Lab 9

Maria Barouti

3/20/2020

It is rarely the case that a dataset will have a single predictor variable. It is also rarely the case that a response variable will only depend on a single variable. So in this lab, we will extend our current linear model to allow a response to depend on *multiple* predictors. For this Lab we will discuss a dataset with information about cars. This dataset, which can be found at the UCI Machine Learning Repository contains a response variable `mpg` which stores the city fuel efficiency of cars, as well as several predictor variables for the attributes of the vehicles.

```r
# read the data from the web
autompg = read.table(
  "http://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data",
  quote = "\"",
  comment.char = "",
  stringsAsFactors = FALSE)
# give the dataframe headers
colnames(autompg) = c("mpg", "cyl", "disp", "hp", "wt", "acc", "year", "origin", "name")
# remove missing data, which is stored as "?"
autompg = subset(autompg, autompg$hp != "?")
# remove the plymouth reliant, as it causes some issues
autompg = subset(autompg, autompg$name != "plymouth reliant")
# give the dataset row names, based on the engine, year and name
rownames(autompg) = paste(autompg$cyl, "cylinder", autompg$year, autompg$name)
# remove the variable for name, as well as origin
autompg = subset(autompg, select = c("mpg", "cyl", "disp", "hp", "wt", "acc", "year"))
# change horsepower from character to numeric
autompg$hp = as.numeric(autompg$hp)
# check final structure of data
str(autompg)
```

```
## 'data.frame':    390 obs. of  7 variables:
##  $ mpg : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cyl : int  8 8 8 8 8 8 8 8 8 8 ...
##  $ disp: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ hp  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ wt  : num  3504 3693 3436 3433 3449 ...
##  $ acc : num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year: int  70 70 70 70 70 70 70 70 70 70 ...
```

We will focus on using two variables, `wt` and `year`, as predictor variables. That is, we would like to model the fuel efficiency (`mpg`) of a car as a function of its weight (`wt`) and model year (`year`). To do so, we will define the following linear model,

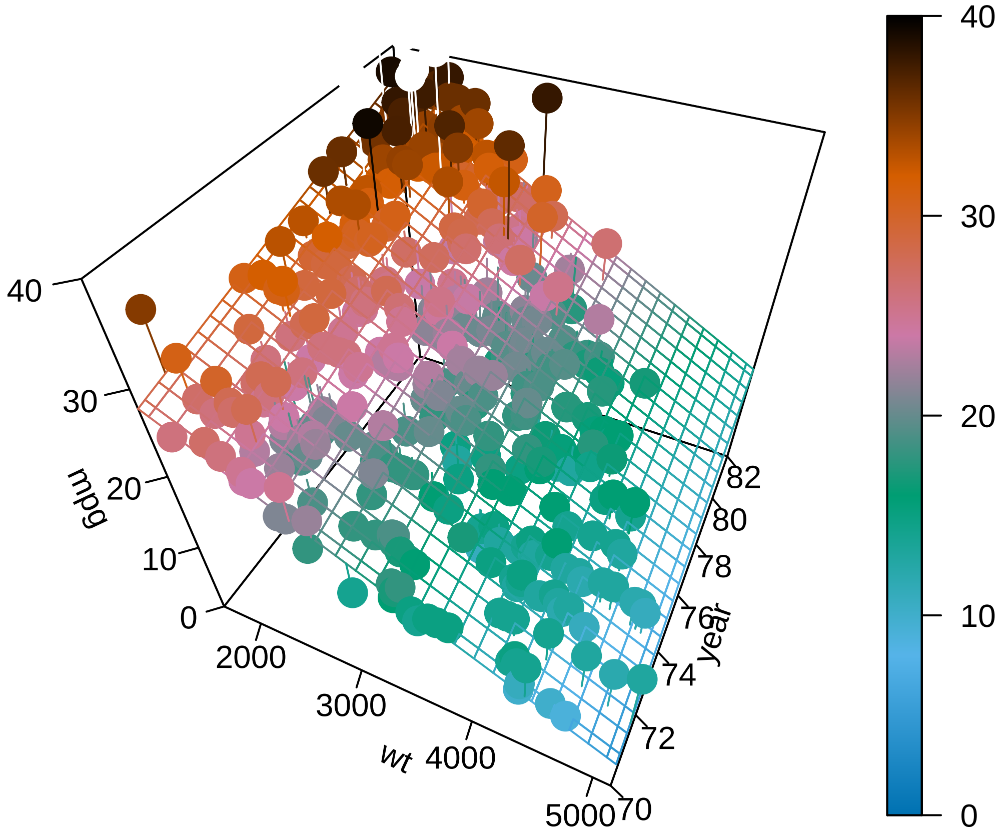
$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i, \qquad i = 1, 2, \ldots, n$$

where $\epsilon_i \sim N(0, \sigma^2)$. In this notation we will define:

- X_{i1} as the weight (`wt`) of the ith car.
- X_{i2} as the model year (`year`) of the ith car.

The picture below will visualize what we would like to accomplish. The data points (X_{i1}, X_{i2}, Y_i) now exist in 3-dimensional space, so instead of fitting a line to the data, we will fit a plane.

```r
library("plot3D")

x = autompg$wt
y = autompg$year
z = autompg$mpg

fit <- lm(z ~ x + y)

grid.lines = 25
x.pred     = seq(min(x), max(x), length.out = grid.lines)
y.pred     = seq(min(y), max(y), length.out = grid.lines)
xy         = expand.grid(x = x.pred, y = y.pred)

z.pred = matrix(predict(fit, newdata = xy),
                nrow = grid.lines, ncol = grid.lines)

fitpoints = predict(fit)

scatter3D(x, y, z, pch = 19, cex = 2, col = gg.col(1000), lighting = TRUE,
          theta = 25, phi = 45, ticktype = "detailed",
          xlab = "wt", ylab = "year", zlab = "mpg", zlim = c(0, 40), clim = c(0, 40),
          surf = list(x = x.pred, y = y.pred, z = z.pred,
                      facets = NA, fit = fitpoints), main = "")
```



Task 1

Use the `lm` function and provide estimates for b_0, b_1, b_2.

```
mpg_model = lm(mpg ~ wt + year, data = autompg)
coef(mpg_model)
```

```
##   (Intercept)            wt          year 
## -14.637641945  -0.006634876   0.761401955
```

Task 2

Obtain estiates for b_0, b_1, b_2 using

$$\mathbf{b} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{Y})$$

Do estimates in Task 2 agree with the estimates obtained from Task 1?

```
n = nrow(autompg)
p = length(coef(mpg_model))
X = cbind(rep(1, n), autompg$wt, autompg$year)
y = autompg$mpg

(beta_hat = solve(t(X) %*% X) %*% t(X) %*% y)
```

```
##               [,1]
## [1,] -14.637641945
## [2,]  -0.006634876
## [3,]   0.761401955
```

Task 3

Often we will be interested in the square root of MSE which is given by

$$RMSE = \sqrt{\frac{SSE}{n-p}}.$$

Calculate $RMSE$ using `residuals` function from `R` as well as using the vector notation of the residuals.

```
MSE <- sum(residuals(mpg_model)^2)/(n-p)
sqrt(MSE)
```

```
## [1] 3.431367
```

And we can now verify that our math above is indeed calculating the same quantities.

```
y_hat = X %*% solve(t(X) %*% X) %*% t(X) %*% y
e     = y - y_hat
sqrt(t(e) %*% e / (n - p))
```

```
##          [,1]
## [1,] 3.431367
```

Task 4

Calculate R^2. How do you interpret the result?

```
summary(mpg_model)$r.squared
```

```
## [1] 0.8082355
```

The interpretation changes slightly as compared to simple linear regression. Here, we say that ‘$rround(100 \ast summary(mpg_{m}odel)$r.squared, 2)‘%$ for the observed variation in miles per gallon is explained by the linear relationship with the two predictor variables, weight and year.