

# Package: gainML (via r-universe)

September 1, 2024

**Type** Package

**Date** 2019-06-25

**Title** Machine Learning-Based Analysis of Potential Power Gain from  
Passive Device Installation on Wind Turbine Generators

**Version** 0.1.0

**Author** Hoon Hwangbo [aut, cre], Yu Ding [aut], Daniel Cabezon [aut],  
Texas A&M University [cph], EDP Renewables [cph]

**Maintainer** Hoon Hwangbo <hhwangb1@utk.edu>

**Copyright** Copyright (c) 2019 Y. Ding, H. Hwangbo, Texas A&M  
University, D. Cabezon, and EDP Renewables

**Description** Provides an effective machine learning-based tool that  
quantifies the gain of passive device installation on wind  
turbine generators. H. Hwangbo, Y. Ding, and D. Cabezon (2019)  
<[arXiv:1906.05776](https://arxiv.org/abs/1906.05776)>.

**Depends** R (>= 3.6.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** fields (>= 9.0), FNN (>= 1.1), utils, stats

**RoxygenNote** 6.1.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Repository** <https://hhwangbo.r-universe.dev>

**RemoteUrl** <https://github.com/hhwangbo/gainml>

**RemoteRef** HEAD

**RemoteSha** 2852835bbaefd6e5f4a28a42a56839c092a3d5e0

## Contents

analyze.gain . . . . .	2
analyze.p1 . . . . .	5
analyze.p2 . . . . .	6
arrange.data . . . . .	7
bootstrap.gain . . . . .	9
pw.freq . . . . .	12
quantify.gain . . . . .	12
wtg . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

analyze.gain	<i>Analyze Potential Gain from Passive Device Installation on WTGs by Using a Machine Learning-Based Tool</i>
--------------	---

---

### Description

Implements the gain analysis as a whole; this includes data arrangement, period 1 analysis, period 2 analysis, and gain quantification.

### Usage

```
analyze.gain(df1, df2, df3, p1.beg, p1.end, p2.beg, p2.end, ratedPW, AEP,
             pw.freq, freq.id = 3, time.format = "%Y-%m-%d %H:%M:%S",
             k.fold = 5, col.time = 1, col.turb = 2, bootstrap = NULL,
             free.sec = NULL, neg.power = FALSE)
```

### Arguments

df1	A dataframe for reference turbine data. This dataframe must include five columns: timestamp, turbine id, wind direction, power output, and air density.
df2	A dataframe for baseline control turbine data. This dataframe must include four columns: timestamp, turbine id, wind speed, and power output.
df3	A dataframe for neutral control turbine data. This dataframe must include four columns and have the same structure with df2.
p1.beg	A string specifying the beginning date of period 1. By default, the value needs to be specified in ‘%Y-%m-%d’ format, for example, ‘2014-10-24’. A user can use a different format as long as it is consistent with the format defined in time.format below.
p1.end	A string specifying the end date of period 1. For example, if the value is ‘2015-10-24’, data observed until ‘2015-10-23 23:50:00’ would be considered for period 1.
p2.beg	A string specifying the beginning date of period 2.
p2.end	A string specifying the end date of period 2. Defined similarly as p1.end.

ratedPW	A kW value that describes the (common) rated power of the selected turbines (REF and CTR-b).
AEP	A kWh value describing the annual energy production from a single turbine.
pw.freq	A matrix or a dataframe that includes power output bins and corresponding frequency in terms of the accumulated hours during an annual period.
freq.id	An integer indicating the column number of pw.freq that describes the frequency of power bins in terms of the accumulated hours during an annual period. By default, this parameter is set to 3.
time.format	A string describing the format of time stamps used in the data to be analyzed. The default value is '%Y-%m-%d %H:%M:%S'.
k.fold	An integer defining the number of data folds for the period 1 analysis and prediction. In the period 1 analysis, $k$ -fold cross validation (CV) will be applied to choose the optimal set of covariates that results in the least prediction error. The value of k.fold corresponds to the $k$ of the $k$ -fold CV. The default value is 5.
col.time	An integer specifying the column number of time stamps in wind turbine datasets. The default value is 1.
col.turb	An integer specifying the column number of turbines' id in wind turbine datasets. The default value is 2.
bootstrap	An integer indicating the current replication (run) number of bootstrap. If set to NULL, bootstrap is not applied. The default is NULL. A user is not recommended to set this value and directly run bootstrap; instead, use <a href="#">bootstrap.gain</a> to run bootstrap.
free.sec	A list of vectors defining free sectors. Each vector in the list has two scalars: one for starting direction and another for ending direction, ordered clockwise. For example, a vector of <code>c(310, 50)</code> is a valid component of the list. By default, this is set to NULL.
neg.power	Either TRUE or FALSE, indicating whether or not to use data points with a negative power output, respectively, in the analysis. The default value is FALSE, i.e., negative power output data will be eliminated.

### Details

Builds a machine learning model for a REF turbine (device installed) and a baseline CTR turbine (CTR-b; without device installation and preferably closest to the REF turbine) by using data measurements from a neutral CTR turbine (CTR-n; without device installation). Gain is quantified by evaluating predictions from the machine learning models and their differences during two different time periods, namely, period 1 (without device installation on the REF turbine) and period 2 (device installed on the REF turbine).

### Value

The function returns a list of several objects (lists) that includes all the analysis results from all steps.

`data` A list of arranged datasets including period 1 and period 2 data as well as  $k$ -folded training and test datasets generated from the period 1 data. See also [arrange.data](#).

`p1.res` A list containing period 1 analysis results. This includes the optimal set of predictor variables, period 1 prediction for the REF turbine and CTR-b turbine, the corresponding error measures such as RMSE and BIAS, and BIAS curves for both REF and CTR-b turbine models; see [analyze.p1](#) for the details.

`p2.res` A list containing period 2 analysis results. This includes period 2 prediction for the REF turbine and CTR-b turbine. See also [analyze.p2](#).

`gain.res` A list containing gain quantification results. This includes effect curve, offset curve, and gain curve as well as the measures of effect (gain without offset), offset, and (the final) gain; see [quantify.gain](#) for the details.

### Note

- This function will execute four other functions in sequence, namely, [arrange.data](#), [analyze.p1](#), [analyze.p2](#), [quantify.gain](#).
- A user can alternatively run the four functions by calling them individually in sequence.

### References

H. Hwangbo, Y. Ding, and D. Cabezon, 'Machine Learning Based Analysis and Quantification of Potential Power Gain from Passive Device Installation,' arXiv:1906.05776 [stat.AP], Jun. 2019. <https://arxiv.org/abs/1906.05776>.

### See Also

[arrange.data](#), [analyze.p1](#), [analyze.p2](#), [quantify.gain](#)

### Examples

```
df.ref <- with(wtg, data.frame(time = time, turb.id = 1, wind.dir = D,
  power = y, air.dens = rho))
df.ctrb <- with(wtg, data.frame(time = time, turb.id = 2, wind.spd = V,
  power = y))
df.ctrn <- df.ctrb
df.ctrn$turb.id <- 3

# For Full Sector Analysis
res <- analyze.gain(df.ref, df.ctrb, df.ctrn, p1.beg = '2014-10-24',
  p1.end = '2014-10-25', p2.beg = '2014-10-25', p2.end = '2014-10-26',
  ratedPW = 1000, AEP = 300000, pw.freq = pw.freq, k.fold = 2)
# In practice, one may use annual data for each of period 1 and period 2 analysis.
# One may typically use k.fold = 5 or 10.

# For Free Sector Analysis
free.sec <- list(c(310, 50), c(150, 260))

res <- analyze.gain(df.ref, df.ctrb, df.ctrn, p1.beg = '2014-10-24',
  p1.end = '2014-10-25', p2.beg = '2014-10-25', p2.end = '2014-10-26',
  ratedPW = 1000, AEP = 300000, pw.freq = pw.freq, k.fold = 2,
  free.sec = free.sec)
```

```
gain.res <- res$gain.res
gain.res$gain #This will provide the final gain value.
```

---

analyze.p1

*Apply Period 1 Analysis*


---

## Description

Conducts period 1 analysis; selects the optimal set of variables that minimizes a  $k$ -fold CV error measure and establishes a machine learning model that predicts power output of REF and CTR-b turbines by using period 1 data.

## Usage

```
analyze.p1(train, test, ratedPW)
```

## Arguments

train	A list containing $k$ datasets that will be used to train the machine learning model.
test	A list containing $k$ datasets that will be used to test the machine learning model and calculate CV error measures.
ratedPW	A kW value that describes the (common) rated power of the selected turbines (REF and CTR-b).

## Value

The function returns a list containing period 1 analysis results as follows.

opt.cov	A character vector presenting the names of predictor variables chosen for the optimal set.
pred.REF	A list of $k$ datasets each representing the $k$ th fold's period 1 prediction for the REF turbine.
pred.CTR	A list of $k$ datasets each representing the $k$ th fold's period 1 prediction for the CTR-b turbine.
err.REF	A data frame containing $k$ -fold CV based RMSE values and BIAS values for the REF turbine model (so $k$ of them for both). The first column includes the RMSE values and the second column includes the BIAS values.
err.CTR	A data frame containing $k$ -fold CV based RMSE values and BIAS values for the CTR-b turbine model. Similarly structured with err.REF.
biasCurve.REF	A $k$ by $m$ matrix describing the binned BIAS (technically speaking, 'residuals' which are the negative BIAS) curve for the REF turbine model, where $m$ is the number of power bins.
biasCurve.CTR	A $k$ by $m$ matrix describing the binned BIAS curve for the CTR-b turbine model.

**Note****VERY IMPORTANT!**

- Selecting the optimal set of variables will take a significant amount of time. For example, with a typical size of an annual dataset, the evaluation of one set of variables for a single fold testing may take about 20-40 minutes (from the authors' experience).
- To help understand the progress of the selection, some informative messages will be displayed while this function runs.

**References**

H. Hwangbo, Y. Ding, and D. Cabezon, 'Machine Learning Based Analysis and Quantification of Potential Power Gain from Passive Device Installation,' arXiv:1906.05776 [stat.AP], Jun. 2019. <https://arxiv.org/abs/1906.05776>.

**Examples**

```
df.ref <- with(wtg, data.frame(time = time, turb.id = 1, wind.dir = D,
  power = y, air.dens = rho))
df.ctrb <- with(wtg, data.frame(time = time, turb.id = 2, wind.spd = V,
  power = y))
df.ctrn <- df.ctrb
df.ctrn$turb.id <- 3

data <- arrange.data(df.ref, df.ctrb, df.ctrn, p1.beg = '2014-10-24',
  p1.end = '2014-10-25', p2.beg = '2014-10-25', p2.end = '2014-10-26',
  k.fold = 2)

p1.res <- analyze.p1(data$train, data$test, ratedPW = 1000)
p1.res$opt.cov #This provides the optimal set of variables.
```

---

analyze.p2

*Apply Period 2 Analysis*


---

**Description**

Conducts period 2 analysis; uses the optimal set of variables obtained in the period 1 analysis to predict the power output of REF and CTR-b turbines in period 2.

**Usage**

```
analyze.p2(per1, per2, opt.cov)
```

**Arguments**

per1	A dataframe containing the period 1 data.
per2	A dataframe containing the period 2 data.
opt.cov	A character vector indicating the optimal set of variables (obtained from the period 1 analysis).

**Value**

The function returns a list of the following datasets.

pred.REF A dataframe including the period 2 prediction for the REF turbine.

pred.CTR A dataframe including the period 2 prediction for the CTR-b turbine.

**References**

H. Hwangbo, Y. Ding, and D. Cabezon, 'Machine Learning Based Analysis and Quantification of Potential Power Gain from Passive Device Installation,' arXiv:1906.05776 [stat.AP], Jun. 2019. <https://arxiv.org/abs/1906.05776>.

**Examples**

```
df.ref <- with(wtg, data.frame(time = time, turb.id = 1, wind.dir = D,
  power = y, air.dens = rho))
df.ctrb <- with(wtg, data.frame(time = time, turb.id = 2, wind.spd = V,
  power = y))
df.ctrn <- df.ctrb
df.ctrn$turb.id <- 3

data <- arrange.data(df.ref, df.ctrb, df.ctrn, p1.beg = '2014-10-24',
  p1.end = '2014-10-25', p2.beg = '2014-10-25', p2.end = '2014-10-26',
  k.fold = 2)

p1.res <- analyze.p1(data$train, data$test, ratedPW = 1000)
p2.res <- analyze.p2(data$per1, data$per2, p1.res$opt.cov)
```

---

arrange.data

*Split, Merge, and Filter Given Datasets for the Subsequent Analysis*


---

**Description**

Generates datasets that consist of the measurements from REF, CTR-b, and CTR-n turbines only. Filters the datasets by eliminating data points with a missing measurement and those with negative power output (optional). Generates training and test datasets for  $k$ -fold CV and splits the entire data into period 1 data and period 2 data.

**Usage**

```
arrange.data(df1, df2, df3, p1.beg, p1.end, p2.beg, p2.end,
  time.format = "%Y-%m-%d %H:%M:%S", k.fold = 5, col.time = 1,
  col.turb = 2, bootstrap = NULL, free.sec = NULL,
  neg.power = FALSE)
```

**Arguments**

df1	A dataframe for reference turbine data. This dataframe must include five columns: timestamp, turbine id, wind direction, power output, and air density.
df2	A dataframe for baseline control turbine data. This dataframe must include four columns: timestamp, turbine id, wind speed, and power output.
df3	A dataframe for neutral control turbine data. This dataframe must include four columns and have the same structure with df2.
p1.beg	A string specifying the beginning date of period 1. By default, the value needs to be specified in ‘%Y-%m-%d’ format, for example, ‘2014-10-24’. A user can use a different format as long as it is consistent with the format defined in time.format below.
p1.end	A string specifying the end date of period 1. For example, if the value is ‘2015-10-24’, data observed until ‘2015-10-23 23:50:00’ would be considered for period 1.
p2.beg	A string specifying the beginning date of period 2.
p2.end	A string specifying the end date of period 2. Defined similarly as p1.end.
time.format	A string describing the format of time stamps used in the data to be analyzed. The default value is ‘%Y-%m-%d %H:%M:%S’.
k.fold	An integer defining the number of data folds for the period 1 analysis and prediction. In the period 1 analysis, $k$ -fold cross validation (CV) will be applied to choose the optimal set of covariates that results in the least prediction error. The value of k.fold corresponds to the $k$ of the $k$ -fold CV. The default value is 5.
col.time	An integer specifying the column number of time stamps in wind turbine datasets. The default value is 1.
col.turb	An integer specifying the column number of turbines’ id in wind turbine datasets. The default value is 2.
bootstrap	An integer indicating the current replication (run) number of bootstrap. If set to NULL, bootstrap is not applied. The default is NULL. A user is not recommended to set this value and directly run bootstrap; instead, use <a href="#">bootstrap.gain</a> to run bootstrap.
free.sec	A list of vectors defining free sectors. Each vector in the list has two scalars: one for starting direction and another for ending direction, ordered clockwise. For example, a vector of <code>c(310, 50)</code> is a valid component of the list. By default, this is set to NULL.
neg.power	Either TRUE or FALSE, indicating whether or not to use data points with a negative power output, respectively, in the analysis. The default value is FALSE, i.e., negative power output data will be eliminated.

**Value**

The function returns a list of several datasets including the following.

- train A list containing  $k$  datasets that will be used to **train** the machine learning model.
- test A list containing  $k$  datasets that will be used to **test** the machine learning model.
- per1 A dataframe containing the period 1 data.
- per2 A dataframe containing the period 2 data.



**Examples**

```

df.ref <- with(wtg, data.frame(time = time, turb.id = 1, wind.dir = D, power = y,
  air.dens = rho))
df.ctrb <- with(wtg, data.frame(time = time, turb.id = 2, wind.spd = V, power = y))
df.ctrn <- df.ctrb
df.ctrn$turb.id <- 3

# For Full Sector Analysis
data <- arrange.data(df.ref, df.ctrb, df.ctrn, p1.beg = '2014-10-24', p1.end = '2014-10-27',
  p2.beg = '2014-10-27', p2.end = '2014-10-30')

# For Free Sector Analysis
free.sec <- list(c(310, 50), c(150, 260))
data <- arrange.data(df.ref, df.ctrb, df.ctrn, p1.beg = '2014-10-24', p1.end = '2014-10-27',
  p2.beg = '2014-10-27', p2.end = '2014-10-30', free.sec = free.sec)

length(data$train) #This equals to k.
length(data$test) #This equals to k.
head(data$per1) #This shows the beginning of the period 1 dataset.
head(data$per2) #This shows the beginning of the period 2 dataset.

```

bootstrap.gain

*Construct a Confidence Interval of the Gain Estimate***Description**

Estimates gain and its confidence interval at a given level of confidence by using bootstrap.

**Usage**

```

bootstrap.gain(df1, df2, df3, opt.cov, n.rep, p1.beg, p1.end, p2.beg,
  p2.end, ratedPW, AEP, pw.freq, freq.id = 3,
  time.format = "%Y-%m-%d %H:%M:%S", k.fold = 5, col.time = 1,
  col.turb = 2, free.sec = NULL, neg.power = FALSE,
  pred.return = FALSE)

```

**Arguments**

df1	A dataframe for reference turbine data. This dataframe must include five columns: timestamp, turbine id, wind direction, power output, and air density.
df2	A dataframe for baseline control turbine data. This dataframe must include four columns: timestamp, turbine id, wind speed, and power output.
df3	A dataframe for neutral control turbine data. This dataframe must include four columns and have the same structure with df2.
opt.cov	A character vector indicating the optimal set of variables (obtained from the period 1 analysis).

n.rep	An integer describing the total number of replications when applying bootstrap. This number determines the confidence level; for example, if n.rep is set to 10, this function will provide an 80% confidence interval.
p1.beg	A string specifying the beginning date of period 1. By default, the value needs to be specified in '%Y-%m-%d' format, for example, '2014-10-24'. A user can use a different format as long as it is consistent with the format defined in time.format below.
p1.end	A string specifying the end date of period 1. For example, if the value is '2015-10-24', data observed until '2015-10-23 23:50:00' would be considered for period 1.
p2.beg	A string specifying the beginning date of period 2.
p2.end	A string specifying the end date of period 2. Defined similarly as p1.end.
ratedPW	A kW value that describes the (common) rated power of the selected turbines (REF and CTR-b).
AEP	A kWh value describing the annual energy production from a single turbine.
pw.freq	A matrix or a dataframe that includes power output bins and corresponding frequency in terms of the accumulated hours during an annual period.
freq.id	An integer indicating the column number of pw.freq that describes the frequency of power bins in terms of the accumulated hours during an annual period. By default, this parameter is set to 3.
time.format	A string describing the format of time stamps used in the data to be analyzed. The default value is '%Y-%m-%d %H:%M:%S'.
k.fold	An integer defining the number of data folds for the period 1 analysis and prediction. In the period 1 analysis, $k$ -fold cross validation (CV) will be applied to choose the optimal set of covariates that results in the least prediction error. The value of k.fold corresponds to the $k$ of the $k$ -fold CV. The default value is 5.
col.time	An integer specifying the column number of time stamps in wind turbine datasets. The default value is 1.
col.turb	An integer specifying the column number of turbines' id in wind turbine datasets. The default value is 2.
free.sec	A list of vectors defining free sectors. Each vector in the list has two scalars: one for starting direction and another for ending direction, ordered clockwise. For example, a vector of c(310, 50) is a valid component of the list. By default, this is set to NULL.
neg.power	Either TRUE or FALSE, indicating whether or not to use data points with a negative power output, respectively, in the analysis. The default value is FALSE, i.e., negative power output data will be eliminated.
pred.return	A logical value whether to return the full prediction results; see <b>Details</b> below. The default value is FALSE.

### Details

For each replication, this function will make a  $k$  of period 1 predictions for each of REF and CTR-b turbine models and an additional period 2 prediction for each model. This results in  $2 \times (k + 1)$

predictions for each replication. With `n.rep` replications, there will be  $n.rep \times 2 \times (k+1)$  predictions in total.

One can avoid storing such many datasets in the memory by setting `pred.return` to `FALSE`; which is the default setting.

## Value

The function returns a list of `n.rep` replication objects (lists) each of which includes the following.

`gain.res` A list containing gain quantification results; see [quantify.gain](#) for the details.

`p1.pred` A list containing period 1 prediction results.

- `pred.REF`: A list of  $k$  datasets each representing the  $k$ th fold's period 1 prediction for the REF turbine.
- `pred.CTR`: A list of  $k$  datasets each representing the  $k$ th fold's period 1 prediction for the CTR-b turbine.

`p2.pred` A list containing period 2 prediction results; see [analyze.p2](#) for the details.

## References

H. Hwangbo, Y. Ding, and D. Cabezon, 'Machine Learning Based Analysis and Quantification of Potential Power Gain from Passive Device Installation,' arXiv:1906.05776 [stat.AP], Jun. 2019. <https://arxiv.org/abs/1906.05776>.

## Examples

```
df.ref <- with(wtg, data.frame(time = time, turb.id = 1, wind.dir = D,
  power = y, air.dens = rho))
df.ctrb <- with(wtg, data.frame(time = time, turb.id = 2, wind.spd = V,
  power = y))
df.ctrn <- df.ctrb
df.ctrn$turb.id <- 3

opt.cov = c('D', 'density', 'Vn', 'hour')
n.rep = 2 # just for illustration; a user may use at least 10 for this.

res <- bootstrap.gain(df.ref, df.ctrb, df.ctrn, opt.cov = opt.cov, n.rep = n.rep,
  p1.beg = '2014-10-24', p1.end = '2014-10-25', p2.beg = '2014-10-25',
  p2.end = '2014-10-26', ratedPW = 1000, AEP = 300000, pw.freq = pw.freq,
  k.fold = 2)

length(res) #2
sapply(res, function(ls) ls$gain.res$gainCurve) #This provides 2 gain curves.
sapply(res, function(ls) ls$gain.res$gain) #This provides 2 gain values.
```

---

pw.freq *Long-Term Frequency of Power Output*

---

### Description

A dataset containing power bins, the proportion of observing each power bin, and the accumulated hours of observing each power bin.

### Usage

pw.freq

### Format

A data frame with 10 rows and 3 columns:

PW\_bin the right end point of the intervals defining power bins

freq the proportion of observing each power bin from historical data

freq\_h the accumulated hours of observing each power bin from historical data

### Note

- This dataset is provided to show how a user is expected to structure the long-term frequency data, which will be used in [analyze.gain](#) or in [quantify.gain](#).
- In the gain analysis performed by [analyze.gain](#), power bins will be defined with 100kW increments. To be consistent, PW\_bin must be defined with 100kW increments. For example, if rated power is 1,000kW (1MW), power bins shall be generated by using the intervals of [0kW, 100kW], [100kW, 200kW], ..., [900kW, 1000kW].
- The gain analysis will only need the information specified in freq\_h, so as long as the elements in this column corresponds to each power bin (with 100kW increments) and the number of elements matches the number of power bins, there should not be any problem.

