Contact Details

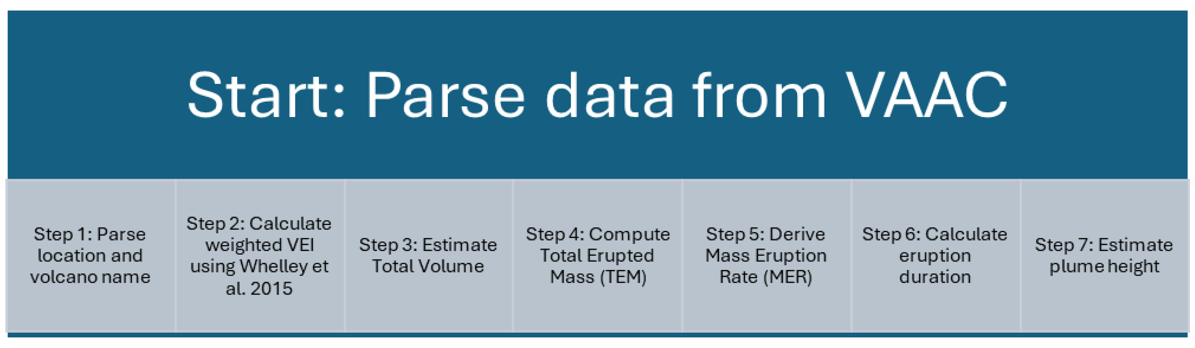
Dr. Mahmud Muhammad
(PhD, MSc, and BSc in Geology)
Email: mahmud.geology@hotmail.com
Website: mahmudm.com

Objective

The goal of this project is to develop a suite of probability density functions (PDFs) that describe Eruption Source Parameter (ESP) inputs required by the NAME model. These estimations are crucial for accurate volcanic ash dispersion modeling.

Key Tasks:

1. **Automated Retrieval and Parsing of Volcanic Advisory Reports (VAA):**
   - Implement a system to automatically retrieve and parse real-time Volcanic Advisory Reports (VAA) from the Tokyo Volcanic Ash Advisory Center (TVAAC).
2. **Annual VEI Probability Parsing:**
   - Utilize the methodology outlined in Whelley et al. (2015) to extract annual Volcanic Explosivity Index (VEI) probabilities for the volcano of concern using data from VAA reports.
3. **Bayesian Sampling for Total Eruption Mass (TEM):**
   - Estimate the range of Total Eruption Mass (TEM) for a known VEI level using Bayesian probability sampling. This estimation will account for a range of tephra particl densities.
4. **Mass Eruption Rate (MER) Estimation:**
   - Calculate the Mass Eruption Rate (MER) based on the derived Total Eruption Mass (TEM).
5. **Eruption Duration Calculation:**
   - Derive the eruption duration by integrating the MER and TEM values.
6. **Plume Height Estimation from MER:**
   - Estimate the volcanic plume height using the Mass Eruption Rate (MER).
7. **MER Estimation from Plume Height:**
   - Estimate the MER using observed Plume Height.
8. **Estimation of Mass fraction of ash particle size below 63 microns:**
   - Estimate the Mass fraction of ash particle size below 63 microns using the Mass Eruption Rate (MER).



Parsing of Volcanic Advisory Reports (VAA)

In [2]:
```python
# Using Whelley et al., 2015  paper that calculated VEI Level eruption for 750 volcanoes in southeast Asia we start estimating Eruption source paramerters from VEI and probablistic volume estimates.

# first we load Whelley 2015 Volcano Database later will be use to filter Volcanish Ash Advisory reports from Tokyo Volcanic Ash Advisory Center in realtime (computer need internet connection to parse VAA reports)

import vei

data_loader=vei.LOADDATA()


whelley_data=data_loader.whelley_2015(as_geodataframe=True)
```

In [3]:
```python
# import Libraries and classes required to parse VAA reports and filter dataset for later use.

import vep_vaa_text
from vep_vaa_text import VEP_TVAAC_VAAText
scraper = VEP_TVAAC_VAAText()
scraper.fetch_webpage()
tables = scraper.extract_all_tables()
tables
# Perform a search Note: for realtime operation set all parameters to None to retrive the latest report VAA report
search_results = scraper.search(query='SINABUNG', date_time=None, advisory_number='2020')  # Replace with your query string, set all parameters to None to retrive latest report dates: Note takes longer processing time
downloaded_VAA_report, latest_VAA_report , latest_date_data=scraper.download_vaa_text(output_dir="./vaa_texts_2", filtered_results=search_results, csv=True, gdf=whelley_data)

downloaded_VAA_report.head(2)
```

```
CSV file created: ./vaa_texts_2\vaa_texts.csv
```

Out[3]:

	DTG	VAAC	VOLCANO	VOLCANO CODE	PSN	AREA	SUMMIT ELEV	ADVISORY NR	INFO SOURCE	AVIATION COLOUR CODE	...	FCST VA CLD +6 HR	FCST VA CLD +12 HR	FCST VA CLD +18 HR	RMK	NXT ADVISORY	Latitude	Longitude	ADVISORY_YEAR	REPORT_NUMBER	DTG_DATETIME
0	20200810/1635Z	DARWIN	SINABUNG	261080	N0310 E09824	INDONESIA	2460M	2020/12	HIMAWARI-8, CVGHM, PIREP, WEBCAM	RED	...	10/2235Z NO VA EXP	11/0435Z NO VA EXP	11/1035Z NO VA EXP	VA TO FL320 LAST PARTIALLY OBS ON SAT IMAGERY ...	NO LATER THAN 20200810/2235Z=	3.166667	98.4	2020	12	2020-08-10 16:35:00
1	20200810/1030Z	DARWIN	SINABUNG	261080	N0310 E09824	INDONESIA	2460M	2020/10	HIMAWARI-8, CVGHM, PIREP, WEBCAM	RED	...	10/1630Z SFC/FL140 N0303 E09852 - N0337	10/2230Z SFC/FL140 N0253 E10049 - N0348	11/0430Z NO VA EXP	ERUPTION HAS CEASED. VA TO FL320 LAST PARTIALL...	NO LATER THAN 20200810/1630Z=	3.166667	98.4	2020	10	2020-08-10 10:30:00

2 rows × 23 columns

In [4]:
```python
latest_date_data
```

Out[4]:

	DTG	VAAC	VOLCANO	VOLCANO CODE	PSN	AREA	SUMMIT ELEV	ADVISORY NR	INFO SOURCE	AVIATION COLOUR CODE	...	FCST VA CLD +12 HR	FCST VA CLD +18 HR	RMK	NXT ADVISORY	Latitude	Longitude	ADVISORY_YEAR	REPORT_NUMBER	DTG_DATETIME	Altitude_Meters
0	20200810/1635Z	DARWIN	SINABUNG	261080	N0310 E09824	INDONESIA	2460M	2020/12	HIMAWARI-8, CVGHM, PIREP, WEBCAM	RED	...	11/0435Z NO VA EXP	11/1035Z NO VA EXP	VA TO FL320 LAST PARTIALLY OBS ON SAT IMAGERY ...	NO LATER THAN 20200810/2235Z=	3.09	9.92	2020	12	2020-08-10 16:35:00	4267.2

1 rows × 24 columns

In [5]:
```python
# Retrive data from the latest report date
from rich.console import Console
from rich.table import Table

# Initialize the console
console = Console()

# Create a table for a professional and structured output
table = Table(title="Volcanic Activity Observations", show_lines=True, title_style="bold blue")

# Define table columns
table.add_column("Parameter", style="dim", width=25)
table.add_column("Value", style="bold white", justify="center")

# Populate the table with data, if available
if 'VOLCANO' in latest_date_data.columns:
    volcano_name = latest_date_data['VOLCANO'].to_list()[0]
    table.add_row("Site of Volcanic Activity", volcano_name)

if 'Altitude_Meters' in latest_date_data.columns:
    observed_ash_altitude_meter = latest_date_data['Altitude_Meters'].to_list()[0]
    table.add_row("Observed Ash Altitude (m ASL)", str(observed_ash_altitude_meter))

if 'Latitude' in latest_date_data.columns:
    latitude = latest_date_data['Latitude'].to_list()[0]
    table.add_row("Latitude", str(latitude))

if 'Longitude' in latest_date_data.columns:
    longitude = latest_date_data['Longitude'].to_list()[0]
    table.add_row("Longitude", str(longitude))



# Call the search_whelley_2015 method on the instance
search_results, vei_range = data_loader.search_whelley_2015(Volcano=volcano_name, max_vei_returns=None)

# View the search results
#print(f"Potential VEI Levels for  Volcano site {volcano_name} is based on Whelley et al. 2015 paper   {vei_range}")
#print(search_results)

table.add_row(f"Possible VEI Levels for  Volcano  {volcano_name}  based on Whelley et al. 2015 paper", str(vei_range))

# Render the table
console.print(table)

import pandas as pd
import plotly.graph_objects as go

search_results = pd.DataFrame(search_results)

# Create a table
fig = go.Figure(data=[go.Table(
    header=dict(values=list(search_results.columns),
                fill_color='paleturquoise',
                align='left'),
    cells=dict(values=[search_results[col] for col in search_results.columns],
               fill_color='lavender',
               align='left'))
])

# Add a bold title
fig.update_layout(
    title=dict(
        text=f"Search Results for Whelley et al. 2015 show possible VEI activity for the volcano {volcano_name}",
        x=0.5,  # Center the title
        font=dict(size=20, family="Arial Black", color="black")  # Use bold font family
    )
)

fig.show()
```

```
Top VEI values returned: [2, 3, 4, 5]
```

Volcanic Activity Observations

Parameter	Value
Site of Volcanic Activity	SINABUNG
Observed Ash Altitude (m ASL)	4267.2
Latitude	3.09
Longitude	9.92
Possible VEI Levels for Volcano SINABUNG based on Whelley et al. 2015 paper	[2, 3, 4, 5]

Search Results for Whelley et al. 2015 show possible VEI activity for the volcano SINABUNG

Volcano Number	GVP Volcano Number	Volcano	Other name or Sub-Feature	Latitude	Longitude	Elevation (m)	Classification	Eruption history source	VEI 2	VEI 3	VEI 4	VEI 5	VEI 6	VEI 7	VEI 8
3	261080	SINABUNG	null	3.166666	98.391666	2460	Semi-plugged stratocone	Class	0.022451791	0.002424242	0.001443001	0.0000909091	0	0	0

Step 1: Define Total Eruption Mass (TEM) from VEI

This cell initializes the `VEI_BulkVolume_Mass` class to perform probabilistic calculations of total eruption mass based on known VEI values.

Key Parameters:

- `use_default_densities=True` : Applies default density values for tephra deposits.
- `density_min=800` and `density_max=1700` (kg/m^3): Define the density range for volcanic materials.
- `num_samples=1000` : Specifies the number of Monte Carlo samples for probabilistic calculations.

Outcome:

The `vei_toMass` object enables generating probabilistic TEM values for subsequent calculations.

