Prediction

Jaime Davila/Matt Richey

2/16/2021

Classification and choice of K

On a first instance we will be exploring our Minneapolis police incidents dataset in a more systematic way and construct visualizations that explore the effect of the value of K in our KNN model.

Let's start by loading the dataset

```
mn.police.tbl <- read_csv("~/Mscs 341 S22/Class/Data/police_incidents.mn.csv")
```

And remember the steps that we took last time, namely:

1. Divide our dataset into training and testing datasets.
2. Construct a model based on the training dataset.
3. Evaluate the fit of the model by calculating the MSE on the testing dataset

These steps can be done as follows:

```
# Divide dataset into testing and training
train.mn.police.tbl <- mn.police.tbl %>%
  filter(year==2016)
test.mn.police.tbl <- mn.police.tbl %>%
  filter(year==2017)

# Build the KNN model
kNear=7
knn.model <- knnreg(tot~week, data=train.mn.police.tbl,k=kNear)

# Evaluate the MSE of the KNN model in the testing dataset
test.pred <- predict(knn.model, test.mn.police.tbl)
(mse.test <- mean ((test.mn.police.tbl$tot-test.pred)^2))
```

```
## [1] 155.8458
```

We are interested in calculating systematically the MSE as we iterate over the parameter k by doing the following steps:

1. Create a function `calc_MSE(kNear, train.tbl, test.tbl)` that trains a KNN model with parameter `kNear` on `train.tbl` and then applies the model on `test.tbl` and calculates the MSE. Test your function with k=7 and k=35 using our testing and training datasets.

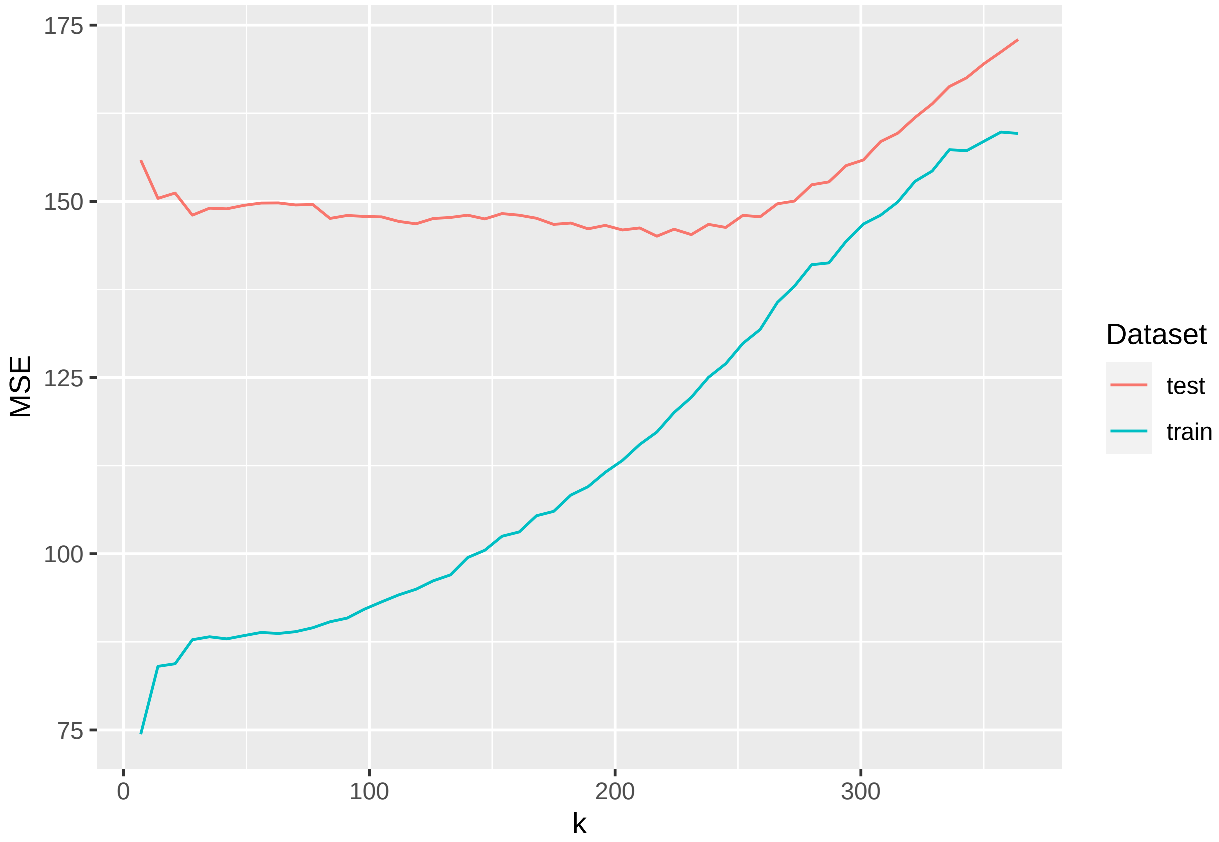
```
## [1] 155.8458
```

```
## [1] 149.0255
```

2. We would like to create a vector with the MSE values for our testing dataset. Notice it only makes sense to look at values of k in increments of 7 (why?). Use a for loop in R(Look at the syntax of for

loops in https://rafalab.github.io/dsbook/programming-basics.html#for-loops) to create the mse for $k = 7, 14, 21, ..., 364$

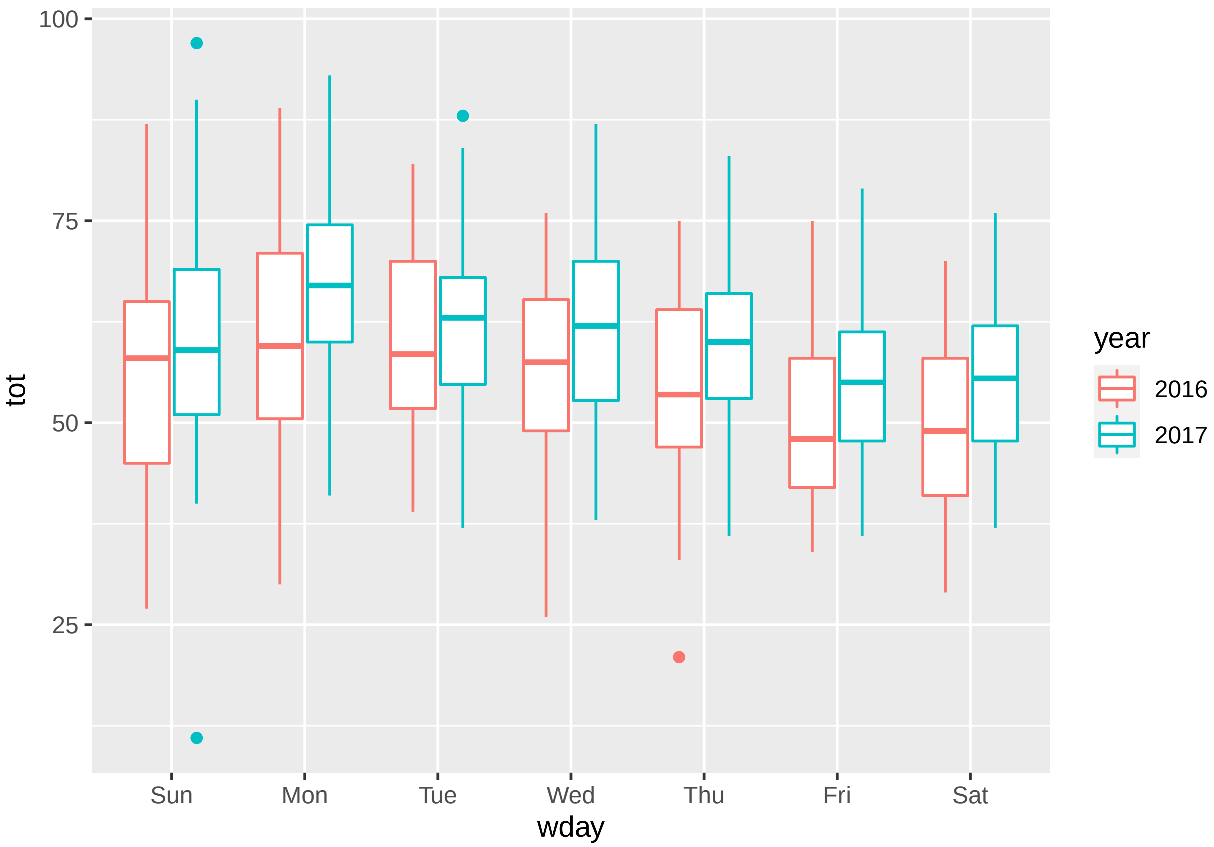
3. Generate the following graph depicting the MSE as a function of `k` for both testing and training datasets. What is the optimal value for `k` based on the testing dataset? Can you find this value in a systematic way? (*Hint*: Check the documentation for function `slice_min`). Compare this graph against figures 2.9 and 2.10 from your book. What would be the equivalent of the parameter `flexibility` in your KNN model?



Improving your model

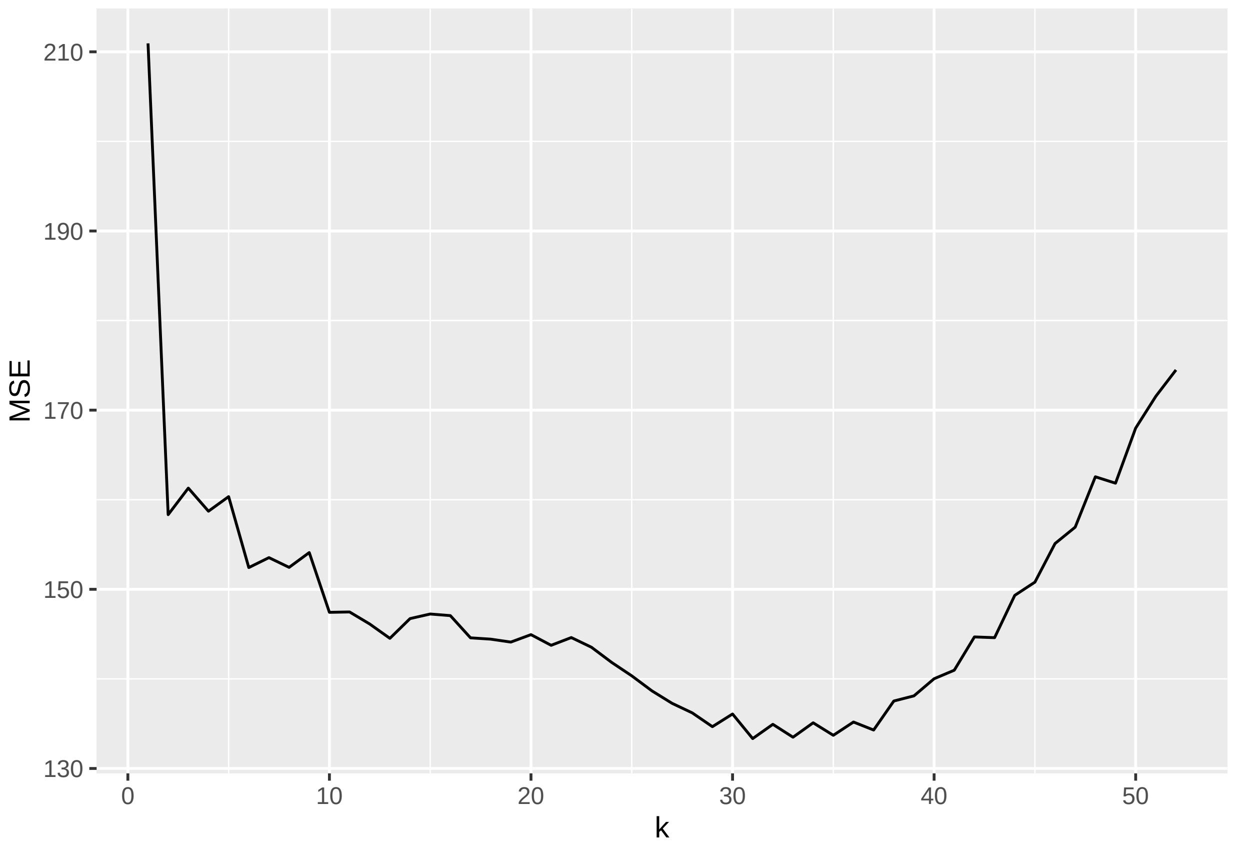
One way to improve the performance of a model is to use the information provided by other variables to create a more accurate prediction. In the following exercises we will explore this in more detail:

4. Does the distribution of number of incidents across the days of the week? Generate a boxplot to explore this question and check if this behavior is consistent across the years



5. It seems police incidents are higher on Monday as opposed to other days. Subset you dataset to only Mondays and construct a KNN model using `week` as input variable and train it using the data from 2016. Plot a graph with the MSE for all different choices of k and select a k that minimizes the MSE on the testing. Is the MSE smaller using this model than our original model?

```
## # A tibble: 1 x 2
##       k   MSE
##   <int> <dbl>
## 1    31  133.
```

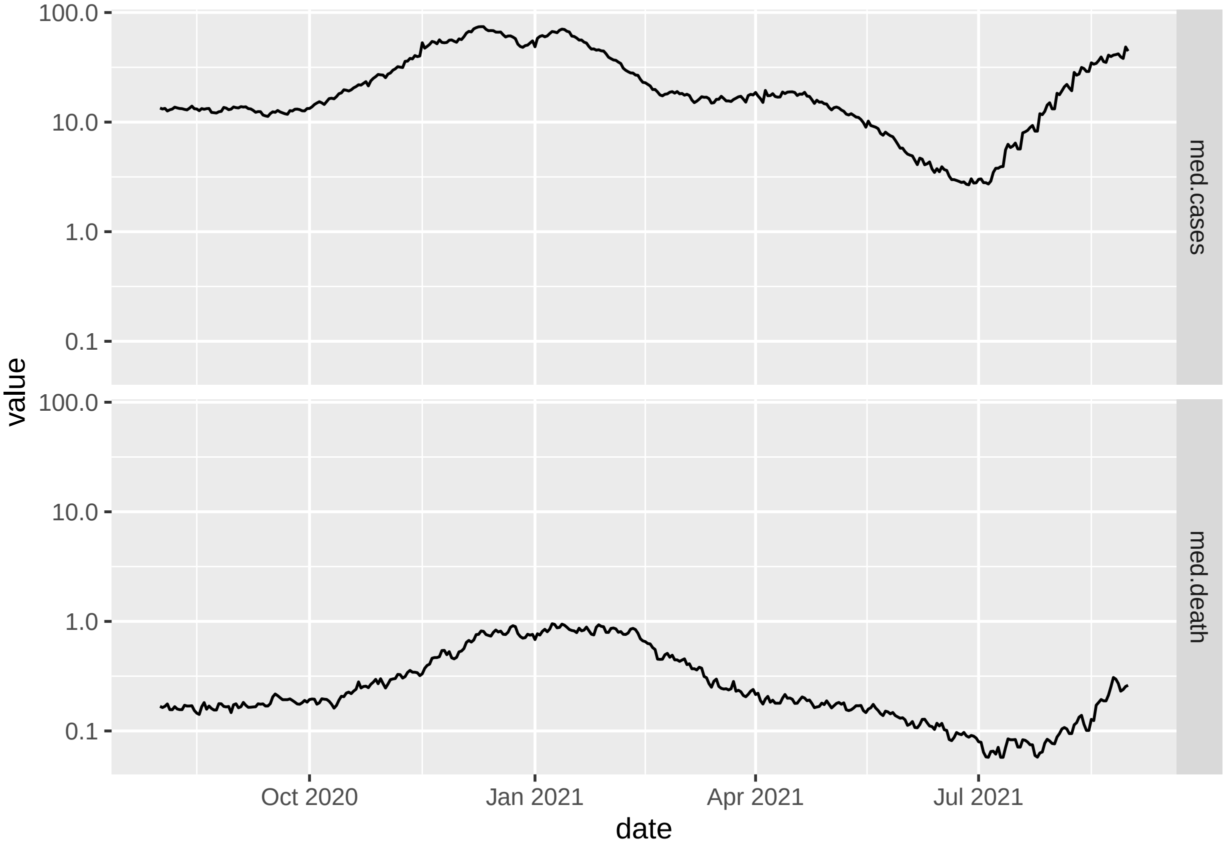


Studying the COVID pandemic

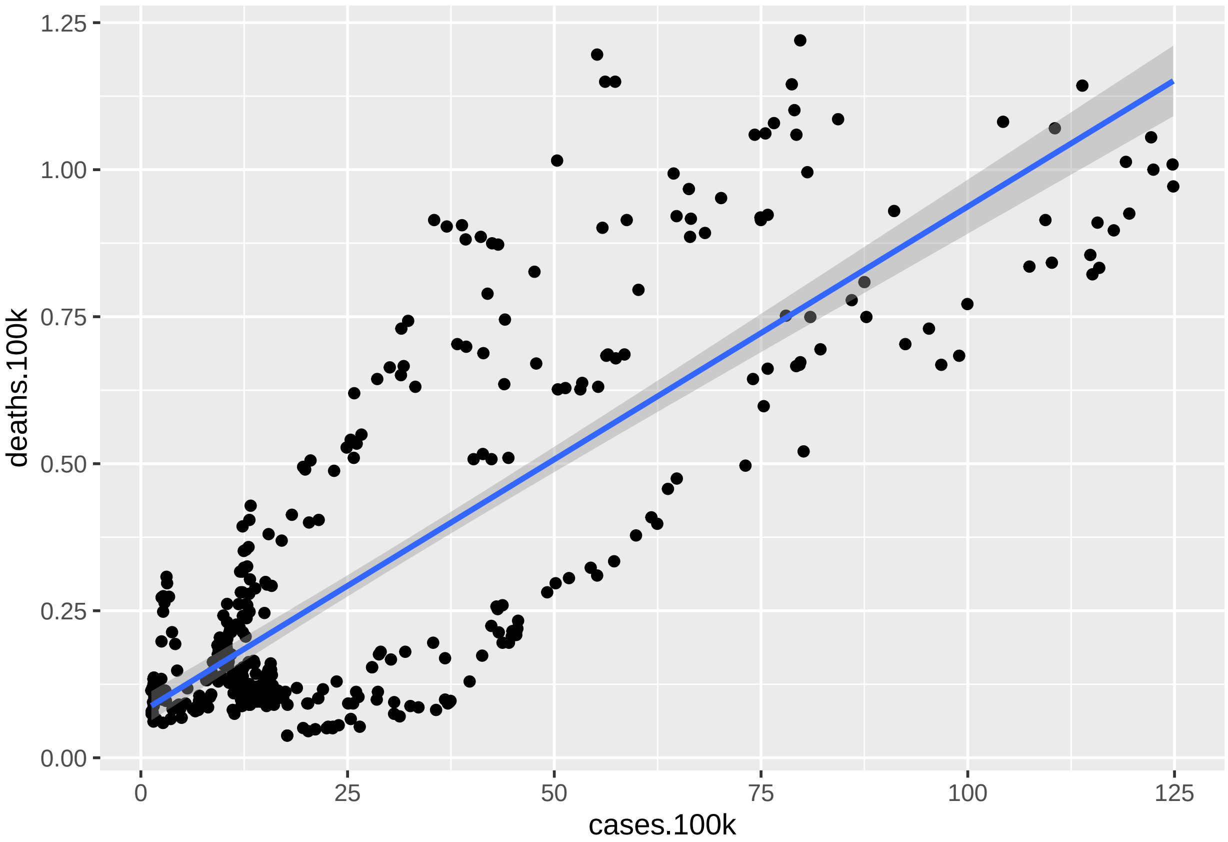
For the next set of exercises we will be using the US covid dataset procured from CovidActNow. We'll start by loading the dataset. Notice that I also loaded the library `lubridate` which allows for convenient use of dates.

```
library(lubridate)
covid.tbl <- read_csv("~/Mscs 341 S22/Class/Data/covid.csv")
```

6. Subset your dataset from August 2020 to August 2021 and plot the median number of cases and the median number of deaths (per 100,000). *Hint*: You can filter dates by using the < or >. You will need to create a reference date with `as.Date`



7. We would like to create a model that would be able to predict the number of deaths based on the number of cases. To do that let's create training and testing datasets from neighboring states, let's say WI and MN. Plot the number of cases against the number of deaths for your training dataset and include a linear trend in your plot



8. Create a linear model using `lm()` using the data from WI (training) and evaluate how well it does in MN (testing).

```
## [1] 0.03737339
```

9. To improve our model, let's make use of the fact that the number of covid cases is a good predictor of the number of deaths a couple of weeks afterwards. Create a function `calc_MSE_lag(time.lag, train.tbl, test.tbl)` that calculates the MSE on the testing dataset by using a linear model where the deaths lag the number of cases by `time.lag`. Test your function using lags of 7,14, and 21 days *Hint:* Make use of the functions `lead/lag` from the tidyverse.

```
calc_MSE_lag(7, covid.train.tbl, covid.test.tbl)
```

```
## [1] 0.02200948
```

```
calc_MSE_lag(14, covid.train.tbl, covid.test.tbl)
```

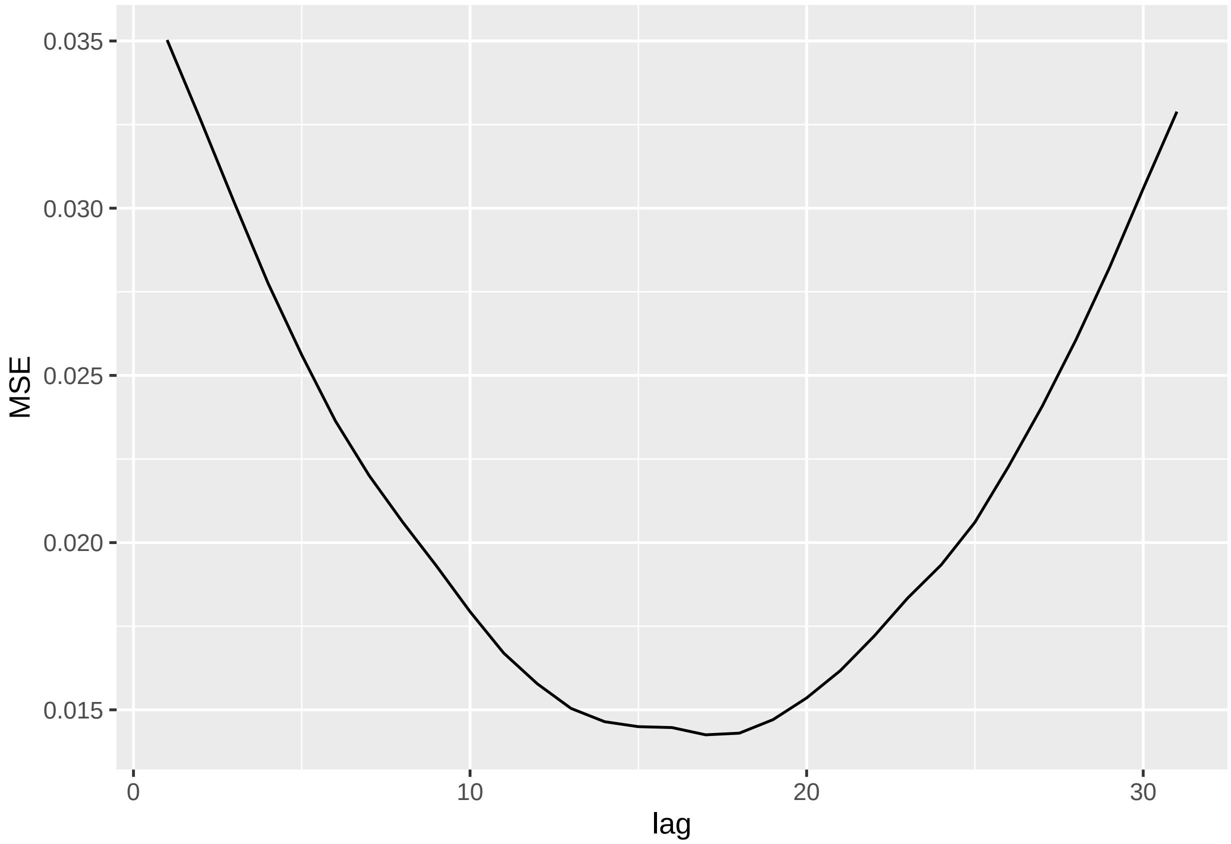
```
## [1] 0.01464344
```

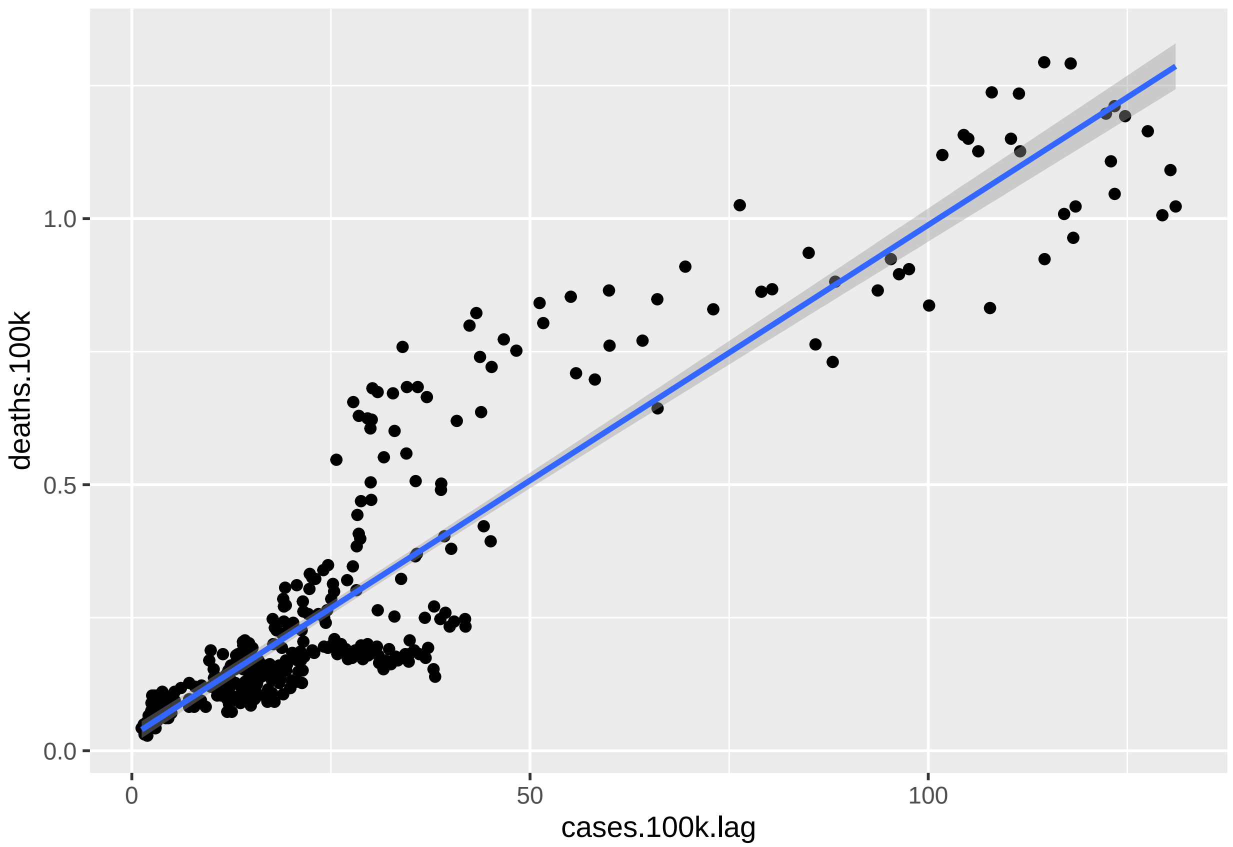
```
calc_MSE_lag(21, covid.train.tbl, covid.test.tbl)
```

```
## [1] 0.01617032
```

10. Plot lag versus MSE on the testing dataset and find the optimal parameter of lag. How do you interpret this optimal lag? Using this value of lag, plot the lagged number of cases versus the deaths and the linear trend line.

```
## # A tibble: 1 x 2
##     lag    MSE
##   <int>  <dbl>
## 1    17 0.0143
```




deaths.100k
1.0
0.5
0.0
0
50
100
cases.100k.lag