Lecture 11 Module 5 – Two-Way Tables and Barplots

CSCI E-5a: Programming in R R

Let's clear the global computing environment:

```
rm( list = ls() )
```

Module Overview

Hello! And welcome to Module 5: Two-Way Tables and Barplots.

In this module, we'll learn how to construct tables and barplots across two variables.

- In Section 1, we'll learn a variety of methods for constructing a two-way table.
- In Section 2, we'll learn how to construct a two-way barplot.

When you've completed this module, you'll be able to:

- Construct two-way tables
- Construct two-way barplots

There is one new built-in R function in this module:

- `addmargins()`

All right! Let's get started by learning how to construct two-way tables.

Section 1: Two-Way Tables

Main Idea: *We can construct two-way tables*

In this section, we'll learn a variety of methods for constructing a two-way table.

Now let's take a look at a more complicated example.

Weight	Color	Region
467	Red	East
483	Red	East
491	Red	East
452	Red	West
441	Red	West

Weight	Color	Region
398	Red	West
447	Red	West
338	Blue	East
325	Blue	East
304	Blue	West
313	Blue	West
317	Blue	West
322	Blue	West
309	Blue	West

Let's put this into R:

```
weight.vector <-
    c( 467, 483, 491, 452, 441, 398, 447,
       338, 325, 304, 313, 317, 322, 309 )

color.vector <-
    c( "Red", "Red", "Red", "Red", "Red", "Red", "Red",
       "Blue", "Blue", "Blue", "Blue",
       "Blue", "Blue", "Blue" )

region.vector <-
    c( "East", "East", "East",
       "West", "West", "West", "West",
       "East", "East",
       "West", "West", "West", "West", "West" )
```

Notice that in this case we now have two categorical variables: `color.vector` and `region.vector`.

We can create a two-dimensional table of the counts by using the `table` function with the two categorical variables:

```
two.way.count.table <-
    table( color.vector, region.vector )

two.way.count.table
```

```
##             region.vector
## color.vector East West
##         Blue    2    5
##         Red     3    4
```

You can convert this two-way frequency count table to a two-way relative proportions table using the `proportions()` function:

```
proportions( two.way.count.table )
```

```
##             region.vector
## color.vector      East      West
##         Blue 0.1428571 0.3571429
##         Red  0.2142857 0.2857143
```

You can add marginal totals to a two-way table by using the `addmargins()` function:

```
addmargins( two.way.count.table )
```

```
##              region.vector
## color.vector East West Sum
##         Blue    2    5   7
##         Red     3    4   7
##         Sum     5    9  14
```

We can also perform two-way summaries with `tapply()`, but in this case we have to bundle the two categorical vectors together using a list:

```
two.way.mean.summary <-
    tapply(
        weight.vector,
        list( color.vector, region.vector ),
        mean
    )

two.way.mean.summary
```

```
##          East  West
## Blue 331.5000 313.0
## Red  480.3333 434.5
```

You can also do a two-way analysis like this using the `aggregate()` function:

```
aggregate(
  x = list( weight = weight.vector ),
  by = list( Color = color.vector, Region = region.vector ),
  FUN = mean
)
```

```
##   Color Region   weight
## 1  Blue   East 331.5000
## 2   Red   East 480.3333
## 3  Blue   West 313.0000
## 4   Red   West 434.5000
```

So those are a variety of methods for constructing a two-way table.

Now let's see how to construct two-way barplots.

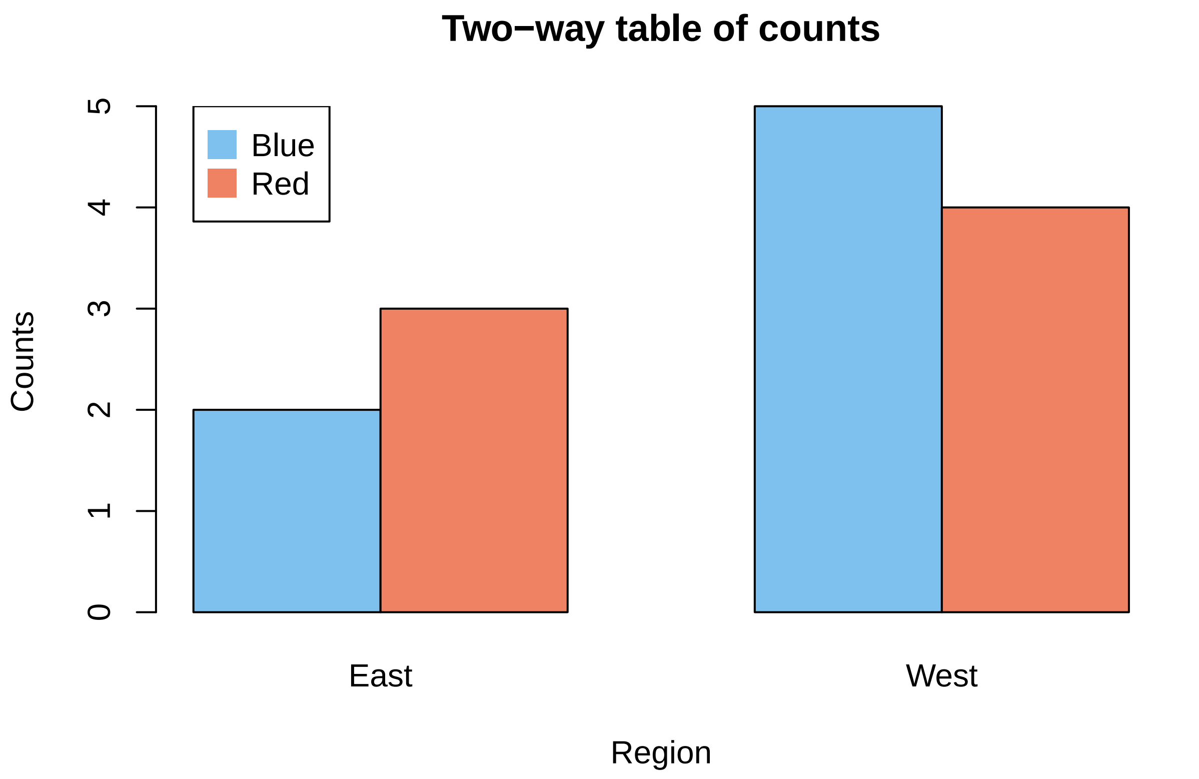
Section 2: Two-Way Barplots

Main Idea: *We can construct two-way barplots*

In this section, we'll learn how to construct a two-way barplot.

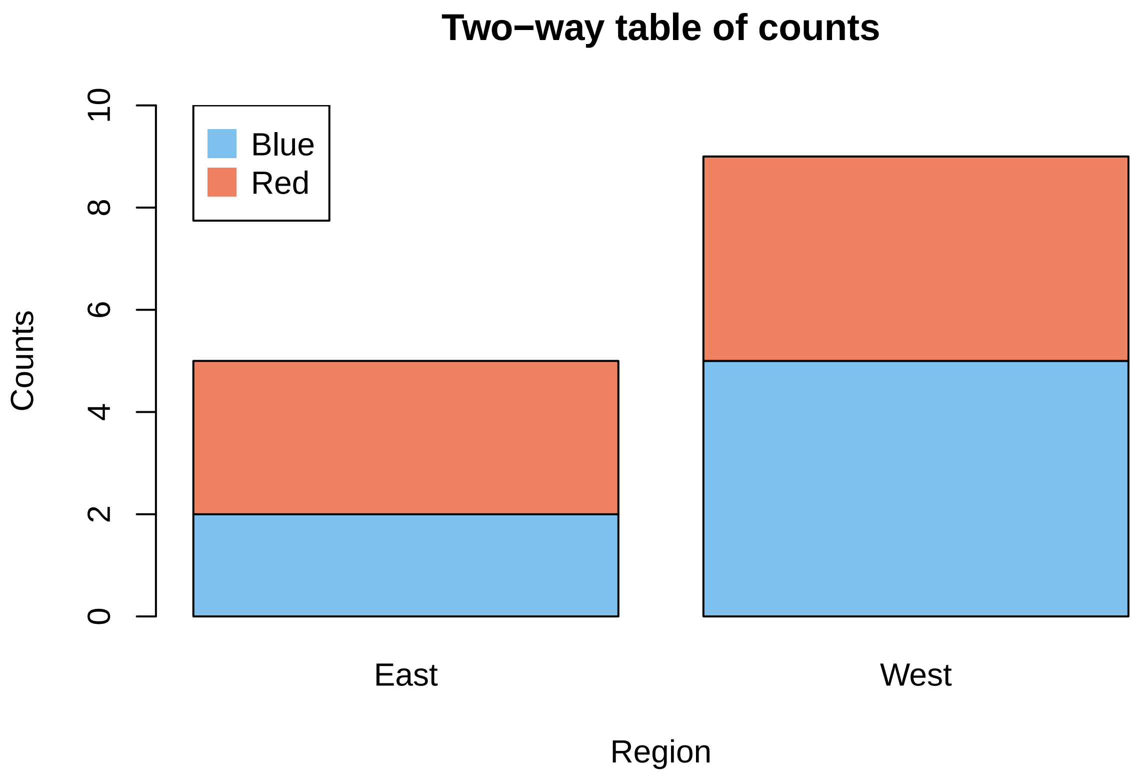
We can make a grouped barplot with this data:

```
barplot(
    two.way.count.table,
    main = "Two-way table of counts",
    xlab = "Region",
    ylab = "Counts",
    beside = TRUE,
    col = c( "skyblue2", "salmon2" )
)

legend(
    x = 1,
    y = 5,
    legend = c( "Blue", "Red" ),
    pch = 15,
    col = c( "skyblue2", "salmon2" ),
    pt.cex = 2
)
```



