Histograms

Introduction

One of the most critical steps in data science – one I'll confess I often overlook in my rush to look at treatment differences – is to inspect the distribution of data. This allows us to confirm whether the distribution follows a bell shape, and also to identify extreme values that should be further investigated. There are a multiple ways to do this. This week, we will focus on the histogram.

Case Study

The first example we will use is a cotton uniformity trial conducted in Greece in 1938. Yes, it is a bit dated, but the data are open-source and the concept is one we deal with every day in agricultural research: how consistent are our plots? In other words, what is the variance among plots?

In a uniformity trial, a several plots are managed identically so that the distribution of their yields can be measured. These data can be used to determine whether that field is a good site for a research trial, or how data might vary n a similar field.

Our first step is to load the data, which is in a .csv file. To do this, we will use the `read.csv` function.

```
cotton = read.csv("data/cotton_uniformity.csv")
```

Basic Histogram

Drawing a simple histogram in R requires two steps:

1. Define the column in which we are interested.
2. Create the histogram.

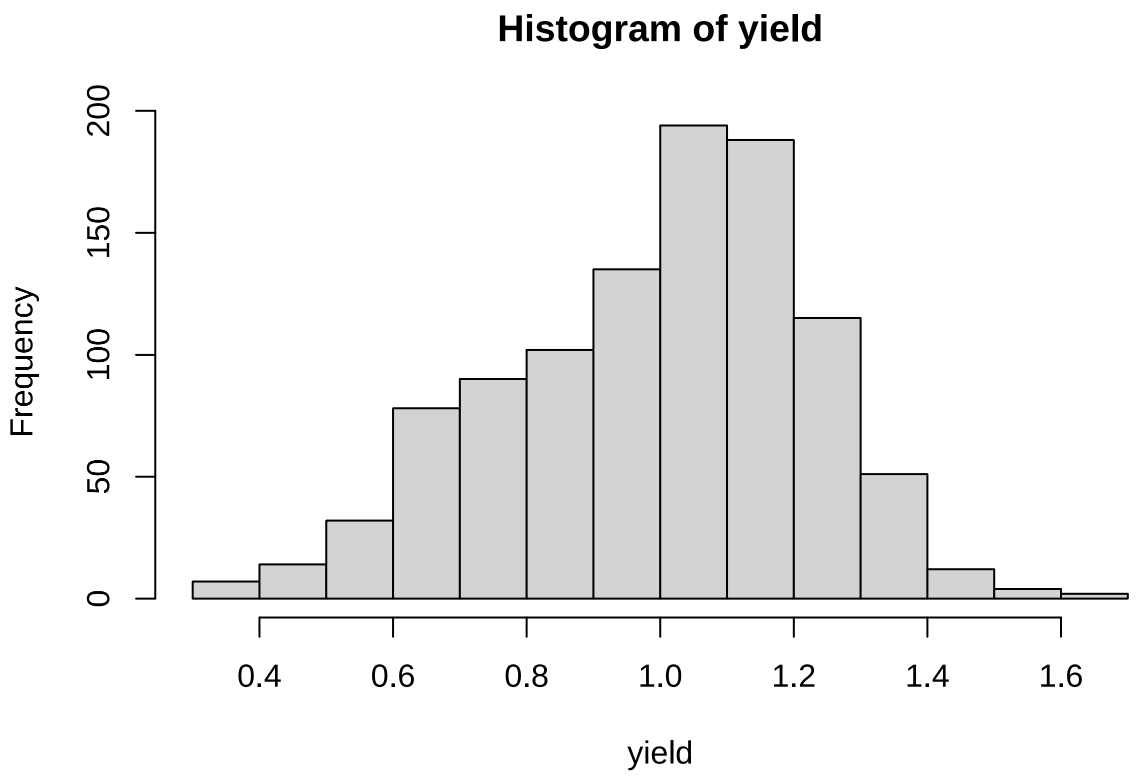
Since we are using a data.frame, we need to tell R which column to plot. We will first extract the data from the data.frame We will call this column "yield". To extract the yield column from our data.frame, we reference the column of the data.frame as `cotton$yield`. To the left of the dollar sign ($) is the name of the data.frame, *cotton*. To the right of the dollar sign is the name of the column, *yield*.

```
yield = cotton$yield
```

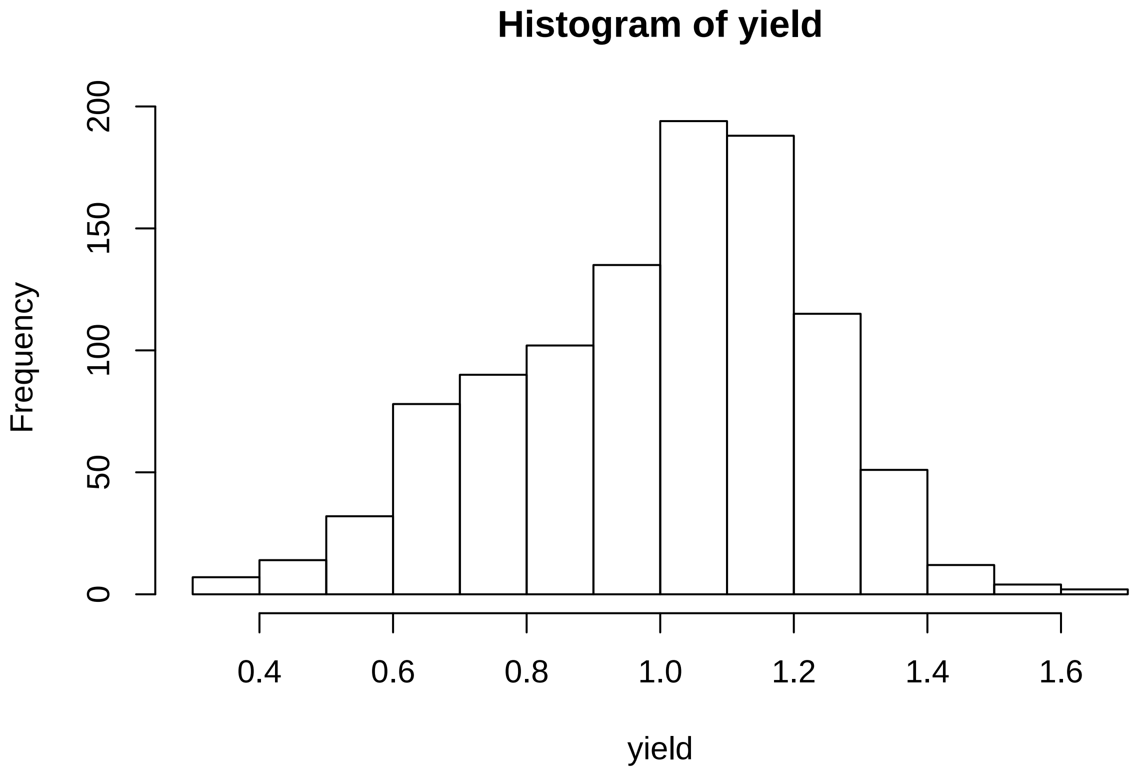
We then can create the histogram using the `hist()` function. We will assign the output to another R object, *histogram*. As we will see, that object contains much more information than just our histogram plot.

