



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**OBSERVAÇÕES DAS PLUMAS DE AEROSSÓIS
DEIXADAS PELA ERUPÇÃO DO VULCÃO HUNGA
TONGA-HUNGA HA'APAI EM SÃO JOSÉ DOS CAMPOS**

Pedro Luís Rocha Maia

INPE

São José dos Campos

2024

RESUMO

O objetivo deste trabalho foi observar a presença de aerossóis na estratosfera utilizando o LIDAR de Sódio e Potássio do Laboratório Conjunto de Cooperação China-Brasil para estudos do Clima Espacial (CBJLSW-NaK), situado no INPE. O Grupo de Física da Alta Atmosfera (FISAT) do INPE tem conduzido pesquisas sobre aerossóis estratosféricos utilizando LiDAR desde 1972, mas desde 2012 que as medidas de aerossóis estavam suspensas. No dia 14 de janeiro de 2022 ocorreu uma erupção vulcânica na ilha de Hunga Tonga-Hunga Ha'apai, no sul do Pacífico. Devido à submersão do vulcão, por razões físico-químicas, quando o magma liberado entrou em contato com a água, transformou-se rapidamente em gás e se expandiu pela atmosfera de forma acelerada. No dia 17 de janeiro de 2022, o LiDAR CBJLSW-NaK estava operando na cidade de São José dos Campos e foram observados sinais correspondentes ao espalhamento por aerossóis entre 20 e 40 km de altitude, na região da estratosfera. Desde então os aerossóis do vulcão Tonga estão sendo analisados. Com os dados de Na e K observados pelo Lidar, no período de 17 de fevereiro a 8 de agosto de 2022, foi realizada uma filtragem dos valores correspondentes à faixa de altitude de 20 a 40 km. Em seguida foi desenvolvido, durante o primeiro semestre de 2023, um programa em Python no software de programação Visual Studio Code (VS Code), com serviço de nuvem gratuito, para análise de dados. Tal programa foi criado com o objetivo de fazer a leitura dos dados de Na e K, além de facilitar a criação rápida e dinâmica dos gráficos. Utilizando as bibliotecas e recursos disponíveis do Python, foi possível implantar as funcionalidades necessárias para manipular os dados e gerar visualizações gráficas de forma eficiente. Essa abordagem permitiu uma análise mais ágil e simplificada dos dados coletados, facilitando a interpretação dos resultados e possibilitando uma compreensão mais aprofundada dos fenômenos observados.

SUMÁRIO

1. INTRODUÇÃO	
1.1. OBJETIVO GERAL	
1.2. OBJETIVO ESPECÍFICO.....	
2. DESENVOLVIMENTO.....	
2.1. DADOS E REFERÊNCIAS	
2.2. METODOLOGIA.....	
2.3. RESULTADO E CONCLUSÕES.....	
3. CONCLUSÃO.....	
4. REFERÊNCIAS.....	

1. INTRODUÇÃO

A emissão de gases e aerossóis na atmosfera provenientes de erupções vulcânicas é um tema amplamente abordado por diversos centros de pesquisa espaciais. O Instituto Nacional de Pesquisas Espaciais (INPE) não é exceção, sendo que o mesmo realiza a medição de aerossóis por meio do Radar de Laser, ou Lidar, há muitos anos.

O LiDAR (Light Detection and Ranging) é um sensor remoto ativo e um método direto de aquisição de dados. Esse sistema utiliza uma fonte própria de energia, um laser que emite feixes de luz para a atmosfera, em comprimentos de onda específicos, e um telescópio para coletar o feixe refletido pela atmosfera. No caso do LiDAR instalado no INPE, os feixes de luz são produzidos em dois comprimentos de ondas distintas, que são 589 nm e 770 nm, para medidas de sódio e potássio, respectivamente.

No dia 17 de janeiro de 2022, o LiDAR do INPE detectou um aumento do sinal de fóton-contagem recebido na região compreendida entre a estratosfera e a baixa mesosfera (de 20 a 40 km). Posteriormente, verificou-se que esse aumento do sinal estava associado à erupção do vulcão submerso Hunga Tonga-Hunga Ha'apai. Na figura 1.1 é mostrado os níveis da atmosfera para uma melhor compreensão.

