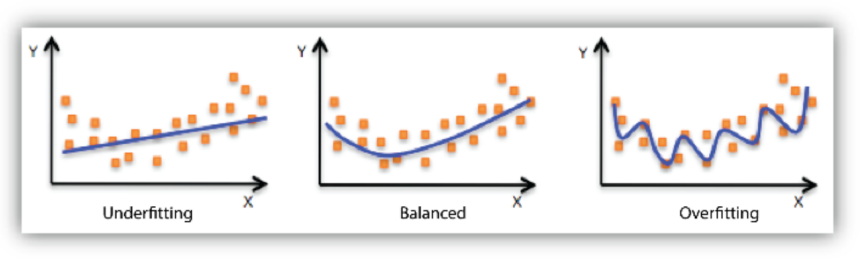
BIOSTATS 640 – Introduction to R
Fall 2023
https://people.umass.edu/biep640w/webpages/demonstrations.html



https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html

07
Introduction to Multiple Linear Regression in R
October 20, 2023

Dataset used
framingham_didactic.xlsx

		Page
1	Introduction to the Framingham Heart Study Didactic Data: **framingham_didactic.xlsx**	2
2	Highlights of Lesson 06 – Introduction to Simple Linear Regression in R	3
3	Create New Variables Using `mutate( ) in {dplyr}`	4
4	Introduction to Quantiles in Regression	6
5	Explore Your Data	7
6	Fit Models and Display Side-by-Side Using `stargazer( ) in {stargazer}`	9
7	Partial F-Test to Compare Hierarchical Models	11
8	Plot Marginal Predictions	13
9	Report	14

Packages used:

```
tidyverse, ggplot2, GGally, stargazer, gridExtra
```

1. Introduction to the Framingham Heart Study Didactic Data

framingham_didactic.xlsx

Source:

The Framingham Didactic dataset was contributed by Dr. Amy Nowacki, Associate Professor, Cleveland Clinic. Please refer to this resource as: Amy S. Nowacki, "Framingham Didactic Dataset", *TSHS Resources Portal (2015).* Available at https://www.causeweb.org/tshs/framingham-didactic/.

Description:

Cardiovascular disease (CVD) is the leading cause of death and serious illness in the United States. In 1948, the Framingham Heart Study, under the direction of the National Heart Institute (now known as the National Heart, Lung, and Blood Institute or NHLBI), was initiated.

The objective of the Framingham Heart Study was to identify the common factors or characteristics that contribute to CVD by following its development over a long period of time in a large group of participants who had not yet developed overt symptoms of CVD or suffered a heart attack or stroke.

Goal:

This lesson is an illustration of multiple linear regression. It walks you through the steps of developing and reporting a multiple predictor linear regression model for a dependent variable that is a selected transformation of systolic blood pressure.

Dependent Variable: **`Y = -1/sqrt(sbp)`**.

Predictor variables: **`sex (sex), body mass index (bmi), serum cholesterol (scl),`** and **`age (age).`**

Data dictionary/Codebook

Position	Variable	Label	Type	Codings
1	id	Patient identifier		
2	sex	Sex at birth	numeric	1=male 2=female
3	sbp	Systolic Blood Pressure (mm Hg)	numeric	
4	dbp	Diastolic Blood Pressure (mm Hg)	numeric	
5	scl	Serum cholesterol (mg/100 ml)	numeric	
6	age	Age at baseline exam, years	numeric	
7	bmi	Body Mass index (kg/m^2)	numeric	
8	month	Month of year of baseline exam	numeric	
9	followup	Follow-up: days since baselinel	numeric	
10	chdfate	Event of CHD at follow up	numeric	1 = event had developed CHD 0 = otherwise

2. Highlights of Lesson 06
Introduction to Simple Linear Regression in R

Simple Linear Regression

Fit model. Save as object.

```
fitobject <- lm(yvar ~ xvar, data=dataframename)
Example:
model_simple <- lm(draft_number ~ day_birth, data=draftlottery1970)
```

Return names of model fit object.

```
names(fitobject)
Example:
names(model_simple)
```

