Geog 205

Contingency Tables, Covariance & Correlation

Contingency Tables

1. In this exercise, we explore some simple forms of bivariate relationships using R.

2. Set your working directory and read in the file arthritis.csv from the Datasets link under Module 2.

```
> setwd("")
> arthritis <- read.csv("arthritis.csv", sep=",", header=T)
> str(arthritis)
'data.frame':     84 obs. of  5 variables:
$ ID      : int  57 46 77 17 36 23 75 39 33 55 ...
$ Treatment: chr  "Treated" "Treated" "Treated" "Treated" ...
$ Sex     : chr  "Male" "Male" "Male" "Male" ...
$ Age     : int  27 29 30 32 46 58 59 59 63 63 ...
$ Improved : chr  "Some" "None" "None" "Marked" ...
```

This file contains 84 observations on a series of variables. The variables ID and age are numeric, while those on Treatment, Sex and Improved are categorical. The observations are individual patients taking part in some clinical trials involving a new drug to treat arthritis. The treatment variable is binary (Placebo, Treated), sex is binary (Male, Female) and Improved takes 3 values (None, Some, Marked).

3. You have already seen how to generate simple frequency counts using the table() function. Here we look at observation counts for the variable Improved (this is known as a one-way table)

```
> mytable <- with(arthritis, table(Improved))
> mytable
Improved
Marked   None   Some
  28     42     14
```

4. You can turn these absolute frequencies into relative frequencies (proportions) as

```
> prop.table(mytable)
Improved
  Marked     None     Some
0.3333333 0.5000000 0.1666667
```

5. You can create two-way tables (contingency tables) using the table or xtabs functions

```
> mytable2 <- table(arthritis$Treatment, arthritis$Improved)
> mytable2
```

	Marked	None	Some
Placebo	7	29	7
Treated	21	13	7

Or using xtabs, where A and B represent variables and mydata is a matrix or data frame
> mytable2b <- xtabs(~ A + B, data=mydata). So let's try this

```
> mytable2b <- xtabs(~ Treatment + Improved, data=arthritis)
> mytable2b
          Improved
Treatment Marked None Some
  Placebo      7   29    7
  Treated     21   13    7
```

6. We can generate relative frequencies by rows (1) and columns (2) of this two-way table

```
> prop.table(mytable2, 1)

             Marked       None       Some
  Placebo 0.1627907 0.6744186 0.1627907
  Treated 0.5121951 0.3170732 0.1707317
```

Here (1) asks for frequencies by row, so we can see the row proportions sum to 1.0, and the results show that 51% of patients in the treatment group responded markedly compared with only 16% in the placebo group.

For column frequencies

```
> prop.table(mytable2, 2)

             Marked       None       Some
  Placebo 0.2500000 0.6904762 0.5000000
  Treated 0.7500000 0.3095238 0.5000000
```

7. And you can make all this a little easier to understand using the addmargins function

```
> addmargins(prop.table(mytable2))

        Marked     None     Some      Sum
 Placebo 0.08333333 0.34523810 0.08333333 0.51190476
 Treated 0.25000000 0.15476190 0.08333333 0.48809524
 Sum     0.33333333 0.50000000 0.16666667 1.00000000
```

8. It is easy to build multi-way tables usinig the table and xtabs functions.

```
> mytable3b <- xtabs(~ Treatment + Sex + Improved, data=arthritis)
> mytable3b
, , Improved = Marked

          Sex
Treatment Female Male
  Placebo      6    1
  Treated     16    5
```

```
, , Improved = None

         Sex
Treatment Female Male
  Placebo     19   10
  Treated      6    7

, , Improved = Some

         Sex
Treatment Female Male
  Placebo      7    0
  Treated      5    2
```

Covariance and Correlation

1. R can generate a variety of measures of covariance and correlation, or measures of the linear association between variables. It can generate the Pearson's product moment correlation for interval and ratio data and the Spearman rank correlation for ordinal data.

2. The cov() function provides covariances and the cor() function correlations. Let's explore using the state.x77 dataset that is in the base version of R (hence no libraries need to be added).

3. The dataset state.x77 provides information on population, income, illiteracy, life expectancy, murder rates etc for the 50 US states in 1977. Learn more about the dataset from

```
> help(state.x77)
```

4. Too make life easier, let's convert the matrix into a dataframe and attach it

```
> states <- as.data.frame(state.x77)
> attach(states)
```

Two variables "Life Exp" and "HS Grad" are still a little tricky to deal with. Let's sort that out by creating two new objects for these variables. Note that the quote marks for the terms on the right side of the assignment operator come from the top left key on most keyboards (under the ~, just left of the number 1 key)

```
> Lifeexp <- `Life Exp`
> HSGrad <- `HS Grad`
```

5. Now we should be good to go. Let's generate the sample covariance between Income and Lifeexp. You would expect this to be positive, right?

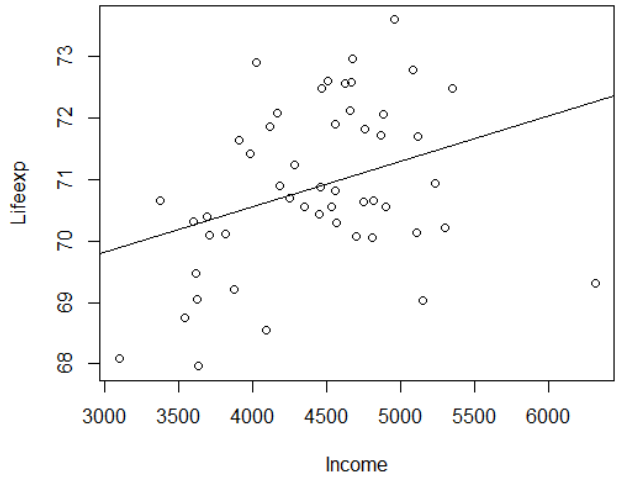
```
> cov(Income, Lifeexp)
[1] 280.6632
```

You can see the relationship is positive, but you can also see the difficulty of interpreting the scale of the covariance measure. So let's use the correlation coefficient and generate a plot

```
> cor(Income, Lifeexp)
[1] 0.3402553
```

The correlation coefficient is always between -1 and 1. Here the value is not much greater than 0, so this is evidence of a weak positive relationship. Let's check the plot and add a regression line

```
> plot(Income, Lifeexp)
> abline(lm(Lifeexp~Income))
```



6. The regression line has a positive slope and that indicates that the linear relationship between Income and Lifeexp is positive as the correlation coefficient suggests. For now don't worry too much about what a regression line is and how it might be fitted. Just remember that it provides an indicator of the nature of the relationship between the two variables, just like the correlation coefficient.

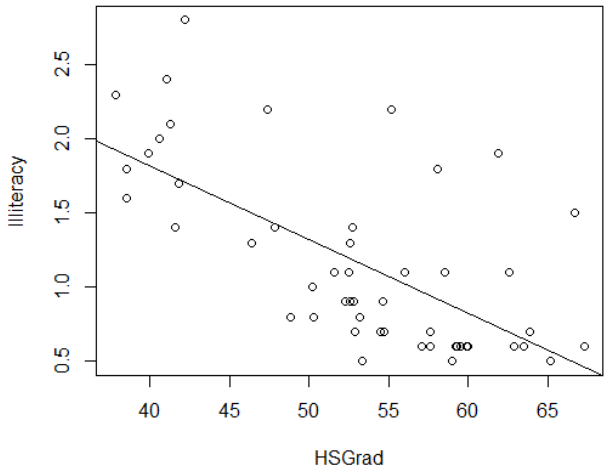
7. Now you should find the sample correlation coefficient the "old fashioned" way. Try and follow the logic here

```
> meanle <- mean(Lifeexp)
> meaninc <- mean(Income)
> incdevtn <- Income-meaninc
> ledevtn <- Lifeexp-meanle
> devproduct <- incdevtn*ledevtn
> numerator <- sum(devproduct)
> numerator
[1] 13752.5

> incdevsq <- incdevtn^2
> ledevsq <- ledevtn^2
> incssdev <- sum(incdevsq)
> lessdev <- sum(ledevsq)
> incsd <- sqrt(incssdev)
> lesd <- sqrt(lessdev)
```

```
> denominator <- incsd*lesd
> denominator
[1] 40418.16

> corrltncoef <- numerator/denominator
> corrltncoef
[1] 0.3402553     ….. and check that this is the same as the value above!
```

8. Finally, what about a negative relationship between two variables? You might expect a negative relationship between high school graduation rates (HSGrad) and the illiteracy rate (Illiteracy)

```
> plot(HSGrad, Illiteracy)
> abline(lm(Illiteracy~HSGrad))
```



That looks about right. What about the correlation?

```
> cov(HSGrad,Illiteracy)
[1] -3.235469
> cor(HSGrad,Illiteracy)
[1] -0.6571886
```