Exploratory Data Analysis

Reading *.csv Data

```r
site <- "https://raw.githubusercontent.com/alanarnholt/STT3850/gh-pages/DataCSV/FlightDelays.csv"
FlightDelays <- read.csv(file=url(site))
head(FlightDelays) # shows first 6 rows of data frame
```

```
##   ID Carrier FlightNo Destination DepartTime Day Month FlightLength Delay
## 1  1      UA      403         DEN      4-8am Fri   May          281    -1
## 2  2      UA      405         DEN    8-Noon Fri   May          277   102
## 3  3      UA      409         DEN      4-8pm Fri   May          279     4
## 4  4      UA      511         ORD    8-Noon Fri   May          158    -2
## 5  5      UA      667         ORD      4-8am Fri   May          143    -3
## 6  6      UA      669         ORD      4-8am Fri   May          150     0
##   Delayed30
## 1        No
## 2       Yes
## 3        No
## 4        No
## 5        No
## 6        No
```

```r
str(FlightDelays)  # shows structure of data frame
```

```
## 'data.frame':    4029 obs. of  10 variables:
##  $ ID          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Carrier     : Factor w/ 2 levels "AA","UA": 2 2 2 2 2 2 2 2 2 2 ...
##  $ FlightNo    : int  403 405 409 511 667 669 673 677 679 681 ...
##  $ Destination : Factor w/ 7 levels "BNA","DEN","DFW",..: 2 2 2 6 6 6 6 6 6 6 ...
##  $ DepartTime  : Factor w/ 5 levels "4-8am","4-8pm",..: 1 4 2 4 1 1 4 4 5 5 ...
##  $ Day         : Factor w/ 7 levels "Fri","Mon","Sat",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Month       : Factor w/ 2 levels "June","May": 2 2 2 2 2 2 2 2 2 2 ...
##  $ FlightLength: int  281 277 279 158 143 150 158 160 160 163 ...
##  $ Delay       : int  -1 102 4 -2 -3 0 -5 0 10 60 ...
##  $ Delayed30   : Factor w/ 2 levels "No","Yes": 1 2 1 1 1 1 1 1 1 2 ...
```

```r
levels(FlightDelays$Month)
```

```
## [1] "June" "May"
```

```r
FlightDelays$Month <- factor(FlightDelays$Month, levels = c("May", "June"))
levels(FlightDelays$Month)
```

```
## [1] "May"  "June"
```

Creating Tables

```r
table(FlightDelays$Carrier)
```

```
## 
##   AA   UA 
## 2906 1123
```

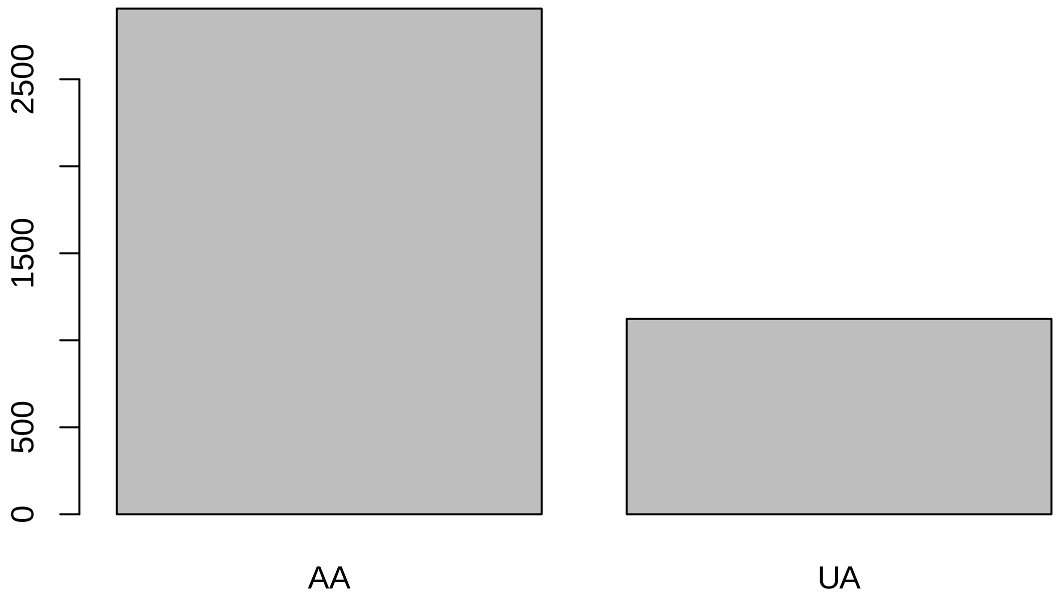
```r
xtabs(~Carrier, data = FlightDelays)
```

```
## Carrier
##   AA   UA
## 2906 1123
```

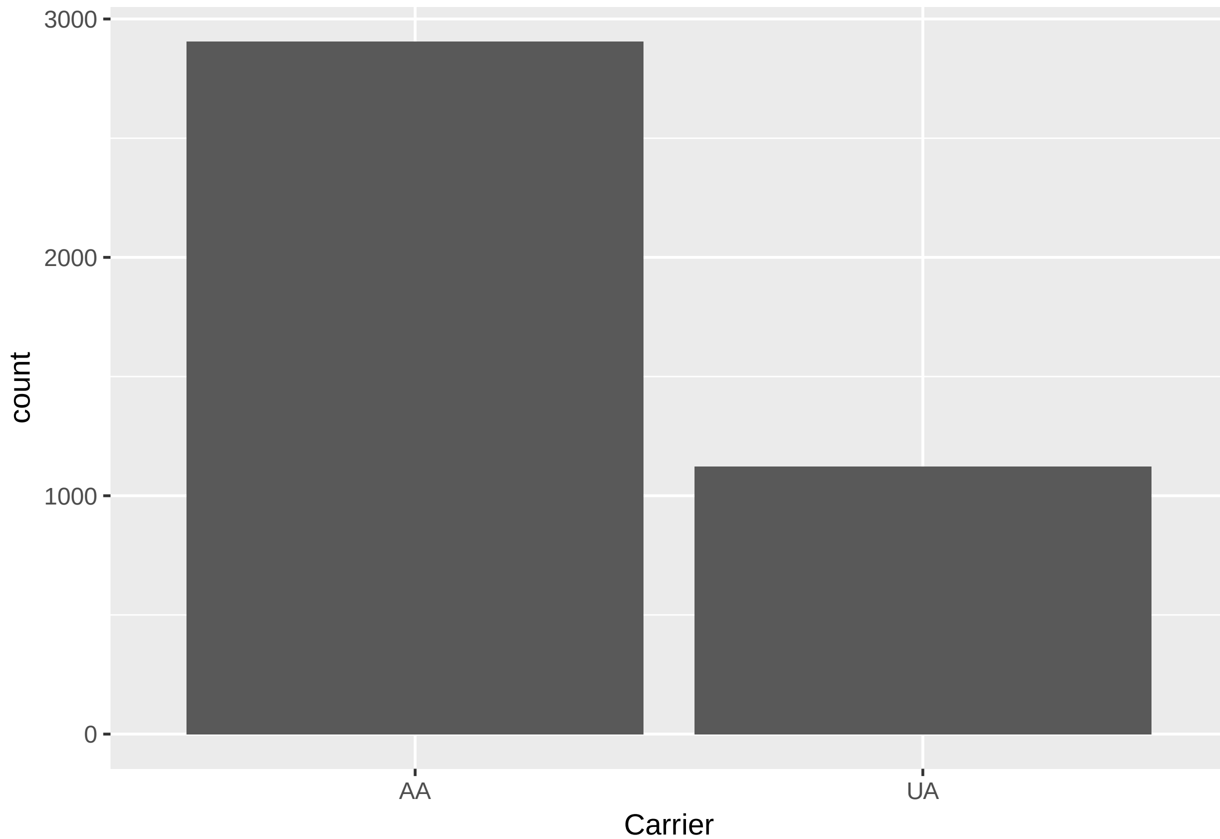
Creating Barplots

```r
barplot(table(FlightDelays$Carrier))
require(ggplot2)
```

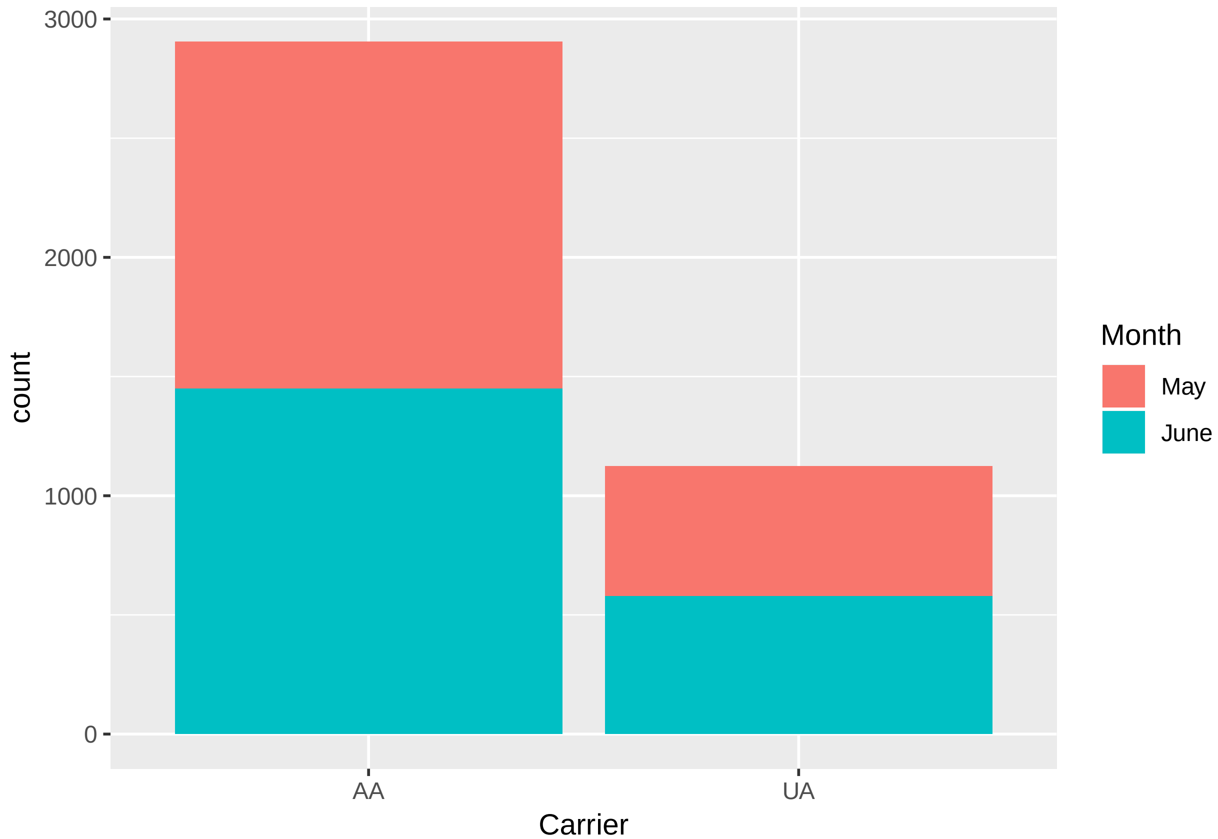
```
## Loading required package: ggplot2
```



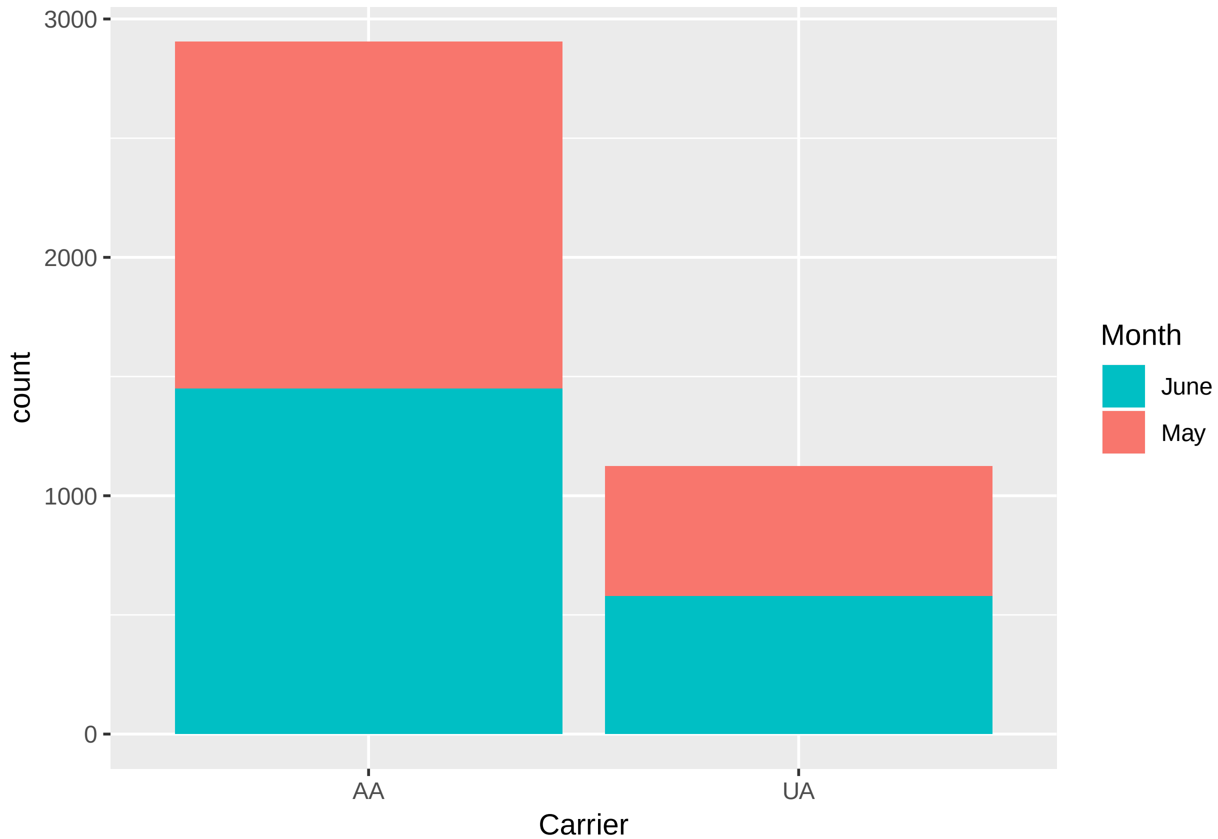

```r
ggplot(data = FlightDelays, aes(x = Carrier)) +
  geom_bar()
```



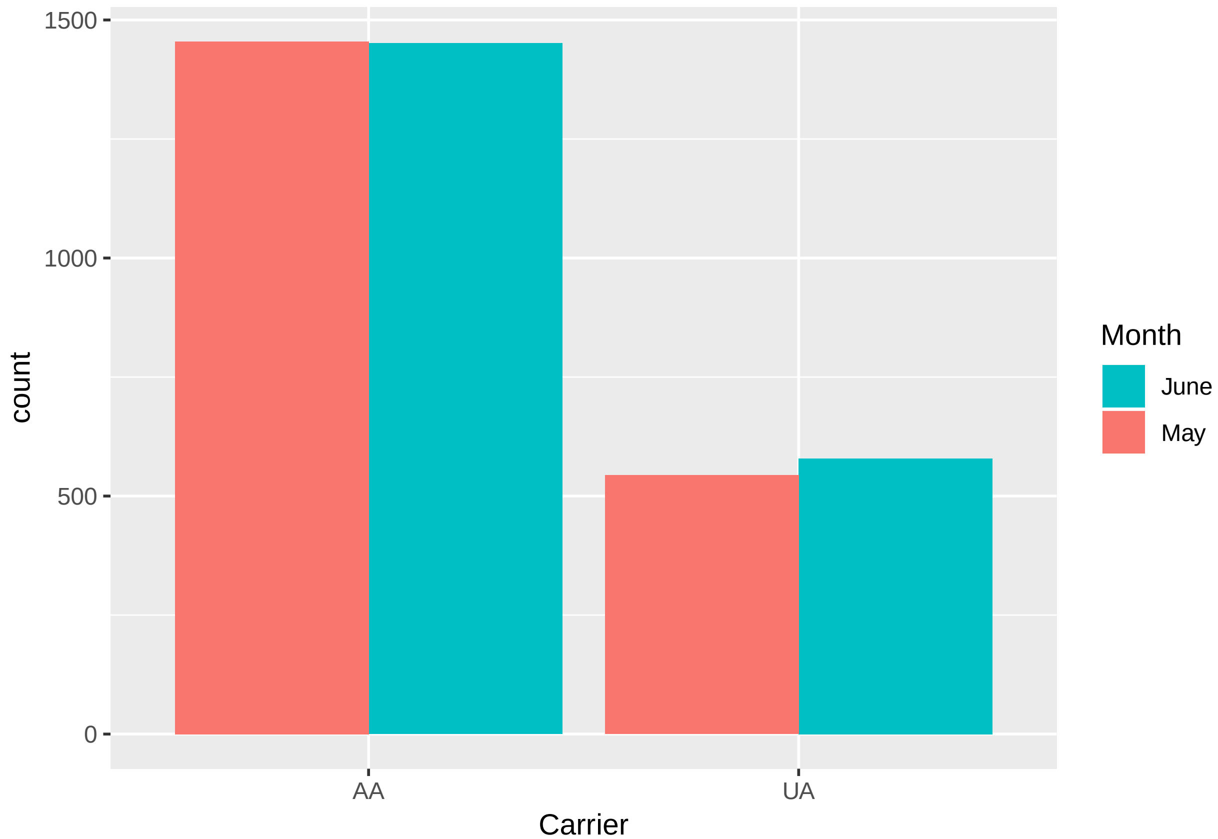

```r
ggplot(data = FlightDelays, aes(x = Carrier, fill= Month)) +
  geom_bar()
```




```r
ggplot(data = FlightDelays, aes(x = Carrier, fill= Month)) +
  geom_bar() +
  guides(fill = guide_legend(reverse = TRUE))
```




```r
ggplot(data = FlightDelays, aes(x = Carrier, fill= Month)) +
  geom_bar(position="dodge") +
  guides(fill = guide_legend(reverse = TRUE))
```



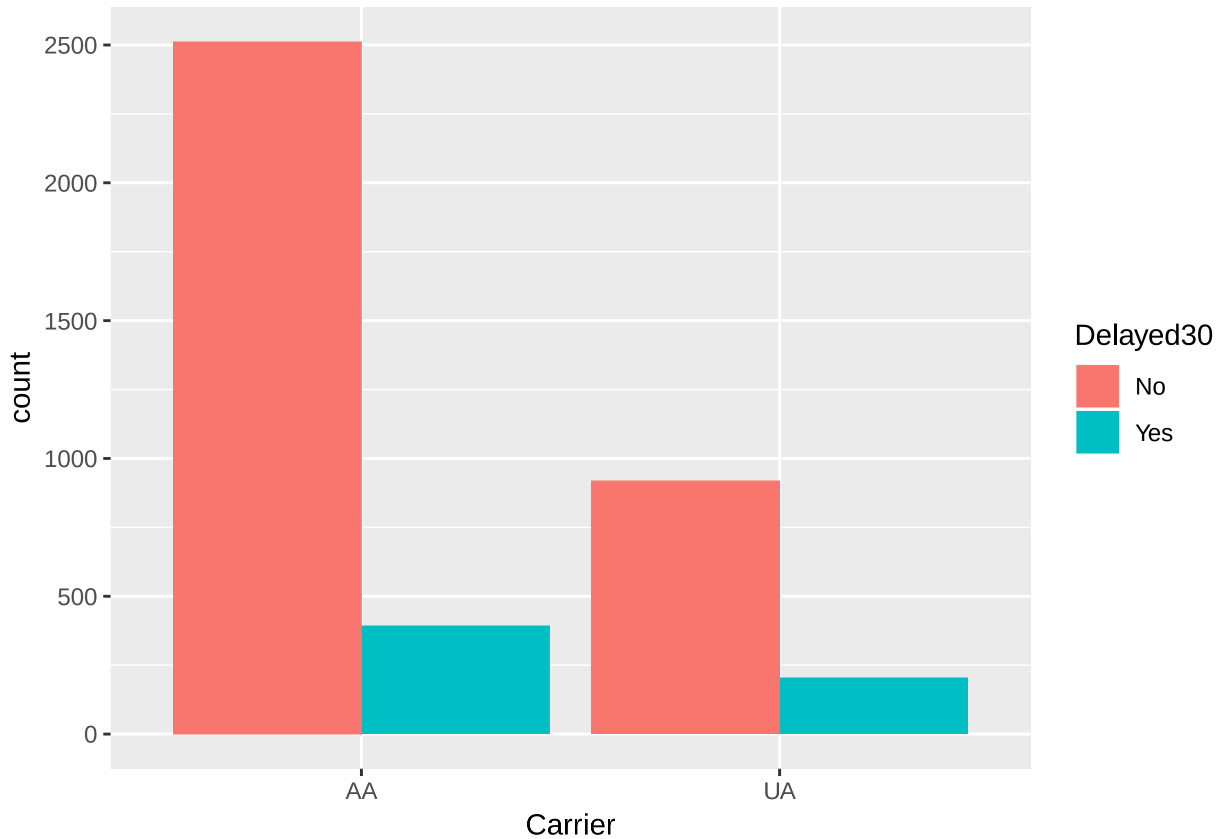

```r
xtabs(~ Carrier + (Delay > 30), data = FlightDelays)
```

```
##        Delay > 30
## Carrier FALSE TRUE
##      AA  2513  393
##      UA   919  204
```

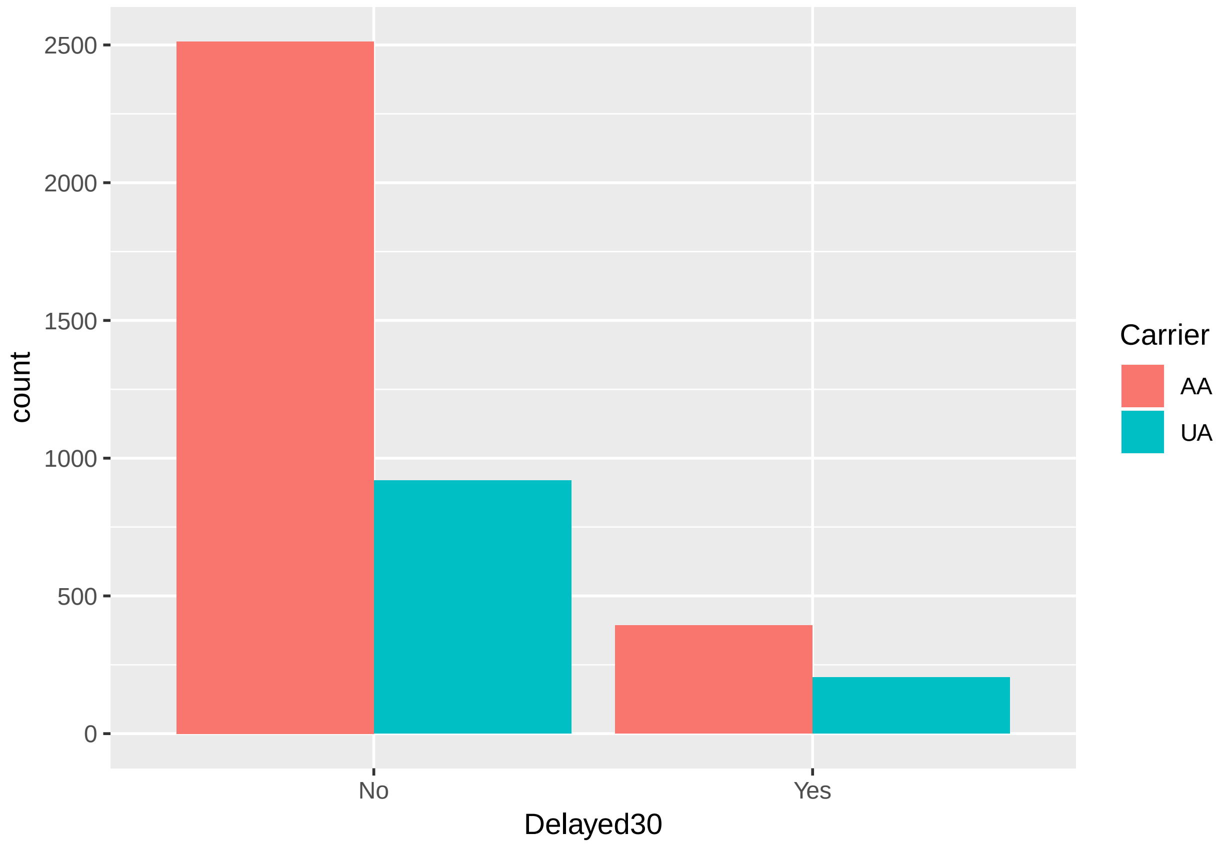
```r
addmargins(xtabs(~ Carrier + (Delay > 30), data = FlightDelays))
```

```
##        Delay > 30
## Carrier FALSE TRUE  Sum
##     AA   2513  393 2906
##     UA    919  204 1123
##     Sum  3432  597 4029
```

```r
ggplot(data = FlightDelays, aes(x = Carrier, fill= Delayed30)) +
  geom_bar(position="dodge")
```

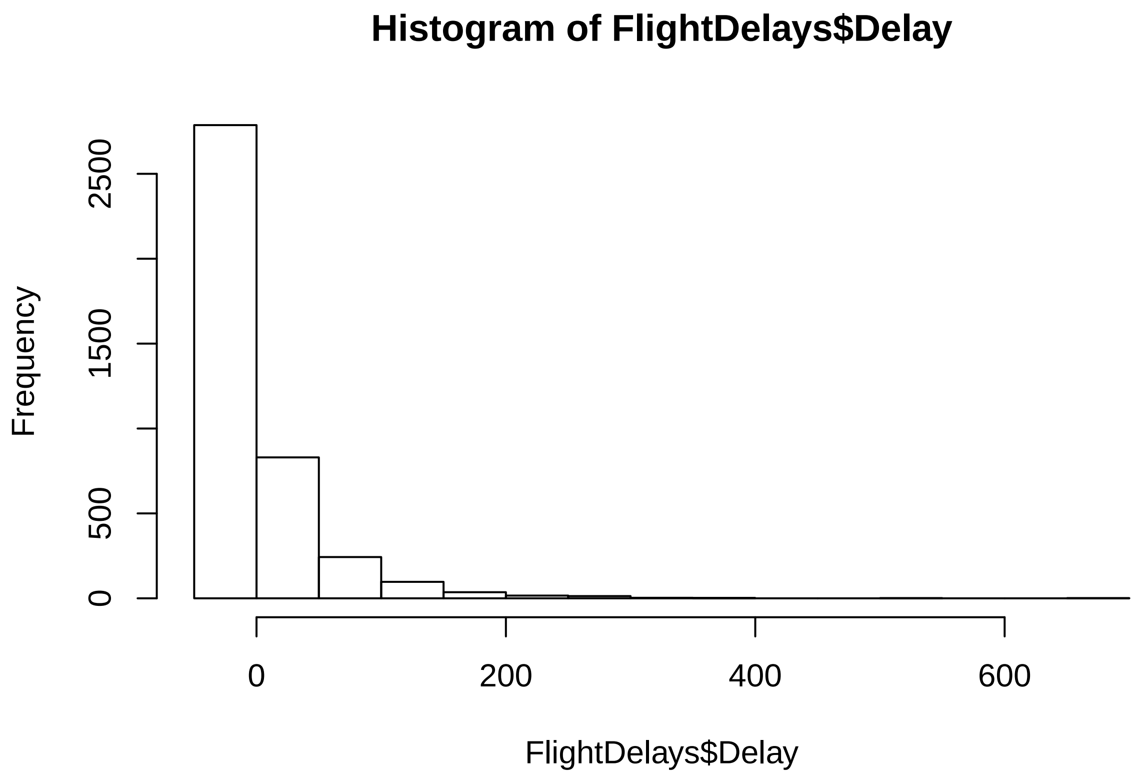



```r
ggplot(data = FlightDelays, aes(fill = Carrier, x= Delayed30)) +
  geom_bar(position="dodge")
```

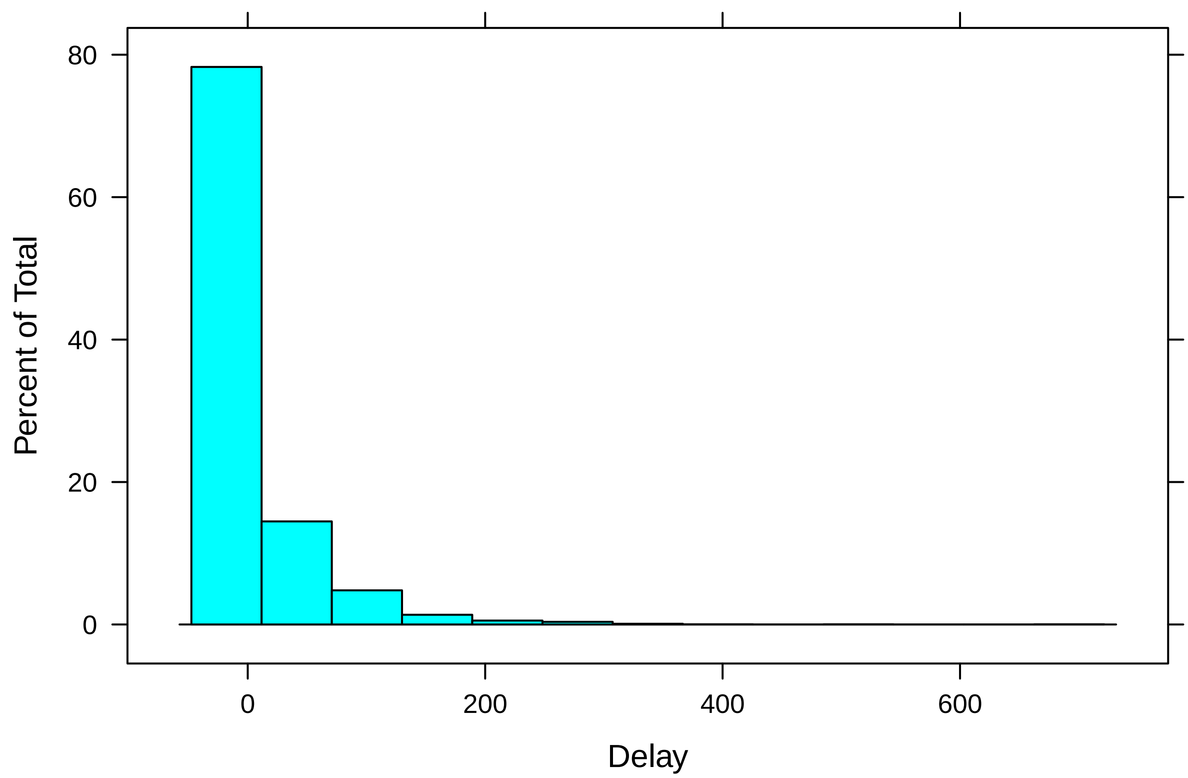


Histograms of `Delay` values.

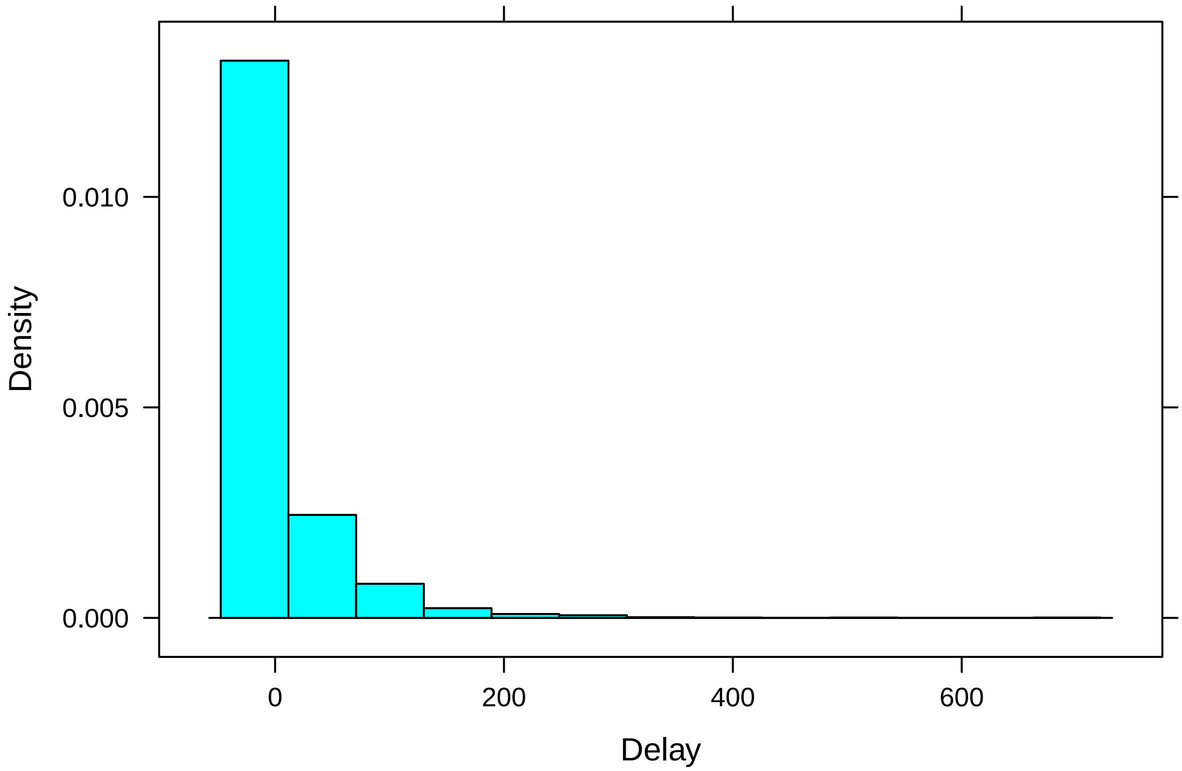
```
hist(FlightDelays$Delay)   # Ugly with Defaults...you change
```



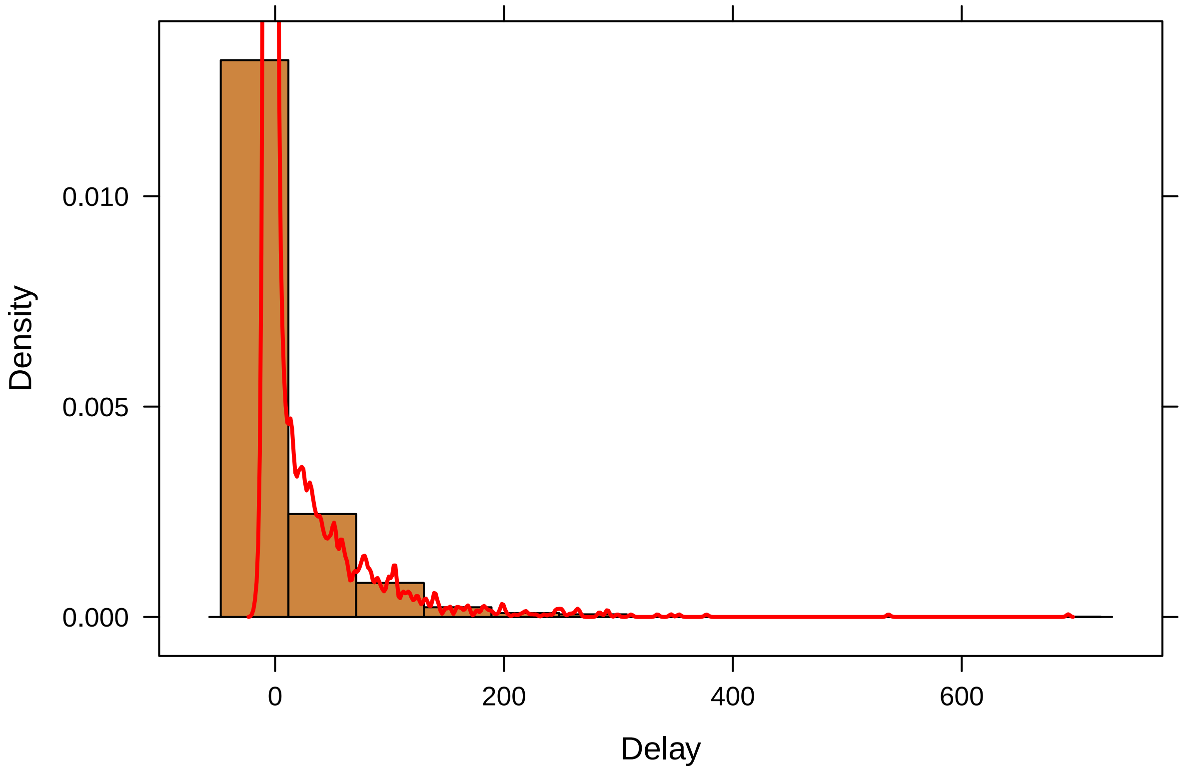

```r
library(lattice)
histogram(~Delay, data = FlightDelays)
```



```r
histogram(~Delay, data = FlightDelays, type = "density")
```

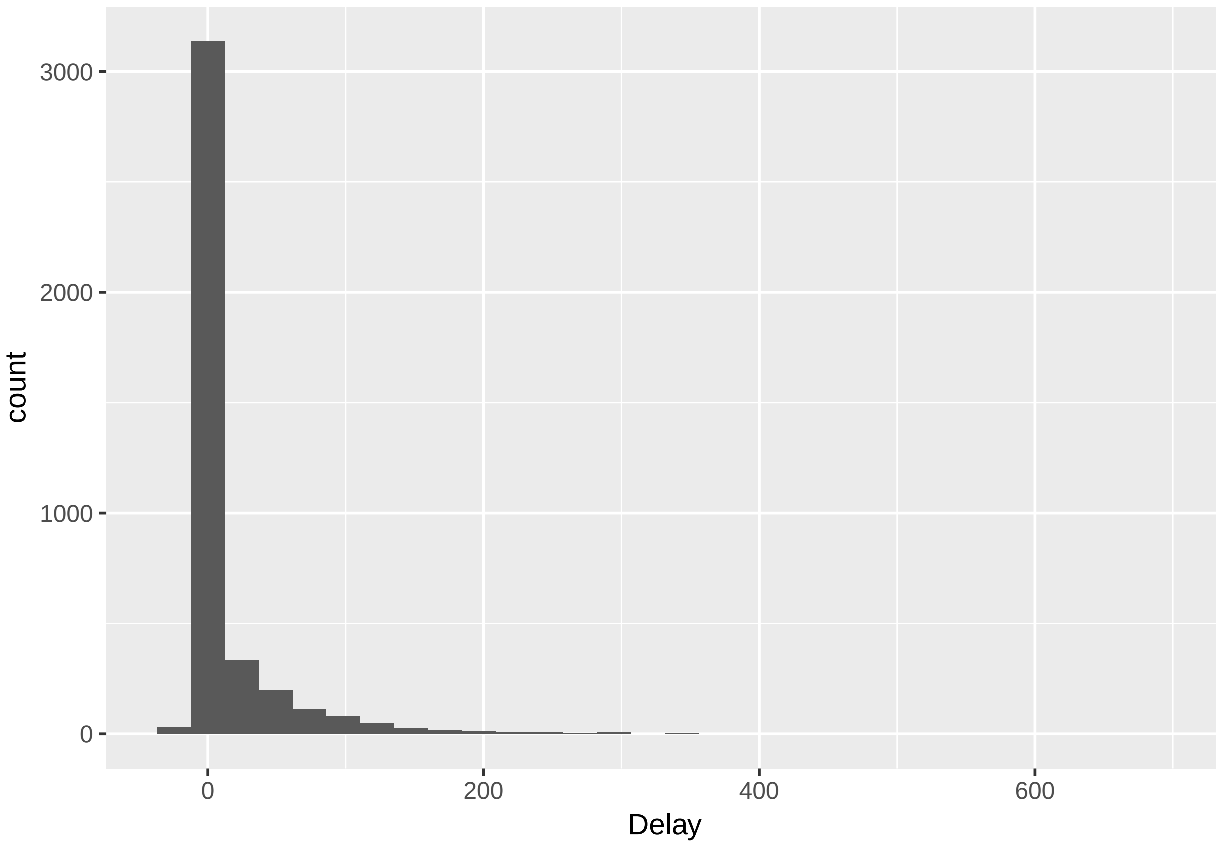



```r
histogram(~Delay, data = FlightDelays, type = "density",
          panel = function(...){
            panel.histogram(col = "peru",...)
            panel.densityplot(col = "red", lwd = 2, ...)
          }
          )
```

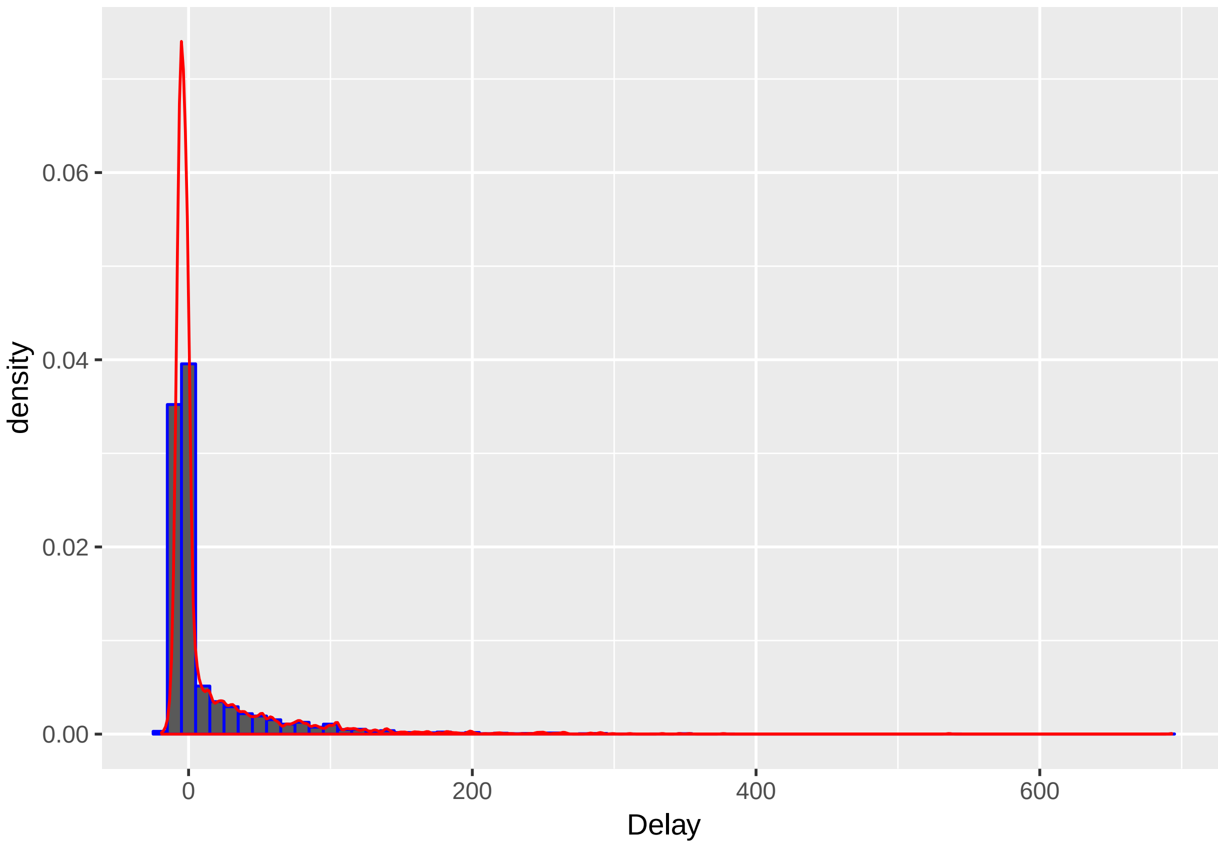



```r
ggplot(data = FlightDelays, aes(x = Delay)) +
  geom_histogram()
```

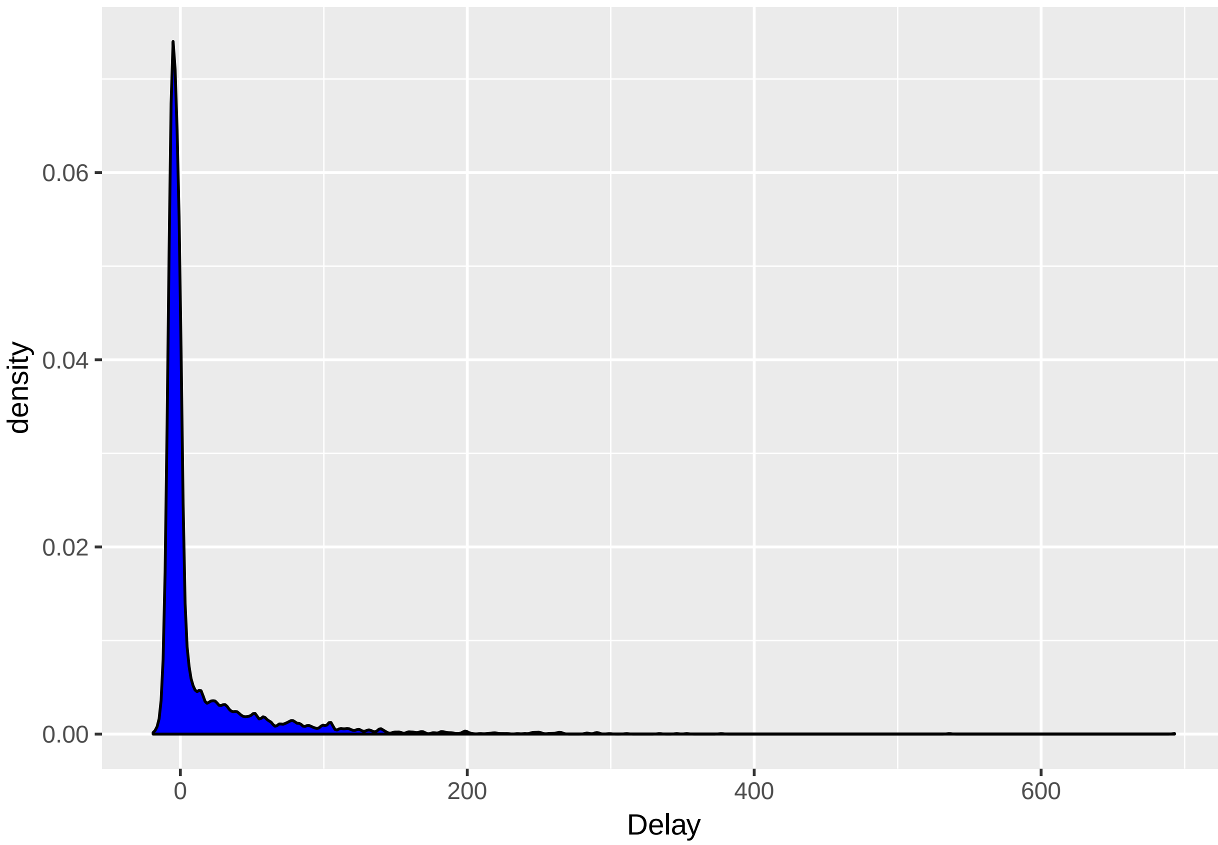
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(data = FlightDelays, aes(x = Delay, y = ..density..)) +
  geom_histogram(binwidth = 10, color = "blue") +
  geom_density(color = "red")
```




```r
ggplot(data = FlightDelays, aes(x = Delay)) +
  geom_density(fill = "blue")
```



Numeric Summaries

```r
summary(FlightDelays)
```

```
##        ID        Carrier      FlightNo      Destination    DepartTime
##  Min.   :   1   AA:2906   Min.   :  71.0   BNA: 172    4-8am   : 699
##  1st Qu.:1008   UA:1123   1st Qu.: 371.0   DEN: 264    4-8pm   : 972
##  Median :2015             Median : 691.0   DFW: 918    8-Mid   : 257
##  Mean   :2015             Mean   : 827.1   IAD:  55    8-Noon  :1053
##  3rd Qu.:3022             3rd Qu.: 787.0   MIA: 610    Noon-4pm:1048
##  Max.   :4029             Max.   :2255.0   ORD:1785
##                                            STL: 225
##   Day         Month        FlightLength        Delay          Delayed30
##  Fri:637   May :1999   Min.   : 68.0   Min.   :-19.00   No :3432
##  Mon:630   June:2030   1st Qu.:155.0   1st Qu.: -6.00   Yes: 597
##  Sat:453               Median :163.0   Median : -3.00
##  Sun:551               Mean   :185.3   Mean   : 11.74
##  Thu:566               3rd Qu.:228.0   3rd Qu.:  5.00
##  Tue:628               Max.   :295.0   Max.   :693.00
##  Wed:564
```

```r
sd(FlightDelays$Delay)
```

```
## [1] 41.6305
```

```r
sd(FlightDelays$Delay)^2
```

```
## [1] 1733.098
```

```r
var(FlightDelays$Delay)
```

```
## [1] 1733.098
```

```r
IQR(FlightDelays$Delay)
```

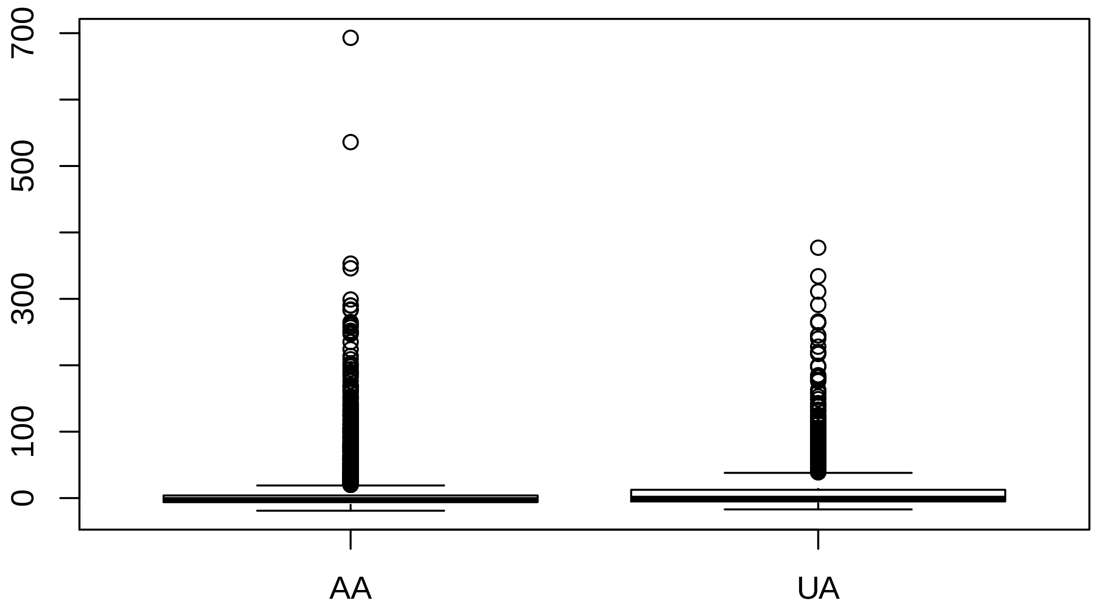
```
## [1] 11
```

```r
quantile(FlightDelays$Delay)
```

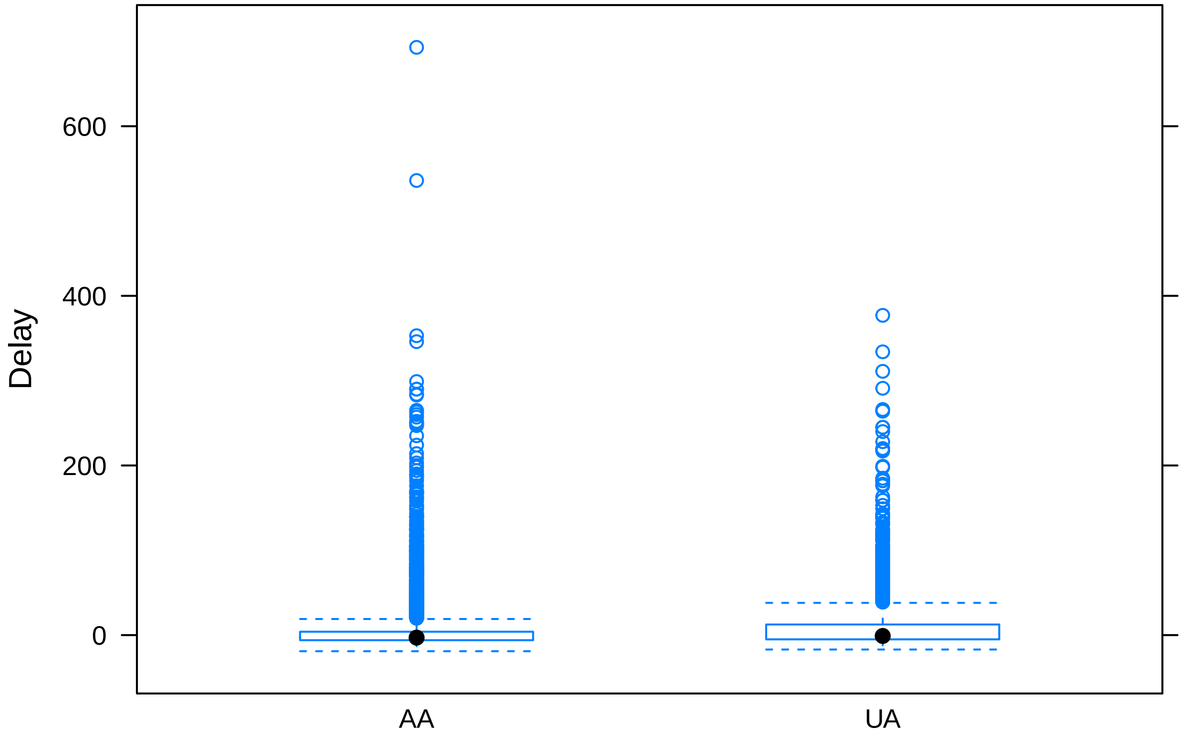
```
##   0%  25%  50%  75% 100%
##  -19   -6   -3    5  693
```

Boxplots

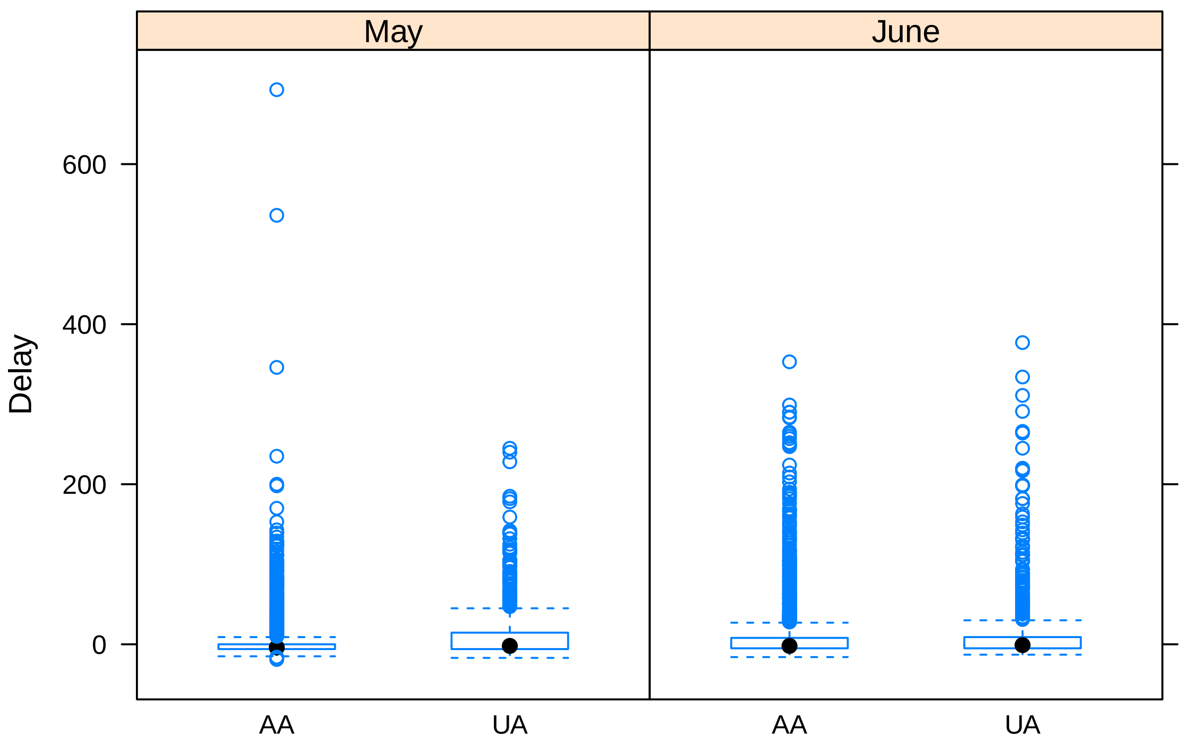
```r
boxplot(Delay ~ Carrier, data = FlightDelays)
```



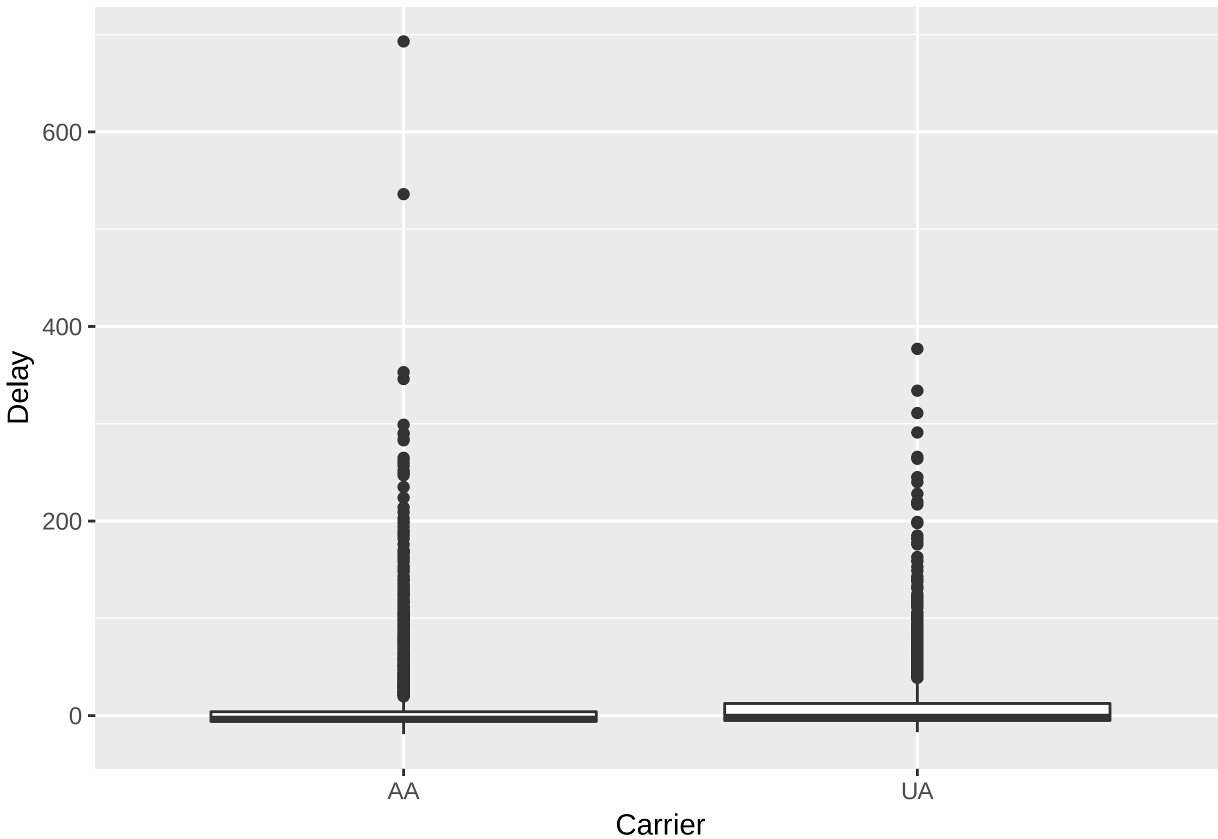

```r
bwplot(Delay ~ Carrier, data = FlightDelays)
```



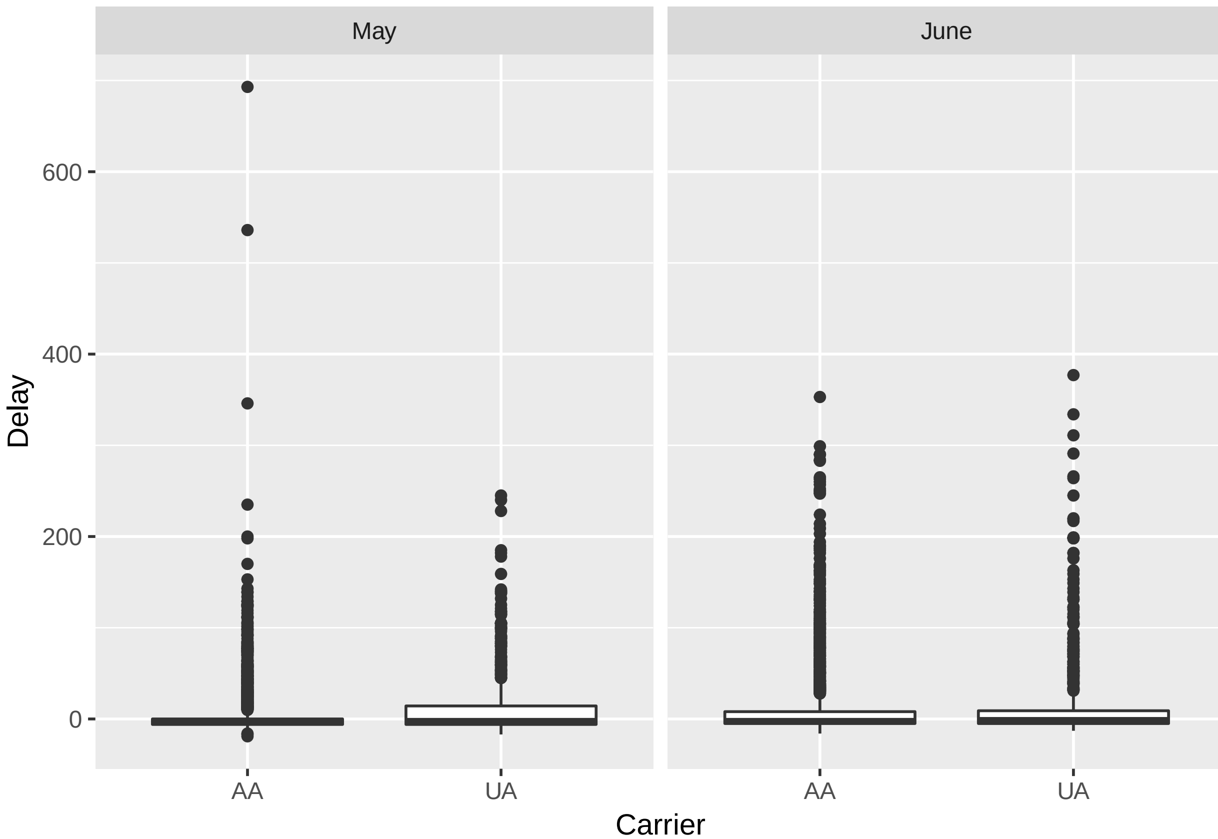

```
bwplot(Delay ~ Carrier|Month, data = FlightDelays, as.table = TRUE, layout = c(2, 1))
```




```
ggplot(data = FlightDelays, aes(x = Carrier, y = Delay)) + geom_boxplot()
```



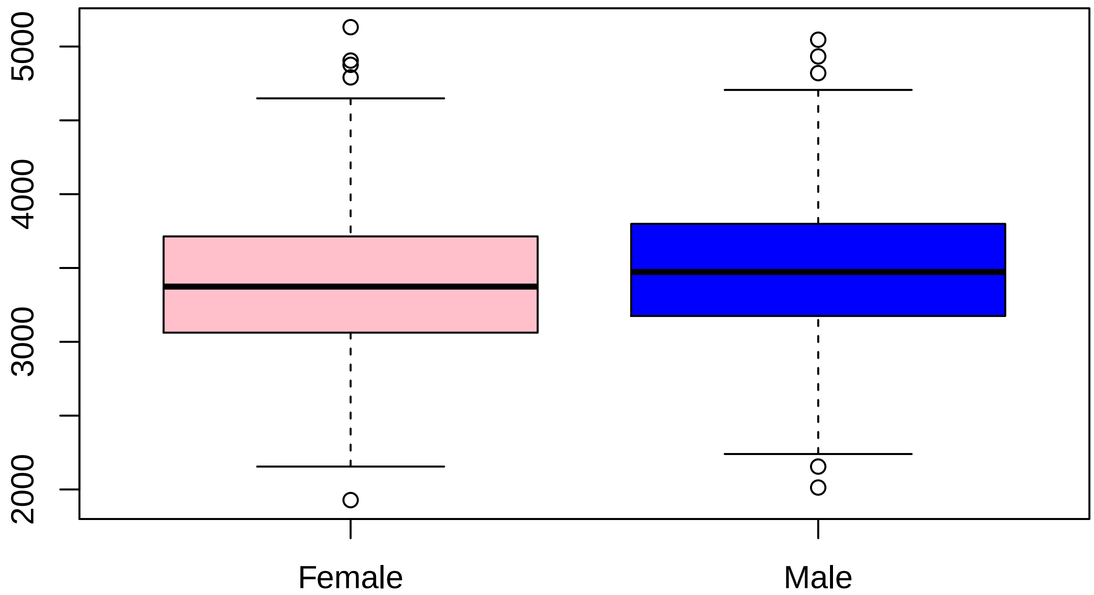

```
ggplot(data = FlightDelays, aes(x = Carrier, y = Delay)) + geom_boxplot() +
  facet_grid(. ~ Month)
```



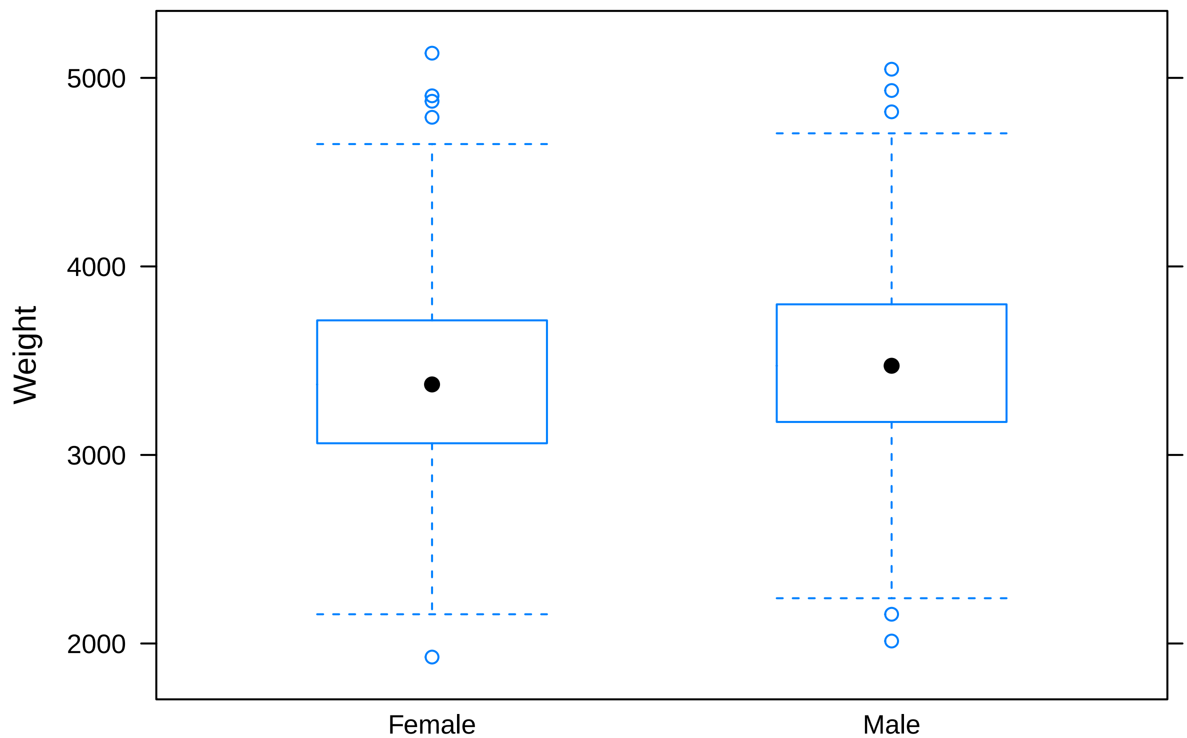

```r
site <- "https://raw.githubusercontent.com/alanarnholt/STT3850/gh-pages/DataCSV/NCBirths2004.csv"
NCBirths <- read.csv(file=url(site))
head(NCBirths)
```

```
##   ID MothersAge Tobacco Alcohol Gender Weight Gestation Smoker
## 1  1      30-34      No      No   Male   3827        40     No
## 2  2      30-34      No      No   Male   3629        38     No
## 3  3      35-39      No      No Female   3062        37     No
## 4  4      20-24      No      No Female   3430        39     No
## 5  5      25-29      No      No   Male   3827        38     No
## 6  6      35-39      No      No Female   3119        39     No
```

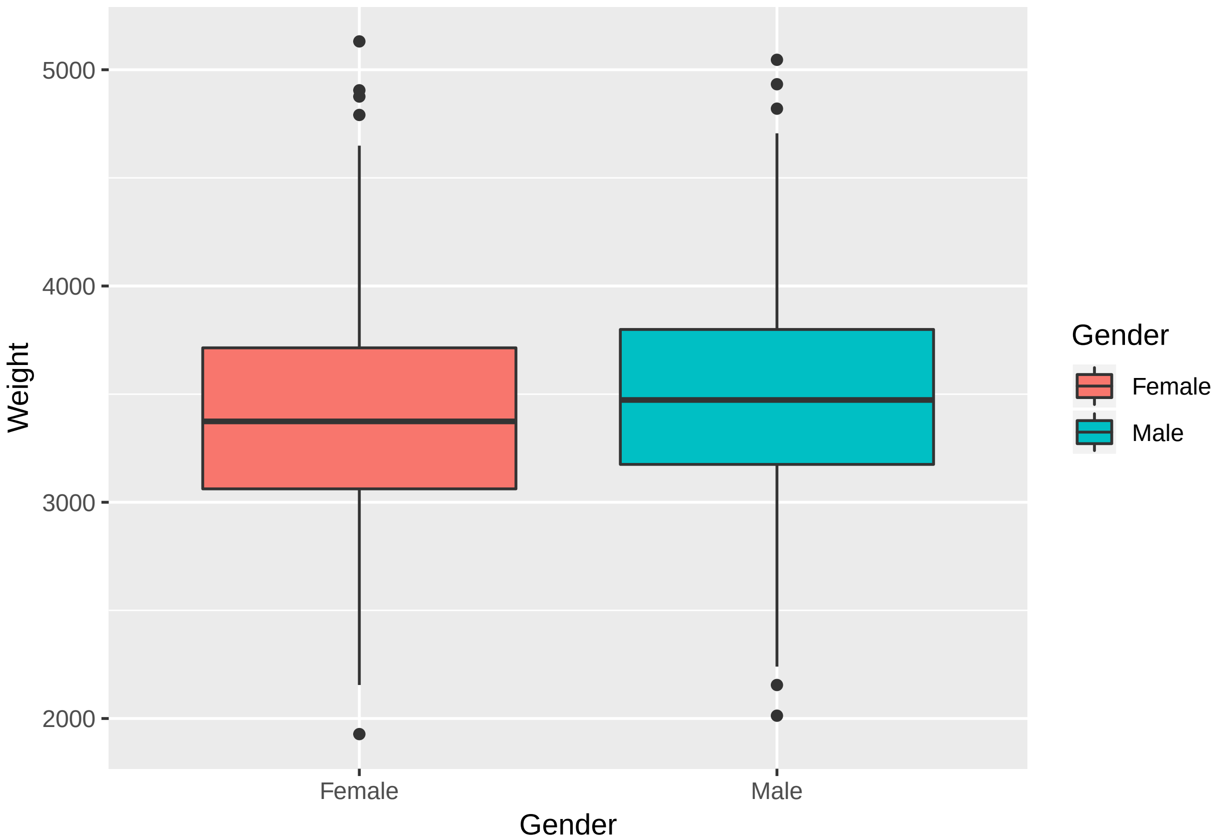
```r
boxplot(Weight ~ Gender, data = NCBirths, col = c("pink", "blue"))
```



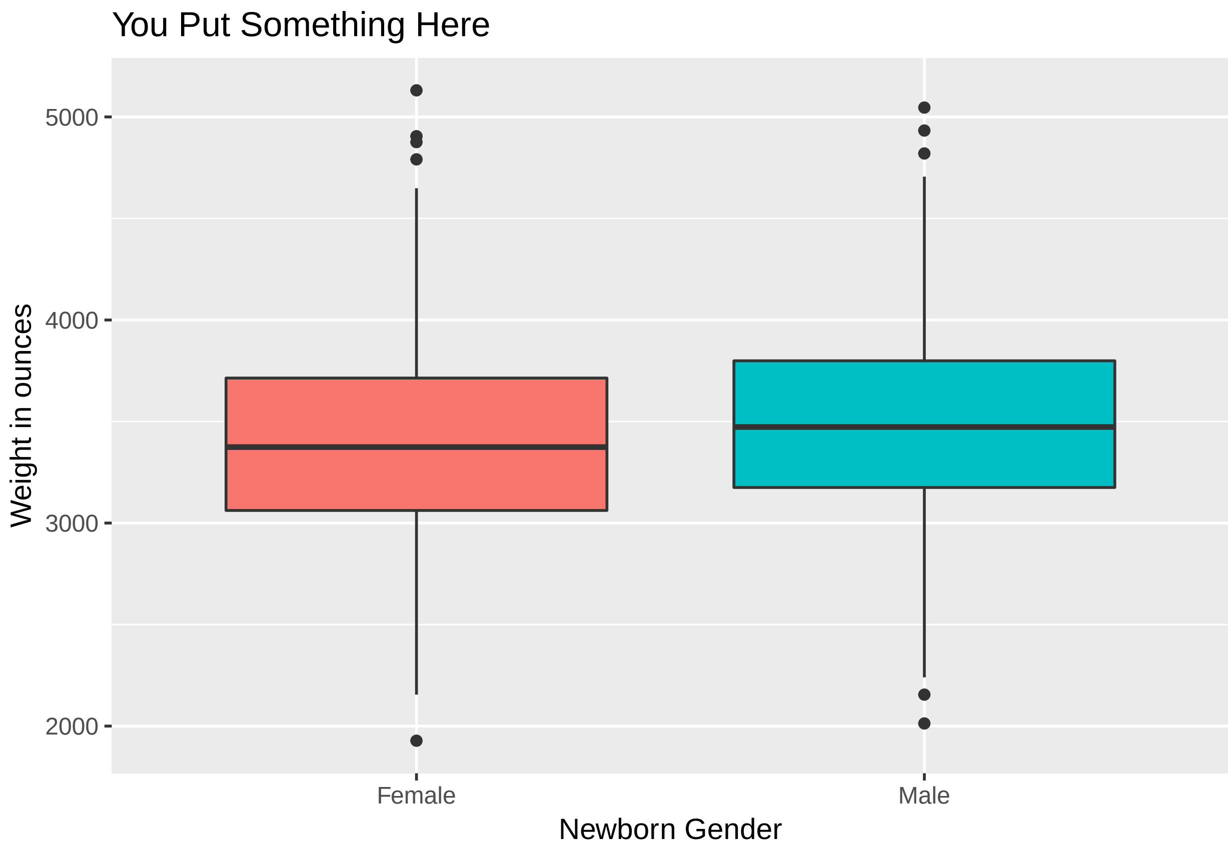

```r
bwplot(Weight ~ Gender, data = NCBirths)
```



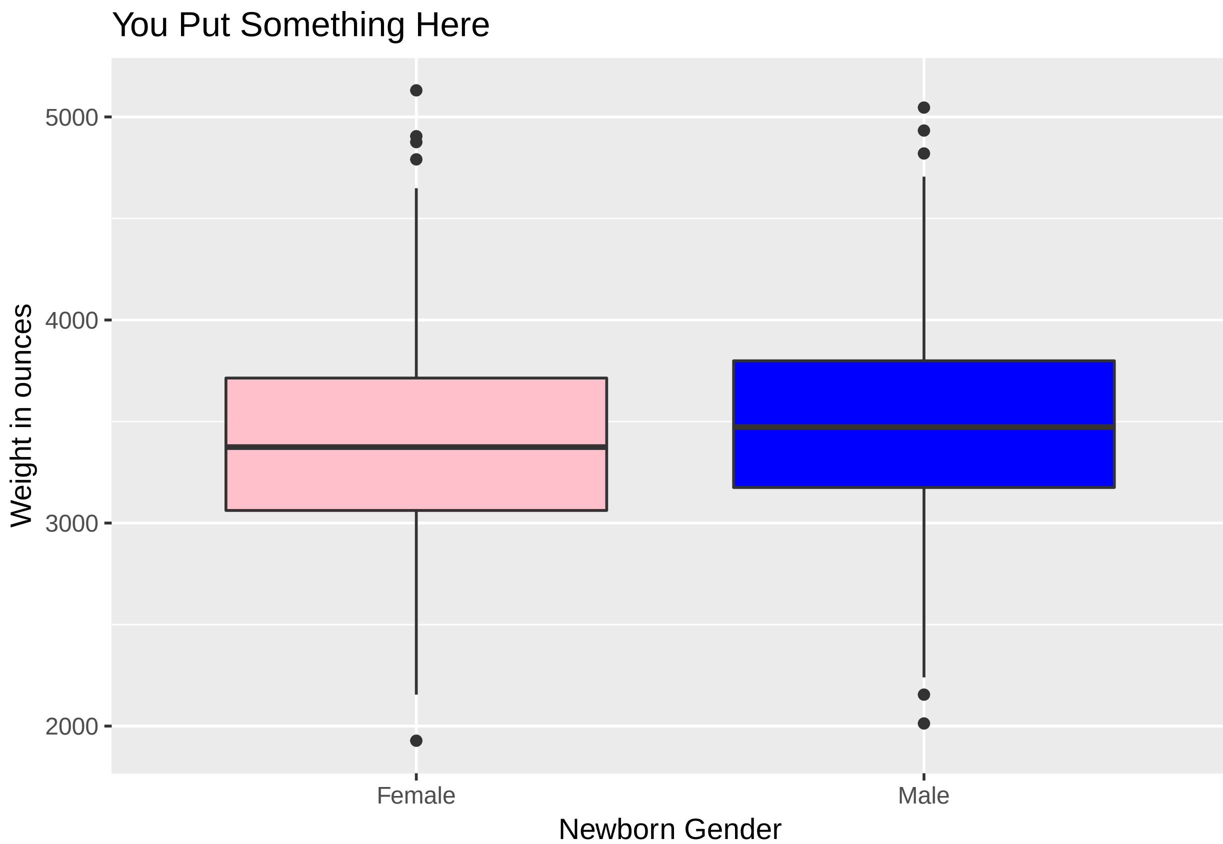

```r
p <- ggplot(data = NCBirths, aes(x = Gender, y = Weight, fill = Gender))
p + geom_boxplot()
```



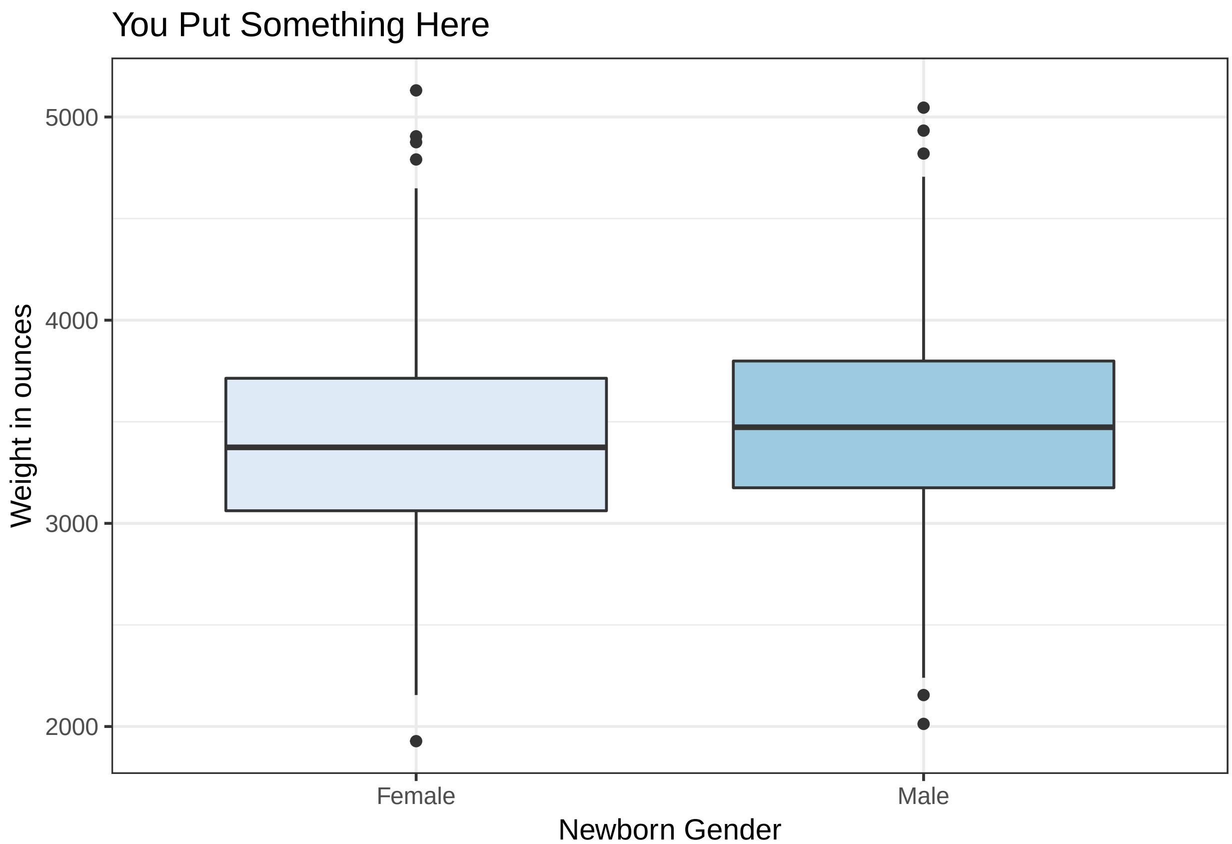

```r
p + geom_boxplot() +
  guides(fill = FALSE) +
  labs( x = "Newborn Gender", y = "Weight in ounces", title = "You Put Something Here")
```




```
p + geom_boxplot() +
  guides(fill = FALSE) +
  labs( x = "Newborn Gender", y = "Weight in ounces", title = "You Put Something Here") +
  scale_fill_manual(values = c('pink', 'blue'))
```

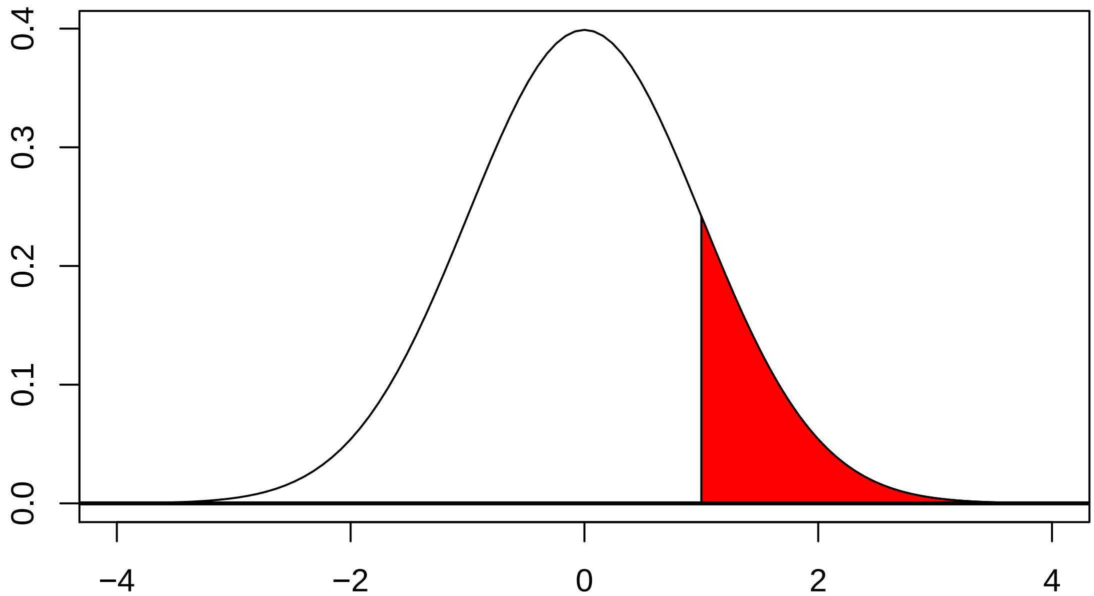



```r
p + geom_boxplot() + guides(fill = FALSE) +
  labs( x = "Newborn Gender", y = "Weight in ounces", title = "You Put Something Here") +
  scale_fill_brewer() + theme_bw()
```

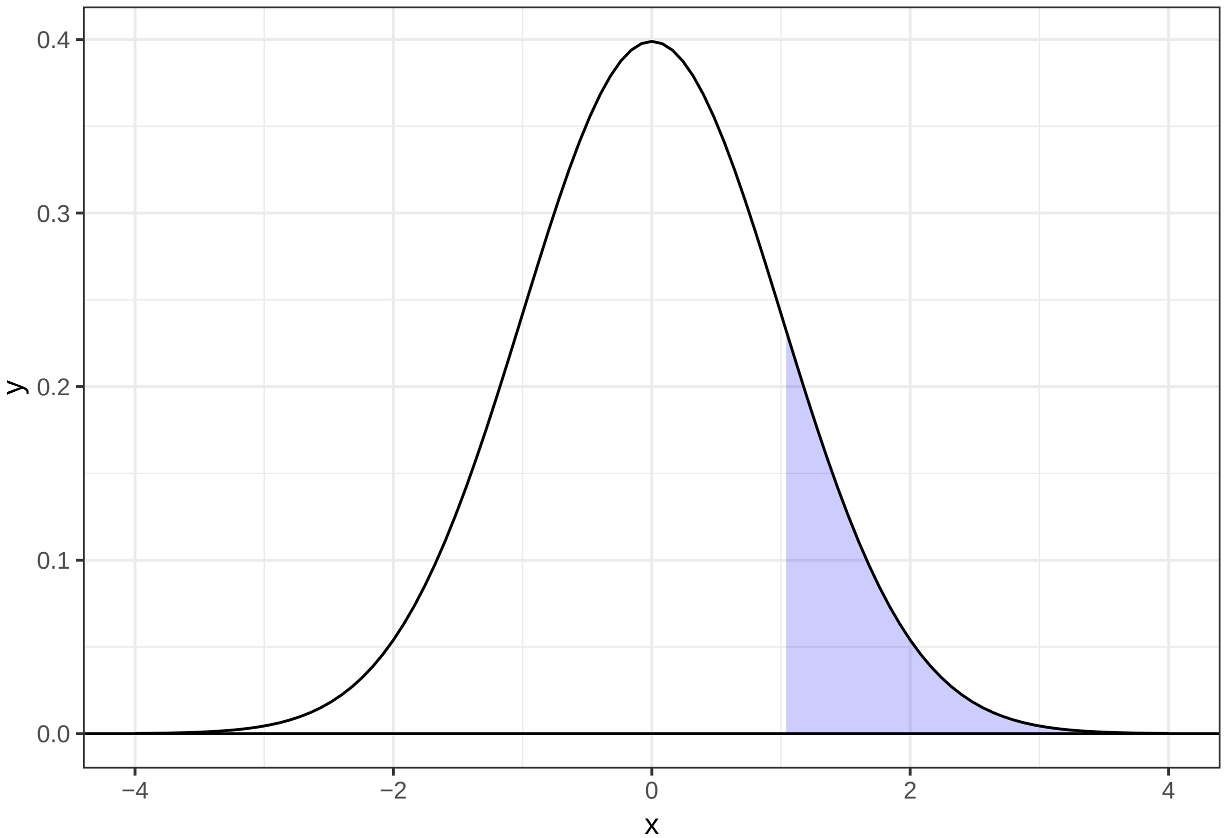


Density Plots

```
curve(dnorm(x), -4, 4, ylab = "", xlab = "")
x.region <- seq(from = 1, to = 4, length.out = 200)
y.region <- dnorm(x.region)
region.x <- c(x.region[1], x.region, x.region[200])
region.y <- c(0, y.region, 0)
polygon(region.x, region.y, col = "red")
abline(h = 0, lwd = 2)
```

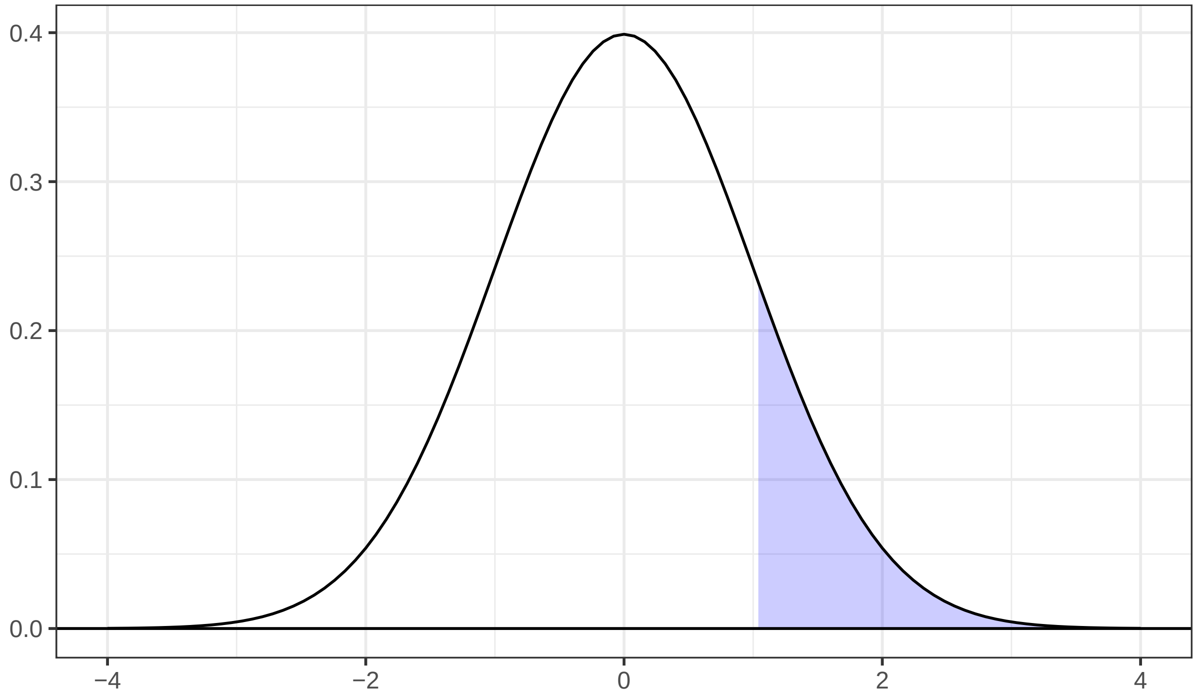



```r
# Same now with ggplot2
p <- ggplot(data = data.frame(x = c(-4, 4)), aes(x = x))
dnorm_func <- function(x){
  y <- dnorm(x)
  y[x<1] <- NA
  return(y)
}
p1 <- p + stat_function(fun = dnorm_func, geom = 'area', fill = 'blue', alpha = 0.2) +
  geom_hline(yintercept = 0) +
  stat_function(fun = dnorm)
p1 + theme_bw()
```




```
p1 + theme_bw() +
  labs(x = '', y = '', title = expression(integral(frac(1, sqrt(2*pi))*e^{-x^2/2}*dx, 1, infinity)==0.15
```

$$\int_{1}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = 0.1586553$$



Example 2.11

Note this is not how `qqnorm` computes the quantiles! The left graph of Figure 2.9 in the book is not quite correct…it does not use the data in the table…the first value 17.7 should be 21.7.

```
x <- c(21.7, 22.6, 26.1, 28.3, 30.0, 31.2, 31.5, 33.5, 34.7, 36.0)
n <- length(x)
p <- (1:10)/(n + 1)
q <- qnorm(p)
rbind(x, p, q)
```

```
##           [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
## x 21.70000000 22.6000000 26.1000000 28.3000000 30.0000000 31.2000000
## p  0.09090909  0.1818182  0.2727273  0.3636364  0.4545455  0.5454545
## q -1.33517774 -0.9084579 -0.6045853 -0.3487557 -0.1141853  0.1141853
##          [,7]       [,8]       [,9]      [,10]
## x 31.5000000 33.5000000 34.7000000 36.0000000
## p  0.6363636  0.7272727  0.8181818  0.9090909
## q  0.3487557  0.6045853  0.9084579  1.3351777
```

```
plot(q, x)
XS <- quantile(q, prob = c(.25, .75))
YS <- quantile(x, prob = c(.25, .75))
slopeA <- (YS[2] - YS[1])/(XS[2] - XS[1])
slopeB <- diff(YS)/diff(XS)
slopeA
```

```
##    75%
```

```
## 5.8728
```

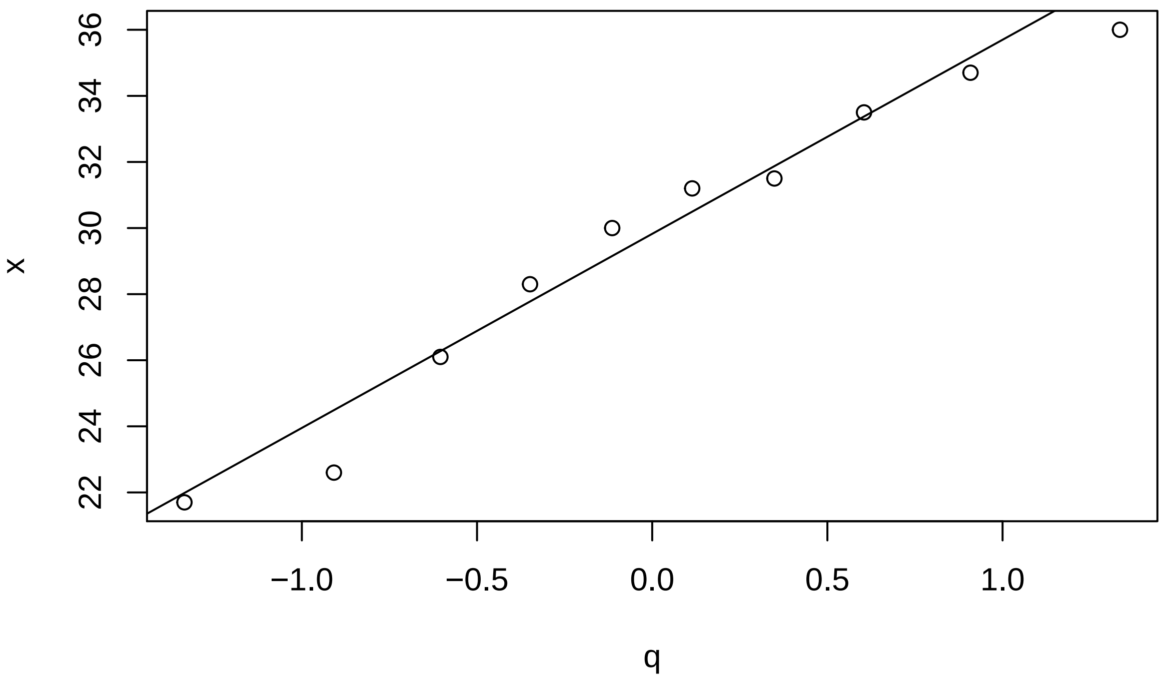
```r
slopeB
```

```
##    75%
## 5.8728
```

```r
Intercept <- YS[1] - slopeA*XS[1]
Intercept
```

```
##    25%
## 29.825
```

```r
abline(a = Intercept, b = slopeA)
```




```r
#
pc <- (1:10 - 3/8)/n
qc <- qnorm(pc)
rbind(x, pc, qc)
```

```
##         [,1]      [,2]      [,3]       [,4]        [,5]       [,6]
## x  21.700000 22.600000 26.100000 28.3000000 30.00000000 31.2000000
## pc  0.062500  0.162500  0.262500  0.3625000  0.46250000  0.5625000
## qc -1.534121 -0.984235 -0.635657 -0.3517843 -0.09413741  0.1573107
##          [,7]       [,8]     [,9]     [,10]
## x  31.5000000 33.5000000 34.70000 36.000000
## pc  0.6625000  0.7625000  0.86250  0.962500
## qc  0.4192958  0.7143674  1.09162  1.780464
```

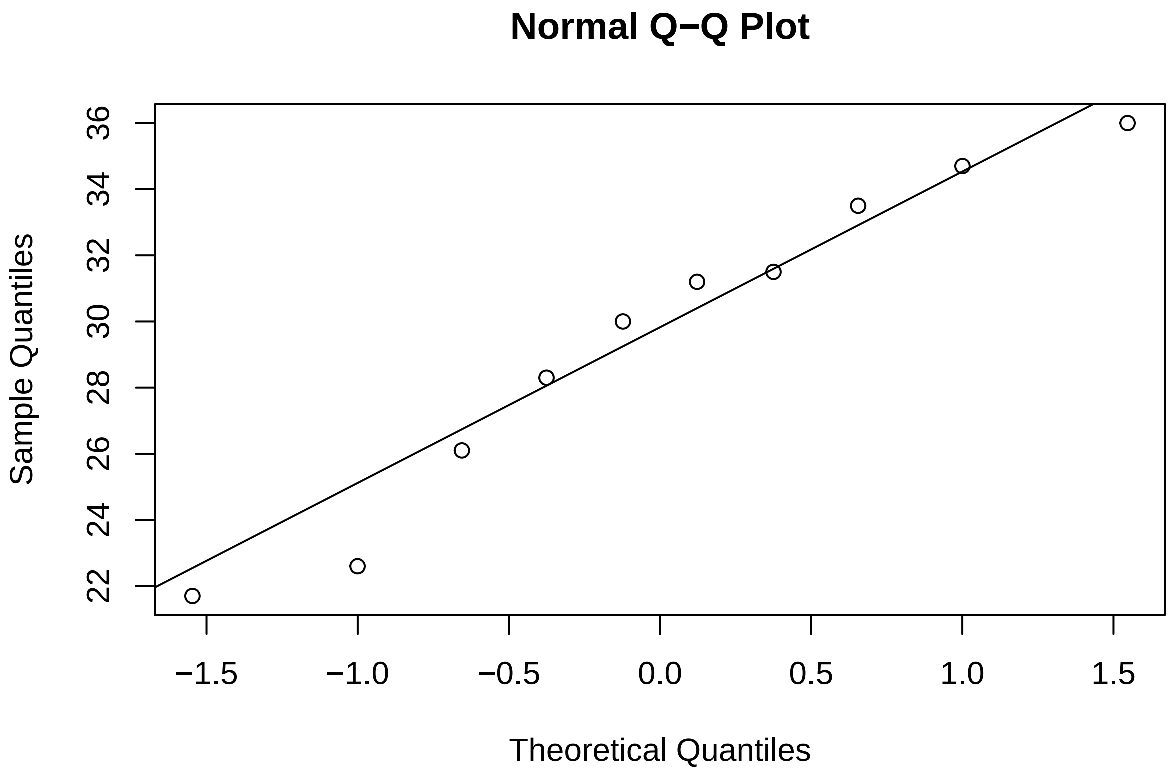
```r
xs <- quantile(qc, prob = c(.25, .75))
ys <- quantile(x, prob = c(.25, .75))
slope <- diff(ys)/diff(xs)
intercept <- ys[1] - slope*xs[1]
c(intercept, slope)
```

```
##       25%       75%
```

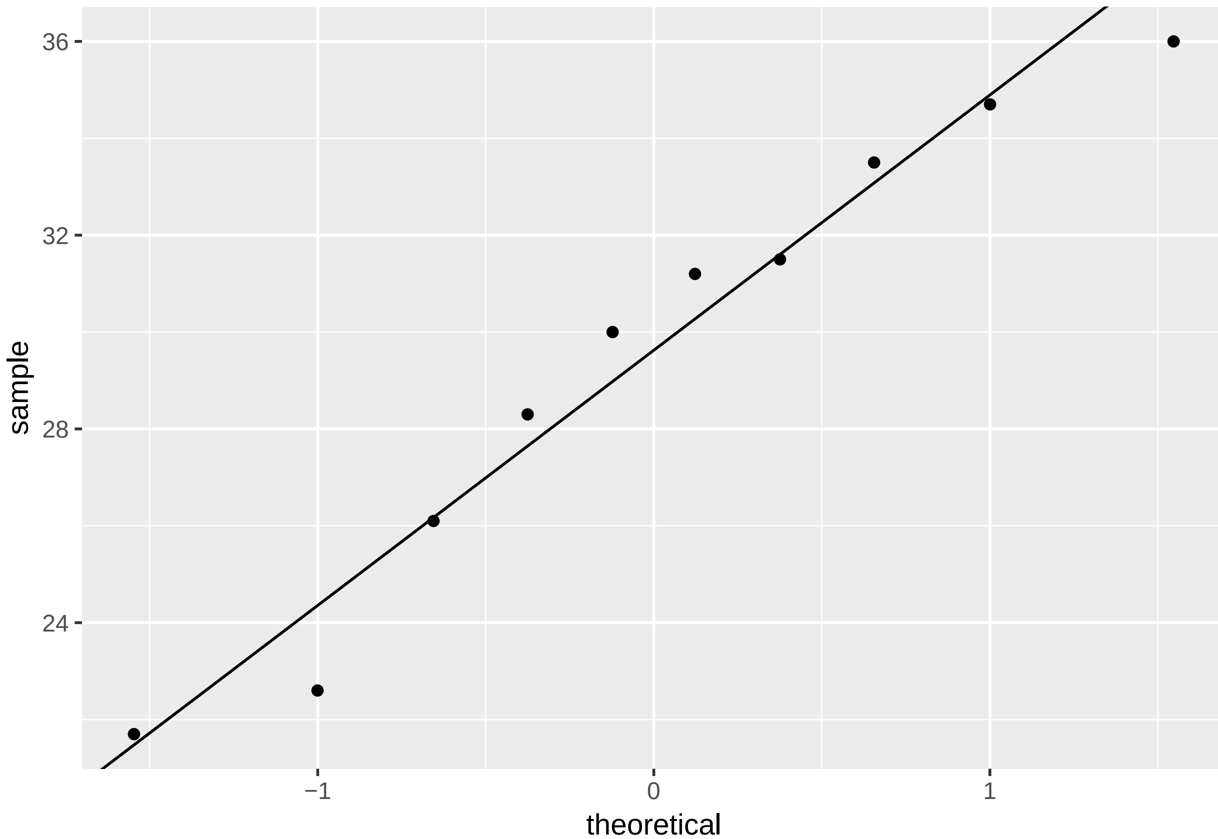
```
## 29.625034  5.268449
```

Consider using the R functions `qqnorm()` and `qqline()`.

```r
qqnorm(x)
qqline(x)
```




```r
# ggplot
ggplot(data = data.frame(x), aes(sample=x)) +
  stat_qq() +
  geom_abline(intercept = intercept, slope = slope)
```



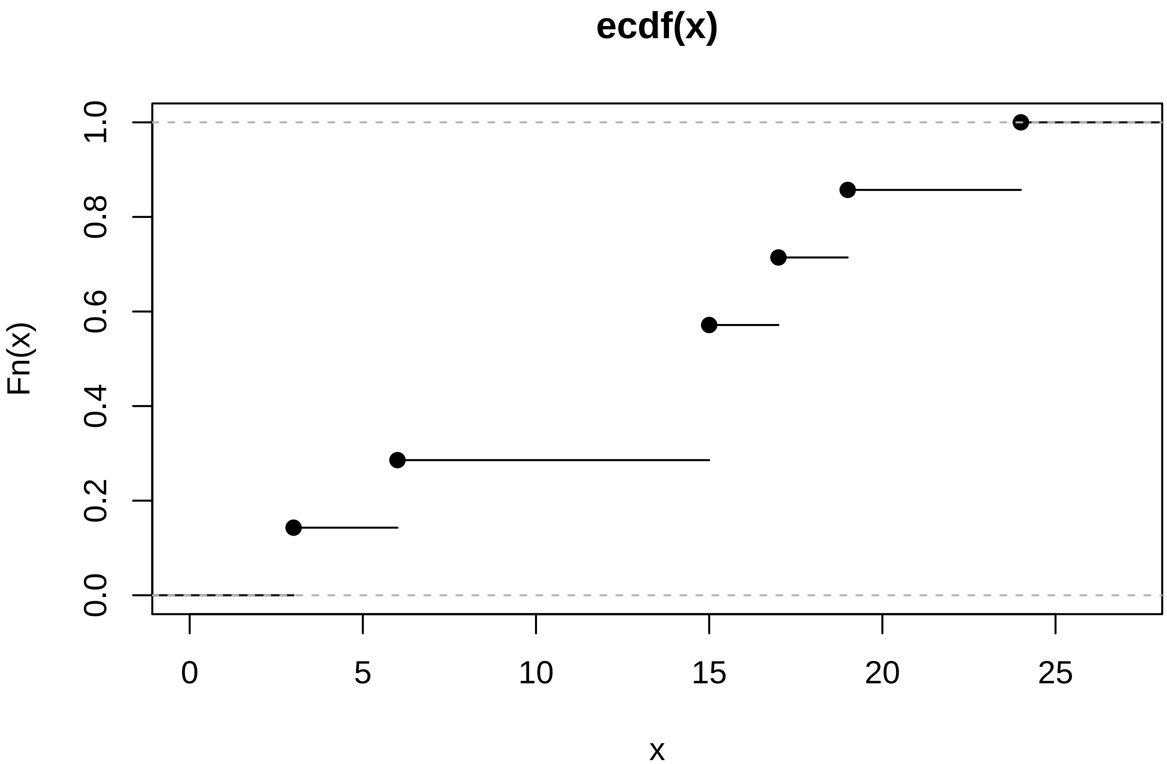
Empirical Cumulative Distribution Function

The *empirical cumulative distribution function* (ecdf) is an estimate of the underlying cumulative distribution function for a sample. The empirical cdf, denoted by $\hat{F}$, is a step function

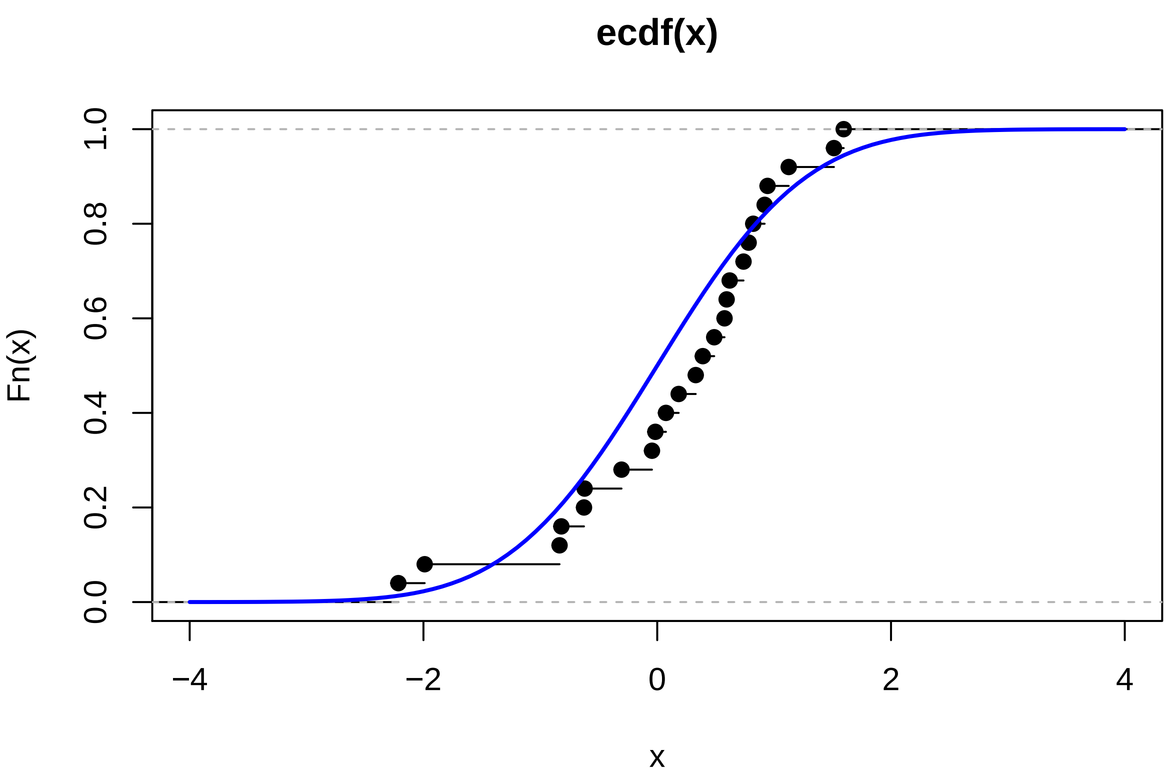
$$\hat{F}(x) = \frac{1}{n}(\text{number of values} \le x),$$

where n is the sample size.

```
y <- c(3, 6, 15, 15, 17, 19, 24)
plot.ecdf(y)
```




```r
set.seed(1)  # set seed for reproducibility
rxs <- rnorm(25)
plot.ecdf(rxs, xlim = c(-4, 4))
curve(pnorm(x), col = "blue", add = TRUE, lwd = 2)
```



An alternative approach to the book's Figure 2.12 is provided using `ggplot2` after first creating Figure 2.12

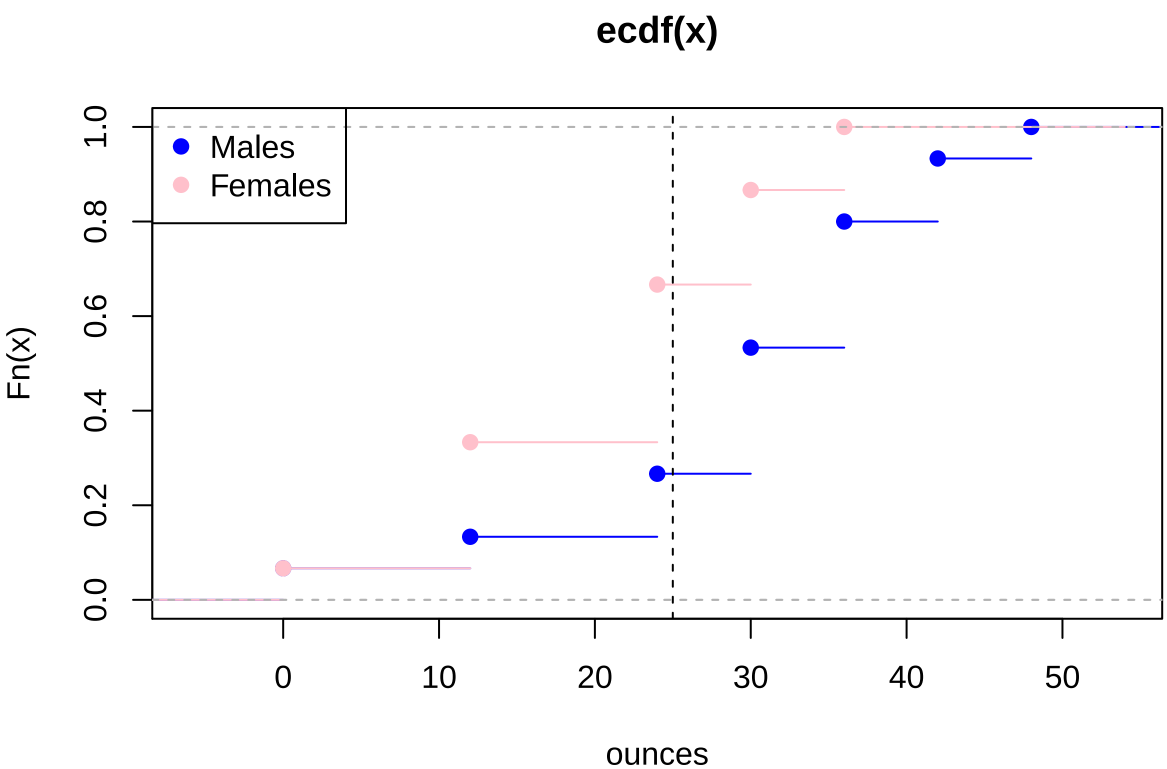
```r
site <- "https://raw.githubusercontent.com/alanarnholt/STT3850/gh-pages/DataCSV/Beerwings.csv"
Beerwings <- read.csv(file=url(site))
head(Beerwings) # shows first 6 rows of data frame
```

```
##   ID Hotwings Beer Gender
## 1  1        4   24      F
## 2  2        5    0      F
## 3  3        5   12      F
## 4  4        6   12      F
## 5  5        7   12      F
## 6  6        7   12      F
```

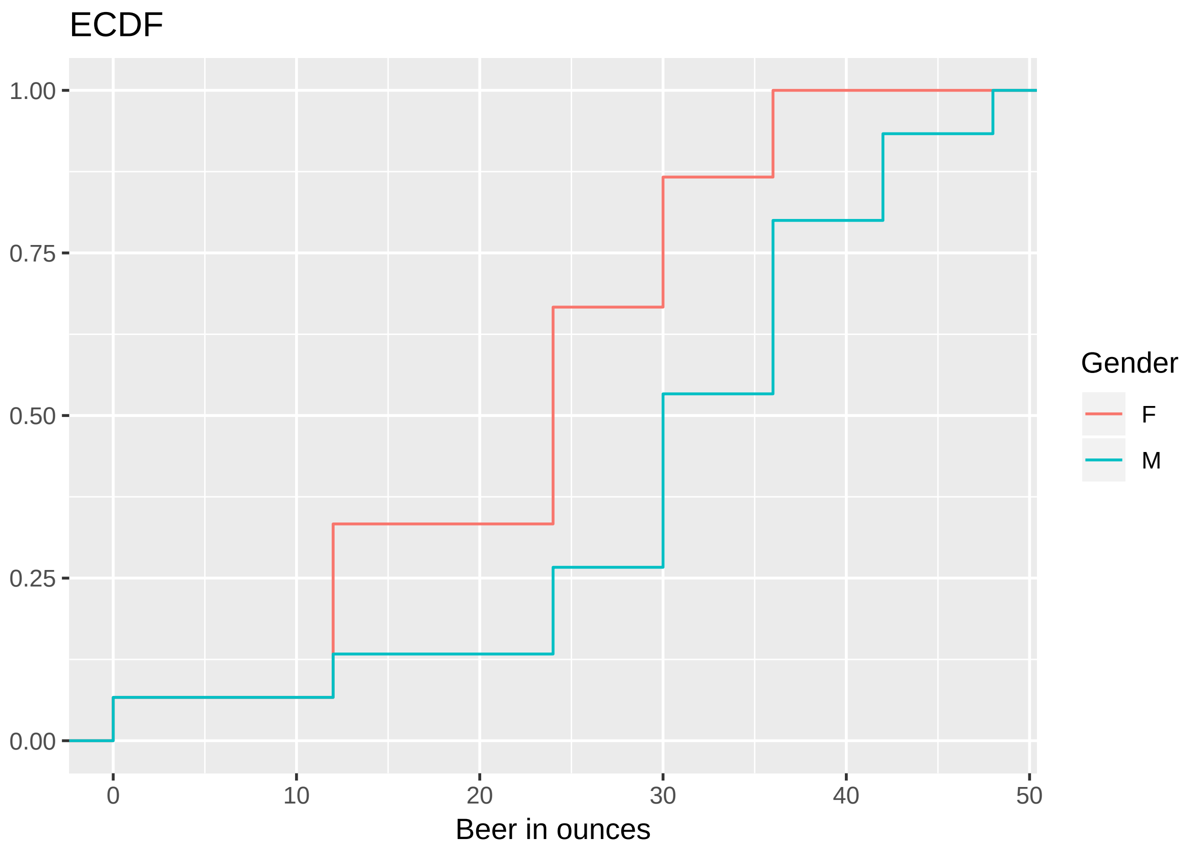
```r
str(Beerwings)  # shows structure of data frame
```

```
## 'data.frame':    30 obs. of  4 variables:
##  $ ID      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Hotwings: int  4 5 5 6 7 7 7 8 8 8 ...
##  $ Beer    : int  24 0 12 12 12 12 24 24 0 12 ...
##  $ Gender  : Factor w/ 2 levels "F","M": 1 1 1 1 1 1 2 1 2 2 ...
```

```r
beerM <- subset(Beerwings, select = Beer, subset = Gender == "M", drop = TRUE)
beerF <- subset(Beerwings, select = Beer, subset = Gender == "F", drop = TRUE)
plot.ecdf(beerM, xlab = "ounces", col = "blue", pch = 19)
plot.ecdf(beerF, col = "pink", pch = 19, add = TRUE)
abline(v = 25, lty = "dashed")
legend("topleft", legend = c("Males", "Females"), pch = 19, col = c("blue", "pink"))
```



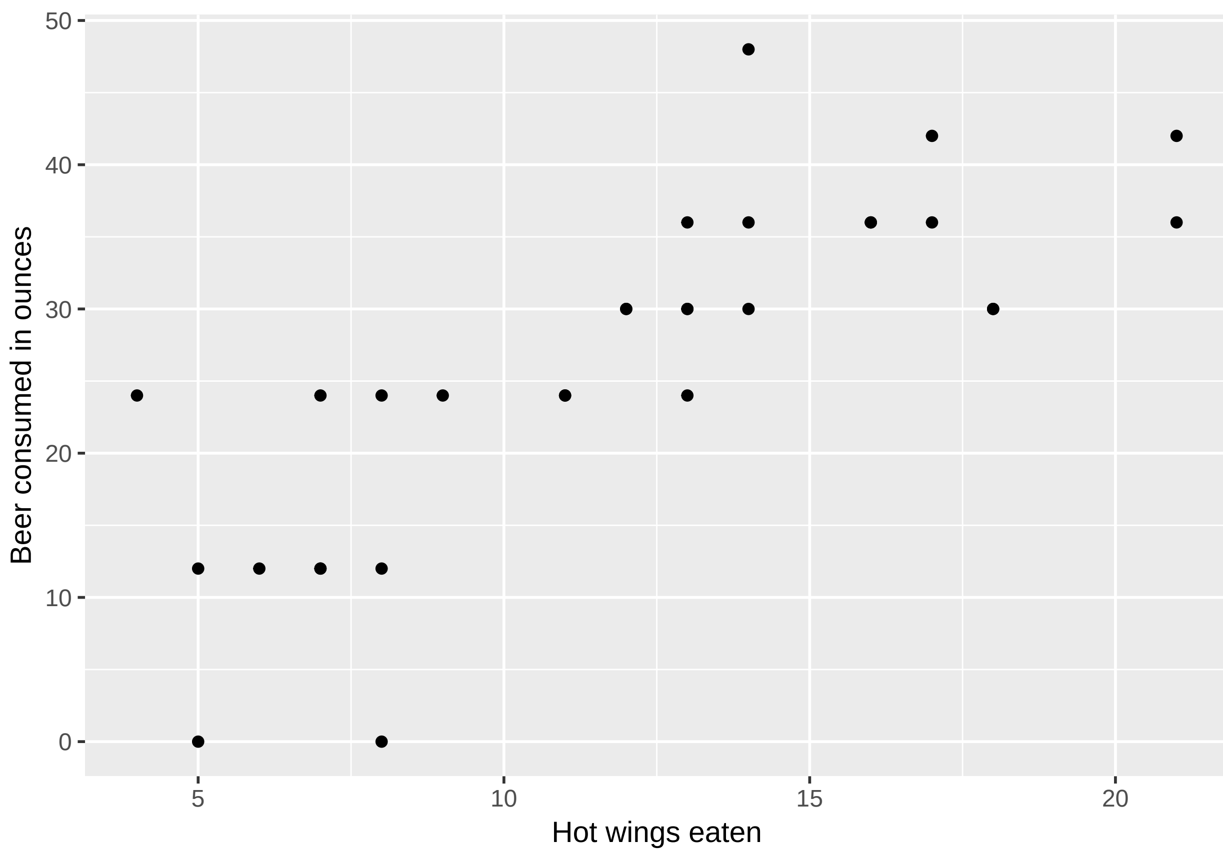

```r
# Using ggplot2 now
ggplot(data = Beerwings, aes(x = Beer, colour = Gender)) +
  stat_ecdf() +
  labs(x = "Beer in ounces", y ="", title = 'ECDF')
```



Scatter Plots

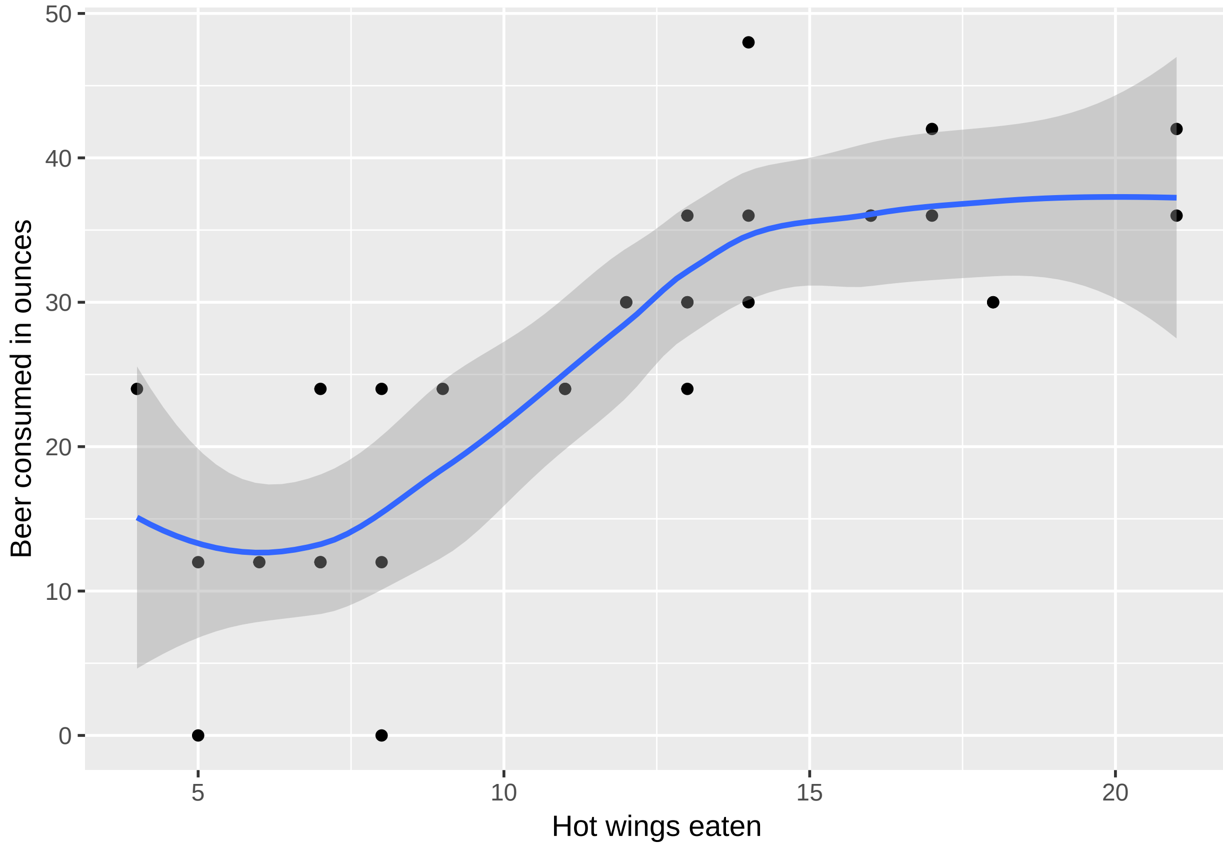
```
with(data = Beerwings, plot(Hotwings, Beer, xlab = "Hot wings eaten", ylab = "Beer consumed",
                              pch = 19, col = "blue"))
```


```
p <- ggplot(data = Beerwings, aes(x = Hotwings, y = Beer)) +
  geom_point() +
  labs(x = "Hot wings eaten", y = "Beer consumed in ounces")
p
```

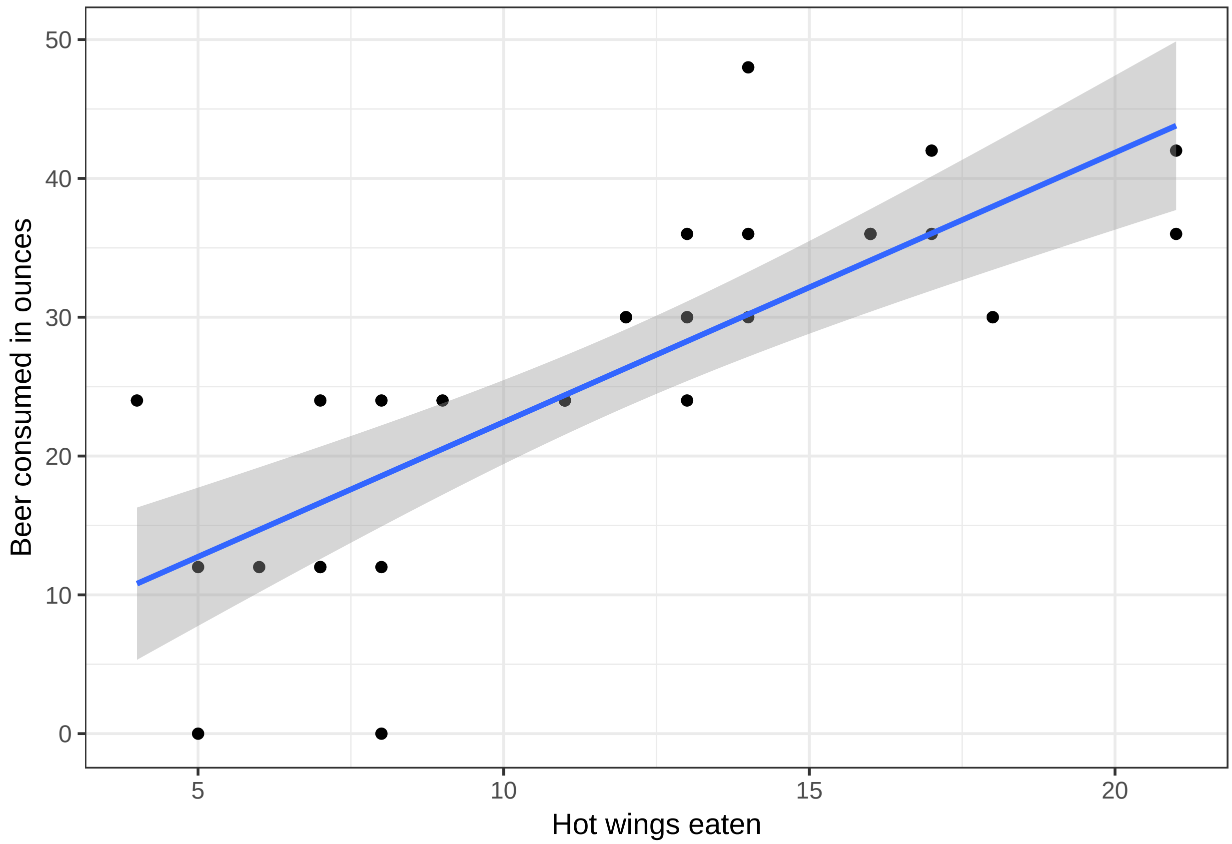


```r
p + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```r
p + geom_smooth(method = lm) + theme_bw()
```



Integrating with R

```
f <- function(x){(x - 1)^3*exp(-x)}
ans <- integrate(f, lower = 0, upper = Inf)$value
ans
```

```
## [1] 2
```