In [1]:
```
# loading libraries
import gzip
import json
!pip install --user pandas==1.0.3
import pandas as pd
from pandas.io.json import json_normalize
pd.__version__
import matplotlib.pyplot as plt
import seaborn as sns

file = gzip.open("C:/Users/amals/Downloads/receipts.json.gz", "rb")
data = file.read()
```

```
Requirement already satisfied: pandas==1.0.3 in c:\users\amals\appdata\roamin
g\python\python37\site-packages (1.0.3)
Requirement already satisfied: numpy>=1.13.3 in c:\users\amals\anaconda3\lib
\site-packages (from pandas==1.0.3) (1.16.4)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\amals\anaco
nda3\lib\site-packages (from pandas==1.0.3) (2.8.0)
Requirement already satisfied: pytz>=2017.2 in c:\users\amals\anaconda3\lib\s
ite-packages (from pandas==1.0.3) (2019.1)
Requirement already satisfied: six>=1.5 in c:\users\amals\anaconda3\lib\site-
packages (from python-dateutil>=2.6.1->pandas==1.0.3) (1.12.0)

C:\Users\amals\Anaconda3\lib\site-packages\statsmodels\tools\_testing.py:19:
FutureWarning: pandas.util.testing is deprecated. Use the functions in the pu
blic API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [2]:
```
# Printing data
# data
```

In [3]:
```
# loading receipt data
import pandas as pd
from ast import literal_eval
fn = 'C:/Users/amals/Downloads/receipts.json.gz'
df1 = pd.read_json(fn, lines=True, compression='gzip')
df1 = pd.DataFrame(df1)
df1
```

Out[3]:

	_id	bonusPointsEarned	bonusPointsEarnedReason	
0	{'$oid': '5ff1e1eb0a720f0523000575'}	500.0	Receipt number 2 completed, bonus point schedu...	1609(
1	{'$oid': '5ff1e1bb0a720f052300056b'}	150.0	Receipt number 5 completed, bonus point schedu...	1609(
2	{'$oid': '5ff1e1f10a720f052300057a'}	5.0	All-receipts receipt bonus	1609(
3	{'$oid': '5ff1e1ee0a7214ada100056f'}	5.0	All-receipts receipt bonus	1609(
4	{'$oid': '5ff1e1d20a7214ada1000561'}	5.0	All-receipts receipt bonus	1609(
...	...	...	...	
1114	{'$oid': '603cc0630a720fde100003e6'}	25.0	COMPLETE_NONPARTNER_RECEIPT	1614(
1115	{'$oid': '603d0b710a720fde1000042a'}	NaN	NaN	1614(
1116	{'$oid': '603cf5290a720fde10000413'}	NaN	NaN	1614(
1117	{'$oid': '603ce7100a7217c72c000405'}	25.0	COMPLETE_NONPARTNER_RECEIPT	1614(
1118	{'$oid': '603c4fea0a7217c72c000389'}	NaN	NaN	1614(

1119 rows × 15 columns

In [4]:
```
# checking data type
df1.dtypes
```

Out[4]:
```
_id                        object
bonusPointsEarned         float64
bonusPointsEarnedReason    object
createDate                 object
dateScanned                object
finishedDate               object
modifyDate                 object
pointsAwardedDate          object
pointsEarned              float64
purchaseDate               object
purchasedItemCount        float64
rewardsReceiptItemList     object
rewardsReceiptStatus       object
totalSpent                float64
userId                     object
dtype: object
```

In [5]:
```
# Normalizing Json format
df1['_id'] = pd.json_normalize(df1['_id'])
df1['createDate'] = pd.json_normalize(df1['createDate'])
df1['dateScanned'] = pd.json_normalize(df1['dateScanned'])
df1['modifyDate'] = pd.json_normalize(df1['modifyDate'])
```

In [6]:
```
df1
```

Out[6]:

	_id	bonusPointsEarned	bonusPointsEarnedReason	cr
0	5ff1e1eb0a720f0523000575	500.0	Receipt number 2 completed, bonus point schedu...	160968
1	5ff1e1bb0a720f052300056b	150.0	Receipt number 5 completed, bonus point schedu...	160968
2	5ff1e1f10a720f052300057a	5.0	All-receipts receipt bonus	160968
3	5ff1e1ee0a7214ada100056f	5.0	All-receipts receipt bonus	160968
4	5ff1e1d20a7214ada1000561	5.0	All-receipts receipt bonus	160968
...	...	...	...	
1114	603cc0630a720fde100003e6	25.0	COMPLETE_NONPARTNER_RECEIPT	161459
1115	603d0b710a720fde1000042a	NaN	NaN	161461
1116	603cf5290a720fde10000413	NaN	NaN	161460
1117	603ce7100a7217c72c000405	25.0	COMPLETE_NONPARTNER_RECEIPT	161460
1118	603c4fea0a7217c72c000389	NaN	NaN	161456

1119 rows × 15 columns

In [7]:
```
# Checking data type after normalization
df1.dtypes
```

Out[7]:
```
_id                       object
bonusPointsEarned        float64
bonusPointsEarnedReason   object
createDate                 int64
dateScanned                int64
finishedDate              object
modifyDate                 int64
pointsAwardedDate         object
pointsEarned             float64
purchaseDate              object
purchasedItemCount       float64
rewardsReceiptItemList    object
rewardsReceiptStatus      object
totalSpent               float64
userId                    object
dtype: object
```

In [8]:
```
# Loading brand data
import pandas as pd
fn = 'C:/Users/amals/Downloads/brands.json.gz'
df2 = pd.read_json(fn, lines=True, compression='gzip')
df2
```

Out[8]:

—

In [9]:
```
# Loading users data
fn = 'C:/Users/amals/Downloads/users.json.gz'
