

# Package ‘flyio’

December 14, 2018

**Type** Package

**Title** Read or Write any Format from Anywhere

**Version** 0.1.0

**Author** Himanshu Sikaria [aut, cre]

**Maintainer** Himanshu Sikaria <himanshu.sikaria@socialcops.com>

**Description** Perform input, output of files in R from data sources like Google Cloud Storage ('GCS') <<https://cloud.google.com/storage/>>, Amazon Web Services ('AWS S3') <<https://aws.amazon.com/s3>> or local drive.

**URL** <https://github.com/socialcopsdev/flyio>

**BugReports** <https://github.com/socialcopsdev/flyio/issues>

**License** GPL-3 | file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Depends** R (>= 3.1.2)

**Imports** stringr, assertthat, googleCloudStorageR, aws.s3, utils, tools

**NeedsCompilation** no

## R topics documented:

|                                |    |
|--------------------------------|----|
| export_file . . . . .          | 2  |
| export_folder . . . . .        | 3  |
| export_raster . . . . .        | 3  |
| export_rda . . . . .           | 4  |
| export_rds . . . . .           | 5  |
| export_shp . . . . .           | 6  |
| export_table . . . . .         | 6  |
| file_exists . . . . .          | 7  |
| flyio_auth . . . . .           | 8  |
| flyio_get_bucket . . . . .     | 8  |
| flyio_get_datasource . . . . . | 9  |
| flyio_set_bucket . . . . .     | 9  |
| flyio_set_datasource . . . . . | 10 |
| import_file . . . . .          | 10 |

|                         |           |
|-------------------------|-----------|
| import_raster . . . . . | 11        |
| import_rda . . . . .    | 12        |
| import_rds . . . . .    | 12        |
| import_shp . . . . .    | 13        |
| import_table . . . . .  | 14        |
| list_files . . . . .    | 15        |
| <b>Index</b>            | <b>16</b> |

---

|             |   |
|-------------|---|
| export_file | <i>Upload a file from the local system to cloud</i> |
|-------------|---|

---

## Description

Write a local file to the cloud, S3 or GCS

## Usage

```
export_file(localfile, bucketpath, data_source = flyio_get_datasource(),
            bucket = flyio_get_bucket(data_source), ...)
```

## Arguments

|             |  |
|-------------|--|
| localfile   | path of the file to be uploaded  |
| bucketpath  | path where the file needs to be uploaded, the file name can or cannot be present |
| data_source | the name of the data source, if not set globally. gcs or s3                      |
| bucket      | the name of the bucket, if not set globally                                      |
| ...         | other parameters for gcs_upload or aws.s3::put_object                            |

## Value

the filename and path of the file in the bucket

## Examples

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
export_file("~/Downloads/dsada.csv", "tests/mtcars.csv")

## End(Not run)
```

---

|               |   |
|---------------|---|
| export_folder | <i>Upload a folder from the local system to cloud</i> |
|---------------|---|

---

**Description**

Write a local folder to the cloud, S3 or GCS

**Usage**

```
export_folder(localfolder, bucketpath, pattern = "*", overwrite = TRUE,
  data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), ...)
```

**Arguments**

|             |  |
|-------------|--|
| localfolder | path of the folder in which all the files are to be uploaded       |
| bucketpath  | path of the folder in which the files are to be uploaded           |
| pattern     | pattern of the file names in the folder to be uploaded             |
| overwrite   | if files need to be overwritten (if already present)               |
| data_source | the name of the data source, if not set globally. can be gcs or s3 |
| bucket      | the name of the bucket, if not set globally                        |
| ...         | other parameters for gcs/s3 upload                                 |

**Value**

the filename and path of the file in the bucket

**Examples**

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
export_folder("~/Downloads/", "tests/")

## End(Not run)
```

---

|               |                     |
|---------------|---------------------|
| export_raster | <i>Write raster</i> |
|---------------|---------------------|

---

**Description**

Write raster

**Usage**

```
export_raster(x, file, FUN = raster::writeRaster,
  data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), ...)
```

**Arguments**

|             |  |
|-------------|--|
| x           | variable name  |
| file        | path of the file to be written to                                  |
| FUN         | the function using which the file is to write                      |
| data_source | the name of the data source, if not set globally. s3, gcs or local |
| bucket      | the name of the bucket, if not set globally                        |
| ...         | other parameters for the FUN function defined above                |

**Value**

No output

**Examples**

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
export_raster(t, "tests/testras.tif", writeRaster, format = "GTiff")

## End(Not run)
```

---

export\_rda

---

*Write RDA files*


---

**Description**

Write R data RDA file to anywhere from R

**Usage**

