Wine-reviews 数据集数据可视化

1、 数据集说明

该数据集是一个描述葡萄酒的数据集，包括

9 个标称属性

- country
- designation
- province
- region_1
- region_2
- taster_name
- taster_twitter_handle
- variety
- winery

2 个数值属性

- points
- price

2 个不予考虑缺失的属性：

- description（每条记录都不相同，且无法其他关联属性）
- title（每条记录都不相同）

2、 原始数据集可视化

文件 wine_130k_origin.py 对原文件 winemag-data-130k-v2.csv 进行了可视化，可视化结果显示在 fig_130k/origin 中。

文件 wine_150k_origin.py 对原文件 winemag-data_first150k.csv 进行了可视化，可视化结果显示在 fig_150k/origin 中。

对于 9 个标称属性，均使用柱状图进行可视化；对于 2 个数值属性，使用盒图进行可视化（对于属性取值超过 25 个的，仅取前 25 个属性进行可视化），并将五数概括的结果存储在各个文件夹的 five_number_summary.csv 文件中。

	A	B	C	D	E	F	G
1	name	minimum	Q1	median	Q3	maximum	
2	points	80	86	88	91	100	
3	price	4	nan	nan	nan	3300	
4							

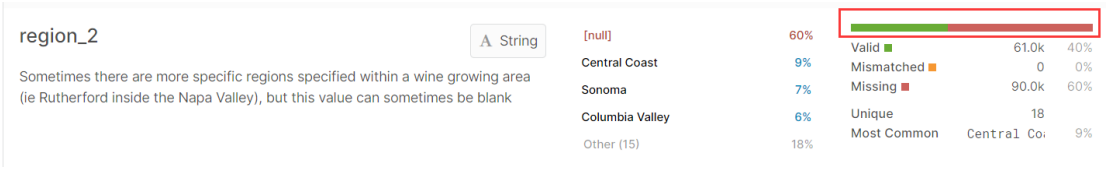
从五数概括的结果可以看出，price 属性的 Q1、median、Q3 均为 nan，比较影响 price 属性的分析。

3、 将缺失部分剔除

文件 wine_130k_dropnan.py 对原文件 winemag-data-130k-v2.csv 直接剔除了缺失值所在行，可视化结果显示在 fig_130k/dropnan 中。

文件 wine_150k_dropnan.py 对原文件 winemag-data_first150k.csv 直接剔除了缺失值所在行，可视化结果显示在 fig_150k/dropnan 中。

在这部分，我并没有直接使用 pandas 中的 dropna 函数直接对缺失值进行剔除，因为数据集中有一个特殊的属性 region_2。从数据集说明中，可以看出这个属性描述的是一个相比 region_1 更为具体的地区数据，但是这个属性不是必须要有的，有时候这个属性没有。而且从 kaggle 的描述中，可以看出，将超过一半的数据都缺少此属性，所以如果一行数据，仅缺少 region_2 属性，不需要剔除。



因此，我有选择性的对缺失除 region_2 以外的其他属性的行进行剔除。

从结果可以看出，使用缺失值剔除的方法，可以去除空值，在五数概括中可以明显的看出数值属性的分布范围。

	A	B	C	D	E	F
1	name	minimum	Q1	median	Q3	maximum
2	points	80	87	89	91	100
3	price	4	20	30	50	2013
4						
5						

4、 用最高频率值来填补缺失值

文件 wine_130k_high_frequency.py 对原文件 winemag-data-130k-v2.csv 中的缺失值，使用最高频率进行填补，可视化结果显示在 fig_130k/high_frequency 中。

文件 wine_150k_high_frequency.py 对原文件 winemag-data_first150k.csv 中的缺失值使用最高频率进行填补，可视化结果显示在 fig_150k/high_frequency 中。

对所有的属性，使用 pandas 自带函数 value_counts 对属性中的所有值出现的频率进行统计，取出现频率最高的非空值填补空缺。

5、 通过属性的相关关系来填补缺失值

文件 wine_130k_relationship.py 对文件 winemag-data-130k-v2.csv 中的缺失值使用属性间的相关关系进行填补，可视化结果显示在 fig_130k/relationship 中。

没有对文件 winemag-data_first150k.csv 中的缺失值使用该方法进行填补，因为经分析，该文件不含有相关关系的属性。

对文件 winemag-data-130k-v2.csv 中的属性进行相关关系分析，可以发现：

1） title 中括号里的内容对应 region_2 属性

14	US	Building on 150 year	87	12	California	Central Coast	Central Co	Matt Ketti	@mattket Mirassou 2012 Chardonnay (Central Coast)
15	Germany	Zesty orar	Devon	87	24	Mosel			Anna Lee C. Iijima Richard B枚cking 2013 Devon Riesling (Mosel)
16	Argentina	Baked plu	Felix	87	30	Other	Cafayate		Michael Sı @winesch Felix Lavaque 2010 Felix Malbec (Cafayate)
17	Argentina	Raw black	Winemake	87	13	Mendoza	Mendoza		Michael Sı @winesch Gaucho Andino 2011 Winemaker Selection Malbec (Mendoza)
18	Spain	Desiccate	Vendimia	87	28	Northern	Ribera del Duero		Michael Sı @winesch Pradorey 2010 Vendimia Seleccionada Finca Valdelayegua Single Vineyard Crianza
19	US	Red fruit aromas per		87	32	Virginia	Virginia		Alexander Peartree Qui矛vremont 2012 Meritage (Virginia)
20	US	Ripe arom	Vin de Ma	87	23	Virginia	Virginia		Alexander Peartree Qui矛vremont 2012 Vin de Maison Red (Virginia)
21	US	A sleek mix of tart b		87	20	Oregon	Oregon	Oregon C	Paul Greg @paulgwi Acrobat 2013 Pinot Noir (Oregon)
22	Italy	Delicate a	Ficiligno	87	19	Sicily & Sa	Sicilia		Kerin O欧 @kerinok Baglio di Pianetto 2007 Ficiligno White (Sicilia)
23	US	This wine	Signature	87	22	California	Paso Robles	Central Co	Matt Ketti @mattket Bianchi 2011 Signature Selection Merlot (Paso Robles)
24	Italy	Aromas of	Aynat	87	35	Sicily & Sa	Sicilia		Kerin O欧 @kerinok Canicatt矛 2009 Aynat Nero d'Avola (Sicilia)
25	US	Oak and e	King Ridg	87	69	California	Sonoma Coast	Sonoma	Virginie Bı @vboone Castello di Amorosa 2011 King Ridge Vineyard Pinot Noir (Sonoma Coast)
26	Italy	Pretty aro	Dalila	87	13	Sicily & Sa	Terre Siciliane		Kerin O欧 @kerinok Stemmari 2013 Dalila White (Terre Siciliane)
27	Italy	Aromas recall ripe d		87	10	Sicily & Sa	Terre Siciliane		Kerin O欧 @kerinok Stemmari 2013 Nero d'Avola (Terre Siciliane)
28	Italy	Aromas su	Mascaria I	87	17	Sicily & Sa	Cerasuolo di Vittoria		Kerin O欧 @kerinok Terre di Giurfo 2011 Mascaria Barricato (Cerasuolo di Vittoria)
29	US	Clarksburg is becomi		86	16	California	Clarksburg	Central Vá	Virginie Bı @vboone Clarksburg Wine Company 2010 Chenin Blanc (Clarksburg)

因此，可以使用正则表达式，提取 title 属性中括号里的内容，对 region_2 的缺失值进行填充，核心代码如下：

```
if type(row['region_2']) == float and math.isnan(row['region_2']):
    region_2 = re.findall(r'[(](.*?)[)]', row['title'])
    if len(region_2) > 0:
        data.loc[index, 'region_1'] = region_2[0]
```

