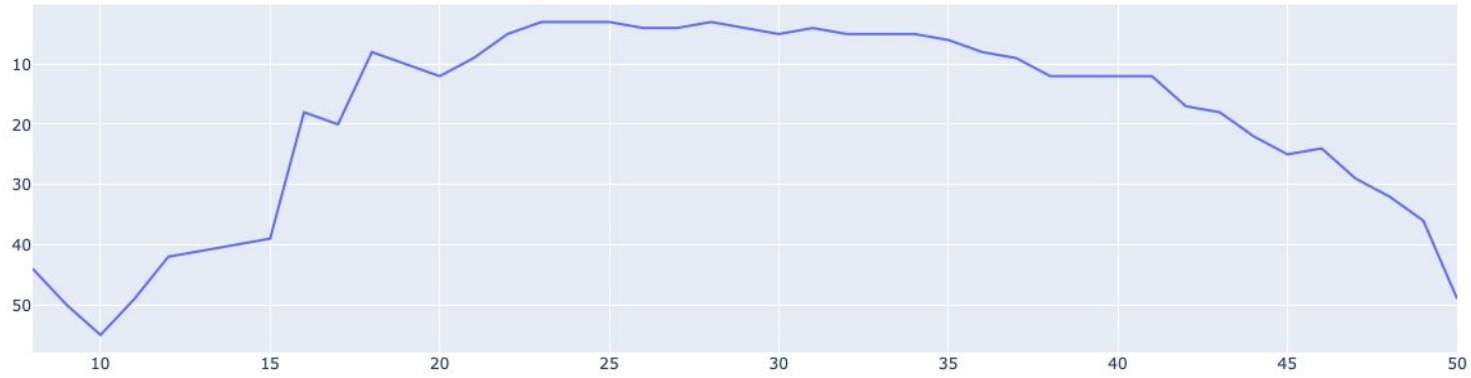
Name:_____ Period:

Do Now: Data Review (5 minutes/ 5 points)

Describe this graph and explain ways that is is useful, not useful, or both



Mini Lesson: Graphing Data with Python (10 minutes/ 5 points)

```
fig = go.Figure([go.Bar(x=dataframe['column_name'], y=dataframe['column_name'])
```

Constructing a visual graph in Python requires commands with several arguments. The line above creates a graph object as a variable “fig”. Broken down it contains…

- A creation of a figure object “go.Figure”
- Designation of the type of figure object “go.Bar”
- Specification for the data to be used on the X and Y axis
    - In this case, “dataframe[‘column_name’]” indicates that the graph will plot the data contained in ‘column_name’ of the dataframe as its axis values

After a graph object is created and stored as a variable “fig”, it can then be plotted and displayed with a simple command

```
fig.show()
```

Activity: Building dataframes (20 minutes/ 10 points)

Based on yesterday's code for creating dataframes, use those data frames to create and display graphs modeling the prompted scenarios

1. Create a vertical bar graph of the top 10 most featured artists descending (count on Y axis, artist name on X axis)

```
fig = go.Figure([go.Bar(x=_______['artist_name'], y=_______['counts'])])
fig.update_layout(title='Top 10 Featured Artists',xaxis=dict(title='Artist Name'), yaxis=dict(title='Songs on Chart'))
fig.show()
```

2. Create a vertical bar graph of the top 10 songs spending the most weeks on the charts (count on Y axis, artist name on X axis)

```
fig = go.Figure([go.Bar(x=____________________, y=_____________________)])
fig.update_layout(title='Top 25 Songs by Weeks on Chart',xaxis=dict(title='Song Title'), yaxis=dict(title='Weeks on Billboard'))
fig.show()
```

3. Pick one song that has been on the chart for at least 25 weeks and graph its weekly position with a line graph (rank on Y axis, date on X axis)

```
fig = _______________(data=go.Scatter(x=____________________, y=_____________________))
fig['layout']['yaxis']['autorange'] = "reversed"
fig.update_layout(title='Song Ranking by Week',xaxis=dict(title='__________'), yaxis=dict(title='__________'))
fig.show()
```

Exit Ticket

Think of your own graph you could create from the data that is not listed in the prompts. Describe what you would be visualizing below and attempt to write the data processing and graphing code for it