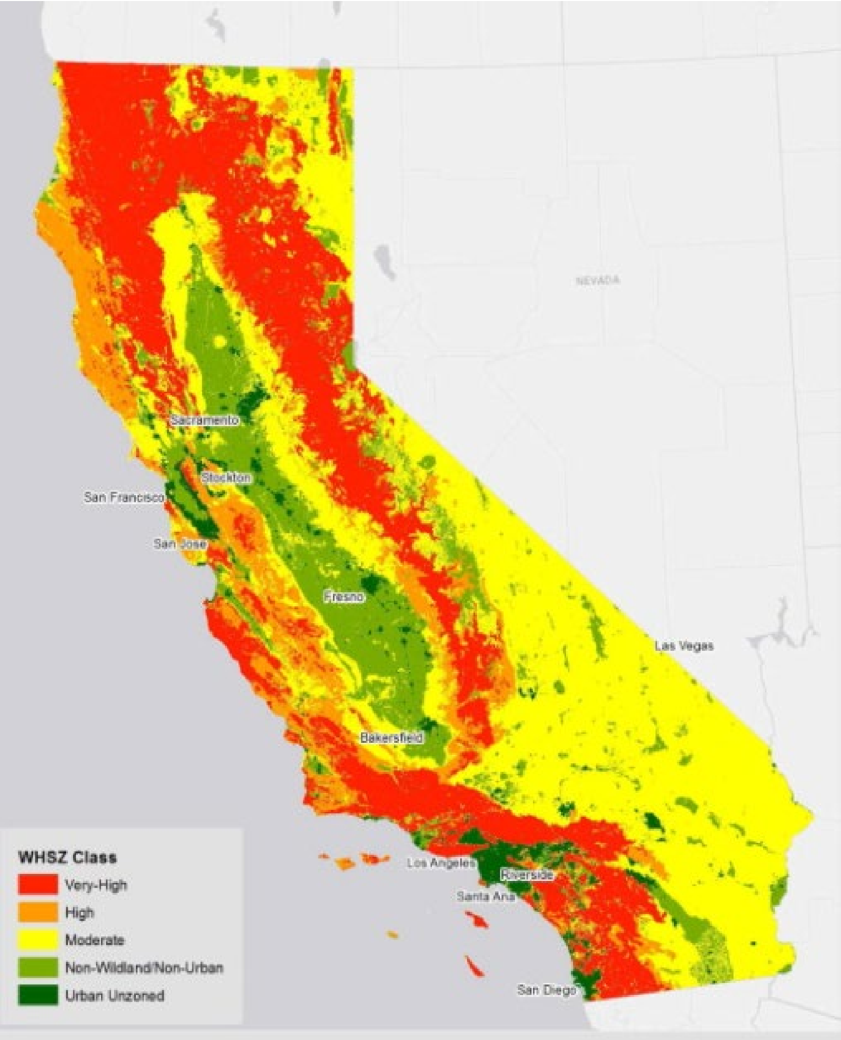


California Fire Incidents Analysis

Agenda:

- Dataset Description + Cleaning
- Directions:

  1)Analysis of Wildfire Duration Time on the effects of Resources (AirTankers, PersonnelInvolved, Dozers and so on) to see how effective Resources help control wildfire duration time

  2)Analysis of Wildfire Severity by Region, County, Admin Unit to see what area deserve more resources

