```r
#Read in the total_crime dataset which contains the LAD values, year, and count
totCrime <- read.csv("total_crime_LAD_year.csv")
totCrime <- totCrime[,-1] #Drop first column, contains an index

#Read in the counts of type
totType <- read.csv("type_LAD_year.csv")
totType <- totType[,-1] #Drop first column, contains an index

#Get LAD/Year for data used in maps
shape <- read.csv("shape.csv")
#Rename column to match other files
names(shape)[names(shape)=="District"] <- "LAD_name"

#Load unemployment data
unemp <- read.csv("UnemploymentLAD.csv")
#Rename columns we are going to use to start
names(unemp)[names(unemp)=="local.authority..district...unitary..prior.to.April.2015."] <- "LAD_name"
names(unemp)[names(unemp)=="Date"] <- "Year"
names(unemp)[names(unemp)=="Unemployment.rate...aged.16.64"] <- "Unemp16to64"
names(unemp)[names(unemp)=="Denominator"] <- "Pop"

#Get rid of some of the extra columns
unemp <- unemp[,-grep("(Conf|Numerator|Denominator)",names(unemp))]

#Try the first regression
#Limit Unemployment data file to just the variables that we need
reg1.unemp <- unemp[,names(unemp) %in% c("LAD_name","Year","Unemp16to64","Pop")]
#Perform merge of unemployment data and crime data
reg1.data <- merge(totCrime, reg1.unemp, by=c("LAD_name","Year"), all=TRUE)
#Perform merge of merged unemp/crime and the shape file for maps
reg1.data <- merge(shape, reg1.data, by=c("LAD_name","Year"), all.x=TRUE)
#Remove observations with weird characters frm Unemp16to64
reg1.data <- reg1.data[!(reg1.data$Unemp16to64 %in% c("!","-")),]
#Change variable formats as needed
reg1.data$Year <- as.factor(reg1.data$Year)
reg1.data$Unemp16to64 <- as.numeric(levels(reg1.data$Unemp16to64))[reg1.data$Unemp16to64]

## Warning:  NAs introduced by coercion

reg1.data$Pop <- as.numeric(levels(reg1.data$Pop))[reg1.data$Pop]

## Warning:  NAs introduced by coercion

#First regression done
reg1 <- lm(count ~ Year + Unemp16to64 + Pop, data=reg1.data)
summary(reg1)

##
## Call:
## lm(formula = count ~ Year + Unemp16to64 + Pop, data = reg1.data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -15415  -1920   -228   1604  45249
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.935e+03  3.656e+02 -18.966  < 2e-16 ***
## Year2012     2.313e+03  3.217e+02   7.190 1.02e-12 ***
## Year2013     1.802e+03  3.213e+02   5.609 2.42e-08 ***
## Year2014     2.103e+03  3.300e+02   6.374 2.44e-10 ***
## Year2015     3.193e+03  3.453e+02   9.248  < 2e-16 ***
## Unemp16to64  5.614e+02  3.965e+01  14.157  < 2e-16 ***
## Pop          1.427e-01  1.466e-03  97.367  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4007 on 1503 degrees of freedom
## Multiple R-squared:  0.891,Adjusted R-squared:  0.8906
## F-statistic:  2047 on 6 and 1503 DF,  p-value: < 2.2e-16

#Get asterisks from regression
reg1sum <- summary(reg1)
pvals <- coef(reg1sum)[,colnames(coef(reg1sum))=="Pr(>|t|)"]
names(pvals) <- rownames(coef(reg1sum))
sig.pvals <- rep(NA,length(pvals))
sig.pvals[pvals<0.01] <- "***"
f.p <- pf(reg1sum$fstatistic[1],reg1sum$fstatistic[2],reg1sum$fstatistic[3],lower.tail=FALSE)
f.sig <- rep(NA,1)
f.sig[f.p<0.01] <- "***"
sig.pvals <- c(sig.pvals,f.sig)
names(sig.pvals) <- c(names(pvals),"fstat")
```

```
set.seed(120587)

#Model validation
#Use cross-validation
k <- 10 #Number of cv folds

#Create index to identify folds
folds <- sample(1:k, nrow(reg1.data), replace=TRUE)

#Create matrix to store error values from each regression
cv.errors <- rep(NA,k)
r.squared <- rep(NA,k)
coeffs    <- matrix(NA,k,7)

#Run cross-validation for best subset selection
for (j in 1:k) {
  #Get best subset for the fold
  lm.fit <- lm(count ~ Year + Unemp16to64 + Pop, data=reg1.data[folds!=j,])
  #Get prediction for this fold and i predictors
  pred <- predict(lm.fit, reg1.data[folds==j,])
  #Save mse
  cv.errors[j] <- mean((reg1.data$count[folds==j]-pred)^2)
  r.squared[j] <- summary(lm.fit)$adj.r.squared
  coeffs[j,]   <- coefficients(lm.fit)
}
```

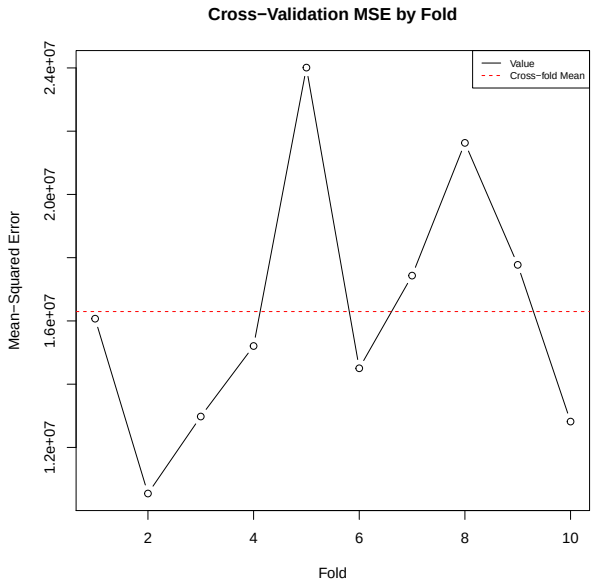
Table 1: Regression Results

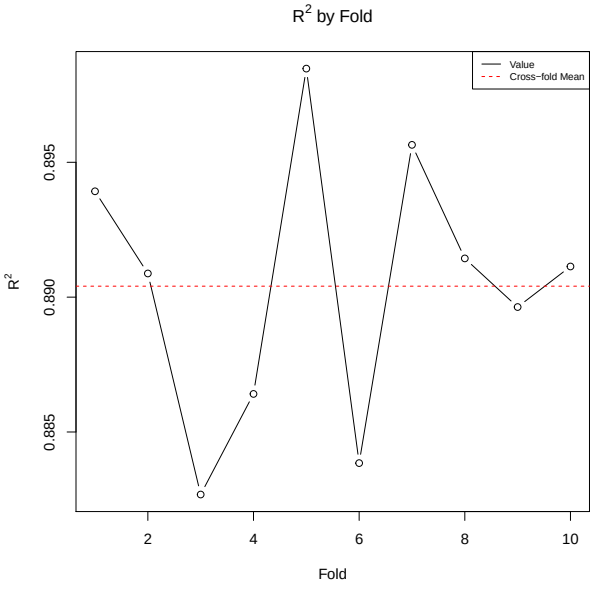
Variables	OLS
Unemployment Ages 16-64	561.398***
	(14.16)
Population Size	0.143***
	(97.37)
Year 2012	2312.862***
	(7.19)
Year 2013	1801.764***
	(5.61)
Year 2014	2103.244***
	(6.37)
Year 2015	3193.214***
	(9.25)
Adjusted R^2	0.891
F	2047.428***
N	1510

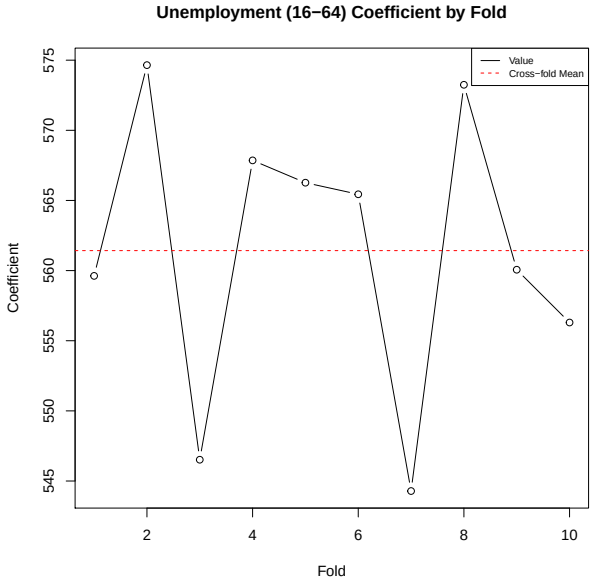
Notes: t/z-values of coefficients in parentheses,
with level of significance shown as *** = (99%), ** = (95%), and * = (90%).
Data is at the Local Authority District level and covers England.

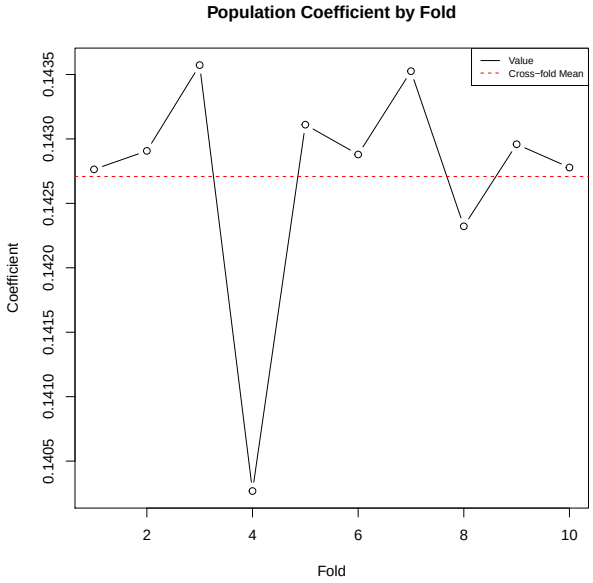
```r
colnames(coeffs) <- names(coefficients(lm.fit))

#Take average of vector to get test mse
mean.cv.errors <- mean(cv.errors)

#CV Error
plot(cv.errors, type="b",main="Cross-Validation MSE by Fold",
     xlab="Fold",
     ylab="Mean-Squared Error")
abline(h=mean.cv.errors,col="red",lty=2) #Plot average fold

legend("topright",
       legend=c("Value","Cross-fold Mean"),
       col=c("black","red"),
       lwd=1, lty=c(1,2), cex=0.7)

#R^2
plot(r.squared, type="b",main=bquote(R^2*" by Fold"),
     xlab="Fold",
     ylab=bquote(R^2))
abline(h=mean(r.squared),col="red",lty=2) #Plot average fold

legend("topright",
       legend=c("Value","Cross-fold Mean"),
       col=c("black","red"),
       lwd=1, lty=c(1,2), cex=0.7)

#Unemployment
plot(coeffs[,colnames(coeffs)=="Unemp16to64"], type="b",
     main="Unemployment (16-64) Coefficient by Fold",
     xlab="Fold",
     ylab="Coefficient")
abline(h=mean(coeffs[,colnames(coeffs)=="Unemp16to64"]),
       col="red",lty=2) #Plot average fold

legend("topright",
       legend=c("Value","Cross-fold Mean"),
       col=c("black","red"),
       lwd=1, lty=c(1,2), cex=0.7)

#Unemployment
plot(coeffs[,colnames(coeffs)=="Pop"], type="b",
     main="Population Coefficient by Fold",
     xlab="Fold",
     ylab="Coefficient")
abline(h=mean(coeffs[,colnames(coeffs)=="Pop"]),
       col="red",lty=2) #Plot average fold

legend("topright",
       legend=c("Value","Cross-fold Mean"),
       col=c("black","red"),
       lwd=1, lty=c(1,2), cex=0.7)
```

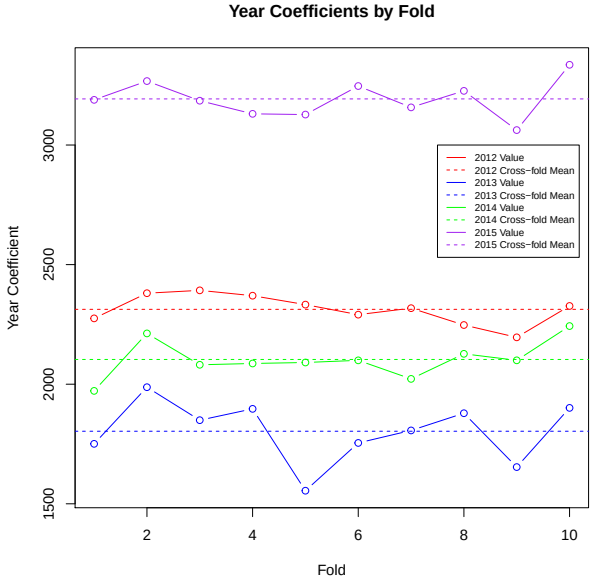
```r
ybounds <- c(min(coeffs[,colnames(coeffs)=="Year2012"],
                 coeffs[,colnames(coeffs)=="Year2013"],
                 coeffs[,colnames(coeffs)=="Year2014"],
                 coeffs[,colnames(coeffs)=="Year2015"]),
             max(coeffs[,colnames(coeffs)=="Year2012"],
                 coeffs[,colnames(coeffs)=="Year2013"],
                 coeffs[,colnames(coeffs)=="Year2014"],
                 coeffs[,colnames(coeffs)=="Year2015"]))

plot(coeffs[,colnames(coeffs)=="Year2012"], type="b",main="Year Coefficients by Fold",
     xlab="Fold",
     ylab="Year Coefficient",
     ylim=ybounds,
     col="red")
abline(h=mean(coeffs[,colnames(coeffs)=="Year2012"]),col="red",lty=2) #Plot average fold
lines(coeffs[,colnames(coeffs)=="Year2013"], type="b",col="blue")
abline(h=mean(coeffs[,colnames(coeffs)=="Year2013"]),col="blue",lty=2) #Plot average fold
lines(coeffs[,colnames(coeffs)=="Year2014"], type="b",col="green")
abline(h=mean(coeffs[,colnames(coeffs)=="Year2014"]),col="green",lty=2) #Plot average fold
lines(coeffs[,colnames(coeffs)=="Year2015"], type="b",col="purple")
abline(h=mean(coeffs[,colnames(coeffs)=="Year2015"]),col="purple",lty=2) #Plot average fold

legend(x=7.5,y=3000,
       legend=c("2012 Value","2012 Cross-fold Mean",
                "2013 Value","2013 Cross-fold Mean",
                "2014 Value","2014 Cross-fold Mean",
                "2015 Value","2015 Cross-fold Mean"),
       col=c("red","red","blue","blue","green","green","purple","purple"),
       lwd=1, lty=c(1,2,1,2,1,2,1,2), cex=0.7)
```






Unemployment (16−64) Coefficient by Fold
Value
Cross−fold Mean
Coefficient
Fold
545
550
555
560
565
570
575
2
4
6
8
10


Population Coefficient by Fold
Value
Cross−fold Mean
Coefficient
Fold
0.1405
0.1410
0.1415
0.1420
0.1425
0.1430
0.1435
2
4
6
8
10


Year Coefficients by Fold
2012 Value
2012 Cross−fold Mean
2013 Value
2013 Cross−fold Mean
2014 Value
2014 Cross−fold Mean
2015 Value
2015 Cross−fold Mean
Year Coefficient
Fold
1500
2000
2500
3000
2
4
6
8
10