Task 2

In [32]:
```
# 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
```

In [35]:
```
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()
```

In [36]:
```
plot_results(df1, 'Harris County')
```