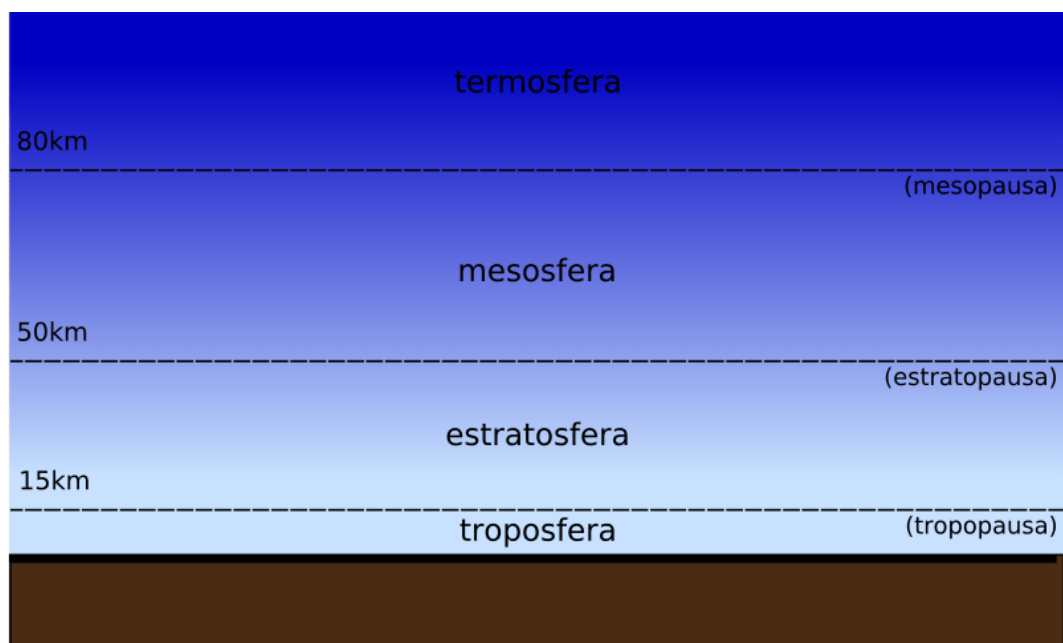


Figura 1.1: Camadas da atmosfera em função da altura.

Devido à configuração do LiDAR ser para medir os metais na mesosfera, foi preciso fazer uma adaptação para que fosse possível observar os aerossóis na estratosfera. Na saída do feixe do laser foi colocada uma íris que, no início da noite, era parcialmente

fechada para que pudesse diminuir a intensidade do feixe. Assim ele chegaria com menos intensidade na média atmosfera e conseguiria observar espalhamento da luz pelos aerossóis, o que anteriormente não era possível com a íris totalmente aberta.

Os registros coletados em relação às plumas de aerossóis deixadas pela erupção vulcânica permitiram a análise e interpretação dos dados recebidos, possibilitando a continuidade da série histórica de medições de aerossóis estratosféricos na cidade de São José dos Campos.

Para garantir a precisão dos dados, foi necessário realizar uma filtragem da base de dados coletada. Isso possibilitou uma melhor visualização das camadas de aerossóis, além de identificar as altitudes específicas na atmosfera onde os aerossóis estavam.

1.1. OBJETIVO GERAL

Desenvolver um software capaz de manipular os dados de forma automatizada, permitindo a análise detalhada da interferência das plumas na região atmosférica estudada para a manipulação eficiente dos dados e geração de gráficos de linha e de contorno para que fosse possível uma análise mais precisa e assertiva sobre a interferência das plumas na região atmosférica analisada.

1.2. OBJETIVO ESPECÍFICO

- Compreensão do princípio de funcionamento do LiDAR
- Compreensão das estruturas da atmosfera e a presença de aerossóis na estratosfera
- Continuação do desenvolvimento de um programa para análise dos dados dos aerossóis do Vulcão Hunga Tonga
- Elaboração do relatório

2. DESENVOLVIMENTO

2.1. DADOS E REFERÊNCIAS

Os dados e referências utilizados neste relatório baseiam-se em uma extensa pesquisa realizada no INPE, que incluiu a leitura de diversos artigos relacionados à geofísica espacial e ao estudo de erupções vulcânicas ocorridas no final do século 20. Esses artigos foram essenciais para o entendimento do funcionamento de um LiDAR e proporcionaram uma contextualização dos principais motivos que impulsionam os estudos sobre aerossóis na atmosfera.

Estudos recentes utilizaram dados de satélites para caracterizar as emissões do vulcão submarino Hunga Tonga-Hunga Ha'apai e investigar a relação entre o poder radioativo vulcânico (VRP) e as taxas de emissão de SO₂ em vulcões com lagos de lava. Por fim, o texto enfatiza a crescente importância de algoritmos avançados para análise de dados de sensoriamento remoto, dado o volume significativo de informações geradas por esses instrumentos, destacando a necessidade de desenvolvimento contínuo nesta área para melhorar a interpretação e utilização dos dados (Earth Science, 2023, p. 02).

A leitura dos artigos do INPE permitiu um aprofundamento significativo no conhecimento sobre as propriedades e características dos aerossóis atmosféricos, bem como a interação entre essas partículas e a radiação laser utilizada no Lidar. Essa compreensão ampla foi fundamental para uma análise consistente dos dados coletados durante o estudo dos aerossóis provenientes do vulcão Hunga Tonga e também para a realização da programação de automação dos dados usando a linguagem de programação Python para estudos estatísticos.

Os métodos estáticos funcionam quase como as funções sem instância simples dentro de uma classe e os métodos de classe recebem uma classe, em vez de uma instância. Funções internas especiais devem ser chamadas dentro da classe para ativar esses modos de método: `staticmethod` e `classmethod`. Como isso também é uma solução para um problema duradouro no Python. (Lutz, 2003, p. 370)

2.2. METODOLOGIA

A metodologia utilizada neste estudo envolveu o uso do radar de laser (LiDAR) disponibilizado pelo INPE, bem como a análise dos dados obtidos.

Inicialmente, a análise dos dados foi realizada utilizando programação no software Excel, onde foram aplicados filtros para selecionar os dados relevantes relacionados aos aerossóis. Uma tabela foi criada com a relação entre a altura e o horário durante a noite em que o Lidar estava operando. Além disso, foi calculada a média dos ruídos existentes entre 120 e 140 km.

