

Danilo de Paula Perl

Análise Estatística do Banco de Dados de Furtos e Roubos de Carros da Secretaria de Segurança Pública do Estado de São Paulo

Brasil
2020

Danilo de Paula Perl

**Análise Estatística do Banco de Dados de Furtos e
Roubos de Carros da Secretaria de Segurança
Pública do Estado de São Paulo**

TRABALHO DE CONCLUSÃO DO
CURSO DE BACHARELADO EM MA-
TEMÁTICA APLICADA E COMPUTA-
CIONAL COM ÊNFASE EM CIÊNCIAS
ATUARIAIS.

UNIVERSIDADE DE SÃO PAULO – USP
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA

Orientador: Prof.^a Dr.^a Cláudia Monteiro Peixoto

Brasil
2020

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me conduzido durante todos os anos da graduação e de forma específica neste projeto com sabedoria, saúde e forças para chegar até o final.

Sou grato à minha mãe pelo apoio que sempre me deu durante toda a minha vida, principalmente durante a elaboração deste projeto.

Deixo aqui registrado um agradecimento especial à minha orientadora Professora Doutora Cláudia Monteiro Peixoto pela dedicação do seu escasso tempo ao meu trabalho de conclusão de curso.

Também quero agradecer à Universidade de São Paulo e a todos os professores do meu curso pela elevada qualidade do ensino oferecido.

RESUMO

Com o incentivo aos programas de transparência das instituições públicas e com o aumento crescente da liberação de bases de dados públicos, torna-se cada vez mais importante ter trabalhos acadêmicos que analisem e explorem tais informações para obter conclusões e poder auxiliar os órgãos públicos. Diante disso, este trabalho tem por objetivo fazer a análise estatística do banco de dados de furtos e roubos de carros da Secretaria de Segurança Pública do Estado de São Paulo no ano de 2018. Inicialmente, buscamos realizar uma análise descritiva para a quantidade de furtos e roubos por dia da semana e por período do dia, além de dois testes de aderência para verificar se a distribuição da quantidade de furtos e roubos de carros por dia da semana se aproxima de uma *Poisson*. Como resultado, observamos, em média, mais roubos do que furtos em cada dia da semana e que ocorrem em média mais roubos do que furtos de noite e de madrugada. Para ambos os testes de aderência, não encontramos evidências para aceitar a hipótese nula, que diz que a distribuição de ocorrências de furtos/roubos para cada dia da semana do ano de 2018 se aproxima de uma *Poisson*.

Palavras-chaves: Distribuição de *Poisson*. Teste de hipótese. Teste Qui-Quadrado.

ABSTRACT

With the incentive to transparency programs of public institutions and with the increasing release of public databases, it becomes more and more important to have academic works that analyze and explore this information to obtain conclusions and be able to assist public agencies. Therefore, this work aims to perform a statistical analysis of the database of thefts and robberies of cars of the São Paulo State Public Security Secretariat in the year 2018. Initially, we sought to perform a descriptive analysis for the number of thefts and robberies per day of the week and per period of the day, in addition to two adherence tests to verify if the distribution of the number of thefts and robberies of cars per day of the week approaches a *Poisson*. As a result, we see, on average, more robberies than thefts on each day of the week, and on average, more robberies occur than night and dawn thefts. For both adherence tests, we found no evidence to accept the null hypothesis, which says that the distribution of thefts/robberies for each day of the week in the year 2018 approaches a *Poisson*.

Key-words: *Poisson* distribution. Hypothesis test. Chi-Square test.

LISTA DE ILUSTRAÇÕES

Figura 3.1 – Mapa de calor da ocorrência de furtos em 2018 na Avenida Paulista.	25
Figura 3.2 – Mapa de calor da ocorrência de furtos em 2018 na Cidade Universitária Armando de Salles Oliveira.	26
Figura 3.3 – Mapa de calor da ocorrência de roubos em 2018 na Avenida Paulista.	27
Figura 3.4 – Mapa de calor da ocorrência de roubos em 2018 na Cidade Universitária Armando de Salles Oliveira.	28
Figura 3.5 – Mapa de marcadores da ocorrência de furtos em 2018 na Avenida Paulista.	29
Figura 3.6 – Mapa de marcadores da ocorrência de furtos em 2018 na Cidade Universitária Armando de Salles Oliveira.	30
Figura 3.7 – Mapa de marcadores da ocorrência de roubos em 2018 na Avenida Paulista.	31
Figura 3.8 – Mapa de marcadores da ocorrência de roubos em 2018 na Cidade Universitária Armando de Salles Oliveira.	31
Figura 3.9 – Gráfico de <i>boxplot</i> para a quantidade de furtos por dia da semana.	33
Figura 3.10–Gráfico de <i>boxplot</i> para a quantidade de roubos por dia da semana.	34
Figura 3.11–Gráficos <i>boxplot</i> para a quantidade de furtos por período do dia.	35
Figura 3.12–Gráficos <i>boxplot</i> para a quantidade de roubos por período do dia.	36

LISTA DE TABELAS

Tabela 1.1 – Exemplo de tabela de frequências usada no teste Qui-Quadrado	19
Tabela 3.2 – Estatísticas para a quantidade de furtos por dia da semana. . .	32
Tabela 3.3 – Estatísticas para a quantidade de roubos por dia da semana. . .	32
Tabela 3.4 – Estatísticas para a quantidade de furtos por período do dia. . .	34
Tabela 3.5 – Estatísticas para a quantidade de roubos por período do dia. . .	35
Tabela 3.6 – Tabela de resultados do teste de Aderência Qui-Quadrado para a quantidade de furtos por dia da semana.	37
Tabela 3.7 – Tabela de resultados do teste de Aderência Qui-Quadrado para a quantidade de roubos por dia da semana.	38

SUMÁRIO

Introdução	15
I CONCEITOS BÁSICOS DA TEORIA DA PROBABILIDADE	17
1.1 Tópicos de probabilidade e estatística	17
1.1.1 Distribuição de Poisson	17
1.1.2 Lei Forte dos Grandes Números	17
1.2 Teste de hipótese	18
1.2.0.1 Decisão do teste baseada no nível descritivo ou p-valor	18
1.2.0.2 Testes de Aderência	18
1.2.0.3 Teste Qui-Quadrado para Aderência	19
II METODOLOGIA	21
2.1 Criação de base de dados da SSP	21
2.2 Criação dos mapas de calor/marcadores com os dados da SSP	21
2.3 Geração de análises descritivas e gráficos	23
2.4 Checagem para testar se as ocorrências de B.O.'s se aproximam de uma distribuição de Poisson	23
III RESULTADOS	25
3.1 Mapas de calor	25
3.1.1 Mapas de calor para furtos	25
3.1.2 Mapas de calor para roubos	27
3.2 Mapas de marcadores	29
3.2.1 Mapas de marcadores para furtos	29
3.2.2 Mapas de marcadores para roubos	30
3.3 Medidas Descritivas	32
3.3.1 Resultados para furtos e roubos por dia da semana	32
3.3.2 Resultados para furtos e roubos por período do dia	34
3.4 Testes de Aderência e Resultados	36
3.4.1 Resultado do teste de Aderência para furtos	36
3.4.2 Resultado do teste de Aderência para roubos	37
IV CONCLUSÕES	39
4.1 Conclusões	39
Referências	41

ANEXOS	43
ANEXO A – SCRAPPER PARA DOWNLOAD DOS DADOS DA SSP	45
ANEXO B – JUNÇÃO DOS DADOS DA SSP	51
ANEXO C – CRIAÇÃO DAS BASES DE FURTOS E ROUBOS DE 2018	55
ANEXO D – CRIAÇÃO DOS MAPAS DE CALOR/MARCADORES	59
ANEXO E – GERAÇÃO DA ANÁLISE DESCRIPTIVA DOS DADOS	63
5.1 Código para realização da análise descritiva referente a base de furtos	63
5.2 Código para realização da análise descritiva referente a base de roubos	69
ANEXO F – REALIZAÇÃO DO TESTE DE HIPÓTESE PARA FURTOS E ROUBOS	75
6.1 Realização do teste de hipótese para os dados de furtos	75
6.2 Realização do teste de hipótese para os dados de roubos	80

INTRODUÇÃO

A análise estatística de bases de dados públicas é útil não somente para obter conclusões mas também para poder auxiliar instituições públicas a utilizarem seus dados da melhor maneira. Utilizando como base a tese de mestrado (FEIJÓ, 2015) que parte do pressuposto "Aqui é dado especial destaque ao caso em que o processo do número de indenizações é de *Poisson*", tivemos inicial interesse em reafirmar este artigo com os dados públicos da SUSEP <http://www2.susep.gov.br/menuestatistica/Autoseg/principal.aspx>. Porém, devido a várias questões levantadas sobre a base de dados, resolvemos apenas realizar uma análise descritiva dos dados de furtos e roubos de carros da Secretaria de Segurança Pública do Estado de São Paulo e testar se as distribuições dos furtos e roubos podem ser aproximadas por distribuições de *Poisson*. Assim, se tivermos uma distribuição de *Poisson* para número de roubos e furtos, seria mais crível que o número de indenizações também teria uma distribuição de *Poisson*. Para isso, acessamos as bases de dados em <http://www.ssp.sp.gov.br/transparenciassp/Consulta.aspx>, realizamos alguns tratamentos descritos no texto e iniciamos a análise estatística e o desenvolvimento dos testes de Aderência. Todos as análises foram feitas utilizando a linguagem de programação Python juntamente com os pacotes científicos como Pandas, Numpy, Folium, entre outros.

Vale destacar aqui a diferença entre furto e roubo: furto é a diminuição do patrimônio de outra pessoa, sem que haja violência e o roubo é a subtração do patrimônio com ameaça ou violência.

No Capítulo I há um resumo dos principais conceitos da teoria da probabilidade e estatística e, em seguida, no Capítulo II, passamos ao tratamento do banco de dados, explicitando a metodologia, e os resultados encontram-se no Capítulo III.

I CONCEITOS BÁSICOS DA TEORIA DA PROBABILIDADE

1.1 TÓPICOS DE PROBABILIDADE E ESTATÍSTICA

Neste capítulo não temos a intenção de expor minuciosamente a teoria utilizada no trabalho, mas apenas um resumo para o leitor.

1.1.1 DISTRIBUIÇÃO DE POISSON

Na área de probabilidade e estatística, a distribuição de *Poisson* é uma distribuição discreta de probabilidade que indica a probabilidade de um número de eventos do mesmo tipo e independentes entre si, ocorrer em um período de tempo.

Para uma variável aleatória X que segue uma distribuição de *Poisson* com parâmetro λ , $\lambda > 0$, $X \sim Po(\lambda)$, temos que a função de probabilidade é dada por:

$$P(X = x) = \frac{e^{-\lambda} \lambda^x}{x!}, x = 0, 1, 2, \dots \quad (1.1)$$

Para a distribuição de *Poisson*, temos que:

- Esperança de X : $E(X) = \lambda$;
- Variância de X : $V(X) = \lambda$;
- Função Geradora de Momentos de X : $M_X(t) = e^{\lambda(e^t - 1)}$, $t \in \mathbb{R}$;
- Função Característica: $\varphi(t) = e^{\lambda(e^t - 1)}$, $t \in \mathbb{R}$.

1.1.2 LEI FORTE DOS GRANDES NÚMEROS

Teorema 1. (ROSS, 2019) *Sejam X_1, X_2, \dots, X_n variáveis aleatórias independentes e identicamente distribuídas, cada uma com média finita $\mu = E(X_i)$. Então:*

$$P\left(\lim_{n \rightarrow \infty} \frac{(X_1 + X_2 + \dots + X_n)}{n} = \mu\right) = 1. \quad (1.2)$$

1.2 TESTE DE HIPÓTESE

Os testes de hipóteses são um conjunto de métodos estatísticos utilizados para tomar uma decisão (aceitar ou rejeitar a hipótese nula H_0) utilizando as hipóteses nula (H_0) ou alternativa (H_1). Na teoria dos testes de hipóteses temos ainda dois tipos de erros:

- Erro do tipo I: é a probabilidade de se rejeitar a hipótese nula quando ela é verdadeira.
- Erro do tipo II: é a probabilidade de se rejeitar a hipótese alternativa quando ela é verdadeira.

A seguir, abordaremos o método do p-valor ou nível descritivo, método este usado neste trabalho.

1.2.0.1 DECISÃO DO TESTE BASEADA NO NÍVEL DESCRIPTIVO OU P-VALOR

O p-valor é definido como a probabilidade de obter-se uma estatística do teste igual ou mais extrema (maior ou menor dependendo da hipótese alternativa) do que a estatística observada a partir de uma amostra ou de uma população assumindo-se a hipótese nula como verdadeira. Se o p-valor for menor que o nível de significância determinado, então não encontramos evidências para aceitar a hipótese nula.

