



EECS E6893 Big Data Analytics

HW3: Data visualization

Tutorial II

Apurva Patel
amp2365@columbia.edu

Agenda

- Using D3.js to do data visualization and directed graph visualization
- Introduction of Apache HTTP Server

Graphical Analysis using D3

- Network analysis and network visualization are more common now with the growth of online social networks like Twitter and Facebook, as well as social media and linked data, all of which are commonly represented with network structures.
- In general, when dealing with networks you refer to the things being connected (like people) as nodes and the connections between them (such as being a friend on Facebook) as edges or links.
- Networks may also be referred to as graphs, because that's what they're called in mathematics.



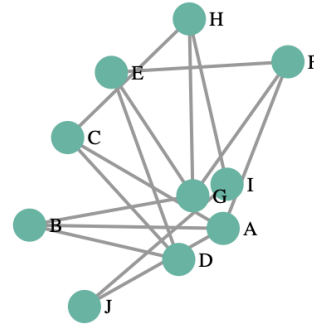
Graphical Analysis using D3

- D3 is particularly adept at creating Graphical visualizations due to its data-binding capability and built-in force simulations
- Real-world applications:
 - Social network analysis – friendships, interactions
 - Biological networks – protein interaction, genetic pathways
 - Infrastructure – internet routing, power grids

Force-directed Network diagrams and simulations in D3

- Automatically positions nodes
- Nodes repel each other, links act as springs
- Custom forces can be added
 - Centering, collision detection, etc.
- The force layout dynamically updates the positions of its elements to find the best fit, it does it continuously in real-time rather than as a preprocessing step before rendering

Force- directed diagram example



```

t2 > <> force.html > html > body > script > simulation.on("tick") callback
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Force-Directed Graph with D3.js</title>
6   <script src="https://d3js.org/d3.v5.min.js"></script>
7 </head>
8 <body>
9   <svg width="800" height="600"></svg>
10  <script>
11    // Sample data
12  const nodes = [
13    { id: "A" },
14    { id: "B" },
15    { id: "C" },
16    { id: "D" },
17    { id: "E" },
18    { id: "F" },
19    { id: "G" },
20    { id: "H" },
21    { id: "I" },
22    { id: "J" }
23  ];
24
25  const links = [
26    { source: "A", target: "B" },
27    { source: "A", target: "C" },
28    { source: "B", target: "D" },
29    { source: "C", target: "D" },
30    { source: "D", target: "E" },
31    { source: "E", target: "F" },
32    { source: "F", target: "G" },
33    { source: "G", target: "H" },
34    { source: "H", target: "I" },
35    { source: "I", target: "J" },
36    { source: "J", target: "A" },
37    { source: "E", target: "G" },
38    { source: "C", target: "H" },
39    { source: "A", target: "F" },
40    { source: "B", target: "G" }
41  ];
42
43  const svg = d3.select("svg");
44  const width = +svg.attr("width");
45  const height = +svg.attr("height");
46

```

```

47  const simulation = d3.forceSimulation(nodes)
48    .force("link", d3.forceLink(links).id(d => d.id).distance(100))
49    .force("charge", d3.forceManyBody())
50    .force("center", d3.forceCenter(width / 2, height / 2));
51
52  const link = svg.append("g")
53    .selectAll("line")
54    .data(links)
55    .enter().append("line")
56    .attr("stroke", "#999")
57    .attr("stroke-width", 2);
58
59  const node = svg.append("g")
60    .selectAll("circle")
61    .data(nodes)
62    .enter().append("circle")
63    .attr("r", 10)
64    .attr("fill", "#69b3a2")
65    .call(d3.drag())
66    .on("start", dragstarted)
67    .on("drag", dragged)
68    .on("end", dragended);
69
70  const label = svg.append("g")
71    .selectAll("text")
72    .data(nodes)
73    .enter().append("text")
74    .attr("font-size", 12)
75    .attr("dx", 12)
76    .attr("dy", ".35em")
77    .text(d => d.id);
78
79  simulation.on("tick", () => {
80    link.attr("x1", d => d.source.x)
81      .attr("y1", d => d.source.y)
82      .attr("x2", d => d.target.x)
83      .attr("y2", d => d.target.y);
84
85    node.attr("cx", d => d.x)
86      .attr("cy", d => d.y);
87
88    label.attr("x", d => d.x)
89      .attr("y", d => d.y);
90  });

