In [63]:
```
import pandas as pd
```

In [64]:
```
import numpy as np
```

In [65]:
```
data = pd.read_excel("Online Retail.xlsx")
```

In [66]:
```
data.head()
```

Out[66]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

In [67]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 541909 entries, 0 to 541908
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype         
---  ------       --------------   -----         
 0   InvoiceNo    541909 non-null  object        
 1   StockCode    541909 non-null  object        
 2   Description  540455 non-null  object        
 3   Quantity     541909 non-null  int64         
 4   InvoiceDate  541909 non-null  datetime64[ns]
 5   UnitPrice    541909 non-null  float64       
 6   CustomerID   406829 non-null  float64       
 7   Country      541909 non-null  object        
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 33.1+ MB
```

In [68]:

```
data.isnull().sum()
```

Out[68]:

```
InvoiceNo           0
StockCode           0
Description      1454
Quantity            0
InvoiceDate         0
UnitPrice           0
CustomerID     135080
Country             0
dtype: int64
```

In [69]:

```
data = data.dropna(subset=['CustomerID'])
```

In [70]:

```
data.isnull().sum()
```

Out[70]:

```
InvoiceNo      0
StockCode      0
Description    0
Quantity       0
InvoiceDate    0
UnitPrice      0
CustomerID     0
Country        0
dtype: int64
```

In [71]:
```
data.duplicated().sum()
```

Out[71]:
```
np.int64(5225)
```

In [72]:
```
data= data.drop_duplicates()
```

In [73]:
```
data.describe()
```

Out[73]:

	Quantity	InvoiceDate	UnitPrice	CustomerID
count	401604.000000	401604	401604.000000	401604.000000
mean	12.183273	2011-07-10 12:08:23.848567552	3.474064	15281.160818
min	-80995.000000	2010-12-01 08:26:00	0.000000	12346.000000
25%	2.000000	2011-04-06 15:02:00	1.250000	13939.000000
50%	5.000000	2011-07-29 15:40:00	1.950000	15145.000000
75%	12.000000	2011-10-20 11:58:30	3.750000	16784.000000
max	80995.000000	2011-12-09 12:50:00	38970.000000	18287.000000
std	250.283037	NaN	69.764035	1714.006089

In [74]:
```
data = data[(data['Quantity'] > 0) & (data['UnitPrice'] >0)]
```

In [75]:
```
data.describe()
```

Out[75]:

	Quantity	InvoiceDate	UnitPrice	CustomerID
count	392692.000000	392692	392692.000000	392692.000000
mean	13.119702	2011-07-10 19:13:07.771892480	3.125914	15287.843865
min	1.000000	2010-12-01 08:26:00	0.001000	12346.000000
25%	2.000000	2011-04-07 11:12:00	1.250000	13955.000000
50%	6.000000	2011-07-31 12:02:00	1.950000	15150.000000
75%	12.000000	2011-10-20 12:53:00	3.750000	16791.000000
max	80995.000000	2011-12-09 12:50:00	8142.750000	18287.000000
std	180.492832	NaN	22.241836	1713.539549

In [76]:

```python
import datetime as dt
```

In [95]:

```python
def get_month(x) : return dt.datetime(x.year, x.month,1)
data['InvoiceMonth'] = data['InvoiceDate'].apply(get_month)
grouping = data.groupby('CustomerID')
data['Cohortmonth']= grouping['InvoiceMonth'].transform('min')
```

In [78]:

```python
def get_month_int(dframe, column):
    year = dframe[column].dt.year
    month = dframe[column].dt.month

    return year, month
```

In [79]:

```python
invoice_year , invoice_month = get_month_int(data, 'InvoiceMonth')
cohort_year , cohort_month = get_month_int(data, 'Cohortmonth')
```

In [80]:

```python
year_diff = invoice_year - cohort_year
month_diff = invoice_month-cohort_month
```

In [81]:

```python
data['cohort_age'] = year_diff * 12 + month_diff + 1
```

In [87]:

```python
data['Cohortmonth'] = data['Cohortmonth'].dt.strftime('%Y-%m-%d')
```

In [88]:

```
grouping = data.groupby(['Cohortmonth', 'cohort_age'])
cohort_data = grouping['CustomerID'].apply(pd.Series.nunique)
```

In [89]:

```
cohort_data = cohort_data.reset_index()
cohort_counts = cohort_data.pivot(index='Cohortmonth',columns='cohort_age',values='CustomerID')
```

In [90]:

```
cohort_size = cohort_counts.iloc[:,0]
retention = cohort_counts.divide(cohort_size,axis=0)
retention.round(3) * 100
```

Out[90]:

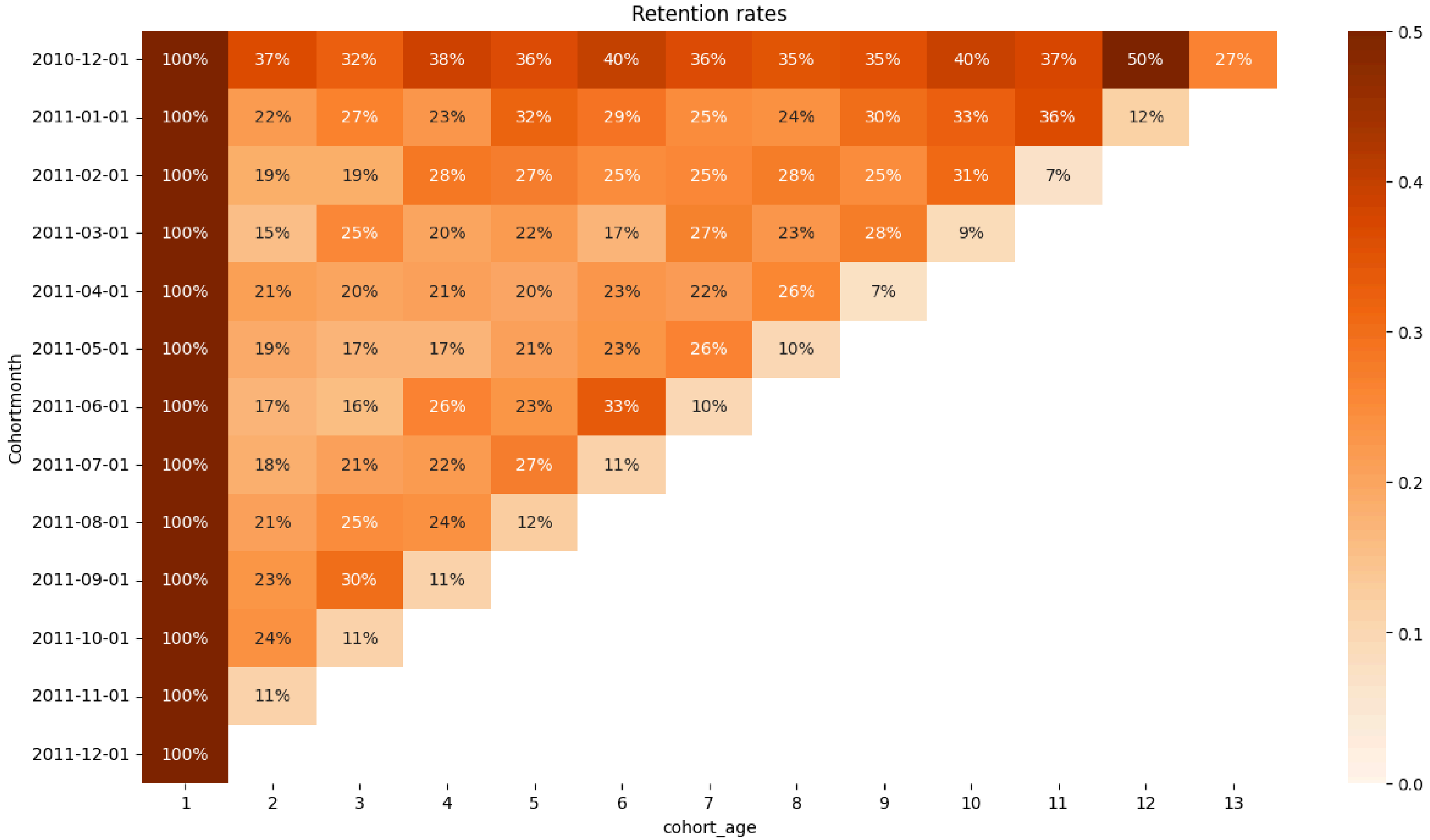
cohort_age	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
Cohortmonth													
2010-12-01	100.0	36.6	32.3	38.4	36.3	39.8	36.3	34.9	35.4	39.5	37.4	50.3	26.6
2011-01-01	100.0	22.1	26.6	23.0	32.1	28.8	24.7	24.2	30.0	32.6	36.5	11.8	NaN
2011-02-01	100.0	18.7	18.7	28.4	27.1	24.7	25.3	27.9	24.7	30.5	6.8	NaN	NaN
2011-03-01	100.0	15.0	25.2	19.9	22.3	16.8	26.8	23.0	27.9	8.6	NaN	NaN	NaN
2011-04-01	100.0	21.3	20.3	21.0	19.7	22.7	21.7	26.0	7.3	NaN	NaN	NaN	NaN
2011-05-01	100.0	19.0	17.3	17.3	20.8	23.2	26.4	9.5	NaN	NaN	NaN	NaN	NaN
2011-06-01	100.0	17.4	15.7	26.4	23.1	33.5	9.5	NaN	NaN	NaN	NaN	NaN	NaN
2011-07-01	100.0	18.1	20.7	22.3	27.1	11.2	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-08-01	100.0	20.7	24.9	24.3	12.4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-09-01	100.0	23.4	30.1	11.4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-10-01	100.0	24.0	11.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-11-01	100.0	11.1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2011-12-01	100.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [91]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [93]:

```
plt.figure(figsize=(15, 8))
plt.title('Retention rates')
sns.heatmap(data=retention,annot = True,fmt = '.0%',vmin = 0.0,vmax = 0.5,cmap="Oranges")
plt.show()
```



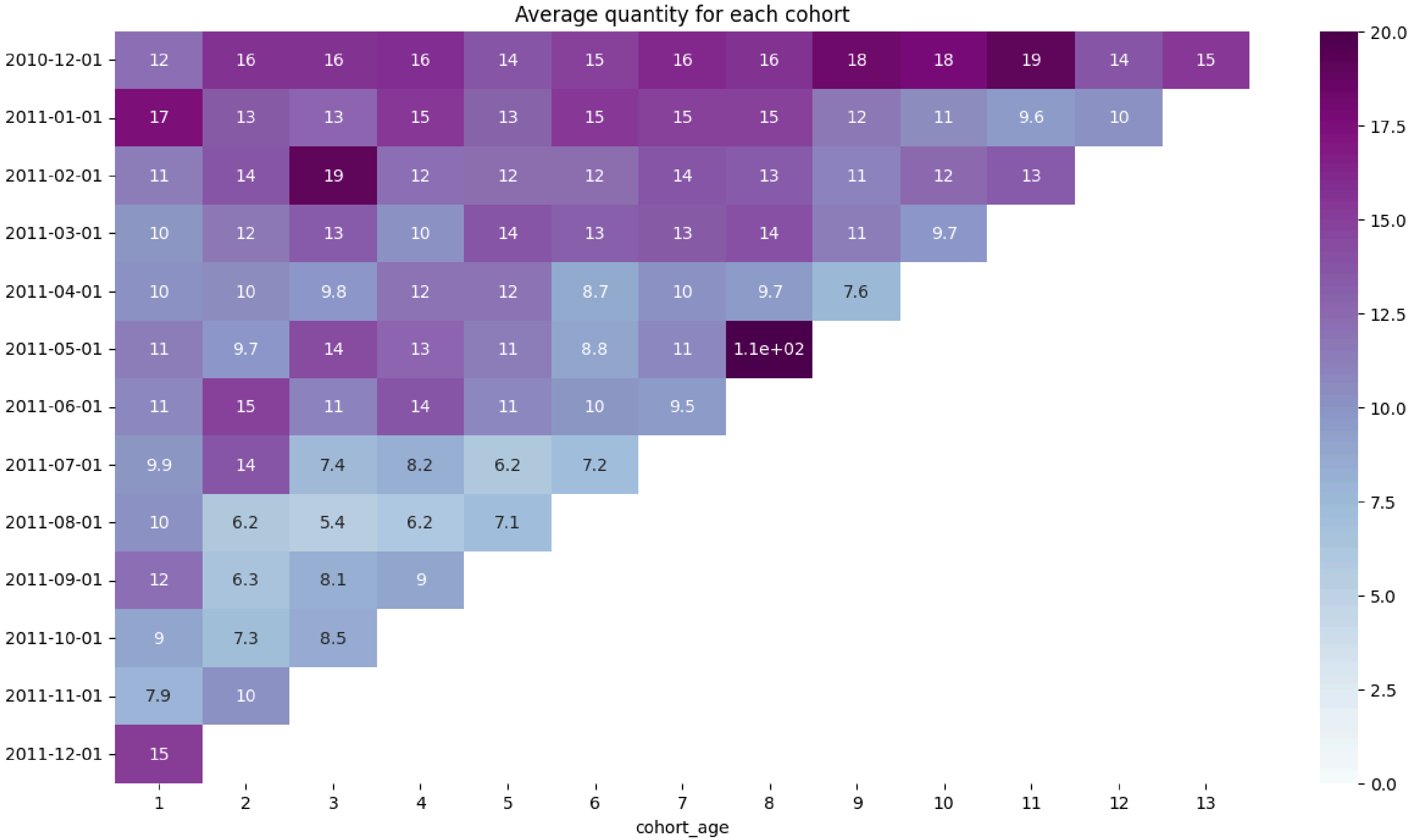
In [97]:

```
grouping = data.groupby(['Cohortmonth', 'cohort_age'])
cohort_data = grouping['Quantity'].mean()
cohort_data = cohort_data.reset_index()
average_quantity = cohort_data.pivot(index='Cohortmonth',columns='cohort_age',values='Quantity')
```

```
average_quantity.round(1)
average_quantity.index = average_quantity.index.date
```

In [98]:

```
plt.figure(figsize=(15, 8))
plt.title('Average quantity for each cohort')
sns.heatmap(data=average_quantity,annot = True,vmin = 0.0,vmax =20,cmap="BuPu")
plt.show()
```


Average quantity for each cohort
2011-05-01 11 9.7 14 13 11 8.8 11 1.1e+02
cohort_age