Show model output.

```
summary(fitobject)
Example:
summary(model_simple)
```

Show show regression estimates and confidence intervals.

```
cbind(coef(fitobject),confint(fitobject))
Example:
cbind(coeff(model_simple), confint(model_simple))
```

Show analysis of variance table.

```
anova(fitobject)
Example:
anova(model_simple)
```

Nice tabular report using package {stargazer}

```
library(stargazer)
stargazer(fitobject, type="text")
Example:
stargazer(model_simple, type="text")
```

3. Create New Variables Using mutate( ) in {dplyr}

initialize session

```
setwd("/cloud/project")                                          # Set working directory
getwd()                                                          # Check working directory
options(scipen=999)                                              # Turn off scientific notation
rm(list = ls())                                                  # Clear the Decks
```

import excel data

```
library(readxl)
source <- read_excel("framingham_didactic.xlsx")
source <- as.data.frame(source)
str(source)

## 'data.frame':    4699 obs. of  10 variables:
##  $ id      : num  2642 4627 2568 4192 3977 ...
##  $ sex     : num  1 1 1 1 1 2 1 1 1 1 ...
##  $ sbp     : num  120 130 144 92 162 212 140 174 142 115 ...
##  $ dbp     : num  80 78 90 66 98 118 85 102 94 70 ...
##  $ scl     : num  267 192 207 231 271 182 276 259 242 242 ...
##  $ age     : num  55 53 61 48 39 61 44 39 47 60 ...
##  $ bmi     : num  25 28.4 25.1 26.2 28.4 33.3 25.3 27.9 26.6 30.8 ...
##  $ month   : num  8 12 8 11 11 2 6 11 5 10 ...
##  $ followup: num  18 35 109 147 169 199 201 209 265 278 ...
##  $ chdfate : num  1 1 1 1 1 1 1 1 1 1 ...                                  We see that all 10 variables are numeric
```

create new variables

```
library(tidyverse)                                               # mutate( ) requires {dplyr} bundled in {tidyverse}

#--- basic ---#
ready <- source %>%
         mutate(y_sbp = -1/sqrt(sbp))                            # KEY:  mutate(NEWVAR = stuff)

#--- 0/1 indicator as numeric ---#
ready <- ready %>%
         mutate(female01 = ifelse(sex==2,1,0))                   # KEY:  ifelse(CONDITION THAT MUST BE TRUE, 1, 0)

# ---  0/1 indicator as factor ---#
ready <- ready %>%
         mutate(female01f= recode_factor(female01,
                           "0" = "male",
                           "1" = "female"))

# ---  Age, grouped ---#
ready <- ready %>%
   mutate(age_group = case_when(                                 # Step 1.  Create character version
                           age %in% 30:39 ~ "30-39",             # KEY:  age %in% OLD ~ "NEW",
                           age %in% 40:49 ~ "40-49",
                           age %in% 50:59 ~ "50-59",
                           age %in% 60:69 ~ "60-69",
                           age %in% 70:79 ~ "70-79")) %>%

    mutate(age_group = recode_factor(age_group,                  # Step 2.  Convert to factor using recode_factor( )
                           "30-39" = "30-39",
                           "40-49" = "40-49",
                           "50-59" = "50-59",
                           "60-69" = "60-69",
                           "70-79" = "70-79"))
```

```
# ---  GOOD PRACTICE: retain complete data only ---#
ready <- ready %>%
         na.omit()


# ---  GOOD TO KNOW: We could have done all of the above in one series of %>% connected lines of code ---#
#ready <- source %>%
#         mutate(y_sbp = -1/sqrt(sbp)) %>%
#         mutate(female01 = ifelse(sex==2,1,0)) %>%
#         na.omit()

# ---  Check ---#
table(ready$sex,ready$female01, dnn=c("sex","female01"))        # KEY:  dnn = c("ROW varname", "COLUMN varname")

##    female01
## sex    0    1
##   1 2040    0
##   2    0 2618      correct.

table(ready$sex,ready$female01f,dnn=c("sex","female01f"))

##    female01f
## sex male female
##   1 2040      0
##   2    0   2618    also correct

str(ready)

## 'data.frame':    4658 obs. of  14 variables:
##  $ id       : num  2642 4627 2568 4192 3977 ...
##  $ sex      : num  1 1 1 1 1 2 1 1 1 1 ...
##  $ sbp      : num  120 130 144 92 162 212 140 174 142 115 ...
##  $ dbp      : num  80 78 90 66 98 118 85 102 94 70 ...
##  $ scl      : num  267 192 207 231 271 182 276 259 242 242 ...
##  $ age      : num  55 53 61 48 39 61 44 39 47 60 ...
##  $ bmi      : num  25 28.4 25.1 26.2 28.4 33.3 25.3 27.9 26.6 30.8 ...
##  $ month    : num  8 12 8 11 11 2 6 11 5 10 ...
##  $ followup : num  18 35 109 147 169 199 201 209 265 278 ...
##  $ chdfate  : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ y_sbp    : num  -0.0913 -0.0877 -0.0833 -0.1043 -0.0786 ...
##  $ female01 : num  0 0 0 0 0 1 0 0 0 0 ...                                    new variable female01 is numeric
##  $ female01f: Factor w/ 2 levels "male","female": 1 1 1 1 1 2 1 1 1 1 ...    new variable female01f is factor
##  $ age_group: Factor w/ 4 levels "30-39","40-49",..: 3 3 4 2 1 4 2 1 2 4 ... new variable age_group is factor
##  - attr(*, "na.action")= 'omit' Named int [1:41] 68 132 364 392 426 433 520 590 645 726 ...
##   ..- attr(*, "names")= chr [1:41] "68" "132" "364" "392" ...
```

4. Introduction to Quantiles in Regression

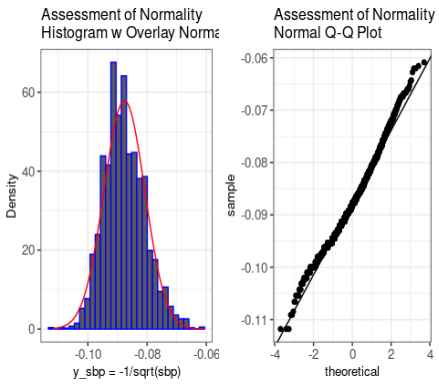
introduction to quantiles

```
library(tidyverse)

quantile(ready$age, probs = c(0, .25, .50, .75, 1.0), na.rm = TRUE)          # get quartiles
##   0%  25%  50%  75% 100%
##   30   39   45   53   66

ready <- ready %>%
         mutate(age_quartile = ntile(age,4), na.rm=T)                        # age_quartile, numeric has values 1, 2,3,4

ready <- ready %>%
         mutate(age_Q2 = ifelse(age_quartile==2,1,0)) %>%                    # create (4-1) 0/1 DESIGN variables
         mutate(age_Q3 = ifelse(age_quartile==3,1,0)) %>%
         mutate(age_Q4 = ifelse(age_quartile==4,1,0))

ready <- ready %>%
         mutate(age_quartile = recode_factor(age_quartile,                     # convert age_quartile to factor
                           "1" = "Q1: 30 - 39",                                # KEY to label:  "source" = "new"
                           "2" = "Q2: 39 - 45",
                           "3" = "Q3: 45 - 53",
                           "4" = "Q4: 53 - 66"))

table(ready$age_quartile)                                                    # check.
##
## Q1: 30 - 39 Q2: 39 - 45 Q3: 45 - 53 Q4: 53 - 66
##        1165        1165        1164        1164                    Pretty close to 25% of sample in each quartile
```

5. Explore Your Data

explore your data

```
library(summarytools)                                                    # descr() in package {summarytools}
library(ggplot2)
library(gridExtra)                                                       # grid.arrange() in package {gridExtra}

descr(ready$y_sbp,
      stats = c("n.valid","mean", "sd", "min","q1", "med", "q3", "max","CV"),
      transpose = TRUE)
## Descriptive Statistics
## ready$y_sbp
## N: 4658
##
##               N.Valid    Mean   Std.Dev      Min      Q1   Median      Q3     Max      CV       Recall: CV = SD/Mean
## ----------- --------- ------- --------- ------- ------- -------- ------- ------- -------
##       y_sbp   4658.00   -0.09      0.01   -0.11   -0.09    -0.09   -0.08   -0.06   -0.08       Nothing odd jumping out

# panel 1 = histogram w overlay normal
p1 <-    ggplot(data=ready) +                                                  # required layer:  data=
         aes(x=y_sbp) +                                                        # required layer:  aes()
         geom_histogram(colour="blue",                                         # required layer:  geom_SOMETHING
                        aes(y=..density..)) +
         stat_function(fun=dnorm,                                              # stat_function()for overlay normal
                       color="red",
                       args=list(mean=mean(ready$y_sbp),                       # overlay normal mean  = sample mean
                                 sd=sd(ready$y_sbp))) +                        # overlay normal sd = sample SD
         ggtitle("Assessment of Normality\nHistogram w Overlay Normal") +      # OPTIONAL aesthetics
         xlab("y_sbp = -1/sqrt(sbp)") +
         ylab("Density") +
         theme_bw() +
         theme(axis.text = element_text(size = 10),
               axis.title = element_text(size = 10),
               plot.title = element_text(size = 12))

# panel 2 = quantile-quantile plot
p2 <- ggplot(data=ready) +
          aes(sample=y_sbp) +
          stat_qq() +
          geom_abline(intercept=mean(ready$y_sbp),
                      slope = sd(ready$y_sbp)) +
          ggtitle("Assessment of Normality\nNormal Q-Q Plot") +
          theme_bw() +
          theme(axis.text = element_text(size = 10),
                axis.title = element_text(size = 10),
                plot.title = element_text(size = 12))

gridExtra::grid.arrange(p1, p2, ncol=2)
```



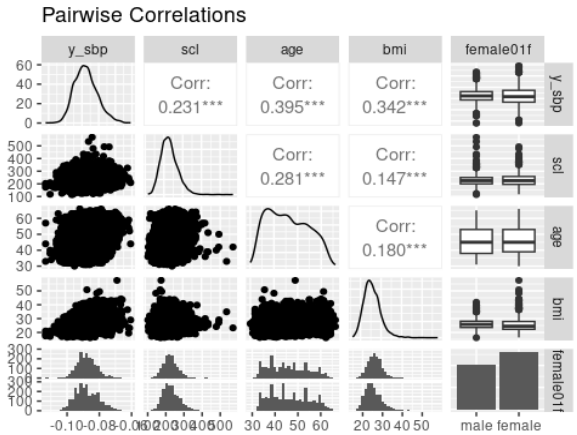
Inspection suggests assumption of normality of y_sbp is reasonable.

explore your data

```
library(summarytools)                                          # descr( ) and freq( ) in {summarytools}

# single variable predictors - continuous
myvars <- c("scl","bmi","age")                          # TIP. Object myvars makes it easier to code several descriptives
.
descr(ready[myvars],
      stats = c("n.valid","mean", "sd", "min","q1", "med", "q3", "max","CV"),
      transpose = TRUE)
## Descriptive Statistics
## ready
## N: 4658
##
##             N.Valid     Mean   Std.Dev      Min       Q1   Median       Q3      Max     CV
## --------- --------- -------- --------- -------- -------- -------- -------- -------- ------
##       age   4658.00    46.03      8.49    30.00    39.00    45.00    53.00    66.00   0.18
##       bmi   4658.00    25.63      4.08    16.20    22.90    25.20    28.00    57.60   0.16
##       scl   4658.00   228.29     44.55   115.00   197.00   225.00   255.00   568.00   0.20      Looks okay

# single variable predictors - discrete
freq(ready$female01f)
## Frequencies
## ready$female01f
## Type: Factor
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ------------ ------ --------- -------------- --------- --------------
##         male   2040     43.80          43.80     43.80          43.80
##       female   2618     56.20         100.00     56.20         100.00
##         <NA>      0                               0.00         100.00
##        Total   4658    100.00         100.00    100.00         100.00                    Also looks okay
```

explore your data

```
library(ggplot2)
library(GGally)
myvars <- c("y_sbp", "scl","age", "bmi", "female01")                  # TIP - put yvariable first for output readability

# Pairwise correlations - all vars must be numeric
cor(ready[myvars])

##                  y_sbp        scl       age         bmi      female01
## y_sbp      1.0000000000 0.23074225 0.3951681  0.34247354 -0.0002310055
## scl        0.2307422501 1.00000000 0.2806521  0.14720909  0.0222176901
## age        0.3951680608 0.28065213 1.0000000  0.18044683  0.0253892950       Strongest corr with y_sbp is with age
## bmi        0.3424735396 0.14720909 0.1804468  1.00000000 -0.0707072708       2nd strongest corr with y_sbp is with bmi
## female01 -0.0002310055 0.02221769 0.0253893 -0.07070727  1.0000000000

# Pairwise correlations - factor variables allowed
GGally::ggpairs(data=ready,
  columns=c("y_sbp", "scl", "age", "bmi", "female01f")) +
  ggtitle("Pairwise Correlations")
```



I find this display to be more informative

6. Fit Models and Display Side-by-Side Using stargazer( ) in {stargazer}

*fit models**

```
library(stargazer)                                                    # stargazer() in package {stargazer}

# 4  single predictor models                                          # save models as objects we can compare.
m_scl <- lm(y_sbp ~ scl, data=ready)
m_age <- lm(y_sbp ~ age, data=ready)
m_bmi <- lm(y_sbp ~ bmi, data=ready)
m_sex <- lm(y_sbp ~ female01f, data=ready)

# Side by side comparison - BASIC
stargazer(m_scl, m_age, m_bmi, m_sex, type="text")                     # basic display:  betas and SE(beta)

##
## ===============================================================================
##                                         Dependent variable:
##                                 ----------------------------------------
##                                                 y_sbp
##                                    (1)        (2)        (3)        (4)
## -------------------------------------------------------------------------------
## scl                             0.00004***
##                                 (0.00000)
##
## age                                        0.0003***                           Best single predictor model
##                                            (0.00001)                           appears to be the one with X = age
##
## bmi                                                    0.001***
##                                                       (0.00002)
##
## female01ffemale                                                     -0.00000
##                                                                     (0.0002)
##
## Constant                        -0.096***  -0.102***  -0.102***  -0.088***
##                                  (0.001)    (0.001)    (0.001)   (0.0002)
##
## -------------------------------------------------------------------------------
## Observations                      4,658      4,658      4,658      4,658
## R2                                0.053      0.156      0.117     0.00000
## Adjusted R2                       0.053      0.156      0.117     -0.0002
## Residual Std. Error (df = 4656)   0.007      0.006      0.006      0.007
## F Statistic (df = 1; 4656)      261.835*** 861.619*** 618.654***  0.0002
## ===============================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01

# Side by side comparison - User defined options

stargazer(m_scl, m_age, m_bmi, m_sex, type="text",                     # TIP Put each option on its own line
          font.size="small",
          align=TRUE,
          ci=TRUE, intercept.bottom=FALSE,
          covariate.labels=c("Intercept",
                             "Serum cholesterol (mg/dL)",
                             "Age at Baseline (years)",
                             "Body Mass Index (kg/m2)",
                             "Female sex at birth"),
          dep.var.labels=c("y_sbp:  -1/sqrt(sbp)"),
          title="Single Predictor Models:  Betas (95% CI)")
```

```
##
## Single Predictor Models: Betas (95% CI)    In screening models, I prefer betas with 95% CI over se(beta)
## ================================================================================================
##                                                 Dependent variable:
##                                 ---------------------------------------------------------------
##                                                    y-1/sqrt(sbp)
##                                       (1)              (2)              (3)              (4)
## ------------------------------------------------------------------------------------------------
## Intercept                          -0.096***        -0.102***        -0.102***        -0.088***
##                                 (-0.097, -0.095) (-0.103, -0.101) (-0.104, -0.101) (-0.088, -0.087)
##
## Serum cholesterol (mg/dL)          0.00004***
##                                 (0.00003, 0.00004)
##
## Age at Baseline (years)                             0.0003***
##                                                  (0.0003, 0.0003)
##
## Body Mass Index (kg/m2)                                               0.001***
##                                                                    (0.001, 0.001)
##
## Female sex at birth                                                                  -0.00000
##                                                                                   (-0.0004, 0.0004)
##
## ------------------------------------------------------------------------------------------------
## Observations                         4,658            4,658            4,658            4,658
## R2                                   0.053            0.156            0.117           0.00000
## Adjusted R2                          0.053            0.156            0.117           -0.0002
## Residual Std. Error (df = 4656)      0.007            0.006            0.006            0.007
## F Statistic (df = 1; 4656)         261.835***       861.619***       618.654***         0.0002
## ================================================================================================
## Note:                                                                *p<0.1; **p<0.05; ***p<0.01
```

fit models

```
library(stargazer)                                                    # stargazer() in package {stargazer}

# saturated model (m1) + all 2-predictor models (m2, m3 and m4)
m1 <- lm(y_sbp ~ scl + age + bmi, data=ready)
m2 <- lm(y_sbp ~ age + bmi, data=ready)
m3 <- lm(y_sbp ~ scl + bmi, data=ready)
m4 <- lm(y_sbp ~ scl + age, data=ready)


# Side-by-side comparison: betas (95% CI), move intercept to top, add some labels
stargazer(m1, m2, m3, m4,type="text",
          font.size="small",
          align=TRUE,
          ci=TRUE,
          intercept.bottom=FALSE,
          covariate.labels=c("Intercept",
                             "Serum cholesterol (mg/dL)",
                             "Age at Baseline (years)",
                              "Body Mass Index (kg/m2)",
                              "Female sex at birth"),
           dep.var.labels=c("y_sbp:  -1/sqrt(sbp)"),
           title="Multiple Predictor Models:  Betas (95% CI)")
```

```
##
## Multiple Predictor Models: Betas (95% CI)
## ================================================================================================================================
##                                                          Dependent variable:
##                          ------------------------------------------------------------------------------------------------------
##                                                              y-1/sqrt(sbp)
##                                  (1)                      (2)                      (3)                      (4)
## --------------------------------------------------------------------------------------------------------------------------------
## Intercept                      -0.115***                -0.113***                -0.108***                -0.106***
##                          (-0.116, -0.113)         (-0.114, -0.111)         (-0.109, -0.106)         (-0.107, -0.104)
##
## Serum cholesterol (mg/dL)      0.00002***                                        0.00003***               0.00002***
##                          (0.00001, 0.00002)                                (0.00002, 0.00003)       (0.00002, 0.00002)
##
## Age at Baseline (years)        0.0003***                0.0003***                                         0.0003***
##                          (0.0002, 0.0003)         (0.0003, 0.0003)                                  (0.0003, 0.0003)
##
## Body Mass Index (kg/m2)        0.0005***                0.0005***                0.001***
##                          (0.0004, 0.0005)         (0.0004, 0.001)          (0.0005, 0.001)
##
## --------------------------------------------------------------------------------------------------------------------------------
## Observations                    4,658                    4,658                    4,658                    4,658
## R2                              0.242                    0.232                    0.151                    0.172
## Adjusted R2                     0.241                    0.232                    0.150                    0.171
## Residual Std. Error       0.006 (df = 4654)        0.006 (df = 4655)        0.006 (df = 4655)        0.006 (df = 4655)
## F Statistic           494.215*** (df = 3; 4654) 703.746*** (df = 2; 4655) 412.432*** (df = 2; 4655) 482.631*** (df = 2; 4655)
## ================================================================================================================================
## Note:                                                                                  *p<0.1; **p<0.05; ***p<0.01
```

At a glance (looking at the adjusted R2), it looks like the better fit is either model 1 or 2. Model 2 has predictors = age + bmi, whereas model 1 has predictors = age + bmi + serum cholesterol. The question is then: given age and bmi are in the model, is the inclusion of the extra predictor serum cholesterol statistically significant? For this we need a Partial F-test, as described below.

7. Partial F-Test to Compare Hierarchical Models

compare models

```
# saturated model:  betas and 95% CI
round(cbind(coef(m1),confint(m1)), digits=4)
```

Just a reminder of how to obtain the betas and 95% CI

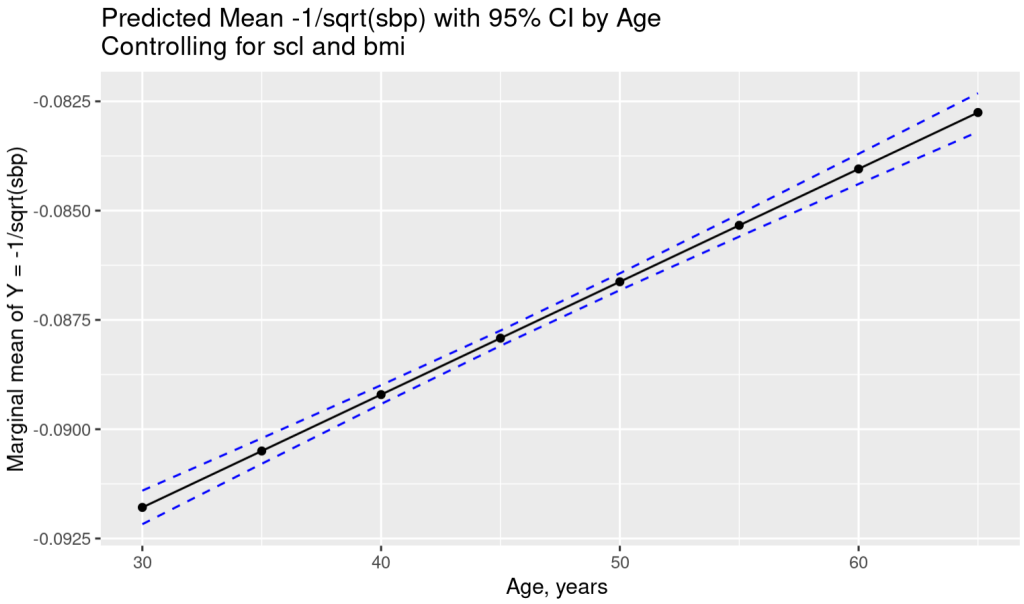
```
##                       2.5 %  97.5 %
## (Intercept) -0.1148 -0.1163 -0.1134
## scl          0.0000  0.0000  0.0000
## age          0.0003  0.0002  0.0003
## bmi          0.0005  0.0004  0.0005

# saturated model:  anova table
round(anova(m1), digits=4)

## Analysis of Variance Table
##
## Response: y_sbp
##             Df Sum Sq Mean Sq F value               Pr(>F)
## scl          1 0.0118  0.0118  326.73 < 0.00000000000000022 ***    serum cholesterol is significant in saturated model
## age          1 0.0262  0.0262  727.22 < 0.00000000000000022 ***
## bmi          1 0.0155  0.0155  428.70 < 0.00000000000000022 ***
## Residuals 4654 0.1680  0.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Partial F-test of hierarchical models m1 (reduced) versus m2 (full)
cat("\nPartial F-Test\nControlling for: age, bmi\nSignificance of scl\n")

##
## Partial F-Test
## Controlling for: age, bmi
## Significance of scl

round(anova(m2,m1), digits=4)

## Analysis of Variance Table
##
## Model 1: y_sbp ~ age + bmi
## Model 2: y_sbp ~ scl + age + bmi
##   Res.Df   RSS Df Sum of Sq      F               Pr(>F)
## 1   4655 0.170                                          Partial-F Statistic = 57.936
## 2   4654 0.168  1    0.0021 57.936 < 0.00000000000000022 ***       Controlling for age and bmi, scl is significant
## ---                                                     Beware:  sometimes significance reflects large n
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Report R-squared, adjusted R-squared, AIC, BIC
sm1 <- summary(m1)                                      # save summary(m1) as object sm1 for later use
sm2 <- summary(m2)                                      # save summary(m1) as object sm1 for later use

#---  Step 1.  Get R-squared, adjusted R-squared, AIC, BIC
R_Squared<- c(sm1$r.squared,sm2$r.squared)                          # vector of R squared values
Adjusted_Rsquared <- c(sm1$adj.r.squared,sm2$adj.r.squared)         # vector of adjusted R Squares
AIC <- c(AIC(m1),AIC(m2))                                           # Note use of function AIC()
BIC <- c(BIC(m1), BIC(m2))

# ---  Step 2.  Put results into a data frame and add column to show model names
Model <- c("Saturated", "Two Predictor")                            # vector of model names
results <- data.frame(Model,R_Squared,Adjusted_Rsquared, AIC, BIC)  # create data frame called results
cat("\nComparison of Saturated Model v Two Predictor Model\n")
results

##
## Comparison of Saturated Model v Two Predictor Model

##           Model R_Squared Adjusted_Rsquared       AIC       BIC
## 1     Saturated  0.241605         0.2411162 -34424.43 -34392.20          Despite the significance of the Partial F
## 2 Two Predictor  0.232164         0.2318341 -34368.80 -34343.02          it's not clear that scl adds much.
```

8. Plot Marginal Predictions

One of the important uses of a fitted multiple predictor model is prediction. The following is an illustration of R code to produce predicted values of the response Y = -1/sqrt(sbp) at a selection of 8 values of **age** while holding the other predictors in the model (**scl** and **bmi**) at their means

```
library(ggplot2)

#1. Predicted mean Y v X=age with other vars set to their means
#1a.  Data for plot
newage <- data.frame(age=c(30,35,40,45,50,55,60,65),              # X = user sets 8 values of age
                     scl=rep(mean(ready$scl),8),                  # scl at its mean for all 8 values of age
                     bmi=rep(mean(ready$bmi),8))                  # bmi at its mean for all 8 values of age
yhat <- predict(m1, newdata=newage, interval="confidence")        # Y = predicted fit w CI
age <- newage$age                                                 # To obtain variable name
plotdata <- data.frame(cbind(age, yhat))                          # ggplot wants dataframe

names(plotdata)                                                  # names() shows you yhat is stored as object fit

## [1] "age" "fit" "lwr" "upr"

#1b.  Plot.
ggplot(data=plotdata) +
  aes(x=age, y=fit) +
  geom_point() +                                                  # Points are predicted means
  stat_smooth(method = lm, size=0.5, color="black") +
     geom_line(aes(y = lwr), color = "blue", linetype = "dashed") +  # lower CI
     geom_line(aes(y = upr), color = "blue", linetype = "dashed") +  # upper CI

   ggtitle("Predicted Mean -1/sqrt(sbp) with 95% CI by Age\nControlling for scl and bmi")  +
          xlab("Age, years ") +
          ylab("Marginal mean of Y = -1/sqrt(sbp)")  +
          labs(caption = "Covariates scl and bmi at their means") +          # footnote
          theme(plot.caption = element_text(hjust = 0, face = "italic"))     # put footnote lower left
```



9. Report

Multiple linear regression was used to explore the relationship of systolic blood pressure [y = -1/sqrt(sbp)] to four variables (sex at birth, body mass index, serum cholesterol and age) in a sample of n = 4659 participants of the Framingham Heart Study who provided complete data on all study variables. Preliminary data exploration did not reveal substantial collinearity among the candidate predictors. In one predictor modeling, sex at birth was uncorrelated with systolic blood pressure (R^2 = 0, approximately) and was not considered further. The saturated model with the 3 remaining predictors (age, bmi, serum cholesterol) was then compared with each of the three 2-predictor models and suggested that a reasonable "good" model is the 2-predictor model utilizing age and bmi. The extra inclusion of serum cholesterol did achieve statistical significance, however (Partial-F Test p-value <<< .0001), but it is unclear if this of biological significance or an artifact of the large sample size (n=4658). There is no evidence of confounding of either the systolic blood pressure – age relationship or the systolic blood pressure – BMI relationship. The betas and SE's for these predictors are not changed after controlling for serum cholesterol (beta for age = .0003, beta for BMI = .0005).