Problem Set 2

Sravan Ramaswamy

2/7/2021

https://github.com/SravanjR/bc-micro-methods/blob/main/psets/pset_np_regression/Problem-Set-2.pdf

```
#Clear Environment
rm(list = ls())

#Load Libraries
library(MASS)
library(ggplot2)
library(evd)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(patchwork)
```

```
##
## Attaching package: 'patchwork'
```

```
## The following object is masked from 'package:MASS':
##
##     area
```

```
library(statar)
library(stats)
library(splines)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:evd':
```

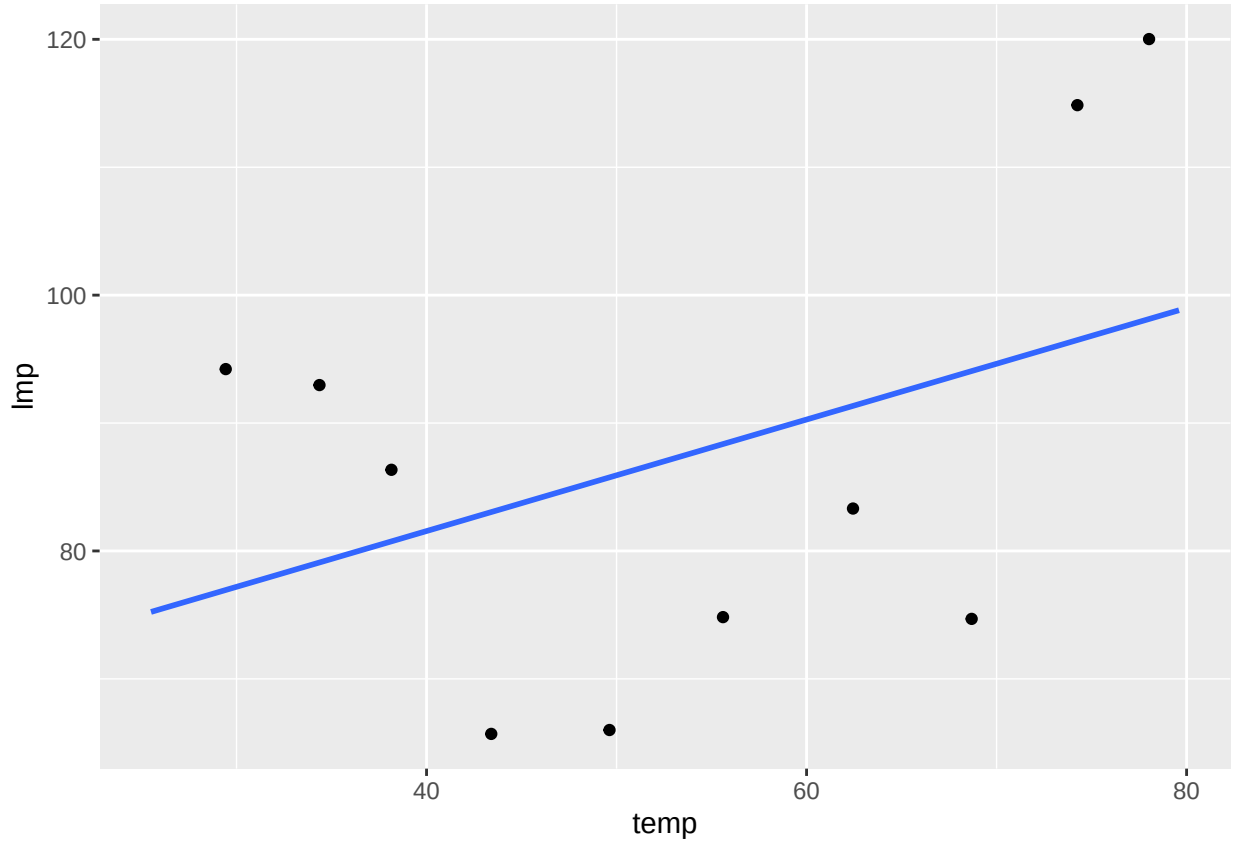
```
##
##     qq
```

```
library(binsreg)
```

```
#Load Bid Data
DataDir = paste(getwd(), "/nodelmp.csv",sep = "")
Data = read.csv(DataDir)
```

