Projeto do Curso de Probabilística e Estatística
Uma Análise Crítica dos Dados de um Provedor de Internet de Médio Porte

Lucas da S. Inocêncio
Universidade Federal do Rio de Janeiro
lucas.inocencio@poli.ufrj.br

1 Dataset

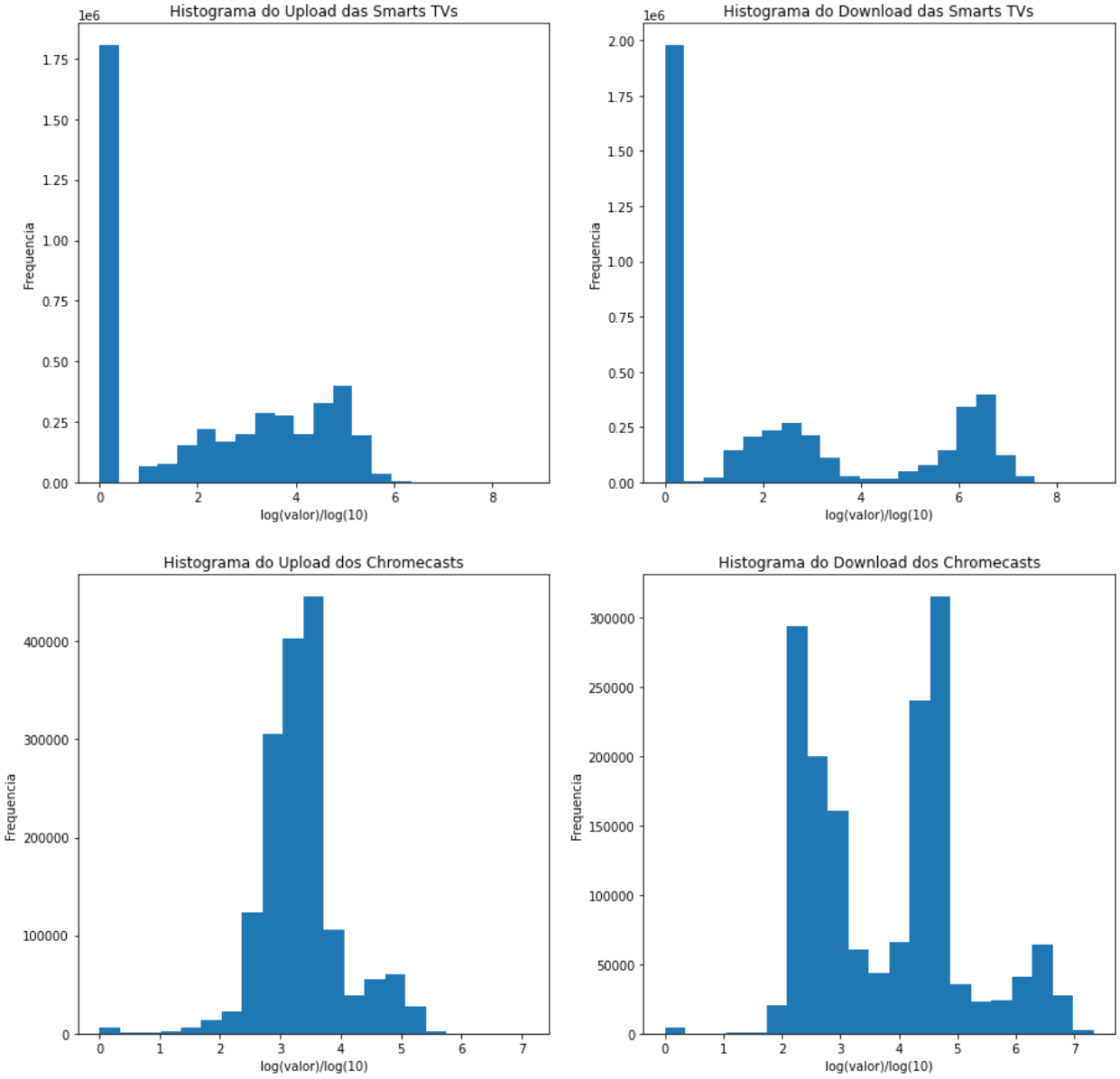
Para a análise dos arquivos foi utilizado a linguagem de programação Python na sua versão 3.10 e as bibliotecas datetime, matplotlib, numpy, pandas, scipy e statsmodels. O código fonte utilizado para fazer as análises pode ser encontrado aqui:
https://github.com/lucas-inocencio/Probability-and-Statistics

Para a leitura dos arquivos (1) dataset chromecast.csv e (2) dataset smart-tv.csv foi utilizado o método read_csv do pandas. Para tratamento das informações, os valores das taxas de upload e download foram somados 1 e depois reescalonadas para log na base 10 com o método log10 do numpy, conforme orientado pela orientadora.

2 Estatísticas gerais

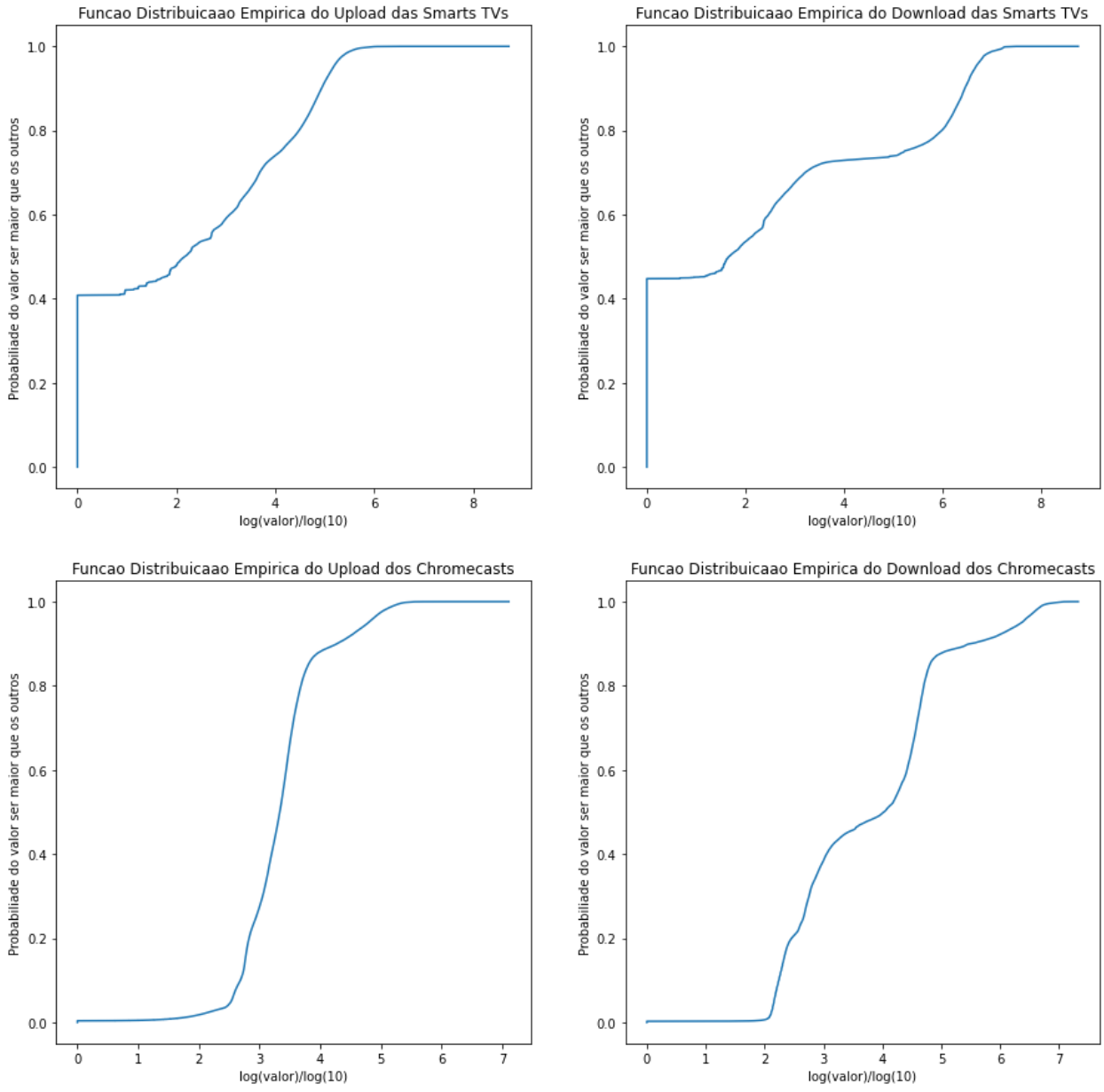
Segue a exibição dos histogramas da upload e download, das smart tvs e chromecasts plotados com matplotlib.pyplot.hist. com o parâmetro bins=sturges(data) como visto em aula sturges(data)=1 + 3.3*log(len(data)).

