Importing Libraries:

In [1]:
```
import tweepy as tw
import pandas as pd
import csv
from celluloid import Camera
import matplotlib.animation as animation
import numpy as np
from nltk.classify import NaiveBayesClassifier
import re
from nltk.corpus import subjectivity
from nltk.sentiment import SentimentAnalyzer
from nltk.sentiment.util import *
import matplotlib.pyplot as plt
```

variables that control user credential:

In [2]:
```
ACCESS_Token = "1263149831339626497-dVJZEnNj1TVp7nGWkKSVYWv6XlubgK"
ACCESS_Token_Secret = "csgqSA3Q6G6jr9cxjl1NUd8qaQRsK9ByrRQPA9hS9GuPE"
CONSUMER_API_KEY = "0QncdDMpWYQ9fewsMaSz2rBzJ"
CONSUMER_API_SECRET = "dBXhrhe9BIGPSeQFtERMIjGYsOU7D4Kn281gFEvBPpHpFIG7H8"
```

In [3]:
```
#creating the authentication object:
auth = tw.OAuthHandler(CONSUMER_API_KEY,CONSUMER_API_SECRET)

#Setting access token and secret:
auth.set_access_token(ACCESS_Token,ACCESS_Token_Secret)

#creating the API object while passing in auth info:
api= tw.API(auth,wait_on_rate_limit=True)
```

Using function for cleaning the text data:

In [4]:
```
def cleanText(tw):
    cleanedText = re.sub(r"http\S+", "", tw)
    cleanedText = re.sub('@[^\s]+',"", cleanedText)
    cleanedText = re.sub(r"[^a-zA-Z0-9]+", ' ', cleanedText)
    cleanedText = re.sub(r"\d+", ' ', cleanedText)
    cleanedText = re.sub(r'\b\w{1,2}\b','',cleanedText)
    return cleanedText
```

Fetching tweets for #Altcoin and saving into csv file

In [ ]:
```
csvFile = open('Tweets_Altcoin.csv', 'a')

with open('Tweets_Altcoin.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames = ["Tweet_Id", "Time_Of_Tweet",
"User_ID", "Tweet_Text"])
    writer.writeheader()
    csvWriter = csv.writer(csvFile)

for tweet in tw.Cursor(api.search,q='"#Altcoin"',rpp=100,count=100, lang="en",
since="2020-06-22",tweet_mode='extended').items():
        new_text=cleanText(tweet.full_text)
        csvWriter.writerow([tweet.id,tweet.created_at,tweet.user.id, new_text
])
```

Fetching tweets for #Bitcoin and saving into csv file

In [ ]:
```
csvFile = open('Tweets_Bitcoin.csv', 'a')

with open('Tweets_Bitcoin.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames = ["Tweet_Id", "Time_Of_Tweet",
"User_ID", "Tweet_Text"])
    writer.writeheader()
    csvWriter = csv.writer(csvFile)

for tweet in tw.Cursor(api.search,q='"#Bitcoin"',rpp=100,count=100, lang="en",
since="2020-06-22",tweet_mode='extended').items():
        new_text=cleanText(tweet.full_text)
        csvWriter.writerow([tweet.id,tweet.created_at,tweet.user.id, new_text
])
```

Fetching tweets for #Coindesk and saving into csv file

In [37]:
```
csvFile = open('Tweets_Coindesk.csv', 'a')

with open('Tweets_Coindesk.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames = ["Tweet_Id", "Time_Of_Tweet",
"User_ID", "Tweet_Text"])
    writer.writeheader()
    csvWriter = csv.writer(csvFile)

for tweet in tw.Cursor(api.search,q='"#Coindesk"',rpp=100,count=100, lang="en"
,since="2020-06-22",tweet_mode='extended').items():
        new_text=cleanText(tweet.full_text)
        csvWriter.writerow([tweet.id,tweet.created_at,tweet.user.id, new_text
])
```

Fetching tweets for #Cryptocurrency and saving into csv file

In [ ]:
```
csvFile = open('Tweets_Cryptocurrency.csv', 'a')

with open('Tweets_Cryptocurrency.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames = ["Tweet_Id", "Time_Of_Tweet",
"User_ID", "Tweet_Text"])
    writer.writeheader()
    csvWriter = csv.writer(csvFile)

for tweet in tw.Cursor(api.search,q='"#Cryptocurrency"',rpp=100,count=100, lan
g="en",since="2020-06-22",tweet_mode='extended').items():
        new_text=cleanText(tweet.full_text)
        csvWriter.writerow([tweet.id,tweet.created_at,tweet.user.id, new_text
])
```