```

```
91
92 function dragstarted(d) {
93     if (!d3.event.active) simulation.alphaTarget(0.3).restart();
94     d.fx = d.x;
95     d.fy = d.y;
96 }
97
98 function dragged(d) {
99     d.fx = d3.event.x;
100    d.fy = d3.event.y;
101 }
102
103 function dragended(d) {
104     if (!d3.event.active) simulation.alphaTarget(0);
105     d.fx = null;
106     d.fy = null;
107 }
108
109
110     </script>
111 </body>
112 </html>
```


Webpages on Apache Webserver

Hosting webpage on Apache web server using virtual host (MAC users)

Step 1: Install xcode

```
(base) srividyaianampudi@Srividya-Air ~ % xcode-select --install
xcode-select: error: command line tools are already installed, use "Software Update" to install updates
(base) srividyaianampudi@Srividya-Air ~ % █
```

Step 2: Install Homebrew

Go to brew.sh in your browser and copy the command there to your terminal-
`/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"`

```
(base) srividyaianampudi@Srividya-Air ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
=> Checking for `sudo` access (which may request your password)...
=> This script will install:
/opt/homebrew/bin/brew
/opt/homebrew/share/doc/homebrew
/opt/homebrew/share/man/man1/brew.1
/opt/homebrew/share/zsh/site-functions/_brew
/opt/homebrew/etc/bash_completion.d/brew
/opt/homebrew
```

⇒ Installation successful!

⇒ Homebrew has enabled anonymous aggregate formulae and cask analytics.

Read the analytics documentation (and how to opt-out) here:

<https://docs.brew.sh/Analytics>

No analytics data has been sent yet (nor will any be during this `install` run).

⇒ Homebrew is run entirely by unpaid volunteers. Please consider donating:

<https://github.com/Homebrew/brew#donations>

⇒ Next steps:

- Run these three commands in your terminal to add Homebrew to your **PATH**:

```
echo '# Set PATH, MANPATH, etc., for Homebrew.' >> /Users/srividyanampudi/.zprofile
echo 'eval "$(curl -sL https://raw.githubusercontent.com/Homebrew/brew/master/bin/brew)"' >> /Users/srividyanampudi/.zprofile
eval "$(curl -sL https://raw.githubusercontent.com/Homebrew/brew/master/bin/brew)"
```

- Run **brew help** to get started

- Further documentation:

<https://docs.brew.sh>

Step 3: Add Homebrew to your PATH

Follow instructions after installation

```
(base) srividyanampudi@Srividyas-Air ~ % echo '# Set PATH, MANPATH, etc., for Homebrew.' >> /Users/srividyanampudi/.zprofile
(base) srividyanampudi@Srividyas-Air ~ % echo 'eval "$(curl -sL https://raw.githubusercontent.com/Homebrew/brew/master/bin/brew)"' >> /Users/srividyanampudi/.zprofile
(base) srividyanampudi@Srividyas-Air ~ % eval "$(curl -sL https://raw.githubusercontent.com/Homebrew/brew/master/bin/brew)"
(base) srividyanampudi@Srividyas-Air ~ % █
```

Step 4: Install apache2

Command - brew install apache2

```
(base) srividyanampudi@Srividyas-Air ~ % brew install apache2
=> Downloading https://ghcr.io/v2/homebrew/core/apr/manifests/1.7.0_3
##### 100.0%
=> Downloading https://ghcr.io/v2/homebrew/core/apr/blobs/sha256:02e6b44b3284fa471cce15592a8666
=> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:02e6b44b328
##### 100.0%
```

Step 5: Start apache server

```
(base) srividyanampudi@Srividyas-Air ~ % sudo apachectl start
(base) srividyanampudi@Srividyas-Air ~ % █
```

This will start Apache HTTP server which can be tested by visiting localhost on the browser. The localhost 8080 gives the response as shown below:

A screenshot of a web browser's address bar. It shows navigation icons (back, forward, refresh) on the left and the text 'localhost:8080' on the right, indicating the current page is being accessed from the local machine on port 8080.

It works!

Step 6: Open httpd config file

- If you're using Intel-based Mac: `vim /usr/local/etc/httpd/httpd.conf`
- If you're using Mac with Apple Silicon: `vim /opt/homebrew/etc/httpd/httpd.conf`

Step 7: Update these lines

Listen 8080 to Listen 80

DocumentRoot "/usr/local/var/www" to DocumentRoot "/Users/your_account/Sites"
<Directory "/usr/local/var/www"> to <Directory "/Users/your_account/Sites">

AllowOverride None to AllowOverride All

```
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 12.34.56.78:80
Listen 80
```

```
#
DocumentRoot "/Users/srividyanampudi/Sites"
<Directory "/Users/srividyanampudi/Sites">
#
# Possible values for the Options directive are "None", "All",
# or any combination of:
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI MultiViews
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
# The Options directive is both complicated and important. Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
#
Options Indexes FollowSymLinks

#
# AllowOverride controls what directives may be placed in .htaccess files.
# It can be "All", "None", or any combination of the keywords:
#   AllowOverride FileInfo AuthConfig Limit
#
AllowOverride All

#
# Controls who can get stuff from this server.
#
```

Step 8: Uncomment this line

LoadModule rewrite_module lib/httpd/modules/mod_rewrite.so

Update these lines

User _www to User your_account

Group _www to Group staff

```
LoadModule alias_module lib/httpd/modules/mod_alias.so
LoadModule rewrite_module lib/httpd/modules/mod_rewrite.so

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User srividyaianampudi
Group staff
```

Step 9: Update this line

ServerName www.example.com:8080 to ServerName localhost

```
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
```

Step 10: Create Sites folder and add your html file to the folder

```
(base) srividyanampudi@Srividyas-Air ~ % mkdir Sites
(base) srividyanampudi@Srividyas-Air ~ %
(base) srividyanampudi@Srividyas-Air ~ %
(base) srividyanampudi@Srividyas-Air ~ % cd Sites
(base) srividyanampudi@Srividyas-Air Sites % ls
jsexample.html
```

Step 11. Restart apache

```
(base) srividyanampudi@Srividyas-Air Sites % sudo apachectl stop
(base) srividyanampudi@Srividyas-Air Sites %
(base) srividyanampudi@Srividyas-Air Sites %
(base) srividyanampudi@Srividyas-Air Sites %
(base) srividyanampudi@Srividyas-Air Sites % sudo apachectl start
```


Step 12: Test localhost to see your hosted website

<http://localhost>

A screenshot of a web browser's address bar. It features a back arrow, a forward arrow, and a refresh icon on the left. The address bar itself is highlighted with a blue border and contains the text 'localhost' preceded by an information icon (i).

