Exercicios29.05.2017

June 11, 2017

0.0.1 Exercício1

- A "classificação" de uma palavra é a sua posição em uma lista de palavras classificadas por frequência: a palavra mais comum tem a classificação 1, a segunda mais comum é 2 etc.
- A lei de Zipf descreve a relação entre classificações e frequências das palavras em linguagens naturais (http://en.wikipedia.org/wiki/Zipf's_law). Ela prevê especificamente que a frequência, f, da palavra com classificação $ r $ é:

$$f = crs$$

- onde s e c são parâmetros que dependem do idioma e do texto. Se você tomar o logaritmo de ambos os lados desta equação, obtemos:

$$\log f = \log cs \log r$$

- Se você traçar o log de f contra o log de r, terá uma linha reta com uma elevação -s e interceptar o log de c.

- Escreva um programa que leia um texto em um arquivo, conte as frequências das palavras e exiba uma linha para cada palavra, em ordem descendente da frequência, com log de f e log de r. Use o programa gráfico de sua escolha para traçar os resultados e verifique se formam uma linha reta. Você pode estimar o valor de s?

```
In [1]: %%writefile Hemingway.txt
        A Very Short Story by Ernest Hemingway
        One hot evening in Padua they carried him up onto the roof and he could look out over th
        of the town There were chimney swifts in the sky After a while it got dark and the searc
        came out The others went down and took the bottles with them He and Luz could hear them
        below on the balcony Luz sat on the bed She was cool and fresh in the hot night.
        Luz stayed on night duty for three months They were glad to let her When they operated o
        him she prepared him for the operating table and they had a joke about friend or enema H
        went under the anaesthetic holding tight on to himself so he would not blab about anythi
        during the silly talky time After he got on crutches he used to take the temperatures so
        would not have to get up from the bed There were only a few patients and they all knew a
        it They all liked Luz As he walked back along the halls he thought of Luz in his bed
        Before he went back to the front they went into the Duomo and prayed It was dim and quie
        and there were other people praying They wanted to get married but there was not enough
        time for the banns and neither of them had birth certificates They felt as though they w
```

```
        married but they wanted everyone to know about it and to make it so they could not lose
        Luz wrote him many letters that he never got until after the armistice Lifteen came in a
        to the front and he sorted them by the dates and read them all straight through They wer
        about the hospital and how much she loved him and how it was impossible to get along
        without him and how terrible it was missing him at night
        After the armistice they agreed he should go home to get a job so they might be married
        would not come home until he had a good job and could come to New York to meet her It wa
        understood he would not drink and he did not want to see his friends or anyone in the St
        Only to get a job and be married On the train from Padua to Milan they quarreled about h
        being willing to come home at once When they had to say good-bye in the station at Milan
        they kissed good-bye but were not finished with the quarrel. He felt sick about saying g
        bye like that
        He went to America on a boat from Genoa Luz went back to Pordonone to open a hospital It
        was lonely and rainy there and there was a battalion of arditi quartered in the town Liv
        the muddy, rainy town in the winter the major of the battalion made love to Luz and she
        never known Italians before and finally wrote to the States that theirs had only been a
        girl affair She was sorry and she knew he would probably not be able to understand but m
        some day forgive her and be grateful to her and she expected absolutely unexpectedly to
        married in the spring She loved him as always but she realized now it was only a boy and
        love She hoped he would have a great career and believed in him absolutely She knew it w
        for the best
        The major did not marry her in the spring or any other time Luz never got an answer to t
        letter to Chicago about it A short time after he contracted gonorrhea from a sales girl
