In [1]:
```
import pandas as pd

import re
from emoji import UNICODE_EMOJI
from textblob import TextBlob
import altair as alt
import numpy as np
from collections import Counter
import string

import nltk
nltk.download('vader_lexicon')
nltk.download('brown')
nltk.download('punkt')
nltk.download('stopwords')

from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords

import matplotlib.pyplot as plt
%matplotlib inline
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /home/jovyan/nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
[nltk_data] Downloading package brown to /home/jovyan/nltk_data...
[nltk_data]   Package brown is already up-to-date!
[nltk_data] Downloading package punkt to /home/jovyan/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /home/jovyan/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

The data cleaning/manipulation technique/functions

In [2]:

```
def extract_tags(text):
    return re.findall("#([a-zA-Z0-9_]{1,50})", text)

def extract_emoji(text):
    return [ch for ch in text if ch in UNICODE_EMOJI['en']]


def clean_tweet(txt):
    temp = re.sub("@[A-Za-z0-9_]+","", txt)
    temp1 = re.sub("#[A-Za-z0-9_]+","", temp)
    temp2 = re.sub(r"http\S+", "", temp1)

    result=''.join(i for i in temp2.lower() if (i.isalpha() or i==' '))
    return result

def word_list(tweet):

    lst = word_tokenize(tweet)
    lst1 = []
    stops = list(stopwords.words('english'))
    for w in lst:
        if w not in stops:
            lst1.append(w)

    return lst1

def sentiment(tweet):
    blob = TextBlob(tweet)

    return blob.sentiment.polarity


def get_date(date):

    return date[:10]

def get_hour(date):

    return date[11:13]
def get_10min(date):

    return date[14]+'0'

def get_min(date):
```

```
    return date[14:16]

def firm_pos(score):
    if score >= 0.7:
        return 1
    else: return 0

def pos(score):
    if (score >= 0.25) & (score < 0.7):
        return 1
    else: return 0

def neutral(score):
    if (score >= -0.25) & (score < 0.25):
        return 1
    else: return 0

def neg(score):
    if (score > -0.7) & (score < -0.25):
        return 1
    else: return 0

def firm_neg(score):
    if score <= -0.7:
        return 1
    else: return 0
```

Import data, check duplicate or missing value, remove if exists.

In [3]:
```
df = pd.read_csv('Project Data/Lebron 2020 playoff.csv')

df['id'].duplicated(keep='last').sum()
```

Out[3]:
```
0
```

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
id      0
date    1
text    1
dtype: int64
```

In [5]:
```
# drop row with missing value and reset index

df = df.dropna(how='any').reset_index()
df.drop(columns=['index'],inplace=True)
```

Apply data cleaning/manipulation techniques on the data, we now have the used words, tags, emojis, sentiment score, and specific date/hour/min data.

In [6]:
```
df['tags']= df.apply(lambda row: extract_tags(row['text']), axis=1)
df['emojis']= df.apply(lambda row: extract_emoji(row['text']), axis=1)
df['clean_text']= df.apply(lambda row: clean_tweet(row['text']), axis=1)
df['words']= df.apply(lambda row: word_list(row['clean_text']), axis=1)
df['sentiment_score']= df.apply(lambda row: sentiment(row['clean_text']), axis=1)
df['day']= df.apply(lambda row: get_date(row['date']), axis=1)
df['hour']= df.apply(lambda row: get_hour(row['date']), axis=1)
df['10min']= df.apply(lambda row: get_10min(row['date']), axis=1)
df['min']= df.apply(lambda row: get_min(row['date']), axis=1)
df['POS']= df.apply(lambda row: firm_pos(row['sentiment_score']), axis=1)
df['pos']= df.apply(lambda row: pos(row['sentiment_score']), axis=1)
df['neu']= df.apply(lambda row: neutral(row['sentiment_score']), axis=1)
df['neg']= df.apply(lambda row: neg(row['sentiment_score']), axis=1)
df['NEG']= df.apply(lambda row: firm_neg(row['sentiment_score']), axis=1)

df.head()
```

Out[6]:

	id	date	text	tags	emojis	clean_text	words	sentiment_score	day	hour	10min	min	POS	pos	neu
0	1295510581181480960	2020-08-17 23:59:26+00:00	@Chris_Auther @KingJames @AntDavis23 I need th...	[]	[]	i need that	[need]	0.0	2020-08-17	23	50	59	0	0	1
1	1295510388411334656	2020-08-17 23:58:40+00:00	25 hours out.. what time you activating? #Zero...	[ZeroDark23, GoatMode]	[👑, 🐐]	hours out what time you activating	[hours, time, activating]	0.0	2020-08-17	23	50	58	0	0	1
2	1295510348775096320	2020-08-17 23:58:31+00:00	@bigreggie85 @Lakers @Keefmorris @KingJames @k...	[]	[]	are you a boy	[boy]	0.0	2020-08-17	23	50	58	0	0	1
3	1295510345268830208	2020-08-17 23:58:30+00:00	@DeanDTD Lebron James, Michael Jackson, Bernie...	[]	[]	lebron james michael jackson bernie mac post ...	[lebron, james, michael, jackson, bernie, mac,...	0.0	2020-08-17	23	50	58	0	0	1
4	1295510271163805699	2020-08-17 23:58:12+00:00	@washingtonpost I really don't want to hear an...	[]	[]	i really dont want to hear anymore from micha...	[really, dont, want, hear, anymore, michael, b...	0.2	2020-08-17	23	50	58	0	0	1

