### Homework 1

### Wordcount in Spark

Name: Eduardo Wang Zheng

E-mail: eduardo@sjtu.edu.cn

1. Experiments
   1. Count the number of each word
      1. Analysis

I do the following transformations:

1) Use ***flatmap*** to get a new distributed dataset of words

2) Use ***map*** to map each word into (*word*, *count*) pairs

3) Use ***reduceByKey*** to accumulate (*word*, *count*) pairs

And the following actions:

1) Use ***collect*** to get all (*word*, *count*) pairs

Further more, I use ***isalpha*** to ensure the *word* is fully consist of letters and use ***lower*** to lowercase the letters.

* + 1. Evaluation

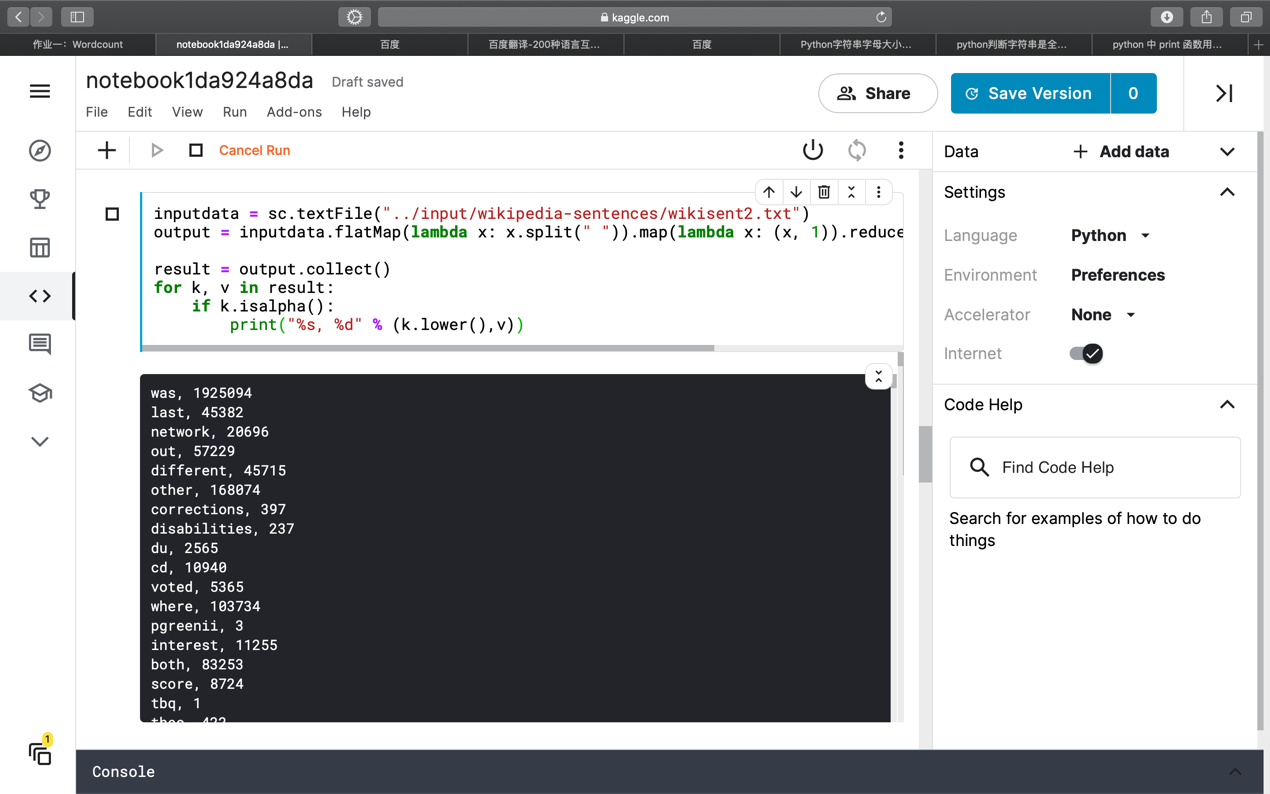


Figure 1 Count the number of each word

* 1. Find the most frequent word
     1. Analysis

I do the following transformations:

1) Use ***flatmap*** to get a new distributed dataset of words

2) Use ***map*** to map each word into (*word*, *count*) pairs

3) Use ***reduceByKey*** to accumulate (*word*, *count*) pairs

4) Use ***sortBy*** to sort (*word*, *count*) pairs by the value of *count* from big to small

And the following actions:

1) Use ***take(1)*** to get the most frequent word

* + 1. Evaluation

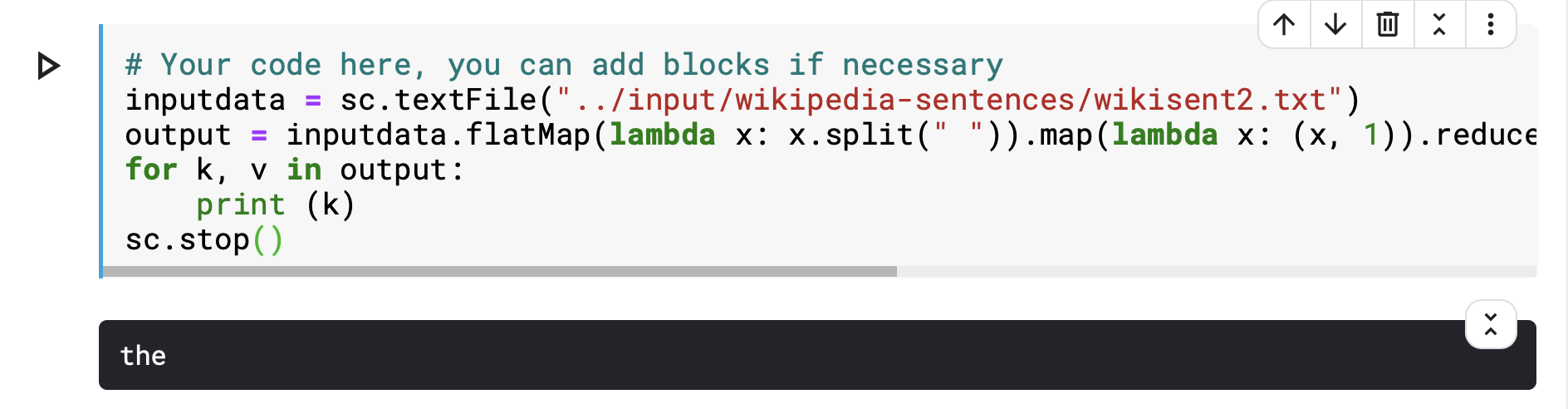


Figure 2 Find the most frequent word

* 1. Find the number of "China"
     1. Analysis

I do the following transformations:

1) Use ***flatmap*** to get a new distributed dataset of words

2) Use ***map*** to map each word into (*word*, *count*) pairs

3) Use ***reduceByKey*** to accumulate (*word*, *count*) pairs

4) Use ***filter*** to get (*China, count*) pair

And the following actions:

1) Use ***collect*** to get the number of "China"

* + 1. Evaluation

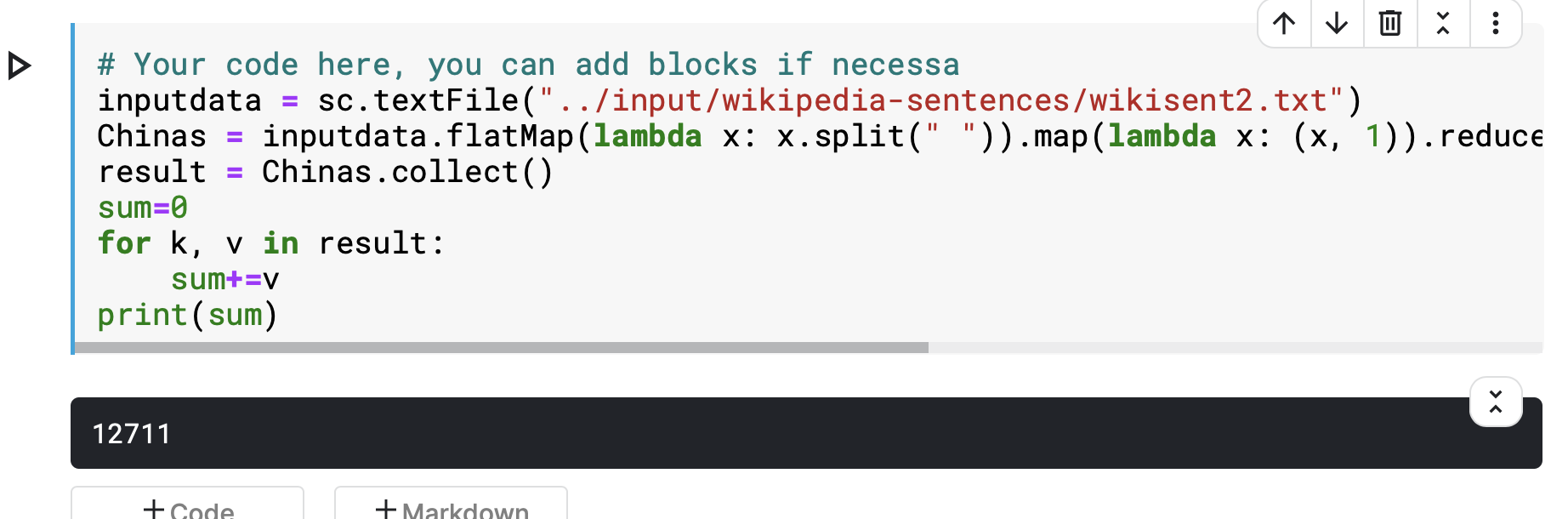


Figure 3 Find the number of "China"

1.4 Count the total number of (non-unique) words that start with each specific letter.

1.4.1 Analysis

I do the following transformations:

1) Use ***flatmap*** to get a new distributed dataset of words

2) Use ***map*** to map each word into (*word*, *count*) pairs

3) Use ***reduceByKey*** to accumulate (*word*, *c*) pairs

And the following actions:

1) Use ***collect*** to get all (*word*, *count*) pairs

Further more, I use ***startswith*** to find the *word* starts with a specific letter and accumulate the value of *count*

1.4.2 Evaluation

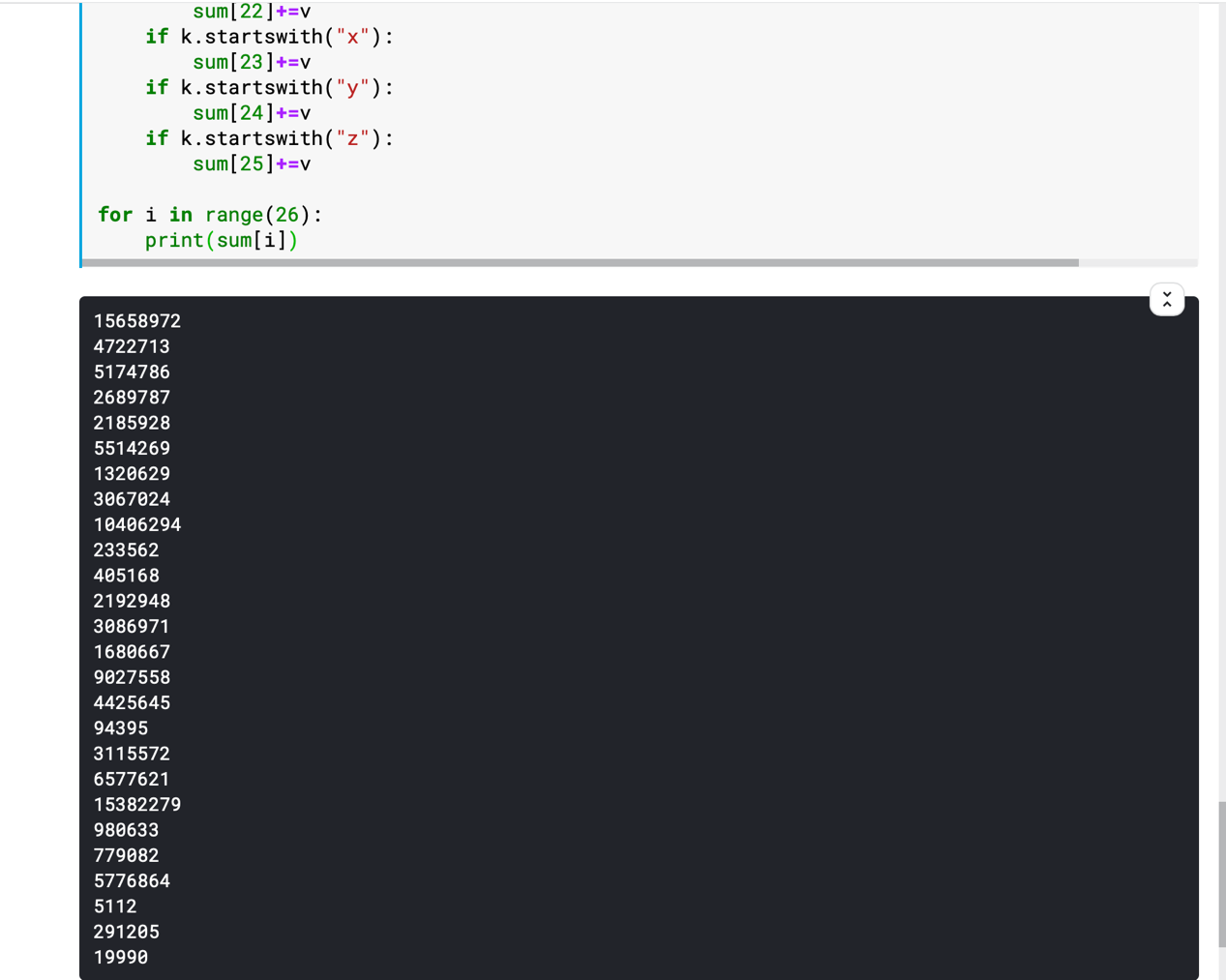


Figure 4 Count the total number of (non-unique) words that start with each specific letter.