Using the R package tlm to fit, visualize and interpret linear, logistic and Poisson regression models with transformed variables

(tlm version 0.2.0)

Jose Barrera-Gómez*[1,2,3] and Xavier Basagaña†[1,2,3]

[1] *ISGlobal*
[2] *Universitat Pompeu Fabra (UPF)*
[3] *CIBER Epidemiología y Salud Pública (CIBERESP)*

January 7, 2025

Contents

*jose.barrera@isglobal.org
†xavier.basagana@isglobal.org

1 Introduction

This document is a user's guide for the R[1] package tlm, which provides the effect of an explanatory variable of interest, X, on a response variable, Y, under a linear, logistic or Poisson regression model with transformations in X and/or Y. In the case of the linear regression model, log and power transformations in any of X and Y are allowed. In the case of logistic and Poisson regression models, log or power transformation in X are allowed. Other explanatory variables can be in the model, in which case adjusted measures for Y and adjusted effects of X on Y are automatically computed. The package also works if there are no transformations. The package provides both numerical and graphical outputs as well as information on interpreting results. The methodology is described in the original work by Barrera-Gómez and Basagaña[1], whose illustrative examples, among others, are reproduced here.

2 Getting started

Start an R session and load the package by typing

```
library(tlm)

##
## This is tlm 0.2.0.  For details, use:
## > help(package = 'tlm') and browseVignettes('tlm')
##
## To cite the methods in the package use:
## > citation('tlm')
```

You can get a brief overview of the package by typing

```
help(package = "tlm")
```

This user's guide can be recovered by typing

```
browseVignettes("tlm")
```

2.1 Fitting the model

The first step is to fit the model in the transformed space (i.e., considering that Y and X are already transformed, if any transformation is assumed), which is performed by the function `tlm`. We can get information on this function using the help:

```
?tlm
```

The main arguments of the function `tlm` are:

- `formula`: the model formula with the usual syntax as in `lm` and `glm`. Left-hand-side indicates the response variable (whose values are assumed to be already transformed). First term in right-hand-side indicates the explanatory variable of interest (whose values are assumed to be already transformed). Right-hand-side can include additional terms (e.g. adjusting variables) but the explanatory variable of interest cannot be involved in any of them.

[1] R is a free and open source software and it is available at http://cran.r-project.org/.

- `family`: the model family. It can be `gaussian`, for linear regression (default); `binomial`, for logistic regression with logit link, o `poisson`, for Poisson regression.
- `data`: a `data.frame` containing the variables in the model.
- `ypow` and `xpow`: the power transformations already done in the response and in the explanatory variable of interest, respectively. The value 1 (default) means no transformation; the value 0 means log transformation.
- `...`: additional arguments for the underlying `lm` or `glm` fitting.

As a result of a call to the function `tlm`, we obtain an object of class `"tlm"` which can be passed to the following functions or methods:

- `summary` to summarize the fitted model (section 2.2)
- `plot` to visualize the fitted model (section 2.3)
- `MY` to obtain adjusted measures (section 2.4)
- `effect` to obtain adjusted effects (section 2.5)

2.2 Summary of the fitted model

The method `summary` provides information on the fitted model. It essentially provides the summary of the underlying `lm` or `glm` fitted in the transformed space.

2.3 Visualization of the fitted model

The specific method `plot` provides, for an object of class `"tlm"` (output of the `tlm` function), a graphical representation of the fitted model. There are three types of visualization, depending on the value of the argument `type` passed to `plot`:

- `type = "original"` (default): the fitted model is plotted in the original space of the variables. If a visualization of the fitted model is reported, this is the proper option.
- `type = "transformed"`: the fitted model is plotted in the transformed space of the variables (where the model has been fitted). Plots obtained for this option are intended to visually explore the model goodness of fit and should not be reported because values of the transformed variables are meaningless (e.g. the logarithm of cotinine levels (ng/ml) has no sense).
- `type = "diagnosis"`: a fitted model diagnostics plot is shown as with `plot.lm` or `plot.glm`.

Other arguments of the specific method `plot` are:

- `observed`: logical indicating whether the observations are shown in the plot. Default is `FALSE`.
- `xname`, `yname`: character indicating the name of the explanatory and the response variable of interest for labeling the plot axes. Default are `"x"` and `"y"`, respectively.
- `level`: numeric indicating the confidence level for the confidence of the expectation of the response variable according to the fitted model. Default is 0.95.

The specific method `plot` automatically labels the Y axis with the appropriate name of the measure (i.e. mean, geometric mean, etc).

2.4 Adjusted measures

Once the model has been fitted with the function tlm, the resulting object can be passed to the function MY, which provides measures of the response variable for given values of the explanatory variable. If the model contains other explanatory variables, then adjusted measures are automatically computed. These adjusted measures are obtained by setting the remaining explanatory variables in the model at their means. We can get information on this function using the help:

```
?MY
```

The main arguments of the function MY are:

- object: an object of class "tlm", the result of a call to the function tlm.
- x: the value(s) of the explanatory variable of interest for which the expected measure of the response variable should be computed.
- npoints: if x is not provided, the number of points where the measure should be computed. Default is 10.
- space: the space of the variables in which measures should be computed. It can be "original" (default) or "transformed".
- level: the confidence level for measures. Default is 0.95.

The function MY automatically provides the unit of the measure (mean, geometric mean, median, probability or logodds, depending on the case).

2.5 Adjusted effects

The fitted model can be passed to the function effect in order to obtain the effect of X on Y in the original space of the variables, adjusted for other potential adjusting variables passed in formula to the tlm function. We can get information on this function using the help:

```
?effect
```

The main arguments of the function effect are:

- object: an object of class "tlm", the result of a call to the function tlm.
- x1: the value(s) of the explanatory variable where the effect should be computed.
- x2: the alternative value(s) of the explanatory variable (changing from x1) for which the effect should be computed.
- c: the additive change in the explanatory variable.
- q: the multiplicative change in the explanatory variable.
- r: the percent change in the explanatory variable.
- npoints: the number of points where the effect should be computed.

- level: the confidence level for the effect.

In order to compute effects, the change in the explanatory variable should be characterized. It can be done in several ways. For instance, providing: (1) x1 and x2; (2) x1 and one of c, q or r; (3) x1, npoints and one of c, q or r. Only one of the arguments c, q or r is used, prevailing c and then q. If no enough arguments are passed, the interquartile range will be considered and a summary effect should be computed, if it exists. If the explanatory variable is categorical, then the effect is computing for a change between the reference level and each of the remaining levels of the explanatory variable. Confidence intervals are computed by transforming the endpoints of the intervals in the transformed scale when it is possible, while non-parametric bootstrap is used otherwise. The function effect automatically provides the unit of the effect measure (mean, geometric mean, median or odds ratio, depending on the case).

In addition, the function effectInfo provides further information on how to interpret the effect (use ?effectInfo for further details).

3 Illustrative examples

3.1 Linear regression model

3.1.1 Log transformation in the response

Consider the evaluation of the association between the intima media thickness of the carotid artery (IMT), measured in mm, Y, and age, in years, X. Variable Y was log transformed to achieve normality. The imt data were simulated to emulate true data pattern observed in a real study[2].

First, we can load data (further information about data is available with the help function, ?imt) and see its first rows and a summary:

```
data(imt)
dim(imt)

## [1] 2784    3

head(imt)

##   age       imt      logimt
## 1  53 0.6588690 -0.41723048
## 2  67 0.9215614 -0.08168583
## 3  57 0.9539836 -0.04710877
## 4  55 0.7226028 -0.32489555
## 5  73 0.7406030 -0.30029063
## 6  57 0.7638349 -0.26940367

summary(imt)

##       age             imt              logimt
##  Min.   :32.00   Min.   :0.3755   Min.   :-0.9794
##  1st Qu.:51.00   1st Qu.:0.6307   1st Qu.:-0.4609
##  Median :59.00   Median :0.7257   Median :-0.3206
##  Mean   :58.48   Mean   :0.7420   Mean   :-0.3196
##  3rd Qu.:66.00   3rd Qu.:0.8324   3rd Qu.:-0.1834
##  Max.   :86.00   Max.   :1.5788   Max.   : 0.4567
```

Suppose that we are interested in the effect of age on IMT, under a linear regression model with log transformation in the response variable, IMT. The model can be fitted as follows:

```
modimt <- tlm(logimt ~ age, data = imt, ypow = 0)
```

where `ypow = 0` indicates that the response variable is already log transformed. Note that we have not set the family since default family is Gaussian. The fitted model results in:

```
modimt

##
## Linear regression fitted model in the transformed space
## -------------------------------------------------------
##
## Transformations:
##    In the response variable: log
##
## Call:
## lm(formula = logimt ~ age, data = imt)
##
## Coefficients:
## (Intercept)          age
##   -0.877692     0.009543
```

Further information on the fitted model is available using the method **`summary`**:

```
summary(modimt)

##
## Linear regression fitted model in the transformed space
## -------------------------------------------------------
##
## Transformations:
##    In the response variable: log
##
## Call:
## lm(formula = logimt ~ age, data = imt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.57891 -0.11792  0.00142  0.11879  0.61968
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.8776919  0.0183979  -47.71   <2e-16 ***
## age          0.0095433  0.0003093   30.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1778 on 2782 degrees of freedom
## Multiple R-squared:  0.255,Adjusted R-squared:  0.2547
## F-statistic: 952.2 on 1 and 2782 DF,  p-value: < 2.2e-16
```

A numerical representation of the relationship between age and IMT is available using the function MY. As default, the measure of the response variable is computed in 10 points along the range of the explanatory variable:

```
MY(modimt)

##
## Estimated adjusted geometric mean of the response variable in the original space:
##
##     X geomMean(Y)  lower95%  upper95%
## 1  32   0.5642233 0.5545100 0.5741068
## 2  38   0.5974736 0.5891267 0.6059388
## 3  44   0.6326834 0.6257680 0.6396752
## 4  50   0.6699681 0.6643811 0.6756020
## 5  56   0.7094501 0.7046577 0.7142750
## 6  62   0.7512587 0.7460592 0.7564945
## 7  68   0.7955312 0.7885807 0.8025430
## 8  74   0.8424128 0.8327802 0.8521567
## 9  80   0.8920571 0.8791028 0.9052023
## 10 86   0.9446270 0.9278224 0.9617360
```

The number of points can be set using **npoints**:

```
MY(modimt, npoints = 3)

##
## Estimated adjusted geometric mean of the response variable in the original space:
##
##    X geomMean(Y)  lower95%  upper95%
## 1 32   0.5642233 0.5545100 0.5741068
## 2 59   0.7300552 0.7252407 0.7349016
## 3 86   0.9446270 0.9278224 0.9617360
```

We can also choose a specific set of values of the explanatory variable for which the measure of the response variable should be computed. For instance, the first and third quartile:

```
q13 <- quantile(imt$age, probs = c(1, 3)/4)
MY(modimt, x = q13)

##
## Estimated adjusted geometric mean of the response variable in the original space:
##
##    X geomMean(Y)  lower95%  upper95%
## 1 51   0.6763924 0.6709922 0.6818362
## 2 66   0.7804912 0.7742493 0.7867834
```

Measures can also be computed in the transformed space:

```
MY(modimt, x = q13, space = "transformed")

##
## Estimated adjusted mean of the response variable in the transformed space:
##
##    X mean(log(Y))   lower95%   upper95%
## 1 51   -0.3909818 -0.3989978 -0.3829659
## 2 66   -0.2478318 -0.2558614 -0.2398023
```

A graphical representation of the relationship between age and IMT is available using the method `plot`. For instance, the two following instructions provide left and right plots in Figure 1, respectively:

```
plot(modimt, type = "transformed", observed = TRUE, xname = "Age (years)", yname = "IMT")
plot(modimt, observed = TRUE, xname = "Age (years)", yname = "IMT (mm)")
```

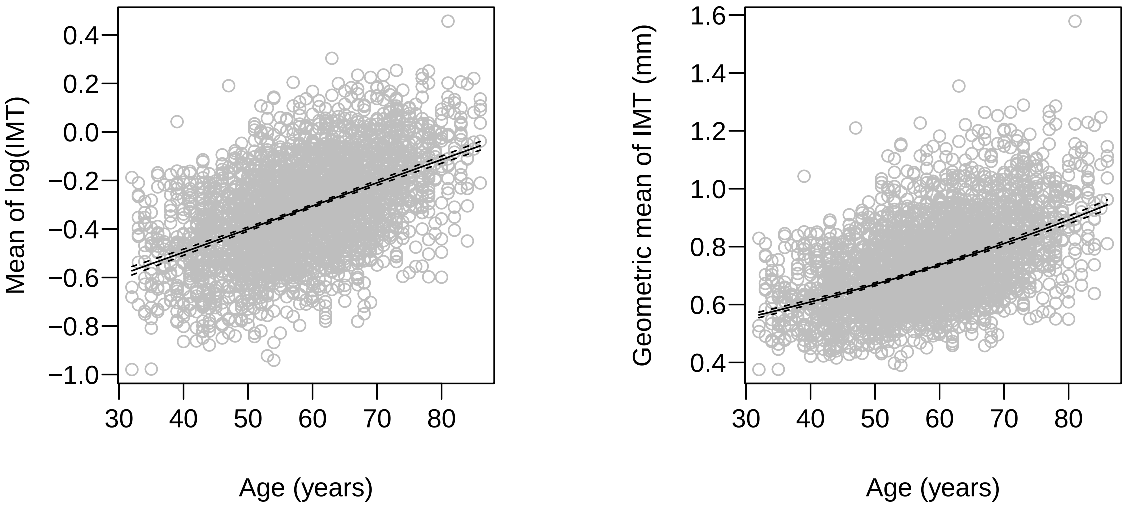

Figure 1: Visualization of the fitted model `modimt`. Left: In the transformed space, the mean of the logarithm of IMT values as a function of age is shown. Note that this plot should not be reported since the logarithm of IMT values are meaningless. This type of plot is intended just to visually explore the model goodness of fit. Right: In the original space, the geometric mean (or equivalently the median) of IMT values as a function of age is shown. This type of plot is appropriate for reporting. Dashed lines represent 95% confidence intervals for the measure.

The argument `observed` controls whether observations are shown in the plot (default is `FALSE`). Further information on the usage of the method `plot` is available using `?tlm`.

