ST404 Assignment 2

Frank Or, Remos Gong, Sam Glanfield, Thomas Broadbent

2022-03-06

Contents

1 Findings

From our model building process we have identified that the major determinants in high mortality rates of cancer in the US are: **Incidence Rate, Percent Unemployed 16 and Over, Percent Employer Provided Private Coverage and Percent Black**. Whereas, the major determinants in low mortality rates are: **Percent Employed 16 and Over, Percent Private Coverage, Education Levels and Median Income**.

In order to ensure an accurate model we transformed the variables Median Income and Percent Black to obtain symmetrical distributions for these variables. We log transformed Median Income which when comparing 2 different incomes instead of having the difference of these we have the percentage change. For Percent Black we shifted all values up by 0.05 which can be seen as taking into account any rounding errors then again taking a log transform to fix having the majority of data points in the 0-5% range. We also had some missing data in Percent Employed 16 and Over which we found there to be no pattern to the missing data and thus we calculated what we would expect these values to be based off of the values we see in the other complete data entries.

We do see examples of counties that do not conform to the general pattern with unusually high or low mortality rates. One example we have is 'Williamsburg City, Virginia' where we see a high incidence rate of 1014 yet a comparatively small death rate of 162, however we see that it's percent private coverage is 19% above the US average and the median male and female ages of 26 and 24 respectively are considerably lower than the median US ages, potentially being causes for this low death rate.

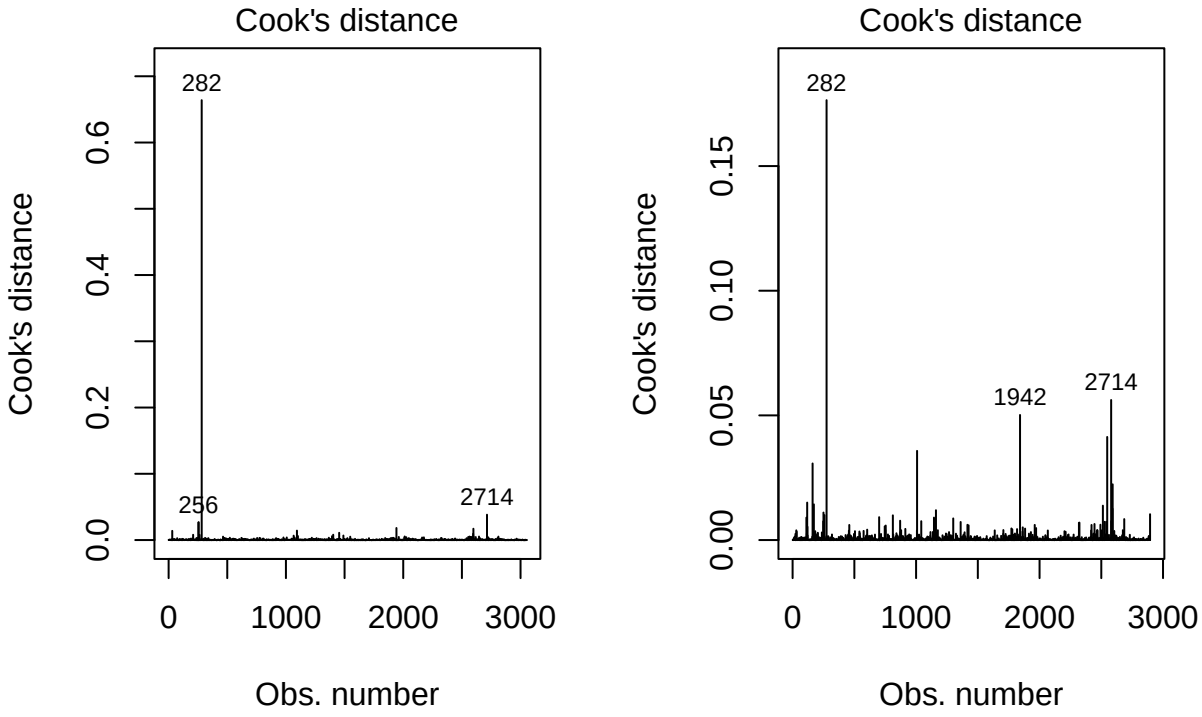
2 Statistical Methodology

We first combined our results and findings from our preliminary EDA which is mainly discussed in Outliers and Transformations.

2.1 Outliers

Counties with high Incidence Rates, namely 'Union County, Florida' and 'Williamsburg City, Virginia.' We looked into the cook's distance plots and noticed only 'Williamsburg City, Virginia' has large cook's distance and hence influential. The first cook's distance plot used a linear model with only incidenceRate as the predictor variable. The second used all the numerical variables. We

concluded although 'Union County, Florida' has high leverage, it is not influential and hence should be kept in our data set.



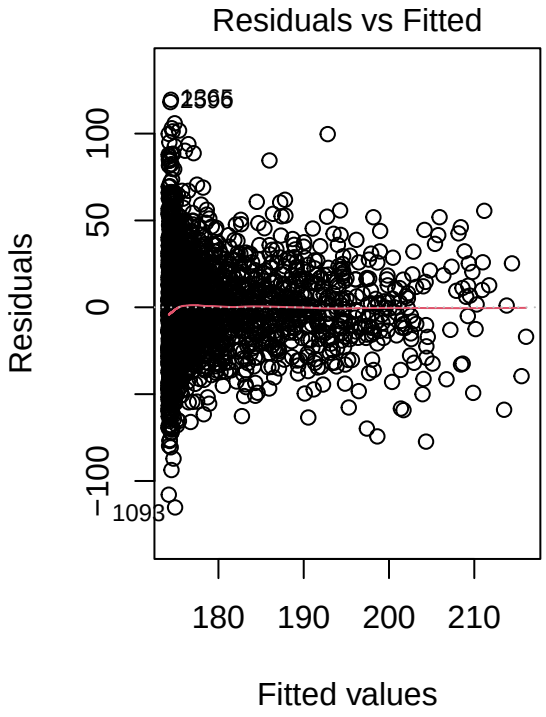
For counties with Average Household Size less than one we took the decision to scale the transformations by 100 and keep them in the dataset. This fixed the normality of AvgHouseholdSize as shown in the histogram.

