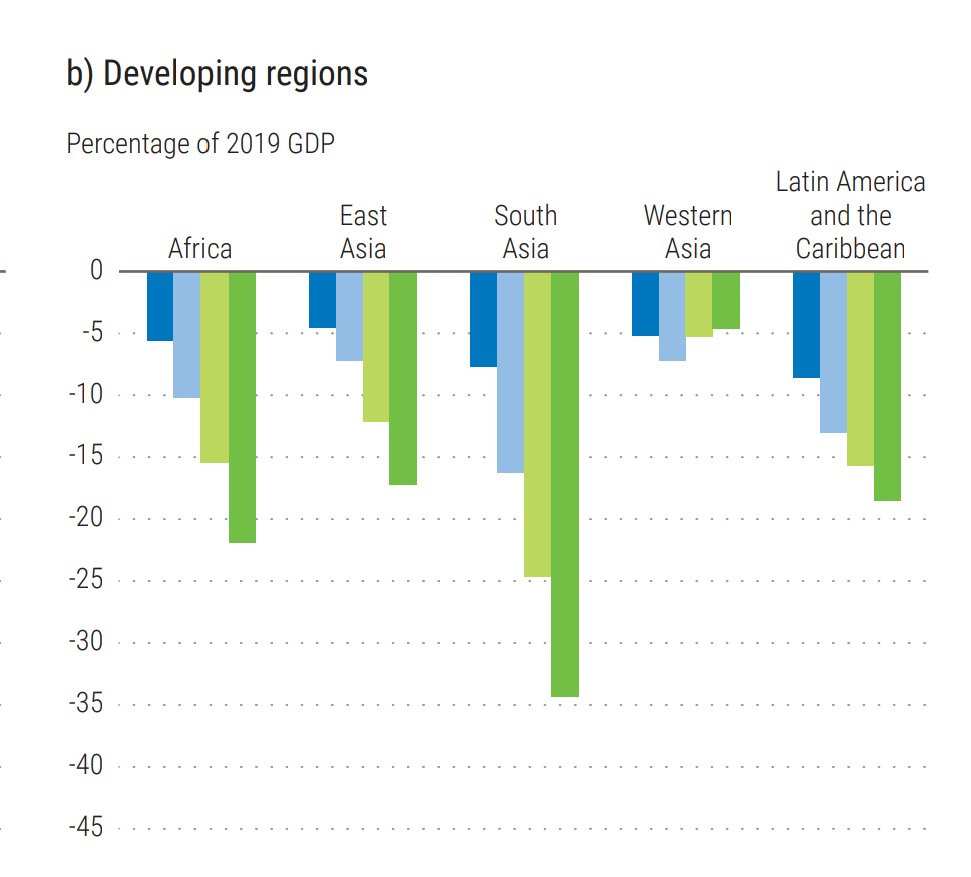
# Research Project Documentation

Date and Time: 2024-06-14 13:03:57

## Input Image: 1711725089159.png



Prompt Used:

Task: Extract detailed information from a given graph image and generate JSON files (data.json, layout.json, and config.json) compatible with Plotly. Use the specified delimiters to wrap each JSON section.  
  
Instructions:  
  
Graph Analysis:  
  
Identify and categorize the plot types present in the graph (e.g., line, scatter, bar, pie).  
Extract the data points for each plot (x-values, y-values, and any other relevant data such as labels for pie charts).  
Note any multiple plots on the same figure and their configurations.  
Generate data.json:  
  
For each plot, include details such as:  
type: The type of plot (e.g., scatter, bar, pie).  
x and y: Arrays of data points for the x and y axes.  
labels and values for pie charts.  
mode: For line and scatter plots (e.g., markers, lines, or markers+lines).  
name: Legend entry for the plot.  
marker: Properties such as color, size, and symbol for markers.  
line: Properties such as color, width, and dash style for lines.  
text: Hover text for each data point.  
hoverinfo: Information displayed on hover (e.g., x+y+text).  
Generate layout.json:  
  
Include details such as:  
title: The title of the graph, including text, font properties (family, size, color).  
xaxis and yaxis: Configuration including title, range, showgrid, gridcolor, zeroline, zerolinecolor, showticklabels, tickangle, tickfont.  
legend: Properties including orientation, x, y, xanchor, font.  
margin: Values for l (left), r (right), b (bottom), t (top), pad.  
plot\_bgcolor and paper\_bgcolor: Background colors of the plot area and paper.  
Generate config.json:  
  
