Untitled

Bao Doquang

March 12, 2020

```
library(foreach)
```

```
## Warning: package 'foreach' was built under R version 3.6.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.6.2
```

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

```
## <U+2713> ggplot2 3.2.1     <U+2713> purrr   0.3.3
## <U+2713> tibble  2.1.3     <U+2713> dplyr   0.8.4
## <U+2713> tidyr   1.0.0     <U+2713> stringr 1.4.0
## <U+2713> readr   1.3.1     <U+2713> forcats 0.4.0
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x purrr::accumulate() masks foreach::accumulate()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x purrr::when()       masks foreach::when()
```

```
library(class)
library(mosaic)
```

```
## Warning: package 'mosaic' was built under R version 3.6.2
```

```
## Loading required package: lattice
```

```
## Loading required package: ggformula
```

```
## Loading required package: ggstance
```

```
## Warning: package 'ggstance' was built under R version 3.6.2
```

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
## Warning: package 'mosaicData' was built under R version 3.6.2
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

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

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(FNN)
```

```
## Warning: package 'FNN' was built under R version 3.6.2
```

```
##
## Attaching package: 'FNN'
```

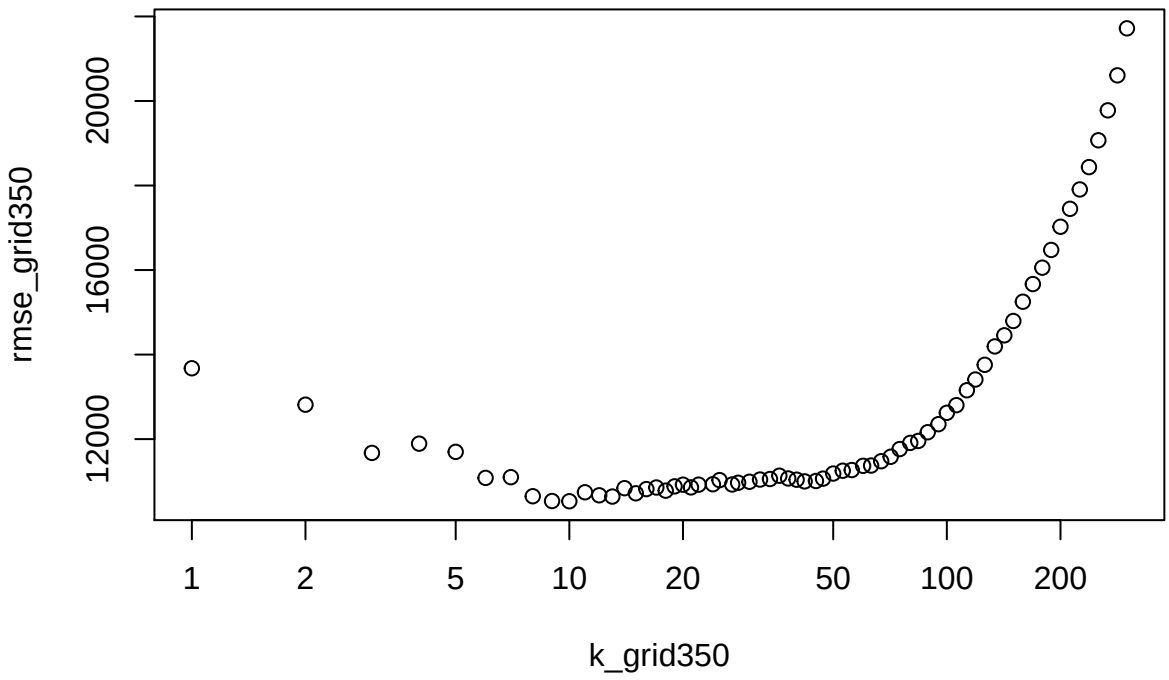
```
## The following objects are masked from 'package:class':
##
##     knn, knn.cv
```

```
sclass <- read.csv("~/GitHub/SDS323_Spring2020/hw2/Question 1/sclass.csv")
```

```
sclass350 = subset(sclass, trim == '350')
sclass65AMG = subset(sclass, trim == '65 AMG')

n350 = nrow(sclass350)
n_train350 = round(0.8*n350)
n_test350 = n350 - n_train350
train_cases350 = sample.int(n350, n_train350, replace=FALSE)
test_cases350 = setdiff(1:n350, train_cases350)
sclass350_train = sclass350[train_cases350,]
sclass350_test = sclass350[test_cases350,]

n65AMG = nrow(sclass65AMG)
n_train65AMG = round(0.8*n65AMG)
n_test65AMG = n65AMG - n_train65AMG
train_cases65AMG = sample.int(n65AMG, n_train65AMG, replace=FALSE)
test_cases65AMG = setdiff(1:n65AMG, train_cases65AMG)
sclass65AMG_train = sclass65AMG[train_cases65AMG,]
sclass65AMG_test = sclass65AMG[test_cases65AMG,]

Xtrain350 = model.matrix(~ mileage, data=sclass350_train)
Xtest350 = model.matrix(~ mileage, data=sclass350_test)

ytrain350 = sclass350_train$price
ytest350 = sclass350_test$price

Xtrain65AMG = model.matrix(~ mileage, data=sclass65AMG_train)
Xtest65AMG = model.matrix(~ mileage, data=sclass65AMG_test)

ytrain65AMG = sclass65AMG_train$price
ytest65AMG = sclass65AMG_test$price

K = 2
rmse = function(y, yhat) {
  sqrt( mean( (y - yhat)^2 ) )
}

knn_model350 = knn.reg(Xtrain350, Xtest350, ytrain350, k=K)

rmse(ytest350, knn_model350$pred)
```

```
## [1] 12814.67
```

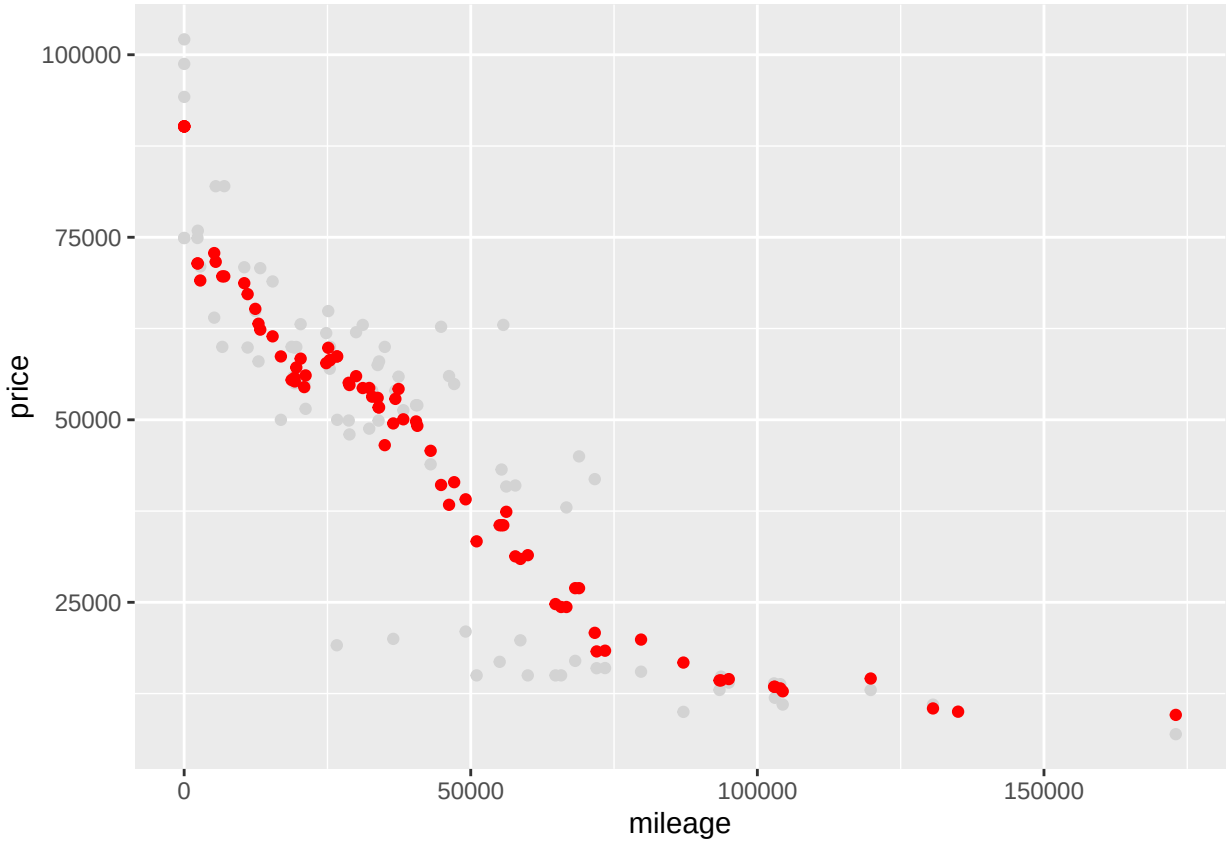
```
k_grid350 = exp(seq(log(1), log(300), length=100)) %>% round %>% unique
rmse_grid350 = foreach(K = k_grid350, .combine='c') %do% {
  knn_model350 = knn.reg(Xtrain350, Xtest350, ytrain350, k=K)
  rmse(ytest350, knn_model350$pred)
}

plot(k_grid350, rmse_grid350, log='x')
```




```
k_grid350[which.min(rmse_grid350)]
```

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

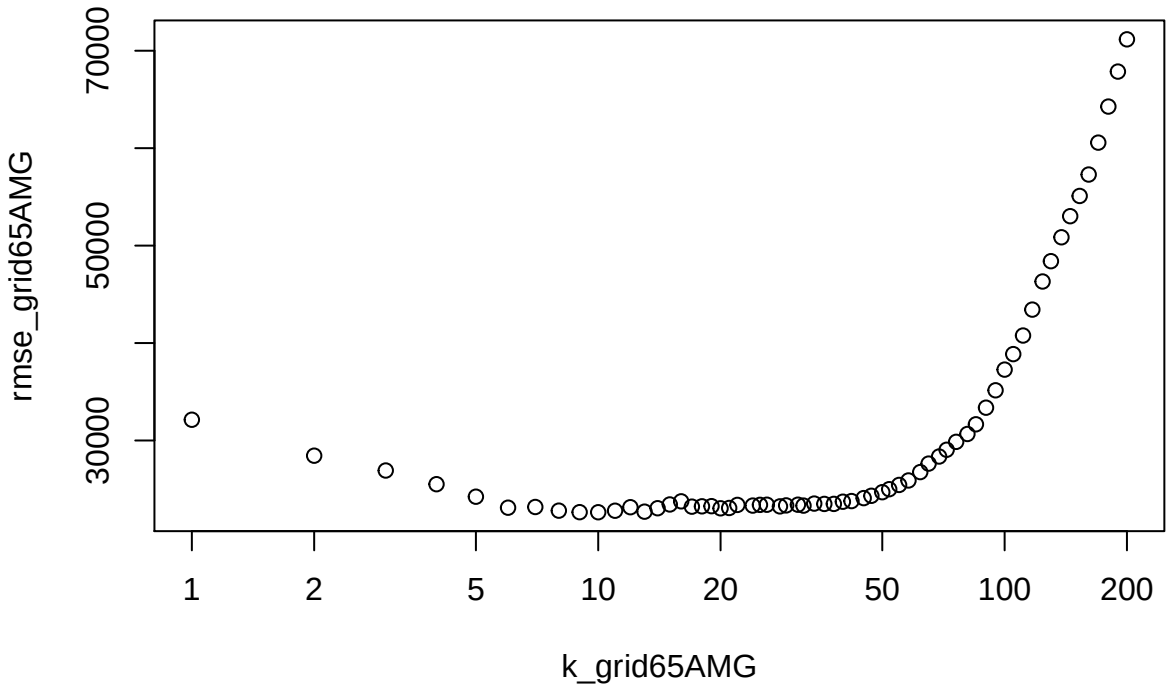
```
finalknn350 = knn.reg(train = Xtrain350, test = Xtest350, y = ytrain350, k=k_grid350[which.min(rmse_gri
)
ypred_finalknn350 = finalknn350$pred

sclass350_test$ypred_finalknn350 = ypred_finalknn350
p_test = ggplot(data = sclass350_test) +
  geom_point(mapping = aes(x = mileage, y = price), color='lightgrey')
p_test + geom_point(aes(x = mileage, y = ypred_finalknn350), color='red')
```




```
K =2
knn_model65AMG = knn.reg(Xtrain65AMG, Xtest65AMG, ytrain65AMG, k=K)

rmse(ytest65AMG, knn_model65AMG$pred)
```

```
## [1] 28441.24
```

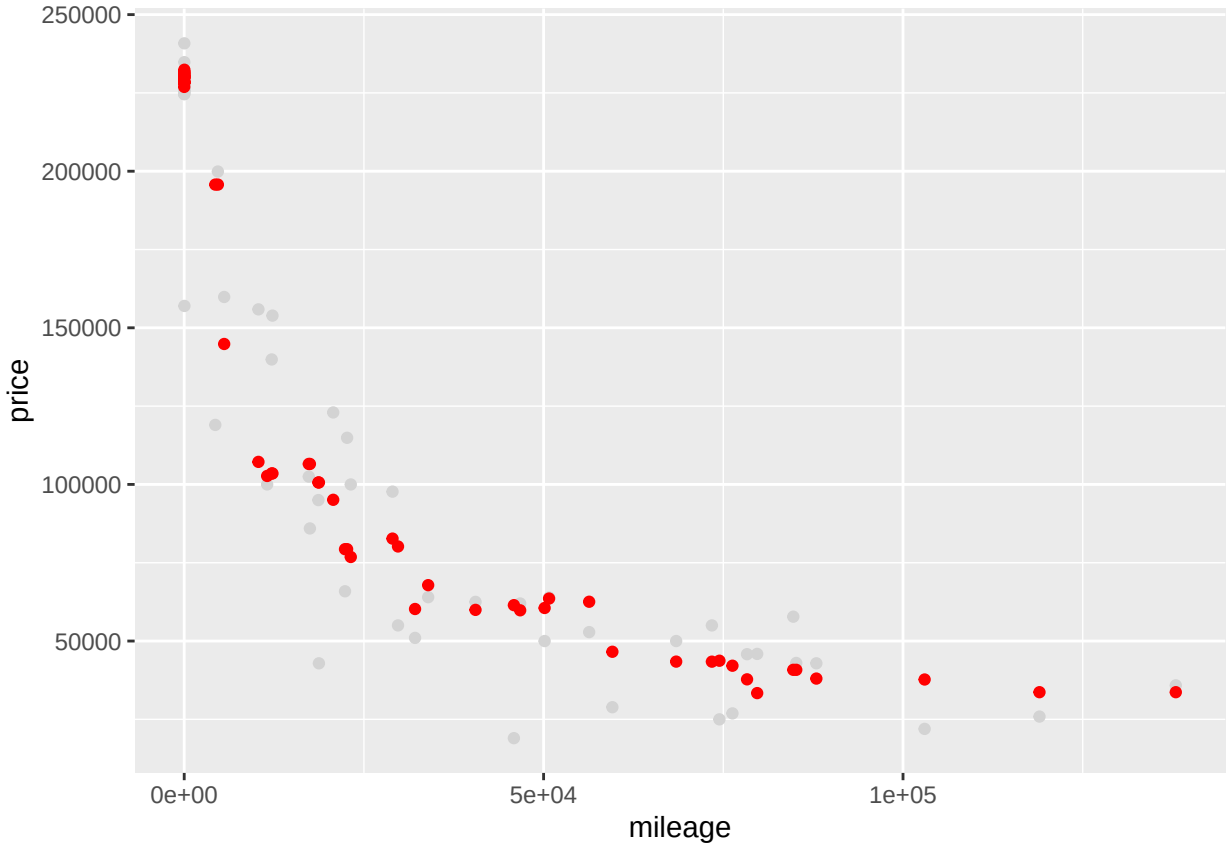
```
k_grid65AMG = exp(seq(log(1), log(200), length=100)) %>% round %>% unique
rmse_grid65AMG = foreach(K = k_grid65AMG, .combine='c') %do% {
  knn_model65AMG = knn.reg(Xtrain65AMG, Xtest65AMG, ytrain65AMG, k=K)
  rmse(ytest65AMG, knn_model65AMG$pred)
}

plot(k_grid65AMG, rmse_grid65AMG, log='x')
```




```
k_grid65AMG[which.min(rmse_grid65AMG)]
```

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

```
finalknn65AMG = knn.reg(train = Xtrain65AMG, test = Xtest65AMG, y = ytrain65AMG, k=k_grid65AMG[which.mi
)
ypred_finalknn65AMG = finalknn65AMG$pred

sclass65AMG_test$ypred_finalknn65AMG = ypred_finalknn65AMG
p_test = ggplot(data = sclass65AMG_test) +
  geom_point(mapping = aes(x = mileage, y = price), color='lightgrey')
p_test + geom_point(aes(x = mileage, y = ypred_finalknn65AMG), color='red')
```


250000
200000
150000
100000
50000
price
0e+00
5e+04
1e+05
mileage