Recode

Recoding all values above 5 to '1' and to '0' if otherwise:

```
iris$dummy <- as.numeric(iris$Sepal.Length>5)
```

Here is a glimpse of the new df:

```
> iris$dummy <- as.numeric(iris$Sepal.Length>5)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species dummy
1          5.1         3.5          1.4         0.2  setosa     1
2          4.9         3.0          1.4         0.2  setosa     0
3          4.7         3.2          1.3         0.2  setosa     0
4          4.6         3.1          1.5         0.2  setosa     0
5          5.0         3.6          1.4         0.2  setosa     0
6          5.4         3.9          1.7         0.4  setosa     1
```

Here is another way to do this:

```
iris$dummy <- ifelse(iris$Sepal.Length>=5, 1, 0)
```

After that we can label the 0's and 1's like this:

```
iris$dummy <- factor(iris$dummy,
                     levels = c(0,1),
                     labels = c("Short", "Long"))
```

And check the levels:

```
levels(iris$dummy)

[1] "Short" "Long"
```

Or using the cut function:

```
iris$new<-cut(iris$Sepal.Length,
    breaks=c(-Inf, 5.5, 6.5, 7.5, Inf),
    labels=c("lowest", "lower","middle","high"))

> iris$new<-cut(iris$Sepal.Length,
+               breaks=c(-Inf, 5.5, 6.5, 7.5, Inf),
+               labels=c("lowest", "lower","middle","high"))
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species    new
1          5.1         3.5          1.4         0.2  setosa lowest
2          4.9         3.0          1.4         0.2  setosa lowest
3          4.7         3.2          1.3         0.2  setosa lowest
4          4.6         3.1          1.5         0.2  setosa lowest
5          5.0         3.6          1.4         0.2  setosa lowest
6          5.4         3.9          1.7         0.4  setosa lowest
```

When working with cont. variables e.g. BMI or income, a common desire is to get a column of quantiles out of the continuous one, which can be done easily using the OneR package:

```
iris$Quintiles <- bin(iris$Sepal.Length, method = "content")
```

Rename

With plyr:

```
> iris <- rename(iris, c("sw"="Sepal.Width" ))
> names(iris)
[1] "Sepal.Length" "sw"           "Petal.Length" "Petal.Width"  "Species"
```

With dplyr:

```
 iris <- iris %>%
+     rename(sw=Sepal.Width,
+            pw=Petal.Length,
+            pl= Petal.Length)
> names(iris)
[1] "Sepal.Length" "sw"           "pl"           "Petal.Width"  "Species"

>
```

With data.table

```
setnames(df, "oldName", "newName")

e.g.

setnames(DF, "Petal.Width", "pw")
```

With base R

```
colnames(df)[colnames(df) == 'oldName'] <- 'newName'
```