Include configuration settings such as:  
responsive: Whether the graph is responsive (boolean).  
displayModeBar: Whether the mode bar is displayed (boolean).  
modeBarButtonsToRemove: List of mode bar buttons to remove (e.g., ["toImage"]).  
scrollZoom: Whether scrolling zoom is enabled (boolean).  
Output Format:  
  
Wrap each JSON output in specific XML-like tags:  
<data> ... </data>  
<layout> ... </layout>  
<config> ... </config>  
Make sure to follow the JSON formatting guidelines:  
Double Quotes for Keys and Strings: Ensure that all keys and string values in JSON are enclosed in double quotes (").  
Example: Instead of "type": bar, it should be "type": "bar".  
No Trailing Commas: Remove any trailing commas after the last element in arrays ([]) or objects ({}).  
Example: Instead of "color": "rgba(55, 128, 191, 0.6)",, it should be "color": "rgba(55, 128, 191, 0.6)"  
Example Output:  
  
<data>  
[  
 {  
 "type": "scatter",  
 "x": [1, 2, 3],  
 "y": [4, 5, 6],  
 "mode": "markers+lines",  
 "name": "Line Plot",  
 "marker": {  
 "color": "rgba(75, 192, 192, 0.6)",  
 "size": 8  
 },  
 "line": {  
 "color": "rgba(75, 192, 192, 1)",  
 "width": 2  
 }  
 }  
 // Additional plots if present  
]  
</data>  
<layout>  
{  
 "title": {  
 "text": "Graph Title",  
 "font": {  
 "family": "Arial, sans-serif",  
 "size": 24,  
 "color": "#000000"  
 }  
 },  
 "xaxis": {  
 "title": {  
 "text": "X Axis",  
 "font": {  
 "family": "Arial, sans-serif",  
 "size": 18,  
 "color": "#000000"  
 }  
 },  
 "showgrid": true,  
 "gridcolor": "rgba(0, 0, 0, 0.1)",  
 "zeroline": true,  
 "zerolinecolor": "rgba(0, 0, 0, 0.1)"  
 },  
 "yaxis": {  
 "title": {  
 "text": "Y Axis",  
 "font": {  
 "family": "Arial, sans-serif",  
 "size": 18,  
 "color": "#000000"  
 }  
 },  
 "showgrid": true,  
 "gridcolor": "rgba(0, 0, 0, 0.1)",  
 "zeroline": true,  
 "zerolinecolor": "rgba(0, 0, 0, 0.1)"  
 },  
 "legend": {  
 "orientation": "h",  
 "x": 0.5,  
 "xanchor": "center",  
 "y": -0.2,  
 "font": {  
 "family": "Arial, sans-serif",  
 "size": 12,  
 "color": "#000000"  
 }  
 },  
 "margin": {  
 "l": 60,  
 "r": 30,  
 "b": 60,  
 "t": 60  
 },  
 "plot\_bgcolor": "#ffffff",  
 "paper\_bgcolor": "#ffffff"  
}  
</layout>  
<config>  
{  
 "responsive": true,  
 "displayModeBar": true,  
 "modeBarButtonsToRemove": ["toImage"],  
 "scrollZoom": true  
}  
</config>

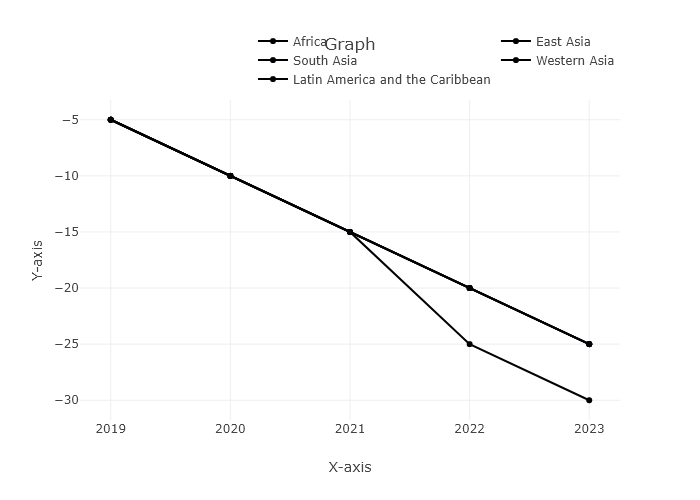
Original JSON:

{  
 "metadata": {  
 "title": "Yearly projected GDP Growth Rate (%) by region from 2019-2023",  
 "xaxis": {  
 "title": "Year"  
 },  
 "yaxis": {  
 "title": "GDP Growth Rate (%)"  
 },  
 "legend": [  
 {  
 "name": "Africa"  
 },  
 {  
 "name": "East Asia"  
 },  
 {  
 "name": "South Asia"  
 },  
 {  
 "name": "Western Asia"  
 },  
 {  
 "name": "Latin America and the Caribbean"  
 }  
 ]  
 },  
 "data": [  
 {  
 "name": "Africa",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-20",  
 "-25"  
 ]  
 },  
 {  
 "name": "East Asia",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-20",  
 "-25"  
 ]  
 },  
 {  
 "name": "South Asia",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-25",  
 "-30"  
 ]  
 },  
 {  
 "name": "Western Asia",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-20",  
 "-25"  
 ]  
 },  
 {  
 "name": "Latin America and the Caribbean",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-20",  
 "-25"  
 ]  
 }  
 ]  
}

Updated JSON:

{  
 "metadata": {  
 "title": "Yearly projected GDP Growth Rate (%) by region from 2019-2023",  
 "xaxis": {  
 "title": "Year"  
 },  
 "yaxis": {  
 "title": "GDP Growth Rate (%)"  
 },  
 "legend": [  
 {  
 "name": "Africa"  
 },  
 {  
 "name": "East Asia"  
 },  
 {  
 "name": "South Asia"  
 },  
 {  
 "name": "Western Asia"  
 },  
 {  
 "name": "Latin America and the Caribbean"  
 }  
 ]  
 },  
 "data": [  
 {  
 "name": "Africa",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-20",  
 "-25"  
 ]  
 },  
 {  
 "name": "East Asia",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-20",  
 "-25"  
 ]  
 },  
 {  
 "name": "South Asia",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-25",  
 "-30"  
 ]  
 },  
 {  
 "name": "Western Asia",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-20",  
 "-25"  
 ]  
 },  
 {  
 "name": "Latin America and the Caribbean",  
 "x": [  
 "2019",  
 "2020",  
 "2021",  
 "2022",  
 "2023"  
 ],  
 "y": [  
 "-5",  
 "-10",  
 "-15",  
 "-20",  
 "-25"  
 ]  
 }  
 ]  
}

Updated Chart:

