Multiple Linear Regression

For this tutorial, we will go over the delivery time example from the textbook. The textbook describes the data as:

A soft drink bottler is analyzing the vending machine service routes in his distribution system. The industrial engineer responsible for the study has suggested that the two most important variables affecting the delivery time (y) are the number of cases of product stocked (x_1) and the distance walked by the route driver (x_2).

Download the data file, `delivery.txt`, from Collab and read the data in.

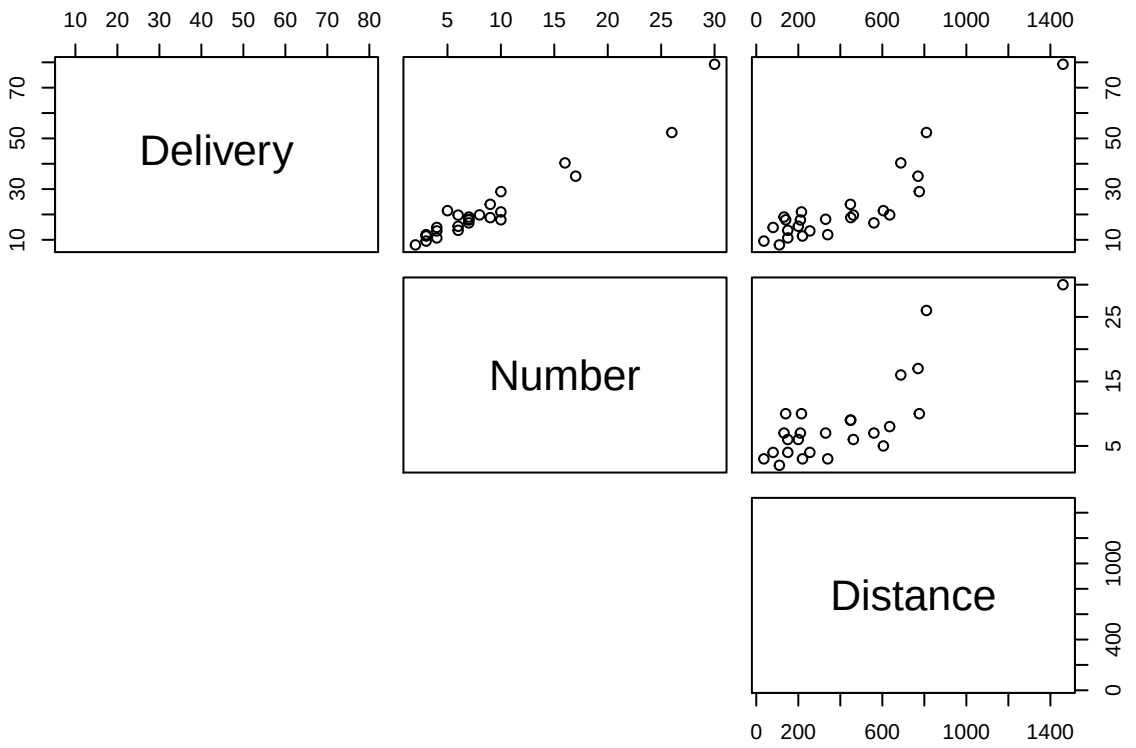
```
Data<-read.table("delivery.txt", header=TRUE)
head(Data)
```

```
##   Delivery Number Distance
## 1    16.68      7      560
## 2    11.50      3      220
## 3    12.03      3      340
## 4    14.88      4       80
## 5    13.75      6      150
## 6    18.11      7      330
```

1. Scatterplot matrix

A scatterplot matrix is useful to create scatterplots involving more than two quantitative variables, via the `pairs()` function

```
pairs(Data, lower.panel = NULL)
```



A data frame containing all the quantitative variables is supplied. The argument `lower.panel = NULL` is optional but I think it makes the scatterplot matrix look less cluttered.

Based on the scatterplot matrix, we can see that delivery time has a positive linear association with the number of cases of product stocked, as well as with the distance walked by the driver. The predictors, the number of cases stocked and the distance walked by the driver, also appear to have a positive linear association with each other.

2. Fit MLR using `lm()`

To fit multiple linear regression (MLR)

```
result<-lm(Delivery~Number+Distance, data=Data)
```

where we list the predictors after `~` with a `+` operator in between the predictors. Another way would be

```
result<-lm(Delivery~., data=Data)
```

The `.` after `~` informs the `lm()` function to use every column other than `Delivery` in the data frame as predictors.

Just like with simple linear regression (SLR) we can get relevant information using `summary()`

```
summary(result)
```

```
##
## Call:
## lm(formula = Delivery ~ ., data = Data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7880 -0.6629  0.4364  1.1566  7.4197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.341231   1.096730   2.135 0.044170 *
## Number      1.615907   0.170735   9.464 3.25e-09 ***
## Distance    0.014385   0.003613   3.981 0.000631 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 22 degrees of freedom
## Multiple R-squared:  0.9596, Adjusted R-squared:  0.9559
## F-statistic: 261.2 on 2 and 22 DF,  p-value: 4.687e-16
```

3. Inference with MLR

We can easily find the results of the hypothesis tests associated with each coefficient in the output from `summary(result)`. Just like SLR, each coefficient is tested against a null hypothesis that $\beta_j = 0$ with a two-sided alternative.

The reported F-statistic of 261.2 is the ANOVA F test where $H_0 : \beta_1 = \beta_2 = 0$ and H_a : at least one of β_1, β_2 is not zero.

The confidence intervals for the coefficients can be found using `confint()`

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

```
##                    2.5 %     97.5 %
## (Intercept) 0.066751987 4.61571030
## Number      1.261824662 1.96998976
## Distance    0.006891745 0.02187791
```

The confidence interval for the mean response and the prediction interval for a new observation given a specific value of the predictors can also be found using `predict()`. For example, when the number of cases is 20 and the distance walked is 200 feet

```
newdata<-data.frame(Number=20, Distance=200)

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

```
##        fit      lwr      upr
## 1 37.53634 32.02142 43.05126
```

```
predict(result, newdata, level=0.95,
        interval="prediction")
```

```
##        fit      lwr      upr
## 1 37.53634 28.81233 46.26035
```

You might realize by now we are using the same functions as we did in SLR.

Practice: On your own, create a residual plot of the MLR model `result`, and assess if the response variable needs to be transformed. If needed, transform the response variable, and then re-assess the resulting model with another residual plot. We will learn about how to transform predictors in MLR in a later module.