IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

Name: Hendi Kushta
Date: Today's Date

****Important** Copying and/or pasting anything from the textbook will not be acceptable for your chapter notes submissions. You must write your notes in your own words and generate your own code, results, and graphs in R. This is what forces your brain to process the material that you read.**

INTRODUCTION

Humans depend on pattern recognition for their existence, and this capacity has aided in the advancement of civilization and technology. The example of our ancestor Og starting a fire and realizing the relationship between the quantity of wood and the heat of the fire illustrates association, one of the most fundamental probabilistic patterns in nature. The majority of relationships in nature are, however, flawed and come in different strengths. One of the earliest multivariate statistics to be established was the Pearson product-moment correlation (PPMC), which is a statistical measure of relationship. The PPMC calculates the correlation between two variables by combining the variances of the two variables, with the covariance between the two variables serving as the shared component.
To explore a situation where two variables are not related, we can use R to analyze and visualize the data. We can begin by generating two sets of random data, each with 30 observations, distributed normally, with means and standard deviations that are close to zero and one, respectively.

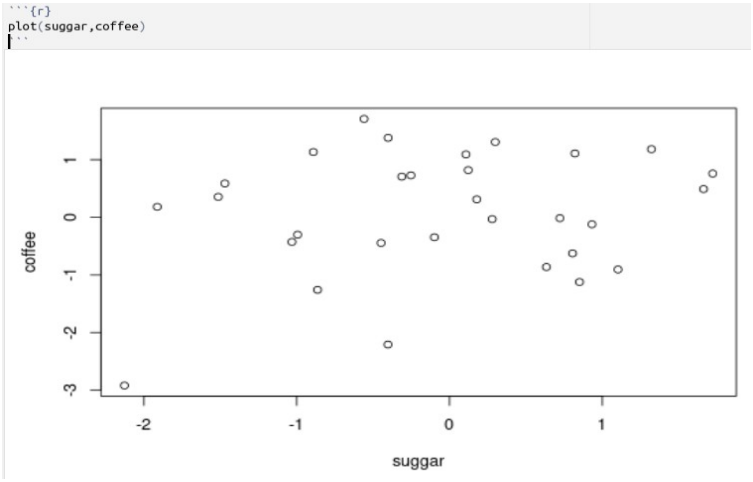
```
> set.seed(223)
> suggar <- rnorm(30)
> coffee <- rnorm(30)
> mean(suggar)
[1] -0.05741684
> mean(coffee)
[1] 0.07460153
> sd(suggar)
[1] 1.015296
> sd(coffee)
[1] 1.072575
```

To obtain consistent results, it is advised to run the set.seed() command before executing other commands. The means and standard deviations of the variables are close to zero and one, respectively, indicating positive and negative values in each vector. Negative values can be considered as the minimum amounts of sugar and coffee. A plot of the variables shows that the ranges are approximately

Originality Assertion: By submitting this file you affirm that this writing is your own.

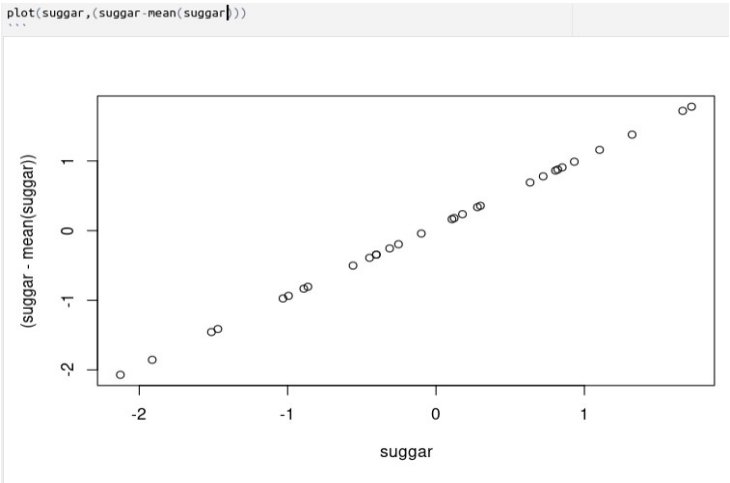
-2 to +2, and many points are near the means of the two variables, which is typical of a normal distribution. The standard normal distribution has a mean of 0 and a standard deviation of 1, which is slightly deviated in the case of the sugar and coffee variables due to randomness and small sample size.

```{r}
plot(suggar,coffee)
```



Both sugar and coffee can be considered standard normal variables for this example. Each observation of sugar and coffee represents a z-score, which is a deviation from the mean calibrated in standard deviations. The relationship between each value of sugar and its deviation from the mean can be seen in scatter plot below, which shows a straight line. By calculating the products of their respective deviations from the mean, we can get an idea of how much these two variables covary. Positive cross-products indicate a strong correlation between high amounts of sugar and high coffee.

