Model Diagnostics for Regression

After fitting a regression model it is important to determine whether all the necessary model assumptions are valid before performing inference. If there are any violations, subsequent inferential procedures may be invalid resulting in faulty conclusions. Therefore, it is crucial to perform appropriate model diagnostics.

In constructing our regression models we assumed that the response y to the explanatory variables were linear in the β parameters and that the errors were independent and identically distributed (i.i.d) normal random variables with mean 0 and constant variance σ^2.

Model diagnostic procedures involve both graphical methods and formal statistical tests. These procedures allow us to explore whether the assumptions of the regression model are valid and decide whether we can trust subsequent inference results.

Ex. Data was collected on 100 houses recently sold in a city. It consisted of the sales price (in $), house size (in square feet), the number of bedrooms, the number of bathrooms, the lot size (in square feet) and the annual real estate tax (in $).

To read in the data and fit a multiple linear regression model with price as the response variable and size and lot as the explanatory variables, use the commands:

```
> Housing =read.table("C:/Users/Martin/Documents/W2024/housing.txt",header=TRUE)
> results = lm(Price ~ Size + Lot, data=Housing)
```

All information about the fitted regression model is now contained in results.

A. Studying the Variables

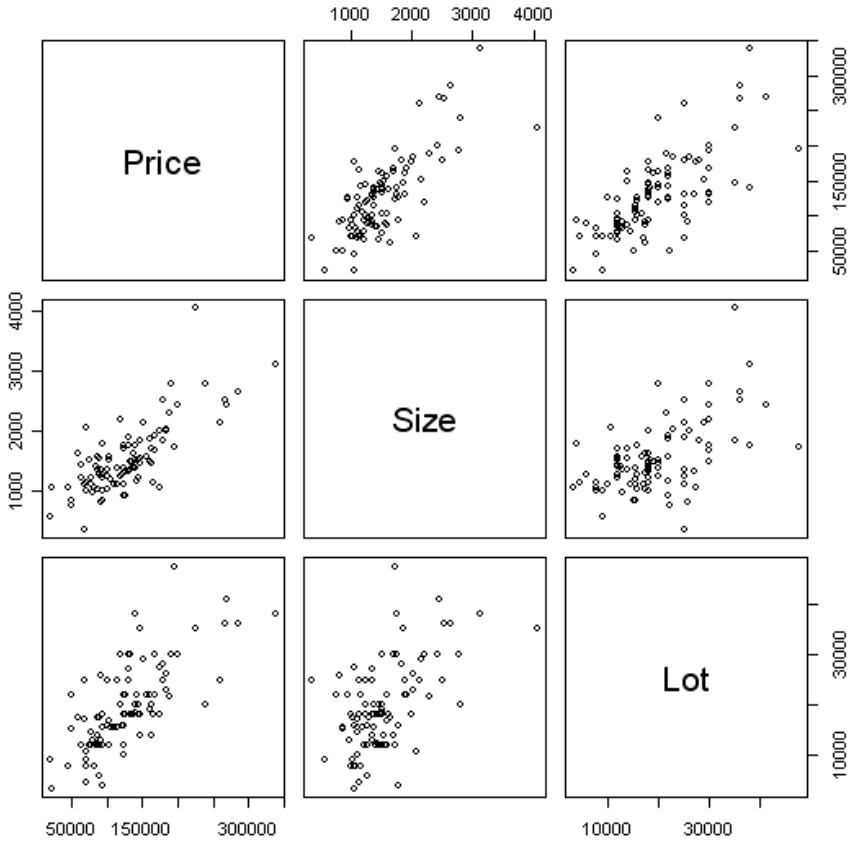
There are several graphical methods appropriate for studying the behavior of the explanatory variables. We are primarily concerned with determining the range and concentration of the values of the X variables and whether there exist any outliers.

Graphical procedures include histograms, boxplots and sequence plots. We can use the functions hist() and boxplot() to make histograms and boxplots, respectively.

In addition, scatter plots of all pair-wise combinations of variables contained in the model can be summarized in a scatter plot matrix.

A scatter plot matrix can be created by typing:

```
> pairs(~ Price + Size + Lot, data=Housing)
```



From this plot we can study the relationship between Price and Size, Price and Lot and Size and Lot.

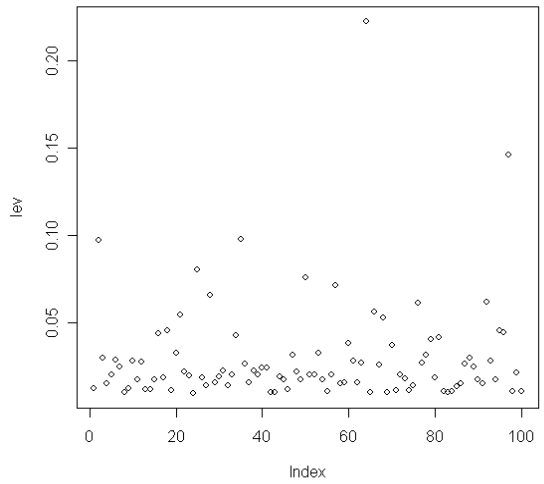
B. Residuals and Leverage

The leverage of an observation measures its ability to move the regression model all by itself by simply moving in the y-direction. The leverage measures the amount by which the predicted value would change if the observation was shifted one unit in the y-direction.

The leverage always takes values between 0 and 1. A point with zero leverage has no effect on the regression model. If a point has leverage equal to 1 the line must follow the point perfectly.

We can compute and plot the leverage of each point using the following commands:

```
> results = lm(Price ~ Size + Lot, data=Housing)
> lev = hat(model.matrix(results))
> plot(lev)
```



Note there is one point that has a higher leverage than all the other points (~0.25). To identify this point type:

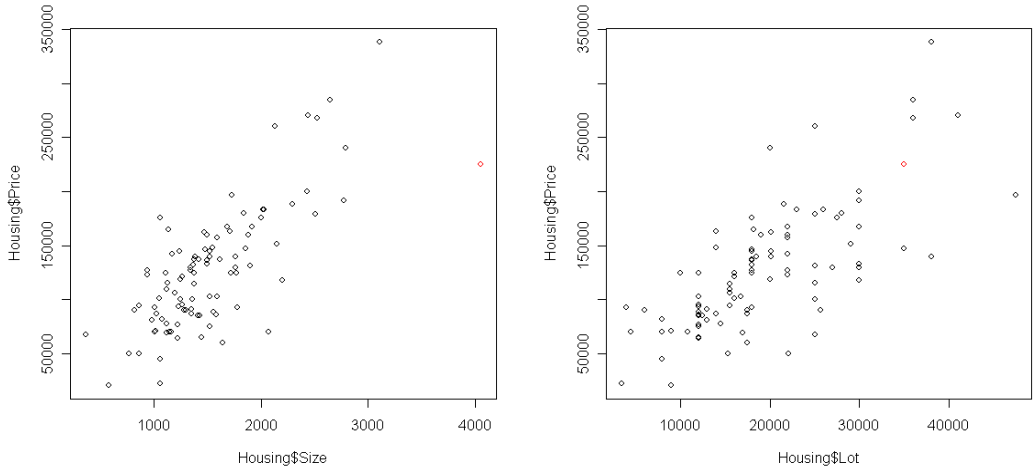
```
> Housing[lev >0.2,]
      Taxes Bedrooms Baths   Price       Size     Lot
64    4350        3          3      225000    4050    35000
```

This command returns all observations with leverage values above 0.2 in the Housing data set.

To study the influence of this particular point we can make scatter plots and specifically mark this point using the points() function.

