Writing Functions In R

Necessity Really is a Mother

Paul E. Johnson[1][2]

[1]University of Kansas, Department of Political Science [2]Center for Research Methods and Data Analysis

October 12, 2016

Outline

Outline

Entropy can summarize diversity for a categorical variable

- Entropy in Physics means disorganization
- Sometimes called Shannon's Information Index
- Basic idea. List the possible outcomes and their probabilities
- The amount of diversity in a collection of observations depends on the equality of the proportions of cases observed within each type.

A Reasonable Person Would Agree . . .

- This distribution is "less diverse"

outcome name	t1	t2	t3	t4	t5
prob(outcome)	0.1	0.3	0.05	0.55	0.0

- than this distribution:

outcome name	t1	t2	t3	t4	t5
prob(outcome)	0.2	0.2	0.2	0.2	0.2

The Information Index

- For each type, calculate the following information (or can I say "diversity"?) value

$$-p_t * log_2(p_t) \tag{1}$$

- Note that if $p_t = 0$, the diversity value is 0
- If $p_t = 1$, then diversity is also 0
- Sum those values across the m categories

$$\sum_{t=1}^{m} -p_t * log_2(p_t) \tag{2}$$

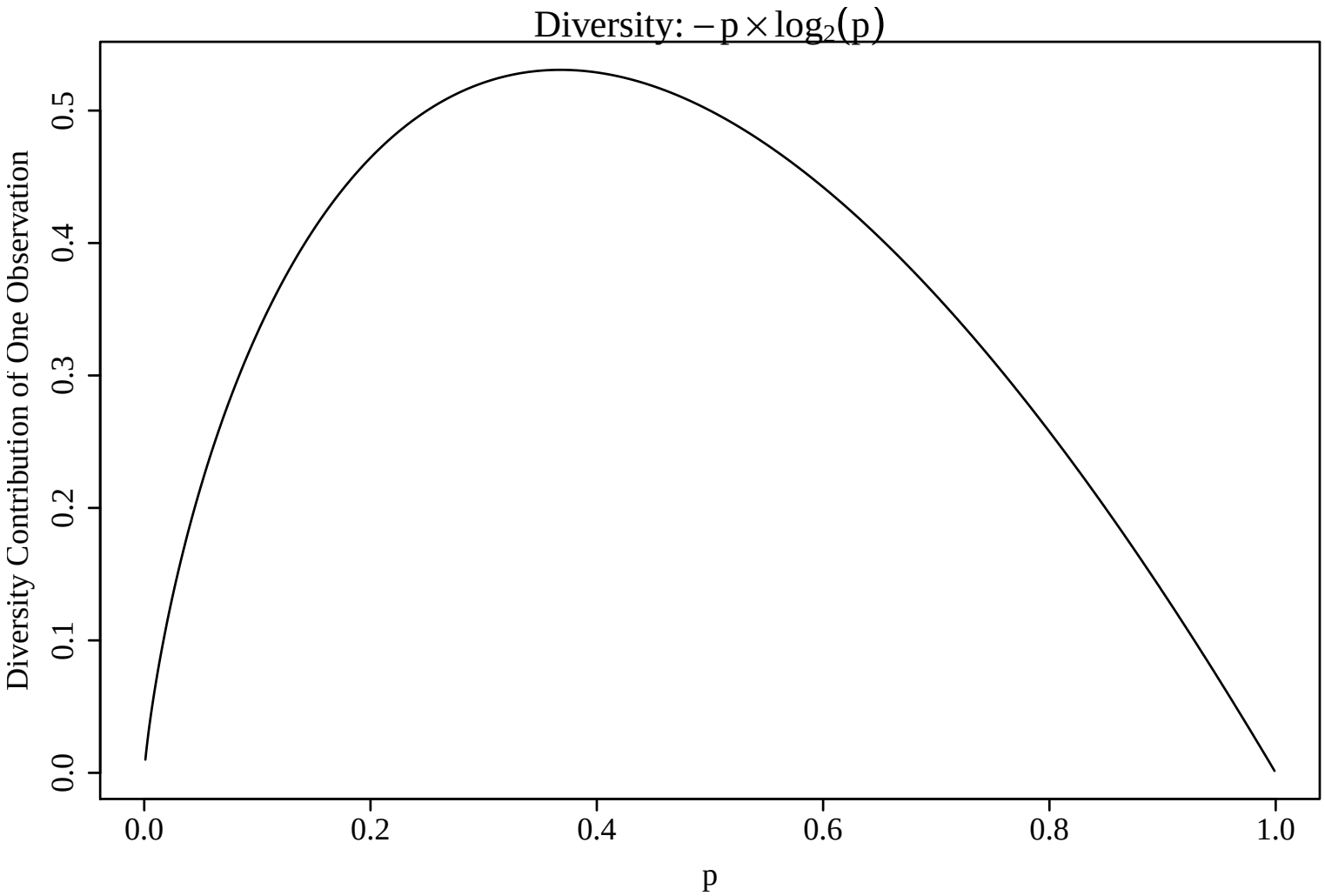
- Diversity is at a maximum when p_t are all equal

Calculate Diversity for One Type

```
divr <- function(p = 0){
  ifelse ( p > 0 & p < 1, -p * log2(p), 0)
}
```

Let's plot that

```
pseq <- seq(0.001, 0.999, length=999)
pseq.divr <- divr(pseq)
plot(pseq.divr ~ pseq, xlab = "p", ylab = "Diversity
    Contribution of One Observation", main = expression(
    paste("Diversity: ", -p %*% log[2](p))), type = "l")
```


Diversity: $-p \times \log_2(p)$
Diversity Contribution of One Observation
0.0
0.1
0.2
0.3
0.4
0.5
0.0
0.2
0.4
0.6
0.8
1.0
p

Diversity Function

- Define an Entropy function that sums those values

```
entropy <- function(p){
    sum( divr(p) )
}
```

- Calculate some test cases

```
entropy( c(1/5, 1/5, 1/5, 1/5, 1/5) )
```

```
[1] 2.321928
```

```
entropy( c(3/5, 1/5, 1/5, 0/5, 0/5) )
```

```
[1] 1.370951
```

There's a Little Problem With This Approach

- Diversity is sensitive to the number of categories
  8 equally likely outcomes (rep(x,y): repeats x y times.)

```
entropy(rep(1/8, 8))
```

```
[1] 3
```

  14 equally likely outcomes

```
entropy(rep(1/14, 14))
```

```
[1] 3.807355
```

- Write it out for a 3 category case

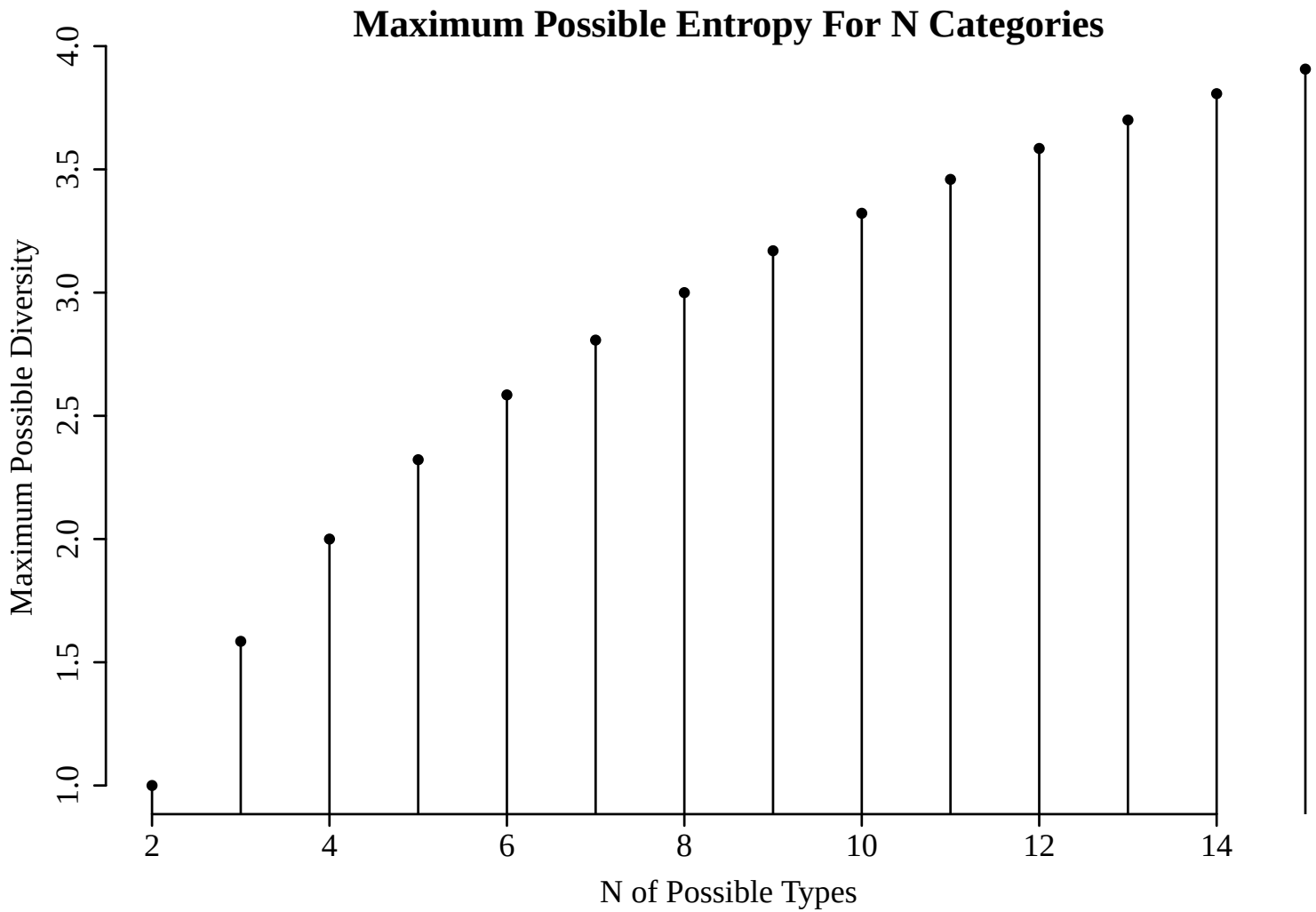
$$-\frac{1}{3}log_2(\frac{1}{3}) - \frac{1}{3}log_2(\frac{1}{3}) - \frac{1}{3}log_2(\frac{1}{3}) = -log_2(\frac{1}{3}) \qquad (3)$$

- The highest possible diversity with 3 types is $-log_2(\frac{1}{3})$
- The highest possible diversity for N types is $-log_2(\frac{1}{N})$

We Might As Well Plot That

```
maximumEntropy <- function(N) - log2(1/N)
Nmax <- 15
M <- 2:Nmax
plot(M, maximumEntropy(M), xlab = "N of Possible Types",
     ylab = "Maximum Possible Diversity", main = "Maximum
     Possible Entropy For N Categories", type = "h", axes =
     FALSE)
axis(1)
axis(2)
points(M, maximumEntropy(M), pch=19)
```

Maximum Entropy as a Function of the Number of Types



Final Result: Normed Entropy as a Diversity Summary

```
normedEntropy <- function(x) entropy(x)/ maximumEntropy(
    length(x))
```

Compare some cases with 4 possible outcomes

```
normedEntropy(c(1/4,1/4,1/4,1/4))
```

```
[1] 1
```

```
normedEntropy(c(1/2, 1/2, 0, 0))
```

```
[1] 0.5
```

```
normedEntropy(c(1, 0, 0, 0))
```

```
[1] 0
```

How about 7 types of outcomes:

```
normedEntropy(rep(1/7, 7))
```

```
[1] 1
```

```
normedEntropy((1:7)/(sum(1:7)))
```

```
[1] 0.9297027
```

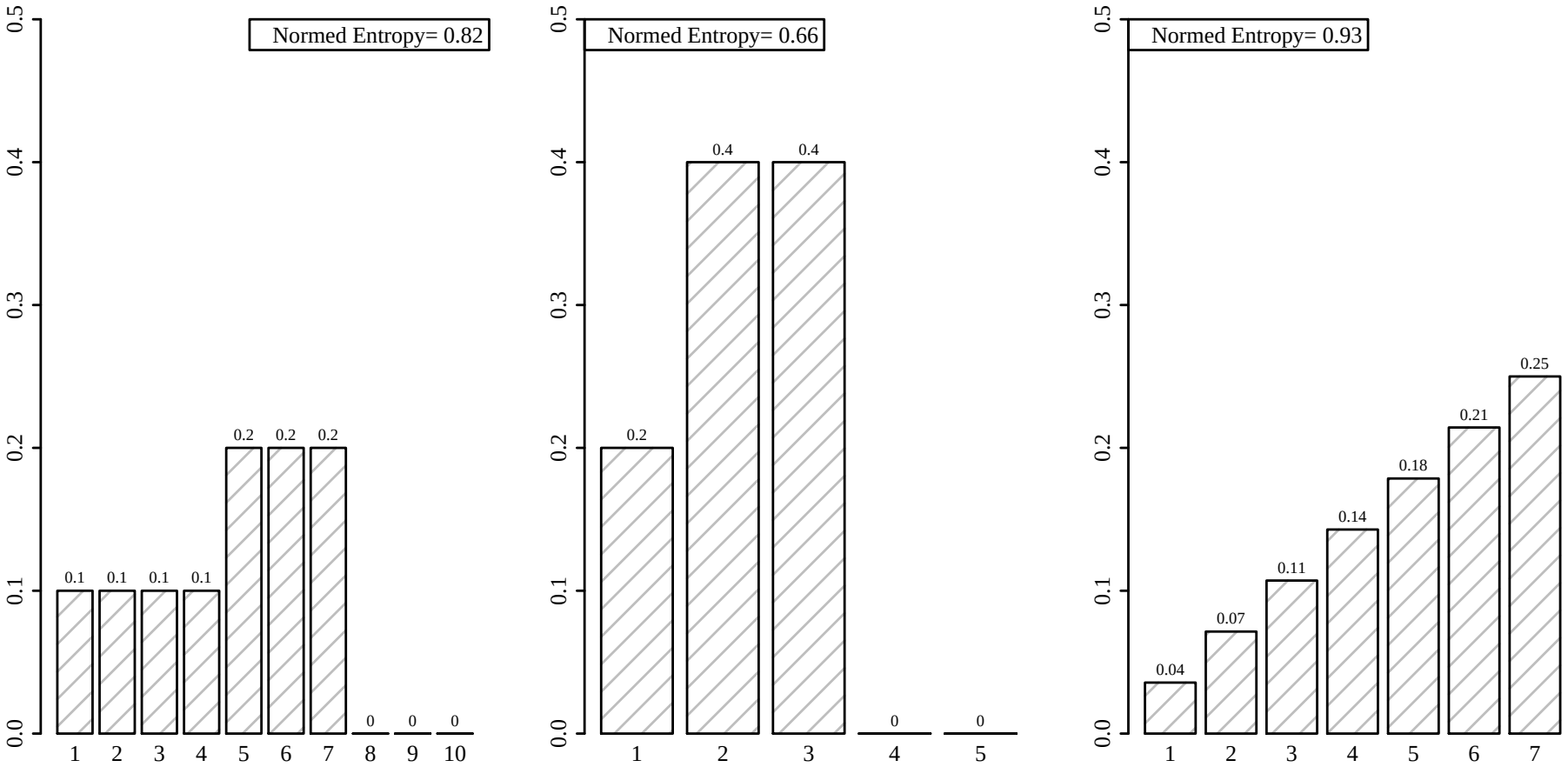
```
normedEntropy(c(2/7, 2/7, 3/7, 0, 0, 0, 0))
```

```
[1] 0.5544923
```

```
normedEntropy(c(5/7, 2/7, 0, 0, 0, 0, 0))
```

```
[1] 0.3074497
```

Compare 3 test cases



Subjectively, I wrestle with the question of whether comparison across variables with different M is meaningful.

Entropy is reported in summarize() and summarizeFactors() in the rockchalk package

Manufacture a variable to re-produce testcase3

```
round(testcase3,2)
```

```
[1] 0.04 0.07 0.11 0.14 0.18 0.21 0.25
```

```
library(rockchalk)
testcase3v <- factor(c(1,2,2,3,3,3, 4,4,4,4, 5,5,5,5,5,
    6,6,6,6,6,6, 7,7,7,7,7,7,7 ))
round((table(testcase3v)/length(testcase3v)), 2)
```

```
testcase3v
   1    2    3    4    5    6    7
0.04 0.07 0.11 0.14 0.18 0.21 0.25
```

```
dat <- data.frame(testcase3v)
summarizeFactors(dat)
```

Entropy is reported in `summarize()` and `summarizeFactors()` in the rockchalk package ...

```
          testcase3v
7                : 7.0000
6                : 6.0000
5                : 5.0000
4                : 4.0000
(All Others)     : 6.0000
NA's             : 0.0000
entropy          : 2.6100
normedEntropy    : 0.9297
N                :28.0000
```