Figura 1: Histogramas das taxas de Upload/Download das Smart TVs/Chromecasts



É possível notar uma diferença nos resultados da coleta. Enquanto as TVs possuem vários registros com valor igual a zero, os chromecasts isso praticamente não existe. Isto talvez ocorra pela natureza dos dispositivos e, consequentemente, pelo comportamento dos usuários em relação a eles.

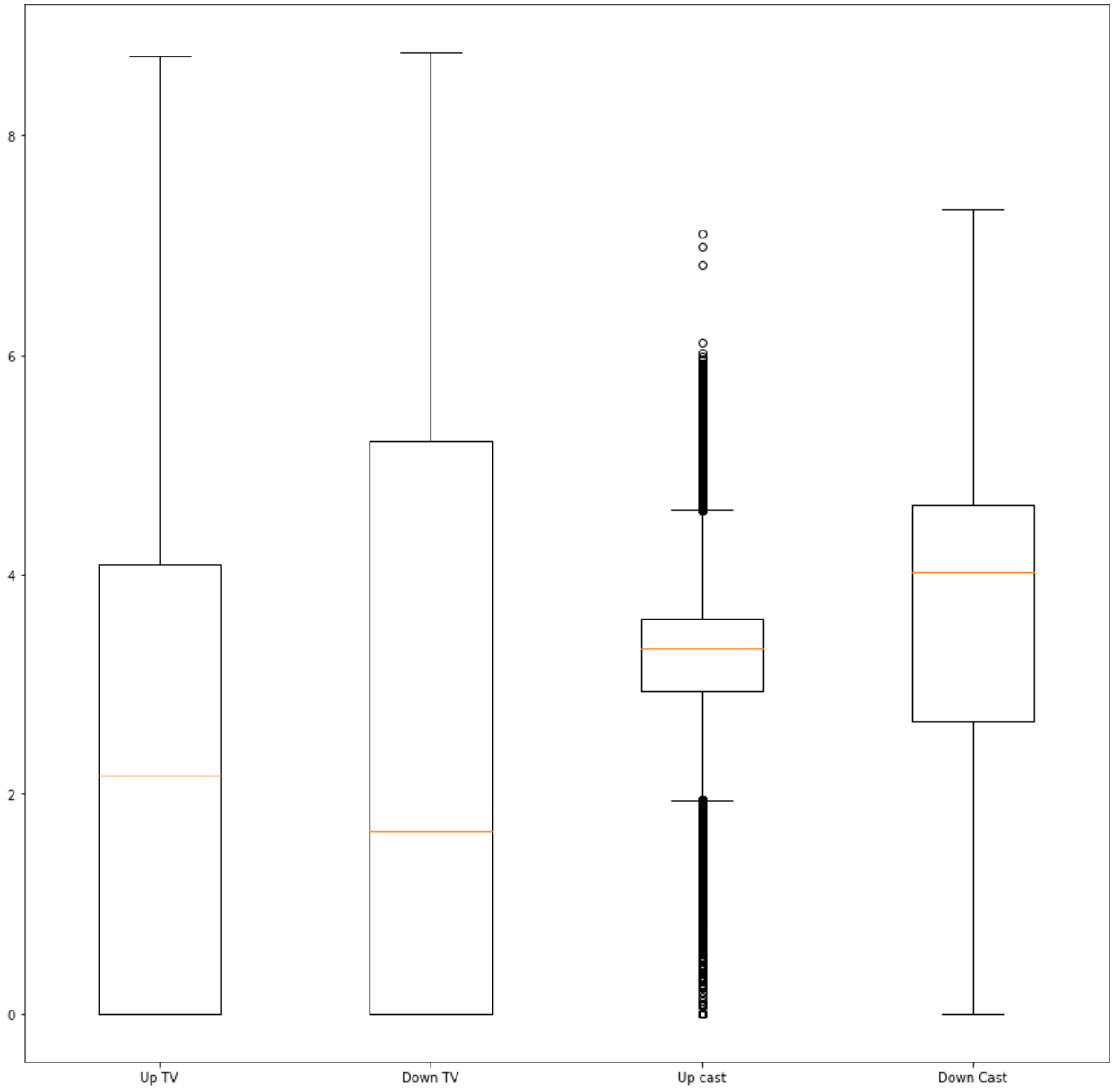
Figura 2: Funcoes Distribuicao Empirica Upload/Download, Smart TV/Chromecast



Para computar a Função Distribuição Empírica foi utilizado o método ECDF da biblioteca statsmodels. Aproximadamente houve um registro de 0 bps no tráfego de upload e download em 40% das ocorrências nas smart tvs, isso fica nítido neste gráfico. E que as medidas tráfego de upload dos chromecasts tem pouca dispersão.

Segue os boxplots do upload e download da smarts tv e do chromecasts, respectivamente. Para computá-los foi utilizado o método boxplot da Bíblioteca plot. Vale destacar a quantidade significativa de outliers na taxa de upload do chromecast.

Figura 3: Boxplots Upload/Download, Smart TV/Chromecast



Por fim essas foram as médias, variâncias e desvios padrões encontrados utilizando os métodos mean, var e std do numpy:

Tabela 1: Estatísticas Gerais dos Datasets

	Upload Smart TV	Download Smart tv	Upload Chromecast	Download Chrome cast
Log(media)	2.158	2.352	3.350	3.800
Log(variância)	4.110	6.721	0.460	1.664
Log(desvio padrão)	2.027	2.593	0.678	1.290