No âmbito do processamento de dados do Lidar, a melhor opção foi utilizar a linguagem Python devido à sua ampla utilização na área de processamento de dados e análise científica. Em relação ao software, foi usado o Google Colab no primeiro ano de pesquisa, posteriormente foi passado para o Visual Studio Code (VS Code), uma plataforma mais desenvolvida para programação no geral. Ambas as ferramentas oferecem recursos avançados para o processamento e análise dos dados do LiDAR, permitindo a extração de informações relevantes sobre os aerossóis do vulcão Hunga Tonga, com o objetivo de automatizar o processo anteriormente realizado manualmente no Excel.

Tal programa permitiu a execução das etapas de filtragem de dados, cálculo da média dos resíduos e geração de gráficos. Essa metodologia permitiu a visualização e interpretação dos resultados obtidos. Na figura 2.1. é mostrada a primeira parte do processo de filtragem dos dados feito no Excel.

DATA	INICIO	FINAL
24/03	23:24	00:24
29/03	22:40	23:17
21/4	22:18	22:59
09/05	22:01	22:40
10/05	02:22	03:03
17/05	22:00	23:00
18/05	23:21	23:51

Figura 2.1: Tabela com os horários de abertura e fechamento da íris.

Nessa parte é mostrada apenas a filtragem feita no Excel, no qual foi usado um relatório físico que é atualizado a cada vez que há abertura ou fechamento da íris do LiDAR. A partir do relatório foi feita uma planilha com a data e os horários de abertura (início) e do fechamento (final) da íris.

Com a tabela feita, foi necessário importar dados de um arquivo .dat usando a biblioteca Pandas no ambiente Python. Os dados contêm informações sobre ruídos atmosféricos em diferentes altitudes, e o foco foi na faixa entre 120 km e 140 km de altitude. Utilizando as funções disponíveis na linguagem Python, foi feita a separação dos valores de altitude correspondentes à faixa de interesse. Em seguida, foi calculada a média dos ruídos para cada valor de altitude presente neste intervalo de altura. Esse cálculo fornece uma visão geral das tendências dos ruídos atmosféricos nessa região da atmosfera.

2.3. RESULTADO E CONCLUSÕES

Com todos os passos descritos na metodologia, foi possível alcançar os seguintes resultados.

Na figura 2.2 estão as primeiras linhas de código em Python no ambiente virtual de programação, o Visual Studio Code.

```
#IMPORTANDO OS DADOS

import pandas as pd

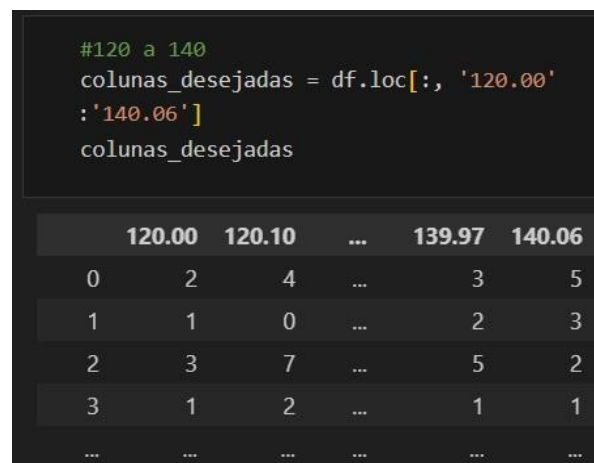
df = pd.read_csv ("c:/Users/Usuario/Desktop/ASCII/2022/20220324/K20220324_photocount.dat",
delimiter='\\s+')
pd.options.display.max_rows = 8
df.drop(['0.00'], axis = 1)
#TRANSPOR MATRIZ
df_transposta = df.T
df_transposta = df_transposta.drop(['0.00'], axis = 0)
primeiras_40_linhas = df_transposta.head(418)
primeiras_40_linhas
```

Figura 2.2: Código para transpor a matriz inicial.

Na importação de dados foi feito um código em Pandas, uma biblioteca de análise de dados de código. Ela fornece estruturas de dados de alto desempenho e fáceis de usar, além de ferramentas para manipulação e análise de dados. A biblioteca em questão foi usada

para que fosse possível a leitura do arquivo da foto contagem em .dat. Logo após, é mostrada a transposição que foi feita do arquivo, para se ajustar aos padrões necessários para uma melhor análise dos aerossóis. Para isso, a função .T foi o suficiente para que a transposição fosse realizada. Após esse passo, foi feita uma filtragem para que pudesse permanecer apenas as altitudes de 120 km até 140 km, região da atmosfera onde se encontra um maior número de ruídos causados pelo espalhamento da luz.

O principal motivo para a diferenciação dessas altitudes reside na necessidade de atenuar o impacto médio dos ruídos nos sinais contidos na faixa de até 40 km. Esta faixa é especialmente significativa devido à maior concentração das plumas de aerossóis emitidas pelo vulcão. Assim, a segmentação das altitudes permite a obtenção de uma estimativa estatística mais precisa dos dados, garantindo sua confiabilidade.. Na figura 2.3 é mostrado o código responsável pela etapa em questão.



```
#120 a 140
colunas_desejadas = df.loc[:, '120.00'
:'140.06']
colunas_desejadas
```

	120.00	120.10	...	139.97	140.06
0	2	4	...	3	5
1	1	0	...	2	3
2	3	7	...	5	2
3	1	2	...	1	1
...	...	...	...	...	...