        department store while riding in a taxicab through Lincoln Park

Overwriting Hemingway.txt
```

```
In [2]: file=open("Hemingway.txt","r")

In [3]: texto=file.readlines()

In [4]: dict={}
        for line in texto:
            for word in line.split():
                if word in dict:
                    dict[word]=dict[word]+1
                else:
                    dict[word]=1

In [5]: dict

Out[5]: {'A': 2,
         'After': 3,
         'America': 1,
         'As': 1,
         'Before': 1,
         'Chicago': 1,
         'Duomo': 1,
```

```
'Ernest': 1,
'Genoa': 1,
'He': 4,
'Hemingway': 1,
'It': 3,
'Italians': 1,
'Lifteen': 1,
'Lincoln': 1,
'Living': 1,
'Luz': 11,
'Milan': 1,
'Milan,': 1,
'New': 1,
'On': 1,
'One': 1,
'Only': 1,
'Padua': 2,
'Park': 1,
'Pordonone': 1,
'She': 5,
'Short': 1,
'States': 2,
'Story': 1,
'The': 2,
'There': 2,
'They': 5,
'Very': 1,
'When': 2,
'York': 1,
'a': 16,
'able': 1,
'about': 8,
'absolutely': 2,
'affair': 1,
'after': 2,
'agreed': 1,
'all': 4,
'along': 2,
'always': 1,
'an': 1,
'anaesthetic': 1,
'and': 30,
'answer': 1,
'any': 1,
'anyone': 1,
'anything': 1,
'arditi': 1,
'armistice': 2,
```

```
'as': 2,
'at': 3,
'back': 3,
'balcony': 1,
'banns': 1,
'battalion': 2,
'be': 5,
'bed': 3,
'been': 1,
'before': 1,
'being': 1,
'believed': 1,
'below': 1,
'best': 1,
'birth': 1,
'blab': 1,
'boat': 1,
'bottles': 1,
'boy': 2,
'bunch': 1,
'but': 5,
'by': 2,
'bye': 1,
'came': 2,
'career': 1,
'carried': 1,
'certificates': 1,
'chimney': 1,
'come': 3,
'contracted': 1,
'cool': 1,
'could': 4,
'crutches': 1,
'dark': 1,
'dates': 1,
'day': 1,
'department': 1,
'did': 2,
'dim': 1,
'down': 1,
'drink': 1,
'during': 1,
'duty': 1,
'enema': 1,
'enough': 1,
'evening': 1,
'everyone': 1,
'expected': 1,
```

```
'felt': 2,
'few': 1,
'finally': 1,
'finished': 1,
'for': 4,
'forgive': 1,
'fresh': 1,
'friend': 1,
'friends': 1,
'from': 4,
'front': 2,
'get': 5,
'girl': 3,
'glad': 1,
'go': 1,
'gonorrhea': 1,
'good': 1,
'good-': 1,
'good-bye': 2,
'got': 4,
'grateful': 1,
'great': 1,
'had': 6,
'halls': 1,
'have': 2,
'he': 16,
'hear': 1,
'her': 6,
'him': 9,
'himself': 1,
'his': 2,
'holding': 1,
'home': 3,
'hoped': 1,
'hospital': 2,
'hot': 2,
'how': 3,
'impossible': 1,
'in': 15,
'into': 1,
'it': 10,
'job': 3,
'joke': 1,
'kissed': 1,
'knew': 3,
'know': 1,
'known': 1,
'let': 1,
```

```
'letter': 1,
'letters': 1,
'like': 1,
'liked': 1,
'lonely': 1,
'look': 1,
'loop': 1,
'lose': 1,
'love': 2,
'loved': 2,
'made': 1,
'major': 2,
'make': 1,
'many': 1,
'married': 5,
'marry': 1,
'meet': 1,
'might': 2,
'missing': 1,
'months': 1,
'much': 1,
'muddy,': 1,
'neither': 1,
'never': 3,
'night': 2,
'night.': 1,
'not': 11,
'now': 1,
'of': 5,
'on': 7,
'once': 1,
'only': 3,
'onto': 1,
'open': 1,
'operated': 1,
'operating': 1,
'or': 3,
'other': 2,
'others': 1,
'out': 2,
'over': 1,
'patients': 1,
'people': 1,
'prayed': 1,
'praying': 1,
'prepared': 1,
'probably': 1,
'quarrel.': 1,
```

```
'quarreled': 1,
'quartered': 1,
'quiet,': 1,
'rainy': 2,
'read': 1,
'realized': 1,
'riding': 1,
'roof': 1,
'sales': 1,
'sat': 1,
'say': 1,
'saying': 1,
'searchlights': 1,
'see': 1,
'she': 6,
'short': 1,
'should': 1,
'sick': 1,
'silly': 1,
'sky': 1,
'so': 4,
'some': 1,
'sorry': 1,
'sorted': 1,
'spring': 2,
'station': 1,
'stayed': 1,
'store': 1,
'straight': 1,
'swifts': 1,
'table': 1,
'take': 1,
'talky': 1,
'taxicab': 1,
'temperatures': 1,
'terrible': 1,
'that': 3,
'the': 37,
'theirs': 1,
'them': 5,
'there': 4,
'they': 13,
'though': 1,
'thought': 1,
'three': 1,
'through': 2,
'tight': 1,
'time': 4,
```

```
 'to': 28,
 'took': 1,
 'top': 1,
 'town': 3,
 'train': 1,
 'under': 1,
 'understand': 1,
 'understood': 1,
 'unexpectedly': 1,
 'until': 2,
 'up': 2,
 'used': 1,
 'walked': 1,
 'want': 1,
 'wanted': 2,
 'was': 11,
 'went': 6,
 'were': 7,
 'while': 2,
 'willing': 1,
 'winter': 1,
 'with': 2,
 'without': 1,
 'would': 6,
 'wrote': 2}
```

```
In [6]: import numpy as np
```

```
In [7]: oco=dict.values()
        oco.sort()
        oco.reverse()
        X=np.array(range(len(oco)))+1
        Y=np.array(oco)
```

```
In [8]: oco
```

```
Out[8]: [37,
         30,
         28,
         16,
         16,
         15,
         13,
         11,
         11,
         11,
         10,
         9,
         8,
```

```
7,
7,
6,
6,
6,
6,
6,
5,
5,
5,
5,
5,
5,
5,
5,
4,
4,
4,
4,
4,
4,
4,
4,
4,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
3,
2,
2,
2,
2,
2,
2,
2,
2,
```

