# Data Visualization with R

# Why ggplot2?

- Follows a grammar, just like any language.

- It defines basic components that make up a sentence. In this case, the grammar defines components in a plot.

- Supports a continuum of expertise.

- Get started right away but with practice you can effortless build complex, publication quality figures.

# Some terminology

- ggplot - The main function where you specify the dataset and variables to plot

- geoms - geometric objects
  geom point(), geom bar(), geom density(), geom line(),
  geom area()

- aes - aesthetics
  shape, transparency (alpha), color, fill, linetype.

- scales - Define how your data will be plotted
  continuous, discrete, log

# Categorical(Factor) Variables

- Categorical variables place cases into groups. Each group has a label called a level. By default, R orders the levels alphabetically.

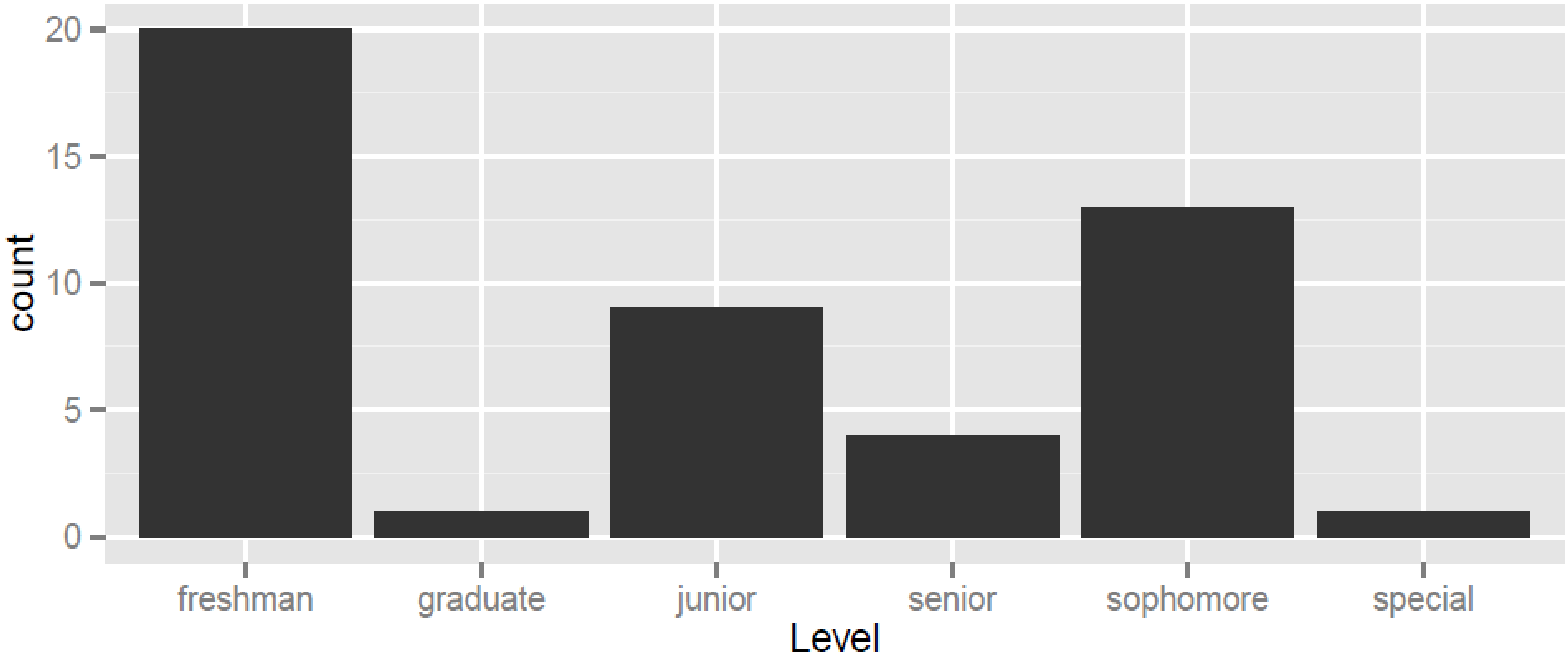
- Bar graphs (or bar charts) are the best way to display categorical variables.

# Plotting 1 Categorical(Factor) Variable

```
with(students, table(Level))
```

```
## Level
               freshman     sophomore     junior   senior   special   graduate
               13           3             2        0        0         1
```

```
ggplot(students, aes(x = Level)) + geom_bar()
```

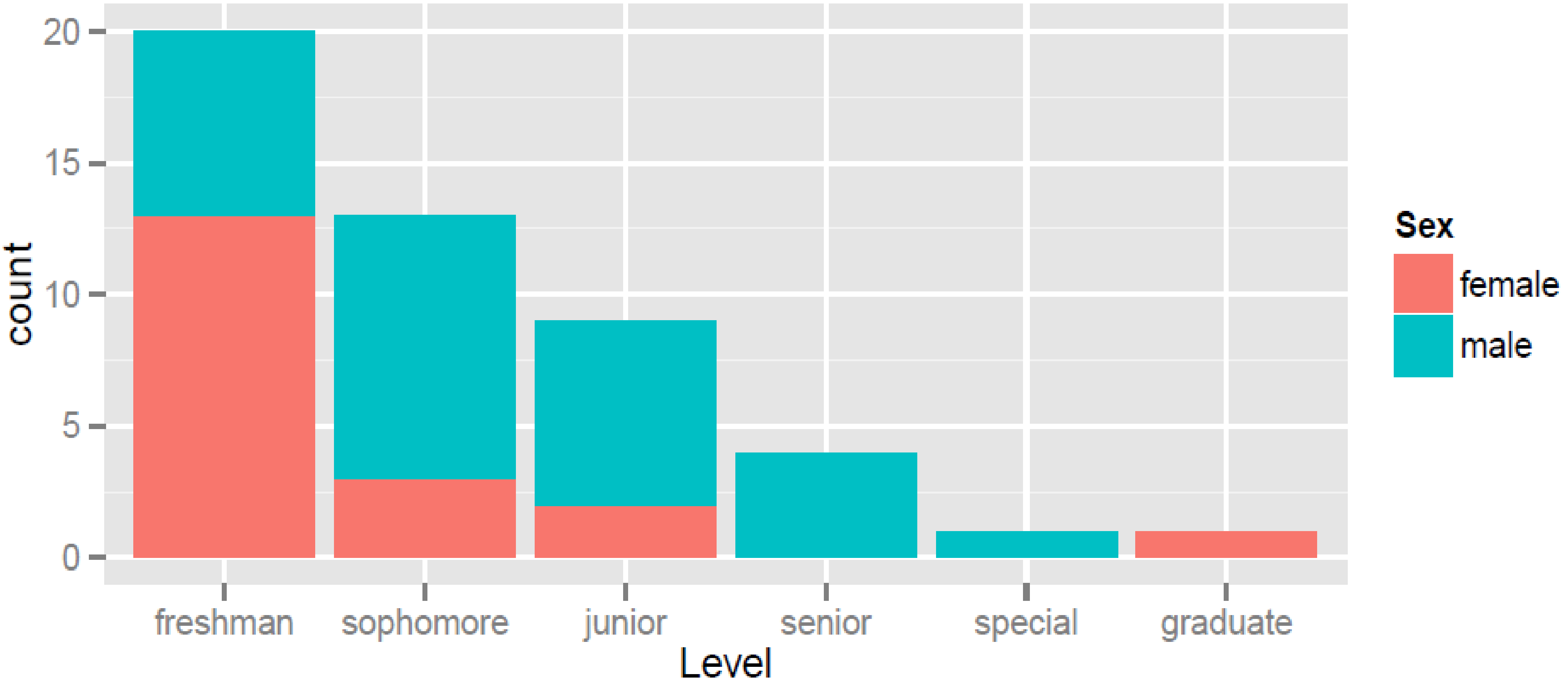


# Plotting 2 Categorical(Factor) Variables

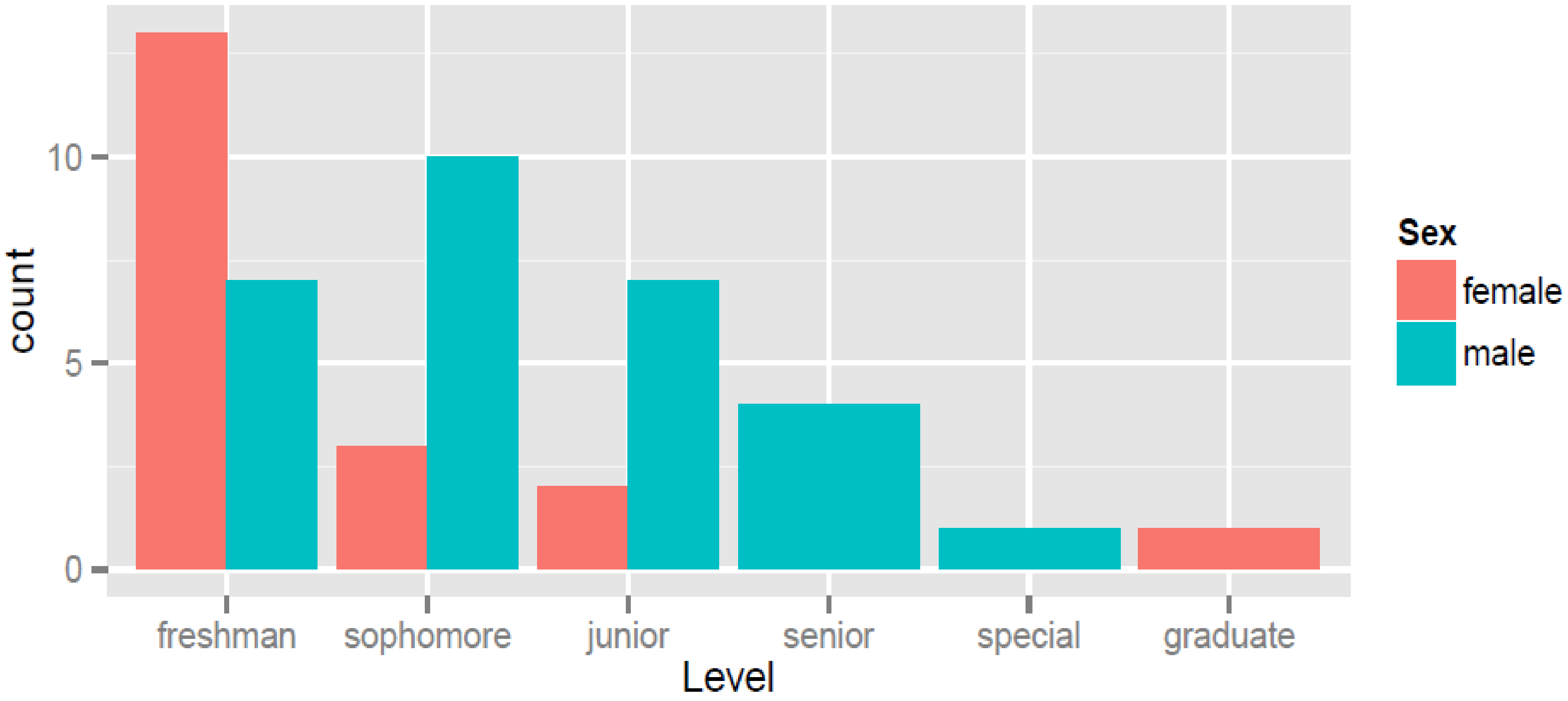
Examine the gender distribution by level

```
with(students, table(Sex, Level))

##              Level
## Sex          freshman     sophomore     junior   senior   special   graduate
## female       13           3             2        0        0         1
## male         7            10            7        4        1         0
```

```
ggplot(students, aes(x = Level, fill = Sex)) + geom_bar()
```



```
ggplot(students, aes(x = Level, fill = Sex)) +
                              geom_bar(position = "dodge")
```



# One Quantitative Variable

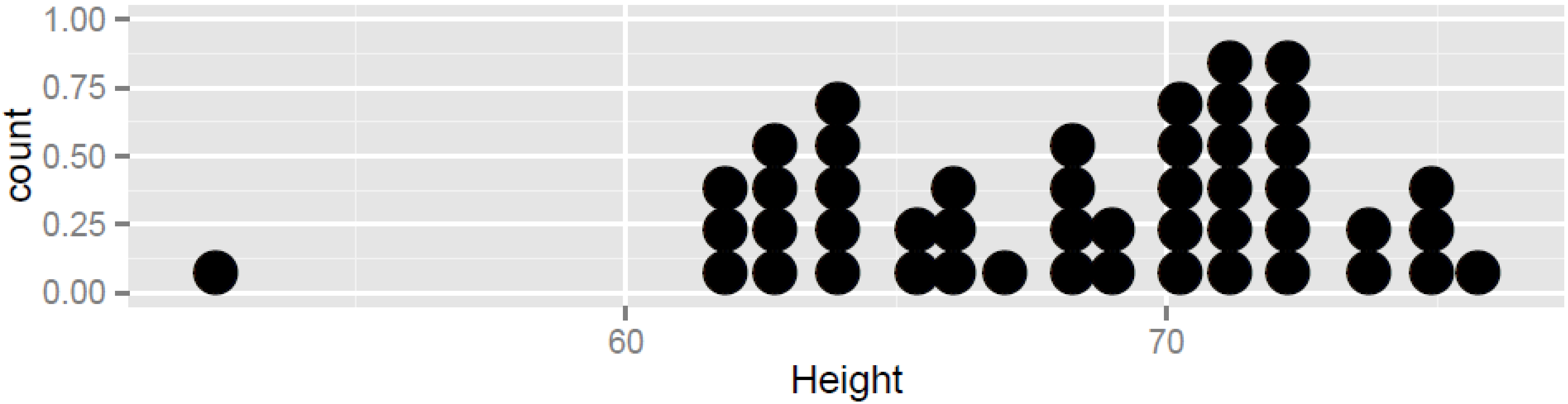
There are multiple ways to graphically display the distribution of a single quantitative variable.
They differ in how clearly they show various features of the distribution:

- Dot Plots
- Histogram Plots
- Density Plots
- Box and Whisker Plots

# Dot plots

For small data sets, a dot plot is an effective way to visualize all of the data. A dot is placed at the appropriate value on the x axis for each case, and dots stack up.

ggplot(students, aes(x = Height)) + geom_dotplot()



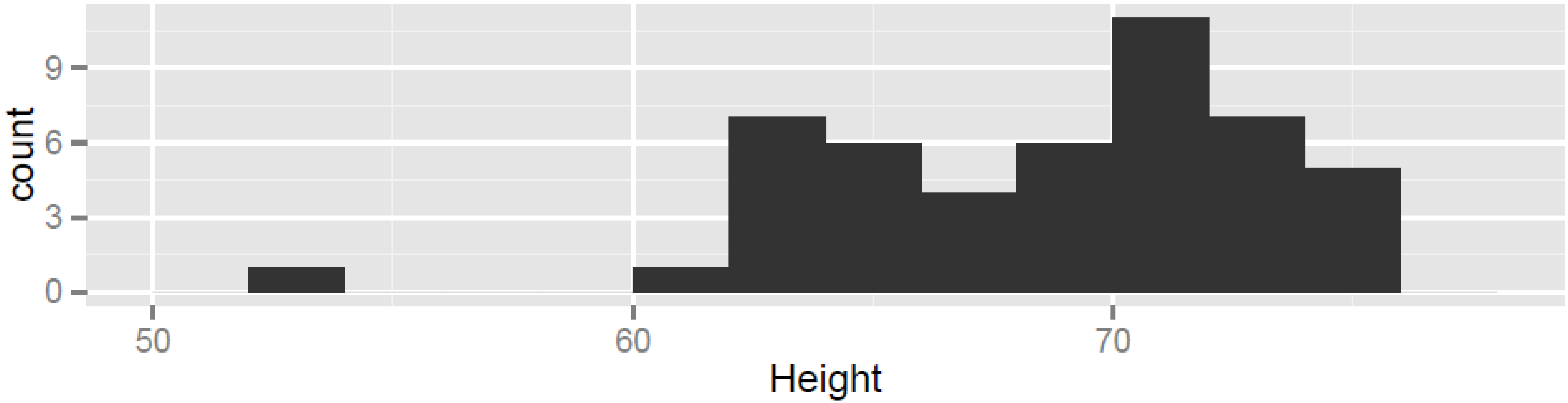
table(students$Height)

# Histograms

Another popular way to graph a single quantitative variable is with a histogram. Just use geom_histogram() instead. There are ways to specify the breakpoints. It is often useful to specify the binwidth and/or origin manually.

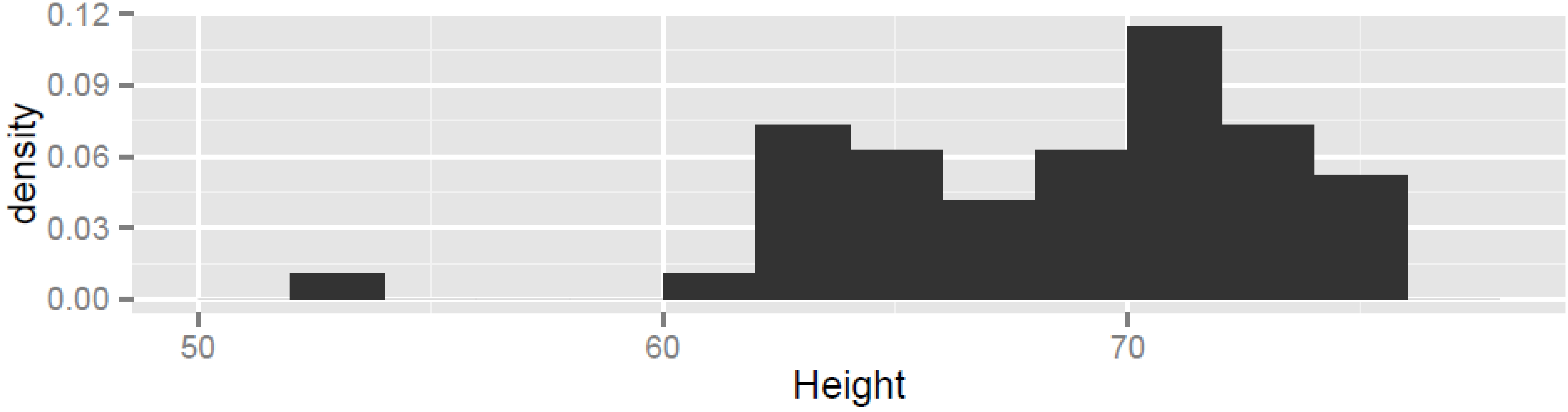
Another common way to present histograms is so that the total area is one, so that the area of a bin represents the proportion of data in the corresponding interval. Adding an aesthetic y=..density.. makes this change.

ggplot(students, aes(x = Height)) + geom_histogram(binwidth = 2)



ggplot(students, aes(x = Height)) + geom_histogram(binwidth = 1,
colour="black",fill="white")

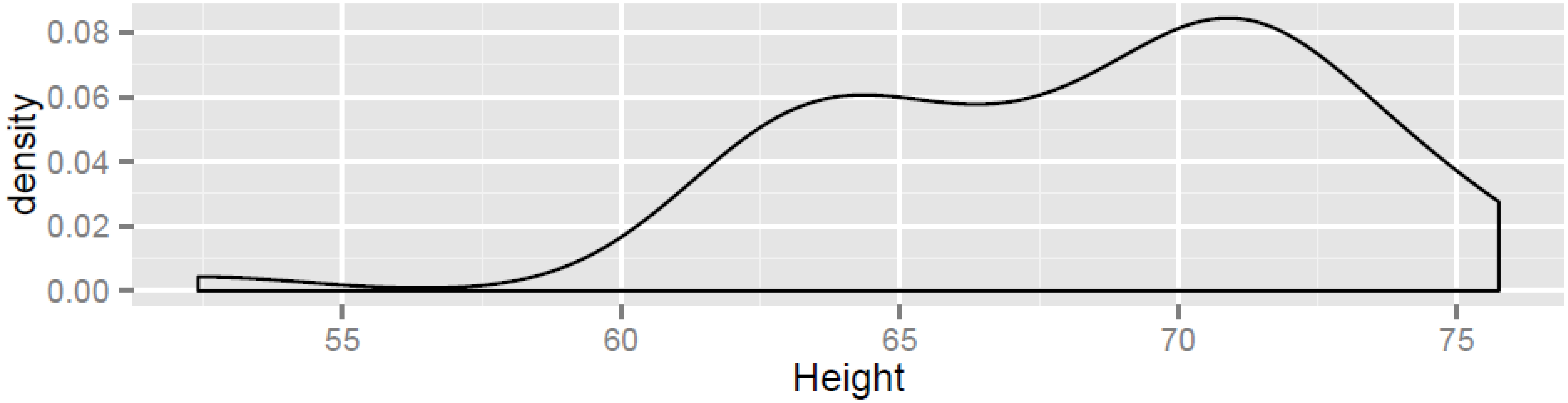
```
ggplot(students, aes(x = Height)) + geom_histogram(binwidth =
2, aes(y = ..density..))
```



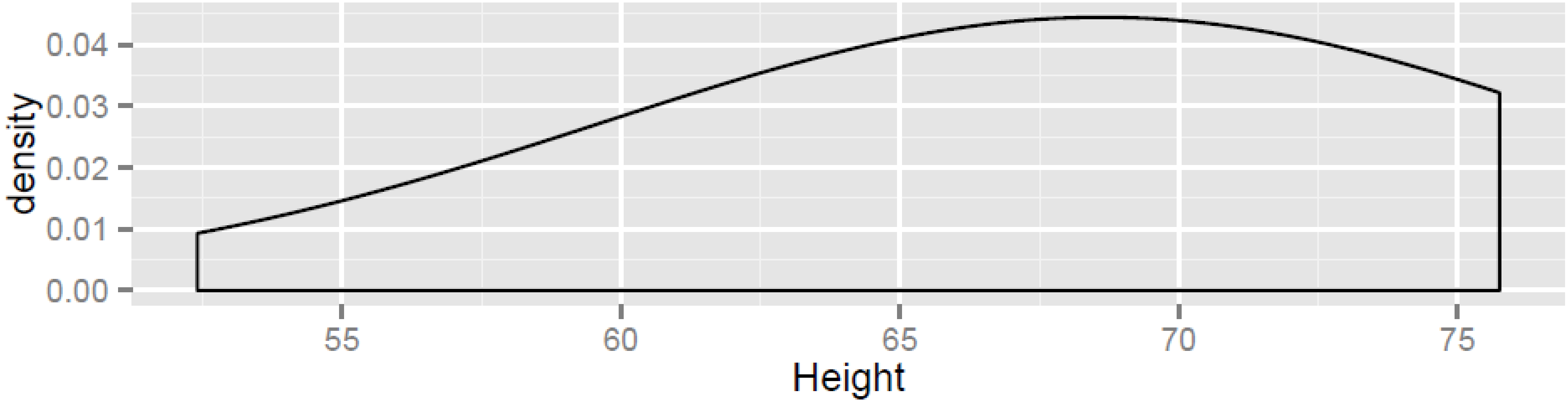
# Density Plots

Density plots are similar to histograms on a density scale, but instead of fixed bins or intervals with jumps at the boundaries, are smooth. The argument adjust to geom_density regulates how smooth the density estimate is, with larger values resulting in smoother graphs.

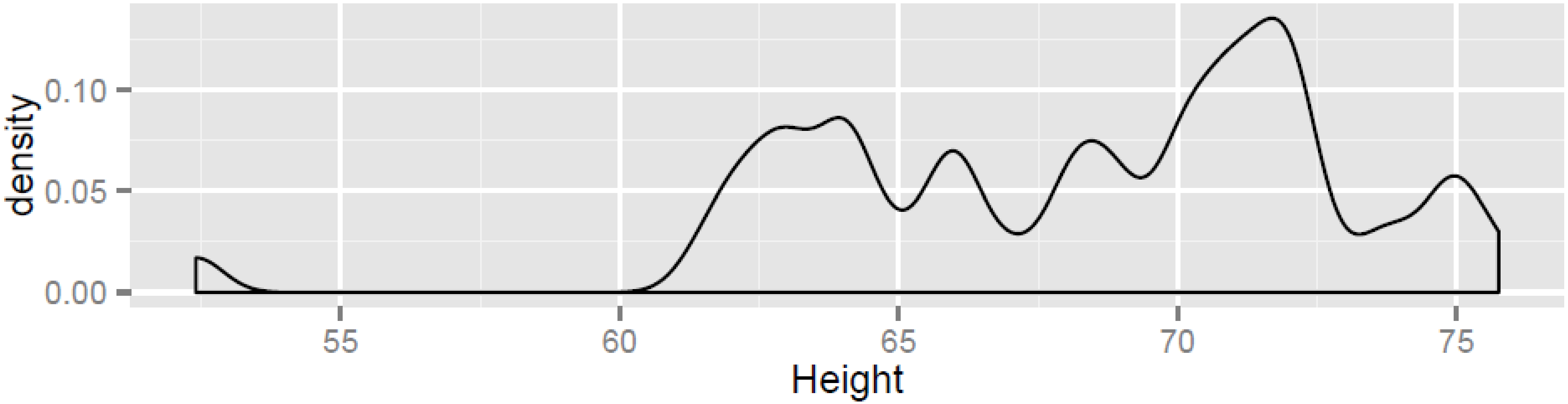
```
ggplot(students, aes(x = Height)) + geom_density()
```



```
ggplot(students, aes(x = Height)) + geom_density(adjust = 4)
```



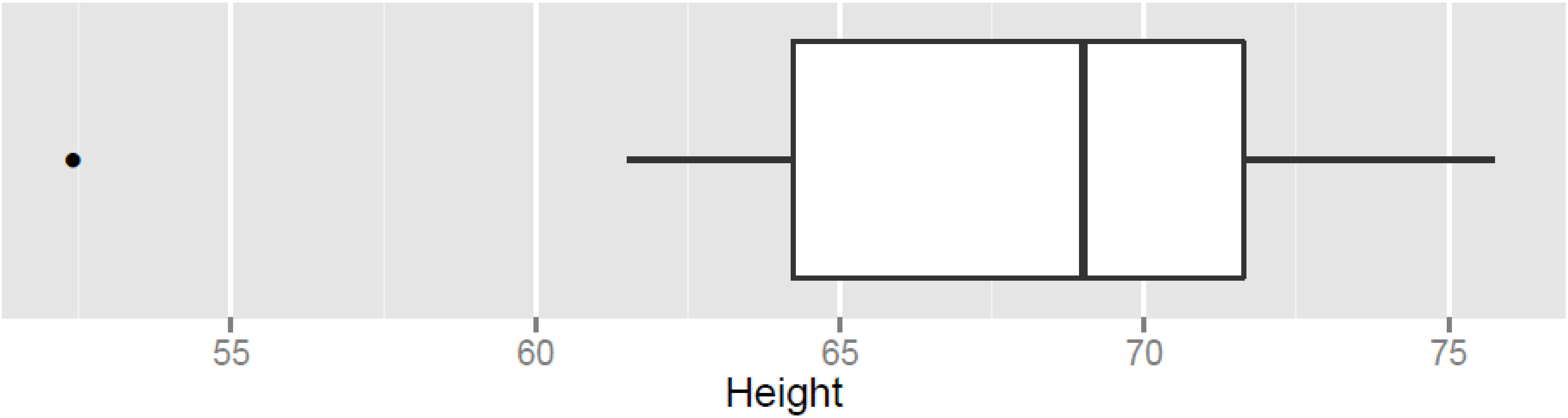
ggplot(students, aes(x = Height)) + geom_density(adjust = 0.25)



# Box and whisker Plots

Box and whisker plots are highly summarized representations of the data, essentially condensing the entire sample to a five number summary, plus locations of outliers as defined by extreme distance outside the range of the middle half of the data.

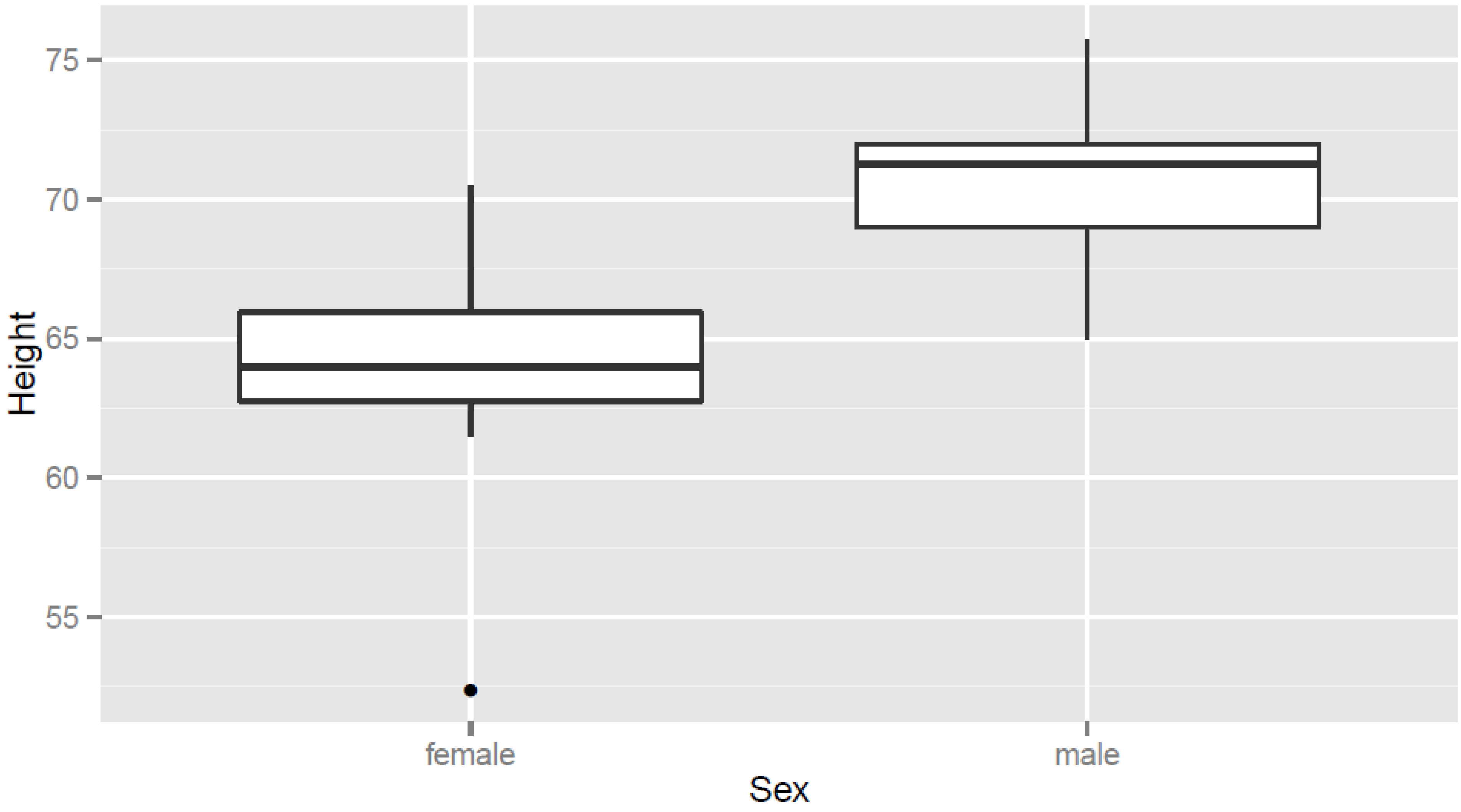
```
ggplot(students, aes(x = factor(0), y = Height)) + geom_boxplot() + coord_flip()
```



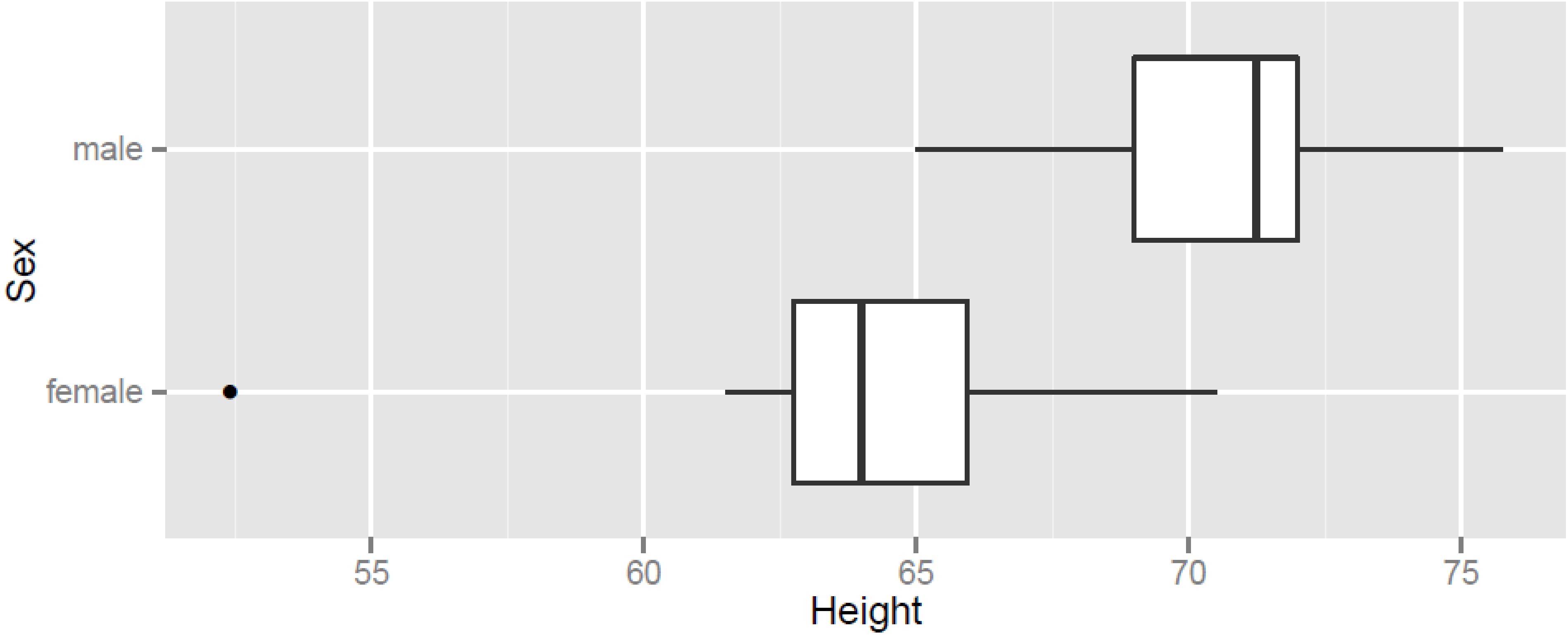
# Relationships Between a Categorical and Quantitative Variable

- Faceting
- Color
- BoxPlots

```
ggplot(students, aes(x = Sex, y = Height)) + geom_boxplot()
```



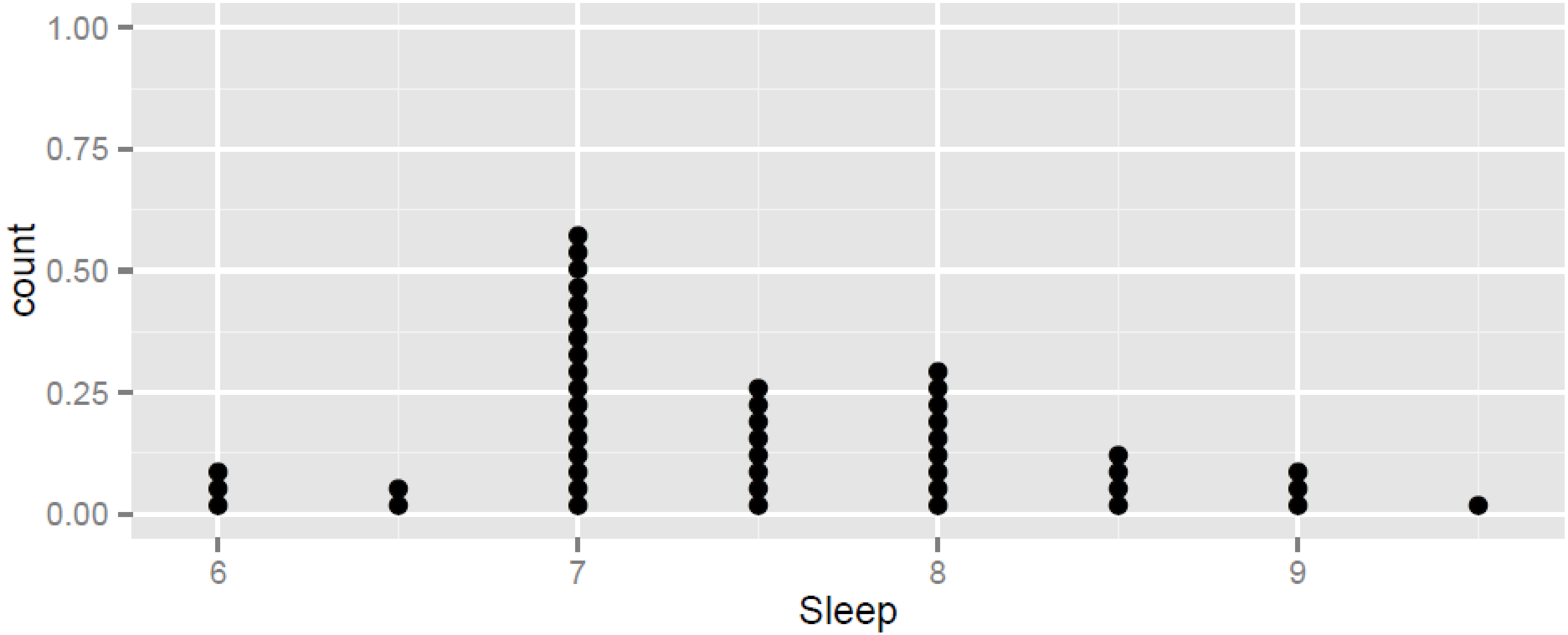
```
ggplot(students, aes(x = Sex, y = Height)) + geom_boxplot() + coord_flip()
```



# “number of hours of sleep students get each night”
# versus
# “their level in school”

ggplot(students, aes(x = Sleep)) + geom_dotplot(dotsize = 0.4)

table(students$Sleep)

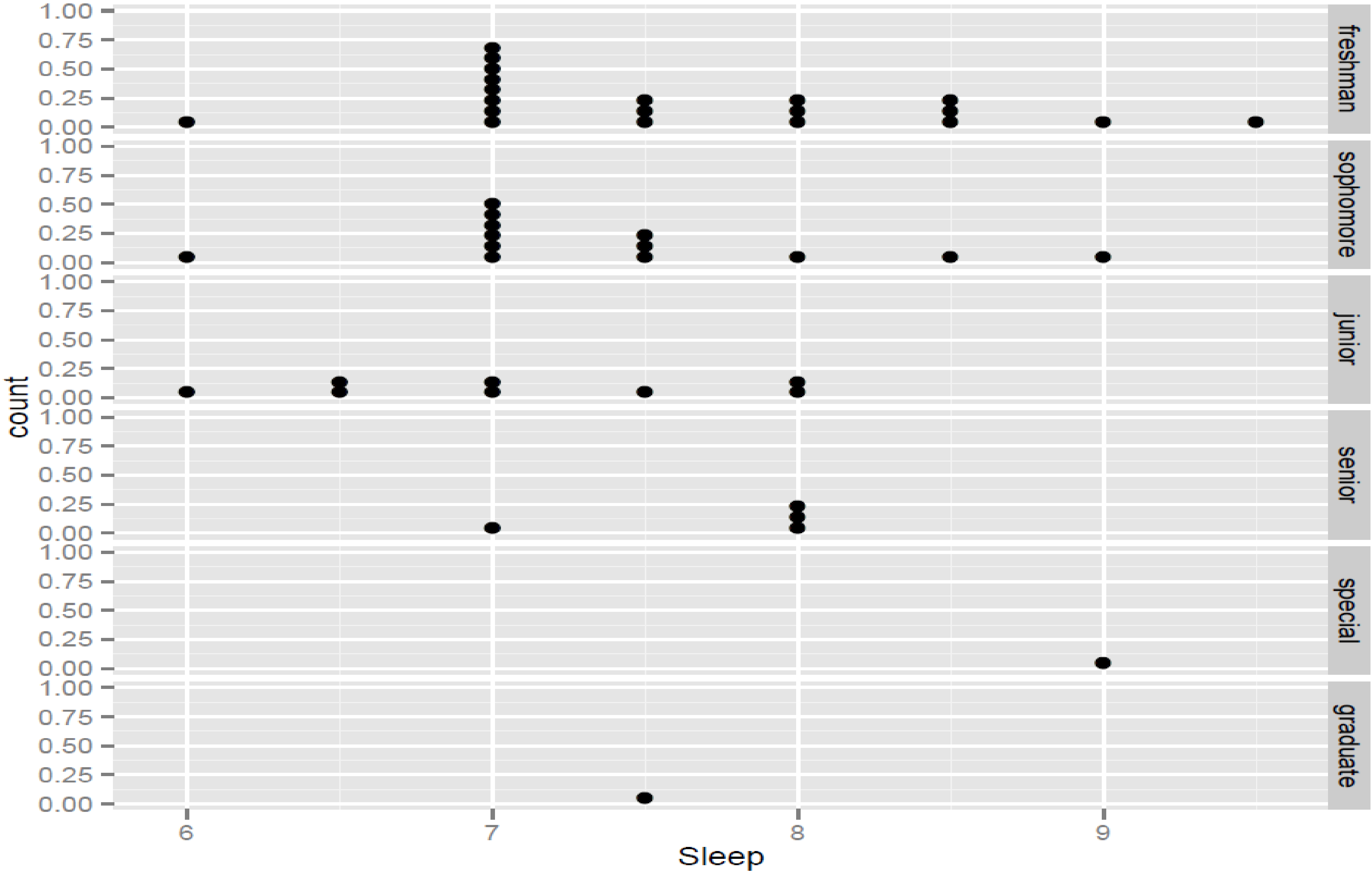


# Faceting

Small multiples displaying different subsets of the data. Useful for exploring conditional relationships.

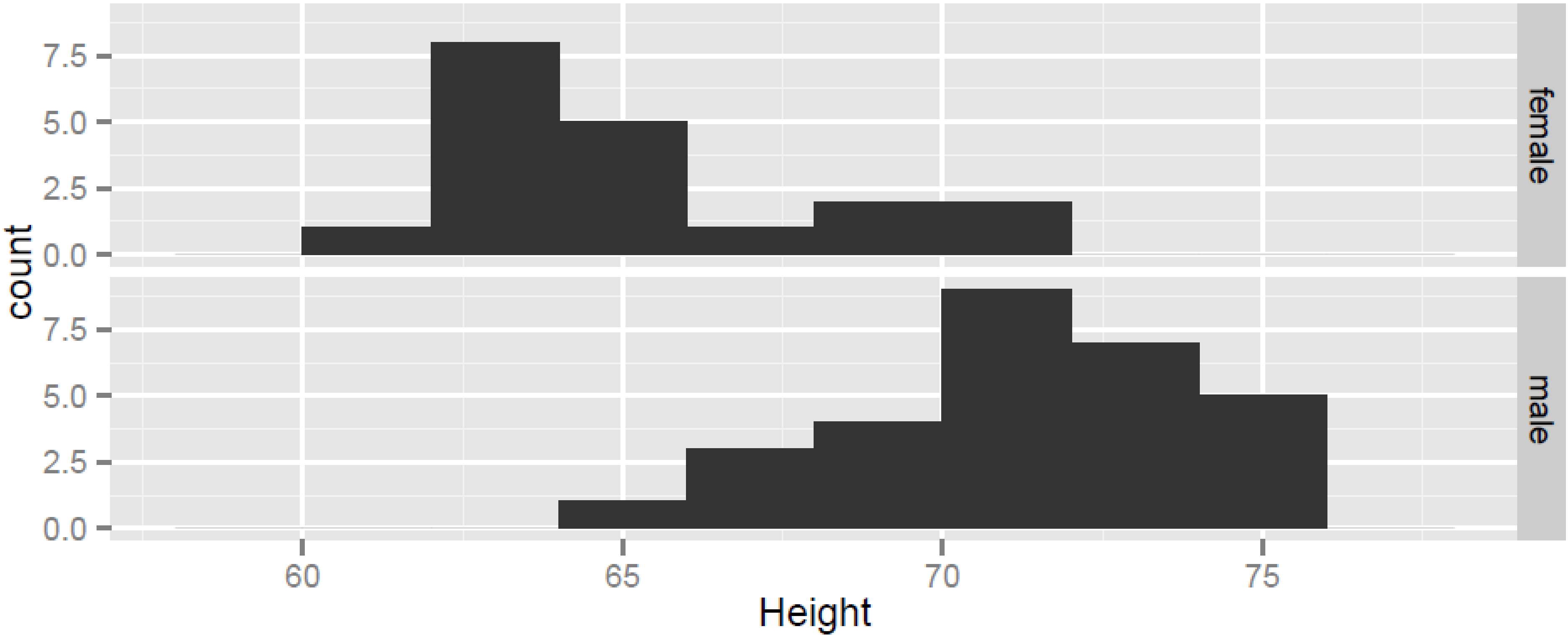
```
facet_grid(Level ~ .) – along columns
facet_grid(.~Level ) --- along rows
facet_wrap(~Level ) ---wrap panels
```

```
table(students$Sleep, students$Level)
ggplot(students, aes(x = Sleep)) + geom_dotplot(dotsize = 0.4) + facet_grid(Level ~ .)
```

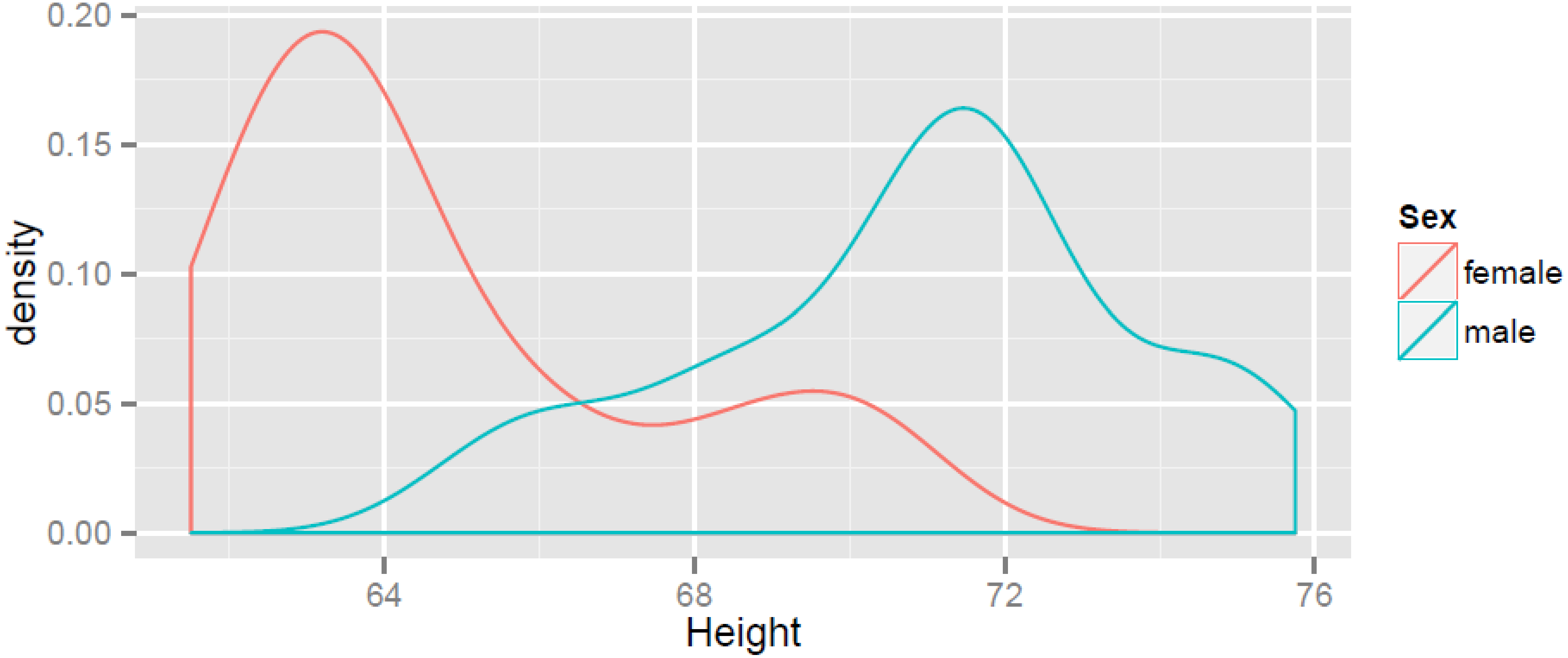


There will be a different row for each plot using facet_grid(), where the argument species which categorical variables to use to split the data by row and or column. A period means do not split in that dimension. In the example, each level of Level gets its own row, but there is only a single column of display.

```
ggplot(students, aes(x = Height)) + geom_histogram(binwidth = 2) + facet_grid(Sex ~.)
```

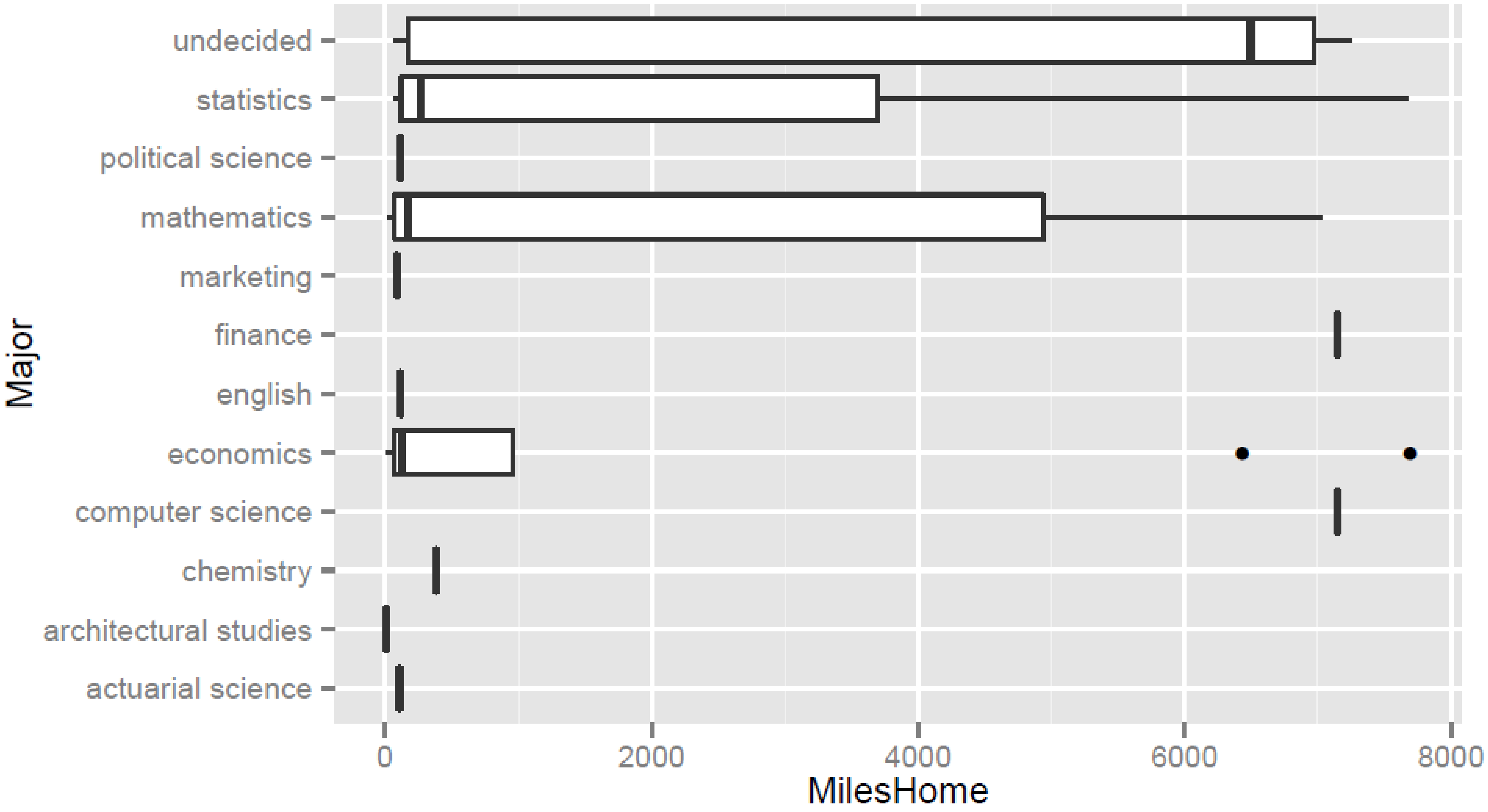


```
ggplot(students, aes(x = Height, color = Sex)) + geom_density()
```



Density plots and histograms can show many features of a distribution, but can be too messy if there are many groups. From the data on students in the course, if we wanted to examine miles from home for each different major (First major as selected by student), side-by-side boxplots will be best because there are too many groups to look at many features at the same time.

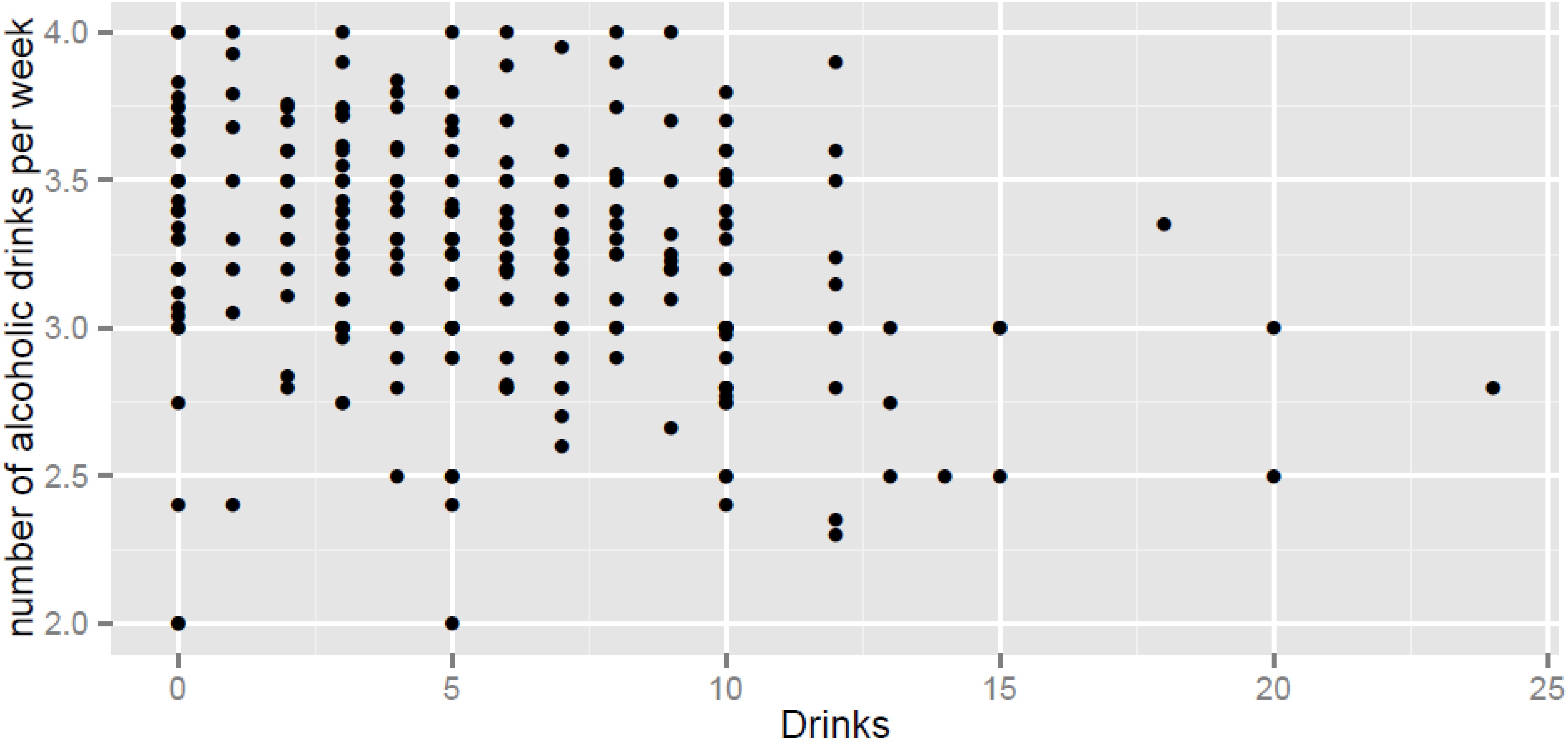
```
ggplot(students, aes(x = MilesHome)) + geom_histogram() + facet_grid(Major ~.)

ggplot(students, aes(x = MilesHome, color = Major)) + geom_density()
ggplot(students, aes(x = Major, y = MilesHome)) + geom_boxplot() + coord_flip()
```



# Two Quantitative Variables

Scatter Plots : A scatterplot is the most useful way to display two quantitative variables. More information can be added to plots by using different colors or symbols for different groups. Overlapping points can be handled by jittering (moving the positions a small random amount) or by using partially opaque points.

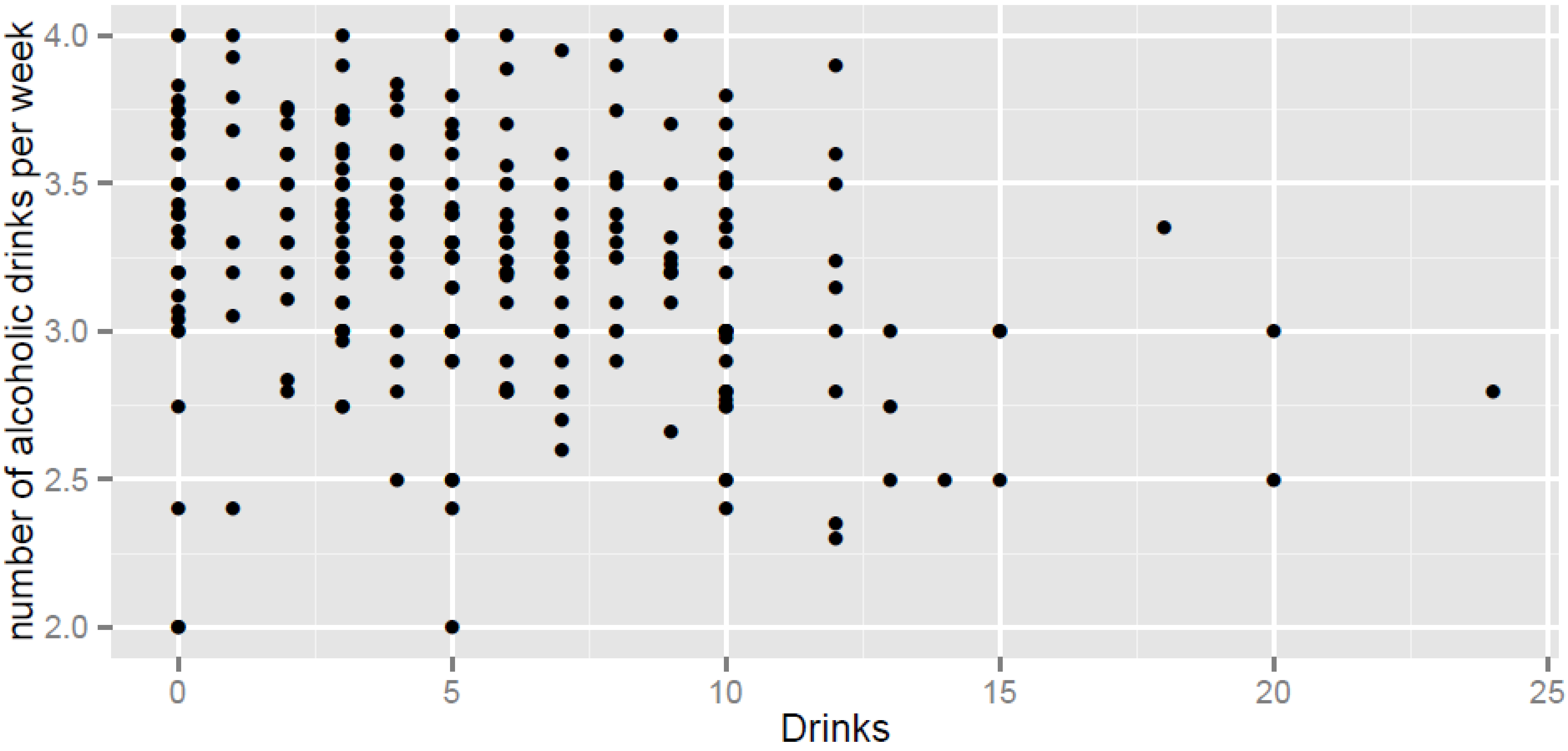
```
ggplot(SleepStudy, aes(x=Drinks,y=GPA)) + geom_point() + xlab('number of alcoholic drinks per week')
```



```
with(SleepStudy, cor(Drinks, GPA))
```

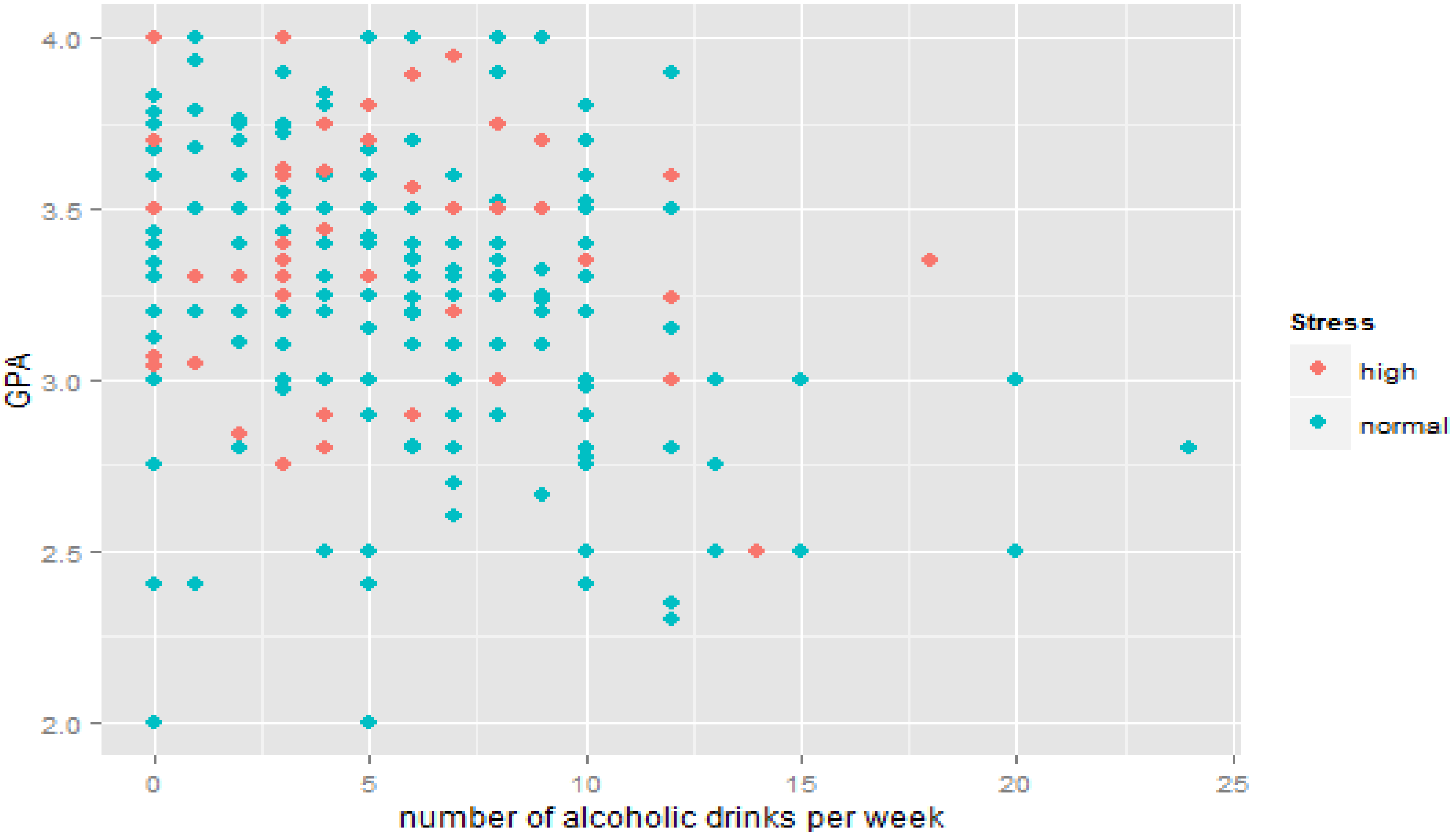
## Increase the size of points

```
ggplot(SleepStudy, aes(x=Drinks,y=GPA)) + geom_point() + xlab('number of alcoholic drinks per week')
```



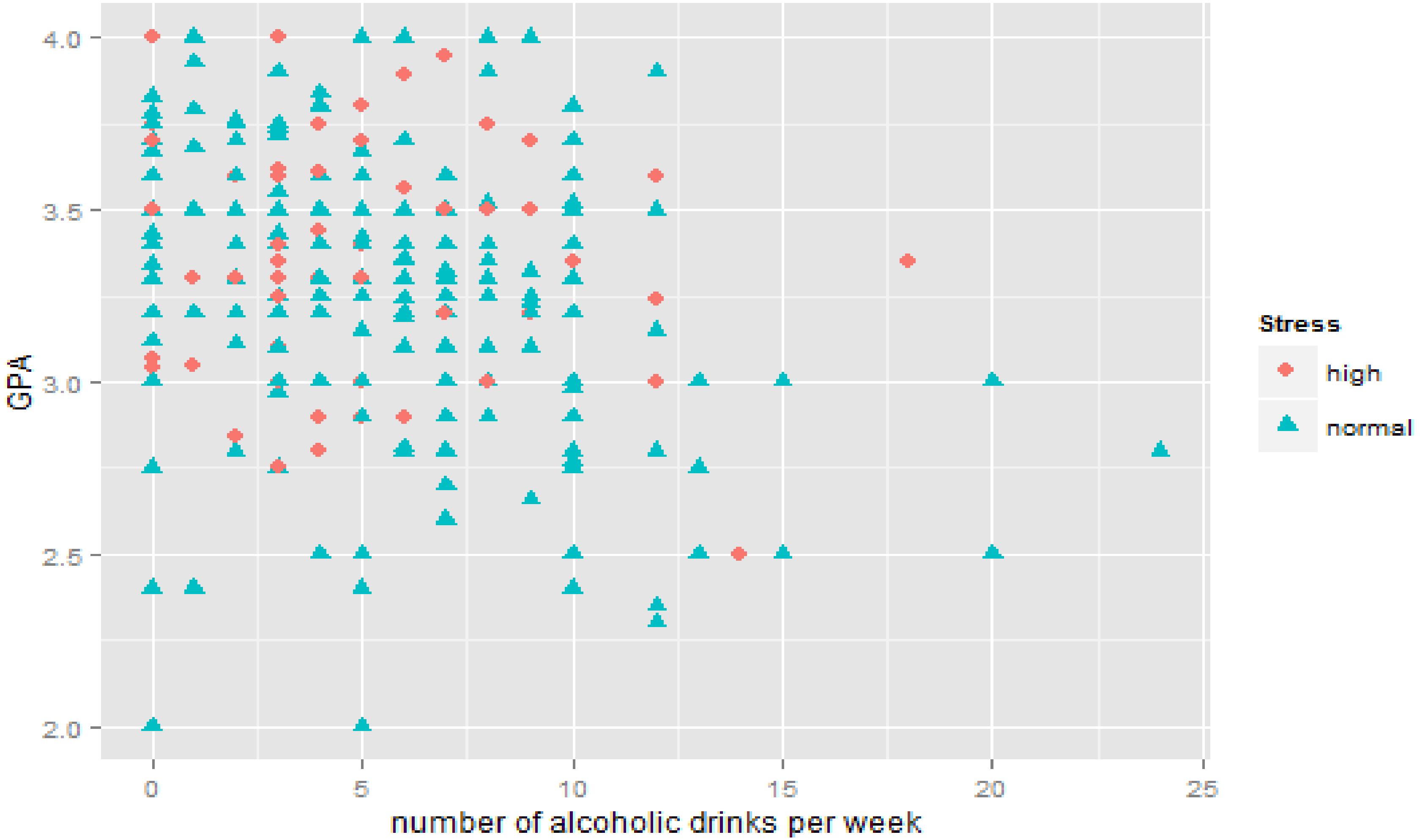
# Add some color

```
ggplot(SleepStudy, aes(x=Drinks,y=GPA,color=Stress)) + geom_point(size=3) + xlab('number of alcoholic drinks per week')
```



# Differentiate points by shape

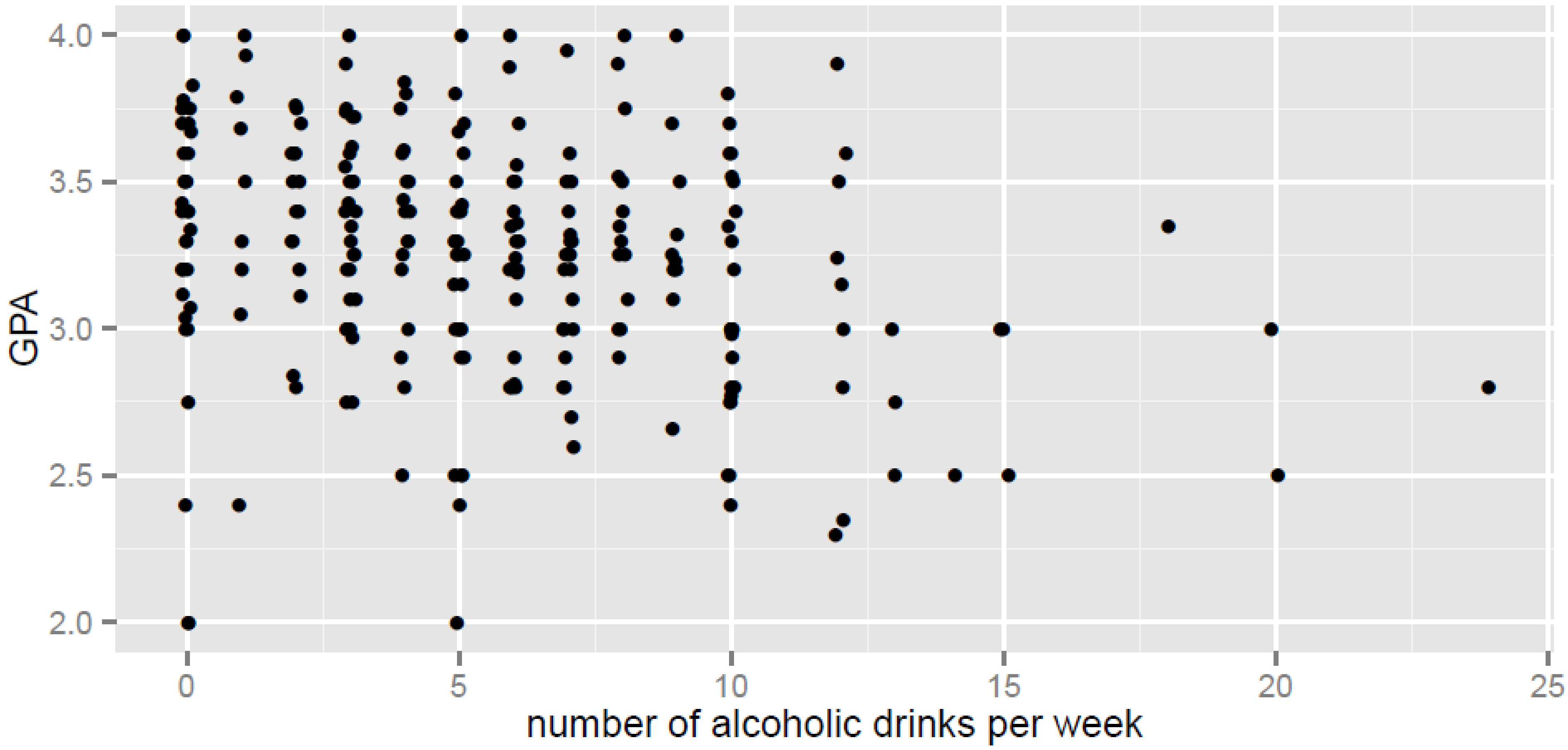
ggplot(SleepStudy, aes(x=Drinks,y=GPA,color=Stress)) + geom_point(aes(shape=Stress),size=3) + xlab('number of alcoholic drinks per week')



The previous graph also has a fair amount of overplotting because drinks is a small integer and some students have equal GPAs.

The following command will jitter all of the points in a controlled way. We choose to jitter only the width by a small amount and not the height so the GPA remains accurate and we can still be clear the true number of drinks.

```
ggplot(SleepStudy, aes(x=Drinks,y=GPA)) + geom_point(position=position_jitter(w=0.1,h=0)) +
xlab('number of alcoholic drinks per week')
```



# More Examples

```
ggplot(pressure, aes(x = temperature, y = pressure)) + geom_line() +  geom_point()
```

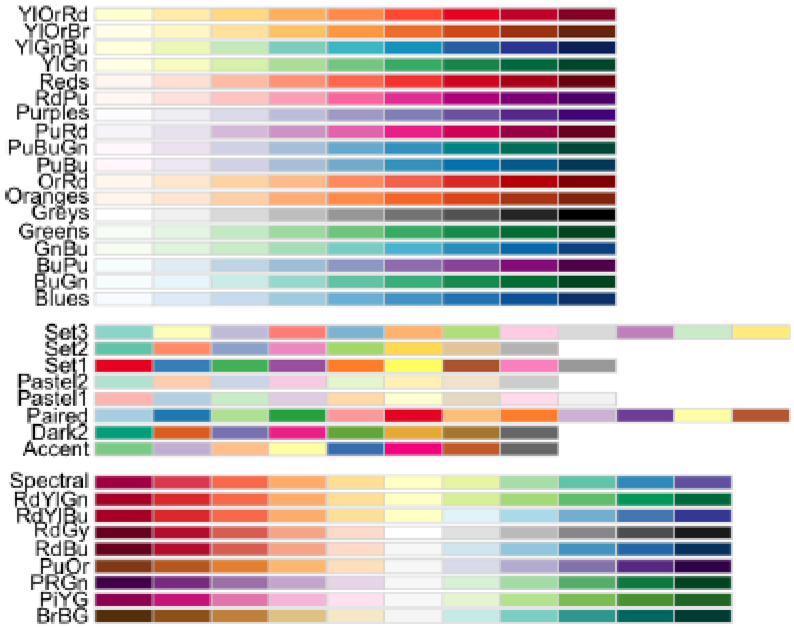
```
with(pressure, cor(temperature, pressure))
```

# Colors

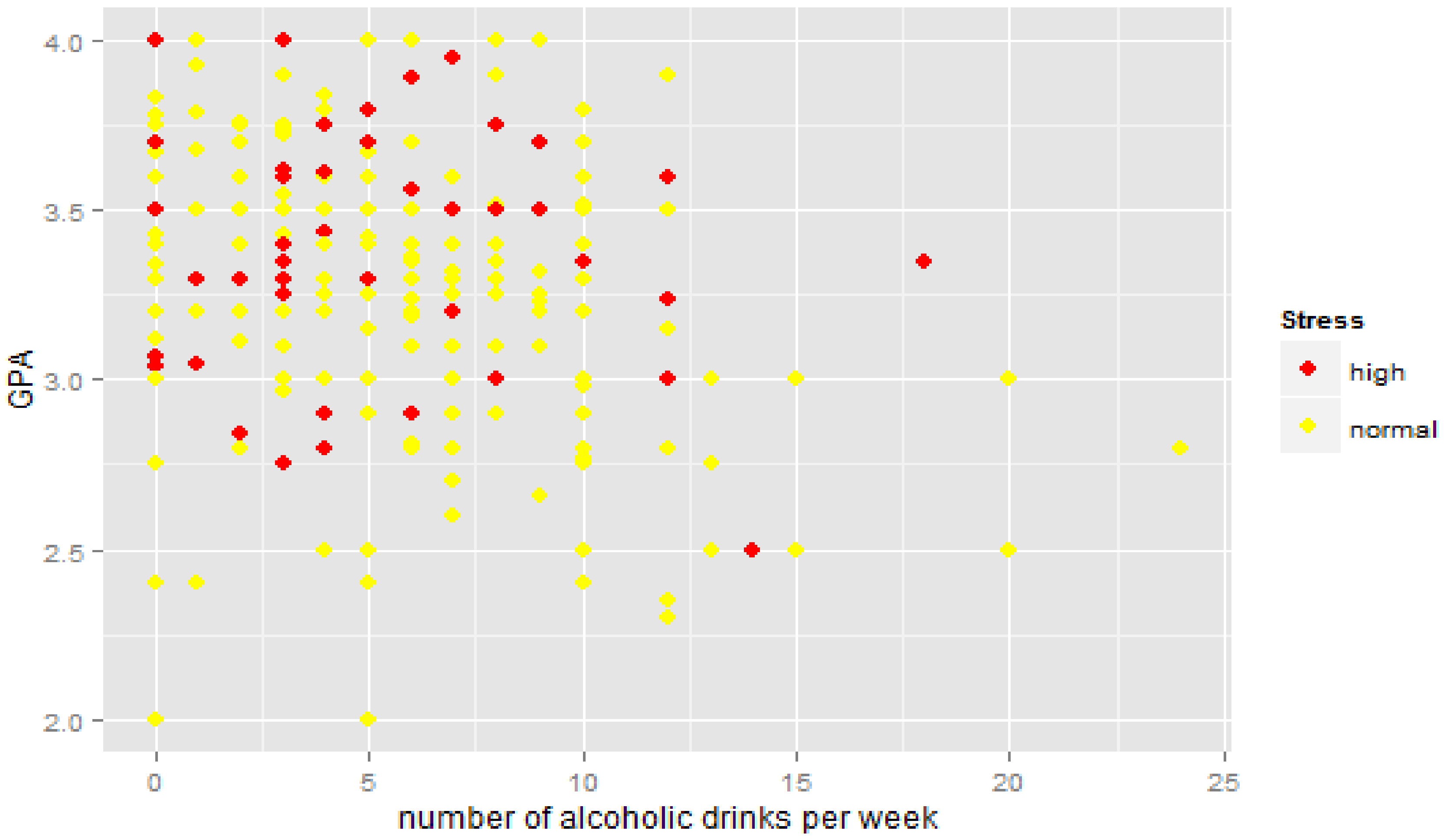
```
# Map all points to one color
aes(color = "black")
# Or map the points to a variavble
aes(color = variable)
# Then add a scale for the colors.  Below we manually
# define colors but there are other ways (see next slide)
scale_fill_manual(values = c("color1", "color2"))
```

# The RColorBrewer package

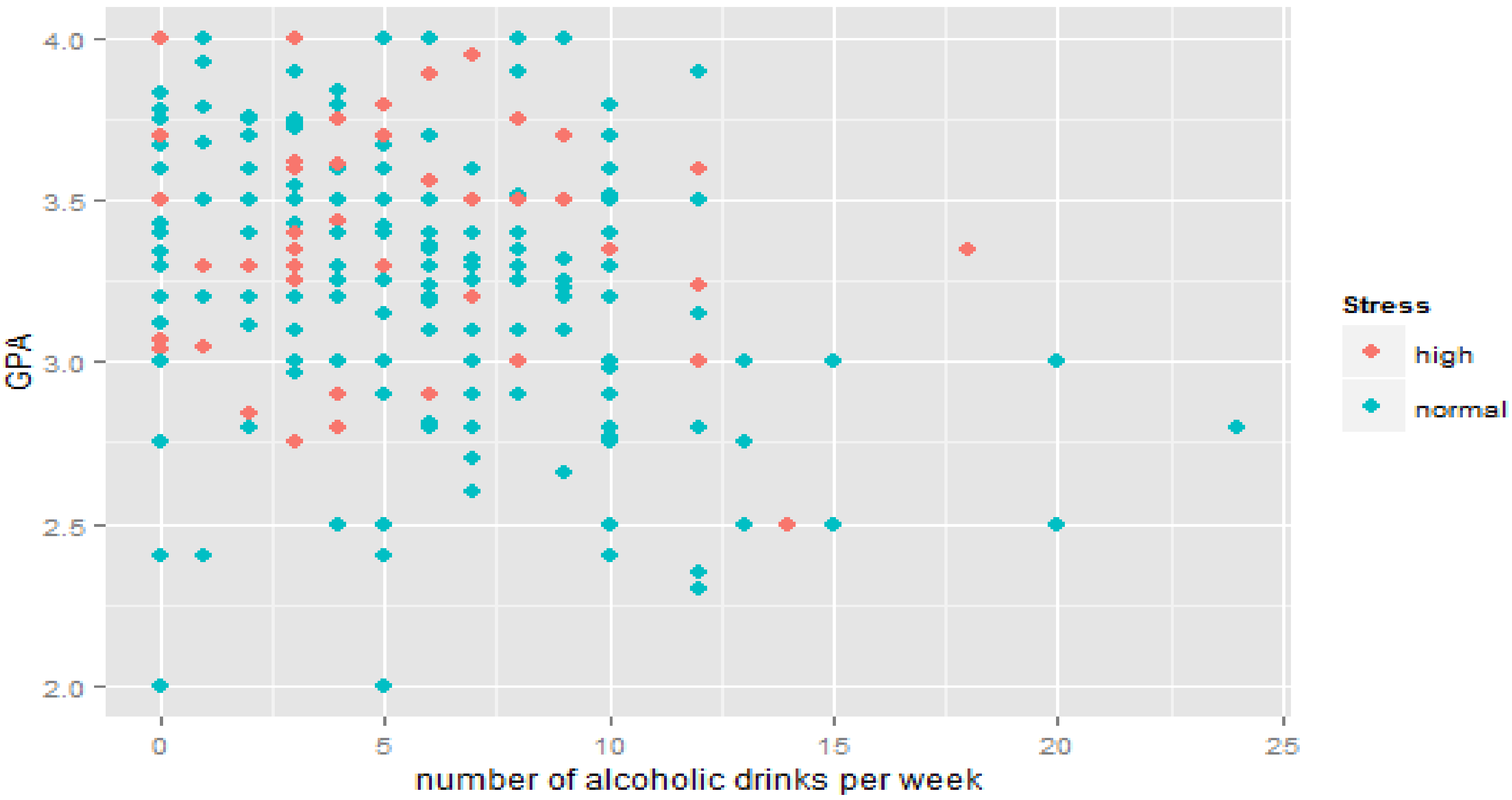
```r
library(RColorBrewer)
display.brewer.all()
```



```
ggplot(SleepStudy, aes(x=Drinks,y=GPA,color=Stress)) + geom_point(size=3) +
xlab('number of alcoholic drinks per week') +
scale_color_manual(values=c("red","yellow"))
```



```
ggplot(SleepStudy, aes(x=Drinks,y=GPA,color=Stress)) + geom_point(size=3) +
xlab('number of alcoholic drinks per week') +
scale_fill_brewer(palette="Set1")
```


GPA
number of alcoholic drinks per week
Stress
high
normal