SRM analyses

Ken

4/7/2022

Introduction

This document provides all the code and output for the analysis that has been previously discussed. These are split into data cleaning steps, summary of the cleaned data, and statistical analysis of the cleaned data. Naturally, each step builds on the earlier steps, so it is important to verify that the earlier steps make sense, before going much further.

The coding is done entirely in the R software, using (where possible) well-known functions, that most R users would understand. More idiosyncratic use of R is noted, when it arises.

Please feel free to ask questions, and to correct anything I've done that doesn't match the intended analysis.

Data cleaning

First we read in the data and note how the column letters (in Excel) match up to the variable names we use in this analysis:

```
library("readxl")
```

```
## Warning: package ’readxl’ was built under R version 4.1.3
```

```
srm <- read_excel( "Copy of 2013 2014 for data entry.xlsx", sheet=1 )
```

```
## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in Z1004 / R1004C26: got ’12.5, 12’
```

```
## Warning in read_fun(path = enc2native(normalizePath(path)), sheet_i = sheet, :
## Expecting numeric in N1320 / R1320C14: got ’<20.0’
```

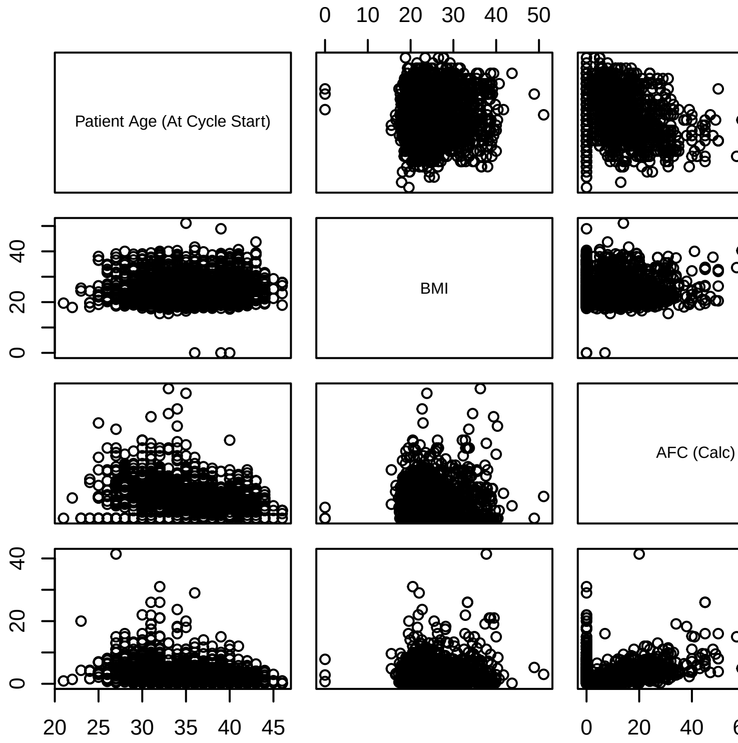
```
#names(srm)
dLETTERS <- sapply(1:26, function(i){paste(LETTERS[i], LETTERS[i], sep="")})
cbind(c(LETTERS,dLETTERS)[1:38] , names(srm))
```

```
##       [,1] [,2]
##  [1,] "A"  "MPI"
##  [2,] "B"  "Patient Age (At Cycle Start)"
##  [3,] "C"  "Treatment #"
##  [4,] "D"  "BMI"
##  [5,] "E"  "AFC (Calc)"
##  [6,] "F"  "AMH"
##  [7,] "G"  "Lupron Protocol"
##  [8,] "H"  "E2 Day 3"
##  [9,] "I"  "FSH Dose Day 3"
## [10,] "J"  "E2 Day 4"
## [11,] "K"  "FSH Dose Day 4"
## [12,] "L"  "E2 Day 5"
## [13,] "M"  "FSH Dose Day 5"
## [14,] "N"  "E2 Day 6"
## [15,] "O"  "FSH Dose Day 6"
## [16,] "P"  "E2 Day 7"
## [17,] "Q"  "FSH Dose Day 7"
## [18,] "R"  "E2 Day 8"
## [19,] "S"  "FSH Dose Day 8"
## [20,] "T"  "E2 Day 9"
## [21,] "U"  "FSH Dose Day 9"
## [22,] "V"  "E2 Day 10"
## [23,] "W"  "FSH Dose Day 10"
## [24,] "X"  "E2 Day 11"
## [25,] "Y"  "FSH Dose Day 11"
## [26,] "Z"  "E2 Day 12"
## [27,] "AA" "FSH Dose Day 12"
## [28,] "BB" "FD US #1"
## [29,] "CC" "FD US #2"
## [30,] "DD" "FD US #3"
## [31,] "EE" "FD US #4"
## [32,] "FF" "E2 Day 13"
## [33,] "GG" "FSH Dose Day 13"
## [34,] "HH" "E2 Day 14"
## [35,] "II" "FSH Dose Day 14"
## [36,] "JJ" "FD US #5"
## [37,] "KK" "E2 Day 15"
## [38,] "LL" "FSH Dose Day 15"
```

Some simple numeric summaries of some variables of interest

```
# variables of interest
summary(srm[,c(2,4,5,6)])
```

```
##  Patient Age (At Cycle Start)      BMI           AFC (Calc)           AMH
##  Min.   :21.00                Min.   : 0.00   Min.   : 0.000   Min.   : 0.017
##  1st Qu.:32.00                1st Qu.:21.40   1st Qu.: 0.000   1st Qu.: 0.790
##  Median :35.00                Median :23.90   Median : 7.000   Median : 2.000
##  Mean   :35.46                Mean   :25.07   Mean   : 9.571   Mean   : 2.971
##  3rd Qu.:39.00                3rd Qu.:27.60   3rd Qu.:15.250   3rd Qu.: 3.900
##  Max.   :46.00                Max.   :51.10   Max.   :83.000   Max.   :41.410
##                                                                NA’s   :46
```



Simple pairwise scatterplots of some variables of interest

Examining protocol

```
# Protocol
table(srm[,7], useNA="ifany") # couple of NA, also unstimulated
```

```
##
##       Antagonist          LPL 10/5           Lupron Lupron Microdose
##              901               622                1              322
##      Not entered       Unstimulated
##                2                 4
```

Examining FSH dose: there are some strange values, which we need to omit and then convert the stored data to be numeric, not character strings:

```
# FSH dose
summary(srm[,c(9,11,13)]) # what does PM, QD mean? Also blank? (Skip for now)
```

```
##  FSH Dose Day 3      FSH Dose Day 4      FSH Dose Day 5
##  Length:1852         Length:1852         Length:1852
##  Class :character    Class :character    Class :character
##  Mode  :character    Mode  :character    Mode  :character
```

```
table(srm[,9]) # some 5 QD are okay? (presume they are, for now)
```

```
##
##      100   100 PM      112    112.5 112.5 PM      125   125 PM      150
##       20        2        1        4        2       14        1      170
##   150 pm   150 PM    162.5      175   175 PM    187.5      200   200 PM
##        1       18        1       14        3        5       72        5
##      225   225 PM      250   250 PM    262.5      275   275 PM      300
##      224       23       40       12        5        9        2      269
##   300 [M   300 PM   300 QD    337.5      350      375   375 PM   375 QD
##        1       39        2        1        2      188       21        5
##  3775 QD   400 PM      425   425 PM      450   450 PM   450 QD      475
##        1        1        2        1      115       10        1        1
##     5 QD      525   525 PM      600   600 PM   600 QD       75    75 PM
##        2       19        1        2        1        1       10        2
```

```
table(srm[,11])
```

```
##
##                125                 15                150             150 PM
##                  1                  1                  6                  1
##                175                200             200 PM                225
##                  2                  1                  1                  2
##                250                300             300 QD                375
##                  2                  7                  1                  2
##             375 QD                400                450                525
##                  1                  1                  4                  2
## 69.599999999999994
##                  1
```

```
table(srm[,13])
```

```
##
##      100   100 PM      112    112.5 112.5 PM      125   125 PM      150
##       26        1        1        5        1       16        2      177
##   150 pm   150 PM      175   175 PM    187.5      200   200 PM      225
##        1       19       16        4        3       68        5      199
##   225 PM   225 QD      250   250 PM      255    262.5      275   275 PM
##       19        1       46        8        1        5        8        2
##      300   300 PM   300 QD      350   350 PM      375   375 PM   375 QD
##      288       43        5        4        1      223       27       11
##      400      414      425   425 PM      450   450 pm   450 PM   450 QD
##        5        1        2        1      327        1       34       32
##      475       50      525   525 PM   525 QD      575      600   600 PM
##        3        1       52        8        7        1        1        1
##       75    75 PM
##       19        6
```

```
# convert to numbers
srm$fsh3 <- as.numeric(apply(srm[,9], 1, function(x){strsplit(x," ", fixed=TRUE)[[1]][1]}))
srm$fsh4 <- as.numeric(apply(srm[,11], 1, function(x){strsplit(x," ", fixed=TRUE)[[1]][1]}))
srm$fsh5 <- as.numeric(apply(srm[,13], 1, function(x){strsplit(x," ", fixed=TRUE)[[1]][1]}))
summary(srm[,c("fsh3","fsh4","fsh5")])
```

```
##       fsh3             fsh4            fsh5      
##  Min.   :   5.0   Min.   : 15.0   Min.   : 50.0  
##  1st Qu.: 200.0   1st Qu.:150.0   1st Qu.:225.0  
##  Median : 300.0   Median :275.0   Median :300.0  
##  Mean   : 279.8   Mean   :270.4   Mean   :312.3  
##  3rd Qu.: 375.0   3rd Qu.:375.0   3rd Qu.:450.0  
##  Max.   :3775.0   Max.   :525.0   Max.   :600.0  
##  NA's   :506      NA's   :1816    NA's   :114    
```

Filling in the missing FSH3 values with those from later in the study, where these are available:

```
# fill in missing FSH3 values with later, if available
table(is.na(srm$fsh3))
```

```
## 
## FALSE  TRUE 
##  1346   506
```

```
srm$fsh3[is.na(srm$fsh3)] <- srm$fsh4[is.na(srm$fsh3)]
table(is.na(srm$fsh3))
```

```
## 
## FALSE  TRUE 
##  1375   477
```

```
srm$fsh3[is.na(srm$fsh3)] <- srm$fsh5[is.na(srm$fsh3)]
table(is.na(srm$fsh3))
```

```
## 
## FALSE  TRUE 
##  1791    61
```

Examining estradiol:

```
# Estradiol
names(srm)[c(20,22,24,26)]
```

```
## [1] "E2 Day 9"  "E2 Day 10" "E2 Day 11" "E2 Day 12"
```

```
summary(srm[,c(20,22,24,26)])
```

```
##    E2 Day 9           E2 Day 10       E2 Day 11       E2 Day 12      
##  Length:1852        Min.   :   60   Min.   :  103   Min.   :   56.1  
##  Class :character   1st Qu.: 1120   1st Qu.: 1053   1st Qu.: 1218.0  
##  Mode  :character   Median : 1834   Median : 1720   Median : 1774.5  
##                     Mean   : 2208   Mean   : 2228   Mean   : 2169.2  
##                     3rd Qu.: 2859   3rd Qu.: 2818   3rd Qu.: 2684.2  
##                     Max.   :14998   Max.   :18712   Max.   :16939.0  
##                     NA's   :952     NA's   :1209    NA's   :1530     
```

There's a single ">3000" value in E2 Day 9. Change it to NA.

```
srm[which(srm[,20]=="> 3000"),20]
```

```
## # A tibble: 1 x 1
##   ‘E2 Day 9‘
##   <chr>
## 1 > 3000
```

```
srm[1436,c(20,22,24,26)]
```

```
## # A tibble: 1 x 4
##   ‘E2 Day 9‘ ‘E2 Day 10‘ ‘E2 Day 11‘ ‘E2 Day 12‘
##   <chr>            <dbl>       <dbl>       <dbl>
## 1 > 3000              NA          NA          NA
```

```
srm[1436,20] <- NA
srm[,20] <- as.numeric(unlist(srm[,20]))
```

```
## Warning: NAs introduced by coercion
```

Construct last value recorded for E2 – after first omitting 339 (!) observations with no E2 at all

```
# first omit 339 (!) observations with no E2 at all
table( apply( srm[,c(20,22,24,26)], 1, function(x){sum( is.na(x) )}))
```

```
##
##   0   1   2   3   4
##   3  99 634 777 339
```

```
na.counts <- apply( srm[,c(20,22,24,26)], 1, function(x){sum( is.na(x) )})
srm <- srm[na.counts !=4,]
#dim(srm)
srm <- as.data.frame(srm)
srm$lastE2 <- srm[,20]
table(is.na(srm$lastE2))
```

```
##
## FALSE  TRUE
##   489  1024
```

```
srm$lastE2 <- ifelse(!is.na(srm[,22]), srm[,22], srm$lastE2)
table(is.na(srm$lastE2))
```

```
##
## FALSE  TRUE
##  1225   288
```

```
srm$lastE2 <- ifelse(!is.na(srm[,24]), srm[,24], srm$lastE2)
table(is.na(srm$lastE2))
```

```
##
## FALSE  TRUE
##  1510     3
```

```
srm$lastE2 <- ifelse(!is.na(srm[,26]), srm[,26], srm$lastE2)
table(is.na(srm$lastE2))
```

```
##
## FALSE
##  1513
```

Removing 2 BMIs of zero:

```
srm <- srm[srm$BMI>0,]
#dim(srm)
```

Remove a single "Lupron" only lupron protocol also single "Unstimulated"

```
srm <- subset(srm, srm[,7]!="Lupron")
srm <- subset(srm, srm[,7]!="Unstimulated")
dim(srm)
```

```
## [1] 1509   42
```

Remove a single FSH3 value in excess of 3000

```
srm <- subset(srm, fsh3 < 3000)
```

Our parameters of interest are: age BMI; antral follicle count (AFC) and AMH, listed in columns B, D, E and F respectively

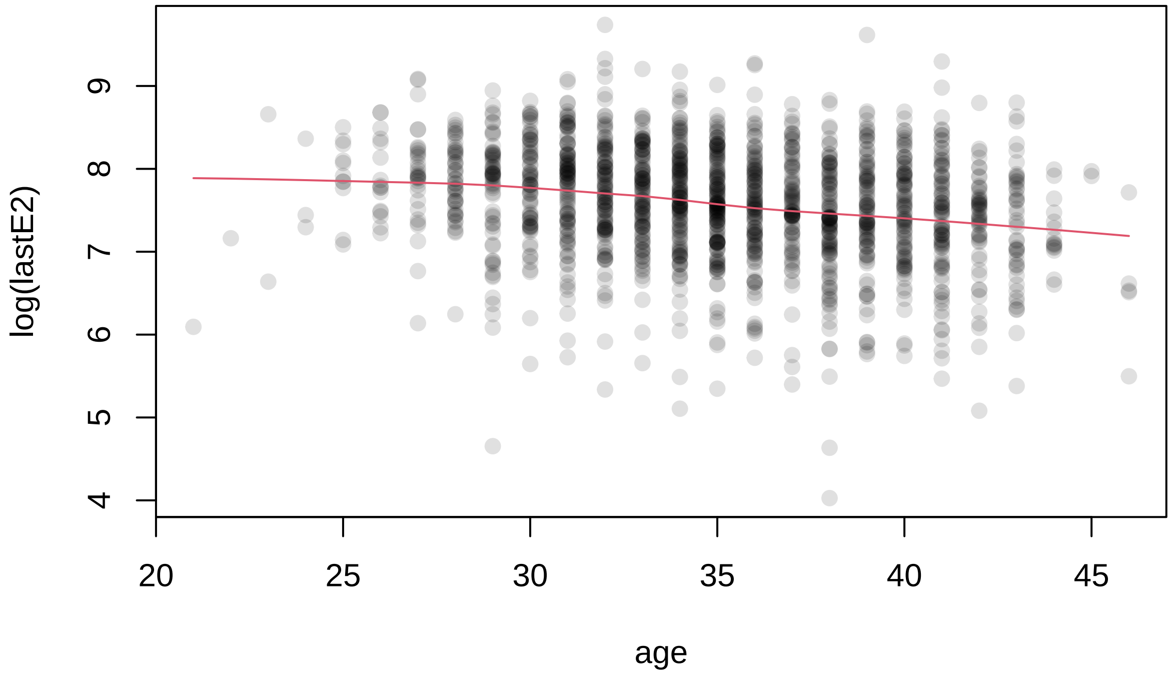
Three "protocols" are used. microdose Lupron or MDL; long lupron or LL and antagonist.

Setting up variable with names that match these, which will make subsequent code easier to read

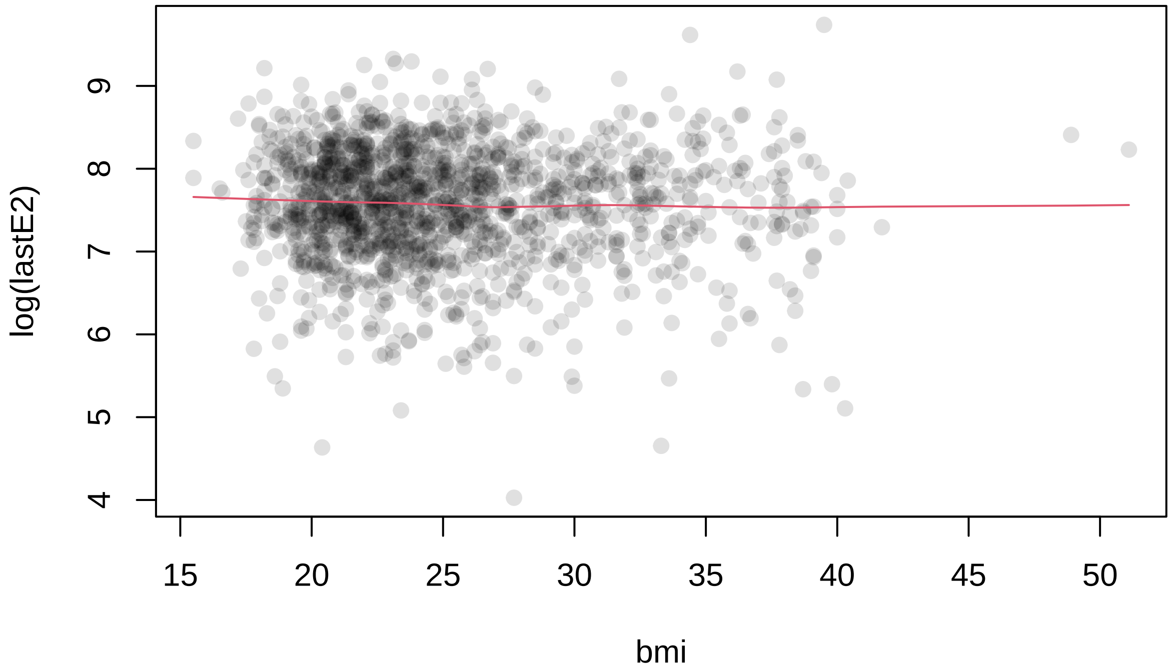
```
srm$age <- srm[,2]
srm$bmi <- srm$BMI
srm$lupprot <- srm[,7]
srm$amh <- srm$AMH
srm$afc <- srm[,5]
```

Data summary

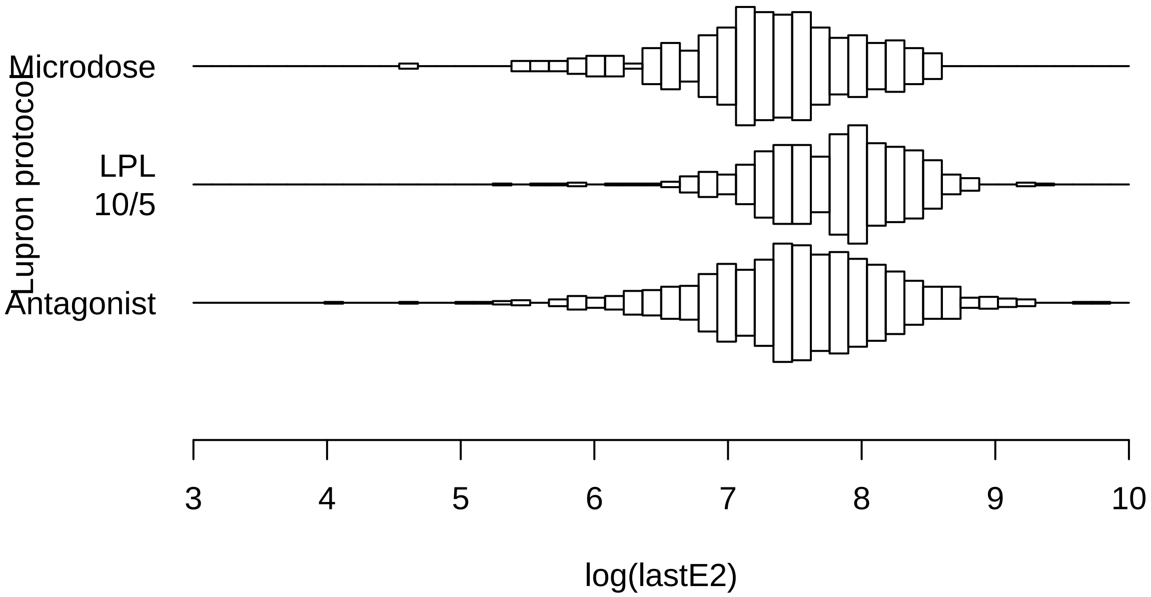
```
plot(log(lastE2)~age , data=srm, pch=19, col="#00000020")
lines(lowess(x=srm$age, y=log(srm$lastE2), iter=0), col=2)
```



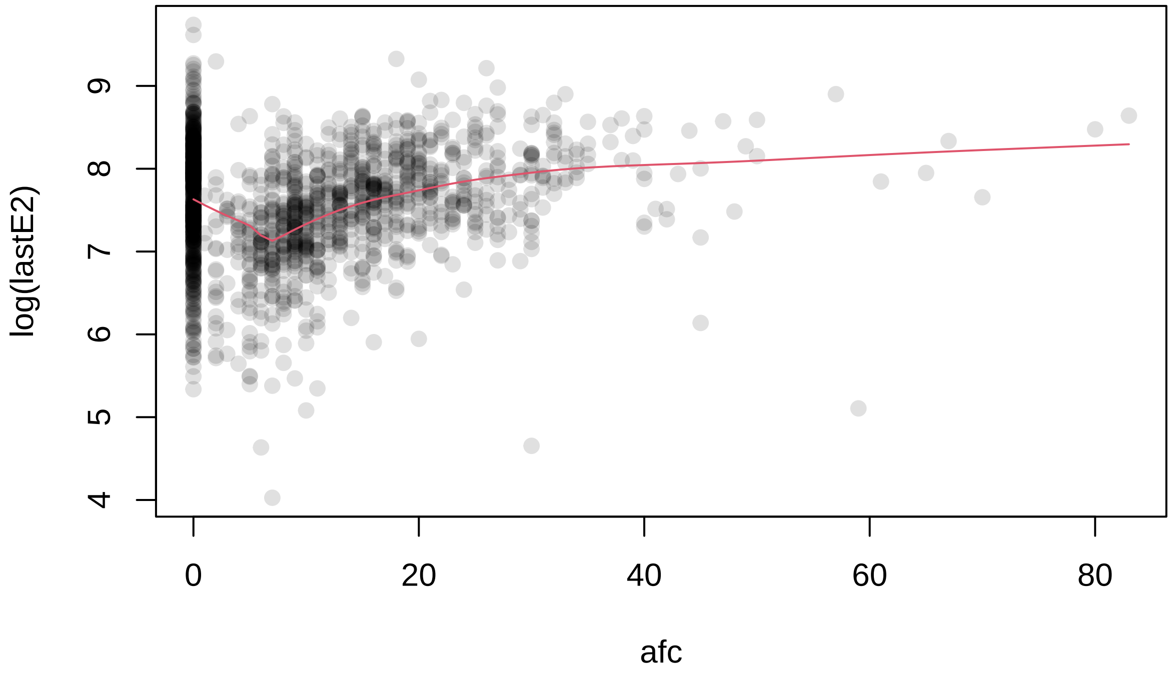
```
plot(log(lastE2)~bmi, data=srm, pch=19, col="#00000020")
lines(lowess(x=srm$bmi, y=log(srm$lastE2), iter=0), col=2)
```



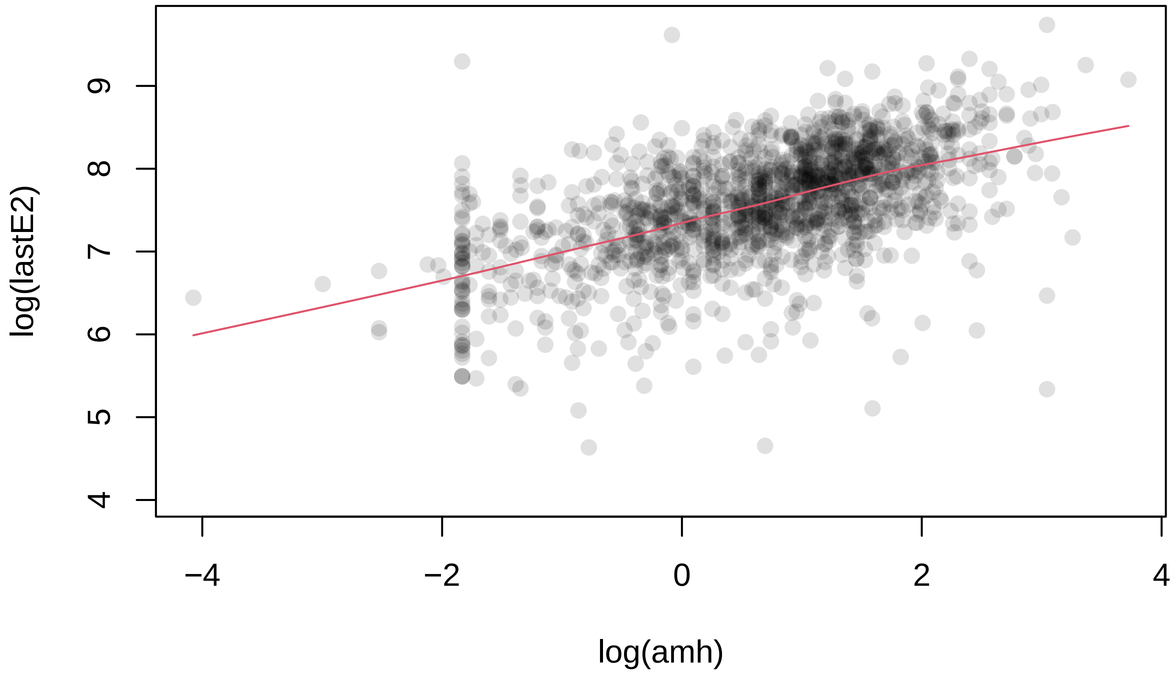
```
#table(srm$lupprot)
library("violinplot")
plot(0,0,xlim=c(3,10), ylim=c(0,4), axes=FALSE, xlab="log(lastE2)", ylab="Lupron protocol")
with(subset(srm, lupprot=="Antagonist"), violinplot(log(lastE2), breaks=seq(3,10,l=51), at=1, add=TRUE))
with(subset(srm, lupprot=="LPL 10/5"), violinplot(log(lastE2), breaks=seq(3,10,l=51), at=2, add=TRUE))
with(subset(srm, lupprot=="Lupron Microdose"), violinplot(log(lastE2), breaks=seq(3,10,l=51), at=3, add=
mtext(side=2, at=1:3, c("Antagonist", "LPL\n10/5", "Lupron Microdose"), las=1)
axis(side=1)
```



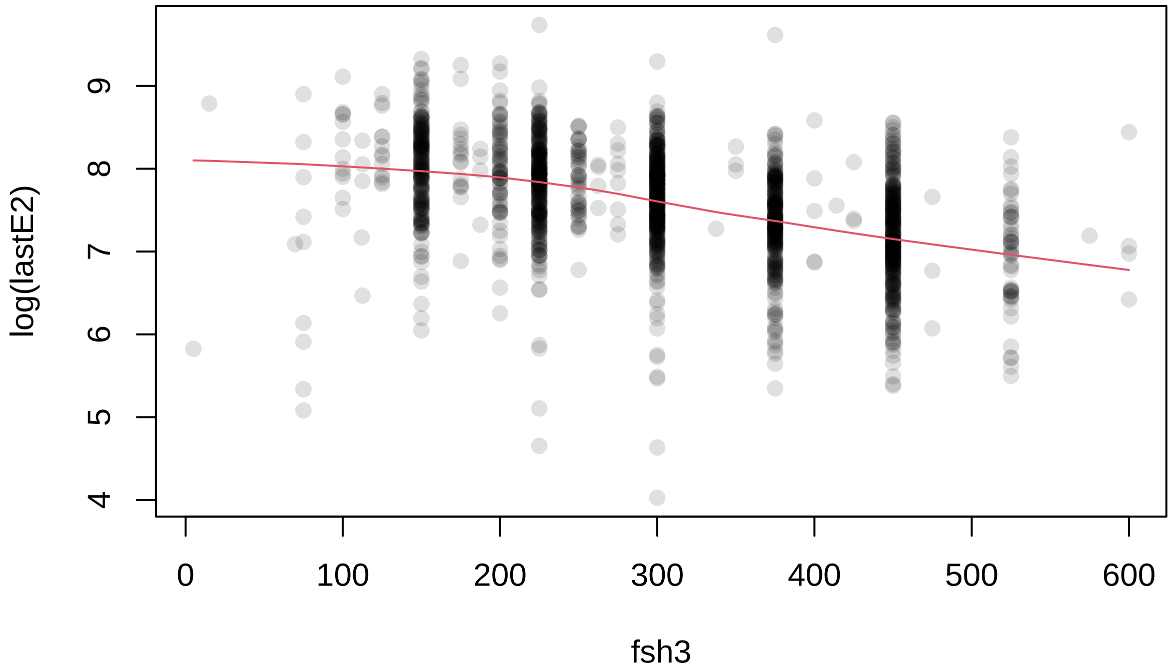
```
plot(log(lastE2)~afc, data=srm, pch=19, col="#00000020")
with(subset(srm), lines(lowess(x=afc, y=log(lastE2), iter=0), col=2))
```



```
plot(log(lastE2)~log(amh), data=srm, pch=19, col="#00000020")
with(subset(srm, !is.na(amh)), lines(lowess(x=log(amh), y=log(lastE2), iter=0), col=2))
```



```
plot(log(lastE2)~fsh3, data=srm, pch=19, col="#00000020")
with(subset(srm, !is.na(amh)), lines(lowess(x=fsh3, y=log(lastE2), iter=0), col=2))
```



Correlations between pairs of variables

```
srm$logamh <- log(srm$amh)
srm$loglastE2 <- log(srm$lastE2)
round(cor(srm[,c("age","bmi","afc","logamh","fsh3", "lastE2")], use="pairwise.complete.obs"),3)
```

```
##           age   bmi    afc logamh   fsh3 lastE2
## age     1.000 0.024 -0.183 -0.360  0.467 -0.208
## bmi     0.024 1.000  0.046  0.041  0.008  0.005
## afc    -0.183 0.046  1.000  0.245 -0.277  0.114
## logamh -0.360 0.041  0.245  1.000 -0.672  0.507
## fsh3    0.467 0.008 -0.277 -0.672  1.000 -0.439
## lastE2 -0.208 0.005  0.114  0.507 -0.439  1.000
```

```
round(cor(srm[,c("age","bmi","afc","logamh","fsh3", "loglastE2")], use="pairwise.complete.obs"),3)
```

```
##              age    bmi    afc logamh   fsh3 loglastE2
## age        1.000  0.024 -0.183 -0.360  0.467    -0.232
## bmi        0.024  1.000  0.046  0.041  0.008    -0.033
## afc       -0.183  0.046  1.000  0.245 -0.277     0.154
## logamh    -0.360  0.041  0.245  1.000 -0.672     0.563
## fsh3       0.467  0.008 -0.277 -0.672  1.000    -0.453
## loglastE2 -0.232 -0.033  0.154  0.563 -0.453     1.000
```

Statistical analysis

For convenience, construct variables indicating whether AFC=0, and “dummy variables” encoding Lupron protocols;

```
srm$afc0 <- ifelse(srm$afc==0, 1, 0)
table(srm$lupprot)
```

```
##
##       Antagonist         LPL 10/5 Lupron Microdose
##              750              519              222
```

```
srm$lup.lpl05<- ifelse(srm$lupprot=="LPL 10/5", 1, 0)
srm$lup.lpmic<- ifelse(srm$lupprot=="Lupron Microdose", 1, 0)
```

A first analysis: linear regression of log-last E2 value on FSH adjusting for age, BMI, lupron protocol, AFC and whether AFC=0, and log AMH. Those with missing AMH values are omitted:

```
#srm$cutafc <- cut(srm$afc, c(-1,0,5,10,15,20,30,100))
#table(srm$cutafc)
clean.srm <- subset(srm, !is.na(amh))
m1 <- lm(log(lastE2)~age + bmi + afc + afc0 + log(amh) + factor(lupprot) +fsh3, data=clean.srm)
cmat <- coef(summary(m1))
#library("rigr")
#m1.r <- regress("mean", loglastE2~age + bmi + cutafc +logamh + factor(lupprot) + fsh3, data=clean.srm)
#print(m1.r)
signif(cbind(est=cmat[,1], confint(m1), p.value=cmat[,4]),3)
```

```
##                                       est    2.5 %    97.5 %   p.value
## (Intercept)                      7.390000  7.08000  7.690000 1.27e-299
## age                              0.005310 -0.00186  0.012500  1.47e-01
## bmi                             -0.007410 -0.01290 -0.001910  8.31e-03
## afc                              0.005340  0.00127  0.009410  1.02e-02
## afc0                             0.163000  0.07340  0.253000  3.77e-04
## log(amh)                         0.302000  0.26600  0.339000  9.41e-55
## factor(lupprot)LPL 10/5          0.241000  0.17700  0.304000  1.54e-13
## factor(lupprot)Lupron Microdose  0.277000  0.18600  0.368000  3.21e-09
## fsh3                            -0.000799 -0.00117 -0.000429  2.40e-05
```

Turn this into a nomogram:

```
## Warning: package 'rms' was built under R version 4.1.3

## Warning: package 'Hmisc' was built under R version 4.1.3

## Warning: package 'Formula' was built under R version 4.1.1

## Warning: package 'ggplot2' was built under R version 4.1.2

##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

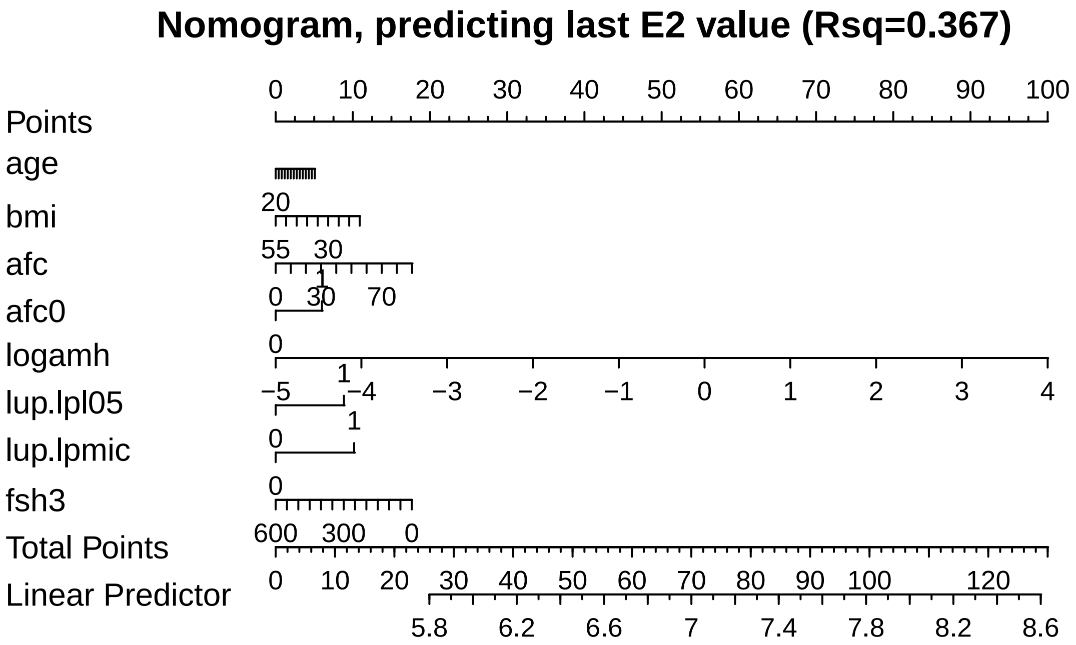
```
## Warning: package 'SparseM' was built under R version 4.1.1
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
## "packedMatrix" of class "replValueSp"; definition not updated
```

```
## Warning in .recacheSubclasses(def@className, def, env): undefined subclass
## "packedMatrix" of class "mMatrix"; definition not updated
```



Interpretation: this plot shows that AMH is doing the bulk of the work when the model determines a value for (mean) E2 among those with particular covariate values. For all the other variables, comparing individuals at opposite ends of the plotted axis, the difference in log E2 value is not impressive. But for those with even minor AMH differences, we see greater differentiation between their mean log E2 values.

Residual confounding might be a concern here, so a version that adjusts more flexibly for age, AMH, and then evaluates what FSH3 contributes after that:

```
library("splines")
m3a <- lm(loglastE2~bs(age) + bmi + afc + afc0 + bs(logamh) + factor(lupprot), data=clean.srm)
m3b <- lm(loglastE2~bs(age) + bmi + afc + afc0 + bs(logamh) + factor(lupprot) + fsh3, data=clean.srm)
summary(m3b)
```

```
##
## Call:
## lm(formula = loglastE2 ~ bs(age) + bmi + afc + afc0 + bs(logamh) +
##     factor(lupprot) + fsh3, data = clean.srm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.15161 -0.29390  0.05363  0.34857  2.65005
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     6.4357650  0.4278927  15.041  < 2e-16 ***
## bs(age)1                        0.4630905  0.4371126   1.059 0.289581
## bs(age)2                        0.1219711  0.1661164   0.734 0.462915
## bs(age)3                        0.3915730  0.2906705   1.347 0.178148
## bmi                            -0.0072002  0.0028250  -2.549 0.010913 *
## afc                             0.0053211  0.0020832   2.554 0.010743 *
## afc0                            0.1642162  0.0459593   3.573 0.000364 ***
## bs(logamh)1                     0.0140001  0.6982855   0.020 0.984007
## bs(logamh)2                     1.4815014  0.2519312   5.881 5.07e-09 ***
## bs(logamh)3                     1.8401793  0.4757431   3.868 0.000115 ***
## factor(lupprot)LPL 10/5         0.2384490  0.0336210   7.092 2.06e-12 ***
## factor(lupprot)Lupron Microdose  0.2820577  0.0467070   6.039 1.97e-09 ***
## fsh3                           -0.0007775  0.0001946  -3.995 6.80e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.5444 on 1440 degrees of freedom
## Multiple R-squared:  0.368,  Adjusted R-squared:  0.3627
## F-statistic: 69.87 on 12 and 1440 DF,  p-value: < 2.2e-16
```

```
anova(m3a,m3b)
```

```
## Analysis of Variance Table
##
## Model 1: loglastE2 ~ bs(age) + bmi + afc + afc0 + bs(logamh) + factor(lupprot)
## Model 2: loglastE2 ~ bs(age) + bmi + afc + afc0 + bs(logamh) + factor(lupprot) +
##     fsh3
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1   1441 431.44
## 2   1440 426.71  1    4.7293 15.96 6.797e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Interpretation: FSH3 appears to contribute, after accounting for AMH and other variables, but the contribution is *much* smaller than for AMH. We can tell this because the coefficient is essentially identical to the less-flexible fit, illustrated by the nomogram.

Interaction analyses

It’s of interest to see whether FSH3 modifies the E2:AMH relationship. No modification does not mean no effect, just that the effect of FSH appears similar regardless of the value of AMH.

```
srm$fsh3cat <- cut(srm$fsh3, c(0,150,250,350,600))
srm$lupprot.f <- factor(srm$lupprot)
srm$amh.f <- cut(srm$amh, quantile(srm$amh, seq(0,1,l=5), na.rm=TRUE))
table( srm$amh.f )
```

```
##
## (0.017,0.94]   (0.94,2.2]    (2.2,4.1]   (4.1,41.4]
##          364          363          362          363
```

```
m3 <- lm(loglastE2~age + bmi + afc + afc0 + logamh*fsh3cat + lupprot.f, data=subset(srm, !is.na(amh)))
summary(m3)
```

```
##
## Call:
## lm(formula = loglastE2 ~ age + bmi + afc + afc0 + logamh * fsh3cat +
##     lupprot.f, data = subset(srm, !is.na(amh)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1918 -0.2936  0.0503  0.3501  2.5256
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.140471   0.169604  42.101  < 2e-16 ***
## age                        0.005065   0.003647   1.389 0.165174
## bmi                       -0.008382   0.002809  -2.984 0.002895 **
## afc                        0.005209   0.002071   2.516 0.011990 *
## afc0                       0.157896   0.045649   3.459 0.000558 ***
## logamh                     0.374511   0.048749   7.682 2.87e-14 ***
## fsh3cat(150,250]           0.014022   0.108876   0.129 0.897544
## fsh3cat(250,350]           0.114207   0.097482   1.172 0.241563
## fsh3cat(350,600]          -0.095019   0.094266  -1.008 0.313629
## lupprot.fLPL 10/5          0.253128   0.032883   7.698 2.56e-14 ***
## lupprot.fLupron Microdose  0.278047   0.046279   6.008 2.38e-09 ***
## logamh:fsh3cat(150,250]    0.010271   0.064314   0.160 0.873144
## logamh:fsh3cat(250,350]   -0.100209   0.060697  -1.651 0.098960 .
## logamh:fsh3cat(350,600]   -0.105572   0.054330  -1.943 0.052190 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.5416 on 1439 degrees of freedom
## Multiple R-squared:  0.3748, Adjusted R-squared:  0.3691
## F-statistic: 66.34 on 13 and 1439 DF,  p-value: < 2.2e-16
```

```
anova(m3)
```

```
## Analysis of Variance Table
```

```
##
## Response: loglastE2
##                   Df  Sum Sq Mean Sq  F value    Pr(>F)
## age                1   34.92  34.917 119.0236 < 2.2e-16 ***
## bmi                1    0.45   0.453   1.5432   0.21434
## afc                1    8.65   8.653  29.4971 6.566e-08 ***
## afc0               1   45.44  45.439 154.8916 < 2.2e-16 ***
## logamh             1  130.84 130.838 445.9970 < 2.2e-16 ***
## fsh3cat            3    8.75   2.918   9.9479 1.717e-06 ***
## lupprot.f          2   21.65  10.825  36.9014 2.352e-16 ***
## logamh:fsh3cat     3    2.31   0.771   2.6281   0.04887 *
## Residuals       1439  422.15   0.293
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
m4 <- lm(loglastE2~age + bmi + afc + afc0 + bs(logamh)*fsh3cat + lupprot.f, data=subset(srm, !is.na(amh)
summary(m4)
```

```
##
## Call:
## lm(formula = loglastE2 ~ age + bmi + afc + afc0 + bs(logamh) *
##     fsh3cat + lupprot.f, data = subset(srm, !is.na(amh)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.96737 -0.29578  0.04343  0.35145  2.55467
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      5.655203   2.052552   2.755 0.005940 **
## age                              0.005346   0.003656   1.462 0.143838
## bmi                             -0.007768   0.002824  -2.751 0.006025 **
## afc                              0.005217   0.002075   2.514 0.012030 *
## afc0                             0.161763   0.045766   3.535 0.000422 ***
## bs(logamh)1                     -0.524092   3.650465  -0.144 0.885861
## bs(logamh)2                      3.086454   1.438620   2.145 0.032087 *
## bs(logamh)3                      2.255026   2.272312   0.992 0.321174
## fsh3cat(150,250]                -1.067720   2.992849  -0.357 0.721326
## fsh3cat(250,350]                -0.178988   2.750973  -0.065 0.948132
## fsh3cat(350,600]                 0.628625   2.100827   0.299 0.764810
## lupprot.fLPL 10/5                0.240643   0.033437   7.197 9.91e-13 ***
## lupprot.fLupron Microdose        0.279813   0.046565   6.009 2.36e-09 ***
## bs(logamh)1:fsh3cat(150,250]     2.576441   5.108696   0.504 0.614111
## bs(logamh)2:fsh3cat(150,250]     0.114454   2.171746   0.053 0.957977
## bs(logamh)3:fsh3cat(150,250]     1.514180   3.334546   0.454 0.649834
## bs(logamh)1:fsh3cat(250,350]     2.505311   5.075767   0.494 0.621677
## bs(logamh)2:fsh3cat(250,350]    -1.508906   1.686949  -0.894 0.371227
## bs(logamh)3:fsh3cat(250,350]     0.833427   3.456105   0.241 0.809476
## bs(logamh)1:fsh3cat(350,600]     0.195600   3.815173   0.051 0.959118
## bs(logamh)2:fsh3cat(350,600]    -1.144828   1.548562  -0.739 0.459855
## bs(logamh)3:fsh3cat(350,600]    -1.471175   2.588029  -0.568 0.569816
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
```

```
## Residual standard error: 0.5414 on 1431 degrees of freedom
## Multiple R-squared:  0.3786, Adjusted R-squared:  0.3695
## F-statistic: 41.53 on 21 and 1431 DF,  p-value: < 2.2e-16
```

```
anova(m4)
```

```
## Analysis of Variance Table
##
## Response: loglastE2
##                      Df Sum Sq Mean Sq  F value    Pr(>F)
## age                   1  34.92  34.917 119.1035 < 2.2e-16 ***
## bmi                   1   0.45   0.453   1.5443   0.21419
## afc                   1   8.65   8.653  29.5169 6.506e-08 ***
## afc0                  1  45.44  45.439 154.9956 < 2.2e-16 ***
## bs(logamh)            3 131.73  43.911 149.7845 < 2.2e-16 ***
## fsh3cat               3   8.30   2.768   9.4419 3.531e-06 ***
## lupprot.f             2  21.44  10.720  36.5665 3.250e-16 ***
## bs(logamh):fsh3cat    9   4.71   0.523   1.7840   0.06676 .
## Residuals          1431 419.52   0.293
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
m5 <- lm(loglastE2~age + bmi + afc + afc0 + fsh3*amh.f + lupprot.f, data=subset(srm,!is.na(amh)))
summary(m5)
```

```
##
## Call:
## lm(formula = loglastE2 ~ age + bmi + afc + afc0 + fsh3 * amh.f +
##     lupprot.f, data = subset(srm, !is.na(amh)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.97317 -0.28904  0.05349  0.34637  2.63098
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                7.0467863  0.2125143  33.159  < 2e-16 ***
## age                        0.0033693  0.0037748   0.893 0.372240
## bmi                       -0.0058471  0.0028815  -2.029 0.042624 *
## afc                        0.0080278  0.0021171   3.792 0.000156 ***
## afc0                       0.2175769  0.0467521   4.654 3.56e-06 ***
## fsh3                      -0.0005598  0.0003666  -1.527 0.126928
## amh.f(0.94,2.2]            0.5229431  0.1920393   2.723 0.006545 **
## amh.f(2.2,4.1]             0.8380345  0.1760925   4.759 2.14e-06 ***
## amh.f(4.1,41.4]            0.9436549  0.1785877   5.284 1.46e-07 ***
## lupprot.fLPL 10/5          0.2251531  0.0340650   6.610 5.42e-11 ***
## lupprot.fLupron Microdose  0.2518950  0.0479928   5.249 1.76e-07 ***
## fsh3:amh.f(0.94,2.2]      -0.0004673  0.0004932  -0.947 0.343563
## fsh3:amh.f(2.2,4.1]       -0.0007345  0.0004858  -1.512 0.130761
## fsh3:amh.f(4.1,41.4]      -0.0006130  0.0005620  -1.091 0.275597
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
```

```
## Residual standard error: 0.5589 on 1438 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.3334, Adjusted R-squared:  0.3274
## F-statistic: 55.34 on 13 and 1438 DF,  p-value: < 2.2e-16
```

```
anova(m5)
```

```
## Analysis of Variance Table
##
## Response: loglastE2
##              Df Sum Sq Mean Sq  F value    Pr(>F)
## age           1  34.85  34.847 111.5593 < 2.2e-16 ***
## bmi           1   0.44   0.444   1.4213    0.2334
## afc           1   8.54   8.538  27.3345 1.964e-07 ***
## afc0          1  44.88  44.882 143.6843 < 2.2e-16 ***
## fsh3          1  66.89  66.889 214.1365 < 2.2e-16 ***
## amh.f         3  49.55  16.517  52.8764 < 2.2e-16 ***
## lupprot.f     2  18.78   9.392  30.0661 1.615e-13 ***
## fsh3:amh.f    3   0.77   0.256   0.8195    0.4831
## Residuals  1438 449.18   0.312
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
m6 <- lm(loglastE2~age + bmi + afc + afc0 + bs(fsh3)*amh.f + lupprot.f, data=subset(srm, !is.na(amh)))
summary(m6)
```

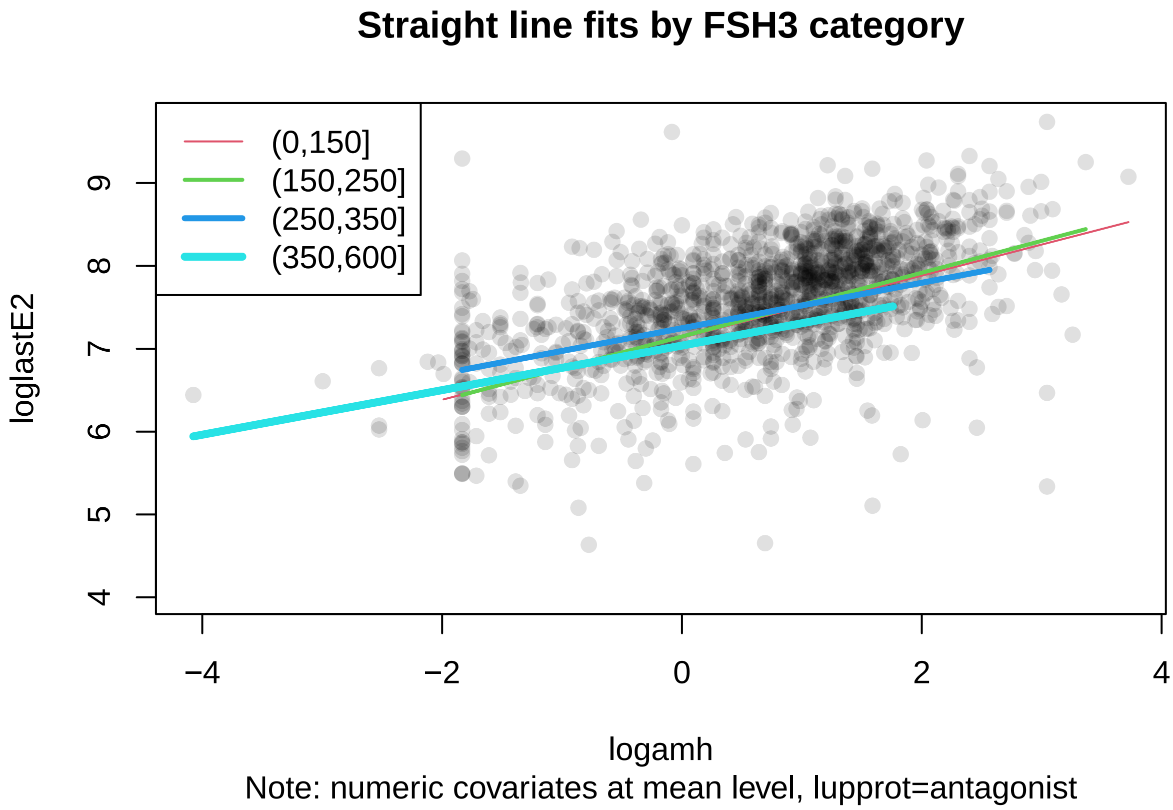
```
##
## Call:
## lm(formula = loglastE2 ~ age + bmi + afc + afc0 + bs(fsh3) *
##     amh.f + lupprot.f, data = subset(srm, !is.na(amh)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.99189 -0.29063  0.04009  0.34040  2.54447
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                5.098059   0.429500  11.870  < 2e-16 ***
## age                        0.005181   0.003779   1.371 0.170559
## bmi                       -0.006444   0.002864  -2.250 0.024608 *
## afc                        0.007469   0.002109   3.542 0.000410 ***
## afc0                       0.209986   0.046482   4.518 6.77e-06 ***
## bs(fsh3)1                  2.991387   0.777222   3.849 0.000124 ***
## bs(fsh3)2                  1.522927   0.417538   3.647 0.000274 ***
## bs(fsh3)3                  1.316812   0.493937   2.666 0.007764 **
## amh.f(0.94,2.2]            1.547508   0.990566   1.562 0.118451
## amh.f(2.2,4.1]             2.426680   0.702006   3.457 0.000563 ***
## amh.f(4.1,41.4]            2.194374   0.556744   3.941 8.49e-05 ***
## lupprot.fLPL 10/5          0.216833   0.033843   6.407 2.01e-10 ***
## lupprot.fLupron Microdose  0.267024   0.048060   5.556 3.29e-08 ***
## bs(fsh3)1:amh.f(0.94,2.2] -1.265755   1.835444  -0.690 0.490546
## bs(fsh3)2:amh.f(0.94,2.2] -1.926768   0.716870  -2.688 0.007277 **
## bs(fsh3)3:amh.f(0.94,2.2] -0.579601   1.216665  -0.476 0.633873
## bs(fsh3)1:amh.f(2.2,4.1]  -2.464358   1.530990  -1.610 0.107695
## bs(fsh3)2:amh.f(2.2,4.1]  -2.190761   0.607582  -3.606 0.000322 ***
## bs(fsh3)3:amh.f(2.2,4.1]  -1.469745   1.215199  -1.209 0.226683
## bs(fsh3)1:amh.f(4.1,41.4] -1.616319   1.318676  -1.226 0.220509
## bs(fsh3)2:amh.f(4.1,41.4] -2.111331   0.873427  -2.417 0.015761 *
## bs(fsh3)3:amh.f(4.1,41.4] -1.426415   1.575084  -0.906 0.365294
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.553 on 1430 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.351,  Adjusted R-squared:  0.3414
## F-statistic: 36.82 on 21 and 1430 DF,  p-value: < 2.2e-16
```

```
anova(m6)
```

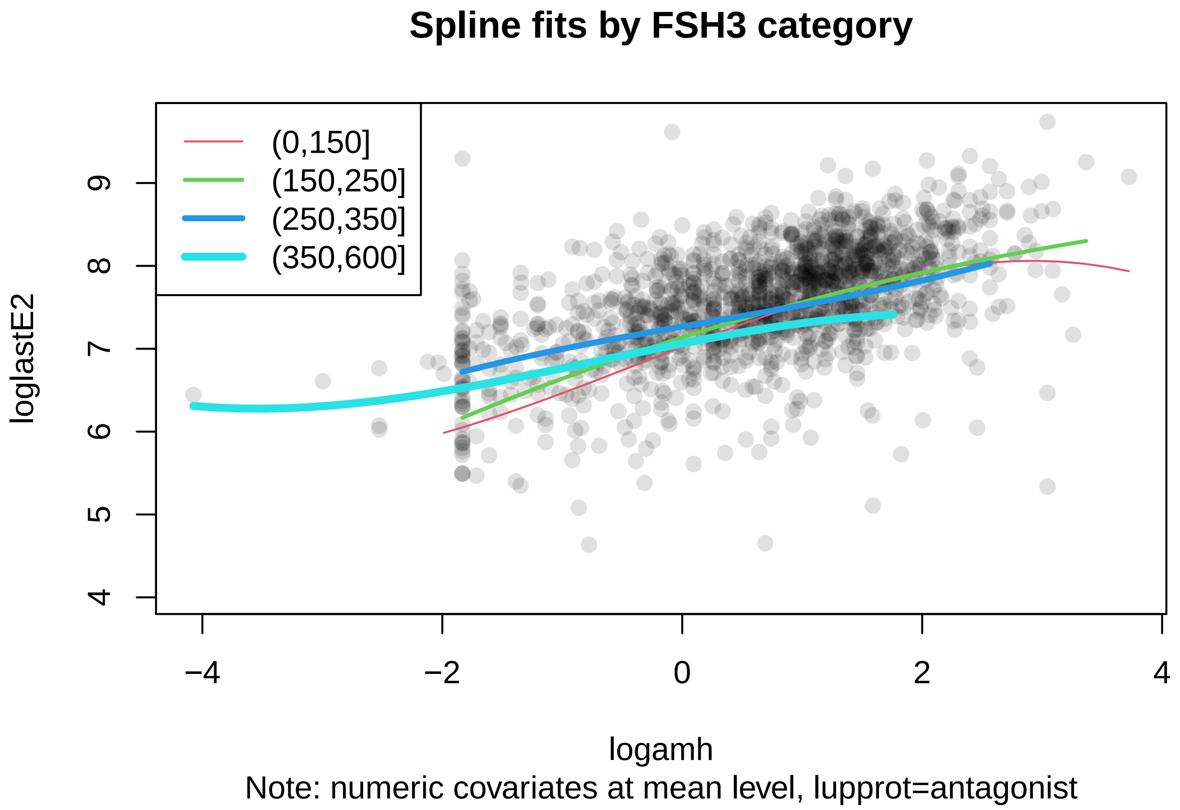
```
## Analysis of Variance Table
##
## Response: loglastE2
##                   Df Sum Sq Mean Sq  F value    Pr(>F)
## age                1  34.85  34.847 113.9312 < 2.2e-16 ***
## bmi                1   0.44   0.444   1.4516  0.228476
## afc                1   8.54   8.538  27.9156 1.464e-07 ***
## afc0               1  44.88  44.882 146.7392 < 2.2e-16 ***
## bs(fsh3)           3  77.45  25.818  84.4106 < 2.2e-16 ***
## amh.f              3  42.85  14.283  46.6967 < 2.2e-16 ***
## lupprot.f          2  19.39   9.696  31.7019 3.377e-14 ***
## bs(fsh3):amh.f     9   8.09   0.899   2.9400  0.001836 **
## Residuals       1430 437.38   0.306
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
myranges <- sapply(1:4, function(i){ range( subset(srm, fsh3cat==levels(srm$fsh3cat)[i])$logamh, na.rm=
myranges2 <- sapply(1:4, function(i){ range( subset(srm, amh.f==levels(srm$amh.f)[i])$fsh3, na.rm=TRUE)


with(srm, plot(loglastE2~ logamh, pch=19, col="#00000020"))
for(i in 1:4){
    mynewdata <- data.frame(age=mean(srm$age), bmi=mean(srm$age), afc=mean(srm$afc),
afc0=mean(srm$afc0), fsh3cat=levels(srm$fsh3cat)[i], lupprot.f="Antagonist",
    logamh=seq(myranges[1,i], myranges[2,i], l=31) )
    myfit <- predict(m3, newdata= mynewdata)
    lines(x=mynewdata$logamh, y=myfit, lwd=i, col=i+1)
}
legend("topleft", col=2:5, lwd=1:4, levels(srm$fsh3cat))
title(main="Straight line fits by FSH3 category", sub="Note: numeric covariates at mean level, lupprot=
```

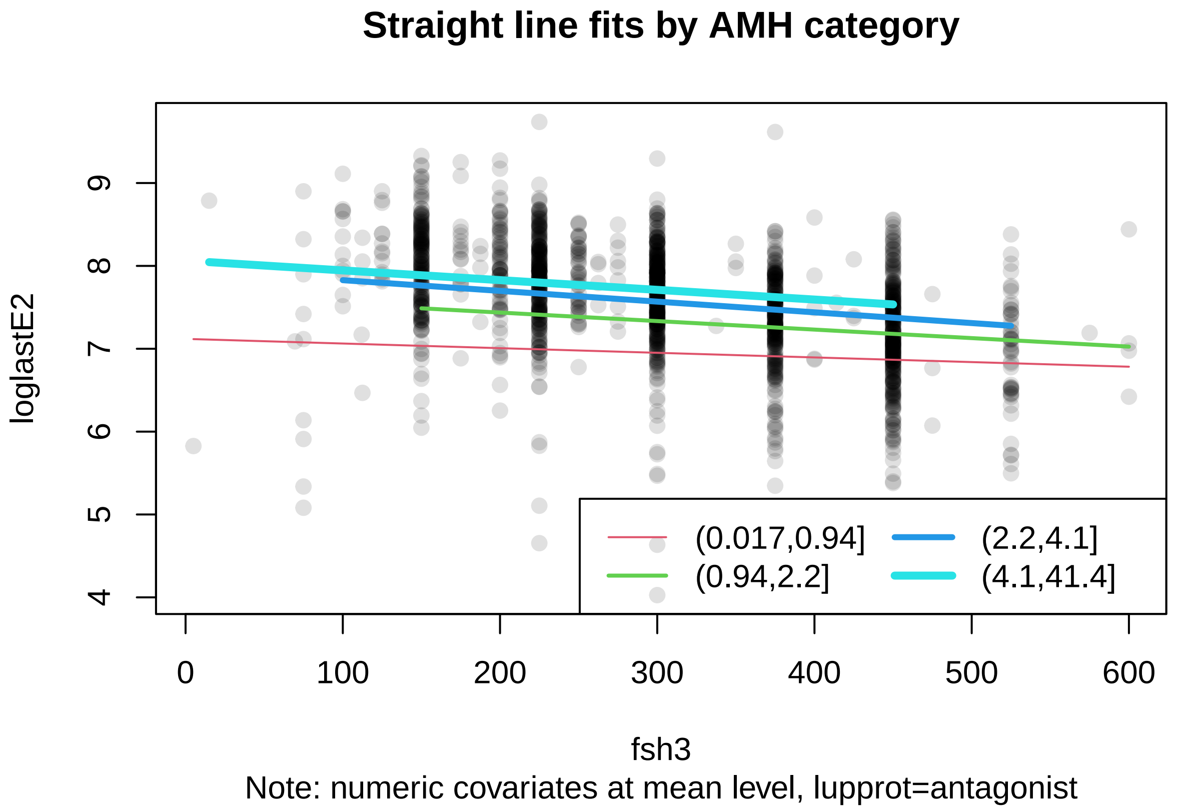


```
with(srm, plot(loglastE2~ logamh, pch=19, col="#00000020"))
for(i in 1:4){
    mynewdata <- data.frame(age=mean(srm$age), bmi=mean(srm$age), afc=mean(srm$afc),
afc0=mean(srm$afc0), fsh3cat=levels(srm$fsh3cat)[i], lupprot.f="Antagonist",
    logamh=seq(myranges[1,i], myranges[2,i], l=31) )
    myfit <- predict(m4, newdata= mynewdata)
    lines(x=mynewdata$logamh, y=myfit, lwd=i, col=i+1)
}
legend("topleft", col=2:5, lwd=1:4, levels(srm$fsh3cat))
title(main="Spline fits by FSH3 category", sub="Note: numeric covariates at mean level, lupprot=antagoni
```



Alternatively, plots of E2 vs fsh by AMH category

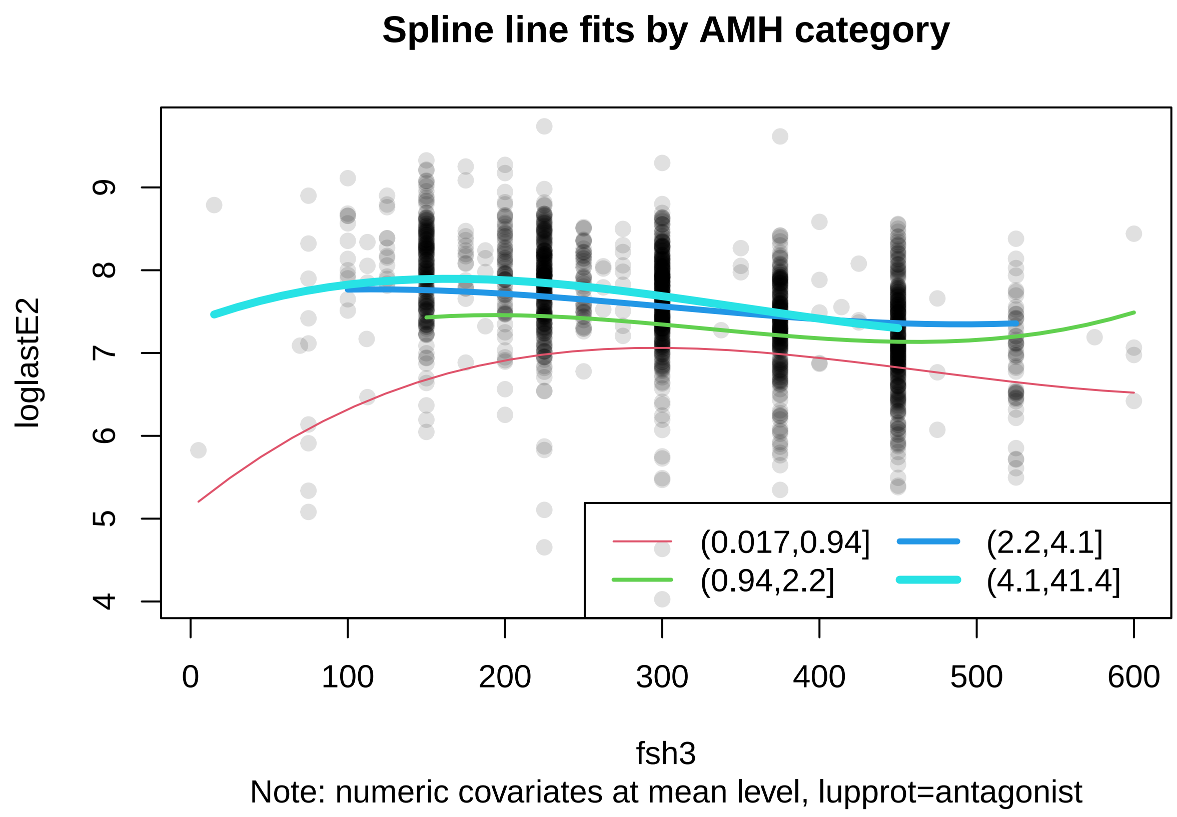
```
with(srm, plot(loglastE2~ fsh3, pch=19, col="#00000020"))
for(i in 1:4){
    mynewdata <- data.frame(age=mean(srm$age), bmi=mean(srm$age), afc=mean(srm$afc),
afc0=mean(srm$afc0), amh.f=levels(srm$amh.f)[i], lupprot.f="Antagonist",
    fsh3=seq(myranges2[1,i], myranges2[2,i], l=31) )
    myfit <- predict(m5, newdata= mynewdata)
    lines(x=mynewdata$fsh3, y=myfit, lwd=i, col=i+1)
}
legend("bottomright", col=2:5, lwd=1:4, legend=levels(srm$amh.f), ncol=2)
title(main="Straight line fits by AMH category", sub="Note: numeric covariates at mean level, lupprot=a
```



Straight line fits by AMH category

Note: numeric covariates at mean level, lupprot=antagonist

```r
with(srm, plot(loglastE2~ fsh3, pch=19, col="#00000020"))
for(i in 1:4){
    mynewdata <- data.frame(age=mean(srm$age), bmi=mean(srm$age), afc=mean(srm$afc),
afc0=mean(srm$afc0), amh.f=levels(srm$amh.f)[i], lupprot.f="Antagonist",
    fsh3=seq(myranges2[1,i], myranges2[2,i], l=31) )
    myfit <- predict(m6, newdata= mynewdata)
    lines(x=mynewdata$fsh3, y=myfit, lwd=i, col=i+1)
}
legend("bottomright", col=2:5, lwd=1:4, legend=levels(srm$amh.f), ncol=2)
title(main="Spline line fits by AMH category", sub="Note: numeric covariates at mean level, lupprot=anta
```



### Statistical learning approaches

```
#install.packages("glmnet")
library("glmnet")
```

```
## Warning: package 'glmnet' was built under R version 4.1.3

## Loading required package: Matrix

## Loaded glmnet 4.1-3
```

```
# comparison of main effects-only model with CV lasso
par(mfrow=c(1,2))
coef(summary(m1))
```

```
##                                   Estimate   Std. Error   t value
## (Intercept)                   7.3877278006 0.1546131602 47.782012
## age                           0.0053086739 0.0036550624  1.452417
## bmi                          -0.0074053399 0.0028020987 -2.642783
## afc                           0.0053402885 0.0020765772  2.571678
## afc0                          0.1632428448 0.0457989484  3.564336
## log(amh)                      0.3022195266 0.0185805167 16.265399
## factor(lupprot)LPL 10/5       0.2407535807 0.0322916426  7.455600
## factor(lupprot)Lupron Microdose  0.2767188357 0.0464490621  5.957469
## fsh3                         -0.0007994601 0.0001886598 -4.237576
##                                      Pr(>|t|)
## (Intercept)                      1.271152e-299
## age                               1.466030e-01
## bmi                               8.311681e-03
## afc                               1.021997e-02
## afc0                              3.766614e-04
## log(amh)                          9.405918e-55
## factor(lupprot)LPL 10/5           1.535637e-13
## factor(lupprot)Lupron Microdose   3.213183e-09
## fsh3                              2.402604e-05
```

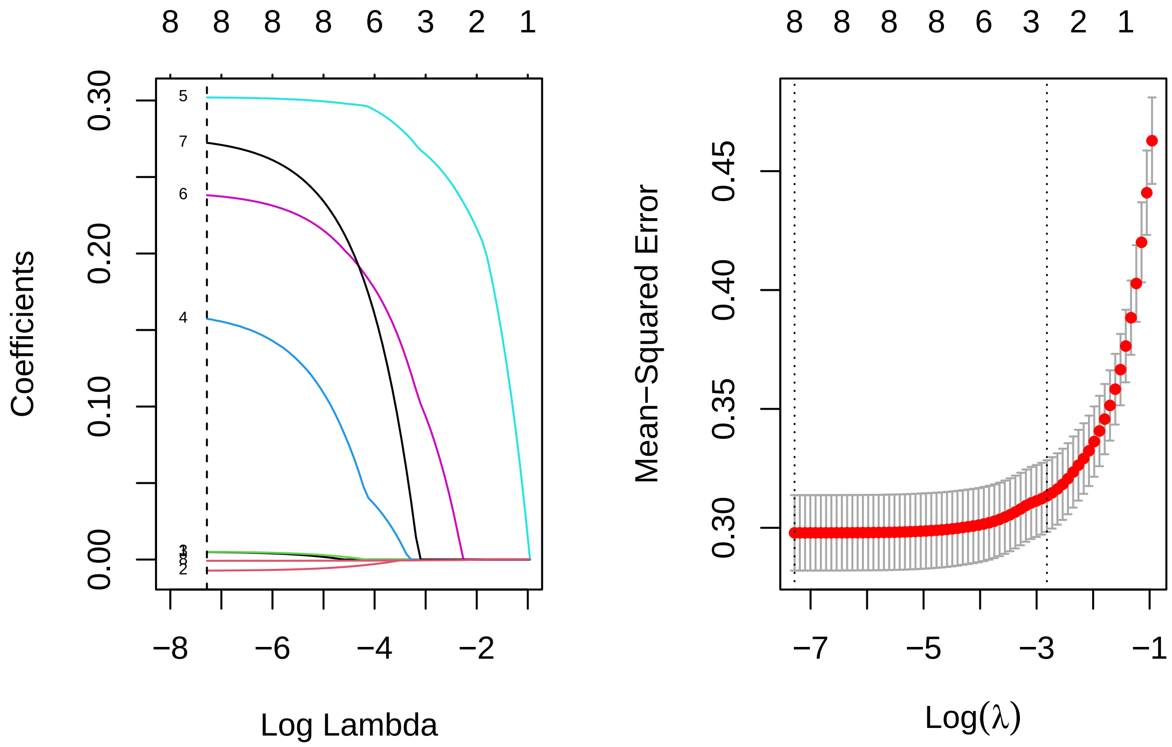
```
plot(glmnet(x=model.matrix(m1)[,-1], y=m1$model[,1]),
xvar="lambda", label=TRUE, xlim=c(-8,-1))
set.seed(4)
cv.glmnet1 <- cv.glmnet(x=model.matrix(m1)[,-1], y=m1$model[,1])
print(cv.glmnet1 )
```

```
##
## Call:  cv.glmnet(x = model.matrix(m1)[, -1], y = m1$model[, 1])
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min 0.00069    69  0.2979 0.01588       8
## 1se 0.05974    21  0.3134 0.01506       3
```

```
tail( anova(m1)[,"Mean Sq"], 1)
```

```
## [1] 0.2959667
```

```
abline(v=log(cv.glmnet1$lambda.min), lty=2)
plot(cv.glmnet1 )
```



```
cv.glmnet1$lambda.min
```

```
## [1] 0.0006869018
```

```
log(cv.glmnet1$lambda.min)
```

```
## [1] -7.283319
```

```
cbind( coef(m1), coef(cv.glmnet1, s = "lambda.min"))
```

```
## 9 x 2 sparse Matrix of class "dgCMatrix"
##                                                       s1
## (Intercept)                     7.3877278006  7.401218144
## age                             0.0053086739  0.004932602
## bmi                            -0.0074053399 -0.007228421
## afc                             0.0053402885  0.005073427
## afc0                            0.1632428448  0.157319768
## log(amh)                        0.3022195266  0.302008222
## factor(lupprot)LPL 10/5         0.2407535807  0.238101183
## factor(lupprot)Lupron Microdose 0.2767188357  0.272357434
## fsh3                           -0.0007994601 -0.000793687
```

Interpretation: the lasso approach can potentially achieve better prediction of logE2 values, by shrinking the "classical" estimates towards zero in a way suggested by the patterns in the data. This makes them more stable, albeit at the cost of some bias. Cross-validation is used to choose the apparently-best degree of shrinkage, i.e. the best tradeoff. But for this large dataset with clear signals, it seems we do best not shrinking at all.

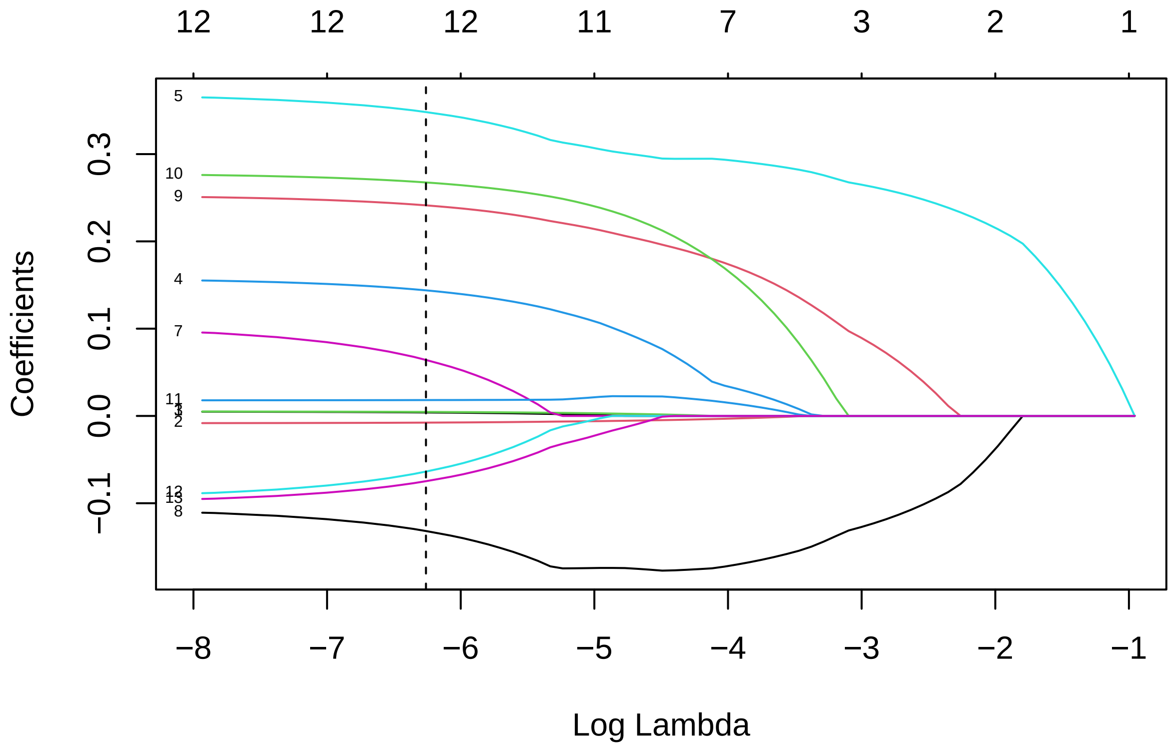
Also note how, if we were to shrink the coefficients anyway, AMH is the last one to be shrunk, emphasizing what we saw in the other analyses.

Trying the same approach for the more flexible representation of FSH3, and its interaction with AMH, we again see that all the AMH terms persist best under shrinkage. Also, lasso's degree of improvement in prediction (proportion of variance explained, known as R^2) is very minor, when optimized via cross-validation.

```
# comparison of main effects-only model with CV lasso
coef(summary(m3))
```

```
##                                Estimate  Std. Error    t value      Pr(>|t|)
## (Intercept)                 7.140471233 0.169604233 42.1007843 4.002870e-253
## age                         0.005064774 0.003647437  1.3885845  1.651740e-01
## bmi                        -0.008382413 0.002809363 -2.9837411  2.895455e-03
## afc                         0.005209434 0.002070826  2.5156312  1.199010e-02
## afc0                        0.157895915 0.045648977  3.4589146  5.581865e-04
## logamh                      0.374510674 0.048748563  7.6824967  2.868745e-14
## fsh3cat(150,250]            0.014021878 0.108876214  0.1287873  8.975439e-01
## fsh3cat(250,350]            0.114207402 0.097482295  1.1715707  2.415634e-01
## fsh3cat(350,600]           -0.095019165 0.094265940 -1.0079904  3.136285e-01
## lupprot.fLPL 10/5           0.253128486 0.032882993  7.6978541  2.556347e-14
## lupprot.fLupron Microdose   0.278046723 0.046279429  6.0079982  2.375589e-09
## logamh:fsh3cat(150,250]     0.010270639 0.064314348  0.1596944  8.731442e-01
## logamh:fsh3cat(250,350]    -0.100209229 0.060696613 -1.6509855  9.895974e-02
## logamh:fsh3cat(350,600]    -0.105572381 0.054329861 -1.9431742  5.218991e-02
```

```
#plot(glmnet(x=model.matrix(m3)[,-1], y=m3$model[,1]))
plot(glmnet(x=model.matrix(m3)[,-1], y=m3$model[,1]),
xvar="lambda", label=TRUE, xlim=c(-8,-1))
set.seed(4)
cv.glmnet3 <- cv.glmnet(x=model.matrix(m3)[,-1], y=m3$model[,1])
abline(v=log(cv.glmnet3$lambda.min), lty=2)
```



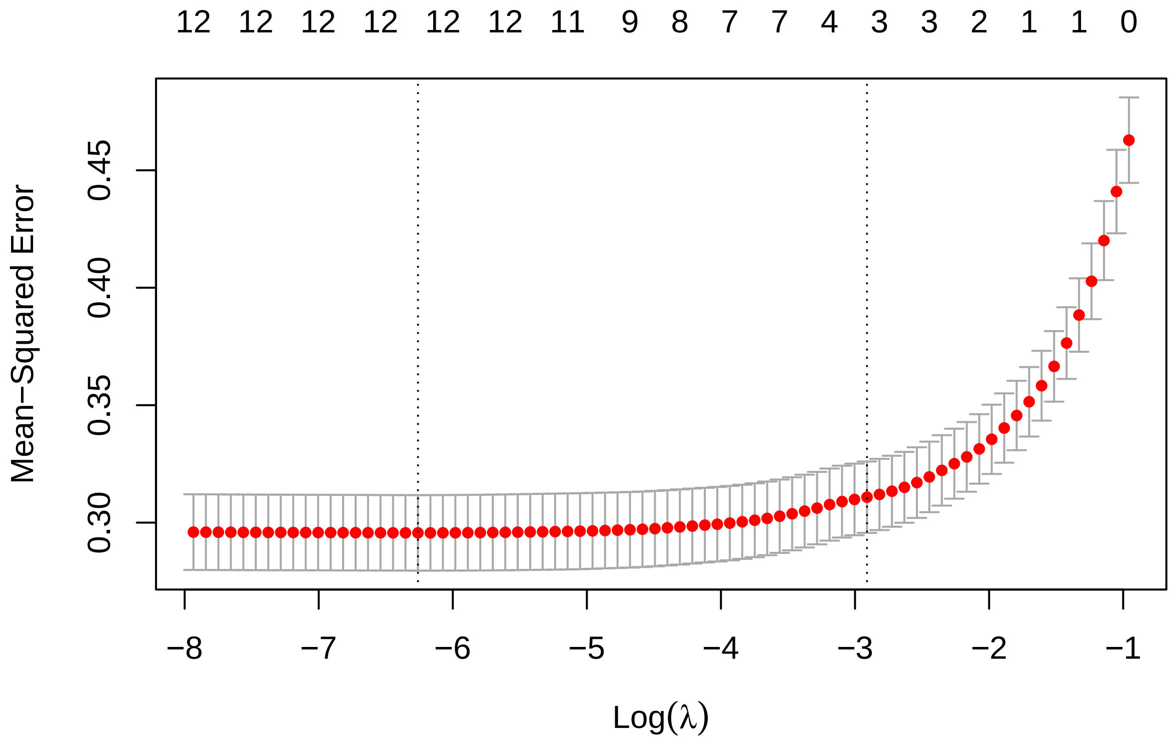
```
print(cv.glmnet3 )
```

```
##
## Call:  cv.glmnet(x = model.matrix(m3)[, -1], y = m3$model[, 1])
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min 0.00191    58  0.2956 0.01607      12
## 1se 0.05444    22  0.3108 0.01520       3
```

```
tail( anova(m3)[,"Mean Sq"], 1)
```

```
## [1] 0.2933611
```

```
plot(cv.glmnet3 )
```



```r
cv.glmnet3$lambda.min
```

```
## [1] 0.001911345
```

```r
log(cv.glmnet3$lambda.min)
```

```
## [1] -6.259948
```

```r
cbind( coef(m3), coef(cv.glmnet3, s = "lambda.min"))
```

```
## 14 x 2 sparse Matrix of class "dgCMatrix"
##                                                s1
## (Intercept)                 7.140471233  7.216584037
## age                         0.005064774  0.003991792
## bmi                        -0.008382413 -0.007649679
## afc                         0.005209434  0.004595668
## afc0                        0.157895915  0.143902048
## logamh                      0.374510674  0.348153586
## fsh3cat(150,250]            0.014021878  .
## fsh3cat(250,350]            0.114207402  0.064152956
## fsh3cat(350,600]           -0.095019165 -0.131888391
## lupprot.fLPL 10/5           0.253128486  0.241275748
## lupprot.fLupron Microdose   0.278046723  0.267527001
## logamh:fsh3cat(150,250]     0.010270639  0.018179282
## logamh:fsh3cat(250,350]   -0.100209229 -0.063734637
## logamh:fsh3cat(350,600]   -0.105572381 -0.074749145
```