In [9]:
```python
import os
import pandas as pd
import cufflinks as cf
import matplotlib.pyplot as plt
cf.go_offline()
cf.set_config_file(theme='white')
# read one file
# data=pd.read_csv('../data/10line_of_part-00000',sep='\t',header=None)
# read all files
```

In [3]:
```python
PITCH_NAMES = {0: 'F', 1: 'C', 2: 'G', 3: 'D', 4: 'A', 5: 'E', 6: 'B'}

def tpc2name(tpc):
    """Return name of a tonal pitch class where
       0 = C, -1 = F, -2 = Bb, 1 = G etc.
    """
    try:
        tpc = int(tpc) + 1 # to make the lowest name F = 0 instead of -1
    except:
        raise ValueError(f"'{tpc}' is not a TPC.")

    acc = abs(tpc // 7) * 'b' if tpc < 0 else tpc // 7 * '#'
    return PITCH_NAMES[tpc % 7] + acc
```

In [4]:
```python
def read_folder(file_dir):
    all_file_list=os.listdir(file_dir)
    for single_file in all_file_list:
        # 逐个读取
        single_data_frame=pd.read_csv(os.path.join(file_dir,single_file),sep='\t',header=None)
        if single_file ==all_file_list[0]:
            all_data_frame=single_data_frame
        else:  #进行concat操作
             all_data_frame=pd.concat([all_data_frame, single_data_frame],ignore_index=True)
    all_data_frame=all_data_frame[1:]
    all_data_frame.columns=['mc','mn','mc_onset','mn_onset','timesig','staff','voice','duration','gracenote','nominal_duration','scalar','tied','tpc','midi','volta','chord_i
    return all_data_frame
```

According to List of works by Ludwig van Beethoven (https://imslp.org/wiki/List_of_works_by_Ludwig_van_Beethoven), the composing date (year) is added to the dataset.

DESCRIPTION OF THE DATA:

- id: which piece
- mn: measure number (=bar number)
- onset: distance from the measure's beginning (expressed as fraction of a whole note, so that 1/4 = quarter note)
- duration: duration (expressed as fraction of a whole note, so that 1/4 = quarter note)
- tpc: tonal pitch class, expressed on the line of fifth with (any) C = 0 (see below)
- midi: pitch expressed as piano key with 60 = C4
- keysig: Key signature of the score, 3 = 3 sharps; -3 = 3 flats
- timesig: Time signature of the score (question: why isn't this column conserved as a string, instead of parsing the fraction?)

TPC (extends in both directions)

tpc	pitch	accidental
-9	B	♭♭
-8	F	♭
-7	C	♭
-6	G	♭
-5	D	♭
-4	A	♭
-3	E	♭
-2	B	♭
-1	F	
0	C	
1	G	
2	D	
3	A	
4	E	
5	B	
6	F	♯
7	C	♯
8	G	♯
9	D	♯
10	A	♯
11	E	♯
12	B	♯
13	F	♯♯

In [5]:
```python
nodes_1801=read_folder("notes/1801")
nodes_1806=read_folder("notes/1806")
nodes_1809=read_folder("notes/1809")
nodes_1810=read_folder("notes/1810")
nodes_1825=read_folder("notes/1825")
nodes_1826=read_folder("notes/1826")
```

In [6]:
```python
nodes_1801.head()
```

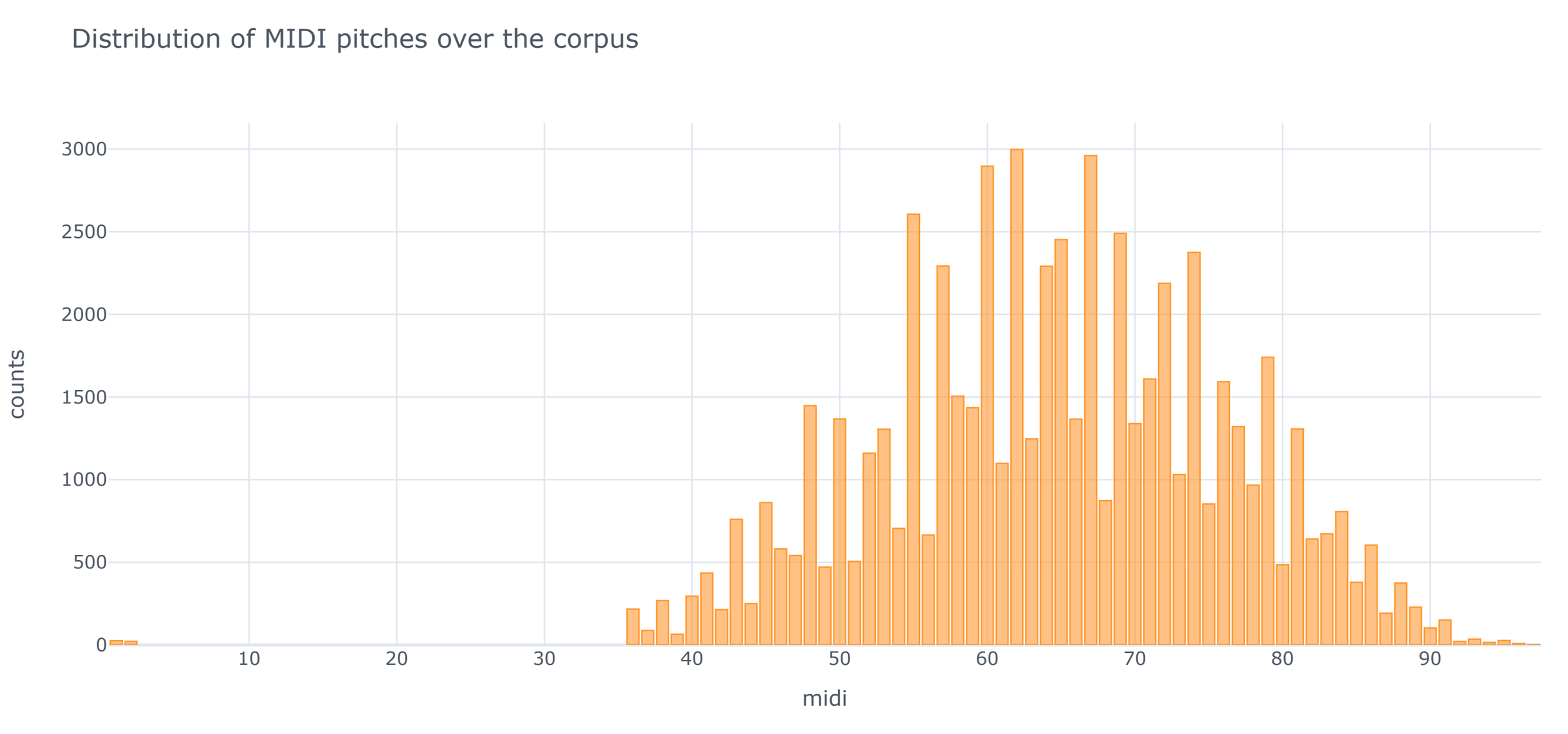
Out[6]:

	mc	mn	mc_onset	mn_onset	timesig	staff	voice	duration	gracenote	nominal_duration	scalar	tied	tpc	midi	volta	chord_id
1	1	1	0	1/2	3/4	1	1	1/4	NaN	1/4	1	NaN	7	73	NaN	0
2	2	2	0	0	3/4	1	1	3/4	NaN	1/2	3/2	NaN	4	76	NaN	1
3	2	2	1/4	1/4	3/4	2	1	1/4	NaN	1/4	1	NaN	7	73	NaN	2
4	2	2	1/2	1/2	3/4	2	1	1/4	NaN	1/4	1	NaN	3	69	NaN	3
5	3	3	0	0	3/4	1	1	3/4	NaN	1/2	3/2	NaN	5	71	NaN	4

Counting notes

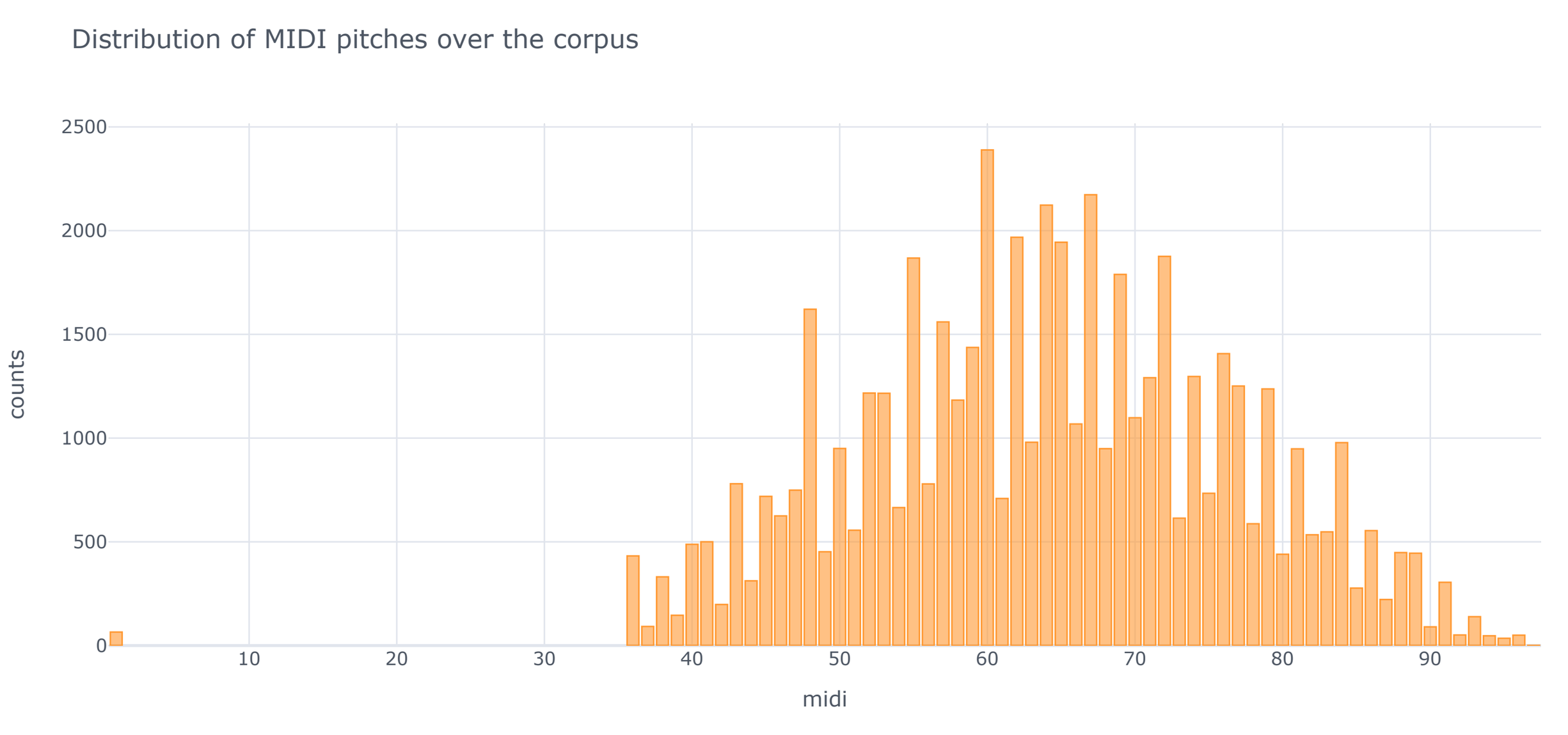
In [12]:
```python
# Distribution of MIDI pitches over the corpus in 1801
nodes_1801.midi.value_counts().iplot('bar', title='Distribution of MIDI pitches over the corpus', xTitle='midi', yTitle='counts')

# nodes_1801.midi.value_counts().plot(kind='bar');
# plt.xticks(rotation=90)
# plt.show()
```



