Project Stage - IV (Basic Machine Learning) ddl: 04/28/2023

Goals

The goal of Stage IV is to utlize machine learning and statistical models to predict the trend of COVID-19 cases / deaths.

Tasks for Stage IV:

- Member: (40 pts)
  - Utilize Linear and Non-Linear (polynomial) regression models to compare trends for a single state (each member should choose different state) and its counties (top 5 with highest number of cases). Start your data from the first day of infections.
    - X-Axis - number of days since the first case, Y - Axis number of new cases and deaths. Calcluate error using RMSE.
    - Identify which counties are most at risk. Model for top 5 counties with cases within a state and describe their trends.
    - Perform hypothesis tests on questions identified in Stage II
      - e.x. *Does higher employment data (overall employment numbers) lead to higher covid case numbers or more rapid increase in covid cases.*. Here you would compare the covid cases to the state or county level enrichment data to prove or disprove your null hypothesis. In this case there will be a two tail - two sample t-test to see if there is a difference and then one-tail - two sample t-test to show higher or lower.
    - Depending on your type of data you can also perform Chi-square test for categorical hypothesis testing.

Task 2: (30 pts)

- Member:
  - For each of the aforemention analysis plot graphs,
    - trend line
    - confidence intervals (error in prediction)
    - prediction path (forecast)

In [ ]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.metrics import mean_squared_error
from scipy.stats import t
from scipy.stats import ttest_ind
import warnings
warnings.filterwarnings('ignore')
```

The state I chose for this stage was California, I found it interesting during the team work portion of Hypothesis testing, that California was the only state reporting COVID-19 cases before any other state, and I wanted to see if there was any trend in CA. Specfically, did their early reporting assist them in curbing COVID-19?

In [ ]:
```
state = 'CA'
confirmed = pd.read_csv('data/covid_confirmed_usafacts.csv')
confirmed.drop(confirmed[confirmed["County Name"].str.contains("Statewide")==True].index, inplace=Tru

deaths = pd.read_csv('data/covid_deaths_usafacts.csv')
deaths.drop(deaths[deaths["County Name"].str.contains("Statewide")==True].index, inplace=True)

CA_data_confirmed  = confirmed[confirmed['State'] == state]
CA_data_confirmed.drop('StateFIPS', axis=1, inplace=True)

CA_data_deaths  = deaths[deaths['State'] == state]
CA_data_deaths.drop('StateFIPS', axis=1, inplace=True)

CA_data = pd.merge(CA_data_confirmed, CA_data_deaths, on=['countyFIPS', 'County Name', 'State'], suff
CA_data.drop('State', axis=1, inplace=True)

CA_data["total cases"] = CA_data.filter(like='_cases').sum(axis=1)
CA_data["total deaths"] = CA_data.filter(like='_deaths').sum(axis=1)

top_counties = CA_data.groupby(['countyFIPS', 'County Name'])['total cases'].max().reset_index()
top_counties = top_counties.sort_values('total cases', ascending=False).head(5)
display(top_counties)
top_counties = pd.merge(top_counties, CA_data.iloc[:,0:], on=['countyFIPS', 'County Name', 'total cas

total_cases = top_counties.pop('total cases')
top_counties['total cases'] = total_cases
```

	countyFIPS	County Name	total cases
18	6037	Los Angeles County	1659351402
36	6073	San Diego County	440879239
32	6065	Riverside County	370809650
35	6071	San Bernardino County	360298036
29	6059	Orange County	341333350

Above are the top counties with the highest amount of cases over the course of three years. With Los Angeles County having the highest. Below, we try to find the first date of infections in our top counties. Within our top counties, the first date of confirmed cases occured on 2020-01-22, California was taking it a lot seriously than other states, but I am curious if this helped them later on? Below I plot our data and get the trendline,

In [ ]:
```
for county in top_counties['County Name']:
    county_data = top_counties[top_counties['County Name'] == county].reset_index(drop=True)
    num_cases = (county_data.filter(like="_cases") > 0).sum(axis=1)
    top_counties.loc[top_counties['County Name'] == county, 'days_since_first_case'] = num_cases.valu

