Package 'dartle'

June 20, 2017

Type Package

Title Air Quality Model Benchmarking

Version 0.1.0

Author Giovanni Bonafè

Maintainer Giovanni Bonafè <giovanni.bonafe@arpa.fvg.it>

Description A toolkit of functions for air quality model benchmarking,
inspired by the DELTA tool (JRC-IES) and the work of the FAIRMODE WP1.

License MIT License + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Depends R (>= 2.10)

Imports stats, dplyr, RcppRoll, data.table, ggplot2, tidyr

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

daily_statistics *Daily statistics*

Description

dMaxAvg8h: daily maximum of 8h rolling mean

dMean: daily average

Usage

```
dMaxAvg8h(data, value, time = "Time", point = "Point", req = 18)

dMean(data, value, time = "Time", point = "Point", req = 18)
```

Arguments

data	data frame
value	name of variable to be processed
time	column containing time (POSIXct)
point	column containing point ID
req	minimum valid required 8h averages to compute their daily max

Value

dMaxAvg8h: daily maxima of 8h rolling mean, for each point

dMean: daily averages, for each point

indicators *Core Set of Statistical Indicators*

Description

RMSE: Root Mean Square Error

R: Correlation coefficient

BIAS: Bias

NMB: Normalised Mean Bias

NMSD: Normalised Mean Standard Deviation

CRMSE: Centered Root Mean Square Error

Usage

```
RMSE(obs, mod)

R(obs, mod)

BIAS(obs, mod)

NMB(obs, mod)

NMSD(obs, mod)

CRMSE(obs, mod)
```

Arguments

obs	numeric vector of observed values
mod	numeric vector of modelled values

Value

RMSE: Root Mean Square Error as in eq.1, tab.1, p.18 in Janssen et al., 2017

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(M_i - O_i)^2}$$

R: Correlation coefficient as in eq.2, tab.1, p.18 in Janssen et al., 2017

$$R = \frac{\sum_{i=1}^{N}(O_i - \bar{O})(M_i - \bar{M})}{\sqrt{\sum_{i=1}^{N}(O_i - \bar{O})^2}\sqrt{\sum_{i=1}^{N}(M_i - \bar{M})^2}}$$

BIAS: bias as in eq.3, tab.1, p.18 in Janssen et al., 2017

$$BIAS = \bar{M} - \bar{O}$$

NMB: Normalised Mean Bias as in eq.3, tab.1, p.18 in Janssen et al., 2017

$$NMB = \frac{\bar{M} - \bar{O}}{\bar{O}}$$

NMSD: Normalised Mean Standard Deviation as in eq.4, tab.1, p.18 in Janssen et al., 2017

$$NMSD = \frac{\sigma_M - \sigma_O}{\sigma_O}$$

CRMSE: Centered Root Mean Square Error as in eq.31, p.29 in Janssen et al., 2017

$$CRMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[(M_i - \bar{M}) - (O_i - \bar{O})]^2}$$

References

Janssen et al., 2017. "Guidance Document on Modelling Quality Objectives and Benchmarking. Version 2.1"

mod.data *Forecasted concentrations*

Description

A dataset containing hourly concentrations of PM10, PM2.5, NO2 and ozone forecasted at background sites in the region of Friuli Venezia Giulia (Italy)

Usage

```
mod.data
```

Format

A data frame with 658944 rows and 4 variables:

Time time in POSIXct format

Var "c_PM10" (PM10), "c_PM25" (PM2.5), "c_NO2" (NO2) or "c_O3" (ozone)

Point station code

Value concentration (in $\mu g/m^3$)

Source

http://www.arpa.fvg.it/

mod_uncertainty *Model Uncertainty*

Description

RMS_U_mod: RMS uncertainty for model time series

U_mod_year: Uncertainty for model yearly averages

Usage

```
RMS_U_mod(obs, mod, pollutant, ...)

U_mod_year(obs, mod, pollutant, ...)
```

Arguments

obs	numeric vector of observed values (yearly averages in U_mod_year)
mod	numeric vector of modelled values (yearly averages in U_mod_year)
pollutant	one of "NO2", "O3", "PM10", "PM2.5"
...	arguments to be passed to U_obs_95 in RMS_U_mod, to U_obs_95_year in U_mod_year

Value

`RMS_U_mod`: root mean square uncertainty for model time series, as in eq.23, p.24 in Janssen et al., 2017

$$RMS_{U_M} = RMS_U \sqrt{\left(\frac{RMSE}{RMS_U}\right)^2 - 1}$$

`MQI_ts`: uncertainty for model yearly averages, as in eq.24, p.24 in Janssen et al., 2017

$$U(\bar{M}) = U_{95}(\bar{O}) \sqrt{\left(\frac{BIAS}{U_{95}(\bar{O})}\right)^2 - 1}$$

References

Janssen et al., 2017. "Guidance Document on Modelling Quality Objectives and Benchmarking. Version 2.1"

`MPI` *Modelling Performance Indicators*

Description

`MPI_bias`: MPI for bias

`MPI_corr_time`: temporal MPI for correlation

`MPI_sdev_time`: temporal MPI for standard deviation

`MPI_corr_space`: spatial MPI for correlation

`MPI_sdev_space`: spatial MPI for standard deviation

`MPI_perc`: MPI for high percentile values

Usage

```
MPI_bias(obs, mod, pollutant, beta = 2, ...)

MPI_corr_time(obs, mod, pollutant, beta = 2, ...)

MPI_sdev_time(obs, mod, pollutant, beta = 2, ...)

MPI_corr_space(obs, mod, pollutant, beta = 2, ...)

MPI_sdev_space(obs, mod, pollutant, beta = 2, ...)

MPI_perc(obs, mod, pollutant, beta = 2, ...)
```

Arguments

`obs`	numeric vector of observed values (yearly averages in `MPI_*_space`)
`mod`	numeric vector of modelled values (yearly averages in `MPI_*_space`)
`pollutant`	one of `"NO2"`, `"O3"`, `"PM10"`, `"PM2.5"`
`beta`	parameter β (default is 2)
`...`	arguments to be passed to `RMS_U_obs` in `MPI_bias` and `MPI_*_time`, to `U_obs_95_year` in `MPI_*_space`, to `U_obs_95` in `MPI_perc`

Value

`MPI_bias`: Modelling Performance Indicator for the bias, as in eq.26, tab.4, p.27 in Janssen et al., 2017

$$MPI_{bias} = \frac{BIAS}{\beta RMS_U}$$

`MPI_corr_time`: temporal Modelling Performance Indicator for the correlation, as in eq.27, tab.4, p.27 in Janssen et al., 2017

$$MPI_{corr,time} = \frac{1}{\left(R + \frac{\beta^2 RMS_U^2}{2\sigma_O \sigma_M}\right)}$$

`MPI_sdev_time`: temporal Modelling Performance Indicator for the standard deviation, as in eq.28, tab.4, p.27 in Janssen et al., 2017

$$MPI_{s.d.,time} = \frac{\sigma_M - \sigma_O}{\beta RMS_U}$$

`MPI_corr_space`: spatial Modelling Performance Indicator for the correlation, as in eq.29, tab.5, p.27 in Janssen et al., 2017

$$MPI_{corr,space} = \frac{1}{\left(R + \frac{\beta^2 RMS_{\bar{U}}^2}{2\sigma_{\bar{O}} \sigma_{\bar{M}}}\right)}$$

`MPI_sdev_space`: spatial Modelling Performance Indicator for the standard deviation, as in eq.30, tab.5, p.27 in Janssen et al., 2017

$$MPI_{s.d.,space} = \frac{\sigma_{\bar{M}} - \sigma_{\bar{O}}}{\beta RMS_{\bar{U}}}$$

`MPI_perc`: Modelling Performance Indicator for high percentile values, as in eq.35, p.35 in Janssen et al., 2017

$$MPI_{perc} = \frac{M_{perc} - O_{perc}}{\beta U_{95}(O_{perc})}$$

References

Janssen et al., 2017. "Guidance Document on Modelling Quality Objectives and Benchmarking. Version 2.1"

`MQI` *Modelling Quality Indicators*

Description

`MQI_ts`: MQI for a time series (vector)

`MQI_ts_synth`: MQI for a time series (synthetic scalar)

`MQI_year`: MQI for yearly average

Usage

```
MQI_ts(obs, mod, pollutant, beta = 2, ...)

MQI_ts_synth(obs, mod, pollutant, beta = 2, ...)

MQI_year(obs, mod, pollutant, beta = 2, ...)
```

Arguments

obs	numeric vector of observed values (yearly averages in MQI_year)
mod	numeric vector of modelled values (yearly averages in MQI_year)
pollutant	one of "NO2", "O3", "PM10", "PM2.5"
beta	parameter β (default is 2)
...	arguments to be passed to U_obs_95 in MQI_ts and MQI_ts_synth, to U_obs_95_year in MQI_year

Value

MQI_ts: Modelling Quality Indicator for a time series, as in eq.15, p.22 in Janssen et al., 2017

$$MQI = \frac{|O_i - M_i|}{\beta U_{95}(O_i)}$$

MQI_ts_synth: scalar synthetic Modelling Quality Indicator for a time series, as in eq.17, p.23 in Janssen et al., 2017

$$MQI = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_i - M_i)^2}}{\beta\sqrt{\frac{1}{N}\sum_{i=1}^{N}U_{95}(O_i)^2}}$$

MQI_year: Modelling Quality Indicator for yearly average, as in eq.18, p.23 in Janssen et al., 2017

$$MQI = \frac{|\bar{O} - \bar{M}|}{\beta U_{95}(\bar{O})}$$

References

Janssen et al., 2017. "Guidance Document on Modelling Quality Objectives and Benchmarking. Version 2.1"

---

multiplot *Multiple plot function*

Description

Multiple plot function

Usage

```
multiplot(..., plotlist = NULL, cols = 1, layout = NULL)
```

Arguments

`...`	ggplot objects to plot
`plotlist`	list of ggplot objects to plot
`cols`	number of columns in layout
`layout`	a matrix specifying the layout. If present, 'cols' is ignored. If the layout is something like `matrix(c(1,2,3,3), nrow=2, byrow=TRUE)`, then plot 1 will go in the upper left, 2 will go in the upper right, and 3 will go all the way across the bottom.

Value

a ggplot object

`obs.no2` *Observed NO2 concentrations*

Description

A dataset containing hourly concentrations of NO2 measured at background sites in the region of Friuli Venezia Giulia (Italy)

Usage

```
obs.no2
```

Format

A data frame with 202752 rows and 8 variables:

Time time (POSIXct format)

ID station code

Station station name

Value concentration (in $\mu g/m^3$)

ZoneType `"URB"` (urban), `"SBR"` (suburban) or `"RUR"` (rural)

StationType always `"BKG"` (background)

Lat latitude

Lon longitude

Source

http://www.arpa.fvg.it/

obs.o3 *Observed ozone concentrations*

Description

A dataset containing hourly concentrations of ozone measured at background sites in the region of Friuli Venezia Giulia (Italy)

Usage

```
obs.o3
```

Format

A data frame with 215424 rows and 8 variables:

Time time (POSIXct format)

ID station code

Station station name

Value concentration (in $\mu g/m^3$)

ZoneType `"URB"` (urban), `"SBR"` (suburban) or `"RUR"` (rural)

StationType always `"BKG"` (background)

Lat latitude

Lon longitude

Source

http://www.arpa.fvg.it/

obs.pm10 *Observed PM10 concentrations*

Description

A dataset containing daily concentrations of PM10 measured at background sites in the region of Friuli Venezia Giulia (Italy)

Usage

```
obs.pm10
```

Format

A data frame with 9504 rows and 8 variables:

Time day (POSIXct format)

ID station code

Station station name

Value concentration (in $\mu g/m^3$)

ZoneType `"URB"` (urban), `"SBR"` (suburban) or `"RUR"` (rural)

StationType always `"BKG"` (background)

Lat latitude

Lon longitude

Source

http://www.arpa.fvg.it/

`obs.pm25` *Observed PM2.5 concentrations*

Description

A dataset containing daily concentrations of PM2.5 measured at background sites in the region of Friuli Venezia Giulia (Italy)

Usage

```
obs.pm25
```

Format

A data frame with 1056 rows and 8 variables:

Time day (POSIXct format)

ID station code

Station station name

Value concentration (in $\mu g/m^3$)

ZoneType `"URB"` (urban), `"SBR"` (suburban) or `"RUR"` (rural)

StationType always `"BKG"` (background)

Lat latitude

Lon longitude

Source

http://www.arpa.fvg.it/

obs_uncertainty *Measurements Uncertainty*

Description

U_obs_95: Measurements uncertainty

RMS_U_obs: Root Mean Square Uncertainty

U_obs_95_year: Measurements uncertainty for yearly averaged values

Usage

```
U_obs_95(obs, pollutant = NULL, U_RV95r = NULL, alpha = NULL, RV = NULL)

RMS_U_obs(obs, pollutant = NULL, U_RV95r = NULL, alpha = NULL,
  RV = NULL)

U_obs_95_year(obs, pollutant = NULL, U_RV95r = NULL, alpha = NULL,
  RV = NULL, Np = NULL, Nnp = NULL)
```

Arguments

obs	numeric vector of observed values (yearly averaged for U_obs_95_year)
pollutant	one of "NO2", "O3", "PM10", "PM2.5"
U_RV95r	parameter $U_{95,r}^{RV}$; optional if pollutant is given, otherwise compulsory
alpha	parameter α; optional if pollutant is given, otherwise compulsory
RV	parameter RV; optional if pollutant is given, otherwise compulsory
Np	parameter N_p; optional if pollutant is given, otherwise compulsory
Nnp	parameter N_{np}; optional if pollutant is given, otherwise compulsory

Value

U_obs_95: measurements uncertainty U_{95}, as in eq.10, p.20 in Janssen et al., 2017

$$U_{95}(O_i) = U_{95,r}^{RV}\sqrt{(1-\alpha^2)O_i^2 + \alpha^2 \cdot RV^2}$$

RMS_U_obs: root mean square uncertainty RMS_U, as in eq.11, p.20 in Janssen et al., 2017

$$RMS_U = U_{95,r}^{RV}\sqrt{(1-\alpha^2)(\bar{O}^2 + \sigma_O^2) + \alpha^2 \cdot RV^2}$$

U_obs_95_year: measurements uncertainty for yearly averaged values U_{95}, as in eq.12, p.20 in Janssen et al., 2017

$$U_{95}(\bar{O}) \cong U_{95,r}^{RV}\sqrt{\frac{(1-\alpha^2)}{N_p}\bar{O}^2 + \frac{\alpha^2 \cdot RV^2}{N_{np}}}$$

References

Janssen et al., 2017. "Guidance Document on Modelling Quality Objectives and Benchmarking. Version 2.1"

Examples

```
# concentrations
cc <- 1:400

# uncertainties for time series
Uo3   <- U_obs_95(cc, pollutant = "O3")
Upm10 <- U_obs_95(cc, pollutant = "PM10")
Upm25 <- U_obs_95(cc, pollutant = "PM2.5")
Uno2  <- U_obs_95(cc, pollutant = "NO2")
plot(cc, Uo3, type="l", log = "xy", xlab="measured concentration",
ylab="uncertainty", ylim=c(1,100))
lines(cc,Upm10,lty=2)
lines(cc,Upm25,lty=3)
lines(cc,Uno2,lty=4)
legend("topleft",lty=1:4,legend=c("O3", "PM10", "PM2.5", "NO2"))

# uncertainties for yearly averages
Uo3   <- U_obs_95_year(cc, pollutant = "O3")
Upm10 <- U_obs_95_year(cc, pollutant = "PM10")
Upm25 <- U_obs_95_year(cc, pollutant = "PM2.5")
Uno2  <- U_obs_95_year(cc, pollutant = "NO2")
plot(cc, Uo3, type="l", log = "xy", xlab="measured concentration",
ylab="uncertainty", ylim=c(1,100))
lines(cc,Upm10,lty=2)
lines(cc,Upm25,lty=3)
lines(cc,Uno2,lty=4)
legend("topleft",lty=1:4,legend=c("O3", "PM10", "PM2.5", "NO2"))
```

---

parameters *Pollutant-dependent parameters*

Description

Parameters used to calculate the measurement uncertainty

Usage

```
params_U(pollutant = c("NO2", "O3", "PM10", "PM2.5"))

perc(pollutant = c("NO2", "O3", "PM10", "PM2.5"))

threshold(pollutant = c("NO2", "O3", "PM10", "PM2.5"))
```

Arguments

`pollutant`	one of `"NO2"`, `"O3"`, `"PM10"`, `"PM2.5"`

Value

`params_U`: numeric list of parameters `U_RV95r`, `alpha`, `RV`, `Np`, `Nnp`, as in tab.2, p.21 in Janssen et al., 2017

`perc`: numeric value in `[0,1]`, corresponding to the high percentile selected for the pollutant (if possible, according to legislation): for hourly NO2 99.8%, for the 8h daily maximum of ozone 92.9%, for daily PM10 and PM2.5 90.4%

`threshold`: numeric values (in $\mu g/m^3$) of the threshold used to calculate exceedances, according to legislation: 200 for hourly NO2, 120 for the 8h daily maximum of ozone, 50 for daily PM10, `NA` otherwise

References

Janssen et al., 2017. "Guidance Document on Modelling Quality Objectives and Benchmarking. Version 2.1"

`summary_plot` *summary plot*

Description

`summary_report`: prepare input

`summary_plot`: summary plot

Usage

```
summary_report(data, obs, mod, point, pollutant = c("NO2", "O3", "PM10",
  "PM2.5"), beta = 2)

summary_plot(s_rep)
```

Arguments

`data`	a data frame including observations and model forecasts
`obs`	name of the column with observed data
`mod`	name of the column with forecasts
`point`	name of the column with station ID
`pollutant`	one of `"NO2"`, `"O3"`, `"PM10"`, `"PM2.5"`
`beta`	parameter β (default is 2)
`s_rep`	output of `summary_report`

Value

`summary_report` returns a list of 3:

`summary_points` a data frame with 8 variables for each station

- `Point` station ID
- `Obs.ave` annual mean of observations (paired with forecasts)
- `Mod.ave` annual mean of forecasts (paired with observations)
- `mpi_bias` Model Performance Indicator for the bias (see MPI_bias)
- `mpi_corr_time` Model Performance Indicator for the correlation in time (see MPI_corr_time)
- `mpi_sdev_time` Model Performance Indicator for the standard deviation in time (see MPI_sdev_time)

`mpi_perc` Model Performance Indicator for high percentile values (see MPI_perc)

`n_valid` no. of valid data

`summary_overall` a data frame with 3 overall indicators

`mpi_corr_space` Model Performance Indicator for the correlation in space (see MPI_corr_space)

`mpi_sdev_space` Model Performance Indicator for the standard deviation in space (see MPI_sdev_space)

`n_points` no. of valid stations

`parameters` a list of 7 parameters

`U_RV95r` see params_U

`alpha` see params_U

`RV` see params_U

`Np` see params_U

`Nnp` see params_U

`pollutant` same as in input, one of `"NO2"`, `"O3"`, `"PM10"`, `"PM2.5"`

`beta` same as in input

`summary_plot` returns a summary plot (object of class `ggplot` and `gg`)

References

Janssen et al., 2017. "Guidance Document on Modelling Quality Objectives and Benchmarking. Version 2.1"

Examples

```
# prepare dataset
require(dplyr)
Mod <- dMean(mod.data %>% filter(Var=="c_PM10"),
             value = "Value", time = "Time", point = "Point")
Obs <- obs.pm10 %>% mutate(Day=format(Time,"%Y-%m-%d"),
                           Point=ID)
Dat <- inner_join(Mod, Obs, by=c("Point", "Day"), suffix = c(".mod", ".obs"))

# calculate indicators
s_rep <- summary_report(Dat, obs = "Value.obs", mod = "Value.mod",
                        point = "Point", pollutant = "PM10")

# plot
summary_plot(s_rep)
```

---

target_plot *Target plot*

Description

`target_report`: prepare input for target plot

`target_plot`: target plot

Usage

```
target_report(data, obs, mod, point, pollutant = c("NO2", "O3", "PM10",
  "PM2.5"), beta = 2)

target_plot(t_rep)
```

Arguments

`data`	a data frame including observations and model forecasts
`obs`	name of the column with observed data
`mod`	name of the column with forecasts
`point`	name of the column with station ID
`pollutant`	one of `"NO2"`, `"O3"`, `"PM10"`, `"PM2.5"`
`beta`	parameter β (default is 2)
`t_rep`	output of `target_report`

Value

`target_report` returns a list of 3:

`quality_points` a data frame with 9 variables for each station

- `Point` station ID
- `Obs.ave` annual mean of observations (paired with forecasts)
- `Mod.ave` annual mean of forecasts (paired with observations)
- `rmsu` RMS_U (see RMS_U_obs)
- `crmse_norm` $CRMSE/(\beta * RMS_U)$
- `r` R (see R)
- `crmse_ratio` $abs(NMSD)/sqrt(2 * (1 - R))$ as in eq.34, p.30 in Janssen et al., 2017
- `bias_norm` $BIAS/(\beta * RMS_U)$
- `mqi_ts` Model Quality Indicator $RMSE/(\beta * RMS_U)$ (see MQI_ts_synth)
- `mqi_year` Model Quality Indicator for yearly averages (see MQI_year)
- `n_valid` no. of valid data

`quality_overall` a data frame with 3 overall indicators

- `mqi_ts_p90` 90th percentile of `mqi_ts`
- `mqi_year_p90` 90th percentile of `mqi_year`
- `n_points` no. of valid stations

`parameters` a list of 7 parameters

- `U_RV95r` see params_U
- `alpha` see params_U
- `RV` see params_U
- `Np` see params_U
- `Nnp` see params_U
- `pollutant` same as in input, one of `"NO2"`, `"O3"`, `"PM10"`, `"PM2.5"`
- `beta` same as in input

`target_plot` returns a target plot (object of class `ggplot` and `gg`)

References

Janssen et al., 2017. "Guidance Document on Modelling Quality Objectives and Benchmarking. Version 2.1"

Examples

```
# prepare dataset
require(dplyr)
Mod <- dMean(mod.data %>% filter(Var=="c_PM10"),
             value = "Value", time = "Time", point = "Point")
Obs <- obs.pm10 %>% mutate(Day=format(Time,"%Y-%m-%d"),
                           Point=ID)
Dat <- inner_join(Mod, Obs, by=c("Point", "Day"), suffix = c(".mod", ".obs"))

# calculate indicators
t_rep <- target_report(Dat, obs = "Value.obs", mod = "Value.mod",
                       point = "Point", pollutant = "PM10")

# plot
target_plot(t_rep)
```

Index