What is the most efficient way of counting occurrences in pandas?

Asked 11 years, 11 months ago Modified 1 year, 5 months ago Viewed 391k times

I have a large (about 12M rows) DataFrame `df` : ← 1 Required Mock Data

```
df.columns = ['word','documents','frequency']
```

The following ran in a timely fashion:

```
word_grouping = df[['word','frequency']].groupby('word')
MaxFrequency_perWord = word_grouping[['frequency']].max().reset_index()
MaxFrequency_perWord.columns = ['word','MaxFrequency']
```

However, this is taking an unexpectedly long time to run: 2 Buggy Behavior

```
Occurrences_of_Words = word_grouping[['word']].count().reset_index()
```

What am I doing wrong here? Is there a better way to count occurrences in a large DataFrame?

```
df.word.describe()
```

ran pretty well, so I really did not expect this `Occurrences_of_Words` DataFrame to take very long to build.

3 Patched Behavior

353

I think `df['word'].value_counts()` should serve. By skipping the groupby machinery, you'll save some time. I'm not sure why `count` should be much slower than `max`. Both take some time to avoid missing values. (Compare with `size`.)

In any case, value_counts has been specifically optimized to handle object type, like your words, so I doubt you'll do much better than that.

Share Edit Follow edited Nov 19, 2013 at 16:23 answered Nov 19, 2013 at 16:18

A Stack Overflow Post

Not Executable | Mock Data Not Generated | Full Code not Replicated

B Auto Code Rover

```
...
<file>your_script.py</file>
<original>
word_grouping = df[['word','frequency']].groupby('word')
MaxFrequency_perWord = word_grouping[['frequency']].max().reset_index()
MaxFrequency_perWord.columns = ['word','MaxFrequency']
Occurrences_of_Words = word_grouping[['word']].count().reset_index()
</original>
<patched>
word_grouping = df[['word','frequency']].groupby('word')
MaxFrequency_perWord = word_grouping[['frequency']].max().reset_index()
MaxFrequency_perWord.columns = ['word','MaxFrequency']
Occurrences_of_Words = df['word'].value_counts().reset_index()
Occurrences_of_Words.columns = ['word', 'Occurrences']
</patched>
...
```

4 Replication of Buggy Behaviour

5 Patched Behavior

Not Executable | Mock Data Not Generated | Full Code not Replicated | Buggy code not replicated

C ArchCode

```
import pandas as pd

# Assuming df is your DataFrame with columns ['word', 'documents', 'frequency']

# Efficient way to count occurrences of words
Occurrences_of_Words = df['word'].value_counts().reset_index()
Occurrences_of_Words.columns = ['word', 'count']
```

6 Did not generate Mock Data

7 Patched Behavior

G Metrics

Generation input tokens: 5908	**Review input tokens:** 6043
Generation output tokens: 693	**Review output tokens:** 264
Geneartion Iteration: 1	**Review Iteration:** 1
Generation Time: 8.31 (Sec)	**Review Time:** 6.78 (Sec)

H Requirements

```
pandas==2.0.3
numpy==1.24.4
```

D StackCodeGen

Executable | Mock Data Generated | Full Code Replicated

E Buggy Code

```
import pandas as pd
import numpy as np
np.random.seed(42)
words = ['apple', 'banana', 'cherry', 'date', 'elderberry'] * 20000
documents = np.random.randint(1, 1000, 100000)
frequency = np.random.randint(1, 100, 100000)
df = pd.DataFrame({
'word': words,
'documents': documents,
'frequency': frequency
})
word_grouping = df[['word','frequency']].groupby('word')
MaxFrequency_perWord = word_grouping[['frequency']].max().reset_index()
MaxFrequency_perWord.columns = ['word','MaxFrequency']
word_grouping_for_count = df[['word','frequency']].groupby('word')
Occurrences_of_Words = word_grouping_for_count.count().reset_index()
Occurrences_of_Words.columns = ['word', 'count']
```

8 Mock Data generated

9 Replication of Buggy Behaviour

Executable | Mock Data Utilized | Full Code Replicated

F Patched Code

```
import pandas as pd
import numpy as np
np.random.seed(42)
words = ['apple', 'banana', 'cherry', 'date', 'elderberry'] * 20000
documents = np.random.randint(1, 1000, 100000)
frequency = np.random.randint(1, 100, 100000)
df = pd.DataFrame({
    'word': words,
    'documents': documents,
    'frequency': frequency
})
word_grouping = df[['word','frequency']].groupby('word')
MaxFrequency_perWord = word_grouping[['frequency']].max().reset_index()
MaxFrequency_perWord.columns = ['word','MaxFrequency']
Occurrences_of_Words = df['word'].value_counts().reset_index()
Occurrences_of_Words.columns = ['word', 'count']
```

10 Patched Behavior