Stat 291 - Recitation 11

Orçun Oltulu

07 / 01 / 2022

Simple Linear Regression

Exercise 1:

Part A:

Import 'Auto' data set from 'ISLR' package.

```
library(ISLR)
data(Auto)
```

Part B:

Check the structure of the variables, and convert 'cylinders' and 'origin' to factors, and drop 'year' and 'name' variables.

```
str(Auto)
```

```
## 'data.frame':    392 obs. of  9 variables:
##  $ mpg         : num  18 15 18 16 17 15 14 14 14 15 ...
##  $ cylinders   : num  8 8 8 8 8 8 8 8 8 8 ...
##  $ displacement: num  307 350 318 304 302 429 454 440 455 390 ...
##  $ horsepower  : num  130 165 150 150 140 198 220 215 225 190 ...
##  $ weight      : num  3504 3693 3436 3433 3449 ...
##  $ acceleration: num  12 11.5 11 12 10.5 10 9 8.5 10 8.5 ...
##  $ year        : num  70 70 70 70 70 70 70 70 70 70 ...
##  $ origin      : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ name        : Factor w/ 304 levels "amc ambassador brougham",..: 49 36 231 14 161
```

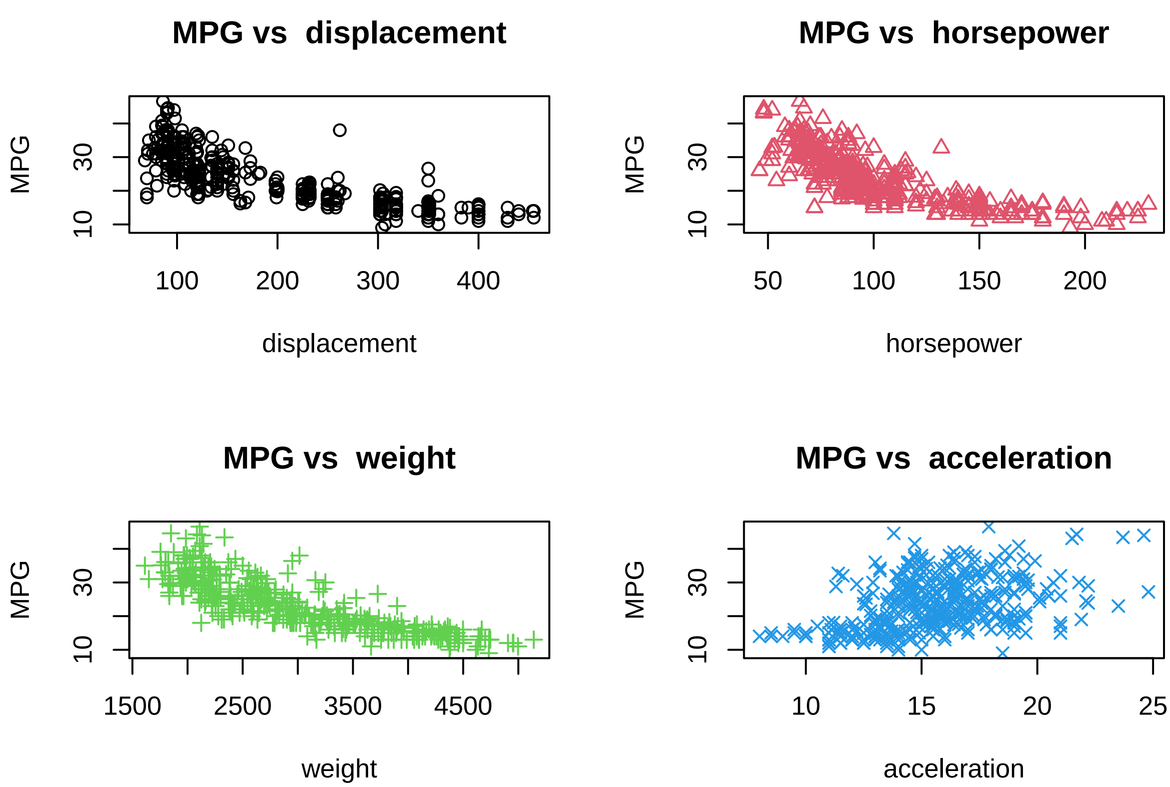
```
Auto$cylinders <- factor(Auto$cylinders)
Auto$origin <- factor(Auto$origin)
Auto$name <- NULL
Auto$year <- NULL
```

Part C:

By using only the numeric variables, construct scatter plots for MPG vs X. Use par() function to plot them at the same time. Comment on your findings.

```
numeric_variables <- names(Auto)[sapply(Auto,is.numeric)]

par(mfrow = c(2,2))
for(i in 1:4){
  variable_index <- setdiff(numeric_variables,"mpg")[i]
  plot(Auto$mpg~Auto[,variable_index],
       main = paste("MPG vs ", variable_index),
       xlab = variable_index, ylab = "MPG",
       col = i, pch = i)
}
```



Part D:

Obtain a correlation matrix to see the correlation between variables. Comment on your findings.

```
cor(Auto[,numeric_variables])
```

```
##                     mpg displacement horsepower     weight acceleration
## mpg           1.0000000   -0.8051269 -0.7784268 -0.8322442    0.4233285
## displacement -0.8051269    1.0000000  0.8972570  0.9329944   -0.5438005
## horsepower   -0.7784268    0.8972570  1.0000000  0.8645377   -0.6891955
## weight       -0.8322442    0.9329944  0.8645377  1.0000000   -0.4168392
## acceleration  0.4233285   -0.5438005 -0.6891955 -0.4168392    1.0000000
```

Part E:

Fit a Linear Model to estimate 'mpg', using only 'weight' variable as an explanatory variable. Write down the estimated regression model and comment on it.

```
fit1 <- lm(mpg ~ weight, data = Auto)
fit1
```

```
##
## Call:
## lm(formula = mpg ~ weight, data = Auto)
##
## Coefficients:
## (Intercept)       weight
##   46.216525    -0.007647
```

Part F:

Use summary function to get more information about your regression model. Comment on this output.

```
summary(fit1)
```

```
##
## Call:
## lm(formula = mpg ~ weight, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9736  -2.7556  -0.3358   2.1379  16.5194
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.216524   0.798673   57.87   <2e-16 ***
## weight      -0.007647   0.000258  -29.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.333 on 390 degrees of freedom
```

```
## Multiple R-squared:  0.6926, Adjusted R-squared:  0.6918
## F-statistic: 878.8 on 1 and 390 DF,  p-value: < 2.2e-16
```

Part G:

Construct a 95% Confidence Interval for β coefficients for your model.

```
confint(fit1, level = 0.95)
```

```
##                    2.5 %      97.5 %
## (Intercept) 44.646282308 47.78676679
## weight      -0.008154515 -0.00714017
```

Part H:

Now assume that you want to buy a brand new car. When you go to the dealer the salesman suggests you to buy 2 different cars. One of them (Car A) has 17 mpg value and the other one (Car B) has 15. This information that salesman gave you immediately raises a doubt and you wanted to use your model.

You know that Car A weighs 2513 lb and Car B weighs 3120 lb. According to your model, what are the predicted MPG values for these cars?

Also, find prediction interval and confidence interval for these cars.

Remark The prediction interval predicts in what range a future individual observation will fall, while a confidence interval shows the likely range of values associated with some statistical parameter of the data, such as the population mean.

```
newcars <- data.frame(weight = c(2513,3120))
predict(fit1, newdata = newcars,
        type = "response",level = 0.95)
```

```
##        1        2
## 26.99875 22.35682
```

```
pi_newcars <- predict(fit1, newdata = newcars,
                      interval = "prediction",level = 0.95)
ci_newcars <- predict(fit1, newdata = newcars,
                      interval = "confidence",level = 0.95)
```

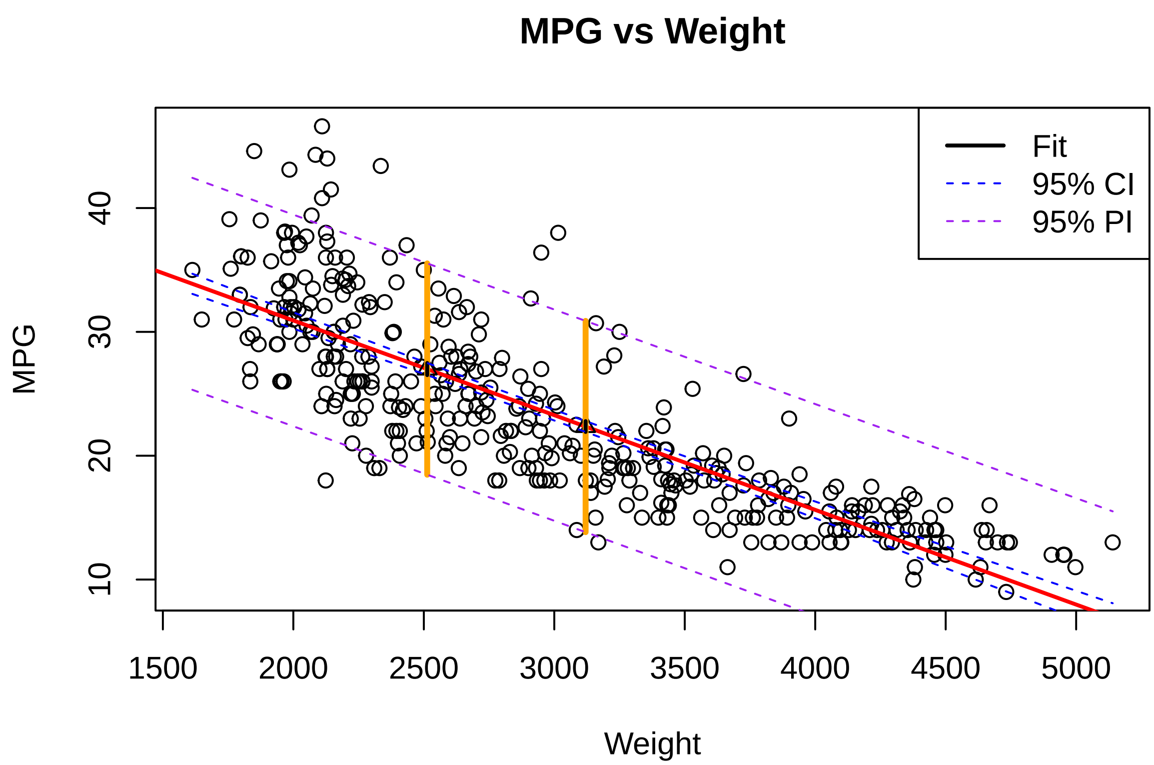
Part I:

Construct a Scatter-Plot 'mpg' vs 'weight', draw the regression line, also add Confidence interval and prediction interval for regression model.

```
x <- data.frame(weight = seq(min(Auto$weight),max(Auto$weight),length = 100))
ci.band <- predict(fit1, newdata = x,
                   interval="confidence",level=0.95)
```

```
pi.band <- predict(fit1, newdata = x,
                   interval="prediction",level=0.95)
```

```
plot(Auto$weight, Auto$mpg,
     xlim = c(min(Auto$weight), max(Auto$weight)),
     ylim = c(min(Auto$mpg), max(Auto$mpg)),
     main = "MPG vs Weight", xlab = "Weight", ylab = "MPG")
abline(fit1,lwd=2,col = "Red")
points(newcars[,1],ci_newcars[,1],pch=2)
segments(x0=c(2513,3120),y0=c(pi_newcars[1,2],pi_newcars[2,2]),
         x1=c(2513,3120),y1=c(pi_newcars[1,3],pi_newcars[2,3]),col="orange",lwd=3)
segments(x0=c(2513,3120),y0=c(ci_newcars[1,2],ci_newcars[2,2]),
         x1=c(2513,3120),y1=c(ci_newcars[1,3],ci_newcars[2,3]),lwd=2)
lines(x[,1], ci.band[,2], lty=2, col="blue")
lines(x[,1], ci.band[,3], lty=2, col="blue")
lines(x[,1], pi.band[,2], lty=2, col="purple")
lines(x[,1], pi.band[,3], lty=2, col="purple")
legend("topright",legend=c("Fit","95% CI","95% PI"),lty=c(1,2,2),
       col=c("black","blue","purple"),lwd=c(2,1,1))
```



Linear Regression with Categorical Predictors:

