Instituto Federal de Educação, Ciência e Tecnologia da Paraíba
Campus Campina Grande
Curso Superior de Engenharia da Computação
Disciplina: Probabilidade e Estatística Aplicada a Computação
1º Projeto: Estatística Descritiva
Professor: Paulo Ribeiro
Aluno: Wesley Wevertton de Azevedo Palmeira (201621250039)

ESTATÍSTICA DESCRITIVA UTILIZANDO PYTHON

Campina Grande – Março de 2017

INTRODUÇÃO

A Estatística é o “ramo da matemática que trata da coleta, da análise, da interpretação e da apresentação de massas de dados numéricos”. Desta forma, a Estatística é também importante parte da Ciência de Dados, utilizada para analisar uma série de situações e problemas através de seus mais variados conceitos.

A escolha da linguagem python não foi meramente aleatória, mas sim porquer a mesma traz uma série de bibliotecas e métodos que facilitam e muito nossa análise.

DESENVOLVIMENTO

Para confecção do trabalho proposto, foram escolhidas algumas bibliotecas de python que já possuem métodos especificos para alguns cálculos de estatística descritiva, são elas: ***pandas, matplotlib, math.***

O primeiro desafio proposto trata-se de ler um arquivo .csv e organizar os dados de forma satisfatória para nosso objetivo. O arquivo .csv proposto para a leitura trata-se de dados que se referem a ao programa Google Trends que, nesse caso, apresenta dados sobre buscas no google relacionadas a gripe no Brasil, por estado.

O objetivo era analisar dados a partir da data 22 de janeiro de 2006 e somente dos estados Ceará, Minas Gerais, Rio de Janeiro, Paraná e os valores nacionais. Para abrir utilizamos uma função própria do ***pandas*** chamada *read_csv()*. Essa função foi chamada e veio toda a base de dados do arquivo .csv, para filtrar apenas os dados de interesse foi implementado um pequeno algoritmo especificado no código em anexo.

As primeiras medidas a serem calculadas foram as medidas de posição, ou seja, Média, Mediana e Moda.

A média é uma medida de tendência central que indica o valor onde estão concentrados os dados de um conjunto de valores, representando um valor significativo para o mesmo.Para o calculo da Média, foi utilizado o método *mean()* e obtemos o seguinte resultado:

ESTADO	**MÉDIA**
Brazil	199.408818
Ceará	161.827655
Minas Gerais	218.727455
Paraná	196.705411
Rio de Janeiro	209.102204

Tabela 1 - Médias

A mediana é o valor que separa a metade superior da metade inferior de uma distribuição de dados, ou o valor no centro da distribuição.Para o calculo da Mediana, foi utilizado o método *median()* e obtemos o seguinte resultado:

ESTADO	**MEDIANA**
Brazil	192
Ceará	153
Minas Gerais	208
Paraná	183

Rio de Janeiro	204

Tabela 2 – Medianas

A moda é simples. Nada mais é que o valor que mais se repete dentro de um conjunto**.**Para o calculo da Moda, foi utilizado o método *mode()* e obtemos o seguinte resultado:

ESTADO	**MODA**
Brazil	149, 193, 196
Ceará	122, 148
Minas Gerais	128
Paraná	181
Rio de Janeiro	260

Tabela 3 – Modas

Após o cálculo das medidas de posição, fizemos todos os calculos relacionados as medidas de Dispersão, ou seja, Amplitude, Variância, Desvio Padrão, Desvio Absoluto, Covariância e Correlação.

A amplitude nada mais é do que a diferença entre o maior e o menor valor de um conjunto de dados. Para fazer este cálculo no Pandas, usaremos as funções *max()* e *min()*, que obviamente, retornam o valor máximo e mínimo de um conjunto de dados, e depois subtrairemos um do outro, logo obtemos o seguinte resultado:

ESTADO	**AMPLITUDE**
Brazil	343
Ceará	258
Minas Gerais	377
Paraná	494
Rio de Janeiro	281

Tabela 4 – Amplitudes

A variância é uma medida que expressa quanto os dados de um conjunto estão afastados de seu valor esperado. Para calcular esse valor, utilizamos o método *var()*, e obtemos o seguinte resultado:

ESTADO	**VARIÂNCIA**
Brazil	4326,282332
Ceará	2388,355780
Minas Gerais	6268,957699
Paraná	6209,702200
Rio de Janeiro	3522,180296

Tabela 5 – Variâncias

O desvio padrão indica quanto os dados estão afastados da média. Um valor de desvio padrão alto indica que os valores estão mais espalhados, mais longe da média, e um desvio padrão baixo indica que os valores estão mais próximos da média. Para calcular o desvio padrão foi utlizada a função *std()*, obtendo os resultados abaixo:

ESTADO	**DESVIO PADRÃO**
Brazil	65,774481
Ceará	48,870807
Minas Gerais	79,176750
Paraná	78,801664
Rio de Janeiro	59,347959

Tabela 6 – Desvios Padrão

O Desvio Absoluto é calculado da seguinte forma: primeiro, encontramos a média dos valores, depois, calculamos a distância de cada ponto desta média; somamos as distâncias e dividimos o resultado pela média destas distâncias. Para calcula-lo, foi utilizada a função *mad()*, e assim obtemos os seguintes resultados:

ESTADO	**DESVIO ABSOLUTO**
Brazil	52,746463
Ceará	39,648202
Minas Gerais	63,868667
Paraná	61,360316
Rio de Janeiro	49,156694

Tabela 7 – Desvios Absolutos

A covariância é uma medida numérica que indica a inter-dependência entre duas variáveis. A covariância indica como duas variáveis se comportam conjuntamente em relação às suas médias. Uma covariância igual a 0 indica que as duas variáveis são totalmente independentes, enquanto que uma covariância alta e positiva indica que uma variável é grande quando a outra é grande. Analogamente, uma covariância negativa e com valor absoluto alto indica que uma variável é pequena quando a outra é grande. Para o calculo da covariância, foi utilizada a função *cov()*. E assim, obtemos os seguintes resultados:

#	Brazil	Ceará	Minas Gerais	Paraná	Rio de Janeiro
Brazil	4326,282332	2798,922005	5082,047388	4876,544370	3673,361748
Ceará	2798,922005	2388,355780	3224,344500	2961,880854	2305,610019
Minas Gerais	5082,047388	3224,344500	6268,957699	5674,783004	4256,724698
Paraná	4876,544370	2961,880854	5674,783004	6209,702200	4032,811293
Rio de Janeiro	3673,361748	2305,610019	4256,724698	4032,811293	3522,180296

Tabela 8 – Covariâncias

A correlação também é outra medida que indica o quanto duas variáveis estão relacionadas. Seu valor fica sempre entre -1, que indica uma anti-correlação perfeita, e 1, que indica uma correlação perfeita. Para o calculo da correlação, foi utilizada a função *corr()*, obtendo os seguintes resultados:

#	Brazil	Ceará	Minas Gerais	Paraná	Rio de Janeiro
Brazil	1,000000	0,870731	0,975851	0,940848	0,941024
Ceará	0,870731	1,000000	0,833286	0,769100	0,794933
Minas Gerais	0,975851	0,833286	1,000000	0,909528	0,905883
Paraná	0,940848	0,769100	0,909528	1,000000	0,862317
Rio de Janeiro	0,941024	0,794933	0,905883	0,862317	1,000000

Tabela 9 – Correlações

Após o calculos de todas as medidas de dispersão e posição, entramos para a construção das tabelas de frequencias absolutas de cada estado. O primeiro passo foi a construção das classes para cada estado, para a quantidade de classes, foi utilizado o método da raiz quadrada. Foi contada a quantidade de medidas de cada estado, e pela fórmula da raiz quadrada, obtemos a quantidade de classe adequada para cada Estado. De posse da quantidade de classe de cada estado, foram construídas as classes em si por meio da função *cut()*. Tendo a classe que cada medida pertencia por meio desta função, foi utilizada a função *pd.value_counts()* para contabilizar a quantidade de ocorrencia de cada classe, ou seja, o calculo da frequencia simples absoluta. Assim, obtemos os seguintes resultados:

CLASSE	**FREQUENCIA SIMPLES ABSOLUTA**
(82.742, 94.727]	11
(94.727, 106.455]	32
(106.455, 118.182]	60
(118.182, 129.909]	56
(129.909, 141.636]	47
(141.636, 153.364]	45
(153.364, 165.0909]	43
(165.0909, 176.818]	32
(176.818, 188.545]	37
(188.545, 200.273]	35
(200.273, 212]	19
(212, 223.727]	27
(223.727, 235.455]	15
(235.455, 247.182]	9
(247.182, 258.909]	9
(258.909, 270.636]	5
(270.636, 282.364]	7

(282.364, 294.0909]	3
(294.0909, 305.818]	2
(305.818, 317.545]	2
(317.545, 329.273]	1
(329.273, 341]	2
TOTAL	**499**

Tabela 10 – Frequências Simples Absolutas do Ceará

CLASSE	**FREQUENCIA SIMPLES ABSOLUTA**
(94.623, 112.136]	18
(112.136, 129.273]	42
(129.273, 146.409]	33
(146.409, 163.545]	53
(163.545, 180.682]	52
(180.682, 197.818]	33
(197.818, 214.955]	35
(214.955, 232.0909]	44
(232.0909, 249.227]	31
(249.227, 266.364]	35
(266.364, 283.5]	23
(283.5, 300.636]	21
(300.636, 317.773]	18
(317.773, 334.909]	16
(334.909, 352.0455]	12
(352.0455, 369.182]	9
(369.182, 386.318]	5
(386.318, 403.455]	5
(403.455, 420.591]	4
(420.591, 437.727]	2
(437.727, 454.864]	3
(454.864, 472]	5
TOTAL	**499**

Tabela 11 – Frequências Simples Absolutas de Minas Gerais

CLASSE	**FREQUENCIA SIMPLES ABSOLUTA**
(105.719, 118.773]	15
(118.773, 131.545]	25

(131.545, 144.318]	31
(144.318, 157.0909]	36
(157.0909, 169.864]	49
(169.864, 182.636]	39
(182.636, 195.409]	31
(195.409, 208.182]	42
(208.182, 220.955]	24
(220.955, 233.727]	31
(233.727, 246.5]	37
(246.5, 259.273]	34
(259.273, 272.0455]	37
(272.0455, 284.818]	19
(284.818, 297.591]	14
(297.591, 310.364]	8
(310.364, 323.136]	5
(323.136, 335.909]	6
(335.909, 348.682]	4
(348.682, 361.455]	5
(361.455, 374.227]	4
(374.227, 387]	3
TOTAL	**499**

Tabela 12 – Frequências Simples Absolutas do Rio de Janeiro

CLASSE	**FREQUENCIA SIMPLES ABSOLUTA**
(67.506, 90.455]	14
(90.455, 112.909]	45
(112.909, 135.364]	61
(135.364, 157.818]	60
(157.818, 180.273]	58
(180.273, 202.727]	58
(202.727, 225.182]	44
(225.182, 247.636]	51
(247.636, 270.0909]	32
(270.0909, 292.545]	25
(292.545, 315]	12
(315, 337.455]	7
(337.455, 359.909]	14

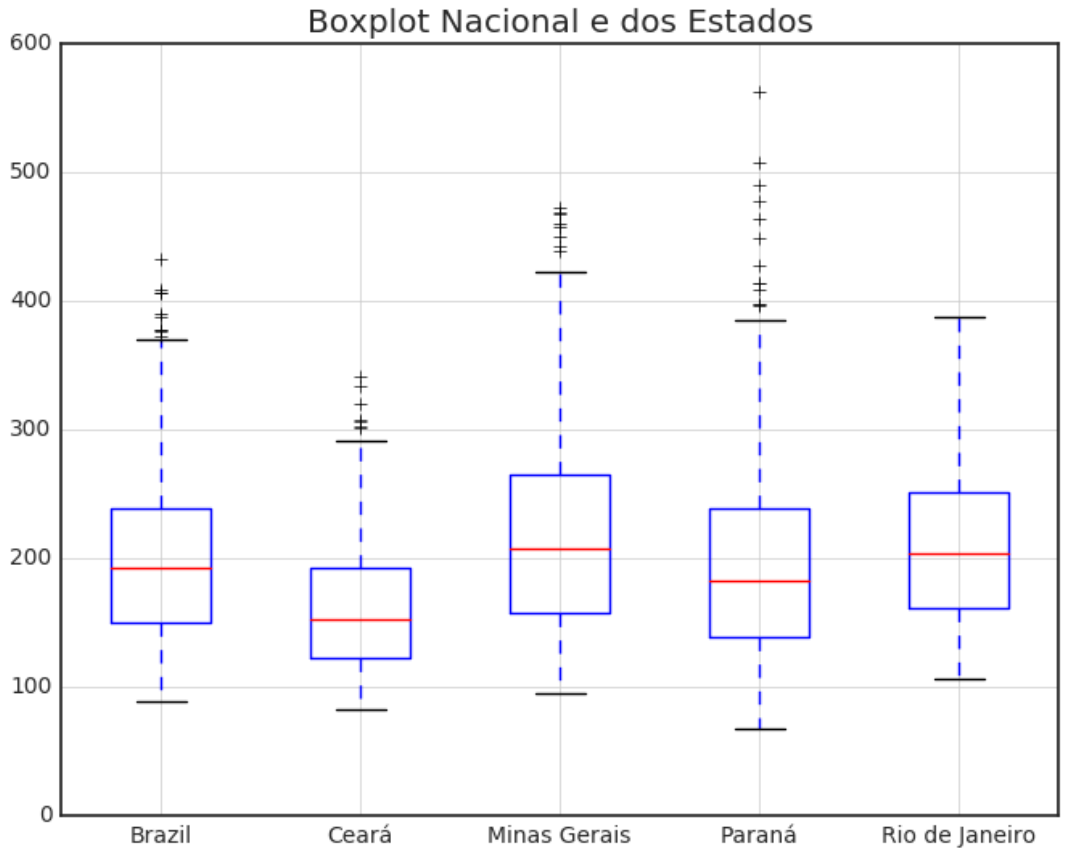
(359.909, 382.364]	4
(382.364, 404.818]	4
(404.818, 427.273]	3
(427.273, 449.727]	2
(449.727, 472.182]	1
(472.182, 494.636]	2
(494.636, 517.0909]	1
(517.0909, 539.545]	0
(539.545, 562]	1
TOTAL	**499**

Tabela 13 – Frequências Simples Absolutas do Paraná

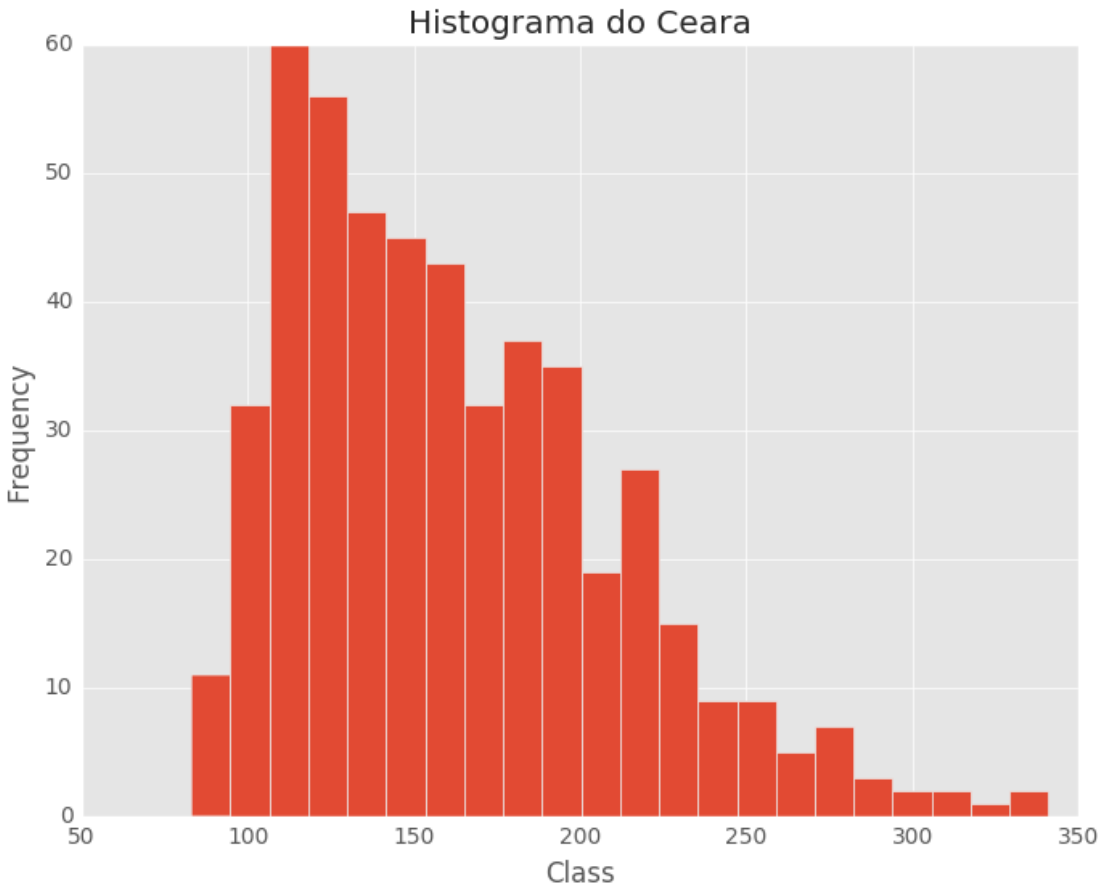
CLASSE	**FREQUENCIA SIMPLES ABSOLUTA**
(88.657, 104.591]	11
(104.591, 120.182]	36
(120.182, 135.773]	34
(135.773, 151.364]	51
(151.364, 166.955]	50
(166.955, 182.545]	52
(182.545, 198.136]	40
(198.136, 213.727]	39
(213.727, 229.318]	38
(229.318, 244.909]	34
(244.909, 260.5]	30
(260.5, 276.0909]	21
(276.0909, 291.682]	21
(291.682, 307.273]	9
(307.273, 322.864]	8
(322.864, 338.455]	6
(338.455, 354.0455]	3
(354.0455, 369.636]	5
(369.636, 385.227]	5
(385.227, 400.818]	2
(400.818, 416.409]	3
(416.409, 432]	1
TOTAL	**499**

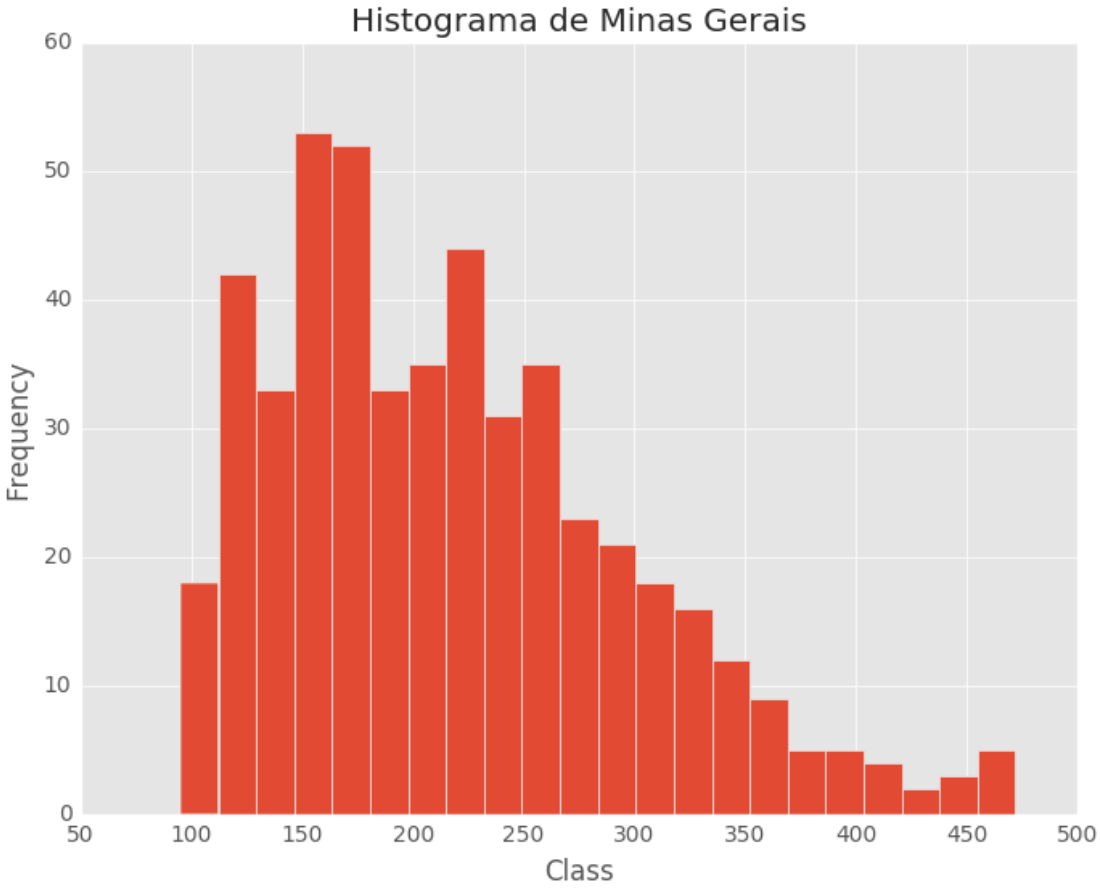
Tabela 13 – Frequências Simples Absolutas do Brazil

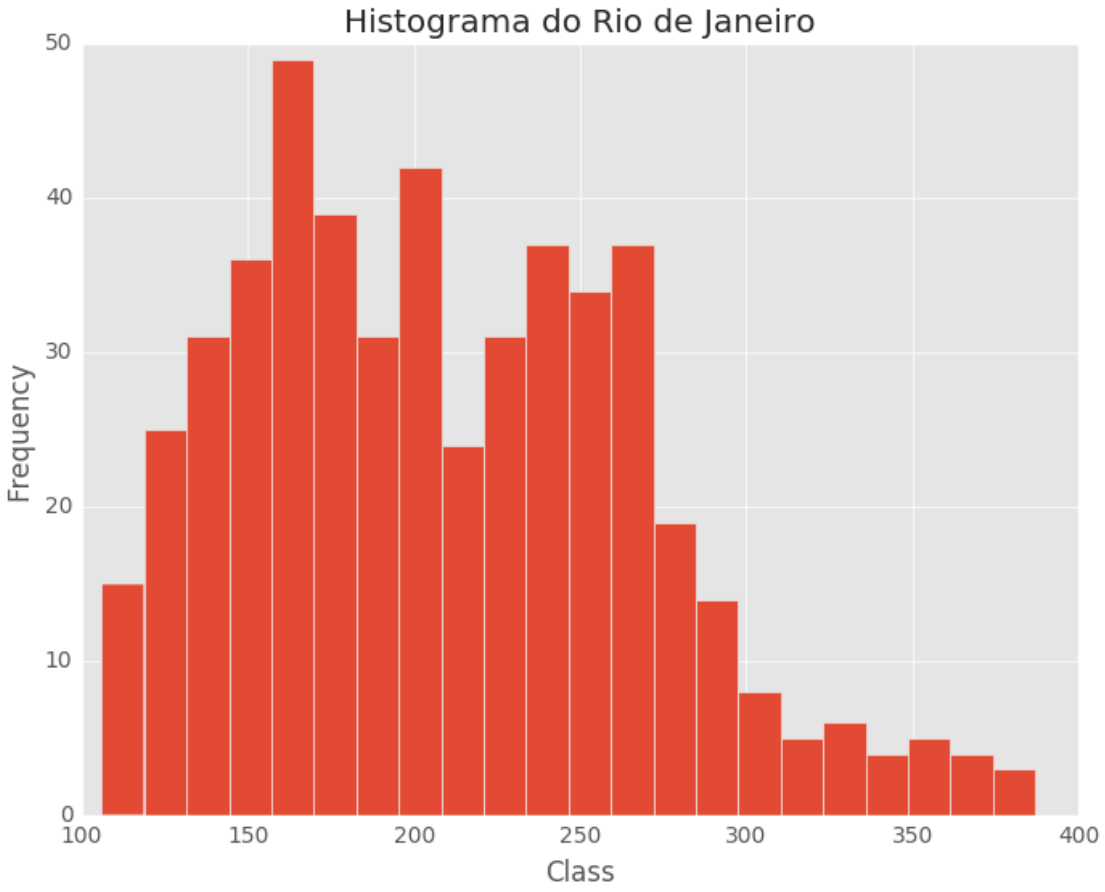
Com todos os dados calculados e organizados, foi escolhido primeiro construir o gráfico boxplot de todos os Estados.O boxplot (gráfico de caixa) é um gráfico utilizado para avaliar a distribuição empírica do dados. O boxplot é formado pelo primeiro e terceiro quartil e pela mediana. As hastes inferiores e superiores se estendem, respectivamente, do quartil inferior até o menor valor não inferior ao limite inferior e do quartil superior até o maior valor não superior ao limite superior. Para confecção do boxplot, utilizamos a função *boxplot()*. Podemos conferir o gráfico gerado abaixo:

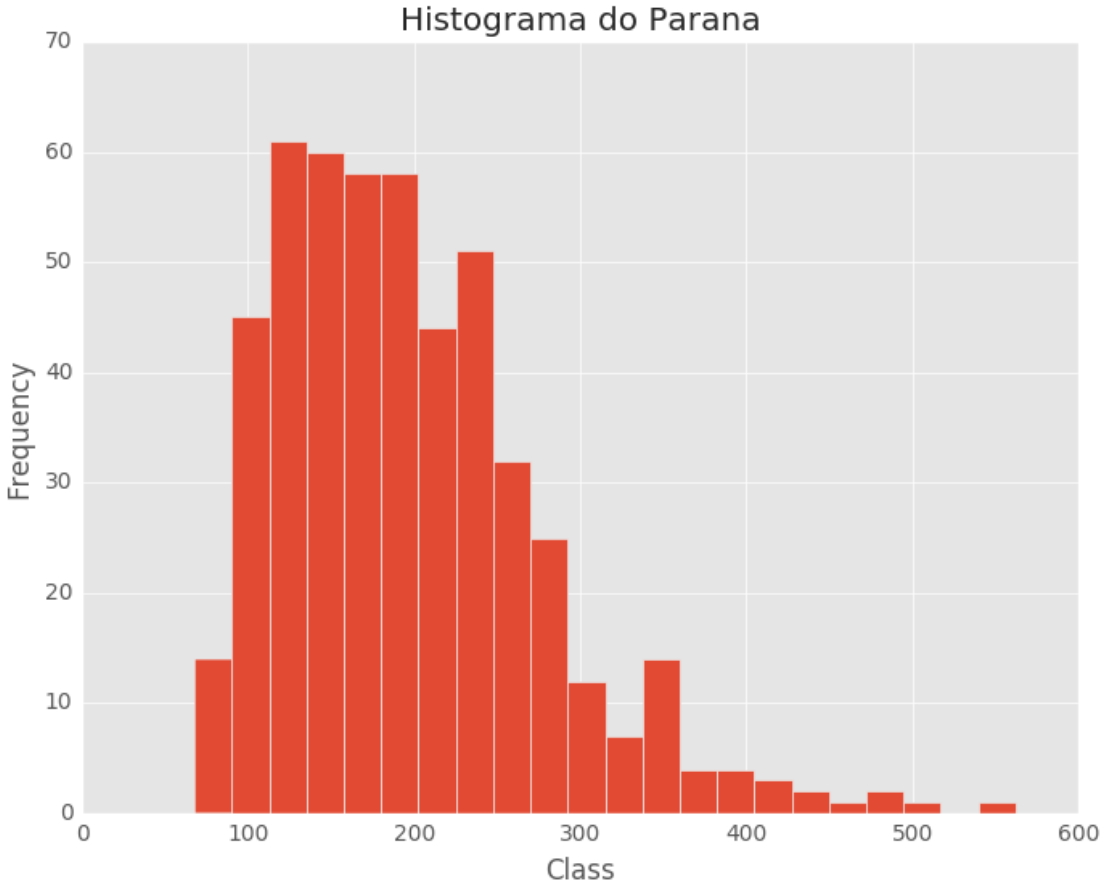


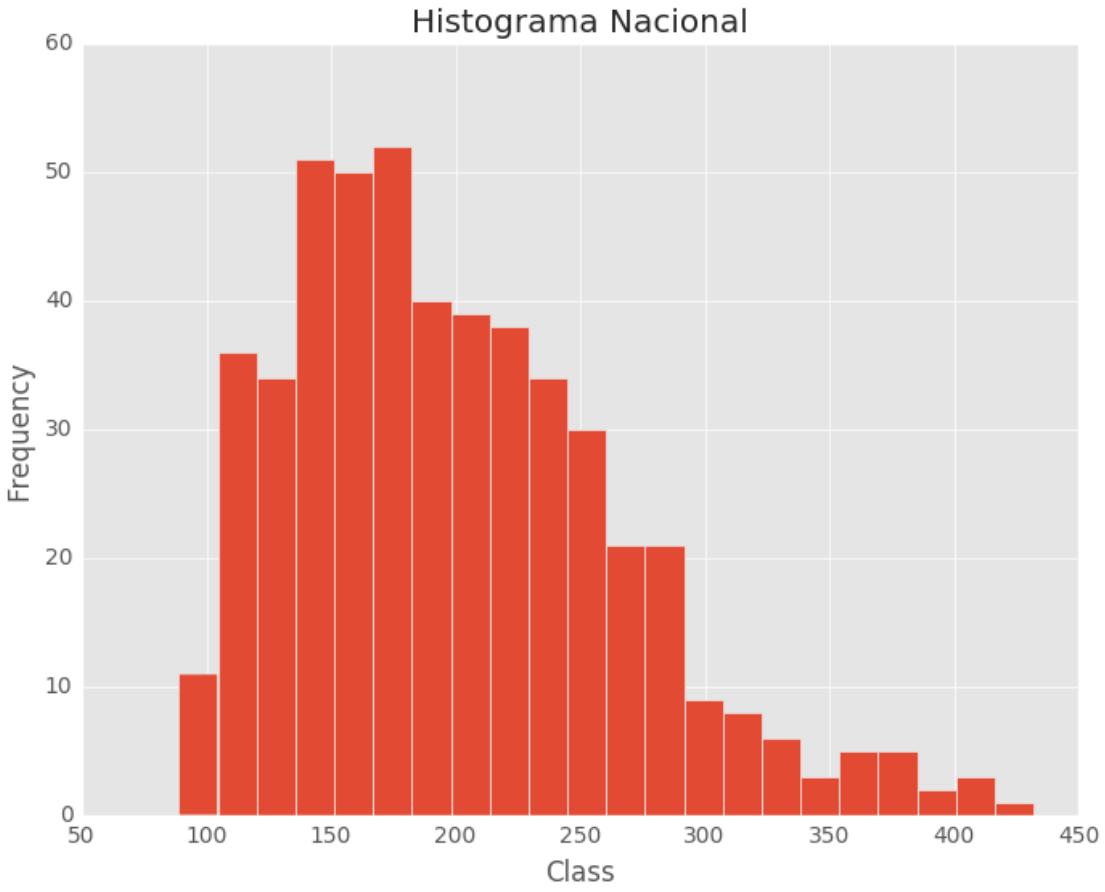
Após a confecção do bloxplot, foi confeccionado todos os histogramas para todos os Estados.Os histogramas são usados para mostrar a frequência com que algo acontece. Por exemplo, em um caso onde fosse necessário mostrar de forma gráfica a distribuição de altura de estudantes de uma escola, uma das maneiras mais adequadas para isso seria fazê-lo por meio de um histograma. Para confecção dos histogramas, utilizamos os dados já de posse e a função *plot.hist()*. Abaixo temos todos os histogramas gerados:


Histograma do Ceara
Frequency
Class
0
10
20
30
40
50
60
50
100
150
200
250
300
350


Histograma de Minas Gerais
Frequency
Class
0
10
20
30
40
50
60
50
100
150
200
250
300
350
400
450
500


Histograma do Rio de Janeiro
Frequency
Class
0
10
20
30
40
50
100
150
200
250
300
350
400


Histograma do Parana
Frequency
Class
0
10
20
30
40
50
60
70
0
100
200
300
400
500
600



CONCLUSÃO

Portanto, vimos que a estatística descritivas nos trás muitos resultados interessantes ao analisar dados não organizados, a partir desses resultados podemos retirar interpretações confiaveis e interessantes e ainda perceber que a linguagem Python em conjunto com algumas bibliotecas faz com que os calculos e gráficos se tornem de fácil implementação, poupando o trabalho de desenvolver algortimos desnecessarios tendo em vista que a mesma já tem diversas funções já implementadas do assunto em questão.

```
# -*- coding: utf-8 -*-
"""
Created on Mon Mar  6 10:31:14 2017

@author: wesley150
"""

# importando bibliotecas necessárias

import pandas as pd

import matplotlib.pyplot as plt

from math import *

dados = pd.read_csv('gripe.csv') #leitura do arquivo por meio da biblioteca pandas

#Exclusão das colunas que não quero analisar
del dados['Distrito Federal']
del dados['Rio Grande do Sul']
del dados['Santa Catarina']
del dados['São Paulo']

#Percorrendo a coluna de datas para achar a data 22 de janeiro de 2006 para começar
for i in range(len(dados)):
    if (dados.get_value(i,'Date') == '2006-01-22'):
        posicao_data = i
#crio um novo dataframe apenas a partir da posição da data 22 de janeiro de 2006
dados = dados[posicao_data:]

print "\n~~~Medidas de Posição~~~" #calculo de medidas de posição por meio de funçõe
print "\nMedia dos valores observados:\n\n", dados.mean()
print "\nMediana dos valores observados:\n\n", dados.median()
print "\nModa dos valores observados:\n\n", dados.median()
print "\n~~~Medidas de Dispersão~~~" #calculo de medidas de dispersão por meio de fu
print "\nAmplitude dos valores observados:\n\n", dados.ix[:, dados.columns != 'Date'
print "\nVariância dos valores observados:\n\n", dados.var()
print "\nDesvio Padrão dos valores observados:\n\n", dados.std()
print "\nDesvio Absoluto dos valores observados:\n\n", dados.mad()
print "\nCovariância dos valores observados:\n\n", dados.cov()
print "\nCorrelação dos valores observados:\n\n", dados.corr()

#calculo do numero de classes e das frequencias em cada classe para cada estado e pa

n_ceara = dados.describe()['Ceará']['count']
k_ceara = int(round(sqrt(n_ceara)))
classes_ceara = pd.cut(dados['Ceará'], k_ceara)
frequencias_ceara = pd.value_counts(classes_ceara)

n_minas = dados.describe()['Minas Gerais']['count']
k_minas = int(round(sqrt(n_minas)))
classes_minas = pd.cut(dados['Minas Gerais'], k_minas)
frequencias_minas = pd.value_counts(classes_minas)

n_rio = dados.describe()['Rio de Janeiro']['count']
k_rio = int(round(sqrt(n_rio)))
classes_rio = pd.cut(dados['Rio de Janeiro'], k_rio)
frequencias_rio = pd.value_counts(classes_rio)

n_parana = dados.describe()['Paraná']['count']
k_parana = int(round(sqrt(n_parana)))
classes_parana = pd.cut(dados['Paraná'], k_parana)
frequencias_parana = pd.value_counts(classes_parana)
```

```
n_br = dados.describe()['Brazil']['count']
k_br = int(round(sqrt(n_br)))
classes_br = pd.cut(dados['Brazil'], k_br)
frequencias_br = pd.value_counts(classes_br, sort = False)

#Exibição na tela das tabelas de frequencias calculadas

print "\nTabela de frequencias simples absolutas do Ceará:\n\n", frequencias_ceara
print "\nTabela de frequencias simples absolutas de Minas Gerais:\n\n", frequencias_
print "\nTabela de frequencias simples absolutas do Rio de Janeiro:\n\n", frequencia
print "\nTabela de frequencias simples absolutas do Paraná:\n\n", frequencias_parana
print "\nTabela de frequencias simples absolutas do Brazil:\n\n", frequencias_br

#Gerando o boxplot para todos os Estados e para o Brasil
plt.figure(1)
plt.style.use('seaborn-white')
#plt.style.available <-- mostra os estilos de graficos disponiveis! (está aqui para
plt.title('Boxplot Nacional e dos Estados')
dados.boxplot()

#Gerando histogramas para todos os Estados e para o Brasil
plt.figure(2)
plt.style.use('ggplot')
plt.style.available
plt.xlabel('Class')
dados['Ceará'].plot.hist(bins = k_ceara, grid = True, title = 'Histograma do Ceara')
plt.figure(3)
plt.xlabel('Class')
dados['Minas Gerais'].plot.hist(bins = k_minas, grid = True, title = 'Histograma de
plt.figure(4)
plt.xlabel('Class')
dados['Rio de Janeiro'].plot.hist(bins = k_rio, grid = True, title = 'Histograma do
plt.figure(5)
plt.xlabel('Class')
dados['Paraná'].plot.hist(bins = k_parana, grid = True, title = 'Histograma do Paran
plt.figure(6)
plt.xlabel('Class')
dados['Brazil'].plot.hist(bins = k_br, grid = True, title = 'Histograma Nacional')
```