1 第六章：整理数据

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 固定一列进行观察

In [84]:

```
import pandas as pd
# 加载数据
pew=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data\pew.csv")
pew
```

Out[84]:

	religion	<$10k	$10-20k	$20-30k	$30-40k	$40-50k	$50-75k	$75-100k	$100-150k	>150k	Don't know/refused
0	Agnostic	27	34	60	81	76	137	122	109	84	96
1	Atheist	12	27	37	52	35	70	73	59	74	76
2	Buddhist	27	21	30	34	33	58	62	39	53	54
3	Catholic	418	617	732	670	638	1116	949	792	633	1489
4	Don’t know/refused	15	14	15	11	10	35	21	17	18	116
5	Evangelical Prot	575	869	1064	982	881	1486	949	723	414	1529
6	Hindu	1	9	7	9	11	34	47	48	54	37
7	Historically Black Prot	228	244	236	238	197	223	131	81	78	339
8	Jehovah's Witness	20	27	24	24	21	30	15	11	6	37
9	Jewish	19	19	25	25	30	95	69	87	151	162
10	Mainline Prot	289	495	619	655	651	1107	939	753	634	1328
11	Mormon	29	40	48	51	56	112	85	49	42	69
12	Muslim	6	7	9	10	9	23	16	8	6	22
13	Orthodox	13	17	23	32	32	47	38	42	46	73
14	Other Christian	9	7	11	13	13	14	18	14	12	18
15	Other Faiths	20	33	40	46	49	63	46	40	41	71
16	Other World Religions	5	2	3	4	2	7	3	4	4	8
17	Unaffiliated	217	299	374	365	341	528	407	321	258	597

In [85]:

```
#由于宽表可以作图，长表适合进行数据处理，在最初的处理操作上，可以先通过长表进行数据处理，然后合成相
pew_long=pd.melt(pew,id_vars="religion")
pew_long.head(5)
```

Out[85]:

	religion	variable	value
0	Agnostic	<$10k	27
1	Atheist	<$10k	12
2	Buddhist	<$10k	27
3	Catholic	<$10k	418
4	Don’t know/refused	<$10k	15

In [86]:

```
#  可以更改其默认值。比如variable和value
pew_long=pd.melt(pew,id_vars="religion",var_name="income",value_name="count")
pew_long
```

Out[86]:

	religion	income	count
0	Agnostic	<$10k	27
1	Atheist	<$10k	12
2	Buddhist	<$10k	27
3	Catholic	<$10k	418
4	Don’t know/refused	<$10k	15
...	...	...	...
175	Orthodox	Don't know/refused	73
176	Other Christian	Don't know/refused	18
177	Other Faiths	Don't know/refused	71
178	Other World Religions	Don't know/refused	8
179	Unaffiliated	Don't know/refused	597

180 rows × 3 columns

1.2 固定多列

In [87]:

```
# 加载数据
billboard=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/billboard
billboard.head()
```

Out[87]:

	year	artist	track	time	date.entered	wk1	wk2	wk3	wk4	wk5	...	wk67	wk68
0	2000	2 Pac	Baby Don't Cry (Keep...	4:22	2000-02-26	87	82.0	72.0	77.0	87.0	...	NaN	NaN
1	2000	2Ge+her	The Hardest Part Of ...	3:15	2000-09-02	91	87.0	92.0	NaN	NaN	...	NaN	NaN
2	2000	3 Doors Down	Kryptonite	3:53	2000-04-08	81	70.0	68.0	67.0	66.0	...	NaN	NaN
3	2000	3 Doors Down	Loser	4:24	2000-10-21	76	76.0	72.0	69.0	67.0	...	NaN	NaN
4	2000	504 Boyz	Wobble Wobble	3:35	2000-04-15	57	34.0	25.0	17.0	17.0	...	NaN	NaN

5 rows × 81 columns

In [88]:

```
billboard.describe()
```

Out[88]:

	year	wk1	wk2	wk3	wk4	wk5	wk6	v
count	317.0	317.000000	312.000000	307.000000	300.000000	292.000000	280.000000	269.000
mean	2000.0	79.958991	71.173077	65.045603	59.763333	56.339041	52.360714	49.219
std	0.0	14.686865	18.200443	20.752302	22.324619	23.780022	24.473273	25.654
min	2000.0	15.000000	8.000000	6.000000	5.000000	2.000000	1.000000	1.000
25%	2000.0	74.000000	63.000000	53.000000	44.750000	38.750000	33.750000	30.000
50%	2000.0	81.000000	73.000000	66.000000	61.000000	57.000000	51.500000	47.000
75%	2000.0	91.000000	84.000000	79.000000	76.000000	73.250000	72.250000	67.000
max	2000.0	100.000000	100.000000	100.000000	100.000000	100.000000	99.000000	100.000

8 rows × 77 columns

In [89]:

```
# 对于以上数据，我们可以看出数据集有很多的时间数据，weeks，因此我们可以进行数据逆透视，日期聚合
billboard_long=pd.melt(billboard,id_vars=["year","artist","track","time","date.enter
billboard_long
```

Out[89]:

	year	artist	track	time	date.entered	week	rating
0	2000	2 Pac	Baby Don't Cry (Keep...	4:22	2000-02-26	wk1	87.0
1	2000	2Ge+her	The Hardest Part Of ...	3:15	2000-09-02	wk1	91.0
2	2000	3 Doors Down	Kryptonite	3:53	2000-04-08	wk1	81.0
3	2000	3 Doors Down	Loser	4:24	2000-10-21	wk1	76.0
4	2000	504 Boyz	Wobble Wobble	3:35	2000-04-15	wk1	57.0
...	...	...	...	...	...	...	...
24087	2000	Yankee Grey	Another Nine Minutes	3:10	2000-04-29	wk76	NaN
24088	2000	Yearwood, Trisha	Real Live Woman	3:55	2000-04-01	wk76	NaN
24089	2000	Ying Yang Twins	Whistle While You Tw...	4:19	2000-03-18	wk76	NaN
24090	2000	Zombie Nation	Kernkraft 400	3:30	2000-09-02	wk76	NaN
24091	2000	matchbox twenty	Bent	4:12	2000-04-29	wk76	NaN

24092 rows × 7 columns

1.3 包含多个变量的列

In [90]:

```
# 加载数据
ebola=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\data/country_times
ebola.columns
```

Out[90]:

```
Index(['Date', 'Day', 'Cases_Guinea', 'Cases_Liberia', 'Cases_SierraLe
one',
       'Cases_Nigeria', 'Cases_Senegal', 'Cases_UnitedStates', 'Cases_
Spain',
       'Cases_Mali', 'Deaths_Guinea', 'Deaths_Liberia', 'Deaths_Sierra
Leone',
       'Deaths_Nigeria', 'Deaths_Senegal', 'Deaths_UnitedStates',
       'Deaths_Spain', 'Deaths_Mali'],
      dtype='object')
```

In [91]:

```
# 输出带有Guinea和liberia的属性
print(ebola.iloc[:5,[0,1,2,3,10,11]])
```

```
         Date  Day  Cases_Guinea  Cases_Liberia  Deaths_Guinea  Deaths
_Liberia
0    1/5/2015  289        2776.0            NaN         1786.0
NaN
1    1/4/2015  288        2775.0            NaN         1781.0
NaN
2    1/3/2015  287        2769.0         8166.0         1767.0
3496.0
3    1/2/2015  286           NaN         8157.0            NaN
3496.0
4  12/31/2014  284        2730.0         8115.0         1739.0
3471.0
```

In [92]:

```
# 变为长表
ebola_long=pd.melt(ebola,id_vars=["Date","Day"])
ebola_long
```

Out[92]:

	Date	Day	variable	value
0	1/5/2015	289	Cases_Guinea	2776.0
1	1/4/2015	288	Cases_Guinea	2775.0
2	1/3/2015	287	Cases_Guinea	2769.0
3	1/2/2015	286	Cases_Guinea	NaN
4	12/31/2014	284	Cases_Guinea	2730.0
...	...	...	...	...
1947	3/27/2014	5	Deaths_Mali	NaN
1948	3/26/2014	4	Deaths_Mali	NaN
1949	3/25/2014	3	Deaths_Mali	NaN
1950	3/24/2014	2	Deaths_Mali	NaN
1951	3/22/2014	0	Deaths_Mali	NaN

1952 rows × 4 columns

1.4 单独拆分和添加列

In [93]:

```
variable_split=ebola_long.variable.str.split("_")
variable_split
```

Out[93]:

```
0       [Cases, Guinea]
1       [Cases, Guinea]
2       [Cases, Guinea]
3       [Cases, Guinea]
4       [Cases, Guinea]
             ...
1947     [Deaths, Mali]
1948     [Deaths, Mali]
1949     [Deaths, Mali]
1950     [Deaths, Mali]
1951     [Deaths, Mali]
Name: variable, Length: 1952, dtype: object
```

In [94]:

```
status_vlaues=variable_split.str.get(0)
country_values=variable_split.str.get(1)
print(status_vlaues)
print(country_values)
```

```
0        Cases
1        Cases
2        Cases
3        Cases
4        Cases
         ...
1947    Deaths
1948    Deaths
1949    Deaths
1950    Deaths
1951    Deaths
Name: variable, Length: 1952, dtype: object
0       Guinea
1       Guinea
2       Guinea
3       Guinea
4       Guinea
         ...
1947      Mali
1948      Mali
1949      Mali
1950      Mali
1951      Mali
Name: variable, Length: 1952, dtype: object
```

In [95]:

```
# 直接赋值
ebola_long["status"]=status_vlaues
ebola_long["country"]=country_values
ebola_long
```

Out[95]:

	Date	Day	variable	value	status	country
0	1/5/2015	289	Cases_Guinea	2776.0	Cases	Guinea
1	1/4/2015	288	Cases_Guinea	2775.0	Cases	Guinea
2	1/3/2015	287	Cases_Guinea	2769.0	Cases	Guinea
3	1/2/2015	286	Cases_Guinea	NaN	Cases	Guinea
4	12/31/2014	284	Cases_Guinea	2730.0	Cases	Guinea
...	...	...	...	...	...	...
1947	3/27/2014	5	Deaths_Mali	NaN	Deaths	Mali
1948	3/26/2014	4	Deaths_Mali	NaN	Deaths	Mali
1949	3/25/2014	3	Deaths_Mali	NaN	Deaths	Mali
1950	3/24/2014	2	Deaths_Mali	NaN	Deaths	Mali
1951	3/22/2014	0	Deaths_Mali	NaN	Deaths	Mali

1952 rows × 6 columns

1.5 在单个步骤中进行拆分和组合

In [96]:

```
# 由于拆分的数据和原先之前的数据具有相同的index，所以我们可以利用这一特性进行处理
variable_split=ebola_long.variable.str.split("_",expand=True)
variable_split
```

Out[96]:

	0	1
0	Cases	Guinea
1	Cases	Guinea
2	Cases	Guinea
3	Cases	Guinea
4	Cases	Guinea
...	...	...
1947	Deaths	Mali
1948	Deaths	Mali
1949	Deaths	Mali
1950	Deaths	Mali
1951	Deaths	Mali

1952 rows × 2 columns

In [97]:

```
# 设置列名
variable_split.columns=["status","country"]
variable_split
```

Out[97]:

	status	country
0	Cases	Guinea
1	Cases	Guinea
2	Cases	Guinea
3	Cases	Guinea
4	Cases	Guinea
...	...	...
1947	Deaths	Mali
1948	Deaths	Mali
1949	Deaths	Mali
1950	Deaths	Mali
1951	Deaths	Mali

1952 rows × 2 columns

In [98]:

```
# 联合
ebola_parsed=pd.concat([ebola_long,variable_split],axis=1)
ebola_parsed
```

Out[98]:

	Date	Day	variable	value	status	country	status	country
0	1/5/2015	289	Cases_Guinea	2776.0	Cases	Guinea	Cases	Guinea
1	1/4/2015	288	Cases_Guinea	2775.0	Cases	Guinea	Cases	Guinea
2	1/3/2015	287	Cases_Guinea	2769.0	Cases	Guinea	Cases	Guinea
3	1/2/2015	286	Cases_Guinea	NaN	Cases	Guinea	Cases	Guinea
4	12/31/2014	284	Cases_Guinea	2730.0	Cases	Guinea	Cases	Guinea
...	...	...	...	...	...	...	...	...
1947	3/27/2014	5	Deaths_Mali	NaN	Deaths	Mali	Deaths	Mali
1948	3/26/2014	4	Deaths_Mali	NaN	Deaths	Mali	Deaths	Mali
1949	3/25/2014	3	Deaths_Mali	NaN	Deaths	Mali	Deaths	Mali
1950	3/24/2014	2	Deaths_Mali	NaN	Deaths	Mali	Deaths	Mali
1951	3/22/2014	0	Deaths_Mali	NaN	Deaths	Mali	Deaths	Mali

1952 rows × 8 columns

In [99]:

```
# 使用zip函数也可以
zp=zip(*ebola_long.variable.str.split("_"))
ebola_long["status"],ebola_long["country"]=zp
ebola_long
```

Out[99]:

	Date	Day	variable	value	status	country
0	1/5/2015	289	Cases_Guinea	2776.0	Cases	Guinea
1	1/4/2015	288	Cases_Guinea	2775.0	Cases	Guinea
2	1/3/2015	287	Cases_Guinea	2769.0	Cases	Guinea
3	1/2/2015	286	Cases_Guinea	NaN	Cases	Guinea
4	12/31/2014	284	Cases_Guinea	2730.0	Cases	Guinea
...	...	...	...	...	...	...
1947	3/27/2014	5	Deaths_Mali	NaN	Deaths	Mali
1948	3/26/2014	4	Deaths_Mali	NaN	Deaths	Mali
1949	3/25/2014	3	Deaths_Mali	NaN	Deaths	Mali
1950	3/24/2014	2	Deaths_Mali	NaN	Deaths	Mali
1951	3/22/2014	0	Deaths_Mali	NaN	Deaths	Mali

