```
required_packages = c( "tidyverse", "corrplot", "gridExtra",
                       "GGally", "cluster", "factoextra",
                       "mlbench", "class", "ggplot2", "FNN",
                       "caret", "plot3D", "Rtsne")

packages_to_install = setdiff(required_packages, installed.packages()[,"Package"])

if (length(packages_to_install)!=0) {
  install.packages(packages_to_install)
}

library(mlbench)
library(ggplot2)
library(class)
library(FNN)
library(caret)
library(dplyr)
library(tidyr)
library(corrplot)
library(cluster)
library(plot3D)
library(Rtsne)
library(GGally)
library(factoextra)

# let's set the seed to 0 for consistency
set.seed(0)
```

Data Visualization

Working with wine dataset from UCI Machine Learning Repository. The data can be downloaded directly from the repository (see code below).

Read in the data and report its dimension:

```
# read in the data using the following
wine = read.csv(
  url("https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data"),
  header = F,
  col.names = c("Wine", "Alcohol", "Malic.acid", "Ash",
                "Acl", "Mg", "Phenols", "Flavanoid",
                "Nonflavanoid.phenols", "Proanth", "Color.int",
                "Hue", "OD", "Proline"))

# display the first few columns using head()

head(wine)
```

```
##   Wine Alcohol Malic.acid  Ash  Acl  Mg Phenols Flavanoid Nonflavanoid.phenols
## 1    1   14.23       1.71 2.43 15.6 127    2.80      3.06                  0.28
```

```
## 2    1   13.20       1.78 2.14 11.2 100    2.65      2.76                0.26
## 3    1   13.16       2.36 2.67 18.6 101    2.80      3.24                0.30
## 4    1   14.37       1.95 2.50 16.8 113    3.85      3.49                0.24
## 5    1   13.24       2.59 2.87 21.0 118    2.80      2.69                0.39
## 6    1   14.20       1.76 2.45 15.2 112    3.27      3.39                0.34
##   Proanth Color.int  Hue   OD Proline
## 1    2.29      5.64 1.04 3.92    1065
## 2    1.28      4.38 1.05 3.40    1050
## 3    2.81      5.68 1.03 3.17    1185
## 4    2.18      7.80 0.86 3.45    1480
## 5    1.82      4.32 1.04 2.93     735
## 6    1.97      6.75 1.05 2.85    1450
```

```
# report dimension

dim(wine)
```

```
## [1] 178  14
```

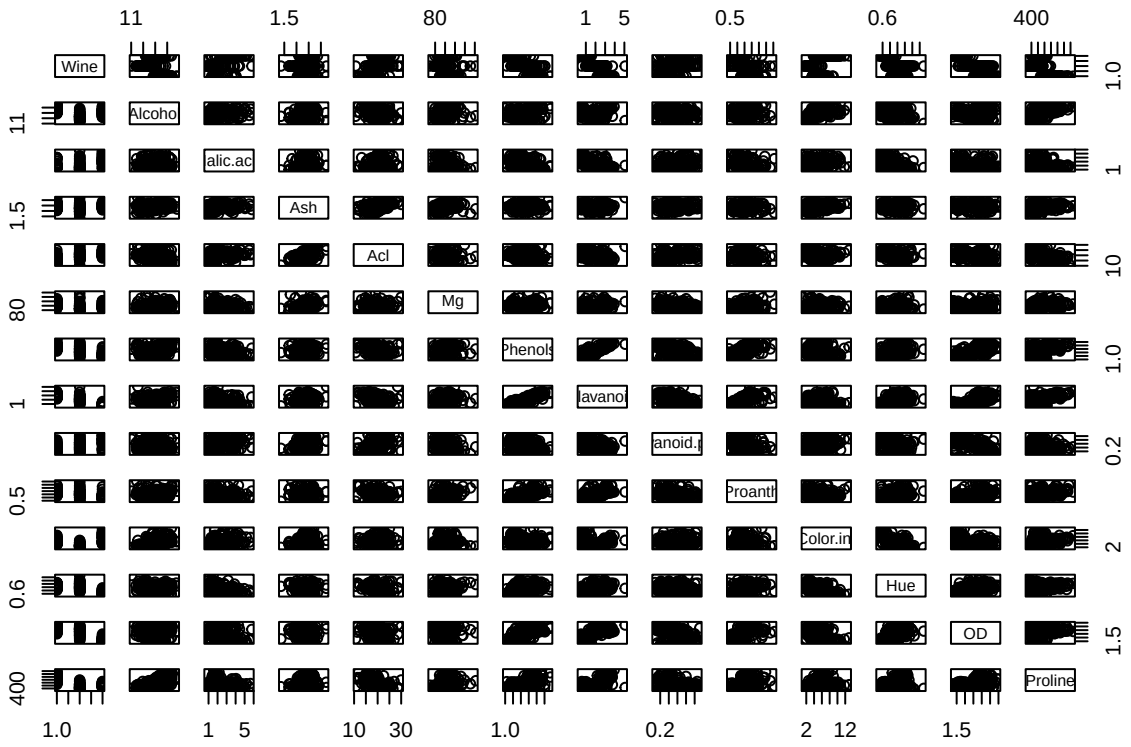
Summary statistics of the dataset:

```
# show summary statistics of the data

summary(wine)
```

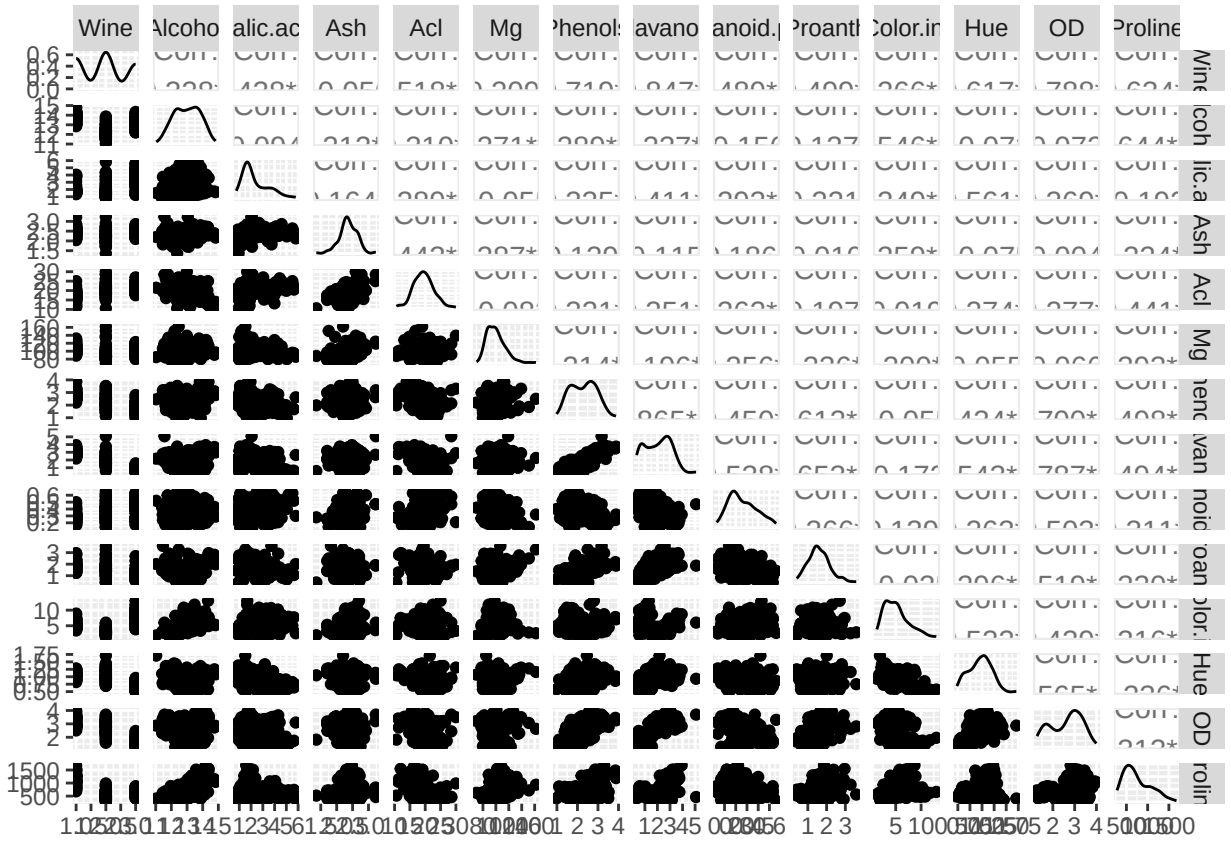
```
##       Wine          Alcohol        Malic.acid         Ash
##  Min.   :1.000   Min.   :11.03   Min.   :0.740   Min.   :1.360
##  1st Qu.:1.000   1st Qu.:12.36   1st Qu.:1.603   1st Qu.:2.210
##  Median :2.000   Median :13.05   Median :1.865   Median :2.360
##  Mean   :1.938   Mean   :13.00   Mean   :2.336   Mean   :2.367
##  3rd Qu.:3.000   3rd Qu.:13.68   3rd Qu.:3.083   3rd Qu.:2.558
##  Max.   :3.000   Max.   :14.83   Max.   :5.800   Max.   :3.230
##       Acl              Mg           Phenols        Flavanoid
##  Min.   :10.60   Min.   : 70.00   Min.   :0.980   Min.   :0.340
##  1st Qu.:17.20   1st Qu.: 88.00   1st Qu.:1.742   1st Qu.:1.205
##  Median :19.50   Median : 98.00   Median :2.355   Median :2.135
##  Mean   :19.49   Mean   : 99.74   Mean   :2.295   Mean   :2.029
##  3rd Qu.:21.50   3rd Qu.:107.00   3rd Qu.:2.800   3rd Qu.:2.875
##  Max.   :30.00   Max.   :162.00   Max.   :3.880   Max.   :5.080
##  Nonflavanoid.phenols    Proanth        Color.int           Hue
##  Min.   :0.1300       Min.   :0.410   Min.   : 1.280   Min.   :0.4800
##  1st Qu.:0.2700       1st Qu.:1.250   1st Qu.: 3.220   1st Qu.:0.7825
##  Median :0.3400       Median :1.555   Median : 4.690   Median :0.9650
##  Mean   :0.3619       Mean   :1.591   Mean   : 5.058   Mean   :0.9574
##  3rd Qu.:0.4375       3rd Qu.:1.950   3rd Qu.: 6.200   3rd Qu.:1.1200
##  Max.   :0.6600       Max.   :3.580   Max.   :13.000   Max.   :1.7100
##        OD           Proline
##  Min.   :1.270   Min.   : 278.0
##  1st Qu.:1.938   1st Qu.: 500.5
##  Median :2.780   Median : 673.5
##  Mean   :2.612   Mean   : 746.9
##  3rd Qu.:3.170   3rd Qu.: 985.0
##  Max.   :4.000   Max.   :1680.0
```

Pairwise relationship of the first five variables:

```
# pair plot of wine

pairs(wine)
```

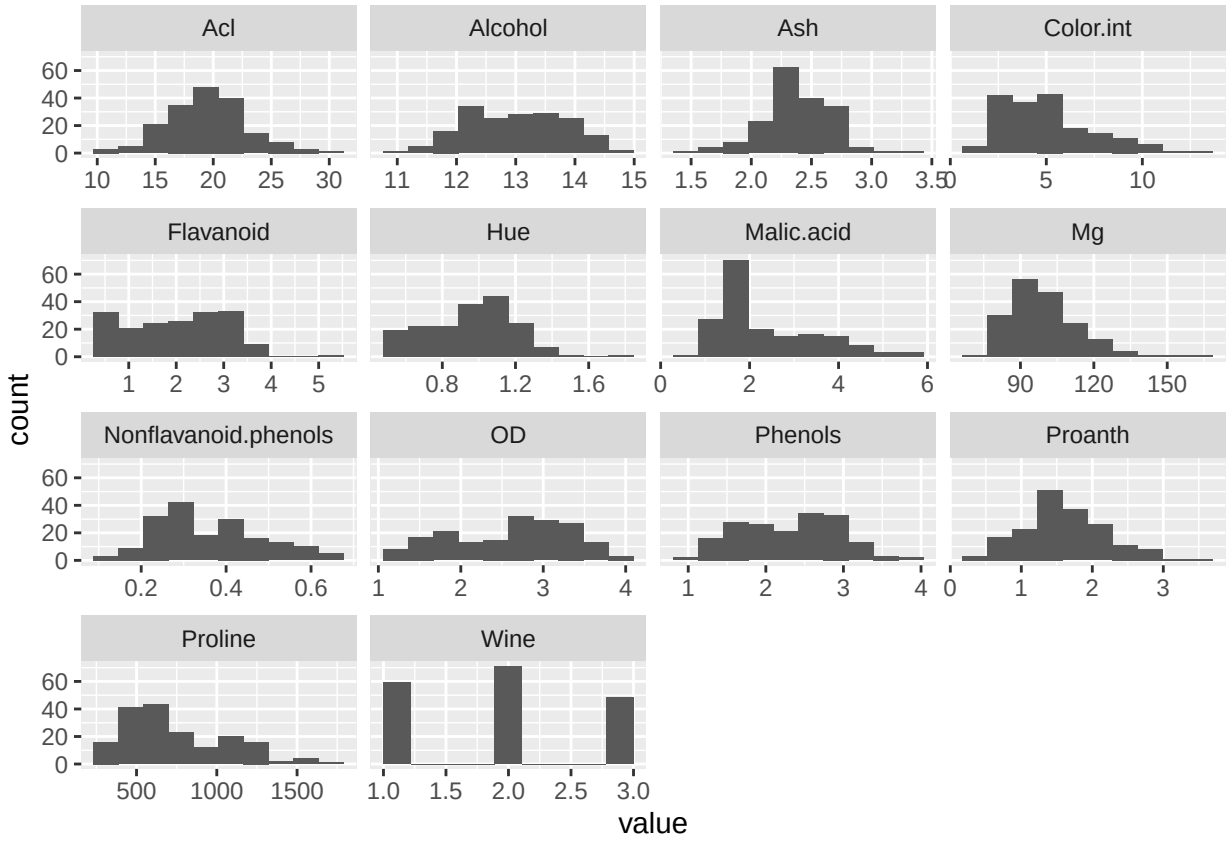



```
ggpairs(wine)
```



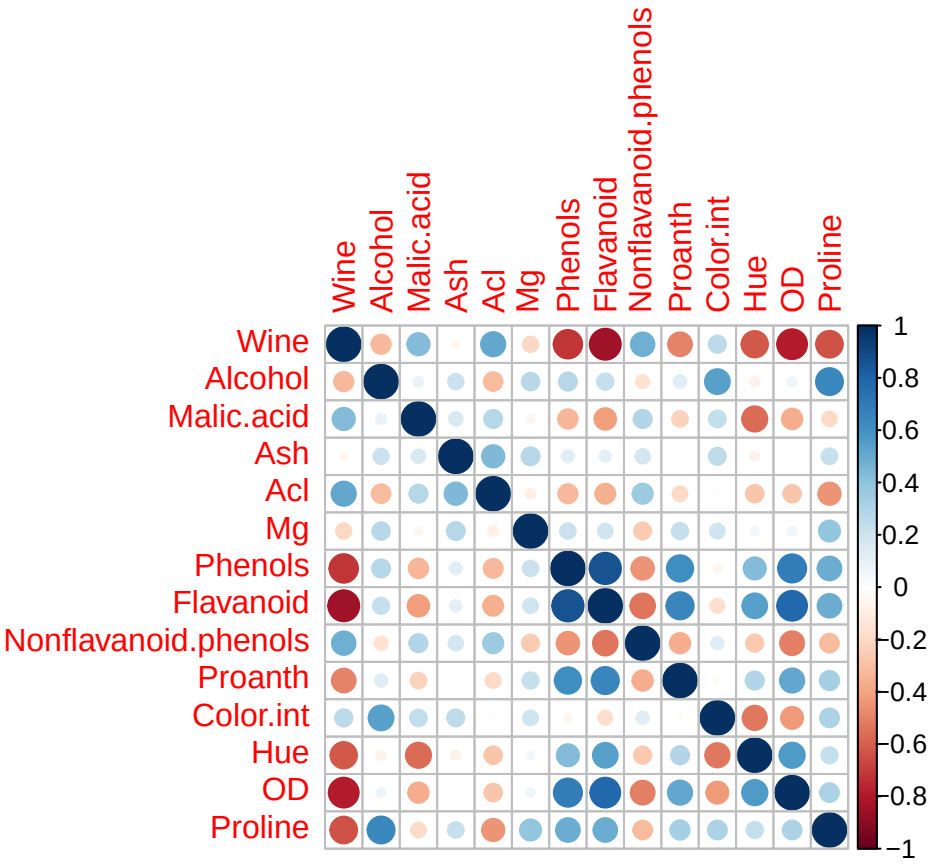
Histogram of each attribute:

```
# histogram plot of each attribute

ggplot(gather(wine), aes(value)) +
        geom_histogram(bins = 10) +
       facet_wrap(~key, scales = 'free_x')
```



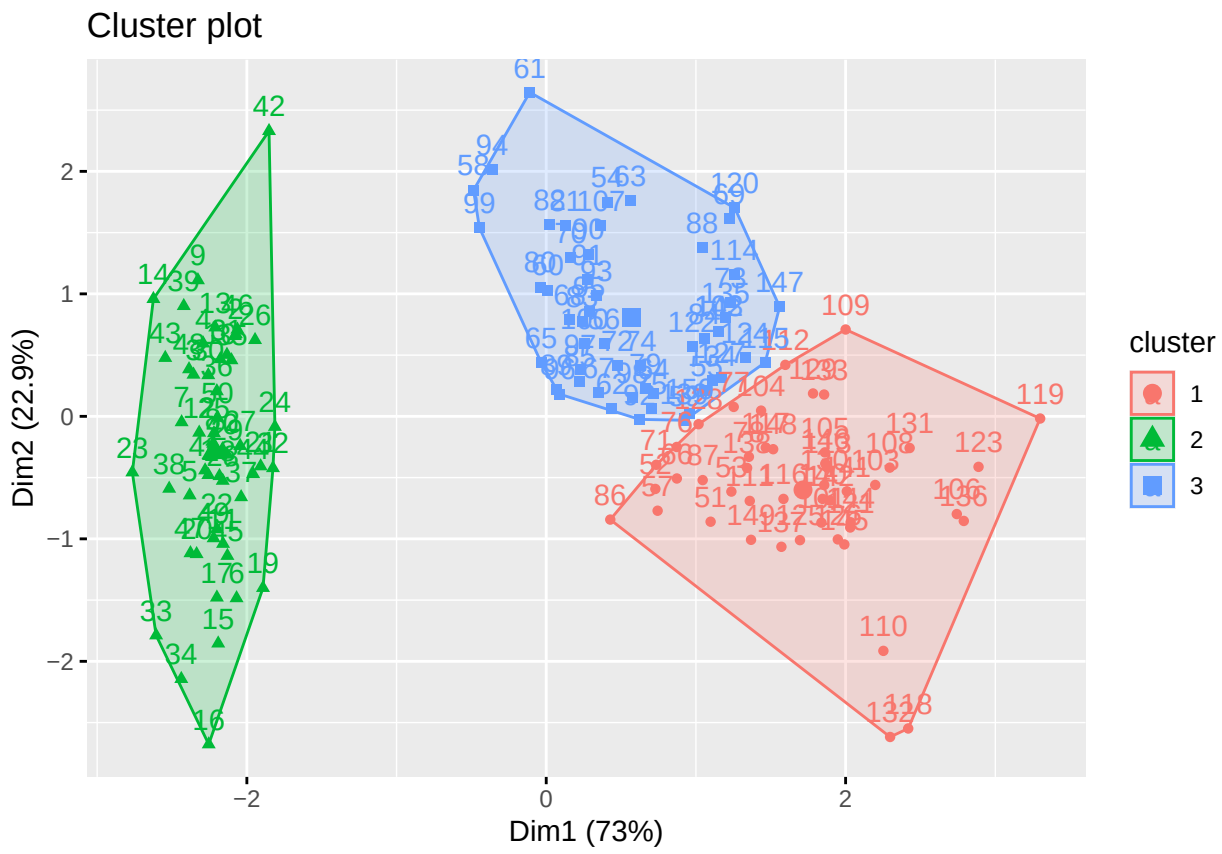
Correlation matrix:

```
# correlation plot

corrplot(cor(wine))
```



Clustering

```
# read in "iris" using data()

data(iris)

# split your data in to a features ("Sepal Length", "Sepal Width", "Petal Length",
# and "Petal Width") and target ("Species)

iris2 = subset(iris,select=-c(Species))

# normalize your features

iris3 = scale(iris2)

# fit a k-means model with k = 3

km = kmeans(iris3,centers = 3,nstart=25)
fviz_cluster(km, data = iris2)
```

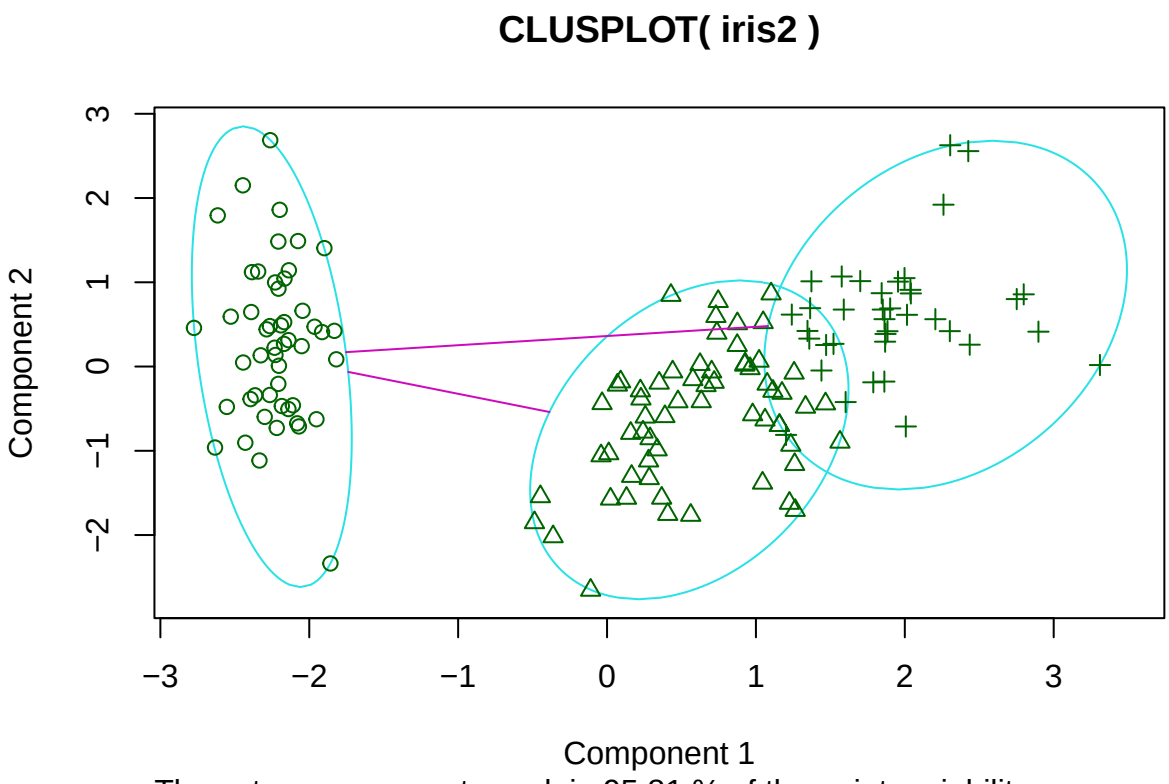



```
# compare your cluster results with the target variable using table()

table(iris$Species)
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

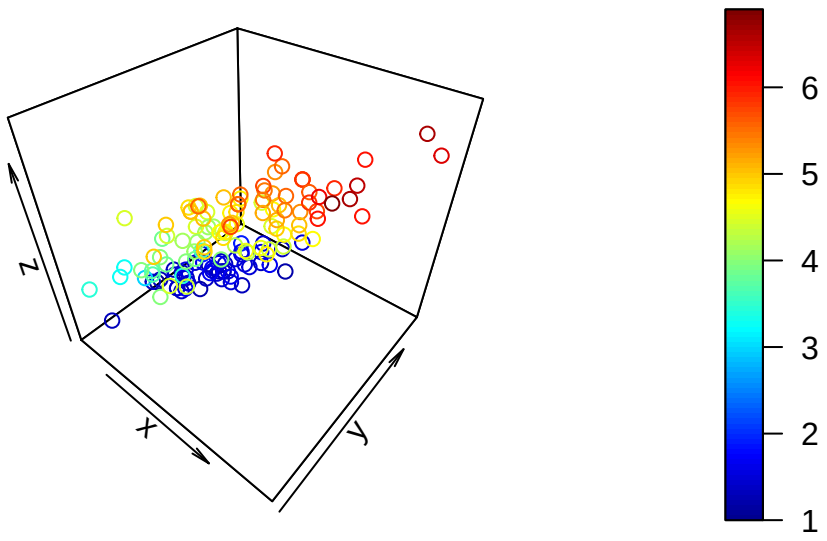
2D clustering plot:

```
# plot a 2D clustering plot using clustplot()

clusters <- pam(iris2, 3)$clustering
clusplot(iris2, clusters)
```



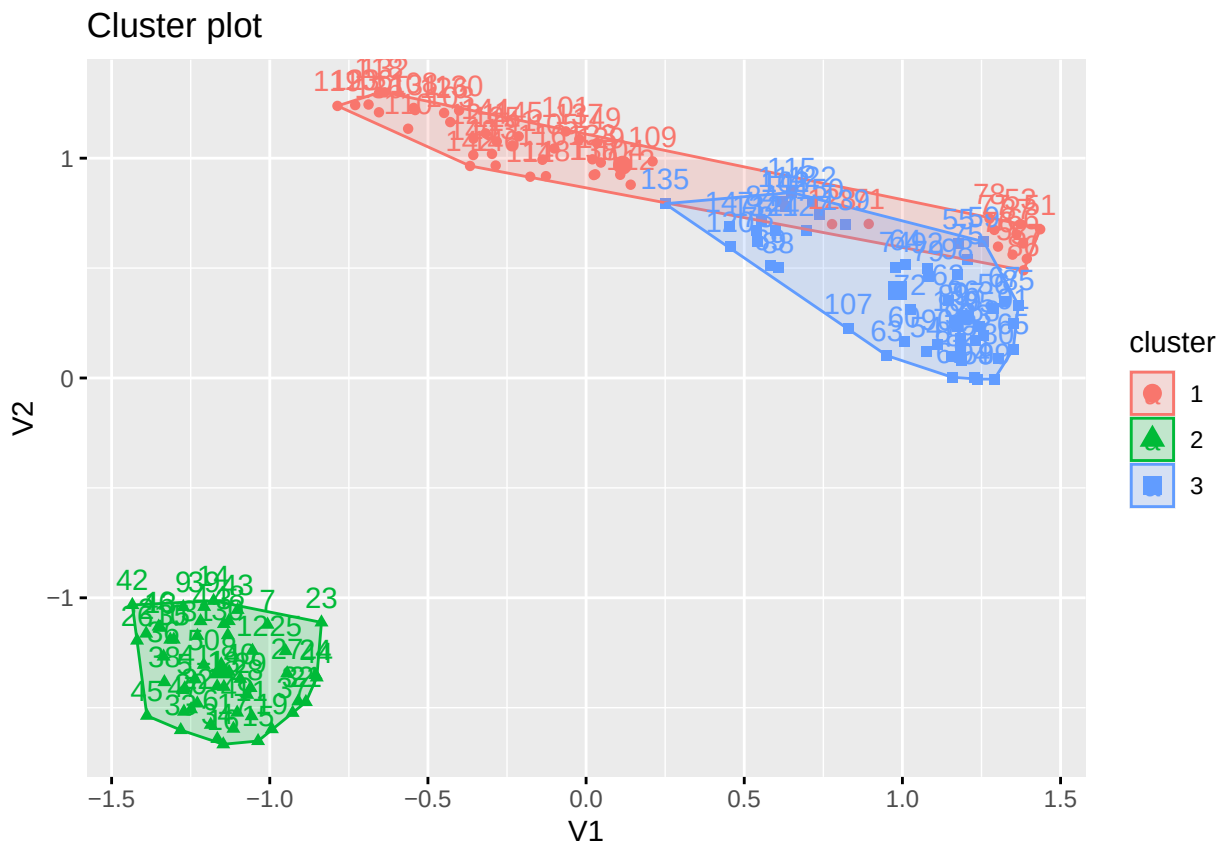
Draw a 3D plot using scatter3D:

```
# draw a 3D plot using scatter3D

scatter3D(iris2$Sepal.Length, iris2$Sepal.Width, iris2$Petal.Length, iris2$Petal.Width)
```



Visualize high-dimensional:

```
# transform your features data into a matrix

mat = as.matrix(iris2)

# use Rtsne() to reduce your dimensions, set perplexity = 20, theta = 0.5, dims = 2

tsne = Rtsne(mat, perplexity = 20, theta = 0.5, dims = 2, check_duplicates = FALSE)

# display the results of t-SNE colored by the cluster labels

dtsne = as.data.frame(tsne$Y)
fviz_cluster(km, dtsne)
```



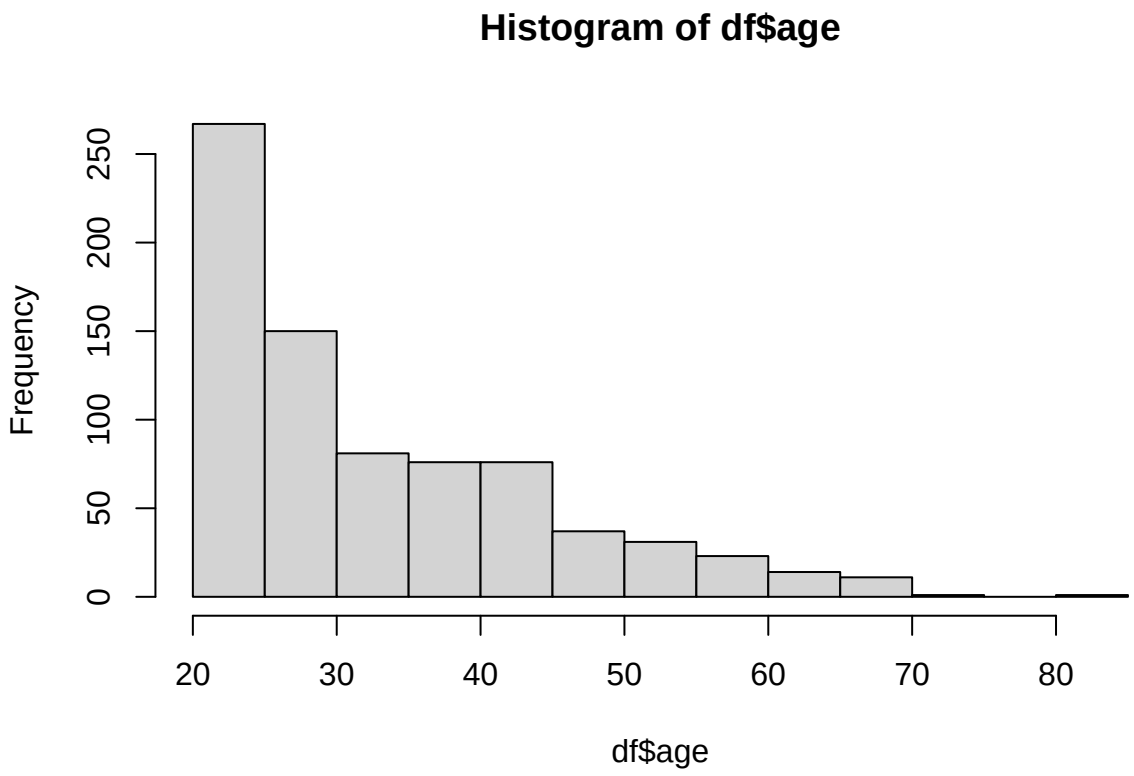
kNN Classification

```
# read in the data, store the data as "df"

data(PimaIndiansDiabetes)
df = PimaIndiansDiabetes

# describe the data using summary()

summary(df)
```

```
##     pregnant         glucose         pressure         triceps
##  Min.   : 0.000   Min.   :  0.0   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 62.00   1st Qu.: 0.00
##  Median : 3.000   Median :117.0   Median : 72.00   Median :23.00
##  Mean   : 3.845   Mean   :120.9   Mean   : 69.11   Mean   :20.54
##  3rd Qu.: 6.000   3rd Qu.:140.2   3rd Qu.: 80.00   3rd Qu.:32.00
##  Max.   :17.000   Max.   :199.0   Max.   :122.00   Max.   :99.00
##     insulin           mass           pedigree         age        diabetes
##  Min.   :  0.0   Min.   : 0.00   Min.   :0.0780   Min.   :21.00   neg:500
##  1st Qu.:  0.0   1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00   pos:268
##  Median : 30.5   Median :32.00   Median :0.3725   Median :29.00
##  Mean   : 79.8   Mean   :31.99   Mean   :0.4719   Mean   :33.24
##  3rd Qu.:127.2   3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
```

```
##  Max.   :846.0   Max.   :67.10   Max.   :2.4200   Max.   :81.00
```

```
# plot histogram of age

hist(df$age)
```




```
# proportion of individuals with diabetes

count(subset(df, diabetes=='pos'))/nrow(df)*100
```

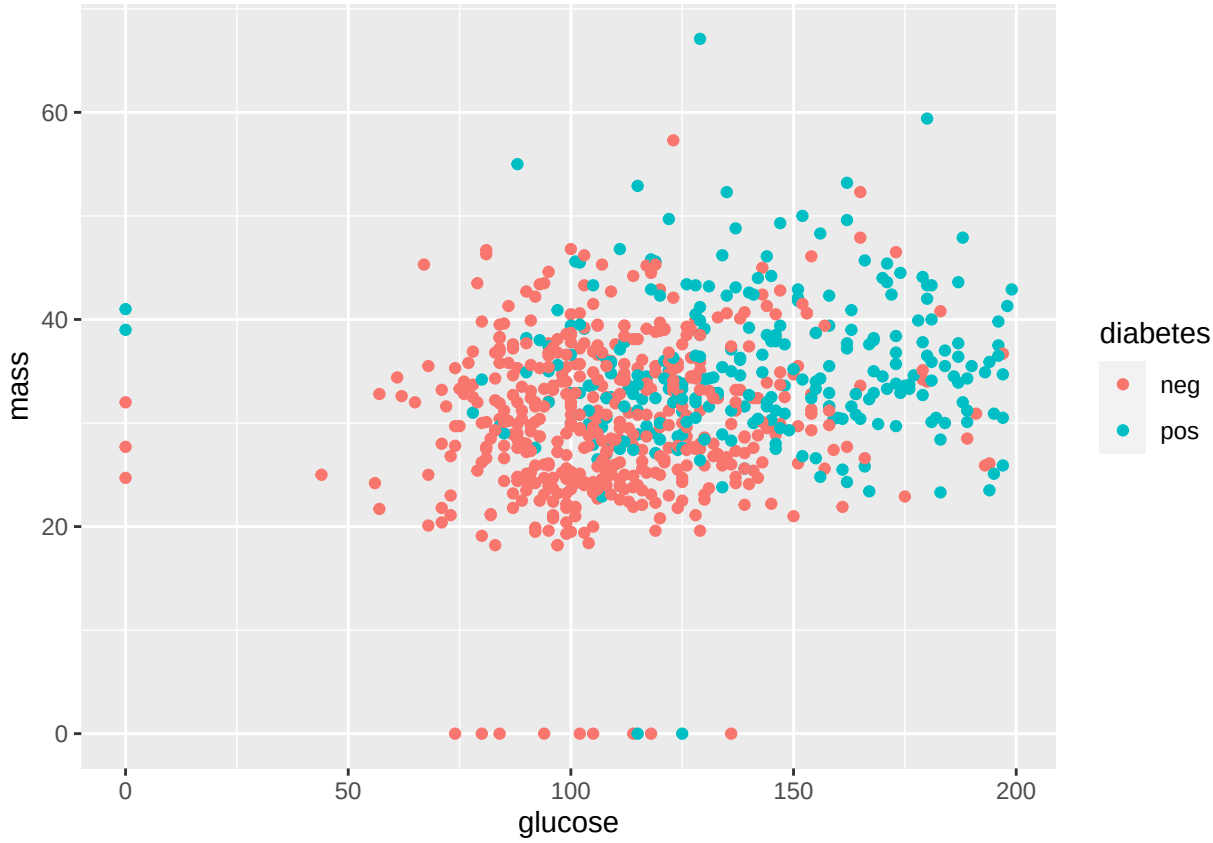
```
##          n
## 1 34.89583
```

The distribution of "age" variable is skewed with more younger individuals than older.

Scatter plot:

```
# convert "pos" in diabetes variable to factor

df$diabetes = as.factor(df$diabetes)

# plot glucose vs mass, colored by diabetes
```

```r
ggplot(df, aes(glucose, mass, colour = diabetes)) +
  geom_point()
```



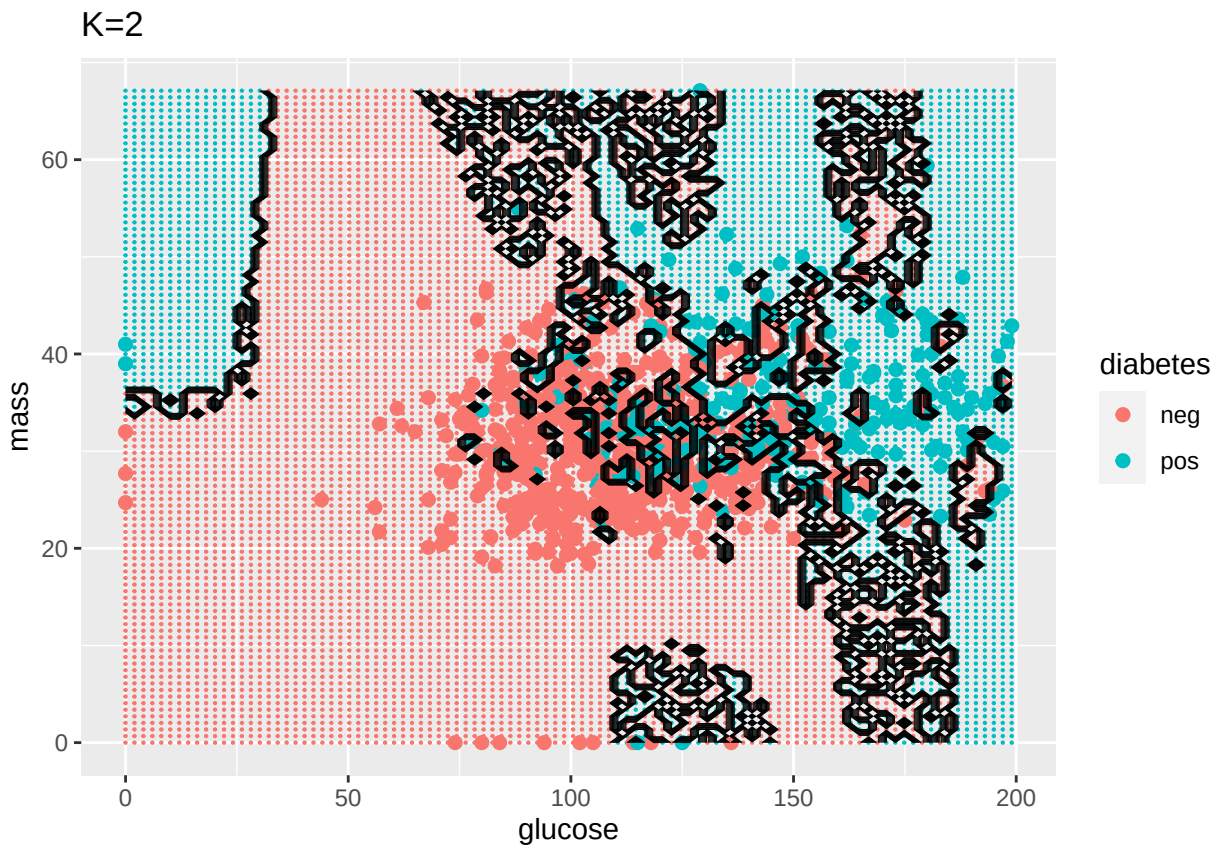
k-NN classifier:

```r
# function to plot decision boundary
decision_boundary_plot <- function(model, data, x1_var, x2_var, y_var, resolution = 100, title) {

  x1 = data[, x1_var]
  x2 = data[, x2_var]
  y = data[, y_var]

  # make grid
  xs1 <- seq(min(x1), max(x1), length.out = resolution)
  xs2 <- seq(min(x2), max(x2), length.out = resolution)
  g <- cbind(rep(xs1, each=resolution), rep(xs2, time = resolution))
  g <- as.data.frame(g)
  names(g) = c('glucose', 'mass')

  p <- predict(model, g, type = "class")

    plt = ggplot() +
    geom_point(aes(g[, 1], g[, 2], col = p), size = 0.1) +
```

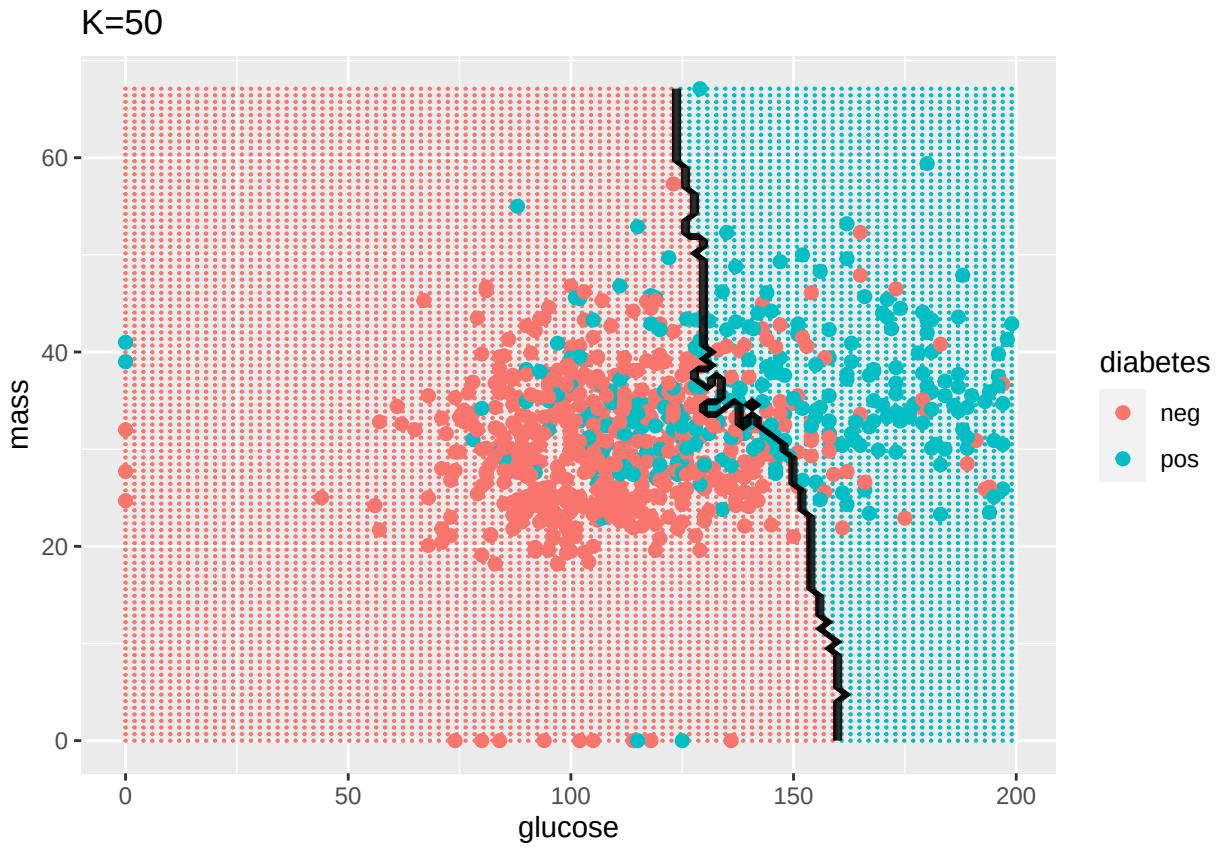
```
    geom_point(aes(x1, x2, col = y), size = 2) +
    geom_contour(
      aes(x = g[, 1], y = g[, 2],z = as.integer(p)),
      col = 'black', size = 0.1) +
    xlab('glucose') + ylab('mass') + scale_colour_discrete(name='diabetes') +
    ggtitle(title)

  return(plt)
}
```

```
# build a k-NN classifier, setting k to 2

knnmod = knn3(formula=diabetes~mass+glucose, data=df, k=2)

# visualize the decision boundary of k-NN, k = 2

decision_boundary_plot(knnmod, df, 2, 6, 9, resolution = 100, title= "K=2")
```

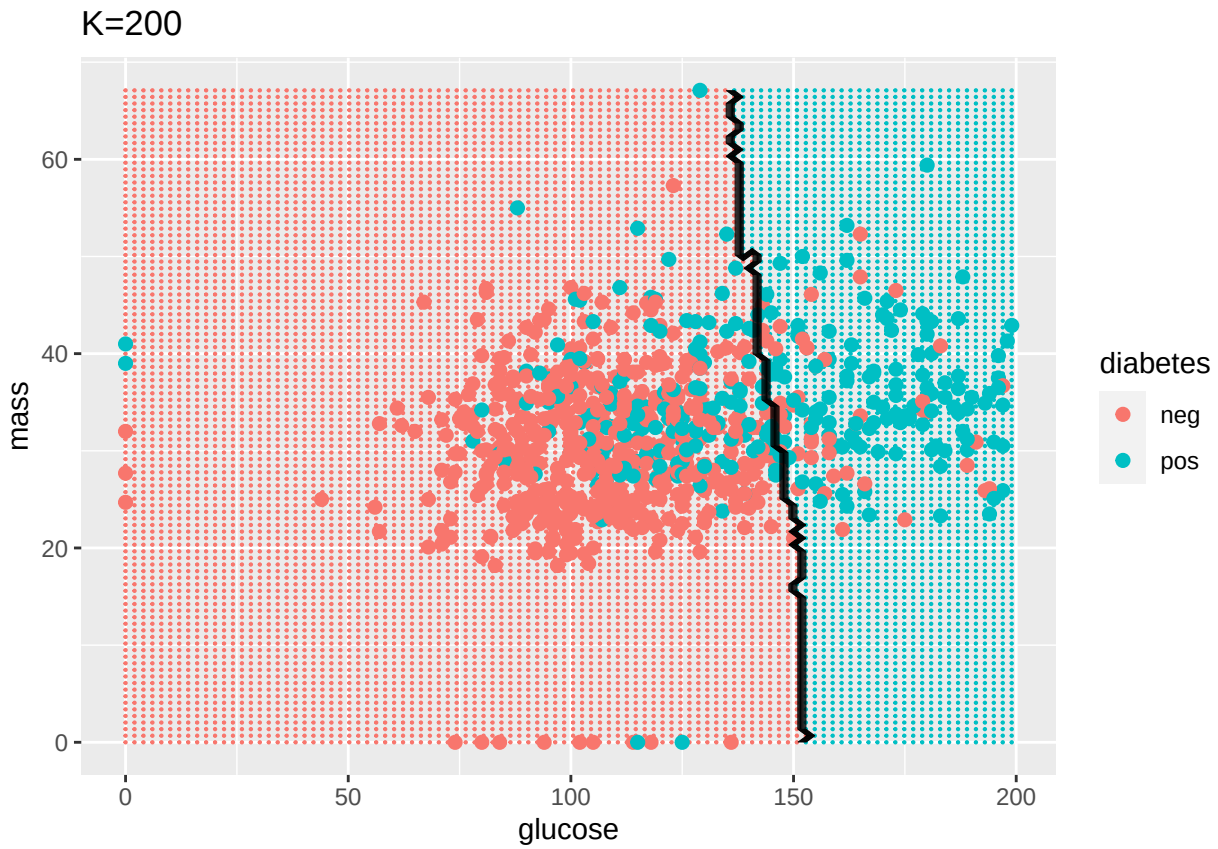
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



```
# build a k-NN classifier, setting k to 50

knnmod2 = knn3(formula=diabetes~mass+glucose, data=df, k=50)

# visualize the decision boundary of k-NN, k = 50

decision_boundary_plot(knnmod2, df, 2, 6, 9, resolution = 100, title= "K=50")
```




```
# build a k-NN classifier, setting k to 200

knnmod3 = knn3(formula=diabetes~mass+glucose, data=df, k=200)

# visualize the decision boundary of k-NN, k = 200

decision_boundary_plot(knnmod3, df, 2, 6, 9, resolution = 100, title= "K=200")
```



As k increases, the boundary becomes less defined, since there are more neighbors considered in the knn calculation. From the visualizations, it appears that as k between 2 <> 50 is optimal.

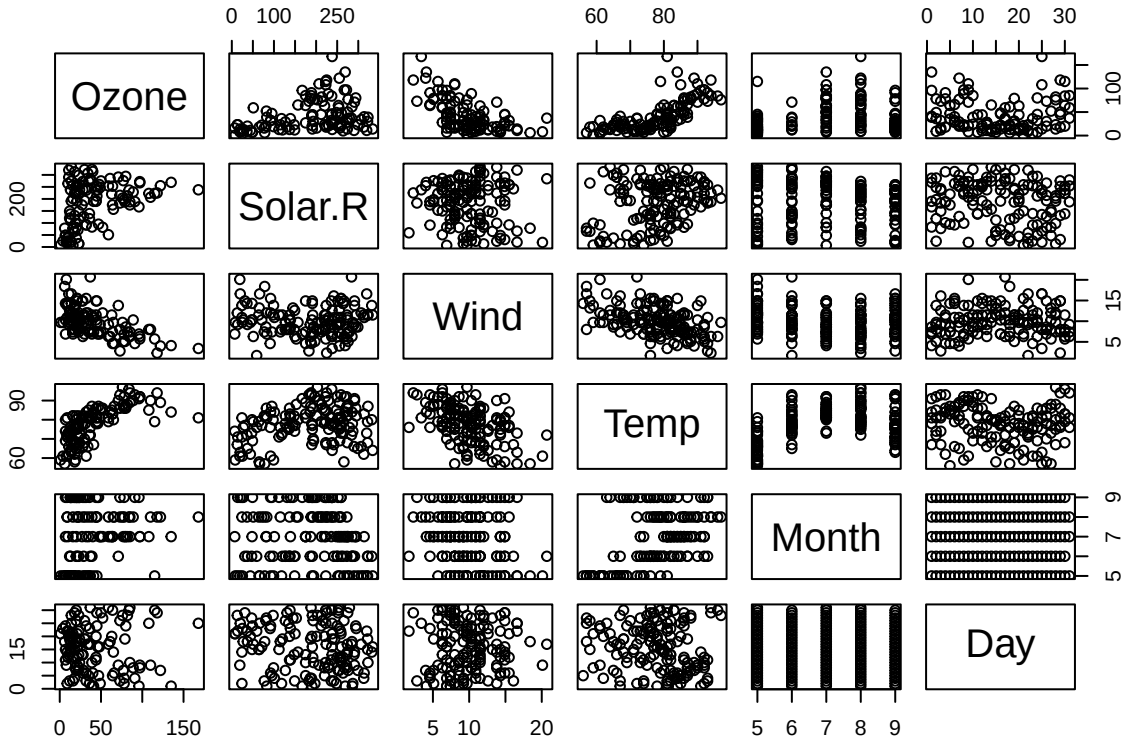
Regression

```
# read in "airquality"

data(airquality)

# store the data as df

dfair = airquality

# describe the data using summary()

summary(dfair)
```

```
##      Ozone           Solar.R          Wind             Temp
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00
```

```
##  NA's   :37       NA's   :7
##      Month            Day
##  Min.   :5.000   Min.   : 1.0
##  1st Qu.:6.000   1st Qu.: 8.0
##  Median :7.000   Median :16.0
##  Mean   :6.993   Mean   :15.8
##  3rd Qu.:8.000   3rd Qu.:23.0
##  Max.   :9.000   Max.   :31.0
##
```

```
# observe the pairwise relationships between the variables

pairs(dfair)
```



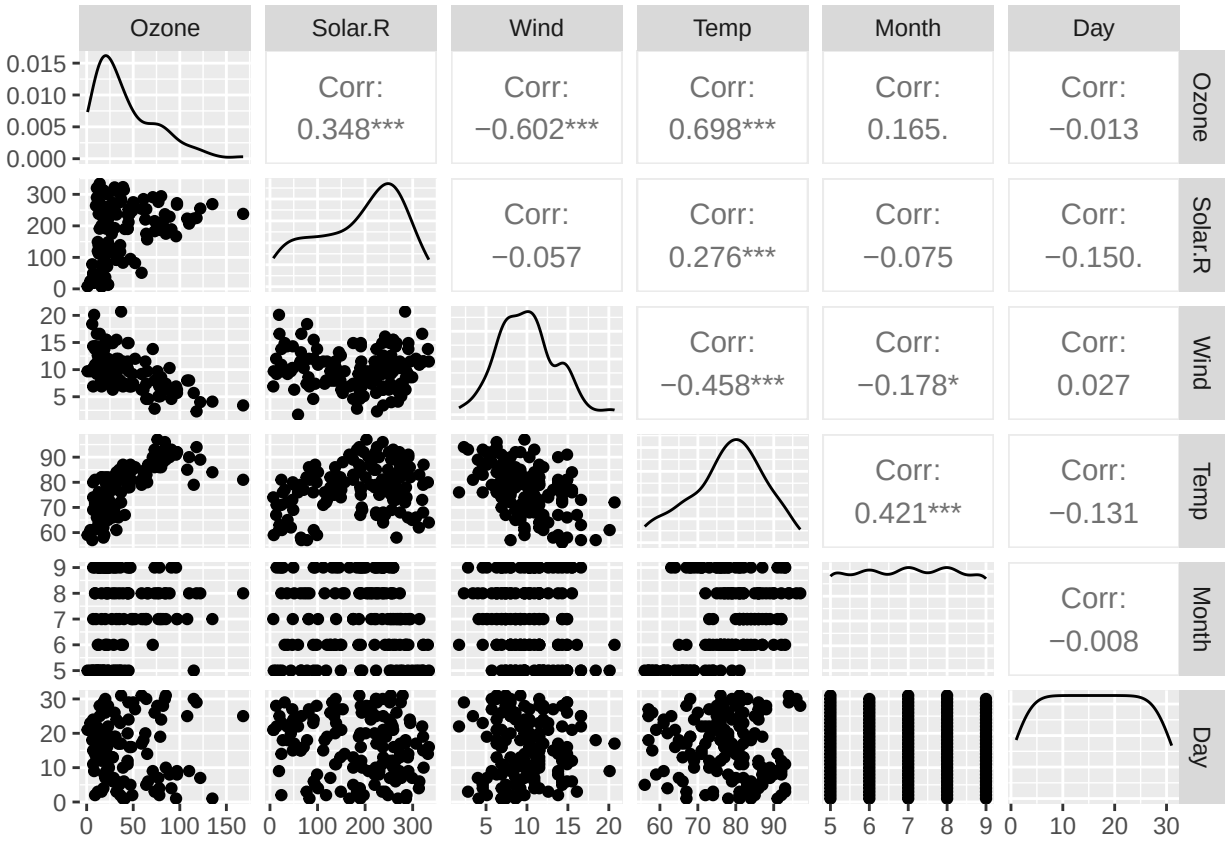

```
ggpairs(dfair)
```

```
## Warning: Removed 37 rows containing non-finite values (`stat_density()`).
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 42 rows containing missing values
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 37 rows containing missing values
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 37 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 37 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 37 rows containing missing values

## Warning: Removed 42 rows containing missing values (`geom_point()`).

## Warning: Removed 7 rows containing non-finite values (`stat_density()`).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 7 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 7 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 7 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 7 rows containing missing values

## Warning: Removed 37 rows containing missing values (`geom_point()`).

## Warning: Removed 7 rows containing missing values (`geom_point()`).

## Warning: Removed 37 rows containing missing values (`geom_point()`).

## Warning: Removed 7 rows containing missing values (`geom_point()`).

## Warning: Removed 37 rows containing missing values (`geom_point()`).

## Warning: Removed 7 rows containing missing values (`geom_point()`).

## Warning: Removed 37 rows containing missing values (`geom_point()`).

## Warning: Removed 7 rows containing missing values (`geom_point()`).
```

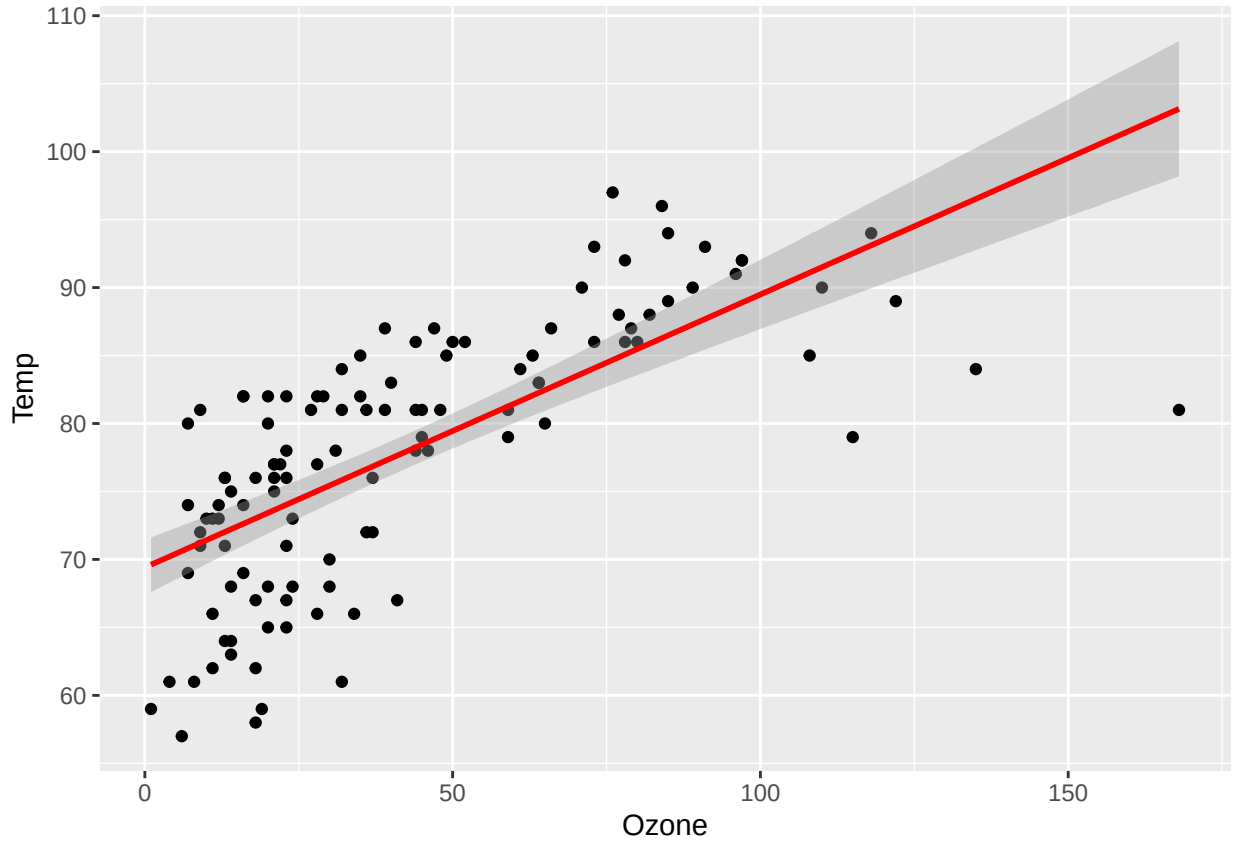


From the pair plot, most of the relationships appear to be nonlinear.

Regress:

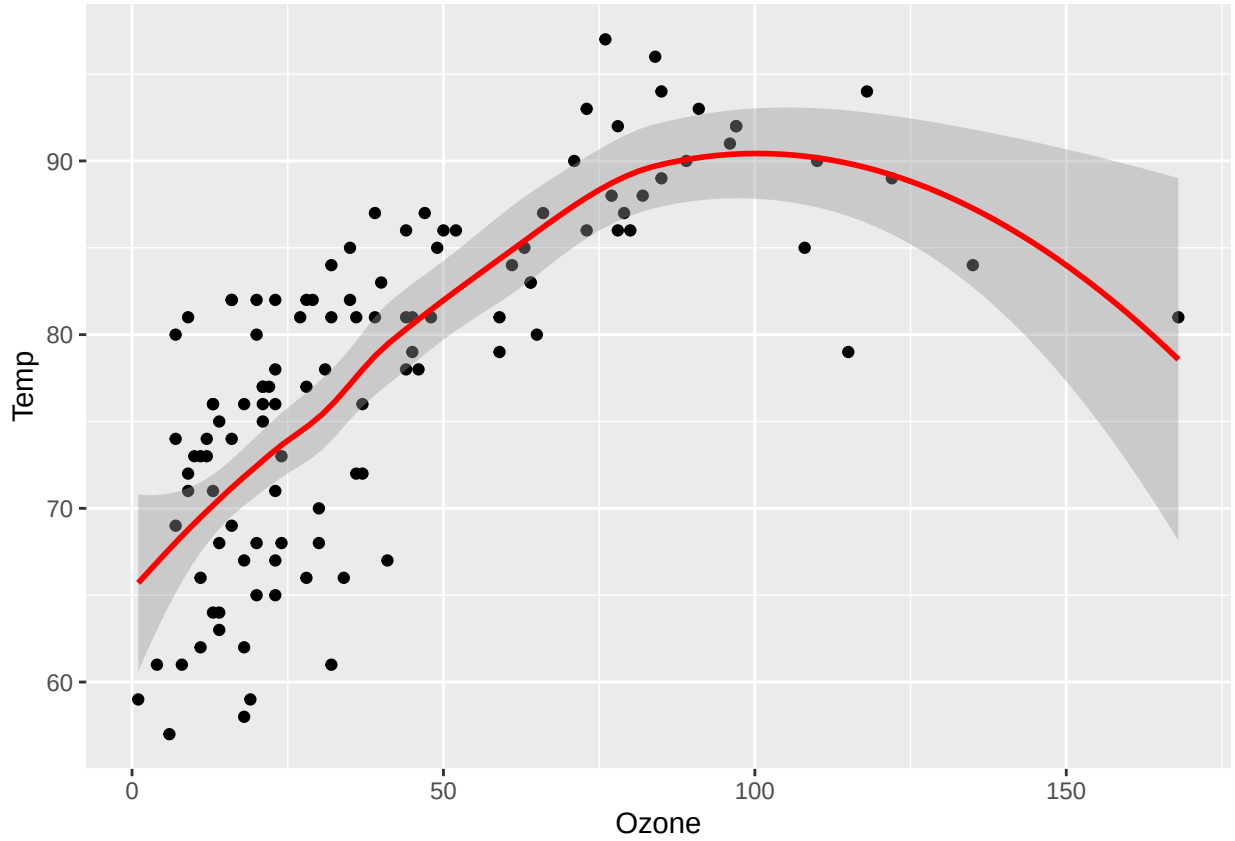
```
# read in airquality


# subset the data to include only "Ozone" and "Temp". Remove missing values.

dfair2 = cbind(dfair[,1], dfair[,4])
dfair2 = na.omit(dfair2)
dfair2 = data.frame(dfair2)
names(dfair2) = c('Ozone', 'Temp')
# regress Temp (y) on Ozone (x) using linear regression

airmod = lm(Temp~Ozone, dfair2)

ggplot(dfair2, aes(x = Ozone, y = Temp)) +
  geom_point() +
  stat_smooth(method = "lm", col = "red")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



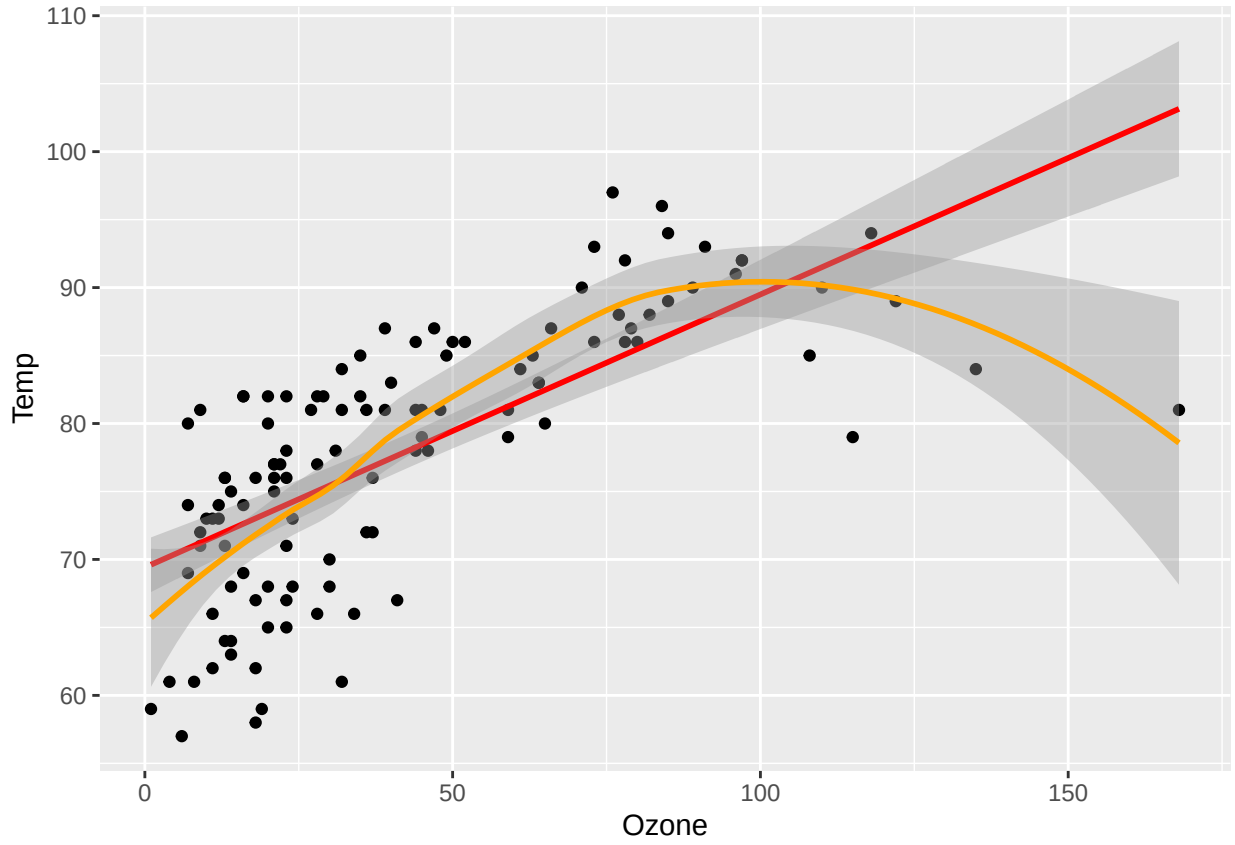

```
# regression Temp (y) on Ozone (x) using 3rd-degree polynomial regression

airmod2 = lm(Temp~poly(Ozone), dfair2)

ggplot(dfair2, aes(x = Ozone, y = Temp)) +
  geom_point() +
  stat_smooth(method = "loess", col = "red")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
# plot Temp vs Ozone with the fitted regression line

ggplot(dfair2, aes(x = Ozone, y = Temp)) +
  geom_point() +
  stat_smooth(method = "lm", col = "red") +
  stat_smooth(method = "loess", col = "orange")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



Visually, polynomial regression fits the data better.

k-NN regression:

```
# fit a kNN regression with k = 2

knnregmod = knn.reg(dfair2, y=dfair2$Temp, k=2)
dfplot = cbind(dfair2, data.frame(knnregmod$pred))
dfplot
```

```
##     Ozone Temp knnregmod.pred
## 1      41   67           72.0
## 2      36   72           74.0
## 3      12   74           73.0
## 4      18   62           62.0
## 5      28   66           69.0
## 6      23   65           66.0
## 7      19   59           60.0
## 8       8   61           61.5
## 9       7   74           72.5
## 10     16   69           67.5
## 11     11   66           64.0
## 12     14   68           70.0
## 13     18   58           60.5
```

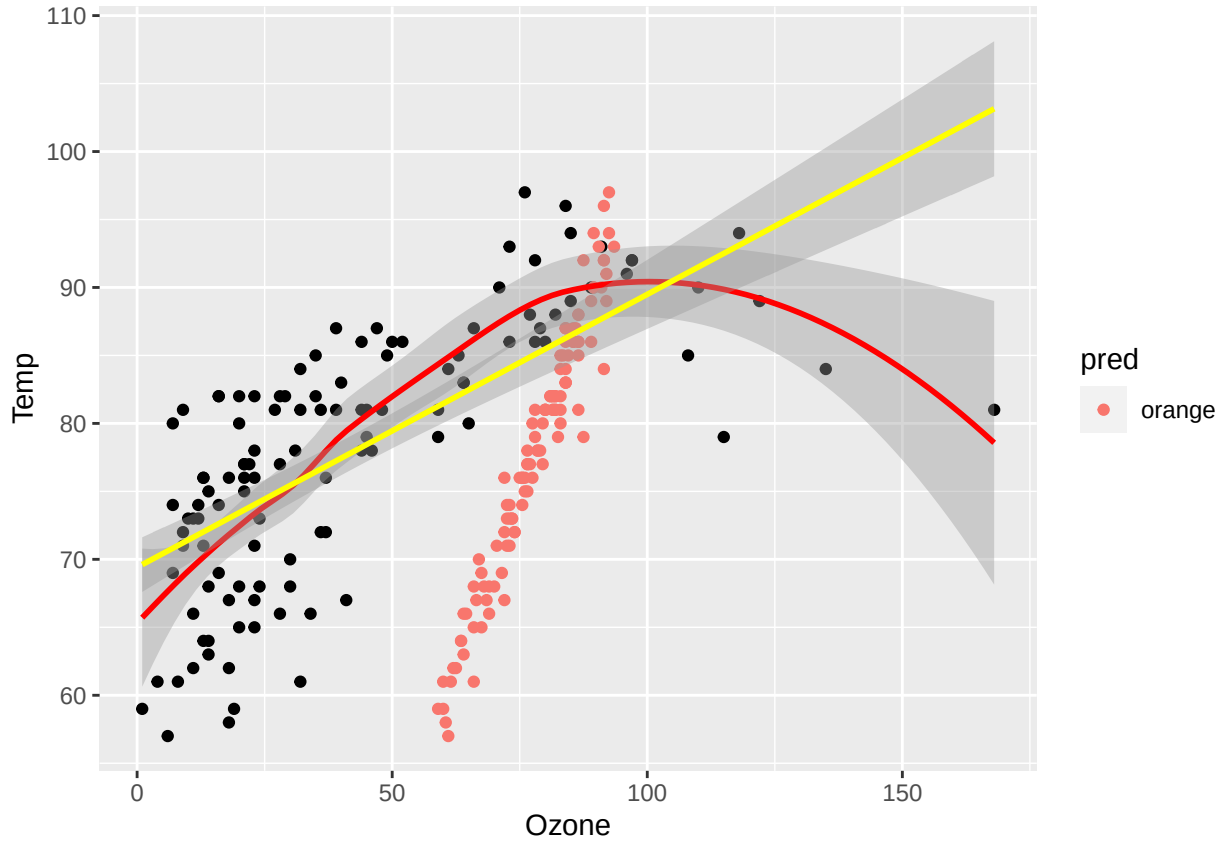
```
## 14     14   64          63.5
## 15     34   66          64.5
## 16      6   57          61.0
## 17     30   68          68.0
## 18     11   62          62.5
## 19      1   59          59.0
## 20     11   73          73.0
## 21      4   61          60.0
## 22     32   61          66.0
## 23     23   67          66.5
## 24     45   81          80.0
## 25    115   79          87.5
## 26     37   76          72.0
## 27     29   82          81.5
## 28     71   90          89.5
## 29     39   87          84.0
## 30     23   82          81.0
## 31     21   77          76.5
## 32     37   72          74.0
## 33     20   65          67.5
## 34     12   73          73.5
## 35     13   76          75.5
## 36    135   84          91.5
## 37     49   85          86.5
## 38     32   81          83.0
## 39     64   83          84.0
## 40     40   83          84.0
## 41     77   88          86.5
## 42     97   92          91.5
## 43     97   92          91.5
## 44     85   89          89.0
## 45     10   73          72.5
## 46     27   81          82.0
## 47      7   80          77.5
## 48     48   81          80.0
## 49     35   82          83.0
## 50     61   84          84.0
## 51     79   87          86.0
## 52     63   85          83.5
## 53     16   74          75.5
## 54     80   86          86.5
## 55    108   85          84.5
## 56     20   82          81.0
## 57     52   86          85.5
## 58     82   88          86.5
## 59     50   86          85.5
## 60     64   83          84.0
## 61     59   81          81.5
## 62     39   81          82.0
## 63      9   81          78.0
## 64     16   82          82.0
## 65     78   86          86.5
## 66     35   85          83.0
## 67     66   87          84.0
```

```
## 68    122   89          92.0
## 69     89   90          91.0
## 70    110   90          89.5
## 71     44   86          84.0
## 72     28   82          81.5
## 73     65   80          83.0
## 74     22   77          77.0
## 75     59   79          82.5
## 76     23   76          77.5
## 77     31   78          79.0
## 78     44   78          78.5
## 79     21   77          76.5
## 80      9   72          72.0
## 81     45   79          78.0
## 82    168   81          86.5
## 83     73   86          89.0
## 84     76   97          92.5
## 85    118   94          89.5
## 86     84   96          91.5
## 87     85   94          92.5
## 88     96   91          92.0
## 89     78   92          87.5
## 90     73   93          93.5
## 91     91   93          90.5
## 92     47   87          85.5
## 93     32   84          83.0
## 94     20   80          79.5
## 95     23   78          76.5
## 96     21   75          76.5
## 97     24   73          73.5
## 98     44   81          80.0
## 99     21   76          76.0
## 100    28   77          79.5
## 101     9   71          72.5
## 102    13   71          73.0
## 103    46   78          78.5
## 104    18   67          68.5
## 105    13   76          75.5
## 106    24   68          69.0
## 107    16   82          82.0
## 108    13   64          63.5
## 109    23   71          70.5
## 110    36   81          81.5
## 111     7   69          71.5
## 112    14   63          64.0
## 113    30   70          67.0
## 114    14   75          76.0
## 115    18   76          75.0
## 116    20   68          66.0
```

```
# plot Temp vs Ozone with kNN predictions, k = 2

ggplot(dfair2, aes(x = Ozone, y = Temp))+
  geom_point()+
```

```
  geom_point(aes(dfplot$knnregmod.pred, col = "orange"))+
  scale_colour_discrete(name='pred') +
  stat_smooth(method = "loess", col = "red")+
   stat_smooth(method = "lm", col = "yellow")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
# fit a kNN regression with k = 10

knnregmod2 = knn.reg(dfair2, y=dfair2$Temp, k=10)
dfplot2 = cbind(dfair2, data.frame(knnregmod2$pred))
dfplot2
```

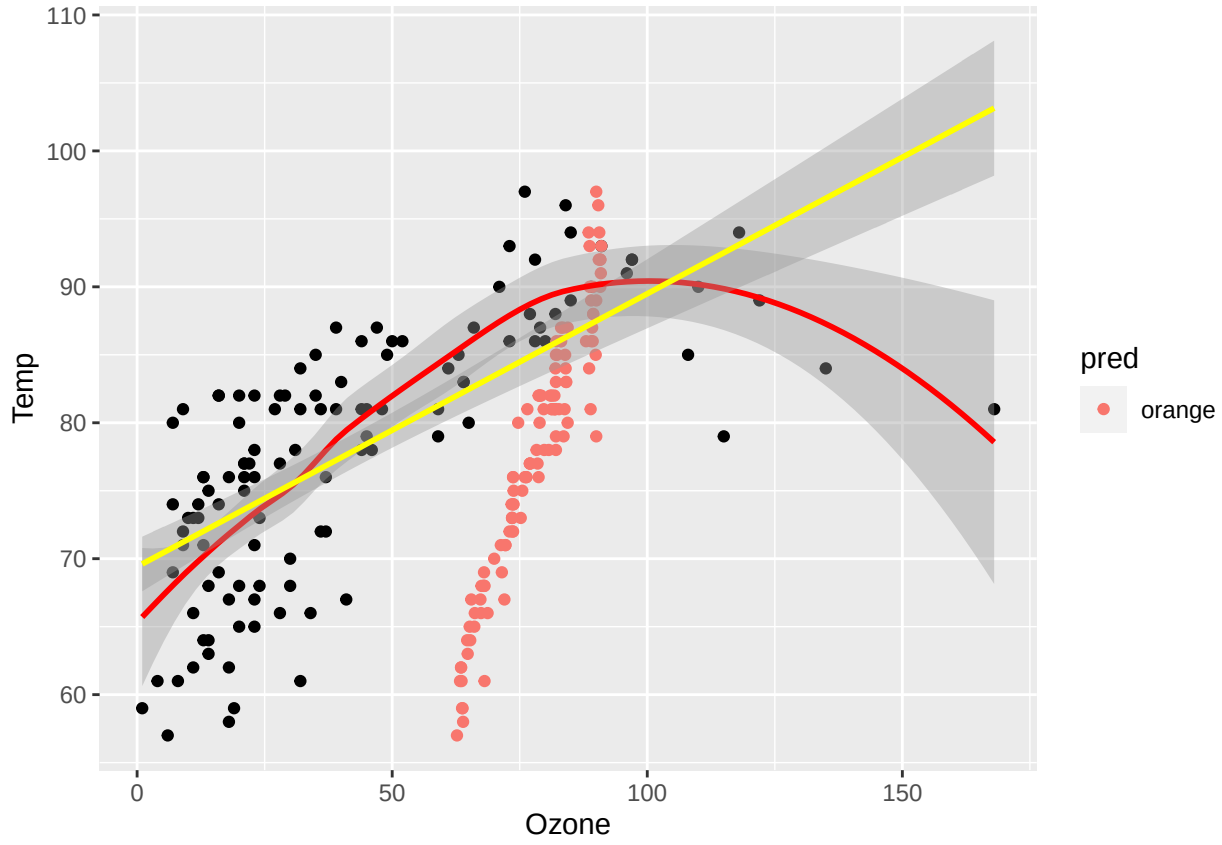
```
##     Ozone Temp knnregmod2.pred
## 1      41   67            72.0
## 2      36   72            73.6
## 3      12   74            73.4
## 4      18   62            63.5
## 5      28   66            67.4
## 6      23   65            66.1
## 7      19   59            63.7
## 8       8   61            63.3
## 9       7   74            73.7
```

```
## 10     16   69           68.0
## 11     11   66           66.2
## 12     14   68           68.1
## 13     18   58           63.9
## 14     14   64           64.7
## 15     34   66           68.7
## 16      6   57           62.7
## 17     30   68           67.9
## 18     11   62           63.5
## 19      1   59           63.8
## 20     11   73           73.5
## 21      4   61           63.6
## 22     32   61           68.1
## 23     23   67           67.3
## 24     45   81           81.9
## 25    115   79           90.0
## 26     37   76           78.7
## 27     29   82           81.3
## 28     71   90           88.7
## 29     39   87           83.1
## 30     23   82           79.2
## 31     21   77           77.0
## 32     37   72           73.7
## 33     20   65           65.2
## 34     12   73           73.5
## 35     13   76           73.7
## 36    135   84           88.6
## 37     49   85           82.3
## 38     32   81           81.3
## 39     64   83           84.1
## 40     40   83           82.1
## 41     77   88           89.4
## 42     97   92           90.8
## 43     97   92           90.8
## 44     85   89           90.0
## 45     10   73           73.5
## 46     27   81           79.7
## 47      7   80           74.7
## 48     48   81           82.7
## 49     35   82           81.8
## 50     61   84           84.0
## 51     79   87           89.2
## 52     63   85           83.9
## 53     16   74           73.8
## 54     80   86           89.0
## 55    108   85           89.9
## 56     20   82           78.7
## 57     52   86           83.1
## 58     82   88           89.4
## 59     50   86           82.2
## 60     64   83           84.1
## 61     59   81           83.8
## 62     39   81           81.3
## 63      9   81           76.5
```

```
## 64     16   82           78.9
## 65     78   86           89.3
## 66     35   85           82.1
## 67     66   87           84.4
## 68    122   89           89.0
## 69     89   90           90.8
## 70    110   90           89.4
## 71     44   86           83.1
## 72     28   82           81.0
## 73     65   80           84.4
## 74     22   77           77.0
## 75     59   79           83.6
## 76     23   76           76.0
## 77     31   78           79.8
## 78     44   78           80.7
## 79     21   77           77.0
## 80      9   72           73.0
## 81     45   79           82.1
## 82    168   81           88.9
## 83     73   86           88.0
## 84     76   97           90.0
## 85    118   94           88.5
## 86     84   96           90.4
## 87     85   94           90.6
## 88     96   91           90.9
## 89     78   92           90.5
## 90     73   93           88.7
## 91     91   93           91.0
## 92     47   87           83.5
## 93     32   84           82.0
## 94     20   80           78.9
## 95     23   78           78.3
## 96     21   75           75.5
## 97     24   73           75.2
## 98     44   81           81.9
## 99     21   76           76.3
## 100    28   77           78.5
## 101     9   71           71.3
## 102    13   71           72.2
## 103    46   78           82.1
## 104    18   67           65.5
## 105    13   76           73.7
## 106    24   68           68.0
## 107    16   82           78.9
## 108    13   64           65.3
## 109    23   71           72.2
## 110    36   81           81.9
## 111     7   69           71.5
## 112    14   63           64.8
## 113    30   70           70.0
## 114    14   75           73.8
## 115    18   76           76.3
## 116    20   68           67.5
```

```
# plot Temp vs Ozone with kNN predictions, k = 10

ggplot(dfair2, aes(x = Ozone, y = Temp))+
  geom_point()+
  geom_point(aes(dfplot2$knnregmod2.pred, col = 'orange'))+
  scale_colour_discrete(name='pred') +
  stat_smooth(method = "loess", col = "red")+
  stat_smooth(method = "lm", col = "yellow")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
# fit a kNN regression with k = 100

knnregmod3 = knn.reg(dfair2, y=dfair2$Temp, k=100)
dfplot3 = cbind(dfair2, data.frame(knnregmod3$pred))
dfplot3
```

```
##     Ozone Temp knnregmod3.pred
## 1      41   67           76.27
## 2      36   72           76.22
## 3      12   74           76.20
## 4      18   62           76.32
## 5      28   66           76.28
```

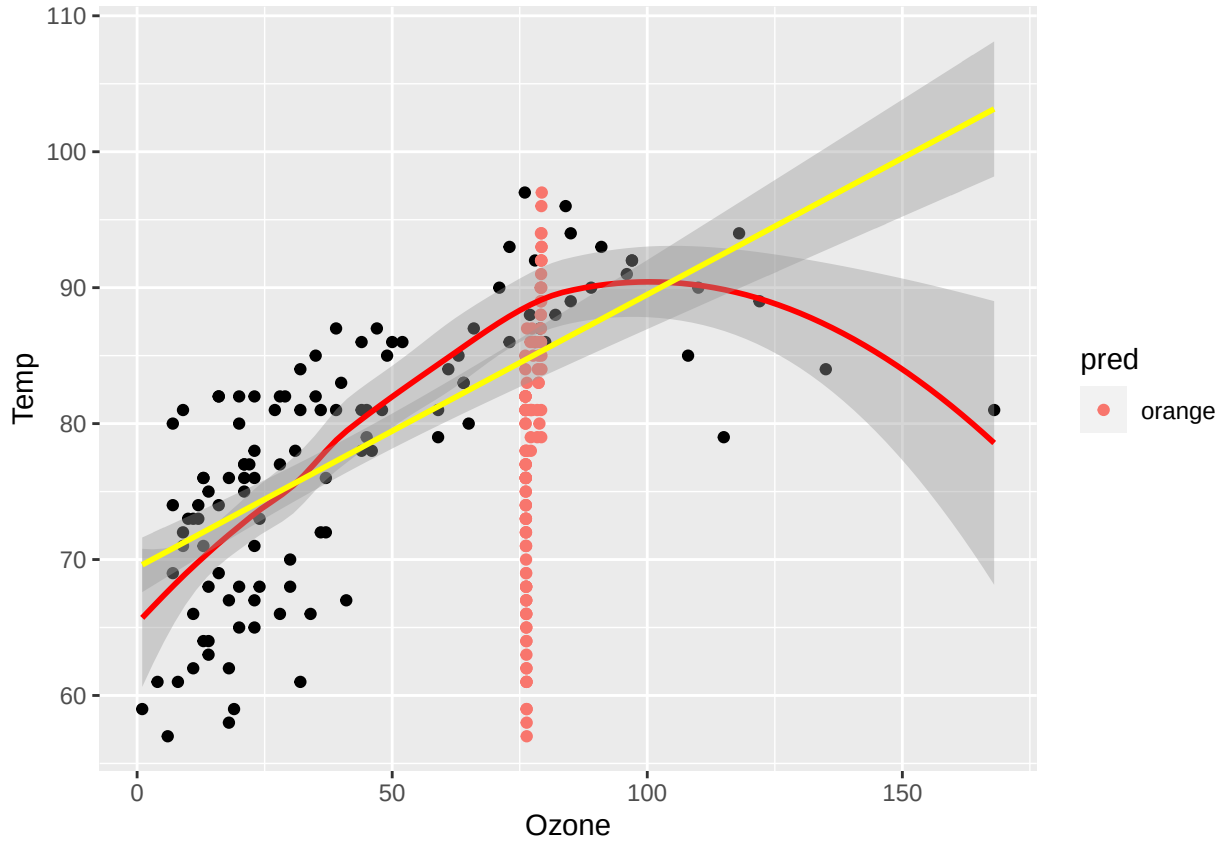
```
## 6      23   65          76.29
## 7      19   59          76.35
## 8       8   61          76.33
## 9       7   74          76.20
## 10     16   69          76.25
## 11     11   66          76.28
## 12     14   68          76.26
## 13     18   58          76.36
## 14     14   64          76.30
## 15     34   66          76.28
## 16      6   57          76.37
## 17     30   68          76.26
## 18     11   62          76.32
## 19      1   59          76.35
## 20     11   73          76.21
## 21      4   61          76.33
## 22     32   61          76.33
## 23     23   67          76.27
## 24     45   81          77.15
## 25    115   79          79.22
## 26     37   76          76.18
## 27     29   82          76.12
## 28     71   90          79.09
## 29     39   87          76.44
## 30     23   82          76.12
## 31     21   77          76.17
## 32     37   72          76.22
## 33     20   65          76.29
## 34     12   73          76.21
## 35     13   76          76.18
## 36    135   84          79.26
## 37     49   85          77.64
## 38     32   81          76.13
## 39     64   83          78.70
## 40     40   83          76.41
## 41     77   88          79.14
## 42     97   92          79.18
## 43     97   92          79.18
## 44     85   89          79.13
## 45     10   73          76.21
## 46     27   81          76.13
## 47      7   80          76.14
## 48     48   81          77.44
## 49     35   82          76.12
## 50     61   84          78.56
## 51     79   87          79.15
## 52     63   85          78.68
## 53     16   74          76.20
## 54     80   86          79.16
## 55    108   85          79.25
## 56     20   82          76.12
## 57     52   86          78.36
## 58     82   88          79.14
## 59     50   86          77.63
```

```
## 60     64   83          78.70
## 61     59   81          78.41
## 62     39   81          76.13
## 63      9   81          76.13
## 64     16   82          76.12
## 65     78   86          79.16
## 66     35   85          76.09
## 67     66   87          79.12
## 68    122   89          79.21
## 69     89   90          79.20
## 70    110   90          79.20
## 71     44   86          77.10
## 72     28   82          76.12
## 73     65   80          78.86
## 74     22   77          76.17
## 75     59   79          78.43
## 76     23   76          76.18
## 77     31   78          76.16
## 78     44   78          76.46
## 79     21   77          76.17
## 80      9   72          76.22
## 81     45   79          77.17
## 82    168   81          79.20
## 83     73   86          79.16
## 84     76   97          79.32
## 85    118   94          79.16
## 86     84   96          79.24
## 87     85   94          79.26
## 88     96   91          79.19
## 89     78   92          79.28
## 90     73   93          79.27
## 91     91   93          79.27
## 92     47   87          77.38
## 93     32   84          76.10
## 94     20   80          76.14
## 95     23   78          76.16
## 96     21   75          76.19
## 97     24   73          76.21
## 98     44   81          77.15
## 99     21   76          76.18
## 100    28   77          76.17
## 101     9   71          76.23
## 102    13   71          76.23
## 103    46   78          77.18
## 104    18   67          76.27
## 105    13   76          76.18
## 106    24   68          76.26
## 107    16   82          76.12
## 108    13   64          76.30
## 109    23   71          76.23
## 110    36   81          76.13
## 111     7   69          76.25
## 112    14   63          76.31
## 113    30   70          76.24
```

```
## 114    14   75          76.19
## 115    18   76          76.18
## 116    20   68          76.26
```

```
# plot Temp vs Ozone with kNN predictions, k = 100

ggplot(dfair2, aes(x = Ozone, y = Temp))+
  geom_point()+
  geom_point(aes(dfplot3$knnregmod3.pred, col = 'orange'))+
  scale_colour_discrete(name='pred') +
  stat_smooth(method = "loess", col = "red") +
  stat_smooth(method = "lm", col = "yellow")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



As k increases, or number of neighbors, predictions become less staggered. (The predictions observed actually align with the bend in the polynomial regression with higher k values). In this situation, the lower the k value, the better (k<2). Nearest neighbors in this situation has a max value of 116, or nrows.

kNN Predictions will be more accurate with more data points. Analysis of kNN regression vs. linear regression would be more appropriate then.