

## Data Tools Package

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Namespace Documentation</b>	<b>9</b>
5.1	DataTools Namespace Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.2	DataTools.DB_datatools Namespace Reference . . . . .	9
5.2.1	Detailed Description . . . . .	9
5.2.2	Function Documentation . . . . .	9
5.2.2.1	closeconnections() . . . . .	10
5.2.2.2	db_addsynonymbylist() . . . . .	10
5.2.2.3	db_connect() . . . . .	10
5.2.2.4	db_finddrug_bysynonym() . . . . .	10
5.2.2.5	db_findrelations() . . . . .	10
5.2.2.6	db_findtarget_bysynonym() . . . . .	11
5.2.2.7	db_findtargetsynonyms() . . . . .	11

5.3	DataTools.language_datatools Namespace Reference	11
5.3.1	Function Documentation	11
5.3.1.1	network_score()	12
5.3.1.2	pos_tag()	12
5.3.1.3	sentence_score()	13
5.3.1.4	sentiment_score()	13
5.3.1.5	split()	14
5.3.1.6	weighted_sentence_score()	14
5.3.1.7	weighted_sentiment_score()	15
5.4	DataTools.network_datatools Namespace Reference	15
5.4.1	Function Documentation	15
5.4.1.1	findaliases()	15
5.4.1.2	findnetworknodes()	16
5.4.1.3	perceptronify()	16
5.5	DataTools.ps_datatools Namespace Reference	16
5.5.1	Detailed Description	16
5.5.2	Function Documentation	16
5.5.2.1	ps_clean_ref()	17
5.5.2.2	ps_clean_sentence_tags()	17
5.5.2.3	ps_clear_end()	18
5.5.2.4	ps_drop_duplicates()	19
5.5.2.5	ps_flag_duplicates()	19
5.5.2.6	ps_has_duplicates()	19
5.5.2.7	ps_perceptronify()	19
5.5.2.8	ps_read_csv()	20
5.5.2.9	ps_read_entitylist()	20
5.5.2.10	ps_replace_syntax()	20
5.5.2.11	ps_return_duplicates()	21
5.5.2.12	ps_write_csv()	21

<b>6</b>	<b>Class Documentation</b>	<b>23</b>
6.1	DataTools.language_datatools.DictionaryTagger Class Reference . . . . .	23
6.1.1	Detailed Description . . . . .	24
6.1.2	Constructor & Destructor Documentation . . . . .	24
6.1.2.1	__init__() . . . . .	24
6.1.3	Member Function Documentation . . . . .	24
6.1.3.1	tag() . . . . .	24
6.1.3.2	tag_sentence() . . . . .	25
6.1.4	Member Data Documentation . . . . .	25
6.1.4.1	dictionary . . . . .	25
6.1.4.2	max_key_size . . . . .	25
<b>7</b>	<b>File Documentation</b>	<b>27</b>
7.1	__init__.py File Reference . . . . .	27
7.2	DB_datatools.py File Reference . . . . .	27
7.3	language_datatools.py File Reference . . . . .	27
7.4	network_datatools.py File Reference . . . . .	28
7.5	ps_datatools.py File Reference . . . . .	28
	<b>Index</b>	<b>29</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">DataTools</a> . . . . .	9
<a href="#">DataTools.DB_datatools</a> . . . . .	9
<a href="#">DataTools.language_datatools</a> . . . . .	11
<a href="#">DataTools.network_datatools</a> . . . . .	15
<a href="#">DataTools.ps_datatools</a> . . . . .	16





## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

object  
    DataTools.language\_datatools.DictionaryTagger . . . . . [23](#)



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">DataTools.language_datatools.DictionaryTagger</a> . . . . .	23
---	----



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">__init__.py</a>	27
<a href="#">DB_datatools.py</a>	27
<a href="#">language_datatools.py</a>	27
<a href="#">network_datatools.py</a>	28
<a href="#">ps_datatools.py</a>	28



## Chapter 5

# Namespace Documentation

### 5.1 DataTools Namespace Reference

#### Namespaces

- [DB\\_datatools](#)
- [language\\_datatools](#)
- [network\\_datatools](#)
- [ps\\_datatools](#)