Sample HTML to host on apache webserver

- Car
 - Ford

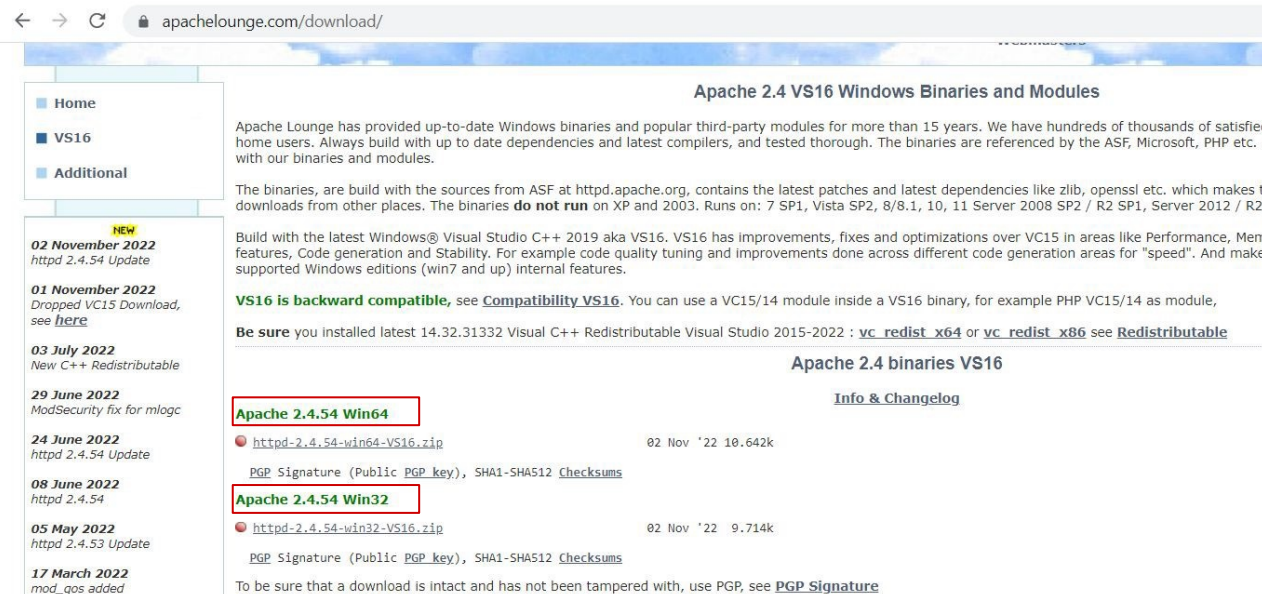
Step 13: Stop apache

```
(base) srividyaianampudi@Srividyas-Air Sites % sudo apachectl stop
Password:
(base) srividyaianampudi@Srividyas-Air Sites %
```

Hosting webpage on Apache web server using virtual host (Windows users)

Step 1: Download Apache lounge zip file -

<https://www.apachelounge.com/download/>



The screenshot shows a web browser window with the address bar displaying "apachelounge.com/download/". The page title is "Apache 2.4 VS16 Windows Binaries and Modules". The main content area contains the following text:

Apache Lounge has provided up-to-date Windows binaries and popular third-party modules for more than 15 years. We have hundreds of thousands of satisfied home users. Always build with up to date dependencies and latest compilers, and tested thorough. The binaries are referenced by the ASF, Microsoft, PHP etc. a with our binaries and modules.

The binaries, are build with the sources from ASF at httpd.apache.org, contains the latest patches and latest dependencies like zlib, openssl etc. which makes it downloads from other places. The binaries **do not run** on XP and 2003. Runs on: 7 SP1, Vista SP2, 8/8.1, 10, 11 Server 2008 SP2 / R2 SP1, Server 2012 / R2,

Build with the latest Windows® Visual Studio C++ 2019 aka VS16. VS16 has improvements, fixes and optimizations over VC15 in areas like Performance, Memory features, Code generation and Stability. For example code quality tuning and improvements done across different code generation areas for "speed". And makes supported Windows editions (win7 and up) internal features.