In the following line, we tell R to plot the histogram, using the `plot()` function.

```
histogram = hist(yield)
```



```r
plot(histogram)
```



And, *voila*!, we have our histogram. Each bar represents a range in values called a **bin**.

If we view the histogram object itself, however, we will see it is composed of many different statistics. histogram is a kind of R object called a **list**. It is a collection of many data frames.

```
histogram
```

```
## $breaks
##  [1] 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7
##
## $counts
##  [1]   7  14  32  78  90 102 135 194 188 115  51  12   4   2
##
## $density
##  [1] 0.06835938 0.13671875 0.31250000 0.76171875 0.87890625 0.99609375
##  [7] 1.31835938 1.89453125 1.83593750 1.12304688 0.49804687 0.11718750
## [13] 0.03906250 0.01953125
##
## $mids
##  [1] 0.35 0.45 0.55 0.65 0.75 0.85 0.95 1.05 1.15 1.25 1.35 1.45 1.55 1.65
##
## $xname
## [1] "yield"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

Each of these objects can be viewed separately, using the dollar sign method we used above.

For example, we can see the upper and lower yield limits of each bins by running `histogram$breaks`.

```
histogram$breaks
```

```
##  [1] 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7
```

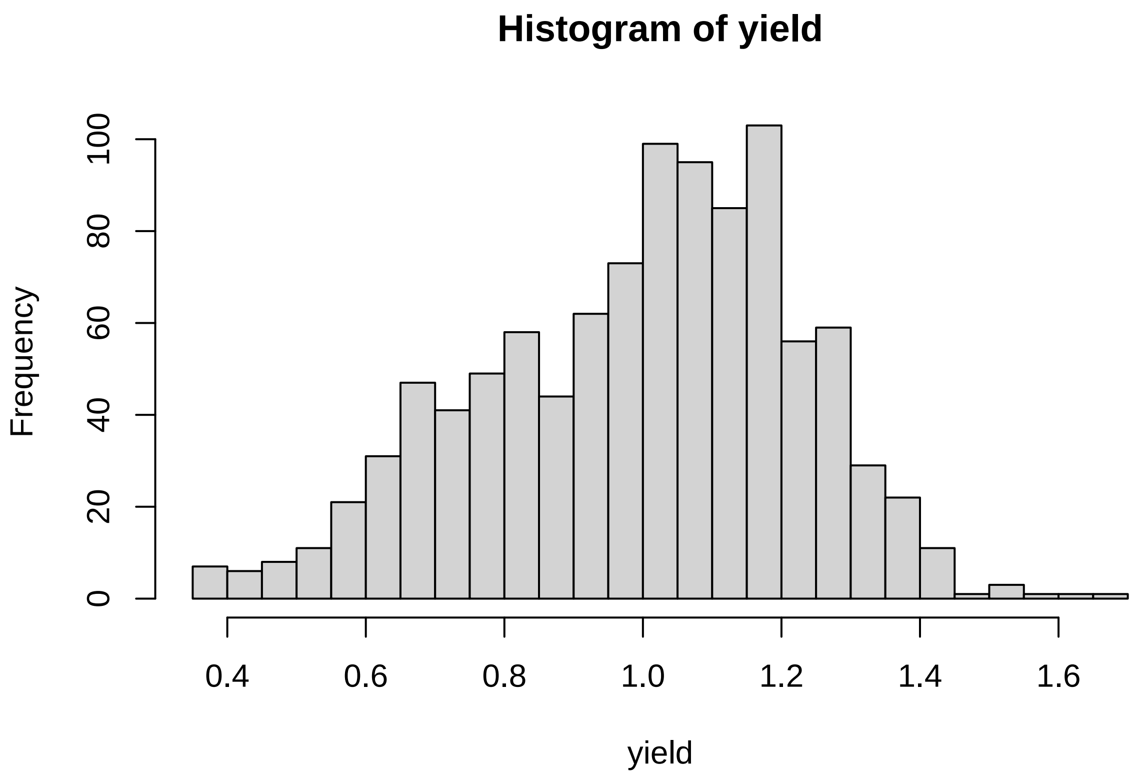
Alternatively, we could see the midpoints, or yield values that define the middle of each bin by running `histogram$mids` as a command.

```
histogram$mids
```

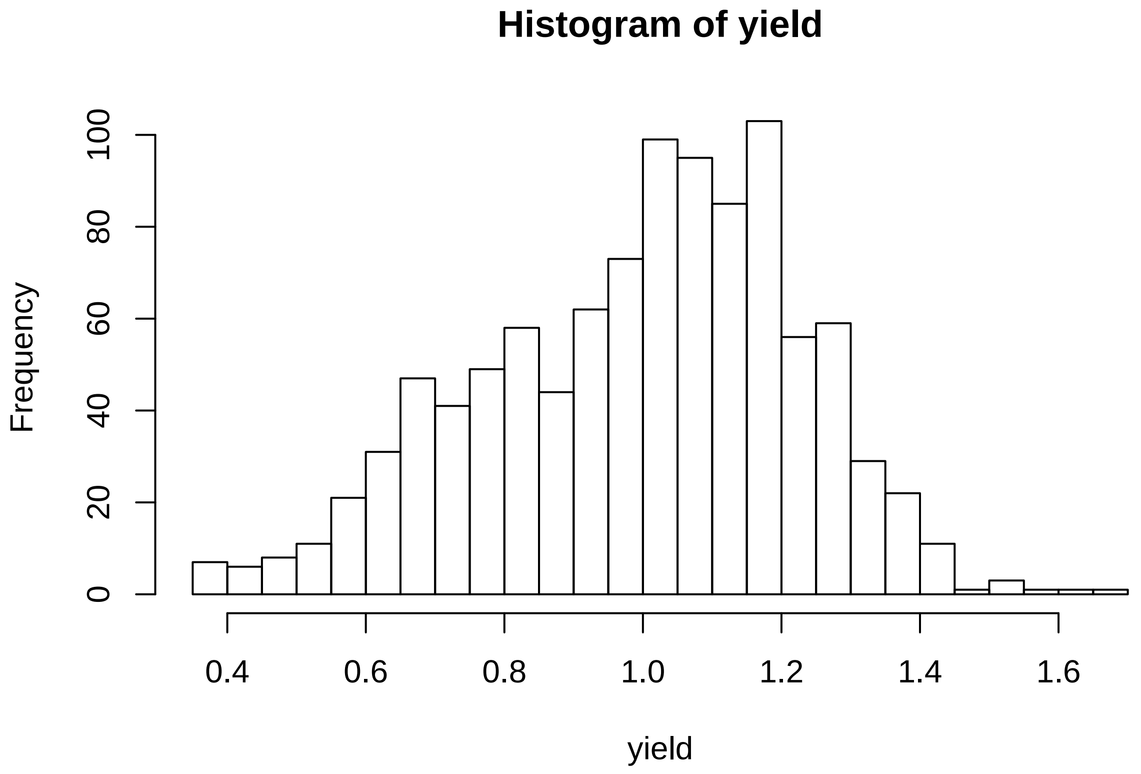
```
##  [1] 0.35 0.45 0.55 0.65 0.75 0.85 0.95 1.05 1.15 1.25 1.35 1.45 1.55 1.65
```

As we saw in the lesson, varying the number of columns can affect how we see patterns in the data. In the plot above, we have 14 bars. Each bar represents a bin width of 0.1. What if we tell R to use 28 bins to draw the histogram? We can do that by adding the argument `breaks=28` to our `hist()` function.

```
histogram = hist(yield, breaks = 28)
```



```
plot(histogram)
```



Note that we ended up with 27, not 28 bins, This is because it takes 28 breaks to define 27 bins. You can count the breaks, or sides of the bars in the histogram, to prove this to yourself. Each bar now has a bin width that is 0.05 kg. We can verify this by again inspecting the data by running `histogram$breaks`.

```
histogram$breaks
```

```
##  [1] 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00 1.05
## [16] 1.10 1.15 1.20 1.25 1.30 1.35 1.40 1.45 1.50 1.55 1.60 1.65 1.70
```

Histograms with ggplot

One of the great features of R is that there are hundreds of **packages** that can be added to it. These are like add-ins that you make have used with *Word* or *Excel*. Packages, like R, are open source, and are composed of specialized functions for many, many areas of research: agronomy, spatial data, ecology, medicine, machine learning, working with online databases like *SSURGO,* and so forth.

`ggplot2` is a one very powerful and population package in R, used to creating highly-customized plots. I use `ggplot2` to create a most of the plots I use in industry. if you can envision a plot, you almost certainly can draw it using `ggplot2`, .

There are entire books written on `ggplot2` (here is one I have consult regularly: https://r-graphics.org/index.html). I don't have time to teach it to you in depth, but I will show you many basics in this course.

The first time we want to use a package in R, we need to install it using the `install.packages()` function. While I try to install all packages you need for this course in *RStudio Cloud*, if you download *RStudio* to your desktop, you will need to install packages yourself. To install `ggplot2`, we run the following command:

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

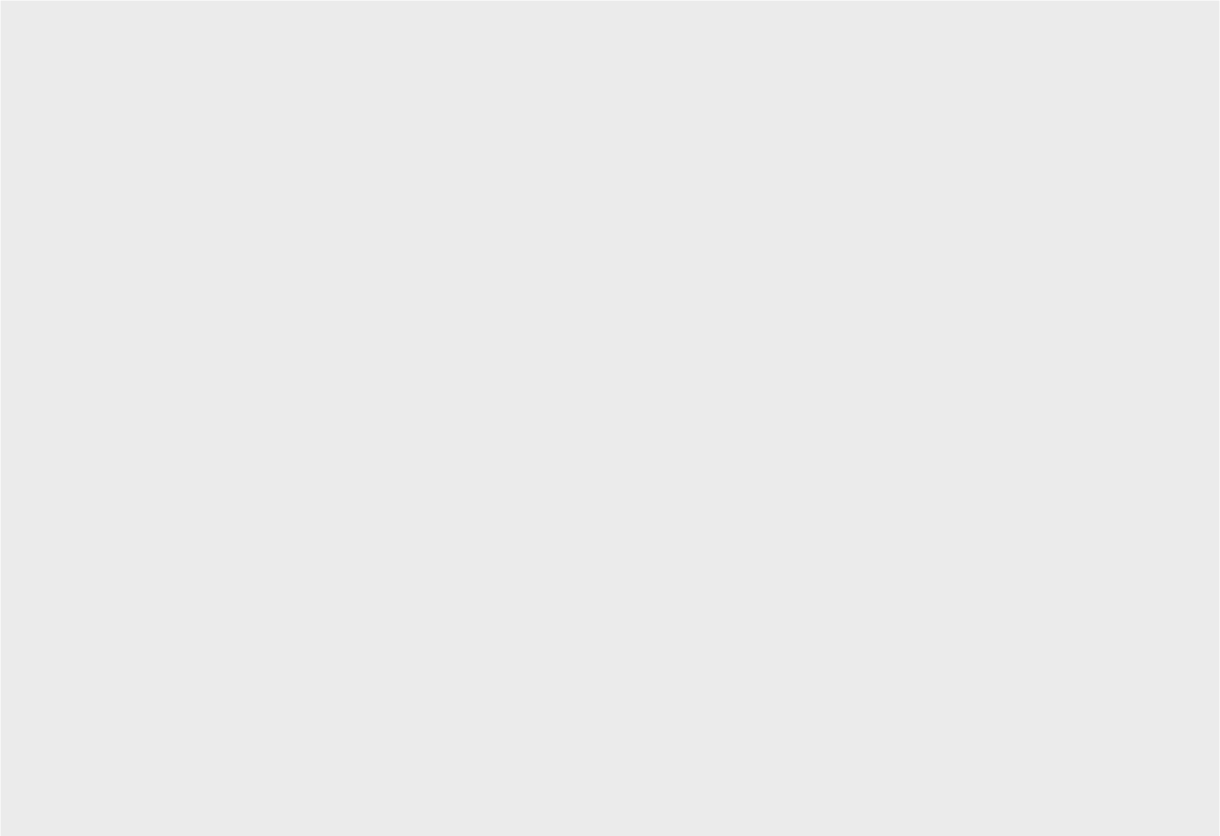
Note: you only need to install a package the very first time you use it. We can now start the package using the `library()` function.

```
library(ggplot2)
```

You only need to load a package once per R session. But if you restart R on your desktop, or log out and back into *RStudio Cloud*, you may need to reload the packages you intend to use. I make it a point to load the package in each notebook, rather than assuming it is loaded from another notebook I was working on at the same time.

The first line of code calls `ggplot()`. Although the package is called `ggplot2`, the function we use is `ggplot.` Confusing, I know.We use `data=cotton` to tell R we are drawing data from the "cotton" data frame.

```
ggplot(data=cotton)
```



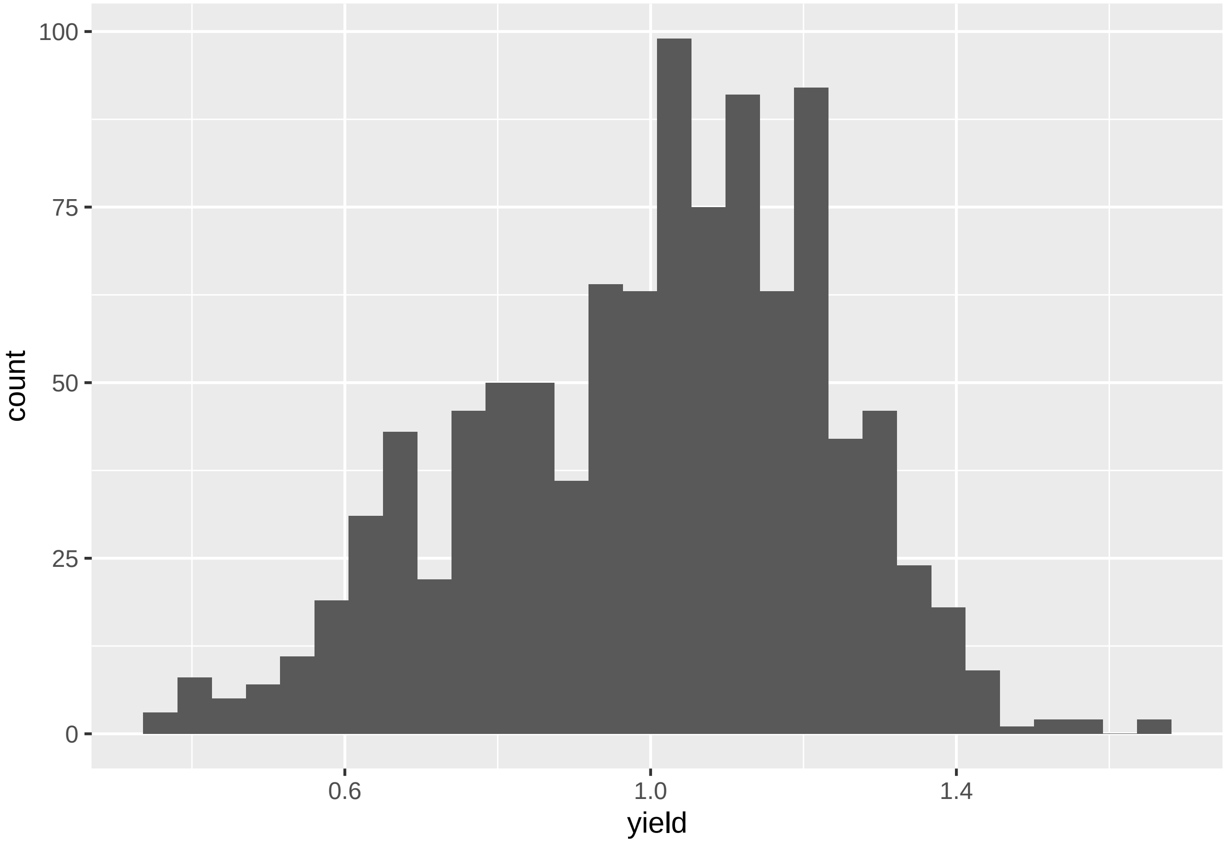
Running that first like alone will give us a blank plot. This is because we need to specify one or more aesthetics for each plot. An **aesthetic** is any property of the plot that relates to a variable, that is, a column in our data frame. In this case, we use aes(x=yield) to tell R that our aesthetic is positioned horizontally according to the value of yield.

Notice the *x-axis* now contains the range of yield values in our dataset. But there are no bars yet. We need to tell R what shape to use to represent the values of yield.

In the second line, we use `geom_histogram` to tell R we are drawing a histogram. Knowing this, R will automatically assign bins and count the number of observations in each bin.

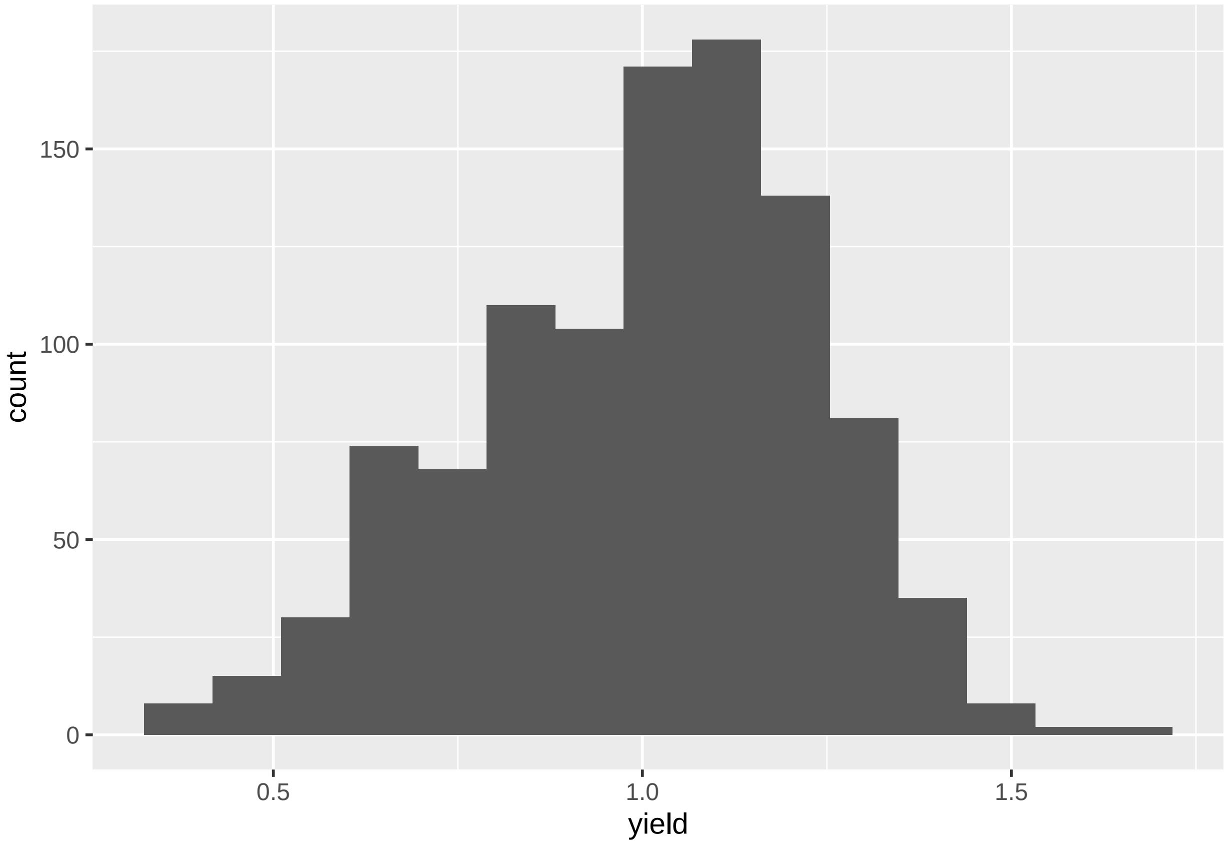
```r
ggplot(data=cotton, aes(x=yield)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



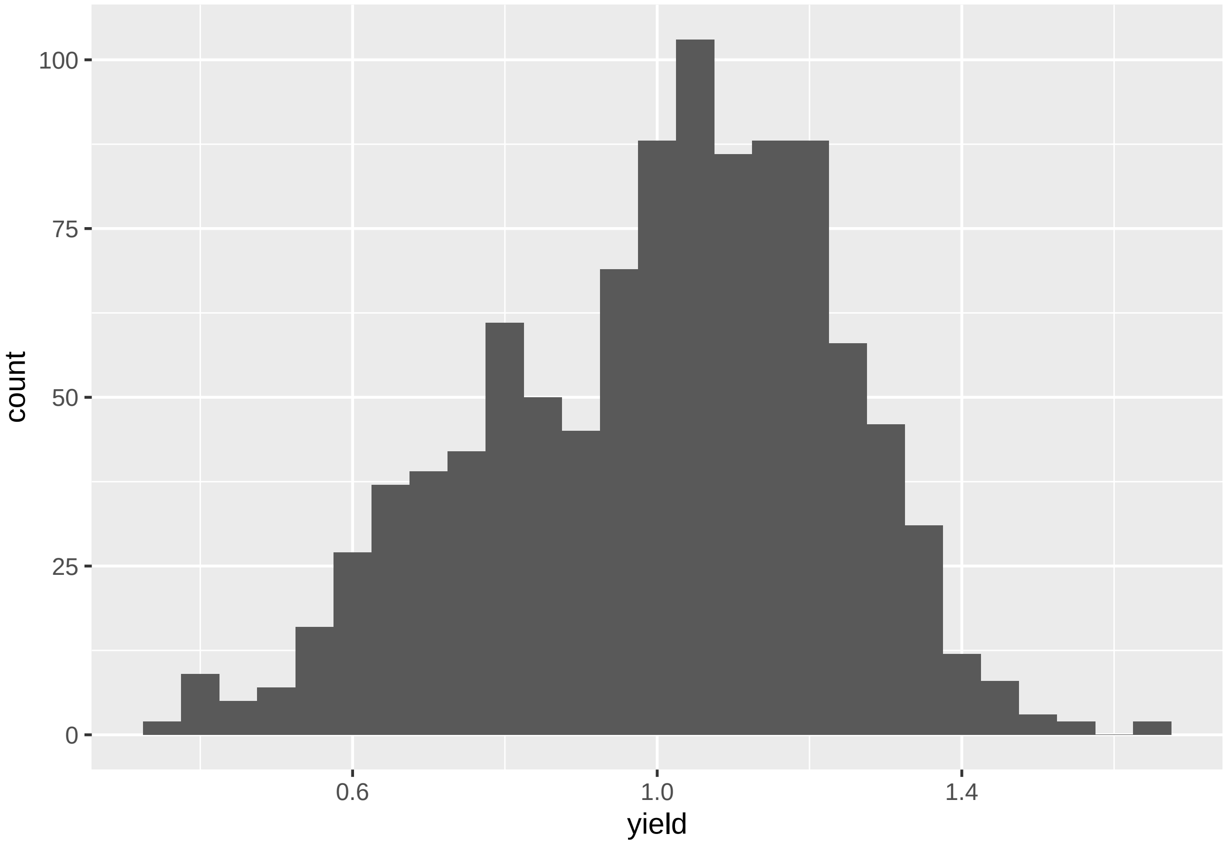
By default, R chooses to create 30 bins. We can easily specify a different number of bins adding the `bins=15` argument to the `geom_histogram()` function.

```
ggplot(data=cotton, aes(x=yield)) +
  geom_histogram(bins=15)
```



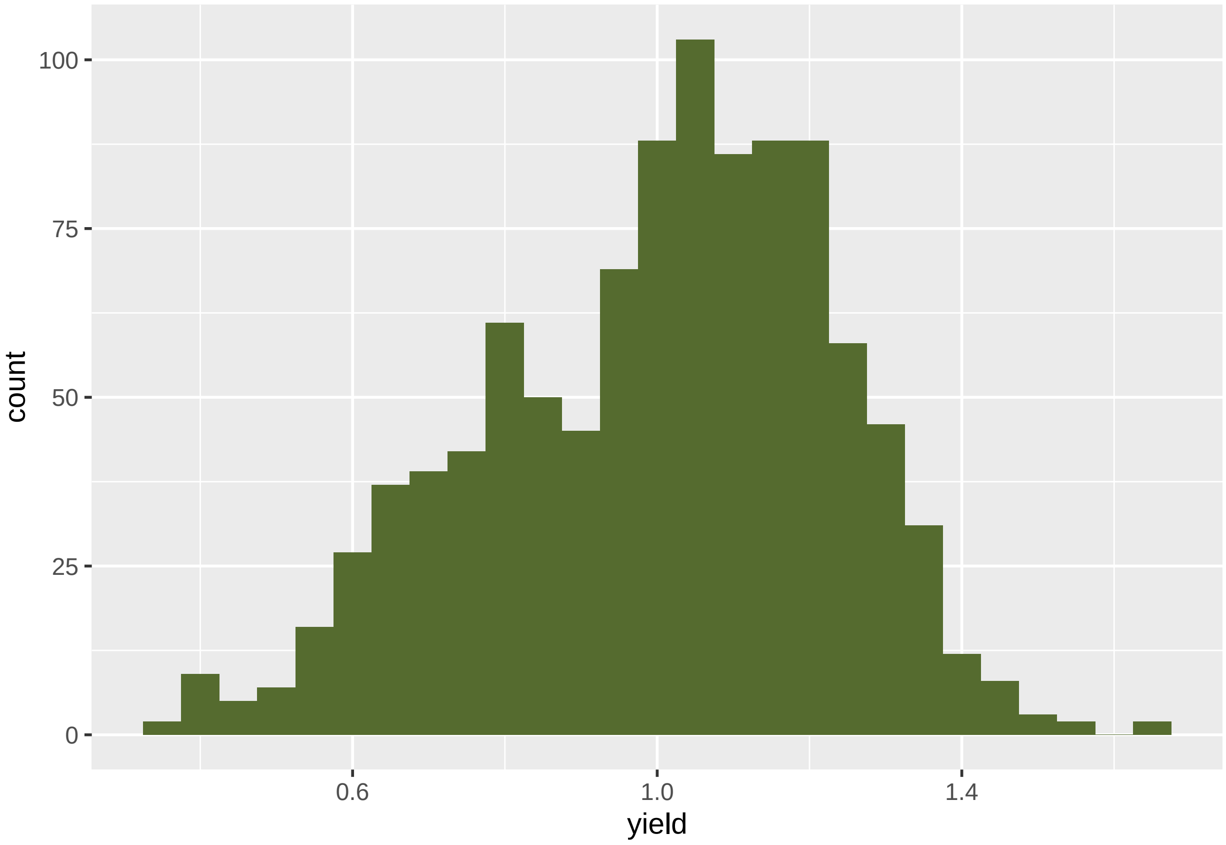
Alternatively, we can set a particular binwidth using the `binwidth =0.05` argument.

```
ggplot(data=cotton, aes(x=yield)) +
  geom_histogram(binwidth=0.05)
```



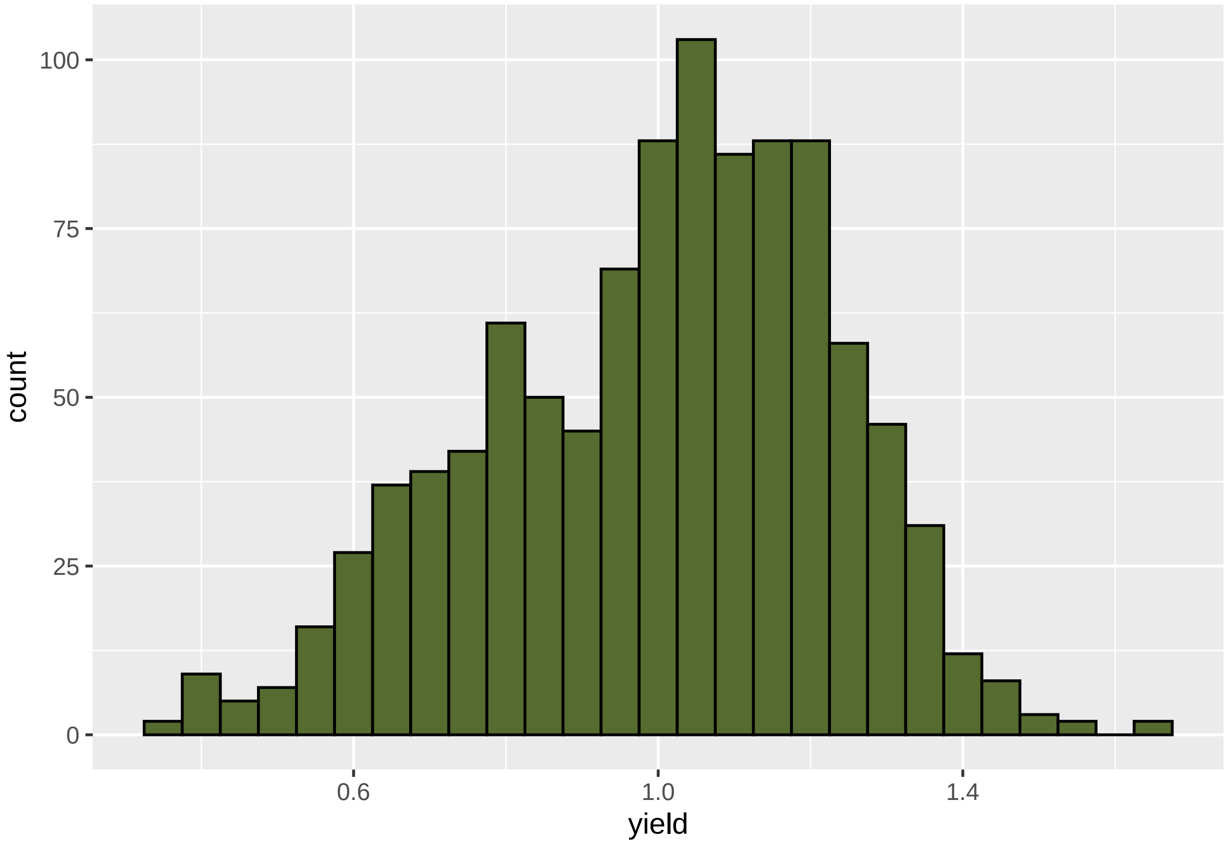
If we want to make our plot a little less bland, we can tell ggplot to use a different color to fill the bars by adding the `fill = darkolivegreen` argument.

```r
ggplot(data=cotton, aes(x=yield)) +
  geom_histogram(binwidth=0.05, fill="darkolivegreen")
```



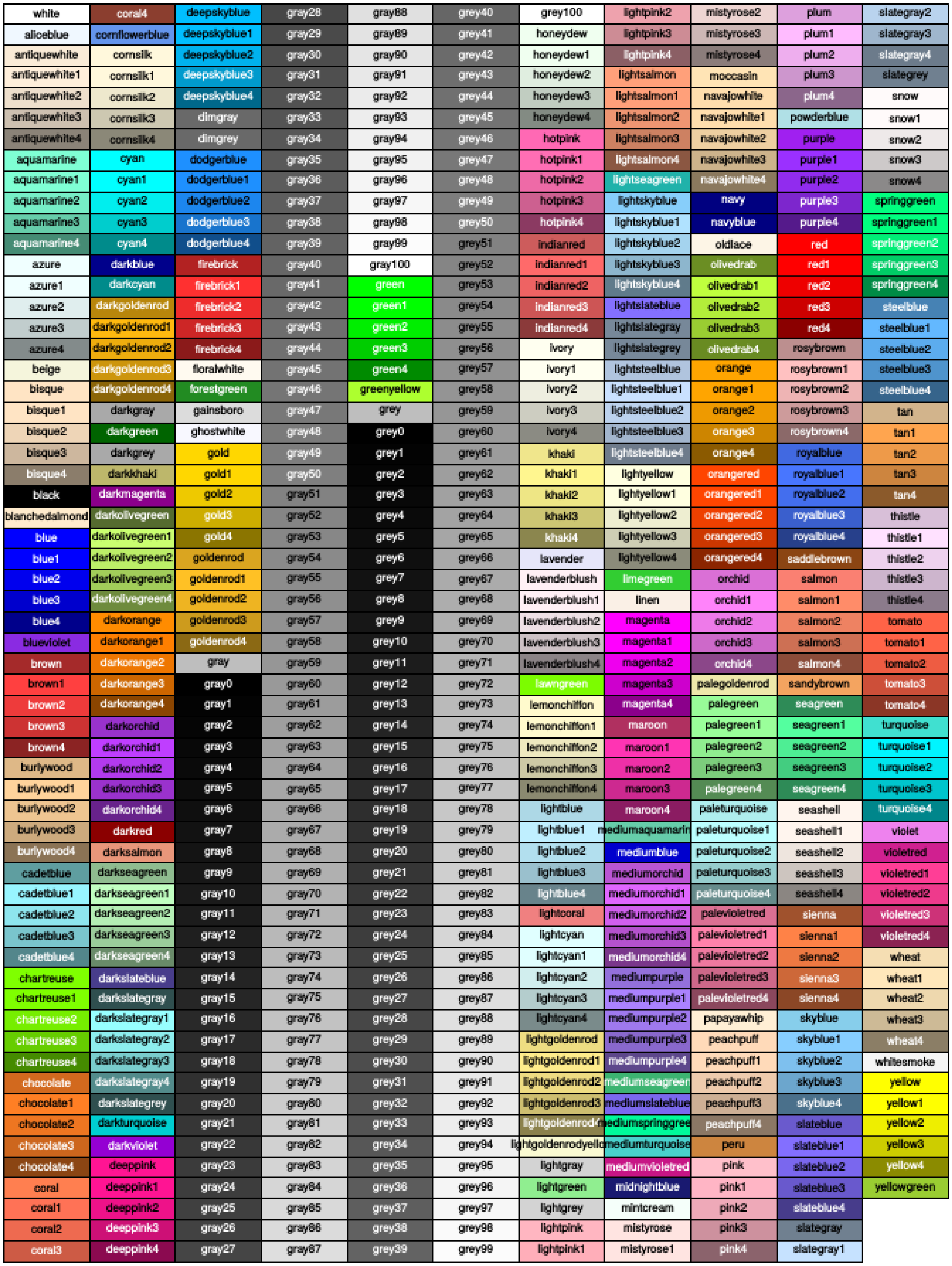
Finally, we can outline the bars using the `color ="black"` argument.

```
ggplot(data=cotton, aes(x=yield)) +
  geom_histogram(binwidth=0.05, fill="darkolivegreen", color="black")
```



We will introduce more ways to fine tune our plots as the course goes on. Here is a table with named colors you can use for fill or lines. (Note: If you right-click on it and select *Open in new tab*, you can enlarge it for much better legibility

R Colors By Name



white
aliceblue
antiquewhite
antiquewhite1
antiquewhite2
antiquewhite3
antiquewhite4
aquamarine
aquamarine1
aquamarine2
aquamarine3
aquamarine4
azure
azure1
azure2
azure3
azure4
beige
bisque
bisque1
bisque2
bisque3
bisque4
black
blanchedalmond
blue
blue1
blue2
blue3
blue4
blueviolet
brown
brown1
brown2
brown3
brown4
burlywood
burlywood1
burlywood2
burlywood3
burlywood4
cadetblue
cadetblue1
cadetblue2
cadetblue3
cadetblue4
chartreuse
chartreuse1
chartreuse2
chartreuse3
chartreuse4
chocolate
chocolate1
chocolate2
chocolate3
chocolate4
coral
coral1
coral2
coral3

coral4
cornflowerblue
cornsilk
cornsilk1
cornsilk2
cornsilk3
cornsilk4
cyan
cyan1
cyan2
cyan3
cyan4
darkblue
darkcyan
darkgoldenrod
darkgoldenrod1
darkgoldenrod2
darkgoldenrod3
darkgoldenrod4
darkgray
darkgreen
darkgrey
darkkhaki
darkmagenta
darkolivegreen
darkolivegreen1
darkolivegreen2
darkolivegreen3
darkolivegreen4
darkorange
darkorange1
darkorange2
darkorange3
darkorange4
darkorchid
darkorchid1
darkorchid2
darkorchid3
darkorchid4
darkred
darksalmon
darkseagreen
darkseagreen1
darkseagreen2
darkseagreen3
darkseagreen4
darkslateblue
darkslategray
darkslategray1
darkslategray2
darkslategray3
darkslategray4
darkslategrey
darkturquoise
darkviolet
deeppink
deeppink1
deeppink2
deeppink3
deeppink4

deepskyblue
deepskyblue1
deepskyblue2
deepskyblue3
deepskyblue4
dimgray
dimgrey
dodgerblue
dodgerblue1
dodgerblue2
dodgerblue3
dodgerblue4
firebrick
firebrick1
firebrick2
firebrick3
firebrick4
floralwhite
forestgreen
gainsboro
ghostwhite
gold
gold1
gold2
gold3
gold4
goldenrod
goldenrod1
goldenrod2
goldenrod3
goldenrod4
gray
gray0
gray1
gray2
gray3
gray4
gray5
gray6
gray7
gray8
gray9
gray10
gray11
gray12
gray13
gray14
gray15
gray16
gray17
gray18
gray19
gray20
gray21
gray22
gray23
gray24
gray25
gray26
gray27

gray28
gray29
gray30
gray31
gray32
gray33
gray34
gray35
gray36
gray37
gray38
gray39
gray40
gray41
gray42
gray43
gray44
gray45
gray46
gray47
gray48
gray49
gray50
gray51
gray52
gray53
gray54
gray55
gray56
gray57
gray58
gray59
gray60
gray61
gray62
gray63
gray64
gray65
gray66
gray67
gray68
gray69
gray70
gray71
gray72
gray73
gray74
gray75
gray76
gray77
gray78
gray79
gray80
gray81
gray82
gray83
gray84
gray85
gray86
gray87

gray88
gray89
gray90
gray91
gray92
gray93
gray94
gray95
gray96
gray97
gray98
gray99
gray100
green
green1
green2
green3
green4
greenyellow
grey
grey0
grey1
grey2
grey3
grey4
grey5
grey6
grey7
grey8
grey9
grey10
grey11
grey12
grey13
grey14
grey15
grey16
grey17
grey18
grey19
grey20
grey21
grey22
grey23
grey24
grey25
grey26
grey27
grey28
grey29
grey30
grey31
grey32
grey33
grey34
grey35
grey36
grey37
grey38
grey39

grey40
grey41
grey42
grey43
grey44
grey45
grey46
grey47
grey48
grey49
grey50
grey51
grey52
grey53
grey54
grey55
grey56
grey57
grey58
grey59
grey60
grey61
grey62
grey63
grey64
grey65
grey66
grey67
grey68
grey69
grey70
grey71
grey72
grey73
grey74
grey75
grey76
grey77
grey78
grey79
grey80
grey81
grey82
grey83
grey84
grey85
grey86
grey87
grey88
grey89
grey90
grey91
grey92
grey93
grey94
grey95
grey96
grey97
grey98
grey99

grey100
honeydew
honeydew1
honeydew2
honeydew3
honeydew4
hotpink
hotpink1
hotpink2
hotpink3
hotpink4
indianred
indianred1
indianred2
indianred3
indianred4
ivory
ivory1
ivory2
ivory3
ivory4
khaki
khaki1
khaki2
khaki3
khaki4
lavender
lavenderblush
lavenderblush1
lavenderblush2
lavenderblush3
lavenderblush4
lawngreen
lemonchiffon
lemonchiffon1
lemonchiffon2
lemonchiffon3
lemonchiffon4
lightblue
lightblue1
lightblue2
lightblue3
lightblue4
lightcoral
lightcyan
lightcyan1
lightcyan2
lightcyan3
lightcyan4
lightgoldenrod
lightgoldenrod1
lightgoldenrod2
lightgoldenrod3
lightgoldenrod4
lightgoldenrodyello
lightgray
lightgreen
lightgrey
lightpink
lightpink1

lightpink2
lightpink3
lightpink4
lightsalmon
lightsalmon1
lightsalmon2
lightsalmon3
lightsalmon4
lightseagreen
lightskyblue
lightskyblue1
lightskyblue2
lightskyblue3
lightskyblue4
lightslateblue
lightslategray
lightslategrey
lightsteelblue
lightsteelblue1
lightsteelblue2
lightsteelblue3
lightsteelblue4
lightyellow
lightyellow1
lightyellow2
lightyellow3
lightyellow4
limegreen
linen
magenta
magenta1
magenta2
magenta3
magenta4
maroon
maroon1
maroon2
maroon3
maroon4
mediumaquamarin
mediumblue
mediumorchid
mediumorchid1
mediumorchid2
mediumorchid3
mediumorchid4
mediumpurple
mediumpurple1
mediumpurple2
mediumpurple3
mediumpurple4
mediumseagreen
mediumslateblue
mediumspringgree
mediumturquoise
mediumvioletred
midnightblue
mintcream
mistyrose
mistyrose1

mistyrose2
mistyrose3
mistyrose4
moccasin
navajowhite
navajowhite1
navajowhite2
navajowhite3
navajowhite4
navy
navyblue
oldlace
olivedrab
olivedrab1
olivedrab2
olivedrab3
olivedrab4
orange
orange1
orange2
orange3
orange4
orangered
orangered1
orangered2
orangered3
orangered4
orchid
orchid1
orchid2
orchid3
orchid4
palegoldenrod
palegreen
palegreen1
palegreen2
palegreen3
palegreen4
paleturquoise
paleturquoise1
paleturquoise2
paleturquoise3
paleturquoise4
palevioletred
palevioletred1
palevioletred2
palevioletred3
palevioletred4
papayawhip
peachpuff
peachpuff1
peachpuff2
peachpuff3
peachpuff4
peru
pink
pink1
pink2
pink3
pink4

plum
plum1
plum2
plum3
plum4
powderblue
purple
purple1
purple2
purple3
purple4
red
red1
red2
red3
red4
rosybrown
rosybrown1
rosybrown2
rosybrown3
rosybrown4
royalblue
royalblue1
royalblue2
royalblue3
royalblue4
saddlebrown
salmon
salmon1
salmon2
salmon3
salmon4
sandybrown
seagreen
seagreen1
seagreen2
seagreen3
seagreen4
seashell
seashell1
seashell2
seashell3
seashell4
sienna
sienna1
sienna2
sienna3
sienna4
skyblue
skyblue1
skyblue2
skyblue3
skyblue4
slateblue
slateblue1
slateblue2
slateblue3
slateblue4
slategray
slategray1

slategray2
slategray3
slategray4
slategrey
snow
snow1
snow2
snow3
snow4
springgreen
springgreen1
springgreen2
springgreen3
springgreen4
steelblue
steelblue1
steelblue2
steelblue3
steelblue4
tan
tan1
tan2
tan3
tan4
thistle
thistle1
thistle2
thistle3
thistle4
tomato
tomato1
tomato2
tomato3
tomato4
turquoise
turquoise1
turquoise2
turquoise3
turquoise4
violet
violetred
violetred1
violetred2
violetred3
violetred4
wheat
wheat1
wheat2
wheat3
wheat4
whitesmoke
yellow
yellow1
yellow2
yellow3
yellow4
yellowgreen

B. Horn, Applied R Code, www.applied-r.com

Practice

In the data folder there are three practice files: barley_uniformity.csv, peanut_uniformity.csv, and tomato_uniformity.csv. Practice creating histograms using both the basic and ggplot methods. To start you off, here is the command to load the barley file.

```
barley = read.csv("data/barley_uniformity.csv")
```

Remember to use CTRL+ALT+I to insert new code chunks below.