Flow of Tweet count & sentiment

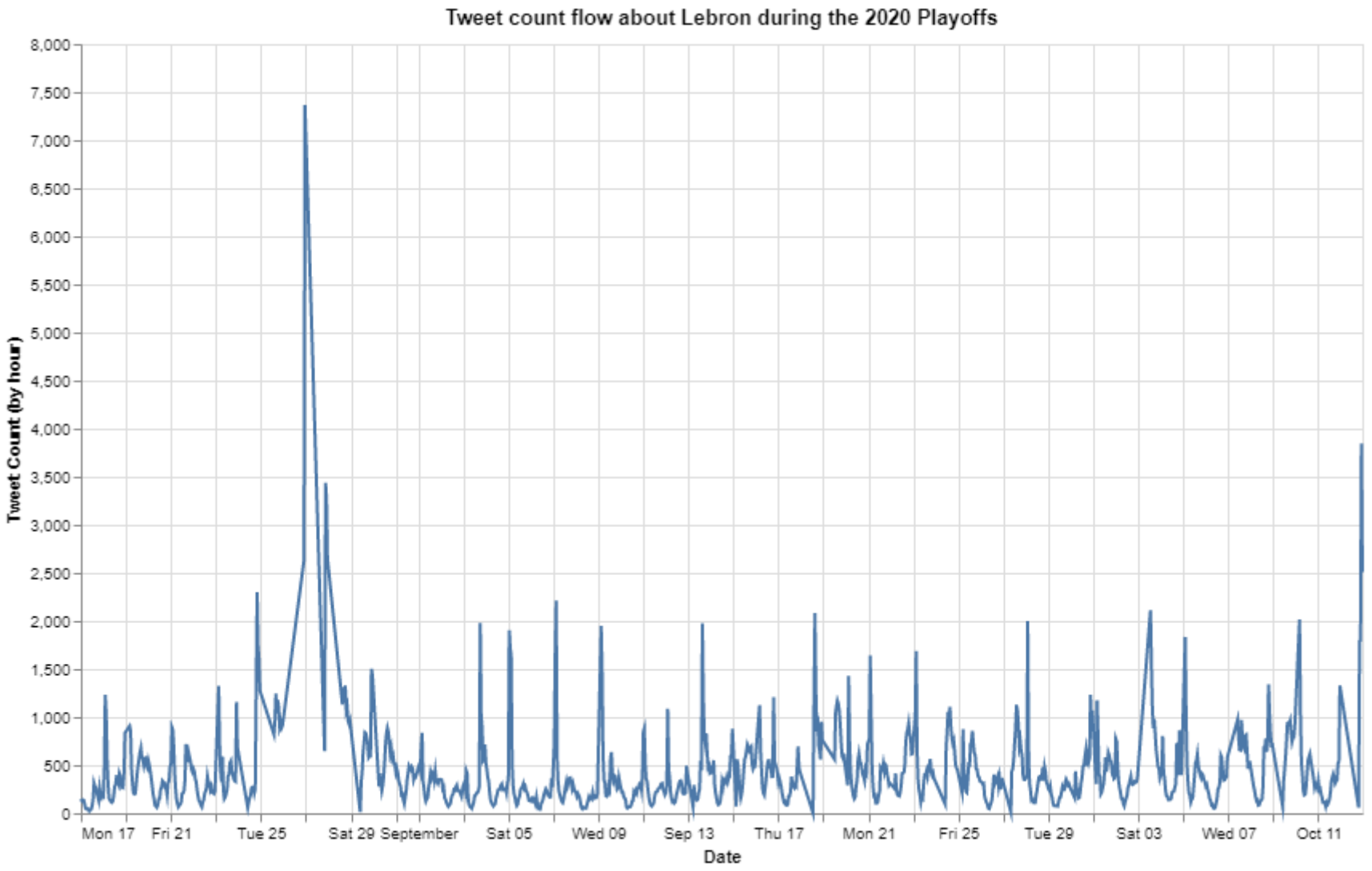
In [7]:
```
score = df.groupby(['day','hour']).agg([np.size,np.sum]).sentiment_score.reset_index()
score[['10hr_count','10hr_senti']] = score.rolling(window=10,min_periods=1).sum()[['size','sum']]
score['avg'] = score['10hr_senti'] / score['10hr_count']

score['date'] = score['day'] + ' ' + score['hour'] + ":" + '00'
score.head()
```

Out[7]:

	day	hour	size	sum	10hr_count	10hr_senti	avg	date
0	2020-08-17	00	149.0	15.605739	149.0	15.605739	0.104737	2020-08-17 00:00
1	2020-08-17	01	130.0	9.961459	279.0	25.567198	0.091639	2020-08-17 01:00
2	2020-08-17	02	140.0	9.248398	419.0	34.815596	0.083092	2020-08-17 02:00
3	2020-08-17	03	136.0	9.632750	555.0	44.448347	0.080087	2020-08-17 03:00
4	2020-08-17	04	95.0	7.217172	650.0	51.665518	0.079485	2020-08-17 04:00