Fetching tweets for #Gold and saving into csv file

In [ ]:
```
csvFile = open('Tweets_Gold.csv', 'a')

with open('Tweets_Gold.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames = ["Tweet_Id", "Time_Of_Tweet",
"User_ID", "Tweet_Text"])
    writer.writeheader()
    csvWriter = csv.writer(csvFile)

for tweet in tw.Cursor(api.search,q='"#Gold"',rpp=100,count=100, lang="en",sin
ce="2020-06-22",tweet_mode='extended').items():
        new_text=cleanText(tweet.full_text)
        csvWriter.writerow([tweet.id,tweet.created_at,tweet.user.id, new_text
])
```

Fetching tweets for #APPL and saving into csv file

In [ ]:
```
csvFile = open('Tweets_APPL.csv', 'a')

with open('Tweets_APPL.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames = ["Tweet_Id", "Time_Of_Tweet",
"User_ID", "Tweet_Text"])
    writer.writeheader()
    csvWriter = csv.writer(csvFile)

for tweet in tw.Cursor(api.search,q='"#APPLE"',rpp=100,count=100, lang="en",si
nce="2020-06-22",tweet_mode='extended').items():
        new_text=cleanText(tweet.full_text)
        csvWriter.writerow([tweet.id,tweet.created_at,tweet.user.id, new_text
])
```

Fetching tweets for #GOOG and saving into csv file

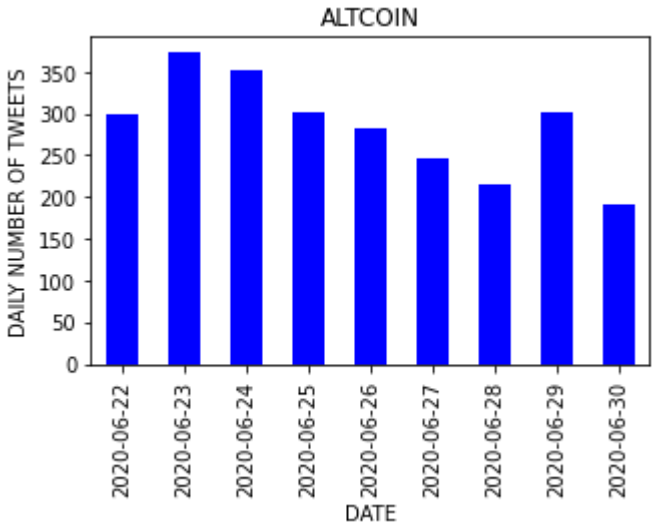
In [14]:
```
csvFile = open('Tweets_GOOG.csv', 'a')

with open('Tweets_GOOG.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames = ["Tweet_Id", "Time_Of_Tweet",
"User_ID", "Tweet_Text"])
    writer.writeheader()
    csvWriter = csv.writer(csvFile)

for tweet in tw.Cursor(api.search,q='"#GOOGLE"',rpp=100,count=100, lang="en",s
ince="2020-06-22",tweet_mode='extended').items():
        new_text=cleanText(tweet.full_text)
        csvWriter.writerow([tweet.id,tweet.created_at,tweet.user.id, new_text
])
```

Fetching tweets for #YHOO and saving into csv file

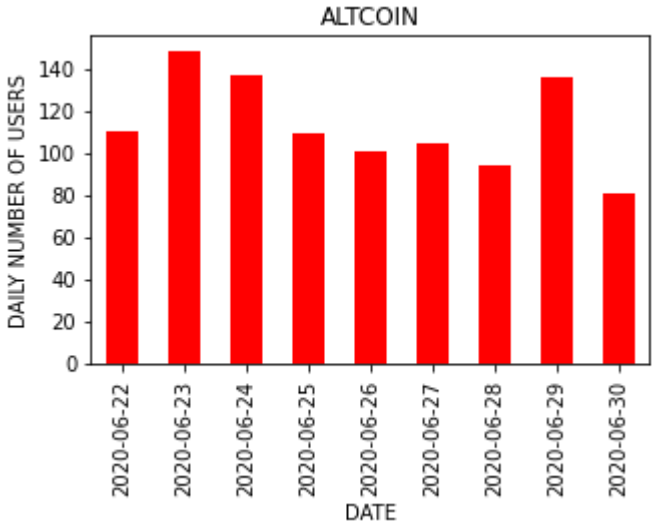
In [16]:
```
csvFile = open('Tweets_YHOO.csv', 'a')

with open('Tweets_YHOO.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames = ["Tweet_Id", "Time_Of_Tweet",
"User_ID", "Tweet_Text"])
    writer.writeheader()
    csvWriter = csv.writer(csvFile)

for tweet in tw.Cursor(api.search,q='"#YAHOO"',rpp=100,count=100, lang="en",si
nce="2020-06-22",tweet_mode='extended').items():
        new_text=cleanText(tweet.full_text)
        csvWriter.writerow([tweet.id,tweet.created_at,tweet.user.id, new_text
])
```

