```r
library(dplyr)
library(ggplot2)
```

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

```r
head(birth)
nrow(birth)
```

```r
# 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)
```

```r
nrow(clean_birth)
```

```r
# Feature engineering

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

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

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

# compute percentage weight gain
```

```r
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 PRIORDEAD
clean_birth$PRIORTERM <- ifelse(clean_birth$PRIORTERM > 0, TRUE, FALSE)

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

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

```r
# Reduce the dimensionality of the dataset

# drop columns where >99% entries are the same
clean_birth <- clean_birth %>% 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 %>% select(!c(WTGAIN, PWgt_R, DWgt_R, DOB_MM,
                                   DOB_WK, DOB_TT, DOB_MM, DLMP_YY,
                                   DLMP_MM, PAY, MHISPX, MRACE15,
                                   MRACE31, MRACEIMP, FHISPX, FRACE15,
                                   FRACE31, RF_CESARN, ILOP_R, ILP_R, ILLB_R))
```

```r
head(clean_birth)
```

```r
clean_birth %>% count(PRIORTERM)
```

```r
lapply(clean_birth, function(x) sum(x == 99))
```

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

```r
# subsample datasets

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

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