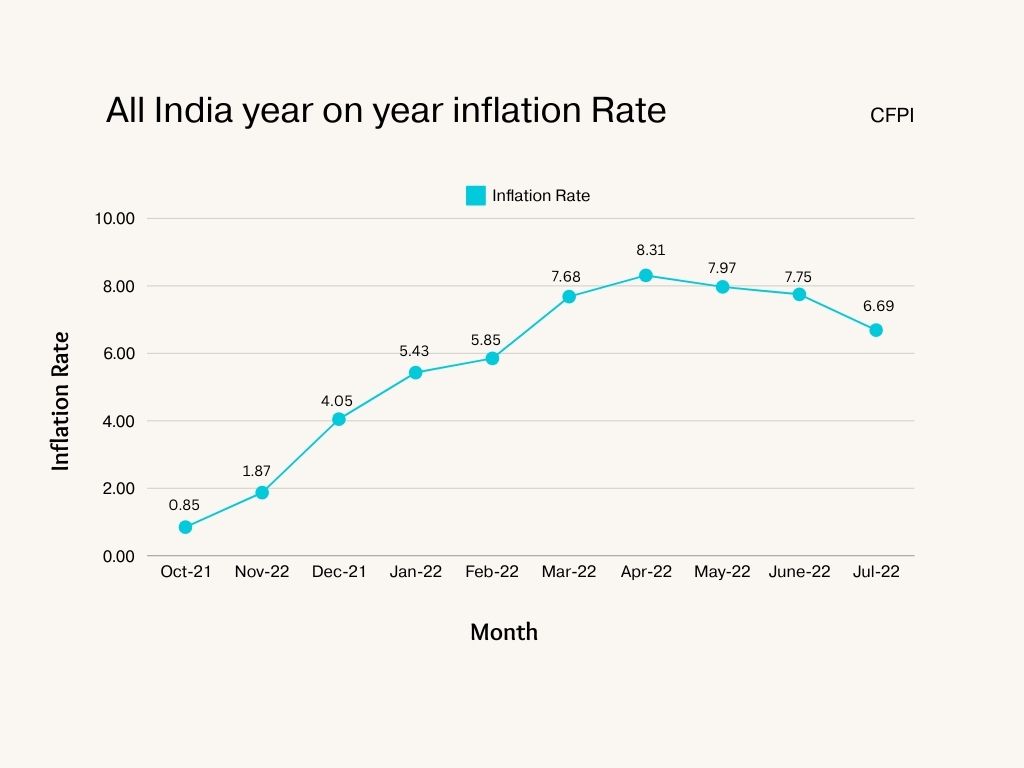


## Performance Metrics

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Model Used | Description | Time Taken (s) |
| Processing Image to JSON | Gemini Vision Pro | Time taken to convert the uploaded image to JSON format | 20.82 |
| Updating JSON | ChatGPT 3.5 | Time taken to update the JSON with new data | 0.00 |
| Plotting Updated JSON | N/A | Time taken to plot the updated JSON | 4.54 |

Date and Time: 2024-06-14 15:25:57

## Input Image: with\_label.jpg



Prompt Used:

Task: Extract detailed information from a given graph image and generate JSON files (data.json, layout.json, and config.json) compatible with Plotly. Use the specified delimiters to wrap each JSON section.  
  
Instructions:  
  
Graph Analysis:  
  
Identify and categorize the plot types present in the graph (e.g., line, scatter, bar, pie).  
Extract the data points for each plot (x-values, y-values, and any other relevant data such as labels for pie charts).  
Note any multiple plots on the same figure and their configurations.  
Generate data.json:  
  
For each plot, include details such as:  
type: The type of plot (e.g., scatter, bar, pie).  
x and y: Arrays of data points for the x and y axes.  
labels and values for pie charts.  
mode: For line and scatter plots (e.g., markers, lines, or markers+lines).  
name: Legend entry for the plot.  
marker: Properties such as color, size, and symbol for markers.  
line: Properties such as color, width, and dash style for lines.  
text: Hover text for each data point.  
hoverinfo: Information displayed on hover (e.g., x+y+text).  
Generate layout.json:  
  
Include details such as:  
title: The title of the graph, including text, font properties (family, size, color).  
xaxis and yaxis: Configuration including title, range, showgrid, gridcolor, zeroline, zerolinecolor, showticklabels, tickangle, tickfont.  
legend: Properties including orientation, x, y, xanchor, font.  
margin: Values for l (left), r (right), b (bottom), t (top), pad.  
plot\_bgcolor and paper\_bgcolor: Background colors of the plot area and paper.  
Generate config.json:  
  
Include configuration settings such as:  
responsive: Whether the graph is responsive (boolean).  
displayModeBar: Whether the mode bar is displayed (boolean).  
modeBarButtonsToRemove: List of mode bar buttons to remove (e.g., ["toImage"]).  
scrollZoom: Whether scrolling zoom is enabled (boolean).  
Output Format:  
  
Wrap each JSON output in specific XML-like tags:  
<data> ... </data>  
<layout> ... </layout>  
<config> ... </config>  
Make sure to follow the JSON formatting guidelines:  
Double Quotes for Keys and Strings: Ensure that all keys and string values in JSON are enclosed in double quotes (").  
Example: Instead of "type": bar, it should be "type": "bar".  
No Trailing Commas: Remove any trailing commas after the last element in arrays ([]) or objects ({}).  
Example: Instead of "color": "rgba(55, 128, 191, 0.6)",, it should be "color": "rgba(55, 128, 191, 0.6)"  
Example Output:  
  
