EDA and Regression for Lap Time, Power and Points

Mario Grugan

2025-04-20

```
lap_time <- c(85.093, 83.299, 83.328, 88.362, 78.062, 92.995, 81.901, 78.84, 84.682, 73.574,
              78.291, 75.55, 75.969, 71.951, 73.189, 83.012, 75.641, 80.752, 102.347, 71.875,
              100.353, 80.426, 94.253, 73.448, 74.508, 71.992, 82.447, 76.535, 86.92, 67.667,
              75.563, 74.089, 73.467, 93.004, 89.961)

energy <- c(0.0723, 0.0813, 0.0896, 0.0858, 0.1105, 0.0957, 0.1115, 0.1182, 0.1103, 0.1271,
            0.127, 0.1342, 0.1403, 0.1555, 0.1542, 0.1364, 0.1507, 0.142, 0.114, 0.1704,
            0.1233, 0.0518, 0.0679, 0.1065, 0.106, 0.1233, 0.1136, 0.1251, 0.1105, 0.1432,
            0.1347, 0.1385, 0.1448, 0.122, 0.1604)

eff_score <- c(100, 87.1, 75.1, 73.3, 59.4, 56.3, 53.7, 51.8, 51.6, 51.5,
               45.9, 44.3, 40.1, 36.2, 35.5, 35.2, 34.7, 34.3, 33, 29.4,
               28.7, 100, 95.9, 93.3, 93.2, 91.4, 90.5, 90.1, 90.1, 89.9,
               89, 88.9, 88.2, 86.9, 81.3)

endurance_time <- c(1581.3, 1582.9, 1609.3, 1610.2, 1618.6, 1630.8, 1662.1, 1664.1, 1671.3, 1717.4,
                    1722.4, 1734.5, 1776.5, 1801.8, 1832.6, 1833.2, 1863.0, 1872.1, 1944.0, 2045.9,
                    2207.8, 2251.6)

endurance_score <- c(250, 249.2, 236, 235.5, 231.4, 225.5, 210.8, 209.9, 206.6, 186.2,
                     184, 178.8, 161.5, 151.4, 139.5, 139.3, 128.2, 124.9, 99.7, 67.1,
                     21.4, 10.2)

eff_model <- lm(eff_score ~ lap_time + energy)
end_model <- lm(endurance_score ~ endurance_time)


summary(end_model)
```

```
##
## Call:
## lm(formula = endurance_score ~ endurance_time)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.494 -7.261 -1.450  6.257 15.373
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    816.825656  15.948179   51.22   <2e-16 ***
## endurance_time  -0.365073   0.008895  -41.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 7.741 on 20 degrees of freedom
## Multiple R-squared:  0.9883, Adjusted R-squared:  0.9877
## F-statistic:  1685 on 1 and 20 DF,  p-value: < 2.2e-16
```

```
summary(eff_model)
```

```
##
## Call:
## lm(formula = eff_score ~ lap_time + energy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.369 -18.008  -7.596  19.816  48.073
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  219.9657    48.1515   4.568 6.94e-05 ***
## lap_time      -1.0134     0.4699  -2.157 0.038636 *
## energy      -595.8162   144.7107  -4.117 0.000252 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.91 on 32 degrees of freedom
## Multiple R-squared:  0.3491, Adjusted R-squared:  0.3084
## F-statistic: 8.582 on 2 and 32 DF,  p-value: 0.001038
```

```
predict_total_points <- function(lap, energy_used) {
  total_time <- lap * 22
  eff <- predict(eff_model, newdata = data.frame(lap_time = lap, energy = energy_used))
  end <- predict(end_model, newdata = data.frame(endurance_time = total_time))
  total <- eff + end
  list(efficiency = eff, endurance = end, total_points = total)
}

objective <- function(x) {
  lap <- x[1]
  energy_used <- x[2]
  -predict_total_points(lap, energy_used)$total_points
}

opt <- optim(par = c(75, 0.1), fn = objective,
             method = "L-BFGS-B",
             lower = c(65, 0.05),
             upper = c(100, 0.18))

opt_lap <- opt$par[1]
opt_energy <- opt$par[2]
opt_result <- predict_total_points(opt_lap, opt_energy)
opt_time <- opt_lap * 22

cat("Optimal Lap Time:", round(opt_lap, 2), "\n")
```

```
## Optimal Lap Time: 65
```

```r
cat("Optimal Energy per Lap:", round(opt_energy, 4), "\n")
```

```
## Optimal Energy per Lap: 0.05
```

```r
cat("Optimal Endurance Time:", round(opt_time, 2), "\n")
```

```
## Optimal Endurance Time: 1430
```

```r
cat("Efficiency Score:", round(opt_result$efficiency, 2), "\n")
```

```
## Efficiency Score: 124.3
```

```r
cat("Endurance Score:", round(opt_result$endurance, 2), "\n")
```

```
## Endurance Score: 294.77
```

