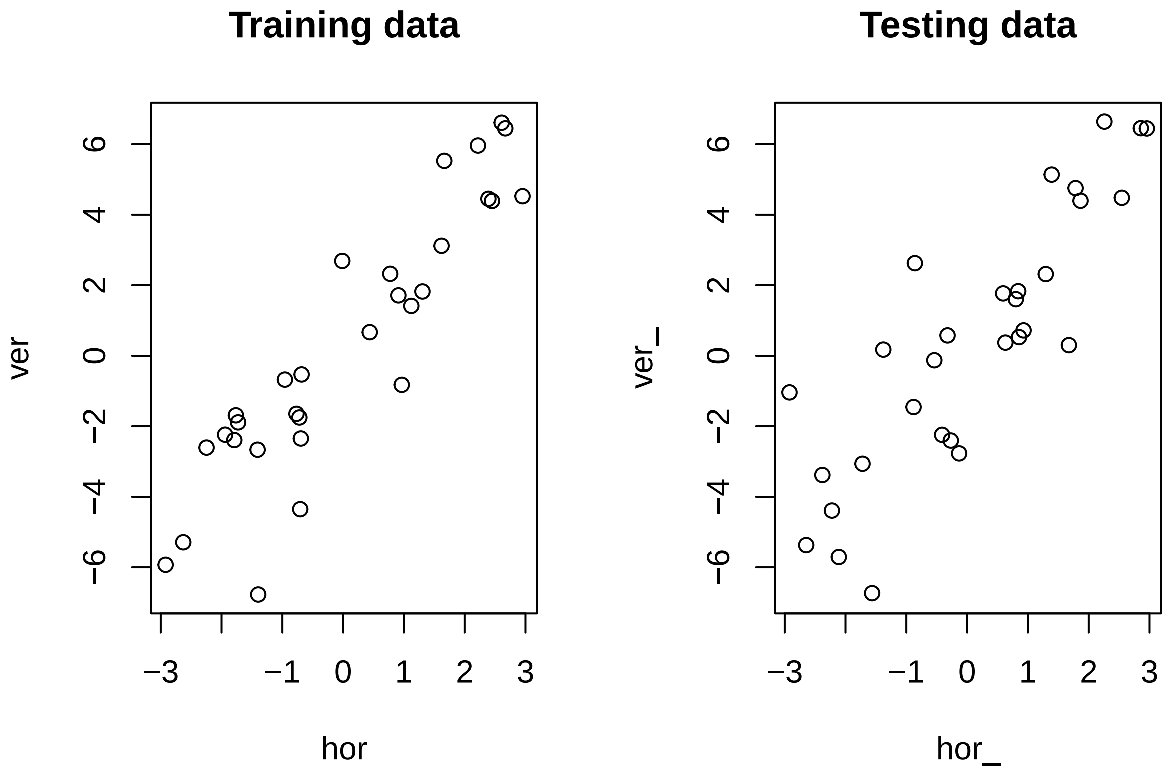
TRAINING AND TESTING DATA

DERRICK ASANTE

2023-06-10

Practice with training and test data

```r
set.seed(1)
n = 30
hor = sort(runif(n, -3, 3))
ver = 2*hor + 2*rnorm(n)
hor_ = sort(runif(n, -3, 3))
ver_ = 2*hor_ + 2*rnorm(n)
par(mfrow=c(1,2))
xlim <- range(c(hor,hor_))
ylim <-range(c(ver,ver_))
plot(hor, ver, xlim=xlim, ylim=ylim, main="Training data")
plot(hor_, ver_, xlim=xlim, ylim=ylim, main="Testing data")
```



```
myfunc<-lapply(2:14, function(daf) lm(ver~poly(hor,daf)))
predict.daf<-function(fau){
rap <- predict(fau, data.frame(hor=hor))
return(rap)
}
yhat.daf<-lapply(myfunc, predict.daf)
train.daf<-function(fau){
mag<-mean((ver-fau)^2)
return(mag)
}
error.train<-sapply(yhat.daf,train.daf)
error.train
```

```
##  [1] 2.275250 2.275065 2.103799 2.103026 1.916462 1.910066 1.604435 1.454798
##  [9] 1.304040 1.280501 1.210219 1.209964 1.196563
```

The training error would significantly decrease if k was as large as 29

Sample splitting with the prostate cancer data

```
dat <-read.table("https://web.stanford.edu/~hastie/ElemStatLearn/datasets/prostate.data")
```