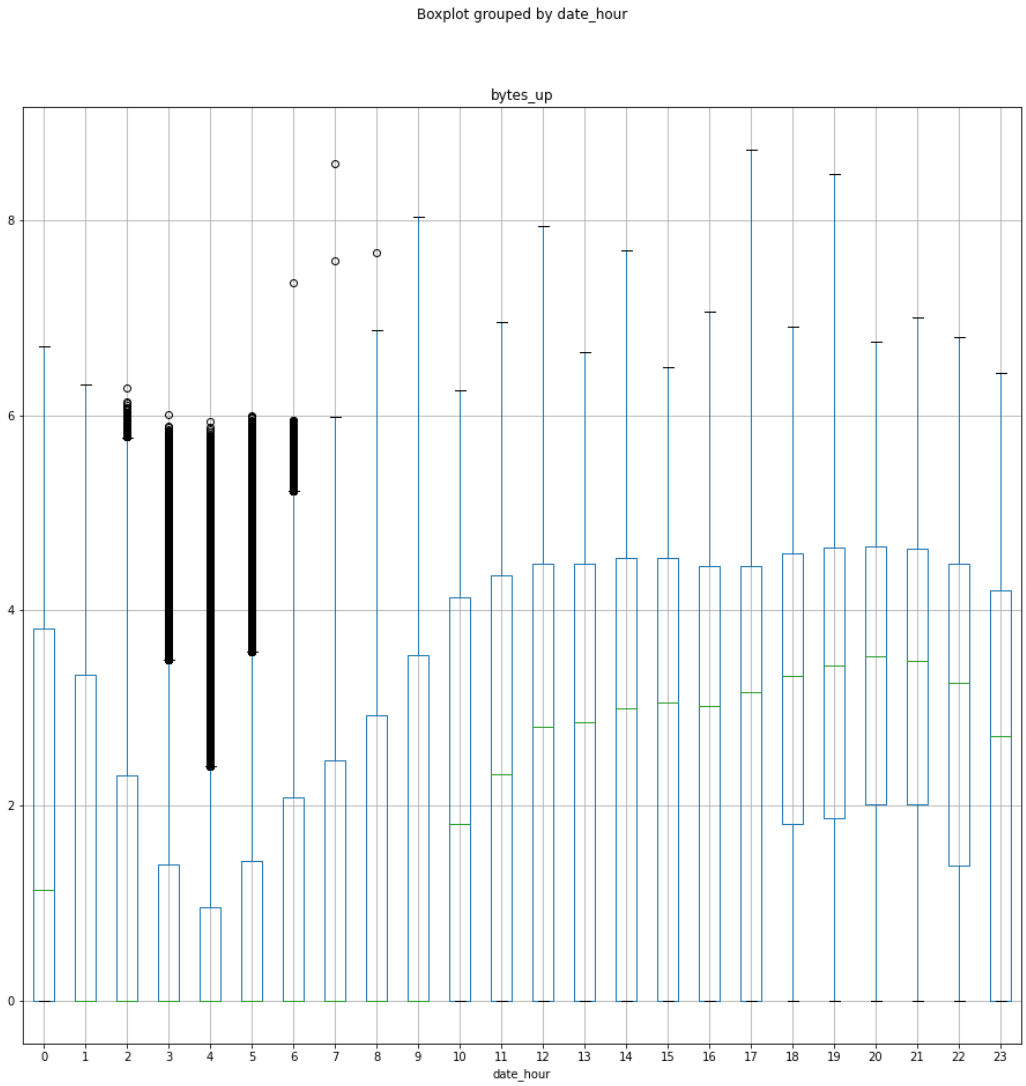
Como podemos observar os chromecasts consomem a internet significativamente mais que as smarts tvs, cerca de mais de uma ordem maior. Frequentemente, entre o mesmo dispositivo, não há diferença entre o consumo de download e upload.

3 Estatísticas por horário

Para fazer a análise por hora como desejado foi lido o arquivo csv e a coluna “date_hour” foi substituída de um tipo datetime completo para apenas um inteiro representando hora por meio do método strptime da biblioteca Datetime.

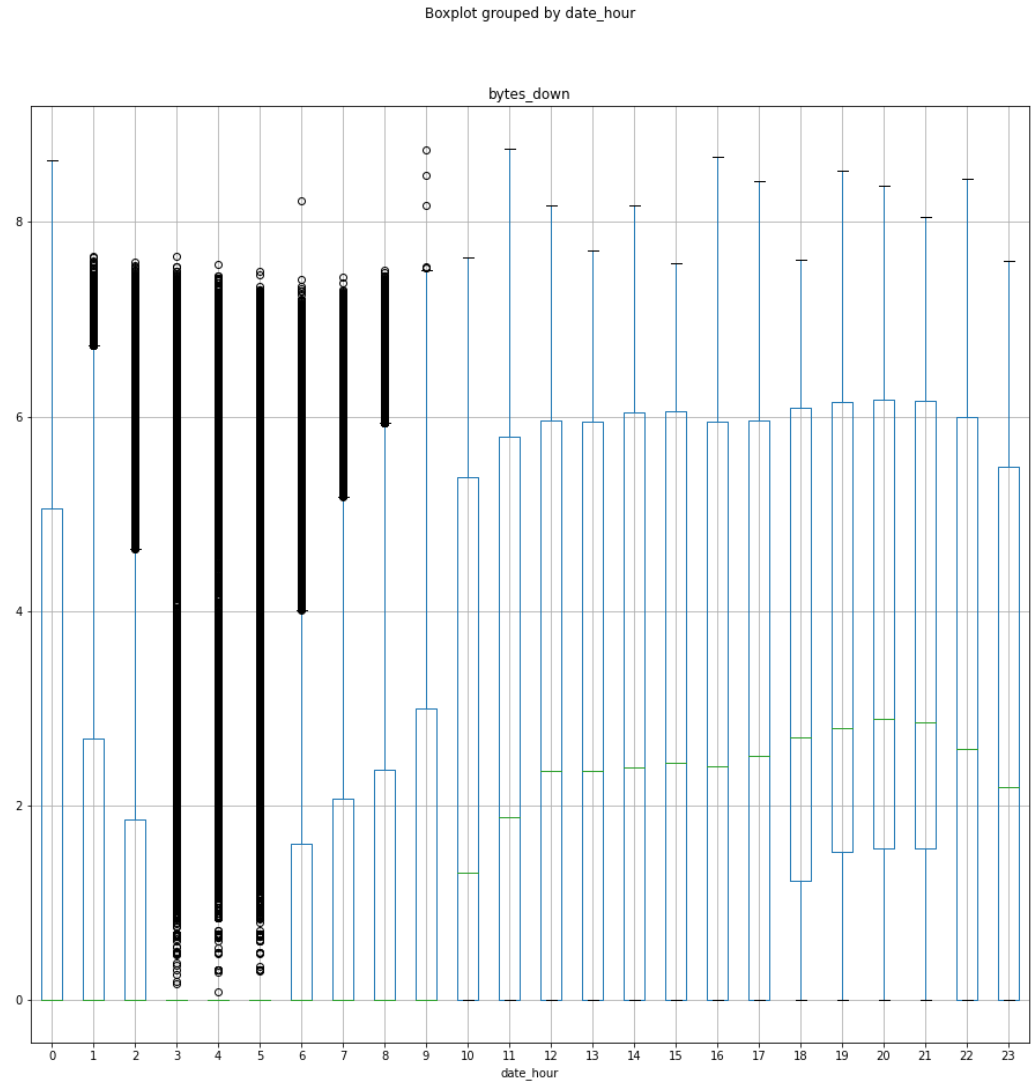
Segue abaixo os boxplots das taxas de uploads das smart tvs utilizando o método boxplot do Pandas e o parâmetro by="date_hour" para separar as colunas por horário. A de se notar a quantidade de outliers durante o período das 2 às 6, com destaque para às 4 horas da manhã. Estas medidas podem indicar um comportamento de alguns usuários ser majoritariamente noturno, apesar da maioria ser matutino.