  3)Analysis of Wildfires in Santa Clara County

- Findings
- Conclusions

Data Set Description

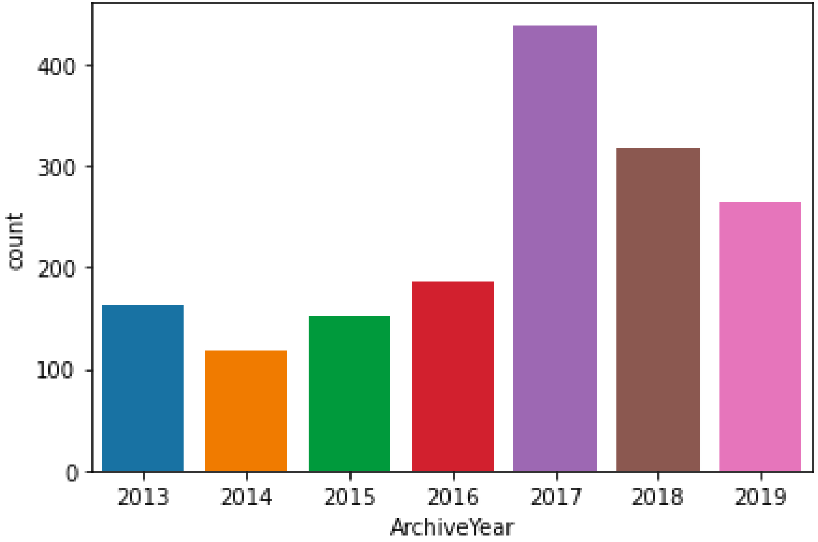
df is a data set for Wildfires that have occurred in California between 2013 and 2019.

Important columns in df are:

1. **AcresBurned:** Acres of land affected by wildfires
2. **MajorIncident :** Whether the incident is considered as “Major Incident”. True means Major Incident and False Not a Major Incident. Major wildfire incidents are large, extended-day wildfires(10 acres or greater) according to CAL Fire department. (https://www.fire.ca.gov/incidents/)
3. **AdminUnit:** Fire Department Name took care of the Incident
4. **AirTankers :** The number of Resources Air Tankers assigned
5. **CrewsInvolved:** The number of Resources Crews assigned
6. **Dozers:** The number of Resources Dozers assigned
7. **Engines:** The number of Resources Engines assigned
8. **Helicopters:** The number of Helicopters assigned
9. **Counties:** County name
10. **Extinguished :** Incident Extinguished time
11. **Latitude:** Incident’s Latitude
12. **Longitude:** Incident’s Longitude
13. **Started:** Incident Started time
14. ….

What is the trend of fire activity over the years?

```
In [98]: sns.countplot(x='ArchiveYear',data=df)
Out[98]: <AxesSubplot:xlabel='ArchiveYear', ylabel='count'>
```



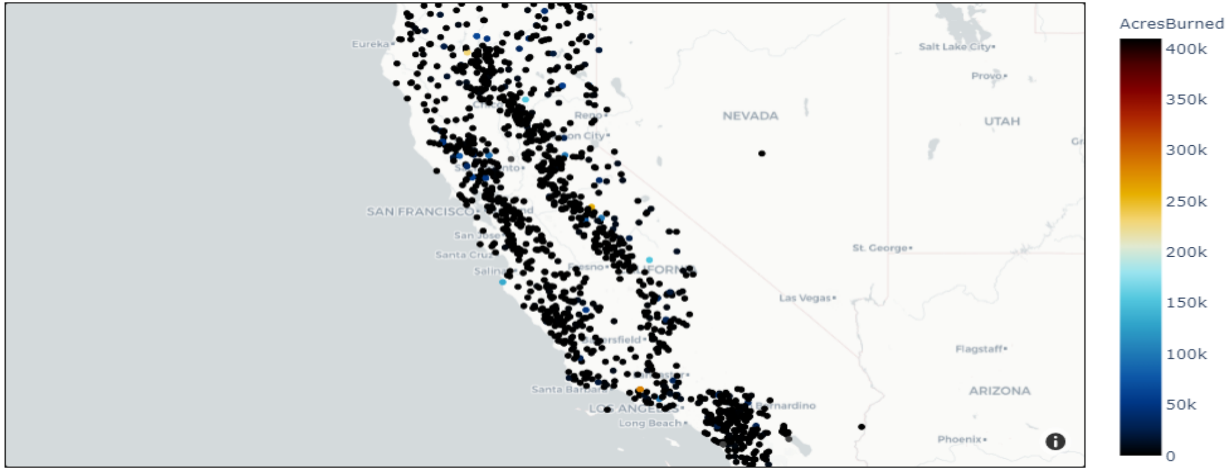
Most Fire incidents in the Year 2017, with over 400 fires.

Since then, fire incidents have decreased.

