# Tutorial 4

ST2137-2420

## Material

This tutorial covers the topics and concepts from chapter 4 of the course textbook. The questions offer practice on understanding and visualising categorical data. The final section introduces yet another measure that we can use to assess association between two categorical random variables.

As always, work out each question with both R and Python unless otherwise stated.

## Dataset: Student Performance

1. The columns `address` and `paid` are both binary variables. Conduct a $\chi^2$-test at 5% significance level to assess if there is any significant association between the two variables.

### R code

```r
# R
stud_perf <- read.table("data/student/student-mat.csv", sep=";",
                        header=TRUE)
add_paid_table <- table(stud_perf[,c("address", "paid")])
(chisq_out <- chisq.test(add_paid_table))
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  add_paid_table
X-squared = 0.86124, df = 1, p-value = 0.3534
```

### Python code

```python
# python
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt

stud_perf  = pd.read_csv("data/student/student-mat.csv", delimiter=";")
add_paid_table = pd.crosstab(stud_perf.address, stud_perf.paid).to_numpy()

chisq_output = stats.chi2_contingency(add_paid_table)
```

When asked to perform a hypothesis test, it is important to lay out all the steps. Here are the steps for this question:

- Steps 1 & 2: The null hypothesis is that the two categorical variables are not associated. The test will be conducted at the 5% significance level.
- Step 3: The value of the test-statistic is 0.86.
- Step 4: The $p$-value is 0.35.

- Step 5: We do not reject $H_0$ at the 5% level. We conclude there is no significant evidence of association.

To assess the assumptions, we need to check the expected counts. The test is valid since none of them are less than 5.

```
# python
chisq_output.expected_freq
```

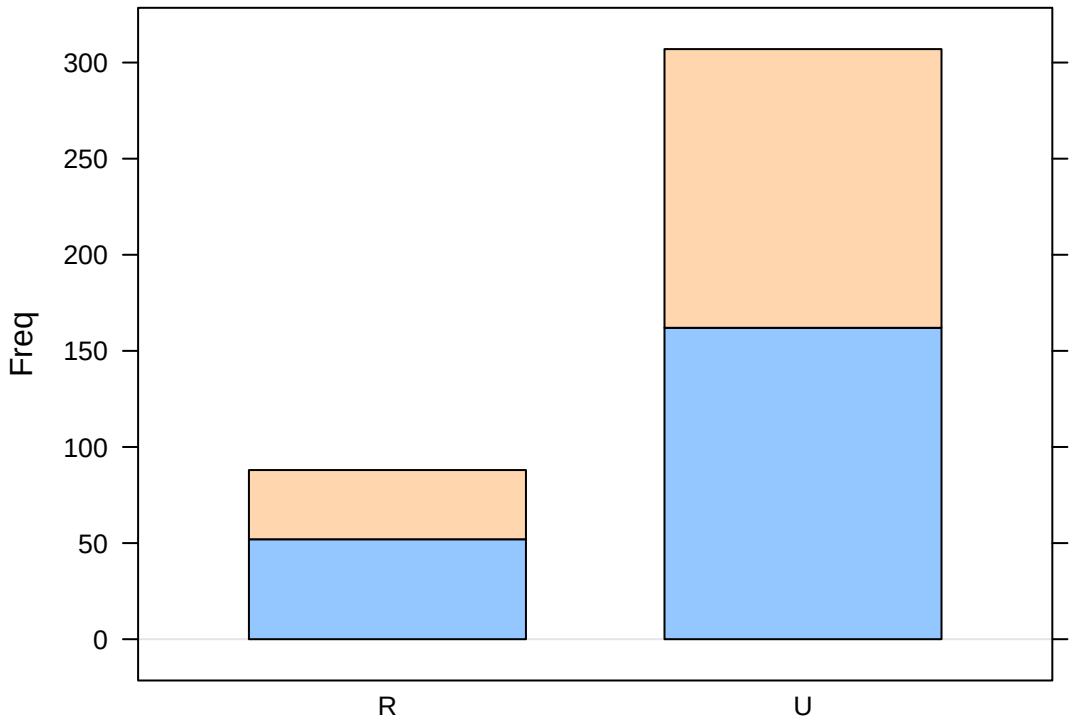
```
array([[ 47.67594937,  40.32405063],
       [166.32405063, 140.67594937]])
```

