1. Data visualization: flights at ABIA

What is interesting in the ABIV.csv? I'm going to concentrate in delay of the flights and try to fond the best time for people to minimize delays.

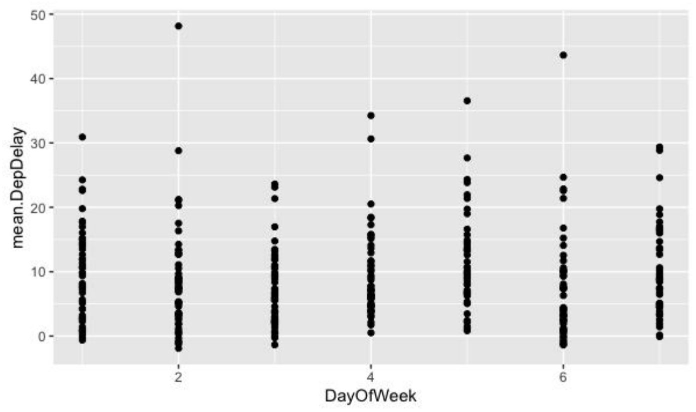
First, before using with the data, we need to remove the NA row in the data.

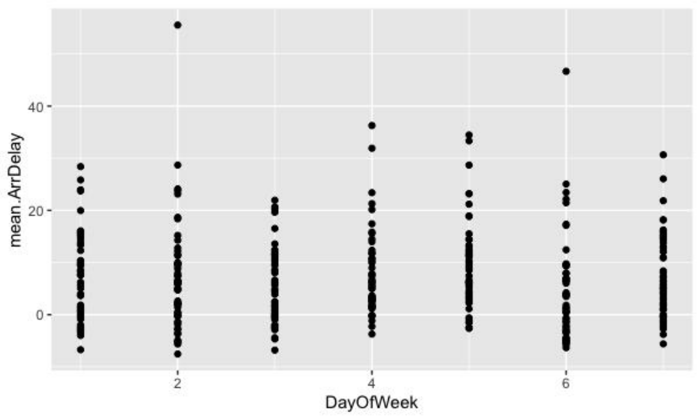
```
ABIA<-subset(ABIA,ArrDelay!="NA")
ABIA<-subset(ABIA,DepDelay!="NA")
```

◆ Day of week

```
ABIA_summ=ABIA%>%
  group_by(Month,DayofMonth,DayOfWeek)%>%
  summarise(mean.ArrDelay=mean.default(ArrDelay),mean.DepDelay=mean.default(DepDelay))

ggplot(ABIA_summ) +
  geom_point(mapping = aes(x = DayOfWeek, y = mean.DepDelay))
ggplot(ABIA_summ) +
  geom_point(mapping = aes(x = DayOfWeek, y = mean.ArrDelay))
```

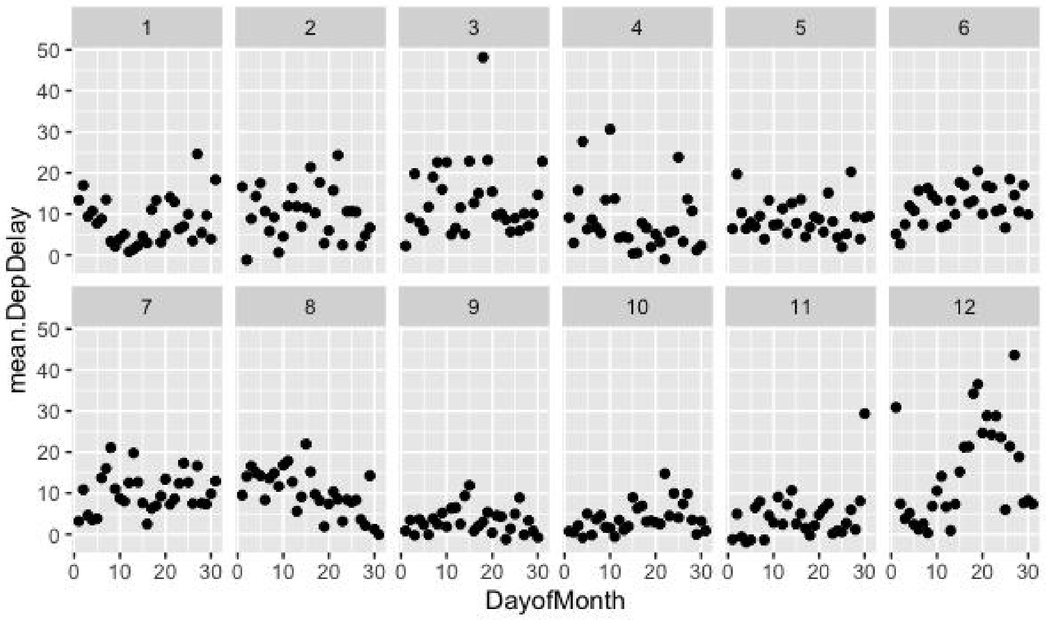


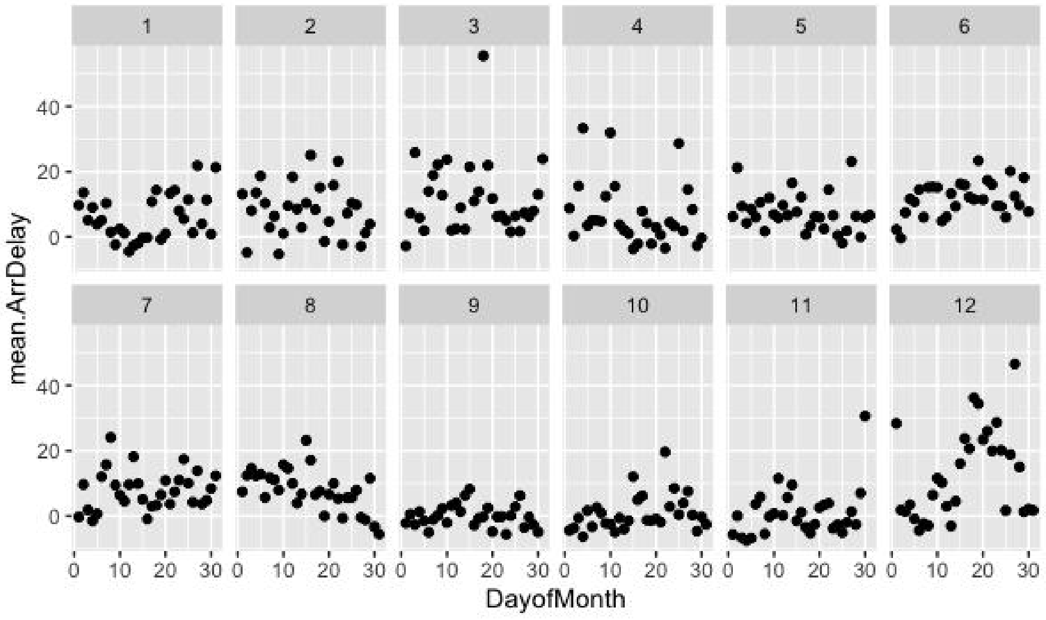


From the plot we could see that Wednesday owns both the least mean Arr-delay and Dep-delay.

- Day of month

```
ggplot(ABIA_summ) +
  geom_point(mapping = aes(x = DayOfWeek, y = mean.DepDelay))
ggplot(ABIA_summ) +
  geom_point(mapping = aes(x = DayOfWeek, y = mean.ArrDelay))
```



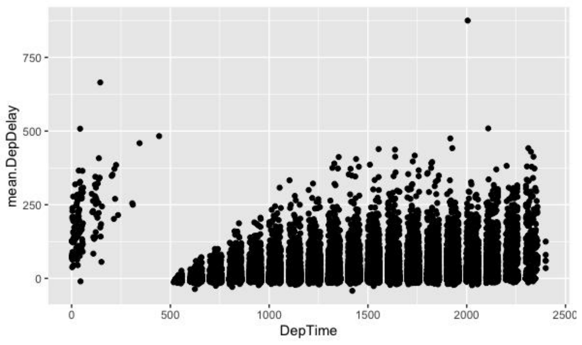


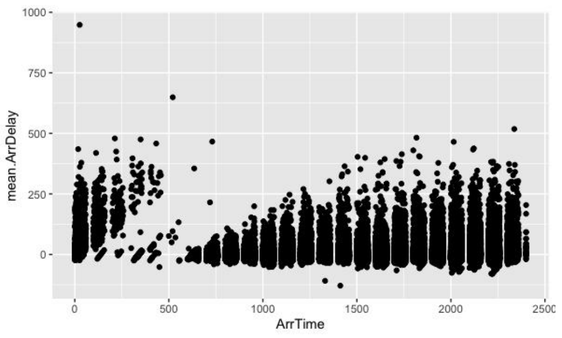
From these figures, we could see that September has the least average delay in the month compare with other months. (September and October are hard to distinguish if we only care about the Depdelay figures.

Here we only care about both the concentrate and the mean which could be obviously observed.

◆ Time of day

```
ABIA_summ=ABIA%>%
  group_by(ArrTime,DepTime,Month,DayofMonth,DayOfWeek)%>%
  summarise(mean.DepDelay=mean.default(DepDelay),mean.ArrDelay=mean(ArrDelay))
ggplot(ABIA_summ) +
  geom_point(mapping = aes(x = DepTime, y = mean.DepDelay))
ggplot(ABIA_summ) +
  geom_point(mapping = aes(x = ArrTime, y = mean.ArrDelay))
```





For different time of a day, it varies between Depdelay and Arrdelay. However, it obeys the normal rule that planes need a period to fly during the route. Thus, we could conclude from the figure that from 5 o'clock, the probability of delay tends to be higher to the end of the day.

◆ As a conclusion, from the data of 2018, it seems we could predict that during the long time of a year, the best time to minimize delays is during the earliest time of Wednesday in September. But actually, there are also other effects like weather condition or emergencies that would also contribute a lot.

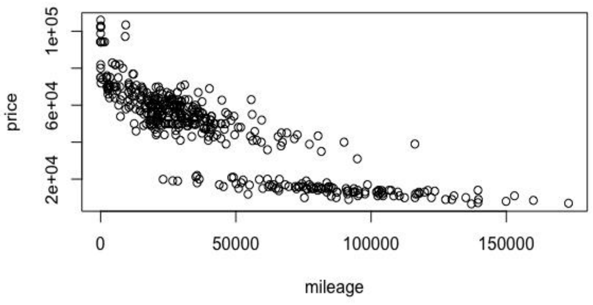
◆ Thus , I prefer to regard this as a conclusion to find the reason of delays in 2018 rather than using this data set into predicting.

2. K-nearest neighbors

For each of these two trim levels:
1.Split the data into a training and a testing set.
2.Run K-nearest-neighbors, for many different values of K, starting at K=2 and going as high as you need to. For each value of K, fit the model to the training set and make predictions on your test set.
3.Calculate the out-of-sample root mean-squared error (RMSE) for each value of K.

For each trim, make a plot of RMSE versus K, so that we can see where it bottoms out. Then for the optimal value of K, show a plot of the fitted model. (Again, separately for each of the two trim levels.)
Which trim yields a larger optimal value of K? Why do you think this is?

◆ Start with **trim=350** level.
Plots given by the exist R code.



1) Split the data into a training and a testing set.

```
N = nrow(sclass350)
N_train = floor(0.8*N)
N_test = N - N_train
train_ind = sample.int(N, N_train, replace=FALSE)

D_train = sclass350[train_ind,]
D_test = sclass350[-train_ind,]

D_test = arrange(D_test, price)
head(D_test)

X_train = select(D_train, price)
y_train = select(D_train, mileage)
X_test = select(D_test, price)
y_test = select(D_test, mileage)
```

2) Run K-nearest-neighbors, for many different values of K, starting at K=3 and using a loop to repeat the calculation then print the result. For each value of K, fit the model to the training set and make predictions on your test set.
3) Calculate the out-of-sample root mean-squared error (RMSE) for each value of K.