Visualizing data, finding total numbers of tweets and users: #Altcoin

In [4]:
```
df = pd.read_csv('Tweets_Altcoin.csv')
df.head()
df['year'] = pd.DatetimeIndex(df['Time_Of_Tweet']).year
df['date'] = pd.DatetimeIndex(df['Time_Of_Tweet']).date
df['month'] = pd.DatetimeIndex(df['Time_Of_Tweet']).month
df.head()

tweet_count=df.groupby('date')['Tweet_Id'].nunique()
total_tweet_count=df['Tweet_Id'].nunique()
tweet_count.plot(kind='bar',color='blue',figsize=(5,3))
plt.title('ALTCOIN')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF TWEETS')
plt.show()
print(tweet_count)
print(total_tweet_count)

user_count=df.groupby('date')['User_ID'].nunique()
total_user_count=df['User_ID'].nunique()
user_count.plot(kind='bar',color='red',figsize=(5,3))
plt.title('ALTCOIN')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF USERS')
plt.show()
print(user_count)
print(total_user_count)
```



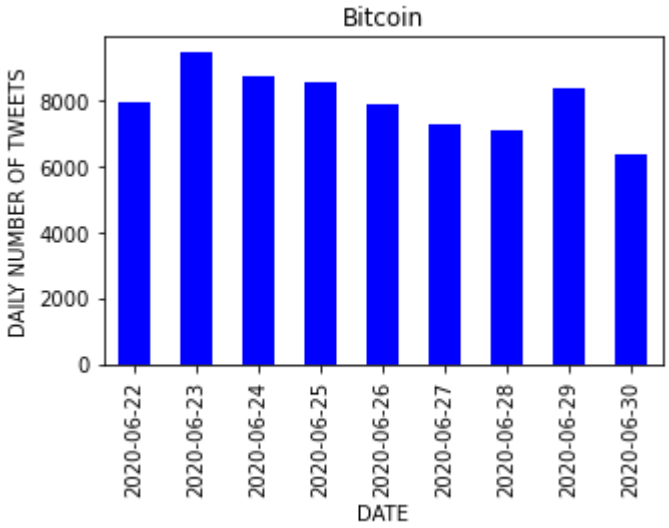

```
date
2020-06-22    300
2020-06-23    373
2020-06-24    351
2020-06-25    301
2020-06-26    282
2020-06-27    246
2020-06-28    215
2020-06-29    301
2020-06-30    191
Name: Tweet_Id, dtype: int64
2560
```



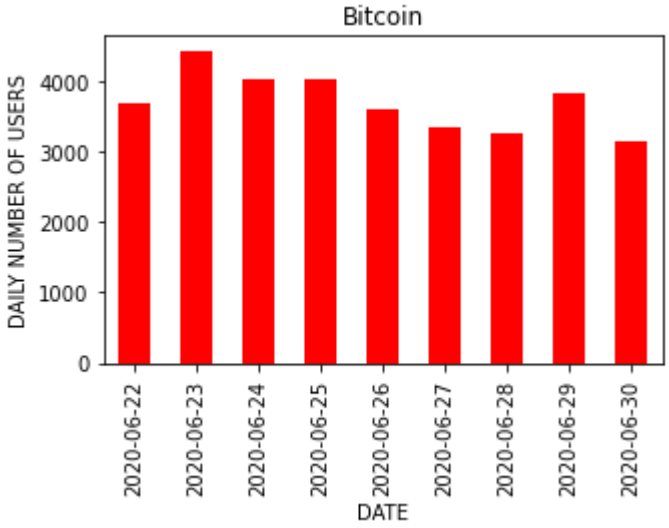
```
date
2020-06-22    110
2020-06-23    148
2020-06-24    137
2020-06-25    109
2020-06-26    101
2020-06-27    104
2020-06-28     94
2020-06-29    136
2020-06-30     81
Name: User_ID, dtype: int64
562
```

Visualizing data, finding total numbers of tweets and users: #Bitcoin

In [14]:
```
df = pd.read_csv('Tweets_Bitcoin.csv')
df.head()
df['year'] = pd.DatetimeIndex(df['Time_Of_Tweet']).year
df['date'] = pd.DatetimeIndex(df['Time_Of_Tweet']).date
df['month'] = pd.DatetimeIndex(df['Time_Of_Tweet']).month
df.head()

tweet_count=df.groupby('date')['Tweet_Id'].nunique()
total_tweet_count=df['Tweet_Id'].nunique()
tweet_count.plot(kind='bar',color='blue',figsize=(5,3))
plt.title('Bitcoin')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF TWEETS')
plt.show()
print(tweet_count)
print(total_tweet_count)

user_count=df.groupby('date')['User_ID'].nunique()
total_user_count=df['User_ID'].nunique()
user_count.plot(kind='bar',color='red',figsize=(5,3))
plt.title('Bitcoin')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF USERS')
plt.show()
print(user_count)
print(total_user_count)
```



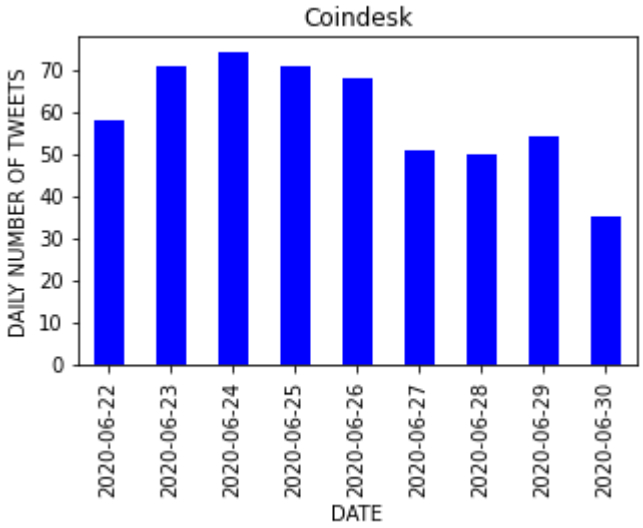

```
date
2020-06-22    7940
2020-06-23    9472
2020-06-24    8745
2020-06-25    8581
2020-06-26    7920
2020-06-27    7262
2020-06-28    7109
2020-06-29    8373
2020-06-30    6355
Name: Tweet_Id, dtype: int64
71757
```



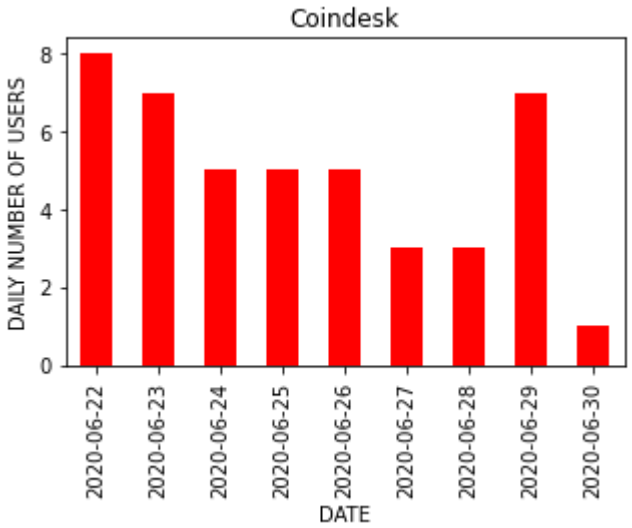
```
date
2020-06-22    3678
2020-06-23    4416
2020-06-24    4035
2020-06-25    4015
2020-06-26    3600
2020-06-27    3346
2020-06-28    3257
2020-06-29    3817
2020-06-30    3137
Name: User_ID, dtype: int64
16571
```

Visualizing data, finding total numbers of tweets and users: #Coindesk

In [18]:
```
df = pd.read_csv('Tweets_Coindesk.csv')
df.head()
df['year'] = pd.DatetimeIndex(df['Time_Of_Tweet']).year
df['date'] = pd.DatetimeIndex(df['Time_Of_Tweet']).date
df['month'] = pd.DatetimeIndex(df['Time_Of_Tweet']).month
df.head()

tweet_count=df.groupby('date')['Tweet_Id'].nunique()
total_tweet_count=df['Tweet_Id'].nunique()
tweet_count.plot(kind='bar',color='blue',figsize=(5,3))
plt.title('Coindesk')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF TWEETS')
plt.show()
print(tweet_count)
print(total_tweet_count)

user_count=df.groupby('date')['User_ID'].nunique()
total_user_count=df['User_ID'].nunique()
user_count.plot(kind='bar',color='red',figsize=(5,3))
plt.title('Coindesk')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF USERS')
plt.show()
print(user_count)
print(total_user_count)
```



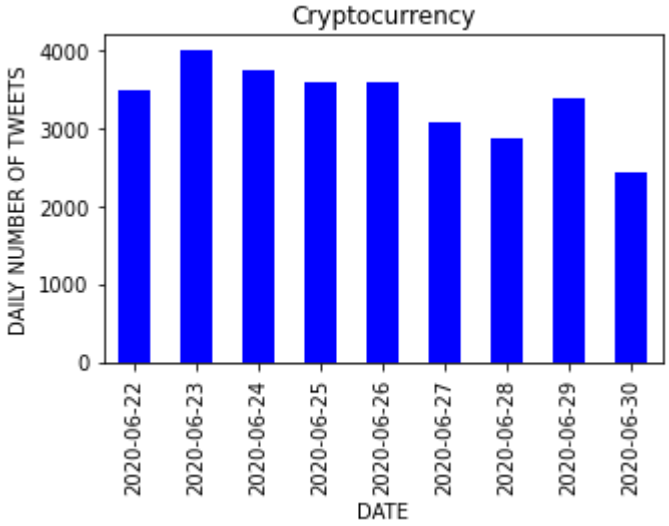

```
date
2020-06-22    58
2020-06-23    71
2020-06-24    74
2020-06-25    71
2020-06-26    68
2020-06-27    51
2020-06-28    50
2020-06-29    54
2020-06-30    35
Name: Tweet_Id, dtype: int64
532
```



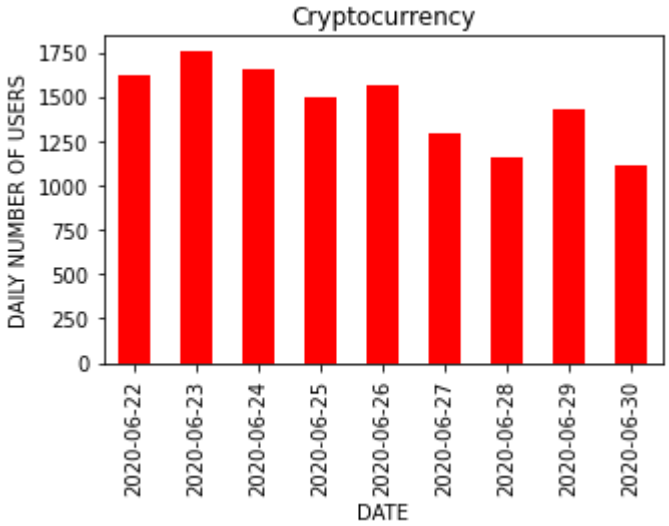
```
date
2020-06-22    8
2020-06-23    7
2020-06-24    5
2020-06-25    5
2020-06-26    5
2020-06-27    3
2020-06-28    3
2020-06-29    7
2020-06-30    1
Name: User_ID, dtype: int64
24
```

Visualizing data, finding total numbers of tweets and users: #Cryptocurrency

In [19]:
```
df = pd.read_csv('Tweets_Cryptocurrency.csv')
df.head()
df['year'] = pd.DatetimeIndex(df['Time_Of_Tweet']).year
df['date'] = pd.DatetimeIndex(df['Time_Of_Tweet']).date
df['month'] = pd.DatetimeIndex(df['Time_Of_Tweet']).month
df.head()

tweet_count=df.groupby('date')['Tweet_Id'].nunique()
total_tweet_count=df['Tweet_Id'].nunique()
tweet_count.plot(kind='bar',color='blue',figsize=(5,3))
plt.title('Cryptocurrency')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF TWEETS')
plt.show()
print(tweet_count)
print(total_tweet_count)

user_count=df.groupby('date')['User_ID'].nunique()
total_user_count=df['User_ID'].nunique()
user_count.plot(kind='bar',color='red',figsize=(5,3))
plt.title('Cryptocurrency')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF USERS')
plt.show()
print(user_count)
print(total_user_count)
```



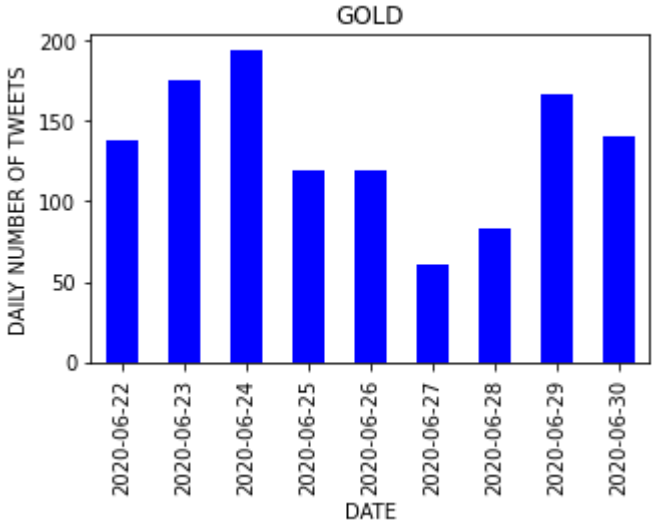

```
date
2020-06-22    3499
2020-06-23    3995
2020-06-24    3736
2020-06-25    3601
2020-06-26    3592
2020-06-27    3081
2020-06-28    2875
2020-06-29    3394
2020-06-30    2441
Name: Tweet_Id, dtype: int64
30214
```



