

# SDM4 in R: Comparing Counts (Chapter 24)

Nicholas Horton ([nhorton@amherst.edu](mailto:nhorton@amherst.edu)) and Sarah McDonald

June 13, 2018

## Introduction and background

This document is intended to help describe how to undertake analyses introduced as examples in the Fourth Edition of *Stats: Data and Models* (2014) by De Veaux, Velleman, and Bock. More information about the book can be found at [http://wps.aw.com/aw\\_deveaux\\_stats\\_series](http://wps.aw.com/aw_deveaux_stats_series). This file as well as the associated R Markdown reproducible analysis source file used to create it can be found at <http://nhorton.people.amherst.edu/sdm4>.

This work leverages initiatives undertaken by Project MOSAIC (<http://www.mosaic-web.org>), an NSF-funded effort to improve the teaching of statistics, calculus, science and computing in the undergraduate curriculum. In particular, we utilize the `mosaic` package, which was written to simplify the use of R for introductory statistics courses. A short summary of the R needed to teach introductory statistics can be found in the `mosaic` package vignettes (<http://cran.r-project.org/web/packages/mosaic>). A paper describing the `mosaic` approach was published in the *R Journal*: <https://journal.r-project.org/archive/2017/RJ-2017-024>.

## Chapter 24: Comparing Counts

### Section 24.1: Goodness-of-fit tests

Here we verify the calculations of expected counts for ballplayers by month (page 656).

```
ballplayer <- c(137, 121, 116, 121, 126, 114,
               102, 165, 134, 115, 105, 122)
national <- c(0.08, 0.07, 0.08, 0.08, 0.08, 0.08,
             0.09, 0.09, 0.09, 0.09, 0.08, 0.09)
n <- sum(~ ballplayer)
n
```

```
## [1] 1478
```

```
sum(~ national)
```

```
## [1] 1
```

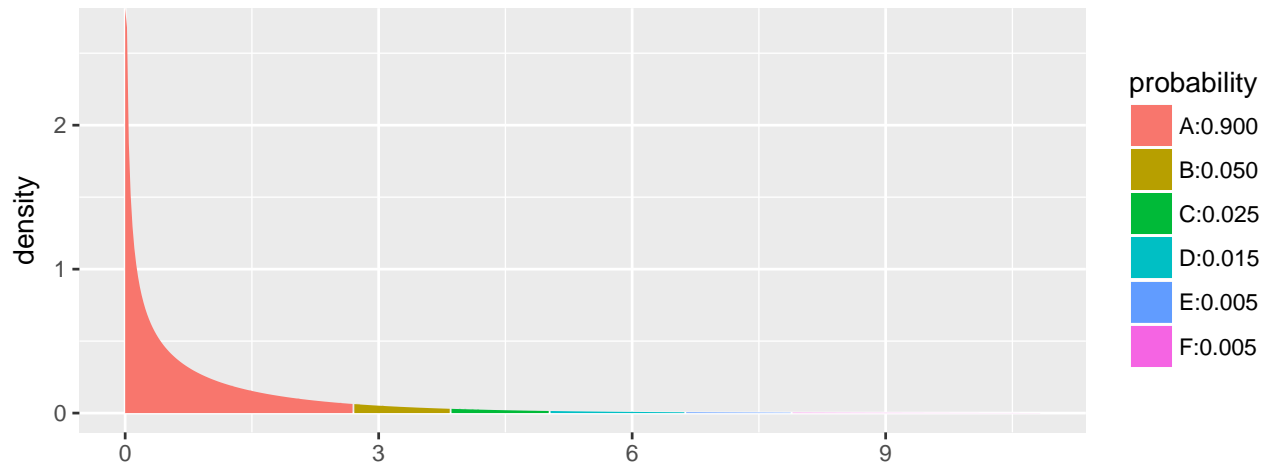
```
expect <- n*national
cbind(ballplayer, expect)
```

```
##      ballplayer expect
## [1,]         137 118.24
## [2,]         121 103.46
## [3,]         116 118.24
## [4,]         121 118.24
## [5,]         126 118.24
## [6,]          114 118.24
## [7,]          102 133.02
## [8,]          165 133.02
```

```
## [9,]      134 133.02
## [10,]     115 133.02
## [11,]     105 118.24
## [12,]     122 133.02
```

