

Boston Housing

Aymen Rumi

Overview

We will investigate data from the Boston Housing Market. We have data from 14 variables for individual housing units. Our goal is to investigate these variables & relationships among them, to ultimately build a Machine Learning system that could predict house prices(our response variable) with high accuracy given instances of new data containing these 13 predictor variables.

Data Investigation

The 14 variables are as listed below

crim- per capita crime rate by town.

zn- proportion of residential land zoned for lots over 25,000 sq.ft.

indus- proportion of non-retail business acres per town.

chas- Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).

nox- nitrogen oxides concentration (parts per 10 million).

rm- average number of rooms per dwelling.

age- proportion of owner-occupied units built prior to 1940.

dis- weighted mean of distances to five Boston employment centres.

rad- index of accessibility to radial highways.

tax- full-value property-tax rate per \$10,000.

ptratio- pupil-teacher ratio by town.

black- $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town.

lstat- lower status of the population (percent).

medv(target variable)- median value of owner-occupied homes in \$1000s.

Import Data

```
housing<-read.csv(file = 'housing.csv',sep="")
names(housing)<-c("crim","zn","indus","chas","nox","rm","age",
                  "dis","rad","tax","ptratio","black","lstat","medv")
```

Data Visualization

We will look at distributions of our given variables & their bivariate plots with respect to the target variable (medv) to identify key variables that may act as significant features in predicting our response variable so we may build a Machine Learning system with minimal noise as input

Models that will be considering are:

1. Multiple Linear Regression

- we will look for linear relationships between variables & target variable

2. Decision Trees

- we will look for splits within the distribution of the variables that may help us identify between the 3 categories of housing prices

I have divided, the target variable (medv-housing prices), into 3 categories: low price(less than equal to \$15K), medium price(\$15K-30K), & high price (\$35K).

```
classifier =function(x)
{
  if(x<=15)
  {
    return ("low")
  }
  else if(x>15 & x<35)
  {
    return ("med")
  }
  else
  {
    return ("high")
  }
}

housing<-housing%>%mutate(class=apply(medv,classifier))

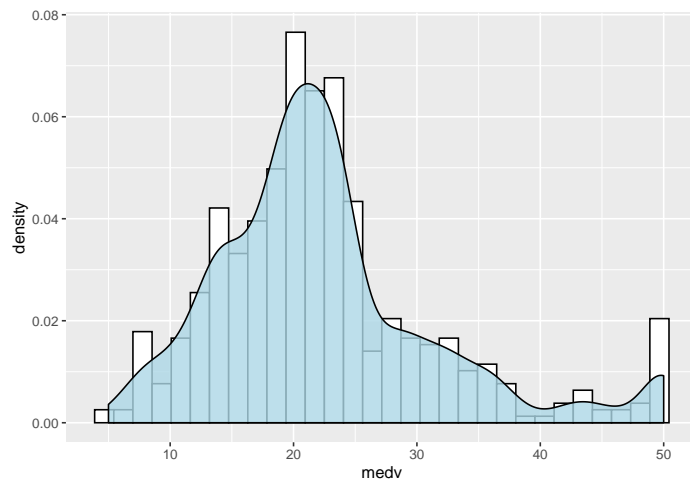
attach(housing)
```

Medv (Target Variable)

Distribution for our Target Variable, & it's Distribution divided by Categorie

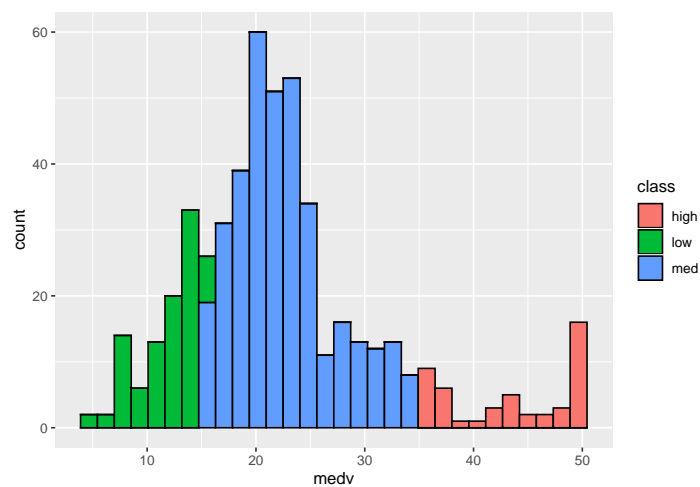
```
ggplot(housing, aes(x=medv)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.8, fill="lightblue")
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



```
ggplot(housing, aes(x=medv,fill=class)) + geom_histogram(colour="black")
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



```
count(filter(housing,class=="low"))/length(class)
```

```

n
1 0.1920792

```

```
count(filter(housing,class=="med"))/length(class)
```

```

n
1 0.7128713

```

```
count(filter(housing,class=="high"))/length(class)
```

```

n
1 0.0950495

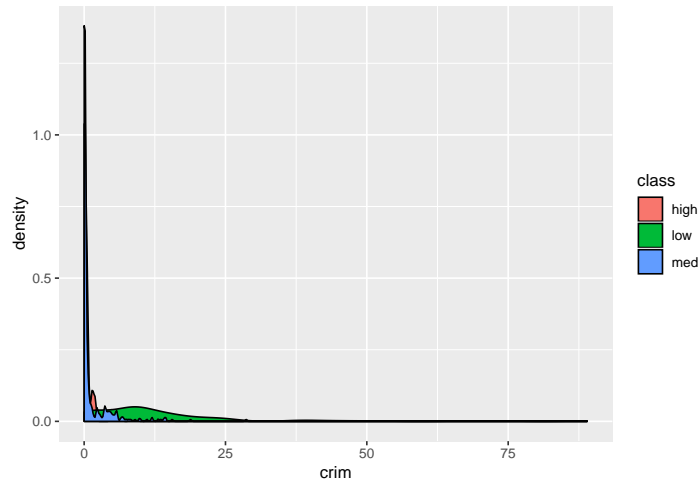
```

Observation:

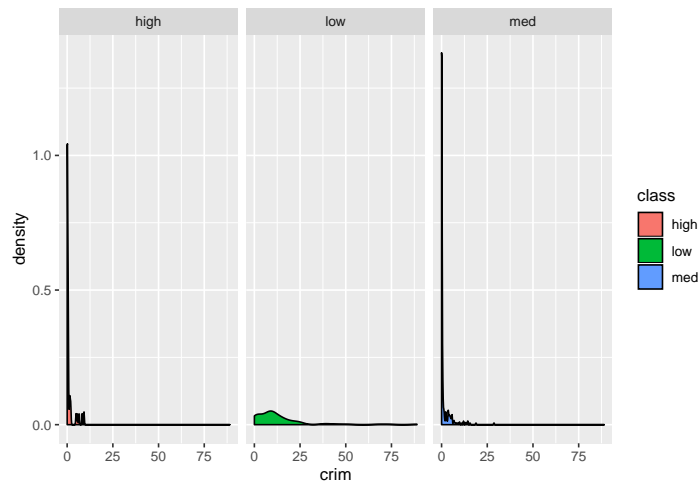
The distribution has a peak around Housing Prices of medium level(\$15K-\$35), accounting for 71% of Prices, low level(below \$15K) accounting for 19%, & high level (above \$35K) accounting for 9%

Variable # 1: Crim

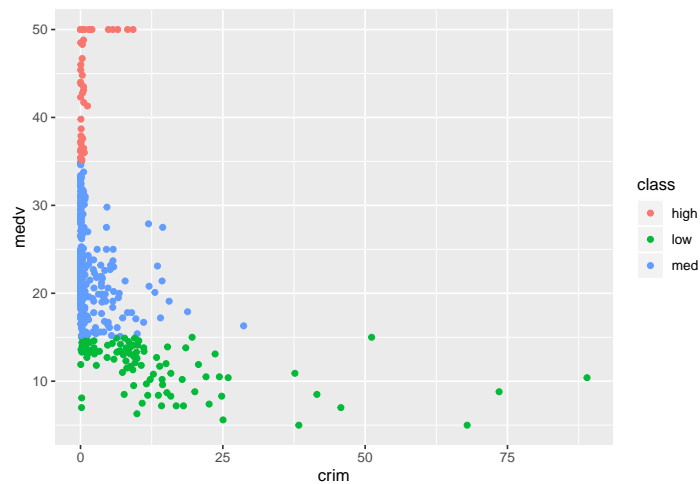
```
ggplot(housing, aes(x=crim,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=crim,fill=class)) + geom_density()+facet_wrap(~class)
```



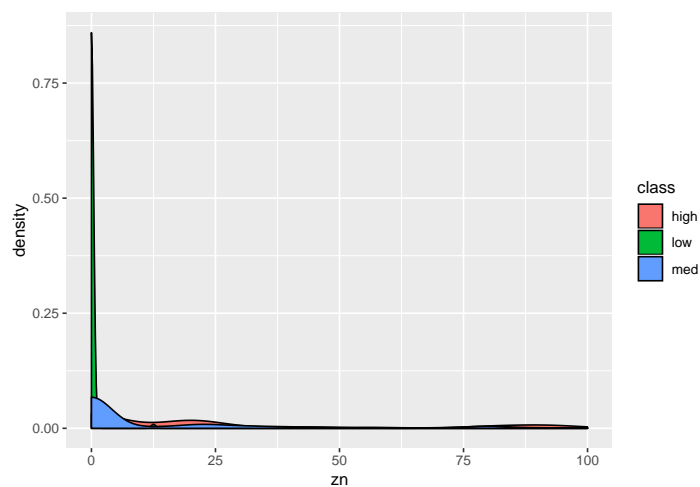
```
ggplot(housing, aes(x=crim,y=medv,color=class)) + geom_point()
```



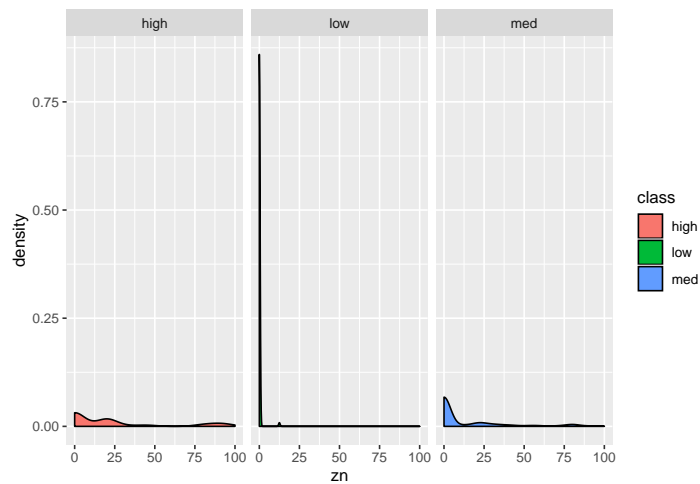
The distribution for crimes appears to have a higher spread for lower house price, and less for medium and high. Crime rates past 25% all appear to be of lower income, while those below 25% have a high concentration of all 3.

