

SRM analysis

Makayla

2022-06-02

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")
data <- read_excel("Copy of 2013 2014 for data entry (2).xlsx", sheet=1)
#names(srm)
dLETTERS <- sapply(1:26, function(i){paste(LETTERS[i], LETTERS[i], sep="")})
cbind(c(LETTERS, dLETTERS)[1:38], names(data))
```

```
## Warning in cbind(c(LETTERS, dLETTERS)[1:38], names(data)): number of rows of
## result is not a multiple of vector length (arg 1)
```

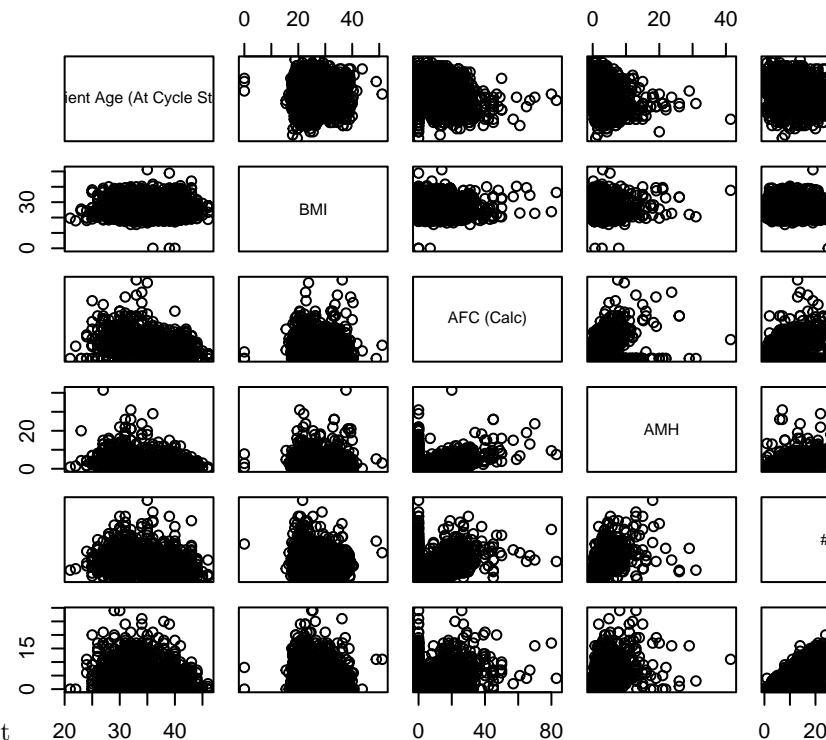
```
##      [,1] [,2]
## [1,] "A"  "MPI"
## [2,] "B"  "Patient Age (At Cycle Start)"
## [3,] "C"  "Treatment #"
## [4,] "D"  "BMI"
## [5,] "E"  "AFC (Calc)"
## [6,] "F"  "AMH"
## [7,] "G"  "Lupron Protocol"
## [8,] "H"  "E2 Day 3"
## [9,] "I"  "FSH Dose Day 3"
## [10,] "J"  "E2 Day 4"
## [11,] "K"  "FSH 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"
## [39,] "A" "#MII"
## [40,] "B" "Number Fertilized (ICSI)"
## [41,] "C" "Number Fertilized (IVF)"
## [42,] "D" "Blast Conversion"
## [43,] "E" "Total Blast Conversion"
## [44,] "F" "PGD"
## [45,] "G" "No. of embryos biopsied"
## [46,] "H" "No. of embryos diagnosed"
```

Some simple numeric summaries of some variables of interest

```
# variables of interest
summary(data[, c(2,4,5, 6, 39, 43)])
```

```
## 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
##      #MII      Total Blast Conversion
## Min. : 1.0      Min. : 0.000
## 1st Qu.: 6.0      1st Qu.: 0.000
## Median :10.0      Median : 2.000
## Mean :11.3      Mean : 3.631
## 3rd Qu.:15.0      3rd Qu.: 5.000
## Max. :55.0      Max. :29.000
## NA's :660      NA's :46
```



Simple pairwise scatterplots of some variables of interest

Examining protocol

```
## Lupron Protocol
##      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(data[,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(data[,9]) # 6 QD
```

```
## FSH Dose Day 3
##      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(data[,11]) # 2 QD

## FSH Dose Day 4
##      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(data[,13]) # 4 QD
```

```
## FSH Dose Day 5
##      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
```

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

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

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

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

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

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

Examining estradiol:

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

```
## [1] "E2 Day 9" "E2 Day 10" "E2 Day 11" "E2 Day 12"
summary(data[,c(20,22,24,26)])
```

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

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

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

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

```
# [1666,20]
data[1666,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> <chr>
## 1 > 3000      NA        NA <NA>
```

```
data[1666,20] <- NA
data[,20] <- as.numeric(unlist(data[,20]))
```

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

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

```
# first omit 338 (!) observations with no E2 at all
table(apply(data[,c(20,22,24,26)], 1, function(x){sum(is.na(x))}))
```

```
##
## 0 1 2 3 4
```

```
## 3 99 634 778 338
na.counts <- apply(data[,c(20,22,24,26)], 1, function(x){sum(is.na(x))})
data <- data[na.counts != 4,]
## dim(data)
data <- as.data.frame(data)
data$lastE2 <- data[,20]
table(is.na(data$lastE2))

##
## FALSE TRUE
## 489 1025

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

##
## FALSE TRUE
## 1225 289

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

##
## FALSE TRUE
## 1510 4

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

##
## FALSE
## 1514

data$lastE2 <- as.numeric(unlist(data$lastE2))

## Warning: NAs introduced by coercion
Removing 2 BMIs of zero:
data <- data[data$BMI>0,]
## dim(data)

Remove a single "Lupron" only lupron protocol also single "Unstimulated"
data <- subset(data, data[,7]!="Lupron")
data <- subset(data, data[,7]!="Unstimulated")
dim(data)

## [1] 1510 50

Remove a single FSH3 value in excess of 3000
data <- subset(data, fsh3 < 3000)

Fill in missing #MII values with Total Blast Conversion, if available
data$`#MII`[is.na(data$`#MII`)] <- data$`Total Blast Conversion`[is.na(data$`#MII`)]

Removing Total Blast Conversion is NA
data <- subset(data, !is.na(data[,43]))
```

#MII should be \geq Total Blast Conversion

```
table(data$`#MII` >= data$`Total Blast Conversion`)      # There are 2
```

```
##
```

```
## FALSE TRUE
```

```
##      2 1451
```

Remove #MII < Total Blast Conversion

```
data <- subset(data, (data$`#MII` >= data$`Total Blast Conversion`)==TRUE)
```

Remove lastE2 is NA

```
data <- subset(data, !is.na(lastE2))
```

```
dim(data)      # Final data after data cleaning
```

```
## [1] 1450  50
```

Our parameters of interest are: age, BMI; antral follicle count (AFC), AMH, #MII (Total number of mature oocytes) and Total Blast Conversion listed in columns B, D, E, F, A, and E respectively

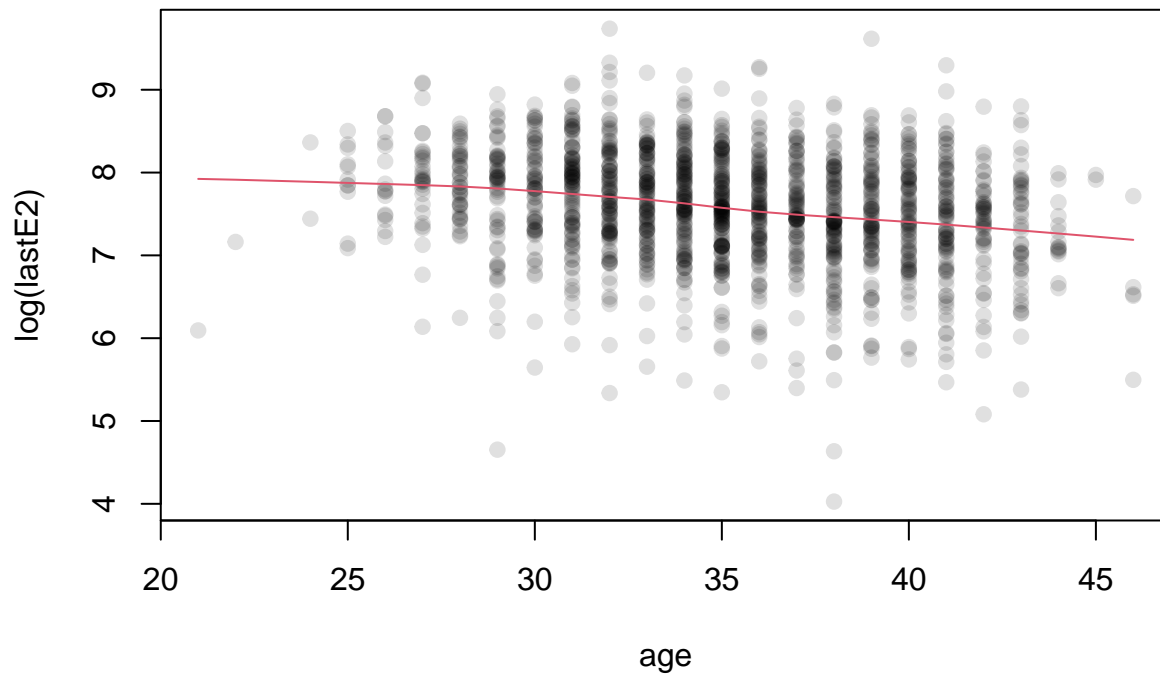
Three “protocols” are used. microdose Lupron or MDL; long lupron or LL and antagonist.

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

```
data$age <- data[,2]
data$bmi <- data$BMI
data$lupprot <- data[,7]
data$amh <- data$AMH
data$afc <- data[,5]
data$mii <- data[,39]
data$Total_bc <- data[,43]
```

Data summary

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

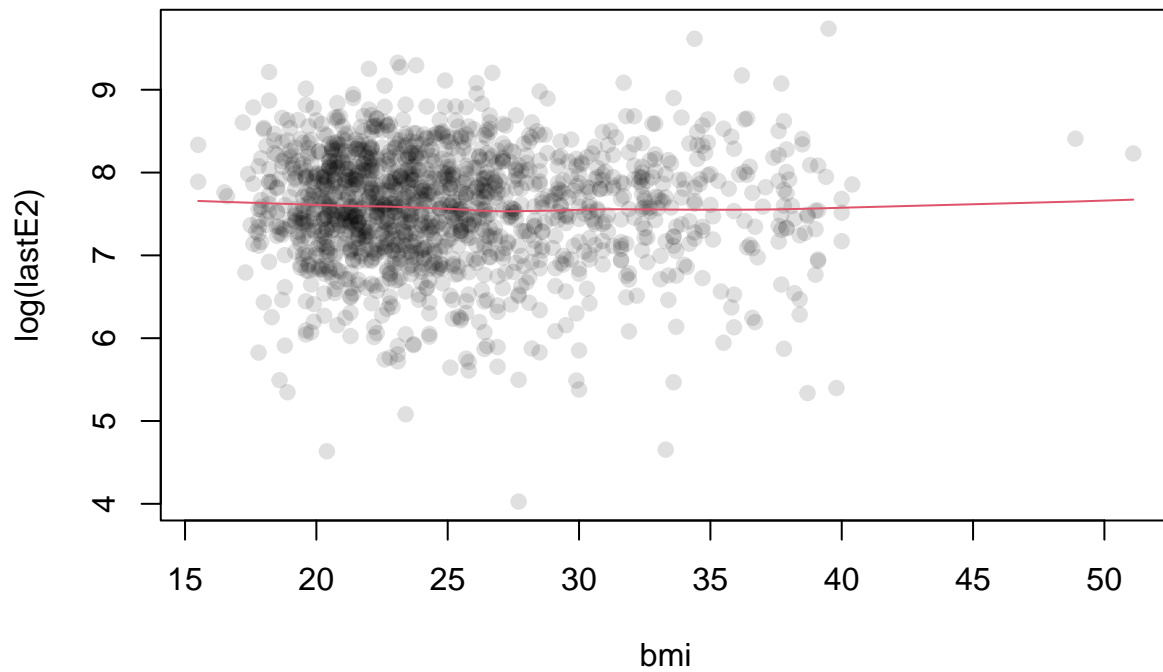


```
plot(log(lastE2)~bmi, data=data, pch=19, col="#00000020")
lines(lowess(x=data$bmi, y=log(data$lastE2), iter=0), col=2)
```

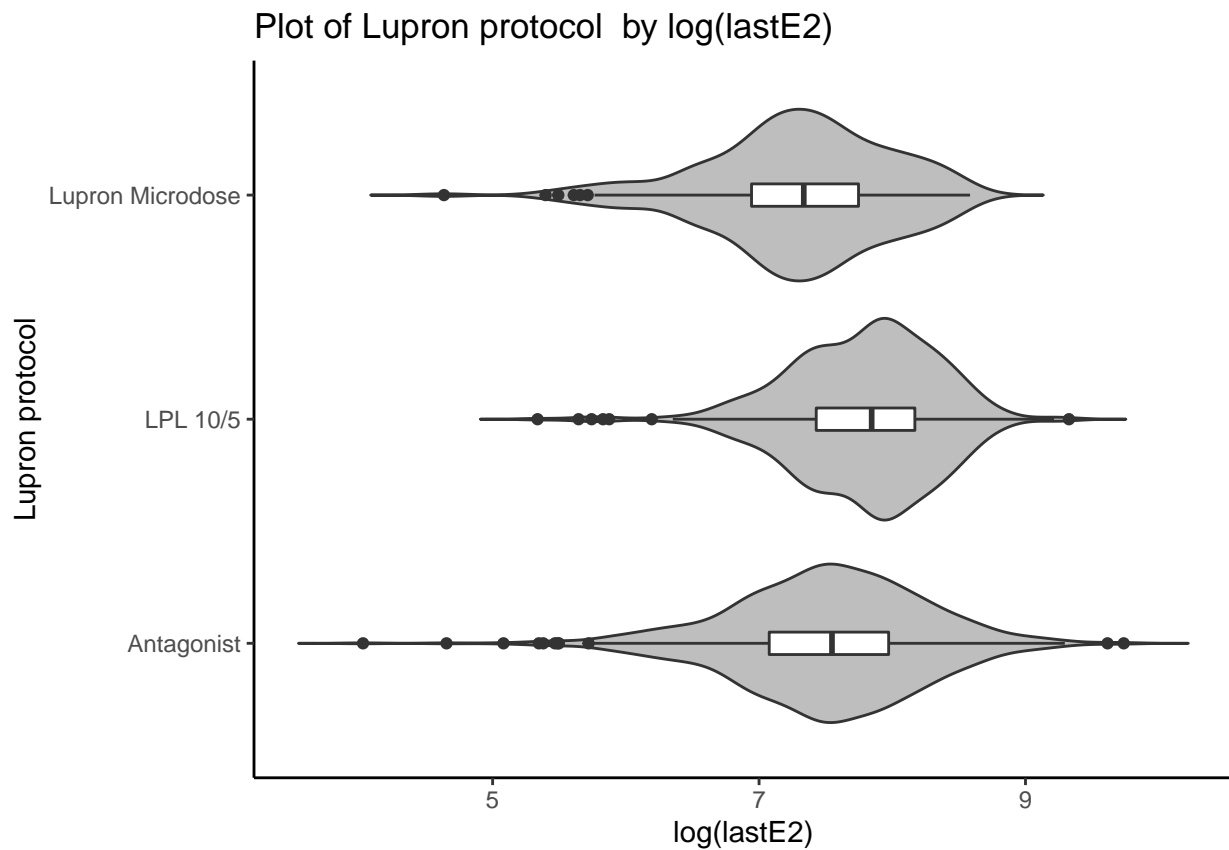
```
#table(data$lupprot)
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
## v ggplot2 3.3.6      v purrr  0.3.4
## v tibble  3.1.7      v dplyr  1.0.9
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.1
```

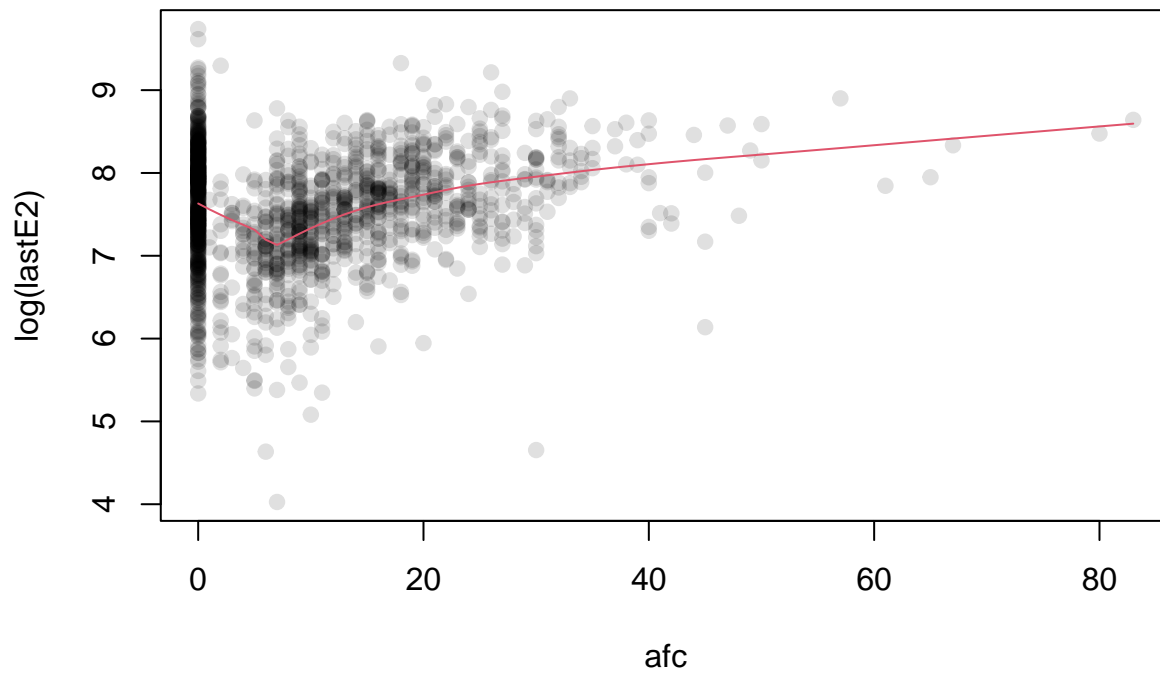
```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

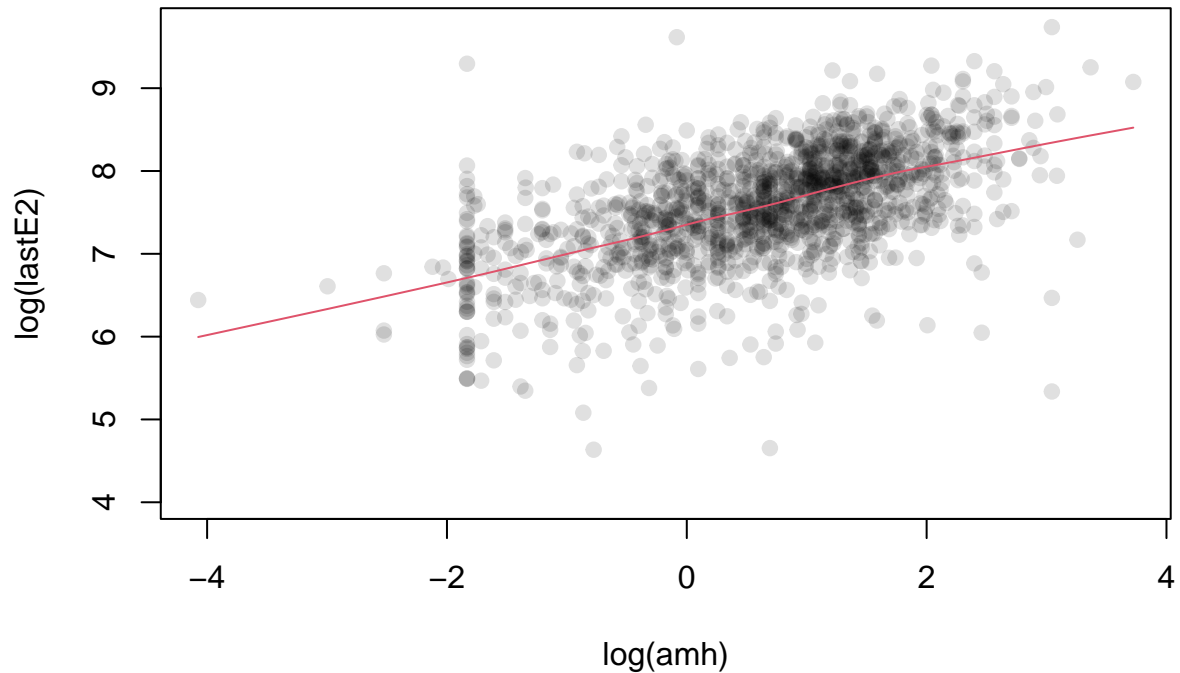
```
ggplot(data, aes(x=log(lastE2), y=lupprot)) +
  geom_violin(trim=FALSE, fill="gray")+
  labs(title="Plot of Lupron protocol by log(lastE2)", x="log(lastE2)", y = "Lupron protocol")+
  geom_boxplot(width=0.1)+
  theme_classic()
```



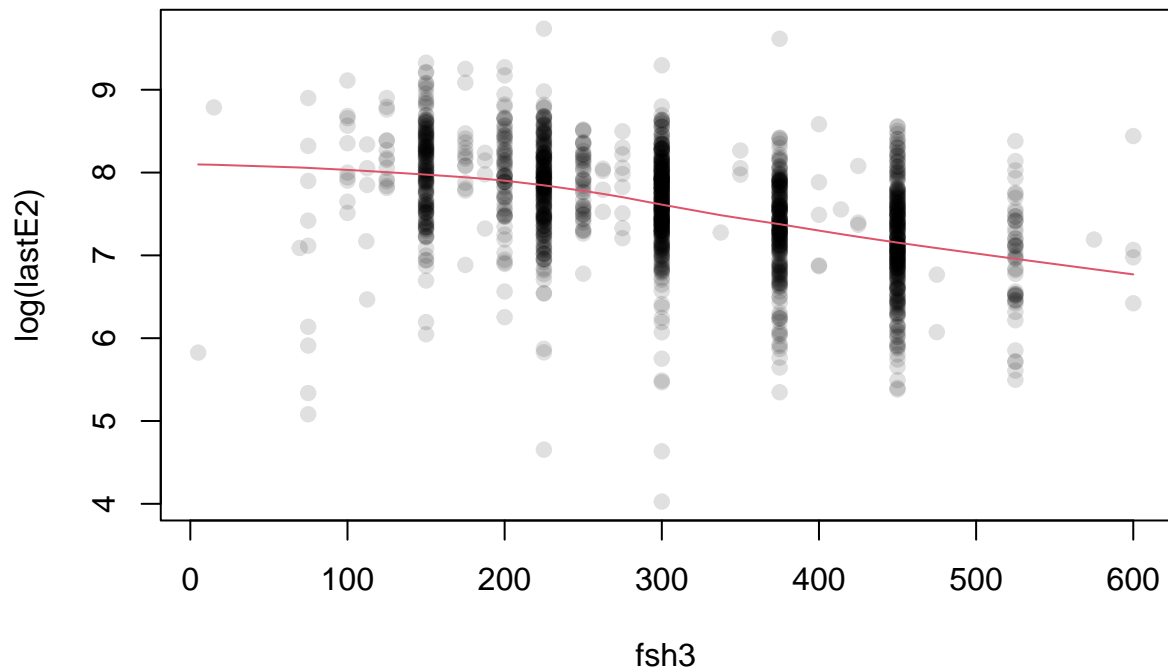
```
plot(log(lastE2)~afc, data=data, pch=19, col="#00000020")
with(subset(data), lines(lowess(x=afc, y=log(lastE2), iter=0), col=2))
```



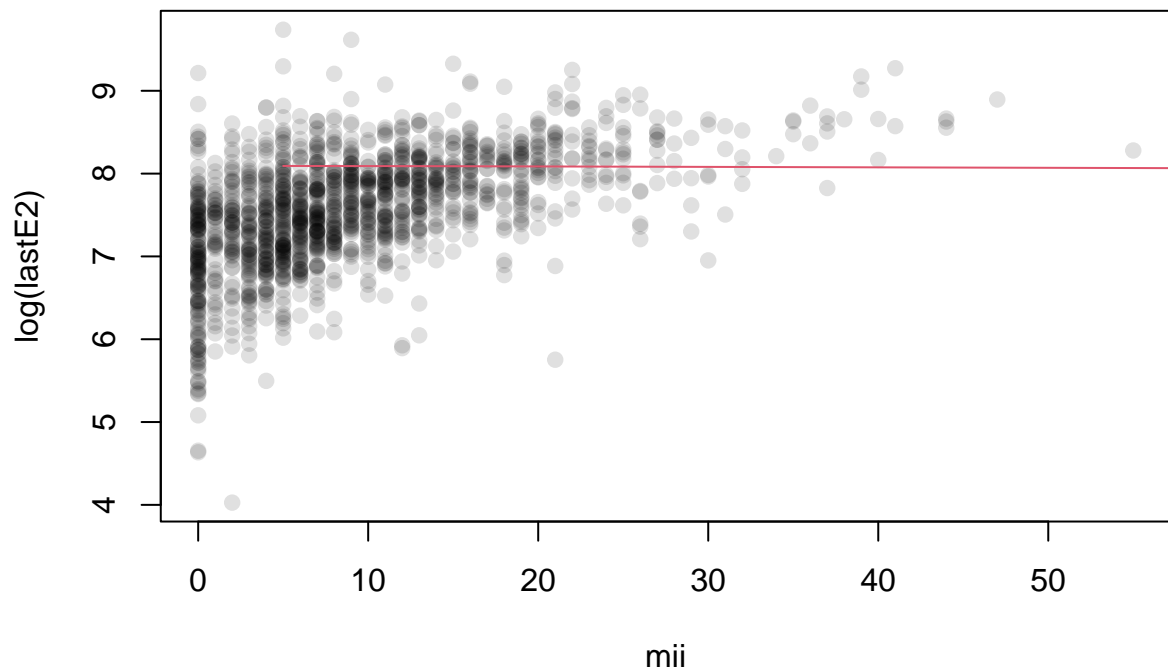
```
plot(log(lastE2)~log(amh), data=data, pch=19, col="#00000020")
with(subset(data, !is.na(amh)), lines(lowess(x=log(amh), y=log(lastE2), iter=0), col=2))
```



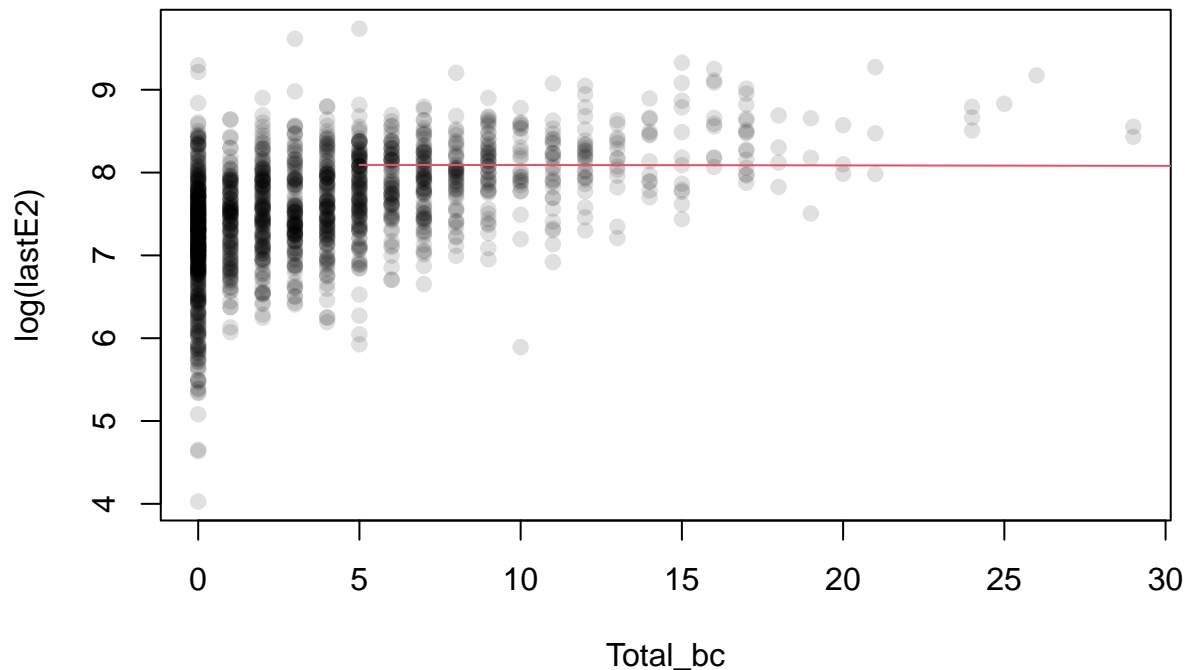
```
plot(log(lastE2)~fsh3, data=data, pch=19, col="#00000020")
with(subset(data, !is.na(amh)), lines(lowess(x=fsh3, y=log(lastE2), iter=0), col=2))
```



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



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



Correlations between pairs of variables

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

##	age	bmi	afc	logamh	lastE2	fsh3	mii	Total_bc
## age	1.000	0.028	-0.191	-0.354	-0.211	0.461	-0.245	-0.280
## bmi	0.028	1.000	0.036	0.038	0.009	0.011	0.007	0.005
## afc	-0.191	0.036	1.000	0.246	0.124	-0.282	0.197	0.079
## logamh	-0.354	0.038	0.246	1.000	0.508	-0.668	0.519	0.460
## lastE2	-0.211	0.009	0.124	0.508	1.000	-0.439	0.505	0.504
## fsh3	0.461	0.011	-0.282	-0.668	-0.439	1.000	-0.438	-0.415
## mii	-0.245	0.007	0.197	0.519	0.505	-0.438	1.000	0.702
## Total_bc	-0.280	0.005	0.079	0.460	0.504	-0.415	0.702	1.000

Statistical analysis

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

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

```
##
##      Antagonist      LPL 10/5 Lupron Microdose
##           722           508           220
```

```
data$lup.lpl05<- ifelse(data$lupprot=="LPL 10/5", 1, 0)
data$lup.lpmic<- ifelse(data$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.data <- subset(data, !is.na(amh))
m1 <- lm(log(lastE2)~age + bmi + afc + afc0 + log(amh) + factor(lupprot) +fsh3, data=clean.data)
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)	7.370000	7.06000	7.670000	2.46e-292
## age	0.004290	-0.00289	0.011500	2.41e-01
## bmi	-0.006520	-0.01200	-0.001010	2.05e-02
## afc	0.007600	0.00342	0.011800	3.77e-04
## afc0	0.198000	0.10800	0.289000	1.85e-05
## log(amh)	0.297000	0.26100	0.334000	1.60e-53
## factor(lupprot)LPL 10/5	0.229000	0.16500	0.292000	2.67e-12
## factor(lupprot)Lupron Microdose	0.269000	0.17900	0.360000	6.03e-09
## fsh3	-0.000748	-0.00112	-0.000379	7.34e-05

Turn this into a nomogram:

```

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##   src, summarize

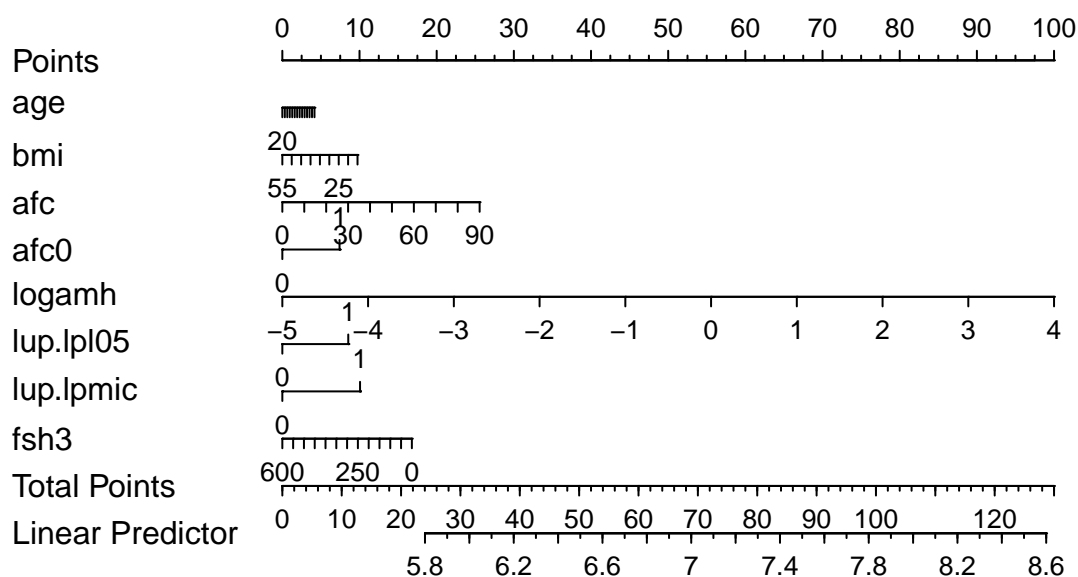
## The following objects are masked from 'package:base':
##
##   format.pval, units

##
## Attaching package: 'SparseM'

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

```

Nomogram, predicting last E2 value (Rsquared=0.367)



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

```
##
## Call:
## lm(formula = loglastE2 ~ bs(age) + bmi + afc + afc0 + bs(logamh) +
##     factor(lupprot) + fsh3, data = clean.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.13203 -0.29452  0.04933  0.34407  2.66486
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.4586239   0.4321844   14.944 < 2e-16 ***
## bs(age)1       0.3429520   0.4604570    0.745 0.456513
## bs(age)2       0.1020228   0.1697126    0.601 0.547837
## bs(age)3       0.2960844   0.3044791    0.972 0.331005
## bmi           -0.0062486   0.0028369   -2.203 0.027785 *
## afc            0.0076014   0.0021376    3.556 0.000389 ***
## afc0          0.1993294   0.0462933    4.306 1.78e-05 ***
## bs(logamh)1   -0.0059009   0.6956270   -0.008 0.993233
## bs(logamh)2    1.4758718   0.2506378    5.888 4.87e-09 ***
## bs(logamh)3    1.7884856   0.4760884    3.757 0.000179 ***
## factor(lupprot)LPL 10/5    0.2258600   0.0337382    6.694 3.12e-11 ***
```

```
## factor(lupprot)Lupron Microdose 0.2747260 0.0463061 5.933 3.75e-09 ***
## fsh3 -0.0007304 0.0001943 -3.758 0.000178 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5372 on 1401 degrees of freedom
## Multiple R-squared: 0.375, Adjusted R-squared: 0.3697
## F-statistic: 70.06 on 12 and 1401 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 1402 408.36
## 2 1401 404.29 1 4.0763 14.126 0.000178 ***
## ---
## 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 - Linear Regression

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.

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

##
## (0.017,0.92] (0.92,2.15] (2.15,4] (4,41.4]
## 355 352 353 353
m3 <- lm(loglastE2~age + bmi + afc + afc0 + logamh*fsh3cat + lupprot.f, data=subset(data, !is.na(amh)))
summary(m3)

##
## Call:
## lm(formula = loglastE2 ~ age + bmi + afc + afc0 + logamh * fsh3cat +
## lupprot.f, data = subset(data, !is.na(amh)))
##
## Residuals:
## Min 1Q Median 3Q Max
## -3.14755 -0.29680 0.04248 0.34734 2.56025
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.154043 0.170369 41.991 < 2e-16 ***
## age 0.003956 0.003648 1.084 0.278346
## bmi -0.007468 0.002817 -2.651 0.008116 **
```

```
## afc                0.007648    0.002126    3.597 0.000333 ***
## afc0               0.195833    0.045994    4.258 2.20e-05 ***
## logamh             0.347124    0.048819    7.110 1.84e-12 ***
## fsh3cat(150,250]   -0.027256    0.108758   -0.251 0.802152
## fsh3cat(250,350]    0.091714    0.097375    0.942 0.346427
## fsh3cat(350,600]   -0.111060    0.094347   -1.177 0.239337
## lupprot.fLPL 10/5   0.240819    0.032967    7.305 4.64e-13 ***
## lupprot.fLupron Microdose 0.273252    0.045876    5.956 3.26e-09 ***
## logamh:fsh3cat(150,250] 0.044834    0.064445    0.696 0.486737
## logamh:fsh3cat(250,350] -0.068549    0.060497   -1.133 0.257363
## logamh:fsh3cat(350,600] -0.084239    0.054290   -1.552 0.120973
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5343 on 1400 degrees of freedom
## Multiple R-squared:  0.3823, Adjusted R-squared:  0.3765
## F-statistic: 66.65 on 13 and 1400 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  35.33   35.334 123.7943 < 2.2e-16 ***
## bmi           1   0.23    0.227   0.7936  0.37316
## afc           1  10.34   10.342  36.2327 2.233e-09 ***
## afc0          1  49.82   49.823 174.5591 < 2.2e-16 ***
## logamh        1 121.79  121.791 426.7043 < 2.2e-16 ***
## fsh3cat       3   8.07    2.689   9.4219 3.642e-06 ***
## lupprot.f     2  19.52    9.762  34.2028 3.143e-15 ***
## logamh:fsh3cat 3   2.18    0.728   2.5496  0.05429 .
## Residuals    1400 399.59    0.285
```

```
## ---
```

```
## 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(data, !is.na(amh)))
summary(m4)
```

```
##
```

```
## Call:
```

```
## lm(formula = loglastE2 ~ age + bmi + afc + afc0 + bs(logamh) *
##     fsh3cat + lupprot.f, data = subset(data, !is.na(amh)))
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.90186 -0.29417  0.04245  0.34270  2.57680
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.395956    2.041065    2.644 0.008293 **
## age            0.004393    0.003656    1.202 0.229685
## bmi           -0.006851    0.002834   -2.418 0.015753 *
## afc            0.007581    0.002127    3.564 0.000378 ***
## afc0           0.198355    0.046041    4.308 1.76e-05 ***
```



```
## bs(logamh)1          -0.050602    3.639390   -0.014 0.988909
## bs(logamh)2           3.327238    1.425955    2.333 0.019772 *
## bs(logamh)3           2.391679    2.262717    1.057 0.290697
## fsh3cat(150,250]      -1.460252    2.992351   -0.488 0.625630
## fsh3cat(250,350]      -0.105330    2.727391   -0.039 0.969199
## fsh3cat(350,600]       0.907774    2.088705    0.435 0.663912
## lupprot.fLPL 10/5      0.226578    0.033571    6.749 2.18e-11 ***
## lupprot.fLupron Microdose 0.273836    0.046138    5.935 3.70e-09 ***
## bs(logamh)1:fsh3cat(150,250] 3.385471    5.132199    0.660 0.509586
## bs(logamh)2:fsh3cat(150,250] 0.053818    2.151782    0.025 0.980050
## bs(logamh)3:fsh3cat(150,250] 2.248333    3.351308    0.671 0.502407
## bs(logamh)1:fsh3cat(250,350] 2.443518    5.037385    0.485 0.627698
## bs(logamh)2:fsh3cat(250,350] -1.826586    1.670233   -1.094 0.274315
## bs(logamh)3:fsh3cat(250,350] 1.088598    3.425294    0.318 0.750676
## bs(logamh)1:fsh3cat(350,600] -0.315495    3.800094   -0.083 0.933845
## bs(logamh)2:fsh3cat(350,600] -1.380538    1.534707   -0.900 0.368518
## bs(logamh)3:fsh3cat(350,600] -1.704504    2.571826   -0.663 0.507594
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5338 on 1392 degrees of freedom
## Multiple R-squared:  0.3869, Adjusted R-squared:  0.3776
## F-statistic: 41.82 on 21 and 1392 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	35.33	35.334	124.0058	< 2.2e-16 ***
bmi	1	0.23	0.227	0.7950	0.37275
afc	1	10.34	10.342	36.2946	2.168e-09 ***
afc0	1	49.82	49.823	174.8573	< 2.2e-16 ***
bs(logamh)	3	122.64	40.882	143.4761	< 2.2e-16 ***
fsh3cat	3	7.65	2.550	8.9501	7.126e-06 ***
lupprot.f	2	19.32	9.660	33.9024	4.205e-15 ***
bs(logamh):fsh3cat	9	4.91	0.546	1.9155	0.04601 *
Residuals	1392	396.63	0.285		

```
## ---
```

```
## 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(data,!is.na(amh)))
summary(m5)
```

```
##
```

```
## Call:
```

```
## lm(formula = loglastE2 ~ age + bmi + afc + afc0 + fsh3 * amh.f +
##     lupprot.f, data = subset(data, !is.na(amh)))
```

```
##
```

```
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-2.89403	-0.29585	0.04869	0.33712	2.61077

```
##
```

```
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.0370910  0.2118514  33.217 < 2e-16 ***
## age              0.0019488  0.0037733   0.516  0.60560
## bmi             -0.0048453  0.0028925  -1.675  0.09413 .
## afc              0.0101473  0.0021767   4.662 3.44e-06 ***
## afc0            0.2510238  0.0471010   5.329 1.15e-07 ***
## fsh3            -0.0005134  0.0003640  -1.411  0.15860
## amh.f(0.92,2.15]  0.5358877  0.1915609   2.797  0.00522 **
## amh.f(2.15,4]     0.8648299  0.1755972   4.925 9.44e-07 ***
## amh.f(4,41.4]     0.8931342  0.1775813   5.029 5.56e-07 ***
## lupprot.fLPL 10/5  0.2050757  0.0340388   6.025 2.16e-09 ***
## lupprot.fLupron Microdose 0.2449580  0.0475173   5.155 2.90e-07 ***
## fsh3:amh.f(0.92,2.15] -0.0004847  0.0004915  -0.986  0.32424
## fsh3:amh.f(2.15,4]   -0.0008742  0.0004814  -1.816  0.06961 .
## fsh3:amh.f(4,41.4]   -0.0004098  0.0005548  -0.739  0.46018
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5512 on 1399 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.3415, Adjusted R-squared:  0.3354
## F-statistic: 55.82 on 13 and 1399 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  35.27  35.268 116.0678 < 2.2e-16 ***
## bmi         1   0.22   0.220   0.7240   0.3950
## afc         1  10.21  10.213  33.6107 8.309e-09 ***
## afc0        1  49.23  49.234 162.0300 < 2.2e-16 ***
## fsh3        1  60.71  60.709 199.7936 < 2.2e-16 ***
## amh.f       3  47.56  15.854  52.1747 < 2.2e-16 ***
## lupprot.f   2  16.29   8.143  26.8000 3.787e-12 ***
## fsh3:amh.f  3   1.01   0.337   1.1084   0.3445
## Residuals 1399 425.10   0.304
```

```
## ---
```

```
## 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(data, !is.na(amh)))
summary(m6)
```

```
##
```

```
## Call:
```

```
## lm(formula = loglastE2 ~ age + bmi + afc + afc0 + bs(fsh3) *
##     amh.f + lupprot.f, data = subset(data, !is.na(amh)))
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.91021 -0.29733  0.03667  0.33304  2.52274
```

```
##
```

```
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)          5.118777    0.424747   12.051 < 2e-16 ***
## age                  0.003909    0.003778    1.035 0.300985
## bmi                 -0.005463    0.002873   -1.901 0.057446 .
## afc                  0.009681    0.002167    4.467 8.59e-06 ***
## afc0                0.245000    0.046835    5.231 1.94e-07 ***
## bs(fsh3)1           2.949543    0.769130    3.835 0.000131 ***
## bs(fsh3)2           1.511968    0.412751    3.663 0.000258 ***
## bs(fsh3)3           1.307875    0.487380    2.683 0.007372 **
## amh.f(0.92,2.15]    1.539051    0.980245    1.570 0.116627
## amh.f(2.15,4]       2.576004    0.700701    3.676 0.000246 ***
## amh.f(4,41.4]       2.121921    0.552007    3.844 0.000127 ***
## lupprot.fLPL 10/5    0.196806    0.033802    5.822 7.20e-09 ***
## lupprot.fLupron Microdose 0.260038    0.047563    5.467 5.41e-08 ***
## bs(fsh3)1:amh.f(0.92,2.15] -1.252039    1.817502   -0.689 0.491014
## bs(fsh3)2:amh.f(0.92,2.15] -1.899026    0.710808   -2.672 0.007636 **
## bs(fsh3)3:amh.f(0.92,2.15] -0.584407    1.202737   -0.486 0.627116
## bs(fsh3)1:amh.f(2.15,4]   -2.800570    1.530203   -1.830 0.067435 .
## bs(fsh3)2:amh.f(2.15,4]   -2.201166    0.603356   -3.648 0.000274 ***
## bs(fsh3)3:amh.f(2.15,4]   -1.776764    1.212976   -1.465 0.143202
## bs(fsh3)1:amh.f(4,41.4]   -1.636239    1.306611   -1.252 0.210679
## bs(fsh3)2:amh.f(4,41.4]   -1.770377    0.864404   -2.048 0.040739 *
## bs(fsh3)3:amh.f(4,41.4]   -1.629766    1.558258   -1.046 0.295794
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5452 on 1391 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared:  0.3595, Adjusted R-squared:  0.3499
## F-statistic: 37.18 on 21 and 1391 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  35.27   35.268 118.6431 < 2.2e-16 ***
## bmi          1   0.22    0.220   0.7401   0.3898
## afc          1  10.21   10.213  34.3564 5.722e-09 ***
## afc0         1  49.23   49.234 165.6250 < 2.2e-16 ***
## bs(fsh3)     3  71.55   23.850  80.2314 < 2.2e-16 ***
## amh.f        3  40.86   13.621  45.8214 < 2.2e-16 ***
## lupprot.f    2  17.07    8.537  28.7175 6.008e-13 ***
## bs(fsh3):amh.f  9   7.69    0.854   2.8732   0.0023 **
## Residuals 1391 413.49    0.297
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
with(data, plot(loglastE2~ logamh, pch=19, col="#00000020"))
```

```
for(i in 1:4){
```

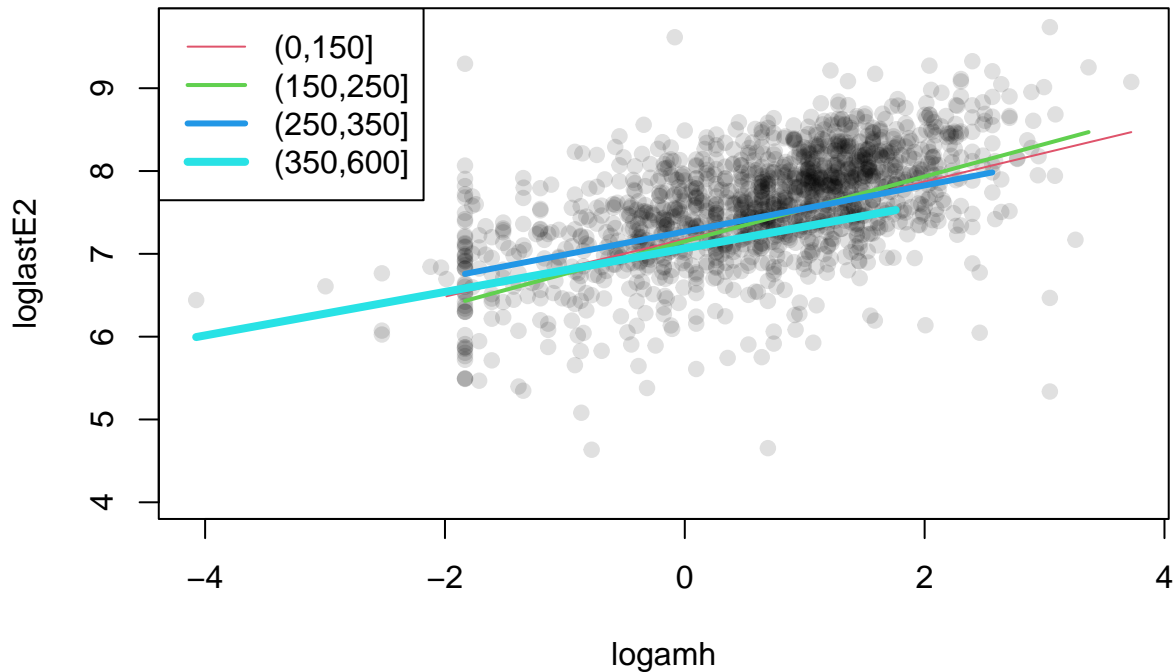
```
  mynewdata <- data.frame(age=mean(data$age), bmi=mean(data$bmi), afc=mean(data$afc),
```

```

afc0=mean(data$afc0), fsh3cat=levels(data$fsh3cat)[i], lupprot.f="Antagonist",
  logamh=seq(myranges[1,i], myranges[2,i], l=31) )
myfit <- predict(m3, newdata= mynewdata)
lines(x=mynewdata$logamh, y=myfit, lwd=i, col=i+1)
}
legend("topleft", col=2:5, lwd=1:4, levels(data$fsh3cat))
title(main="Straight line fits by FSH3 category", sub="Note: numeric covariates at mean level, lupprot="

```

Straight line fits by FSH3 category



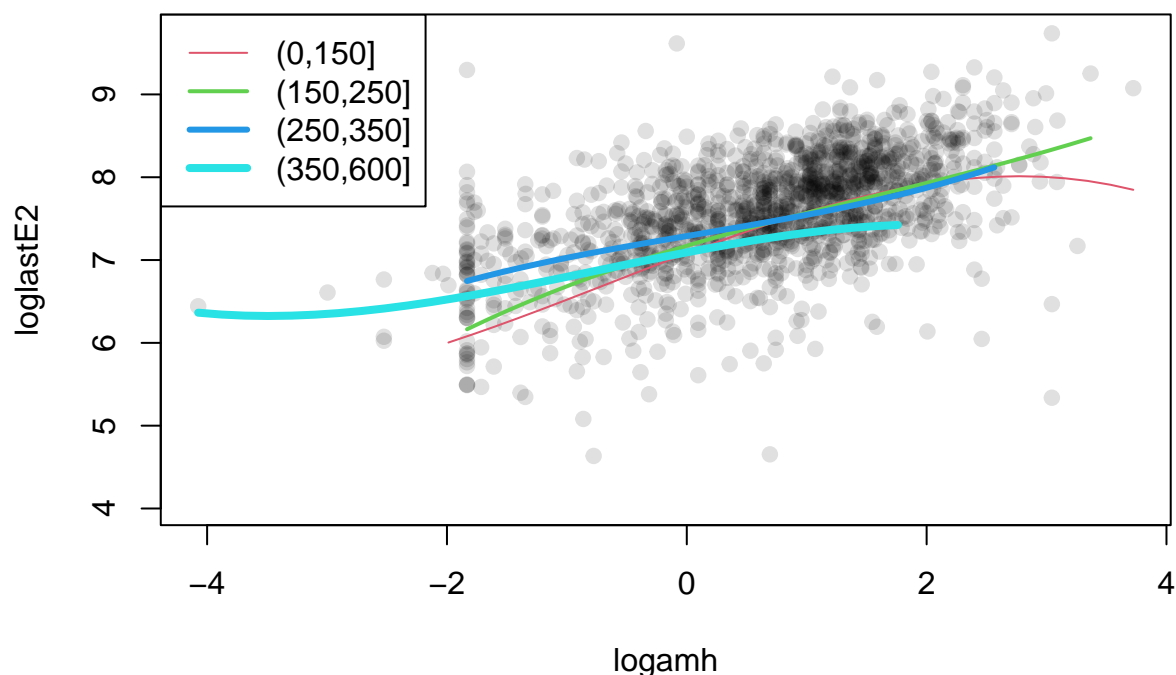
Note: numeric covariates at mean level, lupprot=antagonist

```

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

```

Spline fits by FSH3 category



Note: numeric covariates at mean level, lupprot=antagonist

Statistical learning approaches

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

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
```

```
##
```

```
## expand, pack, unpack
```

```
## Loaded glmnet 4.1-4
```

```
# comparison of main effects-only model with CV lasso
```

```
par(mfrow=c(1,2))
```

```
coef(summary(m1))
```

```
##
```

	Estimate	Std. Error	t value
## (Intercept)	7.3651363633	0.1559526653	47.226742
## age	0.0042864597	0.0036573807	1.172003
## bmi	-0.0065152109	0.0028082777	-2.320002
## afc	0.0075997866	0.0021323218	3.564090
## afc0	0.1983474269	0.0461550992	4.297411
## log(amh)	0.2972937902	0.0184904865	16.078203
## factor(lupprot)LPL 10/5	0.2285524498	0.0323891389	7.056453
## factor(lupprot)Lupron Microdose	0.2694409936	0.0460402640	5.852290
## fsh3	-0.0007484203	0.0001882011	-3.976706

```
##                                Pr(>|t|)
## (Intercept)                   2.463381e-292
## age                           2.413947e-01
## bmi                           2.048343e-02
## afc                           3.773456e-04
## afc0                          1.846991e-05
## log(amh)                      1.596457e-53
## factor(lupprot)LPL 10/5       2.674165e-12
## factor(lupprot)Lupron Microdose 6.025233e-09
## fsh3                          7.343337e-05

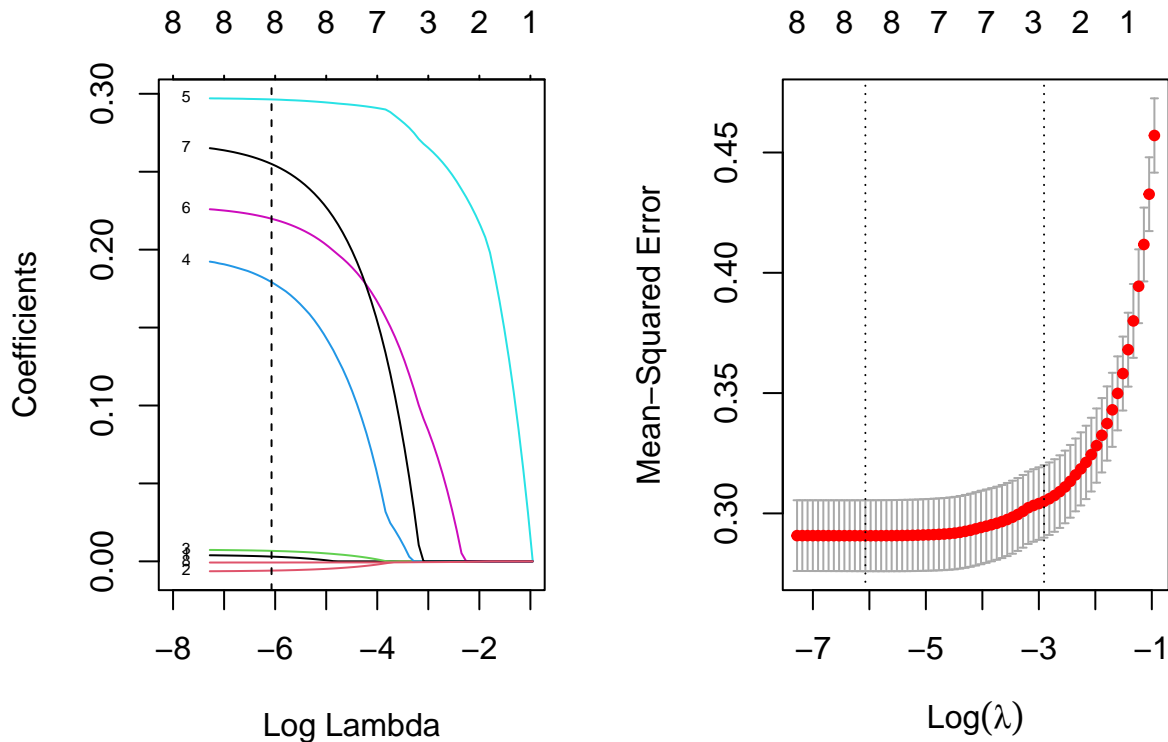
plot(glmnet(x=model.matrix(m1)[,-1], y=m1$model[,1]),
      xvar="lambda", label=TRUE, xlim=c(-8,-1))
set.seed(4)
cv.glmnet1 <- cv.glmnet(x=model.matrix(m1)[,-1], y=m1$model[,1])
print(cv.glmnet1 )

##
## Call:  cv.glmnet(x = model.matrix(m1)[, -1], y = m1$model[, 1])
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min 0.00231    56 0.2907 0.01475        8
## 1se 0.05459    22 0.3050 0.01507        3

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

## [1] 0.2881211

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



```

cv.glmnet1$lambda.min

## [1] 0.002308725
log(cv.glmnet1$lambda.min)

## [1] -6.07106
cbind( coef(m1), coef(cv.glmnet1, s = "lambda.min"))

## 9 x 2 sparse Matrix of class "dgCMatrix"
##
## (Intercept)          7.3651363633  7.4096973184
## age              0.0042864597  0.0030546837
## bmi             -0.0065152109 -0.0059294936
## afc              0.0075997866  0.0067282097
## afc0             0.1983474269  0.1791451556
## log(amh)         0.2972937902  0.2963774450
## factor(lupprot)LPL 10/5  0.2285524498  0.2198354296
## factor(lupprot)Lupron Microdose 0.2694409936  0.2549748141
## fsh3             -0.0007484203 -0.0007308659

```

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)    7.154042882  0.170369051  41.9914464  4.452362e-250
## age            0.003955658  0.003647568   1.0844642  2.783457e-01
## bmi           -0.007468188  0.002817130  -2.6509912  8.116114e-03
## afc            0.007647908  0.002126158   3.5970550  3.330390e-04
## afc0           0.195832542  0.045994078   4.2577773  2.202247e-05
## logamh         0.347124332  0.048818683   7.1104813  1.837952e-12
## fsh3cat(150,250] -0.027255807  0.108757709  -0.2506103  8.021522e-01
## fsh3cat(250,350]  0.091713905  0.097375400   0.9418591  3.464273e-01
## fsh3cat(350,600] -0.111060203  0.094346943  -1.1771468  2.393370e-01
## lupprot.fLPL 10/5  0.240818652  0.032966690   7.3049084  4.641112e-13
## lupprot.fLupron Microdose 0.273252469  0.045876022   5.9563243  3.257934e-09
## logamh:fsh3cat(150,250]  0.044834024  0.064445214   0.6956921  4.867372e-01
## logamh:fsh3cat(250,350] -0.068549482  0.060496937  -1.1331066  2.573635e-01
## logamh:fsh3cat(350,600] -0.084239459  0.054290440  -1.5516444  1.209733e-01

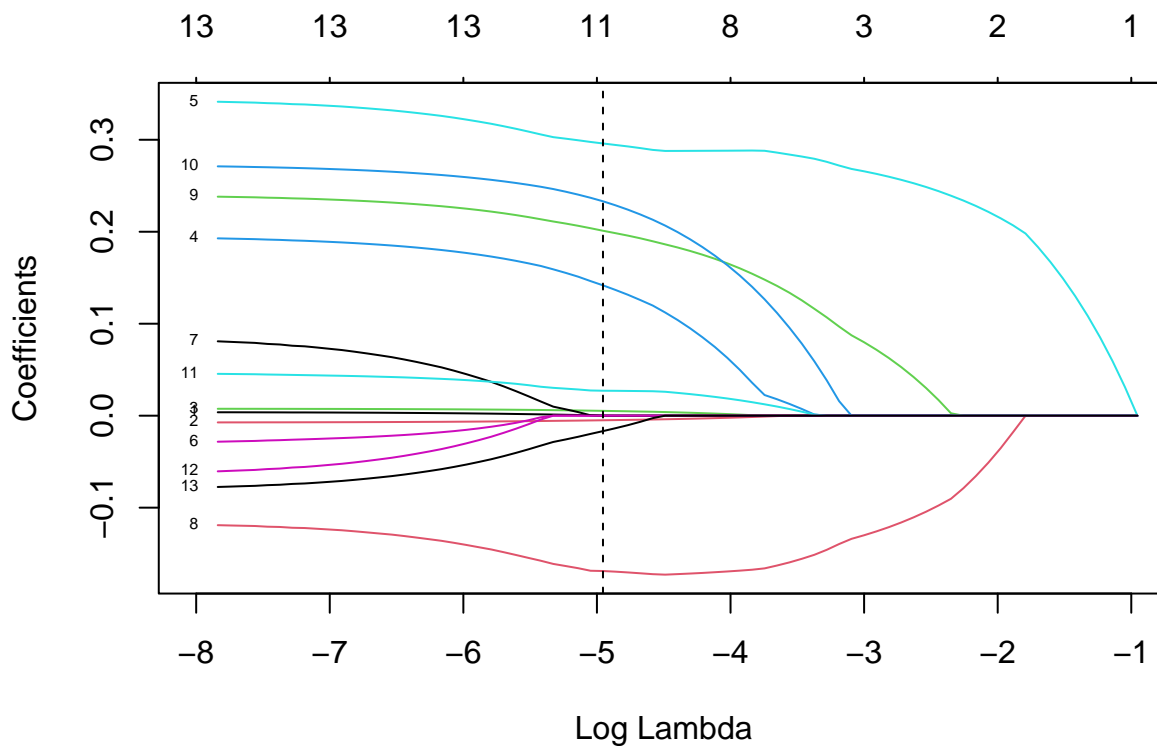
```

```

#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)
```



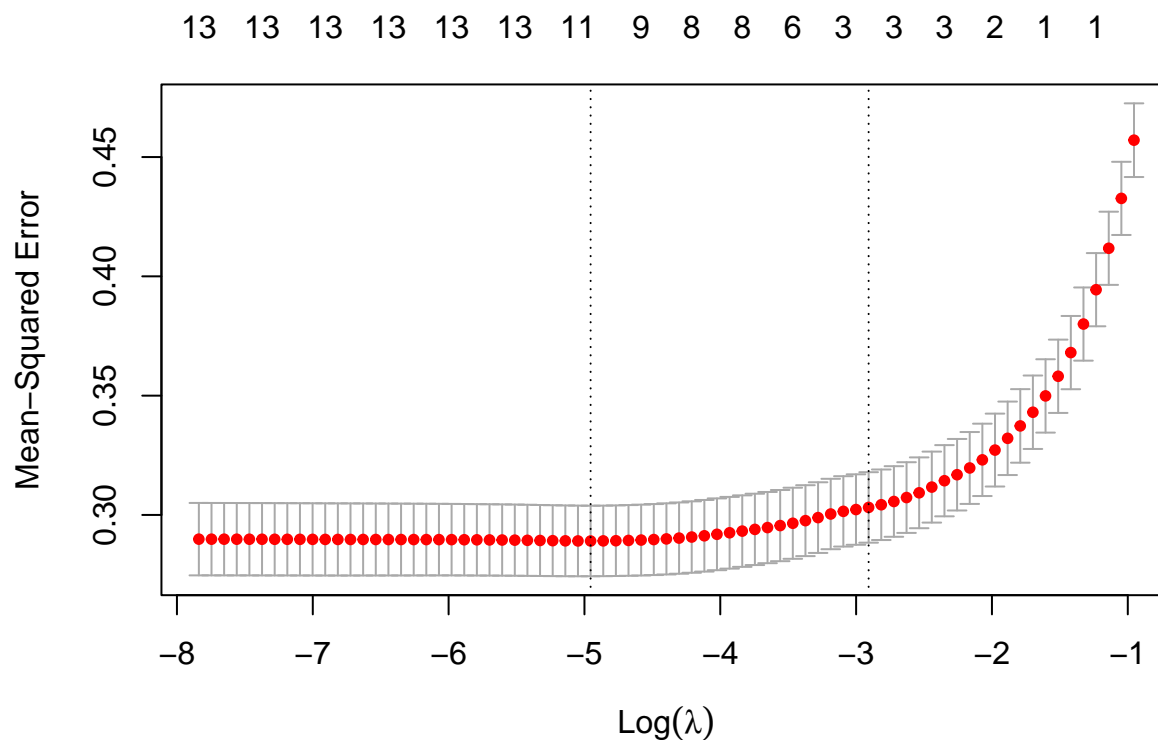
```
print(cv.glmnet3 )
```

```
##
## Call:  cv.glmnet(x = model.matrix(m3)[-1], y = m3$model[, 1])
##
## Measure: Mean-Squared Error
##
##      Lambda Index Measure      SE Nonzero
## min 0.00705   44  0.2891 0.01478      10
## 1se 0.05459   22  0.3031 0.01477       3
```

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

```
## [1] 0.2854232
```

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

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

```
## [1] 0.007050512
```

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

```
## [1] -4.954655
```

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

```
## 14 x 2 sparse Matrix of class "dgCMatrix"
```

```
##                                     s1
## (Intercept)          7.154042882  7.360019e+00
## age                0.003955658  9.132309e-05
## bmi               -0.007468188 -4.984006e-03
## afc                0.007647908  5.226483e-03
## afc0              0.195832542  1.417112e-01
## logamh            0.347124332  2.959619e-01
## fsh3cat(150,250]   -0.027255807 .
## fsh3cat(250,350]    0.091713905 .
## fsh3cat(350,600]   -0.111060203 -1.688898e-01
## lupprot.fLPL 10/5  0.240818652  2.010569e-01
## lupprot.fLupron Microdose 0.273252469  2.331045e-01
## logamh:fsh3cat(150,250] 0.044834024  2.713846e-02
## logamh:fsh3cat(250,350] -0.068549482 .
## logamh:fsh3cat(350,600] -0.084239459 -1.680133e-02
```

Poisson regression

```
# mii
m7 <- glm(mii ~ age + bmi + afc + afc0 + log(amh) + factor(lupprot) + fsh3 + log(lastE2),
          data=clean.data,
```

```

    family = poisson(link="log"))
summary(m7)

##
## Call:
## glm(formula = mii ~ age + bmi + afc + afc0 + log(amh) + factor(lupprot) +
##      fsh3 + log(lastE2), family = poisson(link = "log"), data = clean.data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7.0886  -1.6006  -0.1487   1.0630   6.3090
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.3653566   0.1612047  -8.470  < 2e-16 ***
## age             -0.0067051   0.0022189  -3.022  0.00251 **
## bmi             -0.0028213   0.0016966  -1.663  0.09633 .
## afc              0.0074344   0.0010549   7.048 1.82e-12 ***
## afc0            0.1285807   0.0282731   4.548 5.42e-06 ***
## log(amh)        0.1876585   0.0139569  13.446 < 2e-16 ***
## factor(lupprot)LPL 10/5 -0.1489428  0.0196323  -7.587 3.28e-14 ***
## factor(lupprot)Lupron Microdose -0.2415286  0.0364917  -6.619 3.62e-11 ***
## fsh3            -0.0003607   0.0001267  -2.847  0.00441 **
## log(lastE2)      0.4905072   0.0176285  27.825 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 9327.9  on 1413  degrees of freedom
## Residual deviance: 5556.1  on 1404  degrees of freedom
## AIC: 10551
##
## Number of Fisher Scoring iterations: 5
# offset = log(lastE2)?

## Total_bc
m8 <- glm(Total_bc ~ age + bmi + afc + afc0 + log(amh) + factor(lupprot) + fsh3 + log(lastE2),
    data=clean.data,
    family = poisson(link="log"))
summary(m8)

##
## Call:
## glm(formula = Total_bc ~ age + bmi + afc + afc0 + log(amh) +
##      factor(lupprot) + fsh3 + log(lastE2), family = poisson(link = "log"),
##      data = clean.data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4.8638  -1.5955  -0.4993   0.8944   6.3692
##
## Coefficients:

```

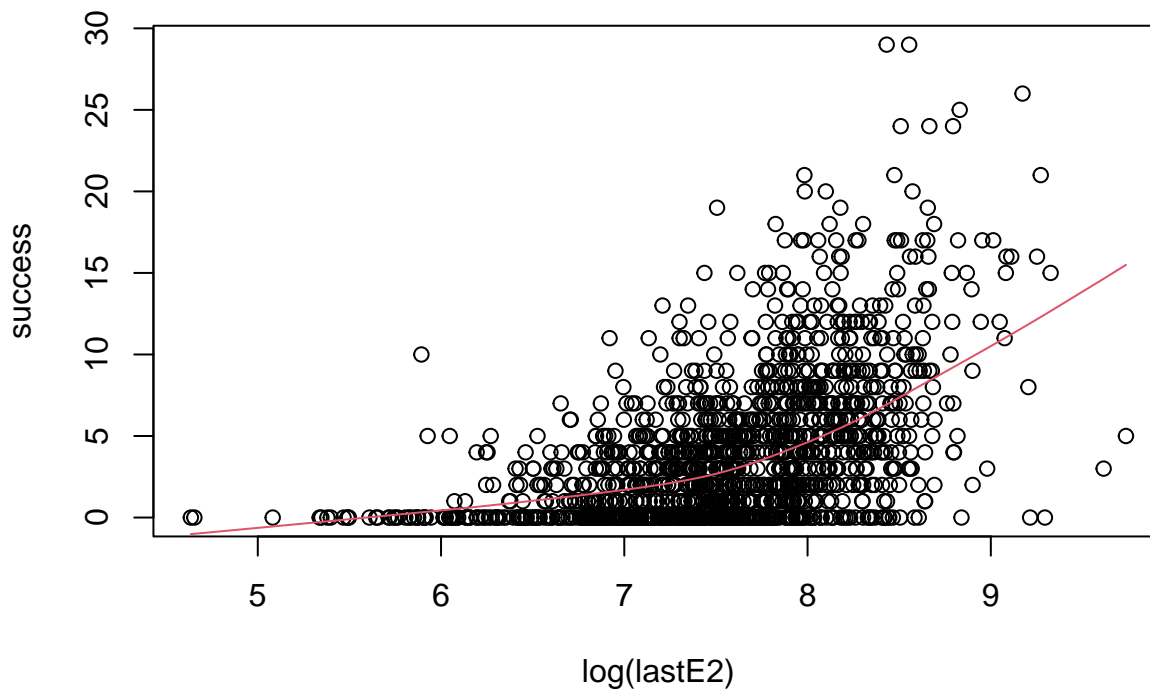
```
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.7969928  0.2465690 -11.344 < 2e-16 ***
## age           -0.0225146  0.0033818  -6.658 2.78e-11 ***
## bmi           -0.0035035  0.0025701  -1.363  0.17283
## afc            0.0090450  0.0016199   5.584 2.36e-08 ***
## afc0           0.4319349  0.0437677   9.869 < 2e-16 ***
## log(amh)       0.1633619  0.0216608   7.542 4.63e-14 ***
## factor(lupprot)LPL 10/5 -0.0341690  0.0293394  -1.165  0.24418
## factor(lupprot)Lupron Microdose -0.3611675  0.0617275  -5.851 4.89e-09 ***
## fsh3           -0.0005260  0.0001963  -2.680  0.00736 **
## log(lastE2)     0.6218147  0.0269875  23.041 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##    Null deviance: 6869.7  on 1413  degrees of freedom
## Residual deviance: 4299.0  on 1404  degrees of freedom
## AIC: 7703.2
##
## Number of Fisher Scoring iterations: 5
```

Logistic Regression

Create counts of success (conversion or not) and counts of failure

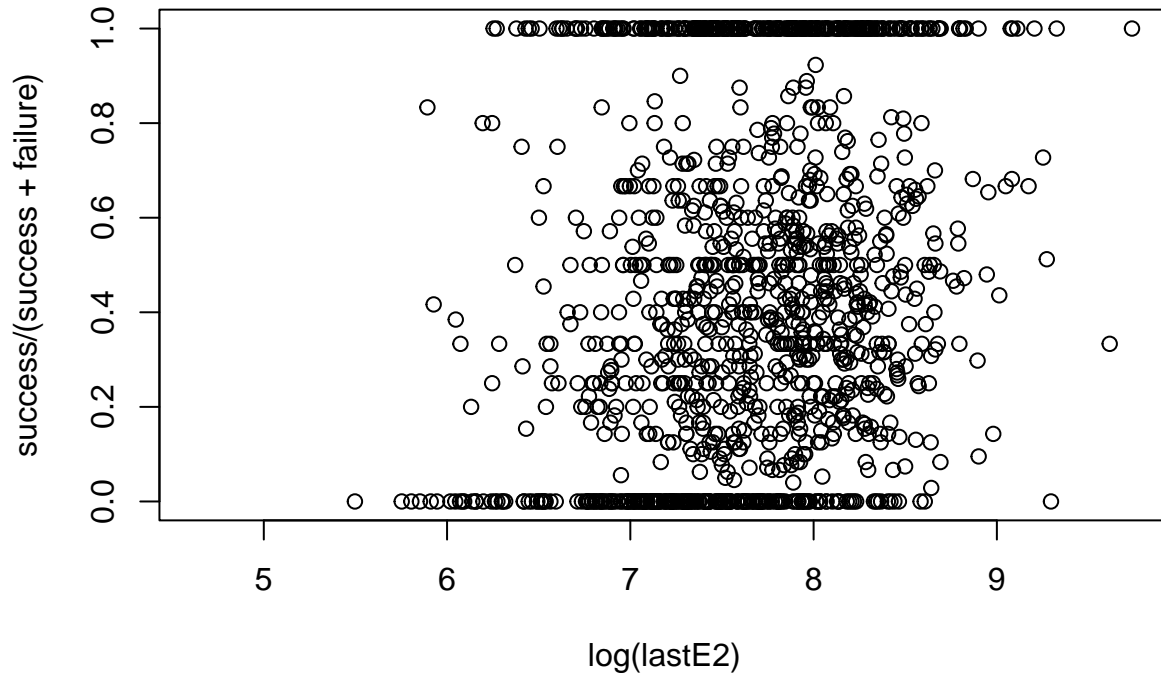
```
clean.data$success <- clean.data$Total_bc
clean.data$failure <- clean.data$mii - clean.data$Total_bc
```

```
plot(success ~ log(lastE2), clean.data)
with(clean.data, lines(lowess(success ~ log(lastE2)),col=2))
```



```
# scatterplot standardized for trial size
```

```
plot(success/(success + failure) ~ log(lastE2), clean.data, ylim = c(0,1))
```



```
m9 <- glm(cbind(success, failure) ~ age + bmi + afc + afc0 + log(amh) + factor(lupprot) + fsh3 + log(la
      data = clean.data,
      family = "binomial")
summary(m9)
```

```
##
## Call:
## glm(formula = cbind(success, failure) ~ age + bmi + afc + afc0 +
##      log(amh) + factor(lupprot) + fsh3 + log(lastE2), family = "binomial",
##      data = clean.data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -6.821  -1.361   0.000   1.481   6.714
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.5703110   0.3410727  -4.604 4.14e-06 ***
## age             -0.0299644   0.0044909  -6.672 2.52e-11 ***
## bmi             -0.0011343   0.0034576  -0.328  0.7429
## afc              0.0027892   0.0023397   1.192  0.2332
## afc0             0.5355732   0.0613794   8.726 < 2e-16 ***
## log(amh)        -0.0360801   0.0294853  -1.224  0.2211
## factor(lupprot)LPL 10/5  0.1834804   0.0406563   4.513 6.39e-06 ***
## factor(lupprot)Lupron Microdose -0.1631726   0.0787306  -2.073  0.0382 *
## fsh3            -0.0002808   0.0002692  -1.043  0.2968
## log(lastE2)       0.2750344   0.0383957   7.163 7.88e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 6168.6  on 1263  degrees of freedom
## Residual deviance: 5730.7  on 1254  degrees of freedom
## AIC: 7772.9
##
## Number of Fisher Scoring iterations: 4
```