VS16 is backward compatible, see [Compatibility VS16](#). You can use a VC15/14 module inside a VS16 binary, for example PHP VC15/14 as module,

Be sure you installed latest 14.32.31332 Visual C++ Redistributable Visual Studio 2015-2022 : [vc_redist_x64](#) or [vc_redist_x86](#) see [Redistributable](#)

Apache 2.4 binaries VS16

[Info & Changelog](#)

Apache 2.4.54 Win64	
httpd-2.4.54-win64-VS16.zip	02 Nov '22 10.642k
PGP Signature (Public PGP key), SHA1-SHA512 Checksums	
Apache 2.4.54 Win32	
httpd-2.4.54-win32-VS16.zip	02 Nov '22 9.714k
PGP Signature (Public PGP key), SHA1-SHA512 Checksums	

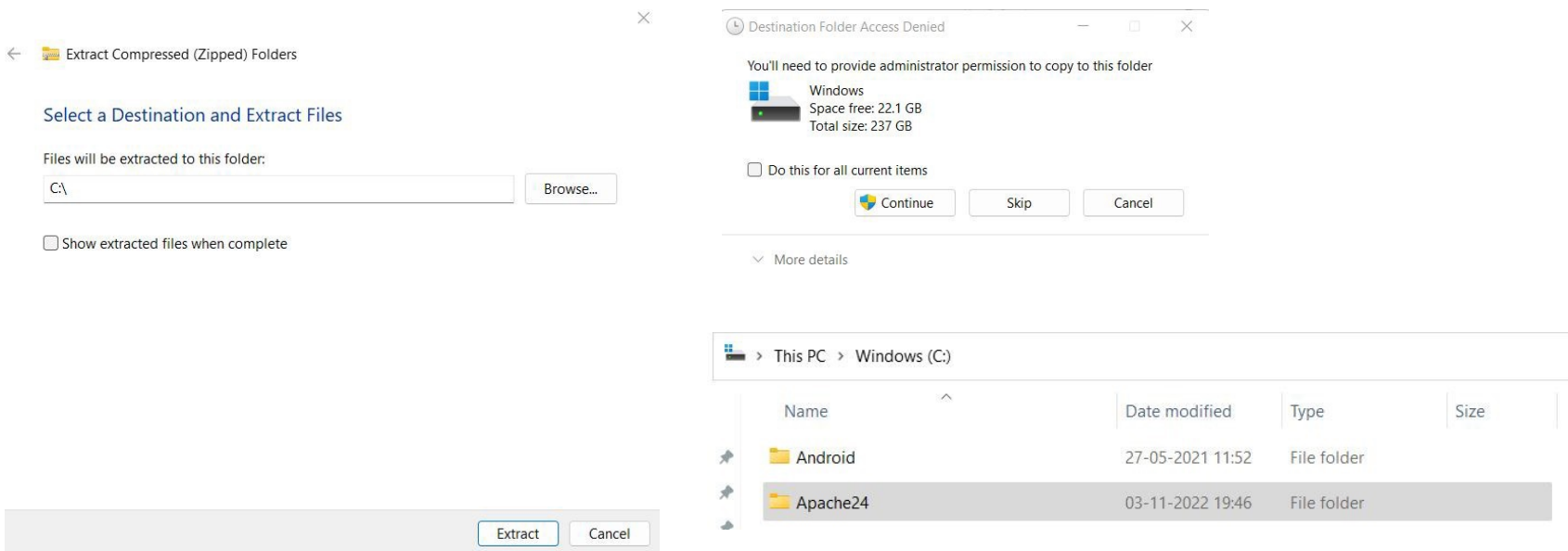
To be sure that a download is intact and has not been tampered with, use PGP, see [PGP Signature](#)

The left sidebar contains a navigation menu with "Home", "VS16", and "Additional". Below the menu is a "NEW" badge and a list of recent updates:

- 02 November 2022**
httpd 2.4.54 Update
- 01 November 2022**
Dropped VC15 Download, see [here](#)
- 03 July 2022**
New C++ Redistributable
- 29 June 2022**
ModSecurity fix for mlogc
- 24 June 2022**
httpd 2.4.54 Update
- 08 June 2022**
httpd 2.4.54
- 05 May 2022**
httpd 2.4.53 Update
- 17 March 2022**
mod_qos added

Download 64bit if your system is 64bit else download 32bit zip file

Step 2: Once zip file is finished downloading unzip it to C drive. Click Yes for administrative privileges.



