STAT 331 Final Project

Deep Antala, Thushar Ishwanthlal, Vanessa Li, Harry Qu

Appendix

```
load("C:/Users/deepa/Desktop/Spring 2021/STAT 331/pollution.Rdata")
# splitting training set and test set
training_set <- pollution[c(1:600), ]
test_set <- pollution[-c(1:600), ]
```

```
# ensuring numerical factor levels are treated as factors
pollution$e3_alcpreg_yn_None <- as.factor(pollution$e3_alcpreg_yn_None)
pollution$h_folic_t1_None <- as.factor(pollution$h_folic_t1_None)
pollution$e3_yearbir_None <- as.factor(pollution$e3_yearbir_None)
pollution$h_edumc_None <- as.factor(pollution$h_edumc_None)

# summary statistics and plots of birthweight
mean(pollution$e3_bw)
```

```
## [1] 3378.482
```

```
var(pollution$e3_bw)
```

```
## [1] 259316.9
```

```
hist(pollution$e3_bw, main = "Histogram of birthweight", xlab = "Birthweight",
    breaks = 20)
boxplot(pollution$e3_bw, main = "Boxplot of birthweight", xlab = "Birthweight",
    horizontal = TRUE)
```

```
# creating a numeric-only version of the dataframe
pollution.numeric <- pollution
for (i in 1:ncol(pollution.numeric)) {
    pollution.numeric[, i] <- as.numeric(pollution.numeric[,
        i])
}

# creating a correlogram
data_cols <- pollution.numeric[, c(1:80)]
data_cor <- cor(data_cols)
```

```
heatmap(x = data_cor)
```

```r
# fitting an initial model
M1 <- lm(e3_bw ~ ., data = pollution)
res1 <- resid(M1)  # raw residuals
stud1 <- res1/(sigma(M1) * sqrt(1 - hatvalues(M1)))  # studentized residuals
```

```r
# plot distribution of studentized residuals
hist(stud1, breaks = 12, probability = TRUE, xlim = c(-4, 4),
    xlab = "Studentized Residuals", main = "Distribution of Residuals")
grid <- seq(-3.5, 3.5, by = 0.05)
lines(x = grid, y = dnorm(grid), col = "blue")  # add N(0,1) pdf

# qqplot of studentized residuals
qqnorm(stud1)
abline(0, 1)  # add 45 degree line

# partial regression (added variable plots)
avPlots(M1)

# plot of studentized residuals vs fitted values
plot(stud1 ~ fitted(M1), xlab = "Fitted Vals", ylab = "Studentized Residuals",
    main = "Residuals vs Fitted")

## standard residual plots
plot(M1)
```

```r
# forward selection based on AIC
M_base <- lm(e3_bw ~ 1, training_set)
M_full <- lm(e3_bw ~ ., training_set)
M_forward <- step(object = M_base, scope = list(lower = M_base,
    upper = M_full), direction = "forward", trace = 0)
summary(M_forward)
```

```
##
## Call:
## lm(formula = e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None +
##     h_mbmi_None + hs_wgtgain_None + e3_asmokcigd_p_None + h_pm10_ratio_preg_None +
##     hs_pfoa_m_Log2 + hs_mepa_madj_Log2 + hs_dmtp_madj_Log2 +
##     hs_pb_m_Log2 + h_edumc_None + hs_pbde153_madj_Log2 + h_dairy_preg_Ter +
##     hs_hg_m_Log2 + hs_dep_madj_Log2 + hs_etpa_madj_Log2 + hs_trcs_madj_Log2 +
##     e3_alcpreg_yn_None, data = training_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1375.85  -239.44     2.98   222.28  1131.22
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -2436.044    411.179  -5.925 5.38e-09 ***
## e3_gac_None                  143.751      9.584  14.999  < 2e-16 ***
## h_bro_preg_Log               -20.294      7.707  -2.633 0.008681 **
## e3_sex_Nonemale              149.987     30.624   4.898 1.26e-06 ***
## h_mbmi_None                   13.528      3.039   4.451 1.03e-05 ***
## hs_wgtgain_None               10.018      2.358   4.249 2.51e-05 ***
```

