Plots

Saria Awadalla

6/2/2021

## Overview of Plots in R

* The plot() function in Base R is a powerful tool for creating informative, high quality graphs and plots.
* The function’s behavior can vary depending on the type of input provided. At a basic level, that is, when provided with one or more vectors of data, plot() creates a scatterplot showing the position and magnitude of each data point; x and y coordinates.
* It also creates a base or canvas for additional graphical layers such as regression lines, text, etc. It also allows for the inclusion of plot titles, captions, and legends.
* The package ggplot2, which is part of the tidyverse, enhances the functionality of the plot function, and is streamlined to work with other data management methods in tidyr and dplyr.

In this lecture, we will focus primarily on ggplot instead of plot. Methods to graphically describe univariate, bivariate or multivariate data - both numeric and categorical - will be discussed. To illustrate these methods, we will use the NHANES data example discussed in the previous set of lectures.

### Univariate Data

Individual variable characteristics are summarized graphically

# Numeric Data

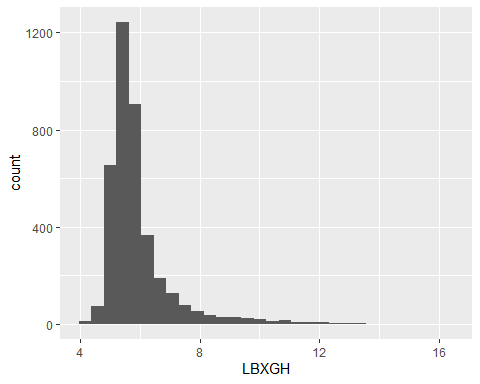
Example: A1c

* Histograms are used to show the distributional shape of a numeric variables. They provide a graphical estimate of the likelihood of each possible value observed in the data by binning values into different non-overlapping ranges, and calculating the frequency (or proportion) of each bin.
* Histograms can be used to assess modality, skewness (vs symmetry), spread (somewhat) and are often used to determine whether a certain parametric distribution (e.g., normality) is appropriate given the data.

# default output is bin counts  
ggplot(nhanesdata2, aes(x = LBXGH)) +  
 geom\_histogram()

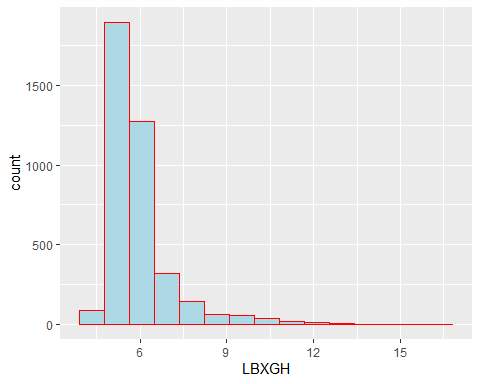
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 177 rows containing non-finite values (stat\_bin).



# additional arguments  
ggplot(nhanesdata2, aes(x = LBXGH)) +  
 geom\_histogram(bins = 15,  
 color = 'red',  
 fill = 'lightblue')

## Warning: Removed 177 rows containing non-finite values (stat\_bin).

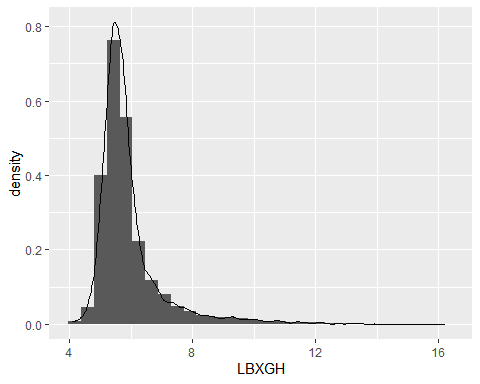


# density/proportion instead of counts  
ggplot(nhanesdata2, aes(x = LBXGH)) +  
 geom\_histogram(aes(y = ..density..)) + # aesthetics can be added in ggplot as well  
 geom\_density()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 177 rows containing non-finite values (stat\_bin).

## Warning: Removed 177 rows containing non-finite values (stat\_density).



Another approach to numeric data is the box-plot. This plot is not of the data directly, but rather a graphical representation of the five-number summary (min, Q1, median, Q3, max). Box-plots are better suited for comparing a numeric variable across multiple groups.

p1 = ggplot(nhanesdata2, aes(y = LBXGH)) +  
 geom\_boxplot()  
p1

## Warning: Removed 177 rows containing non-finite values (stat\_boxplot).

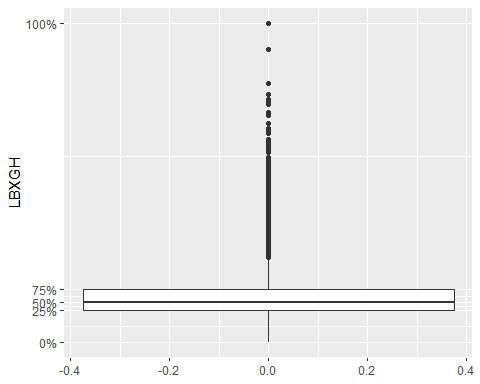


summary(nhanesdata2$LBXGH)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 4.100 5.300 5.600 5.937 6.100 16.200 177

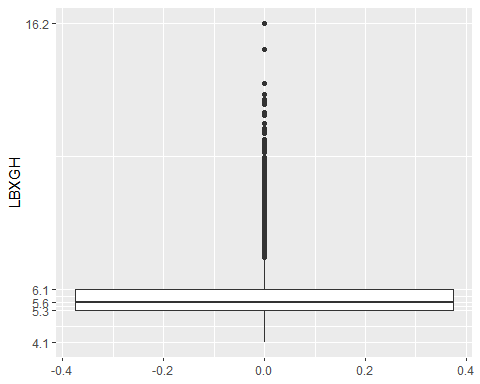
p1 + scale\_y\_continuous(breaks = quantile(nhanesdata2$LBXGH,   
 probs = seq(0,1,.25), na.rm = T))

## Warning: Removed 177 rows containing non-finite values (stat\_boxplot).



p1 + scale\_y\_continuous(breaks = as.vector(quantile(nhanesdata2$LBXGH,   
 probs = seq(0,1,.25), na.rm = T)))

## Warning: Removed 177 rows containing non-finite values (stat\_boxplot).

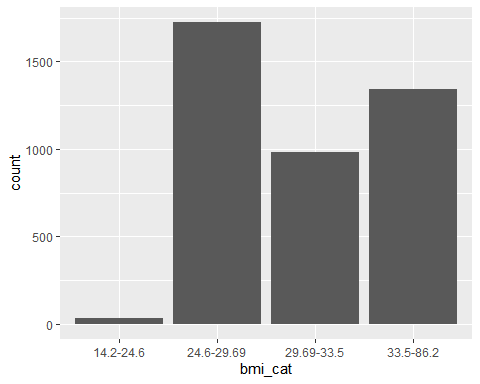


## Categorical Data

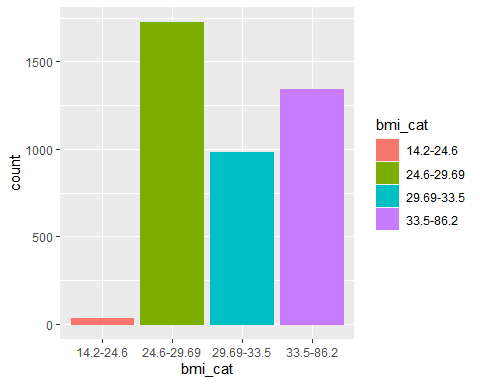
The primary plot used for categorical data is the bar plot. It is similar to a histogram, however the bars represent discrete values corresponding to each level of the variable.

Example: BMI Categories

nhanesdata2 %>%  
 ggplot(aes(x = bmi\_cat)) +  
 geom\_bar()



# we can map color/fill to a variable (aes)  
nhanesdata2 %>%  
 ggplot(aes(x = bmi\_cat)) +  
 geom\_bar(aes(color=bmi\_cat, fill=bmi\_cat))



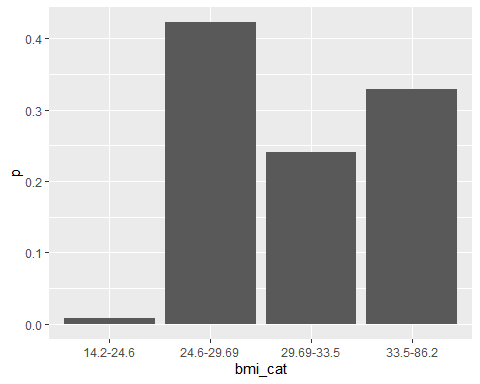
Note that the default metric in a bar plot is counts. In many cases the proportion or percentage is more appropriate. There are many ways to display the latter, but the most efficient is to use ggplot in combination with the summarise function in dplyr.

(p2 = nhanesdata2 %>%  
 group\_by(bmi\_cat) %>%  
 summarise(n = n()) %>%  
 mutate(p = n/sum(n)) %>%  
 ggplot())

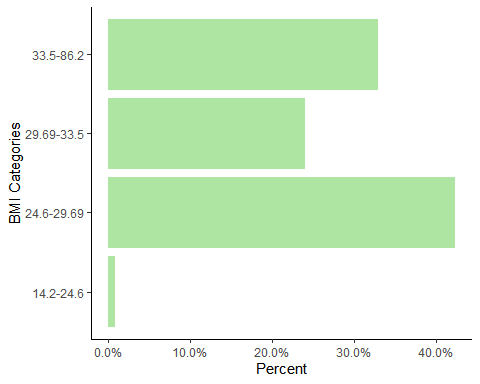
## `summarise()` ungrouping output (override with `.groups` argument)



p2 + geom\_col(aes(x=bmi\_cat, y=p))



p2 + geom\_col(aes(x=bmi\_cat, y=p), fill="#35bd17", alpha=0.4) +   
 scale\_y\_continuous(labels = scales::percent) +   
 xlab("BMI Categories") + ylab("Percent") +   
 coord\_flip() +  
 theme\_classic()



Exercise: geom\_errorbar can be used to include an indicator of precision to a plot. This function requires two values: the lower and upper limits of the error bar. These can be based on confidence intervals, standard errors, or some other measure of variability. For proportions, the standard error is given by sqrt(n \* p \* (1-p)). Use this quantity to create error bars for the proportions plot above.

## Bivariate Plots

The type of bivariate plot depends on the combination of variables used:

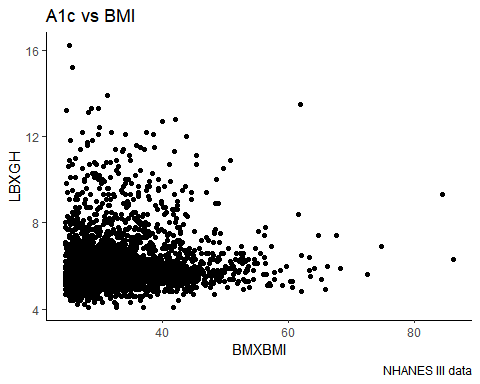
* Numeric vs numeric: scatter plot
* Numeric vs categorical: box-plot or histogram/density+facets
* Categorical vs categorical: bar-plot or column plot

Usually, we map select aesthetics (x, color, fill, etc.) of the main plot to the secondary variable.

# Scatter plot

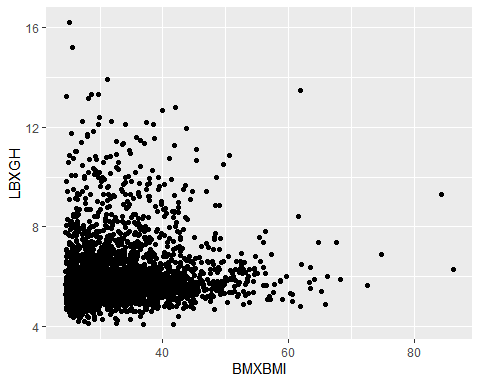
p3 = nhanesdata2 %>%  
 ggplot()  
  
p3 + geom\_point(aes(x=BMXBMI, y=LBXGH)) +   
 labs(title = "A1c vs BMI", caption = "NHANES III data") +  
 theme\_classic()

## Warning: Removed 177 rows containing missing values (geom\_point).



# alternatively, jitters can be used to 'dodge' points  
p3 + geom\_jitter(aes(x=BMXBMI, y=LBXGH))

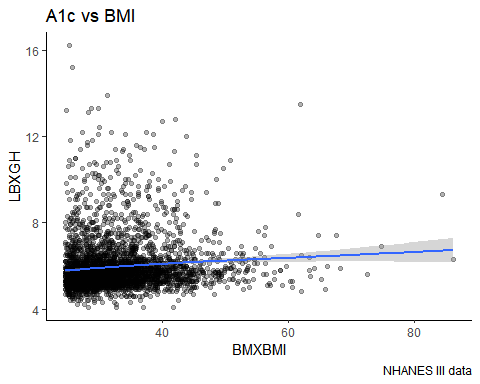
## Warning: Removed 177 rows containing missing values (geom\_point).



# Often a trend line is included to estimate the relationship between variables. This can be based on a Lowess smoother (default) or a regression function.  
p3 + geom\_point(aes(x=BMXBMI, y=LBXGH), alpha=0.3) +  
 geom\_smooth(aes(x=BMXBMI, y=LBXGH)) +  
 labs(title = "A1c vs BMI", caption = "NHANES III data") +  
 theme\_classic()

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

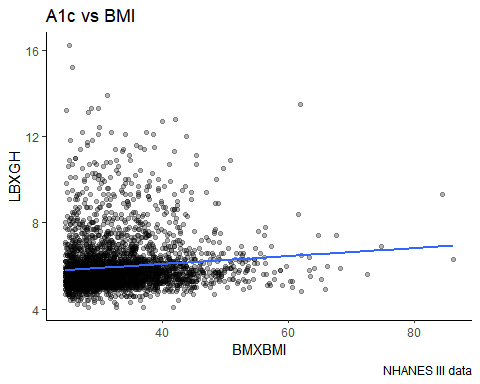
## Warning: Removed 177 rows containing non-finite values (stat\_smooth).  
  
## Warning: Removed 177 rows containing missing values (geom\_point).



p3 + geom\_point(aes(x=BMXBMI, y=LBXGH), alpha=0.3) +  
 geom\_smooth(aes(x=BMXBMI, y=LBXGH), method = 'lm', se = F) +  
 labs(title = "A1c vs BMI", caption = "NHANES III data") +  
 theme\_classic()

## `geom\_smooth()` using formula 'y ~ x'

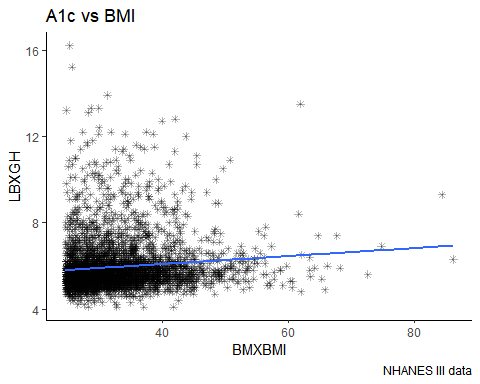
## Warning: Removed 177 rows containing non-finite values (stat\_smooth).  
  
## Warning: Removed 177 rows containing missing values (geom\_point).



nhanesdata2 %>%  
 ggplot(aes(x=BMXBMI, y=LBXGH)) +  
 geom\_point(alpha=0.3, shape=8) +  
 geom\_smooth(method = 'lm', se = F) +  
 labs(title = "A1c vs BMI", caption = "NHANES III data") +  
 theme\_classic()

## `geom\_smooth()` using formula 'y ~ x'

## Warning: Removed 177 rows containing non-finite values (stat\_smooth).  
  
## Warning: Removed 177 rows containing missing values (geom\_point).

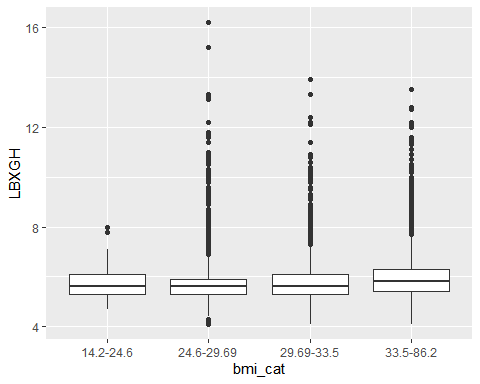


# Box-plots and Violin Plots

Box-plots can be used to display stratified summary statistics as a means of comparing the distribution of a numeric variable across multiple categories. Violin plots alternatively show the raw distribution across multiple groups (similar to a histogram).

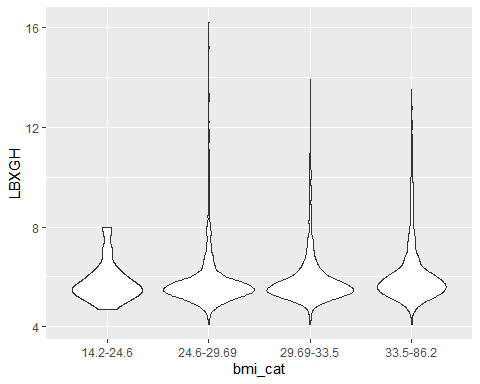
p4 = nhanesdata2 %>%  
 ggplot(aes(y=LBXGH, x=bmi\_cat))  
  
p4 + geom\_boxplot()

## Warning: Removed 177 rows containing non-finite values (stat\_boxplot).



p4 + geom\_violin()

## Warning: Removed 177 rows containing non-finite values (stat\_ydensity).



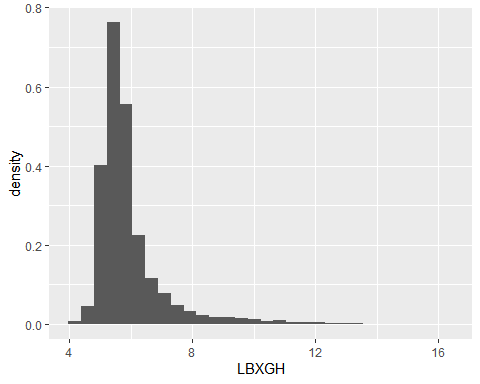
# Histogram+facets

Facets are a ggplot feature for displaying multiple individual figures in a single plot. This is used when the plot doesn’t allow for stratification (e.g., histogram).

p5 = nhanesdata2 %>%  
 ggplot(aes(x=LBXGH)) +  
 geom\_histogram(aes(y=..density..))  
  
p5

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

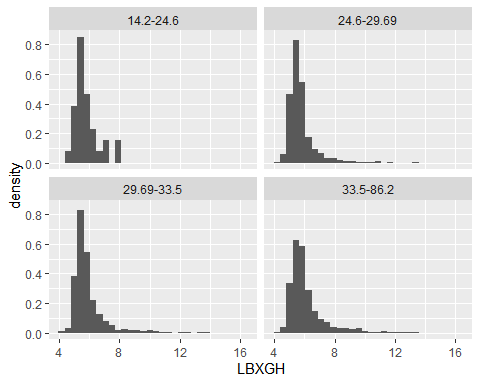
## Warning: Removed 177 rows containing non-finite values (stat\_bin).



p5 + facet\_wrap(~bmi\_cat)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

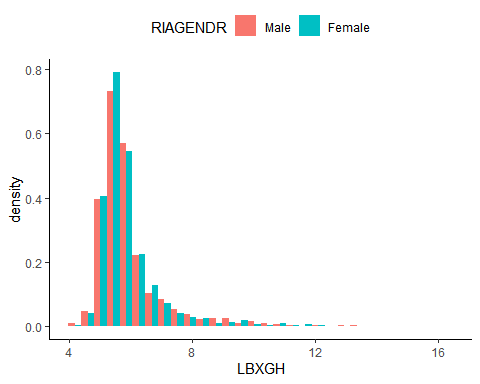
## Warning: Removed 177 rows containing non-finite values (stat\_bin).



# Alternatively, we can display all histograms in a single figure  
nhanesdata2 %>%  
 ggplot(aes(x=LBXGH, fill=RIAGENDR)) +  
 geom\_histogram(aes(y=..density..), position = 'dodge') +  
 theme\_classic() +  
 theme(legend.position="top")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

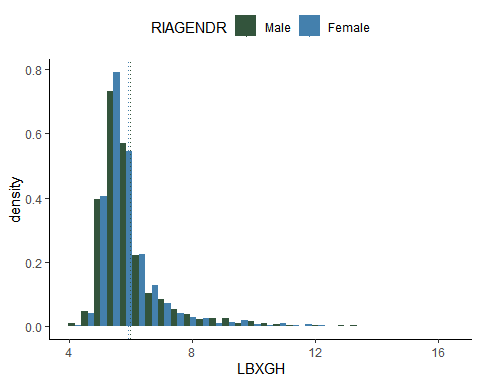
## Warning: Removed 177 rows containing non-finite values (stat\_bin).



# Add vertical lines showing gender-specific means  
nhanesdata2 %>%  
 group\_by(RIAGENDR) %>%  
 mutate(grp.mean = mean(LBXGH, na.rm=T)) %>%  
 ggplot(aes(x=LBXGH, fill=RIAGENDR)) +  
 geom\_histogram(aes(y=..density..), position = 'dodge') +  
 geom\_vline(aes(xintercept=grp.mean, color=RIAGENDR), linetype=3) +  
 scale\_fill\_manual(values = c('#33543c', '#4480ad')) +  
 scale\_color\_manual(values = c('#33543c', '#4480ad')) +  
 theme\_classic() +  
 theme(legend.position="top")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

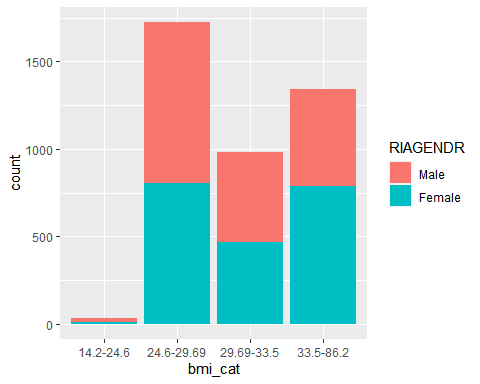
## Warning: Removed 177 rows containing non-finite values (stat\_bin).



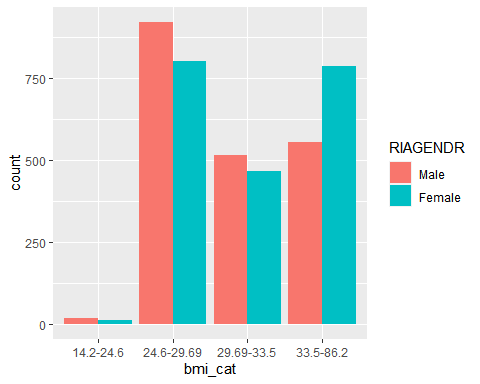
## Bar and column plots

Similar to the univariate, case we can extend bar and column plots to display the relationships between two categorical variables. We can also use facets much in the same way we did for histograms.

nhanesdata2 %>%  
 ggplot(aes(x=bmi\_cat, fill=RIAGENDR)) +  
 geom\_bar()

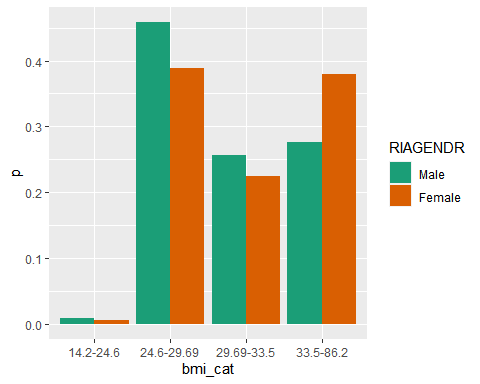


nhanesdata2 %>%  
 ggplot(aes(x=bmi\_cat, fill=RIAGENDR)) +  
 geom\_bar(position = 'dodge')



nhanesdata2 %>%  
 group\_by(RIAGENDR, bmi\_cat) %>%  
 summarise(n = n()) %>%  
 mutate(p = n/sum(n)) %>%  
 ggplot(aes(x=bmi\_cat, y=p, fill=RIAGENDR)) +  
 geom\_col(position = 'dodge') +  
 scale\_fill\_brewer(palette="Dark2")

## `summarise()` regrouping output by 'RIAGENDR' (override with `.groups` argument)

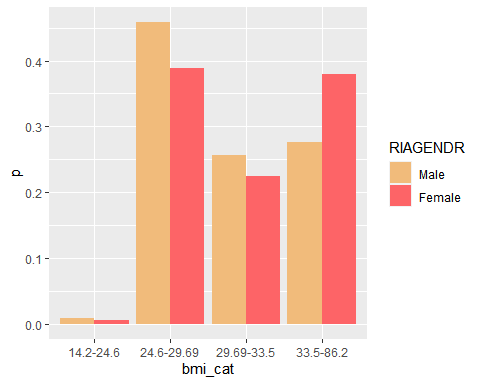


#   
#install.packages("wesanderson")  
# Load  
library(wesanderson)

## Warning: package 'wesanderson' was built under R version 4.0.5

nhanesdata2 %>%  
 group\_by(RIAGENDR, bmi\_cat) %>%  
 summarise(n = n()) %>%  
 mutate(p = n/sum(n)) %>%  
 ggplot(aes(x=bmi\_cat, y=p, fill=RIAGENDR)) +  
 geom\_col(position = 'dodge') +  
 scale\_fill\_manual(values=wes\_palette(n=2, name="GrandBudapest1"))

## `summarise()` regrouping output by 'RIAGENDR' (override with `.groups` argument)



For additional resources, check out this excellent wiki by STHDA:

<http://www.sthda.com/english/wiki/ggplot2-essentials>

Scroll down for additional packages that enhance the features of ggplot2.

Also, the help menu in RStudio contains a graphics cheat sheet for a quick reference guide.