P8451\_final\_ML

Ruixi Li, Junjie Hu, Yuqing Liu, Ruohan Hong

2024-03-26

# Introduction

Recent studies on prenatal exposure and lifestyle effects on birthweight reveal diverse findings. For instance, prenatal exposure to phthalates is linked with a higher risk of preterm birth and potentially lower birthweight (Qian 2020). Exposure to particulate matter during pregnancy has also been associated with varying impacts on birthweight, with some studies showing decreased birthweight when exposure occurs in the second trimester (Bell 2010). Additionally, maternal smoking during pregnancy, particularly in the context of higher stress and anxiety, is significantly associated with low birth weight (Schechter 2020). These findings underscore the complex interplay between environmental factors and maternal lifestyle during pregnancy on fetal development.

Our research question is crucial for several reasons. Understanding how various prenatal lifestyles and chemical exposures affect birthweight is vital for improving public health outcomes. Low birthweight has been linked to numerous adverse health effects, such as increased risk of infant mortality and chronic diseases in later life, like diabetes and heart disease(Negrato 2013). Moreover, identifying these factors can guide public health policies and interventions aimed at minimizing harmful exposures during pregnancy, ultimately enhancing maternal and child health. Furthermore, a data-driven approach can provide objective, quantifiable insights that are essential for developing targeted, evidence-based healthcare strategies.

# Research Question

Our research question is to generate a hypotheses regarding a wide range of prenatal lifestyles and chemical exposures for birth outcome(measured by birthweight) using data-driven methods.

# Load .Rdata file and Data Preparation

rexpose package were introduced here in hope of characterization more easily. But it seemed too rigid to be applied to datasets containing both continuous and categorical variables. So I only deal with continuous exposome with this package.

library(rexposome)  
args(loadExposome)

## function (exposures, description, phenotype, description.famCol = "family",   
## exposures.asFactor = 5, warnings = TRUE)   
## NULL

# Load data using path of where file is stored  
load("exposome.RData")  
  
# modify the dataframes to fit with the data format required by package 'rexposome'.  
## merge covariates and phenotype, phenotype only has birthweight.  
phenotype = phenotype |> select(ID, e3\_bw)  
cov\_phe = inner\_join(covariates, phenotype, by = "ID")  
  
## I am planning on exploring the effect of prenatal exposome towards babies' birth weight. So I just removed all postnatal exposure.  
  
  
### tailor exposome into all continuous prenatal exposome, corresponding with codebook  
pre = codebook |> filter(period == "Pregnancy")  
name = rownames(pre)  
pre\_exp = exposome |> select(ID,any\_of(name))   
pre\_exp\_cont = pre\_exp |> select(where(is.numeric))  
filtered\_name = colnames(pre\_exp\_cont)   
### tailor the codebook(description containing all prenatal exposome description info, including family-each row is an exposure)  
exp\_description <- codebook |> filter(variable\_name %in% filtered\_name)   
### Since codebook's row name was originally the exposure names, I just removed the 'variable name' column   
exp\_description <- exp\_description[,-1]  
### let ID be pre\_exp's rowname and remove the id column  
rownames(pre\_exp) <- pre\_exp\_cont[,1]  
pre\_exp\_cont <- pre\_exp\_cont[,-1]  
### let ID be cov\_phe's rowname and remove the id column  
rownames(cov\_phe) <- cov\_phe[,1]  
cov\_phe <- cov\_phe[,-1]  
  
  
# get the object fit with rexposome  
exp <- loadExposome(  
 exposures = pre\_exp\_cont,   
 description = exp\_description,   
 phenotype = cov\_phe  
)  
# give an overall look at the exposome of interest  
skimr::skim(pre\_exp)

Data summary

|  |  |
| --- | --- |
| Name | pre\_exp |
| Number of rows | 1301 |
| Number of columns | 89 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| factor | 16 |
| numeric | 73 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: factor**

| skim\_variable | n\_missing | complete\_rate | ordered | n\_unique | top\_counts |
| --- | --- | --- | --- | --- | --- |
| e3\_alcpreg\_yn\_None | 0 | 1 | FALSE | 2 | 0: 896, 1: 405 |
| h\_cereal\_preg\_Ter | 0 | 1 | FALSE | 3 | (0,: 531, (9,: 459, (27: 311 |
| h\_dairy\_preg\_Ter | 0 | 1 | FALSE | 3 | (27: 651, (17: 380, (0,: 270 |
| h\_fastfood\_preg\_Ter | 0 | 1 | FALSE | 3 | (0.: 672, (0.: 535, (0,: 94 |
| h\_fish\_preg\_Ter | 0 | 1 | FALSE | 3 | (1.: 490, (4.: 468, (0,: 343 |
| h\_folic\_t1\_None | 0 | 1 | FALSE | 2 | 1: 695, 0: 606 |
| h\_fruit\_preg\_Ter | 0 | 1 | FALSE | 3 | (0.: 922, (18: 373, (0,: 6 |
| h\_legume\_preg\_Ter | 0 | 1 | FALSE | 3 | (2,: 787, (0.: 269, (0,: 245 |
| h\_meat\_preg\_Ter | 0 | 1 | FALSE | 3 | (10: 487, (0,: 427, (6.: 387 |
| h\_pamod\_t3\_None | 0 | 1 | FALSE | 4 | Ver: 594, Oft: 474, Som: 191, Non: 42 |
| h\_pavig\_t3\_None | 0 | 1 | FALSE | 3 | Low: 952, Med: 302, Hig: 47 |
| h\_veg\_preg\_Ter | 0 | 1 | FALSE | 3 | (0,: 539, (8.: 470, (16: 292 |
| hs\_tl\_mdich\_None | 0 | 1 | FALSE | 2 | Und: 1284, Det: 17 |
| h\_blueyn300\_preg\_None | 0 | 1 | FALSE | 2 | 0: 1194, 1: 107 |
| h\_greenyn300\_preg\_None | 0 | 1 | FALSE | 2 | 1: 980, 0: 321 |
| hs\_cotinine\_mcat\_None | 0 | 1 | FALSE | 3 | Non: 759, Smo: 385, SHS: 157 |

