In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
```

In [2]:

```
train=pd.read_csv("train.csv")
center=pd.read_csv("fulfilmentcenter.csv")
meal=pd.read_csv("meal.csv")
test=pd.read_csv("test.csv")
```

In [3]:

```
train.head(10)
```

Out[3]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	home
0	1379560	1	55	1885	136.83	152.29	0	
1	1466964	1	55	1993	136.83	135.83	0	
2	1346989	1	55	2539	134.86	135.86	0	
3	1338232	1	55	2139	339.50	437.53	0	
4	1448490	1	55	2631	243.50	242.50	0	
5	1270037	1	55	1248	251.23	252.23	0	
6	1191377	1	55	1778	183.36	184.36	0	
7	1499955	1	55	1062	182.36	183.36	0	
8	1025244	1	55	2707	193.06	192.06	0	
9	1054194	1	55	1207	325.92	384.18	0	

In [4]:

```
test.head(10)
```

Out[4]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	home
0	1028232	146	55	1885	158.11	159.11	0	
1	1127204	146	55	1993	160.11	159.11	0	
2	1212707	146	55	2539	157.14	159.14	0	
3	1082698	146	55	2631	162.02	162.02	0	
4	1400926	146	55	1248	163.93	163.93	0	
5	1284113	146	55	1778	190.15	190.15	0	
6	1197966	146	55	1062	191.09	192.09	0	
7	1132739	146	55	2707	242.56	240.56	0	
8	1057981	146	55	1207	360.90	360.90	0	
9	1095932	146	55	1230	383.18	384.18	0	

In [5]:

```
meal.head(10)
```

Out[5]:

	meal_id	category	cuisine
0	1885	Beverages	Thai
1	1993	Beverages	Thai
2	2539	Beverages	Thai
3	1248	Beverages	Indian
4	2631	Beverages	Indian
5	1311	Extras	Thai
6	1062	Beverages	Italian
7	1778	Beverages	Italian
8	1803	Extras	Thai
9	1198	Extras	Thai

In [6]:

```
center.head(10)
```

Out[6]:

	center_id	city_code	region_code	center_type	op_area
0	11	679	56	TYPE_A	3.7
1	13	590	56	TYPE_B	6.7
2	124	590	56	TYPE_C	4.0
3	66	648	34	TYPE_A	4.1
4	94	632	34	TYPE_C	3.6
5	64	553	77	TYPE_A	4.4
6	129	593	77	TYPE_A	3.9
7	139	693	34	TYPE_C	2.8
8	88	526	34	TYPE_A	4.1
9	143	562	77	TYPE_B	3.8

In [7]:

```
train.shape,test.shape,center.shape,meal.shape
```

Out[7]:

```
((456548, 9), (32573, 8), (77, 5), (51, 3))
```

In [8]:

```
train = pd.merge(train,center,
                 how="left",
                 left_on='center_id',
                 right_on='center_id')

train = pd.merge(train,meal,
                 how='left',
                 left_on='meal_id',
                 right_on='meal_id')
```

In [9]:

```
test = pd.merge(test,center,
                how="left",
                left_on='center_id',
                right_on='center_id')

test = pd.merge(test,meal,
                how='left',
                left_on='meal_id',
                right_on='meal_id')
```

In [10]:

```
train.shape,test.shape
```

Out[10]:

```
((456548, 15), (32573, 14))
```

In [11]:

```
train.head(10)
```

Out[11]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	home
0	1379560	1	55	1885	136.83	152.29	0	
1	1466964	1	55	1993	136.83	135.83	0	
2	1346989	1	55	2539	134.86	135.86	0	
3	1338232	1	55	2139	339.50	437.53	0	
4	1448490	1	55	2631	243.50	242.50	0	
5	1270037	1	55	1248	251.23	252.23	0	
6	1191377	1	55	1778	183.36	184.36	0	
7	1499955	1	55	1062	182.36	183.36	0	
8	1025244	1	55	2707	193.06	192.06	0	
9	1054194	1	55	1207	325.92	384.18	0	

In [12]:

```
train.isnull().sum()
```

Out[12]:

```
id                    0
week                  0
center_id             0
meal_id               0
checkout_price        0
base_price            0
emailer_for_promotion 0
homepage_featured     0
num_orders            0
city_code             0
region_code           0
center_type           0
op_area               0
category              0
cuisine               0
dtype: int64
```

In [13]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 456548 entries, 0 to 456547
Data columns (total 15 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   id                     456548 non-null  int64  
 1   week                   456548 non-null  int64  
 2   center_id              456548 non-null  int64  
 3   meal_id                456548 non-null  int64  
 4   checkout_price         456548 non-null  float64
 5   base_price             456548 non-null  float64
 6   emailer_for_promotion  456548 non-null  int64  
 7   homepage_featured      456548 non-null  int64  
 8   num_orders             456548 non-null  int64  
 9   city_code              456548 non-null  int64  
 10  region_code            456548 non-null  int64  
 11  center_type            456548 non-null  object 
 12  op_area                456548 non-null  float64
 13  category               456548 non-null  object 
 14  cuisine                456548 non-null  object 
dtypes: float64(3), int64(9), object(3)
memory usage: 55.7+ MB
```

In [14]:

```
s=pd.get_dummies(train["center_type"])
train=pd.concat([train,s],axis=1)
s1=pd.get_dummies(test["center_type"])
test=pd.concat([test,s1],axis=1)
```

In [15]:

```
t=pd.get_dummies(train["category"])
u=pd.get_dummies(train["cuisine"])
train=pd.concat([train,t,u],axis=1)
t1=pd.get_dummies(test["category"])
u1=pd.get_dummies(test["cuisine"])
test=pd.concat([test,t1,u1],axis=1)
```

In [16]:

```
train.drop(['center_type','category','cuisine'],axis=1, inplace=True)
test.drop(['center_type','category','cuisine'],axis=1, inplace=True)
```

In [17]:

```
train.head(10)
```

Out[17]:

	id	week	center_id	meal_id	checkout_price	base_price	emailer_for_promotion	home
0	1379560	1	55	1885	136.83	152.29	0	
1	1466964	1	55	1993	136.83	135.83	0	
2	1346989	1	55	2539	134.86	135.86	0	
3	1338232	1	55	2139	339.50	437.53	0	
4	1448490	1	55	2631	243.50	242.50	0	
5	1270037	1	55	1248	251.23	252.23	0	
6	1191377	1	55	1778	183.36	184.36	0	
7	1499955	1	55	1062	182.36	183.36	0	
8	1025244	1	55	2707	193.06	192.06	0	
9	1054194	1	55	1207	325.92	384.18	0	

10 rows × 33 columns

In [18]:

```
train[train.columns[1:]].corr()['num_orders'][:-1]
```

Out[18]:

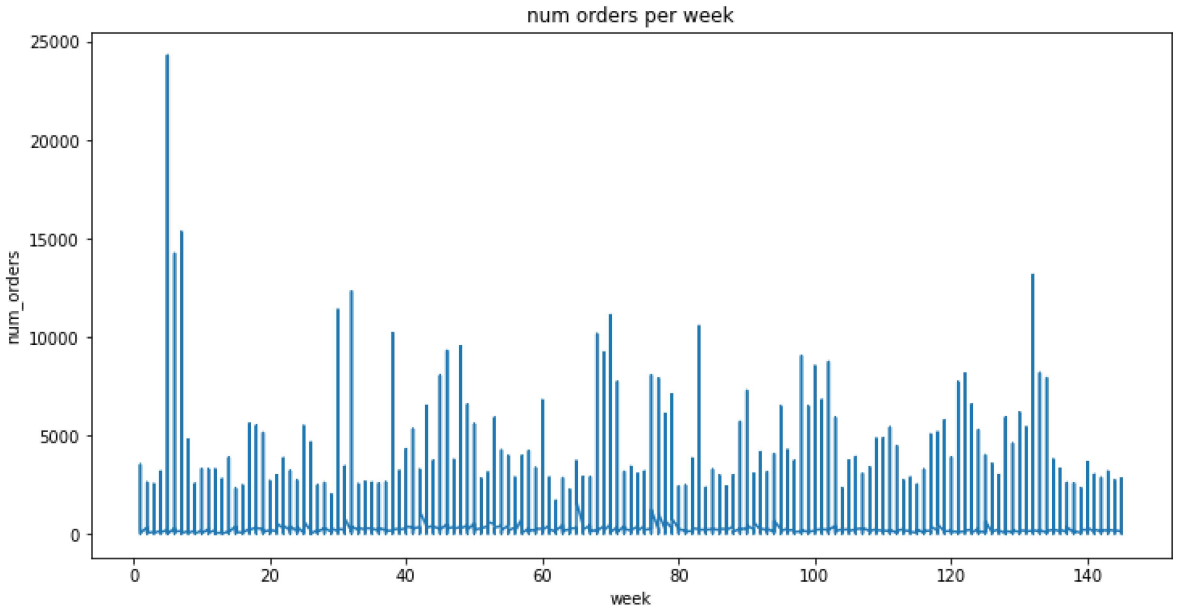
```
week                   -0.017210
center_id              -0.053035
meal_id                 0.010597
checkout_price         -0.282108
base_price             -0.222306
emailer_for_promotion   0.277147
homepage_featured       0.294490
num_orders              1.000000
city_code               0.041596
region_code             0.029744
op_area                 0.176976
TYPE_A                  0.001535
TYPE_B                  0.073322
TYPE_C                 -0.073647
Beverages               0.086110
Biryani                -0.126996
Desert                 -0.129376
Extras                  0.014125
Fish                   -0.067262
Other Snacks           -0.065998
Pasta                  -0.130124
Pizza                  -0.027597
Rice Bowl               0.257584
Salad                   0.079172
Sandwich                0.189771
Seafood                -0.101768
Soup                   -0.076762
Starters               -0.071327
Continental            -0.132514
Indian                 -0.047453
Italian                 0.149443
Name: num_orders, dtype: float64
```

In [19]:

```
train.drop(['id'],axis=1, inplace=True)
test.drop(['id'],axis=1, inplace=True)
```

In [20]:

```
from matplotlib import pyplot as plt
plt.figure(figsize=(12,6))
plt.title('num orders per week')
plt.plot(train.week,train.num_orders)
plt.xlabel('week')
plt.ylabel('num_orders')
plt.show()
```



In [21]:

```
train['month'] = train['week']/4
train['week_from_yr_start'] = train['week']/52
train['quarter'] = train['week']/13
test['month'] = test['week']/4
test['week_from_yr_start'] = test['week']/52
test['quarter'] = test['week']/13
```

In [22]:

```
train.drop(['week'],axis=1, inplace=True)
test.drop(['week'],axis=1, inplace=True)
```

In [23]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 456548 entries, 0 to 456547
Data columns (total 34 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   center_id              456548 non-null  int64  
 1   meal_id                456548 non-null  int64  
 2   checkout_price         456548 non-null  float64
 3   base_price             456548 non-null  float64
 4   emailer_for_promotion  456548 non-null  int64  
 5   homepage_featured      456548 non-null  int64  
 6   num_orders             456548 non-null  int64  
 7   city_code              456548 non-null  int64  
 8   region_code            456548 non-null  int64  
 9   op_area                456548 non-null  float64
 10  TYPE_A                 456548 non-null  uint8  
 11  TYPE_B                 456548 non-null  uint8  
 12  TYPE_C                 456548 non-null  uint8  
 13  Beverages              456548 non-null  uint8  
 14  Biryani                456548 non-null  uint8  
 15  Desert                 456548 non-null  uint8  
 16  Extras                 456548 non-null  uint8  
 17  Fish                   456548 non-null  uint8  
 18  Other Snacks           456548 non-null  uint8  
 19  Pasta                  456548 non-null  uint8  
 20  Pizza                  456548 non-null  uint8  
 21  Rice Bowl              456548 non-null  uint8  
 22  Salad                  456548 non-null  uint8  
 23  Sandwich               456548 non-null  uint8  
 24  Seafood                456548 non-null  uint8  
 25  Soup                   456548 non-null  uint8  
 26  Starters               456548 non-null  uint8  
 27  Continental            456548 non-null  uint8  
 28  Indian                 456548 non-null  uint8  
 29  Italian                456548 non-null  uint8  
 30  Thai                   456548 non-null  uint8  
 31  month                  456548 non-null  float64
 32  week_from_yr_start     456548 non-null  float64
 33  quarter                456548 non-null  float64
dtypes: float64(6), int64(7), uint8(21)
memory usage: 57.9 MB
```

In [24]:

```
train.nunique()
```

Out[24]:

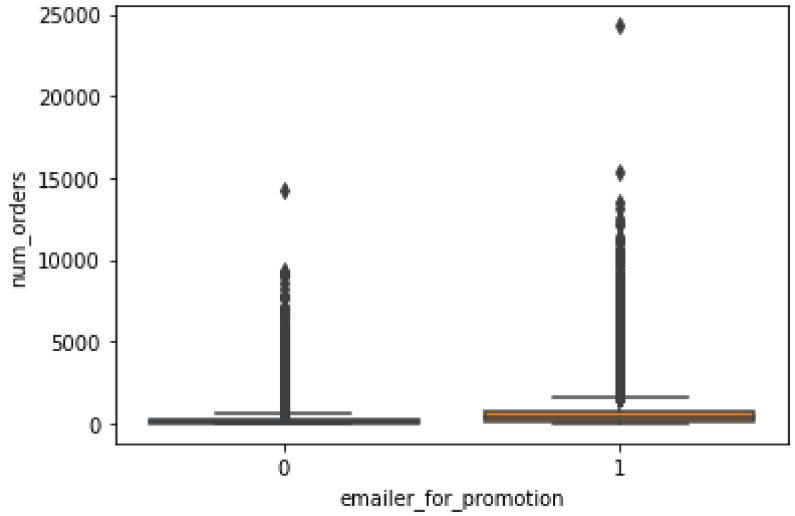
```
center_id                  77
meal_id                    51
checkout_price           1992
base_price               1907
emailer_for_promotion       2
homepage_featured           2
num_orders               1250
city_code                  51
region_code                 8
op_area                    30
TYPE_A                      2
TYPE_B                      2
TYPE_C                      2
Beverages                   2
Biryani                     2
Desert                      2
Extras                      2
Fish                        2
Other Snacks                2
Pasta                       2
Pizza                       2
Rice Bowl                   2
Salad                       2
Sandwich                    2
Seafood                     2
Soup                        2
Starters                    2
Continental                 2
Indian                      2
Italian                     2
Thai                        2
month                     145
week_from_yr_start        145
quarter                   145
dtype: int64
```

In [25]:

```
sns.boxplot(x="emailer_for_promotion",y="num_orders",data=train)
```

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4a2d57760>
```

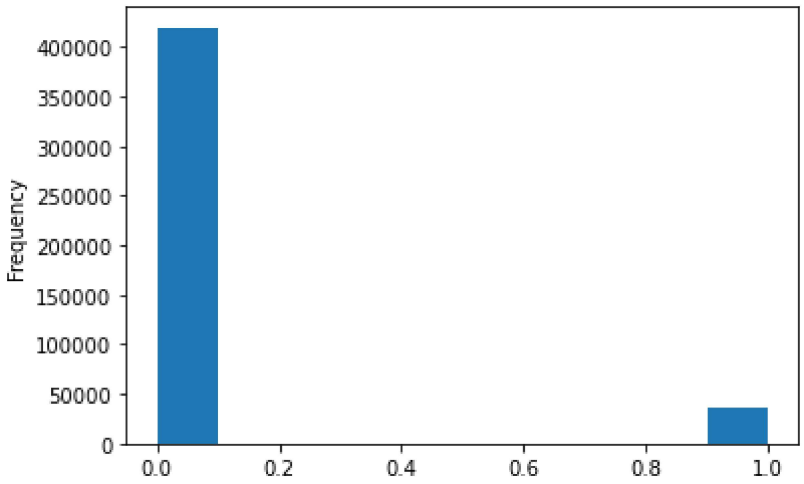


In [26]:

```
train["emailer_for_promotion"].plot.hist()
```

Out[26]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e4a301cc40>
```



In [27]:

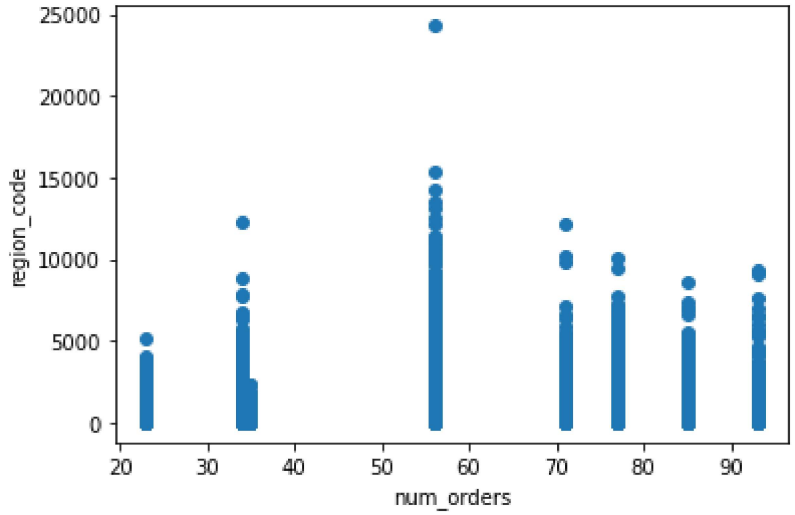
```
train[train.columns[1:]].corr()['num_orders'][:-1]
```

Out[27]:

```
meal_id                 0.010597
checkout_price         -0.282108
base_price             -0.222306
emailer_for_promotion   0.277147
homepage_featured       0.294490
num_orders              1.000000
city_code               0.041596
region_code             0.029744
op_area                 0.176976
TYPE_A                  0.001535
TYPE_B                  0.073322
TYPE_C                 -0.073647
Beverages               0.086110
Biryani                -0.126996
Desert                 -0.129376
Extras                  0.014125
Fish                   -0.067262
Other Snacks           -0.065998
Pasta                  -0.130124
Pizza                  -0.027597
Rice Bowl               0.257584
Salad                   0.079172
Sandwich                0.189771
Seafood                -0.101768
Soup                   -0.076762
Starters               -0.071327
Continental            -0.132514
Indian                 -0.047453
Italian                 0.149443
Thai                    0.021724
month                  -0.017210
week_from_yr_start     -0.017210
Name: num_orders, dtype: float64
```

In [28]:

```
plt.scatter(train.region_code,train.num_orders)
plt.xlabel('num_orders')
plt.ylabel('region_code')
plt.show()
```



In [29]:

```
train.drop(['region_code'],axis=1, inplace=True)
test.drop(['region_code'],axis=1, inplace=True)
```

In [30]:

```
x=train.drop("num_orders",axis=1)
y=train['num_orders']
```

In [31]:

```
train.drop(['num_orders'],axis=1, inplace=True)
```

In [32]:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn import linear_model,datasets
from sklearn.metrics import accuracy_score,mean_squared_error,r2_score
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.20, random_state =
m = RandomForestRegressor(n_jobs=-1,n_estimators=50)
m.fit(x_train,y_train)
pri = m.predict(x_test)
pred=m.predict(test)
print("Random forest = ",r2_score(y_test,pri))
print("Random forest score = ",m.score(x_train,y_train)*100)
```

```
Random forest =  0.8574569027319973
Random forest score =  97.9586120044547
```

In [33]:

```
from math import sqrt
```

In [34]:

```
from sklearn.metrics import mean_squared_log_error
```

In [35]:

```
sqrt(mean_squared_log_error(y_test,pri))
```

Out[35]:

```
0.48841582101853653
```

In [38]:

```
df1=pd.read_csv("test.csv")
dict={'id':df1['id'],'num_orders':pred.reshape(-1,1)[:,-1]}
df2=pd.DataFrame(dict)
df2.to_csv('C:/test/test3.csv',index=False)
```

In [ ]:

```

```