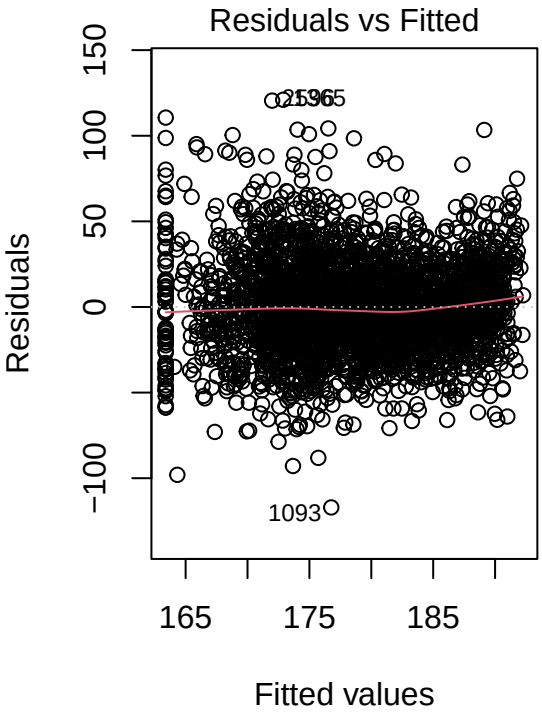
We also see counties with missing values in Percentage Employed 16 and Over and we conclude that the data is Missing Completely at Random. In order to rectify this we impute this data by fitting a linear regression model of Percentage Employed 16 and Over on the remaining variables to estimate what these values would be.

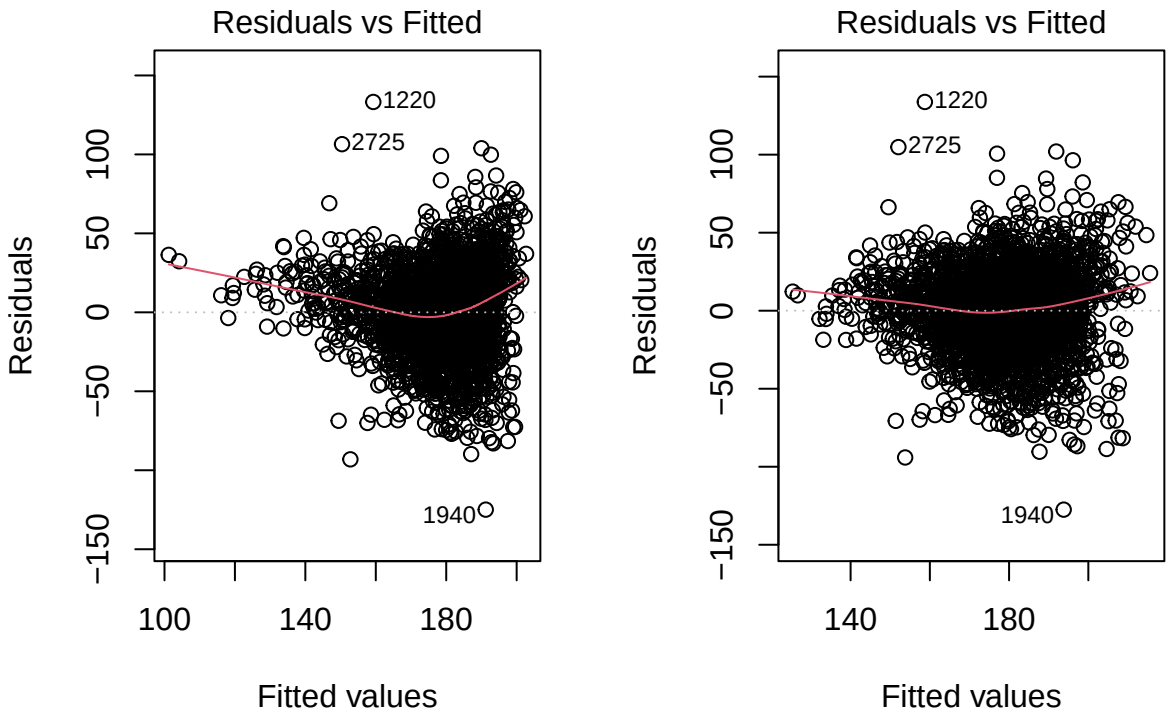
2.2 Transformations

We transform Percent Black by first shifting the values upwards by 0.05, to ensure we have no zero values, then take a log transform. We also transform the Median Income by again taking a log trasformation. We do these transformations to ensure the data is not heavily skewed and allow for a more accurate model.

The following residual plots show the improvements in homoscedasticity in PctBlack and medIncome after log-transform respectively.



