In [32]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# import datetime
pd.set_option('display.max_columns', 100) # set this option to limit the output
pd.set_option('display.max_rows',30)
```

Part 1

let's have an overall look at data

In [2]:
```
data = pd.read_csv('film-permits.csv',delimiter=',')
print(data.shape)
data.head()
```

(52350, 14)

Out[2]:

	EventID	EventType	StartDateTime	EndDateTime	EnteredOn	EventAgency	ParkingHeld	Boroug
0	43547	Shooting Permit	2012-01-10T07:00:00	2012-01-10T19:00:00	2012-01-04T12:25:37	Mayor's Office of Film, Theatre & Broadcasting	EAGLE STREET between FRANKLIN STREET and WEST ...	Brookly
1	43997	Theater Load in and Load Outs	2012-01-19T07:00:00	2012-02-20T22:00:00	2012-01-09T18:22:29	Mayor's Office of Film, Theatre & Broadcasting	WEST 46 STREET between BROADWAY and 8 AVENUE	Manhata
2	43675	Shooting Permit	2012-01-09T07:00:00	2012-01-09T20:00:00	2012-01-05T13:03:51	Mayor's Office of Film, Theatre & Broadcasting	ALLEN STREET between EAST HOUSTON STREET and R...	Manhata
3	44536	Shooting Permit	2012-01-23T07:00:00	2012-01-23T21:00:00	2012-01-18T12:08:17	Mayor's Office of Film, Theatre & Broadcasting	WEST 64 STREET between BROADWAY and CENTRAL ...	Manhata
4	44061	Shooting Permit	2012-01-18T06:00:00	2012-01-18T21:00:00	2012-01-10T14:57:29	Mayor's Office of Film, Theatre & Broadcasting	INGRAHAM STREET between STEWART AVENUE and GAR...	Brookly

In [3]:
```
data.isnull().sum(axis=0)
```

Out[3]:
```
EventID              0
EventType            0
```

```
StartDateTime         0
EndDateTime           0
EnteredOn             0
EventAgency           0
ParkingHeld           0
Borough               0
CommunityBoard(s)    12
PolicePrecinct(s)    12
Category              0
SubCategoryName       0
Country               0
ZipCode(s)           12
dtype: int64
```

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 52350 entries, 0 to 52349
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   EventID            52350 non-null  int64 
 1   EventType          52350 non-null  object
 2   StartDateTime      52350 non-null  object
 3   EndDateTime        52350 non-null  object
 4   EnteredOn          52350 non-null  object
 5   EventAgency        52350 non-null  object
 6   ParkingHeld        52350 non-null  object
 7   Borough            52350 non-null  object
 8   CommunityBoard(s)  52338 non-null  object
 9   PolicePrecinct(s)  52338 non-null  object
 10  Category           52350 non-null  object
 11  SubCategoryName    52350 non-null  object
 12  Country            52350 non-null  object
 13  ZipCode(s)         52338 non-null  object
dtypes: int64(1), object(13)
memory usage: 5.6+ MB
```

Dataset of filming activity consists of columns which report detail about permitted filming activities. we can interpret this dataset from three-point of view:

1, filming activities is from various Countries which need permission for shooting. Each permission contains just one type of activity: Theatre Load In, Load Out, Shooting, Rigging, and DCAS Permit for filming. After the permission is granted on a specific date(EnteredOn), the activity is due to start on StartDateTime and complete on EndDateTime in a particular duration. Each permission contains the location of the activity (Borough)and precincts. There are various category for filming activity such as Commercial, Documentary, Music Video etc.

2, From the dataset, we can tell the most popular season in a year or the unpopular days of weak for filming. Furthermore, we can recognize the most popular category and location from data. We can extract some information like the number of permission in each category group by country or borough, which gives a good prospect about the interest of each country in each category. The duration of each activity may vary based on the kind of activity (EventType) or category of a film. Also, the delay of starting (LeadingTime) may vary according to activity or category.

3, The type of each column determine the kind of data we should interpret. Except for EventID, which is numerical, other columns have the categorical type and should be interpreted by the count of its column or numerical column that comes

from date-time columns. A column like EventAgency does not imply any information, and we can remove it without any consequences. Also, the number of nulls in each column is essential and should be replaced or the row deleted.

Part 2

In [5]:
```
# EventID does not have a usfull information for interpretation
data = data.drop('EventID',1)
```

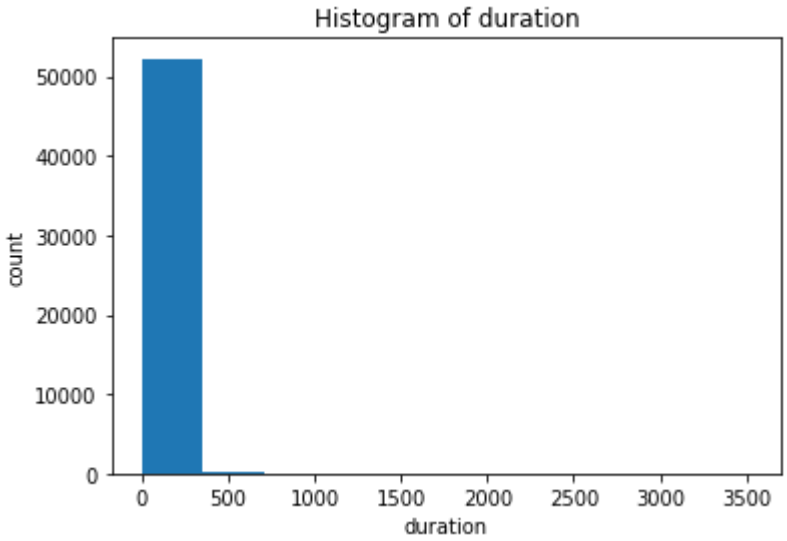
In [6]:
```
# calculate the duration in hour of each permitted filming activity by subtracti
start_date = pd.to_datetime(data['StartDateTime'])
end_date = pd.to_datetime(data['EndDateTime'])
duration = end_date-start_date
data['duration_activity_hour'] = duration.apply(lambda a : a.days*24+ a.seconds/
```

In [7]:
```
# we can have a better inference by adding median, skew and kurt to discribe rep
describe_ = data['duration_activity_hour'].describe()
describe_.reindex(['count', 'mean', 'std', 'min', '25%', '50%', '75%', 'max','me
describe_['median']=data['duration_activity_hour'].median()
describe_['skew']=data['duration_activity_hour'].skew()
describe_['kurt']=data['duration_activity_hour'].kurt()
describe_
```