**Variable type: numeric**

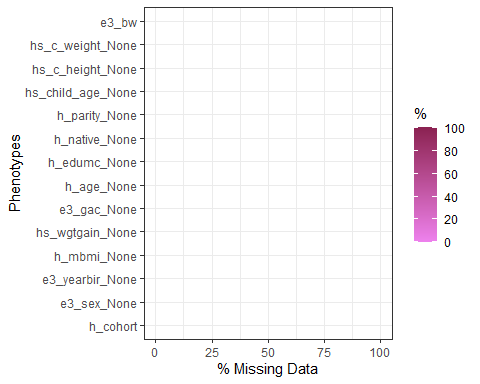
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | 0 | 1 | 651.00 | 375.71 | 1.00 | 326.00 | 651.00 | 976.00 | 1301.00 | ▇▇▇▇▇ |
| h\_abs\_ratio\_preg\_Log | 0 | 1 | 0.39 | 0.40 | -0.48 | 0.10 | 0.30 | 0.73 | 1.71 | ▂▇▅▃▁ |
| h\_no2\_ratio\_preg\_Log | 0 | 1 | 3.00 | 0.47 | 2.10 | 2.67 | 2.96 | 3.30 | 4.52 | ▅▇▆▂▁ |
| h\_pm10\_ratio\_preg\_None | 0 | 1 | 23.50 | 7.94 | 8.07 | 17.54 | 23.02 | 27.68 | 47.70 | ▃▇▅▃▁ |
| h\_pm25\_ratio\_preg\_None | 0 | 1 | 15.03 | 2.70 | 6.96 | 13.29 | 14.88 | 17.00 | 22.24 | ▁▃▇▅▁ |
| h\_accesslines300\_preg\_dic0 | 0 | 1 | 0.20 | 0.40 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | ▇▁▁▁▂ |
| h\_accesspoints300\_preg\_Log | 0 | 1 | 2.67 | 0.82 | 1.27 | 1.96 | 2.88 | 3.35 | 4.53 | ▅▆▇▇▁ |
| h\_builtdens300\_preg\_Sqrt | 0 | 1 | 417.06 | 143.48 | 11.02 | 340.04 | 401.49 | 502.97 | 807.57 | ▁▃▇▃▁ |
| h\_connind300\_preg\_Sqrt | 0 | 1 | 12.74 | 4.67 | 1.89 | 9.98 | 12.93 | 15.90 | 27.28 | ▂▆▇▂▁ |
| h\_fdensity300\_preg\_Log | 0 | 1 | 11.61 | 1.30 | 10.26 | 10.26 | 11.36 | 12.83 | 15.60 | ▇▃▅▂▁ |
| h\_frichness300\_preg\_None | 0 | 1 | 0.07 | 0.08 | 0.00 | 0.00 | 0.04 | 0.12 | 0.42 | ▇▃▁▁▁ |
| h\_landuseshan300\_preg\_None | 0 | 1 | 0.42 | 0.13 | 0.00 | 0.34 | 0.42 | 0.51 | 1.00 | ▁▆▇▁▁ |
| h\_popdens\_preg\_Sqrt | 0 | 1 | 77.02 | 44.62 | 0.00 | 53.79 | 74.98 | 96.21 | 261.50 | ▃▇▂▁▁ |
| h\_walkability\_mean\_preg\_None | 0 | 1 | 0.27 | 0.07 | 0.10 | 0.20 | 0.25 | 0.32 | 0.62 | ▅▇▅▁▁ |
| hs\_as\_m\_Log2 | 0 | 1 | -3.01 | 5.72 | -38.62 | -5.42 | -1.93 | 1.01 | 6.49 | ▁▁▁▇▇ |
| hs\_cd\_m\_Log2 | 0 | 1 | -2.18 | 0.91 | -7.84 | -2.67 | -2.43 | -1.71 | 4.80 | ▁▂▇▁▁ |
| hs\_co\_m\_Log2 | 0 | 1 | -1.69 | 1.19 | -5.18 | -2.51 | -2.01 | -0.55 | 2.50 | ▁▇▅▆▁ |
| hs\_cs\_m\_Log2 | 0 | 1 | 0.48 | 0.57 | -1.16 | 0.07 | 0.40 | 0.81 | 3.45 | ▁▇▃▁▁ |
| hs\_cu\_m\_Log2 | 0 | 1 | 10.40 | 0.26 | 9.04 | 10.25 | 10.44 | 10.54 | 11.17 | ▁▁▃▇▁ |
| hs\_hg\_m\_Log2 | 0 | 1 | 0.57 | 1.40 | -9.02 | -0.31 | 0.58 | 1.57 | 5.44 | ▁▁▂▇▁ |
| hs\_mn\_m\_Log2 | 0 | 1 | 3.54 | 0.47 | 1.66 | 3.29 | 3.57 | 3.81 | 5.45 | ▁▂▇▂▁ |
| hs\_mo\_m\_Log2 | 0 | 1 | -0.69 | 0.59 | -2.72 | -0.98 | -0.73 | -0.40 | 6.13 | ▅▇▁▁▁ |
| hs\_pb\_m\_Log2 | 0 | 1 | 3.21 | 0.70 | 1.22 | 2.62 | 3.19 | 3.81 | 7.55 | ▂▇▃▁▁ |
| h\_humidity\_preg\_None | 0 | 1 | 76.56 | 9.87 | 55.83 | 70.63 | 77.10 | 86.54 | 90.67 | ▃▂▆▅▇ |
| h\_pressure\_preg\_None | 0 | 1 | 991.51 | 13.22 | 974.90 | 980.84 | 983.45 | 1002.32 | 1015.49 | ▇▁▁▅▂ |
| h\_temperature\_preg\_None | 0 | 1 | 11.20 | 4.47 | 3.12 | 8.13 | 10.16 | 13.80 | 22.57 | ▂▇▃▂▂ |
| h\_ndvi100\_preg\_None | 0 | 1 | 0.39 | 0.16 | 0.11 | 0.25 | 0.41 | 0.52 | 0.74 | ▇▆▇▇▂ |
| h\_lden\_cat\_preg\_None | 0 | 1 | 57.47 | 7.81 | 33.92 | 50.00 | 58.63 | 64.36 | 77.40 | ▁▆▇▇▁ |
| hs\_dde\_madj\_Log2 | 0 | 1 | 5.84 | 1.87 | 0.86 | 4.46 | 5.57 | 7.00 | 10.89 | ▁▇▇▅▂ |
| hs\_ddt\_madj\_Log2 | 0 | 1 | 0.87 | 2.43 | -14.14 | -0.26 | 0.68 | 1.51 | 6.56 | ▁▁▁▇▂ |
| hs\_hcb\_madj\_Log2 | 0 | 1 | 2.95 | 1.27 | -9.42 | 2.32 | 2.80 | 3.49 | 7.36 | ▁▁▁▇▂ |
| hs\_pcb118\_madj\_Log2 | 0 | 1 | 1.25 | 0.97 | -1.17 | 0.63 | 1.05 | 1.83 | 7.43 | ▃▇▂▁▁ |
| hs\_pcb138\_madj\_Log2 | 0 | 1 | 2.87 | 1.36 | -10.19 | 1.79 | 2.92 | 3.79 | 8.21 | ▁▁▁▇▁ |
| hs\_pcb153\_madj\_Log2 | 0 | 1 | 3.89 | 1.17 | 1.11 | 2.85 | 3.85 | 4.74 | 9.84 | ▅▇▅▁▁ |
| hs\_pcb170\_madj\_Log2 | 0 | 1 | 1.09 | 1.70 | -2.04 | -0.32 | 0.87 | 2.20 | 7.78 | ▆▇▅▂▁ |
| hs\_pcb180\_madj\_Log2 | 0 | 1 | 2.95 | 1.61 | -10.12 | 2.07 | 2.99 | 4.03 | 9.35 | ▁▁▂▇▁ |
| hs\_sumPCBs5\_madj\_Log2 | 0 | 1 | 4.86 | 1.30 | 2.30 | 4.01 | 4.71 | 5.74 | 9.34 | ▃▇▅▂▁ |
| hs\_dep\_madj\_Log2 | 0 | 1 | 1.70 | 1.59 | -13.41 | 0.99 | 1.66 | 2.67 | 7.59 | ▁▁▁▇▁ |
| hs\_detp\_madj\_Log2 | 0 | 1 | -1.57 | 4.03 | -28.38 | -3.93 | -0.53 | 1.01 | 5.47 | ▁▁▁▅▇ |
| hs\_dmp\_madj\_Log2 | 0 | 1 | 2.24 | 2.91 | -17.14 | 2.01 | 2.80 | 3.76 | 8.33 | ▁▁▁▇▅ |
| hs\_dmtp\_madj\_Log2 | 0 | 1 | 1.61 | 3.15 | -15.33 | 1.07 | 2.23 | 3.49 | 7.78 | ▁▁▂▇▅ |
| hs\_pbde153\_madj\_Log2 | 0 | 1 | -1.74 | 2.85 | -15.00 | -1.88 | -0.95 | -0.03 | 6.43 | ▁▁▁▇▁ |
| hs\_pbde47\_madj\_Log2 | 0 | 1 | -0.78 | 1.42 | -11.58 | -1.76 | -0.97 | 0.12 | 5.12 | ▁▁▃▇▁ |
| hs\_pfhxs\_m\_Log2 | 0 | 1 | -0.98 | 1.43 | -17.83 | -1.73 | -0.93 | -0.16 | 3.76 | ▁▁▁▇▅ |
| hs\_pfna\_m\_Log2 | 0 | 1 | -0.75 | 1.36 | -10.75 | -1.31 | -0.59 | 0.09 | 2.56 | ▁▁▁▇▅ |
| hs\_pfoa\_m\_Log2 | 0 | 1 | 1.05 | 1.04 | -5.48 | 0.41 | 1.20 | 1.74 | 4.98 | ▁▁▅▇▁ |
| hs\_pfos\_m\_Log2 | 0 | 1 | 2.56 | 1.12 | -1.82 | 1.96 | 2.65 | 3.21 | 5.58 | ▁▂▇▇▂ |
| hs\_pfunda\_m\_Log2 | 0 | 1 | -2.66 | 1.50 | -26.21 | -3.21 | -2.48 | -1.71 | -0.04 | ▁▁▁▁▇ |
| hs\_bpa\_madj\_Log2 | 0 | 1 | 1.47 | 1.71 | -11.02 | 0.29 | 1.15 | 2.34 | 6.74 | ▁▁▁▇▂ |
| hs\_bupa\_madj\_Log2 | 0 | 1 | 1.02 | 3.46 | -15.58 | -1.34 | 1.42 | 3.60 | 8.53 | ▁▁▅▇▃ |
| hs\_etpa\_madj\_Log2 | 0 | 1 | 3.33 | 3.58 | -12.12 | 1.24 | 3.28 | 5.13 | 12.73 | ▁▁▆▇▁ |
| hs\_mepa\_madj\_Log2 | 0 | 1 | 7.30 | 2.37 | -0.31 | 5.88 | 7.72 | 8.62 | 15.26 | ▁▃▇▃▁ |
| hs\_oxbe\_madj\_Log2 | 0 | 1 | 3.03 | 3.14 | -10.51 | 0.76 | 2.55 | 4.78 | 13.65 | ▁▂▇▅▁ |
| hs\_prpa\_madj\_Log2 | 0 | 1 | 5.23 | 3.18 | -14.15 | 3.75 | 5.77 | 7.07 | 13.61 | ▁▁▂▇▂ |
| hs\_trcs\_madj\_Log2 | 0 | 1 | 3.43 | 3.44 | -4.81 | 0.55 | 2.66 | 6.59 | 10.69 | ▁▇▇▅▅ |
| hs\_mbzp\_madj\_Log2 | 0 | 1 | 2.98 | 1.60 | -3.74 | 1.86 | 2.89 | 4.10 | 9.30 | ▁▂▇▃▁ |
| hs\_mecpp\_madj\_Log2 | 0 | 1 | 5.03 | 0.98 | 2.43 | 4.33 | 4.85 | 5.63 | 10.41 | ▂▇▃▁▁ |
| hs\_mehhp\_madj\_Log2 | 0 | 1 | 4.16 | 1.25 | -0.46 | 3.46 | 4.07 | 4.79 | 9.92 | ▁▅▇▁▁ |
| hs\_mehp\_madj\_Log2 | 0 | 1 | 2.94 | 1.43 | -7.47 | 1.79 | 3.06 | 3.81 | 8.70 | ▁▁▅▇▁ |
| hs\_meohp\_madj\_Log2 | 0 | 1 | 3.78 | 1.22 | -0.02 | 3.10 | 3.68 | 4.42 | 9.61 | ▁▇▆▁▁ |
| hs\_mep\_madj\_Log2 | 0 | 1 | 7.77 | 1.82 | 3.29 | 6.40 | 7.78 | 8.91 | 14.11 | ▂▇▇▃▁ |
| hs\_mibp\_madj\_Log2 | 0 | 1 | 5.31 | 1.05 | 0.93 | 4.59 | 5.34 | 5.92 | 9.46 | ▁▂▇▂▁ |
| hs\_mnbp\_madj\_Log2 | 0 | 1 | 4.96 | 1.21 | -0.71 | 4.20 | 4.86 | 5.57 | 12.65 | ▁▆▇▁▁ |
| hs\_ohminp\_madj\_Log2 | 0 | 1 | -0.30 | 1.53 | -11.46 | -0.72 | -0.21 | 0.27 | 6.06 | ▁▁▁▇▁ |
| hs\_oxominp\_madj\_Log2 | 0 | 1 | -0.06 | 1.59 | -11.55 | -0.70 | -0.02 | 0.52 | 5.55 | ▁▁▁▇▁ |
| hs\_sumDEHP\_madj\_Log2 | 0 | 1 | 6.01 | 1.08 | 3.21 | 5.23 | 5.88 | 6.70 | 11.69 | ▂▇▃▁▁ |
| e3\_asmokcigd\_p\_None | 0 | 1 | 0.49 | 1.76 | 0.00 | 0.00 | 0.00 | 0.00 | 15.24 | ▇▁▁▁▁ |
| h\_distinvnear1\_preg\_Log | 0 | 1 | -3.15 | 1.61 | -10.02 | -3.98 | -3.00 | -2.26 | 2.79 | ▁▁▇▃▁ |
| h\_trafload\_preg\_pow1over3 | 0 | 1 | 75.54 | 60.16 | 0.35 | 33.65 | 66.61 | 113.08 | 294.27 | ▇▆▃▁▁ |
| h\_trafnear\_preg\_pow1over3 | 0 | 1 | 14.99 | 8.23 | 0.00 | 7.94 | 12.12 | 21.40 | 39.32 | ▁▇▃▂▁ |
| h\_bro\_preg\_Log | 0 | 1 | 1.26 | 2.30 | -2.98 | -0.50 | 1.87 | 2.75 | 4.90 | ▃▆▁▇▃ |
| h\_clf\_preg\_Log | 0 | 1 | 0.96 | 2.94 | -6.91 | -0.50 | 2.08 | 3.18 | 3.83 | ▁▁▂▃▇ |
| h\_thm\_preg\_Log | 0 | 1 | 2.71 | 1.59 | -1.60 | 1.85 | 2.91 | 3.84 | 5.03 | ▁▂▃▅▇ |

