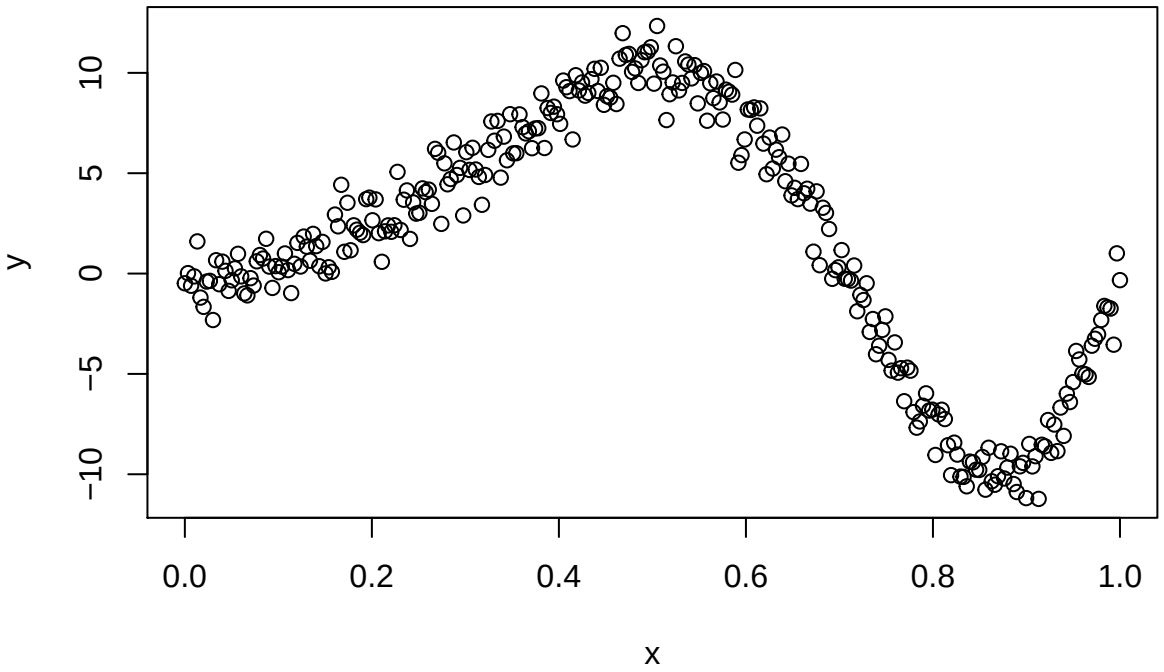
Introduction to Splines

Andy Shen, Devin Francom

Let's say you want to fit a model using some wiggly data. Maybe

```
n<-300
x<-seq(0,1,length.out=n)
y<-sin(2*pi*x^2)*10+rnorm(n)
plot(x,y)
```



One way to fit a model to data like this is to come up with a linear basis and fit a linear model using the basis as the X matrix (which we will call B). People often use splines as a basis. The simplest set of spline basis functions would be to make the ith basis function (i.e., the ith column of B) look like

$$B_{ij} = [s_i(x_j - t_i)]_+$$

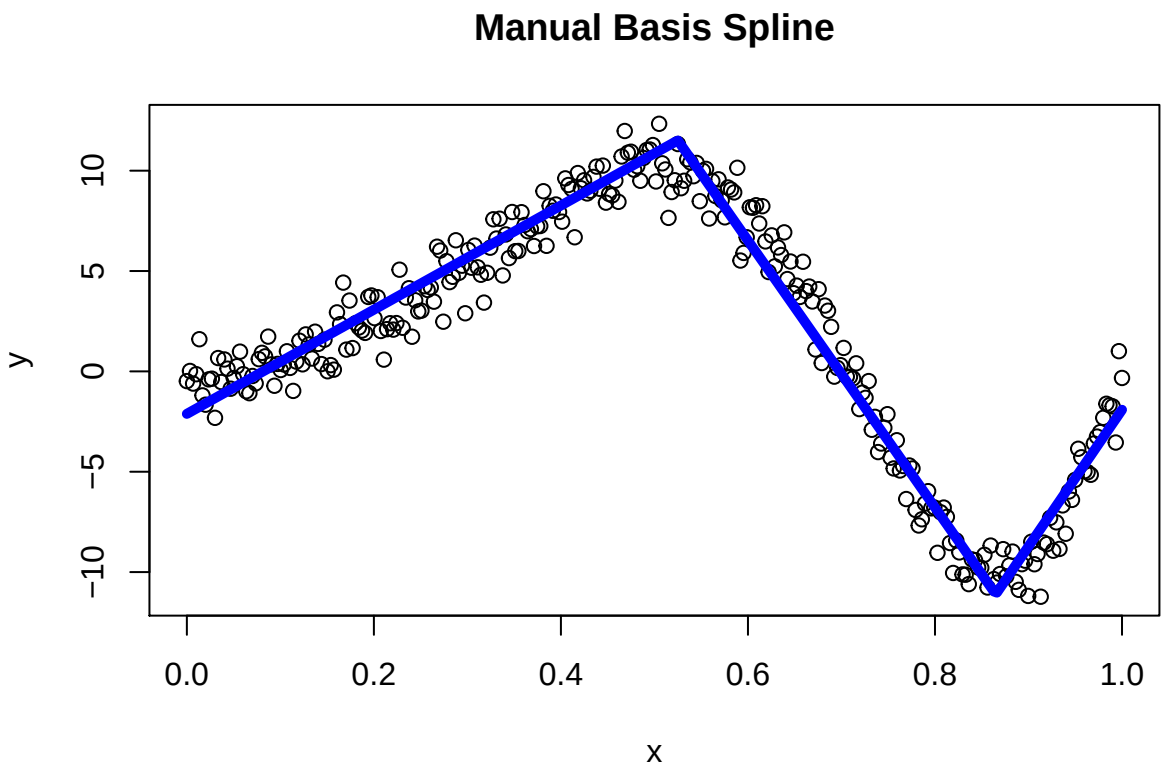
where $s \in \{-1, 1\}$, which we'll call the sign, and t is a value in the domain of x, which we will call a knot. Also, $[a]_+ = max(0, a)$.

Try some combinations of s and t to see what your basis functions look like, and what the corresponding linear model fit looks like (using the lm function or your Bayesian linear model code). Try with different numbers of basis functions, also.

Manual Spline Function

```
generate_spline <- function(tvec, s = 1, y, x, nknot = length(tvec)) {
  Bmat <- matrix(NA, nknot, length(x))
  hs <- Bmat
  for(i in 1:nknot) {
    for(j in 1:length(x)) {
      Bmat[i,j] <- max(s * (x[j] - tvec[i]), 0)
    }
  }

  mBmat <- t(Bmat)

  mod <- lm(y ~ x + mBmat)
  pred <- predict(mod)
  sq <- x

  for(ii in 1:nknot) {
    hs[ii,] <- sq - tvec[ii]
    hs[ii,][sq < tvec[ii]] <- 0
  }

  plot(x,y, main = "Manual Basis Spline")
  lines(x, pred, type = "l", lwd = 5, col="blue1")
}
```

```
tv <- c(0.525, 0.865)
generate_spline(tv, y = y, x = x)
```