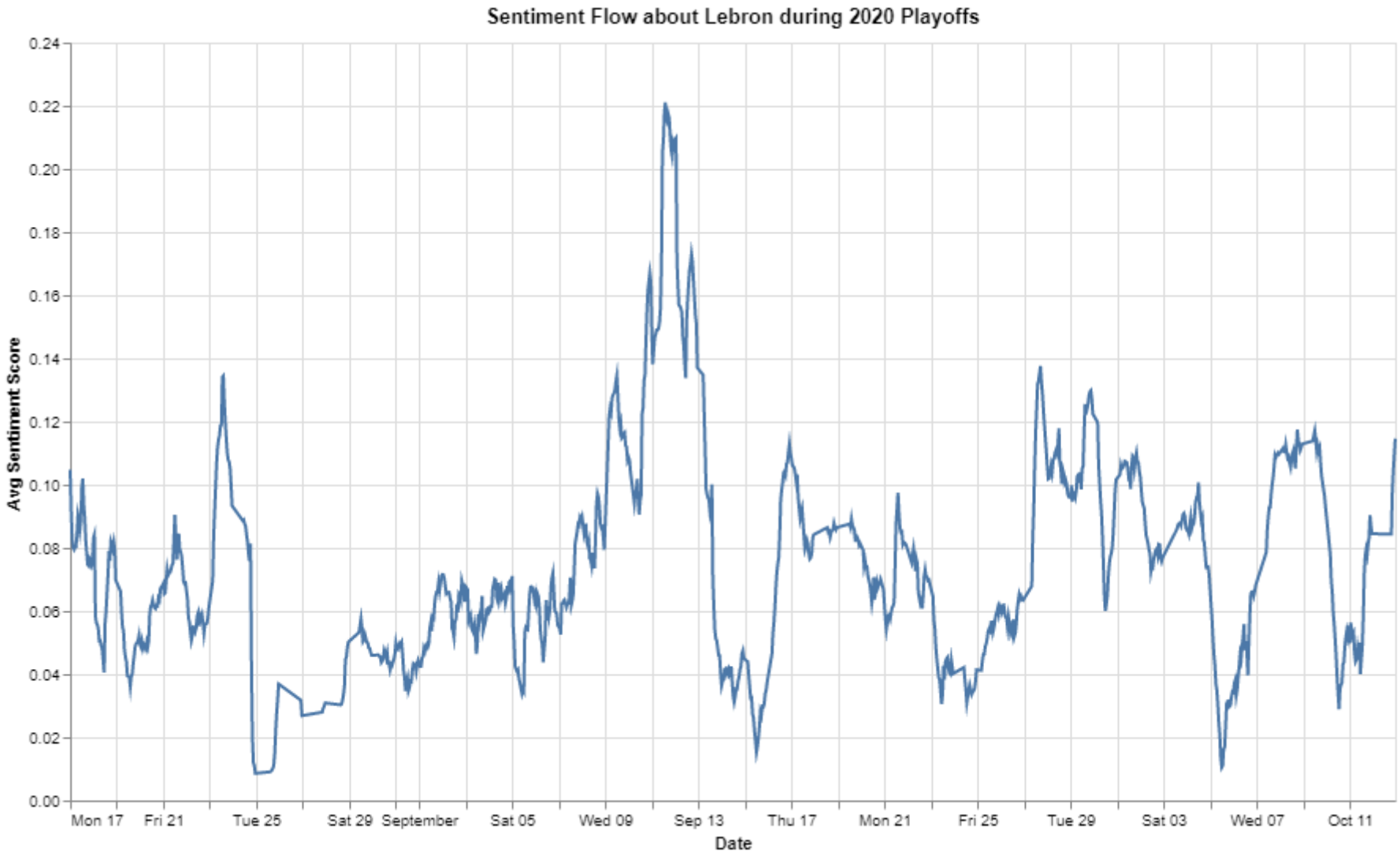
In [8]:
```
alt.Chart(score).mark_line().encode(
    x=alt.X('date:T',title='Date'),
    y=alt.Y('size:Q',title='Tweet Count (by hour)')
).properties(width=800,height=480,title='Tweet count flow about Lebron during the 2020 Playoffs')
```

Out[8]:



In [9]:
```
senti_line = alt.Chart(score).mark_line().encode(
    x=alt.X('date:T',title='Date'),
    y=alt.Y('avg:Q',title='Avg Sentiment Score')
).properties(width=840,height=480,title='Sentiment Flow about Lebron during 2020 Playoffs')
senti_line
```

