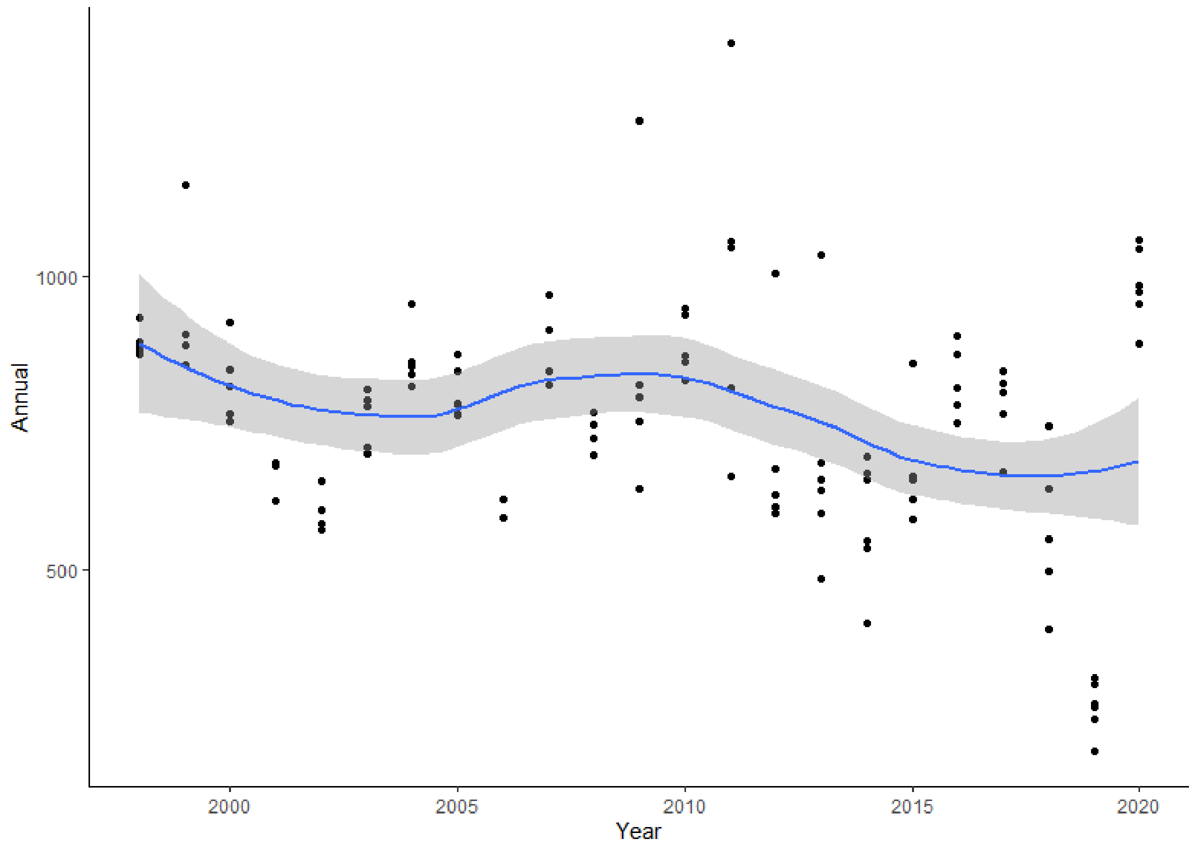

Figure 1: Annual vs Year

The above scatter plot suggests a non-linear relationship between the two variables

Figure 2 shows a plot of adjusted R^2 value gotten from different polynomial degree, and it would be seen that the on lower degrees, the R^2 value gotten indicates that the model barely explains any variability in the data. And as the degrees increases, the R^2 value increases and the optimal model is gotten when the degrees of polynomial that explains a more variability of the model is **"9".** Higher polynomial does not bring any significant change to the model; therefore, our optimal model is created with polynomial degree of 9.

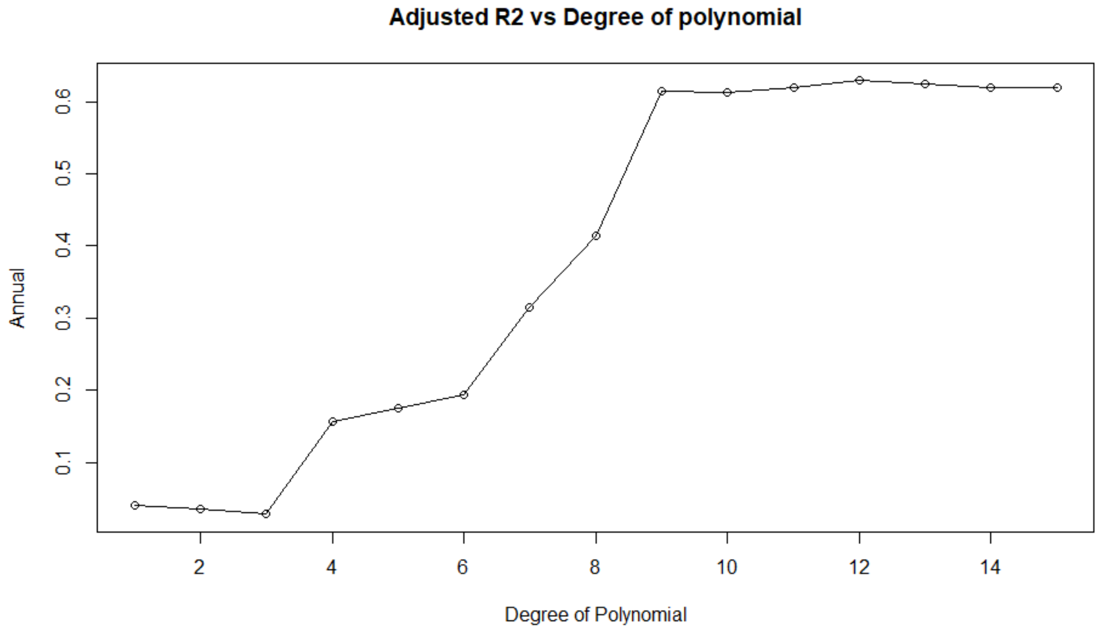

Figure 2: Adjusted R^2 vs Degree of polynomial

```
Call:
lm(formula = Annual ~ poly(Year, 9), data = rain)
Residuals:
    Min      1Q  Median      3Q     Max
-237.34  -76.41   -4.46   53.00  524.66
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      752.52      11.57  65.031  < 2e-16 ***
poly(Year, 9)1  -577.88     125.17  -4.617 1.09e-05 ***
poly(Year, 9)2  -123.16     125.17  -0.984   0.3274
poly(Year, 9)3   -64.25     125.17  -0.513   0.6088
poly(Year, 9)4   737.49     125.17   5.892 4.48e-08 ***
poly(Year, 9)5   324.45     125.17   2.592   0.0109 *
poly(Year, 9)6   254.10     125.17   2.030   0.0448 *
poly(Year, 9)7   680.73     125.17   5.439 3.42e-07 ***
poly(Year, 9)8   626.86     125.17   5.008 2.18e-06 ***
poly(Year, 9)9   981.45     125.17   7.841 3.55e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 125.2 on 107 degrees of freedom
Multiple R-squared:  0.6326,	Adjusted R-squared:  0.6017
F-statistic: 20.47 on 9 and 107 DF,  p-value: < 2.2e-16
```

From the above regression output, it will be seen that the model with the 9th polynomial degree explains about 61.53% of the variability in the data (Adj R^2). And only the regression coefficient of poly(Year, 9)3 and poly(Year, 9)4 appear to be insignificant to the model as their p-value is **> 0.05** which is enough to reject the null hypothesis. And other terms in the model are significant as their p-value is **< 0.05**

Null Hypothesis H_0: $\beta_i = 0$

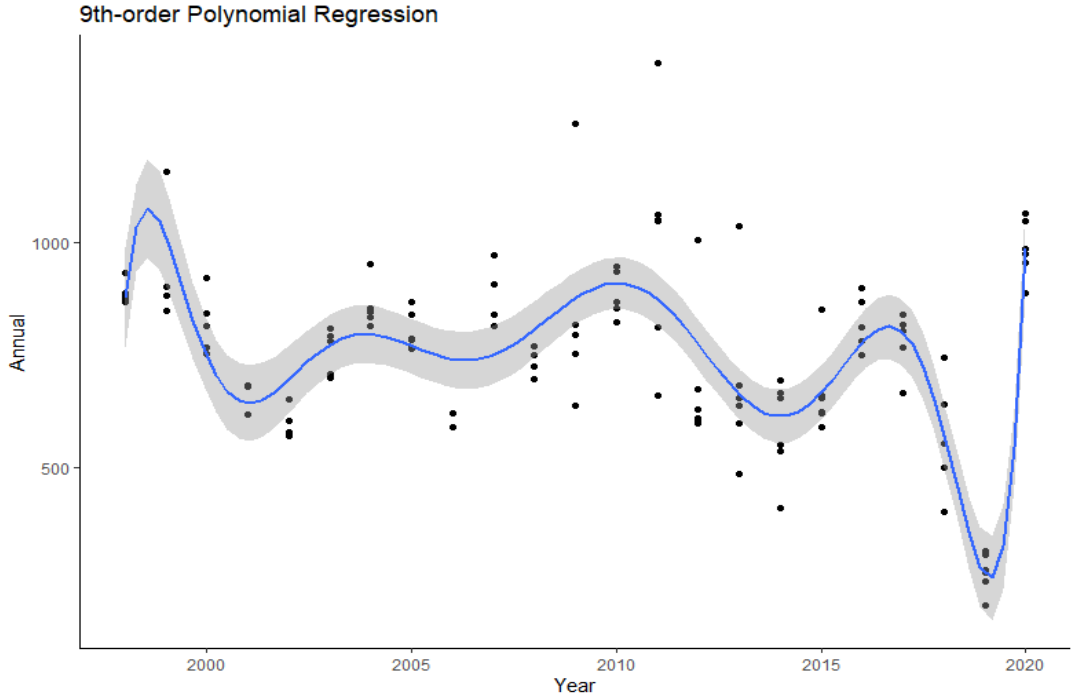

Figure 3: 9th-order Polynomial Regression of training data

Kernel Smoothing

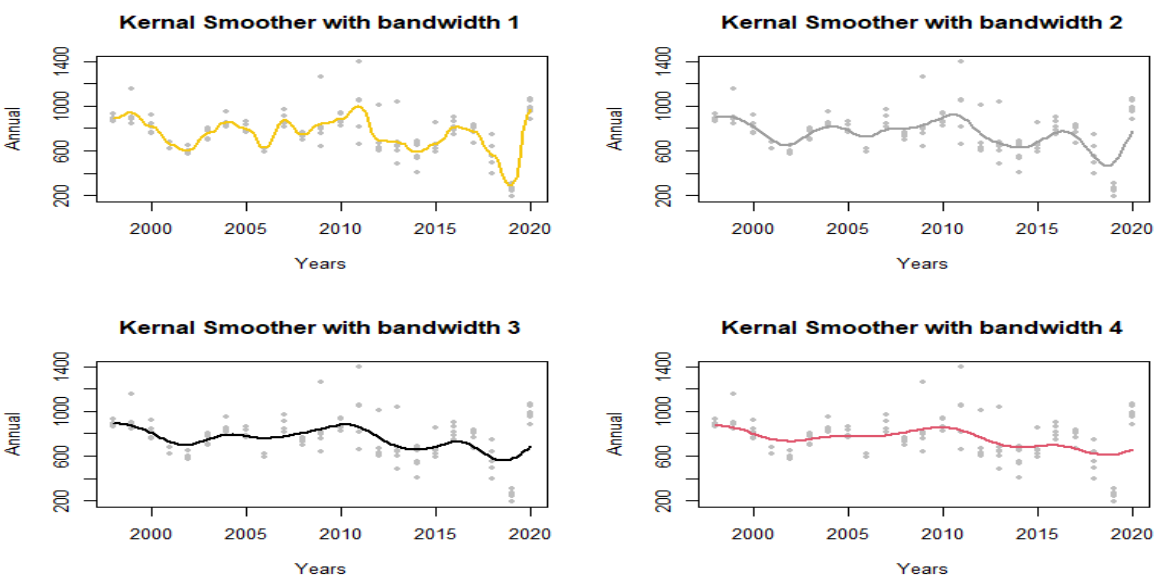

Figure 4: Kernel Smoothing

The kernels are scaled so that their quartiles (viewed as probability densities) are at +/- 0.25*bandwidth. Where the bandwidth is specified manually. Lesser the bandwidth, the better fit we will get.

Local Polynomial

Local polynomials fitting each subset of data are usually of the first or second degree, i.e., either locally linear (in the straight-line sense) or locally quadratic. LOESS becomes a weighted moving average when a zero-degree polynomial is used. In theory, higher-degree polynomials would work, but they would produce models that were not in the spirit of LOESS. LOESS is built on the principles that a low-order polynomial may accurately estimate any function in a narrow region and that simple models can be easily fitted to data. High-degree polynomials are numerically unstable and tend to overfit the data in each subset, making accurate computations challenging.

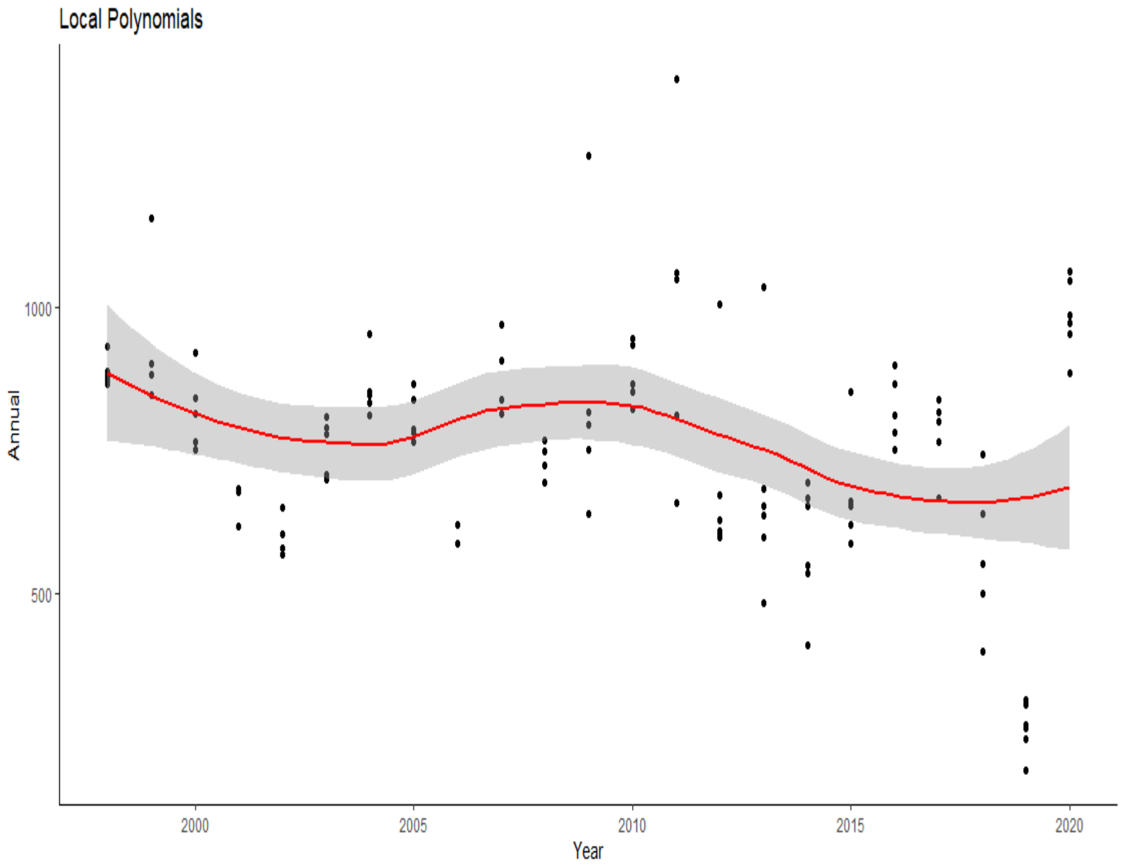

Figure 5: Local Polynomial

<u>Splines</u>

Polynomial regression only captures a definite amount of curvature in a nonlinear association. Another approach to modelling nonlinear relationships is to use splines.

Splines provide a way to smoothly interpolate between fixed points, called knots. Polynomial regression is computed between knots. In other words, splines are series of polynomial segments strung together, joining at knots.

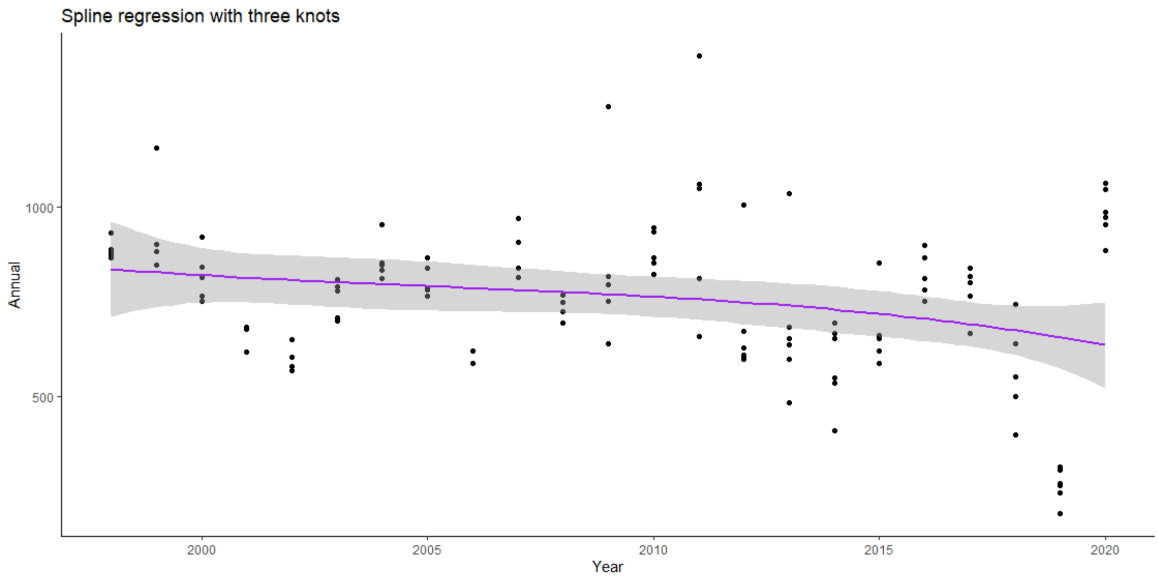

Figure 6: Splines Smoother

General Additive Model

It will be observed that the variables have a non-linear relationship, the polynomial terms may not be flexible enough to capture the relationship, and spline terms require specifying the knots. Generalized additive models, or GAM, are a technique to automatically fit a spline regression without putting the knots manually.

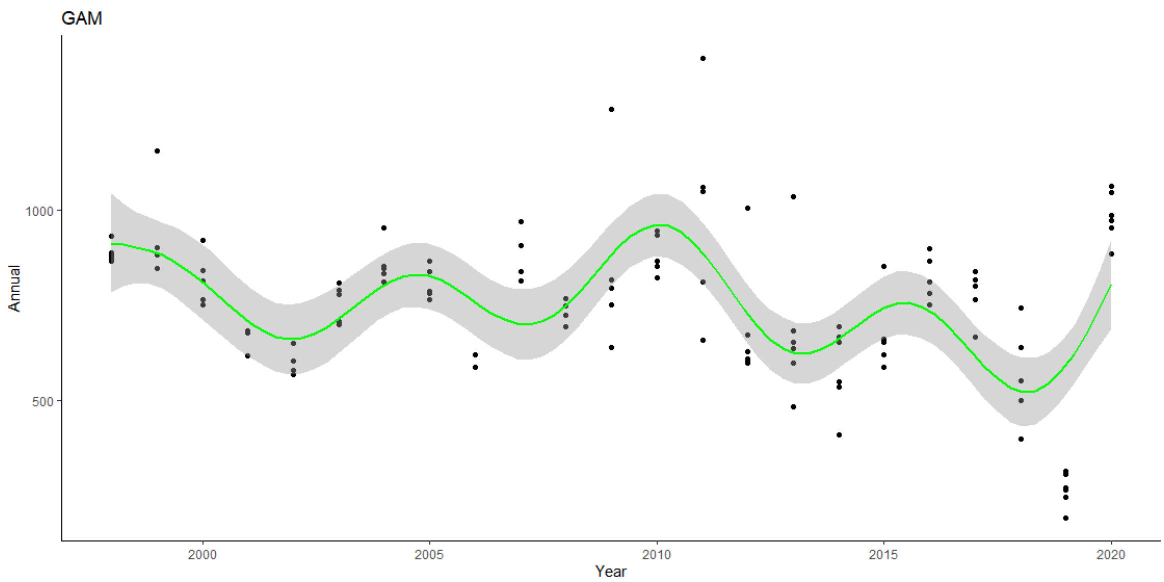

Figure 7: General Additive Model

Colour codes:

Black: 9th order polynomial regression

Green: GAM

Blue: Local polynomial regression

Red: Spline with three knots

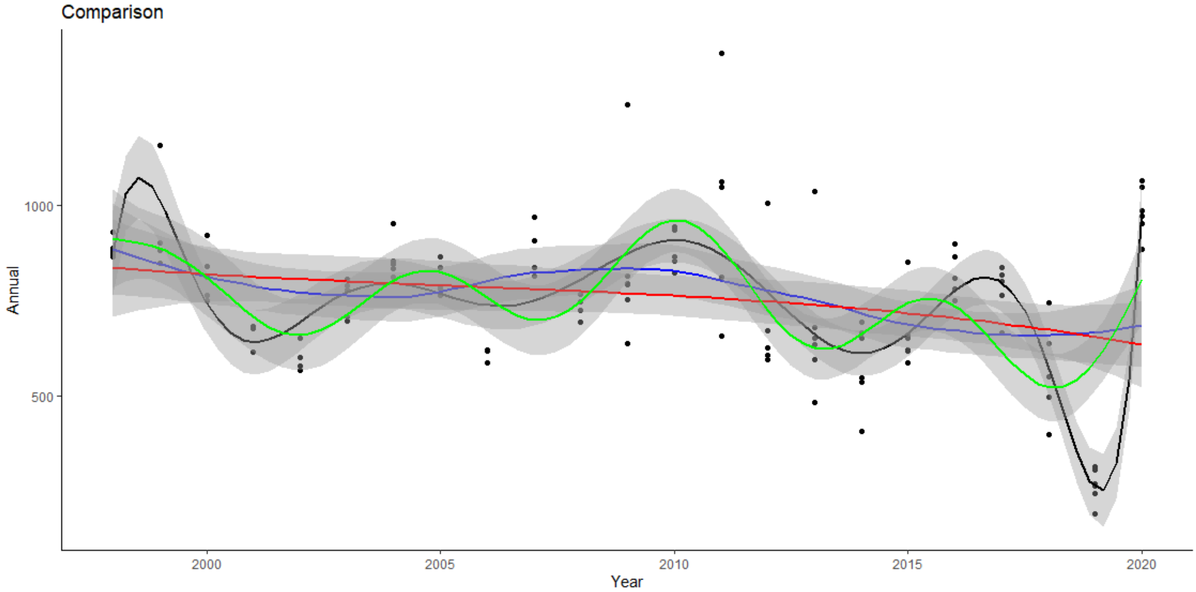

Figure 8: Model Comparison

As we can see, *all the models clearly show the decrease in annual rainfall over time as there is a slope towards the right and this is identifiable with the splines smooth.*

9th order polynomial is a best-fitted model from all of them where the spline with three knots is the most poorly fitted mode. But there is a possibility that the best-fitted model does not predict well on the unseen data, because it might be the case of overfitting. Where GAM is the 2nd model with a good fit. Spline with three knots barely explains any variance in data.

Appendix: R Code

```
rm(list=ls())
#Loading required libraries

library(tidyverse)
library(ggplot2)
library(splines)
theme_set(theme_classic())


#Importing our dataset
setwd("C:/Users/olley/Downloads/Documents")
df <- read.table("ArmidaleRainfall.txt",header=T)


# Visualize Data
ggplot(data=df, aes(Year,Annual)) +
  geom_point() +
  geom_smooth()


# Polynomial regression Modeling
adj_r2 = 1
for (i in 1:15){
  model=lm(Annual~poly(Year,i),data=df)
  adj_r2[i] = summary(model)$adj.r.squared
}
plot(adj_r2~seq.int(1,15),xlab="Degree of
Polynomial",ylab="Annual",main="Adjusted R2 vs Degree of Polynomial")
lines(adj_r2~seq.int(1,15))

# Final polynomial regression model
poly_model=lm(Annual~poly(Year,9),data=df)
summary(poly_model)

par(mfrow=c(2,2))
plot(poly_model)
par(mfrow=c(1,1))


#Visualizing Polynomial Regression
ggplot(data=df, aes(Year,Annual)) +
  geom_point() +
  geom_smooth(method="lm", formula=y~poly(x,9))+ labs(title = "9th-
order Polynomial Regression")


# Kernal smoother
par(mfrow=c(2,2))
for (i in 1:4){
  plot(df$Year,df$Annual,pch=20,col='grey', xlab="Years",ylab =
"Annual",
       main = paste("Kernal Smoother with bandwidth",i))
  lines(ksmooth(df$Year,df$Annual, "normal", bandwidth = i),
        col = 7862+i,lwd = 2)
}
par(mfrow=c(1,1))
```

```
# Local Polynomial
local=loess(Annual ~ Year, df)
summary(local)
# Visualize the smoother
ggplot(data=df, aes(Year,Annual)) +
  geom_point() +
  geom_smooth(method="loess", formula= y ~ x ,col = "red")+
  labs(title = "Local Polynomials")


# Splines Model
knots <- quantile(df$Year, p = c(0.25,0.5,0.75))
model <- lm (Annual ~ bs(Year, knots = knots), data = df)
summary(model)
# Visualization
ggplot(df, aes(Year, Annual) ) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ splines::bs(x, 3),col =
"purple")+
  labs(title = "Spline regression with three knots")



# GAM
ggplot(df, aes(x = Year, y = Annual)) + geom_point()+
  stat_smooth(method = "gam",formula = y ~s(x), size = 1, se = T,
colour = "green")+
  labs(title = "GAM")



# Comparison on a same plot
m <- ggplot(df, aes(x = Year, y = Annual)) + geom_point()
print(m)
m + stat_smooth(method = "lm", formula = y~poly(x,9), size = 1, se = T,
                colour = "black") + stat_smooth(method = "loess",
formula = y ~ x,
                                                size = 1, se = T,
colour = "blue") + stat_smooth(method = "lm",

formula =  y ~ splines::bs(x, df = 3), size = 1, se = T, colour =
"red") + stat_smooth(method = "gam",

formula = y ~s(x), size = 1, se = T, colour = "green") +labs(title =
"Comparison")
```