df3 = pd.read_json(fn, lines=True, compression='gzip')
df3
```

Out[9]:

	_id	active	createdDate	lastLogin	role	signUpSour
0	{'$oid': '5ff1e194b6a9d73a3a9f1052'}	True	{'$date': 1609687444800}	{'$date': 1609687537858}	consumer	Em
1	{'$oid': '5ff1e194b6a9d73a3a9f1052'}	True	{'$date': 1609687444800}	{'$date': 1609687537858}	consumer	Em
2	{'$oid': '5ff1e194b6a9d73a3a9f1052'}	True	{'$date': 1609687444800}	{'$date': 1609687537858}	consumer	Em
3	{'$oid': '5ff1e1eacfcf6c399c274ae6'}	True	{'$date': 1609687530554}	{'$date': 1609687530597}	consumer	Em
4	{'$oid': '5ff1e194b6a9d73a3a9f1052'}	True	{'$date': 1609687444800}	{'$date': 1609687537858}	consumer	Em
...	...	...	...	...	...	
490	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	N
491	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	N
492	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	N
493	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	N
494	{'$oid': '54943462e4b07e684157a532'}	True	{'$date': 1418998882381}	{'$date': 1614963143204}	fetch-staff	N

495 rows × 7 columns

In [10]:
```
# Normalizing Json format
df3['_id'] = pd.json_normalize(df3['_id'],errors="ignore")
df3['createdDate'] = pd.json_normalize(df3['createdDate'],errors="ignore")
```

In [11]:
```
df3
```

Out[11]:

	_id	active	createdDate	lastLogin	role	signUpSource
0	5ff1e194b6a9d73a3a9f1052	True	1609687444800	{'$date': 1609687537858}	consumer	Emai
1	5ff1e194b6a9d73a3a9f1052	True	1609687444800	{'$date': 1609687537858}	consumer	Emai
2	5ff1e194b6a9d73a3a9f1052	True	1609687444800	{'$date': 1609687537858}	consumer	Emai
3	5ff1e1eacfcf6c399c274ae6	True	1609687530554	{'$date': 1609687530597}	consumer	Emai
4	5ff1e194b6a9d73a3a9f1052	True	1609687444800	{'$date': 1609687537858}	consumer	Emai
...	...	...	...	...	...	..
490	54943462e4b07e684157a532	True	1418998882381	{'$date': 1614963143204}	fetch-staff	NaN
491	54943462e4b07e684157a532	True	1418998882381	{'$date': 1614963143204}	fetch-staff	NaN
492	54943462e4b07e684157a532	True	1418998882381	{'$date': 1614963143204}	fetch-staff	NaN
493	54943462e4b07e684157a532	True	1418998882381	{'$date': 1614963143204}	fetch-staff	NaN
494	54943462e4b07e684157a532	True	1418998882381	{'$date': 1614963143204}	fetch-staff	NaN

495 rows × 7 columns

In [12]:
```
# Checking Null values for receipt data
df1.isna().sum()
```

Out[12]:
```
_id                          0
bonusPointsEarned          575
bonusPointsEarnedReason    575
createDate                   0
dateScanned                  0
finishedDate               551
modifyDate                   0
pointsAwardedDate          582
pointsEarned               510
purchaseDate               448
purchasedItemCount         484
rewardsReceiptItemList     440
rewardsReceiptStatus         0
totalSpent                 435
userId                       0
dtype: int64
```

In [13]:
```
df1.shape[0]
```

Out[13]:
```
1119
```

After Checking Null values for receipt data, we can say that almost 50% of the data related to bonus point, purchase date, Purchase count and total price is missing. There data attributes are critical and import for tracking

In [14]:
```
# Checking Null values for users data
df3.isna().sum()
```

Out[14]:
```
_id              0
active           0
createdDate      0
lastLogin       62
role             0
signUpSource    48
state           56
dtype: int64
```

In [15]:
```
df3.shape[0]
```

Out[15]:
```
495
```

In user data, almost 10% of the data is missing related to state and signup source

Data Exploration for Receipt Data

In [16]:
```
df1.describe()
```

Out[16]:

	bonusPointsEarned	createDate	dateScanned	modifyDate	pointsEarned	purchasedIt
count	544.000000	1.119000e+03	1.119000e+03	1.119000e+03	609.000000	6
mean	238.893382	1.611800e+12	1.611800e+12	1.611847e+12	585.962890	
std	299.091731	1.484091e+09	1.484091e+09	1.361576e+09	1357.166947	
min	5.000000	1.604089e+12	1.604089e+12	1.609687e+12	0.000000	
25%	5.000000	1.610652e+12	1.610652e+12	1.610660e+12	5.000000	
50%	45.000000	1.611941e+12	1.611941e+12	1.611941e+12	150.000000	
75%	500.000000	1.612704e+12	1.612704e+12	1.612704e+12	750.000000	
max	750.000000	1.614641e+12	1.614641e+12	1.614641e+12	10199.800000	6

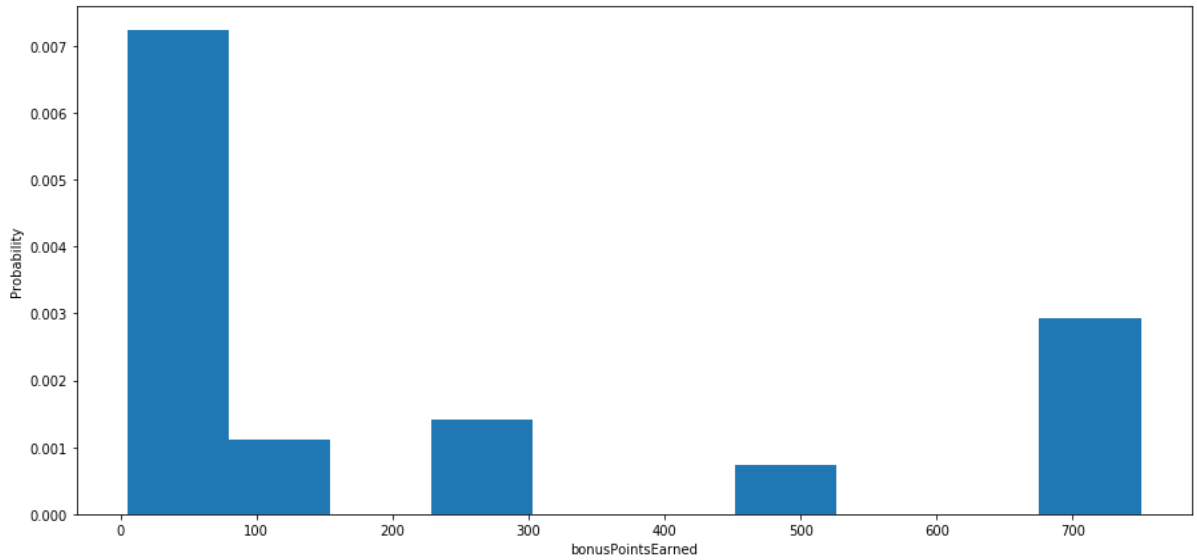
In [17]:
```
# Counting data for bonusPointsEarned category
df1["bonusPointsEarned"].value_counts()
```

Out[17]:
```
5.0      183
750.0    119
25.0      71
45.0      31
250.0     31
500.0     30
150.0     27
300.0     26
100.0     18
27.0       6
21.0       1
40.0       1
Name: bonusPointsEarned, dtype: int64
```

In [18]:
```
# Plotting graph for bonusPointsEarned
plt.figure(figsize=(15,7))
plt.hist(df1["bonusPointsEarned"], density=True)
plt.ylabel('Probability')
plt.xlabel('bonusPointsEarned')
```

```
C:\Users\amals\Anaconda3\lib\site-packages\numpy\lib\histograms.py:824: Runti
meWarning: invalid value encountered in greater_equal
  keep = (tmp_a >= first_edge)
C:\Users\amals\Anaconda3\lib\site-packages\numpy\lib\histograms.py:825: Runti
meWarning: invalid value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```

Out[18]:
```
Text(0.5, 0, 'bonusPointsEarned')
```



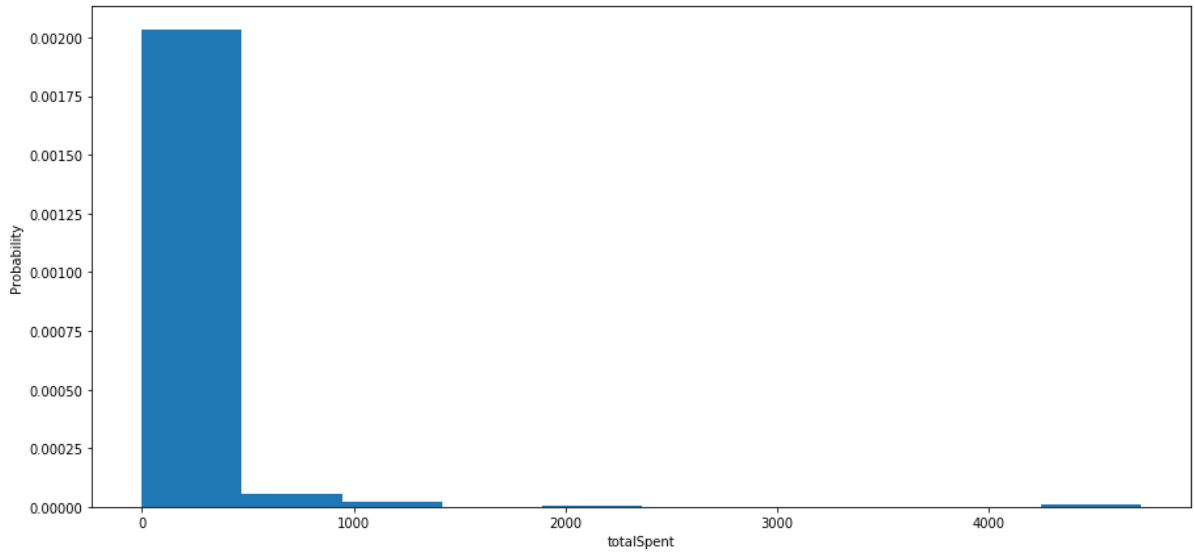
We can say that most of the data points are concentrated in between 0-100 bonus Points

In [19]:
```
# Counting data for total spent category
df1["totalSpent"].value_counts()
```

Out[19]:
```
1.00      172
10.00      54
28.57      50
34.96      44
49.95      43
          ...
427.81      1
612.95      1
271.63      1
99.95       1
574.65      1
Name: totalSpent, Length: 94, dtype: int64
```

In [20]:
```
# Plotting graph for total spent
plt.figure(figsize=(15,7))
plt.hist(df1["totalSpent"], density=True)
plt.ylabel('Probability')
plt.xlabel('totalSpent')
```