Question 1

```
#Obtain the Maximum Price each day and the corresponding temperature
lmp_Maximum_df <- subset(Data, hrank == 1)

#Make a binned scatterplot of the relationship
lmp_max <- lmp_Maximum_df$lmp
temperature <- lmp_Maximum_df$temp

ggplot(lmp_Maximum_df, aes(x=temp,y=lmp)) +
  stat_smooth(method = "lm", formula = y~x, size = 1, se = FALSE) +
   stat_binmean(n = 10)
```



Based on the binned scatterplot, the relationship between the maximum price per day and the temperature does not appear to be linear.

Question 2:

```
rmse <- function(sm)
    sqrt(mean(sm$residuals^2))

fit <- lm( lmp_max~temperature)
deg = 0

for(i in 1:10){
  fit_curr <- lm(lmp_max~poly(temperature,i))
  if(rmse(fit_curr) < rmse(fit)){
    fit <- fit_curr
    deg = i
  }
}

deg
```

```
## [1] 10
```

```
rmse(fit)
```

```
## [1] 53.78921
```

According to the mean squared error, the polynomial with a degree of 10 features the best fit in-sample.

Question 3

```
kmse <- function(x){
  # Define training control
  set.seed(123)
  train.control <- trainControl(method = "cv", number = 10)
# Train the model
  if(x == 1){
  model <- train(lmp_max ~ temperature, data = data.frame(lmp_max, temperature), method = "lm",
                 trControl = train.control)
  }
  else{
    f <- bquote(lmp_max ~ poly(temperature, .(x)))
  model <- train(as.formula(f), data = data.frame(lmp_max, temperature), method = "lm",
                 trControl = train.control)
  }

  # Summarize the results
  return(model)
}


model <- kmse(1)
deg = 1
for(i in 2:10){
  model_curr <- kmse(i)
  if(model_curr$results$RMSE < model$results$RMSE){
    model <- model_curr
    deg = i
```

```
  }
}
deg
```

```
## [1] 7
```

