Non Linear modelling (22IM10040)

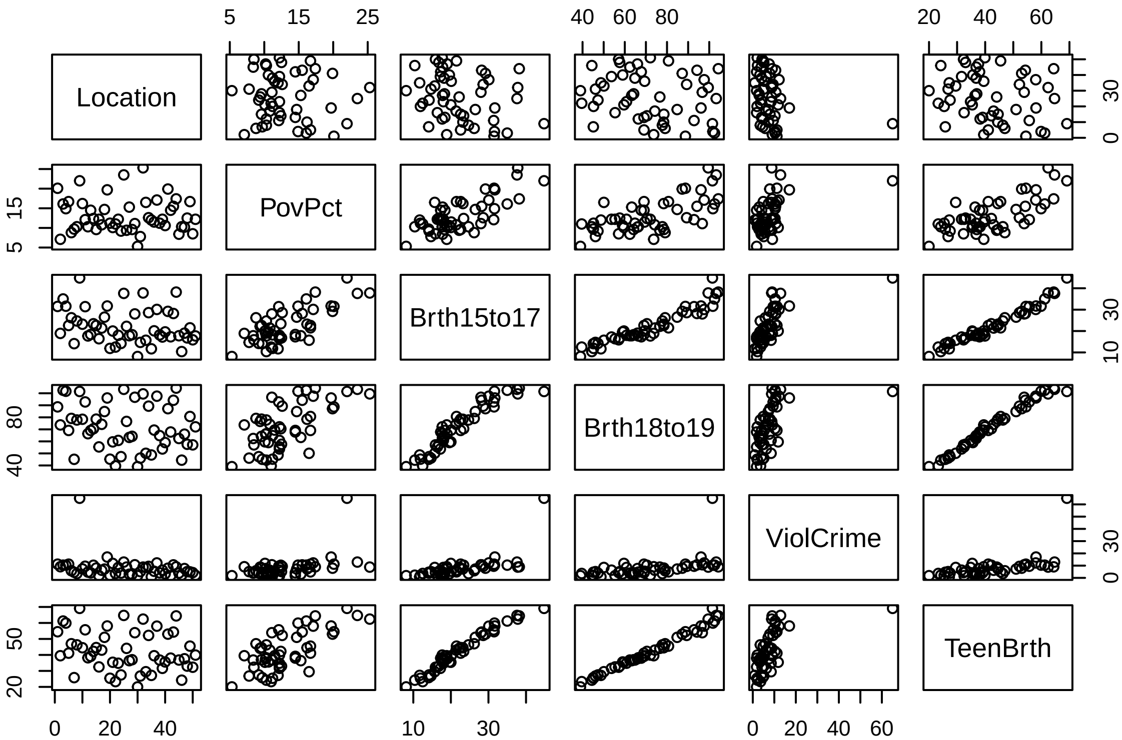
Sunny Kumar

2025-03-19

```r
head(df)
```

```
##     Location PovPct Brth15to17 Brth18to19 ViolCrime TeenBrth
## 1    Alabama   20.1       31.5       88.7      11.2     54.5
## 2     Alaska    7.1       18.9       73.7       9.1     39.5
## 3    Arizona   16.1       35.0      102.5      10.4     61.2
## 4   Arkansas   14.9       31.6      101.7      10.4     59.9
## 5 California   16.7       22.6       69.1      11.2     41.1
## 6   Colorado    8.8       26.2       79.1       5.8     47.0
```

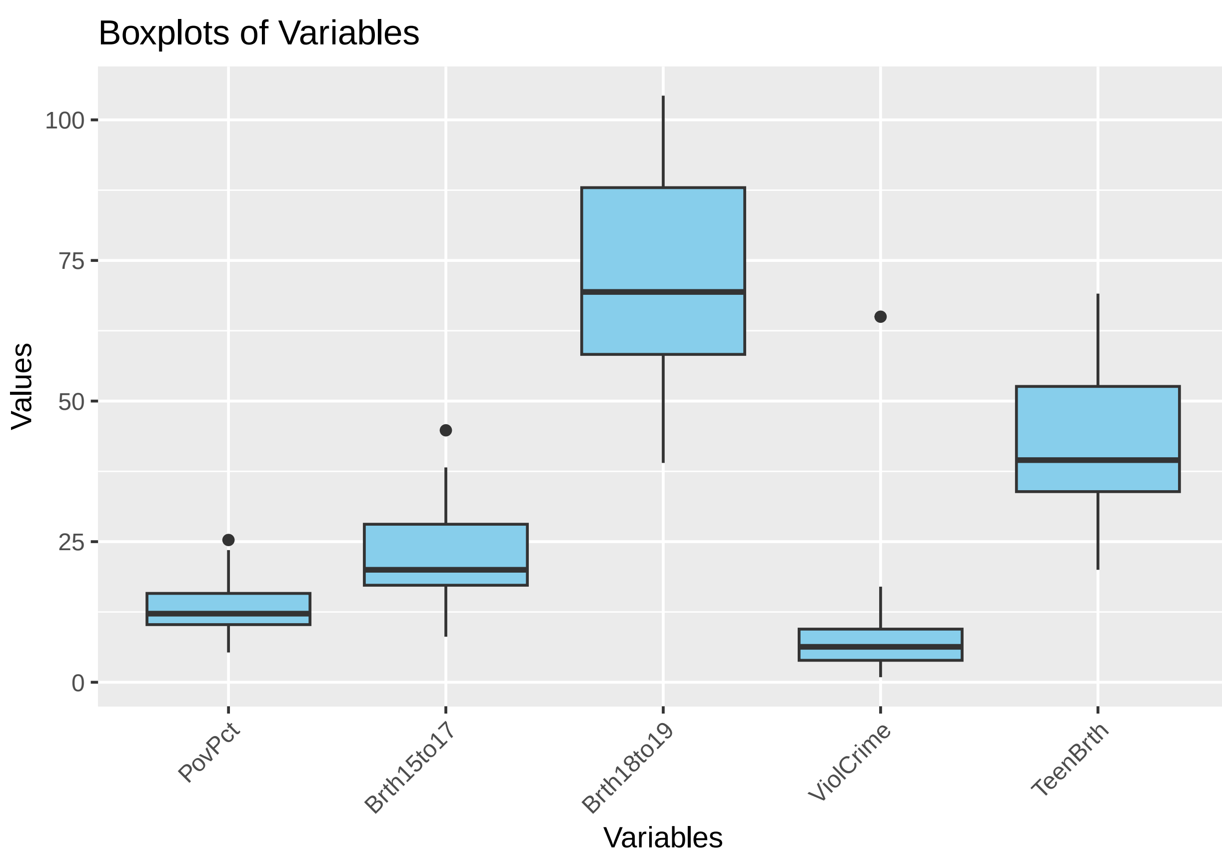
```r
plot(df)
```



```r
library(ggplot2)
library(reshape2)

# Reshape data for ggplot
df_melt <- melt(df, id.vars = "Location")  # Assuming 'Product_id' is categorical

# Create boxplots
ggplot(df_melt, aes(x = variable, y = value)) +
  geom_boxplot(fill = "skyblue") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Boxplots of Variables", x = "Variables", y = "Values")
```




```r
colSums(is.na(df))  # Shows the count of missing values in each column
```

```
##   Location     PovPct Brth15to17 Brth18to19  ViolCrime   TeenBrth
##          0          0          0          0          0          0
```

```r
unique(df$Location)
```

```
##  [1] "Alabama"               "Alaska"                "Arizona"
##  [4] "Arkansas"              "California"            "Colorado"
##  [7] "Connecticut"           "Delaware"              "District_of_Columbia"
## [10] "Florida"               "Georgia"               "Hawaii"
## [13] "Idaho"                 "Illinois"              "Indiana"
```

```
## [16] "Iowa"                 "Kansas"               "Kentucky"
## [19] "Louisiana"            "Maine"                "Maryland"
## [22] "Massachusetts"        "Michigan"             "Minnesota"
## [25] "Mississippi"          "Missouri"             "Montana"
## [28] "Nebraska"             "Nevada"               "New_Hampshire"
## [31] "New_Jersey"           "New_Mexico"           "New_York"
## [34] "North_Carolina"       "North_Dakota"         "Ohio"
## [37] "Oklahoma"             "Oregon"               "Pennsylvania"
## [40] "Rhode_Island"         "South_Carolina"       "South_Dakota"
## [43] "Tennessee"            "Texas"                "Utah"
## [46] "Vermont"              "Virginia"             "Washington"
## [49] "West_Virginia"        "Wisconsin"            "Wyoming"
```

```
dim(df)
```

```
## [1] 51  6
```

Since we can see that there are 51 different location , so it will not impact the model so we will drop this column

```
# Drop the 'Location' column
df <- df[, !names(df) %in% "Location"]
```

```
linmodel <- lm(PovPct~. , data = df) # Fitting a linear model
summary(linmodel)
```

```
##
## Call:
## lm(formula = PovPct ~ ., data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5239 -1.9763 -0.1048  1.6729  5.6012
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.22349    1.82549   3.409  0.00136 **
## Brth15to17  -0.45769    0.44681  -1.024  0.31102
## Brth18to19  -0.82144    0.27311  -3.008  0.00426 **
## ViolCrime   -0.07786    0.06683  -1.165  0.24997
## TeenBrth     1.81957    0.66635   2.731  0.00893 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.773 on 46 degrees of freedom
## Multiple R-squared:  0.6132, Adjusted R-squared:  0.5796
## F-statistic: 18.23 on 4 and 46 DF,  p-value: 4.916e-09
```

```r
# Load necessary library
library(ggplot2)

# Fit polynomial regression models for different degrees
fit1 <- lm(PovPct ~ poly(Brth15to17, 1, raw = TRUE), data = df)  # Linear
fit2 <- lm(PovPct ~ poly(Brth15to17, 2, raw = TRUE), data = df)  # Quadratic
fit3 <- lm(PovPct ~ poly(Brth15to17, 3, raw = TRUE), data = df)  # Cubic
fit4 <- lm(PovPct ~ poly(Brth15to17, 4, raw = TRUE), data = df)  # Quartic
anova(fit1, fit2, fit3, fit4)
```

```
## Analysis of Variance Table
##
## Model 1: PovPct ~ poly(Brth15to17, 1, raw = TRUE)
## Model 2: PovPct ~ poly(Brth15to17, 2, raw = TRUE)
## Model 3: PovPct ~ poly(Brth15to17, 3, raw = TRUE)
## Model 4: PovPct ~ poly(Brth15to17, 4, raw = TRUE)
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     49 426.88
## 2     48 409.38  1    17.496 2.1663 0.14788
## 3     47 409.38  1     0.000 0.0000 0.99649
## 4     46 371.52  1    37.864 4.6882 0.03559 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Fit polynomial regression models for different degrees
fit1 <- lm(PovPct ~ poly(TeenBrth, 1, raw = TRUE), data = df)  # Linear
fit2 <- lm(PovPct ~ poly(TeenBrth, 2, raw = TRUE), data = df)  # Quadratic
fit3 <- lm(PovPct ~ poly(TeenBrth, 3, raw = TRUE), data = df)  # Cubic
fit4 <- lm(PovPct ~ poly(TeenBrth, 4, raw = TRUE), data = df)  # Quartic
anova(fit1, fit2, fit3, fit4)
```

```
## Analysis of Variance Table
##
## Model 1: PovPct ~ poly(TeenBrth, 1, raw = TRUE)
## Model 2: PovPct ~ poly(TeenBrth, 2, raw = TRUE)
## Model 3: PovPct ~ poly(TeenBrth, 3, raw = TRUE)
## Model 4: PovPct ~ poly(TeenBrth, 4, raw = TRUE)
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     49 462.30
## 2     48 421.43  1    40.870 4.8867 0.03208 *
## 3     47 406.27  1    15.162 1.8129 0.18476
## 4     46 384.72  1    21.547 2.5763 0.11532
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(fit1, fit2, fit3, fit4)  # Lower AIC is better
```

```
##      df      AIC
## fit1  3 263.1553
## fit2  4 260.4348
## fit3  5 260.5661
## fit4  6 259.7868
```

```r
summary(fit1)$adj.r.squared
```

```
## [1] 0.4842952
```

```r
summary(fit2)$adj.r.squared
```

```
## [1] 0.5200923
```

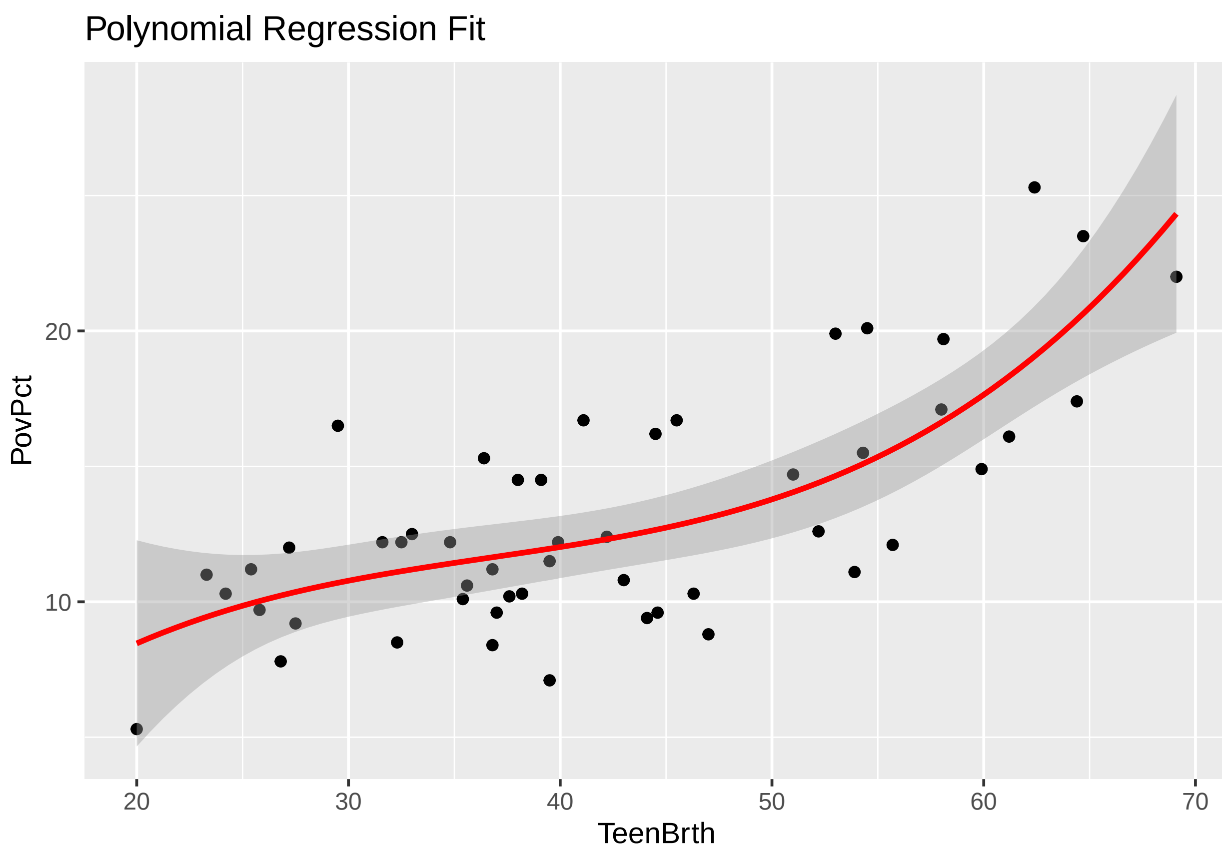
```r
summary(fit3)$adj.r.squared
```

```
## [1] 0.5275148
```

```r
summary(fit4)$adj.r.squared
```

```
## [1] 0.5428474
```

```r
ggplot(df, aes(x = TeenBrth, y = PovPct)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ poly(x, 3, raw = TRUE), color = "red") +
  labs(title = "Polynomial Regression Fit", x = "TeenBrth", y = "PovPct")
```



```r
library(splines)   # For spline regression
library(mgcv)      # For Generalized Additive Model (GAM)
```

```
## Warning: package 'mgcv' was built under R version 4.4.3
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```r
library(ggplot2)   # For visualization

# Fit a spline model with 3 knots
fit_spline_3 <- lm(PovPct ~ bs(TeenBrth, knots = c(10, 20, 30)), data = df)

# Fit a spline model with 5 knots
fit_spline_5 <- lm(PovPct ~ bs(TeenBrth, knots = c(10, 15, 20, 25, 30)), data = df)

# Compare models
anova(fit_spline_3, fit_spline_5)
```

```
## Analysis of Variance Table
##
## Model 1: PovPct ~ bs(TeenBrth, knots = c(10, 20, 30))
## Model 2: PovPct ~ bs(TeenBrth, knots = c(10, 15, 20, 25, 30))
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     46 389.09
## 2     45 388.74  1   0.34999 0.0405 0.8414
```

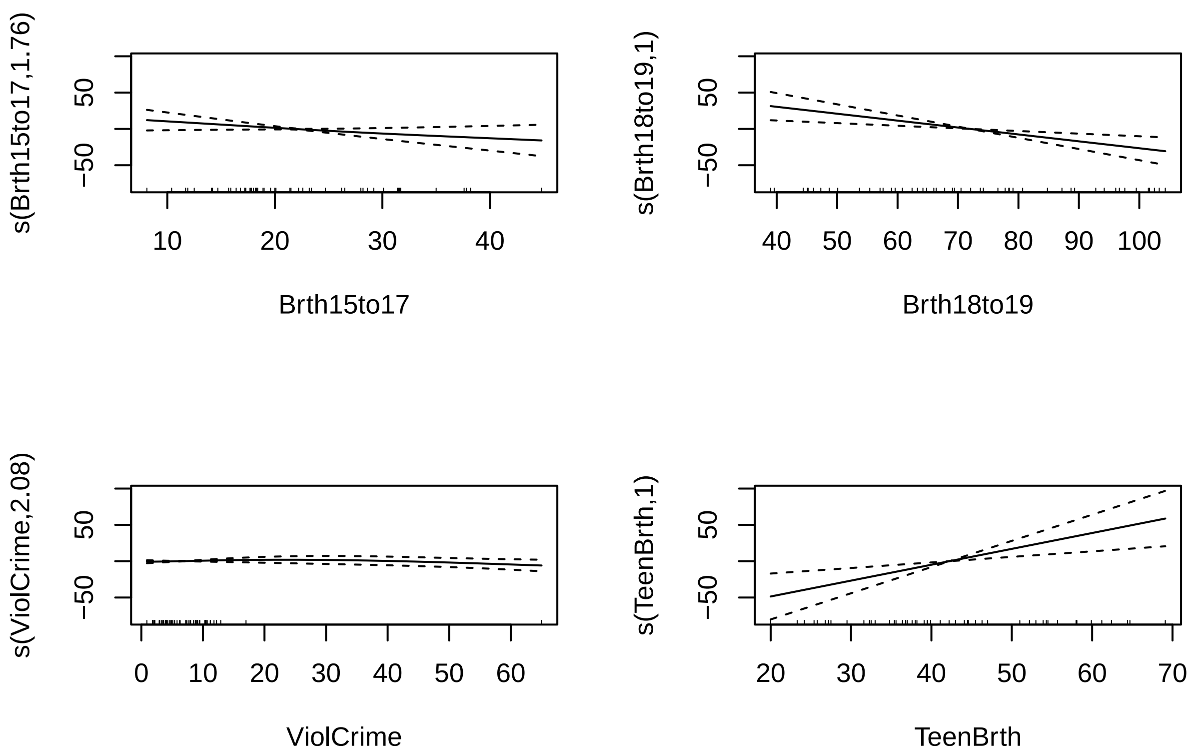
```r
# Fit a GAM model using smoothing splines
gam_fit <- gam(PovPct ~ s(Brth15to17) + s(Brth18to19) + s(ViolCrime) + s(TeenBrth), data = df)

# Summary of the model
summary(gam_fit)
```

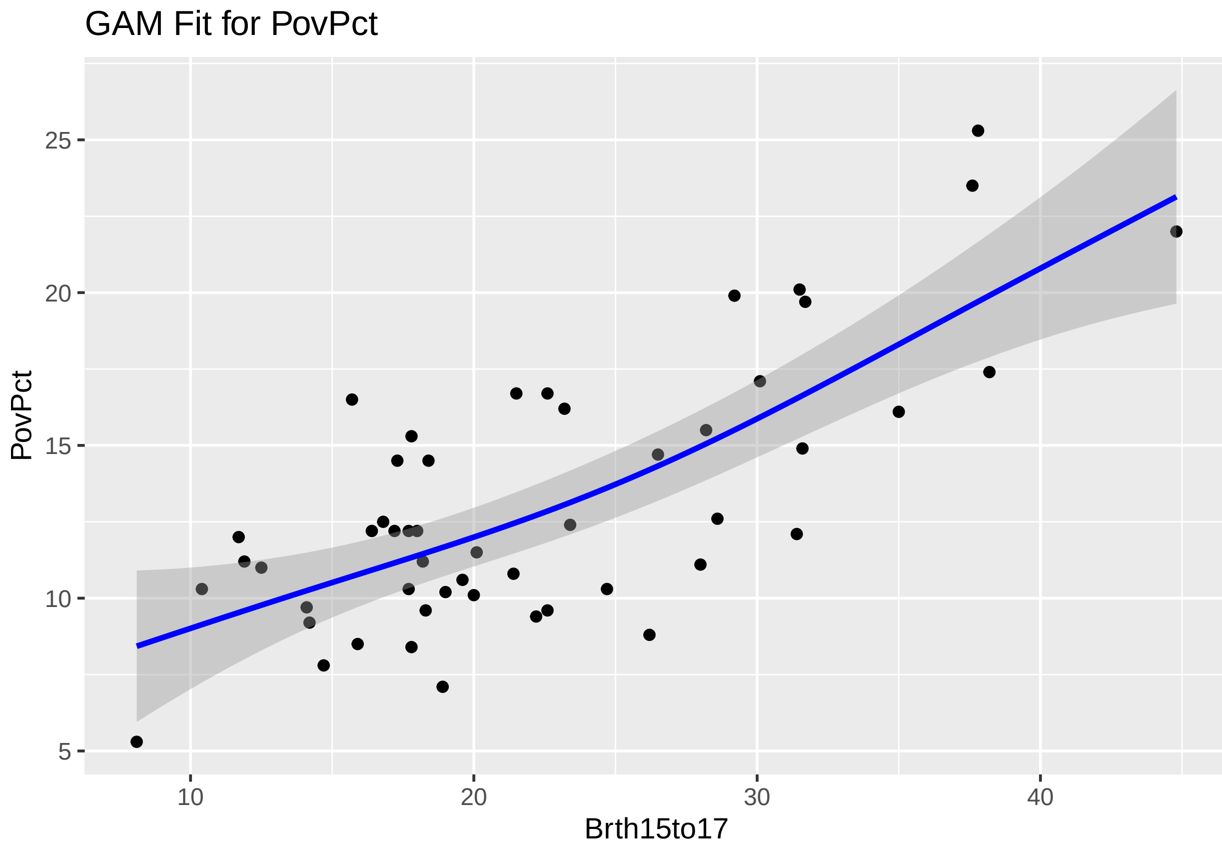
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## PovPct ~ s(Brth15to17) + s(Brth18to19) + s(ViolCrime) + s(TeenBrth)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.1176     0.3777   34.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                  edf Ref.df      F p-value
## s(Brth15to17) 1.7630 2.2300  1.624 0.20315
## s(Brth18to19) 0.9999 0.9999 10.432 0.00235 **
```

```
## s(ViolCrime)   2.0816 2.1533  1.698 0.16608
## s(TeenBrth)    0.9999 0.9999  9.471 0.00359 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Rank: 36/37
## R-sq.(adj) =  0.602   Deviance explained = 64.9%
## GCV = 8.4029  Scale est. = 7.2752    n = 51
```

```r
# Visualize the effect of each predictor
plot(gam_fit, pages = 1, se = TRUE)
```




```r
ggplot(df, aes(x = Brth15to17, y = PovPct)) +
  geom_point() +
  geom_smooth(method = "gam", formula = y ~ s(x), color = "blue") +
  labs(title = "GAM Fit for PovPct", x = "Brth15to17", y = "PovPct")
```



R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.