2） title 年份后，括号前的内容，与 designation 属性相关。

	D	E	F	G	H	I	J	K	L
1	designation	points	price	province	region_	region_2	taster_nar	taster	title
2	Vulk胨 Bianco	87		Sicily & Sa	Etna		Kerin O鈥	@keri	Nicosia 2013 Vulk胨 Bianco (Etna)
3	Avidagos	87	15	Douro			Roger Vos	@vos	Quinta dos Avidagos 2011 Avidagos Red (Douro)
4		87	14	Oregon	Willam	Willame	Paul Greg	@pau	Rainstorm 2013 Pinot Gris (Willamette Valley)
5	Reserve Late Harvest	87	13	Michigan	Lake Michigan		Alexander	Pearti	St. Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shore)
6	Vintner's Reserve Wild Child Block	87	65	Oregon	Willam	Willame	Paul Greg	@pau	Sweet Cheeks 2012 Vintner's Reserve Wild Child Block Pinot Noir (Willamette Valley)
7	Ars In Vitro	87	15	Northern	Navarra		Michael S	@win	Tandem 2011 Ars In Vitro Tempranillo-Merlot (Navarra)
8	Belsito	87	16	Sicily & Sa	Vittoria		Kerin O鈥	@keri	Terre di Giurfo 2013 Belsito Frappato (Vittoria)
9		87	24	Alsace	Alsace		Roger Vos	@vos	Trimbach 2012 Gewurztraminer (Alsace)
10	Shine	87	12	Rheinhessen			Anna Lee C. Iijim		Heinz Eifel 2013 Shine Gewürztraminer (Rheinhessen)
11	Les Natures	87	27	Alsace	Alsace		Roger Vos	@vos	Jean-Baptiste Adam 2012 Les Natures Pinot Gris (Alsace)
12	Mountain Cuv矛e	87	19	California	Napa \	Napa	Virginie B	@vbo	Kirkland Signature 2011 Mountain Cuv矛e Cabernet Sauvignon (Napa Valley)
13		87	30	Alsace	Alsace		Roger Vos	@vos	Leon Beyer 2012 Gewurztraminer (Alsace)
14		87	34	California	Alexan	Sonoma	Virginie B	@vbo	Louis M. Martini 2012 Cabernet Sauvignon (Alexander Valley)
15	Rosso	87		Sicily & Sa	Etna		Kerin O鈥	@keri	Masseria Setteporte 2012 Rosso (Etna)
16	barely ripe pineapple prove approa	87	12	California	Centra	Central	Matt Kett	@mat	Mirassou 2012 Chardonnay (Central Coast)
17	Devon	87	24	Mosel			Anna Lee C. Iijim		Richard B枚cking 2013 Devon Riesling (Mosel)
18	Felix	87	30	Other	Cafayate		Michael S	@win	Felix Lavaque 2010 Felix Malbec (Cafayate)
19	Winemaker Selection	87	13	Mendoza	Mendoza		Michael S	@win	Gaucho Andino 2011 Winemaker Selection Malbec (Mendoza)
20	Vendimia Seleccionada Finca Valdel	87	28	Northern	Ribera del Due		Michael S	@win	Pradorey 2010 Vendimia Seleccionada Finca Valdelayegua Single Vineyard Crianza (
21	auvignon and Cabernet Franc is app	87	32	Virginia	Virginia		Alexander	Pearti	Qui矛vremont 2012 Meritage (Virginia)
22	Vin de Maison	87	23	Virginia	Virginia		Alexander	Pearti	Qui矛vremont 2012 Vin de Maison Red (Virginia)
23		87	20	Oregon	Oregon	Oregon	Paul Greg	@pau	Acrobat 2013 Pinot Noir (Oregon)
24	Ficiligno	87	19	Sicily & Sa	Sicilia		Kerin O鈥	@keri	Baglio di Pianetto 2007 Ficiligno White (Sicilia)
25	Signature Selection	87	22	California	Paso R	Central	Matt Kett	@mat	Bianchi 2011 Signature Selection Merlot (Paso Robles)
26	Aynat	87	35	Sicily & Sa	Sicilia		Kerin O鈥	@keri	Canicatt矛 2009 Aynat Nero d'Avola (Sicilia)
27	King Ridge Vineyard	87	69	California	Sonom	Sonoma	Virginie B	@vbo	Castello di Amorosa 2011 King Ridge Vineyard Pinot Noir (Sonoma Coast)
28	Dalila	87	13	Sicily & Sa	Terre Siciliane		Kerin O鈥	@keri	Stemmari 2013 Dalila White (Terre Siciliane)
29		87	10	Sicily & Sa	Terre Siciliane		Kerin O鈥	@keri	Stemmari 2013 Nero d'Avola (Terre Siciliane)
30	Mascaria Barricato	87	17	Sicily & Sa	Cerasuolo di Vi		Kerin O鈥	@keri	Terre di Giurfo 2011 Mascaria Barricato (Cerasuolo di Vittoria)

