Package 'SALURhelper'

April 24, 2025

Title Helper Functions for SALURBAL Climate Analyses

Version 1.0.1

Description This package provides helper functions analysis types commonly used by SALURBAL, such as case time series, along with access to public or simulated data sets to run the associated analyses.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Import tibble

Contents

`get_df_RRs` *Create a Dataframe from Reduced Predictions*

Description

This function extracts the predicted exposure response curve and its associated confidence bands from a DLNM prediction. This function accepts a `crossreduce` object and, depending on the model link used in the prediction, will return values associated with the RR or output.

Usage

```
get_df_RRs(pred, var_name = "Temperature")
```

Arguments

`pred`	An object of class "`crossreduce`".
`var_name`	The name of the exposure variable of interest.

Value

A tibble with with the following columns:

- `Temperature`: The values of the predictor variable. Column name determined by the `var_name` argument.
- `RR`: The predicted risk, rate, or odds ratio from `crossreduce`.
- `RR_low`, `RR_high`: The lower/upper bound of the confidence bands for `RR`. Depends on the `ci.level` argument in `crossreduce`.
- `fit`: The predicted output from `crossreduce`.
- `low`, `high`: The lower/upper bound of the confidence bands for `fit`. Depends on the `ci.level` argument in `crossreduce`.

Examples

```
# The NYC airquality sample dataset
help(airquality)

# Arguments to specify cross-basis
arg_var <- list(fun = "ns", knots = quantile(airquality$Temp, c(.1, .75, .9)))
arg_lag <- list(fun = "ns", knots = logknots(7, nk = 3))

# Cross-basis and model fit
cb <- crossbasis(airquality$Temp, lag = 7, argvar = arg_var, arglag = arg_lag)
fit1 <- glm(Ozone ~ cb, family = "gaussian",     data = airquality)
fit2 <- glm(Ozone ~ cb, family = "quasipoisson", data = airquality)

# Prediction and extracted RRs (or output)
pred1 <- crossreduce(cb, fit1)
df_RR1 <- get_df_RRs(pred1)

pred2 <- crossreduce(cb, fit2)
df_RR2 <- get_df_RRs(pred2)

# Plots with ggplot2
df_RR1 |>
  ggplot() +
  geom_line(aes(Temperature, fit)) +
  geom_ribbon(aes(Temperature, ymin = low, ymax = high), alpha = .2)

df_RR2 |>
  ggplot() +
  geom_line(aes(Temperature, RR)) +
  geom_ribbon(aes(Temperature, ymin = RR_low, ymax = RR_high), alpha = .2)
```

`get_MMT` *Get the MMT*

Description

When making predictions with distributed lag nonlinear models, this function retrieves the value of the exposure that corresponds to the minimum value of the response. These models are commonly used to predict mortality as a function of temperature, for which this function returns the minimum mortality temperature (MMT).

Usage

```
get_MMT(crosspred)
```

Arguments

crosspred	An object of class "crosspred" or "crossreduce".

Value

The numeric value of the exposure corresponding to the minimum value of the response.

Examples

```
# The NYC airquality sample dataset
help(airquality)

# Arguments to specify cross-basis
arg_var <- list(fun = "ns", knots = quantile(airquality$Temp, c(.1, .75, .9)))
arg_lag <- list(fun = "ns", knots = logknots(7, nk = 3))

# Cross-basis and model fit
cb <- crossbasis(airquality$Temp, lag = 7, argvar = arg_var, arglag = arg_lag)
fit <- glm(Ozone ~ cb, family = "quasipoisson", data = airquality)

# Prediction with default centering
pred <- crosspred(cb, fit)
plot(pred, "overall")

# Prediction centering at the MMT
pred_centered <- crosspred(cb, fit, cen = get_MMT(pred))
plot(pred_centered, "overall")
```

QAIC *Calculate QAIC and QBIC*

Description

This function calculates quasi-Akaike information criterion (QAIC) or the quasi-Bayesian information criterion (QBIC) for quasi-Poisson regression as defined in Gasparrini, Armstrong, and Kenward (2010). When comparing multiple models, the "best" model is that which minimizes these criteria.

Usage

```
QAIC(model)

QBIC(model)
```

Arguments

model	A model object inheriting from the "glm" class, whose family parameter was specified as quasipoisson.

Details

The equations used to calculate the QAIC and QBIC are QAIC $= -2L(\hat{\boldsymbol{\theta}}) + 2\hat{\phi}k$, and QBIC $= -2L(\hat{\boldsymbol{\theta}}) + 2\hat{\phi}k$, where L is the log-likelihood of the fitted model with parameters $\hat{\boldsymbol{\theta}}$, $\hat{\phi}$ is the overdispersion parameter, k is the number of parameters, and n is the number of observations.

Value

Returns a number, either the QAIC or QBIC.

References

Gasparrini, A., B. Armstrong, and M. G. Kenward. "Distributed Lag Non-linear Models." *Statistics in Medicine* 29, no. 21 (2010): 2224–34. https://doi.org/10.1002/sim.3940.

Examples

```
fit <- glm(hp ~ mpg + disp + wt, family = "quasipoisson", data = mtcars)
AIC(fit)
QAIC(fit)
```

Index