

$r \times c$ ($I \times J$) Contingency Tables

Berkeley Admission Data

The following table describes numbers of accepted and rejected applicants to 6 graduate programs at the University of California, Berkeley.

	Accepted	Rejected
A	601	332
B	370	215
C	322	596
D	269	523
E	147	437
F	46	668

We can examine the acceptance rates by considering the row percentages. In the following `progAdmits` is a 6×2 matrix.

```
> percents(progAdmits, 1)

      Y              N
A 601/933 (64.4%) 332/933 (35.6%)
B 370/585 (63.2%) 215/585 (36.8%)
C 322/918 (35.1%) 596/918 (64.9%)
D 269/792 (34.0%) 523/792 (66.0%)
E 147/584 (25.2%) 437/584 (74.8%)
F  46/714 ( 6.4%) 668/714 (93.6%)
```

We can test the hypothesis of independence of acceptance and program (which is equivalent to saying the all programs have the same underlying acceptance probability) by applying Pearson's Chi-squared test, e.g.

```
> chisq.test(progAdmits)

Pearson's Chi-squared test

data:  progAdmits
X-squared = 778.9065, df = 5, p-value < 2.2e-16
```

We can make pairwise comparisons between programs, while controlling the overall type I error rate, using `pairwise.prop.test()`

```
> pairwise.prop.test(progAdmits)
```

Pairwise comparisons using Pairwise comparison of proportions

data: progAdmits

	A	B	C	D	E
B	1.00000	-	-	-	-
C	< 2e-16	< 2e-16	-	-	-
D	< 2e-16	< 2e-16	1.00000	-	-
E	< 2e-16	< 2e-16	0.00027	0.00168	-
F	< 2e-16	< 2e-16	< 2e-16	< 2e-16	< 2e-16

P value adjustment method: holm

Job Satisfaction Data

A sample of 901 men filled out a questionnaire that asked about job satisfaction and yearly salary with the following results (this was quite a few years ago!)

	very.dissatisfied	little.dissatisfied	moderately.satisfied	very.satisfied
<6,000	20	24	80	82
6,000-15,000	22	38	104	125
15,000-25,000	13	28	81	113
>25,000	7	18	54	92

Row percents make the table easier to interpret (of course if we had reversed rows and columns, we would use column percents).

```
> percents(incomeSat,1)
              very.dissatisfied little.dissatisfied
moderately.satisfied
<6,000      20/206( 9.7%)      24/206(11.7%)      80/206(38.8%)
6,000-15,000 22/289( 7.6%)      38/289(13.1%)     104/289(36.0%)
15,000-25,000 13/235( 5.5%)      28/235(11.9%)     81/235(34.5%)
>25,000      7/171( 4.1%)      18/171(10.5%)     54/171(31.6%)

              very.satisfied
<6,000      82/206(39.8%)
6,000-15,000 125/289(43.3%)
15,000-25,000 113/235(48.1%)
>25,000      92/171(53.8%)
```

We can test for a relationship between salary and job satisfaction using Pearson's Chi-square.

```
> chisq.test(incomeSat)
```

Pearson's Chi-squared test

```
data: incomeSat
X-squared = 11.9886, df = 9, p-value = 0.2140
```

Purum Marriage Data

A sociologist studying the marriage customs of members of an old and isolated tribe known as the Purum. The Purum consisted of 5 clans - the Marrim, Makan, Parpa, Thao and Kheyang. The sociologist knew that marriages were forbidden within clans, and between certain sets of clans. The following table describes the clan membership of 128 husbands and wives. Cells with dashes (-) denote forbidden unions.

	Marrim	Makan	Parpa	Thao	Kheyang
Marrim	-	5	17	-	6
Makan	5	-	0	16	2
Parpa	-	2	-	10	11
Thao	10	-	-	-	9
Kheyang	6	20	8	0	1

The sociologist wished to examine if husbands exhibited preferences for wives from certain clans, or whether there was no preference. Lack

of any preference implied a certain type independence between the clan of the wife and the clan of the husband, at least amongst the clans that the husband was free to choose a wife from. This kind of restricted independence is referred to as quasi-independence.

It is possible to represent the multiplicative relationship of quasi-independence using a log-linear model. First, one re-organizes the data as if we had Poisson counts.

```
> print(marDf)
      Wife Husband count
2   Makan  Marrim     5
4   Thao  Marrim    10
5  Kheyang  Marrim     6
6   Marrim   Makan     5
8   Parpa   Makan     2
10  Kheyang   Makan    20
11  Marrim   Parpa    17
12  Makan   Parpa     0
15  Kheyang   Parpa     8
17  Makan   Thao    16
18  Parpa   Thao    10
20  Kheyang   Thao     0
21  Marrim  Kheyang     6
22  Makan  Kheyang     2
23  Parpa  Kheyang    11
24   Thao  Kheyang     9
25  Kheyang  Kheyang     1
```

Then one fits log-linear models using the *poisson* family. This seems somewhat un-natural, as the counts are multinomial, not independent Poisson. However the appropriate multinomial likelihood happens to have the same algebraic form as a Poisson likelihood, so we can use this trick to test for independence, as follows.

```
> attach(marDf)
> llfit <- glm(count ~ Wife + Husband, family=poisson)
> llfitSat <- glm(count ~ factor(row.names(marDf)), family=poisson)
> anova(llfit,llfitSat,test="Chi")
Analysis of Deviance Table

Model 1: count ~ Wife + Husband
Model 2: count ~ factor(row.names(marDf))
  Resid. Df Resid. Dev Df Deviance P(>|Chi|)
1         8      76.251
2         0  3.033e-10  8    76.251 2.770e-13
```