

```
In [1]: install.packages("mediation")
library(mediation)

# Sample data frame 'data' with columns 'Panas..', 'Panas...1', 'BFI..E.', 'BFI..A.'
data <- read.csv("Pilot_modified_data_1.csv")
```

also installing the dependencies 'diffobj', 'brio', 'desc', 'pkgload', 'praise', 'processx', 'waldo', 'checkmate', 'testthat', 'gridExtra', 'htmlTable', 'viridis', 'Formula', 'minqa', 'nloptr', 'mvtnorm', 'sandwich', 'lpSolve', 'Hmisc', 'lme4'

Updating HTML index of packages in '.Library'

Making 'packages.html' ...
done

Loading required package: MASS

Loading required package: Matrix

Loading required package: mvtnorm

Loading required package: sandwich

mediation: Causal Mediation Analysis
Version: 4.5.0

```
In [3]: model.0 <- lm(Panas.. ~ BFI..E., data)
summary(model.0)
```

Call:

lm(formula = Panas.. ~ BFI..E., data = data)

Residuals:

Min	1Q	Median	3Q	Max
-0.0396	-0.0297	-0.0297	-0.0297	0.9703

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.009901	0.201318	9.984	2.36e-11 ***
BFI..E.	0.009901	0.101031	0.098	0.923

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1741 on 32 degrees of freedom

Multiple R-squared: 0.0003, Adjusted R-squared: -0.03094

F-statistic: 0.009604 on 1 and 32 DF, p-value: 0.9225

```
In [4]: model.M <- lm(PAQ ~ BFI..E., data)
summary(model.M)
```

Call:

```
lm(formula = PAQ ~ BFI..E., data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.8317	-0.6238	0.3762	0.3762	1.3762

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.2079	0.7620	1.585	0.123
BFI..E.	0.2079	0.3824	0.544	0.590

Residual standard error: 0.6591 on 32 degrees of freedom

Multiple R-squared: 0.009154, Adjusted R-squared: -0.02181

F-statistic: 0.2956 on 1 and 32 DF, p-value: 0.5904

```
In [5]: model.Y <- lm(Panas.. ~ BFI..E. + PAQ, data)
summary(model.Y)
```

Call:

```
lm(formula = Panas.. ~ BFI..E. + PAQ, data = data)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.06695	-0.03989	-0.02849	-0.01282	0.96011

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.977208	0.211301	9.357	1.53e-10 ***
BFI..E.	0.004274	0.102578	0.042	0.967
PAQ	0.027066	0.047202	0.573	0.571

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.176 on 31 degrees of freedom

Multiple R-squared: 0.01079, Adjusted R-squared: -0.05303

F-statistic: 0.1691 on 2 and 31 DF, p-value: 0.8452

```
In [6]: results <- mediate(model.M, model.Y, treat='BFI..E.', mediator='PAQ',
boot=TRUE, sims=500)
summary(results)
```

Running nonparametric bootstrap

```
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
```

[illegible]

[illegible]

```
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
```

```
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
```

```

t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.
c):
"prediction from a rank-deficient fit may be misleading"
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.
t):
"prediction from a rank-deficient fit may be misleading"

```



```
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.c):  
"prediction from a rank-deficient fit may be misleading"
```

[illegible]

```
t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.  
c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.  
t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.M, type = "response", newdata = pred.data.  
c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.  
t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.  
c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.  
t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.  
c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.  
t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.  
c):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.  
t):  
"prediction from a rank-deficient fit may be misleading"  
Warning message in predict.lm(new.fit.Y, type = "response", newdata = pred.data.  
c):  
"prediction from a rank-deficient fit may be misleading"
```

Causal Mediation Analysis

Nonparametric Bootstrap Confidence Intervals with the Percentile Method

	Estimate	95% CI Lower	95% CI Upper	p-value
ACME	0.00563	-0.02165	0.04	0.56
ADE	0.00427	-0.01685	0.04	0.42
Total Effect	0.00990	-0.03226	0.09	0.68
Prop. Mediated	0.56838	-2.38000	1.00	NA

Sample Size Used: 34

Simulations: 500

```
In [10]: results1 <- mediate(model.M, model.Y, treat='BFI..E.', mediator='PAQ',
                             boot=TRUE, sims=500, test = "Sobel")
summary(results1)
```

Running nonparametric bootstrap

Error in statistic(data, original, ...): unused argument (test = "Sobel")
Traceback:

```
1. mediate(model.M, model.Y, treat = "BFI..E.", mediator = "PAQ",
.       boot = TRUE, sims = 500, test = "Sobel")
2. boot::boot(data = y.data, statistic = med.fun, R = sims, sim = "ordinary",
.       m.data = m.data, ...)
```

```
In [11]: # Independent variable
independent_var <- data$BFI..E. [[1]]

# Dependent variable
dependent_var <- data$Panas.. [[1]]

# Mediator variable
mediator_var <- data$PAQ [[1]]

# Load the mediation package
library(mediation)

# Create the formula for the mediation analysis
med_formula <- formula(dependent_var ~ independent_var + mediator_var)

# Perform the mediation analysis with Sobel's test
med_result <- mediate(med_formula, data)

# View the mediation results
summary(med_result)
```

Error in mediate(med_formula, data): number of observations do not match between m
ediator and outcome models
Traceback:

```
1. mediate(med_formula, data)
2. stop("number of observations do not match between mediator and outcome models")
```

```
In [12]: # Load the mediation package
library(mediation)

# Select the first observation from each variable
independent_var <- data$BFI..E. [[1]]
dependent_var <- data$Panas.. [[1]]
mediator_var <- data$PAQ [[1]]

# Combine the variables into a data frame
mediation_data <- data.frame(independent_var, dependent_var, mediator_var)

# Create the formula for the mediation analysis
med_formula <- formula(dependent_var ~ independent_var + mediator_var)

# Perform the mediation analysis with Sobel's test
med_result <- mediate(med_formula, mediation_data)
```

```
# View the mediation results
summary(med_result)
```

```
Error in `[.data.frame`(m.data, , treat): undefined columns selected
Traceback:
```

```
1. mediate(med_formula, mediation_data)
2. m.data[, treat]
3. `[.data.frame`(m.data, , treat)
4. stop("undefined columns selected")
```

```
In [16]: install.packages("multilevel")
library(multilevel)

result <- sobel(pred=data$BFI..E., med=data$PAQ, out=data$Panas..)
print(result)
```

```
Updating HTML index of packages in '.Library'
```

```
Making 'packages.html' ...
done
```

```
Loading required package: nlme
```

```
$`Mod1: Y~X`
              Estimate Std. Error    t value    Pr(>|t|)
(Intercept) 2.00990099  0.2013184  9.98369040 2.359462e-11
pred         0.00990099  0.1010313  0.09799919 9.225442e-01
```

```
$`Mod2: Y~X+M`
              Estimate Std. Error    t value    Pr(>|t|)
(Intercept) 1.977207977  0.2113013  9.35729041 1.528108e-10
pred         0.004273504  0.1025785  0.04166083 9.670363e-01
med          0.027065527  0.0472017  0.57340148 5.705096e-01
```

```
$`Mod3: M~X`
              Estimate Std. Error    t value    Pr(>|t|)
(Intercept) 1.2079208   0.7619984  1.5852012 0.1227546
pred         0.2079208   0.3824077  0.5437149 0.5904059
```

```
$Indirect.Effect
[1] 0.005627486
```

```
$SE
[1] 0.01426333
```

```
$z.value
[1] 0.3945421
```

```
$N
[1] 34
```

```
In [23]: #install bda package if not already installed
install.packages('bda')
```

```
#Load bda package
library(bda)
```

```
result <- mediation.test(data$PAQ,data$BFI..E.,data$Panas..)
print(result)
```

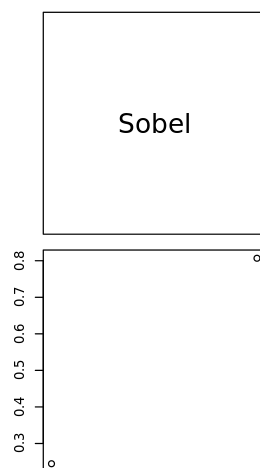
Updating HTML index of packages in '.Library'

Making 'packages.html' ...
done

	Sobel	Aroian	Goodman
z.value	0.3945421	0.2446141	NA
p.value	0.6931809	0.8067552	NA
[1]	0.3945421	0.6931809	

Warning message in min(x):
"no non-missing arguments to min; returning Inf"
Warning message in max(x):
"no non-missing arguments to max; returning -Inf"
Error in plot.window(...): need finite 'ylim' values
Traceback:

1. plot(result)
2. plot.data.frame(result)
3. pairs(data.matrix(x), ...)
4. pairs.default(data.matrix(x), ...)
5. localPlot(x[, j], x[, i], xlab = "", ylab = "", axes = FALSE,
 . type = "n", ..., log = 1)
6. plot(...)
7. plot.default(...)
8. localWindow(xlim, ylim, log, asp, ...)
9. plot.window(...)



```
In [28]: # Load the gvlma package
install.packages('gvlma')
library(gvlma)

# Perform the mediation analysis and obtain regression model objects
model1 <- lm(PAQ ~ BFI..E., data = data)
model2 <- lm(Panas.. ~ PAQ, data = data)
model3 <- lm(Panas.. ~ BFI..E. + PAQ, data = data)

# Open a text file to save the output
sink("gvlma_results.txt")

# Test assumptions for each model using gvlma and print the results
print("Model 1 - PAQ regressed on BFI..E.:")
result1 <- gvlma(model1)
print(result1)
plot(result1)

print("\nModel 2 - Panas.. regressed on PAQ:")
result2 <- gvlma(model2)
print(result2)
plot(result2)

print("\nModel 3 - Panas.. regressed on BFI..E. and PAQ:")
result3 <- gvlma(model3)
print(result3)
plot(result3)

# Close the text file
sink()
```

Updating HTML index of packages in '.Library'

Making 'packages.html' ...
done

```
[1] "Model 1 - PAQ regressed on BFI..E.:"
```

Call:

```
lm(formula = PAQ ~ BFI..E., data = data)
```

Coefficients:

(Intercept)	BFI..E.
1.2079	0.2079

ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance = 0.05

Call:

```
gvlma(x = model1)
```

	Value	p-value	Decision
Global Stat	5.3912	0.24946	Assumptions acceptable.
Skewness	1.6147	0.20384	Assumptions acceptable.
Kurtosis	0.5316	0.46594	Assumptions acceptable.
Link Function	2.7560	0.09689	Assumptions acceptable.
Heteroscedasticity	0.4889	0.48441	Assumptions acceptable.

```
[1] "\nModel 2 - Panas.. regressed on PAQ:"
```

Call:

```
lm(formula = Panas.. ~ PAQ, data = data)
```

Coefficients:

(Intercept)	PAQ
1.98532	0.02725

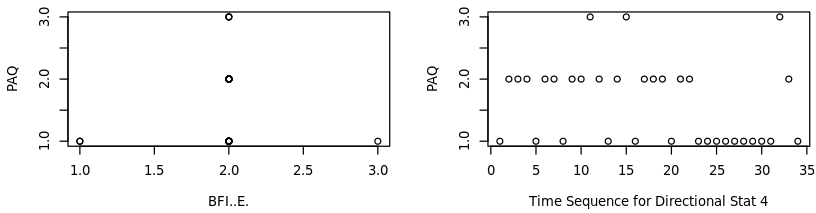
ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance = 0.05

Call:

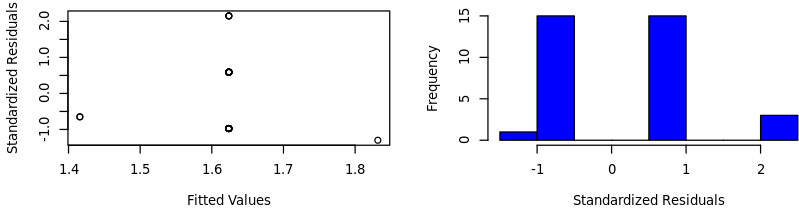
```
gvlma(x = model2)
```

	Value	p-value	Decision
Global Stat	1309.6453	0.0000	Assumptions NOT satisfied!
Skewness	169.8470	0.0000	Assumptions NOT satisfied!
Kurtosis	1138.5189	0.0000	Assumptions NOT satisfied!
Link Function	0.9502	0.3297	Assumptions acceptable.
Heteroscedasticity	0.3292	0.5661	Assumptions acceptable.

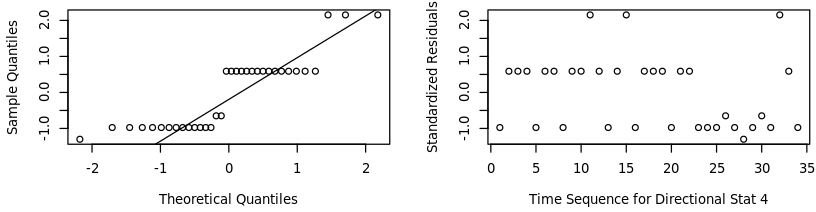
Plot of Response Variable versus Predictor Vari Plot of Response Variable versus Time Sequer



of the Standardized Residuals versus the Fitte Histogram of the Standardized Residuals



Probability Plot of the Standardized Residuals: of the Standardized Residuals versus Time Se



```
[1] "\nModel 3 - Panas.. regressed on BFI..E. and PAQ:"
```

```
Call:
lm(formula = Panas.. ~ BFI..E. + PAQ, data = data)
```

Coefficients:

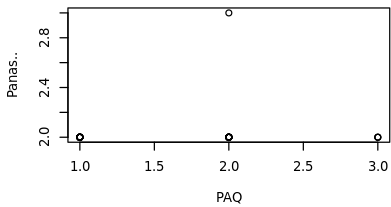
(Intercept)	BFI..E.	PAQ
1.977208	0.004274	0.027066

ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance = 0.05

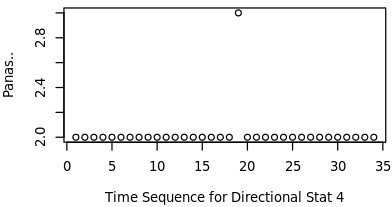
```
Call:
gvlma(x = model3)
```

	Value	p-value	Decision
Global Stat	1309.3317	0.0000	Assumptions NOT satisfied!
Skewness	169.8184	0.0000	Assumptions NOT satisfied!
Kurtosis	1138.2363	0.0000	Assumptions NOT satisfied!
Link Function	0.9482	0.3302	Assumptions acceptable.
Heteroscedasticity	0.3288	0.5663	Assumptions acceptable.

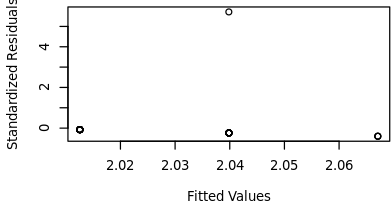
Plot of Response Variable versus Predictor Variable



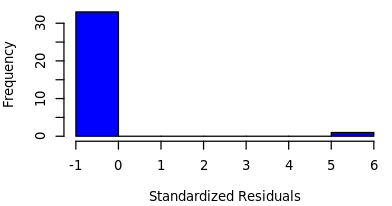
Plot of Response Variable versus Time Sequence



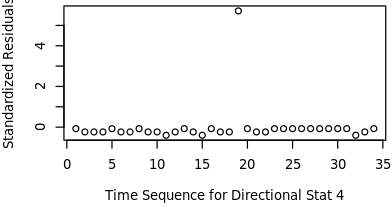
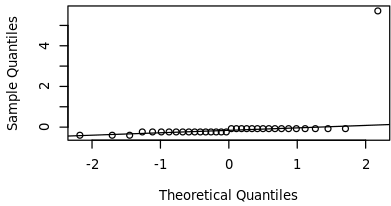
Plot of the Standardized Residuals versus the Fitted Values



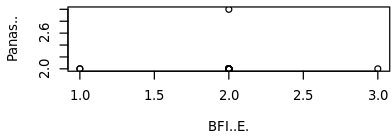
Histogram of the Standardized Residuals



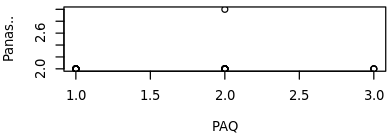
Probability Plot of the Standardized Residuals



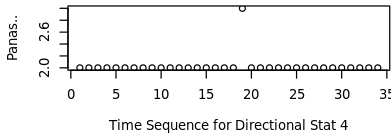
Plot of Response Variable versus Predictor Variable



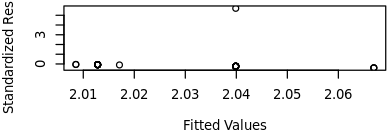
Plot of Response Variable versus Predictor Variable



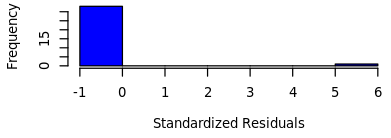
Plot of Response Variable versus Time Sequence



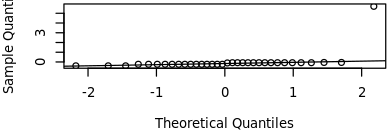
Plot of the Standardized Residuals versus the Fitted Values



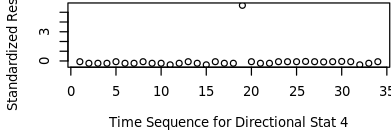
Histogram of the Standardized Residuals



Probability Plot of the Standardized Residuals



Plot of the Standardized Residuals versus Time Sequence



In []:

In [1]:

```
Error in add(4 + 5): could not find function "add"  
Traceback:
```

```
In [2]:
```

```
9
```

```
In [3]:
```

```
Error in divide(40/5): could not find function "divide"  
Traceback:
```

```
In [ ]:
```