# Presenting Data: Plotting Conditional Means

The idea when plotting conditional means is to show how the outcome, or variable of interest, varies as a function of predictors.

Today we'll be working with a dataset from IBM which provide a standard HR dataset, which we can use to predict attrition. (*NB this is simulated data, no employee information is being disclosed*) Attrition in this case is defined as an employee leaving without being fired or retiring. Companies generally attempt to avoid attrition, as it's very expensive to search for and hire a replacement-- better in general to keep the employees you have, provided they are doing their jobs. This means that it's important to predict who might leave in a given year. This information can be used in a targeted way in order to focus resources on the employees most likely to leave.

## Setup for plotting conditional means

We start with a standard set of setup commands. Today we'll be working with tidyverse, as usual, along with a library called forcats which helps us to deal with the dreaded factor variables. To handle colors, we'll need the package RColorBrewer.

Next we load in the data, using the readr package. Note that this data is saved as comma separated or csv. This is an easy file format to recognize. When we use the readr package, it gives us some output that says how it interprets the data-- is it a string variable, numeric (float), integer and so on.

## Loading Data

load("attrition.Rdata")

Today, our primary outcome of interest will be attrition. This is a binary variable that is currently encoded as text-- "Yes" or "No." We need to encode it as a binary variable with 1 meaning yes and 0 meaning no. After recoding, we need to make sure that the new variable looks correct.

## Crate a new variable named attrit and define it as 0  
at<-at%>%mutate(attrit=ifelse(Attrition=="Yes",1,0))  
  
table(at$Attrition)

##   
## No Yes   
## 1233 237

table(at$attrit)

##   
## 0 1   
## 1233 237

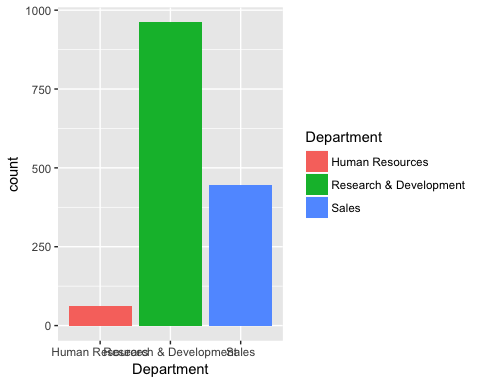
table(at$attrit,at$Attrition)

##   
## No Yes  
## 0 1233 0  
## 1 0 237

## Univariate Graphics

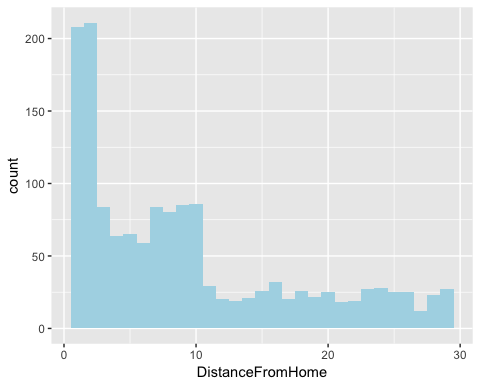
Univariate graphics help us understand what individual variables look like-- how are they distibuted across the sample? Here's a quick rundown on some univariate graphics. Say we wanted a quick count of who was in each department. We can use geom\_bar to get this done. By default, this will give us a count in each department.

gg<-ggplot(at,aes(x=Department,fill=Department))  
gg<-gg+geom\_bar()  
gg



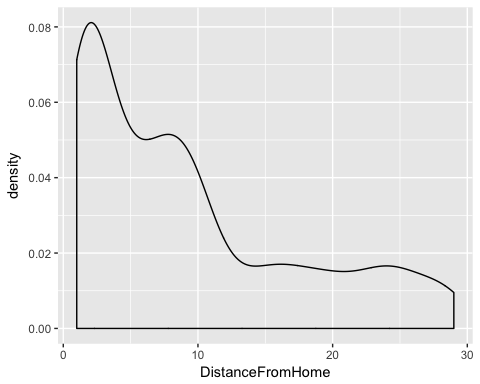
The next univariate graphic you should know is for continuous variables. The first thing you generally want is a histogram.

gg<-ggplot(at,aes(x=DistanceFromHome))  
gg<-gg+geom\_histogram(binwidth = 1,fill="lightblue")  
gg

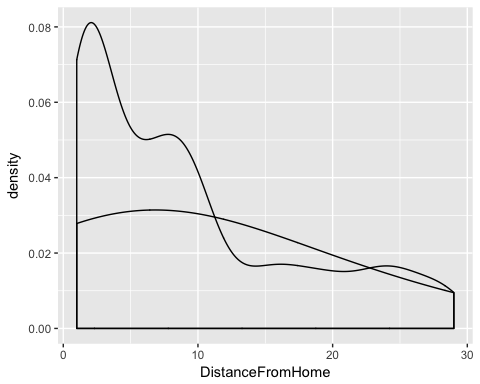


Density plots provide a continous graphic of the distribution of a variable:

gg<-ggplot(at,aes(x=DistanceFromHome))  
gg<-gg+geom\_density()  
gg



## Changing bandwidth-- not recommended, just showing you how.   
gg<-gg+geom\_density(bw=10)  
gg



## Predicting Attrition

Our first prediction will use business travel as a predictor for attrition. There are three categories here-- non travel, travel infrequently, and frequent travel. We'll calculate levels of attrtion at each level and then take a look at the data.

at\_sum<-at%>%  
 group\_by(BusinessTravel)%>%  
 summarize(attr\_avg=mean(attrit))  
  
at\_sum

## # A tibble: 3 x 2  
## BusinessTravel attr\_avg  
## <chr> <dbl>  
## 1 Non-Travel 0.0800000  
## 2 Travel\_Frequently 0.2490975  
## 3 Travel\_Rarely 0.1495686

Remember that the mean of a binary variable indicates the proportion of the population that has a certain characteristcs. So, in our case, 0.25 of the sample that travels frequently left the company in the last year. Our first plot will be a basic bar plot, showing the average levels of attrition.

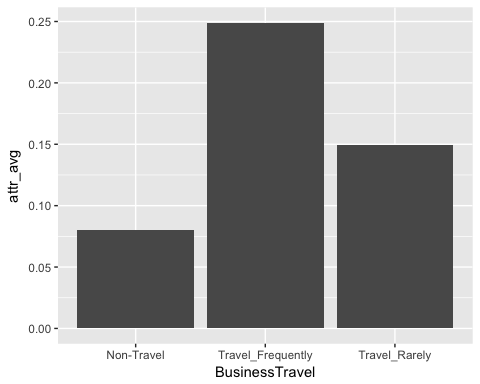
To get started, let's see what this looks like in a table.

at %>%  
 count(BusinessTravel,attrit) %>%  
 group\_by(BusinessTravel)%>%  
 mutate(prop = prop.table(n)) %>%  
 select(-n) %>%  
 spread(attrit, prop)%>%kable()

|  |  |  |
| --- | --- | --- |
| BusinessTravel | 0 | 1 |
| Non-Travel | 0.9200000 | 0.0800000 |
| Travel\_Frequently | 0.7509025 | 0.2490975 |
| Travel\_Rarely | 0.8504314 | 0.1495686 |

Now we can plot it:

## Bar Plot with aesthetics: mean attrition as height, business travel as cateogry  
gg<-ggplot(at\_sum,aes(x=BusinessTravel,y=attr\_avg))  
## Use bar plot geometry, height of bars set by level observed in dataset  
gg<-gg+geom\_bar(stat="Identity")  
## Print  
gg

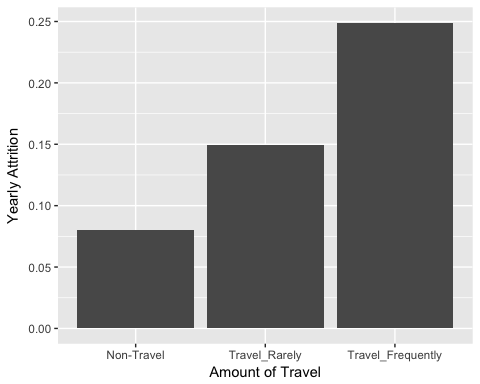


This is fine, but it should really be in the order of the underlying variable. We can use fct\_reorder to do this.

*Side* *Note*

What is a factor variable? In R, factor variables are used for categorical data. These are data elements that can take on one and only one value of a mutually exclusive and exhaustive list of elements. In our case, the travel variable is a factor-- employees can be in Non-Travel, Travel Frequently or Travel Rarely bins. Everyone is one bin, and the bins cover all possible options. We use factors when numbers won't work-- for characteristics like race or religion or political affiliation.

## Same asethetics, but now orderred by level  
gg<-ggplot(at\_sum,aes(x=fct\_reorder(BusinessTravel,attr\_avg),y=attr\_avg))  
  
gg<-gg+geom\_bar(stat="identity")  
  
## Labeling  
gg<-gg+xlab("Amount of Travel")+ylab("Yearly Attrition")  
##Print  
gg



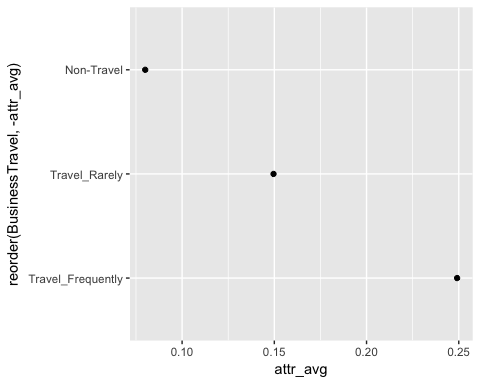
*Quick Exercise: Create a bar plot showing average attrition by department instead of travel*

A dot plot can be a good way of displaying conditional means as well. Many times dot plots are more easily understood if they are horizontal, so we'll use coord\_flip to make it horizontal.

at\_sum<-at%>%  
 group\_by(BusinessTravel)%>%  
 summarize(attr\_avg=mean(attrit))  
  
at\_sum

## # A tibble: 3 x 2  
## BusinessTravel attr\_avg  
## <chr> <dbl>  
## 1 Non-Travel 0.0800000  
## 2 Travel\_Frequently 0.2490975  
## 3 Travel\_Rarely 0.1495686

## Now a dot plot  
gg<-ggplot(at\_sum,aes(x=reorder(BusinessTravel,-attr\_avg),y=attr\_avg))  
gg<-gg+geom\_point()  
gg<-gg+coord\_flip()  
gg



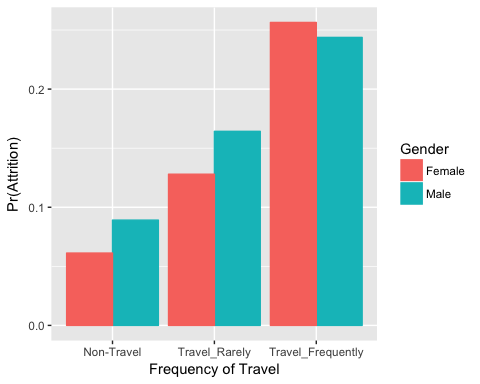
## Conditional means using two predictors

We can use graphics to display conditonal means at multiple levels of predictor levels. There are a couple of ways to get this done. When using bar plots we've got two basic tools: location and color. In the first example, we're going to plot attrition by travel and gender, We'll use color to indicate gender, and location to indicate travel.

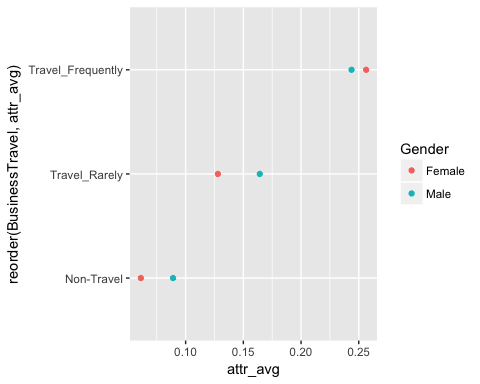
## Summarize attrition by travel AND gender  
at\_sum<-at%>%  
 group\_by(BusinessTravel,Gender)%>%  
 summarize(attr\_avg=mean(attrit))  
  
## Get the results  
at\_sum

## # A tibble: 6 x 3  
## # Groups: BusinessTravel [?]  
## BusinessTravel Gender attr\_avg  
## <chr> <chr> <dbl>  
## 1 Non-Travel Female 0.06122449  
## 2 Non-Travel Male 0.08910891  
## 3 Travel\_Frequently Female 0.25641026  
## 4 Travel\_Frequently Male 0.24375000  
## 5 Travel\_Rarely Female 0.12796209  
## 6 Travel\_Rarely Male 0.16425121

## PLot it using a bar plot  
gg<-ggplot(at\_sum,aes(x=fct\_reorder(BusinessTravel,attr\_avg),y=attr\_avg,color=Gender))  
gg<-gg+geom\_bar(stat="identity",aes(fill=Gender),position="dodge")  
gg<-gg+ylab("Pr(Attrition)")+xlab("Frequency of Travel")  
gg



## Plot it using a dot plot  
gg<-ggplot(at\_sum,aes(x=reorder(BusinessTravel,attr\_avg),y=attr\_avg),color=Gender)  
gg<-gg+geom\_point(aes(color=Gender))  
gg<-gg+coord\_flip()  
gg



* Quick Exercise: Create either a bar plot or a dot plot showing attrition by department AND field of education \*

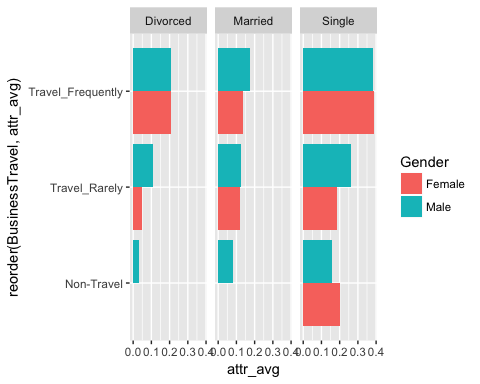
## More Variables: faceting

We can continue this logic with three variables. Now we're going to summarize by Travel, Gender and Marital status. Here we're going to use an additional tool in our arsenal: Faceting. Faceting means making multiple graphs with the same structure. In the code below, we will arrange positions based on travel, color based on gender, and then split the graphic by marital status.

at\_sum<-at%>%  
 group\_by(BusinessTravel,Gender,MaritalStatus)%>%  
 summarize(attr\_avg=mean(attrit))  
  
at\_sum

## # A tibble: 18 x 4  
## # Groups: BusinessTravel, Gender [?]  
## BusinessTravel Gender MaritalStatus attr\_avg  
## <chr> <chr> <chr> <dbl>  
## 1 Non-Travel Female Divorced 0.00000000  
## 2 Non-Travel Female Married 0.00000000  
## 3 Non-Travel Female Single 0.20000000  
## 4 Non-Travel Male Divorced 0.03030303  
## 5 Non-Travel Male Married 0.08333333  
## 6 Non-Travel Male Single 0.15625000  
## 7 Travel\_Frequently Female Divorced 0.20833333  
## 8 Travel\_Frequently Female Married 0.13636364  
## 9 Travel\_Frequently Female Single 0.38775510  
## 10 Travel\_Frequently Male Divorced 0.20512821  
## 11 Travel\_Frequently Male Married 0.17567568  
## 12 Travel\_Frequently Male Single 0.38297872  
## 13 Travel\_Rarely Female Divorced 0.04878049  
## 14 Travel\_Rarely Female Married 0.12195122  
## 15 Travel\_Rarely Female Single 0.18518519  
## 16 Travel\_Rarely Male Divorced 0.10869565  
## 17 Travel\_Rarely Male Married 0.12714777  
## 18 Travel\_Rarely Male Single 0.26041667

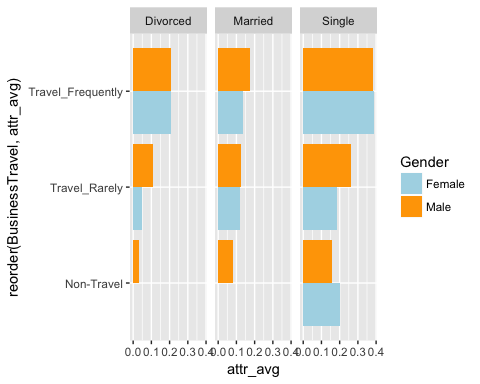
gg<-ggplot(at\_sum,aes(x=reorder(BusinessTravel,attr\_avg),  
 y=attr\_avg,  
 fill=Gender))  
## Bar plot, with unstacked (dodge)  
 gg<-gg+geom\_bar(stat="identity",position="dodge")  
## Separate out by Marital Status  
gg<-gg+facet\_wrap(~MaritalStatus)  
## Change orientation to sideways  
gg<-gg+coord\_flip()  
## Print  
gg



*Side* *Note*: Colors

What if we want to change colors? This is a little tricky for most people at first. ggplot thinks in terms of palettes, so you need to associate a palette with a characteristics of the graphic. Below, I replace the default palette with my own ugly one.

## Changing Colors  
mypal<-c("lightblue","orange")  
  
gg<-gg+scale\_fill\_manual(values =mypal )  
## Print  
gg

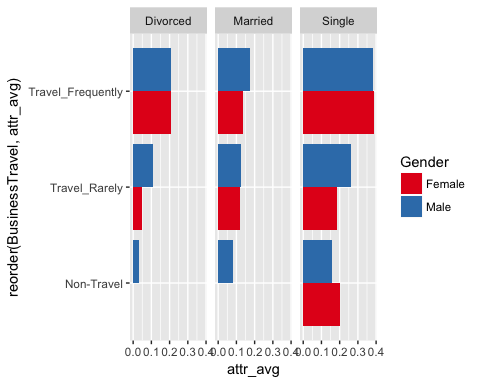


You can also use RColorBrewer which has a wide variety of palettes already built. Below I use the qualitative palette creatively named "Set1".

## Another way, using color brewer palettes:   
gg<-gg+scale\_fill\_brewer(palette = "Set1")

## Scale for 'fill' is already present. Adding another scale for 'fill',  
## which will replace the existing scale.

gg



*Quick Exercise: Plot predicted attrition by Education Field, Department and Gender*

## Multiple Predictors for Conditional Means

One solution is to use facets, or lots of little graphs, which show how the pattern varies across different groups. In this case, our groups will be defined by gender and work/life balance.

