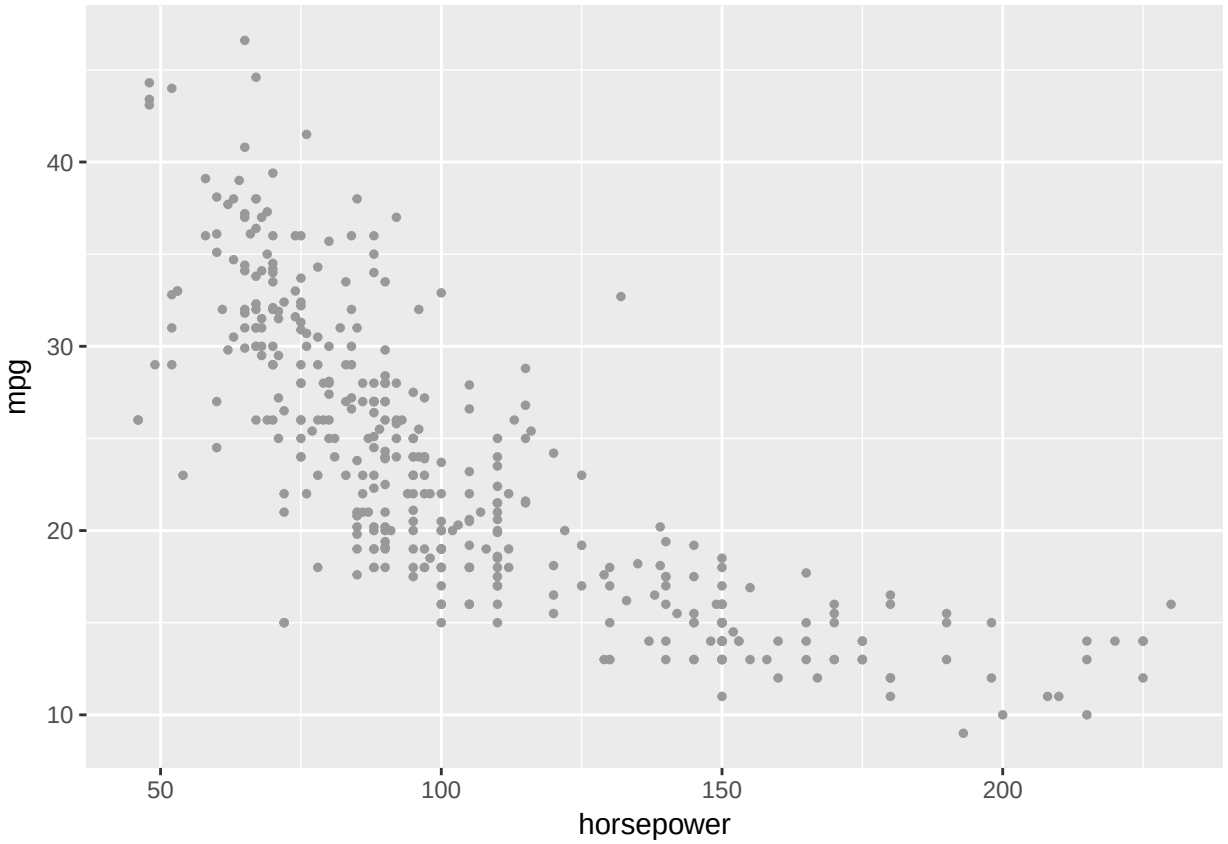
Project II

Zhijian Liu

12/4/2018

data

```
Auto %>%
  ggplot() +
  geom_point(aes(x= horsepower, y = mpg, color = set.type), size = 1, color = "#999999")
```



$CV_{i,j}$ ## K-fold

```
attach(Auto)
```

```
## The following object is masked from package:ggplot2:
##
##     mpg
```

```
K <- 5
n <- nrow(Auto)
fold <- ceiling(n/K)
s <- sample(rep(1:K, fold), n)
table(s)
```

```
## s
##  1  2  3  4  5
```

```
## 78 78 79 78 79
```

```
#graph folds
plots <- list()
for (i in 1:K){
  validate.index <- seq_len(n)[s == i]
  train.index <- seq_len(n)[s != i]
  set.type <- rep(NA, n)
  set.type[validate.index] <- "validation set"
  set.type[train.index] <- "training set"
  df <- cbind(Auto, set.type)
  plots[[i]] <- df %>%
    ggplot() +
    geom_point(aes(x = horsepower, y = mpg, color = set.type), size = 1) +
    scale_color_manual(values=c("#999999", "#E69F00"))
  plots[[i]] %>%
    ggsave(paste0("/Users/liubeixi/Desktop/Regression/Project II/","sample fold",i, ".jpg"),.)
}
```

```
## Saving 6.5 x 4.5 in image
```

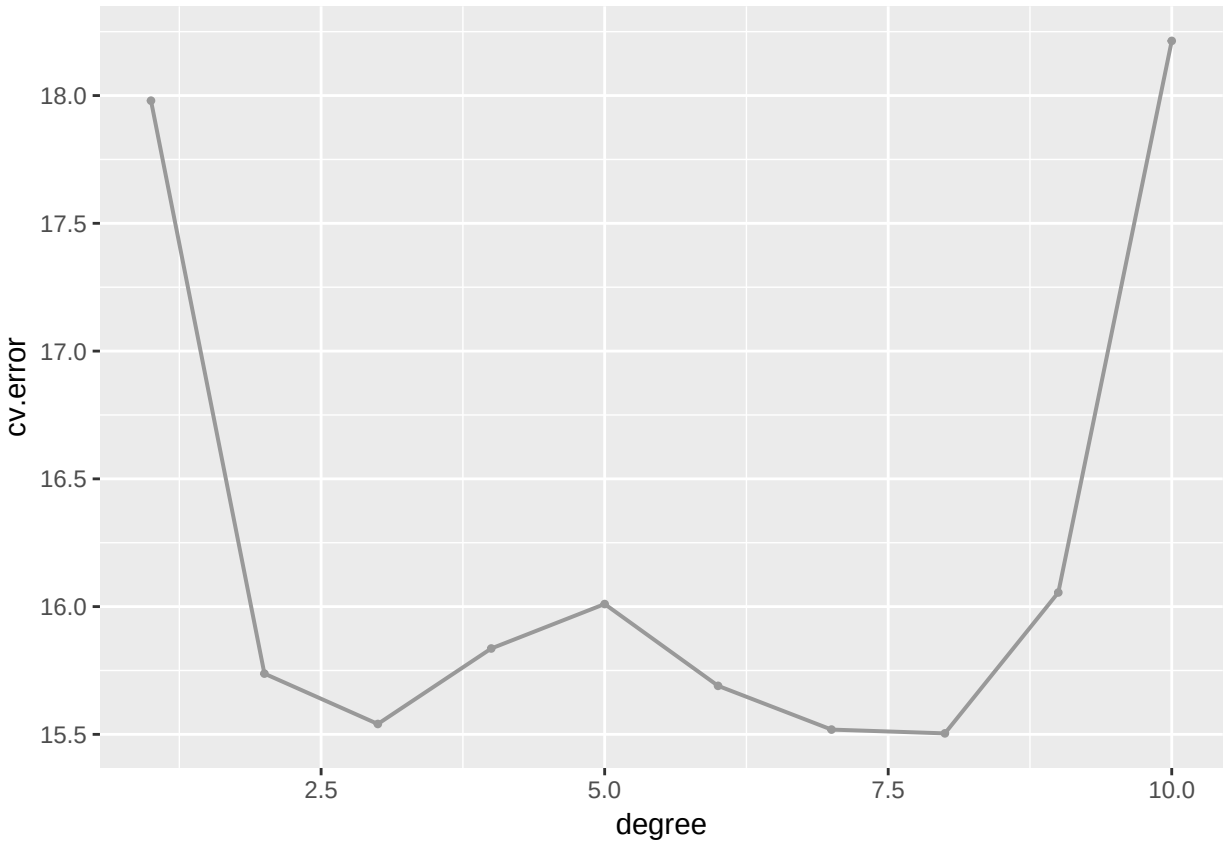
```
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
```

```
#graph folds + lines
plots.reg <- list()
for (i in 1:K){
  validate.index <- seq_len(n)[s == i]
  train.index <- seq_len(n)[s != i]
  set.type <- rep(NA, n)
  set.type[validate.index] <- "validation set"
  set.type[train.index] <- "training set"
  df <- cbind(Auto, set.type)
  plots.reg[[i]] <- list()
  f <- list()
  for (j in 1:5){
    f[[j]] <- formula(paste0('y ~ poly(x, ',j,')'))
    plots.reg[[i]][[j]] <- df %>%
      ggplot() +
      geom_point(aes(x= horsepower, y = mpg, color = set.type), size = 1) +
      scale_color_manual(values=c("#999999", "#E69F00")) +
      geom_smooth(aes(x= horsepower, y = mpg), color = "#999999", method = "glm", formula = f[[j]], se =
    plots.reg[[i]][[j]] %>%
      ggsave(paste0("/Users/liubeixi/Desktop/Regression/Project II/","sample fold",i,".",j, ".jpg"),.)
  }
}
```

```
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
```

```
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
```

```r
#what power of "horsepoewer" is optimal for this prediction?
cv.error <- rep(0, 5)
for (p in 1:10){
  glm.fit <- glm(mpg~ weight + poly(horsepower,p) + acceleration) #with power p = 1:10
  cv.error[p] <- boot::cv.glm(Auto, glm.fit, K=5)$delta[1] #prediction error
}
data.frame(cv.error, degree = c(1:10)) %>%
  ggplot(aes(x=degree, y=cv.error)) +
  geom_point(size = 0.85, color = "#999999") +
  geom_line(size = 0.7, color = "#999999")
```




```
which.min(cv.error)
```

```
## [1] 8
```

LOOCV

```
#what power of "horsepoewer" is optimal for this prediction?
cv.error <- rep(0, 10)
for (p in 1:10){
  glm.fit <- glm(mpg~ weight + poly(horsepower,p) + acceleration) #with power p = 1:10
  cv.error[p] <- boot::cv.glm(Auto, glm.fit)$delta[1] #prediction error, specify K=10
}
cv.error
```

```
##  [1] 18.25595 15.90163 15.72995 15.86879 15.74517 15.74989 15.70073
##  [8] 15.79314 15.95933 16.38301
```

```
which.min(cv.error)
```

```
## [1] 7
```