If you leave out the parameter `beside = TRUE`, the bars are stacked one on top of the other:

```
barplot(
    two.way.count.table,
    ylim = c(0, 10),
    main = "Two-way table of counts",
    xlab = "Region",
    ylab = "Counts",
    col = c( "skyblue2", "salmon2" )
)

legend(
    x = 0.2,
    y = 10,
    legend = c( "Blue", "Red" ),
    pch = 15,
    col = c( "skyblue2", "salmon2" ),
    pt.cex = 2
)
```

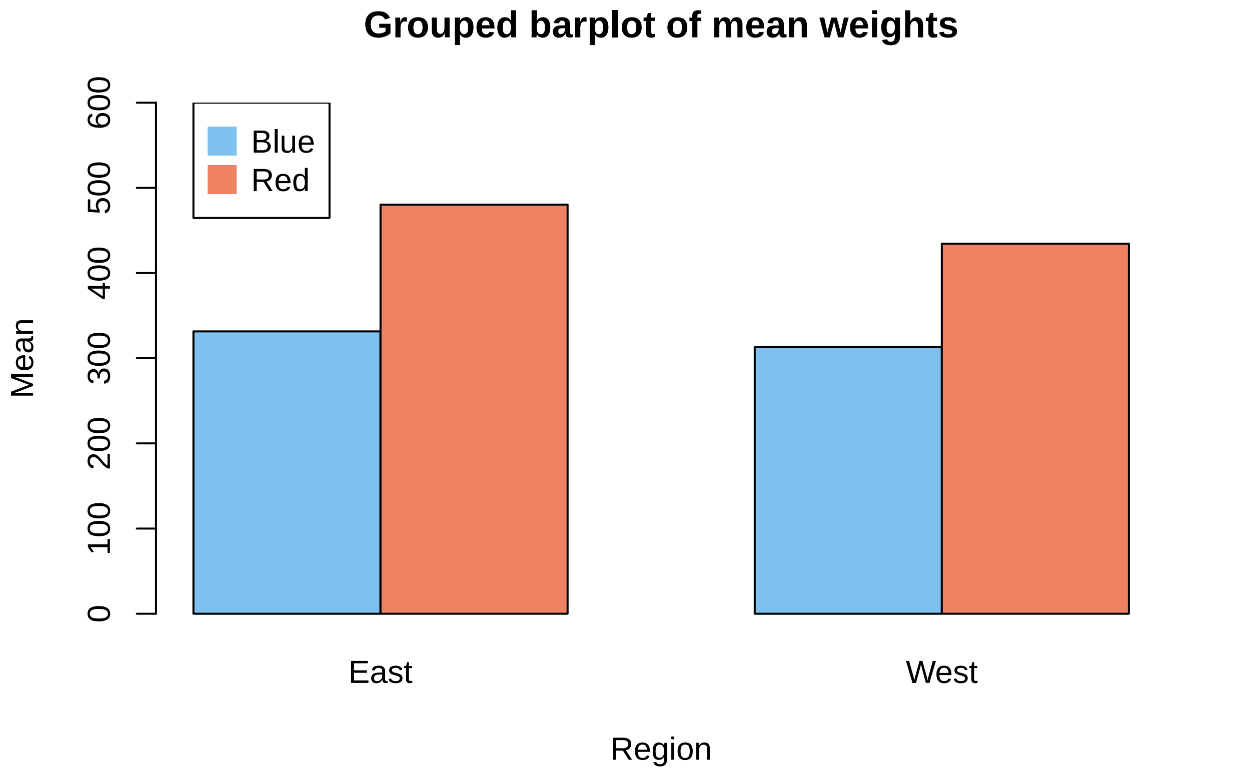


Personally, I find these sorts of charts hard to interpret, and I don't think they are used very often in the real world.