# Data Exploration

## 1. Covariates and phenotypes

1. Since this dataset is a combine data from 6 cohorts, I would explore population characteristics(covariates) and outcomes(phenotypes) by cohorts.

plotMissings(exp, set = "phenotypes")



#tableMissings(exp, set = "phenotypes", output = "n")

### categorical variables

cate\_cov\_phe = cov\_phe |> select(where(is.factor))  
  
# Frequency table  
frequency\_tables\_cov\_phe <- cate\_cov\_phe %>% map(~ as.data.frame(table(.x)))  
frequency\_tables\_cov\_phe |>knitr::kable()# name of variables in these freq tables are displayed in the plot below

| .x | Freq |
| --- | --- |
| 1 | 202 |
| 2 | 198 |
| 3 | 224 |
| 4 | 207 |
| 5 | 272 |
| 6 | 198 |

| .x | Freq |
| --- | --- |
| female | 608 |
| male | 693 |

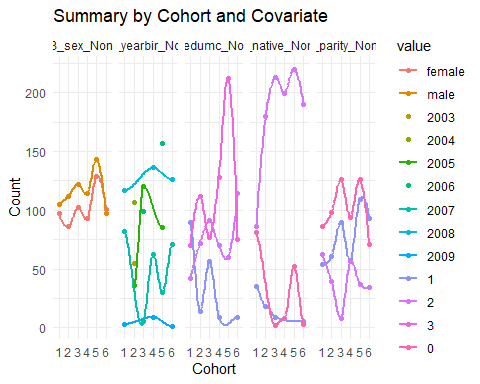
| .x | Freq |
| --- | --- |
| 2003 | 55 |
| 2004 | 107 |
| 2005 | 241 |
| 2006 | 256 |
| 2007 | 250 |
| 2008 | 379 |
| 2009 | 13 |

| .x | Freq |
| --- | --- |
| 1 | 178 |
| 2 | 449 |
| 3 | 674 |

| .x | Freq |
| --- | --- |
| 0 | 146 |
| 1 | 67 |
| 2 | 1088 |

| .x | Freq |
| --- | --- |
| 0 | 601 |
| 1 | 464 |
| 2 | 236 |

# Difference by cohort  
cate\_cov\_phe\_summ = cate\_cov\_phe %>%  
 pivot\_longer(cols = -h\_cohort, names\_to = "covariate", values\_to = "value") %>%  
 group\_by(h\_cohort, covariate,value) %>%  
 summarize(n = n(), .groups = 'drop') %>%  
 ungroup()   
  
cate\_cov\_phe\_summ %>%  
 ggplot(aes(x = h\_cohort, y = n, color = value)) +  
 geom\_point() +  
 geom\_smooth(aes(group=value), method = "loess", se = FALSE) +   
 facet\_grid(. ~ covariate) +  
 theme\_minimal() + # For a cleaner look  
 labs(x = "Cohort", y = "Count", title = "Summary by Cohort and Covariate")

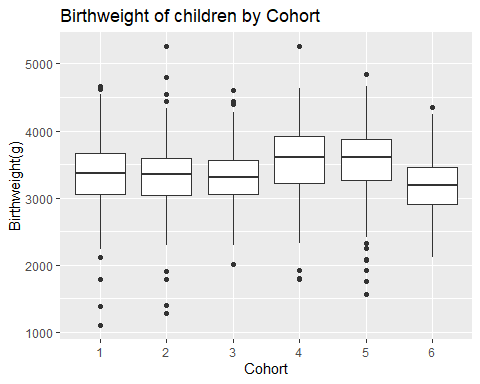


### continuous variables

cont\_cov\_phe = cov\_phe|> select(where(is.numeric))  
  
summary\_table\_cov\_phe <- cont\_cov\_phe %>%  
 summarise(across(where(is.numeric), list(  
 Mean = ~mean(.x, na.rm = TRUE),  
 SD = ~sd(.x, na.rm = TRUE),  
 Median = ~median(.x, na.rm = TRUE),  
 IQR = ~IQR(.x, na.rm = TRUE)  
 ))) |>  
 pivot\_longer(  
 cols = everything(),   
 names\_to = c(".value", "Statistic"),   
 names\_pattern = "(.\*)\_(.\*)"  
 )   
summary\_table\_cov\_phe|> knitr::kable()

| Statistic | h\_mbmi\_None | hs\_wgtgain\_None | e3\_gac\_None | h\_age\_None | hs\_child\_age\_None | hs\_c\_height\_None | hs\_c\_weight\_None | e3\_bw |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mean | 25.030708 | 13.504842 | 39.625673 | 30.795058 | 7.975708 | 1.2914643 | 28.517218 | 3389.3021 |
| SD | 5.183072 | 6.240749 | 1.708398 | 4.857010 | 1.612297 | 0.1050532 | 7.689055 | 509.9945 |
| Median | 24.023809 | 12.000000 | 40.000000 | 31.000000 | 8.032854 | 1.2800000 | 26.900000 | 3398.0000 |
| IQR | 6.077137 | 9.000000 | 2.000000 | 6.418957 | 2.420032 | 0.1560000 | 9.800000 | 640.0000 |

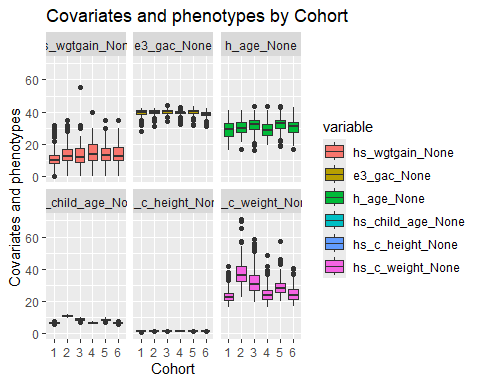
# I specify the continuous variables' names in 'name' vector to add h\_cohort in this dataset(I want to assess if there's difference between cohorts-especially the exposome)  
name\_cont\_cov\_phe = colnames(cont\_cov\_phe)[-1]  
results\_cov\_phe = cov\_phe |> select(h\_cohort, any\_of(name\_cont\_cov\_phe))  
  
results\_summary\_cov\_phe = results\_cov\_phe %>%  
 group\_by(h\_cohort) %>%  
 summarize(across(where(is.numeric),   
 list(mean = ~mean(.x, na.rm = TRUE),   
 std = ~sd(.x, na.rm = TRUE))))   
  
# Since birthweight is too large(also my outcome), I displayed it separately.  
results\_cov\_phe |> ggplot(aes(x=h\_cohort, y=e3\_bw)) + geom\_boxplot() +  
 labs(x = "Cohort", y = "Birthweight(g)", title = "Birthweight of children by Cohort")



results\_cov\_phe\_nobw = results\_cov\_phe |> select(-e3\_bw)  
  
