```
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
```

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

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

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

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

```
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)

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)
```

```
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))

# 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)
```

```r
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))
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)
```

```
# 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))
```

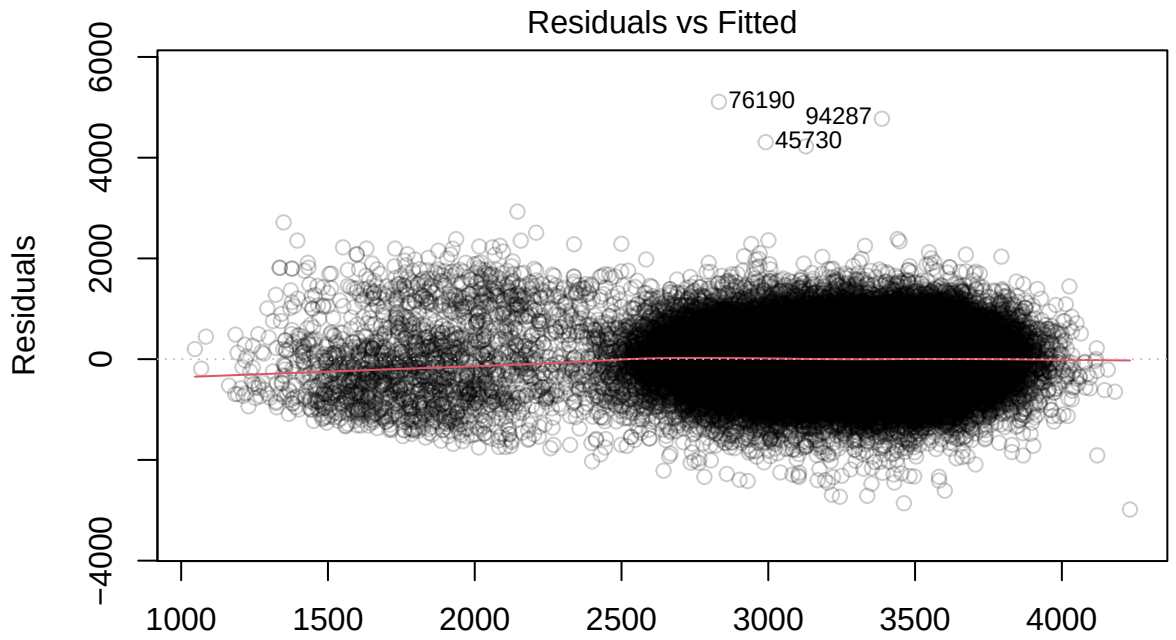
```
# Add interaction terms by F-test
full.lm <- lm(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER +
    PRIORLIVE + CIG_0 + NO_RISKS + RDMETH_REC + PREVIS + ATTEND +
    MBSTATE_REC + FRACE6 + PAY_REC + LD_INDL + FEDUC + NO_MMORB +
    BFACIL + FAGECOMB + NO_INFEC + RESTATUS + MEDUC + PRECARE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

no_BMI_PRECARE <- 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  + 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)

no_WTGAIN_PER_PRECARE <- 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 +  PRECARE * MEDUC +
    PREVIS * PREG_LEN + PREG_LEN * MEDUC + PRECARE * CIG_0 + CIG_0 * SEX +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Train)

no_PRECARE_MEDUC <- 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 +
    PREVIS * PREG_LEN + PREG_LEN * MEDUC + PRECARE * CIG_0 + CIG_0 * SEX +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Train)

no_PREVIS_PREG_LEN <- 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 +
    PREG_LEN * MEDUC + PRECARE * CIG_0 + CIG_0 * SEX +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Train)
```

```
no_PREG_LEN_MEDUC <- 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 + PRECARE * CIG_0 + CIG_0 * SEX +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Train)

no_PRECARE_CIG_0 <- 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 + CIG_0 * SEX +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Train)

no_CIG_0_SEX <- 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 +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Train)

no_PRECARE_PREG_LEN <- 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 +
    CIG_0 * PREG_LEN, data = Train)

no_CIG_PREG_LEN <- 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, data = Train)

# F-test on each interaction term
anova(no_BMI_PRECARE, full.lm) # p-value = 0.3041

anova(no_WTGAIN_PER_PRECARE, full.lm) # p-value = 0.5564

anova(no_PRECARE_MEDUC, full.lm) # p-value = 0.4754

anova(no_PREVIS_PREG_LEN, full.lm) # p-value = 2.2e-16

anova(no_PREG_LEN_MEDUC, full.lm) # p-value = 2.2e-16

anova(no_PRECARE_CIG_0, full.lm) # p-value = 0.09273
```

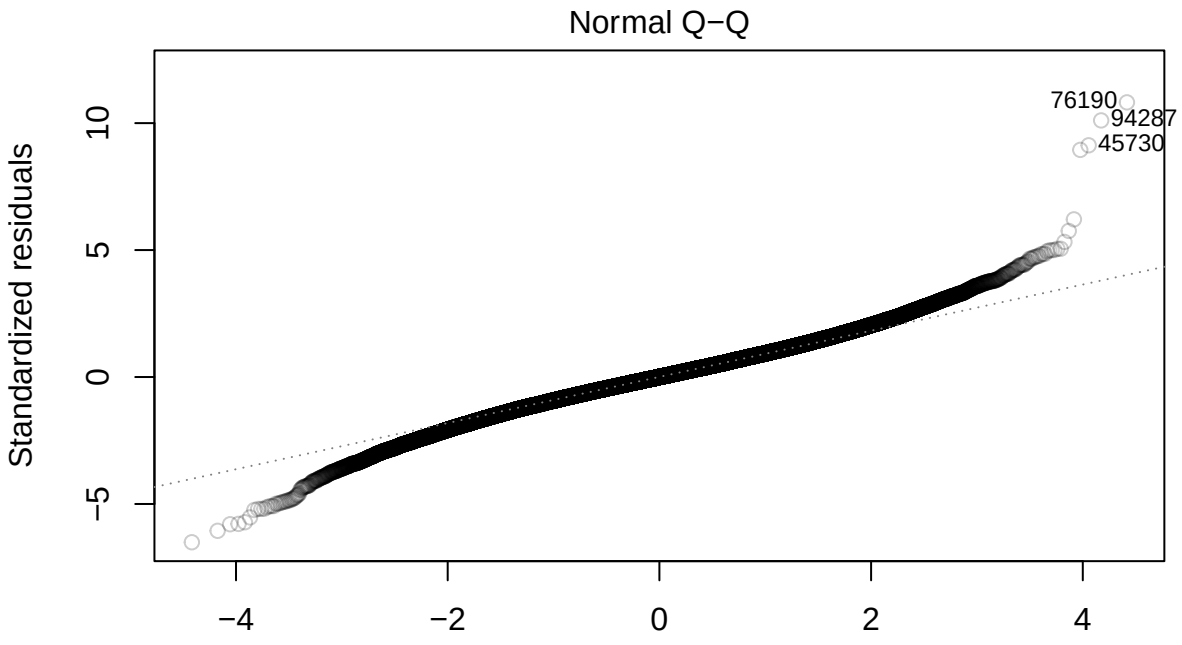
```r
anova(no_CIG_0_SEX, full.lm) # p-value = 0.6727

anova(no_PRECARE_PREG_LEN, full.lm) # p-value = 6.01e-09

anova(no_CIG_PREG_LEN, full.lm) # p-value = 3.828e-05
```

```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 +
    PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Train)
```

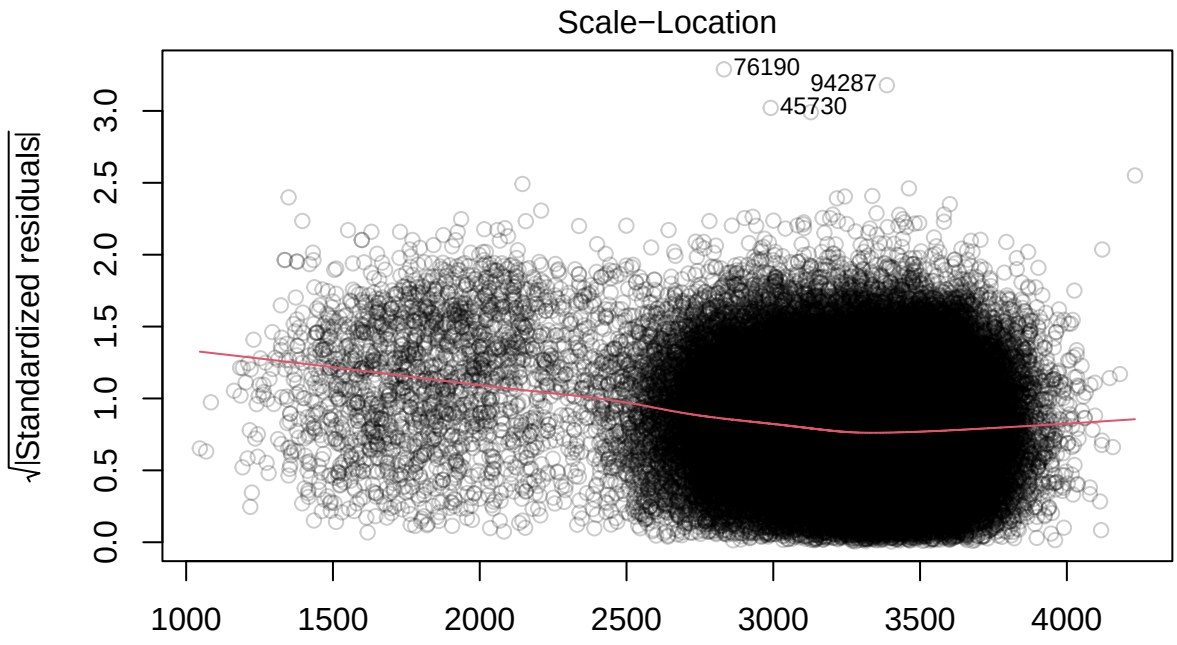
```r
# Model diagnostic
plot(final.lm, col = rgb(red = 0, green = 0, blue = 0, alpha = 0.2))
```



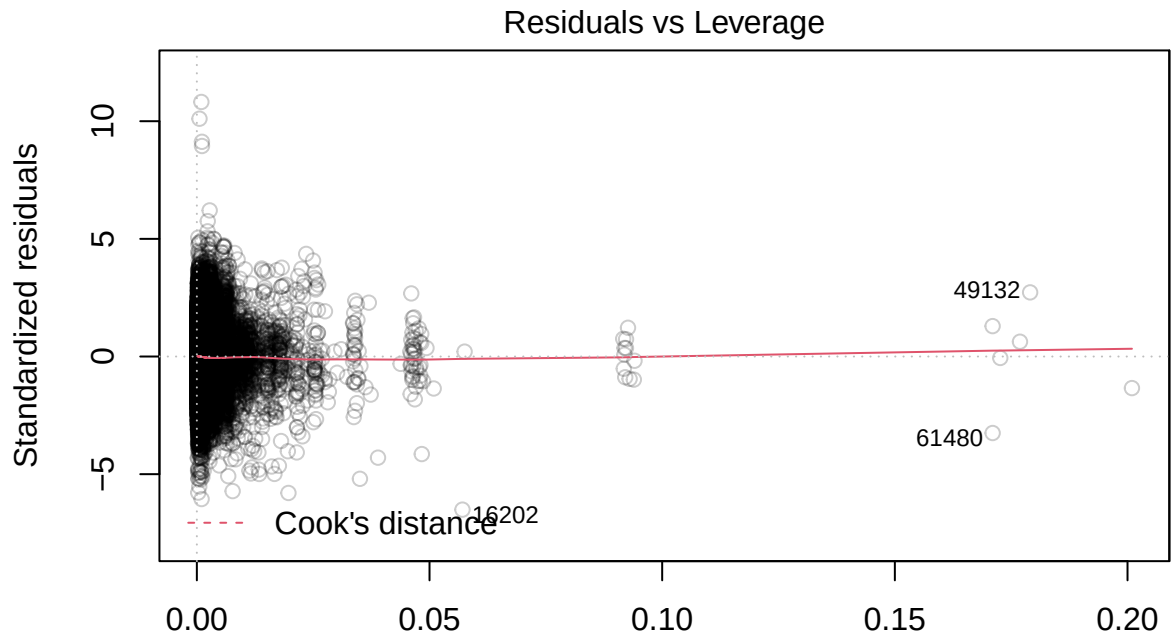
n(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER + PRIORL



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



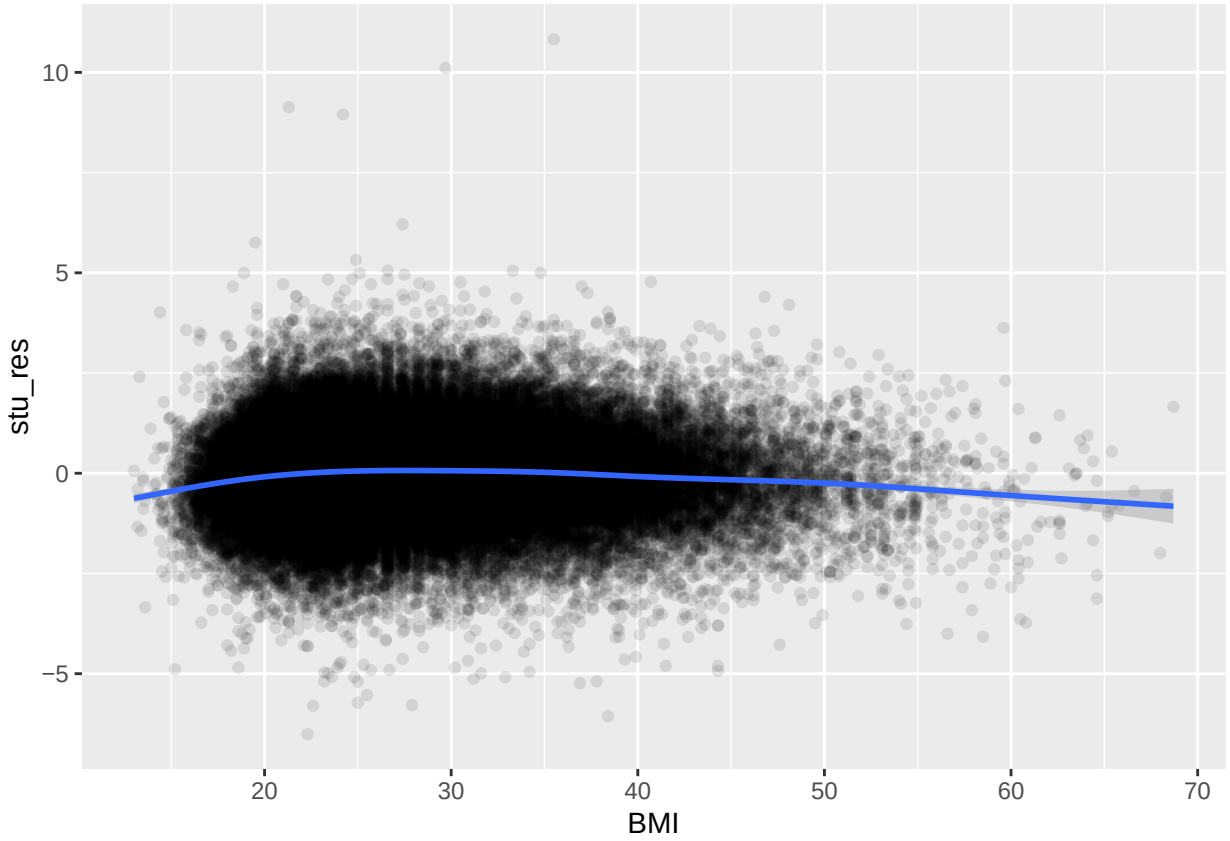
n(DBWT ~ PREG_LEN + M_Ht_In + MRAVE6 + SEX + BMI + WTGAIN_PER + PRIORL



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 10.824628
## 94287   F        9 0.08695652       FALSE 10.111643
## 45730   M        8 0.23333333       FALSE  9.129778
## 34402   M        8 0.51612903        TRUE  8.950188
## 16202   M    Early 0.30508475       FALSE -6.508479
## 82652   M    Early 0.36666667       FALSE  6.210748
```

```
# 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 -6.508479
## 61480   M     Late 0.21678322        TRUE -3.252054
## 49132   M     Late 0.01685393       FALSE  2.726224
```

```
# 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 +
    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 +
##     PRECARE * PREG_LEN + CIG_0 * PREG_LEN, data = Test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3225.4  -286.7     0.0   292.7  4736.6
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -457.6911    96.1446  -4.760 1.93e-06 ***
## PREG_LEN8                 769.8055    97.7727   7.873 3.49e-15 ***
## PREG_LEN9                1082.5793    89.4878  12.098  < 2e-16 ***
## PREG_LEN10               1170.8060   102.7288  11.397  < 2e-16 ***
## PREG_LENLate              405.4862   241.6683   1.678 0.093377 .
## M_Ht_In                    30.1909     0.5617  53.753  < 2e-16 ***
## MRAVE62                  -115.5823     7.8107 -14.798  < 2e-16 ***
## MRAVE63                    57.8509    19.3231   2.994 0.002755 **
## MRAVE64                   -14.9598    10.9813  -1.362 0.173108
## MRAVE65                   -57.5668    37.2215  -1.547 0.121962
## MRAVE66                   -40.1172    10.1873  -3.938 8.22e-05 ***
## SEXM                      116.7919     2.9905  39.054  < 2e-16 ***
## BMI                        17.6708     0.2887  61.198  < 2e-16 ***
## WTGAIN_PER                886.4036    16.2828  54.438  < 2e-16 ***
## PRIORLIVETRUE             101.8609     3.5075  29.041  < 2e-16 ***
## CIG_0TRUE                  50.0692    33.0966   1.513 0.130329
## NO_RISKS1                 135.7533     4.3023  31.554  < 2e-16 ***
## RDMETH_REC2               115.1989    11.2566  10.234  < 2e-16 ***
## RDMETH_REC3               -39.3807     4.1728  -9.437  < 2e-16 ***
## RDMETH_REC4               126.5436     6.1584  20.548  < 2e-16 ***
## PREVIS                     49.2232     2.1659  22.726  < 2e-16 ***
## ATTEND2                     1.0756     5.4209   0.198 0.842715
## ATTEND3                    50.8642     5.4605   9.315  < 2e-16 ***
## ATTEND4                    70.2404    20.5073   3.425 0.000615 ***
## ATTEND5                    43.7746    19.1012   2.292 0.021924 *
## MBSTATE_REC2               53.2305     4.5899  11.597  < 2e-16 ***
## FRACE62                   -57.7496     7.3992  -7.805 6.01e-15 ***
## FRACE63                    29.1277    19.9379   1.461 0.144040
## FRACE64                  -129.7232    11.1710 -11.612  < 2e-16 ***
## FRACE65                    38.6713    35.7298   1.082 0.279110
## FRACE66                   -33.2670    10.1685  -3.272 0.001070 **
## PAY_REC2                   17.9945     4.0698   4.421 9.82e-06 ***
## PAY_REC3                   34.6460     8.9455   3.873 0.000108 ***
## PAY_REC4                   21.5551     8.3916   2.569 0.010211 *
## LD_INDLY                   35.2240     3.4799  10.122  < 2e-16 ***
## FEDUC2                    -37.7180    12.3376  -3.057 0.002235 **
## FEDUC3                    -17.0130    11.8354  -1.437 0.150587
## FEDUC4                     -3.9605    12.1875  -0.325 0.745207
## FEDUC5                     10.3839    12.9451   0.802 0.422471
## FEDUC6                     18.3131    12.4403   1.472 0.141004
## FEDUC7                     16.4220    13.2591   1.239 0.215518
## FEDUC8                     17.4785    14.7608   1.184 0.236368
## NO_MMORB1                 -83.8203    12.4537  -6.731 1.70e-11 ***
## BFACIL2                    60.0529    19.4559   3.087 0.002025 **
## BFACIL3                    98.9730    20.1654   4.908 9.21e-07 ***
```

```
## BFACIL4                  -124.4845    50.5516  -2.463 0.013798 *
## BFACIL5                     9.0099   136.8784   0.066 0.947518
## BFACIL6                    51.8987   101.2071   0.513 0.608095
## BFACIL7                    26.7593    57.1501   0.468 0.639622
## FAGECOMB                   -0.7571     0.2595  -2.918 0.003527 **
## NO_INFEC1                   7.7741    10.6255   0.732 0.464387
## RESTATUS2                 -14.9637     3.3040  -4.529 5.93e-06 ***
## RESTATUS3                 -20.9266     9.1247  -2.293 0.021827 *
## RESTATUS4                 -54.8471    30.4543  -1.801 0.071712 .
## MEDUC2                   -289.2116    74.2387  -3.896 9.80e-05 ***
## MEDUC3                   -356.4013    69.7745  -5.108 3.26e-07 ***
## MEDUC4                   -498.4400    70.5275  -7.067 1.59e-12 ***
## MEDUC5                   -456.3014    74.4833  -6.126 9.03e-10 ***
## MEDUC6                   -588.3851    71.0441  -8.282  < 2e-16 ***
## MEDUC7                   -646.0644    75.6582  -8.539  < 2e-16 ***
## MEDUC8                   -455.0322    89.1500  -5.104 3.33e-07 ***
## PRECARETRUE              -278.1423    59.1244  -4.704 2.55e-06 ***
## DMAR2                     -13.6844     4.0225  -3.402 0.000669 ***
## PREG_LEN8:PREVIS          -40.1604     2.3891 -16.810  < 2e-16 ***
## PREG_LEN9:PREVIS          -43.8423     2.2226 -19.726  < 2e-16 ***
## PREG_LEN10:PREVIS         -42.9064     2.4236 -17.704  < 2e-16 ***
## PREG_LENLate:PREVIS       -39.4310     4.3791  -9.004  < 2e-16 ***
## PREG_LEN8:MEDUC2          236.3338    80.3988   2.940 0.003288 **
## PREG_LEN9:MEDUC2          242.0825    75.3856   3.211 0.001322 **
## PREG_LEN10:MEDUC2         221.8355    79.9230   2.776 0.005511 **
## PREG_LENLate:MEDUC2       483.9039   132.5346   3.651 0.000261 ***
## PREG_LEN8:MEDUC3          284.8119    75.3665   3.779 0.000158 ***
## PREG_LEN9:MEDUC3          314.0149    70.7322   4.439 9.03e-06 ***
## PREG_LEN10:MEDUC3         315.3976    74.7966   4.217 2.48e-05 ***
## PREG_LENLate:MEDUC3       519.9849   123.2808   4.218 2.47e-05 ***
## PREG_LEN8:MEDUC4          419.0912    76.1339   5.505 3.71e-08 ***
## PREG_LEN9:MEDUC4          459.2510    71.4397   6.429 1.29e-10 ***
## PREG_LEN10:MEDUC4         470.6331    75.5232   6.232 4.63e-10 ***
## PREG_LENLate:MEDUC4       623.3764   124.8993   4.991 6.02e-07 ***
## PREG_LEN8:MEDUC5          337.4939    80.4192   4.197 2.71e-05 ***
## PREG_LEN9:MEDUC5          429.8917    75.4575   5.697 1.22e-08 ***
## PREG_LEN10:MEDUC5         416.6346    79.7429   5.225 1.75e-07 ***
## PREG_LENLate:MEDUC5       583.1616   130.9589   4.453 8.48e-06 ***
## PREG_LEN8:MEDUC6          466.4981    76.5327   6.095 1.10e-09 ***
## PREG_LEN9:MEDUC6          571.2319    71.8657   7.949 1.91e-15 ***
## PREG_LEN10:MEDUC6         572.2926    75.8261   7.547 4.48e-14 ***
## PREG_LENLate:MEDUC6       711.8414   124.7070   5.708 1.15e-08 ***
## PREG_LEN8:MEDUC7          481.0204    81.4116   5.909 3.46e-09 ***
## PREG_LEN9:MEDUC7          626.9834    76.4986   8.196 2.51e-16 ***
## PREG_LEN10:MEDUC7         642.6966    80.5894   7.975 1.54e-15 ***
## PREG_LENLate:MEDUC7       727.6808   131.4688   5.535 3.12e-08 ***
## PREG_LEN8:MEDUC8          290.1406    95.8599   3.027 0.002473 **
## PREG_LEN9:MEDUC8          431.2399    90.1704   4.782 1.73e-06 ***
## PREG_LEN10:MEDUC8         442.2842    94.9702   4.657 3.21e-06 ***
## PREG_LENLate:MEDUC8       516.8991   154.0390   3.356 0.000792 ***
## PREG_LEN8:PRECARETRUE     185.1995    70.9929   2.609 0.009090 **
## PREG_LEN9:PRECARETRUE     267.0036    62.6354   4.263 2.02e-05 ***
## PREG_LEN10:PRECARETRUE    278.7855    78.1599   3.567 0.000361 ***
## PREG_LENLate:PRECARETRUE  740.2093   225.3974   3.284 0.001024 **
```

```
## PREG_LEN8:CIG_0TRUE      -181.9591    36.2728  -5.016 5.27e-07 ***
## PREG_LEN9:CIG_0TRUE      -162.3534    33.8059  -4.803 1.57e-06 ***
## PREG_LEN10:CIG_0TRUE     -159.5006    36.2981  -4.394 1.11e-05 ***
## PREG_LENLate:CIG_0TRUE   -215.2208    60.0484  -3.584 0.000338 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 472 on 99897 degrees of freedom
## Multiple R-squared:  0.3269, Adjusted R-squared:  0.3262
## F-statistic: 475.6 on 102 and 99897 DF,  p-value: < 2.2e-16
```

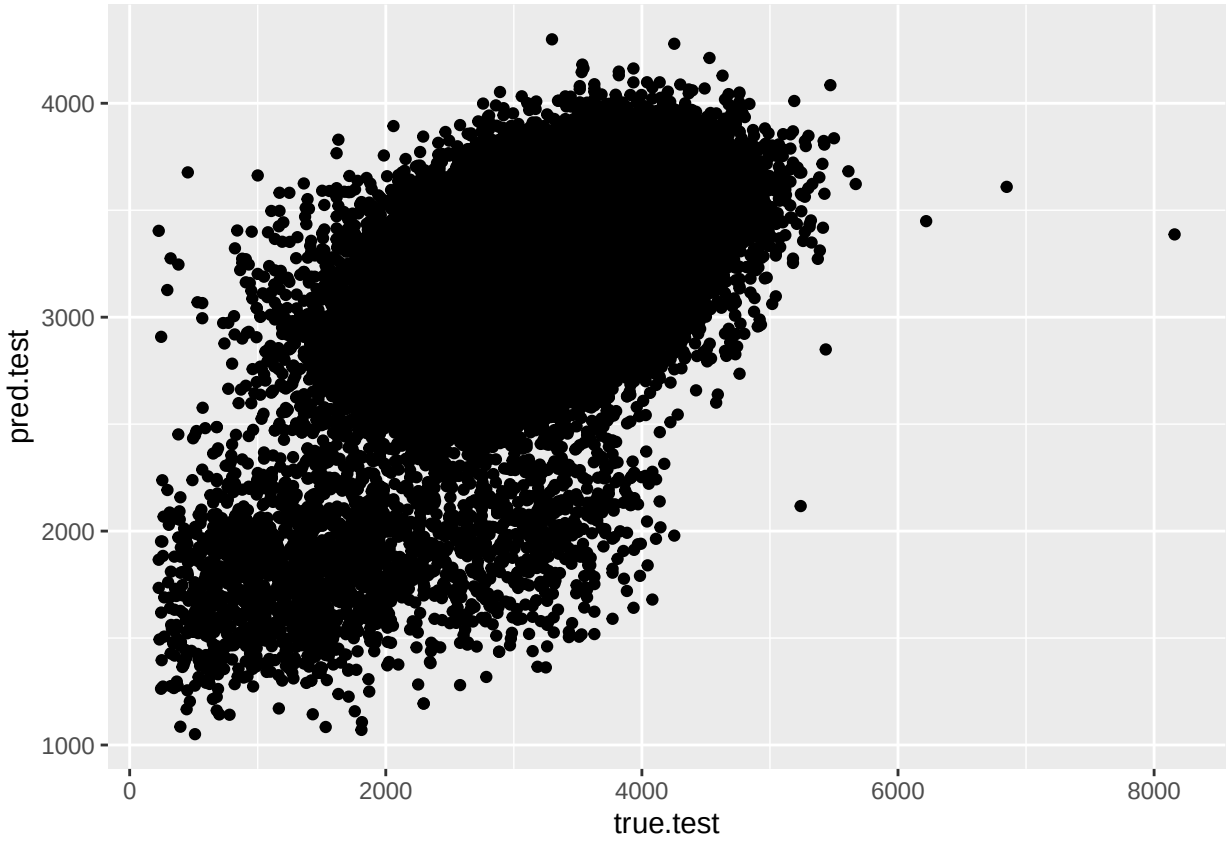
```
# Model Prediction
MSE.train <- mean(final.lm$residuals ^ 2)
pred.test <- predict(final.lm, Test)
MSE.test <- mean((pred.test - Test$DBWT) ^ 2)
MSE.train
```

```
## [1] 222936.8
```

```
MSE.test
```

```
## [1] 223183.1
```

```
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 3305.875 2380.621 4231.130
## 2 3432.951 2507.584 4358.318
## 3 3058.351 2132.935 3983.767
## 4 3494.885 2568.818 4420.952
## 5 3527.893 2602.634 4453.152
```