R Notebook

```
library(tidyverse)
```

```
## -- Attaching packages --------

## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.5
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts -----------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 3.0-2
```

Assignment 1: Figure 5.4

Build reasonable replications of the two graphs in Figure 5.4, page 180 of ISLR. For the right-hand graph, start by just producing one 10-Fold plot. If that works, think about how to layer nine of them on top of each other!

What does the result tell you about the best degree to use in a linear model (ie. in lm) in orger to predict mpg as a function of horsepower?

Hint: After you get an idea of how to do this, you might want to eventually to build a simple function that takes a value of the degree and returns the CV estimate of the error (for both LOOCV and k-Fold).

Solution

Get the data…

```r
##
library(ISLR)
data(Auto)
auto.df <- Auto %>%
  dplyr::select(mpg,horsepower)
```

This will be handy…modify the buildFormula function to build a formula for the Auto data.

```r
buildFormAuto <- function(degMax){
 form <- "mpg ~ horsepower"
 if(degMax==1)
   return(form)
  for(deg in 2:degMax){
    form <- paste(form,sprintf("+I(horsepower^%s)",deg),sep=" + ")
  }
 form
}
```

Start by building an example calculation, say with degree=3

```r
deg <- 3
form <- buildFormAuto(deg)
N <- nrow(auto.df)
numFolds <- 10
folds <- sample(1:numFolds,N,rep=T)
mseVals <- numeric(numFolds)
for(fold in 1:numFolds){
  train.df <- auto.df[folds != fold,]
  test.df <- auto.df[folds == fold,]
  mod <- lm(formula(form),data=train.df)
  pred <- predict(mod,newdata=test.df)
  mseVals[fold] <- with(test.df,mean((mpg - pred)^2))
}
mean(mseVals)
```

```
## [1] 19.39487
```

This seemed to work ok. Pack all of this into a function.

```r
compMSEDeg <- function(deg){
  form <- buildFormAuto(deg)
  N <- nrow(auto.df)
  numFolds <- 10
  folds <- sample(1:numFolds,N,rep=T)
  mseVals <- numeric(numFolds)
  for(fold in 1:numFolds){
    train.df <- auto.df[folds != fold,]
    test.df <- auto.df[folds == fold,]
    mod <- lm(formula(form),data=train.df)
    pred <- predict(mod,newdata=test.df)
    mseVals[fold] <- with(test.df,mean((mpg - pred)^2))
  }
  mean(mseVals)
}
```

Test it out

```
compMSEDeg(3)
```

```
## [1] 19.46934
```

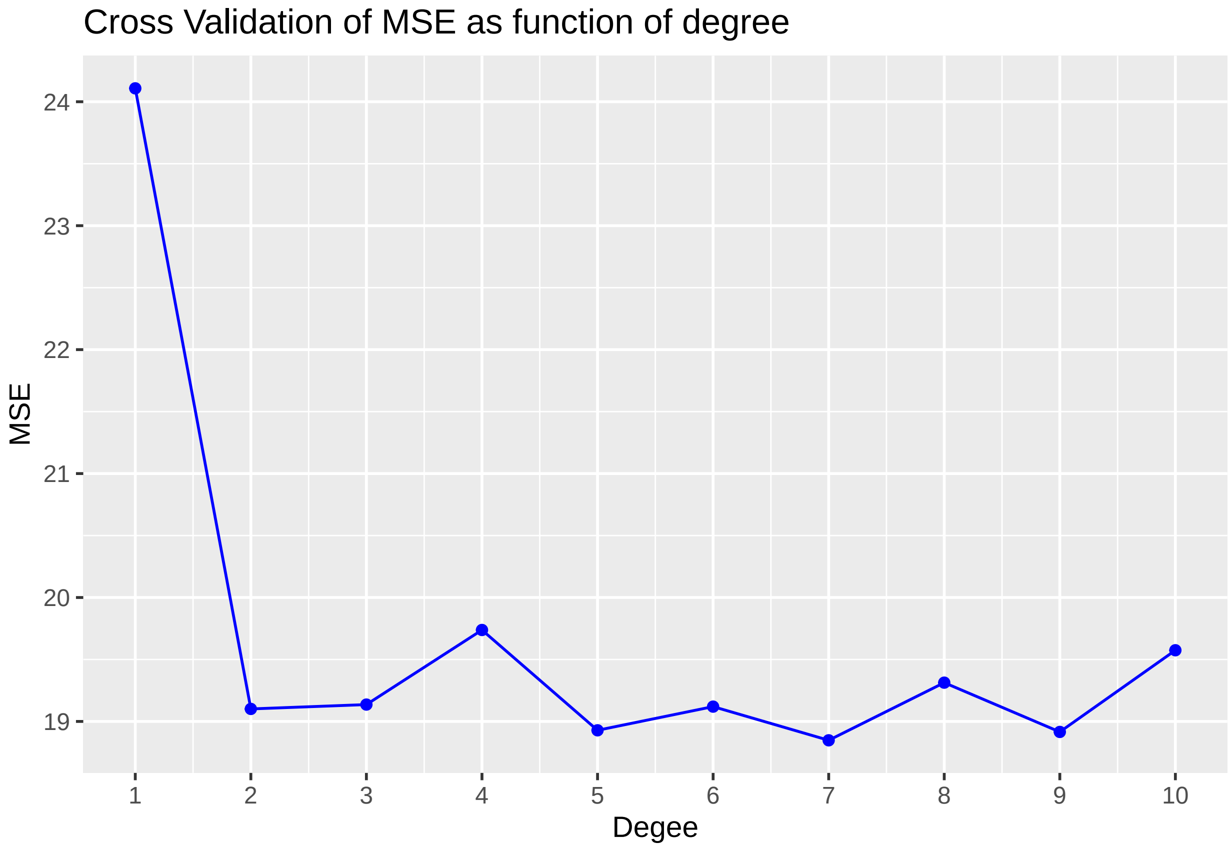
```
compMSEDeg(6)
```

```
## [1] 18.60811
```

Now make a plot of one 10-fold cross validation of MSE as a functon of the degree of the polynomial.

```
## map_dbl time
maxDeg <- 10
degs <- 1:maxDeg
errs <- map_dbl(degs,compMSEDeg)

data.frame(deg=degs,err=errs) %>%
  ggplot()+
  geom_point(aes(deg,err),color="blue")+
  geom_line(aes(deg,err),color="blue")+
  scale_x_continuous(breaks=1:maxDeg)+
  labs(title="Cross Validation of MSE as function of degree",
       x="Degee",y="MSE")
```



Now the final picture. Repeat the cross validation 9 times and combine all of them into a single plot.

First build 9 different cross validation results. Put the errors into a matrix.

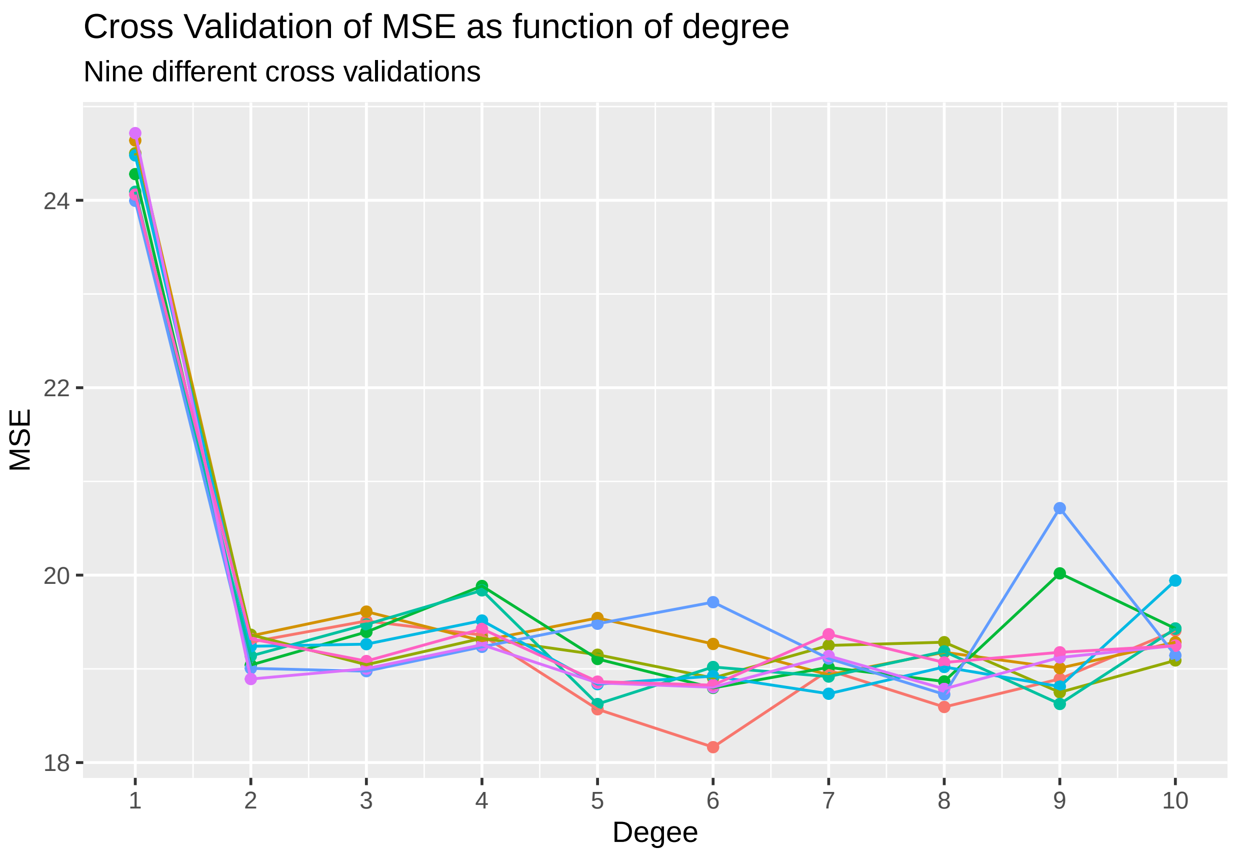
```
numRuns <- 9
errsDegs <- matrix(nrow=maxDeg,
                   ncol=numRuns)
##Here we go
for(run  in  1:numRuns){
  errsDegs[,run] <-  map_dbl(degs,compMSEDeg)
}
```

Pack into a data frame.

```
errsDegs.df <- data.frame(  errsDegs)
##assign names
names(errsDegs.df) <- paste0("run",1:numRuns)
##add degrees
errsDegs.df$deg <- degs
```

And the plot. . . .

```
errsDegs.df %>%
  gather(run,err,run1:run9) %>%
  ggplot()+
  geom_point(aes(deg,err,color=run))+
    geom_line(aes(deg,err,color=run))+
  scale_x_continuous(breaks=1:maxDeg)+
  labs(title="Cross Validation of MSE as function of degree",
       subtitle="Nine different cross validations",
       x="Degee",y="MSE")+
  guides(color=FALSE)
```


Cross Validation of MSE as function of degree
Nine different cross validations
MSE
24
22
20
18
1
2
3
4
5
6
7
8
9
10
Degee