Variable # 2: Zn

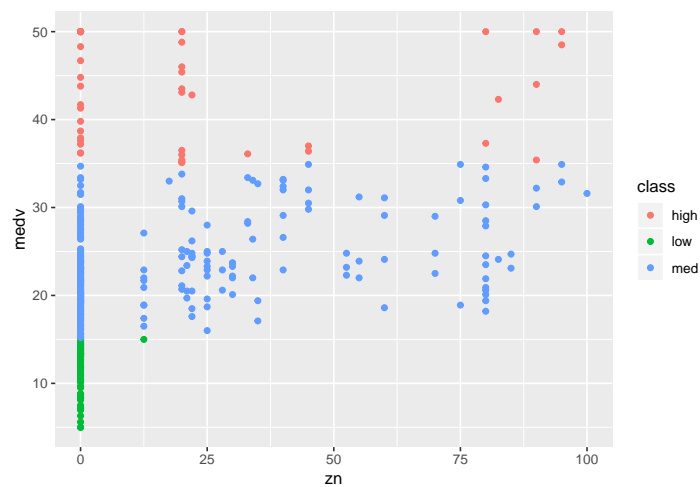
```
ggplot(housing, aes(x=zn,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=zn,fill=class)) + geom_density()+facet_wrap(~class)
```



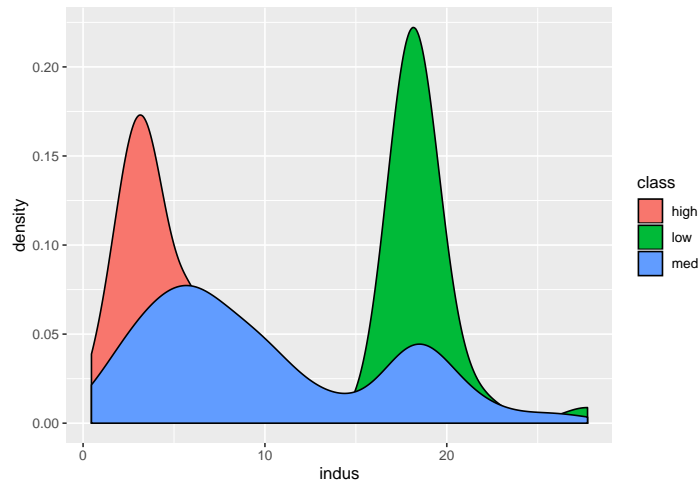
```
ggplot(housing, aes(x=zn,y=medv,color=class)) + geom_point()
```



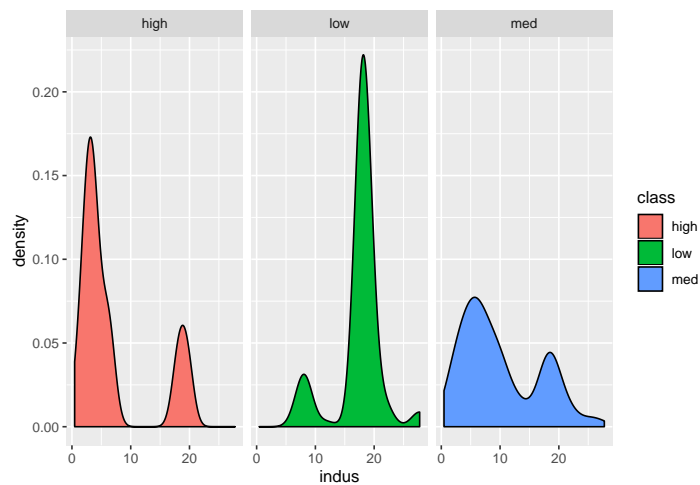
The distribution for proportion of residential land zoned for lots appears to have equal distribution for values of 0, while greater values have a blend of med & high, no apparent relationship seems to be apparent as there is a high variance in response variable (medv)

Variable # 3: Indus

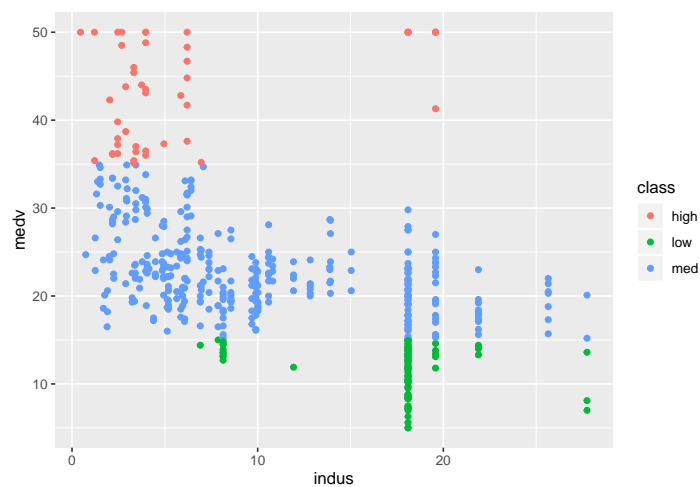
```
ggplot(housing, aes(x=indus,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=indus, fill=class)) + geom_density() + facet_wrap(~class)
```



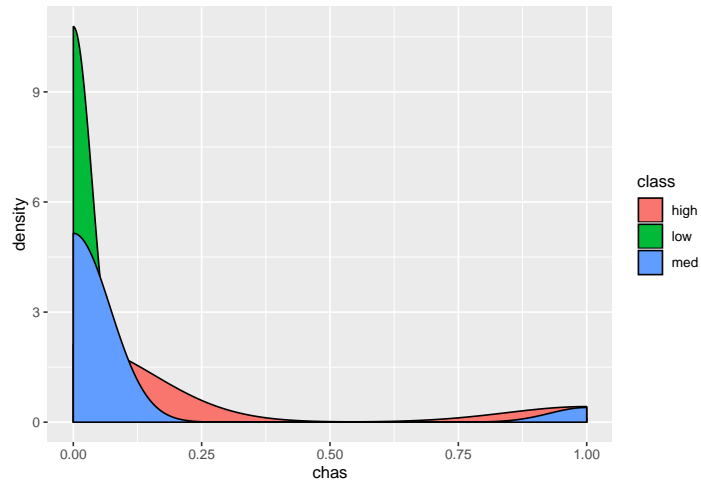
```
ggplot(housing, aes(x=indus, y=medv, color=class)) + geom_point()
```



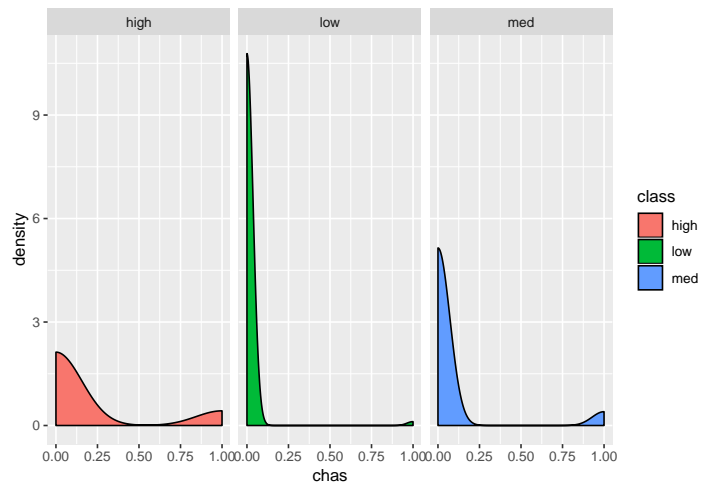
The distribution for proportion of non-retail business acres per town appears to have apparent distinctions for each categorie of house prices. There seems to be an apparent negative linear relationship, as proportion increases house prices get lower

Variable # 4: Chas

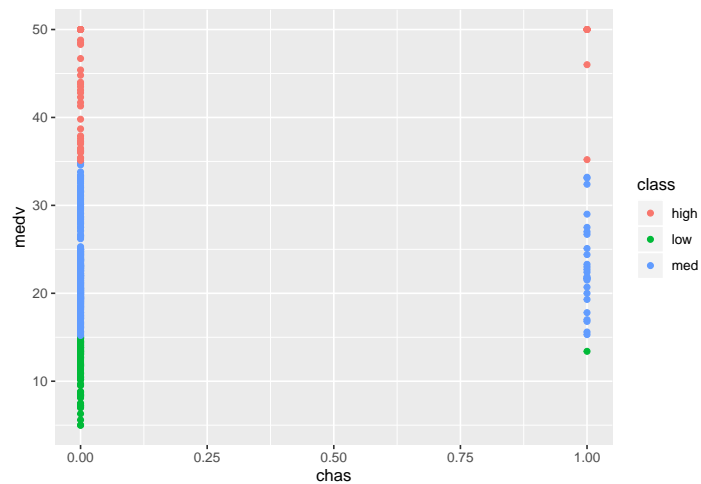
```
ggplot(housing, aes(x=chas,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=chas,fill=class)) + geom_density()+facet_wrap(~class)
```



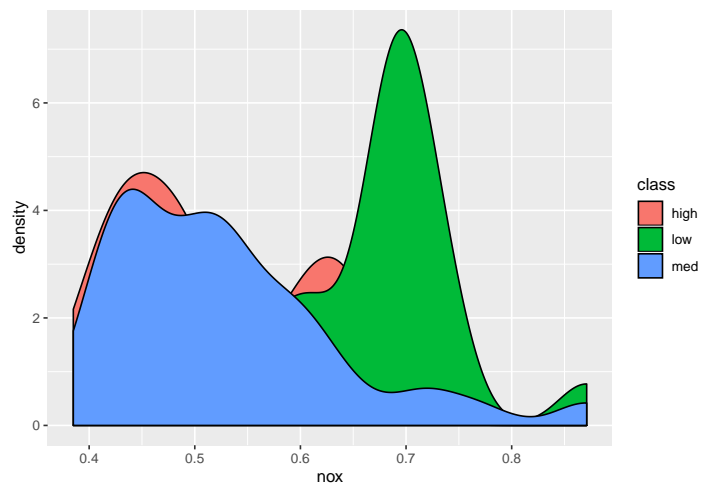
```
ggplot(housing, aes(x=chas,y=medv,color=class)) + geom_point()
```

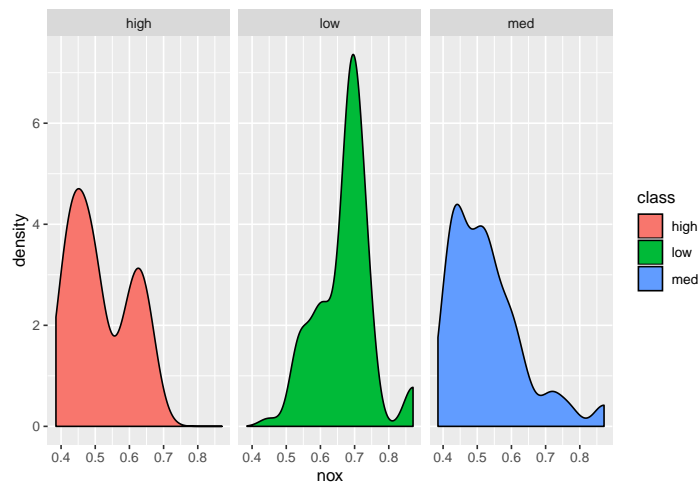
We can see that house prices are of medium range when closer to the river while those that arent are of equal distribution

Variable # 5: Nox

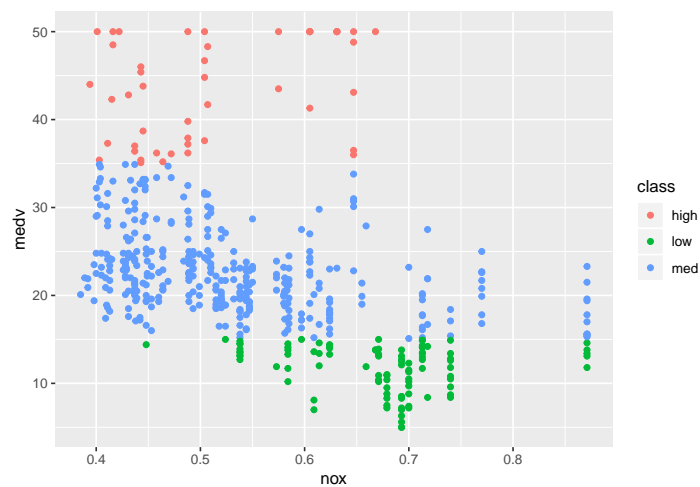
```
ggplot(housing, aes(x=nox,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=nox,fill=class)) + geom_density()+facet_wrap(~class)
```



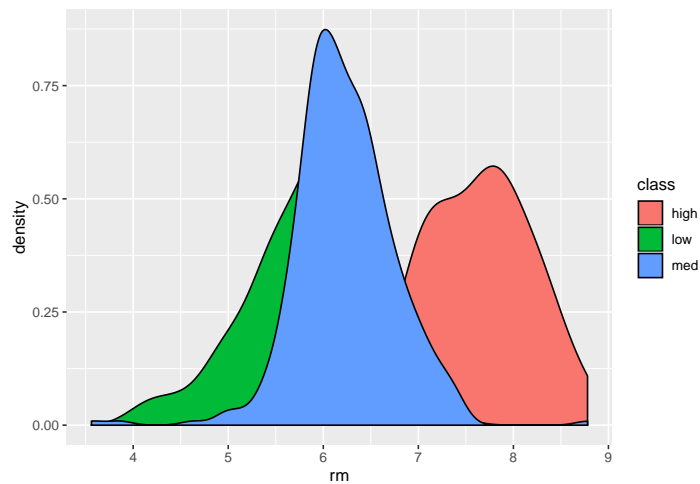
```
ggplot(housing, aes(x=nox,y=medv,color=class)) + geom_point()
```



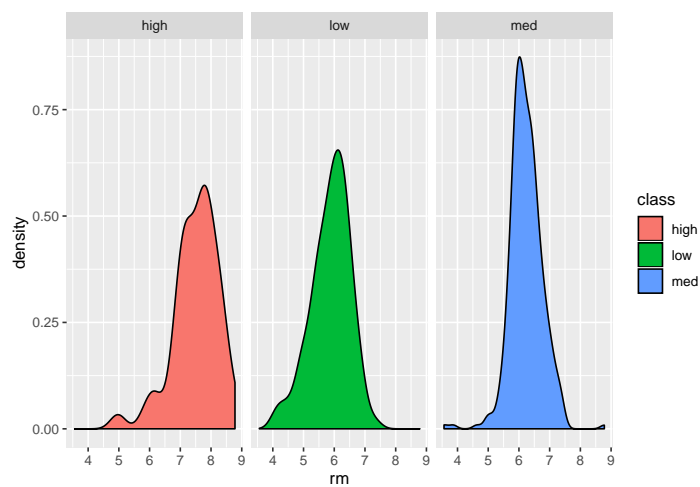
The distribution for nitrogen oxide concentration appears to have apparent distinctions for low vs med/high. There seems to be an apparent negative linear relationship, as concentration increases house prices get lower. However there seems to be high response variability for values of med and high

Variable # 6: Rm

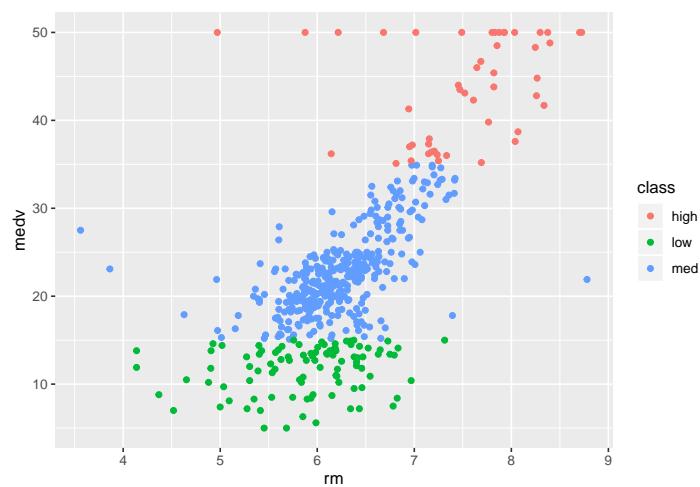
```
ggplot(housing, aes(x=rm,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=rm,fill=class)) + geom_density()+facet_wrap(~class)
```



```
ggplot(housing, aes(x=rm,y=medv,color=class)) + geom_point()
```

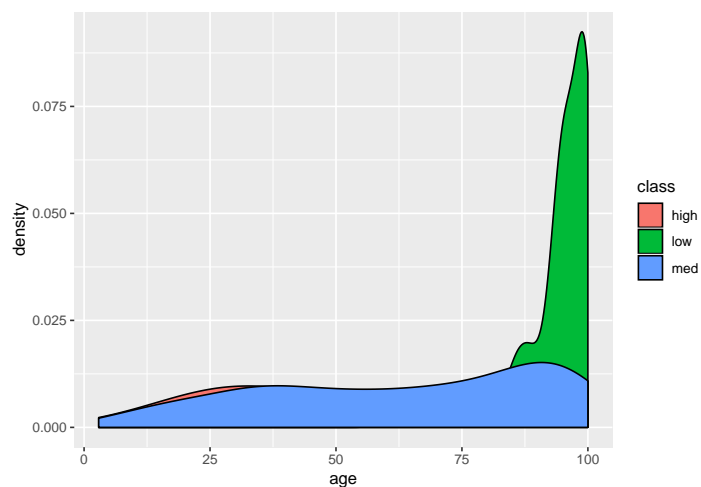


There is a high correlation between average number of rooms per dwelling & house prices, very apparent with a positive linear relationship, although there exists high variance for lower prices houses, but still follows

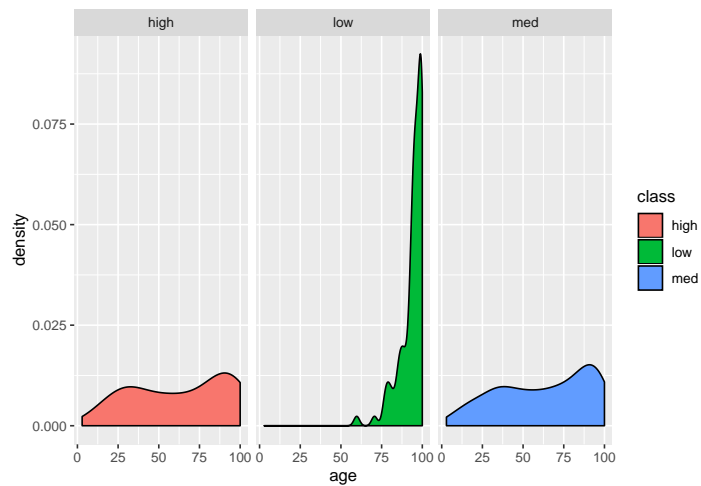
linear relationship regardless with a few exceptions that we may be able to capture with the help of other variables

Variable # 7: Age

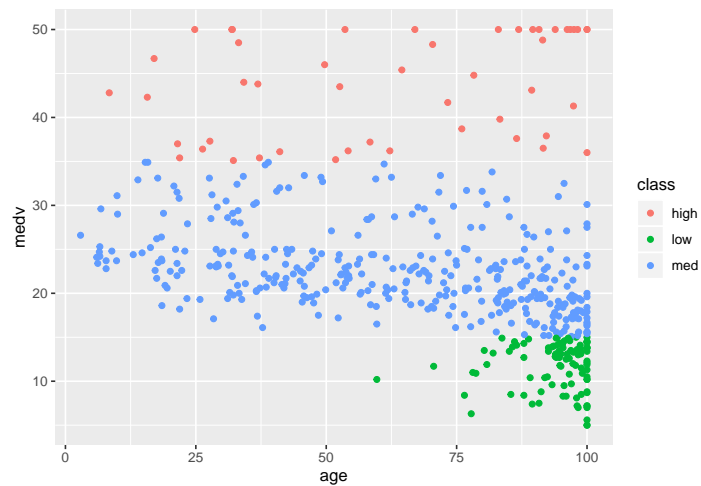
```
ggplot(housing, aes(x=age,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=age,fill=class)) + geom_density()+facet_wrap(~class)
```



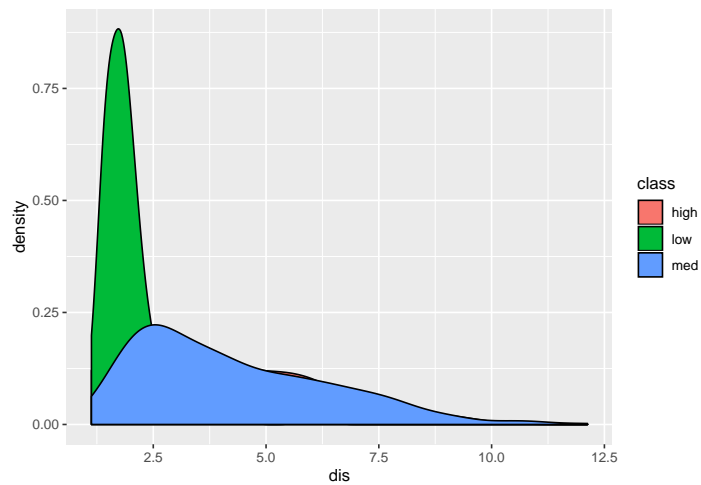
```
ggplot(housing, aes(x=age,y=medv,color=class)) + geom_point()
```



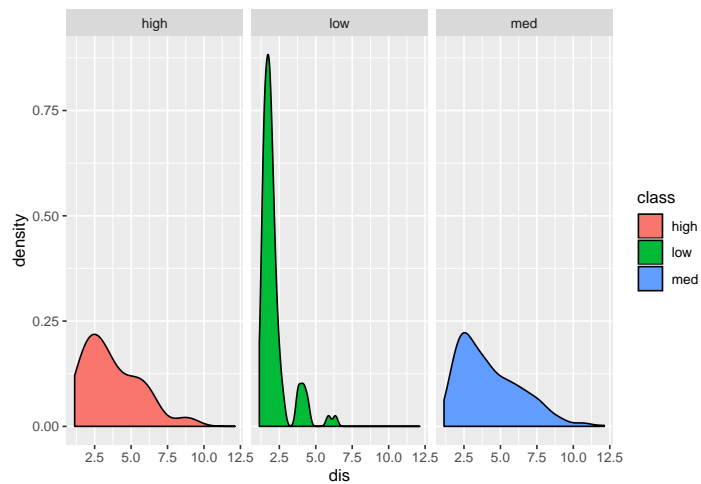
The distributions shows all low priced houses being of old age, but distributions for med & low are the same distribution with high variance in response and predictor