In [25]:
```python
from vei import VEI_BulkVolume_Mass

# Initialize the class
```

```python
vei_toMass = VEI_BulkVolume_Mass(use_default_densities=False, density_min=800, density_max=1200, num_samples=10000000)


# Generate probabilistic volumes
vei_toMass.generate_probabilistic_volumes()

# Calculate masses using probabilistic volumes
vei_toMass.calculate_mass(calculate_emperical=True, calculate_probabilistic=True)

vei_toMass.calculate_percentile_bands()



# # Generate summary statistics
vei_toMass.generate_summary_statistics()

# # Visualize the results
vei_toMass.visualize_statistics()

# # Calculate and plot percentile bands

vei_toMass.plot_percentile_bands()

# # Export results
#vei_toMass.export_statistics(filename="summary_statistics.csv")
vei_toMass.export_volumes_and_masses()

vei_toMass.export_data()

#vei_toMass.save_outputs_to_pdf()
```

```
Generating probabilistic volumes...
Probabilistic volumes successfully generated.
Best density sampling strategy: normal
### Volume Percentiles ###
  VEI  emp_Volume_5th  Uncertainty_emp_Volume_5th  Prob_Volume_5th  \
0   0       2300000.0                    230000.0     2.528941e+07
1   1       2300000.0                    230000.0     2.528941e+07
2   2       2300000.0                    230000.0     2.528941e+07
3   3       2300000.0                    230000.0     2.528941e+07
4   4       2300000.0                    230000.0     2.528941e+07
5   5       2300000.0                    230000.0     2.528941e+07
6   6       2300000.0                    230000.0     2.528941e+07
7   7       2300000.0                    230000.0     2.528941e+07
8   8       2300000.0                    230000.0     2.528941e+07

   Uncertainty_Prob_Volume_5th  emp_Volume_25th  Uncertainty_emp_Volume_25th  \
0                 2.528941e+06      100000000.0                   10000000.0
1                 2.528941e+06      100000000.0                   10000000.0
2                 2.528941e+06      100000000.0                   10000000.0
3                 2.528941e+06      100000000.0                   10000000.0
4                 2.528941e+06      100000000.0                   10000000.0
5                 2.528941e+06      100000000.0                   10000000.0
6                 2.528941e+06      100000000.0                   10000000.0
7                 2.528941e+06      100000000.0                   10000000.0
8                 2.528941e+06      100000000.0                   10000000.0

   Prob_Volume_25th  Uncertainty_Prob_Volume_25th  emp_Volume_50th  ...  \
0      5.498412e+08                  5.498412e+07     3.000000e+10  ...
1      5.498412e+08                  5.498412e+07     3.000000e+10  ...
2      5.498412e+08                  5.498412e+07     3.000000e+10  ...
3      5.498412e+08                  5.498412e+07     3.000000e+10  ...
4      5.498412e+08                  5.498412e+07     3.000000e+10  ...
5      5.498412e+08                  5.498412e+07     3.000000e+10  ...
6      5.498412e+08                  5.498412e+07     3.000000e+10  ...
7      5.498412e+08                  5.498412e+07     3.000000e+10  ...
8      5.498412e+08                  5.498412e+07     3.000000e+10  ...

   Prob_Volume_50th  Uncertainty_Prob_Volume_50th  emp_Volume_75th  \
0      5.496897e+10                  5.496897e+09     5.000000e+11
1      5.496897e+10                  5.496897e+09     5.000000e+11
2      5.496897e+10                  5.496897e+09     5.000000e+11
3      5.496897e+10                  5.496897e+09     5.000000e+11
4      5.496897e+10                  5.496897e+09     5.000000e+11
5      5.496897e+10                  5.496897e+09     5.000000e+11
6      5.496897e+10                  5.496897e+09     5.000000e+11
7      5.496897e+10                  5.496897e+09     5.000000e+11
8      5.496897e+10                  5.496897e+09     5.000000e+11

   Uncertainty_emp_Volume_75th  Prob_Volume_75th  \
0                 5.000000e+10      5.499184e+12
1                 5.000000e+10      5.499184e+12
2                 5.000000e+10      5.499184e+12
3                 5.000000e+10      5.499184e+12
4                 5.000000e+10      5.499184e+12
5                 5.000000e+10      5.499184e+12
6                 5.000000e+10      5.499184e+12
7                 5.000000e+10      5.499184e+12
8                 5.000000e+10      5.499184e+12

   Uncertainty_Prob_Volume_75th  emp_Volume_95th  Uncertainty_emp_Volume_95th  \
0                  5.499184e+11     1.900000e+12                 1.900000e+11
1                  5.499184e+11     1.900000e+12                 1.900000e+11
2                  5.499184e+11     1.900000e+12                 1.900000e+11
3                  5.499184e+11     1.900000e+12                 1.900000e+11
4                  5.499184e+11     1.900000e+12                 1.900000e+11
5                  5.499184e+11     1.900000e+12                 1.900000e+11
6                  5.499184e+11     1.900000e+12                 1.900000e+11
7                  5.499184e+11     1.900000e+12                 1.900000e+11
8                  5.499184e+11     1.900000e+12                 1.900000e+11

   Prob_Volume_95th  Uncertainty_Prob_Volume_95th
0      3.519900e+14                  3.519900e+13
1      3.519900e+14                  3.519900e+13
2      3.519900e+14                  3.519900e+13
3      3.519900e+14                  3.519900e+13
4      3.519900e+14                  3.519900e+13
5      3.519900e+14                  3.519900e+13
6      3.519900e+14                  3.519900e+13
7      3.519900e+14                  3.519900e+13
8      3.519900e+14                  3.519900e+13

[9 rows x 21 columns]
### Mass Percentiles ###
  VEI      Mass_5th     Mass_25th     Mass_50th     Mass_75th     Mass_95th  \
0   0  4.323678e+09  5.296240e+09  5.973196e+09  6.649864e+09  7.623536e+09
1   1  4.318758e+10  5.290707e+10  5.966745e+10  6.643474e+10  7.615098e+10
2   2  4.320693e+11  5.293171e+11  5.969547e+11  6.646344e+11  7.618763e+11
3   3  4.321326e+12  5.294734e+12  5.970550e+12  6.647372e+12  7.621213e+12
4   4  4.318416e+13  5.290895e+13  5.966591e+13  6.642700e+13  7.615413e+13
5   5  4.319608e+14  5.293744e+14  5.970898e+14  6.647567e+14  7.621085e+14
6   6  4.320823e+15  5.293540e+15  5.969779e+15  6.646075e+15  7.619693e+15
7   7  4.317606e+16  5.291018e+16  5.966578e+16  6.642399e+16  7.614434e+16
8   8  4.320384e+17  5.293882e+17  5.970276e+17  6.646589e+17  7.618751e+17

   90%_interpercentile_Mass_Estimate
0                       3.299858e+09
1                       3.296340e+10
2                       3.298070e+11
3                       3.299887e+12
4                       3.296997e+13
5                       3.301477e+14
6                       3.298870e+15
7                       3.296828e+16
8                       3.298367e+17

### Generating Summary Statistics Grouped by VEI Level ###
Ensuring VEI is a Series...
Flattening and cleaning VEI column...
Unique VEI values after cleaning: ['0' '1' '2' '3' '4' '5' '6' '7' '8']
Performing groupby operation...

### Summary Statistics Grouped by VEI ###
```

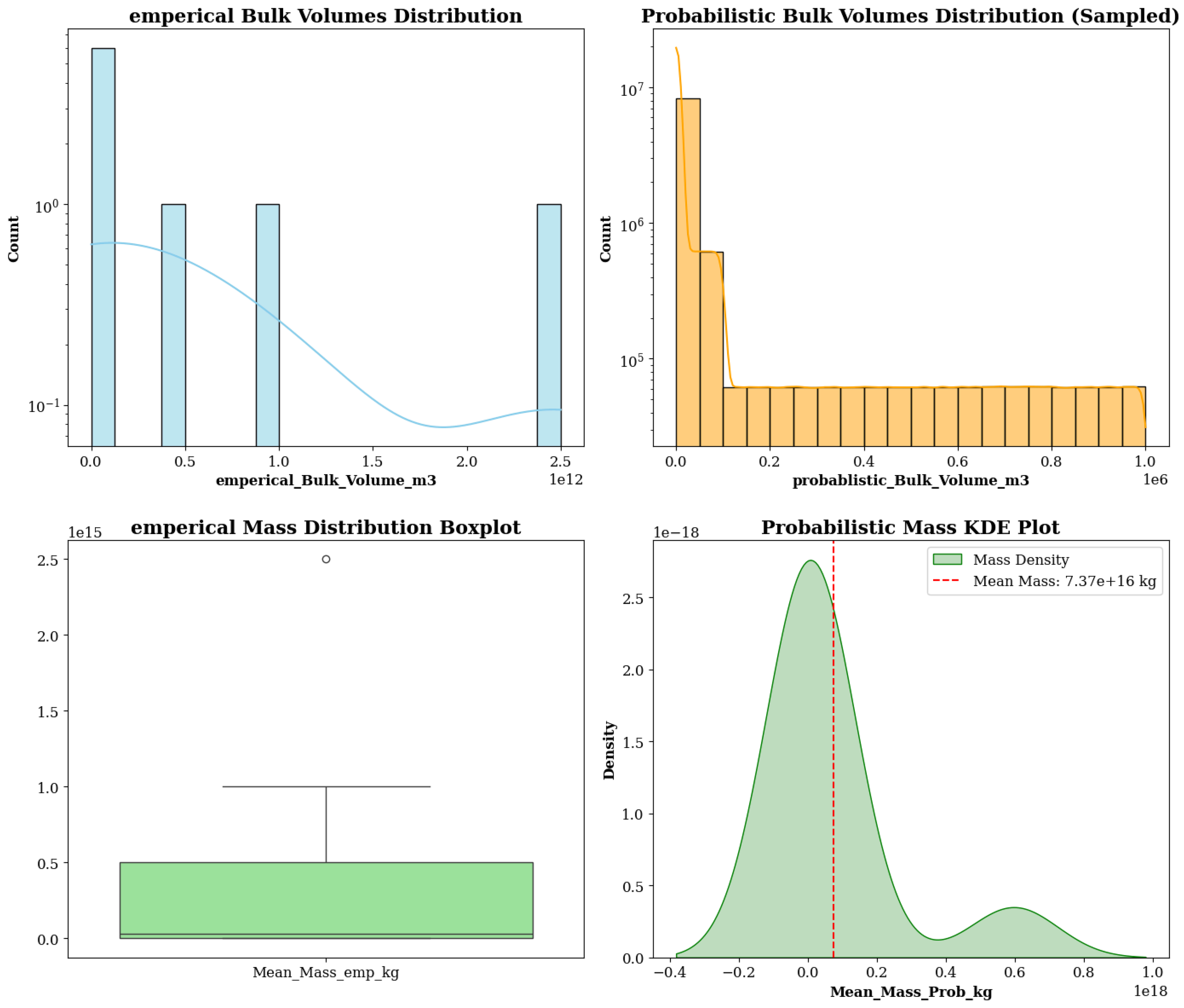
VEI_	emperical_Bulk_Volume_m3_mean	emperical_Bulk_Volume_m3_std	emperical_Bulk_Volume_m3_median	Probabilistic_Bulk_Volume_m3_mean	Probabilistic_Bulk_Volume_m3_std	Probabilistic_Bulk_Volume_m3_median	Mean_Mass_emp_kg_mean	Mean_Mass_emp_kg_std	Mean_Mass_emp_kg_median	Mean_Mass_Prob_kg_mean	Mean_Mass_Prob_kg_std	Mean_Mass_Prob_kg_median
0	500000	nan	500000	5.50227e+06	nan	5.50227e+06	4.99982e+08	nan	4.99982e+08	5.97306e+09	nan	5.97306e+09
1	5e+06	nan	5e+06	5.49701e+07	nan	5.49701e+07	4.99988e+09	nan	4.99988e+09	5.96691e+10	nan	5.96691e+10
2	1e+08	nan	1e+08	5.49841e+08	nan	5.49841e+08	9.99968e+10	nan	9.99968e+10	5.96962e+11	nan	5.96962e+11
3	3e+09	nan	3e+09	5.50081e+09	nan	5.50081e+09	2.99985e+12	nan	2.99985e+12	5.97108e+12	nan	5.97108e+12
4	3e+10	nan	3e+10	5.4969e+10	nan	5.4969e+10	3.00003e+13	nan	3.00003e+13	5.96677e+13	nan	5.96677e+13
5	1e+11	nan	1e+11	5.50046e+11	nan	5.50046e+11	1.00003e+14	nan	1.00003e+14	5.97059e+14	nan	5.97059e+14
6	5e+11	nan	5e+11	5.49918e+12	nan	5.49918e+12	5.00004e+14	nan	5.00004e+14	5.96994e+15	nan	5.96994e+15
7	1e+12	nan	1e+12	5.49655e+13	nan	5.49655e+13	9.99991e+14	nan	9.99991e+14	5.96664e+16	nan	5.96664e+16
8	2.5e+12	nan	2.5e+12	5.50006e+14	nan	5.50006e+14	2.50005e+15	nan	2.50005e+15	5.9702e+17	nan	5.9702e+17

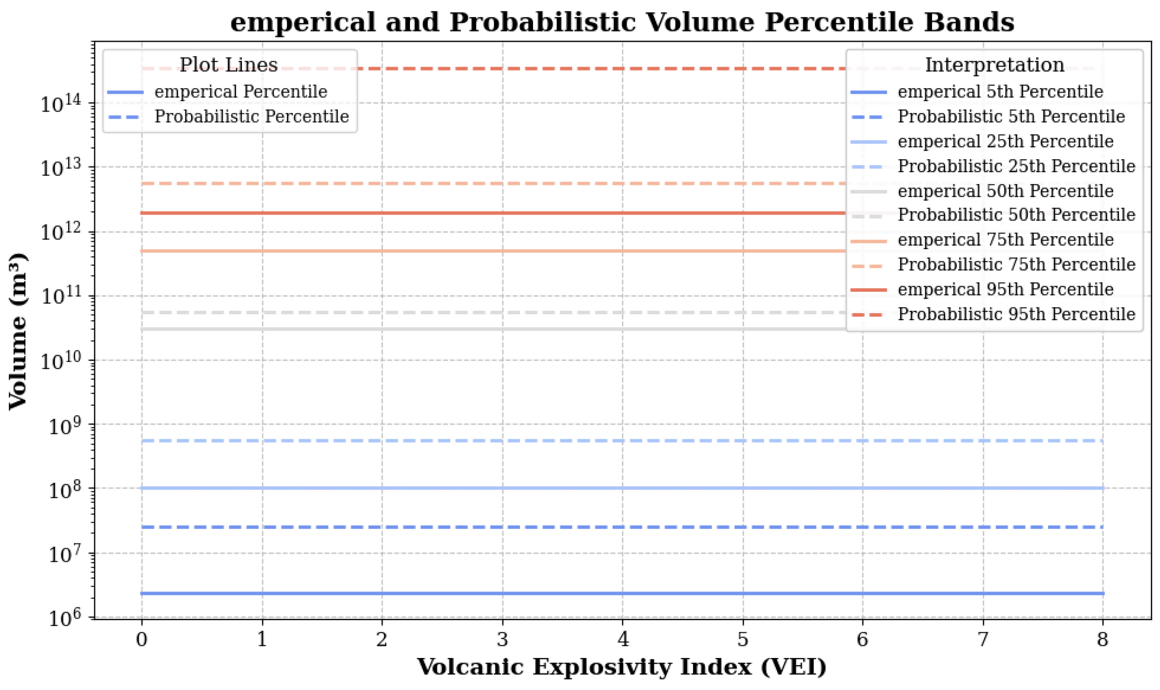
```
### Overall Dataset Statistics ###
```

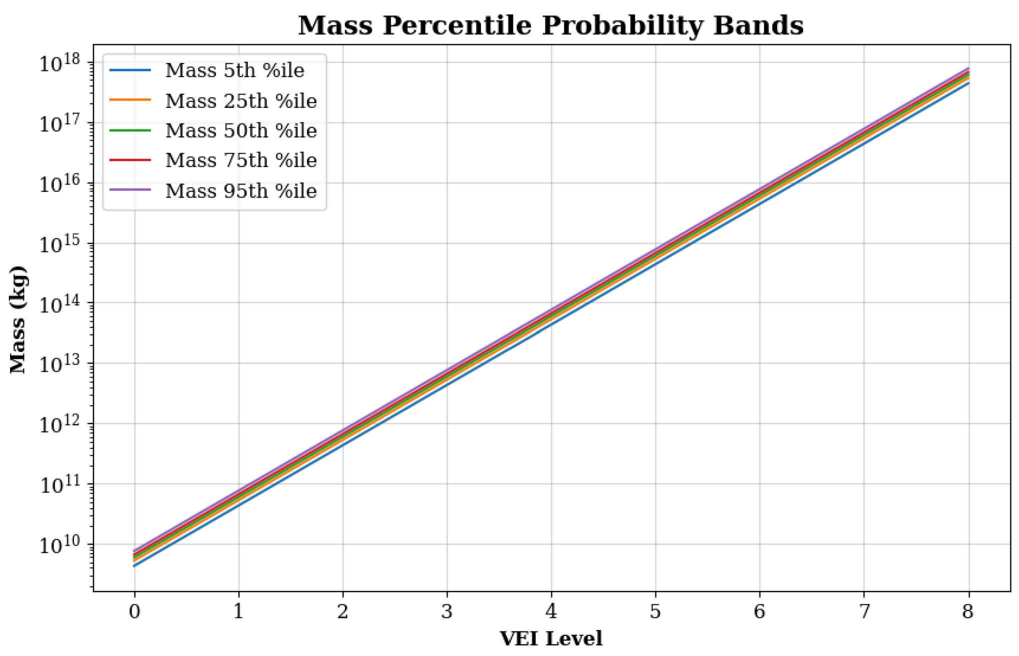
	Bulk_Volume_km3	emperical_Bulk_Volume_m3	Probabilistic_Bulk_Volume_m3	Mean_Mass_emp_kg	Std_Mass_emp_kg	Mean_Mass_Prob_kg	Std_Mass_Prob_kg	emp_Volume_5th	Uncertainty_emp_Volume_5th	Prob_Volume_5th	Uncertainty_Prob_Volume_5th	emp_Volume_25th	Uncertainty_emp_Volume_25th	Prob_Volume_25th	Uncertainty_Prob_Volume_25th	emp_Volume_50th
count	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
mean	459.234	4.59234e+11	6.7898e+13	4.59239e+14	5.30198e+13	7.37022e+16	1.23764e+16	2.3e+06	230000	2.52894e+07	2.52894e+06	1e+08	1e+07	5.49841e+08	5.49841e+07	3e+10
std	837.619	8.37619e+11	1.81682e+14	8.37632e+14	9.66937e+13	1.97212e+17	3.3116e+16	0	0	3.95127e-09	0	0	0	0	7.90253e-09	0
min	0.0005	500000	5.50227e+06	4.99982e+08	5.77406e+07	5.97306e+09	1.00317e+09	2.3e+06	230000	2.52894e+07	2.52894e+06	1e+08	1e+07	5.49841e+08	5.49841e+07	3e+10
25%	0.1	1e+08	5.49841e+08	9.99968e+10	1.15475e+10	5.96962e+11	1.00256e+11	2.3e+06	230000	2.52894e+07	2.52894e+06	1e+08	1e+07	5.49841e+08	5.49841e+07	3e+10
50%	30	3e+10	5.4969e+10	3.00003e+13	3.46484e+12	5.96677e+13	1.0021e+13	2.3e+06	230000	2.52894e+07	2.52894e+06	1e+08	1e+07	5.49841e+08	5.49841e+07	3e+10
75%	500	5e+11	5.49918e+12	5.00004e+14	5.77352e+13	5.96994e+15	1.00278e+15	2.3e+06	230000	2.52894e+07	2.52894e+06	1e+08	1e+07	5.49841e+08	5.49841e+07	3e+10
max	2500	2.5e+12	5.50006e+14	2.50005e+15	2.88584e+14	5.9702e+17	1.00252e+17	2.3e+06	230000	2.52894e+07	2.52894e+06	1e+08	1e+07	5.49841e+08	5.49841e+07	3e+10

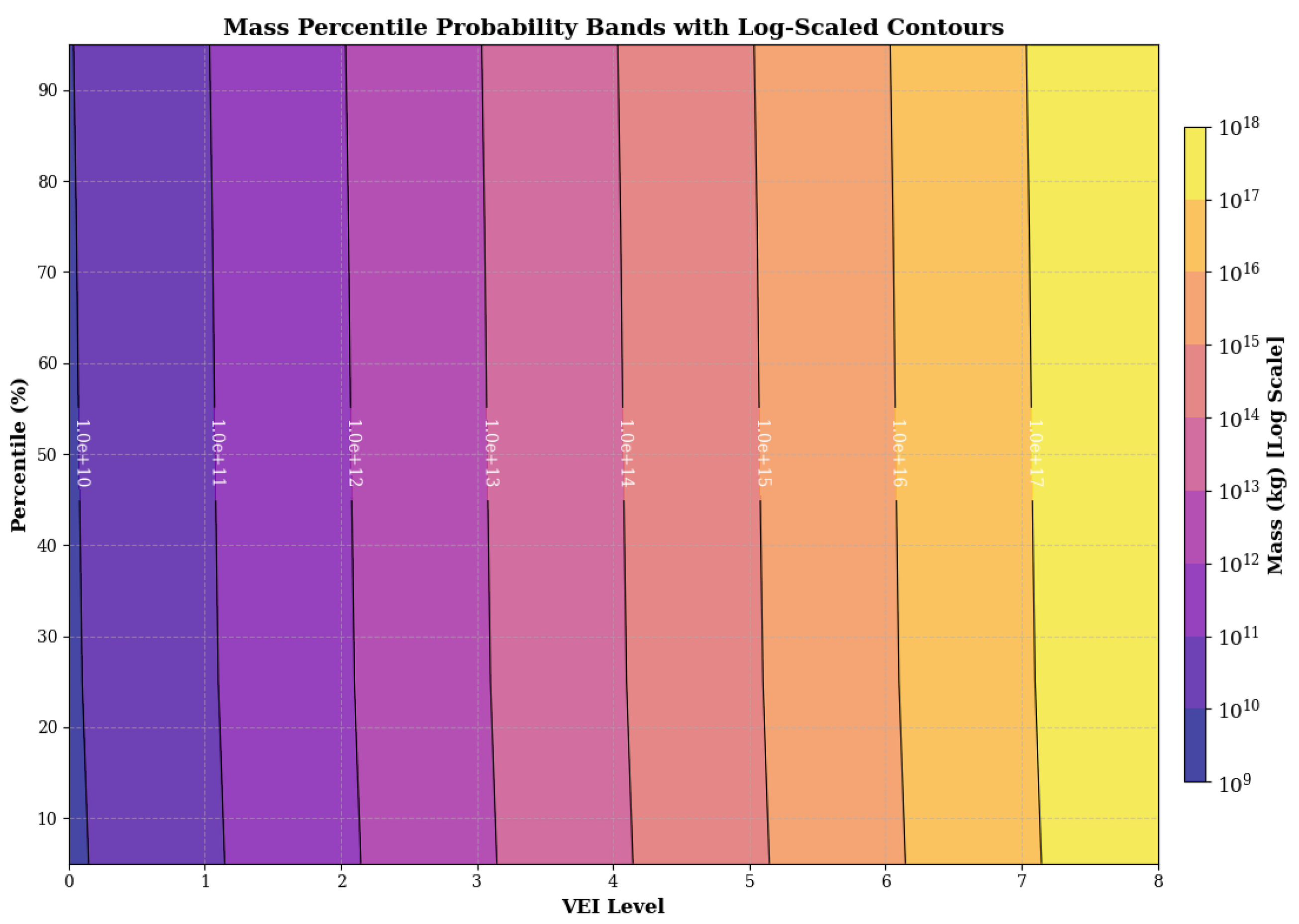
	Uncertainty_emp_Volume_50th	Prob_Volume_50th	Uncertainty_Prob_Volume_50th	emp_Volume_75th	Uncertainty_emp_Volume_75th	Prob_Volume_75th	Uncertainty_Prob_Volume_75th	emp_Volume_95th	Uncertainty_emp_Volume_95th	Prob_Volume_95th	Uncertainty_Prob_Volume_95th	Mass_5th	Mass_25th	Mass_50th	Mass_75th	Mass_95th	90%_interpercentile_Mass_Estimate
count	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
mean	3e+09	5.4969e+10	5.4969e+09	5e+11	5e+10	5.49918e+12	5.49918e+11	1.9e+12	1.9e+11	3.5199e+14	3.5199e+13	5.3335e+16	6.53533e+16	7.37029e+16	8.20519e+16	9.4054e+16	4.0719e+16
std	0	8.09219e-06	0	0	0	0	0	0.00025895	0	0.0662913	0	1.42714e+17	1.74871e+17	1.97215e+17	2.19555e+17	2.51668e+17	1.08954e+17
min	3e+09	5.4969e+10	5.4969e+09	5e+11	5e+10	5.49918e+12	5.49918e+11	1.9e+12	1.9e+11	3.5199e+14	3.5199e+13	4.32368e+09	5.29624e+09	5.9732e+09	6.64986e+09	7.62354e+09	3.29986e+09
25%	3e+09	5.4969e+10	5.4969e+09	5e+11	5e+10	5.49918e+12	5.49918e+11	1.9e+12	1.9e+11	3.5199e+14	3.5199e+13	4.32069e+11	5.29317e+11	5.96955e+11	6.64634e+11	7.61876e+11	3.29807e+11
50%	3e+09	5.4969e+10	5.4969e+09	5e+11	5e+10	5.49918e+12	5.49918e+11	1.9e+12	1.9e+11	3.5199e+14	3.5199e+13	4.31842e+13	5.2909e+13	5.96659e+13	6.6427e+13	7.61541e+13	3.297e+13
75%	3e+09	5.4969e+10	5.4969e+09	5e+11	5e+10	5.49918e+12	5.49918e+11	1.9e+12	1.9e+11	3.5199e+14	3.5199e+13	4.32082e+15	5.29354e+15	5.96978e+15	6.64608e+15	7.61969e+15	3.29887e+15
max	3e+09	5.4969e+10	5.4969e+09	5e+11	5e+10	5.49918e+12	5.49918e+11	1.9e+12	1.9e+11	3.5199e+14	3.5199e+13	4.32038e+17	5.29388e+17	5.97028e+17	6.64659e+17	7.61875e+17	3.29837e+17

```
c:\Users\mahmud\OneDrive\Singapore_project\data\Modules\phase_1\vei.py:789: UserWarning:

Mismatched number of handles and labels: len(handles) = 0 len(labels) = 4
```










```
Volume data successfully exported to volumes_data.csv
Mass data successfully exported to masses_data.csv
Data successfully exported to VEI_MASS_VolumeResults.csv
```

Step 2: Load Volcanic Data for Model Inputs and Validation

This cell loads datasets containing eruption source parameters and metadata.

Data Sources:

1. `Mastin` : Dataset of VEI-related eruption parameters.
2. `Aubrey` : Dataset with parameters for eruption dynamics.
3. `IVESPA` : Integrated Volcanic Eruption Source Parameters Archive for further modeling.

Purpose:

- These datasets provide observational constraints and inputs for modeling eruption parameters.
- Support statistical analysis and correlation studies to validate models.

In [7]:
```
import vei
from vei import  analyze_correlations

data_loader=vei.LOADDATA()


Mastin = data_loader.load_Mastin(as_geodataframe=False)
Aubrey=data_loader.load_Aubry(as_geodataframe=False)
IVESPA=data_loader.load_IVESPA(as_geodataframe=False)
Sparks=data_loader.load_Sparks(as_geodataframe=False)
mastin_a=data_loader.load_Mastin_a(as_geodataframe=True)

# analyze_correlations(Mastin, 'Mastin', threshold=0.7)

# analyze_correlations(Aubrey, 'Aubrey', threshold=0.7)

# analyze_correlations(Sparks, 'Sparks', threshold=0.7)

# analyze_correlations(IVESPA, 'IVESPA', threshold=0.7)

#analyze_correlations(mastin_a, 'IVESPA', threshold=0.9)

import pandas as pd

Combined_datasets= pd.concat([ Aubrey, IVESPA, mastin_a], axis=0, join='inner')

aubrey_datacombined= pd.concat([ Aubrey, IVESPA], axis=0, join='inner')
```

Step 3: Estimate Mass Eruption Rate (MER)

This cell prepares TEM data for use with the `MERPredictor` class.

Inputs:

- `tem_values` : Extracted mean probabilistic TEM values (`Mean_Mass_Prob_kg`) generated by the VEI model.

Purpose:

- `tem_values` serve as input to estimate MER, an essential eruption source parameter.
- MER will be used in subsequent steps to calculate eruption duration and plume rise.