Step 3: Go to **"C:\Apache24\bin\"** and find httpd and we need to install this httpd. This is what is actually going to run apache. So we will go to start menu and type cmd to get command prompt. Right click as select **Run as administrator**. Click yes if it asks you for permission.

Step 4: Change directory to **C:\Apache24\bin** by typing **cd C:\Apache24\bin** in the terminal

Step 5: Now that we are in bin directory we will install httpd by typing the command **httpd -k install**. Allow access

> This PC > Windows (C:) > Apache24 > bin

Name	Date modified	Type	Size
iconv	03-11-2022 19:46	File folder	
ab	03-11-2022 19:46	Application	97 KB
abs	03-11-2022 19:46	Application	108 KB
ApacheMonitor	03-11-2022 19:46	Application	42 KB
apr_crypto_openssl-1.dll	03-11-2022 19:46	Application extens...	19 KB
apr_dbd_odbc-1.dll	03-11-2022 19:46	Application extens...	31 KB
apr_idap-1.dll	03-11-2022 19:46	Application extens...	15 KB
dbmmanage.pl	03-11-2022 19:46	PL File	9 KB
htcacheclean	03-11-2022 19:46	Application	100 KB
htdbm	03-11-2022 19:46	Application	121 KB
htdigest	03-11-2022 19:46	Application	84 KB
htpasswd	03-11-2022 19:46	Application	117 KB
httpd	03-11-2022 19:46	Application	30 KB

cmd

All Apps Documents Web More

Best match

Command Prompt App

Apps

- Git CMD (Deprecated)
- Anaconda Prompt
- mkTO.cmd
- Google Cloud SDK Shell

Search the web

cmd - See web results

Command Prompt App

- Open
- Run as administrator
- Open file location
- Pin to Start
- Pin to taskbar

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Apache24\bin

C:\Apache24\bin>
```

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.

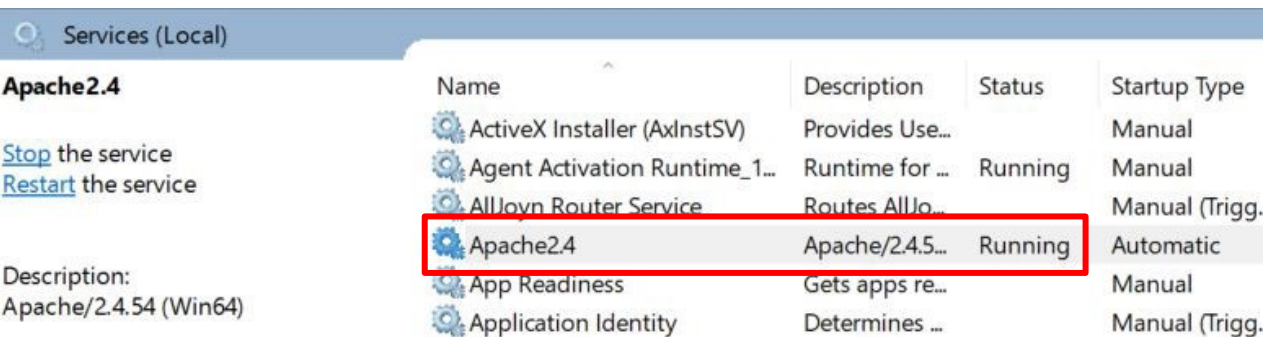
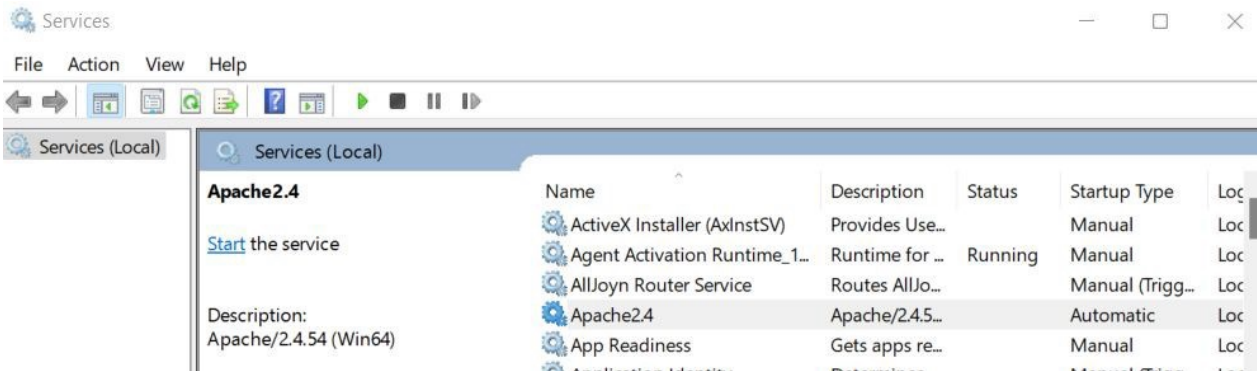
C:\WINDOWS\system32>cd C:\Apache24\bin

C:\Apache24\bin>httpd -k install
Installing the 'Apache2.4' service
The 'Apache2.4' service is successfully installed.
Testing httpd.conf...
Errors reported here must be corrected before the service can be started.

C:\Apache24\bin>
```

Step 6: Now Apache is installed as a service. If we go to Windows and type in Services in search bar and open Services we can see that Apache2.4 is installed.

Step 7: Now we will start the APache service by clicking start on left side. And then it will say that it is running. We can stop or restart the service on the left side if we need to.



Step 8: Now to check if it is running. We will go to 127.0.0.1 in your browser it shows that it works!



Step 9: If you have any errors you can check them in the logs folder in **C:\Apache24\logs** in errors.txt file

Step 10: Now to host our webpage we will go to **C:\Apache24\htdocs** and change the index.html to be your required html file and then again open 127.0.0.1 to see our webpage.

I have hosted the basic histogram html file for example.

```

EXPLORER
  X < index.html
  HTDOCS
    index.html

index.html X
  index.html > script
1 <!DOCTYPE html>
2 <meta charset="utf-8">
3
4 <!-- Loading v4 d3.js -->
5 <script src="https://d3js.org/d3.v4.js"></script>
6
7 <!-- Creating a div where the graph will be plotted -->
8 <div id="histogram"></div>
9
10 <script>
11 // setting the dimensions and margins of the graph
12 var margin = {top: 10, right: 30, bottom: 30, left: 40},
13   width = 400 - margin.left - margin.right,
14   height = 400 - margin.top - margin.bottom;
15
16 // appending the svg object to the body of the page
17 var svg = d3.select("#histogram")
18   .append("svg")
19   .attr("width", width + margin.left + margin.right)
20   .attr("height", height + margin.top + margin.bottom)
21   .append("g")
22   .attr("transform",
23     "translate(" + margin.left + "," + margin.top + ")");
24
25 // getting the data in csv format
26 d3.csv("https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_dataset/1_OneNum.csv", function(data) {
27
28   // X axis: scale and draw:
29   var x = d3.scaleLinear()
30     .domain([0,1000])
31     // .domain([0, d3.max(data, function(d) { return +d.price })])
32     // this will set the scale of x from 0 to max of the price column
33     .range([0, width]);
34   svg.append("g")
35     .attr("transform", "translate(0," + height + ")")
36     .call(d3.axisBottom(x));
37
38   // setting the parameters for the histogram
39   var histogram = d3.histogram()
40     .value(function(d) { return d.price; }) // giving a vector of value
41     .domain(x.domain()) // then setting the domain of the graphic
42     .thresholds(x.ticks(30)); // then the numbers of bins

```

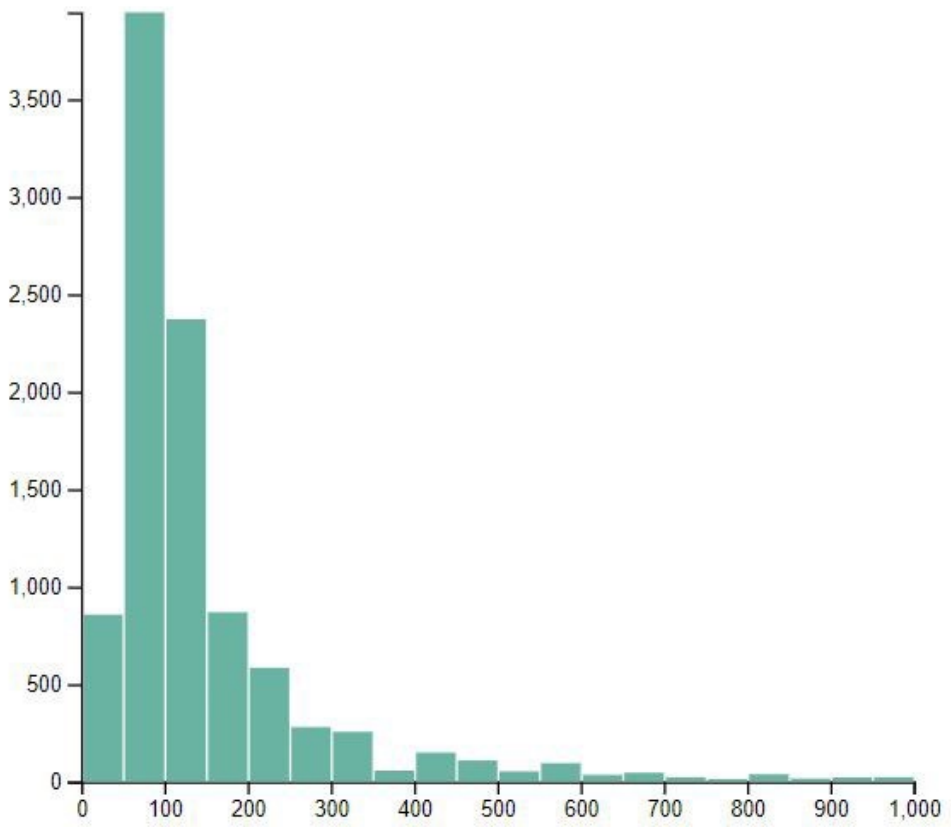
```

var bins = histogram(data);

// Y axis: scale and draw:
var y = d3.scaleLinear()
  .range([height, 0]);
  y.domain([0, d3.max(bins, function(d) { return d.length; })]);
svg.append("g")
  .call(d3.axisLeft(y));

// append the bar rectangles to the svg element
svg.selectAll("rect")
  .data(bins)
  .enter()
  .append("rect")
  .attr("x", 1)
  .attr("transform", function(d) { return "translate(" + x(d.x0) + "," + y(d.length) + ")"; })
  .attr("width", function(d) { return x(d.x1) - x(d.x0) - 1; })
  .attr("height", function(d) { return height - y(d.length); })
  .style("fill", "#69b3a2");
});
</script>

```



We can see that our d3 html file is hosted on Apache web server when we go to 127.0.0.1 on browser

Explore more examples

- <https://d3js.org/>
- <https://observablehq.com/@d3/gallery>
- <https://github.com/d3/d3>
- https://www.tutorialspoint.com/d3js/d3js_working_example.htm

THANK YOU