Projeto de Curso

Estatística e Modelos Probabilísticos

2023.2

Aluno: Nick Hermogenio Correa

DRE: 119024656

Rio de Janeiro, RJ

# 1. Introdução

Projeto realizado como parte da avaliação da disciplina de Estatística e Modelos Probabilísticos oferecida na UFRJ no período de 2023.2 para o curso de Engenharia de Computação e Informação. O código utilizado na realização do projeto pode ser encontrado na íntegra no seguinte link: <https://github.com/NickHCorrea/TrabalhoProbest>. O projeto foi feito utilizando a linguagem python e bibliotecas math, numpy, pandas, matplotlib e scipy

# 2. Estatísticas Gerais

## Dataset

Definição dos datasets utilizando a biblioteca Pandas

Texto

Descrição gerada automaticamente

## Histogramas

* Obtenção

Texto

Descrição gerada automaticamente

Foi calculado o número de bins utilizando o método de Sturges. Então a função histograma da biblioteca numpy foi utilizada passando como parâmetro o número de bins. O objetivo era receber em uma variável os intervalos correspondentes no dataset ao número de bins. Então foi plotado um resultado utilizando a biblioteca matplotlib.

* Resultados

Gráfico, Histograma

Descrição gerada automaticamenteGráfico, Histograma

Descrição gerada automaticamenteGráfico, Histograma

Descrição gerada automaticamenteGráfico, Histograma

Descrição gerada automaticamente

* Análise

Podemos observar que para a Smart TV é mais frequente ter taxa de upload/download baixa ou nula, enquanto para o Chrome Cast temos a concentração de download por volta de 104~105 bps e de upload por volta de 103~104 bps.

Uma possível melhoria para o provedor seria focar em prover uma maior qualidade de banda para Chromecasts do que para Smart Tvs, pois eles são os mais utilizados.

## Funções Distribuição Empírica

* Obtenção

Texto

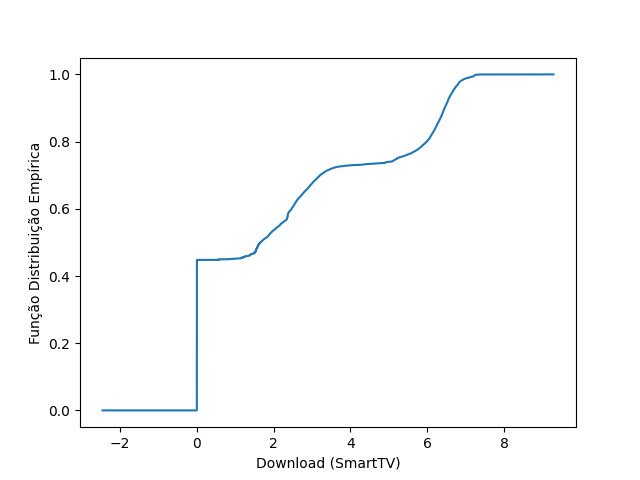
Descrição gerada automaticamente

A função distribuição empírica foi obtida utilizando a função ecdf da biblioteca scipy. O resultado então foi plotado utilizando a biblioteca matplotlib.

* Análise

Gráfico, Gráfico de linhas

Descrição gerada automaticamenteGráfico, Gráfico de linhas

Descrição gerada automaticamenteGráfico

Descrição gerada automaticamente

Podemos observar que há um pico tanto para download quanto upload da Smart Tv quando a taxa é da ordem de 100, de forma que aproximadamente 40% de todos os dados estão naquela faixa. Para o Chromecast temos um pico em 104 para o upload que representa por volta de 90% de todos os dados, ou seja, há uma concentração de dados para taxas a partir desse valor. Para o download o pico se desloca para 105.

Como há uma concentração de taxas maiores para Chromecasts, o provedor pode se preocupar em disponibilizar uma largura de banda maior para esses dispositivos. Enquanto isso, as taxas usadas pelas Smart Tvs são mais distribuídas, então há um uso de taxas pequenas também, ou seja, não terá tanto acúmulo para taxas mais altas quanto os Chromecasts.

