5-2023

Automated Visualization Pipeline for Near Real-time Risk Management System

Paris Joslin
University of Arkansas, Fayetteville

Automated Visualization Pipeline for Near Real-time Risk Management System

This research is supported by the Arkansas High Performance Computing Center which is funded through multiple National Science Foundation grants and the Arkansas Economic Development Commission. This project is also supported by the Walmart Food Safety Collaboration Center with funds from the Walmart Foundation.

Abstract:
In modern society, technological capabilities and the amount of data readily available to users continue to grow exponentially. Many have adopted these new capabilities but lack the infrastructure needed to efficiently utilize high-powered software and programs. Without a method to collect, store, and process large datasets in real-time, individuals and businesses can quickly become overwhelmed, inhibiting effective decision-making processes. There is potential to improve decision-making abilities by enhancing the computing infrastructure. To accomplish this task, we will explore the ideas surrounding High Performance Computing (HPC) and data visualization software. High Performance Computing is the ability to process data and perform complex calculations at high speeds by using a cluster of computer servers that work in parallel to boost computer power. This research will seek to expand on the benefits of joining HPC systems with data visualization software such as Power BI. The Arkansas High Performance Computing Center (AHPCC) will be used to demonstrate data processing capabilities, and Power BI will be used to translate the data into effective decision-making visuals. While this project focuses on a small test scenario, the results from this research serve as motivation to implement this type of visualization pipeline in a variety of industries to help key stakeholders better analyze the state of businesses in a real-time environment.

1. General Problem and Motivation

With the expansion of technological capabilities in modern society, the need to be able to analyze and externally visualize large amounts of data in real-time situations has become increasingly important. Both capabilities require significant computer processing power and highly adaptive software tools. The research described in this paper is generally motivated by these topics. This paper also expands on other efforts focused on learning-based approaches to dynamic risk and the need for supply chain decision makers to consume results outside of a computational environment.

The risk assessment of poultry supply chains in China is used as a test scenario to demonstrate the benefits of creating an automated visualization pipeline [5]. With multiple test locations in the supply chain and new data being captured frequently, the amount of data and how to effectively extract key takeaways can become overwhelming for decision makers and stakeholders. This paper will demonstrate how a visualization pipeline of this type is developed and used to target areas of risk and, ultimately, enhance decision making abilities.

2. Solution Requirements

2.1. AHPCC

As stated previously, the need to process large amounts of data in real-time situations has become an integral part both in everyday decision making and in groundbreaking research environments. High Performance Computing (HPC) has begun to fulfill this need with the ability to perform high-speed data analyses across a variety of industries. The HPC infrastructure relies on a network of servers, also known as a cluster, that can run algorithms or programs in parallel to boost computer processing power.

The Arkansas High Performance Computing Center (AHPCC) is a facility at the University of Arkansas that aims to provide high performance computing services for use in research or multidisciplinary work. The AHPCC primarily uses the Pinnacle cluster, which consists of 100 Intel based nodes, 20 NVIDIA V100 GPU nodes, and 8 big memory nodes with 768 GB of ram per node that enable data science and machine learning capabilities [1]. While HPC systems have a wide array of applications, the AHPCC's primary goal is to "substantially enhance the productivity of a growing community of researchers, engineers, and scholars" as well as to "coordinate and add significant value to the research and discovery effort" [1]. Examples of research disciplines that utilize the capabilities of the HPC software include industrial engineering, chemistry, biology, and physics for various projects involving transportation systems, disease patterns, material behavior, and supply chain networks.

2.2. Power BI

Coupled with the need to store and analyze data is the ability to easily translate data findings into effective and interactive visuals. Microsoft Power BI is a business intelligence (BI) tool that specializes in creating these visualizations to aid in the decision-making process for both individuals and large corporations. With over 500 data source options, Power BI automates the process of securely connecting directly to cloud-based or remote data locations [3]. Power BI also provides the user with the ability to create workspaces where dashboards, reports, and data insights can be publicly shared with others to foster a more collaborative work environment.

2.3. General Capabilities

The pipeline that is developed in this paper requires several general capabilities to be deemed useful by stakeholders. The pipeline needs to be able to access data generated on a UNIX-based cluster at any time desired by the user or developer. This is satisfied by using the Pinnacle cluster on the AHPCC. The ability to move data to a public source for external consumption is also needed. Lastly, the public source needs to be able to process and visualize information while also being easily understood by both technical and non-technical decision makers. These requirements are met by utilizing the Power BI application discussed previously.

3. Solution Overview

3.1. Data Locations

There are two main datasets associated with this project. All data is located in the AHPCC in the *safety-data* directory in Figure 1.

Figure 1: The location of the safety-data directory in the AHPCC

The data files included in this directory contain information about risk values, which will be referred to as Risk Data, and sampling phase data by location, which will be referred to as Sampling Phase Data for the purposes of this project.

The Risk Data files include data about the risk values and locations of each node. The exact fields present in the data are ID, Latitude, Longitude, Node, Risk Value, Time, Date as shown in Figure 2.

ID	Latitude	Longitude	Node	Risk Value	Time	Date
1	48.1	126.2	Breeder	0.2024	1:22:42	9/8/22
2	21.28	109.06	Production	0.4033	2:32:52	8/30/22

Figure 2: The structure of the Risk Data files

These files are csv files, and the data structure must remain consistent with Figure 2 to be compatible with the data transformations performed in the Power BI dashboard.

The Sampling Phase Data files include information about the sampling phases and positive rates in each of the four sampling locations: Kwang Feng, Liuhe, Sunner, and Market. These files are Excel files with eight total worksheets. The fields present in the data are Sampling Date, Sampling Phase, Numbers of Samples, Numbers of positives, and Positive rate as shown in Figure 3. Note that the Sampling Phase in this dataset corresponds to the step of the production process in which the sample is taken.

Sampling Date	Sampling Phase		Numbers of Samples	Numbers of postive	Positive rate
2016.10	Anal Swab		20	6	30.0%
	Hair Eliminating		20	3	15.0%
	Evisceration		20	8	40.0%
	Cleaning after Evisceration		20	2	10.0%
	Chilling		20	0	0.0%
	Water Sample 9:00AM	Cleaning after Evisceration	5	1	20.0%
		Washing	5	0	0.0%
		Disinfection	3	0	0.0%
	Water Sample 11:00AM	Cleaning after Evisceration	5	0	0.0%
		Washing	5	0	0.0%
		Disinfection	3	0	0.0%
2016.11	Anal Swab		20	0	0.0%
	Hair Eliminating		20	4	20.0%
	Evisceration		20	1	5.0%
	Cleaning after Evisceration		20	5	25.0%
	Chilling		20	1	5.0%
	Water Sample 9:00AM	Cleaning after Evisceration	5	0	0.0%
		Washing	5	0	0.0%
		Disinfection	3	0	0.0%
	Water Sample 11:00AM	Cleaning after Evisceration	5	0	0.0%
		Washing	5	0	0.0%
		Disinfection	3	0	0.0%

Figure 3: The structure of the Sampling Phase Data files

The *safety-data* directory contains all necessary data and scripts for this project as shown in Figure 4 with the core data files highlighted.

