Residual odds ratios from 2x2xk tables

Rodolphe Priam

Residual odds ratios from 2x2xk tables*

R. Priam†

October 26, 2021

Abstract

In health statistics, it exists several indicators in order to find which factors are more related to a health issue such as a disease. Herein, it is proposed a data analysis method for the k related 2x2 contingency tables in order to help decide the importance of the modalities and binary variables involved. This leads to define from the counts of the contingency tables an indicator named residual odds ratio. This indicator allows a visual interpretation of the usual odds ratio computed via a logistic regression or quotient in the literature. Another indicator with an additive definition is also retrieved by linearity of the rotation while a transformation is involved for the non linear indicator.

Keywords: odds ratio, contingency table, principal component analysis, measure of interaction

1 Introduction

Medical research often define contingency tables from variables called exposures and a variable for the disease, such as indicators are obtained for judging the importance of the factors with these tables or with the observed data. From these tables, odds ratios (OR) [Cornfield, 1951] appear in many analysis of the literature which report diseases as a function of factors. The usual approach is to have a full table with most of the variables of the study plus the values of the odds ratios.

Health tables

In general, a health study defines a set of p variables which are measured on n persons which belong to two groups. The variables are the exposure ones while the groups are the case for the sick persons and the control for the unsick ones. The idea is that some factors induce the health of a person such as it may be sick if the factor is present and unsick otherwise. Examples are the gender or the age of someone which are of main interest as they are the first variables to check as they describe the observed sample in an intuitive way. Other variables would be the working activity, the salary, the habitation, the marital status or the car ownership for instance. To have a simple notation, it is supposed that all the variables are binary, without loss of generality because categorical variables can be broken into binary ones and continuous ones can be transformed into categorical ones. It is defined the contingency table:

---

*A nearly similar version of this document was sent to review since 05/2021.

†rpriam@gmail.com

	Exposure=1	Exposure=0	
Sick = 1	$a = n_{11}$	$b = n_{10}$	$n_{1.}$
Sick = 0	$c = n_{01}$	$d = n_{00}$	$n_{0.}$
	$n_{.1}$	$n_{.0}$	$n_{..}$

According to the literature, the chi-squared of the 2x2 table is written as follows:

$$\begin{aligned} t_{\chi^2} &= \frac{n_{..}(n_{11}n_{00}-n_{10}n_{01})^2}{n_{1.}n_{0.}n_{.1}n_{.0}} \\ &\propto (1-OR)^2 \end{aligned}$$

This statistics does not look like proportional [Beh et al., 2013] to OR (see next paragraph), such that the projection from the test χ^2 is more related to the independence of the contingency table.

Numerical indicators (OR and RR)

The odds ratio, relative risk [Schechtman, 2002] and other indicators are defined from the ratio of probabilities obtained from the contingency table, they are written:

$$\begin{aligned} OR &= \frac{n_{11}n_{00}}{n_{01}n_{10}} \\ RR &= \frac{\frac{n_{11}}{n_{11}+n_{10}}}{\frac{n_{01}}{n_{01}+n_{00}}} \\ AS &= \frac{n_{11}}{n_{11}+n_{10}} - \frac{n_{10}}{n_{11}+n_{10}} - \frac{n_{01}}{n_{01}+n_{00}} + \frac{n_{00}}{n_{01}+n_{00}} \end{aligned}$$

Expressions for the standard error are available from the literature. Similarly, the chi-squared of the contingency table is written with the counts. The indicator AS is extensively explained in [VanderWeele and Knol, 2014] where it is named as a measure of interaction on the additive scale [Rothman, 1976].

A remark is about the interpretation of the odds ratio which is not always clear. It may be written as follows,

$$\frac{n_{11}}{n_{01}} = OR \times \frac{n_{10}}{n_{00}} .$$

When the odds ratios are computed with the available data, this induces that the two ratios are proportional, with the usual claim that if it is superior or inferior to one, the corresponding factor is or is not relevant. More precisely one would say that the factor is involved in the disease if the estimation interval of the odds ratio does not contain one. The power of such statistical decision is not discussed herein: only the geometry of this indicator is explained as it is different from correspondence analysis [Greenacre and Hastie, 1987] for count data. Note that other indicators have been suggested in the literature such as [Mosteller, 1968, Beh et al., 2013] but out of the scope.

Purpose and plan

Herein, it is presented a new visual method and its related statistics. The plan of the paper is as follows. Section 1 is for the introduction, section 2 for the method and the indicator(s), section 3 the numerical experiments while section 4 concludes with perspectives.

2 Projection from the counts

In this section, a new data analysis allows to visualize the indicators on a projection map for exploratory analysis.

Matrix to reduce

For each exposure variable, a counting table N_ℓ is contructed such that the whole set of tables is written in an unique matrix:

$$N = \begin{pmatrix} n_{11}^{(1)} & n_{01}^{(1)} & n_{01}^{(1)} & n_{00}^{(1)} \\ n_{11}^{(2)} & n_{01}^{(2)} & n_{01}^{(2)} & n_{00}^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ n_{11}^{(k)} & n_{01}^{(k)} & n_{01}^{(k)} & n_{00}^{(k)} \end{pmatrix}$$

The problem with this table is that it is not a contingency table because the row margins are not completly meaningful, thus a transformation is proposed. The matrix after transformations is as follows:

$$X = \begin{pmatrix} \frac{n_{11}^{(1)}}{n_{11}^{(1)}+n_{10}^{(1)}} & \frac{n_{10}^{(1)}}{n_{11}^{(1)}+n_{10}^{(1)}} & \frac{n_{01}^{(1)}}{n_{01}^{(1)}+n_{00}^{(1)}} & \frac{n_{00}^{(1)}}{n_{01}^{(1)}+n_{00}^{(1)}} \\ \frac{n_{11}^{(2)}}{n_{11}^{(2)}+n_{10}^{(2)}} & \frac{n_{10}^{(2)}}{n_{11}^{(2)}+n_{10}^{(2)}} & \frac{n_{01}^{(2)}}{n_{01}^{(2)}+n_{00}^{(2)}} & \frac{n_{00}^{(2)}}{n_{01}^{(2)}+n_{00}^{(2)}} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{n_{11}^{(k)}}{n_{11}^{(k)}+n_{10}^{(k)}} & \frac{n_{10}^{(k)}}{n_{11}^{(k)}+n_{10}^{(k)}} & \frac{n_{01}^{(k)}}{n_{01}^{(k)}+n_{00}^{(k)}} & \frac{n_{00}^{(k)}}{n_{01}^{(k)}+n_{00}^{(k)}} \end{pmatrix}$$

This matrix is such as the sums of the two first columns and the two last ones are a column vector with all component equal to one. As the space is four-dimensional, the two first dimensions allows to project all the inertia from principal component analysis (pca) [Lebart et al., 1984]. Hence, for this matrix, pca is considered for the reduction of the matrix X next paragraph. This allows the visualization of the rows which are from binary variables herein or modalities of polytomous variables. The approach is different from the usual way to use a (multiple) correspondence analysis [Lebart et al., 1984] in the literature.

Reduction method

For pca, each column $X = (x_{ij})$ is first normalized, with zero means and unit variances. The new columns are in a new matrix $Y = (y_{ij})$ such that,

$$y_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} .$$

Let denote Z a matrix with k rows and four columns,

$$Z = [Z_1|Z_2|0_k|0_k] ,$$

where $Z_s = (z_{1s}, z_{2s}, \cdots, z_{ks})^T$, for the coordinates of the non zero orthogonal projection. With an eigenvalue decomposition of the matrix $C = USU^T$, it is deduced the principal coordinates,

$$Z = YU .$$

A solution in closed-form for U leads to an expression in full closed-form for Z as explained next paragraphs.