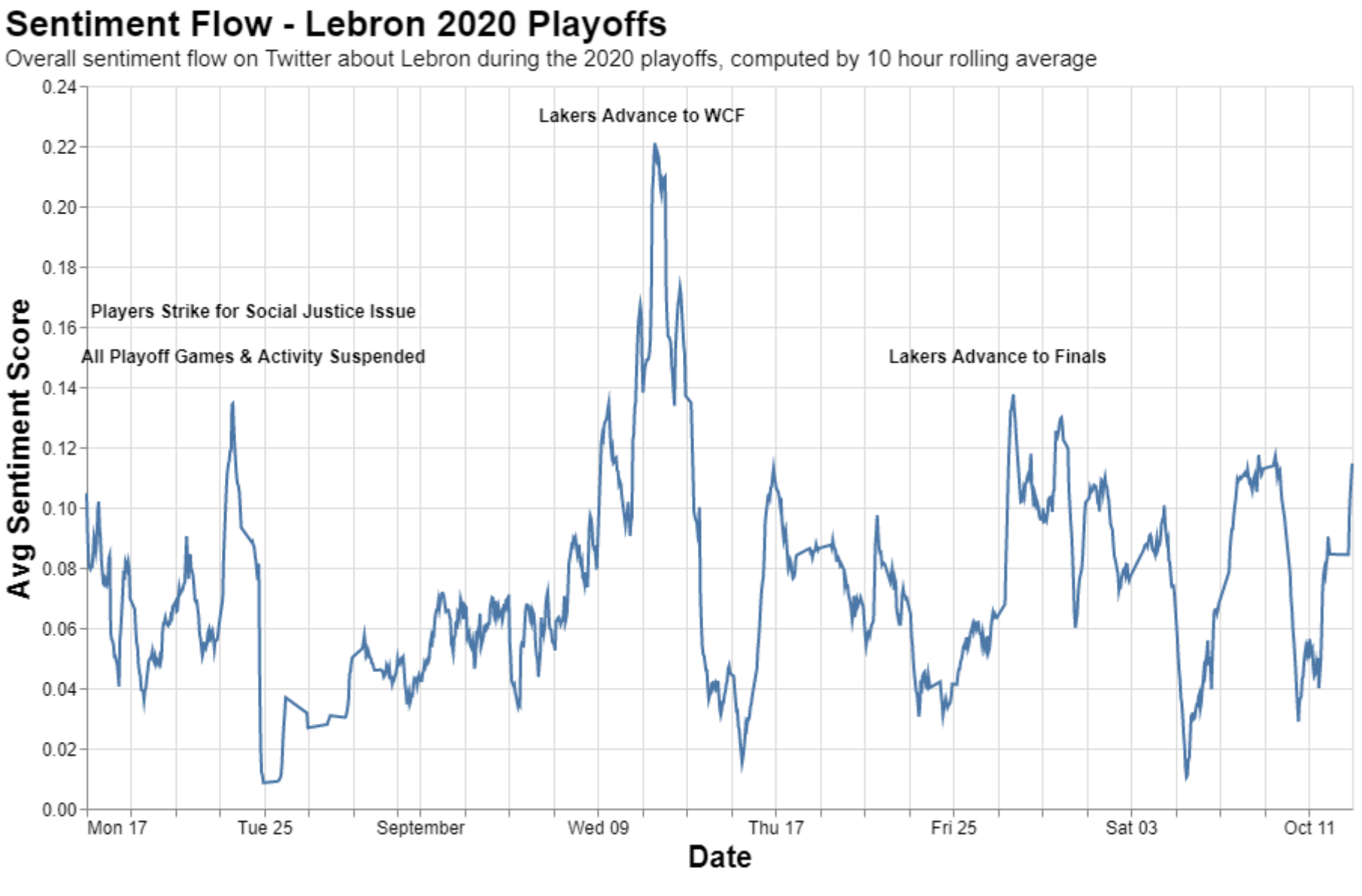
Out[9]:



In [10]:

```
annotations = [['2020-08-24 12:00:00',0.165, 'Players Strike for Social Justice Issue'],
               ['2020-08-24 12:00:00',0.15, 'All Playoff Games & Activity Suspended'],
               ['2020-09-11 00::00',0.23, 'Lakers Advance to WCF'],
               ['2020-09-27 00:00:00',0.15, 'Lakers Advance to Finals']]
a_df = pd.DataFrame(annotations, columns=['date','values','note'])
a_df

text=alt.Chart(a_df).encode(
    x=alt.X('date:T'),
    y=alt.Y('values:Q'),
    text='note').mark_text(size=12,fontWeight='bold')

(senti_line + text).properties(
    width=840,height=480,
    title={
      "text": ["Sentiment Flow - Lebron 2020 Playoffs"],
      "subtitle": ["Overall sentiment flow on Twitter about Lebron during the 2020 playoffs, computed by 10 hour rolling average"],
      "color": "black",
      "subtitleFontSize":15
    }).configure_axis(
    labelFontSize=12,
    titleFontSize=20
).configure_title(
    anchor='start',
    fontSize = 24)
```

Out[10]:


Sentiment Flow - Lebron 2020 Playoffs
Overall sentiment flow on Twitter about Lebron during the 2020 playoffs, computed by 10 hour rolling average
Lakers Advance to WCF
Players Strike for Social Justice Issue
All Playoff Games & Activity Suspended
Lakers Advance to Finals
Avg Sentiment Score
0.24
0.22
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Mon 17
Tue 25
September
Wed 09
Thu 17
Fri 25
Sat 03
Oct 11
Date

Emoji/Tag

In [11]:
```
# this return the top 50 most common items in the columns (emoji/tag/word)

def top_item(data,label):

    lst = []
    for i in data[label]:
        lst += i

    C = Counter(lst)
    top50 = C.most_common(50)
    count_df = pd.DataFrame(top50,columns = [label,'count'])

    return count_df
```

