In [1]:
```python
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
```

```
[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 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, and check if duplicate/missing value exist

In [3]:
```
df = pd.read_csv('Project Data/Kershaw 2020 WS.csv')
df['id'].duplicated(keep='last').sum()
```

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

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

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

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 [5]:

```
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[5]:

	id	date	text	tags	emojis	clean_text	words	sentiment_score	day	hour	10min	min	POS	pos	neu
0	1317978642094305282	2020-10-18 23:59:29+00:00	My ideal rotation for today:\n\nMay (3 innings...	[WorldSeries, GoDodgers]	[]	my ideal rotation for todaymay innings gonsol...	[ideal, rotation, todaymay, innings, gonsolin,...	0.900000	2020-10-18	23	50	59	1	0	0
1	1317978495197171713	2020-10-18 23:58:54+00:00	Imagine if it comes down to this, top of the n...	[]	[]	imagine if it comes down to this top of the ni...	[imagine, comes, top, ninth, save, situation, ...	0.114815	2020-10-18	23	50	58	0	0	1
2	1317977821613088775	2020-10-18 23:56:14+00:00	@grcate @Buccaneers @RaysBaseball I'll probabl...	[]	[]	ill probably root for lad tonight no real r...	[ill, probably, root, lad, tonight, real, reas...	-0.241667	2020-10-18	23	50	56	0	0	1
3	1317977505492537346	2020-10-18 23:54:58+00:00	#Dodgers want to win tonight they have to stay...	[Dodgers, NLCS]	[]	want to win tonight they have to stay away fr...	[want, win, tonight, stay, away, kershaw, ever...	0.200000	2020-10-18	23	50	54	0	0	1
4	1317977421010948097	2020-10-18 23:54:38+00:00	Listen Wouldn't It Be Good by Nik Kershaw on h...	[]	[]	listen wouldnt it be good by nik kershaw on	[listen, wouldnt, good, nik, kershaw]	0.700000	2020-10-18	23	50	54	1	0	0

