Trabalho 01 de Métodos Quantitativos

Pedro Henrique Ventura e Allan Landau de Carvalho Hilgemberg

Bibliotecas ¶

In [4]:

```
import numpy as np
import pandas as pd
```

Primeira visualização dos dados

Base do IMDB de ranking de filmes

In [9]:

```
import matplotlib.pyplot as plt
df = pd.read_csv(r'C:\Users\pe-ri\Documents\Python Scripts\PGCC-Métodos Quantitativos\PGCC-MQ\data.csv', delimiter='\t', low_memory=False)
print(df.head())
df.describe()
```

```
      tconst  averageRating  numVotes
0  tt0000001            5.6      1538
1  tt0000002            6.1       186
2  tt0000003            6.5      1198
3  tt0000004            6.2       114
4  tt0000005            6.1      1909
```

Out[9]:

	averageRating	numVotes
count	978337.000000	9.783370e+05
mean	6.885982	9.590946e+02
std	1.401628	1.563819e+04
min	1.000000	5.000000e+00
25%	6.100000	9.000000e+00
50%	7.100000	2.000000e+01
75%	7.900000	7.600000e+01
max	10.000000	2.139781e+06

Trabalhando os dados

1. Pegar somente o campo das notas e as ordenar

2. Criar variaveis com dados em formato discretas (Xd) e contínuo (Xc)

In [10]:

```
Xc = df['averageRating'].sort_values(ascending=True)
Xd = Xc.astype(int)
totalAtributos = len(Xc)
```

CDF

1. Discreto

1.1 Calculando

In [12]:

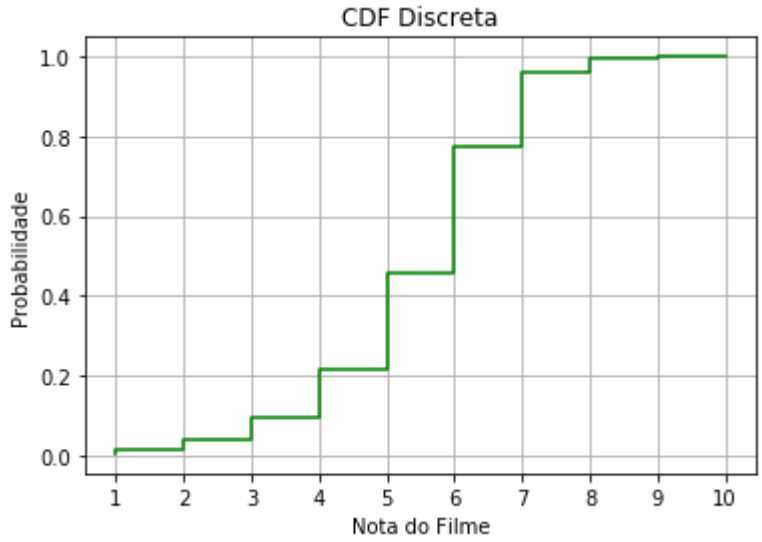
```
"""1. Discreto"""
"""1.1 Calculando"""
"""Pegando a quantidade de ocorrências e calculando a probabilidade"""

print("DISCRETO")
ocorrenciasD = dict()
totalOcorrenciaD = 0
#calculando quantidade de ocorrências
for i in Xd:
    try:
        ocorrenciasD[i] += 1
    except KeyError:
        ocorrenciasD[i] = 1
        totalOcorrenciaD = totalOcorrenciaD + 1

print("Total de ocorrências: ", totalOcorrenciaD)
print("Ocorrências: ", ocorrenciasD)

#calculando a probabilidade
probabilidadeD = []
eixoXD = []
for itemD, totalOcorrenciaIndividualD in ocorrenciasD.items():
    probabilidadeD.append(totalOcorrenciaIndividualD/totalAtributos)
    eixoXD.append(itemD)
print("\nProbabilidade:", probabilidadeD)

#Calculando eixo X da CDF
ValorEixoXD = []
ValorEixoXD.append(probabilidadeD[0])

for i in range(itemD):
    if i != 0:
        ValorEixoXD.append(probabilidadeD[i]+ValorEixoXD[i-1])


print("\nEixo X:", eixoXD, ValorEixoXD)
```

```
DISCRETO
Total de ocorrências:  10
Ocorrências:  {1: 4034, 2: 9294, 3: 22694, 4: 53962, 5: 122091, 6: 236913,
7: 308646, 8: 182690, 9: 34949, 10: 3064}

Probabilidade: [0.004123323558242201, 0.009499794038250624, 0.023196505907
4736, 0.055156863125896294, 0.12479442155412705, 0.24215888799053906, 0.31
54802486259847, 0.18673524562599594, 0.03572286441175178, 0.00313184516173
87464]

Eixo X: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] [0.004123323558242201, 0.013623117
596492824, 0.03681962350396642, 0.09197648662986271, 0.21677090818398975,
0.4589297961745288, 0.7744100448005136, 0.9611452904265095, 0.996868154838
2613, 1.0]
```

1.2 Plotando

In [14]:

```
"""1.2 Plotando"""
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('CDF Discreta')
plt.grid(True)
plt.xticks(eixoXD)
plt.step(eixoXD, ValorEixoXD, color='g')
plt.show()

ValorEixoXD1 = ValorEixoXD
```



2. Continuo

2.1 Calculando

In [15]:

```
"""2. Continuo"""
"""2.1 Calculando"""
"""Pegando a quantidade de ocorrências e calculando a probabilidade"""

print("Continuo")
ocorrenciasC = dict()
totalOcorrenciaC = 0
#calculando quantidade de ocorrências
for j in Xc:
    try:
        ocorrenciasC[j] +=1
    except KeyError:
        ocorrenciasC[j] = 1
        totalOcorrenciaC = totalOcorrenciaC + 1

print("Total de Ocorrências: ", totalOcorrenciaC)
print("Ocorrências: ", ocorrenciasC)

#calculando a probabilidade
probabilidadeC = []
eixoXC = []
for itemC, totalOcorrenciaIndividualC in ocorrenciasC.items():
    probabilidadeC.append(totalOcorrenciaIndividualC/totalAtributos)
    eixoXC.append(itemC)
print("\nProbabilidade:", probabilidadeC)

#Calculando eixo X da CDF
ValorEixoXC = []
ValorEixoXC.append(probabilidadeC[0])

for i in range(totalOcorrenciaC):
    if i != 0:
        ValorEixoXC.append(probabilidadeC[i]+ValorEixoXC[i-1])


print("\nEixo X:", eixoXC, ValorEixoXC)
```

Continuo
Total de Ocorrências: 91
Ocorrências: {1.0: 968, 1.1: 237, 1.2: 251, 1.3: 240, 1.4: 287, 1.5: 288, 1.6: 420, 1.7: 395, 1.8: 500, 1.9: 448, 2.0: 675, 2.1: 643, 2.2: 757, 2.3: 762, 2.4: 894, 2.5: 897, 2.6: 1010, 2.7: 1014, 2.8: 1482, 2.9: 1160, 3.0: 1608, 3.1: 1443, 3.2: 1980, 3.3: 1769, 3.4: 2295, 3.5: 2101, 3.6: 2830, 3.7: 2529, 3.8: 3436, 3.9: 2703, 4.0: 4103, 4.1: 3424, 4.2: 4735, 4.3: 4168, 4.4: 5240, 4.5: 4989, 4.6: 6654, 4.7: 6076, 4.8: 8006, 4.9: 6567, 5.0: 9869, 5.1: 8163, 5.2: 11095, 5.3: 10067, 5.4: 12016, 5.5: 11354, 5.6: 14029, 5.7: 13278, 5.8: 17487, 5.9: 14733, 6.0: 20618, 6.1: 17474, 6.2: 23644, 6.3: 20506, 6.4: 24398, 6.5: 22648, 6.6: 26624, 6.7: 24786, 6.8: 30843, 6.9: 25372, 7.0: 33339, 7.1: 28035, 7.2: 34881, 7.3: 29370, 7.4: 32542, 7.5: 29166, 7.6: 32902, 7.7: 29079, 7.8: 33380, 7.9: 25952, 8.0: 30601, 8.1: 24364, 8.2: 26630, 8.3: 19490, 8.4: 19012, 8.5: 15556, 8.6: 15165, 8.7: 12238, 8.8: 11880, 8.9: 7754, 9.0: 8625, 9.1: 5132, 9.2: 5852, 9.3: 3413, 9.4: 3387, 9.5: 2061, 9.6: 2573, 9.7: 1545, 9.8: 1855, 9.9: 506, 10.0: 3064}

Probabilidade: [0.0009894341111498389, 0.00024224781440342131, 0.0002565578118787289, 0.0002453142424338444, 0.00029335494824380554, 0.00029437709092061323, 0.00042929992425922765, 0.00040374635733903555, 0.0005110713384038425, 0.00045791991920984284, 0.0006899463068451873, 0.000657237741187341 4, 0.0007737620063434174, 0.0007788727197274559, 0.0009137955530660703, 0.0009168619810964934, 0.0010323641035757617, 0.0010364526742829925, 0.0015114815447028989, 0.0011856855050969145, 0.0016436054243067573, 0.0014749518826334893, 0.002023842500079216, 0.0018081703952727945, 0.0023458174432736366, 0.002147521763972946, 0.0028926637753657483, 0.002584998829646635, 0.0035120822375112053, 0.002762851655411172, 0.004193851402941931, 0.0034998165253895 13, 0.0048398455746843875, 0.00426029067693443, 0.0053560276264722 68, 0.00509946981459354, 0.006801337371478335, 0.006210538904283493, 0.008183274270522326, 0.006712410958596 0665, 0.010087526077415043, 0.008343750670781133, 0.011340672999181263, 0.010289910327422963, 0.012282066404521142, 0.011605407952474453, 0.014339639612935011, 0.01357201046265244, 0.017874208989335984, 0.015059228057407621, 0.021074537710420848, 0.017860921134537487, 0.0241675414504409, 0.020960057730618386, 0.024938237028753897, 0.023149487344340 45, 0.027213526627327803, 0.025334828387355277, 0.031525946580779424, 0.02593380399596458, 0.03407721470209141, 0.028655769944303446, 0.035653358709728855, 0.030020330417841704, 0.03326256698867568, 0.029811813311772937, 0.03363053835232645, 0.029722886898890667, 0.034119122551840 52, 0.026526646748513037, 0.03127858805299197, 0.024903484177742435, 0.027219659481483420345, 0.015500793693788541, 0.012508982078772447, 0.012143055000475297, 0.007925694315966788, 0.008815980587466282, 0.005245636217377039, 0.005981578944678572, 0.0034885729559446287, 0.0034619972463476286, 0.002106636056900 6386, 0.0026299731074261733, 0.001579210435667873, 0.0018960746654782554, 0.0005172041944646886, 0.0031318451617387464]

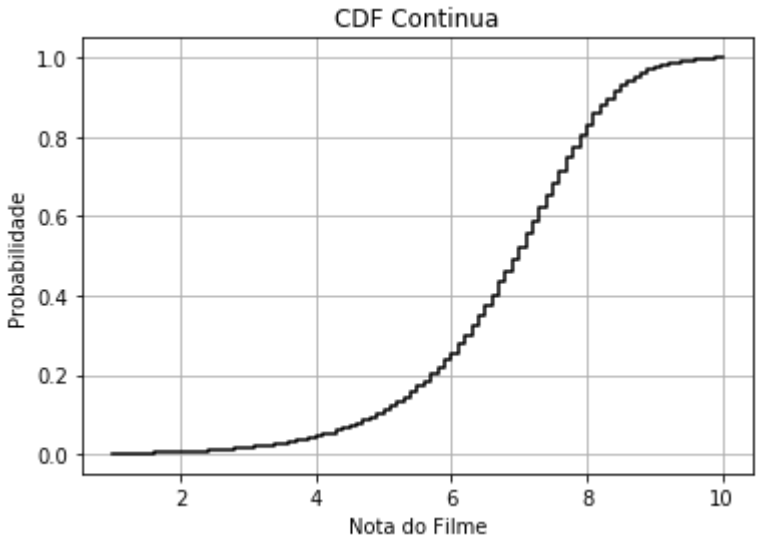
Eixo X: [1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4.0, 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5.0, 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6.0, 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9, 7.0, 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.0, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 9.0, 9.1, 9.2, 9.3, 9.4, 9.5, 9.6, 9.7, 9.8, 9.9, 10.0] [0.0009894341111498389, 0.0012316819255532602, 0.0014882397374319 89, 0.0017335539798658335, 0.002026908928109639, 0.0023212860190302523, 0.0027505859432894798, 0.0031543323006285153, 0.0036654036390323576, 0.0041233233558242201, 0.004813269865087388, 0.0054705076062747295, 0.0062442696126181 47, 0.007023142332345604, 0.007936937885411674, 0.0088537998665 08168, 0.00988616397008393, 0.010922616644366922, 0.012437432091395912, 0.013623117596492826, 0.015266723020799584, 0.016741674903433073, 0.0187655174035122 9, 0.020573687798785083, 0.02291950524205872, 0.025067027006031667, 0.027959690781397414, 0.03054468961104405, 0.03405677184855525, 0.03681

```
962350396643, 0.041013474906908355, 0.04451329143229787, 0.049353137006982
25, 0.053613427683916685, 0.058969455310388955, 0.0640689251249825, 0.0708
7026249646083, 0.07708080140074433, 0.08526407567126666, 0.091976486629862
73, 0.10206401270727777, 0.11040776337805891, 0.12174843637724017, 0.13203
834670466313, 0.14432041310918428, 0.15592582106165873, 0.1702654606745937
5, 0.1838374711372462, 0.20171168012658217, 0.21677090818398978, 0.2378454
4589441062, 0.2557063670289481, 0.27987390847938903, 0.3008339662100074,
0.3257722032387613, 0.3489216905831018, 0.3761352172104296, 0.401470045597
7849, 0.4329959921785643, 0.4589297961745289, 0.4930070108766203, 0.521662
7808209238, 0.5573161395306526, 0.5873364699484943, 0.62059903693717, 0.65
0410850248943, 0.6840413886012694, 0.7137642755001601, 0.7478833980520005,
0.7744100448005136, 0.8056886328535056, 0.830592117031248, 0.8578117765146
367, 0.8777333372856184, 0.8971663138570861, 0.9130667653375064, 0.9285675
59031295, 0.9410765411100674, 0.9532195961105427, 0.9611452904265095, 0.96
99612710139758, 0.9752069072313528, 0.9811884861760314, 0.984677059131976,
0.9881390563783237, 0.9902456924352243, 0.9928756655426505, 0.994454875978
3184, 0.9963509506437966, 0.9968681548382613, 1.0]
```

2.2 Plotando

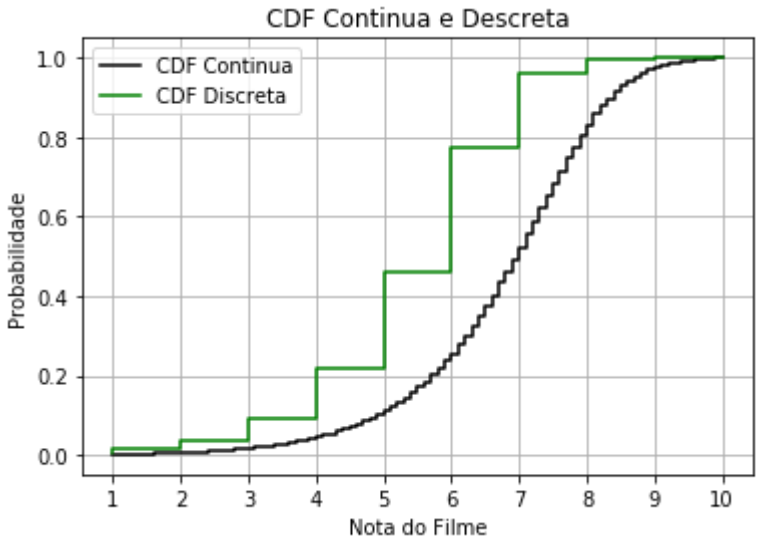
In [16]:

```
"""2.2 Plotando"""
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('CDF Continua')
plt.grid(True)
plt.step(eixoXC, ValorEixoXC, color='k')
plt.show()

ValorEixoXC1 = ValorEixoXC
```



In [17]:

```
"""3 Plotando"""
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('CDF Continua e Descreta')
plt.grid(True)
plt.step(eixoXC, ValorEixoXC, color='k',label='CDF Continua')
plt.xticks(eixoXD)
plt.step(eixoXD, ValorEixoXD, color='g',label='CDF Discreta')
plt.legend(loc='best')
plt.show()
```



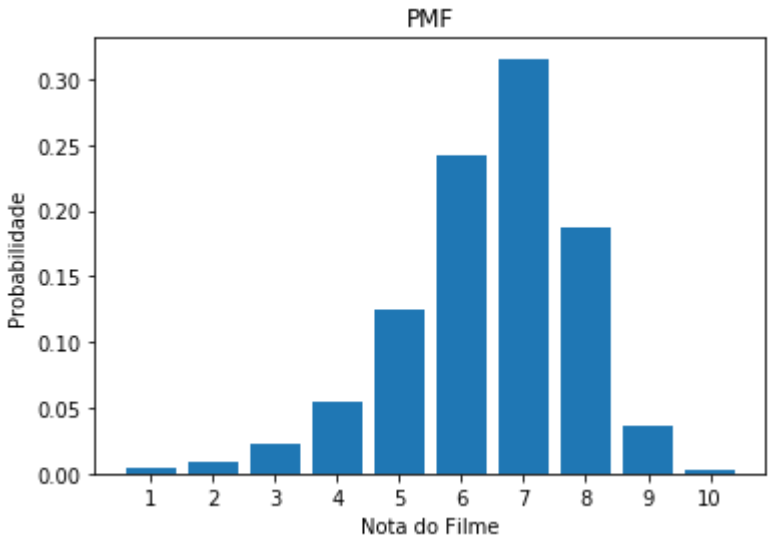
PMF

Calculando e Reaproveitando os cálculos da CDF

In [9]:

```
# Criando tabela de relação nota PMF
data = {'Nota':eixoXD, 'PMF':probabilidadeD}
pmf = pd.DataFrame(data)

print("PMF por nota")
print(pmf)

"""Plotando"""
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('PMF')
plt.xticks(eixoXD)
plt.bar(eixoXD, probabilidadeD,label='PMF')
plt.show()
```

```
PMF por nota
   Nota       PMF
0     1  0.004123
1     2  0.009500
2     3  0.023197
3     4  0.055157
4     5  0.124794
5     6  0.242159
6     7  0.315480
7     8  0.186735
8     9  0.035723
9    10  0.003132
```



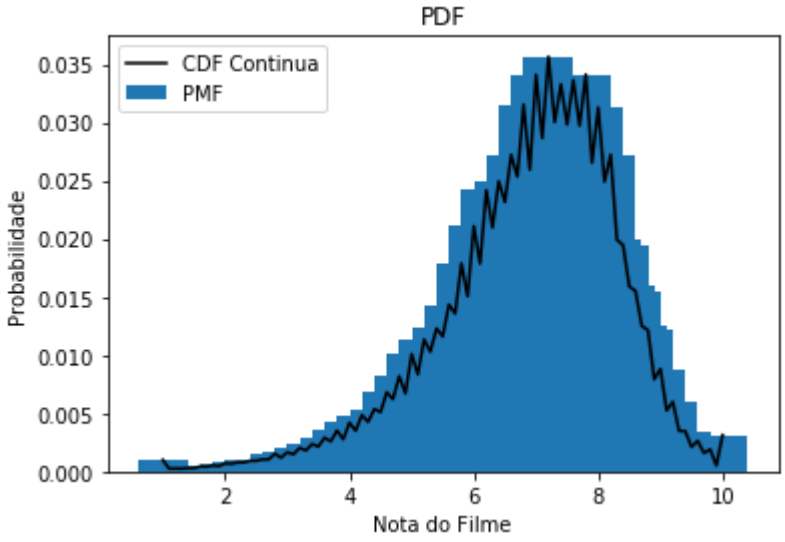
PDF

Calculando e Reaproveitando os cálculos da CDF

In [10]:

```
# Criando tabela de relação nota PMF
data = {'Nota':eixoXC, 'PDF':probabilidadeC}
pmf = pd.DataFrame(data)

print("PDF por nota")
print(pmf.head())

"""Plotando"""
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('PDF')
plt.bar(eixoXC, probabilidadeC, label='PMF')
plt.plot(eixoXC, probabilidadeC, color='k',label='CDF Continua')
plt.legend(loc='best')
plt.show()
```

```
PDF por nota
   Nota       PDF
0   1.0  0.000989
1   1.1  0.000242
2   1.2  0.000257
3   1.3  0.000245
4   1.4  0.000293
```



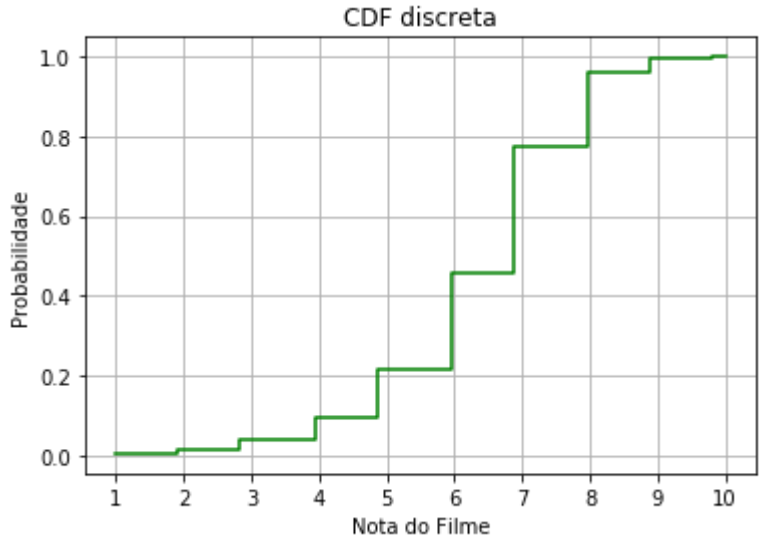
Parte 2

Plotagem de gráficos usando bibliotecas prontas

CDF discreta

In [11]:

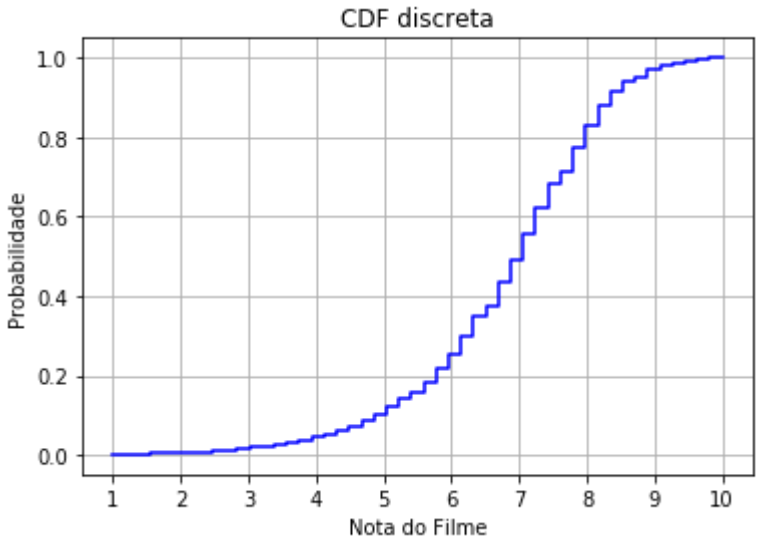
```
"""CDF discreta"""
import statsmodels.api as sm
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.distributions.empirical_distribution import ECDF
sample = Xd
ecdf=ECDF(sample)
x = np.linspace(min(sample), max(sample))
y = ecdf(x)
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('CDF discreta')
plt.xticks(eixoXD)
plt.step(x,y,color='g')
plt.grid(True)
plt.show()
```



CDF Contínua

In [12]:

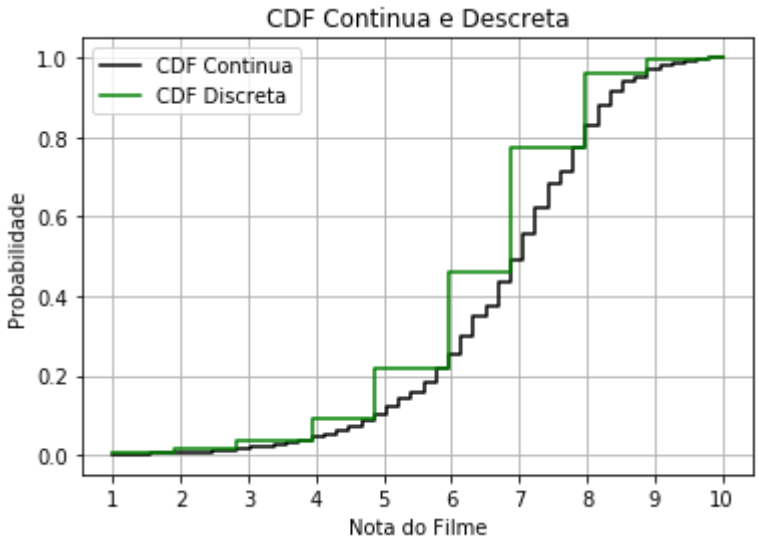
```
import statsmodels.api as sm
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.distributions.empirical_distribution import ECDF
sample = Xc
ecdf=ECDF(sample)
# ecdf = sm.distributions.ECDF(sample)
x1 = np.linspace(min(sample), max(sample))
y1 = ecdf(x)
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('CDF discreta')
plt.xticks(eixoXD)
plt.step(x1,y1,color='b')
plt.grid(True)
plt.show()
```



Comparação entre discreto e contínuo

In [13]:

```
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('CDF Continua e Descreta')
plt.grid(True)
plt.step(x1, y1, color='k',label='CDF Continua')
plt.xticks(eixoXD)
plt.step(x, y, color='g',label='CDF Discreta')
plt.legend(loc='best')
plt.show()
```



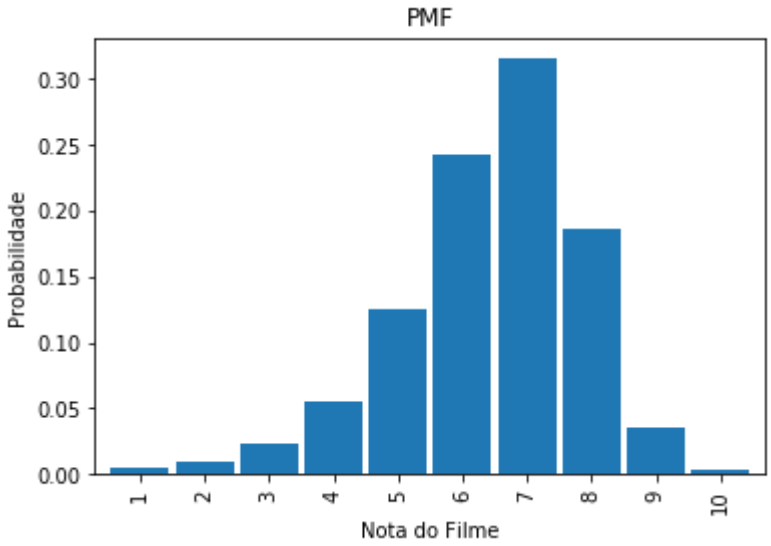
PMF

In [14]:

```
"""PMF"""
sample=Xd
pmf = sample.value_counts().sort_index() / len(sample)
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('PMF')
plt.xticks(eixoXD,rotation=90)
pmf.plot(kind="bar",width=0.9)
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efbe60d8b00>
```



PDF

In [16]:

```
import seaborn as sns
x = Xc
sns.set_style('white')
plt.xlabel('Nota do Filme')
plt.ylabel('Probabilidade')
plt.title('PDF')
sns.distplot(x)
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efbe29646a0>
```