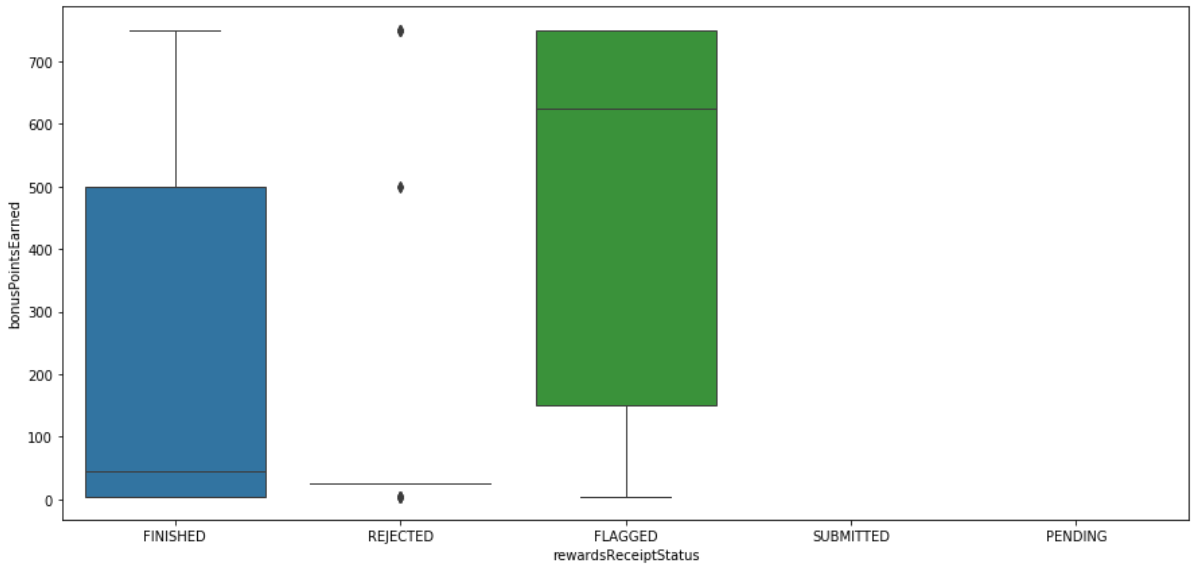
Out[20]:
```
Text(0.5, 0, 'totalSpent')
```



We can say that most of the data points are concentrated in between 0-500 dollars

In [21]:
```
# Plotting data based on rewardsReceiptStatus
plt.figure(figsize=(15,7))
sns.boxplot(x="rewardsReceiptStatus", y="bonusPointsEarned",
                  data=df1, linewidth=1, dodge=True)
```

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0x22202210eb8>

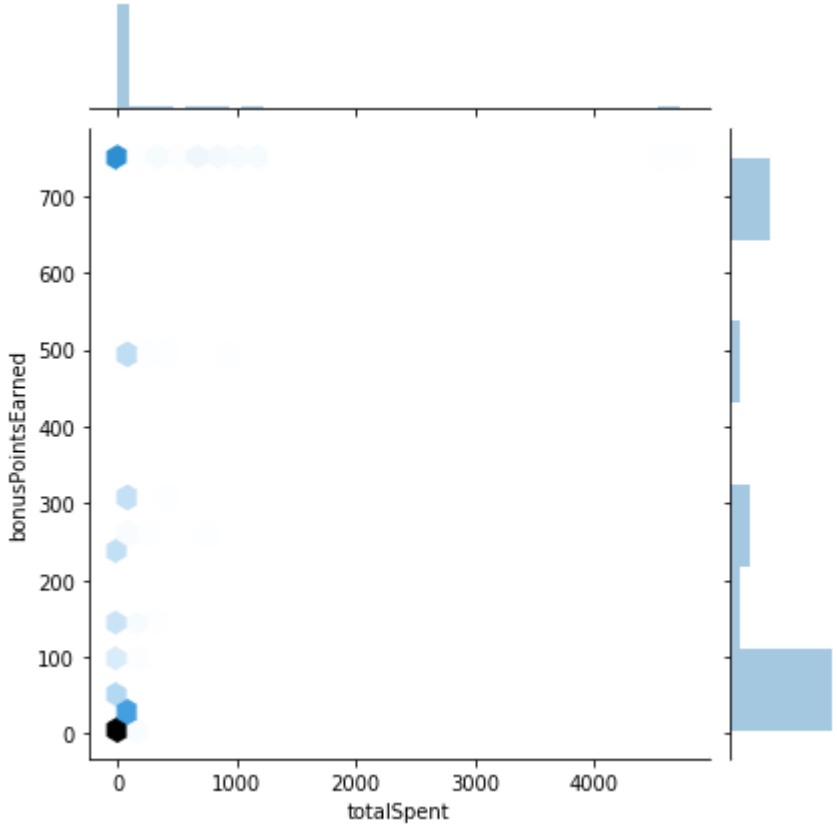


Here we can say that interquartile range is high. Median bonus points is higher for Flagged receipts, rejected receipt status has only 3 points whereas submitted & pending has zero data points

In [22]:
```
# Checking Relation between bonusPointsEarned and total spent
plt.figure(figsize=(15,10))
sns.jointplot(data=df1, x="totalSpent", y="bonusPointsEarned", kind="hex")
```

Out[22]:
```
<seaborn.axisgrid.JointGrid at 0x222024c2f98>

<Figure size 1080x720 with 0 Axes>
```

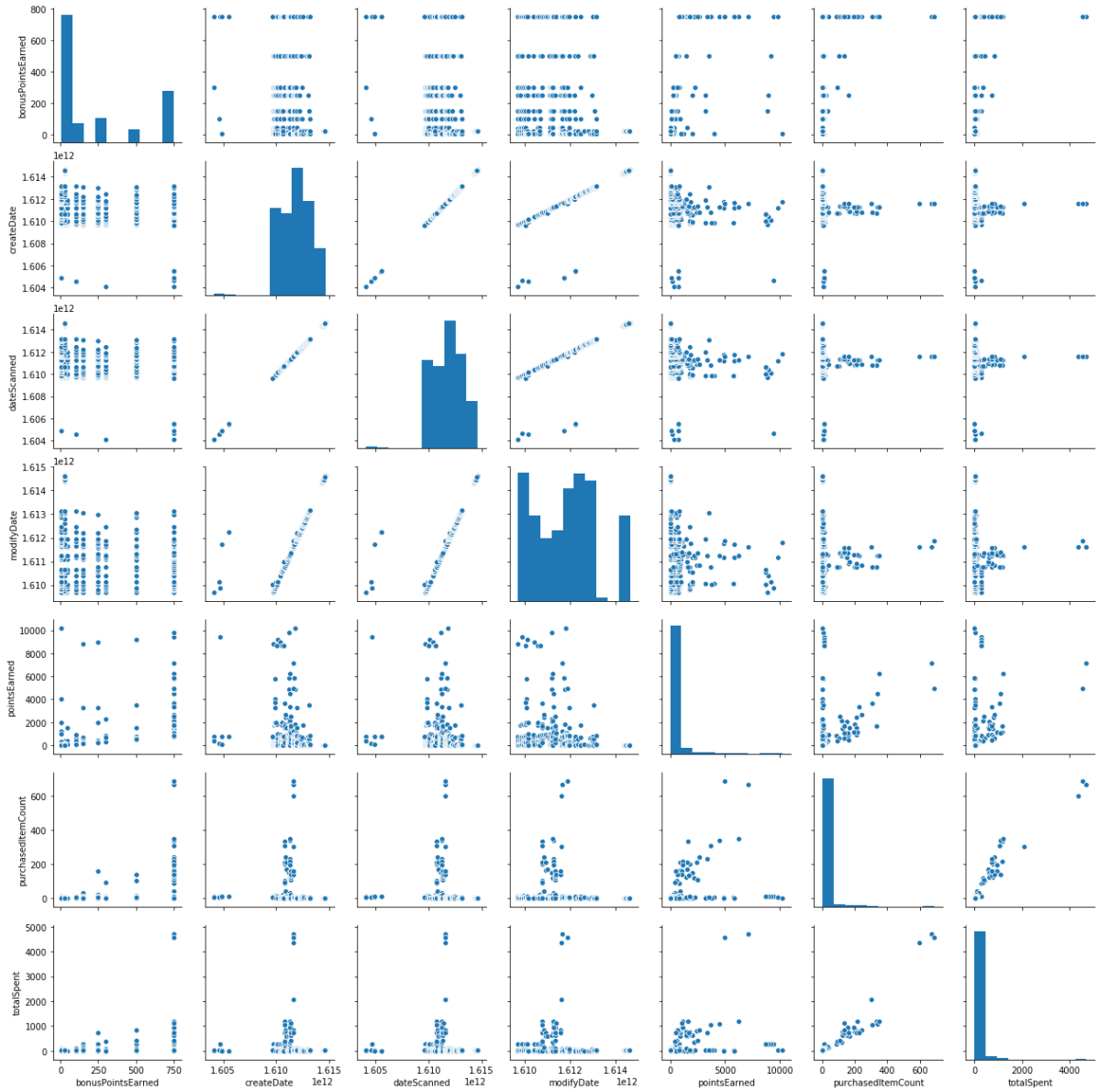


Correlation between total spent and bonus point is strong in between 0-50

In [23]:
```
# Checking data distribution
sns.pairplot(df1)
```

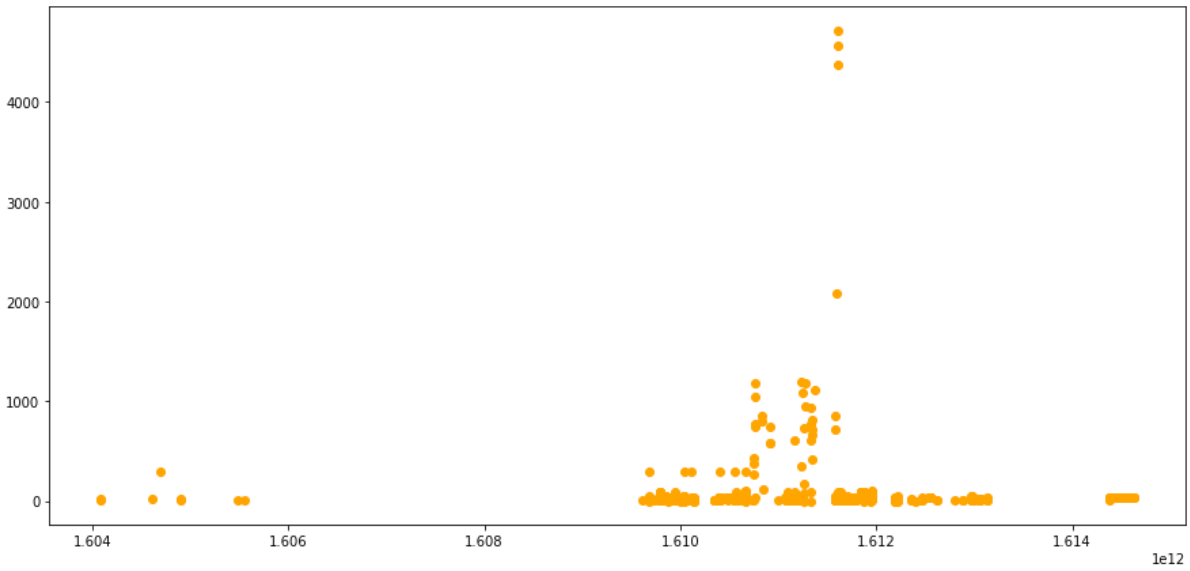
```
C:\Users\amals\Anaconda3\lib\site-packages\numpy\lib\histograms.py:824: Runti
meWarning: invalid value encountered in greater_equal
  keep = (tmp_a >= first_edge)
C:\Users\amals\Anaconda3\lib\site-packages\numpy\lib\histograms.py:825: Runti
meWarning: invalid value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```

Out[23]: <seaborn.axisgrid.PairGrid at 0x2227f997390>



In [24]:
```
# plotting total spent distribution based on created date
plt.figure(figsize=(15,7))
plt.scatter(df1["createDate"], df1["totalSpent"], Color = 'Orange')
```

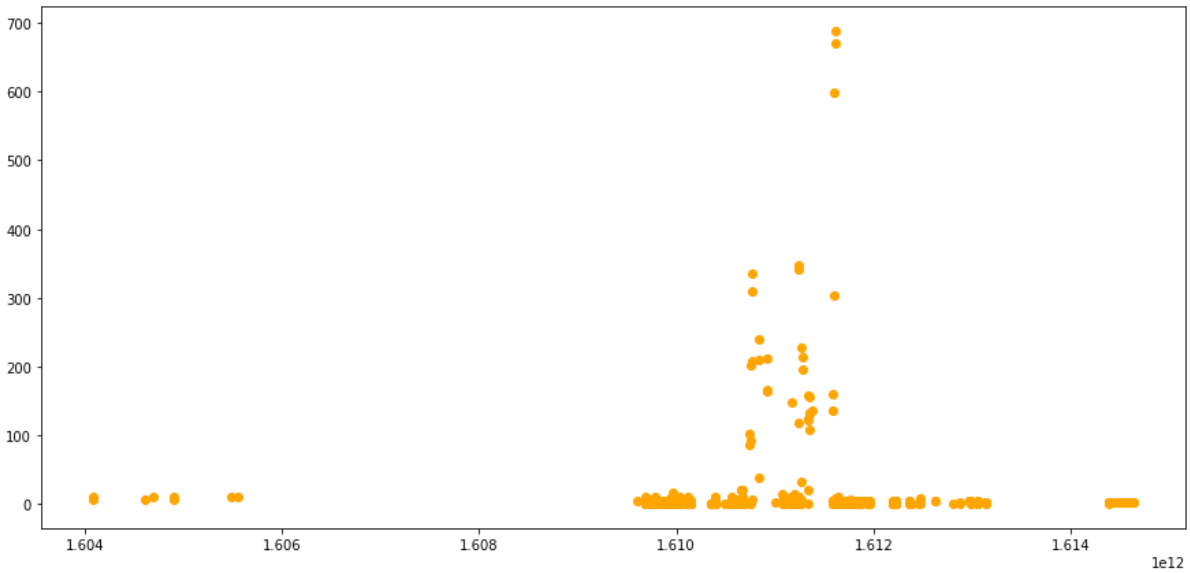
Out[24]: <matplotlib.collections.PathCollection at 0x22203bb43c8>



Maximum purchase happened between 1610 and 1614 and Format is unknown. It need to be convered to YYYY-MM-DD format to know the exact timeline.

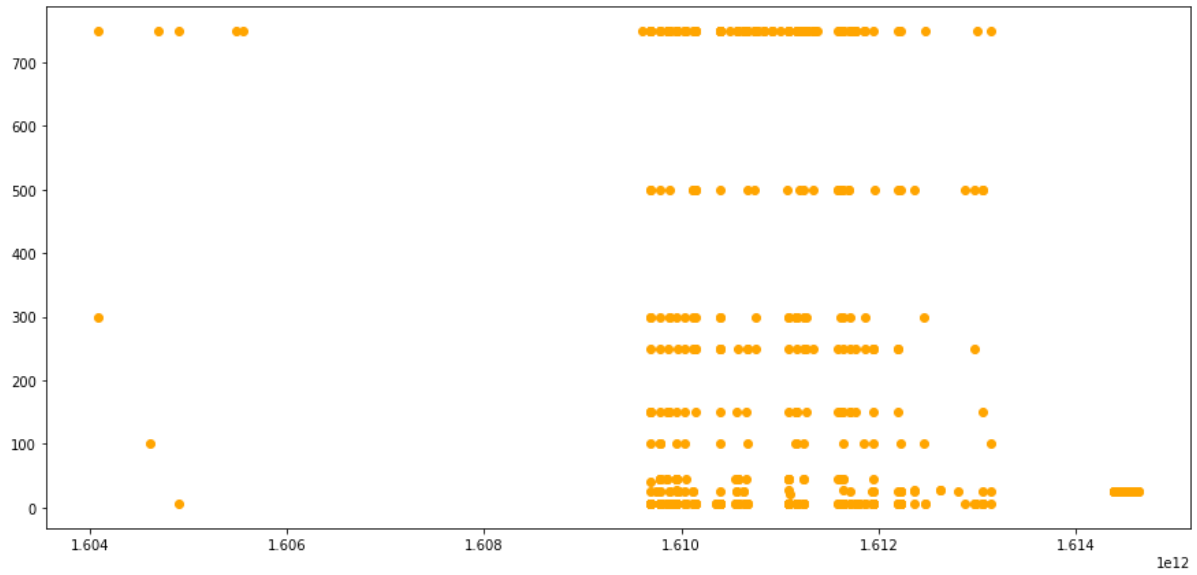
In [25]:
```
# plotting purchased Item Count distribution based on created date
plt.figure(figsize=(15,7))
plt.scatter(df1["createDate"], df1["purchasedItemCount"], Color = 'Orange')
```

Out[25]: <matplotlib.collections.PathCollection at 0x222054c7710>



In [26]:
```
# plotting bonusPointsEarned distribution based on created date
plt.figure(figsize=(15,7))
plt.scatter(df1["createDate"], df1["bonusPointsEarned"], Color = 'Orange')
```

Out[26]:
```
<matplotlib.collections.PathCollection at 0x2220544e550>
```

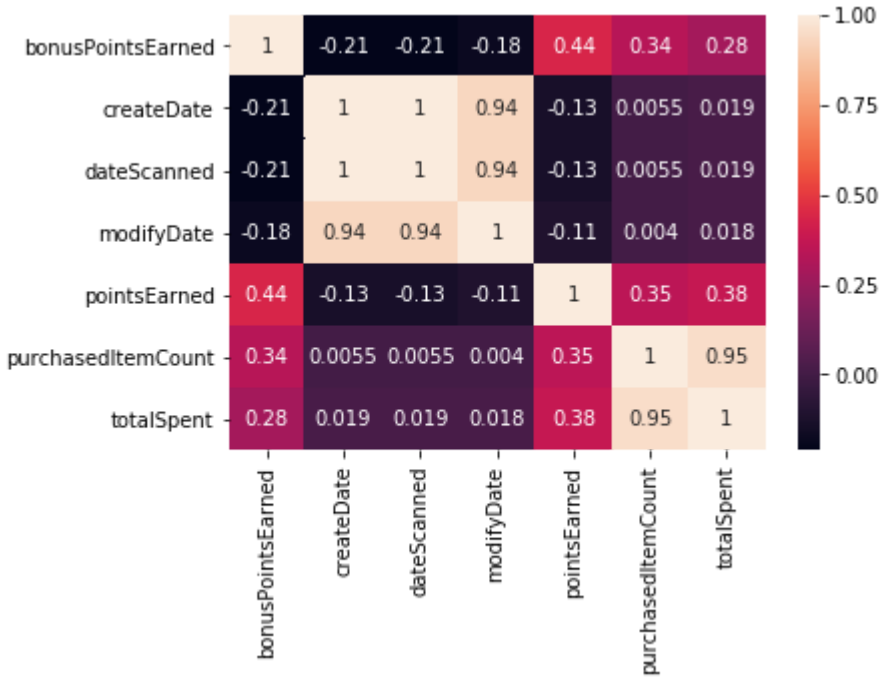


In [27]:
```
df1.corr()
```

Out[27]:

	bonusPointsEarned	createDate	dateScanned	modifyDate	pointsEarned	pu
bonusPointsEarned	1.000000	-0.210862	-0.210862	-0.177035	0.440762	
createDate	-0.210862	1.000000	1.000000	0.937038	-0.133617	
dateScanned	-0.210862	1.000000	1.000000	0.937038	-0.133617	
modifyDate	-0.177035	0.937038	0.937038	1.000000	-0.107876	
pointsEarned	0.440762	-0.133617	-0.133617	-0.107876	1.000000	
purchasedItemCount	0.337626	0.005496	0.005496	0.004049	0.347553	
totalSpent	0.283143	0.018761	0.018761	0.018426	0.375839	

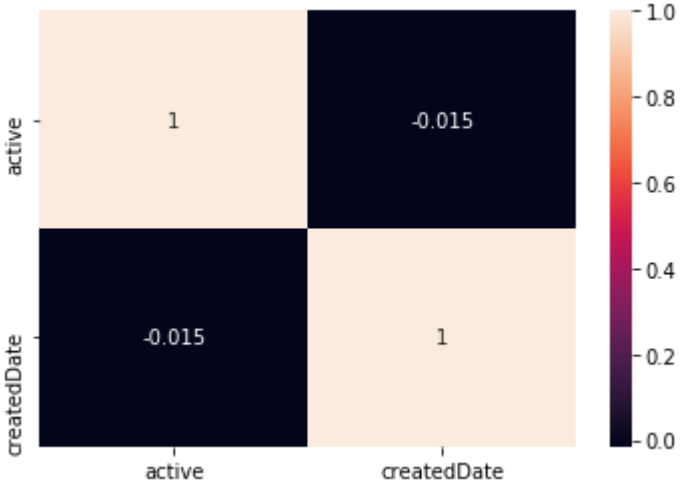
In [28]:
```
#Finding coorelation between variables for receipt data
corrMatrix = df1.corr()
sns.heatmap(corrMatrix, annot=True)
plt.show()
```



Not strong coorelation between data found

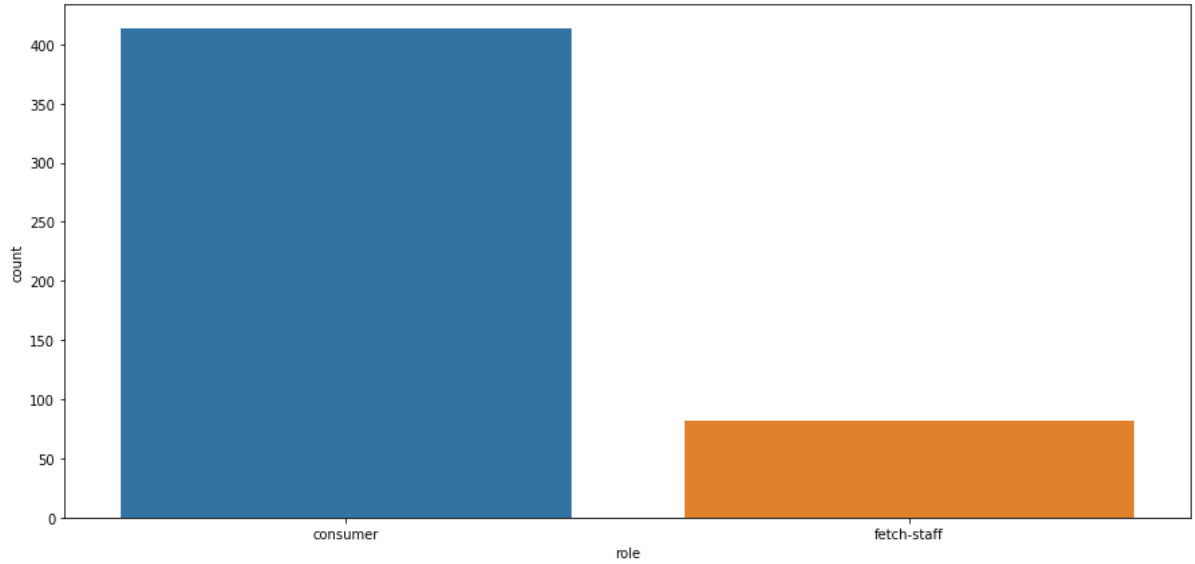
Data exploration for User Data

In [29]:
```
#Finding coorelation between variables for user data
corrMatrix = df3.corr()
sns.heatmap(corrMatrix, annot=True)
plt.show()
```



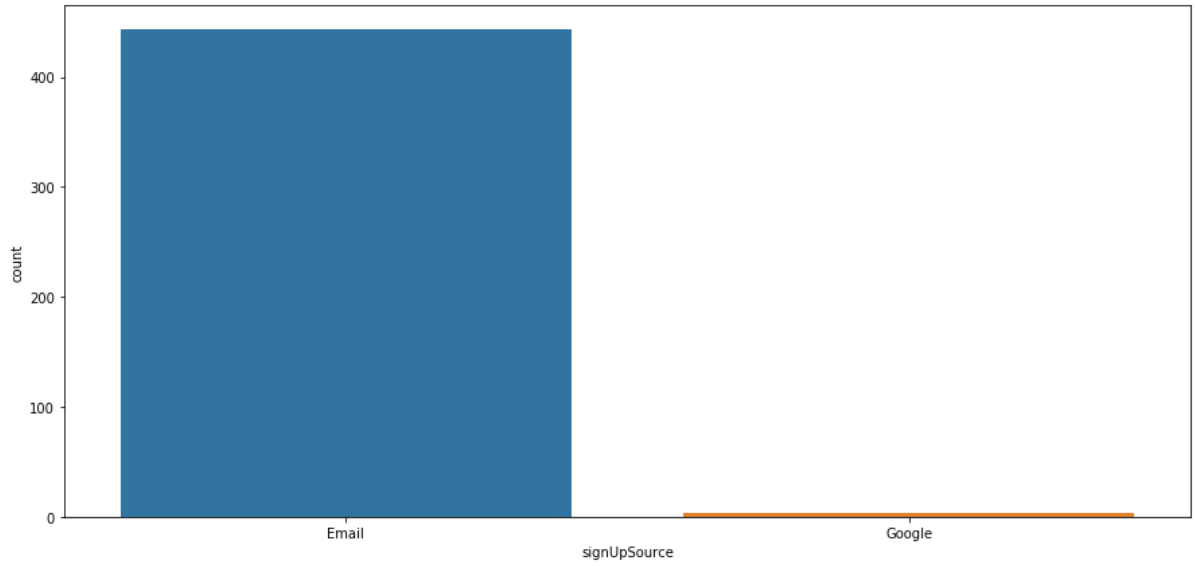
In [30]:
```
# Finding role distribution
plt.figure(figsize=(15,7))
sns.countplot('role', data=df3)
```

Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x222056f32b0>



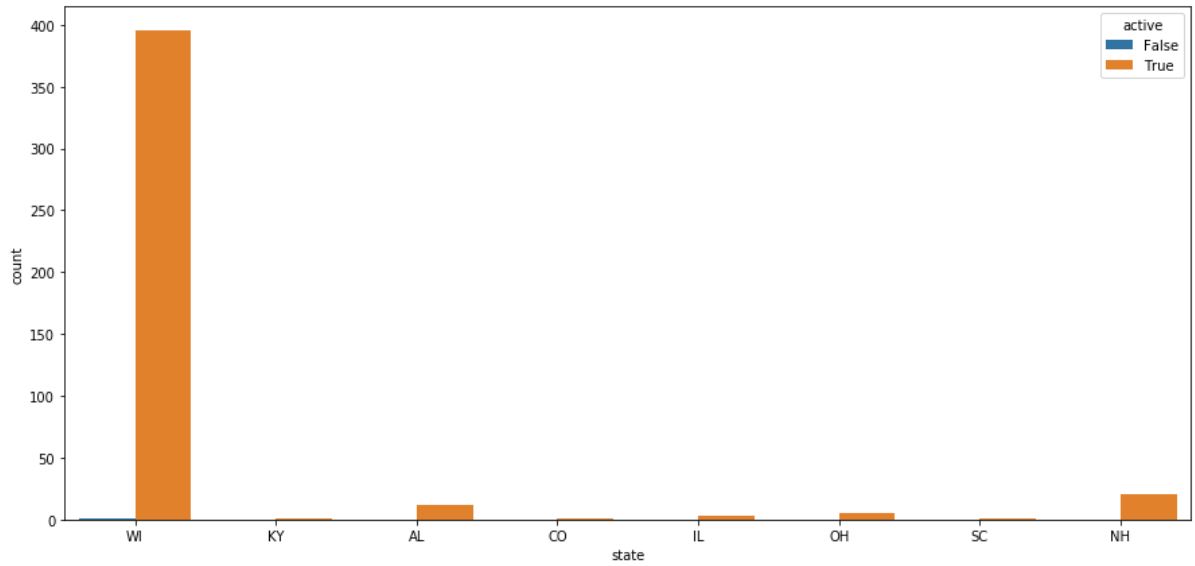
In [31]:
```
# Finding signup source distribution
plt.figure(figsize=(15,7))
sns.countplot('signUpSource', data=df3)
```

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x22205c71eb8>



In [32]:
```
# Finding state distribution wrt active status
plt.figure(figsize=(15,7))
sns.countplot('state', data=df3, hue="active")
```

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x22205ca0630>



In [33]:
```
# Finding Active status distribution
plt.figure(figsize=(15,7))
sns.countplot('active', data=df3)
```

Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x22205c82198>