```
## e3_asmokcigd_p_None            -27.326      8.520  -3.207 0.001415 **
## h_pm10_ratio_preg_None          -4.376      2.233  -1.960 0.050499 .
## hs_pfoa_m_Log2                 -58.949     16.711  -3.528 0.000453 ***
## hs_mepa_madj_Log2              -21.246      7.006  -3.033 0.002533 **
## hs_dmtp_madj_Log2               12.774      5.186   2.463 0.014059 *
## hs_pb_m_Log2                   -45.749     23.697  -1.931 0.054020 .
## h_edumc_None2                   97.419     52.268   1.864 0.062849 .
## h_edumc_None3                  151.839     51.866   2.928 0.003551 **
## hs_pbde153_madj_Log2             8.980      5.597   1.604 0.109175
## h_dairy_preg_Ter(17.1,27.1]     56.926     44.109   1.291 0.197364
## h_dairy_preg_Ter(27.1,Inf]     -33.296     43.136  -0.772 0.440500
## hs_hg_m_Log2                   -23.683     12.435  -1.904 0.057346 .
## hs_dep_madj_Log2               -19.076     10.566  -1.806 0.071513 .
## hs_etpa_madj_Log2                9.252      4.741   1.952 0.051463 .
## hs_trcs_madj_Log2               -7.243      4.375  -1.655 0.098385 .
## e3_alcpreg_yn_None1             52.284     33.924   1.541 0.123810
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 366.5 on 578 degrees of freedom
## Multiple R-squared:  0.4401, Adjusted R-squared:  0.4198
## F-statistic: 21.64 on 21 and 578 DF,  p-value: < 2.2e-16
```

```r
D <- cooks.distance(M_forward)
```

```r
plot(M_forward, which = 4)
```

```r
# outliers defined by first rule of thumb (compare to F
# value)
n <- nobs(M_forward)
p <- length(M_forward$coef) - 1
inf_ind <- which(pf(D, p + 1, n - p - 1, lower.tail = TRUE) >
    0.1)  # from lec20 code
```

```r
plot(D, ylab = "Cook's Distance")
points(D[inf_ind] ~ inf_ind, col = "red", pch = 19)  ## add red points
text(y = D[inf_ind], x = inf_ind, labels = inf_ind, pos = 4)  ## label high influence points
```

```r
# no outliers using cook's distance

# DFFITS
dffits_fwd <- dffits(M_forward)
```

```r
## plot DFFITS
plot(dffits_fwd, ylab = "DFFITS")
abline(h = 2 * sqrt((p + 1)/n), lty = 2)  ## add thresholds
abline(h = -2 * sqrt((p + 1)/n), lty = 2)
```

```r
## highlight influential points
dff_ind_fwd <- which(abs(dffits_fwd) > 2 * sqrt((p + 1)/n))  # 20 outliers
```

```r
points(dffits_fwd[dff_ind_fwd] ~ dff_ind_fwd, col = "red", pch = 19)  ## add red points
text(y = dffits_fwd[dff_ind_fwd], x = dff_ind_fwd, labels = dff_ind_fwd,
    pos = 2)  ## label high influence points
```

```r
# create training data removing outliers found using DFFITS
# above
training_set_no_outlier_fwd <- training_set[-dff_ind_fwd, ]

# fit model on this training set
M_base_no_outlier <- lm(e3_bw ~ 1, training_set_no_outlier_fwd)
M_full_no_outlier <- lm(e3_bw ~ ., training_set_no_outlier_fwd)
M_forward_no_outlier <- step(object = M_base_no_outlier, scope = list(lower = M_base_no_outlier,
    upper = M_full_no_outlier), direction = "forward", trace = 0)
summary(M_forward_no_outlier)
```

```
##
## Call:
## lm(formula = e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None +
##     h_mbmi_None + h_pm10_ratio_preg_None + hs_wgtgain_None +
##     hs_dmtp_madj_Log2 + hs_pfoa_m_Log2 + e3_asmokcigd_p_None +
##     hs_mepa_madj_Log2 + h_dairy_preg_Ter + hs_etpa_madj_Log2 +
##     hs_detp_madj_Log2 + hs_pb_m_Log2 + hs_hg_m_Log2 + h_edumc_None +
##     hs_pbde153_madj_Log2 + hs_mep_madj_Log2, data = training_set_no_outlier_fwd)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -789.07 -213.69   14.82  205.12  817.43
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -2478.536    376.293  -6.587 1.07e-10 ***