因此，可以使用正则表达式，提取 title 属性中年份以后，括号以前的内容，对 designation 属性的缺失值进行填充，核心代码如下：

```
    if type(row['designation']) == float and
math.isnan(row['designation']):
    designation = re.findall(r'[0-9][0-9][0-9][0-9](.*?)[(]', row['title'])
    if len(designation) > 0:
        data.loc[index, 'designation'] = designation[0]
```

6、 通过数据对象之间的相似性来填补缺失值

文件 wine_130k_similarity 对原文件 winemag-data-130k-v2.csv 中的缺失值使用数据对象间的相似性来填补缺失值，可视化结果显示在 fig_130k/similarity 中。

文件 wine_150k_similarity 对原文件 winemag-data_first150k.csv 中的缺失值使用数据对象间的相似性来填补缺失值，可视化结果显示在 fig_150k/similarity 中。

6.1 标称属性的相似性度量

标称属性的相似性度量采用参考书《Data Mining Concepts and Techniques》中 2.4.6 节的度量方法，如果标称属性的两个值相同，非相似性取 0；否则，非相似性取 1。核心代码如下：

```
    def nomi_similarity(word1, word2):
    if word1.strip() == word2.strip():
        return 0.0
    else:
        return 1.0
```

6.2 数值属性的相似性度量

数值属性的相似性度量采用参考书中 2.4.6 节的度量方法，本数据集中的 price 属于数值型属性，其距离度量公式为：

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$$

核心代码如下：

```
def num_similarity(r1, r2, max_f, min_f):
    if math.isnan(r1) or math.isnan(r2):
        return 0
    return abs(r1 - r2) / (max_f - min_f)
```

6.3 序列属性的相似性度量

序列属性的相似性度量参考书中 2.4.6 节的度量方法，本数据集中 points 属于序列性属性，其距离度量方式为：

首先，计算z_{if}：

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

之后，计算距离度量公式，我选择的是曼哈顿距离。核心代码如下：

```
def order_similarity(r1, r2, max_f):
    if math.isnan(r1) or math.isnan(r2):
        return 0
    z1 = (r1 - 1) / (max_f - 1)
    z2 = (r2 - 1) / (max_f - 1)
    return abs(z1 - z2)
```