The chi-square quantile values in the table on the bottom of page 658 can be verified using the `xqt()` function.

```
xqchisq(c(.90, .95, .975, .99, .995), df = 1)
```



```
## [1] 2.7055 3.8415 5.0239 6.6349 7.8794
```

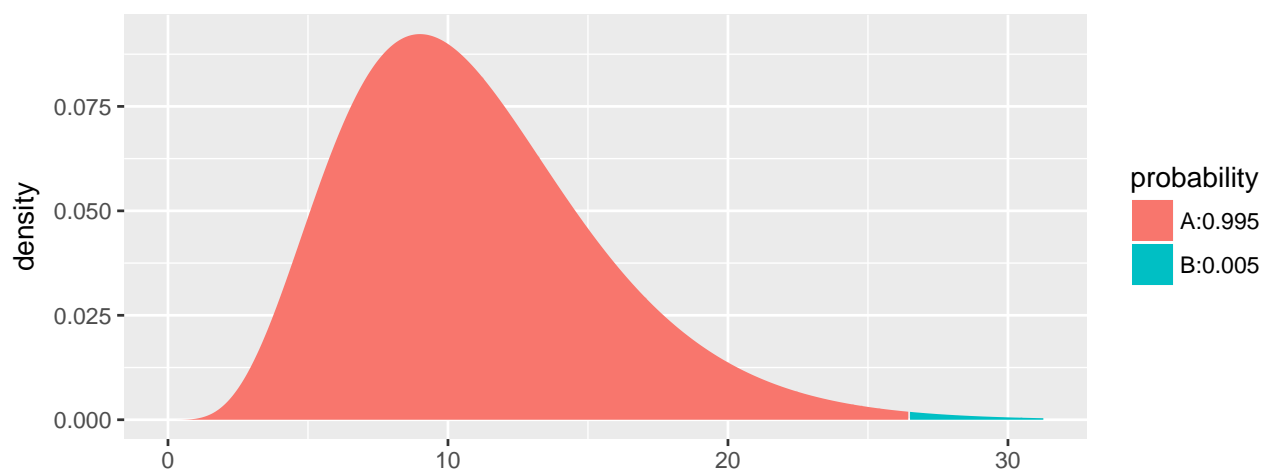
These results match the first row: other values can be calculated by changing the `df` argument.

The goodness of fit test on page 659 can be verified by calculating the chi-square statistic.

```
chisq <- sum((ballplayer-expect)^2/expect)
chisq
```

```
## [1] 26.484
```

```
1-xpchisq(chisq, df = 11)
```



```
## [1] 0.005494
```

## Section 24.2: Chi-square test of homogeneity

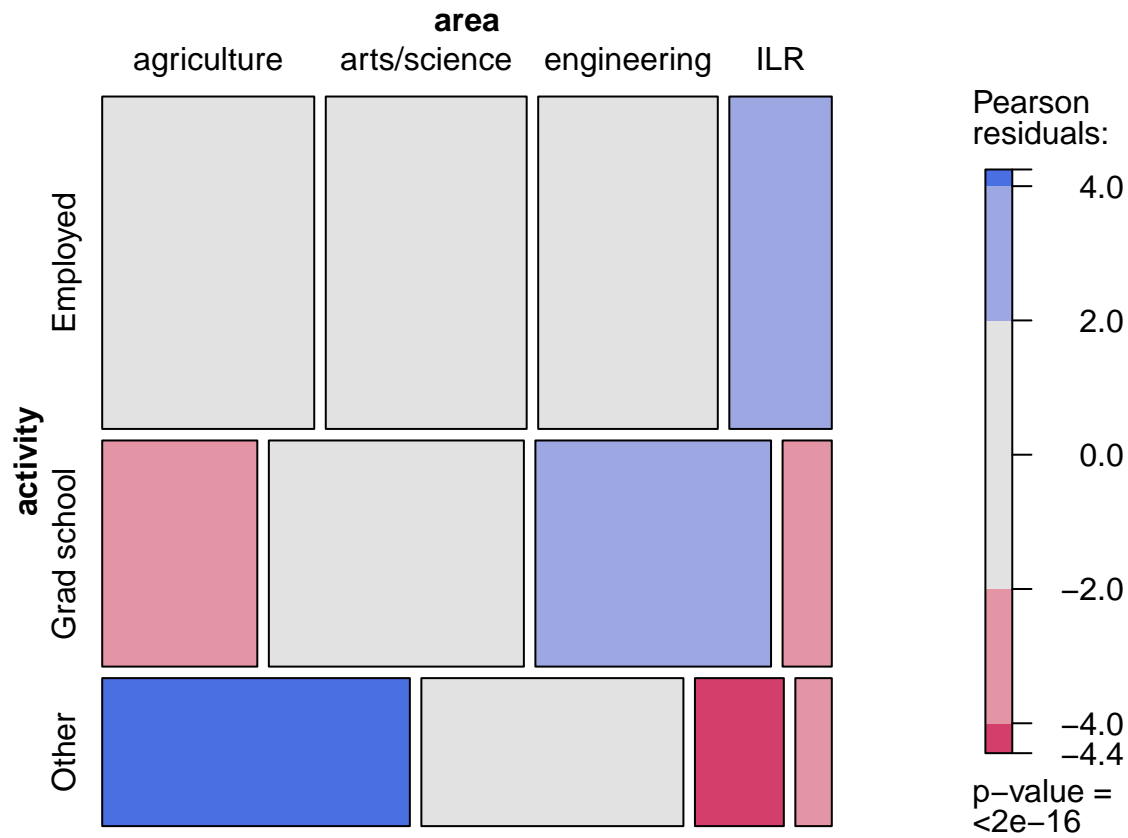
Data from one university regarding the association between postgraduation activity and area of study is displayed in Table 24.1 (page 663).

```
area <- c(rep("agriculture", 209), rep("arts/science", 198),
         rep("engineering", 177), rep("ILR", 101),
         rep("agriculture", 104), rep("arts/science", 171),
         rep("engineering", 158), rep("ILR", 33),
         rep("agriculture", 135), rep("arts/science", 115),
         rep("engineering", 39), rep("ILR", 16))
activity <- c(rep("Employed", 685), rep("Grad school", 466),
             rep("Other", 305))
tally(~ activity + area, margins = TRUE)
```

```
##           area
## activity  agriculture arts/science engineering  ILR Total
##  Employed           209           198           177  101  685
##  Grad school        104           171           158   33  466
##  Other              135           115            39   16  305
##  Total              448           484           374  150 1456
```

```
vcd::mosaic(tally(~ activity + area), main = "mosaicplot of activity by area",
            shade = TRUE)
```

# mosaicplot of activity by area



```
xchisq.test(tally(~ activity + area))
```

```
##
## Pearson's Chi-squared test
##
## data: x
## X-squared = 93.7, df = 6, p-value <2e-16
##
## 209 198 177 101
## (210.77) (227.71) (175.95) ( 70.57)
## [ 0.0149] [ 3.8754] [ 0.0062] [13.1215]
## <-0.122> <-1.969> < 0.079> < 3.622>
##
## 104 171 158 33
## (143.38) (154.91) (119.70) ( 48.01)
## [10.8181] [ 1.6720] [12.2543] [ 4.6918]
## <-3.289> < 1.293> < 3.501> <-2.166>
##
## 135 115 39 16
## ( 93.85) (101.39) ( 78.34) ( 31.42)
## [18.0470] [ 1.8277] [19.7590] [ 7.5689]
## < 4.248> < 1.352> <-4.445> <-2.751>
```

```
##  
## key:  
## observed  
## (expected)  
## [contribution to X-squared]  
## <Pearson residual>
```

### Section 24.3: Examining the residuals

Note that the `xchisq.test()` function displays the standardized residuals as the last item in each cell of the table (and these match the results in Table 24.4 on page 668).