

# STAT 477/577 - Technology Guide

## Module 2 - Section 1

### Numerical and Graphical Summaries of Contingency Tables

Below is an explanation of the R commands and functions needed to analyze contingency tables.

- **Obtaining a Contingency Table**

In lecture, we looked at data on the eye color and gender of Stat 101 students. This is the same data set as used in Module 1 - Section 1. These data are located in the `ISUeyecolorgender.csv` file. To read this data file into R, use the function `read.csv()` as given below.

```
ecgendata <- read.csv(file.choose(), header = T)
```

We will then use the function `factor()` to reorder the categories for the **Eye Color** variable.

```
ecgendata$EyeColor <- factor(ecgendata$EyeColor,
                             levels = c("Blue", "Hazel", "Green", "Brown", "Other"))
```

The contingency table for the two categorical variables can be found using the `table()` function as follows. Note: the first variable listed will be the row variable and the second variable will be the column variable in the contingency table.

```
ecgentable <- table(ecgendata$Gender, ecgendata$EyeColor)
ecgentable
```

```
##
##           Blue Hazel Green Brown Other
## Female   370   187   198   352    18
## Male     359   160   110   290    24
```

- **Marginal Distributions**

The function `margin.table()` will find the counts for the marginal distributions for the two variables. To find the proportions, we can divide these counts by the total number of observations, found by using the `sum()` function. For example, the counts and proportions (rounded to 4 decimal places) for the row variable **Gender** are:

```
margin.table(ecgentable, 1)
```

```
##
## Female   Male
##   1125    943
```

```
round(margin.table(ecgentable, 1)/sum(ecgentable), 4)

##
## Female    Male
## 0.544    0.456
```

and the counts and proportions (rounded to 4 decimal places) for the column variable `Eye Color` are:

```
margin.table(ecgentable, 2)

##
## Blue Hazel Green Brown Other
## 729 347 308 642 42

round(margin.table(ecgentable, 2)/sum(ecgentable), 4)

##
## Blue Hazel Green Brown Other
## 0.3525 0.1678 0.1489 0.3104 0.0203
```

Note: You can obtain the marginal distributions for the two categorical variables directly from the data file using the methods from Module 1 - Section 1. The method above calculates the marginal distributions using only the contingency table.

- **Conditional Distributions**

The function `prop.table()` will find the conditional distributions for the column variable given the row variable and for the row variable given the column variable. The conditional distributions (rounded to four decimal places) for `Eye Color` given `Gender` are:

```
round(prop.table(ecgentable, 1), 4)

##
##           Blue Hazel Green Brown Other
## Female 0.3289 0.1662 0.1760 0.3129 0.0160
## Male   0.3807 0.1697 0.1166 0.3075 0.0255
```

and for `Gender` given `Eye Color` are:

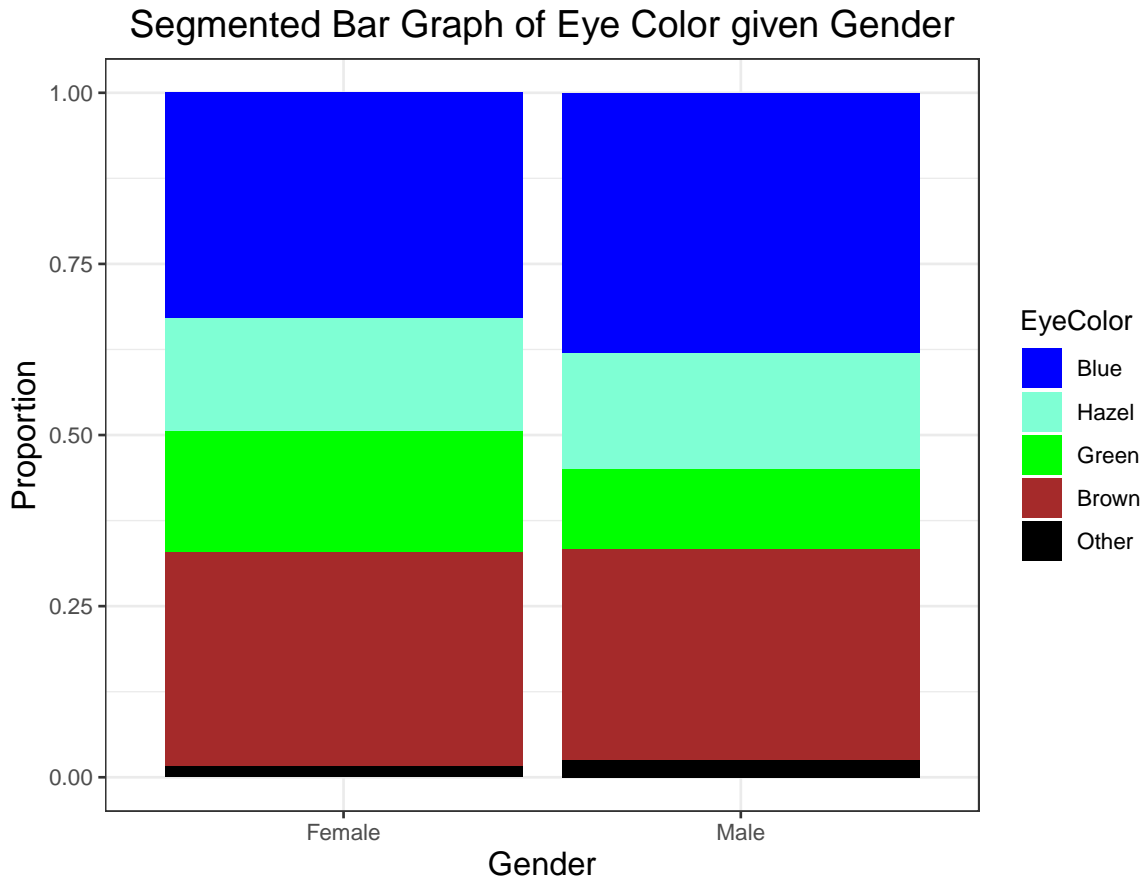
```
round(prop.table(ecgentable, 2), 4)
```

```
##
##           Blue  Hazel  Green  Brown  Other
##   Female 0.5075 0.5389 0.6429 0.5483 0.4286
##   Male   0.4925 0.4611 0.3571 0.4517 0.5714
```