## e3_gac_None                   144.515      8.717  16.579  < 2e-16 ***
## h_bro_preg_Log                -32.484      6.740  -4.820 1.87e-06 ***
## e3_sex_Nonemale               139.257     26.579   5.239 2.31e-07 ***
## h_mbmi_None                    12.358      2.715   4.551 6.60e-06 ***
## h_pm10_ratio_preg_None         -6.920      1.971  -3.511 0.000483 ***
## hs_wgtgain_None                 8.703      2.065   4.214 2.94e-05 ***
## hs_dmtp_madj_Log2              16.122      4.530   3.559 0.000405 ***
## hs_pfoa_m_Log2                -48.686     15.023  -3.241 0.001266 **
## e3_asmokcigd_p_None           -21.727      7.929  -2.740 0.006344 **
## hs_mepa_madj_Log2             -22.805      6.131  -3.720 0.000220 ***
## h_dairy_preg_Ter(17.1,27.1]    44.622     38.378   1.163 0.245467
## h_dairy_preg_Ter(27.1,Inf]    -48.788     37.484  -1.302 0.193616
## hs_etpa_madj_Log2               7.741      4.137   1.871 0.061876 .
## hs_detp_madj_Log2              -6.819      3.573  -1.908 0.056862 .
## hs_pb_m_Log2                  -35.010     20.661  -1.694 0.090753 .
## hs_hg_m_Log2                  -20.874     10.811  -1.931 0.054028 .
## h_edumc_None2                  66.470     46.359   1.434 0.152200
## h_edumc_None3                 108.437     45.450   2.386 0.017382 *
## hs_pbde153_madj_Log2            8.725      4.870   1.792 0.073755 .
## hs_mep_madj_Log2               11.851      7.588   1.562 0.118883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 308 on 541 degrees of freedom
## Multiple R-squared:  0.5154, Adjusted R-squared:  0.4975
## F-statistic: 28.77 on 20 and 541 DF,  p-value: < 2.2e-16
```

```
# we use MSPE to assess the prediction accuracy of of the
# models
MSPE <- function(yi_new, yi_new_hat) {
    mean((yi_new - yi_new_hat)^2)
}
yi_new <- test_set$e3_bw

# MSPE on model 1
M1.pred <- predict(M_forward, newdata = test_set[, -1])
MSPE_M1 <- MSPE(yi_new, M1.pred)
MSPE_M1
```

```
## [1] 189595.1
```

```
# MSPE on model 2
M2.pred <- predict(M_forward_no_outlier, newdata = test_set[,
    -1])
MSPE_M2 <- MSPE(yi_new, M2.pred)
MSPE_M2
```

```
## [1] 185221.9
```

```
# MSPE using second model lower than that using first model
```

```
# backward selection based on AIC
M_base <- lm(e3_bw ~ 1, training_set)
M_full <- lm(e3_bw ~ ., training_set)
M_backward <- step(object = M_full, scope = list(lower = M_base,
    upper = M_full), direction = "backward", trace = 0)
summary(M_backward)
```

```
##
## Call:
## lm(formula = e3_bw ~ h_pm10_ratio_preg_None + e3_alcpreg_yn_None +
##     h_cereal_preg_Ter + h_dairy_preg_Ter + hs_co_m_Log2 + hs_hg_m_Log2 +
##     hs_pb_m_Log2 + h_pressure_preg_None + h_temperature_preg_None +
##     hs_pcb153_madj_Log2 + hs_pcb180_madj_Log2 + hs_dep_madj_Log2 +
##     hs_dmp_madj_Log2 + hs_dmtp_madj_Log2 + hs_pbde153_madj_Log2 +
##     hs_pfoa_m_Log2 + hs_etpa_madj_Log2 + hs_mepa_madj_Log2 +
##     hs_trcs_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, data = training_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1316.99  -232.36     2.25   222.77  1204.29
##
```

```
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  -7334.312   1959.495  -3.743 0.000200 ***