2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
2,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,

```
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
```

```
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
```

```
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
1,
```

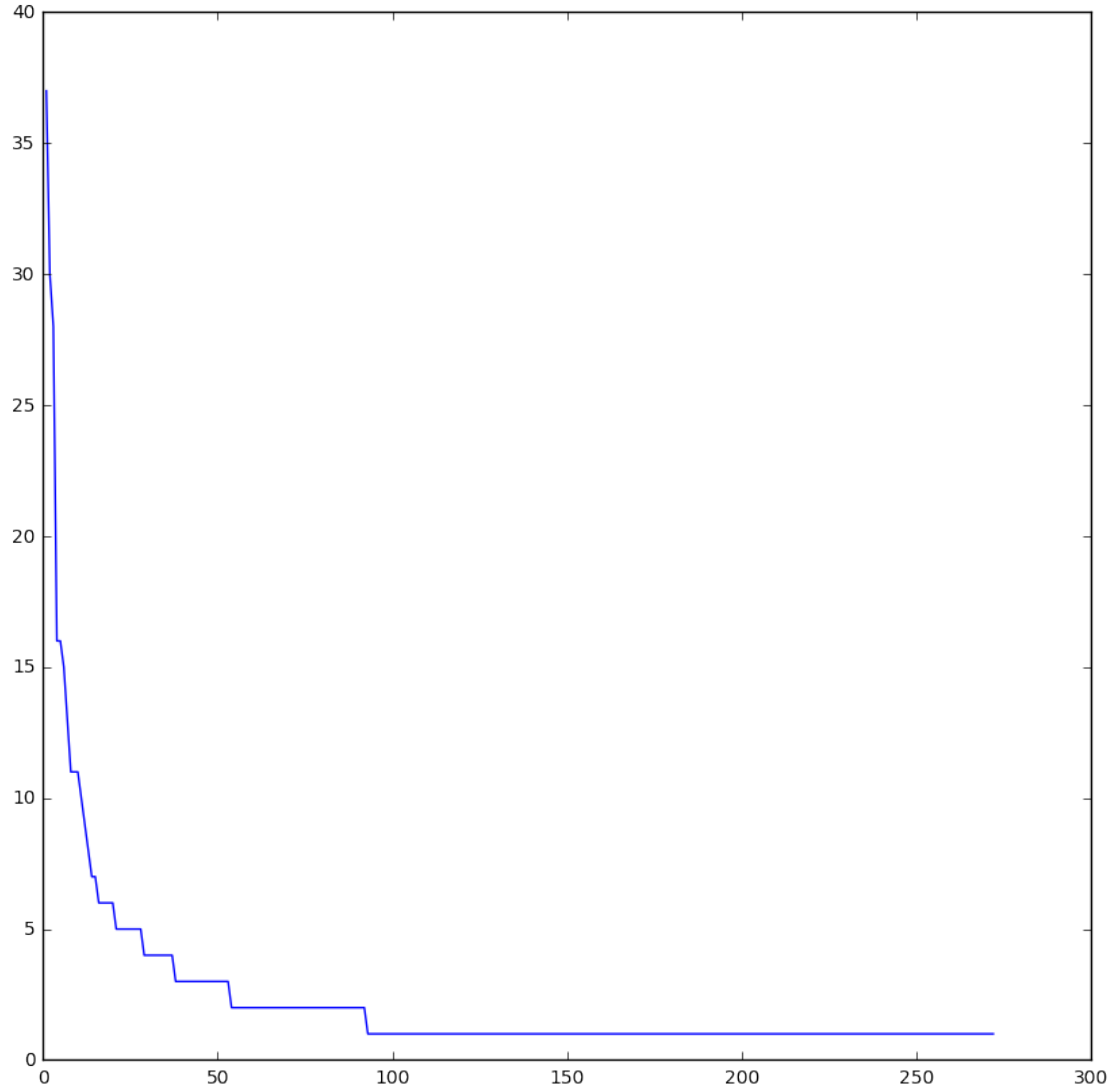
```
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1,
 1]
```

```
In [9]: import matplotlib.pyplot as plt
        fig = plt.figure(figsize=(10, 10)) # define área do plot
        ax = fig.gca() # define eixo
        x=fig.gca()
        plt.plot(X,Y)
        ax.set_title('Frequência Hemingway') # Give the plot a main title
        ax.set_xlabel('palavras') # Set text for the x axis
        ax.set_ylabel('Frequência')# Set text for y axis
```

```
        ---------------------------------------------------------------------------

        UnicodeDecodeError                        Traceback (most recent call last)

        <ipython-input-9-b296f39bdbc7> in <module>()
          4 x=fig.gca()
          5 plt.plot(X,Y)
    ----> 6 ax.set_title('Frequência Hemingway') # Give the plot a main title
          7 ax.set_xlabel('palavras') # Set text for the x axis
          8 ax.set_ylabel('Frequência')# Set text for y axis

        /projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/axes/_axes.py in se
        170             'verticalalignment': 'baseline',
        171             'horizontalalignment': loc.lower()}
    --> 172         title.set_text(label)
        173         title.update(default)
        174         if fontdict is not None:
```

```
/projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/text.py in set_text
   1204         ACCEPTS: string or anything printable with '%s' conversion.
   1205         """
-> 1206         self._text = '%s' % (s,)
   1207         self.stale = True
   1208

UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 5: ordinal not in r
```