In [12]:
```
c= top_item(df, 'tags')
c

# the top 50 most frequently used tags      within the 'lebron james' tweets during the 2020 playoffs
```

Out[12]:

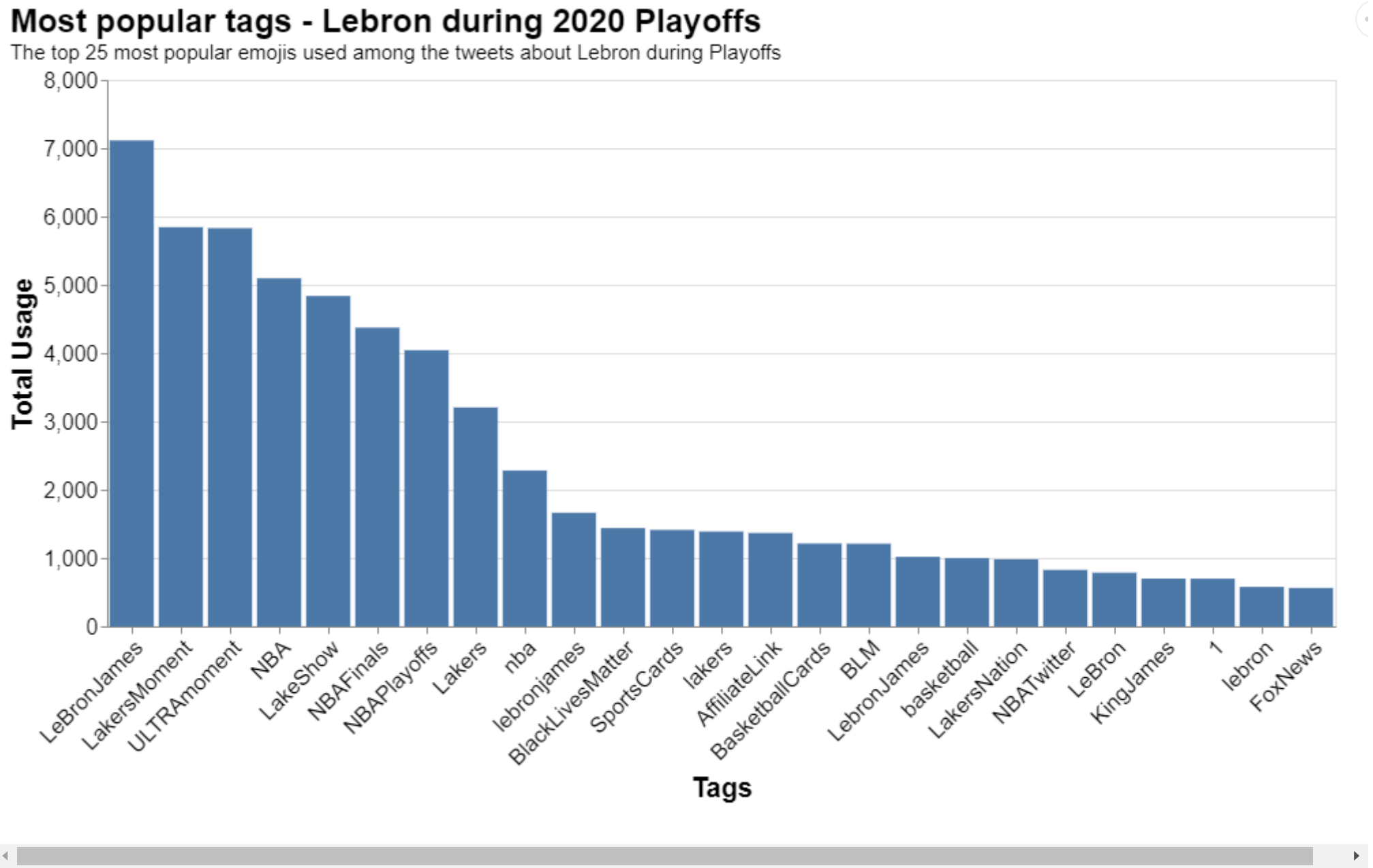
	tags	count
0	LeBronJames	7114
1	LakersMoment	5844
2	ULTRAmoment	5830
3	NBA	5096
4	LakeShow	4837
5	NBAFinals	4373
6	NBAPlayoffs	4043
7	Lakers	3205
8	nba	2282
9	lebronjames	1663
10	BlackLivesMatter	1439
11	SportsCards	1412
12	lakers	1388
13	AffiliateLink	1367
14	BasketballCards	1213
15	BLM	1210
16	LebronJames	1018
17	basketball	999
18	LakersNation	981
19	NBATwitter	826
20	LeBron	786
21	KingJames	699
22	1	698
23	lebron	578
24	FoxNews	562

	tags	count
25	MambaMentality	551
26	sports	540
27	LakerNation	504
28	GOAT	480
29	BreonnaTaylor	480
30	lakeshow	470
31	AnthonyDavis	466
32	LosAngelesLakers	462
33	MVP	453
34	WholeNewGame	434
35	HEATTwitter	419
36	NFL	388
37	MiamiHeat	376
38	SmartNews	356
39	BoycottNBA	349
40	kingjames	339
41	NBA2K21	336
42	2	334
43	KobeBryant	331
44	nbaplayoffs	330
45	Lakeshow	329
46	JacobBlake	327
47	Lebron	322
48	Trump2020	317
49	4	314