## h_pm10_ratio_preg_None          -6.021      3.126  -1.926 0.054576 .  
## e3_alcpreg_yn_None1             56.019     34.059   1.645 0.100572    
## h_cereal_preg_Ter(9,27.3]      -45.607     38.451  -1.186 0.236086    
## h_cereal_preg_Ter(27.3,Inf]   -113.645     55.788  -2.037 0.042105 *  
## h_dairy_preg_Ter(17.1,27.1]     49.746     44.367   1.121 0.262664    
## h_dairy_preg_Ter(27.1,Inf]     -50.650     43.827  -1.156 0.248302    
## hs_co_m_Log2                    28.195     15.917   1.771 0.077039 .  
## hs_hg_m_Log2                   -22.581     12.945  -1.744 0.081641 .  
## hs_pb_m_Log2                   -45.395     23.639  -1.920 0.055308 .  
## h_pressure_preg_None             5.338      1.942   2.749 0.006169 ** 
## h_temperature_preg_None        -10.535      5.777  -1.824 0.068742 .  
## hs_pcb153_madj_Log2            -57.767     24.576  -2.351 0.019087 *  
## hs_pcb180_madj_Log2             38.106     17.350   2.196 0.028473 *  
## hs_dep_madj_Log2               -21.032     10.556  -1.992 0.046806 *  
## hs_dmp_madj_Log2                -8.434      5.922  -1.424 0.154932    
## hs_dmtp_madj_Log2               15.366      5.578   2.755 0.006058 ** 
## hs_pbde153_madj_Log2             8.196      5.621   1.458 0.145353    
## hs_pfoa_m_Log2                 -58.802     17.569  -3.347 0.000872 ***
## hs_etpa_madj_Log2                9.751      4.735   2.059 0.039913 *  
## hs_mepa_madj_Log2              -21.617      7.013  -3.082 0.002153 ** 
## hs_trcs_madj_Log2               -7.133      4.459  -1.600 0.110250    
## hs_meohp_madj_Log2              31.427     20.319   1.547 0.122501    
## hs_sumDEHP_madj_Log2           -33.714     23.279  -1.448 0.148092    
## e3_asmokcigd_p_None            -24.878      8.640  -2.879 0.004135 ** 
## h_bro_preg_Log                 -19.033      9.944  -1.914 0.056118 .  
## e3_sex_Nonemale                147.517     30.524   4.833 1.74e-06 ***
## h_mbmi_None                     14.246      3.231   4.409 1.24e-05 ***
## hs_wgtgain_None                  9.872      2.351   4.200 3.10e-05 ***
## e3_gac_None                    144.792      9.724  14.890  < 2e-16 ***
## h_edumc_None2                  111.427     53.846   2.069 0.038963 *  
## h_edumc_None3                  175.669     53.992   3.254 0.001207 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 363 on 568 degrees of freedom
## Multiple R-squared:  0.4603, Adjusted R-squared:  0.4308
## F-statistic: 15.63 on 31 and 568 DF,  p-value: < 2.2e-16
```

```r
D_b <- cooks.distance(M_backward)
```

```r
plot(M_backward, which = 4)
```

```r
# outliers defined by first rule of thumb (compare to F
# value)
n <- nobs(M_backward)
p <- length(M_backward$coef) - 1
inf_ind <- which(pf(D_b, p + 1, n - p - 1, lower.tail = TRUE) >
    0.1)  # from lec20 code
```

```r
plot(D_b, ylab = "Cook's Distance")
points(D_b[inf_ind] ~ inf_ind, col = "red", pch = 19)  ## add red points
text(y = D_b[inf_ind], x = inf_ind, labels = inf_ind, pos = 4)  ## label high influence points
```

```r
# no outliers using cook's distance


# DFFITS
dffits_back <- dffits(M_backward)
```

```r
## plot DFFITS
plot(dffits_back, ylab = "DFFITS")
abline(h = 2 * sqrt((p + 1)/n), lty = 2)  ## add thresholds
abline(h = -2 * sqrt((p + 1)/n), lty = 2)
```

```r
## highlight influential points
dff_ind_back <- which(abs(dffits_back) > 2 * sqrt((p + 1)/n))  # 35 outliers
```

```r
points(dffits_back[dff_ind_back] ~ dff_ind_back, col = "red",
    pch = 19)  ## add red points
text(y = dffits_back[dff_ind_back], x = dff_ind_back, labels = dff_ind_back,
    pos = 2)  ## label high influence points
```

```r
# create training data removing outliers found using DFFITS
# above
training_set_no_outlier_back <- training_set[-dff_ind_back, ]

# fit model on this training set
M_base_no_outlier <- lm(e3_bw ~ 1, training_set_no_outlier_back)
M_full_no_outlier <- lm(e3_bw ~ ., training_set_no_outlier_back)
M_backward_no_outlier <- step(object = M_full_no_outlier, scope = list(lower = M_base_no_outlier,
    upper = M_full_no_outlier), direction = "backward", trace = 0)
summary(M_backward_no_outlier)
```

```
##
## Call:
## lm(formula = e3_bw ~ h_abs_ratio_preg_Log + h_pm10_ratio_preg_None +
##     h_cereal_preg_Ter + h_dairy_preg_Ter + h_fish_preg_Ter +
##     h_pavig_t3_None + hs_co_m_Log2 + hs_hg_m_Log2 + hs_pb_m_Log2 +
##     h_pressure_preg_None + h_temperature_preg_None + hs_pcb153_madj_Log2 +
##     hs_pcb180_madj_Log2 + hs_sumPCBs5_madj_Log2 + hs_dep_madj_Log2 +
##     hs_dmp_madj_Log2 + hs_dmtp_madj_Log2 + hs_pbde153_madj_Log2 +
##     hs_pfoa_m_Log2 + hs_bpa_madj_Log2 + hs_etpa_madj_Log2 + hs_mepa_madj_Log2 +
##     hs_mehp_madj_Log2 + e3_asmokcigd_p_None + h_bro_preg_Log +
##     h_thm_preg_Log + e3_sex_None + h_mbmi_None + hs_wgtgain_None +
##     e3_gac_None + h_edumc_None, data = training_set_no_outlier_back)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -846.10 -215.78   -0.93  195.39  851.01
##
## Coefficients:
```

```
