Package ltable 2.0

Ocheredko Oleksandr

FUNCTIONALITY

1. Constructs tables of counts and proportions out of data sets.
2. Inserts table into Excel and Word documents using clipboard, into LaTeX, HTML, Markdown and reStructuredText documents by the knitr::kable agency.
3. Moulds table into acceptable for log-linear modeling data.frame.
4. Performs log-linear modeling.
5. Performs power analysis.

In order to perform log-linear and power analyses GSL: GNU Scientific Library has to be installed first. GSL: GNU Scientific Library[1]. So far features 4,5 included in Unix (MacOS) package installer v.2.0.1 only.

[1] http://www.gnu.org/software/gsl

Construction of tables of counts and proportions out of data sets

Use function table_f():

$table\_f(data,\ datavars,\ type\ =\ 1,\ digits\ =\ 2,\ extended\ =\ FALSE,\ MV\ =\ FALSE,\ cb\ =\ FALSE)$

Examples:

```
data(sdata, package="ltable")
sdata
```

```
##        a  b      c    d
## 1   TRUE NA   male    A
## 2     NA  1   male    B
## 3  FALSE  1   male    A
## 4   TRUE  1   male <NA>
## 5   TRUE  1   male    A
## 6   TRUE  2 female    B
## 7  FALSE  2 female    A
## 8  FALSE  2 female    B
## 9   TRUE  2 female    A
## 10 FALSE  2 female    B
## 11    NA NA   <NA> <NA>
## 12  TRUE  1   male    A
## 13 FALSE  1   male    B
## 14 FALSE  1   male    A
## 15  TRUE  1   male    B
## 16  TRUE  1   male    A
## 17  TRUE  2 female    B
## 18 FALSE  2 female    A
## 19 FALSE  2 female    B
## 20  TRUE  2 female    A
## 21 FALSE  2 female    B
## 22    NA NA   <NA> <NA>
```

```
lapply(sdata,class)
```

```
## $a
## [1] "logical"
##
## $b
## [1] "numeric"
##
## $c
## [1] "factor"
##
## $d
## [1] "character"
```

I built data.frame *sdata* with fields of different basic classes just for demonstration. No other meaning applies. Let's build a simple table across levels of *a*:

```
ltable::table_f(sdata, "a")
```

```
##   a:FALSE a:TRUE Total, N
## 1       9     10       19
```

One might have interest in *NA* values for there may be quite informative pattern across levels or levels combinations. Use *MV=TRUE*. It's a part of data exploration:

```
ltable::table_f(sdata, "a", MV=TRUE, ext=TRUE)
```

```
##   a:FALSE a:TRUE NA Total, N
## 1       9     10  3       22
```

Unrelated option *extended=TRUE* is used just to demonstrate that abundant args have no effect. If one wants to tabulate numerous factors it's important to arrange them properly in sequence of presentation delimited with comma ",". Sorted levels of all but last variable are rolled out vertically in indicated sequence, the last has its sorted levels spread by columns.

```
ltable::table_f(sdata, "b,c")
```

```
##            b c:female c:male Total, N
## 1          1        0      9        9
## 2          2       10      0       10
## sum Total, N       10      9       19
```

One can also obtain the table of frequencies by choosing arg *type* values { *2*, *3*, *4* } as shown below:

```
ltable::table_f(sdata, "a,c",
                type=2, digits=3)
```

```
##            a c:female c:male Total, p
## 1      FALSE    0.667  0.333         1
## 2       TRUE      0.4    0.6         1
## sum Total, p    0.534  0.466         1
```

```
ltable::table_f(sdata, "a,c",
                type=3, digits=2)
```

```
##            a c:female c:male Total, p
## 1      FALSE      0.6   0.33      0.47
## 2       TRUE      0.4   0.67      0.53
## sum Total, p        1      1         1
```

```
ltable::table_f(sdata, "a,c",
                type=4, digits=3)
```

```
##            a c:female c:male Total, p
## 1      FALSE    0.316  0.158    0.474
## 2       TRUE    0.211  0.316    0.527
## sum Total, p    0.527  0.474    1.001
```

One can include number of fields (variables):

```
options(width=40)
ltable::table_f(sdata, "a,b,c,d",
        type=2, digits=3)
```

```
##            a        b        c   d:A
## 1      FALSE        1   female     0
## 2      FALSE        1     male 0.667
## 3      FALSE        2   female 0.333
## 4      FALSE        2     male     0
## 5       TRUE        1   female     0
## 6       TRUE        1     male  0.75
## 7       TRUE        2   female   0.5
## 8       TRUE        2     male     0
## sum Total, p Total, p Total, p 0.562
##       d:B Total, p
## 1       0        0
## 2   0.333        1
## 3   0.667        1
## 4       0        0
## 5       0        0
## 6    0.25        1
## 7     0.5        1
## 8       0        0
## sum 0.438        1
```

arg value *extended=TRUE* adds margins of counts, applied only for proportions and frequencies, value is *FALSE* by default. In last two examples options(width) was used to accommodate tables:

```
options(width=40)
ltable::table_f(sdata, "b,c,a,d", type=2,
                digits=3, extended=TRUE)
```

```
##            b        c        a   d:A
## 1          1   female    FALSE     0
## 2          1   female     TRUE     0
## 3          1     male    FALSE 0.667
## 4          1     male     TRUE  0.75
## 5          2   female    FALSE 0.333
## 6          2   female     TRUE   0.5
## 7          2     male    FALSE     0
## 8          2     male     TRUE     0
## sum Total, p Total, p Total, p 0.562
##     Total, N Total, N Total, N     9
```

```
##       d:B Total, p Total, N
## 1       0         0         0
## 2       0         0         0
## 3   0.333         1         3
## 4    0.25         1         4
## 5   0.667         1         6
## 6     0.5         1         4
## 7       0         0         0
## 8       0         0         0
## sum 0.438         1        17
##         8        17        17
```