## Box Plot

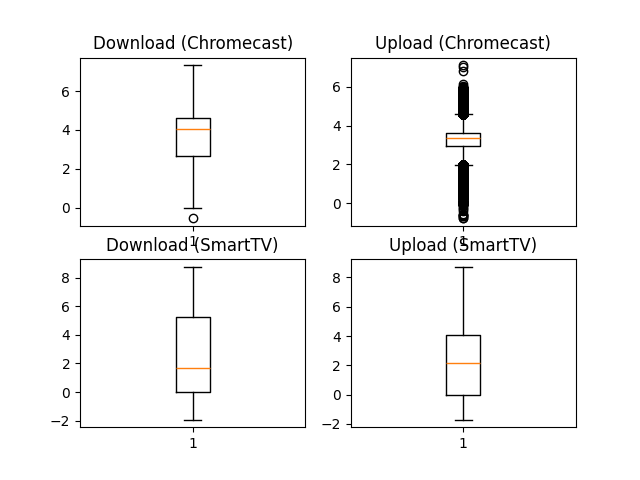
* Obtenção

Texto

Descrição gerada automaticamente

Os gráficos de boxplot foram obtidos utilizando a função boxplot da biblioteca matplotlib.

* Análise



Observando os boxes plot podemos ver que as taxas do Chromecast possuem menos variância e médias mais altas do que as taxas da Smart TV. Entretanto vemos que o upload do Chromecast possui muitos outliers, o que diz que essa variância na verdade não é tão pequena assim. Os dados de Smart TVs mostram uma média baixa com uma variância maior.

Esse gráfico endorsa as análises previamente realizadas.

## Média, variância e desvio padrão

* Obtenção

Texto

Descrição gerada automaticamente

* Análise

Chromecast

Download

Média: 3.799335488086478

Variância: 1.665979814323226

Desvio padrão: 1.2907284045542757

Upload

Média: 3.3496717251158694

Variância: 0.46160016892790384

Desvio padrão: 0.679411634377793

Smart TV

Download

Média: 2.350172639895904

Variância: 6.723920763635554

Desvio padrão: 2.5930524027939645

Upload

Média: 2.1565902037563727

Variância: 4.1130827523161635

Desvio padrão: 2.0280736555451244

Os dados numéricos obtidos confirmam as hipóteses levantadas analisando os gráficos.

