UFRJ

Estatística e Modelos Probabilísticos - COE241

Relatório - Projeto do Curso
2022 - Professora Rosa Leão

Nome: Matheus Lomba de Rezende Conde
DRE: 117085216

1 - Dataset

Objetivo

O trabalho consiste em analisar um conjunto de dados referentes à taxa de envio e recebimento de dados de um provedor de internet de médio porte. O dataset, fornecido pela professora no formato csv, foi utilizado durante o trabalho de forma que pudéssemos trazer inúmeros métodos de avaliação da realidade da empresa a partir de uma grande gama de valores coletados no dia a dia.

Tratamento dos dados

O primeiro passo antes de qualquer análise foi realizar o tratamento do dataset de acordo com o especificado no documento do projeto. Para isto, e também já visando a análise futura, foi criado um código em Python utilizando o Google Colaboratory e que pode ser acessado pela seguinte url: https://github.com/matheuslomba/COE241ProbEst.

Uma vez o código inicializado e os datasets importados, foi realizado o tratamento das colunas referentes à quantidade de bytes de ambos upload e download, “**bytes_up**” e “**bytes_down**” respectivamente, para ambos datasets, criando-se novas colunas que receberam tais dados reescalonados para log10, assim como uma nova coluna de hora, valor retirado da coluna pré-existente “**date_hour**”.

Análise dos dados

Uma vez que os dados foram tratados e estão no formato correto, a análise foi iniciada de acordo com o proposto no roteiro.

2 - Estatísticas Gerais

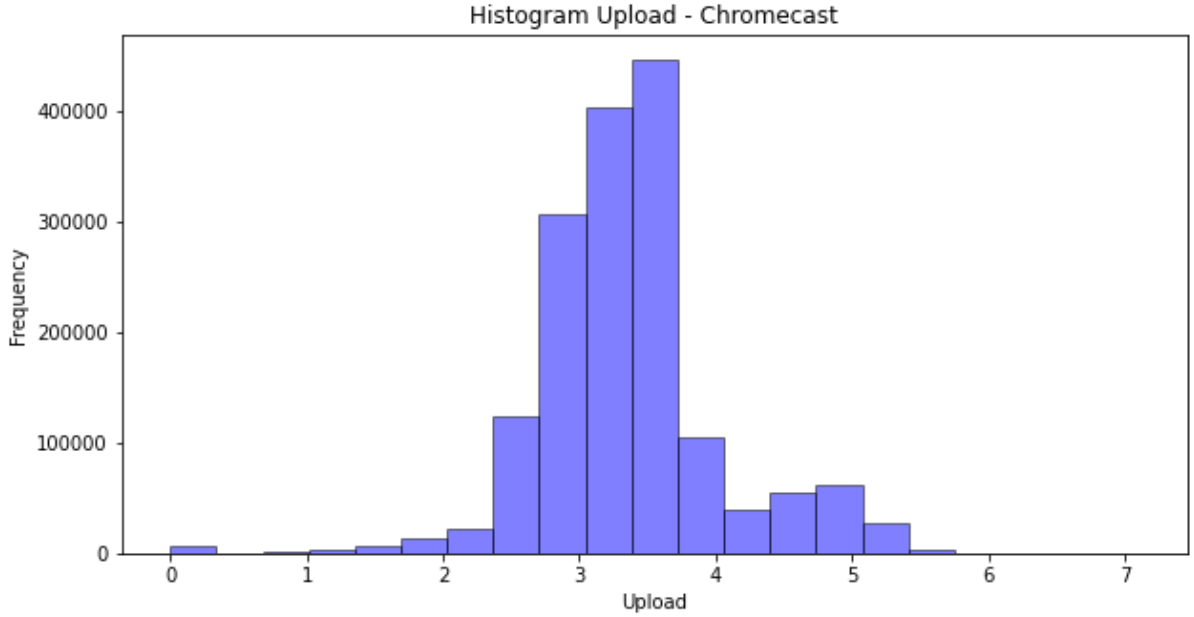
Para esta primeira etapa de análises foram utilizadas as bibliotecas “**numpy**” e “**matplotlib**”, assim como “**pandas**” que já estava sendo utilizado, para facilitar a obtenção dos dados e cálculos necessários para a realização da análise.

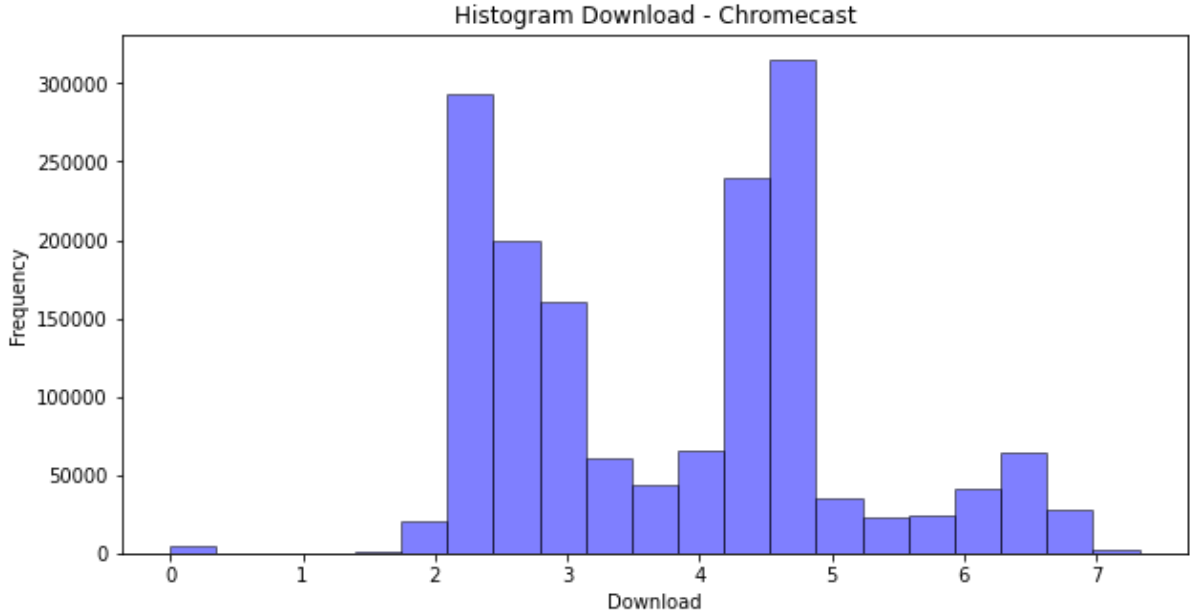
2.1 - Histograma

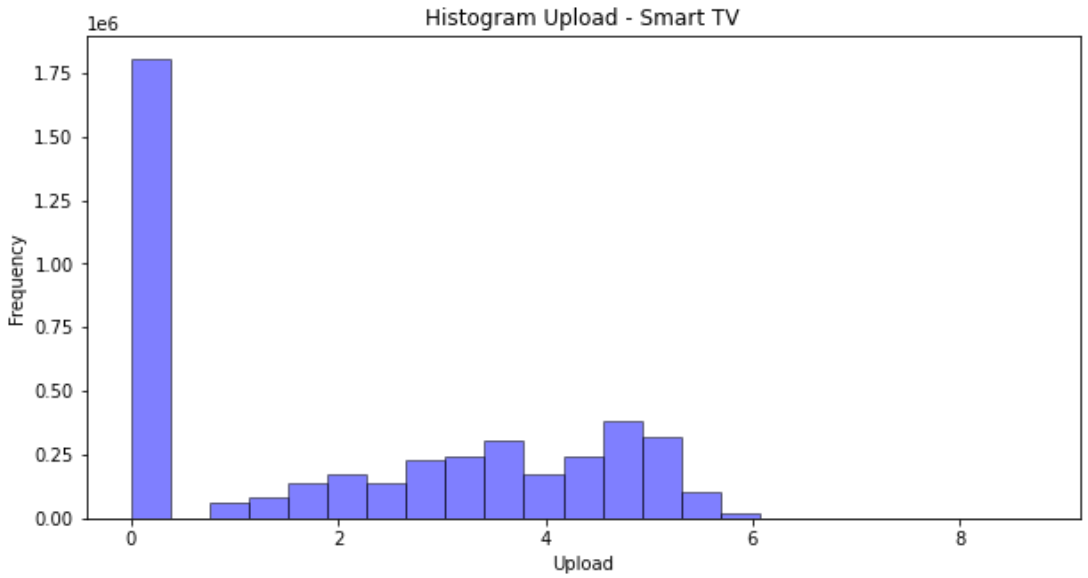
Os histogramas a seguir indicam a distribuição dos valores de Upload e Download de ambos datasets, sendo os valores utilizados no eixo y o log10 dos valores existentes inicialmente no dataset, assim como foi proposto no roteiro.
Além disso, o tamanho do bin utilizado na plotagem dos gráficos foi calculado através do método de Sturges pela seguinte fórmula:

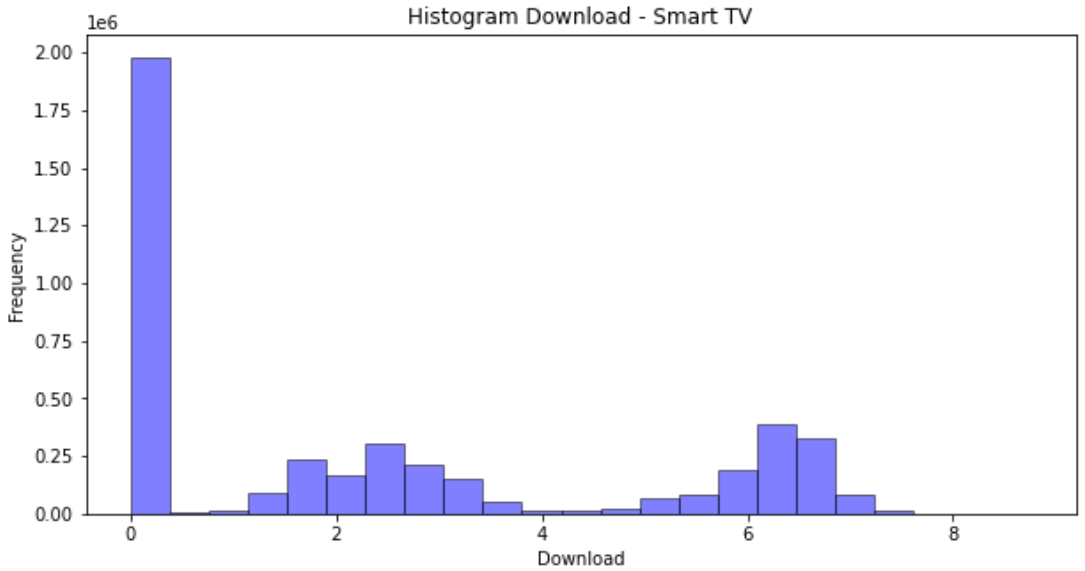
```
def calculate_binSize(data):
  binSize = int(1 + 3.322 * np.log10(len(data)))
  return binSize
```

Através disso, foi possível estimar o melhor valor possível para o tamanho do bin, de forma a otimizar e melhorar a visualização dos histogramas.


Histogram Upload - Chromecast
Frequency
Upload
0
100000
200000
300000
400000
0
1
2
3
4
5
6
7


Histogram Download - Chromecast
Frequency
Download
0
50000
100000
150000
200000
250000
300000
0
1
2
3
4
5
6
7



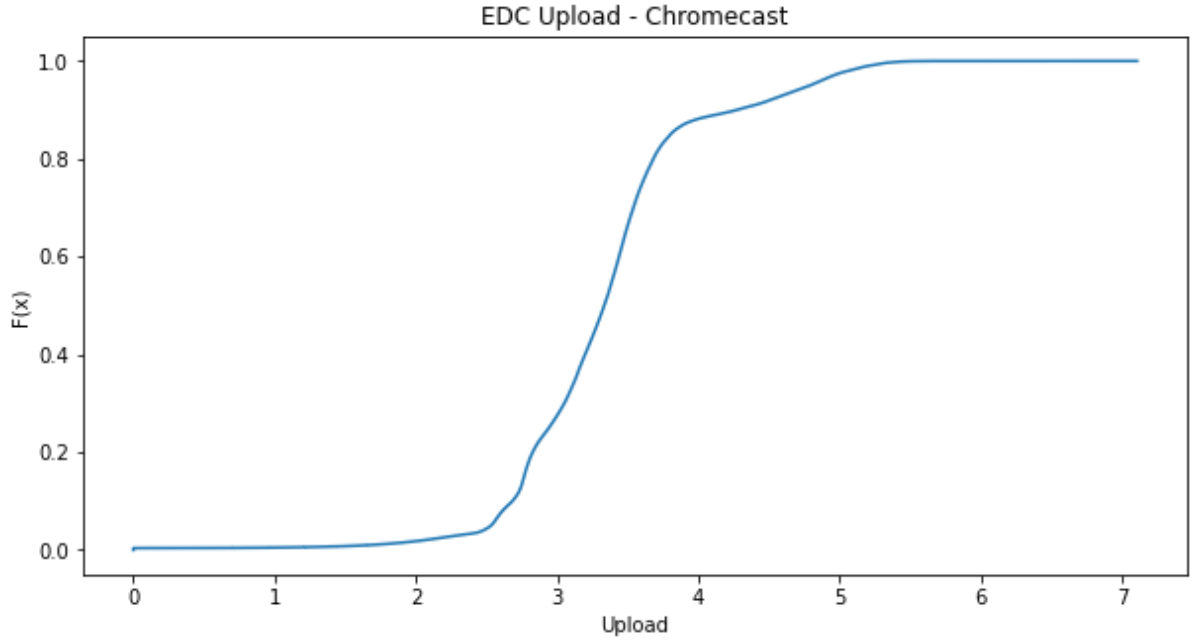


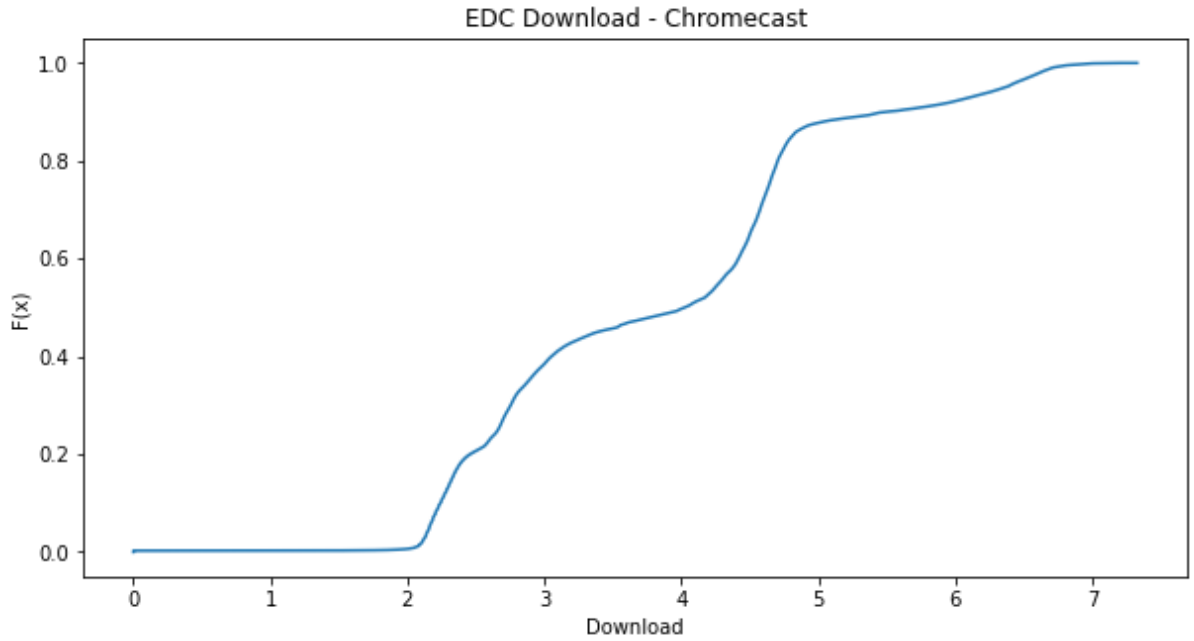
Nesta etapa conseguimos analisar alguns pontos muito interessantes. O primeiro seria a diferença entre o Chromecast e a Smart TV no que tange a quantidade de valores próximos a zero. Podemos facilmente observar que a Smart TV possui a grande maioria dos seus dados, para ambos download e upload, próximos do zero, o que pode ser traduzido ou num erro do dataset ou apenas num baixo uso da funcionalidade por parte dos usuários, preferindo se manter às opções “offline” que a televisão oferece ao invés do acesso à internet.

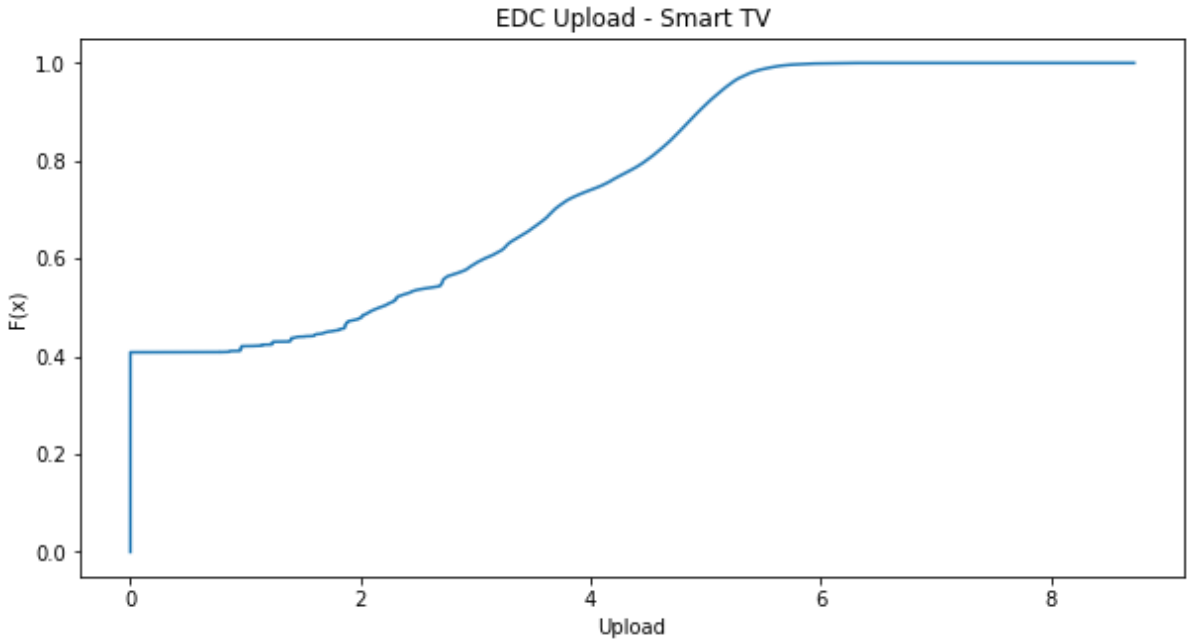
Outro ponto interessante é a similaridade entre os histogramas de download de ambos datasets, o que pode indicar uma relação entre os conteúdos baixados nessas duas plataformas, visto que ambas se relacionam ao mercado do entretenimento. Já os histogramas de upload não são tão similares, o que pode indicar que os dados enviados pela plataforma de volta às empresas são diferentes para cada uma delas.

