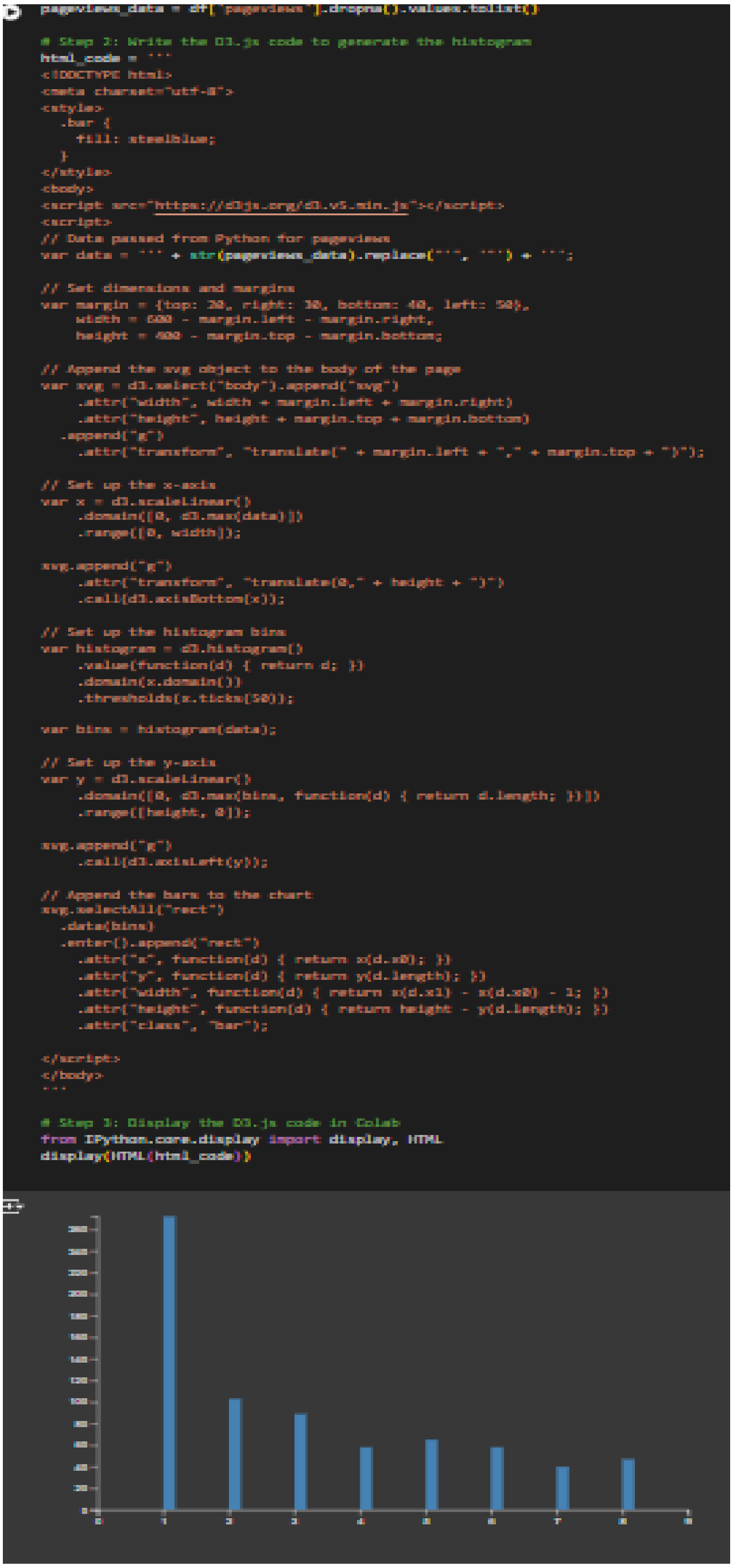
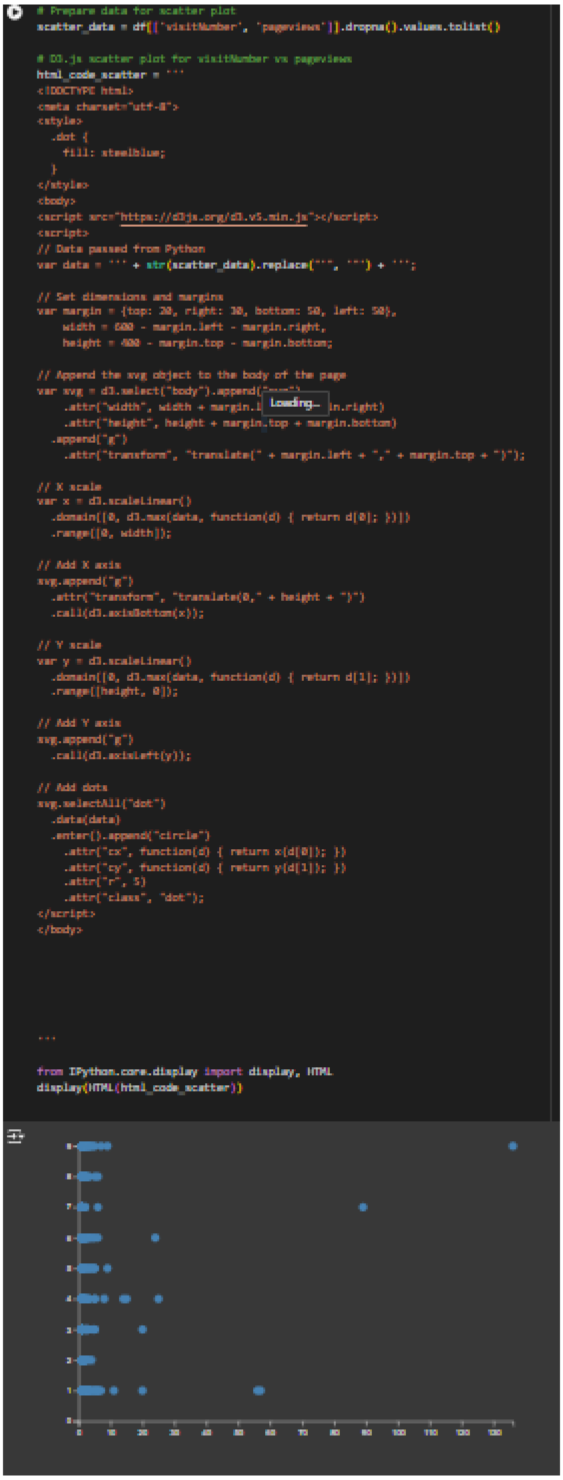
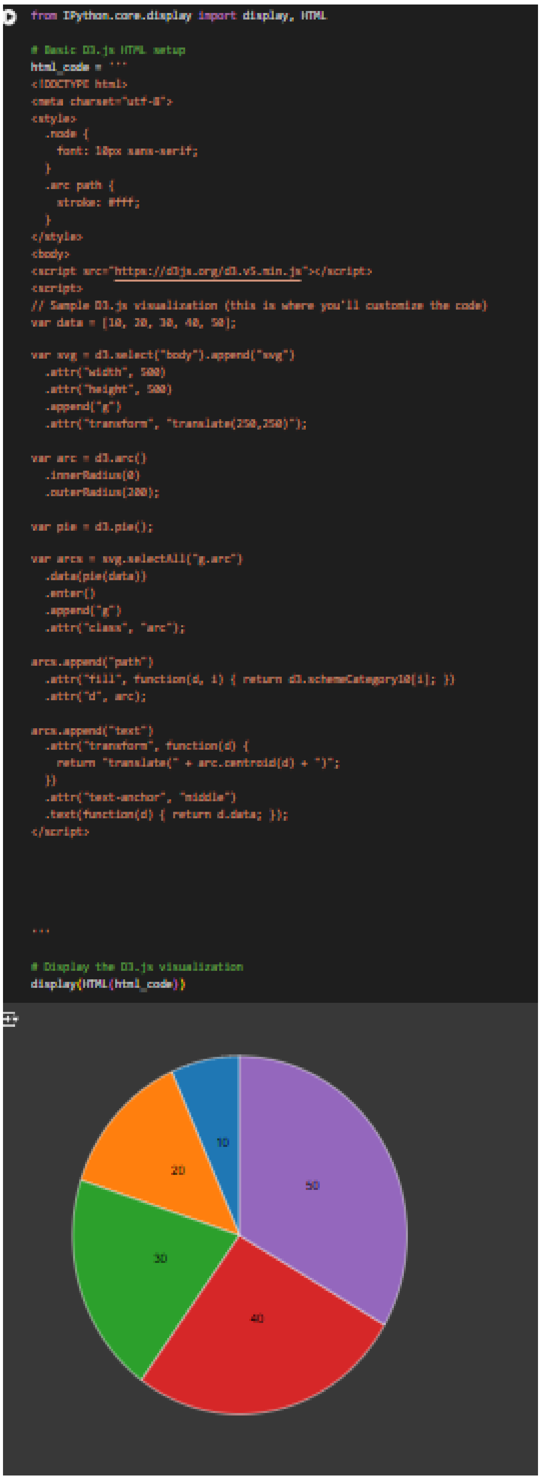
D3.js Visualizations



```
pageviews_data = df['pageviews'].dropna().values.tolist()

# Step 2: Write the D3.js code to generate the histogram
html_code = """
<!DOCTYPE html>
<meta charset="utf-8">
<style>
  .bar {
    fill: steelblue;
  }
</style>
<body>
<script src="https://d3js.org/d3.v5.min.js"></script>
<script>
// Data passed from Python for pageviews
var data = """ + str(pageviews_data).replace("'", '"') + """;

// Set dimensions and margins
var margin = {top: 20, right: 30, bottom: 40, left: 50},
    width = 800 - margin.left - margin.right,
    height = 400 - margin.top - margin.bottom;

// Append the svg object to the body of the page
var svg = d3.select("body").append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
  .append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

// Set up the x-axis
var x = d3.scaleLinear()
    .domain([0, d3.max(data)])
    .range([0, width]);

svg.append("g")
    .attr("transform", "translate(0," + height + ")")
    .call(d3.axisBottom(x));

// Set up the histogram bins
var histogram = d3.histogram()
    .value(function(d) { return d; })
    .domain(x.domain())
    .thresholds(x.ticks(50));

var bins = histogram(data);

// Set up the y-axis
var y = d3.scaleLinear()
    .domain([0, d3.max(bins, function(d) { return d.length; })])
    .range([height, 0]);

svg.append("g")
    .call(d3.axisLeft(y));

// Append the bars to the chart
svg.selectAll("rect")
  .data(bins)