#### 5.1.1 Detailed Description

The DataTools Module

### 5.2 DataTools.DB\_datatools Namespace Reference

#### Functions

- def [db\\_connect](#) (dbfile)
- def [db\\_findrelations](#) (cursor, search\_terms)
- def [db\\_findtarget\\_bysynonym](#) ()
- def [db\\_finddrug\\_bysynonym](#) ()
- def [db\\_addsynonymbylist](#) ()
- def [db\\_findtargetsynonyms](#) (curs, target)
- def [closeconnections](#) (conn, curs)

#### 5.2.1 Detailed Description

Module to interact with SQLite DB for CSB Lab tools

#### 5.2.2 Function Documentation

#### 5.2.2.1 closeconnections()

```
def DataTools.DB_datatools.closeconnections (
    conn,
    curs )
```

Closes DB cursor and connection  
:param conn:  
:param curs:  
:return:

#### 5.2.2.2 db\_addsynonymbylist()

```
def DataTools.DB_datatools.db_addsynonymbylist ( )
```

#### 5.2.2.3 db\_connect()

```
def DataTools.DB_datatools.db_connect (
    dbfile )
```

connect to DB and return cursor and db connection.  
:param dbfile:  
:return: Database Cursor, Connection

#### 5.2.2.4 db\_finddrug\_bysynonym()

```
def DataTools.DB_datatools.db_finddrug_bysynonym ( )
```

#### 5.2.2.5 db\_findrelations()

```
def DataTools.DB_datatools.db_findrelations (
    cursor,
    search_terms )
```

Find all the Drug Target relations, as well as off Target drug relations of relevant drugs  
:param cursor:  
:param search\_terms:  
:return: List of drug relations



#### 5.2.2.6 db\_findtarget\_bysynonym()

```
def DataTools.DB_datatools.db_findtarget_bysynonym ( )
```

#### 5.2.2.7 db\_findtargetsynonyms()

```
def DataTools.DB_datatools.db_findtargetsynonyms (
    curs,
    target )
```

Finds all the synonyms of a given target.  
:param curs: Cursor  
:param target: Target to be searched  
:return: List of all the target's synonyms

## 5.3 DataTools.language\_datatools Namespace Reference

### Classes

- class [DictionaryTagger](#)

### Functions

- def [split](#) (text)
- def [pos\\_tag](#) (sentences)
- def [network\\_score](#) (network, add\_polarity=True, flag\_changes=True)
- def [sentence\\_score](#) (sentence\_tokens, previous\_token, acum\_score)
- def [sentiment\\_score](#) (review)
- def [weighted\\_sentence\\_score](#) (sentence\_tokens)
- def [weighted\\_sentiment\\_score](#) (review, matrix\_flag)

#### 5.3.1 Function Documentation

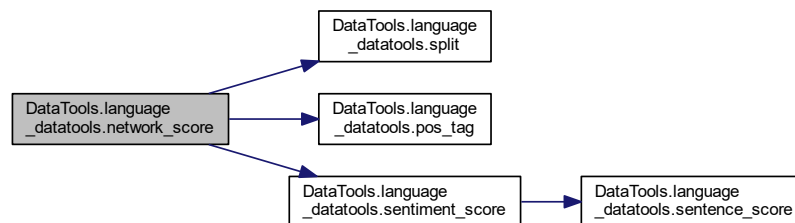
### 5.3.1.1 network\_score()

```
def DataTools.language_datatools.network_score (
    network,
    add_polarity = True,
    flag_changes = True )
```

Runs Sentiment analysis on a network and returns an analyzed and scored network.

```
:param network:
:param add_polarity:
:param flag_changes:
:return: scored network
```

Here is the call graph for this function:



### 5.3.1.2 pos\_tag()

```
def DataTools.language_datatools.pos_tag (
    sentences )
```

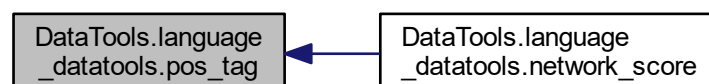
input format: list of lists of words

e.g.: [['this', 'is', 'a', 'sentence'], ['this', 'is', 'another', 'one']]

output format: list of lists of tagged tokens. Each tagged tokens has a form, a lemma, and a list of tags

e.g.: [[('this', 'this', ['DT']), ('is', 'be', ['VB']), ('a', 'a', ['DT']), ('sentence', 'sentence', ['NN']),  
 (('this', 'this', ['DT']), ('is', 'be', ['VB']), ('another', 'another', ['DT']),  
 ('one', 'one', ['CARD'])]]