Figura 2.3: Filtragem de 120 km até 140 km

Na figura 2.4 encontra-se uma parte simples do código, no qual foi escrito uma estrutura que consegue fazer uma média geral dos ruídos por horário entre as altitudes de 120 km até 140 km.

```
#PRECISA FAZER A COLUNA SO DE MÉDIA
colunas_desejadas_media = colunas_desejadas
colunas_desejadas_media['media'] = colunas_desejadas_media.mean(axis = 1)
colunas_desejadas_media
```

	...	139.30	139.39	139.49	139.58	139.68	139.78	139.87	139.97	140.06	media
0	...	5	6	5	1	6	2	2	3	5	4.338095
1	...	5	2	3	1	4	2	2	2	3	2.566667
2	...	5	3	3	1	2	2	2	5	2	3.142857
3	...	1	4	0	0	1	4	2	1	1	1.952381
...	...	...	...	...	...	...	...	...	...	...	...
1506	...	0	0	0	0	0	0	0	0	0	0.000000
1507	...	0	0	0	0	0	0	0	0	0	0.000000
1508	...	0	0	0	0	0	0	0	0	0	0.000000
1509	...	0	0	0	0	0	0	0	0	0	0.000000

Figura 2.4: Tabela com as médias feitas

Na Figura 2.5, é apresentado um dos passos fundamentais para alcançar uma certeza estatística dos dados observados. Como mencionado anteriormente, para garantir a precisão dos dados na região dos 40 km de altitude, foi necessário realizar o cálculo da média dos ruídos. Nesse contexto, a Figura 2.5 exibe o código desenvolvido para gerar uma linha completa contendo essa média calculada. Em seguida, é apresentado um trecho breve que segmenta o quadro de dados até a altitude de 40 km e também extrai a tabela das médias, realizando uma subtração meticulosa para cada coluna correspondente das duas tabelas.

```
#APENAS MÉDIA
coluna_media = colunas_desejadas.loc[:, '139.97':'media']
coluna_media = coluna_media.drop(['139.97', '140.06'], axis=1)
coluna_media = coluna_media.T
coluna_media
#SUBTRAÇÃO DO RUÍDO NOS DADOS
sub_linha = coluna_media.iloc[0]
sub_total = primeiras_40_linhas.sub(sub_linha, axis = 1)
sub_total
```

	0	1	2	...	1500	1501	1502
media	4.338095	2.566667	3.142857	...	10.4	11.261905	11.104762

	0	1	2	...	1500	1501
0.10	2949.661905	2929.433333	2848.857143	...	5347.6	5357.738095
0.19	3319.661905	3292.433333	3260.857143	...	4405.6	4333.738095
0.29	3766.661905	3757.433333	3701.857143	...	3945.6	3972.738095
0.38	3344.661905	3245.433333	3256.857143	...	3259.6	3271.738095
...	...	...	...	...	...	...

Figura 2.5: Etapa de subtração

Na figura 2.6, foi realizada uma média geral das colunas, resultando em uma única coluna contendo essa média. Posteriormente, essa única coluna foi dividida em duas variáveis distintas, o que possibilitou a plotagem de um gráfico de linha.

```
#MATRIZ PARA O GRAFICO
media_total = pd.DataFrame(sub_total.mean(
axis = 1), columns=['MÉDIA'])
media_total = media_total.reset_index()
media_total = media_total.rename(
columns={'index':'ALTURA',
'MÉDIA':'MÉDIA'})

media_total
```

	ALTURA	MÉDIA
0	0.10	3433.522624
1	0.19	3834.39812
2	0.29	3891.757723
3	0.38	3228.002094
...	...	...

Figura 2.6: Etapa de subtração

Ao concluir o primeiro arquivo de códigos e finalizar o primeiro passo do projeto, na Figura 2.7 é apresentada a programação dedicada exclusivamente à plotagem de gráficos de linha, bem como à automatização do processo de salvamento dos gráficos gerados em um diretório específico no computador.

```

#GRAFICO FINAL
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter

plt.plot(media_total['MÉDIA'], media_total['ALTURA'])
plt.yticks([0,104,208,312,416])

def y_formatter(value, _):
    if value == 0:
        return "0"
    elif value == 104:
        return "10"
    elif value == 208:
        return "20"
    elif value == 312:
        return "30"
    elif value == 416:
        return "40"

plt.xlabel('Sinal (nº de fótons)')
plt.ylabel('Altura(Km)')

plt.gca().yaxis.set_major_formatter(FuncFormatter(y_formatter))

plt.savefig('C:/Users/Usuario/Desktop/aerosois/gráficos/K/Grafico_K20230807')

```

Figura 2.7: Código para plotar gráfico

Para possibilitar a plotagem do gráfico, foi necessário importar três bibliotecas adicionais: `Matplotlib.pyplot`, `Matplotlib.ticker` e `FuncFormatter`.

Matplotlib.pyplot é uma biblioteca de visualização de dados em Python, que facilita a criação de gráficos e plots de forma simples e intuitiva. *Matplotlib.ticker* permite personalizar os formatos dos rótulos nos eixos de um gráfico, enquanto *FuncFormatter* possibilita a criação de formatações personalizadas para rótulos nos eixos de gráficos em Python.

Após a importação dessas bibliotecas, foi criado um laço de repetição utilizando *if* e *elif* para garantir que os dados exibidos nos gráficos estivessem de acordo com as normas estabelecidas. Os resultados da plotagem dos gráficos para o comprimento de onda do Potássio (770 nm) e do Sódio (589 nm) estão ilustrados nas Figuras 2.11 e 2.12, respectivamente.