---

quantify.gain *Quantify Gain Based on Period 1 and Period 2 Prediction*

---

### Description

Calculates effect curve, offset curve, and gain curve, and quantifies gain by using both period 1 and period 2 prediction results.

### Usage

quantify.gain(p1.res, p2.res, ratedPW, AEP, pw.freq, freq.id = 3)

**Arguments**

p1.res	A list containing the period 1 analysis results.
p2.res	A list containing the period 2 prediction results.
ratedPW	A kW value that describes the (common) rated power of the selected turbines (REF and CTR-b).
AEP	A kWh value describing the annual energy production from a single turbine.
pw.freq	A matrix or a dataframe that includes power output bins and corresponding frequency in terms of the accumulated hours during an annual period.
freq.id	An integer indicating the column number of pw.freq that describes the frequency of power bins in terms of the accumulated hours during an annual period. By default, this parameter is set to 3.

**Value**

The function returns a list containing the following.

effectCurve	A vector of length $m$ illustrating REF turbine's power output difference between period 1 and 2, where $m$ is the number of power bins.
offsetCurve	A vector of length $m$ illustrating CTR-b turbine's power output difference between period 1 and 2.
gainCurve	A vector of length $m$ illustrating the bin-wise gain. Equivalent to $\text{effCurve} - \text{offCurve}$ .
gain	A scalar representing the final gain after offset adjustment (derived from gainCurve).
effect	A scalar representing the initial effect without offset correction (derived from effCurve).
offset	A scalar representing the offset value for the final gain quantification (derived from offCurve).

**References**

H. Hwangbo, Y. Ding, and D. Cabezon, 'Machine Learning Based Analysis and Quantification of Potential Power Gain from Passive Device Installation,' arXiv:1906.05776 [stat.AP], Jun. 2019. <https://arxiv.org/abs/1906.05776>.

**Examples**

```
df.ref <- with(wtg, data.frame(time = time, turb.id = 1, wind.dir = D,
  power = y, air.dens = rho))
df.ctrb <- with(wtg, data.frame(time = time, turb.id = 2, wind.spd = V,
  power = y))
df.ctrn <- df.ctrb
df.ctrn$turb.id <- 3

data <- arrange.data(df.ref, df.ctrb, df.ctrn, p1.beg = '2014-10-24',
  p1.end = '2014-10-25', p2.beg = '2014-10-25', p2.end = '2014-10-26',
  k.fold = 2)

p1.res <- analyze.p1(data$train, data$test, ratedPW = 1000)
p2.res <- analyze.p2(data$per1, data$per2, p1.res$opt.cov)
```

```
res <- quantify.gain(p1.res, p2.res, ratedPW = 1000, AEP = 300000, pw.freq = pw.freq)

res$effect - res$offset #This should be equivalent to the final gain below.
res$gain

res$gainCurve #This shows the bin-wise gain (after offset adjustment).
```

---

wtg

*Wind turbine operational data*

---

### Description

A dataset containing the measurements of wind-related and other environmental variables as well as the actual power output measurements of an operating wind turbine.

### Usage

wtg

### Format

A data frame with 1000 rows and 7 variables:

- time: timestamp,
- V: wind speed ( $m/s$ ),
- D: wind direction (degree),
- rho: air density ( $kg/m^3$ ),
- I: turbulence intensity,
- Sb: below-hub wind shear,
- y: power output ( $kW$ ).

### Note

This dataset is generated by using windpw dataset in kernplus package. Timestamp has been added (randomly), and the power output of windpw dataset has been arbitrarily multiplied by 10 to represent  $kW$  values.

# Index

## \* datasets

pw.freq, [12](#)

wtg, [14](#)

analyze.gain, [2](#), [12](#)

analyze.p1, [4](#), [5](#)

analyze.p2, [4](#), [6](#), [11](#)

arrange.data, [3](#), [4](#), [7](#)

bootstrap.gain, [3](#), [8](#), [9](#)

pw.freq, [12](#)

quantify.gain, [4](#), [11](#), [12](#), [12](#)

wtg, [14](#)