# 3. Estatísticas por Horário

## Dataset

Definição dos datasets utilizando a biblioteca Pandas

Texto

Descrição gerada automaticamente

## Boxes plot

* Obtenção

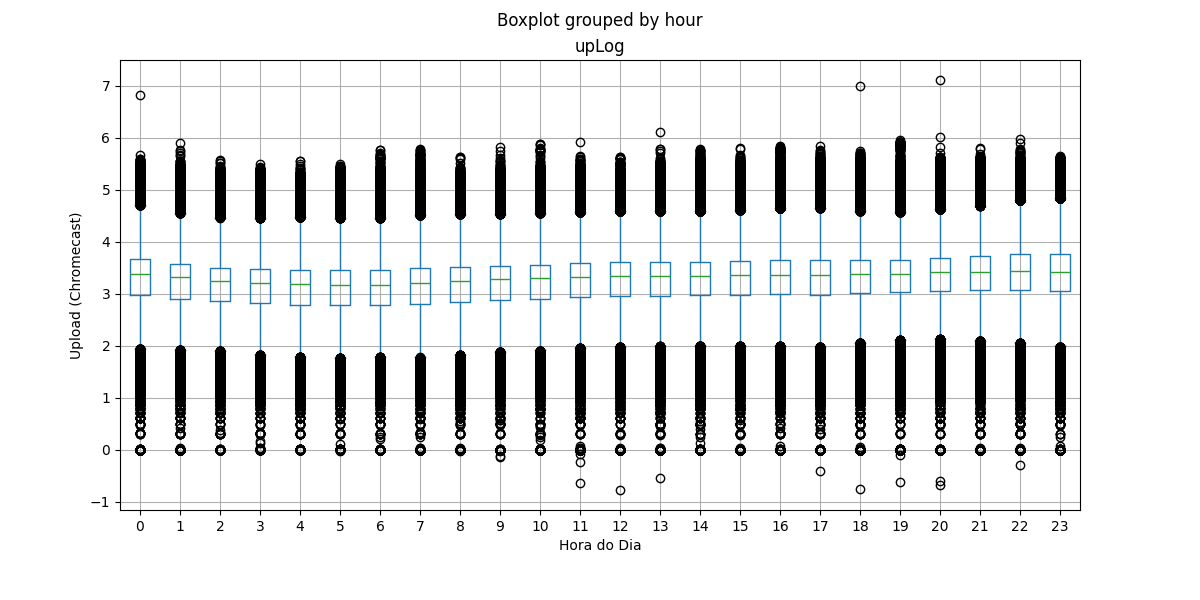
Texto

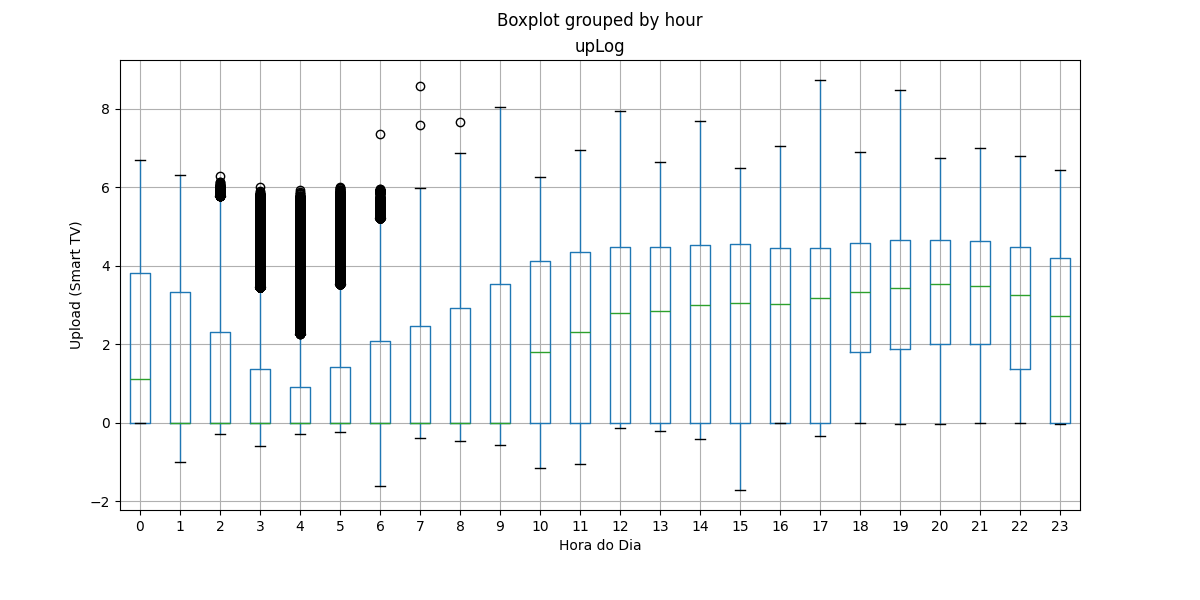
Descrição gerada automaticamente

Os gráficos de boxplot foram obtidos utilizando a função boxplot da biblioteca matplotlib.

* Análise

Gráfico

Descrição gerada automaticamenteGráfico, Gráfico de caixa estreita

Descrição gerada automaticamente

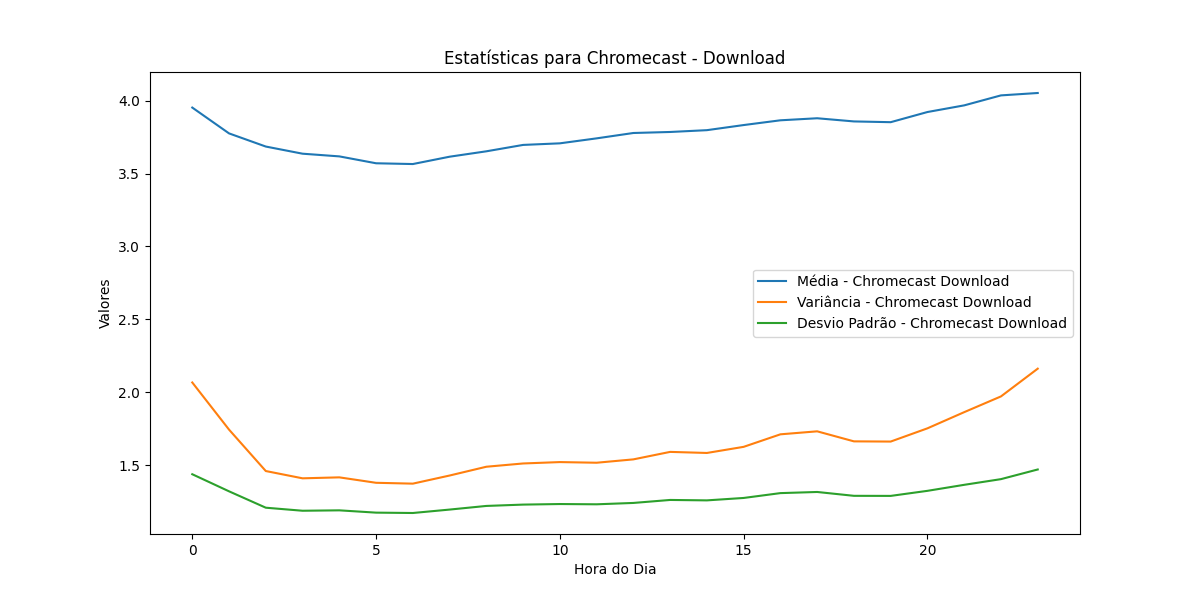
## Média, variância e desvio padrão

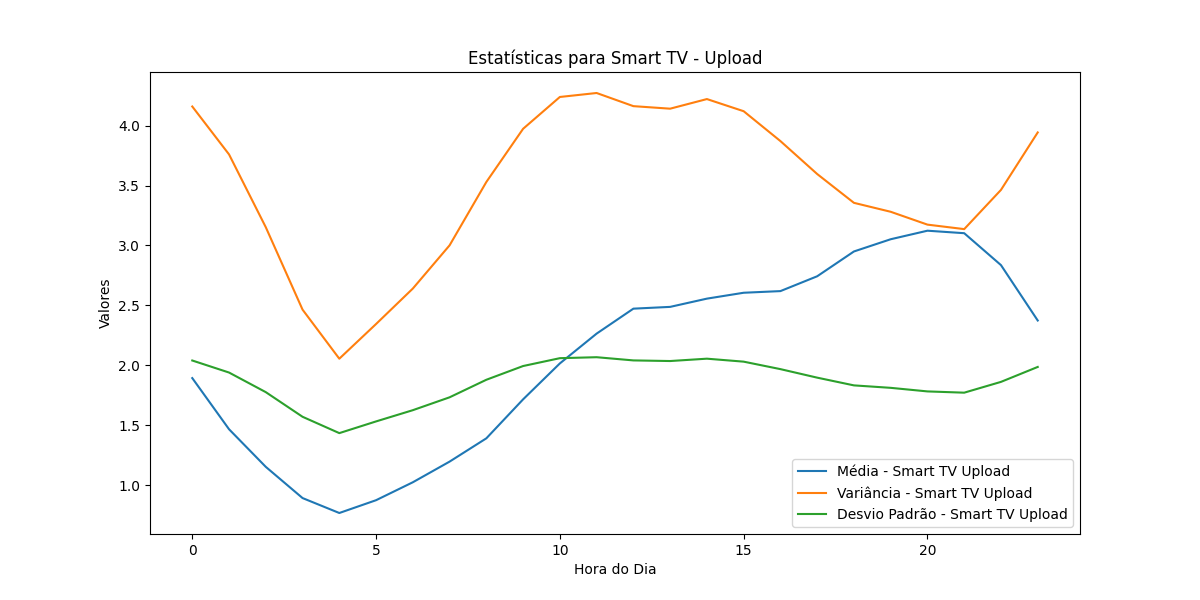
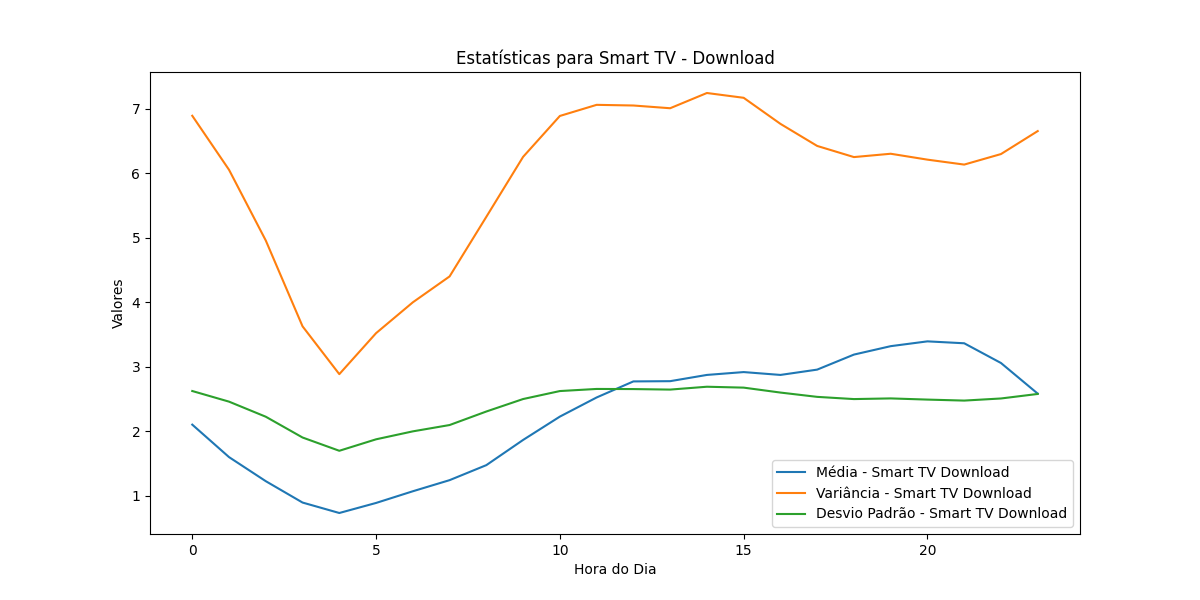
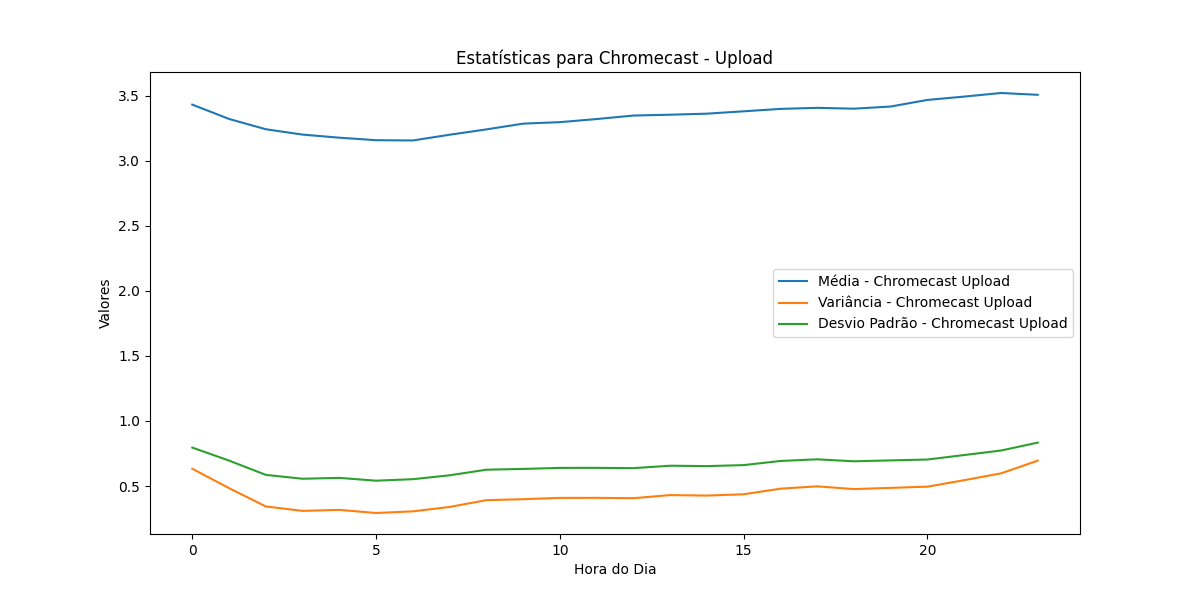
* Obtenção

Texto

Descrição gerada automaticamente

* Análise





# 4.Caracterizando horários com maior valor de tráfego

## Dataset

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

## Histogramas

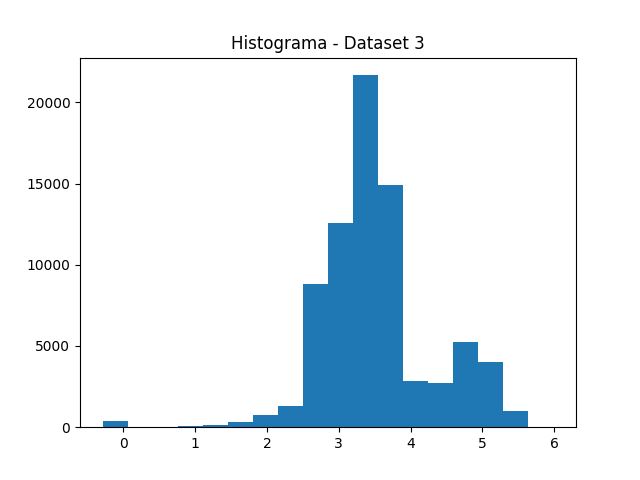
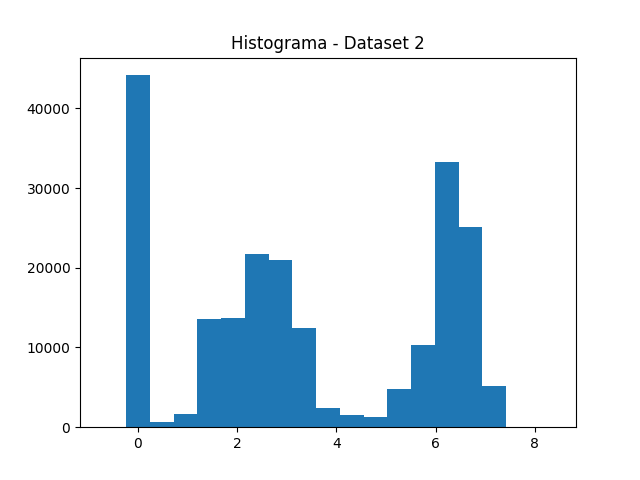
* Obtenção

Texto

Descrição gerada automaticamente

* Análise

Gráfico, Histograma

Descrição gerada automaticamenteGráfico, Histograma

Descrição gerada automaticamente

## QQPlot

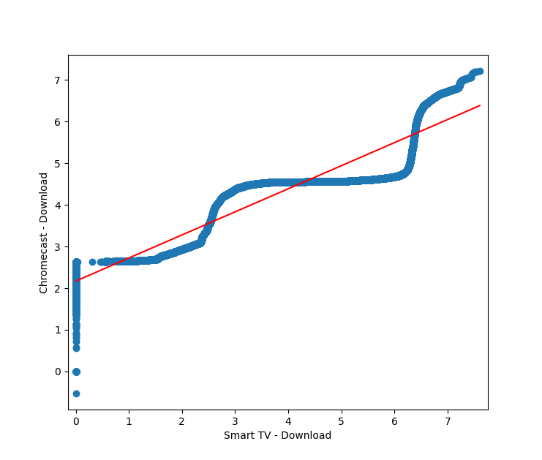
* Obtenção

Texto

Descrição gerada automaticamente

* Análise

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

## Conclusões

# 5.Análise da correlação entre as taxas de upload e download para os horários com o maior valor de tráfego

## Dataset

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

## Coeficiente de Correlação Amostral

* Obtenção

Texto

Descrição gerada automaticamente

* Análise

1-2

Coeficiente de Correlação: 0.9154767447400012

P-value: 0.0

3-4

Coeficiente de Correlação: 0.7919586388344998

P-value: 0.0

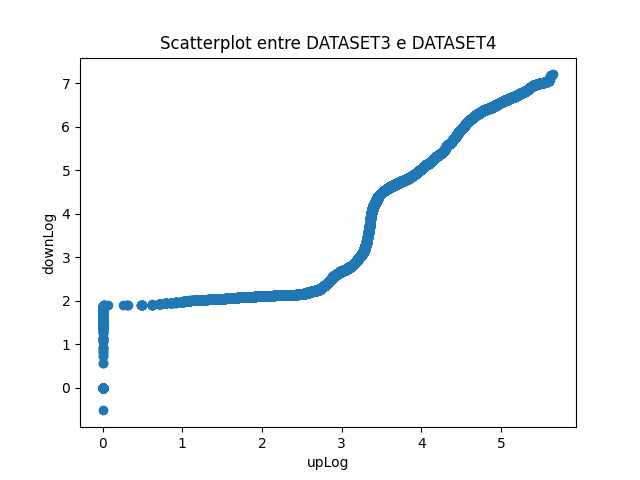
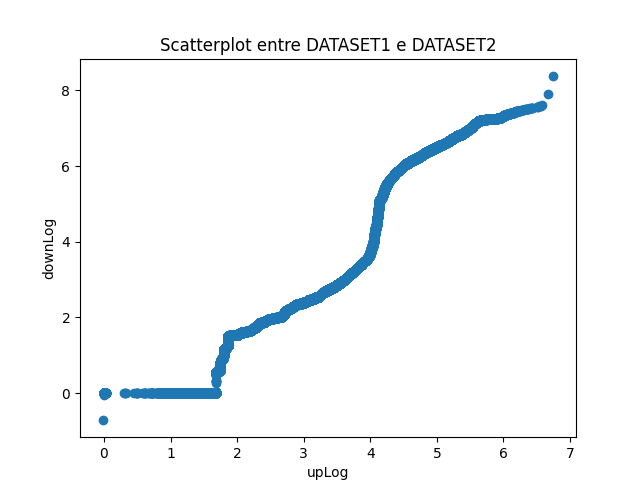
## Scatter Plot

* Obtenção

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

* Análise



## Conclusões

# 6.Conclusão Geral