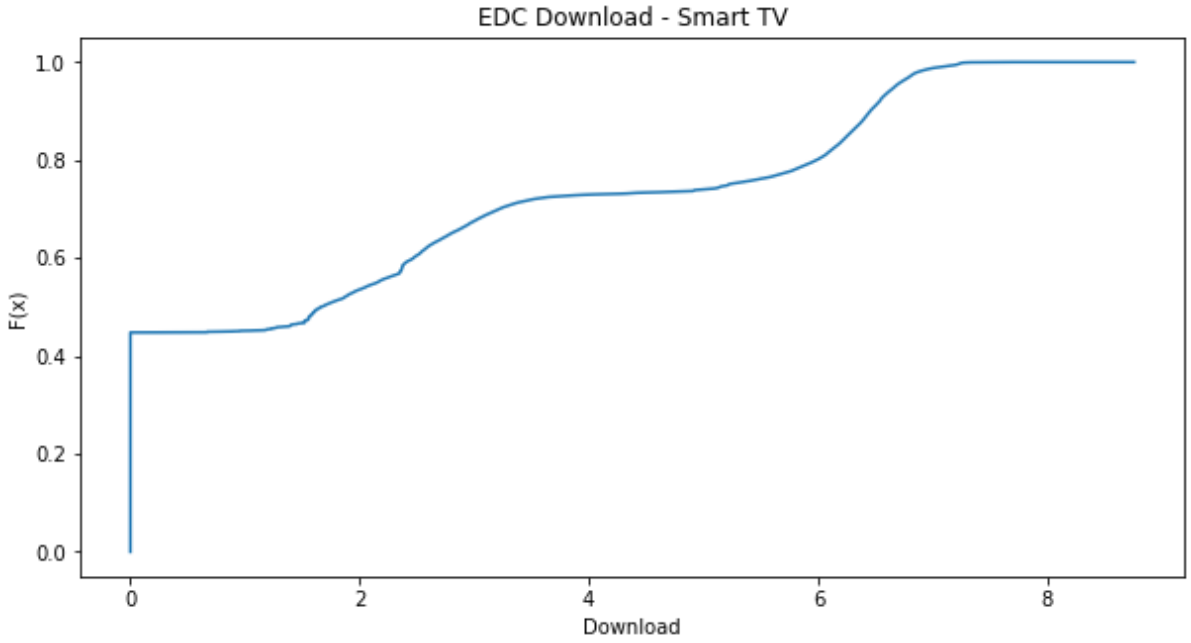
2.2 - Função Distribuição Empírica

Já as funções de distribuição empírica foram encontradas utilizando a biblioteca "**statsmodels**" através da função ECDF, que calcula automaticamente a distribuição do dataset fornecido. A partir disso foi necessário apenas criar o gráfico do resultado utilizando novamente a biblioteca "**matplotlib**", conforme está disponibilizado abaixo:





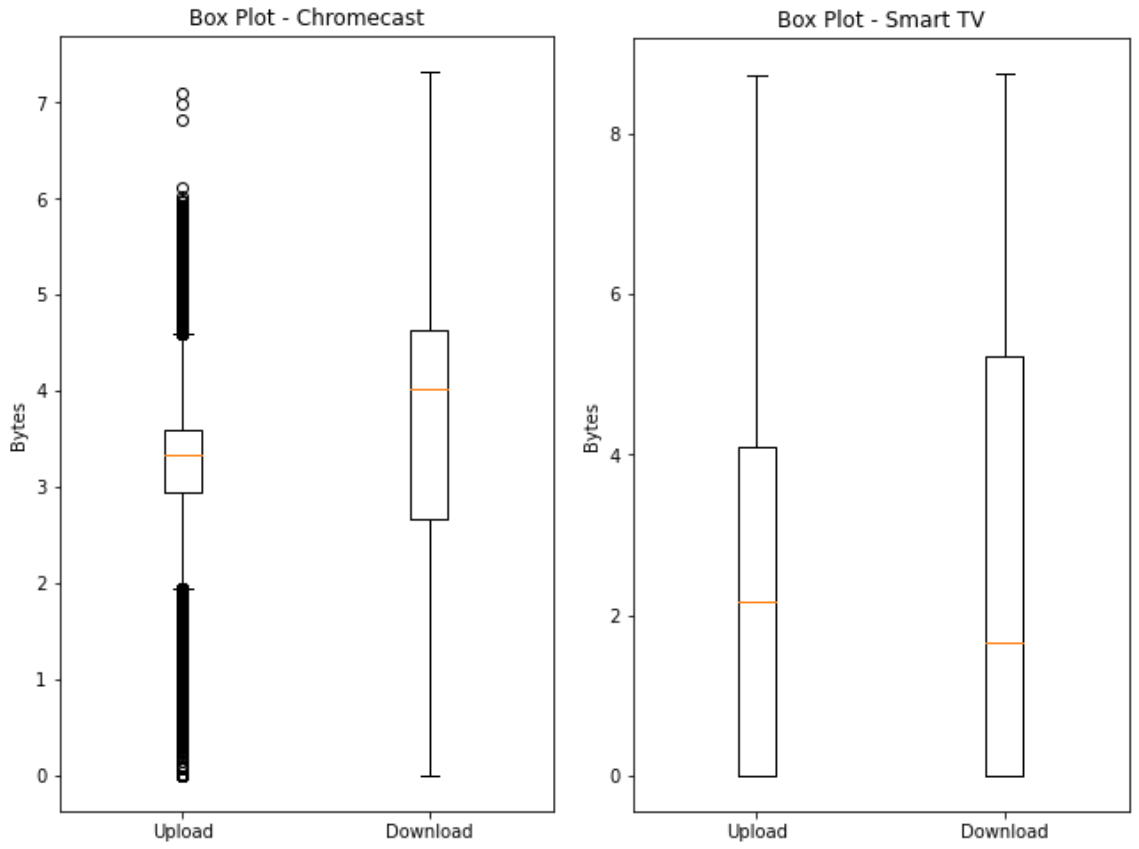




Aqui podemos observar os mesmos resultados discutidos sobre os histogramas, sendo ainda bem visível a grande quantidade de valores nulos para o dataset da Smart TV.

2.3 - Box Plot

Os box plots requisitados foram criados a partir da função “**boxplot**” da “**matplotlib**”, indicando lado a lado os dados de upload e download.



Os box plots acima nos indica também que, apesar da visível diferença apresentada nos gráficos anteriores, as medianas de ambos download e upload para cada dataset se mantém muito parecida, flutuando de 3 a 4 para o Chromecast e de aparentemente 1.7 a 2.2 para a Smart TV.
Além disso, também conseguimos identificar uma grande quantidade de dados discrepantes no Upload do Chromecast, indicando uma grande quantidade de outliers. Outro ponto interessante é a simetria de ambos box plots do Chromecast, ainda mais quando comparados com os referentes à Smart TV, que possuem um desvio negativo intenso devido à grande quantidade de dados com valor zero para as transmissões, assim como vimos nos gráficos anteriores.

2.4 - Média, Variância e Desvio Padrão

Os valores em questão foram calculados através das funções disponíveis no “**pandas**” mean, var e std, referentes à média, variância e desvio padrão respectivamente, e também foram divididos por dataset e por tipo de tráfego dos dados.

```
Mean Chromecast Upload = 3.35
Mean Chromecast Download = 3.8
Variance Chromecast Upload = 0.46
Variance Chromecast Download = 1.664
Standard Deviation Chromecast Upload = 0.678
Standard Deviation Chromecast Download = 1.29
```

```
Mean Smart TV Upload = 2.158
Mean Smart TV Download = 2.352
Variance Smart TV Upload = 4.11
Variance Smart TV Download = 6.721
Standard Deviation Smart TV Upload = 2.027
Standard Deviation Smart TV Download = 2.593
```

E por último, através dos dados calculados de média, variância e desvio padrão, conseguimos finalmente numerar com maior exatidão algumas das informações obtidas nos gráficos anteriores.
Também conseguimos perceber que a variância, assim como o desvio padrão das taxas de download de ambos os datasets são muito maiores que os valores referentes ao upload, o que também é visível através do histograma.
Ao mesmo tempo, os valores de média parecem bastante uniformes dentro dos datasets, mas possuindo grande diferença entre eles, sendo a média de ambos download e upload do Chromecast consideravelmente maior que o da Smart TV. Inclusive, exceto pela média, conseguimos identificar que os valores de variância e desvio padrão da Smart TV são consideravelmente maiores que o do Chromecast, confirmando ainda mais o impacto da grande quantidade de valores nulos.

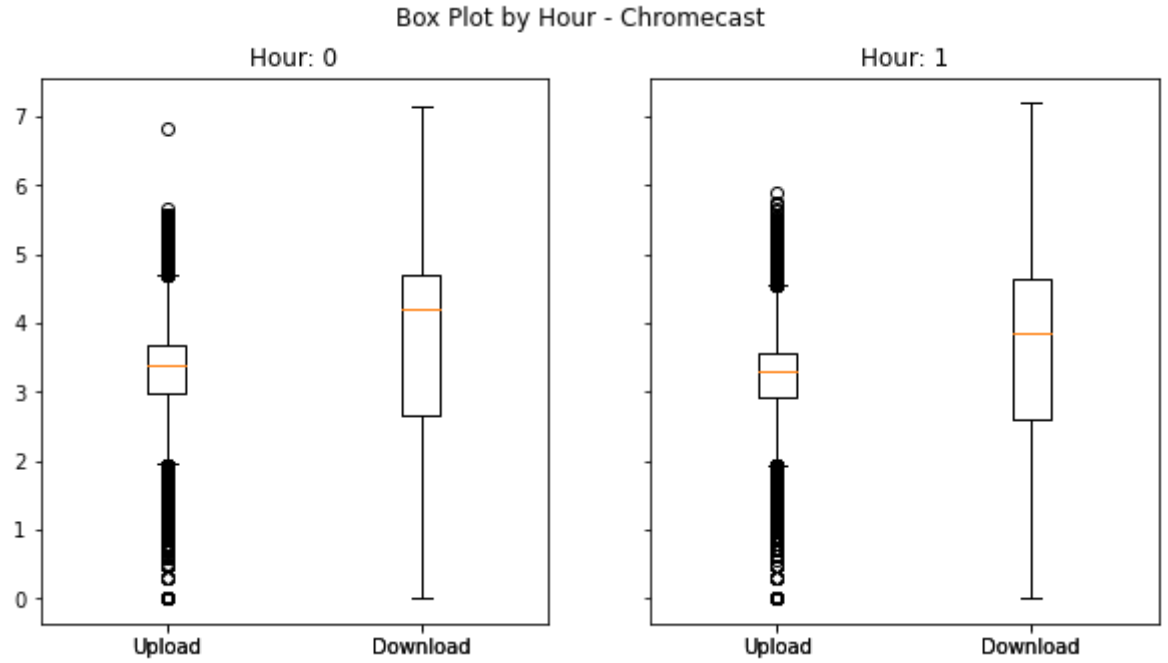
3 - Estatísticas por horário

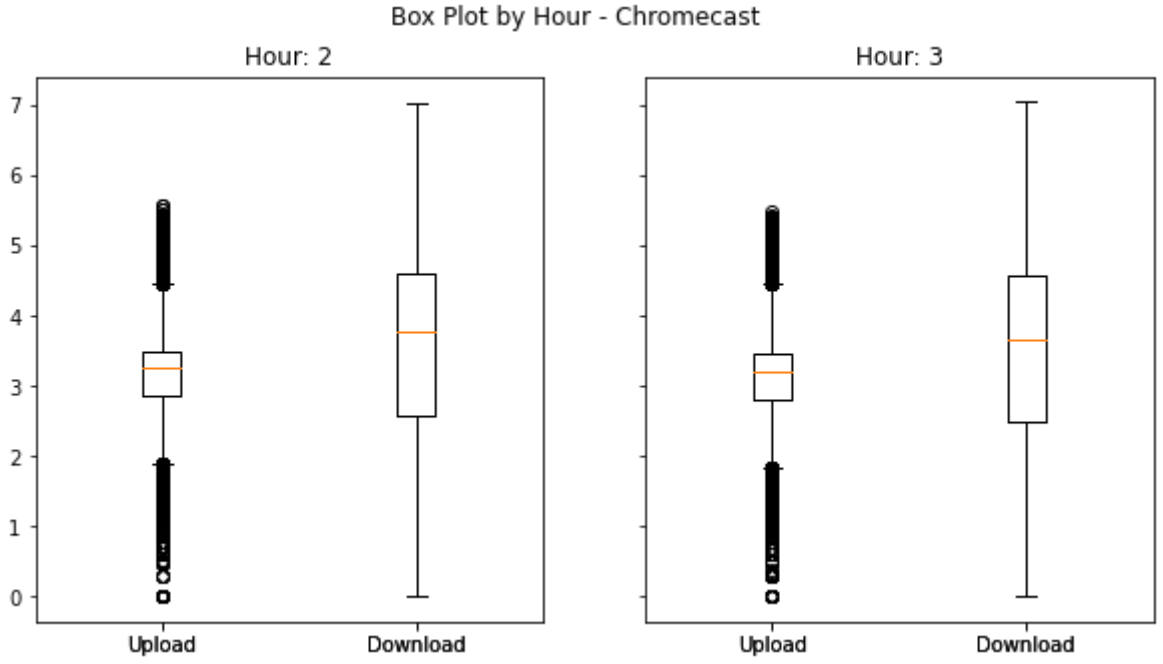
Para esta etapa do estudo foi criada uma nova coluna nos datasets com o objetivo de armazenar a hora em que o dado foi coletado. A nova coluna, chamada de “**hour**” foi adicionada utilizando como base os dados existentes em outra coluna já existente “**date_hour**”, de onde foi retirado apenas o valor necessário através da biblioteca ”**pandas**”.

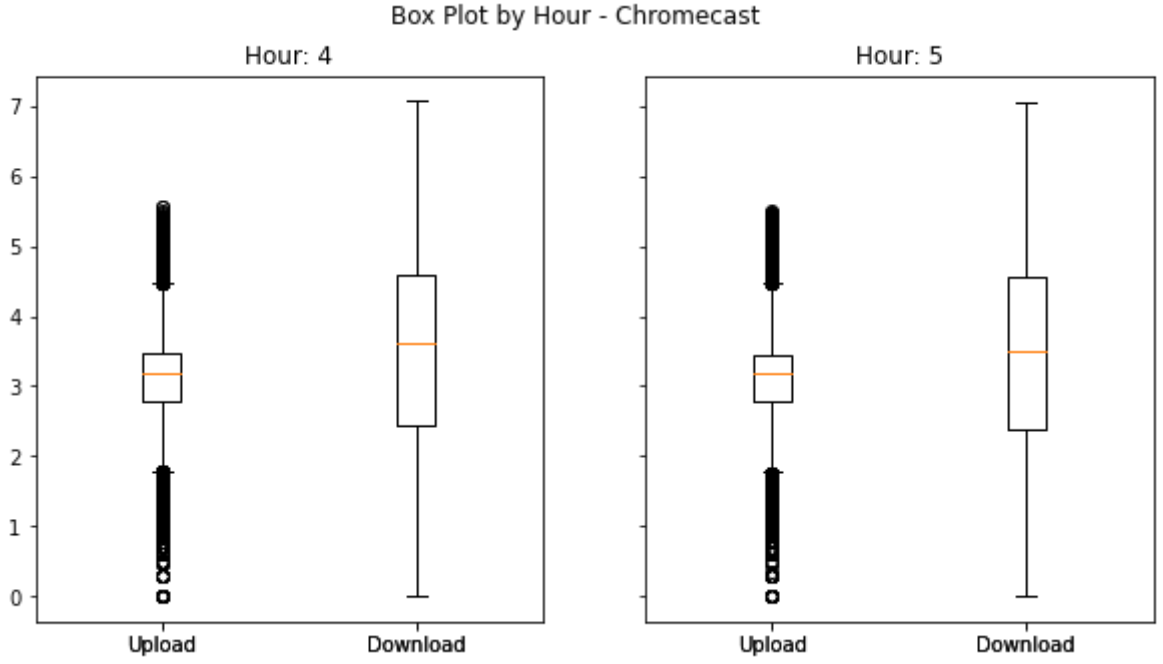
3.1 - Box Plot

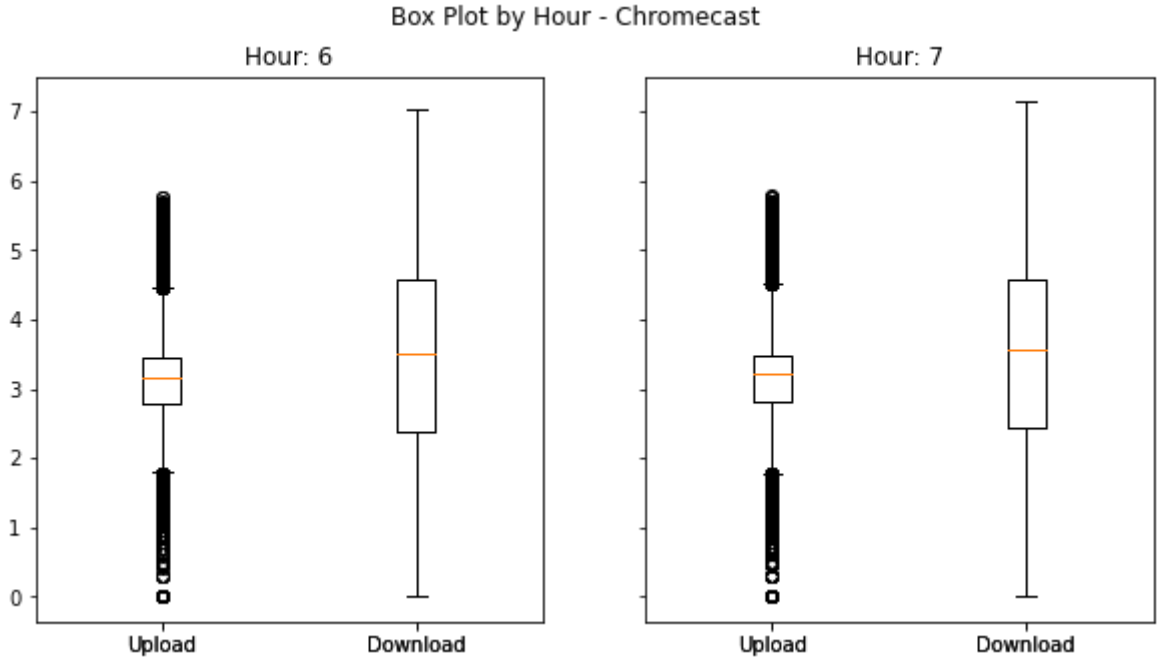
Os box plots foram montados, novamente, através da função “**boxplot**” do “**matplotlib**” e selecionando as horas, determinadas na coluna “**hour**”, de acordo com o loop criado para a contagem das mesmas, de zero a 23.

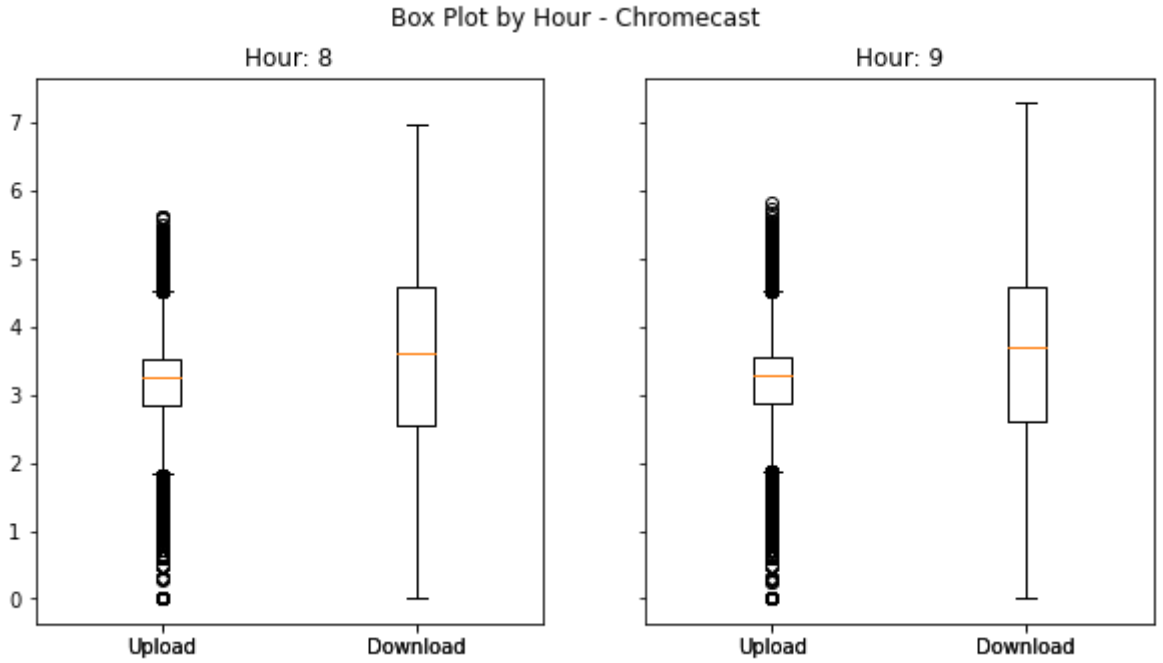
3.1.1 - Chromecast

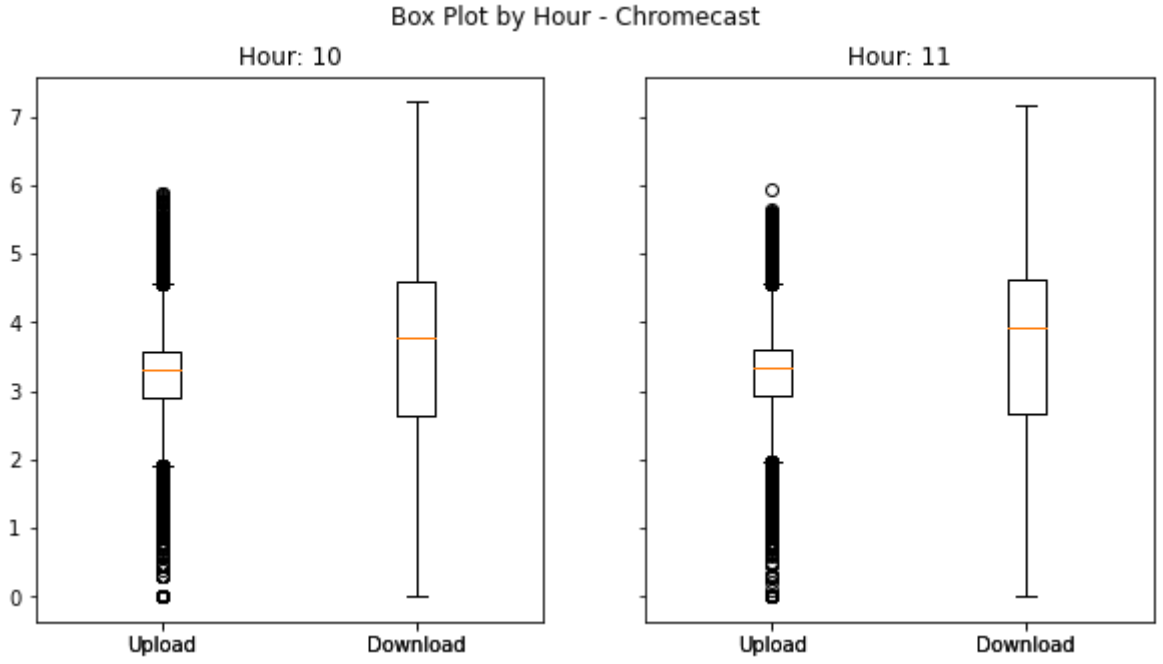


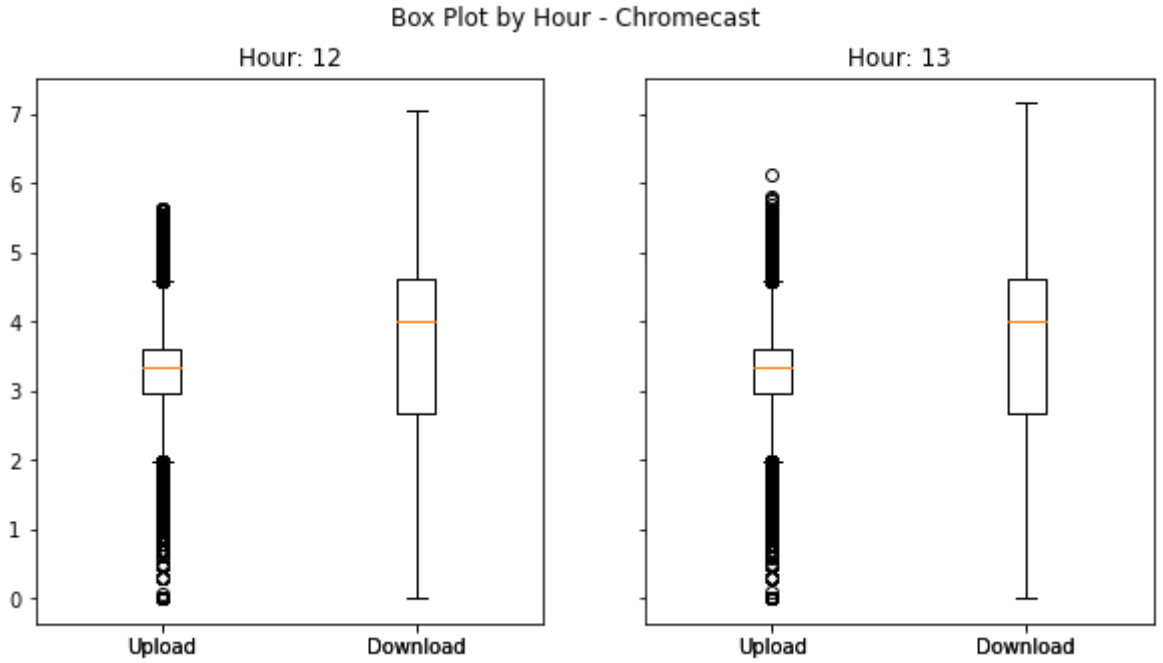

Box Plot by Hour - Chromecast
Hour: 2
Hour: 3
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

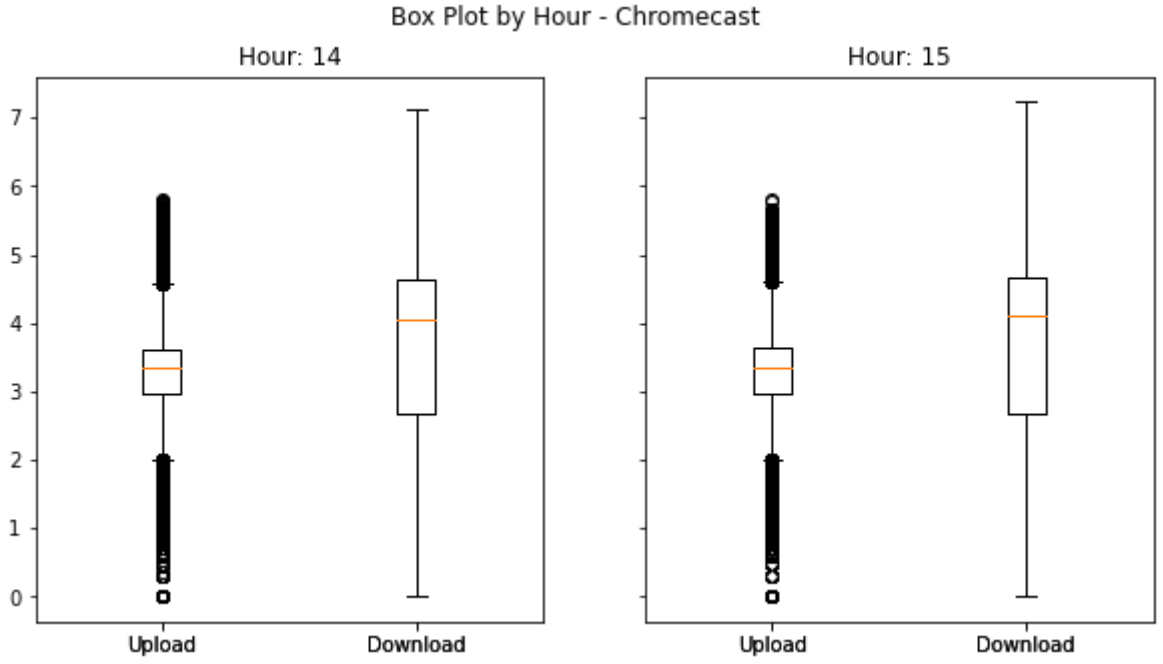

Box Plot by Hour - Chromecast
Hour: 4
Hour: 5
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

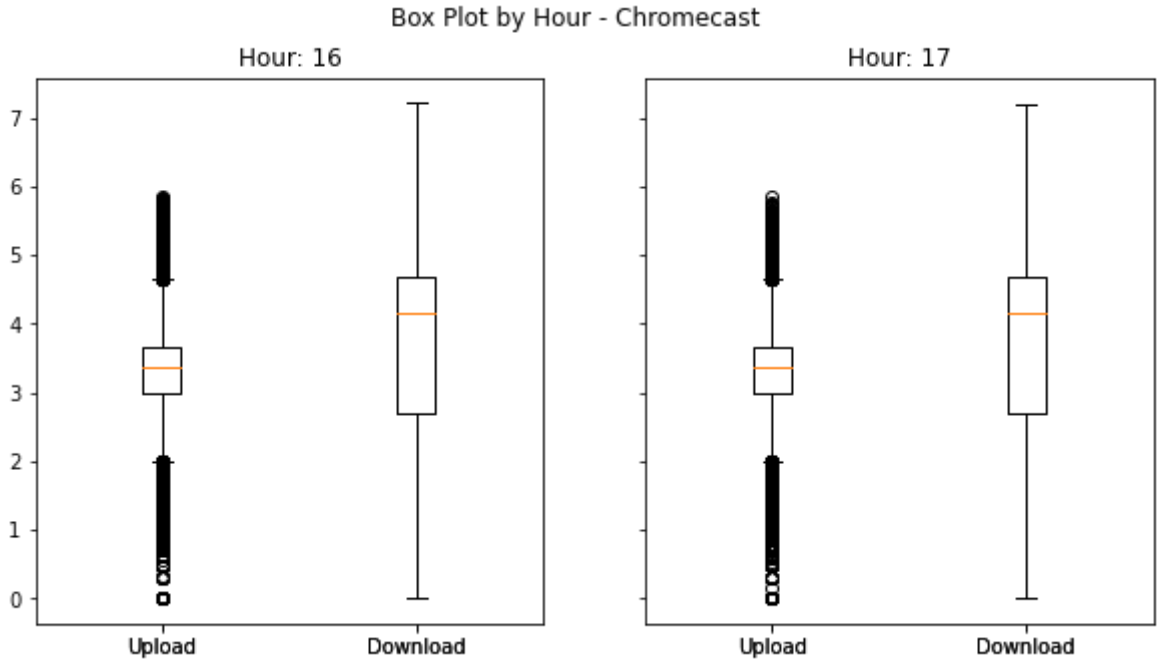

Box Plot by Hour - Chromecast
Hour: 6
Hour: 7
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

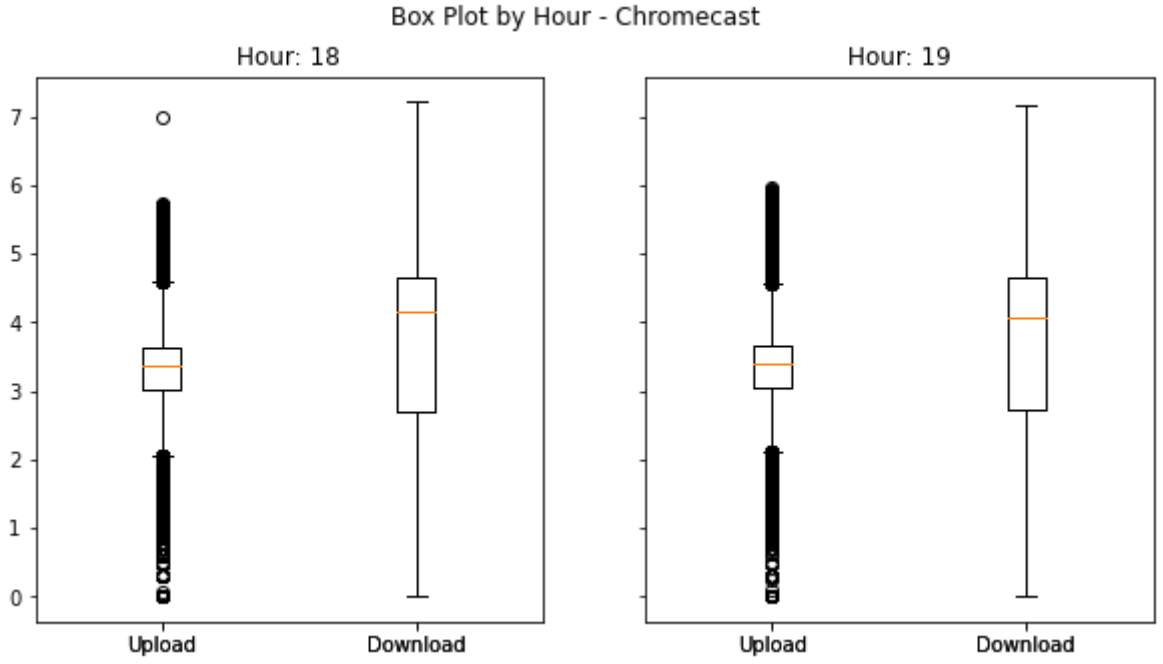

Box Plot by Hour - Chromecast
Hour: 8
Hour: 9
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

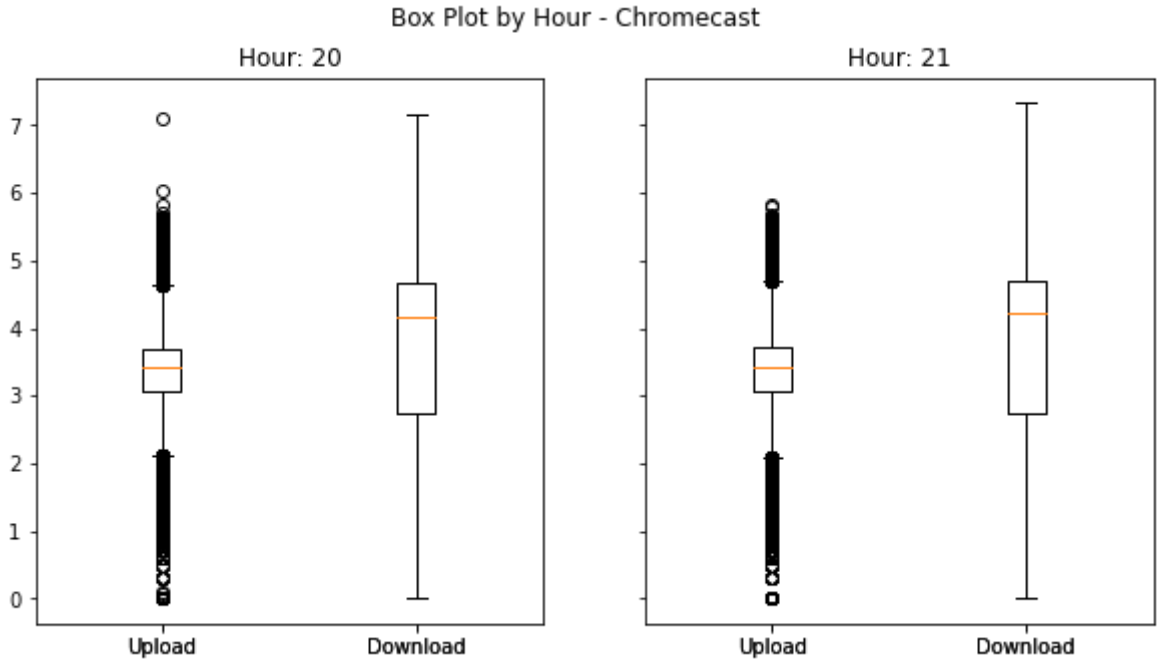

Box Plot by Hour - Chromecast
Hour: 10
Hour: 11
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

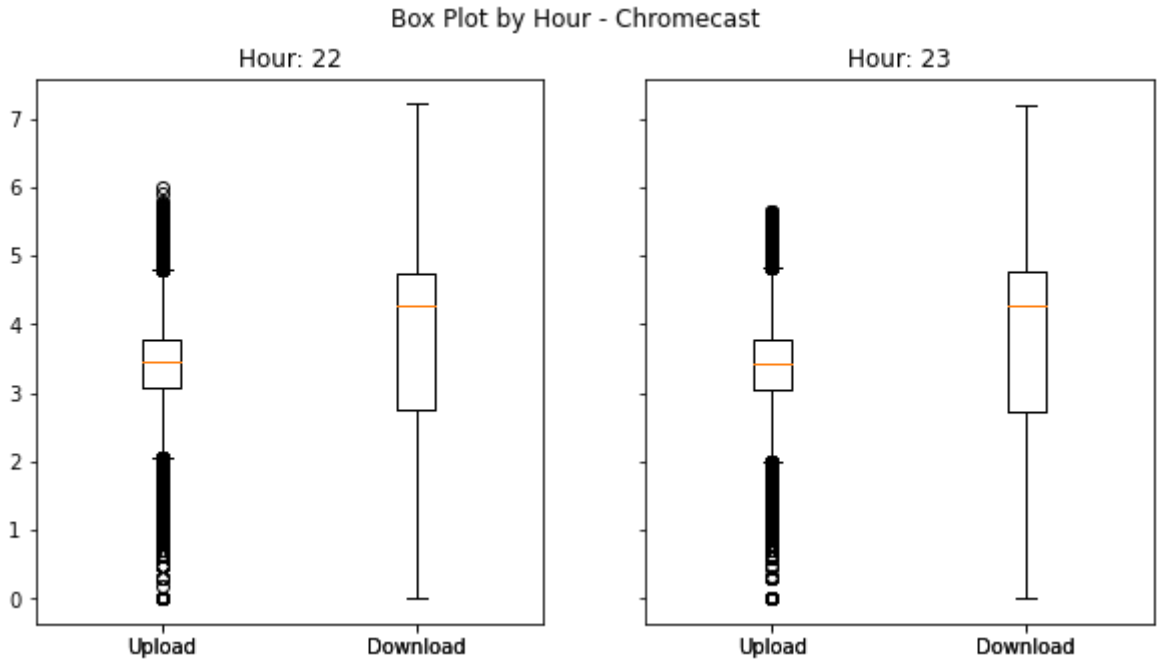

Box Plot by Hour - Chromecast
Hour: 12
Hour: 13
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download