Overall California Fire Activity by Acres Burned

```
df1 = df[['Latitude','Longitude','AcresBurned']]
```

```
px.set_mapbox_access_token ='pk.eyJ1IjoiYWxhbnR6dGFuIiwiYSI6ImNqa2VqeHFnejB1dTEzcHBxczBhaGZhZ2gifQ.moVxmxTkE53tZDfBFr3uYA'
fig = px.scatter_mapbox(data_frame=df1,
                        lat="Latitude",
                        lon="Longitude",
                        color="AcresBurned",
                        color_continuous_scale=px.colors.cyclical.IceFire,
                        #center = {"lat": 37.35, "lon": -121.96}
                       )
fig.update_layout(mapbox_style="carto-positron",\
                  mapbox_zoom=6, mapbox_center = {"lat": 37.35, "lon": -121.96})
```



Data Cleaning

Column Name	Method
AcresBurned	Drop Rows with Nan
AirTankers	Replace Nan with zero
CrewsInvolved	Replace Nan with zero
Dozers	Replace Nan with zero
Engines	Replace Nan with zero
Fatalities	Replace Nan with zero
Helicopters	Replace Nan with zero
Injuries	Replace Nan with zero
PersonnelInvolved	Replace Nan with zero
StructuresDamaged	Replace Nan with zero
StructuresDestroyed	Replace Nan with zero
StructuresEvacuated	Replace Nan with zero
StructuresThreatened	Replace Nan with zero
WaterTenders	Replace Nan with zero
ConditionStatement	Replace Nan with 'Unknown'
ControlStatement	Replace Nan with 'Unknown'
SearchDescription	Replace Nan with 'Unknown'
SearchKeywords	Replace Nan with 'Unknown'
FuelType	Drop Column
Extinguished	Replace Nan with 'Unknown'

More data cleaning…

1. *Create new Column "Duration_days" by subtracting Extinguished time by Started*

```
In [530]: #Step one: Calculate the 'Duration hours' for each wildfire incident:
          df['Duration_days'] = (pd.to_datetime(df.Extinguished)-pd.to_datetime(df.Started)).astype('timedelta64[h]')/24
```

1. *Column "AdminUnit": Change all names from uppercase to lower case, then remove common terms and "/" "-" and other signs from the list.*

```
In [542]: df['AdminUnit_updated'] = df.AdminUnit.apply(lambda x: x.lower().rstrip('!?,.').replace("/","").replace("-"," ")\
                                        .replace("cal","").replace("california","").replace("fire","")\
                                         .replace("department","").replace("county","").replace("national","")\
                                         .replace("forest","").replace("unit","").replace("usfs","").replace("us","")\
                                        .replace("city","").replace("(","").replace(")",""))

In [543]: df['AdminUnit_updated'] = df.AdminUnit_updated.apply(lambda x: x.lstrip().rstrip().replace("  "," "))

In [544]: df.AdminUnit_updated.nunique()

Out[544]: 292
```

1. *Column "Duration_days": Remove error inputs; convert negative values to the absolute value*

```
In [555]: #To convert negative days to the absolute values
          df['Duration_days'] = np.abs(df.Duration_days)
```

Q1： How Effective Resources Help Control Wildfire Duration Time

Machine Learning - Decision Tree: Date preparation

1), Dropping the duplicated columns and unnecessary columns

```
In [573]: # Dropping the duplicated columns and unnecessary columns
df_dt.drop(columns=['Active', 'AdminUnit','ArchiveYear', 'CalFireIncident', 'CanonicalUrl','ConditionStatement',\
                    'ControlStatement', 'Counties', 'CountyIds', 'Extinguished', 'Featured', 'Final','Location','Name', \
                    'PercentContained','Public','SearchDescription', 'SearchKeywords','Started', 'Status', 'UniqueId', \
                    'Updated','AdminUnit', 'Duration_days', 'Latitude', 'Longitude', 'Fatalities', \
                    'Injuries', 'StructuresDamaged', 'StructuresDestroyed', 'StructuresEvacuated', \
                    'StructuresThreatened', 'AcresBurned'], inplace=True)
```