2. Create a barchart to visualise these two columns.

The following barchart represents the counts directly:

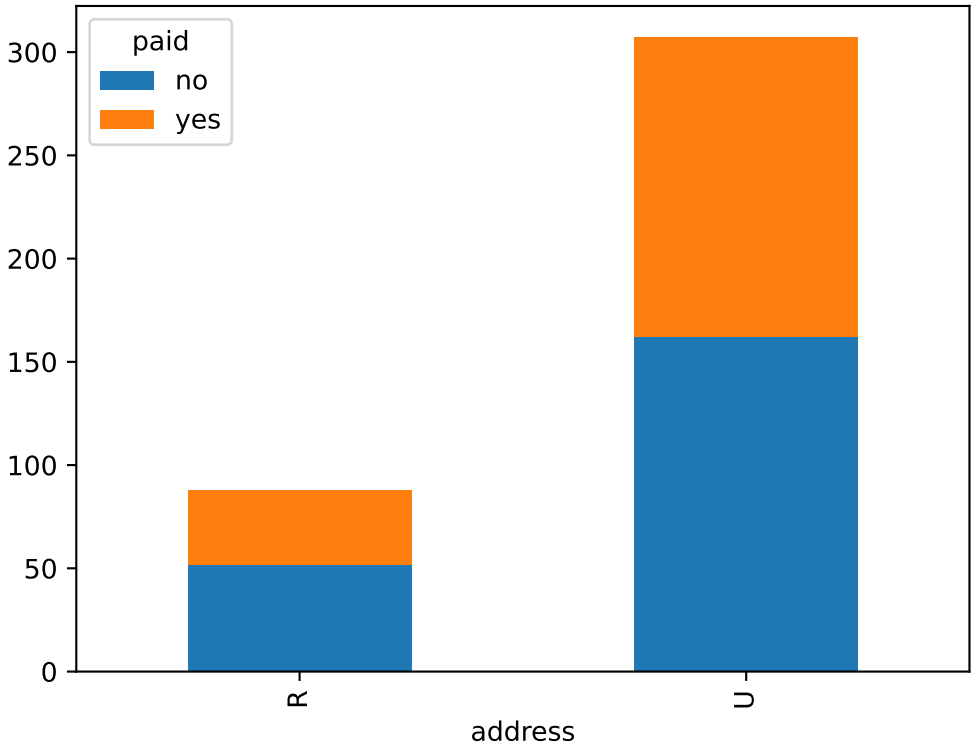
**R code**

```
# R
library(lattice)
barchart(add_paid_table, horizontal = FALSE)
```




**Python code**

```
# python
pd.crosstab(stud_perf.address,
            stud_perf.paid).plot(kind='bar', stacked=True);
```




3. Modify the table to present row-wise proportions instead of raw counts. The `address` variable should appear in the rows.

The learning point in this question is to practice with the recycling rule in R and broadcasting in numpy.

Remember that:

- In R, the shorter vector will be recycled to match the length of the longer vector. Matrices will be filled column-wise instead of row-wise.
- In numpy, if the dimensions of two arrays do not match for an operation, the array with fewer dimensions will be repeated in that dimension to make them match.