<data>  
[  
 {  
 "type": "scatter",  
 "x": [1, 2, 3],  
 "y": [4, 5, 6],  
 "mode": "markers+lines",  
 "name": "Line Plot",  
 "marker": {  
 "color": "rgba(75, 192, 192, 0.6)",  
 "size": 8  
 },  
 "line": {  
 "color": "rgba(75, 192, 192, 1)",  
 "width": 2  
 }  
 }  
 // Additional plots if present  
]  
</data>  
<layout>  
{  
 "title": {  
 "text": "Graph Title",  
 "font": {  
 "family": "Arial, sans-serif",  
 "size": 24,  
 "color": "#000000"  
 }  
 },  
 "xaxis": {  
 "title": {  
 "text": "X Axis",  
 "font": {  
 "family": "Arial, sans-serif",  
 "size": 18,  
 "color": "#000000"  
 }  
 },  
 "showgrid": true,  
 "gridcolor": "rgba(0, 0, 0, 0.1)",  
 "zeroline": true,  
 "zerolinecolor": "rgba(0, 0, 0, 0.1)"  
 },  
 "yaxis": {  
 "title": {  
 "text": "Y Axis",  
 "font": {  
 "family": "Arial, sans-serif",  
 "size": 18,  
 "color": "#000000"  
 }  
 },  
 "showgrid": true,  
 "gridcolor": "rgba(0, 0, 0, 0.1)",  
 "zeroline": true,  
 "zerolinecolor": "rgba(0, 0, 0, 0.1)"  
 },  
 "legend": {  
 "orientation": "h",  
 "x": 0.5,  
 "xanchor": "center",  
 "y": -0.2,  
 "font": {  
 "family": "Arial, sans-serif",  
 "size": 12,  
 "color": "#000000"  
 }  
 },  
 "margin": {  
 "l": 60,  
 "r": 30,  
 "b": 60,  
 "t": 60  
 },  
 "plot\_bgcolor": "#ffffff",  
 "paper\_bgcolor": "#ffffff"  
}  
</layout>  
<config>  
{  
 "responsive": true,  
 "displayModeBar": true,  
 "modeBarButtonsToRemove": ["toImage"],  
 "scrollZoom": true  
}  
</config>

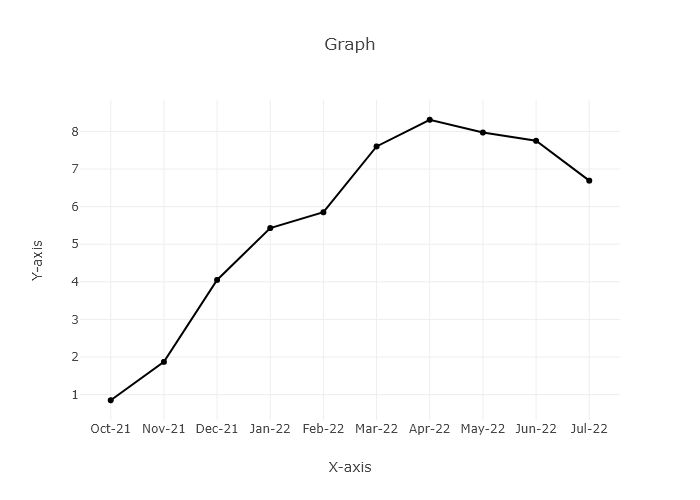
Original JSON:

{  
 "metadata": {  
 "title": "Inflation Rate",  
 "xaxis": {  
 "title": "Month"  
 },  
 "yaxis": {  
 "title": "Inflation Rate"  
 }  
 },  
 "data": [  
 {  
 "name": "Inflation Rate",  
 "x": [  
 "Oct-21",  
 "Nov-21",  
 "Dec-21",  
 "Jan-22",  
 "Feb-22",  
 "Mar-22",  
 "Apr-22",  
 "May-22",  
 "Jun-22",  
 "Jul-22"  
 ],  
 "y": [  
 0.85,  
 1.87,  
 4.05,  
 5.43,  
 5.85,  
 7.6,  
 8.31,  
 7.97,  
 7.75,  
 6.69  
 ]  
 }  
 ]  
}

Updated JSON:

{  
 "metadata": {  
 "title": "Inflation Rate",  
 "xaxis": {  
 "title": "Month"  
 },  
 "yaxis": {  
 "title": "Inflation Rate"  
 }  
 },  
 "data": [  
 {  
 "name": "Inflation Rate",  
 "x": [  
 "Oct-21",  
 "Nov-21",  
 "Dec-21",  
 "Jan-22",  
 "Feb-22",  
 "Mar-22",  
 "Apr-22",  
 "May-22",  
 "Jun-22",  
 "Jul-22"  
 ],  
 "y": [  
 0.85,  
 1.87,  
 4.05,  
 5.43,  
 5.85,  
 7.6,  
 8.31,  
 7.97,  
 7.75,  
 6.69  
 ]  
 }  
 ]  
}

Updated Chart:



## Performance Metrics

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Model Used | Description | Time Taken (s) |
| Processing Image to JSON | Gemini Vision Pro | Time taken to convert the uploaded image to JSON format | 13.88 |
| Updating JSON | ChatGPT 3.5 | Time taken to update the JSON with new data | 0.00 |
| Plotting Updated JSON | N/A | Time taken to plot the updated JSON | 5.51 |