WFYI: Data Visualization in PowerBI

Team Leader: Izzy Austin

Maddie Neely, Nick Donnelly, Maddy Bloom,

Erin Penner, Nolan Knight, Nic Reilly

EPICS 1: Engineering Projects in Community Service

Dr. Panos Linos

December 15, 2022

Table of Contents

Abstract

During this semester, our group used PowerBI, which is Microsoft’s interactive data visualization tool, to design and implement a set of reports/dashboards, based on input from WFYI, to replace an old reporting tool that they were losing access to. First, we had to upload the data given to us by our client in a .csv file. Then, we had to spend some time carefully removing duplicate records and performing data validation to ensure the data being used in our reports is the same that was being used in their old reporting tool. Now, within the dashboards we created, our client is able to see the multi-year data in easy-to-use graphs and manipulate them to see the information over a specific time and date ranges as well as on specific days of the week, month, or even year.

Chapter 1: Introduction

WFYI came to us asking for help with creating data analytic reports because they would be losing access to the tool that they had previously been using. We did a lot of research on an application that would be a good replacement for the one that they would be losing access to. We ultimately decided to use PowerBI to create all of the reports and dashboards needed to fulfill the functional requirements. We split into two groups to tackle these reports, alongside producing step by step documentation of how we created everything in PowerBI so that it can be easily replicated by WFYI if need be. Throughout the semester we created Sprint presentations to present to the class to showcase the progress that we had been making, and give us a chance to receive feedback and help us determine what we should change moving forward. We then ended our semester by performing data validation and troubleshooting problems that arose with discrepancies in the data. We wanted to check and make sure that the data in our reports matched the ones that were being generated by our client's old tool. They did not match initially, but after playing around with duplicates we were able to get the data to match. Finally, we created a poster and final presentation showcasing all of our work.

Throughout this report you will find a description of the reports that we made along with the functional requirements and images of the reports. You will also see a deeper dive into the architecture of our project, the design and implementation of our reports, our efforts towards quality assurance, and our project organization. We also have included our progress reports and Sprint presentations.

Chapter 2: Requirements Specifications

Due to the sunsetting of WFYI's previous system, WFYI needed a way of storing and accessing its historic data from its old system. Additionally, WFYI wanted to recreate its past reports from its old system in a visually appealing way. WFYI did not have much room in its budget for this, so on top of finding a software or system that had the same functionality as its previous system, the new software or system needed to be free or at very low cost to WFYI.

One report that needed to be replicated in the new system was the Date Range Listener report. This report needed to show the amount of hosts accessing the stream, the total number of GBs sent (AKA total size of bandwidth), and the total seconds spent listening over a given date/time range. These metrics needed to be shown based on a day, week, or month. Historically, this was shown graphically via bar charts.

The second report that needed to be replicated was the Daypart Calculator report. This report needed to show a count of unique visitors (or people tuning in to WFYI), and the average duration each visitor spent listening during an input of a date range. The output should be viewed on a day-to-day basis, weekly basis, monthly basis, and yearly basis. Like the first report, historically this was shown graphically via bar charts as well as in a tabular form.

Finally, a necessary deliverable for this project was documentation showing how each report was created, and overall documentation to show how we used the new software to replicate the features of the previous software.

Here are the exact functional requirements that were given to us by our WFYI client Chris:

WFYI Live Radio Stream Data Dashboard

The data set for this reporting dashboard is the set of files in the sgreport-all-data.zip archive with the filename prefix "log_detail". Be careful not to import duplicate data from overlapping files, e.g. log_detail_Jan-2019.csv contains records also found in log_detail_2019.csv. You can use the columns Date/time and Hostname to identify 'unique' rows of data.

Report 1 – Date Range Listener Report

A report that lets the user pick a date range (begin and end) and a reporting interval for grouping the data within the selected date range.

Report outputs should consist of a table with these columns:

- Accesses – raw tally of hosts accessing the stream in a given date range and interval
- Visitors – tally of unique hostnames accessing the stream in a given date range and interval
- Size – total of GBs sent in a given date range and interval
- Duration – total seconds spent listening in a given date range and interval
- Average duration – the average of time spent listening divided by the number of Accesses in a given date range and interval

Report filtering criteria are:

- Date Range: Begin and end date – where begin date <= access date < end date. We will only be looking at whole days so begin date is midnight of the date and the end date is midnight of the next day. So with begin date = 2020-09-01 and end date = 2020-09-30 we include all records with: Access Date/time between 2020-09-01 00:00:00 and 2020-10-01 00:00:00 or 2020-09-01 00:00:00 <= Date/time < 2020-10-01 00:00:00
- Interval – Day, Week or Month as the only options.
    - Day – output one row for each day in the selected date range
    - Week – output one row for each week (M-Su) in the selected date range
        - When not a full week, tally the data for the days available and indicate on the output row with '*' – incomplete week
    - Month – Output one row for each month in the selected date range

Report 2 – Daypart Calculator

This report groups data by the day and hour for a given date range and outputs a single summary table for days of the week and the 24 hours within those days with data. This could be a line graph and/or individual charts for each day.

Report outputs should consist of a table with these columns:

- Visitors – tally of unique hostnames accessing the stream in a given date range and interval
- Average duration – the average of time spent listening divided by the number of Accesses in a given date range and interval

Report filtering criteria are:

- Date Range: Begin and end date – where begin date <= access date < end date. We will only be looking at whole days so begin date is midnight of the date and the end date is midnight of the next day. So with begin date = 2020-09-01 and end date = 2020-09-30 we include all records with: Access Date/time between 2020-09-01 00:00:00 and 2020-10-01 00:00:00 or 2020-09-01 00:00:00 <= Date/time < 2020-10-01 00:00:00

Chapter 3: Architecture

First, we were given the historical data from the sunsetting system in .csv files. One file included all daily data, with the date, number of accesses, visitors, size, duration of time spent listening, and average duration of time spent listening. This file was around 1,350 rows. The next five files, the Log Detail Data csv files, included columns such as date, page, hostname, referrer, authenticated user, server response, whether or not the user listened for more than 15 minutes, accesses, size, and duration. These files totaled to over 20,000,000 rows.