In [8]:
```
import pandas as pd
```

```python
data=vei_toMass.data

# Drop first row (VEI zero) because VEI zero is None Explosive

#data = data.drop(index=data.index[0]).reset_index(drop=True)


# Ensuring the VEI column is numeric
data['VEI'] = pd.to_numeric(data['VEI'], errors='coerce')

# Filtering the dataframe based on VEI column and list
data = data[data['VEI'].isin(vei_range)].reset_index(drop=True)

data
```

Out[8]:

	VEI	Bulk_Volume_km3	emperical_Bulk_Volume_m3	Probabilistic_Bulk_Volume_m3	Mean_Mass_emp_kg	Std_Mass_emp_kg	Mean_Mass_Prob_kg	Std_Mass_Prob_kg	emp_Volume_5th	Uncertainty_emp_Volume_5th	...	emp_Volume_95th	Uncertainty_emp_Volume_95th	Prob_Volume_95th	Uncertainty_Prob_Volume_95th	Mass_5th	Mass_25th	Mass_50th	Mass_75th	Mass_95th	90%_interpercentile_Mass_Estimate
0	2	0.1	1.000000e+08	5.500434e+08	1.000021e+11	1.154804e+10	5.971176e+11	1.003206e+11	2300000.0	230000.0	...	1.900000e+12	1.900000e+11	3.522455e+14	3.522455e+13	4.321832e+11	5.294759e+11	5.970752e+11	6.647686e+11	7.621995e+11	3.300163e+11
1	3	3.0	3.000000e+09	5.502371e+09	2.999953e+12	3.464152e+11	5.973516e+12	1.003485e+12	2300000.0	230000.0	...	1.900000e+12	1.900000e+11	3.522455e+14	3.522455e+13	4.323738e+12	5.296897e+12	5.973094e+12	6.650092e+12	7.624805e+12	3.301067e+12
2	4	30.0	3.000000e+10	5.500256e+10	2.999957e+13	3.464464e+12	5.970940e+13	1.003092e+13	2300000.0	230000.0	...	1.900000e+12	1.900000e+11	3.522455e+14	3.522455e+13	4.321492e+13	5.294634e+13	5.970905e+13	6.647528e+13	7.620809e+13	3.299317e+13
3	5	100.0	1.000000e+11	5.498882e+11	9.999955e+13	1.154736e+13	5.969182e+14	1.002678e+14	2300000.0	230000.0	...	1.900000e+12	1.900000e+11	3.522455e+14	3.522455e+13	4.319571e+14	5.292828e+14	5.969258e+14	6.645503e+14	7.618505e+14	3.298934e+14

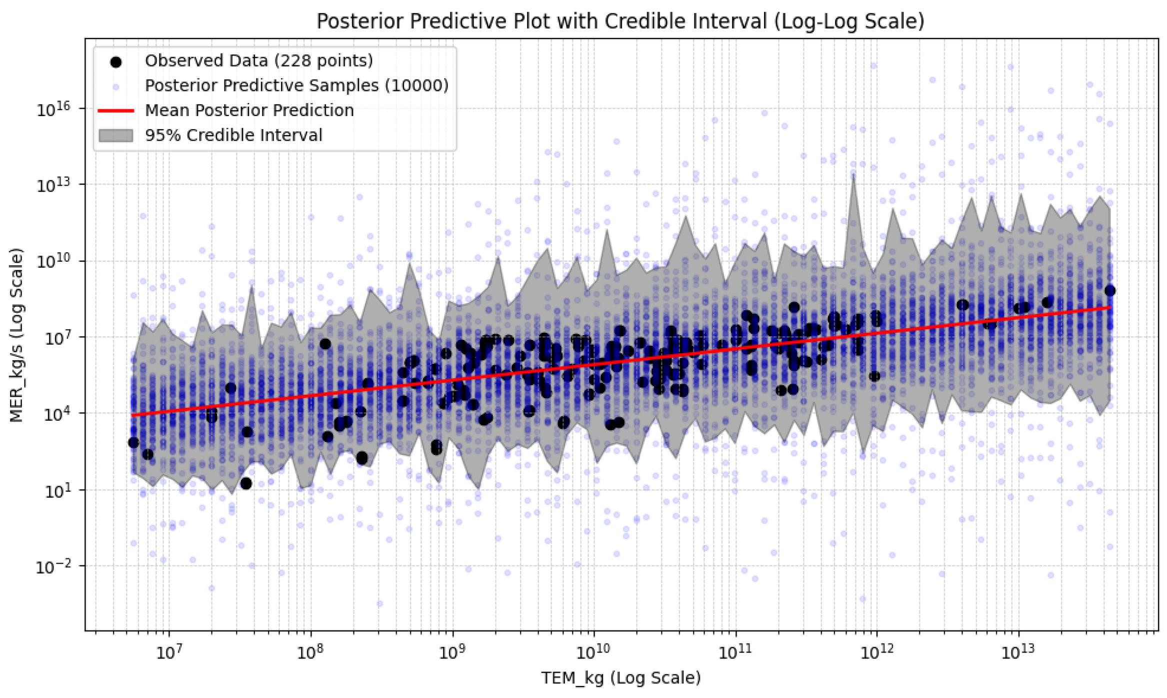
4 rows × 34 columns

In [9]:
```python
data.columns
```

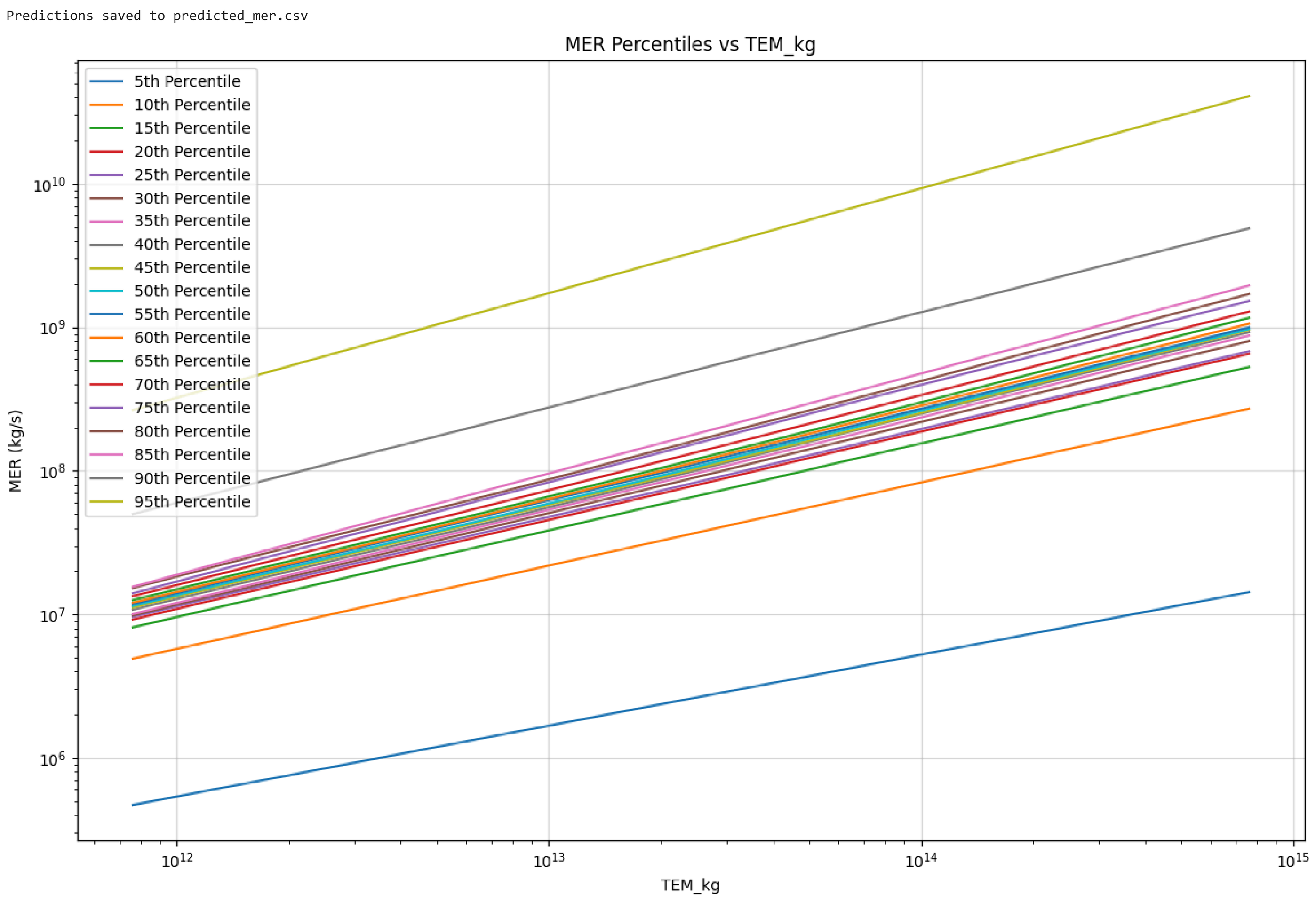
Out[9]:
```
Index(['VEI', 'Bulk_Volume_km3', 'emperical_Bulk_Volume_m3',
       'Probabilistic_Bulk_Volume_m3', 'Mean_Mass_emp_kg', 'Std_Mass_emp_kg',
       'Mean_Mass_Prob_kg', 'Std_Mass_Prob_kg', 'emp_Volume_5th',
       'Uncertainty_emp_Volume_5th', 'Prob_Volume_5th',
       'Uncertainty_Prob_Volume_5th', 'emp_Volume_25th',
       'Uncertainty_emp_Volume_25th', 'Prob_Volume_25th',
       'Uncertainty_Prob_Volume_25th', 'emp_Volume_50th',
       'Uncertainty_emp_Volume_50th', 'Prob_Volume_50th',
       'Uncertainty_Prob_Volume_50th', 'emp_Volume_75th',
       'Uncertainty_emp_Volume_75th', 'Prob_Volume_75th',
       'Uncertainty_Prob_Volume_75th', 'emp_Volume_95th',
       'Uncertainty_emp_Volume_95th', 'Prob_Volume_95th',
       'Uncertainty_Prob_Volume_95th', 'Mass_5th', 'Mass_25th', 'Mass_50th',
       'Mass_75th', 'Mass_95th', '90%_interpercentile_Mass_Estimate'],
      dtype='object')
```

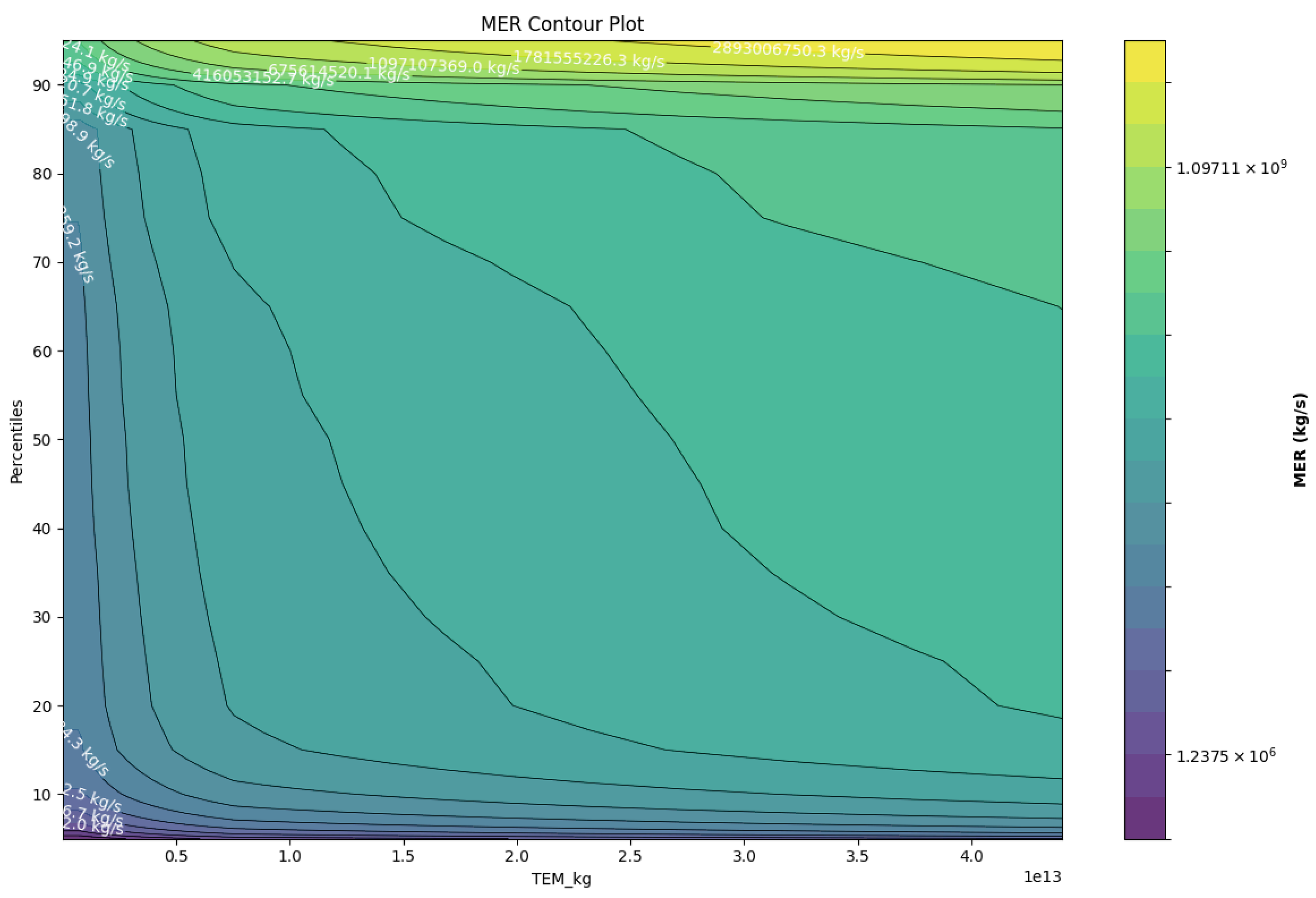
In [10]:
```python
from mer_predict import MERPredictor

tem_values=data.Mass_95th.to_list()



#tem_values=[1.998468e+11 , 3.585472e+14 ]



model = MERPredictor(aubrey_datacombined)

percentiles=list(range(5, 96, 5))

model.plot_posterior_predictive()
mer_results=model.predict_mer(tem_values, percentiles=percentiles)
#model.plot_percentiles_vs_tem()
# durations = model.convert_percentiles_to_duration(tem_values)
# print(durations)
```

```
WARNING:tensorflow:From c:\Users\mahmud\anaconda3\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

WARNING:tensorflow:From c:\Users\mahmud\anaconda3\Lib\site-packages\tensorflow_probability\python\internal\backend\numpy\_utils.py:48: The name tf.logging.TaskLevelStatusMessage is deprecated. Please use tf.compat.v1.logging.TaskLevelStatusMessage instead.

WARNING:tensorflow:From c:\Users\mahmud\anaconda3\Lib\site-packages\tensorflow_probability\python\internal\backend\numpy\_utils.py:48: The name tf.control_flow_v2_enabled is deprecated. Please use tf.compat.v1.control_flow_v2_enabled instead.

Selected model: linear with BIC = 628.8633422851562
```




```
Predictions saved to predicted_mer.csv
```





Step 4: Correlate VEI with MER Results

This cell adds the VEI values from the processed data into the `mer_results` DataFrame.

Purpose:

- Ensures that MER results are directly associated with their corresponding VEI values.
- Facilitates traceability and cross-comparison during analysis.

In [11]:
```python
mer_results["VEI"]=data['VEI']
```

Step 5: Review Mass Eruption Rate (MER) Results

This cell displays the `mer_results` DataFrame, which includes the calculated MER values and associated VEI data.

Purpose:

- Provides an overview of the MER estimates to verify accuracy and consistency.

In [12]:
```python
mer_results.columns
```

Out[12]:
```
Index(['TEM_kg', 'MERPercentile_5', 'MERPercentile_10', 'MERPercentile_15',
       'MERPercentile_20', 'MERPercentile_25', 'MERPercentile_30',
       'MERPercentile_35', 'MERPercentile_40', 'MERPercentile_45',
       'MERPercentile_50', 'MERPercentile_55', 'MERPercentile_60',
       'MERPercentile_65', 'MERPercentile_70', 'MERPercentile_75',
       'MERPercentile_80', 'MERPercentile_85', 'MERPercentile_90',
       'MERPercentile_95', '90%_interpercentile_MERrange_kg/s',
       'Duration_hr_P5', 'Duration_hr_P10', 'Duration_hr_P15',
       'Duration_hr_P20', 'Duration_hr_P25', 'Duration_hr_P30',
       'Duration_hr_P35', 'Duration_hr_P40', 'Duration_hr_P45',
       'Duration_hr_P50', 'Duration_hr_P55', 'Duration_hr_P60',
       'Duration_hr_P65', 'Duration_hr_P70', 'Duration_hr_P75',
       'Duration_hr_P80', 'Duration_hr_P85', 'Duration_hr_P90',
       'Duration_hr_P95', 'Duration_hr_90%_interpercentile', 'VEI'],
      dtype='object')
```

Step 6: Select Central MER Values for Plume Height Prediction

This cell extracts the 50th percentile (median) MER values from the `mer_results` DataFrame.

Purpose:

- Focuses on the central tendency of the MER distribution, minimizing the influence of outliers.
- The Best MER values (`mer_list`) are determined using the range between the (`95th and 5th percentiles`) and are used as input for plume height predictions..

In [13]:
```
mer_list=mer_results['MERPercentile_95'].to_list()
vei_list=mer_results.VEI.to_list()
vei_list
```

Out[13]:
```
[2, 3, 4, 5]
```

Step 7: Estimate Plume Rise Based on MER

This cell initializes the `PlumeHeightPredictor` class and calculates plume heights using the median MER values.

Methodology:

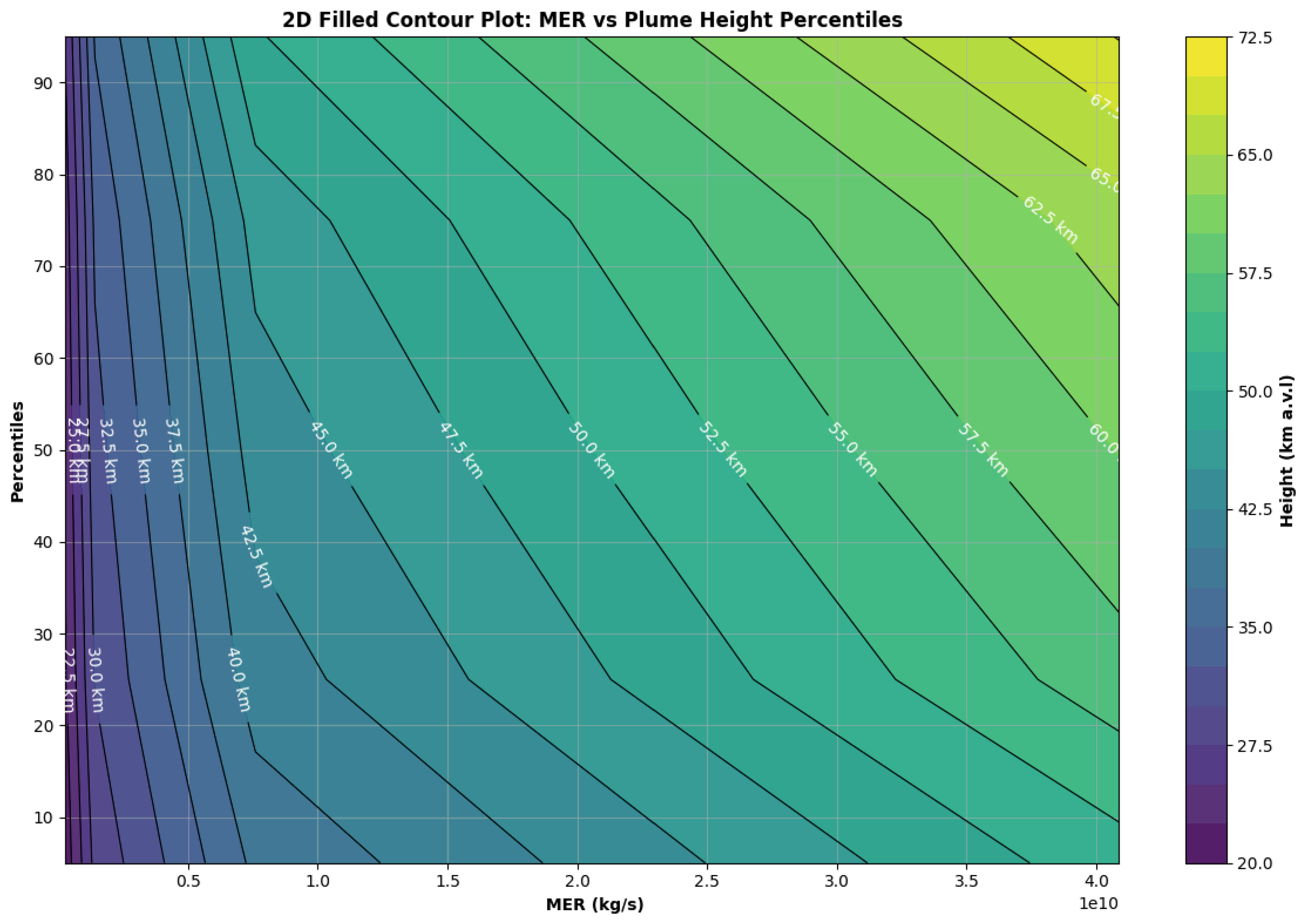
- Utilizes the `IVESPA` dataset for plume height predictions with uncertainty quantification.
- Outputs plume height estimates based on the MER values (`mer_list`).

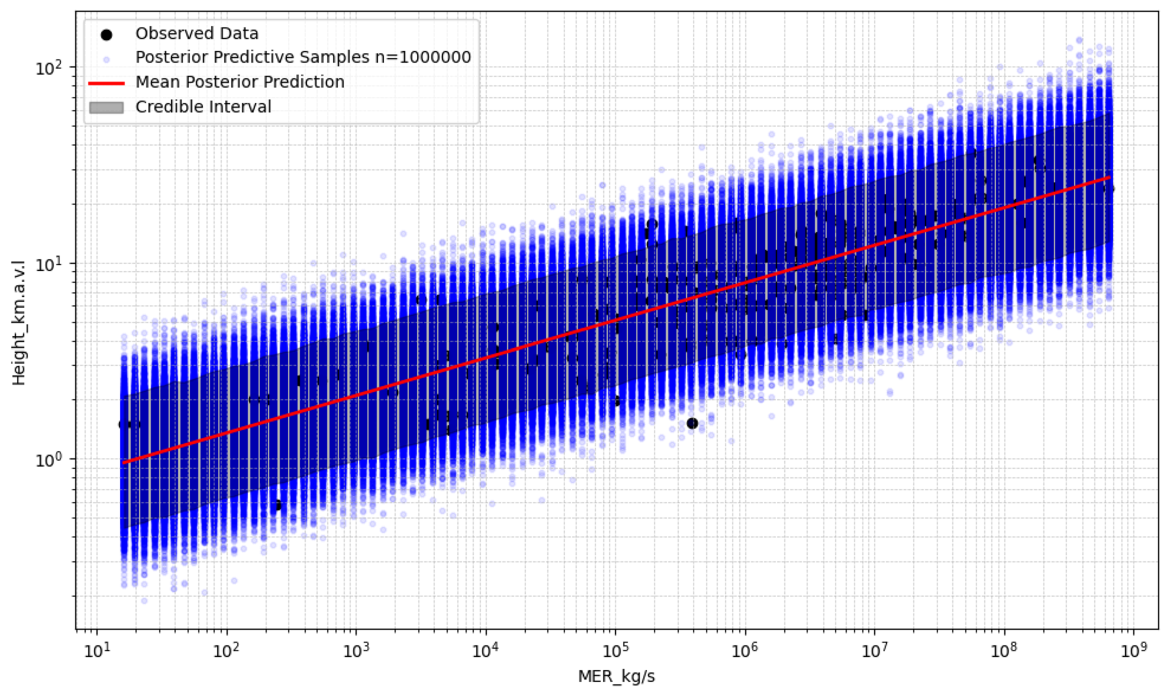
Purpose:

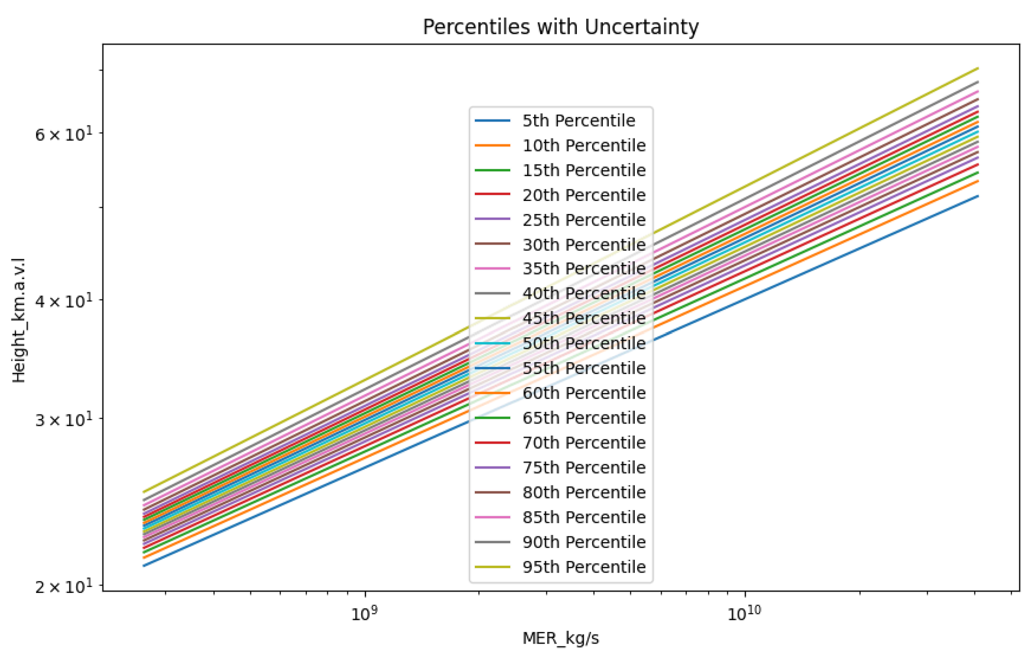
- Plume rise estimation is critical for assessing the environmental and aviation impacts of volcanic eruptions.

In [14]:
```
from plume_rise_predict import PlumeHeightPredictor

# Initialize the PlumeHeightPredictor
predictor = PlumeHeightPredictor(aubrey_datacombined)

heights=predictor.predict_height_with_uncertainty(mer_results['MERPercentile_95'].to_list(), output_file='predicted_height_with_uncertainty.csv', percentiles=percentiles  )


heights
```

```
Predictions with uncertainty saved to predicted_height_with_uncertainty.csv
```







Out[14]:

	MER_kg/s	HeightPercentile_5	HeightPercentile_10	HeightPercentile_15	HeightPercentile_20	HeightPercentile_25	HeightPercentile_30	HeightPercentile_35	HeightPercentile_40	HeightPercentile_45	...	HeightPercentile_55	HeightPercentile_60	HeightPercentile_65	HeightPercentile_70	HeightPercentile_75	HeightPercentile_80	HeightPercentile_85	HeightPercentile_90	HeightPercentile_95	90%_interpercentile_Height_range_km
0	2.637347e+08	20.915370	21.331430	21.613819	21.846972	22.059058	22.246575	22.423805	22.583176	22.732437	...	23.057809	23.210690	23.387119	23.556670	23.740064	23.975067	24.233409	24.544383	25.037020	4.121650
1	1.416006e+09	28.241585	28.951933	29.411540	29.818273	30.170567	30.480591	30.770946	31.029136	31.307137	...	31.858484	32.126473	32.404716	32.709332	33.020670	33.418236	33.880200	34.440346	35.253662	7.012077
2	7.603739e+09	38.076356	39.267137	39.994580	40.693994	41.252351	41.756176	42.223867	42.638044	43.116196	...	44.022723	44.458900	44.939674	45.432922	45.952393	46.605362	47.372896	48.301666	49.731648	11.655293
3	4.087218e+10	51.375579	53.282612	54.406904	55.483776	56.427325	57.193282	57.927141	58.647137	59.370243	...	60.855017	61.544015	62.330858	63.091283	63.932874	65.033164	66.275561	67.818301	70.117453	18.741874

4 rows × 21 columns

Step 8: Review Plume Height Estimates

This cell displays the predicted plume heights.

Purpose:

- Offers an immediate review of the plume height results for quality control and interpretation.

In [15]:
```
mer_results.columns
```

Out[15]:
```
Index(['TEM_kg', 'MERPercentile_5', 'MERPercentile_10', 'MERPercentile_15',
       'MERPercentile_20', 'MERPercentile_25', 'MERPercentile_30',
       'MERPercentile_35', 'MERPercentile_40', 'MERPercentile_45',
       'MERPercentile_50', 'MERPercentile_55', 'MERPercentile_60',
       'MERPercentile_65', 'MERPercentile_70', 'MERPercentile_75',
       'MERPercentile_80', 'MERPercentile_85', 'MERPercentile_90',
       'MERPercentile_95', '90%_interpercentile_MERrange_kg/s',
       'Duration_hr_P5', 'Duration_hr_P10', 'Duration_hr_P15',
       'Duration_hr_P20', 'Duration_hr_P25', 'Duration_hr_P30',
       'Duration_hr_P35', 'Duration_hr_P40', 'Duration_hr_P45',
       'Duration_hr_P50', 'Duration_hr_P55', 'Duration_hr_P60',
       'Duration_hr_P65', 'Duration_hr_P70', 'Duration_hr_P75',
       'Duration_hr_P80', 'Duration_hr_P85', 'Duration_hr_P90',
       'Duration_hr_P95', 'Duration_hr_90%_interpercentile', 'VEI'],
      dtype='object')
```

In [16]:
```
heights['VEI']=mer_results['VEI']
# List of columns to be assigned
columns_to_assign = ['Duration_hr_P5', 'Duration_hr_P25', 'Duration_hr_P50', 'Duration_hr_P75', 'Duration_hr_P95', 'Duration_hr_90%_interpercentile'] #'Duration_hr_best_estimate_uncertainty'

# Assign values from mer_results to heights for these columns
heights[columns_to_assign] = mer_results[columns_to_assign]



heights.to_csv('Final_prediction_results.csv')

heights
```

Out[16]:

	MER_kg/s	HeightPercentile_5	HeightPercentile_10	HeightPercentile_15	HeightPercentile_20	HeightPercentile_25	HeightPercentile_30	HeightPercentile_35	HeightPercentile_40	HeightPercentile_45	...	HeightPercentile_90	HeightPercentile_95	90%_interpercentile_Height_range_km	VEI	Duration_hr_P5	Duration_hr_P25	Duration_hr_P50	Duration_hr_P75	Duration_hr_P95	Duration_hr_90%_interpercentile
0	2.637347e+08	20.915370	21.331430	21.613819	21.846972	22.059058	22.246575	22.423805	22.583176	22.732437	...	24.544383	25.037020	4.121650	2	451.148534	22.069162	18.531088	15.088152	0.802784	0.804215
1	1.416006e+09	28.241585	28.951933	29.411540	29.818273	30.170567	30.480591	30.770946	31.029136	31.307137	...	34.440346	35.253662	7.012077	3	1445.341002	52.417107	42.635834	30.553609	1.495757	1.497307
2	7.603739e+09	38.076356	39.267137	39.994580	40.693994	41.252351	41.756176	42.223867	42.638044	43.116196	...	48.301666	49.731648	11.655293	4	4628.288861	128.057035	96.568842	63.982951	2.784014	2.785689
3	4.087218e+10	51.375579	53.282612	54.406904	55.483776	56.427325	57.193282	57.927141	58.647137	59.370243	...	67.818301	70.117453	18.741874	5	14822.270572	312.278345	218.349011	139.006366	5.177730	5.179540

4 rows × 28 columns

In [17]:
```
# Add a bold title using Styler
styled_table = heights.style.set_caption("<b>VEI and Related Data</b>")
styled_table
```

Out[17]:

VEI and Related Data

	MER_kg/s	HeightPercentile_5	HeightPercentile_10	HeightPercentile_15	HeightPercentile_20	HeightPercentile_25	HeightPercentile_30	HeightPercentile_35	HeightPercentile_40	HeightPercentile_45	HeightPercentile_50	HeightPercentile_55	HeightPercentile_60	HeightPercentile_65	HeightPercentile_70	HeightPercentile_75	HeightPercentile_80	HeightPercentile_85	HeightPercentile_90	HeightPercentile_95	90%_interpercentile_Height_range_km
0	263734738.339319	20.915370	21.331430	21.613819	21.846972	22.059058	22.246575	22.423805	22.583176	22.732437	22.895094	23.057809	23.210690	23.387119	23.556670	23.740064	23.975067	24.233409	24.544383	25.037020	4.121650
1	1416005871.096091	28.241585	28.951933	29.411540	29.818273	30.170567	30.480591	30.770946	31.029136	31.307137	31.583728	31.858484	32.126473	32.404716	32.709332	33.020670	33.418236	33.880200	34.440346	35.253662	7.012077
2	7603738971.918003	38.076356	39.267137	39.994580	40.693994	41.252351	41.756176	42.223867	42.638044	43.116196	43.576116	44.022723	44.458900	44.939674	45.432922	45.952393	46.605362	47.372896	48.301666	49.731648	11.655293
3	40872184089.891838	51.375579	53.282612	54.406904	55.483776	56.427325	57.193282	57.927141	58.647137	59.370243	60.101794	60.855017	61.544015	62.330858	63.091283	63.932874	65.033164	66.275561	67.818301	70.117453	18.741874

In [18]:
```
from rich.console import Console
from rich.table import Table

# Ensure VEI is first
if "VEI" not in heights.columns:
    raise KeyError("The 'VEI' column is missing from the DataFrame.")

cols = ["VEI"] + [col for col in heights.columns if col != "VEI"]
heights = heights[cols]

# Rich Console for better formatting
console = Console()

# Function to print DataFrame in chunks of six columns, keeping VEI first
def print_six_columns_with_vei_first(df):
    num_cols = len(df.columns)
    for i in range(1, num_cols, 3):  # 5 because VEI is fixed
        # Slice five columns at a time, after the first column
        cols = ["VEI"] + list(df.columns[i:i+3])
```

```python
        # Create a table for this chunk
        table = Table(title="VEI and Related Data")
        for col in cols:
            table.add_column(col, justify="center")

        # Add rows to the table
        for _, row in df[cols].iterrows():
            table.add_row(*[str(cell) for cell in row])

        # Print the table
        console.print(table)

# Display the DataFrame
print_six_columns_with_vei_first(heights)
```

VEI and Related Data

VEI	MER_kg/s	HeightPercentile_5	HeightPercentile_10
2.0	263734738.33931884	20.915369546884616	21.331429911766858
3.0	1416005871.0960913	28.241585300901246	28.951933087831016
4.0	7603738971.918003	38.076355785647905	39.26713707196505
5.0	40872184089.89184	51.3755789018824	53.28261213731151

VEI and Related Data

VEI	HeightPercentile_15	HeightPercentile_20	HeightPercentile_25
2.0	21.613819391639456	21.84697161894229	22.059057606035502
3.0	29.411540437974523	29.81827314889459	30.17056664361438
4.0	39.99457973600822	40.69399355593317	41.25235061535111
5.0	54.40690431501138	55.48377603870605	56.42732532401312

VEI and Related Data

VEI	HeightPercentile_30	HeightPercentile_35	HeightPercentile_40
2.0	22.246575114862893	22.42380470257502	22.583176414439016
3.0	30.480590876761344	30.77094562365109	31.029135834526937
4.0	41.75617598951781	42.22386736536354	42.63804386288488
5.0	57.19328215742592	57.92714138033494	58.647136971993596

VEI and Related Data

VEI	HeightPercentile_45	HeightPercentile_50	HeightPercentile_55
2.0	22.732436772142417	22.895094282778047	23.057808700834116
3.0	31.30713665639755	31.583727871348618	31.858484018109245
4.0	43.11619637338617	43.57611610958067	44.022723092844224
5.0	59.37024344739864	60.10179441562113	60.85501749374209

VEI and Related Data

VEI	HeightPercentile_60	HeightPercentile_65	HeightPercentile_70
2.0	23.210689842837628	23.38711907299264	23.556670431727383
3.0	32.126472858206355	32.40471585358821	32.70933202652877
4.0	44.45889992344718	44.9396743140395	45.43292225100868
5.0	61.544014895480494	62.33085820499958	63.09128261246717

VEI and Related Data

VEI	HeightPercentile_75	HeightPercentile_80	HeightPercentile_85
2.0	23.74006439707223	23.975066627420354	24.233408623387593
3.0	33.02066981132512	33.41823569886797	33.88019987500591
4.0	45.952392748571384	46.6053622940062	47.372896412347515
5.0	63.932873853739785	65.03316438470173	66.27556073642538

VEI and Related Data

VEI	HeightPercentile_90	HeightPercentile_95	90%_interpercentile_Height_range_km
2.0	24.544383099325774	25.03701982289551	4.121650276010893
3.0	34.44034604949838	35.25366217166669	7.012076870765444
4.0	48.301665797482215	49.73164840773057	11.655292622082662
5.0	67.81830086317025	70.11745295185922	18.741874049976822

VEI and Related Data

VEI	Duration_hr_P5	Duration_hr_P25	Duration_hr_P50
2.0	451.1485341090696	22.069162047488895	18.53108808511808
3.0	1445.3410022229234	52.41710686727613	42.63583418923697
4.0	4628.288861281014	128.05703457866034	96.56884214551671
5.0	14822.270572316527	312.27834475899505	218.34901134200308

VEI and Related Data

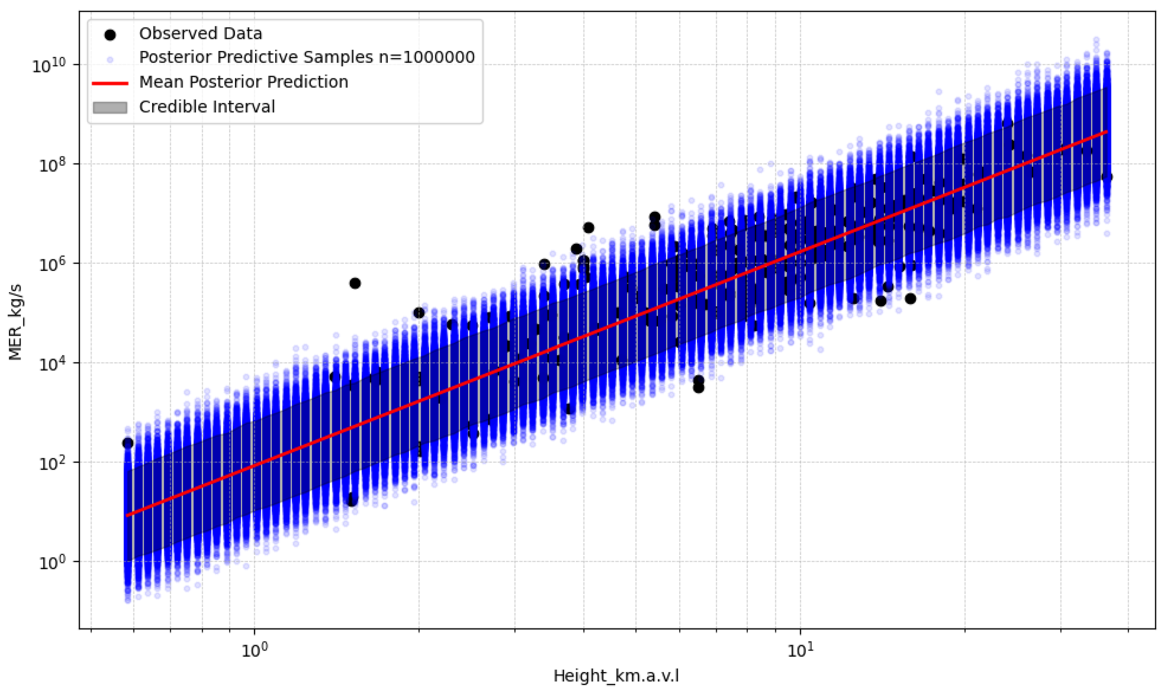
VEI	Duration_hr_P75	Duration_hr_P95	Duration_hr_90%_interpercentile
2.0	15.088151759080407	0.8027842183454168	0.8042152577221199
3.0	30.553609033576713	1.495757453709167	1.4973069898304596
4.0	63.98295122142996	2.7840138361930653	2.785689487456169
5.0	139.0063660800366	5.17773024112124	5.179539563012893

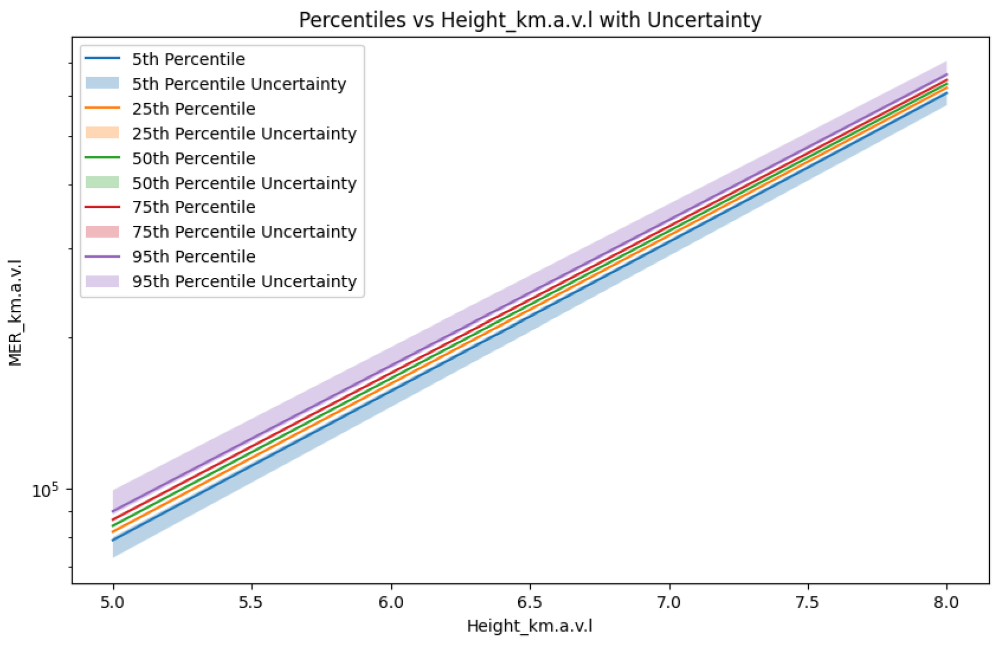
Refining Mass Eruption Rate (MER) Estimates Using Observed Plume Height

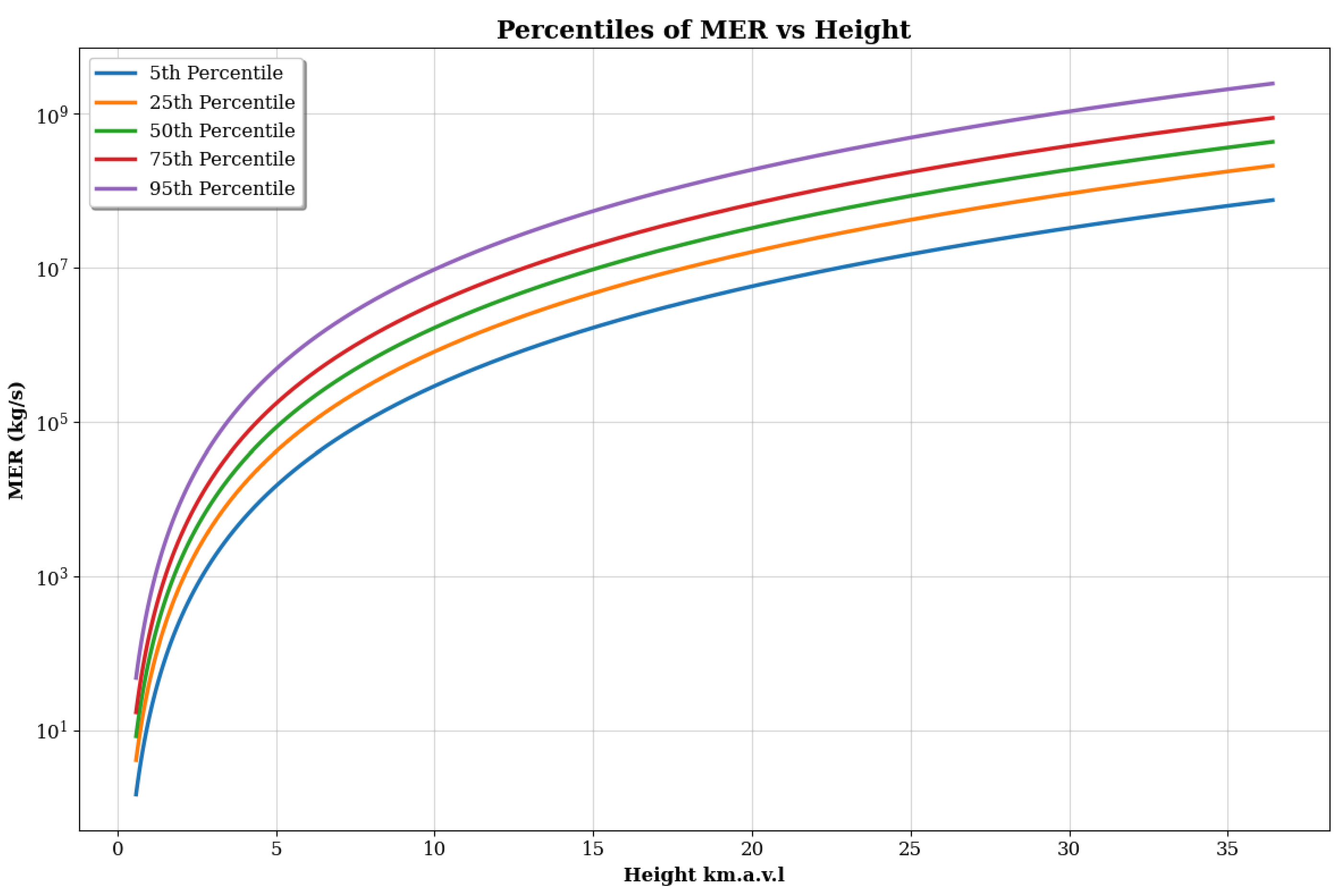
When plume height is observed through satellite imagery or infrasound data, it provides an opportunity to enhance previously predicted MER values. By leveraging this direct observation of plume height, we can derive a more accurate estimate of the MER, aligning it closely with the actual eruption dynamics.

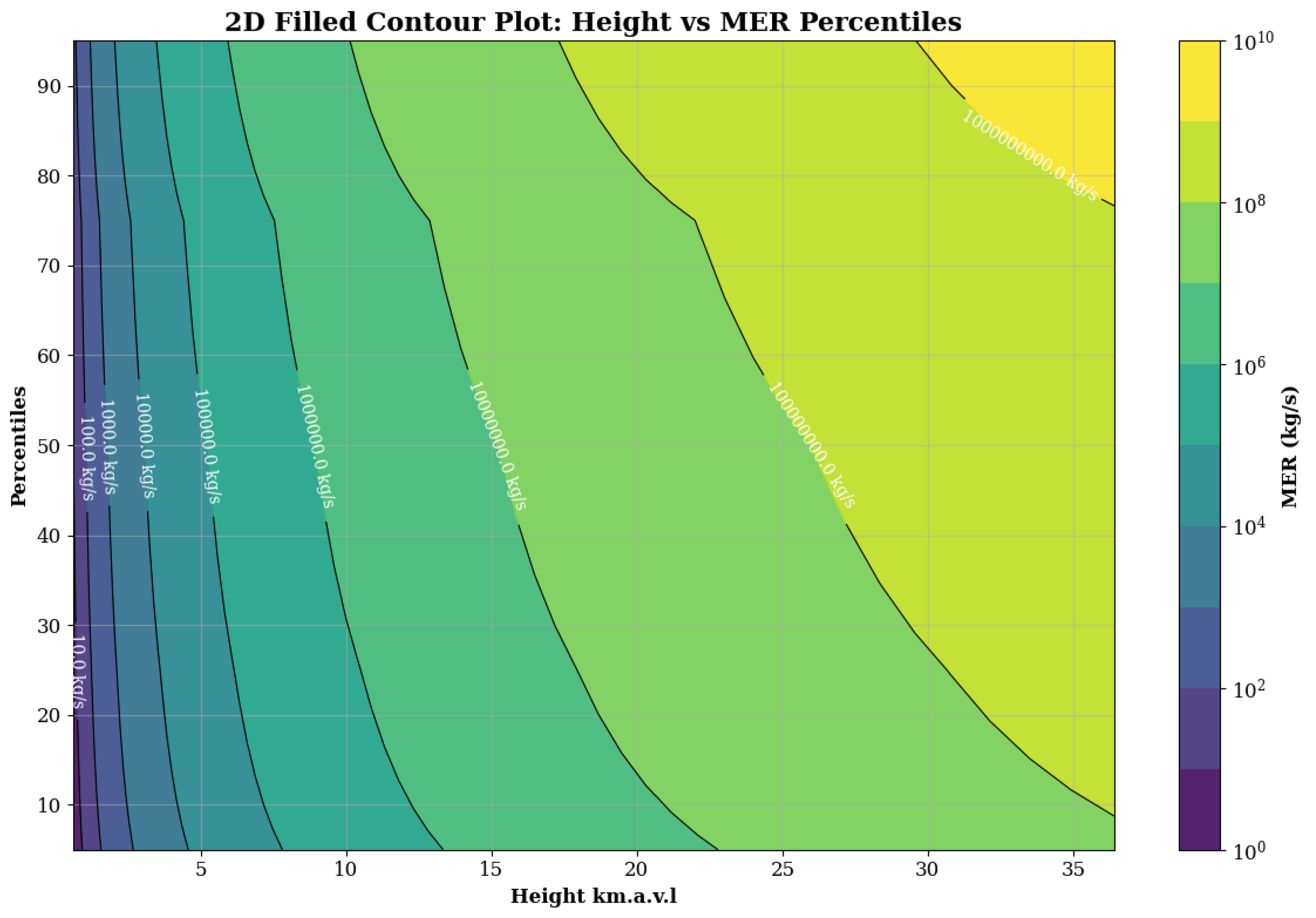
In [19]:
```python
from mer_from_height import MERPredictorFromHeight

predictor = MERPredictorFromHeight(Combined_datasets)
# Predict MER for specific heights
# Save predictions to a CSV
Observed_plume_height=[5,8]
predictions=predictor.predict_MER_with_uncertainty(Observed_plume_height, output_file='predicted_mer_fromheight.csv')


# Visualize the posterior predictive distribution
predictor.plot_posterior_predictive()

# Visualize percentiles with uncertainty
predictor.plot_percentiles_with_uncertainty(Observed_plume_height)

predictor.plot_percentiles_vs_mer()

predictions
```

```
Predictions with uncertainty saved to predicted_mer_fromheight.csv
```









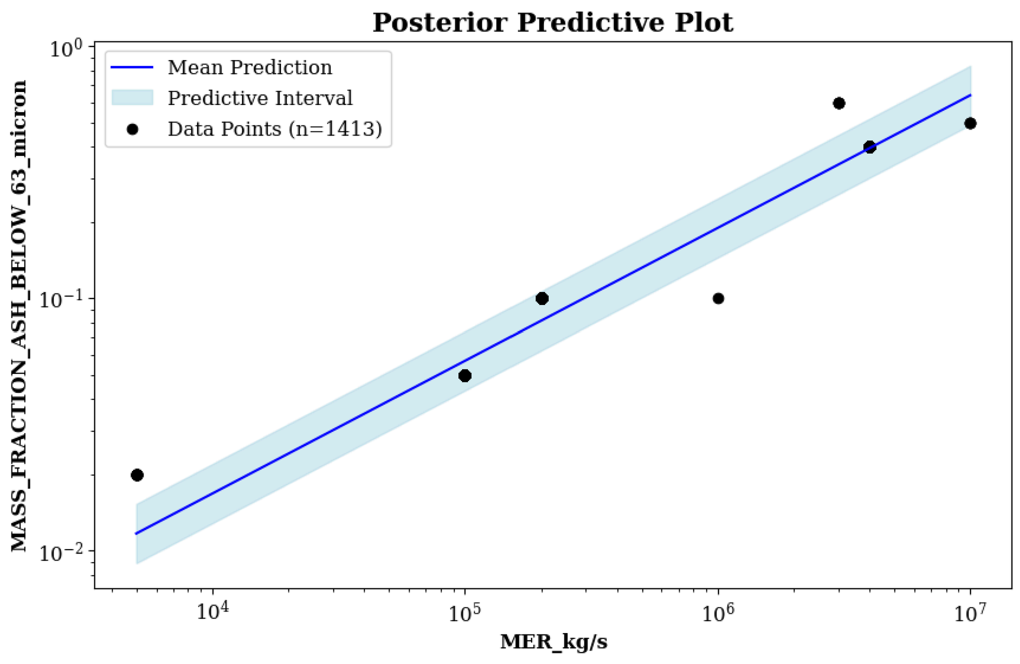
Out[19]:

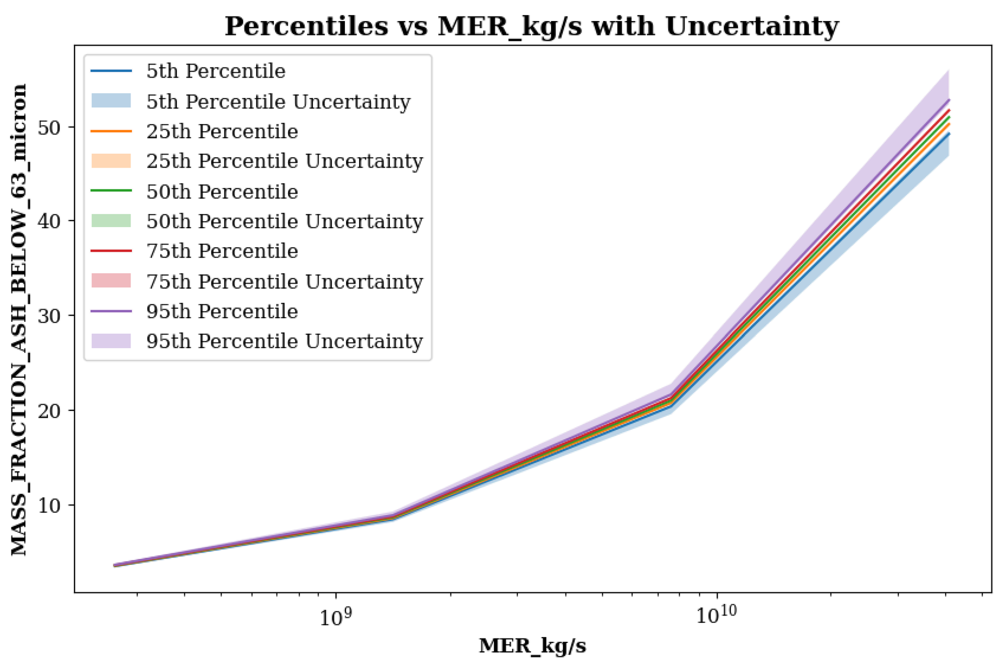
	Height_km.a.v.l	MERPercentile_5	Uncertainty_Lower_5	Uncertainty_Upper_5	MERPercentile_25	Uncertainty_Lower_25	Uncertainty_Upper_25	MERPercentile_50	Uncertainty_Lower_50	Uncertainty_Upper_50	MERPercentile_75	Uncertainty_Lower_75	Uncertainty_Upper_75	MERPercentile_95	Uncertainty_Lower_95	Uncertainty_Upper_95	Best_MER_Estimate_km	Best_MER_Estimate_km_Uncertainty
0	5	78892.286871	72753.555476	80015.857555	82011.604587	81451.349253	82538.735752	84271.983168	83842.317081	84736.828325	86656.573911	86098.898191	87243.454000	90054.256717	88672.015678	99136.123217	11161.969846	19120.265661
1	8	608619.516516	575831.605351	614226.709984	623748.137283	621135.854514	626352.255729	634947.913629	632969.021044	637016.419802	646340.111457	643765.208387	649292.414642	662707.458575	656473.656001	704862.135429	54087.942059	90635.425445

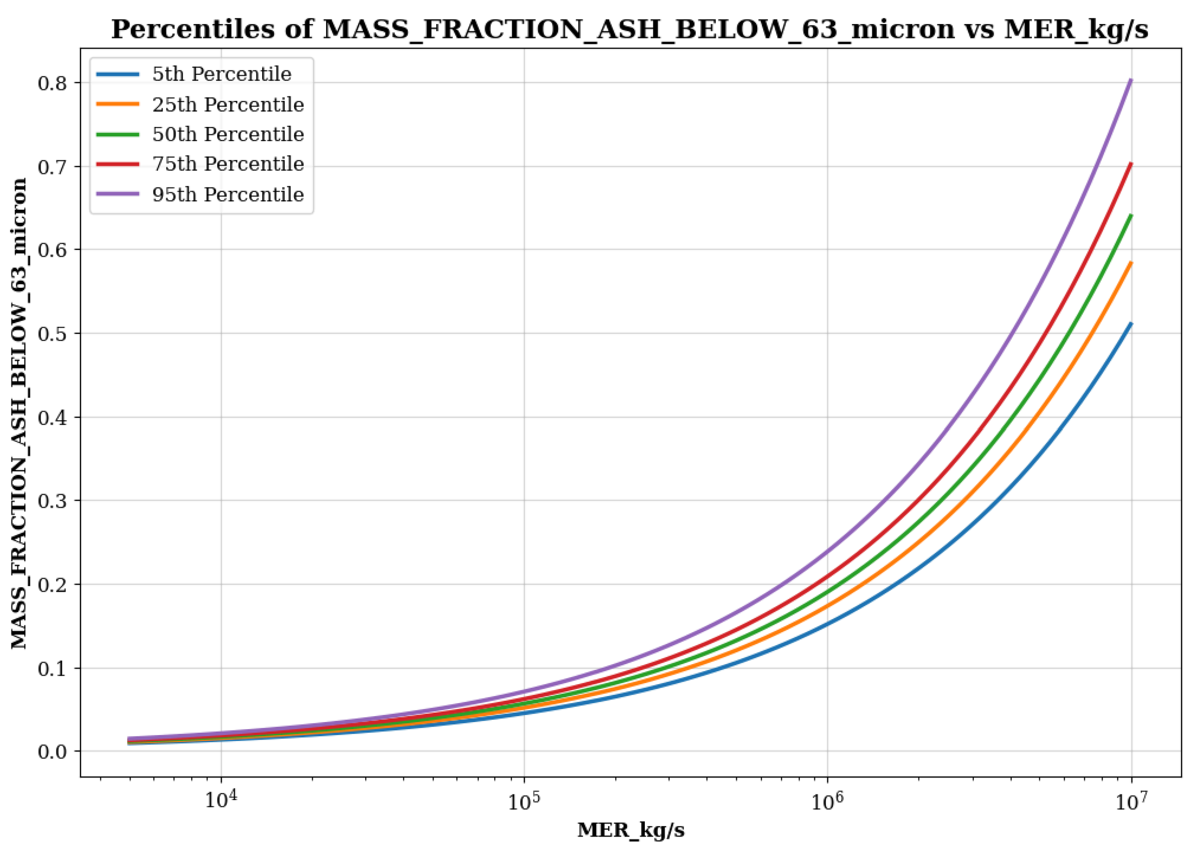
In [20]:

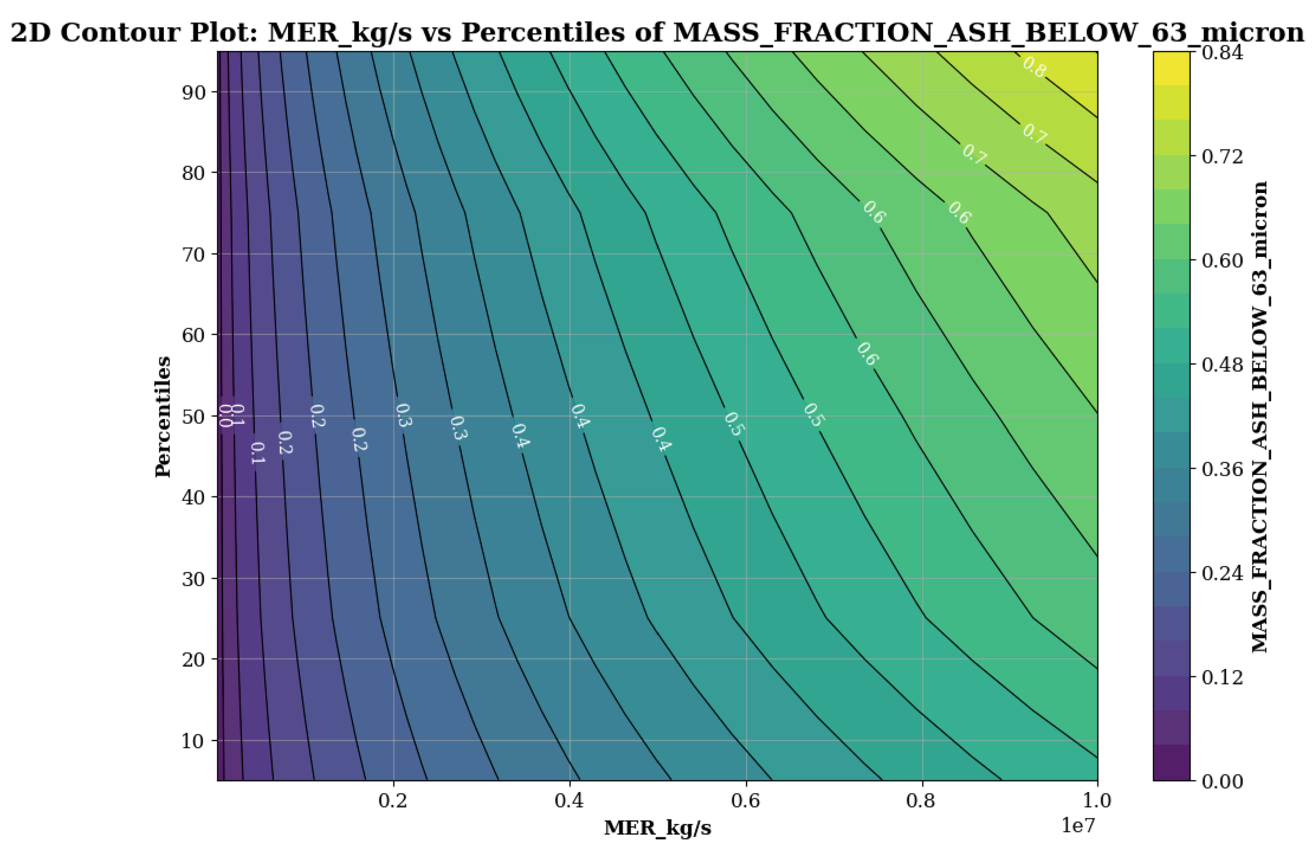
```python
from ParticleSize_MER import Predict_ASH_BELOW_63_Micron

from vei import data_loader

#mastin_a=data_loader.load_Mastin_a(as_geodataframe=True)

psize_model=Predict_ASH_BELOW_63_Micron(mastin_a)

psize_model.set_xvar('MER_kg/s')
psize_model.set_yvar('MASS_FRACTION_ASH_BELOW_63_micron')


psize_model.calculate_percentiles_with_uncertainty(mer_list)
psize_model.predict_with_uncertainty(mer_list)

psize_model.plot_posterior_predictive()


# Visualize percentiles with uncertainty
psize_model.plot_percentiles_with_uncertainty(mer_list)

psize_model.plot_percentiles()

psize_predictions=psize_model.predict_with_uncertainty(mer_list)
```

```
Model updated with xvar='MER_kg/s' and yvar='MER_kg/s'.
Model updated with xvar='MER_kg/s' and yvar='MASS_FRACTION_ASH_BELOW_63_micron'.
```









In [21]:

```python
psize_predictions
```

Out[21]:

	MER_kg/s	MASS_FRACTION_ASH_BELOW_63_micron_Percentile_5	Uncertainty_Lower_5	Uncertainty_Upper_5	MASS_FRACTION_ASH_BELOW_63_micron_Percentile_25	Uncertainty_Lower_25	Uncertainty_Upper_25	MASS_FRACTION_ASH_BELOW_63_micron_Percentile_50	Uncertainty_Lower_50	Uncertainty_Upper_50	MASS_FRACTION_ASH_BELOW_63_micron_Percentile_75	Uncertainty_Lower_75	Uncertainty_Upper_75	MASS_FRACTION_ASH_BELOW_63_micron_
0	2.637347e+08	3.510268	3.415337	3.525689	3.551713	3.544573	3.557974	3.580712	3.575253	3.586068	3.609362	3.603023	3.616482	
1	1.416006e+09	8.459594	8.174954	8.506458	8.585589	8.563127	8.604689	8.673290	8.656394	8.690388	8.760266	8.741047	8.782273	
2	7.603739e+09	20.388173	19.569087	20.524390	20.753400	20.687811	20.809752	21.009633	20.961152	21.058963	21.265665	21.208511	21.330891	
3	4.087218e+10	49.159041	46.852932	49.545688	50.194972	50.005336	50.353139	50.918376	50.783619	51.057364	51.647173	51.484248	51.832791	

In [22]:

```python
columns_to_assign=list(psize_predictions.columns)
columns_to_assign=columns_to_assign[1:]

# Assuming `columns_to_assign` contains column names and `heights` and `psize_predictions` are DataFrames

heights.loc[:, columns_to_assign] = psize_predictions[columns_to_assign]


heights.to_csv('Final_prediction_results.csv')
```

In [23]:

```python
psize_predictions
```

Out[23]:

	MER_kg/s	MASS_FRACTION_ASH_BELOW_63_micron_Percentile_5	Uncertainty_Lower_5	Uncertainty_Upper_5	MASS_FRACTION_ASH_BELOW_63_micron_Percentile_25	Uncertainty_Lower_25	Uncertainty_Upper_25	MASS_FRACTION_ASH_BELOW_63_micron_Percentile_50	Uncertainty_Lower_50	Uncertainty_Upper_50	MASS_FRACTION_ASH_BELOW_63_micron_Percentile_75	Uncertainty_Lower_75	Uncertainty_Upper_75	MASS_FRACTION_ASH_BELOW_63_micron_
0	2.637347e+08	3.510268	3.415337	3.525689	3.551713	3.544573	3.557974	3.580712	3.575253	3.586068	3.609362	3.603023	3.616482	
1	1.416006e+09	8.459594	8.174954	8.506458	8.585589	8.563127	8.604689	8.673290	8.656394	8.690388	8.760266	8.741047	8.782273	
2	7.603739e+09	20.388173	19.569087	20.524390	20.753400	20.687811	20.809752	21.009633	20.961152	21.058963	21.265665	21.208511	21.330891	
3	4.087218e+10	49.159041	46.852932	49.545688	50.194972	50.005336	50.353139	50.918376	50.783619	51.057364	51.647173	51.484248	51.832791	