Figura 4: Boxplots Upload Smart TV



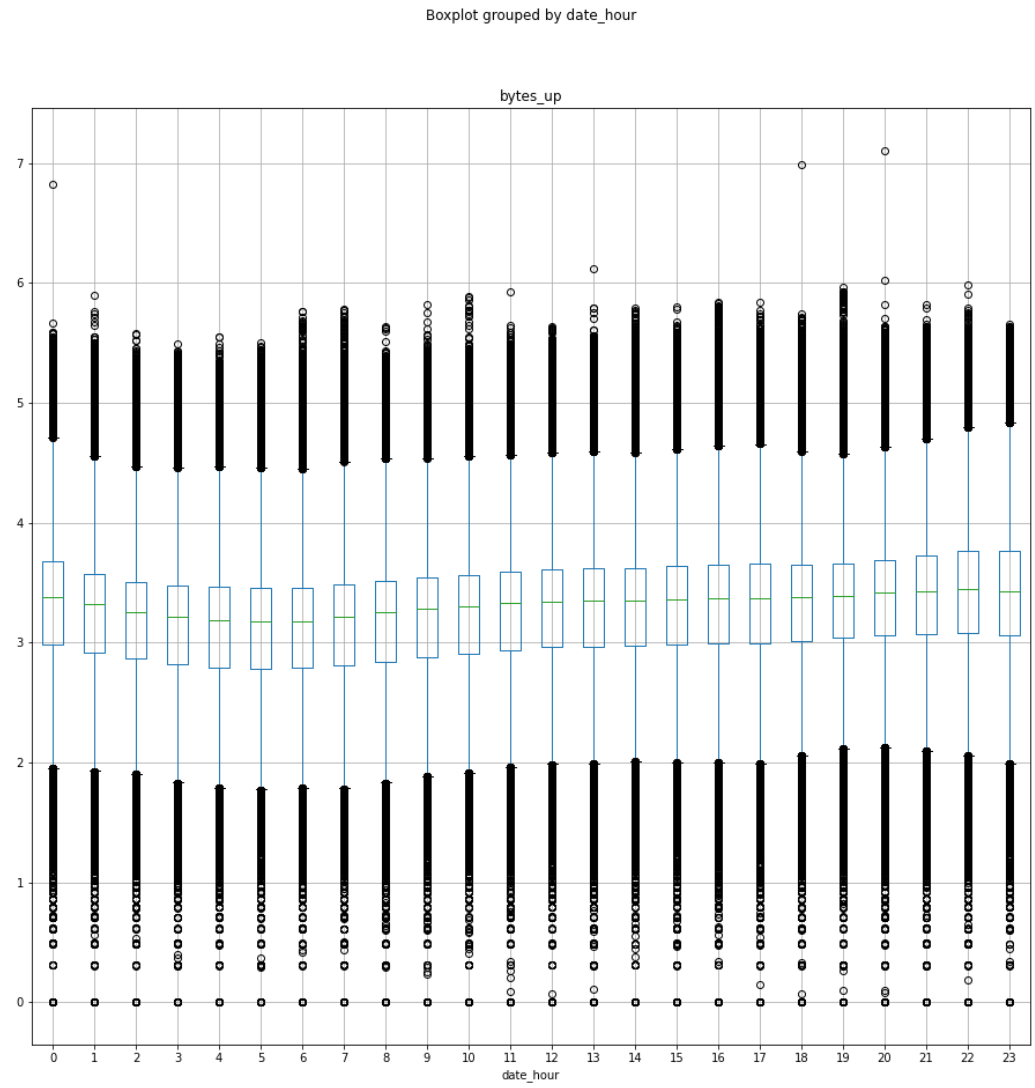
Para computar os outros boxplots foi seguido um algoritmo análogo. Segue abaixo os outros boxplots.

Figura 5: Boxplots Download Smart TV



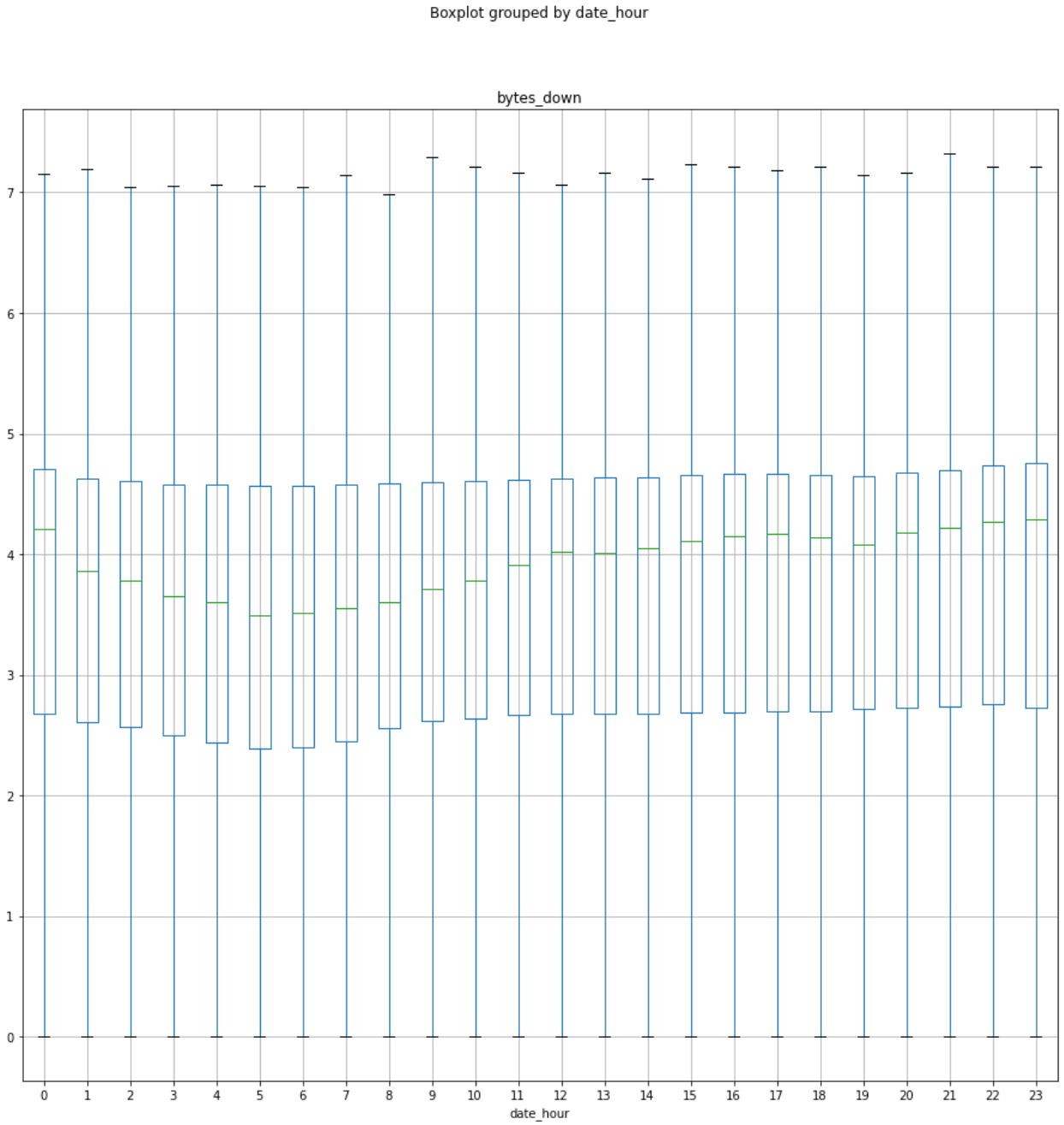
Observamos um padrão semelhante, mas agora estendido das 1 às 9.

Figura 6: Boxplots Upload Chromecast



Enquanto nos chromecasts os dispositivos mantêm uma média de taxa de upload constante durante todo o dia. Até os outliers são constantes.