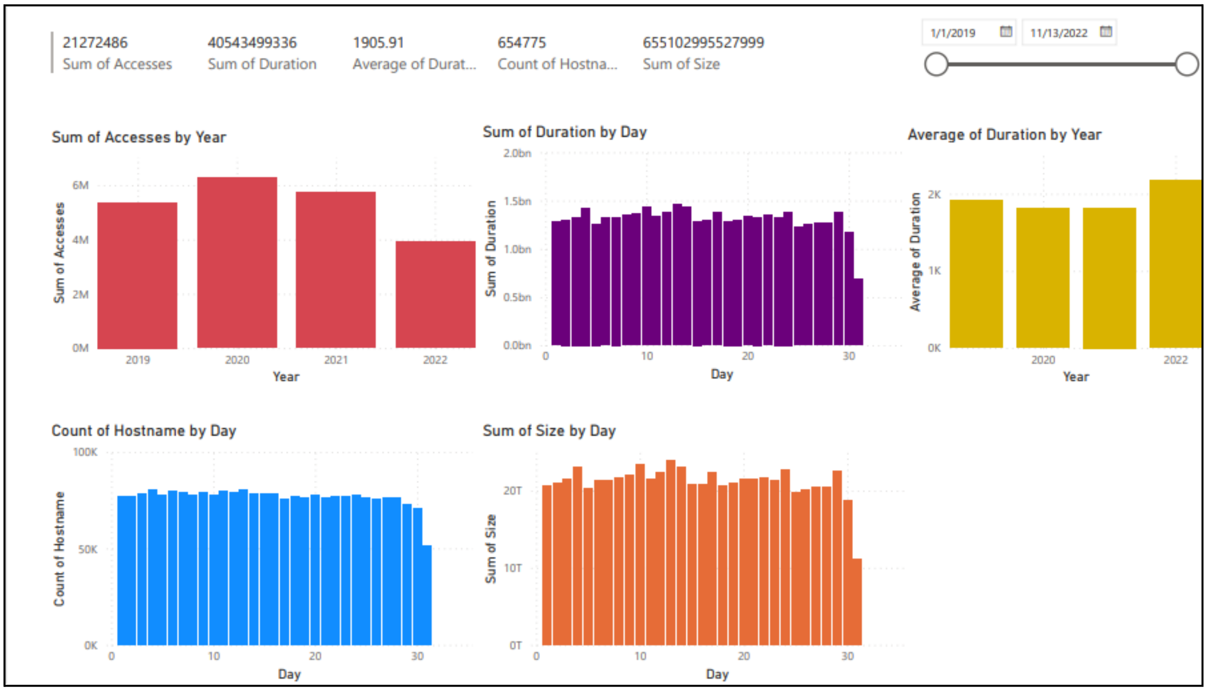
We then put these files into PowerBI, utilizing the PowerBI Desktop version. Once the data was cleaned in the Desktop version, as well as appended all together so that it could be looked at as a whole instead of just by separate years, the newly created dataset was uploaded to the PowerBI web version. In the web version, it's possible to collaborate with one another on the reports, as well as easily share the reports with the client. It is also possible to create numerous reports based on one dataset, so we were easily able to create the two reports as outlined in the section above. There are numerous features that can be added to reports, such as different forms of bar graphs, line charts, scatter charts, tree maps, geographical maps, KPIs, slicers, and much more. For our two reports, we primarily used bar graphs and slicers.

Power BI essentially was the architecture of our entire project once we were able to upload the data that our client, Chris, sent us. The workspace that we shared with our client consists of both the dataset along with the two report files. Unlike other groups, we do not have any code as a part of our project. So we do not have three tier architecture or anything like that. Everything is simply encompassed by Power BI.

Chapter 4: Design

We created reports based on the functional requirements listed in chapter 2. Each report includes an interactive user interface. Users are able to select a date range using the slicer in the top right corner of each report. Furthermore, for Report 1, there is the capability to drill up or drill down and show the graphs by day, week, month or even year. An image of report 1 is shown below.

Report 1 - Date Range Listener

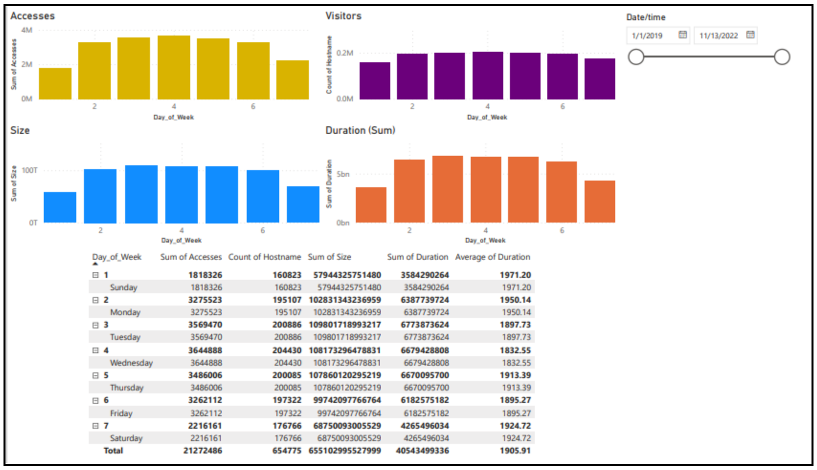


21272486 Sum of Accesses
40543499336 Sum of Duration
1905.91 Average of Durat...
654775 Count of Hostna...
655102995527999 Sum of Size
1/1/2019 11/13/2022

Report 2 consists of two pages. The first page shows the range of time broken down by day of week. The second page shows the date range broken down by hour. Again, the user is able to interact with these reports and select the date range that they would like to see. Furthermore, they can also click on a single bar in the graphs to see the outputs for just that given day of week or hour of day.

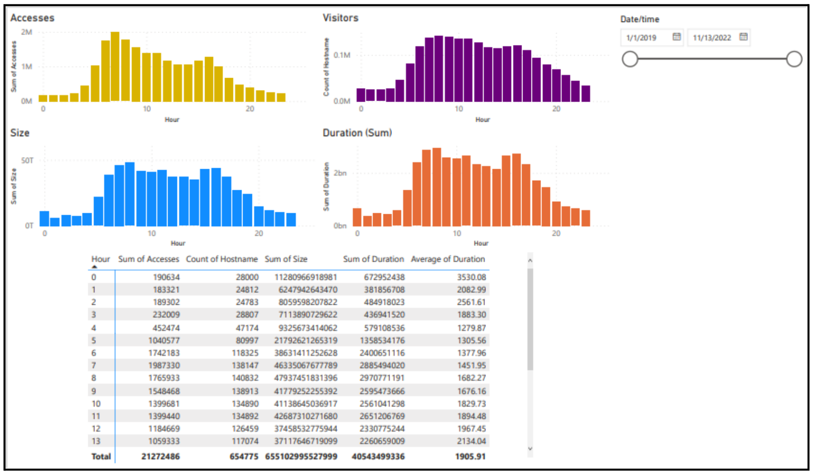
Report 2 - Daypart Calculator

Page 1



Day_of_Week	Sum of Accesses	Count of Hostname	Sum of Size	Sum of Duration	Average of Duration
⊟ **1**	**1818326**	**160823**	**57944325751480**	**3584290264**	**1971.20**
Sunday	1818326	160823	57944325751480	3584290264	1971.20
⊟ **2**	**3275523**	**195107**	**102831343236959**	**6387739724**	**1950.14**
Monday	3275523	195107	102831343236959	6387739724	1950.14
⊟ **3**	**3569470**	**200886**	**109801718993217**	**6773873624**	**1897.73**
Tuesday	3569470	200886	109801718993217	6773873624	1897.73
⊟ **4**	**3644888**	**204430**	**108173296478831**	**6679428808**	**1832.55**
Wednesday	3644888	204430	108173296478831	6679428808	1832.55
⊟ **5**	**3486006**	**200085**	**107860120295219**	**6670095700**	**1913.39**
Thursday	3486006	200085	107860120295219	6670095700	1913.39
⊟ **6**	**3262112**	**197322**	**99742097766764**	**6182575182**	**1895.27**
Friday	3262112	197322	99742097766764	6182575182	1895.27
⊟ **7**	**2216161**	**176766**	**68750093005529**	**4265496034**	**1924.72**
Saturday	2216161	176766	68750093005529	4265496034	1924.72
Total	**21272486**	**654775**	**655102995527999**	**40543499336**	**1905.91**

Page 2



Hour	Sum of Accesses	Count of Hostname	Sum of Size	Sum of Duration	Average of Duration
0	190634	28000	11280966918981	672952438	3530.08
1	183321	24812	6247942643470	381856708	2082.99
2	189302	24783	8059598207822	484918023	2561.61
3	232009	28807	7113890729622	436941520	1883.30
4	452474	47174	9325673414062	579108536	1279.87
5	1040577	80997	21792621265319	1358534176	1305.56
6	1742183	118325	38631411252628	2400651116	1377.96
7	1987330	138147	46335067677789	2885494020	1451.95
8	1765933	140832	47937451831396	2970771191	1682.27
9	1548468	138913	41779252255392	2595473666	1676.16
10	1399681	134890	41138645036917	2561041298	1829.73
11	1399440	134892	42687310271680	2651206769	1894.48
12	1184669	126459	37458532775944	2330775244	1967.45
13	1059333	117074	37117646719099	2260659009	2134.04
Total	**21272486**	**654775**	**655102995527999**	**40543499336**	**1905.91**

Overall, our design was dictated by the wants and needs of our client. He wanted the reports to match their old reporting tool's reports as closely as possible. This is why each report shows the information that it does and uses the specific graphs and tables that it does. In the future, these reports can be added to and changed; however ,the product we ended up with is almost an exact replica of the client's old reporting tool, which is exactly what we were tasked with. Furthermore, documentation for how these reports were created is included in a later chapter of this report.