Variable # 8: Dis

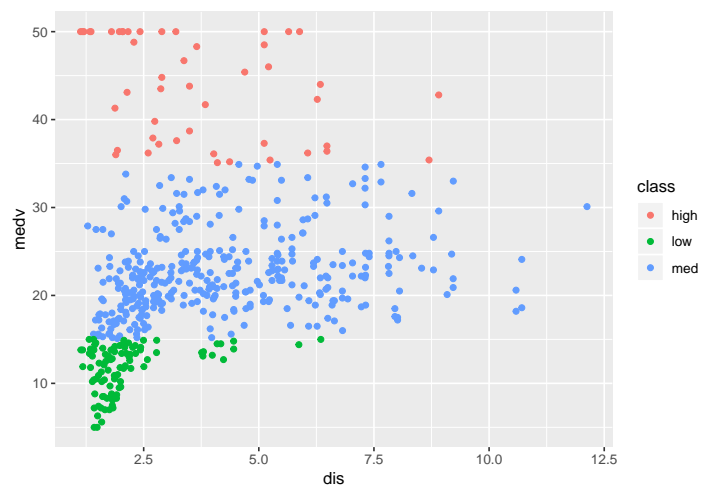
```
ggplot(housing, aes(x=dis, fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=dis, fill=class)) + geom_density()+facet_wrap(~class)
```



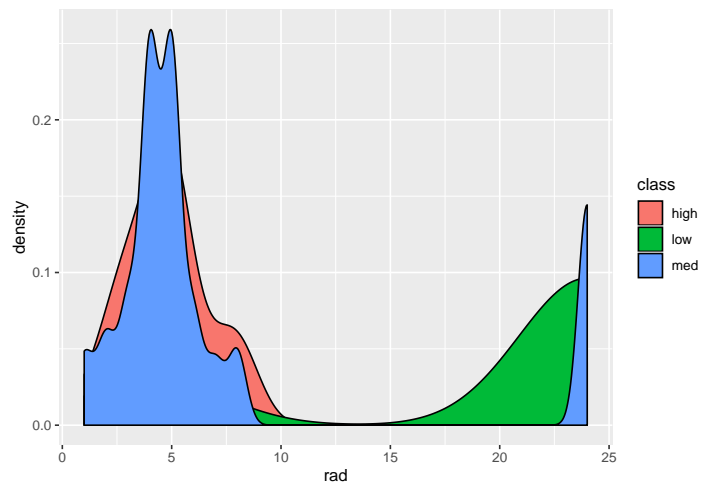
```
ggplot(housing, aes(x=dis,y=medv,color=class)) + geom_point()
```



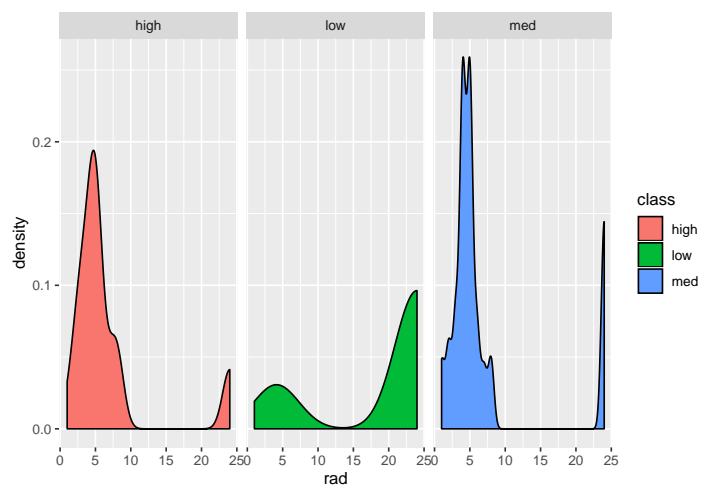
The distribution shows as weighted mean of distances to five Boston employment centres increases, the house prices are more towards the median and high side, with lower values having a mix of all 3, however the variance is quite high

Variable # 9: Rad

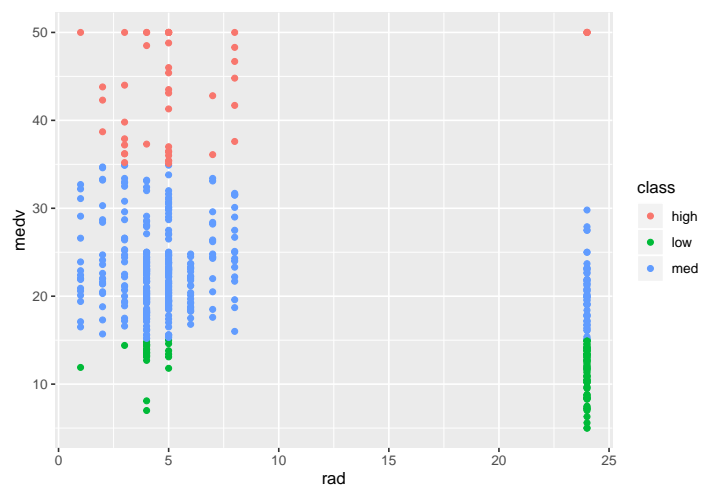
```
ggplot(housing, aes(x=rad,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=rad,fill=class)) + geom_density()+facet_wrap(~class)
```



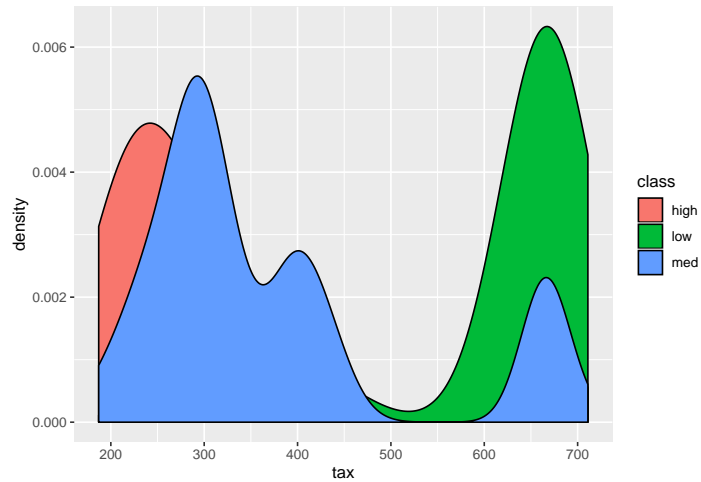
```
ggplot(housing, aes(x=rad,y=medv,color=class)) + geom_point()
```



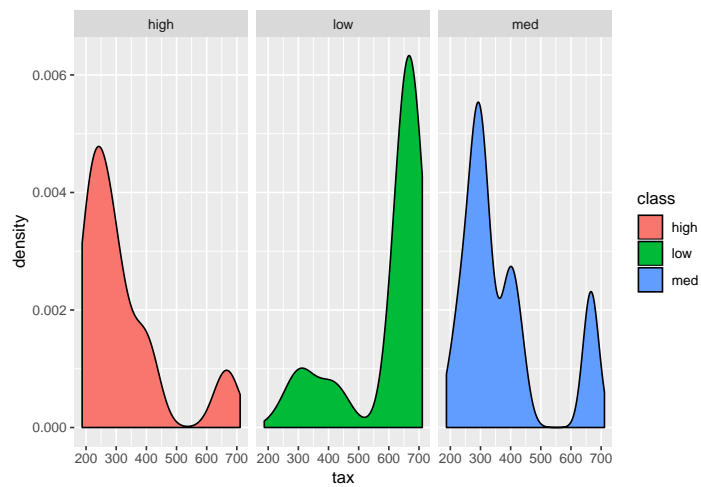
We can see that index of accessibility to radial highways, help indicate the prices of higher valued houses but fail to distinguish between low and med

Variable # 10: Tax

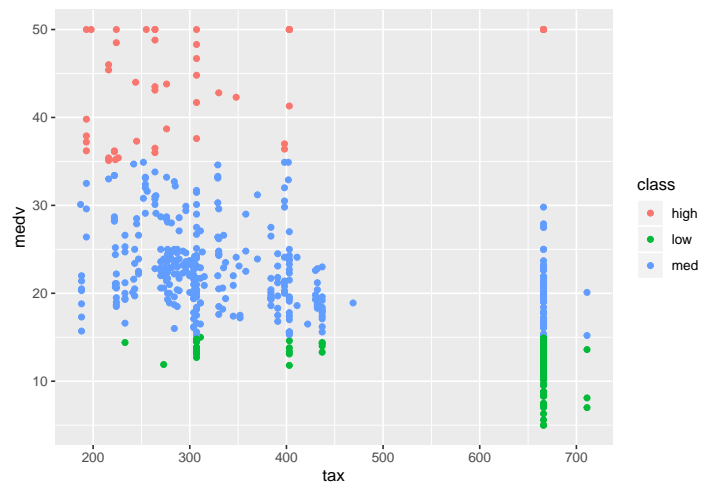
```
ggplot(housing, aes(x=tax,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=tax,fill=class)) + geom_density()+facet_wrap(~class)
```



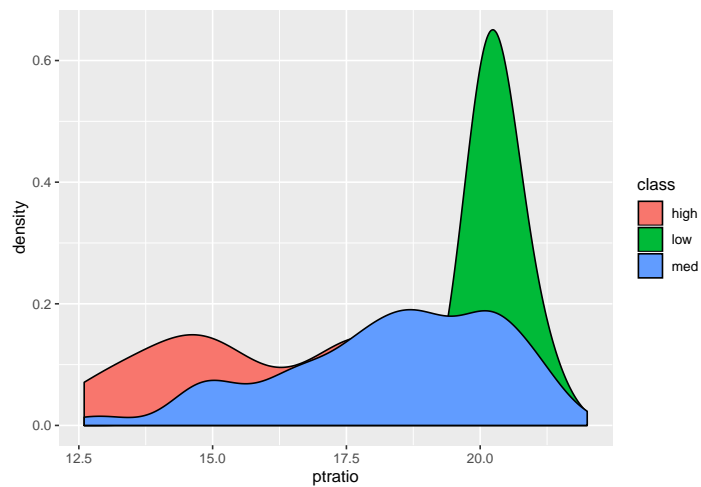
```
ggplot(housing, aes(x=tax,y=medv,color=class)) + geom_point()
```

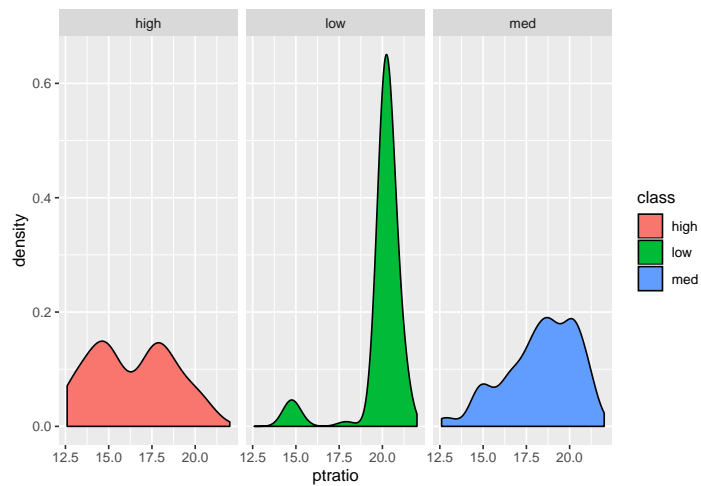
We see from the distribution that it is easy to view the distributional difference between low houses and med/high houses, and there exists large variability in med/low houses

Variable # 11: Ptratio

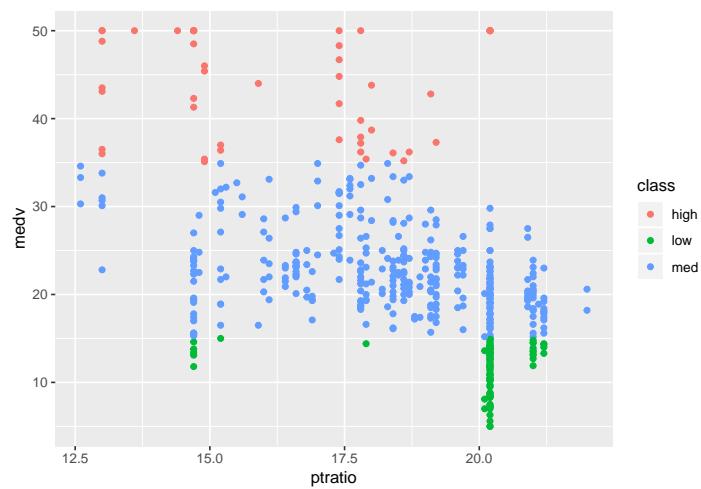
```
ggplot(housing, aes(x=ptratio, fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=ptratio, fill=class)) + geom_density()+facet_wrap(~class)
```



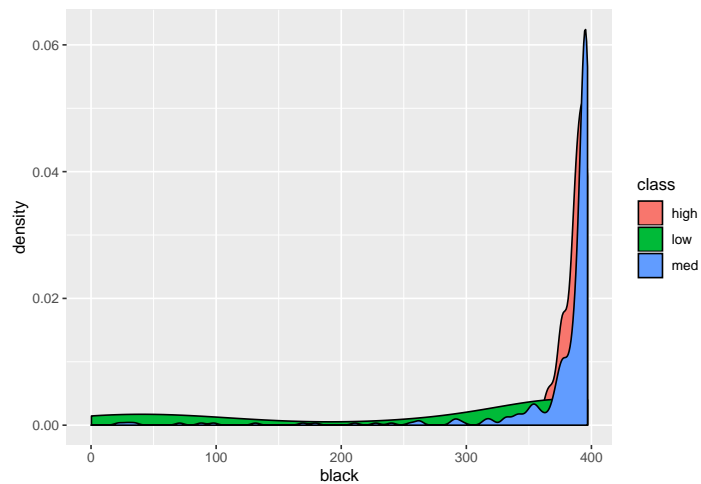
```
ggplot(housing, aes(x=ptratio, y=medv, color=class)) + geom_point()
```



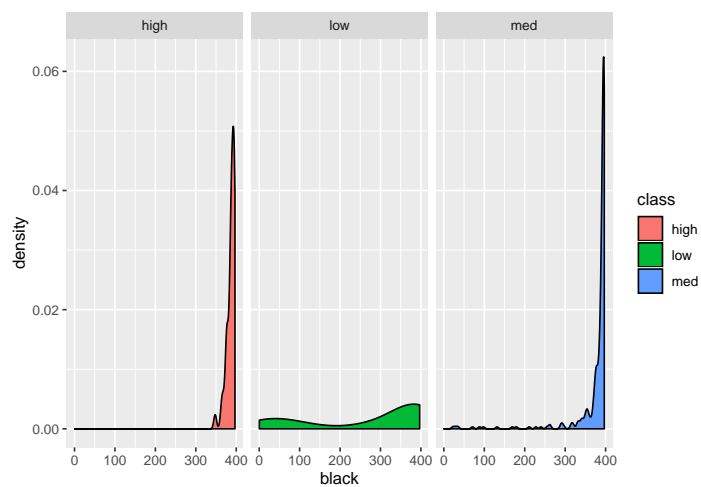
The distribution seems to have a negative linear relationship, however a very high level of variance

Variable # 12: Black

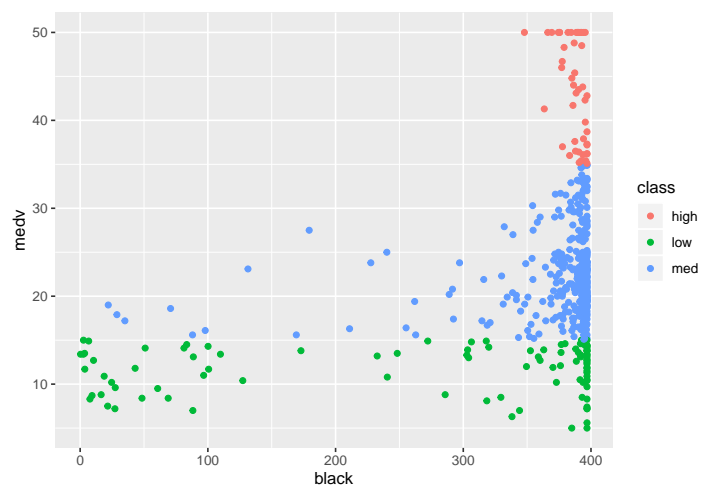
```
ggplot(housing, aes(x=black, fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=black, fill=class)) + geom_density() + facet_wrap(~class)
```



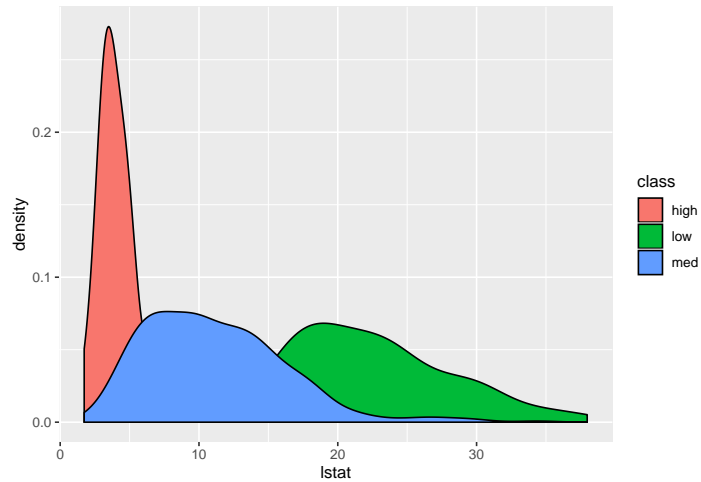
```
ggplot(housing, aes(x=black, y=medv, color=class)) + geom_point()
```



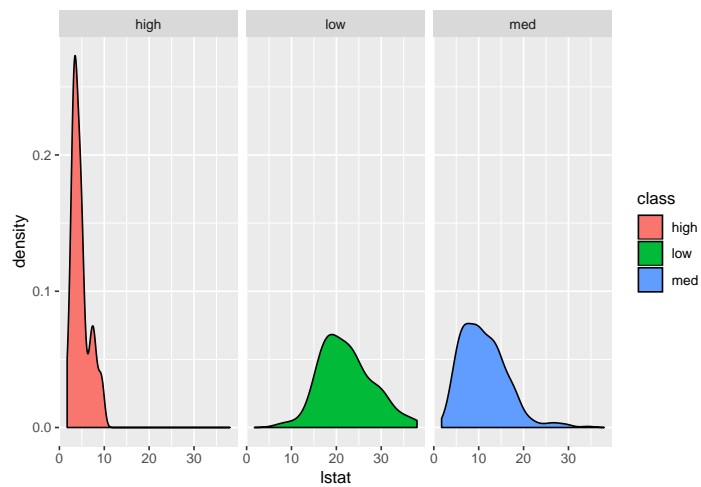
Aside from the fact that lower values tend to reflect a lower prices in houses, it seems to show no other pattern for values over 400

Variable # 13: Lstat

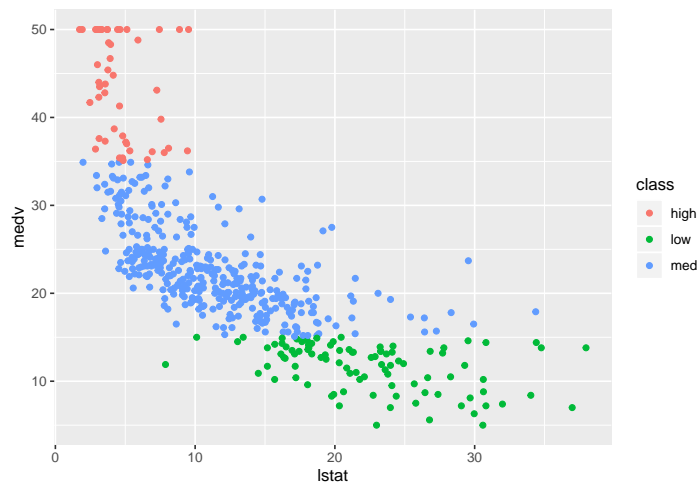
```
ggplot(housing, aes(x=lstat,fill=class)) + geom_density()
```



```
ggplot(housing, aes(x=lstat,fill=class)) + geom_density()+facet_wrap(~class)
```



```
ggplot(housing, aes(x=lstat,y=medv,color=class)) + geom_point()
```



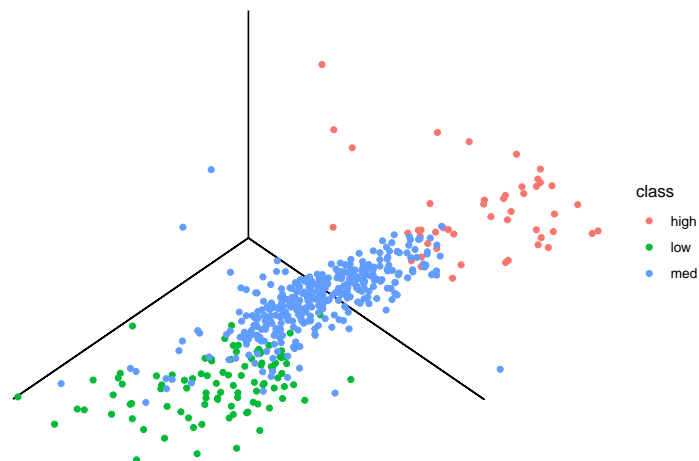
A clearly negatively linear correlation exists, with low variance comparatively as well, there seems to exist a little skew we may hope to capture

3D Plots

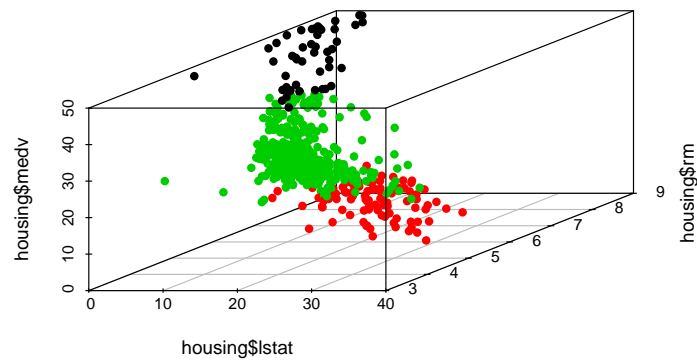
I shall plot 3 Dimensional Scatterplots of variables I believe give a strong indicator of house prices, and hypothesis on the multiple regression model to be fit

Lstat & Rm

```
ggplot(housing, aes(x=lstat, y=rm, z=medv, color=class)) +  
  theme_void() +  
  axes_3D() +  
  stat_3D()
```

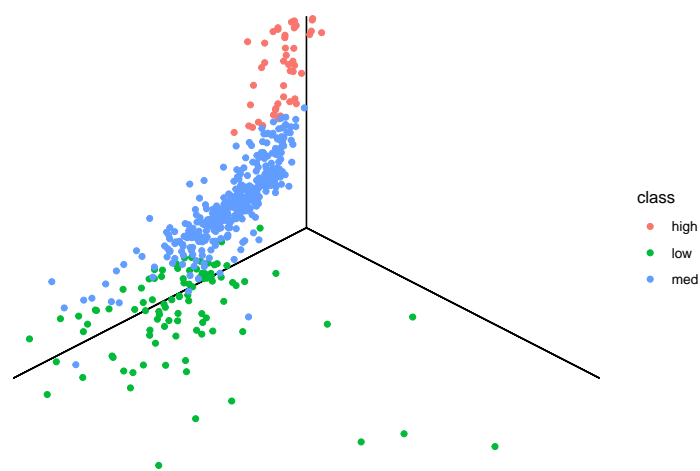


```
housing$class <- as.factor(housing$class)  
  
scatterplot3d(x=housing$lstat, y=housing$rm, z=housing$medv, pch=16, color=as.numeric(housing$class))
```

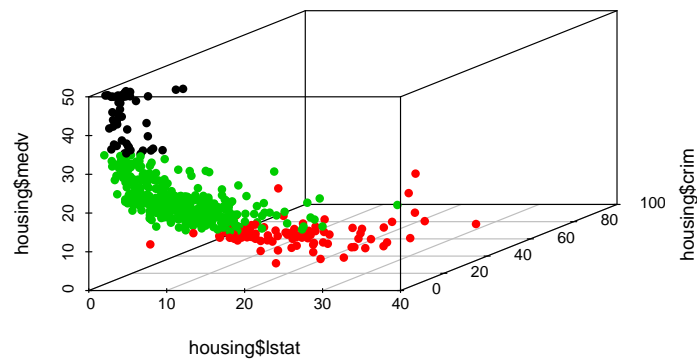


Lstat & Crim

```
ggplot(housing, aes(x=lstat, y=crim, z=medv, color=class)) +
  theme_void() +
  axes_3D() +
  stat_3D()
```

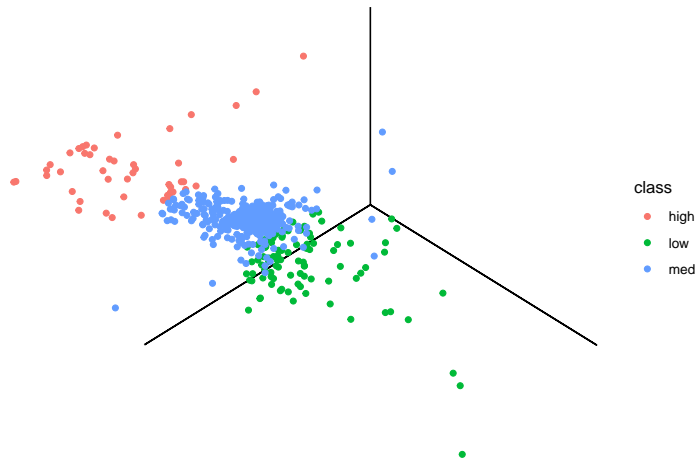


```
scatterplot3d(x=housing$lstat, y=housing$crim, z=housing$medv, pch=16, color=as.numeric(housing$class))
```

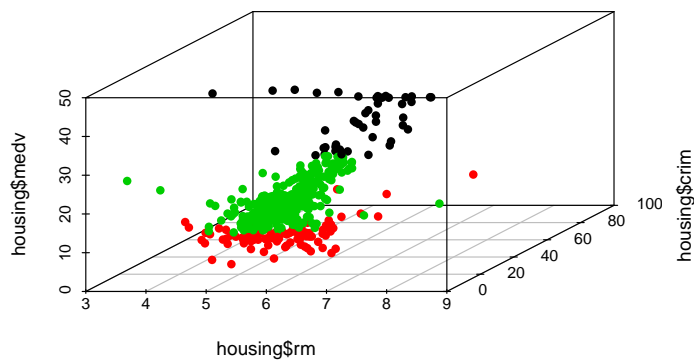


Rm & Crim

```
ggplot(housing, aes(x=rm, y=crim, z=medv, color=class)) +  
  theme_void() +  
  axes_3D() +  
  stat_3D()
```

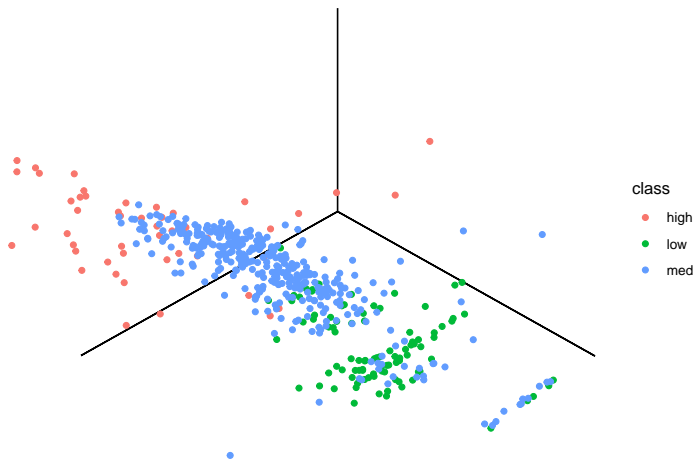


```
scatterplot3d(x=housing$rm, y=housing$crim, z=housing$medv, pch=16, color=as.numeric(housing$class))
```

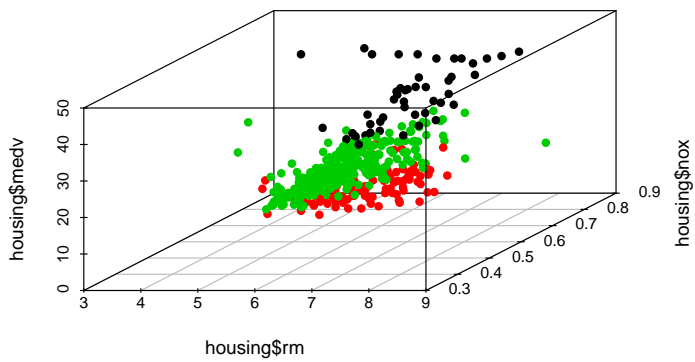


Rm & Nox

```
ggplot(housing, aes(x=rm, y=nox, z=medv, color=class)) +  
  theme_void() +  
  axes_3D() +  
  stat_3D()
```

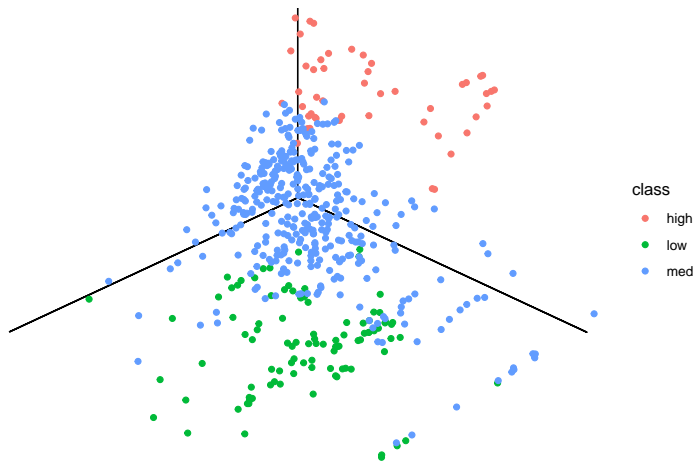


```
scatterplot3d(x=housing$rm, y=housing$nox, z=housing$medv, pch=16, color=as.numeric(housing$class))
```

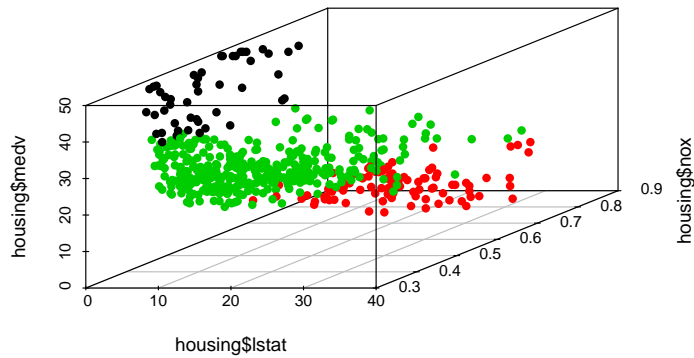


Lstat & Nox

```
ggplot(housing, aes(x=lstat, y=nox, z=medv, color=class)) +  
  theme_void() +  
  axes_3D() +  
  stat_3D()
```




```
scatterplot3d(x=housing$lstat, y=housing$nox, z=housing$medv, pch=16, color=as.numeric(housing$class))
```



Observation:

These variables studied above show clear linear relationships between the predictors and the response, the variability in Medv is explained by these inputs + Noise. We shall explore Building MLR models with them, and further investigate statistically significant variable detected beyond via R commands, this allows to view insights beyond the eye test

Multiple Linear Regression

Feature Selection

We will try out a Multiple Linear Regression Model, but first we must ensure proper features are selected for our model to reduce noise, and we shall train the model & test performance on a test set

```
housing<-housing[1:14]

model <- lm(medv ~ crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat, data = housing)

summary(model)
```

Call:

```
lm(formula = medv ~ crim + zn + indus + chas + nox + rm + age +
    dis + rad + tax + ptratio + black + lstat, data = housing)
```

Residuals:

Min	1Q	Median	3Q	Max
-15.5642	-2.7248	-0.5312	1.7687	26.1511

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	36.634941	5.102043	7.180	2.59e-12 ***
crim	-0.107417	0.032847	-3.270	0.001150 **
zn	0.046121	0.013721	3.361	0.000836 ***
indus	0.014269	0.061653	0.231	0.817071
chas	2.671108	0.861115	3.102	0.002033 **

```

nox          -17.633641    3.818719   -4.618  4.96e-06 ***
rm           3.794307     0.417835    9.081   < 2e-16 ***
age          0.001076     0.013205    0.081  0.935079
dis         -1.479179     0.199347   -7.420  5.19e-13 ***
rad          0.301534     0.066398    4.541  7.04e-06 ***
tax         -0.012053     0.003765   -3.202  0.001454 **
ptratio     -0.958874     0.130831   -7.329  9.60e-13 ***
black        0.009305     0.002684    3.467  0.000573 ***
lstat       -0.527600     0.050732  -10.400 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Residual standard error: 4.742 on 491 degrees of freedom
Multiple R-squared: 0.7415, Adjusted R-squared: 0.7346
F-statistic: 108.3 on 13 and 491 DF, p-value: < 2.2e-16

```

sample <- sample.split(medv, SplitRatio = .70)

train <- subset(housing, sample == TRUE)

test  <- subset(housing, sample == FALSE)

ggplot(train, aes(x=medv)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.8, fill="lightblue")+ggtitle("Distribution of Training Set")

```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



```

ggplot(test, aes(x=medv)) +
  geom_histogram(aes(y=..density..), colour="black", fill="white")+
  geom_density(alpha=.8, fill="lightblue")+ggtitle("Distribution of Testing Set")

```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



```
regfit.fwd=regsubsets(medv~.,data=train, nvmax=13,method="forward")

summary(regfit.fwd)
```

Subset selection object

Call: regsubsets.formula(medv ~ ., data = train, nvmax = 13, method = "forward")

13 Variables (and intercept)

	Forced in	Forced out
crim	FALSE	FALSE
zn	FALSE	FALSE
indus	FALSE	FALSE
chas	FALSE	FALSE
nox	FALSE	FALSE
rm	FALSE	FALSE
age	FALSE	FALSE
dis	FALSE	FALSE
rad	FALSE	FALSE
tax	FALSE	FALSE
ptratio	FALSE	FALSE
black	FALSE	FALSE
lstat	FALSE	FALSE

1 subsets of each size up to 13

Selection Algorithm: forward

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat
1 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
2 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
3 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
4 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
5 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
6 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
7 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
8 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
9 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
10 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
11 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
12 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "
13 (1)	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "	" "

```
test.mat=model.matrix(medv~.,data=test)

validation.errors=rep(NA,13)

for(i in 1:13){
  coefi=coef(regfit.fwd,id=i)
  pred=test.mat[,names(coefi)]%*%coefi
  validation.errors[i]=mean((test$medv-pred)^2)
}
```

```
validation.errors
```

```
[1] 36.21700 29.11124 25.94498 25.39358 23.32370 24.37883 23.77584 22.52542
[9] 22.36304 21.77664 20.65098 20.85605 20.89573
```

```
which.min(validation.errors)
```

```
[1] 11
```

```
coef(regfit.fwd,id=which.min(validation.errors))
```

(Intercept)	crim	zn	chas	nox
40.708506655	-0.101416843	0.038176087	1.915972649	-22.052695249
rm	dis	rad	tax	ptratio
3.819294696	-1.581392947	0.254704983	-0.008969070	-1.027981738
black	lstat			
0.008315321	-0.549929286			

Observation:

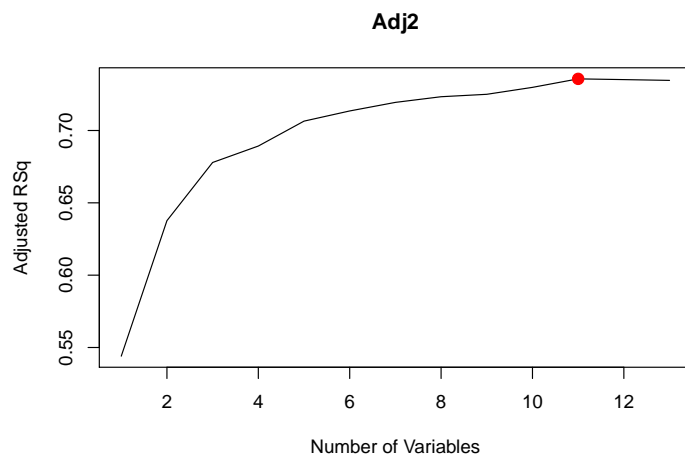
We can see that 11 of these variables have statistical significance in the model because of their contribution in explaining the variance for the variable. The variables not included are indus and age. We shall test which number of variables are most optimal with more performance measures, & the test set

```
regfit.full=regsubsets(medv~.,data=housing,nvmax=13,method="forward")
reg.summary=summary(regfit.full)

which.max(reg.summary$adjr2)
```

```
[1] 11
```

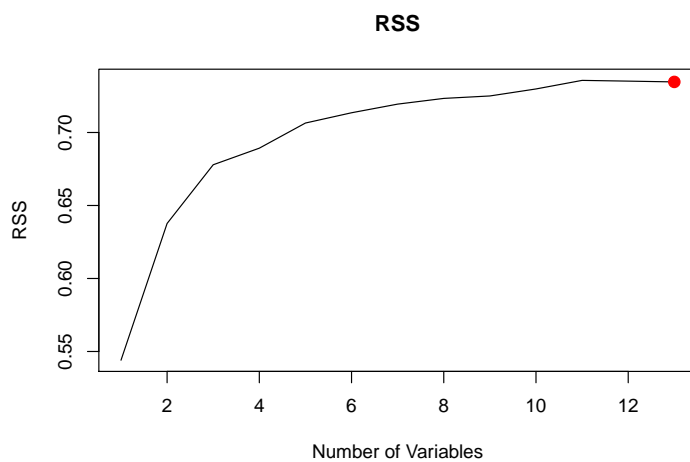
```
plot(reg.summary$adjr2,xlab="Number of Variables",
      ylab="Adjusted RSq",type="l",main="Adj2")
points(which.max(reg.summary$adjr2),reg.summary$adjr2[which.max(reg.summary$adjr2)], col="red",cex=2,pch=1)
```



```
which.min(reg.summary$rss)
```

```
[1] 13
```

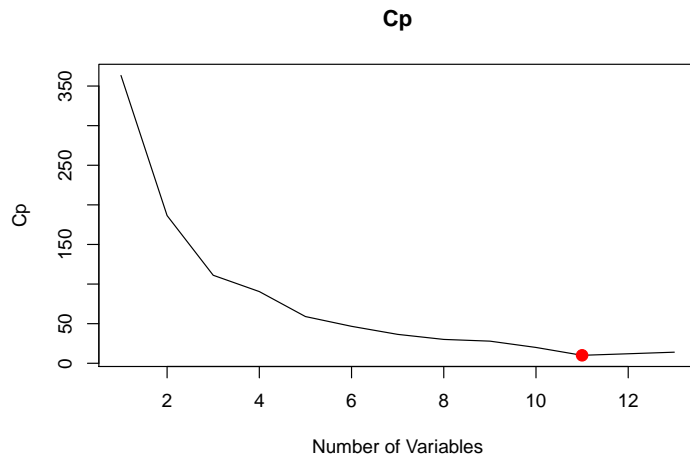
```
plot(reg.summary$adjr2,xlab="Number of Variables",
      ylab="RSS",type="l",main="RSS")
points(which.min(reg.summary$rss),reg.summary$adjr2[which.min(reg.summary$rss)], col="red",cex=2,pch=20)
```



```
which.min(reg.summary$cp)
```

```
[1] 11
```

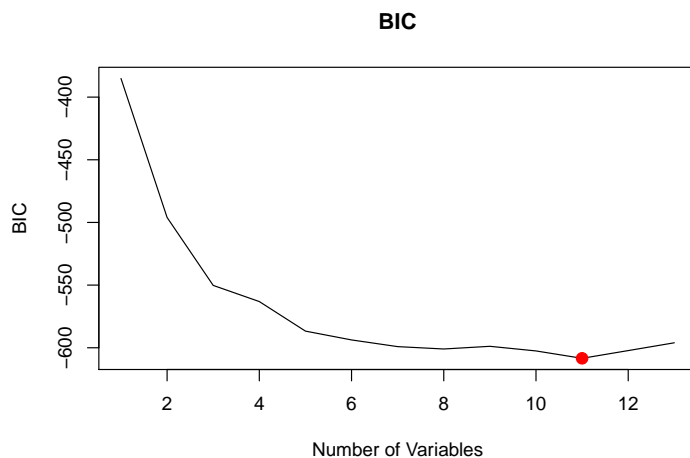
```
plot(reg.summary$cp,xlab="Number of Variables",
      ylab="Cp",type="l",main="Cp")
points(which.min(reg.summary$cp),reg.summary$cp[which.min(reg.summary$cp)], col="red",cex=2,pch=20)
```



```
which.min(reg.summary$bic)
```

```
[1] 11
```

```
plot(reg.summary$bic,xlab="Number of Variables",
      ylab="BIC",type="l",main="BIC")
points(which.min(reg.summary$bic),reg.summary$bic[which.min(reg.summary$bic)], col="red",cex=2,pch=20)
```



Observation:

We further confirm that 11 variables is the most optimal model to select for out Multiple Linear Regression, we shall now average error and compare predictions with the actual target. we will plot the distribution of our prediction

```
coefficients<-as.matrix(coef(regfit.fwd,id=which.min(validation.errors)))

predictions<-function()
{
  predictions<- c()
  for(i in 1:length(medv))
  {
```

```

    input<-housing[i,][c("crim","zn","chas","nox","rm","dis","rad","tax","ptratio","black","lstat")]
    input<-as.data.frame(c(1,input))
    input<-as.matrix(input)

    prediction<-t(coefficients) %*% t(input)

    predictions<-c(predictions,prediction)
  }

  return (predictions)
}

predictions<-as.data.frame(predictions())

targetAndPrediction<-cbind(medv,predictions)
targetAndPrediction<-as.data.frame(targetAndPrediction)
names(targetAndPrediction)<-c("target","predictions")

filter(targetAndPrediction,predictions<0)

```

```

target predictions
1  13.8 -0.09870132
2   7.0 -4.80441298

```

```

for(i in 1:length(targetAndPrediction$predictions)){

  if(targetAndPrediction$predictions[i]<0)
  {
    targetAndPrediction$predictions[i]=0
  }
}

targetAndPrediction<-targetAndPrediction%>%mutate(error=abs(target-predictions))

head(targetAndPrediction)

```

```

target predictions      error
1  21.6    25.34659 3.7465946
2  34.7    31.04083 3.6591669
3  33.4    28.96020 4.4398007
4  36.2    28.23010 7.9699010
5  28.7    25.53852 3.1614848
6  22.9    23.10215 0.2021473

```

```
mean(targetAndPrediction$error)
```

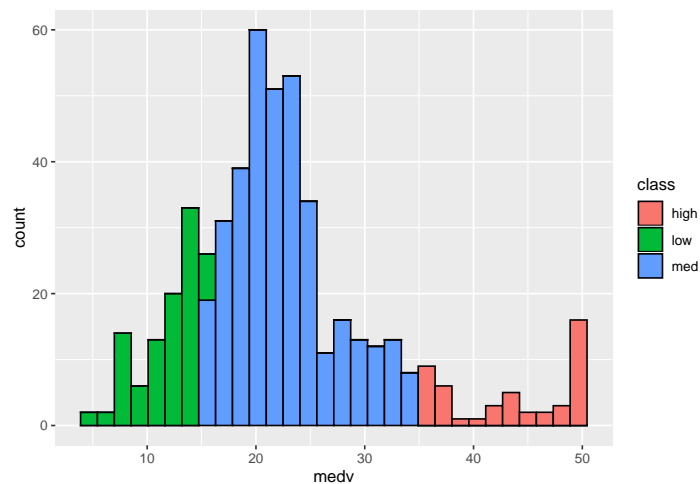
```
[1] 3.308083
```

```
filter(targetAndPrediction,error>=10)
```

	target	predictions	error
1	14.4	3.358644	11.04136
2	50.0	37.413933	12.58607
3	50.0	37.794343	12.20566
4	50.0	36.343483	13.65652
5	23.7	10.957131	12.74287
6	46.7	35.962094	10.73791
7	48.3	37.592931	10.70707
8	42.8	30.007106	12.79289
9	21.9	36.372981	14.47298
10	27.5	13.870271	13.62973
11	23.1	10.923490	12.17651
12	50.0	23.875270	26.12473
13	50.0	31.957109	18.04289
14	50.0	33.947312	16.05269
15	50.0	24.926991	25.07301
16	50.0	25.090505	24.90949
17	13.8	0.000000	13.80000
18	15.0	25.054959	10.05496
19	7.2	17.218790	10.01879
20	17.9	1.566281	16.33372
21	11.9	22.321902	10.42190

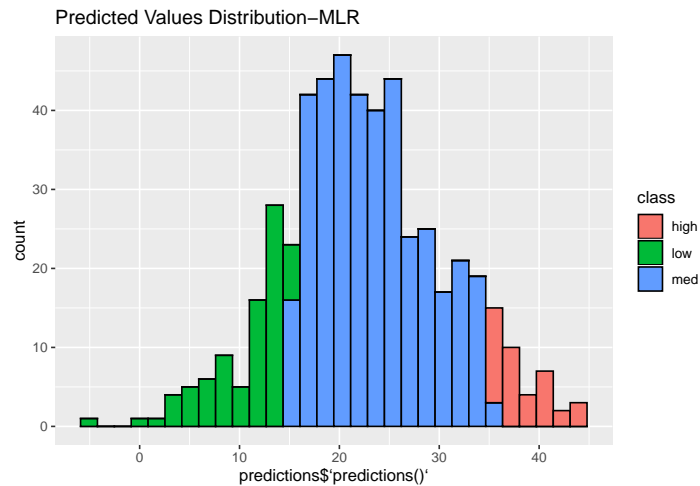
```
ggplot(housing, aes(x=medv,fill=class)) + geom_histogram(colour="black")
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



```
predictions<-predictions%>%mutate(class=apply(predictions$`predictions()`,classifier))
ggplot(predictions, aes(x=predictions$`predictions()`,fill=class)) + geom_histogram(colour="black")+geom
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

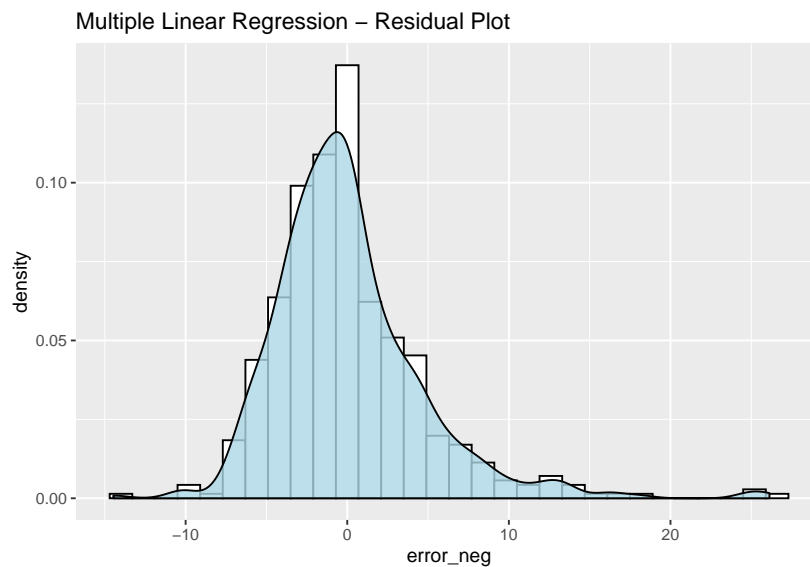


```
targetAndPrediction<-targetAndPrediction%>%mutate(error_neg=target-predictions)
```

Residual Plot-MLR

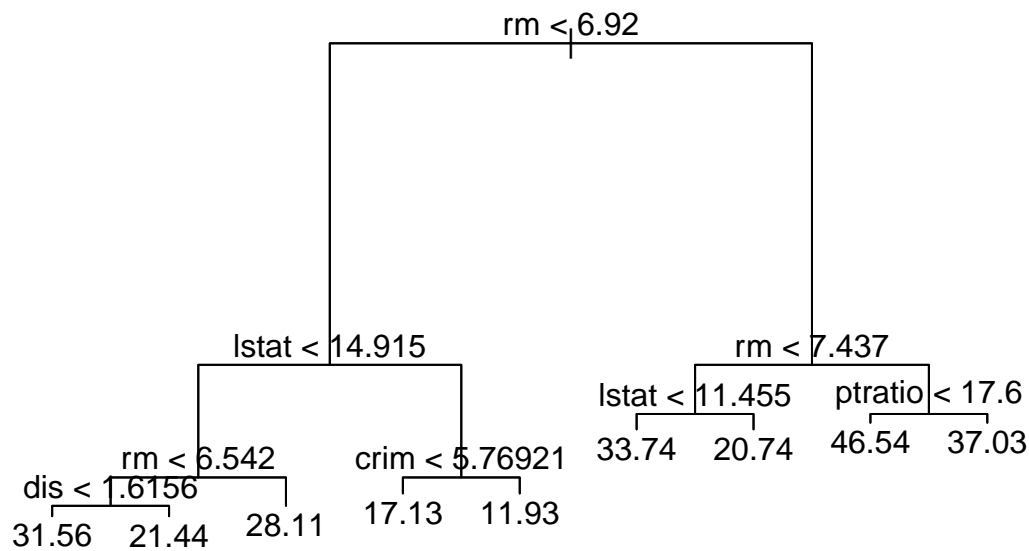
```
ggplot(targetAndPrediction, aes(x=error_neg)) +  
  geom_histogram(aes(y=..density..), colour="black", fill="white")+  
  geom_density(alpha=.8, fill="lightblue")+ggtitle("Multiple Linear Regression - Residual Plot")
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Decision Trees

```
tree.boston=tree(medv~.,train)
plot(tree.boston)
text(tree.boston,pretty=0)
```



```
predictions_tree=predict(tree.boston,newdata=housing)
targetAndPrediction_tree<-cbind(medv,predictions_tree)
targetAndPrediction_tree<-as.data.frame(targetAndPrediction_tree)
names(targetAndPrediction_tree)<-c("target","predictions")
filter(targetAndPrediction_tree,predictions<0)
```

```
[1] target      predictions
<0 rows> (or 0-length row.names)
```

```
for(i in 1:length(targetAndPrediction_tree$predictions)){
  if(targetAndPrediction_tree$predictions[i]<0)
  {
    targetAndPrediction_tree$predictions[i]=0
  }
}

targetAndPrediction_tree<-targetAndPrediction_tree%>%mutate(error=abs(target-predictions))
head(targetAndPrediction_tree)
```

	target	predictions	error
1	21.6	21.44043	0.1595745
2	34.7	33.73939	0.9606061
3	33.4	33.73939	0.3393939
4	36.2	33.73939	2.4606061
5	28.7	21.44043	7.2595745
6	22.9	21.44043	1.4595745

```
mean(targetAndPrediction_tree$error)
```

```
[1] 2.954402
```

```
filter(targetAndPrediction_tree,error>=10)
```

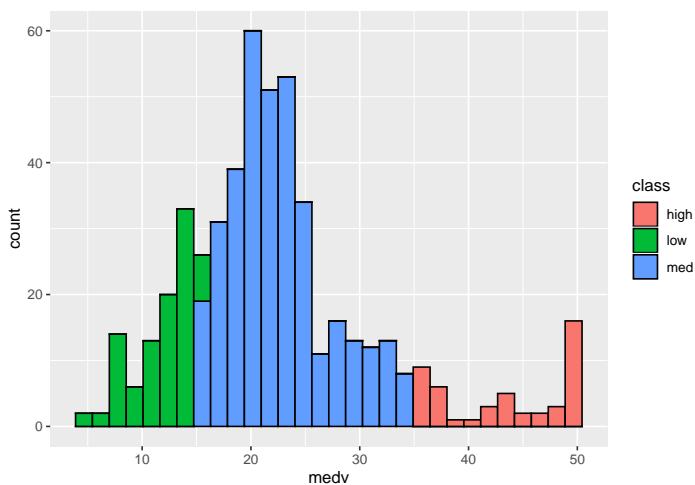
	target	predictions	error
1	23.6	33.73939	10.13939
2	19.6	31.56000	11.96000
3	15.3	31.56000	16.26000
4	36.2	21.44043	14.75957
5	50.0	37.03333	12.96667
6	21.9	37.03333	15.13333
7	50.0	31.56000	18.44000
8	50.0	28.11136	21.88864
9	50.0	33.73939	16.26061
10	50.0	31.56000	18.44000
11	50.0	31.56000	18.44000
12	10.4	20.74000	10.34000
13	27.5	11.93036	15.56964
14	15.0	31.56000	16.56000
15	10.9	21.44043	10.54043
16	7.0	17.12632	10.12632

```
targetAndPrediction_tree<-targetAndPrediction_tree%>%mutate(error_neg=target-predictions)
```

```
targetAndPrediction_tree<-targetAndPrediction_tree%>%mutate(class=sapply(predictions,classifier))
```

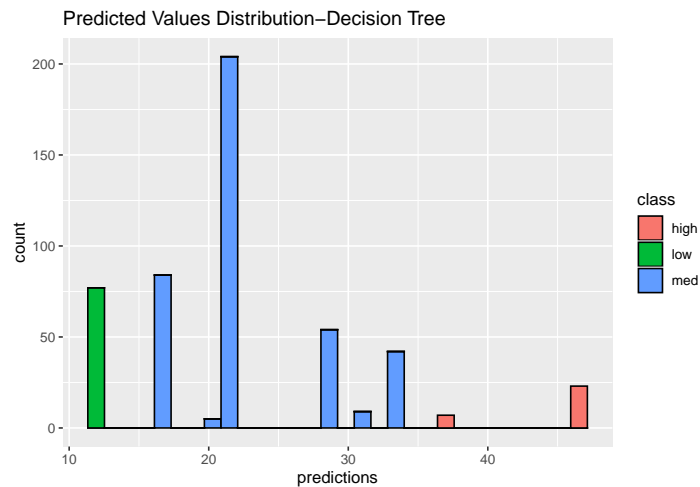
```
ggplot(housing, aes(x=medv,fill=class)) + geom_histogram(colour="black")
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



```
ggplot(targetAndPrediction_tree, aes(x=predictions,fill=class)) + geom_histogram(colour="black")+ggtitle("Predicted Values Distribution-Decision Tree")
```

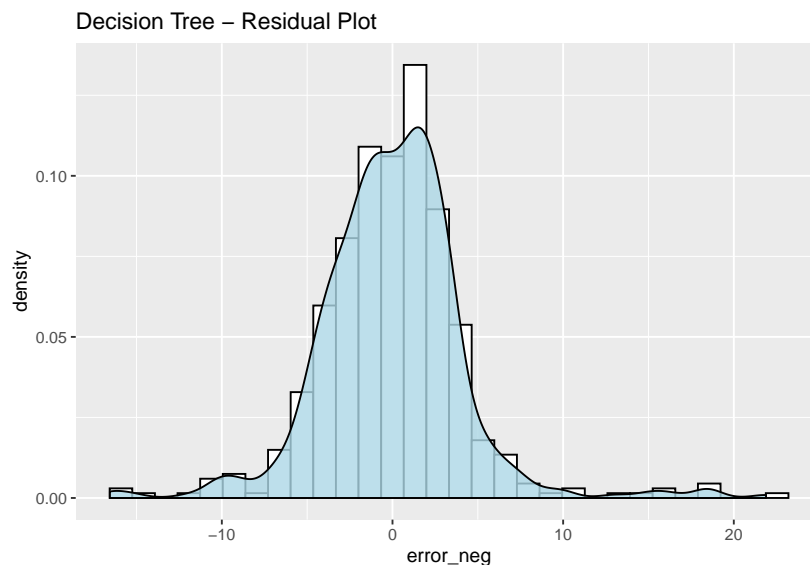
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Residual Plot - Decision Tree

```
ggplot(targetAndPrediction_tree, aes(x=error_neg)) +  
  geom_histogram(aes(y=..density..), colour="black", fill="white")+  
  geom_density(alpha=.8, fill="lightblue")+ggtitle("Decision Tree - Residual Plot")
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



Conclusions:

We have investigated our data & made visualizations for understanding key variables and their relationships, we build models for predictions that have relatively high accuracy. Multiple Linear Regression Models &

Decision Tree Models were both used, Decision trees outperformed Multiple Linear Regression slightly, by an average of 0.4 over our testing data. Errors were plotted to show a normal distribution indicating our model captures variability in our response variable