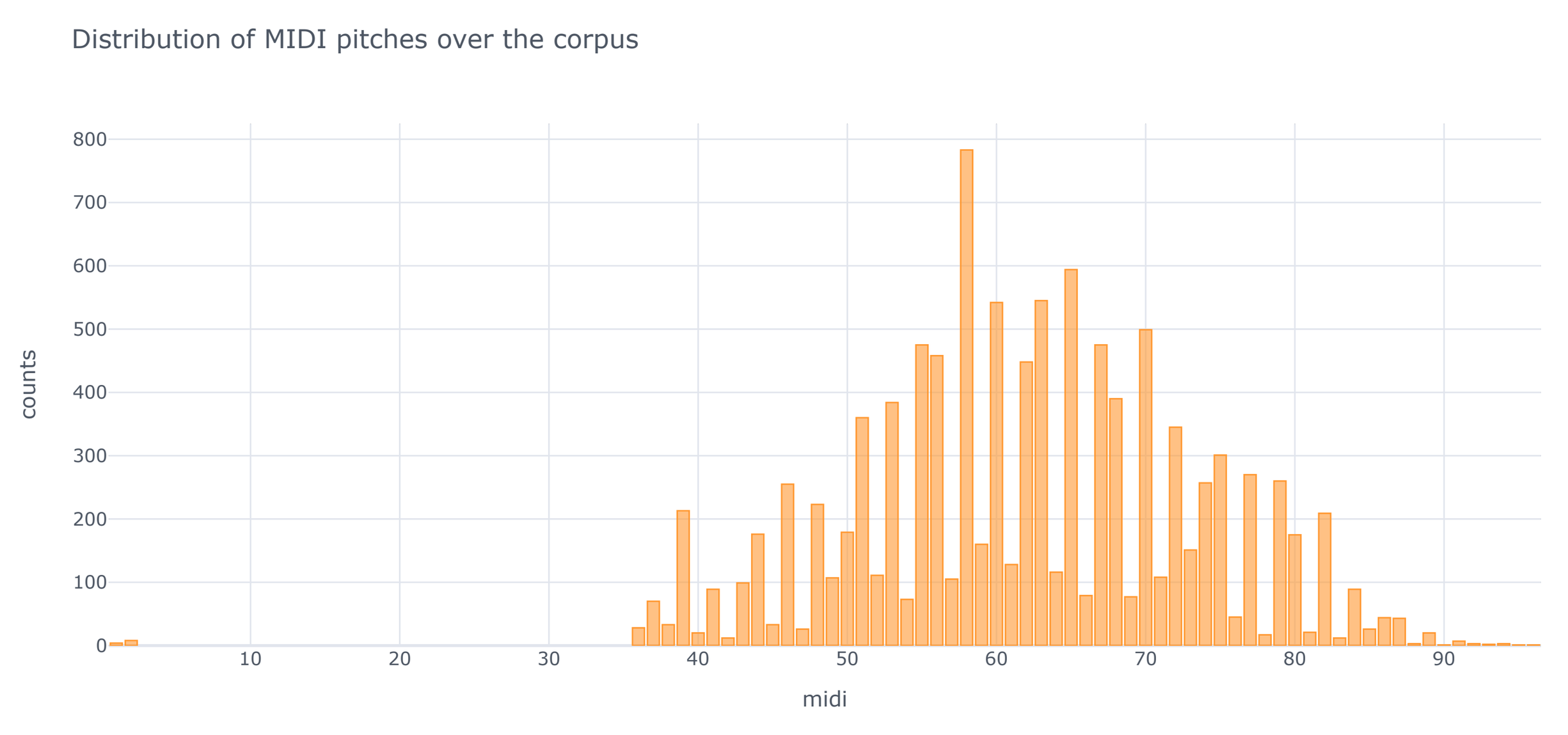
Export to plot.ly »

In [109…
```python
# Distribution of MIDI pitches over the corpus in 1806
nodes_1806.midi.value_counts().iplot('bar', title='Distribution of MIDI pitches over the corpus', xTitle='midi', yTitle='counts')
```



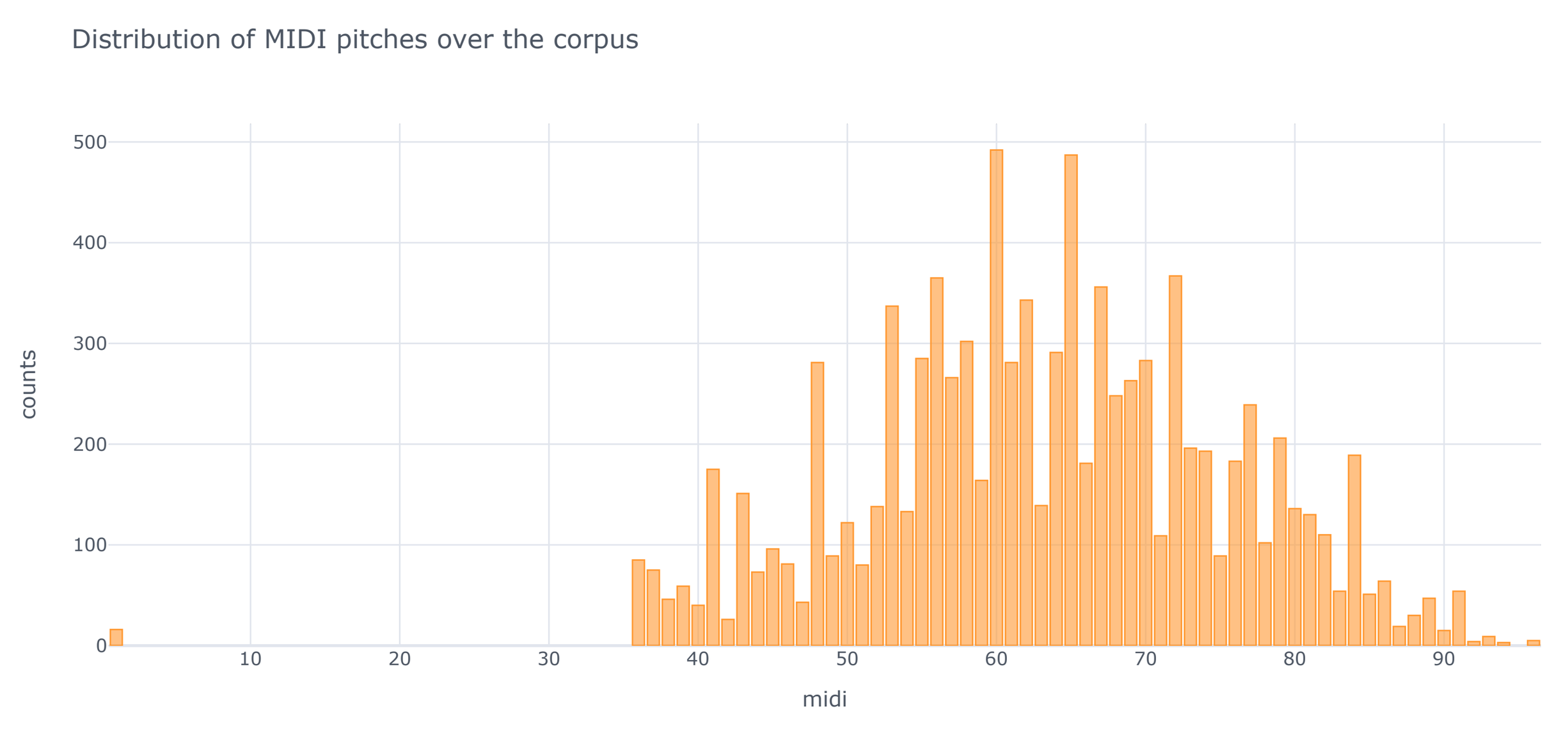
Export to plot.ly »

In [112…
```python
# Distribution of MIDI pitches over the corpus in 1809
nodes_1809.midi.value_counts().iplot('bar', title='Distribution of MIDI pitches over the corpus', xTitle='midi', yTitle='counts')
```