See the overall flow of tweet & sentiment

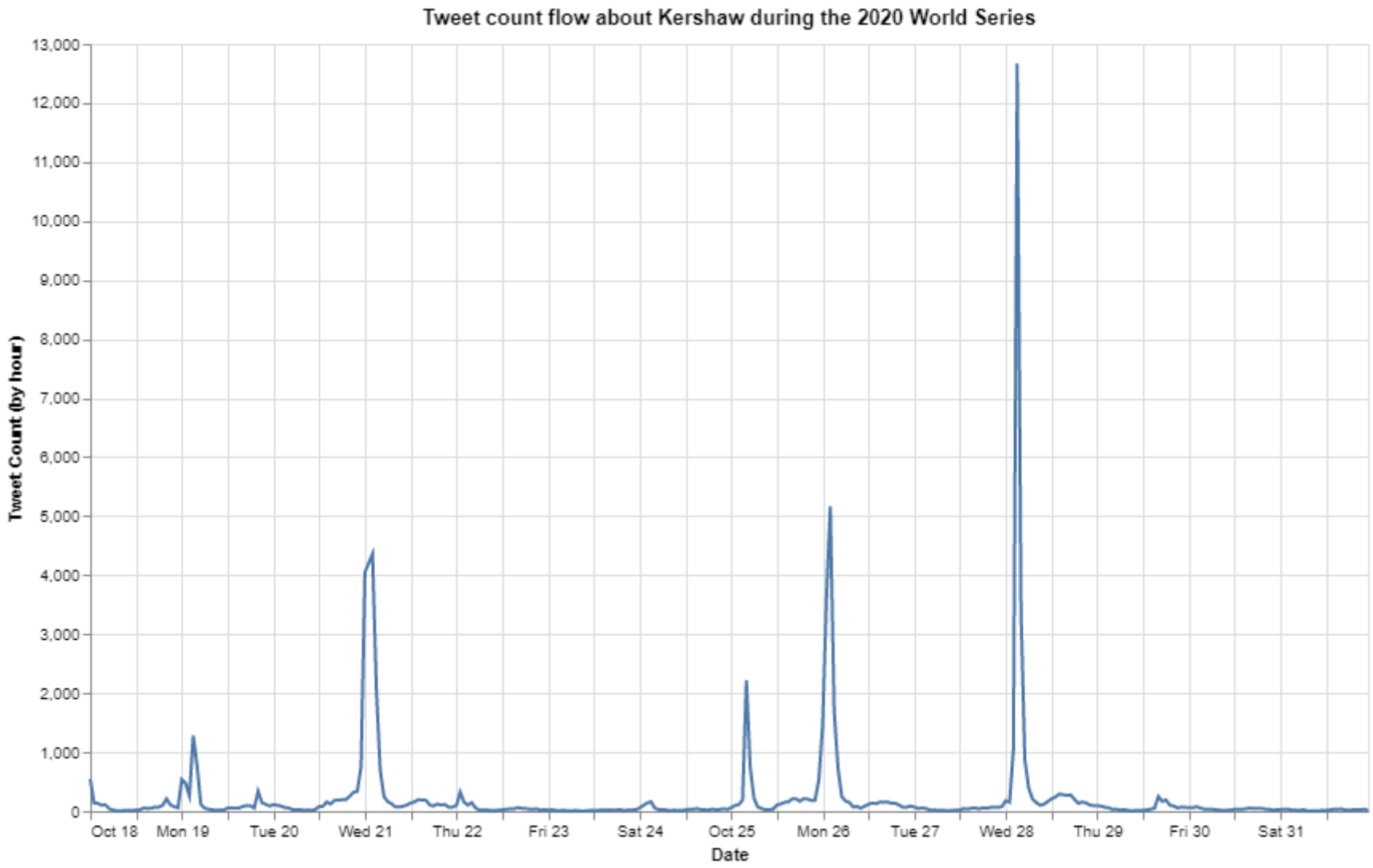
In [6]:
```
score = df.groupby(['day','hour']).agg([np.sum,np.size]).sentiment_score
score = score.reset_index()
score['date'] = score['day'] + ' ' + score['hour'] + ':00'
score[['12hr_count','12hr_sum']] = score.rolling(window=12,min_periods=1).sum()[['size','sum']]

score['12hr_avg'] = score['12hr_sum'] / score['12hr_count']
score.head()
```

Out[6]:

	day	hour	sum	size	date	12hr_count	12hr_sum	12hr_avg
0	2020-10-18	00	7.868114	544.0	2020-10-18 00:00	544.0	7.868114	0.014463
1	2020-10-18	01	7.687393	141.0	2020-10-18 01:00	685.0	15.555507	0.022709
2	2020-10-18	02	-0.561022	131.0	2020-10-18 02:00	816.0	14.994485	0.018376
3	2020-10-18	03	10.825824	102.0	2020-10-18 03:00	918.0	25.820310	0.028127
4	2020-10-18	04	15.502814	108.0	2020-10-18 04:00	1026.0	41.323124	0.040276

In [7]:
```
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 Kershaw during the 2020 World Series')
```

Out[7]:



Plot the sentiment flow

In [8]:
```
alt.Chart(score).mark_line().encode(
    x=alt.X('date:T',title='Date'),
    y=alt.Y('12hr_avg:Q',title='Sentiment Score')
).properties(
    width=840,height=480,
    title={
      "text": ["Sentiment Flow - Kershaw 2020 World Series"],
      "subtitle": ["Overall sentiment flow on Twitter about Kershaw during the 2020 WS, computed by 12-hour rolling average"],
      "color": "black",
      "subtitleFontSize":15
    }).configure_axis(
    labelFontSize=12,
    titleFontSize=20
).configure_title(
    anchor='start',
    fontSize = 24)
```

Out[8]:


Sentiment Flow - Kershaw 2020 World Series
Overall sentiment flow on Twitter about Kershaw during the 2020 WS, computed by 12-hour rolling average
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
Oct 18
Mon 19
Tue 20
Wed 21
Thu 22
Fri 23
Sat 24
Oct 25
Mon 26
Tue 27
Wed 28
Thu 29
Fri 30
Sat 31
Date