For example:

```
> plot(Housing$Size, Housing$Price)
> points(Housing[64,]$Size, Housing[64,]$Price, col='red')
> plot(Housing$Lot, Housing$Price)
> points(Housing[64,]$Lot, Housing[64,]$Price, col='red')
```



Note the 64th observation is plotted in red in both graphs for easy identification.

The residuals can be accessed using the command results$res where results is the variable where the output from the function lm() is stored, i.e.

```
> results = lm(Price ~ Size + Lot, data=Housing)
```

Prior to studying the residuals it is common to standardize them to compensate for differences in leverage. The studentized residuals are given by:

$$r_i = \frac{e_i}{s\sqrt{1-h_{ii}}}$$

where e_i is the residual and h_{ii} the leverage for the i^{th} observation. Studentized residuals can be computed in R using the command:

```
> r = rstudent(results)
```

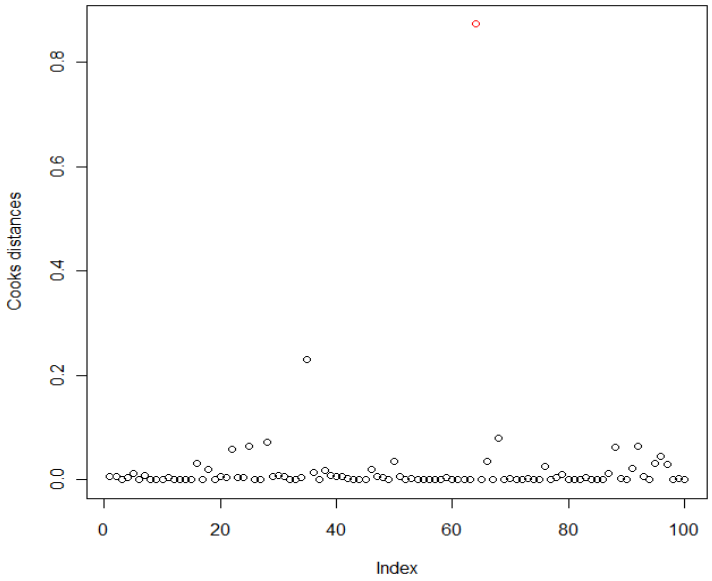
An influential point is one if removed from the data would significantly change the fit. An influential point may either be an outlier or have large leverage, or both, but it will tend to have at least one of those properties. Cook's distance is a commonly used influence measure that combines these two properties. It can be expressed as

$$C_i = \left(\frac{r_i}{p+1}\right)\left(\frac{p_{ii}}{1-p_{ii}}\right)$$

Typically, points with C_i greater than 1 are classified as being influential.

We can compute the Cook's distance using the following commands:

```
> cook = cooks.distance(results)
> plot(cook,ylab="Cooks distances")
> points(64,cook[64],col='red')
```



Note the Cook's distance for the 64th observation is roughly 0.90.

If we ultimately decide we need to remove an observation from the data set and refit the model, we can use the command:

```
> dat2 = dat[-64,]
```

This command creates a new data frame called dat2, which is a copy of the data frame dat except it excludes the 64th observation.

C. Residual Plots

We can use residuals to study whether:

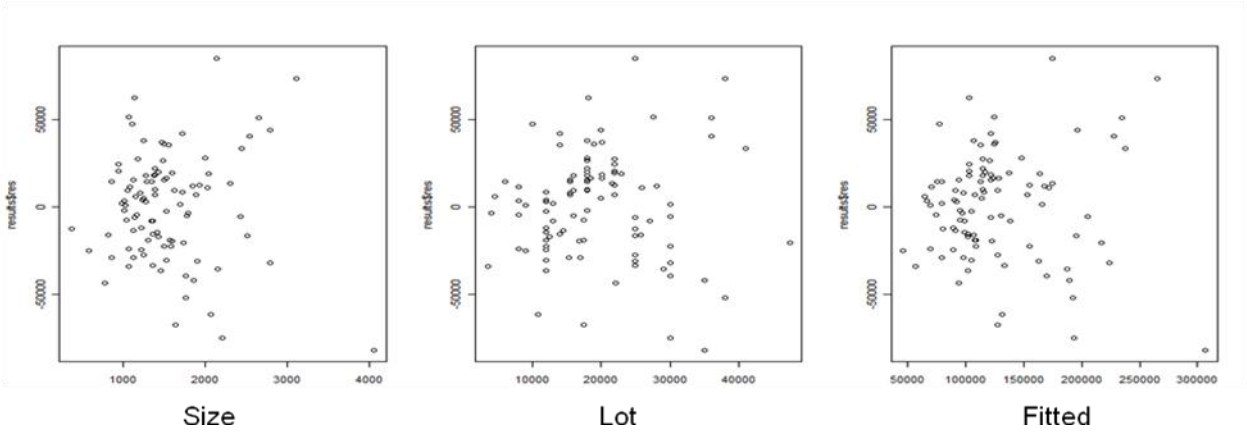
- The regression function is nonlinear.
- The error terms have nonconstant variance.
- The error terms are not independent.
- There are outliers.
- The error terms are not normally distributed.

We can check for violations of these assumptions by making plots of the residuals, including:

- Plots of the residuals against the explanatory variable or fitted values.
- Histograms or boxplots of the residuals.
- Normal probability plots of the residuals.

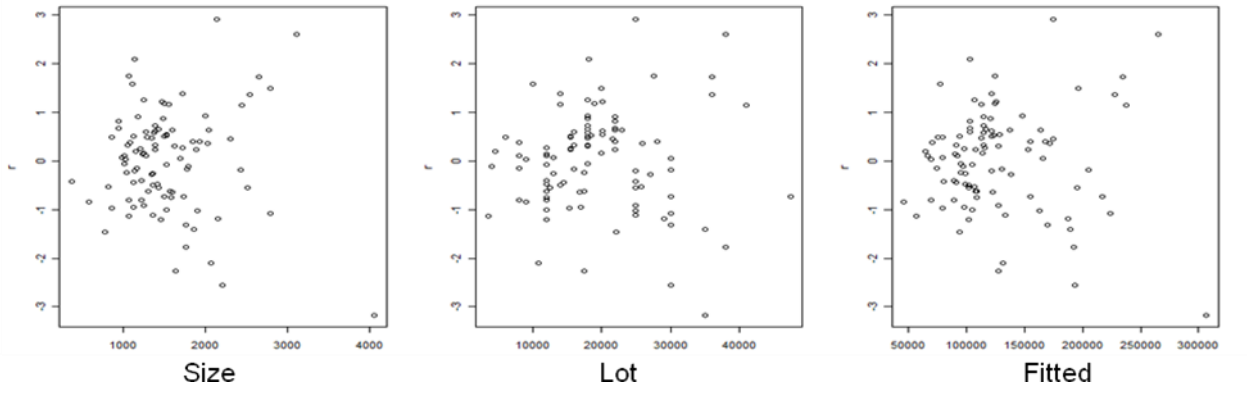
We plot the residuals against both the predicted values and the explanatory variables. The residuals should be randomly scattered about 0 and the width should be equal throughout for the constant variance assumption to hold.

```
> par(mfrow=c(1,3))
> plot(Size, results$res)
> plot(Lot, results$res)
> plot(results$fitted, results$res)
```



To make the same residual plots using the studentized residuals type:

```
> r = rstudent(results)
> plot(Size, r)
> plot(Lot, r)
> plot(results$fitted, r)
```



A Normal probability plot of the residuals can be used to check the normality assumption. Here each residual is plotted against its expected value under normality. To make normal probability plots, as well as a histogram, type:

```
> qqnorm(results$res)
> qqline(results$res)
> hist(results$res)
```