STAT151A Final Project, Fall 2021

Wenhao Pan, Rachel Chen, Richard Shuai

December 17, 2021

Contents

1 Introduction

Baby's mass is correlated with mortality risk and potential future developmental problems. For example, researchers in Denmark found that babies with birth weights of less than 5 pounds are more likely to experience health complications and even a lower intelligence quotient as children. Thus, it makes sense for healthcare workers and parents to want to predict a baby's weight based on current information. Intuitively speaking, a baby's mass could be predicted by a lot of factors such as the health of the parents, the sex of the baby, the mother's pregnancy records, etc. In this project, we aim to use linear models to answer the following two questions regarding the baby's weight: (1) How does intervening a pregnant woman's living habits or behaviors affect her baby's birth weight in the future? (2) Given the information about an expecting family, what is our best prediction of their baby's weight?

The first question is more related to causal inference, and its answer could help doctors give suggestions to a pregnant woman for delivering a normally weighted baby. The second one is more related to prediction, and its answer could help doctors conjecture a baby's weight right before delivery.

2 Data Description

This dataset was taken from the National Center for Health Statistics, and contains information about 3.8 million childbirths in the US in 2018. There are 55 columns, so we grouped them into the following categories:

- Delivery situation ex) place of birth, number of people around, birth time
- The baby's health information ex) period of gestation, birth weight
- Parents information ex) marital status, education, race
- Parents health records ex) smoking history, age
- Mother's pregnancy records ex) number of prenatal visits, prior births

The User Guide on the website contains the detailed explanations of each column. We will use the baby birth weight column (`DBWT`) as the response variable, and all other variables will be used as explanatory variables.

3 Data Preprocessing

We first propose that due to the excessive size of the original dataset, 3.8 million observations, we plan to randomly subsample three subsets, one with 5000 observations and two with 100000 observations, with replacement as datasets for EDA, training, and testing. This plan balances the computational cost of the analysis and the complexity of our dataset well. We conduct this subsampling plan at the end of data preprocessing.

The priority of our data preprocessing is to reduce the dimensionality of our dataset by filtering out unuseful features. We have 54 explanatory variables, but it is not efficient to analyze each of them evenly. We suspect that some variables can be combined and condensed into a new variable. To systematically filter the features, we split the explanatory variables into five exclusive categories: Homogeneous: Variables of which more than 99% of entries have the same values. Minor: We do not consider interaction terms involving these variables. Major: We do consider interaction terms involving these variables. Obsolete: After we create a new variable based on these variables, they essentially do not provide enough extra information to be kept. The new variable belongs to "Major". Redundant: After we select one from a group of variables including similar information, the rest become redundant or unnecessary to be kept. The selected variable belongs to "Major". See the appendix for the names of the variables in each category.

Next, we drop all the observations including any missing value in the columns of "Minor", "Major", and "Obsolete" categories. After dropping, we still have about 2.8 million observations left, which are sufficient for subsampling. The missing values in our dataset are not left blank or NA. Instead, they are recoded into

values such as '9' or '999', depending on the variable. Thus, we manually look up the recodings from the User Guide and drop the missing values for each feature. Imputing those missing values might be a better approach since if the missing values exemplify a systematic pattern, we might introduce bias by removing all the missing values. However, given the already complicated structure of our dataset, we choose the simpler approach-dropping missing values, noting that the imputation approach is still valuable to be explored.

Cleaning up missing values allows us to conduct feature engineering, which aims to use domain knowledge to simplify the dataset while maintaining the original information. For example, we create the feature `PREG_LEN` which estimates the pregnancy length by computing the number of months between the last normal mense and delivery date. Then, we categorize `PREG_LEN` into `Early`, `8`, `9`,`10`,`Late` by common sense. Thus, `PREG_LEN` becomes a "Major" variable, and variables about the last normal menses and delivery dates become "Obsolete" variables. `FRACE6`, `FRACE15`, `FRACE31`, and `FHISPX` all describe a father's race but with different granularity levels. We select `FRACE6` to solely describe a father's race since we think six racial categories already sufficiently differentiate people. Thus, `FRACE6` is a "Major" variable, and other father's race variables become "Redundant". See the code appendix for the complete work of feature engineering.

Finally, we drop "Homogeneous", "Obsolete", and "Redundant" variables. It is obvious that we should drop "Obsolete" and "Redundant" variables to mitigate collinearity and duplication in our dataset. We drop the "Homogeneous" variables because it is highly likely that they essentially have one unique value or a negligible number of other values in subsample datasets. We warn that the entire preprocessing approach is considerably subjective, and the following regression analysis is highly dependent on our approach. It may sound like a compromise to the complexity of our dataset to some audiences. We encourage different preprocessing approaches, but we continue with ours since we think it is still reasonable.

