Exercise 1

Zhengyi Lin

```
library(ggplot2)
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## v purrr   0.3.5
```

```
## Warning: package 'tidyr' was built under R version 4.2.2
```

```
## Warning: package 'readr' was built under R version 4.2.2
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
## Warning: package 'forcats' was built under R version 4.2.2
```

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

```
library(dplyr)
library(rsample)
```

```
## Warning: package 'rsample' was built under R version 4.2.2
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.2
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(modelr)
```

```
## Warning: package 'modelr' was built under R version 4.2.2
```

```
library(parallel)
library(foreach)
```

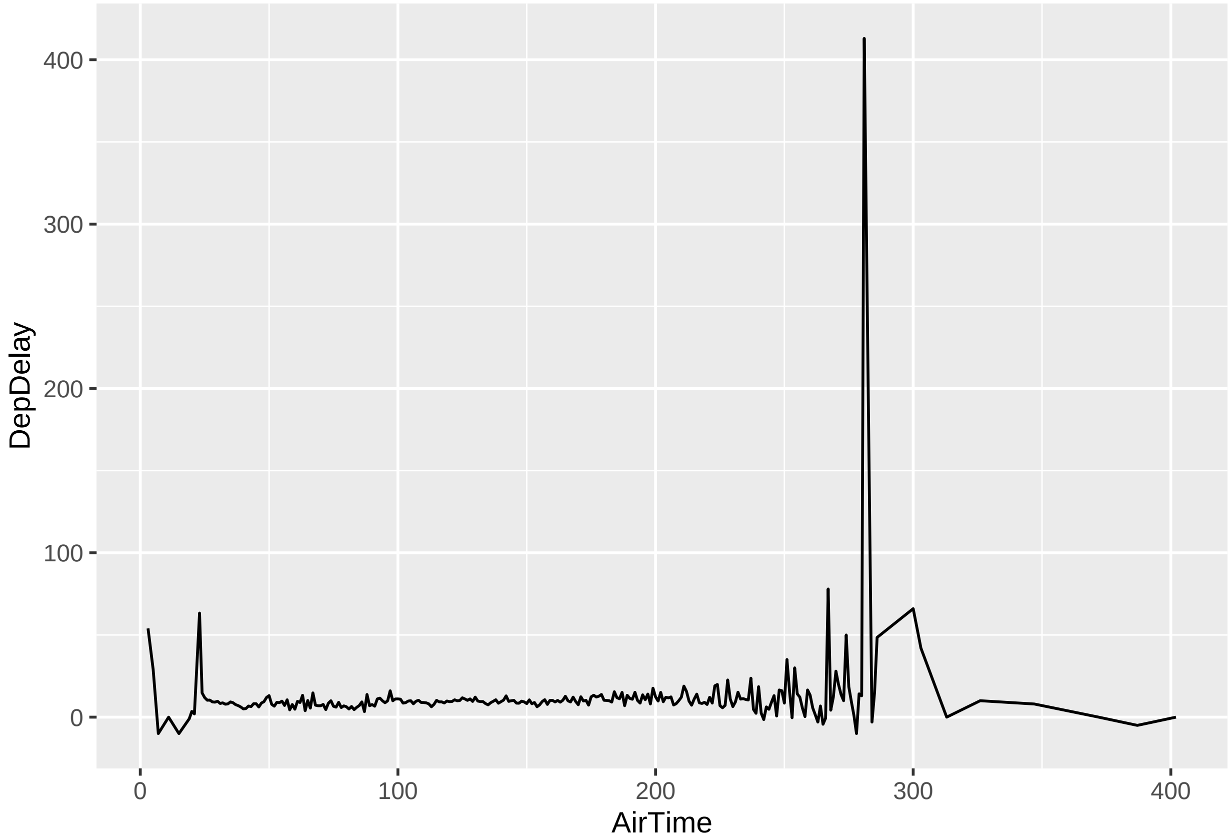
```
## Warning: package 'foreach' was built under R version 4.2.2
```

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

```
library(rsample)
```

1) Data visualization: flights at ABIA

```
ABIA = read.csv("ABIA.csv")

AVG_Delay <- aggregate(DepDelay ~ AirTime, ABIA, mean)

ggplot(AVG_Delay, aes(x=AirTime, y=DepDelay))+
  geom_line()
```



If the air time is from 20-30 minutes, it is the least possible to delay. But if the air time is beween 250-300 minutes, it is the most likely to delay.

2) Wrangling the Olympics

A

```
olympics_top20=read.csv("olympics_top20.csv")
olympics_female<-olympics_top20 %>%
  filter(sex =="F", sport=='Athletics')

quantile(olympics_top20$height, probs=0.95)
```

```
## 95%
## 197
```

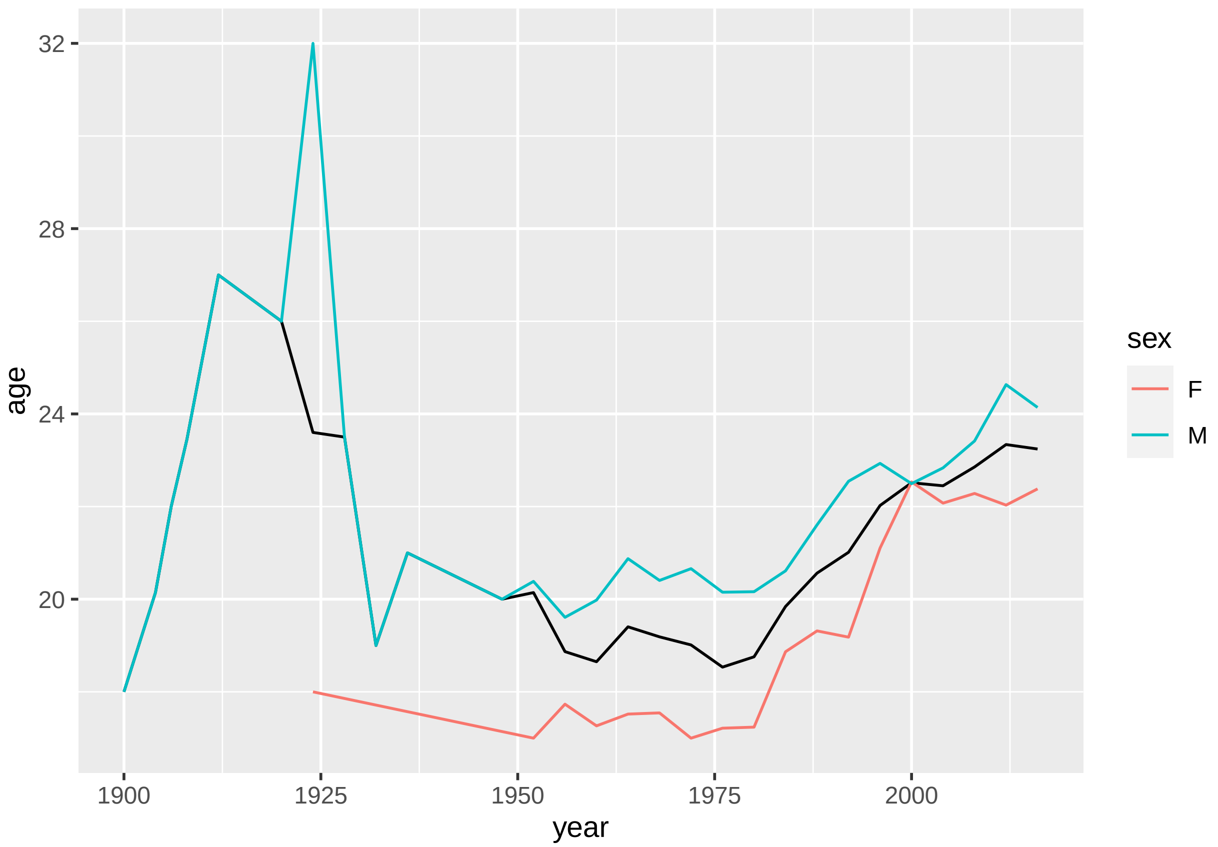
B

```
olympics_female<-olympics_top20 %>%
  filter(sex =="F")

result<-split(olympics_female, olympics_female$event)%>%
  lapply(., function(x)sd(x$height))%>%
  unlist(.)%>%as.data.frame(.)
res<-cbind(row.names(result),result)
colnames(res)<-c("event","height_std_dev")

res_order<-res[order(res$height_std_dev, decreasing = TRUE),]

head(res_order)
```

```
##                                                                        event
## Rowing Women’s Coxed Fours                       Rowing Women’s Coxed Fours
## Basketball Women’s Basketball                 Basketball Women’s Basketball
## Rowing Women’s Coxed Quadruple Sculls Rowing Women’s Coxed Quadruple Sculls
## Rowing Women’s Coxed Eights                     Rowing Women’s Coxed Eights
## Swimming Women’s 100 metres Butterfly Swimming Women’s 100 metres Butterfly
## Volleyball Women’s Volleyball                 Volleyball Women’s Volleyball
##                                       height_std_dev
## Rowing Women’s Coxed Fours                 10.865490
## Basketball Women’s Basketball               9.700255
## Rowing Women’s Coxed Quadruple Sculls       9.246396
## Rowing Women’s Coxed Eights                 8.741931
## Swimming Women’s 100 metres Butterfly       8.134399
## Volleyball Women’s Volleyball               8.101521
```

C

```
olympics_swimmer <- olympics_top20 %>%
  filter(sport == 'Swimming')

AVG_Age_Swimmer <- aggregate(age ~ year, olympics_swimmer, mean)

AVG_Age_Gender <- aggregate(age ~ year + sex, olympics_swimmer, mean)

ggplot()+
  geom_line(data = AVG_Age_Swimmer, aes(x=year, y=age))+
  geom_smooth(span = 1)+
  geom_line(data=AVG_Age_Gender, aes(x=year, y=age, color = sex))
```



Before 1925, female's average age increase sharply. But after 1925, the average age decrease a lot until 1950. Female's average age is generally higher than male's average age.

3) K-nearest neighbors: cars

```
sclass=read.csv("sclass.csv")
trim_350 <- sclass %>%
  filter(trim =='350')
trim_65_AMG <- sclass %>%
  filter(trim == '65 AMG')

trim_350_split =  initial_split(trim_350, prop=0.8)
trim350_train = training(trim_350_split)
trim350_test  = testing(trim_350_split)
trim_AMG_split = initial_split(trim_65_AMG, prop=0.8)
trimAMG_train = training(trim_AMG_split)
trimAMG_test = testing(trim_AMG_split)

# k-value cross validation
k_folds = 5

trim350_folds = crossv_kfold(trim_350, k=k_folds)
trimAMG_folds = crossv_kfold(trim_65_AMG,k=k_folds)

# define a series of k
```

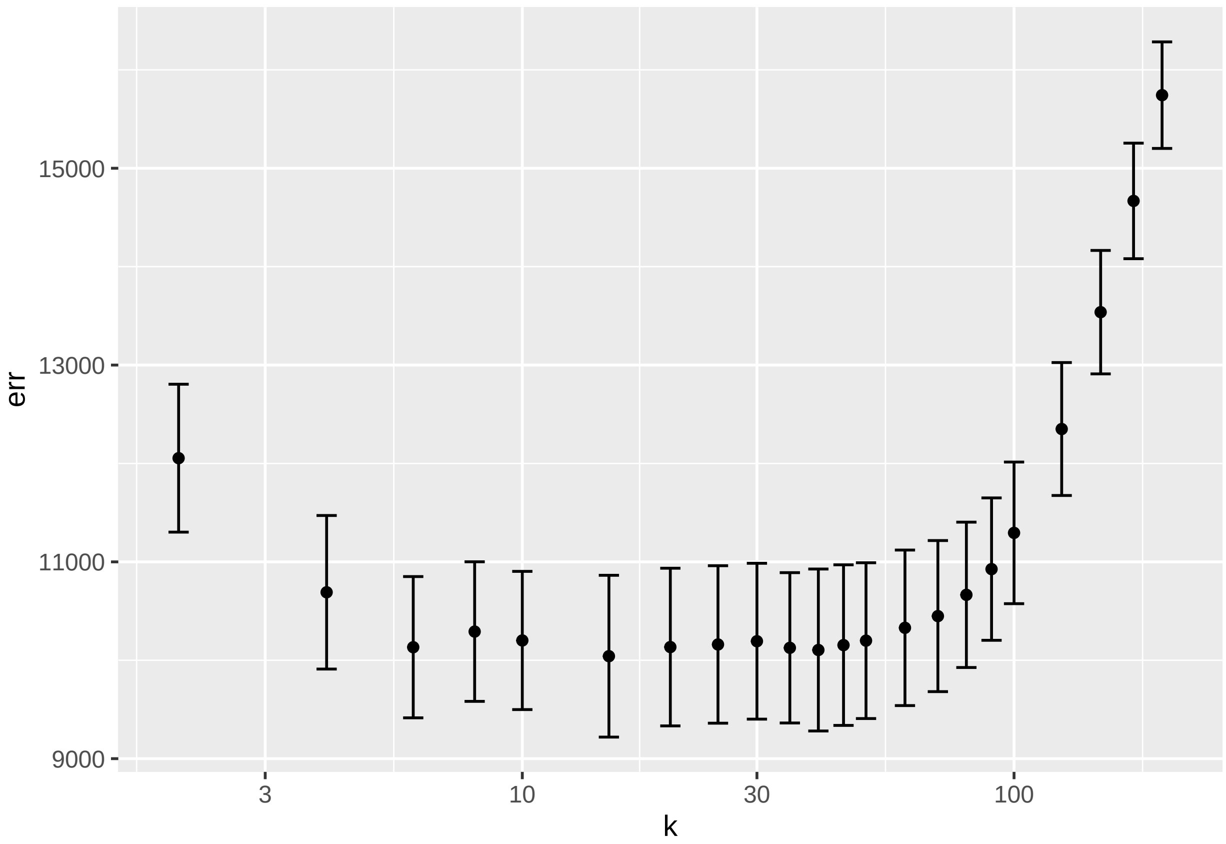
```
k_grid = c(2, 4, 6, 8, 10, 15, 20, 25, 30, 35, 40, 45,
           50, 60, 70, 80, 90, 100, 125, 150, 175, 200)

cv_grid350 = foreach(k = k_grid, .combine='rbind') %dopar% {
  models = map(trim350_folds$train, ~ knnreg(price ~ mileage, k=k, data = ., use.all=FALSE))
  errs = map2_dbl(models, trim350_folds$test, modelr::rmse)
  c(k=k, err = mean(errs), std_err = sd(errs)/sqrt(k_folds))
} %>% as.data.frame
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
head(cv_grid350)
```

```
##           k      err  std_err
## result.1  2 12053.79 751.4725
## result.2  4 10690.99 780.2537
## result.3  6 10132.67 717.7182
## result.4  8 10291.42 709.1802
## result.5 10 10201.08 702.3212
## result.6 15 10041.33 822.2276
```

```
ggplot(cv_grid350) +
  geom_point(aes(x=k, y=err)) +
  geom_errorbar(aes(x=k, ymin = err-std_err, ymax = err+std_err)) +
  scale_x_log10()
```

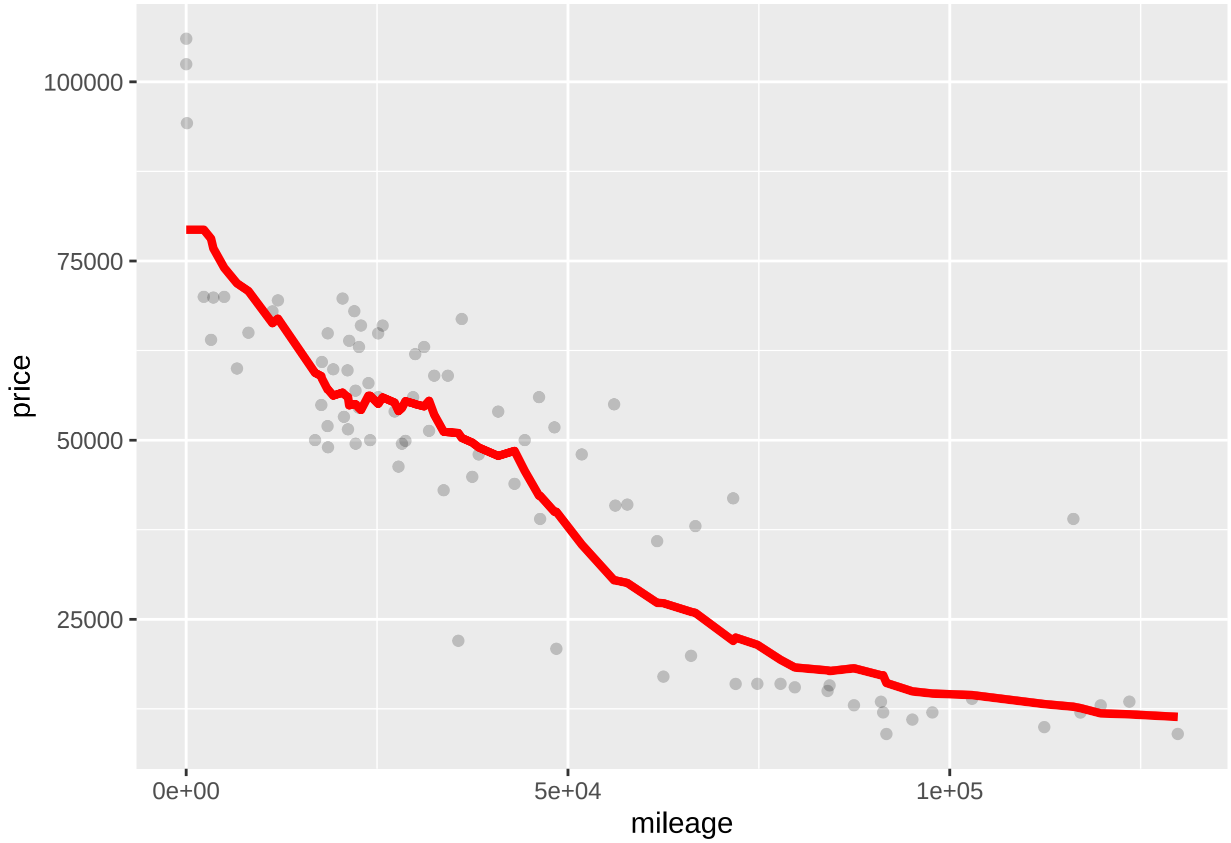


```
knn15 = knnreg(price ~ mileage, data=trim350_train, k=30)
modelr::rmse(knn15, trim350_test)
```

```
## [1] 9917.012
```

```
trim350_test = trim350_test %>%
  mutate(Price_pred = predict(knn15, trim350_test))

p_test = ggplot(data = trim350_test) +
  geom_point(mapping = aes(x = mileage, y = price), alpha=0.2)

# add the predictions
p_test + geom_line(aes(x = mileage, y = Price_pred), color='red', size=1.5)
```

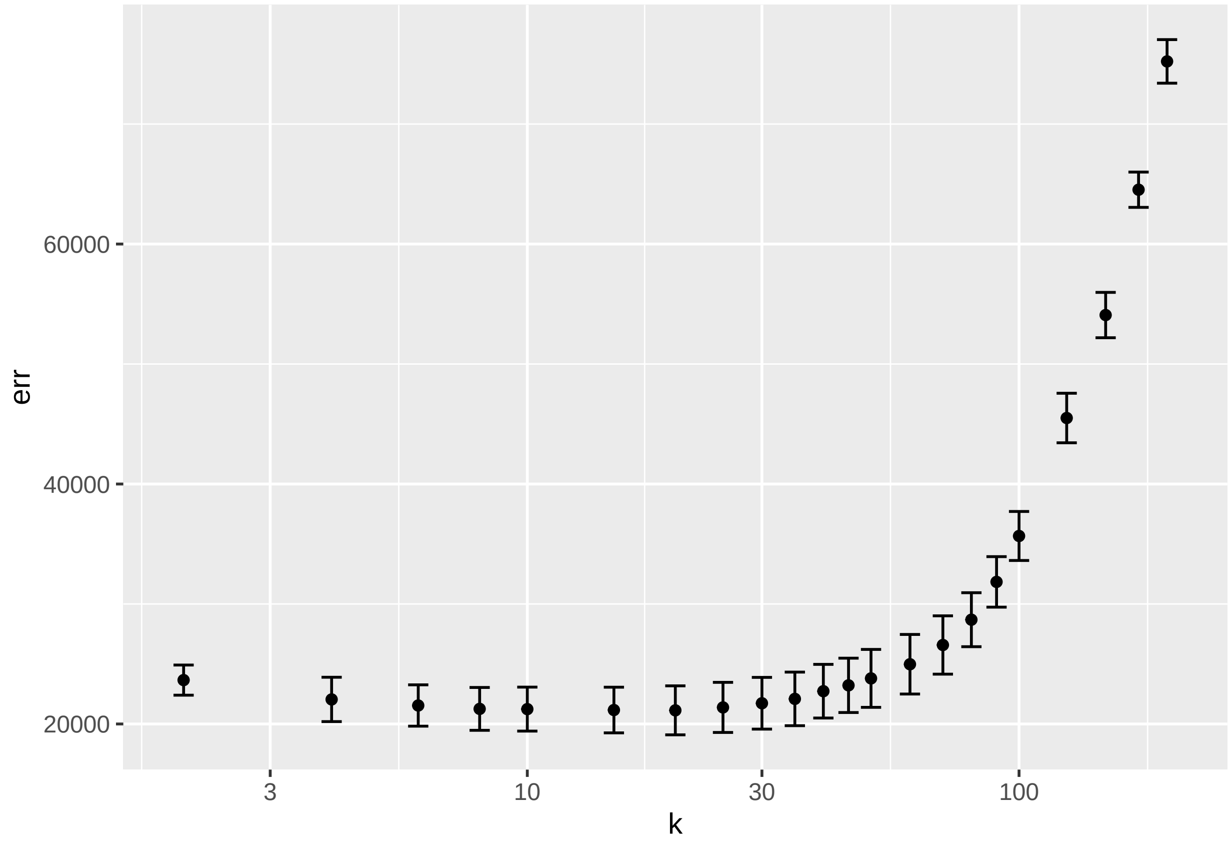
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
```



```
cv_gridAMG = foreach(k = k_grid, .combine='rbind') %dopar% {
  models = map(trimAMG_folds$train, ~ knnreg(price ~ mileage, k=k, data = ., use.all=FALSE))
  errs = map2_dbl(models, trimAMG_folds$test, modelr::rmse)
  c(k=k, err = mean(errs), std_err = sd(errs)/sqrt(k_folds))
} %>% as.data.frame

head(cv_gridAMG)
```

```
##             k      err  std_err
## result.1  2 23656.26 1256.196
## result.2  4 22045.48 1847.822
## result.3  6 21539.51 1723.042
## result.4  8 21256.56 1785.276
## result.5 10 21238.27 1831.737
## result.6 15 21161.66 1901.615
```

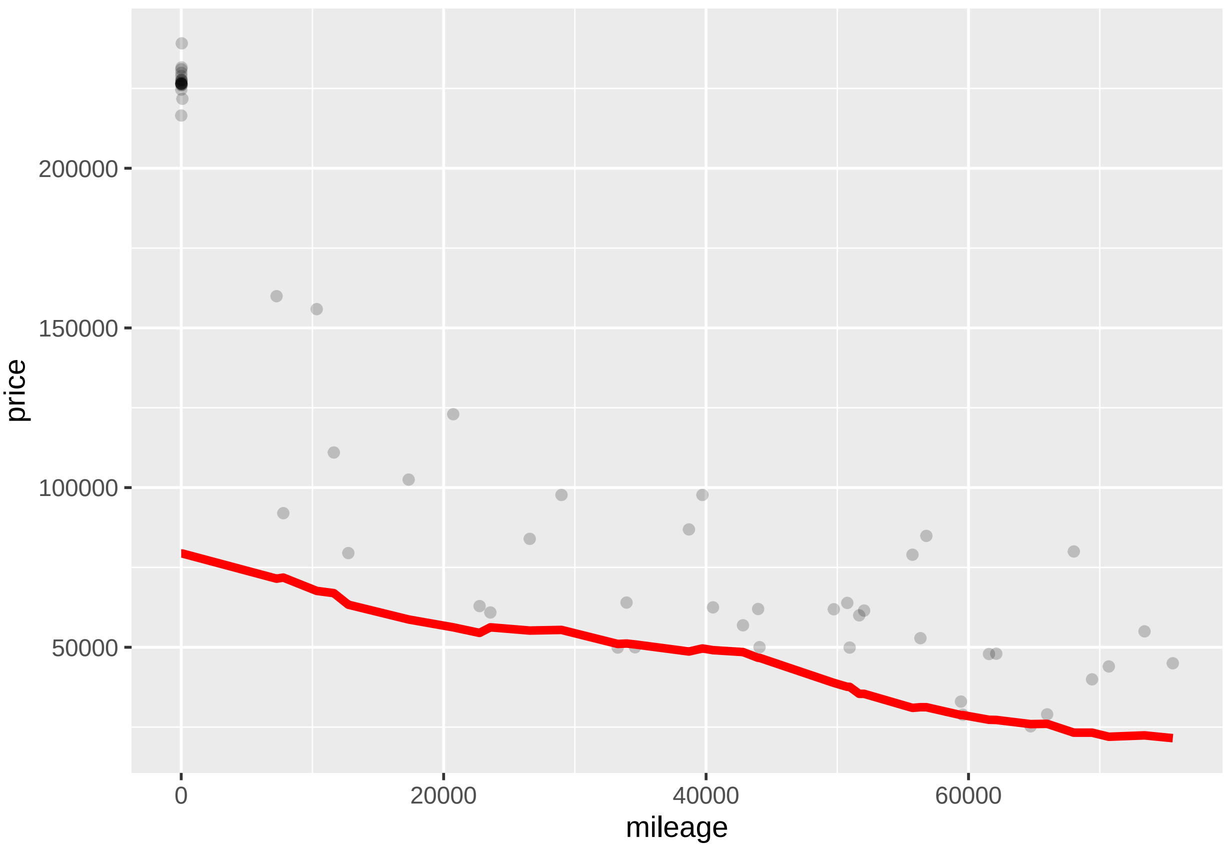
```
ggplot(cv_gridAMG) +
  geom_point(aes(x=k, y=err)) +
  geom_errorbar(aes(x=k, ymin = err-std_err, ymax = err+std_err)) +
  scale_x_log10()
```



```
knn20 = knnreg(price ~ mileage, data=trim350_train, k=30)
modelr::rmse(knn20, trimAMG_test)
```

```
## [1] 90568.55
```

```
trimAMG_test = trimAMG_test %>%
  mutate(Price_pred = predict(knn20, trimAMG_test))

p_test = ggplot(data = trimAMG_test) +
  geom_point(mapping = aes(x = mileage, y = price), alpha=0.2)

p_test + geom_line(aes(x = mileage, y = Price_pred), color='red', size=1.5)
```


200000
150000
100000
50000
price
0
20000
40000
60000
mileage