Property 1: two null eigenvalues

For the reduction method, two eigenvalues are trivial and non informative. Thus only two components are obtained for the rows and the columns. The proof is as follows. When c is a correlation between the first and the third column for instance, the correlation matrix for Y is written as,

$$C = \begin{pmatrix} 1 & -1 & c & -c \\ -1 & 1 & -c & c \\ c & -c & 1 & -1 \\ -c & c & -1 & 1 \end{pmatrix}.$$

It is looked after the following values for an eigen decomposition, $C = USU^T$, which is computed as,

$$U = \begin{pmatrix} -0.5 & 0.5 & b & 0 \\ 0.5 & -0.5 & b & 0 \\ -0.5 & -0.5 & 0 & b \\ 0.5 & 0.5 & 0 & b \end{pmatrix}.$$

It is checked that this leads to eigenvalues by verifying that $Cu_\ell = \lambda_\ell u_\ell$, which concludes to the eigenvalues written as:

$$\begin{aligned} \lambda_1 &= 2(1+c) \\ \lambda_2 &= 2(1-c) \\ \lambda_3 &= 0 \\ \lambda_4 &= 0\,. \end{aligned}$$

It is also checked that the vectors u_1 and u_2 are normed to the unity and orthogonal, it is deduced b with $u_3^T u_3 = 1$ or $u_4^T u_4 = 1$ such that,

$$b = \frac{\sqrt{2}}{2}\,.$$

Another solution would be to write a determinant and deduce the eigenvalues in a more direct approach. This result is different to correspondence analysis [Lebart et al., 1984] with only one trivial eivenvalue and equal to one. This leads to the following coordinates for the matrix X from the second factor, when the normalisation is with the mean μ and standard deviation σ and same indexing than for the columns,

$$\begin{aligned} z_{\ell 2} &= \tfrac{0.5}{\sigma_{11}}\left(\tfrac{n_{11}^{(\ell)}}{n_{11}^{(\ell)}+n_{10}^{(\ell)}} - \mu_{11}\right) \\ &- \tfrac{0.5}{\sigma_{10}}\left(\tfrac{n_{10}^{(\ell)}}{n_{11}^{(\ell)}+n_{10}^{(\ell)}} - \mu_{10}\right) \\ &- \tfrac{0.5}{\sigma_{01}}\left(\tfrac{n_{01}^{(\ell)}}{n_{01}^{(\ell)}+n_{00}^{(\ell)}} - \mu_{01}\right) \\ &+ \tfrac{0.5}{\sigma_{00}}\left(\tfrac{n_{00}^{(\ell)}}{n_{01}^{(\ell)}+n_{00}^{(\ell)}} - \mu_{00}\right). \end{aligned}$$

Instead of comparing the counts with a difference of their logarithm as in the log odd ratios, the approach with a pca compares the counts by differences with a normalization into proportions but without any further transformation except the standardization.

Property 2: link with odds ratio

The previous definition of a pca for count data from the health matrices leads to a rotation of the column space after normalization and standardization. During the preliminary numerical experiments with the projection method, it has been observed that the second component of the projection is closely related to the odds ratios. For the ℓ^{th} binary variable where $z_{\ell 2}$ is the corresponding ℓ^{th} component in the vector Z_2, this leads to define the residual odds ratio as follows,

$$ROR_\ell = \alpha \, exp(z_{\ell 2}) \, .$$

Here α is the median of the ratios $(OR_\ell / exp(z_{\ell 2}))_\ell$ in order to correct for the observed bias from the exponential transformation. This eventual multiplicative correction is possible in order to get a closer indicator to the usual OR_ℓ, the odds ratio OR for the same variable. The considered quantity is a residual projection from a principal component analysis with exponential transformation because the first axis of the projection is the most contributing to the whole variance. Such transformation is also required in logistic regression [Hailpern and Visintainer, 2003] but this is the original data which are involved instead of directly the 2x2 tables.

A justification of the reason why the two statistics are related may be as follows. Let rewrite the proportions,

$$p_{uv} \propto n_{uv} \text{ with } u, v \in \{0, 1\} \, .$$

A very rough approximation is with ϵ_{uv} small such as, $p_{uv} = 0.5 + \epsilon_{uv}$, and:

$$\log OR_\ell \approx \tfrac{1}{0.5} (\epsilon_{11} + \epsilon_{00} - \epsilon_{01} - \epsilon_{10}) \, .$$

Similarly,

$$\begin{aligned} z_{\ell 2} &= \tfrac{0.5}{\sigma_{11}} (0.5 - \mu_{11}) - \tfrac{0.5}{\sigma_{10}} (0.5 - \mu_{10}) \\ &- \tfrac{0.5}{\sigma_{01}} (0.5 - \mu_{01}) + \tfrac{0.5}{\sigma_{00}} (0.5 - \mu_{00}) \\ &+ \tfrac{0.5}{\sigma_{11}} \epsilon_{11} + \tfrac{0.5}{\sigma_{00}} \epsilon_{00} - \tfrac{0.5}{\sigma_{10}} \epsilon_{10} - \tfrac{0.5}{\sigma_{01}} \epsilon_{01} \\ &\approx \tfrac{0.5}{\sigma} (\epsilon_{11} + \epsilon_{00} - \epsilon_{01} - \epsilon_{10}) \, . \end{aligned}$$

This induces that for μ_{uv} and σ_{uv} enough constant to a same value $\mu = 0.5$ and σ respectively, the correlation between the two statistics may become higher. The definition for the residual odds ratio is empirically justified in the next experimental section by checking the correlation from k pairs of indicators instead of just one pair.

Property 3: graphical interpretation

In the expression for the second component, it is recognized also the indicator AS with weights and additive terms. When the matrix N is not scaled, this is exactly this indicator which is retrieved, while the exponentiation suggests a link with the indicator OR. The main remark about the proposed data analysis approach may be that just projecting the first and the third columns of N (after transformation) leads to the same result, which may induce that we just need to show the scatter plot from the principal component analysis of these data,

$$\left\{ \left(\frac{n_{11}^{(\ell)}}{n_{11}^{(\ell)} + n_{10}^{(\ell)}}, \frac{n_{01}^{(\ell)}}{n_{01}^{(\ell)} + n_{00}^{(\ell)}} \right) ; 1 \leq \ell \leq k \right\} \, .$$

The usual indicator is read graphically as the vertical coordinate after rotation while the proposed one needs a transformation. This is different to the usual approach where one may compute ratios for the helping the interpretation of the causes for a disease.

3 Numerical results

In this section the usual odds ratios are compared with the proposed one called residual odds ratio, for several datasets. First a dataset is studied in order to visualize the indicators before the comparisons.

Comparisons between OR and ROR

In this current paragraph, it is considered seven datasets with n rows and d variables. The correlations between the usual odds ratios and the residual ones are computed in order to check the similarity between the two indicators. The resulting correlations are in Table 1, when removing the eventual variables with non available usual odds ratios (for only one dataset). The datasets are described at the appendix section.

	n	d	k	α	Pe	Sp	Rb	c
D1	4406	19	24	1.010	0.889	0.792	0.850	0.97
D2	929	32	14	1.003	0.938	0.952	0.899	0.87
D3	3154	15	16	1.003	0.232	0.826	0.998	0.86
D4	139	10	17	0.944	0.954	0.978	0.983	0.12
D5	100	15	19	1.045	0.973	0.974	0.995	0.86
D6	189	10	19	1.053	0.589	0.579	0.850	0.88
D7	532	16	28	1.040	0.883	0.945	0.977	0.73

Table 1: Three different statistical correlations (Pe for Pearson, Sp for Spearman and Rb for robust) between the usual and the proposed indicator and the correlation c.

For these same datasets, the values for the standard deviations and means from the standardization, and a robust linear regression between $\log OR$ and $z_{\ell 2}$ with the slope β and the intercept γ plus a star if their p-value for testing is smaller than 0.05, are given in Table 3 for a further comparison.

	μ_{11}	μ_{00}	σ_{11}	σ_{00}	γ	β
D1	0.417	0.583	0.266	0.293	0.018	0.451*
D2	0.357	0.643	0.287	0.285	0.016	0.483*
D3	0.438	0.562	0.253	0.251	0.011	0.934*
D4	0.471	0.529	0.242	0.196	-0.023	1.031*
D5	0.368	0.632	0.180	0.172	0.031	1.218*
D6	0.421	0.579	0.203	0.274	0.011	0.496*
D7	0.250	0.750	0.242	0.208	-0.013	0.645*

Table 2: Complements for the seven datasets.

Note that the continuous variables were discretized into categorical ones with three equidistributed modalities, while the variable bmi was created if not available and categorized into its usual categorical version. Only a few existing categorical variables had some modalities aggregated because of low frequencies for mostly apagar. The recoding were nearly arbitrary while keeping meaningful for illustration purposes and may be improved for a better interpretation, and with a more relevant modeling such as per class which is left as a perspective.

The Tables 1, 2 justify empirically the proposed approach of an indicator closely related to the odds ratio via a principal component analysis. It may be noticed that the outliers from high values of the usual odds ratio in the computation of the correlation may explain often most of the differences between the two indicators for some datasets. The linear trend has been checked visually with the scatterplots, a test with a regression is given in Table 2 with the result of a significant proportionality for the seven datasets. The value of α is often almost equal to one while the correlations near one too, hence the two indicators have very similar trend at the middle of the distribution. According to the results for the dataset wcgs and the Pearson correlation, it may happen a large difference which would need more investigation in order to decide which indicator is to be preferred.

Visualization from ROR

From the dataset D1 [Achim et al., 2008, Deb and Trivedi, 1997] it is selected a variable for the health issue named hospital (see also the appendix) which is binarized into a variable hosp2groups with a value at one for at least one visit at the hospital. The corresponding matrix N which was obtained after discretizing the variable age into three categories is in Table 1 after normalization per couples of columns with also the statistical indicators.

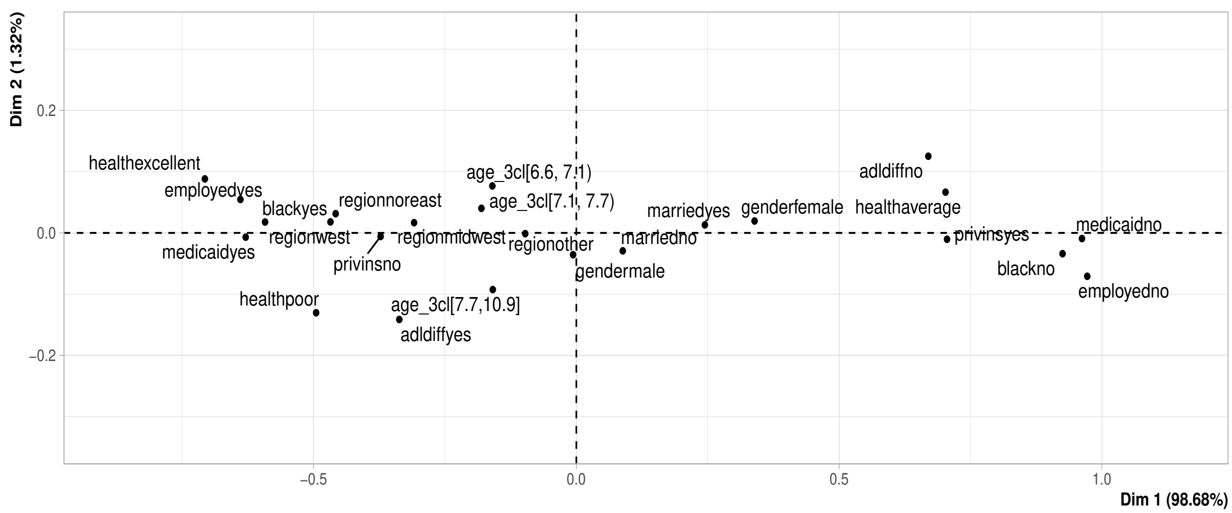

Figure 1: Projection for the dataset DebTrivedi.

The projection from principal component analysis for this dataset is as follows in Figure 1 from FactoMiner [Lê et al., 2008]. It is observed all the modalities in a same map without loss because the two last eigen vectors are zero. The correlation circle is not given but as expected it is obtained two separated groups of two arrows, one for the cases and one for the controls, by symmetry of the paiwise columns. The corresponding correlation matrix is,

$$C = \begin{pmatrix} 1.00 & -1.00 & 0.97 & -0.97 \\ -1.00 & 1.00 & -0.97 & 0.97 \\ 0.97 & -0.97 & 1.00 & -1.00 \\ -0.97 & 0.97 & -1.00 & 1.00 \end{pmatrix}.$$

	$\frac{a}{a+b}$	$\frac{b}{a+b}$	$\frac{c}{c+d}$	$\frac{d}{c+d}$	OR_ℓ	ROR_ℓ	$z_{\ell 2}$	$\mathrm{crt}_{\ell 2}$
healthaverage	0.72	0.28	0.82	0.18	0.567	0.789	-0.247	005
healthpoor	0.25	0.75	0.10	0.90	3.154	1.633	0.480	018
healthexcellent	0.03	0.97	0.09	0.91	0.371	0.734	-0.320	008
adldiffno	0.67	0.33	0.83	0.17	0.424	0.637	-0.462	017
adldiffyes	0.33	0.67	0.17	0.83	2.359	1.698	0.520	021
regionother	0.37	0.63	0.37	0.63	1.026	1.015	0.005	000
regionmidwest	0.26	0.74	0.26	0.74	0.992	0.952	-0.060	000
regionnoreast	0.18	0.82	0.19	0.81	0.942	0.902	-0.113	001
regionwest	0.18	0.82	0.18	0.82	1.033	0.948	-0.064	000
blackyes	0.13	0.87	0.11	0.89	1.110	0.949	-0.062	000
blackno	0.87	0.13	0.89	0.11	0.901	1.139	0.120	001
gendermale	0.43	0.57	0.40	0.60	1.159	1.151	0.131	001
genderfemale	0.57	0.43	0.60	0.40	0.863	0.939	-0.073	000
marriedyes	0.53	0.47	0.55	0.45	0.904	0.962	-0.049	000
marriedno	0.47	0.53	0.45	0.55	1.106	1.124	0.107	001
employedyes	0.08	0.92	0.11	0.89	0.762	0.829	-0.197	003
employedno	0.92	0.08	0.89	0.11	1.312	1.303	0.255	005
privinsyes	0.76	0.24	0.78	0.22	0.874	1.046	0.035	000
privinsno	0.24	0.76	0.22	0.78	1.144	1.034	0.023	000
medicaidno	0.88	0.12	0.92	0.08	0.663	1.040	0.029	000
medicaidyes	0.12	0.88	0.08	0.92	1.509	1.039	0.028	000
age_3cl[6.6, 7.1)	0.30	0.70	0.37	0.63	0.734	0.763	-0.280	006
age_3cl[7.1, 7.7)	0.31	0.69	0.34	0.66	0.867	0.873	-0.146	002
age_3cl[7.7,10.9]	0.39	0.61	0.29	0.71	1.567	1.419	0.340	009

Table 3: The matrices 2x2 in a flat version after normalization, and the indicators OR_ℓ and ROR_ℓ plus the vector Z_2 and the contribution $\mathrm{ctr}_{\ell 2} = 100\, z_{\ell 2}^2 / \sum_{\ell'} (z_{\ell' 2}^2)$ to the axis, for DebTrivedi.

When comparing the values for the odds ratios and the residual odds ratios for this data set, it has been observed that the proposed indicator is not far to the usual one with at most a few 0.1 as the difference, while being smaller for the largest values in the case of this dataset. It is worth to notice that other indicators such as the coordinate or the signed (with the sign of the coordinate) contribution [Lebart et al., 1984] which is the squared of the principal coordinate are less correlated to the usual indicator. The transformation is not the same and under one the square has not the same behavior than the exponential function.

4 Discussion and perspectives

Herein it is proposed a geometrical interpretation of the odds ratio with an indicator named residual odds ratio for disease-exposure matrices in order to compare several factors. The residual variance involved in the corresponding component depends on the correlation between the columns of the counts for computing the odds ratios, thus from the choice of the set of variables. According to the available numerical experiments, the usual also called common odds ratio can be understood as the transformation of the residual projection from a principal component analysis when the main component is removed, this is a new result to our knowledge. This allows new visual representations such as in Figure 1 with a seriation of the modalities which may be more informative than the usual plots for odds ratios. The method is implemented in an available[1] r package *dataepi* which computes from a dataset the tables for the odds ratios, the risk ratios and the residual odds ratios proposed

[1] https://github.com/rpriam/dataepi

herein. The r package is fully automatic for generating a tex report by construct also the odds ratios from the logistic regressions, the confidence intervals and tables from several statistical tests.

A visualization of the matrices of counts for comparing factors is also proposed via a pca of a transformed matrix, with an exact two dimensional reduced projection. According to the numerical experiments, the resulting indicator may be able to be less prone to very large values in some cases. It is also always available even when a count out four ones is zero in some matrices or collinearities exist among the variables. These two properties are often required in the literature in order to improve the empirical interpretations. Perspectives are an analytical expression for the correlations from the k pairs of indicators and alternative approaches for a multivariate modeling of the k matrices 2×2 of counts such as via tensors for instance.

Appendix

Let denote the dependent variable y_ℓ and the explicative variables x_ℓ. The datasets come from *cran.r-project.org* with several r packages,
- DebTrivedi (D1) from *pscl* with *hospital* binarized for y_ℓ and *health*, $adldiff$, *region*, *black*, *gender*, *married*, *employed*, *privins*, *medicaid*, *age* for x_ℓ,
- colon_s (D2) from *finalfit* with *mort_5yr* for $y\ell$ and $age.factor$, $sex.factor$, $obstruct.factor$, $perfor.factor$, *hospital* for x_ℓ,
- wcgs (D3) from *finalfit* with *coronheart* for y_ℓ and *personality_2L*, *smoking*, *arcus*, *bmi*, *age*, *hypertension*, *cholesterol* for x_ℓ,
- breastfeed (D4) from *mpath* with *breast* binarized for y_ℓ and *partner*, *pregnancy*, $howfed$, $howfedfr$, *smokenow*, $smokebf$, *ethnic*, *age* for x_ℓ,
- Delivery (D5) from *lazyWeave* with *delivery_type* for y_ℓ and *child.sex*, *grava*, *para*, $apgar1$, $apgar5$, *ga.weeks*, *ga.days* for x_ℓ,
- birthwt (D6) from *MASS* with *low* for y_ℓ and *age*, *lwt*, *race*, *smoke*, *ptl*, *ht*, *ui*, ftv for x_ℓ,
- Pima.tr and Pima.te (D7) appended from *MASS* with *diabet* for y_ℓ and *npreg*, *bp*, *bmigroups*, *glu*, *skin*, *ped*, *age* for x_ℓ.

References

[Achim et al., 2008] Achim, Z., Kleiber, C., and Jackman, S. (2008). Regression models for count data in r. *Journal of Statistical Software*, 27:1–25.

[Beh et al., 2013] Beh, E. J., Tran, D., and Hudson, I. L. (2013). A reformulation of the aggregate association index using the odds ratio. *Computational Statistics & Data Analysis*, 68:52–65.

[Cornfield, 1951] Cornfield, J. (1951). A Method of Estimating Comparative Rates from Clinical Data. Applications to Cancer of the Lung, Breast, and Cervix. *JNCI: Journal of the National Cancer Institute*, 11(6):1269–1275.

[Deb and Trivedi, 1997] Deb, P. and Trivedi, P. (1997). Demand for medical care by the elderly: A finite mixture approach. *Journal of Applied Econometrics*, 12:313–36.

[Greenacre and Hastie, 1987] Greenacre, M. and Hastie, T. (1987). The geometric interpretation of correspondence analysis. *Journal of the American Statistical Association*, 82(398):437–447.

[Hailpern and Visintainer, 2003] Hailpern, S. M. and Visintainer, P. F. (2003). Odds ratios and logistic regression: Further examples of their use and interpretation. *The Stata Journal*, 3(3):213–225.

[Lê et al., 2008] Lê, S., Josse, J., and Husson, F. (2008). FactoMineR: A package for multivariate analysis. *Journal of Statistical Software*, 25(1):1–18.

[Lebart et al., 1984] Lebart, L., Morineau, A., and Warwick, K. (1984). *Multivariate Descriptive Statistical Analysis*. J. Wiley.

[Mosteller, 1968] Mosteller, F. (1968). Association and estimation in contingency tables. *Journal of the American Statistical Association*, 63(321):1–28.

[Rothman, 1976] Rothman, K. J. (1976). Causes. *American Journal of Epidemiology*, 104(6):587–592.

[Schechtman, 2002] Schechtman, E. (2002). Odds ratio, relative risk, absolute risk reduction, and the number needed to treat?which of these should we use? *Value in Health*, 5(5):431–436.

[VanderWeele and Knol, 2014] VanderWeele, T. and Knol, M. (2014). A tutorial on interaction. *Epidemiologic Methods*, 3.

Package 'dataepi'

June 2, 2021

Version 0.1

Date 2021-04-17

Title An R Package For Health Dataset

Description A set of functions for automatically generating a draft report from a health dataset in tabular form with a binary variable for the disease and any type of variable (categorical or binary, continuous or discrete) for the exposures which are the factors considered in the study. The package should be used with cautious by checking while completing the obtained output.

Author R. Priam <rpriam@gmail.com>

Maintainer R. Priam <rpriam@gmail.com>

Imports MASS, car, plyr, pracma, pwr, officer, xtable

License GPL-3

RoxygenNote 7.1.1

NeedsCompilation no

R topics documented:

`data_prepare` *A function for preparing the data.frame before the analysis*

Description

This function takes as input a data.frame and the names of the variables in order check the variables and their r types.

Usage

```
data_prepare(
  A,
  var_y = NULL,
  vars_cont = NULL,
  vars_disc = NULL,
  vars_int = NULL,
  var_id = NULL
)
```

Arguments

`A`	The data.frame with the variables to test.
`var_y`	The variable for the disease yes/not.
`vars_cont`	The names of the variables with continuous values.
`vars_disc`	The names of the variables with categorical values.
`vars_int`	The names of the variables with integer (ordered) values.
`var_id`	The name of the variable for the unique identifier per row.

Value

A list with the following entries.

A The data.frame from the dataset after checking and updating.

var_disc_from_cont The names of the discretized variables from continuous ones (not implemented).

vars_disc The vectors of names received from the input parameters.

vars_cont The vectors of names received from the input parameters.

var_y The same name received from the input parameters.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi[,c("age","ofp","gender","region","health")]
A$id      <- 1:nrow(A)
A$hospbin <- DebTrivedi$hosp>0
vars_cont <- c("age","ofp")
vars_disc <- c("gender","region","health")
var_id    <- "id"
var_y     <- "hospbin"
fp = data_prepare(A,var_y,vars_cont,vars_disc,var_id)
print(head(fp$A))
```

`data_rename` *A function for renaming the modalities of categorical variables*

Description

This function take as input a data.frame and the names of the categorical variables in order to rename the modalities with strings.

Usage

```
data_rename(A, vars_disc_to_recode)
```

Arguments

`A` The data.frame with the variables to rename.

`vars_disc_to_recode`
The vector with the string names of the variables to rename.

Value

A A data.frame for the dataset after renaming the modalities.

dico A list with the correspondences between the old and new characters strings as modalities of the selected categorical variables.

Examples

```
data(DebTrivedi)
A <- DebTrivedi
A$id      <- 1:nrow(A)
A$hospbin <- as.integer(A$hosp>0)
var_id    = "id"
vars_cont = c("age","ofp","ofnp","opp","opnp","emer","numchron","hosp","school")
vars_disc = c("health","adldiff","region","black","gender","married","employed",
              "privins","medicaid")
vars_int  = NULL
```

```
var_y     = "hospbin" #binary 0/1
label_y   = "hospibin"
A <- data_prepare(A,var_y,vars_cont,vars_disc,var_id)$A
vars_disc_to_recode  <- c("health","gender","region")
resu_     <- data_rename(A, vars_disc_to_recode)
for (nv in vars_disc_to_recode) {
  resu_nv <-  data.frame(resu_$dico[[nv]])
  rownames(resu_nv) <- nv
  print(resu_nv)
}
```

DebTrivedi *Dataset of 4406 individuals aged 66 and over with 19 variables*

Description

Deb and Trivedi (1997) dataset of 4406 individuals, aged 66 and over, who are covered by Medicare, a public insurance program. Originally prepared for an R package accompanying Kleiber and Zeileis (2008) paper and also available as with Zeileis (2006) paper.

Usage

```
data(DebTrivedi)
```

Format

An object of class "data.frame".

Source

JSTAT (http://www.jstatsoft.org/v27/i08/paper)

References

A. Zeileis et al. (2008) Journal of Statistical Software 27(8):1-25.
Deb, P. and Trivedi, P. (1997). Demand for medical care by the elderly: A finite mixture approach. Journal of Applied Econometrics, 12:313-36.

Examples

```
data(DebTrivedi)
head(DebTrivedi)
```

`rep_compute` *A function for generating the tables for the report from a data.frame*

Description

This function returns tables from a variable for disease which is binary and a set of variables which are categorical for exposures. The descriptive statistics and statistical tests are computed and aggregated in several data.frames.

Usage

```
rep_compute(A, var_y, vars_x, vars_cont, vars_disc, vars_int, var_id)
```

Arguments

`A`	The data.frame for the analysis.
`var_y`	The variable for the disease yes/not.
`vars_x`	The variables for the table with glm for or and rr.
`vars_cont`	The name of the variables with continuous values.
`vars_disc`	The name of the variables with categorical values.
`vars_int`	The name of the variables with integer (ordered) values.
`var_id`	The name of the variable for the unique identifier per row.

Value

A list with the following entries.

Anew The new matrix A after pre-traitment with data_prepare().

desc_all The result from tab_contents().

desc_cont The result from tab_desc_cont().

desc_disc The result from tab_desc_disc().

desc_biv The result from tab_desc2class_cont().

test_tt The result from tab_tt2classes_cont().

test_anova The result from tab_ttanova_cont().

test_chi2 The result from tab_chi2all().

or The result from tab_all2x2() and stat_oddsratio().

rr The result from tab_all2x2() and stat_relativerisk().

gg The result from tab_glmorr().

fv The result from viz_all2x2().

args The variables at the call of the function.

Examples

```
## Not run:
data(DebTrivedi)
A=DebTrivedi
A$id=1:nrow(A)
A$hospbin  = as.integer(A$hosp>0)
var_id    = "id"
vars_cont = c("age","ofp","ofnp","opp","opnp","emer","numchron","hosp","school")
vars_disc = c("health","adldiff","region","black","gender","married","employed",
              "privins","medicaid")
vars_int  = NULL
var_y     = "hospbin" #binary 0/1
label_y   = "hospibin"
A0 = A
fp = data_prepare(A,var_y,vars_cont,vars_disc,var_id)#,discretize_=TRUE)
A  = fp$A
A$age_3cl = as.character(1*(A$age<7.1)+2*(A$age>=7.1&A$age<7.7)+3*(A$age>=7.7))
A$age_3cl[A$age_3cl=="1"]="[6.6, 7.1)"
A$age_3cl[A$age_3cl=="2"]="[7.1, 7.7)"
A$age_3cl[A$age_3cl=="3"]="[7.7,10.9]"
vars_disc = c(vars_disc,"age_3cl")
vars_x =c(vars_disc,fp$var_disc_from_cont)[c(1:9,10)] #only discrete variables relevant
au = dataepi::rep_compute(A, var_y, vars_x, vars_cont, vars_disc, vars_int, var_id)

## End(Not run)
```

---

rep_write — *A function for writing the full report from a data.frame*

Description

This function allows to generate a report from a binary variable for a disease, and a set of categorical variables for the exposures.

Usage

```
rep_write(
  fullpathfile = NULL,
  formatfile = "docx",
  A,
  var_y,
  vars_x,
  vars_cont,
  vars_disc,
  vars_int,
  var_id,
  list_supp = NULL,
  add_ORpca_ = FALSE
)
```

Arguments

`fullpathfile`	A character string with the full path for report saving.
`formatfile`	A character string for the format of the file with "doc","docx" or "rtf" for a word document and "tex" for a latex one.
`A`	The data.frame for the analysis.
`var_y`	The variable for the disease yes/not.
`vars_x`	The variables for the table with glm for or and rr.
`vars_cont`	The name of the variables with continuous values.
`vars_disc`	The name of the variables with categorical values.
`vars_int`	The name of the variables with integer (ordered) values.
`var_id`	The name of the variable for the unique identifier per row.
`list_supp`	A list with supplementary information for adding to the report, with optional entries. **where** A brief descriptive of the place(s) where the study took place. **who** A brief descriptive of the population targeted. **objective** A brief descriptive of the objectives and purposes. **disease** The name of the disease. **descriptive** A brief descriptive of the disease. **project** For the type of project, for instance "descriptive". **keywords** A list of key words corresponding to the study or analysis. **inex** A descriptive for the criteria for the inclusion and exclusion. **topics** A list with terms to classify the variables and each subset of variable names corresponding, for instance, "biological", with blood test results, "socio-demographic" with age, gender, etc.
`add_ORpca_`	A boolean variable for including or not including ORpca in the table with odds ratio and relative risk.

Value

A list with following entries.

au The resulting output from rep_compute().

fullpathfile The copy of the variable from the parameters with the same name for the full path for report saving.

note Message to user not null if the file exists already in order to avoid file loss.

Examples

```
## Not run:
data(DebTrivedi)
A=DebTrivedi
A$id=1:nrow(A)
A$hospbin  = as.integer(A$hosp>0)
var_id    = "id"
```

```
vars_cont = c("age","ofp","ofnp","opp","opnp","emer","numchron","hosp","school")
vars_disc = c("health","adldiff","region","black","gender","married","employed",
               "privins","medicaid")
vars_int  = NULL
var_y     = "hospbin" #binary 0/1
label_y   = "hospibin"
A0 = A
fp = data_prepare(A,var_y,vars_cont,vars_disc,var_id)#,discretize_=TRUE)
A  = fp$A
A$age_3cl = as.character(1*(A$age<7.1)+2*(A$age>=7.1&A$age<7.7)+3*(A$age>=7.7))
A$age_3cl[A$age_3cl=="1"]="[6.6, 7.1)"
A$age_3cl[A$age_3cl=="2"]="[7.1, 7.7)"
A$age_3cl[A$age_3cl=="3"]="[7.7,10.9]"
vars_disc = c(vars_disc,"age_3cl")
vars_x =c(vars_disc,fp$var_disc_from_cont)[c(1:9,10)] #only discrete variables relevant
#au = dataepi::rep_compute(A, var_y, vars_x, vars_cont, vars_disc, vars_int, var_id)
wr = dataepi::rep_write("./report_dataepi.docx","docx",
                          A, var_y, vars_x, vars_cont, vars_disc, vars_int, var_id)

## End(Not run)
```

`stat_oddsratio` *A function for computing the odds ratio from a table for an exposure*

Description

This function computes the odds ratio of a binary variable from the crosstable or contingency table between a disease and an exposure.

Usage

```
stat_oddsratio(X)
```

Arguments

`X`	The object of type table of size 2x2.

Value

A list with the following entries.

stat The odds ratio from the table X.

SE The standard-deviation of the odds ratio from the table X.

I.95left The left part of the confidence interval at 0.95%.

I.95right The right part of the confidence interval at 0.95%.

name The name of the statistics, "OR".

warning A boolean value for or for not having a=0 or b=0 or c=0 or d=0.

a The value a from the 2x2 input table.

b The value b from the 2x2 input table.

c The value c from the 2x2 input table.

d The value d from the 2x2 input table.

table2x2 The table from the input parameter.

Examples

```
X=matrix(c(2534,459,487,142),ncol=2,byrow=TRUE)
X=as.table(X)
colnames(X)<-c("0","1")
rownames(X)<-c("0","1")
print(X)
resu_=stat_oddsratio(X)
cat(resu_$name,"=",round(resu_$stat,2),
paste("(",round(resu_$stdstat,2),")",sep=""),
"\n")
```

`stat_relativerisk` *A function for computing the relative risk from a table for an exposure*

Description

This function computes the relative risk of a binary variable from the crosstable or contingency table between a disease and an exposure.

Usage

```
stat_relativerisk(X)
```

Arguments

`X`	The object of type table of size 2x2.

Value

stat The relative risk from the table X

SE The standard-deviation of the odds ratio from the table X.

I.95left The left part of the confidence interval at 0.95%.

I.95right The right part of the confidence interval at 0.95%.

name The name of the statistics, "OR".

warning A boolean value for or for not having a=0 or b=0 or c=0 or d=0.

a The value a from the 2x2 input table.

b The value b from the 2x2 input table.

c The value c from the 2x2 input table.

d The value d from the 2x2 input table.

table2x2 The table from the input parameter.

Examples

```
X=matrix(c(2534,459,487,142),ncol=2,byrow=TRUE)
X=as.table(X)
colnames(X)<-c("0","1")
rownames(X)<-c("0","1")
print(X)
resu_=stat_relativerisk(X)
cat(resu_$name,"=",round(resu_$stat,2),
paste("(",round(resu_$stdstat,2),")",sep=""),
"\n")
```

`tab_all2x2` *A function for generating a table with statistics such as odds ratios*

Description

This function returns the whole table with the statistics odd ratio, risk ratio of any other defined by the user from a set of binary variables (exposures) vs one binary variable (disease).

Usage

```
tab_all2x2(A, vars_disc, var_y, stat_f = NULL)
```

Arguments

`A`	The data.frame with the data.
`vars_disc`	The name of the variables with categorical values.
`var_y`	The name of the categorical variable.
`stat_f`	A function such as stat_oddsratio,stat_relativerisk,...

Value

A list with following entries.

tabstat A data.frame with eight columns: the variable name, the modality or level name, the contents of the 2x2 contingency table with A for a, B for b, C for c and D for D, and the corresponding statistics from the function stat_.... in the list of parameters, for instance OR and SE_OR for the odds ratios and their standard-deviations.

binarized_1_s The binarized variable for or not the variable equal to modality, in a list of lists.

binarized_0_s The binarized variable for or not the variable not equal to modality, in a list of lists.

tables2x2 The contingency tables for all the variables and all the modalities in the format of a list a lists, from the function stat_f in the parameters.

Examples

```
library(dataepi)
data(DebTrivedi)
A         <- DebTrivedi
A$hospbin <- as.integer(A$hosp>0)
vars_disc <- c("health","gender","region")
var_y     <- "hospbin"
resu_or   <- dataepi::tab_all2x2(A,vars_disc,var_y,dataepi::stat_oddsratio)
print(resu_or$tabstat)
```

`tab_chi2all` *A function for computing the chi2 test from several variables of a data.frame*

Description

This function take as input a data.frame and the names of the categorical variables in order to compute the tests and aggregates them in a data.frame. The resulting data.frame contains the chi-square tests for each variable in vars_disc minus the last which appears in the second column. There are two loops: k in (1;p-1) while l in (k+1;p) in order to compute the upper part to the diagonal.

Usage

```
tab_chi2all(A, vars_disc, pvalue_seuil_ = 0.015)
```

Arguments

`A`	The data.frame with the variables to test.
`vars_disc`	The vector with the names of the variables to test.
`pvalue_seuil_`	The maximal p-value for keeping the pairs of variables.

Value

A list with three entries

tabchi2 The data.frame with the chi2 tests by pairs of variables with the following columns

- **row** The variable from the rows of the contingency table.

- **col** The variable from the cols of the contingency table.
- **nbr** The number of modalities of the first variable.
- **nbc** The number of modalities of the second variable.
- **chi2** The statistics as computed from the chi2 test.
- **df** The number of free parameters in the chi2 test.
- **p.val** The p-value from the chi2 test.
- **mnij** The minimum coun in the cells of the table.
- **p.val.e** The p-value from the exact Fisher test.
- **pow** The power (if available) from the chi2 test.
- **nb** The total number of counts in the table.

pairs_no_pchi2 A data.frame with by rows the pairs of variables with no chi2 test available because of their corresponding contingency table.

pairs_large_pchi2 A data.frame with by rows the pairs of variables with no chi2 test available because their p-value is larger than the threshold.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
vars_disc <- c("health","gender","region")
resu_    <- tab_chi2all(A,vars_disc,0.05)
print(head(resu_$tabchi2))
```

---

tab_chi2oneall — *A function for computing the chi2 test from one against several variables*

Description

This function take as input a data.frame and the names of the categorical variables plus one additional variable in order to compute the tests and aggregates them in a data.frame, without filtering. The tests are computed with a loop on the whole set in vars_disc.

Usage

```
tab_chi2oneall(A, vars_disc, var_y)
```

Arguments

`A`	The data.frame with the variables to test.
`vars_disc`	The vector with the string names of the variables to test.
`var_y`	The string name of one variable.

Value

A data.frame with with the chi2 tests by pairs of variables with the following columns

row The variables from the input vector of variable names vars_disc.

col The variable with its name in the input variable var_y.

nbr The number of modalities of the first variable.

nbc The number of modalities of the second variable.

chi2 The statistics as computed from the chi2 test.

df The number of free parameters in the chi2 test.

p.val The p-value from the chi2 test.

mnij The minimum coun in the cells of the table.

p.val.e The p-value from the exact Fisher test.

pow The power (if available) from the chi2 test.

nb The total number of counts in the table.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
vars_disc <- c("health","gender","region")
var_y     <- "hosp"
resu_    <- tab_chi2oneall(A,vars_disc,var_y)
print(head(resu_))
print(head(resu_[resu_$p.val<=0.05,]))
```

`tab_contents` *A function for generating a very simple description of the variables in a data.frame*

Description

This function show the list of variables with the name of their classes of variable (numerical,integer,...) in r and the number of unique values for each variable.

Usage

```
tab_contents(D)
```

Arguments

`D`	The data.frame with the variables to describe.

Value

A data.frame with each row for a variable from the input data.frame and with the following columns.

variable The variable name from the column names of D.

r_class The type of the variable from the R langage.

nblevels The total number of unique observations.

nbobs The total number of non missing observations.

Warning: with class factor, it may exist empty levels not counted. The function may be considered only after the function data_prepare().

Examples

```
data(DebTrivedi)
A    <- DebTrivedi
A$id <- 1:nrow(A)
resu_<-tab_contents(A)
print(head(resu_))
```

---

tab_desc2class_cont *A function for generating a table with statistics of continuous variables*

Description

This function returns the whole table with the statistics number of observations, mean, standard-error, median, min, max from a set of continuous variables versus one categorical variable with two (or eventually more modalities).

Usage

```
tab_desc2class_cont(A, vars_cont, var_y, nbdigits = 2)
```

Arguments

`A`	The data.frame with the data.
`vars_cont`	The names of the continuous variables.
`var_y`	The name of the categorical variable.
`nbdigits`	The number of decimals to keep.

Value

A data.frame with each row for a continuous variable and with the following columns, where ¡modality¿ is one of the modalities of the variable whose name in written in var_y.

MEAN_¡modality¿ The mean of the continuous variable for the modality.

STD_¡modality¿ The standard-deviation of the variable for the modality.

MD_¡modality¿ The median of the continuous variable for the modality.

MIN_¡modality¿ The minimum of the continuous variable for the modality.

MAX_¡modality¿ The maximum of the continuous variable for the modality.

Nnotna__¡modality¿ The number of non missing observation for the modality.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
vars_cont <- c("age","ofp")
resu_    <- tab_desc2class_cont(A,vars_cont,"gender")
print(resu_)
```

`tab_desc_cont` *A function for describing the data.frame for the continuous variables*

Description

This function take as input a data.frame and the names of the continuous variables in order to print in a table the names, plus the median, mean, standard-deviation, minimum, maximum, number of NA, and the number of not NA.

Usage

```
tab_desc_cont(A, vars_cont, nbdigits = 2)
```

Arguments

`A`	The data.frame with the variables to test.
`vars_cont`	The names of the variables with continuous values.
`nbdigits`	The number of decimals to keep.

Value

A data.frame with each row for a continuous variable and with the following columns.

var The name of a variable from the vector of names vars_cont.

median The median of the variable.

mean The mean of the variable.

sd The standard-deviation of the variable.

min The minimum of the variable.

max The maximum of the variable.

nb_na The number of missing values of the variable.

nb The number of non missing values of the variable.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
vars_cont <- c("age","ofp")
resu_    <- tab_desc_cont(A,vars_cont)
print(resu_)
```

`tab_desc_disc` *A function for describing the data.frame for the categorical variables*

Description

This function take as input a data.frame and the names of the categorical variables in order to print in a table the names, frequencies per modalities, percentable per modalities, and the number of levels and number of NA values.

Usage

```
tab_desc_disc(A, vars_disc, nbdigits = 2)
```

Arguments

`A`	The data.frame with the variables to test.
`vars_disc`	The names of the variables with categorical values.
`nbdigits`	The number of decimals to keep.

Value

A data.frame with each row for a categorical variable and with the following columns.

var The name of a variable from the vector of names vars_disc.

nb_na The number of missing values.

nblevel The total number of unique observations.

nbperlevel The numbers of observations by modality.

propperlevel The proportions of observations by modality.

namelevel The corresponding modality names.

Examples

```
data(DebTrivedi)
A          <- DebTrivedi
vars_disc <- c("health","gender","region")
resu_     <- tab_desc_disc(A,vars_disc)
print(resu_)
```

---

tab_glmorr — *A function for computing the OR from the logistic regression*

Description

This function take the data.frame with all the variable and construct a new matrix with the odds ratio from a logistic regression, from the full model and from the reduced model after a selection by AIC.

Usage

```
tab_glmorr(A, vars_x, var_y)
```

Arguments

`A`	The data.frame for the analysis.
`vars_x`	The variables for the table with glm for or and rr.
`var_y`	The variable for the disease yes/not.

Value

A list with following entries.

frm The formula for the logistic regression for computing the odds ratio on a full model before selecting the variables.

tabl_coeff_full The resulting table of coefficients regression with all the variables.

tabl_coeff_small The resulting table of coefficients regression with the variable kept after a selection by AIC.

allcoeffs The two tables of coefficients side to side for comparison purpose.

allors The odds ratios from the two tables of coefficients side to side in allcoeffs.

fit_full The r object from the glm regression with all variables.

fit_small The r object from the glm regression with selected variables.

yX The dataframe restricted to the variable in vars_x and var_y.

Examples

```
## Not run:
data(DebTrivedi)
A=DebTrivedi
A$hospbin = as.integer(A$hosp>0)
vars_x = c("health","region","gender","married","employed")
var_y  = "hospbin"
gg     = dataepi::tab_glmorr(A,vars_x,var_y)
print(gg$allors)

## End(Not run)
```

---

tab_tt2classes_cont *A function for generating a table with the t-test of diverse variables*

Description

This function returns the whole table with the t-tests from a set of continuous variables versus one categorical variable with two (or eventually more modalities).

Usage

```
tab_tt2classes_cont(A, vars_cont, var_y)
```

Arguments

A	The data.frame with the data.
vars_cont	The names of the continuous variables.
var_y	The name of the categorical variable.

Value

A data.frame with each row for a continuous variable and with the following columns.

var1 The name of the continuous variable from group 1.

median1 The median of the continuous variable from group 1.

mean1 The mean of the continuous variable from group 1.

sd1 The standard-deviation of the continuous variable from group 1.

nb1 The sample size of group 1.

var2 The name of the continuous variable from group 2.

median2 The median of the continuous variable from group 2.

mean2 The mean of the continuous variable from group 2.

sd2 The standard-deviation of the continuous variable from group 2.

nb2 The sample size of group 2.

T_t.test (2cl) The statistics computed for the t-Student test.

P_t.test (2cl) The p-value computed for the t-Student test.

P¡0.05 Power_t.t The power computed for the t-Student test.

Examples

```
data(DebTrivedi)
A          <- DebTrivedi
vars_cont <- c("age","ofp")
resu_     <- tab_tt2classes_cont(A,vars_cont,"gender")
print(resu_)
```

---

tab_ttanova_cont *A function for generating a table with the anova of diverse variables*

Description

This function returns the whole table with the anova tests from a set of continuous variables and categorical variables with different number of modalities, plus other related tests.

Usage

```
tab_ttanova_cont(A, vars_cont, vars_disc)
```

Arguments

`A`	The data.frame with the data.
`vars_cont`	The names of the continuous variables.
`vars_disc`	The names of the categorical variables.

Value

A list with the following entries.

tabstat A data.frame with each row for a continuous variable and a categorical variable (binary or poly) with the following columns.

- **var_cont** The name of the continuous variable.
- **var_disc** The name of the discrete variable.
- **n1** The size of groupe 1.
- **n2** The size of groupe 2.
- **...** ...
- **ng** The size of groupe g (if it exists for the variable in var_disc).
- **norm1** The p-value of the normality test from group 1.
- **norm2** The p-value of the normality test from group 2.
- **...** ...
- **normg** The p-value of the normality test from group g (if exists).
- **test12_vr** The p-value of the equality test of the variances for two groups.
- **test12_eq** The p-value of the equality t-test of the means for two groups.
- **test12_gt** The p-value of the equality t-test of the means for two groups with option "greater".
- **test12_ls** The p-value of the equality t-test of the means for two groups with option "less".
- **test12_wx_eq** The p-value of the wilkox rank sum (not paired) equality test of the means for two groups.
- **test12_wx_gq** The p-value of the wilkox rank sum (not paired) equality test of the means for two groups with option "greater".
- **test12_wx_ls** The p-value of the wilkox rank sum (not paired) equality test of the means for two groups with option "less".
- **test_vr** The p-value of the equality test of the variances for g¿2 groups.
- **test_aov** The p-value of the equality test of the means for g¿2 groups.
- **test_aov_check** The p-value from the normality test of the residual from the anova test for g¿2 groups.
- **test_welch** The p-value from the welch test for g¿=2 groups and variances not equal, under normality and variances supposed not equal.
- **test_krusk** The p-value from the kruskal-wallis rank sum test for g¿=2 groups to check the equality in distribution of the means.

pairs_no_panova A list of pairs of variables with non available anova test.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
vars_cont <- c("age","ofp")
vars_disc <- c("gender","region","health")
resu_     <- tab_ttanova_cont(A,vars_cont,vars_disc)
print(resu_)
```

`viz_all2x2` *A function for showing the modalities in a 2d view plus a related indicator*

Description

This function take the set of 2x2 table for the odds ratio in order to construct a new matrix of the whole set of modalities and performs pca. The obtained projection is visualized and an indicator is also obtained.

Usage

```
viz_all2x2(TABSTAT, g_ = 3, graph_ = "PCA")
```

Arguments

`TABSTAT`	The table from the list of 2x2 tables from the data.
`g_`	The number of groups for the clustering.
`graph_`	The characters string with null value for no projection, with the value to show the "PCA", the value "COSTAT" to show the percentages for Sick=1, and the value "CASTAT" to show the percentages for the Sick=0.

Value

A list with the following entries.

tabstat A data.frame with in the first left columns the matrix from the tables 2x2 aggregated from the function tab_all2x2() after normalization of its pairs of columns, followed by two columns, one for the variable name and one for the modality name.

pca The r object from a pca of the normalized matrix of counts.

kmeans The r object from a kmeans of the normalized matrix of counts.

ORpca An alternative indicator for odds ratios.

Examples

```
data(DebTrivedi)
A         <- DebTrivedi
A$hospbin <- as.integer(A$hosp>0)
vars_disc <- c("health","gender","region")
var_y     <- "hospbin"
resu_or   <- dataepi::tab_all2x2(A,vars_disc,var_y,dataepi::stat_oddsratio)
resu_fv   <- viz_all2x2(resu_or$tabstat,g_=7,graph_="PCA")
print(resu_fv$tabstat)
```

Index

Exemple of generating a report from a health dataset with the r package *dataepi*

R. Priam*

June 1, 2021

Abstract

The r package dataepi allows to generate a report for the analysis of a dataset via a kx2x2 health table with odds ratios, relative risks, descriptive statistics of the variable and statistical tests for two variables. This document presents un example of first analysis and a description of several functionalities of the package.

Loading of the dataset and preparation of the variables

First the library is loaded with rstudio. The data.frame is in the variable A for analyzing with the r package dataepi. The command for loading the library is *library()* preparing the variable for the package are as follows:

```
> library(MASS)
> data("Pima.tr")
> data("Pima.te")
> A = rbind(Pima.tr,Pima.te)
> A$id=1:nrow(A)
> A$npreg = as.character(A$npreg)
> A$npreg[A$npreg%in%as.character(10:17)]="10_17"
> A$diabet = as.numeric(as.character(A$type)=="Yes")
> A$bp_pb = as.character(as.numeric(as.numeric(as.character(A$bp))>90))
> A$glu_cl3 = as.character(cut(A$glu,c(56,103,129,199),
+                              labels=c("56_103","103_129","129_199")))
> A$skin_cl3 = as.character(cut(A$skin,c(7,24,33,99),
+                               labels=c("7_24","24_33","33_99")))
> A$bmigroups = as.character(cut(A$bmi,c(0,25,30,100),
+                                labels=c("a_normal","b_overweight","c_obese")))
> A$ped_cl3 = as.character(cut(A$ped,c(0.085,0.295,0.557,2.420),
+                              labels=c("low","middle","large")))
> A$age_cl3 = as.character(cut(A$age,c(21,24,33,81),labels=c("21_24","24_33","33_81")))
>
> var_y = "diabet"
> vars_cont = c("age","bp","glu","skin","bmi","ped","age")
> vars_disc = c("npreg","bp_pb","bmigroups","glu_cl3","skin_cl3","ped_cl3","age_cl3")
```

*rpriam@gmail.com

```
> vars_x = c("npreg","bp_pb","bmigroups","skin_cl3","ped_cl3","age_cl3")
> vars_int = NULL
> var_id = "id"
```

Checking last character
It is not allows a digit as last character of the name of a variable, hence, this may be added,

```
> vars_cont = unique(vars_cont)
> vars_disc = unique(vars_disc)
> vars_x    = unique(vars_x)
>
> for (j in 1:ncol(A)) {
+     nv=names(A)[j]
+     if(substr(nv,nchar(nv),nchar(nv))%in%paste(0:9))
+       names(A)[j] = paste(names(A)[j],"_",sep="")
+     if (sum(nv%in%vars_cont))
+       { l = which(vars_cont%in%nv); vars_cont[l] = names(A)[j]; }
+     if (sum(nv%in%vars_disc))
+       { l = which(vars_disc%in%nv); vars_disc[l] = names(A)[j]; }
+     if (sum(nv%in%vars_x))
+       { l = which(vars_x%in%nv); vars_x[l] = names(A)[j]; }
+ }
```

Adding the description of the study (facultative)
The descriptive for the study may be added as,

```
> list_supp = list()
> list_supp$where     = " "
> list_supp$who       = " "
> list_supp$disease   = " "
> list_supp$objective = " "
> list_supp$project   = " "
> list_supp$inex      = " "
```

Checking the variables
Let's have a look to the variables, the corresponding output is as follows.

```
> str(A)
'data.frame': 532 obs. of  16 variables:
 $ npreg    : chr  "5" "7" "5" "0" ...
 $ glu      : int  86 195 77 165 107 97 83 193 142 128 ...
 $ bp       : int  68 70 82 76 60 76 58 50 80 78 ...
 $ skin     : int  28 33 41 43 25 27 31 16 15 37 ...
 $ bmi      : num  30.2 25.1 35.8 47.9 26.4 35.6 34.3 25.9 32.4 43.3 ...
 $ ped      : num  0.364 0.163 0.156 0.259 0.133 ...
 $ age      : int  24 55 35 26 23 52 25 24 63 31 ...
```

```
 $ type     : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 2 1 1 1 2 ...
 $ id       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ diabet   : num  0 1 0 0 0 1 0 0 0 1 ...
 $ bp_pb    : chr  "0" "0" "0" "0" ...
 $ glu_cl3_ : chr  "56_103" "129_199" "56_103" "129_199" ...
 $ skin_cl3_: chr  "24_33" "24_33" "33_99" "33_99" ...
 $ bmigroups: chr  "c_obese" "b_overweight" "c_obese" "c_obese" ...
 $ ped_cl3_ : chr  "middle" "low" "low" "low" ...
 $ age_cl3_ : chr  "21_24" "33_81" "33_81" "24_33" ...
```

It is recognized variables with discrete values, continuous values, binary values or polytomous values. Normally, the dataset may be known because it may have been produced by the investigator of the study, otherwise it is wise to have a look of the number of values of each variable, and when they are not too numerous their unique set.

Preparation of the dataset with the function data_prepare()
A function of the r package prepares the dataset for the other functions,

```
> A0 = A;
> fp = data_prepare(A,var_y,vars_cont,vars_disc,var_id)
> A  = fp$A
> A  = A[,unique(c(var_y,vars_cont,vars_disc,vars_x,var_id))]
```

Checking the variables with the function tab_contents()
A description of the variables for the analysis is as follows,

```
> desc_all          <- tab_contents(A)
> print(desc_all)
    variable   r_class nblevels nbobs
1     diabet   numeric        2   532
2        age   numeric       46   532
3         bp   numeric       42   532
4        glu   numeric      126   532
5       skin   numeric       50   532
6        bmi   numeric      222   532
7        ped   numeric      413   532
8      npreg character       11   532
9      bp_pb character        2   532
10 bmigroups character        3   532
11  glu_cl3_ character        4   532
12 skin_cl3_ character        4   532
13  ped_cl3_ character        4   532
14  age_cl3_ character        4   532
15        id   integer      532   532
```

Checking the generated tables, one by one
A way to get the tables for the analysis leads to,

```
> desc_cont         <- tab_desc_cont(A,vars_cont)
> desc_disc         <- tab_desc_disc(A,vars_disc)
> desc_biv          <- tab_desc2class_cont(A,vars_cont,var_y)
> test_tt           <- tab_tt2classes_cont(A,vars_cont,var_y)
> test_anova        <- tab_ttanova_cont(A,vars_cont,vars_disc)
> test_chi2         <- tab_chi2all(A,c(var_y,vars_disc),pvalue_seuil_ = 0.05)
> or                <- tab_all2x2(A,vars_x,var_y,stat_oddsratio)
> rr                <- tab_all2x2(A,vars_x,var_y,stat_relativerisk)
> gg                <- tab_glmorr(A,vars_x,var_y)
```

Checking the generated tables, all once, with the function rep_compute()
A way to get all thes table for the analysis is as:

```
> au = dataepi::rep_compute(A, var_y, vars_x, vars_cont, vars_disc, vars_int, var_id)
> print(names(au[1:7]))
[1] "Anew"       "desc_all"   "desc_cont"  "desc_disc"  "desc_biv"   "test_tt"    "test_anova"
> print(names(au[8:13]))
[1] "test_chi2" "or"        "rr"        "gg"        "fv"        "args"
> print(au$args$vars_x)
[1] "npreg"     "bp_pb"     "bmigroups" "skin_cl3_" "ped_cl3_"  "age_cl3_"
> print(au$args$var_y)
[1] "diabet"
```

Generating the report with .tex extension with the function rep_write()
The command lines for the report creation before compilation into a pdf or ps file are,

```
> fnl <- paste("./report_dataepi_", data_,".tex",sep="")
> if (file.exists(fnl)) {file.remove(fnl);}
[1] TRUE
> if (!exists("list_supp")) list_supp=NULL;
> wr = dataepi::rep_write(fnl,"tex",A, var_y, vars_x, vars_cont,
+                          vars_disc, vars_int, var_id, list_supp)
```

This function executes the function dataepi::au() and then write the report in the file with the name in the variable fnl. To check the header of the file,

```
> file_tex = read.csv(fnl, header = FALSE)
> print(file_tex[1:10,])
 [1] \\documentclass[12pt]{article}
 [2] \\usepackage[margin=0.7in]{geometry}
 [3] \\usepackage[utf8]{inputenc}
 [4] \\usepackage{graphics}
 [5] \\usepackage{datetime}
 [6] \\usepackage{pdflscape}
 [7]
 [8] \\author{ }
 [9] \\date{\\today (\\currenttime)}
[10] \\title{Report\\footnote{This document is auto-generated from the r package daatepi.} for
```