

# Psychonomics 2023: Chi-Square Test

Joanna Morris

2023-11-13

Let's import our data

```
library(readr)
subjectProfiles <- read_csv("M21_subjectProfiles_hampshire_2.csv")

## Rows: 45 Columns: 3
## -- Column specification -----
## Delimiter: ","
## chr (3): Sensitivity, ReadingProfile, SubjID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
n400 <- read_csv("m21_vsl_300500_150050.csv")
```

```
## Rows: 32805 Columns: 6
## -- Column specification -----
## Delimiter: ","
## chr (3): chlabel, binlabel, SubjID
## dbl (3): value, chindex, bini
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Let's examine the relationship between sensitivity and reading profile using a **mosaic plot** for contingency tables. A **mosaic plot** is a graphical representation of two way contingency table which pictographically represents the relationship among two or more categorical variables. The area of the tiles, also known as the bin size, is proportional to the number of observations within that category

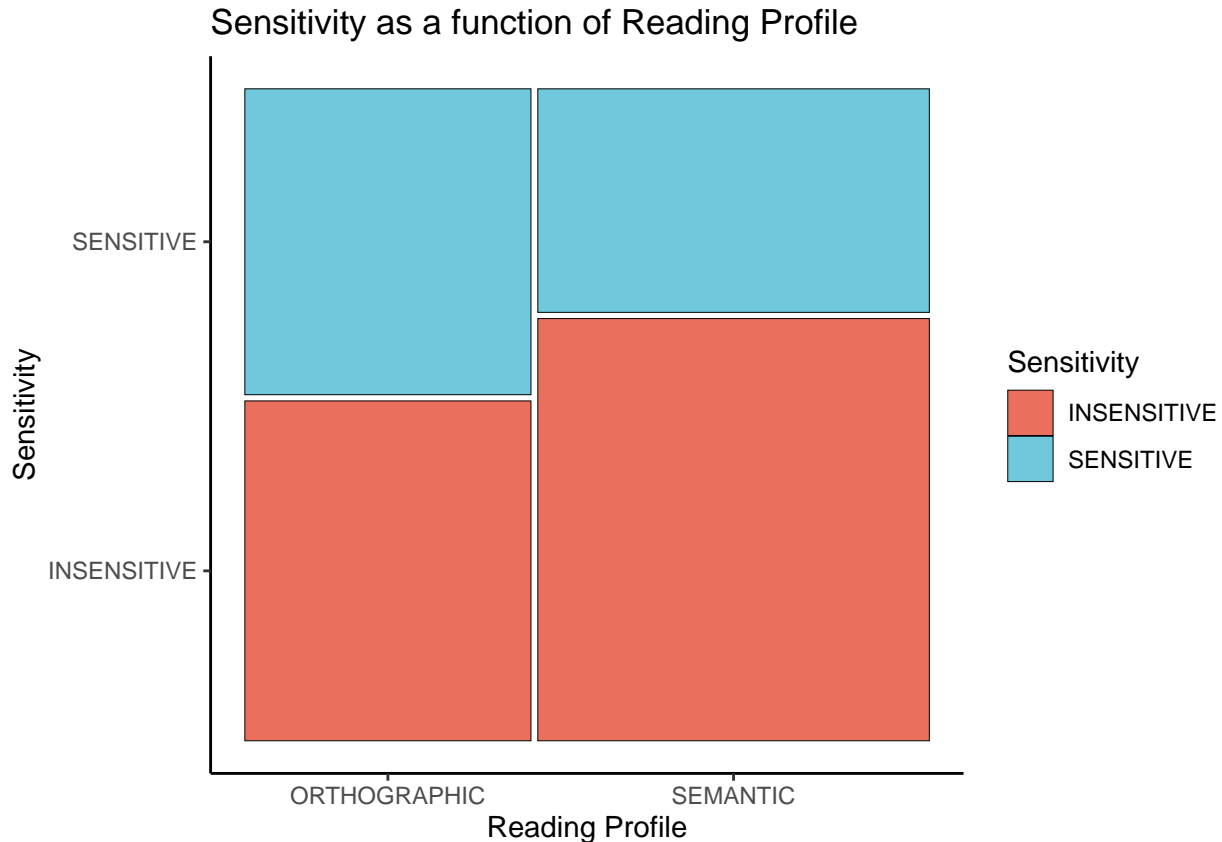
```
library(ggmosaic)
```

```
## Loading required package: ggplot2
```

```
library(ggmosaic)
library(ggsci)
mosaic_plot1 <- ggplot(data = subjectProfiles) +
  geom_mosaic(aes(x = product(Sensitivity, ReadingProfile), fill = Sensitivity), colour = "black") +
  labs(y="Sensitivity", x="Reading Profile", title = "Sensitivity as a function of Reading Profile") +
  # scale_color_manual(values = c("darkslategray4", "pink3"))+
  # scale_fill_manual(values = c("darkslategray4", "pink3"))
mosaic_plot1
```

```
## Warning: `unite()` was deprecated in tidyr 1.2.0.
## i Please use `unite()` instead.
## i The deprecated feature was likely used in the ggmosaic package.
## Please report the issue at <https://github.com/haleyjeppson/ggmosaic>.
```

```
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Now we can do a chi-square test to see if the proportions are significantly different. We begin by creating a frequency table

```
(responses_table <- table(subjectProfiles$Sensitivity, subjectProfiles$ReadingProfile))
```

```
##
##           ORTHOGRAPHIC SEMANTIC
## INSENSITIVE          10         17
## SENSITIVE             9          9
```

The chi-square approximation to the distribution of the test statistic relies on the counts being roughly normally distributed. If many of the expected counts are very small, the approximation may be poor. You can deal with the problem by setting the `simulate.p.value` argument to `TRUE`; If `simulate.p.value` is `FALSE`, the `p`-value is computed from the asymptotic chi-squared distribution of the test statistic; Otherwise the `p`-value is computed for a Monte Carlo test (Hope, 1968) with `B` replicates. In this case, you aren't reliant on the chi-square approximation to the distribution of the test statistic.

```
(chisqr <- chisq.test(responses_table, simulate.p.value = TRUE, B = 100000))
```

```
##
## Pearson's Chi-squared test with simulated p-value (based on 1e+05
## replicates)
##
## data:  responses_table
## X-squared = 0.74393, df = NA, p-value = 0.5367
```

The result of `chisq.test()` function is a list containing the following components:

*statistic*: the value the chi-squared test statistic. *parameter*: the degrees of freedom *p.value*: the p-value of the test *observed*: the observed count *expected*: the expected count

```
chisqr$statistic
```

```
## X-squared
## 0.7439271
```

```
chisqr$parameter
```

```
## df
## NA
```

```
chisqr$p.value
```

```
## [1] 0.5367446
```

```
chisqr$observed
```

```
##
##               ORTHOGRAPHIC SEMANTIC
## INSENSITIVE           10         17
## SENSITIVE             9          9
```

```
chisqr$expected
```

```
##
##               ORTHOGRAPHIC SEMANTIC
## INSENSITIVE          11.4        15.6
## SENSITIVE             7.6         10.4
```

For such small counts, you could use Fisher's exact test.

From Wikipedia "With large samples, a chi-squared test can be used in this situation. However, the significance value it provides is only an approximation, because the sampling distribution of the test statistic that is calculated is only approximately equal to the theoretical chi-squared distribution. The approximation is inadequate when sample sizes are small, or the data are very unequally distributed among the cells of the table, resulting in the cell counts predicted on the null hypothesis (the "expected values") being low. The usual rule of thumb for deciding whether the chi-squared approximation is good enough is that the chi-squared test is not suitable when the expected values in any of the cells of a contingency table are below 5, or below 10 when there is only one degree of freedom"

```
fisher.test(responses_table)
```

```
##
## Fisher's Exact Test for Count Data
##
## data: responses_table
## p-value = 0.5391
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.1486407 2.3290971
## sample estimates:
## odds ratio
## 0.5953282
```