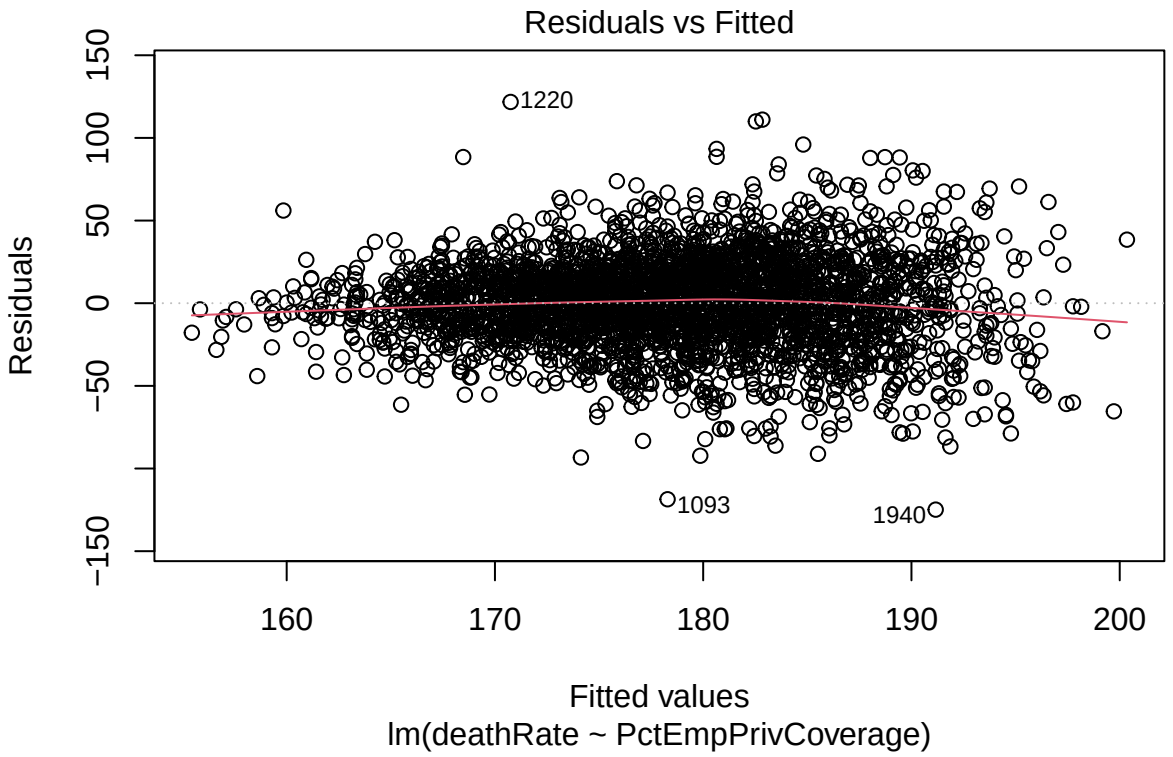




We also notice slight evidence of heteroscedasticity in PctEmpPrivCoverage as shown in the residual plot below. However, since it is not severe and the suggested power transformation using spread level plot makes it more heteroscedastic, we decided not to perform transformation on PctEmpPrivCoverage.

```
# Finding power transformation for PctEmpPrivCoverage
spreadLevelPlot(lm(deathRate~PctEmpPrivCoverage, data=cancer))
```

```
##
## Suggested power transformation:  -4.380067
```



2.3 Modelling Approach and Variable Selection

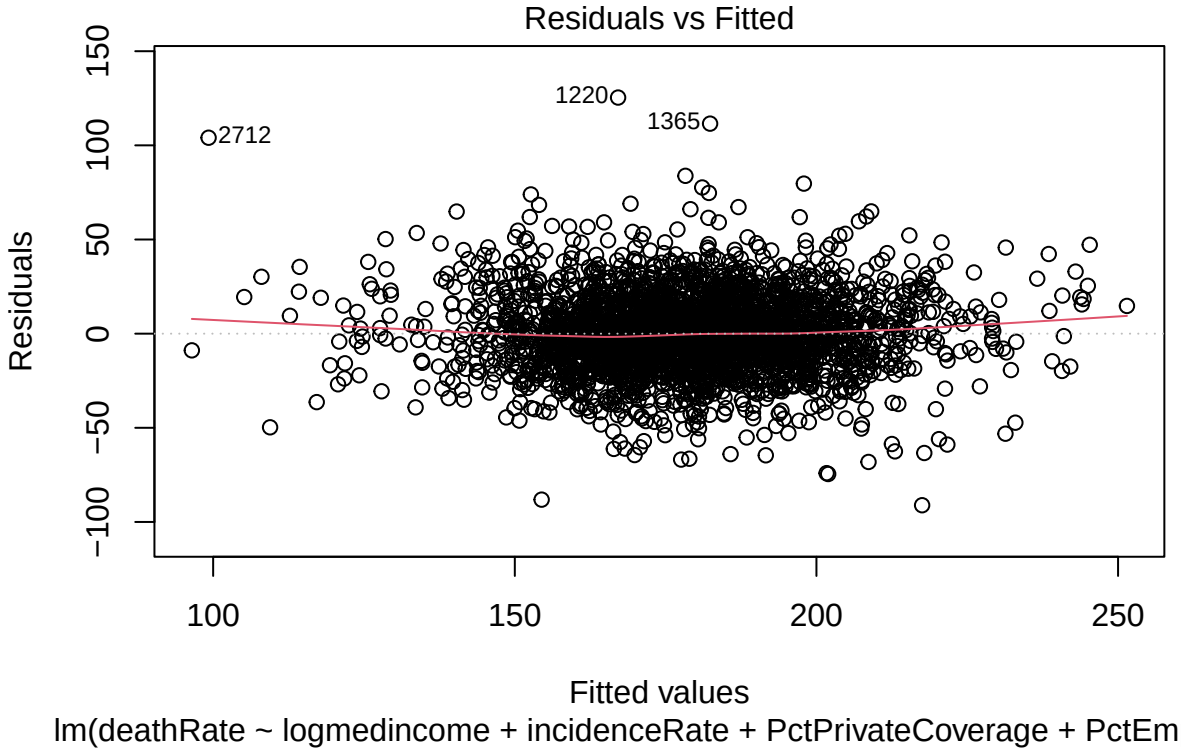
2.3.1 AIC and BIC Forward and Backward Variable Selection

We perform forward, backward and hybrid stepwise regression according to both AIC and BIC. We see that the models generated for AIC are all the same and ones generated for BIC are al the same. In the AIC case we have the model has 12 parameters, whereas the BIC model has 11 variables.

From the summary output we see that for the AIC model all coefficients have strong evidence that they are different from zero other than Percent Unemployed 16 and Over and Percent Public Coverage. For the BIC model these variables are not in the model and further from the summary output we see with strong evidence that all the coefficients are different from zero. From observing the output we see that we would expect there to be multicollinearity in these models due to variables that measure similar or opposite quantities, for example Percent Married and Percent Married Households.

Below we see the residual plots for the stepwise BIC model and can see in the left plot that we

appear to have constant variance and the residuals have a mean of zero satisfying these assumptions.

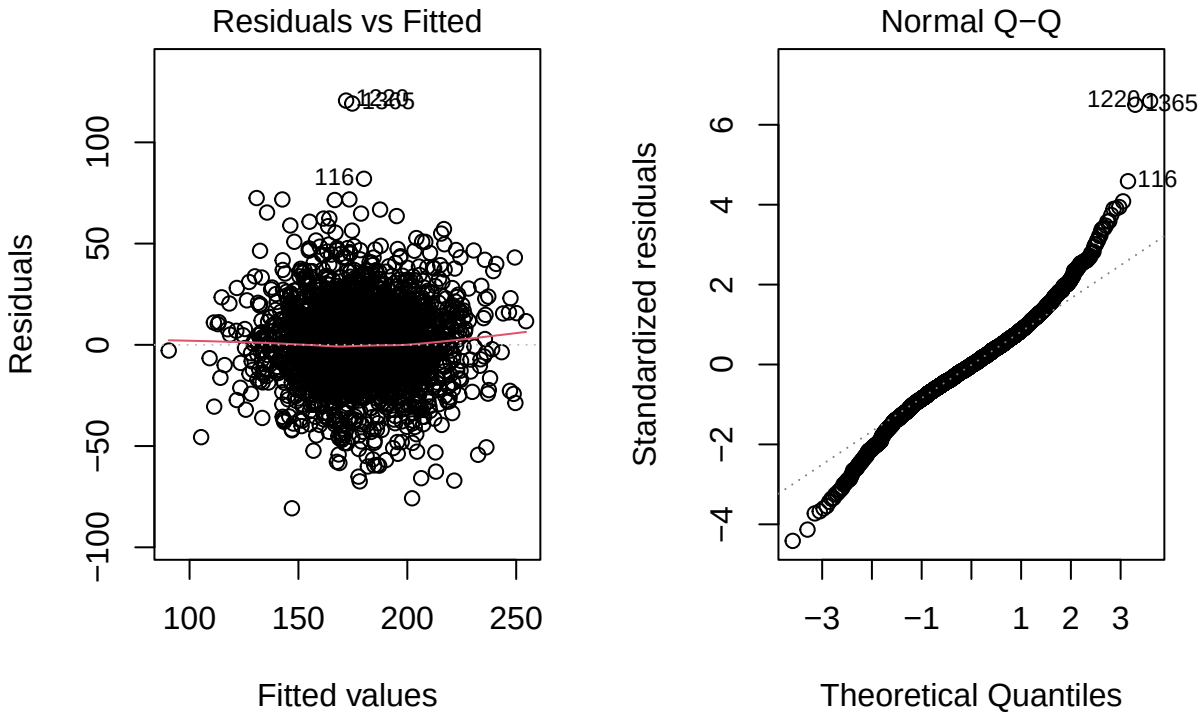


We compute the Variance Inflation Factors (VIF) for the BIC model and see that there are numerous values that are at least 5 indicating that we have multicollinearity. Specifically, we see a VIF of at least 10 in Percent Married and a VIF of 8 in Percent Married Households. As well as in log(Median Income), Percent Employed 16 and Over, Percent Employers Private Coverage and Percent Private Coverage. Due to this frequent multicollinearity between the predictors in the stepwise model this suggests that the stepwise model is not a suitable approach as multicollinearity is not taken into account and thus we proceed below with stepwise regression with the states where we encounter a similar problem and further to Ridge Regression and Lasso to reduce the effects of multicollinearity on the model.

Now we perform stepwise with states included as a factor and as can be seen in both the AIC and BIC case the factor State is kept in the model and otherwise the other variables included are similar with the main different exclusion being log(Percent Black) which was included in the model without the addition of the states.

We see from the summary output that the majority of the coefficients are significant with the departures from this coming from the State factor variable for certain states. In the AIC model we see that Poverty Percent is not significant with a p value of 0.0628, however we see in the BIC model this variable is dropped and as Poverty Percent will have a high correlation with Percent Unemployed 16 and Over we adopt the BIC model.

Below we can see the residual and QQplot for the BIC model. In the residual plot on the left hand side we see constant variance and a mean of zero with the majority of points near the horizontal red line. This is similar to what we saw when we didn't include the states in the selection process. Similarly, we see in the QQplot the majority of points on or near the reference line suggesting that the errors are indeed normally distributed, again as we see in the above case of not including states in the selection process.



In order to test whether the addition of states in the model makes a significant difference on the model in the stepwise selection process we perform an F-Test using the anova function. Computing this we get a p-value of less than $2.2x10^{-}16$, thus giving us strong evidence that we should reject the null hypothesis that the model not including states is better. Hence, the BIC model including

the states as a factor variable is a better fit of the data according to the anova function.

To complete the section on stepwise regression we perform leave one out cross validation in order to compute the R^2 of the stepwise models and also the Root Mean Squared Error. This allows us to compare these models with the Ridge and Lasso models we generate in the enxt sections. Performing this we see the AIC has an R^2 value of 0.4629, whereas in the BIC model we have an R^2 value of 0.4621.

2.3.2 RIDGE Regression

To address the multi-collinearity present in the data we assess the viability of a RIDGE Regression Model. For this we fit all continuous variables as predictors using the `glmnet` library.

The trace shows that the ridge penalisation method reduces many variables close to 0 but does not remove any variables from the model itself.

Fitting a ridge model and using the value of λ one standard error further away from the minimum does not remove any of the terms from the model and gives the following coefficients. We also include the parameter estimates when the parameters are scaled to highlight how significant the parameter is.

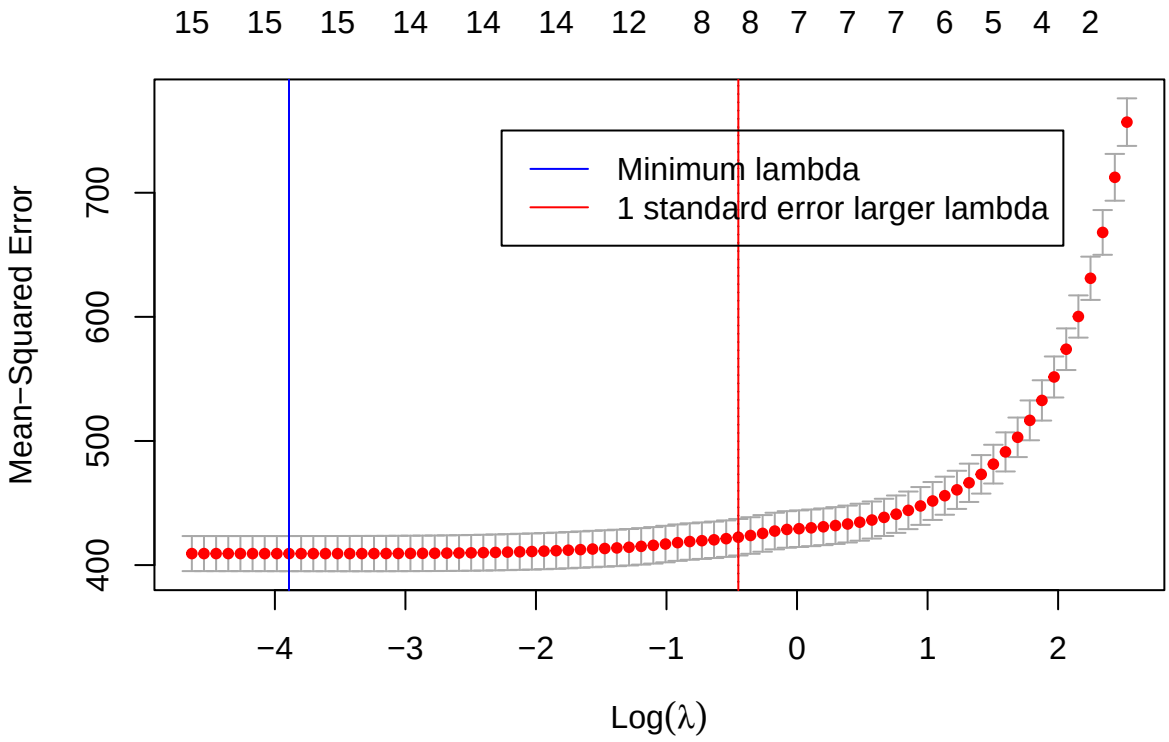
	Parameter Estimates
(Intercept)	360.246
incidenceRate	0.188
povertyPercent	0.217
MedianAgeMale	-0.126
MedianAgeFemale	-0.227
AvgHouseholdSize	-5.197
PercentMarried	0.196
PctEmployed16_Over	-0.314
PctUnemployed16_Over	0.452
PctPrivateCoverage	-0.269
PctEmpPrivCoverage	0.278
PctPublicCoverage	0.281
PctMarriedHouseholds	-0.098
Edu18_24	-11.836
logpctblack	1.071
logmedincome	-19.646

Performing a leave one out cross validation gives an R^2 statistic of 0.436. Which is similar to the previous step wise regression method and is harder to diagnose therefore we conclude that it it is not a suitable model.

2.4 Lasso Regression

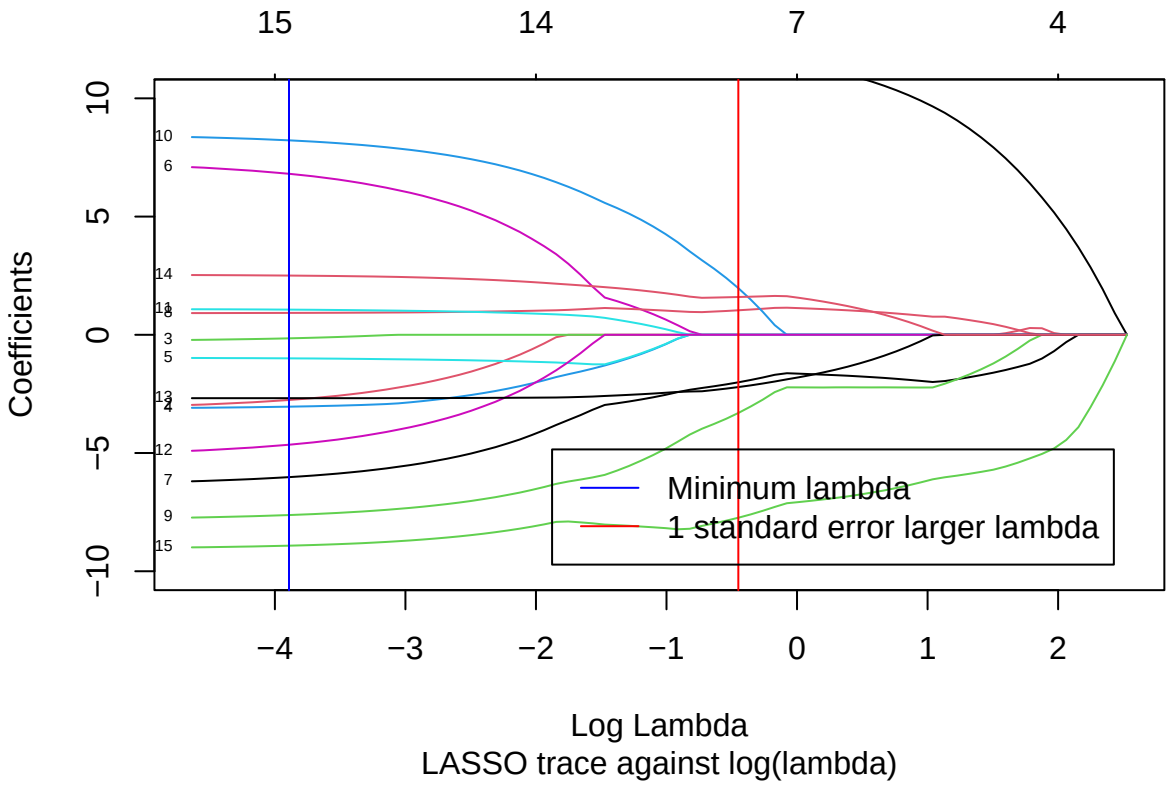
First we removed Geography because it is an id variable. We also removed binnedInc because it is measuring the same thing as medIncome. We also removed PctMarriedHouseholds and MedianAgeFemale for similar reasons. This should reduce the problem of multicollinearity. We used the log-transformed PctBlack and medIncome to improve homoscedasticity and normality.

We used cross-validation from the glmnet library which leaves out a 10th of the data every time. We produced the following plot of mean-squared error against log(lambda). We decided to use the 1-standard-error-lambda because this likely to shrink some predictor variables to zero, performing variable selection. We prefer a simpler model.




```
## [1] 0.6378303
```

We produced a plot that shows a trace of each parameter estimate for different values of log(lambda). We scaled the model for a better visual interpretation.



The Lasso regression using 1-standard-error-lambda produces the following parameter estimates.