6.4 混合属性的相似性度量

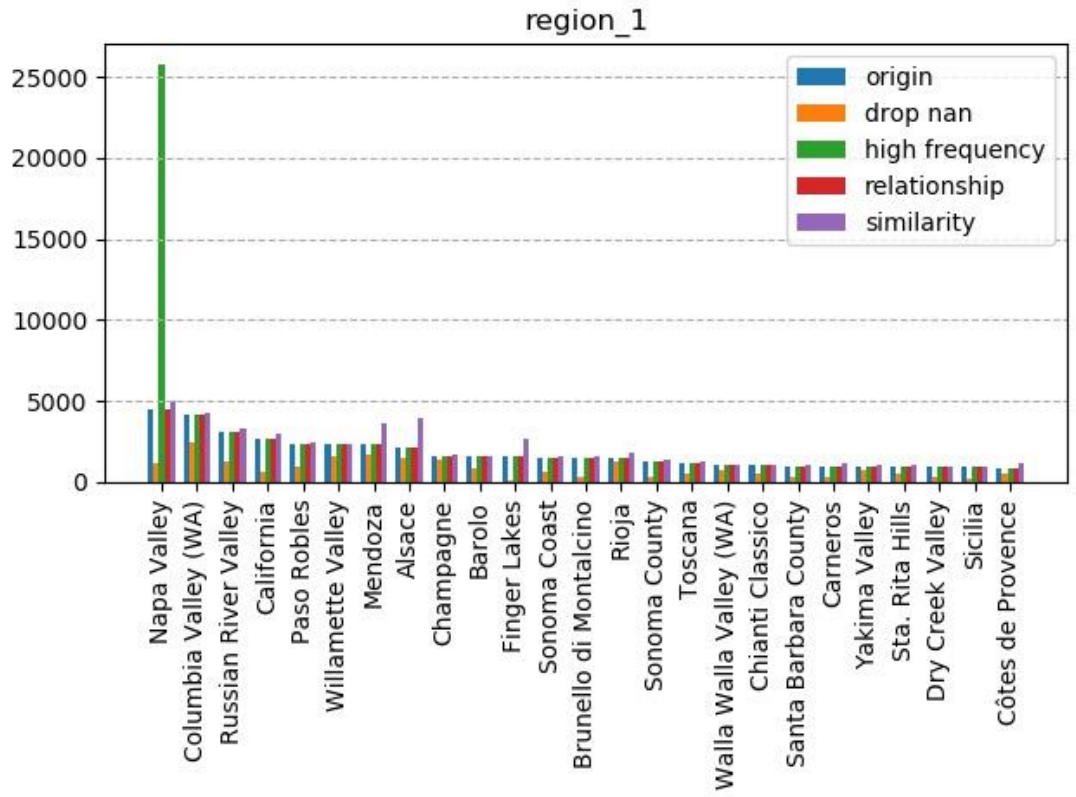
因为此数据集中，既有标称属性，又有数值属性，所以，使用参考书中 2.4.6 节将各个属性间的相似性进行相加，得到两条记录之间的非相似性：

$$\mathrm{d(i,j)} = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{f}}$$

7、 缺失数据处理后新旧数据集可视化对比

文件 wine_130k_contrast.py 对各种缺失值处理后的结果进行读取，将可视化结果存储在 fig_130k/contrast 中。

文件 wine_150k_high_frequency.py 对原文件 winemag-data_first150k.csv 中的缺失值使用最高频率进行填补，可视化结果显示在 fig_150k/high_frequency 中。



比如，对于 region_1 属性的可视化对比，可以看出通过直接删除空缺值的方法会导致数据数量大大降低，同时对数据的原本分布有较大影响。使用高频值填充的方法，对于空缺值较多的属性，可能导致最高频的值出现过多。使用相关关系和相似度的空缺值填充方法对于原数据集分布的影响较小。但是使用相关关系的方法可能找不到数据属性间的相关关系，使用范围有局限性。

Consumer & Visitor Insights For Neighborhoods

1、 数据集说明

本数据集包含

4 个标称属性

- census_block_group
- related_same_day_brand
- related_same_month_brand
- top_brands

3 个数值属性

- distance_from_home
- raw_visit_count
- raw_visitor_count

6 个不需可视化的属性：

- date_range_start（所有记录该值都相同）
- date_range_end（所有记录该值都相同）
- visitor_home_cbgs（所有记录该值均不同）
- visitor_work_cbgs（所有记录该值均不同）
- popularity_by_hour（所有该记录值均不同）
- popularity_by_day（所有该记录值均不同）

