

# Package ‘argo’

May 7, 2026

**Type** Package

**Title** Accurate Estimation of Influenza Epidemics using Google Search Data

**Version** 3.0.3

**Date** 2025-11-21

**Author** Shaoyang Ning [aut],  
Shihao Yang [aut, cre],  
S. C. Kou [aut]

**Maintainer** Shihao Yang <shihao.yang@isye.gatech.edu>

**Description** Augmented Regression with General Online data (ARGO) for accurate estimation of influenza epidemics in United States on national level, regional level and state level. It replicates the method introduced in paper Yang, S., Santilana, M. and Kou, S.C. (2015) <[doi:10.1073/pnas.1515373112](https://doi.org/10.1073/pnas.1515373112)>; Ning, S., Yang, S. and Kou, S.C. (2019) <[doi:10.1038/s41501-019-41559-6](https://doi.org/10.1038/s41501-019-41559-6)>; Yang, S., Ning, S. and Kou, S.C. (2021) <[doi:10.1038/s41598-021-83084-5](https://doi.org/10.1038/s41598-021-83084-5)>.

**License** GPL-2

**Imports** xts, glmnet, zoo, XML, xtable, Matrix, boot

**Suggests** testthat

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2025-11-23 16:50:02 UTC

## Contents

argo . . . . .	2
argo2 . . . . .	3
argo2_main . . . . .	4
argox_main . . . . .	5
argo_main . . . . .	6
bootstrap_relative_efficiency . . . . .	6

boot_re . . . . .	8
gt.parser.pub.api . . . . .	9
gt.parser.pub.web . . . . .	9
heatmap_argo . . . . .	10
heatmap_cor . . . . .	11
load_data . . . . .	11
load_reg_data . . . . .	13
logit . . . . .	14
logit_inv . . . . .	14
parse_gt_weekly . . . . .	15
parse_unrevised_ili . . . . .	15
plot_argo . . . . .	16
summary_argo . . . . .	17

## Index 19

---

argo	<i>Construct ARGO object</i>
------	------------------------------

---

### Description

Wrapper for ARGO. The real work horse is glmnet package and/or linear model.

### Usage

```
argo(
  data,
  exogen = xts::xts(NULL),
  N_lag = 1:52,
  N_training = 104,
  alpha = 1,
  use_all_previous = FALSE,
  mc.cores = 1,
  schedule = list()
)
```

### Arguments

data	response variable as xts, last element can be NA. If the response is later revised, it should be an xts that resembles upper triangular square matrix, with each column being the data available as of date of column name
exogen	exogenous predictors, default is NULL
N_lag	vector of the AR model lags used, if NULL then no AR lags will be used
N_training	number of training points, if use_all_previous is true, this is the least number of training points required
alpha	penalty between lasso and ridge, alpha=1 represents lasso, alpha=0 represents ridge, alpha=NA represents no penalty

<code>use_all_previous</code>	boolean variable indicating whether to use "all available data" (when TRUE) or "a sliding window" (when FALSE) for training
<code>mc.cores</code>	number of cores to compute argo in parallel
<code>schedule</code>	list to specify prediction schedule. Default to have <code>y_gap</code> as 1, and <code>forecast</code> as 0, i.e., nowcasting with past week ILI available from CDC.

### Details

This function takes the time series and exogenous variables (optional) as input, and produces out-of-sample prediction for each time point.

### Value

A list of following named objects

- `pred` An xts object with the same index as input, which contains historical nowcast estimation
- `coef` A matrix contains historical coefficient values of the predictors.
- `parm` Parameter values passed to argo function.
- `penalfac` the value of lambda ratio selected by cross-validation, NULL if `lamid` is NULL or has only one level.
- `penalregion` the lambda ratios that has a cross validation error within one standard error of minimum cross validation error

### References

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. *Proceedings of the National Academy of Sciences*. <doi:10.1073/pnas.1515373112>.

