```
# Loading libraries
import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings("ignore")

# For visvalization
import matplotlib.pyplot as plt
# import seaborn as sns
import seaborn as sns

# Plotting using plotly
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots

# For creating widgets
from ipywidgets import widgets, interact
from IPython.display import display

import os
os.chdir(r'C:\Users\manog\OneDrive\Desktop\Data Science')

# Reading the files
df_cases = pd.read_csv('covid_confirmed_usafacts.csv')
df_deaths = pd.read_csv('covid_deaths_usafacts.csv')

# County wise population
df_population = pd.read_csv('covid_county_population_usafacts.csv')
df_population.head()
```

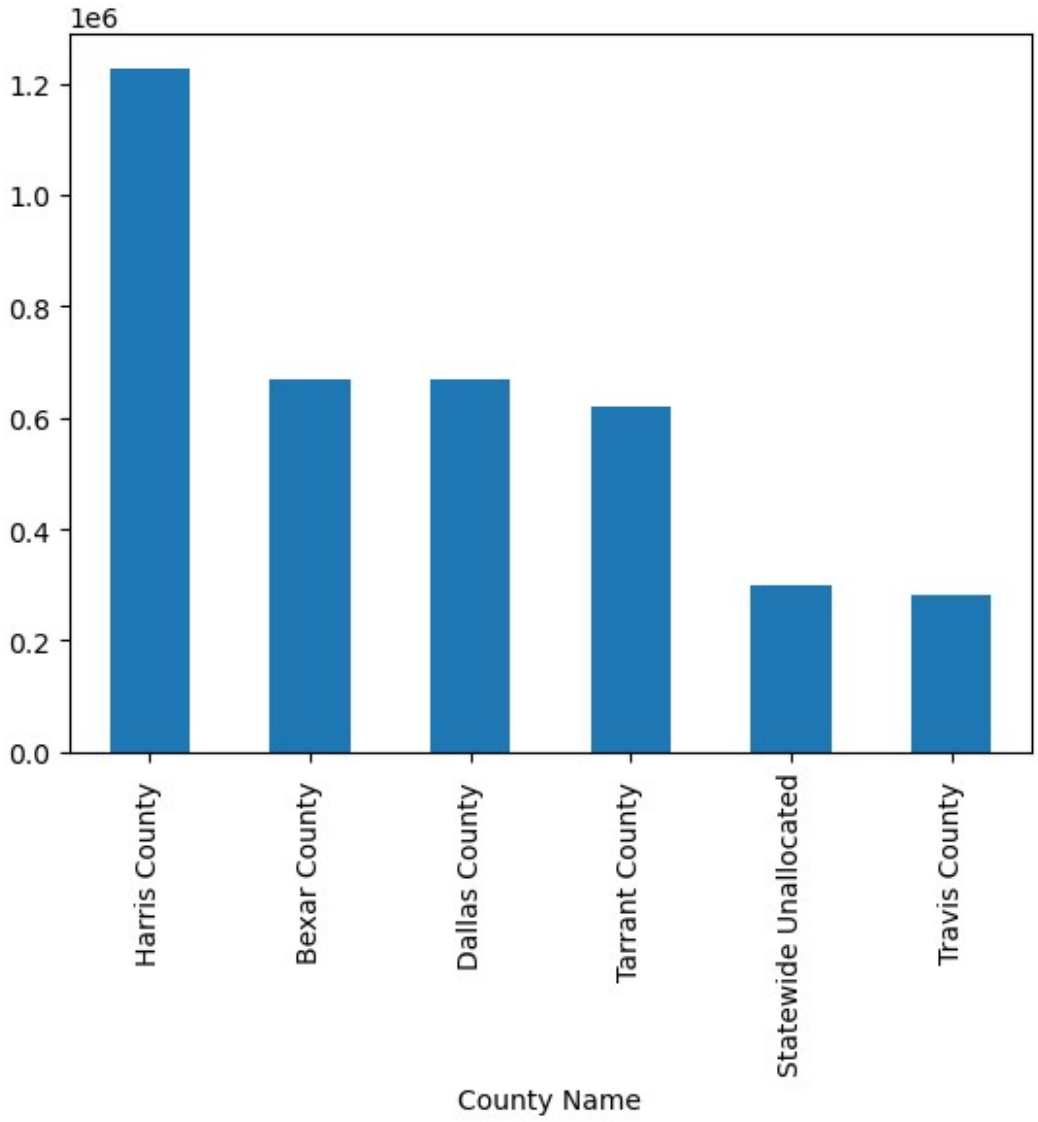
	countyFIPS	County Name	State	population
0	0	Statewide Unallocated	AL	0
1	1001	Autauga County	AL	55869
2	1003	Baldwin County	AL	223234
3	1005	Barbour County	AL	24686
4	1007	Bibb County	AL	22394

