

```

In [1]: #Glm Test for P-value

# Load your data
data <- read.csv("Pilot_modified_data_1.csv")

# Define your variables
independent_variables <- c('BFI..E.', 'BFI..A.', 'BFI..C.', 'BFI..N.', 'BFI..O.', '
mediator_variable <- 'PAQ'
dependent_variables <- c('Panas..', 'Panas...1', 'CBCL')

# Function to calculate star ratings
get_star_rating <- function(p_value) {
  if (p_value < 0.001) return("***")
  else if (p_value < 0.01) return("**")
  else if (p_value < 0.05) return("*")
  else return("check")
}

# Fit separate Poisson GLM models for each dependent variable
for (dependent_variable in dependent_variables) {
  formula <- paste(dependent_variable, "~ . +", mediator_variable)
  poisson_model <- glm(formula, data = data, family = poisson(link = "log"))

  # Summarize the model
  cat("Summary for", dependent_variable, ":\n")
  model_summary <- summary(poisson_model)
  print(model_summary)

  # Get and print star ratings for p-values
  p_values <- model_summary$coefficients[, "Pr(>|z|)"]
  star_ratings <- sapply(p_values, get_star_rating)
  star_ratings_df <- data.frame(Coefficient = rownames(model_summary$coefficients),
  print(star_ratings_df)
  print("=====")
}

```

Summary for Panas.. :

Call:

```
glm(formula = formula, family = poisson(link = "log"), data = data)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.127254	-0.030803	-0.004841	0.030804	0.236661

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.8376069	2.7176884	0.308	0.758
Panas...1	-0.0035951	0.2941212	-0.012	0.990
BFI..E.	-0.0490939	0.4710388	-0.104	0.917
BFI..A.	0.0071564	0.2813401	0.025	0.980
BFI..C.	0.0366116	0.4633167	0.079	0.937
BFI..N.	0.0525955	0.2210784	0.238	0.812
BFI..O.	-0.0122170	0.3629494	-0.034	0.973
EI..Self.A.	0.0008079	0.0183715	0.044	0.965
EI..Self.M.	-0.0021908	0.0143532	-0.153	0.879
EI..Social.A.	0.0023679	0.0235086	0.101	0.920
EI..RM.	-0.0001474	0.0164597	-0.009	0.993
Total	-0.0173123	0.3458867	-0.050	0.960
PAQ	0.0146337	0.2688049	0.054	0.957
CBCL	-0.1406602	0.4055748	-0.347	0.729

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 0.41816 on 33 degrees of freedom  
 Residual deviance: 0.15941 on 20 degrees of freedom  
 AIC: 117.4

Number of Fisher Scoring iterations: 3

	Coefficient	Stars
(Intercept)	(Intercept)	check
Panas...1	Panas...1	check
BFI..E.	BFI..E.	check
BFI..A.	BFI..A.	check
BFI..C.	BFI..C.	check
BFI..N.	BFI..N.	check
BFI..O.	BFI..O.	check
EI..Self.A.	EI..Self.A.	check
EI..Self.M.	EI..Self.M.	check
EI..Social.A.	EI..Social.A.	check
EI..RM.	EI..RM.	check
Total	Total	check
PAQ	PAQ	check
CBCL	CBCL	check

[1] "=====

Summary for Panas...1 :

Call:

```
glm(formula = formula, family = poisson(link = "log"), data = data)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.6553	-0.3270	-0.0017	0.2423	0.5873

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.889891	4.349013	0.205	0.838
Panas..	-0.149219	1.404358	-0.106	0.915
BFI..E.	-0.146126	0.531573	-0.275	0.783
BFI..A.	-0.033461	0.324314	-0.103	0.918
BFI..C.	0.154399	0.543706	0.284	0.776
BFI..N.	0.062337	0.295745	0.211	0.833
BFI..O.	-0.280332	0.419729	-0.668	0.504
EI..Self.A.	0.003700	0.020197	0.183	0.855
EI..Self.M.	-0.000276	0.017381	-0.016	0.987
EI..Social.A.	0.002127	0.027020	0.079	0.937
EI..RM.	-0.007422	0.017756	-0.418	0.676
Total	0.060747	0.382359	0.159	0.874
PAQ	0.082828	0.304525	0.272	0.786
CBCL	0.089480	0.656400	0.136	0.892

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 5.7187 on 33 degrees of freedom  
 Residual deviance: 3.8829 on 20 degrees of freedom  
 AIC: 110.93

Number of Fisher Scoring iterations: 4

	Coefficient	Stars
(Intercept)	(Intercept)	check
Panas..	Panas..	check
BFI..E.	BFI..E.	check
BFI..A.	BFI..A.	check
BFI..C.	BFI..C.	check
BFI..N.	BFI..N.	check
BFI..O.	BFI..O.	check
EI..Self.A.	EI..Self.A.	check
EI..Self.M.	EI..Self.M.	check
EI..Social.A.	EI..Social.A.	check
EI..RM.	EI..RM.	check
Total	Total	check
PAQ	PAQ	check
CBCL	CBCL	check

[1] "=====

Summary for CBCL :

Call:

glm(formula = formula, family = poisson(link = "log"), data = data)

Deviance Residuals:

Min	1Q	Median	3Q	Max
-0.60355	-0.06141	0.02928	0.07397	0.28235

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	2.104e+00	3.339e+00	0.630	0.529

Panas..	-8.260e-01	1.114e+00	-0.741	0.459
Panas...1	2.974e-02	3.034e-01	0.098	0.922
BFI..E.	-1.124e-01	4.808e-01	-0.234	0.815
BFI..A.	-4.170e-02	2.832e-01	-0.147	0.883
BFI..C.	1.188e-01	4.719e-01	0.252	0.801
BFI..N.	9.358e-02	2.512e-01	0.373	0.709
BFI..O.	-7.562e-02	3.670e-01	-0.206	0.837
EI..Self.A.	1.366e-04	1.894e-02	0.007	0.994
EI..Self.M.	-3.774e-03	1.527e-02	-0.247	0.805
EI..Social.A.	7.694e-03	2.340e-02	0.329	0.742
EI..RM.	-1.222e-05	1.706e-02	-0.001	0.999
Total	-9.119e-02	3.408e-01	-0.268	0.789
PAQ	6.153e-02	2.703e-01	0.228	0.820

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 2.2148 on 33 degrees of freedom  
 Residual deviance: 1.1489 on 20 degrees of freedom  
 AIC: 115.56

Number of Fisher Scoring iterations: 4

	Coefficient	Stars
(Intercept)	(Intercept)	check
Panas..	Panas..	check
Panas...1	Panas...1	check
BFI..E.	BFI..E.	check
BFI..A.	BFI..A.	check
BFI..C.	BFI..C.	check
BFI..N.	BFI..N.	check
BFI..O.	BFI..O.	check
EI..Self.A.	EI..Self.A.	check
EI..Self.M.	EI..Self.M.	check
EI..Social.A.	EI..Social.A.	check
EI..RM.	EI..RM.	check
Total	Total	check
PAQ	PAQ	check

[1] "=====

```
In [2]: #Test for k-fold cross validation in R

# Load necessary libraries
library(caret)

# Assuming you have already performed the GLM analysis and have the data in a variable
# Set the number of folds for cross-validation (e.g., 5-fold)
num_folds <- 5

# Create a cross-validation control object
cv <- trainControl(method = "cv", number = num_folds)

# Define the formula for the model (e.g., Panas.. ~ . + PAQ)
formula <- as.formula("Panas.. ~ . + PAQ")

# Fit the GLM model with 5-fold cross-validation
```

```
glm_cv <- train(formula, data = data, method = "glm", family = poisson(link = "log")
# View cross-validation results
print(glm_cv)
```

Loading required package: ggplot2

Loading required package: lattice

Warning message in train.default(x, y, weights = w, ...):

“You are trying to do regression and your outcome only has two possible values Are you trying to do classification? If so, use a 2 level factor as your outcome column.”

Warning message in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
“There were missing values in resampled performance measures.”

Generalized Linear Model

34 samples

13 predictors

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 27, 28, 27, 27, 27

Resampling results:

RMSE	Rsquared	MAE
0.2632421	0.4166667	0.161648

In [3]: *#Main Code*

```
#SOURCES ::
# https://www.statology.org/k-fold-cross-validation-in-r/
# https://www.geeksforgeeks.org/k-fold-cross-validation-in-r-programming/
# http://www.sthda.com/english/articles/38-regression-model-validation/157-cross-validation
# https://cran.r-project.org/web/packages/caret/vignettes/caret.html
# https://www.youtube.com/watch?v=CmOZy8WKpTM

# Load necessary libraries
library(caret)

# Assuming you have already performed the GLM analysis and have the data in a variable

# Set the number of folds for cross-validation (e.g., 5-fold)
num_folds <- 5

# Create a cross-validation control object
cv <- trainControl(method = "cv", number = num_folds)

# Define the independent variables and mediator variable
independent_variables <- c('BFI..E.', 'BFI..A.', 'BFI..C.', 'BFI..N.', 'BFI..O.', 'BFI..P.')
mediator_variable <- 'PAQ'

# Define a list of dependent variables
```

```

dependent_variables <- c('Panas..', 'Panas...1', 'CBCL')

# Loop through each dependent variable
for (dependent_variable in dependent_variables) {
  # Define the formula for the model
  formula <- as.formula(paste(dependent_variable, "~ . +", mediator_variable))

  # Fit the GLM model with 5-fold cross-validation
  glm_cv <- train(formula, data = data, method = "glm", family = poisson(link = "log"))

  # View cross-validation results for each dependent variable
  cat("Cross-validation results for", dependent_variable, ":\n")
  print(glm_cv)
  cat("=====\n")
}

```

Warning message in train.default(x, y, weights = w, ...):  
 "You are trying to do regression and your outcome only has two possible values Are you trying to do classification? If so, use a 2 level factor as your outcome column."

Warning message in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
 "There were missing values in resampled performance measures."

Cross-validation results for Panas.. :  
 Generalized Linear Model

34 samples

13 predictors

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 27, 27, 27, 27, 28

Resampling results:

RMSE	Rsquared	MAE
0.316962	0.775641	0.1954197

=====  
 Warning message in train.default(x, y, weights = w, ...):  
 "You are trying to do regression and your outcome only has two possible values Are you trying to do classification? If so, use a 2 level factor as your outcome column."

Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == "se") "se" else "fitted", :  
 "prediction from a rank-deficient fit may be misleading"

Cross-validation results for Panas...1 :  
Generalized Linear Model

34 samples  
13 predictors

No pre-processing  
Resampling: Cross-Validated (5 fold)  
Summary of sample sizes: 27, 27, 28, 27, 27  
Resampling results:

RMSE	Rsquared	MAE
0.8321274	0.1950006	0.7156902

```
=====
Warning message in train.default(x, y, weights = w, ...):
" You are trying to do regression and your outcome only has two possible values Are
you trying to do classification? If so, use a 2 level factor as your outcome colum
n."
Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
"prediction from a rank-deficient fit may be misleading"
Warning message in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainI
nfo, :
"There were missing values in resampled performance measures."
Cross-validation results for CBCL :
Generalized Linear Model
```

34 samples  
13 predictors

No pre-processing  
Resampling: Cross-Validated (5 fold)  
Summary of sample sizes: 27, 27, 27, 28, 27  
Resampling results:

RMSE	Rsquared	MAE
0.7018955	0.03946961	0.4665882

```
=====
```

```
In [4]: #Code to Check the confidence intervals of the K-fold Cross Validation

# Load necessary libraries
library(caret)
data <- read.csv("Pilot_modified_data_1.csv")

# Assuming you have already performed the GLM analysis and have the data in a varia

# Set the number of folds for cross-validation (e.g., 5-fold)
num_folds <- 5

# Create a cross-validation control object
cv <- trainControl(method = "cv", number = num_folds)

# Define the independent variables and mediator variable
```

```

independent_variables <- c('BFI..E.', 'BFI..A.', 'BFI..C.', 'BFI..N.', 'BFI..O.', '
mediator_variable <- 'PAQ'

# Define a list of dependent variables
dependent_variables <- c('Panas..', 'Panas...1', 'CBCL')

# Loop through each dependent variable
for (dependent_variable in dependent_variables) {
  # Define the formula for the model
  formula <- as.formula(paste(dependent_variable, "~ . +", mediator_variable))

  # Fit the GLM model with 5-fold cross-validation
  glm_cv <- train(formula, data = data, method = "glm", family = poisson(link = "logit"))

  # View cross-validation results for each dependent variable
  cat("Cross-validation results for", dependent_variable, ":\n")
  print(glm_cv)

  # Get and print confidence intervals
  confidence_level <- 0.95 # You can set the desired confidence level
  conf_intervals <- confint(glm_cv$finalModel, level = confidence_level)
  cat("Confidence Intervals (", confidence_level * 100, "%):\n")
  print(conf_intervals)

  cat("=====\n")
}

```

Warning message in train.default(x, y, weights = w, ...):  
 "You are trying to do regression and your outcome only has two possible values Are you trying to do classification? If so, use a 2 level factor as your outcome column."

Warning message in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :  
 "There were missing values in resampled performance measures."

Cross-validation results for Panas.. :  
 Generalized Linear Model

34 samples  
 13 predictors

No pre-processing  
 Resampling: Cross-Validated (5 fold)  
 Summary of sample sizes: 27, 27, 28, 27, 27  
 Resampling results:

RMSE	Rsquared	MAE
0.2221788	0.25	0.1459224

Waiting for profiling to be done...



Confidence Intervals ( 95 %):

	2.5 %	97.5 %
(Intercept)	-4.52962974	6.19383713
Panas...1	-0.58139715	0.57591853
BFI..E.	-0.94538271	0.91902446
BFI..A.	-0.55155455	0.55688115
BFI..C.	-0.87269423	0.95169455
BFI..N.	-0.38116926	0.48957416
BFI..O.	-0.74325364	0.68585922
EI..Self.A.	-0.03535082	0.03736765
EI..Self.M.	-0.03086436	0.02567329
EI..Social.A.	-0.04459332	0.04791043
EI..RM.	-0.03208577	0.03272308
Total	-0.70616649	0.66038196
PAQ	-0.52554052	0.53362686
CBCL	-0.90081535	0.70831595

=====

Warning message in train.default(x, y, weights = w, ...):

“You are trying to do regression and your outcome only has two possible values Are you trying to do classification? If so, use a 2 level factor as your outcome column.”

Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :

“prediction from a rank-deficient fit may be misleading”

Cross-validation results for Panas...1 :

Generalized Linear Model

34 samples

13 predictors

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 28, 27, 27, 27, 27

Resampling results:

RMSE	Rsquared	MAE
0.7501959	0.1720654	0.624577

Waiting for profiling to be done...

Confidence Intervals ( 95 %):

	2.5 %	97.5 %
(Intercept)	-7.67581222	9.78392888
Panas..	-3.44866963	2.44509983
BFI..E.	-1.16637525	0.94183865
BFI..A.	-0.68201205	0.59768039
BFI..C.	-0.90784178	1.23864245
BFI..N.	-0.50794309	0.66014285
BFI..O.	-1.14248521	0.51574656
EI..Self.A.	-0.03630188	0.04361503
EI..Self.M.	-0.03499845	0.03358241
EI..Social.A.	-0.05176683	0.05481635
EI..RM.	-0.04192160	0.02815058
Total	-0.69532455	0.81425853
PAQ	-0.52931454	0.67185137
CBCL	-1.13737476	1.48942119

=====

Warning message in train.default(x, y, weights = w, ...):

“You are trying to do regression and your outcome only has two possible values Are you trying to do classification? If so, use a 2 level factor as your outcome column.”

Warning message in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :

“prediction from a rank-deficient fit may be misleading”

Warning message in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :

“There were missing values in resampled performance measures.”

Cross-validation results for CBCL :

Generalized Linear Model

34 samples

13 predictors

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 27, 27, 27, 27, 28

Resampling results:

RMSE	Rsquared	MAE
0.7841941	0.09133986	0.5067433

Waiting for profiling to be done...

Confidence Intervals ( 95 %):

	2.5 %	97.5 %
(Intercept)	-4.20614399	9.44624851
Panas..	-3.80116832	1.01994749
Panas...1	-0.56454388	0.62989737
BFI..E.	-1.03346545	0.87317891
BFI..A.	-0.60788946	0.50934793
BFI..C.	-0.81252554	1.04477742
BFI..N.	-0.39524798	0.59674386
BFI..O.	-0.81288041	0.63220966
EI..Self.A.	-0.03735695	0.03758648
EI..Self.M.	-0.03413890	0.02604045
EI..Social.A.	-0.03917482	0.05293202
EI..RM.	-0.03304823	0.03415683
Total	-0.76991674	0.57731010
PAQ	-0.48363315	0.58198734

=====

In [ ]: