Diamonds Predictive Analysis with Smoothing Splines

```
# load the ggplot2 library and the data set "Diamonds" from that library
# then check the data using the head() function
library(ggplot2)
data(diamonds)
ddat <- diamonds
head(ddat)
```

```
## # A tibble: 6 x 10
##   carat cut       color clarity depth table price     x     y     z
##   <dbl> <ord>     <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal     E     SI2      61.5    55   326  3.95  3.98  2.43
## 2  0.21 Premium   E     SI1      59.8    61   326  3.89  3.84  2.31
## 3  0.23 Good      E     VS1      56.9    65   327  4.05  4.07  2.31
## 4  0.29 Premium   I     VS2      62.4    58   334  4.2   4.23  2.63
## 5  0.31 Good      J     SI2      63.3    58   335  4.34  4.35  2.75
## 6  0.24 Very Good J     VVS2     62.8    57   336  3.94  3.96  2.48
```

```
# check the dimensions of the data set to look at rows and columns
dim(ddat)
```

```
## [1] 53940    10
```

We have 53940 rows and 10 columns, meaning 10 variables with 53940 instances.

```
# use the View function to look at the data in a tabular format (similar to an Excel Spreadsheet)
# View(ddat)
```

Create a new variable in the dataset called "luxury" that assigns a value of 1 to any diamonds with a selling price $10,000 or greater, and a 0 otherwise. Use the table function in R to check that the luxury variable has 5,223 observations coded as a 1.

```
# new variable called "luxury"
add_luxury_column <- function(ddat) {
  ddat$luxury <- ifelse(ddat$price >= 10000, 1, 0)
  return(ddat)
}

# Add the luxury column to the diamonds dataset
ddat <- add_luxury_column(ddat)

# Use the table function to check that the luxury variable has 5,223 observations of 1 in the updated d
table(ddat$luxury)
```

```
##
##     0     1
## 48717  5223
```

Fit a logistic regression with the "luxury" variable as the dependent variable and the "carat" variable as the independent variable and save the results in an object called "mod1." Use the summary function to show the regression output. Is carat size significantly related to the probability that the diamond will sell for

$10,000 or more? Explain your interpretation of the regression output and specifically how carat size affects probability of a diamond having a sales price greater than $10,000. (Hint: When fitting your regression model, you'll need to use the function glm and specify the argument family=binomial in order to produce a logistic regression).

```
mod1 <- glm(formula = luxury ~ carat,
            data = ddat,
            family = "binomial")

summary(mod1)
```

```
##
## Call:
## glm(formula = luxury ~ carat, family = "binomial", data = ddat)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.54685    0.12501  -84.37   <2e-16 ***
## carat         6.83885    0.08821   77.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 34312  on 53939  degrees of freedom
## Residual deviance: 12308  on 53938  degrees of freedom
## AIC: 12312
##
## Number of Fisher Scoring iterations: 8
```

```
pscl::pR2(mod1)["McFadden"]
```

```
## fitting null model for pseudo-r2
```

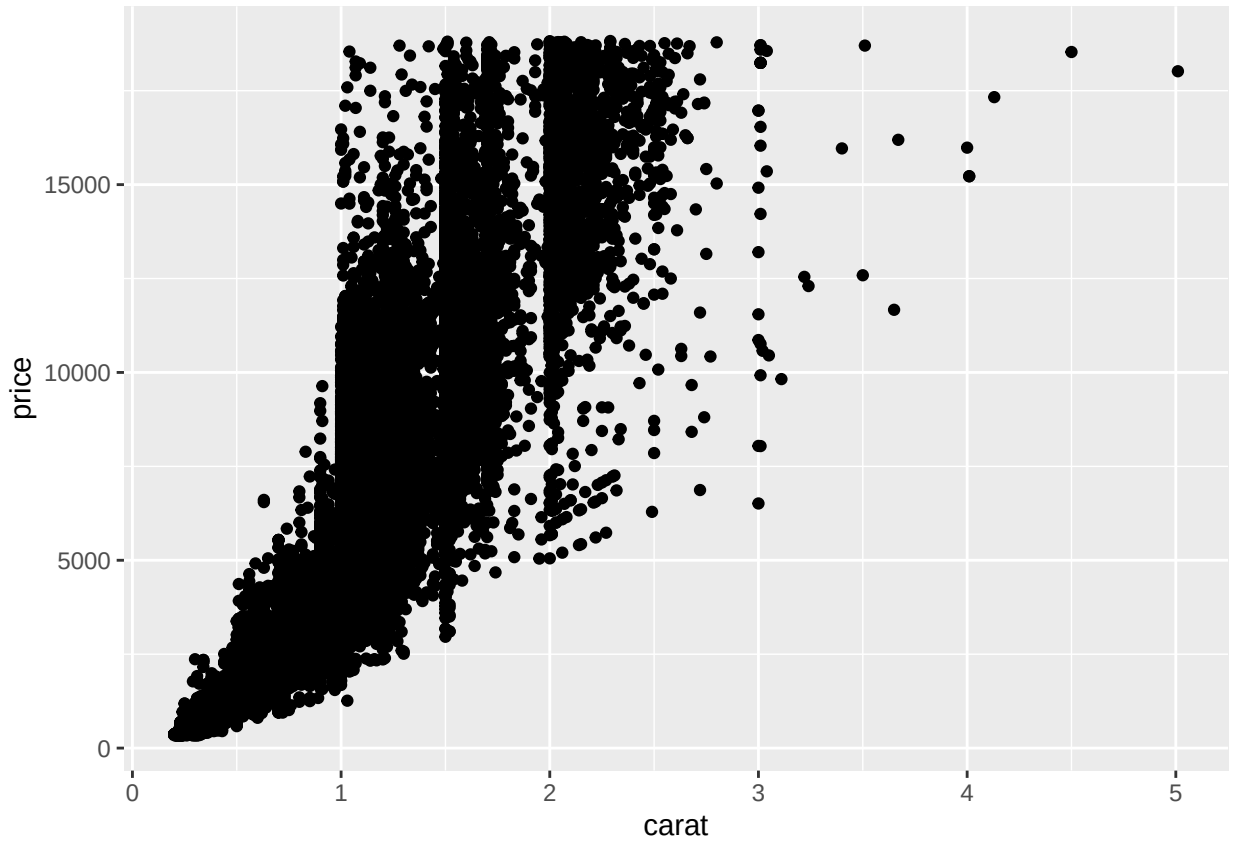
```
##  McFadden
## 0.6412906
```

Load the "splines" library into your R environment.

```
library(splines)
```

Run the following line of code in R. Include the output as part of your assignment submission. What is the line of code doing? As part of your answer, be sure to describe what relationship is being plotted. g <- ggplot(diamonds, aes(x=carat, y=price)) + geom_point()

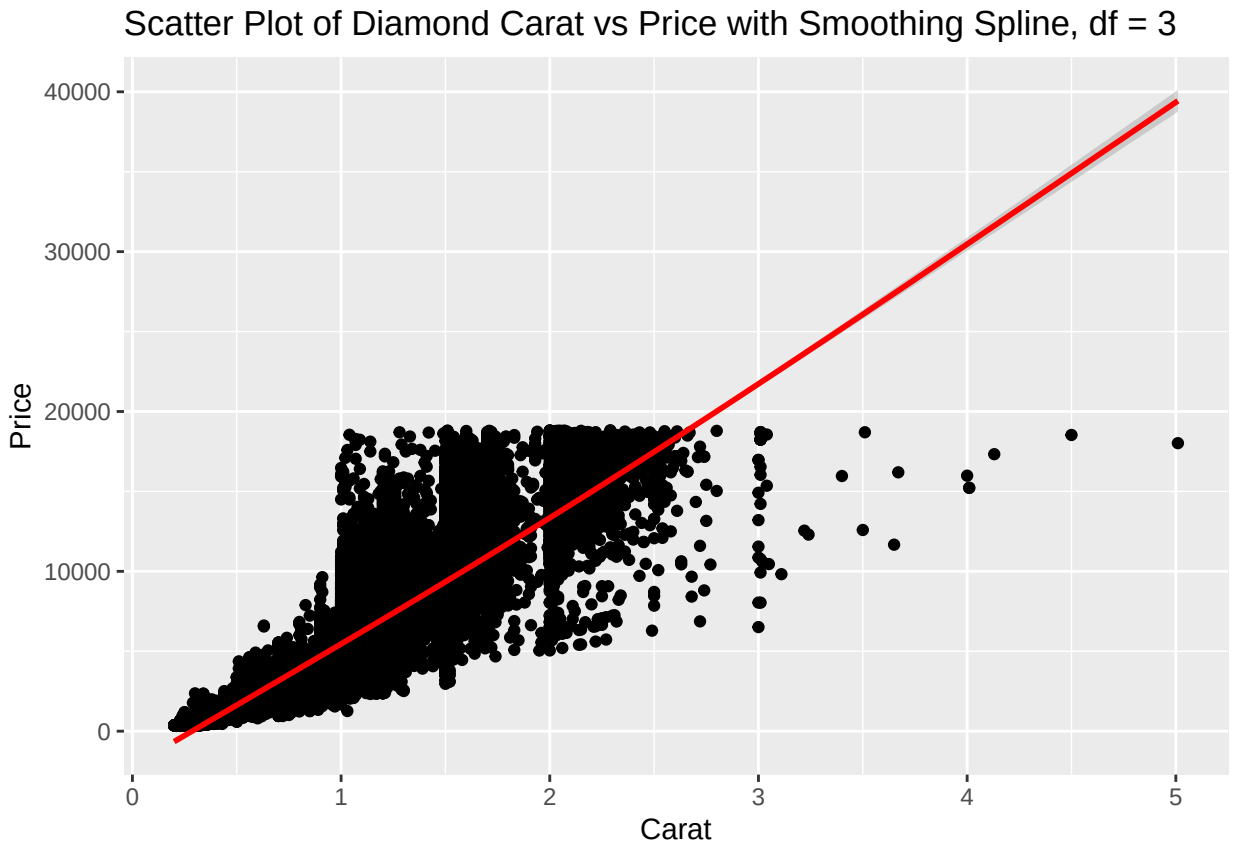
```
g <- ggplot(diamonds, aes(x=carat, y=price)) + geom_point()
g
```



Using the stat_smooth function, use the plot created in Step 2 to show three different smoothing splines, specifying df=3, df=6, and df=9 in your code. (Hint: you will need three lines of code for this step, one line for each spline. Each line of code will start with g + to call the object you created in Step 2, and add the smoothing spline to it.). Include each plot as part of your output.

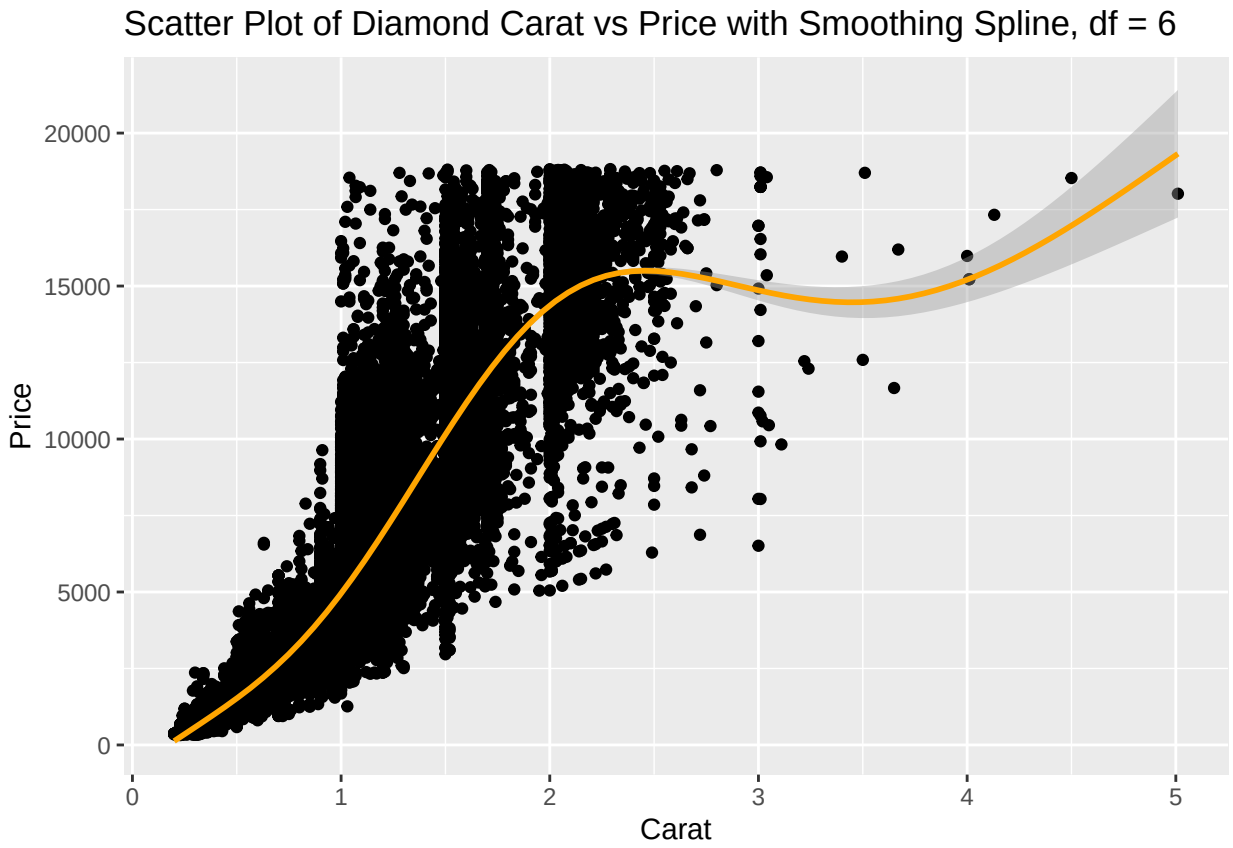
```
# Create the basic scatter plot
g <- ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point() +
  stat_smooth(method = "gam", formula = y ~ s(x, k = 3), color = "red") +
  labs(title = "Scatter Plot of Diamond Carat vs Price with Smoothing Spline, df = 3",
       x = "Carat",
       y = "Price")

# Display the plot
print(g)
```



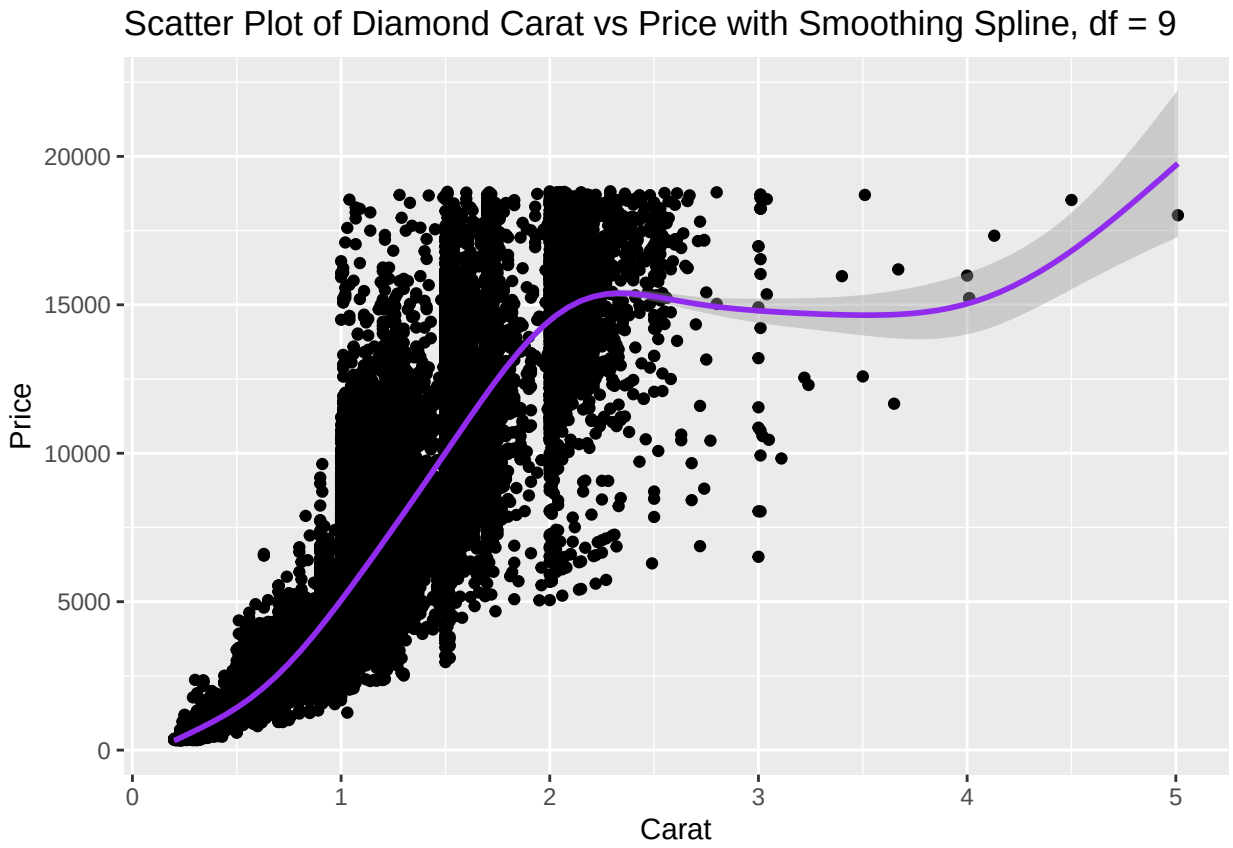

```r
# Create the scatter plot for df = 6
g <- ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point() +
  stat_smooth(method = "gam", formula = y ~ s(x, k = 6), color = "orange") +
  labs(title = "Scatter Plot of Diamond Carat vs Price with Smoothing Spline, df = 6",
       x = "Carat",
       y = "Price")

# Display the plot
print(g)
```




```r
# Create the basic scatter plot
g <- ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point() +
  stat_smooth(method = "gam", formula = y ~ s(x, k = 9), color = "purple2") +
  labs(title = "Scatter Plot of Diamond Carat vs Price with Smoothing Spline, df = 9",
       x = "Carat",
       y = "Price")

# Display the plot
print(g)
```


Scatter Plot of Diamond Carat vs Price with Smoothing Spline, df = 9
Price
20000
15000
10000
5000
0
0
1
2
3
4
5
Carat