# difference across cohorts  
results\_melted\_cov\_phe = melt(results\_cov\_phe\_nobw)

## Using h\_cohort as id variables

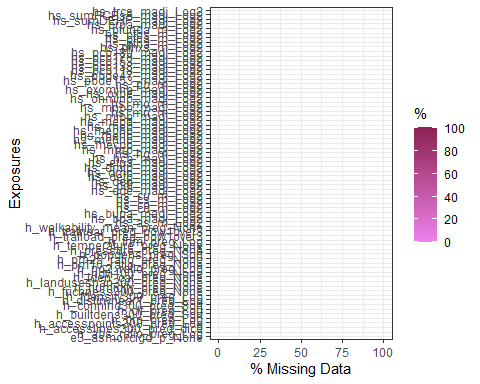
results\_melted\_cov\_phe |>  
 ggplot(aes(x=h\_cohort, y=value, fill=variable)) + geom\_boxplot() + facet\_wrap(~variable)+  
 labs(x = "Cohort", y = "Covariates and phenotypes", title = "Covariates and phenotypes by Cohort")



## 2. Exposome

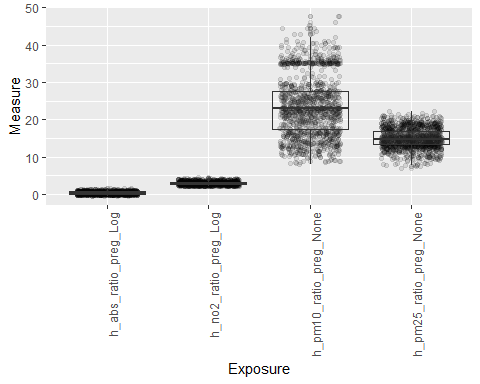
1. Look at the exposome characteristics by family and the correlation between exposome

#tableMissings(exp, set = "exposures", output = "n")  
plotMissings(exp, set = "exposures")

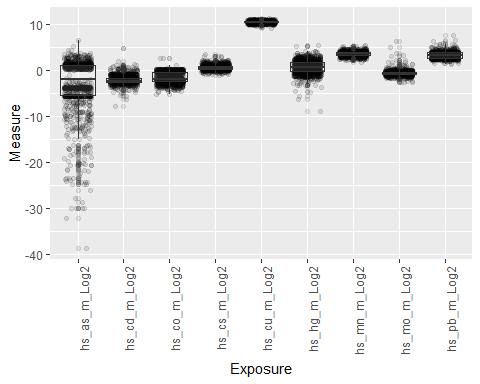
 The current exposome data has no missing in the exposures nor in the phenotypes

### continuous

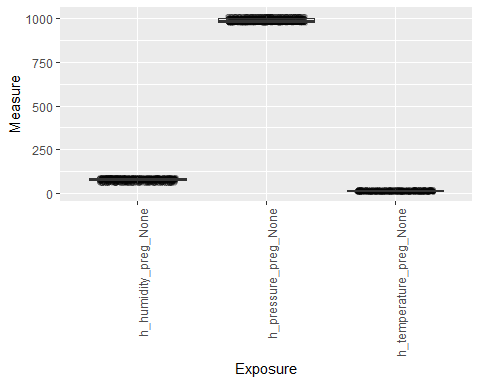
# exposome behavior  
# plotFamily(exp, family = "all") generate error message, I don't know why, the for loop didn't work either. it seemed like the 'build environment' had a mixed falimy  
plotFamily(exp, family = "Air Pollution")



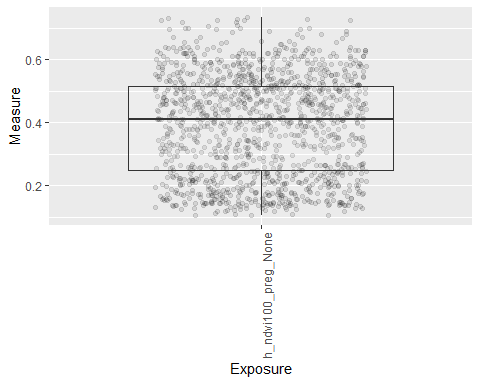
# plotFamily(exp, family = "Built environment")  
plotFamily(exp, family = "Metals")



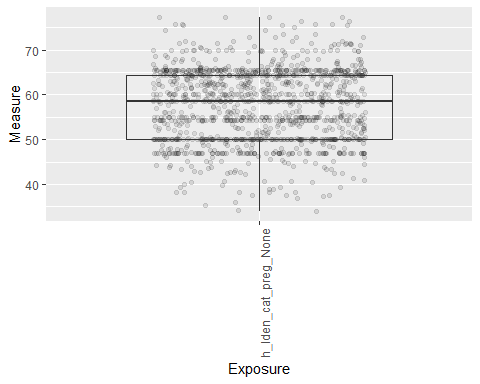
plotFamily(exp, family = "Meteorological")



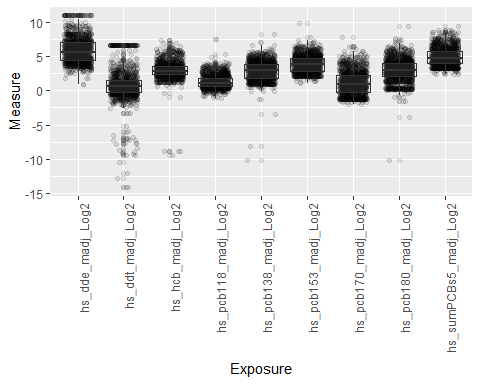
plotFamily(exp, family = "Natural Spaces")



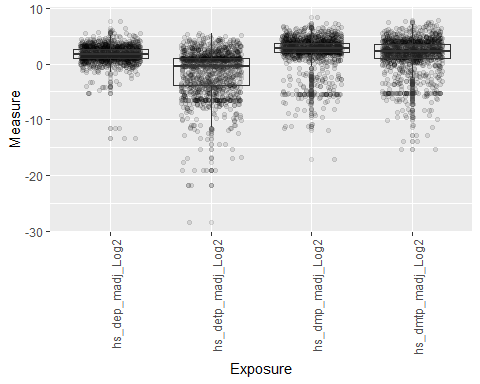
plotFamily(exp, family = "Noise")



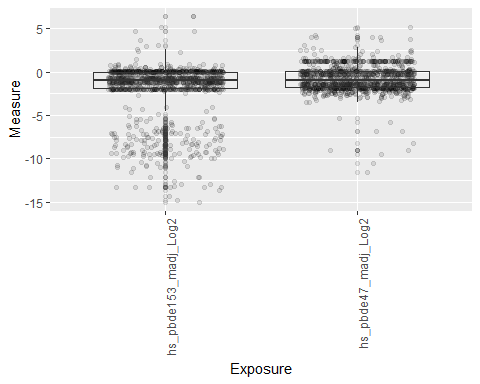
plotFamily(exp, family = "Organochlorines")



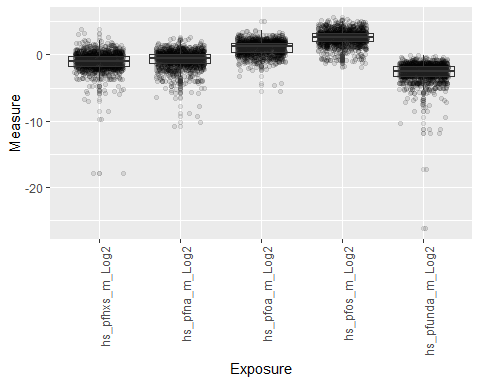
plotFamily(exp, family = "Organophosphate pesticides")



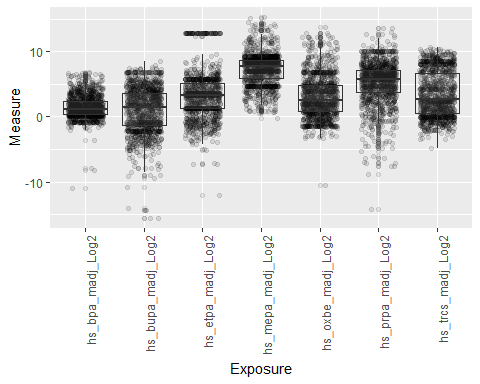
plotFamily(exp, family = "Polybrominated diphenyl ethers (PBDE)")



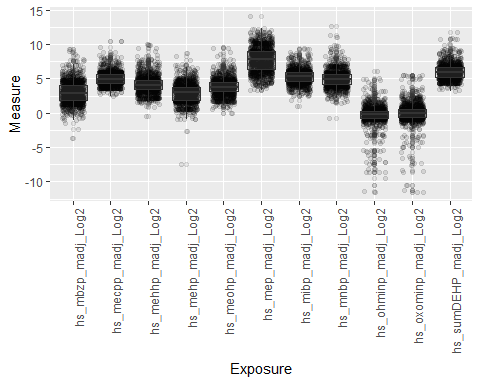
plotFamily(exp, family = "Per- and polyfluoroalkyl substances (PFAS)")



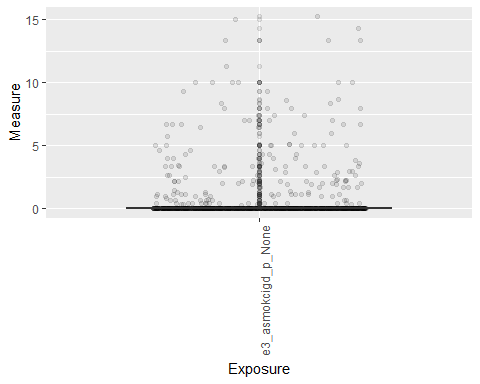
plotFamily(exp, family = "Phenols")



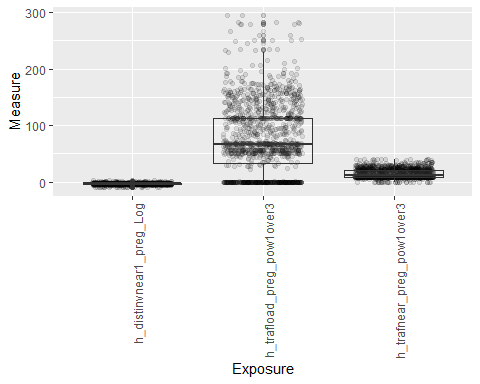
plotFamily(exp, family = "Phthalates")



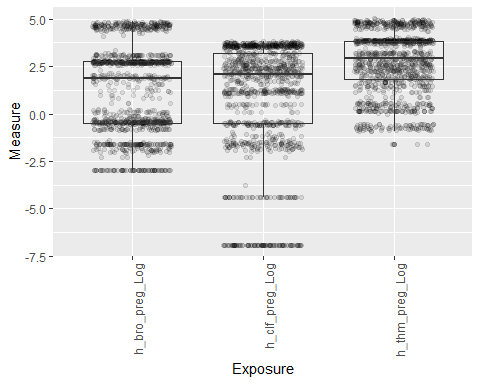
plotFamily(exp, family = "Tobacco Smoke")



plotFamily(exp, family = "Traffic")



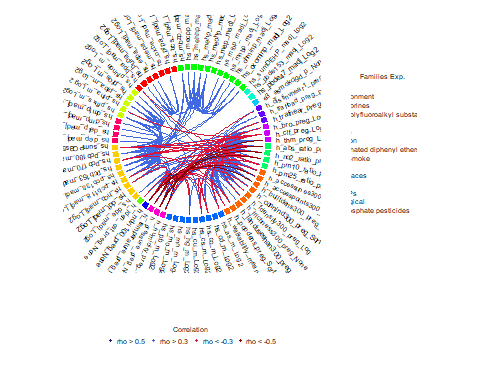
plotFamily(exp, family = "Water DBPs")



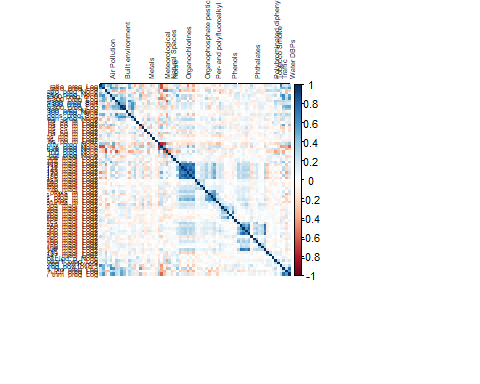
# exposome are not normally distributed  
nm <- normalityTest(exp)  
table(nm$normality)

##   
## FALSE   
## 71

# correlation  
exp\_cr <- correlation(exp, use = "pairwise.complete.obs", method.cor = "pearson")  
plotCorrelation(exp\_cr, type = "circos")



plotCorrelation(exp\_cr, type = "matrix")



# Prepare datasets

# merge features together  
feature = merge(pre\_exp,covariates,by="ID") |> select(-c(hs\_c\_height\_None, hs\_c\_weight\_None, hs\_child\_age\_None))  
outcome = phenotype |> select(ID, e3\_bw)  
studydata = merge(feature,outcome,by="ID")   
  
# keep all interested features, no matter continuous or categorical, in a new dataframe 'rf\_ln' for random forest algorithm and classic linear regression   
rf\_ln = studydata |> select(-ID)  
  
  
# transform all categorical variables into dummy variables and keep all features in a dataframe 'la\_data' for lasso algorithm  
# One-hot encode the categorical variables  
la\_data <- dummyVars("~ .", data = rf\_ln)  
la\_data <- data.frame(predict(la\_data, newdata = rf\_ln))

# PCA to reduce dimensionality(create a new variable)

We reduced the dimensionality by conducting a separate PCA within each of the 19 pre-defined exposure groups, and retained only the first principal component for all of them. In this analysis, a composite index variable was created for each exposure group and then averaged by cohort to compare the levels. This approach significantly simplifies the data, facilitating the examination of key patterns and relationships between exposome exposures and birth weight outcomes across different groups. (<https://www.sciencedirect.com/science/article/pii/S0160412018316295#t0005>)

library(FactoMineR)

# get all family of exposome  
pre\_exp\_grp = codebook |> filter(period=="Pregnancy" & !(family %in% c("Covariates", "Phenotype")))  
exp\_groups = unique(pre\_exp\_grp$family)  
  
# Transform the nominal variables into binary variables using the dummy variable object  
dummies\_pre\_exp <- dummyVars("~ .", data = pre\_exp)  
pre\_exp\_transformed <- data.frame(predict(dummies\_pre\_exp, newdata = pre\_exp))  
  
# Delete the id column of pre\_exp  
pre\_exp\_transformed\_noid = pre\_exp\_transformed |> select(-ID)  
  
# conduct PCA within each family iteratively and get a composite index variable (principal component scores) for each exposure group  
for (family in exp\_groups){  
   
 ori\_comp\_names = codebook |> filter(family==family) |> pull(variable\_name)  
 ori\_comp = pre\_exp\_transformed\_noid |> select(matches(paste0("^", ori\_comp\_names, ".\*")))  
 res.pca <- PCA(ori\_comp, scale.unit = TRUE, ncp = 5, graph = FALSE)  
  
 # Extracting the first principal component scores  
 pc1\_scores <- res.pca$ind$coord[, 1]  
}  
  
  
# Add the PCA score to the whole dataset with cohort info  
studydata$pc1\_score <- pc1\_scores  
  
# Averaging the PCA scores by cohort  
average\_scores\_by\_cohort <- studydata %>%  
 group\_by(h\_cohort) %>%  
 summarise(Average\_pc1\_Score = mean(pc1\_scores, na.rm = TRUE))  
  
average\_scores\_by\_cohort

## # A tibble: 6 × 2  
## h\_cohort Average\_pc1\_Score  
## <fct> <dbl>  
## 1 1 -2.40e-14  
## 2 2 -2.40e-14  
## 3 3 -2.40e-14  
## 4 4 -2.40e-14  
## 5 5 -2.40e-14  
## 6 6 -2.40e-14

**In a nutshell, the final dataset ‘rf\_ln’ has 1 outcome *e3bw*, 10 covariates that were know at birth(excluding hs\_c\_height\_None, hs\_c\_weight\_None, hs\_child\_age\_None) and 88 prenatal exposures.** There’s no missing or duplicate in this dataset. All variables were correctly classified as numeric or factor. **To better application of some algorithms, some transformation of the dataset is needed(e.g. ‘la\_data’)**

EDA findings:

1. Some covariates (e.g. education and nativity of parents) might different across cohorts. We should assess the heterogenitity before we could pool the data from six cohorts. But due to limited time, I just assumed that we can pool them together and continue the further application od machine learning.
2. exposomes vary in their level even though they are from the same family, so we need to do centering and scaling. Using a innovative methods of PCA, we can compare the overall level of exposome to see if there’s cohort effect.
3. More detail on PCA that created a new variable that not involved in our main analysis:

By conducting PCA separately within each family, we aimed to:

* Reduce Dimensionality: We extracted the first principal component (PC) from each family to summarize the key variance within that family. This step reduces the complexity of the dataset while retaining critical information.
* Increase Comparability and Interpretability: Separate PCAs allowed the researchers to better understand and interpret the main drivers of variability within each exposure family. This approach helps in comparing levels of exposure across different cohorts. Through the results, I can conclude that levels of exposure are not different across cohorts.

1. Correlation between exposome were view through two methods of visualization, it seemed like most exposome are independent.

#Partition data for use in demonstration  
set.seed(123)  
train.indices<-createDataPartition(y=rf\_ln$e3\_bw,p=0.7,list=FALSE)  
train.data<-rf\_ln[train.indices, ]  
test.data<-rf\_ln[-train.indices, ]  
  
set.seed(123)  
control <- trainControl(method = "repeatedcv", number = 10, repeats = 5)

# LASSO

la.model<- train(  
 e3\_bw ~.,   
 data = train.data,   
 method = "glmnet",  
 trControl = control,   
 preProc = c("center", "scale"),  
 tuneGrid = expand.grid(alpha = 1,   
 lambda = exp(seq(3, -3, length = 100)))  
 )  
  
coef(la.model$finalModel, la.model$bestTune$lambda)