Figure 4: The safety-data directory and all project data

3.2. Generating Data Files – Risk Data

The Risk Data files have a .csv file extension, so no file type conversions are needed. However, to build a history of risk values for each location, all risk data files need to be joined. This results in a singular file with all relevant risk value information across a range of dates for use in the Power BI dashboard visualization. The contents of the files are concatenated into a singular generated file using the commands shown in Figure 5.

```
awk '(NR == 1) || (FNR > 1)' *csv > SampleDataAll.csv
cat SampleDataAll.csv
```

Figure 5: Concatenation commands for Risk Data files

The awk command allows for the use of built-in variables like NR and FNR. ‘NR == 1’ locks the first line of the file to be used as headers, and ‘FNR > 1’ excludes line 1 of all other files to

prevent header duplication. '*csv' allows all csv files in the folder to be concatenated to the file "SampleDataAll", which is the file that is pushed to the GitHub repository [4].

3.3. Generating Data Files – Sampling Phase Data

As stated previously, the Sampling Phase Data files are Excel files; however, they need to be csv files in order to easily connect to the Power BI dashboard through GitHub. The standard Excel to csv conversion is only able to be used when the Excel workbook has a single worksheet. The original Sampling Phase Data file has eight total worksheets, which violates the standard conversion conditions.

To fix this issue, a Python script was written and executed using the Nano text editor. The commands to load the Python module and text editor is shown in Figure 6.

```
pinnacle-l5:pnjoslin:~$ module load gcc/11.2.1 python/anaconda-3.10b
pinnacle-l5:pnjoslin:~$ source /share/apps/bin/conda-3.10b.sh
(base) pinnacle-l5:pnjoslin:~$ conda activate xlrd
(xlrd) pinnacle-l5:pnjoslin:~$ nano XlsToCsv.py
```

Figure 6: The Python commands to access the Nano text editor

Once in the Nano text editor, the Python script utilizes a For loop to loop through the first four worksheets of the "PhaseData.xls" workbook and create a separate csv file for each individual sheet. The completed Python script is shown in Figure 7.

```
# import all required library
import xlrd
import csv
import pandas as pd

# open workbook by sheet index,
#gets the first four sheets in the PhaseData workbook
#writer object is created
i = 0
values = range(4)
for i in values:
        sheet = xlrd.open_workbook("PhaseData.xls").sheet_by_index(i)
        col = csv.writer(open("PhaseData_" + str(i) + ".csv", 'w', newline=""))

# writing the data into csv file
for row in range(sheet.nrows):
         # row by row write
         # operation is perform
         col.writerow(sheet.row_values(row))

#read csv file and convert
#into a dataframe object
#df = pd.DataFrame(pd.read_csv("T.csv"))

#show the dataframe
#df
```

Figure 7: The XlsToCsv.py Python script to convert the Sampling Phase Data files to csv files

The results of the executed Python script are the four csv files titled "PhaseData_0" with "0" being equal to the index of the worksheet in the "PhaseData.xls" workbook. The four files are found in Figure 8.

Figure 8: The four converted Sampling Phase Data csv files in the *safety-data* directory

3.4. Setting Node Value to Allow GitHub-AHPCC Connection:

All files must be pushed to a GitHub repository to be utilized in the Power BI dashboard. A direct connection from HPC to Power BI is not currently a capability of Power BI due to the authorizations needed to access directories on the HPC. To establish a connection to GitHub from the HPC, Data Analytics that are Robust and Trusted – Arkansas Research Platform (DART-ARP) needs to be joined. DART-ARP is an organization set up by the AHPCC to allow data collaboration between the networks. This prompts the owner of the directory to enter their GitHub username, password, and generated GitHub token to authorize all future attempts to push data to a specific GitHub repository. The specific GitHub repository that is used for this project is also called *safety-data* to remain consistent with the HPC directory naming convention. The information on joining DART-ARP can be found on the HPC Wiki [2].

The commands to push the files to the repository are executed on the Pinnacle shell in the HPC. All jobs pushed to sites outside of the University of Arkansas network, such as a GitHub repository, must be executed on Pinnacle nodes [5,6,7,12,13,14] as they are the only authorized nodes to handle this connection. For the purposes of this project, all commands are executed on Pinnacle node 5. When the Pinnacle shell is opened, the default node is node 1, but this needs to be manually altered by entering a 5 in the Pinnacle shell URL instead of a 1. After the node is changed, the working directory is set to the safety-data folder that houses the applicable data files using the command shown in Figure 9.

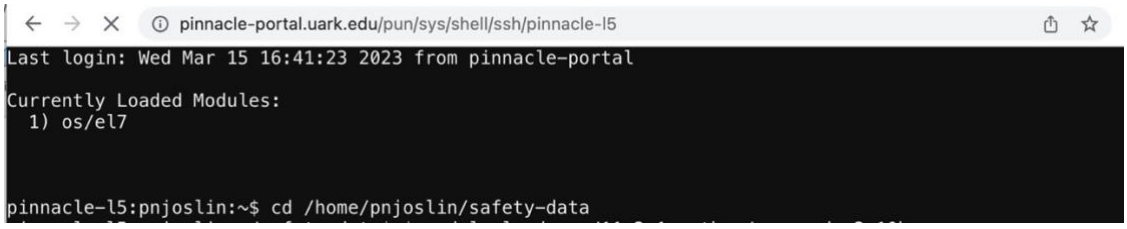


Figure 9: The command to set the working directory to the safety-data directory and the URL with the Pinnacle node 5 change.

3.5. GitHub Push

Once all files are converted, the push to the GitHub repository can be executed with all generated files for the Risk Data and Sampling Phase Data. To begin the push to the *safety-data*

repository, the files are added to the git push using the git add command followed by the path to the specific file. The files then must be committed using the git commit command. Finally, the files were pushed to the main branch of the *safety-data* repository using the git push command. All commands that were used to push the files to the GitHub repository are shown in Figure 10.

```
#adds the files to git push
git add /home/pnjoslin/safety-data/SampleDataAll.csv
git add /home/pnjoslin/safety-data/PhaseData_0.csv
git add /home/pnjoslin/safety-data/PhaseData_1.csv
git add /home/pnjoslin/safety-data/PhaseData_2.csv
git add /home/pnjoslin/safety-data/PhaseData_3.csv

# commits the files and any changes
git commit -m "commit files"

#push files to correct git repo
git branch -M main
git remote add origin https://github.com/pnjoslin/safety-data.git
git push -u origin main
```

Figure 10: The commands to push the files to the GitHub *safety-data* repository

After executing the git push commands, the files are shown in *the safety-data* repository in GitHub as shown in Figure 11.