  .enter().append("rect")
    .attr("x", function(d) { return x(d.x0); })
    .attr("y", function(d) { return y(d.length); })
    .attr("width", function(d) { return x(d.x1) - x(d.x0) - 1; })
    .attr("height", function(d) { return height - y(d.length); })
    .attr("class", "bar");

</script>
</body>
"""

# Step 3: Display the D3.js code in Colab
from IPython.core.display import display, HTML
display(HTML(html_code))
```



```
# Prepare data for scatter plot
scatter_data = df[['visitNumber', 'pageviews']].dropna().values.tolist()

# D3.js scatter plot for visitNumber vs pageviews
html_code_scatter = '''
<!DOCTYPE html>
<meta charset="utf-8">
<style>
  .dot {
    fill: steelblue;
  }
</style>
<body>
<script src="https://d3js.org/d3.v5.min.js"></script>
<script>
// Data passed from Python
var data = ''' + str(scatter_data).replace("'", '"') + ''';

// Set dimensions and margins
var margin = {top: 20, right: 30, bottom: 50, left: 50},
    width = 600 - margin.left - margin.right,
    height = 400 - margin.top - margin.bottom;

// Append the svg object to the body of the page
var svg = d3.select("body").append("svg")
    .attr("width", width + margin.l Loading… in.right)
    .attr("height", height + margin.top + margin.bottom)
  .append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

// X scale
var x = d3.scaleLinear()
  .domain([0, d3.max(data, function(d) { return d[0]; })])
  .range([0, width]);

// Add X axis
svg.append("g")
  .attr("transform", "translate(0," + height + ")")
  .call(d3.axisBottom(x));

// Y scale
var y = d3.scaleLinear()