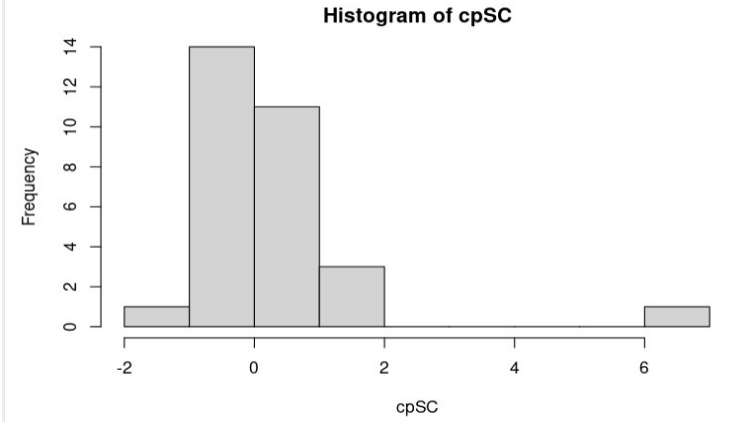
```
plot(suggar,(suggar-mean(suggar)))
```



IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

The variables are assumed to be standard normal variables, with means close to 0 and standard deviations close to 1. The cross-products sugar and coffee' deviations from the mean are calculated, and their mean is used as a measure of covariance. A histogram is also used to visualize the cross-products. The initial dataset shows no association between the variables, with a mean cross-product close to 0. Finally, the article explains how to create a positive correlation between the variables using a simple transformation.

```
cpSC <- suggar * coffee
mean(cpSC)
hist(cpSC)
```



IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

BOX ON P.126: FORMULA FOR PEARSON'S CORRELATION

Formula for Pearson's Correlation

Pearson's product-moment correlation (r) is a commonly used method to measure the connection between two variables. The name product-moment suggests that it involves the multiplication of data values. This measurement ranges from -1 to 1, which indicates a standardized scale. However when dealing with variables with different variances, a different approach is required to calculate r.

$$r = \frac{\Sigma(x-\bar{x})(y-\bar{y})}{\sqrt{\Sigma(x-\bar{x})^2 \Sigma(y-\bar{y})^2}}$$

→ sum of the cross products of the deviations of the two variables from their respective means.

→ Denominator includes separate calculations for the sum of squares for each variable. These sums of squares are multiplied together, and then the square root of that product is taken.

All 3 summation symbols indicate that we are adding up all the observations in the sample.

IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

INFERENTIAL REASONING ABOUT CORRELATION

To examine the possible outcomes, an informal model of the population and random sampling can be created by sampling pairs of variables from a dataset. To achieve this, a fake "population" of values is created for each of the two variables and placed in a data frame.

```{r}
set.seed(223)               # Start with a random number seed
sugar <- rnorm(2500)              # Make two vectors of N=2500
coffee <- rnorm(2500)
coffeeDF <- data.frame(sugar, coffee)    # Put them in a data frame
nrow(coffeeDF)                   # Verifying 2500 rows of two variables
coffeeDF[sample(nrow(coffeeDF), 25), ]     # Generates one sample of n=25
```

	sugar <dbl>	coffee <dbl>
1912	1.7455466	2.5598282
2285	-0.9030516	-0.5528993
1825	0.7983780	-0.2773980
1082	1.8136144	-0.2660671
644	-2.0365594	-0.3458332
2241	-0.9502869	0.3740736
1965	1.6810605	0.1500966
642	0.4994116	0.6547802
151	1.0286828	0.5471201

The correlation between sugar and coffee in the fake population data should be verified to be near 0. The sample() function is used in a new way by randomly selecting 25 row numbers from 1 to 2,500 to create "mini" data frames. Pearson's r correlation can then be calculated from the sampled data sets.

```{r}
cor(coffeeDF[sample(nrow(coffeeDF), 25), ])
```

```
              sugar      coffee
sugar    1.00000000 -0.09071247
coffee -0.09071247  1.00000000
```

The section discusses how to obtain a correlation value between two variables from a correlation matrix. The matrix includes correlation values between each variable and itself and between pairs of variables. The focus is on extracting the value of interest, which is in the lower left corner of the matrix and reflects the correlation between two specific variables. It is noted that the same correlation

value may appear in two places in the matrix, but it is not necessary to extract both values.

```r
cor(coffeeDF[sample(nrow(coffeeDF), 25), ])[1,2]
```

```
[1] -0.04505274
```

The code is used to simulate the Pearson's correlation coefficient between two variables (sugar and coffee) in a data frame. The output is a correlation matrix containing the correlation values between each pair of variables, including the correlation of a variable with itself. The output is filtered to extract the correlation value between the two variables of interest (wood and heat). The process is then repeated multiple times to observe the range of possible outcomes.

BOX ON P.129: READING A CORRELATION MATRIX

The author explains that instead of getting a single correlation coefficient, providing a data frame to the cor() function generates a square table with multiple correlation coefficients. They then suggest generating a correlation matrix with more than two variables to make sense out of the result.

```r
cor(iris[,1:4])
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Sepal.Length	1.0000000	-0.1175698	0.8717538	0.8179411
Sepal.Width	-0.1175698	1.0000000	-0.4284401	-0.3661259
Petal.Length	0.8717538	-0.4284401	1.0000000	0.9628654
Petal.Width	0.8179411	-0.3661259	0.9628654	1.0000000

The "cor()" command to generate a correlation matrix for the first four columns of the iris dataset, which has measurements of 150 iris flowers. The matrix is square with one diagonal containing all 1s, indicating perfect correlation between a variable and itself. There are two triangles of correlation data, one above and one below the diagonal, which contain the same information. The precision of the correlation values is given to seven decimal places, but in social science research, leaving off the leading zero is common practice. The resulting correlation matrix may be reported differently depending on the journal or report.
A truncated correlation matrix is easier to read as it eliminates the diagonal and the leading zero. They focus on larger correlations, such as the negative relationship of Sepal.Width with other variables and the high correlation between

Petal.Length and Petal.Width. Researchers may also report the results of a null hypothesis significance test on each correlation coefficient, marked with one to three asterisks indicating the level of significance. To test a whole matrix of correlations, the "corr.test()" function in the "psych" package can be used.

NULL HYPOTHESIS TESTING ON THE CORRELATION

In the previous section, was discussed the sampling distribution of correlations and how to build a simulated confidence interval around a sample-based estimate of the correlation coefficient. The standard approach to testing the significance of the correlation coefficient assumes a null hypothesis of rho = 0. The "cor.test()" function in R provides a simple procedure for conducting a null hypothesis test on a correlation coefficient.

```
```{r}
set.seed(223)
sugar <- rnorm(25)
coffee <- rnorm(25)
cor.test(sugar,coffee)
```
```

```
        Pearson's product-moment correlation

data:  sugar and coffee
t = 0.149, df = 23, p-value = 0.8829
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.3685997  0.4210189
sample estimates:
       cor
0.03105406
```

The section discusses the procedure for testing the significance of the correlation coefficient using the cor.test() function in R. A small random sample of n=25 observations is used to explain the process. The output of the cor.test() function includes a conventional null hypothesis test with an assumption of rho = 0 and a 95% confidence interval around the point estimate of r. The output also includes the test statistic, t-value, degrees of freedom, and the corresponding probability value. The p-value is compared to the conventional threshold of p<.05 to evaluate the result. The confidence interval is defined as the range of values that would contain the true population value of rho if the sampling process were repeated many times. The confidence interval for rho is reported to range from -0.36 to 0.42. The wide range of the confidence interval indicates the small sample size used.

IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

BAYESIAN TESTS ON THE CORRELATION COEFFICIENT

The BayesFactor package can be used to create a custom function for Bayesian testing. The code for creating this function is also provided below:

```{r}
#install.packages("BayesFactor")
library("BayesFactor")
bfCorTest <- function (x,y) # Get r from BayesFactor
{
zx <- scale(x)                          # Standardize X
zy <- scale(y)                          # Standardize Y
zData <- data.frame(x=zx,rhoNot0=zy)     # Put in a data frame
bfOut <- generalTestBF(x ~ rhoNot0, data=zData)    # linear coefficient
mcmcOut <- posterior(bfOut,iterations=10000)       # posterior samples
print(summary(mcmcOut[,"rhoNot0"]))      # Show the HDI for r
return(bfOut)                          # Return Bayes factor object
}
```

The code above creates a custom function in R called bfCorTest() which can be used to test correlations in a Bayesian framework. The function takes two arguments, x and y, which should be two variables with the same number of observations. The function will report a point estimate and a 95% HDI for the correlation coefficient from the posterior population distribution, as well as a BayesFactor object representing the odds in favor of the alternative hypothesis that the population correlation coefficient is not equal to 0. This function can be used on any two variables in R.

```{r}
set.seed(223)
sugar <- rnorm(25)
coffee <- rnorm(25)
bfCorTest(sugar,coffee)
```

IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

CAREGORICAL ASSOCIATIONS

This means that in order to determine whether there is a statistically significant association between the two categorical variables, we need to compare the observed frequencies in the contingency table to the expected frequencies under the assumption of independence. The way we do this is by calculating a chi-squared statistic, which measures the discrepancy between the observed and expected frequencies, and comparing it to a distribution of chi-squared values that would be expected under the null hypothesis of independence.

If the chi-squared value we calculate from our contingency table is larger than the critical value from the distribution, then we reject the null hypothesis and conclude that there is a statistically significant association between the two variables. Conversely, if the chi-squared value is smaller than the critical value, we fail to reject the null hypothesis and conclude that there is no evidence of an association.

To illustrate this procedure, let's return to our toast example. We can calculate the expected frequencies for each cell in the contingency table using the formula we discussed earlier. For example, the expected frequency for the jelly-down cell is (30 * 50)/100 = 15, since there are 30 jelly toast drops in total and 50 total drops overall. Similarly, the expected frequency for the butter-up cell is (70 * 50)/100 = 35. We can use these expected frequencies to calculate the chi-squared statistic for our table.

The formula for the chi-squared statistic is:

chi-squared = sum((observed - expected)^2 / expected)

where the sum is taken over all cells in the contingency table, observed is the observed frequency for a cell, and expected is the expected frequency for that cell. Applying this formula to our toast example, we get:

chi-squared = ((20 - 15)^2 / 15) + ((10 - 15)^2 / 15) + ((35 - 40)^2 / 40) + ((35 - 30)^2 / 30) = 4.17

To determine whether this value is statistically significant, we need to compare it to the distribution of chi-squared values that would be expected under the null hypothesis of independence. This distribution depends on the degrees of freedom, which for a 2x2 contingency table is 1. We can use a chi-squared table or calculator to find the critical value for our desired level of significance (e.g., 0.05).

IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

Assuming a significance level of 0.05, the critical value for 1 degree of freedom is 3.84. Since our calculated chi-squared value of 4.17 is larger than the critical value of 3.84, we reject the null hypothesis of independence and conclude that there is a statistically significant association between topping type and landing result.

In summary, when we have two categorical variables and want to determine whether there is an association between them, we use a chi-squared test of independence. This involves comparing the observed frequencies in a contingency table to the expected frequencies under the assumption of independence, and calculating a chi-squared statistic to measure the discrepancy between them. We then compare this statistic to a distribution of chi-squared values to determine whether it is statistically significant.

EXPLORING THE CHI-SQUARE DISTRIBUTION WITH A SIMULATION

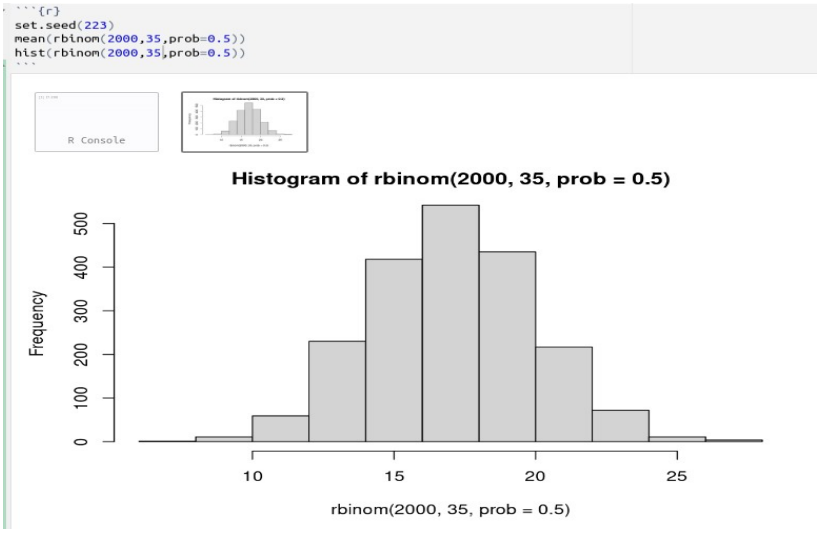
How to measure the difference between a sampled contingency table and the expected values table. They propose calculating the square of the difference between the actual value and the expected value for each cell, dividing the squared difference by the expected value, and summing the results to obtain the "chi-square" statistic. The author also mentions Karl Pearson's influence on the development of the chi-square test. Finally, the author suggests using an R function to create a 2x2 contingency table.

```r
# The user supplies the count for the upper-left
make2x2table <- function(ul)
{
ll
<- 50 - ul     # Calculate the lower-left cell
ur <- 30 - ul    # Calculate the upper-right cell
lr <- 50 - ur    # Calculate the lower-right cell
# Put all of the cells into a 2 x 2 matrix
matrix(c(ul,ur,ll,lr), nrow=2, ncol=2, byrow=TRUE)
}
```

Function in R that takes an actual and an expected values matrix as arguments, subtracts the contents of one matrix from another, squares the results, and normalizes them by dividing by the expected value in each cell. This function is used to measure the variation between the sampled contingency table and the expected values table, which is called "chi-square". The author also retests the minimum, expected, and maximum frequencies for the upper-left cell before moving on to sampling.

IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

How to use R to create a contingency table with expected values and calculate chi-squared values. They then demonstrate how to test extreme values for a specific cell in the table and show that they produce the same chi-squared value. The author then uses the binomial distribution to simulate different jelly-down counts and generate an empirical distribution. They calculate the mean of the distribution to check that it centers on 15, which is the expected value under the null hypothesis of no association.



The author explains how to use R to create a contingency table with expected values and calculate chi-squared values. They then demonstrate how to test extreme values for a specific cell in the table and show that they produce the same chi-squared value. The author then uses the binomial distribution to simulate different jelly-down counts and generate an empirical distribution. They calculate the mean of the distribution to check that it centers on 15, which is the expected value under the null hypothesis of no association.

THE CHI-SQUARE TEST WITH REAL DATA

Use of the ftable() function to flatten contingency data into a table for input into a chi-square test. The "Titanic" data set, which contains information about passengers and their survival status, is used as an example. The author extracts a split of survivors and nonsurvivors by gender using ftable() and performs a chi-square test to determine if there is independence between gender and survival status.

IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

```r
badBoatMF <- ftable(Titanic, row.vars=2, col.vars="Survived")
badBoatMF
chisq.test(badBoatMF, correct=FALSE)
```

```
        Survived   No  Yes
Sex
Male             1364  367
Female            126  344

        Pearson's Chi-squared test

data:  badBoatMF
X-squared = 456.87, df = 1, p-value < 2.2e-16
```

ftable() function to extract a split of survivors and non-survivors by gender from the Titanic dataset. The contingency table created from this split is used as input to the chi-square test, and the resulting p-value is well below the standard alpha level of 0.05. Therefore, the null hypothesis of independence between gender and survival status is rejected, indicating that these two factors are not independent, and the proportion of survivors among males was considerably lower than the proportion of survivors among females.

THE BAYESIAN APPROACH TO THE CHI-SQUARE TEST

The Bayesian approach to the chi-square test involves calculating the posterior probability of the null hypothesis being true, given the data. This is done using Bayes' theorem, which states that the posterior probability of a hypothesis is proportional to the likelihood of the data given the hypothesis, multiplied by the prior probability of the hypothesis.

To apply this to the chi-square test, we first need to specify a prior distribution for the parameters of interest (e.g. the expected frequencies in each cell of the contingency table). This can be done using subjective or objective methods, depending on the situation.

Next, we calculate the likelihood of the data given the parameters under the null hypothesis (i.e. assuming no association between the variables of interest). This is done using the same formula as in the classical approach to the chi-square test.

Finally, we use Bayes' theorem to calculate the posterior probability of the null hypothesis being true, given the data. This can be used to make decisions about whether to reject or accept the null hypothesis.

IST772 Summary Template: Chapter 7 – Associations between Variables
Originality Assertion: By submitting this file you affirm that this writing is your own.

The Bayesian approach to the chi-square test has several advantages over the classical approach, including the ability to incorporate prior knowledge and uncertainty into the analysis, and the ability to calculate the probability of the null hypothesis being true, rather than just the probability of observing the data under the null hypothesis. However, it can also be more computationally complex and requires careful specification of the prior distribution.