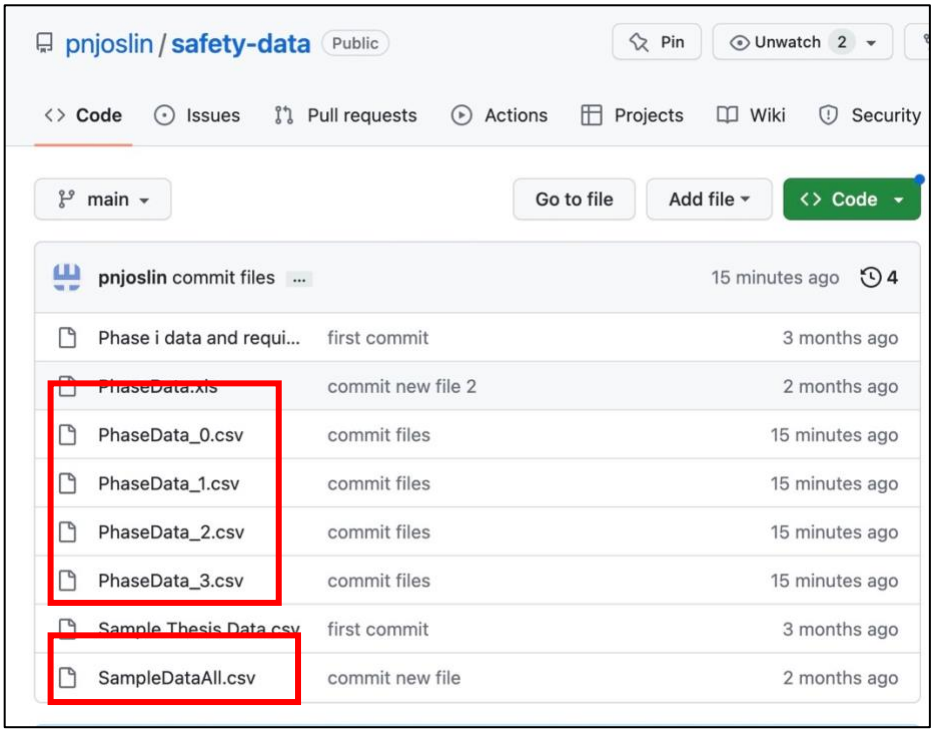


Figure 11: All files after being pushed to the *safety-data* GitHub repository

3.6. Automatic Refreshes:

To automatically refresh the data found in the GitHub repository, a cron job is scheduled. Creating the cron job is done using the Nano text editor that was used previously. Once the text

editor is open, the commands that are executed when running the cron job are added. These commands consist of all file generation and git push commands discussed in previous sections. After the cron job is saved, the job is scheduled to be executed every day at 2 a.m. All cron job specific commands are shown in Figure 12.

```
#opens text editor
EDITOR=nano crontab -e
#lists contents of cron table
crontab -l
# schedules refresh
0 2 * * * cd <safety-data>>/home/pnjoslin/cron.out
```

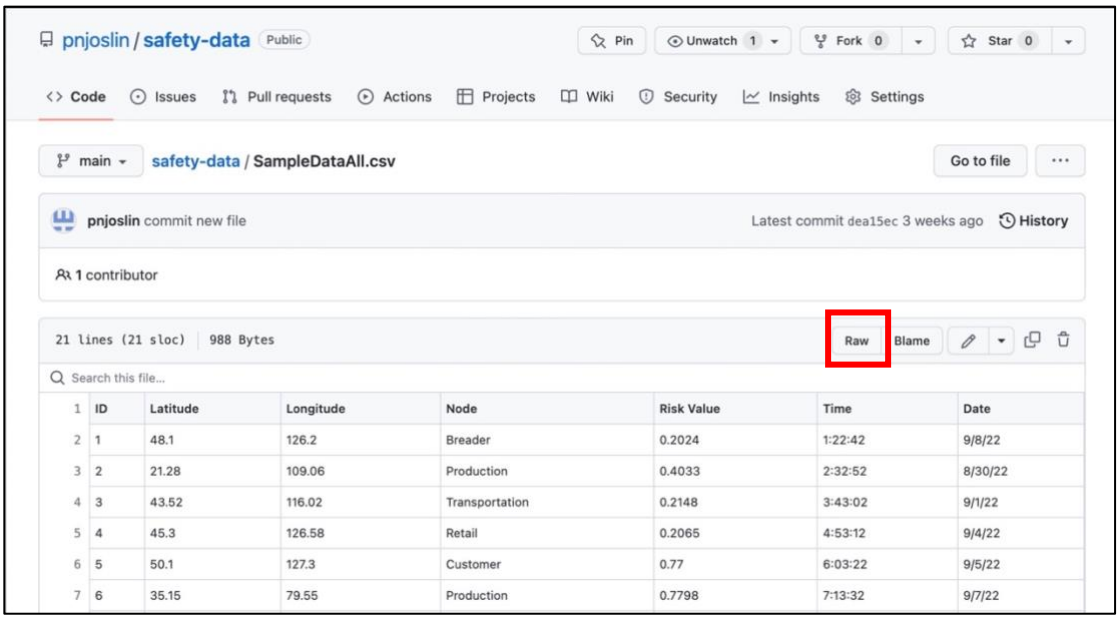
Figure 12: The cron job commands

These first command opens the Nano text editor, and the second lists the contents of the cron table. The third command schedules the 2 a.m. data pull to refresh the file in the GitHub repository. If the data pull needs to be edited to refresh at a different time, there are other scheduler commands to execute using the unix-cron string format, * * * * *. Each asterix in the string format represents a time unit: minute, hour, day of the month, month, and day of the week, respectively. For example, if the data needs to be refreshed every weekday at 2 a.m., the command would be altered to have a unix-cron string '0 2 * * 1-5'. Once the cron job has been successfully executed, an email is received by the owner of the directory in which the cron job was scheduled. However, this feature can be disabled by inserting the '- specify >/dev/null 2>&1' command directly after the third command in Figure 12.

If the commands in the cron job need to ping a site outside of the network within the cron.out file, such as a GitHub repository as mentioned previously, the Pinnacle nodes [5,6,7,12,13,14] will need to be used. Nodes [1,2,3,4,8,9,10,11] cannot be used in this case but can be used if the node does not need to reach a site outside of the University of Arkansas system.

3.7. Connection to Power BI:

The Power BI dashboard is linked to the raw URL of the pushed data files in the GitHub repository. Power BI must have access to the raw URL, which is captured by navigating to the raw data in the individual file in the GitHub repository as shown in Figure 13 and Figure 14.