Box Plot by Hour - Chromecast
Hour: 14
Hour: 15
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

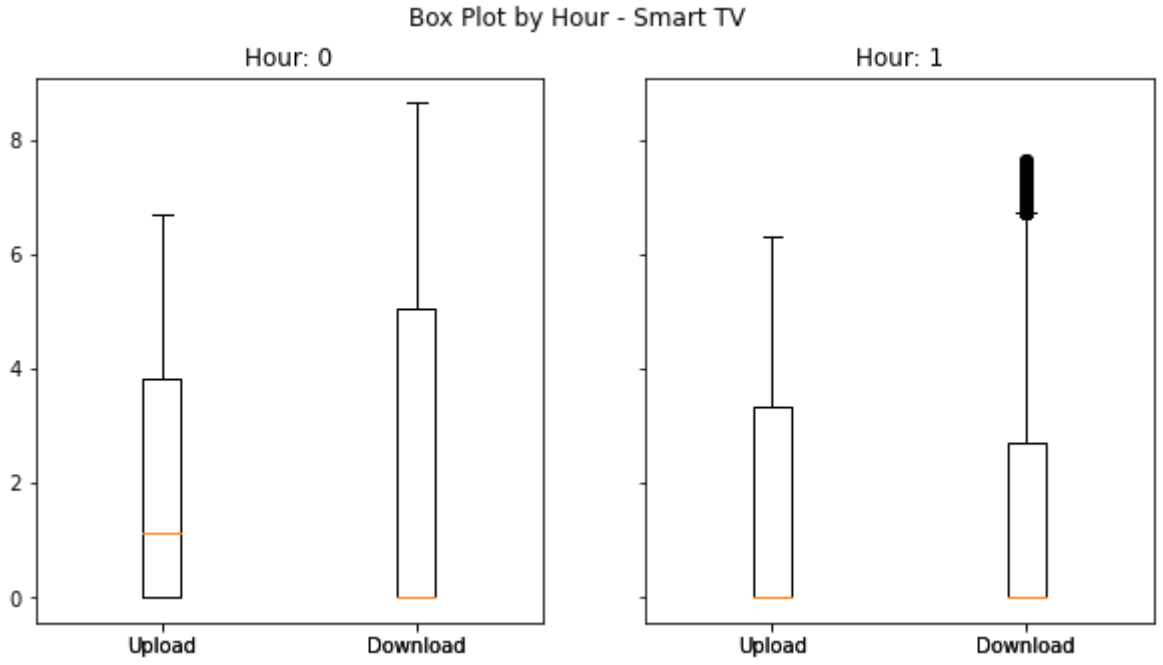

Box Plot by Hour - Chromecast
Hour: 16
Hour: 17
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

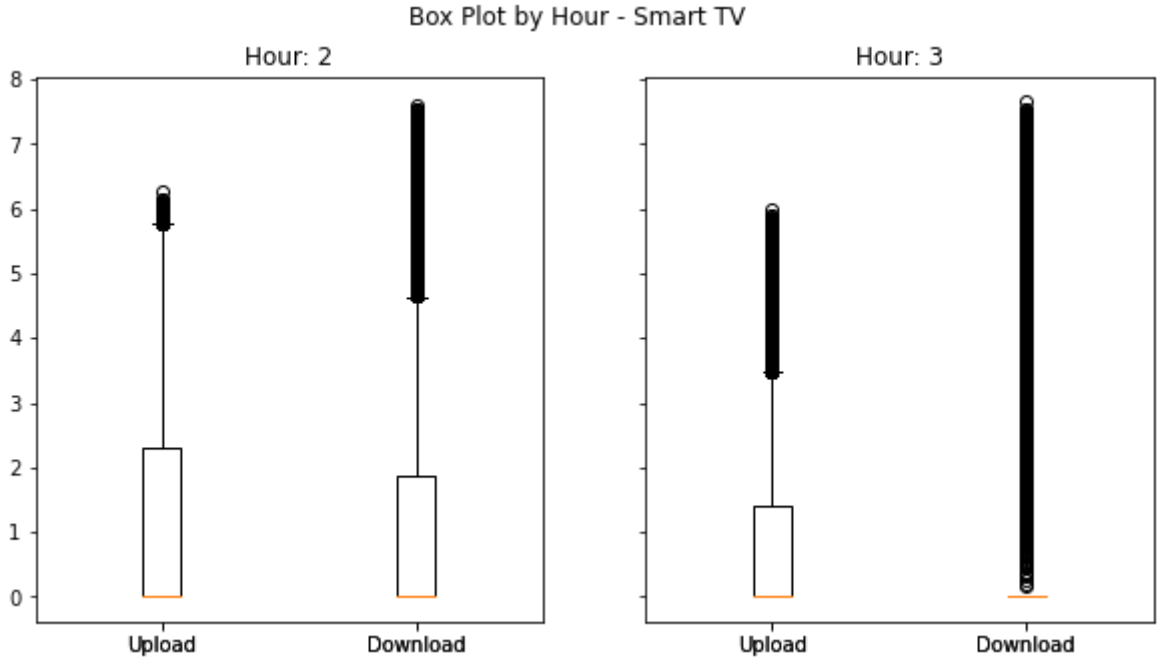

Box Plot by Hour - Chromecast
Hour: 18
Hour: 19
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

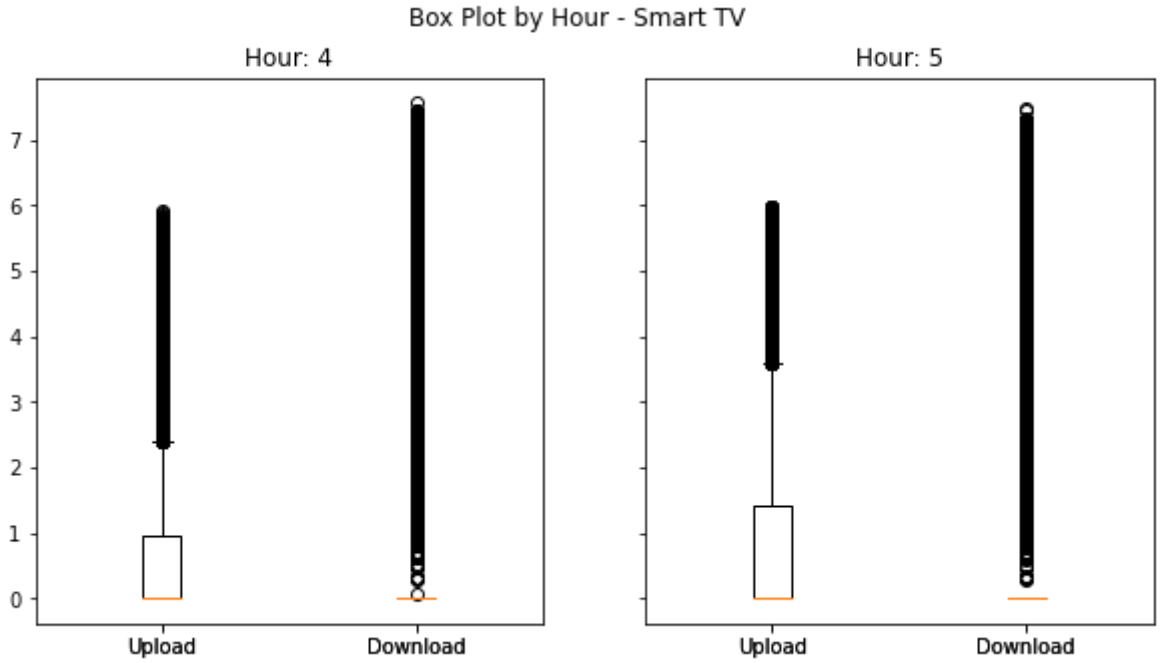

Box Plot by Hour - Chromecast
Hour: 20
Hour: 21
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

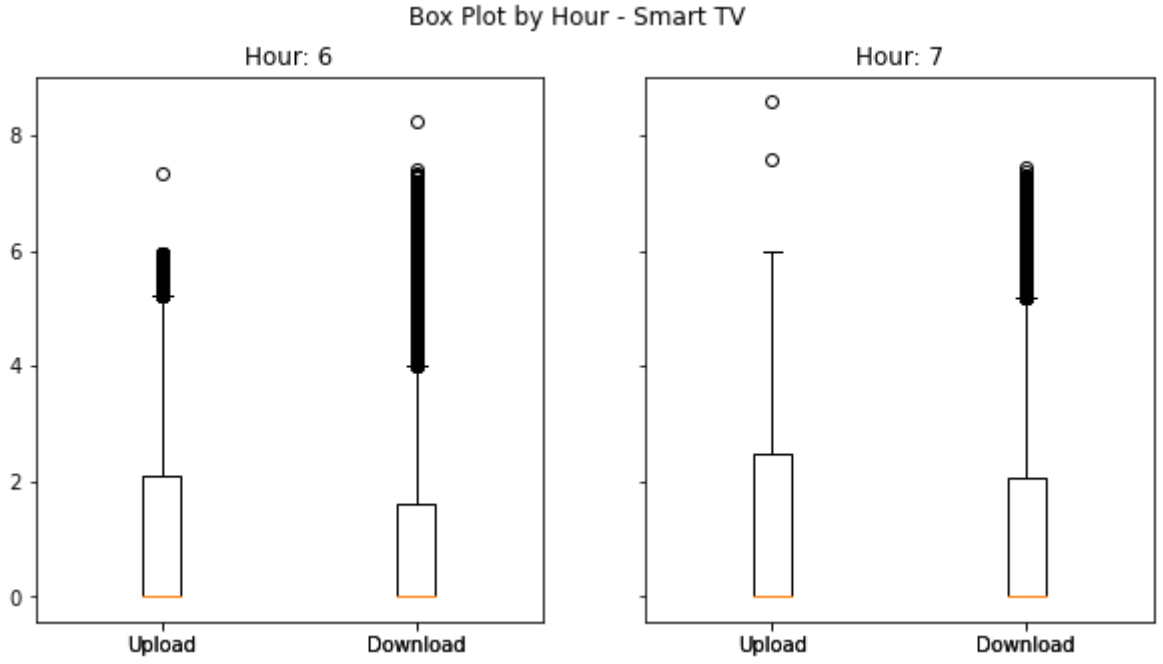

Box Plot by Hour - Chromecast
Hour: 22
Hour: 23
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

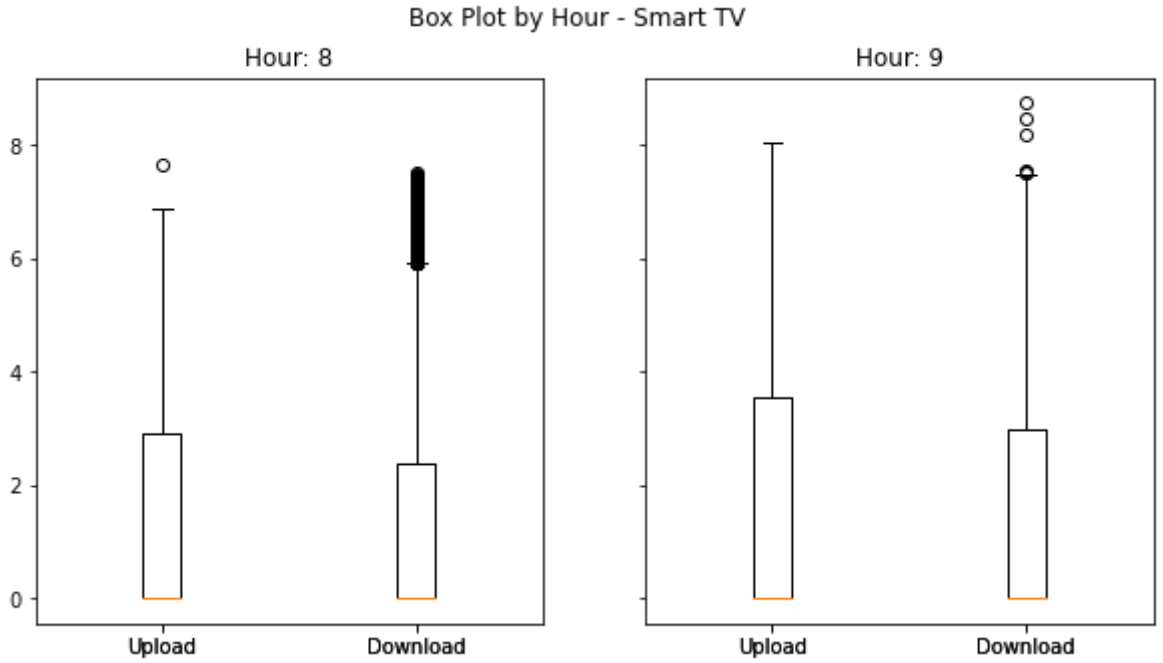
3.1.2 - Smart TV

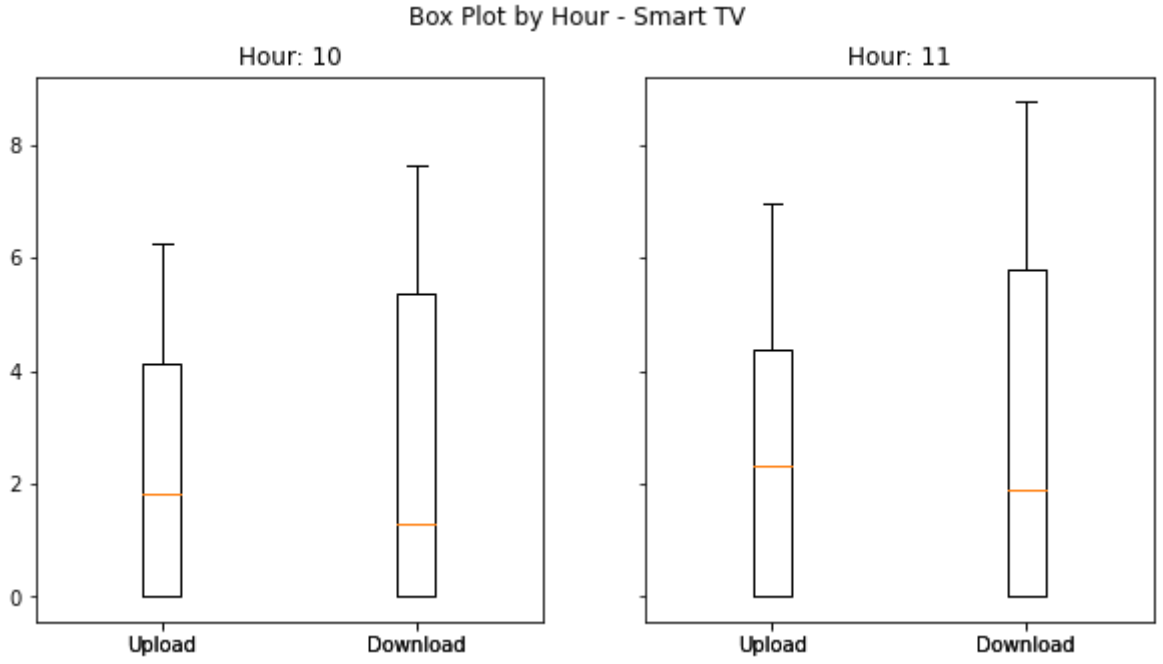

Box Plot by Hour - Smart TV
Hour: 0
Hour: 1
0
2
4
6
8
Upload
Download
Upload
Download

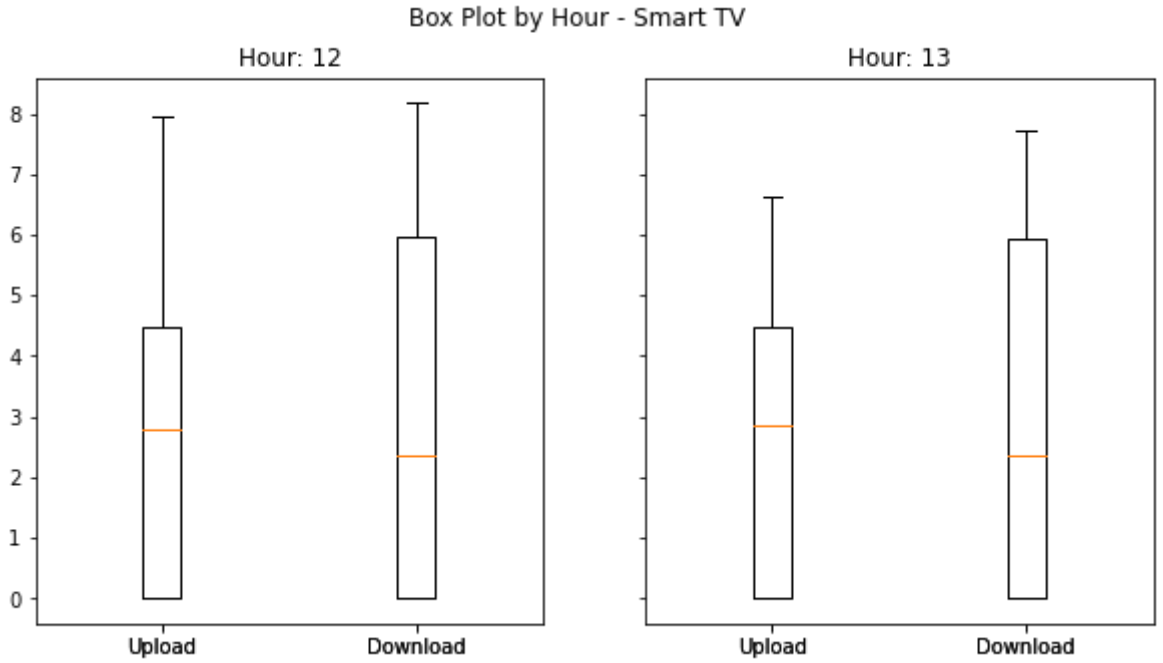

Box Plot by Hour - Smart TV
Hour: 2
Hour: 3
0
1
2
3
4
5
6
7
8
Upload
Download
Upload
Download

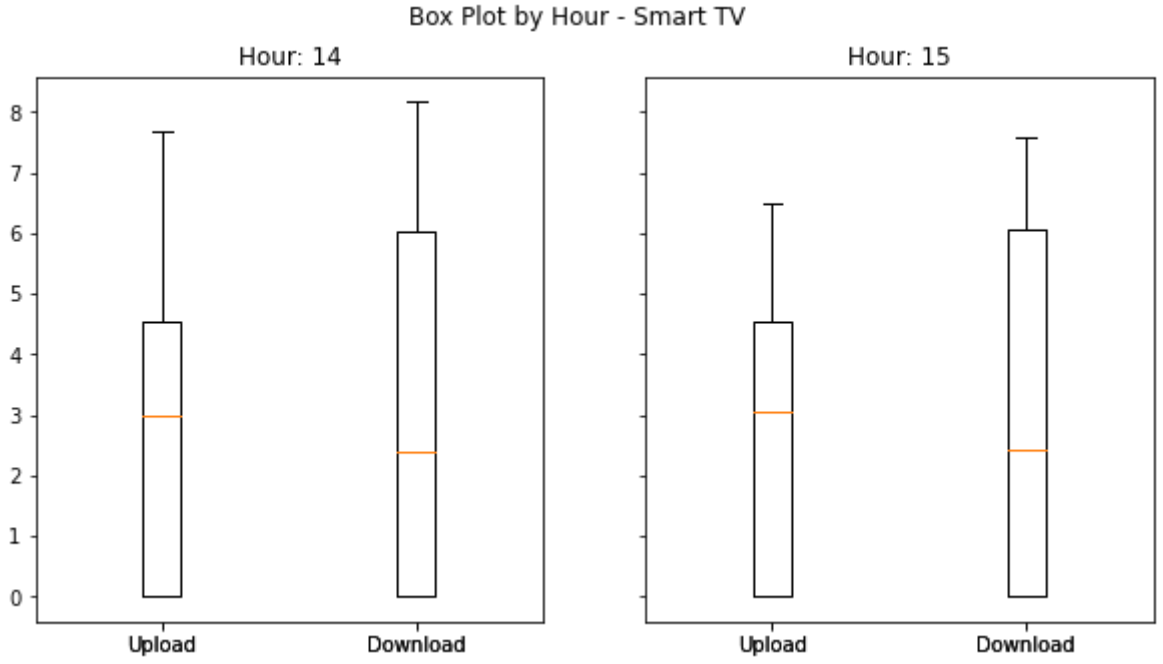

Box Plot by Hour - Smart TV
Hour: 4
Hour: 5
0
1
2
3
4
5
6
7
Upload
Download
Upload
Download

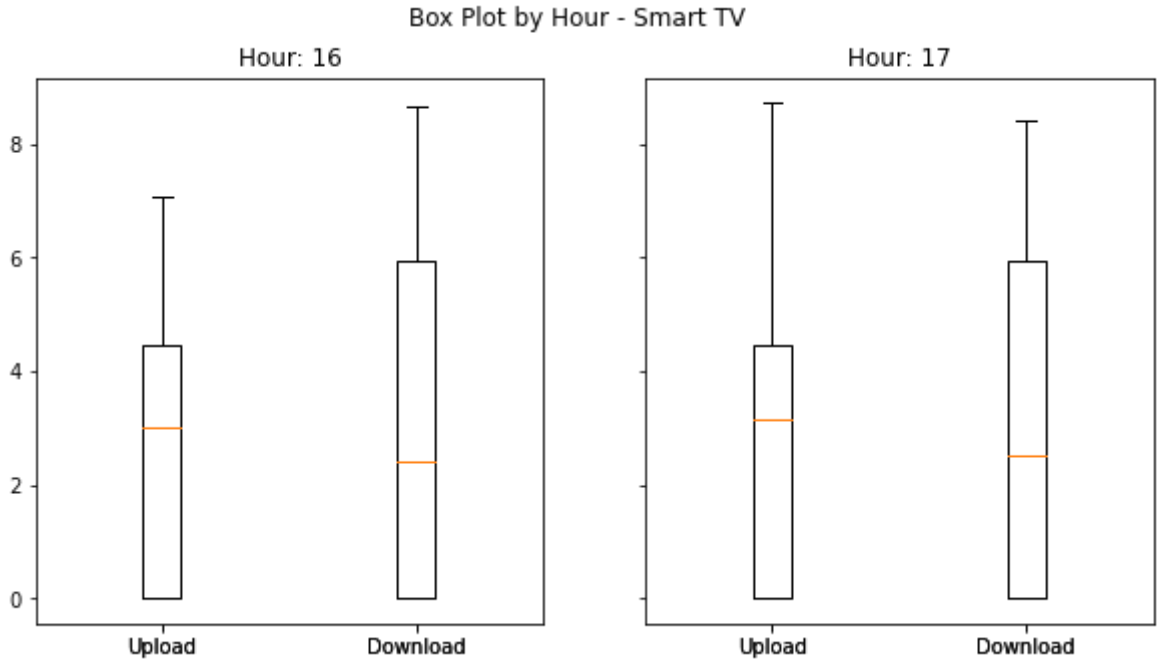

Box Plot by Hour - Smart TV
Hour: 6
Hour: 7
8
6
4
2
0
Upload
Download
Upload
Download

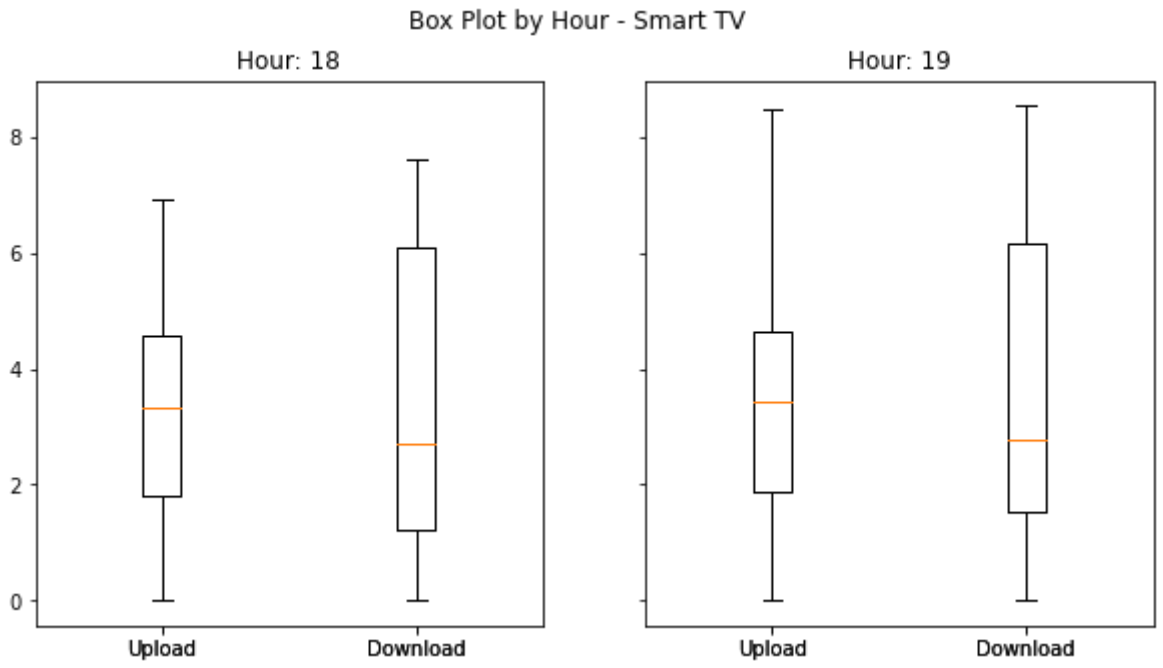

Box Plot by Hour - Smart TV
Hour: 8
Hour: 9
8
6
4
2
0
Upload
Download
Upload
Download

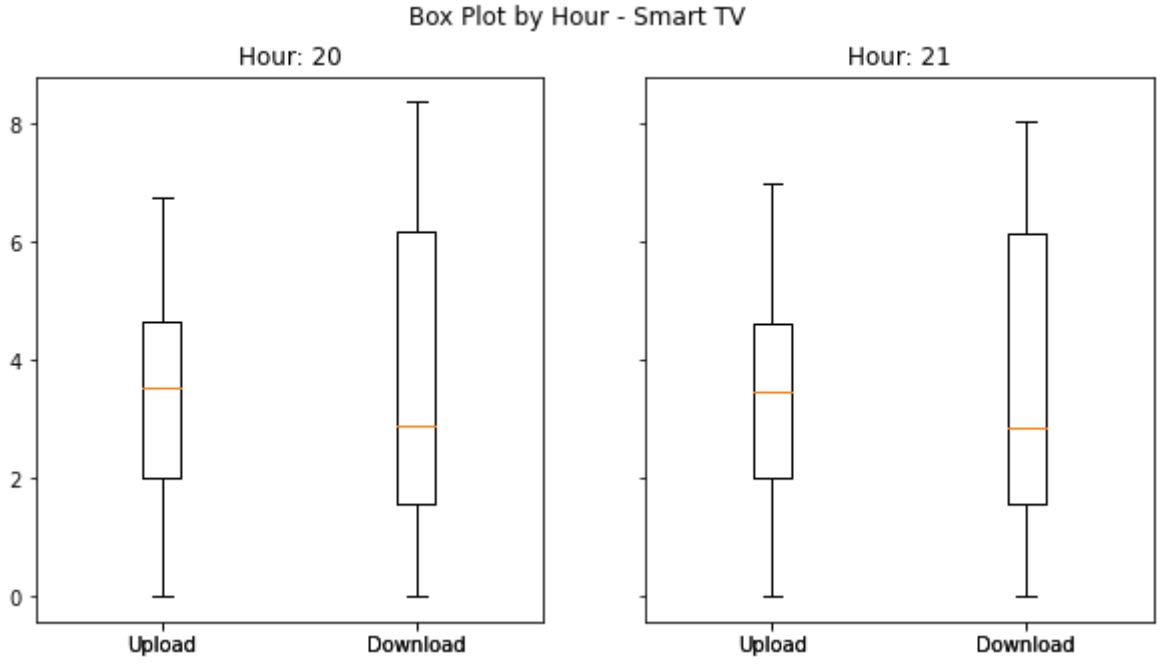

Box Plot by Hour - Smart TV
Hour: 10
Hour: 11
0
2
4
6
8
Upload
Download
Upload
Download

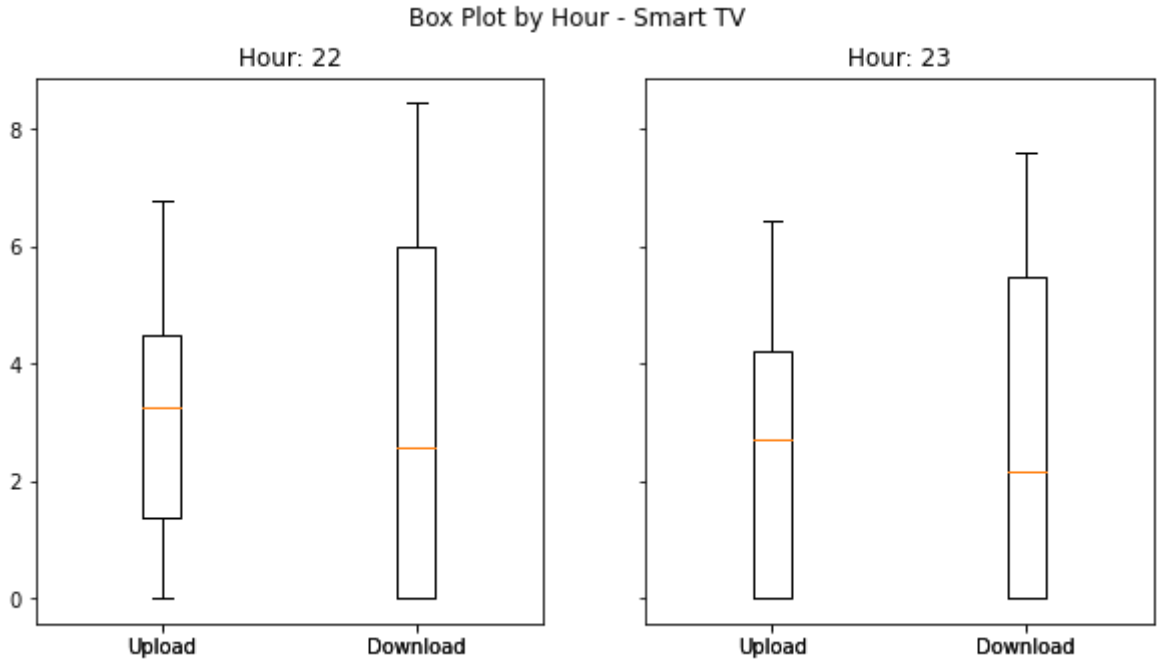

Box Plot by Hour - Smart TV
Hour: 12
Hour: 13
0
1
2
3
4
5
6
7
8
Upload
Download
Upload
Download


Box Plot by Hour - Smart TV
Hour: 14
Hour: 15
0
1
2
3
4
5
6
7
8
Upload
Download
Upload
Download


Box Plot by Hour - Smart TV
Hour: 16
Hour: 17
0
2
4
6
8
Upload
Download
Upload
Download


Box Plot by Hour - Smart TV
Hour: 18
Hour: 19
8
6
4
2
0
Upload
Download
Upload
Download

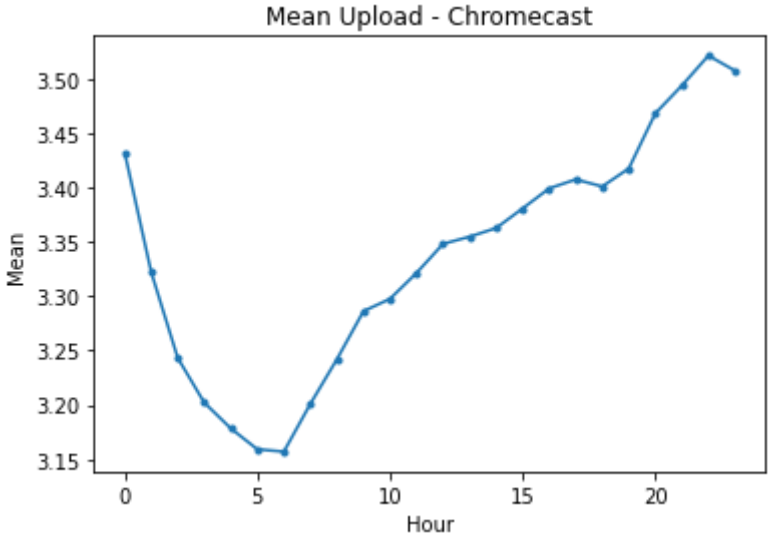

Box Plot by Hour - Smart TV
Hour: 20
Hour: 21
0
2
4
6
8
Upload
Download
Upload
Download

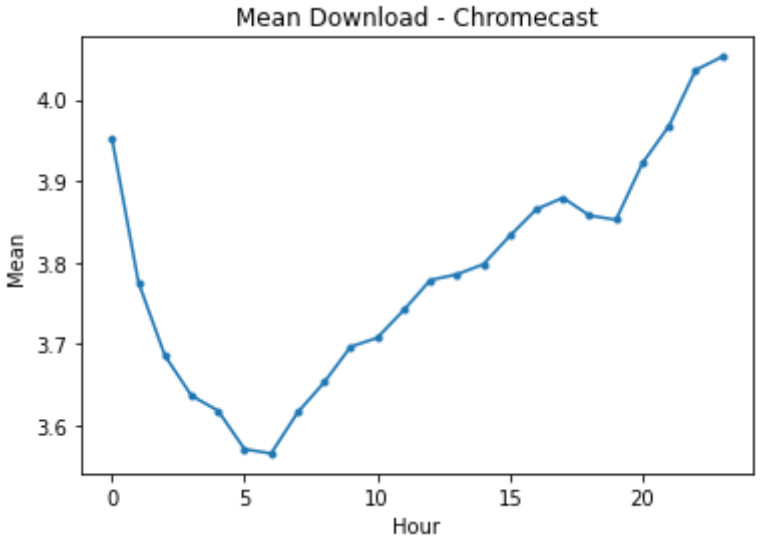

Box Plot by Hour - Smart TV
Hour: 22
Hour: 23
0
2
4
6
8
Upload
Download
Upload
Download

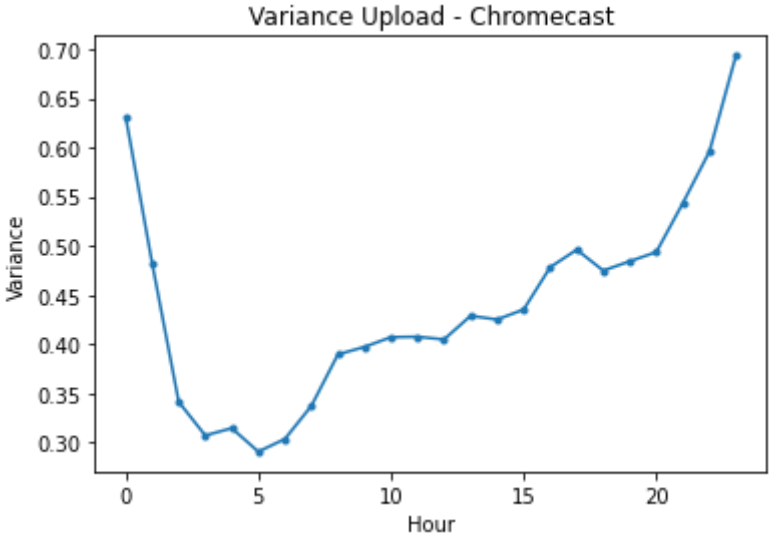
3.2 - Média, Variância e Desvio Padrão

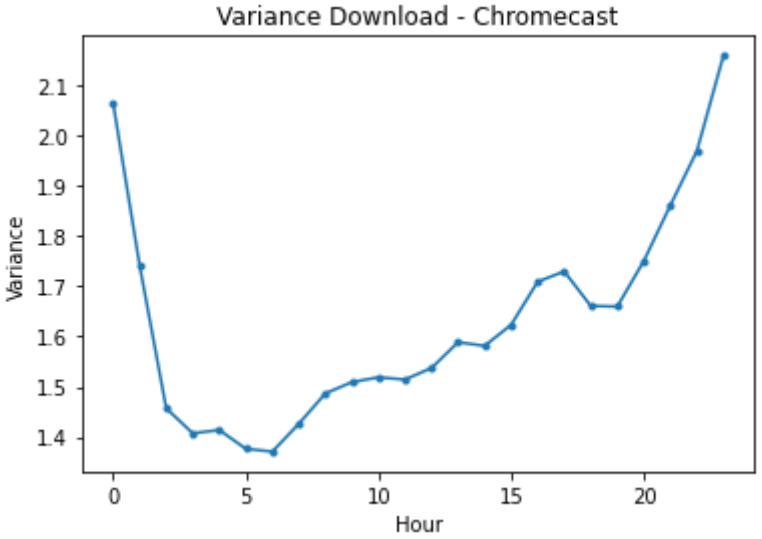
Para o cálculo desses valores de acordo com a hora, assim como foi pedido no roteiro, foi utilizada a função “**groupby**” do “**pandas**”, utilizando-se a nova coluna de hora como parâmetro.

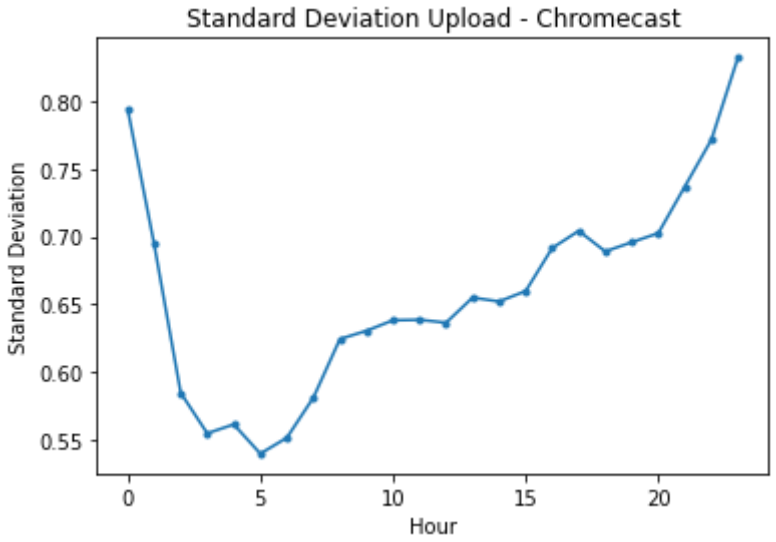
3.2.1 - Chromecast

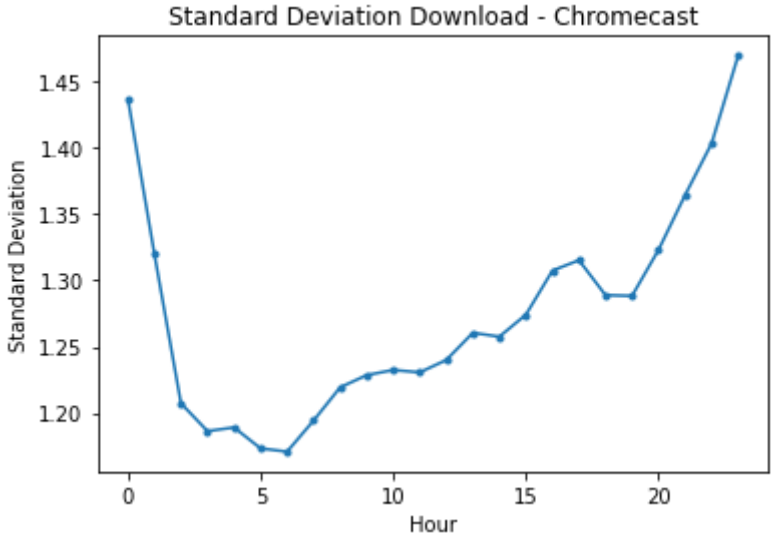




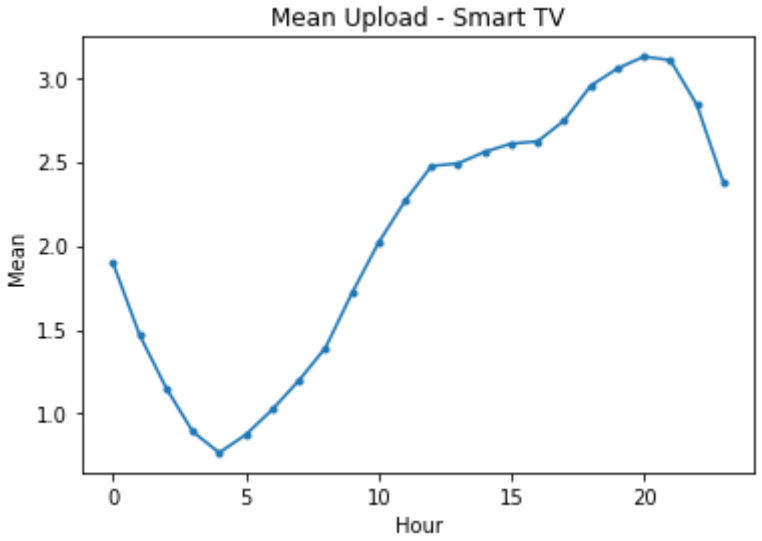

Variance Upload - Chromecast
Variance
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0
5
10
15
20
Hour

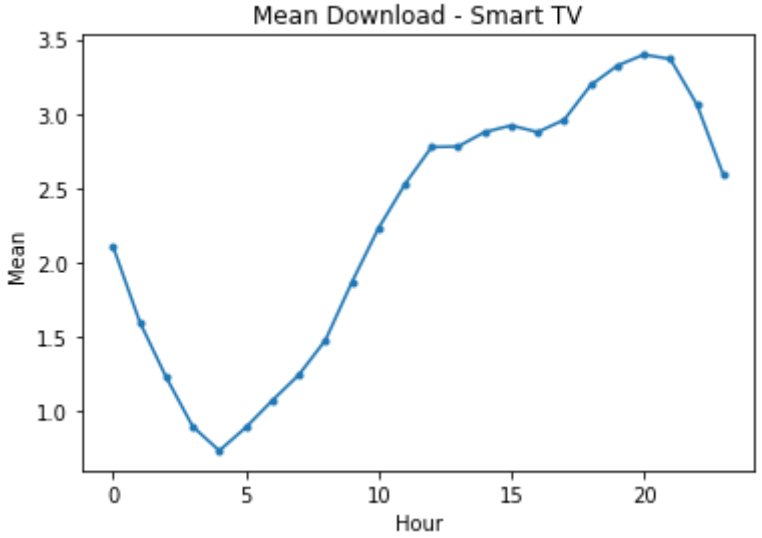

Variance Download - Chromecast
Variance
2.1
2.0
1.9
1.8
1.7
1.6
1.5
1.4
0
5
10
15
20
Hour

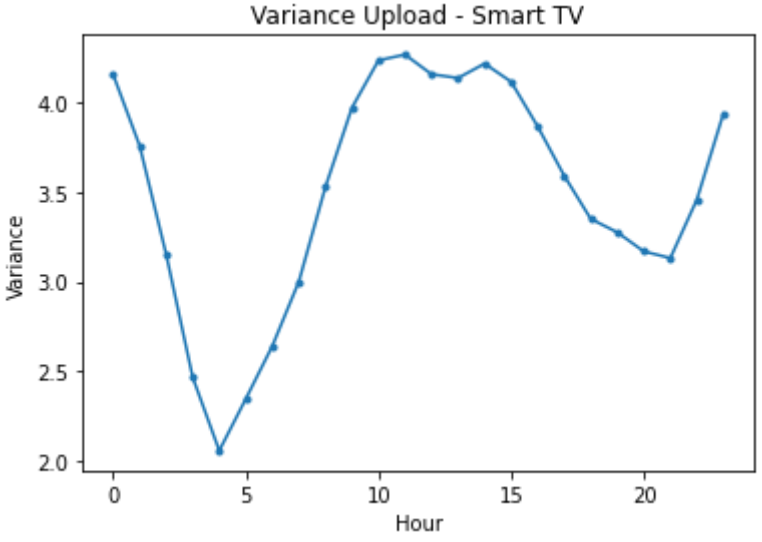

Standard Deviation Upload - Chromecast
Standard Deviation
0.80
0.75
0.70
0.65
0.60
0.55
0
5
10
15
20
Hour

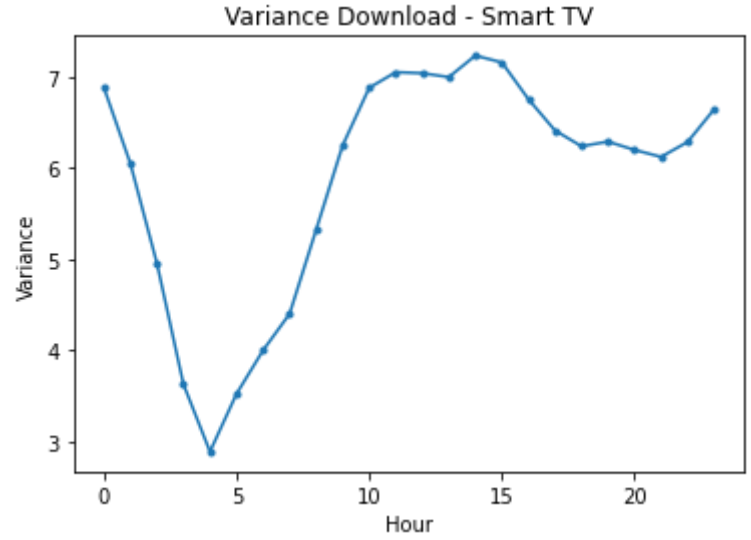

Standard Deviation Download - Chromecast
Standard Deviation
1.45
1.40
1.35
1.30
1.25
1.20
0
5
10
15
20
Hour

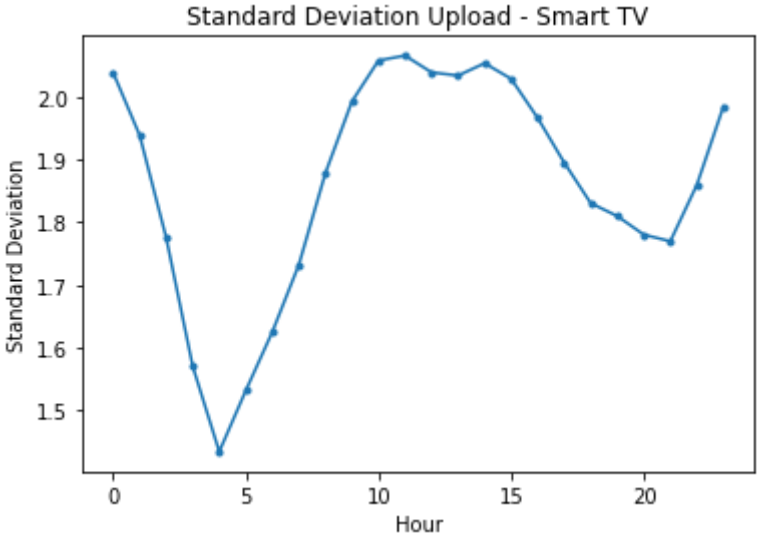
3.2.2 - Smart TV

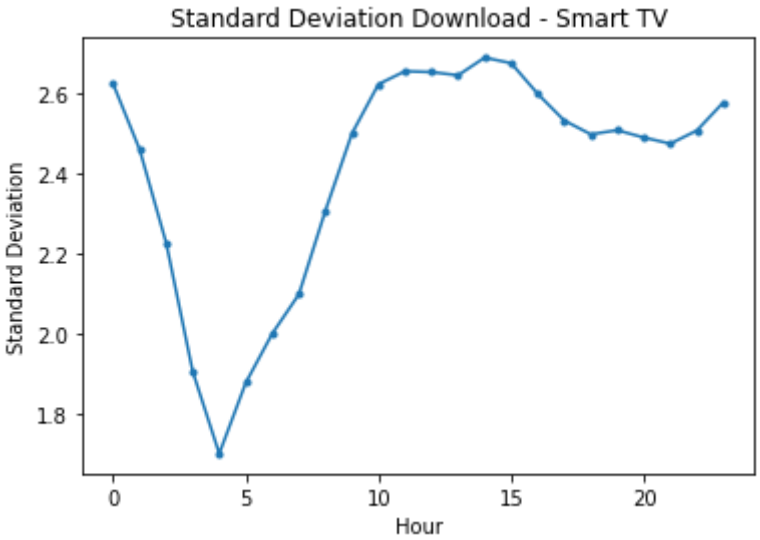

Mean Upload - Smart TV
Mean
Hour
3.0
2.5
2.0
1.5
1.0
0
5
10
15
20


Mean Download - Smart TV
Mean
Hour
3.5
3.0
2.5
2.0
1.5
1.0
0
5
10
15
20


Variance Upload - Smart TV
Variance
Hour
4.0
3.5
3.0
2.5
2.0
0
5
10
15
20


Variance Download - Smart TV
Variance
Hour
3
4
5
6
7
0
5
10
15
20


Standard Deviation Upload - Smart TV
Standard Deviation
Hour
1.5
1.6
1.7
1.8
1.9
2.0
0
5
10
15
20


Standard Deviation Download - Smart TV
Standard Deviation
Hour
1.8
2.0
2.2
2.4
2.6
0
5
10
15
20

3.3 - Análise dos Dados

Algo que podemos verificar logo de imediato através dos box plots e os gráficos de média é a grande quantidade de tráfego existente no período do final da tarde até a madrugada, até cerca de 3 horas da manhã. Isso condiz muito com a realidade da nossa sociedade, tendo em vista que estudamos e trabalhamos durante o dia e, no final da noite, aproveitamos para desfrutar de meios de entretenimento como um filme ou série através do chromecast e da smart tv, por exemplo, que estão sendo utilizadas como fonte de estudo.
Outro ponto muito interessante foi a grande diferença nos gráficos de Variância e Desvio Padrão quando comparamos o Chromecast e a Smart TV. Nos gráficos da Smart TV conseguimos ver grandes valores para a variância e desvio padrão, seja para upload quanto download, entretanto, no Chromecast estes valores são muito mais baixos, exceto pelo horário das 20 horas até a meia-noite.
Esta diferença provavelmente se dá devido aos valores nulos existentes no dataset da Smart TV, indicando que tais horas (por volta de meio-dia e meia-noite) é o maior horário em que não há consumo ou envio de dados pela Smart TV.

4 - Caracterizando os horários com maior valor de tráfego

4.1 - Seleção dos horários

Para esta seleção foram utilizados os maiores valores de média e mediana de cada hora dos dados no dataset. O cálculo foi realizado utilizando as funções “**groupby**”, “**median**”, “**mean**” e “**idxmax**” disponíveis no “**pandas**”, com o objetivo de agrupar os dados pela hora em que foram coletados e determinar sua média e mediana para, então, determinar o maior valor.

```
Hora Dataset 1: 20
Hora Dataset 2: 20
Hora Dataset 3: 20
Hora Dataset 4: 20
Hora Dataset 5: 22
Hora Dataset 6: 22
Hora Dataset 7: 23
Hora Dataset 8: 23
```

4.2 - MLE

4.2.1 - Gamma

Os valores de shape, offset e scale foram calculados utilizando a função “**gamma.fit**” da biblioteca “**stats**”.

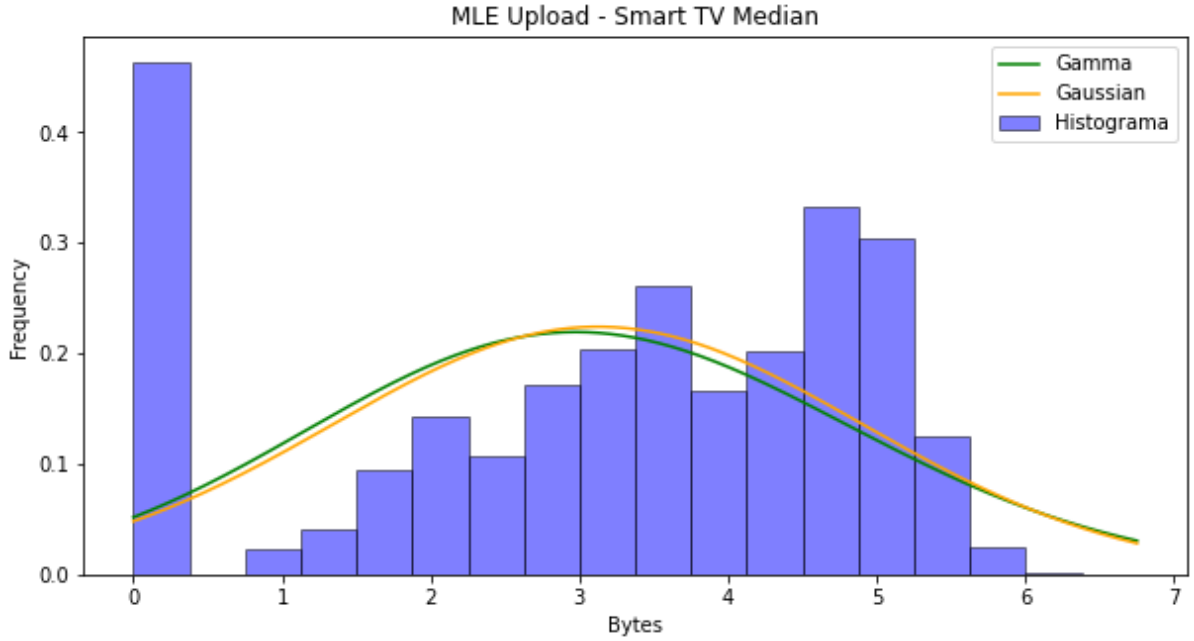
```
df1: Shape = 220.48073768362616 | Loc = -23.96174486447611 | Scale = 0.12272309565740211
df2: Shape = 220.48073768362616 | Loc = -23.96174486447611 | Scale = 0.12272309565740211
df3: Shape = 896.5469322463027 | Loc = -71.06216506397283 | Scale = 0.08304989773768084
df4: Shape = 896.5469322463027 | Loc = -71.06216506397283 | Scale = 0.08304989773768084
df5: Shape = 3148.881521123096 | Loc = -39.808982624042514 | Scale = 0.013760617086978861
df6: Shape = 3148.881521123096 | Loc = -39.808982624042514 | Scale = 0.013760617086978861
df7: Shape = 27.130143662659037 | Loc = -3.6313681850617665 | Scale = 0.2832298335465073
df8: Shape = 27.130143662659037 | Loc = -3.6313681850617665 | Scale = 0.2832298335465073
```

