```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
birth <- read.csv("data/US_births(2018).csv")
```

```r
head(birth)
```

```
##   ATTEND BFACIL  BMI CIG_0 DBWT DLMP_MM DLMP_YY DMAR DOB_MM DOB_TT DOB_WK
## 1      1      1 30.7     0 3657       4    2017    1      1   1227      2
## 2      1      1 33.3     2 3242      99    9999    2      1   1704      2
## 3      1      1 30.0     0 3470       4    2017    1      1    336      2
## 4      3      1 23.7     0 3140       5    2017    2      1    938      2
## 5      1      1 35.5     0 2125      99    9999    1      1    830      3
## 6      4      2 31.3     0 4082       3    2017    1      1     28      2
##   DOB_YY DWgt_R FAGECOMB FEDUC FHISPX FRACE15 FRACE31 FRACE6 ILLB_R ILOP_R
## 1   2018    231       31     3      1       1       1      1     16     33
## 2   2018    185       35     4      0       3       3      3    180    888
## 3   2018    273       31     4      0       1       1      1    999    888
```

```
## 4   2018    138       26     2      0       3       3      3     43    888
## 5   2018    219       35     3      0       2       2      2    999    999
## 6   2018    247       28     6      6       1       1      1     39    888
##   ILP_R IMP_SEX IP_GON LD_INDL MAGER MAGE_IMPFLG MAR_IMP MBSTATE_REC MEDUC
## 1    16      NA      N       N    30          NA      NA           1     6
## 2   180      NA      N       N    35          NA      NA           1     9
## 3   999      NA      N       N    28          NA      NA           1     6
## 4    43      NA      N       N    23          NA      NA           1     2
## 5   999      NA      N       N    37          NA      NA           1     4
## 6    39      NA      N       N    26          NA      NA           1     6
##   MHISPX MM_AICU MRACE15 MRACE31 MRACEIMP MRAVE6 MTRAN M_Ht_In NO_INFEC
## 1      0       N       1       1       NA      1     N      66        1
## 2      0       N       3       3       NA      3     N      63        1
## 3      0       N       1       1       NA      1     N      71        1
## 4      0       N       3       3       NA      3     N      64        1
## 5      0       N       1       1       NA      1     N      66        1
## 6      0       N       1       1       NA      1     N      67        1
##   NO_MMORB NO_RISKS PAY PAY_REC PRECARE PREVIS PRIORDEAD PRIORLIVE PRIORTERM
## 1        1        1   2       2       3      8         0         1         2
## 2        1        0   1       1       3      9         0         2         0
## 3        1        0   5       4       5     17         0         1         0
## 4        1        1   1       1       5      6         0         2         0
## 5        1        1   1       1       5     15         0         1         4
## 6        1        1   2       2       2     13         0         1         0
##   PWgt_R RDMETH_REC RESTATUS RF_CESAR RF_CESARN SEX WTGAIN
## 1    190          1        2        N         0   M     41
## 2    188          4        2        Y         2   F      0
## 3    215          1        1        N         0   M     58
## 4    138          1        2        N         0   F      0
## 5    220          3        1        N         0   M      0
## 6    200          1        1        N         0   F     47
```

```
nrow(birth)
```

```
## [1] 3801534
```

```
# Remove missing values

# remove missing values in the response variable
clean_birth <- subset(birth, DBWT != 9999)

# remove missing values in the features to be considered for adding interactions
clean_birth <- subset(clean_birth, PRECARE != 99 & CIG_0 != 99 & BMI != 99.9
               & PREVIS != 99 & MRAVE6 != 9 & PAY_REC != 9
               & FRACE6 != 9 & MEDUC != 9 & FEDUC != 9
               & NO_RISKS != 9)

# remove missing values in the features not to be considered for adding interactions
clean_birth <- subset(clean_birth, ATTEND != 9 & BFACIL != 9 & FAGECOMB != 99
               & RF_CESAR != "U" & LD_INDL != "U" & MBSTATE_REC != 3
               & M_Ht_In != 99 & NO_INFEC != 9 & NO_MMORB != 9
               & PRIORLIVE != 99 & PRIORTERM != 99 & RDMETH_REC != 9)
```

```r
clean_birth <- clean_birth %>% filter(!is.na(DMAR))

# remove missing values in the features for feature engineering
clean_birth <- subset(clean_birth, DLMP_YY != 9999 & DLMP_MM != 99)
clean_birth <- subset(clean_birth, PWgt_R != 999  & WTGAIN != 99)
clean_birth <- subset(clean_birth, ILLB_R != 999)
```

```r
nrow(clean_birth)
```

```
## [1] 2354840
```

```r
# Feature engineering

# estimate pregnancy length
clean_birth$PREG_LEN <- 12*(2018 - clean_birth$DLMP_YY) +
                        (clean_birth$DOB_MM - clean_birth$DLMP_MM)

# categorize and cap pregnancy length
clean_birth$PREG_LEN[clean_birth$PREG_LEN < 8] <- -1
clean_birth$PREG_LEN[clean_birth$PREG_LEN > 10] <- 99
clean_birth$PREG_LEN <- factor(clean_birth$PREG_LEN)
levels(clean_birth$PREG_LEN) <- c("Early", "8", "9", "10", "Late")

# recode PRECARE
clean_birth$PRECARE[clean_birth$PRECARE < 4 & clean_birth$PRECARE > 0] <- 1
clean_birth$PRECARE[clean_birth$PRECARE < 7 & clean_birth$PRECARE > 3] <- 2
clean_birth$PRECARE[ clean_birth$PRECARE > 6] <- 3

# compute percentage weight gain
clean_birth$WTGAIN_PER <- clean_birth$WTGAIN / clean_birth$PWgt_R

# binarize CIG_0
clean_birth$CIG_0 <- ifelse(clean_birth$CIG_0 > 0, TRUE, FALSE)

# binarize PRIORDEAD
clean_birth$PRIORDEAD <- ifelse(clean_birth$PRIORDEAD > 0, TRUE, FALSE)

# binarize PRIORTERM
clean_birth$PRIORTERM <- ifelse(clean_birth$PRIORTERM > 0, TRUE, FALSE)

# binarize PRIORLIVE
clean_birth$PRIORLIVE <- ifelse(clean_birth$PRIORLIVE > 0, TRUE, FALSE)

# compute first time live birth
clean_birth$FIRST_BIRTH <- ifelse(clean_birth$ILLB_R == 888, TRUE, FALSE)
```

```r
# Reduce the dimensionality of the dataset

# drop columns where >99% entries are the same
clean_birth <- clean_birth %>% dplyr::select(!c(DOB_YY, IMP_SEX, IP_GON, MAGE_IMPFLG,
                                  MAR_IMP, MM_AICU, MTRAN))
```

```r
# drop redundant columns due to feature engineering
clean_birth <- clean_birth %>% dplyr::select(!c(WTGAIN, PWgt_R, DWgt_R, DOB_MM,
                                    DOB_WK, DOB_TT, DOB_MM, DLMP_YY,
                                    DLMP_MM, PAY, MHISPX, MRACE15,
                                    MRACE31, MRACEIMP, FHISPX, FRACE15,
                                    FRACE31, RF_CESARN, ILOP_R, ILP_R, ILLB_R))
```

```r
# write.csv(clean_birth, "data/clean_birth.csv", row.names = FALSE)
```

```r
# Factorize categorical variables
clean_birth <- clean_birth %>% mutate_if(is.character, as.factor)
clean_birth <- clean_birth %>% mutate_if(is.logical, as.factor)
clean_birth <- clean_birth %>% mutate(ATTEND = factor(ATTEND), BFACIL = factor(BFACIL),
                                    DMAR = factor(DMAR), FEDUC = factor(FEDUC),
                                    FRACE6 = factor(FRACE6), MBSTATE_REC = factor(MBSTATE_REC),
                                    MEDUC = factor(MEDUC), MRAVE6 = factor(MRAVE6),
                                    NO_INFEC = factor(NO_INFEC),NO_MMORB = factor(NO_MMORB),
                                    NO_RISKS = factor(NO_RISKS), PAY_REC = factor(PAY_REC),
                                    PRECARE = factor(PRECARE), RDMETH_REC = factor(RDMETH_REC),
                                    RESTATUS = factor(RESTATUS))
```

```r
# Subsample datasets

set.seed(151)
EDA_size = 3000
Train_size = 100000
Test_size = 100000
EDA_df <- clean_birth %>% slice_sample(n = EDA_size, replace = TRUE)
Train <- clean_birth %>% slice_sample(n = Train_size, replace = TRUE)
Test <- clean_birth %>% slice_sample(n = Test_size, replace = TRUE)
```

```r
# EDA
# TODO
```

```r
# Second time feature engineering from EDA

# Binarize PRECARE
EDA_df$PRECARE <- ifelse(EDA_df$PRECARE != 0, TRUE, FALSE)
Train$PRECARE <- ifelse(Train$PRECARE != 0, TRUE, FALSE)
Test$PRECARE <- ifelse(Test$PRECARE != 0, TRUE, FALSE)
```

```r
# write.csv(EDA_df, "data/EDA.csv", row.names = FALSE)
# write.csv(Train, "data/Train.csv", row.names = FALSE)
# write.csv(Test, "data/Test.csv", row.names = FALSE)
```

```r
# Model Selection

biggest.model <- lm(DBWT ~ ., data = Train)
# summary(biggest.model)

# Remove the columns causing singularity
Train <- Train %>% dplyr::select(!c(RF_CESAR))
```

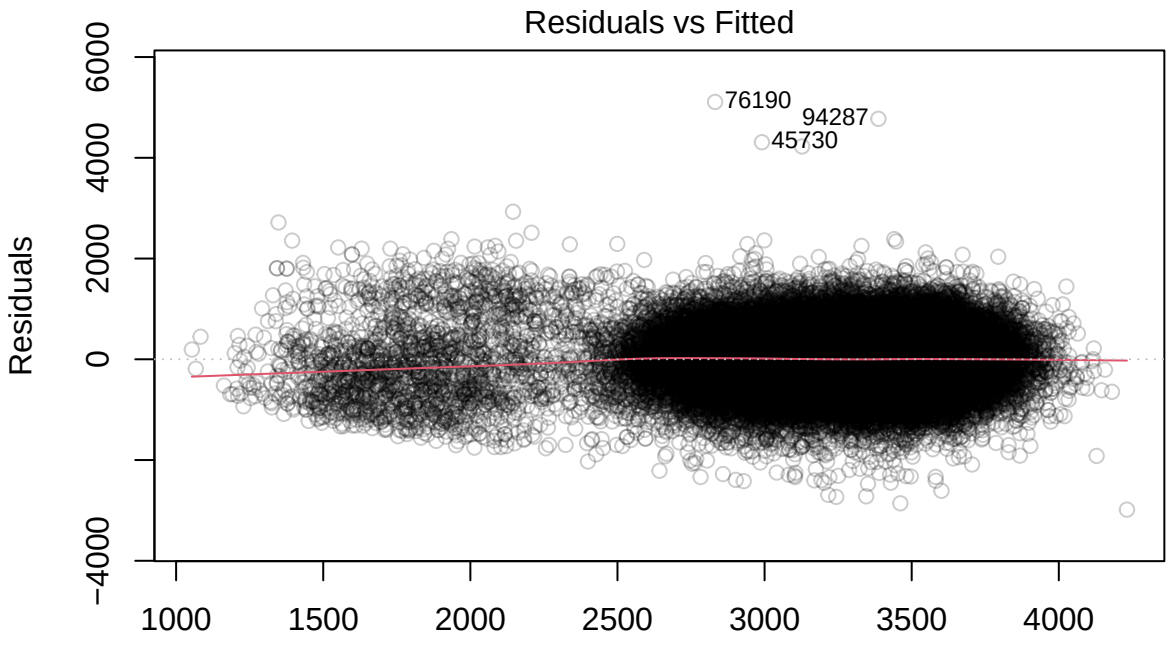
```r
biggest.model <- lm(DBWT ~ ., data = Train)
min.model <- lm(DBWT ~ 1, data = Train)
# summary(biggest.model)
# Forward selection with BIC
forward.BIC = step(min.model, direction="forward", scope = formula(biggest.model),
                   k = log(nrow(Train)), trace = 0)

# Backward selection with BIC
backward.BIC = step(biggest.model, direction="backward",
                  k = log(nrow(Train)), trace = 0)

# Forward selection with AIC
forward.AIC = step(min.model, direction="forward", scope = formula(biggest.model),
                  k = 2, trace = 0)

# Backward selection with AIC
backward.AIC = step(biggest.model, direction="backward",
                  k = 2, trace = 0)
```

```r
# Compute the leave-one-out cross-validation errors
for_AIC.cv = mean((residuals(forward.AIC) / (1 - hatvalues(forward.AIC))) ^ 2)
back_AIC.cv = mean((residuals(backward.AIC) / (1 - hatvalues(backward.AIC))) ^ 2)
for_BIC.cv = mean((residuals(forward.BIC) / (1 - hatvalues(forward.BIC))) ^ 2)
back_BIC.cv = mean((residuals(backward.BIC) / (1 - hatvalues(backward.BIC))) ^ 2)
which.min(c(for_AIC.cv, back_AIC.cv, for_BIC.cv, back_BIC.cv))
```

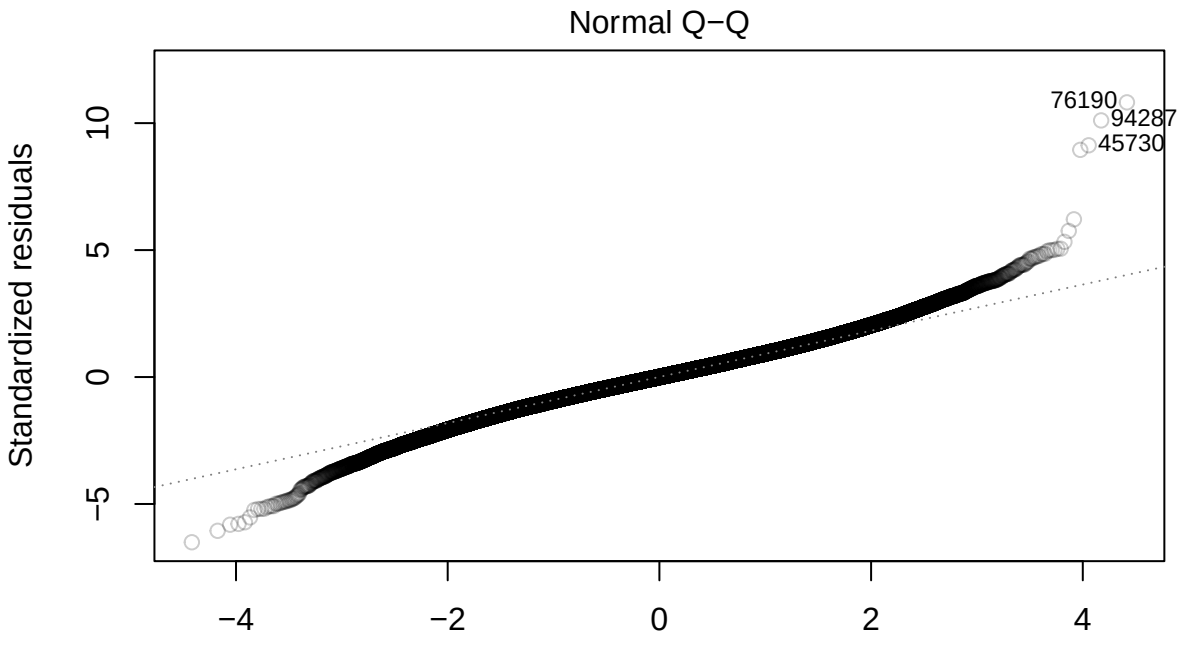
```r
# Add interaction terms by F-test
full.lm <- lm(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER +
    PRIORLIVE + CIG_0 + NO_RISKS + RDMETH_REC + PREVIS + ATTEND +
    MBSTATE_REC + FRACE6 + PAY_REC + LD_INDL + FEDUC + NO_MMORB +
    BFACIL + FAGECOMB + NO_INFEC + RESTATUS + MEDUC + PRECARE +
    DMAR + BMI * PRECARE + WTGAIN_PER * PRECARE + PRECARE * MEDUC +
    PREVIS * PREG_LEN + PREG_LEN * MEDUC + PRECARE * CIG_0 + CIG_0 * SEX +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Train)

# Type II Anova
Anova(full.lm)
```

```r
# Final model
final.lm <- lm(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER +
    PRIORLIVE + CIG_0 + NO_RISKS + RDMETH_REC + PREVIS + ATTEND +
    MBSTATE_REC + FRACE6 + PAY_REC + LD_INDL + FEDUC + NO_MMORB +
    BFACIL + FAGECOMB + NO_INFEC + RESTATUS + MEDUC + PRECARE +
    DMAR + PREVIS * PREG_LEN + PREG_LEN * MEDUC + CIG_0 * PRECARE +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Train)
```

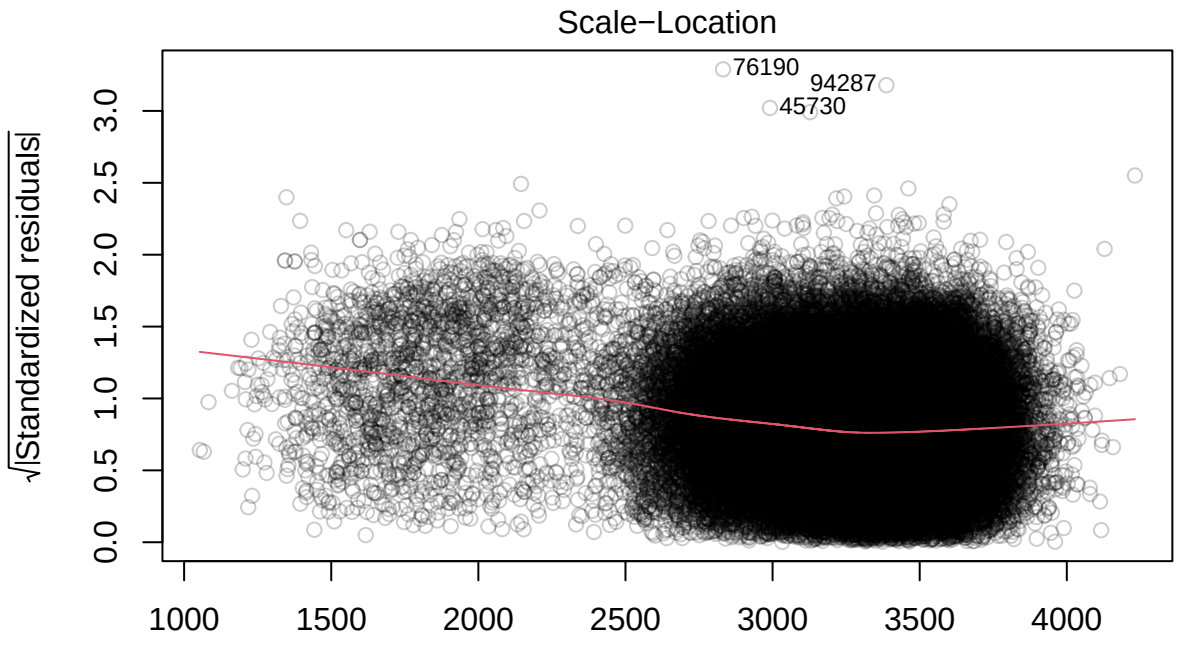
```r
# Model diagnostic
plot(final.lm, col = rgb(red = 0, green = 0, blue = 0, alpha = 0.2))
```



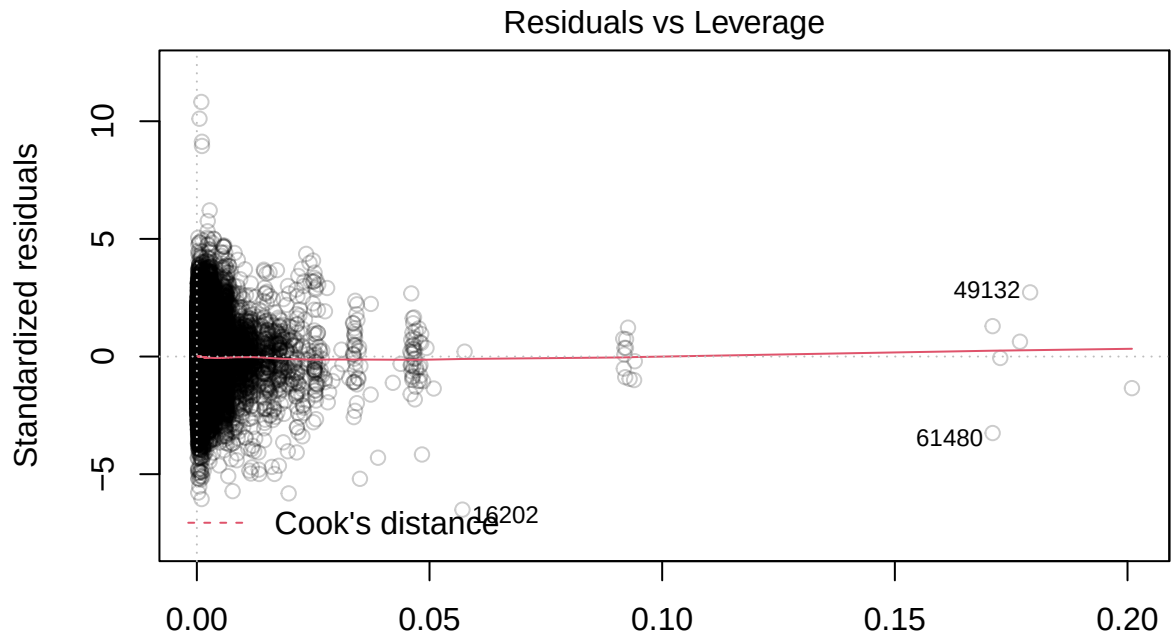
n(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER + PRIORL



n(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER + PRIORL



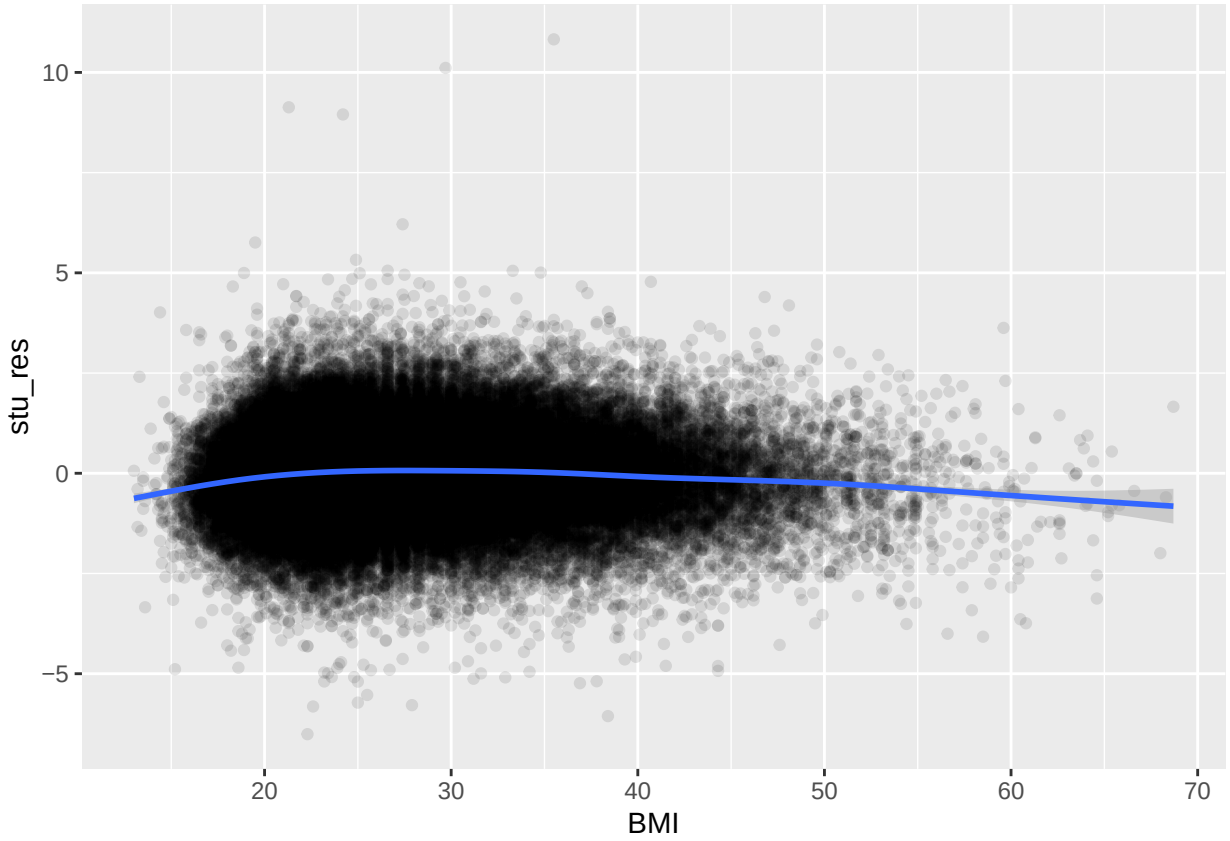
n(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER + PRIORL



n(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER + PRIORL

```
# Potential outliers: 76190, 94287, 45730

# compute studentized residuals
stu_res <- studres(final.lm)
Train <- cbind(Train, stu_res)
stu_res_dec <- stu_res[order(abs(stu_res), decreasing = TRUE)]

# Check linearity and constant variance
ggplot(Train, aes(x = BMI, y = stu_res)) +
    geom_point(alpha = 0.1) +
    geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
# Outliers

# test the largest studentized residual
alpha = 0.5
p_value <- pt(stu_res_dec[1], df = final.lm$df.residual - 1, lower.tail = FALSE)
p_value < alpha / nrow(Train) # Bonferroni Correction
```

```
## 76190
##  TRUE
```

```
# check observations with top 5 largest studentized residuals
Train[head(names(stu_res_dec)),]
```

```
##       ATTEND BFACIL  BMI CIG_0 DBWT DMAR FAGECOMB FEDUC FRACE6 LD_INDL MAGER
## 76190      1      1 35.5 FALSE 7940    1       36     6      1       N    34
## 94287      1      1 29.7 FALSE 8160    1       31     6      1       N    38
## 45730      1      1 21.3 FALSE 7300    1       25     4      2       N    22
## 34402      1      1 24.2 FALSE 7352    2       35     3      1       Y    35
## 16202      1      1 22.3 FALSE 1247    2       22     4      2       N    24
## 82652      3      1 27.4 FALSE 5075    1       36     3      1       N    36
##       MBSTATE_REC MEDUC MRAVE6 M_Ht_In NO_INFEC NO_MMORB NO_RISKS PAY_REC
## 76190           1     7      1      64        1        1        0       2
## 94287           1     8      1      66        1        1        1       2
## 45730           2     4      1      63        1        1        1       3
## 34402           1     5      1      60        1        1        1       2
```

```
## 16202           1     4      2      61        1        1        1       1
## 82652           1     6      1      62        1        1        1       2
##       PRECARE PREVIS PRIORDEAD PRIORLIVE PRIORTERM RDMETH_REC RESTATUS RF_CESAR
## 76190    TRUE     15     FALSE     FALSE      TRUE          3        1        N
## 94287    TRUE     12     FALSE      TRUE      TRUE          1        1        N
## 45730    TRUE     10     FALSE      TRUE     FALSE          1        2        N
## 34402    TRUE     14     FALSE     FALSE      TRUE          1        1        N
## 16202    TRUE     60     FALSE      TRUE      TRUE          3        1        N
## 82652    TRUE      8     FALSE      TRUE     FALSE          1        1        N
##       SEX PREG_LEN WTGAIN_PER FIRST_BIRTH   stu_res
## 76190   F        8 0.16425121        TRUE 10.825073
## 94287   F        9 0.08695652       FALSE 10.112046
## 45730   M        8 0.23333333       FALSE  9.130358
## 34402   M        8 0.51612903        TRUE  8.951312
## 16202   M    Early 0.30508475       FALSE -6.507789
## 82652   M    Early 0.36666667       FALSE  6.212057
```

```
# Influential Points
Train[c(16202, 61480, 49132),]
```

```
##       ATTEND BFACIL  BMI CIG_0 DBWT DMAR FAGECOMB FEDUC FRACE6 LD_INDL MAGER
## 16202      1      1 22.3 FALSE 1247    2       22     4      2       N    24
## 61480      1      1 23.8 FALSE 1760    2       32     4      1       N    33
## 49132      1      1 29.6 FALSE 4082    1       28     2      2       N    28
##       MBSTATE_REC MEDUC MRAVE6 M_Ht_In NO_INFEC NO_MMORB NO_RISKS PAY_REC
## 16202           1     4      2      61        1        1        1       1
## 61480           2     4      1      65        1        1        1       1
## 49132           2     2      2      65        1        1        1       1
##       PRECARE PREVIS PRIORDEAD PRIORLIVE PRIORTERM RDMETH_REC RESTATUS RF_CESAR
## 16202    TRUE     60     FALSE      TRUE      TRUE          3        1        N
## 61480   FALSE      0     FALSE     FALSE      TRUE          1        1        N
## 49132   FALSE      0     FALSE      TRUE     FALSE          1        1        N
##       SEX PREG_LEN WTGAIN_PER FIRST_BIRTH   stu_res
## 16202   M    Early 0.30508475       FALSE -6.507789
## 61480   M     Late 0.21678322        TRUE -3.251874
## 49132   M     Late 0.01685393       FALSE  2.726319
```

```
# Model Interpretation (Causual Inference)
final_test.lm <- lm(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER +
    PRIORLIVE + CIG_0 + NO_RISKS + RDMETH_REC + PREVIS + ATTEND +
    MBSTATE_REC + FRACE6 + PAY_REC + LD_INDL + FEDUC + NO_MMORB +
    BFACIL + FAGECOMB + NO_INFEC + RESTATUS + MEDUC + PRECARE +
    DMAR + PREVIS * PREG_LEN + PREG_LEN * MEDUC + CIG_0 * PRECARE +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Test)

summary(final_test.lm)
```

```
##
## Call:
## lm(formula = DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI +
##     WTGAIN_PER + PRIORLIVE + CIG_0 + NO_RISKS + RDMETH_REC +
##     PREVIS + ATTEND + MBSTATE_REC + FRACE6 + PAY_REC + LD_INDL +
##     FEDUC + NO_MMORB + BFACIL + FAGECOMB + NO_INFEC + RESTATUS +
```

```
##     MEDUC + PRECARE + DMAR + PREVIS * PREG_LEN + PREG_LEN * MEDUC +
##     CIG_0 * PRECARE + PRECARE * PREG_LEN + CIG_0 * PREG_LEN,
##     data = Test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3225.4  -286.8     0.0   292.7  4736.6
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -446.8676    96.3540  -4.638 3.53e-06 ***
## PREG_LEN8                 771.3050    97.7757   7.889 3.09e-15 ***
## PREG_LEN9                1079.8240    89.5016  12.065  < 2e-16 ***
## PREG_LEN10               1167.9886   102.7412  11.368  < 2e-16 ***
## PREG_LENLate              412.3407   241.6996   1.706 0.088010 .
## M_Ht_In                    30.1895     0.5617  53.751  < 2e-16 ***
## MRAVE62                  -115.5742     7.8106 -14.797  < 2e-16 ***
## MRAVE63                    57.8392    19.3229   2.993 0.002761 **
## MRAVE64                   -14.9299    10.9812  -1.360 0.173964
## MRAVE65                   -57.9029    37.2216  -1.556 0.119801
## MRAVE66                   -40.0173    10.1873  -3.928 8.57e-05 ***
## SEXM                      116.8002     2.9905  39.057  < 2e-16 ***
## BMI                        17.6692     0.2887  61.193  < 2e-16 ***
## WTGAIN_PER                886.2944    16.2828  54.431  < 2e-16 ***
## PRIORLIVETRUE             101.8852     3.5075  29.048  < 2e-16 ***
## CIG_0TRUE                 -36.2658    60.5929  -0.599 0.549497
## NO_RISKS1                 135.7091     4.3023  31.543  < 2e-16 ***
## RDMETH_REC2               115.2483    11.2566  10.238  < 2e-16 ***
## RDMETH_REC3               -39.3779     4.1728  -9.437  < 2e-16 ***
## RDMETH_REC4               126.5207     6.1584  20.544  < 2e-16 ***
## PREVIS                     49.2285     2.1659  22.729  < 2e-16 ***
## ATTEND2                     1.0809     5.4209   0.199 0.841957
## ATTEND3                    50.8428     5.4605   9.311  < 2e-16 ***
## ATTEND4                    70.2091    20.5071   3.424 0.000618 ***
## ATTEND5                    43.5789    19.1014   2.281 0.022524 *
## MBSTATE_REC2               53.1461     4.5901  11.578  < 2e-16 ***
## FRACE62                   -57.7418     7.3991  -7.804 6.06e-15 ***
## FRACE63                    29.0755    19.9377   1.458 0.144757
## FRACE64                  -129.7002    11.1709 -11.611  < 2e-16 ***
## FRACE65                    39.0590    35.7301   1.093 0.274323
## FRACE66                   -33.3422    10.1685  -3.279 0.001042 **
## PAY_REC2                   17.9928     4.0698   4.421 9.83e-06 ***
## PAY_REC3                   34.6201     8.9454   3.870 0.000109 ***
## PAY_REC4                   21.5713     8.3915   2.571 0.010154 *
## LD_INDLY                   35.1811     3.4800  10.110  < 2e-16 ***
## FEDUC2                    -37.7633    12.3375  -3.061 0.002208 **
## FEDUC3                    -16.9855    11.8353  -1.435 0.151244
## FEDUC4                     -3.9415    12.1874  -0.323 0.746385
## FEDUC5                     10.4299    12.9450   0.806 0.420414
## FEDUC6                     18.3435    12.4402   1.475 0.140341
## FEDUC7                     16.4602    13.2590   1.241 0.214446
## FEDUC8                     17.5042    14.7607   1.186 0.235677
## NO_MMORB1                 -83.8594    12.4536  -6.734 1.66e-11 ***
## BFACIL2                    60.1776    19.4558   3.093 0.001982 **
```

```
## BFACIL3                    98.9886    20.1652   4.909 9.17e-07 ***
## BFACIL4                  -122.2570    50.5680  -2.418 0.015622 *
## BFACIL5                     7.8084   136.8790   0.057 0.954509
## BFACIL6                    51.5385   101.2064   0.509 0.610584
## BFACIL7                    26.7850    57.1495   0.469 0.639298
## FAGECOMB                   -0.7553     0.2595  -2.911 0.003607 **
## NO_INFEC1                   7.6718    10.6255   0.722 0.470288
## RESTATUS2                 -15.0041     3.3041  -4.541 5.60e-06 ***
## RESTATUS3                 -20.9953     9.1247  -2.301 0.021397 *
## RESTATUS4                 -55.4778    30.4563  -1.822 0.068525 .
## MEDUC2                   -289.5230    74.2383  -3.900 9.63e-05 ***
## MEDUC3                   -356.4660    69.7739  -5.109 3.25e-07 ***
## MEDUC4                   -498.3213    70.5268  -7.066 1.61e-12 ***
## MEDUC5                   -456.3809    74.4826  -6.127 8.97e-10 ***
## MEDUC6                   -588.3729    71.0434  -8.282  < 2e-16 ***
## MEDUC7                   -645.8416    75.6576  -8.536  < 2e-16 ***
## MEDUC8                   -454.7297    89.1494  -5.101 3.39e-07 ***
## PRECARETRUE              -289.1129    59.4746  -4.861 1.17e-06 ***
## DMAR2                     -13.6937     4.0225  -3.404 0.000664 ***
## PREG_LEN8:PREVIS          -40.1639     2.3891 -16.811  < 2e-16 ***
## PREG_LEN9:PREVIS          -43.8478     2.2226 -19.728  < 2e-16 ***
## PREG_LEN10:PREVIS         -42.9118     2.4235 -17.706  < 2e-16 ***
## PREG_LENLate:PREVIS       -39.4345     4.3790  -9.005  < 2e-16 ***
## PREG_LEN8:MEDUC2          236.8374    80.3986   2.946 0.003222 **
## PREG_LEN9:MEDUC2          242.5954    75.3855   3.218 0.001291 **
## PREG_LEN10:MEDUC2         222.2184    79.9226   2.780 0.005430 **
## PREG_LENLate:MEDUC2       484.5443   132.5339   3.656 0.000256 ***
## PREG_LEN8:MEDUC3          284.9772    75.3659   3.781 0.000156 ***
## PREG_LEN9:MEDUC3          314.2224    70.7317   4.442 8.90e-06 ***
## PREG_LEN10:MEDUC3         315.5045    74.7959   4.218 2.46e-05 ***
## PREG_LENLate:MEDUC3       520.8866   123.2808   4.225 2.39e-05 ***
## PREG_LEN8:MEDUC4          419.1204    76.1332   5.505 3.70e-08 ***
## PREG_LEN9:MEDUC4          459.2697    71.4390   6.429 1.29e-10 ***
## PREG_LEN10:MEDUC4         470.4747    75.5225   6.230 4.69e-10 ***
## PREG_LENLate:MEDUC4       623.6254   124.8982   4.993 5.95e-07 ***
## PREG_LEN8:MEDUC5          337.7226    80.4185   4.200 2.68e-05 ***
## PREG_LEN9:MEDUC5          430.1436    75.4569   5.701 1.20e-08 ***
## PREG_LEN10:MEDUC5         416.7550    79.7422   5.226 1.73e-07 ***
## PREG_LENLate:MEDUC5       583.8511   130.9582   4.458 8.27e-06 ***
## PREG_LEN8:MEDUC6          466.6621    76.5320   6.098 1.08e-09 ***
## PREG_LEN9:MEDUC6          571.3802    71.8651   7.951 1.87e-15 ***
## PREG_LEN10:MEDUC6         572.2985    75.8254   7.548 4.47e-14 ***
## PREG_LENLate:MEDUC6       712.4874   124.7064   5.713 1.11e-08 ***
## PREG_LEN8:MEDUC7          480.9709    81.4108   5.908 3.48e-09 ***
## PREG_LEN9:MEDUC7          626.9303    76.4978   8.195 2.53e-16 ***
## PREG_LEN10:MEDUC7         642.4993    80.5887   7.973 1.57e-15 ***
## PREG_LENLate:MEDUC7       728.1016   131.4678   5.538 3.06e-08 ***
## PREG_LEN8:MEDUC8          289.9933    95.8590   3.025 0.002485 **
## PREG_LEN9:MEDUC8          431.1129    90.1696   4.781 1.75e-06 ***
## PREG_LEN10:MEDUC8         442.0147    94.9694   4.654 3.26e-06 ***
## PREG_LENLate:MEDUC8       517.2541   154.0377   3.358 0.000785 ***
## CIG_0TRUE:PRECARETRUE      89.6994    52.7335   1.701 0.088947 .
## PREG_LEN8:PRECARETRUE     183.8012    70.9970   2.589 0.009631 **
## PREG_LEN9:PRECARETRUE     269.9576    62.6589   4.308 1.65e-05 ***
```

```
## PREG_LEN10:PRECARETRUE    281.9624    78.1815   3.607 0.000310 ***
## PREG_LENLate:PRECARETRUE  733.0657   225.4344   3.252 0.001147 **
## PREG_LEN8:CIG_0TRUE      -183.6907    36.2868  -5.062 4.15e-07 ***
## PREG_LEN9:CIG_0TRUE      -164.8395    33.8371  -4.872 1.11e-06 ***
## PREG_LEN10:CIG_0TRUE     -162.2409    36.3335  -4.465 8.00e-06 ***
## PREG_LENLate:CIG_0TRUE   -217.8188    60.0673  -3.626 0.000288 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 472 on 99896 degrees of freedom
## Multiple R-squared:  0.3269, Adjusted R-squared:  0.3262
## F-statistic:   471 on 103 and 99896 DF,  p-value: < 2.2e-16
```

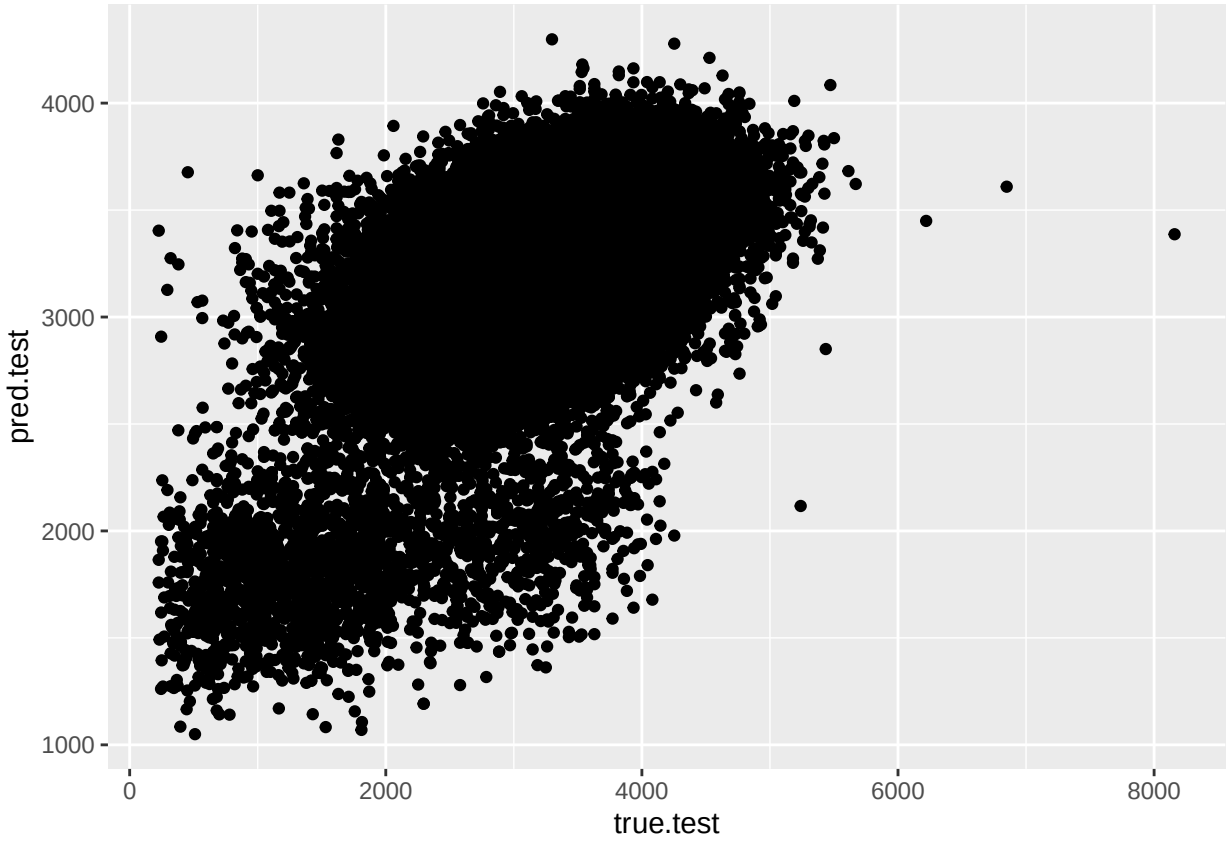
```
# Model Prediction
MSE.train <- mean(final.lm$residuals ^ 2)
pred.test <- predict(final.lm, Test)
MSE.test <- mean((pred.test - Test$DBWT) ^ 2)
MSE.train
```

```
## [1] 222923.6
```

```
MSE.test
```

```
## [1] 223176.5
```

```
true.test <- Test$DBWT
test.pred.df <- data.frame(cbind(true.test, pred.test))
ggplot(test.pred.df, aes(x = true.test, y = pred.test)) +
  geom_point()
```




```
# prediction intervals of five points
five_samples <- Test %>% sample_n(5, replace = TRUE)
predict(final_test.lm, five_samples, interval = "prediction")
```

```
##        fit      lwr      upr
## 1 3305.784 2380.539 4231.029
## 2 3432.877 2507.518 4358.235
## 3 3058.253 2132.846 3983.660
## 4 3494.928 2568.869 4420.986
## 5 3527.827 2602.576 4453.077
```