### Examples

```
GFT_xts <- xts::xts(exp(matrix(rnorm(180), ncol=1)), order.by = Sys.Date() - (180:1))
randomx <- xts::xts(exp(matrix(rnorm(180*100), ncol=100)), order.by = Sys.Date() - (180:1))

argo_result1 <- argo(GFT_xts)
argo_result2 <- argo(GFT_xts, exogen = randomx)
```

---

argo2

*ARGO second step*

---

### Description

Wrapper for ARGO second step. Best linear predictor / Bayesian posterior

### Usage

```
argo2(truth, argo1.p, argo.nat.p)
```

**Arguments**

truth	prediction target
argo1.p	argo first step prediction
argo.nat.p	argo national level prediction

**References**

Shaoyang Ning, Shihao Yang, S. C. Kou. Accurate Regional Influenza Epidemics Tracking Using Internet Search Data. Scientific Reports

**Examples**

```
truth <- xts::xts(exp(matrix(rnorm(180*10), ncol=10)), order.by = Sys.Date() - (180:1))
argo1.p <- xts::xts(exp(matrix(rnorm(180*10), ncol=10)), order.by = Sys.Date() - (180:1))
argo.nat.p <- xts::xts(exp(matrix(rnorm(180*10), ncol=10)), order.by = Sys.Date() - (180:1))
argo2result <- argo2(truth, argo1.p, argo.nat.p)
```

---

argo2_main	<i>main function for argo2</i>
------------	--------------------------------

---

**Description**

main function that reproduce the results in ARGO2 paper

**Usage**

```
argo2_main(
  gt.folder,
  ili.folder,
  population.file,
  gft.file,
  save.folder = NULL
)
```

**Arguments**

gt.folder	folder with Google Trends files, which should be thousands of csv file such as "US-MA_fever cough.csv" or "US-NY_cold or flu.csv"
ili.folder	folder with ILINet data files: "ILINet_nat.csv" and "ILINet_regional.csv"
population.file	file path to population csv file
gft.file	file path to Google Flu Trends csv file
save.folder	output folder to save graphics. If NULL then do not output graphics.

## References

Shaoyang Ning, Shihao Yang, S. C. Kou. Accurate Regional Influenza Epidemics Tracking Using Internet Search Data. Scientific Reports

## Examples

```
## Not run:
download.file("https://zenodo.org/records/17681160/files/GT2016-10-24.zip",
file.path(tempdir(), "gt2016-10-24.zip"))
unzip(file.path(tempdir(), "gt2016-10-24.zip"), exdir = tempdir())
gt.folder <- file.path(tempdir(), "2016-10-19")
argox_main(
  gt.folder=gt.folder,
  ili.folder=system.file("regiondata", "ili20161121", package = "argox"),
  population.file=system.file("regiondata", "Population.csv", package = "argox"),
  gft.file=system.file("regiondata", "GFT.txt", package = "argox")
)

## End(Not run)
```

---

argox\_main

*main function for argox*

---

## Description

Main function that reproduce the results in ARGOX paper. The datasets are available at Harvard Dataverse <doi:10.7910/DVN/2IVDGGK>.

## Usage

```
argox_main(
  gt.folder,
  ili.folder,
  population.file,
  gft.file,
  mix,
  save.folder = NULL,
  NCORES = 8
)
```

## Arguments

gt.folder	folder with Google Trends files, which should be thousands of csv file such as "US-MA_fever cough.csv" or "US-NY_cold or flu.csv"
ili.folder	folder with ILINet data files: "ILINet_nat.csv" and "ILINet_regional.csv"
population.file	file path to population csv file

<code>gft.file</code>	file path to Google Flu Trends csv file
<code>mix</code>	the weighted average mixing of raw state-level Google Trends data. Set to be 0 for stand-alone model. Set to be 1/3 for spatial-pooling model.
<code>save.folder</code>	output folder to save graphics. If NULL then do not output graphics.
<code>NCORES</code>	number of parallel cpu cores to be used.

## References

Yang, S., Ning, S. & Kou, S.C. Use Internet search data to accurately track state level influenza epidemics. *Sci Rep* 11, 4023 (2021)

---

<code>argo_main</code>	<i>main function for argo</i>
------------------------	-------------------------------

---

## Description

main function that reproduce the results in ARGO paper

## Usage

```
argo_main(save.folder = NULL)
```

## Arguments

`save.folder` output folder to save graphics. If NULL then do not output graphics.

## Examples

```
argo_main()
```

---

<code>bootstrap_relative_efficiency</code>	<i>bootstrap relative efficiency confidence interval</i>
--	--

---

## Description

This function is used to reproduce the ARGO bootstrap confidence interval

**Usage**

```
bootstrap_relative_efficiency(
  pred_data,
  model_good,
  model_bench,
  l = 50,
  N = 10000,
  truth = "CDC.data",
  sim = "geom",
  conf = 0.95,
  type = c("mse", "mape", "mae", "mspe", "rmse", "rmspe")
)
```

**Arguments**

pred_data	A matrix that contains the truth vector and the predictions. It can be data.frame or xts object
model_good	The model to evaluate, must be in the column names of pred_data
model_bench	The model to compare to, must be in the column names of pred_data
l	stationary bootstrap mean block length
N	number of bootstrap samples
truth	the column name of the truth
sim	simulation method, pass to boot::tsboot
conf	confidence level
type	Must be one of "mse" (mean square error), "mape" (mean absolute percentage error), or "mae" (mean absolute error)

**Value**

A vector of point estimate and corresponding bootstrap confidence interval

**Examples**

```
GFT_xts = xts::xts(exp(matrix(rnorm(1000), ncol=5)), order.by = Sys.Date() - (200:1))
names(GFT_xts) <- paste0("col", 1:ncol(GFT_xts))
names(GFT_xts)[1] <- "CDC.data"
bootstrap_relative_efficiency(
  pred_data = GFT_xts,
  model_good = "col2",
  model_bench = "col3",
  truth="CDC.data",
  N = 100
)
```

---

 boot\_re

*wrapper for bootstrap relative efficiency confidence interval*


---

### Description

This function is used to wrap the `bootstrap_relative_efficiency`, taking vectorized arguments.

### Usage

```
boot_re(
  pred_data,
  period.all,
  model_good,
  bench.all,
  type,
  truth = "CDC.data",
  l = 50,
  N = 10000,
  sim = "geom",
  conf = 0.95
)
```

### Arguments

<code>pred_data</code>	A matrix that contains the truth vector and the predictions. It can be <code>data.frame</code> or <code>xts</code> object
<code>period.all</code>	vector of the periods to evaluate relative efficiency
<code>model_good</code>	The model to evaluate, must be in the column names of <code>pred_data</code>
<code>bench.all</code>	vector of the models to compare to, must be in the column names of <code>pred_data</code>
<code>type</code>	Must be one of "mse" (mean square error), "mape" (mean absolute percentage error), or "mae" (mean absolute error)
<code>truth</code>	the column name of the truth
<code>l</code>	stationary bootstrap mean block length
<code>N</code>	number of bootstrap samples
<code>sim</code>	simulation method, pass to <code>boot::tsboot</code>
<code>conf</code>	confidence level

### Value

A vector of point estimate and corresponding bootstrap confidence interval

**Examples**

```
GFT_xts = xts::xts(exp(matrix(rnorm(500), ncol=5)), order.by = Sys.Date() - (100:1))
names(GFT_xts) <- paste0("col", 1:ncol(GFT_xts))
names(GFT_xts)[1] <- "CDC.data"

boot_re(
  pred_data = GFT_xts,
  period.all = c(paste0(zoo::index(GFT_xts)[1], "/", zoo::index(GFT_xts)[50]),
                paste0(zoo::index(GFT_xts)[51], "/", zoo::index(GFT_xts)[100])),
  model_good = "col2",
  bench.all = c("col3", "col4"),
  type = "mse",
  truth="CDC.data",
  l = 5,
  N = 20
)
```

---

gt.parser.pub.api      *Parsing each Google Trends file downloaded from Google Trends API*

---

**Description**

Parsing each Google Trends file downloaded from Google Trends API

**Usage**

```
gt.parser.pub.api(gt.folder, f)
```

**Arguments**

gt.folder	folder that contains Google Trends file
f	filename for Google Trends file

---

gt.parser.pub.web      *Parsing each Google Trends file downloaded from website*

---

**Description**

Parsing each Google Trends file downloaded from website

**Usage**

```
gt.parser.pub.web(gt.folder, f)
```

**Arguments**

gt.folder	folder that contains Google Trends file
f	filename for Google Trends file

---

heatmap_argo	<i>Heatmap plot of ARGO coefficients applied on CDC's ILI data</i>
--------------	--

---

**Description**

Heatmap plot of ARGO coefficients applied on CDC's ILI data

**Usage**

```
heatmap_argo(argo_coef, lim = 0.1, na.grey = TRUE, scale = 1)
```

**Arguments**

argo_coef	The coefficient matrix
lim	the limit to truncate for large coefficients for better presentation
na.grey	whether to plot grey for NA values
scale	margin scale

**Value**

a graph on the default plot window

**Examples**

```
cor_coef <- matrix(runif(100, -1, 1), ncol=10)
colnames(cor_coef) <- as.character(Sys.Date() - 10:1)
rownames(cor_coef) <- paste0("row", 1:10)
pdf(file.path(tempdir(), "heatmap_argo.pdf"), height=11,width=12)
heatmap_argo(cor_coef)
dev.off()
```

---

heatmap_cor	<i>Heatmap plot of correlation matrix</i>
-------------	---

---

**Description**

Heatmap plot of correlation matrix

**Usage**

```
heatmap_cor(cor_heat, lim = 1)
```

**Arguments**

cor_heat	The coefficient matrix to draw heatmap
lim	the limit to truncate for large coefficients for better presentation

**Value**

a graph on the default plot window

**Examples**

```
cor_coef <- matrix(runif(100, -1, 1), ncol=10)
colnames(cor_coef) <- paste0("col", 1:10)
rownames(cor_coef) <- paste0("row", 1:10)
heatmap_cor(cor_coef)
```

---

load_data	<i>Parsing of raw data</i>
-----------	----------------------------

---

**Description**

Data related to the PNAS paper. Accessed on Nov 14, 2015.

**Usage**

```
load_data(type = "extdata", ili.weighted = TRUE)
```

**Arguments**

type	the type of the data to be loaded. If type=="extdata" it loads the data to reproduce the PNAS paper, and if type=="athdata" it loads the data to reproduce the CID(?) paper.
ili.weighted	logical indicator to specify whether to load weighted ILI or not, if FALSE un-weighted ILI is loaded.

## Details

Parse and load CDC's ILI data, Google Flu Trend data, Google Correlate data trained with ILI as of 2010, Google Correlate data trained with ILI as of 2009, Google Trend data with search terms identified from Google Correlate (2010 version).

Each week ends on the Saturday indicated in the xts object

Google Correlate data is standardized by Google, and we rescale it to 0 – 100 during parsing. Google Trends data is in the scale of 0 – 100.

## Value

A list of following named xts objects if type=="extdata"

- GC10 Google Correlate trained with ILI available as of 2010. Google Correlate has been deprecated by Google as of Dec 2019 and is no longer publicly available.
- GC09 Google Correlate trained with ILI available as of 2009.
- GT Google Trends data for search queries identified using Google Correlate. Not directly available online, you have to manually input query terms at <https://trends.google.com/trends/>
- CDC CDC's ILI dataset. Available online at <https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html>
- GFT Google Flu Trend (historical predictions).

A list of following named xts objects if type=="athdata"

- GT Google Trends data for search queries identified. Not directly available online, you have to manually input query terms at <https://trends.google.com/trends/>
- CDC CDC's ILI dataset. Available online at <https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html>
- ili\_idx the indexing information that includes the week number and year number, the date of ending Saturday, and the season number Available online at <https://www.cdc.gov/fluview/index.html>
- ATH Athenahealth data that includes the proportion of "Flu Visit", "ILI Visit", and "Unspecified Viral or ILI Visit" compared to total number of visit to the Athenahealth partner healthcare providers.
- ili\_unrevised Historical unrevised ILI activity level. The unrevised ILI published on week ZZ of season XXXX-YYYY is available at [www.cdc.gov/flu/weekly/weeklyarchivesXXXX-YYYY/data/senAllregor.htm](http://www.cdc.gov/flu/weekly/weeklyarchivesXXXX-YYYY/data/senAllregor.htm). Historical data is also available at <https://github.com/cdcepi/FluSight-forecast-hub>.

## References

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. Proceedings of the National Academy of Sciences. <doi:10.1073/pnas.1515373112>.

## Examples

```
system.file("extdata", "correlate-Influenza_like_Illness_h1n1_CDC_.csv", package = "argo")
system.file("extdata", "correlate-Influenza_like_Illness_CDC_.csv", package = "argo")
system.file("extdata", "GFT.csv", package = "argo")
system.file("extdata", "ILINet.csv", package = "argo")
load_data()
```

---

load\_reg\_data

*Parsing of raw data for regional ILI estimation*

---

## Description

Parsing of raw data for regional ILI estimation

## Usage

```
load_reg_data(
  gt.folder,
  ili.folder,
  population.file,
  gft.file,
  gt.parser = gt.parser.pub.web
)
```

## Arguments

gt.folder	folder with all Google Trends data
ili.folder	folder with all ILI data
population.file	csv file path with state population data
gft.file	csv file path for Google Flu Trends
gt.parser	Google Trends data parser function, could be 'gt.parser.pub.web' or 'gt.parser.pub.api'

## References

Shaoyang Ning, Shihao Yang, S. C. Kou. Accurate Regional Influenza Epidemics Tracking Using Internet Search Data. Scientific Reports

## Examples

```
download.file("https://zenodo.org/records/17681160/files/GT2016-10-24.zip",
file.path(tempdir(), "gt2016-10-24.zip"))
unzip(file.path(tempdir(), "gt2016-10-24.zip"), exdir = tempdir())
gt.folder <- file.path(tempdir(), "2016-10-19")

data_parsed <- load_reg_data(
```

```
gt.folder=gt.folder,  
ili.folder=system.file("regiondata", "ili20161121", package = "argo"),  
population.file=system.file("regiondata", "Population.csv", package = "argo"),  
gft.file=system.file("regiondata", "GFT.txt", package = "argo")  
)
```

---

logit	<i>logit function</i>
-------	-----------------------

---

**Description**

logit function

**Usage**

```
logit(x)
```

**Arguments**

x                    numeric value for logit transformation

**Examples**

```
logit(0.5)
```

---

logit_inv	<i>inverse logit function</i>
-----------	-------------------------------

---

**Description**

inverse logit function

**Usage**

```
logit_inv(x)
```

**Arguments**

x                    numeric value for inverse logit transformation

**Examples**

```
logit_inv(0)
```

---

parse\_gt\_weekly      *Parsing of Google Trends data*

---

**Description**

Parsing of Google Trends data

**Usage**

```
parse_gt_weekly(folder)
```

**Arguments**

folder              folder with weekly Google Trends file

**References**

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. *Proceedings of the National Academy of Sciences*. <doi:10.1073/pnas.1515373112>.

**Examples**

```
download.file("https://zenodo.org/records/17681160/files/GT2016-10-24.zip",
file.path(tempdir(), "gt2016-10-24.zip"))
unzip(file.path(tempdir(), "gt2016-10-24.zip"), exdir = tempdir())
gt.folder <- file.path(tempdir(), "2016-10-19")
parsed_data <- parse_gt_weekly(gt.folder)
```

---

parse\_unrevised\_ili      *Parsing of unrevised ili from online source*

---

**Description**

Parsing of unrevised ili from online source

**Usage**

```
parse_unrevised_ili(type = "extdata", ili.weighted = TRUE)
```

**Arguments**

type                  the type of data folder to parse  
ili.weighted          indicator to use weighted ILI or not

## References

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. Proceedings of the National Academy of Sciences. <doi:10.1073/pnas.1515373112>.

## Examples

```
parse_unrevised_ili()
```

---

plot\_argo

*Time series plot of ARGO applied on CDC's ILI data*

---

## Description

This function is used to reproduce the ARGO plot.

## Usage

```
plot_argo(GFT_xts, GC_GT_cut_date, model_names, legend_names, zoom_periods)
```

## Arguments

GFT_xts	dataframe with all predicted values
GC_GT_cut_date	cutting date for switching datasets
model_names	name of predicting models
legend_names	legend for predicting models
zoom_periods	vector of periods to zoom into

## Value

a graph on the default plot window

## Examples

```
GFT_xts = xts::xts(exp(matrix(rnorm(1000), ncol=5)), order.by = Sys.Date() - (200:1))
names(GFT_xts) <- paste0("col", 1:ncol(GFT_xts))
names(GFT_xts)[1] <- "CDC.data"
zoom_periods = c()
for (i in 0:5){
  zoom_periods = c(
    zoom_periods,
    paste0(zoo::index(GFT_xts)[i*30+1], "/", zoo::index(GFT_xts)[i*30+30])
  )
}
plot_argo(
```

```
GFT_xts = GFT_xts,  
GC_GT_cut_date = zoo::index(GFT_xts)[50],  
model_names = colnames(GFT_xts)[-1],  
legend_names = paste0(colnames(GFT_xts)[-1], "legend"),  
zoom_periods = zoom_periods  
)
```

---

summary\_argo

*performance summary of ARGO applied on CDC's ILI data*

---

## Description

performance summary of ARGO applied on CDC's ILI data

## Usage

```
summary_argo(  
  GFT_xts,  
  model_names,  
  legend_names,  
  periods,  
  whole_period = "2009-03/2015-10"  
)
```

## Arguments

GFT_xts	dataframe with all predicted values
model_names	name of predicting models
legend_names	legend for predicting models
periods	vector of periods to zoom into
whole_period	the whole period duration

## Value

A list of summary tables for the input periods, including RMSE, MAE, MAPE, corr

## References

Yang, S., Santillana, M., & Kou, S. C. (2015). Accurate estimation of influenza epidemics using Google search data via ARGO. *Proceedings of the National Academy of Sciences*. <doi:10.1073/pnas.1515373112>. Shaoyang Ning, Shihao Yang, S. C. Kou. Accurate Regional Influenza Epidemics Tracking Using Internet Search Data. *Scientific Reports*

**Examples**

```
GFT_xts = xts::xts(exp(matrix(rnorm(1000), ncol=10)), order.by = Sys.Date() - (100:1))
names(GFT_xts) <- paste0("col", 1:10)
names(GFT_xts)[1] <- "CDC.data"
summary_argo(
  GFT_xts = GFT_xts,
  model_names = colnames(GFT_xts)[-1],
  legend_names = paste0(colnames(GFT_xts)[-1], "legend"),
  periods = c(paste0(zoo::index(GFT_xts)[1], "/", zoo::index(GFT_xts)[49]),
              paste0(zoo::index(GFT_xts)[50], "/", zoo::index(GFT_xts)[100])),
  whole_period="2009-03/"
)
```

# Index

argo, [2](#)  
argo2, [3](#)  
argo2\_main, [4](#)  
argo\_main, [6](#)  
argox\_main, [5](#)

boot\_re, [8](#)  
bootstrap\_relative\_efficiency, [6](#)

gt.parser.pub.api, [9](#)  
gt.parser.pub.web, [9](#)

heatmap\_argo, [10](#)  
heatmap\_cor, [11](#)

load\_data, [11](#)  
load\_reg\_data, [13](#)  
logit, [14](#)  
logit\_inv, [14](#)

parse\_gt\_weekly, [15](#)  
parse\_unrevised\_ili, [15](#)  
plot\_argo, [16](#)

summary\_argo, [17](#)