Visualize bar chart for the most popular tags

In [13]:
```
c1=c[:25]

alt.Chart(c1).mark_bar().encode(
    x=alt.X('tags',sort=['count'],title='Tags',axis=alt.Axis(labelAngle=-45)),
    y=alt.Y('count',title='Total Usage')
).properties(width=900,height=400,title={
      "text": ["Most popular tags - Lebron during 2020 Playoffs"],
      "subtitle":["The top 25 most popular emojis used among the tweets about Lebron during Playoffs"]
    }).configure_axis(
    labelFontSize=16,
    titleFontSize=20
).configure_title(
    anchor='start',
    fontSize = 24,
    subtitleFontSize = 15
)
```

Out[13]:


Most popular tags - Lebron during 2020 Playoffs
The top 25 most popular emojis used among the tweets about Lebron during Playoffs
Total Usage
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
LeBronJames
LakersMoment
ULTRAmoment
NBA
LakeShow
NBAFinals
NBAPlayoffs
Lakers
nba
lebronjames
BlackLivesMatter
SportsCards
lakers
AffiliateLink
BasketballCards
BLM
LebronJames
basketball
LakersNation
NBATwitter
LeBron
KingJames
1
lebron
FoxNews
Tags

Emoji

In [14]:
```
count = top_item(df,'emojis')
count
```

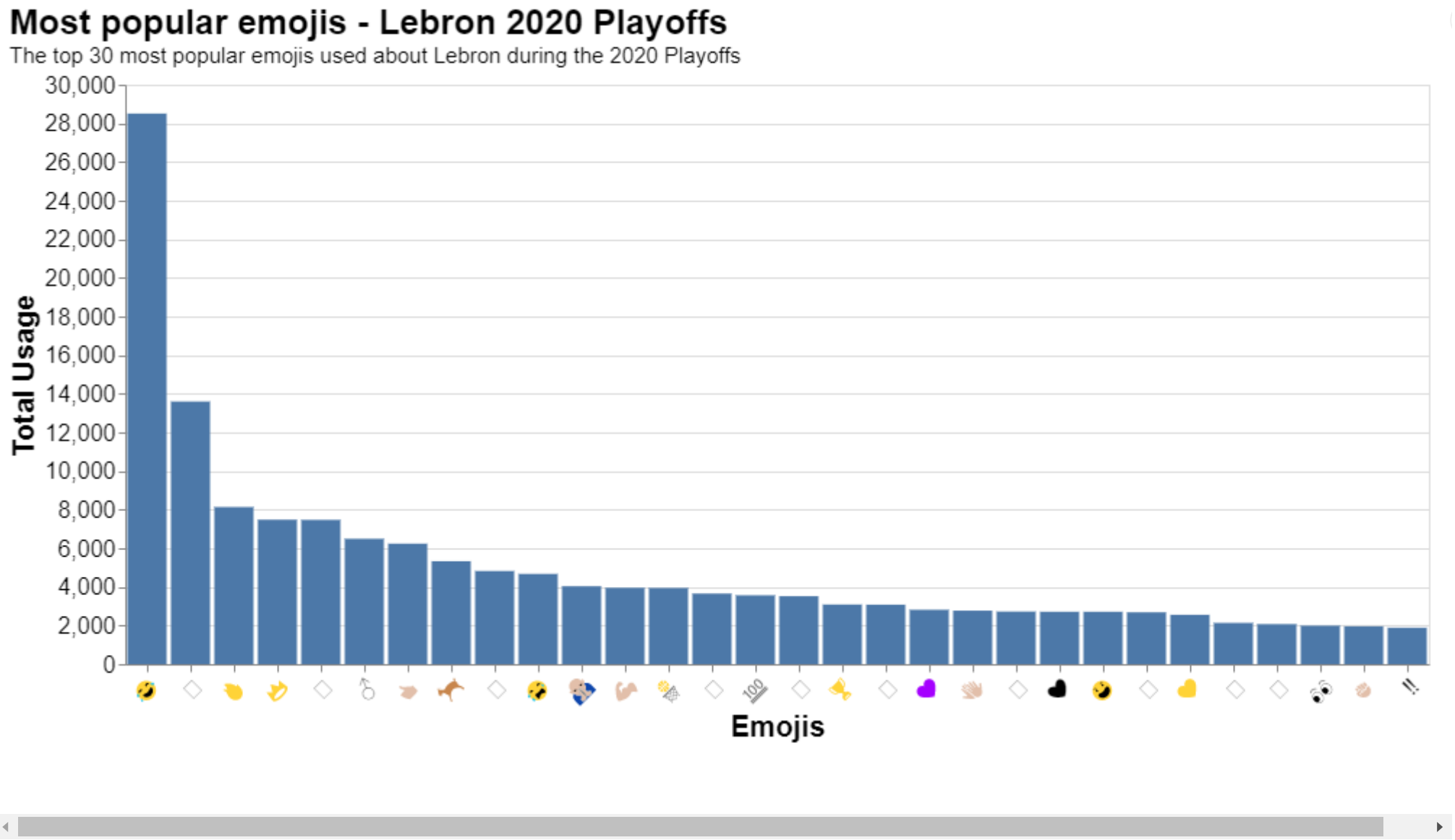
Out[14]:

	emojis	count
0	😂	28509
1		13599
2	🔥	8136
3	👑	7480
4		7471
5	♂	6492
6	👍	6240
7	🐈	5330
8		4824
9	😭	4673
10	🤦	4035
11	💪	3953
12	💐	3944
13		3655
14	💯	3565
15		3519
16	🏆	3089
17		3081
18	💓	2817
19	🤲	2771
20		2724
21	❤	2714
22	😊	2713
23		2685
24	💞	2556

	emojis	count
25		2136
26		2073
27	👀	1994
28	✊	1960
29	‼️	1887
30		1510
31	🗣️	1416
32	🤱	1343
33	😂	1275
34	💀	1241
35	😎	1115
36		1113
37	♀️	988
38		955
39	💩	905
40	🤨	862
41		816
42	😟	788
43	😆	748
44	😍	719
45	🤦	708
46	😒	694
47	😅	666
48		619
49	🤌	618

In [15]:
```
c1=count[:30]

alt.Chart(c1).mark_bar().encode(
    x=alt.X('emojis',sort=['count'],title='Emojis',axis=alt.Axis(labelAngle=-45)),
    y=alt.Y('count',title='Total Usage')
).properties(width=900,height=400,title={
      "text": ["Most popular emojis - Lebron 2020 Playoffs"],
      "subtitle":["The top 30 most popular emojis used about Lebron during the 2020 Playoffs"]
    }).configure_axis(
    labelFontSize=16,
    titleFontSize=20
).configure_title(
    anchor='start',
    fontSize = 24,
    subtitleFontSize = 15
)
```

Out[15]:



Let's dive in deep about the strike

On 2020/8/25, the Milwaukee Bucks leave the court just before their scheduled playoff game to raise awareness about social justice issue (the shooting incident in Wisconsin). Let's see what people were talking about Lebron on Twitter during that period of players' strike.

In [16]:
```
df['Date'] = pd.to_datetime(df['date'])
mask = (df['Date'] > '2020-08-25 00:00') & (df['Date'] < '2020-08-28 23:59')
strike = df.loc[mask].sort_values('Date')
strike = strike.reset_index()
strike.drop(columns=['index','Date'],inplace=True)

strike.head()
```

Out[16]:

	id	date	text	tags	emojis	clean_text	words	sentiment_score	day	hour	10min	min	POS	pos	neu
0	1298260501088989184	2020-08-25 14:06:38+00:00	@CNN Hey @KingJames what about the people in H...	[]	[]	hey what about the people in hong kong arent...	[hey, people, hong, kong, arent, scared, peopl...	0.242857	2020-08-25	14	00	06	0	0	1
1	1298260505933434886	2020-08-25 14:06:39+00:00	@KingJames Lebron said, in reference to Jacob ...	[]	[, , ♂]	lebron said in reference to jacob blake and k...	[lebron, said, reference, jacob, blake, kenosh...	0.135714	2020-08-25	14	00	06	0	0	1
2	1298260506747142145	2020-08-25 14:06:39+00:00	Whitlock: LeBron James Is A Bigot https://t.co...	[]	[]	whitlock lebron james is a bigot	[whitlock, lebron, james, bigot]	0.000000	2020-08-25	14	00	06	0	0	1
3	1298260517559996416	2020-08-25 14:06:42+00:00	@KingJames \nKingJames you should sue Lazy #Ja...	[Jasonwhitlock, LeBronJames, LakeShow, Lakerna...	[]	kingjames you should sue lazy for repeatedly...	[kingjames, sue, lazy, repeatedly, using, name...	-0.375000	2020-08-25	14	00	06	0	0	0
4	1298260528217780232	2020-08-25 14:06:45+00:00	@KingJames if you want to make change, but bod...	[]	[]	if you want to make change but body cams for ...	[want, make, change, body, cams, every, police...	0.000000	2020-08-25	14	00	06	0	0	1

In [17]:
```
c=top_item(strike,'tags')
c
```

Out[17]:

	tags	count
0	LeBronJames	379
1	BlackLivesMatter	281
2	NBA	244
3	BLM	159
4	JacobBlake	152
5	NBABoycott	125
6	NBAPlayoffs	120
7	nba	105
8	AffiliateLink	75
9	SportsCards	75
10	BoycottNBA	66
11	BasketballCards	60
12	LebronJames	52
13	Lakers	48
14	lebronjames	48
15	blm	38
16	MAGA	38
17	China	37
18	1	37
19	BidenHarris2020	37
20	LakeShow	36
21	MambaMentality	36
22	KingJames	34
23	JacobBlakeshooting	32
24	AllLivesMatter	31
25	Trump2020	31