Split the data into training and testing data, fitting a linear model and finding the test error.

```
train<-dat[dat$train=="TRUE",]
test<-dat[dat$train=="FALSE",]
lm.train<-lm(lpsa~lcavol+lweight,data = train)
tap.1<- predict(lm.train,data.frame(lcavol=train$lcavol,lweight=train$lweight))
train_error <- mean((train$lpsa-tap.1)^2)
tap.2 <- predict(lm.train,data.frame(lcavol=test$lcavol,lweight=test$lweight))
test_error.1<- mean((test$lpsa-tap.2)^2)
test_error.1
```

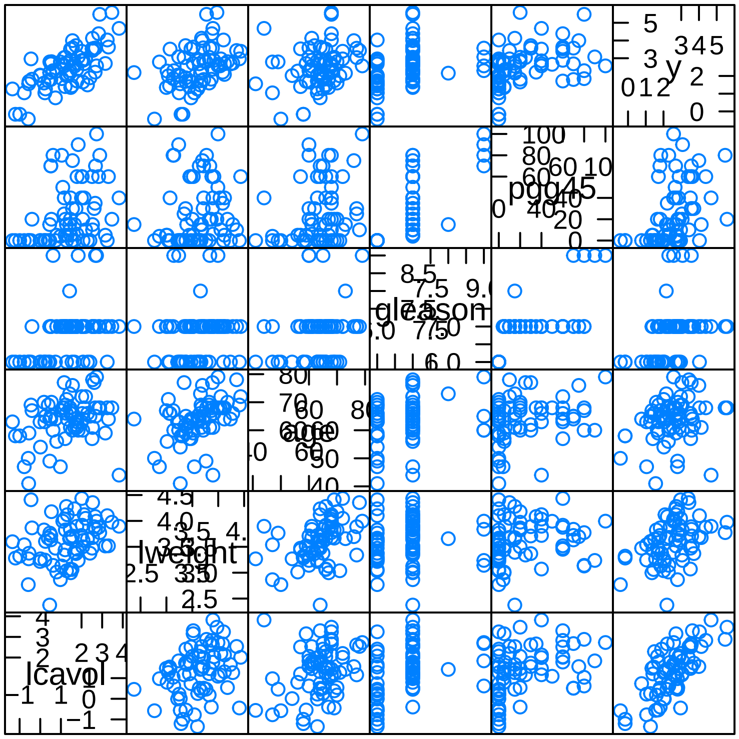
```
## [1] 0.4924823
```

Plot the training dataset using a feature plot to see all variables plotted against each other.

```
inTrain <- createDataPartition(y=dat$lpsa, p=0.7, list=FALSE)
training <- dat[inTrain,]
testing <- dat[-inTrain,]
dim(training)
```

```
## [1] 70 10
```

```
library(caret)
featurePlot(x=training[,c("lcavol", "lweight", "age","gleason","pgg45")],y=training$lpsa, plot="pairs")
```



Scatter Plot Matrix

fit a linear model on the training set lpsa on age, gleason, and pgg45 and finding the test error

```
lm.pred<-lm(lpsa~age+gleason+pgg45,data = train)
pred<- predict(lm.pred,data.frame(age=train$age,gleason=train$gleason,pgg45<-train$pgg45))
error_pred <-mean((train$lpsa-pred)^2)
wab <- predict(lm.pred,data.frame(age=test$age,gleason=test$gleason,pgg45<-test$pgg45))
test_error.2<- mean((test$lpsa-wab)^2)
test_error.2
```

```
## [1] 1.022471
```

the test error of the first model is lower than the second one, Hence it is advisable to choose the first

#Split the wage data randomly into training and test sets of roughly equal size

```
library(ISLR)
library(ggplot2)
library(caret)
data("Wage")
set.seed(1)
n <-sample(rep(1:2, length=nrow(Wage)))
```

```
wage_train <- Wage[n==1,]
wage_test <-Wage[n==2,]
nrow(wage_train)
```

```
## [1] 1500
```

```
nrow(wage_test)
```

