MATH 498 HW2

Drew Remmenga

1.a.

```
library(splines)
HW2Test<- function(x){
pgamma( x, shape =10, scale =.02) - .5*x
}
xGrid <- seq( 0,1, length.out=200)
y<-HW2Test(xGrid)
fit1<-lm(y ~ poly(xGrid, 17, raw=TRUE))
lm(y ~ poly(xGrid, 17, raw=TRUE))
```

```
##
## Call:
## lm(formula = y ~ poly(xGrid, 17, raw = TRUE))
##
## Coefficients:
##                 (Intercept)   poly(xGrid, 17, raw = TRUE)1
##                   7.864e-03                     -3.314e+00
##  poly(xGrid, 17, raw = TRUE)2   poly(xGrid, 17, raw = TRUE)3
##                   1.851e+02                     -4.771e+03
##  poly(xGrid, 17, raw = TRUE)4   poly(xGrid, 17, raw = TRUE)5
##                   6.071e+04                     -4.262e+05
##  poly(xGrid, 17, raw = TRUE)6   poly(xGrid, 17, raw = TRUE)7
##                   1.855e+06                     -5.356e+06
##  poly(xGrid, 17, raw = TRUE)8   poly(xGrid, 17, raw = TRUE)9
##                   1.060e+07                     -1.447e+07
## poly(xGrid, 17, raw = TRUE)10  poly(xGrid, 17, raw = TRUE)11
##                   1.337e+07                     -7.820e+06
## poly(xGrid, 17, raw = TRUE)12  poly(xGrid, 17, raw = TRUE)13
##                   2.355e+06                             NA
## poly(xGrid, 17, raw = TRUE)14  poly(xGrid, 17, raw = TRUE)15
##                  -1.734e+05                             NA
## poly(xGrid, 17, raw = TRUE)16  poly(xGrid, 17, raw = TRUE)17
##                   1.297e+04                             NA
```

```
fit2<-lm(y ~ poly(xGrid, 15, raw=TRUE))
lm(y ~ poly(xGrid, 15, raw=TRUE))
```

```
##
## Call:
## lm(formula = y ~ poly(xGrid, 15, raw = TRUE))
##
## Coefficients:
```

```
##                 (Intercept)   poly(xGrid, 15, raw = TRUE)1
##                   4.063e-03                     -2.274e+00
##  poly(xGrid, 15, raw = TRUE)2   poly(xGrid, 15, raw = TRUE)3
##                   1.279e+02                     -3.457e+03
##  poly(xGrid, 15, raw = TRUE)4   poly(xGrid, 15, raw = TRUE)5
##                   4.458e+04                     -3.062e+05
##  poly(xGrid, 15, raw = TRUE)6   poly(xGrid, 15, raw = TRUE)7
##                   1.274e+06                     -3.451e+06
##  poly(xGrid, 15, raw = TRUE)8   poly(xGrid, 15, raw = TRUE)9
##                   6.292e+06                     -7.762e+06
## poly(xGrid, 15, raw = TRUE)10  poly(xGrid, 15, raw = TRUE)11
##                   6.324e+06                     -3.162e+06
## poly(xGrid, 15, raw = TRUE)12  poly(xGrid, 15, raw = TRUE)13
##                   7.798e+05                             NA
## poly(xGrid, 15, raw = TRUE)14  poly(xGrid, 15, raw = TRUE)15
##                  -2.989e+04                             NA
```