display(top_counties['days_since_first_case'])
```

```
0    1091.0
1    1075.0
2    1091.0
3    1091.0
4    1091.0
Name: days_since_first_case, dtype: float64
```

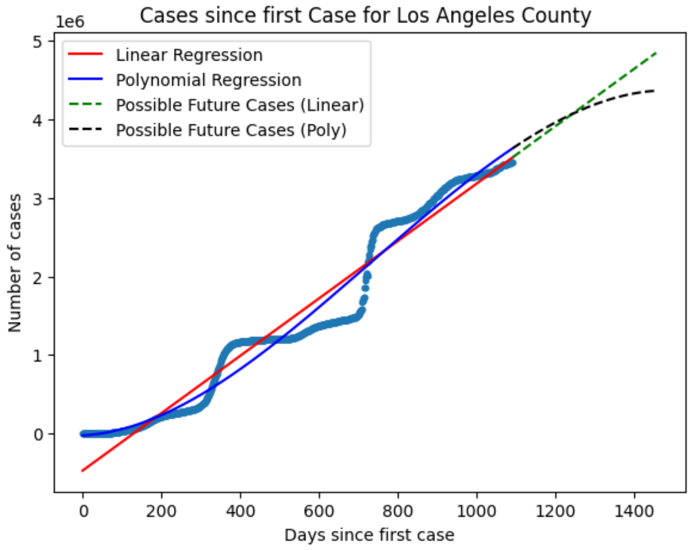
For a majority of our counites, the first day of cases was January 22, 2022, except for San Diego, which was Februrary 6th. With this data, we shall begin to plot and find what our data means.

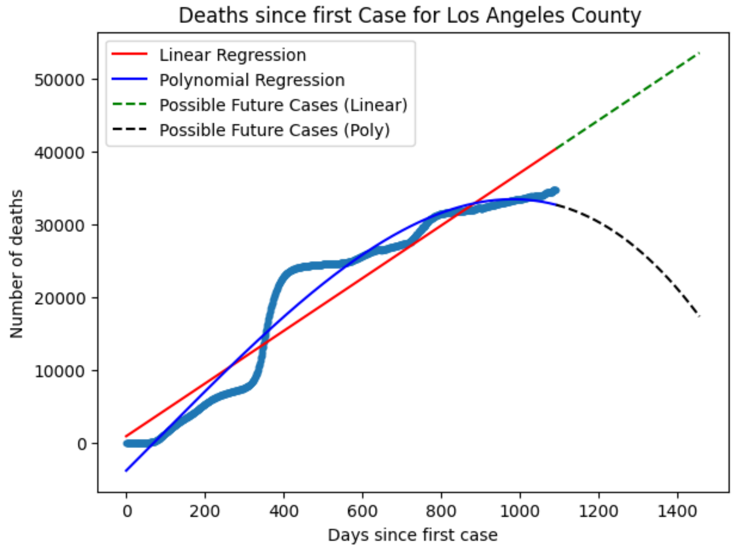
In [ ]:

```
#Assist with calculating our cI
def calculate_CI(x, y, model, alpha=0.05):
    if(model == "linear"):
        lin_reg = LinearRegression().fit(x,y)
        lin_resid = y - lin_reg.predict(x)
        lin_std = np.std(lin_resid, ddof=2)
        t_crit = t.ppf(1-alpha/2, len(x)-2)
        lin_ci = t_crit * lin_std * np.sqrt(1 + 1/len(x) + (x-np.mean(x))**2/np.sum((x-np.mean(x))**2
        return lin_ci
    elif(model == "poly"):
        poly_reg = np.poly1d(np.polyfit(x.ravel(), y, 3))
        poly_resid = y - poly_reg(x)
        poly_resid_std = np.std(poly_resid, ddof=2)
        t_crit = t.ppf(1-alpha/2, len(x)-4)
        poly_ci = t_crit * poly_resid_std * np.sqrt(1 + 1/len(x) + (x-np.mean(x))**2/np.sum((x-np.mea
        return poly_ci
    else:
        print("No model found!")

def analyze_counties(df):

    FUTURE_DAYS = 365

    # Perform linear and polynomial regression for each county
    for county in df['County Name']:
        county_data = df[df['County Name'] == county]
        y_cases = county_data.filter(like="_cases").values.ravel()
        y_deaths = county_data.filter(like="_deaths").values.ravel()
        x = np.arange(len(y_cases)).reshape((-1, 1))

