1 第四章：数据组合

数据小鱼Rexa
CSDN：https://blog.csdn.net/qq_38395376?spm=1011.2124.3001.5343
Blibli：https://space.bilibili.com/283181288
Github：https://github.com/Rexa-Yu

1.1 连接

In [160]:

```
# 加载数据
import pandas as pd
df1=pd.read_csv(r'E:\jupyter notebook storage\Practice in Pandas\data/concat_1.csv')
df2=pd.read_csv(r'E:\jupyter notebook storage\Practice in Pandas\data/concat_2.csv')
df3=pd.read_csv(r'E:\jupyter notebook storage\Practice in Pandas\data/concat_3.csv')
print(df1)
print(df2)
print(df3)
```

```
    A   B   C   D
0  a0  b0  c0  d0
1  a1  b1  c1  d1
2  a2  b2  c2  d2
3  a3  b3  c3  d3
    A   B   C   D
0  a4  b4  c4  d4
1  a5  b5  c5  d5
2  a6  b6  c6  d6
3  a7  b7  c7  d7
     A    B    C    D
0   a8   b8   c8   d8
1   a9   b9   c9   d9
2  a10  b10  c10  d10
3  a11  b11  c11  d11
```

In [161]:

```
row_concat=pd.concat([df1,df2,df3])
row_concat
```

Out[161]:

	A	B	C	D
0	a0	b0	c0	d0
1	a1	b1	c1	d1
2	a2	b2	c2	d2
3	a3	b3	c3	d3
0	a4	b4	c4	d4
1	a5	b5	c5	d5
2	a6	b6	c6	d6
3	a7	b7	c7	d7
0	a8	b8	c8	d8
1	a9	b9	c9	d9
2	a10	b10	c10	d10
3	a11	b11	c11	d11

1.2 不同索引下的连接（使用join）

In [162]:

```
df1.columns=["A","B","C","D"]
df2.columns=['E','F','G','H']
df3.columns=["A","C","F","H"]
row_concat=pd.concat([df1,df2,df3])
row_concat
```

Out[162]:

	A	B	C	D	E	F	G	H
0	a0	b0	c0	d0	NaN	NaN	NaN	NaN
1	a1	b1	c1	d1	NaN	NaN	NaN	NaN
2	a2	b2	c2	d2	NaN	NaN	NaN	NaN
3	a3	b3	c3	d3	NaN	NaN	NaN	NaN
0	NaN	NaN	NaN	NaN	a4	b4	c4	d4
1	NaN	NaN	NaN	NaN	a5	b5	c5	d5
2	NaN	NaN	NaN	NaN	a6	b6	c6	d6
3	NaN	NaN	NaN	NaN	a7	b7	c7	d7
0	a8	NaN	b8	NaN	NaN	c8	NaN	d8
1	a9	NaN	b9	NaN	NaN	c9	NaN	d9
2	a10	NaN	b10	NaN	NaN	c10	NaN	d10
3	a11	NaN	b11	NaN	NaN	c11	NaN	d11

In [163]:

```
# inner会保留相同的列，但是同时符合三个数据集的列并没有，所以返回空
row_concat_inner=pd.concat([df1,df2,df3],join="inner")
row_concat_inner
```

Out[163]:

0
1
2
3
0
1
2
3
0
1
2
3

In [164]:

```
#  若只连接df1和df3，可以看到他们具有相同的列就返回了结果
row_concat_inner=pd.concat([df1,df3],join="inner")
row_concat_inner
```

Out[164]:

	A	C
0	a0	c0
1	a1	c1
2	a2	c2
3	a3	c3
0	a8	b8
1	a9	b9
2	a10	b10
3	a11	b11

In [165]:

```
# 重新设置df的index
df1.index=['0','1','2','3']
df2.index=['4','5','6','7']
df3.index=['0','2','5','7']
df3
```

Out[165]:

	A	C	F	H
0	a8	b8	c8	d8
2	a9	b9	c9	d9
5	a10	b10	c10	d10
7	a11	b11	c11	d11

In [166]:

```
# 横向链接，axis=1，可以用index连接
col_concat=pd.concat([df1,df2,df3])
col_concat
```

Out[166]:

	A	B	C	D	E	F	G	H
0	a0	b0	c0	d0	NaN	NaN	NaN	NaN
1	a1	b1	c1	d1	NaN	NaN	NaN	NaN
2	a2	b2	c2	d2	NaN	NaN	NaN	NaN
3	a3	b3	c3	d3	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	a4	b4	c4	d4
5	NaN	NaN	NaN	NaN	a5	b5	c5	d5
6	NaN	NaN	NaN	NaN	a6	b6	c6	d6
7	NaN	NaN	NaN	NaN	a7	b7	c7	d7
0	a8	NaN	b8	NaN	NaN	c8	NaN	d8
2	a9	NaN	b9	NaN	NaN	c9	NaN	d9
5	a10	NaN	b10	NaN	NaN	c10	NaN	d10
7	a11	NaN	b11	NaN	NaN	c11	NaN	d11

In [167]:

```
# 若只连接df1和df3，可以看到他们具有相同的列就返回了结果
col_concat=pd.concat([df1,df3],axis=1,join='inner')
col_concat
```

Out[167]:

	A	B	C	D	A	C	F	H
0	a0	b0	c0	d0	a8	b8	c8	d8
2	a2	b2	c2	d2	a9	b9	c9	d9

1.3 合并多个数据集使用merge函数

In [168]:

```
# 数据加载
person=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/survey_perso
site=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/survey_site.cs
survey=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/survey_surve
visited=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/survey_visi
```

In [169]:

```
person
```

Out[169]:

	ident	personal	family
0	dyer	William	Dyer
1	pb	Frank	Pabodie
2	lake	Anderson	Lake
3	roe	Valentina	Roerich
4	danforth	Frank	Danforth

In [170]:

```
site
```

Out[170]:

	name	lat	long
0	DR-1	-49.85	-128.57
1	DR-3	-47.15	-126.72
2	MSK-4	-48.87	-123.40

In [171]:

```
survey.head()
```

Out[171]:

	taken	person	quant	reading
0	619	dyer	rad	9.82
1	619	dyer	sal	0.13
2	622	dyer	rad	7.80
3	622	dyer	sal	0.09
4	734	pb	rad	8.41

In [172]:

```
visited
```

Out[172]:

	ident	site	dated
0	619	DR-1	1927-02-08
1	622	DR-1	1927-02-10
2	734	DR-3	1939-01-07
3	735	DR-3	1930-01-12
4	751	DR-3	1930-02-26
5	752	DR-3	NaN
6	837	MSK-4	1932-01-14
7	844	DR-1	1932-03-22

In [173]:

```
# 假设一张表由person和surveyhecheng, 另一个是visted和survey合成
ps=person.merge(survey,left_on="ident",right_on="person")
vs=visited.merge(survey,left_on="ident",right_on="taken")
ps.head()
```

Out[173]:

	ident	personal	family	taken	person	quant	reading
0	dyer	William	Dyer	619	dyer	rad	9.82
1	dyer	William	Dyer	619	dyer	sal	0.13
2	dyer	William	Dyer	622	dyer	rad	7.80
3	dyer	William	Dyer	622	dyer	sal	0.09
4	pb	Frank	Pabodie	734	pb	rad	8.41

In [174]:

```
vs.head()
```

Out[174]:

	ident	site	dated	taken	person	quant	reading
0	619	DR-1	1927-02-08	619	dyer	rad	9.82
1	619	DR-1	1927-02-08	619	dyer	sal	0.13
2	622	DR-1	1927-02-10	622	dyer	rad	7.80
3	622	DR-1	1927-02-10	622	dyer	sal	0.09
4	734	DR-3	1939-01-07	734	pb	rad	8.41

In [175]:

```
# ps 和 vs 联合
ps_vs=ps.merge(vs,left_on=["taken","person","quant","reading"],right_on=["taken","pe
ps_vs.head()
```

Out[175]:

	ident_x	personal	family	taken	person	quant	reading	ident_y	site	dated
0	dyer	William	Dyer	619	dyer	rad	9.82	619	DR-1	1927-02-08
1	dyer	William	Dyer	619	dyer	sal	0.13	619	DR-1	1927-02-08
2	dyer	William	Dyer	622	dyer	rad	7.80	622	DR-1	1927-02-10
3	dyer	William	Dyer	622	dyer	sal	0.09	622	DR-1	1927-02-10
4	pb	Frank	Pabodie	734	pb	rad	8.41	734	DR-3	1939-01-07

In [176]:

```
# how默认inner
ps_vs=ps.merge(vs,left_on=["taken","person","quant","reading"],right_on=["taken","pe
ps_vs.head()
```

Out[176]:

	ident_x	personal	family	taken	person	quant	reading	ident_y	site	dated
0	dyer	William	Dyer	619	dyer	rad	9.82	619	DR-1	1927-02-08
1	dyer	William	Dyer	619	dyer	sal	0.13	619	DR-1	1927-02-08
2	dyer	William	Dyer	622	dyer	rad	7.80	622	DR-1	1927-02-10
3	dyer	William	Dyer	622	dyer	sal	0.09	622	DR-1	1927-02-10
4	pb	Frank	Pabodie	734	pb	rad	8.41	734	DR-3	1939-01-07

2 第五章：缺失数据处理

2.1 定义缺失值

In [177]:

```
# 在Numpy中NAN、Nan、nan都代表缺失值，但是在Pandas中只有NaN是指缺失值
print(pd.isnull(ps_vs.iloc[16,9]))
# ps_vs的16行第9列为空
```

```
True
```

In [178]:

```
print(pd.notnull(ps_vs.iloc[16,9]))
```

```
False
```

2.2 加载测试集

In [179]:

```
visited_file=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/survey
visited_file
```

Out[179]:

	ident	site	dated
0	619	DR-1	1927-02-08
1	622	DR-1	1927-02-10
2	734	DR-3	1939-01-07
3	735	DR-3	1930-01-12
4	751	DR-3	1930-02-26
5	752	DR-3	NaN
6	837	MSK-4	1932-01-14
7	844	DR-1	1932-03-22

In [180]:

```
print(pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/survey_visite
# 不显示空值
```

```
   ident   site       dated
0    619   DR-1  1927-02-08
1    622   DR-1  1927-02-10
2    734   DR-3  1939-01-07
3    735   DR-3  1930-01-12
4    751   DR-3  1930-02-26
5    752   DR-3            
6    837  MSK-4  1932-01-14
7    844   DR-1  1932-03-22
```

In [181]:

```
print(pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/survey_visite
# 手动设置空值
```

```
   ident   site       dated
0    619   DR-1  1927-02-08
1    622   DR-1  1927-02-10
2    734   DR-3  1939-01-07
3    735   DR-3  1930-01-12
4    751   DR-3  1930-02-26
5    752   DR-3         NaN
6    837  MSK-4  1932-01-14
7    844   DR-1  1932-03-22
```

2.3 合并数据生成空值

In [182]:

```
visited
```

Out[182]:

	ident	site	dated
0	619	DR-1	1927-02-08
1	622	DR-1	1927-02-10
2	734	DR-3	1939-01-07
3	735	DR-3	1930-01-12
4	751	DR-3	1930-02-26
5	752	DR-3	NaN
6	837	MSK-4	1932-01-14
7	844	DR-1	1932-03-22

In [183]:

```
survey.head()
```

Out[183]:

	taken	person	quant	reading
0	619	dyer	rad	9.82
1	619	dyer	sal	0.13
2	622	dyer	rad	7.80
3	622	dyer	sal	0.09
4	734	pb	rad	8.41

In [184]:

```
s_v=survey.merge(visited,left_on="taken",right_on="ident")
s_v.head()
```

Out[184]:

	taken	person	quant	reading	ident	site	dated
0	619	dyer	rad	9.82	619	DR-1	1927-02-08
1	619	dyer	sal	0.13	619	DR-1	1927-02-08
2	622	dyer	rad	7.80	622	DR-1	1927-02-10
3	622	dyer	sal	0.09	622	DR-1	1927-02-10
4	734	pb	rad	8.41	734	DR-3	1939-01-07

2.4 重建索引应用（即使用空值来达到建立连续索引的目的）

In [185]:

```
# 加载测试数据
gampinder=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/gapminder
gampinder
```

Out[185]:

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
...	...	...	...	...	...	...
1699	Zimbabwe	Africa	1987	62.351	9216418	706.157306
1700	Zimbabwe	Africa	1992	60.377	10704340	693.420786
1701	Zimbabwe	Africa	1997	46.809	11404948	792.449960
1702	Zimbabwe	Africa	2002	39.989	11926563	672.038623
1703	Zimbabwe	Africa	2007	43.487	12311143	469.709298

1704 rows × 6 columns

In [186]:

```
life_exp=gamminder.groupby(["year"])["lifeExp"].mean()
life_exp
```

Out[186]:

```
year
1952    49.057620
1957    51.507401
1962    53.609249
1967    55.678290
1972    57.647386
1977    59.570157
1982    61.533197
1987    63.212613
1992    64.160338
1997    65.014676
2002    65.694923
2007    67.007423
Name: lifeExp, dtype: float64
```

In [187]:

```
# 使用空值方法填充缺失的index
year2000=life_exp[life_exp.index>2000]
year2000
```

