IndexingDataFrames

January 25, 2022

1 Indexing Dataframes

[11]:
```
#a função set_index é um processo destrutivo e não mantém o index atual
#se quisermos manter o index atual, precisamos manualmente criar uma nova␣
 ↪coluna e copiá-los para ela
#os valores
import pandas as pd
df = pd.read_csv('resources/week-1/datasets/Admission_Predict.csv', index_col=0)
df.head()
```

[11]:

Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
1	337	118	4	4.5	4.5	9.65	1	0.92
2	324	107	4	4.0	4.5	8.87	1	0.76
3	316	104	3	3.0	3.5	8.00	1	0.72
4	322	110	3	3.5	2.5	8.67	1	0.80
5	314	103	2	2.0	3.0	8.21	0	0.65

[12]:
```
#vamos fazer de conta que não queremos manter o serial number como index do␣
 ↪nosso DF mas sim o chance of admit, porém queremos
#manter esse valor do serial number para usar mais tarde
#fazemos isso usando o set_index para setar o index na coluna chance of admit

#primeiro copiamos o index atua lpara uma nova coluna
df['Serial No.'] = df.index

#daí setamos o index para uma nova coluna
df = df.set_index('Chance of Admit ')
df.head()
```

[12]:
```
                   GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA  \
Chance of Admit
0.92                     337          118                  4  4.5   4.5  9.65
0.76                     324          107                  4  4.0   4.5  8.87
0.72                     316          104                  3  3.0   3.5  8.00
0.80                     322          110                  3  3.5   2.5  8.67
0.65                     314          103                  2  2.0   3.0  8.21

                   Research  Serial No.
Chance of Admit
0.92                      1           1
0.76                      1           2
0.72                      1           3
0.80                      1           4
0.65                      0           5
```

[13]:
```
#quando criamos um index a aprtir de uma nova coluna, ele recebe o nome dessa␣
 ↪nova coluna
#podemos nos desfazer disso usando a função reset_index() para mover o index␣
 ↪para uma nova coluna
#e cria um index com números default

#ele pega o chance of admit que tinha ficado como indexador e coloca ele numa␣
 ↪nova coluna e cria um novo indexador
df = df.reset_index()
df.head()
```

[13]:
```
   Chance of Admit   GRE Score  TOEFL Score  University Rating  SOP  LOR   \
0              0.92        337          118                  4  4.5   4.5
1              0.76        324          107                  4  4.0   4.5
2              0.72        316          104                  3  3.0   3.5
3              0.80        322          110                  3  3.5   2.5
4              0.65        314          103                  2  2.0   3.0

   CGPA  Research  Serial No.
0  9.65         1           1
1  8.87         1           2
2  8.00         1           3
3  8.67         1           4
4  8.21         0           5
```

[ ]:
```
#uma coisa muito legal do pandas é a indexação multi-level que é similar Às␣
 ↪chaves compostas nos bancos de dados relacionais
#para criar um indexador multi-level simplesmente chamamos o set_index e␣
 ↪passamos uma lista com nomes de colunas
#que queremos transformar em indexadores
```

[14]:
```
df2 = pd.read_csv('resources/week-2/datasets/census.csv')
df2.head()
```