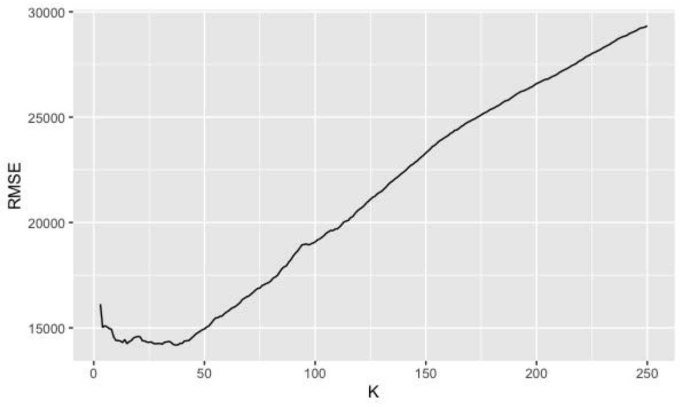
```
rmse = function(y, ypred) {
  sqrt(mean(data.matrix((y-ypred)^2)))
}

for(i in 3:250){
print(rmse(y_test,knn.reg(train = X_train, test = X_test ,y = y_train, k=i)$pred ))
}
```

Put the print value in a new table which contains different K value and the RMSE for it, name RMSE1.

4) Make a plot of RMSE versus K, so that we can see where it bottoms out.

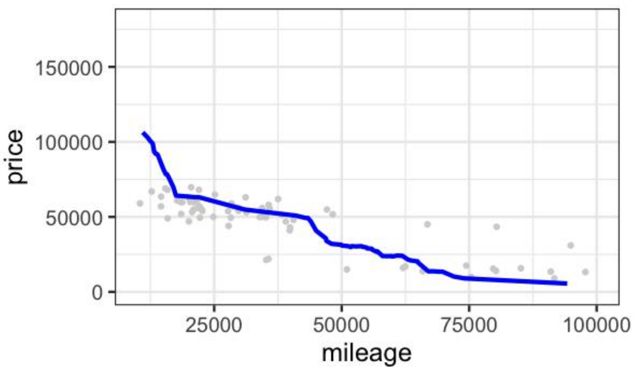
```
summary(RMSE1)
ggplot(data=RMSE1)+
geom_line(aes(x=K,y=RMSE))
```



5) Then for the optimal value of K, show a plot of the fitted model.

Here, the optimal value of K, means the K that could minimize RMSE, and under the output of this time, it is 37.

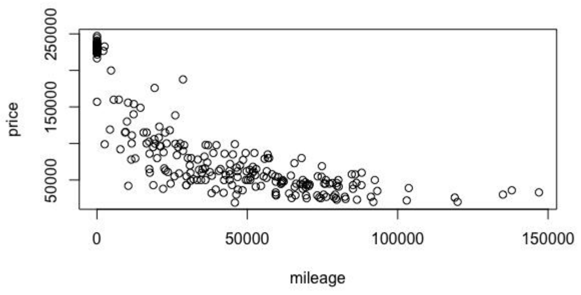
```
knn37=knn.reg(train = X_train, test = X_test ,y = y_train, k=37)
ypred_knn37 = knn37$pred
D_test$ypred_knn37 = ypred_knn37
p_test= ggplot(data = D_test) +
  geom_point(mapping = aes(x = mileage, y = price), color='lightgrey') +
  theme_bw(base_size=18)+
  ylim(0, 180000) + xlim(10000,100000)
p_test
p_test + geom_path(aes(x = price, y = ypred_knn37), color='blue',size=1.5)
```



Conclude: In this data set, we could easily see that at the beginning, RMSE value goes down and has little waves, but when K>=37, it grows quickly with little wave and looks like a linear model.
And if we look into the final result, it seems could fit the value well.

◆ Then we could concentrate on **trim=65AMG**

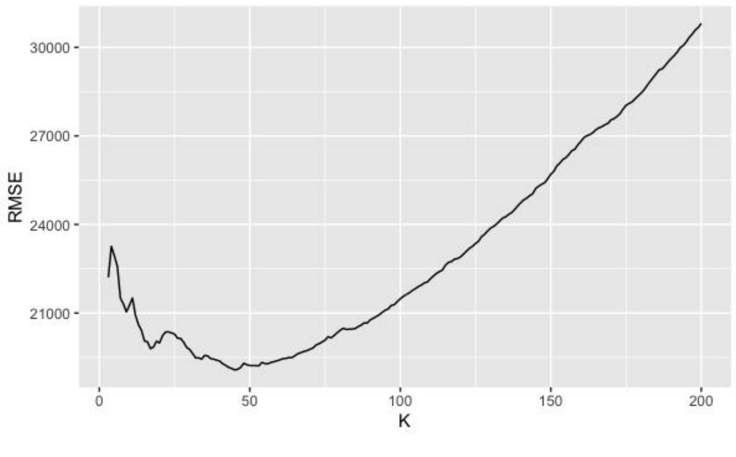
The code is almost the same just with a new data set. So I just print the plot here.



Since K could not be larger then the number of data, I repeat K from 3 to 200 here. Also, name a new table contains K and RMSE "RMSE2".

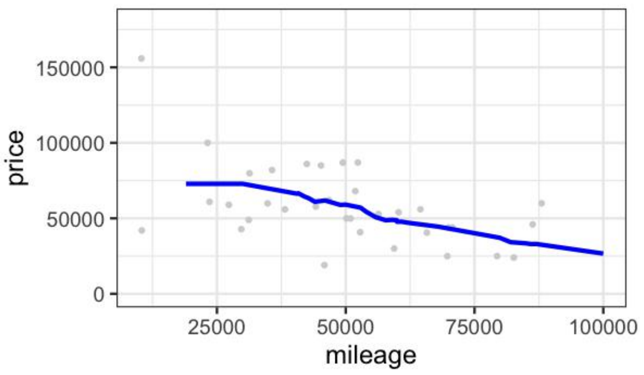
```
for(i in 3:200){
  print(rmse(y_test,knn.reg(train = X_train, test = X_test ,y = y_train, k=i)$pred ))
}

summary(RMSE2)
ggplot(data=RMSE2)+
  geom_line(aes(x=K,y=RMSE))
```



The optimal K is 45 here.

```
knn45=knn.reg(train = X_train, test = X_test ,y = y_train, k=45)
ypred_knn45 = knn45$pred
D_test$ypred_knn45 = ypred_knn45
p_test= ggplot(data = D_test) +
  geom_point(mapping = aes(x = mileage, y = price), color='lightgrey') +
  theme_bw(base_size=18)+
  ylim(0, 180000) + xlim(10000,100000)
p_test
p_test + geom_path(aes(x = price, y = ypred_knn45), color='blue',size=1.5)
```



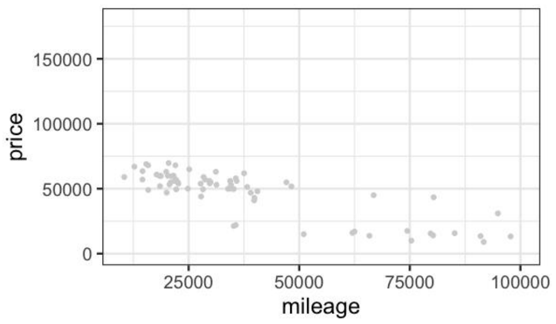
Conclude: Under this data set, the K verses RMSE plot is similar to a quadratic model, with a obvious lowest point, where k=45.
And in the final plot compare with trim=350, it seems smoother and with the same tendency of decrease.

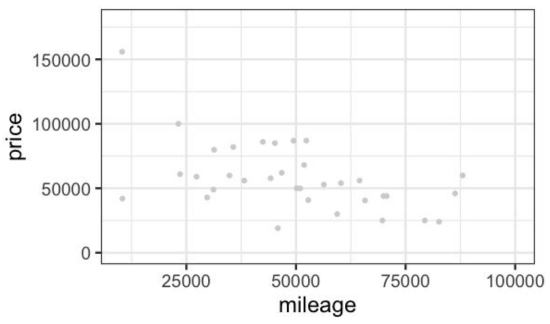
- Which trim yields a larger optimal value of K?

  Trim=350, K=37; trim=65AMG, K=45.

  Thus, trim yield 65AMG has larger optimal K value.

- Why do you think this is?





Here are the plots of two subset.

As we can observe, in same range, trim350 has more data and is more concentrated compared with trim65AMG. And if we need to predict accurately, we need to combine more neighbor data, to avoid random mistake in trim65AMG.

Thus, K in trim65AMG is larger than another one.