Usando os Diferentes Contadores

```
In [391]: import pandas as pd
In [392]: | s = pd.Series(list('asdeasdeasdeasdea'))
Out[392]: 0
                d
          10
                d
          11
                е
          12
                а
          13
                s
          14
                d
          15
                e
          16
          dtype: object
```

Verifica cada valor que aparece pelo menos uma vez

```
In [393]: s.unique()
Out[393]: array(['a', 's', 'd', 'e'], dtype=object)
```

Conta quantas vezes cada valor aparece

Usando os mesmos contadores para Dataframe

```
In [395]: dados = pd.read csv('../dados/aluguel residencial.csv', sep=';')
In [396]: dados.Tipo.unique()
Out[396]: array(['Quitinete', 'Casa', 'Apartamento', 'Casa de Condomínio',
                  'Casa de Vila'], dtype=object)
In [397]: dados.Tipo.value counts()
Out[397]: Apartamento
                                 19532
          Casa de Condomínio
                                   996
          Casa
                                   967
          Ouitinete
                                   836
          Casa de Vila
                                   249
          Name: Tipo, dtype: int64
```

Exercício

Suponha que estejamos testando as moedas que serão utilizadas por árbitros de futebol nas competições da Copa do Mundo. Atualmente, estas moedas são personalizadas e utilizadas em certos momentos dos jogos para decidir, na sorte, uma disputa. Nosso objetivo é verificar se as moedas que serão utilizadas nas competições não sejam viciadas.

Para isso testes foram realizados com cinco moedas e os resultados foram os seguintes:

Acima, temos o resultado de 50 lançamentos de cada moeda (m1, m2, m3, m4 e m5), onde c representa a ocorrência do evento CARA e C representa a ocorrência do evento COROA.

Para tirarmos nossas conclusões, precisamos montar o seguinte DataFrame:

obs: para carregar imagens e links no markdown não se usa áspas para passar o caminho

Onde Freq. C e Freq. c são, respectivamente, as frequências de COROAS e de CARAS em cada teste.

Qual o código necessário para realizar tal tarefa?

```
In [399]: | df = pd.DataFrame({'m1': list(m1),
                             'm2': list(m2),
                             'm3': list(m3),
                             'm4': list(m4),
                             'm5': list(m5)})
In [400]: | df = pd.DataFrame({'m1': dict(df.m1.value_counts()),
                                           'm2': dict(df.m2.value_counts()),
                                                       'm3': dict(df.m3.value_counts()),
                                                                   'm4': dict(df.m4.value co
           unts()),
                                                                               'm5': dict(df.
           m5.value_counts())})
           df
Out[400]:
                  m2 m3 m4 m5
              m1
                               25
               25
                   26
                       19
                           23
               25
                  24
                       31
                           27
                               25
In [401]: | df['Faces'] = ['Cara', 'Coroa']
In [402]: | df.sort_index(axis=1, inplace=True)
           df
Out[402]:
              Faces m1 m2 m3 m4 m5
               Cara
                     25
                         26
                              19
                                  23
                                      25
            С
                             31
             Coroa
                     25
                         24
                                  27
                                      25
In [403]:
           df.sort index(inplace=True)
Out[403]:
                        m2 m3 m4 m5
              Faces m1
           С
              Coroa
                     25
                         24
                             31
                                  27
                                      25
                Cara
                     25
                         26
                                      25
                             19
                                  23
```

Resolução proposta

```
In [406]: # para concatenar dataframes é necessário passar uma lista deles
          # de mesmo tamanho e index
          df = pd.DataFrame(data = ['Cara', 'Coroa'],
                              index = ['c', 'C'],
                               columns = ['Faces'])
          for item in moedas:
              df = pd.concat([df, moedas[item].value_counts()],
                               axis = 1
          df
Out[406]:
              Faces m1 m2 m3 m4 m5
               Cara
                    25
                        26
                            19
                                23
                                    25
```

Passo a passo

25

24

31

27 25

C Coroa

Concatenando lado a lado

```
In [408]:
          for i in moedas:
               df = pd.concat([df, moedas[i].value_counts()], axis=1)
           df
Out[408]:
              Faces m1 m2 m3 m4 m5
                             19
                     25
                         26
                                 23
                                     25
               Cara
           C Coroa
                     25
                        24
                             31
                                 27
                                     25
```

Utilizando outro index como exemplo

```
In [410]: events = pd.DataFrame({'n1': list(n1),
                                   'n2': list(n2),
                                   'n3': list(n3)})
In [411]: | df2 = pd.DataFrame(data = ['Maiúsc', 'Minúsc'], index = ['A', 'a'], columns =
           ['Letra'])
           df2
Out[411]:
               Letra
              Maiúsc
              Minúsc
In [412]:
          for lista in events:
               df2 = pd.concat([df2, events[lista].value_counts()], axis=1)
           df2
Out[412]:
               Letra n1 n2 n3
              Maiúsc
                    14
                         14
              Minúsc 17 17 23
```

- não importa a ordem que colocarmos o index
- o pandas relacionará automaticamente o index com o dicionário passado
- · repare o exemplo abaixo:

Repare agora como ficaria o dataframe se invertêssemos a oredem do index

A Maiúsc 14 14 8

a Minúsc 17 17 23

Podemos também reorganizar o index a qualquer momento

In [420]: # lembrando que para fixar o método, deve-se utilizar o inplace=True
no python a letra maiúscula sempre vem primeiro na ordem alfabética
df3.sort_index()

Out[420]:

	Letra	mı	112	IIS
Α	Maiúsc	14	14	8
а	Minúsc	17	17	23

In [419]: df3

Out[419]:

In [423]: df2

Out[423]:

	Letra	n1	n2	n3
Α	Maiúsc	14	14	8
а	Minúsc	17	17	23

In [424]: # df2 já esta na ordem crescente, portanto o método não tem efeito
df2.sort_index()

Out[424]:

 Letra
 n1
 n2
 n3

 A
 Maiúsc
 14
 14
 8

 a
 Minúsc
 17
 17
 23

In [422]: # para trocar a ordem de crescente para decrescente usa-se ascending=False
df2.sort_index(ascending=False)

Out[422]:

	Letra	n1	n2	n3
а	Minúsc	17	17	23
Α	Maiúsc	14	14	8