```r
# Week 4, Recitation


# VIDEO 2

# Read in data
boston = read.csv("boston.csv")
str(boston)

# Plot observations
plot(boston$LON, boston$LAT)

# Tracts alongside the Charles River
points(boston$LON[boston$CHAS==1], boston$LAT[boston$CHAS==1],
 col="blue", pch=19)

# Plot MIT
points(boston$LON[boston$TRACT==3531],boston$LAT[boston
 $TRACT==3531],col="red", pch=20)

# Plot polution
summary(boston$NOX)
points(boston$LON[boston$NOX>=0.55], boston$LAT[boston$NOX>=0.55],
 col="green", pch=20)

# Plot prices
plot(boston$LON, boston$LAT)
summary(boston$MEDV)
points(boston$LON[boston$MEDV>=21.2], boston$LAT[boston$MEDV>=21.2],
 col="red", pch=20)



# VIDEO 3

# Linear Regression using LAT and LON
plot(boston$LAT, boston$MEDV)
plot(boston$LON, boston$MEDV)
latlonlm = lm(MEDV ~ LAT + LON, data=boston)
summary(latlonlm)

# Visualize regression output
plot(boston$LON, boston$LAT)
points(boston$LON[boston$MEDV>=21.2], boston$LAT[boston$MEDV>=21.2],
 col="red", pch=20)

latlonlm$fitted.values
points(boston$LON[latlonlm$fitted.values >= 21.2], boston
```

```r
 $LAT[latlonlm$fitted.values >= 21.2], col="blue", pch="$")


# Video 4

# Load CART packages
library(rpart)
library(rpart.plot)

# CART model
latlontree = rpart(MEDV ~ LAT + LON, data=boston)
prp(latlontree)

# Visualize output
plot(boston$LON, boston$LAT)
points(boston$LON[boston$MEDV>=21.2], boston$LAT[boston$MEDV>=21.2],
 col="red", pch=20)

fittedvalues = predict(latlontree)
points(boston$LON[fittedvalues>21.2], boston$LAT[fittedvalues>=21.2],
 col="blue", pch="$")

# Simplify tree by increasing minbucket
latlontree = rpart(MEDV ~ LAT + LON, data=boston, minbucket=50)
plot(latlontree)
text(latlontree)

# Visualize Output
plot(boston$LON,boston$LAT)
abline(v=-71.07)
abline(h=42.21)
abline(h=42.17)
points(boston$LON[boston$MEDV>=21.2], boston$LAT[boston$MEDV>=21.2],
 col="red", pch=20)



# VIDEO 5

# Let's use all the variables

# Split the data
library(caTools)
set.seed(123)
split = sample.split(boston$MEDV, SplitRatio = 0.7)
train = subset(boston, split==TRUE)
test = subset(boston, split==FALSE)
```

```
# Create linear regression
linreg = lm(MEDV ~ LAT + LON + CRIM + ZN + INDUS + CHAS + NOX + RM +
 AGE + DIS + RAD + TAX + PTRATIO, data=train)
summary(linreg)

# Make predictions
linreg.pred = predict(linreg, newdata=test)
linreg.sse = sum((linreg.pred - test$MEDV)^2)
linreg.sse

# Create a CART model
tree = rpart(MEDV ~ LAT + LON + CRIM + ZN + INDUS + CHAS + NOX + RM +
 AGE + DIS + RAD + TAX + PTRATIO, data=train)
prp(tree)

# Make predictions
tree.pred = predict(tree, newdata=test)
tree.sse = sum((tree.pred - test$MEDV)^2)
tree.sse



# Video 7

# Load libraries for cross-validation
library(caret)
library(e1071)

# Number of folds
tr.control = trainControl(method = "cv", number = 10)

# cp values
cp.grid = expand.grid( .cp = (0:10)*0.001)

# What did we just do?
1*0.001
10*0.001
0:10
0:10 * 0.001

# Cross-validation
tr = train(MEDV ~ LAT + LON + CRIM + ZN + INDUS + CHAS + NOX + RM +
 AGE + DIS + RAD + TAX + PTRATIO, data = train, method = "rpart",
 trControl = tr.control, tuneGrid = cp.grid)

# Extract tree
best.tree = tr$finalModel
```

```
prp(best.tree)

# Make predictions
best.tree.pred = predict(best.tree, newdata=test)
best.tree.sse = sum((best.tree.pred - test$MEDV)^2)
best.tree.sse
```