**R code**

```r
# R
add_paid_prop <- add_paid_table/rowSums(add_paid_table)
```

**Python code**

```python
# python
add_paid_prop  = add_paid_table / add_paid_table.sum(axis=1).reshape((2,1))
```

4. *Relative risk* is another measure of association between two categorical variables. With the above two variables, it is defined as follows:
   - Let $\hat{p}_1$ be the proportion of rural residents who paid for extra classes.
   - Let $\hat{p}_2$ be the proportion of urban residents who paid for extra classes. Then

$$\text{relative risk} = \hat{p}_1/\hat{p}_2$$

   Compute the relative risk for the above variables.

**R code**

```
# R
rr <- add_paid_prop[1,2] / add_paid_prop[2,2]
```

**Python code**

```
# python
rr = add_paid_prop[0,1]/add_paid_prop[1,1]
```

5. What are the range of values for $RR$? How is it similar/different to Odds Ratio?

RR takes values from 0 to infinity. A value of 1 indicates no association.

Consider a 2x2 table with values a,b,c and d. The RR (row-wise) is computed as

$$a/(a+b)/c/(c+d)$$

The OR is computed as

$$ad/bc = (a/b)/(c/d)$$

When a is small compared to b, *and* c is small compared to d, then the RR will be close to $ad/bc$. It will be close to the Odds Ratio. The downside of RR is that we have to choose to do it row-wise or column-wise. The strength of association will be different depending on the choice. For OR, it does not matter; the strength of the association will be the same.

6. Read up on the `cut` function in R and `pd.cut` in `pandas`. Use it to divide the `G3` column into letter grades:

| G3 score | Letter grade |
|---|---|
| [0, 10] | F |
| (10, 12] | D |
| (12, 15] | C |
| (15, 18] | B |
| (18, 20] | A |

**R code**

```
# R
stud_perf$letter_grade <- cut(stud_perf$G3,
                              breaks= c(-Inf, 10, 12, 15, 18, 20),
                              labels=c("F", "D", "C", "B", "A"))
```

**Python code**

```
# Python
stud_perf['letter_grade'] = pd.cut(stud_perf.G3, [-2, 10, 12, 15, 18 , 20])
```

7. There are 6 Likert-scale survey questions (`famrel` to `health`). Which one of them is most strongly associated with `letter_grade` (from question 6)? Which measure of association did you use?

**R code**

```
# R
library(DescTools)
for (i in 24:29){
  tmp_table <- table(stud_perf[, c(i, 34)])
```

```
  all_assocs <- Desc(tmp_table, plotit = FALSE)[[1]]$assocs
  cat("Kendall Tau-b for", names(stud_perf)[i], ":\t", round(all_assocs[3,1], 3),  "\n")
}
```

```
Kendall Tau-b for famrel :   0.028
Kendall Tau-b for freetime :     -0.007
Kendall Tau-b for goout :    -0.124
Kendall Tau-b for Dalc :     -0.115
Kendall Tau-b for Walc :     -0.13
Kendall Tau-b for health :   -0.027
```

### Python code

```
# Python
for var in stud_perf.columns[23:29]:
    kt = stats.kendalltau(stud_perf.letter_grade, stud_perf[var])
    print(f"Kendall tau-b for {var}:\t {kt.statistic:.3f}")
```

```
Kendall tau-b for famrel:    0.028
Kendall tau-b for freetime:  -0.007
Kendall tau-b for goout:     -0.124
Kendall tau-b for Dalc:  -0.115
Kendall tau-b for Walc:  -0.130
Kendall tau-b for health:    -0.027
```

The strongest association is with Weekend alcohol. The value of Kendall $\tau_b$ is -0.13. The association is negative, which means that lower grades are associated with increased alcohol on weekends.

8. Recreate the following mosaic plot for `Dalc` and `Walc`. Interpret what the plot shows. In addition, identify the cell counts corresponding to the labelled rectangles ("Cell Count?").

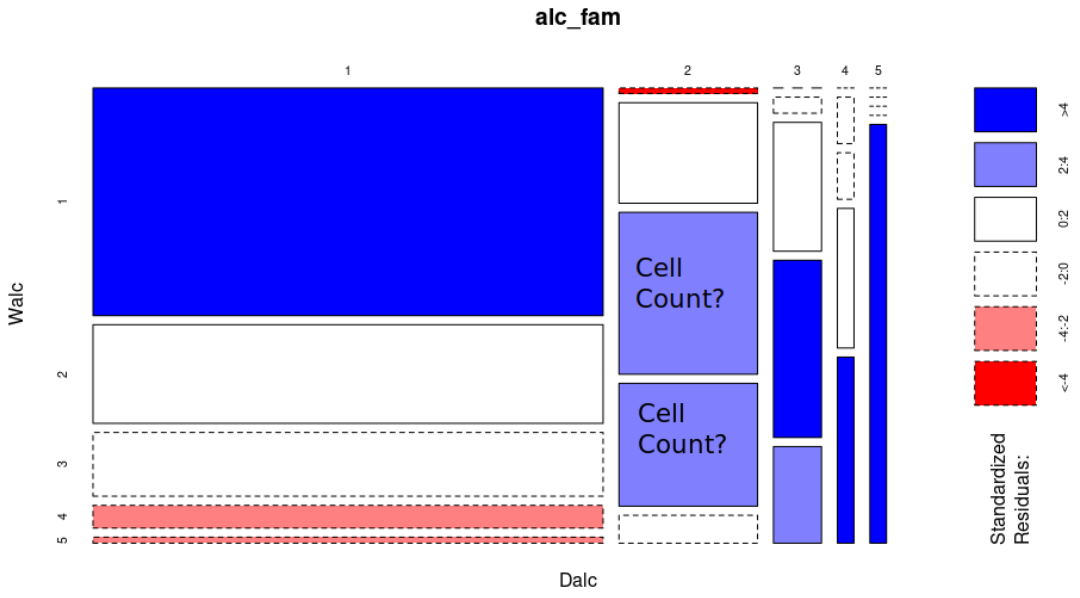


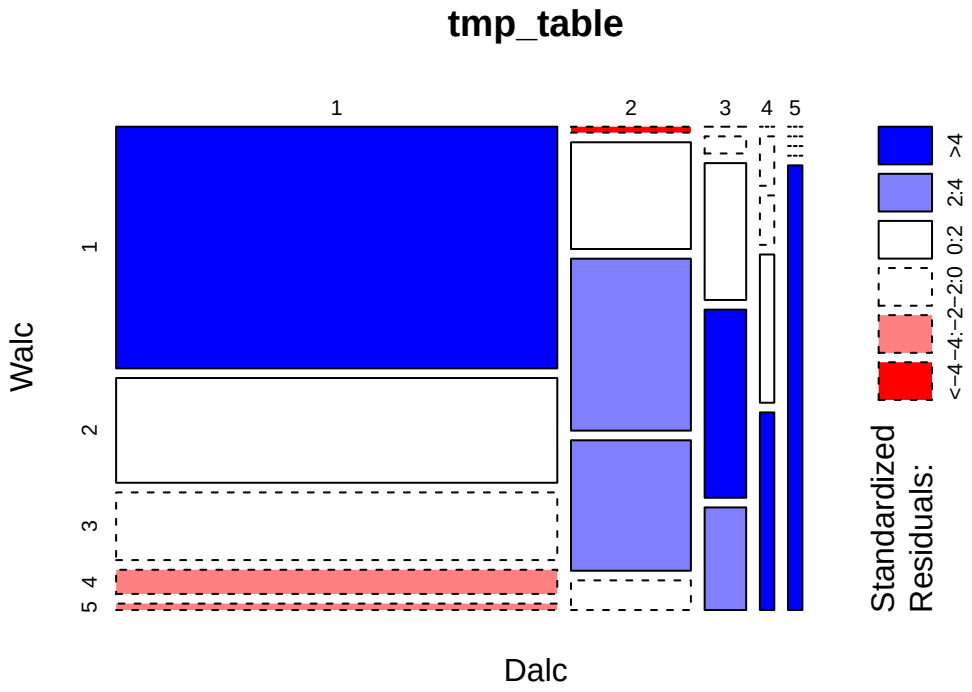

Figure 1: Mosaic plot

The cell counts are 29 and 22. There is a strong positive association between Weekend and Weekday

alcohol consumption. Most students drink very little on weekdays, as indicated by the wide rectangles in the first column.

**R code**

```
# R
tmp_table <- table(stud_perf$Dalc, stud_perf$Walc)
mosaicplot(tmp_table, shade=TRUE, xlab="Dalc", ylab="Walc")
```




## Mutual Information

If $X$ and $Y$ are two discrete random variables, then the mutual information between them is given by

$$I(X,Y) = \sum_{i,j} P(X=i, Y=j) \times \ln\left(\frac{P(X=i,Y=j)}{P(X=i)P(Y=j)}\right)$$

To clarify, you have to consider all possible values that $X$ takes on, and all possible values that $Y$ takes on in the summation.

Here is some intuition behind MI: $I(X,Y)$ measures how different the joint pmf is from the product of the marginals. If $X$ and $Y$ are independent, then $I(X,Y)$ will be 0. In this case, $X$ and $Y$ hold no information about each other - $X$ would probably not be useful in predicting $Y$. In general, the more positive this value $I(X,Y)$ is, the more associated these two variables are, and the more useful $X$ could be for predicting $Y$.

In order to estimate probabilities, a typical approach is to compute the sample proportions.

9. Find the mistakes in the following functions that have been written to compute MI.

```r
## Solution 1:
mi1 <- function(x, y) {
  total <- length(x)
  output = 0

  pX = table(x)/length(x)
  pY = table(y)/length(y)
  X <- unique(pX)
  Y <- unique(pY)

  pXY <- as.vector(table(x,y)/length(x))
  XY <- unique(pXY)

  for(i in X) {
    for (j in Y){
      for (k in XY){
        output <- output + k * log(k/(i+j))
      }
    }
  }
  return(output)
}

## Solution 2:
mi2 <- function(X, Y) {
  total <- 0
  for (i in 1:length(X)) {
    for (j in 1:length(Y)) {
      ProbX <- length(which(X == X[[i]])) / length(X)
      ProbY <- length(which(Y == Y[[j]])) / length(Y)
      ProbXY <- ProbX * ProbY
      prob <- ln((ProbX + ProbY) / ProbXY)
      total1 <- ProbXY * prob
    }
    total <- total + total1
  }
  total
}
```

Here are the issues with solution 1:

- The solution works with `unique(x)` and `unique(y)`. However, these will take the unique probabilities, not the levels of the variables.
- The ratio `k/(i+j)` should be `k/(i*j)`.

There are several issues with solution 2:

- The loop is over the entire length of X and Y, which is incorrect.
- The line that computes `ProbXY` in fact assumes independence; the estimate of the joint probability is not computed correctly.
- There is no `ln()` function in R.
- That line is doubly incorrect because the joint probability should be in the numerator. Also, the marginals should be multiplied, not added.

10. Implement your own function to compute Mutual Information in R. Use it to compute the MI between `address` and `paid`.

```r
mi0 <- function(x, y){
  # browser()
```

```
  mi <- 0
  n <- length(x)
  xtab <- table(x)
  pX <- as.vector(xtab)/n
  names(pX) <- names(xtab)

  ytab <- table(y)
  pY <- as.vector(ytab)/n
  names(pY) <- names(ytab)

  for(i in unique(x))
    for(j in unique(y)){
      pXY <- mean(x == i & y == j)
      if(pXY != 0){
        mi <- mi + pXY * log(pXY/(pX[i]*pY[j]))
      }
    }
  return(mi)

}
mi0(stud_perf$address, stud_perf$paid)
```

```
          U
0.001401274
```