Non-linear Regression

Alex Weirth

2023-02-10

Load Libraries

```
library(tidyverse)
library(mgcv)
library(magrittr)
```

Import Data

```
life_df <- read.csv("life_df.csv")
```

Step 1: is there a non linear relationship between life_exp_yrs and any feature variable?

All developed countries have differing life expectancies, however the same amount of HIV deaths per/1000 people. I am going to focus on the relationship of life expectancy ~ hiv_aids for developing countries.

Filtering for developing countries

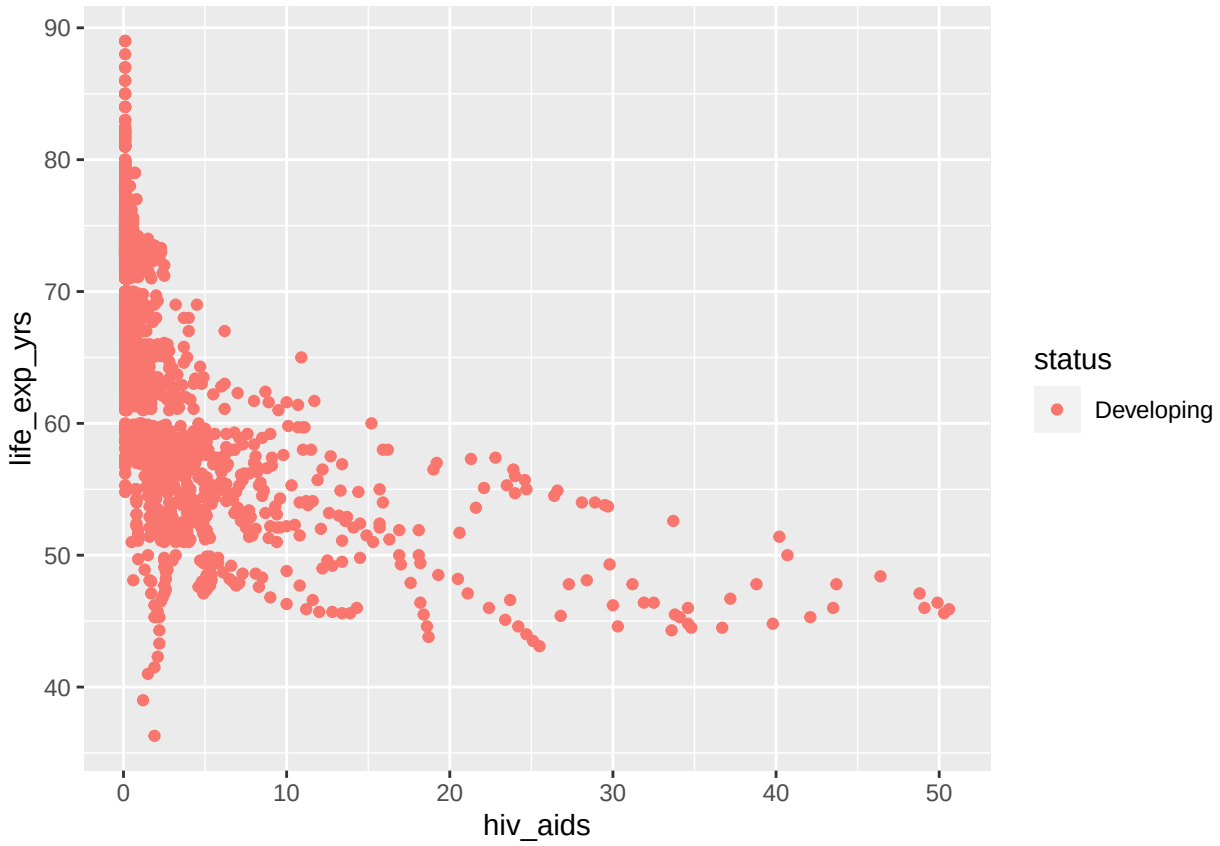
```
life_df_dvling <- life_df %>%
  filter(!is.na(hiv_aids) & status == 'Developing')
```

Testing & Training Data

```
#dim(life_df_dvling) # 1407

set.seed(123)

#70-30 Split
trainInd<-sample(1:1407, 985)

life_df_dvling_train<-life_df_dvling[trainInd, ]
life_df_dvling_test<-life_df_dvling[-trainInd, ]
```

Goal: attempt to fit a nonlinear model for life expectancy ~ hiv_aids for developing countries.

```
library(tidyverse)
ggplot(data = life_df_dvling, aes(x = hiv_aids,  y = life_exp_yrs, color = status))+
  geom_point()
```



Step 2 - Fitting non-linear models

Polynomial Model

```
poly_mod <- lm(life_exp_yrs ~ poly(hiv_aids, 10), data = life_df_dvling_train)
summary(poly_mod)
```

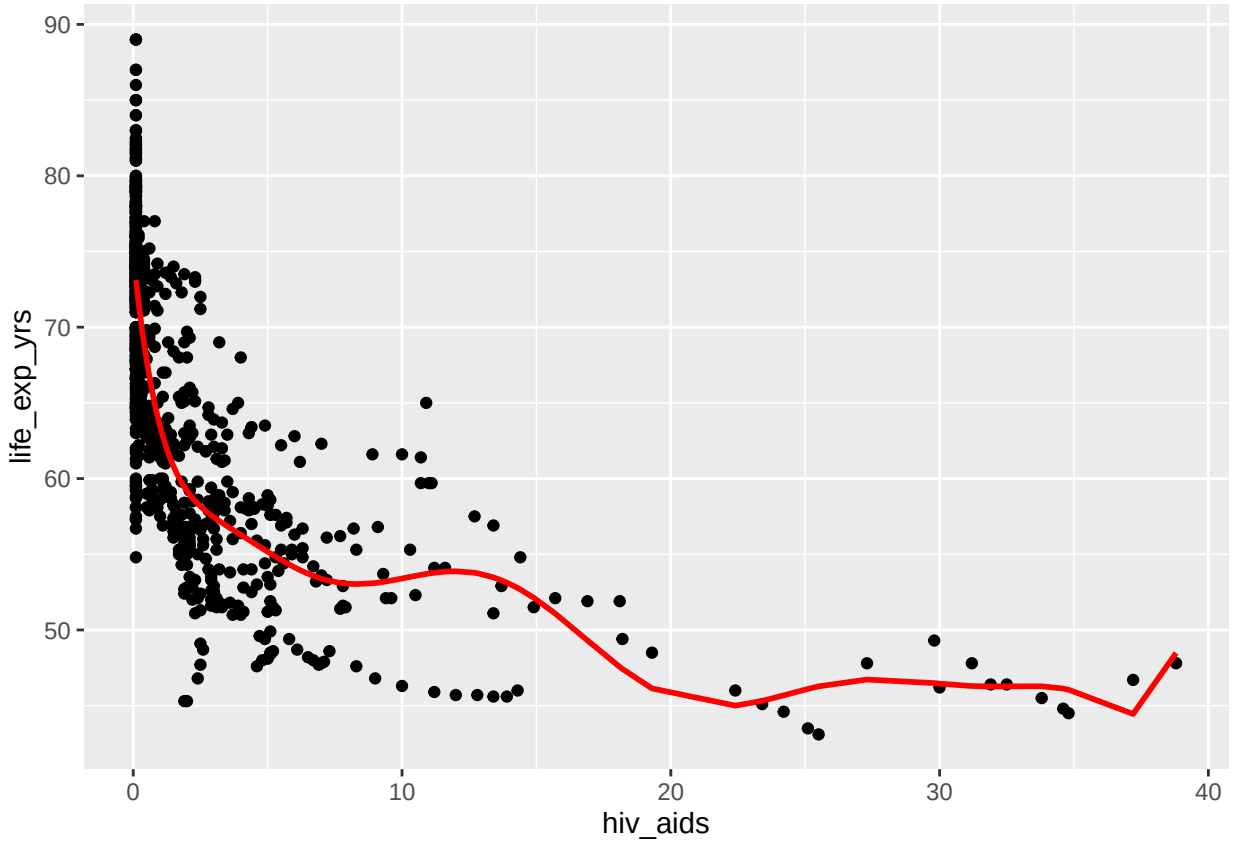
```
##
## Call:
## lm(formula = life_exp_yrs ~ poly(hiv_aids, 10), data = life_df_dvling_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.3113  -3.5113   0.2542   3.0750  15.8887
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              67.2552     0.1683 399.675  < 2e-16 ***
## poly(hiv_aids, 10)1    -176.9445     5.2744 -33.548  < 2e-16 ***
## poly(hiv_aids, 10)2     123.3254     5.2745  23.381  < 2e-16 ***
## poly(hiv_aids, 10)3     -74.7194     5.2740 -14.168  < 2e-16 ***
## poly(hiv_aids, 10)4      61.1495     5.2738  11.595  < 2e-16 ***
## poly(hiv_aids, 10)5     -40.5053     5.2736  -7.681 3.85e-14 ***
## poly(hiv_aids, 10)6      24.4530     5.2736   4.637 4.02e-06 ***
```

```
## poly(hiv_aids, 10)7   -10.4561     5.2735  -1.983  0.04767 *
## poly(hiv_aids, 10)8    14.0543     5.2735   2.665  0.00782 **
## poly(hiv_aids, 10)9   -12.3210     5.2735  -2.336  0.01967 *
## poly(hiv_aids, 10)10    8.4398     5.2734   1.600  0.10983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.273 on 971 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.6842, Adjusted R-squared:  0.681
## F-statistic: 210.4 on 10 and 971 DF,  p-value: < 2.2e-16
```

Significance drops off after the 9th Degree.