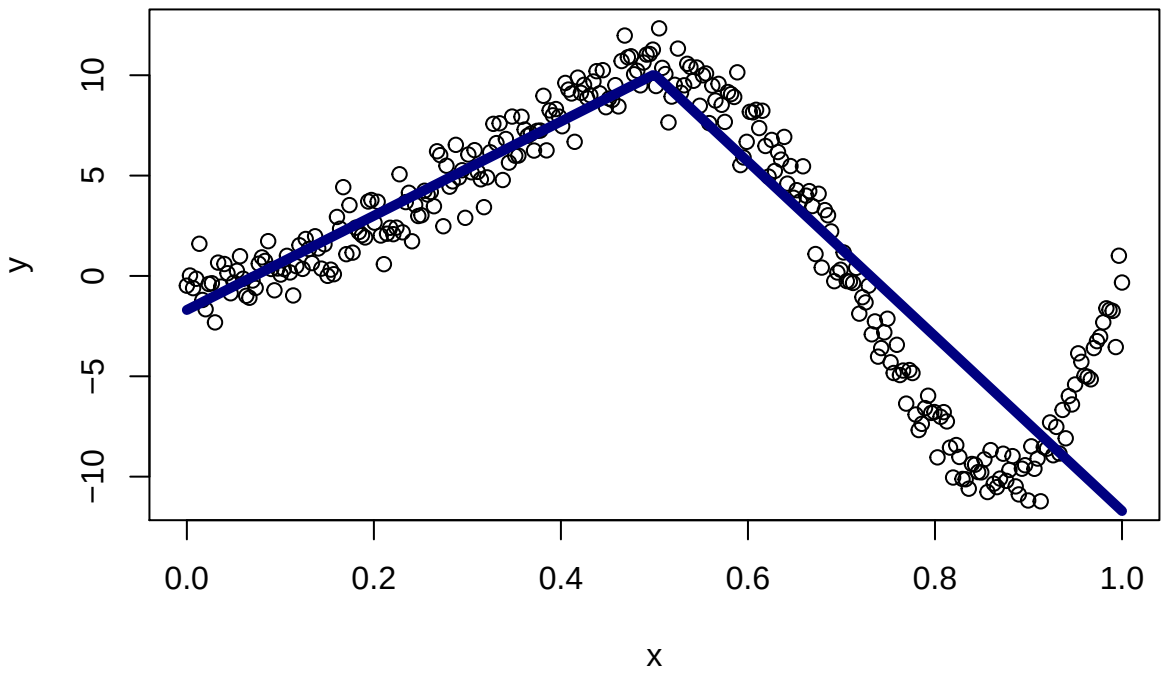


Trying things out

```
t1 <- 0.5 #knot at 0.5
s <- 1
B1 <- rep(NA, length(x))

for(i in 1:length(x)) {
  B1[i] <- max(s * (x[i] - t1), 0)
}
mod <- lm(y ~ x + B1)
summary(mod)
```

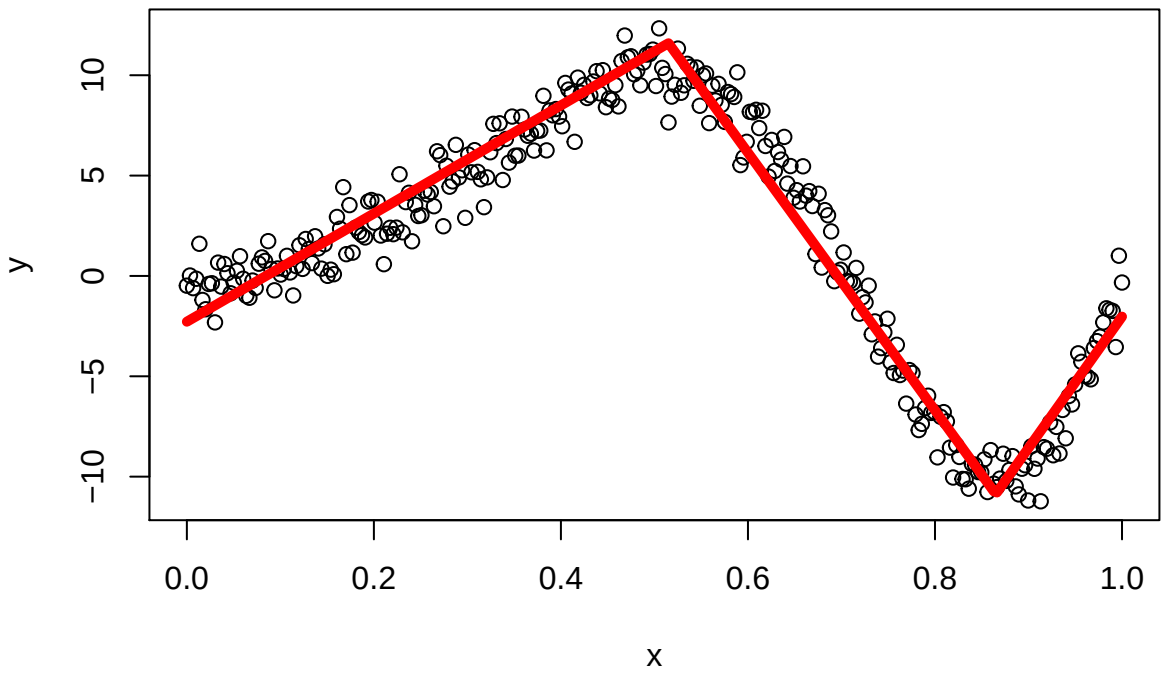
```
##
## Call:
## lm(formula = y ~ x + B1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1910 -1.3288  0.0851  1.2505 12.5727
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.6949     0.4212  -4.024 7.28e-05 ***
## x            23.4813     1.2345  19.022  < 2e-16 ***
## B1          -66.9931     2.2083 -30.337  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.77 on 297 degrees of freedom
## Multiple R-squared:  0.808,  Adjusted R-squared:  0.8067
## F-statistic: 624.7 on 2 and 297 DF,  p-value: < 2.2e-16
```

```
cf <- mod$coefficients
sq <- x
hs <- sq - t1
hs[sq < t1] <- 0
yfit <- cf[1] + cf[2]*x + cf[3]*hs

plot(x,y, main = "Manual Basis Spline")
lines(x, yfit, type = "l", lwd = 5, col="navy")
```



Add another knot

```
t1 <- 0.515
t2 <- 0.865
s <- 1
B1 <- rep(NA, length(x))
B2 <- B1

for(i in 1:length(x)) {
  B1[i] <- max(s * (x[i] - t1), 0)
  B2[i] <- max(s * (x[i] - t2), 0)
}
mod <- lm(y ~ x + B1 + B2)

cf <- mod$coefficients
sq <- x
hs1 <- sq - t1
hs1[sq < t1] <- 0

hs2 <- sq - t2
hs2[sq < t2] <- 0

yfit <- cf[1] + cf[2]*x + cf[3]*hs1 + cf[4]*hs2
yfit2 <- predict(mod) #same thing

plot(x,y, main = "Manual Basis Spline")
lines(x, yfit2, type = "l", lwd = 5, col="red")
```



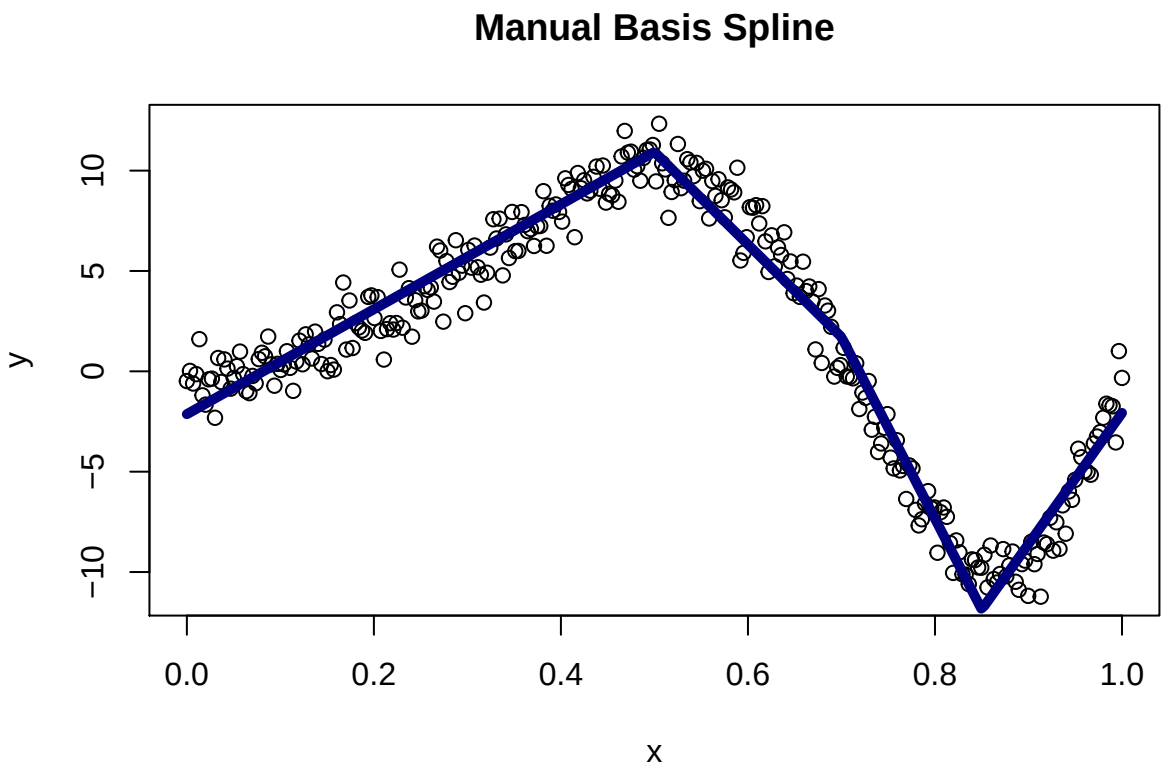
Add another knot:

```
t1 <- 0.5 #knot at 0.5
t2 <- 0.85 #another knot at 0.85
t3 <- 0.7
s <- 1
B1 <- rep(NA, length(x))
B3 <- B2 <- B1

for(i in 1:length(x)) {
  B1[i] <- max(s * (x[i] - t1), 0)
  B2[i] <- max(s * (x[i] - t2), 0)
  B3[i] <- max(s * (x[i] - t3), 0)
}
mod <- lm(y ~ x + B1 + B2 + B3)
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x + B1 + B2 + B3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4754 -0.8950  0.0041  0.8281  3.3876
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.1344     0.1927  -11.08   <2e-16 ***
## x            26.1359     0.6095   42.88   <2e-16 ***
```

```
## B1          -72.2697     1.9538  -36.99   <2e-16 ***
## B2          155.8006     5.1595   30.20   <2e-16 ***
## B3          -44.3651     3.6692  -12.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.234 on 295 degrees of freedom
## Multiple R-squared:  0.9621, Adjusted R-squared:  0.9616
## F-statistic:  1873 on 4 and 295 DF,  p-value: < 2.2e-16
```

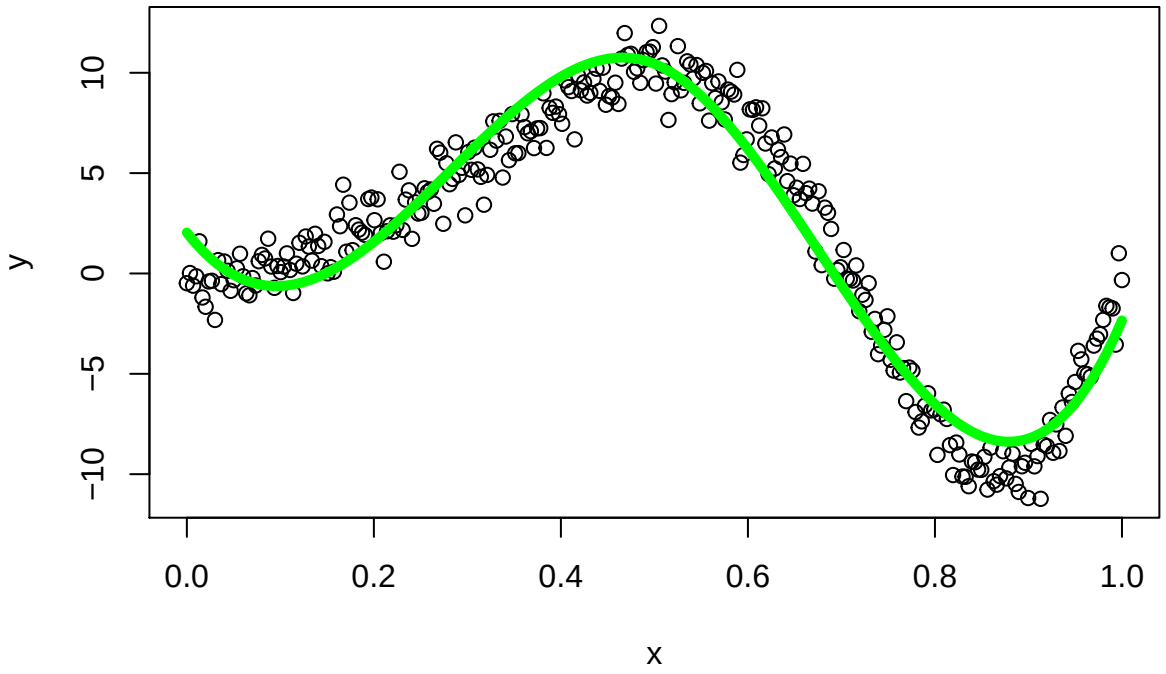
```r
cf <- mod$coefficients
sq <- x

hs1 <- sq - t1
hs1[sq < t1] <- 0

hs2 <- sq - t2
hs2[sq < t2] <- 0

hs3 <- sq - t3
hs3[sq < t3] <- 0

yfit2 <- cf[1] + cf[2]*x + cf[3]*hs1 + cf[4]*hs2 + cf[5]*hs3
yfit <- predict(mod)
plot(x,y, main = "Manual Basis Spline")
lines(x, yfit, type = "l", lwd = 5, col="navy")
```



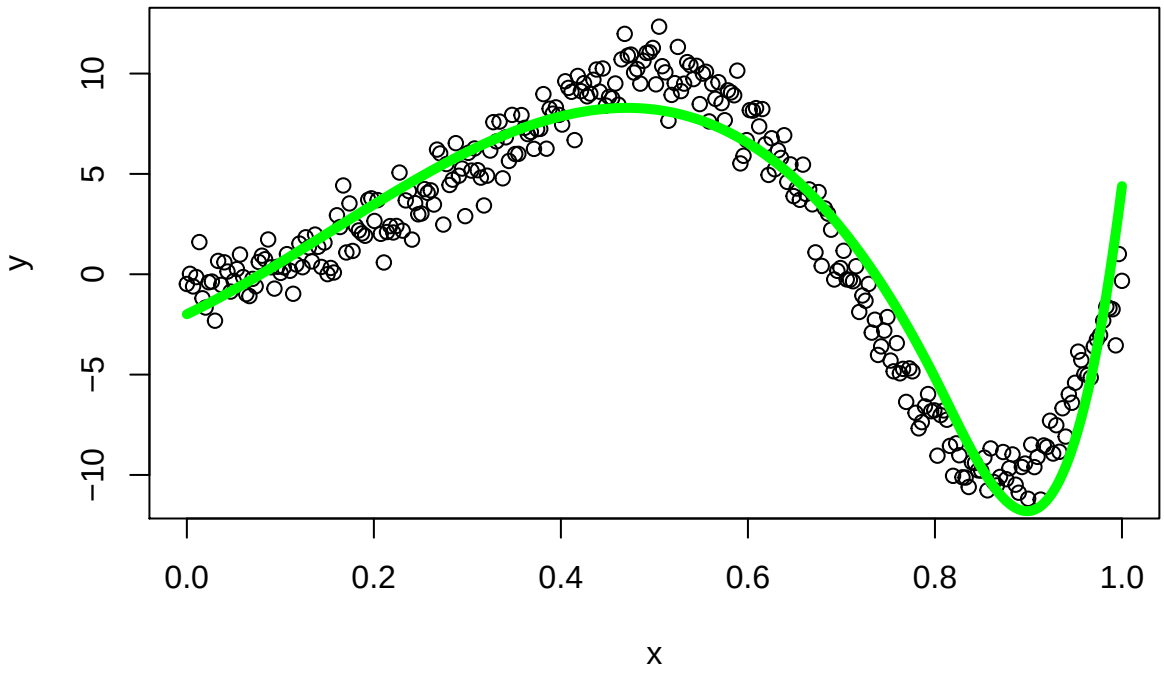
Using the `bs()` Function

1 Knot

```
library(splines)
df <- data.frame(y, x)
m2 <- lm(y ~ bs(x, knots = 0.5), data = df)
pred <- predict(m2)

plot(x,y)
lines(x, pred, lwd = 5, col = "green")
```

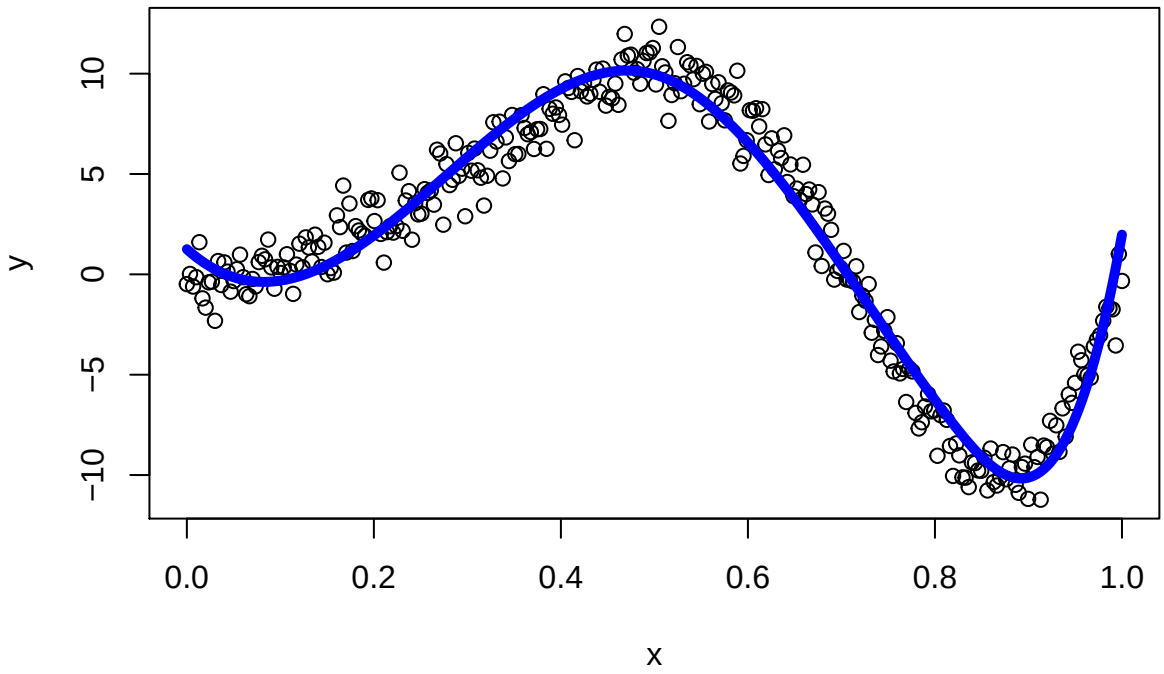



```
m2 <- lm(y ~ bs(x, knots = 0.8), data = df)
pred <- predict(m2)

plot(x,y)
lines(x, pred, lwd = 5, col = "green")
```



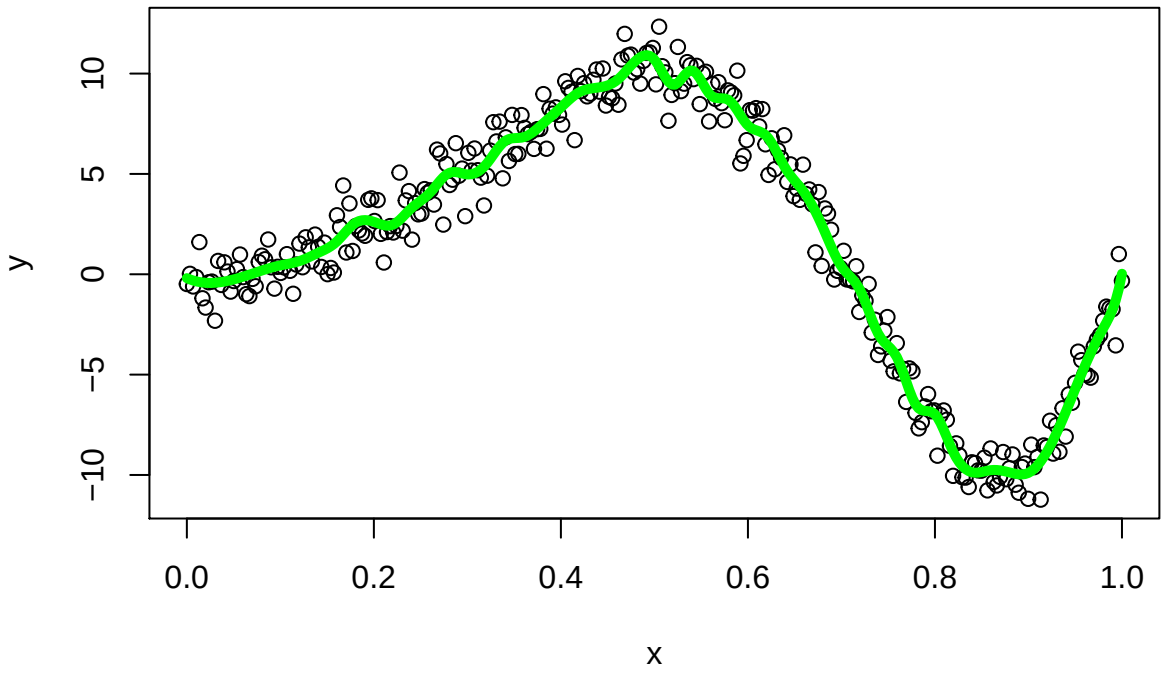
2 Knots (Expected)

```r
m1 <- lm(y ~ bs(x, knots = c(0.5, 0.82)), data = df)
pred <- predict(m1)

plot(x,y)
lines(x, pred, lwd = 5, col = "blue")
```



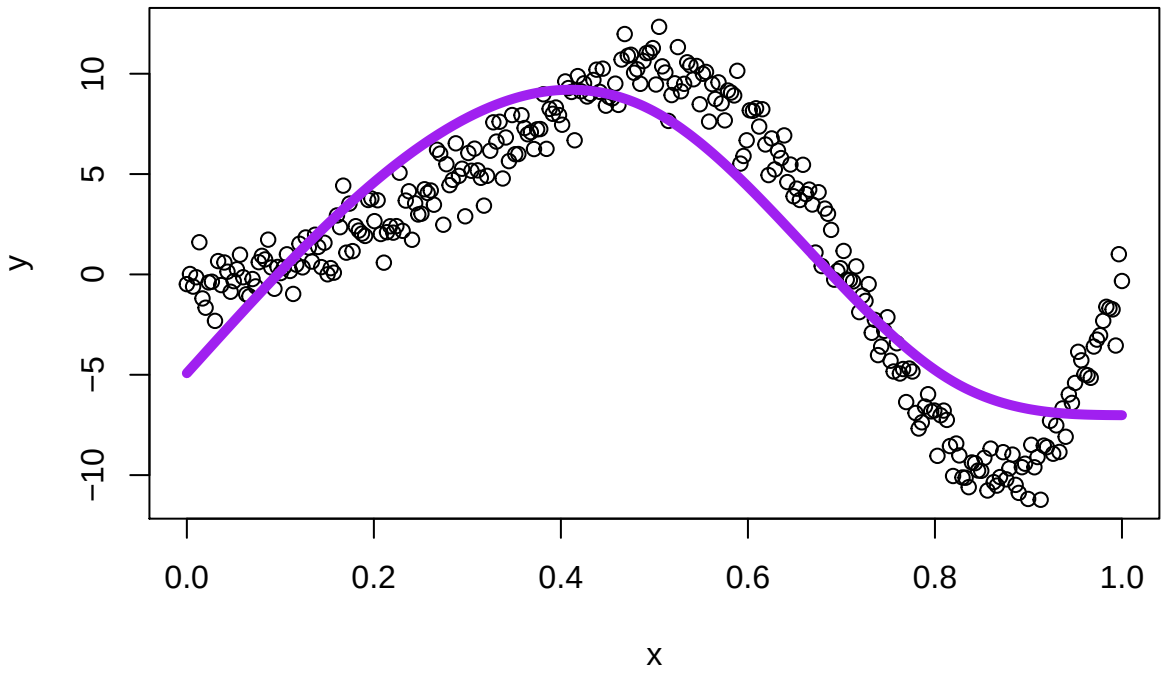
Too Many Knots

```
m2 <- lm(y ~ bs(x, knots = seq(0.1,1,by=0.02)), data = df)
pred <- predict(m2)

plot(x,y)
lines(x, pred, lwd = 5, col = "green")
```



Natural Splines

```
m3 <- lm(y ~ ns(x, knots = c(0.5, 0.82)), data = df)
pred <- predict(m3)

plot(x,y)
lines(x, pred, lwd = 5, col = "purple")
```




```
summary(m1)
```

```
##
## Call:
## lm(formula = y ~ bs(x, knots = c(0.5, 0.82)), data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8618 -0.8136 -0.0282  0.9196  3.5106
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.2535     0.3524   3.557 0.000437 ***
## bs(x, knots = c(0.5, 0.82))1  -6.9914     0.7558  -9.250  < 2e-16 ***
## bs(x, knots = c(0.5, 0.82))2  22.4016     0.5092  43.992  < 2e-16 ***
## bs(x, knots = c(0.5, 0.82))3  -7.8265     0.6272 -12.478  < 2e-16 ***
## bs(x, knots = c(0.5, 0.82))4 -14.8258     0.5276 -28.100  < 2e-16 ***
## bs(x, knots = c(0.5, 0.82))5   0.7245     0.6347   1.141 0.254625
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.271 on 294 degrees of freedom
## Multiple R-squared:   0.96,  Adjusted R-squared:  0.9593
## F-statistic:  1410 on 5 and 294 DF,  p-value: < 2.2e-16
```