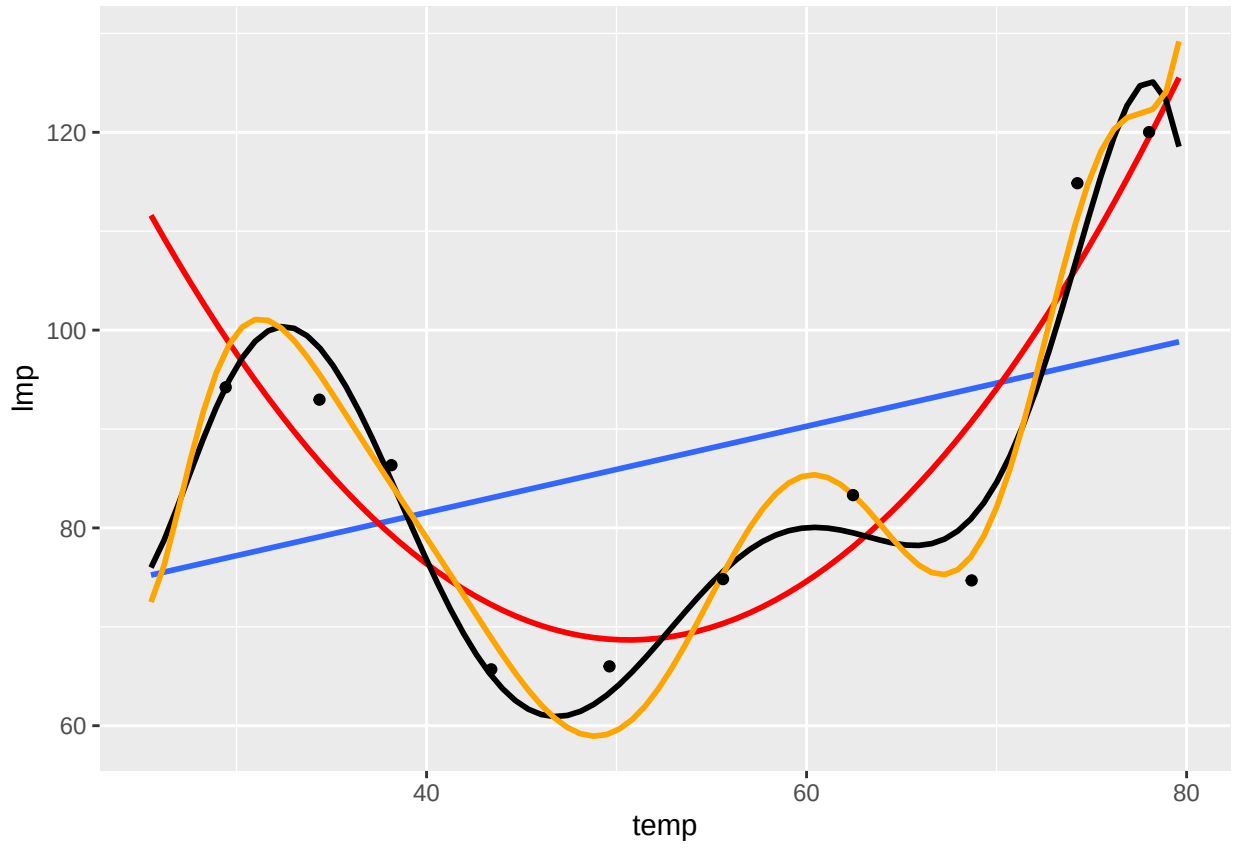
```
model
```

```
## Linear Regression
##
## 362 samples
##   1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 326, 325, 326, 326, 326, 325, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   52.63948  0.125066  35.65637
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

According to the 10-fold cross-validation, the polynomial with the best fit is of degree 7.

Question 4

```
ggplot(lmp_Maximum_df, aes(x=temp,y=lmp)) +
   stat_smooth(method = "lm", formula = y~x, size = 1, se = FALSE) +
  stat_smooth(method = "lm", formula = y~poly(x,2), size = 1, se = FALSE, color = "red") +
  stat_smooth(method = "lm", formula = y~poly(x,7), size = 1, se = FALSE, color = "black") +
 stat_smooth(method = "lm", formula = y~poly(x,10), size = 1, se = FALSE, color = "orange") +
  stat_binmean(n = 10)
```



Question 5

```
lmp_Maximum_df <- lmp_Maximum_df[order(lmp_Maximum_df$temp),]

ncsmse <- function(x){
  # Define training control
  set.seed(123)
  train.control <- trainControl(method = "cv", number = 10)
# Train the model
  f <- bquote(lmp_max ~ ns(temperature, .(df = x + 1)))
  model <- train(as.formula(f), data = data.frame(lmp_max, temperature), method = "lm",
                 trControl = train.control)

  # Return the results
  return(model)
}

modelspline <- ncsmse(1)
knots = 1
for(i in 2:10){
  model_curr <- ncsmse(i)
  if(model_curr$results$RMSE < model$results$RMSE){
    modelspline <- model_curr
    knots = i
  }
```

```
}
knots
```

```
## [1] 4
```

```
modelspline
```

```
## Linear Regression
##
## 362 samples
##   1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 326, 325, 326, 326, 326, 325, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   52.56075  0.1281762  35.56501
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