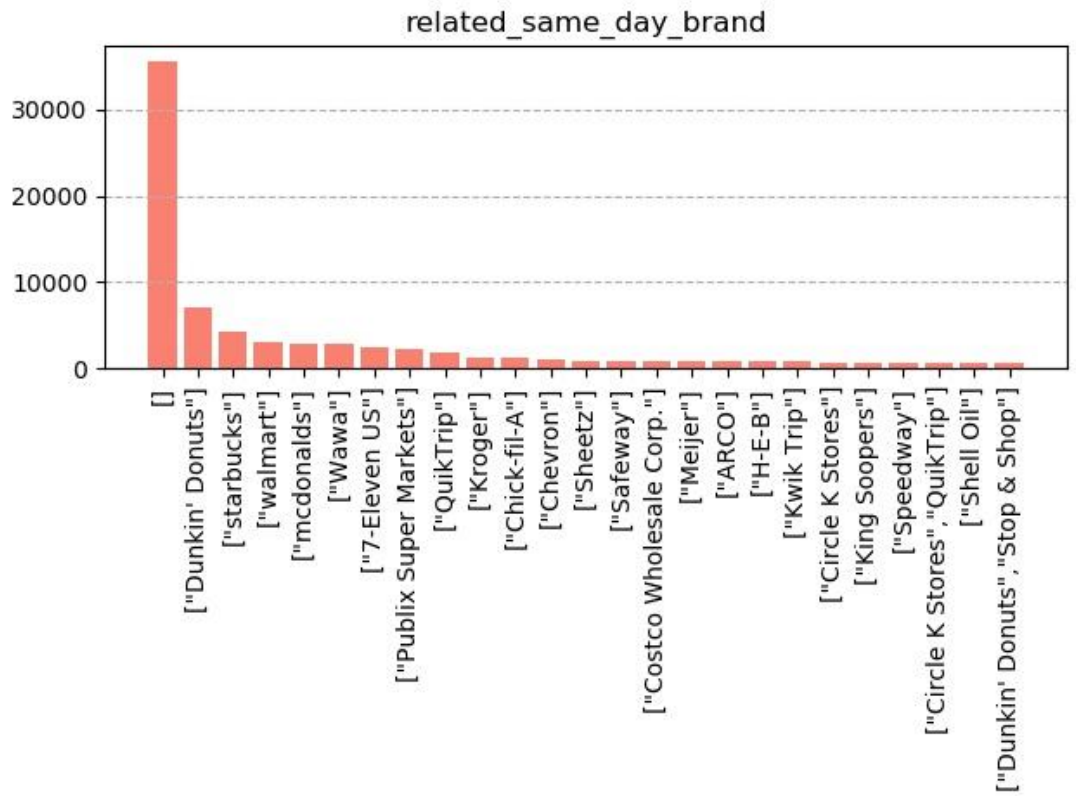
2、 原始数据集可视化

文件 visit_origin.py 中将原始数据集不加处理，进行可视化。

对标称属性中的 census_block_group 统计直方图，可以观察 census_block_group 的分布区间。对 related_same_day_brand，related_same_month_brand 和 top_brands 画柱状图，可以看出这三个标称属性各自对应的出现频率。

对 3 个数值属性进行 5 数概括，并绘制盒图，显示数值分布的区间范围。

结果显示在\fig_visit\origin 中。



可以看出，对于标称属性，由于有很多无效的空值，所以会影响结果的统计。

	A	B	C	D	E	F
1	name	minimum	Q1	median	Q3	maximum
2	raw_visit_count	60	nan	nan	nan	7179900
3	raw_visitor_count	50	nan	nan	nan	6113949
4	distance_from_home	706	nan	nan	nan	6297845
5						