Graphic of first polynomial model

```
# Since NA observations were removed in the model I had to remove them
# in the dataset as well so fitted values could match
life_df_dvling_train %>%
  filter(!is.na(life_exp_yrs)) %>%
    ggplot(aes(x = hiv_aids, y = life_exp_yrs))+
      geom_point()+
      geom_line(aes(y=poly_mod$fitted.values), color="red", size=1)
```



To better determine the degree to use for our polynomial model I must use a K-folds cross validation grid search.

K-Folds Cross Validation & Grid Search

```
dim(life_df_dvling)
```

```
## [1] 2426   24
```

1407 observations. This means 5 folds will divide it into 281, 281, 281, 282, 282

```
set.seed(7)
life_trainInd<-sample(1:1407, 985)
life_trainDat<-life_df_dvling[trainInd, ]
life_testDat<-life_df_dvling[-trainInd, ]
```

```
### HOLD MANY FOLDS
kf<-5

### RANDOM SPLIT INTO K FOLDS
### RANDOM INDEXES
life_ind<-sample(1:1407)

### CREATE DF
life_folds<-data.frame(life_ind,
                  fold=c(rep(1, 281), rep(2, 281),
                  rep(3, 281), rep(4, 282), rep(5, 282)))

### ADD ON COLUMNS TO ORIGINAL DAT
life_foldPoly<-life_df_dvling[life_ind,]%>%
  cbind(life_folds)

### INITIALIZE RMSE DATAFRAME TO HOLD OUTPUT
RMSE <- data.frame('fold' = NA,
                   'kth.order' = NA,
                   'RMSE' = NA,
                   'TestRMSE'=NA)


### LOOP FOR CROSS-VALIDATION
for(i in 1:kf){
  life_trainDat<-life_foldPoly%>%
    filter(fold!=i)

  life_testDat<-life_foldPoly%>%
    filter(fold==i)

  ### INNER LOOP FOR POLY DEGREE
  k <- 1:10 #k-th order

  for (j in 1:length(k)){
    row<-length(k)*(i-1)+j

    # build models
    poly_model_2 <- lm(life_exp_yrs ~ poly(hiv_aids,k[j]), data = life_trainDat)

    # calculate RSME and store it for further usage
    RMSE[row,1] <-i
```

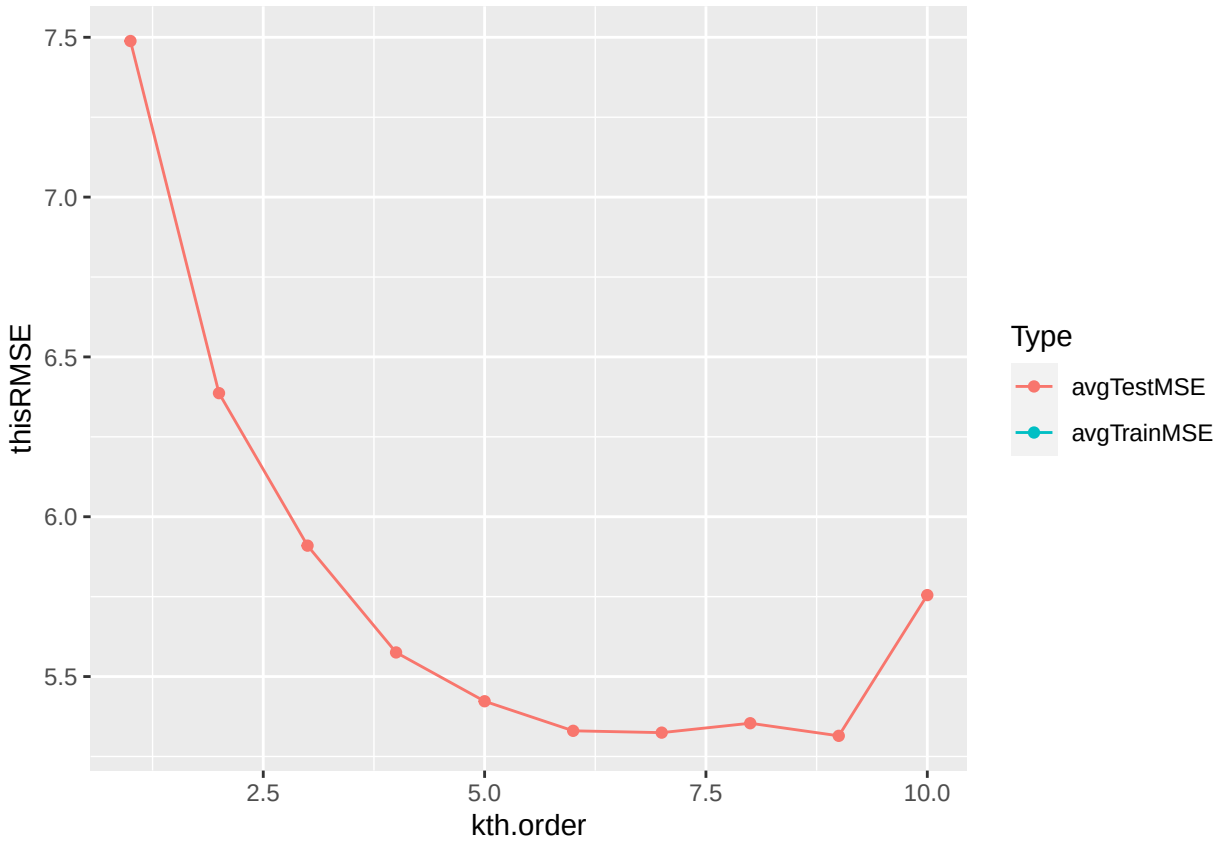
```
    RMSE[row,2] <- k[j] # store k-th order
    RMSE[row,3] <- sqrt(sum((fitted(poly_model_2)-life_trainDat$life_exp_yrs)^2)/
                          length(life_trainDat$life_exp_yrs)) # calculate RSME

    life_predTest<-predict(poly_model_2, life_testDat)

    RMSE[row, 4]<-sqrt(sum((life_predTest-life_testDat$life_exp_yrs)^2,  na.rm=TRUE)/
                         length(life_predTest-sum(is.na(life_predTest)))) # calculate RSME

  }
}
```

Aggregate the folds

```
### AGGREGATE CROSS_VALIDATION
cvRMSE<-RMSE%>%
  group_by(kth.order)%>%
  summarise(avgTrainMSE=mean(RMSE),
            avgTestMSE=mean(TestRMSE),
            sdTrain=sd(RMSE),
            sdTest=sd(TestRMSE))

cvRMSE%>%
  gather(key="Type", value=thisRMSE, -c(kth.order))%>%
  filter(Type %in% c("avgTrainMSE", "avgTestMSE"))%>%
  ggplot(aes(x=kth.order, y=thisRMSE, color=Type))+
  geom_line()+
  geom_point()
```

```
## Warning: Removed 10 rows containing missing values (`geom_line()`).
```

```
## Warning: Removed 10 rows containing missing values (`geom_point()`).
```




```
### WHICH MINIMIZES
which.min(cvRMSE$avgTestMSE)
```

```
## [1] 9
```

```
cvRMSE$avgTestMSE[which.min(cvRMSE$avgTestMSE)]
```

```
## [1] 5.314564
```

The grid search is showing what the summary of the original model is telling me; at about the 9th Degree the model begins to overfit and the RMSE is increasing. After using the 'which.min()' function R shows that the 9th degree minimizes the RMSE so that is what I will use for the final model as it concurs with the original model summary. It is important to note that the polynomial model I used is not a great model overall for our data as its tendency is to over fit.

Final Polynomial Model

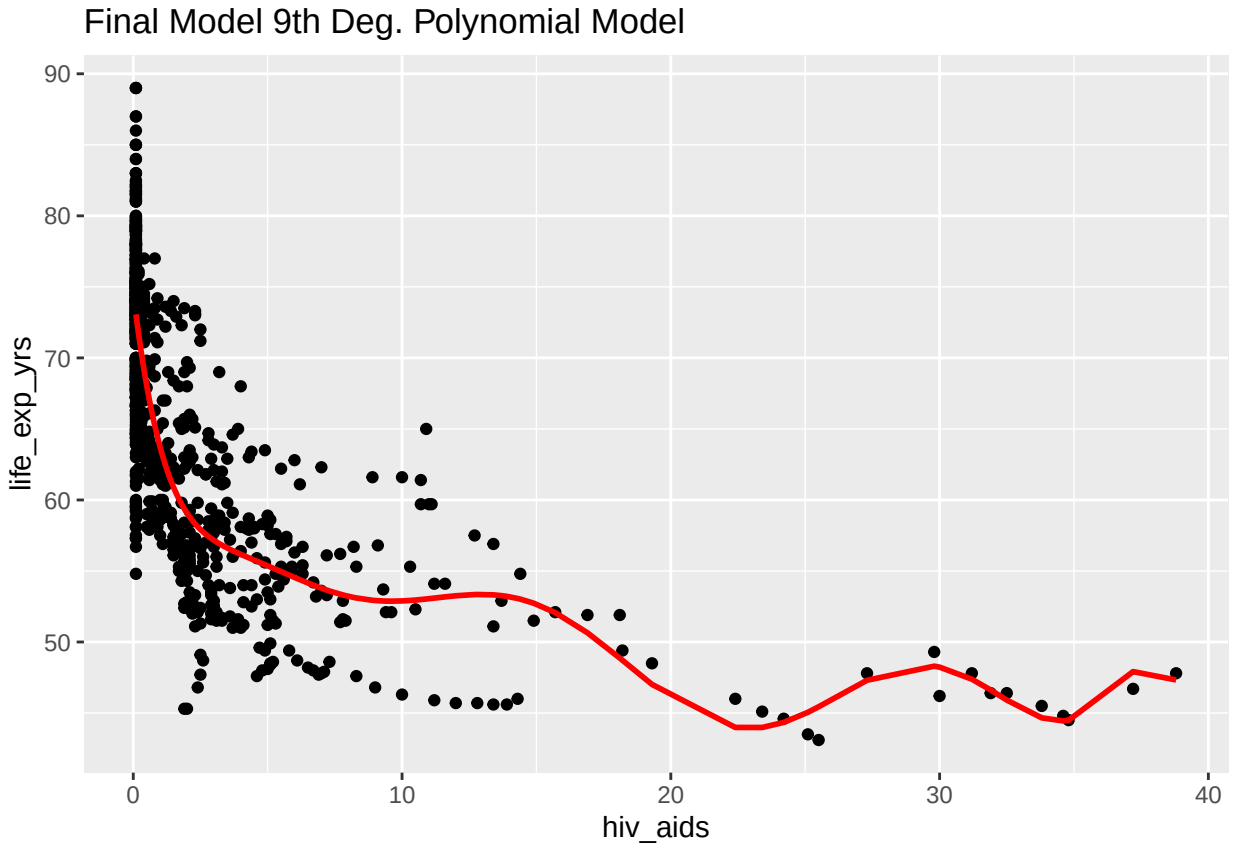
```
poly_mod_final <- lm(life_exp_yrs ~ poly(hiv_aids, 9), data = life_df_dvling_train)
summary(poly_mod_final)
```

```
##
## Call:
## lm(formula = life_exp_yrs ~ poly(hiv_aids, 9), data = life_df_dvling_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.2637  -3.6532   0.2482   3.0363  15.9363
##
## Coefficients:
```

```
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           67.2550     0.1684 399.354  < 2e-16 ***
## poly(hiv_aids, 9)1 -176.9427     5.2787 -33.520  < 2e-16 ***
## poly(hiv_aids, 9)2  123.3235     5.2788  23.362  < 2e-16 ***
## poly(hiv_aids, 9)3  -74.7179     5.2782 -14.156  < 2e-16 ***
## poly(hiv_aids, 9)4   61.1483     5.2780  11.585  < 2e-16 ***
## poly(hiv_aids, 9)5  -40.5043     5.2779  -7.674 4.04e-14 ***
## poly(hiv_aids, 9)6   24.4519     5.2778   4.633 4.09e-06 ***
## poly(hiv_aids, 9)7  -10.4552     5.2777  -1.981  0.04787 *
## poly(hiv_aids, 9)8   14.0534     5.2777   2.663  0.00788 **
## poly(hiv_aids, 9)9  -12.3202     5.2777  -2.334  0.01978 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.277 on 972 degrees of freedom
##   (3 observations deleted due to missingness)
## Multiple R-squared:  0.6834, Adjusted R-squared:  0.6805
## F-statistic: 233.1 on 9 and 972 DF,  p-value: < 2.2e-16
```

Graphic

```
# Since NA observations were removed in the model we had to remove them in the dataset as well so fitte
life_df_dvling_train %>%
  filter(!is.na(life_exp_yrs)) %>%
    ggplot(aes(x = hiv_aids, y = life_exp_yrs))+
      geom_point()+
      geom_line(aes(y=poly_mod_final$fitted.values), color="red", size=1)+
      ggtitle("Final Model 9th Deg. Polynomial Model")
```



GAM Model

```
gam_mod <- gam(life_exp_yrs ~ s(hiv_aids), data = life_df_dvling_train, method = "REML")
summary(gam_mod)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## life_exp_yrs ~ s(hiv_aids)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  67.2373     0.1725   389.7   <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(hiv_aids) 7.796  8.585 226.6  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.665   Deviance explained = 66.7%
```

```
## -REML = 3064.3  Scale est. = 29.235    n = 982
```

```
gam_mod2 <- gam(life_exp_yrs ~ s(hiv_aids) + adult_mortality +
                  infant_deaths + perc_expend +X5yr_deaths + hiv_aids +inc_comp_resources +
                  schooling + hiv_deaths_cat,
                  data = life_df_dvling_train, method = "REML")
summary(gam_mod2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## life_exp_yrs ~ s(hiv_aids) + adult_mortality + infant_deaths +
##     perc_expend + X5yr_deaths + hiv_aids + inc_comp_resources +
##     schooling + hiv_deaths_cat
##
## Parametric coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     58.0777245  1.1101305  52.316  < 2e-16 ***
## adult_mortality                 -0.0137434  0.0013071 -10.514  < 2e-16 ***
## infant_deaths                    0.0859194  0.0177699   4.835 1.56e-06 ***
## perc_expend                      0.0010842  0.0001319   8.217 6.93e-16 ***
## X5yr_deaths                     -0.0673215  0.0134711  -4.997 6.94e-07 ***
## hiv_aids                        -0.9162995  0.3076825  -2.978  0.00298 **
## inc_comp_resources               7.4788002  0.8458298   8.842  < 2e-16 ***
## schooling                        0.6862022  0.0632925  10.842  < 2e-16 ***
## hiv_deaths_catUnder 1 Death/1000  1.7779263  0.6968629   2.551  0.01089 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df     F p-value
## s(hiv_aids) 3.762  4.846 12.91  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 17/18
## R-sq.(adj) =  0.842   Deviance explained = 84.4%
## -REML = 2585.5  Scale est. = 13.31     n = 943
```

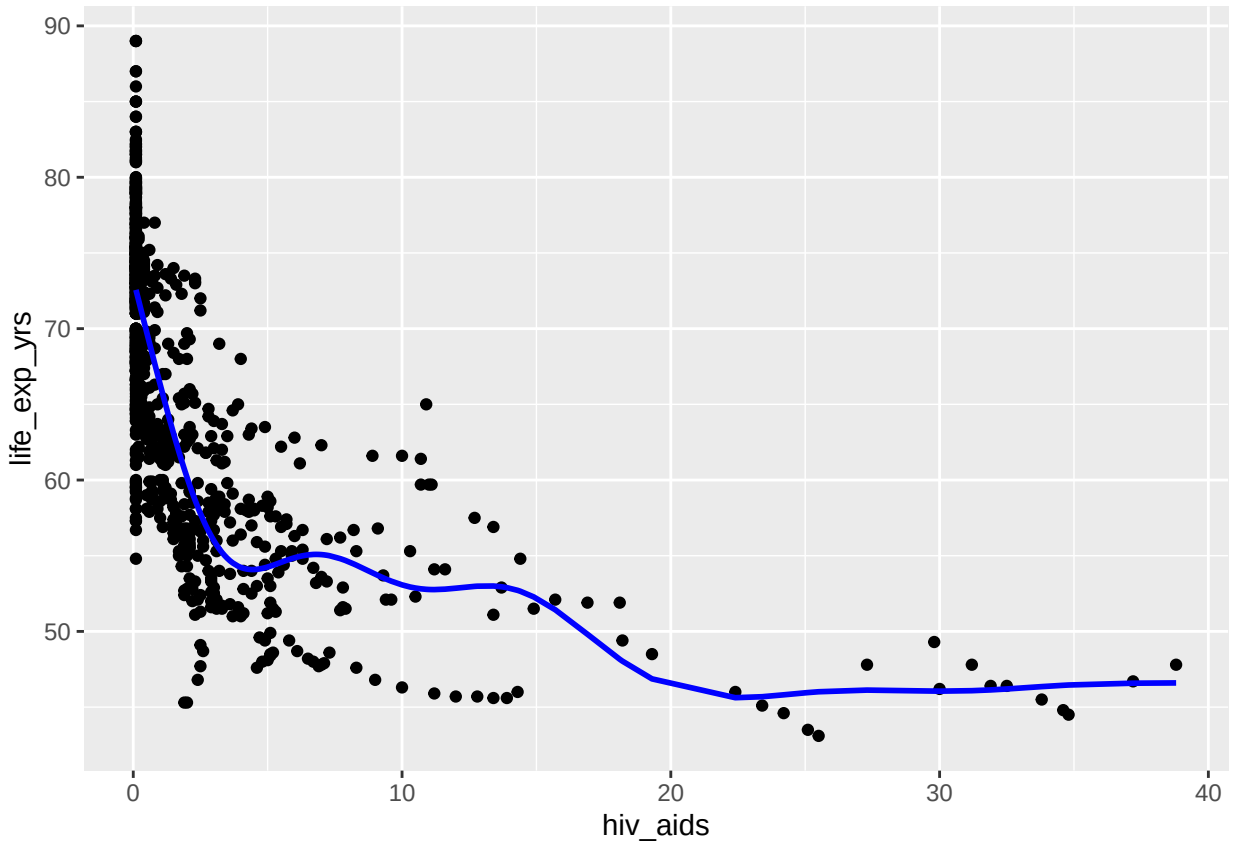
```
sqrt(mean(resid(gam_mod2)^2))
```

```
## [1] 3.623558
```

GAM achieved 0.84 r squared value and 3.63 RMSE.

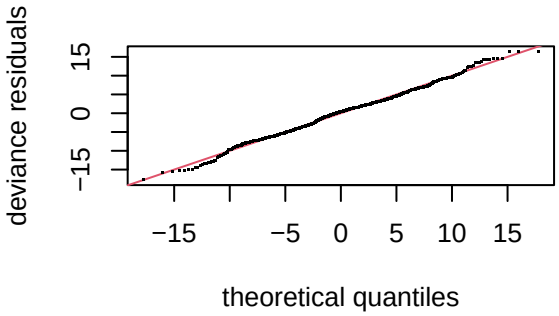
Gam Graphic

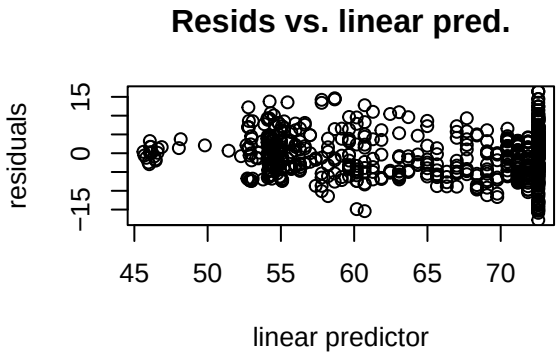
```
life_df_dvling_train %>%
  filter(!is.na(life_exp_yrs)) %>%
    ggplot(aes(x = hiv_aids, y = life_exp_yrs))+
      geom_point()+
      geom_line(aes(y=gam_mod$fitted.values), color="blue", size=1)
```

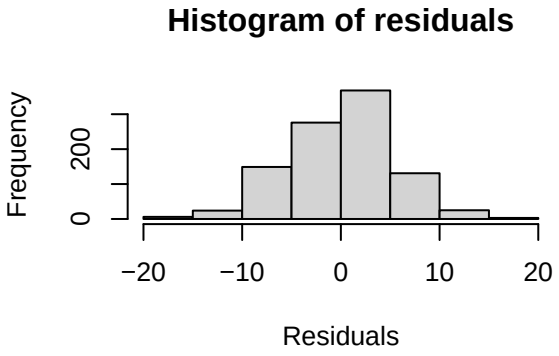


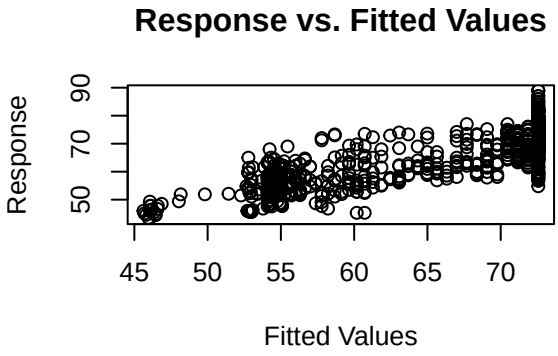
Checking the GAM model with gam.check()

```
gam.check(gam_mod)
```










```
##
## Method: REML   Optimizer: outer newton
## full convergence after 6 iterations.
## Gradient range [-0.0005240586,0.0004195202]
## (score 3064.272 & scale 29.23481).
## Hessian positive definite, eigenvalue range [3.271028,490.0243].
## Model rank =  10 / 10
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##              k' edf k-index p-value
## s(hiv_aids) 9.0 7.8    0.96   0.085 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Takeaways from gam.check():

Good:

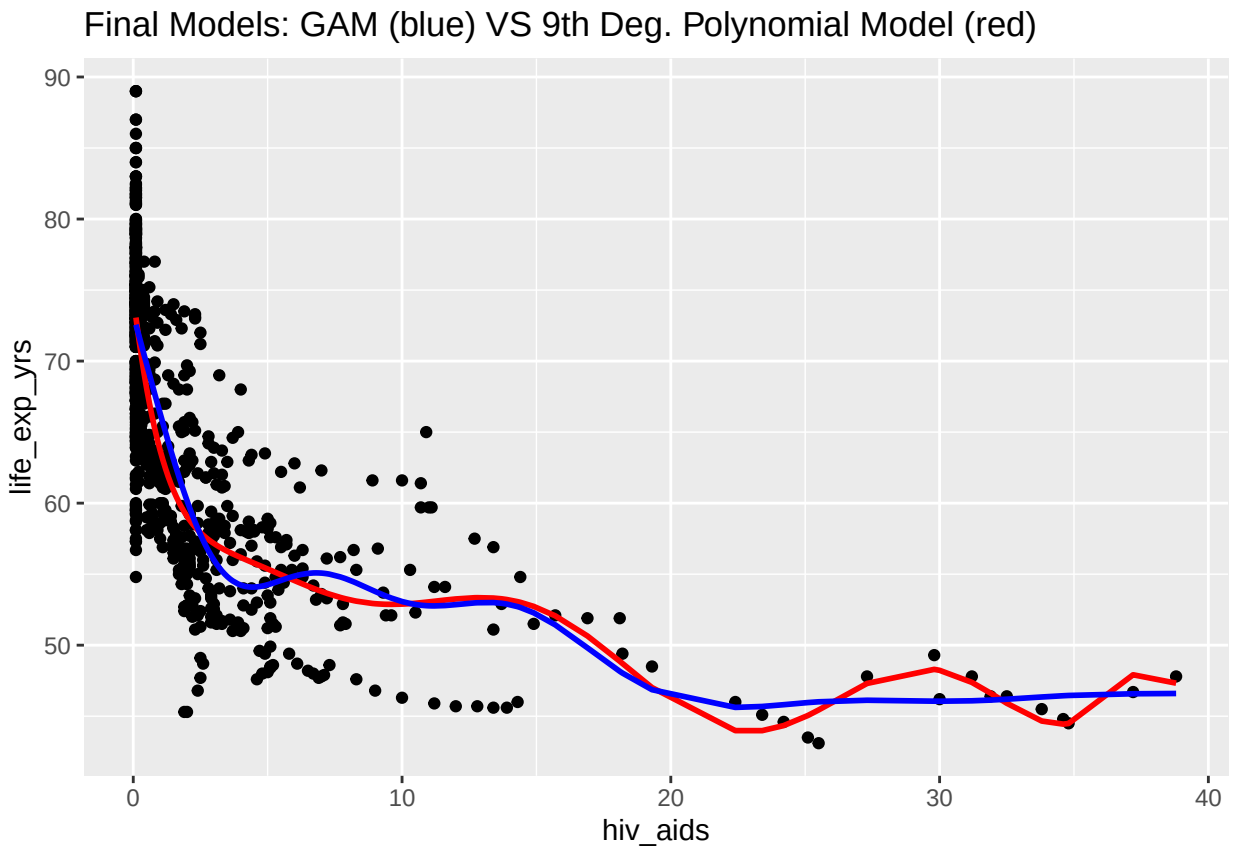
- Residuals vs. linear predictions are centered around 0
- Histogram of residuals appears relatively bell shaped and is not skewed.
- Achieved full convergence

Bad:

- QQ plot is not a straight line
- Response vs. fitted values is not a great line on the x=y line.

Final Graphics

```
# Since NA observations were removed in the model we had to remove them in the dataset as well so fitte
life_df_dvling_train %>%
  filter(!is.na(life_exp_yrs)) %>%
    ggplot(aes(x = hiv_aids, y = life_exp_yrs))+
      geom_point()+
      geom_line(aes(y=poly_mod_final$fitted.values), color="red", size=1)+
      geom_line(aes(y=gam_mod$fitted.values), color="blue", size=1)+
      ggtitle("Final Models: GAM (blue) VS 9th Deg. Polynomial Model (red)")
```



Conclusions

I was able to perform a grid search to find the best hyperparameter for the polynomial which was the 9th degree, however the polynomial model was not best for the data as it tended to prefer overfitting and you can tell by the final model that it was much more sensitive to noise compared to the GAM model, especially when the hiv_aids variable increases. On the contrary, the GAM model seemed to do better and be less sensitive to noise on that end but fit more of the data for lower hiv_aids values.