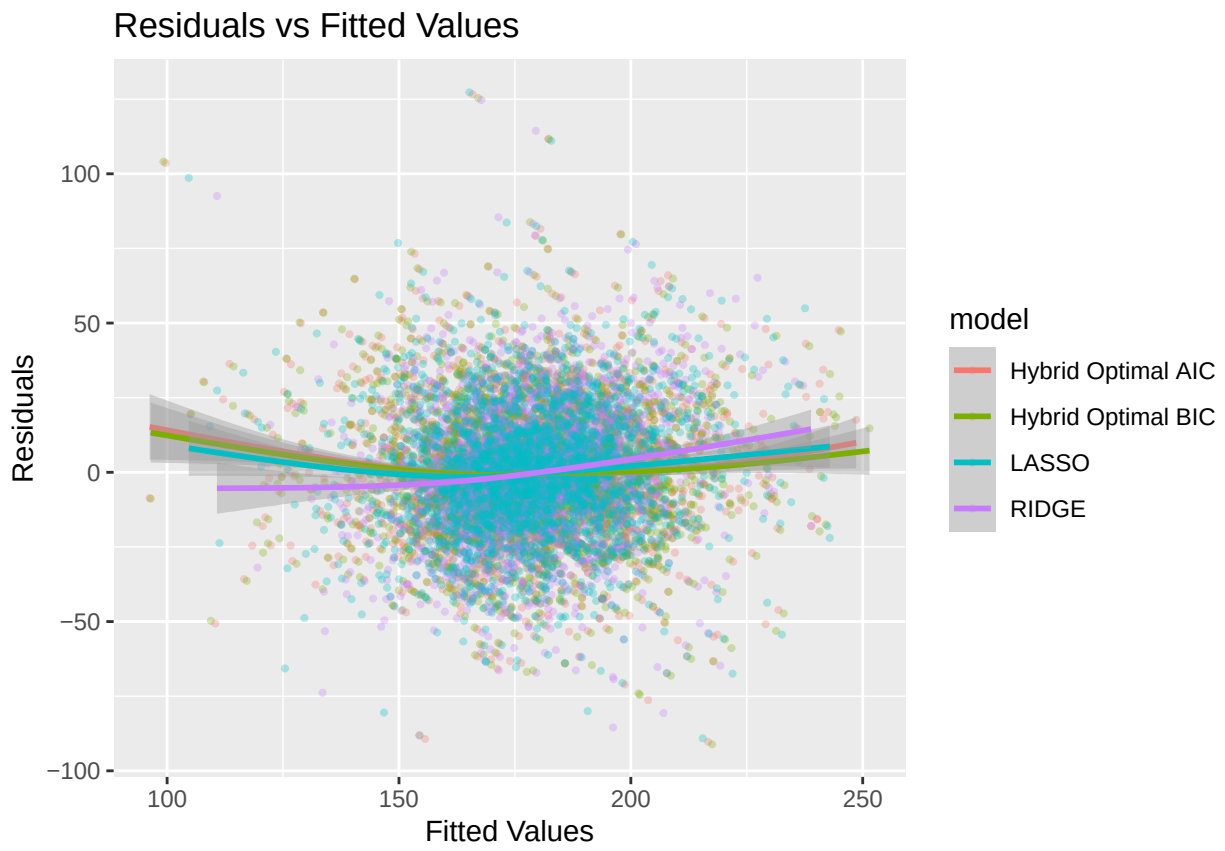
	Parameter Estimates
(Intercept)	471.168
incidenceRate	0.226
povertyPercent	0.000
MedianAgeMale	0.000
MedianAgeFemale	0.000
AvgHouseholdSize	0.000
PercentMarried	0.000
PctEmployed16_Over	-0.244
PctUnemployed16_Over	0.299
PctPrivateCoverage	-0.313
PctEmpPrivCoverage	0.210
PctPublicCoverage	0.000
PctMarriedHouseholds	0.000
Edu18_24	-10.110
logpctblack	0.906
logmedincome	-32.485

2.5 Statistical Interpretation and Validation

```
# So wanna make a plot of residuals vs fitteds
library(ggplot2)

multResidualPlot <- function(residual.list, fitted.list, models) {



  # a <- data.frame(x=fitted.list[[1]], y=residual.list[[1]], col=rep('blue', times=length(fit
  a <- data.frame()
  for (i in 1:length(models)) {
    x <- fitted.list[[i]]
    y <- residual.list[[i]]
    model <- rep(models[i], times=length(x))
```

```
    df.temp <- data.frame(x, y, model)
    colnames(df.temp) <- c("x", "y", "model")
    a <- rbind(a, df.temp)
  }
  mrp <- ggplot(a, aes(x=x, y=y, colour=model)) +
    geom_point(alpha=0.3, size=0.75) + geom_smooth() +
    labs(x="Fitted Values", y="Residuals", title="Residuals vs Fitted Values", model="Models")
  return(mrp)
}

lm.ridge.residuals <- cancer$deathRate - lm.ridgefitted
residual.list <- list(
  hybridoptimalAIC$residuals,
  hybridoptimalBIC$residuals,
  lm.ridge.residuals,
  lm.lasso.1se.residuals
)

fitted.list <- list(
  hybridoptimalAIC$fitted.values,
  hybridoptimalBIC$fitted.values,
  lm.ridgefitted,
  lm.lasso.1se.fitted
)
mrp <- multResidualPlot(residual.list, fitted.list,
                c("Hybrid Optimal AIC", "Hybrid Optimal BIC", "RIDGE", "LASSO"))
mrp
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



3 References

4 Appendix

```
library(dplyr)
library(car)
library(tidyr)
library(glmnet)
## Included Libraries
# For pipe operator and general mutation
load('cancer.rdata')


# Cook's distance Plot
par(mfrow=c(1,2))
plot(lm(deathRate ~ incidenceRate,data=cancer),4)
```

```
plot(lm(deathRate ~ .,data=cancer[-c(1,4)]),4)
# Removing outlier incidence rates 'Williamsburg City, Virgin850ia'
cancer <- filter(cancer, incidenceRate <= 850)
# Scale average household sizes that are less than 1 by 100
cancer[which(cancer$AvgHouseholdSize < 1), ]$AvgHouseholdSize <-
  100*cancer[which(cancer$AvgHouseholdSize < 1), ]$AvgHouseholdSize
hist(cancer$AvgHouseholdSize, breaks=30, xlab="AvgHouseholdSize", main="Histogram of AvgHouseho
# Impute the missing data seen in the dataset
mod1=lm(PctEmployed16_Over~+incidenceRate+medIncome+binnedInc+povertyPercent+MedianAgeMale+Medi
missdf = cancer[which(is.na(cancer$PctEmployed16_Over)==TRUE),]
imputed = predict(mod1,missdf)
cancer[which(is.na(cancer$PctEmployed16_Over)==TRUE),"PctEmployed16_Over"] = imputed

# Log transforming the heavily skewed distributions of PctBlack and medIncome
cancer$logpctblack = log(cancer$PctBlack+0.05)
cancer$logmedincome = log(cancer$medIncome)
# Showing improvements in homoscedasticity in PctBlack and medIncome
par(mfrow=c(1,2))
plot(lm(deathRate~PctBlack,data=cancer),1)
plot(lm(deathRate~logpctblack,data=cancer),1)
plot(lm(deathRate~medIncome,data=cancer),1)
plot(lm(deathRate~logmedincome,data=cancer),1)
# Finding power transformation for PctEmpPrivCoverage
spreadLevelPlot(lm(deathRate~PctEmpPrivCoverage, data=cancer))
plot(lm(deathRate~PctEmpPrivCoverage,data=cancer),1)
# Below we perform stepwise regression for both AIC and BIC
# cancermodel = cancer[,-c(1,3,15)]
cancermodel <- cancer %>% select(
  !c("Geography", "medIncome", "binnedInc", "PctBlack"))
c0=lm(deathRate~1,cancermodel)
cmax=lm(deathRate~.,cancermodel)
```

```
forwardoptimalAIC = step(c0,direction="forward",
scope=list("lower"=c0,"upper"=cmax),trace=0)
backwardoptimalAIC = step(cmax,direction="backward",
scope=list("lower"=c0,"upper"=cmax),trace=0)
hybridoptimalAIC = step(c0,direction="both",
scope=list("lower"=c0,"upper"=cmax),trace=0)
forwardoptimalBIC = step(c0,direction="forward",
scope=list("lower"=c0,"upper"=cmax),trace=0,k=log(3044))
backwardoptimalBIC = step(cmax,direction="backward",
scope=list("lower"=c0,"upper"=cmax),trace=0,k=log(3044))
hybridoptimalBIC = step(c0,direction="both",
scope=list("lower"=c0,"upper"=cmax),trace=0,k=log(3044))
# We compute the summaries of the stepwise models
summary(hybridoptimalAIC)
summary(hybridoptimalBIC)
plot(hybridoptimalBIC,which=1)
# Computing the VIF of the BIC stepwise regression model
vif(hybridoptimalBIC)
# Create stepwise regression models including the states
cancermodel2 = separate(cancer,"Geography", into=c("County","State"),sep=",")[-c(1,4,5,16)]
c0=lm(deathRate~1,cancermodel2)
cmax=lm(deathRate~.,cancermodel2)
hybridoptimalAIC2 = step(c0,direction="both",
scope=list("lower"=c0,"upper"=cmax),trace=0)
hybridoptimalBIC2 = step(c0,direction="both",
scope=list("lower"=c0,"upper"=cmax),trace=0,k=log(3044))
# Produce summary and coefficients of state stepwise regression models
coef(hybridoptimalAIC2)
coef(hybridoptimalBIC2)
summary(hybridoptimalAIC2)
summary(hybridoptimalBIC2)
```

```
# Residual plots of state stepwise regression
par(mfrow=c(1,2))
plot(hybridoptimalBIC2, which=c(1,2))
# Perform F-Test on the BIC models above
anova(hybridoptimalBIC,hybridoptimalBIC2)
library(caret)
#specify the cross-validation method
#fit a regression model and use LOOCV to evaluate performance
loocv <- function(lm1, data=cancer) {
  ctrl <- trainControl(method = "LOOCV")
  xnam <- names(lm1$coefficients)[-1]
  fmla <- as.formula(paste("deathRate ~ ", paste(xnam, collapse= "+")))
  model <- train(fmla, data = data, method = "lm", trControl = ctrl)
  return(model)
}
hybridoptimalAIC.loocv <- loocv(hybridoptimalAIC, data=cancermodel)
hybridoptimalBIC.loocv <- loocv(hybridoptimalBIC,data=cancermodel)
library(glmnet)

# Create Data Matrix
cancer.dm <- cancer %>%
  select(!c("Geography", "medIncome", "binnedInc", "PctBlack", "deathRate")) %>%
  data.matrix()

lm.ridge <- glmnet(cancer.dm, cancer$deathRate, alpha=0)
lm.ridge.cv <- cv.glmnet(cancer.dm, cancer$deathRate, alpha=0)

# Create Model Plot Func
traceLogLambda <- function(lm1, lm1.cv, ylim=NULL, sub=NULL) {
  plot(lm1,"lambda",label = T, ylim=ylim)
  abline(v=log(lm1.cv$lambda.1se),col="red")
  abline(v=log(lm1.cv$lambda.min),col="blue")
```

```
  legend("bottomright",legend=c("Minimum lambda", "1 standard error larger lambda"),lty=c(1,1)
  title(sub=sub)

}

traceLogLambda(lm.ridge, lm.ridge.cv, ylim=c(-20, 1), sub="RIDGE trace against log(lambda)")

lm.ridge.1se <- glmnet(cancer.dm, cancer$deathRate, alpha = 0,
                lambda = lm.ridge.cv$lambda.1se)


lm.ridgefitted <- predict(lm.ridge.1se, newx=cancer.dm)
scatter.smooth(cancer$deathRate - lm.ridgefitted, x=lm.ridgefitted,
                xlab="Fitted", ylab="Residuals",
                sub="Residuals vs Fitted for RIDGE Model")

library(parallel)
library(foreach)
library(doParallel)
numCores <- detectCores()
registerDoParallel(numCores)

n <- dim(cancer)[1]
dev.ratios <- rep(NA, n)
dev.ratios1 <- rep(NA, n)
lm.ridge.cv <- cv.glmnet(cancer.dm, cancer$deathRate, alpha=0, nfolds=n, parallel=TRUE)

lm.ridge.cv$lambda.1se

for (i in 1:n) {
  lm.ridge.1se <- glmnet(cancer.dm[-i,], cancer$deathRate[-i], alpha = 0,
```

```
                lambda = lm.ridge.cv$lambda.1se)



  dev.ratios1[i] <- lm.ridge.1se$dev.ratio
  # lm.residuals <- predict(lm.ridge.1se, newx=cancer.dm[-i, ]) - cancer$deathRate[-i]
  # dev.ratios[i] <- 1 - sum(lm.residuals^2)/sum((cancer$deathRate[-i] -mean(cancer$deathRate[

}
mean(dev.ratios1)

library(knitr)



a <- data.frame(sapply(round(coef(lm.ridge.1se), 3), FUN=identity))
colnames(a) <- "Parameter Estimates"

# cancer.dm.scale <- cancer %>%
#   select(!c("Geography", "medIncome", "binnedInc", "PctBlack", "deathRate")) %>%
#   scale %>%
#   data.matrix()
# lm.ridge.scale.cv <- cv.glmnet(cancer.dm.scale, scale(cancer$deathRate), alpha = 0)
# lm.ridge.1se.scale <- glmnet(cancer.dm.scale, scale(cancer$deathRate), alpha = 0, lambda = l
# a$"Scaled Parameter Estimates" <-sapply(round(coef(lm.ridge.1se.scale), 3), FUN=identity)

rownames(a) <- rownames(round(coef(lm.ridge.1se), 3))
kable(a)
# cancer.lasso.dm <- cancer %>%
#   select(!c("Geography", "medIncome", "binnedInc", "PctBlack", "deathRate","MedianAgeFemale"
#   data.matrix()
cancer.lasso.dm <- cancer %>%
  select(!c("Geography", "medIncome", "binnedInc", "PctBlack", "deathRate")) %>%
```

```
  data.matrix()
set.seed((934))
lm.lasso <-glmnet(cancer.lasso.dm,cancer$deathRate,alpha=1)
lm.lasso.cv <- cv.glmnet(cancer.lasso.dm,cancer$deathRate,alpha=1)
plot(lm.lasso.cv)
abline(v=log(lm.lasso.cv$lambda.1se),col="red")
abline(v=log(lm.lasso.cv$lambda.min),col="blue")
legend("topright",legend=c("Minimum lambda", "1 standard error larger lambda"),lty=c(1,1),col=c
lm.lasso.cv$lambda.1se
lm.lasso.1se <- glmnet(cancer.lasso.dm,cancer$deathRate,lambda = lm.lasso.cv$lambda.1se,alpha=1

lm.lasso.1se.fitted <- predict(lm.lasso.1se, newx=cancer.dm)
lm.lasso.1se.residuals <- cancer$deathRate - lm.lasso.1se.fitted

lm.lasso.scaled<-glmnet(scale(cancer.lasso.dm),cancer$deathRate,alpha=1)
traceLogLambda(lm.lasso.scaled,lm.lasso.cv,ylim=c(-10,10), sub="LASSO trace against log(lambda)
b <- data.frame(sapply(round(coef(lm.lasso.1se), 3), FUN=identity))
colnames(b) <- "Parameter Estimates"
rownames(b) <- rownames(round(coef(lm.lasso.1se), 3))
kable(b)
numCores <- detectCores()
registerDoParallel(numCores)

n <- dim(cancer)[1]
lasso.dev.ratios <- rep(NA, n)
lasso.dev.ratios1 <- rep(NA, n)
lm.lasso.cv <- cv.glmnet(cancer.lasso.dm, cancer$deathRate, alpha=0, nfolds=n, parallel=TRUE)

lm.lasso.cv$lambda.1se

for (i in 1:n) {
```

```
  lm.lasso.1se <- glmnet(cancer.dm[-i,], cancer$deathRate[-i], alpha = 0,
                lambda = lm.lasso.cv$lambda.1se)



  lasso.dev.ratios1[i] <- lm.ridge.1se$dev.ratio
  # lm.residuals <- predict(lm.ridge.1se, newx=cancer.dm[-i, ]) - cancer$deathRate[-i]
  # dev.ratios[i] <- 1 - sum(lm.residuals^2)/sum((cancer$deathRate[-i] -mean(cancer$deathRate[

}
mean(lasso.dev.ratios1)

for (i in 1:n) {
  lm.lasso.min <- glmnet(cancer.dm[-i,], cancer$deathRate[-i], alpha = 0,
                lambda = lm.lasso.cv$lambda.min)



  lasso.dev.ratios1[i] <- lm.ridge.1se$dev.ratio
  # lm.residuals <- predict(lm.ridge.1se, newx=cancer.dm[-i, ]) - cancer$deathRate[-i]
  # dev.ratios[i] <- 1 - sum(lm.residuals^2)/sum((cancer$deathRate[-i] -mean(cancer$deathRate[

}
mean(lasso.dev.ratios1)




# So wanna make a plot of residuals vs fitteds
library(ggplot2)

multResidualPlot <- function(residual.list, fitted.list, models) {
```

```
  # a <- data.frame(x=fitted.list[[1]], y=residual.list[[1]], col=rep('blue', times=length(fit
  a <- data.frame()
  for (i in 1:length(models)) {
    x <- fitted.list[[i]]
    y <- residual.list[[i]]
    model <- rep(models[i], times=length(x))

    df.temp <- data.frame(x, y, model)
    colnames(df.temp) <- c("x", "y", "model")
    a <- rbind(a, df.temp)
  }
  mrp <- ggplot(a, aes(x=x, y=y, colour=model)) +
    geom_point(alpha=0.3, size=0.75) + geom_smooth() +
    labs(x="Fitted Values", y="Residuals", title="Residuals vs Fitted Values", model="Models")
  return(mrp)
}

lm.ridge.residuals <- cancer$deathRate - lm.ridgefitted
residual.list <- list(
  hybridoptimalAIC$residuals,
  hybridoptimalBIC$residuals,
  lm.ridge.residuals,
  lm.lasso.1se.residuals
)

fitted.list <- list(
  hybridoptimalAIC$fitted.values,
  hybridoptimalBIC$fitted.values,
  lm.ridgefitted,
  lm.lasso.1se.fitted
)
```

```
mrp <- multResidualPlot(residual.list, fitted.list,
                  c("Hybrid Optimal AIC", "Hybrid Optimal BIC", "RIDGE", "LASSO"))
mrp
library(plotly)
### INTERACTIVE RESIDUAL
ggplotly(mrp)
```