	ID	Latitude	Longitude	Node	Risk Value	Time	Date
2	1	48.1	126.2	Breader	0.2024	1:22:42	9/8/22
3	2	21.28	109.06	Production	0.4033	2:32:52	8/30/22
4	3	43.52	116.02	Transportation	0.2148	3:43:02	9/1/22
5	4	45.3	126.58	Retail	0.2065	4:53:12	9/4/22
6	5	50.1	127.3	Customer	0.77	6:03:22	9/5/22
7	6	35.15	79.55	Production	0.7798	7:13:32	9/7/22

Figure 13: The button to extract the raw URL for each data file

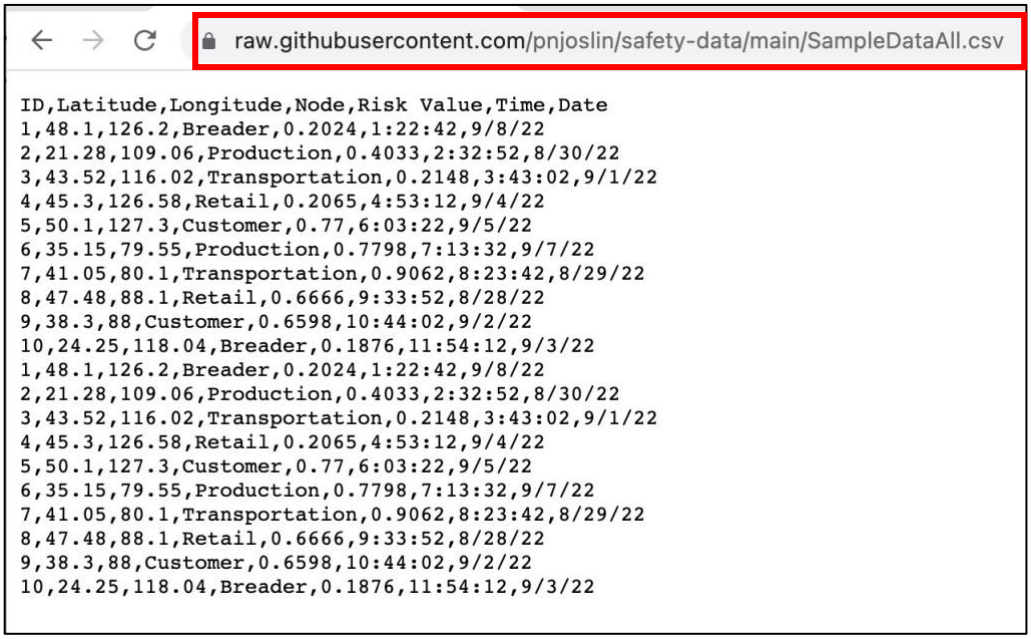


Figure 14: The raw URL for the Risk Data file as well as the raw data

This URL remains constant with each new HPC push to GitHub. The raw URLs for the four Sampling Phase Data and Risk Data files will need to be connected to the Power BI dashboard.

Once the raw URL has been copied, the connection can be made through the ‘Get Data’ tab in the desktop version of Power BI. Since the raw URLs are being used, the type of connection source needed is the ‘Web’ connection as shown in Figure 15.

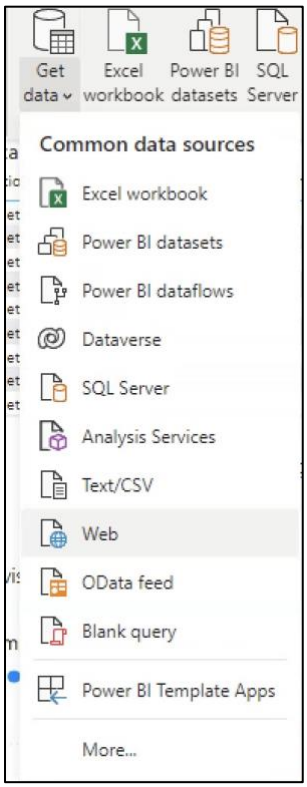


Figure 15: The location of the 'Web' connection option in Power BI

The Risk Data file URL is copied into the Web data source prompt, and the connection between the file and the dashboard is made. This process is repeated for all raw URLs for the Sampling Phase Data files.

3.8. Transforming Sample Risk Data

Once the data connections are established, the data must be transformed into a more user-friendly structure in the 'Transform data' tab in PowerBI as shown in Figure 16.

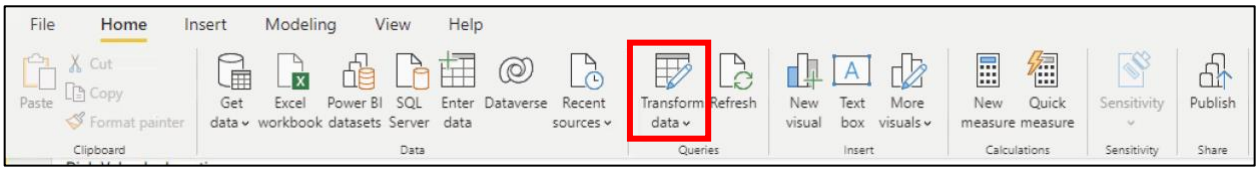
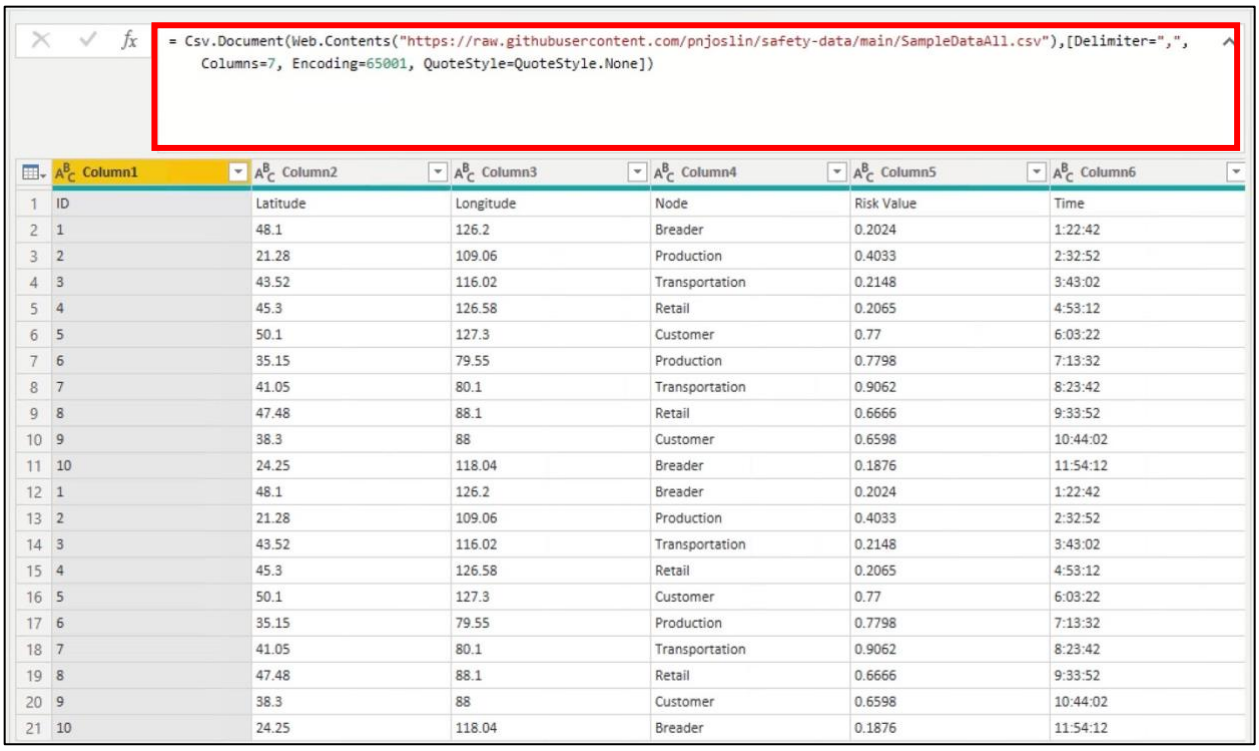