Diagnostics plot as in Figure 2 can be obtained with the following instruction:

```
plot(modimt, type = "diagnosis")
```

The function `effectInfo` provides information on interpreting the relationship between age (X) and IMT (Y):

```
effectInfo(modimt)

##
## The effect of X on Y can be summarized with a single number as follows:
##
##  - Change in X: additive of c units
##  - Type of effect on Y: percent change in the geometric mean of Y
##  - Effect size: 100 * [exp(c * beta) - 1]%
##
##    beta coefficient estimate:
##        Estimate   Std. Error t value      Pr(>|t|)
## age 0.009543334 0.0003092751 30.8571 4.531758e-180
##
## Further details can be obtained using effect().
```

Thus, we can see that, in this case, if we use additive changes in X and percent (or multiplicative) changes in the geometric mean of Y, a summary effect can be obtained, which is independent of the value of X for which the effect is computed.

The function `effect` provides as default the expected change in IMT for an additive change in age equal to the interquartile range:

```
effect(modimt)

##
## Computing effects...
##
## Adjusted percent change in the geometric mean of the response variable
## for a 'c' units additive change in the explanatory variable equivalent
## to the interquartile range:
##
##    c Estimate lower95% upper95%
## 1 15 15.39029  14.3454 16.44472
##
## For further information on interpreting the effect use effectInfo().
```

Other measures of effects can be obtained. For instance, we may be interested in a difference of (geometric) means when changing age across the first quartile, the median and the third quartile:

```
q123 <- quantile(imt$age, probs = 1:3/4)    # quartiles
effect(modimt, x1 = q123)

##
## Computing effects...
##
```

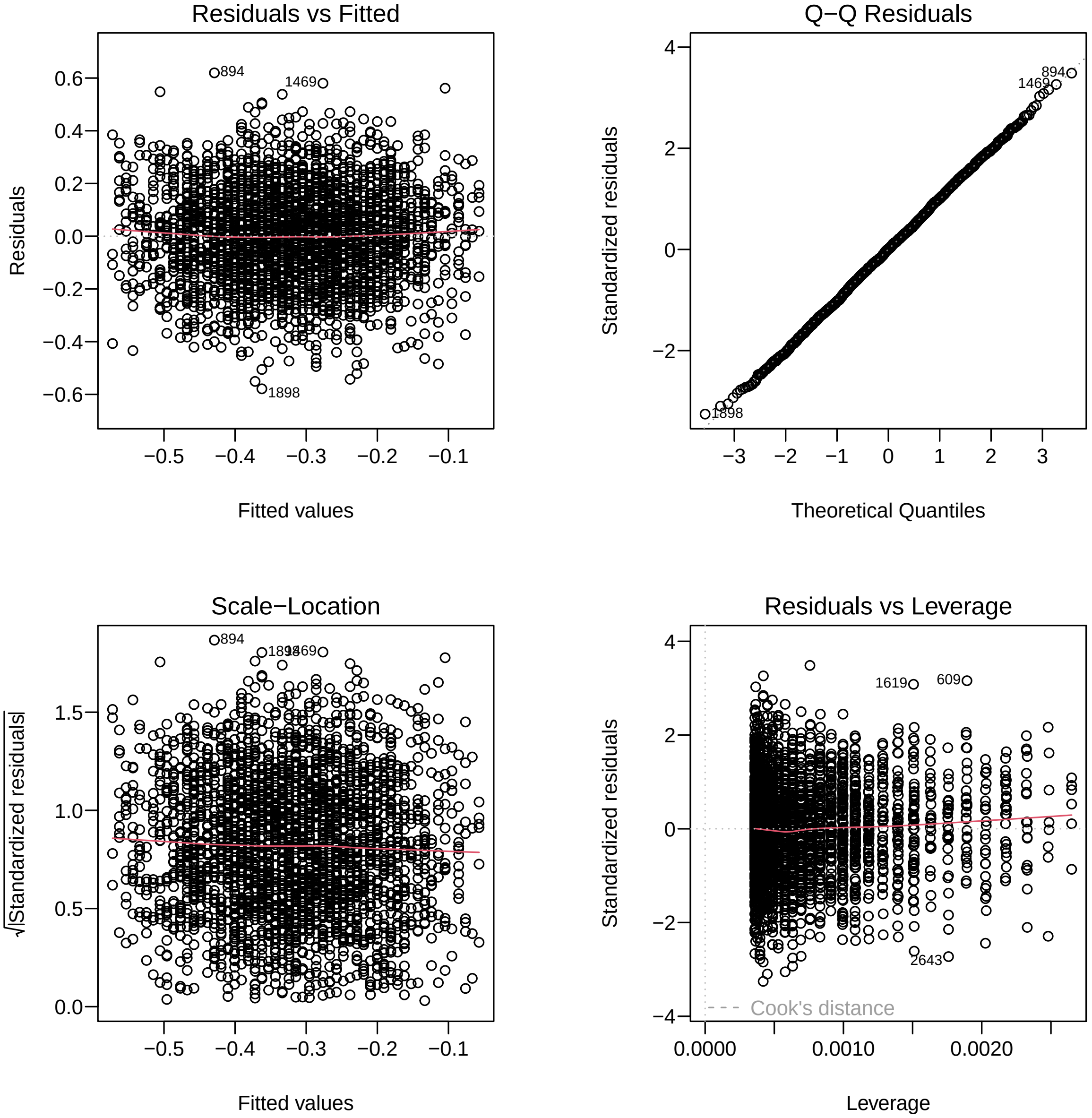

Figure 2: Diagnostics plot for the model `modimt`.

```
## Adjusted change in the geometric mean of the response variable when the
## explanatory variable changes from x1 to x2 (confidence interval for the
## difference change based on 999 bootstrap samples):
##
##     x1 x2 EstimateDiff   lower95%   upper95% EstimatePercent lower95% upper95%
## 25% 51 59   0.05366274 0.05010240 0.05714970        7.933669 7.411302 8.458576
## 50% 59 66   0.05043602 0.04707039 0.05362892        6.908521 6.455654 7.363315
```

```
##
## For further information on interpreting the effect use effectInfo().
```

As in this example, when a summary effect is not computed, then both difference and percent changes in the response are computed. The number of bootstrap samples is controlled by the argument `nboot` whose default value is 999[3].

3.1.2 Log transformation in the explanatory variable

Consider now the evaluation of the association between the birth weight, in grams, Y, and the cord serum cotinine level in the mother, in ng/ml, X. Variable X was log transformed to achieve a more homogeneous distribution. The `cotinine` data were simulated to emulate true data patterns observed in a real study[4]. Data also contains a binary variable indicating whether the birth weight was low (defined as lower than 2500 g).

We can load and explore data (further information about data is available with the help function, `?cotinine`):

```
data(cotinine)
dim(cotinine)

## [1] 351   4

head(cotinine)

##     cotinine logcotinine weight underweight
## 2  5.1584035    1.640627   3626          no
## 5  0.2909473   -1.234613   3672          no
## 11 4.1119142    1.413889   3779          no
## 12 3.0037959    1.099877   3540          no
## 14 5.9240779    1.779025   3179          no
## 17 7.3854370    1.999510   2494         yes

summary(cotinine)

##     cotinine        logcotinine         weight     underweight
##  Min.   :  0.200   Min.   :-1.609   Min.   :1656   no :327
##  1st Qu.:  3.234   1st Qu.: 1.174   1st Qu.:2920   yes: 24
##  Median :  7.385   Median : 2.000   Median :3232
##  Mean   : 39.431   Mean   : 2.353   Mean   :3218
##  3rd Qu.: 39.226   3rd Qu.: 3.669   3rd Qu.:3522
##  Max.   :910.000   Max.   : 6.813   Max.   :4771
```

Suppose that we are interested in the effect of cotinine level on birth weight, under a linear regression model with log transformation in the explanatory variable, cotinine. The model can be fitted as follows:

```
modcot <- tlm(weight ~ logcotinine, data = cotinine, xpow = 0)
```

where `xpow = 0` indicates that the explanatory variable is already log transformed. The fitted model provides the following results:

```r
summary(modcot)

##
## Linear regression fitted model in the transformed space
## -----------------------------------------------------
##
## Transformations:
##    In the explanatory variable: log
##
## Call:
## lm(formula = weight ~ logcotinine, data = cotinine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1390.64  -280.11    -2.95   300.47  1422.31
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3406.49      42.59  79.978  < 2e-16 ***
## logcotinine   -80.00      14.95  -5.351 1.58e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 450.1 on 349 degrees of freedom
## Multiple R-squared:  0.07583,Adjusted R-squared:  0.07318
## F-statistic: 28.64 on 1 and 349 DF,  p-value: 1.585e-07
```

In Figure 3 (obtained with the following instructions) we can see the fitted model (left) and the relationship between cotinine level and birth weight, under the model, in the original space of the variables (right).

```r
plot(modcot, type = "transformed", observed = TRUE, xname = "Cotinine", yname = "weight (kg)")
plot(modcot, xname = "Cotinine (ng/ml)", yname = "weight (kg)")
```

The function `effectInfo` provides information on interpreting the relationship between cotinine levels (X) and weight (Y):

```r
effectInfo(modcot)

##
## The effect of X on Y can be summarized with a single number as follows:
##
##  - Change in X: multiplicative of factor q (equivalently, adding an r = 100 * (q - 1)% to X)
##  - Type of effect on Y: additive change in the mean of Y
##  - Effect size: beta * log(q) units of Y
##
##    beta coefficient estimate:
##              Estimate Std. Error   t value     Pr(>|t|)
## logcotinine -80.00108   14.94986 -5.351292 1.584903e-07
##
## Further details can be obtained using effect().
```

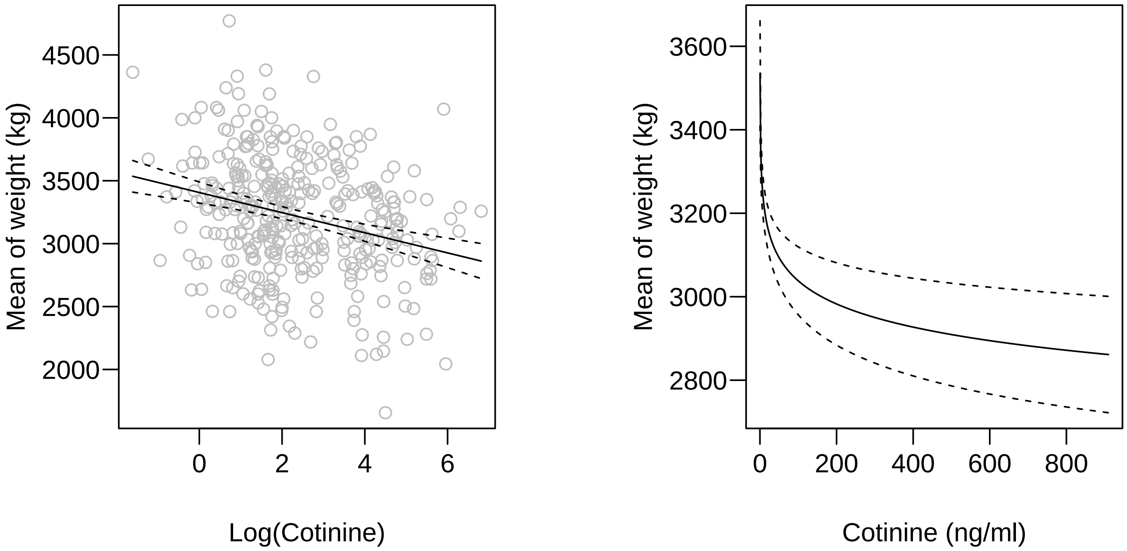

Figure 3: Visualization of the fitted model `modcot`. Left: In the transformed space, mean weight (kg) as a function of the logarithm of cotinine levels is shown. Note that this plot should not be reported since the logarithm of cotinine levels are meaningless. This type of plot is intended just to visually explore the model goodness of fit. Right: In the original space, the mean weight (kg) as a function of cotinine levels is shown. This type of plot is appropriate for reporting. Dashed lines represent 95% confidence intervals for the measure.

In this case, if we use multiplicative (or percent) changes in X and additive changes in the mean of Y, a summary effect can be obtained, which is independent of the value of X for which the effect is computed. The function `effect` provides as default the expected change in weight for a percent change in cotinine levels equal to the interquartile ratio:

```
effect(modcot)

##
## Computing effects...
##
## Adjusted additive change in the mean of the response variable for an
## 'r'% change in the explanatory variable equivalent to the interquartile
## ratio:
##
##          r  Estimate  lower95%  upper95%
## 1 1112.878 -199.6492 -273.0272 -126.2712
##
## For further information on interpreting the effect use effectInfo().
```

Alternatively, by exploring Figure 3, we can see that several 10-fold changes occur in the population and choose the more common number $q = 10$, in which case the effect is

```
effect(modcot, q = 10)

##
## Computing effects...
##
## Adjusted additive change in the mean of the response variable for an
## 'r' = 900% change in the explanatory variable:
##
##     r  Estimate  lower95% upper95%
## 1 900 -184.2093 -251.9126 -116.506
##
## For further information on interpreting the effect use effectInfo().
```

If we are interested in explore the effect of an additive change in cotinine levels, we can obtain, for example, effects for additive changes in X along its range. For instance:

```
range(cotinine$cotinine)

## [1]   0.2 910.0

effect(modcot, x1 = 100, c = 200, npoints = 4)

##
## Computing effects...
##
## Adjusted change in the mean of the response variable when the explanatory
## variable changes from x1 to x2 (confidence interval for the percent change
## based on 999 bootstrap samples):
##
##    x1  x2 EstimateDiff   lower95%  upper95% EstimatePercent   lower95%
## 1 100 300    -87.89017 -120.19284 -55.58750      -2.8929637 -4.0411918
## 2 300 500    -40.86660  -55.88649 -25.84671      -1.3852256 -1.9996276
## 3 500 700    -26.91814  -36.81149 -17.02479      -0.9252415 -1.3195109
## 4 700 900    -20.10543  -27.49486 -12.71599      -0.6975258 -0.9923362
##     upper95%
## 1 -1.7854058
## 2 -0.8817584
## 3 -0.5283016
## 4 -0.4061337
##
## For further information on interpreting the effect use effectInfo().
```

3.1.3 Log transformation in both the response and the explanatory variable

Consider an epidemiological study to assess the association between cat allergen levels (*Fel d 1*) in the bed mattress, X, and in the living room, Y, in homes of study participants, taking into account cat ownership, C. Both variables X and Y were log transformed to achieve linearity in their relationship. The `feld1` data were simulated to emulate true data patterns observed in a real study[5].

We can load and explore data (further information about data is available with the help function, `?feld1`):

```
data(feld1)
dim(feld1)

## [1] 471   5

head(feld1)

##     mattress        room logmattress    logroom cat
## 1 0.66504894  0.26758221  -0.4078946 -1.3183284 yes
## 2 0.02723504  0.16796992  -3.6032509 -1.7839704  no
## 3 0.16773827  0.76728932  -1.7853504 -0.2648913  no
## 4 0.01391101  0.05480368  -4.2750750 -2.9039979  no
## 5 0.04216982  0.11931900  -3.1660504 -2.1259547 yes
## 6 1.44212520 17.29388484   0.3661179  2.8503530 yes

summary(feld1)

##     mattress             room             logmattress         logroom
##  Min.    :  0.0030   Min.    :   0.0017   Min.    :-5.7975   Min.    :-6.3565
##  1st Qu.:  0.0636   1st Qu.:   0.1076   1st Qu.:-2.7559   1st Qu.:-2.2295
##  Median :  0.1773   Median :   0.3081   Median :-1.7297   Median :-1.1774
##  Mean   :  3.1976   Mean   :  27.5263   Mean   :-1.5942   Mean   :-0.8528
##  3rd Qu.:  0.5565   3rd Qu.:   1.1713   3rd Qu.:-0.5861   3rd Qu.: 0.1581
##  Max.   :467.7814   Max.   :2120.4743   Max.   : 6.1480   Max.   : 7.6594
##   cat
##  no :403
##  yes: 68
##
##
##
##
```

Suppose that we are interested in the association between allergen levels in the mattress and in the living room, under a linear regression model with log transformation in both variables, adjusting for cat ownership. The model can be fitted as follows:

```
modcat <- tlm(logroom ~ logmattress + cat, data = feld1, ypow = 0, xpow = 0)
```

where `ypow = 0` and `xpow = 0` indicate that both the explanatory and the response variables are already log transformed. The fitted model provides the following results:

```
summary(modcat)

##
## Linear regression fitted model in the transformed space
## -------------------------------------------------------
##
## Transformations:
##    In the response variable: log
##    In the explanatory variable: log
##
## Call:
## lm(formula = logroom ˜ logmattress + cat, data = feld1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9172 -1.1228 -0.0631  0.9464  6.0440
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.05412    0.11651  -0.465    0.642
## logmattress  0.63936    0.04454  14.354  < 2e-16 ***
## catyes       1.52747    0.23098   6.613 1.03e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.68 on 468 degrees of freedom
## Multiple R-squared:  0.4192,Adjusted R-squared:  0.4167
## F-statistic: 168.9 on 2 and 468 DF,  p-value: < 2.2e-16
```

We can create Figure 4 with

```
plot(modcat, type = "transformed", observed = TRUE, xname = "Mattress levels",
     yname = "living room levels")
plot(modcat, xname = "Mattress levels", yname = "living room levels")
```

which provides a graphical representation of the relationship between allergen levels in the mattress and in the living room. Measures in Figure 4 have been obtained after averaging the expected measure of Y over all subjects in the data set. This is equivalent to saying that measures and effects are calculated for an average individual in the population and they can be interpreted as adjusted measures and adjusted effects[6] .

The function `effectInfo` provides information on interpreting the relationship between allergen levels in mattress (X) and in living room (Y):

```
effectInfo(modcat)

##
## The effect of X on Y can be summarized with a single number as follows:
##
##  - Change in X: multiplicative of factor q (equivalently, adding an r = 100 * (q - 1)% to X)
##  - Type of effect on Y: percent change in the geometric mean of Y
##  - Effect size: 100 * (qˆbeta - 1)%
##
##    beta coefficient estimate:
##             Estimate Std. Error  t value     Pr(>|t|)
```

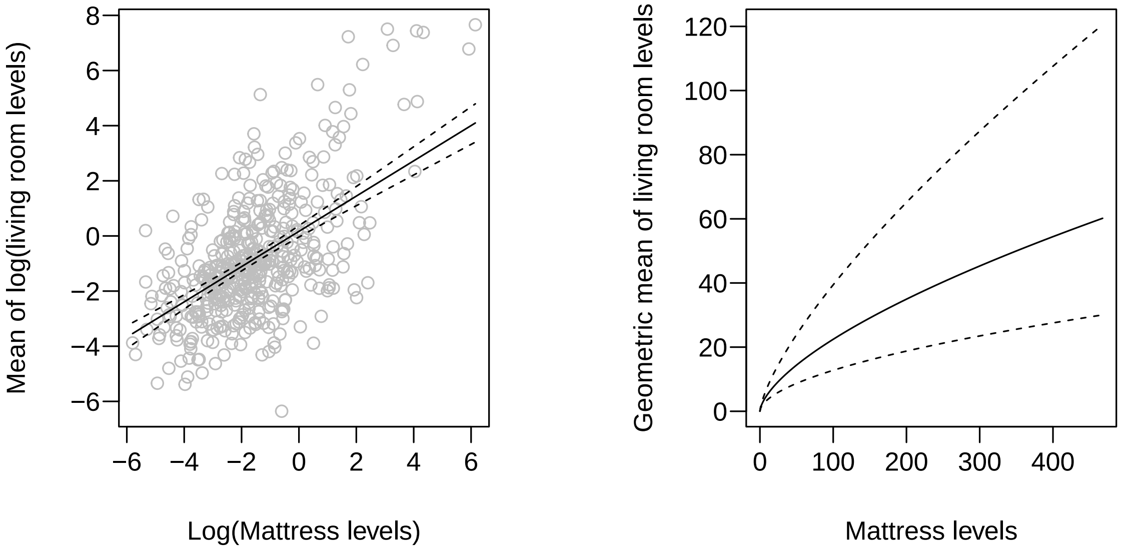

Figure 4: Visualization of the fitted model `modcat`. Left: In the transformed space, the logarithm of mean *Fel d 1* levels in room as a function of the logarithm of *Fel d 1* levels in mattress is shown. Note that this plot should not be reported since the logarithm of *Fel d 1* levels are meaningless. This type of plot is intended just to visually explore the model goodness of fit. Right: In the original space, the geometric mean (or equivalently median) of *Fel d 1* levels in room as a function of *Fel d 1* levels in mattress is shown. This type of plot is appropriate for reporting. Dashed lines represent 95% confidence intervals for the measure.

```
## logmattress 0.6393565 0.04454319 14.35363 5.610118e-39
##
## Further details can be obtained using effect().
```

In this case, if we use multiplicative (or percent) changes in both variables, a summary effect can be obtained, which is independent of the value of the explanatory variable for which the effect is computed.

The function `effect` provides as default the expected change in allergen levels in the living room for a percent change in allergen levels in mattress equal to the interquartile ratio:

```
effect(modcat)

##
## Computing effects...
##
```

```
## Adjusted percent change in the geometric mean of the response variable
## for an 'r'% change in the explanatory variable equivalent to the
## interquartile ratio:
##
##          r Estimate lower95% upper95%
## 1 775.6541  300.393 231.1347  384.137
##
## For further information on interpreting the effect use effectInfo().
```

If we are interested in the effect of cat ownership on allergen levels in the living room, with log transformation in this, we should run a new model with cat ownership as the explanatory variable:

```
modcat2 <- tlm(logroom ~ cat, data = feld1, ypow = 0)
modcat2

##
## Linear regression fitted model in the transformed space
## -------------------------------------------------------
##
## Transformations:
##    In the response variable: log
##
## Call:
## lm(formula = logroom ~ cat, data = feld1)
##
## Coefficients:
## (Intercept)       catyes
##      -1.218        2.528
```

Then, we can computed measures:

```
MY(modcat2)

##
## Estimated adjusted geometric mean of the response variable in the original space:
##
##   xlevel geomMean(Y)  lower95% upper95%
## 1     no    0.295888 0.2429528 0.360357
## 2    yes    3.706041 2.2935543 5.988407
```

and the effect:

```
effect(modcat2)

##
## Computing effects...
##
## Adjusted change in the geometric mean of the response variable when
## the explanatory variable changes from its reference level, 'no', to
## an alternative level (confidence interval for the difference based
## on 999 bootstrap samples):
##
##             EstimateDiff lower95% upper95% EstimatePercent lower95% upper95%
## no -> yes     3.410153 1.571001  7.22778        1152.515 645.5631 2004.171
##
## For further information on interpreting the effect use effectInfo().
```

A graphical representation of the relationship can be obtained (see Figure 5):

```
plot(modcat2, yname = "room levels")
```

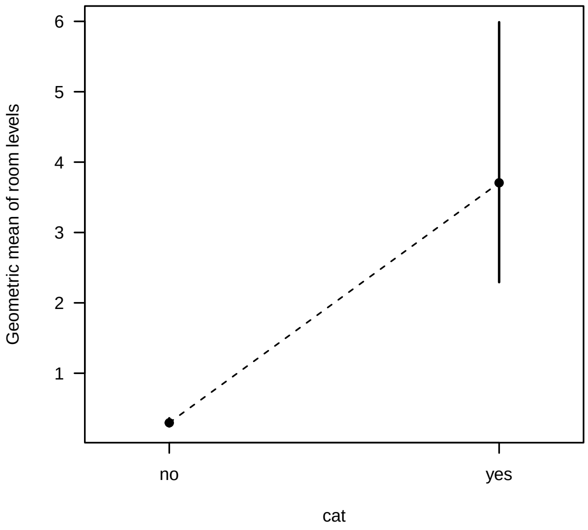

Figure 5: Geometric mean (and 95% confidence intervals) of allergen levels in the living room as a function of cat ownership.

3.1.4 Power transformations

Consider now the modeling of the association between triglycerides, X, and glucose, Y, levels in blood, both measured in mg/dl. Variables X and Y were transformed under power functions $g(X) = 1/\sqrt{X} = X^{-1/2}$ and $f(Y) = 1/Y^2 = Y^{-2}$, respectively, to achieve linearity. The `glucose` data were simulated to emulate true data pattern observed in a real study[2].

First, we can load and explore data (further information about data is available with the help function, `?glucose`):

```
data(glucose)
dim(glucose)

## [1] 400   4
```

```
head(glucose)

##   trigly gluco   inv12tri      inv2glu
## 1    264   116 0.06154575 7.431629e-05
## 2    151   123 0.08137885 6.609822e-05
## 3     67    96 0.12216944 1.085069e-04
## 4     73    86 0.11704115 1.352082e-04
## 5    180   104 0.07453560 9.245562e-05
## 6    130   114 0.08770580 7.694675e-05

summary(glucose)

##      trigly           gluco           inv12tri          inv2glu
##  Min.   : 33.00   Min.   : 63.00   Min.   :0.03875   Min.   :1.457e-05
##  1st Qu.: 70.75   1st Qu.: 85.00   1st Qu.:0.08805   1st Qu.:8.573e-05
##  Median : 94.50   Median : 95.00   Median :0.10287   Median :1.108e-04
##  Mean   :112.38   Mean   : 99.13   Mean   :0.10297   Mean   :1.131e-04
##  3rd Qu.:129.00   3rd Qu.:108.00   3rd Qu.:0.11889   3rd Qu.:1.384e-04
##  Max.   :666.00   Max.   :262.00   Max.   :0.17408   Max.   :2.520e-04
```

Then, we can fit and explore the model of interest:

```
modglucose <- tlm(inv2glu ~ inv12tri, data = glucose, ypow = -2, xpow = -1/2)
summary(modglucose)

##
## Linear regression fitted model in the transformed space
## -------------------------------------------------------
##
## Transformations:
##    In the response variable: power, exponent = -2
##    In the explanatory variable: power, exponent = -0.5
##
## Call:
## lm(formula = inv2glu ~ inv12tri, data = glucose)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -9.831e-05 -2.282e-05 -8.110e-07  2.037e-05  1.226e-04
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.424e-05  8.409e-06    6.45 3.25e-10 ***
## inv12tri    5.715e-04  7.982e-05    7.16 3.92e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.557e-05 on 398 degrees of freedom
## Multiple R-squared:  0.1141,Adjusted R-squared:  0.1119
## F-statistic: 51.27 on 1 and 398 DF,  p-value: 3.915e-12
```

The function MY provides a numerical representation of the relationship between triglycerides (X) and glucose (Y), under the fitted model, in the original scale of the variables:

```
MY(modglucose)

##
## Estimated adjusted median of the response variable in the original space:
##
##            X median(Y)  lower95%  upper95%
## 1   33.0000  80.65215  77.74929  83.90648
## 2  103.3333  95.14538  93.64405  96.72131
## 3  173.6667 101.21681  98.47839 104.19717
## 4  244.0000 104.92701 101.08180 109.24731
## 5  314.3333 107.53510 102.82668 112.95582
## 6  384.6667 109.51322 104.11239 115.85237
## 7  455.0000 111.08770 105.11481 118.21011
## 8  525.3333 112.38364 105.92678 120.18625
## 9  595.6667 113.47702 106.60300 121.87907
## 10 666.0000 114.41723 107.17821 123.35386
```

This relationship can also be represented graphically in Figure 6:

```
plot(modglucose, type = "transformed", observed = TRUE, xname = "Triglycerides (mg/dl)",
     yname = "glucose (mg/dl)")
plot(modglucose, xname = "Triglycerides (mg/dl)", yname = "glucose (mg/dl)")
```

The function effectInfo indicates that, under the fitted model, there is no summary effect:

```
effectInfo(modglucose)

##
## The effect of X on Y cannot be summarized with a single number.
## Its behavior can be explored using effect().
```

Indeed, for general transformations it is not possible to find a summary measure that works for all values of X and its change. In such cases, we can create tables with the four possible combinations that result when considering both additive and multiplicative changes in both X and Y. These tables should report the effects for several basal values of X along the observed range[7]. For instance, the 2.5th and 97.5th percentiles of triglycerides level were 47 mg/dl and 313 mg/dl, respectively. Thus, we can report the effects between pairs of consecutive values of $X = 50, 100, 150, 200$ and 250 mg/dl, that is to say, for an additive increase of $c = 50$ mg/dl, and also between pairs of consecutive values: 50, 75, 112.5, 168.8 and 253.1 mg/dl, that is, for an $r = 50\%$ increase:

```
# Effects for an additive change in triglycerides level:
xc <- 50 * (1:5)
xc

## [1]  50 100 150 200 250

effectXdiff <- effect(modglucose, x1 = xc)
```

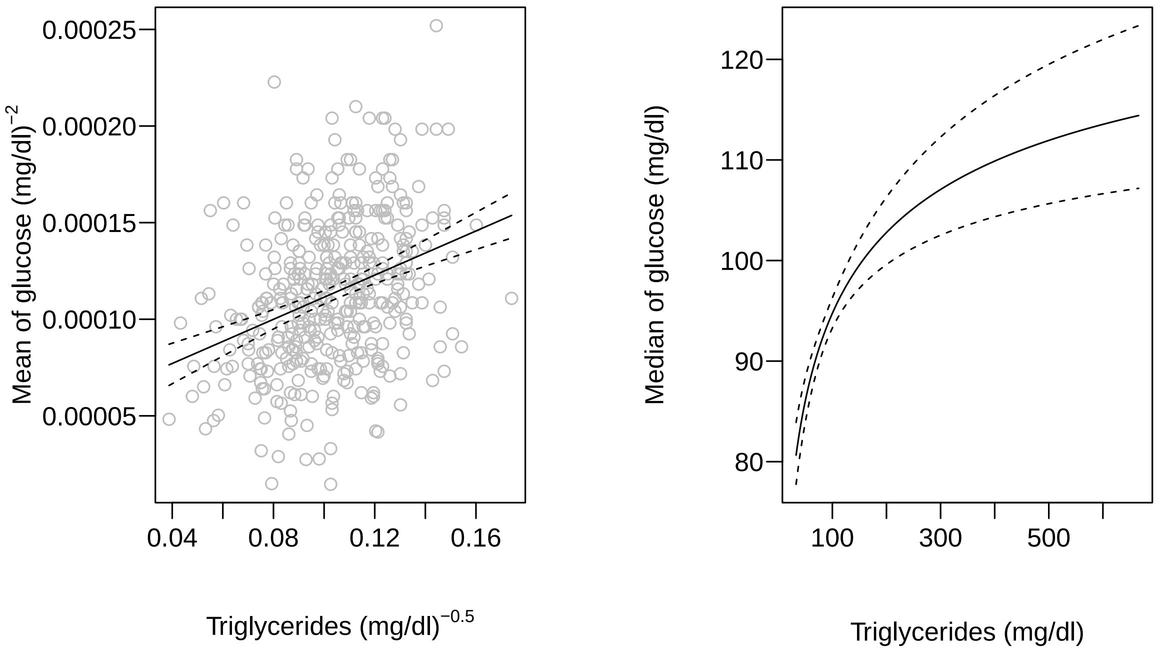

Figure 6: Visualization of the fitted model `modglucose`. Left: In the transformed space, the reciprocal of the squared mean of glucose levels as a function of the reciprocal of the square root of triglycerides levels is shown. Note that this plot should not be reported since the reciprocal of the squared mean of glucose levels and the square root of triglycerides levels are meaningless. This type of plot is intended just to visually explore the model goodness of fit. Right: In the original space, the geometric mean (or equivalently median) of glucose levels as a function of triglycerides levels is shown. This type of plot is appropriate for reporting. Dashed lines represent 95% confidence intervals for the measure.

```
##
## Computing effects...

effectXdiff

##
## Adjusted change in the median of the response variable when the explanatory
## variable changes from x1 to x2 (confidence intervals based on 999 bootstrap
## samples):
##
##    x1  x2 EstimateDiff lower95%  upper95% EstimatePercent lower95%  upper95%
## 1  50 100     8.703222 6.482939 10.849441       10.114803 7.354460 12.871487
## 2 100 150     4.802290 3.408412  6.335814        5.068507 3.592610  6.660220
## 3 150 200     3.235200 2.188123  4.493912        3.249826 2.242664  4.439129
## 4 200 250     2.397980 1.624646  3.312092        2.333003 1.615346  3.140562
```

```
# Effects for an percent change in triglycerides level:
xq <- 50 * 1.5^(0:4)
xq

## [1]  50.000  75.000 112.500 168.750 253.125

effectXperc <- effect(modglucose, x1 = xq)

##
## Computing effects...

effectXperc

##
## Adjusted change in the median of the response variable when the explanatory
## variable changes from x1 to x2 (confidence intervals based on 999 bootstrap
## samples):
##
##        x1      x2 EstimateDiff lower95% upper95% EstimatePercent lower95%
## 1  50.00  75.000     5.152950 3.897949 6.328020        5.988710 4.421962
## 2  75.00 112.500     4.971764 3.635856 6.362094        5.451653 3.942760
## 3 112.50 168.750     4.724024 3.271731 6.385583        4.912204 3.436794
## 4 168.75 253.125     4.420430 3.018380 6.055787        4.381299 3.040427
##   upper95%
## 1 7.504820
## 2 7.013019
## 3 6.603796
## 4 5.883341
```

Exploring previous results, one can see that the more easily interpretable effect appears to be the additive change in the median of glucose level associated to a percent change in triglycerides level. Indeed, we can see that, for any given value of the triglycerides level along the observed range, a 50% increase in triglycerides level is associated to around a 4.8 mg/dl increase in the median glucose level. For the other measures, the effect is more dependent on the basal value of the triglycerides level.

3.2 Logistic regression model with log transformation in the explanatory variable

Revisiting the cotinine example (Section 3.1.2), supose we are now interested in the association between low birth weight (defined as weight lower than 2500 g), Y, and cotinine level, X, after log transforming X. The model can be fitted as follows:

```
modcot2 <- tlm(underweight ~ logcotinine, family = binomial, data = cotinine, xpow = 0)
```

where `xpow = 0` indicates that the explanatory variable is already log transformed and the argument `family = binomial` indicates that the regression model is logistic with logit link (default is `family = gaussian`, for the lineal regression model). The fitted model provides the following results:

```
summary(modcot2)

##
## Logistic regression fitted model in the transformed space
## --------------------------------------------------------
##
## Transformations:
##    In the response variable: logit link for logistic regression
##    In the explanatory variable: log
##
## Call:
## glm(formula = underweight ˜ logcotinine, family = binomial, data = cotinine)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.5146     0.4539  -7.744 9.65e-15 ***
## logcotinine   0.3306     0.1289   2.566   0.0103 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 175.09  on 350  degrees of freedom
## Residual deviance: 168.45  on 349  degrees of freedom
## AIC: 172.45
##
## Number of Fisher Scoring iterations: 5
```

Then, we can obtain the probability of low birth weight as a function of cotinine level:

```
MY(modcot2)

##
## Estimated adjusted probability of the response variable in the original space:
##
##           X       P(Y)    lower95%   upper95%
## 1    0.2000 0.01717866 0.004920113 0.05819342
## 2  101.2889 0.12049711 0.072138318 0.19447904
## 3  202.3778 0.14693118 0.079055696 0.25683106
## 4  303.4667 0.16452717 0.082757167 0.30061256
## 5  404.5556 0.17801453 0.085288433 0.33466844
## 6  505.6444 0.18906298 0.087213645 0.36260410
## 7  606.7333 0.19847559 0.088768165 0.38629112
## 8  707.8222 0.20670597 0.090072508 0.40684022
## 9  808.9111 0.21403755 0.091196603 0.42496992
## 10 910.0000 0.22066025 0.092184631 0.44117414
```

A graphical representation of the relationship between cotinine level and low birth weight can be obtained by:

```
plot(modcot2, type = "transformed", xname = "Cotinine (ng/ml) levels",
     yname = "low birth weight")
plot(modcot2, xname = "Cotinine (ng/ml) levels", yname = "low birth weight")
```

which provides Figure 7.

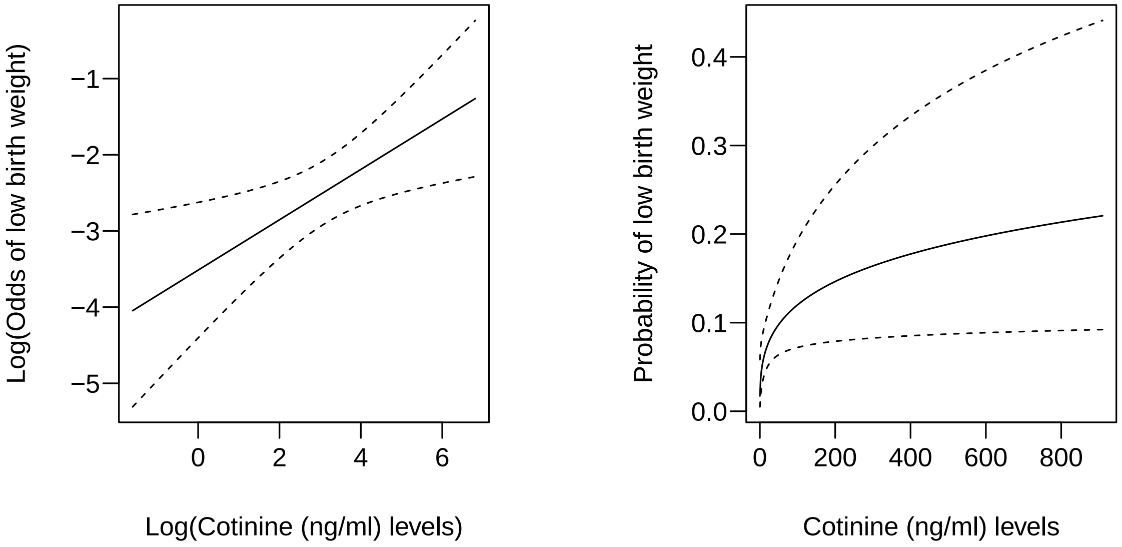

Figure 7: Visualization of the fitted model `modcot2`. Left: In the transformed space, the logarithm of the odds of low birth weight as a function of the logarithm of cotinine levels is shown. Note that this plot should not be reported since the logarithm of cotinine levels is meaningless, and the logarithm of the odds is difficult to interpret. This type of plot is intended just to visually explore the model goodness of fit. Right: In the original space, the probability of low birth weight as a function of cotinine levels is shown. This type of plot is appropriate for reporting. Dashed lines represent 95% confidence intervals for the measure.

Regarding effects, the function `effectInfo` indicates that, under the fitted model, we can summarize the effect of cotinine level (X) on low birth weight (Y) in terms of odds ratio (OR) for a percent (or multiplicative) change in cotinine level:

```
effectInfo(modcot2)

##
## The effect of X on Y can be summarized with a single number as follows:
##
##  - Change in X: multiplicative of factor q (equivalently, adding an r = 100 * (q - 1)% to X)
##  - Type of effect on Y: odds ratio of Y
##  - Effect size: q^beta
```

```
##
##    beta coefficient estimate:
##              Estimate Std. Error  z value   Pr(>|z|)
## logcotinine 0.3306394  0.1288549 2.565982 0.01028842
##
## Further details can be obtained using effect().
```

Thus, the function `effect` provides as default the OR of low birth weight for a percent change in the cotinine level equal to the interquartile ratio:

```
effect(modcot2)

##
## Computing effects...
##
## Adjusted odds ratio of the response variable for an 'r'% change in
## the explanatory variable equivalent to the interquartile ratio:
##
##          r Estimate lower95% upper95%
## 1 1112.878 2.282194  1.21516 4.286193
##
## For further information on interpreting the effect use effectInfo().
```

Alternatively, the effect for a 10-fold change in the cotinine level is:

```
effect(modcot2, q = 10)

##
## Computing effects...
##
## Adjusted odds ratio of the response variable for an 'r' = 900% change
## in the explanatory variable:
##
##     r Estimate lower95% upper95%
## 1 900 2.141112 1.196984 3.829926
##
## For further information on interpreting the effect use effectInfo().
```

Bibliography

[1] J. Barrera-Gómez and X. Basagaña. Models with transformed variables: interpretation and software. *Epidemiology*, 26(2):e16–17, 2015. URL https://doi.org/10.1097/EDE.0000000000000247.

[2] M. Rivera, X. Basagaña, I. Aguilera, M. Foraster, D. Agis, E. de Groot, L. Perez, MA. Mendez, L. Bouso, J. Targa, R. Ramos, J. Sala, J. Marrugat, R. Elosua, and N. Künzli. Association between long-term exposure to traffic-related air pollution and subclinical atherosclerosis: The REGICOR study. *Environmental Health Perspectives*, 121(2):223–230, 2013. URL https://doi.org/10.1289/ehp.1205146.

[3] AC. Davison and DV. Hinkley. *Bootstrap Methods and their Application*. Cambridge University Press, New York, 1997.

[4] S. Pichini, X. Basagaña, R. Pacifici, O. Garcia, C. Puig, O. Vall, J. Harris, P. Zuccaro, J. Segura, and J. Sunyer. Cord serum cotinine as a biomarker of fetal exposure to cigarette smoke at the end of pregnancy. *Environmental Health Perspectives*, 108(11):1079–1083, 2000. URL https://doi.org/10.2307/3434962.

[5] X. Basagaña, M. Torrent, W. Atkinson, C. Puig, M. Barnes, O. Vall, M. Jones, J. Sunyer, P. Cullinan, and AMICS study. Domestic aeroallergen levels in Barcelona and Menorca (Spain). *Pediatric Allergy and Immunology*, 13(6):412–417, 2002. URL https://doi.org/10.1034/j.1399-3038.2002.02081.x.

[6] SR. Searle, FM. Speed, and GA. Milliken. Population marginal means in the linear model: An alternative to least squares means. *The American Statistician*, 34(4):216–221, 1980. URL https://doi.org/10.2307/2684063.

[7] RK Jr. Elswick, PF. Schwartz, and JA. Welsh. Interpretation of the odds ratio from logistic regression after a transformation of the covariate vector. *Statistics in Medicine*, 16(15):1695–1703, 1997. URL https://doi.org/10.1002/(sici)1097-0258(19970815)16:15<1695::aid-sim601>3.0.co;2-v.