## 123 x 1 sparse Matrix of class "dgCMatrix"  
## s1  
## (Intercept) 3386.7960526  
## h\_abs\_ratio\_preg\_Log .   
## h\_no2\_ratio\_preg\_Log .   
## h\_pm10\_ratio\_preg\_None -15.8184671  
## h\_pm25\_ratio\_preg\_None .   
## h\_accesslines300\_preg\_dic0 .   
## h\_accesspoints300\_preg\_Log .   
## h\_builtdens300\_preg\_Sqrt .   
## h\_connind300\_preg\_Sqrt .   
## h\_fdensity300\_preg\_Log .   
## h\_frichness300\_preg\_None .   
## h\_landuseshan300\_preg\_None .   
## h\_popdens\_preg\_Sqrt .   
## h\_walkability\_mean\_preg\_None .   
## e3\_alcpreg\_yn\_None1 .   
## h\_cereal\_preg\_Ter(9,27.3] .   
## h\_cereal\_preg\_Ter(27.3,Inf] .   
## h\_dairy\_preg\_Ter(17.1,27.1] 7.9710319  
## h\_dairy\_preg\_Ter(27.1,Inf] .   
## h\_fastfood\_preg\_Ter(0.25,0.83] .   
## h\_fastfood\_preg\_Ter(0.83,Inf] -1.5167790  
## h\_fish\_preg\_Ter(1.9,4.1] .   
## h\_fish\_preg\_Ter(4.1,Inf] .   
## h\_folic\_t1\_None1 -13.5643674  
## h\_fruit\_preg\_Ter(0.6,18.2] .   
## h\_fruit\_preg\_Ter(18.2,Inf] .   
## h\_legume\_preg\_Ter(0.5,2] .   
## h\_legume\_preg\_Ter(2,Inf] .   
## h\_meat\_preg\_Ter(6.5,10] .   
## h\_meat\_preg\_Ter(10,Inf] .   
## h\_pamod\_t3\_NoneOften 3.9781030  
## h\_pamod\_t3\_NoneSometimes -0.1056258  
## h\_pamod\_t3\_NoneVery Often .   
## h\_pavig\_t3\_NoneLow -5.8999337  
## h\_pavig\_t3\_NoneMedium .   
## h\_veg\_preg\_Ter(8.8,16.5] .   
## h\_veg\_preg\_Ter(16.5,Inf] -3.8750667  
## hs\_as\_m\_Log2 -8.8144787  
## hs\_cd\_m\_Log2 -4.6852357  
## hs\_co\_m\_Log2 .   
## hs\_cs\_m\_Log2 -1.3928696  
## hs\_cu\_m\_Log2 11.8108218  
## hs\_hg\_m\_Log2 .   
## hs\_mn\_m\_Log2 .   
## hs\_mo\_m\_Log2 .   
## hs\_pb\_m\_Log2 .   
## hs\_tl\_mdich\_NoneUndetected 5.2662545  
## h\_humidity\_preg\_None .   
## h\_pressure\_preg\_None .   
## h\_temperature\_preg\_None .   
## h\_blueyn300\_preg\_None1 .   
## h\_greenyn300\_preg\_None1 .   
## h\_ndvi100\_preg\_None 14.3946448  
## h\_lden\_cat\_preg\_None .   
## hs\_dde\_madj\_Log2 .   
## hs\_ddt\_madj\_Log2 .   
## hs\_hcb\_madj\_Log2 .   
## hs\_pcb118\_madj\_Log2 .   
## hs\_pcb138\_madj\_Log2 .   
## hs\_pcb153\_madj\_Log2 .   
## hs\_pcb170\_madj\_Log2 .   
## hs\_pcb180\_madj\_Log2 .   
## hs\_sumPCBs5\_madj\_Log2 .   
## hs\_dep\_madj\_Log2 -8.7728556  
## hs\_detp\_madj\_Log2 .   
## hs\_dmp\_madj\_Log2 .   
## hs\_dmtp\_madj\_Log2 17.6705884  
## hs\_pbde153\_madj\_Log2 .   
## hs\_pbde47\_madj\_Log2 11.3946652  
## hs\_pfhxs\_m\_Log2 .   
## hs\_pfna\_m\_Log2 -10.4312255  
## hs\_pfoa\_m\_Log2 -7.6724629  
## hs\_pfos\_m\_Log2 .   
## hs\_pfunda\_m\_Log2 .   
## hs\_bpa\_madj\_Log2 .   
## hs\_bupa\_madj\_Log2 .   
## hs\_etpa\_madj\_Log2 18.0331278  
## hs\_mepa\_madj\_Log2 -10.8937394  
## hs\_oxbe\_madj\_Log2 6.7639741  
## hs\_prpa\_madj\_Log2 .   
## hs\_trcs\_madj\_Log2 .   
## hs\_mbzp\_madj\_Log2 .   
## hs\_mecpp\_madj\_Log2 .   
## hs\_mehhp\_madj\_Log2 .   
## hs\_mehp\_madj\_Log2 .   
## hs\_meohp\_madj\_Log2 .   
## hs\_mep\_madj\_Log2 .   
## hs\_mibp\_madj\_Log2 8.6389188  
## hs\_mnbp\_madj\_Log2 .   
## hs\_ohminp\_madj\_Log2 .   
## hs\_oxominp\_madj\_Log2 .   
## hs\_sumDEHP\_madj\_Log2 .   
## e3\_asmokcigd\_p\_None -12.9413098  
## hs\_cotinine\_mcat\_NoneSHS smokers .   
## hs\_cotinine\_mcat\_NoneSmokers -1.1695623  
## h\_distinvnear1\_preg\_Log .   
## h\_trafload\_preg\_pow1over3 .   
## h\_trafnear\_preg\_pow1over3 -2.4134593  
## h\_bro\_preg\_Log -42.9311999  
## h\_clf\_preg\_Log .   
## h\_thm\_preg\_Log .   
## h\_cohort2 .   
## h\_cohort3 -6.5180591  
## h\_cohort4 .   
## h\_cohort5 10.4907085  
## h\_cohort6 .   
## e3\_sex\_Nonemale 73.3793752  
## e3\_yearbir\_None2004 .   
## e3\_yearbir\_None2005 .   
## e3\_yearbir\_None2006 .   
## e3\_yearbir\_None2007 .   
## e3\_yearbir\_None2008 .   
## e3\_yearbir\_None2009 .   
## h\_mbmi\_None 41.8537906  
## hs\_wgtgain\_None 43.7369604  
## e3\_gac\_None 233.4141577  
## h\_age\_None .   
## h\_edumc\_None2 .   
## h\_edumc\_None3 13.9204259  
## h\_native\_None1 .   
## h\_native\_None2 17.5434887  
## h\_parity\_None1 28.2029278  
## h\_parity\_None2 14.0549894

print(la.model$bestTune)

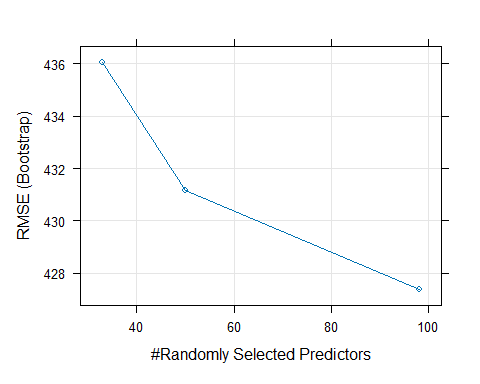
## alpha lambda  
## 93 1 13.14127

print(la.model$results$RMSE[which.min(la.model$results$RMSE)])

## [1] 405.021

# Random Forest

# For random forest with lots of features, mtry = p(the number of all features)/3 tend to be a good default mtry for regression. mtry = sqrt(p) is a good default for classification. (https://scholarworks.utep.edu/cgi/viewcontent.cgi?article=4251&context=open\_etd)  
  
feat.count<-c(ncol(rf\_ln)/3, ncol(rf\_ln)/2, ncol(rf\_ln)-1)  
  
grid.rf<-expand.grid(mtry=round(feat.count))  
  
tree.num<-seq(100,500, by=200)  
  
results.trees<-list()  
  
for (ntree in tree.num){  
 set.seed(123)  
 rf.model<-train(e3\_bw ~ .,   
 data=train.data,   
 method="rf",   
 metric="RMSE",   
 tuneGrid=grid.rf,   
 importance=TRUE,   
 ntree=ntree)  
index<-toString(ntree)  
results.trees[[index]]<-rf.model$results  
}  
plot(rf.model)



rf.model$results

## mtry RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 33 436.0520 0.2631721 341.6810 15.13138 0.04502143 12.92401  
## 2 50 431.1423 0.2742838 338.6898 15.30234 0.04840918 13.02320  
## 3 98 427.3793 0.2821984 336.2643 16.02849 0.04725333 13.34851

output.rf<-bind\_rows(results.trees, .id = "ntrees")  
best.tune<-output.rf[which.max(output.rf[,"mtry"]==50),]  
best.tune$mtry

## [1] 50

mtry.grid<-expand.grid(.mtry=best.tune$mtry)  
set.seed(123)  
 rf.model.bt<-train(  
 e3\_bw~.,   
 data=train.data,   
 method="rf",   
 trControl=control,   
 metric="RMSE",   
 tuneGrid=mtry.grid,   
 importance=TRUE,  
 ntree=as.numeric(best.tune$ntrees))  
  
   
  
  
rf.model.bt$results

## mtry RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 50 422.1253 0.2907465 332.1946 28.64586 0.07488186 21.35613

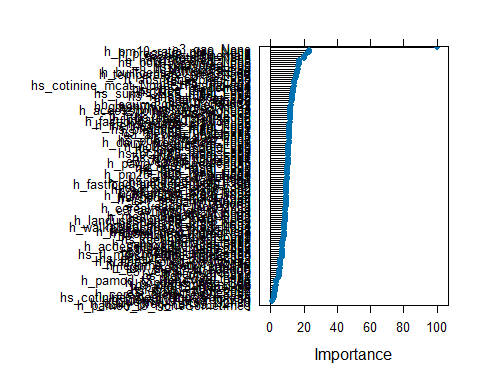
rf.model.bt$finalModel

##   
## Call:  
## randomForest(x = x, y = y, ntree = ..2, mtry = param$mtry, importance = TRUE)   
## Type of random forest: regression  
## Number of trees: 100  
## No. of variables tried at each split: 50  
##   
## Mean of squared residuals: 181604  
## % Var explained: 26.95

varImp(rf.model.bt)

## rf variable importance  
##   
## only 20 most important variables shown (out of 122)  
##   
## Overall  
## e3\_gac\_None 100.00  
## h\_pm10\_ratio\_preg\_None 23.55  
## h\_pressure\_preg\_None 22.97  
## e3\_sex\_Nonemale 21.63  
## hs\_wgtgain\_None 20.50  
## h\_ndvi100\_preg\_None 19.96  
## hs\_pcb153\_madj\_Log2 17.51  
## h\_bro\_preg\_Log 17.31  
## hs\_pfna\_m\_Log2 17.07  
## h\_thm\_preg\_Log 16.77  
## h\_mbmi\_None 16.76  
## h\_builtdens300\_preg\_Sqrt 15.99  
## h\_temperature\_preg\_None 15.86  
## hs\_cu\_m\_Log2 15.72  
## h\_abs\_ratio\_preg\_Log 15.63  
## hs\_as\_m\_Log2 15.05  
## hs\_mo\_m\_Log2 14.03  
## hs\_cotinine\_mcat\_NoneSHS smokers 14.01  
## h\_cohort4 13.96  
## hs\_bupa\_madj\_Log2 13.50

plot(varImp(rf.model.bt))



Although when mtry is close to the number of all features(p), RMSE is the lowest. But it increases the risk of overfitting. What’s more, values of mtry that are close to the total number of variables in the model may weaken the forest by making the individual decision trees more correlated; when the decision trees consider similar sets of variables to split on, they are more likely to be similar, even if each is fit to a different bootstrapped data set. Ensemble models usually strive for independence of their members, as that improves predictive ability. We can also see from the plot that the decreasing trend of RMSE flattened after 50. I would choose 50 as my final mtry.

# Linear regression

# Initialize a model with all predictors  
intercept.model <- lm(e3\_bw ~ ., data = rf\_ln)  
  
# Both-direction stepwise regression  
ln.model <- step(intercept.model, direction = "both", trace = 0)  
  
summary(ln.model)

##   
## Call:  
## lm(formula = e3\_bw ~ h\_fdensity300\_preg\_Log + h\_frichness300\_preg\_None +   
## h\_walkability\_mean\_preg\_None + h\_dairy\_preg\_Ter + h\_folic\_t1\_None +   
## h\_meat\_preg\_Ter + h\_pamod\_t3\_None + hs\_cs\_m\_Log2 + h\_temperature\_preg\_None +   
## hs\_dde\_madj\_Log2 + hs\_pcb170\_madj\_Log2 + hs\_dep\_madj\_Log2 +   
## hs\_dmtp\_madj\_Log2 + hs\_pfos\_m\_Log2 + hs\_bupa\_madj\_Log2 +   
## hs\_mepa\_madj\_Log2 + hs\_meohp\_madj\_Log2 + hs\_mibp\_madj\_Log2 +   
## hs\_mnbp\_madj\_Log2 + e3\_asmokcigd\_p\_None + h\_bro\_preg\_Log +   
## h\_cohort + e3\_sex\_None + h\_mbmi\_None + hs\_wgtgain\_None +   
## e3\_gac\_None + h\_edumc\_None + h\_parity\_None, data = rf\_ln)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1536.46 -243.54 -3.73 236.60 1469.24   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -2.942e+03 3.924e+02 -7.496 1.23e-13 \*\*\*  
## h\_fdensity300\_preg\_Log -3.433e+01 2.189e+01 -1.568 0.117102   
## h\_frichness300\_preg\_None 7.421e+02 2.853e+02 2.601 0.009406 \*\*   
## h\_walkability\_mean\_preg\_None -3.555e+02 2.228e+02 -1.596 0.110842   
## h\_dairy\_preg\_Ter(17.1,27.1] 5.479e+01 3.249e+01 1.687 0.091931 .   
## h\_dairy\_preg\_Ter(27.1,Inf] 2.672e-02 3.257e+01 0.001 0.999345   
## h\_folic\_t1\_None1 -4.417e+01 2.622e+01 -1.684 0.092401 .   
## h\_meat\_preg\_Ter(6.5,10] 5.863e+00 2.835e+01 0.207 0.836178   
## h\_meat\_preg\_Ter(10,Inf] -4.462e+01 2.710e+01 -1.647 0.099888 .   
## h\_pamod\_t3\_NoneOften 1.477e+02 6.360e+01 2.322 0.020378 \*   
## h\_pamod\_t3\_NoneSometimes 8.696e+01 6.748e+01 1.289 0.197793   
## h\_pamod\_t3\_NoneVery Often 1.226e+02 6.299e+01 1.946 0.051838 .   
## hs\_cs\_m\_Log2 -6.307e+01 2.439e+01 -2.586 0.009819 \*\*   
## h\_temperature\_preg\_None 1.159e+01 4.729e+00 2.452 0.014360 \*   
## hs\_dde\_madj\_Log2 -1.015e+01 6.608e+00 -1.536 0.124692   
## hs\_pcb170\_madj\_Log2 1.715e+01 1.177e+01 1.457 0.145313   
## hs\_dep\_madj\_Log2 -1.897e+01 7.304e+00 -2.598 0.009495 \*\*   
## hs\_dmtp\_madj\_Log2 1.341e+01 3.790e+00 3.538 0.000418 \*\*\*  
## hs\_pfos\_m\_Log2 -2.684e+01 1.223e+01 -2.194 0.028417 \*   
## hs\_bupa\_madj\_Log2 6.836e+00 3.704e+00 1.846 0.065190 .   
## hs\_mepa\_madj\_Log2 -1.423e+01 5.016e+00 -2.837 0.004622 \*\*   
## hs\_meohp\_madj\_Log2 1.466e+01 1.042e+01 1.406 0.159843   
## hs\_mibp\_madj\_Log2 2.147e+01 1.209e+01 1.776 0.076053 .   
## hs\_mnbp\_madj\_Log2 -1.626e+01 1.074e+01 -1.514 0.130327   
## e3\_asmokcigd\_p\_None -1.639e+01 6.518e+00 -2.514 0.012051 \*   
## h\_bro\_preg\_Log -1.646e+01 1.108e+01 -1.486 0.137623   
## h\_cohort2 -1.527e+02 6.777e+01 -2.253 0.024410 \*   
## h\_cohort3 -9.812e+01 5.638e+01 -1.740 0.082051 .   
## h\_cohort4 7.476e+01 7.018e+01 1.065 0.286988   
## h\_cohort5 9.766e+01 6.318e+01 1.546 0.122430   
## h\_cohort6 -1.699e+02 8.583e+01 -1.979 0.047994 \*   
## e3\_sex\_Nonemale 1.633e+02 2.212e+01 7.384 2.79e-13 \*\*\*  
## h\_mbmi\_None 1.116e+01 2.359e+00 4.730 2.49e-06 \*\*\*  
## hs\_wgtgain\_None 8.431e+00 1.810e+00 4.658 3.54e-06 \*\*\*  
## e3\_gac\_None 1.554e+02 6.811e+00 22.818 < 2e-16 \*\*\*  
## h\_edumc\_None2 8.394e+01 3.912e+01 2.146 0.032099 \*   
## h\_edumc\_None3 9.677e+01 3.918e+01 2.470 0.013656 \*   
## h\_parity\_None1 1.023e+02 2.485e+01 4.117 4.08e-05 \*\*\*  
## h\_parity\_None2 7.245e+01 3.184e+01 2.276 0.023023 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 390.6 on 1262 degrees of freedom  
## Multiple R-squared: 0.4306, Adjusted R-squared: 0.4134   
## F-statistic: 25.11 on 38 and 1262 DF, p-value: < 2.2e-16

set.seed(123)  
  
  
linear.model <- train(  
 e3\_bw ~ h\_fdensity300\_preg\_Log + h\_frichness300\_preg\_None +   
 h\_walkability\_mean\_preg\_None + h\_dairy\_preg\_Ter + h\_folic\_t1\_None +   
 h\_meat\_preg\_Ter + h\_pamod\_t3\_None + hs\_cs\_m\_Log2 + h\_temperature\_preg\_None +   
 hs\_dde\_madj\_Log2 + hs\_pcb170\_madj\_Log2 + hs\_dep\_madj\_Log2 +   
 hs\_dmtp\_madj\_Log2 + hs\_pfos\_m\_Log2 + hs\_bupa\_madj\_Log2 +   
 hs\_mepa\_madj\_Log2 + hs\_meohp\_madj\_Log2 + hs\_mibp\_madj\_Log2 +   
 hs\_mnbp\_madj\_Log2 + e3\_asmokcigd\_p\_None + h\_bro\_preg\_Log +   
 h\_cohort + e3\_sex\_None + h\_mbmi\_None + hs\_wgtgain\_None +   
 e3\_gac\_None + h\_edumc\_None + h\_parity\_None,  
 data = train.data,  
 method = "lm",  
 trControl = control,  
 preProcess = c("center", "scale")  
)  
  
linear.model$results

## intercept RMSE Rsquared MAE RMSESD RsquaredSD MAESD  
## 1 TRUE 400.9836 0.3570733 316.0789 29.73559 0.0784362 23.89301

# Model Comparasion

The RMSE (Root Mean Squared Error) values for the LASSO, Random Forest, and Linear Regression algorithms were 405.02, 422.12, and 400.98 respectively. RMSE measures the average magnitude of the errors between predicted and actual values, with a lower RMSE indicating better model performance. Among the three, Linear Regression achieved the lowest RMSE, suggesting its superior predictive accuracy for birth weight. This superiority is further supported by comparing R-squared values, which represent the proportion of the variance in the dependent variable that is predictable from the independent variable(s). The R-squared values for Random Forest and Linear Regression were 0.2907 and 0.3571 respectively. Since a higher R-squared value indicates better model performance, the higher R-squared of the Linear Regression model confirms its effectiveness in explaining the variation in birth weight outcomes. Based on these evaluations of RMSE and R-squared values along with the features they selected, Linear Regression is the optimal model among the three for generating a hypothesis for the birthweight.