[14]:
```
   SUMLEV  REGION  DIVISION  STATE  COUNTY   STNAME         CTYNAME  \
0      40       3         6      1       0  Alabama         Alabama
1      50       3         6      1       1  Alabama  Autauga County
2      50       3         6      1       3  Alabama  Baldwin County
3      50       3         6      1       5  Alabama  Barbour County
4      50       3         6      1       7  Alabama     Bibb County

   CENSUS2010POP  ESTIMATESBASE2010  POPESTIMATE2010  ...  RDOMESTICMIG2011  \
0        4779736            4780127          4785161  ...          0.002295
1          54571              54571            54660  ...          7.242091
2         182265             182265           183193  ...         14.832960
3          27457              27457            27341  ...         -4.728132
4          22915              22919            22861  ...         -5.527043

   RDOMESTICMIG2012  RDOMESTICMIG2013  RDOMESTICMIG2014  RDOMESTICMIG2015  \
0         -0.193196          0.381066          0.582002         -0.467369
1         -2.915927         -3.012349          2.265971         -2.530799
2         17.647293         21.845705         19.243287         17.197872
3         -2.500690         -7.056824         -3.904217        -10.543299
4         -5.068871         -6.201001         -0.177537          0.177258

   RNETMIG2011  RNETMIG2012  RNETMIG2013  RNETMIG2014  RNETMIG2015
0     1.030015     0.826644     1.383282     1.724718     0.712594
1     7.606016    -2.626146    -2.722002     2.592270    -2.187333
2    15.844176    18.559627    22.727626    20.317142    18.293499
3    -4.874741    -2.758113    -7.167664    -3.978583   -10.543299
4    -5.088389    -4.363636    -5.403729     0.754533     1.107861

[5 rows x 100 columns]
```

[15]:
```
df2['SUMLEV'].unique()
```

[15]:
```
array([40, 50])
```

[20]:
```
#excluindo todas as linhas que são sumários a nível estadual e manter apenas os␣
 ↪dados do país
df2 = df2[df2['SUMLEV'] == 50]
#df2.head()
df2.columns
```

[20]:
```
Index(['SUMLEV', 'REGION', 'DIVISION', 'STATE', 'COUNTY', 'STNAME', 'CTYNAME',
       'CENSUS2010POP', 'ESTIMATESBASE2010', 'POPESTIMATE2010',
       'POPESTIMATE2011', 'POPESTIMATE2012', 'POPESTIMATE2013',
       'POPESTIMATE2014', 'POPESTIMATE2015', 'NPOPCHG_2010', 'NPOPCHG_2011',
       'NPOPCHG_2012', 'NPOPCHG_2013', 'NPOPCHG_2014', 'NPOPCHG_2015',
       'BIRTHS2010', 'BIRTHS2011', 'BIRTHS2012', 'BIRTHS2013', 'BIRTHS2014',
       'BIRTHS2015', 'DEATHS2010', 'DEATHS2011', 'DEATHS2012', 'DEATHS2013',
       'DEATHS2014', 'DEATHS2015', 'NATURALINC2010', 'NATURALINC2011',
       'NATURALINC2012', 'NATURALINC2013', 'NATURALINC2014', 'NATURALINC2015',
```

```
       'INTERNATIONALMIG2010', 'INTERNATIONALMIG2011', 'INTERNATIONALMIG2012',
       'INTERNATIONALMIG2013', 'INTERNATIONALMIG2014', 'INTERNATIONALMIG2015',
       'DOMESTICMIG2010', 'DOMESTICMIG2011', 'DOMESTICMIG2012',
       'DOMESTICMIG2013', 'DOMESTICMIG2014', 'DOMESTICMIG2015', 'NETMIG2010',
       'NETMIG2011', 'NETMIG2012', 'NETMIG2013', 'NETMIG2014', 'NETMIG2015',
       'RESIDUAL2010', 'RESIDUAL2011', 'RESIDUAL2012', 'RESIDUAL2013',
       'RESIDUAL2014', 'RESIDUAL2015', 'GQESTIMATESBASE2010',
       'GQESTIMATES2010', 'GQESTIMATES2011', 'GQESTIMATES2012',
       'GQESTIMATES2013', 'GQESTIMATES2014', 'GQESTIMATES2015', 'RBIRTH2011',
       'RBIRTH2012', 'RBIRTH2013', 'RBIRTH2014', 'RBIRTH2015', 'RDEATH2011',
       'RDEATH2012', 'RDEATH2013', 'RDEATH2014', 'RDEATH2015',
       'RNATURALINC2011', 'RNATURALINC2012', 'RNATURALINC2013',
       'RNATURALINC2014', 'RNATURALINC2015', 'RINTERNATIONALMIG2011',
       'RINTERNATIONALMIG2012', 'RINTERNATIONALMIG2013',
       'RINTERNATIONALMIG2014', 'RINTERNATIONALMIG2015', 'RDOMESTICMIG2011',
       'RDOMESTICMIG2012', 'RDOMESTICMIG2013', 'RDOMESTICMIG2014',
       'RDOMESTICMIG2015', 'RNETMIG2011', 'RNETMIG2012', 'RNETMIG2013',
       'RNETMIG2014', 'RNETMIG2015'],
      dtype='object')
```

[22]:
```
#vamos reduzir o dataset para mostrar apenas o estimado para a população e o␣
 ↪número total de nascimentos
#daí criamos uma lista com nomes de colunas que desejamos manter, projetá-las e␣
 ↪então
#definir o dataframe resultante para nossa variavel df2

columns_to_keep = ['STNAME', 'CTYNAME', 'BIRTHS2010', 'BIRTHS2011',␣
 ↪'BIRTHS2012', 'BIRTHS2013',
                   'BIRTHS2014', 'POPESTIMATE2010', 'POPESTIMATE2011',␣
 ↪'POPESTIMATE2012', 'POPESTIMATE2013',
                   'POPESTIMATE2014', 'POPESTIMATE2015']
df2 = df2[columns_to_keep]
df2.head()

#cria uma lista com as colunas que dejseamos manter e definimos a nova variavel␣
 ↪df2 com essas colunas
```

[22]:

	STNAME	CTYNAME	BIRTHS2010	BIRTHS2011	BIRTHS2012	BIRTHS2013	BIRTHS2014	POPESTIMATE2010	POPESTIMATE2011	POPESTIMATE2012
1	Alabama	Autauga County	151	636	615	574	623	54660	55253	55175
2	Alabama	Baldwin County	517	2187	2092	2160	2186	183193	186659	190396
3	Alabama	Barbour County	70	335	300	283	260	27341	27226	27159
4	Alabama	Bibb County	44	266	245	259				
5	Alabama	Blount County	183	744	710	646				

```
4         247            22861            22733            22642
5         618            57373            57711            57776

   POPESTIMATE2013  POPESTIMATE2014  POPESTIMATE2015
1            55038            55290            55347
2           195126           199713           203709
3            26973            26815            26489
4            22512            22549            22583
5            57734            57658            57673
```

[24]:
```
#o censo separa população por estimado por estado e país
#podemos carregar os dados e setar o index para ser a combinação do estado e do␣
 ↪país
#e daí ver como o pandas trabalha com isso num dataframe
#faremos isso criando uma lista com os identificadores de colunas que desejamos␣
 ↪indexar. daí chamamos o set_index()
#com essa lista e atribuimos o output como apropriado.
#vemos acima que temos dois indexadores, o STNAME e CTYNAME

df2 = df2.set_index(['STNAME', 'CTYNAME'])
df2.head()
```

[24]:
```
                        BIRTHS2010  BIRTHS2011  BIRTHS2012  BIRTHS2013  \
STNAME  CTYNAME
Alabama Autauga County         151         636         615         574
        Baldwin County         517        2187        2092        2160
        Barbour County          70         335         300         283
        Bibb County             44         266         245         259
        Blount County          183         744         710         646

                        BIRTHS2014  POPESTIMATE2010  POPESTIMATE2011  \
STNAME  CTYNAME
Alabama Autauga County         623            54660            55253
        Baldwin County        2186           183193           186659
        Barbour County         260            27341            27226
        Bibb County            247            22861            22733
        Blount County          618            57373            57711

                        POPESTIMATE2012  POPESTIMATE2013  POPESTIMATE2014  \
STNAME  CTYNAME
Alabama Autauga County            55175            55038            55290
        Baldwin County           190396           195126           199713
        Barbour County            27159            26973            26815
        Bibb County               22642            22512            22549
        Blount County             57776            57734            57658

                        POPESTIMATE2015
STNAME  CTYNAME
```

```
Alabama Autauga County            55347
        Baldwin County           203709
        Barbour County            26489
        Bibb County               22583
        Blount County             57673
```

[ ]:

[ ]:

[ ]:

[ ]:

[ ]: