Intro to ADM

Jaime Davila/Matt Richey

2/2/2021

Introduction

We will be using a dataset with the number of police incidents in Minneapolis in 2016 and 2017. If you are interested in knowing how we generated the datasets, please look at `0_Minneapolis_crime.Rmd` in our folder. Let's start by loading the dataset

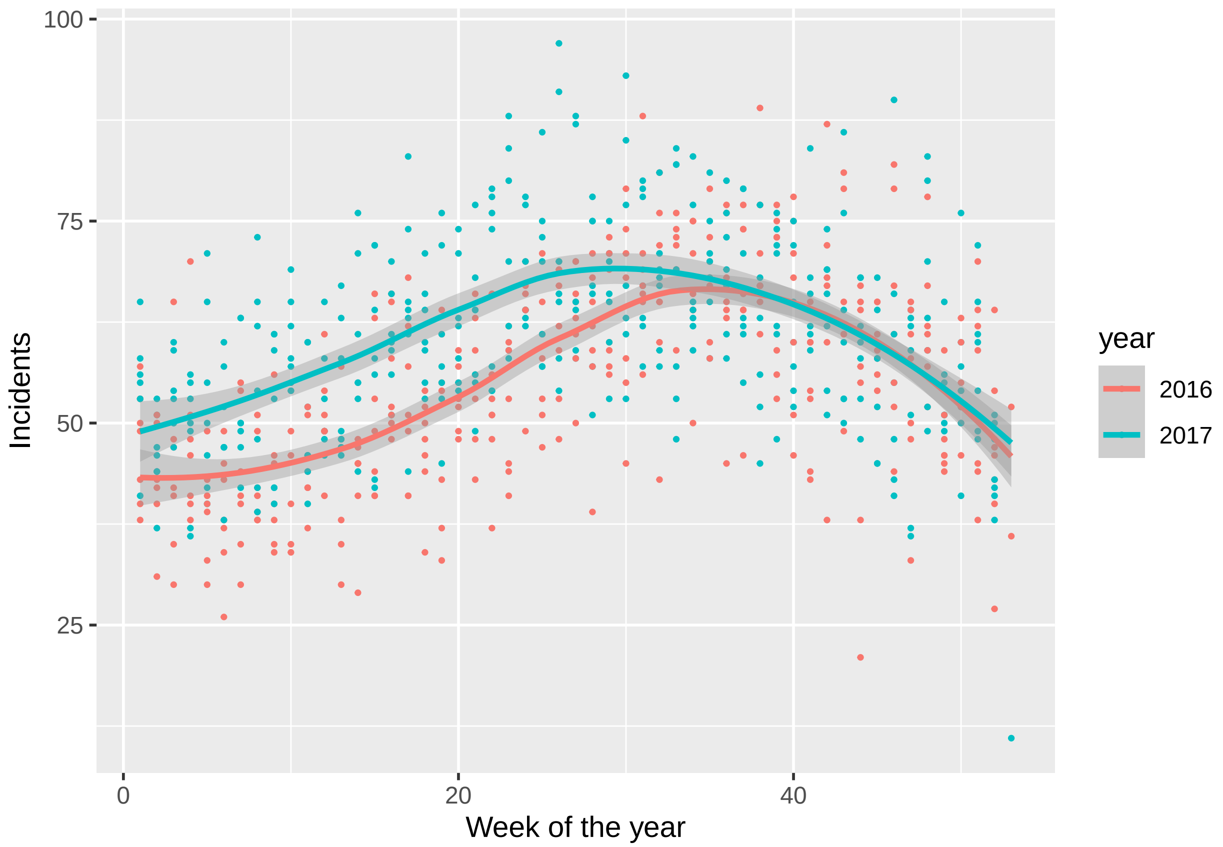
```
mn.police.tbl <- read_csv("~/Mscs 341 S22/Class/Data/police_incidents.mn.csv")
```

Our dataset only has four variables:

- `wday`: Day of the week, e.g. "Sun", "Mon", etc..
- `week`: Week in the year, e.g. 1,2,..
- `year`: Year, our dataset only has years 2016 and 2017.
- `tot`: The number of total police incidents reported on that day.

Let's start by looking at the number of incidents at different weeks of the year and use color for each year.

```
mn.police.tbl %>%
  mutate(year=as.factor(year)) %>%
  ggplot(aes(week,tot,color=year))+
  geom_point(size=.5)+
  geom_smooth()+
  labs(x="Week of the year",
       y="Incidents")
```



Notice the trends are relatively similar across the two years of our dataset.

The prediction problem and the KNN model

We are interested in predicting the number of police incidents in a particular week of the year in Minneapolis. To do that we will do the following steps:

- We will divide our dataset into a training dataset and a testing dataset. In our case we will use the year 2016 as our training and 2017 will be our testing.
- We will build a model using our training dataset.
- We will evaluate how good our model is using our testing dataset.

Our first prediction model will be "K Nearest Neighbor" (KNN) model. Given a fix k (=3, for example) and an input value, we find the k closest neighbors and use the average of their response variables as your prediction.

This can be done using the following code:

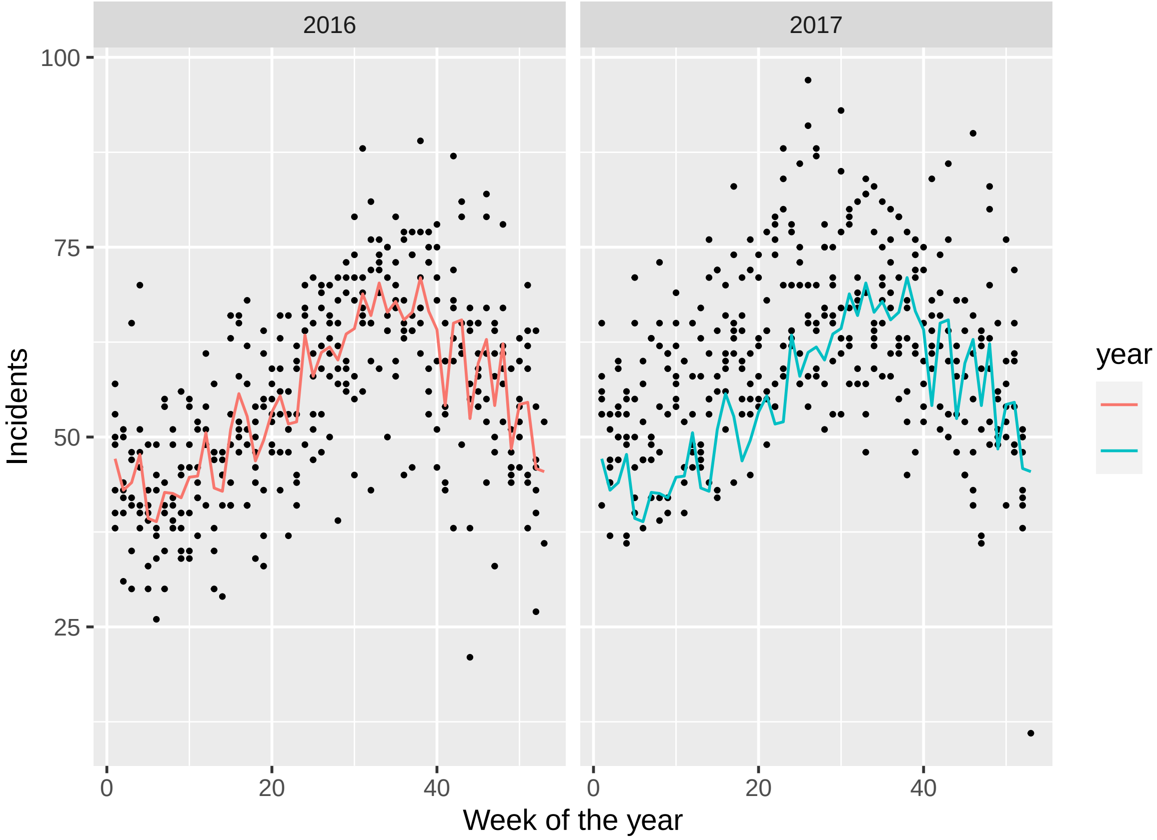
```
train.mn.police.tbl <- mn.police.tbl %>%
  filter(year==2016)
test.mn.police.tbl <- mn.police.tbl %>%
  filter(year==2017)

kNear=3
knn.model <- knnreg(tot~week, data=train.mn.police.tbl,k=kNear)
```

Notice how we use the formula `tot~week`, which means that `tot` is our response variable and `week` is our input variable.

Finally we can use the function `predict` to see how well our model works on testing dataset

```
predict(knn.model, test.mn.police.tbl)
```

1. Plot the predicted values of the KNN model on our testing and training and assess visually how close the model is to the testing and training datasets



2. A standard way to measure how good a model fits the data is to calculate the **Mean Square Error (MSE)**. Simply stated the MSE is the mean of the square of the difference between the predicted value and the actual response variable. In other words:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$$

where y_i are the actual values $\hat{y}_i$ are the predicted values.

Calculate the **MSE** for our model in the training and testing datasets separately

```
## [1] 74.38533
```

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

Using functions and different models

We would like to experiment with different values of k to see what is the effect on the fit of the model. We would like to do that using functions in R, so we will review briefly their syntax below. For more details, please consult https://r4ds.had.co.nz/functions.html

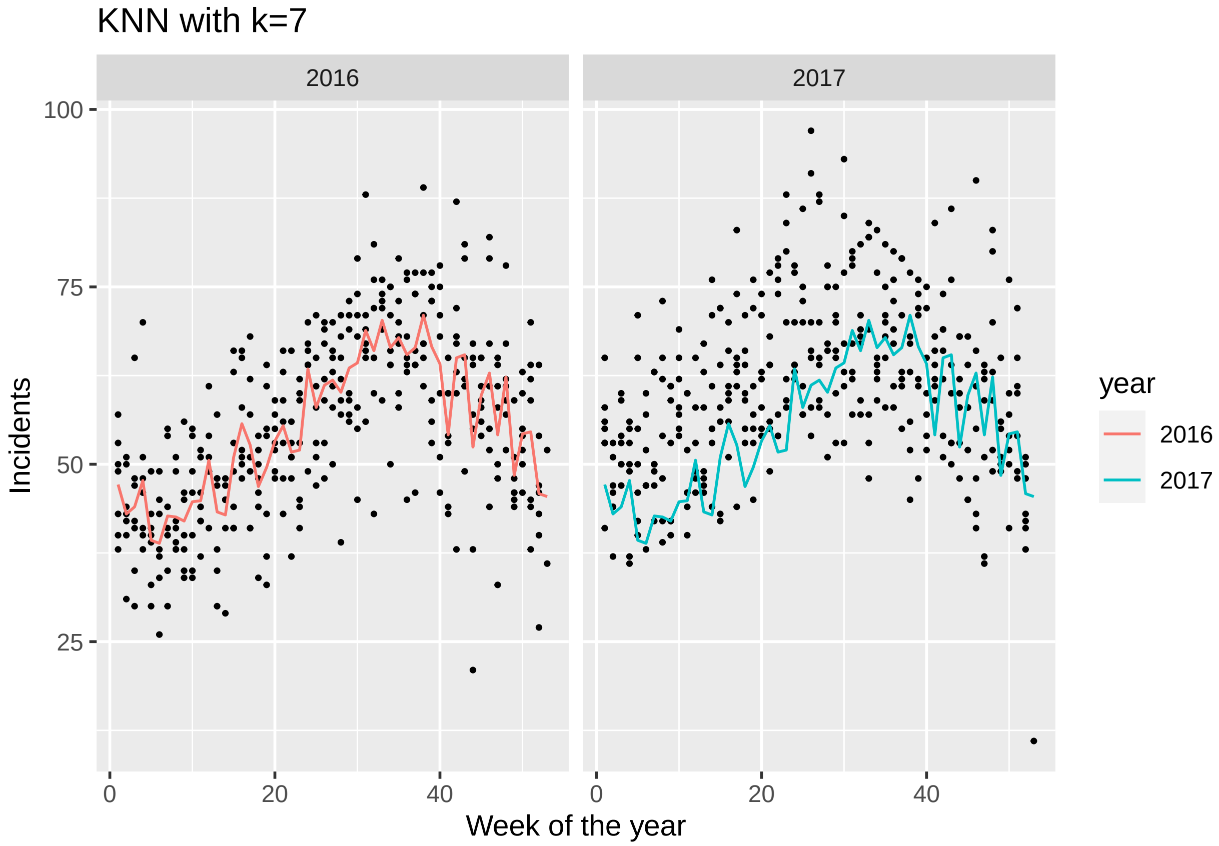
A simple function that calculates the average can be written and tested as:

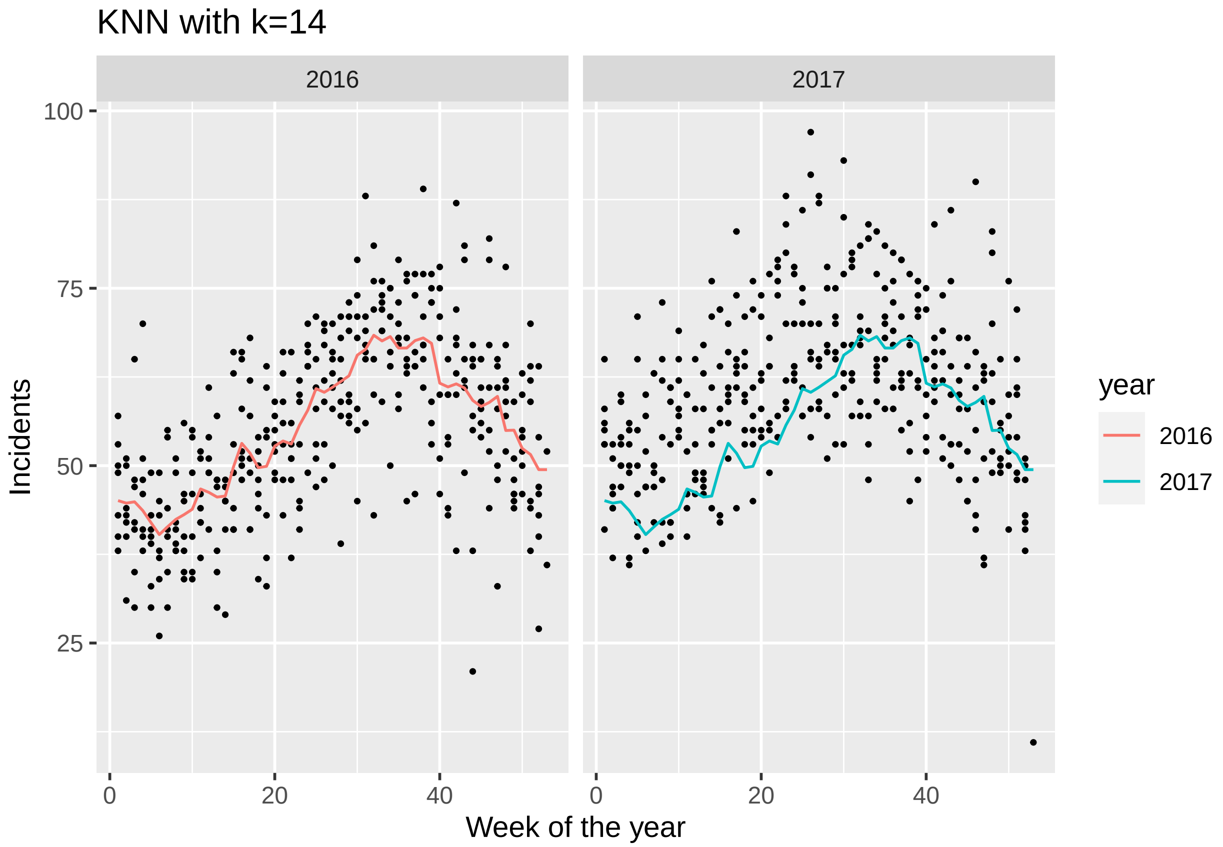
```
avg <- function(x){
  s <- sum(x)
  n <- length(x)
  s/n
}

avg(c(1,2,3,4,5))
```

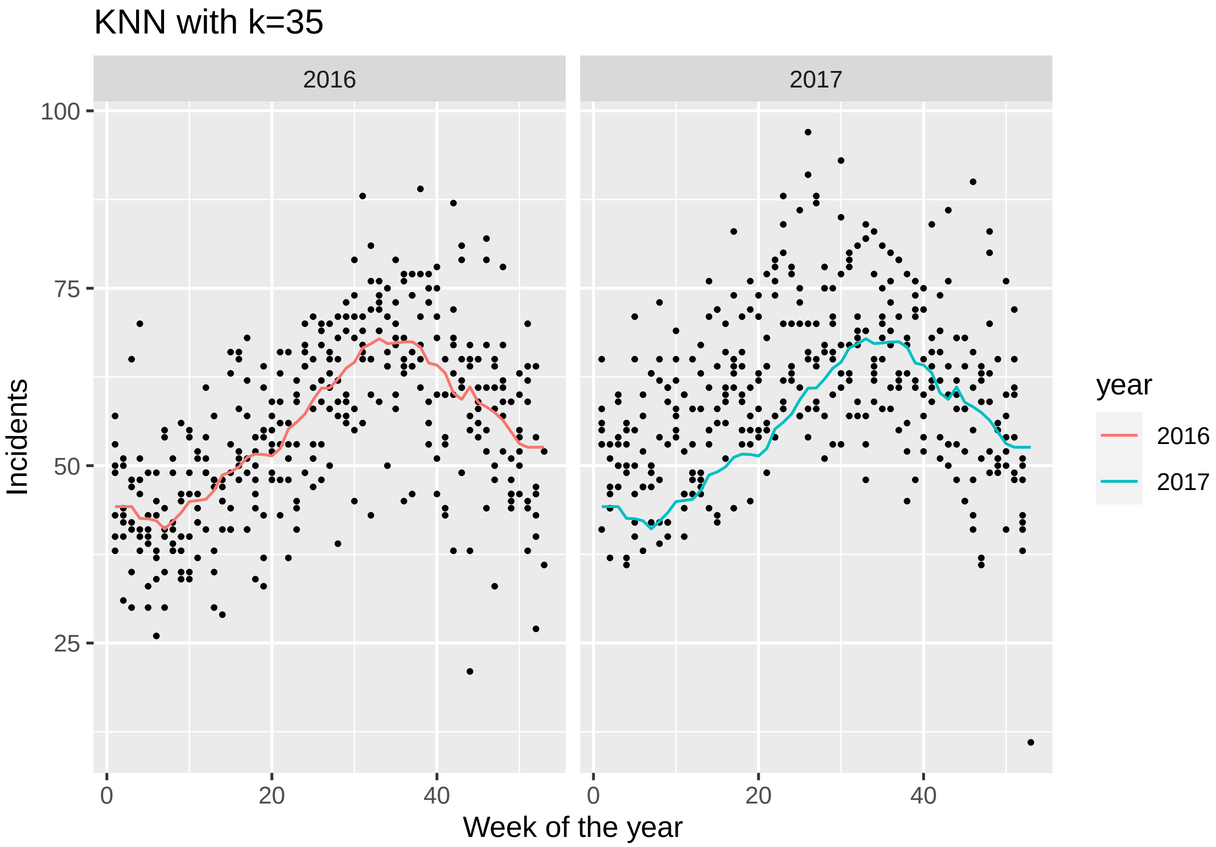
```
## [1] 3
```

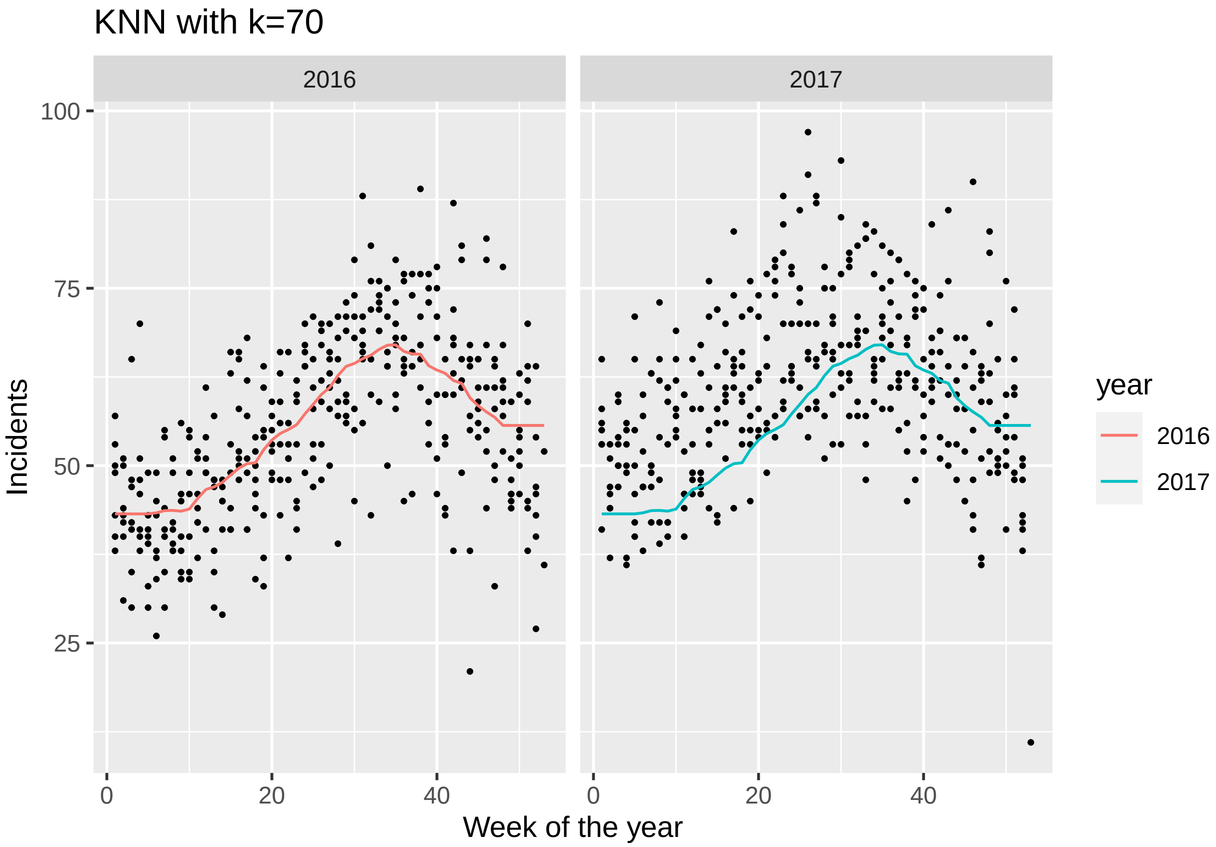
Notice how `x` is the input variable and how the return value corresponds to the last line of the function (s/n).

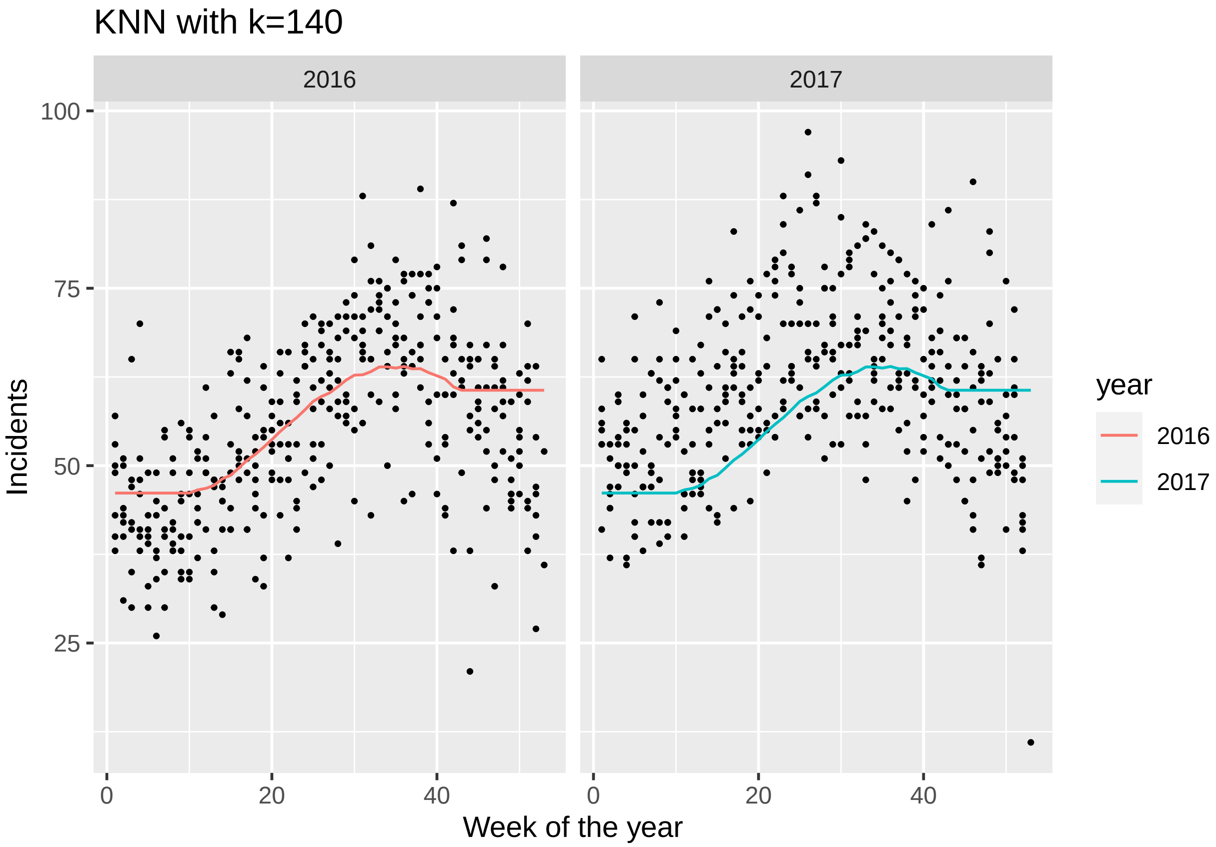
3. Create a function `plot_knn (kNear, tbl)` that returns a plot similar to 1) but for a particular value of `k` in the KNN models. Experiment with different values of `k` (7,14,35,70,140, 350) and describe the effect.

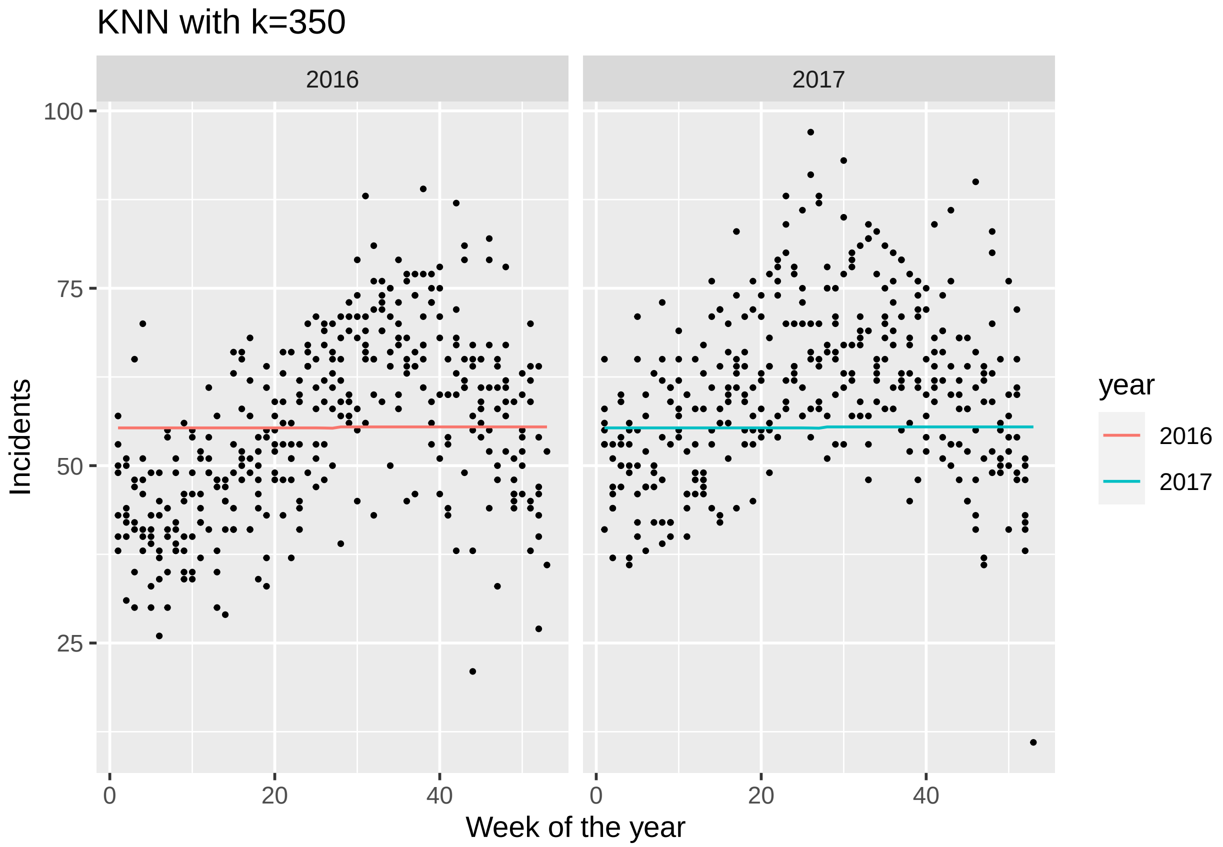



KNN with k=14
2016
2017
Incidents
100
75
50
25
0
20
40
Week of the year
year
2016
2017

KNN with k=35
2016
2017
Incidents
100
75
50
25
0
20
40
Week of the year
year
2016
2017


KNN with k=70
2016
2017
100
75
50
25
0
20
40
0
20
40
Incidents
Week of the year
year
2016
2017


KNN with k=140
2016
2017
Incidents
100
75
50
25
0
20
40
0
20
40
Week of the year
year
2016
2017



4. Create a function `calc_MSE_knn(kNear, tbl)` that returns the error in the training and testing (as vector of two values). Experiment with different values of `k` (7,14,35,70,140,350) and describe the effect of K in the MSE on the testing and the training datasets. What values makes the MSE minimum for the testing dataset? What value makes the MSE minimum for the training dataset?

```
## [1]  74.38533 155.84577
```

```
## [1]  84.02415 150.41840
```

```
## [1]  88.22261 149.02552
```

```
## [1]  88.94294 149.48290
```

```
## [1]  99.45887 148.03404
```

```
## [1] 158.5026 169.4999
```

5. (**Optional**) Would the KNN model perform better if we added the day of the week as part of our KNN model? Calculate the MSE for the testing and training dataset using k=50.

```
## [1] 86.87529
```

```
## [1] 148.0734
```