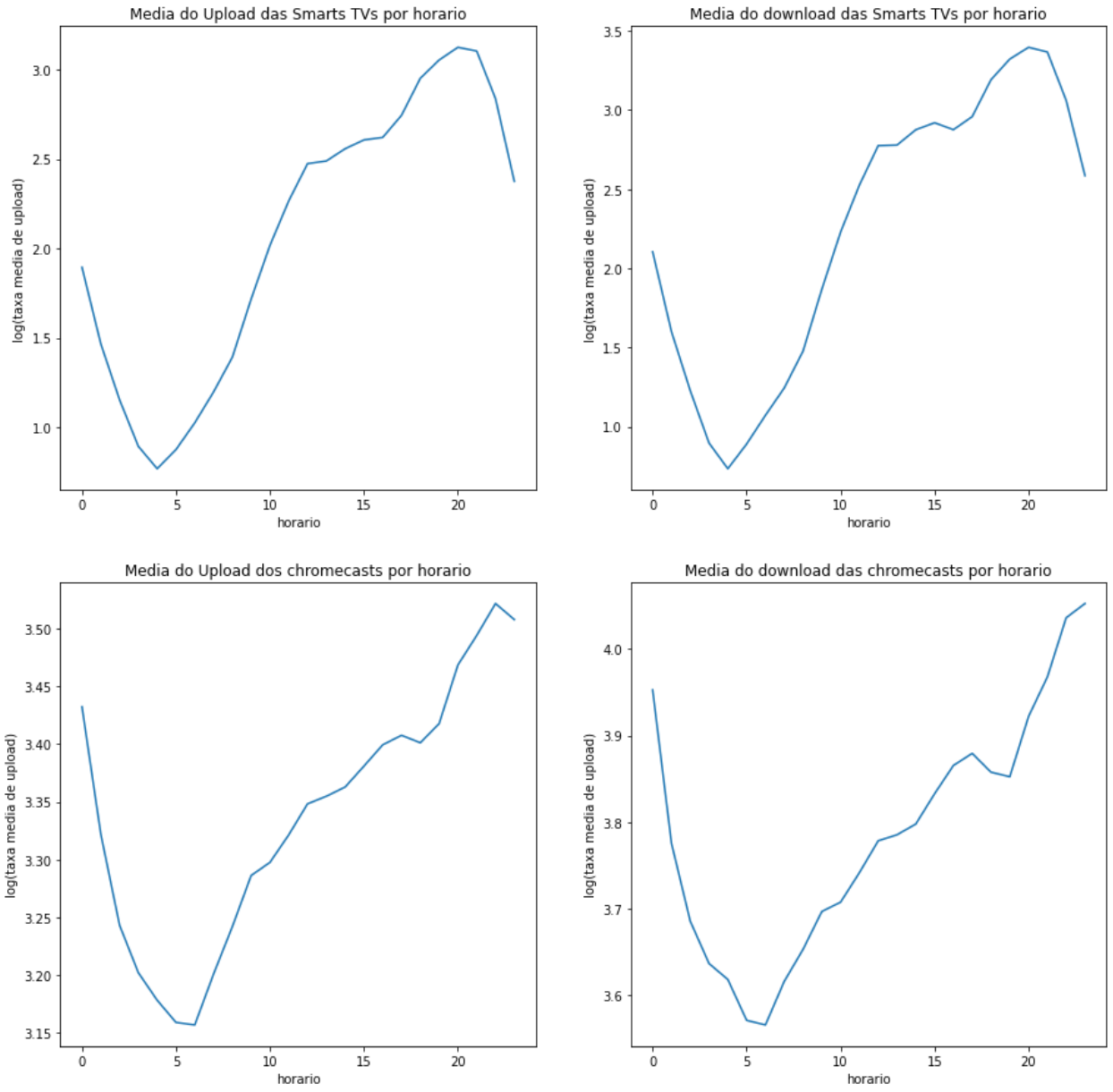
Figura 7: Boxplots Download Chromecast



Mesmo padrão de constância, mas sem nenhuma presença de outliers observada.

Segue abaixo da média por horário, por taxa de upload e download e por dispositivo. Para computá-los foram utilizados os métodos groupby e mean do Pandas.

Figura 8: Medias Upload/Download por Horario das Smart TVs/Chromecasts

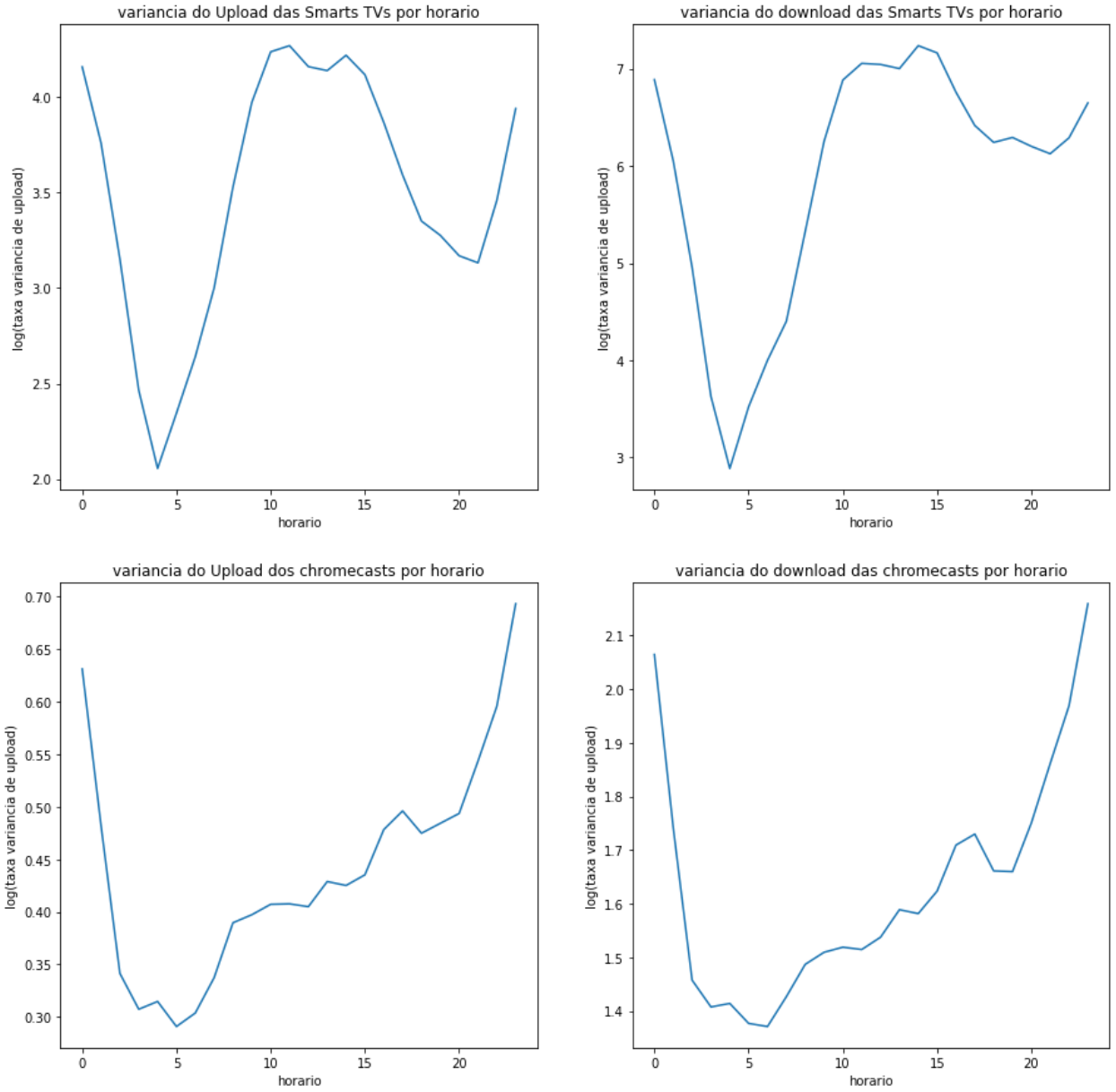


Novamente podemos observar a semelhança entre as taxas de upload e download no mesmo dispositivo e a diferença entre as taxas quando comparamos os

dispositivos. Devemos nos atentar neste gráfico é que o pico de consumo ocorre por volta das 20 horas da noite nas smarts tvs e as 23 nos chromecasts. E que o vale acontece aproximadamente às 5 para ambos.