Emoji/tag

In [9]:
```
# 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 [10]:
```
c=top_item(df,'tags')
c
```

Out[10]:

	tags	count
0	WorldSeries	5950
1	Dodgers	5107
2	LATogether	733
3	Kershaw	691
4	RaysUp	601
5	dodgers	558
6	Rays	460
7	MLB	426
8	WorldSeries2020	323
9	kershaw	280
10	worldseries	245
11	Postseason	235
12	LADvsTB	179
13	postseason	142
14	mlb	129
15	MLBPlayoffs	118
16	LADodgers	100
17	22	92
18	DodgersNation	86
19	TBvsLAD	84
20	GoDodgers	82
21	TheBachelorette	81
22	LetsGoDodgers	78
23	DODGERS	78
24	NLCS	72
25	Mookie	66

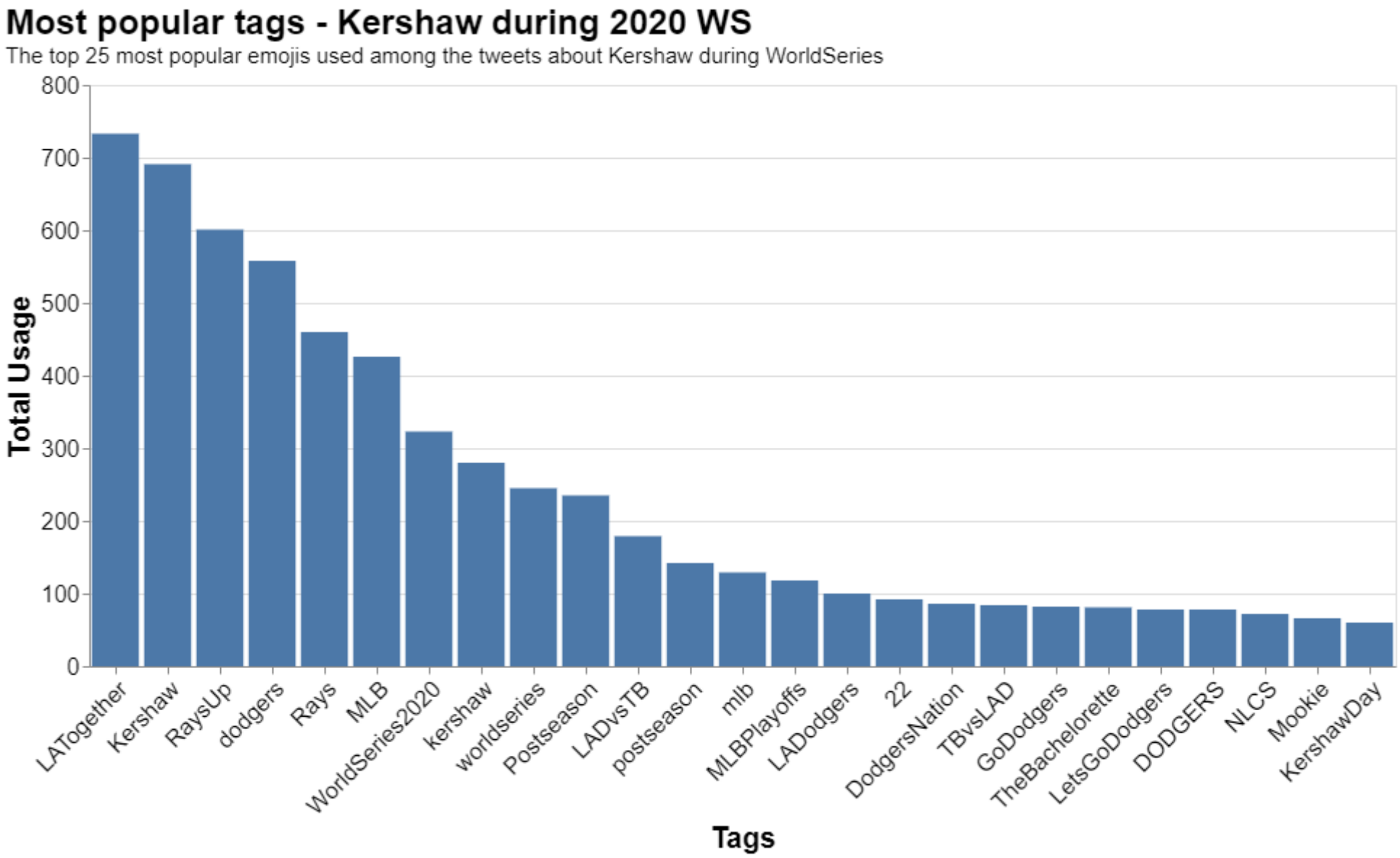
	tags	count
26	KershawDay	60
27	1	56
28	baseball	55
29	ClaytonKershaw	54
30	LA_Dodgers	50
31	GamblingTwitter	49
32	ITFDB	49
33	ATLvsLAD	49
34	Sports	48
35	TampaBayRays	40
36	NowPlaying	39
37	sports	39
38	iscore	39
39	BeatLA	38
40	Baseball	38
41	rays	37
42	LosAngelesDodgers	36
43	Bitcoin	35
44	Sportsbook	35
45	SerieMundial	33
46	WorldSeriesChamps	33
47	Worlds2020	31
48	MLBPostseason	30
49	DaveRoberts	29

Visualize bar chart for the most frequent tags

In [11]:

```
c1=c[2:27]

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 - Kershaw during 2020 WS"],
      "subtitle":["The top 25 most popular emojis used among the tweets about Kershaw during WorldSeries"]
    }).configure_axis(
    labelFontSize=16,
    titleFontSize=20
).configure_title(
    anchor='start',
    fontSize = 24,
    subtitleFontSize = 15
)
```

Out[11]:


Most popular tags - Kershaw during 2020 WS
The top 25 most popular emojis used among the tweets about Kershaw during WorldSeries
Total Usage
800
700
600
500
400
300
200
100
0
LATogether
Kershaw
RaysUp
dodgers
Rays
MLB
WorldSeries2020
kershaw
worldseries
Postseason
LADvsTB
postseason
mlb
MLBPlayoffs
LADodgers
22
DodgersNation
TBvsLAD
GoDodgers
TheBachelorette
LetsGoDodgers
DODGERS
NLCS
Mookie
KershawDay
Tags

Emoji

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

Out[12]:

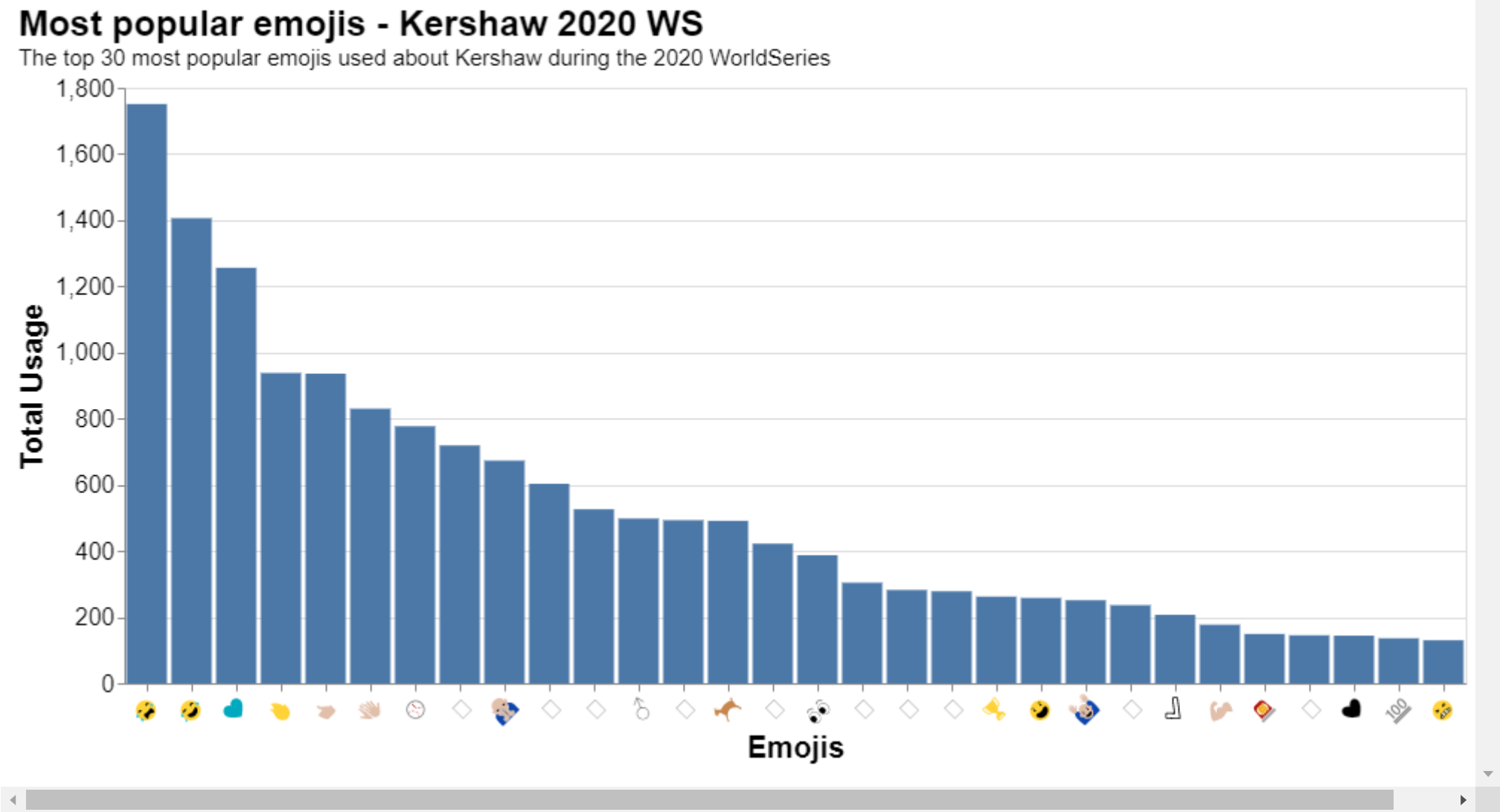
	emojis	count
0	😷	1751
1	😂	1406
2	💖	1256
3	🔥	938
4	👍	936
5	🙌	830
6	😍	777
7		719
8	🙈	673
9		603
10		526
11	♂	498
12		493
13	🐈	491
14		422
15	👀	387
16		304
17		282
18		278
19	🏆	262
20	😊	258
21	🐰	251
22		236
23	✅	207
24	💪	177

	emojis	count
25	🙊	149
26		145
27	❤	144
28	💯	136
29	😆	130
30	💍	129
31	💰	126
32	😍	117
33		115
34	😜	115
35		113
36	😘	112
37	♀	107
38	😟	103
39	😢	99
40		92
41		92
42		91
43	💀	91
44	😕	90
45	😊	88
46		79
47	🍥	77
48	👇	74
49		74

In [13]:

```
c1=c[: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 - Kershaw 2020 WS"],
      "subtitle":["The top 30 most popular emojis used about Kershaw during the 2020 WorldSeries"]
    }).configure_axis(
    labelFontSize=16,
    titleFontSize=20
).configure_title(
    anchor='start',
    fontSize = 24,
    subtitleFontSize = 15
)
```

Out[13]:



Specifically target game6 for in-depth analysis

After making the postseason for 7 straight years but lost every time (lost back-to-back in World Series in 2017,18), the best left-handed starting pitcher of the generation finally get the championship ring in 2020. Let's see what Twitter have to say about this.

In [14]:
```
df['Date'] = pd.to_datetime(df['date'])
mask = (df['Date'] > '2020-10-28 00:00') & (df['Date'] < '2020-10-28 05:59')
game6 = df.loc[mask].sort_values('Date')
game6 = game6.reset_index()
game6.drop(columns=['index','Date'],inplace=True)

game6.head()
```

Out[14]:

	id	date	text	tags	emojis	clean_text	words	sentiment_score	day	hour	10min	min	POS	pos	neu
0	1321240296705777667	2020-10-28 00:00:08+00:00	@Tugboat_35 I want Seager honestly but would b...	[]	[]	i want seager honestly but would be surprised...	[want, seager, honestly, would, surprised, ker...	0.266667	2020-10-28	00	00	00	0	1	0
1	1321240297259421696	2020-10-28 00:00:08+00:00	It's Clayton Kershaw's time. #WorldSeries htt...	[WorldSeries]	[]	its clayton kershaws time	[clayton, kershaws, time]	0.000000	2020-10-28	00	00	00	0	0	1
2	1321240405589938178	2020-10-28 00:00:34+00:00	get Kershaw his ring	[]	[]	get kershaw his ring	[get, kershaw, ring]	0.000000	2020-10-28	00	00	00	0	0	1
3	1321240462951284736	2020-10-28 00:00:48+00:00	I just want Kershaw to have a ring already	[]	[]	i just want kershaw to have a ring already	[want, kershaw, ring, already]	0.000000	2020-10-28	00	00	00	0	0	1
4	1321240599249539074	2020-10-28 00:01:20+00:00	@VeniceMase Kershaw for 1st out. Jansen for th...	[]	[😊]	kershaw for st out jansen for the nd out and ...	[kershaw, st, jansen, nd, broxton, last]	0.000000	2020-10-28	00	00	01	0	0	1

In [15]:
```
c= top_item(game6,'emojis')
c
```

Out[15]:

	emojis	count
0	😂	1177
1	💝	553
2		299
3	🐈‍⬛	249
4	👈	238
5	🖐	227
6	😄	197
7		168
8	🔥	143
9		138
10	🏆	138
11	😒	118
12		113
13	🐻‍❄️	112
14	🙈	111
15		94
16	💍	87
17	❤️	56
18		54
19	☺️	53
20		51
21	🏉	50
22	👀	50
23	☹️	49
24	♂	43

	emojis	count
25	🎉	43
26	💯	40
27		36
28	💪	36
29		35
30		31
31	‼	31
32	😄	31
33	😍	30
34	😕	29
35		24
36		23
37		21
38	✅	21
39	♀	20
40	👑	20
41	🍥	19
42	😆	18
43		18
44		17
45	😂	16
46	🤔	15
47		15
48	😎	13
49		13

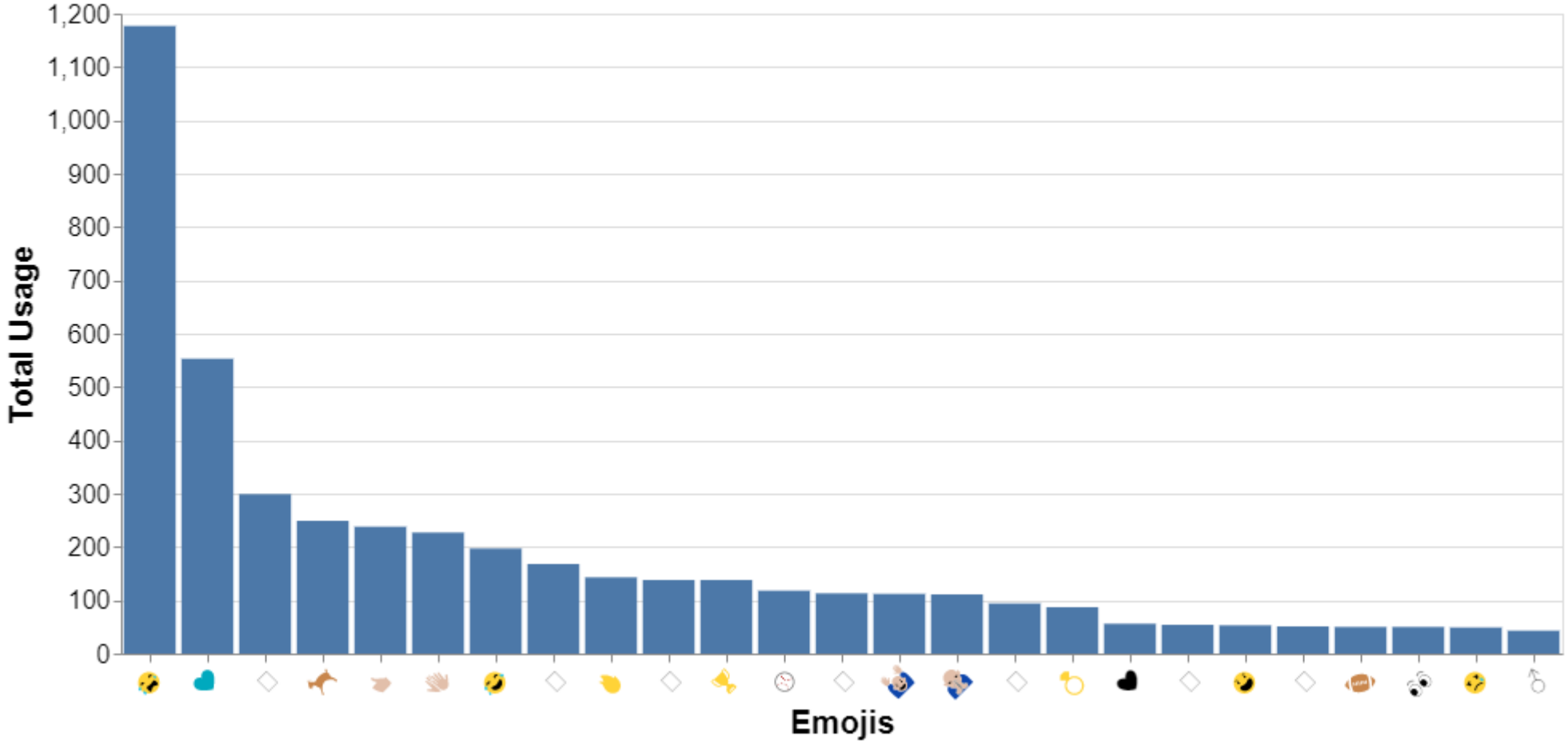
In [16]:

```
c1=c[:25]

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 - Kershaw Get a Ring"],
      "subtitle":["The top 25 most popular emojis used about Kershaw after winning the championship ring"]
    }).configure_axis(
    labelFontSize=16,
    titleFontSize=20
).configure_title(
    anchor='start',
    fontSize = 24,
    subtitleFontSize = 15
)
```

Out[16]:

Most popular emojis - Kershaw Get a Ring

The top 25 most popular emojis used about Kershaw after winning the championship ring



In [17]:
```
c=top_item(game6,'tags')
c
```

Out[17]:

	tags	count
0	WorldSeries	1502
1	Dodgers	906
2	dodgers	166
3	LATogether	122
4	Kershaw	82
5	kershaw	65
6	WorldSeries2020	58
7	DodgersNation	56
8	MLB	45
9	worldseries	28
10	TBvsLAD	27
11	Rays	26
12	DODGERS	26
13	RaysUp	25
14	LADodgers	24
15	Postseason	18
16	GoDodgers	16
17	Champions	14
18	postseason	12
19	MVP	10
20	mlb	10
21	dodgerswin	10
22	GOAT	10
23	Mookie	10
24	WorldSeriesChampions	9
25	HOF	8

	tags	count
26	2020	8
27	22	8
28	WorldSeriesChamps	8
29	ClaytonKershaw	7
30	BleedBlue	7
31	SerieMundial	7
32	mookiebetts	7
33	SFGiants	7
34	WORLDSERIES	7
35	Lakers	7
36	Champs	6
37	champions	6
38	LA	6
39	LosAngeles	6
40	latogether	6
41	ITFDB	5
42	WeLoveLA	5
43	LosAngelesDodgers	5
44	MLBPlayoffs	5
45	daveroberts	5
46	MLBnaESPN	5
47	LFGM	5
48	baseball	4
49	LADvsTB	4

In [ ]: