Simulated Data Example 02 - Blood Pressure

Bob Horton and the HS616 Class of 2015

March 13, 2015

This is a function to generate a simulated data set relating blood pressure to various drivers.

```
generate_dataset <- function(N=100){
    HEIGHT_MEAN <- c( F = 1.6, M = 1.8 )
    HEIGHT_SD <- 0.15
    WEIGHT_MEAN <- c( F = 54, M = 70 )
    WEIGHT_SD <- 20

    bmi <- function(height, weight) weight/height^2

    sbp <- function(sex, salt, bmi, etoh){
        ifelse (sex == "M",
            90 + 0.005 * salt + 1.0 * bmi - 0.01 * etoh,
            80 + 0.005 * salt + 1.5 * bmi - 0.01 * etoh)
    }

    sex <- sample(c("M", "F"), N, replace=TRUE)
    salt <- rnorm(N, mean=2200, sd=50)
    height <- rnorm(N, mean=HEIGHT_MEAN[sex], sd=HEIGHT_SD)
    weight <-  1.2 * ( height - HEIGHT_MEAN[sex] ) + WEIGHT_MEAN[sex] + rnorm(N, sd=WEIGHT_SD)
    etoh <- 50 * rpois(N, lambda=6)

    systolic <- sbp(sex, salt, bmi(height, weight), etoh) + rnorm(N, sd=5)

    # add some distractors
    car_makes <- unique(sapply( strsplit(row.names(mtcars), " "), "[", 1))
    car <- sample( car_makes, N, replace=TRUE)

    zodiac <- c("Aries", "Taurus", "Gemini", "Cancer", "Leo", "Virgo",
     "Libra", "Scorpio", "Sagittarius", "Capricorn", "Aquarius", "Pisces")
    sign <- sample(zodiac, N, replace=TRUE)

    data.frame( sex, salt, height, weight, etoh, car, sign, systolic)
}
```

Calling this function will create a simulated data set where each row represents a patient and columns represent attributes of the patient. The simulation produces a data set with specifc patterns of relationships. An analyst should be able to deduce these relationships by statistical modeling; since they have been simulated, we can check whether the analyst deduces the actual relationships in the data.

Here we call the function and store the resulting data frame for analysis. The parameter specifies the number of patients, so we can make any size sample we need.

```
set.seed(123)
bpdata <- generate_dataset(150)
knitr::kable(head(bpdata))
```

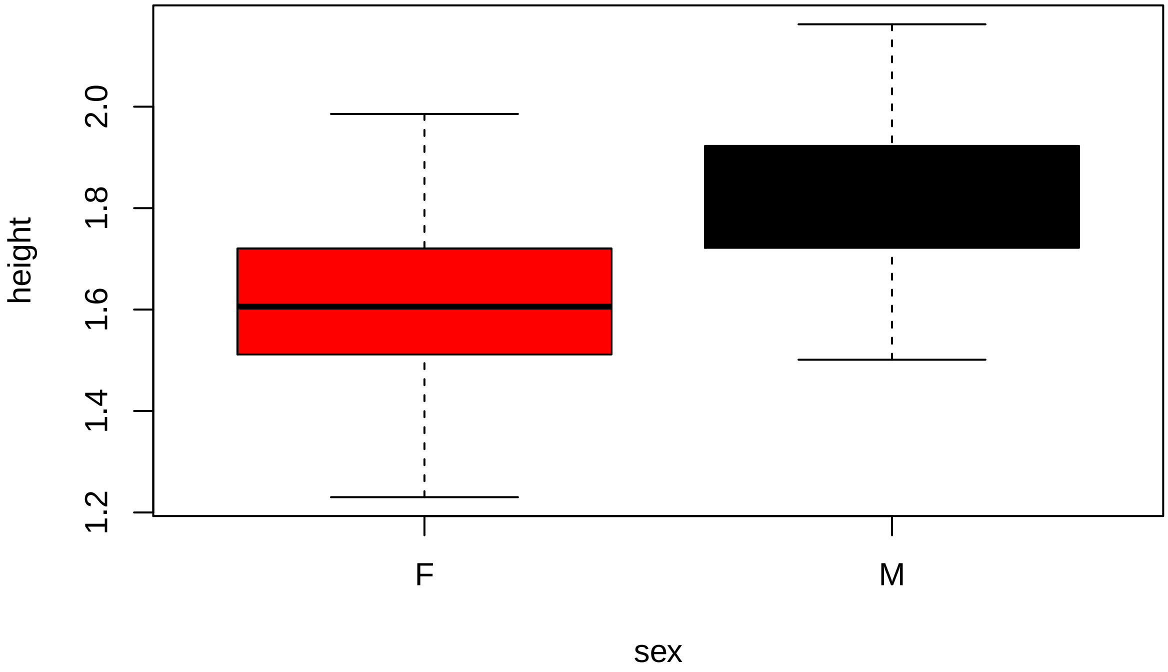
sex	salt	height	weight	etoh	car	sign	systolic
M	2251.279	1.808963	58.22113	350	Maserati	Cancer	120.9183
F	2185.761	1.494311	33.93756	400	Volvo	Sagittarius	109.5897
M	2138.964	1.692417	72.76041	450	Honda	Leo	121.4309
F	2209.065	1.732698	53.87309	300	Merc	Leo	107.3814
F	2193.055	1.447661	18.01157	300	Pontiac	Aries	104.8089
M	2200.288	2.093294	71.04297	200	Dodge	Libra	114.1607

The outcome variable "systolic" is the patient's blood pressure. The analytical challenge is to find the drivers among the other variables, and to describe their relationships to the outcome and to the other drivers.

Exploratory Visualization

We examined the outcome repeatedly when designing the simulation and adjusted the coefficients and other parameters so that the outcomes fit the patterns we wanted to create. First we want to be sure that the drivers are in reasonable ranges, and have the appropriate collinear relationships. In this simulation, weight is related to height.

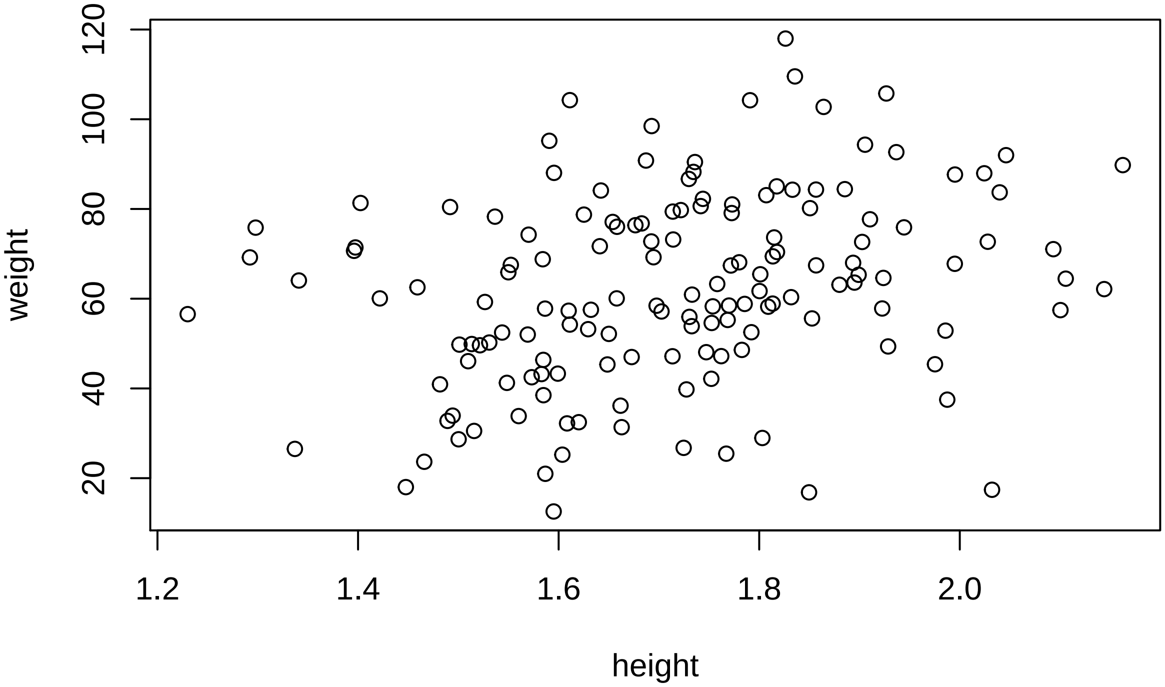
```
with(bpdata, plot( height ~ sex, col=sex))
```



```
plot( weight ~ height, border=sex, data=bpdata )
```

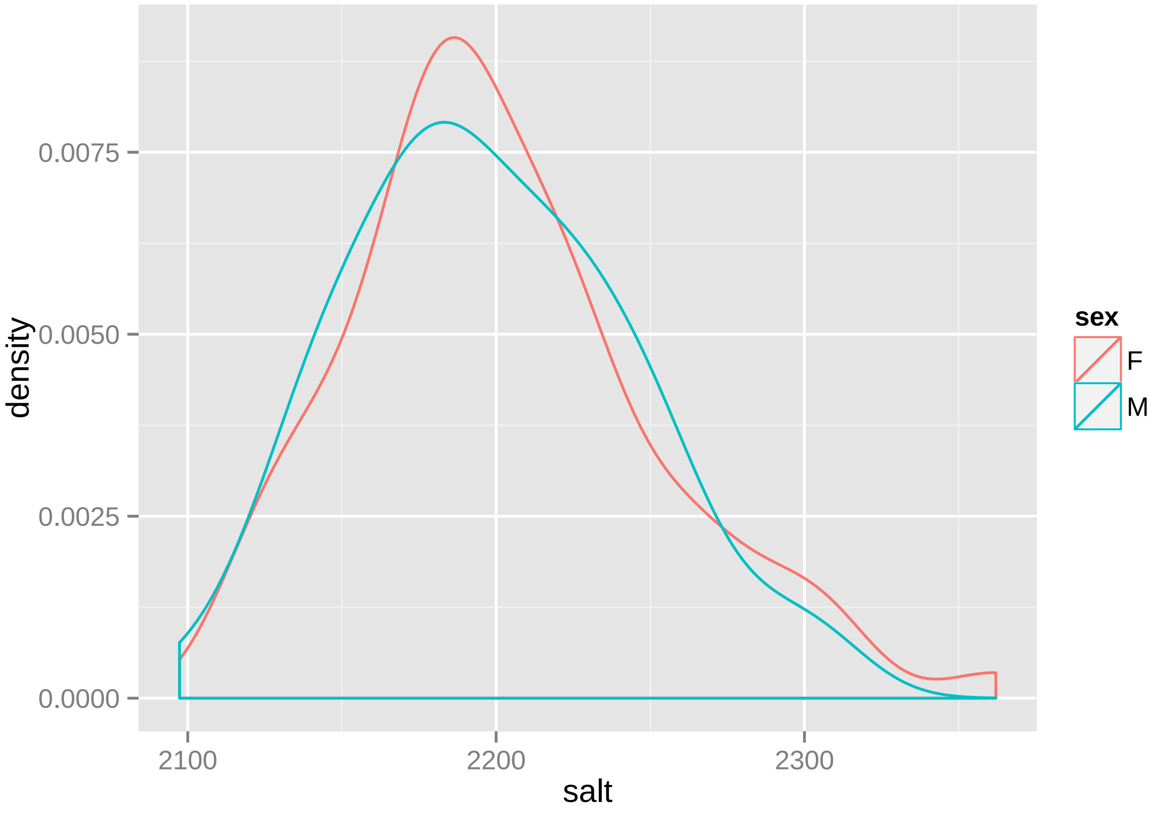
```
## Warning in plot.window(...): "border" is not a graphical parameter
```

```
## Warning in plot.xy(xy, type, ...): "border" is not a graphical parameter
## Warning in axis(side = side, at = at, labels = labels, ...): "border" is
## not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "border" is
## not a graphical parameter

## Warning in box(...): "border" is not a graphical parameter

## Warning in title(...): "border" is not a graphical parameter
```

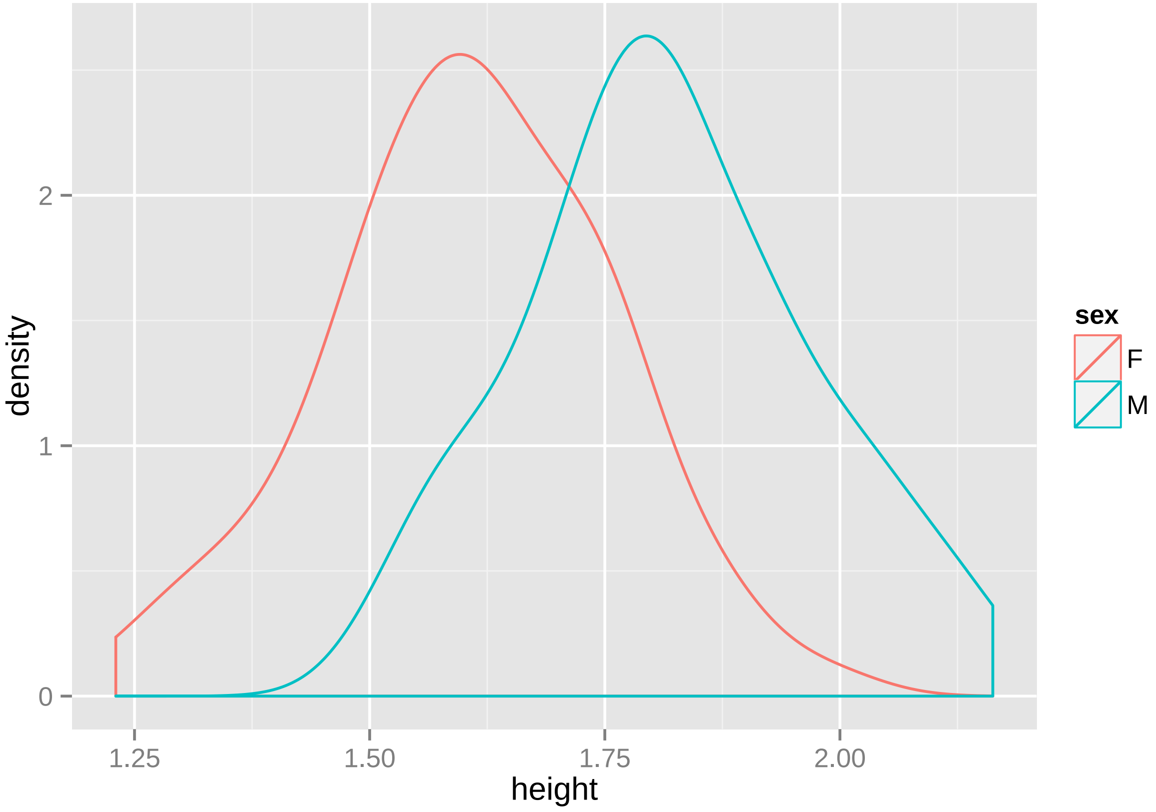


The ggplot2 package makes it easy to plot densities, so we can get a quick overview of how the sample attributes are distributed:

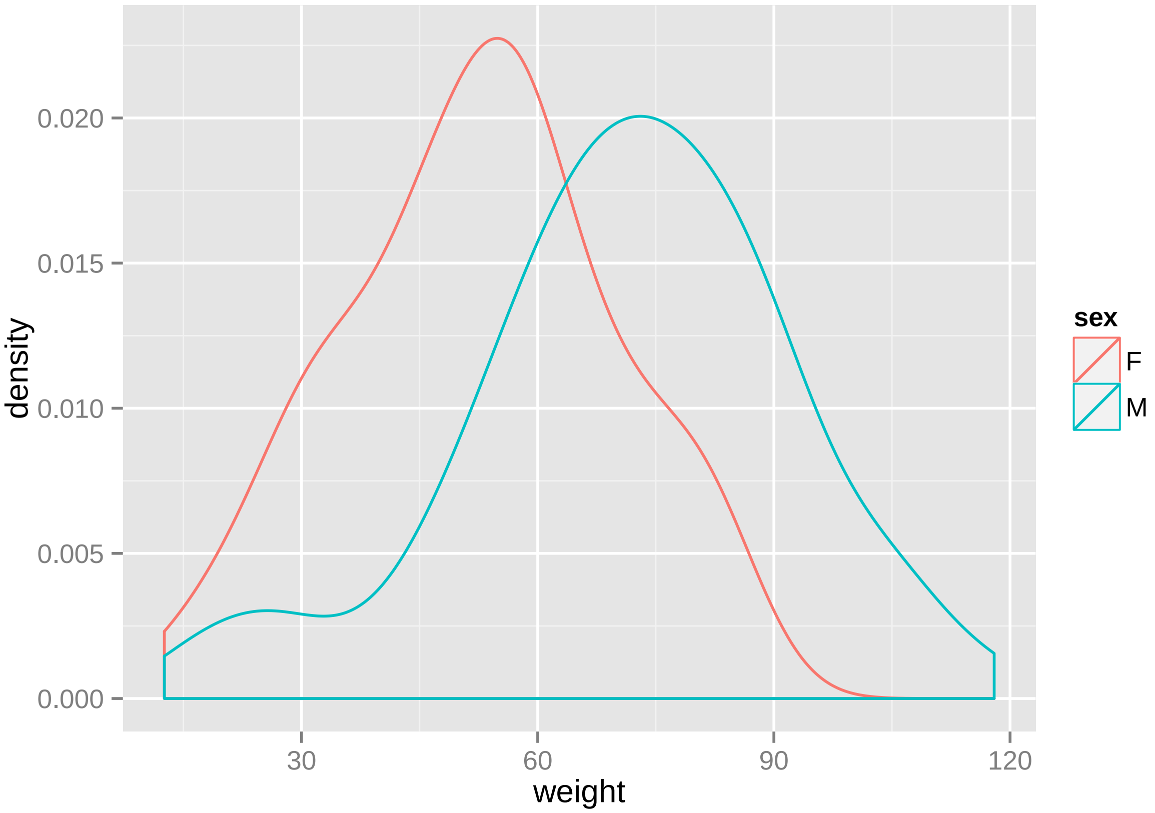
```
library(ggplot2)
ggplot(bpdata, aes(x=salt, col=sex)) + geom_density()
```



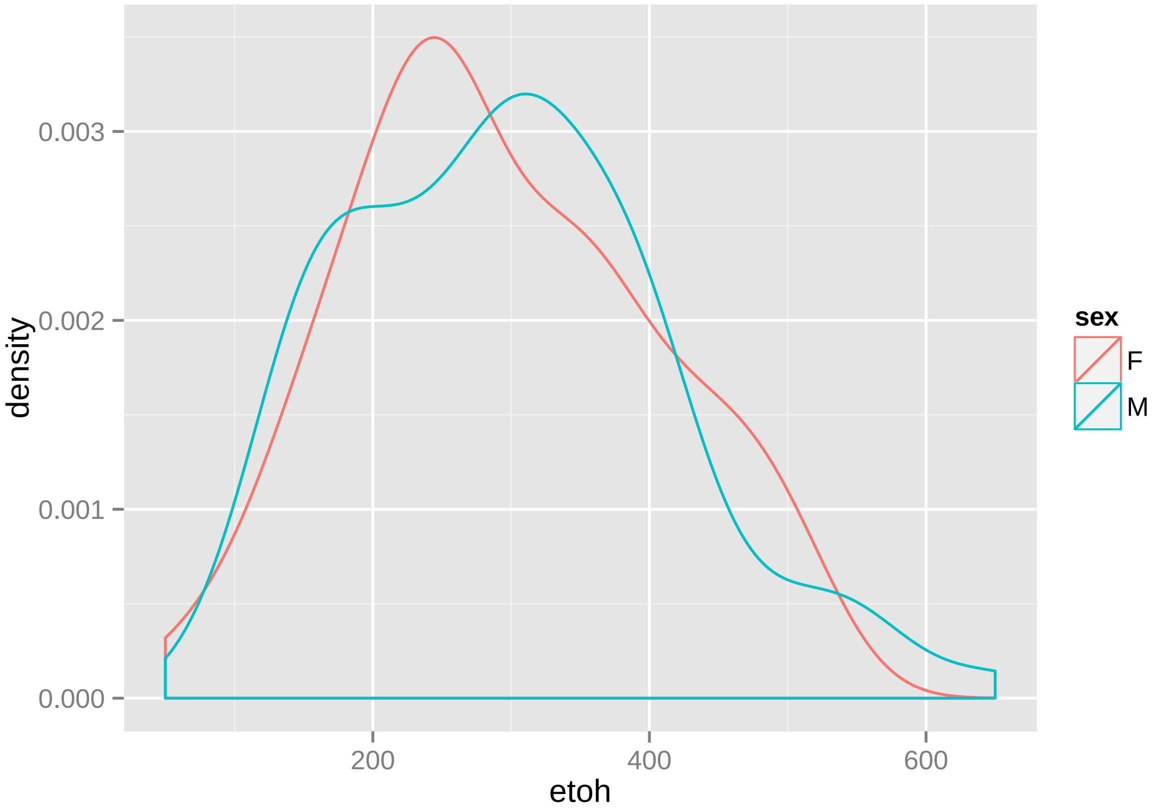
```
ggplot(bpdata, aes(x=height, col=sex)) + geom_density()
```



```r
ggplot(bpdata, aes(x=weight, col=sex)) + geom_density()
```

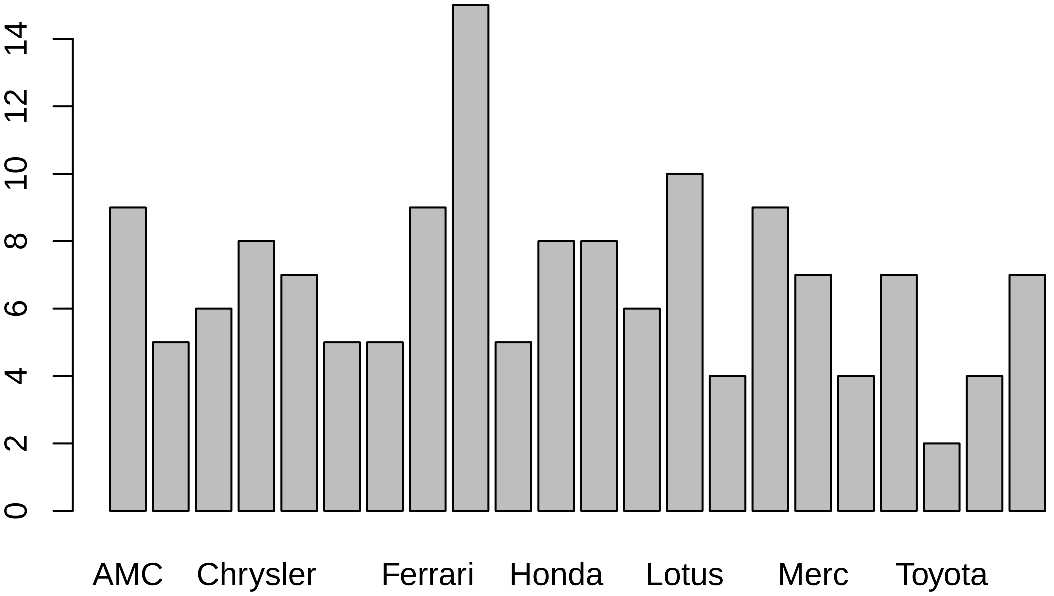


```r
ggplot(bpdata, aes(x=etoh, col=sex)) + geom_density()
```

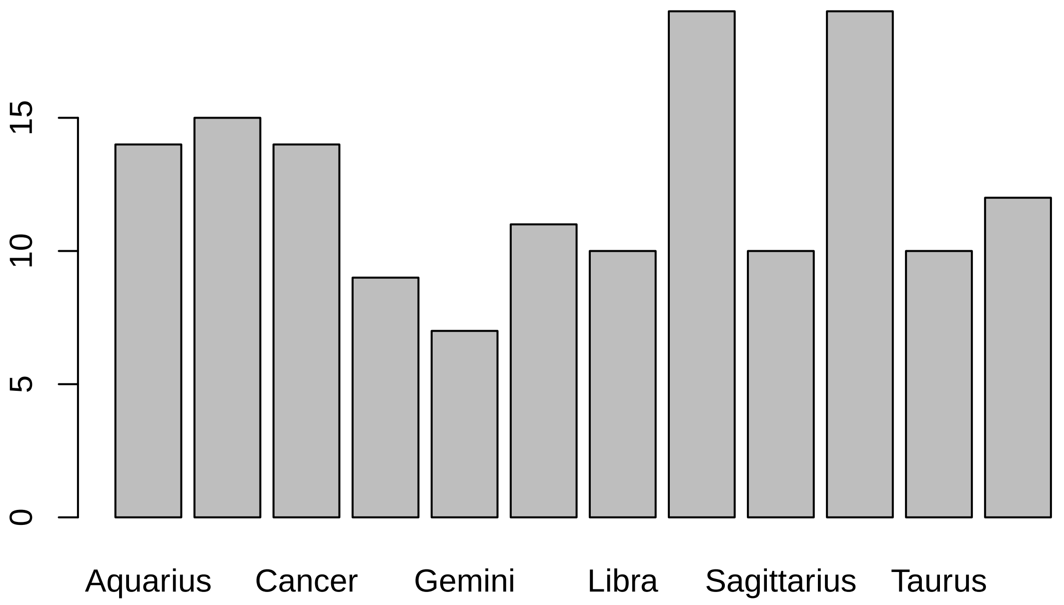


Categorical inputs

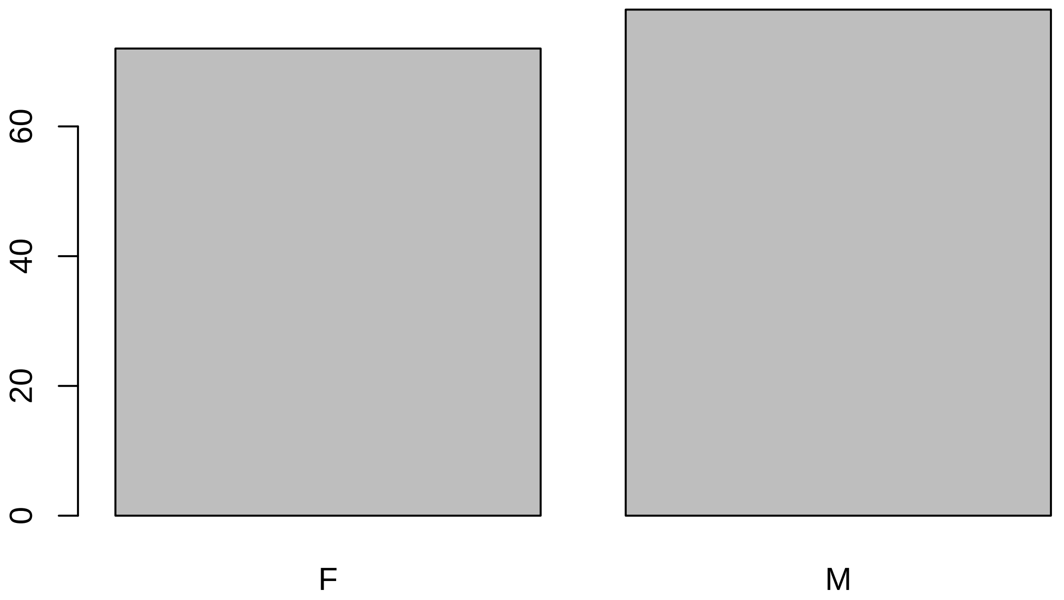
```
plot( ~ car, data=bpdata)
```



```
plot( ~ sign, data=bpdata)
```

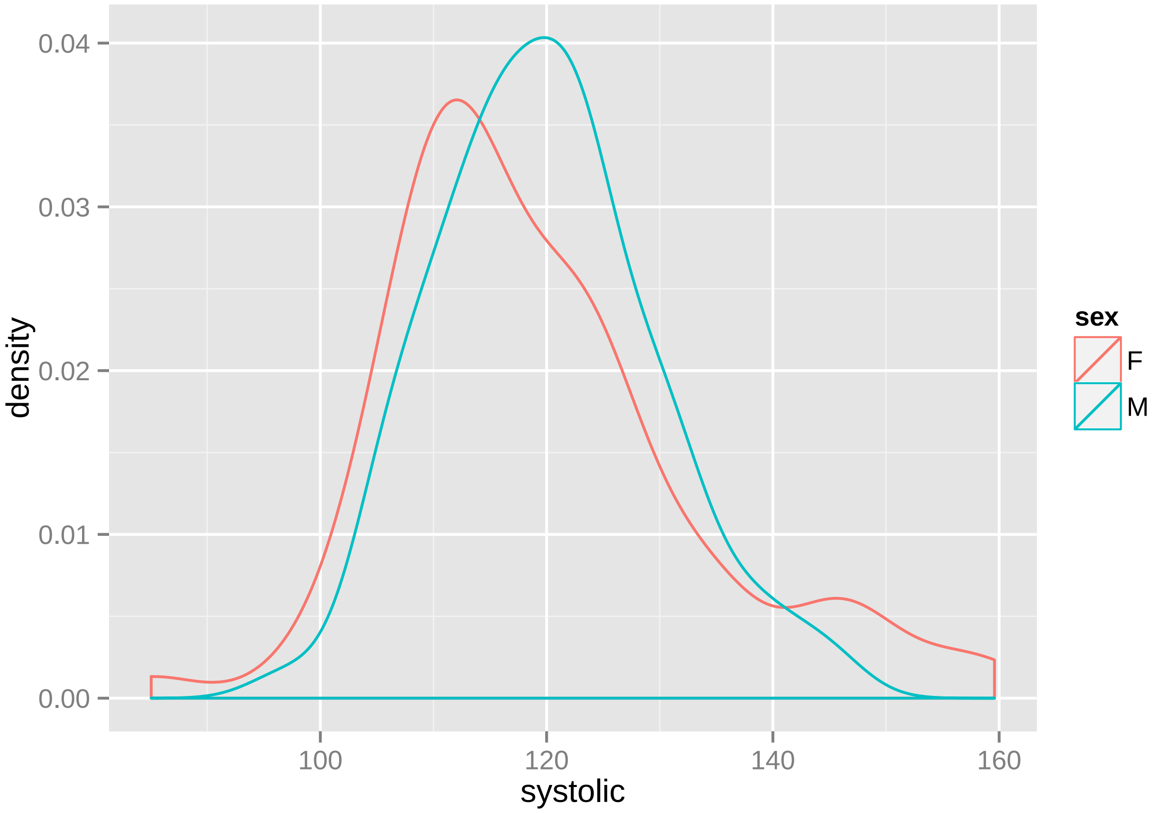


```
plot( ~ sex, data=bpdata)
```



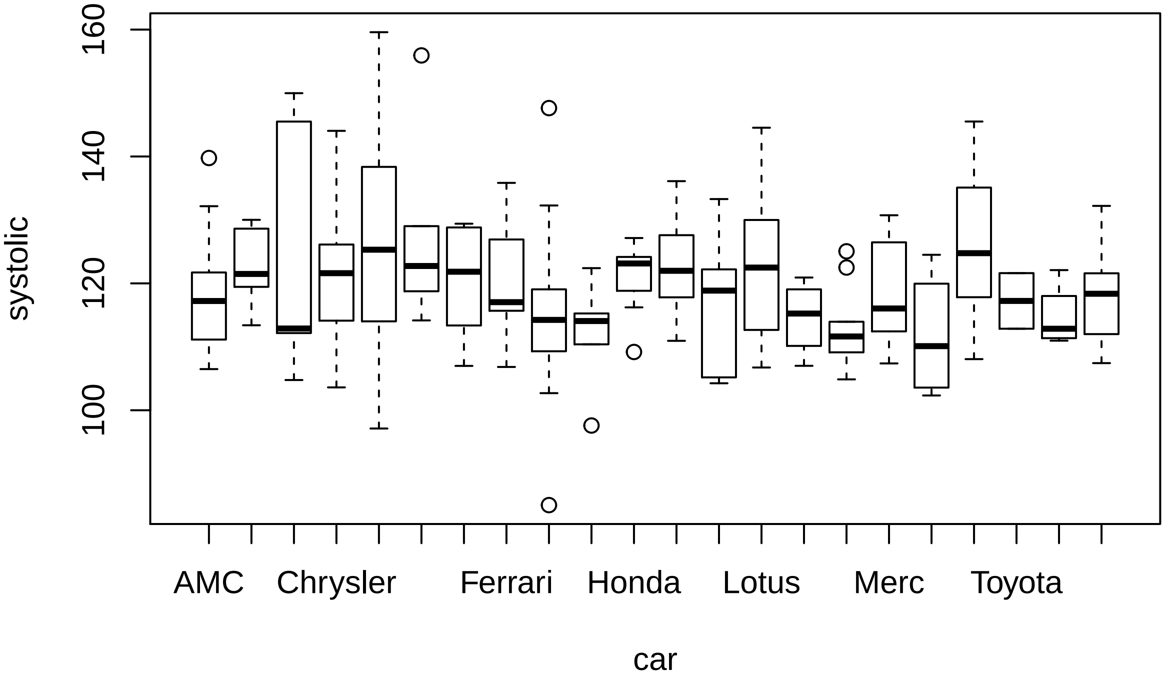
We can also examine the outcome distribution overall, and conditioned on various inputs:

```
ggplot(bpdata, aes(x=systolic, col=sex)) + geom_density()
```

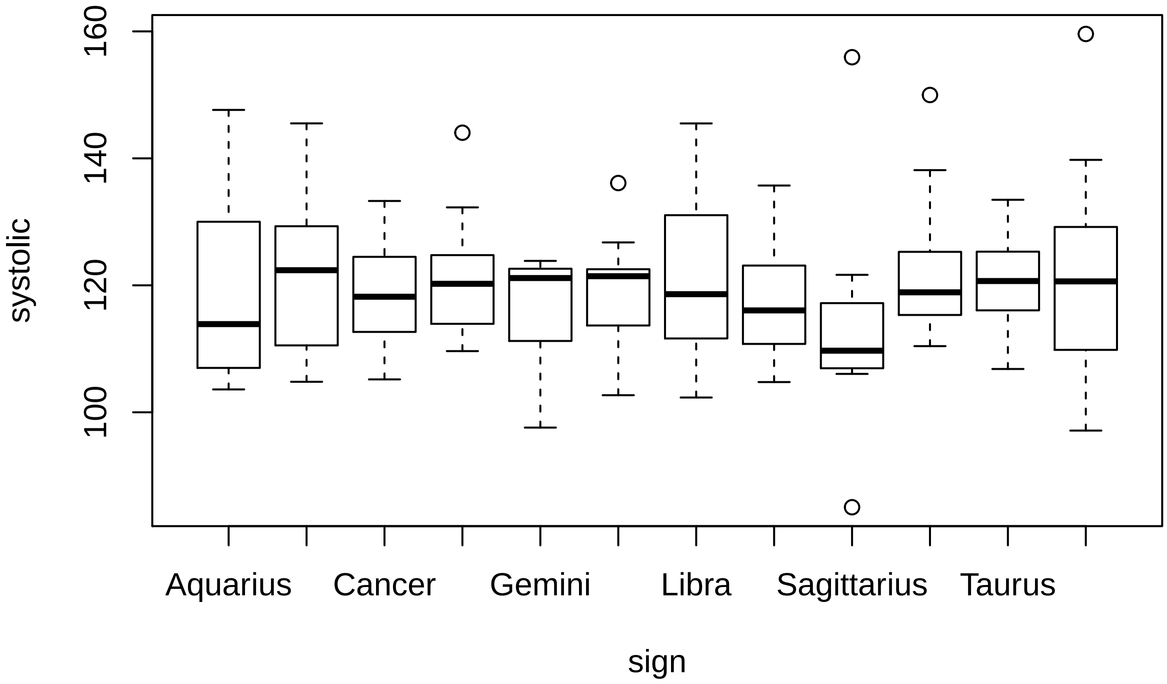


Outcome conditioned on inputs

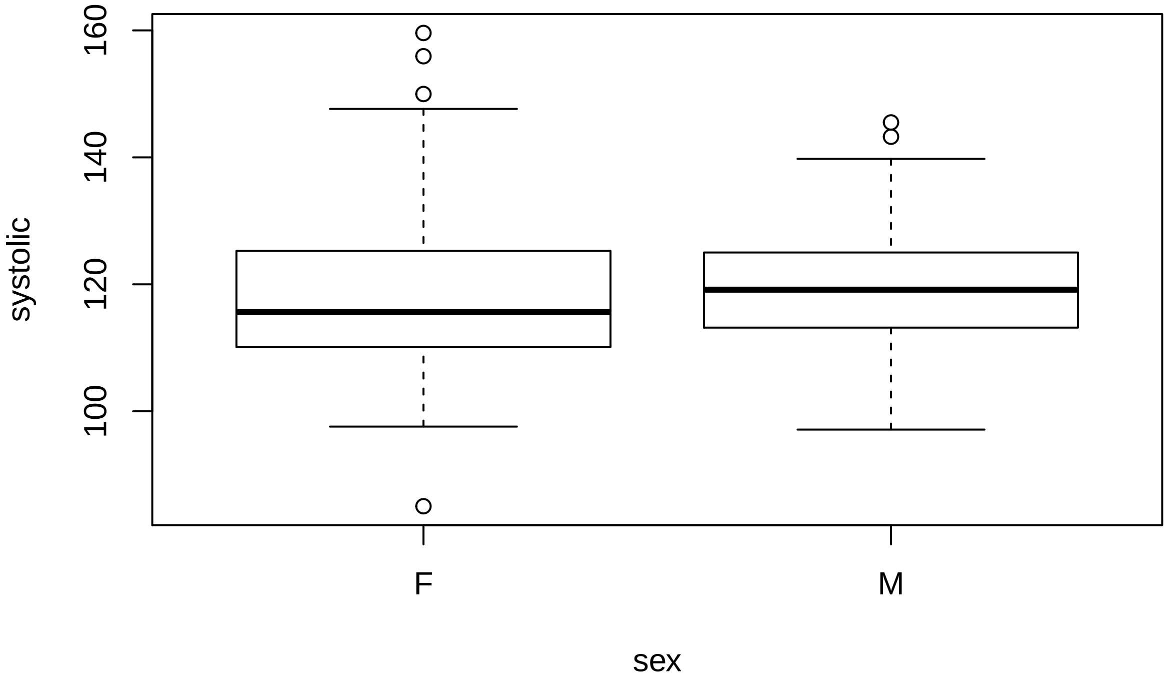
```
plot( systolic ~ car, data=bpdata)
```



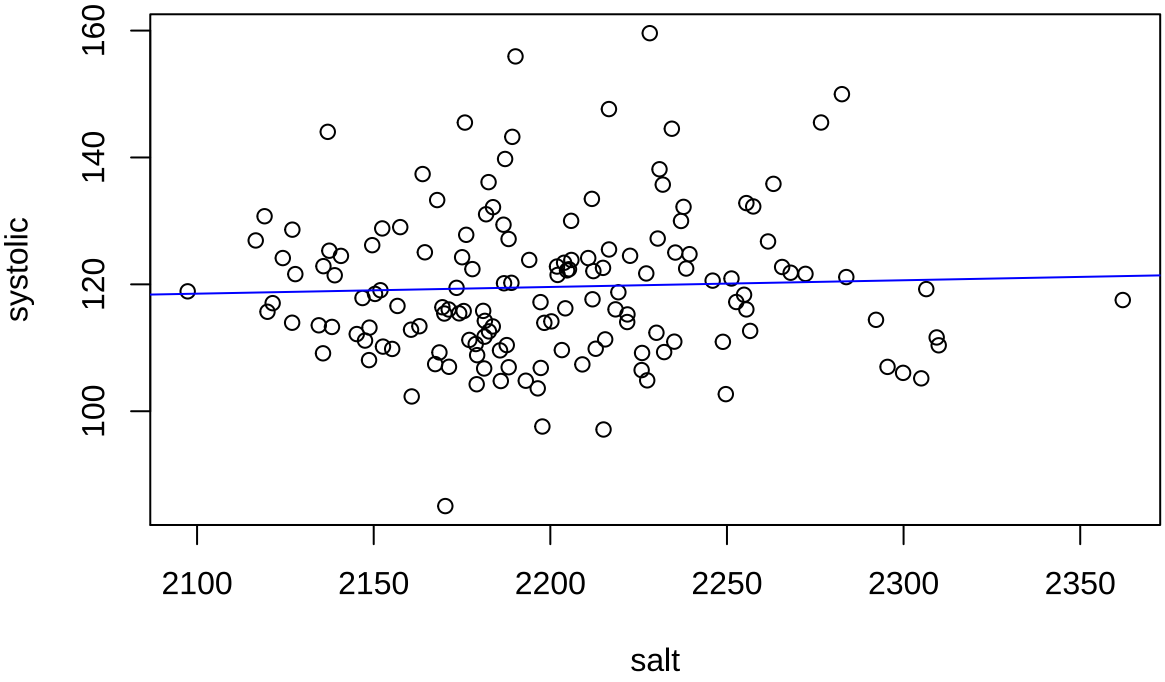
```
plot( systolic ~ sign, data=bpdata)
```



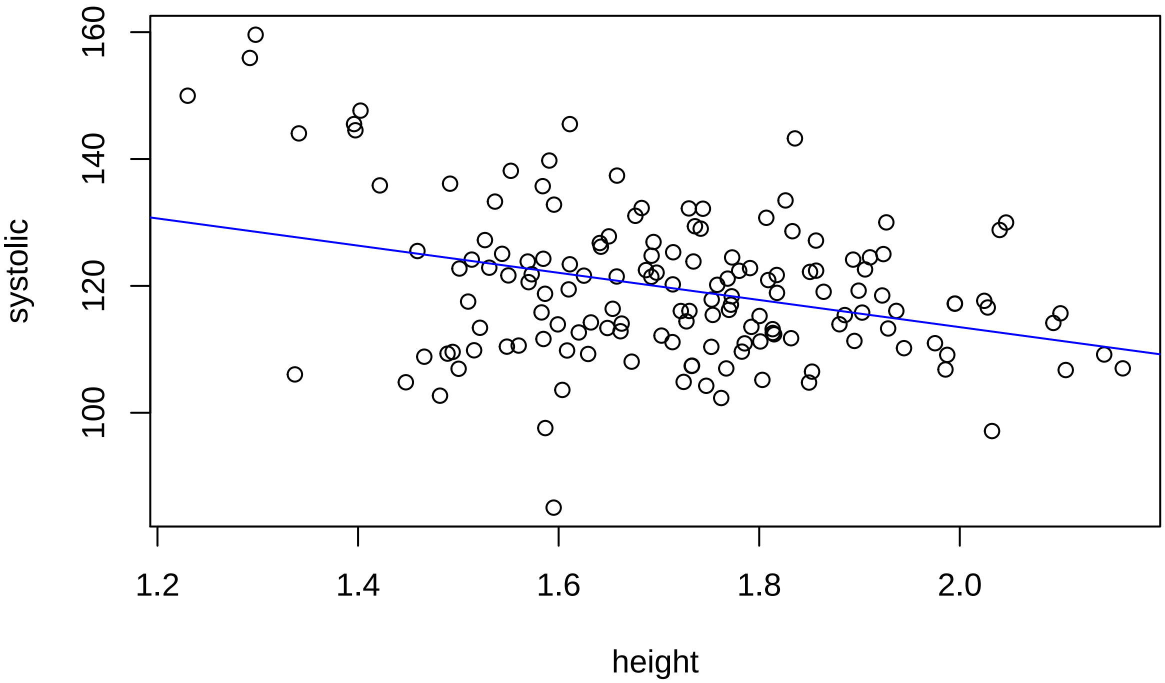
```
plot( systolic ~ sex, data=bpdata)
```



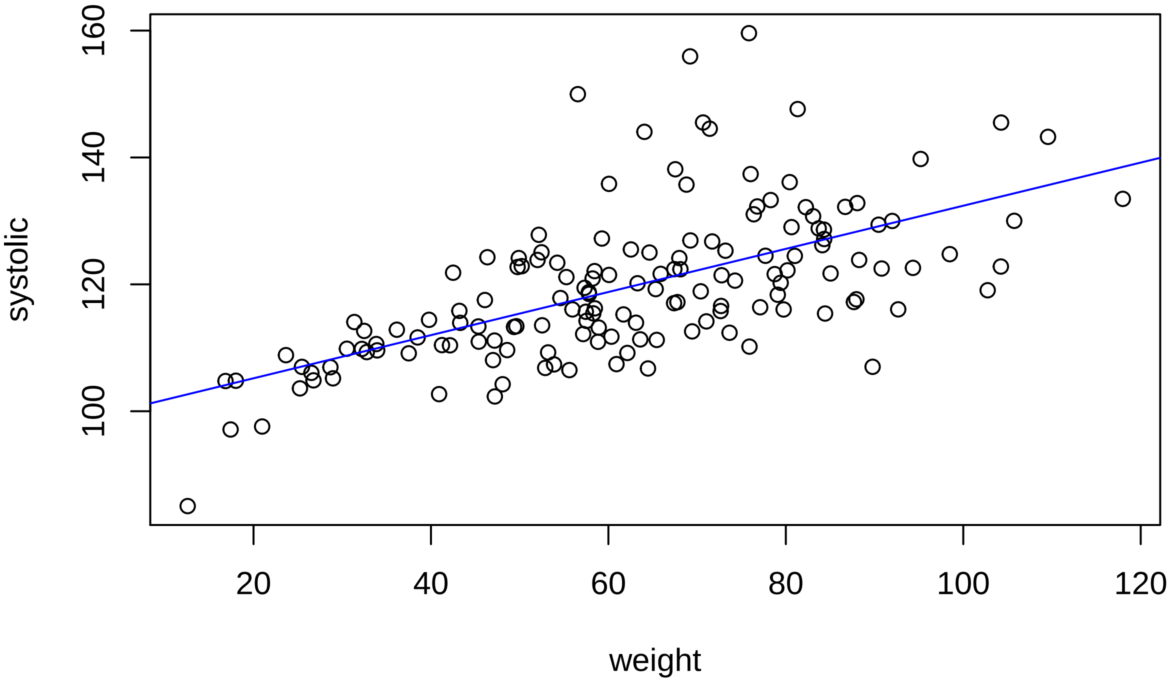
```
plot( systolic ~ salt, data=bpdata)
abline( lm(systolic ~ salt, data=bpdata), col="blue")
```



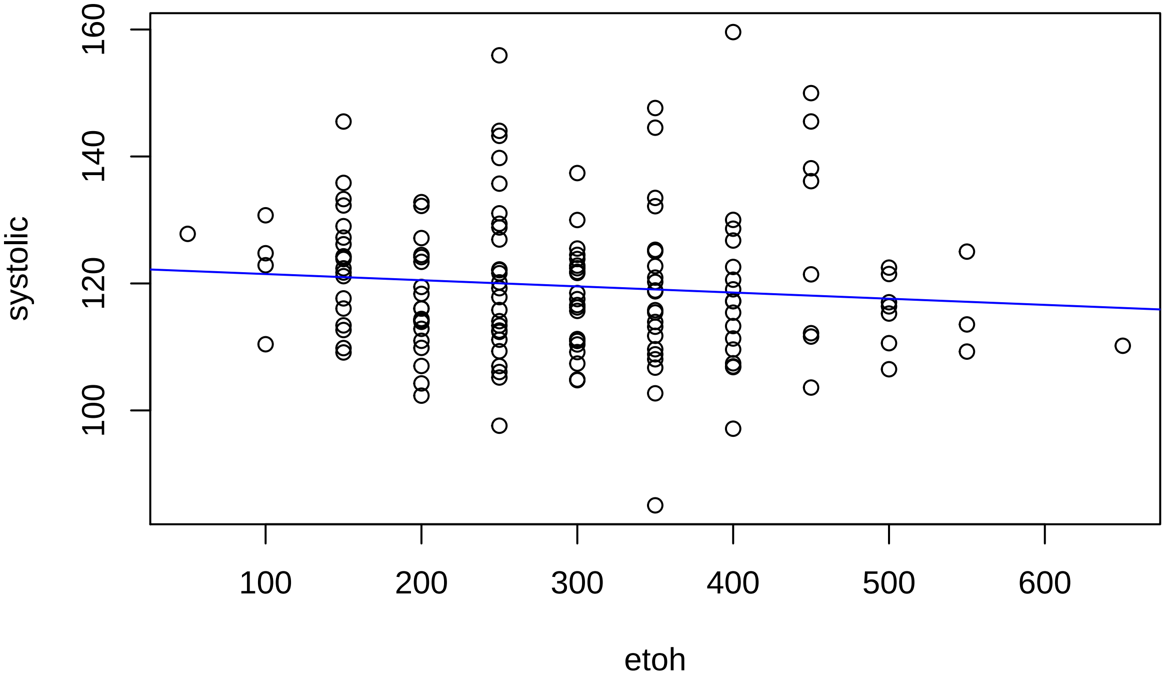
```
plot( systolic ~ height, data=bpdata)
abline( lm(systolic ~ height, data=bpdata), col="blue")
```



```
plot( systolic ~ weight, data=bpdata)
abline( lm(systolic ~ weight, data=bpdata), col="blue")
```

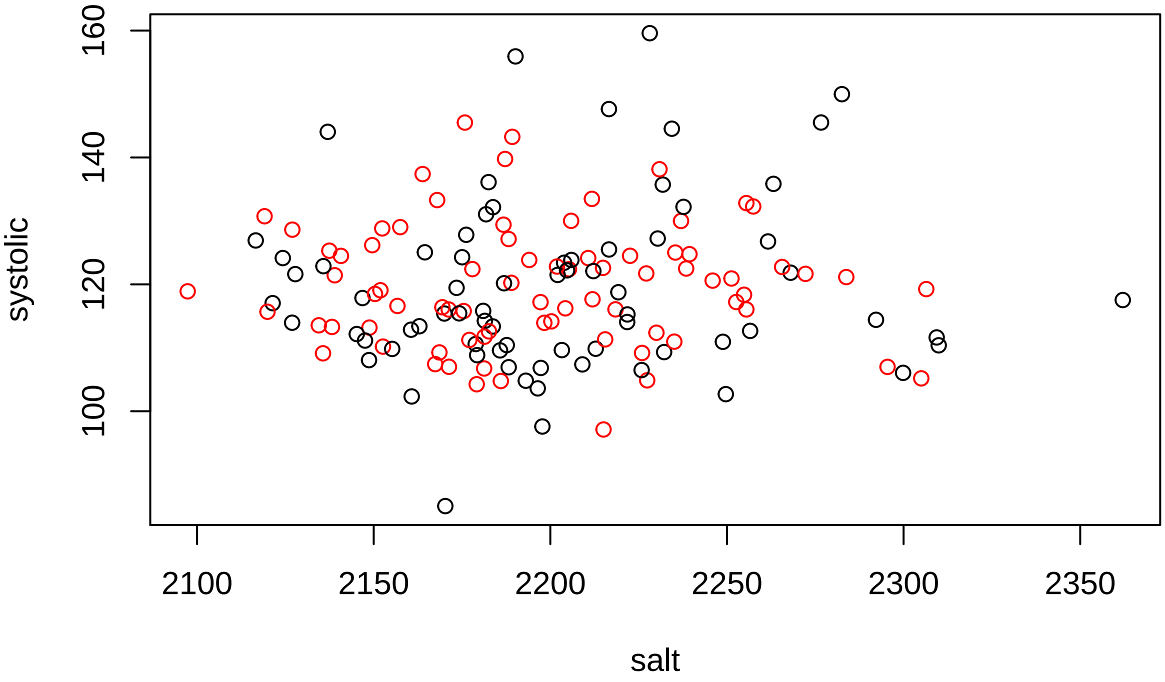


```
plot( systolic ~ etoh, data=bpdata)
abline( lm(systolic ~ etoh, data=bpdata), col="blue")
```

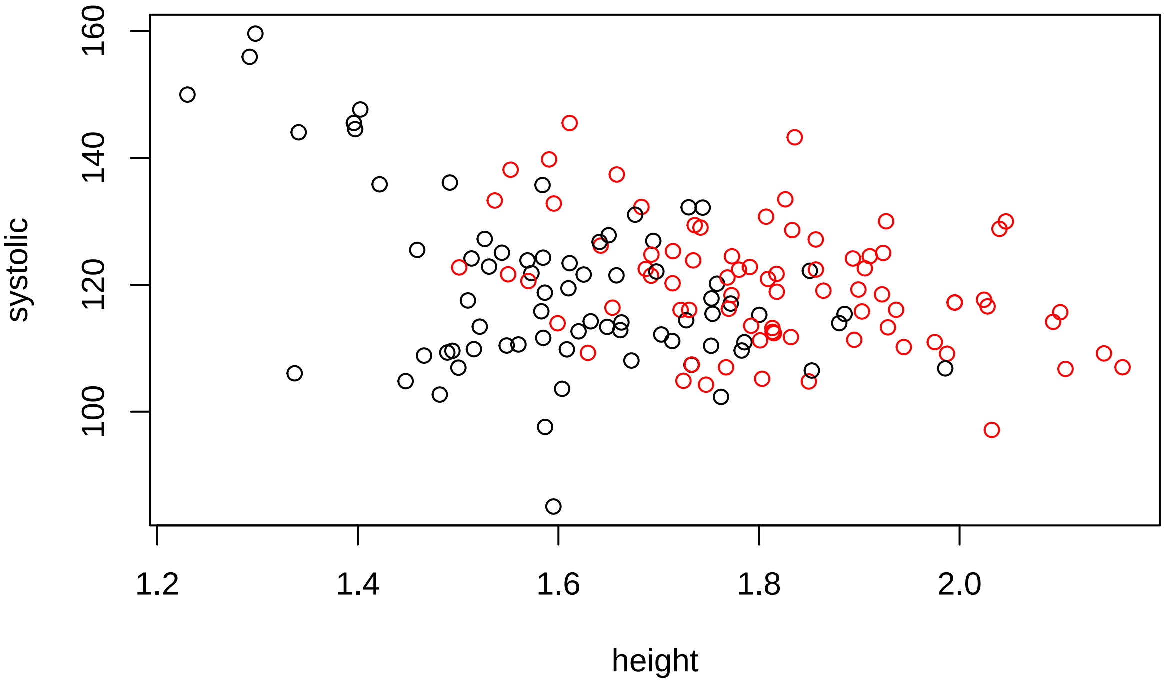


Interactions between continuous predictors and sex

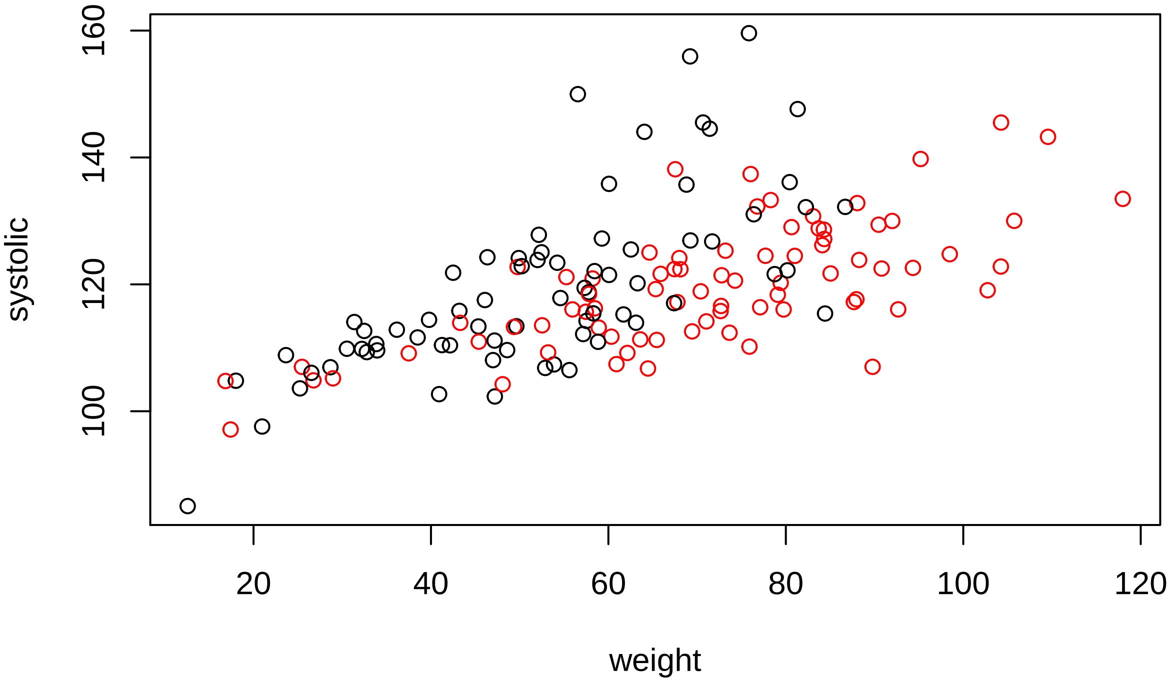
```
# ggplot scatterplot
# ggplot( data=bpdata, aes(x=height, y=systolic, col=sex)) + geom_point()

plot( systolic ~ salt, col=sex, data=bpdata)
```



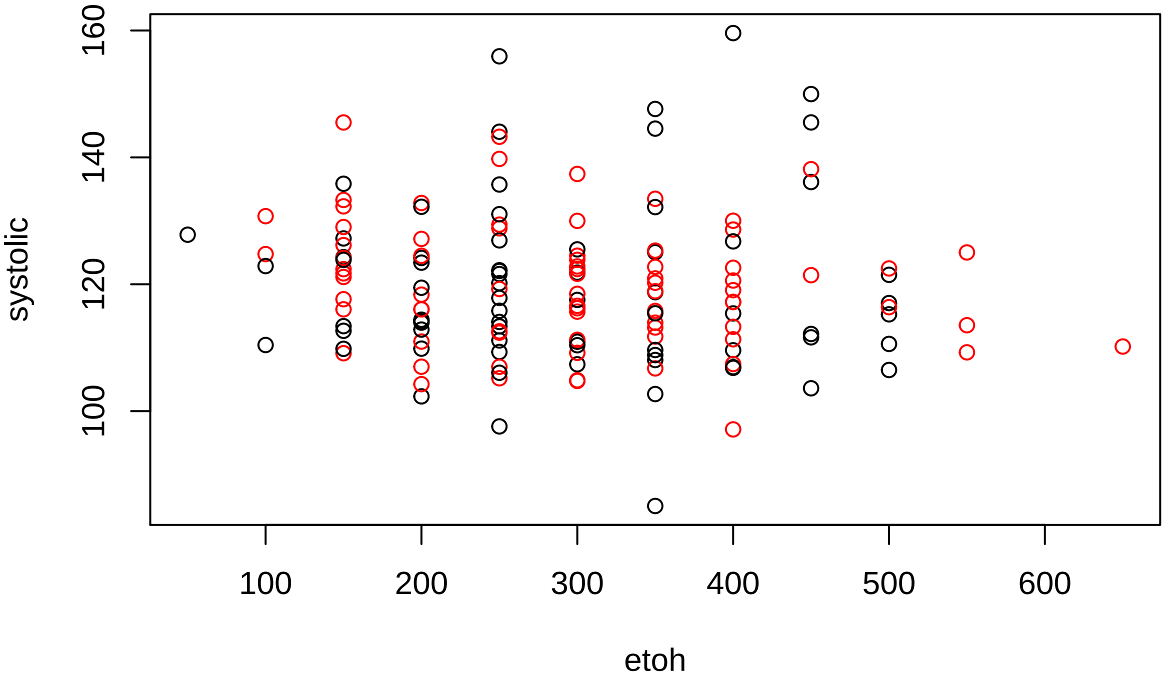
```
plot( systolic ~ height, col=sex, data=bpdata)
```



```
plot( systolic ~ weight, col=sex, data=bpdata)
```

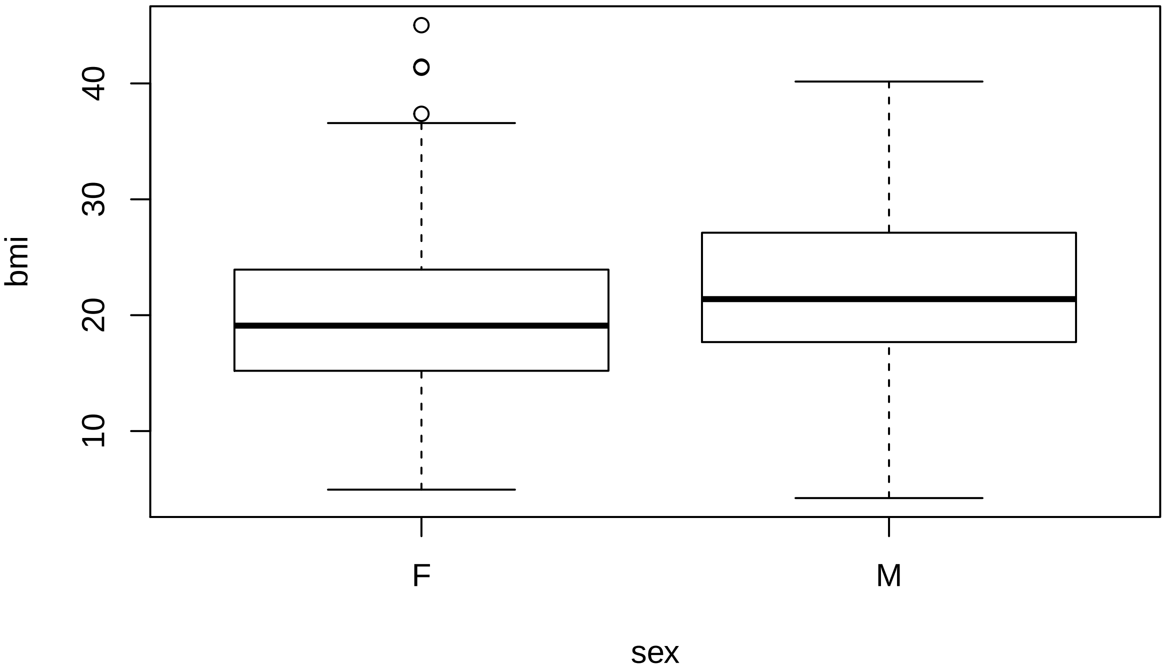


```
plot( systolic ~ etoh, col=sex, data=bpdata)
```

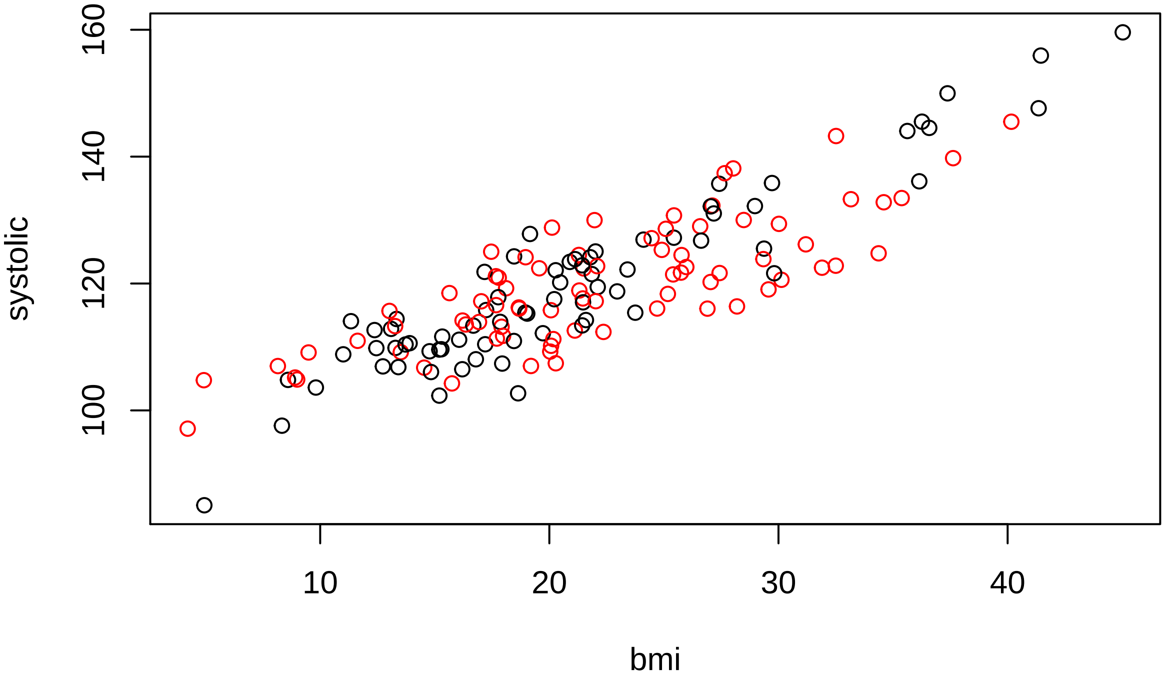


The simulation uses height and weight to calculate body mass index (BMI), which is linearly related to the outcome. This transformation of variables is left for the analyst to discover, but we will add the transformed column to the data set to simplify our notation. Note that we can perform transformations dynamically in the model formula, without actually changing the dataframe.

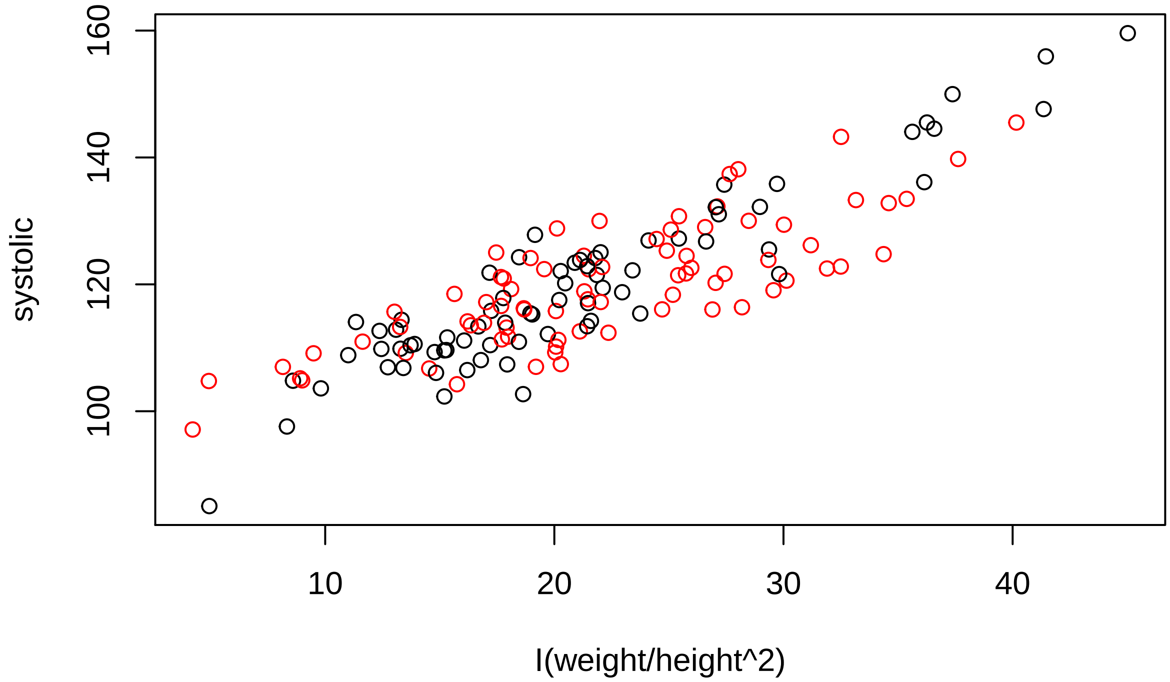
```
bpdata <- transform(bpdata, bmi = weight/(height^2))

plot( bmi ~ sex, data=bpdata )
```



```
plot( systolic ~ bmi, col=sex, data=bpdata)
```



```
# same thing, without adding a column to the data frame
plot( systolic ~ I(weight/height^2), col=sex, data=bpdata)
```



Some of the attributes in the dataset are distractors, and they have nothing to do with the outcome. The type of car a person drives and their zodiac sign are examples of categorical distractors.