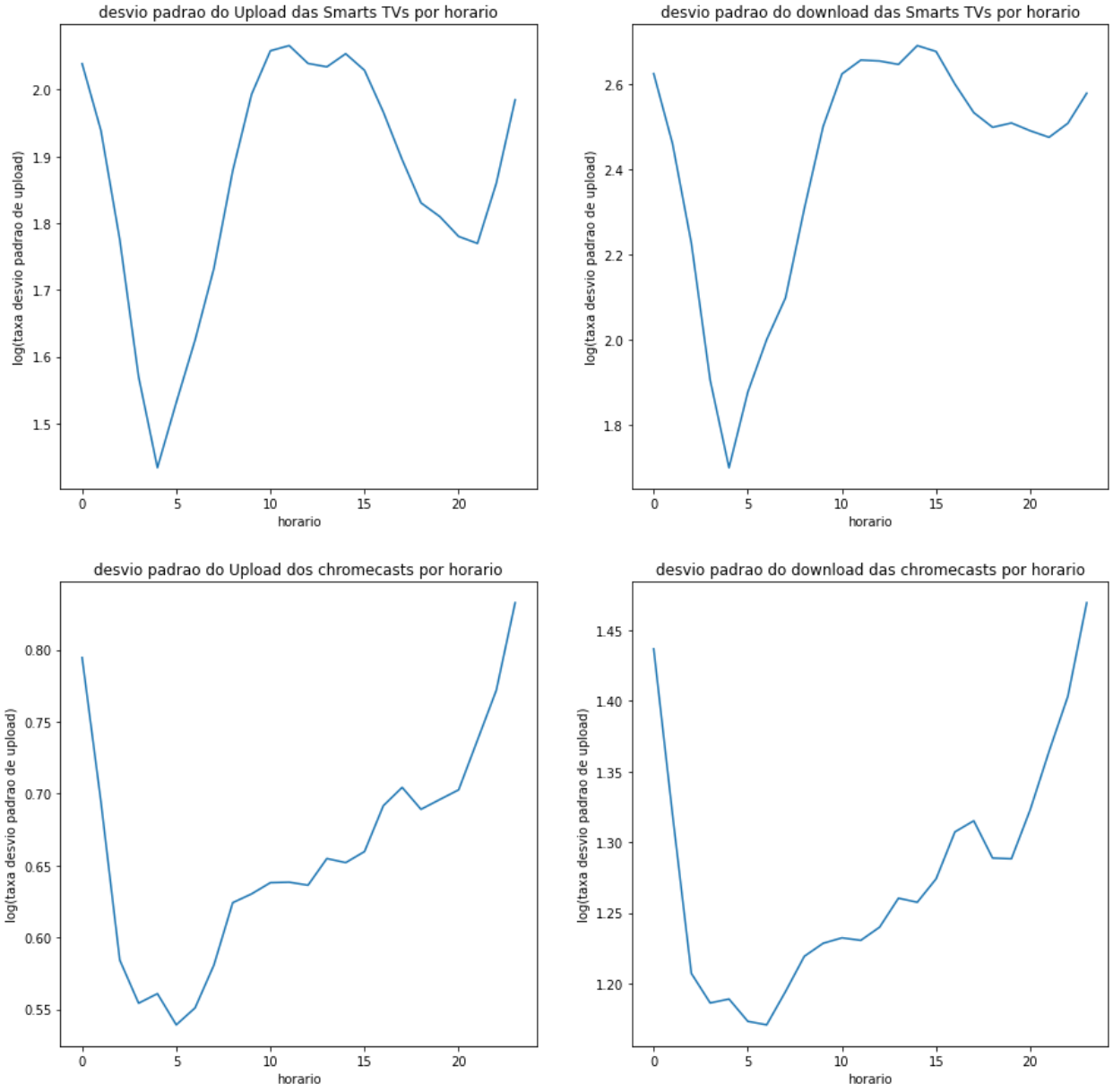
Segue abaixo da variância por horário, por taxa de upload e download e por dispositivo. Para computá-los foram utilizados os métodos groupby e var do Pandas.

Figura 9: Variancias Upload/Download por Horario das Smart TVs/Chromecasts



Podemos observar a diferença de tráfego nas Smart TVs durante o período das 9 às 17 através do gráfico da variância. Nos chromecasts esta diferença ocorre de das 20 às 0. Uma hipótese é que isto possa indicar o comportamento do público infanto-juvenil de consumo de TV durante o horário comercial, mais dados são necessários para realizar esta afirmação.

Figura 10: Desvios Padrões Upload/Download por Horário das Smart TVs/Chromecasts

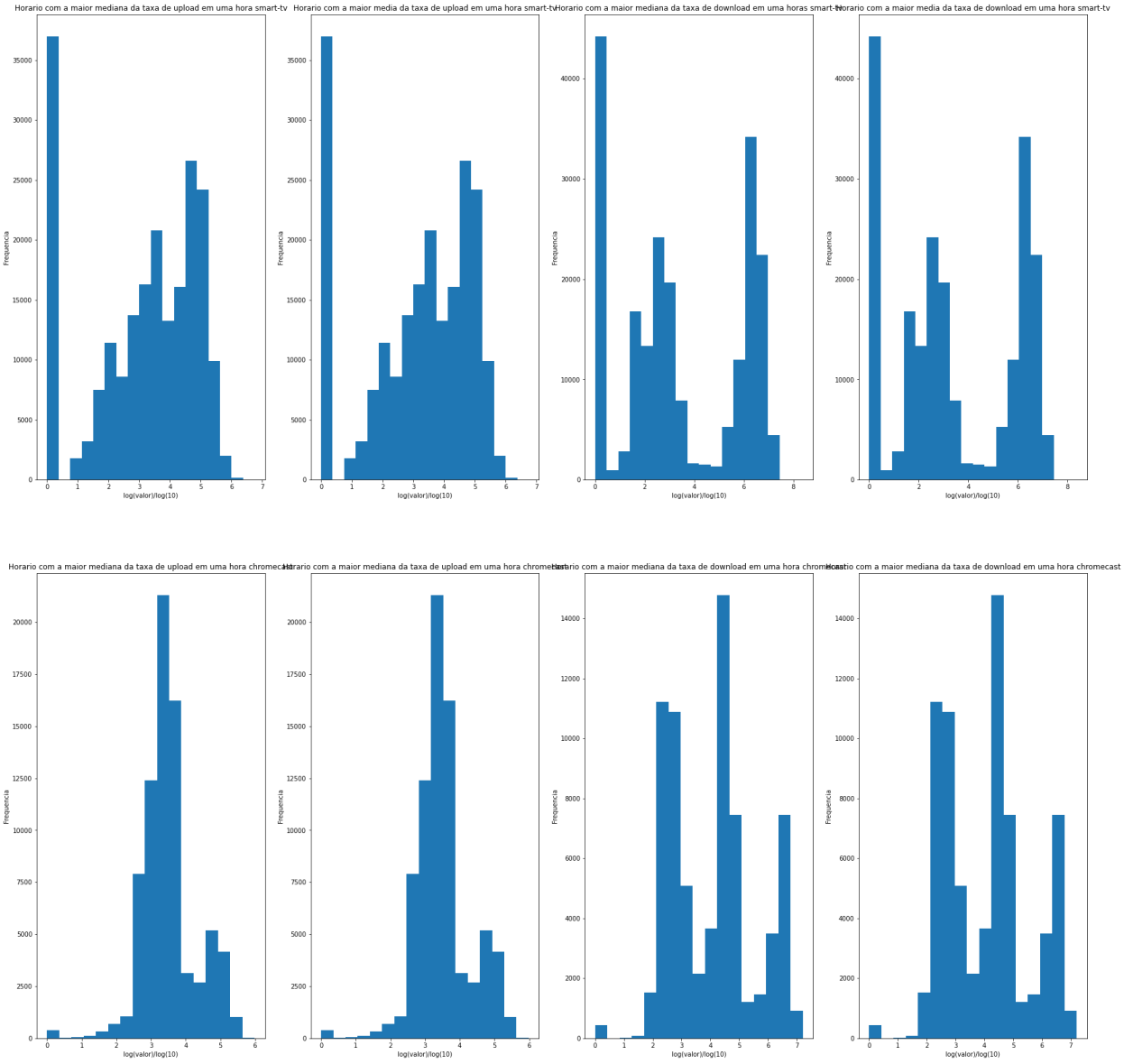


Desvio Padrão semelhante a variância. Nada a pontuar.

4 Caracterizando os horários com maior valor de tráfego

Utilizando os métodos groupby, mean, median e argmax foi possível encontrar os horários para os 8 conjuntos de dados, como especificado e requisitado, que são, respectivamente: 20, 20, 20, 20, 22, 22, 23 e 23. Segue os histogramas para os 8 datasets.