1952 rows × 6 columns

1.6 加载多个文件

加载数据文件夹，可以利用多个变量去获取文件

import os

path = os.getcwd()#获取当前路径

with open(path+"文件名", "r") as file:

2 第七章：数据类型

2.1 转换为字符串对象

In [100]:

```
# 加载数据集
tips=pd.read_csv("E:\jupyter notebook storage\Practice in Pandas\seaborn-data-master
tips
```

Out[100]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

In [101]:

```
# 查看数据类型
tips.dtypes
```

Out[101]:

```
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
dtype: object
```

In [102]:

```
# 可以看到sex的类型为object，我们将它转化为str
tips["sex_str"]=tips["sex"].astype(str)
tips.dtypes
```

Out[102]:

```
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
sex_str        object
dtype: object
```

In [103]:

```
tips["total_bill"]=tips["total_bill"].astype(str)
tips.dtypes
```

Out[103]:

```
total_bill     object
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
sex_str        object
dtype: object
```

In [104]:

```
# 将str换为float即可
tips["total_bill"]=tips["total_bill"].astype(float)
tips.dtypes
```

Out[104]:

```
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
sex_str        object
dtype: object
```

2.2 数值型转换

2.2.1 to_numeric函数

In [105]:

```
# 制造一个带有其他字符的数据集
tips_sub_miss=tips.head(10)
tips_sub_miss.loc[[1,3,5,7],"total_bill"]="missing"
tips_sub_miss
```

```
E:\anaconda\lib\site-packages\pandas\core\indexing.py:1765: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  isetter(loc, value)
```

Out[105]:

	total_bill	tip	sex	smoker	day	time	size	sex_str
0	16.99	1.01	Female	No	Sun	Dinner	2	Female
1	missing	1.66	Male	No	Sun	Dinner	3	Male
2	21.01	3.50	Male	No	Sun	Dinner	3	Male
3	missing	3.31	Male	No	Sun	Dinner	2	Male
4	24.59	3.61	Female	No	Sun	Dinner	4	Female
5	missing	4.71	Male	No	Sun	Dinner	4	Male
6	8.77	2.00	Male	No	Sun	Dinner	2	Male
7	missing	3.12	Male	No	Sun	Dinner	4	Male
8	15.04	1.96	Male	No	Sun	Dinner	2	Male
9	14.78	3.23	Male	No	Sun	Dinner	2	Male

In [106]:

```
# 原始数据total_bill的类型
tips_sub_miss.dtypes
```

Out[106]:

```
total_bill     object
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
sex_str        object
dtype: object
```

In [107]:

```
tips_sub_miss["total_bill"]=pd.to_numeric(tips_sub_miss["total_bill"],errors="coerce
```

```
<ipython-input-107-1d96fd55dd61>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  tips_sub_miss["total_bill"]=pd.to_numeric(tips_sub_miss["total_bil
l"],errors="coerce")
```

to_numeri的errors有三个参数：

1. raise：此为默认参数，当函数遇到无法转换的数值时，返回一个错误。
2. coerce： 当函数遇到无法转换的数值时，返回Nan。
3. ignore： 当函数遇到无法转换的的数值时，放弃转换，直接返回列。

In [108]:

```
tips_sub_miss.dtypes
```

Out[108]:

```
total_bill    float64
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
sex_str        object
dtype: object
```

In [109]:

```
tips_sub_miss["total_bill"]=pd.to_numeric(tips_sub_miss["total_bill"],downcast="floa
```

```
<ipython-input-109-c3ff399e0197>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  tips_sub_miss["total_bill"]=pd.to_numeric(tips_sub_miss["total_bil
l"],downcast="float")
```

to_numeric的downcast可以选择参数，默认为None，还可以选择“integer”，“signed”，“unsigned”，“float”。 这样操作会使所选取的类型占内存最小，比如tips_bill的float64可以变成float32。

In [110]:

```
tips_sub_miss.dtypes
```

Out[110]:

```
total_bill    float32
tip           float64
sex            object
smoker         object
day            object
time           object
size            int64
sex_str        object
dtype: object
```

2.2.2 category类型

这个类型可以使用astype去操作，转换类型。 同时还会有其他分类的API，比如Series.cat.ordered()表示是否有顺序等等