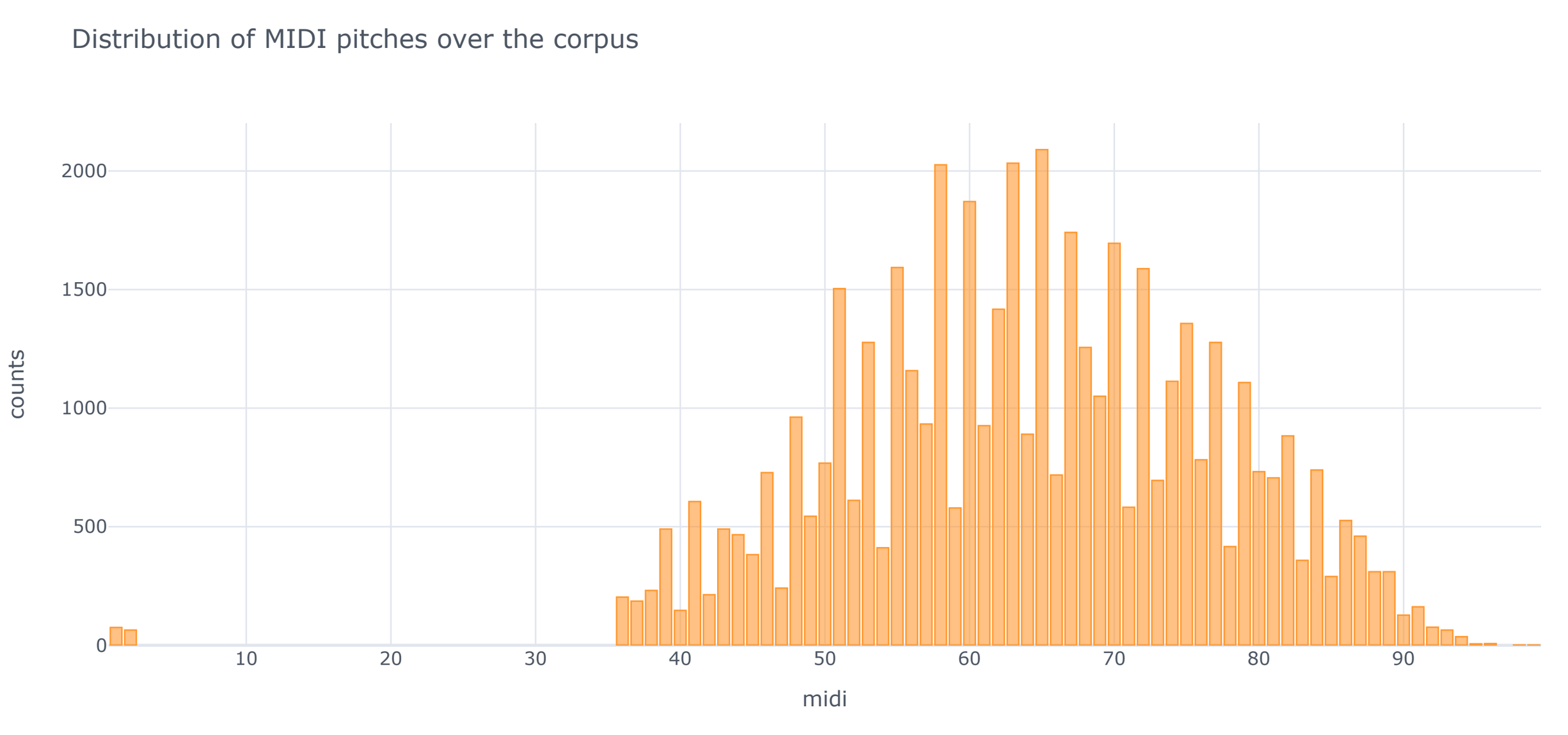
Export to plot.ly »

In [113…
```python
# Distribution of MIDI pitches over the corpus in 1810
nodes_1810.midi.value_counts().iplot('bar', title='Distribution of MIDI pitches over the corpus', xTitle='midi', yTitle='counts')
```



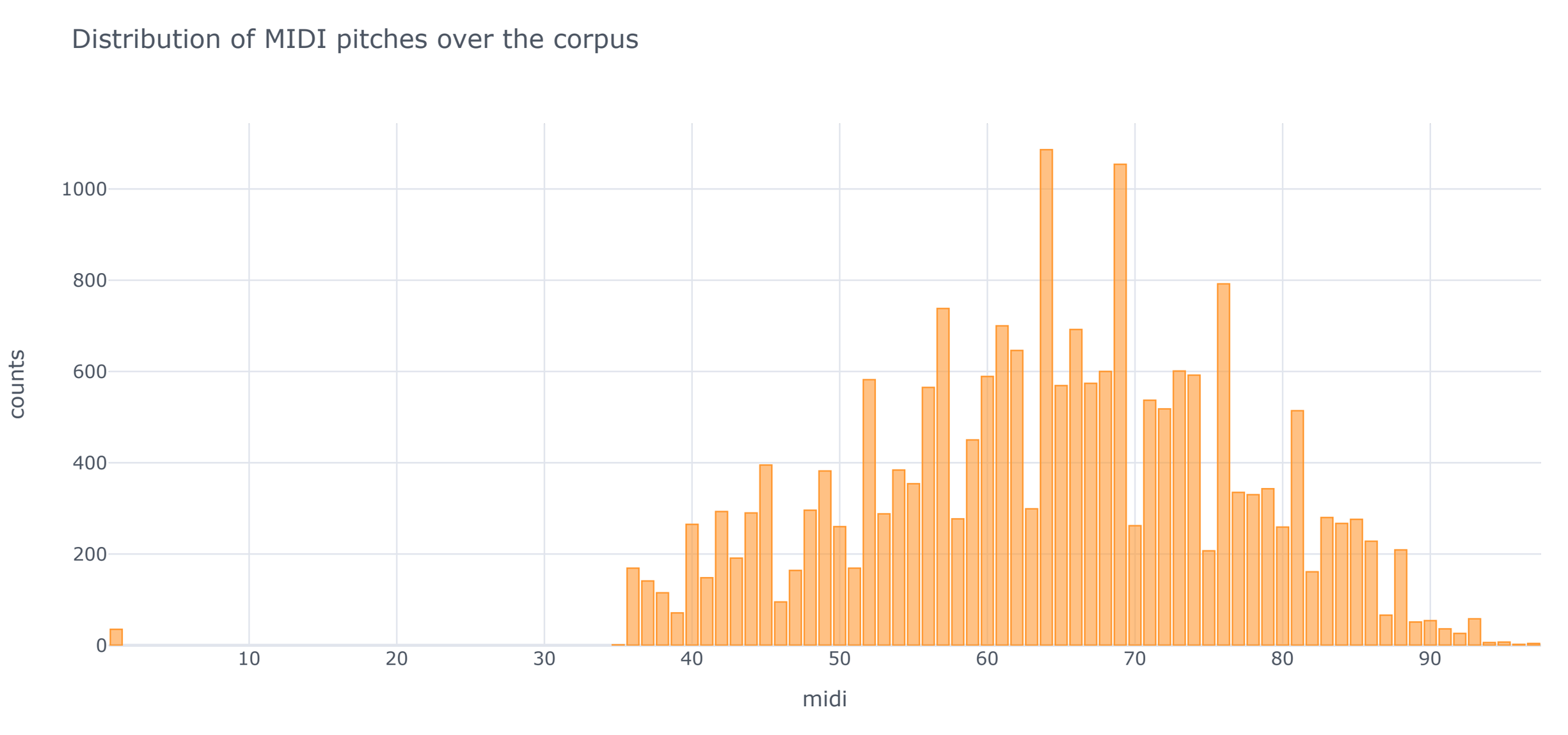
Export to plot.ly »