##                              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 -7182.617   1982.635  -3.623 0.000320 ***
## h_abs_ratio_preg_Log           77.793     50.019   1.555 0.120486    
## h_pm10_ratio_preg_None         -7.451      2.994  -2.489 0.013125 *  
## h_cereal_preg_Ter(9,27.3]     -49.659     34.306  -1.448 0.148342    
## h_cereal_preg_Ter(27.3,Inf]  -134.600     49.300  -2.730 0.006540 ** 
## h_dairy_preg_Ter(17.1,27.1]    23.765     39.119   0.608 0.543769    
## h_dairy_preg_Ter(27.1,Inf]    -54.581     38.587  -1.414 0.157809    
## h_fish_preg_Ter(1.9,4.1]      -10.062     35.582  -0.283 0.777455    
## h_fish_preg_Ter(4.1,Inf]      -63.692     38.089  -1.672 0.095078 .  
## h_pavig_t3_NoneLow           -137.703     65.891  -2.090 0.037108 *  
## h_pavig_t3_NoneMedium        -131.667     70.564  -1.866 0.062608 .  
## hs_co_m_Log2                   23.276     14.084   1.653 0.099009 .  
## hs_hg_m_Log2                  -24.968     11.522  -2.167 0.030679 *  
## hs_pb_m_Log2                  -42.912     20.854  -2.058 0.040103 *  
## h_pressure_preg_None            5.489      1.950   2.815 0.005065 ** 
## h_temperature_preg_None        -7.791      5.207  -1.496 0.135191    
## hs_pcb153_madj_Log2           -81.773     24.302  -3.365 0.000821 ***
## hs_pcb180_madj_Log2            43.823     15.011   2.919 0.003657 ** 
## hs_sumPCBs5_madj_Log2          26.696     19.320   1.382 0.167622    
## hs_dep_madj_Log2              -18.843      9.277  -2.031 0.042744 *  
## hs_dmp_madj_Log2              -14.061      5.331  -2.638 0.008590 ** 
## hs_dmtp_madj_Log2              19.882      4.896   4.061 5.62e-05 ***
## hs_pbde153_madj_Log2           10.666      4.956   2.152 0.031844 *  
## hs_pfoa_m_Log2                -61.348     16.230  -3.780 0.000175 ***
## hs_bpa_madj_Log2               17.347      8.822   1.966 0.049774 *  
## hs_etpa_madj_Log2              10.664      4.205   2.536 0.011498 *  
## hs_mepa_madj_Log2             -24.917      6.151  -4.051 5.87e-05 ***
## hs_mehp_madj_Log2             -16.761     10.920  -1.535 0.125415    
## e3_asmokcigd_p_None           -30.271      7.986  -3.790 0.000168 ***
## h_bro_preg_Log                -28.583     11.848  -2.412 0.016187 *  
## h_thm_preg_Log                 23.101     15.805   1.462 0.144442    
## e3_sex_Nonemale               137.844     26.658   5.171 3.31e-07 ***
## h_mbmi_None                    12.754      2.903   4.393 1.35e-05 ***
## hs_wgtgain_None                 8.021      2.080   3.856 0.000130 ***
## e3_gac_None                   138.705      8.952  15.494  < 2e-16 ***
## h_edumc_None2                 121.272     47.946   2.529 0.011717 *  
## h_edumc_None3                 165.886     47.634   3.482 0.000538 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 306.7 on 528 degrees of freedom
## Multiple R-squared:  0.5427, Adjusted R-squared:  0.5115 
## F-statistic: 17.41 on 36 and 528 DF,  p-value: < 2.2e-16
```

```r
# we use MSPE to assess the prediction accuracy of of the
# models
MSPE <- function(yi_new, yi_new_hat) {
    mean((yi_new - yi_new_hat)^2)
}
yi_new <- test_set$e3_bw

# MSPE on model 1
M1.pred <- predict(M_backward, newdata = test_set[, -1])
```

```r
MSPE_M1 <- MSPE(yi_new, M1.pred)
MSPE_M1
```

```
## [1] 196500.4
```

```r
# MSPE on model 2
M2.pred <- predict(M_backward_no_outlier, newdata = test_set[,
    -1])
MSPE_M2 <- MSPE(yi_new, M2.pred)
MSPE_M2
```

```
## [1] 200789.4
```

```r
# MSPE using second model slightly higher than that using
# first model
```

```r
M0 <- lm(e3_bw ~ 1, data = training_set)
Mfull <- lm(e3_bw ~ ., data = training_set)
Mstep <- step(object = M0, scope = list(lower = M0, upper = Mfull),
    direction = "both", trace = 0)
summary(Mstep)
```

```
##
## Call:
## lm(formula = e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None +
##     h_mbmi_None + hs_wgtgain_None + e3_asmokcigd_p_None + h_pm10_ratio_preg_None +
##     hs_pfoa_m_Log2 + hs_mepa_madj_Log2 + hs_dmtp_madj_Log2 +
##     hs_pb_m_Log2 + h_edumc_None + hs_pbde153_madj_Log2 + h_dairy_preg_Ter +
##     hs_hg_m_Log2 + hs_dep_madj_Log2 + hs_etpa_madj_Log2 + hs_trcs_madj_Log2 +
##     e3_alcpreg_yn_None, data = training_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1375.85  -239.44     2.98   222.28  1131.22
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -2436.044    411.179  -5.925 5.38e-09 ***
## e3_gac_None                 143.751      9.584  14.999  < 2e-16 ***
## h_bro_preg_Log              -20.294      7.707  -2.633 0.008681 **
## e3_sex_Nonemale             149.987     30.624   4.898 1.26e-06 ***
## h_mbmi_None                  13.528      3.039   4.451 1.03e-05 ***
## hs_wgtgain_None              10.018      2.358   4.249 2.51e-05 ***
## e3_asmokcigd_p_None         -27.326      8.520  -3.207 0.001415 **
## h_pm10_ratio_preg_None       -4.376      2.233  -1.960 0.050499 .
## hs_pfoa_m_Log2              -58.949     16.711  -3.528 0.000453 ***
## hs_mepa_madj_Log2           -21.246      7.006  -3.033 0.002533 **
## hs_dmtp_madj_Log2            12.774      5.186   2.463 0.014059 *
## hs_pb_m_Log2                -45.749     23.697  -1.931 0.054020 .
## h_edumc_None2                97.419     52.268   1.864 0.062849 .
## h_edumc_None3               151.839     51.866   2.928 0.003551 **
```

```
## hs_pbde153_madj_Log2            8.980      5.597   1.604 0.109175
## h_dairy_preg_Ter(17.1,27.1]    56.926     44.109   1.291 0.197364
## h_dairy_preg_Ter(27.1,Inf]    -33.296     43.136  -0.772 0.440500
## hs_hg_m_Log2                  -23.683     12.435  -1.904 0.057346 .
## hs_dep_madj_Log2              -19.076     10.566  -1.806 0.071513 .
## hs_etpa_madj_Log2               9.252      4.741   1.952 0.051463 .
## hs_trcs_madj_Log2              -7.243      4.375  -1.655 0.098385 .
## e3_alcpreg_yn_None1            52.284     33.924   1.541 0.123810
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 366.5 on 578 degrees of freedom
## Multiple R-squared:  0.4401, Adjusted R-squared:  0.4198
## F-statistic: 21.64 on 21 and 578 DF,  p-value: < 2.2e-16
```

```r
M <- Mstep
n <- nobs(M)  ##600 observations
p <- length(M$coef) - 1  ##21
dffits_m <- dffits(M)  ##using DFFITS
```

```r
## plot DFFITS
plot(dffits_m, ylab = "DFFITS", main = "DFFITS plot of First model")
abline(h = 2 * sqrt((p + 1)/n), lty = 2)  ## add thresholds
abline(h = -2 * sqrt((p + 1)/n), lty = 2)
```

```r
## highlight influential points
dff_ind <- which(abs(dffits_m) > 2 * sqrt((p + 1)/n))
```

```r
points(dffits_m[dff_ind] ~ dff_ind, col = "red", pch = 19)  ## add red points
text(y = dffits_m[dff_ind], x = dff_ind, labels = dff_ind, pos = 2)  ## label high influence points
```

```r
# new dataset after removing outliers
new.set <- training_set[-dff_ind, ]

# fitting the same model with new dataset
M0 <- lm(e3_bw ~ 1, data = new.set)
Mfull <- lm(e3_bw ~ ., data = new.set)
Mstep.new <- step(object = M0, scope = list(lower = M0, upper = Mfull),
    direction = "both", trace = 0)
summary(Mstep.new)
```

```
##
## Call:
## lm(formula = e3_bw ~ e3_gac_None + h_bro_preg_Log + e3_sex_None +
##     h_mbmi_None + h_pm10_ratio_preg_None + hs_wgtgain_None +
##     hs_dmtp_madj_Log2 + hs_pfoa_m_Log2 + e3_asmokcigd_p_None +
##     hs_mepa_madj_Log2 + h_dairy_preg_Ter + hs_etpa_madj_Log2 +
##     hs_detp_madj_Log2 + hs_pb_m_Log2 + hs_hg_m_Log2 + h_edumc_None +
##     hs_pbde153_madj_Log2 + hs_mep_madj_Log2, data = new.set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -789.07 -213.69   14.82  205.12  817.43
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -2478.536    376.293  -6.587 1.07e-10 ***
## e3_gac_None                   144.515      8.717  16.579  < 2e-16 ***
## h_bro_preg_Log                -32.484      6.740  -4.820 1.87e-06 ***
## e3_sex_Nonemale               139.257     26.579   5.239 2.31e-07 ***
## h_mbmi_None                    12.358      2.715   4.551 6.60e-06 ***
## h_pm10_ratio_preg_None         -6.920      1.971  -3.511 0.000483 ***
## hs_wgtgain_None                 8.703      2.065   4.214 2.94e-05 ***
## hs_dmtp_madj_Log2              16.122      4.530   3.559 0.000405 ***
## hs_pfoa_m_Log2                -48.686     15.023  -3.241 0.001266 **
## e3_asmokcigd_p_None           -21.727      7.929  -2.740 0.006344 **
## hs_mepa_madj_Log2             -22.805      6.131  -3.720 0.000220 ***
## h_dairy_preg_Ter(17.1,27.1]    44.622     38.378   1.163 0.245467
## h_dairy_preg_Ter(27.1,Inf]    -48.788     37.484  -1.302 0.193616
## hs_etpa_madj_Log2               7.741      4.137   1.871 0.061876 .
## hs_detp_madj_Log2              -6.819      3.573  -1.908 0.056862 .
## hs_pb_m_Log2                  -35.010     20.661  -1.694 0.090753 .
## hs_hg_m_Log2                  -20.874     10.811  -1.931 0.054028 .
## h_edumc_None2                  66.470     46.359   1.434 0.152200
## h_edumc_None3                 108.437     45.450   2.386 0.017382 *
## hs_pbde153_madj_Log2            8.725      4.870   1.792 0.073755 .
## hs_mep_madj_Log2               11.851      7.588   1.562 0.118883
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 308 on 541 degrees of freedom
## Multiple R-squared:  0.5154, Adjusted R-squared:  0.4975
## F-statistic: 28.77 on 20 and 541 DF,  p-value: < 2.2e-16
```

```r
M2 <- Mstep.new
n <- nobs(M2)  ##600 observations
p <- length(M2$coef) - 1  ##20
dffits_m <- dffits(M2)
```

```r
M1.res <- test_set$e3_bw - # test observations
  predict(M, newdata = test_set) # prediction with training data
M2.res <- test_set$e3_bw - predict(M2, newdata = test_set)

#MSPE for First stepwise selected model
mean(M1.res^2)
```

```
## [1] 189595.1
```

```r
#MSPE for no outliers stepwise selected model
mean(M2.res^2)
```

```
## [1] 185221.9
```

```r
y = pollution$e3_bw
X = pollution[!names(pollution) %in% c("e3_bw")]


# Splitting the columns into four domains
pollution_chemicals <- pollution[c(18:27, 31:70)]  # chemical domain
pollution_outdoors <- pollution[c(2:5, 28:30, 71:73)]  # outdoor exposures domain
pollution_lifestyles <- pollution[c(6:17)]  # lifestyles domain
pollution_others <- pollution[c(74:80)]  # covariates domain