从 5 数概括的结果也可以看出，由于没有去除空值，导致 Q1、median、Q3 值出现了 nan，无法对于数值属性的分布区间进行一个有效的展示。

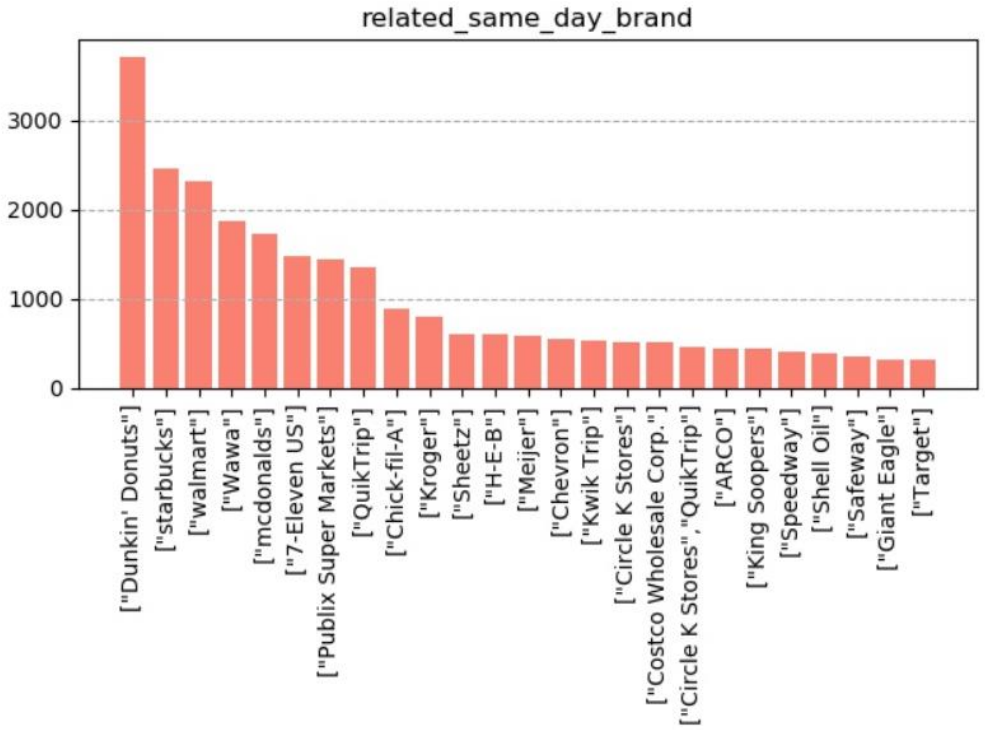
3、 将缺失部分剔除

文件 visit_dropnan.py 中直接将缺失值剔除，再对数据进行可视化。

对于标称属性，不仅仅是没有数据要算作缺失，空串（“{}”或者“[]”）也要算作属性值缺失。使用 dropnan 函数对标称属性的缺失进行具体判断，把所有需要剔除的行记录在 indexes 列表中，最后使用 pandas 自带的 drop 函数剔除缺失的行。