  .domain([0, d3.max(data, function(d) { return d[1]; })])
  .range([height, 0]);

// Add Y axis
svg.append("g")
  .call(d3.axisLeft(y));

// Add dots
svg.selectAll("dot")
  .data(data)
  .enter().append("circle")
    .attr("cx", function(d) { return x(d[0]); })
    .attr("cy", function(d) { return y(d[1]); })
    .attr("r", 5)
    .attr("class", "dot");
</script>
</body>




'''

from IPython.core.display import display, HTML
display(HTML(html_code_scatter))
```



```
from IPython.core.display import display, HTML

# Basic D3.js HTML setup
html_code = '''
<!DOCTYPE html>
<meta charset="utf-8">
<style>
  .node {
    font: 10px sans-serif;
  }
  .arc path {
    stroke: #fff;
  }
</style>
<body>
<script src="https://d3js.org/d3.v5.min.js"></script>
<script>
// Sample D3.js visualization (this is where you'll customize the code)
var data = [10, 20, 30, 40, 50];

var svg = d3.select("body").append("svg")
  .attr("width", 500)
  .attr("height", 500)
  .append("g")
  .attr("transform", "translate(250,250)");

var arc = d3.arc()
  .innerRadius(0)
  .outerRadius(200);

var pie = d3.pie();

var arcs = svg.selectAll("g.arc")
  .data(pie(data))
  .enter()
  .append("g")
  .attr("class", "arc");

arcs.append("path")
  .attr("fill", function(d, i) { return d3.schemeCategory10[i]; })
  .attr("d", arc);

arcs.append("text")
  .attr("transform", function(d) {
    return "translate(" + arc.centroid(d) + ")";
  })
  .attr("text-anchor", "middle")
  .text(function(d) { return d.data; });
</script>




'''

# Display the D3.js visualization
display(HTML(html_code))
```