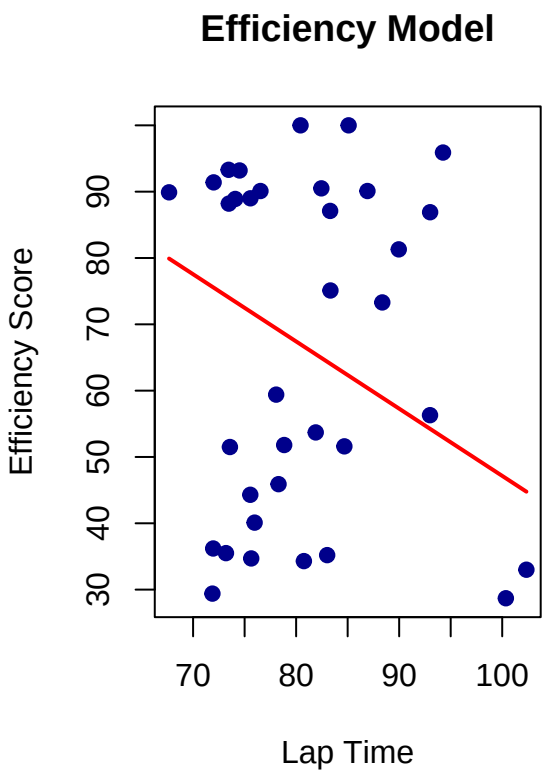
```r
cat("Total Points:", round(opt_result$total_points, 2), "\n")
```

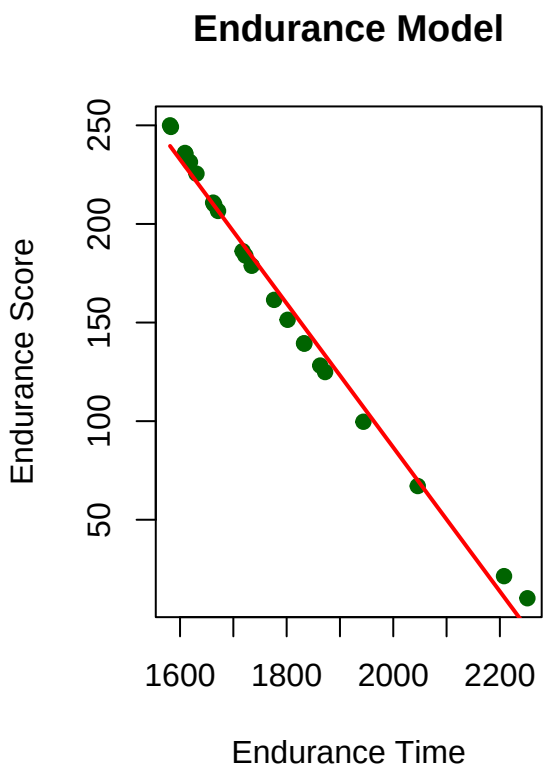
```
## Total Points: 419.07
```

```r
par(mfrow = c(1, 2))

plot(lap_time, eff_score, pch = 19, col = "darkblue",
     xlab = "Lap Time", ylab = "Efficiency Score", main = "Efficiency Model")
grid_lap <- seq(min(lap_time), max(lap_time), length.out = 100)
grid_eff <- predict(eff_model, newdata = data.frame(
  lap_time = grid_lap,
  energy = mean(energy)
))
lines(grid_lap, grid_eff, col = "red", lwd = 2)

plot(endurance_time, endurance_score, pch = 19, col = "darkgreen",
     xlab = "Endurance Time", ylab = "Endurance Score", main = "Endurance Model")
grid_end <- seq(min(endurance_time), max(endurance_time), length.out = 100)
grid_score <- predict(end_model, newdata = data.frame(endurance_time = grid_end))
lines(grid_end, grid_score, col = "red", lwd = 2)
```



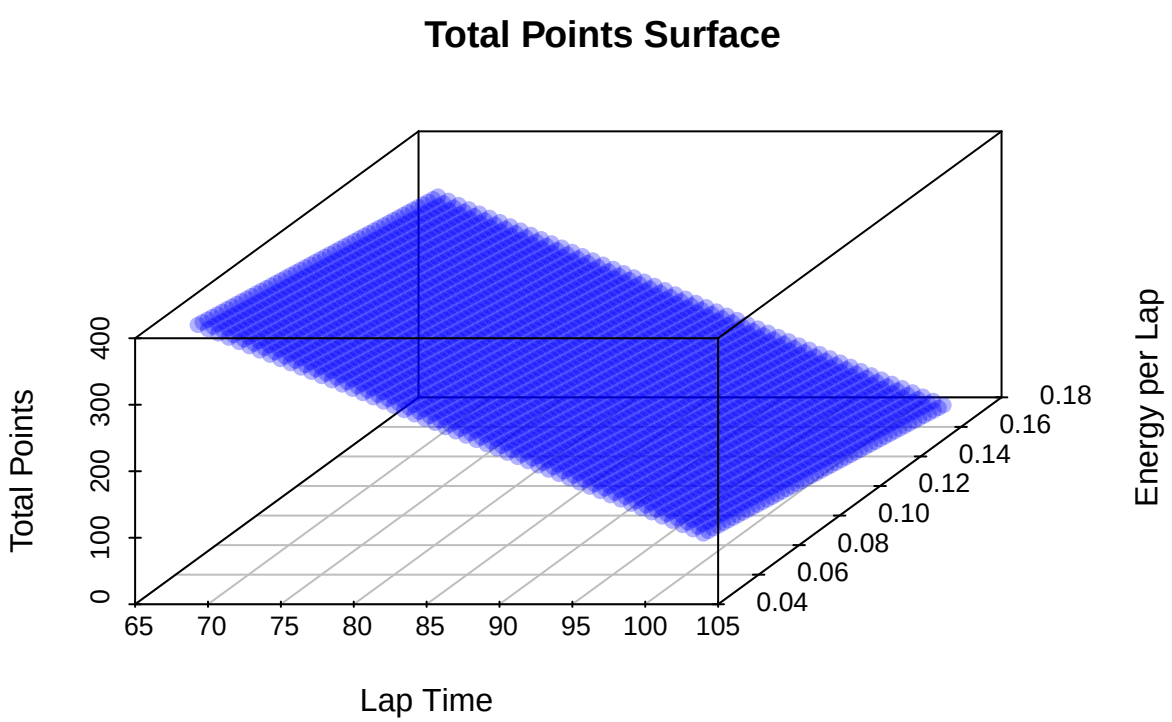



```r
library(scatterplot3d)

grid_lap <- seq(min(lap_time), max(lap_time), length.out = 50)
grid_energy <- seq(min(energy), max(energy), length.out = 50)
grid <- expand.grid(lap_time = grid_lap, energy = grid_energy)
grid$total_time <- grid$lap_time * 22
grid$eff <- predict(eff_model, newdata = grid)
grid$end <- predict(end_model, newdata = data.frame(endurance_time = grid$total_time))
grid$total_points <- grid$eff + grid$end

color_transparent <- rgb(0, 0, 1, alpha = 0.3)

scatterplot3d(grid$lap_time, grid$energy, grid$total_points,
              pch = 19, color = color_transparent,
              xlab = "Lap Time", ylab = "Energy per Lap", zlab = "Total Points",
              main = "Total Points Surface")
```


Total Points Surface
Total Points
0
100
200
300
400
65
70
75
80
85
90
95
100
105
Lap Time
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
Energy per Lap