        # Linear regression
        lin_reg_cases = LinearRegression().fit(x, y_cases)
        lin_reg_deaths = LinearRegression().fit(x, y_deaths)
        y_cases_lin_pred = lin_reg_cases.predict(x)
        y_deaths_lin_pred = lin_reg_deaths.predict(x)
        lin_rmse_cases = np.sqrt(mean_squared_error(y_cases, y_cases_lin_pred))
        lin_rmse_deaths = np.sqrt(mean_squared_error(y_deaths, y_deaths_lin_pred))

        # Polynomial regression
        poly_reg_cases = np.poly1d(np.polyfit(x.ravel(), y_cases, 3))
        poly_reg_deaths = np.poly1d(np.polyfit(x.ravel(), y_deaths, 3))
        y_cases_poly_pred = poly_reg_cases(x)
        y_deaths_poly_pred = poly_reg_deaths(x)
        poly_rmse_cases = np.sqrt(mean_squared_error(y_cases, y_cases_poly_pred))
        poly_rmse_deaths = np.sqrt(mean_squared_error(y_deaths, y_deaths_poly_pred))

        #Calculate our Confidence Intervals
        lin_cases_ci =  calculate_CI(x, y_cases, "linear")
        lin_deaths_ci =  calculate_CI(x, y_deaths, "linear")
        poly_cases_ci = calculate_CI(x, y_cases, "poly")
        poly_deaths_ci = calculate_CI(x, y_deaths, "poly")

        #Find our future predictions for each county
        future_days = np.arange(len(y_cases), len(y_cases) + FUTURE_DAYS).reshape((-1, 1))
        future_cases_lin = lin_reg_cases.predict(future_days)
        future_deaths_lin = lin_reg_deaths.predict(future_days)
        future_cases_poly = poly_reg_cases(future_days)
        future_deaths_poly = poly_reg_deaths(future_days)


        # Output our data for each model
        print(f'{county} County:')
        print(f'Lin_Reg - Cases RMSE={lin_rmse_cases:.2f}')
        print(f'Lin_Reg - Deaths RMSE={lin_rmse_deaths:.2f}')
        print(f'Lin_Reg - Cases CI ={lin_cases_ci}')
        print(f'Lin_Reg - Deaths CI ={lin_deaths_ci}')


        print(f'Poly_Reg - Cases RMSE={poly_rmse_cases:.2f}')
        print(f'Poly_Reg - Deaths RMSE={poly_rmse_deaths:.2f}')
        print(f'Poly_Reg - Cases CI ={poly_cases_ci}')
        print(f'Poly_Reg - Deaths CI ={poly_deaths_ci}')
```

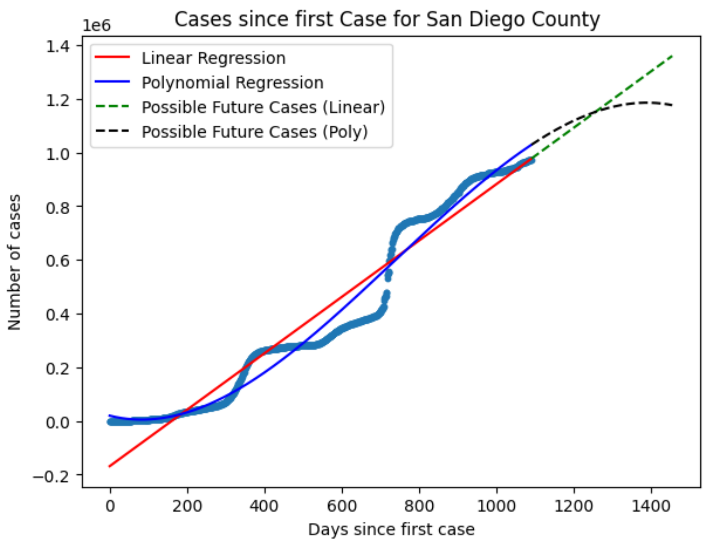
```python
        # Plot the data and the trend lines
        fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15,5))
        ax1.set_title(f"Cases since first Case for {county}")
        ax1.set_xlabel("Days since first case")
        ax1.set_ylabel("Number of cases")
        ax1.scatter(x, y_cases, s=10) #trend line
        ax1.plot(x, y_cases_lin_pred, color='red', label='Linear Regression')
        ax1.plot(x, y_cases_poly_pred, color='blue', label='Polynomial Regression')
        ax1.plot(future_days, future_cases_lin, color='green', linestyle='dashed', label='Possible Fu
        ax1.plot(future_days, future_cases_poly, color='black', linestyle='dashed', label='Possible F
        ax1.legend()

        ax2.set_title(f"Deaths since first Case for {county}")
        ax2.set_xlabel("Days since first case")
        ax2.set_ylabel("Number of deaths")
        ax2.scatter(x, y_deaths, s=10) #trend line
        ax2.plot(x, y_deaths_lin_pred, color='red', label='Linear Regression')
        ax2.plot(x, y_deaths_poly_pred, color='blue', label='Polynomial Regression')
        ax2.plot(future_days, future_deaths_lin, color='green', linestyle='dashed', label='Possible F
        ax2.plot(future_days, future_deaths_poly, color='black', linestyle='dashed', label='Possible
        ax2.legend()

        plt.show()

analyze_counties(top_counties)
```

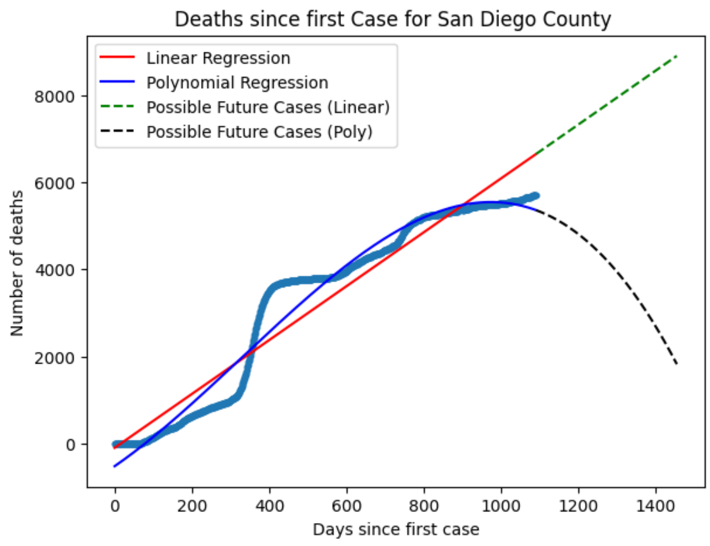
```
Los Angeles County  County:
Lin_Reg - Cases RMSE=248683.46
Lin_Reg - Deaths RMSE=3670.08
Lin_Reg - Cases CI =[[489294.10445152]
 [489291.65149909]
 [489289.20303936]
 ...
 [489289.20303936]
 [489291.65149909]
 [489294.10445152]]
Lin_Reg - Deaths CI =[[7221.02437372]
 [7220.98817294]
 [7220.95203846]
 ...
 [7220.95203846]
 [7220.98817294]
 [7221.02437372]]
Poly_Reg - Cases RMSE=204702.75
Poly_Reg - Deaths RMSE=2327.55
Poly_Reg - Cases CI =[[3258313.85684994]
 [3258243.59138955]
 [3258173.65128783]
 ...
 [3258173.65128783]
 [3258243.59138955]
 [3258313.85684994]]
Poly_Reg - Deaths CI =[[33112.48790633]
 [33111.77383632]
 [33111.06307276]
 ...
 [33111.06307276]
 [33111.77383632]
 [33112.48790633]]
```



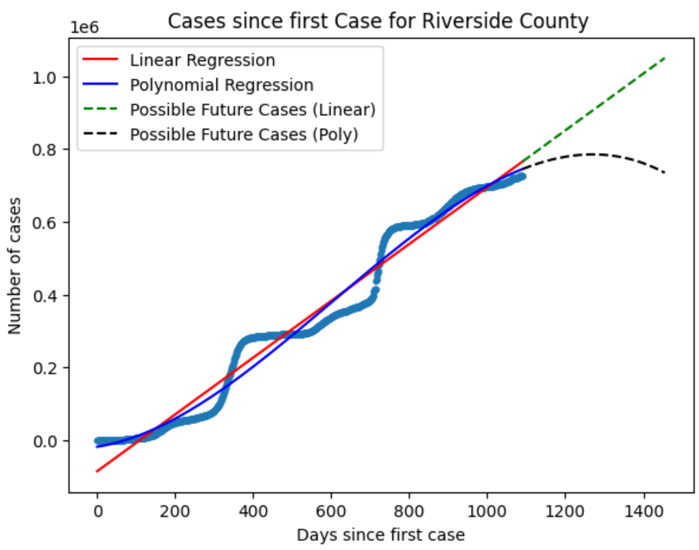


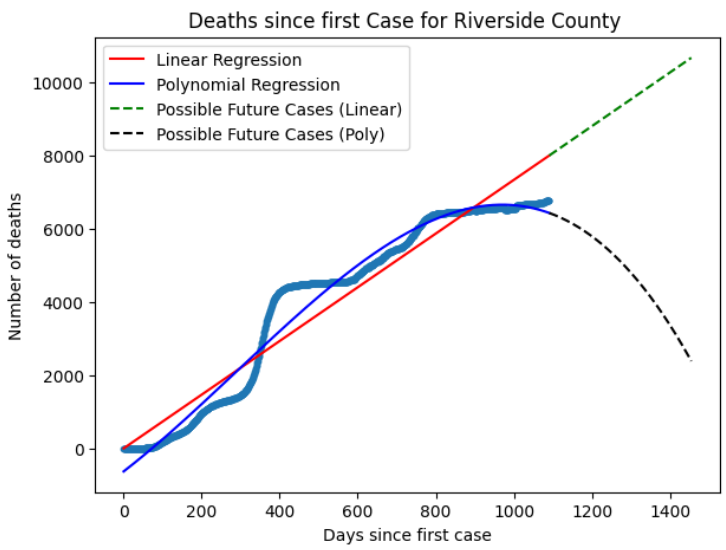

