

# Package: kernplus (via r-universe)

September 4, 2024

**Version** 0.1.2

**Date** 2019-02-25

**Type** Package

**Title** A Kernel Regression-Based Multidimensional Wind Turbine Power Curve

**Author** Yu Ding [aut], Hoon Hwangbo [aut, cre]

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

**Description** Provides wind energy practitioners with an effective machine learning-based tool that estimates a multivariate power curve and predicts the wind power output for a specific environmental condition.

**Depends** R (>= 3.3.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** circular (>= 0.4-7), KernSmooth (>= 2.23-15), mixtools (>= 1.1.0), graphics (>= 3.3.0), stats (>= 3.3.0)

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Date/Publication** 2019-02-24 21:50:06 UTC

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

**RemoteUrl** <https://github.com/cran/kernplus>

**RemoteRef** HEAD

**RemoteSha** 71c6ffde035a6aff7204909b455f4fa750e82cbc

## Contents

kp.pwcurv . . . . .	2
windpw . . . . .	4

<b>Index</b>	<b>5</b>
--------------	----------

kp.pwcurv

*Predict Wind Power Output by Using a Multivariate Power Curve***Description**

Takes multiple environmental variable inputs measured on an operating wind farm and predicts the wind power output under the given environmental condition.

**Usage**

```
kp.pwcurv(y, x, x.new = x, id.spd = 1, id.dir = NA)
```

**Arguments**

y	An $n$ -dimensional vector or a matrix of size $n$ by 1 containing wind power output data. This along with $x$ trains the multidimensional power curve model.
x	An $n$ by $p$ matrix or a data frame containing the input data for $p$ predictor variables (wind and weather variables). This $x$ must have the same number of rows as $y$ , i.e., $n$ .
x.new	A matrix or a data frame containing new input conditions of the $p$ predictor variables for which a prediction of wind power output will be made. This is an optional parameter and will be set to $x$ by default, if it is not supplied.
id.spd	The column number of $x$ (and of $x.new$ , if supplied) indicating wind speed data. Default to 1.
id.dir	The column number of $x$ (and of $x.new$ , if supplied) indicating wind direction data. Default to NA, but this parameter needs to be set if $x$ includes wind direction data.

**Value**

A vector representing the predicted power output for the new wind/weather condition specified in  $x.new$ . If  $x.new$  is not supplied, this function returns the fitted power output for the given  $x$ .

**Note**

- This function is developed for wind power prediction. As such, the response  $y$  represents wind power output and the covariates  $x$  include multiple wind and weather variables that potentially affect the power output.
- The data matrix  $x$  is expected to include at least wind speed and wind direction data. As measurements of other environmental variables become available, they can be added to the  $x$ . Typically, the first column of  $x$  corresponds to wind speed data and the second column to wind direction data and, as such, `id.spd = 1` and `id.dir = 2`.
- If  $x$  has a single variable of wind speed, i.e.,  $p = 1$  and `id.spd = 1`, this function returns an estimate (or prediction) of the Nadaraya-Watson estimator with a Gaussian kernel by using the `ksmooth` function in the **stats** package.

## References

Lee, G., Ding, Y., Genton, M.G., and Xie, L. (2015) Power Curve Estimation with Multivariate Environmental Factors for Inland and Offshore Wind Farms, *Journal of the American Statistical Association* 110(509):56-67.

## See Also

[windpw](#), [ksmooth](#)

## Examples

```
head(windpw)

### Power curve estimation.

# By using a single input of wind speed.
pwcurv.est <- kp.pwcurv(windpw$y, windpw$V)

# By using wind speed and direction: id.dir needs to be set.
pwcurv.est <- kp.pwcurv(windpw$y, windpw[, c('V', 'D')], id.dir = 2)

# By using full covariates: confirm whether id.spd and id.dir are correctly specified.
pwcurv.est <- kp.pwcurv(windpw$y, windpw[, c('V', 'D', 'rho', 'I', 'Sb')], id.spd = 1, id.dir = 2)

### Wind power prediction.

# Suppose only 90% of data are available and use the rest 10% for prediction.
df.tr <- windpw[1:900, ]
df.ts <- windpw[901:1000, ]
id.cov <- c('V', 'D', 'rho', 'I', 'Sb')
pred <- kp.pwcurv(df.tr$y, df.tr[, id.cov], df.ts[, id.cov], id.dir = 2)

### Evaluation of wind power prediction based on 10-fold cross validation.

# Partition the given dataset into 10 folds.
index <- sample(1:nrow(windpw), nrow(windpw))
n.fold <- round(nrow(windpw) / 10)
ls.fold <- rep(list(c()), 10)
for(fold in 1:9) {
  ls.fold[[fold]] <- index[((fold-1)*n.fold+1):(fold*n.fold)]
}
ls.fold[[10]] <- index[(9*n.fold+1):nrow(windpw)]

# Predict wind power output.
pred.res <- rep(list(c()), 10)
id.cov <- c('V', 'D', 'rho', 'I', 'Sb')
for(k in 1:10) {
  id.fold <- ls.fold[[k]]
  df.tr <- windpw[-id.fold, ]
```

```
df.ts <- windpw[id.fold, ]
pred <- kp.pwcurv(df.tr$y, df.tr[, id.cov], df.ts[, id.cov], id.dir = 2)
pred.res[[k]] <- list(obs = df.ts$y, pred)
}

# Calculate rmse and its mean and standard deviation.
rmse <- sapply(pred.res, function(res) with(res, sqrt(mean((obs - pred)^2))))
mean(rmse)
sd(rmse)
```

---

windpw

*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

windpw

### Format

A data frame with 1000 rows and 6 variables:

- V: wind speed ( $m/s$ ),
- D: wind direction (degree),
- rho: air density ( $kg/m^3$ ),
- I: turbulence intensity,
- Sb: below-hub wind shear,
- y: normalized power output relative to the rated power (%).

### Note

This dataset is a subset of an actual operational dataset, which is available at <https://aml.engr.tamu.edu/2001/09/01/publications/> where other operational datasets are also available. To access the datasets, click the link ‘data’ attached to J53.

This dataset was generated by drawing 1000 random samples from the original dataset. As such, the sequence of rows is not arranged in time.

# Index

\* **datasets**

    windpw, 4

kp.pwcurv, 2

ksmooth, 3

windpw, 3, 4