2), Create dummy variable

```
In [570]: # Create dummy variable: 1 means the incident is a major incident, 0 means it's not.
Replace MajorIncident with 1 (True), and 0 (False)
df_dt['MajorIncident_1']=df_dt['MajorIncident'].apply(lambda x: 1.0 if x==True else 0.0)
```

3), Make Duration Time Above Average binary

```
In [572]: # Make Duration binary
df_dt['Duration_AboveAvg'] = df.Duration_days.apply(lambda x: 1.0 if x>df.Duration_days.mean() else 0)
```

4), Overview of the cleaned dataset

```
In [574]: df_dt.head()
Out[574]:
```

	AirTankers	CrewsInvolved	Dozers	Engines	Helicopters	PersonnelInvolved	WaterTenders	MajorIncident	Duration_AboveAvg
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5), Create Y and X for Decision Tree Classifier

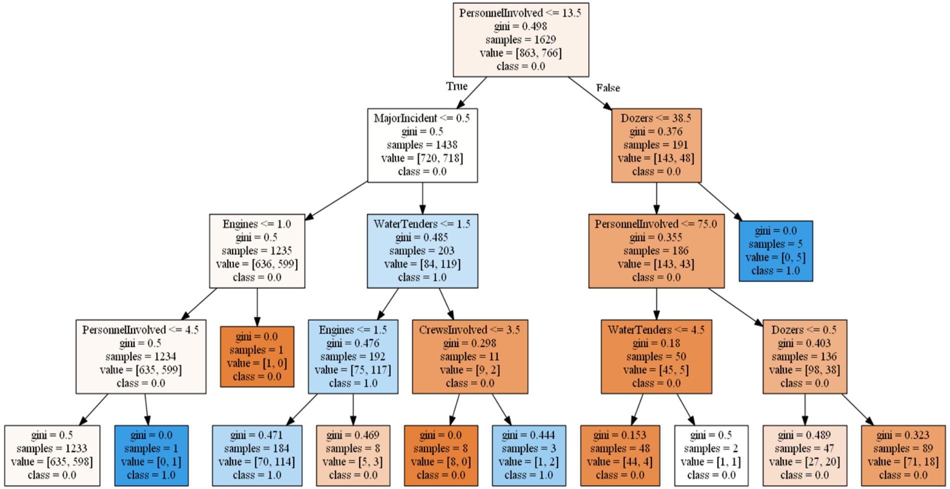
Train the decision tree

```
In [576]: # Creat Y and X for Decision Tree Classifier
Y = df_dt.Duration_AboveAvg
X = df_dt.drop(columns='Duration_AboveAvg')

In [577]: dt = tree.DecisionTreeClassifier(max_depth=4)
dt.fit(X,Y)

Out[577]: DecisionTreeClassifier(max_depth=4)
```

Q1 : How Effective Resources Help Control Wildfire Duration Time



Machine Learning - Classification

Decision Tree: Explanation

1. *If the number of Personnel Involved is greater than 13.5 and the Dozers involved is greater than 38.5, the probability that the wildfire incident lasts more than 84 days is very high.*

1. *If the number of Personnel Involved is not greater than 13.5, and it's a Major incident, when the WaterTnders smaller or equal to 1.5, Engines smaller or equal to 1.5, and PersonnalInvolved not greater than 9, the probability that the wildfire incident lasts more than 84 days is high too.*

1. *This Analysis shows that Personnel Involved, Whether it's a Major incident or not and other resources are important factors in controlling wirefire duration time below its average 84 days.*

Q1： How Effective Resources Help Control Wildfire Duration Time

Machine Learning - Classification: Random Forest

In [583]:
```
from sklearn.model_selection import train_test_split
```

In [584]:
```
# Create training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state =0)
```

In [585]:
```
from sklearn.ensemble import RandomForestClassifier
```

In [590]:
```
(cl.predict(X_test) == Y_test).mean()
```

Out[590]: 0.556237218813906

Cross-validation

In [600]:
```
from sklearn.model_selection import KFold
```

In [601]:
```
nfolds=10
```

In [603]:
```
kf = KFold(n_splits=nfolds, random_state=0, shuffle=True)
```

In [605]:
```
(sk.model_selection.cross_val_score(cl, X, Y, cv=kf, n_jobs=-1, scoring='roc_auc')).mean()
#n_jobs=-1 means use all CPU
```