Figura 11: Histogramas dos datasets



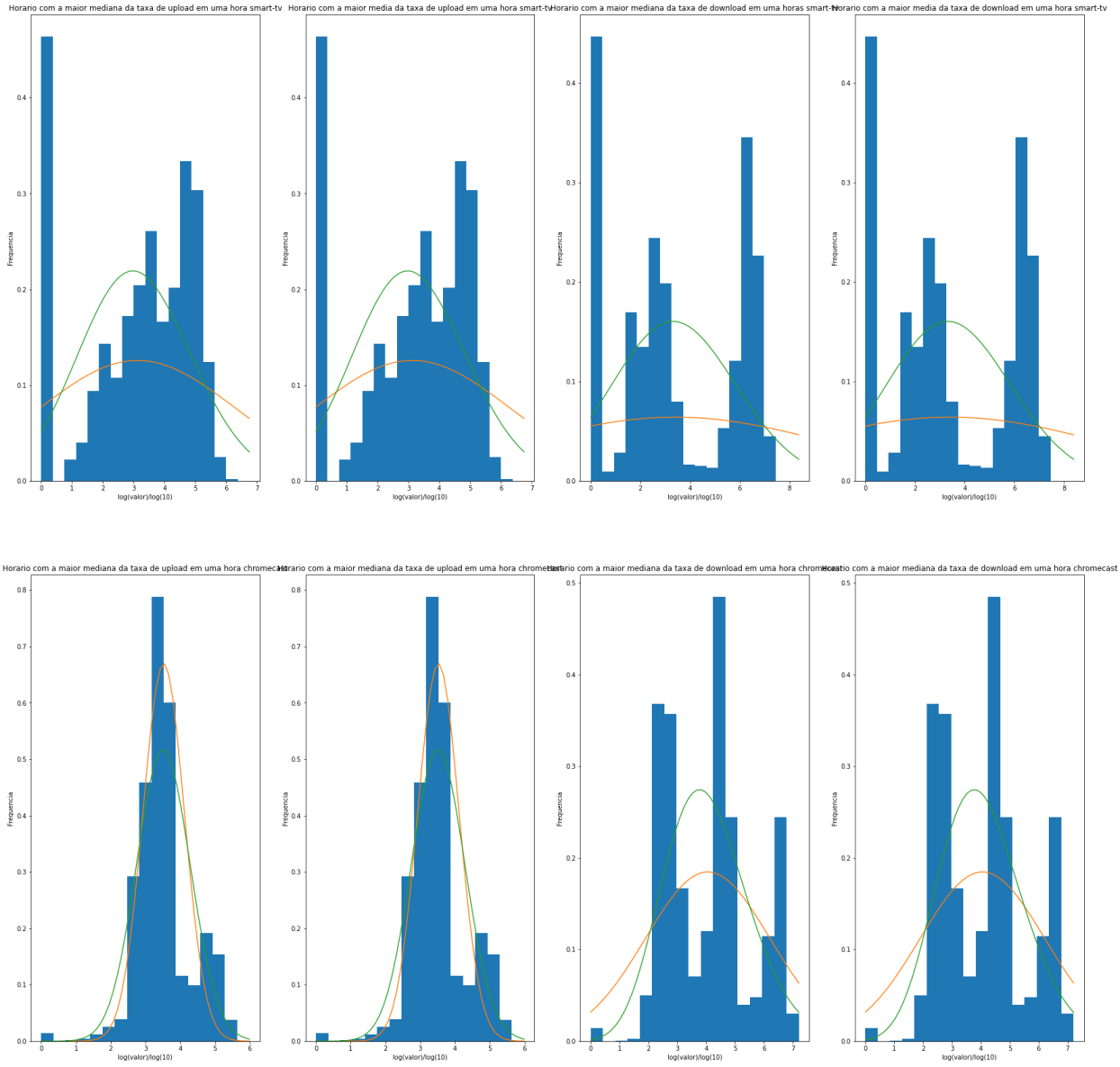
Pela definição de MLE podemos derivar que para Gaussiana o MLE da média será a média amostral e a variância a variância amostral. Os parâmetros da gaussiana foram estimados utilizando os métodos mean e var do Numpy para cada dataset. Enquanto para a gamma foi usado o método gamma.fit da biblioteca scipy.

Tabela 2: Parâmetros MLE dos datasets

	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5	Dataset 6	Dataset 7	Dataset 8
Gaussiana média	3.124	3.124	3.396	3.396	3.521	3.521	4.053	4.053
Gaussiana variância	3.168	3.168	6.201	6.201	0.596	0.596	2.159	2.159
Gamma "shape”	220.481	220.481	896.547	896.547	3148.881	3148.881	27.130,	27.130,
Gamma localização	-23.961	-23.961	-71.062	-71.062	-39.809	-39.809	-3.631	-3.631
Gamma escala	0.122	0.122	0.083	0.083	0.014	0.014	0.283	0.283

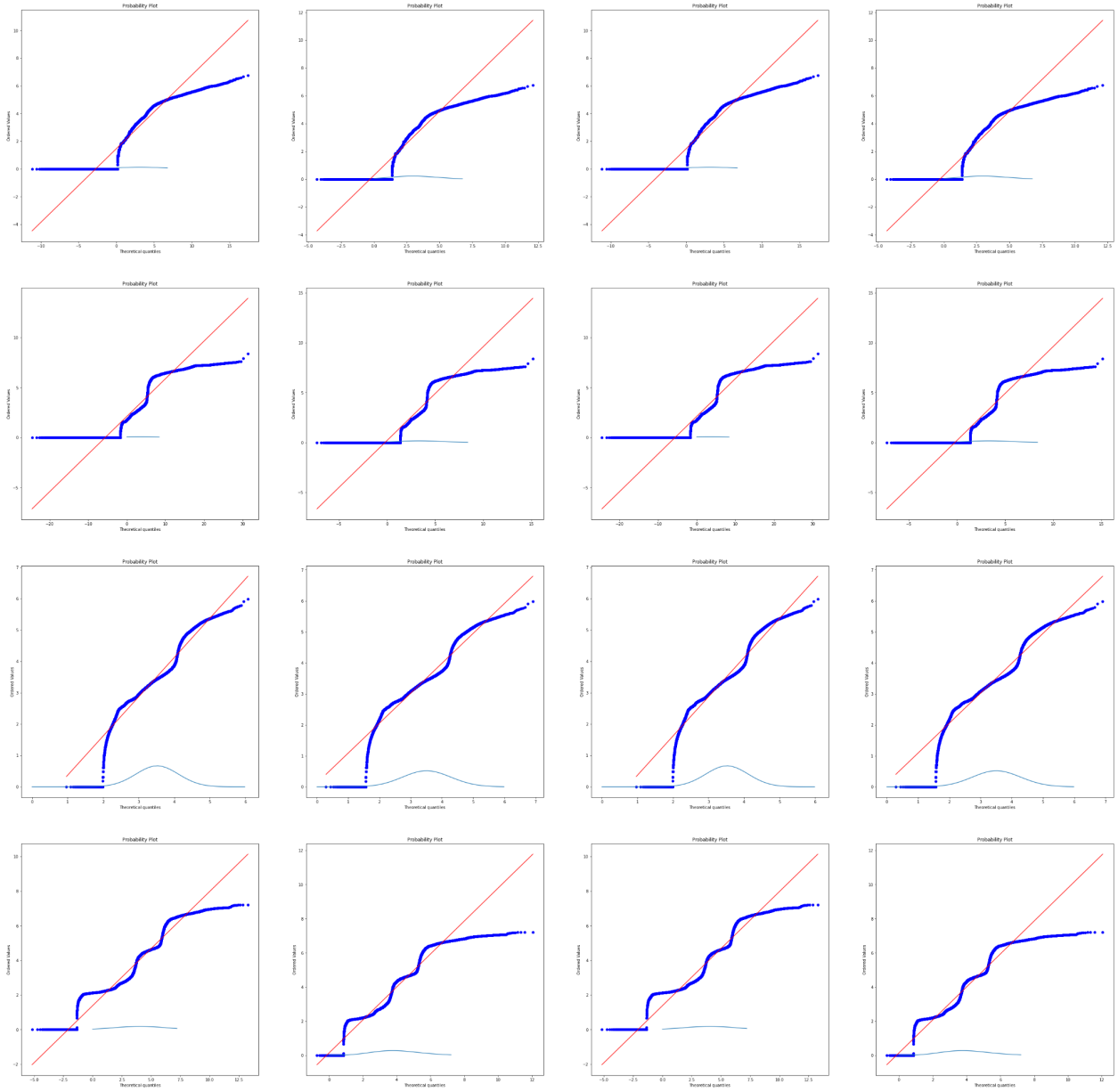
Dado os dados descritos acima foi plotado os três graficos juntos como especificado com os metodos linespace, min e max do numpy sob os datasets para delimitar o eixo das abscissas e para das ordenas foi utilizado os metodos norm.pdf e gamma.pdf juntamente com os parametros supracitdados.

