BST210 HW3

Wenjie

Question 1

#1(a)

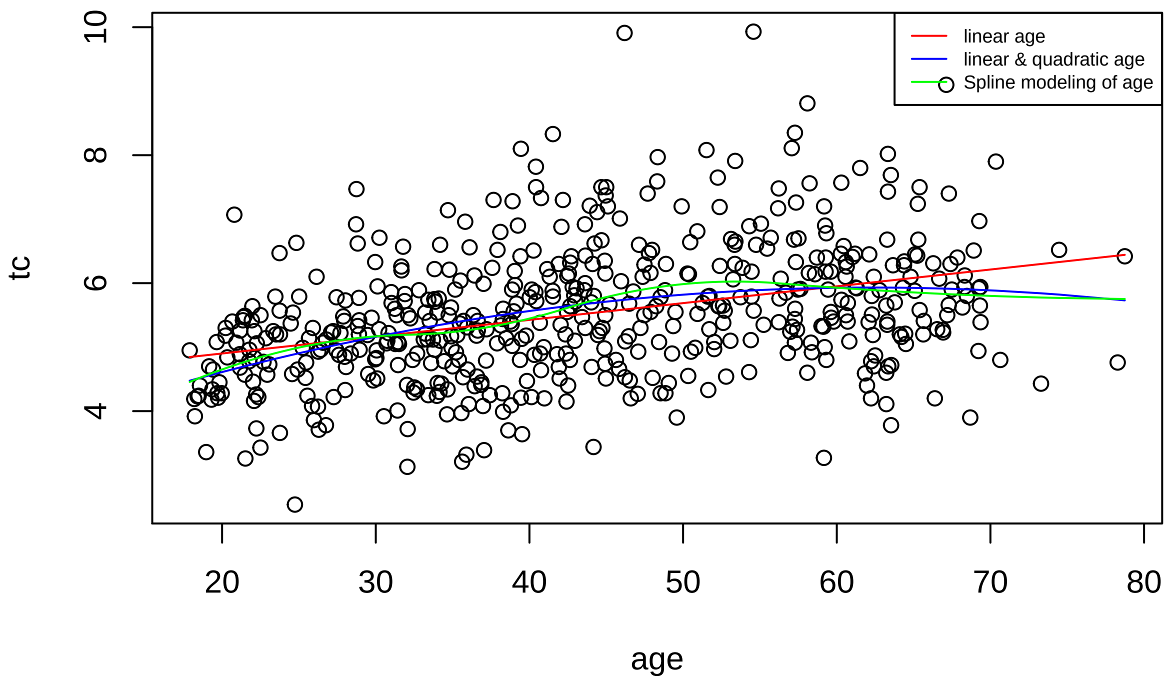
```r
dat <- read.csv("/Users/BiankaUrsul/Documents/Class Info/Harvard HSPH/Fall2019/BST210/Homework#1/Data ar
library(splines2)
library(gam)
```

```
## Loading required package: splines

## Loading required package: foreach

## Loaded gam 1.16.1
```

```r
sorted_dat = dat[order(dat$age),]
head(sorted_dat)
```

```
##       X caseID ethnic height weight waist    hip   tc   tg  hdl  ldl
## 44   44    417      2 170.30   55.1 68.00  88.75 4.95 0.68 1.14 3.50
## 343 343   2818      1 147.05   47.0 65.00  89.00 4.19 1.78 1.29 2.09
## 405 405   3456      3 166.65   58.0 64.75  99.50 3.92 0.60 1.47 2.18
## 342 342   2817      1 164.45   52.3 62.50  94.00 4.23 0.89 1.81 2.02
## 404 404   3452      3 161.05   59.5 69.00 101.00 4.24 0.73 1.25 2.66
## 22   22    227      1 170.45   71.5 75.00 100.00 4.40 0.73 1.33 2.74
##     diabetes hypertension educ drink smoke gender alcohol      age ihd
## 44         0            1    3     0     1      0       1 17.88912   0
## 343        0            1    3     0     1      1       1 18.19028   0
## 405        0            1    4     0     1      1       1 18.22861   0
## 342        0            1    3     0     1      1       1 18.40931   0
## 404        0            1    2     1     1      1       2 18.47502   0
## 22         0            1    3     1     1      0       2 18.55989   0
##     Dummy2
## 44       0
## 343      0
## 405      2
## 342      0
## 404      0
## 22       0
```

```r
mod1 = lm(tc~age, data = sorted_dat)
mod2 = lm(tc~age+I(age^2), data = sorted_dat)
mod3 = lm(tc~bSpline(age, knots = quantile(age, c(0.25,0.5,0.75)), degree =2), data = sorted_dat)

attach(sorted_dat)
plot(age, tc)
lines(age, fitted(mod1), col = "red")
lines(age, fitted(mod2), col = "blue")
lines(age, fitted(mod3), col = "green")
legend("topright", c("linear age", "linear & quadratic age", "Spline modeling of age"), col = c("red",
```




```
summary1 = summary(mod1)
summary2 = summary(mod2)
summary3 = summary(mod3)
summary1
```

```
##
## Call:
## lm(formula = tc ~ age, data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6589 -0.6383 -0.0557  0.5009  4.3216
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.376191   0.135848  32.214   <2e-16 ***
## age         0.026243   0.002966   8.849   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9801 on 525 degrees of freedom
## Multiple R-squared:  0.1298, Adjusted R-squared:  0.1281
## F-statistic: 78.31 on 1 and 525 DF,  p-value: < 2.2e-16
```

```
summary2
```

```
##
## Call:
## lm(formula = tc ~ age + I(age^2), data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6542 -0.6410 -0.0461  0.5151  4.1698
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.0658041  0.3907005   7.847 2.41e-14 ***
## age          0.0920305  0.0186508   4.934 1.08e-06 ***
## I(age^2)    -0.0007389  0.0002069  -3.572 0.000387 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9693 on 524 degrees of freedom
## Multiple R-squared:  0.1505, Adjusted R-squared:  0.1472
## F-statistic: 46.41 on 2 and 524 DF,  p-value: < 2.2e-16
```

```
summary3
```

```
##
## Call:
## lm(formula = tc ~ bSpline(age, knots = quantile(age, c(0.25,
##     0.5, 0.75)), degree = 2), data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6732 -0.6199 -0.0385  0.5465  4.0640
##
## Coefficients:
##                                                                   Estimate
## (Intercept)                                                         4.4568
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   0.7025
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   0.7616
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3   1.6759
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4   1.3197
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5   1.2961
##                                                                   Std. Error
## (Intercept)                                                           0.2402
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1     0.4027
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2     0.2427
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3     0.3012
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4     0.3183
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5     0.5712
##                                                                   t value
## (Intercept)                                                        18.557
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   1.745
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   3.138
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3   5.563
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4   4.146
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5   2.269
##                                                                   Pr(>|t|)
## (Intercept)                                                        < 2e-16
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   0.0817
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   0.0018
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3 4.24e-08
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4 3.94e-05
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5   0.0237
##
## (Intercept)                                                       ***
```

```
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1 .
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2 **
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3 ***
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4 ***
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9672 on 521 degrees of freedom
## Multiple R-squared:  0.1591, Adjusted R-squared:  0.1511
## F-statistic: 19.72 on 5 and 521 DF,  p-value: < 2.2e-16
```

```r
detach(sorted_dat)
```

For the spline modeling of age, I choose the knot points to be the points at 0.25, 0.5 and 0.75 percentile of age data and the order of polynomial set to be 2 (quadratic). Comparing the three sets of fitted values, the spline model curve has a similar pattern as the linear and quadratic age model with age larger than 60, and for age < 60, the spline model curve has a few more twists than the quadratic model. At age < 40, the trends of all three models look similar to each other. The R^2 value is 0.1281 for the linear age model, 0.1472 for the linear and quadratic age model and 0.1511 for the spline model. Therefore, according to the R^2 value, I would recommend the spline model as the "best".

#1(b)

```r
library(splines2)
mod4 = lm(tc~age + bSpline(age, knots = quantile(age, c(0.25,0.5,0.75)), degree = 2), data = sorted_dat
summary4 = summary(mod4)
summary4
```

```
##
## Call:
## lm(formula = tc ~ age + bSpline(age, knots = quantile(age, c(0.25,
##     0.5, 0.75)), degree = 2), data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6732 -0.6199 -0.0385  0.5465  4.0640
##
## Coefficients: (1 not defined because of singularities)
##                                                                      Estimate
## (Intercept)                                                          4.075773
## age                                                                  0.021299
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1 0.551468
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2 0.349448
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3 1.006160
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4 0.263129
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5       NA
##                                                                      Std. Error
## (Intercept)                                                            0.349471
## age                                                                    0.009386
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   0.379161
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   0.253330
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3   0.294457
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4   0.578250
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5         NA
```

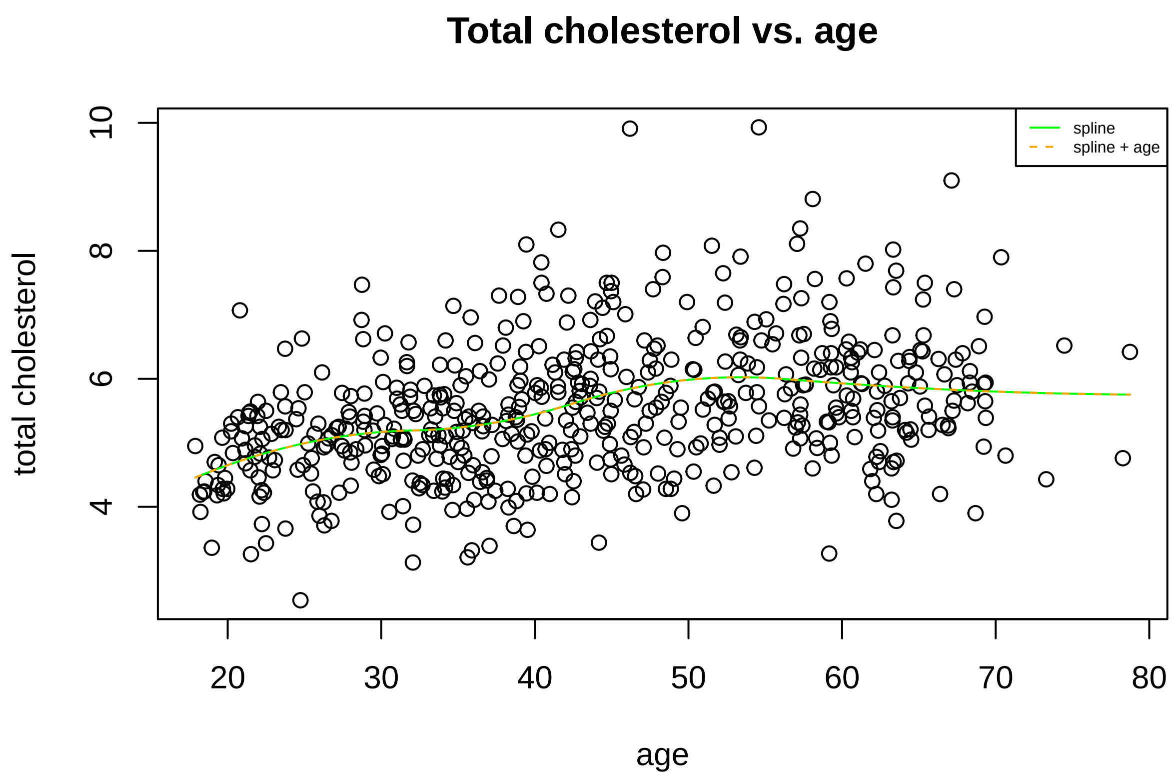
```
##                                                                   t value
## (Intercept)                                                        11.663
## age                                                                 2.269
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   1.454
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   1.379
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3   3.417
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4   0.455
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5      NA
##                                                                   Pr(>|t|)
## (Intercept)                                                         < 2e-16
## age                                                                0.023660
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1 0.146426
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2 0.168357
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3 0.000683
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4 0.649267
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5       NA
##
## (Intercept)                                                        ***
## age                                                                *
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3 ***
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9672 on 521 degrees of freedom
## Multiple R-squared:  0.1591, Adjusted R-squared:  0.1511
## F-statistic: 19.72 on 5 and 521 DF,  p-value: < 2.2e-16
```

```
summary3
```

```
##
## Call:
## lm(formula = tc ~ bSpline(age, knots = quantile(age, c(0.25,
##     0.5, 0.75)), degree = 2), data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6732 -0.6199 -0.0385  0.5465  4.0640
##
## Coefficients:
##                                                                   Estimate
## (Intercept)                                                         4.4568
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   0.7025
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   0.7616
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3   1.6759
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4   1.3197
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5   1.2961
##                                                                   Std. Error
## (Intercept)                                                           0.2402
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1     0.4027
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2     0.2427
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3     0.3012
```

```
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4     0.3183
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5     0.5712
##                                                                       t value
## (Intercept)                                                            18.557
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   1.745
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   3.138
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3   5.563
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4   4.146
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5   2.269
##                                                                       Pr(>|t|)
## (Intercept)                                                            < 2e-16
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   0.0817
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   0.0018
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3 4.24e-08
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4 3.94e-05
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5   0.0237
##
## (Intercept)                                                           ***
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1 .
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2 **
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3 ***
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4 ***
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9672 on 521 degrees of freedom
## Multiple R-squared:  0.1591, Adjusted R-squared:  0.1511
## F-statistic: 19.72 on 5 and 521 DF,  p-value: < 2.2e-16
```

```r
attach(sorted_dat)
plot(age, tc, main = "Total cholesterol vs. age", xlab = "age", ylab = "total cholesterol")
lines(age, fitted(mod3), col = "green", lty = "solid")
lines(age, fitted(mod4), col = "orange", lty = "dashed")
legend("topright", c("spline", "spline + age"), col = c("green", "orange"), lty = c("solid", "dashed"),
```




```
difference = fitted(mod3)-fitted(mod4)
difference
```

```
##            44           343           405           342           404
## -8.881784e-15 -3.375078e-14  4.263256e-14 -8.881784e-16  8.881784e-16
##            22           341            21            20           340
##  8.881784e-16  8.881784e-16  8.881784e-16  8.881784e-16  8.881784e-16
##           339           338            19            18           337
##  8.881784e-16  1.776357e-15  8.881784e-16  8.881784e-16  8.881784e-16
##           336           335           403           334            47
##  8.881784e-16  8.881784e-16  8.881784e-16  0.000000e+00  8.881784e-16
##            43           333           504           282           232
##  0.000000e+00  8.881784e-16  0.000000e+00  8.881784e-16  8.881784e-16
##           236           233           483           485            45
##  1.776357e-15  8.881784e-16  8.881784e-16  8.881784e-16  8.881784e-16
##           385           124           384           383            17
##  8.881784e-16  8.881784e-16  8.881784e-16  0.000000e+00  8.881784e-16
##            15           402            30            14            16
##  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  8.881784e-16
##           153           332            13           172           331
##  8.881784e-16  8.881784e-16  0.000000e+00  0.000000e+00  8.881784e-16
##           401           330           382            42            95
##  8.881784e-16  0.000000e+00  0.000000e+00  8.881784e-16  0.000000e+00
##           400            41            12           399            40
##  8.881784e-16  8.881784e-16  8.881784e-16  0.000000e+00  8.881784e-16
##           329            94           328            93            39
##  8.881784e-16  0.000000e+00  8.881784e-16  0.000000e+00  8.881784e-16
##            87            38            11           321           378
##  0.000000e+00  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
##           396           423             8            86             6
```

```
##  0.000000e+00  0.000000e+00  8.881784e-16  0.000000e+00  8.881784e-16
##           318           320             9           190             7
##  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00  8.881784e-16
##           319            10            29            85            84
##  0.000000e+00  0.000000e+00  8.881784e-16  0.000000e+00  0.000000e+00
##            37            83           377           316           357
##  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           463            36             5           314           315
##  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##            82            46            79           149            80
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##            81           148           313           147           376
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           509           356           310           188             3
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##            78           123           311            35           375
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16
##           146           189           312             4            77
##  0.000000e+00  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
##            28           462           186           307           308
##  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16  0.000000e+00
##           309           122           374           373           254
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16
##             2           448           381            73           372
## -8.881784e-16 -8.881784e-16  0.000000e+00  0.000000e+00  8.881784e-16
##            76           305           468            75            74
##  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16  0.000000e+00
##           306            27            26           327           187
##  0.000000e+00  0.000000e+00 -8.881784e-16 -8.881784e-16  0.000000e+00
##           184           486            72           145           121
## -8.881784e-16 -8.881784e-16  0.000000e+00 -8.881784e-16  0.000000e+00
##           231           415           503           496           271
##  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
##           501           345           281           235           412
## -8.881784e-16  0.000000e+00 -8.881784e-16  0.000000e+00 -8.881784e-16
##           183           495           494           304           469
## -8.881784e-16  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00
##           371           251           252           253           502
## -8.881784e-16  0.000000e+00 -8.881784e-16  0.000000e+00 -8.881784e-16
##           250           241           135           136           248
## -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16
##           269           240           270           482           249
##  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00 -8.881784e-16
##           279           182           414           325           505
##  0.000000e+00 -8.881784e-16  0.000000e+00 -8.881784e-16  0.000000e+00
##           230           395           303           492           368
## -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##             1           491           493           521           369
##  0.000000e+00  0.000000e+00 -8.881784e-16  0.000000e+00 -8.881784e-16
##           370           154           422           239           394
## -8.881784e-16 -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
##           246           234           128           247            91
## -8.881784e-16 -8.881784e-16 -8.881784e-16  0.000000e+00  0.000000e+00
##           524           511            90           238           181
```

```
## -8.881784e-16 -8.881784e-16 -8.881784e-16  0.000000e+00 -8.881784e-16
##           291           289           461           366           364
## -8.881784e-16 -8.881784e-16 -8.881784e-16  0.000000e+00 -8.881784e-16
##            53            89           322           460           459
##  0.000000e+00  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00
##           120           297           326           420           298
## -8.881784e-16  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00
##            57           392           179           150           290
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           142           440            56           191           293
## -8.881784e-16 -8.881784e-16  0.000000e+00  0.000000e+00 -8.881784e-16
##           407           355            32            52           292
## -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           268           219           294            64           119
## -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##            96           139           441           413           217
##  0.000000e+00  0.000000e+00 -8.881784e-16 -8.881784e-16  0.000000e+00
##           299           449            55           218           446
## -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           140           464            63           284           302
## -8.881784e-16 -8.881784e-16  0.000000e+00 -8.881784e-16  0.000000e+00
##           443           323           185            58           406
##  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16 -8.881784e-16
##           466            62            59           439            50
##  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
##           237           365            31           215           344
## -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           419           193           317           437           418
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16
##            33            51           180           424           450
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           213           411            97           467           520
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           359           152           143           214            60
##  0.000000e+00  0.000000e+00  8.881784e-16  0.000000e+00  0.000000e+00
##           442            61           300           223           398
##  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00  8.881784e-16
##           391           397            65           144           465
##  0.000000e+00  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00
##           278           386           141           220           216
##  0.000000e+00  0.000000e+00  0.000000e+00  8.881784e-16 -8.881784e-16
##            67           380           296            88           447
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           451            70           192           295           301
## -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           489           444           245            68           488
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           367            34           379           324           523
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           277           390            69            54           133
## -8.881784e-16  8.881784e-16  8.881784e-16  0.000000e+00  0.000000e+00
##           487           151            92            66           421
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           280           393           425           484           127
```

```
##  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##            71            98           244           490           134
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16
##           445            49           473           417           225
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           137           258           125           458           206
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           516           260           352           228           480
##  8.881784e-16  0.000000e+00  0.000000e+00  8.881784e-16  0.000000e+00
##           431           138           515           109           227
##  8.881784e-16  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
##           508           507           288           203           452
##  0.000000e+00  0.000000e+00  0.000000e+00  8.881784e-16  0.000000e+00
##           427           525           160           360           197
##  0.000000e+00  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
##           255           243           112           388           363
##  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
##           500           113           428           438           171
##  0.000000e+00  0.000000e+00  0.000000e+00  8.881784e-16  0.000000e+00
##           103            48           272           478           117
##  0.000000e+00  0.000000e+00  8.881784e-16  0.000000e+00  0.000000e+00
##           472           129           259           498           130
##  0.000000e+00  8.881784e-16  0.000000e+00 -8.881784e-16  0.000000e+00
##           209           351           456           257           165
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           409           201           261           226           256
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           174           207           499           354           116
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           426           349           476           156           512
##  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16 -8.881784e-16
##           518           430           114            24           470
##  0.000000e+00  0.000000e+00  8.881784e-16 -8.881784e-16  8.881784e-16
##           454           285           102           204           205
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           202           162           389           434           361
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16
##           265           166           347           111           164
##  0.000000e+00  0.000000e+00  8.881784e-16  8.881784e-16  0.000000e+00
##           132           479           436           350           416
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           159           194           155           198           527
##  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16
##           513           100           286           358           510
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           517           474           263           211           110
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           432           283           199           453           176
##  0.000000e+00  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00
##           471           221           353           266           519
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           481           433           210           167           169
##  0.000000e+00  0.000000e+00 -8.881784e-16  0.000000e+00  0.000000e+00
##           105           387           497           107           408
```

```
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##            25           212           170           208           195
##  0.000000e+00 -8.881784e-16  0.000000e+00  8.881784e-16  0.000000e+00
##           178           158           131           115            23
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           410           275           222           506           168
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16
##           522           104           526           126           224
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           161           264           276           196           287
##  0.000000e+00  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
##           173           118           514           108           157
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           435           457           475           267           362
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           274           175           346            99           229
##  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           477           429           106           177           348
## -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
##           200           455           101           273           262
##  0.000000e+00  0.000000e+00  0.000000e+00 -8.881784e-16  0.000000e+00
##           163           242
## -8.881784e-16  0.000000e+00
```

```r
detach(sorted_dat)
```

By plotting out the fitted line for spline(age) model and age+spline(age) model, we can see that the two curves overlap, and we found no difference in the predicted values in the two models, which means that the two models give the same prediction. Comparing the residual standard errors, multiple R-squared values and adjusted R-squared values, we found the two models having the exactly same statistics (RSE = 0.9672, multiple R^2 = 0.1591, adjusted R^2 = 0.1511). We can therefore conclude that the linear age model is nested within the spline model.

#1(c)

```r
mod5 = lm(tc~age+I(age^2) + bSpline(age, knots = quantile(age, c(0.25,0.5,0.75)), degree = 2), data = s
summary5 = summary(mod5)
summary5
```

```
##
## Call:
## lm(formula = tc ~ age + I(age^2) + bSpline(age, knots = quantile(age,
##     c(0.25, 0.5, 0.75)), degree = 2), data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6732 -0.6199 -0.0385  0.5465  4.0640
##
## Coefficients: (2 not defined because of singularities)
##                                                                     Estimate
## (Intercept)                                                        3.5341536
## age                                                                0.0584534
## I(age^2)                                                          -0.0003845
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1  0.3855986
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2  0.0303667
```

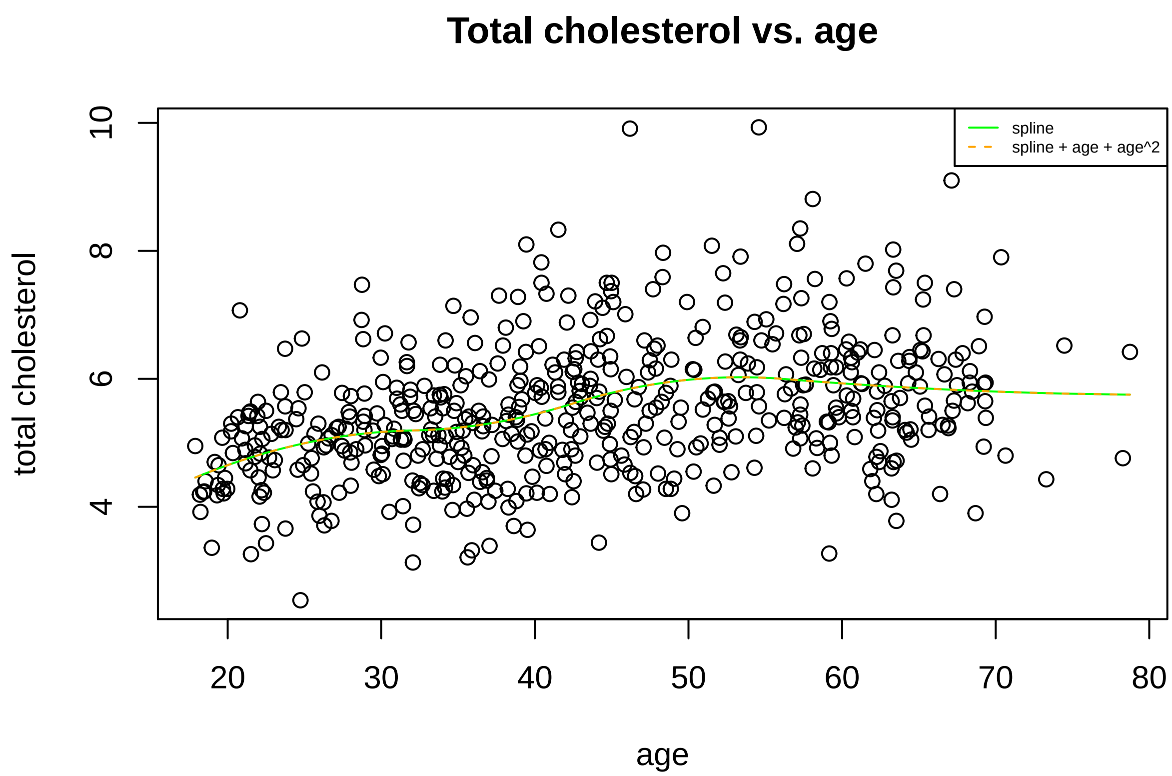
```
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3  0.6321756
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4         NA
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5         NA
##                                                                      Std. Error
## (Intercept)                                                           1.1037063
## age                                                                   0.0739899
## I(age^2)                                                              0.0008450
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1  0.4938852
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2  0.5458165
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3  0.6778777
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4         NA
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5         NA
##                                                                      t value
## (Intercept)                                                            3.202
## age                                                                    0.790
## I(age^2)                                                              -0.455
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   0.781
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   0.056
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3   0.933
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4      NA
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5      NA
##                                                                      Pr(>|t|)
## (Intercept)                                                           0.00145
## age                                                                   0.42988
## I(age^2)                                                              0.64927
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1  0.43531
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2  0.95565
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3  0.35147
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4       NA
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5       NA
##
## (Intercept)                                                          **
## age
## I(age^2)
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9672 on 521 degrees of freedom
## Multiple R-squared:  0.1591, Adjusted R-squared:  0.1511
## F-statistic: 19.72 on 5 and 521 DF,  p-value: < 2.2e-16
```

```
summary3
```

```
##
## Call:
## lm(formula = tc ~ bSpline(age, knots = quantile(age, c(0.25,
##     0.5, 0.75)), degree = 2), data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.6732 -0.6199 -0.0385  0.5465  4.0640 
## 
## Coefficients:
##                                                                        Estimate
## (Intercept)                                                              4.4568
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   0.7025
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   0.7616
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3   1.6759
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4   1.3197
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5   1.2961
##                                                                        Std. Error
## (Intercept)                                                                0.2402
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1     0.4027
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2     0.2427
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3     0.3012
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4     0.3183
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5     0.5712
##                                                                        t value
## (Intercept)                                                             18.557
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   1.745
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   3.138
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3   5.563
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4   4.146
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5   2.269
##                                                                        Pr(>|t|)
## (Intercept)                                                             < 2e-16
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1   0.0817
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2   0.0018
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3 4.24e-08
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4 3.94e-05
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5   0.0237
##                                                                           
## (Intercept)                                                            ***
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)1 .  
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)2 ** 
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)3 ***
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)4 ***
## bSpline(age, knots = quantile(age, c(0.25, 0.5, 0.75)), degree = 2)5 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.9672 on 521 degrees of freedom
## Multiple R-squared:  0.1591, Adjusted R-squared:  0.1511 
## F-statistic: 19.72 on 5 and 521 DF,  p-value: < 2.2e-16
```

```r
attach(sorted_dat)
plot(age, tc, main = "Total cholesterol vs. age", xlab = "age", ylab = "total cholesterol")
lines(age, fitted(mod3), col = "green", lty = "solid")
lines(age, fitted(mod5), col = "orange", lty = "dashed")
legend("topright", c("spline", "spline + age + age^2"), col = c("green", "orange"), lty = c("solid", "da
```




```r
detach(sorted_dat)
```

Similar as in 1(b), the fitted lines of the spline model and the linear + quadratic age model overlap and the residual standard error and R^2 statistics are exactly the same for the two models (Residual standard error = 0.9672, multiple R^2 = 0.1591, adjusted R^2 = 0.1511). Therefore, linear+quadratic age model is nested within the spline model.

#1(d)

```r
anova(mod2, mod5)
```

```
## Analysis of Variance Table
##
## Model 1: tc ~ age + I(age^2)
## Model 2: tc ~ age + I(age^2) + bSpline(age, knots = quantile(age, c(0.25,
##     0.5, 0.75)), degree = 2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    524 492.35
## 2    521 487.34  3    5.0098 1.7853 0.1489
```

Because linear+quadratic age model is nested within the spline model, we can do F test to compare the two models. The p-value for the F test is 0.1489 (> 0.05), indicating that the spline model is not significantly better than the linear and quadratic age model. It is sufficient to use linear and quadratic age model to model the effects of age.

Problem 2

#2(a)

```r
sorted_dat$bmi= sorted_dat$weight / (sorted_dat$height/100)^2
age.gender.bmi.1 = lm(tc~ bmi, data = sorted_dat)
summary(age.gender.bmi.1)
```

```
##
## Call:
## lm(formula = tc ~ bmi, data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7806 -0.6333 -0.1273  0.5735  4.2889
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.11005    0.25007  16.435  < 2e-16 ***
## bmi          0.05825    0.01019   5.719  1.8e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.019 on 525 degrees of freedom
## Multiple R-squared:  0.05864,    Adjusted R-squared:  0.05685
## F-statistic:  32.7 on 1 and 525 DF,  p-value: 1.803e-08
```

```r
age.gender.bmi.2 = lm(tc~ bmi + I(bmi^2) , data = sorted_dat)
summary(age.gender.bmi.2)
```

```
##
## Call:
## lm(formula = tc ~ bmi + I(bmi^2), data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7664 -0.6523 -0.1061  0.5651  4.2049
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.619210   0.996621   1.625  0.10483
## bmi          0.257627   0.077911   3.307  0.00101 **
## I(bmi^2)    -0.003859   0.001495  -2.581  0.01012 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.014 on 524 degrees of freedom
## Multiple R-squared:  0.07046,    Adjusted R-squared:  0.06691
## F-statistic: 19.86 on 2 and 524 DF,  p-value: 4.859e-09
```

```r
age.gender.bmi.3 = lm(tc ~ bmi +I(bmi^2) + age, data = sorted_dat)
summary(age.gender.bmi.3)
```

```
##
## Call:
## lm(formula = tc ~ bmi + I(bmi^2) + age, data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.7752 -0.6372 -0.0587  0.5478  4.2095
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.954701   0.951957   2.053   0.0405 *
## bmi          0.173209   0.075236   2.302   0.0217 *
## I(bmi^2)    -0.002647   0.001436  -1.843   0.0659 .
## age          0.022385   0.003084   7.258 1.43e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9674 on 523 degrees of freedom
## Multiple R-squared:  0.1555, Adjusted R-squared:  0.1507
## F-statistic: 32.11 on 3 and 523 DF,  p-value: < 2.2e-16
```

```
age.gender.bmi.4 = lm(tc ~ bmi +I(bmi^2) + gender, data = sorted_dat)
summary(age.gender.bmi.4)
```

```
##
## Call:
## lm(formula = tc ~ bmi + I(bmi^2) + gender, data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7357 -0.6449 -0.0942  0.5621  4.1637
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.478818   1.013540   1.459 0.145149
## bmi          0.266748   0.078836   3.384 0.000769 ***
## I(bmi^2)    -0.004044   0.001515  -2.669 0.007836 **
## gender       0.069322   0.090022   0.770 0.441613
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.014 on 523 degrees of freedom
## Multiple R-squared:  0.07151,    Adjusted R-squared:  0.06618
## F-statistic: 13.43 on 3 and 523 DF,  p-value: 1.875e-08
```

```
age.gender.bmi.5 = lm(tc~ bmi + I(bmi^2) + age + I(age*bmi), data = sorted_dat)
summary(age.gender.bmi.5)
```

```
##
## Call:
## lm(formula = tc ~ bmi + I(bmi^2) + age + I(age * bmi), data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7573 -0.6371 -0.0585  0.5458  4.2104
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.7770981  1.0340123   1.719   0.0863 .
## bmi           0.1745595  0.0753564   2.316   0.0209 *
## I(bmi^2)     -0.0023859  0.0015544  -1.535   0.1254
```

```
## age           0.0302682  0.0181064   1.672   0.0952 .
## I(age * bmi) -0.0003324  0.0007522  -0.442   0.6588
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9681 on 522 degrees of freedom
## Multiple R-squared:  0.1558, Adjusted R-squared:  0.1494
## F-statistic: 24.09 on 4 and 522 DF,  p-value: < 2.2e-16
```

```r
age.gender.bmi.6 = lm(tc~ bmi + I(bmi^2) + age + I(gender*bmi), data = sorted_dat)
summary(age.gender.bmi.6)
```

```
##
## Call:
## lm(formula = tc ~ bmi + I(bmi^2) + age + I(gender * bmi), data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7220 -0.6087 -0.0610  0.5387  4.1390
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.747319   0.964539   1.812   0.0706 .
## bmi               0.188251   0.076067   2.475   0.0136 *
## I(bmi^2)         -0.003010   0.001462  -2.058   0.0400 *
## age               0.022650   0.003089   7.333 8.64e-13 ***
## I(gender * bmi)   0.004604   0.003533   1.303   0.1931
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9667 on 522 degrees of freedom
## Multiple R-squared:  0.1583, Adjusted R-squared:  0.1518
## F-statistic: 24.54 on 4 and 522 DF,  p-value: < 2.2e-16
```

```r
final.model = age.gender.bmi.3
summary(final.model)
```

```
##
## Call:
## lm(formula = tc ~ bmi + I(bmi^2) + age, data = sorted_dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7752 -0.6372 -0.0587  0.5478  4.2095
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.954701   0.951957   2.053   0.0405 *
## bmi          0.173209   0.075236   2.302   0.0217 *
## I(bmi^2)    -0.002647   0.001436  -1.843   0.0659 .
## age          0.022385   0.003084   7.258 1.43e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9674 on 523 degrees of freedom
```

```
## Multiple R-squared:  0.1555, Adjusted R-squared:  0.1507
## F-statistic: 32.11 on 3 and 523 DF,  p-value: < 2.2e-16
```

We first compare the linear bmi model and the linear and quadratic bmi model, finding out that the coefficient for quadratic bmi term is significant. Therefore, we can make adjustment on top of the linear and quadratic bmi model. Now we're considering age as a confounder of bmi, which in conventional definition makes sense (age is a common cause of bmi and total cholesterol), and after including age term, the coefficients for linear and quadratic bmi terms change significantly, confirming that age is a confounder. But including gender does not effectively change the coefficients for linear or quadratic bmi. Therefore, we only consider age as a confounder.

Now we are considering whether age modifies the effect of bmi on total cholesterol. However the coefficient for the interaction term of age & bmi is not significant, indicating that age does not modify the effect of bmi on total cholesterol. Similarly, the interaction between gender and bmi is not significant, either.

Therefore, the final model I recommend is the model including linear bmi, quadratic bmi and age to predict total cholesterol. (Among the models I tried above, this model has the highest adjusted R^2 value as well.)

#2(b)

```r
confint(final.model) *38.67
```

```
##                  2.5 %       97.5 %
## (Intercept)  3.2703611 1.479062e+02
## bmi          0.9824549 1.241351e+01
## I(bmi^2)    -0.2114889 6.753188e-03
## age          0.6313305 1.099916e+00
```

```r
summary(final.model)$coefficients * 38.67
```

```
##               Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) 75.5882942 36.81219501  79.40302 1.567507e+00
## bmi          6.6979836  2.90939118  89.02585 8.397622e-01
## I(bmi^2)    -0.1023679  0.05554619 -71.26619 2.548531e+00
## age          0.8656231  0.11926258 280.67182 5.520816e-11
```

Total Cholesterol = 1.9547 + 0.1732 * bmi - 0.002647 * bmi^2 + 0.02239 * age

My final model uses linear bmi(in kg/m^2) ($p = 0.0217$), quadratic bmi(in $(kg/m^2)^2$) ($p = 0.0659$) and age ($p = 1.43 \times 10^{-12}$) to predict total cholesterol (in mg/dl) of subjects.

It is not sensible to interpret the intercept 75.59 mg/dl (95% confidence interval: [3.27, 147.91]), as bmi cannot be 0. After adjusting for age, every unit (kg/m^2) increase in the linear bmi term (bmi) only will contribute a 6.698 mg/dl (95% confidence interval: [0.983, 12.41]) increase in total cholesterol, and every unit($(kg/m^2)^2$) increase in the quadratic bmi term (bmi^2) will contribute a -0.1024 mg/dl (95% confidence interval: [-0.211, -6.75 $\times 10^{-3}$]) decrease in total cholesteral.