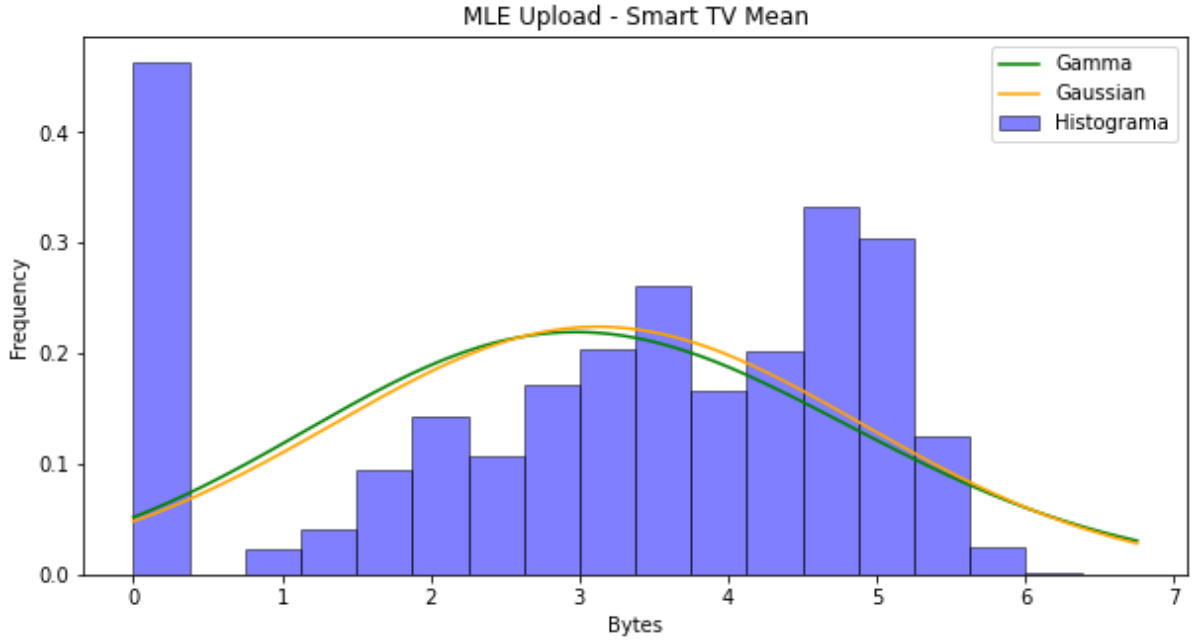
4.2.2 - Gaussiana

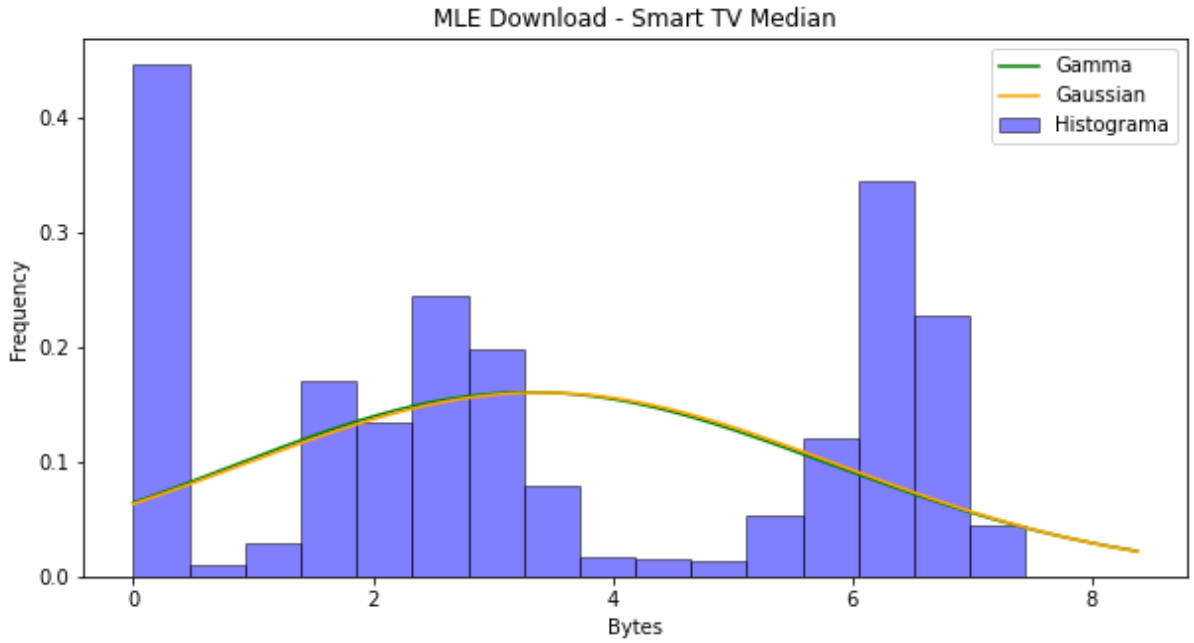
Os valores de média e desvio padrão para a Gaussiana foram calculados conforme explicado anteriormente.

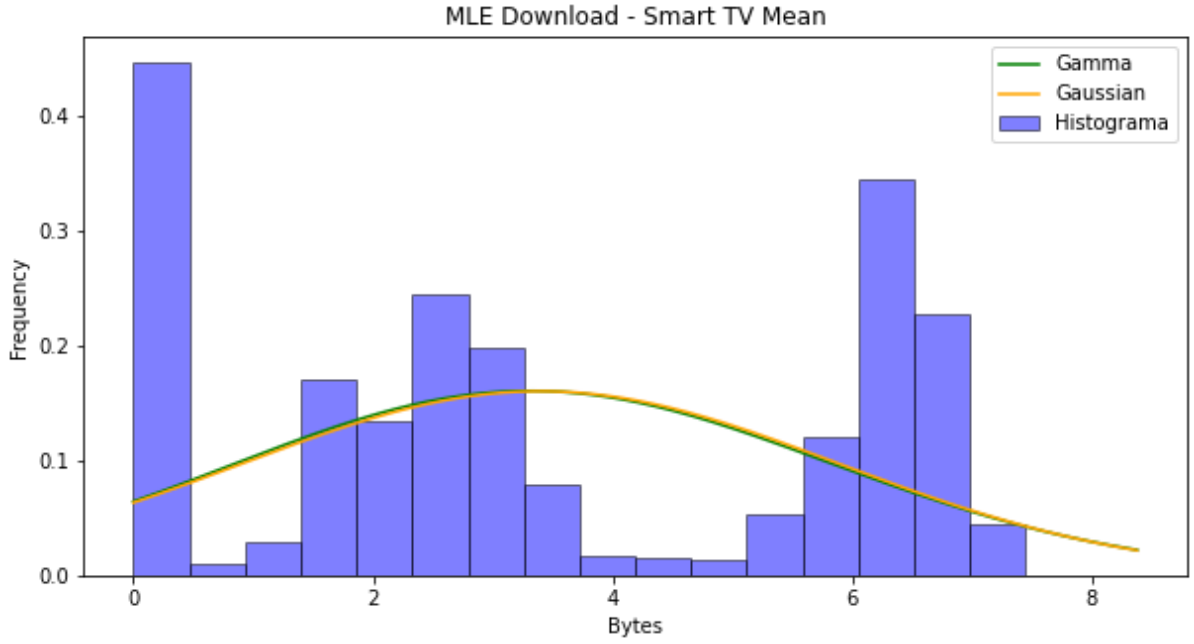
```
df1: Média = 3.124258107506722 | Desvio Padrão = 1.7800995973531673
df2: Média = 3.124258107506722 | Desvio Padrão = 1.7800995973531673
df3: Média = 3.3960945564366285 | Desvio Padrão = 2.4902555259728576
df4: Média = 3.3960945564366285 | Desvio Padrão = 2.4902555259728576
df5: Média = 3.521546370674634 | Desvio Padrão = 0.7718286854202558
df6: Média = 3.521546370674634 | Desvio Padrão = 0.7718286854202558
df7: Média = 4.052698112658847 | Desvio Padrão = 1.4694860227345465
df8: Média = 4.052698112658847 | Desvio Padrão = 1.4694860227345465
```

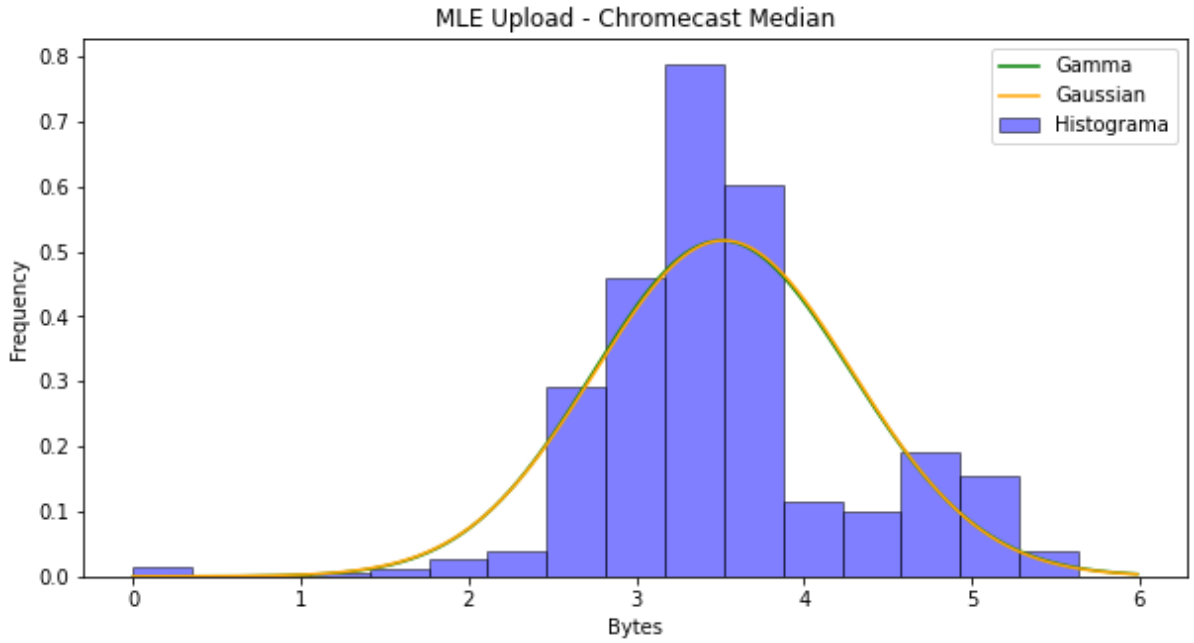
4.3 - Histograma e MLE

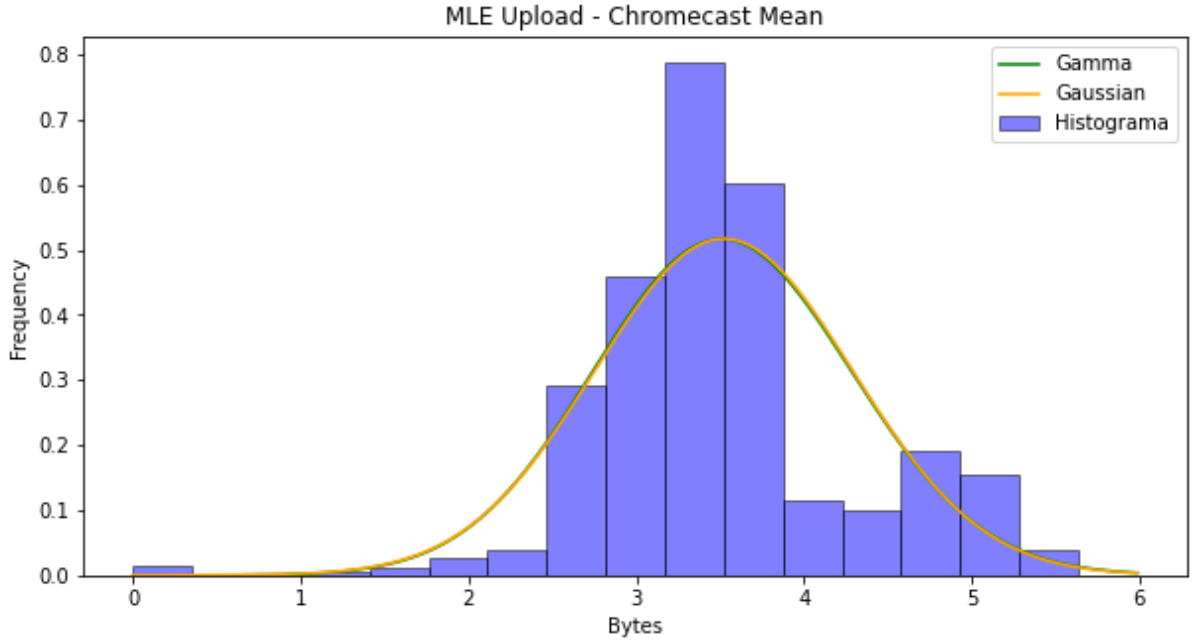


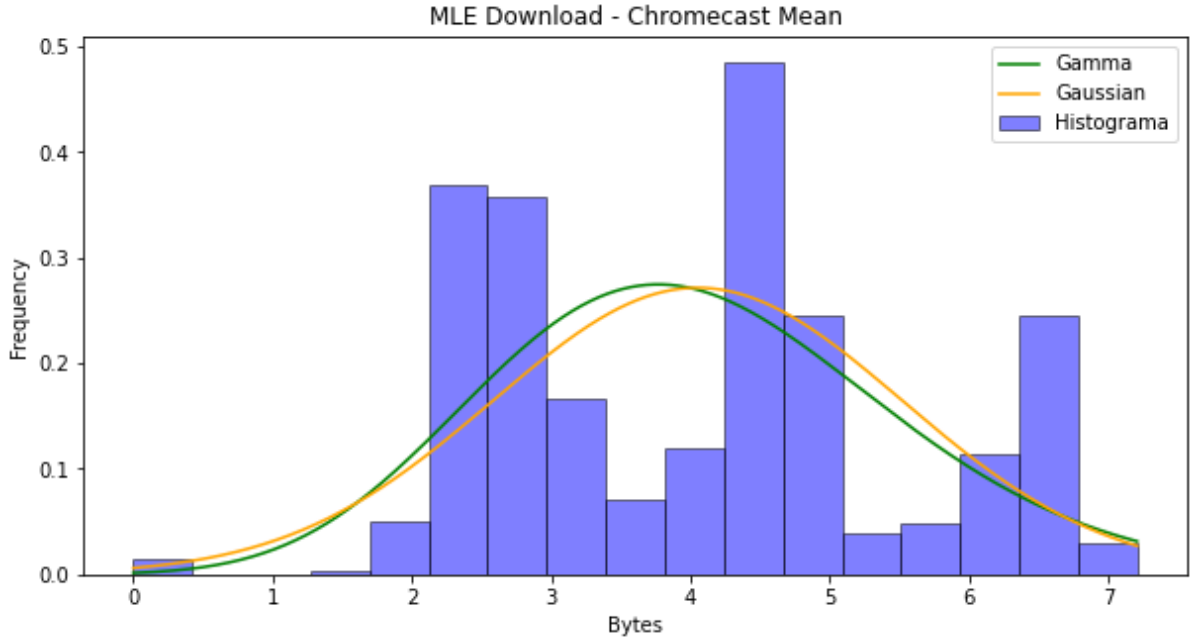


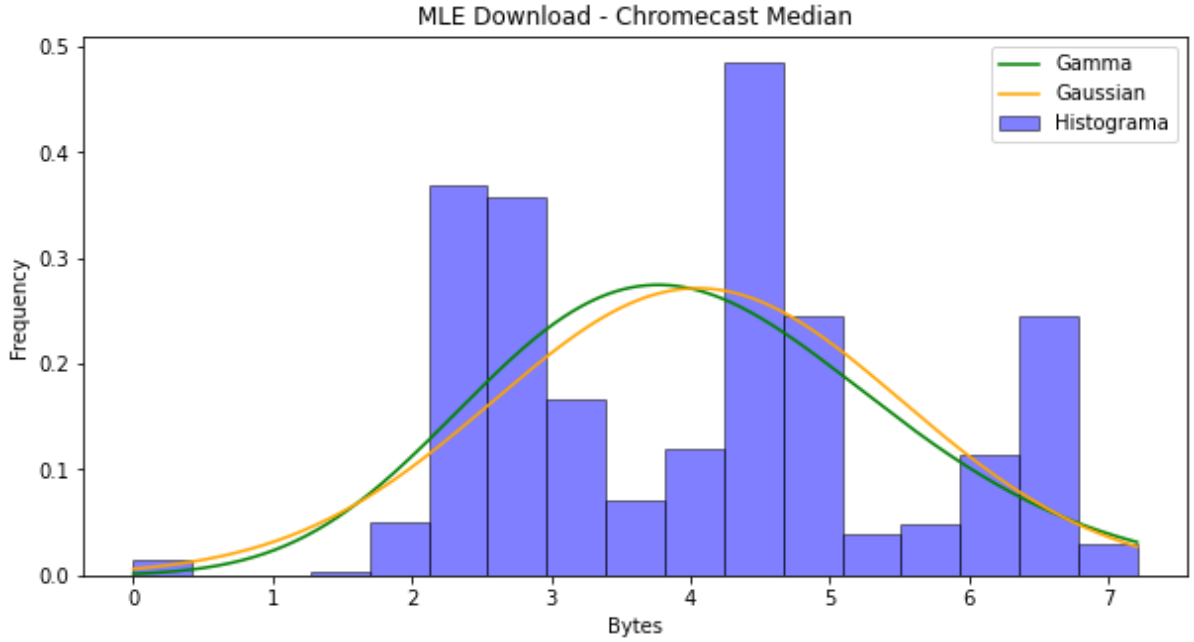

MLE Download - Smart TV Median
Gamma
Gaussian
Histograma
Frequency
Bytes
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8


MLE Download - Smart TV Mean
Gamma
Gaussian
Histograma
Frequency
Bytes
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8

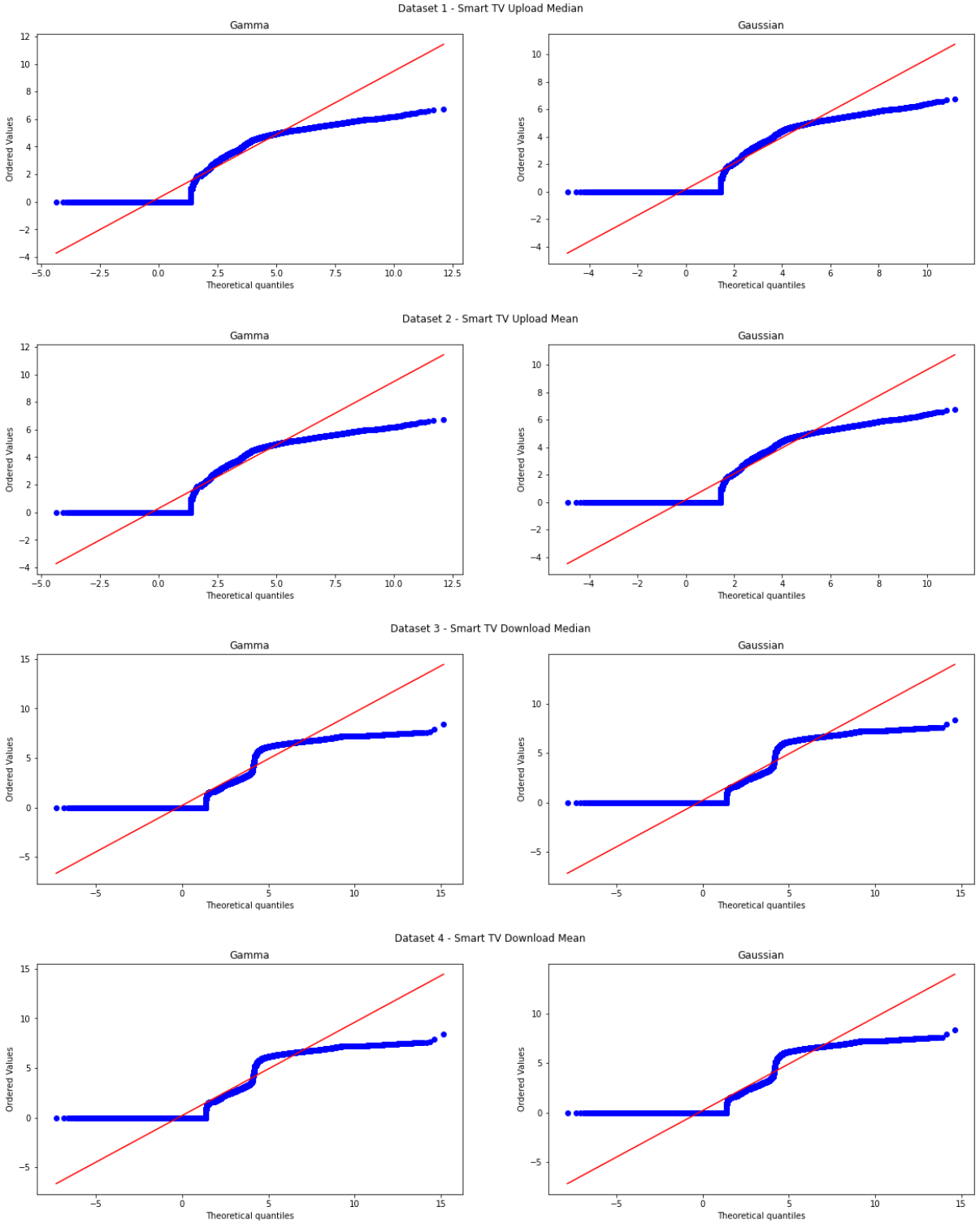

MLE Upload - Chromecast Median
Gamma
Gaussian
Histograma
Frequency
Bytes
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
1
2
3
4
5
6