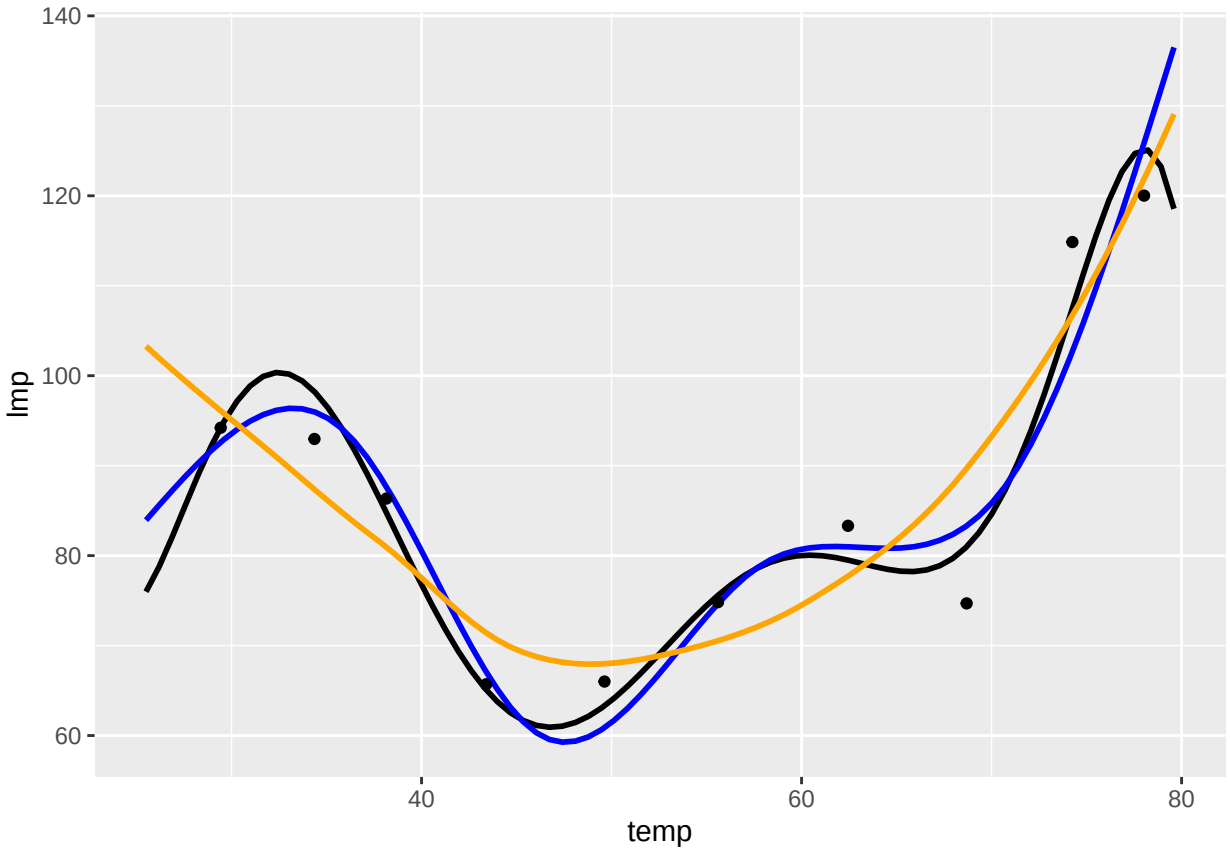
The Optimal number of knots here is 4 knots.

Question 6

```
fm1 <- loess(lmp_max ~ temperature)
fm1
```

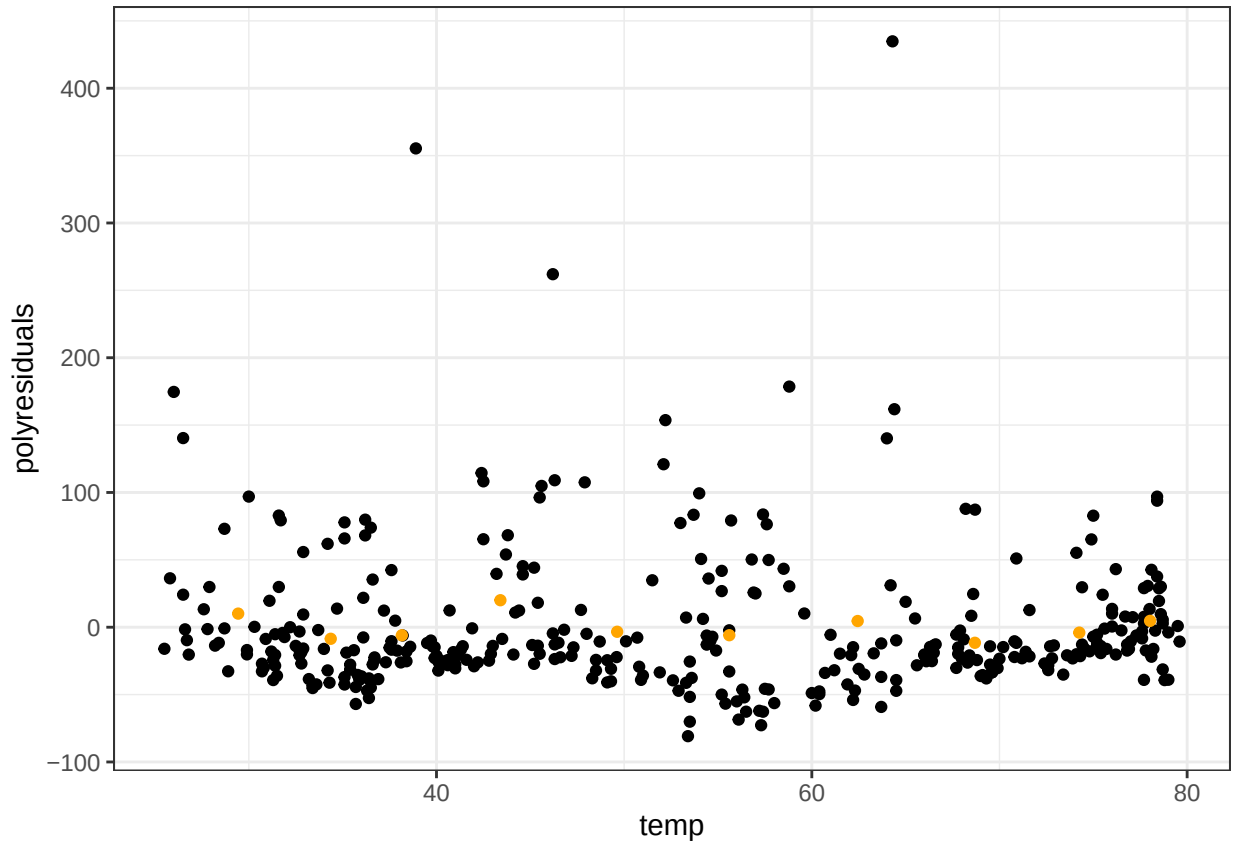
```
## Call:
## loess(formula = lmp_max ~ temperature)
##
## Number of Observations: 362
## Equivalent Number of Parameters: 4.18
## Residual Standard Error: 54.69
```

```
ggplot(lmp_Maximum_df, aes(x=temp,y=lmp)) +
  stat_smooth(method = "lm", formula = y~poly(x,7), size = 1, se = FALSE, color = "black") +
  stat_smooth(method = "lm", formula = y~ns(x, df = 4+1), size = 1, se = FALSE, color = "blue") +
 stat_smooth(method = "loess", formula = y~x, size = 1, se = FALSE, color = "orange") +
  stat_binmean(n = 10)
```

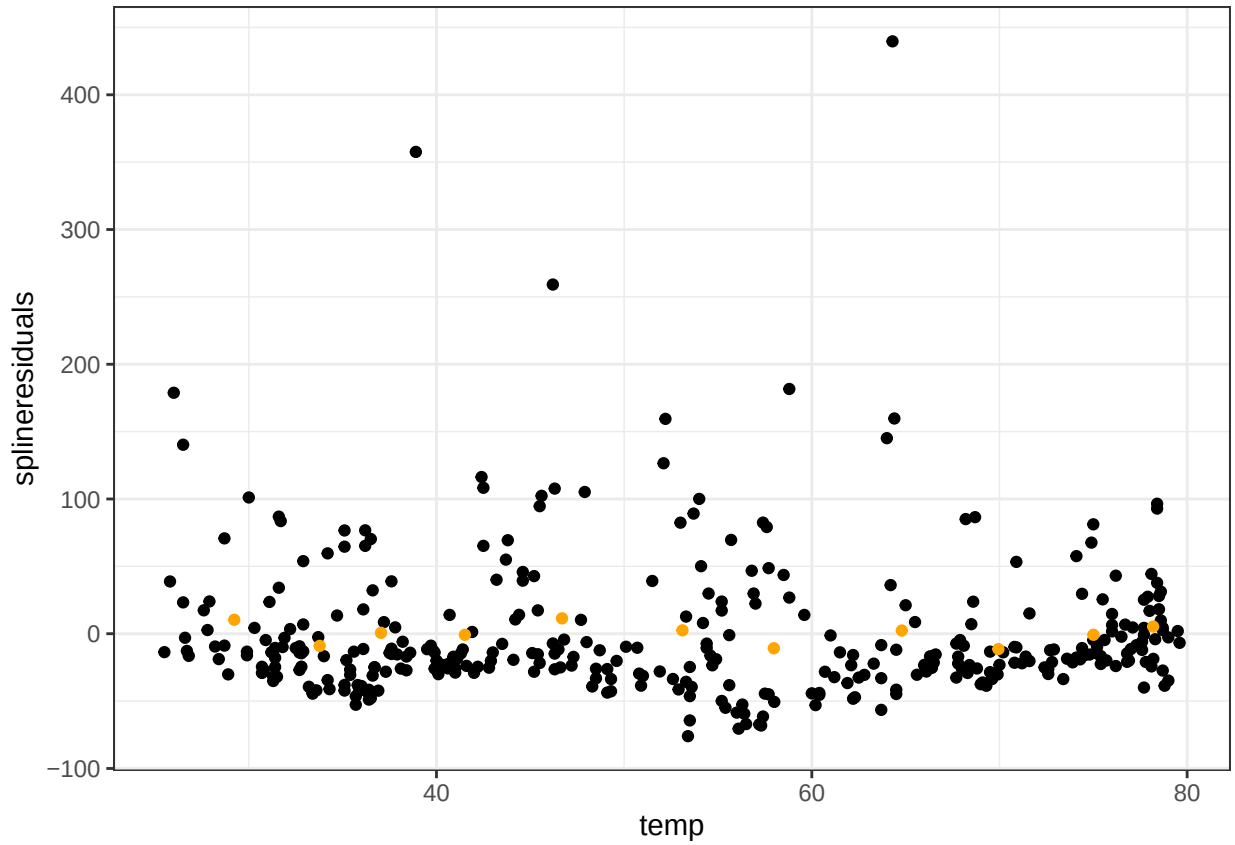


Question 7

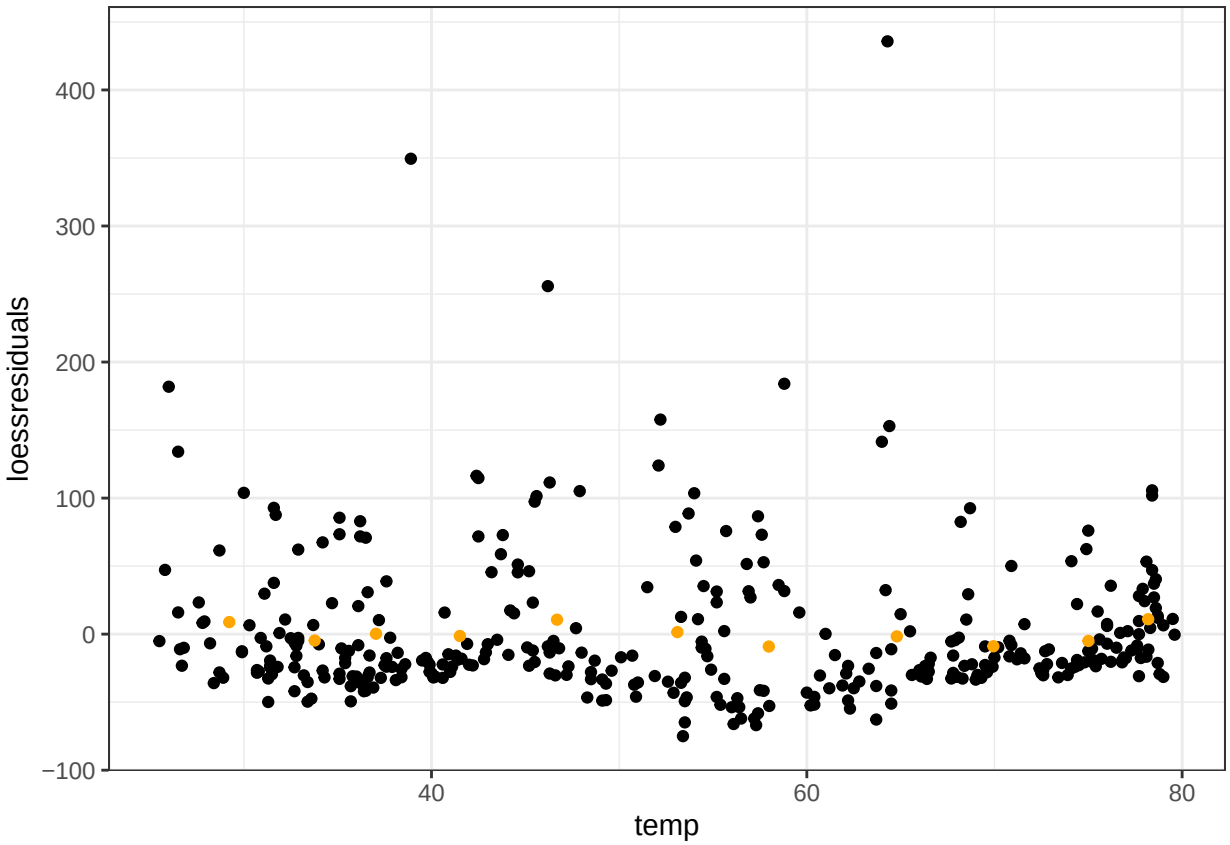
```
loessresiduals <- fm1$residuals

lmp_Maximum_df$polypredicted <- predict(model$finalModel)
lmp_Maximum_df$polyresiduals <- residuals(model$finalModel)

lmp_Maximum_df$splinepredicted <- predict(modelspline$finalModel)
lmp_Maximum_df$splineresiduals <- residuals( modelspline$finalModel)

lmp_Maximum_df$loesspredicted <- fm1$fitted
lmp_Maximum_df$loessresiduals <- fm1$residuals

ggplot(lmp_Maximum_df, aes(x=temp,y=polyresiduals)) +
  geom_point() +
  stat_binmean(n = 10, color = "orange") +
  theme_bw()
```




```
ggplot(lmp_Maximum_df, aes(x=temp,y=splineresiduals)) +
  geom_point() +
  stat_binmean(n = 11, color = "orange") +
  theme_bw()
```




```
ggplot(lmp_Maximum_df, aes(x=temp,y=loessresiduals)) +
  geom_point() +
  stat_binmean(n = 11, color = "orange") +
  theme_bw()
```



For the 7-th degree polynomial, it seems to perform better towards the ends of the distribution where we can see more clustering around zero. Towards the center of the distribution, we see a weaker performance.

For the cubic spline, we can see similar outcomes as the 7th degree polynomial, but the residuals that are positive are smaller in magnitude while the residuals that are negative are larger in magnitude.

The lowess distribution performs similarly as the previous two towards the high end of temperature, but towards the lower end of temperature and the center of the distribution, the residuals are marger in magnitude compared to the previous two models. However, in between these boundaries, the residuals seem to decrease in magnitude with this model compared to the others.