1.2.0.2 TESTES DE ADERÊNCIA

Os testes de aderência, são muito utilizados para testar a aderência de um modelo probabilístico (como por exemplo distribuição Normal, Exponencial, *Poisson*, etc) a um conjunto de dados observado.

A metodologia geral dos testes de aderência consiste em calcular a proximidade entre os dados observados (que chamamos de O) e os dados esperados sob a hipótese nula (que chamamos de E). Os dados esperados normalmente são obtidos através de geradores de números aleatórios, seguindo a distribuição de probabilidade desejada. Normalmente, em um teste de aderência, as hipóteses H_0 e H_1 são:

- H_0 : a população tem uma distribuição especificada.
- H_1 : a população não tem a distribuição especificada.

1.2.0.3 TESTE QUI-QUADRADO PARA ADERÊNCIA

Este teste é utilizado para verificar a qualidade do ajuste, ao comparar a distribuição das frequências observadas com as frequências esperadas. Neste trabalho, por exemplo, teremos a geração da seguinte tabela de frequências, na qual realizaremos um teste Qui-Quadrado para Aderência para cada dia da semana de 2018:

Categoría	Freq. Observada	Freq. Esperada
Segunda	O_1	E_1
\vdots	\vdots	\vdots
Segunda	O_{53}	E_{53}
Terça	O_1	E_1
\vdots	\vdots	\vdots
Terça	O_{52}	E_{52}
\vdots	\vdots	\vdots
\vdots	\vdots	\vdots
Domingo	O_1	E_1
\vdots	\vdots	\vdots
Domingo	O_{52}	E_{52}

Tabela 1.1 – Exemplo de tabela de frequências usada no teste Qui-Quadrado

A estatística do teste Qui-Quadrado é a seguinte:

$$\chi^2 = \sum_{i=1}^s \frac{(O_i - E_i)^2}{E_i}, \quad (1.3)$$

em que s é o número de observações, no nosso caso 53 para as segundas-feiras e 52 para o restante dos dias da semana.

II METODOLOGIA

2.1 CRIAÇÃO DE BASE DE DADOS DA SSP

Para a criação da base de dados de B.O. de furtos e roubos da SSP efetuamos os seguintes passos:

- Download dos arquivos em Excel no site <http://www.ssp.sp.gov.br/transparenciassp/Consulta.aspx> através do *scrapper* no Apêndice A (pode ser utilizado para baixar outros dados também);
- Como nós temos um arquivo de Excel por mês para B.O. de furto e roubo, concatenamos cada arquivo para formar um arquivo completo de furtos e outro de roubos, respectivamente furtosConcat.csv e roubosConcat.csv como está no Apêndice B;
- Com as bases criadas no passo anterior, mantivemos apenas as informações do ano de 2018 e que são da cidade de São Paulo e adicionamos a variável que indica o dia da semana da ocorrência, que chamamos de 'DIASEMANA'. Como resultado, teremos duas bases: furtos_2018.csv e roubos_2018.csv.

O respectivo código encontra-se no Apêndice C.

2.2 CRIAÇÃO DOS MAPAS DE CALOR/MARCADORES COM OS DADOS DA SSP

Uma forma útil de analisar os focos de furtos e de roubos em uma determinada região, é a geração de mapas de calor, onde as "regiões de calor" com tons mais avermelhados representam os locais que tiveram maior ocorrência de furtos/-roubos e as "regiões de calor" com tons mais azulados representam os locais que tiveram menor ocorrência de furtos/roubos.

Uma outra abordagem é a geração de mapas de marcadores, onde contamos quantos furtos/roubos ocorrem em um mesmo local e colocamos um marcador com a respectiva quantidade de ocorrências.

Para a criação dos mapas de calor, efetuamos os seguintes passos, tanto para furtos, tanto para roubos:

- Com as bases descritas na seção anterior, removemos as linhas que não contém valores nos campos de latitude, longitude, bairro ou cidade;
- Utilizando o pacote Folium, iniciamos um mapa com as coordenadas centrais da cidade de São Paulo: [-23.5475, -46.63611];
- A fim de que o Folium possa identificar corretamente os valores de geolocalização, atribuímos às colunas de latitude e de longitude a uma estrutura de lista;
- Adicionamos essa lista a um mapa de calor e o salvamos no formato HTML.

Para a geração dos mapas de marcadores, efetuamos os seguintes passos, tanto para furtos, quanto para roubos:

- Com as bases descritas na seção anterior, removemos as linhas que não contêm valores nos campos de latitude, longitude, bairro ou cidade;
- Utilizando o pacote Folium, iniciamos um mapa com as coordenadas centrais da cidade de São Paulo: [-23.5475, -46.63611];
- A fim de que o Folium possa identificar corretamente os valores de geolocalização, atribuímos às colunas de latitude e de longitude a uma estrutura de lista;
- Adicionamos cada elemento da lista ao marcador e automaticamente o pacote Folium agrupa e soma a quantidade de furtos e roubos dependendo do *zoom* que é escolhido¹.

¹Para agrupamentos menores, dependendo do *zoom* escolhido, alguns focos poderão ser agrupados com outros e centralizados em uma região de acordo com a latitude e longitude. Assim, a comparação de focos menores para o mapa de calor e mapa de marcadores pode ter algumas diferenças.

- Salvamos o mapa de marcador no formato HTML.

O código deste item encontra-se no Apêndice D.

2.3 GERAÇÃO DE ANÁLISES DESCRIPTIVAS E GRÁFICOS

Para verificar a distribuição dos dados, geramos um *boxplot* da quantidade de furtos de carros por dia da semana, um histograma para cada dia da semana mostrando a distribuição dos furtos, medidas descritivas da quantidade de furtos por dia da semana, um *boxplot* da quantidade de furtos de carros por período do dia e por último medidas descritivas para a quantidade de furtos por período do dia. Todas essas análises também foram feitas para o número de roubos. Todo o processo consta no Apêndice E.

2.4 CHECAGEM PARA TESTAR SE AS OCORRÊNCIAS DE B.O.'S SE APROXIMAM DE UMA DISTRIBUIÇÃO DE POISSON

Como nosso objetivo é ver se a distribuição da ocorrência dos B.O.'s de furto e roubo de carros na cidade de São Paulo em 2018 aproxima-se de uma *Poisson* e assim afirmar que a ocorrência de sinistros de seguro de carros (subconjunto do conjunto de carros com B.O. por furto ou roubo) também se aproxima de uma distribuição de *Poisson*, realizamos as seguintes etapas:

- Agrupamos os dados de B.O. relativos ao ano de 2018 por dia da semana e contamos quantos B.O.'s foram abertos em cada dia da semana em 2018;
- Tendo isso, para obtermos o parâmetro λ da nossa *Poisson*, dividimos estes valores pela quantidade dos respectivos dias da semana em 2018 (53 para segunda-feira e 52 para os outros dias da semana);
- Após isso, simulamos 53 valores aleatórios de *Poisson* para a segunda-feira e 52 para cada um dos outros dias da semana usando a $seed = 42$ para permitir a replicabilidade da simulação;

- Tendo assim os valores observados (número de boletins de ocorrência para cada dia da semana do ano) e os valores esperados (número simulado através do método aleatório `numpy.random.poisson(λ, QTD)`) realizamos um teste Qui-Quadrado de aderência para cada dia da semana.

O respectivo código encontra-se no Apêndice F.

III RESULTADOS

3.1 MAPAS DE CALOR

Na presente seção, mostraremos imagens estáticas dos mapas de calor. Para tal, escolhemos, a título de ilustração, as regiões da Avenida Paulista e da Cidade Universitária Armando de Salles Oliveira.

A seguir analisaremos os mapas de calor para furtos e em seguida, para roubos.

3.1.1 MAPAS DE CALOR PARA FURTOS

Para furtos, mostramos abaixo os mapas de calor da Avenida Paulista e da Cidade Universitária Armando de Salles Oliveira.



Figura 3.1 – Mapa de calor da ocorrência de furtos em 2018 na Avenida Paulista.

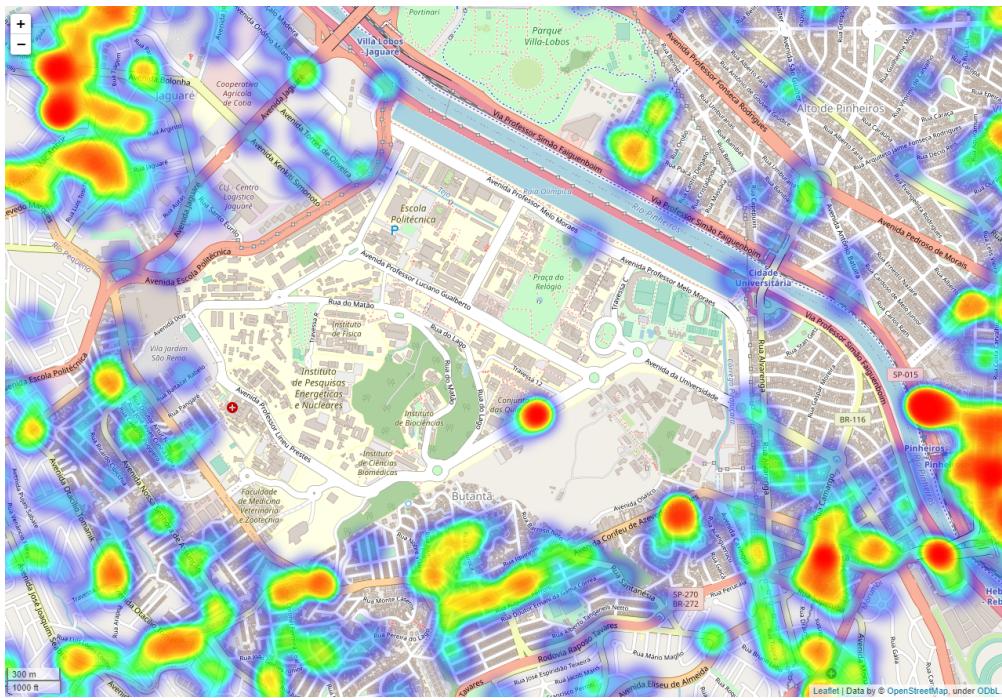


Figura 3.2 – Mapa de calor da ocorrência de furtos em 2018 na Cidade Universitária Armando de Salles Oliveira.

Vemos com os dois mapas acima que temos mais focos de furtos na região da Avenida Paulista do que na Cidade Universitária Armando de Salles Oliveira.

Na região da Avenida Paulista, vemos que os focos de furtos concentram-se na região do Museu de Arte de São Paulo Assis Chateaubriand e nas ruas paralelas, principalmente na direção do centro da cidade de São Paulo. Vemos também que existe grandes focos de furtos no entorno da região da Avenida 23 de Maio, que pertence ao bairro do Paraíso.

Já na região da Cidade Universitária Armando de Salles Oliveira, temos quatro focos principais de furtos: Portão 1, Avenida Lineu Prestes próximo ao Instituto de Química, Portão 3 e Avenida Professor Melo Moraes próximo à Praça do Relógio. Além destes focos, temos grandes focos de furtos no entorno da cidade universitária, principalmente na região da Vila Indiana e do Jaguaré.

3.1.2 MAPAS DE CALOR PARA ROUBOS

Para roubos, como anteriormente, mostramos a seguir os mapas de calor da Avenida Paulista e da Cidade Universitária Armando de Salles Oliveira.

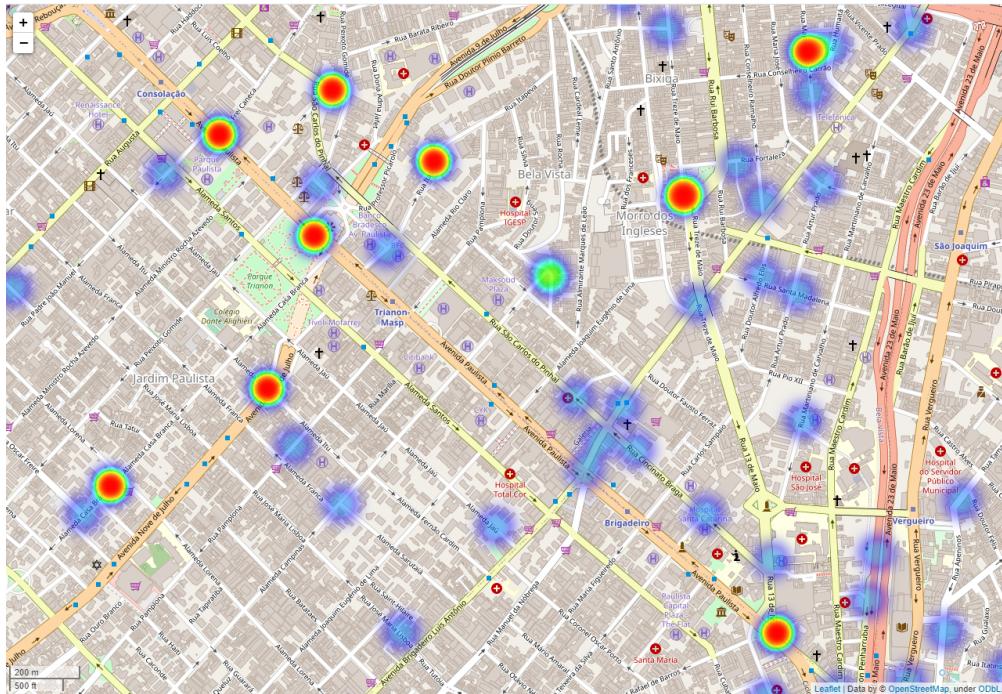


Figura 3.3 – Mapa de calor da ocorrência de roubos em 2018 na Avenida Paulista.



Figura 3.4 – Mapa de calor da ocorrência de roubos em 2018 na Cidade Universitária Armando de Salles Oliveira.

Vemos com os dois mapas acima que temos mais focos de roubos na região da Avenida Paulista do que na Cidade Universitária Armando de Salles Oliveira.

Na região da Avenida Paulista, vemos que os focos de roubos concentram-se na região do Museu de arte de São Paulo Assis Chateaubriand e nas ruas paralelas, principalmente na direção do centro da cidade de São Paulo. Vemos também que existe grandes focos de roubos no entorno da região da Avenida 23 de Maio, que pertence ao bairro do Paraíso.

Já na região da Cidade Universitária Armando de Salles Oliveira, temos quatro focos principais de roubos em ordem de concentração: Portão 1, Avenida Lineu Prestes próximo ao Instituto de Química, Portão 3 e Avenida Professor Melo Moraes próximo á Praça do Relógio. Além destes focos, temos grandes focos de roubos no entorno da cidade universitária, principalmente na região da Vila Indiana e do Jaguaré.

Outro fato que se evidencia com os mapas de calor para furtos e roubos é que vemos focos menores de roubos do que de furtos. Principalmente na região da

Avenida Paulista.

3.2 MAPAS DE MARCADORES

Para acompanhar e refinar a análise feita através dos mapas de calor, mostraremos imagens estáticas de mapas de marcadores. Aqui, escolhemos novamente as regiões da Avenida Paulista e da Cidade Universitária Armando de Salles Oliveira para a ilustração.

A seguir analisaremos os mapas de marcadores para furtos e em seguida, para roubos.

3.2.1 MAPAS DE MARCADORES PARA FURTOS

Para furtos, mostramos abaixo os mapas de marcadores da Avenida Paulista e da Cidade Universitária Armando de Salles Oliveira.

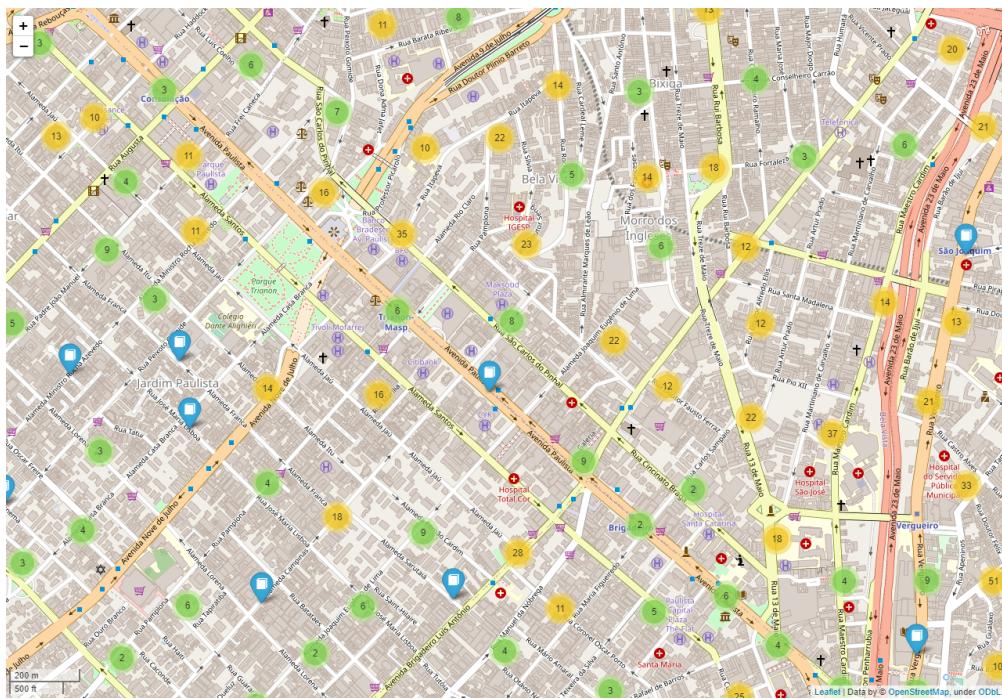


Figura 3.5 – Mapa de marcadores da ocorrência de furtos em 2018 na Avenida Paulista.

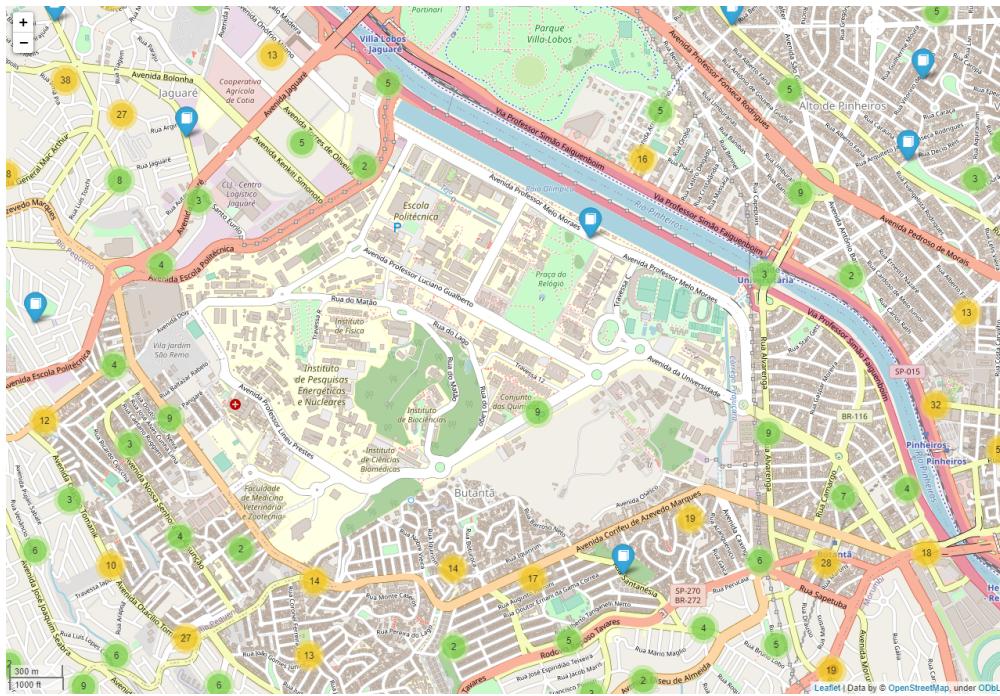


Figura 3.6 – Mapa de marcadores da ocorrência de furtos em 2018 na Cidade Universitária Armando de Salles Oliveira.

Aqui podemos quantificar melhor os focos de furtos abordados na seção anterior. Vemos, por exemplo, que o foco de furto no Portão 1 é de mesma magnitude que o foco na Avenida Lineu Prestes próximo ao Instituto de Química. Vemos também que os focos no Portão 3 encontram-se mais espalhados e descentralizados, principalmente na vizinhança externa à universidade.

3.2.2 MAPAS DE MARCADORES PARA ROUBOS

Para roubos, mostramos abaixo os mapas de marcadores da Avenida Paulista e da Cidade Universitária Armando de Salles Oliveira.

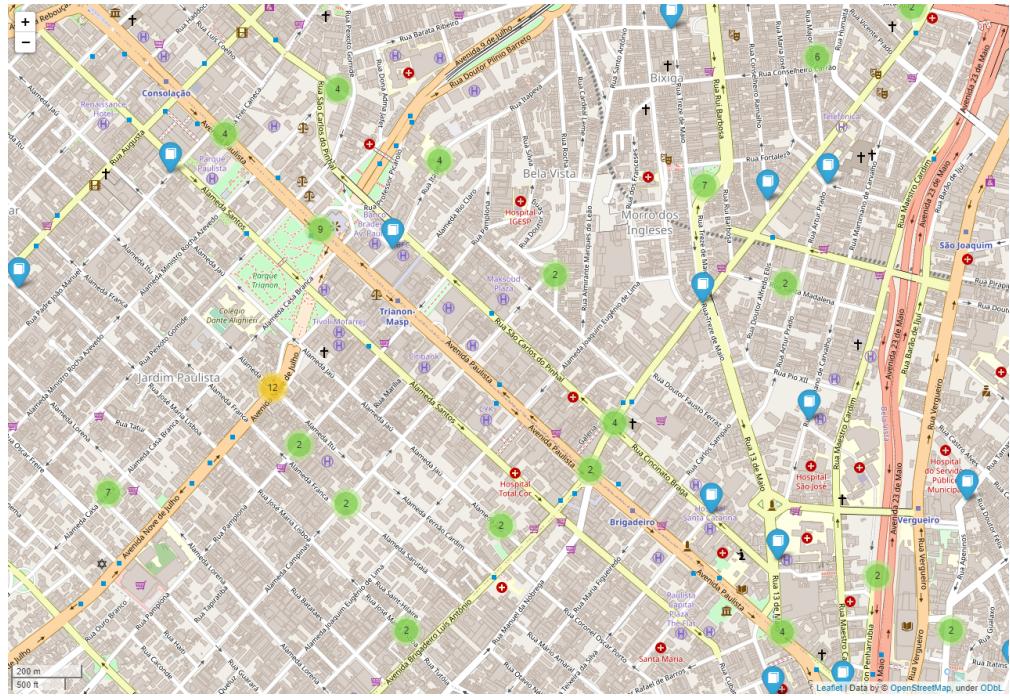


Figura 3.7 – Mapa de marcadores da ocorrência de roubos em 2018 na Avenida Paulista.

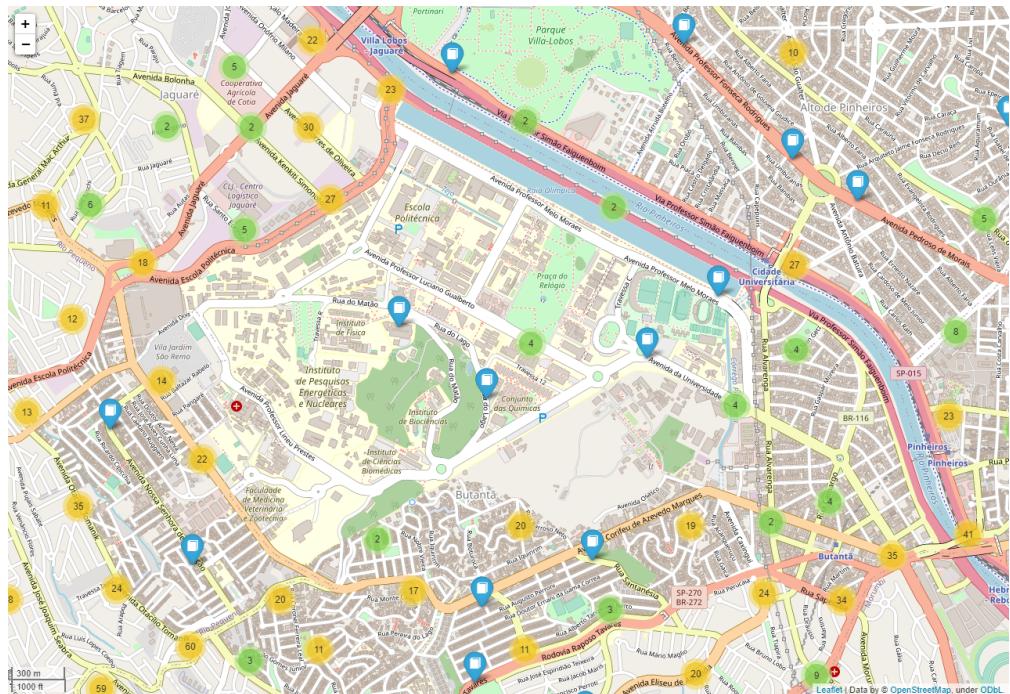


Figura 3.8 – Mapa de marcadores da ocorrência de roubos em 2018 na Cidade Universitária Armando de Salles Oliveira.

Vemos com o mapa da região da Avenida Paulista, que temos focos pequenos de roubos na região com excessão do foco que fica na Avenida Nove de Julho próximo a Alameda Itu.

Para o mapa da região da Cidade Universitária Armando de Salles Oliveira vemos grandes focos de roubos na região da Vila Indiana e do Jaguare.

3.3 MEDIDAS DESCRIPTIVAS

3.3.1 RESULTADOS PARA FURTOS E ROUBOS POR DIA DA SEMANA

Inicialmente, apresentamos medidas descritivas para a quantidade de furtos e roubos por dia da semana.

DIASEMANA	count	mean	std	min	25%	50%	75%	max
Segunda	53.0	136.52830188679246	35.904040791224055	47.0	115.0	139.0	156.0	225.0
Terça	52.0	154.55769230769232	40.71874524316754	60.0	122.5	169.0	187.25	228.0
Quarta	52.0	173.6153846153846	41.96971543356781	51.0	150.5	188.0	197.0	234.0
Quinta	52.0	149.17307692307693	40.613781626257291	48.0	131.5	161.0	176.25	204.0
Sexta	52.0	139.15384615384616	28.713175642991477	71.0	119.75	139.0	157.75	194.0
Sábado	52.0	121.0	39.05953224020392	62.0	99.0	110.0	131.25	230.0
Domingo	52.0	125.40384615384616	39.17495178889365	39.0	100.5	120.5	143.5	207.0

Tabela 3.2 – Estatísticas para a quantidade de furtos por dia da semana.

DIASEMANA	count	mean	std	min	25%	50%	75%	max
Segunda	53.0	160.43396226415095	46.162967773031376	95.0	133.0	154.0	178.0	368.0
Terça	52.0	175.0	37.04263574241307	105.0	153.25	175.0	190.5	250.0
Quarta	52.0	173.1153846153846	31.29659121082994	106.0	153.5	173.0	195.25	256.0
Quinta	52.0	167.6153846153846	44.087325930388516	51.0	139.25	170.5	193.25	315.0
Sexta	52.0	179.25	45.73447448985005	84.0	149.75	172.0	212.5	263.0
Sábado	52.0	183.78846153846155	37.96610280622188	114.0	156.5	185.0	204.25	274.0
Domingo	52.0	174.5	39.44094617645624	108.0	147.0	165.0	203.0	263.0

Tabela 3.3 – Estatísticas para a quantidade de roubos por dia da semana.

Vemos que em 2018 ocorreu, em média, mais roubos por dia da semana do que furtos. Além disso, o número mínimo de roubos por dia da semana também é maior do que o número mínimo de furtos. O mesmo ocorre com o número máximo

de roubos e furtos. Estes resultados tornam-se mais relevantes nos gráficos *boxplot* da quantidade de furtos e roubos por dia da semana, que segue abaixo:

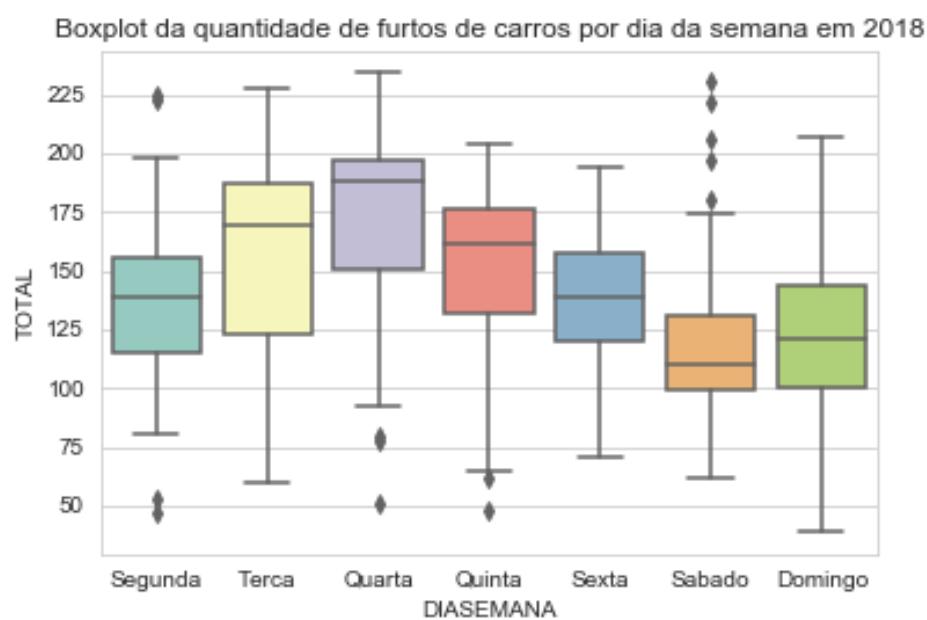


Figura 3.9 – Gráfico de *boxplot* para a quantidade de furtos por dia da semana.

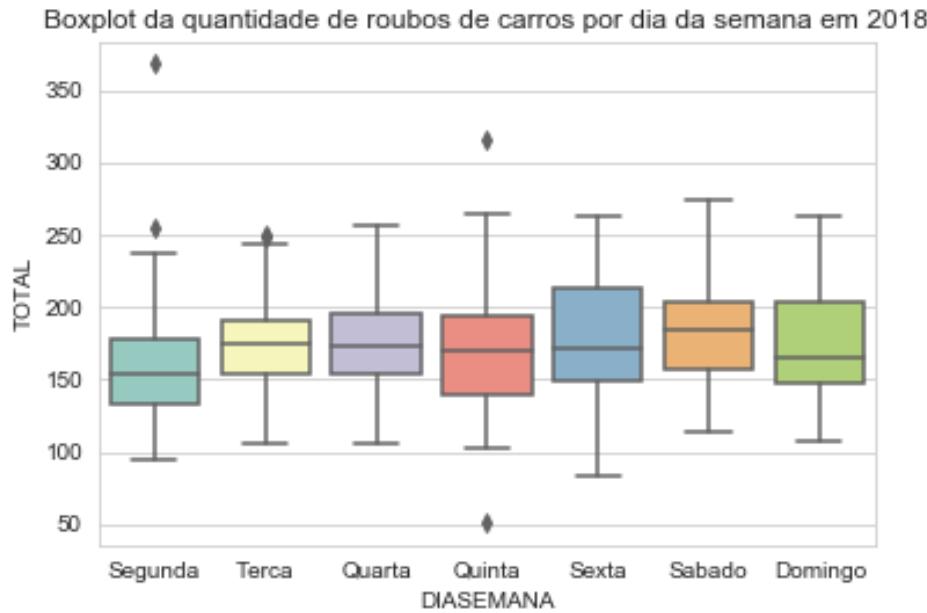


Figura 3.10 – Gráfico de *boxplot* para a quantidade de roubos por dia da semana.

Em uma primeira vista, vemos que para furtos, temos uma tendência de aumento de ocorrências na terça e na quarta, decaindo nos outros dias da semana, vemos também que temos mais *outliers* em furtos do que em roubos.

3.3.2 RESULTADOS PARA FURTOS E ROUBOS POR PERÍODO DO DIA

Como na seção anterior, realizamos também a análise descritiva para a quantidade de furtos e roubos por período do dia.

PERÍODO_OCORRÊNCIA	count	mean	std	min	25%	50%	75%	max
A NOITE	365.0	35.66575342465753	13.220366189776318	6.0	26.0	35.0	44.0	83.0
A TARDE	365.0	36.11232876712329	13.862194072620683	7.0	25.0	35.0	46.0	79.0
DE MADRUGADA	365.0	18.53698630136986	7.999442062434814	2.0	13.0	18.0	23.0	48.0
PELA MANHÃ	365.0	44.23835616438356	19.919636827922215	5.0	27.0	41.0	61.0	94.0

Tabela 3.4 – Estatísticas para a quantidade de furtos por período do dia.

PERIODO_OCORRENCIA	count	mean	std	min	25%	50%	75%	max
A NOITE	365.0	75.33424657534246	23.636860178857486	18.0	60.0	72.0	80.0	198.0
A TARDE	365.0	35.30684931506849	16.80296553469651	5.0	23.0	34.0	43.0	112.0
DE MADRUGADA	365.0	28.46301369863014	18.287594928128367	3.0	15.0	24.0	38.0	122.0
PELA MANHÃ	365.0	34.06027397260274	15.582523492241918	1.0	23.0	32.0	42.0	118.0

Tabela 3.5 – Estatísticas para a quantidade de roubos por período do dia.

Vemos, inicialmente, que em média, ocorre mais roubos do que furtos de noite e de madrugada. Vemos também que a mediana acompanha o crescimento da média. Além disso, vemos claramente que, em todos os períodos do dia, temos mais roubos do que furtos.

Agora veremos os gráficos *boxplot* para para a quantidade de furtos e roubos por período do dia.

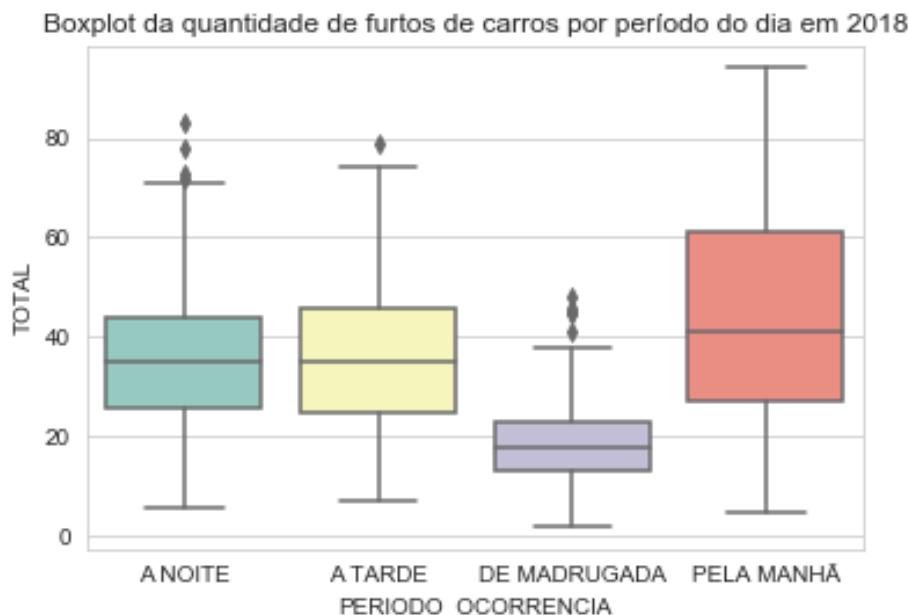


Figura 3.11 – Gráficos *boxplot* para a quantidade de furtos por período do dia.

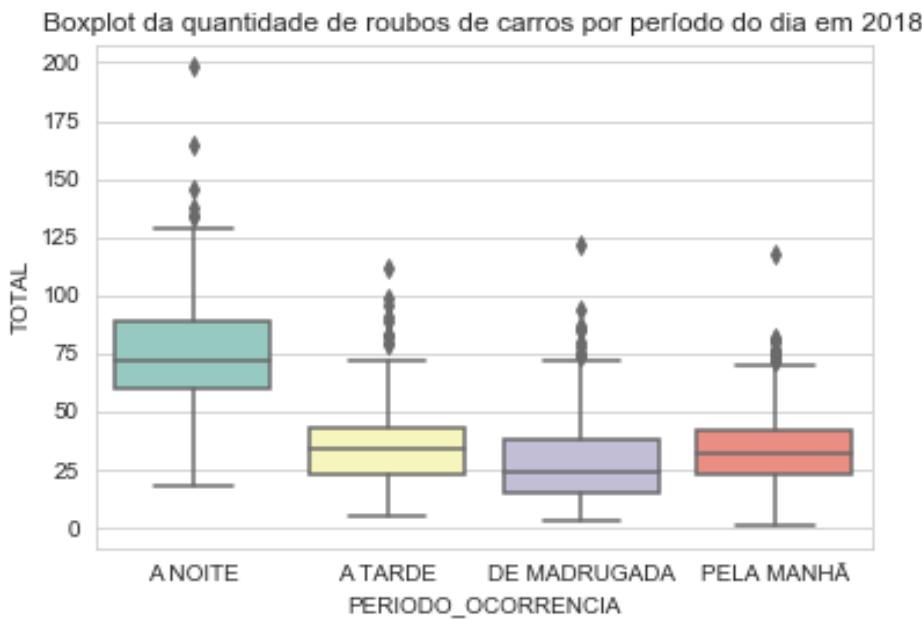


Figura 3.12 – Gráficos *boxplot* para a quantidade de roubos por período do dia.

Vemos que, no geral, os dados para roubos têm menor dispersão, porém têm mais *outliers*. Vemos também que para furtos, temos uma maior dispersão no período da manhã e para roubos temos uma maior dispersão no período da noite.

3.4 TESTES DE ADERÊNCIA E RESULTADOS

Nesta seção, realizamos um teste de Qui-Quadrado para Aderência para testar a hipótese nula de que a distribuição da quantidade de furtos e roubos por dia da semana em 2018 se aproxima de uma distribuição de *Poisson*.

3.4.1 RESULTADO DO TESTE DE ADERÊNCIA PARA FURTOS

Para a base de furtos de 2018, realizamos 7 testes de Aderência, um para cada dia da semana, querendo testar se a distribuição da quantidade de furtos por dia da semana se aproxima de uma distribuição de *Poisson*. Logo, para cada dia da semana temos:

- H_0 : a distribuição da quantidade de furtos para cada dia da semana durante o ano de 2018 se aproxima de uma distribuição de *Poisson*.
- H_1 : a distribuição da quantidade de furtos para cada dia da semana durante o ano de 2018 não se aproxima de uma distribuição de *Poisson*.

Abaixo vemos os resultados dos 7 testes:

DIASEMANA	TOTAL	QTD	LAMBDA	chi_square	p_value
Segunda	7236	53	136.52830188679246	554.8785406231481	2.7482145688117356e-85
Terça	8037	52	154.55769230769232	631.7751479072313	3.942301277145255e-101
Quarta	9028	52	173.6153846153846	522.5711423651504	1.9838739067033342e-79
Quinta	7757	52	149.17307692307693	592.9091220292021	2.3025501526354616e-93
Sexta	7236	52	139.15384615384616	354.63131922426317	4.588993741895699e-47
Sábado	6292	52	121.0	717.5878360748935	2.0540130078375486e-118
Domingo	6521	52	125.40384615384616	682.4224769701956	2.60413582167709e-111

Tabela 3.6 – Tabela de resultados do teste de Aderência Qui-Quadrado para a quantidade de furtos por dia da semana.

A coluna "TOTAL" refere-se à quantidade de furtos ocorridos no dia da semana e "QTD" é a quantidade de dias para cada dia da semana em 2018. Assim, fazendo TOTAL/QTD, obtemos a coluna "LAMBDA" que é o parâmetro usado para simular os QTD valores esperados da distribuição de *Poisson* e assim realizar o teste de Aderência.

A coluna "chi_square" é a estatística do teste Qui-Quadrado para Aderência e em seguida vemos a coluna do p-valor.

Para todos os testes realizados, vemos que o nosso p-valor é demasiadamente pequeno, ou seja, há evidências para rejeitar H_0 .

3.4.2 RESULTADO DO TESTE DE ADERÊNCIA PARA ROUBOS

Para a base de roubos de 2018, também realizamos 7 testes de Aderência, um para cada dia da semana, querendo testar se a distribuição da quantidade de roubos por dia da semana se aproxima de uma distribuição de *Poisson*. Logo, igual ao caso anterior, temos para cada dia da semana:

- H_0 : a distribuição da quantidade de roubos para cada dia da semana durante o ano de 2018 se aproxima de uma distribuição de *Poisson*.
- H_1 : a distribuição da quantidade de roubos para cada dia da semana durante o ano de 2018 não se aproxima de uma distribuição de *Poisson*.

Abaixo vemos os resultados dos 7 testes:

DIASEMANA	TOTAL	QTD	LAMBDA	chi_square	p_value
Segunda	8503	53	160.43396226415095	884.1669142848176	9.501105571372304e-152
Terça	9100	52	175.0	417.5142780584904	5.414300490177047e-59
Quarta	9002	52	173.1153846153846	313.57130031960685	1.89475958983527e-39
Quinta	8716	52	167.6153846153846	665.3216697148953	7.242436505683686e-108
Sexta	9321	52	179.25	699.5217058382061	9.228287721287365e-115
Sábado	9557	52	183.78846153846155	441.3153003612102	1.419225582544405e-63
Domingo	9074	52	174.5	534.3408105975193	9.501014169259076e-82

Tabela 3.7 – Tabela de resultados do teste de Aderência Qui-Quadrado para a quantidade de roubos por dia da semana.

Igualmente à seção anterior, a coluna "TOTAL" refere-se a quantidade de roubos ocorridos no dia da semana e "QTD" é a quantidade de dias para cada dia da semana em 2018. Assim, fazendo TOTAL/QTD, obtemos a coluna "LAMBDA" que é o parâmetro usado para simular os QTD valores esperados da distribuição de *Poisson* e assim realizar o teste de Aderência.

A coluna "chi_square" é a estatística do teste Qui-Quadrado para Aderência e em seguida vemos a coluna do p-valor.

Para todos os testes realizados, também vemos que o nosso p-valor é demasiadamente pequeno, ou seja, há evidências para rejeitar a hipótese H_0 .

IV CONCLUSÕES

4.1 CONCLUSÕES

No presente trabalho, observamos que a base de dados da Secretaria de Segurança Pública do Estado de São Paulo, além de ser de fácil acesso, é de fácil interpretação. Isso se mostrou de muito valor para o desenvolvimento das análises. Com isso, conseguimos rapidamente estudar e analisar os dados.

Um fato importante a ressaltar é que apesar das análises estatísticas mostrarem uma ocorrência maior de roubos do que de furtos da cidade de São Paulo, de uma forma geral os dois mapas escolhidos mostraram o contrário, indicando que existem outras regiões da cidade que possuem focos maiores de roubos.

Por fim, concluímos que na cidade de São Paulo no ano de 2018 ocorreu, em média, mais roubos do que furtos por dia da semana e mais roubos do que furtos de noite e de madrugada. Além desses resultados iniciais, vimos que a distribuição de roubos e furtos por dia da semana não se aproxima de uma *Poisson*. Este último resultado nos faz refletir sobre a utilização da distribuição de Poisson nos modelos de seguradoras, o que pode ser de interesse de estudo num futuro próximo.

REFERÊNCIAS

- FEIJÓ, Margarida B. C. *Aproximações para a probabilidade de ruína de uma companhia seguradora*. 2015. Tese (Mestrado) – Universidade do Porto.
- ROSS, Sheldon M. *A First Course in Probability*. 10 ed. [S.l.]: Pearson, 2019.

Anexos

ANEXO A – SCRAPPER PARA DOWNLOAD DOS DADOS DA SSP

```

import re
import pandas as pd
import xlrd
import requests
from bs4 import BeautifulSoup

headers = {
    'Origin': 'http://www.ssp.sp.gov.br',
    'Accept-Encoding': 'gzip,deflate',
    'Accept-Language': 'pt-BR,pt;q=0.9,en-US;q=0.8,en;q=0.7,es;q=0.6
        ',
    'Upgrade-Insecure-Requests': '1',
    'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/65.0.3325.181 Safari/537.36',
    'Content-Type': 'application/x-www-form-urlencoded',
    'Accept': 'text/html,application/xhtml+xml,application/xml;q
        =0.9,image/webp,image/apng,*/*;q=0.8',
    'Cache-Control': 'max-age=0',
    'Referer': 'http://www.ssp.sp.gov.br/transparenciassp/',
    'Connection': 'keep-alive',
}

def get_viewstate_eventvalidation(html):
    """
    Extract __VIEWSTATE and __EVENTVALIDATION
    """
    soup = BeautifulSoup(html, 'lxml')
    viewstate = soup.find('input', attrs={'id': '__VIEWSTATE'})
    viewstate_value = viewstate['value']
    eventvalidation = soup.find('input', attrs={'id': '__EVENTVALIDATION'})
    eventvalidation_value = eventvalidation['value']

    return viewstate_value, eventvalidation_value

```

```
def get_response(session, viewstate, event_validation, event_target
    ↪ , outro=None, stream=False, hdfExport=''):
    """
Handles all the responses received from every request made to
    ↪ the website.
    """
url = "http://www.ssp.sp.gov.br/transparenciassp/"
data = [
    ('__EVENTTARGET', event_target),
    ('__EVENTARGUMENT', ''),
    ('__VIEWSTATE', viewstate),
    ('__EVENTVALIDATION', event_validation),
    ('ctl00$cphBody$hdfExport', hdfExport),
]

if outro:
    data.append(('ctl00$cphBody$filterDepartamento', '0'))
    data.append(('__LASTFOCUS', ''))

response = session.post(url, headers=headers, data=data, stream=
    ↪ stream)
return response

def extract_file_name(response_headers):
    """
Tries to extract the filename returned from the response of
    ↪ the request.
    """
try:
    file_name = re.search('=.*xls', response_headers)
    file_name = file_name.group().replace('=', '')
except Exception:
    file_name = "dados.xls"

return file_name

def extract_year(information, directory, write_to_disk=True):
    """
```

*Returns a dataframe with the information from the website.
If write_to_disk is True, then a xls file is created on disk.*

```

"""
print("Extracting")
session = requests.session()

url = "http://www.ssp.sp.gov.br/transparenciassp/"

response = session.post(url, headers=headers)
viewstate, eventvalidation = get_viewstate_eventvalidation(
    ↪ response.text)

for j in range(2003, 2020):
    year = str(j)
    print("Ano:" +year)
    year = year[-2:]
    year = year.lstrip("0")
    year_value = "ctl00$cphBody$lkAno{}".format(year)

    for i in range(1, 13):
        month = str(i)
        month_value = "ctl00$cphBody$lkMes{}".format(month)
        print("Mes:" +month)

        parameters_list = [
            [information],
            [month_value, True, False],
            [year_value, True, False],
        ]
        for parameters in parameters_list:
            response = get_response(
                session, viewstate, eventvalidation, *parameters)
            html = response.text
            viewstate, eventvalidation =
                ↪ get_viewstate_eventvalidation(html)

            response = get_response(session,
                                    viewstate,
                                    eventvalidation,
                                    'ctl00$cphBody$ExportarBOLink',
                                    True,
                                    True,

```

```

        0)
file_name = extract_file_name(response.headers['content-
    ↪ disposition'])
print(file_name)
ssp_data = response.text.split('\n')
corrected_ssp_data = []
for dado in ssp_data:
    dado_corrigido = re.split('\t{1}', dado)
    corrected_ssp_data.append(dado_corrigido)

if write_to_disk:
    header = corrected_ssp_data[0]
    corrected_ssp_data = corrected_ssp_data[1:]
    df = pd.DataFrame(corrected_ssp_data)
    df.to_excel(directory + "\\\" +
                file_name, index=False, encoding='utf-8',
                ↪ header=header)

def run(directory, write_to_disk=True):
    """
    Interactive option to run the scraper.
    Choose an option, a month and a year to download the corrected
    ↪ information.
    """
    print("Opcoes:")
    print("1 - Homicidio Doloso")
    print("2 - Latrocínio")
    print("3 - Lesão Corporal Seguida de Morte")
    print("4 - Morte Decorrente de Oposição A Intervenção Policial")
    print("5 - Morte Suspeita")
    print("6 - Furto de Veículo")
    print("7 - Roubo de Veículo")
    print("8 - Furto de Celular")
    print("9 - Roubo de Celular")
    print("10 - Feminicídio")
    print("11 - Registro de Óbitos - IML")
    option = int(input("Escolha a opção:"))

informations = {
    1: "ctl00$cphBody$btnHomicicio",
    2: "ctl00$cphBody$btnLatrocinio",
    3: "ctl00$cphBody$btnLesaoMorte",
}

```

```
4: "ctl00$cphBody$btnMortePolicial",
5: "ctl00$cphBody$btnMorteSuspeita",
6: "ctl00$cphBody$btnFurtoVeiculo",
7: "ctl00$cphBody$btnRouboVeiculo",
8: "ctl00$cphBody$btnFurtoCelular",
9: "ctl00$cphBody$btnRouboCelular",
10: "ctl00$cphBody$btnFemicidio",
11: "ctl00$cphBody$btnIML"
}

information = informations[option]

return extract_year(information, directory, write_to_disk)

def main():

    directory = str(input("Digite o diretório para salvar os dados:
    ↪ "))
    run(directory, True)

if __name__ == "__main__":
    main()
```


ANEXO B – JUNÇÃO DOS DADOS DA SSP

```
[1]: import os
import csv
import pandas as pd
import numpy as np
```



```
[3]: # Empilhamos um primeiro arquivo de furtos para poder empilhar
      ↵os outros nele
df1 = pd.read_excel('D:
      ↵\\Danilo\\Desktop\\EPS\\TCC\\data_cleaned\\SSP\\Furtos de
      ↵veiculos\\DadosBO_2003_1.xls', usecols = "A:AB,AJ,AL,AU:BB", ↵
      ↵converters={'ANO_FABRICACAO':int,'ANO_MODELO':int})
```



```
[8]: # Nao necessariamente um arquivo de um dado mes tem dados
      ↵daquele mes. Por isso empilhamos tudo
for filename in os.listdir('D:
      ↵\\Danilo\\Desktop\\EPS\\TCC\\data_cleaned\\SSP\\Furtos de
      ↵veiculos'):
    if(filename != "DadosBO_2003_1.xls"):
        df = pd.read_excel('D:
      ↵\\Danilo\\Desktop\\EPS\\TCC\\data_cleaned\\SSP\\Furtos de
      ↵veiculos\\'+filename, usecols = "A:AB,AJ,AL,AU:BB", ↵
      ↵converters={'ANO_FABRICACAO':int,'ANO_MODELO':int})
        df1 = pd.concat([df1, df])
```



```
[9]: df1.to_csv('D:
      ↵\\Danilo\\Desktop\\EPS\\TCC\\data_cleaned\\SSP\\furtosConcat.
      ↵csv', sep=",", index=False)
```



```
[10]: # Empilhamos um primeiro arquivo de roubos para poder empilhar
      ↵os outros nele
df1 = pd.read_excel('D:
      ↵\\Danilo\\Desktop\\EPS\\TCC\\data_cleaned\\SSP\\Roubos de
      ↵veiculos\\DadosBO_2003_1.xls', usecols = "A:AB,AJ,AL,AU:BB", ↵
      ↵converters={'ANO_FABRICACAO':int,'ANO_MODELO':int})
```



```
[11]: # Nao necessariamente um arquivo de um dado mes tem dados
      ↵daquele mes. Por isso empilhamos tudo
for filename in os.listdir('D:
      ↵\\Danilo\\Desktop\\EPS\\TCC\\data_cleaned\\SSP\\Roubos de
      ↵veiculos'):
```

```
if(filename != "DadosBO_2003_1.xls"):
    df = pd.read_excel('D:
    ↵\\Danilo\\Desktop\\EPS\\TCC\\data_cleaned\\SSP\\Roubos de_
    ↵veiculos\\'+filename, usecols = "A:AB,AJ,AL,AU:BB",_
    ↵converters={'ANO_FABRICACAO':int,'ANO_MODELO':int})
    df1 = pd.concat([df1, df])
```

```
[12]: df1.to_csv('D:
    ↵\\Danilo\\Desktop\\EPS\\TCC\\data_cleaned\\SSP\\roubosConcat.
    ↵csv', sep=",", index=False)
```


ANEXO C – CRIAÇÃO DAS BASES DE FURTOS E ROUBOS DE 2018

```
[ ]: import pandas as pd
      import datetime as dt

[ ]: # Criação da base de 2018 para furtos
df_furtos = pd.read_csv('D:\\\\EPS\\\\TCC\\\\data_cleaned\\\\SSP\\\\furtosConcat.csv', sep=",", decimal=".")

[ ]: df_furtos.columns

[ ]: df_furtos.head()

[ ]: # Convertemos para datetime
df_furtos['DATAOCORRENCIA'] = pd.
    to_datetime(df_furtos['DATAOCORRENCIA'], errors = 'coerce')

[ ]: df_furtos.head()

[ ]: # Filtramos os dados de 2018
furtos_2018 = df_furtos[df_furtos['DATAOCORRENCIA'].dt.year == 2018]

[ ]: furtos_2018.head()

[ ]: furtos_2018['CIDADE'] = furtos_2018['CIDADE'].str.upper()

[ ]: # Dropamos outras cidades que não são SP
furtos_2018.drop(furtos_2018[(furtos_2018.CIDADE != 'S.PAULO') &
    (furtos_2018.CIDADE != 'SAO PAULO') & (furtos_2018.CIDADE != 'SÃO PAULO') & (furtos_2018.CIDADE != 'SP')].index, inplace=True)

[ ]: furtos_2018['CIDADE'].unique()

[ ]: furtos_2018['DIASEMANA'] = furtos_2018['DATAOCORRENCIA'].dt.
    dayofweek

[ ]: furtos_2018.sort_values('DIASEMANA', inplace=True)
```

```
[ ]: furtos_2018['DIASEMANA'].unique()
```

```
[ ]: furtos_2018.head()
```

```
[ ]: furtos_2018.to_csv('D:\\EPS\\TCC\\data_cleaned\\SSP\\furtos2018.
↪csv', sep=",", index=False)
```

```
[ ]: # Criação da base de 2018 para roubos
df_roubos = pd.read_csv('D:
↪\\EPS\\TCC\\data_cleaned\\SSP\\roubosConcat.csv', sep=",",
↪decimal=".")
```

```
[ ]: df_roubos.columns
```

```
[ ]: df_roubos.head()
```

```
[ ]: # Convertemos para datetime
df_roubos['DATAOCORRENCIA'] = pd.
↪to_datetime(df_roubos['DATAOCORRENCIA'], errors = 'coerce')
```

```
[ ]: df_roubos.head()
```

```
[ ]: roubos_2018 = df_roubos[df_roubos['DATAOCORRENCIA'].dt.year ==
↪2018]
```

```
[ ]: roubos_2018.head()
```

```
[ ]: roubos_2018['CIDADE'] = roubos_2018['CIDADE'].str.upper()
```

```
[ ]: # Dropamos outras cidades que não são SP
roubos_2018.drop(roubos_2018[(roubos_2018.CIDADE != 'S.PAULO') &
↪(roubos_2018.CIDADE != 'SAO PAULO') & (roubos_2018.CIDADE !=
↪'SÃO PAULO') & (roubos_2018.CIDADE != 'SP')].index, 
↪inplace=True)
```

```
[ ]: roubos_2018['CIDADE'].unique()
```

```
[ ]: roubos_2018['DIASEMANA'] = roubos_2018['DATAOCORRENCIA'].dt.
↪dayofweek
```

```
[ ]: roubos_2018.sort_values('DIASEMANA', inplace=True)

[ ]: roubos_2018['DIASEMANA'].unique()

[ ]: roubos_2018.head()

[ ]: roubos_2018.to_csv('D:\\EPS\\TCC\\data_cleaned\\SSP\\roubos_2018.
→CSV', sep=",", index=False)
```

ANEXO D – CRIAÇÃO DOS MAPAS DE CALOR/MARCADORES

```
[ ]: import pandas as pd
import folium
from folium.plugins import MarkerCluster
from folium.plugins import HeatMap
```



```
[ ]: df_furtos = pd.read_csv('D:
    ↪\\EPS\\TCC\\data_cleaned\\SSP\\furtos_2018.csv', sep=",", ↪
    decimal=".")
```



```
[ ]: df_furtos.head()
```



```
[ ]: df_furtos.dropna(subset=['BAIRRO'], inplace=True)
df_furtos.dropna(subset=['CIDADE'], inplace=True)
```



```
[ ]: df_furtos['LONGITUDE'] = df_furtos['LONGITUDE'].str.replace(',','.', ↪
    astype(float))
df_furtos.dropna(subset=['LONGITUDE'], inplace=True)
df_furtos['LATITUDE'] = df_furtos['LATITUDE'].str.replace(',', '.', ↪
    astype(float))
df_furtos.dropna(subset=['LATITUDE'], inplace=True)
```



```
[ ]: # Inicializamos o mapa com as coordenadas centrais da cidade de São Paulo
mapa = folium.Map(location=[-23.5475, -46.63611], ↪
    control_scale=True)

# Passamos as latitudes e longitudes para listas para poder adicioná-las ao mapa
lat = df_furtos['LATITUDE'].tolist()
lng = df_furtos['LONGITUDE'].tolist()
lista = list(zip(lat, lng))

# Adicionamos a lista ao mapa e salvamos no formato de HTML
HeatMap(lista).add_to(mapa)
mapa.save('Heat_SP_Furtos_2018.html')

# Aqui recriamos o mapa, mas agora com marcadores
mapa = folium.Map(location=[-23.5475, -46.63611], ↪
    control_scale=True)
mc = MarkerCluster()
```

```

# Contamos cada par de latitude e longitude e adicionamos a um
# marcador e salvamos o mapa como HTML
for val in lista:
    mc.add_child(folium.Marker([val[0], val[1]], icon=folium.
    Icon(icon='book'))).add_to(mapa)
mapa.save('Marker_SP_Furtos_2018.html')

[ ]:

[ ]: df_roubos = pd.read_csv('D:
    ↪\\EPS\\TCC\\data_cleaned\\SSP\\roubos_2018.csv', sep=",",
    ↪decimal=".")

[ ]: df_roubos.dropna(subset=['BAIRRO'], inplace=True)
df_roubos.dropna(subset=['CIDADE'], inplace=True)

[ ]: df_roubos['LONGITUDE'] = df_roubos['LONGITUDE'].str.replace(',',
    ↪'.').astype(float)
df_roubos.dropna(subset=['LONGITUDE'], inplace=True)
df_roubos['LATITUDE'] = df_roubos['LATITUDE'].str.replace(',',
    ↪').astype(float)
df_roubos.dropna(subset=['LATITUDE'], inplace=True)

[ ]: # Inicializamos o mapa com as coordenadas centrais da cidade de
    ↪São Paulo
mapa = folium.Map(location=[-23.5475, -46.63611],
    ↪control_scale=True)

# Passamos as latitudes e longitudes para listas para poder
# adicionar ao mapa
lat = df_roubos['LATITUDE'].tolist()
lng = df_roubos['LONGITUDE'].tolist()
lista = list(zip(lat, lng))

# Adicionamos a lista ao mapa e salvamos no formato de HTML
HeatMap(lista).add_to(mapa)
mapa.save('Heat_SP_Roubos_2018.html')

# Aqui recriamos o mapa, mas agora com marcadores

```

```
mapa = folium.Map(location=[-23.5475, -46.63611],  
    ↪control_scale=True)  
mc = MarkerCluster()  
  
# Contamos cada par de latitude e longitude e adicionamos a um  
# marcador e salvamos o mapa como HTML  
for val in lista:  
    mc.add_child(folium.Marker([val[0], val[1]], icon=folium.  
    ↪Icon(icon='book'))).add_to(mapa)  
mapa.save('Marker_SP_Roubos_2018.html')
```

ANEXO E – GERAÇÃO DA ANÁLISE DESCRIPTIVA DOS DADOS

5.1 CÓDIGO PARA REALIZAÇÃO DA ANÁLISE DESCRIPTIVA REFERENTE A BASE DE FURTOS

```
[ ]: import pandas as pd
from pandas.plotting import table
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import datetime as dt
import seaborn as sns
import six
```

ESTATISTICAS PARA FURTOS EM 2018

```
[ ]: df_furtos = pd.read_csv('D:
    ↪\EPS\\TCC\\data_cleaned\\SSP\\furtos2018.csv', sep=",", ↪
    ↪decimal=".")
```

```
[ ]: df_furtos.columns
```

```
[ ]: df_furtos.head()
```

PROCESSO PARA BOXPLOT E HISTOGRAMA DA QUANTIDADE DE FURTOS DE CARROS POR DIA DA SEMANA EM 2018

```
[ ]: # Agrupamos por dia da semana e data da ocorrencia e contamos
    ↪quantas datas tiveram
furtos_2018_grp = df_furtos.groupby(['DIASEMANA', ↪
    ↪'DATAOCORRENCIA'])['DATAOCORRENCIA'].count().rename('TOTAL'). ↪
    ↪reset_index()
```

```
[ ]: # Substituimos cada numero pelo dia da semana correspondente
furtos_2018_grp['DIASEMANA'] = furtos_2018_grp['DIASEMANA']. ↪
    ↪map({0: 'Segunda', 1: 'Terca', 2: 'Quarta', 3: 'Quinta', 4: ↪
    ↪'Sexta', 5: 'Sabado', 6: 'Domingo'})
```

```
[ ]: furtos_2018_grp
```

```
[ ]: for index, item in furtos_2018_grp.iterrows():
    df_day = furtos_2018_grp.loc[furtos_2018_grp['DIASEMANA'] == ↪
    ↪item['DIASEMANA']]
```

```

if((item['DIASEMANA'] == ('Domingo')) | (item['DIASEMANA'] ==
→('Sabado'))):
    ax = df_day.plot(kind='bar', x='DATAOCORRENCIA',_
→y='TOTAL', figsize=(20,5), title='Histograma da quantidade de_
→ocorrências de furtos nos '+item['DIASEMANA']+''s de 2018')
else:
    ax = df_day.plot(kind='bar', x='DATAOCORRENCIA',_
→y='TOTAL', figsize=(20,5), title='Histograma da quantidade de_
→ocorrências de furtos nas '+item['DIASEMANA']+''s de 2018')
    ax.set_xlabel("DATA DA OCORRENCIA ("+item['DIASEMANA']+"'s)")
    ax.set_ylabel("QTD DE OCORRENCIAS")
    plt.tight_layout()
    plt.savefig('hist_'+item['DIASEMANA']+'.png')

```

```
[ ]: sns.set_style("whitegrid")
ax = sns.boxplot(x="DIASEMANA", y="TOTAL", data=furtos_2018_grp,_
→palette="Set3").set_title('Boxplot da quantidade de furtos de_
→carros por dia da semana em 2018')
```

```
[ ]: ax.get_figure().savefig('boxplot_dia_da_semana_furtos_2018.png')
```

DESCRIBE DOS FURTOS EM 2018 POR DIA DA SEMANA

```
[ ]: # Agrupamos por dia da semana e data da ocorrencia e contamos
→quantas datas tiveram
furtos_2018_grp = df_furtos.groupby(['DIASEMANA',_
→'DATAOCORRENCIA'])['DATAOCORRENCIA'].count().rename('TOTAL').
→reset_index()
```

```
[ ]: furtos_2018_grp.head()
```

```
[ ]: df = furtos_2018_grp.groupby('DIASEMANA')['TOTAL'].describe().
→reset_index()
```

```
[ ]: df
```

```
[ ]: df['DIASEMANA'] = df['DIASEMANA'].map({0: 'Segunda', 1: 'Terca',_
→2: 'Quarta', 3: 'Quinta', 4: 'Sexta', 5: 'Sabado', 6:_
→'Domingo'})
```

```
[ ]: df.reset_index(drop=True, inplace=True)
```

```
[ ]: df
```

```
[ ]: def render_mpl_table(data, col_width=3.0, row_height=0.625,
   ↪font_size=14,
               header_color="#40466e",
   ↪row_colors=['#f1f1f2', 'w'], edge_color='w',
               bbox=[0, 0, 1, 1], header_columns=0,
               ax=None, **kwargs):
    if ax is None:
        size = (np.array(data.shape[::-1]) + np.array([0, 1])) *
   ↪np.array([col_width, row_height])
        fig, ax = plt.subplots(figsize=size)
        ax.axis('off')

    mpl_table = ax.table(cellText=data.values, bbox=bbox,
   ↪colLabels=data.columns, **kwargs)

    mpl_table.auto_set_font_size(False)
    mpl_table.set_fontsize(font_size)

    for k, cell in six.iteritems(mpl_table._cells):
        cell.set_edgecolor(edge_color)
        if k[0] == 0 or k[1] < header_columns:
            cell.set_text_props(weight='bold', color='w')
            cell.set_facecolor(header_color)
        else:
            cell.set_facecolor(row_colors[k[0]]%len(row_colors) )
    fig.savefig('describe_furtos_dia_da_semana.png')

    return ax

render_mpl_table(df, header_columns=0, col_width=3.0)
```

PROCESSO PARA BOXPLOT DA QUANTIDADE DE FURTOS DE CARROS PARA PERÍODOS DO DIA EM 2018

```
[ ]: df_furtos['PERIDOOCORRENCIA'].unique()

[ ]: df_furtos.info()

[ ]: df_furtos['PERIDOOCORRENCIA'].value_counts()

[ ]: df_furtos.drop(df_furtos[(df_furtos.PERIDOOCORRENCIA == 'EM HORA' →INCERTA')].index, inplace=True)

[ ]: df_furtos.rename({'PERIDOOCORRENCIA': 'PERIODO_OCORRENCIA'}, ↴axis=1, inplace=True)

[ ]: furtos_2018_grp = df_furtos.groupby(['PERIODO_OCORRENCIA', ↴'DATAOCORRENCIA'])['PERIODO_OCORRENCIA'].count(). ↴rename('TOTAL').reset_index()

[ ]: furtos_2018_grp

[ ]: sns.set_style("whitegrid")
ax = sns.boxplot(x="PERIODO_OCORRENCIA", y="TOTAL", ↴data=furtos_2018_grp, palette="Set3").set_title('Boxplot da ↴quantidade de furtos de carros por período do dia em 2018')
ax.get_figure().savefig('boxplot_periodo_do_dia_furtos_2018.png')
```

DESCRIÇÃO DOS FURTOS EM 2018 POR PERÍODO DO DIA

```
    ax=None, **kwargs):
if ax is None:
    size = (np.array(data.shape[::-1]) + np.array([0, 1])) * ↵
    ↵np.array([col_width, row_height])
    fig, ax = plt.subplots(figsize=size)
    ax.axis('off')

    mpl_table = ax.table(cellText=data.values, bbox=bbox, ↵
    ↵colLabels=data.columns, **kwargs)

    mpl_table.auto_set_font_size(False)
    mpl_table.set_fontsize(font_size)

    for k, cell in six.iteritems(mpl_table._cells):
        cell.set_edgecolor(edge_color)
        if k[0] == 0 or k[1] < header_columns:
            cell.set_text_props(weight='bold', color='w')
            cell.set_facecolor(header_color)
        else:
            cell.set_facecolor(row_colors[k[0]]%len(row_colors) ])
fig.savefig('describe_furtos_periodo_ocorrencia.png')

return ax

render_mpl_table(df, header_columns=0, col_width=3.0)
```

5.2 CÓDIGO PARA REALIZAÇÃO DA ANÁLISE DESCRIPTIVA REFERENTE A BASE DE ROUBOS

```
[ ]: import pandas as pd
from pandas.plotting import table
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import datetime as dt
import seaborn as sns
import six
```

ESTATISTICAS PARA ROUBOS EM 2018

```
[ ]: df_roubos = pd.read_csv('D:
    ↪\EPS\\TCC\\data_cleaned\\SSP\\roubos2018.csv', sep=",", ↪
    ↪decimal=".")
```

```
[ ]: df_roubos.columns
```

```
[ ]: df_roubos.head()
```

PROCESSO PARA BOXPLOT E HISTOGRAMA DA QUANTIDADE DE ROUBOS DE CARROS POR DIA DA SEMANA EM 2018

```
[ ]: # Agrupamos por dia da semana e data da ocorrencia e contamos
    ↪quantas datas tiveram
roubos_2018_grp = df_roubos.groupby(['DIASEMANA', ↪
    ↪'DATAOCORRENCIA'])['DATAOCORRENCIA'].count().rename('TOTAL').
    ↪reset_index()
```

```
[ ]: # Substituimos cada numero pelo dia da semana correspondente
roubos_2018_grp['DIASEMANA'] = roubos_2018_grp['DIASEMANA'].
    ↪map({0: 'Segunda', 1: 'Terca', 2: 'Quarta', 3: 'Quinta', 4: ↪
    ↪'Sexta', 5: 'Sabado', 6: 'Domingo'})
```

```
[ ]: roubos_2018_grp
```

```
[ ]: for index, item in roubos_2018_grp.iterrows():
    df_day = roubos_2018_grp.loc[roubos_2018_grp['DIASEMANA'] == ↪
    ↪item['DIASEMANA']]
```

```

if((item['DIASEMANA'] == ('Domingo')) | (item['DIASEMANA'] ==
→('Sabado'))):
    ax = df_day.plot(kind='bar', x='DATAOCORRENCIA',_
→y='TOTAL', figsize=(20,5), title='Histograma da quantidade de_
→ocorrências de roubos nos '+item['DIASEMANA']+''s de 2018')
else:
    ax = df_day.plot(kind='bar', x='DATAOCORRENCIA',_
→y='TOTAL', figsize=(20,5), title='Histograma da quantidade de_
→ocorrências de roubos nas '+item['DIASEMANA']+''s de 2018')
    ax.set_xlabel("DATA DA OCORRENCIA ("+item['DIASEMANA']+""s)")
    ax.set_ylabel("QTD DE OCORRENCIAS")
    plt.tight_layout()
    plt.savefig('hist_roubos_'+item['DIASEMANA']+'.png')

```

```
[ ]: sns.set_style("whitegrid")
ax = sns.boxplot(x="DIASEMANA", y="TOTAL", data=roubos_2018_grp,_
→palette="Set3").set_title('Boxplot da quantidade de roubos de_
→carros por dia da semana em 2018')
```

```
[ ]: ax.get_figure().savefig('bloxplot_dia_da_semana_roubos_2018.png')
```

DESCRIBE DOS ROUBOS EM 2018 POR DIA DA SEMANA

```
[ ]: # Agrupamos por dia da semana e data da ocorrencia e contamos
→quantas datas tiveram
roubos_2018_grp = df_roubos.groupby(['DIASEMANA',_
→'DATAOCORRENCIA'])['DATAOCORRENCIA'].count().rename('TOTAL').
→reset_index()
```

```
[ ]: roubos_2018_grp.head()
```

```
[ ]: df = roubos_2018_grp.groupby('DIASEMANA')['TOTAL'].describe().
→reset_index()
```

```
[ ]: df
```

```
[ ]: df['DIASEMANA'] = df['DIASEMANA'].map({0: 'Segunda', 1: 'Terca',_
→2: 'Quarta', 3: 'Quinta', 4: 'Sexta', 5: 'Sabado', 6:_
→'Domingo'})
```

```
[ ]: df.reset_index(drop=True, inplace=True)
```

```
[ ]: df
```

```
[ ]: def render_mpl_table(data, col_width=3.0, row_height=0.625,
   ↪font_size=14,
               header_color="#40466e",
   ↪row_colors=['#f1f1f2', 'w'], edge_color='w',
               bbox=[0, 0, 1, 1], header_columns=0,
               ax=None, **kwargs):
    if ax is None:
        size = (np.array(data.shape[::-1]) + np.array([0, 1])) *
   ↪np.array([col_width, row_height])
        fig, ax = plt.subplots(figsize=size)
        ax.axis('off')

    mpl_table = ax.table(cellText=data.values, bbox=bbox,
   ↪colLabels=data.columns, **kwargs)

    mpl_table.auto_set_font_size(False)
    mpl_table.set_fontsize(font_size)

    for k, cell in six.iteritems(mpl_table._cells):
        cell.set_edgecolor(edge_color)
        if k[0] == 0 or k[1] < header_columns:
            cell.set_text_props(weight='bold', color='w')
            cell.set_facecolor(header_color)
        else:
            cell.set_facecolor(row_colors[k[0]]%len(row_colors) )
    fig.savefig('describe_roubos_dia_da_semana.png')

    return ax

render_mpl_table(df, header_columns=0, col_width=3.0)
```

PROCESSO PARA BOXPLOT DA QUANTIDADE DE ROUBOS DE CARROS PARA PERÍODOS DO DIA EM 2018

```
[ ]: df_roubos['PERÍODOOCORRÊNCIA'].unique()
```

```
[ ]: df_roubos.info()
```

```
[ ]: df_roubos['PERIDOOCORRENCIA'].value_counts()
```

```
[ ]: df_roubos.drop(df_roubos[(df_roubos.PERIDOOCORRENCIA == 'EM HORA  
↓INCERTA')].index, inplace=True)
```

```
[ ]: df_roubos.rename({'PERIDOOCORRENCIA': 'PERIODO_OCORRENCIA'},  
                     axis=1, inplace=True)
```

```
[ ]: roubos_2018_grp = df_roubos.groupby(['PERIODO_OCORRENCIA',  
    ↴'DATAOCORRENCIA'])['PERIODO_OCORRENCIA'].count().  
    ↴rename('TOTAL').reset_index()
```

[]: roubos_2018_grp

```
[ ]: sns.set_style("whitegrid")
ax = sns.boxplot(x="PERIODO_OCORRENCIA", y="TOTAL",
                  data=roubos_2018_grp, palette="Set3").set_title('Boxplot da'
                                                               'quantidade de roubos de carros por período do dia em 2018')
ax.get_figure().savefig('boxplot_periodo_do_dia_roubos_2018.png')
```

DESCRIÇÃO DOS ROUBOS EM 2018 POR PERÍODO DO DIA

```
[ ]: df = roubos_2018_grp.groupby('PERIODO_OCORRENCIA')['TOTAL'].  
     ↵describe().reset_index()
```

```
[ ] : df
```

```
[ ]: def render_mpl_table(data, col_width=3.0, row_height=0.625, font_size=14,
                           header_color='#40466e', row_colors=['#f1f1f2', 'w'],
                           edge_color='w', bbox=[0, 0, 1, 1], header_columns=0,
```

```
    ax=None, **kwargs):
if ax is None:
    size = (np.array(data.shape[::-1]) + np.array([0, 1])) * ↵
    ↵np.array([col_width, row_height])
    fig, ax = plt.subplots(figsize=size)
    ax.axis('off')

    mpl_table = ax.table(cellText=data.values, bbox=bbox, ↵
    ↵colLabels=data.columns, **kwargs)

    mpl_table.auto_set_font_size(False)
    mpl_table.set_fontsize(font_size)

    for k, cell in six.iteritems(mpl_table._cells):
        cell.set_edgecolor(edge_color)
        if k[0] == 0 or k[1] < header_columns:
            cell.set_text_props(weight='bold', color='w')
            cell.set_facecolor(header_color)
        else:
            cell.set_facecolor(row_colors[k[0]]%len(row_colors) ])
fig.savefig('describe_roubos_periodo_ocorrencia.png')

return ax

render_mpl_table(df, header_columns=0, col_width=3.0)
```

ANEXO F – REALIZAÇÃO DO TESTE DE HIPÓTESE PARA FURTOS E ROUBOS

6.1 REALIZAÇÃO DO TESTE DE HIPÓTESE PARA OS DADOS DE FURTOS

```
[ ]: import pandas as pd
from pandas.api.types import CategoricalDtype
from pandas.plotting import table
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import datetime as dt
import seaborn as sns
import six
from scipy.stats import chisquare

[ ]: df_furtos = pd.read_csv('D:\\\\EPS\\\\TCC\\\\data_cleaned\\\\SSP\\\\furtos2018.csv', sep=",", decimal=".")

[ ]: furtos_2018_grp = df_furtos.groupby(['DIASEMANA', 'DATAOCORRENCIA'])['DATAOCORRENCIA'].count().rename('TOTAL').reset_index()

[ ]: furtos_2018_grp

[ ]: furtos_2018_grp['DIASEMANA'] = furtos_2018_grp['DIASEMANA'].map({0: 'Segunda', 1: 'Terca', 2: 'Quarta', 3: 'Quinta', 4: 'Sexta', 5: 'Sabado', 6: 'Domingo'})

[ ]: df = furtos_2018_grp.rename({'DIASEMANA': 'DAY_WEEK', 'DATAOCORRENCIA': 'DATE', 'TOTAL': 'NUMBER_OF_OCCURRENCES'}, axis=1)

[ ]: df['DAY_WEEK'] = df['DAY_WEEK'].map({'Segunda': 'Monday', 'Terca': 'Tuesday', 'Quarta': 'Wednesday', 'Quinta': 'Thursday', 'Sexta': 'Friday', 'Sabado': 'Saturday', 'Domingo': 'Sunday'})

[ ]: df.sort_values('DAY_WEEK', inplace=True)

[ ]: df

[ ]: df1 = df.groupby(['DAY_WEEK'])['NUMBER_OF_OCCURRENCES'].sum().rename('TOTAL').reset_index()
```

```
[ ]: df1  
  
[ ]: df2 = df.groupby(['DAY_WEEK'])['DAY_WEEK'].count().rename('QTD').  
    ↪reset_index()  
  
[ ]: df2  
  
[ ]: df3 = df1.merge(df2, how='left', on=['DAY_WEEK'])  
  
[ ]: df3  
  
[ ]: df3['LAMBDA'] = df3['TOTAL']/df3['QTD']  
  
[ ]: df3  
  
[ ]: df3.sort_values('DAY_WEEK', inplace=True)  
  
[ ]: df3  
  
[ ]: x = []  
x=np.array(x)  
  
for index, item in df3.iterrows():  
    np.random.seed(42)  
    y = np.random.poisson(item['LAMBDA'], item['QTD'])  
    x = np.concatenate((x, y))  
  
[ ]: x  
  
[ ]: x.size  
  
[ ]: df['expected_value'] = x.tolist()  
  
[ ]: df  
  
[ ]: chi_square = []  
p_value = []
```

```

for index, item in df3.iterrows():
    df_day = df.loc[df['DAY_WEEK'] == item['DAY_WEEK']]

    chi_square.append(chisquare(f_obs = df_day['NUMBER_OF_OCCURRENCES'],
                                f_exp = df_day['expected_value'])[0])
    p_value.append(chisquare(f_obs = df_day['NUMBER_OF_OCCURRENCES'],
                                f_exp = df_day['expected_value'])[1])

```

```
[ ]: chi_square = np.array(chi_square)
      p_value = np.array(p_value)
```

```
[ ]: chi_square
```

```
[ ]: p_value
```

```
[ ]: df3['chi_square'] = chi_square.tolist()
      df3['p_value'] = p_value.tolist()
```

```
[ ]: df3
```

```
[ ]: df3.rename({'DAY_WEEK': 'DIASEMANA'}, axis=1, inplace=True)
      df3
```

```
[ ]: df3['DIASEMANA'] = df3['DIASEMANA'].map({'Monday': 'Segunda',
                                                'Tuesday': 'Terca',
                                                'Wednesday': 'Quarta',
                                                'Thursday': 'Quinta',
                                                'Friday': 'Sexta',
                                                'Saturday': 'Sabado',
                                                'Sunday': 'Domingo'})
      df3
```

```
[ ]: cats = ['Segunda', 'Terca', 'Quarta', 'Quinta', 'Sexta',
            'Sabado', 'Domingo']
      cat_type = CategoricalDtype(categories=cats, ordered=True)
      df3['DIASEMANA'] = df3['DIASEMANA'].astype(cat_type)
      df3.sort_values("DIASEMANA", inplace=True)
      df3
```

```
[ ]: def render_mpl_table(data, col_width=3.0, row_height=0.625, ▾
    ↪font_size=14,
                  header_color="#40466e", ▾
    ↪row_colors=['#f1f1f2', 'w'], edge_color='w',
                  bbox=[0, 0, 1, 1], header_columns=0,
                  ax=None, **kwargs):
    if ax is None:
        size = (np.array(data.shape[::-1]) + np.array([0, 1])) * ▾
    ↪np.array([col_width, row_height])
        fig, ax = plt.subplots(figsize=size)
        ax.axis('off')

    mpl_table = ax.table(cellText=data.values, bbox=bbox, ▾
    ↪colLabels=data.columns, **kwargs)

    mpl_table.auto_set_font_size(False)
    mpl_table.set_fontsize(font_size)

    for k, cell in six.iteritems(mpl_table._cells):
        cell.set_edgecolor(edge_color)
        if k[0] == 0 or k[1] < header_columns:
            cell.set_text_props(weight='bold', color='w')
            cell.set_facecolor(header_color)
        else:
            cell.set_facecolor(row_colors[k[0]]%len(row_colors) ])
    fig.savefig('qui-quadrado_resultados_furtos_2018.png')

    return ax

render_mpl_table(df3, header_columns=0, col_width=4.0)
```

6.2 REALIZAÇÃO DO TESTE DE HIPÓTESE PARA OS DADOS DE ROUBOS

```
[ ]: import pandas as pd
from pandas.api.types import CategoricalDtype
from pandas.plotting import table
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import datetime as dt
import seaborn as sns
import six
from scipy.stats import chisquare
```

```
[ ]: df_roubos = pd.read_csv('D:\\\\EPS\\\\TCC\\\\data_cleaned\\\\SSP\\\\roubos2018.csv', sep=",", decimal=".")
```

```
[ ]: roubos_2018_grp = df_roubos.groupby(['DIASEMANA', 'DATAOCORRENCIA'])['DATAOCORRENCIA'].count().rename('TOTAL').reset_index()
```

```
[ ]: roubos_2018_grp
```

```
[ ]: roubos_2018_grp['DIASEMANA'] = roubos_2018_grp['DIASEMANA'].map({0: 'Segunda', 1: 'Terca', 2: 'Quarta', 3: 'Quinta', 4: 'Sexta', 5: 'Sabado', 6: 'Domingo'})
```

```
[ ]: df = roubos_2018_grp.rename({'DIASEMANA': 'DAY_WEEK', 'DATAOCORRENCIA': 'DATE', 'TOTAL': 'NUMBER_OF_OCCURRENCES'}, axis=1)
```

```
[ ]: df['DAY_WEEK'] = df['DAY_WEEK'].map({'Segunda': 'Monday', 'Terca': 'Tuesday', 'Quarta': 'Wednesday', 'Quinta': 'Thursday', 'Sexta': 'Friday', 'Sabado': 'Saturday', 'Domingo': 'Sunday'})
```

```
[ ]: df.sort_values('DAY_WEEK', inplace=True)
```

```
[ ]: df
```

```
[ ]: df1 = df.groupby(['DAY_WEEK'])['NUMBER_OF_OCCURRENCES'].sum().rename('TOTAL').reset_index()
```

```
[ ]: df1
[ ]: df2 = df.groupby(['DAY_WEEK'])['DAY_WEEK'].count().rename('QTD').
    ↪reset_index()
[ ]: df2
[ ]: df3 = df1.merge(df2, how='left', on=['DAY_WEEK'])
[ ]: df3
[ ]: df3['LAMBDA'] = df3['TOTAL']/df3['QTD']
[ ]: df3
[ ]: df3.sort_values('DAY_WEEK', inplace=True)
[ ]: df3
[ ]: x = []
x=np.array(x)

for index, item in df3.iterrows():
    np.random.seed(42)
    y = np.random.poisson(item['LAMBDA'], item['QTD'])
    x = np.concatenate((x, y))
[ ]: x
[ ]: x.size
[ ]: df['expected_value'] = x.tolist()
[ ]: df
[ ]: chi_square = []
p_value = []
```

```

for index, item in df3.iterrows():
    df_day = df.loc[df['DAY_WEEK'] == item['DAY_WEEK']]

    chi_square.append(chisquare(f_obs = df_day['NUMBER_OF_OCCURRENCES'],
                                f_exp = df_day['expected_value'])[0])
    p_value.append(chisquare(f_obs = df_day['NUMBER_OF_OCCURRENCES'],
                                f_exp = df_day['expected_value'])[1])

```

```
[ ]: chi_square = np.array(chi_square)
      p_value = np.array(p_value)
```

```
[ ]: chi_square
```

```
[ ]: p_value
```

```
[ ]: df3['chi_square'] = chi_square.tolist()
      df3['p_value'] = p_value.tolist()
```

```
[ ]: df3
```

```
[ ]: df3.rename({'DAY_WEEK': 'DIASEMANA'}, axis=1, inplace=True)
      df3
```

```
[ ]: df3['DIASEMANA'] = df3['DIASEMANA'].map({'Monday': 'Segunda',
                                                'Tuesday': 'Terca',
                                                'Wednesday': 'Quarta',
                                                'Thursday': 'Quinta',
                                                'Friday': 'Sexta',
                                                'Saturday': 'Sabado',
                                                'Sunday': 'Domingo'})
      df3
```

```
[ ]: cats = ['Segunda', 'Terca', 'Quarta', 'Quinta', 'Sexta',
            'Sabado', 'Domingo']
      cat_type = CategoricalDtype(categories=cats, ordered=True)
      df3['DIASEMANA'] = df3['DIASEMANA'].astype(cat_type)
      df3.sort_values("DIASEMANA", inplace=True)
      df3
```

```
[ ]: def render_mpl_table(data, col_width=3.0, row_height=0.625,
                         font_size=14,
                         header_color="#40466e",
                         row_colors=['#f1f1f2', 'w'],
                         edge_color='w',
                         bbox=[0, 0, 1, 1], header_columns=0,
                         ax=None, **kwargs):
    if ax is None:
        size = (np.array(data.shape[::-1]) + np.array([0, 1])) * \
            np.array([col_width, row_height])
        fig, ax = plt.subplots(figsize=size)
        ax.axis('off')

    mpl_table = ax.table(cellText=data.values, bbox=bbox,
                         colLabels=data.columns, **kwargs)

    mpl_table.auto_set_font_size(False)
    mpl_table.set_fontsize(font_size)

    for k, cell in six.iteritems(mpl_table._cells):
        cell.set_edgecolor(edge_color)
        if k[0] == 0 or k[1] < header_columns:
            cell.set_text_props(weight='bold', color='w')
            cell.set_facecolor(header_color)
        else:
            cell.set_facecolor(row_colors[k[0]]%len(row_colors) )
    fig.savefig('qui-quadrado_resultados_roubos_2018.png')

    return ax

render_mpl_table(df3, header_columns=0, col_width=4.0)
```