Transporting table into documents

One can paste table into clipboard by using arg *cb=TRUE*. To insert table into Word document one should first open Excel, choose left high corner of placement by mouse click and use copy and paste key combinations or click on the Copy and Paste icons (the clipboard), then open Word document, use Copy icon to place the table. $table\_f(sdata,\ "a,c",\ type = 2,\ digits = 3,\ cb = TRUE)$

Use knitr::kable() to import table to other available formats through .Rmd or other engines:

$t <- table\_f(sdata,\ "a,c",\ type = 2,\ digits = 3)$

$knitr :: kable(t)$

	a	c:female	c:male	Total, p
1	FALSE	0.667	0.333	1
2	TRUE	0.4	0.6	1
sum	Total, p	0.534	0.466	1

Transforming table into acceptable for modelling data.frame.

Use function tableToData():

tableToData(tname, numerictype = "", orderedtype = "")

Example:

```
data(sdata, package="ltable")
stab<-ltable::table_f(sdata, "a,b,c")
```

```
sdat<-ltable::tableToData(stab,
                          numerictype ="b",
                          orderedtype="a,c")
sdat
```

```
##       a b      c Counts
## 1 FALSE 1 female      0
## 2 FALSE 2 female      6
## 3  TRUE 1 female      0
## 4  TRUE 2 female      4
## 5 FALSE 1   male      3
## 6 FALSE 2   male      0
## 7  TRUE 1   male      5
## 8  TRUE 2   male      0
```

```
lapply(sdat,class)
```

```
## $a
## [1] "ordered" "factor"
##
## $b
## [1] "numeric"
##
## $c
## [1] "ordered" "factor"
##
## $Counts
## [1] "numeric"
```

Arg *tname* is the name of table created by function *table_f()*. In both next args *numerictype* and *orderedtype* variable names separated by comma to be transformed to numeric or ordered factor classes. Variable "Counts" shouldn't be listed in both.

Log-linear modeling

Use function PowerPoisson():

$PowerPoisson(formula,\ data,\ scale\_min = 1,\ scale\_max = 5,\ effect,\ p\_alpha = 0.05,\ contrasts = NULL)$

Log-linear analysis features some advantages against $glm\{stats\}$, first of all due to stability of GSL IWLS algorithms that insures distinctly less biased covariances estimates, the pivot issue for implemented power analysis. In some instances hypothesis testing of higher order effects disagrees with that of *glm* on account of larger GSL estimated errors. Another though related enhancement is distinct better fit assessed by sum of squared differences between observed and expected counts.

Example

Let's begin with historical example of log-linear modeling with Tromboembolism Data. This case-control data first considered by Worcester, J. (1971). The data y[ijk] cross-classify thromboembolism and control patients (i=1 and 2 respectively) by two risk factors: oral contraceptive user (j =1 for user, j = 2 for non-user) and smoking (k=1 for smokers, k =2 for non-smokers). Test quantifies boosting effect of contraceptive on odds of thromboembolism using log-linear analysis. Reproduced grouped data frame with 8 rows of factors' levels combinations is given below. Factors are: smoking status (Yes, No), contraceptive usage (Yes, No), thromboembolism status (Trombol, Control).

```
data(tdata, package="ltable")
tdata
```

```
  smoker contraceptive   tromb Counts
1    Yes           Yes Trombol     14
2    Yes           Yes Control      2
3    Yes            No Trombol      7
4    Yes            No Control     22
5     No           Yes Trombol     12
6     No           Yes Control      8
7     No            No Trombol     25
8     No            No Control     84
```

Data has been used in subsequent model choice studies, such as Spiegelhalter and Smith (1982), Pettit and Young (1990), Congdon (2005).

Under the potentially informative priors used, the Bayes factor estimate was B_{21} = 23.8, quite strongly in favour of the smaller model with single interaction effect *contraceptive*tromboembolism* that was opted for consideration in example. The fact that the reduced model gives a close fit implies that the use of oral contraceptives indeed instigates the odds of thromboembolism, effect significancy supported by classical and MCMC based log-linear estimates. Further inclusion of third order interaction indicated that the use of oral contraceptives particularly among those who smoke, is a risk for thromboembolism, but for smokers who do not take the pill there is no excess risk.

Let's check hypothesis by $glm\{stats\}$ function:

```
res<-glm(Counts~ smoker +contraceptive +
         tromb + contraceptive*tromb,
         family="poisson",
         data=tdata)
```

Then compare output with that of $PowerPoisson\{ltable\}$ function:

```
pres<-ltable::PowerPoisson(Counts~
      smoker +contraceptive +tromb +
      contraceptive*tromb,
      effect="contraceptive*tromb",
      scale_max=1.5, data=tdata)
```

Results of both log-linear modelings are given below.

Results of glm {stats} modeling

```
options(width=80)
summary(res)$coefficients
```

```
                             Estimate Std. Error   z value     Pr(>|z|)
(Intercept)                  4.364196  0.1069522 40.805108 0.000000e+00
smokerYes                   -1.053150  0.1731305 -6.082984 1.179660e-09
contraceptiveYes            -2.360854  0.3308080 -7.136628 9.564814e-13
trombTrombol                -1.197703  0.2017027 -5.937964 2.885828e-09
contraceptiveYes:trombTrombol  2.153215  0.4232558  5.087265 3.632639e-07
```

With the help of external calculus facilitated by function *condition_number*() {*GpGp*}:
Jacobian reciprocal condition number = 0.179
chisq/dof = 56.14