Out[187]:

```
year
2002    65.694923
2007    67.007423
Name: lifeExp, dtype: float64
```

In [188]:

```
year2000.reindex(range(2000,2010))
```

Out[188]:

```
year
2000          NaN
2001          NaN
2002    65.694923
2003          NaN
2004          NaN
2005          NaN
2006          NaN
2007    67.007423
2008          NaN
2009          NaN
Name: lifeExp, dtype: float64
```

2.5 计算缺失值个数

In [189]:

```
ebola=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/country_times
ebola.head(5)
```

Out[189]:

	Date	Day	Cases_Guinea	Cases_Liberia	Cases_SierraLeone	Cases_Nigeria	Cases_Sen
0	1/5/2015	289	2776.0	NaN	10030.0	NaN	
1	1/4/2015	288	2775.0	NaN	9780.0	NaN	
2	1/3/2015	287	2769.0	8166.0	9722.0	NaN	
3	1/2/2015	286	NaN	8157.0	NaN	NaN	
4	12/31/2014	284	2730.0	8115.0	9633.0	NaN	

In [190]:

```
ebola.describe()
```

Out[190]:

	Day	Cases_Guinea	Cases_Liberia	Cases_SierraLeone	Cases_Nigeria	Cases_Sene
count	122.000000	93.000000	83.000000	87.000000	38.000000	25
mean	144.778689	911.064516	2335.337349	2427.367816	16.736842	1
std	89.316460	849.108801	2987.966721	3184.803996	5.998577	C
min	0.000000	49.000000	3.000000	0.000000	0.000000	1
25%	66.250000	236.000000	25.500000	64.500000	15.000000	1
50%	150.000000	495.000000	516.000000	783.000000	20.000000	1
75%	219.500000	1519.000000	4162.500000	3801.000000	20.000000	1
max	289.000000	2776.000000	8166.000000	10030.000000	22.000000	3

In [191]:

```
# 查看现有的属性的计数
ebola.count()
```

Out[191]:

```
Date                   122
Day                    122
Cases_Guinea            93
Cases_Liberia           83
Cases_SierraLeone       87
Cases_Nigeria           38
Cases_Senegal           25
Cases_UnitedStates      18
Cases_Spain             16
Cases_Mali              12
Deaths_Guinea           92
Deaths_Liberia          81
Deaths_SierraLeone      87
Deaths_Nigeria          38
Deaths_Senegal          22
Deaths_UnitedStates     18
Deaths_Spain            16
Deaths_Mali             12
dtype: int64
```

In [192]:

```
# 查看原本每个属性的计数(即行数)
ebola.shape[0]
```

Out[192]:

```
122
```

In [193]:

```
# 缺失计数
missing_ebola=ebola.shape[0]-ebola.count()
print(missing_ebola)
print(sum(missing_ebola))
```

```
Date                     0
Day                      0
Cases_Guinea            29
Cases_Liberia           39
Cases_SierraLeone       35
Cases_Nigeria           84
Cases_Senegal           97
Cases_UnitedStates     104
Cases_Spain            106
Cases_Mali             110
Deaths_Guinea           30
Deaths_Liberia          41
Deaths_SierraLeone      35
Deaths_Nigeria          84
Deaths_Senegal         100
Deaths_UnitedStates    104
Deaths_Spain           106
Deaths_Mali            110
dtype: int64
1214
```

In [194]:

```
# 可以用特殊函数（即np的count_nonzero）
import numpy as np
print(np.count_nonzero(ebola.isnull()))
```

```
1214
```

In [195]:

```
# 查询单独的缺失值，在数据上加上限定即可
print(np.count_nonzero(ebola["Deaths_Mali"].isnull()))
```

```
110
```

2.6 清理缺失数据（特定值填充，前、后值填充、插值填充）

In [196]:

```
# 源数据
ebola
```

Out[196]:

	Date	Day	Cases_Guinea	Cases_Liberia	Cases_SierraLeone	Cases_Nigeria	Cases_S
0	1/5/2015	289	2776.0	NaN	10030.0	NaN	
1	1/4/2015	288	2775.0	NaN	9780.0	NaN	
2	1/3/2015	287	2769.0	8166.0	9722.0	NaN	
3	1/2/2015	286	NaN	8157.0	NaN	NaN	
4	12/31/2014	284	2730.0	8115.0	9633.0	NaN	
...	...	...	...	...	...	...	
117	3/27/2014	5	103.0	8.0	6.0	NaN	
118	3/26/2014	4	86.0	NaN	NaN	NaN	
119	3/25/2014	3	86.0	NaN	NaN	NaN	
120	3/24/2014	2	86.0	NaN	NaN	NaN	
121	3/22/2014	0	49.0	NaN	NaN	NaN	

122 rows × 18 columns

In [197]:

```
# 直接替换,此函数有inplace选项，因此直接真实操作数据
ebola.fillna(0)
```

Out[197]:

	Date	Day	Cases_Guinea	Cases_Liberia	Cases_SierraLeone	Cases_Nigeria	Cases_S
0	1/5/2015	289	2776.0	0.0	10030.0	0.0	
1	1/4/2015	288	2775.0	0.0	9780.0	0.0	
2	1/3/2015	287	2769.0	8166.0	9722.0	0.0	
3	1/2/2015	286	0.0	8157.0	0.0	0.0	
4	12/31/2014	284	2730.0	8115.0	9633.0	0.0	
...	...	...	...	...	...	...	
117	3/27/2014	5	103.0	8.0	6.0	0.0	
118	3/26/2014	4	86.0	0.0	0.0	0.0	
119	3/25/2014	3	86.0	0.0	0.0	0.0	
120	3/24/2014	2	86.0	0.0	0.0	0.0	
121	3/22/2014	0	49.0	0.0	0.0	0.0	

122 rows × 18 columns

In [198]:

```
# 前值填充（fill forward）和后值填充（fill backward）
ebola.fillna(method="ffill")
#  因为有些是从空值开始的，所以按照前置填充，并不会找到前置，因此保留空值。
```

Out[198]:

	Date	Day	Cases_Guinea	Cases_Liberia	Cases_SierraLeone	Cases_Nigeria	Cases_S
0	1/5/2015	289	2776.0	NaN	10030.0	NaN	
1	1/4/2015	288	2775.0	NaN	9780.0	NaN	
2	1/3/2015	287	2769.0	8166.0	9722.0	NaN	
3	1/2/2015	286	2769.0	8157.0	9722.0	NaN	
4	12/31/2014	284	2730.0	8115.0	9633.0	NaN	
...	...	...	...	...	...	...	
117	3/27/2014	5	103.0	8.0	6.0	0.0	
118	3/26/2014	4	86.0	8.0	6.0	0.0	
119	3/25/2014	3	86.0	8.0	6.0	0.0	
120	3/24/2014	2	86.0	8.0	6.0	0.0	
121	3/22/2014	0	49.0	8.0	6.0	0.0	

122 rows × 18 columns

In [199]:

```
ebola.fillna(method="bfill")
# 后置填充是根据后一个值进行赋值的，因此在后置为空的情况下，也会保留。
```

Out[199]:

	Date	Day	Cases_Guinea	Cases_Liberia	Cases_SierraLeone	Cases_Nigeria	Cases_S
0	1/5/2015	289	2776.0	8166.0	10030.0	20.0	
1	1/4/2015	288	2775.0	8166.0	9780.0	20.0	
2	1/3/2015	287	2769.0	8166.0	9722.0	20.0	
3	1/2/2015	286	2730.0	8157.0	9633.0	20.0	
4	12/31/2014	284	2730.0	8115.0	9633.0	20.0	
...	...	...	...	...	...	...	
117	3/27/2014	5	103.0	8.0	6.0	NaN	
118	3/26/2014	4	86.0	NaN	NaN	NaN	
119	3/25/2014	3	86.0	NaN	NaN	NaN	
120	3/24/2014	2	86.0	NaN	NaN	NaN	
121	3/22/2014	0	49.0	NaN	NaN	NaN	

122 rows × 18 columns

In [200]:

```
# 删除空值，可以使用dropna
print(ebola.dropna())
```

```
          Date  Day  Cases_Guinea  Cases_Liberia  Cases_SierraLeone  \
19  11/18/2014  241        2047.0         7082.0             6190.0   

    Cases_Nigeria  Cases_Senegal  Cases_UnitedStates  Cases_Spain  Cas
es_Mali  \
19           20.0            1.0                 4.0          1.0  
6.0   

    Deaths_Guinea  Deaths_Liberia  Deaths_SierraLeone  Deaths_Nigeria  
\
19         1214.0          2963.0              1267.0             8.0   

    Deaths_Senegal  Deaths_UnitedStates  Deaths_Spain  Deaths_Mali  
19             0.0                  1.0           0.0          6.0  
```

特别的，一般参数都会有skipna参数来调节是否忽视空值。