CODE

```
# Part I data exploration

# read data
library(xlsx)
mortality <- read.xlsx("mortality.xls", sheetIndex = 1)
mortality <- mortality[, -8]
plot(mortality)
par(mfrow = c(3, 3))
sapply(colnames(mortality), function(x) hist(mortality[[x]], main = x))

# summary statistics
apply(mortality, 2, summary)

# single regression
single_regression <- lapply(colnames(mortality)[-7], function(x) {
 fit <- lm(as.formula(sprintf("MORTALITY ~ %s", x)), data = mortality_std)
 summary(fit)
})


# compare initial, log, square, square root
par(mfrow = c(3, 4))
sapply(colnames(mortality)[1:6], function(x) {
 hist(mortality[[x]], main = sprintf("Hist of %s", x), xlab = x)
 hist(log(mortality[[x]]), main = sprintf("Hist of log(%s)", x), xlab = sprintf("%s", x))
 hist(sqrt(mortality[[x]]), main = sprintf("Hist of sqrt(%s)", x), xlab = sprintf("%s", x))
 hist(mortality[[x]]^2, main = sprintf("Hist of square(%s)", x), xlab = sprintf("%s", x))
})

# do the transformation based on our discussion, log for NOX and SO2, sqrt for NONWHITE
mortality_new <- mortality
mortality_new$NONWHITE <- sqrt(mortality_new$NONWHITE)
mortality_new$NOX <- log(mortality_new$NOX)
mortality_new$SO2 <- log(mortality_new$SO2)

# standardize
mortality_std <- as.data.frame(scale(mortality_new))

# see the correlation
plot(mortality_std)
cor(mortality_std)
library(corrplot)
par(mfrow=c(1,1), mar=c(1,1,1,1))
corrplot(cor(mortality_std), method = "circle", pch = 1, tl.cex=0.75)
corrplot(cor(mortality_std), method = "number", pch = 1, tl.cex=0.75)


# vif
```

```
vif_regrssion <- sapply(colnames(mortality_std)[-7], function(x) {
 fit <- lm(as.formula(sprintf("%s ~ . - MORTALITY", x)), data = mortality_std)
 return(1/(1 - summary(fit)$r.squared))
})

# Additional: regress y only on pollution
lm_fit0 <- lm(MORTALITY ~ NOX + SO2, data = mortality_std)
summary(lm_fit0)
# not significant

#-------------------------------------------------------------------------------
# Part II stepwise regression

# first order initial
lm_fit1 <- lm(MORTALITY ~ 0 + ., data = mortality_std)
summary(lm_fit1)
par(mfrow = c(2, 2))
plot(lm_fit1, which = 1:4)

# first order stepwise
library(MASS)
lm_fit2 <- stepAIC(lm(MORTALITY ~ 1, data = mortality_std), scope = list(upper = lm_fit1),
              direction = "both", k = 2)
lm_fit2 <- lm(formula = MORTALITY ~ 0 + NONWHITE + EDUC + SO2 + PRECIP, data =
mortality_std)
plot(lm_fit2, which = 1:4)

# we can see outliers, but to compare this model to subsequent methods easily, we do not
# drop them, also because this is not a severe proble

# second order initial
lm_fit3 <- stepAIC(lm(MORTALITY ~ 1, data = mortality_std),
              scope = list(upper = lm(MORTALITY ~ 0 + . ^ 2, data = mortality_std)),
              direction = "both", k = 2)
lm_fit3 <- lm(formula = MORTALITY ~ 0 + NONWHITE + EDUC + SO2 + PRECIP +
NONWHITE:SO2 +
            EDUC:SO2 + SO2:PRECIP, data = mortality_std)
plot(lm_fit3, which = 1:4)

# cross validation statistics
lm_fit2_cv <- 1 / 60 * sum((residuals(lm_fit2) / (1 - influence(lm_fit2)$hat))^2)
lm_fit3_cv <- 1 / 60 * sum((residuals(lm_fit3) / (1 - influence(lm_fit3)$hat))^2)
# cross validation 0.384, 0.394, so select linear model

#-------------------------------------------------------------------------------
# Part III PLS

library(pls)
par(mfrow = c(1,1))
set.seed(1)
pls_fit1 <- plsr(MORTALITY ~ 0 + ., 5, data = mortality_std, validation = 'CV')
```

```r
summary(pls_fit1)

# after the random seed is set, I see 4 components
set.seed(1)
pls_fit2 <- plsr(MORTALITY ~ 0 + ., 4, data = mortality_std, validation = 'CV')
summary(pls_fit2)
pls_fit2_coef <- pls_fit2$coefficients[,,4]
pls_fit2_fitted <- pls_fit2$fitted.values[,,4]
pls_fit2_residual <- mortality_std$MORTALITY - pls_fit2_fitted
loadings(pls_fit2)

# diagnostic plot
plot_diagnostic <- function(fitted_values, residual_values, string) {
 par(mfrow = c(2, 2))
 # the observed against the fitted values
 plot(mortality_std$MORTALITY, fitted_values,
     main = sprintf("observed value vs fitted values (%s)", string),
     xlab = "mortality", ylab = 'fitted mortality')
 abline(a = 0, b = 1)
 # qqnorm
 qqnorm(fitted_values, main = sprintf("QQ plot (%s)", string))
 qqline(fitted_values)
 # residuals against fitted
 plot(residual_values, fitted_values,
     main = sprintf("residuals vs fitted values (%s)", string),
     xlab = "residuals", ylab = 'fitted mortality')
 abline(h = 0)
 # histogram of residuals
 hist(residual_values, main = sprintf('histogram of residuals (%s)', string),
     xlab = 'residuals')
 par(mfrow = c(1, 1))
}

plot_diagnostic(pls_fit2_fitted, pls_fit2_residual, "pls")

# cross-validation result
summary(pls_fit2)

#-------------------------------------------------------------------------------
# Part IV ridge

set.seed(1)
ridge_fit1 <- cv.glmnet(as.matrix(mortality_std[, -7]), mortality_std$MORTALITY,
                        alpha = 0, intercept = F)
plot(ridge_fit1)
ridge_lambda <- ridge_fit1$lambda.min
ridge_coef <- coef(ridge_fit1, s = ridge_lambda)@x
ridge_fit1_fitted <- as.matrix(mortality_std[, -7]) %*% ridge_coef
ridge_fit1_residual <- mortality_std$MORTALITY - ridge_fit1_fitted

# ridge_vif
```

```
R <- as.matrix(t(mortality_std[, -7])) %*% as.matrix(mortality_std[, -7])
A <- solve(R + diag(ridge_lambda, 6)) %*% R %*% solve(R + diag(ridge_lambda, 6))
ridge_vif <- diag(solve(A)) * diag(A)
# only a little smaller, since our lambda is very small

# diagnostic plot
plot_diagnostic(ridge_fit1_fitted, ridge_fit1_residual, "ridge")

# cross validation result
ridge_fit1_cv <- ridge_fit1$cvm[which(ridge_fit1$lambda == ridge_lambda)]

#------------------------------------------------------------------------------
# Part V lasso

set.seed(1)
lasso_fit1 <- cv.glmnet(as.matrix(mortality_std[, -7]), mortality_std$MORTALITY,
                        alpha = 1, intercept = F)
plot(lasso_fit1)
lasso_lambda <- lasso_fit1$lambda.min
lasso_coef <- coef(lasso_fit1, s = lasso_lambda)@x
# someone is set to zero, see here
lasso_coef <- c(lasso_coef[1:3], 0, lasso_coef[4:5])
lasso_fit1_fitted <- as.matrix(mortality_std[, -7]) %*% lasso_coef
lasso_fit1_residual <- mortality_std$MORTALITY - ridge_fit1_fitted

# diagnostic plot
plot_diagnostic(lasso_fit1_fitted, lasso_fit1_residual, "lasso")

# cross validation result
lasso_fit1_cv <- lasso_fit1$cvm[which(lasso_fit1$lambda == lasso_lambda)]

#------------------------------------------------------------------------------
# Part VI linear model revisit (validation)

# we know the outliers may be 37.
lm_fit4 <- lm(formula = MORTALITY ~ 0 + NONWHITE + EDUC + SO2 + PRECIP,
              data = mortality_std[-37, ])
par(mfrow = c(2, 2))
plot(lm_fit4, which = 1:4)
lm_fit4_cv <- 1 / 60 * (sum((residuals(lm_fit4) / (1 - influence(lm_fit4)$hat))^2) +
                        sum(mortality_std[37, 7] - predict(lm_fit4, mortality_std[37, ])))
```