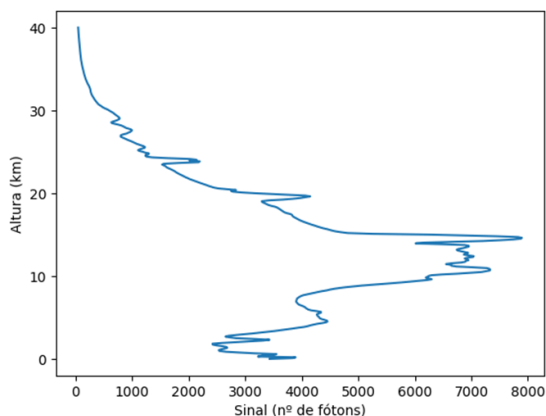


Figura 2.8: Perfil de aerossóis para 770 nm.

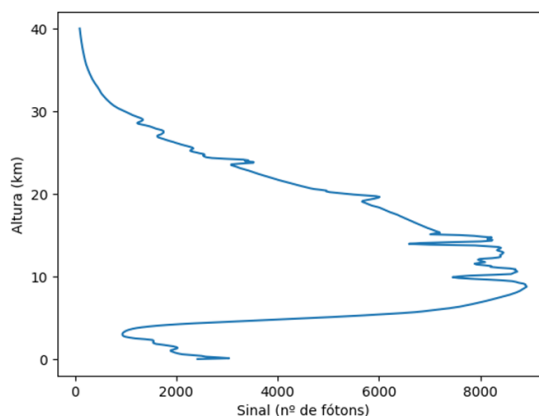


Figura 2.9: Perfil de aerossóis para 589 nm.

Os resultados obtidos da análise dos dados coletados durante o estudo dos aerossóis provenientes do vulcão Hunga Tonga foram altamente significativos, fornecendo informações valiosas sobre as propriedades e características dessas partículas na atmosfera. A análise dos gráficos revelou claramente a interferência causada pelas plumas de aerossóis na atmosfera, destacando especialmente as regiões entre 10 km e 20 km de altitude.

As próximas etapas do relatório envolvem a criação de um gráfico de contorno, visando uma análise mais precisa da atmosfera. Ao contrário dos gráficos apresentados nas figuras 2.8 e 2.9, o gráfico de contorno proporcionará uma visualização mais clara de anomalias na atmosfera. Além disso, será elaborado de forma a permitir uma análise abrangente, abrangendo desde o primeiro mês de observação (março de 2022) até o último registrado para este relatório (agosto de 2023).

média diária -> média mensal -> scatter -> gráfico final

A primeira etapa do processo foi pegar os dados separados e filtrados no excel, dito anteriormente, e fazer uma média diária do determinado mês analisado.

```

import pandas as pd
# Defina as variáveis para o ano, mês e dia
ano = 2023
mes = '08'
dia = '07'
# Construa o caminho do arquivo usando as variáveis
caminho_arquivo = f"c:/Users/Usuario/Desktop/ASCII/{ano}/{ano}{mes}{dia}/Na{ano}{mes}{dia}_photocount.dat"

# Leia o arquivo CSV
df = pd.read_csv(caminho_arquivo, delimiter='\\s+')
# Configure a exibição máxima de linhas
pd.options.display.max_rows = 8
# Remova a coluna '0.00'
df = df.drop(['0.0000'], axis=1)

#TRANSPOR MATRIZ
df_transposta = df.T
#df_transposta = df_transposta.drop(['0.00'], axis = 0)
primeiras_40_linhas = df_transposta.head(418)
primeiras_40_linhas
#120 a 140
colunas_desejadas = df.loc[:, '120.0000':'140.0640']
colunas_desejadas

#PRECISA FAZER A COLUNA SO DE MÉDIA
colunas_desejadas_media = colunas_desejadas
colunas_desejadas_media['media'] = colunas_desejadas_media.mean(axis = 1)
colunas_desejadas_media

#APENAS MÉDIA
coluna_media = colunas_desejadas.loc[:, '139.9680':'media']
coluna_media = coluna_media.drop(['139.9680', '140.0640'], axis=1)
coluna_media = coluna_media.T
coluna_media

#SUBTRAÇÃO DO RUIDO NOS DADOS
sub_linha = coluna_media.iloc[0]
sub_total = primeiras_40_linhas.sub(sub_linha, axis = 1)
sub_total

#MATRIZ PARA O GRAFICO
media_total = pd.DataFrame(sub_total.mean(axis = 1), columns=['MÉDIA'])
media_total = media_total.reset_index()
media_total = media_total.rename(columns={'index':'ALTURA', 'MÉDIA':'MÉDIA'})

media_total

```

Figura 2.9 : Código de criação da média diária

Na figura 2.9, mostra a seção do código onde há uma significativa repetição do que foi mostrado na etapa de criação do gráfico de linha. As primeiras linhas do

código introduzem uma solução para automatizar a busca de arquivos no PC, exigindo a definição de variáveis para ano, mês e dia, e a construção de um caminho de arquivo correspondente. Em seguida, ocorre a leitura do arquivo CSV, configurando a exibição de um número máximo de linhas e removendo a coluna '0.00'.

Posteriormente, são replicados os códigos utilizados na criação do gráfico de linha. Isso inclui a transposição da matriz, a segmentação dos ruídos presentes na quilometragem entre 120 e 140, a criação de uma coluna exclusiva para as médias e a subsequente subtração dos ruídos. Dessa forma, na parte final do código, é possível observar a criação da média diária para um determinado mês.

Essa repetição de passos pode indicar uma oportunidade para refatoração do código, visando eliminar redundâncias e torná-lo mais eficiente e fácil de manter. Na Figura 2.10 está a média feita.

	ALTURA	MÉDIA
0	0.0960	3585.305253
1	0.1920	4272.483031
2	0.2880	4518.295729
3	0.3840	4330.575094
...	...	...
414	39.8400	77.417952
415	39.9360	76.051285
416	40.0320	74.268745
417	40.1280	73.065571

Figura 2.10: Média diária

Após calcular as médias diárias de todos os dias analisados, o próximo passo foi calcular uma média mensal dessas médias diárias. Esse procedimento envolve agregar as médias diárias de cada mês e calcular a média desses valores agregados.

```

import pandas as pd
import glob
import calendar

diretorio_base = "c:/Users/Usuario/Desktop/ASCII/2023/"
mes = 8
padrao_arquivo = "Na*_photocount.dat"

dataframes = []

for dia in range(1, calendar.monthrange(2023, mes)[1] + 1):
    data_variavel = f"2023{mes:02d}{dia:02d}"
    caminhos = glob.glob(f"{diretorio_base}/{data_variavel}/{padrao_arquivo}")

    for caminho in caminhos:
        df = pd.read_csv(caminho, delimiter='\\s+')
        df = df.drop(['0.0000'], axis=1)
        dataframes.append(df)

df_media = sum(dataframes) / len(dataframes)
df_media

```

Figura 2.11

Para automatizar a busca de arquivos novamente, implementou-se um laço `for` no código. Esse laço possibilitou que o dataframe de saída fosse formatado de maneira adequada desde o início, facilitando assim a continuidade do processo estatístico de análise dos dados. Essa abordagem permitiu lidar de forma eficiente com múltiplos arquivos de entrada, garantindo uma análise sistemática e completa dos dados ao longo do tempo.

A etapa seguinte, demonstrada na figura 2.12, envolveu a criação do código para calcular a média mensal. Ao analisar mais detalhadamente, percebe-se que este processo segue o mesmo passo a passo da criação do código para a média diária. Até mesmo as variáveis foram reaproveitadas, indicando uma abordagem consistente e coesa na manipulação dos dados. Essa consistência no processo de desenvolvimento sugere uma organização cuidadosa e uma compreensão clara dos requisitos do projeto.

```

#TRANSPOR MATRIZ
df_media_transposta = df_media.T
primeiras_40_linhas = df_media_transposta.head(418)
primeiras_40_linhas

#120 a 140
colunas_desejadas = df_media.loc[:, '120.0000':'140.0640']
colunas_desejadas

#PRECISA FAZER A COLUNA SO DE MÉDIA
colunas_desejadas_media = colunas_desejadas
colunas_desejadas_media['media'] = colunas_desejadas_media.mean(axis = 1)

#APENAS MÉDIA
coluna_media = colunas_desejadas.loc[:, '139.9680':'media']
coluna_media = coluna_media.drop(['139.9680', '140.0640'], axis=1)
coluna_media = coluna_media.T

#SUBTRAÇÃO DO RUÍDO NOS DADOS
sub_linha = coluna_media.iloc[0]
sub_total = primeiras_40_linhas.sub(sub_linha, axis = 1)
df = sub_total
df

#MATRIZ DO MÊS
media_total = pd.DataFrame(df.mean(axis = 1), columns=['MÉDIA'])
media_total = media_total.reset_index()
media_total = media_total.rename(columns={'index': 'ALTURA', 'MÉDIA': 'MÉDIA'})

media_total

```

Figura 2.12: Média mensal feita

Para concluir a etapa de cálculo das médias, a Figura 2.13 apresenta o código final para o salvamento automático em uma pasta do computador. Na última linha da imagem, é evidente a mensagem indicando que o arquivo foi salvo com sucesso. Essa implementação garante que os resultados obtidos sejam armazenados de forma organizada e acessível para referência futura, proporcionando uma conclusão satisfatória para essa fase do processo de análise de dados.

```

from datetime import datetime

data_folder = "C:/Users/Usuario/Desktop/aerosois/média mensal/Na"

hoje = datetime.now()

import locale
locale.setlocale(locale.LC_TIME, 'pt_BR')

ano = 2023
mes = 8

nome_mes = calendar.month_name[mes]

file_name = f"media_{nome_mes}_Na_{ano}.csv"

file_path = f"{data_folder}/{file_name}"

media_total.to_csv(file_path, sep=' ', index=False)

print(f"O DataFrame foi salvo em '{file_path}'.")

```

O DataFrame foi salvo em 'C:/Users/Usuario/Desktop/aerosois/média mensal/Na'.

Figura 2.13: Salvamento do dataframe

Na Figura 2.14, apresenta-se uma nova etapa do código, onde, após o cálculo da média mensal dos dados de dispersão, é gerado o gráfico de contorno final. Nesta figura inicial, é mostrada a parte preliminar da programação. Utilizando um laço de repetição, foi possível localizar e acessar os arquivos no computador. Posteriormente, por meio de uma função do Python, concatenam-se todos os arquivos contendo as médias mensais dos dados analisados, resultando em uma tabela única e consolidada.

```
import pandas as pd
import numpy as np
import os
df = pd.read_csv ("C:/Users/Usuario/Desktop/aerosois/Scattering/Não Saturados/K/K20220517_scater_ratio.dat",
delimiter='\\s+')
df = df.iloc[4:]
df = df.reset_index(drop=True)
altura_df = pd.DataFrame({'altura': df.iloc[:, 0]})

diretorio = 'C:/Users/Usuario/Desktop/aerosois/media_mensal_scatter/K_final'

dfs_para_concatenar = []
for arquivo in os.listdir(diretorio):
    caminho_completo = os.path.join(diretorio, arquivo)
    if arquivo.endswith('.csv'):
        df_temp = pd.read_csv(caminho_completo)
        dfs_para_concatenar.append(df_temp)
resultado_final = pd.concat([altura_df] + dfs_para_concatenar, axis=1)

df=resultado_final
novas_colunas = {
    'nov/2022': {'indice': 9},
    'fev/2023': {'indice': 12},
    'jul/2023': {'indice': 17}
}

for nome_coluna, config in novas_colunas.items():
    df.insert(config['indice'], nome_coluna, 0.00)
```

Figura 2.14 : Primeira etapa do gráfico final

Após essa etapa, o desenvolvimento focou na finalização do código, especificamente na plotagem do gráfico de contorno. Durante essa fase, foram configuradas as informações a serem exibidas no gráfico final, incluindo as legendas dos eixos x e y, a coloração do gráfico de contorno, o valor máximo representado, entre outros parâmetros, conforme ilustrado na Figura 2.15.

```

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

altura = df.iloc[:, 0]
df_data = df.iloc[:, 1:]
df_data = df_data.apply(pd.to_numeric, errors='coerce')
data = df_data.to_numpy()
mask = np.isnan(data)
nomes_colunas = list(df_data.columns)
y_labels = ['38.83', '34.03', '29.23', '24.43', '19.63', '14.83', '0.00']

num_levels = 12

plt.figure(figsize=(15, 8))
contour = plt.contourf(data, cmap='jet', levels=num_levels, vmin=data.min(), vmax=data.max())
plt.colorbar(contour)
plt.yticks(np.linspace(0, len(altura) - 1, len(y_labels)), y_labels[::-1])

plt.xlabel('Colunas')
plt.ylabel('Altura')
plt.title('K')

plt.xticks(range(len(nomes_colunas)), nomes_colunas, rotation=45, ha='right')
plt.xlim(0, len(nomes_colunas) - 1)
plt.ylim(0, len(altura))
plt.grid(True, linestyle='--', alpha=0.7)

caminho_do_diretorio = 'C:/Users/Usuario/Desktop/aerosois/grafico_contorno/K'
nome_do_grafico = 'Gráfico_K.png'
plt.savefig(f"{caminho_do_diretorio}/{nome_do_grafico}")

plt.show()

```

Figura 2.15 : Segunda etapa do gráfico final

Dito isso, todo esse passo a passo demonstrado foi para chegar na conclusão, que seria os gráficos de contorno, tanto do potássio (K) quanto do sódio (Na) mostrado nas figuras 2.16 e 2.17.

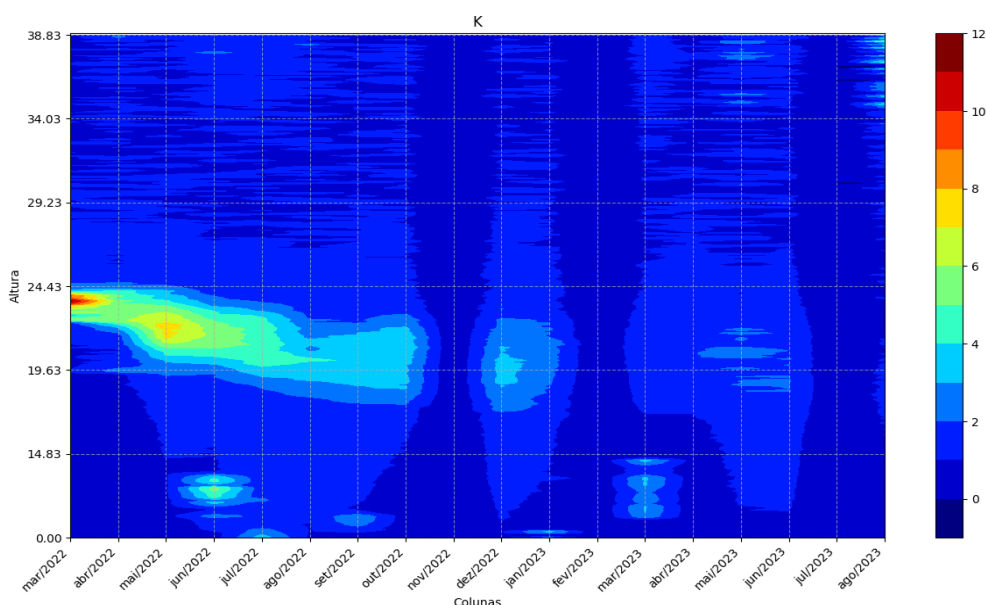


Figura 2.16 : Gráfico de contorno Potássio (K)