Chapter 5: Implementation

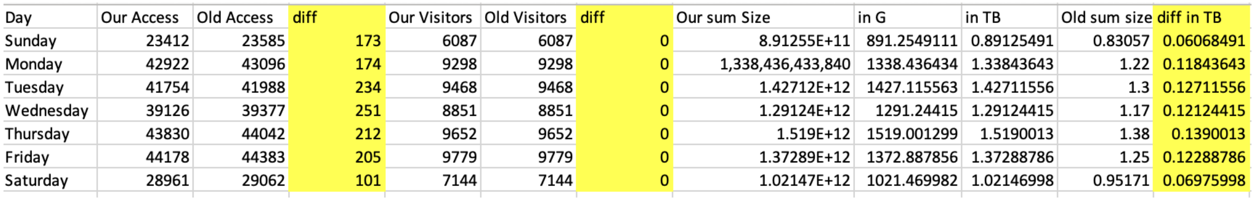
Our project initially utilized the statistical software R to view and compile all data from our client into a .csv file. We determined R was the most optimal software to view and compile our data due to the ability to manage large data sets, including the data from our client. Outside of R, our project did not require any coding or coding languages due to us primarily utilizing PowerBI. Refer to this code for the standards and comments used in R.

Our implementation process of this code was, after compiling in R, we uploaded the .csv files to PowerBI where we used that data to create those reports. Everyone in our group interacted with the data in PowerBI to help generate and modify the reports dashboards.

Organization of code base is not applicable due to us primarily using PowerBI on the web browser and desktop versions. We did not use any outside or additional packages other than PowerBI.

Chapter 6: Quality Assurance & Testing

The primary objective of our quality assurance and testing process was to ensure that our reports were generating the exact same data as WFYI's old reporting tool. To begin, we met with our client, Chris, over zoom and he walked us through the reports that the old reporting tool generated. He then selected a set of dates for each report and sent the criteria and outputs for each of those reports to us. After this, we created an Excel document and ran our reports with the exact same dates. We, then, entered the outputs from our report, and the outputs from his reports in order to test the accuracy of our reports. An example of this is shown in the image below. Here, the differences are highlighted, and, unfortunately, we found that our outputs did not align with the outputs from the reports that Chris shared with us, with the exception of the category visitors.



Day	Our Access	Old Access	diff	Our Visitors	Old Visitors	diff	Our sum Size	in G	in TB	Old sum size	diff in TB
Sunday	23412	23585	173	6087	6087	0	8.91255E+11	891.2549111	0.89125491	0.83057	0.06068491
Monday	42922	43096	174	9298	9298	0	1,338,436,433,840	1338.436434	1.33843643	1.22	0.11843643
Tuesday	41754	41988	234	9468	9468	0	1.42712E+12	1427.115563	1.42711556	1.3	0.12711556
Wednesday	39126	39377	251	8851	8851	0	1.29124E+12	1291.24415	1.29124415	1.17	0.12124415
Thursday	43830	44042	212	9652	9652	0	1.519E+12	1519.001299	1.5190013	1.38	0.1390013
Friday	44178	44383	205	9779	9779	0	1.37289E+12	1372.887856	1.37288786	1.25	0.12288786
Saturday	28961	29062	101	7144	7144	0	1.02147E+12	1021.469982	1.02146998	0.95171	0.06975998

As shown above, the visitors category perfectly matched the old reporting tool. The visitors is calculated by looking at the unique hostnames during the given time. This helped give us insight into the potential cause of the data discrepancies. We came up with a theory that the discrepancy was being caused by duplicated data. We created an R script to test this theory before changing all the data since that's a large undertaking, and we were able to verify that we had made a false assumption that the old reporting tool was removing duplicate records when in actuality it was not. Both the Excel document and R scripts used have been included in the appendix.

After identifying the problem, we went through the process of re-uploading our dataset to Power BI without removing duplicates. After doing this, we performed the exact same test comparing our report outputs to the outputs of Chris' old reporting tool, and we found that this time, they matched exactly.

Since none of our project involved physical code, we do not have examples of sample code showing different test cases. However, the Excel document shows the tests we ran by comparing the data our client gave us to the data that our Power BI reports were outputting.

Chapter 7: Project Organization & Management

Izzy Austin was the team leader throughout the entire semester. The rest of our team members: Maddie Neely, Nick Donnelly, Erin Penner, Nic Reilly, Nolan Knight, and Maddy Bloom all held similar roles. We all worked on creating the PowerBI reports according to our clients functional requirements. We, then, worked together to perform data validation and troubleshoot the problems we were seeing with discrepancies of the data.

Izzy was the team leader who helped to keep the team on track throughout the semester. She handled the majority of the communication with our client and made sure everyone on the team knew what needed to be done and when. On top of that she helped out with the reports, documentation, and presentation creation whenever necessary.

To begin, everyone spent time researching different data storage and visualization options to present to our client in order to select a tool that would be used for the rest of the semester. However, this was specifically spearheaded by Nic, Nolan and Maddy.

While the rest of the team was working on researching a tool to use for the project, Nick and Maddie, who were the R experts, helped with reading in the initial data files that were too big to look at in excel. This involved reading in the data to understand what data was available, writing scripts to combine all the individual .csv files and remove duplicates, and, later, creating new R scripts to troubleshoot the causes of the data discrepancies.

After understanding and combining the data, Erin and Maddie played a primary role in creating Reports 1 and 2. First, Erin specifically helped with getting all the data from the .csv files into the proper formatting into PowerBI so it could be used to create the reports that our client requested. This was a large undertaking due to the extremely large quantity of data that was updated several times throughout the course of the semester.

After creating the reports Erin, Izzy, and Maddie all helped with creating the documentation, which were step by step guides on how to create each report. This was a vital part of our project as our client was unfamiliar with PowerBI and requested instructions so he could replicate and build on the reports that we created throughout the semester.

Finally, Nolan and Maddy played a key role in presenting our project in a clear and concise manner. They spent lots of their time helping create our poster as well as our other Sprint presentations. They helped to pull the variety of work that was being done by the team together to make it more presentable and easy for our client as well as our class to understand.

In order to accomplish everything that our team was able to throughout this semester, one of our keys to success was communication. We had weekly meetings where we discussed who was doing what and if anyone needed help on what they were working on at any given time. After every Sprint, we performed a retrospective meeting to evaluate what we were doing well as a team and what areas we could improve on. Communication was something that we consistently did well as we talked when we were together and had a group chat if we needed to share ideas or ask questions outside of class time.

Another thing we did after each Sprint was plan a meeting with our client to get his feedback and then we would incorporate this into our planning and creation of goals for the next Sprint. All of our work was clearly organized on PowerBI and in Google Drive so each team member was aware of exactly what they needed to be doing at any given time.