Out[605]: 0.5736645435095795

In [593]:
```
#confusion Matrix
from sklearn.metrics import confusion_matrix
```

In [594]:
```
import sklearn.metrics as met
```

In [595]:
```
confusion_matrix(Y_test, y_pred)
```

Out[595]:
```
array([[239,  21],
       [196,  33]], dtype=int64)
```

In [596]:
```
#Accuracy
met.accuracy_score(Y_test, y_pred)
```

Out[596]: 0.556237218813906

In [597]:
```
#Precision
met.precision_score(Y_test, y_pred)
```

Out[597]: 0.6111111111111112

In [598]:
```
#Recall
met.recall_score(Y_test, y_pred)
```

Out[598]: 0.14410480349344978

In [599]:
```
#AUC Score
met.roc_auc_score(Y_test, y_pred_proba)
#only need one side of y_pred
```

Out[599]: 0.5861689620423245

Q1： How Effective Resources Help Control Wildfire Duration Time

Machine Learning - Classification:

Which one is better? ¶

```
In [607]: maxAUC = -1
          bestCL = ""
          for cl in clfs:
              auc = sk.model_selection.cross_val_score(cl,X,Y,cv=kf,n_jobs=-1,scoring='roc_auc').mean()
              print (str(cl) + ' ' + str(auc))
              if auc > maxAUC:
                  bestCL = cl
                  maxAUC = auc
          print('*********************************************')
          print ('Best is... ' + str(bestCL) + ' ' + str(maxAUC))
```

```
DecisionTreeClassifier() 0.5669020123034356
RandomForestClassifier(n_jobs=-1) 0.5788542978818454
GaussianNB() 0.5234141744449083
LogisticRegression(n_jobs=-1) 0.5304800476325139
DecisionTreeClassifier() 0.5664252943806624
AdaBoostClassifier() 0.5654133928360521
QuadraticDiscriminantAnalysis() 0.5304331740569623
MLPClassifier() 0.5679811823727741
SVC() 0.5584607364583526
*********************************************
Best is... RandomForestClassifier(n_jobs=-1) 0.5788542978818454
```

Q1: How Effective Resources Help Control Wildfire Duration Time

Machine Learning - Clustering

```
In [579]: #Clustering with K-Means
          from sklearn.cluster import KMeans
          df_Kmean=df_dt.copy()

In [580]: clu = KMeans(n_clusters=3, random_state=0)
          clu.fit(df_Kmean)

Out[580]: KMeans(n_clusters=3, random_state=0)

In [581]: clu.labels_[:20]

Out[581]: array([0, 0, 0, 0, 1, 0, 1, 0, 0, 2, 2, 1, 1, 2, 2, 0, 2, 0, 2, 1])

In [582]: df2=pd.DataFrame.copy(df_dt)
          df2['cluster']=clu.labels_
          df2.groupby('cluster').mean()

Out[582]:
```

cluster	AirTankers	CrewsInvolved	Dozers	Engines	Helicopters	PersonnelInvolved	WaterTenders	MajorIncident	Duration_AboveAvg
0	0.032443	0.426845	0.148855	0.893766	0.078244	10.375954	0.240458	0.208651	0.473282
1	0.300000	50.700000	37.900000	173.100000	15.300000	2267.900000	40.700000	1.000000	0.400000
2	1.276596	17.000000	6.808511	29.042553	3.702128	596.489362	7.574468	0.957447	0.382979

Clustering conclusions:

For those minor incidents, people would just let it burn without taking any action. So the duration days for them became the longest.

However for the major incidents, the increase of resources doesn't necessarily reduce the probability of incidents' duration days below 84 days, except AirTankers. For AirTankers, as we see, the increase of use of AirTanker, do reduce the probability of incidents' duration days below 84 days.

Q1 continued..

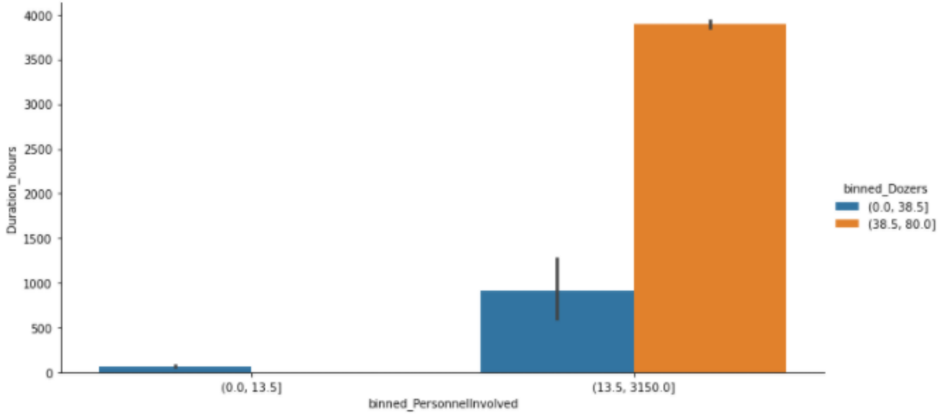
Validating the findings(1)

```
In [38]: df['binned_PersonnelInvolved'] = pd.cut(df['PersonnelInvolved'],\
                                                  bins=[0,13.5,3150])
```

```
In [ ]: df['binned_Dozers'] = pd.cut(df['Dozers'],bins=[0,38.5,80])
```

```
In [43]: sns.catplot(x='binned_PersonnelInvolved',y='Duration_hours',hue = 'binned_Dozers
                    kind='bar',data=df, aspect=2)
```

```
Out[43]: <seaborn.axisgrid.FacetGrid at 0x27fe476afa0>
```

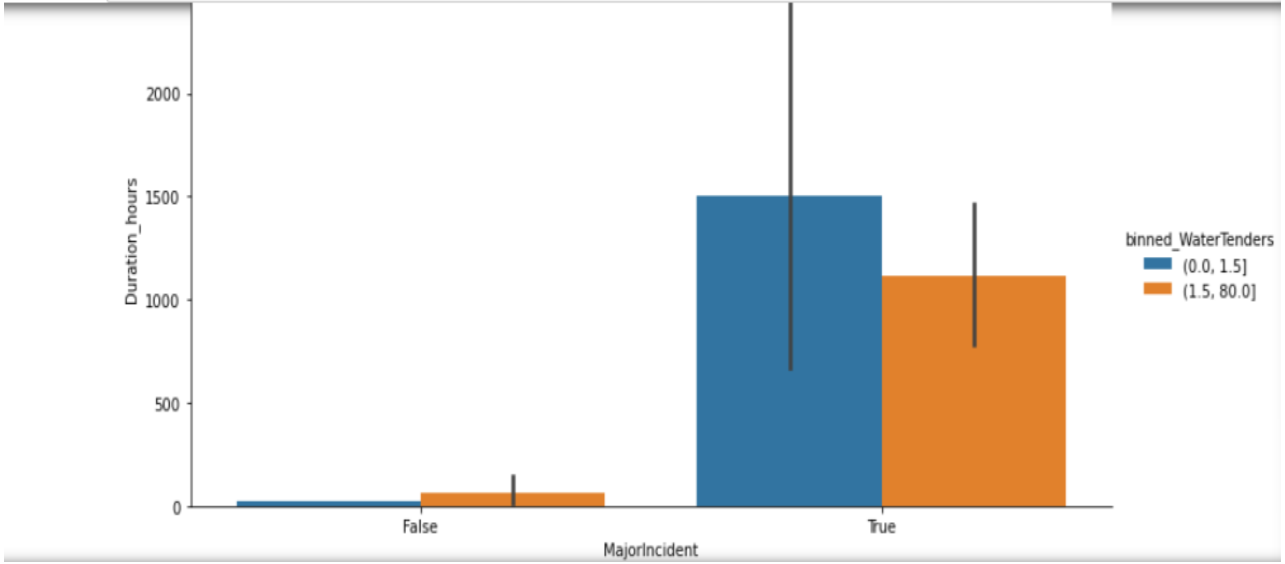


Q1 continued..

Validating the findings (2)

```
In [30]: df['binned_WaterTenders'] = pd.cut(df['WaterTenders'],\
                                             bins=[0,1.5,80])
```

```
In [31]: sns.catplot(x='MajorIncident',y='Duration_hours',hue = 'binned_WaterTenders',\
                    kind='bar',data=df, aspect=2)
```



Q2: What area need more resources - By Region (Northern California and Southern California)?

In [3216]:
```
#Create a datafram wich contains the Southern California counties
df_County_Classification = pd.DataFrame({'Counties' : ['Imperial', 'Los Angeles', 'Orange', \
    'Riverside', 'San Bernardino', 'San Diego', 'Santa Barbara', 'Ventura', 'San Luis Obispo', 'Kern'],\
   'NC' : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]})
```

In [3217]:
```
df_County_Classification
```

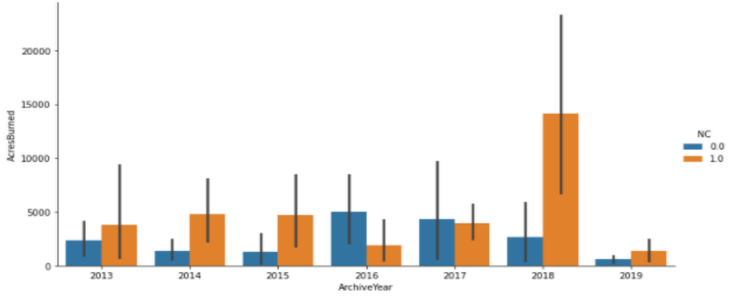
Out[3217]:

	Counties	NC
0	Imperial	0
1	Los Angeles	0
2	Orange	0
3	Riverside	0
4	San Bernardino	0
5	San Diego	0
6	Santa Barbara	0
7	Ventura	0
8	San Luis Obispo	0
9	Kern	0

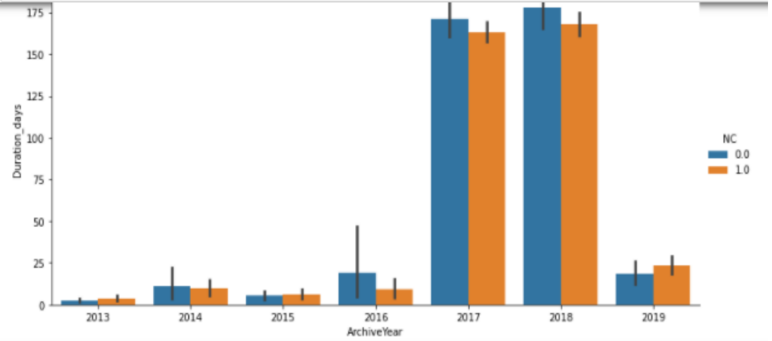
Acres and Duration by Region

In [3221]:
```
sns.catplot(x='ArchiveYear', y='AcresBurned', hue='NC', data=df, kind='bar', aspect=2)
```

Out[3221]: <seaborn.axisgrid.FacetGrid at 0x1d7691477f0>



In [3304]:
```
sns.catplot(x='ArchiveYear', y='Duration_days', hue='NC', data=df, kind='bar', aspect=2)
```



What we did:

Create a DataFrame containing list of Southern California counties and merge to our dataset df.

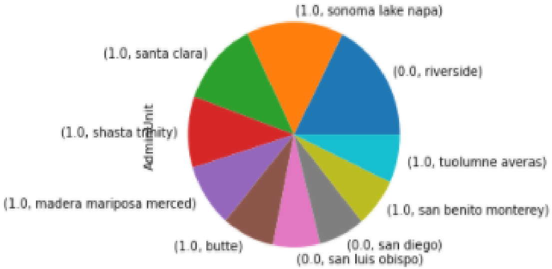
Our Finding:

1. *Over year 2013-2019, there are more AcresBurned in Northern California than in Southern California except year 2016;*

1. *However Duration time in Northern California are a little less than those in Southern California except year 2019.*

Q2: What area need more resources - By Region (NC/SC) and AdminUnit?

```
In [3260]: df.groupby(['NC','AdminUnit'])['AdminUnit'].count().nlargest(10).plot(kind='pie')
Out[3260]: <AxesSubplot:ylabel='AdminUnit'>
```



Our Finding:

1. *From year 2013-2019, the top three AdminUnit have taken care of the most wildfire incidents is Riverside in Southern California, Sonoma Lake Nape in Northern California, and Santa Clara in Northern California.*

1. *They definitely deserve more resources!!*

Q3 Analysis of Wildfires in Santa Clara County

Santa Clara County Fire Details

```
df_sc = df[df.Counties == 'Santa Clara']
```

Statistics for fires that took place in Santa Clara County

```
df_sc.describe()
```

[67]:

	AcresBurned	AirTankers	ArchiveYear	CrewsInvolved	Dozers	Engines	Fatalities	Helicopters	Injuries	Latitude	Longitude	PercentContair
count	39.000000	39.0	39.000000	39.000000	39.000000	39.000000	39.0	39.0	39.0	39.000000	39.000000	3
mean	193.435897	0.0	2017.205128	0.051282	0.025641	0.205128	0.0	0.0	0.0	32.481574	-106.088387	10
std	706.100291	0.0	1.734776	0.223456	0.160128	0.922796	0.0	0.0	0.0	12.619947	41.215123	
min	16.000000	0.0	2013.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.000000	-121.888070	10
25%	41.000000	0.0	2016.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	37.065897	-121.774340	10
50%	70.000000	0.0	2018.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	37.216380	-121.697840	10
75%	100.500000	0.0	2019.000000	0.000000	0.000000	0.000000	0.0	0.0	0.0	37.370098	-121.545690	10
max	4474.000000	0.0	2019.000000	1.000000	1.000000	5.000000	0.0	0.0	0.0	37.660740	0.000000	10

Q3 Analysis of Wildfires in Santa Clara County

Santa Clara County Fire Details

Statistics for fires that took place in Santa Clara County

```
df_sc.describe()
```

57]:

de	PercentContained	PersonnelInvolved	StructuresDamaged	StructuresDestroyed	StructuresEvacuated	StructuresThreatened	WaterTenders	Duration_days
00	39.0	39.000000	39.000000	39.000000	39.0	39.0	39.000000	39.000000
87	100.0	5.435897	0.025641	0.743590	0.0	0.0	0.128205	89.568372
23	0.0	25.747555	0.160128	4.482234	0.0	0.0	0.800641	93.287478
70	100.0	0.000000	0.000000	0.000000	0.0	0.0	0.000000	-0.041667
40	100.0	0.000000	0.000000	0.000000	0.0	0.0	0.000000	1.125000
40	100.0	0.000000	0.000000	0.000000	0.0	0.0	0.000000	84.874947
90	100.0	0.000000	0.000000	0.000000	0.0	0.0	0.000000	173.750000
00	100.0	150.000000	1.000000	28.000000	0.0	0.0	5.000000	357.791667

Q3 Analysis of Wildfires in Santa Clara County
How many highway fires in Santa Clara County?

First placing a boolean mask to only show fires that happened in Santa Clara County

```
df_sc = df[df.Counties == 'Santa Clara']
```

Then adding a lambda function to only show fires that occurred on a highway

```
df_sc[df_sc.Location.apply(lambda s : True if 'highway' in s.lower() else False)]
```

4]:

	AcresBurned	Active	AdminUnit	AirTankers	ArchiveYear	CalFireIncident	CanonicalUrl	ConditionStatement	ControlStatement	Counties
1582	29.0	False	CAL FIRE Santa Clara Unit	0.0	2019	True	/incidents/2019/10/7/point-fire/	Unknown	Unknown	Santa Clara

Checking to see what year this fire occurred

```
df_sc[df_sc.Location.apply(lambda s : True if 'highway' in s.lower() else False)]['ArchiveYear']
```

5]:
```
1582    2019
Name: ArchiveYear, dtype: int64
```

```
len(df_sc[df_sc.Location.apply(lambda s : True if 'highway' in s.lower() else False)])
```

6]: 1

Thank you