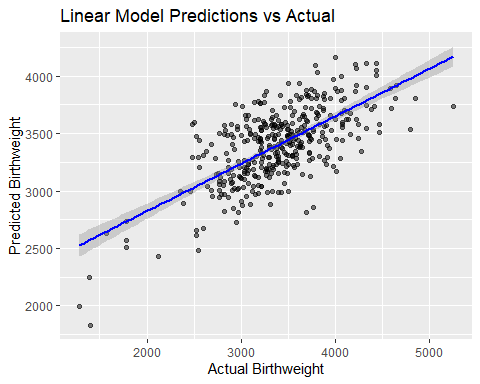
predictions <- predict(linear.model, test.data)

results <- postResample(predictions, test.data$e3\_bw)  
results

## RMSE Rsquared MAE   
## 395.6557533 0.4575206 303.4334489

ggplot(data = test.data, aes(x = e3\_bw, y = predictions)) +  
 geom\_point(alpha = 0.5) +  
 geom\_smooth(method = "lm", col = "blue") +  
 labs(x = "Actual Birthweight", y = "Predicted Birthweight", title = "Linear Model Predictions vs Actual")

## `geom\_smooth()` using formula = 'y ~ x'



Interpretation of the final model: The fact that the majority of points are clustered around the line suggests that there is a moderate to strong positive linear relationship between the actual and predicted birthweights. There is some variability in the predictions, as indicated by the scatter of points around the line. Notably, the model seems to underpredict for lower actual birthweights and overpredict for higher actual birthweights, as evidenced by the points below the line on the left and above the line on the right. Linear regression also shrink more than 100 features to 28 features, which considerably narrow down the essential components for the hypothesis. The performance for test data also indicated a good generalization for data other than training data, which is measured by testing RMSE(395.66<testing RMSE=400.98) However, the model showed a low performance, meaning that pre-natal exposures and lifestyles might not be good features to generate hypothesis for birthweight. Also, to further generate causal hypothesis, other applications including causal discovery through super learner might be used.

# Limitation

**Treatment of Categorical Variables**

In our analysis, we utilized Lasso regression. However, a notable limitation of Lasso regression is that lasso regression doesn’t work well with categorical variables in terms of both feature selection and prediction accuracy(performance). We just converted categorical variables into factors and treated them as ordinal variables, which can be approximately analyzed as continuous variables. However, some categorical variables were nominal variables, our analysis ignored the nature of the data and might lead to inaccurate analytical results. However, group lasso may be a good alternative for robust feature selection although at the cost of prediction accuracy(Huang 2024). We can first transform all categorical variables into dummy variables using the one-hot encoding method, followed by a group lasso model, which considers the grouping effect.In this case, dummy variables derived from the same categorical variable are grouped, ensuring that either all or none of the dummy variables in a group were selected. This method helps preserve the integrity of the categorical data structure, maintaining the complete set of information provided by each categorical variable while still benefiting from the regularization and feature selection properties of Lasso.This approach increases the dimensionality of the data,which will decrease the performance of this model. But our research question is hypothesis generation, robustness of feature selection is more important.

**Model Interpretability** While our final model, linear regression, provides interpretability, it can potentially oversimplify the association between prenatal factors and birth weight, especially if there are interactions between variables or the association is not linear. Hence, to address this problem, a more flexible model should be considered to apply, such as random forest and generalized additive models (GAMs). However, it’s crucial to balance model complexicity with interpretability.

**Causal Inference** The final model, linear regression, is only eligible to identify associations between prenatal factors and birth weight, which fail to determine causal inference. An alternative method to strengthen casual relationship between prenatal factors and birth weight is to apply advanced casual inference methods, such as propensity score matching, or conducting a longitudinal research. These methods enable more robust and reliable conclusions regarding causal relationships between variables.

**Ethical Concerns**

Data privacy and potential stigamatized certain populations. Exposome research involves collecting extensive personal and environmental data from participants, including personal sensitive information in lifestyle choices, environmental exposures, and genetic data. Ensuring the privacy and confidentiality of this data is crucial. Therefore, the handling, storage, and processing of such data should comply with legal regulations and ethical standards to protect participants’ privacy. Moreover, findings from studies investigating the impact of environmental and lifestyle factors on birth weight could potentially lead to stigmatization of individuals or groups. For instance, if particular behaviors or environmental exposures linked to lower birth weights are prevalent in specific population, these groups might face stigma or discrimination. Therefore, it’s important to handle such information sensitively, emphasizing that these findings are not indicative of individual or collective fault and should be used to inform better health policies and interventions. Additionally, there may be bias related to the selection of study participants. For instance, our study participants have a higher likelihood to have access to smart phones and internet, hence it may not accurately represent the entire population of the studied area. Consequently, individuals without these resources—who were not included in the study—might become further marginalized and underrepresented in research that uses machine learning approaches.

# Reference

Qian, Y., Shao, H., Ying, X., Huang, W., & Hua, Y. (2020). The Endocrine Disruption of Prenatal Phthalate Exposure in Mother and Offspring. Frontiers in public health, 8, 366. <https://doi.org/10.3389/fpubh.2020.00366>

Bell, M. L., Belanger, K., Ebisu, K., Gent, J. F., Lee, H. J., Koutrakis, P., & Leaderer, B. P. (2010). Prenatal exposure to fine particulate matter and birth weight: variations by particulate constituents and sources. Epidemiology (Cambridge, Mass.), 21(6), 884–891. <https://doi.org/10.1097/EDE.0b013e3181f2f405>

Schechter, J., Do, E. K., Zhang, J. J., Hoyo, C., Murphy, S. K., Kollins, S. H., & Fuemmeler, B. (2020). Effect of Prenatal Smoke Exposure on Birth Weight: The Moderating Role of Maternal Depressive Symptoms. Nicotine & tobacco research : official journal of the Society for Research on Nicotine and Tobacco, 22(1), 40–47. <https://doi.org/10.1093/ntr/nty267>

Negrato, C. A., & Gomes, M. B. (2013). Low birth weight: causes and consequences. Diabetology & metabolic syndrome, 5, 49. <https://doi.org/10.1186/1758-5996-5-49> (Retraction published Diabetol Metab Syndr. 2014;6:60)

Huang, Y., Tibbe, T., Tang, A., & Montoya, A. (2024). Lasso and Group Lasso with Categorical Predictors: Impact of Coding Strategy on Variable Selection and Prediction. Journal of Behavioral Data Science, 3(2), 15-42. <https://doi.org/10.35566/jbds/v3n2/montoya>

# Optional: One-hot coding and Group LASSO

Attempt of applying group lasso. Not sure if it’s correct, but it shrink

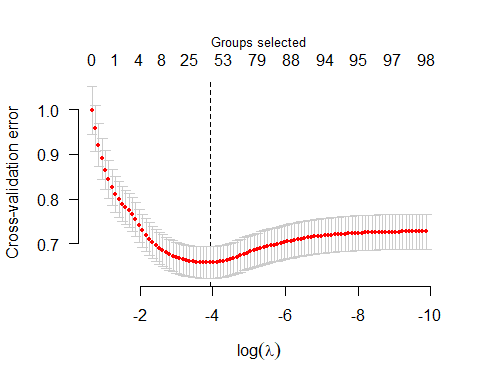
# Partitioning the data into training and testing sets  
set.seed(123)  
train.indices <- createDataPartition(la\_data$e3\_bw, p = 0.7, list = FALSE)  
train.data <- la\_data[train.indices, ]  
test.data <- la\_data[-train.indices, ]  
  
# Feature scaling (center and scale)  
train.data.scaled <- scale(train.data)  
test.data.scaled <- scale(test.data)  
  
  
# Define groups-ensure dummies from one categorical variable is considered as a group  
la\_data\_grp = la\_data[, -which(names(train.data) == "e3\_bw")]  
  
group\_labels <- gsub("\\..\*", "", colnames(la\_data\_grp))  
group\_numbers <- as.numeric(factor(group\_labels))  
  
  
library(grpreg)

## Warning: package 'grpreg' was built under R version 4.3.3

##   
## Attaching package: 'grpreg'

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

# Prepare the features and outcome  
x.train <- as.data.frame(train.data.scaled[, -which(names(train.data) == "e3\_bw")])  
y.train <- as.data.frame(train.data.scaled)$e3\_bw  
x.test <- as.matrix(test.data.scaled[, -which(names(test.data) == "e3\_bw")])  
y.test <- as.data.frame(test.data.scaled)$e3\_bw  
# To perform cross-validation to find the optimal lambda  
cv\_model <- cv.grpreg(x.train, y.train, group = group\_numbers, penalty = "grLasso")  
plot(cv\_model)



optimal\_lambda <- 55 # reason were written below  
# Retrieve cross-validated mean squared errors for the optimal lambda  
cv\_error <- sqrt(min(cv\_model$cve))  
cv\_error

## [1] 0.8113468

la.model.bt = grpreg(x.train, y.train, group = group\_numbers, penalty = "grLasso", lambda = optimal\_lambda)  
  
coefficients = coef(la.model.bt)

Given my aim of generating hypothesis, I want to make my model as simple as possible (fewer predictors) while still having good predictive power (low cross-validation error). The larger  represents larger amount of shrinkage in the lasso regression.Thus, I chose log()= 4 as my final .

A Weird thing is that nearly coefficients for all features were 0, meaning that group lasso shrunk all features. cross-validation error were used in this package to evaluate model performance, but I am not sure weather the cross-validation error is RMSE.(not likely)