stats503hw3

Ningyuan Wang

2/27/2020

Question 1

a. As lambda goes to infinity, in $\hat{g}_1$, the first derivative of g becomes a straight line; similarly, in $\hat{g}_2$, the second derivative of g becomes a straight line. In that case, $\hat{g}_2$ will have a smaller training error since it has higher degrees and therefore more flexibility.

b. As lambda goes to infinity, since we do not know the structure of testing data, we cannot tell which model will have smaller testing error.

c. For lambda = 0, $\hat{g}_1$ and $\hat{g}_2$ basically are the same, and there is no constraint on g(x_i), and we can always get zero on both $\hat{g}_!$ and $\hat{g}_2$. Therefore, this would be a perfect model fit on training data and the training error can be zero, but the model may not smooth (i.e very wiggly), which potentially caused overfitting problem on training data. For testing data set, the two models will give same testing error rate but we won't know how large or small of the testing error since no information about testing set.

Question 2

a.

In the linear regression model, all predictors are significant. R squared is 0.7174, which means around 71.74% variation of cubic root of ozone can be caught with the model. Based on the model diagnosis plots, this is a valid linear model for data and no violation of model assumptions. Overall, this is a good fit. The testing error of the model is 0.318.

```
dat = read.table("ozone_data.txt", header = T)
dat$ozne_cubr = dat$ozone^(1/3)
summary(dat$ozne_cubr) # check missing values
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.621   3.141   3.248   3.958   5.518
```

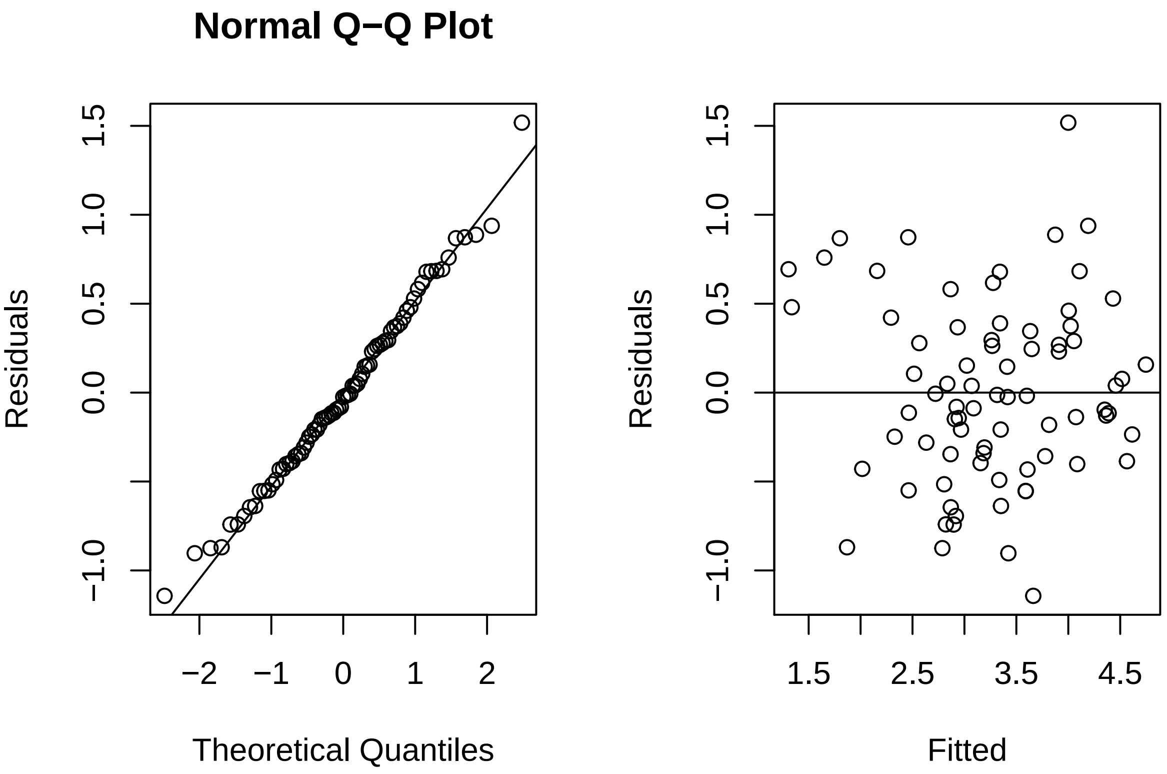
```
# split dataset
train_id = sample(1:nrow(dat), floor(0.7*nrow(dat)))
train = dat[train_id, ]
test = dat[-train_id,]

lm.fit = lm(ozne_cubr ~ temperature + wind + radiation, data = train )
summary(lm.fit)
```

```
##
## Call:
## lm(formula = ozne_cubr ~ temperature + wind + radiation, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.14281 -0.35754 -0.02483  0.34558  1.51792
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.6901796  0.6778639  -1.018 0.311961
## temperature  0.0540850  0.0076927   7.031 9.14e-10 ***
## wind        -0.0669379  0.0177700  -3.767 0.000332 ***
## radiation    0.0022555  0.0007117   3.169 0.002235 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5222 on 73 degrees of freedom
## Multiple R-squared:  0.6999, Adjusted R-squared:  0.6876
## F-statistic: 56.75 on 3 and 73 DF,  p-value: < 2.2e-16
```

```r
# check model assumpiton
par(mfrow=c(1,2))
qqnorm(lm.fit$residuals, ylab = "Residuals")
qqline(lm.fit$residuals) # normality is good

plot(lm.fit$fitted.values, lm.fit$residuals, xlab = "Fitted", ylab = "Residuals")
abline(h=0) # constant variance
```



```r
# calculate testing error
prd_test = predict(lm.fit, test)
lm_test_err = mean((test$ozne_cubr - prd_test)^2) #0.186826
```

b.

Above linear model has significant advantanges in terms of interpretation and inference. However, it also has significant limitations in terms of prediction power. Therefore, we try to fit a generalized additive model (GAM) to relax linear limitation and achieve a non-linear fit.

We fit a GAM in order to predict the cubic root of ozone concentration using smoothing splines on predicotrs temprature, radiation and wind speed. We apply packae "mgcv" to help us decide the optimal degree of freedom based on the rule of minimizing cross validation MSE.

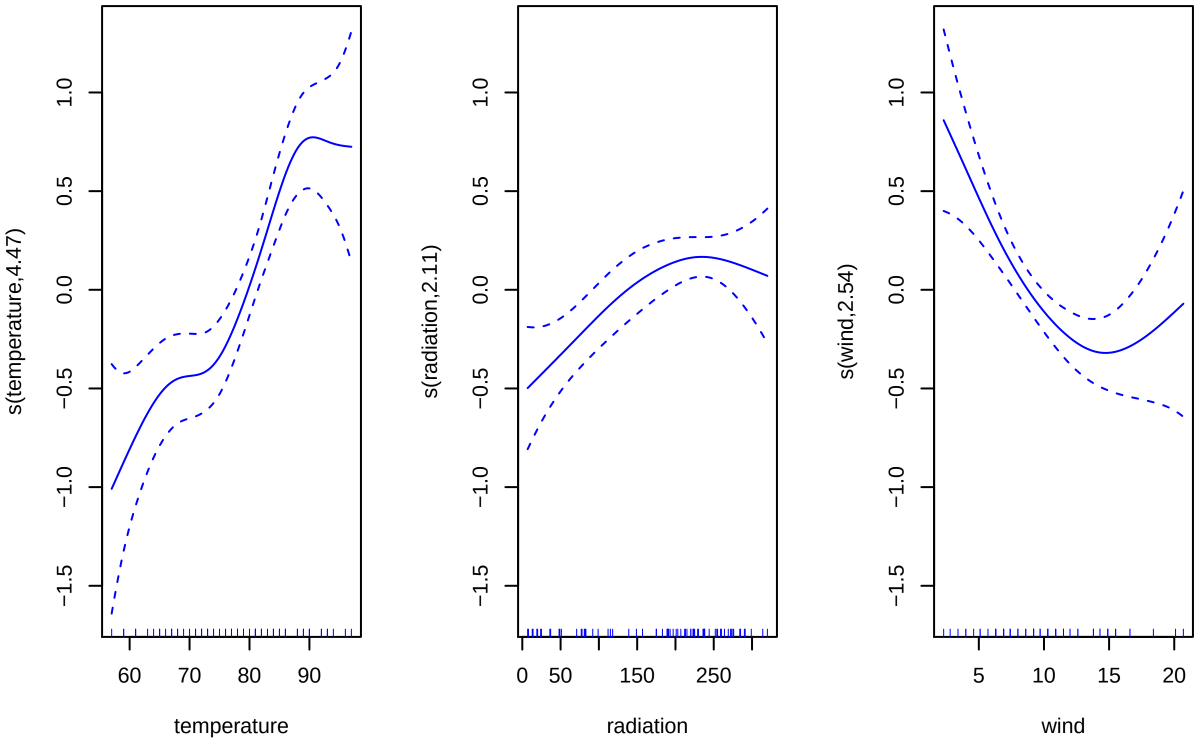
```r
library(gam)
library(mgcv)
```

The plots below show the relationships between the response (cubic root of ozone) and each predictor variable. Based on the plot and summary of the model, we think the model is a good fit of training data set since all predictions are fell in the confidence intervals for each predictor. Also, all smoothing splines on predictors are significant in the model. The GAM has a R-squared value with 73.3% which indicated a good fit.

```r
# smoothing spline approach
gam2=gam(ozne_cubr~s(temperature)+s(radiation)+s(wind), method = "GCV.Cp", data= train)
summary(gam2)
```

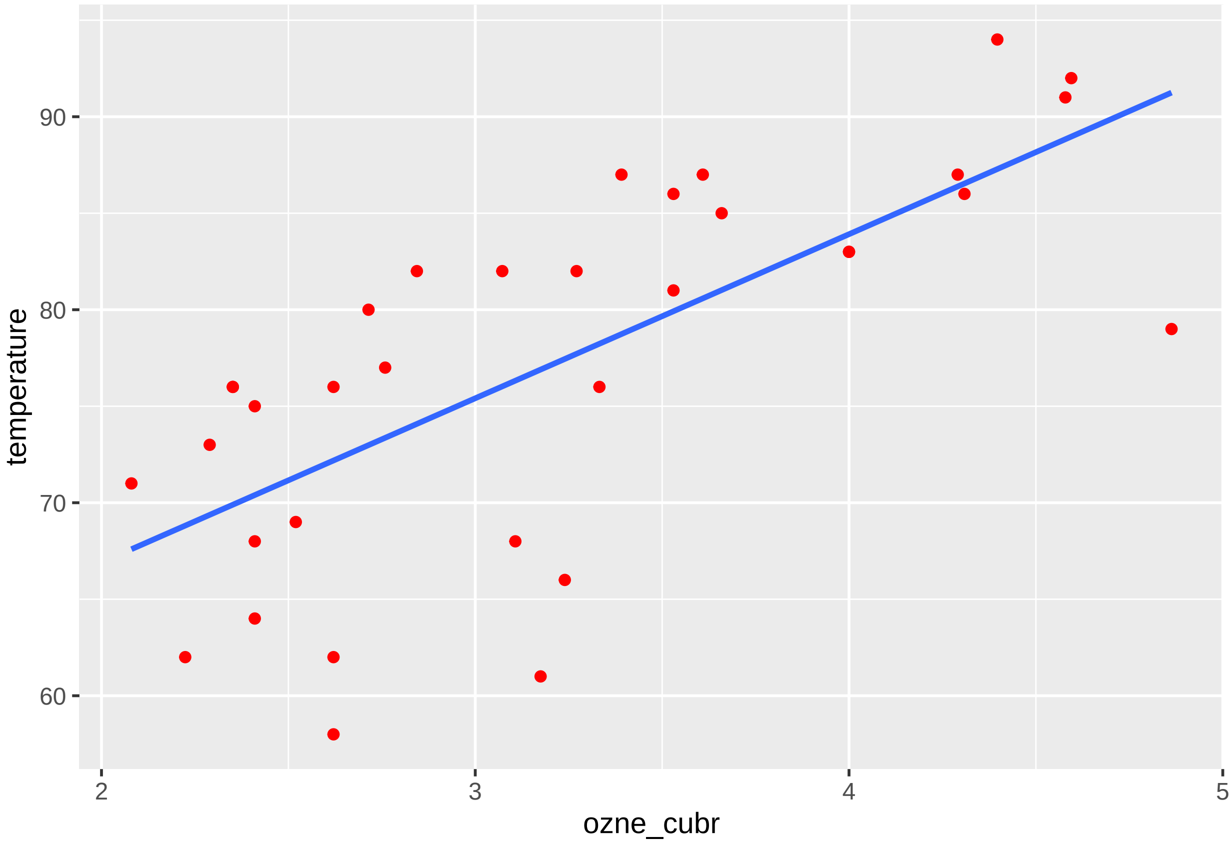
```
## 
## Family: gaussian 
## Link function: identity 
## 
## Formula:
## ozne_cubr ~ s(temperature) + s(radiation) + s(wind)
## 
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  3.26400    0.05325    61.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Approximate significance of smooth terms:
##                  edf Ref.df      F  p-value    
## s(temperature) 4.467  5.474 10.281 5.69e-08 ***
## s(radiation)   2.113  2.648  5.145 0.004095 ** 
## s(wind)        2.536  3.197  6.844 0.000309 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## R-sq.(adj) =   0.75   Deviance explained =   78%
## GCV = 0.25137  Scale est. = 0.21835   n = 77
```

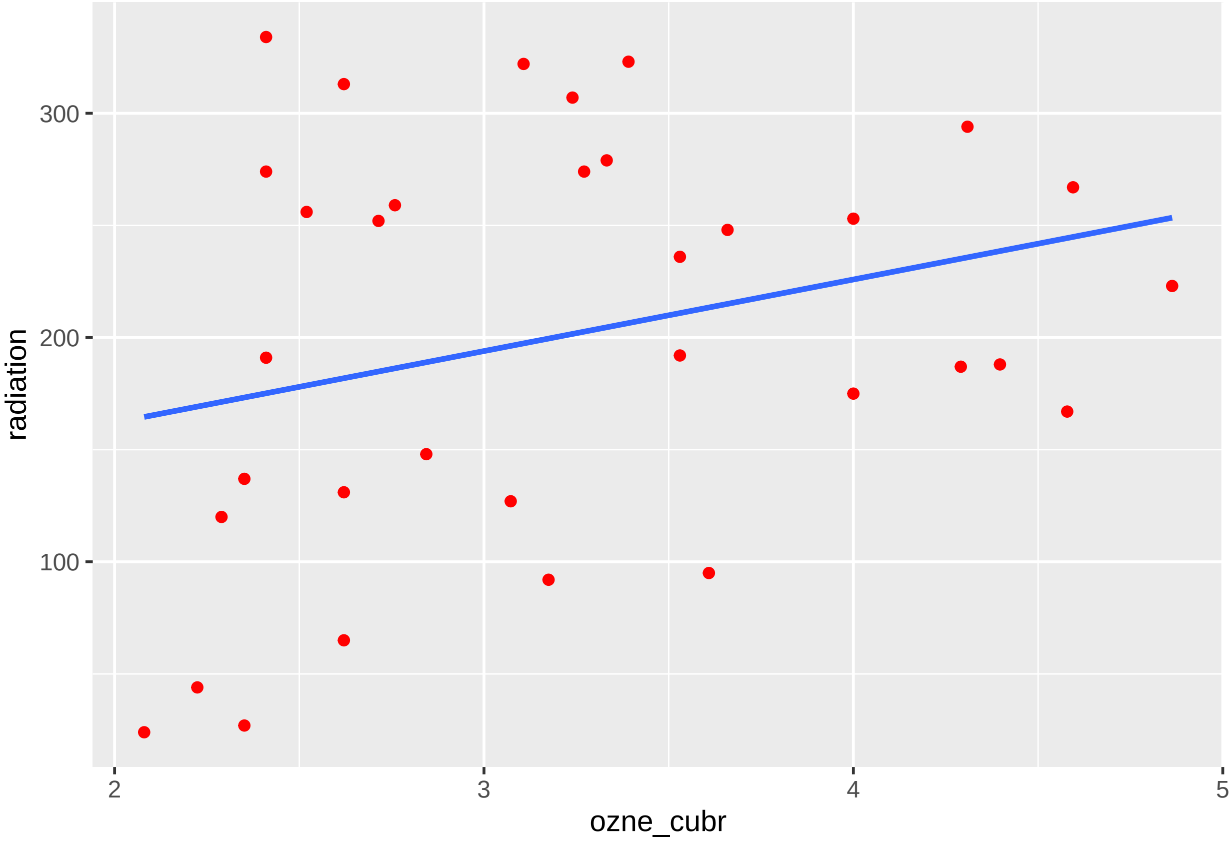
```r
par(mfrow=c(1,3))
plot(gam2 ,col="blue")
```

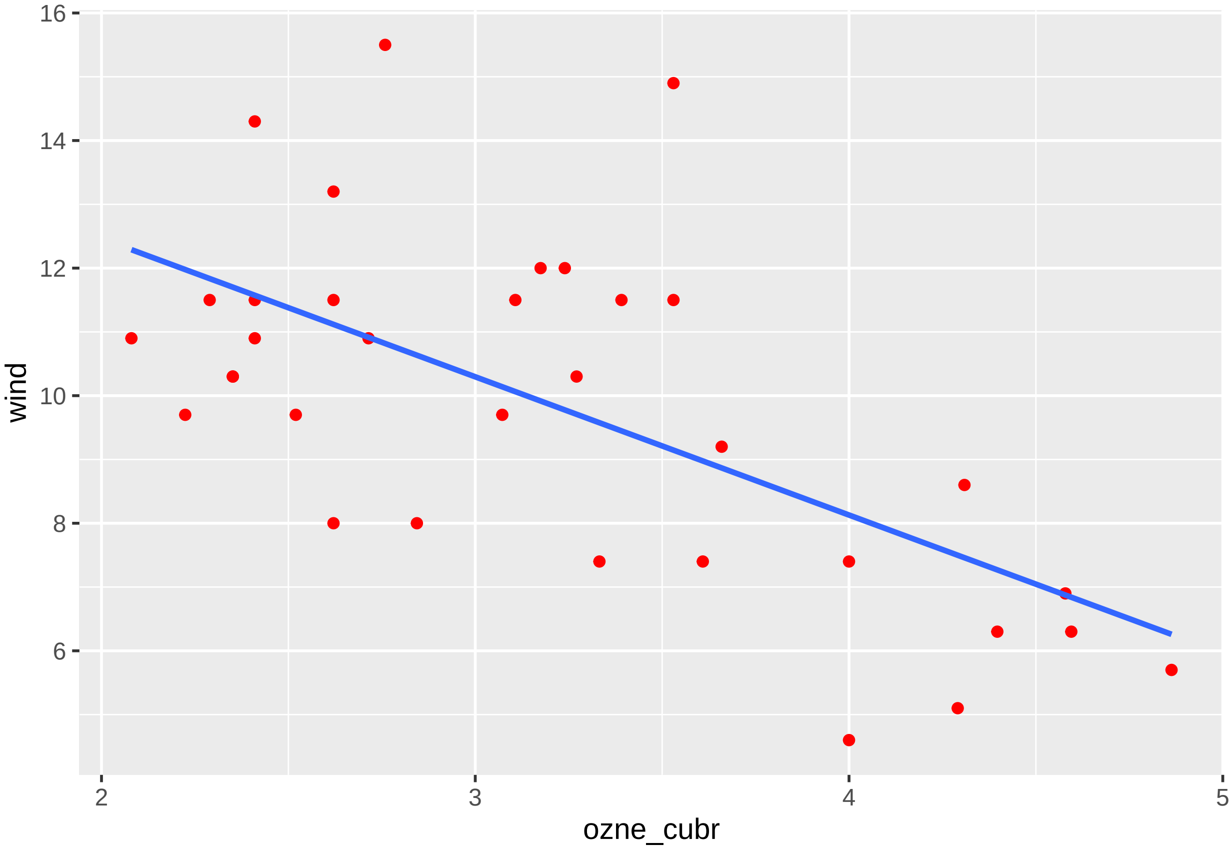


c.

Based on two model appproaches (i.e. linear and non-linear models), linear regression and GAM offer similar testing error: 0.185 for linear regression and 0.161 for GAM. Therefore, we conclude two mdoel approaches worked well to the data.


temperature
90
80
70
60
2
3
4
5
ozne_cubr


radiation
300
200
100
2
3
4
5
ozne_cubr



Based on the above plot sets of single predictor vs response for GAM and linear regression models, we conclude that variables temperature and wind speed have relative non-linear relationships to the response variable. However, their non-linearities were not obvious based on the comparison to the plots on linear regression model. Unremarkable non-linearity of predictors in GAM could be the reason that both linear model and GAM worked for data.