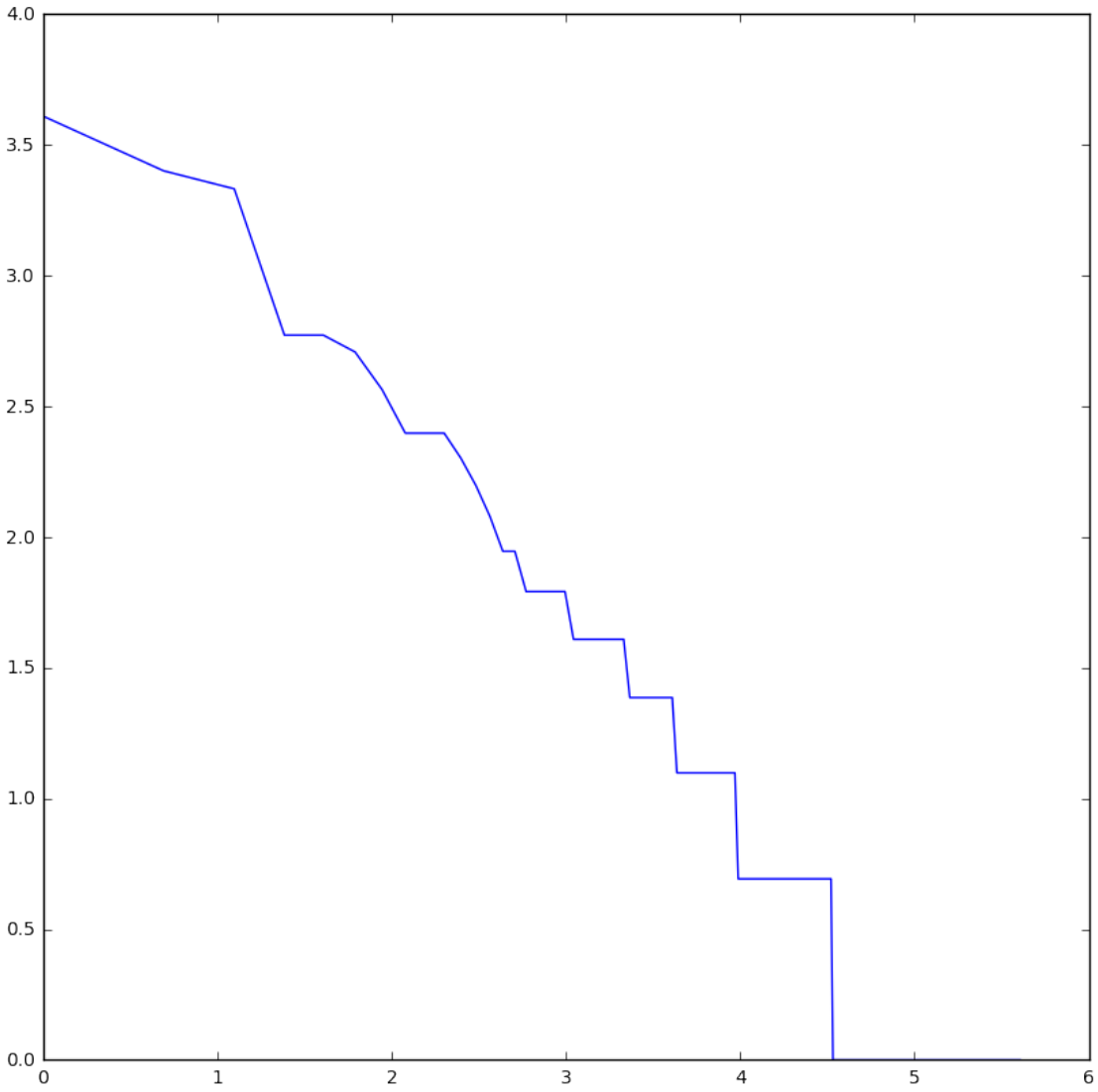
Out[9]:



In [10]:
```
logx=np.log(X)
logy=np.log(Y)
fig = plt.figure(figsize=(10, 10)) # define área do plot
ax = fig.gca() # define eixo
x=fig.gca()
plt.plot(logx,logy)
ax.set_title('Frequência Hemingway') # Give the plot a main title
ax.set_xlabel('palavras') # Set text for the x axis
ax.set_ylabel('Frequência')# Set text for y axis
```

```
---------------------------------------------------------------------------

UnicodeDecodeError                        Traceback (most recent call last)

<ipython-input-10-beabee0d2965> in <module>()
  5 x=fig.gca()
  6 plt.plot(logx,logy)
----> 7 ax.set_title('Frequência Hemingway') # Give the plot a main title
  8 ax.set_xlabel('palavras') # Set text for the x axis
  9 ax.set_ylabel('Frequência')# Set text for y axis


/projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/axes/_axes.py in se
170             'verticalalignment': 'baseline',
171             'horizontalalignment': loc.lower()}
--> 172         title.set_text(label)
173         title.update(default)
174         if fontdict is not None:


/projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/text.py in set_text
1204         ACCEPTS: string or anything printable with '%s' conversion.
1205         """
-> 1206         self._text = '%s' % (s,)
1207         self.stale = True
1208


UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 5: ordinal not in r
```

Out[10]:



0.0.2 Exercício 2

Agora que você já viu como criar alguns plots simples, é sua vez de realizar uma visualização. Crie o seguinte gráfico de dispersão:

- Traçar o tamanho do motor contra o preço.
- Defina o tamanho da figura como 8 x 8.
- Forneça um título significativo, rótulo do eixo x e rótulo do eixo y.

```
In [11]: def read_auto_data(fileName = "Automobile price data.csv"):
             'Function to load the auto price data set from a .csv file'
             import pandas as pd
             import numpy as np
```

```
        ## Read the .csv file with the pandas read_csv method
        auto_prices = pd.read_csv(fileName)

        ## Remove rows with missing values, accounting for mising values coded as '?'
        cols = ['price', 'bore', 'stroke',
              'horsepower', 'peak-rpm']
        for column in cols:
            auto_prices.loc[auto_prices[column] == '?', column] = np.nan
        auto_prices.dropna(axis = 0, inplace = True)

        ## Convert some columns to numeric values
        for column in cols:
            auto_prices[column] = pd.to_numeric(auto_prices[column])
#     auto_prices[cols] = auto_prices[cols].as_type(int64)

        return auto_prices
    auto_prices = read_auto_data()
```

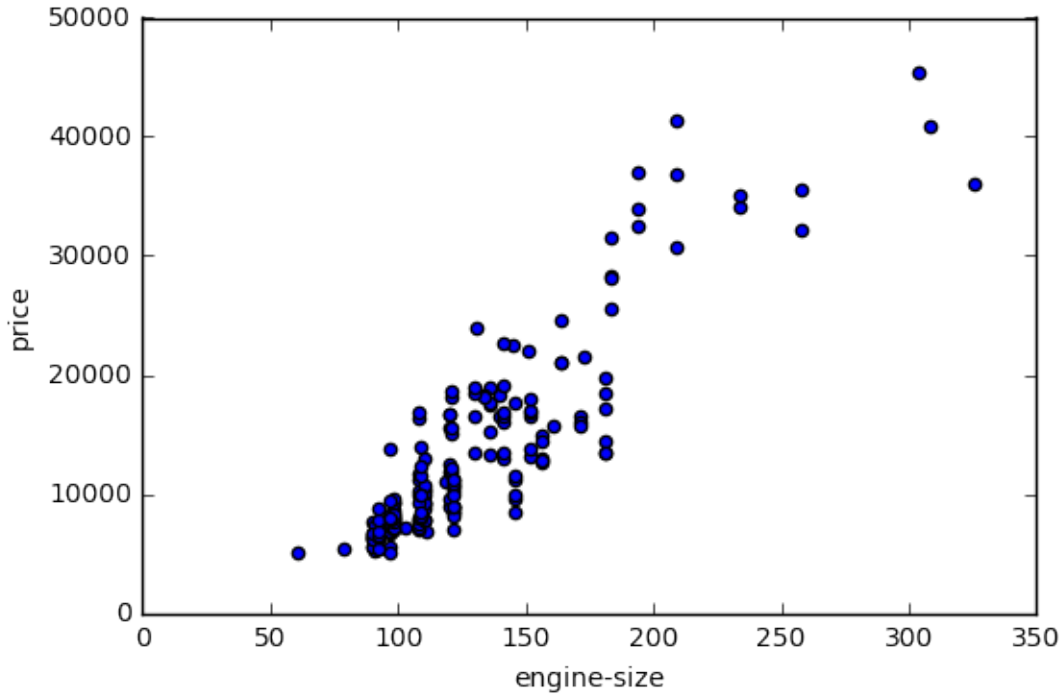
```
In [12]: auto_prices.head()
```

```
In [13]: auto_prices.describe()
```

```
In [14]: %matplotlib inline
         auto_prices.plot(kind = 'scatter', x = 'engine-size', y = 'price')
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe026830a50>
```

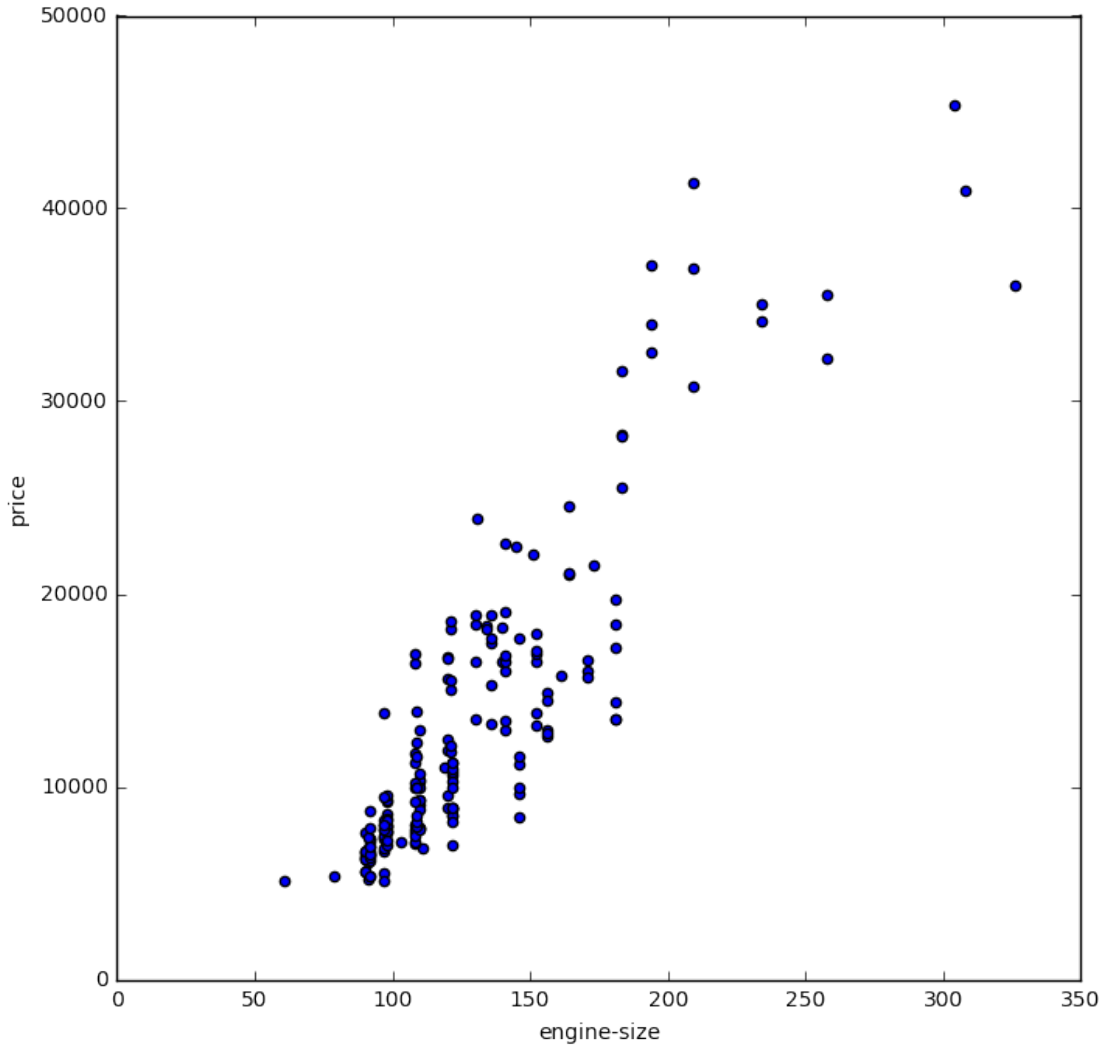
Out[14]:



In [15]:
```
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(8, 8)) # define plot area
ax = fig.gca() # define axis
auto_prices.plot(kind = 'scatter', x = 'engine-size', y = 'price', ax = ax)
ax.set_title('Scatter plot de preço vs tamanho do motor') # Give the plot a main title
ax.set_xlabel('tamanho do motor') # Set text for the x axis
ax.set_ylabel('Preço (US$)')# Set text for y axis
```

```
---------------------------------------------------------------------------

UnicodeDecodeError                        Traceback (most recent call last)

<ipython-input-15-335f291d29c7> in <module>()
  3 ax = fig.gca() # define axis
  4 auto_prices.plot(kind = 'scatter', x = 'engine-size', y = 'price', ax = ax)
----> 5 ax.set_title('Scatter plot de preço vs tamanho do motor') # Give the plot a main tit
  6 ax.set_xlabel('tamanho do motor') # Set text for the x axis
  7 ax.set_ylabel('Preço (US$)')# Set text for y axis


/projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/axes/_axes.py in se
170             'verticalalignment': 'baseline',
171             'horizontalalignment': loc.lower()}
--> 172         title.set_text(label)
173         title.update(default)
174         if fontdict is not None:


/projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/text.py in set_text
1204         ACCEPTS: string or anything printable with '%s' conversion.
1205         """
-> 1206         self._text = '%s' % (s,)
1207         self.stale = True
1208


UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 19: ordinal not in
```

Out[15]:



0.1 Exercício 3

- Faça um gráfico de Barras com os dados contidos no DataFrame do exercício dos epsódios do Pokemon. Represente o número de episódios para cada Temporada. Neste exercicio o gráfico deve conter legenda, titulo, nome dos eixos e cada barra deve conter uma cor diferente.

```
In [16]: import pandas as pd
```

```
In [17]: dict=[{'Série':'Série Original' , 'Geração':'Primeira' , 'Temporada':'Liga Índigo', 'Pr
                 {'Série': 'Série Original' , 'Geração':'Primeira' , 'Temporada':'Aventuras na Ilh
               {'Série':'Séria Original' , 'Geração':'Segunda' , 'Temporada':'A Jornada de Johto'
               {'Série':'Séria Original' , 'Geração':'Segunda' , 'Temporada':'Campeões da Liga de
               {'Série':'Séria Original' , 'Geração':'Segunda' , 'Temporada':'Master Quest' , 'Pr
               {'Série':'Geração Avançada' , 'Geração':'Terçeira' , 'Temporada':'Pokémon: Avançad
```

```
        {'Série':'Geração Avançada' , 'Geração':'Terçeira' , 'Temporada':'Desafio Avançado
        {'Série':'Geração Avançada' , 'Geração':'Terçeira' , 'Temporada':'Batalha Avançada
        {'Série':'Geração Avançada' , 'Geração':'Terçeira' , 'Temporada':'Batalha da Front
        {'Série':'Diamante e Pérola' , 'Geração':'Qarta' , 'Temporada':'Diamante e Pérola'
        {'Série':'Diamante e Pérola' , 'Geração':'Qarta' , 'Temporada':'Batalha Dimensiona
        {'Série':'Diamante e Pérola' , 'Geração':'Qarta' , 'Temporada':'Batalha Glácticas'
        {'Série':'Diamante e Pérola' , 'Geração':'Qarta' , 'Temporada':'Vencedores da Liga
        {'Série':'Preto e Branco' , 'Geração':'Quinta' , 'Temporada':'Preto e Branco' , 'P
        {'Série':'Preto e Branco' , 'Geração':'Quinta' , 'Temporada':'Destinos Rivais' , '
        {'Série':'Preto e Branco' , 'Geração':'Quinta' , 'Temporada':'Aventuras em Unova (
        {'Série':'XY' , 'Geração':'Sexta' , 'Temporada':'A Série XY' , 'Primeiro Episódio'
        {'Série':'XY' , 'Geração':'Sexta' , 'Temporada':'Kalos Quest' , 'Primeiro Episódio
        {'Série':'XY' , 'Geração':'Sexta' , 'Temporada':'XY e Z' , 'Primeiro Episódio':'89
        {'Série':'Sun and Moon' , 'Geração':'Sétima' , 'Temporada':'Sun and Moon' , 'Prime
df=pd.DataFrame(dict)
df[['Série' , 'Geração' , 'Temporada' , 'Primeiro Episódio' , 'Último Episódio', 'Regiã
```

In [18]:
```
dict1=[{'Diferença':int(84),'Temporada':'Liga Índigo'},{'Diferença':int(34),'Temporada'
       {'Diferença':int(51),'Temporada':'Desafio Avançado'},{'Diferença':int(52),'Tempo
       {'Diferença':int(33),'Temporada':'Vencedores da Liga Sinnoh'},{'Diferença':int(4
       {'Diferença':int(48),'Temporada':'Kalos Quest'},{'Diferença':int(1),'Temporada':
df1=pd.DataFrame(dict1)
df1[['Diferença','Temporada']]
```

In [19]:
```
import matplotlib.pyplot as plt
```

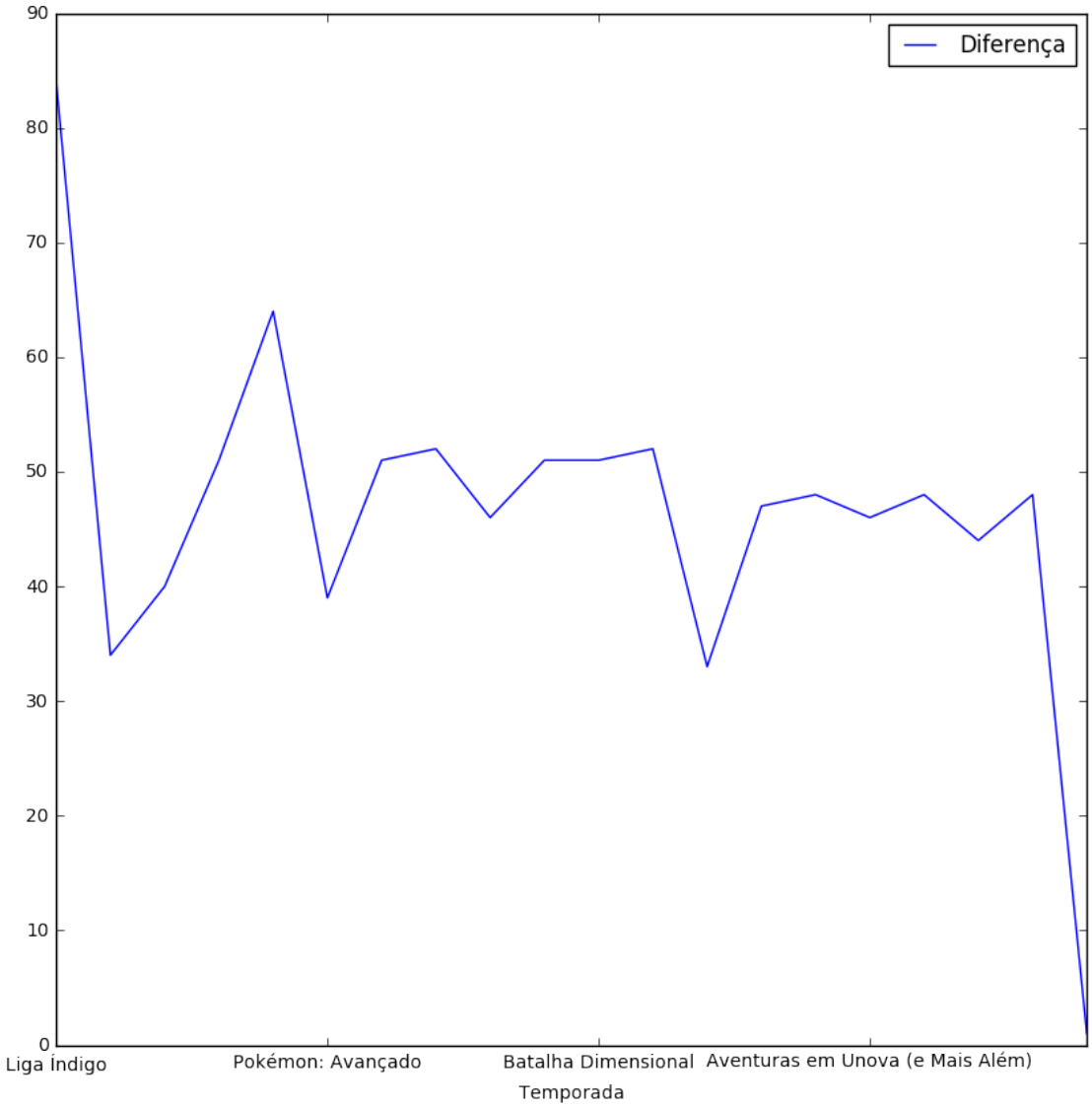
In [20]:
```
df1.describe()
```

In [21]:
```
fig = plt.figure(figsize=(10, 10)) # define área do plot
ax = fig.gca() # define eixo
df1.plot(x = 'Temporada', y = 'Diferença', ax = ax) ## linha é o formato padrão
ax.set_title('Pokémon') # Título Principal
ax.set_xlabel('Temporadas') # Eixo x
ax.set_ylabel('Número de Epsódios')# Eixo y
```

```
---------------------------------------------------------------------------

UnicodeDecodeError                        Traceback (most recent call last)

<ipython-input-21-c76d82fa5eef> in <module>()
  2 ax = fig.gca() # define eixo
  3 df1.plot(x = 'Temporada', y = 'Diferença', ax = ax) ## linha é o formato padrão
----> 4 ax.set_title('Pokémon') # Título Principal
  5 ax.set_xlabel('Temporadas') # Eixo x
  6 ax.set_ylabel('Número de Epsódios')# Eixo y


/projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/axes/_axes.py in se
```

```
    170             'verticalalignment': 'baseline',
    171             'horizontalalignment': loc.lower()}
--> 172         title.set_text(label)
    173         title.update(default)
    174         if fontdict is not None:


/projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/text.py in set_text
   1204         ACCEPTS: string or anything printable with '%s' conversion.
   1205         """
-> 1206         self._text = '%s' % (s,)
   1207         self.stale = True
   1208

UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 3: ordinal not in r
```

Out[21]:

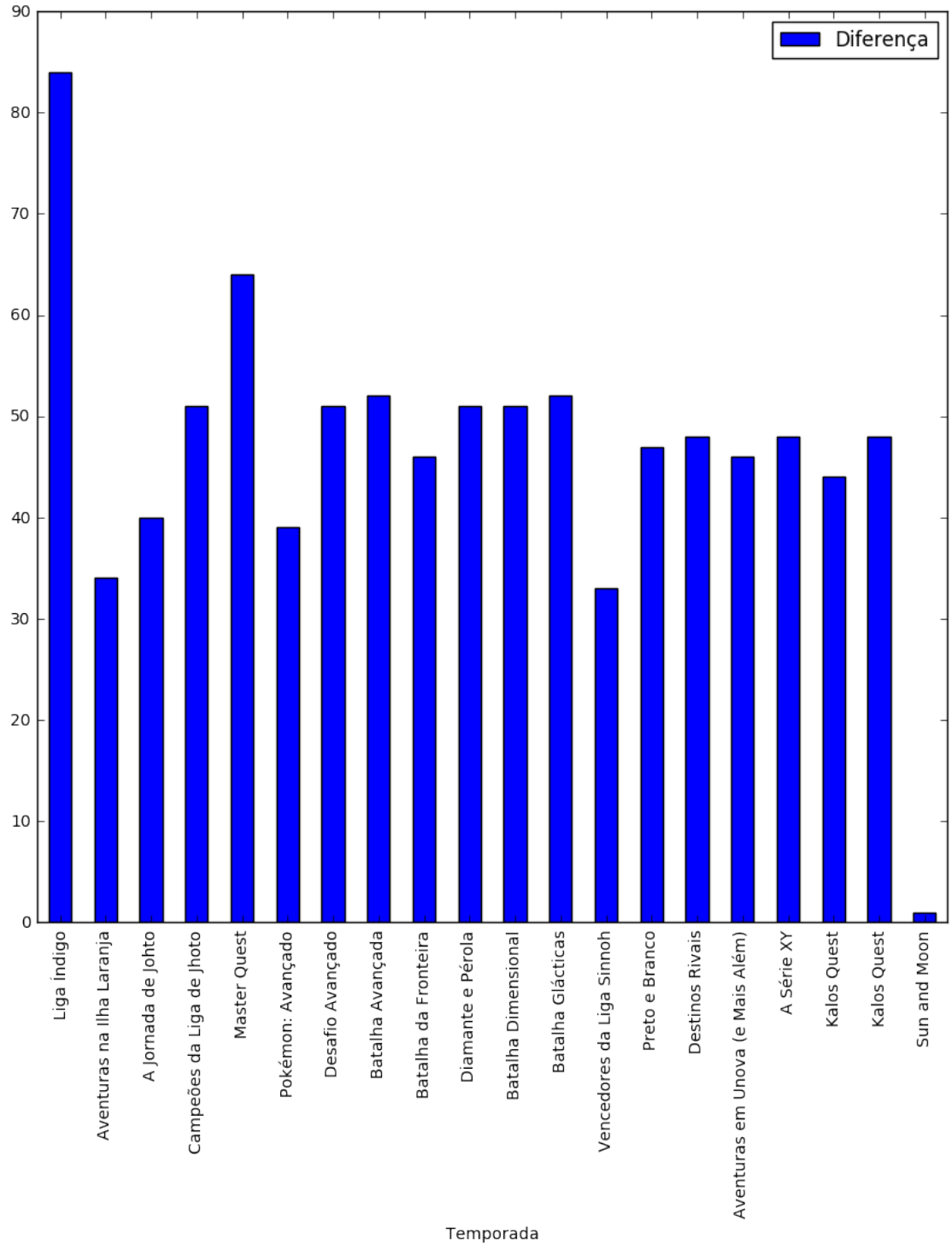



```
In [22]: fig = plt.figure(figsize=(10,10)) # define plot area
         ax = fig.gca() # define axis
         df1.plot.bar(x = 'Temporada', y='Diferença', ax = ax) # Use the plot.bar method on the
         ax.set_title('Pokémon') # Give the plot a main title
         ax.set_xlabel('Temporadas') # Set text for the x axis
         ax.set_ylabel('Número de Epsódios')# Set text for y axis
```

```
---------------------------------------------------------------------------

UnicodeDecodeError                        Traceback (most recent call last)

<ipython-input-22-afad881bdf4c> in <module>()
```

```
      2 ax = fig.gca() # define axis
      3 df1.plot.bar(x = 'Temporada', y='Diferença', ax = ax) # Use the plot.bar method on t
----> 4 ax.set_title('Pokémon') # Give the plot a main title
      5 ax.set_xlabel('Temporadas') # Set text for the x axis
      6 ax.set_ylabel('Número de Epsódios')# Set text for y axis


    /projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/axes/_axes.py in se
    170             'verticalalignment': 'baseline',
    171             'horizontalalignment': loc.lower()}
--> 172         title.set_text(label)
    173         title.update(default)
    174         if fontdict is not None:


    /projects/sage/sage-7.6/local/lib/python2.7/site-packages/matplotlib/text.py in set_text
   1204         ACCEPTS: string or anything printable with '%s' conversion.
   1205         """
-> 1206         self._text = '%s' % (s,)
   1207         self.stale = True
   1208


    UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 3: ordinal not in r
```

Out[22]:


Diferença
90
80
70
60
50
40
30
20
10
0
Liga Índigo
Aventuras na Ilha Laranja
A Jornada de Johto
Campeões da Liga de Jhoto
Master Quest
Pokémon: Avançado
Desafio Avançado
Batalha Avançada
Batalha da Fronteira
Diamante e Pérola
Batalha Dimensional
Batalha Galácticas
Vencedores da Liga Sinnoh
Preto e Branco
Destinos Rivais
Aventuras em Unova (e Mais Além)
A Série XY
Kalos Quest
Kalos Quest
Sun and Moon
Temporada

In [0]: