Package: gilmour (via r-universe)

October 16, 2026

Type Package

Title The Interpretation of Adjusted Cp Statistic

Version 0.1.1

VignetteBuilder knitr

Suggests knitr, rmarkdown

Description Several methods may be found for selecting a subset of regressors from a set of k candidate variables in multiple linear regression. One possibility is to evaluate all possible regression models and comparing them using Mallows's Cp statistic (Cp) according to Gilmour original study. Full model is calculated, all possible combinations of regressors are generated, adjusted Cp for each submodel are computed, and the submodel with the minimum adjusted value Cp (ModelMin) is calculated. To identify the final model, the package applies a sequence of hypothesis tests on submodels nested within ModelMin, following the approach outlined in Gilmour's original paper. For more details see the help of the function final_model() and the original study (1996) <doi:10.2307/2348411>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.5)

NeedsCompilation no

Author Josef Dolejs [aut, cre] (ORCID: <https://orcid.org/0000-0002-3224-585X>)

Maintainer Josef Dolejs <josef.dolejs@uhk.cz>

Repository https://dolejs1993.r-universe.dev

Date/Publication 2025-07-10 13:50:02 UTC

RemoteUrl https://github.com/cran/gilmour

RemoteRef HEAD

RemoteSha e951ac8c94d584577a0e920e4a9268a3a843d737

Contents

crit . . . 2
EU2019 . . . 3
final_model . . . 3
Gilmour9p . . . 8
Modified_Gilmour9p . . . 9
Parks5p . . . 9
Patents5p . . . 10
submodels . . . 11
T1 . . . 14
T2 . . . 15
Trivial . . . 16

Index **17**

`crit` *This data contains critical values for testing at the significance level 5 Data published in the original article. Crit was taken from the original study (see Table2). Values in Crit were recalculated by Josef Dolejs for more possible steps of freedom*

Description

This data contains critical values for testing at the significance level 5 Data published in the original article. Crit was taken from the original study (see Table2). Values in Crit were recalculated by Josef Dolejs for more possible steps of freedom

Usage

```
crit
```

Format

An object of class `data.frame` with 28 rows and 31 columns.

Author(s)

Steven G. Gilmour and recalculated by Josef Dolejs for more possible steps of freedom

References

https://rss.onlinelibrary.wiley.com/doi/10.2307/2348411

`EU2019` *illustrative econometric data from Eurostat for 5 variables in 17 countries in 2019 columns: Country, LifExp , HDP, Unempl, Obesity, APassangers remove the first column: rownames(EU2019)= EU2019[,1]; EU2019=EU2019[,-1] data has real meaning*

Description

illustrative econometric data from Eurostat for 5 variables in 17 countries in 2019 columns: Country, LifExp , HDP, Unempl, Obesity, APassangers remove the first column: rownames(EU2019)= EU2019[,1]; EU2019=EU2019[,-1] data has real meaning

Usage

```
EU2019
```

Format

An object of class `data.frame` with 17 rows and 6 columns.

Author(s)

Josef Dolejs

References

Josef Dolejs

`final_model` *The function calculates all models and find final model according Gilmour*

Description

The original Gilmour study proposed a method for interpreting adjusted Cp statistic (Cp) to identify the best regression model when dealing with a relatively large number of numerical regressors and a smaller number of observations.
(The interpretation of Mallows's Cp-statistic.The Statistician. 1996. 45(1):49-56. doi:10.2307/2348411).
The procedure consists of two main steps (two diagrams are avaiable in the vignette of the package):
Initial Selection:
Adjusted $\bar{C}_p$ values are calculated for all possible combinations of regressors.
The submodel with the minimum of adjusted $\bar{C}_p$value is selected and labeled as ModelMin.
Reduction and Testing:
All submodels nested within ModelMin are considered.
Among them, the submodel with the lowest value of adjusted $\bar{C}_p$ is selected as a new candidate.

A hypothesis test is then performed where the null hypothesis states that ModelMin provides a better fit than the candidate submodel.
If the null hypothesis is not rejected, ModelMin is accepted as the final model.
If the null hypothesis is rejected, the candidate submodel becomes the new ModelMin, and the process is repeated with submodels nested within this new model.
The procedure continues iteratively until a null hypothesis is not rejected or until the so-called trivial model (i.e., a model using only the arithmetic mean) is reached.
Consequently, to identify the final model, the package applies a sequence of hypothesis tests on submodels nested within ModelMin, following the approach outlined in Gilmour's original paper.
The function final_model returns all regression results for:
the full model,
the selected ModelMin, and
the resulting final model with results of all tests.
Additionally, it outputs $\bar{C}_p$ values for all submodels, enabling users to easily generate a adjusted $\bar{C}_p$ vs. p plot (where p is the number of regressors+1), as recommended by Gilmour.
The input data should be provided as a data.frame, where the first column contains the numerical dependent variable, and the subsequent columns contain the regressors. The package also includes nine illustrative examples to demonstrate its functionality.
Adjusted $\bar{C}_p$ is calculated using the formula:
$\bar{C}_p = C_p - \frac{2(k-p+1)}{(n-k-3}$)
where k is the number of regressors in the full model, and p - 1 is the number of regressors in the submodel (thus, p includes the intercept term).
Further details and theoretical background can be found in the Gilmour's original paper.
For example, in the case of the trivial model (i.e., a model with only an intercept), the adjusted statistic is:
$\bar{C}_1 = C_1 - \frac{2(k-p+1)}{(n-k\check{}3)} = \frac{var(y)(n-1)}{MSE} - n + 2 - \frac{2(k-p+1)}{(n-k\check{}3)}$
where the function var() of sample variation is used for calculation of sum of squares in the trivial model and MSE is estimation of σ^2 .

Usage

```
final_model(d)
```

Arguments

d	dataframe

Value

y (numeric vector of dependend variable, the first column in data.frame input "d"); X (numeric matrix of regressors, the columns 2:k in data.frame input "d"); n number of observations (numeric value, the number of observations in "d"); k number of regressors (numeric value); t step of freedom (t=n-k-1, numeric value); regressors (set of strings of regressors); expression of full_model (string); regression results of full model (list calculated using lm()); MSE (characteristic in full model, numeric value); submodels_number (number of all possible submodels, numeric value); Cq_plus_1A (numeric value, Cp characteristic in model_min with the minimal adjusted Cp); q (level of model_min, number of regressors in model_min, numeric value); submodels (matrix with four columns: row number, p, Cp,submodel); model_min (textual expression of model min, string); regression results of model_min (results in model min, list calculated using lm()); FinalCp (Cp

characteristic in final_model, numeric value); expression of final_model (textual expression of final model, string); regression results of final_model (results in final model, list calculated using lm()).

Examples

```
mtcars ; MyResults=final_model(mtcars) # well known data in R
# the red points correspond to the starting submodel in the testing process
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp)) ; FinalCp= MyResults$FinalCp
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( ( round(yCp,4)== round(min(yCp),4)| round(yCp,4)== round(FinalCp,4) ),
 "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

MyResults$full_model
summary(MyResults$full_model_results)
MyResults$final_model
summary(MyResults$final_model_results)


# 8 tables are avaiable for illustration of the function final_model
# Gilmour9p, T1, T2, Trivial
# Modified_Gilmour9p, Parks5p, Patents5p and EU2019
# ilustrative data from the original Gilmour paper Gilmour9p
Gilmour9p ; MyResults=final_model(Gilmour9p)
# the red points correspond to the starting submodel in the testing process
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp)) ; FinalCp= MyResults$FinalCp
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
col=ifelse( ( round(yCp,4)== round(min(yCp),4)| round(yCp,4)== round(FinalCp,4) ),
"red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)


MyResults$full_model
summary(MyResults$full_model_results)
MyResults$final_model
summary(MyResults$final_model_results)

# T1 contains 12 regressors and 16 observations, simulated data without real meaning
# more submodels and calculation takes about 5 seconds
# the null hypothesis is not rejected in the first test
# it illustrates the situation if full model is final model
# data has no practical meaning
T1; MyResults=final_model(T1)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
```

```
YRange=c( ymin ,1.5*max(xp)) ; FinalCp= MyResults$FinalCp
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
col=ifelse( ( round(yCp,4)== round(min(yCp),4)| round(yCp,4)== round(FinalCp,4) ),
"red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

MyResults$full_model
summary(MyResults$full_model_results)
MyResults$final_model
summary(MyResults$final_model_results)

# T2 illustrates the situation if
# the loop of tests is finished by the Trivial model
# data has no practical meaning
T2; MyResults=final_model(T2)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp)) ; FinalCp= MyResults$FinalCp
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( ( round(yCp,4)== round(min(yCp),4)| round(yCp,4)== round(FinalCp,4) ),
 "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

MyResults$full_model
summary(MyResults$full_model_results)
MyResults$final_model
summary(MyResults$final_model_results)

# Trivial contains illustrates the situation if
# the Trivial model is model_min without testing process
# data has no practical meaning
Trivial; MyResults=final_model(Trivial)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp)) ; FinalCp= MyResults$FinalCp
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( ( round(yCp,4)== round(min(yCp),4)| round(yCp,4)== round(FinalCp,4) ),
 "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

MyResults$full_model
summary(MyResults$full_model_results)
MyResults$final_model
summary(MyResults$final_model_results)

# special illustrative data if more than two tests
# are done in the loop in the function final_model()
```

```
# original Gilmour table was modified
# data has no practical meaning
Modified_Gilmour9p ; MyResults=final_model(Modified_Gilmour9p)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp)) ; FinalCp= MyResults$FinalCp
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( ( round(yCp,4)== round(min(yCp),4)| round(yCp,4)== round(FinalCp,4) ),
 "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)


MyResults$full_model
summary(MyResults$full_model_results)
MyResults$final_model
summary(MyResults$final_model_results)

# number of visitors in parks, citation:
# Stemberk Josef, Josef Dolejs, Petra Maresova, Kamil Kuca.
# Factors affecting the number of Visitors in National Parks
# in the Czech Republic, Germany and Austria.
# International Journal of Geo-Information. http://www.mdpi.com/2220-9964/7/3/124
# ISPRS Int. J. Geo-Inf. 2018, 7(3), 124; doi:10.3390/ijgi7030124
d=Parks5p ; rownames(d)= d[,1]; d=d[,-1]; d; MyResults=final_model(d); rm(d)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp)) ; FinalCp= MyResults$FinalCp
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( ( round(yCp,4)== round(min(yCp),4)| round(yCp,4)== round(FinalCp,4) ),
 "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

MyResults$full_model
summary(MyResults$full_model_results)
MyResults$final_model
summary(MyResults$final_model_results)

# number of patents in universities (see column names - regressors)
# Perspective and Suitable Research Area in
# Public Research-Patent Analysis of the Czech Public Universities
# Education and Urban Society, 54(7), https://doi.org/10.1177/00131245211027362
# Maresova Petra, Soukal Ivan, Stemberkova Ruzena, Selamat Ali
Patents5p; MyResults=final_model(Patents5p)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp)) ; FinalCp= MyResults$FinalCp
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( ( round(yCp,4)== round(min(yCp),4)| round(yCp,4)== round(FinalCp,4) ),
 "red", "darkblue")  )
```

```
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

MyResults$full_model
summary(MyResults$full_model_results)
MyResults$final_model
summary(MyResults$final_model_results)

# illustrative econometric data from Eurostat for 5 variables in 17 countries in 2019
# columns: Country, LifExp , HDP, Unempl, Obesity, APassangers
d=EU2019 ; d
rownames(d)= d[,1]; d=d[,-1]; d  # the same data without the first column (country names)
MyResults=final_model(d)
# plot of all Cp, the red line is the relationship: "Cp ~ p"
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
FinalCp= MyResults$FinalCp
# plot without y limits "ylim=c( ymin ,1.5*max(xp)"
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="",
 col=ifelse( ( round(yCp,4)== round(min(yCp),4)| round(yCp,4)== round(FinalCp,4) ),
 "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

MyResults$full_model
summary(MyResults$full_model_results)
MyResults$final_model
summary(MyResults$final_model_results)
```

`Gilmour9p` *Ilustrative data published in the original article (House prices) Data with the practical meaning Gilmour9p was taken from the original study (see Table1).*

Description

Ilustrative data published in the original article (House prices) Data with the practical meaning Gilmour9p was taken from the original study (see Table1).

Usage

```
Gilmour9p
```

Format

An object of class `data.frame` with 24 rows and 10 columns.

Author(s)

Steven G. Gilmour

References

https://rss.onlinelibrary.wiley.com/doi/10.2307/2348411

`Modified_Gilmour9p` *special illustrative data if more than two tests are done in the loop in the function final_model() original Gilmour table was modified data has no real meaning*

Description

special illustrative data if more than two tests are done in the loop in the function final_model() original Gilmour table was modified data has no real meaning

Usage

```
Modified_Gilmour9p
```

Format

An object of class `data.frame` with 24 rows and 10 columns.

Author(s)

Josef Dolejs

References

Josef Dolejs

`Parks5p` *number of visitors in parks, citation: Stemberk Josef, Josef Dolejs, Petra Maresova, Kamil Kuca. Factors affecting the number of Visitors in National Parks in the Czech Republic, Germany and Austria. International Journal of Geo-Information. ISPRS Int. J. Geo-Inf. 2018, 7(3), 124; doi:10.3390/ijgi7030124 data has real meaning*

Description

number of visitors in parks, citation: Stemberk Josef, Josef Dolejs, Petra Maresova, Kamil Kuca. Factors affecting the number of Visitors in National Parks in the Czech Republic, Germany and Austria. International Journal of Geo-Information. ISPRS Int. J. Geo-Inf. 2018, 7(3), 124; doi:10.3390/ijgi7030124 data has real meaning

Usage

```
Parks5p
```

Format

An object of class `data.frame` with 12 rows and 7 columns.

Author(s)

Stemberk Josef, Josef Dolejs, Petra Maresova, Kamil Kuca

References

https://www.mdpi.com/2220-9964/7/3/124

`Patents5p` *number of patents in universities, citation: Maresova Petra, Soukal Ivan, Stemberkova Ruzena, Selamat Ali Perspective and Suitable Research Area in Public Research-Patent Analysis of the Czech Public Universities Education and Urban Society, 54(7), Maresova Petra, Soukal Ivan, Stemberkova Ruzena, Selamat Ali data has real meaning*

Description

number of patents in universities, citation: Maresova Petra, Soukal Ivan, Stemberkova Ruzena, Selamat Ali Perspective and Suitable Research Area in Public Research-Patent Analysis of the Czech Public Universities Education and Urban Society, 54(7), Maresova Petra, Soukal Ivan, Stemberkova Ruzena, Selamat Ali data has real meaning

Usage

```
Patents5p
```

Format

An object of class `data.frame` with 11 rows and 6 columns.

Author(s)

Maresova Petra, Soukal Ivan, Stemberkova Ruzena, Selamat Ali

References

doi:10.1177/00131245211027362

`submodels` *The function calculates all possible submodels and find model with the minimal value of Cp according Gilmour*

Description

The original Gilmour study proposed a method for interpreting adjusted Cp statistic (Cp) to identify the best regression model when dealing with a relatively large number of numerical regressors and a smaller number of observations.
(The interpretation of Mallows's Cp-statistic. The Statistician. 1996. 45(1):49-56. doi:10.2307/2348411)
The procedure consists of two main steps (two diagrams are avaiable in the vignette of the package):
Initial Selection:
Adjusted $\bar{C}_p$ values are calculated for all possible combinations of regressors.
The submodel with the minimum of adjusted $\bar{C}_p$value is selected and labeled as ModelMin.
Reduction and Testing (it is only in the function 'final_model') :
All submodels nested within ModelMin are considered.
Among them, the submodel with the lowest value of adjusted $\bar{C}_p$ is selected as a new candidate.
A hypothesis test is then performed where the null hypothesis states that ModelMin provides a better fit than the candidate submodel.
If the null hypothesis is not rejected, ModelMin is accepted as the 'final model'.
If the null hypothesis is rejected, the candidate submodel becomes the new ModelMin, and the process is repeated with submodels nested within this new model.
The procedure continues iteratively until a null hypothesis is not rejected or until the so-called trivial model (i.e., a model using only the arithmetic mean) is reached.
The second step is procced only in the function 'final model'
The function returns all regression results for:
the full model and the ModelMin.
Additionally, it outputs $\bar{C}_p$ values for all submodels, enabling users to easily generate a adjusted $\bar{C}_p$ vs. p plot (where p is the number of regressors+1), as recommended by Gilmour.
The input data should be provided as a data.frame, where the first column contains the numerical dependent variable, and the subsequent columns contain the regressors. The package also includes nine illustrative examples to demonstrate its functionality.
Adjusted $\bar{C}_p$ is calculated using the formula:
$\bar{C}_p = C_p - \frac{2(k-p+1)}{(n-k-3}$)
where k is the number of regressors in the full model, and p - 1 is the number of regressors in the submodel (thus, p includes the intercept term).
Further details and theoretical background can be found in the Gilmour's original paper.
For example, in the case of the trivial model (i.e., a model with only an intercept), the adjusted statistic is:
$\bar{C}_1 = C_1 - \frac{2(k-p+1)}{(n-k\tilde{}3)} = \frac{var(y)(n-1)}{MSE} - n + 2 - \frac{2(k-p+1)}{(n-k\tilde{}3)}$
where the function var() of sample variation is used for calculation of sum of squares in the trivial model and MSE is estimation of σ^2 .

Usage

```
submodels(d)
```

Arguments

d	dataframe

Value

y (numeric vector of dependend variable, the first column in data.frame input "d"); X (numeric matrix of regressors, the columns 2:k in data.frame input "d"); n number of observations (numeric value, the number of observations in "d"); k number of regressors (numeric value); t step of freedom (t=n-k-1, numeric value); regressors (set of strings of regressors); expression of full_model (string); regression results of full model (list calculated using lm()); MSE (characteristic in full model, numeric value); submodels_number (number of all possible submodels, numeric value); Cq_plus_1A (numeric value, Cp characteristic in model_min with the minimal adjusted Cp); q (level of model_min, number of regressors in model_min, numeric value); submodels (matrix with four columns: row number, p, Cp,submodel); model_min (textual expression of model min, string); regression results of model_min (results in model min, list calculated using lm()).

Examples

```
# the standard well known data
d=mtcars ; d ; MyResults=submodels(d)
# the red points correspond to the starting submodel in the testing process
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp))
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( round(yCp,4)== round(min(yCp),4), "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)




# 8 tables are avaiable for illustration of the functions 'submodels' and 'final_model'

# ilustrative data from the original Gilmour paper
Gilmour9p;MyResults=submodels(Gilmour9p)
# the red points correspond to the starting submodel in the testing process
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp))
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( round(yCp,4)== round(min(yCp),4), "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

# 12 regressors and 16 observations, simulated data without real meaning
# more submodels and calculation takes about 5 seconds
# the null hypothesis is not rejected in the first test
T1 ; MyResults=submodels(T1)
```

```
# the red points correspond to the starting submodel in the testing process
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp))
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( round(yCp,4)== round(min(yCp),4), "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

# the loop is finished by the Trivial model for data T2
T2 ; MyResults=submodels(T2)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp))
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( round(yCp,4)== round(min(yCp),4), "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

# Trivial is illustrative data in which Trivial model is model_min without testing process
Trivial ; MyResults=submodels(Trivial)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp))
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( round(yCp,4)== round(min(yCp),4), "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

# special illustrative data for more than two tests in the loop in the function 'final_model'
Modified_Gilmour9p ; MyResults=submodels(Modified_Gilmour9p)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp))
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( round(yCp,4)== round(min(yCp),4), "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

# number of visitors in parks
# citation: Stemberk Josef, Josef Dolejs, Petra Maresova, Kamil Kuca.
# Factors affecting the number of Visitors in National Parks
# in the Czech Republic, Germany and Austria.
# International Journal of Geo-Information. https://www.mdpi.com/2220-9964/7/3/124
# ISPRS Int. J. Geo-Inf. 2018, 7(3), 124; doi:10.3390/ijgi7030124
d=Parks5p ; rownames(d)= d[,1]; d=d[,-1]; d
MyResults=submodels(d)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
```

```
YRange=c( ymin ,1.5*max(xp))
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( round(yCp,4)== round(min(yCp),4), "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

# number of patents in universities (see column names - regressors)
# citation: Perspective and Suitable Research Area
# in Public Research-Patent Analysis of the Czech Public Universities
# Education and Urban Society, 54(7), https://doi.org/10.1177/00131245211027362
Patents5p ; MyResults=submodels(Patents5p)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
ymin= ifelse(min(yCp)<0, 1.1* min(yCp), 0.9* min(yCp))
YRange=c( ymin ,1.5*max(xp))
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="", ylim=YRange ,
 col=ifelse( round(yCp,4)== round(min(yCp),4), "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)

# illustrative econometric data from Eurostat for 5 variables in 17 countries in 2019
# columns: LifExp , HDP, Unempl, Obesity, APassangers
d= EU2019
rownames(d)= d[,1]; d=d[,-1]; d  # the same data without the first column (country names)
MyResults=submodels(d)
yCp= as.numeric(MyResults$submodels[,3]) ; xp= as.numeric(MyResults$submodels[,2])
# plot without y limits "ylim=c( ymin ,1.5*max(xp)"
plot(yCp ~ xp, xlab="Number of Parameters in Submodel",ylab="",
 col=ifelse( round(yCp,4)== round(min(yCp),4), "red", "darkblue")  )
lines(xp, xp, col="red")
mtext(bquote(paste( bar(C) , "p")), side=2, line=3, padj=1, cex=1.2)
mtext(bquote(paste("All Submodels: ",bar(C),"p ~ p")), side=3, line=3, padj=1, cex=1.2)
```

T1 *T1 contains simulated data without real meaning the null hypothesis is not rejected in the first test it illustrates the situation if full model is final model data has no real meaning*

Description

T1 contains simulated data without real meaning the null hypothesis is not rejected in the first test it illustrates the situation if full model is final model data has no real meaning

Usage

```
T1
```

Format

An object of class `data.frame` with 16 rows and 13 columns.

Author(s)

Josef Dolejs

References

Josef Dolejs

`T2` *T2 illustrates the situation if the loop of tests is finished by the Trivial model data has no practical meaning*

Description

T2 illustrates the situation if the loop of tests is finished by the Trivial model data has no practical meaning

Usage

```
T2
```

Format

An object of class `data.frame` with 16 rows and 6 columns.

Author(s)

Josef Dolejs

References

Josef Dolejs

`Trivial` *Trivial illustrates the situation if the Trivial model is model_min without testing process data has no real meaning*

Description

Trivial illustrates the situation if the Trivial model is model_min without testing process data has no real meaning

Usage

```
Trivial
```

Format

An object of class `data.frame` with 16 rows and 3 columns.

Author(s)

Josef Dolejs

References

Josef Dolejs

Index

∗ **data**
crit, 2
EU2019, 3
Gilmour9p, 8
Modified_Gilmour9p, 9
Parks5p, 9
Patents5p, 10
T1, 14
T2, 15
Trivial, 16

crit, 2

EU2019, 3

final_model, 3

Gilmour9p, 8

Modified_Gilmour9p, 9

Parks5p, 9
Patents5p, 10

submodels, 11

T1, 14
T2, 15
Trivial, 16