4 Exploratory Data Analysis

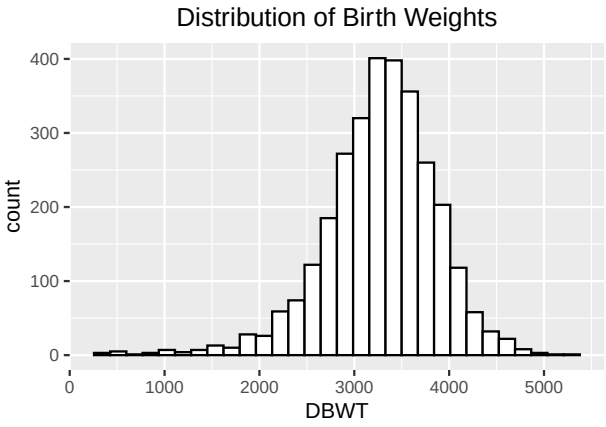


Figure 1a: Distribution of birth weights

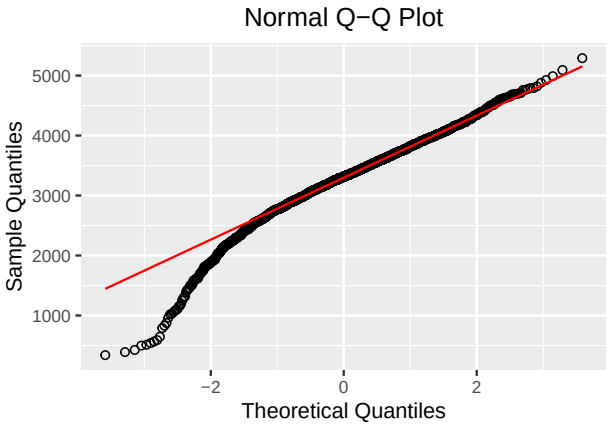


Figure 1b: Normal Q−Q plot for birth weights

To gain a better understanding of the variables and data, we performed EDA on our dataset. First, we plotted the distribution of the response variable to verify that the normality assumption in linear regression is satisfied. From Figures 1a and 1b, based on the histogram and Q-Q plot, we saw that the distribution has a heavy left tail but is otherwise normal-looking. To verify symmetry of `DBWT`, we used the formula $\frac{\text{Upper quartile - Median}}{\text{Median - Lower Quartile}}$, which yields 0.8986. Since the ratio is close to 1, we concluded that our response variable is sufficiently symmetric. Although we could have used a Box-Cox transformation to alleviate the left skew, we chose not to transform `DBWT` for ease of interpretation in downstream analysis.

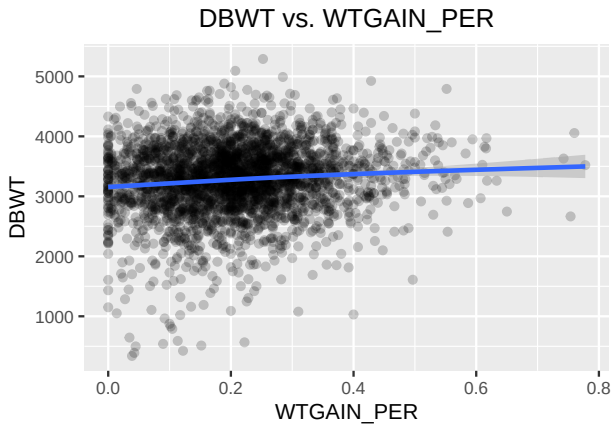


Figure 2a: Birth weight vs. percentage of weight gained

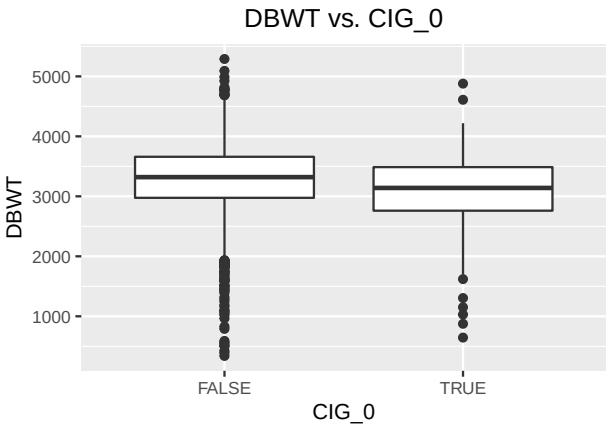


Figure 2b: Bbirth weight vs. mother's smoking status

We next examined bivariate relationships between our response variable and each explanatory variable in our dataset. In Figure 2a, we saw that as the percentage of weight gained due to pregnancy increases, the birth weight tends to increase. We also observed that `CIG_0` may also be an important explanatory variable to include in our model, since from the box plot in Figure 2b, we saw that the 1st quartile, median, and 3rd quartile of birth weight are all lower if the mother smokes, than if she does not. In Figure 3a, when plotting birth weight against the prenatal care status of the mother, we found that the distributions of birth weight seemed to be most different between mothers who didn't receive prenatal care and mothers who did. This suggested that the most important difference would be observed when we binarize the explanatory variable for the mother's prenatal care status. From Figure 3b, we saw that this holds, and we therefore binarized the mother's prenatal care status for downstream analysis.

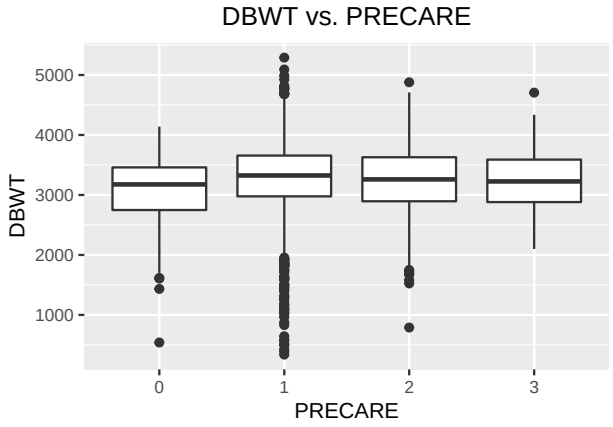


Figure 3a: Birth weight vs. mother's prenatal care status

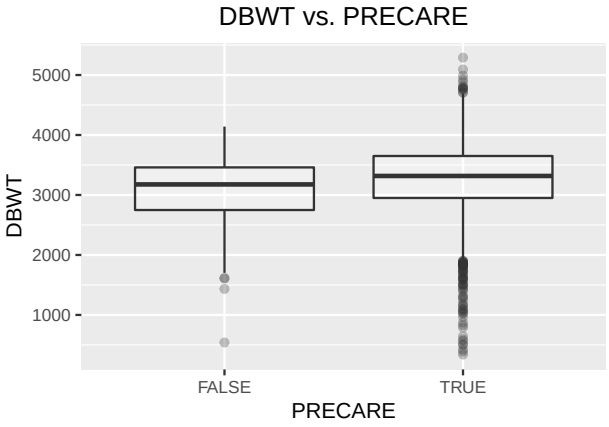


Figure 3b: Birth weight vs. binarized prenatal care status

We also visualized interactions explanatory variables using box plots and scatter plots. The box plot in Figure 4a shows the interaction between `SEX` and `PRECARE`. We noticed that the difference in the median birth weight for mothers who received prenatal care and who did not changes based on the sex of the baby, which motivates using an interaction term. More specifically, the difference is larger in male babies than female babies.

In Figure 4b, we observed a possible interaction between the mother's prenatal care status and her BMI when predicting the baby's birth weight. We saw that if a pregnant woman does not receive prenatal care, as her BMI increases, she is more likely to have lighter babies, which could be a signal for an unhealthy baby. This might make sense because obesity is linked to health problems, which consequently affect the health and birth weight of the baby. Therefore, the interaction may be a result of mothers receiving proper

care and taking actions to mitigate their health problems, leading to more normal birth weights. However, because of the few data points for the case where the mother didn't go to prenatal care, we saw a high error in the slope, meaning that the interaction seen here may not be statistically significant.

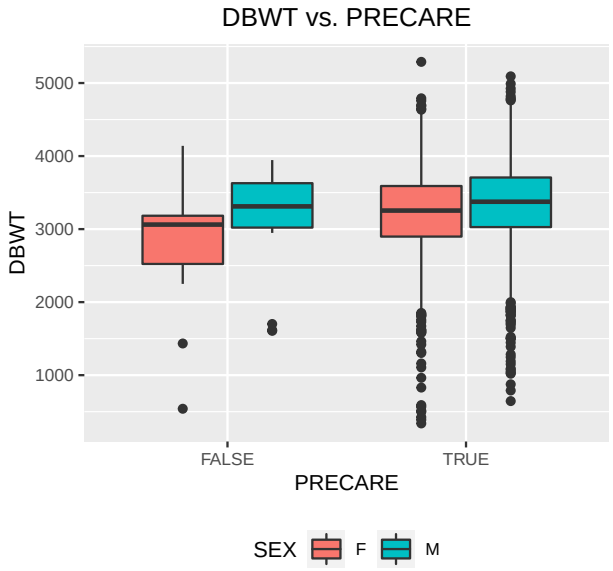


Figure 4b: Interaction between sex and prenatal care status

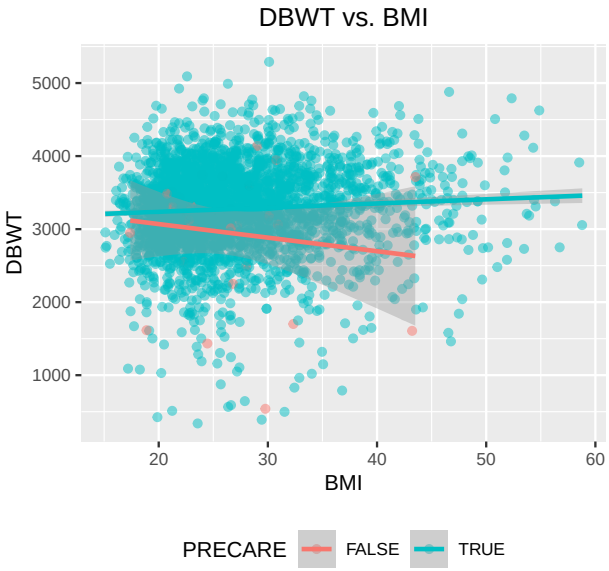


Figure 4b: Interaction between prenatal care status and BMI

5 Model Selection

We first fit the fullest model with only the main effect terms, which contains 30 explanatory variables or 68 regressors. Such a complicated model with so many regressors is not desired for prediction or causal inference because it will tend to overfit on the training data and therefore generalize poorly on new datasets, even if drawn from the same distribution. Also, such a model would require us to collect lots of information to predict, which will limit the usability of the model in a realistic setting. Moreover, such a model is hard to interpret for causal inference. Thus, we conduct model selection to select a simpler model.

We first removed explanatory variables introducing singularity, which have fitted coefficients `NA`. Next, we constructed four models using four different approaches: forward/backward selection with AIC/BIC by `step` function and then select the one with the lowest leave-one-out cross-validation (LOOCV) error. Both AIC and BIC measure the in-sample fitness of a model, while BIC penalizes the model size more when the sample size is large. Lower AIC or BIC in value means a better model. We chose forward and backward selection instead of all subset selection because of the large number of explanatory variables, which makes all subset selection computationally infeasible. We chose the `step` function because it adds or drops categorical variables only as an entire unit instead of splitting them up into unconnected dummy regressors. Ideally, we hope that these four models using different criteria and search strategies would explore diverse model choices, so the final one selected by LOOCV error is the most descriptive.

It turns out that both the models returned by forward and backward selection with AIC have the lowest LOOCV error and are identical in terms of the set of included explanatory variables, so both of them are the best model. We then added the selected interaction terms based on our findings from the EDA to the best model and use the incremental F-test with `Anova` to filter out the insignificant interaction terms. See the code appendix for more details about the entire process. Our final model, to be used for both causal inference and prediction, contains 30 explanatory variables or 103 regressors:

```
## 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
```

6 Model Diagnostics

Although we require a statistically significant linear model from model selection, the linear model made strong and specific assumptions about the structure of our data (Fox, p266). These assumptions-linearity, constant variance, independent noise, and normality-do not often hold in applications. Moreover, the method of least squares can be very sensitive to unusual or influential data points (Fox, p266). Thus, to examine the credibility and validity of our model, we used a series of model diagnostics techniques to check the model assumptions and identify unusual or influential data points.

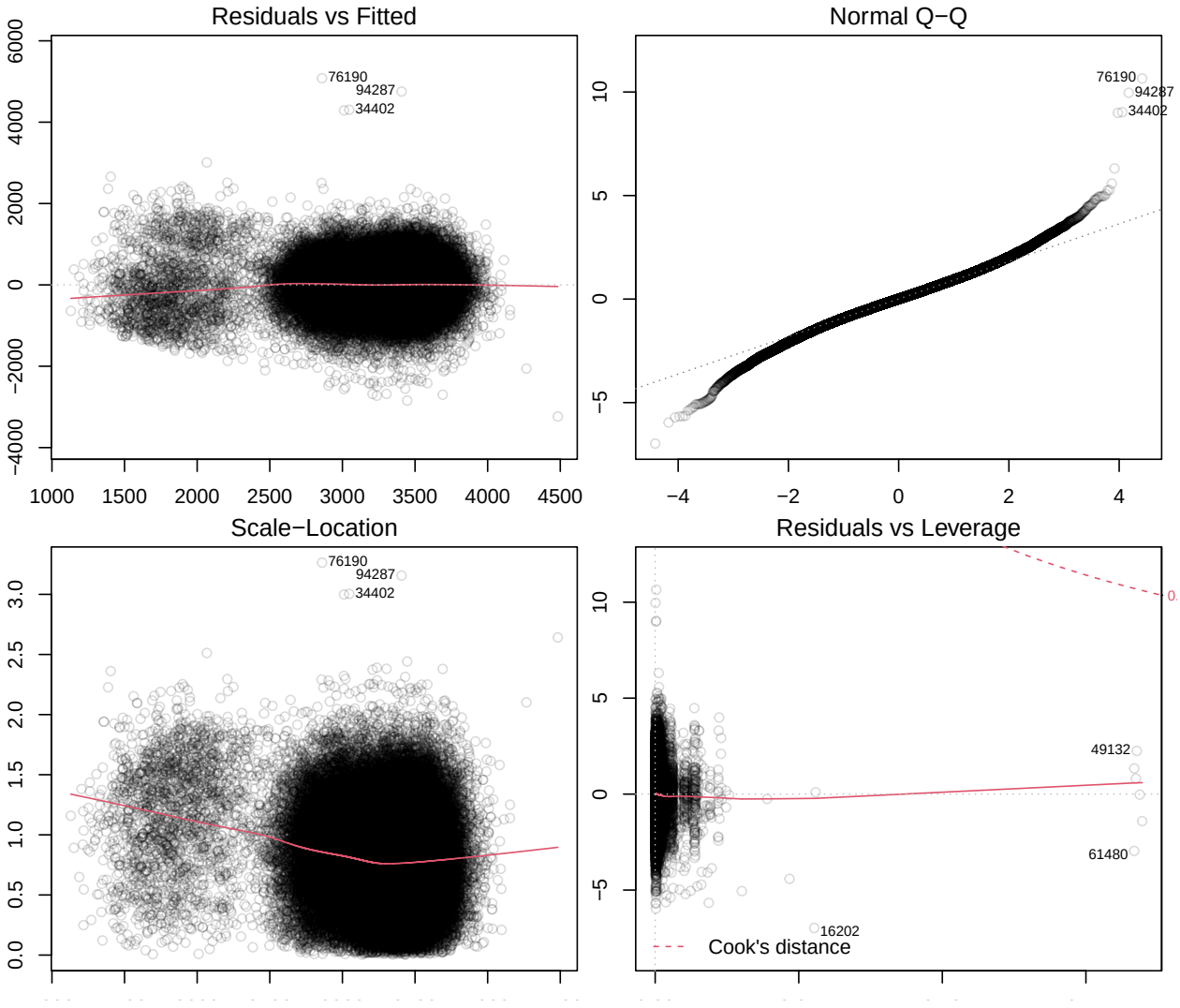


Figure 5: Model diagnostics plot

6.1 Linear Modeling Assumptions

First, we verified our modeling assumptions using various diagnostic plots. We skipped the independent noise assumption because we are not dealing with geospatial and time series data, so we could safely assume that noises are independent of each other.

When plotting the residuals against the fitted values (Figure 5), although the heavier cluster seems to have smaller studentized residuals, the difference is very slight, so we said that we do not see a clear trend in the spread of residuals as a function of fitted values, which supports our constant variance assumption. Additionally, because the residuals do not show any clear non-linear pattern, the plot supports our linearity assumption. To help us verify normality assumptions, we also plotted a quantile-comparison plot of the standardized residuals against the normal distribution (Figure 5). Examining the shape of the Q-Q plot, we see that the distribution of the residuals has slightly heavy tails, indicating a potential violation of this assumption. Although we can use case bootstrapping to alleviate this issue, we chose to continue with our original data since the issue is not severe. In the scale-location plot, the red line is roughly horizontal, providing additional evidence for the validity of the constant variance assumption (Figure 5).

Finally, we plot the studentized residuals versus one of the explanatory variables, BMI, and look for any patterns (Figure 6a). The studentized residuals appear to be mostly centered around 0 with no clear pattern, which supports our linearity assumption. However, we notice some downward curvature towards the extreme values of BMI, indicating a possible slight violation of our linearity assumption. Additionally, the spread of the studentized residuals does not seem to have a strong dependence on BMI, thus demonstrating homoscedasticity and indicating support for our constant variance assumption.

6.2 Unusual, Influential Data Points

Next, we detected and analyzed unusual data points that may significantly affect the fitted coefficients of our model. In our diagnostic plots, we observed a few unusual data points, possibly outliers, with indices 76190, 94287, and 45730 in Figure 5. Sample `76190` has the largest studentized residual `10.82` in magnitude. Its p-value is much smaller than `0.05` with Bonferroni correction. This observation seems to imply that our model fails to capture some important characteristics of the data (Fox, p267), but we found that the birth weight `DBWT` of this sample is `7940` which is extremely rare in reality. Thus, we claim that this outlier is due to an unpredictable event instead of the model defect.

To identify the influential data points, we checked if there is any point outside the contour of the Cook's distance equal to 0.5 in the residuals vs. leverage plot. A larger Cook's distance means a larger influence of a data point on the coefficient estimation. As we cannot even see the contour in the plot, we claim that no data point is highly influential.

It is worth noting that all the diagnostic plots suffer from overplotting due to large training data size. Looking at Figure 5, there might be underlying patterns in the large cluster on the right (from fitted values = 2500 to 4000) that we cannot see. Thus, we might want to zoom in on different parts of the plot for future study.

7 Model Interpretation

7.1 Causal Inference

To avoid the issue of post-selection inference, we re-fit the model to the test set. We used the coefficients of this new model to answer our causal inference question. Because we aim to change a pregnant woman's behavior to control her baby's weight, we do not need to interpret the coefficients of variables that are almost impossible to intervene, such as race, education, and age.

Technically, each fitted coefficient can be interpreted as "the average difference in `DBWT` associated with one-unit change in the variable if we can hold all other variables constant." For example, the coefficient of `BMI`

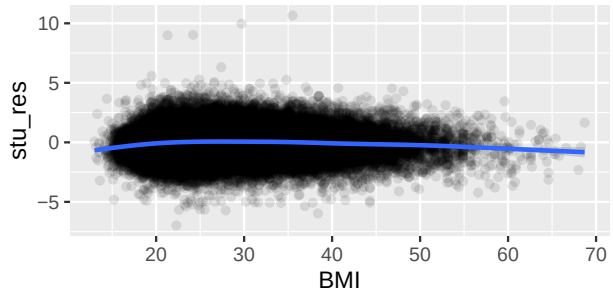
is `17.67`, so we would expect that on average, if we can hold all other regressors constant, a unit increase in a mother's pre-pregnancy BMI will be associated with an increase of baby birth weight by 17.67 grams. In this way, we can understand the effect of the intervention on each variable quantitatively. However, the interpretation becomes complicated when interactions between categorical variables exist, because changing the main effect regressor will change the interaction regressor simultaneously. Thus, depending on the regressor, we may have to consider multiple coefficients at the same time, although we can make analogous statements to "average difference associated with change" at the beginning.

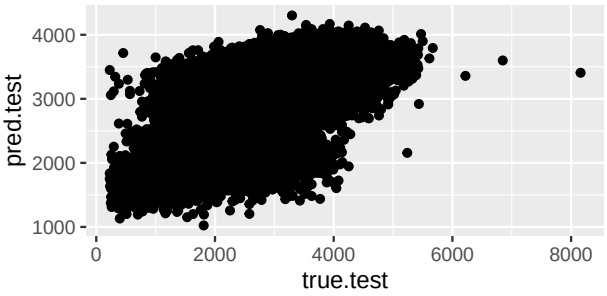
Therefore, if we naively use our model for causal inference, if we predict that a mother will deliver a baby with a dangerously low weight, we could suggest that the mother follows a healthy diet plan to increase her BMI. According to the model coefficients, increasing the mother's BMI should increase the weight of the baby. Similarly, the negative coefficient on `CIG_0TRUE` and all interaction terms that include `CIG_0TRUE` would lead us to claim that a mother should not smoke to ensure that her baby will be delivered with a healthy weight.

To prioritize intervention strategies based on our model, we could interpret the relative significance of explanatory variables. For all intervenable numerical variables, we could compute standardized coefficients and compare the corresponding magnitudes. For categorical variables or interaction terms, we could rely on the significance of the coefficients as determined by incremental F-tests.

However, the possibility of confounding variables can undermine the credibility of our causal inference outcome. For example, during EDA, we saw that mothers that underwent prenatal care delivered higher weight babies. However, based on the coefficients of the fitted model, we observed that undergoing prenatal care is associated with *lower* delivered baby weights for all mothers except those with estimated pregnancy lengths longer than 10 months (making up only 0.975% of the mothers in the test dataset). This indicates that the underlying true relationship between an explanatory variable and response variable may be completely different from the one explained by our model due to the existence of confounding variables. We will discuss more about confounding variables in the Discussion section. Therefore, because we have not properly controlled for confounding variables in our dataset, we cannot reliably use our model for causal inference. A more careful experimental design for collecting our data would be necessary for eliminating confounding variables. For example, if we want to further explore causal inference, we might want to have a control group.

7.2 Prediction






```
##        fit      lwr      upr
## 1 2571.102 1636.744 3505.461
## 2 2774.201 1840.244 3708.159
## 3 3717.029 2783.186 4650.871
```

The final model has an adjusted R^2 of 0.3262 on the training set, which is decent given the complexity of our dataset and questions in a social study research. Our model achieves a mean squared error (MSE) of 223183.1 on the test set, compared with a MSE of 222923.6 on the training set. The relatively small

difference between the train and test MSEs indicate that our model is not overfitting to the training set, implying that the model generalizes fairly well to the test set.

To further evaluate whether the model will predict future baby birth weights with high precision, we also examined the 95% prediction intervals of three randomly selected data points from the test set. Looking at the distribution of baby birth weights in Figure 1a, we see that the prediction intervals tend to be very wide relative to the entire distribution of values of `DBWT`, indicating that the model's predictions are imprecise. Therefore, the model will likely predict future baby birth weights with low precision.

We plotted the predicted v.s. actual values in Figure 6b. Clearly, we can see two clusters with very different centers. It may imply that our data is a mixture of samples from different populations. We will discuss this observation in detail in the next section.

8 Discussion

The primary purpose of this project was to determine whether we can predict the birth weight of a baby given information about the expecting family, and which factors we can intervene with to change the baby's birth weight. Therefore, we must consider the extent to which our linear model can be used for prediction and for causal inference.

As shown above, although our final model does not show signs of overfitting, the prediction intervals on the test set indicate that the model's predictions of birth weight for new data points are imprecise. This indicates that the model is unsuited for reliable prediction for new data points. Additionally, because the dataset used in this report is specific to US births, it is uncertain how well the model will generalize when predicting the birth weights globally. Furthermore, because the test dataset uses only births in 2018, the model's performance has not been evaluated for predicting birth weights for babies born in the current year.

For causal inference, based on our model coefficients, we could suggest the mother to increase her weight since the coefficient for `BMI` and `WTGAIN_PER` are positive. We could also suggest the mother to increase the number of prenatal visits since the coefficient for `PREVIS` is also positive. However, it is difficult to draw definitive applications for causal inference, because, according to a study from the University of Exeter, a baby's weight is mostly determined by his or her genetic code. This suggests that the explanatory variables in our dataset act primarily as proxies for the true cause of a baby's birth weight. Intervening with the explanatory variables in this dataset therefore does not guarantee that the baby's birth weight will change. Thus, we suggest caution when attempting to use this model for causal inference.

As we discovered at the end of the last section, our dataset seems to contain different groups of samples from different populations. Thus, it might be wiser to fit an individual linear model for each group rather than a single model for all the groups together. To identify these groups, we could use the domain knowledge to manually classify our data or clustering algorithms, such as K-means and Gaussian Mixture, to automatically classify our data.

9 Conclusion

In this report, we explored data about child births in the United States in 2018 from the National Center for Health Statistics. Given a set of explanatory variables describing information about the baby's parents, such as education status, BMI, and smoking history, we constructed linear models to predict a baby's birth weight. Using this model, we wanted to determine possible interventions for a pregnant woman to alter the delivery weight of the baby, as well as to determine the extent to which the weight can be predicted based on the given information about the baby's parents. However, from our analysis, we determined that the model is unsuited for causal inference. Furthermore, although the model does not overfit, our analysis demonstrated shortcomings in terms of precision when predicting birth weight.

For further analysis, we might explore imputation to minimize bias in our dataset. For numerical variables, we could replace the missing values by the mean. For categorical variables, we could replace the missing values by the mode. We may also leverage more sophisticated machine learning methods such as random forests, KNNs, or deep learning.

10 Appendix

10.1 Final Model Summary

```
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
## -3263.2  -288.9     1.8   296.2  4742.4
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 669.4327    67.1470   9.970  < 2e-16 ***
## PREG_LEN8                  -341.2629    63.8595  -5.344 9.11e-08 ***
## PREG_LEN9                   -70.8245    54.4715  -1.300 0.193531
## PREG_LENEarly             -1298.7271    78.5353 -16.537  < 2e-16 ***
## PREG_LENLate               -486.0230   220.9719  -2.199 0.027846 *
## M_Ht_In                      29.4585     0.5618  52.433  < 2e-16 ***
## MRAVE6                      -18.9812     1.7105 -11.097  < 2e-16 ***
## SEXM                        116.5869     3.0171  38.642  < 2e-16 ***
## BMI                          16.4867     0.2879  57.263  < 2e-16 ***
## WTGAIN_PER                  883.9004    16.4083  53.869  < 2e-16 ***
## PRIORLIVETRUE               128.0624     3.3242  38.525  < 2e-16 ***
## CIG_0TRUE                  -197.0948    54.9352  -3.588 0.000334 ***
## NO_RISKS                     90.0103     3.8886  23.148  < 2e-16 ***
## RDMETH_REC                    9.3614     1.6024   5.842 5.18e-09 ***
## PREVIS                        6.4970     1.0965   5.925 3.13e-09 ***
## ATTEND                       25.7226     2.2589  11.387  < 2e-16 ***
## MBSTATE_REC                  47.4383     4.2523  11.156  < 2e-16 ***
## FRACE6                      -22.2585     1.7217 -12.928  < 2e-16 ***
## PAY_REC                      19.3510     2.3795   8.132 4.25e-16 ***
## LD_INDLY                     23.8135     3.4710   6.861 6.89e-12 ***
## FEDUC                         8.6329     1.2521   6.895 5.44e-12 ***
## NO_MMORB                    -92.5574    12.5585  -7.370 1.72e-13 ***
## BFACIL                       24.1557     5.6651   4.264 2.01e-05 ***
## FAGECOMB                     -1.1534     0.2588  -4.457 8.32e-06 ***
```

```
## NO_INFEC                     21.0743    10.7060   1.968 0.049018 *
## RESTATUS                    -11.1342     2.7538  -4.043 5.28e-05 ***
## MEDUC                         7.3097     2.5170   2.904 0.003683 **
## PRECARETRUE                  12.0915    51.7849   0.233 0.815377
## DMAR                        -40.0643     3.8891 -10.302  < 2e-16 ***
## PREG_LEN8:PREVIS              2.3277     1.4952   1.557 0.119524
## PREG_LEN9:PREVIS             -1.2635     1.2041  -1.049 0.294053
## PREG_LENEarly:PREVIS         42.5822     2.4426  17.434  < 2e-16 ***
## PREG_LENLate:PREVIS           3.6253     3.9908   0.908 0.363660
## PREG_LEN8:MEDUC             -28.9368     3.4343  -8.426  < 2e-16 ***
## PREG_LEN9:MEDUC              -2.6232     2.5852  -1.015 0.310259
## PREG_LENEarly:MEDUC         -75.5421     6.5383 -11.554  < 2e-16 ***
## PREG_LENLate:MEDUC          -21.7878     9.3236  -2.337 0.019449 *
## CIG_0TRUE:PRECARETRUE       102.2863    53.1934   1.923 0.054494 .
## PREG_LEN8:PRECARETRUE      -106.8422    65.1086  -1.641 0.100804
## PREG_LEN9:PRECARETRUE       -19.7858    55.6356  -0.356 0.722116
## PREG_LENEarly:PRECARETRUE  -282.4792    78.8215  -3.584 0.000339 ***
## PREG_LENLate:PRECARETRUE    431.5071   224.9434   1.918 0.055077 .
## PREG_LEN8:CIG_0TRUE         -20.9958    21.2412  -0.988 0.322936
## PREG_LEN9:CIG_0TRUE          -4.6454    16.5995  -0.280 0.779593
## PREG_LENEarly:CIG_0TRUE     161.2289    36.4611   4.422 9.79e-06 ***
## PREG_LENLate:CIG_0TRUE      -48.6144    52.2220  -0.931 0.351899
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 476.4 on 99954 degrees of freedom
## Multiple R-squared:  0.3139, Adjusted R-squared:  0.3136
## F-statistic:  1016 on 45 and 99954 DF,  p-value: < 2.2e-16
```

10.2 Code

```
knitr::opts_chunk$set(
  echo = FALSE,          # don't show code
  warning = FALSE,       # don't show warnings
  message = FALSE,       # don't show messages (less serious warnings)
  cache = FALSE,         # set to TRUE to save results from last compilation
  fig.align = "center"   # center figures
)

library(dplyr)
library(ggplot2)
library(MASS)
library(car)
library(gridExtra)
birth <- read.csv("data/US_births(2018).csv")
# 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)
# 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)
# Reduce the dimensionality of the dataset
```

```r
# 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))
# 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))

# 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)
# TODO: REMOVE THIS
EDA_df <- read.csv("data/EDA.csv")
Train <- read.csv("data/Train.csv")
Test <- read.csv("data/Test.csv")

EDA_df$PRECARE <- factor(EDA_df$PRECARE)
Train$PRECARE <- factor(Train$PRECARE)
Test$PRECARE <- factor(Test$PRECARE)

# EDA

# response variable
fig1a <- ggplot(EDA_df, aes(x = DBWT)) +
  geom_histogram(color = "black", fill = "white") +
  labs(title = "Distribution of Birth Weights",
       caption = "Figure 1a: Distribution of birth weights") +
  theme(text = element_text(size = 10),
        plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(size = 8))

fig1b <-
  ggplot(data = EDA_df, aes(sample = DBWT)) +
  stat_qq(shape = 1) + stat_qq_line(color = "red") +
```

```
  labs(
    x = "Theoretical Quantiles",
    y = "Sample Quantiles",
    title = "Normal Q-Q Plot",
    caption = "Figure 1b: Normal Q-Q plot for birth weights"
  ) +
  theme(text = element_text(size = 10),
        plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(size = 8))

# Measure of symmetry
DBWT_sym = (quantile(EDA_df$DBWT, 0.75) - median(EDA_df$DBWT)) /
  (median(EDA_df$DBWT) - quantile(EDA_df$DBWT, 0.25))

fig2a <- ggplot(EDA_df, aes(x = WTGAIN_PER, y = DBWT)) +
  geom_point(alpha = 0.2) +
  geom_smooth() +
  labs(title = "DBWT vs. WTGAIN_PER",
       caption = "Figure 2a: Birth weight vs. percentage of weight gained") +
  theme(text = element_text(size = 10),
        plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(size = 8))

fig2b <- ggplot(EDA_df, aes(x = CIG_0, y = DBWT)) +
  geom_boxplot() +
  labs(title = "DBWT vs. CIG_0",
       caption = "Figure 2b: Bbirth weight vs. mother's smoking status") +
  theme(text = element_text(size = 10),
        plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(size = 8))

fig3a <- ggplot(EDA_df, aes(x = PRECARE, y = DBWT)) +
  geom_boxplot() +
  labs(title = "DBWT vs. PRECARE",
       caption = "Figure 3a: Birth weight vs. mother's prenatal care status") +
  theme(text = element_text(size = 10),
        plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(size = 8))

# 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)

fig3b <- ggplot(EDA_df, aes(x = PRECARE, y = DBWT)) +
  geom_boxplot(alpha = 0.3) +
  labs(title = "DBWT vs. PRECARE",
       caption = "Figure 3b: Birth weight vs. binarized prenatal care status") +
  theme(text = element_text(size = 10),
        plot.title = element_text(hjust = 0.5),
        plot.caption = element_text(size = 8))

# # Interactions
```

```r
fig4a <- ggplot(EDA_df, aes(x = PRECARE, y = DBWT)) +
  geom_boxplot(aes(fill = SEX)) +
  labs(title = "DBWT vs. PRECARE",
       caption = "Figure 4b: Interaction between sex and prenatal care status")  +
  theme(
    text = element_text(size = 10),
    plot.title = element_text(hjust = 0.5),
    plot.caption = element_text(size = 8),
    legend.position = "bottom"
  )

fig4b <- ggplot(EDA_df, aes(x = BMI, y = DBWT)) +
  geom_point(position = "jitter", aes(colour = PRECARE), alpha = 0.5) +
  geom_smooth(method = "lm", aes(colour = PRECARE)) +
  labs(title = "DBWT vs. BMI",
       caption = "Figure 4b: Interaction between prenatal care status and BMI")  +
  theme(
    text = element_text(size = 10),
    plot.title = element_text(hjust = 0.5),
    plot.caption = element_text(size = 8),
    legend.position = "bottom"
  )
grid.arrange(fig1a, fig1b, ncol = 2)
grid.arrange(fig2a, fig2b, ncol = 2)
grid.arrange(fig3a, fig3b, ncol = 2)
grid.arrange(fig4a, fig4b, ncol = 2)
# 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
```

```r
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)

# Type II Anova
Anova(full.lm)

# Final model including only significant interaction terms
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)

formula(final.lm)
# Model diagnostic
par(mfrow = c(2, 2), mai = c(0.20, 0.3, 0.4, 0.1))
plot(final.lm, col = rgb(red = 0, green = 0, blue = 0, alpha = 0.15))

# 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
fig6a <- ggplot(Train, aes(x = BMI, y = stu_res)) +
    geom_point(alpha = 0.1) +
    geom_smooth()

# 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

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

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

# 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)

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

true.test <- Test$DBWT
test.pred.df <- data.frame(cbind(true.test, pred.test))
fig6b <- ggplot(test.pred.df, aes(x = true.test, y = pred.test)) +
  geom_point()
grid.arrange(fig6a, fig6b, ncol = 2)
# prediction intervals of five points
five_samples <- Test %>% sample_n(3, replace = TRUE)
predict(final_test.lm, five_samples, interval = "prediction")

summary(final_test.lm)
```