In [114…
```python
# Distribution of MIDI pitches over the corpus in 1825
nodes_1825.midi.value_counts().iplot('bar', title='Distribution of MIDI pitches over the corpus', xTitle='midi', yTitle='counts')
```



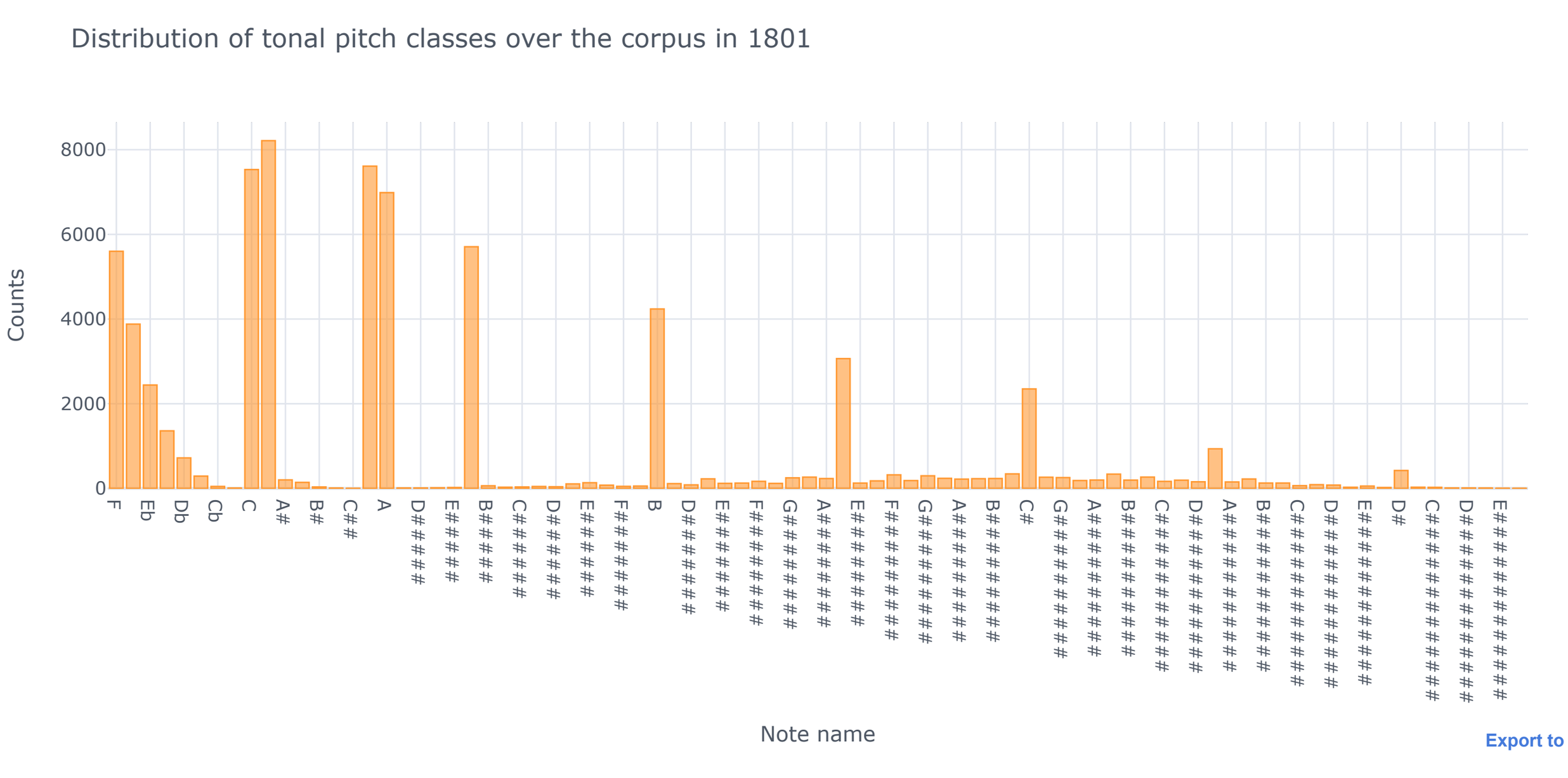
Export to plot.ly »

In [115…
```python
# Distribution of MIDI pitches over the corpus in 1826
nodes_1826.midi.value_counts().iplot('bar', title='Distribution of MIDI pitches over the corpus', xTitle='midi', yTitle='counts')
```