Out[7]:
```
count     52350.000000
mean         19.576409
std          43.021020
min           0.000000
25%          13.000000
50%          15.000000
75%          16.000000
max        3528.000000
median       15.000000
skew         27.731170
kurt       1398.272685
Name: duration_activity_hour, dtype: float64
```

the median for the duration of filming activity is about 19.5 hours while the max is 3528 hours which shows a high variance (std = 43)

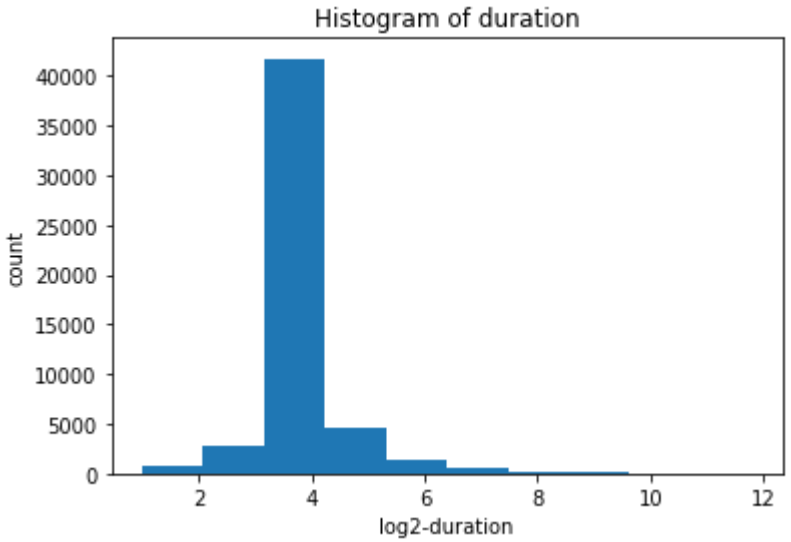
In [8]:
```
# histogram for duration
plt.hist(data['duration_activity_hour'])
plt.xlabel('duration');
plt.ylabel("count");
plt.title("Histogram of duration");
```



The duration of activity does not have a normal distribution (right-skewed) and shows most of the activities have a duration between 0 to ~300 hour

In [9]:
```
# we can solve the skewness by applying transformer of log2
data2= data.copy()
data2['duration_activity_hour'] = np.log2(data2['duration_activity_hour'])
data2= data2[data2['duration_activity_hour']>0]
plt.hist(data2['duration_activity_hour']);
plt.xlabel('log2-duration');
plt.ylabel("count");
plt.title("Histogram of duration");
```

```
/home/mahmood/project/myenv/lib/python3.8/site-packages/pandas/core/series.py:726: RuntimeWarning: divide by zero encountered in log2
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



In [10]:
```
# geting mean and median and std for each category
# grouped_cat = data.groupby('Category').median().drop('EventID',1)
grouped_cat = data.groupby('Category')['duration_activity_hour']
grouped_cat.agg([np.mean,np.median ,np.std])
```

Out[10]:

	mean	median	std

Category	mean	median	std
Category			
Commercial	14.970394	15	8.840373
Documentary	15.421801	12	15.809500
Film	15.267306	14	12.232862
Music Video	15.271084	14	9.386257
Red Carpet/Premiere	17.000000	17	NaN
Still Photography	13.928658	13	8.811145
Student	10.721408	9	11.002745
Television	17.410755	15	34.560539
Theater	50.077157	29	106.396753
WEB	15.853833	14	13.665191

The theatre has the highest median duration for filming and student lowest duration.

In [11]:
```
# geting mean and median and std for each category
grouped_cat = data.groupby('Country')['duration_activity_hour']
grouped_cat.agg([np.mean,np.median ,np.std])
```

Out[11]:

	mean	median	std
Country			
Australia	14.600000	16.0	2.607681
Canada	14.000000	14.0	5.873670
France	18.571429	12.0	17.840564
Germany	26.000000	26.0	NaN
Ireland	12.000000	12.0	1.414214
Japan	9.000000	9.5	2.070197
Netherlands	13.000000	13.0	3.000000
Panama	13.285714	14.0	0.951190
United Kingdom	12.062500	13.0	4.312289
United States of America	19.583282	15.0	43.043690

Germany with 26 hours has the highest median of filming activity among countries, and Japan has the lowest.

In [12]:
```
# geting mean and median and std for each Borough
grouped_boro = data.groupby('Borough')['duration_activity_hour']
grouped_boro.agg([np.mean,np.median ,np.std])
```

Out[12]:

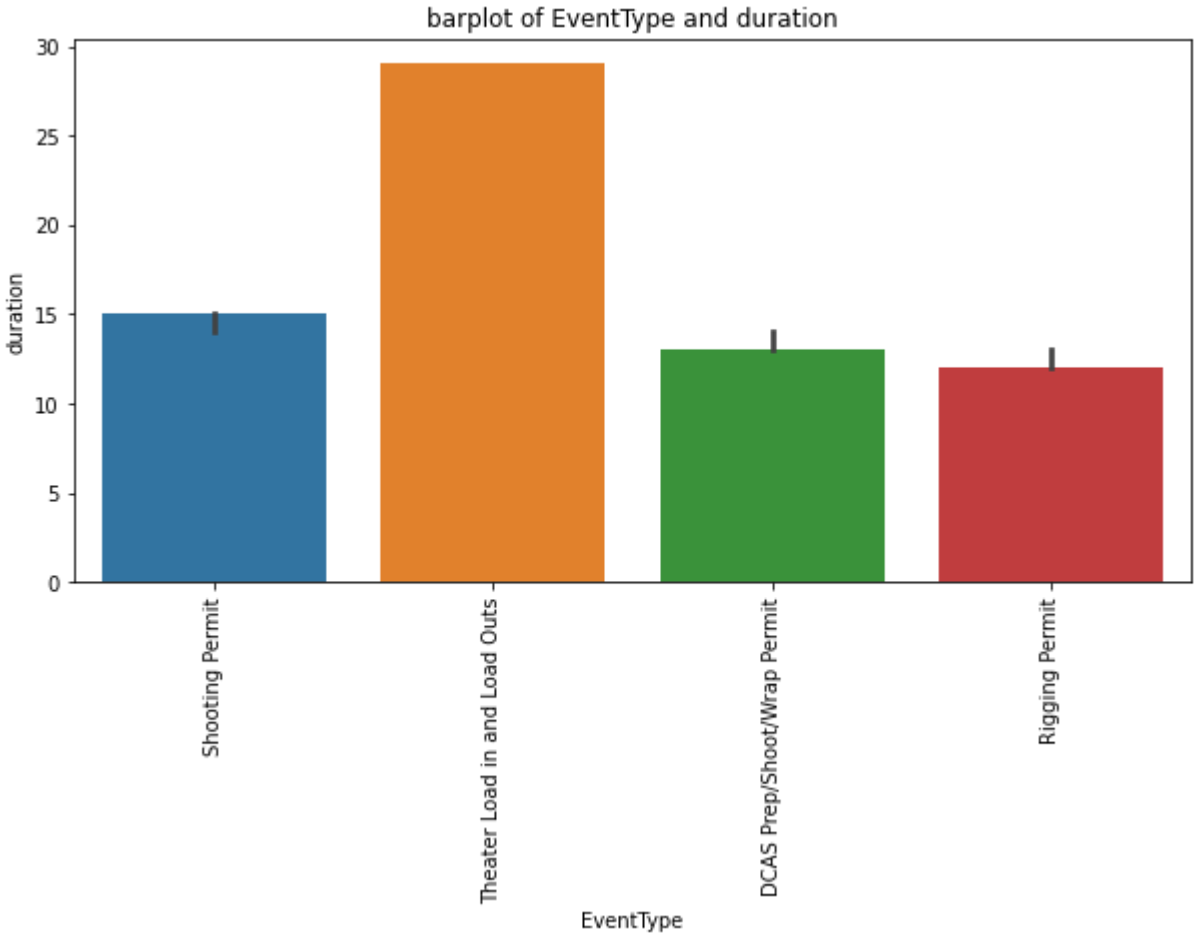
	mean	median	std

Borough	mean	median	std

Borough			
Bronx	16.030550	15	16.198909
Brooklyn	16.142947	14	18.178803
Manhatan	23.104362	15	58.246605
Queens	15.888203	15	13.947296
Staten Island	15.236364	15	8.536908

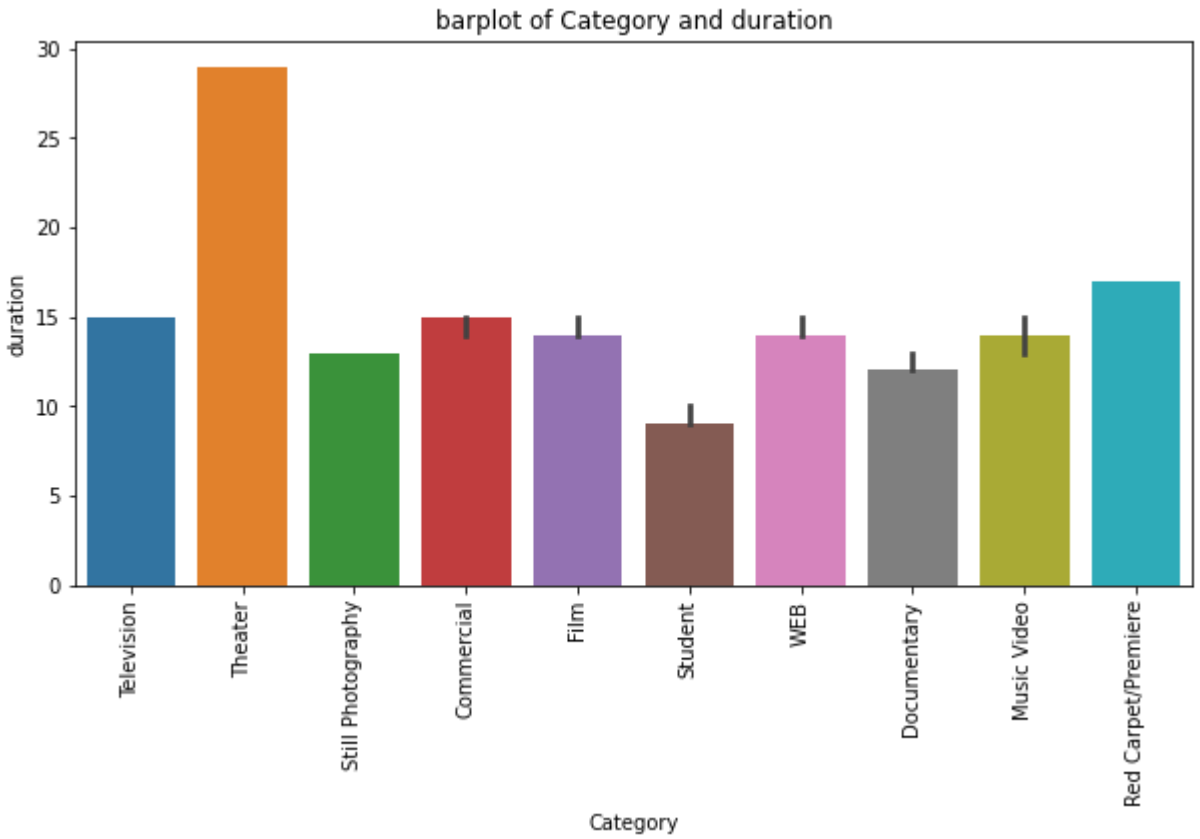
The duration of filming activity in different Borough is roughly the same

In [13]:
```
# bar plot of EventType and duration
plt.figure(figsize=(10,5))
sns.barplot(x='EventType',y='duration_activity_hour', data=data, estimator=np.me
plt.xticks(rotation=90);
plt.xlabel('EventType');
plt.ylabel("duration");
plt.title("barplot of EventType and duration");
```

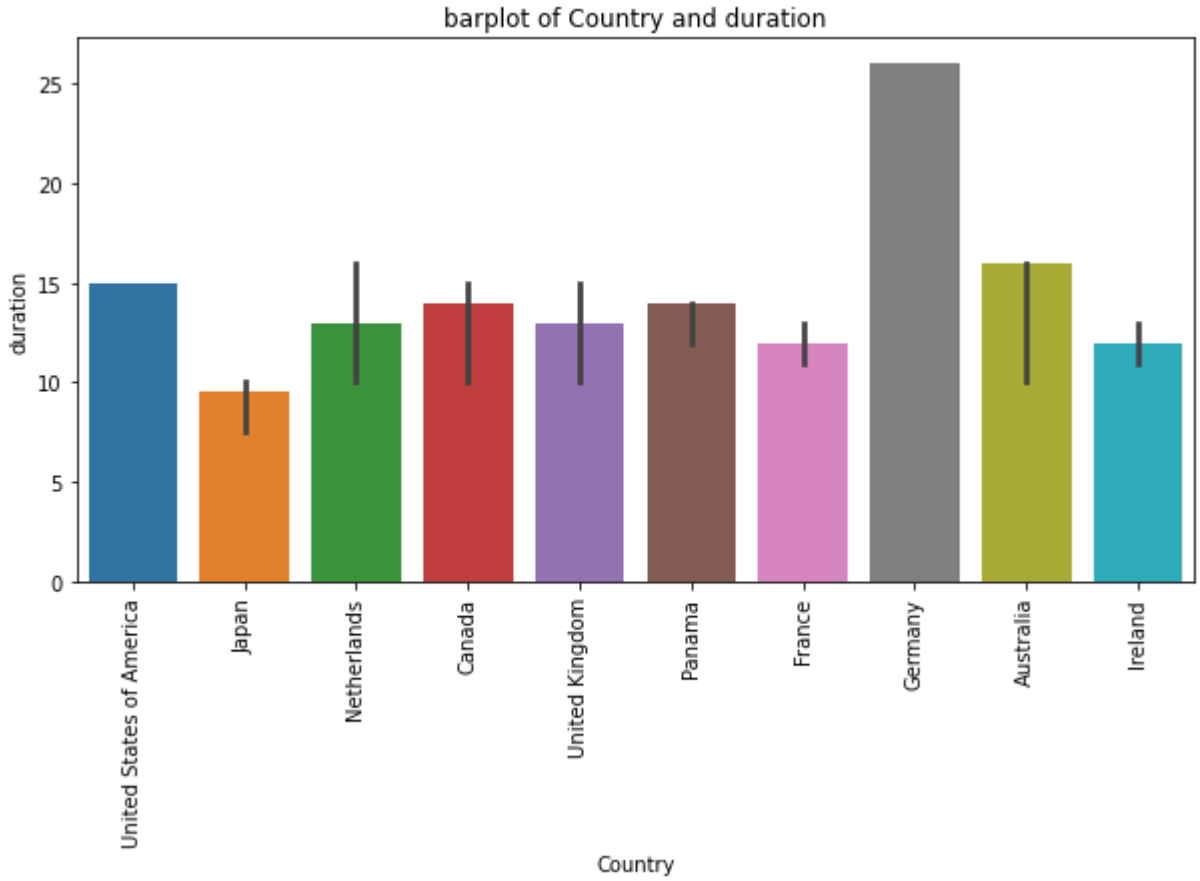


In [14]:
```
# bar plot of Category and duration
plt.figure(figsize=(10,5))
sns.barplot(x='Category',y='duration_activity_hour', data=data, estimator=np.mec
plt.xticks(rotation=90);
```

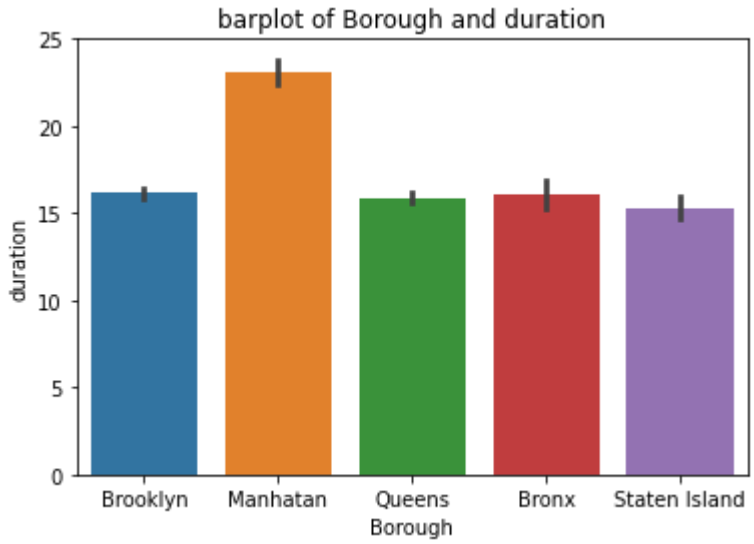
```
plt.xlabel('Category');
plt.ylabel("duration");
plt.title("barplot of Category and duration");
```



In [15]:
```
# bar plot of Country and duration
plt.figure(figsize=(10,5))
sns.barplot(x='Country',y='duration_activity_hour', data=data, estimator=np.medi
plt.xticks(rotation=90);
plt.xlabel('Country');
plt.ylabel("duration");
plt.title("barplot of Country and duration");
```



In [16]:
```
sns.barplot(x='Borough',y='duration_activity_hour', data=data);
plt.xlabel('Borough');
plt.ylabel("duration");
plt.title("barplot of Borough and duration");
```



Part 3

In [17]:
```
# get the count of permited activity for each category seperately
data['Category'].value_counts()
```

Out[17]:
```
Television              28136
Film                     9072
```

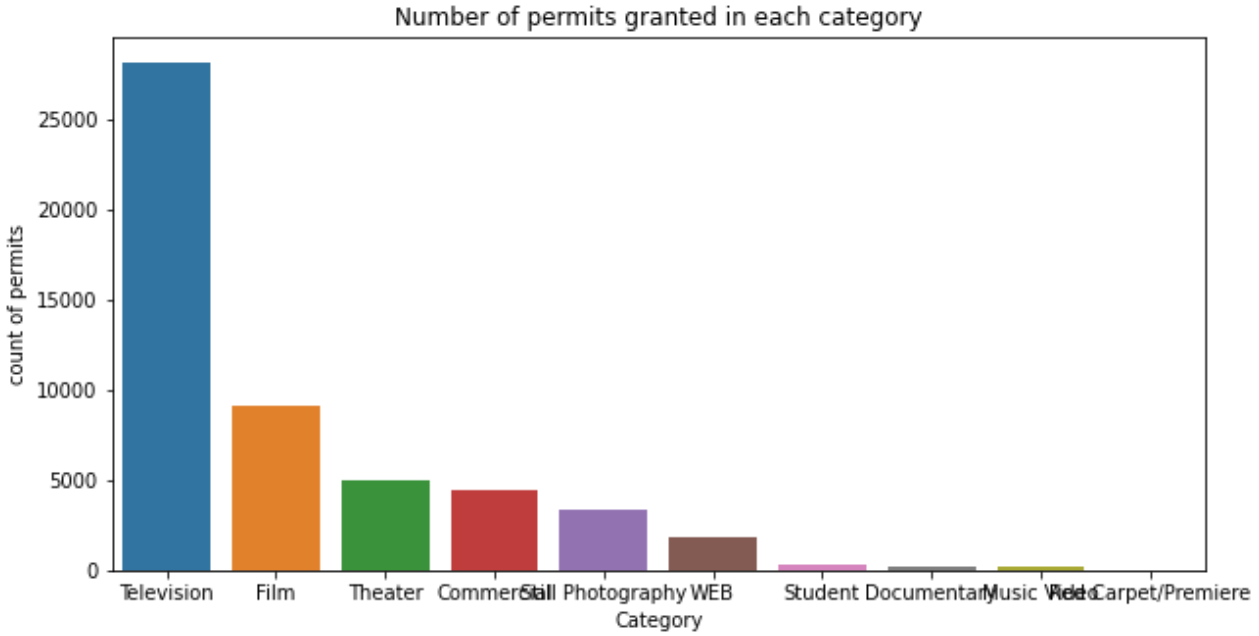
```
Theater                 4925
Commercial              4391
Still Photography       3294
WEB                     1813
Student                  341
Documentary              211
Music Video              166
Red Carpet/Premiere        1
Name: Category, dtype: int64
```

In [18]:
```
# get the count of permited activity for each country seperately
data['Country'].value_counts()
```

Out[18]:
```
United States of America    52292
United Kingdom                 16
Canada                          9
Japan                           8
Panama                          7
France                          7
Australia                       5
Netherlands                     3
Ireland                         2
Germany                         1
Name: Country, dtype: int64
```

As you can see, majority of permited activity is for USA

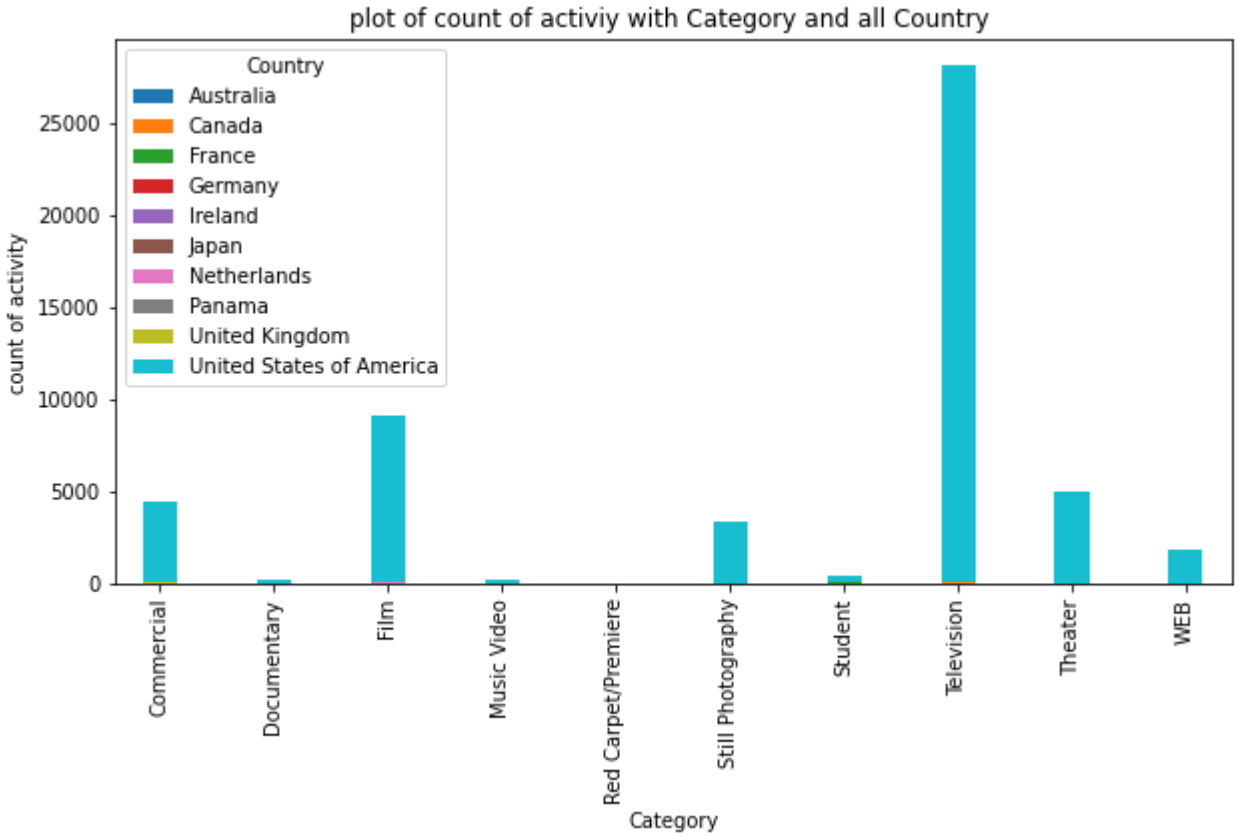
In [19]:
```
# visualize the count of permited activity for each category by barplot
plt.figure(figsize=(10,5))
sns.countplot(x='Category',data=data,order=data['Category'].value_counts().index
plt.title("Number of permits granted in each category")
plt.xlabel("Category")
plt.ylabel("count of permits");
```



In [20]:
```
# visualising the number of permited activity by two categorical variables (Cate
plt.figure(figsize=(15,10))
df_plot = data.groupby(['Category', 'Country']).size()
df_plot=df_plot.reset_index().pivot(columns='Country', index='Category', values=
df_plot.plot(kind='bar',width=0.3, stacked=True,figsize=(10,5))
plt.title("plot of count of activiy with Category and all Country ")
```

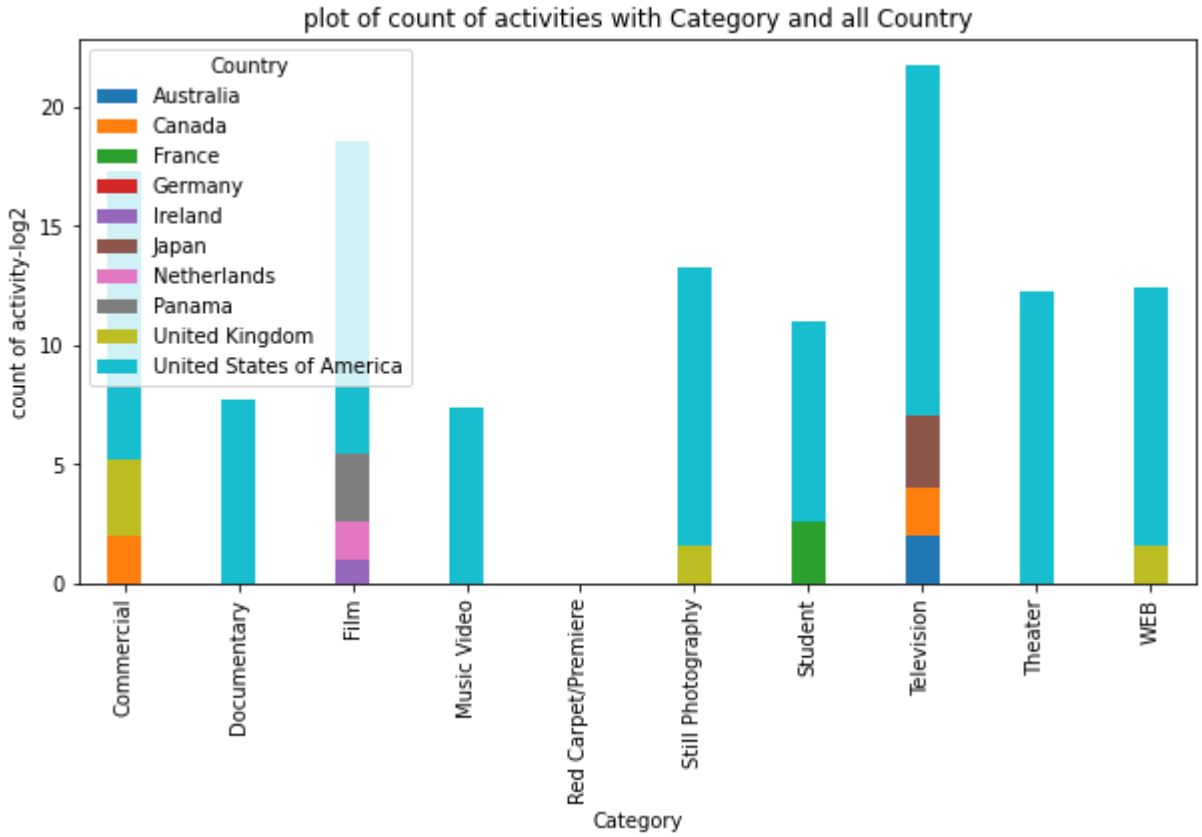
```
plt.xlabel("Category")
plt.ylabel("count of activity");
```

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