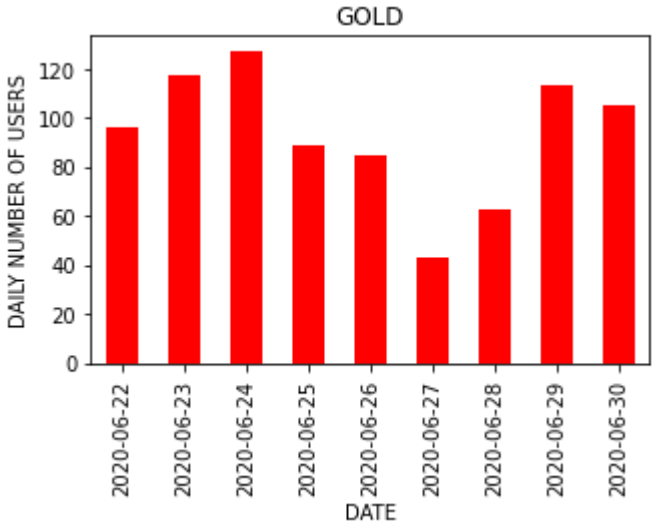
```
date
2020-06-22    1624
2020-06-23    1753
2020-06-24    1653
2020-06-25    1497
2020-06-26    1566
2020-06-27    1296
2020-06-28    1157
2020-06-29    1424
2020-06-30    1110
Name: User_ID, dtype: int64
7163
```

Visualizing data, finding total numbers of tweets and users: #Gold

In [20]:
```
df = pd.read_csv('Tweets_gold.csv')
df.head()
df['year'] = pd.DatetimeIndex(df['Time_Of_Tweet']).year
df['date'] = pd.DatetimeIndex(df['Time_Of_Tweet']).date
df['month'] = pd.DatetimeIndex(df['Time_Of_Tweet']).month
df.head()


tweet_count=df.groupby('date')['Tweet_Id'].nunique()
total_tweet_count=df['Tweet_Id'].nunique()
tweet_count.plot(kind='bar',color='blue',figsize=(5,3))
plt.title('GOLD')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF TWEETS')
plt.show()
print(tweet_count)
print(total_tweet_count)

user_count=df.groupby('date')['User_ID'].nunique()
total_user_count=df['User_ID'].nunique()
user_count.plot(kind='bar',color='red',figsize=(5,3))
plt.title('GOLD')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF USERS')
plt.show()
print(user_count)
print(total_user_count)
```



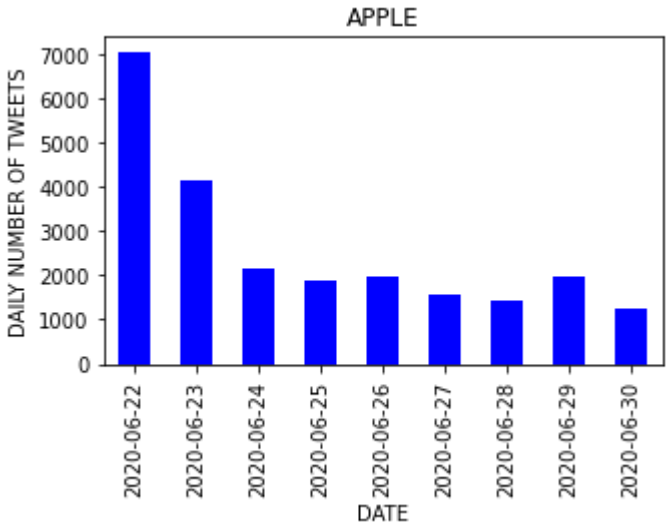

```
date
2020-06-22    138
2020-06-23    175
2020-06-24    194
2020-06-25    119
2020-06-26    119
2020-06-27     61
2020-06-28     83
2020-06-29    167
2020-06-30    141
Name: Tweet_Id, dtype: int64
1197
```



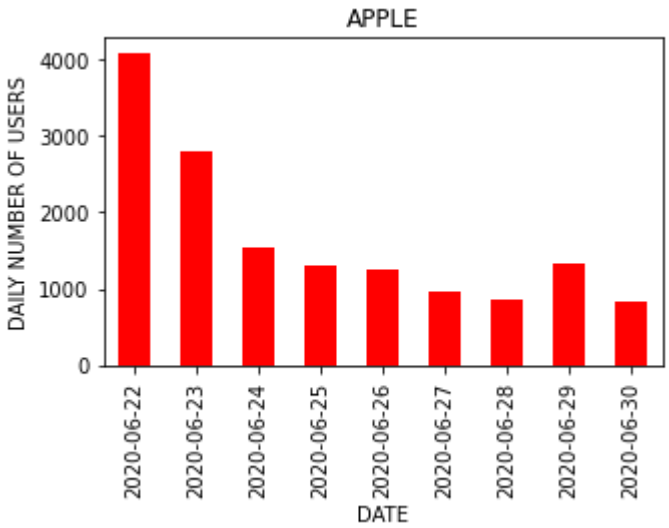
```
date
2020-06-22     96
2020-06-23    117
2020-06-24    127
2020-06-25     89
2020-06-26     85
2020-06-27     43
2020-06-28     63
2020-06-29    113
2020-06-30    105
Name: User_ID, dtype: int64
532
```

Visualizing data, finding total numbers of tweets and users: #APPL

In [21]:
```
df = pd.read_csv('Tweets_APPL.csv')
df.head()
df['year'] = pd.DatetimeIndex(df['Time_Of_Tweet']).year
df['date'] = pd.DatetimeIndex(df['Time_Of_Tweet']).date
df['month'] = pd.DatetimeIndex(df['Time_Of_Tweet']).month
df.head()


tweet_count=df.groupby('date')['Tweet_Id'].nunique()
total_tweet_count=df['Tweet_Id'].nunique()
tweet_count.plot(kind='bar',color='blue',figsize=(5,3))
plt.title('APPLE')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF TWEETS')
plt.show()
print(tweet_count)
print(total_tweet_count)

user_count=df.groupby('date')['User_ID'].nunique()
total_user_count=df['User_ID'].nunique()
user_count.plot(kind='bar',color='red',figsize=(5,3))
plt.title('APPLE')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF USERS')
plt.show()
print(user_count)
print(total_user_count)
```



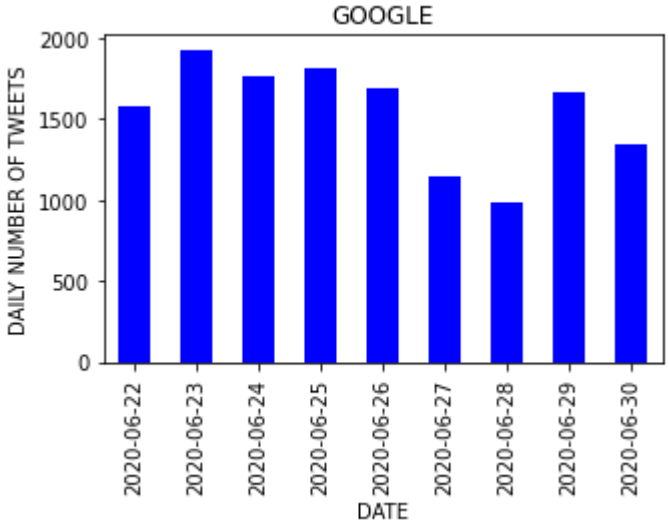

```
date
2020-06-22    7042
2020-06-23    4142
2020-06-24    2166
2020-06-25    1905
2020-06-26    1977
2020-06-27    1577
2020-06-28    1414
2020-06-29    1953
2020-06-30    1257
Name: Tweet_Id, dtype: int64
23433
```



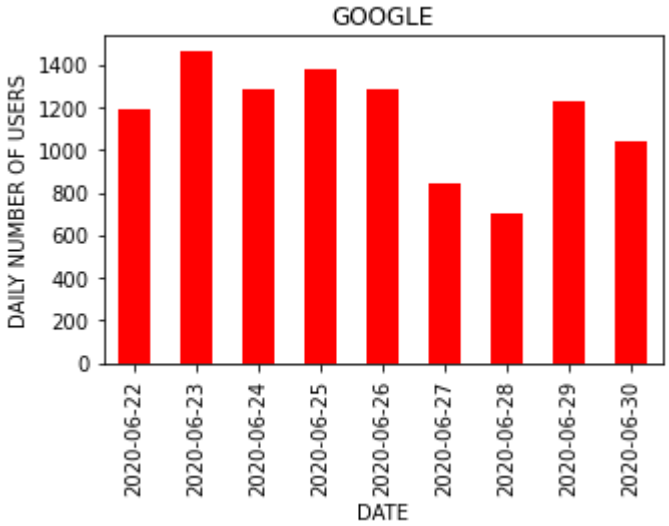
```
date
2020-06-22    4083
2020-06-23    2811
2020-06-24    1551
2020-06-25    1290
2020-06-26    1260
2020-06-27     956
2020-06-28     861
2020-06-29    1317
2020-06-30     835
Name: User_ID, dtype: int64
11545
```

Visualizing data, finding total numbers of tweets and users: #GOOG

In [22]:
```
df = pd.read_csv('Tweets_GOOG.csv')
df.head()
df['year'] = pd.DatetimeIndex(df['Time_Of_Tweet']).year
df['date'] = pd.DatetimeIndex(df['Time_Of_Tweet']).date
df['month'] = pd.DatetimeIndex(df['Time_Of_Tweet']).month
df.head()


tweet_count=df.groupby('date')['Tweet_Id'].nunique()
total_tweet_count=df['Tweet_Id'].nunique()
tweet_count.plot(kind='bar',color='blue',figsize=(5,3))
plt.title('GOOGLE')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF TWEETS')
plt.show()
print(tweet_count)
print(total_tweet_count)

user_count=df.groupby('date')['User_ID'].nunique()
total_user_count=df['User_ID'].nunique()
user_count.plot(kind='bar',color='red',figsize=(5,3))
plt.title('GOOGLE')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF USERS')
plt.show()
print(user_count)
print(total_user_count)
```



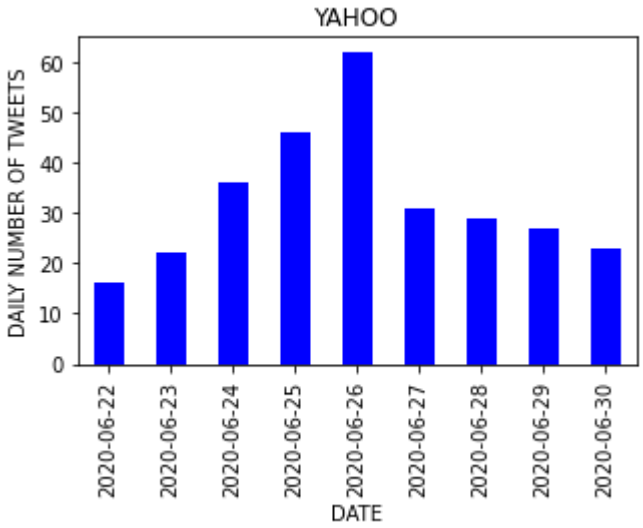

```
date
2020-06-22    1575
2020-06-23    1923
2020-06-24    1764
2020-06-25    1820
2020-06-26    1691
2020-06-27    1150
2020-06-28     990
2020-06-29    1663
2020-06-30    1349
Name: Tweet_Id, dtype: int64
13925
```



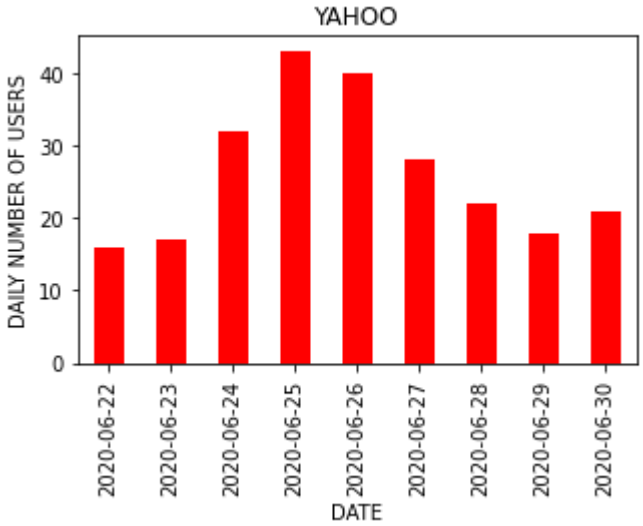
```
date
2020-06-22    1192
2020-06-23    1461
2020-06-24    1284
2020-06-25    1382
2020-06-26    1280
2020-06-27     846
2020-06-28     700
2020-06-29    1226
2020-06-30    1036
Name: User_ID, dtype: int64
7766
```

Visualizing data, finding total numbers of tweets and users: #YHOO

In [23]:
```
df = pd.read_csv('Tweets_YHOO.csv')
df.head()
df['year'] = pd.DatetimeIndex(df['Time_Of_Tweet']).year
df['date'] = pd.DatetimeIndex(df['Time_Of_Tweet']).date
df['month'] = pd.DatetimeIndex(df['Time_Of_Tweet']).month
df.head()


tweet_count=df.groupby('date')['Tweet_Id'].nunique()
total_tweet_count=df['Tweet_Id'].nunique()
tweet_count.plot(kind='bar',color='blue',figsize=(5,3))
plt.title('YAHOO')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF TWEETS')
plt.show()
print(tweet_count)
print(total_tweet_count)

user_count=df.groupby('date')['User_ID'].nunique()
total_user_count=df['User_ID'].nunique()
user_count.plot(kind='bar',color='red',figsize=(5,3))
plt.title('YAHOO')
plt.xlabel('DATE')
plt.ylabel('DAILY NUMBER OF USERS')
plt.show()
print(user_count)
print(total_user_count)
```




```
date
2020-06-22    16
2020-06-23    22
2020-06-24    36
2020-06-25    46
2020-06-26    62
2020-06-27    31
2020-06-28    29
2020-06-29    27
2020-06-30    23
Name: Tweet_Id, dtype: int64
292
```



```
date
2020-06-22    16
2020-06-23    17
2020-06-24    32
2020-06-25    43
2020-06-26    40
2020-06-27    28
2020-06-28    22
2020-06-29    18
2020-06-30    21
Name: User_ID, dtype: int64
201
```

In [ ]: