CMDA-4654

Project 1

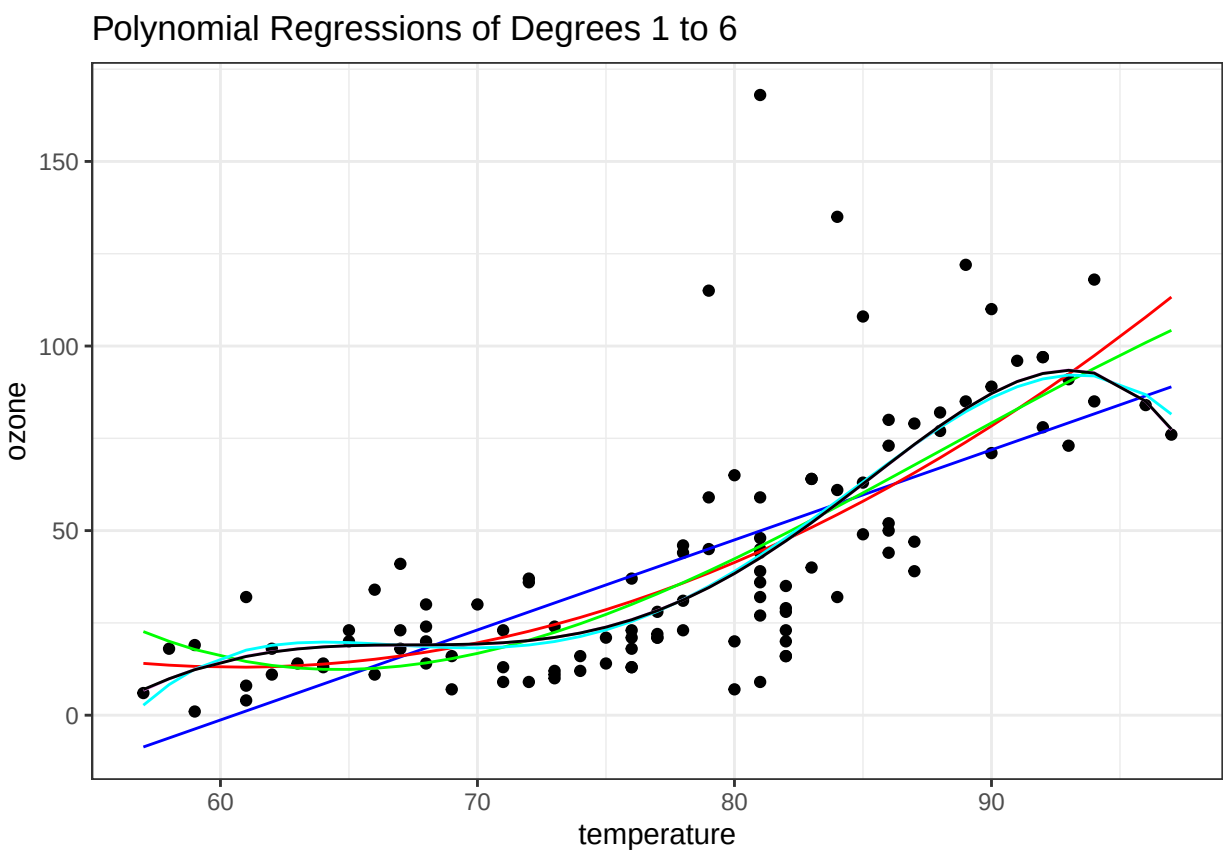
Matthew Pinho and Daniel Schlicht

November 16, 2020

Problem 1

Part 1

```
# load data
load('data/ozone.RData')
data('ozone')

fit1 <- lm(ozone ~ poly(temperature, 1, raw=TRUE), data = ozone)
ozone$p1 <- predict(fit1)
fit2 <- lm(ozone ~ poly(temperature, 2, raw=TRUE), data = ozone)
ozone$p2 = predict(fit2)
fit3 <- lm(ozone ~ poly(temperature, 3, raw=TRUE), data = ozone)
ozone$p3 = predict(fit3)
fit4 <- lm(ozone ~ poly(temperature, 4, raw=TRUE), data = ozone)
ozone$p4 = predict(fit4)
fit5 <- lm(ozone ~ poly(temperature, 5, raw=TRUE), data = ozone)
ozone$p5 = predict(fit5)
fit6 <- lm(ozone ~ poly(temperature, 6, raw=TRUE), data = ozone)
ozone$p6 = predict(fit6)

polyplot <- ggplot(ozone, aes(x = temperature, y = ozone)) + theme_bw() +
  geom_point() +
  geom_line(aes(y=p1), size = 0.5, color = 'blue') +
  geom_line(aes(y=p2), size = 0.5, color = 'red') +
  geom_line(aes(y=p3), size = 0.5, color = 'green') +
  geom_line(aes(y=p4), size = 0.5, color = 'cyan') +
  geom_line(aes(y=p5), size = 0.5, color = 'violet') +
  geom_line(aes(y=p6), size = 0.5, color = 'black') +
  ggtitle('Polynomial Regressions of Degrees 1 to 6')

polyplot.margin=margin(0.5,1,0.5,1,"cm")

polyplot
```



Answer here.

Part 2

```
## Warning in kableExtra::landscape(.): Please specify format in kable. kableExtra
## can customize either HTML or LaTeX outputs. See https://haozhu233.github.io/
## kableExtra/ for details.
```

span	degree	SSE
0.25	1	48998.90
0.30	1	50155.43
0.35	1	51181.93
0.40	1	52361.24
0.45	1	52797.53
0.50	1	53101.64
0.55	1	53421.74
0.60	1	53582.66
0.65	1	53752.42
0.70	1	53961.10
0.75	1	54192.65
0.25	2	44980.23
0.30	2	46202.13
0.35	2	46998.60
0.40	2	48411.85
0.45	2	49967.18
0.50	2	50717.64
0.55	2	51544.86
0.60	2	52004.65
0.65	2	52186.67
0.70	2	52428.99
0.75	2	52706.79

Part 3

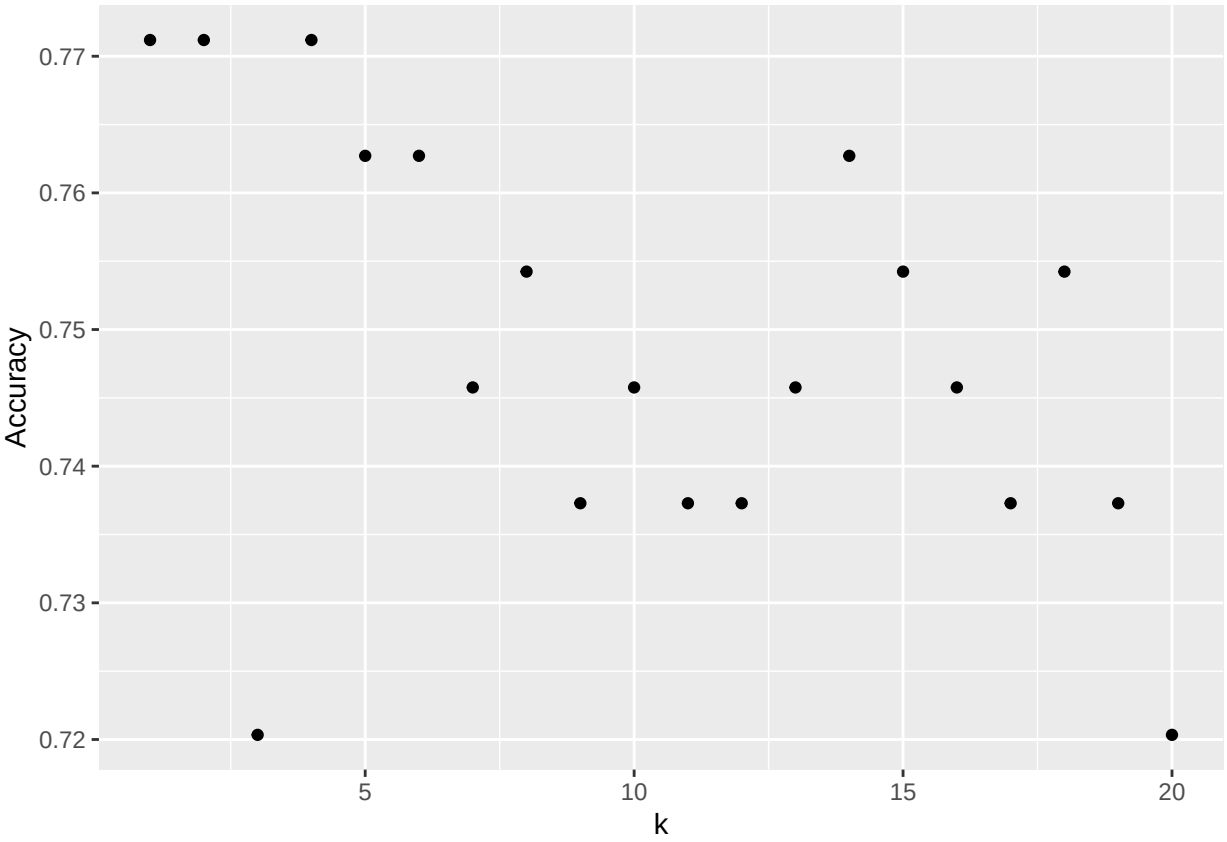
```
std_plots <- vector()
for(d in 1:2){
  for(s in seq(0.25, 0.75, 0.05)) {
    fit <- loess(ozone$ozone ~ ozone$temperature, degree = d, span = s)
  }
}
```

Problem 2

```
# Split 70/30 randomly
library(ISLR)

Auto_new <- Auto[, -9]
newOrigin <- c("USA", "European", "Japanese")
Auto_new$origin <- factor(newOrigin[Auto_new$origin], newOrigin)

set.seed(567); split = sample(1:nrow(Auto_new), 0.7*nrow(Auto_new))
train = Auto_new[split,1:ncol(Auto_new)-1]
test = Auto_new[-split,1:ncol(Auto_new)-1]
cl_train = Auto_new[split,ncol(Auto_new)]
cl_test = Auto_new[-split,ncol(Auto_new)]

acc = integer(0)
for (K in 1:20) {
  out <- mykNN(train, test, cl_train, cl_test, k=K)
  acc[K] = out$accuracy
}


accdf = data.frame("k" = 1:20, "Accuracy" = acc)

knitr::kable(accdf)
```

k	Accuracy
1	0.7711864
2	0.7711864
3	0.7203390
4	0.7711864
5	0.7627119
6	0.7627119
7	0.7457627
8	0.7542373
9	0.7372881
10	0.7457627
11	0.7372881
12	0.7372881
13	0.7457627
14	0.7627119
15	0.7542373
16	0.7457627
17	0.7372881
18	0.7542373
19	0.7372881
20	0.7203390

```
library(ggplot2)
ggplot(accdf, aes(x=k,y=Accuracy)) + geom_point()
```



It seems that for this dataset using $k = 1$ or $k = 2$ seems to give the best accuracy and for the least amount of neighbors.