

# Package "regressr"

## Overview/Synopsis

Create "regressr", an R library dedicated to helping users build regression models, providing text output to ease interpretation of the model's results, and optimizing model specifications. The purpose of this library is to make the process of regression modeling and coefficient interpretation easier for those unfamiliar with regressions.

## Use

Users who are not familiar with our regression methodologies can use these four functions to run regression analysis, and one function (function 5) that is an auxiliary function, but can be used outside of the context of the package.

- Function 1 - summarizer: Users input a dataset. The function returns with a table of variable descriptions and summary statistics, with instructions on choosing a dependent variable and a set of independent variable(s). The function also returns instructions on choosing dependent and independent variables for regression, using accessible language for those unfamiliar with regressions.
- Function 2 - dvaluator: Users input a dataset and dependent variable. The function checks the dependent variable for class and recommends an initial regression technique. For instance, if the dependent variable is binary, the function recommends a logit or probit regression as opposed to recommending OLS for a continuous dependent variable.
- Function 3 - interpreter: Users input a dataset and the formula of a regression. The function runs the regression and outputs the coefficients of predictors and the model's diagnostics, along with text interpretation for each parameter and explanations of the model's various diagnostics. The interpretation will be based on model specifications -- log-log, lin-log, quadratics. For instance, the function could describe what a coefficient means in terms of the user's independent and dependent variables, for example: "a one unit change in x correlates with a B unit change in y."
- Function 4 - optimizer: Users input a dependent variable and a set of potential independent variables. Function checks combinations of independent variables looking for those with the best model fit, such as the lowest error rate or highest adjusted R-squared, and outputs data frames sorted by the model fit statistic. The function also allows users to input options including what variables should be included in all specifications, whether to include quadratic terms, etc.
- Function 5 - package.testers: Simple function that tests whether the R instance on the computer has certain packages and either installs them and then runs them, or simply runs them.

## Installation

1. First, you will need to install the package "devtools" in order to install packages on github:

```
install.packages("devtools")
```

1. Load the devtool package:

```
library(devtools)
```

1. Install the regressr package using `install_github("author/package")`:

```
devtools::install_github("GeorgetownMcCourt/regressr-library-ghl")
```

1. Load the regressr package to use it: `library(regressr)`

## Usage

### summarizer

```
summarizer(df)
```

### dvaluator

```
dvaluator(depvar, order = F, panel = F)
```

### interpreter

```
interpreter <- function(modelType = "none", df, dependentVar,  
                        independentVar, logDepen = F,  
                        logIndepen = NULL, squareIndepend = NULL,  
                        detail = FALSE)
```

### optimizer

```
optimizer(df, depVar,  
          indepVar = colnames(df)[colnames(df) != depVar], include = NULL,  
          model = c("OLS", "binary probit", "binary logit", "ordered probit",  
                    "ordered logit", "multinomial logit"),  
          time.series = FALSE, time.var = NULL,  
          sqIndep = NULL)
```

### package.testers

```
package.testers(package)
```

# Progress Log

Write history in bullet form of what progress has been made and when. \* Proposal submitted, 3/31/2017 \* Project mock code agreed upon, 4/18/2017 \* Built a mock dataset, 4/15/2017 \* Created code that generates a dataset, 4/15/2017 \* Added more variable types to check code robustness. 4/22/2017 \* Ready, 4/27/2017 \* Function 1 - summarizer \* Started, 4/15/2017 \* Basic Code Structure Finalized, 4/19/2017 \* Finalized, 4/23/2017 \* Robustness Checks, in Progress 4/23/2017 \* Ready, 5/7/2017 \* Function 2 - possibleModels \* Started, 4/15/2017 \* Ready, 5/7/2017 \* Function 3 - interpreter started \* Started, 4/22/2017 \* Ready, 5/7/2017 \* Function 4 - optimizer \* Base function created, 4/20/2017 \* More user input options added, 4/23/2017

## Credits

Author: Justin Goss, Yixuan Huang, Nghia-Piotr Le

Credits: Jeff Chen, Dan Hammer :hammer:

## License

MIT License

Copyright (c) 2017 Goss, Huang, and Le

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.