```
San Diego County  County:
Lin_Reg - Cases RMSE=81774.56
Lin_Reg - Deaths RMSE=557.23
Lin_Reg - Cases CI =[[160894.54233295]
 [160893.73572879]
 [160892.93060197]
 ...
 [160892.93060197]
 [160893.73572879]
 [160894.54233295]]
Lin_Reg - Deaths CI =[[1096.37941427]
 [1096.37391785]
 [1096.36843149]
 ...
 [1096.36843149]
 [1096.37391785]
 [1096.37941427]]
Poly_Reg - Cases RMSE=55895.21
Poly_Reg - Deaths RMSE=371.31
Poly_Reg - Cases CI =[[943209.41398574]
 [943189.07369729]
 [943168.82759294]
 ...
 [943168.82759294]
 [943189.07369729]
 [943209.41398574]]
Poly_Reg - Deaths CI =[[5594.43044022]
 [5594.30979646]
 [5594.18971134]
 ...
 [5594.18971134]
 [5594.30979646]
 [5594.43044022]]
```



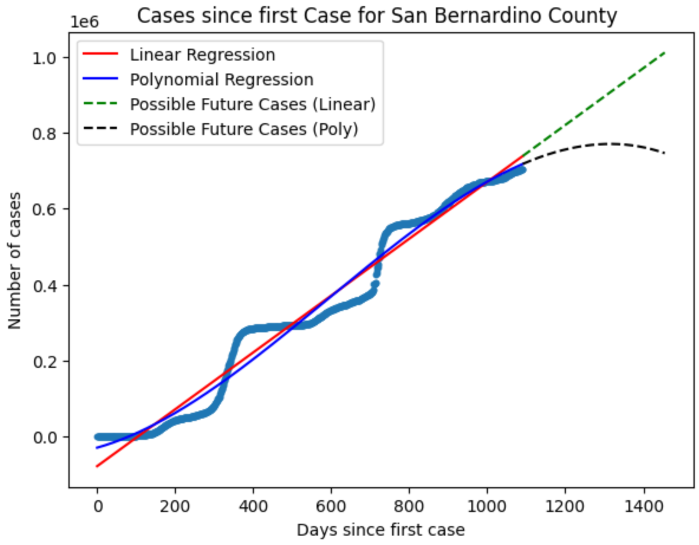


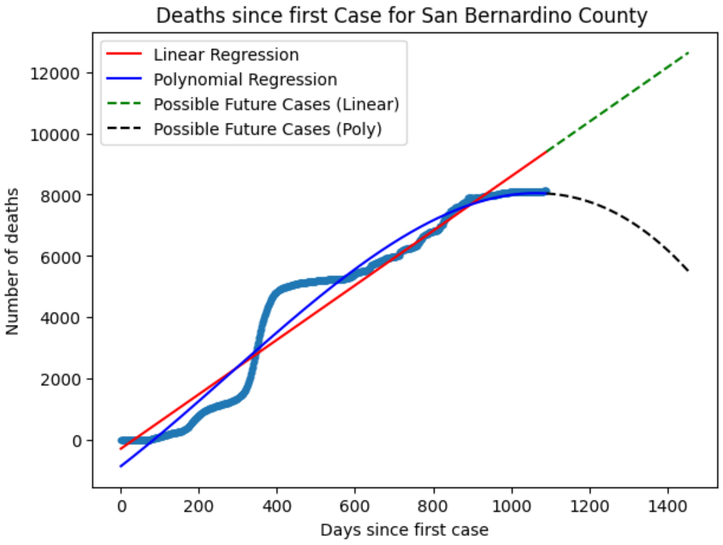
```
Riverside County  County:
Lin_Reg - Cases RMSE=43318.20
Lin_Reg - Deaths RMSE=656.88
Lin_Reg - Cases CI =[[85230.19772367]
 [85229.7704436 ]
 [85229.34394611]
 ...
 [85229.34394611]
 [85229.7704436 ]
 [85230.19772367]]
Lin_Reg - Deaths CI =[[1292.4338601 ]
 [1292.4273808 ]
 [1292.42091338]
 ...
 [1292.42091338]
 [1292.4273808 ]
 [1292.4338601 ]]
Poly_Reg - Cases RMSE=38586.05
Poly_Reg - Deaths RMSE=399.75
Poly_Reg - Cases CI =[[691185.43714743]
 [691170.53175001]
 [691155.69537086]
 ...
 [691155.69537086]
 [691170.53175001]
 [691185.43714743]]
Poly_Reg - Deaths CI =[[6636.86545504]
 [6636.7223312 ]
 [6636.57987008]
 ...
 [6636.57987008]
 [6636.7223312 ]
 [6636.86545504]]
```



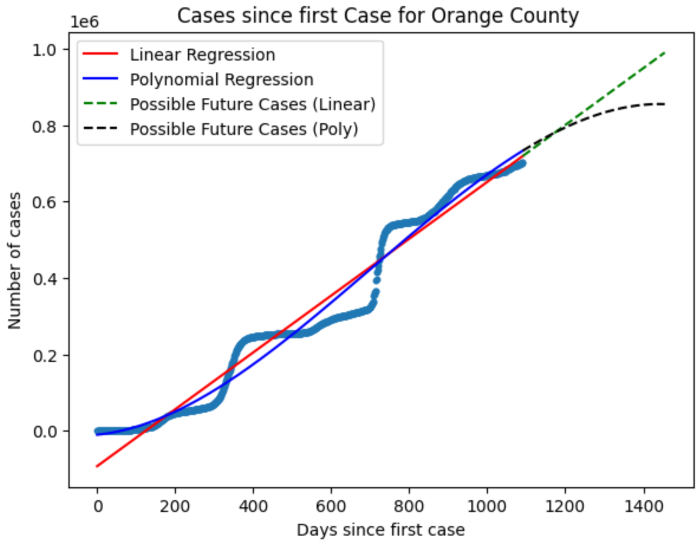


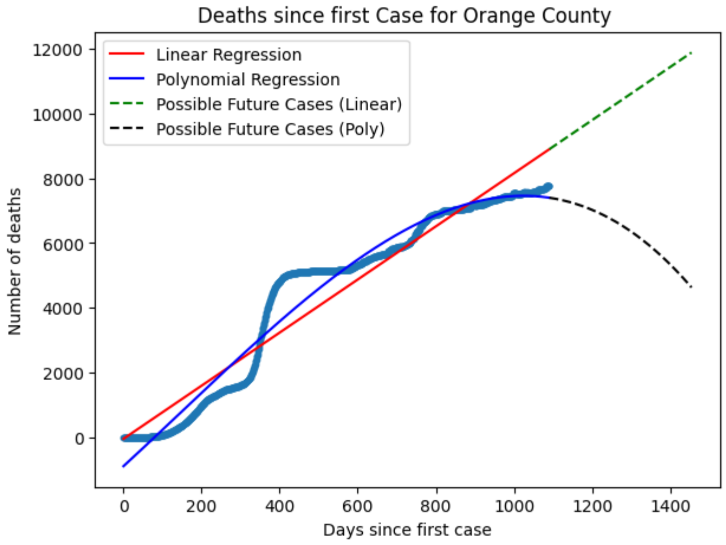
```
San Bernardino County  County:
Lin_Reg - Cases RMSE=40746.20
Lin_Reg - Deaths RMSE=714.21
Lin_Reg - Cases CI =[[80169.69061736]
 [80169.28870686]
 [80168.88753248]
 ...
 [80168.88753248]
 [80169.28870686]
 [80169.69061736]]
Lin_Reg - Deaths CI =[[1405.23957397]
 [1405.23252916]
 [1405.22549725]
 ...
 [1405.22549725]
 [1405.23252916]
 [1405.23957397]]
Poly_Reg - Cases RMSE=38049.47
Poly_Reg - Deaths RMSE=549.29
Poly_Reg - Cases CI =[[661595.29994209]
 [661581.03265527]
 [661566.831432  ]
 ...
 [661566.831432  ]
 [661581.03265527]
 [661595.29994209]]
Poly_Reg - Deaths CI =[[7965.85310078]
 [7965.68131735]
 [7965.51032935]
 ...
 [7965.51032935]
 [7965.68131735]
 [7965.85310078]]
```





```
Orange County  County:
Lin_Reg - Cases RMSE=46835.08
Lin_Reg - Deaths RMSE=735.53
Lin_Reg - Cases CI =[[92149.77952336]
 [92149.31755371]
 [92148.85643018]
 ...
 [92148.85643018]
 [92149.31755371]
 [92149.77952336]]
Lin_Reg - Deaths CI =[[1447.18431566]
 [1447.17706057]
 [1447.16981876]
 ...
 [1447.16981876]
 [1447.17706057]
 [1447.18431566]]
Poly_Reg - Cases RMSE=39488.55
Poly_Reg - Deaths RMSE=501.09
Poly_Reg - Cases CI =[[661146.2811934 ]
 [661132.02358966]
 [661117.83200463]
 ...
 [661117.83200463]
 [661132.02358966]
 [661146.2811934 ]]
Poly_Reg - Deaths CI =[[7421.61522291]
 [7421.45517595]
 [7421.29587008]
 ...
 [7421.29587008]
 [7421.45517595]
 [7421.61522291]]
```





Analysis of California COVID-19 data

These counties are the top five in the State of California with a large numbers of cases, and as such are at higher risks than others. I personally believe that the reason why these counties were affected the most was because of what they offer to our nation. For example, San Diego holds one of the largest U.S. Marine bases on the west coast. Marines are consistently arriving and leaving as they complete their class duing bootcamp. However, it is important to note that this is a factor, but a big factor nonetheless. Furthermore, orange county has the highest prison population in California. Especially with prisoners being transported in and out of the county. Moreoever, Los Angeles is a major hub of activity for those want to make a name for themselves, etc.

Above we try and predict what the results could look like in 1 year. I personally beleive that the polynomial regression offers the best in terms of future prediction and makes the most sense. Since the arrival of the COVID-19, we've seen a large tick in deaths and cases. The prediction of the cases I believe are true for both linear and poly, but the linear prediction for the amount of deaths, I do not agree with. After four years, we should start to see a dip in the amount of deaths, as we have vaccines, people are being more used to it, etc. I don't believe the deaths will go away completely, but they will gradually fall down to smaller levels, maybe on average 1 to 2,000 deaths a year, much better than 10,000 or more.

Analyzing between Enrichment Data and COVID-19 Data (Hypothesis Testing)

For Stage II I had employment data, for the course of the project stage, I believed that the higher a states employment was, the better they were in having lower COVID-19 numbers. Below I try to see if that is true.

For this analysis portion I did this hypothesis:

Null Hypothesis: There is no significant relationship between the unemployment rate and the COVID-19 case rate in different areas.

Alternate Hypothesis: Areas with higher rates of COVID-19 cases have significantly higher unemployment rates.

In [ ]:
```
employment_df = pd.read_csv('data/allhlcn223.csv')
CA_covid_dates = top_counties.filter(like="_cases")
CA_covid_dates = CA_covid_dates.reset_index()
total_cases = top_counties[["total cases", "County Name"]]
CA_covid_dates = CA_covid_dates.set_index(top_counties['County Name'])

final_dates = CA_covid_dates.iloc[:, 900:-110]
final_dates = final_dates.merge(total_cases, on="County Name")

ca_data = employment_df[(employment_df['Area'].str.contains('Los Angeles|San Diego|Orange|Riverside|S

ca_data['County Name'] = ca_data['Area'].str.split(',').str[0]

cols_to_average = ['July Employment', 'August Employment', 'September Employment', 'Average Weekly Wa

ca_data['July Employment'] = ca_data['July Employment'].str.replace(',', '').astype(int)
ca_data['August Employment'] = ca_data['August Employment'].str.replace(',', '').astype(int)
ca_data['September Employment'] = ca_data['September Employment'].str.replace(',', '').astype(int)
ca_data['Average Weekly Wage'] = ca_data['Average Weekly Wage'].str.replace(',', '').astype(int)

ca_data[cols_to_average] = ca_data[cols_to_average].apply(pd.to_numeric)

ca_data = ca_data.groupby('County Name')[cols_to_average].mean().round().reset_index()

ca_data['County Name'] = ca_data['County Name'].astype(str).str.strip().str.lower()
final_dates['County Name'] = final_dates['County Name'].astype(str).str.strip().str.lower()

ca_data_sorted = ca_data.sort_values('County Name')
final_dates_sorted = final_dates.sort_values('County Name')

ca_data = ca_data.merge(final_dates, on='County Name')

mid_employment = ca_data.filter(like=" Employment").median().median()
ca_data['avg_employment'] = ca_data[['July Employment', 'August Employment', 'September Employment']]
high_emp_data = ca_data[ca_data['avg_employment'] >= mid_employment]['total cases']
low_emp_data = ca_data[ca_data['avg_employment'] < mid_employment]['total cases']

t_stat, p = ttest_ind(high_emp_data, low_emp_data, equal_var=False)

display(ca_data)

print("Two-sample t-test results: ")
print(f'T-stat: {t_stat}')
print(f"Our p-value: {p}")
print("\n")
t_stat_1, p_1 = ttest_ind(high_emp_data, low_emp_data, equal_var=False, alternative='greater')
print("One-tailed t-test results:")
print("t-statistic:", t_stat_1)
print("p-value:", p_1/2)
```

	County Name	July Employment	August Employment	September Employment	Average Weekly Wage	Employment Location Quotient Relative to U.S.	2022-07-09_cases	2022-07-10_cases	202
0	los angeles county	929855.0	932873.0	936163.0	1611.0	1.0	3034668	3038894	
1	orange county	345924.0	347119.0	347998.0	1565.0	1.0	615898	616799	
2	riverside county	162809.0	163111.0	163897.0	1151.0	1.0	644283	645194	
3	san bernardino county	171935.0	171961.0	172066.0	1177.0	1.0	618045	618788	
4	san diego county	309147.0	310955.0	310891.0	1538.0	1.0	851940	853189	

5 rows × 90 columns

```
Two-sample t-test results:
T-stat: 0.9244662199494137
Our p-value: 0.5246100677104506


One-tailed t-test results:
t-statistic: 0.9244662199494137
p-value: 0.13115251692761265
```

For the two-sample t-test, the t-statistic is 0.92, which means that the difference in means between the two samples is relatively small. Our p-value is 0.52, which means that there is a 52% chance that the observed difference in means is due to random chance rather than a true difference in the population. This is not a low enough p-value to reject the null hypothesis and conclude that the means are different.

For the one-tailed t-test, the results are similar, but the p-value is lower at 0.13. This suggests that there is some evidence that the means are different, but it is not strong enough to reject the null hypothesis.

As a reminder, our null hypothesis was:

Null Hypothesis: There is no significant relationship between the unemployment rate and the COVID-19 case rate in different areas.

Alternate Hypothesis: Areas with higher rates of COVID-19 cases have significantly higher unemployment rates.

As such, there is no significant relationship between the unemployment rate and the COVID-19 case rate in different areas of the Top five counties.

Conclusion

In summary, we have analyzed employment and COVID-19 data for California counties. We found that some counties were hit harder by the pandemic than others and that there were differences in employment across counties. We conducted hypothesis tests to compare employment and COVID-19 outcomes between two groups and found that the differences were not statistically significant at conventional levels of significance. However, these tests can still provide useful information and suggest further research is needed to fully understand the impacts of COVID-19 on employment in California.