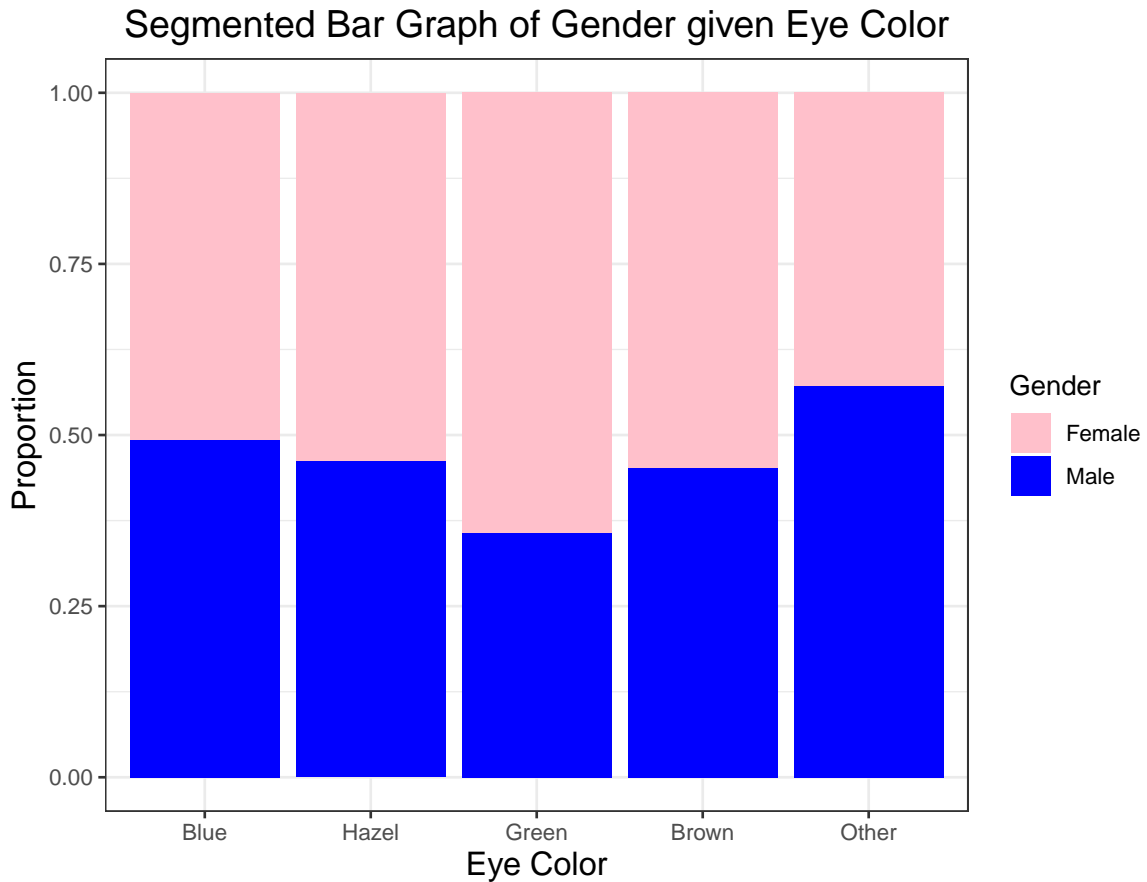
## • Graphics

There are several ways to structure graphics for contingency tables. The first set of commands will produce segmented bar graphs for Eye Color given Gender and the second set will produce segmented bar graphs for Gender given Eye Color.

```
ggplot(ecgendata, aes(x = Gender, fill = EyeColor))+
  geom_bar(position = "fill")+
  scale_fill_manual(values = c("blue", "aquamarine",
                                "green", "brown", "black"))+
  theme_bw()+
  theme(axis.title.y = element_text(size = rel(1.2)),
        axis.title.x = element_text(size = rel(1.2)),
        axis.text.x = element_text(size = rel(1)),
        axis.text.y = element_text(size = rel(1)),
        plot.title = element_text(hjust=0.5, size = rel(1.4)))+
  labs(y = "Proportion",
       title = "Segmented Bar Graph of Eye Color given Gender")
```



```
ggplot(ecgendata, aes(x = EyeColor, fill = Gender))+
  geom_bar(position = "fill")+
  scale_fill_manual(values=c("pink", "blue"))+
  theme_bw()+
  theme(axis.title.y = element_text(size = rel(1.2)),
        axis.title.x = element_text(size = rel(1.2)),
        axis.text.x = element_text(size = rel(1)),
        axis.text.y = element_text(size = rel(1)),
        plot.title = element_text(hjust=0.5, size = rel(1.4)))+
  labs(y = "Proportion",
       x = "Eye Color",
       title = "Segmented Bar Graph of Gender given Eye Color")
```

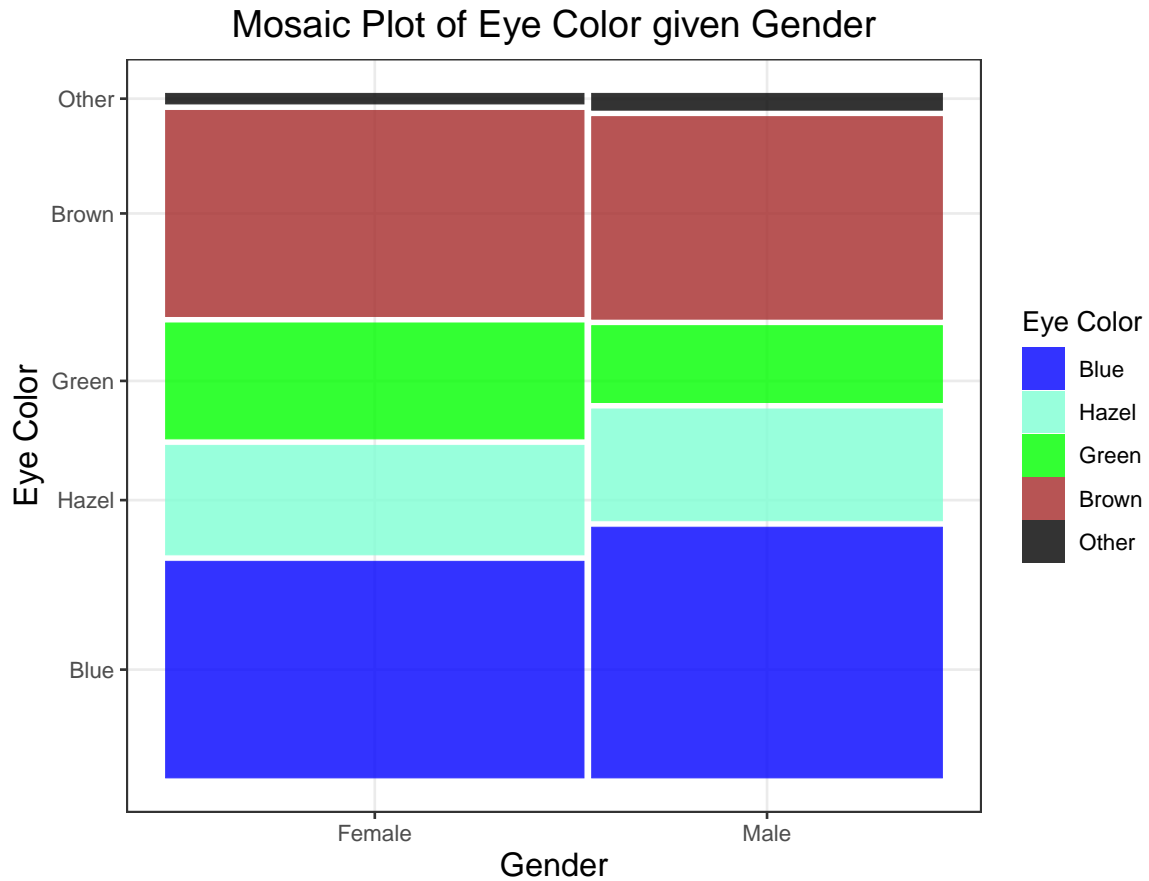


In order to graph a mosaic plot, we will need to install a new R package called `ggmosaic`. The function `geom_mosaic` can then be used to produce a mosaic plot for Eye Color given Gender (first set of commands) or Gender given Eye Color (second set of commands).

```
ggplot(data = ecgendata)+
  geom_mosaic(aes(x = product(EyeColor, Gender),
                        fill = EyeColor),
             na.rm = TRUE, divider = mosaic("h"))+
  scale_fill_manual(values = c("blue", "aquamarine",
                              "green", "brown", "black"))+

  theme_bw()+
  theme(plot.title = element_text(hjust=0.5, size = rel(1.4)))+
  theme(axis.title.y = element_text(size = rel(1.2)))+
  theme(axis.title.x = element_text(size = rel(1.2)))+
  theme(axis.text.x = element_text(size = rel(1)))+
  theme(axis.text.y = element_text(size = rel(1)))+
  theme(strip.text.y = element_text(size = rel(1.2)))+
  labs(x = "Gender",
       y = "Eye Color",
       fill = "Eye Color",
```

```
title = "Mosaic Plot of Eye Color given Gender")
```



```
ggplot(data = ecgendata)+
  geom_mosaic(aes(x = product(Gender, EyeColor), fill = Gender),
    na.rm = TRUE, divider = mosaic("h"))+
  scale_fill_manual(values=c("pink", "blue"))+
  theme_bw()+
  theme(plot.title = element_text(hjust=0.5, size = rel(1.4)),
    axis.title.y = element_text(size = rel(1.2)),
    axis.title.x = element_text(size = rel(1.2)),
    axis.text.x = element_text(size = rel(1)),
    axis.text.y = element_text(size = rel(1)),
    strip.text.y = element_text(size = rel(1.2)))+
  labs(x = "Eye Color",
    y = "Gender",
    fill = "Gender",
    title = "Mosaic Plot of Gender given Eye Color")
```