MLE Upload - Chromecast Mean
Gamma
Gaussian
Histograma
Frequency
Bytes
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0
1
2
3
4
5
6

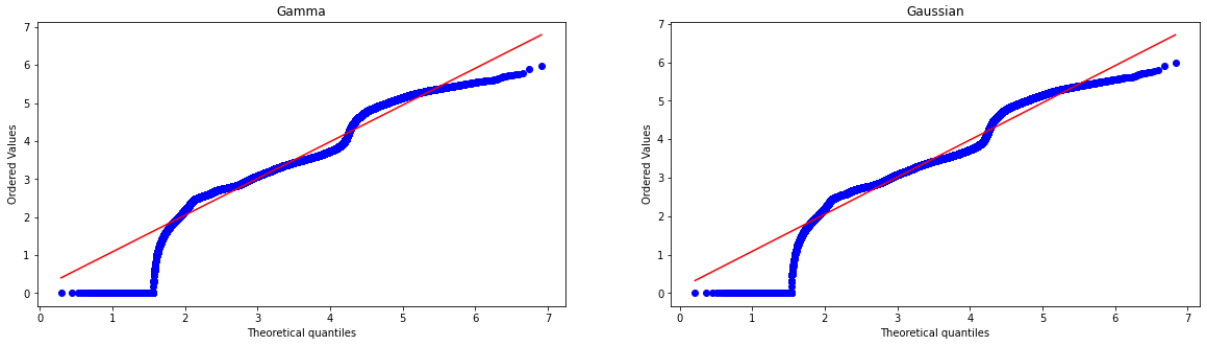

MLE Download - Chromecast Mean
Gamma
Gaussian
Histograma
Frequency
Bytes
0.0
0.1
0.2
0.3
0.4
0.5
0
1
2
3
4
5
6
7


MLE Download - Chromecast Median
Gamma
Gaussian
Histograma
Frequency
Bytes
0.0
0.1
0.2
0.3
0.4
0.5
0
1
2
3
4
5
6
7

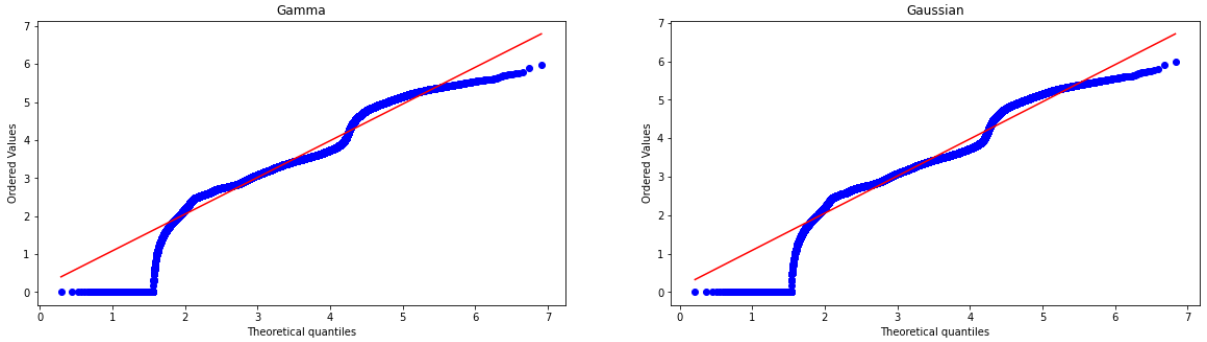
4.4 - Gráfico de Probabilidade



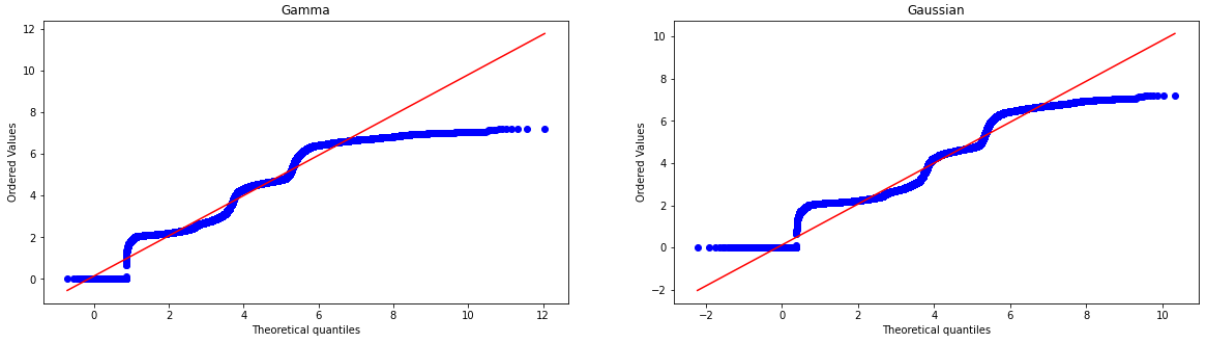
Dataset 5 - Chromecast Upload Median



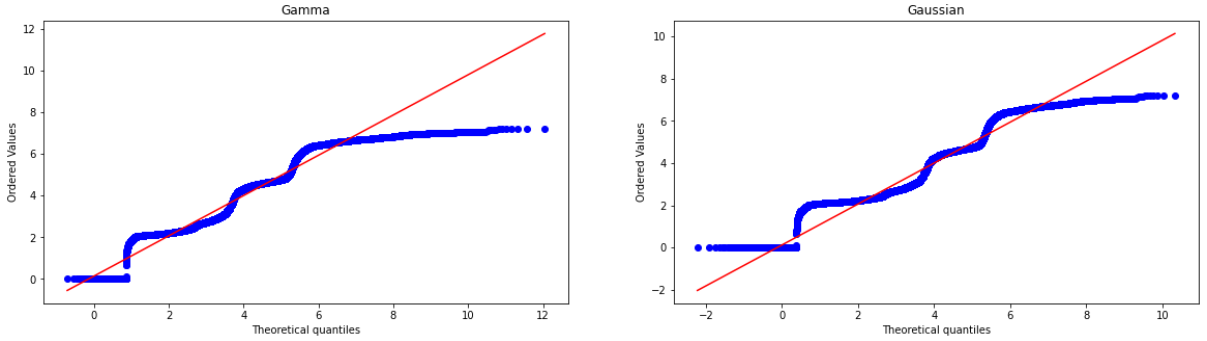
Dataset 6 - Chromecast Upload Mean



Dataset 7 - Chromecast Download Median



Dataset 8 - Chromecast Download Mean



4.5 - Análise dos Dados

Os horários escolhidos foram os seguintes:

```
Hora Dataset 1: 20
Hora Dataset 2: 20
Hora Dataset 3: 20
Hora Dataset 4: 20
Hora Dataset 5: 22
Hora Dataset 6: 22
Hora Dataset 7: 23
Hora Dataset 8: 23
```

Os resultados dos datasets 1 e 2, 3 e 4, 5 e 6 e 7 e 8 são iguais pois os datasets são os mesmos, já que vêm da mesma fonte e foram filtrados pelo mesmo horário.

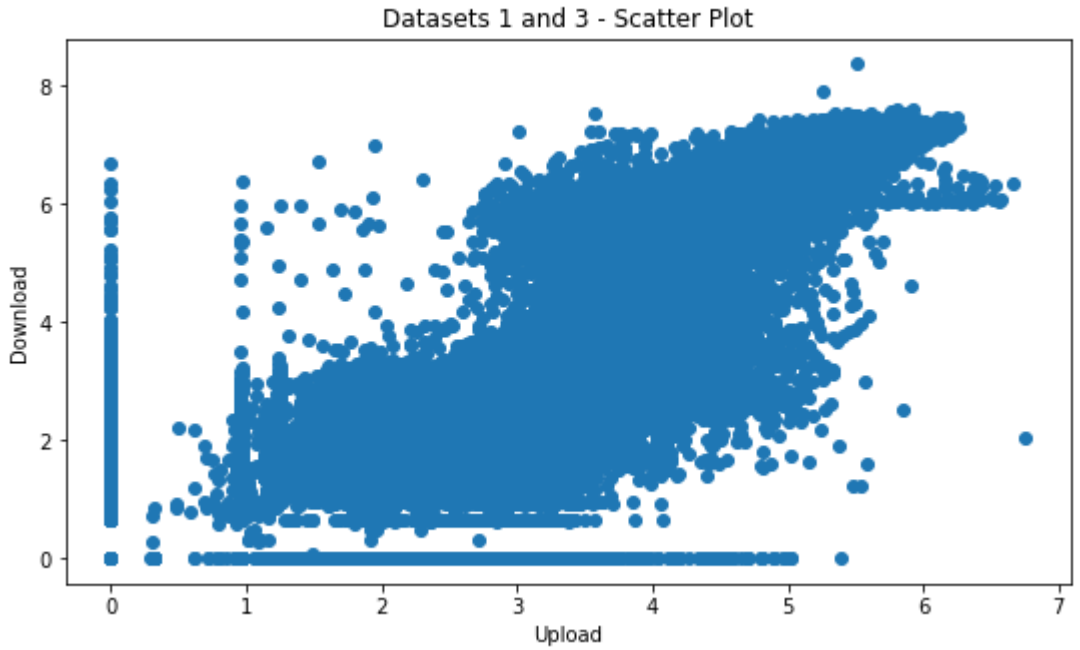
Mais um ponto interessante é a igualdade entre os gráficos de média e mediana para cada dataset, indicando que os dados utilizados nos cálculos são bem centralizados (não estão agrupados em direção em uma extremidade), além de valores de gamma e gaussiana muito próximos para a maioria dos casos.
Também conseguimos afirmar, tanto devido aos histogramas quanto os gráficos de probabilidade, que os datasets 5 e 6 parecem ter bons resultados com a gamma e a gaussiana, sendo possível mapeá-los em variáveis aleatórias.

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

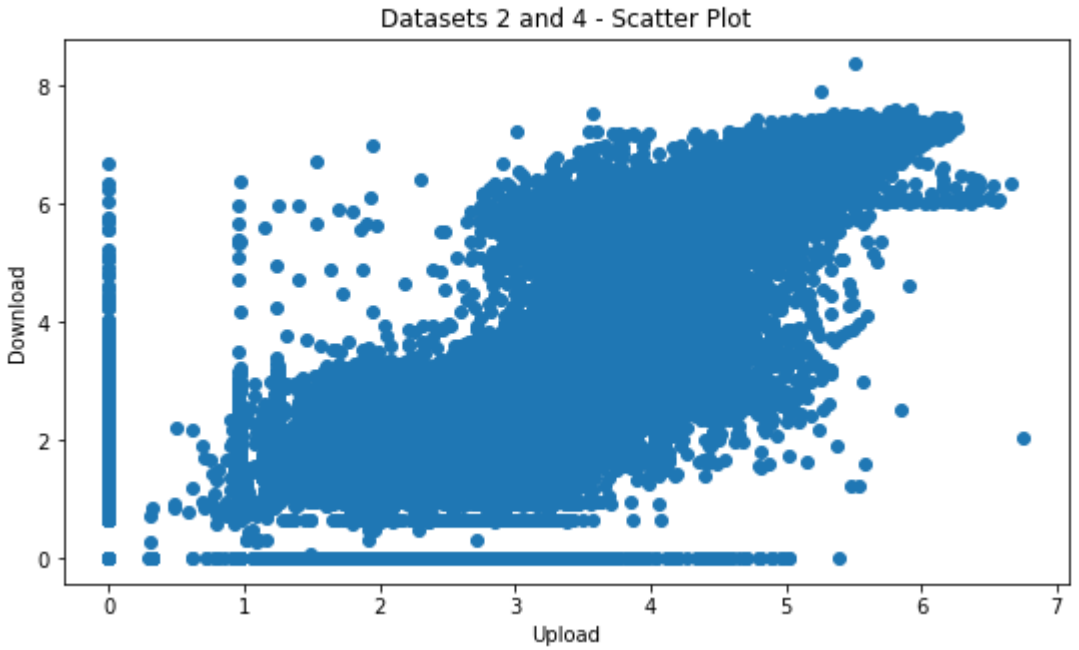
Para a obtenção dos gráficos de dispersão para a análise da correlação foi necessário mudar alguns datasets que se baseavam em horários diferentes e, portanto, tinham quantidades de dados diferentes. Para isso, foi recomendado que se utilizasse o horário obtido para o dataset de download.
Uma vez que isto estava resolvido, foi utilizada a função “**scatter**” do “**matplotlib**” para criar os gráficos desejados.

5.1 - Gráficos de Dispersão

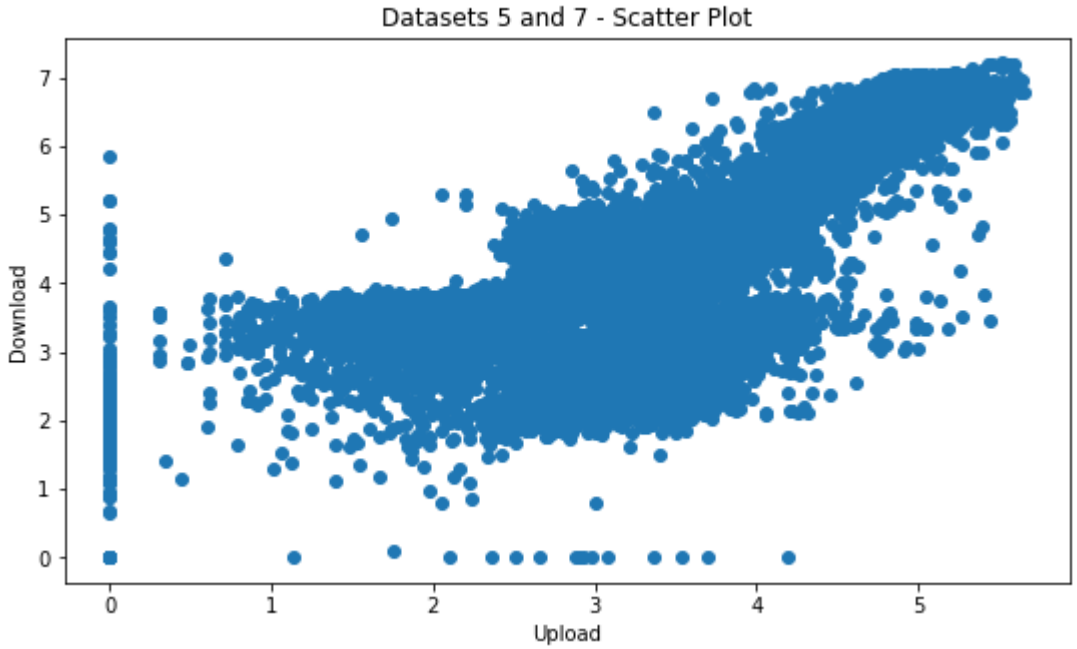
```
Correlation Coefficient = 0.9156089964784122
```



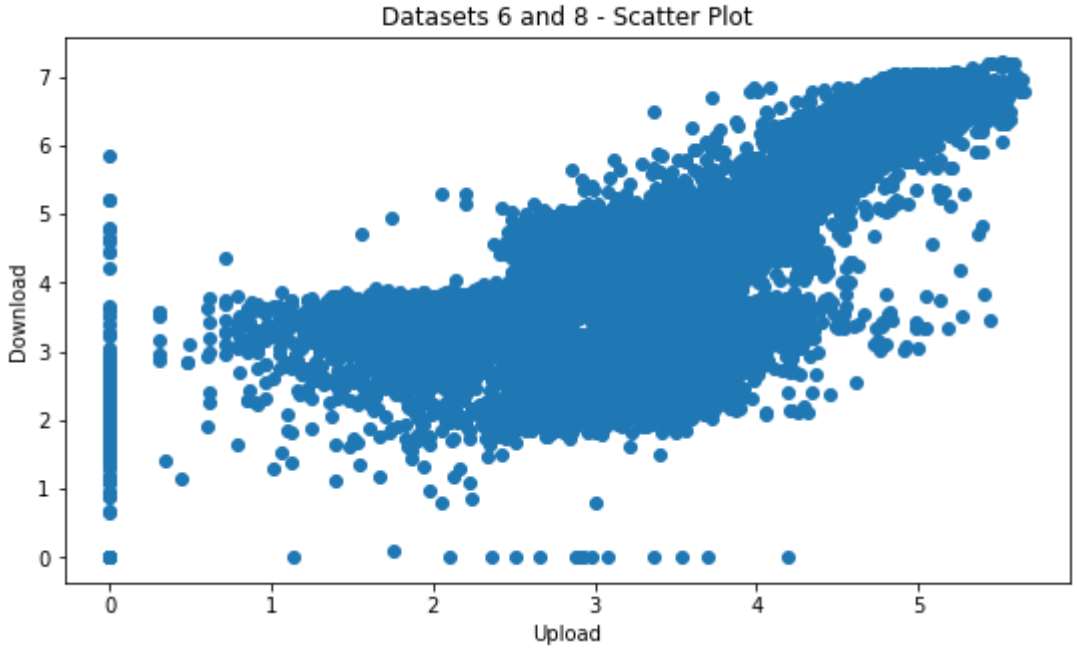

```
Correlation Coefficient = 0.9156089964784122
```



```
Correlation Coefficient = 0.7925043015217004
```




```
Correlation Coefficient = 0.7925043015217004
```



5.2 - Análise dos Dados

A partir dos gráficos e dos valores obtidos para o coeficiente de correlação (disponíveis acima de cada gráfico) podemos afirmar que as taxas de download e upload possuem sim uma correlação, embora na Smart TV essa correlação seja muito maior (com resultados maiores que 0.9) em comparação aos do Chromecast (que quase chegam a 0.8).

6 - Comparação dos dados gerados pelos dispositivos Smart-TV e Chromecast

6.1 - Comparações

```
Comparison - Datasets 1 and 5:
G_test = 1.705620571584578
P_value = 0.9999897858789901

Comparison - Datasets 2 and 6:
G_test = 1.705620571584578
P_value = 0.9999897858789901

Comparison - Datasets 3 and 7:
G_test = 2.4187217934803282
P_value = 0.9998971436246753

Comparison - Datasets 4 and 8:
G_test = 2.4187217934803282
P_value = 0.9998971436246753
```

6.2 - Análise dos Dados

Tendo em vista que o p_value de todas as comparações está muito alto (muito próximo de 1) e, portanto, é considerado insignificante, não podemos determinar nada com esta análise.