Here is the caller graph for this function:

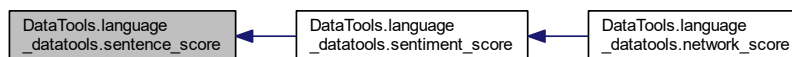


#### 5.3.1.3 sentence\_score()

```
def DataTools.language_datatools.sentence_score (
    sentence_tokens,
    previous_token,
    acum_score )

    scores sentences thats been tagged.
:param sentence_tokens:
:param previous_token:
:param acum_score:
:return:
```

Here is the caller graph for this function:

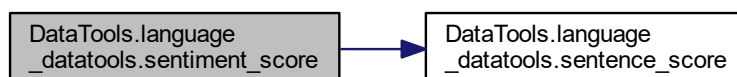


#### 5.3.1.4 sentiment\_score()

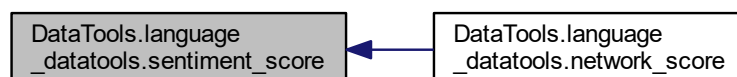
```
def DataTools.language_datatools.sentiment_score (
    review )

Returns cumulative score of scored sentences
:param review:
:return:
```

Here is the call graph for this function:



Here is the caller graph for this function:



#### 5.3.1.5 split()

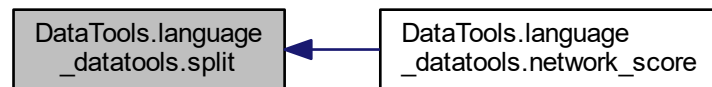
```
def DataTools.language_datatools.split (  
    text )
```

input format: a paragraph of text

output format: a list of lists of words.

e.g.: [['this', 'is', 'a', 'sentence'], ['this', 'is', 'another', 'one']]

Here is the caller graph for this function:



#### 5.3.1.6 weighted\_sentence\_score()

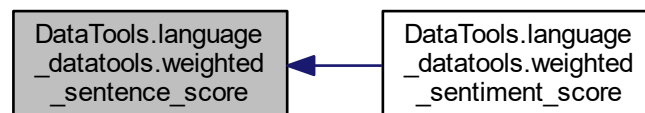
```
def DataTools.language_datatools.weighted_sentence_score (  
    sentence_tokens )
```

Weighted score...

:param sentence\_tokens:

:return: Totally arbitrary weight based on hokey pokey

Here is the caller graph for this function:

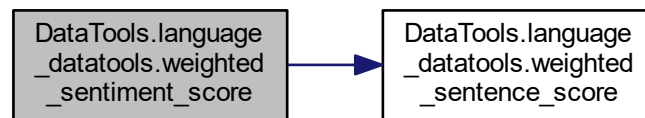


## 5.3.1.7 weighted\_sentiment\_score()

```
def DataTools.language_datatools.weighted_sentiment_score (
    review,
    matrix_flag )
```

```
    Totally arbitrary weight based on hokey pokey
:param review:
:param matrix_flag:
:return: Totally arbitrary weight based on hokey pokey
```

Here is the call graph for this function:



## 5.4 DataTools.network\_datatools Namespace Reference

## Functions

- def [findnetworknodes](#) (network)
- def [findaliases](#) (entitylist)
- def [perceptronify](#) (network, indegree\_max=8)

## 5.4.1 Function Documentation

## 5.4.1.1 findaliases()

```
def DataTools.network_datatools.findaliases (
    entitylist )
```

```
    Take in a dataframe that has nodes and their aliases, return a list that joins the Aliases
:param entitylist:
:return: List of [Name, Alias]
```

#### 5.4.1.2 findnetworknodes()

```
def DataTools.network_datatools.findnetworknodes (
    network )
```

Take a network with columns 'Source' and 'Target',  
Return unique nodes in said network  
:param network:  
:return: list of unique nodes

#### 5.4.1.3 perceptronify()

```
def DataTools.network_datatools.perceptronify (
    network,
    indegree_max = 8 )
```

## 5.5 DataTools.ps\_datatools Namespace Reference

### Functions

- def [ps\\_read\\_csv](#) (filename, ps\_format=True)
- def [ps\\_read\\_entitylist](#) (filename)
- def [ps\\_has\\_duplicates](#) (network)
- def [ps\\_return\\_duplicates](#) (network)
- def [ps\\_flag\\_duplicates](#) (network)
- def [ps\\_drop\\_duplicates](#) (network)
- def [ps\\_clean\\_ref](#) (network)
- def [ps\\_clean\\_sentence\\_tags](#) (s)
- def [ps\\_clear\\_end](#) (s)
- def [ps\\_replace\\_syntax](#) (s)
- def [ps\\_perceptronify](#) (in\_network, max\_indegree=8)
- def [ps\\_write\\_csv](#) (network, csv\_name='network\_please\_rename.csv', exclude\_unknown=False)

#### 5.5.1 Detailed Description

This Module is for use in importing networks from Pathway Studio,  
as well as exporting networks to other consumers

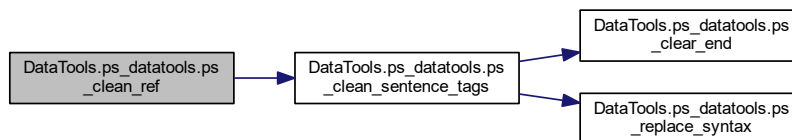
#### 5.5.2 Function Documentation

## 5.5.2.1 ps\_clean\_ref()

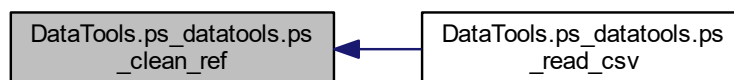
```
def DataTools.ps_datatools.ps_clean_ref (  
    network )
```

Take in network, clean ALL references column of PS tags  
:param network: network to have references cleaned from  
:return: Network, with new Clean\_References column.

Here is the call graph for this function:



Here is the caller graph for this function:

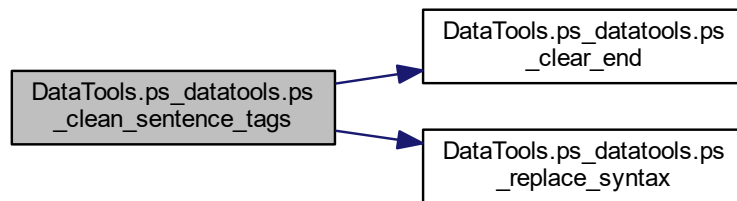


## 5.5.2.2 ps\_clean\_sentence\_tags()

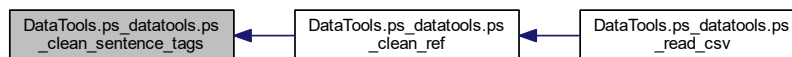
```
def DataTools.ps_datatools.ps_clean_sentence_tags (  
    s )
```

Removes PS tags from SINGLE reference.  
:param s:  
:return:

Here is the call graph for this function:



Here is the caller graph for this function:



### 5.5.2.3 ps\_clear\_end()

```
def DataTools.ps_datatools.ps_clear_end (
    s )
```

Removes trailing whitespace from PS sentences.  
:param s:  
:return:

Here is the caller graph for this function:





#### 5.5.2.4 ps\_drop\_duplicates()

```
def DataTools.ps_datatools.ps_drop_duplicates (
    network )
```

Drops Duplicates from network

```
:param network:
:return: network df minus the duplicates
```

#### 5.5.2.5 ps\_flag\_duplicates()

```
def DataTools.ps_datatools.ps_flag_duplicates (
    network )
```

Changes Duplicated bool to True if duplicated.

```
:param network:
:return:
```

#### 5.5.2.6 ps\_has\_duplicates()

```
def DataTools.ps_datatools.ps_has_duplicates (
    network )
```

Boolean check to see if there are duplicated rows

```
:param network:
:return:
```

#### 5.5.2.7 ps\_perceptronify()

```
def DataTools.ps_datatools.ps_perceptronify (
    in_network,
    max_indegree = 8 )
```

Function to create a perceptron network based on input network from pathway studio. Splits nodes with high in-degrees into a sub network of nodes that feed into a perceptron node, thus limiting the number of edges coming into any single node.

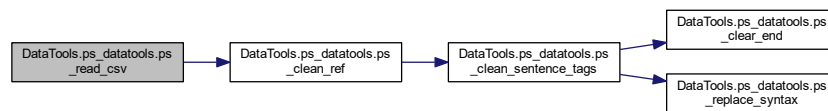
```
:param in_network: File Location of network to perceptronify
:param max_indegree: Max number of edges node can have coming in.
:return: Network that has been perceptronified.
```

### 5.5.2.8 ps\_read\_csv()

```
def DataTools.ps_datatools.ps_read_csv (
    filename,
    ps_format = True )
```

Read CSV from Pathway Studio, with or without initial data clean  
:param filename: File to be read  
:param ps\_format: Boolean to format PS raw to cleaned  
:return: Raw pandas DataFrame, or cleaned pandas DF

Here is the call graph for this function:



### 5.5.2.9 ps\_read\_entitylist()

```
def DataTools.ps_datatools.ps_read_entitylist (
    filename )
```

Read the Pathway Studio Entity List file export  
:param filename: path/to/file  
:return: pandas DataFrame of Name and Alias

### 5.5.2.10 ps\_replace\_syntax()

```
def DataTools.ps_datatools.ps_replace_syntax (
    s )
```

Takes in sentence `s` and removes PS internal formatting  
:param s: Sentence to be formatted  
:return: cleaned sentence.

Here is the caller graph for this function:



### 5.5.2.11 ps\_return\_duplicates()

```
def DataTools.ps_datatools.ps_return_duplicates (
    network )
```

Returns duplicated rows in DF

:param network:  
:return:

### 5.5.2.12 ps\_write\_csv()

```
def DataTools.ps_datatools.ps_write_csv (
    network,
    csv_name = 'network_please_rename.csv',
    exclude_unknown = False )
```

Write network, formatted to CSV

:param network: network to be written to csv.  
:param csv\_name: Name for new file, defaulted 'network\_please\_rename.csv'  
:param exclude\_unknown: flag to exclude rows where Polarity is unknown  
:return: none

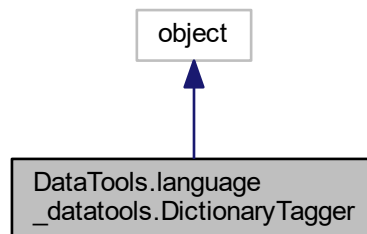


## Chapter 6

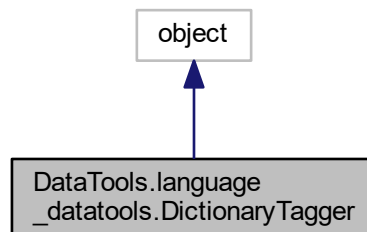
# Class Documentation

### 6.1 DataTools.language\_datatools.DictionaryTagger Class Reference

Inheritance diagram for DataTools.language\_datatools.DictionaryTagger:



Collaboration diagram for DataTools.language\_datatools.DictionaryTagger:



## Public Member Functions

- `def __init__ (self, dictionary_paths)`
- `def tag (self, postagged_sentences)`
- `def tag_sentence (self, sentence, tag_with_lemmas=False)`

## Public Attributes

- `dictionary`
- `max_key_size`

### 6.1.1 Detailed Description

Class to tag sentences

### 6.1.2 Constructor & Destructor Documentation

#### 6.1.2.1 \_\_init\_\_()

```
def DataTools.language_datatools.DictionaryTagger.__init__ (
    self,
    dictionary_paths )
```

Opens YAML files and instantiates object of type DictionaryTagger  
:param dictionary\_paths:

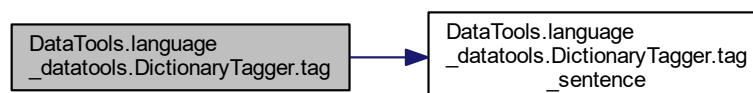
### 6.1.3 Member Function Documentation

#### 6.1.3.1 tag()

```
def DataTools.language_datatools.DictionaryTagger.tag (
    self,
    postagged_sentences )
```

Tags sentences with internal tags for positive and negative  
:param postagged\_sentences:  
:return: tagged sentences

Here is the call graph for this function:



### 6.1.3.2 tag\_sentence()

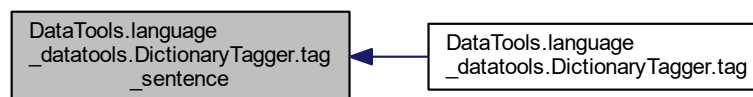
```
def DataTools.language_datatools.DictionaryTagger.tag_sentence (
    self,
    sentence,
    tag_with_lemmas = False )
```

the result is only one tagging of all the possible ones.

The resulting tagging is determined by these two priority rules:

- longest matches have higher priority
- search is made from left to right

Here is the caller graph for this function:



## 6.1.4 Member Data Documentation

### 6.1.4.1 dictionary

`DataTools.language_datatools.DictionaryTagger.dictionary`

### 6.1.4.2 max\_key\_size

`DataTools.language_datatools.DictionaryTagger.max_key_size`

The documentation for this class was generated from the following file:

- [language\\_datatools.py](#)





## Chapter 7

# File Documentation

### 7.1 `__init__.py` File Reference

#### Namespaces

- [DataTools](#)

### 7.2 `DB_datatools.py` File Reference

#### Namespaces

- [DataTools.DB\\_datatools](#)

#### Functions

- def [DataTools.DB\\_datatools.db\\_connect](#) (dbfile)
- def [DataTools.DB\\_datatools.db\\_findrelations](#) (cursor, search\_terms)
- def [DataTools.DB\\_datatools.db\\_findtarget\\_bysynonym](#) ()
- def [DataTools.DB\\_datatools.db\\_finddrug\\_bysynonym](#) ()
- def [DataTools.DB\\_datatools.db\\_addsynonymbylist](#) ()
- def [DataTools.DB\\_datatools.db\\_findtargetsynonyms](#) (curs, target)
- def [DataTools.DB\\_datatools.closeconnections](#) (conn, curs)

### 7.3 `language_datatools.py` File Reference

#### Classes

- class [DataTools.language\\_datatools.DictionaryTagger](#)

#### Namespaces

- [DataTools.language\\_datatools](#)

## Functions

- def [DataTools.language\\_datatools.split](#) (text)
- def [DataTools.language\\_datatools.pos\\_tag](#) (sentences)
- def [DataTools.language\\_datatools.network\\_score](#) (network, add\_polarity=True, flag\_changes=True)
- def [DataTools.language\\_datatools.sentence\\_score](#) (sentence\_tokens, previous\_token, acum\_score)
- def [DataTools.language\\_datatools.sentiment\\_score](#) (review)
- def [DataTools.language\\_datatools.weighted\\_sentence\\_score](#) (sentence\_tokens)
- def [DataTools.language\\_datatools.weighted\\_sentiment\\_score](#) (review, matrix\_flag)

## 7.4 network\_datatools.py File Reference

### Namespaces

- [DataTools.network\\_datatools](#)

### Functions

- def [DataTools.network\\_datatools.findnetworknodes](#) (network)
- def [DataTools.network\\_datatools.findaliases](#) (entitylist)
- def [DataTools.network\\_datatools.perceptronify](#) (network, indegree\_max=8)

## 7.5 ps\_datatools.py File Reference

### Namespaces

- [DataTools.ps\\_datatools](#)

### Functions

- def [DataTools.ps\\_datatools.ps\\_read\\_csv](#) (filename, ps\_format=True)
- def [DataTools.ps\\_datatools.ps\\_read\\_entitylist](#) (filename)
- def [DataTools.ps\\_datatools.ps\\_has\\_duplicates](#) (network)
- def [DataTools.ps\\_datatools.ps\\_return\\_duplicates](#) (network)
- def [DataTools.ps\\_datatools.ps\\_flag\\_duplicates](#) (network)
- def [DataTools.ps\\_datatools.ps\\_drop\\_duplicates](#) (network)
- def [DataTools.ps\\_datatools.ps\\_clean\\_ref](#) (network)
- def [DataTools.ps\\_datatools.ps\\_clean\\_sentence\\_tags](#) (s)
- def [DataTools.ps\\_datatools.ps\\_clear\\_end](#) (s)
- def [DataTools.ps\\_datatools.ps\\_replace\\_syntax](#) (s)
- def [DataTools.ps\\_datatools.ps\\_perceptronify](#) (in\_network, max\_indegree=8)
- def [DataTools.ps\\_datatools.ps\\_write\\_csv](#) (network, csv\_name='network\_please\_rename.csv', exclude\_↵ unknown=False)

# Index

- `__init__`
  - `DataTools::language_datatools::DictionaryTagger`, 24
- `__init__.py`, 27
- `closeconnections`
  - `DataTools::DB_datatools`, 9
- `DB_datatools.py`, 27
- `DataTools`, 9
- `DataTools.DB_datatools`, 9
- `DataTools.language_datatools`, 11
- `DataTools.language_datatools.DictionaryTagger`, 23
- `DataTools.network_datatools`, 15
- `DataTools.ps_datatools`, 16
- `DataTools::DB_datatools`
  - `closeconnections`, 9
  - `db_addsynonymbylist`, 10
  - `db_connect`, 10
  - `db_finddrug_bysynonym`, 10
  - `db_findrelations`, 10
  - `db_findtarget_bysynonym`, 10
  - `db_findtargetsynonyms`, 11
- `DataTools::language_datatools`
  - `network_score`, 11
  - `pos_tag`, 12
  - `sentence_score`, 12
  - `sentiment_score`, 13
  - `split`, 13
  - `weighted_sentence_score`, 14
  - `weighted_sentiment_score`, 14
- `DataTools::language_datatools::DictionaryTagger`
  - `__init__`, 24
  - `dictionary`, 25
  - `max_key_size`, 25
  - `tag`, 24
  - `tag_sentence`, 24
- `DataTools::network_datatools`
  - `findaliases`, 15
  - `findnetworknodes`, 15
  - `perceptronify`, 16
- `DataTools::ps_datatools`
  - `ps_clean_ref`, 16
  - `ps_clean_sentence_tags`, 17
  - `ps_clear_end`, 18
  - `ps_drop_duplicates`, 18
  - `ps_flag_duplicates`, 19
  - `ps_has_duplicates`, 19
  - `ps_perceptronify`, 19
  - `ps_read_csv`, 19
  - `ps_read_entitylist`, 20
  - `ps_replace_syntax`, 20
  - `ps_return_duplicates`, 20
  - `ps_write_csv`, 21
- `db_addsynonymbylist`
  - `DataTools::DB_datatools`, 10
- `db_connect`
  - `DataTools::DB_datatools`, 10
- `db_finddrug_bysynonym`
  - `DataTools::DB_datatools`, 10
- `db_findrelations`
  - `DataTools::DB_datatools`, 10
- `db_findtarget_bysynonym`
  - `DataTools::DB_datatools`, 10
- `db_findtargetsynonyms`
  - `DataTools::DB_datatools`, 11
- `dictionary`
  - `DataTools::language_datatools::DictionaryTagger`, 25
- `findaliases`
  - `DataTools::network_datatools`, 15
- `findnetworknodes`
  - `DataTools::network_datatools`, 15
- `language_datatools.py`, 27
- `max_key_size`
  - `DataTools::language_datatools::DictionaryTagger`, 25
- `network_datatools.py`, 28
- `network_score`
  - `DataTools::language_datatools`, 11
- `perceptronify`
  - `DataTools::network_datatools`, 16
- `pos_tag`
  - `DataTools::language_datatools`, 12
- `ps_clean_ref`
  - `DataTools::ps_datatools`, 16
- `ps_clean_sentence_tags`
  - `DataTools::ps_datatools`, 17
- `ps_clear_end`
  - `DataTools::ps_datatools`, 18
- `ps_datatools.py`, 28
- `ps_drop_duplicates`
  - `DataTools::ps_datatools`, 18
- `ps_flag_duplicates`
  - `DataTools::ps_datatools`, 19
- `ps_has_duplicates`

- DataTools::ps\_datatools, [19](#)
- ps\_perceptronify
  - DataTools::ps\_datatools, [19](#)
- ps\_read\_csv
  - DataTools::ps\_datatools, [19](#)
- ps\_read\_entitylist
  - DataTools::ps\_datatools, [20](#)
- ps\_replace\_syntax
  - DataTools::ps\_datatools, [20](#)
- ps\_return\_duplicates
  - DataTools::ps\_datatools, [20](#)
- ps\_write\_csv
  - DataTools::ps\_datatools, [21](#)
- sentence\_score
  - DataTools::language\_datatools, [12](#)
- sentiment\_score
  - DataTools::language\_datatools, [13](#)
- split
  - DataTools::language\_datatools, [13](#)
- tag
  - DataTools::language\_datatools::DictionaryTagger,  
[24](#)
- tag\_sentence
  - DataTools::language\_datatools::DictionaryTagger,  
[24](#)
- weighted\_sentence\_score
  - DataTools::language\_datatools, [14](#)
- weighted\_sentiment\_score
  - DataTools::language\_datatools, [14](#)