Figure 16: The 'Transform data' tab in the Power BI ribbon

This tab opens the Power Query Editor which uses Power Query M Formula Language to perform all data transformations. Power BI has M functions built-in to make it easier for the user to make transformations with little coding knowledge, but the command line can also be used to execute more complicated commands or streamline the execution of a command that is to be made to multiple columns or rows. The command line is shown in Figure 17.



```
= Csv.Document(Web.Contents("https://raw.githubusercontent.com/pnjoslin/safety-data/main/SampleDataAll.csv"),[Delimiter=",",
    Columns=7, Encoding=65001, QuoteStyle=QuoteStyle.None])
```

	Column1	Column2	Column3	Column4	Column5	Column6
1	ID	Latitude	Longitude	Node	Risk Value	Time
2	1	48.1	126.2	Breader	0.2024	1:22:42
3	2	21.28	109.06	Production	0.4033	2:32:52
4	3	43.52	116.02	Transportation	0.2148	3:43:02
5	4	45.3	126.58	Retail	0.2065	4:53:12
6	5	50.1	127.3	Customer	0.77	6:03:22
7	6	35.15	79.55	Production	0.7798	7:13:32
8	7	41.05	80.1	Transportation	0.9062	8:23:42
9	8	47.48	88.1	Retail	0.6666	9:33:52
10	9	38.3	88	Customer	0.6598	10:44:02
11	10	24.25	118.04	Breader	0.1876	11:54:12
12	1	48.1	126.2	Breader	0.2024	1:22:42
13	2	21.28	109.06	Production	0.4033	2:32:52
14	3	43.52	116.02	Transportation	0.2148	3:43:02
15	4	45.3	126.58	Retail	0.2065	4:53:12
16	5	50.1	127.3	Customer	0.77	6:03:22
17	6	35.15	79.55	Production	0.7798	7:13:32
18	7	41.05	80.1	Transportation	0.9062	8:23:42
19	8	47.48	88.1	Retail	0.6666	9:33:52
20	9	38.3	88	Customer	0.6598	10:44:02
21	10	24.25	118.04	Breader	0.1876	11:54:12

Figure 17: The Power Query Editor command line with the M Formula Language code to connect the query to the SampleDataAll.csv GitHub file.

Each data source or grouping of data is considered a query to which transformations can be performed. This dashboard has five main data sources as discussed previously but six queries overall as shown in Figure 18.

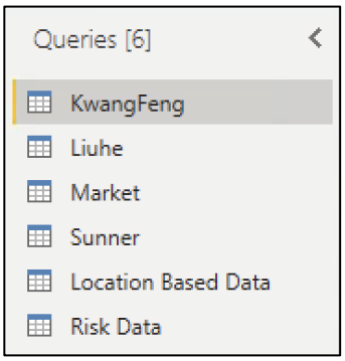


Figure 18: The six queries used to build the Power BI dashboard

The ‘Risk Data’ query pulls from the “SampleDataAll.csv” file in the *safety-data* GitHub repository. When a data source is connected and a query is made, all data is included in the rows of the data table starting with column headers in row 1. To recognize the first row of the data table as the column headers, the ‘Use First Row as Headers’ button is used. Once pressed, a ‘Promoted Headers’ step is added to the ‘Applied Steps’ section in the Query Settings Toolbar as shown in Figure 19.

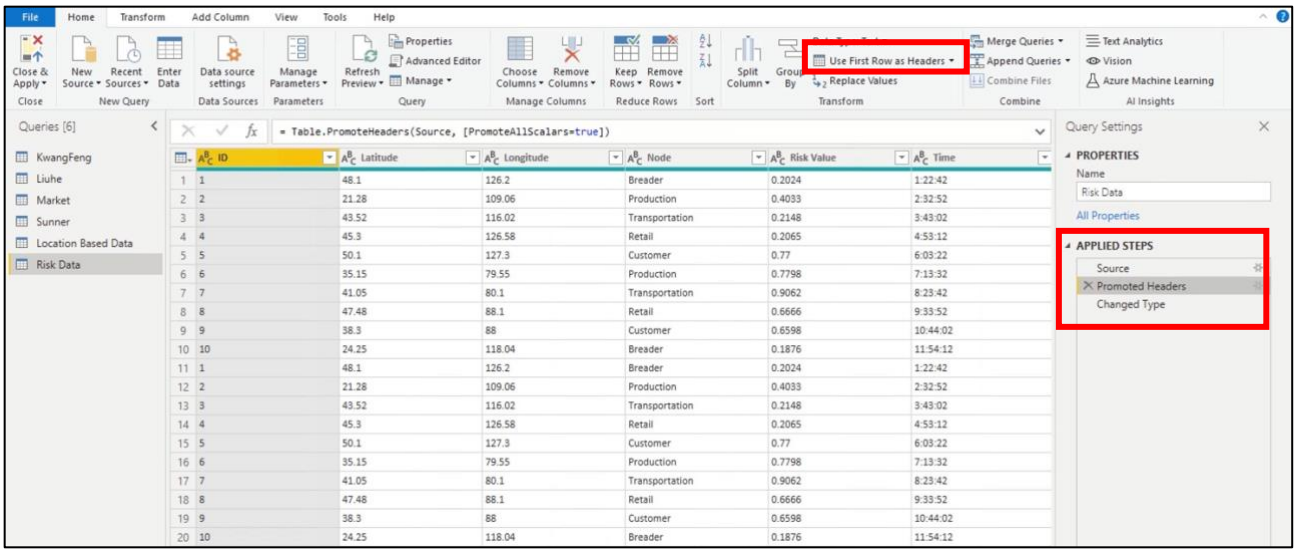


Figure 19: The Promoted Headers step

Power BI automatically interprets the data type of each column, but additional data type changes are needed based on the context of the data in this dashboard. For example, the node 'ID' was kept as a text value in column 1 as these values need to be interpreted as the data labels for each node to make analyzing the data associated with a specific node easier. However, 'Latitude', 'Longitude', and 'Risk Value' were all changed to the numeric data type so that summary statistics for these values can be calculated. The final data types of all columns are found in Figure 20.

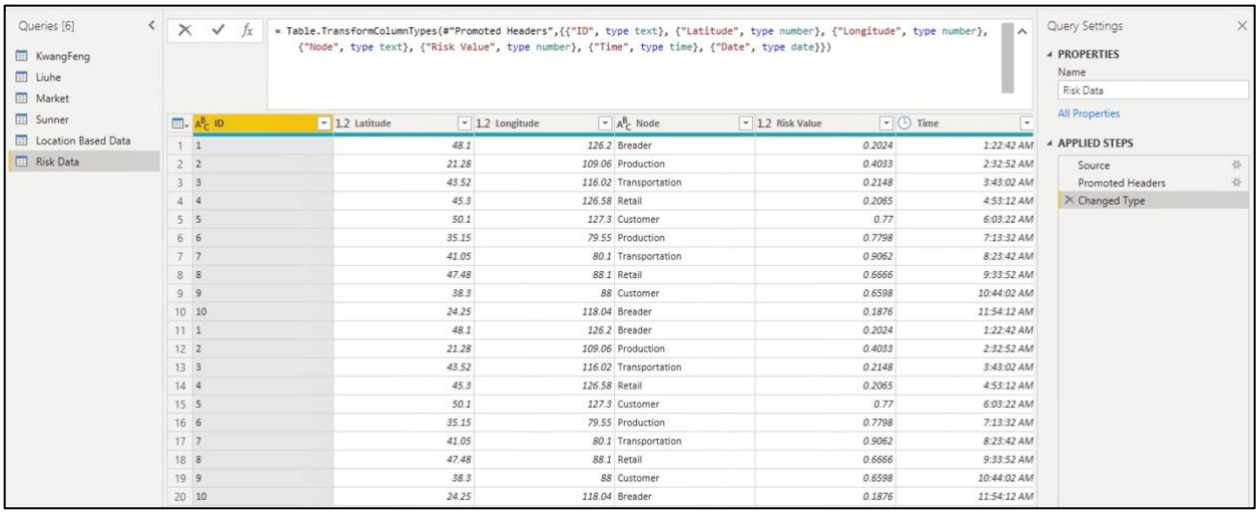


Figure 20: The command line code showing the data types for all columns in the Risk Data query

The data included in the four Sampling Phase Data source files are transformed in a similar process to the Risk Data file. The modification of the KwangFeng query will be used to demonstrate the data transformations on the location data as all transformations performed in this query are identical to the transformations done to the Liuhe, Market and Sunner queries. To begin, the headers are promoted, and the data types are changed in the same fashion as described previously. These data types can be found in Figure 21; however, this command shows the data

types of all columns in the original data file, including the columns that are removed in the next step of the transformation process.

```
= Table.TransformColumnTypes(#"Promoted Headers1",{{"Sampling Date", type number}, {"Season", type text}, {"Broiler breeds", type text}, {"Slaughter volume on the sampling day (birds) ",
    type text}, {"Sampling Phase", type text}, {"Slaughterhouse temperature#(lf) (℃) ", type text}, {"Slaughterhouse humidity#(lf) (%) ", type text}, {"Whether environmental disinfection
    is carried out on the sampling day", type text}, {"Evisceration method#(lf) (mechanical or manual) ", type text}, {"Evisceration  water temp  (℃) ", type text}, {"Chilling water pH",
    type text}, {"Available chlorine concentration#(lf) (chilling phase, ppm) ", type text}, {"Contact surface disinfection frequency (Segment phase, times / day)", type text}, {"Numbers
    of Samples", Int64.Type}, {"Numbers of postive", Int64.Type}, {"Positive rate", Percentage.Type}})
```

Figure 21: Data types for KwangFeng data table columns

Several columns in the original data have null values, so these columns are removed to simplify the data table to only relevant data. These columns are removed using the command shown in Figure 22.

```
= Table.RemoveColumns(#"Changed Type2",{"Season", "Broiler breeds", "Slaughter volume on the sampling day (birds) ",
    "Slaughterhouse temperature#(lf) (℃) ", "Slaughterhouse humidity#(lf) (%) ", "Whether environmental disinfection is
    carried out on the sampling day", "Evisceration method#(lf) (mechanical or manual) ", "Evisceration  water temp  (℃) ",
    "Chilling water pH", "Available chlorine concentration#(lf) (chilling phase, ppm) ", "Contact surface disinfection
    frequency (Segment phase, times / day)"})
```

Figure 22: The M command to remove unnecessary columns from the KwangFeng data table

The date column from the original data file is also formatted in a manner that is not easily recognizable for Power BI slicers and filters when building the dashboard. For example, January 2016 is represented as “2016.01”. To fix this issue, the ‘Split Column by Delimiter’ function is used as shown in Figure 23.

```
= Table.SplitColumn(Table.TransformColumnTypes(#"Added Custom", {{"Sampling Date", type text}}, "en-US"), "Sampling Date",
    Splitter.SplitTextByDelimiter(".", QuoteStyle.Csv), {"Sampling Date.1", "Sampling Date.2"})
```

Figure 23: The M command to split the Sampling Date column

This command resulted in a month column populated with the digits following the ‘.’ Delimiter. A conditional column is then added to translate the numeric month column to the corresponding month names as text values. The command to execute the conditional column is shown in Figure 24.

```
= Table.AddColumn(#"Changed Type3", "Month", each if [Sampling Date.2] = 1 then "January" else if [Sampling Date.2] = 2 then
    "February" else if [Sampling Date.2] = 3 then "March" else if [Sampling Date.2] = 4 then "April" else if [Sampling Date.2]
     = 5 then "May" else if [Sampling Date.2] = 6 then "June" else if [Sampling Date.2] = 7 then "July" else if [Sampling
    Date.2] = 8 then "August" else if [Sampling Date.2] = 9 then "September" else if [Sampling Date.2] = 10 then "October"
    else if [Sampling Date.2] = 11 then "November" else if [Sampling Date.2] = 12 then "December" else null)
```

Figure 24: The M command to add the conditional column for the month

These preliminary transformations are made to all four location queries: KwangFeng, Liuhe, Market, and Sunner. The next step is to append or merge these queries into one query with all Sampling Phase data to limit the number of queries being applied to the dashboard. To append the queries, a custom column named ‘Location’ is added to the individual location queries as a way to identify the origin of the row of data once appended. The commands to create the custom column for KwangFeng is shown in Figure 25.

```
= Table.AddColumn(#"Removed Columns", "Location", each "KwangFeng")
```

Figure 25: The M command to create the custom Location column for the KwangFeng query

After all Location columns are added to the four individual location queries, the single location-based query is made using the 'Append Query as New' function in the Power Query Editor toolbar as shown in Figure 26.

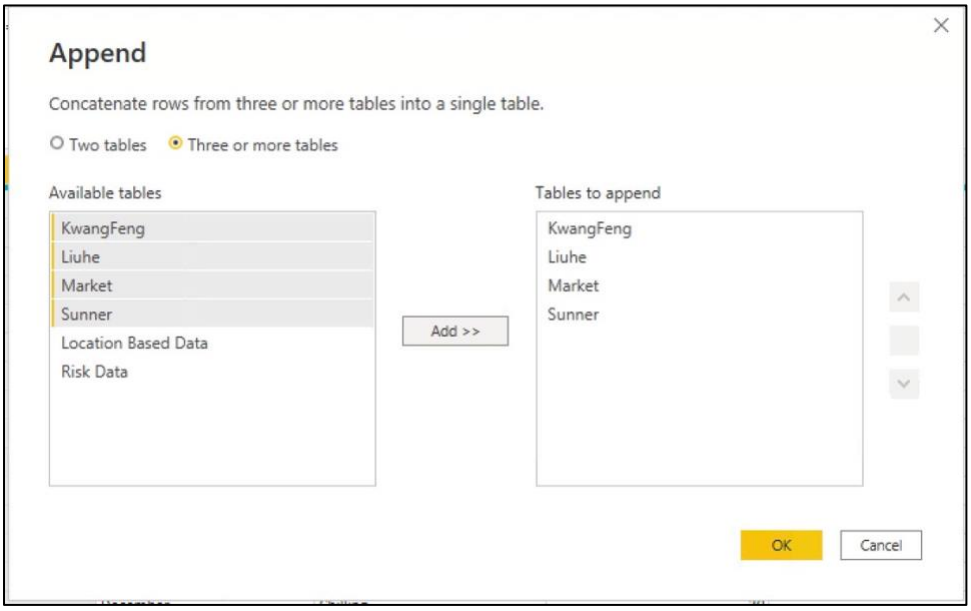


Figure 26: The Append Query as New function that combines the four individual location queries

This function can also be executed from the command line using the "Table.Combine" command as shown in Figure 27.

```
= Table.Combine({KwangFeng, Liuhe, Market, Sunner})
```

Figure 27: The resulting M command to append the individual location queries

When combining the four location queries, there are some data discrepancies. The Market query holds data that does not contain a Sampling Phase field, resulting in null values. To complete the data, the null values are replaced with the string 'Market Phase' using the command shown in Figure 28.

```
= Table.ReplaceValue(#"Removed Columns",null,"Market phase",Replacer.ReplaceValue,{"Sampling Phase"})
```

Figure 28: The M command to replace null values

After replacing the null values, the data transformations are complete, and all changes can be applied to the queries. The resulting entity relationship diagrams are found in the Relationships tab which show each field included in both the Location Based Data and Risk Data relations. This tab and the relations are found in Figure 29.

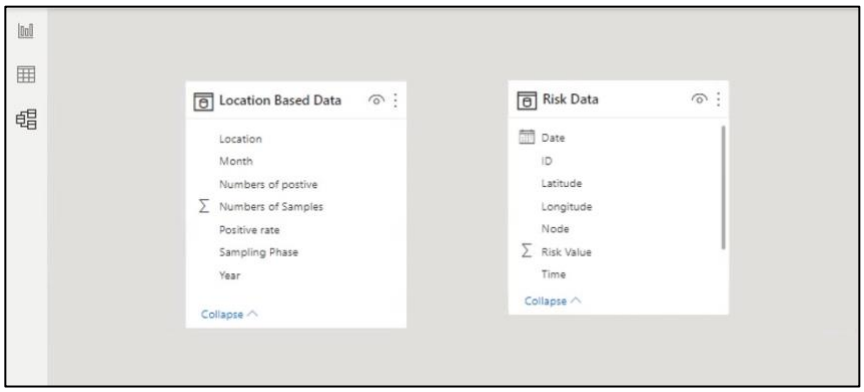


Figure 29: The resulting relations after the data transformations in the Relationships tab in the Power BI dashboard

3.9. Building the Dashboard – Sample Risk Data Page

The purpose of the Sample Risk Data Page is to show the location and corresponding information associated with the supply chain nodes as well as the magnitude of the nodes' latest-date risk values. To determine a node's latest-date risk value, a new measure called 'Latest Risk Value' is created using the New Measure feature of Power BI. The Latest Risk Value measure searches the date column of the risk data to find the max date and returns the risk value associated with that date for a particular node. The command to create the measure is shown in Figure 30.

```
1 Latest risk value = lookupvalue('Risk Data'[Risk Value],'Risk Data'[Date], max('Risk Data'[Date]))
```

Figure 30: The command to create the Latest Risk Value measure

Once the new measure is created, the map showing risk value magnitudes is created. Each node displays information about the latitude, longitude, risk value, node ID, node type, and the time and date of the data entry for the latest risk value. Conditional formatting is applied to each latest risk value to easily signify the value range of a risk value. The value ranges are 0 to 0.10, 0.11 to 0.50, and 0.51 to 0.1 and are colored green, yellow, and red respectively. This data is also found in a table below the map titled "Risk Value and Node Information". The data table and map can be filtered by date and node type as well. The completed Sample Risk Data page of the dashboard is shown in Figure 31.

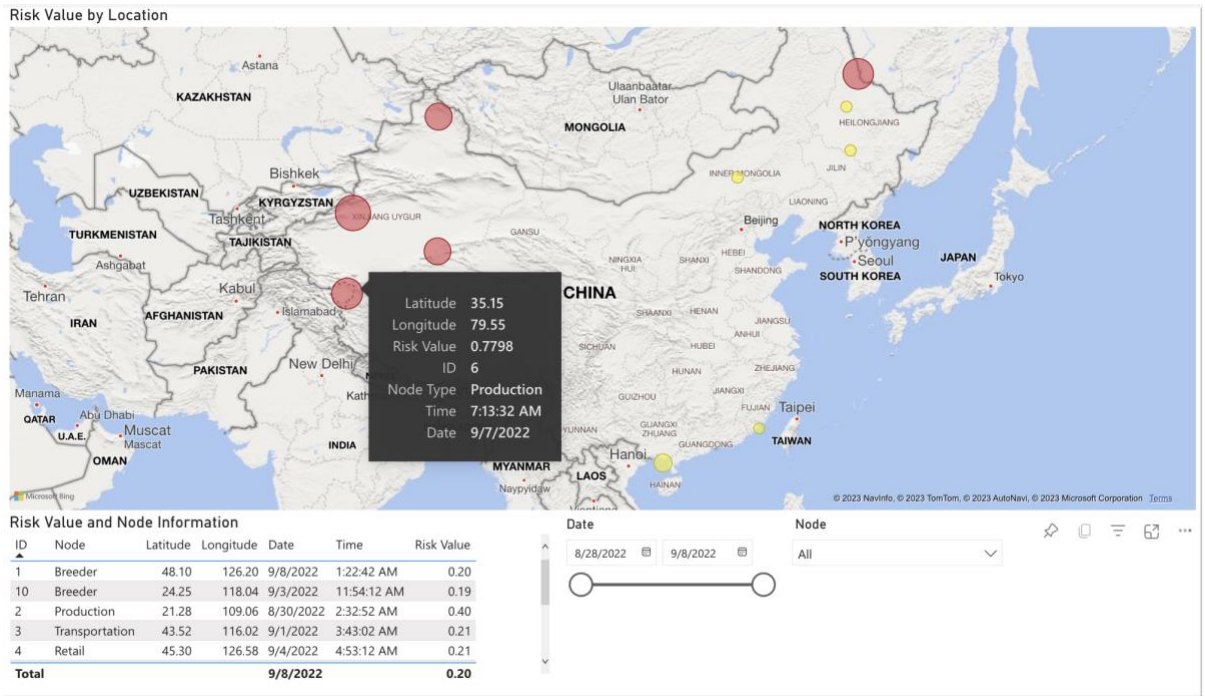


Figure 31: The Sample Risk Data page of the Power BI dashboard

3.10. Building the Dashboard – Sampling Data by Location Page

The purpose of the Sampling Data by Location page of the dashboard is to analyze and display the positive rates of samples from varying sampling phases and locations. The table titled "Positive Sample Data" contains data associated with a sample's positive rate, number of positives, sampling phase, location, month, and year. The bar chart titled "Positive Rate by Location" shows the positive rate for each of the four locations: Market, Liuhe, Sunner, and Kwang Feng. Each location has a specified color for use as an identifier in the stacked bar chart titled "Positive Rate by Sampling Phase and Location". This chart shows the locations' positive rates for each sampling phase, which makes it easier for stakeholders to locate areas of higher positive rates and manage the risk in the supply chain system. All tables and charts included in this page can be filtered by sampling phase, location, month, and year. The completed Sampling Data by Location page is shown in Figure 32.

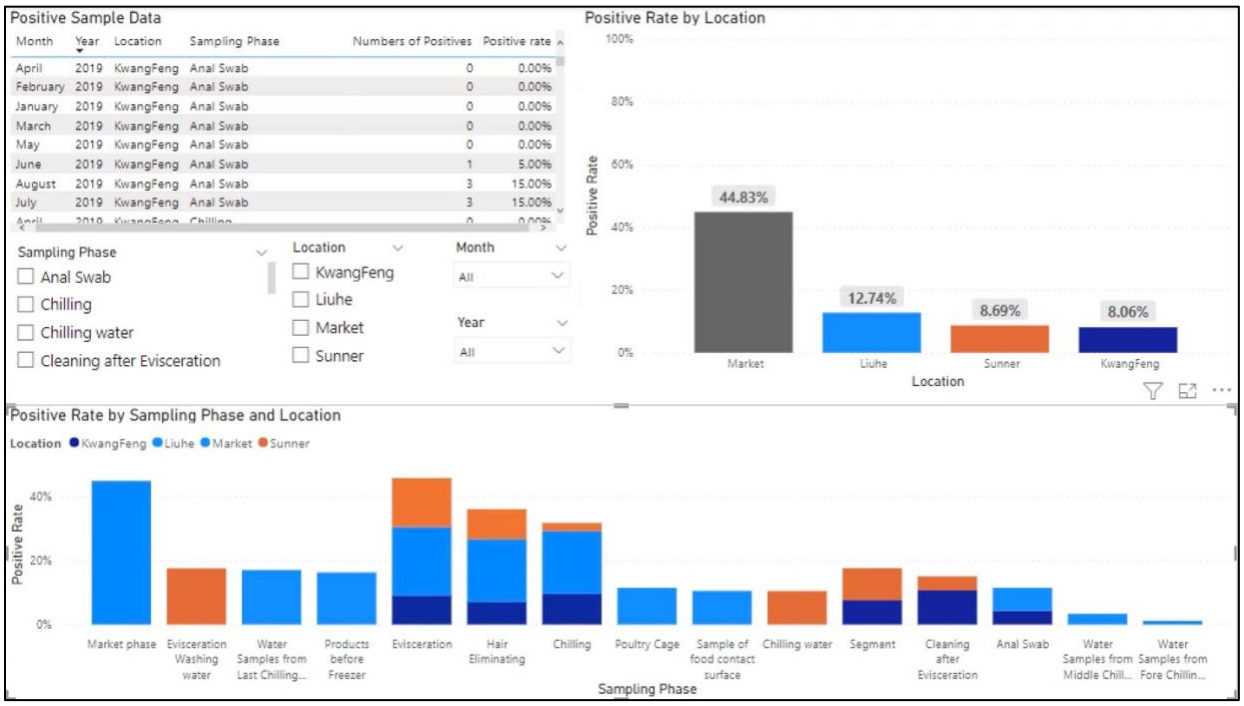


Figure 32: The Sampling Data by Location page of the Power BI dashboard

3.11. Publishing and Updating the Dashboard

After both pages of the dashboard are completed, the dashboard is published to the online version of Power BI using a Power BI Pro license to be shared with other project stakeholders. In the online app, dashboards are stored as a report and dataset. The dataset version is used to schedule data refreshes to automate the dashboard and provide insights on the most relevant data. The data is set to refresh every day at 3 a.m. to be consistent with the HPC data pulls while allowing a buffer of an hour to account for any possible delays in running the cron script. All data in the dashboard can also be exported to an Excel file or PDF to streamline the analysis process.

4. Conclusion

This paper discusses the development of an automated visualization pipeline and the benefits that this type of pipeline poses to key stakeholders and decision makers. HPC software and data visualization programs like Power BI are becoming increasingly more useful in these situations where real-time data analysis capabilities are needed. This research tests the use of an automated data pipeline with the risk assessment data for a specific poultry supply chain in China. However, this process could be applied to larger supply chain analysis projects. This research could also be used for different types of problems in a variety of industries. Decision makers in healthcare, transportation, manufacturing, and other leading industries could greatly benefit from viewing data in an easily digestible manner to make the pathway to effective and strategic solutions more evident.

References

[1] “AHPCC,” *Home | AHPCC | University of Arkansas*. [Online]. Available: https://hpc.uark.edu/.

[2] “Arkansas High Performace Computing Center [hpcwiki],” *[Arkansas High Performace Computing Center [hpcwiki]]*. [Online]. Available: https://hpcwiki.uark.edu/doku.php.

[3] “Data Visualization: Microsoft power bi,” *Data Visualization | Microsoft Power BI*. [Online]. Available: https://powerbi.microsoft.com/en-us/.

[4] “GitHub.com help documentation,” *GitHub Docs*. [Online]. Available: https://docs.github.com/en.

[5] “Poultry excellence in China,” *Poultry Excellence in China*. [Online]. Available: http://en.poultryexcellence.com/.