```r
fit3<-lm(y ~ poly(xGrid, 13, raw=TRUE))
lm(y ~ poly(xGrid, 13, raw=TRUE))
```

```
##
## Call:
## lm(formula = y ~ poly(xGrid, 13, raw = TRUE))
##
## Coefficients:
##                 (Intercept)   poly(xGrid, 13, raw = TRUE)1
##                  -8.951e-03                      7.375e-01
##  poly(xGrid, 13, raw = TRUE)2   poly(xGrid, 13, raw = TRUE)3
##                  -1.645e+01                     -5.663e+02
##  poly(xGrid, 13, raw = TRUE)4   poly(xGrid, 13, raw = TRUE)5
##                   1.361e+04                     -1.058e+05
##  poly(xGrid, 13, raw = TRUE)6   poly(xGrid, 13, raw = TRUE)7
##                   4.349e+05                     -1.091e+06
##  poly(xGrid, 13, raw = TRUE)8   poly(xGrid, 13, raw = TRUE)9
##                   1.759e+06                     -1.840e+06
## poly(xGrid, 13, raw = TRUE)10  poly(xGrid, 13, raw = TRUE)11
##                   1.208e+06                     -4.536e+05
## poly(xGrid, 13, raw = TRUE)12  poly(xGrid, 13, raw = TRUE)13
##                   7.435e+04                             NA
```

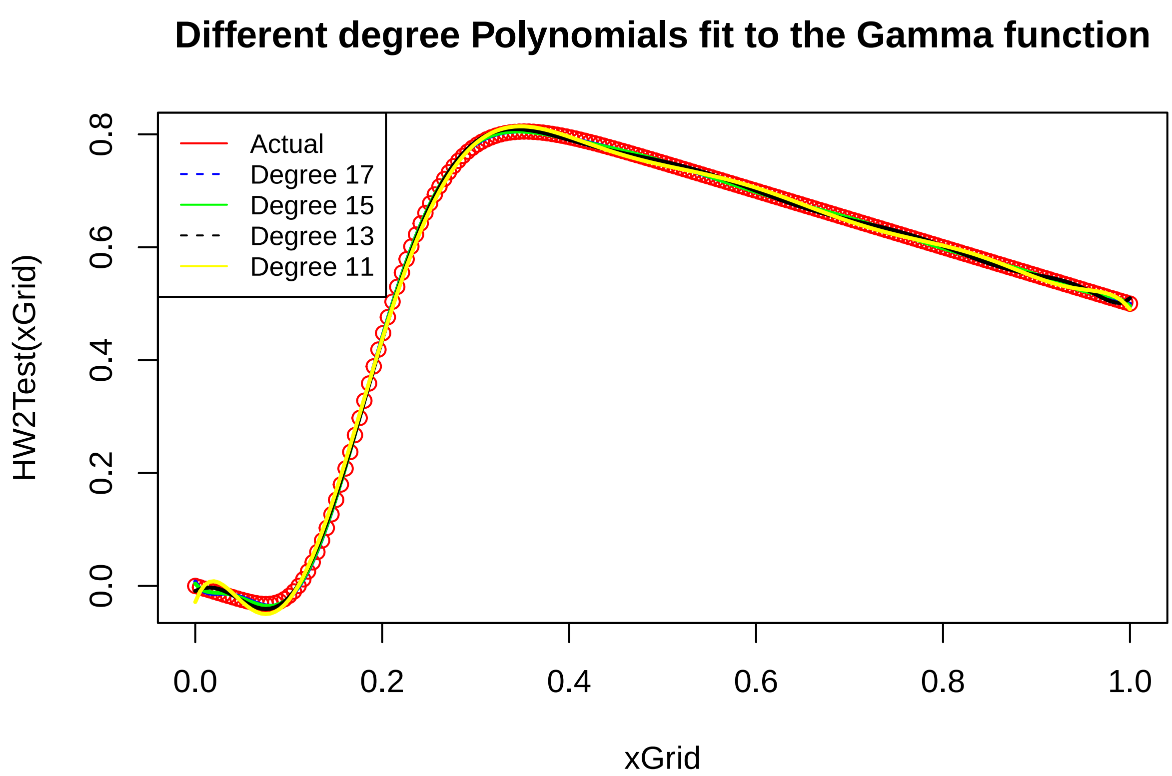
```r
fit4<-lm(y ~ poly(xGrid, 11, raw=TRUE))
lm(y ~ poly(xGrid, 11, raw=TRUE))
```

```
##
## Call:
## lm(formula = y ~ poly(xGrid, 11, raw = TRUE))
##
## Coefficients:
##                 (Intercept)   poly(xGrid, 11, raw = TRUE)1
##                  -2.856e-02                      4.507e+00
##  poly(xGrid, 11, raw = TRUE)2   poly(xGrid, 11, raw = TRUE)3
##                  -1.706e+02                      2.076e+03
##  poly(xGrid, 11, raw = TRUE)4   poly(xGrid, 11, raw = TRUE)5
```

```
##                      -1.054e+04                        2.688e+04
##  poly(xGrid, 11, raw = TRUE)6   poly(xGrid, 11, raw = TRUE)7
##                      -3.220e+04                        3.614e+02
##  poly(xGrid, 11, raw = TRUE)8   poly(xGrid, 11, raw = TRUE)9
##                       4.987e+04                       -6.389e+04
## poly(xGrid, 11, raw = TRUE)10  poly(xGrid, 11, raw = TRUE)11
##                       3.509e+04                       -7.484e+03
```

1.b.

```r
plot(xGrid, HW2Test(xGrid), col="Red")
pred1<- predict(fit1)
pred2<- predict(fit2)
pred3<- predict(fit3)
pred4<- predict(fit4)
lines(xGrid, pred1, col='Blue', lwd=2)
lines(xGrid, pred2, col='Green', lwd=2)
lines(xGrid, pred3, col='Black', lwd=2)
lines(xGrid, pred4, col='Yellow', lwd=2)
legend("topleft", legend=c("Actual", "Degree 17", "Degree 15", "Degree 13", "Degree 11"),
       col=c("Red", "Blue", "Green", "Black", "Yellow"), lty=1:2, cex=0.8)
title("Different degree Polynomials fit to the Gamma function")
```



1.c. Degrees 17,15,13, and 11.

```r
sqrt( mean( ( y - pred1)^2 ))
```

```
## [1] 0.001742046
```

```r
sqrt( mean( ( y - pred2)^2 ))
```

```
## [1] 0.002157356
```

```r
sqrt( mean( ( y - pred3)^2 ))
```

```
## [1] 0.004626954
```

```r
sqrt( mean( ( y - pred4)^2 ))
```

```
## [1] 0.007413881
```

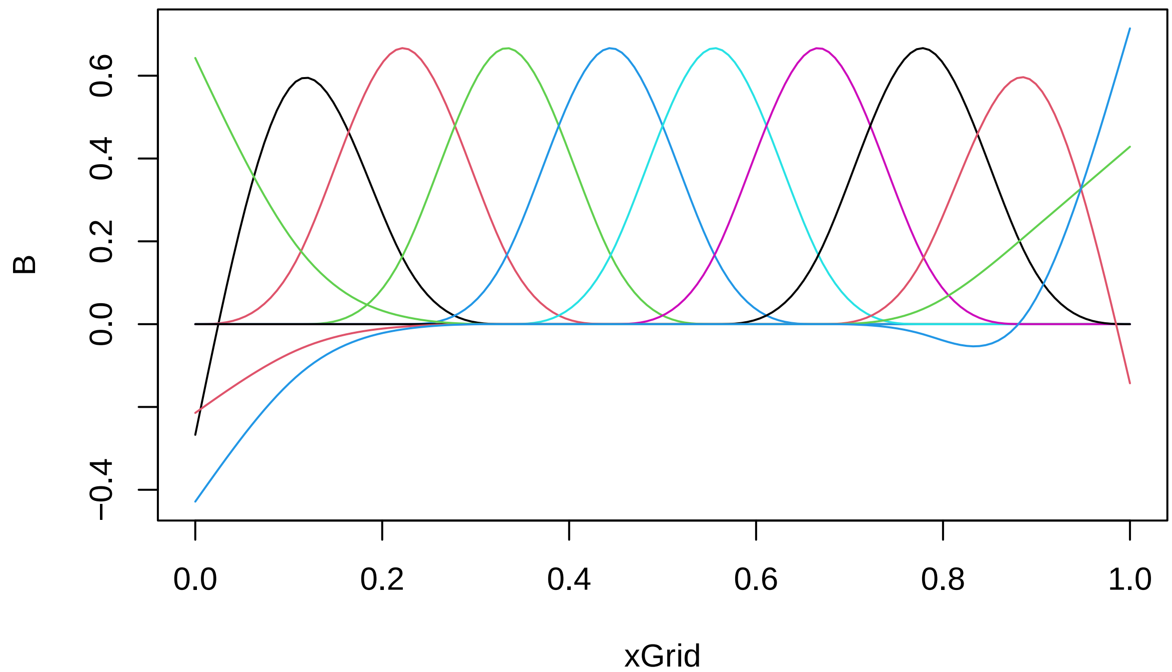
2.a.

```r
xGrid<- seq( 0,1,length.out=150)
KN<- seq( 0,1,length.out=10)
naturalSplineBasis <- function(sGrid,
                               sKnots,
                               degree = 3,
                               derivative = 0) {
  boundaryKnots<- c( min(sKnots),max(sKnots))
  sKnots0<- c( rep( boundaryKnots[1],degree),sort(sKnots),
               rep( boundaryKnots[2],degree) )
  testRight<- sGrid < min(sKnots)
  testLeft <- sGrid > max(sKnots)
  if( any(testRight |testLeft) )
  {stop("some points for evaluation outside knot range.")}

  basis <- splineDesign(sKnots0, sGrid,
                        ord= degree+1, outer.ok=TRUE,
                        derivs=derivative)
  # set up constraints to enforce natural BCs.
  const <- splineDesign(sKnots0, boundaryKnots, ord = degree+1,
                        derivs = c(2,2))
  qr.const <- qr(t(const))
  QBasis<- t(qr.qty( qr.const, t(basis) ))
  basis <- QBasis[,-(1:2)]
  basis

  return( basis )

}
B<- naturalSplineBasis( xGrid, sKnots=KN)
matplot( xGrid, B, type="l", lty=1)
```



Just one. 2.b.

```
NBSFit<- function( x, y, xGrid){
N<- length( x)
AData<- naturalSplineBasis( x, sKnots=x)
coef<- solve( AData, y)
AGrid<- naturalSplineBasis( xGrid, sKnots=x )
yFit <-AGrid%*%coef
return( yFit)
}
y <- HW2Test(xGrid)
NBSFit(xGrid,y,xGrid)
```

```
##                  [,1]
##    [1,] -2.220446e-16
##    [2,] -3.355705e-03
##    [3,] -6.711407e-03
##    [4,] -1.006700e-02
##    [5,] -1.342126e-02
##    [6,] -1.676781e-02
##    [7,] -2.008511e-02
##    [8,] -2.331980e-02
##    [9,] -2.636568e-02
##   [10,] -2.904338e-02
##   [11,] -3.108679e-02
##   [12,] -3.214119e-02
```

```
##  [13,] -3.177501e-02
##  [14,] -2.950489e-02
##  [15,] -2.483043e-02
##  [16,] -1.727415e-02
##  [17,] -6.421174e-03
##  [18,]  8.046000e-03
##  [19,]  2.632126e-02
##  [20,]  4.845925e-02
##  [21,]  7.437099e-02
##  [22,]  1.038300e-01
##  [23,]  1.364874e-01
##  [24,]  1.718933e-01
##  [25,]  2.095226e-01
##  [26,]  2.488027e-01
##  [27,]  2.891401e-01
##  [28,]  3.299456e-01
##  [29,]  3.706559e-01
##  [30,]  4.107509e-01
##  [31,]  4.497671e-01
##  [32,]  4.873061e-01
##  [33,]  5.230393e-01
##  [34,]  5.567090e-01
##  [35,]  5.881264e-01
##  [36,]  6.171677e-01
##  [37,]  6.437677e-01
##  [38,]  6.679132e-01
##  [39,]  6.896354e-01
##  [40,]  7.090017e-01
##  [41,]  7.261088e-01
##  [42,]  7.410753e-01
##  [43,]  7.540354e-01
##  [44,]  7.651329e-01
##  [45,]  7.745168e-01
##  [46,]  7.823371e-01
##  [47,]  7.887410e-01
##  [48,]  7.938712e-01
##  [49,]  7.978632e-01
##  [50,]  8.008444e-01
##  [51,]  8.029328e-01
##  [52,]  8.042372e-01
##  [53,]  8.048563e-01
##  [54,]  8.048795e-01
##  [55,]  8.043867e-01
##  [56,]  8.034492e-01
##  [57,]  8.021297e-01
##  [58,]  8.004837e-01
##  [59,]  7.985594e-01
##  [60,]  7.963989e-01
##  [61,]  7.940386e-01
##  [62,]  7.915097e-01
##  [63,]  7.888391e-01
##  [64,]  7.860498e-01
##  [65,]  7.831612e-01
##  [66,]  7.801898e-01
```

```
##  [67,]  7.771498e-01
##  [68,]  7.740527e-01
##  [69,]  7.709085e-01
##  [70,]  7.677254e-01
##  [71,]  7.645103e-01
##  [72,]  7.612689e-01
##  [73,]  7.580060e-01
##  [74,]  7.547254e-01
##  [75,]  7.514305e-01
##  [76,]  7.481239e-01
##  [77,]  7.448078e-01
##  [78,]  7.414839e-01
##  [79,]  7.381538e-01
##  [80,]  7.348186e-01
##  [81,]  7.314794e-01
##  [82,]  7.281368e-01
##  [83,]  7.247916e-01
##  [84,]  7.214443e-01
##  [85,]  7.180953e-01
##  [86,]  7.147449e-01
##  [87,]  7.113934e-01
##  [88,]  7.080411e-01
##  [89,]  7.046880e-01
##  [90,]  7.013344e-01
##  [91,]  6.979804e-01
##  [92,]  6.946260e-01
##  [93,]  6.912714e-01
##  [94,]  6.879165e-01
##  [95,]  6.845614e-01
##  [96,]  6.812062e-01
##  [97,]  6.778509e-01
##  [98,]  6.744955e-01
##  [99,]  6.711401e-01
## [100,]  6.677845e-01
## [101,]  6.644290e-01
## [102,]  6.610734e-01
## [103,]  6.577178e-01
## [104,]  6.543622e-01
## [105,]  6.510065e-01
## [106,]  6.476509e-01
## [107,]  6.442952e-01
## [108,]  6.409395e-01
## [109,]  6.375838e-01
## [110,]  6.342281e-01
## [111,]  6.308724e-01
## [112,]  6.275167e-01
## [113,]  6.241610e-01
## [114,]  6.208053e-01
## [115,]  6.174496e-01
## [116,]  6.140939e-01
## [117,]  6.107382e-01
## [118,]  6.073825e-01
## [119,]  6.040268e-01
## [120,]  6.006711e-01
```

```
## [121,]  5.973154e-01
## [122,]  5.939597e-01
## [123,]  5.906040e-01
## [124,]  5.872483e-01
## [125,]  5.838926e-01
## [126,]  5.805369e-01
## [127,]  5.771812e-01
## [128,]  5.738255e-01
## [129,]  5.704698e-01
## [130,]  5.671141e-01
## [131,]  5.637584e-01
## [132,]  5.604027e-01
## [133,]  5.570470e-01
## [134,]  5.536913e-01
## [135,]  5.503356e-01
## [136,]  5.469799e-01
## [137,]  5.436242e-01
## [138,]  5.402685e-01
## [139,]  5.369128e-01
## [140,]  5.335570e-01
## [141,]  5.302013e-01
## [142,]  5.268456e-01
## [143,]  5.234899e-01
## [144,]  5.201342e-01
## [145,]  5.167785e-01
## [146,]  5.134228e-01
## [147,]  5.100671e-01
## [148,]  5.067114e-01
## [149,]  5.033557e-01
## [150,]  5.000000e-01
```

$\int_{-\infty}^{\infty} \sqrt(2) cos(2\pi x) * \sqrt(2) cos(2\pi x) = 1$ $\int_{-\infty}^{\infty} \sqrt(2) sin(2\pi x) * \sqrt(2) sin(2\pi x) = 1$ $\int_{-\infty}^{\infty} \sqrt(2) cos(4\pi x) * \sqrt(2) cos(4\pi x) = 1$ $\int_{-\infty}^{\infty} \sqrt(2) sin(4\pi x) * \sqrt(2) sin(4\pi x) = 1$