Figura 12: Histogramas dos Datasets com as Gaussianas e Gammas Estimadas



Segue abaixo os gráficos do probability plot de cada dataset por gaussiana e gamma, respectivamente.

Figura 13: Probability plots Dataset x MLEs estimados



1. Quais foram os horários escolhidos para cada dataset?

Os horários para os 8 conjuntos de dados respectivamente são (20, 20, 20, 20, 22, 22, 23, 23).

2. O que você pode observar a partir dos histogramas dos datasets?

Os histogramas dos chromecasts puderam ser aproximados pela normal e gamma, enquantos das tvs não foi ideal.

3. Comente sobre as diferenças e/ou similaridades entre os datasets 1 e 2, 3 e 4, 5 e 6, 7 e 8. O objetivo é comparar as características dos datasets com a maior mediana em um determinado horário com os datasets com a maior média em um determinado horário.

Todos datasets que deveriam ser diferentes ao método de seleção da média e da mediana demonstraram ser semelhantes. Isto se dá porque a mediana se aproximou muito da média e a dispersão se assemelha a uma gaussiana.

4. É possível caracterizar os datasets acima por uma variável aleatória da literatura?

Sim. Os datasets 1, 2, 3 e 4 poderiam ser aproximados por uma bimodal, enquanto os 5, 6, 7 e 8 uma normal já se mostrou suficiente.

5. Se a resposta for não, qual o motivo?

Foi respondido sim.

6. O que você pode observar a partir dos gráficos Probability Plot?

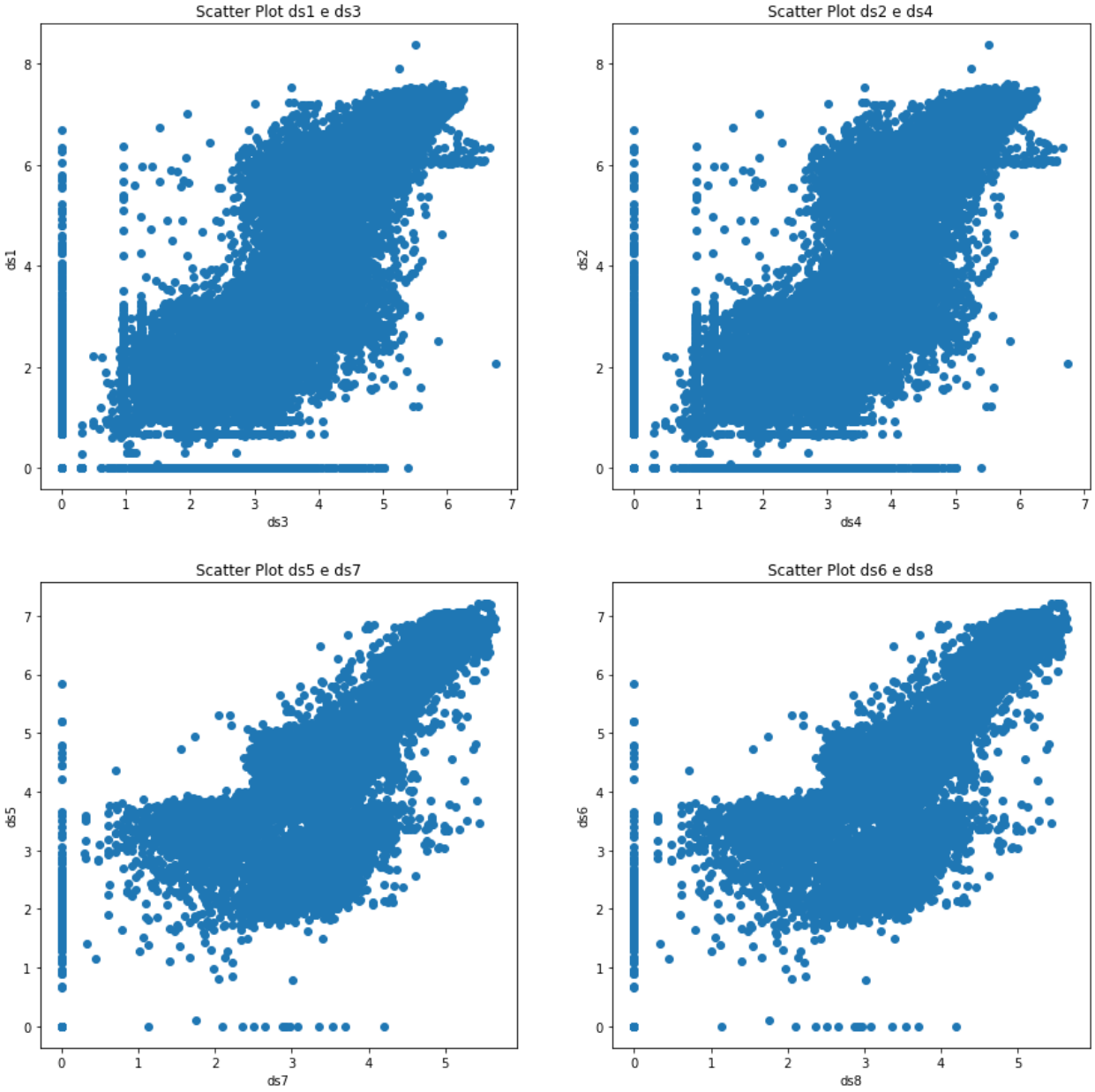
A hipótese originada da observação matematicamente descrita que os histogramas dos chromecasts puderam ser aproximados pela normal e gamma, enquanto das tvs não foi ideal. Respondido na questão 2.

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

Para computar o coeficiente de correlação foram utilizados os métodos query e corr do Pandas. Para exibir foi utilizado o método scatter do matplotlib. Segue abaixo os dados e gráficos dos datasets 1-3, 2-4, 5-7, 6-8. Como o horário dos datasets 5 e 6 divergiam de 7 e 8, para fazer a correlação igualá-los pelo do download, conforme recomendação da orientadora.

0.915, 0.915, 0.792 e 0.792

Figura 14: Scatter plots



Como podemos ver os os datasets 5-7 e 6-8 retornaram valores menores de correlação, mas ainda relativamente bem altos como os 1-3 e 2-4. Isto se deve à natureza do comportamento dos usuários frente a estes dispositivos.

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

O g-test foi feito utilizando o argumento g do método chi2_contingency da biblioteca scipy conforme especificado. Segue abaixo os resultados de upload e download das tvs e chromecats, respectivamente.

0.416, 0.416, 0.541 e -6.778e-16

7 Relatório

Definitivamente estes dados podem ajudar a compreender o produto consumido e ajudar a oferecer um melhor serviço. Conforme explicado no texto irei resumir aqui diversos pontos. Como por exemplo, agendar manutenções nas redes para o início da manhã, por exemplo, que o upload e download estão relacionados, que o consumo depende do dispositivo, um possível perfil de quem consome determinado horário, a quantidade das taxas de tráfego da maioria vs dos outliers, dentre outros.