# A function for finding the DFFits outliers at a specific
# tolerance. Ploting the values is an included option.
Get_DFFITS_Outliers <- function(M, tol = 2, to_plot = FALSE,
    ylab = "DFFITS") {
    D = dffits(M)
    n = nobs(M)
    p <- length(M$coef) - 1
    Out_Indices = which(abs(D) > tol * sqrt((p + 1)/n))

    if (to_plot) {
        plot(D, ylab = ylab)
        abline(h = tol * sqrt((p + 1)/n), lty = 2)
        abline(h = -tol * sqrt((p + 1)/n), lty = 2)

        points(D[Out_Indices] ~ Out_Indices, col = "red", pch = 19)
        text(y = D[Out_Indices], x = Out_Indices, labels = Out_Indices,
            pos = 2)

    }

    return(as.vector(Out_Indices))
}

# One instance of the simulation
Simulation_Instance <- function(y_train, X_train, y_test, X_test,
    cols, plot_outliers = FALSE) {

    M = lm(y_train ~ ., data = X_train[cols])  # Create the model using the entire training set
    M_Outliers = c()

    M_Outliers = Get_DFFITS_Outliers(M, to_plot = plot_outliers)  # Identify the outliers

    X_train_rm_out = X_train[-M_Outliers, cols]
    y_train_rm_out = y_train[-M_Outliers]

    # Remove the outliers from the training set and remake
    # the model
    M_Outliers_Removed = lm(y_train_rm_out ~ ., data = X_train_rm_out)

    mspe_1 = MSPE(y_test, X_test, M)
    mspe_2 = MSPE(y_test, X_test, M_Outliers_Removed)  #Compare the MSPEs of the 2 models
```

```r
    # Return information about the instance
    return(list(length(cols), length(M_Outliers), mspe_1, mspe_2,
        mspe_1 - mspe_2))

}

# Full simulation:
Run_Simulation <- function(ITERS, y, X, CV = 1, tts = 0.6, prop0 = 1,
    prop1 = 1) {
    COLS = length(X)  # number of columns
    table = data.frame(features = rep(NA, ITERS * CV), num_outliers = rep(NA,
        ITERS * CV), mspe_original = rep(NA, ITERS * CV), mspe_no_outliers = rep(NA,
        ITERS * CV), mspe_difference = rep(NA, ITERS * CV))

    for (cv in 1:CV) {
        # different training splits used for
        # cross-validation
        sampler = 1:(length(y) * tts) + (((length(y) * (1 - tts) *
            (cv - 1))/(CV - 1)))

        y_train = y[sampler]
        X_train = X[sampler, ]
        y_test = y[-sampler]
        X_test = X[-sampler, ]

        # Creates the bool matrix of chosen columns
        RandomFeatureSelectionList <- list()
        for (i in 1:ITERS) {
            RandomFeatureSelectionList[[i]] = sample(c(rep(0,
                prop0), rep(1, prop1)), replace = TRUE, size = COLS)
        }
        # Converts The booleans of the matrix into a list
        # of columns

        for (i in 1:length(RandomFeatureSelectionList)) {
            col_list = c()
            for (j in 1:COLS) {
                if (RandomFeatureSelectionList[[i]][j] == 1) {
                  col_list = append(col_list, j)
                }
            }
            sim_inst = try(Simulation_Instance(y_train, X_train,
                y_test, X_test, col_list), silent = TRUE)
            if (length(sim_inst) == 5) {
                for (j in 1:5) {

                  table[((cv - 1) * ITERS) + i, j] = sim_inst[[j]]

                }
            }
        }

    }
```

```
    return(table)
}

# Draws a histogram and the ascosiated normal distribution
Hist_Norm <- function(d, breaks = 20, x_lab = "x_lab", y_lab = "y_lab",
    main = "main") {
    full = d[!is.na(d)]
    print(min(full))
    h <- hist(full, breaks = breaks, x_lab = "x_lab", y_lab = "y_lab",
        main = "main")
    xfit <- seq(min(full), max(full), length = 40)
    yfit <- dnorm(xfit, mean = mean(full), sd = sd(full))
    yfit <- yfit * diff(h$mids[1:2]) * length(full)
    lines(xfit, yfit, col = "black", lwd = 2)

}


# Set the seed for ease of replication:
set.seed(20776408)

# Models with equal chance of including or excluding
# columns, across the different domains
domain1 = Run_Simulation(1000, y, pollution_chemicals, CV = 5)
domain2 = Run_Simulation(1000, y, pollution_outdoors, CV = 5)
domain3 = Run_Simulation(1000, y, pollution_lifestyles, CV = 5)
domain4 = Run_Simulation(1000, y, pollution_others, CV = 5)
domain12 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors), CV = 5)
domain13 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_lifestyles), CV = 5)
domain14 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_others), CV = 5)
domain23 = Run_Simulation(1000, y, cbind(pollution_outdoors,
    pollution_lifestyles), CV = 5)
domain24 = Run_Simulation(1000, y, cbind(pollution_outdoors,
    pollution_others), CV = 5)
domain34 = Run_Simulation(1000, y, cbind(pollution_lifestyles,
    pollution_others), CV = 5)
domain123 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors, pollution_lifestyles), CV = 5)
domain124 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors, pollution_others), CV = 5)
domain134 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_lifestyles, pollution_others), CV = 5)
domain234 = Run_Simulation(1000, y, cbind(pollution_outdoors,
    pollution_lifestyles, pollution_others), CV = 5)
domain1234 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors, pollution_lifestyles, pollution_others),
    CV = 5)

# Models with a higher chance of excluding columns, across
# the different domains
```

```r
low_col_domain1 = Run_Simulation(1000, y, pollution_chemicals,
    CV = 5, prop0 = 2)
low_col_domain2 = Run_Simulation(1000, y, pollution_outdoors,
    CV = 5, prop0 = 2)
low_col_domain3 = Run_Simulation(1000, y, pollution_lifestyles,
    CV = 5, prop0 = 2)
low_col_domain4 = Run_Simulation(1000, y, pollution_others, CV = 5,
    prop0 = 2)
low_col_domain12 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors), CV = 5, prop0 = 2)