All our weekly status reports and PowerBI documentation has been included below:

Weekly Status Reports

Sprint 1

Week 1

Week 2

Week 3

Sprint 2

Week 4

Week 5

Sprint 3

Week 6

Week 7

Week 8

Sprint 4

Week 9

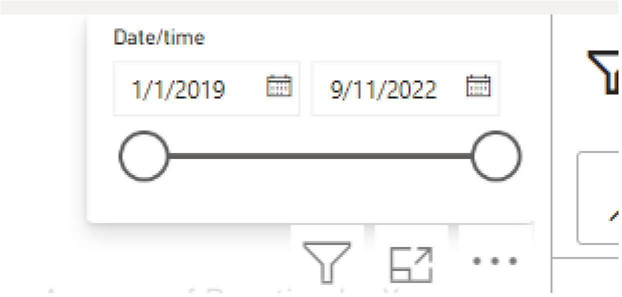
Week 10

Sprint 5

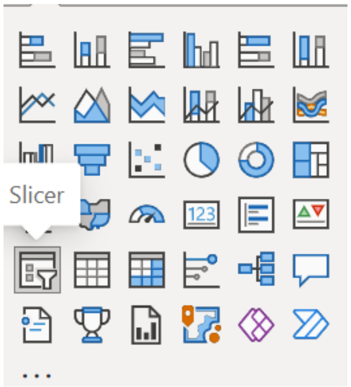
Week 11

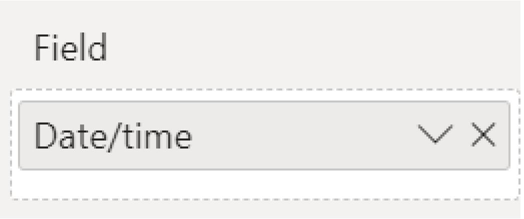
Making Report 1 (Date Range Listener)

Slicer with Date/Time



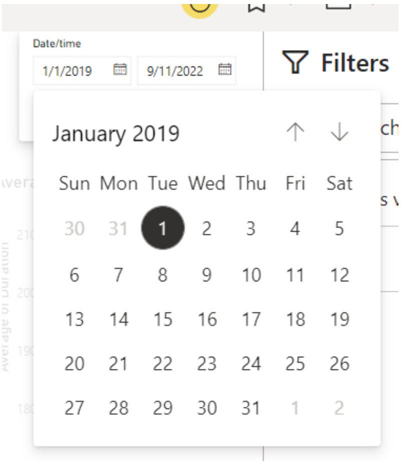
1. From the visualizations tab, add the slicer.
2. Add the field, Date/Time.



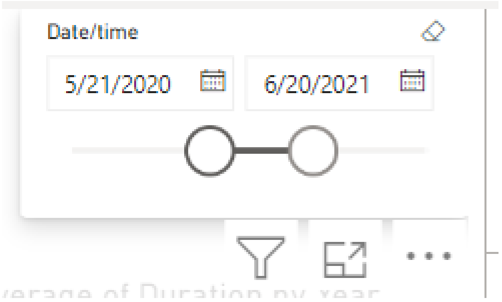

Field

Date/time

3. On the report, change the date range by dragging the slicer or entering a start date and end date using the calendar.

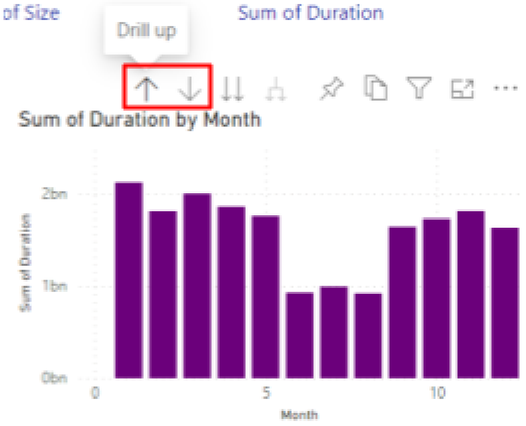

Date/time

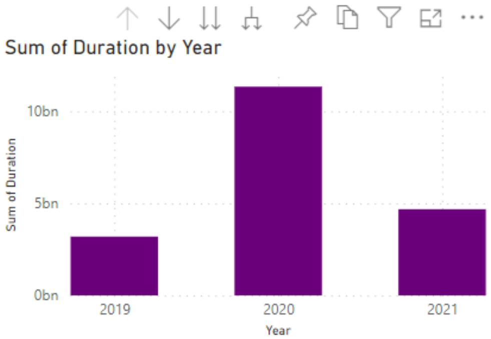
1/1/2019 9/11/2022


Date/time

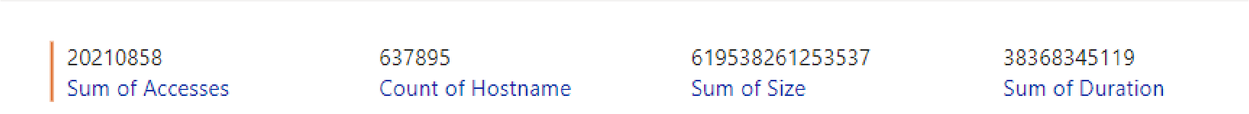
5/21/2020 6/20/2021

4. You can also use the *Drill Up* and *Drill Down* buttons to change the graph based on the next level of date hierarchy in the report.





Multi-Row Card with Sum of Accesses, Count of Hostname, Sum of Size, and Sum of Duration



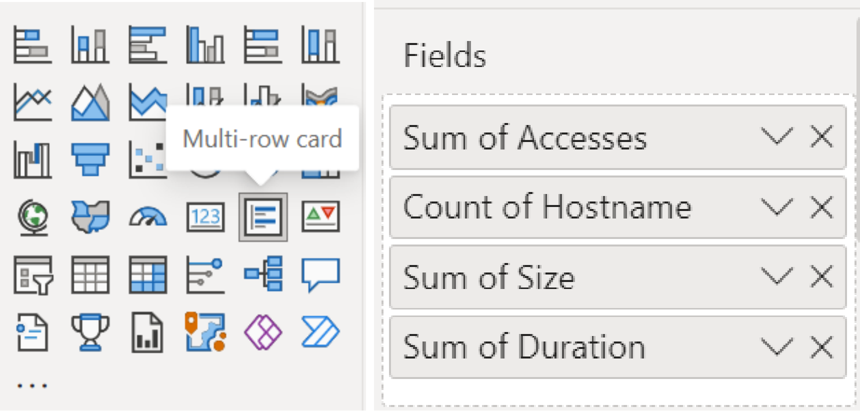
20210858 Sum of Accesses

637895 Count of Hostname

619538261253537 Sum of Size

38368345119 Sum of Duration

1. From the visualizations tab, add a multi-row card.
2. Add the fields:
    a. Accesses
    b. Hostname
    c. Size
    d. Duration



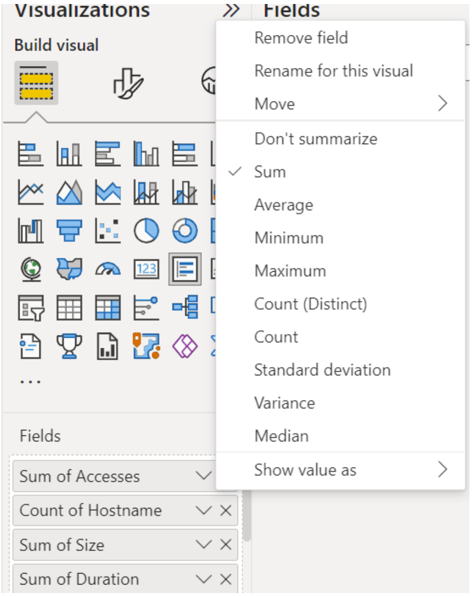
Sum of Accesses

Count of Hostname

Sum of Size

Sum of Duration

3. Using the dropdown next to Accesses, Hostname, Size, and Duration, change the fields to:
    a. Sum of Accesses
    b. Count of Hostname
    c. Sum of Size
    d. Sum of Duration



Sum of Accesses

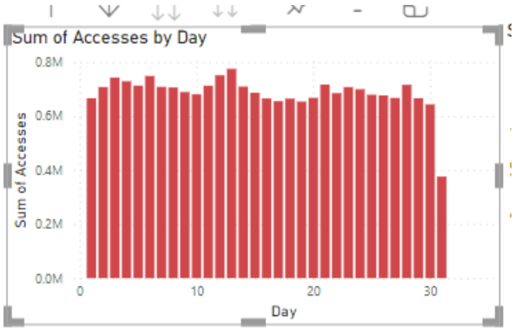
Count of Hostname

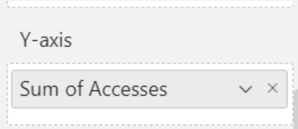
Sum of Size

Sum of Duration

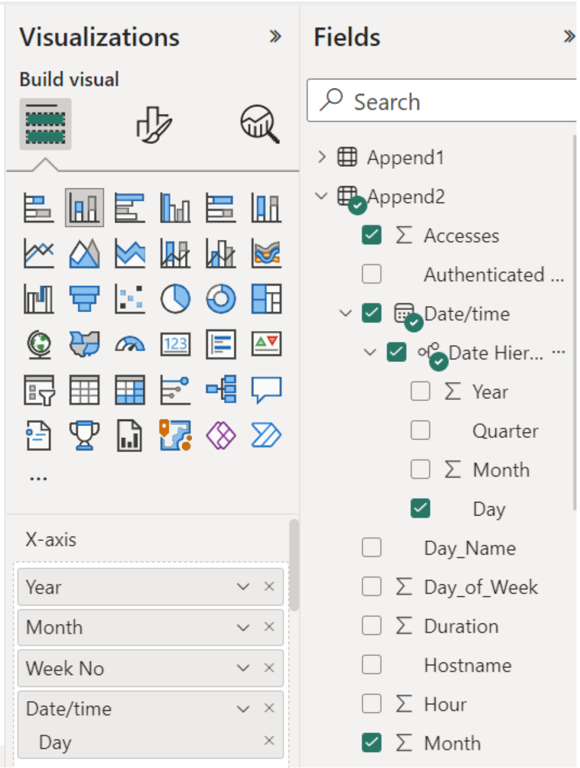
Bar Charts

- Sum of Accesses




Y-axis

Sum of Accesses


Visualizations

Build visual

X-axis

Year

Month

Week No

Date/time
Day

Fields

Search

Append1

Append2

☑ Σ Accesses

□ Authenticated ...

☑ Date/time

☑ Date Hier...

□ Σ Year

□ Quarter

□ Σ Month

☑ Day

□ Day_Name

□ Σ Day_of_Week

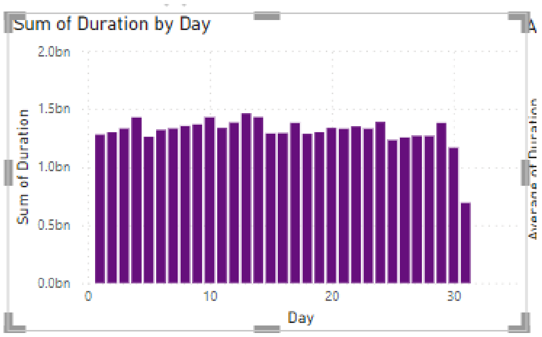
□ Σ Duration

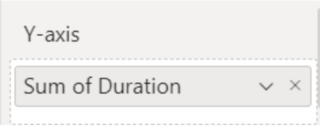
□ Hostname

□ Σ Hour

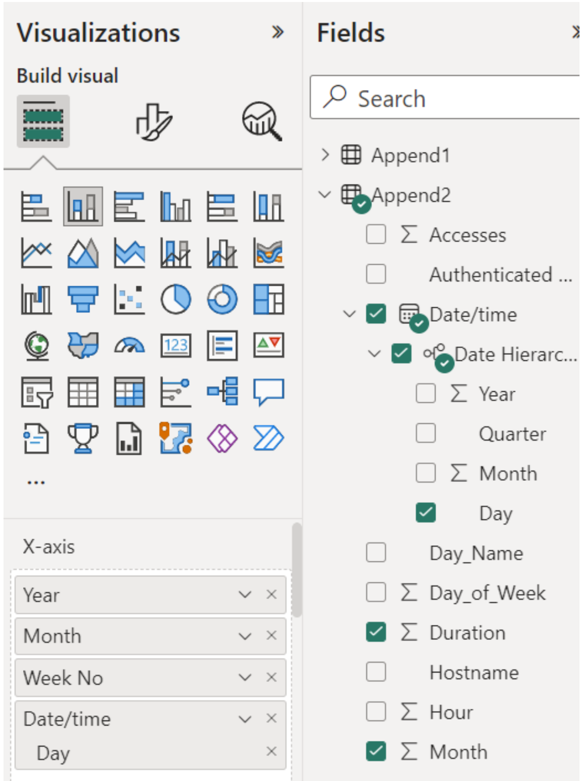
☑ Σ Month

- Sum of Duration




Y-axis

Sum of Duration


Visualizations

Build visual

X-axis

Year

Month

Week No

Date/time
Day

Fields

Search

Append1

Append2

□ Σ Accesses

□ Authenticated ...

☑ Date/time

☑ Date Hierarc...

□ Σ Year

□ Quarter

□ Σ Month

☑ Day

□ Day_Name

□ Σ Day_of_Week

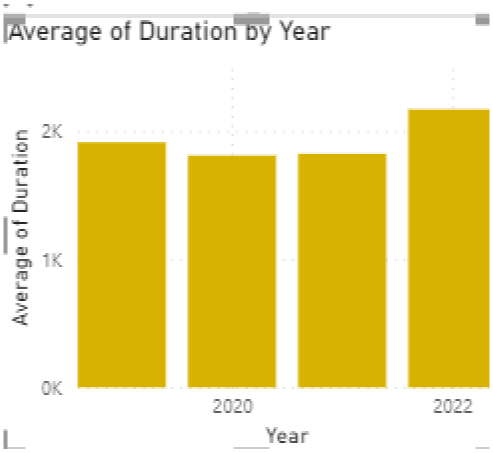
☑ Σ Duration

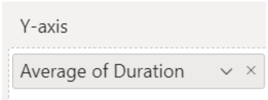
□ Hostname

□ Σ Hour

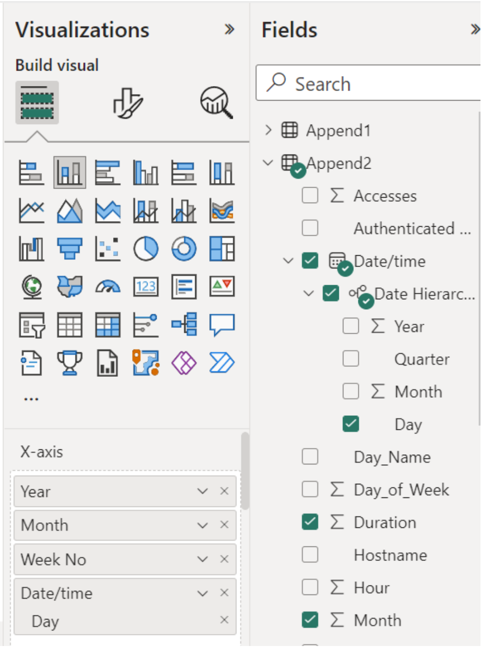
☑ Σ Month

- Average Duration




Y-axis

Average of Duration ⌄ ×



X-axis

Year ⌄ ×

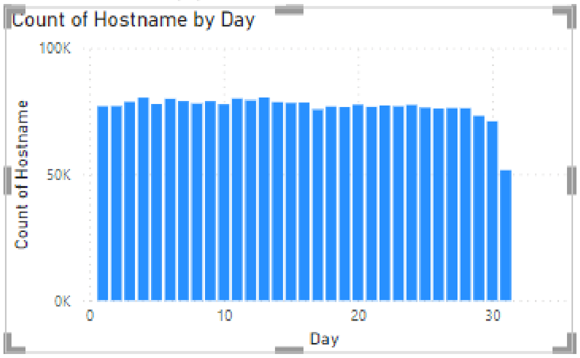
Month ⌄ ×

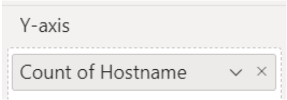
Week No ⌄ ×

Date/time ⌄ ×
Day ×

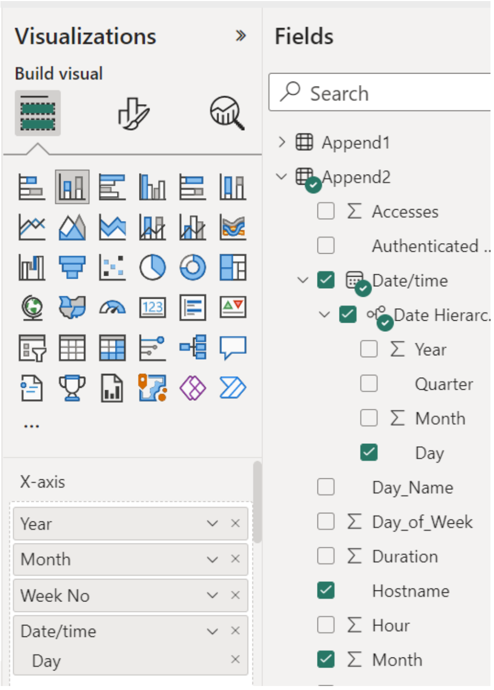
› Append1
⌄ Append2
□ Σ Accesses
□ Authenticated ...
⌄ ☑ Date/time
⌄ ☑ Date Hierarc...
□ Σ Year
□ Quarter
□ Σ Month
☑ Day
□ Day_Name
□ Σ Day_of_Week
☑ Σ Duration
□ Hostname
□ Σ Hour
☑ Σ Month

- Count of Hostname




Y-axis

Count of Hostname ⌄ ×



X-axis

Year ⌄ ×

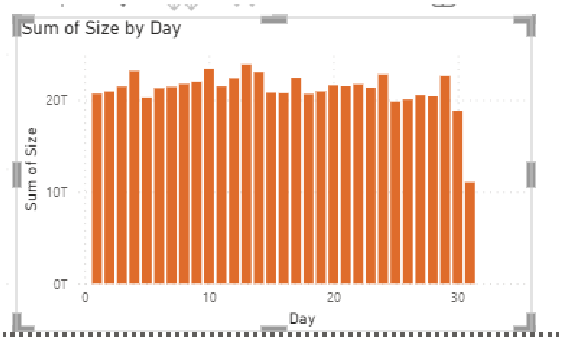
Month ⌄ ×

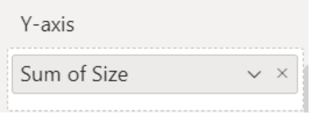
Week No ⌄ ×

Date/time ⌄ ×
Day ×

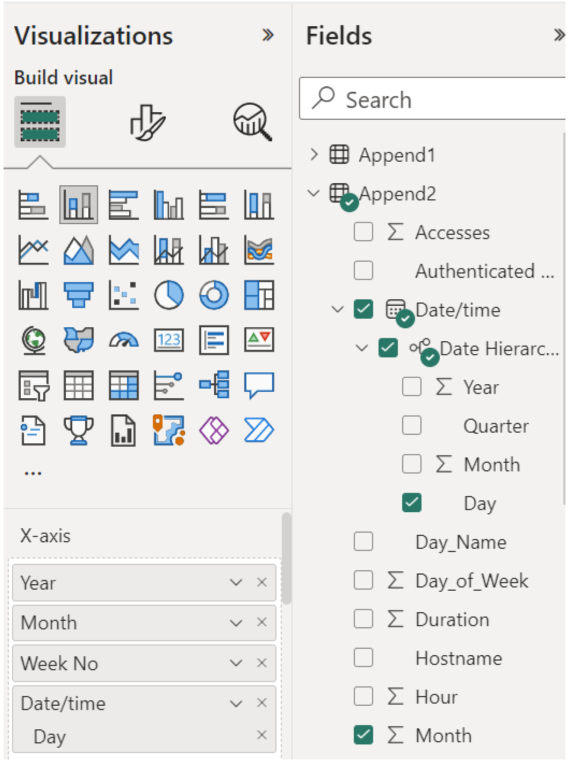
› Append1
⌄ Append2
□ Σ Accesses
□ Authenticated .
⌄ ☑ Date/time
⌄ ☑ Date Hierarc.
□ Σ Year
□ Quarter
□ Σ Month
☑ Day
□ Day_Name
□ Σ Day_of_Week
□ Σ Duration
☑ Hostname
□ Σ Hour
☑ Σ Month

- Sum of Size




Y-axis

Sum of Size



X-axis

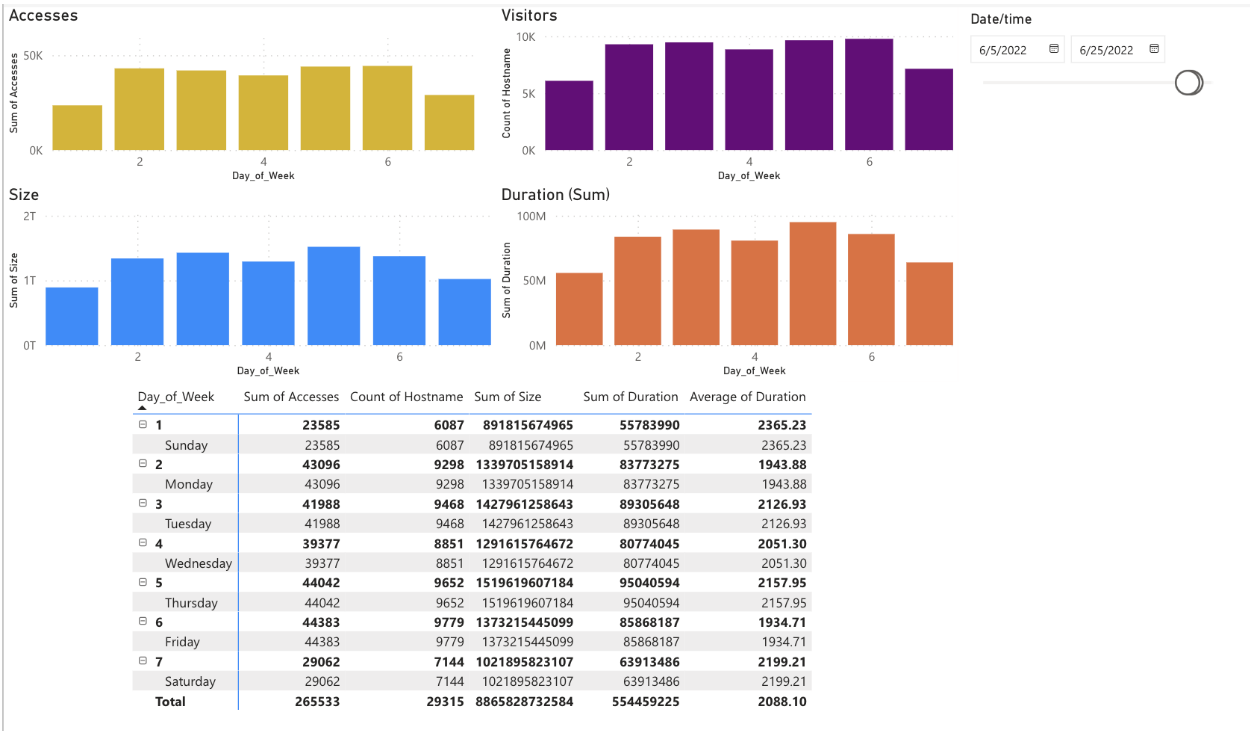
- Year
- Month
- Week No
- Date/time
  - Day

Fields

- Append1
- Append2
  - □ Σ Accesses
  - □ Authenticated ...
  - ☑ Date/time
    - ☑ Date Hierarc...
      - □ Σ Year
      - □ Quarter
      - □ Σ Month
      - ☑ Day
  - □ Day_Name
  - □ Σ Day_of_Week
  - □ Σ Duration
  - □ Hostname
  - □ Σ Hour
  - ☑ Σ Month

Making Report 2 (Daypart Calculator)

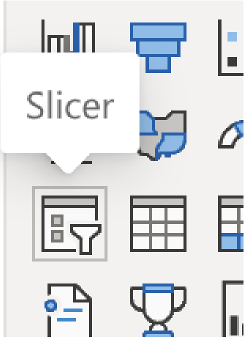
Day of Week Report

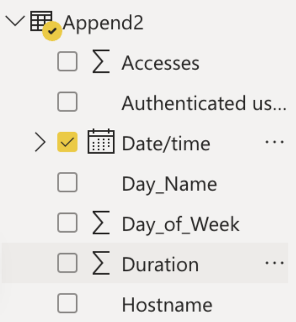


Day_of_Week	Sum of Accesses	Count of Hostname	Sum of Size	Sum of Duration	Average of Duration
⊟ **1**	**23585**	**6087**	**891815674965**	**55783990**	**2365.23**
Sunday	23585	6087	891815674965	55783990	2365.23
⊟ **2**	**43096**	**9298**	**1339705158914**	**83773275**	**1943.88**
Monday	43096	9298	1339705158914	83773275	1943.88
⊟ **3**	**41988**	**9468**	**1427961258643**	**89305648**	**2126.93**
Tuesday	41988	9468	1427961258643	89305648	2126.93
⊟ **4**	**39377**	**8851**	**1291615764672**	**80774045**	**2051.30**
Wednesday	39377	8851	1291615764672	80774045	2051.30
⊟ **5**	**44042**	**9652**	**1519619607184**	**95040594**	**2157.95**
Thursday	44042	9652	1519619607184	95040594	2157.95
⊟ **6**	**44383**	**9779**	**1373215445099**	**85868187**	**1934.71**
Friday	44383	9779	1373215445099	85868187	1934.71
⊟ **7**	**29062**	**7144**	**1021895823107**	**63913486**	**2199.21**
Saturday	29062	7144	1021895823107	63913486	2199.21
Total	**265533**	**29315**	**8865828732584**	**554459225**	**2088.10**

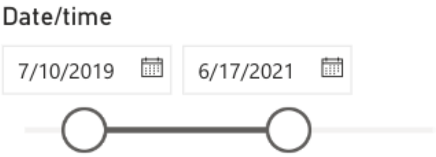
1. Select Slicer

2. Select Date/Time field to add it to the slicer





Append2
- □ ∑ Accesses
- □ Authenticated us...
- ☑ Date/time
- □ Day_Name
- □ ∑ Day_of_Week
- □ ∑ Duration
- □ Hostname

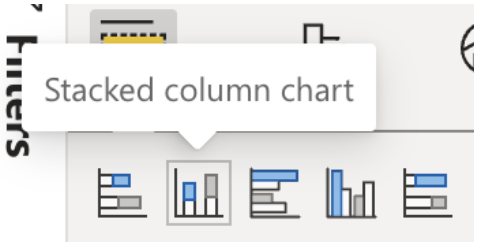


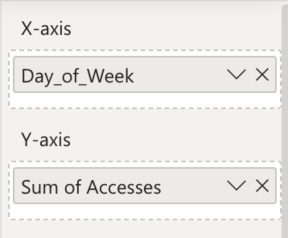
Date/time

7/10/2019 6/17/2021

3. Now add Stacked Column Chart to make bar graph for accesses

4. Add the fields Day_of_Week and Sum of Accesses



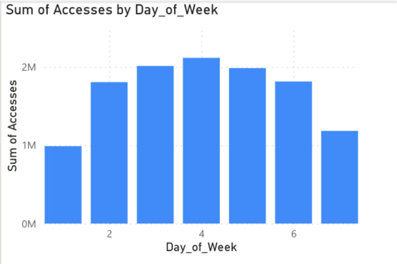


X-axis

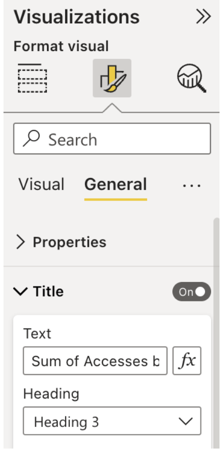
Day_of_Week

Y-axis

Sum of Accesses



5. Change Title Names - Click format visual on the visualization tabs, and fields will pop up to change the title and axis titles.



Visualizations

Format visual

Search

Visual General

Properties

Title On

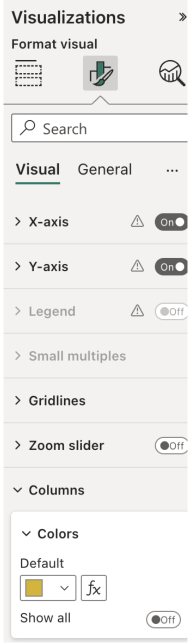
Text

Sum of Accesses b

Heading

Heading 3

6. Change Colors - Under the Visualizations tab select format visualizations, then select columns, and then change color there.



Visualizations

Format visual

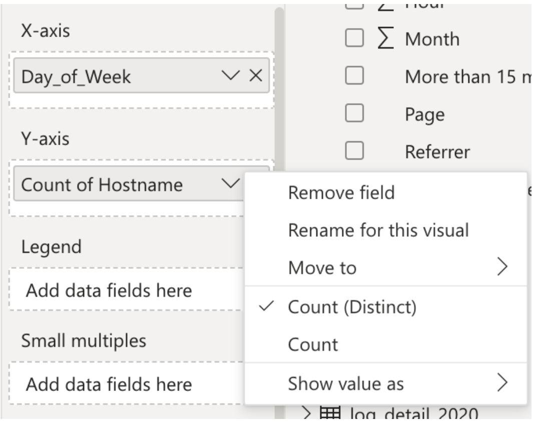
Search

Visual General ...

- X-axis On
- Y-axis On
- Legend Off
- Small multiples
- Gridlines
- Zoom slider Off
- Columns
  - Colors
  - Default 🟨 *fx*
  - Show all Off

7. Visitors Bar Graph

- Create a Stacked Column Chart, the same as above
- Select Day_of_week for the x-axis
- Select Count_of_Hostname for the y-axis, then click the arrow next to Count_of_Hostname and then choose the option for count(distinct)



X-axis

Day_of_Week

Y-axis

Count of Hostname

Legend

Add data fields here

Small multiples

Add data fields here

8. Size Bar Graph

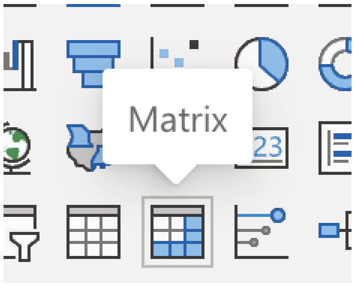
- Follow the same steps above, except select Sum Of Size for the y-axis

9. Duration Bar Graph

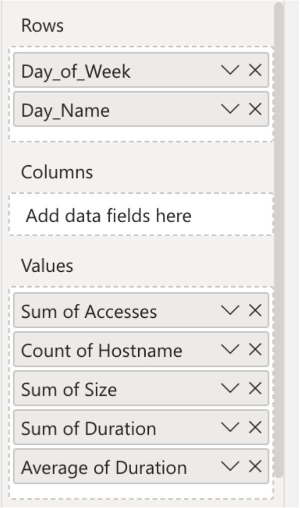
- Follow the same steps above, except select Sum Of Duration for the y-axis

10. Creating the table

- Select Matrix Option



- Add the field names as shown below
  - Make to click the drop down for count of Hostname and select the “ count distinct” option



Rows

- Day_of_Week
- Day_Name

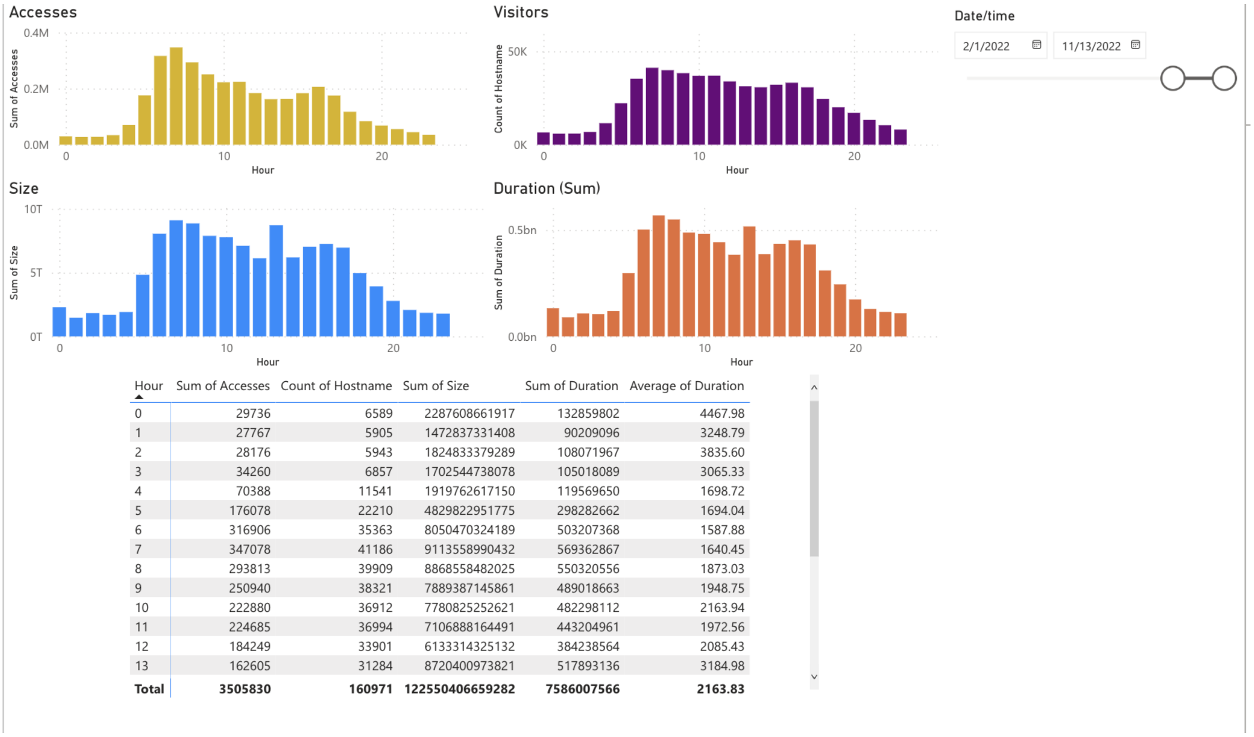
Columns

Add data fields here

Values

- Sum of Accesses
- Count of Hostname
- Sum of Size
- Sum of Duration
- Average of Duration

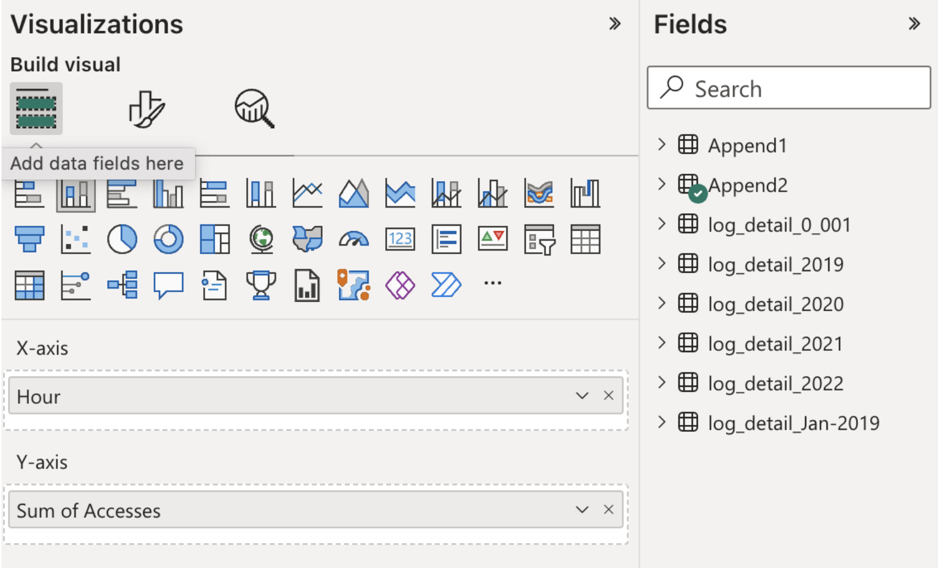
Hour of Day Report



Hour	Sum of Accesses	Count of Hostname	Sum of Size	Sum of Duration	Average of Duration
0	29736	6589	2287608661917	132859802	4467.98
1	27767	5905	1472837331408	90209096	3248.79
2	28176	5943	1824833379289	108071967	3835.60
3	34260	6857	1702544738078	105018089	3065.33
4	70388	11541	1919762617150	119569650	1698.72
5	176078	22210	4829822951775	298282662	1694.04
6	316906	35363	8050470324189	503207368	1587.88
7	347078	41186	9113558990432	569362867	1640.45
8	293813	39909	8868558482025	550320556	1873.03
9	250940	38321	7889387145861	489018663	1948.75
10	222880	36912	7780825252621	482298112	2163.94
11	224685	36994	7106888164491	443204961	1972.56
12	184249	33901	6133314325132	384238564	2085.43
13	162605	31284	8720400973821	517893136	3184.98
Total	**3505830**	**160971**	**122550406659282**	**7586007566**	**2163.83**

To Create this report, add a new page to the original report. Then you can recreate in one of two ways.

1. Copy all of the reports individually onto the new page and simply change the X-axis to be Append2 Hour instead of Day of Week.



X-axis

Hour

Y-axis

Sum of Accesses

Append1

Append2

log_detail_0_001

log_detail_2019

log_detail_2020

log_detail_2021

log_detail_2022

log_detail_Jan-2019

2. Individually recreate each report following the same steps listed above, but using hour in place of Day of Week.

Chapter 8: Future Work

Due to the sunsetting of WFYI's previous system, this project was designed to be a semester-long project, with no future work necessary. For EPICS students, all functional requirements were satisfied, both the reports requested by our client, along with the documentation, have already been completed and shared with our client. However, WFYI will now have the tools to add-on and create additional reports with our report Documentation, which includes a general overview and step-by-step approach to creating and managing the report dashboards. Any additional functional requirements will be determined by WFYI.

References

PowerBI
https://powerbi.microsoft.com/en-us/what-is-power-bi/

R
https://www.r-project.org/

WFYI
https://www.wfyi.org/

Appendices

Sprint Presentations

Sprint 1 Presentation

Sprint 2 Presentation

Sprint 3 Presentation

Sprint 4 Presentation

Sprint 5 Presentation

Peer Evaluation Feedback

Sprint 1 Peer Evaluation

Sprint 2 Peer Evaluation

Sprint 3 Peer Evaluation

Sprint 4 Peer Evaluation

Quality Assurance Files

Data Validation Excel

Data Validation R Script