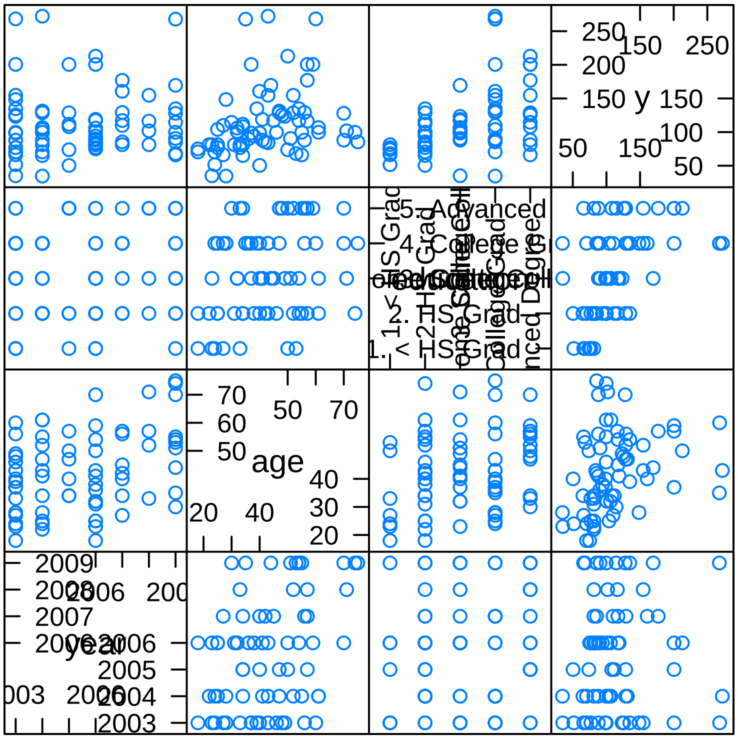
```
## [1] 1500
```

Plot the training dataset using a feature plot to see all variables plotted against each other.

```
inTrain.2 <- createDataPartition(y=Wage$wage, p=0.9, list=FALSE)
training.2<- Wage[inTrain,]
testing.2 <- Wage[-inTrain,]
dim(training)
```

```
## [1] 70 10
```

```
featurePlot(x=training.2[,c("year", "age", "education")],y=training.2$wage, plot="pairs")
```

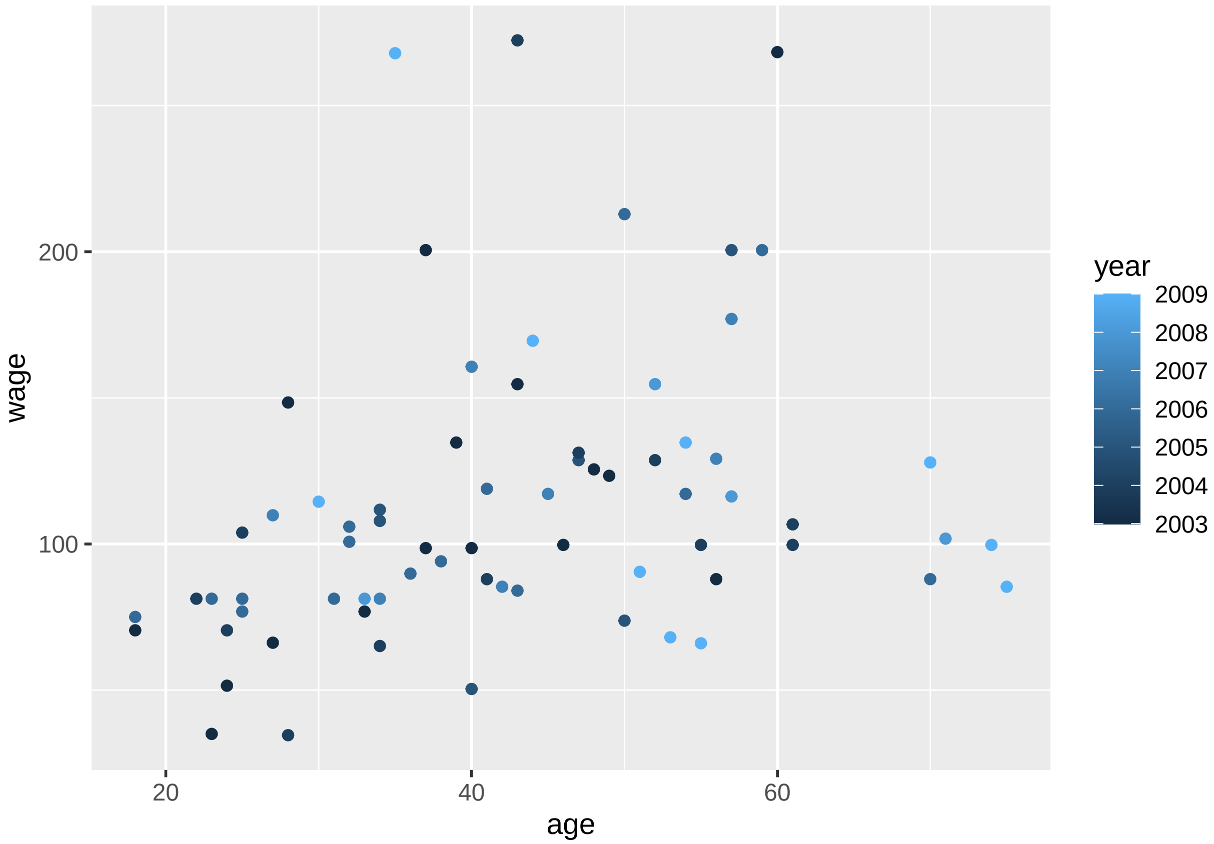


Scatter Plot Matrix

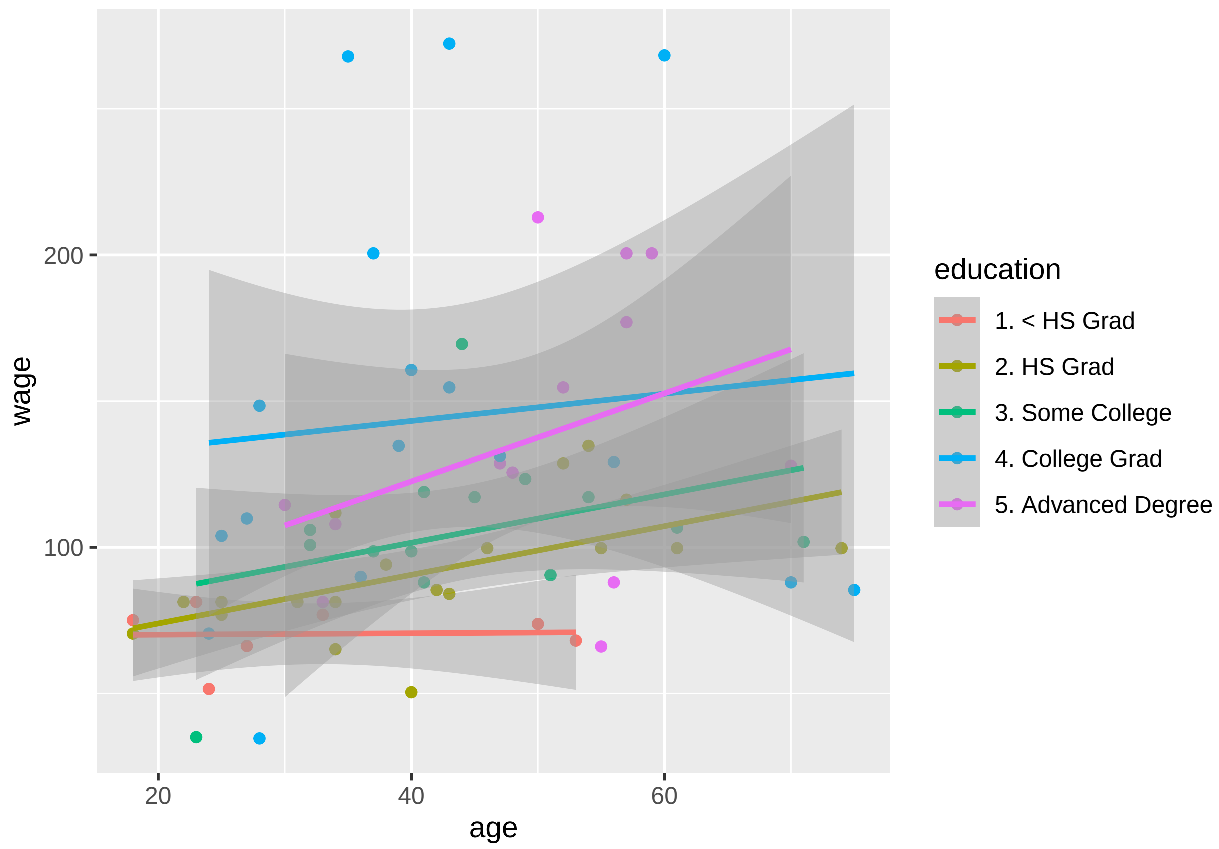
#Plot the training dataset using a qplot with color.

```
qplot(age, wage, color=year, data=training.2)
```

```
## Warning: ‘qplot()‘ was deprecated in ggplot2 3.4.0.
```



```
qq <- qplot(age, wage, color=education, data=training.2)
qq + geom_smooth(method="lm", formula = y ~ x)
```



#Fitting two models (linear and additive) on the training part of the wage data and comparing them

```
wage_train_lm<-lm(wage~year+age+education,data = wage_train)
kab<- predict(wage_train_lm,data.frame(year=wage_test$year,age=wage_test$age,education=wage_test$educat
wage_error.1<- mean((wage_test$wage-kab)^2)
wage_error.1
```

```
## [1] 1286.681
```

```
library(gam)
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loaded gam 1.22-2
```

```
train_gam<-gam(wage~year+s(age)+education, data = wage_train)
rast<-predict(train_gam,data.frame(year=wage_test$year,age=wage_test$age,education <-wage_test$educatio
wage_error.2<- mean((wage_test$wage-rast)^2)
wage_error.2
```

```
## [1] 1241.161
```

The second model predicts better because of the lower test error.