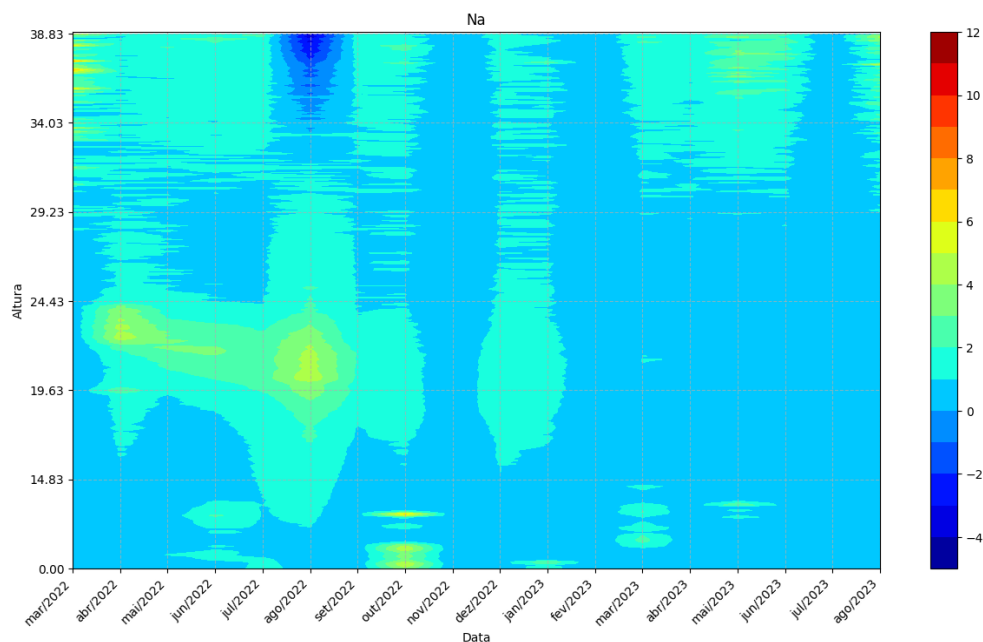


Figura 2.17: Gráfico de contorno do Sódio

Os resultados obtidos a partir da análise dos dados coletados durante o estudo dos aerossóis provenientes do vulcão Hunga Tonga foram significativos e forneceram informações valiosas sobre as propriedades e características dessas partículas na atmosfera, onde a análise dos gráficos deixa claro a interferência que as plumas de aerossóis causaram na atmosfera, principalmente nas regiões entre 10 km e 20 km.

3. CONCLUSÃO

Este relatório descreve o desenvolvimento das funcionalidades do sistema de processamento de dados LiDAR durante o período de observação dos aerossóis provenientes do vulcão Hunga Tonga, bem como a análise dos dados resultantes. Através do uso do radar de laser do INPE e da coleta de dados sobre a poeira atmosférica proveniente da erupção vulcânica, foi possível obter informações valiosas sobre a presença de aerossóis na estratosfera.

A utilização do programa em Python no ambiente do Visual Studio proporcionou uma abordagem eficiente para a leitura automatizada dos dados e a geração de gráficos, facilitando a análise dos resultados. As funcionalidades implementadas no programa permitiram a manipulação dos dados de K e Na, bem como a criação rápida e dinâmica de visualizações gráficas.

Através da compreensão do funcionamento do LiDAR e das estruturas da atmosfera, foi possível interpretar os dados coletados e obter uma compreensão mais aprofundada dos fenômenos observados. A filtragem dos valores correspondentes à faixa de altitude de 20 a 40 km possibilitou uma melhor visualização das camadas de aerossóis e identificou as altitudes específicas onde elas ocorreram. No geral, este estudo demonstrou a importância do LiDAR na análise de aerossóis estratosféricos e apresentou uma abordagem eficaz para o processamento e interpretação dos dados coletados. As informações obtidas contribuem para o avanço do conhecimento científico nessa área e podem ser aplicadas em estudos futuros relacionados à atmosfera e aos efeitos das erupções vulcânicas.

4. REFERÊNCIAS

BATISTA, P. P.; CLEMESHA, B.R.; SIMONICH M. D. OBSERVATIONS OF ATMOSPHERIC DENSITY AND TEMPERATURE BETWEEN 35 AND 70 KM BY RAYLEIGH LIDAR AT SÃO JOSÉ DOS CAMPO, SP. 13. vol. São Paulo: 1995

CLEMESHA, B.R.; SIMONICH M. D. COMPARISON BETWEEN THE EL CHICHON AND PINATUBO AEROSOL, SP. São Paulo: 1990

LUTZ, M.; ASCHER, D. APRENDENDO PYTHON, RS. 2ª edição. Porto Alegre: 2003

KERN, C.; ARELLANO, S.; CAMPION, R.; HIDALGO, S.; KAZAHAYA, R. REMOTE SENSING OF VOLCANIC GAS EMISSION FROM THE GROUND, AIR, AND SPACE. Roma: 2023

PÉREZ, B. A. AEROSOL IMPACTS ON SUBTROPICAL LOW-LEVEL CLOUDS: A SATELLITE AND MODELING PERSPECTIVE. Stockholm: 2023

MILLÁN, L.; SANTEE, M. L.; LAMBERT, A.; LIVESEY, N. J.; WERNER, F.; SCHWARTZ, M. J.; PUMPHREY, H. C.; MANNEY, G. L.; WANG, Y.; SU, H.; WU, L. ; READ, W. G.; FROIDEVAUX, L. THE HUNGA TONGA-HUNGA HA'APAI HYDRATION OF THE STRATOSPHERE, CA. Pasadena: 2022