Data Prepration

```
df_cases_modified = df_cases.melt(id_vars=['countyFIPS', 'County
Name', 'State', 'StateFIPS'],
                                  var_name="Date",
                                  value_name="Cases")

df_deaths_modified = df_deaths.melt(id_vars=['countyFIPS', 'County
Name', 'State', 'StateFIPS'],
                                    var_name="Date",
                                    value_name="Deaths")
```

```
# Modifying the data for the 2nd half of 2022
df_cases_modified['Date'] = pd.to_datetime(df_cases_modified['Date'])
df_cases_modified =
df_cases_modified[(df_cases_modified['Date']>"6/1/2022") &
(df_cases_modified['Date']<"12/31/2022")]

df_deaths_modified['Date'] =
pd.to_datetime(df_deaths_modified['Date'])
df_deaths_modified =
df_deaths_modified[(df_deaths_modified['Date']>"6/1/2022") &
(df_deaths_modified['Date']<"12/31/2022")]

# Saving a copy
df_cases_modified_copy = df_cases_modified.copy()

# Selecting a state - 'TX'
df_cases_modified =
df_cases_modified[df_cases_modified['State']=='TX']
df_deaths_modified =
df_deaths_modified[df_deaths_modified['State']=='TX']

# Selecting top 5 counties based on number of cases
df_cases_modified.groupby(['County Name'])
['Cases'].max().sort_values(ascending=False)[:6].plot(kind='bar')
```

```
<AxesSubplot:xlabel='County Name'>
```




```
# Filtering the data based on the top 5 counties
top_five_counties = ['Harris County ', 'Bexar County ', 'Dallas County
',
                     'Tarrant County ', 'Travis County ']

df_deaths_top5 = df_deaths_modified[df_deaths_modified['County
Name'].isin(top_five_counties)]
df_cases_top5 = df_cases_modified[df_cases_modified['County
Name'].isin(top_five_counties)]

# Merging the cases and deaths data along with population data
cols_to_use =
df_cases_top5.columns.difference(df_deaths_top5.columns).to_list() +
['Date', 'County Name']
df_merged = pd.merge(df_deaths_top5, df_cases_top5[cols_to_use],
on=['County Name', 'Date'])
```

```
# Removing leading and training spaces
df_merged['County Name'] = df_merged['County Name'].apply(lambda x:
x.strip())

# Merging with population
df_merged = pd.merge(df_merged, df_population, on=['State', 'County
Name'])

df_merged.head()
```

```
   countyFIPS_x   County Name State  StateFIPS       Date  Deaths
Cases  \
0         48029  Bexar County    TX         48 2022-06-02    6146
573582
1         48029  Bexar County    TX         48 2022-06-03    6146
574304
2         48029  Bexar County    TX         48 2022-06-04    6149
574974
3         48029  Bexar County    TX         48 2022-06-05    6149
575487
4         48029  Bexar County    TX         48 2022-06-06    6153
575980

   countyFIPS_y  population
0         48029     2003554
1         48029     2003554
2         48029     2003554
3         48029     2003554
4         48029     2003554
```

```
# Removing unecessary or duplicate information columns and renaming
others
columns_to_remove = ['countyFIPS_y', 'State', 'countyFIPS_x',
'StateFIPS']
df_merged = df_merged.drop(columns=columns_to_remove)
df_merged = df_merged.rename(columns={'countyFIPS_x':'countyFIPS'})

# Extracting Day, Month and Year from the Date
df_merged['Date'] = pd.to_datetime(df_merged['Date'])
df_merged_copy = df_merged.copy()
df_merged['day'] = df_merged['Date'].dt.day
df_merged['month'] = df_merged['Date'].dt.month
# Droping year as it is same for all the present data
# df_merged['year'] = df_merged['Date'].dt.year
```

The features `day` and `month` are cyclic in nature. If I do not do any preprocessing on them and directly feed them to model, the model can give more or less importance based on the values. Eg. days will have values from 1 to 31, so model thinks that value 31 is more than 1,

but actually they are just days so our model can go wrong. Thus, I will be performing a transformation on these features to make them cyclic.

```
# months and days in a cyclic continuous feature.
def encode(data, col, max_val):
    data[col + '_sin'] = np.sin(2 * np.pi * data[col]/max_val)
    data[col + '_cos'] = np.cos(2 * np.pi * data[col]/max_val)
    return data

df_merged = encode(df_merged, 'month', 6)
df_merged = encode(df_merged, 'day', 31)

# Normalizing the year and population column
# def normalize_data(df, col):
#     normalized_df = (df[col]-df[col].min())/(df[col].max()-
df[col].min())
#     return normalized_df

# df_merged['year'] = normalize_data(df_merged, 'year')

# Removing unecessary or duplicate information columns
df_merged = df_merged.drop(columns=['day', 'month', 'Date'])
```

Linear and Non-Linear (polynomial) regression models

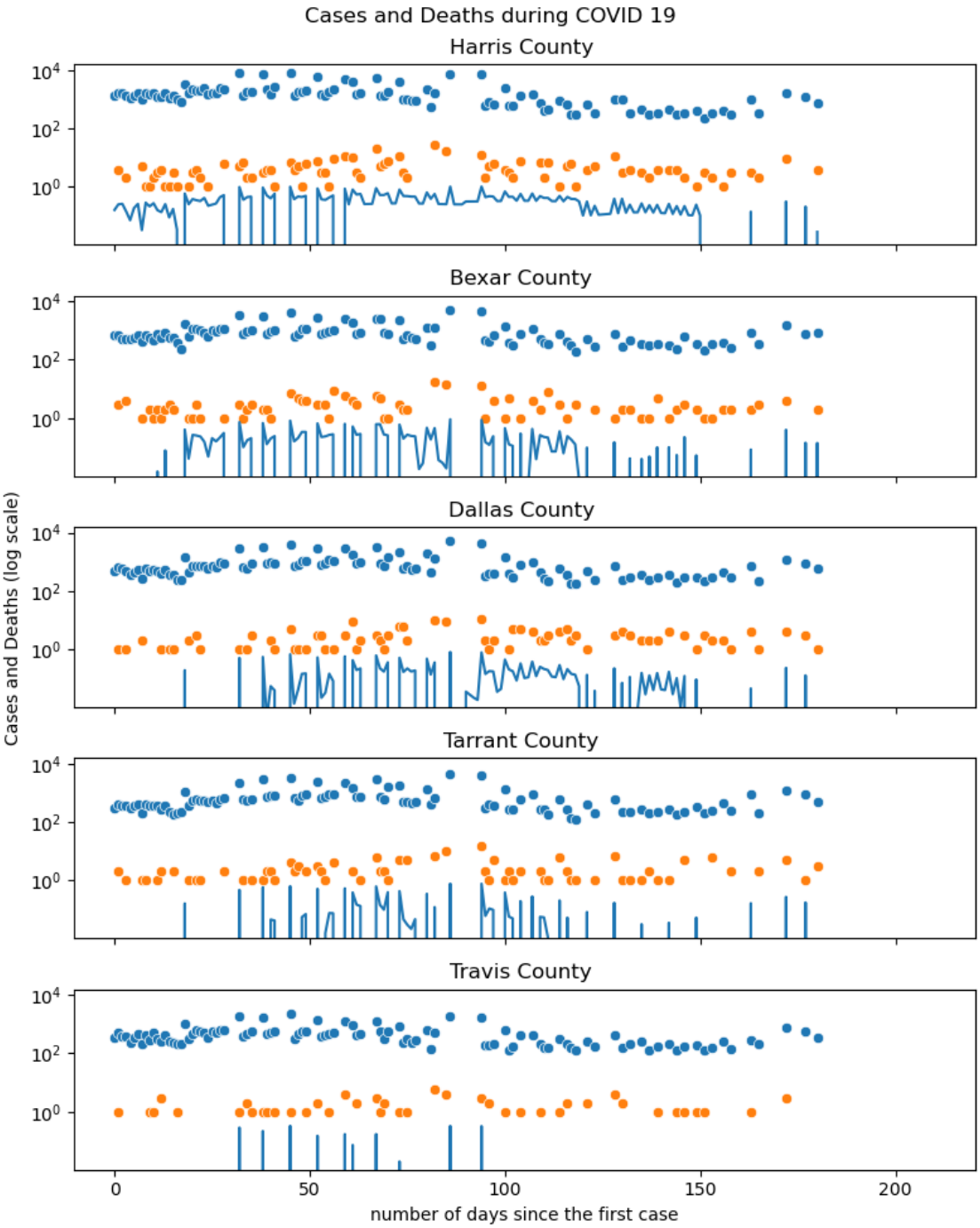
```
# 5 models for top 5 counties
# Statewide Unallocated County which is not a countie so not
considering it
df_harris = df_merged[df_merged['County Name']=='Harris County']
df_bexar = df_merged[df_merged['County Name']=='Bexar County']
df_dallas = df_merged[df_merged['County Name']=='Dallas County']
df_tarrant = df_merged[df_merged['County Name']=='Tarrant County']
df_travis = df_merged[df_merged['County Name']=='Travis County']

# Linear Model
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error


def create_linear_model(data, name=None):
    lr = LinearRegression() #RandomForestRegressor(max_depth=3,
min_samples_leaf=50)
    data_fn = data.copy()

    # Cumulative to normal values
    data_fn['Cases'] = data_fn['Cases'].diff().fillna(0)
    data_fn['Deaths'] = data_fn['Deaths'].diff().fillna(0)

    # From start of infection
    start_of_infection = data_fn[data_fn['Cases']>0].index[0]
```

```
    data_fn = data_fn.loc[start_of_infection:, :]
    data_fn.reset_index(drop=True, inplace=True)

    X = data_fn.drop(columns=['County Name', 'population', 'Deaths'])
    y = data_fn['Deaths']

    # Fitting the model
    lr.fit(X, y)

    # Prediction
    y_predict = lr.predict(X)
    data_fn['prediction'] = y_predict

    # RMSE
    rmse = mean_squared_error(y, y_predict)**0.5
    print(f"The RMSE for {name} county is {rmse}")

    return data_fn

# RMSE for top 5 counties
df1 = create_linear_model(df_harris, 'Harris County')
df2 = create_linear_model(df_bexar, 'Bexar County')
df3 = create_linear_model(df_dallas, 'Dallas County')
df4 = create_linear_model(df_tarrant, 'Tarrant County')
df5 = create_linear_model(df_travis, 'Travis County')
```

```
The RMSE for Harris County county is 2.9558204441911546
The RMSE for Bexar County county is 1.9807025618971916
The RMSE for Dallas County county is 1.522824329172117
The RMSE for Tarrant County county is 1.590593273543241
The RMSE for Travis County county is 0.7607077342890752
```

```
# Plotting data
fig, axis = plt.subplots(5, 1, figsize=(8, 10), sharey=True,
sharex=True)
df_list = [df1, df2, df3, df4, df5]
i = 0
top_five_counties = ['Harris County ', 'Bexar County ', 'Dallas County
',
                     'Tarrant County ', 'Travis County ']
for data_frame, ax, name in zip(df_list, axis.ravel(),
top_five_counties):
  ax.set(yscale="log")
  sns.scatterplot(x=data_frame.index, y=data_frame['Cases'], ax=ax,
legend='brief')
  sns.scatterplot(x=data_frame.index, y=data_frame['Deaths'], ax=ax)
  sns.lineplot(x=data_frame.index, y=data_frame['prediction'], ax=ax)
  ax.set_title(name)
  ax.set_ylabel("")
  if i == 2:
    ax.set_ylabel("Cases and Deaths (log scale)")
```

```
  else: ax.set_ylabel("")
  ax.set_xlabel("number of days since the first case")
  i+=1

fig.suptitle("Cases and Deaths during COVID 19")
plt.tight_layout()
plt.show()
```



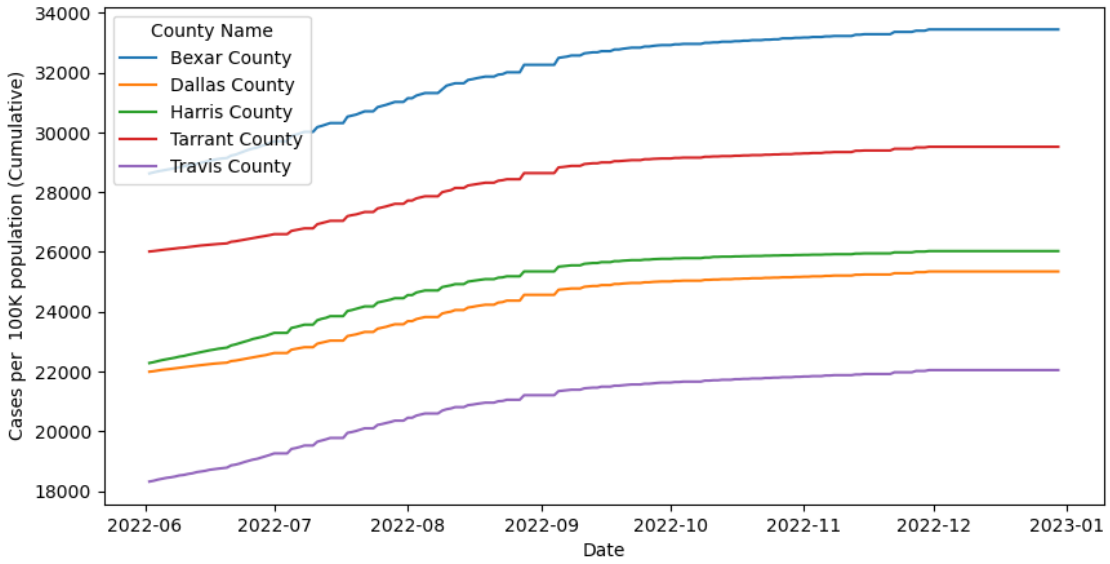
For all the top 5 counties in TX, the trend of cases and deaths is similar. It has got peaks after around 90 days of the first case from 1st June 2022.

Identify which counties are most at risk.

```
df_merged_copy['per_capita_cases'] =
df_merged_copy['Cases']*100000/df_merged_copy['population']
df_merged_copy['per_capita_deaths'] =
df_merged_copy['Deaths']*100000/df_merged_copy['population']

df_merged_copy.head()
```

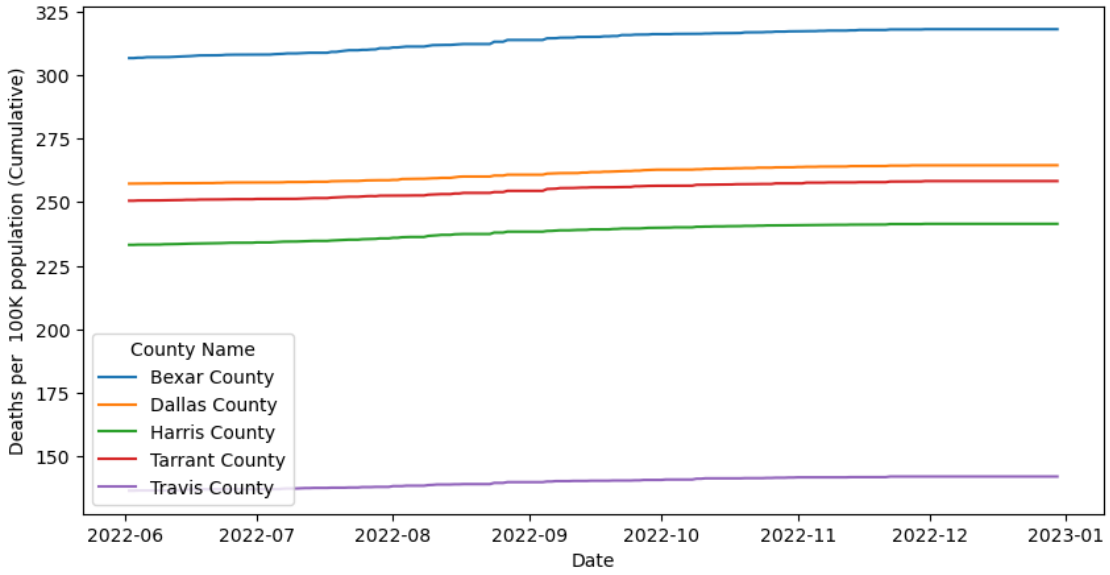
```
    County Name       Date  Deaths   Cases  population
per_capita_cases  \
0  Bexar County 2022-06-02    6146  573582     2003554
28628.227639
1  Bexar County 2022-06-03    6146  574304     2003554
28664.263604
2  Bexar County 2022-06-04    6149  574974     2003554
28697.704180
3  Bexar County 2022-06-05    6149  575487     2003554
28723.308680
4  Bexar County 2022-06-06    6153  575980     2003554
28747.914955

   per_capita_deaths
0         306.754897
1         306.754897
2         306.904630
3         306.904630
4         307.104276
```

```
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
sns.lineplot(data=df_merged_copy, x='Date', y='per_capita_cases',
hue='County Name', ax=ax)
ax.set_ylabel("Cases per  100K population (Cumulative)");
```



For Bexar county, the cases are rising faster than others. Specially in July-2022 to Sept-2022, the rise in COVID cases were highest (which is true for all other counties as well).

```
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
sns.lineplot(data=df_merged_copy, x='Date', y='per_capita_deaths',
hue='County Name', ax=ax)
ax.set_ylabel("Deaths per  100K population (Cumulative)");
```



The deaths per 100K population is almost constant except Bexar county which is a concern as deaths other counties are very less compared to Bexar

Hypothesis Testing

```
# Loading employment data
employment_data = pd.read_csv('EmployementEnrichment.csv')
employment_data.head()
```

```
  Area\nCode  St  Cnty  Own  NAICS  Year  Qtr Area Type St Name
Area  \
0      US000  US   0.0    0     10  2022    1    Nation     NaN  U.S.
TOTAL
1      US000  US   0.0    1     10  2022    1    Nation     NaN  U.S.
TOTAL
2      US000  US   0.0    2     10  2022    1    Nation     NaN  U.S.
TOTAL
3      US000  US   0.0    3     10  2022    1    Nation     NaN  U.S.
TOTAL
4      US000  US   0.0    5     10  2022    1    Nation     NaN  U.S.
TOTAL

   ...                   Industry Status Code Establishment Count  \
0  ...  10 Total, all industries         NaN          11,294,919
1  ...  10 Total, all industries         NaN              60,882
2  ...  10 Total, all industries         NaN              70,526
3  ...  10 Total, all industries         NaN             171,395
4  ...  10 Total, all industries         NaN          10,992,116

  January Employment February Employment March Employment  \
0        145,582,488         147,172,213      147,781,400
1          2,860,360           2,858,451        2,853,067
2          4,456,377           4,529,802        4,534,064
3         13,954,426          14,065,705       14,114,268
4        124,311,325         125,718,255      126,280,001

  Total Quarterly Wages Average Weekly Wage  \
0     2,622,652,420,011               1,374
1        60,805,753,244               1,637
2        78,903,410,196               1,347
3       203,425,519,443               1,114
4     2,279,517,737,128               1,398

  Employment Location Quotient Relative to U.S.  \
0                                           1.0
1                                           1.0
2                                           1.0
3                                           1.0
4                                           1.0

   Total Wage Location Quotient Relative to U.S.
0                                            1.0
1                                            1.0
2                                            1.0
```

```
3                                                   1.0
4                                                   1.0

[5 rows x 21 columns]
```

```
# Hypothesis 1: Higher Employment leads to lower COVID cases as the
spread is lower.
condition = ((df_cases_modified['Date']>="2022-6-1") &
((df_cases_modified['Date']<="2022-12-31")))
cases = df_cases_modified[condition]['Cases'].values

# Merging Jan, Feb and March employemnt
employment = employment_data[['January Employment', 'February
Employment', 'March Employment']].apply(lambda x : x.str.replace(',',
'').astype(float)).values
# Flattening data
employment = np.reshape(employment, (-1, 1))

# Print the variance of both data groups
print(np.var(employment)/np.var(cases))
```

```
76.42421821967889
```

As the variance ratio is > 4:1, the Welch's t-test will be conducted by not taking into consideration the equal population variances.

```
import scipy.stats as stats

# Performing the two sample t-test with unequal variances
stats.ttest_ind(a=employment, b=cases, equal_var=False)
```

```
Ttest_indResult(statistic=array([1.51309677]),
pvalue=array([0.13025662]))
```

p-value is > 0.05 (significance value), so the null hypothesis can't be rejected. Thus, higher Employment does lead to lower COVID cases.

```
# Hypothesis 2: Higher weekly wages might lead to lower COVID cases as
people can spend
# more on immunity build up or take better precautionary measure in
controlling COVID.

weekly_wages = employment_data['Average Weekly Wage'].str.replace(',',
'').astype(float).values
# Print the variance of both data groups
print(np.var(cases)/np.var(weekly_wages))
```

```
42181.86466729231
```

```
# Performing the two sample t-test with unequal variances
stats.ttest_ind(a=cases, b=weekly_wages, equal_var=False)
```

```
Ttest_indResult(statistic=63.60272393583431, pvalue=0.0)
```

p-value is < 0.05 (significance value), so the null hypothesis can be rejected. Thus higher weekly wages might not lead to lower COVID cases.

```
# Confidance Interval
from statistics import NormalDist

def confidence_interval(data, confidence=0.95):
  dist = NormalDist.from_samples(data)
  z = NormalDist().inv_cdf((1 + confidence) / 2.)
  h = dist.stdev * z / ((len(data) - 1) ** .5)
  return dist.mean - h, dist.mean + h

def plot_results(data, name=None):
  fig = go.Figure()
  x = data.index.values
  CI = confidence_interval(data['prediction'], 0.95)

  fig.add_trace(go.Scatter(x=x, y=data['Deaths'],
                      mode='markers',
                      name='Deaths'))

  fig.add_traces(go.Scatter(x=x, y = data['prediction'],
                            mode='markers+lines',
                             name='prediction'))

  fig.add_traces([go.Scatter(x=x, y = data['prediction']+CI[1],
                            mode = 'lines', line_color =
'rgba(0,0,0,0)',
                            showlegend = False),
                  go.Scatter(x=x, y = data['prediction']-CI[0],
                            mode = 'lines', line_color =
'rgba(0,0,0,0)',
                            name = '95% confidence interval',
                            fill='tonexty', fillcolor = 'rgba(255, 0,
0, 0.2)')])

  fig.update_yaxes(title_text="No. of deaths")
  fig.update_xaxes(title_text="number of days since the first case")
  fig.update_layout(
      title=dict(text=f"Analysis for {name}")
  )

  fig.show()

plot_results(df1, 'Harris County')
```


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":
[{"mode":"markers","name":"Deaths","type":"scatter","x":
[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,2
```

```
6,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49
,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,
73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,9
6,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114
,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,1
32,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149
,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,1
67,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184
,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,2
02,203,204,205,206,207,208,209,210],"y":
[0,4,0,2,0,0,0,5,1,1,2,3,4,1,1,3,1,0,0,1,3,4,2,0,1,0,0,0,6,0,0,0,5,7,2
,2,0,0,3,4,4,1,0,0,0,7,4,5,1,6,0,0,8,3,3,1,9,0,0,11,0,10,3,2,0,0,0,20,
5,6,8,0,0,11,3,2,0,0,0,0,0,0,27,0,0,17,0,0,0,0,0,0,0,0,12,2,5,6,0,0,4,
3,2,0,8,0,0,0,0,7,2,7,0,0,1,0,5,6,1,0,0,4,0,5,0,0,0,0,11,0,3,0,4,0,0,3
,0,2,0,4,0,0,4,0,4,0,2,0,0,1,0,3,0,2,0,0,1,0,3,0,0,0,0,3,0,2,0,0,0,0,0
,0,9,0,0,0,0,0,0,0,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0]},
{"mode":"markers+lines","name":"prediction","type":"scatter","x":
[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,2
6,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49
,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,
73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,9
6,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114
,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,1
32,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149
,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,1
67,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184
,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,2
02,203,204,205,206,207,208,209,210],"y":
[1.4545493764955362,1.7940552827711205,1.8344450562560226,1.3918805668
566774,1.1818547378034012,1.5447040101919873,1.8307426034300867,1.0794
007332181639,1.9379596294577155,1.6425480995670954,1.8906671822051275,
1.4442764027323585,1.3700581218907695,1.8078584315660295,1.23306385999
87447,1.4949555633451552,1.0828430140945282,0.9158054814484033,3.92396
8289698833,1.7998721735093124,2.415727144214398,2.232908974952678,2.13
39811672371254,2.630966583998275,1.6834219254969573,1.761523703376743,
1.8606324445689473,2.7667731443238246,3.3408123854407963,0.69302771758
65462,0.6970344070361552,0.7027150142096863,9.724267216113748,2.313816
5866942244,2.812739330991701,2.9037102669982024,0.7462853307264268,0.7
555145027985632,8.836630385795942,3.251636539205776,2.5487550274946598
,3.8173773825839863,0.7846180488757459,0.7853481215823382,0.7841365381
629113,10.245152205424667,2.4039927936829844,2.9390208385298378,2.8268
921912559195,3.1210219530977685,0.7439208916473314,0.7343247848062643,
7.743237366218968,2.386234805943079,2.285298637948,2.7075502087164995,
3.242582020901076,0.6923116340769615,0.6906169832331099,7.471867201034
43,1.7898285258780118,6.428385759562312,3.5443552724469356,3.753378444
447308,1.8149095219583493,1.823992394377555,1.8335145460670605,8.04695
9738825123,3.3814875574922185,3.389996465296645,4.055910023541834,1.87
43315683855402,1.8787770175969878,6.482701588089328,3.088967536936897,
3.044163646418233,2.920607581729283,2.9088559582058604,1.8665307856461
```

```
046,1.858794242215097,4.444961624695786,2.4762516508066237,3.728538117
890844,1.8216767315332436,1.8127619527479821,1.804746228832856,10.6093
3958839415,1.7926743618160526,1.789112442368427,1.7874177915245757,2.0
783883405489445,2.080557077937798,2.084563767387407,2.090244374560938,
10.695375327069655,2.793937659795565,3.026717897592435,2.9341422773916
22,2.1338146910776787,2.143043863149815,4.983905566428364,2.8564699723
934854,2.8751558597723745,2.1695055696567738,3.7586779544439723,2.1728
774819335896,2.171665898514163,3.916843209210934,2.1636936336706154,3.
031399811643227,2.631332504319084,2.637523606110708,2.131450251998583,
2.121854145157516,3.1081453510177113,2.1034905048077683,2.845721329390
0407,2.442013397225625,2.4206697109067794,2.0798409944282135,1.2723160
84317264,2.0706180092553446,1.2784915111557265,1.6765028822223924,1.29
12940785015836,1.2995658177864549,1.3086486902056607,1.318170841895166
3,2.4726140791891225,1.3369716069181345,2.4708503374686583,1.352920802
5740098,1.7593487899408506,1.3634333134250933,1.3660751529953172,1.869
263923319433,1.3655936422824826,1.7215529556851417,1.357621377438935,1
.7423706793910494,1.3434505380432027,1.3347284821429826,1.827836693384
4261,1.3157818889258355,1.6608073140367254,1.2974182485760877,1.658789
9690281822,1.2826140202948642,1.2773306576441583,1.756725614217029,1.2
720740873526812,0.43706911862122244,0.1776840134146529,0.5487838152788
908,0.18737131003779295,0.19449327021011908,0.6948992133258522,0.21184
78819141959,0.578138652231669,0.23094162655453332,0.2401707986266699,0
.24867970643109627,0.25611999428254506,1.3841153807393913,0.2666325051
336287,0.6455447849288471,0.2700044174104449,0.26879283399101794,0.265
68919681311787,0.2608205691474702,0.25438627318274565,0.24664972975173
804,2.073393235924663,0.22857718747543798,0.21898108063437094,0.209532
21906988462,0.20061744028462314,1.630000684668029,0.1858132120033995,0
.18052984935269367,1.0683158934868393,-0.11521327603398723,-
0.11304453864513375,-0.10903784919552484,-0.1033572420219937,-
9.623528184966745e-2,-8.796354256479622e-2,-7.888067014559064e-2,-
6.935851845608498e-2,-5.978692550525322e-2,-5.055775343311675e-2,-
4.204884562869027e-2,-3.460855777724148e-2,-2.854149613760537e-2,-
2.4096046926157966e-2,-2.1454207355934063e-2,-2.0724134649341863e-2,-
2.1935718068768706e-2,-2.503935524666856e-2,-2.9907982912316333e-2,-
3.634227887704067e-2,-4.4078822308048604e-2,-5.280087820826873e-2,-
6.2151364584348556e-2,-7.17474714254156e-2,-8.119633298990192e-2,-
9.01111117751634e-2,-9.812683569028935e-2,-0.10491534005638714,-
0.11019870270709298,-0.11376062215471827]},{"line":
{"color":"rgba(0,0,0,0)"},"mode":"lines","showlegend":false,"type":"sc
atter","x":
[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,2
6,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49
,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,
73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,9
6,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114
,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,1
32,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149
,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,1
67,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184
,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,2
```

02,203,204,205,206,207,208,209,210],"y":
[3.5618192308308303,3.9013251371064146,3.9417149105913167,3.4991504211
919713,3.2891245921386956,3.6519738645272817,3.938012457765381,3.18667
0587553458,4.04522948379301,3.7498179539023893,3.9979370365404217,3.55
15462570676526,3.4773279762260634,3.9151282859013237,3.340333714334039
,3.602225417680449,3.1901128684298223,3.0230753357836972,6.03123814403
4127,3.9071420278446065,4.522996998549692,4.3401788292879715,4.2412510
215724195,4.738236438333569,3.7906917798322515,3.868793557712037,3.967
9022989042414,4.874042998659119,5.44808223977609,2.80029757192184,2.80
4304261371449,2.8099848685449804,11.831537070449043,4.4210864410295185
,4.920009185326995,5.010980121333496,2.8535551850617207,2.862784357133
857,10.943900240131237,5.35890639354107,4.656024881829953,5.9246472369
19281,2.89188790321104,2.8926179759176325,2.8914063924982054,12.352422
059759961,4.5112626480182785,5.046290692865131,4.934162045591213,5.228
291807433063,2.851190745982626,2.8415946391415585,9.850507220554261,4.
493504660278373,4.392568492283294,4.814820063051794,5.34985187523637,2
.7995814884122554,2.797886837568404,9.579137055369724,3.89709838021330
6,8.535655613897607,5.65162512678223,5.860648298782602,3.9221793762936
43,3.931262248712849,3.9407844004023547,10.154229593160418,5.488757411
827512,5.4972663196319385,6.163179877877129,3.9816014227208343,3.98604
6871932282,8.589971442424622,5.1962373912721915,5.151433500753527,5.02
7877436064577,5.016125812541155,3.9738006399813988,3.966064096550391,6
.5522314790310805,4.583521505141918,5.835807972226139,3.92894658586853
75,3.9200318070832765,3.91201608316815,12.716609442729444,3.8999442161
513467,3.896382296703721,3.89468764585987,4.185658194884239,4.18782693
2273092,4.191833621722701,4.197514228896232,12.80264518140495,4.901207
514130859,5.133987751927729,5.041412131726917,4.241084545412972,4.2503
1371748511,7.091175420763658,4.96373982672878,4.982425714107668,4.2767
75423992068,5.865947808779266,4.280147336268884,4.278935752849457,6.02
4113063546228,4.2709634880059095,5.1386696659785205,4.7386023586543775
,4.744793460446003,4.2387201063338775,4.22912399949281,5.2154152053530
06,4.2107603591430625,4.952991183725334,4.549283251560919,4.5279395652
420735,4.187110848763508,3.379585938652558,4.177887863590639,3.3857613
654910206,3.7837727365576868,3.3985639328368777,3.406835672121749,3.41
5918544540955,3.4254406962304604,4.579883933524417,3.4442414612534287,
4.5781201918039525,3.4601906569093037,3.8666186442761448,3.47070316776
03872,3.4733450073306114,3.976533777654727,3.4728634966177765,3.828822
810020436,3.464891231774229,3.8496405337263435,3.450720392378497,3.441
9983364782767,3.9351065477197205,3.4230517432611296,3.7680771683720193
,3.404688102911382,3.7660598233634763,3.3898838746301583,3.38460051197
94524,3.8639954685523232,3.379343941687975,2.5443389729565165,2.284953
8677499472,2.656053669614185,2.294641164373087,2.3017631245454133,2.80
2169067661146,2.31911773624949,2.6854085065669633,2.3382114808898273,2
.3474406529619642,2.3559495607663905,2.3633898486178393,3.491385235074
6852,2.373902359468923,2.752814639264141,2.377274271745739,2.376062688
326312,2.372959051148412,2.3680904234827644,2.3616561275180397,2.35391
95840870324,4.180663090259957,2.335847041810732,2.326250934969665,2.31
6802073405179,2.3078872946199174,3.7372705390033234,2.293083066338694,
2.2877997036879876,3.1755857478221334,1.992056578301307,1.994225315690
1604,1.9982320051397693,2.0039126123133,2.0110345724856264,2.019306311

7704977,2.0283891841897033,2.037911335879209,2.047482928830041,2.05671
21009021774,2.0652210087066036,2.072661296558053,2.0787283581976888,2.
0831738074091364,2.08581564697936,2.0865457196859523,2.085334136266525
6,2.0822304990886256,2.077361871422978,2.0709275754582537,2.0631910320
272455,2.0544689761270254,2.0451184897509456,2.0355223829098783,2.0260
73521345392,2.017158742560131,2.0091430186450046,2.002354514278907,1.9
970711516282011,1.9935092321805759]},
{"fill":"tonexty","fillcolor":"rgba(255, 0, 0, 0.2)","line":
{"color":"rgba(0,0,0,0)"},"mode":"lines","name":"95% confidence
interval","type":"scatter","x":
[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,2
6,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49
,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,
73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,9
6,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114
,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,1
32,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149
,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,1
67,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184
,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,2
02,203,204,205,206,207,208,209,210],"y":[-
0.13486323362414554,0.20464267265143876,0.2450324461363409,-
0.19753204326300433,-0.4075578723162805,-4.470859992769438e-
2,0.24132999331040494,-
0.5100118769015178,0.3485470193380338,5.313548944741364e-
2,0.3012545720854458,-0.14513620738732325,-
0.21935448822891224,0.21844582144634783,-0.35634875012093703,-
9.445704677452649e-2,-0.5065695960251535,-
0.6736071286712784,2.334555679579151,0.21045956338963068,0.82631453409
47164,0.6434963648329961,0.5445685571174437,1.0415539738785933,9.40093
1537727564e-
2,0.1721110932570613,0.27121983444926556,1.1773605342041429,1.75139977
53211146,-0.8963848925331355,-0.8923782030835266,-
0.8866975959099954,8.134854605994066,0.7244039765745427,1.223326720872
019,1.3142976568785207,-0.8431272793932549,-
0.8338981073211185,7.24721777567626,1.6622239290860943,0.9593424173749
78,2.2279647724643046,-0.8047945612439358,-0.8040644885373435,-
0.8052760719567704,8.655739595304984,0.8145801835633026,1.349608228410
156,1.2374795811362378,1.5316093429780868,-0.8454917184723503,-
0.8550878253134174,6.1538247560992865,0.7968221958233972,0.69588602782
83181,1.1181375985968178,1.6531694107813943,-0.8971009760427202,-
0.8987956268865718,5.882454590914747,0.20041591575833007,4.83897314944
263,1.9549426623272539,2.1639658343276262,0.2254969118386676,0.2345797
8425787318,0.24410193594737883,6.457547128705441,1.7920749473725368,1.
800583855176963,2.4664974134221525,0.28491895826585845,0.2893644074773
061,4.893288977969647,1.4995549268172153,1.4547510362985512,1.33119497
16096012,1.3194433480861787,0.2771181755264229,0.2693816320954152,2.85
5549014576104,0.886839040686942,2.1391255077711624,0.2322641214135619,
0.22334934262830042,0.21533361871317425,9.019926978274468,0.2032617516
9637084,0.19969983224874532,0.19800518140489398,0.4889757304292628,0.4

```
911444678181165,0.4951511572677254,0.5008317644412563,9.10596271694997
3,1.2045250496758833,1.4373052874727534,1.3447296672719404,0.544402080
957997,0.5536312530301335,3.394492956308682,1.2670573622738037,1.28574
32496526928,0.5800929595370921,2.1692653443242906,0.5834648718139079,0
.5822532883944813,2.3274305990912523,0.5742810235509337,1.441987201523
545,1.0419198941994021,1.0481109959910264,0.5420376418789012,0.5324415
350378344,1.5187327408980296,0.5140778946880866,1.256308719270359,0.85
26007871059433,0.8312571007870977,0.49042838430853175,-
0.31709652580241765,0.48120539913566285,-
0.31092109896395526,8.70902721027107e-2,-0.2981185316180981,-
0.28984679233322685,-0.28076391991402105,-
0.2712417682245154,0.8832014690694407,-
0.25244100320154716,0.8814377273489766,-
0.2364918075456719,0.16993617982116893,-0.22597929669458838,-
0.22333745712436448,0.2798513131997513,-
0.22381896783719912,0.13214034556545995,-
0.23179123268074675,0.15295806927136768,-0.24596207207647902,-
0.25468412797669915,0.2384240832647444,-
0.27363072119384624,7.139470391704372e-2,-
0.29199436154359404,6.93773589085005e-2,-0.30679858982481756,-
0.3120819524755234,0.1673130040973474,-0.3173385227670005,-
1.1523434914984594,-1.4117285967050288,-1.0406287948407908,-
1.4020413000818888,-1.3949193399095625,-0.8945133967938295,-
1.3775647282054857,-1.0112739578880126,-1.3584709835651485,-
1.3492418114930118,-1.3407329036885853,-1.3332926158371365,-
0.20529722938029038,-1.322780104986053,-0.9438678251908346,-
1.3194081927092367,-1.3206197761286638,-1.3237234133065638,-
1.3285920409722114,-1.3350263369369362,-
1.3427628803679437,0.48398062580498147,-1.3608354226442438,-
1.3704315294853107,-1.379880391049797,-
1.3887951698350585,4.058807454834734e-2,-1.4035993981162822,-
1.408882760766988,-0.5210967166328424,-1.704625886153669,-
1.7024571487648155,-1.6984504593152066,-1.6927698521416754,-
1.6856478919693492,-1.677376152684478,-1.6682932802652723,-
1.6587711285757667,-1.649199535624935,-1.6399703635527985,-
1.631461455748372,-1.6240211678969232,-1.617954106257287,-
1.6135086570458397,-1.6108668174756158,-1.6101367447690236,-
1.6113483281884504,-1.6144519653663503,-1.619320593031998,-
1.6257548889967224,-1.6334914324277303,-1.6422134883279504,-
1.6515639747040303,-1.6611600815450973,-1.6706089431095836,-
1.6795237218948451,-1.687539445809971,-1.6943279501760689,-
1.6996113128267747,-1.7031732322744]}],"layout":{"template":{"data":
{"bar":[{"error_x":{"color":"#2a3f5f"},"error_y":
{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
```

```
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
```

```
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
```

lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Analysis for Harris County"},"xaxis":{"title":{"text":"number
of days since the first case"}},"yaxis":{"title":{"text":"No. of
deaths"}}}}