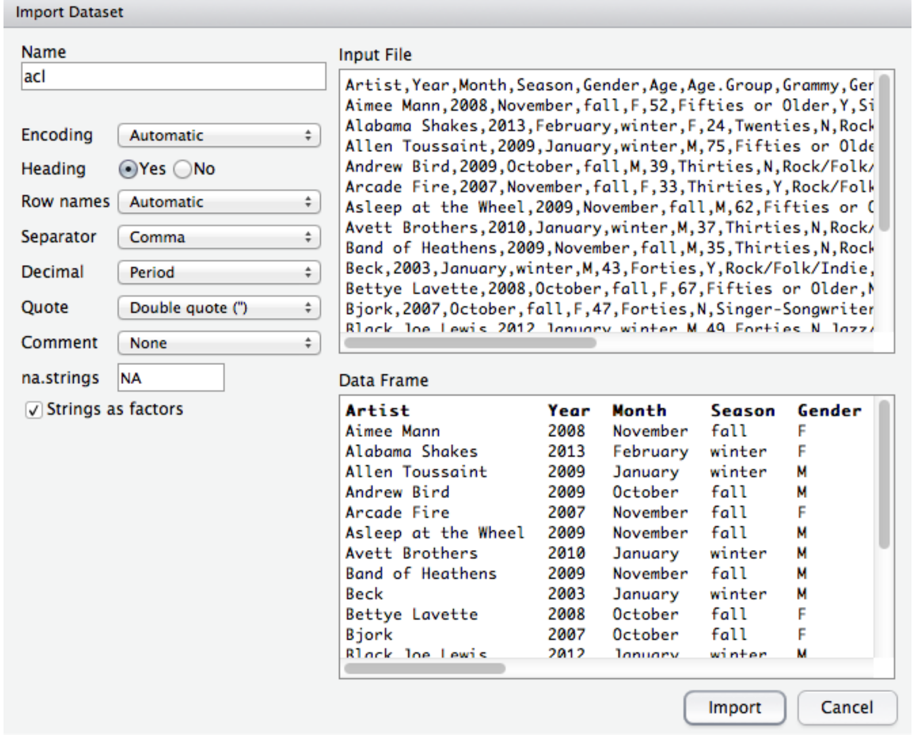
Foundations of Data Analysis

The University of Texas at Austin

R Tutorials: Week 4

Table Proportions

In this R tutorial, we're going to extract probabilities from a table of a categorical variable and then learn how we can calculate the marginal and conditional probabilities of categorical variables. So for this tutorial, we're going to be using the Austin City Limits data set. So I'm going to need to go ahead and import that from a text file. And if you're using this server, you'll see this file in your data folder. I've got it saved on my desktop, so I'm going to go ahead and click Open. And let's just shorten the name up a little bit and call it acl, all lowercase. I do have variable names in my Header row, so I'm going to change that to yes. And then hit Import.



Import Dataset

Name: acl

Encoding: Automatic

Heading: ● Yes ○ No

Row names: Automatic

Separator: Comma

Decimal: Period

Quote: Double quote (")

Comment: None

na.strings: NA

☑ Strings as factors

Input File

```
Artist,Year,Month,Season,Gender,Age,Age.Group,Grammy,Ger
Aimee Mann,2008,November,fall,F,52,Fifties or Older,Y,Si
Alabama Shakes,2013,February,winter,F,24,Twenties,N,Rock
Allen Toussaint,2009,January,winter,M,75,Fifties or Olde
Andrew Bird,2009,October,fall,M,39,Thirties,N,Rock/Folk/
Arcade Fire,2007,November,fall,F,33,Thirties,Y,Rock/Folk
Asleep at the Wheel,2009,November,fall,M,62,Fifties or (
Avett Brothers,2010,January,winter,M,37,Thirties,N,Rock/
Band of Heathens,2009,November,fall,M,35,Thirties,N,Rock
Beck,2003,January,winter,M,43,Forties,Y,Rock/Folk/Indie,
Bettye Lavette,2008,October,fall,F,67,Fifties or Older,N
Bjork,2007,October,fall,F,47,Forties,N,Singer-Songwriter
Black Joe Lewis,2012,January,winter,M,49,Forties,N,Jazz/
```

Data Frame

```
Artist               Year   Month      Season   Gender
Aimee Mann           2008   November   fall     F
Alabama Shakes       2013   February   winter   F
Allen Toussaint      2009   January    winter   M
Andrew Bird          2009   October    fall     M
Arcade Fire          2007   November   fall     F
Asleep at the Wheel  2009   November   fall     M
Avett Brothers       2010   January    winter   M
Band of Heathens     2009   November   fall     M
Beck                 2003   January    winter   M
Bettye Lavette       2008   October    fall     F
Bjork                2007   October    fall     F
Black Joe Lewis      2012   January    winter   M
```

Figure 1: Importing Data in RStudio.

This data set contains various information about artists performing on the show, ACL Live. We have information on the gender of the front person of the band, or artist, their age, whether or not they've won a Grammy, and several other variables. So in this video, we're going to be looking at the variables, Gender, and also whether or not they've won a Grammy.

So we've already learned how to get the counts across a categorical variable by using the "**table()**" function. So if we want to see, say, the distribution across this Grammy variable – whether or not these artists have won a Grammy – we could just ask for a table of acl$Grammy.

```
table(acl$Grammy)
```

```
##
##  N  Y
## 67 49
```

And it gives us all possible values in that variable, which is no and yes, and then the counts of each. So we can see there are 67 non-Grammy winners and 49 Grammy winners in our 116 artists. Now our new function that can help us calculate the proportion across each of these categories is called "**prop.table()**". And for this function to work, we need to give it a table object. So just to make our lives a little bit easier, I'm going to go ahead up here and call this table of the Grammy variable, let's say *gtab* and assign it.

```
gtab <- table(acl$Grammy)
```

If I run this line, this *gtab* object is now going to be assigned a table of these counts. Then I can pass gtab into my "**prop.table()**" function, and it will calculate for me the proportion in each category.

```
prop.table(gtab)
```

```
##
##         N         Y
## 0.5775862 0.4224138
```

So remember, we have 116 artists in our data set. If I were to take 67 divided by the overall number, 116, I'm going to get the proportion in the no category. And it should match this 0.577, and it does. Likewise, if I take 49 divided by 116, I get the 0.422 that showed up in yes of our **prop.table()**. Where **prop.table()** can be a little bit more useful is when we give it a contingency table, or a table with 2 categorical variables.

So let's say, instead of looking at the overall distribution of Grammy winners, which would be the marginal distribution, say, of Grammy winners, we can maybe look within each gender of artists to see how many Grammy winners there were. So I'm going to make a new table, gtab2. And it's going to be, again, a table, and now instead of just giving it 1 categorical variable, Grammy, I'm going to separate another one with a comma. And this will produce a contingency table assigned to the object name, gtab2.

```
gtab2 <- table(acl$Grammy, acl$Gender)
```

So I'm going to Submit that. And we can look at gtab2 and see what it gives us.

```
gtab2
```

```
##
##      F  M
##   N 21 46
##   Y 14 35
```

So now, instead of just yes and no, we've got males and females across the columns, Grammy yes or no across the rows. And so we could see that there are 21 female non-Grammy winners in our data set. Similarly, there are 35 male Grammy winners. So this table of counts gives us a nice snapshot of the distribution across these two categorical variables, but it'll be nice to see the proportions within a certain category of the other variable. So in order to do that, we're going to utilize **prop.table()** again. So we'll ask for "**prop.table**" and give it our gtab2. And let's see what that gives us.

```r
prop.table(gtab2)
```

```
##
##             F         M
##   N 0.1810345 0.3965517
##   Y 0.1206897 0.3017241
```

So what the default **prop.table()** is going to give you is the proportion of the 116 overall observations that fall within each of these cells. So just to confirm that, I know there are 21 females who have not won a Grammy. If I divide 21 by 116, I'm going to get this 0.181 that I found here. So this could be useful.

But usually, we want to take the proportions either by row or by column to give us some more meaningful information about our data. And I can do that with just an option here. After my table object, I'm going to include a comma and then give it either a 1 or 2. Let's see what the 1 gives us.

```r
prop.table(gtab2, 1)
```

```
##
##             F         M
##   N 0.3134328 0.6865672
##   Y 0.2857143 0.7142857
```

Here, we can see that, if we look across the rows, we're going to get proportions that add up to 1, or 100%. So we can say within the no Grammy row, 31% of observations were females, 68% were males. Likewise, across the yes Grammy category, we can see the conditional distribution of gender. All right, so we're conditioning on what Grammy category they were. And then we're looking at the proportions of males and females. So these are conditional probabilities.

There's another type of conditional probability we can ask for. And to get that, we'll change the 1 to a 2.

```r
prop.table(gtab2, 2)
```

```
##
##             F         M
##   N 0.6000000 0.5679012
##   Y 0.4000000 0.4320988
```

Now, we're calculating the proportions across each column. So of all the females in the data set, 60% have not won a Grammy, 40% have. Likewise, across all the males in the data set, about 57% have not won a Grammy and 43% percent have.