Results of PowerPoisson {ltable} modeling

```
options(width=80)
ltable::print(pres, choice="model")
```

```
Coefficients:
                                Estimate Std.Error |z-score|   Pr(>|z|)
(Intercept)                     4.43e+00  1.06e-01  4.18e+01   0.0e+00
smokerYes                      -1.32e+00  1.90e-01  6.97e+00   3.1e-12
contraceptiveYes               -2.35e+00  1.44e-01  1.63e+01   8.2e-60
trombTrombol                   -1.20e+00  1.88e-01  6.41e+00   1.5e-10
contraceptiveYes:trombTrombol   1.68e+00  5.44e-01  3.09e+00   2.0e-03

Model fit:
Weighted nonlinear least-squares fitting
Method|Solver: Levenberg-Marquardt
Algorithm: trust region
initial |f(x)| =  17.95263
final |f(x)| =  0.9301596
Jacobian reciprocal condition number =  11.12248
number of iterations =  10
reason for stopping:  small step size
chisq/dof =  0.288399
status:  success
```

The *print* {*ltable*} also conveys info:

Jacobian reciprocal condition number measures the inverse sensitivity of the solution to small perturbations in the input data. It tends to zero as J tends to singularity indicating solution instability.")

The value of ch-squared per degree of freedom (chisq/dof) approximately 1 indicates a good fit.) If chisq/dof » 1 the error estimates obtained from the covariance matrix will be too small and should be multiplied by square root of chisq/dof .

Poor fit will result from the use of an inappropriate model, and the scaled error estimates may then be outside the range of validity for Gaussian errors.

BEWARE: Poor fit jeopardizes the validity of power analysis.

Juxtaposing two results we have the same conclusion on effects, specifically on hypothesized second order interaction term *contraceptive*tromb*, though differences are conspicuous on a part of error terms, higher order effect in particular. Checking with other data sets the regularity holds, that is higher order effects estimates feature larger errors against *glm* {*stats*} counterparts. GLS estimates are more reliable given much better condition of parameters covariance matrix. Given example just follows the suit: Jacobian reciprocal condition number in *glm* {*stats*} is alarming while keeping good property in GSL IWLS algorithms. The same rests with *chisq/dof* statistic. Repercussion on power analysis is about to be demonstrated.

Power analysis

Outlines of offered power study methodology can be found in ISDSA[2] paper.

Use function PowerPoisson():

$PowerPoisson(formula,\ data,\ scale\_min = 1,\ scale\_max = 5,\ effect,\ p\_alpha = 0.05,\ contrasts = NULL)$

formula

- Incorporation of formula based approach facilitates extracting true influence of hypothesized effect by catching other intermingled influences. It's up to investigator's acumen and experience in process under study to delineate and separate hypothesized effect by appropriate data collection design and model formulation.
- The issue resolved is contrasts that constitute effect. Mostly investigator is interested in contrasts rather than effect. Say, if one proceeds with clinical trial to test medicines A, B, C, D it's A (new drug) against traditional set that usually implied. If the optimal dosage is under consideration, they are contrasts that help out (average against min, max; max against others, etc.).

scale_min, scale_max

Indicate the range of sample sizes. *scale_min* is the smallest number of sample size scale range, 1 signifies the given data sample size (observed total counts). *scale_max* is maximal sample size considered in power analysis. 5 by default means 5 times observed counts. The inspected sample size range defined by scale_min - scale_max automatically is divided into 11 consecutive values investigated by

[2] https://meeting.isdsa.org/index.php/isdsa/2019/paper/viewPaper/3

function. Given the results one can change sample size range, for example to scrutinize some particular interval to ensure power and p-value.

effect

Represents quoted effect tested by hypothesis; it should be one from the model formula, of second or higher order, introduced by * delimiter, i.e., "y*x", "y1*y2*x1*x2", "y1*y2", etc.

p_alpha

Serves to signify Z to check simulated z-scores against in power analysis, 0.05 by default.

contrasts

Serves to choose types of contrasts to study effects of factors, the same with *glm* {*stats*}, orthogonal polynomials by default.

Example

Let's begin with Tromboembolism Data.

```
options(width=40)
pres<-ltable::PowerPoisson(Counts~
      smoker +contraceptive +tromb +
      contraceptive*tromb,
      effect="contraceptive*tromb",
      scale_max=1.5, data=tdata)
ltable::print(pres, choice="power")
```

```
Test statistic Z:  Quantiles
Sample size:    Q0.025  Q0.05   Q0.5
174             1.463   1.548  3.196
183             1.569   1.634  3.277
191             1.321   1.903  3.308
200             1.773   1.849  3.421
209             1.801   1.974  3.463
218             1.938   2.184  3.707
226             2.057   2.315  3.888
235             2.186   2.501  3.783
244             2.457   2.682  4.113
252             2.087   2.589  4.326
261             2.508   2.831  4.193

Power:          Quantiles
Sample size:    Q0.025  Q0.05   Q0.5
174              0.86    0.86   0.92
183              0.86    0.88   0.94
191              0.89    0.90   0.96
200              0.92    0.92   0.96
209              0.91    0.94   0.98
218              0.94    0.94   0.98
226              0.95    0.96   1.00
235              0.96    0.96   1.00
244              0.96    0.98   1.00
252              0.98    0.98   1.00
261              0.98    0.98   1.00
```

This is a short print. Real print also lists quantiles Q0.1, Q0.2, Q0.3, Q0.4. What we can deduce from the result is that 235 total counts is enough to secure *alpha* and *beta* errors. I suggest the most secure Q0.025 quantile to weight decision on. So 235 secures Z=1.96 and power 0.9 given Q0.025 estimates. Results of power analysis backed up with MCMC delivered approach, see Ocheredko O.M. MCMC Bootstrap Based Approach to Power and Sample Size Evaluation.[3].

Discussion

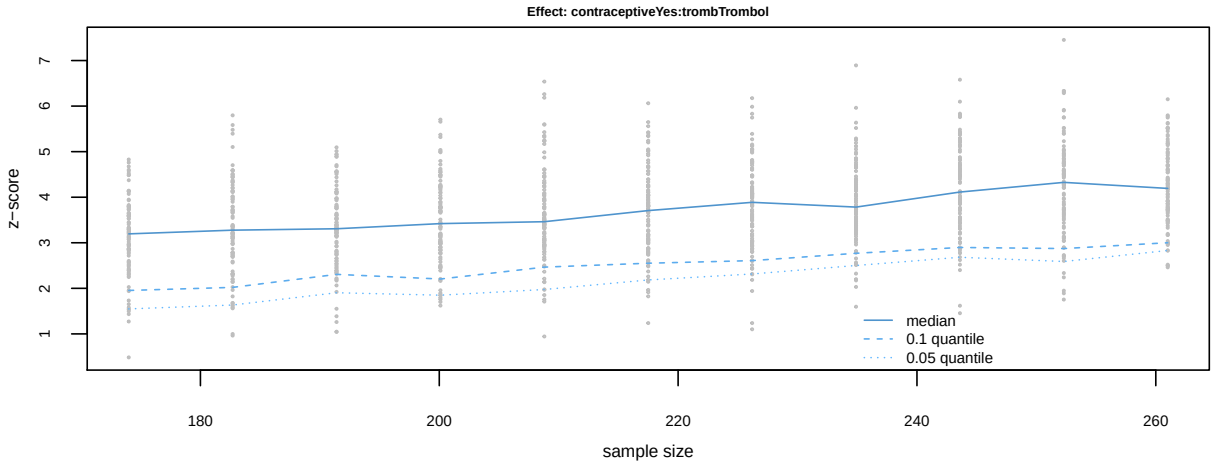
The log-linear estimates of *contraceptiveYes*tromb Trombol* effect tested to be significant. Is it not strong enough evidence of association? Why should we collect almost 1.5-fold as many data? The answer of course is related to the specifics of the sample. The basic design itself is a sample, not status quo that represents true frequencies ratios in population. Therefore, we have to secure that the sample data brings in enough information to overpower sample specifics. Of course, the more complex design is the larger sample variation has to be outbalanced by signal, the larger sample size is required.

[3] https://www.amazon.com/gp/product/1946728039/

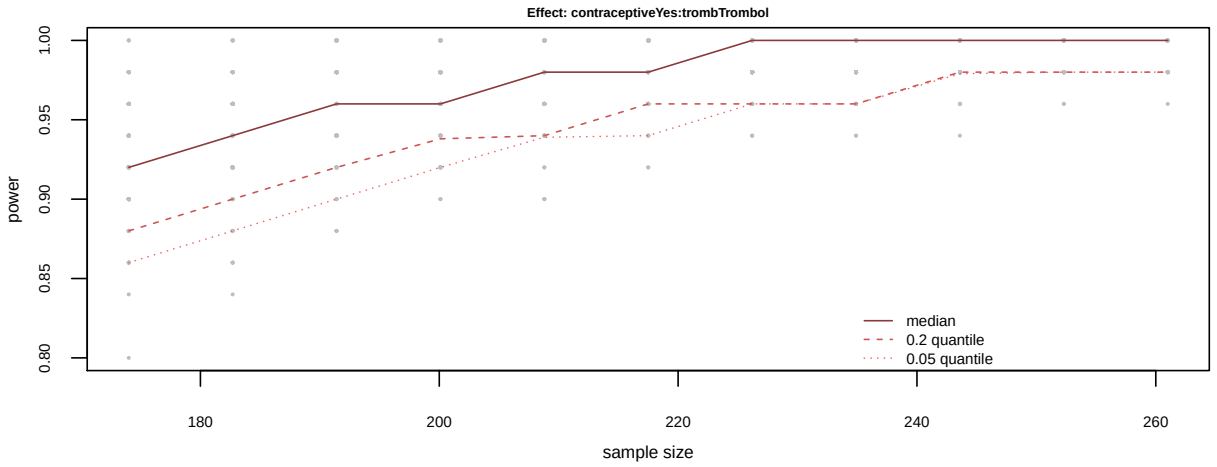
The original data is one of the random snapshots of reality and we have to put as much credit as sensible to it. Not all snapshots of size 174 guarantee a 95% CI with zero excluded. Under MCMC approach it was indicated that the sample size of 260 affords enough power to assure the significance of the association in practically all samples. The same logic is behind any application of power analysis.

The other lay belief is that with the increase of sample size any association is doomed to be significant. For sure, it is not, and the strength of power analysis is to determine the optimal sample size of hypothesis testing. The power analysis assures that given H_0 is true there is no prospect of decisive augmentation of power and significance following the increase in sample size that will shortly be demonstrated. Before turning to another example the graphic output produced by function *plot* {*ltable*} is paneled:

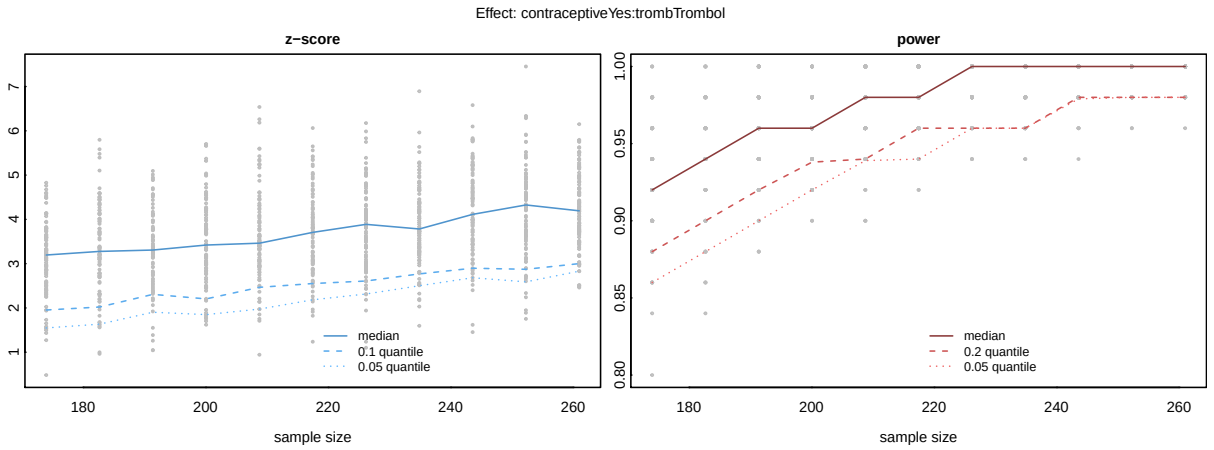
```
ltable::plot(pres, stencil=1)
```




```
ltable::plot(pres, stencil=2)
```



```
ltable::plot(pres, stencil=3)
```

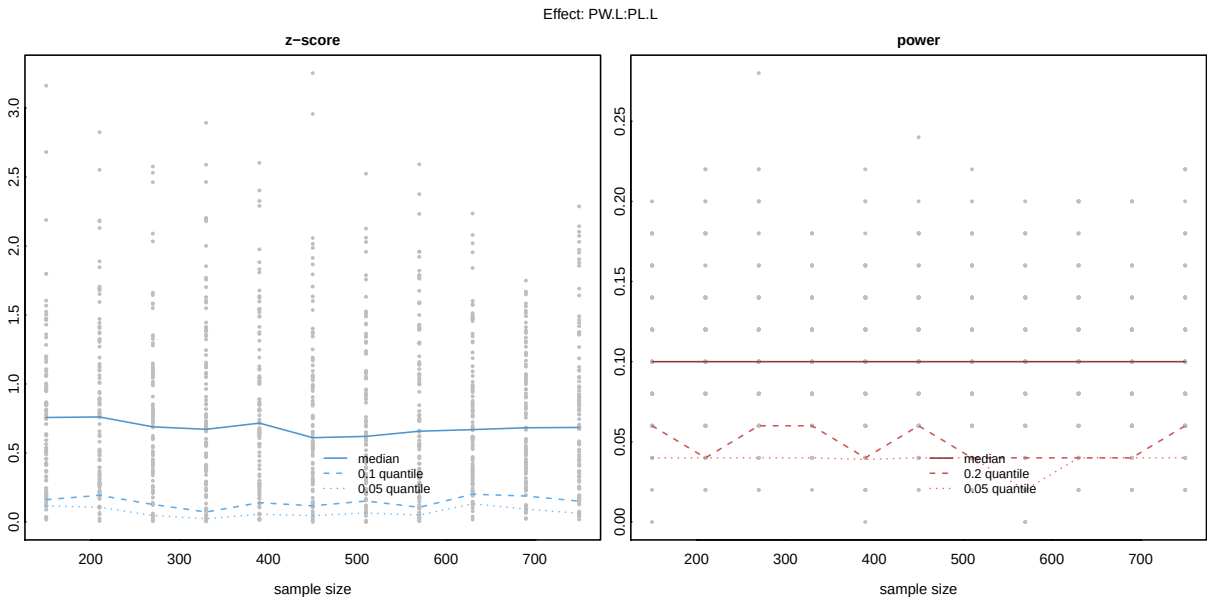


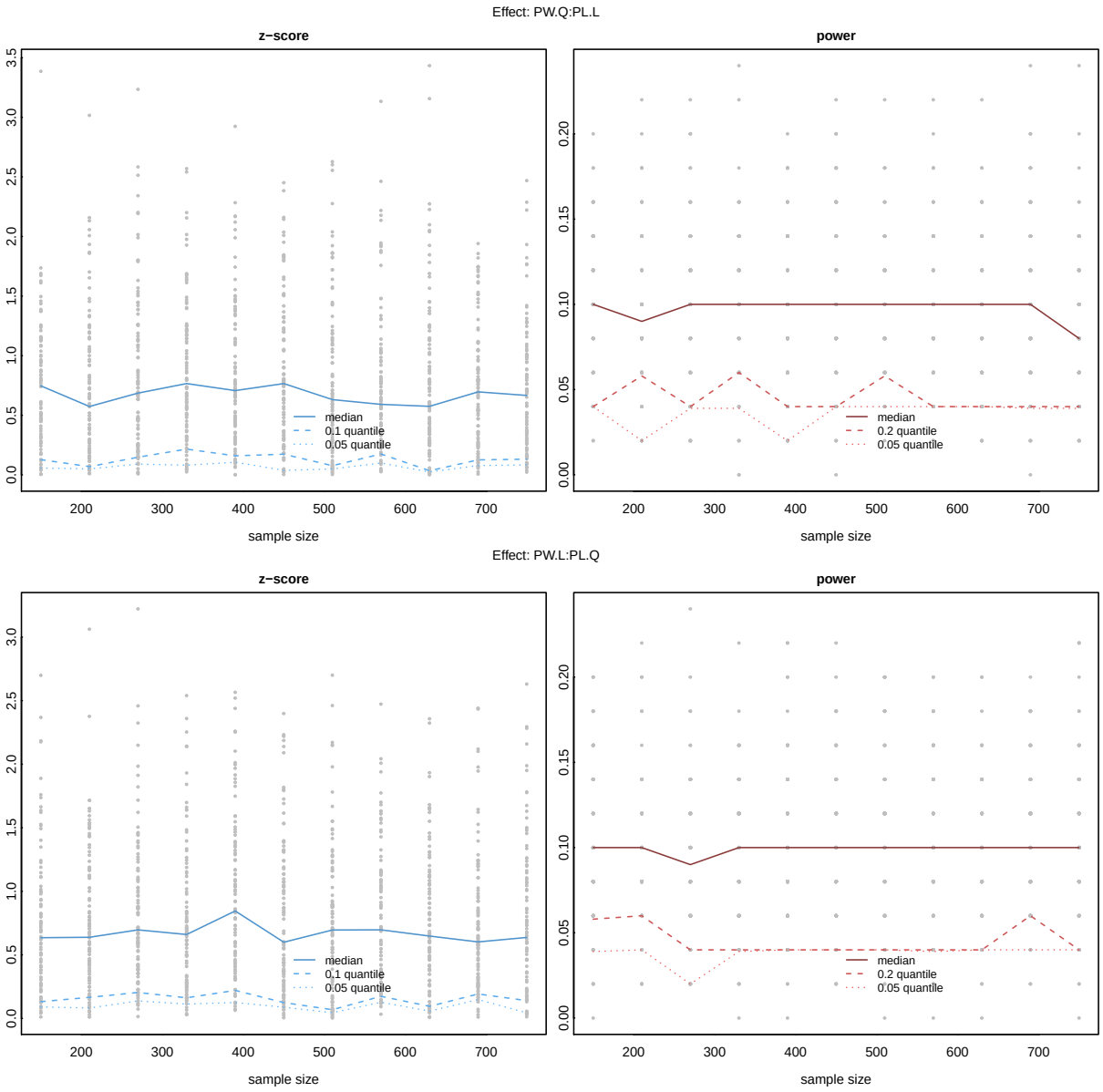
Example

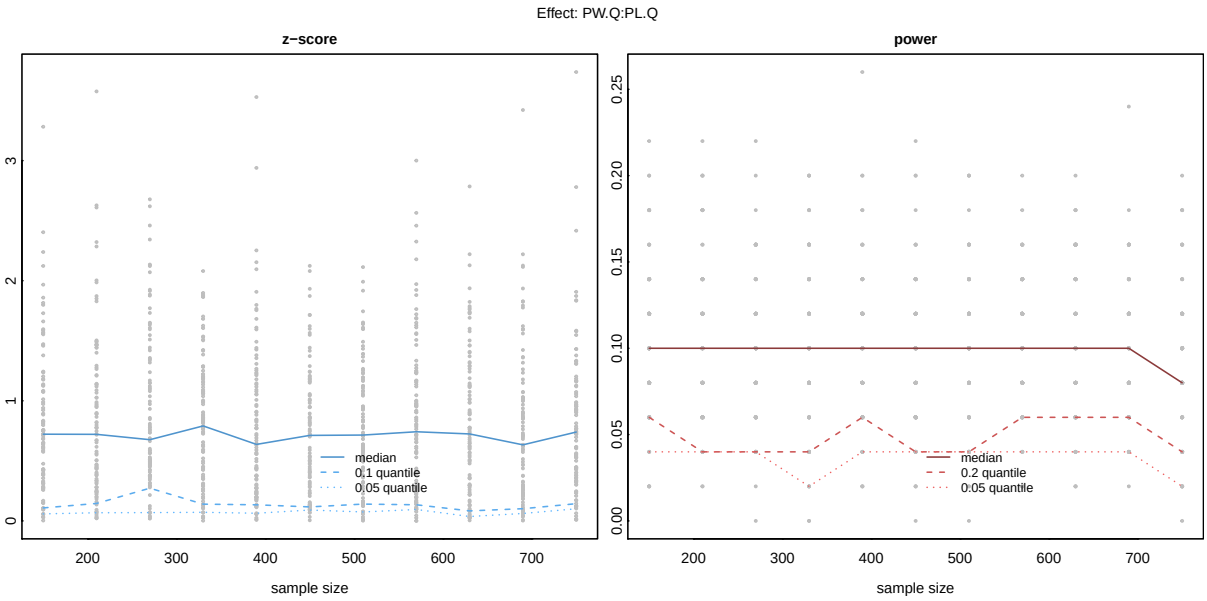
This is example of no observed association

```
data(iris)
iriscut<-with(iris, data.frame(PL=cut(Petal.Length,3),
                               PW=cut(Petal.Width,3)))
irist<-ltable::table_f(iriscut,"PL,PW")
```

```
irisd<-ltable::tableToData(irist, ordered="PL,PW")
irisres<-ltable::PowerPoisson(Counts~PW+PL+PW*PL, effect="PW*PL", data=irisd)
ltable::plot(irisres, st=3)
```




Effect: PW.Q:PL.L
z−score
power
median
0.1 quantile
0.05 quantile
median
0.2 quantile
0.05 quantile
sample size
sample size
Effect: PW.L:PL.Q
z−score
power
median
0.1 quantile
0.05 quantile
median
0.2 quantile
0.05 quantile
sample size
sample size



What do we make of it?

1. There is no chance to observe significant association by accumulating data if used tabulated design reproduces natural frequencies that indicate no natural relationship.
2. There is no increase in both significance and power with sample size growth given H_0 is true.
3. Power and significance may behave differently with sample size dynamic, so that we can't play one against the other as classical power methodology implies. Usually one is less responsive than another and it is former that defines necessary data load.

Explaining differences

Implemented is classical IWLS approach based upon minimization of squared differences between observed (**p**) and expected (π) proportions. Observed proportions are transformed into response functions of proportions, usually denoted by **F(p)**, to accommodate logarithmic link and dimension reduction. Weighting is done with asymptotic covariance matrix of **F(p)**, analytically derived by *delta method* from covariance matrix of observed proportions. For analytical details see for example Agresti, Alan. Categorical Data Analysis, 3rd Edition. Chapter 16.7[4]. Resulting is usually well conditioned parameters covariance matrix. Apparently, $gls\{stat\}$ uses different approach, subsumed into *minimum chi-squared estimators*. I checked tromboembolism data with *modified chi-square minimization* approach using GSL. Model estimation is given below and coincides with $gls\{stat\}$ output perfectly:

[4]https://www.wiley.com/en-gb/Categorical+Data+Analysis%2C+3rd+Edition-p-9780470463635

Model estimation with modified chi-square minimization approach using GSL

Coefficients:	Estimate	Std.Error	|z-score|	Pr(>|z|)
(Intercept)	4.36e+00	1.07e-01	4.08e+01	0.0e+00
smokerYes	-1.05e+00	1.73e-01	6.08e+00	1.2e-09
contraceptiveYes	-2.36e+00	3.31e-01	7.14e+00	9.6e-13
trombTrombol	-1.20e+00	2.02e-01	5.94e+00	2.9e-09
contraceptiveYes: trombTrombol	2.15e+00	4.23e-01	5.09e+00	3.6e-07

Model fit:
Weighted nonlinear least-squares fitting
initial deviance = 166.4391
final deviance = 11.13575
Jacobian reciprocal condition number = 0.1791111
number of iterations = 5
reason for stopping: converged
chisq/dof = 56.14586
fitted value on the boundary: FALSE

Model fit estimates coincide with those made externally. Neyman (1949) [5] introduced modified chi-square minimization estimators. He noted that minimum modified chi-square estimates result from minimizing

$$\sum_{i=1}^{N} \frac{(p_i - \pi_i)^2}{p_i} + \sum_{j=1}^{N-q} \lambda_j g_j(\pi_1, ..., \pi_N)$$

with constraint equations g_j of the form:

$$log\pi_i j - log\pi_{i,j+1} - log\pi_{i+1,j} + log\pi_{i+1,j+1} = 0$$

with respect to π, where λ_j are Lagrange multipliers. When the constraint equations are linear in π, the resulting estimating equations are linear. Then Bhapkar (1996) [6] showed that these estimators are identical to WLS estimators. Usually, however, constraint equations are nonlinear in π, such as for independence model. Given our interest in interaction effects (in particular issue for power analysis) estimates generally should not coincide advocating for IWLS.
Finally, function $MCMCglmm\{MCMCglmm\}$ is used arbitrary on tromboembolism data. MCMC has numerous advantages over classical estimators YET required sophisticated users to tune in. Let's look at the MCMC estimates, in particular at interaction effect, which significance is close to produced by

[5] Neyman, J. 1949. Contributions to the theory of the chi-squared test. In the Proceedings First Berkeley Symposium on Mathematical Statistics and Probability, ed. J. Neyman. Berkeley, CA: University of California Press, pp. 239-273.

[6] Bhapkar, V.P. 1996. A note on the equivalence of two test criteria for hypotheses in categorical data. J.Am.Stat.Assoc. 61: 228-235.

$PowerPoisson\{ltable\}$ yet even less significant and far from that of $glm\{stats\}$. Another culprit is variable estimates, p values in particular, due to different run-to-run chains.

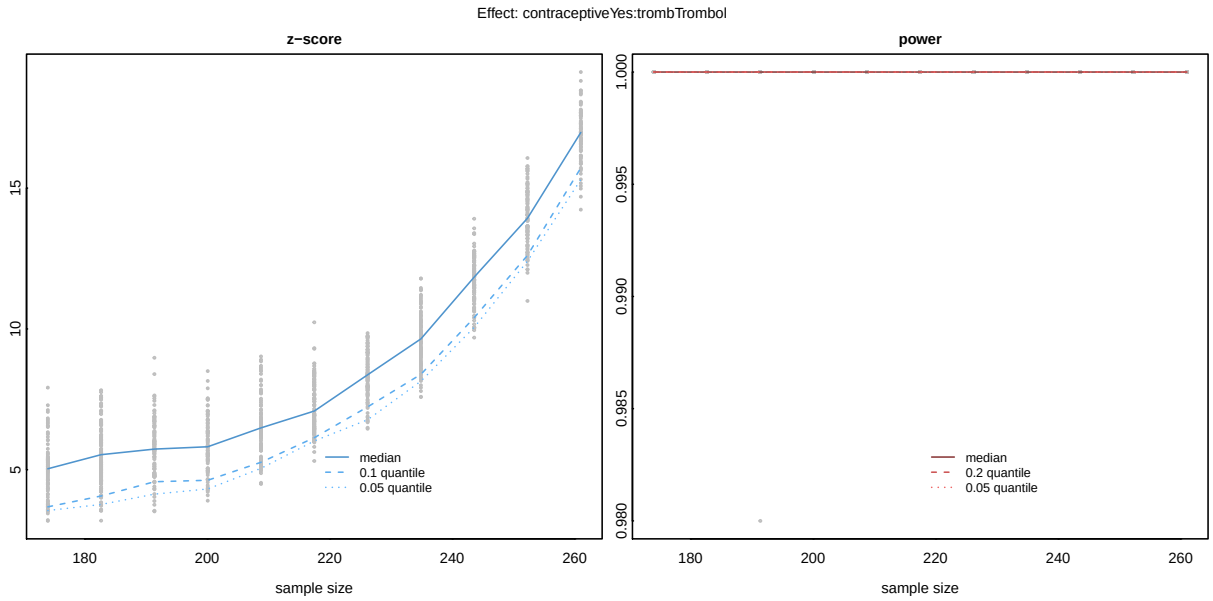
```
data(tdata, package="ltable")
require(MCMCglmm)
options(digits=3, width=80)
mcmcres<-MCMCglmm(Counts~smoker +contraceptive +tromb +  contraceptive*tromb,
                  family = "poisson",data = tdata, verbose = FALSE,pl = TRUE)
summary(mcmcres)$solutions
```

```
                            post.mean l-95% CI u-95% CI eff.samp pMCMC
(Intercept)                     4.244    3.002   5.0732      680 0.004
smokerYes                      -0.961   -1.843   0.0429      764 0.056
contraceptiveYes               -2.355   -3.891  -1.1846      420 0.008
trombTrombol                   -1.160   -2.401   0.2177     1000 0.070
contraceptiveYes:trombTrombol   2.215    0.647   4.4134      454 0.034
```

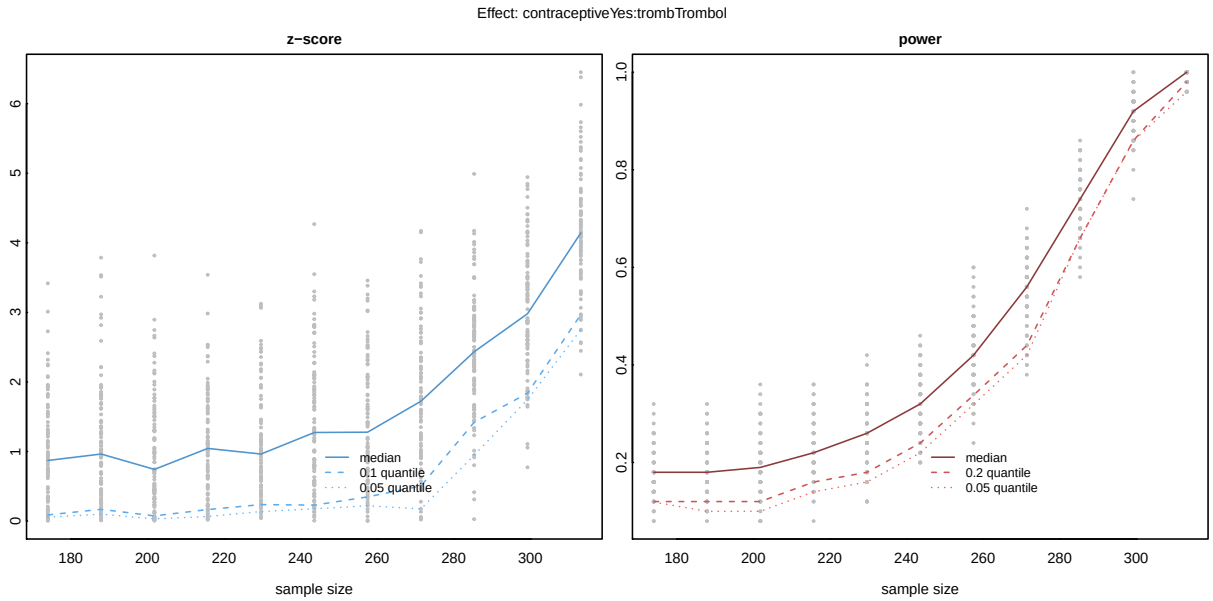
Bearing on power analysis

Let's conduct power analysis with $glm\{stats\}$ estimates. I used *glm* directly for a purpose and built package *RPower* just to illustrate a point. *RPower* uses the same syntaxes with *ltable* function *PowerPoisson*, but has additional logical arg *correction*. Value *FALSE* manages to conduct power analysis without modification, while value *TRUE* assures modification of error terms by multiplication by square root of chisq/dof.

```
require(RPower)
trombglm<-PowerPoissonGLM(Counts~smoker +contraceptive +tromb +
                          contraceptive*tromb, scale_max=1.5,
                          effect="contraceptive*tromb",
                          data=tdata, correction=FALSE)
plot(trombglm, st=3)
```




```
trombglmc1.8<-PowerPoissonGLM(Counts~smoker +contraceptive +tromb +
                              contraceptive*tromb, scale_max=1.8,
                              effect="contraceptive*tromb",
                              data=tdata, correction=TRUE)
plot(trombglmc1.8, st=3)
```



The invalidity of $glm\{stats\}$ estimates based power analysis is apparent. It is tremendously biased toward small sample sizes solutions. On the other hand, corrected error terms procure too large sample sizes solutions. Both solutions are biased to extremes.

In the light of findings I checked validity of power solutions suggested by other R packages, *pwr* and *lmSupport*, namely functions $pwr.f2.test\{pwr\}$ and $modelPower\{lmSupport\}$.

```
glmres<-glm(Counts~smoker +contraceptive +
              tromb + contraceptive*tromb,
            family="poisson", data=tdata)
an<-anova(glmres)
an[2]
```

```
                    Deviance
NULL
smoker                  42.3
contraceptive           63.8
tromb                   19.7
contraceptive:tromb     29.5
```

```
require(lmSupport)
modelEffectSizes(glmres)
```

```
glm(formula = Counts ~ smoker + contraceptive + tromb + contraceptive *
    tromb, family = "poisson", data = tdata)

Coefficients
               SSR df pEta-sqr dR-sqr
smoker        42.3  1    0.792     NA
contraceptive 92.7  1    0.893 0.0192
tromb         41.8  1    0.790 0.0086

Sum of squared errors (SSE): 11.1
Sum of squared total  (SST): 4837.5
```

```
modelPower(pc=4, pa=5, N=174, alpha=0.05,
           peta2=0.7)
```

```
Results from Power Analysis

pEta2 = 0.700
pa =     5
pc =     4
alpha = 0.050

N = 174.000
power = 1.000
```

ANOVA is used to check the lower bound of *contraceptive*tromb* effect that is of effect *tromb*. As function $modelEffectSizes\{lmSupport\}$ omits interactions effects from consideration, to be on the safe side I put 0.7 as value of partial effect size (*pEta-sqr*) of effect *tromb*. *pc* is number of parameters in the model, i.e., intercept + all parameters excluding the effect of interest. This is the numerator *df* of the *F* test for the effect. *pa* is the same but with the effect of interest included. As *contraceptive*tromb* effect is not compound of several contrasts but of one, the difference is 1. I checked the power for original sample size of 174.
Result follows the biased to small samples pattern, indicating power of 1.0.

BE ON THE SAFE SIDE