对于数值型属性，使用 pandas 自带的 dropna 函数，即可剔除缺失的行。

结果显示在\fig_visit\dropnan 中



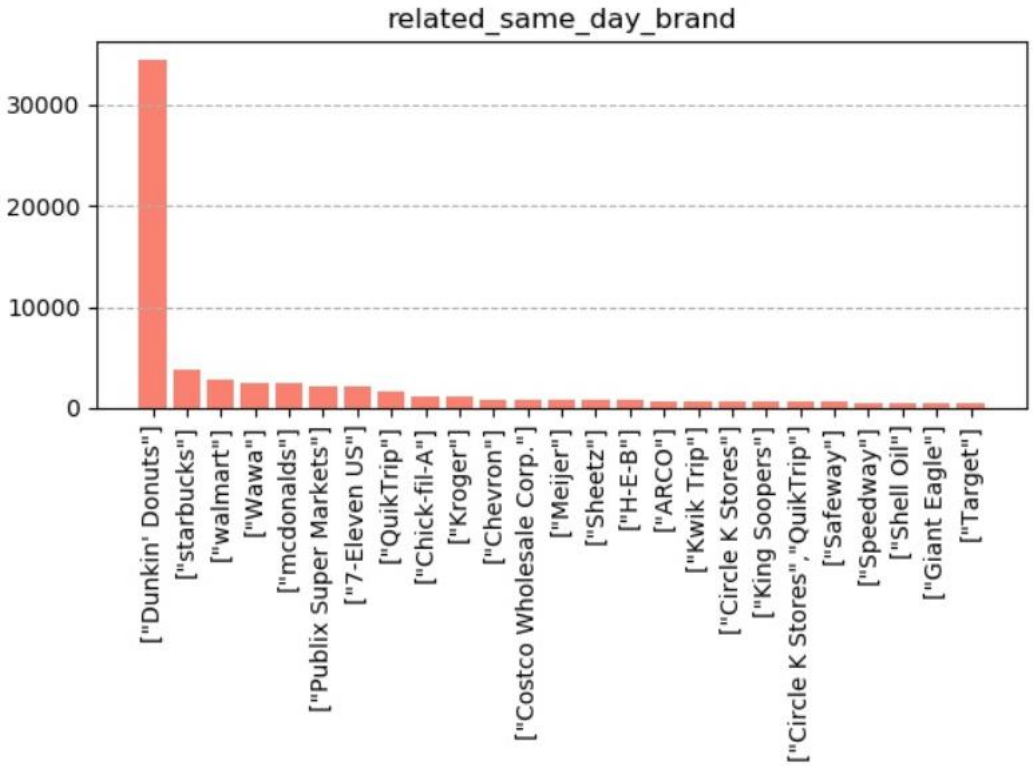
可以看出，剔除了空串，能够更明显的比对出属性的频率。

	A	B	C	D	E	F
1	name	minimum	Q1	median	Q3	maximum
2	raw_visit_count	2334	28142	43365.5	71893.25	1340323
3	raw_visitor_count	334	5621	9233.5	16366.75	353611
4	distance_from_home	1178	9826	16085	33875.75	4218226
5						

从五数概括中，可看出已经剔除了 nan 数据，能够明显判断出数值型数据所在的范围。但是最大最小值可能发生改变。

4、 用最高频率值来填补缺失值

文件 visit_high_frequency.py 统计除空值以外各个属性出现的最高频的值，并且使用高频值替换缺失数据。具体方法同第一个数据集。可视化结果保存在 fig_vis/high_frequency 中



可以看出，使用高频替换方法后，最高频的属性值的频数明显增加。使用高频替换的方法，数值型数据的最高值和最小值不会改变，但是 Q1、median 和 Q3 的值会有所改变。

	A	B	C	D	E	F	G
1	name	minimum	Q1	median	Q3	maximum	
2	raw_visit_count	2334	25496.25	40626.5	69938.75	7179900	
3	raw_visitor_count	334	5109.25	8828	16531	6113949	
4	distance_from_home	1178	9003	14403.5	28087	4218226	
5							

5、 通过属性的相关关系来填补缺失值

由于此数据集中，没有相关属性，所以无法通过属性的相关关系来填补缺失值。

6、 通过数据对象之间的相似性来填补缺失值

文件 visit_similarity.py 中使用数据对象之间的相似性来填补缺失值。此数据集中包含了标称属性和数值属性，具体的度量方法与数据集一中的方法相同，不再赘述。结果保存在 fig_vis/similarity 中。

比较特殊的是属性 popularity_by_hour，popularity_by_day 这两个属性，由于他们是字典的形式，所以不能直接按照标称属性的度量来计算，应该先转为数值型向量，再计算他们之间的余弦相似度，核心代码如下：

```
    def cos_similarity2(vec1, vec2):
    vec1 = vec1[1:-1].split(',')
    vec1 = [re.findall("\d+", v)[0] for v in vec1]
    vec1 = np.asarray(list(map(float, vec1)))

    vec2 = vec2[1:-1].split(',')
    vec2 = [re.findall("\d+", v)[0] for v in vec2]
    vec2 = np.asarray(list(map(float, vec2)))

    return np.sum(vec1 * vec2) / (np.linalg.norm(vec1, ord=2) *
np.linalg.norm(vec2, ord=2))
```

```
    def cos_similarity1(vec1, vec2):
    vec1 = vec1[1:-1].split(',')
    vec1 = np.asarray(list(map(float, vec1)))

    vec2 = vec2[1:-1].split(',')
    vec2 = np.asarray(list(map(float, vec2)))
    return np.sum(vec1 * vec2) / (np.linalg.norm(vec1, ord=2) *
np.linalg.norm(vec2, ord=2))
```

对于 top_brands、related_same_month_brand、related_same_day_brand 这三个标称属性，也不能直接使用标称属性的度量方式，应该将列表中的品牌逐一对比，得出相似度。核心代码如下：

```
    def brand_similarity(vec1, vec2):
    vec1 = vec1[1:-1].split(',')
    vec1 = [b.strip("\"") for b in vec1]

    vec2 = vec2[1:-1].split(',')
    vec2 = [b.strip("\"") for b in vec2]
    cnt = 0
    for b1 in vec1:
        for b2 in vec2:
            if b1 == b2:
                cnt += 1.0
    cnt /= max(len(vec1), len(vec2))
    return cnt
```

7、 新旧数据集可视化对比

结果呈现在 fig_vis/contrast 中。下图为一个标称属性的对比图和一个数值型属性的对比图。