low_col_domain13 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_lifestyles), CV = 5, prop0 = 2)
low_col_domain14 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_others), CV = 5, prop0 = 2)
low_col_domain23 = Run_Simulation(1000, y, cbind(pollution_outdoors,
    pollution_lifestyles), CV = 5, prop0 = 2)
low_col_domain24 = Run_Simulation(1000, y, cbind(pollution_outdoors,
    pollution_others), CV = 5, prop0 = 2)
low_col_domain34 = Run_Simulation(1000, y, cbind(pollution_lifestyles,
    pollution_others), CV = 5, prop0 = 2)
low_col_domain123 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors, pollution_lifestyles), CV = 5, prop0 = 2)
low_col_domain124 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors, pollution_others), CV = 5, prop0 = 2)
low_col_domain134 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_lifestyles, pollution_others), CV = 5, prop0 = 2)
low_col_domain234 = Run_Simulation(1000, y, cbind(pollution_outdoors,
    pollution_lifestyles, pollution_others), CV = 5, prop0 = 2)
low_col_domain1234 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors, pollution_lifestyles, pollution_others),
    CV = 5, prop0 = 2)

# Models with a higher chance of including columns, across
# the different domains
high_col_domain1 = Run_Simulation(1000, y, pollution_chemicals,
    CV = 5, prop1 = 2)
high_col_domain2 = Run_Simulation(1000, y, pollution_outdoors,
    CV = 5, prop1 = 2)
high_col_domain3 = Run_Simulation(1000, y, pollution_lifestyles,
    CV = 5, prop1 = 2)
high_col_domain4 = Run_Simulation(1000, y, pollution_others,
    CV = 5, prop1 = 2)
high_col_domain12 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors), CV = 5, prop1 = 2)
high_col_domain13 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_lifestyles), CV = 5, prop1 = 2)
high_col_domain14 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_others), CV = 5, prop1 = 2)
high_col_domain23 = Run_Simulation(1000, y, cbind(pollution_outdoors,
    pollution_lifestyles), CV = 5, prop1 = 2)
high_col_domain24 = Run_Simulation(1000, y, cbind(pollution_outdoors,
    pollution_others), CV = 5, prop1 = 2)
high_col_domain34 = Run_Simulation(1000, y, cbind(pollution_lifestyles,
```

```r
    pollution_others), CV = 5, prop1 = 2)
high_col_domain123 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors, pollution_lifestyles), CV = 5, prop1 = 2)
high_col_domain124 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors, pollution_others), CV = 5, prop1 = 2)
high_col_domain134 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_lifestyles, pollution_others), CV = 5, prop1 = 2)
high_col_domain234 = Run_Simulation(1000, y, cbind(pollution_outdoors,
    pollution_lifestyles, pollution_others), CV = 5, prop1 = 2)
high_col_domain1234 = Run_Simulation(1000, y, cbind(pollution_chemicals,
    pollution_outdoors, pollution_lifestyles, pollution_others),
    CV = 5, prop1 = 2)


# combining the observations across the column inclusion
# probabilities
tot_domain1 = unique(rbind(low_col_domain1, domain1, high_col_domain1))
tot_domain2 = unique(rbind(low_col_domain2, domain2, high_col_domain2))
tot_domain3 = unique(rbind(low_col_domain3, domain3, high_col_domain3))
tot_domain4 = unique(rbind(low_col_domain4, domain4, high_col_domain4))
tot_domain12 = unique(rbind(low_col_domain12, domain12, high_col_domain12))
tot_domain13 = unique(rbind(low_col_domain13, domain13, high_col_domain13))
tot_domain14 = unique(rbind(low_col_domain14, domain14, high_col_domain14))
tot_domain23 = unique(rbind(low_col_domain23, domain23, high_col_domain23))
tot_domain24 = unique(rbind(low_col_domain24, domain24, high_col_domain24))
tot_domain34 = unique(rbind(low_col_domain34, domain34, high_col_domain34))
tot_domain123 = unique(rbind(low_col_domain123, domain123, high_col_domain123))
tot_domain124 = unique(rbind(low_col_domain124, domain124, high_col_domain124))
tot_domain134 = unique(rbind(low_col_domain134, domain134, high_col_domain134))
tot_domain234 = unique(rbind(low_col_domain234, domain234, high_col_domain234))
tot_domain1234 = unique(rbind(low_col_domain1234, domain1234,
    high_col_domain1234))



# Calcluating mean and standard deviation of the MSPE
# differences observed above, and plotting the respective
# histograms.
Hist_Norm(tot_domain1234$mspe_difference, xlab = "Original Model MSPE minus Outlier Removed Model MSPE"
    main = "MSPE Difference in models across all domains")
mean(tot_domain1234$mspe_difference, na.rm = TRUE)
sd(tot_domain1234$mspe_difference, na.rm = TRUE)

Hist_Norm(tot_domain1$mspe_difference, xlab = "Original Model MSPE minus Outlier Removed Model MSPE",
    main = "MSPE Difference in models within the Chemicals Domain")
mean(tot_domain1$mspe_difference, na.rm = TRUE)
sd(tot_domain1$mspe_difference, na.rm = TRUE)
Hist_Norm(tot_domain2$mspe_difference, xlab = "Original Model MSPE minus Outlier Removed Model MSPE",
    main = "MSPE Difference in models Within the Outdoors Domain")
mean(tot_domain2$mspe_difference, na.rm = TRUE)
sd(tot_domain2$mspe_difference, na.rm = TRUE)
Hist_Norm(tot_domain3$mspe_difference, xlab = "Original Model MSPE minus Outlier Removed Model MSPE",
    main = "MSPE Difference in models within the Lifestyle Domain")
```

```
mean(tot_domain3$mspe_difference, na.rm = TRUE)
sd(tot_domain3$mspe_difference, na.rm = TRUE)
Hist_Norm(tot_domain4$mspe_difference, xlab = "Original Model MSPE minus Outlier Removed Model MSPE",
    main = "MSPE Difference in models Within the Misc. Domain")
mean(tot_domain4$mspe_difference, na.rm = TRUE)
sd(tot_domain4$mspe_difference, na.rm = TRUE)
```