We can also make these grouped barplots using the summary from the `tapply()` function:

```
barplot(
    two.way.mean.summary,
    ylim = c(0, 600),
    main = "Grouped barplot of mean weights",
    xlab = "Region",
    ylab = "Mean",
    beside = TRUE,
    col = c( "skyblue2", "salmon2" )
)
```

```
legend(
    x = 1,
    y = 600,
    legend = c( "Blue", "Red" ),
    pch = 15,
    col = c( "skyblue2", "salmon2" ),
    pt.cex = 2
)
```

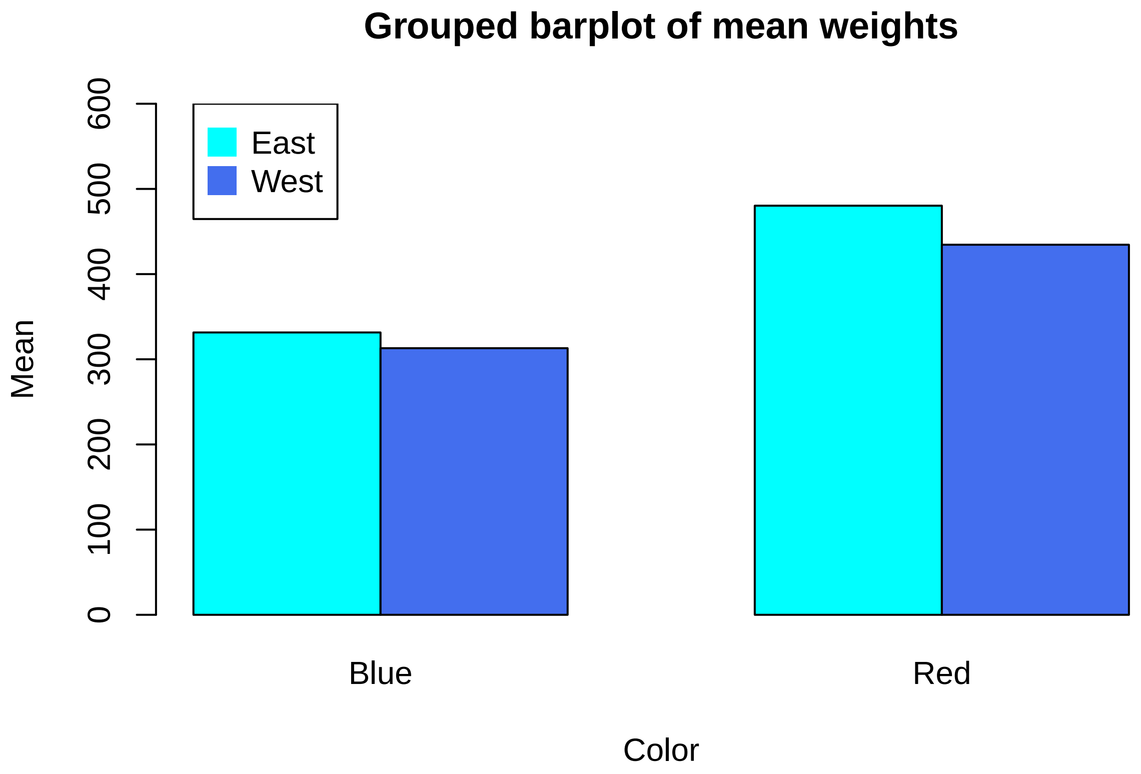


Notice that the bars are grouped by colors first, and then regions.

That's because of how we specified the call to the `tapply()` function.

If we reverse this order, we'll get a different barplot:

```
barplot(
    tapply(
        weight.vector,
        list( region.vector, color.vector ),
        mean
    ),
    ylim = c(0, 600),
    main = "Grouped barplot of mean weights",
    xlab = "Color",
    ylab = "Mean",
    beside = TRUE,
    col = c( "cyan1", "royalblue2" )
)

legend(
    x = 1,
    y = 600,
    legend = c( "East", "West" ),
    pch = 15,
    col = c( "cyan1", "royalblue2" ),
    pt.cex = 2
)
```



So that's how to construct a two-way barplot.

Now let's review what we've learned in this module.

Module Review

In this module, we learned how to construct tables and barplots across two variables.

- In Section 1, we learned a variety of methods for constructing a two-way table.
- In Section 2, we learned how to construct a two-way barplot.

Now that you've completed this module, you should be able to:

- Construct two-way tables
- Construct two-way barplots

There was one new built-in R function in this module:

- `addmargins()`

All right! That's it for Module 5: Two-Way Tables and Barplots.

In fact, that's the end of the content for Week 11: Summarizing Data.

And it's also the end of the core content for our course!