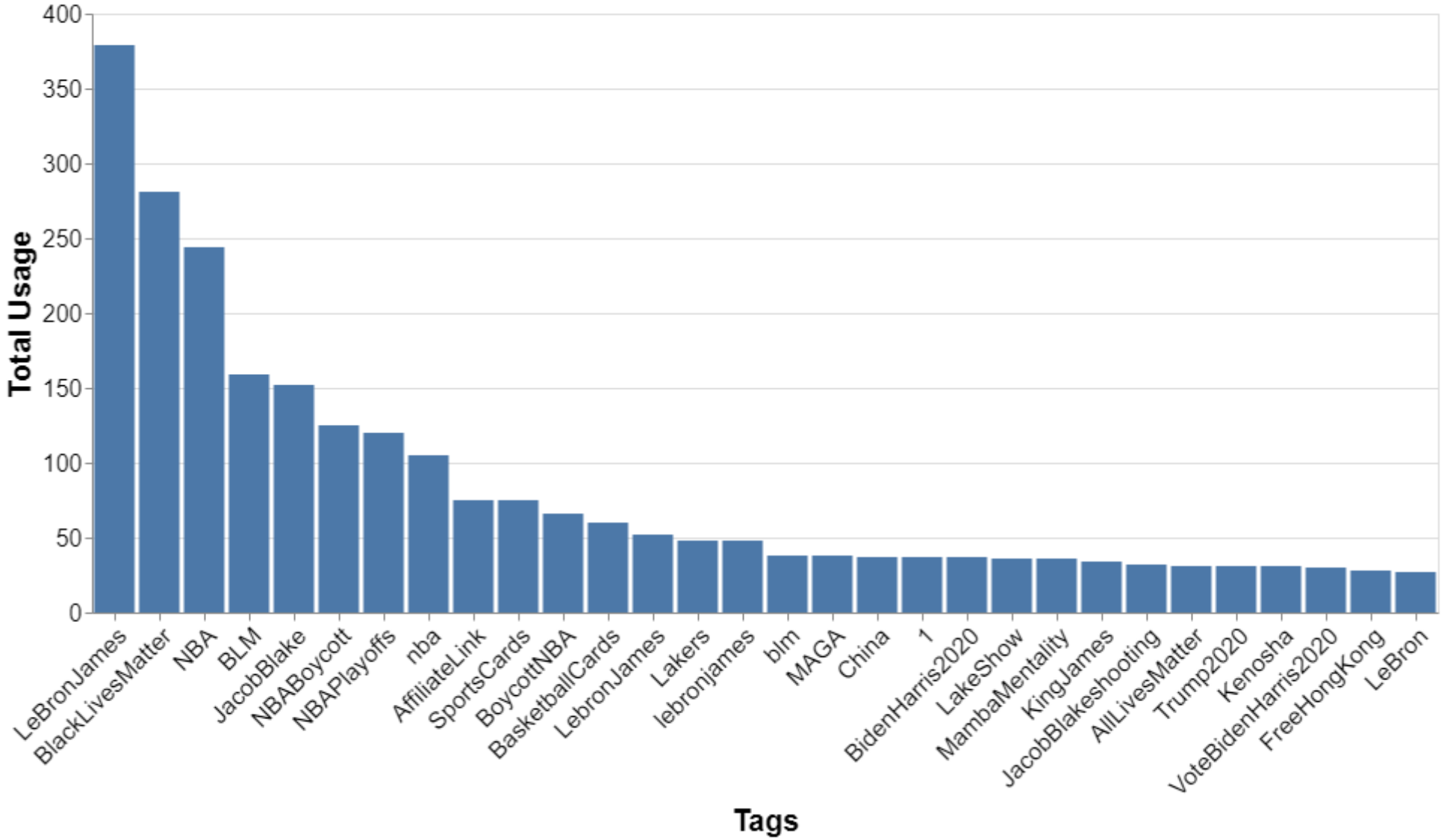
	tags	count
26	Kenosha	31
27	VoteBidenHarris2020	30
28	FreeHongKong	28
29	LeBron	27
30	EnoughIsEnough	27
31	nbastrike	27
32	basketball	26
33	blacklivesmatter	25
34	FoxNews	24
35	Lebron	24
36	BlacklivesStillMatter	24
37	LosAngelesLakers	23
38	BreonnaTaylor	23
39	NFL	22
40	VoteBidenHarrisToSaveAmerica	22
41	BlueLivesMatter	21
42	lakers	21
43	lebron	21
44	Vote	21
45	Trump	21
46	SmartNews	20
47	JusticeForJacobBlake	20
48	vote	20
49	CannonHinnant	19

In [18]:
```
c1=c[:30]

alt.Chart(c1).mark_bar().encode(
    x=alt.X('tags',sort=['count'],title='Tags',axis=alt.Axis(labelAngle=-45)),
    y=alt.Y('count',title='Total Usage')
).properties(width=900,height=400,title={
      "text": ["Most frequent tags during Strike"],
      "subtitle":["The top 30 most used tags among the tweets about Lebron during NBA Players' Strike"]
    }).configure_axis(
    labelFontSize=16,
    titleFontSize=20
).configure_title(
    anchor='start',
    fontSize = 24,
    subtitleFontSize = 15
)
```

Out[18]:

Most frequent tags during Strike

The top 30 most used tags among the tweets about Lebron during NBA Players' Strike



In [ ]: