1
3      75235          1
4      75390          1
5      76520          1
6      76531          1
7      76645          1
8      77065          1
9      78336          1
10     78613          1
11     79520          1
12     79529          1
13     79902          1

```

2.

```

zips_texas = zips_texas.merge(
    texas_closures_by_zip, left_on='ZCTA5', right_on='ZIP_CD', how='left'
)
zips_texas['closure_count'] = zips_texas['closure_count'].fillna(0)
zips_texas = zips_texas.drop(columns=['ZIP_CD_x', 'ZIP_CD_y'])
print(zips_texas.head())

fig, ax = plt.subplots(figsize=(14, 14))
zips_texas.plot(
    column='closure_count',
    cmap='Blues',
    linewidth=0.8,
    ax=ax,
    edgecolor='0.6',
    legend=True,
    legend_kwds={'label': "Number of Closures", 'orientation': "horizontal"}
)
ax.set_title('Texas zip codes directly affected by a closure in 2016-2019',
    ↵   fontsize=15
)
ax.set_axis_off()

```