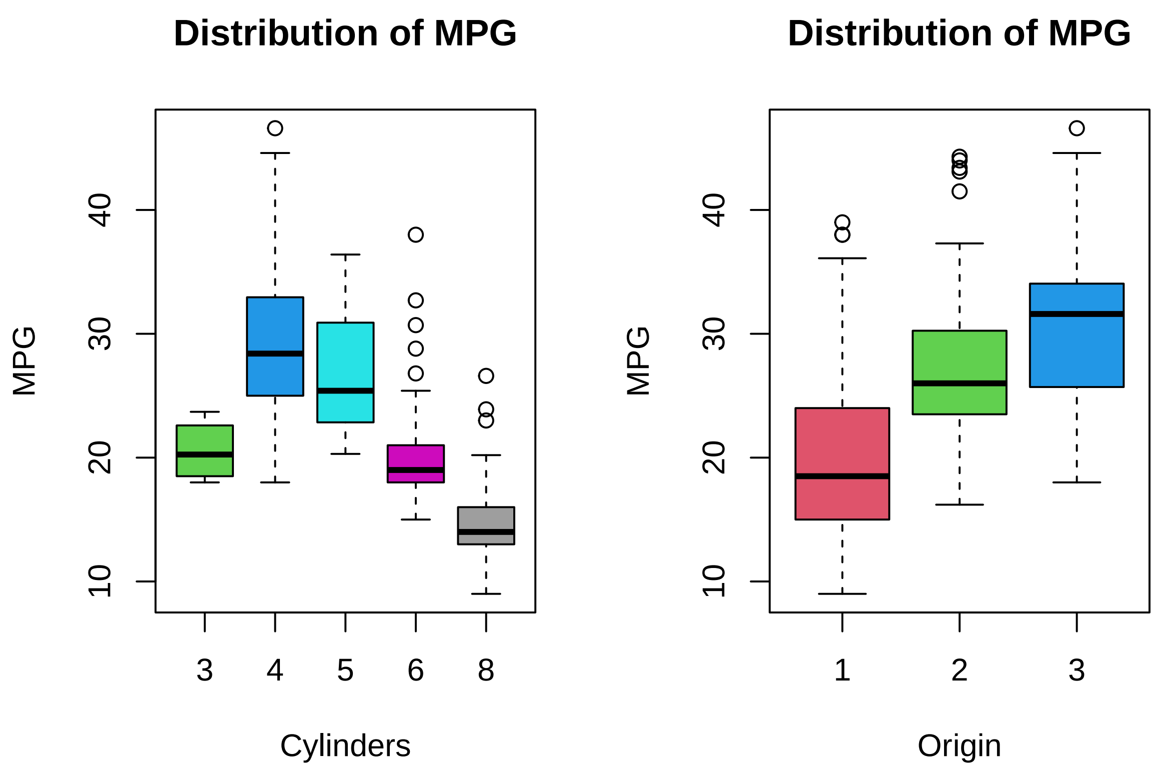
Exercise 2:

Again use the same Auto data set for this exercise.

Part A:

Construct Box-Plots for both 'mpg' vs 'origin' and 'mpg' vs 'cylinders' at the same time using par() function.

```
par(mfrow=c(1,2))
boxplot(Auto$mpg ~ Auto$cylinders,
        col = levels(Auto$cylinders),
        main = "Distribution of MPG",
        xlab = "Cylinders",
        ylab = "MPG")
boxplot(Auto$mpg ~ Auto$origin,
        col = 2:4,
        main = "Distribution of MPG",
        xlab = "Origin",
        ylab = "MPG")
```



Part B:

Construct a model where 'Cylinders' is an explanatory variable. Write down the estimated regression model and comment on it.

```r
fit2 <- lm(mpg ~ cylinders, data = Auto)
fit2
```

```
##
## Call:
## lm(formula = mpg ~ cylinders, data = Auto)
##
## Coefficients:
## (Intercept)   cylinders4   cylinders5   cylinders6   cylinders8
##     20.5500       8.7339       6.8167      -0.5765      -5.5869
```

Part C:

Use summary function to get more information about your regression model. Comment on this output.

```r
summary(fit2)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2839  -2.9037  -0.9631   2.3437  18.0265
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  20.5500     2.3494   8.747  < 2e-16 ***
## cylinders4    8.7339     2.3729   3.681 0.000266 ***
## cylinders5    6.8167     3.5888   1.899 0.058250 .
## cylinders6   -0.5765     2.4053  -0.240 0.810708
## cylinders8   -5.5869     2.3946  -2.333 0.020153 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.699 on 387 degrees of freedom
## Multiple R-squared:  0.6413, Adjusted R-squared:  0.6376
## F-statistic:   173 on 4 and 387 DF,  p-value: < 2.2e-16
```

Part D:

Using your estimated model in part c, make predictions for each level of 'cylinders' variable.

```
newcylinders <- data.frame(cylinders = factor(c(3,4,5,6,8)))
predict(fit2, newdata = newcylinders,
        type = "response",level=0.95)
```

```
##        1        2        3        4        5
## 20.55000 29.28392 27.36667 19.97349 14.96311
```

Part F:

Obtain prediction and confidence intervals for each level of 'cylinders' variable.

```
predict(fit2,newdata = newcylinders,
        interval = "predict",level=0.95)
```

```
##        fit       lwr      upr
## 1 20.55000 10.221175 30.87883
## 2 29.28392 20.022354 38.54548
## 3 27.36667 16.699102 38.03423
## 4 19.97349 10.679625 29.26736
## 5 14.96311  5.679986 24.24623
```

```
predict(fit2,newdata = newcylinders,
        interval = "confidence",level=0.95)
```

```
##        fit      lwr      upr
## 1 20.55000 15.93081 25.16919
## 2 29.28392 28.62903 29.93881
## 3 27.36667 22.03288 32.70045
## 4 19.97349 18.95945 20.98754
## 5 14.96311 14.05282 15.87339
```

Exercise 3:

Conduct every step in the second exercise for 'origin' variable and comment on each step.