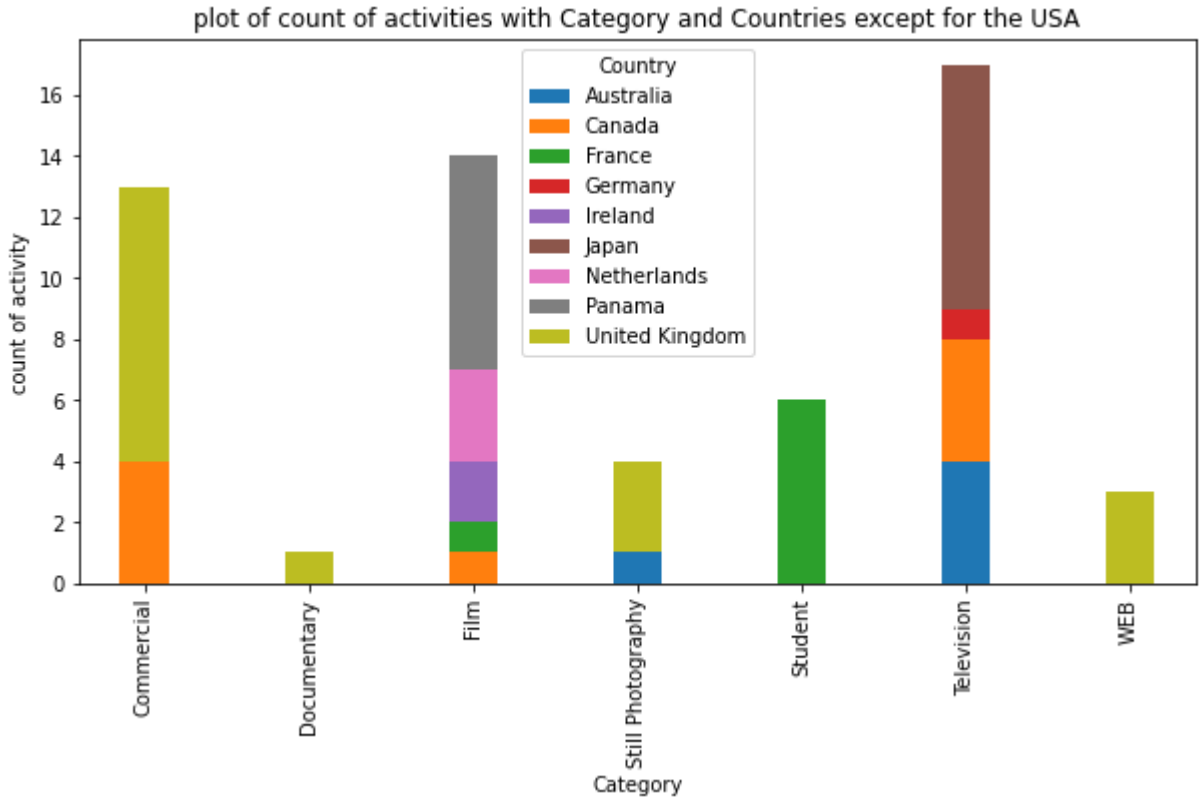
because most of the activity belongs to the USA, this plot can not demonstrate the difference in a good way

In [21]:
```
# approach 1: we apply log2 (or other transformer) to better depict the bar plot
df_plot = np.log2(data.groupby(['Category', 'Country']).size())
df_plot.unstack().plot(kind='bar', width=0.3,stacked=True,figsize=(10,5));
plt.title("plot of count of activities with Category and all Country")
plt.xlabel("Category")
plt.ylabel("count of activity-log2");
```



we should notice that in this barplot the difference is exponential

In [22]:
```
# approach 2:  number of filming activity in all country except for USA
usa= data[data.Country!='United States of America']
# aa = usa.groupby(['Country','Category']).size()
aa = usa.groupby(['Category', 'Country']).size()
aa.unstack().plot(kind='bar', width=0.3,stacked=True,figsize=(10,5));
plt.title("plot of count of activities with Category and Countries except for th
plt.xlabel("Category")
plt.ylabel("count of activity");
```



Variance bar plots in this part depict that the most permitted activity mostly belongs to the USA in nearly all categories. Television is the most popular category for filmmakers, and Film and theatre are second and third, respectively. On the other hand, among countries except for the USA, this pattern does not continue. The United Kingdom requests permission for filming mostly for commercial programs and Panama for film category.

Part 4

In [23]:
```
# transform multivalue column of PolicePrecinct to single value by replicating r
ploic_cat = data.loc[:,('PolicePrecinct(s)','Category')]
ploic_cat['PolicePrecinct(s)'] = ploic_cat['PolicePrecinct(s)'].str.split(',') #
ploic_cat = ploic_cat.explode('PolicePrecinct(s)').reset_index(drop=True)
ploic_cat
```

Out[23]:

	PolicePrecinct(s)	Category
0	108	Television
1	94	Television
2	18	Theater
3	5	Still Photography
4	7	Still Photography
...	...	...
69396	19	Television
69397	20	Television

	PolicePrecinct(s)	Category
69398	24	Television
69399	114	Film
69400	9	Film

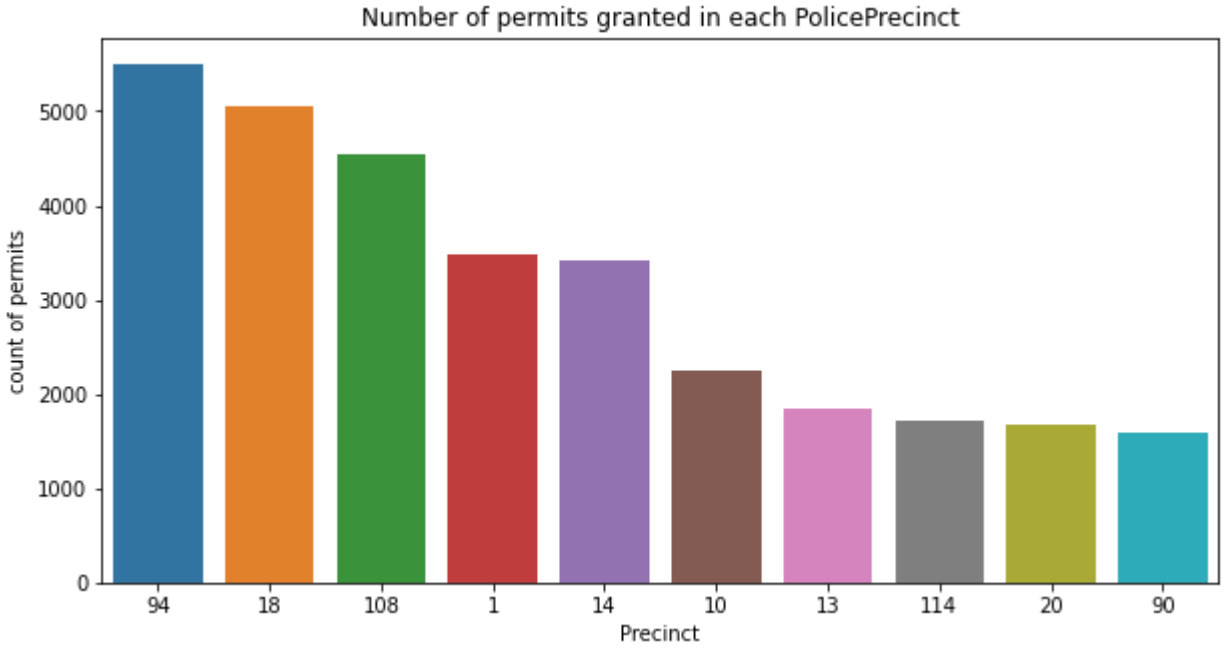
69401 rows × 2 columns

In [24]:
```
# count of PolicePrecinct in each category
ploic_cat.value_counts()
```

Out[24]:
```
PolicePrecinct(s)  Category
94                 Television     4433
108                Television     3921
18                 Television     2588
                   Theater        1494
1                  Television     1377
                                  ...
 9                 Student           1
67                 Theater           1
                   Student           1
81                 Documentary       1
 1                 WEB               1
Length: 967, dtype: int64
```

As you can see, popular precincts for television activity are 94, 108 and 18, respectively.

In [25]:
```
# visualizing count of PolicePrecinct, 10 first Precinct which are more popular
plt.figure(figsize=(10,5))
sns.countplot(x='PolicePrecinct(s)',data=ploic_cat,order=ploic_cat['PolicePrecir
plt.title("Number of permits granted in each PolicePrecinct")
plt.xlabel("Precinct")
plt.ylabel("count of permits");
```



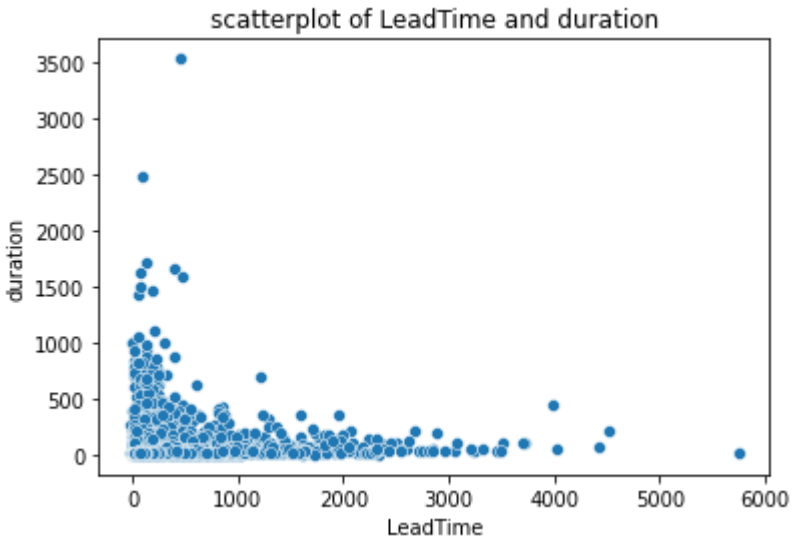
According to the plot, the most popular Precinct among all filming

activities is 94, and 18 and 108 have second and third rates, respectively.

Part 5

In [26]:
```
# calculate LeadTime
start_date = pd.to_datetime(data['StartDateTime'])
submit_date = pd.to_datetime(data['EnteredOn'])
duration = start_date - submit_date
data['LeadTime'] = duration.apply(lambda a : a.days*24+ a.seconds//3600) # conve
```

In [27]:
```
#  scatterplot for LeadTime and duration
sns.scatterplot(data= data, x = 'LeadTime', y='duration_activity_hour')
plt.title("scatterplot of LeadTime and duration")
plt.xlabel("LeadTime")
plt.ylabel("duration");
```

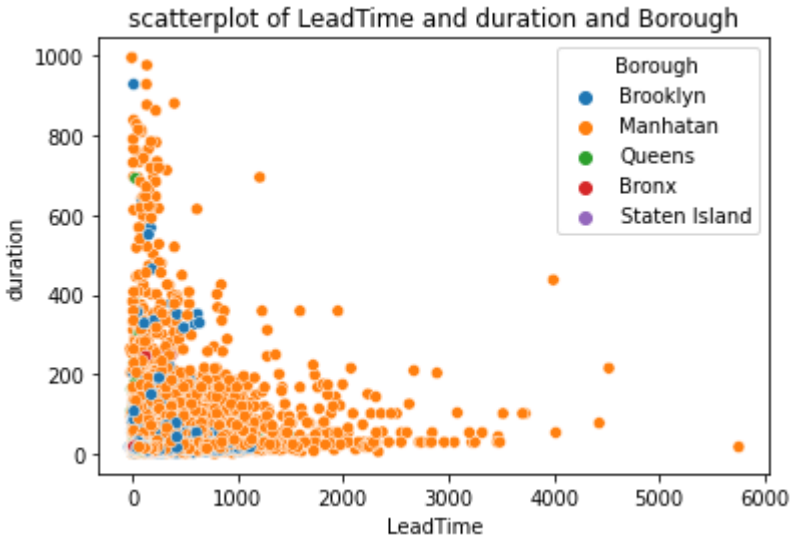


In [28]:
```
# correlation between LeadTime and duration
data.corr()[['LeadTime','duration_activity_hour']]
```

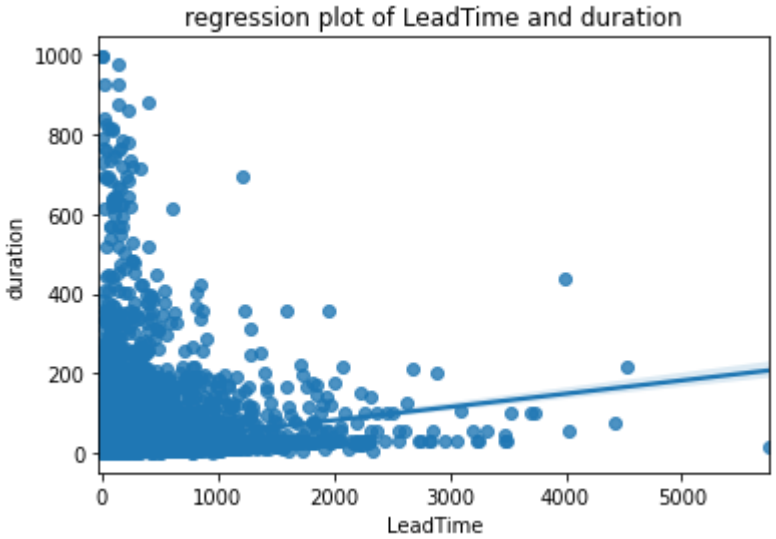
Out[28]:

	LeadTime	duration_activity_hour
duration_activity_hour	0.142604	1.000000
LeadTime	1.000000	0.142604

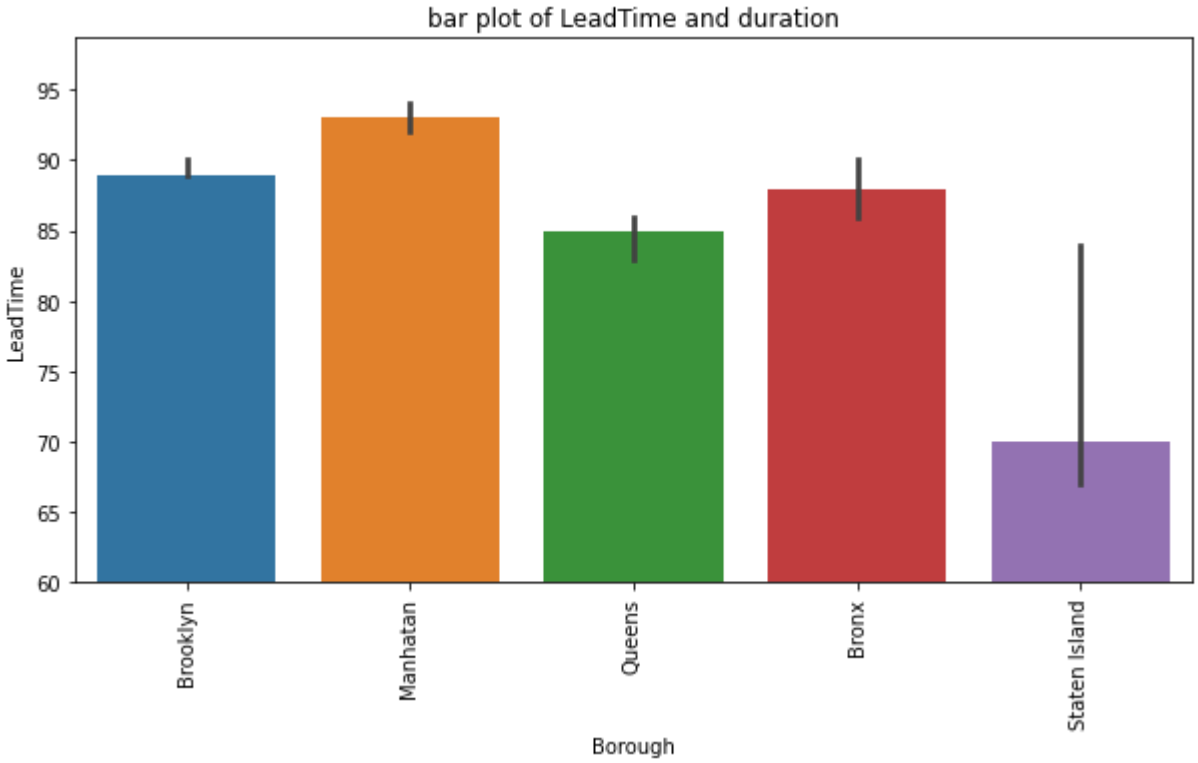
In [29]:
```
# scatterplot for LeadTime and duration and Borough
data1 = data[data['duration_activity_hour']<1000]
sns.scatterplot(data= data1, x = 'LeadTime', y='duration_activity_hour',hue='Bor
plt.title("scatterplot of LeadTime and duration and Borough")
plt.xlabel("LeadTime")
plt.ylabel("duration");
```



In [30]:
```
#  regression plot
sns.regplot(data= data1, x='LeadTime', y='duration_activity_hour')
plt.title("regression plot of LeadTime and duration")
plt.xlabel("LeadTime")
plt.ylabel("duration");
```



In [31]:
```
plt.figure(figsize=(10,5))
sns.barplot(x='Borough',y='LeadTime', data=data, estimator=np.median);
plt.ylim(ymin=60)
plt.xticks(rotation=90);
plt.title("bar plot of LeadTime and duration");
```



there is a weak correlation between LeadTime duration which is a requirement for regression

conclusion

There is 52350 filming permission which majority of them (28136) is in the Television category with a median duration of about 15 hours. Theatre has the most prolonged median duration, about 28 hours but is in third place for a count of permitted activities with around 5000 filming activities. USA number of filming activities is the pioneer in approximately all categories while in duration activity they take third place and Germany is first. It is noteworthy that precincts of 94, 18 and 108 are the most popular for filming. Furthermore, filming in Manhattan takes the most extended Leadtime while Staten Island is the shortest.