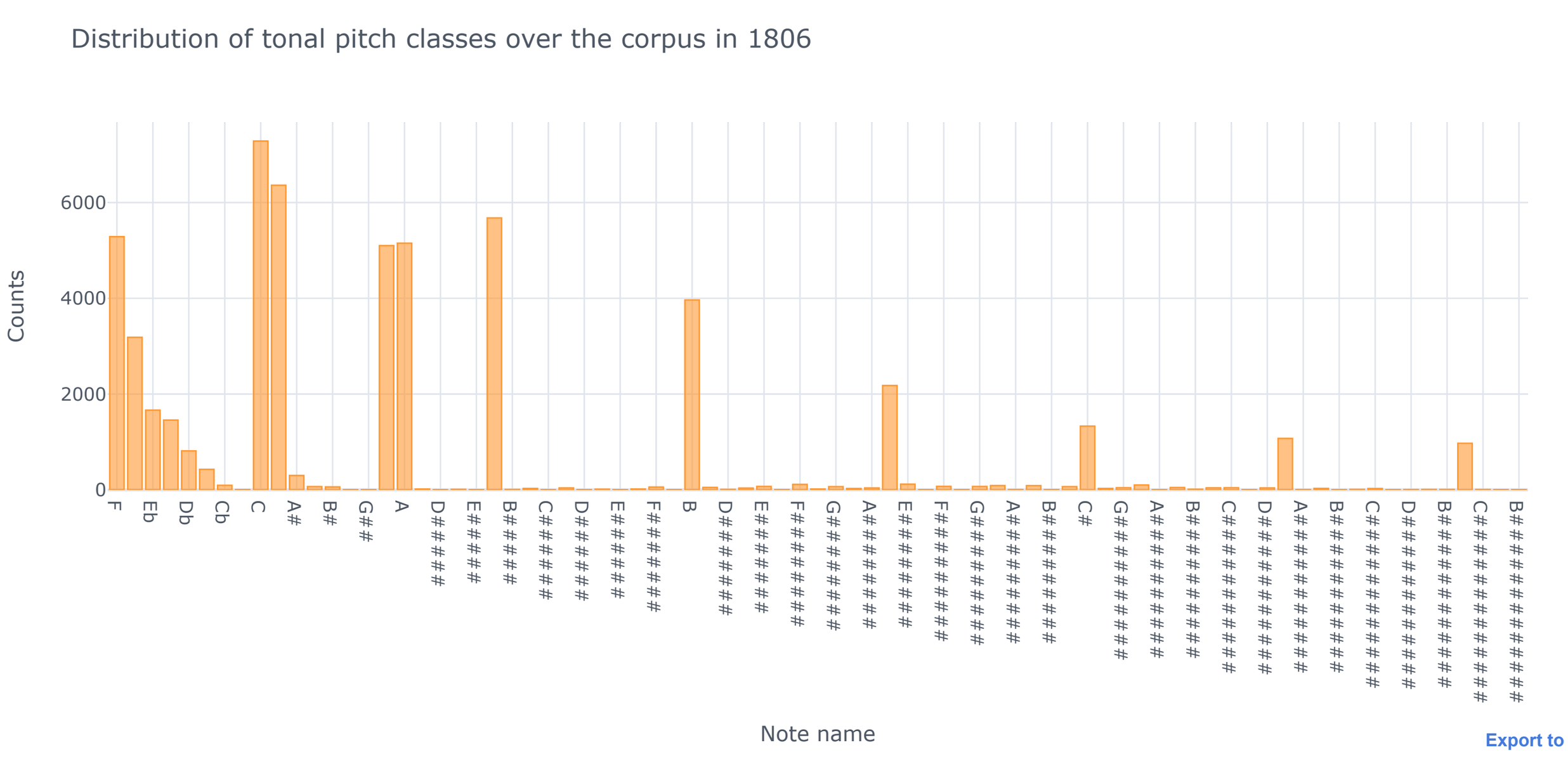
Export to plot.ly »

In [133…
```python
nodes_1801.tpc.value_counts().sort_index()[:-2].rename(tpc2name).iplot('bar', title='Distribution of tonal pitch classes over the corpus in 1801', xTitle='Note name', yTitle='Counts')
```



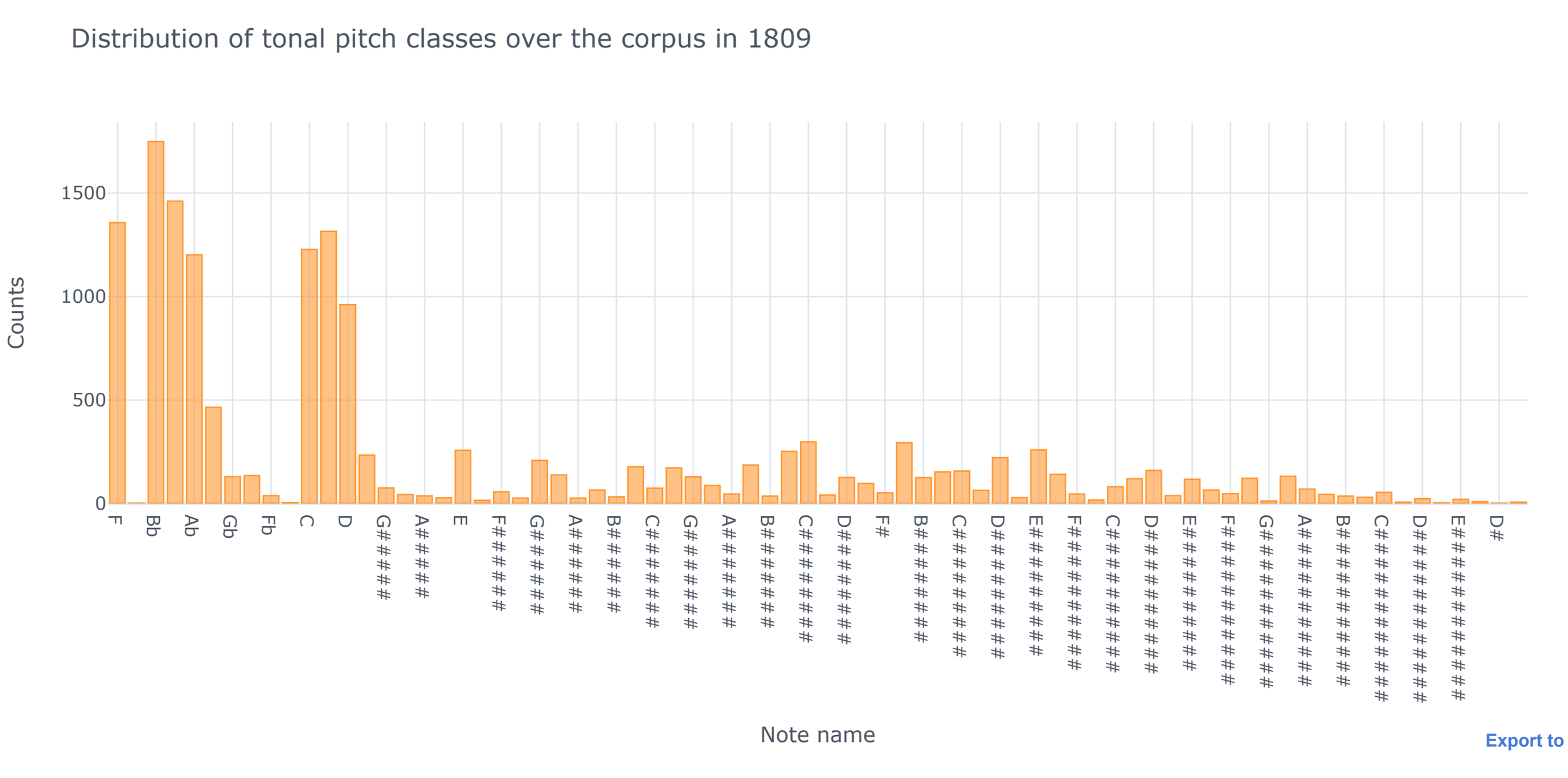
Export to plot.ly »

In [134…
```python
nodes_1806.tpc.value_counts().sort_index()[:-2].rename(tpc2name).iplot('bar', title='Distribution of tonal pitch classes over the corpus in 1806', xTitle='Note name', yTitle='Counts')
```



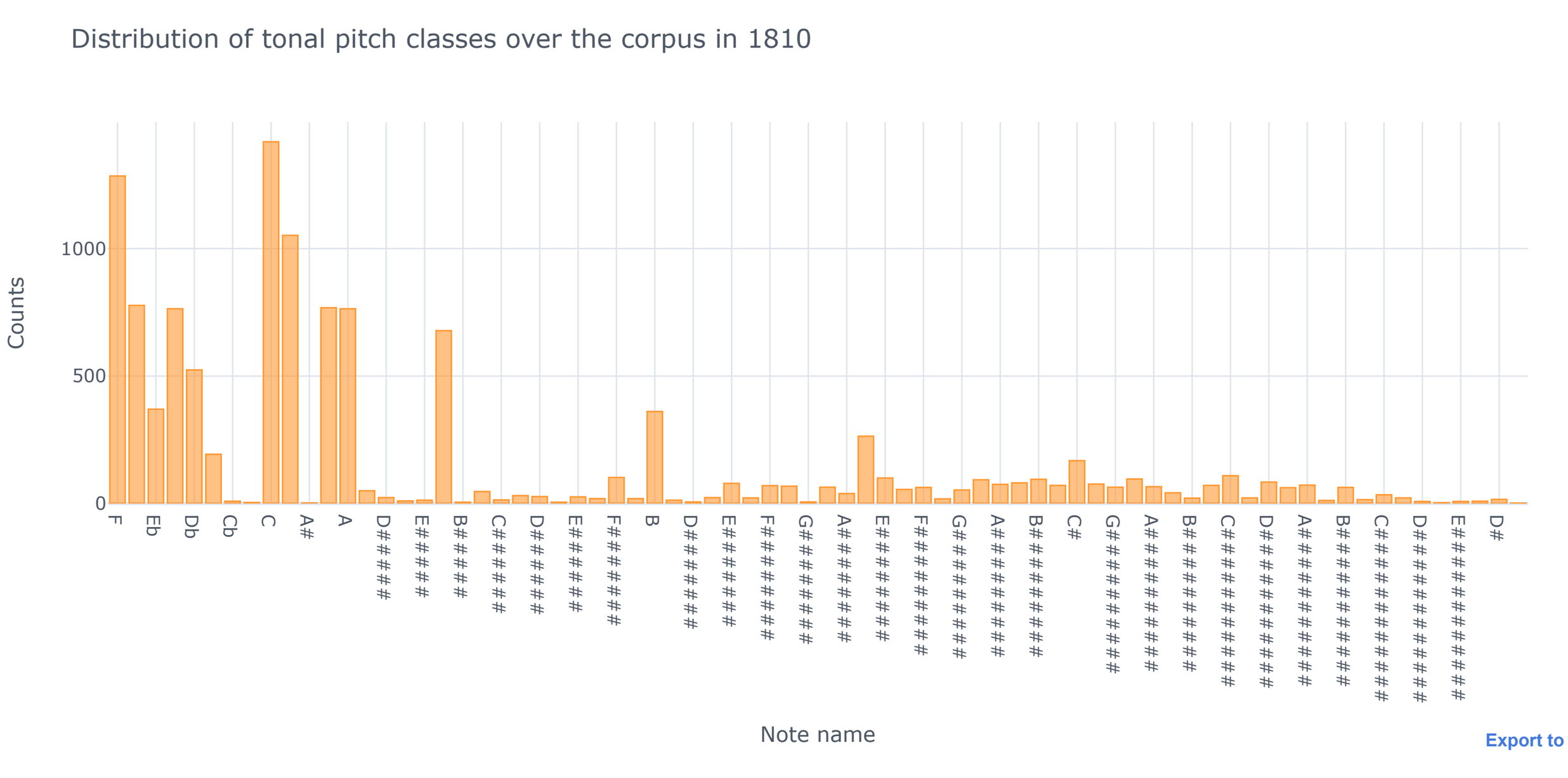
Export to plot.ly »

In [136…
```python
nodes_1809.tpc.value_counts().sort_index()[:-2].rename(tpc2name).iplot('bar', title='Distribution of tonal pitch classes over the corpus in 1809', xTitle='Note name', yTitle='Counts')
```



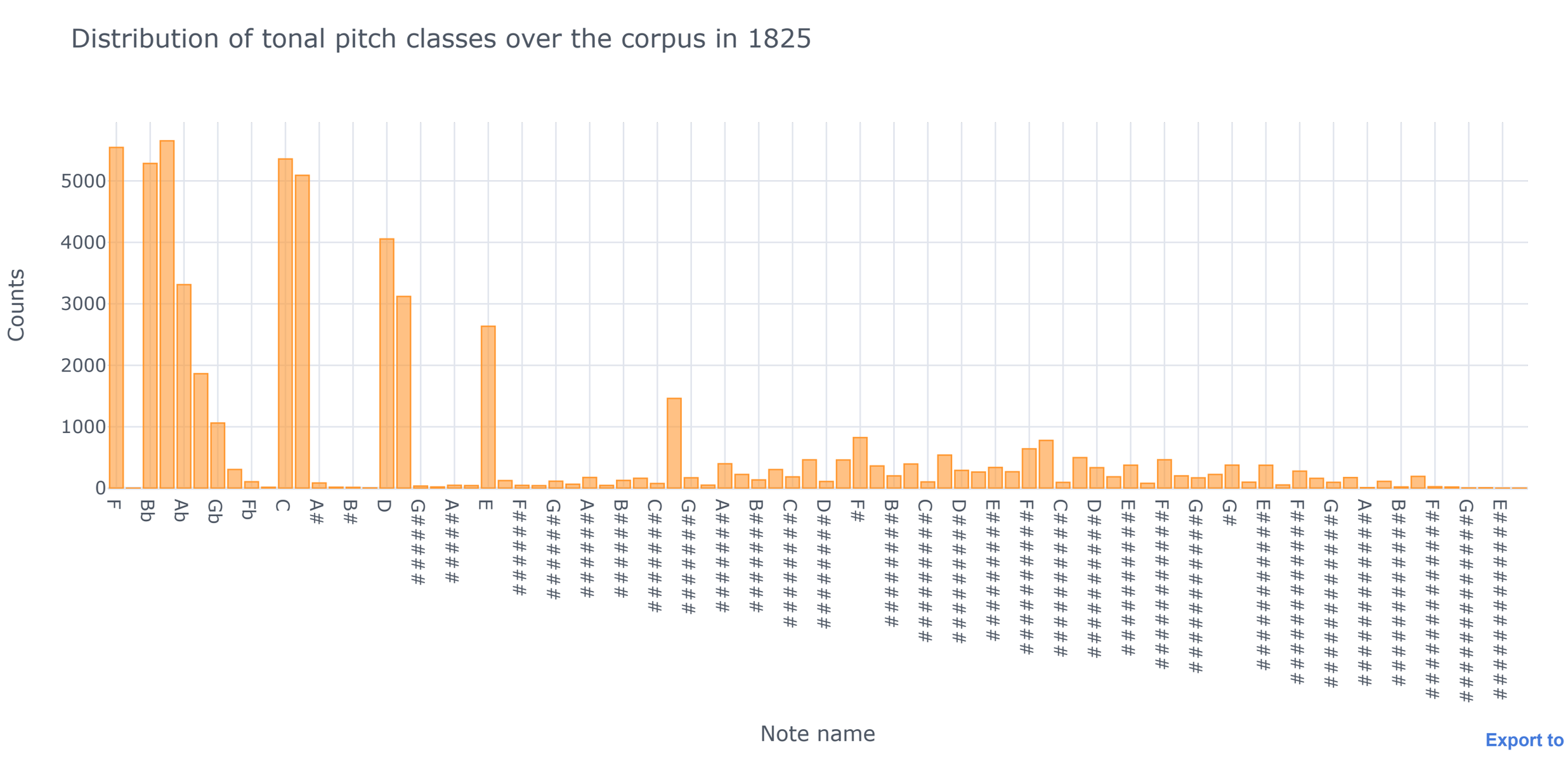
Export to plot.ly »

In [137…
```python
nodes_1810.tpc.value_counts().sort_index()[:-2].rename(tpc2name).iplot('bar', title='Distribution of tonal pitch classes over the corpus in 1810', xTitle='Note name', yTitle='Counts')
```



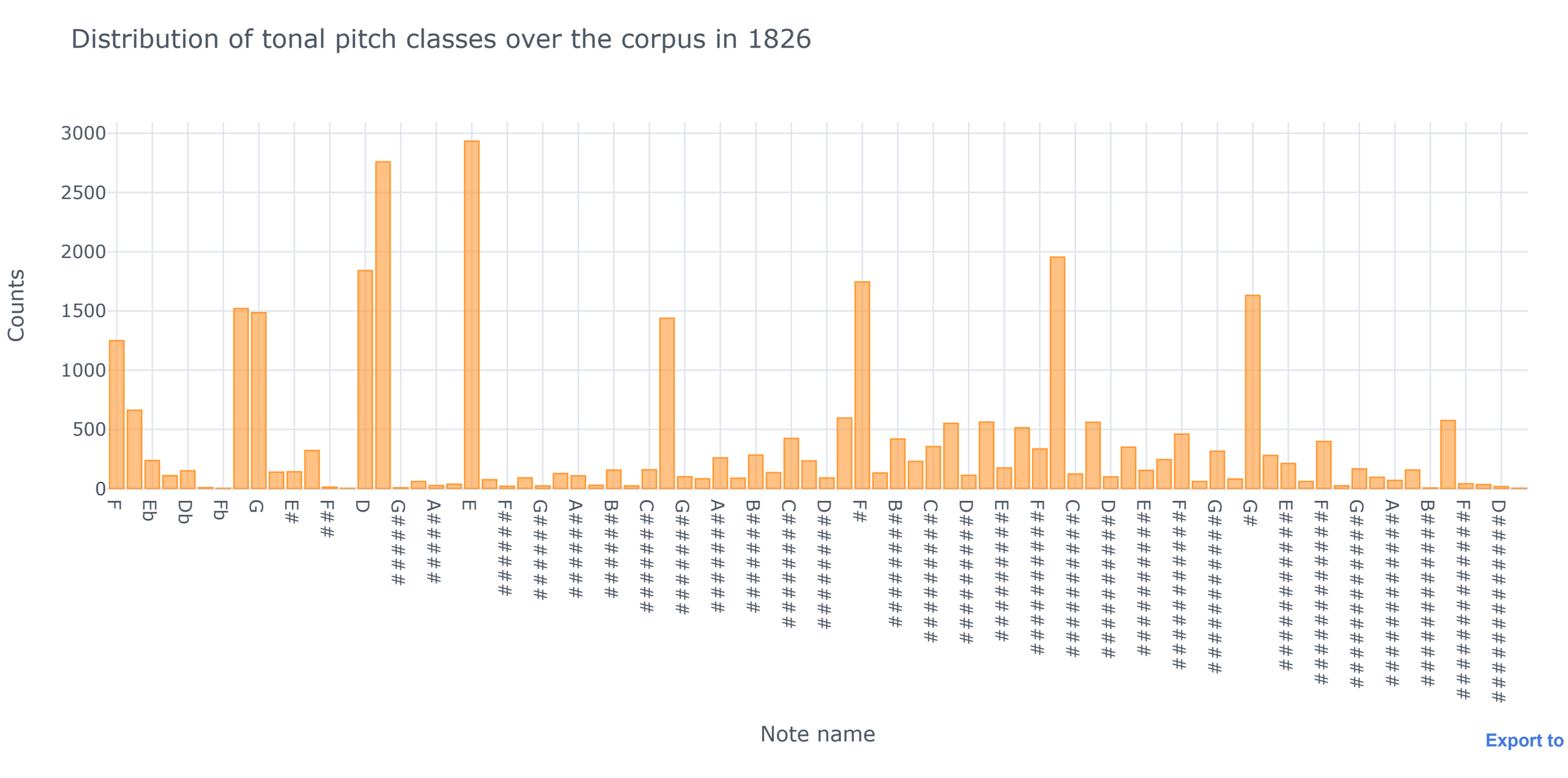
Export to plot.ly »

In [138…
```python
nodes_1825.tpc.value_counts().sort_index()[:-2].rename(tpc2name).iplot('bar', title='Distribution of tonal pitch classes over the corpus in 1825', xTitle='Note name', yTitle='Counts')
```



Export to plot.ly »

In [139…
```python
nodes_1826.tpc.value_counts().sort_index()[:-2].rename(tpc2name).iplot('bar', title='Distribution of tonal pitch classes over the corpus in 1826', xTitle='Note name', yTitle='Counts')
```



Export to plot.ly »

In [ ]:

In [ ]:

In [ ]: