STAT 331 Final Project

Chang Liu, Yusong Weng, Nicole Seyler

03/08/2021

Summary

The goal of this report is to investigate which variables should be included in a model that most accurately predicts child birth weight and find a relatively sound model for predicting birth weight accurately. We first removed outliers, checked if current transformations(or lack of) of the variables are suitable, dealt with multicollinearity. Then, we used Automatic Selection of Forward Selection, Backword Selection, Stepwise selection with AIC and BIC to find candidate models.

We used 10-fold cross validation on the models found and computed the mean squared error to find the model with the least mean squared error (Model 1) and the simple model that's best for explanation (Model 3). We checked our model assumptions after we found our optimal model for prediction, which is Model 1. We were able to find a list of covariates from Model 1 (because all the covariates we obtained from Model 3 is already included in our list for Model 1) that best predicts and explains birth weight. Furthermore, we have two useful models -M odel 1 that can predict child birth weight accurately (with relatively low mean squared error) and Model 3 that can predict child birth weight relatively accurately (less accurate than Model 1) but only need to use half of the covariates compared to Model 1.

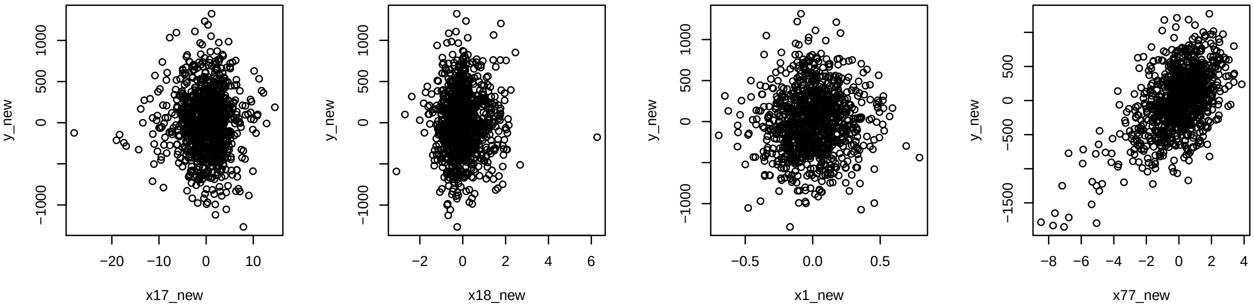
Objective

The objective of this report is to investigate which variables should be included in a model that most accurately predicts birth weight. In particular, we are interested in determining which covariates are most relevant to creating this best model for predication and explanation. Since we wanted to investigate the variables affect birth weight using a statistic approach, we assume that no background knowledge exists in terms of which variables affect birth weight to avoid confirmation bias, which makes our method of Automatic Selection a suitable approach.

Exploratory Data Analysis

For our exploratory data analysis, we wanted to get a sense of the relationship of each covariate with the outcome, and also determine if there were any potential outliers. To do this, for each covariate x^* we regressed y on the other covariates x_j, found the fitted values and residuals, and then regressed x^* on the other covariates x_j and got the fitted values and residuals. We then plotted the residuals e_y against e_{x^*}.

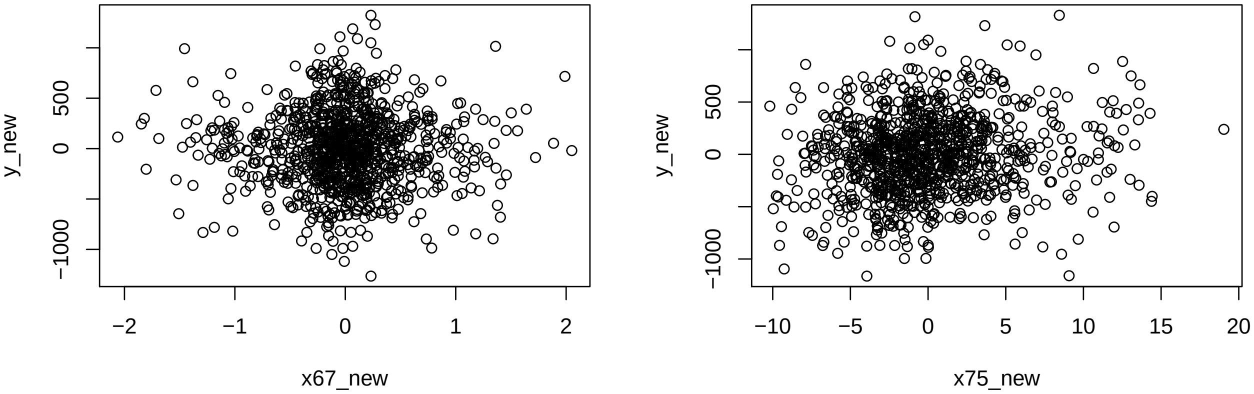
The figure below shows four examples of such plots. The two on the left show plots where we identified x-outliers. while the third plot is an example of a plot where we saw no issues. The plot on the right is a plot where we could easily see a linear relationship.



In total, we identified 42 outliers, and removed them from the dataset.

We also noticed that some of the plots looked vaguely like they might perform better if they were transformed, but since it was not a clear problem, we decided to leave them untransformed and maintain interpretability. Below are some examples of plots which we thought vaguely looked like they would be improved with a quadratic transformation. %talk about what the transformations would be, print out the numeric stuff We also used R^2 values to determine which transformations were the best numerically. We considered the following transformations: x^2, $\sqrt{x}$, $log(x)$, e^x, $(e^x)^2$, $\sqrt{(e^x)}$.

The two figures below are the two instances we saw that most appeared to need transformation. There is a slight quadratic appearance to each of them, with the plot on the left appearing concave-up, and the plot on the right appearing concave-down. Still, these are hardly patterns at all, and only really appear because there is less data around the edges of the graph. These patterns may dissappear entirely if we had more data. Furthermore, we can see that the R^2 values hardly differ at all when transforming these covariates. For the 67th covariate, the R^2 value for the untransformed data is 2.346344e-04, while if we transform it using the function x^2, then the R^2 value only drops to 9.584646e-05. Likewise, for the 75th covariate, the R^2 value for the untransformed data is 0.01695889, while if we transform it using the function x^2, then the R^2 value only drops to 0.01614864.So, we decided to leave the data untransformed.



[1] “x67:” [1] “numeric” [1] “r squared values:” [1] 2.346344e-04 9.584646e-05 2.943289e-04 3.404270e-04 8.914341e-04 9.793620e-04 1.692032e-04 function(x){return(xˆ2)}

[1] “x75:” [1] “numeric” [1] “r squared values:” [1] 0.01695889 0.01614864 0.01717182 0.01724086 function(x){return(xˆ2)}

Method

First, we splitted our data into training and test data, with an 80-20 split. We chose the 80-20 split because that is the approach most often taken and proves to be most effective. In other words, it is an excellent rule of thumb for statisticians.

We used the training data to build a collection of models, and chose the ‘best’ model using cross-validation. By using cross-validation, we avoid having to split our training data again, which allows us to use more of our data to fit the model, and it should allow us to create a better model. We will use the test data to evaluate how successful our final model is as a predictive model.

Multicollinearity

Next, we wanted to eliminate multicollinearity.

We would expect strong multicollinearity among the explanatory variables in this dataset, since the cereal, fast food, fish, legumes, and fruit consumptions of the mother tend to relate to each other based on whether the mother's eating habit is healthy. Furthermore, the concentration of beneficial and harmful chemicals in mothers are related, as well as the body mass index and weight of the mother. There are many more related variables such as humidity, temperature temperature and pressure during pregnancy.

If all the explanatory variables are used in a multiple linear regression model regardless of the presence of high multicollinearity, then the statistical significance of the independent variable will be undermined by the model. Furthermore, the variance of the estimated regression coefficient would be inflated. Hence, we really need to deal with multicollinearity carefully in our dataset to ensure the model's accuracy, explainability and predictability.

As a rule of thumb, a VIF larger than 10 indicates high multicollinearity. Our approach is that beginning with a model, regressing birth weight on all the explanatory variables, remove explanatory variables one by one, excluding the covariate with the largest VIF each time until there are no more variables with VIF $>$ 10.

We prefer this approach compared to excluding all explanatory variables with VIF $>$ 10 simultaneously because the model could retain the maximum number of covariates. For instance, assuming we have 2 covariates which are highly correlated out of 80 covariates, then the two in question will have very high VIFs. However, there is no need to exclude them both. If we exclude 1 and calculate the updated VIFs, the other will have a much lower VIF (since it is uncorrelated with the other 78 covariates).

```
## [1] 77
```

After removing the variables with high multicollinearity, we have 77 explanatory variables left.

Automatic Selection

Now that we are satisfied with the state of our data, we can move on to creating the best predictive model of birth weight.

A regression model fitted in cases where the sample size is not much larger than the number of predictors will perform poorly in terms of out-of-sample accuracy. In these cases (and in our case), reducing the number of predictors in the model will improve out-of-sample accuracy and make our model more generalizable.

Since we have 77 covariates, All Possible Models or Best Subsets methods would require too much computational power as this would require fitting 2^{77} = 1.511157e+23 models. Manual model selection is also unsuitable due to the sheer size of the covariates.

Instead, we will use Automatic Model Selection methods (Forward Selection, Backward Selection, and Stepwise Selection) to build a collection of models based on AIC and BIC, and then use 10-fold cross-validation to choose the best predictive model of this collection. It may not truly be the best predictive model, but it will provide us with the a list of variables that could be looked at for future studies. When other statisticians is trying to select variables from our provided list of variables, they could choose the variables without confirmation bias since the list of variables we selected is using a statistical approach with Automatic Selection.

Automatic Selection also provides a reproducible and objective way to reduce the number of predictors compared to manually choosing variables based on expert opinion which, more often than we would like to admit, is biased towards proving one's own hypothesis.

This report is an exploratory investigation. To avoid confirmation bias and appraoch this issue from a purely statistical point of view, we assume background knowledge is not available with which factors affect birth weight of infants.

Forward Selection Forward selection at most could consider $\frac{(p+1)p}{2}$ models, although most times the number of models considered is less than that. Forward selection can even be applied in settings where the number of variables under consideration is larger than the sample size!

However, forward selection is not guaranteed to find the "best" model because not all models are considered - it would be way too expensive computationally if we consider all the models since we have 77 covariates!

Using forward selection, we determine the two following models by using AIC and BIC as criteria respectively:

```
## e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None + hs_wgtgain_None +
##     hs_pfoa_m_Log2 + e3_asmokcigd_p_None + h_mbmi_None + h_edumc_None +
##     hs_dmtp_madj_Log2 + hs_hg_m_Log2 + hs_mepa_madj_Log2 + hs_etpa_madj_Log2 +
##     h_dairy_preg_Ter
```

```
## e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None + hs_wgtgain_None +
##     hs_pfoa_m_Log2 + e3_asmokcigd_p_None + h_mbmi_None + hs_dmtp_madj_Log2
```

Backward Selection Backward selection sometimes perform better than forward selection because it has the advantage of considering the effects of all the explanatory variables simultaneously in the beginning.

This is especially important in case of collinearity (which is highly relevant to our model as explained before) because backward stepwise may be forced to keep them all in the model unlike forward selection where none of them might be entered.

Using backward selection, we determine the two following models by using AIC and BIC as criteria respectively:

```
## e3_bw ~ h_abs_ratio_preg_Log + h_pm10_ratio_preg_None + h_dairy_preg_Ter +
##     h_meat_preg_Ter + hs_cs_m_Log2 + h_pressure_preg_None + hs_hcb_madj_Log2 +
##     hs_dmtp_madj_Log2 + hs_pfoa_m_Log2 + hs_etpa_madj_Log2 +
##     hs_mepa_madj_Log2 + hs_meohp_madj_Log2 + hs_sumDEHP_madj_Log2 +
##     e3_asmokcigd_p_None + h_bro_preg_Log + e3_sex_None + h_mbmi_None +
##     hs_wgtgain_None + e3_gac_None + h_edumc_None
```

```
## e3_bw ~ hs_dmtp_madj_Log2 + hs_pfoa_m_Log2 + e3_asmokcigd_p_None +
##     h_bro_preg_Log + e3_sex_None + h_mbmi_None + hs_wgtgain_None +
##     e3_gac_None
```

Stepwise Selection Stepwise selection is a compromise between forward selection and backward selection.

Using stepwise selection, with the base model being the model with no covariates, just the intercept, we determine the two following models by using AIC and BIC as criteria respectively:

```
## e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None + hs_wgtgain_None +
##     hs_pfoa_m_Log2 + e3_asmokcigd_p_None + h_mbmi_None + h_edumc_None +
##     hs_dmtp_madj_Log2 + hs_hg_m_Log2 + hs_mepa_madj_Log2 + hs_etpa_madj_Log2 +
##     h_dairy_preg_Ter
```

```
## e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None + hs_wgtgain_None +
##     hs_pfoa_m_Log2 + e3_asmokcigd_p_None + h_mbmi_None + hs_dmtp_madj_Log2
```

For our Automatic Selection, we wanted to minimize AIC or BIC. We could see that BIC yields smaller models from our results because BIC is more restrictive. BIC is recommended when working with larger sample size. We have a quite large sample size in our case so we were able to apply BIC.

Larger models have more risks of overfitting, they fit better and have a smaller RSS but uses more parameters. BIC penalizes larger model more heavily and from our results, we could see that BIC tend to prefer smaller models compared to AIC. Furthermore, AIC is better for prediction but BIC is better for explanation since BIC have simpler models are always better for explaining and BIC allows consistent estimation of the underlying data generating process.

From the models we obtained using BIC as criteria (thus good for explaining), we could see that we essentially had the same model using the three approaches (forward selection, backward selection, and stepwise selection) because the covariates selected are the same.

Furthermore, note that the model we found using Forward Selection with AIC as our criteria is the same model as the model we found using Stepwise Selection with AIC. They would bring out interesting results once we look at the cross validation results in our following sections.

Automatic Selection - Again

We attempted applying stepwise selection using the models we found using forward and backward selection in step 1 to investigate if results are different.

Specifically, we set the base model to the models we found using forward and backward selection with AIC and BIC from the steps above.

```
## e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None + hs_wgtgain_None +
##     hs_pfoa_m_Log2 + e3_asmokcigd_p_None + h_mbmi_None + h_edumc_None +
##     hs_dmtp_madj_Log2 + hs_hg_m_Log2 + hs_mepa_madj_Log2 + hs_etpa_madj_Log2 +
##     h_dairy_preg_Ter
```

```
formula(Mstep_2)
```

```
## e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None + hs_wgtgain_None +
##     hs_pfoa_m_Log2 + e3_asmokcigd_p_None + h_mbmi_None + hs_dmtp_madj_Log2
```

```
formula(Mstep_3)
```

```
## e3_bw ~ h_abs_ratio_preg_Log + h_pm10_ratio_preg_None + h_dairy_preg_Ter +
##     h_meat_preg_Ter + hs_cs_m_Log2 + h_pressure_preg_None + hs_hcb_madj_Log2 +
##     hs_dmtp_madj_Log2 + hs_pfoa_m_Log2 + hs_etpa_madj_Log2 +
##     hs_mepa_madj_Log2 + hs_meohp_madj_Log2 + hs_sumDEHP_madj_Log2 +
##     e3_asmokcigd_p_None + h_bro_preg_Log + e3_sex_None + h_mbmi_None +
##     hs_wgtgain_None + e3_gac_None + h_edumc_None
```

```
formula(Mstep_4)
```

```
## e3_bw ~ hs_dmtp_madj_Log2 + hs_pfoa_m_Log2 + e3_asmokcigd_p_None +
##     h_bro_preg_Log + e3_sex_None + h_mbmi_None + hs_wgtgain_None +
##     e3_gac_None
```

As expected, using BIC as the criteria results in more parsimonious models than if AIC was used. We anticipate that models selected this way will be more likely to be our most explainable model than their AIC counterparts, and they are even less susceptible to overfitting. We also notice that the models selected using stepwise selection when we were fitting again are the same as the ones as when we were fitting for the first time using forward selection, which is great since it again confirms the choice of explanatory variables.

Cross Validation

We are considering using k-fold cross validation or Leave-One-Out cross validation (LOOCV) for the model.

When we perform LOOCV, we are in effect averaging the outputs of n fitted models, each of which is trained on an almost identical set of observations; therefore, these outputs are highly (positively) correlated with each other.

In contrast, when we perform k-fold CV with $k < n$, we are averaging the outputs of k fitted models that are somewhat less correlated with each other since the overlap between the training sets in each model is smaller. As the mean of many highly correlated quantities has higher variance than does the mean of many quantities that are not as highly correlated, the test error estimate resulting from LOOCV tends to have higher variance than does the test error estimate resulting from k-fold CV.

Now the variance in fitting the model tends to be higher if it is fitted to a small dataset (as it is more sensitive to any noise/sampling artifacts in the particular training sample used). This means that 10-fold cross-validation is likely to have a high variance (as well as a higher bias) if you only have a limited amount of data, as the size of the training set will be smaller than for LOOCV. So k-fold cross-validation can have variance issues as well, but for a different reason. This is why LOOCV is often mosre suitable when the size of the dataset is small.

Another the main reason for using LOOCV is that it is computationally inexpensive for some models (such as linear regression, most kernel methods, nearest-neighbour classifiers, etc.). However, our dataset of 1000 in size is large enough for the data to be more suited for using k-fold cross validation and small enough to justify the computational power. Hence, k-fold cross validation is a better choice for evaluating our model.

Furthermore, there is a bias-variance trade-off associated with the choice of k in k-fold cross-validation. Typically, given these considerations, one performs k-fold cross-validation with k=5 or k=10, as these values have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance. In our case, we're performing k-fold cross-validation with k=10. Although using 10 fold would be more computationally expensive, it evaluates our model against more unseen data without compromising on bias or variance too much.

Now we can determine our best model using 10-fold cross validation.

```
##  Mfwd_AIC  Mfwd_BIC  Mbwd_AIC  Mbwd_BIC Mstep_AIC Mstep_BIC   Mstep_1   Mstep_2
##  153556.8  158682.7  151001.4  158682.7  153556.8  158682.7  153556.8  158682.7
##   Mstep_3   Mstep_4
##  151001.4  158682.7
```

It appears as though the model that we selected using backward selection with the criteria AIC will be the best in terms of prediction. This makes sense statistically since AIC penalizes for extra covariates less than BIC, and the model selected using AIC tend to predict better while the model selected using BIC tend to be simpler but easier to explain.

In the end, we obtained 3 different model.

Model 1: The Backword AIC model (Mbwd_AIC), which is the best for prediction. The model is obtained from backward selection using AIC as the criteria.

Model 2: The Forward AIC model (Mfwd_AIC, Mstep_AIC). We obtained the same model from Forward Selection and Stepwise Selection using AIC as criteria. The mean squared error is larger than the Backward AIC model, but

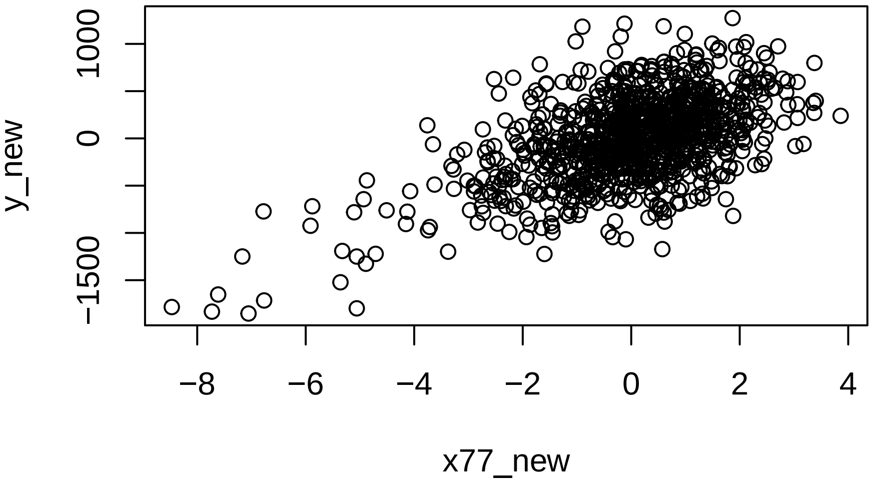
Model 3: The BIC Model, which is the best for explanation. We obtained the same model from Mfwd_BIC, Mbwd_BIC, Mstep_BIC using three approaches of Automatic Selection with BIC as the criteria.

Now we need to test the assumptions of our model with our Model 1, our best model for prediction.

Testing the Assumptions

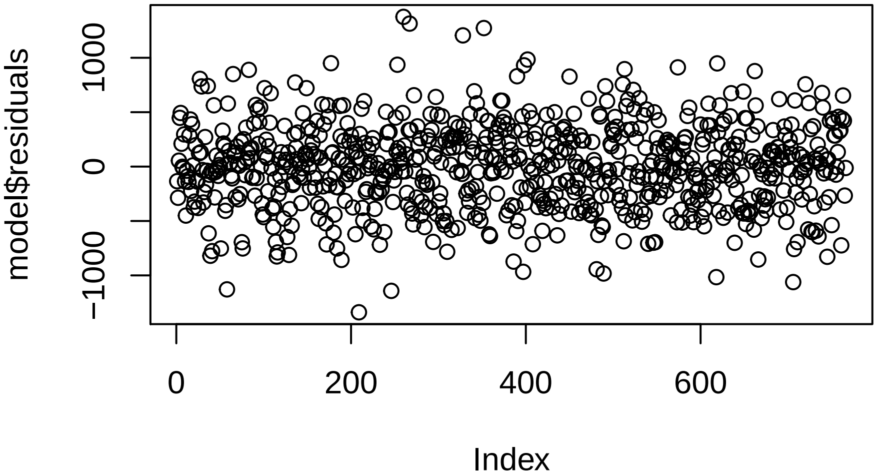
Linearity

We want to check if the underlying model is truly linear. Earlier in EDA we saw that the covariates generally appeared to be linear. A good example of this was the e3_yearbir_None covariate which was the year of birth.



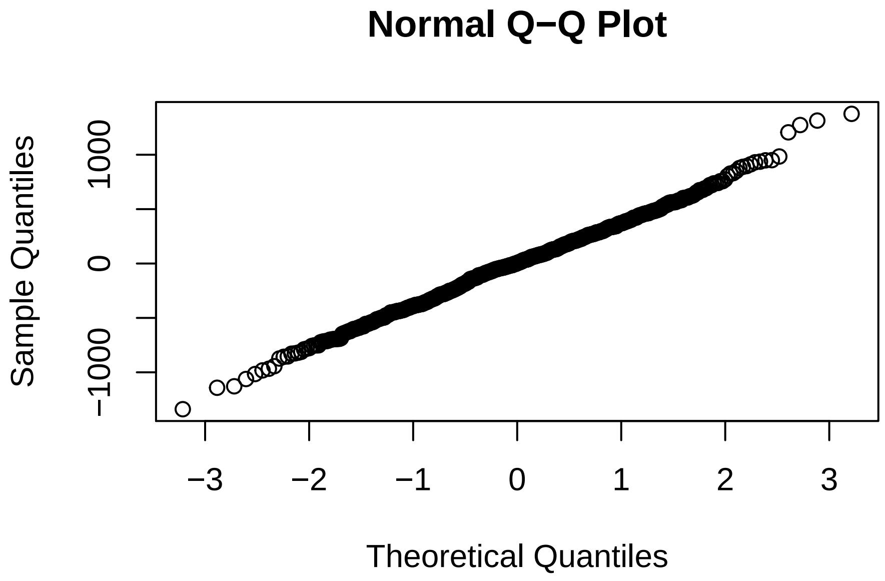
Independence

Our error terms should be independent. This is difficult to visualize since our data is not exactly time-series data, so even though we can plot the residuals, since there is no order to our data, it will not be particularly useful. Rather, we usually evaluate this assumption based on how the data was collected. We don't know how this data was collected. It is possible that the observations were made in such a way that there will be clusters, but hopefully the people conducting the study did not do that. It is just a possibility we have to be aware of.



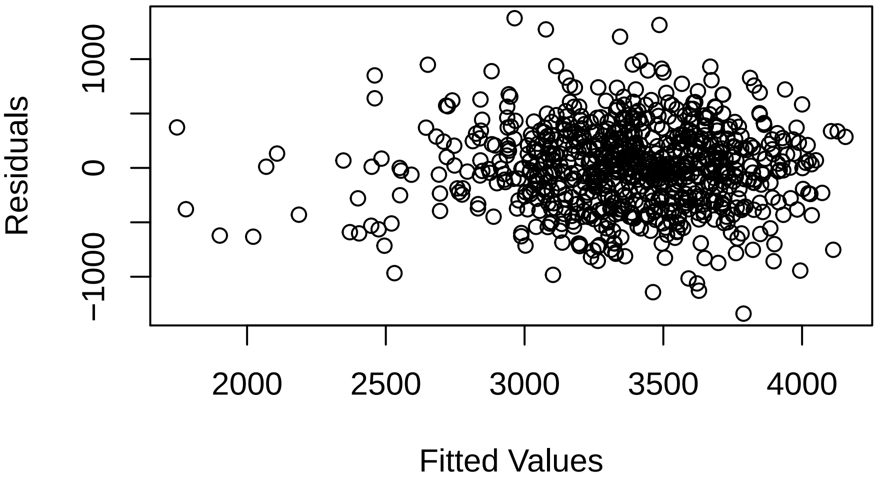
Normality

The error terms should be normally distributed.



In the figure above we can see that the residuals appear to be normally distributed. There is some deviation at the tails, so it could be argued that the distribution is slightly more heavy-tailed than normal, but, ultimately, these residuals look fairly normally distributed, and we can consider this assumption to be met.

Equal variance



There does appear to be a funnel shape to the plot above, suggesting that the variance may not be constant. However, it is also possible that we just do not have enough data, and this effect would disappear if we had more observations. However, we know that even if the homoskedasticity assumption does not hold, the LS estimators will remain unbiased. As we are not trying to produce confidence intervals, but rather a predictive model, we can feel comfortable moving forward with our model despite some concerns about this assumption not being met.

Hence, our model meets the necessary assumptions.

Results

We have two resulting models:

Model 1: The Backword AIC model (Mbwd_AIC), which is the best for prediction. The model is obtained from backward selection using AIC as the criteria.

Model 3: The BIC Model, which is the best for explanation. We obtained the same model from Mfwd_BIC, Mbwd_BIC, Mstep_BIC using three approaches of Automatic Selection with BIC as the criteria.

Discussion

Model 1

From our Model 1, we have a supplemental list of variables that affects the outcomes as follows. These explanatory variables are all good candidates to include in building good model for predication accuracy.

```
## e3_bw ~ h_abs_ratio_preg_Log + h_pm10_ratio_preg_None + h_dairy_preg_Ter +
##     h_meat_preg_Ter + hs_cs_m_Log2 + h_pressure_preg_None + hs_hcb_madj_Log2 +
##     hs_dmtp_madj_Log2 + hs_pfoa_m_Log2 + hs_etpa_madj_Log2 +
##     hs_mepa_madj_Log2 + hs_meohp_madj_Log2 + hs_sumDEHP_madj_Log2 +
##     e3_asmokcigd_p_None + h_bro_preg_Log + e3_sex_None + h_mbmi_None +
##     hs_wgtgain_None + e3_gac_None + h_edumc_None
```

We tested Model 1, Model 2, Model 3 for against our test data and found their mean squared error. This further confirmed our conclusion that Model 1 above is the best model in terms of predication since it has the smallest mean squared error. Furthmore, note that Model 1 is likely not overfitted (fitted just right) since it generalizes over to the test data quite well.

```
## [1] 141525.2
```

```
## [1] 150139.4
```

```
## [1] 160408.2
```

Model 3

From our Model 3, we have a good model for explaining which variables should be included in models that predict the outcome. The following 8 covariates are of importance for our model, and they're the following:

Covariates	Descriptions
e3_sex_None	Child sex (female / male)
hs_wgtgain_None	Maternal weight gain
h_mbmi_None	Maternal pre-pregnancy body mass index (kg/m2) during pregnancy (kg)
e3_gac_None	Gestational age at birth (week)
e3_asmokcigd_p_None	Maternal active Tobacco Smoke pregnancy mean nb cig/day
h_bro_preg_Log	Total concentration of Brominated during pregnancy
hs_pfoa_m_Log2	Perfluorooctanoate (PFOA) in mother
hs_dmtp_madj_Log2	Dimethyl thiophosphate (DMTP) in child adjusted for creatinine

The explanatory variables such as child sex, maternal weight gain, body mass index during pregnancy, gestational age would make sense logically to be associated with the weight of the infant. However, variables such as Tobacco smoked during preganancy and PFOA in mother may not seem immediately obvious to influence children's birth weight. Although currently there are researches investigating the effects of theses factors on child birth weight, there have been more limited researches on these factors compared to more obvious factors mentioned above.

Furthermore, factors such as the concentration of Bromine during pregnancy and DMTP in children are still factors that are not studied by researchers. Although it is entirely possible that these factors are noises in the dataset and model selection process, it is still worth further investigation to see if these chemical compound affect child birth weight.

It is simple for researchers to disregard the factors that may not seem initially obvious, and here is why Automatic Model Selection is helpful for us to isolate the important variables to see if there are some things that we are missing. Automatic Model Selection is useful in providing statisticians with a set of variables to focus on before building a best possible model.

Limitations

Stepwise selection does not consider all possible combination of potential predictors. Although we had a computational advantage over methods that do consider all these combinations, it is not guaranteed to select the best possible combination of variables. So it is still possible that there are other models that performs better. However, we still prefer the stepwise approach since it's way too computationally expensive to try other combinations. It may be worthwhile to try to fit LASSO and Ridge and look at dimensiality reduction to supplement our discussion for future improvement.

The regression coefficients, confidence intervals, p-values and r squared values outputted by stepwise selection are biased and cannot be trusted. The confidence intervals will appear narrower, the p-values Will appear smaller, and the regression coefficients and r squared values would appear larger. However, these are not relevant for our purposes since we are exploring and investigating the useful models and variables to our outcome for future studies. Theoretically, in the future, there will be other dataset from researchers where other statisticians could use our selected models and variables as a suggestion and reference while building models. This is still a point worth noting considering we can't refer to confidence intervals, p-values and standard deviations while using Automatic Selection.

In addition, the selection of variables using a stepwise regression will be unstable because many variable combinations can fit the data in a similar way. This is especially when we have a small sample size compared to the number of variables we want to study. In our case though, our sample size is relatively sound compared to the number of variables we want to study, however, this is still an issue worth noting.

Appendix

R Code

```
load("/Users/chang/OneDrive - University of Waterloo/3B/STAT 331/project/pollution.Rdata")

set.seed(331)
# a function to plot y against xi isolated
IsolatedPlot=function(data, x_index){
  y_mod=lm(paste0("y~.-x",x_index), data=data)
  x_mod=lm(paste0("x",x_index,"~.-y"),data=data)
  y_new=residuals(y_mod)
  x_new=residuals(x_mod)
  plot(x_new,y_new,xlab=paste0("x",x_index,"_new"))
  print(which(x_new>=30)) # find the indices of outliers
}

data=pollution
newnames=c("y")
for(i in 1:79){newnames=c(newnames,paste0("x",i))}
names(data)=newnames

par(mfrow=c(1,4))

IsolatedPlot(data,17)
IsolatedPlot(data,18)
IsolatedPlot(data,1)
IsolatedPlot(data,77)
outlier_rows <- c(114,316,530,884,896,314,723,882,55,485,558, 289,
76,604,770, 222,452,702,909,921,848,413,433,771,
                922,184,762,910,912,938,40, 142,850,379,716,455,967,566,631,853,984,162)

pollution_no <- pollution[-outlier_rows,]

# helper
CheckFit=function(data, x_index, plot){
  y_mod=lm(paste0("y~.-x",x_index), data=data)
  x_mod=lm(paste0("x",x_index,"~.-y"),data=data)
  y_new=residuals(y_mod)
  x_new=residuals(x_mod)
  if (plot) plot(x_new,y_new,xlab=paste0("x",x_index,"_new"))
  isolated_mod=lm(y_new~x_new)
  rsq=summary(isolated_mod)$r.squared
  return(rsq)
}


# lofunc - all covariates
# lofunc2 - covariates > 0
# lofunc3 - covariates < 15
SelectTrans=function(data,lofunc, lofunc2, lofunc3, plot_index){
  nCov=length(colnames(data))-1
  for(x_index in 1:nCov){
  print(paste0("x",x_index,":"))
  print(class(data[,x_index+1]))
  if(class(data[1,x_index+1])!="factor"){
    bestFun=function(x){return(x)}
    rsq_min=CheckFit(data,x_index,x_index==plot_index)
    print(paste0("r squared values:"))
    rsq_list=NULL
    rsq_list=c(rsq_list,rsq_min)
    for(f in lofunc){
      data_mod=data
      data_mod[,x_index+1]=f(data_mod[,x_index+1])
      rsq_new=CheckFit(data_mod,x_index,FALSE)
      rsq_list=c(rsq_list,rsq_new)
      if(rsq_new<rsq_min){
        bestFun=f
        rsq_min=rsq_new
      }
    }
    if(sum(data[,x_index+1]>0)==length(data[,x_index+1])){#all entries positive
      for(f in lofunc2){
        data_mod=data
        data_mod[,x_index+1]=f(data_mod[,x_index+1])
        rsq_new=CheckFit(data_mod,x_index,FALSE)
        rsq_list=c(rsq_list,rsq_new)
        if(rsq_new<rsq_min){
          bestFun=f
          rsq_min=rsq_new
        }
      }
    }
    if(sum(data[,x_index+1]<15)==length(data[,x_index+1])){#all entries < 30
```

```
      for(f in lofunc3){
        data_mod=data
        data_mod[,x_index+1]=f(data_mod[,x_index+1])
        rsq_new=CheckFit(data_mod,x_index,FALSE)
        rsq_list=c(rsq_list,rsq_new)
        if(rsq_new<rsq_min){
          bestFun=f
          rsq_min=rsq_new
        }
      }
    }
    print(rsq_list)
    print(bestFun)
    cat("\n")
  }
  }
}

f1=function(x){return(x^2)}
f2=function(x){return(sqrt(x))}
f3=function(x){return(log(x))} # not checking log base 2 because log2*log2x=logx
f4=function(x){return(exp(x))}
f5=function(x){return(exp(2*x))} # (e^x)^2
f6=function(x){return(exp(0.5*x))} # sqrt(e^x)

par(mfrow=c(1,2))

IsolatedPlot(data,67)
IsolatedPlot(data,75)

#First we need to split the data into training and test data
set.seed(331)
ntot <- nrow(pollution_no)
n <- ntot
ntrain <- 766 # 0.8 * 958
ntest <- ntot - ntrain
indices <- seq(from = 1, to = ntot, by = 1)
train.ind <- sample(indices, ntrain, replace = FALSE)
test.ind <- setdiff(indices, train.ind)


library(car)
signal <- TRUE
#set the temporary variable to store the training set
pollution_train <- pollution_no[train.ind, ]
pollution_temp <- pollution_train
#Removing explanatory variables with VIF>10 one by one
while(signal){
  #build the regression model
  model <- lm(e3_bw ~ ., data<-pollution_temp)
  #If the max VIF is more than 10 in current model
  if(max(vif(model)) > 10){
    #then we will drop the corresponding variable which has the max VIF
    #Here using +1, this is because the first column is the predictor
    index_remove=which.max(vif(model))+1
    pollution_temp <- pollution_temp[,-index_remove]
    pollution_no <- pollution_no[,-index_remove]
  }
  #If the max VIF is less than 10 in current model
  else{
    #set the signal to false
    signal <- FALSE
  }
}


M0 <- lm(e3_bw ~ 1, data = pollution_no[train.ind, ])
Mfull <- lm(e3_bw ~ ., data= pollution_no[train.ind, ])
#forward selection
Mfwd_AIC <- step(object = M0, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="forward",
                 trace=0,
                 k=2)

Mfwd_BIC <- step(object = M0, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="forward",
                 trace=0,
                 k=log(n))

formula(Mfwd_AIC)

formula(Mfwd_BIC)

#backward selection
Mbwd_AIC <- step(object = Mfull, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="backward",
                 trace=0,
                 k=2)

Mbwd_BIC <- step(object = Mfull, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="backward",
                 trace=0,
                 k=log(n))
formula(Mbwd_AIC)

formula(Mbwd_BIC)

#stepwise selection
Mstep_AIC <- step(object = M0, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="both",
                 trace=0,
                 k=2)
```

```
Mstep_BIC <- step(object = M0, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="both",
                 trace=0,
                 k=log(n))
formula(Mstep_AIC)

formula(Mstep_BIC)

M0_1 <- Mfwd_AIC
M0_2 <- Mfwd_BIC
M0_3 <- Mbwd_AIC
M0_4 <- Mbwd_BIC

#stepwise selection
Mstep_1 <- step(object = M0_1, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="both",
                 trace=0,
                 k=2)

Mstep_2 <- step(object = M0_2, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="both",
                 trace=0,
                 k=log(n))

Mstep_3 <- step(object = M0_3, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="both",
                 trace=0,
                 k=2)

Mstep_4 <- step(object = M0_4, #base model
                 scope = list(lower=M0, upper=Mfull),
                 direction="both",
                 trace=0,
                 k=log(n))

formula(Mstep_1)

formula(Mstep_2)

formula(Mstep_3)

formula(Mstep_4)

M1 <- Mfwd_AIC
M2 <- Mfwd_BIC
M3 <- Mbwd_AIC
M4 <- Mbwd_BIC
M5 <- Mstep_AIC
M6 <- Mstep_BIC
M7 <- Mstep_1
```

```
M8 <- Mstep_2
M9 <- Mstep_3
M10 <- Mstep_4

Mnames <- expression(M[FULL], M[STEP])

#number of cross-validation replications
kfolds <- 10

#we're only performing CV on the training data, so ntot is 766 here
ntot <- 766

pol <- pollution_no[train.ind, ]
pol <- pol[sample(ntot),] #permute rows

#pol$index <- rep(1:kfolds, each=ntot/kfolds)
temp <- rep(1:10, each=76)
pol$index <- c(temp, 1,2,3,4,5,6)

#storage space
mspe1 <- rep(NA, kfolds)
mspe2 <- rep(NA, kfolds)
mspe3 <- rep(NA, kfolds)
mspe4 <- rep(NA, kfolds)
mspe5 <- rep(NA, kfolds)
mspe6 <- rep(NA, kfolds)
mspe7 <- rep(NA, kfolds)
mspe8 <- rep(NA, kfolds)
mspe9 <- rep(NA, kfolds)
mspe10 <- rep(NA, kfolds)


for(ii in 1:kfolds){
  train.ind.cv <- which(pol$index!=ii)

  M1.cv <- update(M1, subset = train.ind.cv)
  M2.cv <- update(M2, subset = train.ind.cv)
  M3.cv <- update(M3, subset = train.ind.cv)
  M4.cv <- update(M4, subset = train.ind.cv)
  M5.cv <- update(M5, subset = train.ind.cv)
  M6.cv <- update(M6, subset = train.ind.cv)
  M7.cv <- update(M7, subset = train.ind.cv)
  M8.cv <- update(M8, subset = train.ind.cv)
  M9.cv <- update(M9, subset = train.ind.cv)
  M10.cv <- update(M10, subset = train.ind.cv)

  M1.res <- pol$e3_bw[-train.ind.cv] - predict(M1.cv, newdata = pol[-train.ind.cv, ])
  M2.res <- pol$e3_bw[-train.ind.cv] - predict(M2.cv, newdata = pol[-train.ind.cv, ])
  M3.res <- pol$e3_bw[-train.ind.cv] - predict(M3.cv, newdata = pol[-train.ind.cv, ])
  M4.res <- pol$e3_bw[-train.ind.cv] - predict(M4.cv, newdata = pol[-train.ind.cv, ])
  M5.res <- pol$e3_bw[-train.ind.cv] - predict(M5.cv, newdata = pol[-train.ind.cv, ])
  M6.res <- pol$e3_bw[-train.ind.cv] - predict(M6.cv, newdata = pol[-train.ind.cv, ])
  M7.res <- pol$e3_bw[-train.ind.cv] - predict(M7.cv, newdata = pol[-train.ind.cv, ])
  M8.res <- pol$e3_bw[-train.ind.cv] - predict(M8.cv, newdata = pol[-train.ind.cv, ])
  M9.res <- pol$e3_bw[-train.ind.cv] - predict(M9.cv, newdata = pol[-train.ind.cv, ])
  M10.res <- pol$e3_bw[-train.ind.cv] - predict(M10.cv, newdata = pol[-train.ind.cv, ])

  mspe1[ii] <- mean(M1.res^2)
  mspe2[ii] <- mean(M2.res^2)
  mspe3[ii] <- mean(M3.res^2)
  mspe4[ii] <- mean(M4.res^2)
  mspe5[ii] <- mean(M5.res^2)
  mspe6[ii] <- mean(M6.res^2)
  mspe7[ii] <- mean(M7.res^2)
  mspe8[ii] <- mean(M8.res^2)
  mspe9[ii] <- mean(M9.res^2)
  mspe10[ii] <- mean(M10.res^2)

}

c1 <- c("Mfwd_AIC", "Mfwd_BIC", "Mbwd_AIC", "Mbwd_BIC", "Mstep_AIC", "Mstep_BIC",
"Mstep_1", "Mstep_2", "Mstep_3", "Mstep_4")
c2 <- c( mean(mspe1), mean(mspe2), mean(mspe3), mean(mspe4),
mean(mspe5), mean(mspe6), mean(mspe7), mean(mspe8), mean(mspe9),
mean(mspe10))
names(c2) <- c1
c2


M1.res <- pollution$e3_bw[test.ind] - predict(Mbwd_AIC, newdata = pollution[test.ind, ])

M2.res <- pollution$e3_bw[test.ind] - predict(Mfwd_AIC, newdata = pollution[test.ind, ])

M3.res <- pollution$e3_bw[test.ind] - predict(Mfwd_BIC, newdata = pollution[test.ind, ])

mean(M1.res^2)
mean(M2.res^2)
mean(M3.res^2)
```