

Getting the data into R

- Put frequencies directly into tables
- Read a data frame with frequencies, number of rows = number of cells
- Read a raw data file, number of rows = n

Put frequencies directly into tables

Table 3-2

Counts for Structural Habitat Categories for *Anolis* Lizards of Bimini: *sagrei* Adult Males vs. *angusticeps* Adult Males (Schoener [1968])

(a) Observed data

		perch diameter (inches)			
		<i>sagrei</i>		<i>angusticeps</i>	
		≤ 2.5	> 2.5	≤ 2.5	> 2.5
perch height (feet)	> 5.0	15	18	21	1
	≤ 5.0	48	84	3	2

```

> lizards <- numeric(8); lizards
[1] 0 0 0 0 0 0 0 0
> dim(lizards) <- c(2,2,2) # Now a 2x2x2 table: Rows, cols, layers
> lizards
, , 1

      [,1] [,2]
[1,]    0    0
[2,]    0    0

, , 2

      [,1] [,2]
[1,]    0    0
[2,]    0    0

> # 1 = Perch Height, 2 = Perch Diameter, 3 = Species
> lizards[, ,1] <- rbind( c(15,18),
+                          c(48,84) )
> lizards[, ,2] <- rbind( c(21,1),
+                          c(3,2) )

```

Need Labels

```
> lizards
```

```
, , 1
```

	[,1]	[,2]
[1,]	15	18
[2,]	48	84

```
, , 2
```

	[,1]	[,2]
[1,]	21	1
[2,]	3	2

Labels are dimnames of the array: A list

```
> # Labels are dimnames: A list
> lizlabels <- list() # An empty list
> lizlabels$Height <- c("gt 5.0", "le 5.0")
> lizlabels$Diameter <- c("le 2.5", "gt 2.5")
> lizlabels$Species <- c("Sagrei", "Angusticeps")
> lizlabels
$Height
[1] "gt 5.0" "le 5.0"

$Diameter
[1] "le 2.5" "gt 2.5"

$Species
[1] "Sagrei"          "Angusticeps"
```

This is Better

```
> dimnames(lizards) <- lizlabels ; lizards
, , Species = Sagrei
```

```
      Diameter
Height le 2.5 gt 2.5
      gt 5.0      15      18
      le 5.0      48      84
```

```
, , Species = Angusticeps
```

```
      Diameter
Height le 2.5 gt 2.5
      gt 5.0      21      1
      le 5.0       3      2
```

Table 3-2

Counts for Structural Habitat Categories for *Anolis* Lizards of Bimini: *sagrei* Adult Males vs. *angusticeps* Adult Males (Schoener [1968])

(a) Observed data

		perch diameter (inches)			
		<i>sagrei</i>		<i>angusticeps</i>	
		≤ 2.5	> 2.5	≤ 2.5	> 2.5
perch height (feet)	> 5.0	15	18	21	1
	≤ 5.0	48	84	3	2

margin.table

```
> species_by_height = margin.table(lizards,margin=c(3,1))
```

```
> species_by_height
```

```
      Height
Species  gt 5.0 le 5.0
  Sagrei      33    132
Angusticeps  22     5
```

```
> spec_by_height_by_diam = margin.table(lizards,margin=c(3,1,2))
```

```
> spec_by_height_by_diam
```

```
, , Diameter = le 2.5
```

```
      Height
Species  gt 5.0 le 5.0
  Sagrei      15    48
Angusticeps  21     3
```

```
, , Diameter = gt 2.5
```

```
      Height
Species  gt 5.0 le 5.0
  Sagrei      18    84
Angusticeps   1     2
```

Method 2: Read a data frame

```
> # See what a data frame looks like  
> as.data.frame.table(lizards)
```

	Height	Diameter	Species	Freq
1	gt 5.0	le 2.5	Sagrei	15
2	le 5.0	le 2.5	Sagrei	48
3	gt 5.0	gt 2.5	Sagrei	18
4	le 5.0	gt 2.5	Sagrei	84
5	gt 5.0	le 2.5	Angusticeps	21
6	le 5.0	le 2.5	Angusticeps	3
7	gt 5.0	gt 2.5	Angusticeps	1
8	le 5.0	gt 2.5	Angusticeps	2

Read data frame from an external file, say a plain text file

	Admit	Gender	Dept	Freq
1	Admitted	Male	A	512
2	Rejected	Male	A	313
3	Admitted	Female	A	89
4	Rejected	Female	A	19
5	Admitted	Male	B	353
6	Rejected	Male	B	207
7	Admitted	Female	B	17
8	Rejected	Female	B	8
9	Admitted	Male	C	120
10	Rejected	Male	C	205
11	Admitted	Female	C	202
12	Rejected	Female	C	391
13	Admitted	Male	D	138
14	Rejected	Male	D	279
.
.
.

Berkeley data are on the class website

```
> berkeley <- read.table("http://www.utstat.toronto.edu/~brunner/312f12/  
code_n_data/BerkeleyFrame.txt")
```

```
>
```

```
> berkeley
```

	Admit	Gender	Dept	Freq
1	Admitted	Male	A	512
2	Rejected	Male	A	313
3	Admitted	Female	A	89
4	Rejected	Female	A	19
5	Admitted	Male	B	353
6	Rejected	Male	B	207
7	Admitted	Female	B	17
8	Rejected	Female	B	8
9	Admitted	Male	C	120
10	Rejected	Male	C	205

```
etc.
```

`xtabs(Counts ~ Vars separated by + signs, data =
Name of data frame)`

```
> UCB <- xtabs(Freq ~ Dept + Admit + Gender, data = berkeley); UCB  
, , Gender = Female
```

```
      Admit  
Dept Admitted Rejected  
A      89      19  
B      17       8  
C     202     391  
D     131     244  
E      94     299  
F      24     317
```

```
, , Gender = Male
```

```
      Admit  
Dept Admitted Rejected  
A     512     313  
B     353     207  
C     120     205  
D     138     279  
E      53     138  
F      22     351
```

Marginal Tables are Easy

```
> GenderAdmit <- xtabs(Freq ~ Gender + Admit, data = berkeley)
> GenderAdmit
```

```
      Admit
Gender  Admitted Rejected
Female      557     1278
Male      1198     1493
```

```
> prop.table(GenderAdmit,1)
      Admit
Gender  Admitted Rejected
Female 0.3035422 0.6964578
Male   0.4451877 0.5548123
```

Probably a data frame and `xtabs` is the easiest way to import data from a published table, if the table has more than 2 dimensions.

Read data into a data frame, Number of rows = n

```
> florida <- read.table(file='http://www.utstat.toronto.edu/~brunner/312f12/  
code_n_data/deathpen.data', col.names=c('Prace', 'Vrace', 'DeathPen'))
```

```
> florida
```

	Prace	Vrace	DeathPen
1	1	1	1
2	1	1	1
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1
.	.	.	.
.	.	.	.
.	.	.	.

The table function

```
> PR_by_DP = table(florida$Prace, florida$DeathPen); PR_by_DP
```

```
      1    2
1  19 141
2  17 149
```

```
> # This is pretty hard to read. We can do better by creating factors.
```

```
> Prace <- factor(florida$Prace, labels=c('White','Black')) # In order 1,2
```

```
> Vrace <- factor(florida$Vrace, labels=c('White','Black'))
```

```
> DeathPen <- factor(florida$DeathPen, labels=c('Yes','No'))
```

```
> PR_by_DP = table(Prace, DeathPen); PR_by_DP
```

```
      DeathPen
Prace  Yes  No
White  19 141
Black  17 149
```

3-Dimensional table

```
> deathrow <- table(Prace, DeathPen, Vrace); deathrow  
, , Vrace = White
```

	DeathPen	
Prace	Yes	No
White	19	132
Black	11	52

```
, , Vrace = Black
```

	DeathPen	
Prace	Yes	No
White	0	9
Black	6	97

```
> prop.table(deathrow[, , 1], 1) # Proportions of row totals
```

	DeathPen	
Prace	Yes	No
White	0.1258278	0.8741722
Black	0.1746032	0.8253968

Fitting and testing models with the loglin function

- Hierarchical models only
- Very close to bracket notation
- Give it a table and a list of vectors
- Vectors are vars in a bracket, like `c(1,2,4)` means `[1 2 4]`
- Iterative proportional model fitting
- Returns estimated expected frequencies as an option

loglin(table,margin,fit=F,param=F)

```
> lizards  
, , Species = Sagrei
```

	Diameter	
Height	le 2.5	gt 2.5
gt 5.0	15	18
le 5.0	48	84

```
, , Species = Angusticeps
```

	Diameter	
Height	le 2.5	gt 2.5
gt 5.0	21	1
le 5.0	3	2

```
> lizmodel1 <- loglin(lizards,list(1,c(2,3))) # [1] [23]  
2 iterations: deviation 0
```

```
> lizmodel1
$lrt
[1] 43.87073

$spearson
[1] 47.46099

$df
[1] 3

$margin
$margin[[1]]
[1] "Height"

$margin[[2]]
[1] "Diameter" "Species"

> 1-pchisq(43.87073,df=3)
[1] 1.607684e-09
> 1-pchisq(lizmodel1$lrt,df=lizmodel1$df)
[1] 1.607688e-09
```

Some options

```
> lizmodel1b <- loglin(lizards,list('Height',c('Diameter','Species')),
+                       fit=T,param=T)
2 iterations: deviation 0
> lizmodel1b$lrt
[1] 43.87073
> # Same as before, of course
> lizmodel1b$fit # Estimated expected values
```

```
, , Species = Sagrei
```

```
      Diameter
Height  le 2.5  gt 2.5
      gt 5.0 18.04688 29.21875
      le 5.0 44.95312 72.78125
```

 $\hat{\mu}_{ij1}$

```
, , Species = Angusticeps
```

```
      Diameter
Height  le 2.5  gt 2.5
      gt 5.0  6.875 0.859375
      le 5.0 17.125 2.140625
```

 $\hat{\mu}_{ij2}$

Parameter estimates

```
> lizmodel1b$param
```

```
$ '(Intercept)'
```

```
[1] 2.467355
```

 $\hat{\lambda}$

```
$Height
```

```
gt 5.0 le 5.0
```

```
-0.4563239 0.4563239
```

 $\hat{\lambda}_1^X$
 $\hat{\lambda}_2^X$

```
$Diameter
```

```
le 2.5 gt 2.5
```

```
0.3994009 -0.3994009
```

 $\hat{\lambda}_1^Y$
 $\hat{\lambda}_2^Y$

```
$Species
```

```
Sagrei Angusticeps
```

```
1.122860 -1.122860
```

 $\hat{\lambda}_1^Z$
 $\hat{\lambda}_2^Z$

```
$Diameter.Species
```

```
Species
```

```
Diameter Sagrei Angusticeps
```

```
le 2.5 -0.6403199 0.6403199
```

```
gt 2.5 0.6403199 -0.6403199
```

 $\hat{\lambda}_{jk}^{YZ}$

Two more models

```
> # [1] [2] [3] Complete independence
> lizmodel2 <- loglin(lizards,list(1,2,3))
2 iterations: deviation 0
> lizmodel2$lrt; lizmodel2$df
[1] 70.07975
[1] 4

> # [1 2] [1 3] [2 3] All 2-way relationships
> lizmodel3 <- loglin(lizards,list(c(1,2),c(1,3),c(2,3)))
4 iterations: deviation 0.07225579
> lizmodel3$lrt; lizmodel3$df
[1] 2.706392
[1] 1
```