```
export_rda(..., file, FUN = save, data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source))
```

**Arguments**

|             |  |
|-------------|--|
| ...         | R ojects need to be saved  |
| file        | path of the file to be written to                                  |
| FUN         | the function using which the file is to write                      |
| data_source | the name of the data source, if not set globally. s3, gcs or local |
| bucket      | the name of the bucket, if not set globally                        |

**Value**

No output

## Examples

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
export_rda(iris, mtcars, "tests/iris.rda")

## End(Not run)
```

---

export\_rds

*Write RDS files*

---

## Description

Write R data RDS file to anywhere from R

## Usage

```
export_rds(x, file, FUN = saveRDS,
  data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), ...)
```

## Arguments

|             |  |
|-------------|--|
| x           | variable name  |
| file        | path of the file to be written to                                  |
| FUN         | the function using which the file is to write                      |
| data_source | the name of the data source, if not set globally. s3, gcs or local |
| bucket      | the name of the bucket, if not set globally                        |
| ...         | other parameters for the FUN function defined above                |

## Value

if FUN returns anything

## Examples

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
export_rds(iris, "tests/iris.rds", saveRDS)

## End(Not run)
```

---

|            |                         |
|------------|-------------------------|
| export_shp | <i>Write shapefiles</i> |
|------------|-------------------------|

---

### Description

Write shapefiles

### Usage

```
export_shp(obj, pathshp, FUN = rgdal::writeOGR, dsnlayerbind = F,
  data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), ...)
```

### Arguments

|              |   |
|--------------|---|
| obj          | R object to be written  |
| pathshp      | the path of the shapefile, which may or may not include the extension |
| FUN          | the function using which the file is to be read                       |
| dsnlayerbind | if the FUN needs dsn and layer binded or not                          |
| data_source  | the name of the data source, if not set globally. s3, gcs or local    |
| bucket       | the name of the bucket, if not set globally                           |
| ...          | other parameters for the FUN function defined above                   |

### Value

output of the FUN function if any

### Examples

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
export_shp(t, "tests/shptest/", "new", driver = "ESRI Shapefile", overwrite = T)

## End(Not run)
```

---

|              |                                    |
|--------------|------------------------------------|
| export_table | <i>Write csv, Excel files, txt</i> |
|--------------|------------------------------------|

---

### Description

Write csv, Excel files, txt

### Usage

```
export_table(x, file, FUN = write.csv,
  data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), ...)
```

**Arguments**

|             |  |
|-------------|--|
| x           | variable name  |
| file        | path of the file to be written to                                  |
| FUN         | the function using which the file is to write                      |
| data_source | the name of the data source, if not set globally. s3, gcs or local |
| bucket      | the name of the bucket, if not set globally                        |
| ...         | other parameters for the FUN function defined above                |

**Value**

No output

**Examples**

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
export_table(iris, "tests/iris.csv", write.csv)

## End(Not run)
```

---

|             |                               |
|-------------|-------------------------------|
| file_exists | <i>Check if a file exists</i> |
|-------------|-------------------------------|

---

**Description**

Check if a file exists

**Usage**

```
file_exists(path, data_source = flyio_get_datasource(),
            bucket = flyio_get_bucket(data_source))
```

**Arguments**

|             |  |
|-------------|--|
| path        | the entire path for the file                                       |
| data_source | the name of the data source, if not set globally. s3, gsc or local |
| bucket      | the name of the bucket, if not set globally                        |

**Value**

logical. if the file exists or not

**Examples**

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
file_exists(path = "tests/mtcars.csv")

## End(Not run)
```

---

|            |                           |
|------------|---------------------------|
| flyio_auth | <i>Authenticate flyio</i> |
|------------|---------------------------|

---

### Description

Authenticate any of the cloud storage platforms to perform any I/O

### Usage

```
flyio_auth(auth_list = "", data_source = flyio_get_datasource(),
  scope = "https://www.googleapis.com/auth/devstorage.full_control")
```

### Arguments

|             |   |
|-------------|---|
| auth_list   | path to the json file or the system environment name in case of gcs. For s3 a vector for access_key, secret_access_key, region (optional; default us-east-1) and session_id (optional); this could also be a single comma-separated string. |
| data_source | default to local. Possible options : gcs, s3, local. Case insensitive   |
| scope       | the scope of the auth if gcs. Default: https://www.googleapis.com/auth/devstorage.full_control  |

### Examples

```
flyio_set_datasource("local")
flyio_auth()
```

---

|                  |   |
|------------------|---|
| flyio_get_bucket | <i>Get global bucket name for flyio</i> |
|------------------|---|

---

### Description

Get global bucket name to be used for all the functions in flyio

### Usage

```
flyio_get_bucket(data_source = flyio_get_datasource())
```

### Arguments

|             |  |
|-------------|--|
| data_source | the data source used for I/O. Default chooses the data source set using flyio_set_datasource() |
|-------------|--|

### Details

if the data source is local, then an empty string is returned

### Value

the string - bucket name stored



**Examples**

```
# first setting the bucket for a data source
flyio_set_bucket(bucket = "socialcops-test", data_source = "S3")
# retrieving the bucket for S3
flyio_get_bucket(data_source = "S3")
```

---

|                      |  |
|----------------------|--|
| flyio_get_datasource | <i>Get global data source name for flyio</i> |
|----------------------|--|

---

**Description**

Get global data source name to be used for all the functions in flyio. Returns the value stored using flyio\_set\_datasource

**Usage**

```
flyio_get_datasource()
```

**Value**

the string - data source name stored

**Examples**

```
# first setting the data source
flyio_set_datasource("s3")
# getting the data source
flyio_get_datasource()
```

---

|                  |   |
|------------------|---|
| flyio_set_bucket | <i>Set global bucket name for flyio</i> |
|------------------|---|

---

**Description**

Set global bucket name to be used for all the functions in flyio

**Usage**

```
flyio_set_bucket(bucket, data_source = flyio_get_datasource())
```

**Arguments**

|             |  |
|-------------|--|
| bucket      | the bucket name to be set  |
| data_source | the data source used for I/O. Default chooses the data source set using flyio_set_datasource() |

**Value**

stores the bucket name in a global environment under rioBucketGcs or rioBucketS3

**Examples**

```
flyio_set_bucket(bucket = "socialcops-test", data_source = "S3")
```

---

|                      |  |
|----------------------|--|
| flyio_set_datasource | <i>Set global data source name for flyio</i> |
|----------------------|--|

---

### Description

Set global data source name to be used for all the function in flyio

### Usage

```
flyio_set_datasource(data_source)
```

### Arguments

|             |                               |
|-------------|-------------------------------|
| data_source | the DataSource name to be set |
|-------------|-------------------------------|

### Value

stores the data source name in a global environment under rioDataSource

### Examples

```
flyio_set_datasource("local")
```

---

|             |   |
|-------------|---|
| import_file | <i>Download file from cloud to local system</i> |
|-------------|---|

---

### Description

Save a single file from the cloud to your local drive

### Usage

```
import_file(bucketpath, localfile, data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), overwrite = TRUE, ...)
```

### Arguments

|             |   |
|-------------|---|
| bucketpath  | path of file in the bucket  |
| localfile   | path where the file needs to be downloaded. The file name and extension also need to be present; if not, the current file name will be considered |
| data_source | the name of the data source, if not set globally, gcs or s3   |
| bucket      | the name of the bucket, if not set globally   |
| overwrite   | logical. If the files should be overwritten if already present  |
| ...         | other parameters for gcs_get_object or save_object  |

### Value

the filename and path of the object saved to local

**Examples**

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
import_file("mtcars.csv", "~/Downloads/dsada.csv", overwrite = T)

## End(Not run)
```

import\_raster

*Read raster files***Description**

Read raster data from anywhere using a function defined by you

**Usage**

```
import_raster(file, FUN = raster::raster,
  data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), ...)
```

**Arguments**

|             |  |
|-------------|--|
| file        | path of the file to be read  |
| FUN         | the function using which the file is to be read                    |
| data_source | the name of the data source, if not set globally. s3, gcs or local |
| bucket      | the name of the bucket, if not set globally                        |
| ...         | other parameters for the FUN function defined above                |

**Value**

the output of the FUN function

**Examples**

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
t = import_raster("tests/testras.tif", raster)

## End(Not run)
```

---

import\_rda

*Read RDA file*


---

### Description

Read RData or rda file from anywhere

### Usage

```
import_rda(file, FUN = load, data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), ...)
```

### Arguments

|             |  |
|-------------|--|
| file        | path of the file to be read  |
| FUN         | the function using which the file is to be read                    |
| data_source | the name of the data source, if not set globally. s3, gcs or local |
| bucket      | the name of the bucket, if not set globally                        |
| ...         | other parameters for the FUN function defined above                |

### Value

the output of the FUN function

### Examples

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
load("tests/googletest.rda")

## End(Not run)
```

---

import\_rds

*Read RDS file*


---

### Description

Read R data - RDS file from anywhere

### Usage

```
import_rds(file, FUN = readRDS, data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), ...)
```

**Arguments**

|             |  |
|-------------|--|
| file        | path of the file to be read  |
| FUN         | the function using which the file is to be read                    |
| data_source | the name of the data source, if not set globally. s3, gcs or local |
| bucket      | the name of the bucket, if not set globally                        |
| ...         | other parameters for the FUN function defined above                |

**Value**

the output of the FUN function

**Examples**

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
import_rds("tests/googletest.rds", readRDS)

## End(Not run)
```

---

import\_shp

---

*Read shapefiles*


---

**Description**

Read shapefiles data from anywhere using a function defined by you

**Usage**

```
import_shp(dsn, layer, FUN = rgdal::readOGR, dsnlayerbind = F,
  data_source = flyio_get_datasource(),
  bucket = flyio_get_bucket(data_source), ...)
```

**Arguments**

|              |  |
|--------------|--|
| dsn          | path of the file to be read  |
| layer        | the name of the shapefile without extension                        |
| FUN          | the function using which the file is to be read                    |
| dsnlayerbind | if the FUN needs dsn and layer binded or not                       |
| data_source  | the name of the data source, if not set globally. s3, gcs or local |
| bucket       | the name of the bucket, if not set globally                        |
| ...          | other parameters for the FUN function defined above                |

**Value**

the output of the FUN function

## Examples

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
t = import_shp("tests/shptest/", "testshp", FUN = readOGR, dsnlayerbind = F)
t = import_shp("tests/shptest/", "testshp", FUN = shapefile, dsnlayerbind = T)

## End(Not run)
```

---

|              |                                   |
|--------------|-----------------------------------|
| import_table | <i>Read csv, Excel files, txt</i> |
|--------------|-----------------------------------|

---

## Description

Read tabular data from anywhere using a function defined by you

## Usage

```
import_table(file, FUN = read.csv,
             data_source = flyio_get_datasource(),
             bucket = flyio_get_bucket(data_source), ...)
```

## Arguments

|             |  |
|-------------|--|
| file        | path of the file to be read  |
| FUN         | the function using which the file is to be read                    |
| data_source | the name of the data source, if not set globally. s3, gcs or local |
| bucket      | the name of the bucket, if not set globally                        |
| ...         | other parameters for the FUN function defined above                |

## Value

the output of the FUN function

## Examples

```
## Not run:
flyio_set_datasource("gcs")
flyio_set_bucket("socialcops-test")
import_table("tests/googletest.xlsx", read_excel)

## End(Not run)
```

---

|            |   |
|------------|---|
| list_files | <i>List the Files in a Directory/Folder</i> |
|------------|---|

---

**Description**

list the files in cloud or locally - similar to list.files()

**Usage**

```
list_files(path = "", pattern = NULL, recursive = FALSE,  
  ignore.case = FALSE, full.names = TRUE,  
  data_source = flyio_get_datasource(),  
  bucket = flyio_get_bucket(data_source))
```

**Arguments**

|             |  |
|-------------|--|
| path        | the folder for which the files need to be listed   |
| pattern     | an optional regular expression. Only file path names that match the regular expression will be returned. |
| recursive   | logical. Should the listing recurse into directories?  |
| ignore.case | logical. Should pattern-matching be case-insensitive?  |
| full.names  | logical. Should the entire path be returned or only after the path inputted?                             |
| data_source | the name of the data source, gcs, s3 or local; if not set globally                                       |
| bucket      | the name of the bucket, if not set globally  |

**Value**

a vector of full file names

**Examples**

```
## Not run:  
flyio_set_datasource("s3")  
flyio_set_bucket("socialcops-test")  
gcsListFiles(path = "tests/", pattern = ".*csv")  
  
## End(Not run)
```

# Index

`export_file`, [2](#)  
`export_folder`, [3](#)  
`export_raster`, [3](#)  
`export_rda`, [4](#)  
`export_rds`, [5](#)  
`export_shp`, [6](#)  
`export_table`, [6](#)  
  
`file_exists`, [7](#)  
`flyio_auth`, [8](#)  
`flyio_get_bucket`, [8](#)  
`flyio_get_datasource`, [9](#)  
`flyio_set_bucket`, [9](#)  
`flyio_set_datasource`, [10](#)  
  
`import_file`, [10](#)  
`import_raster`, [11](#)  
`import_rda`, [12](#)  
`import_rds`, [12](#)  
`import_shp`, [13](#)  
`import_table`, [14](#)  
  
`list_files`, [15](#)