```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('sample_data/hotel_bookings.csv', sep=",")
print(data.shape)
```

```
(119390, 32)
```

```
import os
```

```
os.getcwd()
```

```
'/content'
```

```
total_count = data.shape[0]
cat_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        cat_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.format(col
```

```
Колонка country. Тип данных object. Количество пустых значений 488, 0.41%.
```

```
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 119390
```

```
# Удаление колонок, содержащих пустые значения
data_new_1 = data.dropna(axis=1, how='any')
(data.shape, data_new_1.shape)
```

```
((119390, 32), (119390, 28))
```

Удалим строки, содержащие null значения

```
data_new_2 = data.dropna(axis=0, how='any')
(data.shape, data_new_2.shape)
```

```
((119390, 32), (217, 32))
```

```
data_new_2.head()
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arr
2392	Resort Hotel	0	6	2015	October	
2697	Resort Hotel	0	24	2015	October	
2867	Resort Hotel	0	24	2015	November	
2877	Resort Hotel	0	24	2015	November	
2878	Resort Hotel	0	24	2015	November	

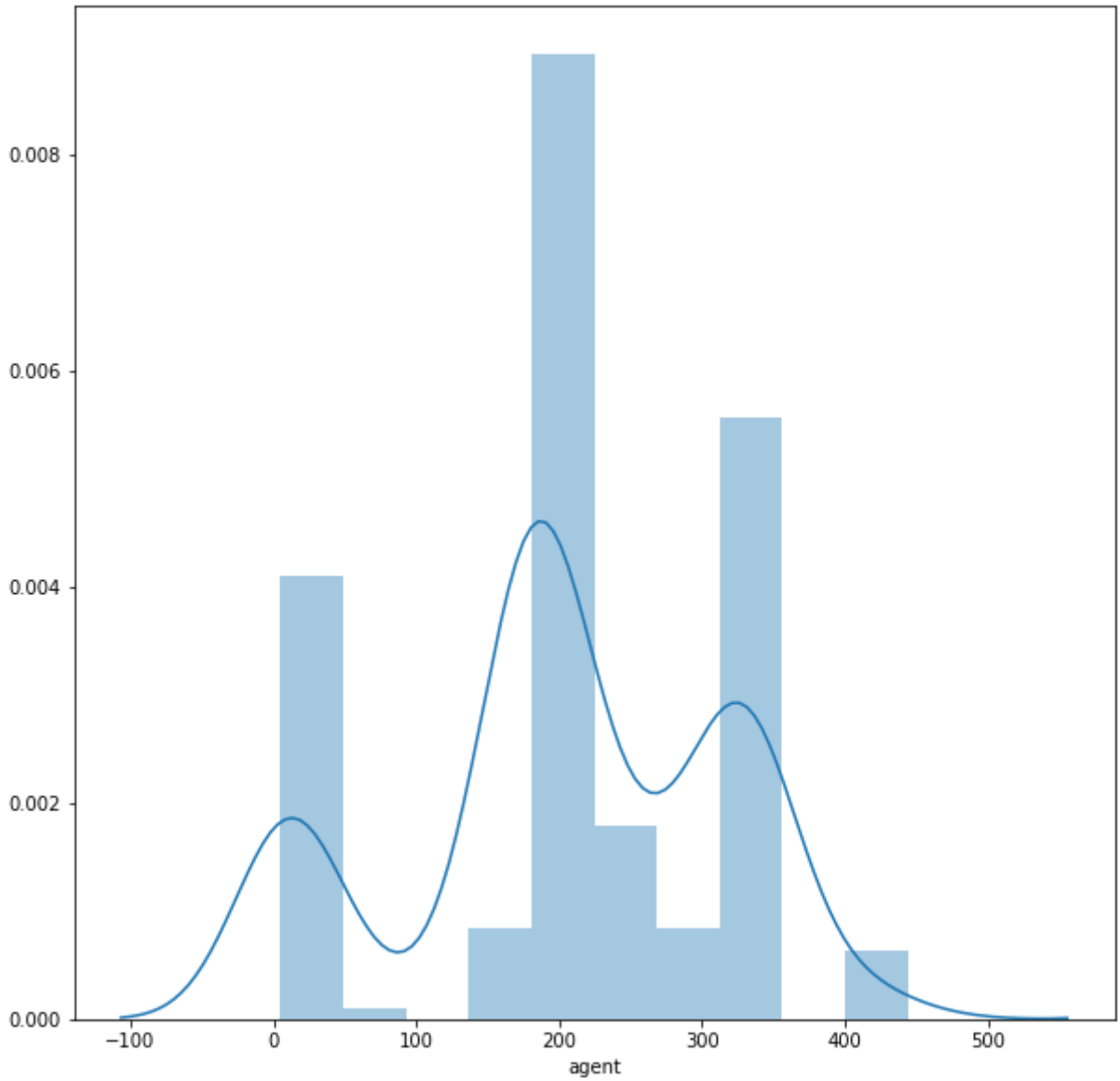
```
data_new_2.describe()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arr
count	217.000000	217.000000	217.000000	217.000000	
mean	0.078341	40.520737	2015.465438	38.198157	
std	0.269329	61.748375	0.720053	12.890292	
min	0.000000	0.000000	2015.000000	1.000000	
25%	0.000000	12.000000	2015.000000	33.000000	
50%	0.000000	27.000000	2015.000000	45.000000	
75%	0.000000	36.000000	2016.000000	46.000000	
max	1.000000	364.000000	2017.000000	53.000000	

Оценим плотность вероятности распределения данных

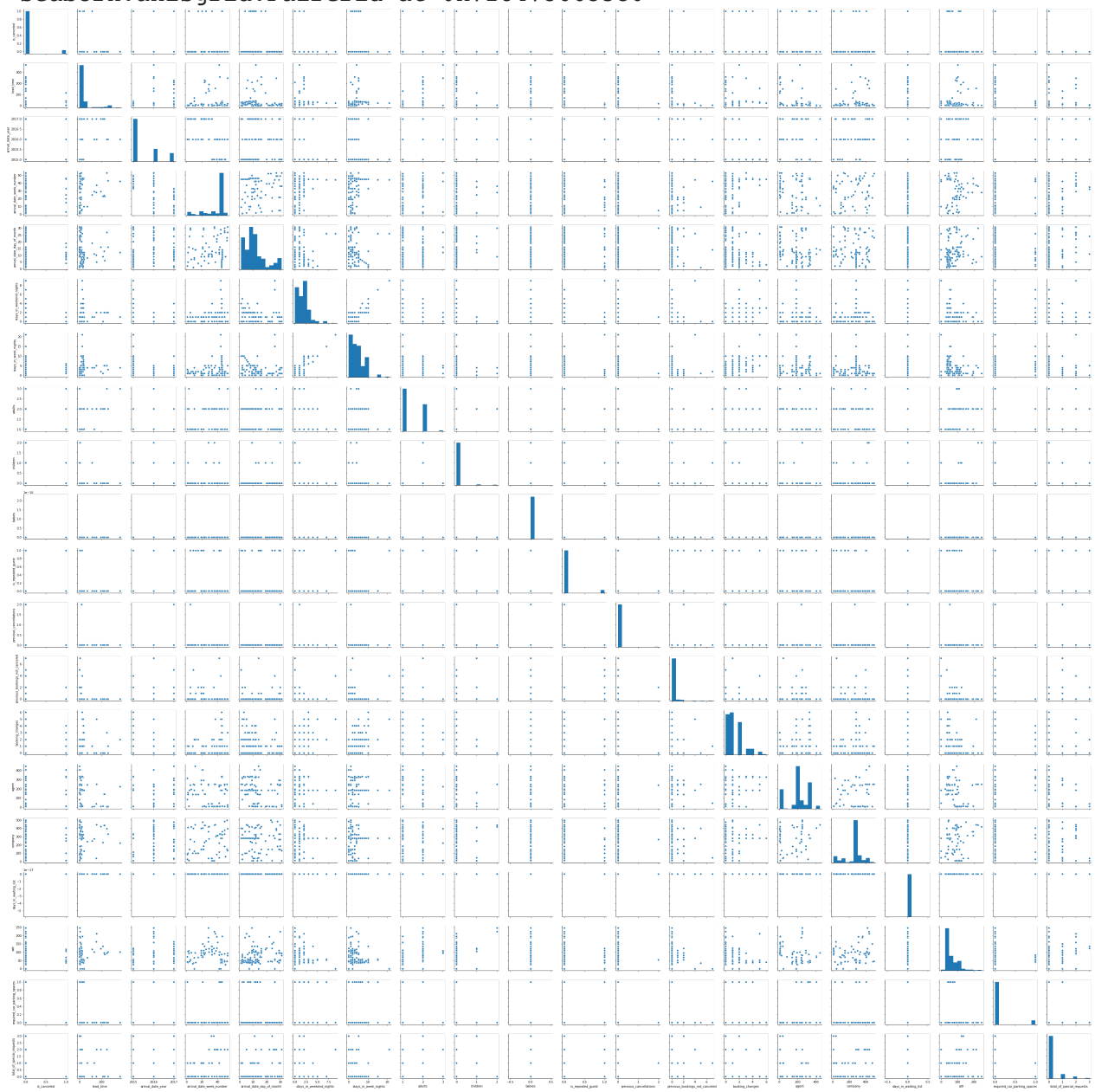
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data_new_2['agent'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8469858240>
```




```
sns.pairplot(data_new_2)
```

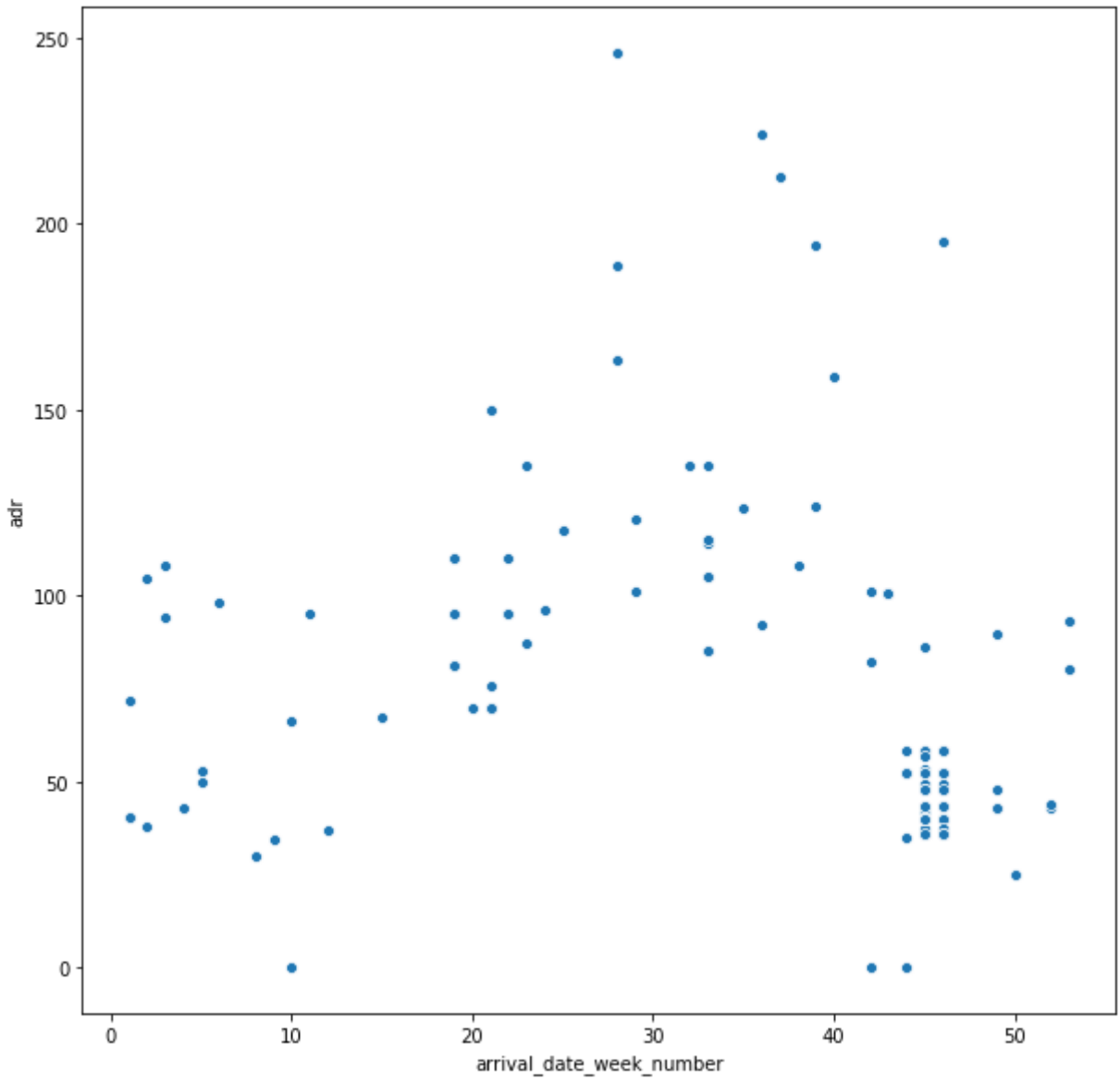
```
<seaborn.axisgrid.PairGrid at 0x7f84736635c0>
```



Находим почти линейную зависимость между значениями двух колонок: arrival_date_week_

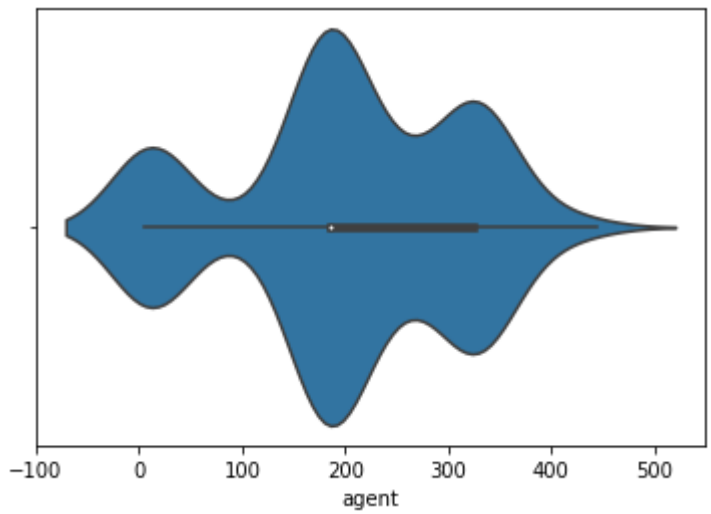
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='arrival_date_week_number', y='adr', data=data_new_2)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8469608da0>
```




```
sns.violinplot(x=data_new_2['agent'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f846951efd0>
```



Из приведенных графиков видно, что violinplot действительно показывает распределение

Корреляционный анализ

Построим корреляционную матрицу

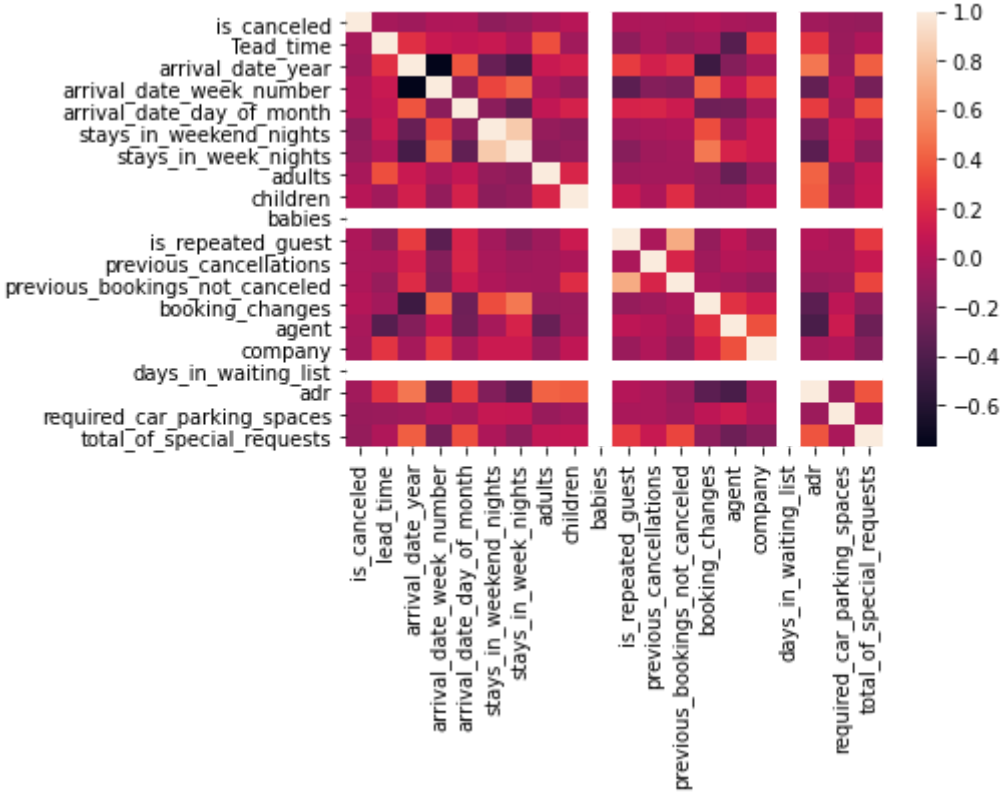
```
data_new_2.corr()
```

	is_canceled	lead_time	arrival_date_year	arriva
is_canceled	1.000000	-0.039767	-0.069527	
lead_time	-0.039767	1.000000	0.232138	
arrival_date_year	-0.069527	0.232138	1.000000	
arrival_date_week_number	-0.003159	0.106130	-0.768645	
arrival_date_day_of_month	0.002215	0.068615	0.367765	
stays_in_weekend_nights	-0.133996	0.097390	-0.293971	
stays_in_week_nights	-0.105148	-0.001527	-0.437521	
adults	-0.032118	0.345290	0.105956	
children	0.027563	-0.061248	0.145736	
babies	NaN	NaN	NaN	
is_repeated_guest	-0.011839	-0.136182	0.278636	
previous_cancellations	-0.019837	-0.024873	0.145344	
previous_bookings_not_canceled	-0.012140	-0.102228	0.206977	
booking_changes	0.015698	-0.052918	-0.482208	
agent	-0.034460	-0.369766	-0.175110	
company	-0.043769	0.249593	-0.037900	
days_in_waiting_list	NaN	NaN	NaN	
adr	-0.057069	0.240089	0.495254	
required_car_parking_spaces	-0.092895	-0.089066	-0.073376	
total_of_special_requests	-0.105983	-0.001975	0.399946	

Также построим матрицу корреляций по Пирсону
Так как значений довольно много, выберем матрицу без подписания числовых значений

```
sns.heatmap(data_new_2.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f846799fa20>
```



В примере тепловая карта помогает определить сильную корреляцию, например, между пр
stays_in_week_nights и *stays_in_weekeend_nights*, следовательно только один из этих признак
в модель.