Linear Regression

Synferlo

May, 10

1 Simple Linear Regression

1.1 Estimation

$$\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x$$

$$\widehat{\beta}_1 = \rho_{xy}\frac{s_y}{s_x}, \text{ and } \widehat{\beta}_0 = \overline{y} - \widehat{\beta}_1\overline{x}$$

$\overline{x}, \overline{y}$: sample means

s_x, s_y: sample standard deviation

ρ_{xy}: the estimate of correlation between X and Y based on the data.

1.2 Statistical Inference

Table 1: Price and Volume (unnormalized result)

	Coefs	**SE**	**t-value**	**p-value**
(Intercept)	2.342e+03	8.799e+01	26.62	$< 2e-16$ ***
Volume	2.696e-07	5.252e-09	51.33	$< 2e-16$ ***
Multiple R-squared:	0.5061	Adjusted R-squared:	0.506	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3803 on 2571 degrees of freedom

F-statistic: 2635 on 1 and 2571 DF, p-value: $< 2.2e-16$

p-value and t-value for the coefs are the results of a two- tailed test based on t-distribution

with DOF = 2.

$$H_0 : \beta_j = 0$$

$$H_1 : \beta_j \neq 0$$

where $j = 0$ for the intercept β_0, and $j = 1$ for the coef. of the volume.

1.2.1 R^2 and adjusted R^2

$$R^2 = \rho_{xy}^2$$

In R,

```
# compute correlation then square it.
cor(df$Open_price, df$Volume, use = 'complete.obs')**2
# [1] 0.5061467
```

You can check this 0.506 with the R-squared we've got from $summary(result)$. They are exactly the same.

R-squared tells us 50 percent of the variation in the price can be attributed to volume.

The adjusted R-squared is important ONLY IF you are using the coef of determination to assess the overall quality of the fitted model in terms of a balance between goodness of fit(GOF) and complexity.

1.2.2 Other summary output

Residual standard error is the estimated SE of the ε, i.e., $\widehat{\sigma}$.

1.3 Categorical Predictor

Explanatory variables can be categorical.

1.3.1 Binary Variables

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

X can be either 0 or 1.

If so, the interpretation of β would be different. It's better to think of them like two intercepts, where β_0 provides the baseline of the response when $X = 0$, and β_1 represents the additive effect on the mean response if $X = 1$.

1.3.2 Multilevel variables

The categorical variables have more than two levels. For example, there can be many levels under education, e.g., high school, college, master, phd, etc.

Suppose there are k levels, then variable X can be written as

$$X = 1, 2, 3, ..., k$$

$$X_{(1)} = 0, 1 \quad X_{(2)} = 0, 1 \quad ...X_{(k)} = 0, 1$$

And we can write the model,

$$\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 X_{(2)} + \widehat{\beta}_{(2)} X_{(3)} + ... + \widehat{\beta}_{k-1} X_{(k)}$$

We normally use $k - 1$ of the dummy variables. Also, each observation only satisfy one of the levels, i.e., if you are a Ph.D. student, then you cannot be a high school student. Hence, when $X_{(i)} = 1$, all others equals to zero.

So, if one observation belongs to level 3, then the model (the predicted mean) would be

$$\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}2$$

Since the reference level(that omitted dummy) is defined as 1, if an observation has values of zero for all other dummies, it implies the obs originally had X = 1.

$$\widehat{y} = \widehat{\beta}_0$$

Here we have the result of a model with a dummy has four levels (Heavy, Never, Occas, Regular), where Heavy is the reference.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 173.7720     3.1028  56.005   <2e-16 ***
SmokeNever   -1.9520     3.1933  -0.611    0.542
SmokeOccas   -0.7433     3.9553  -0.188    0.851
SmokeRegul    3.6451     4.0625   0.897    0.371
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The observation in the reference category Heavy is represented solely by $\widehat{\beta}_0 = 173.7720$.

2 Multiple Linear Regression

2.1 Terminology

A lurking variable, is what we've learned of the omitted variable. It can lead to a omitted variable bias.

A nuisance or extraneous variable is a predictor of no interest, but has the potential to confound (mess up) relationships between other variables, and so affect your estimation. We are not interested in it, but we must add it to the model.

2.2 The model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \varepsilon$$

You have n obs and p explanatory variables. For each obs, the regression equation would be

$$y_i = \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + ... + \beta_p x_{p,i} + \varepsilon_i$$

where $i = 1, 2, ..., n$ stand for the i^{th} obs.

Least-squared:

$$\min \left(\sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + ... + \beta_p x_{p,i}))^2 \right)$$

Matrix Form:

$$\boldsymbol{Y} = \boldsymbol{X} \cdot \boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where $\boldsymbol{Y}$ and $\boldsymbol{\varepsilon}$ are $n \times 1$ matrices

$$\boldsymbol{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \text{and} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \quad \text{and} \quad \boldsymbol{X} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{p,1} \\ 1 & x_{1,2} & \cdots & x_{p,2} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1,3} & \cdots & x_{p,3} \end{bmatrix}$$

Matrix $\boldsymbol{X}$ is $n \times (p+1)$ dimension.

The OLS estimator $\widehat{\boldsymbol{\beta}}$:

$$\widehat{\boldsymbol{\beta}} = \begin{bmatrix} \widehat{\beta}_0 \\ \widehat{\beta}_1 \\ \vdots \\ \widehat{\beta}_p \end{bmatrix} = (\boldsymbol{X}^T \cdot \boldsymbol{X})^{-1} \cdot \boldsymbol{X}^T \cdot \boldsymbol{Y}$$

2.3 Interpretation of the coef

A coefficient for a specific variable should be interpreted as the change in the mean response for a one-unit increase in the variable, while holding all other variables constant.

2.4 Transformation

Two ways to approach the transformation: polynomial and logarithmic.

The transformation in general does not represent a universal solution to solving problems of nonlinearity in the trend, but it can at least improve how faithfully a linear model is able to represent those trends.

2.4.1 Polynomial

Add squared, cubic, and other polynomial terms to fit curved trend. For example,

$$\text{Income } = \beta_0 + \beta_1 edu + \beta_2 edu^2 + \beta_3 exp + \beta_4 exp^2 + \varepsilon$$

Use $I()$ function in R to add a polynomial term in $lm()$ function.

Code:

```
lm_poly_result = lm(norm_price~norm_volume+norm_supply
+I(norm_supply**2),data = df)
summary(lm_poly_result)
```

If the effect of adding one term is not good, we can try to add another cubic term and compare these two models.

Table 2: Quadratic vs. Cubic term model

	Model 1	**Model 2**
(Intercept)	-0.024260	-0.095378 ***
	(-1.523)	(-5.557)
norm_volume	0.465420***	0.418280 ***
	(28.338)	(24.920)
norm_supply	0.397987***	0.525587 ***
	(24.030)	(25.473)
I($norm\_supply^2$)	0.024269*	0.088337 ***
	(2.446)	(7.588)
I($norm\_supply^3$)		-0.030171 ***
		(-10.035)

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Now we can see, by adding cubic term, the model performs better.

2.4.2 Logarithmic

Transforming to a logarithmic scale can help reduce the severity of heavily skewed data.

2.5 Interaction Terms

When estimating regression models, you always have to accompany interactions with the main effect of the relevant predictors.

In R, use a ":" specify an interaction term.

Code:

```
inter_lm_result = lm(
                norm_price~norm_volume+norm_supply+norm_volume:norm_supply,
                data = df
)
summary(inter_lm_result)
```

3 Linear Model Selection

3.1 Goodness of fit vs. Complexity

GOF: refers to the goal of obtaining a model that best represents the relationships between the dependent and explanatory variables.

Complexity: how many terms (e.g., polynomial, etc) and additional functions in your model. The more you have, the more complex the model would be.

The principle of parsimony: The balancing act between GOF and complexity.
Our goal is to find a model that is as simple as possible without sacrificing too much GOF. The model satisfies this notion is a parsimonious fit.

3.1.1 General Guideline

1. You CANNOT remove individual levels of a categorical variable in a given model. Suppose college, master, phd are under edu, but only phd are significant. You cannot remove the college and master. You can ONLY remove the entire categorical variable, i.e., edu.

2. If an interaction term is present in the fitted model, ALL lower- order interactions and main effects of the relevant variables MUST remain in the model. Suppose you add an interaction term, $edu \times exp \times age$, then the main effect (edu, exp, age) and all lower-order interaction terms $(edu \times exp, edu \times age, exp \times age)$ should be shown in the model. Hence, in R, you'd better use $VR_1 * VR_2$ for interaction term, so that R will add all these terms for you. You will not miss any one of them.

3. Keep ALL lower-order polynomial terms in the model if the highest is deemed significant.

3.2 Model Selection Algorithms

3.2.1 Nested Comparisons: The Partial F-Test

It is the most direct way to compare several different models. It looks at two or more nested models. The less complex model is a reduced version of the more complex model.

$$\widehat{y}_{\text{redu}} = \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \widehat{\beta}_2 x_2 + ... + \widehat{\beta}_p x_p$$
$$\widehat{y}_{\text{full}} = \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + ... + ...\widehat{\beta}_p x_p + ... + \widehat{\beta}_q x_q$$

Clearly, $\widehat{y}_{\text{full}}$ is more complex than $\widehat{y}_{\text{redu}}$, so we say that $\widehat{y}_{\text{redu}}$ is nested within $\widehat{y}_{\text{full}}$.
The **partial F-Test** tries to answer if it is worth it to include any additional variables. Its goal is to test whether include those extra $q - p$ terms in $\widehat{y}_{\text{full}}$ provide a statistically significant improvement GOF.

$$H_0 : \beta_{p+1} = \beta_{p+2} = ... = \beta_q = 0$$
$$H_1 : \text{At least one of the } \beta_j \neq 0 (\text{ for } j = p, ..., q)$$

If the p-value is less than the significant level, then we say it is worth it because at least one of those extra $q - p$ terms is non-zero.
F statistics:

$$F = \frac{(R^2_{\text{full}} - R^2_{\text{redu}}\,(n - q - 1))}{(1 - R^2_{\text{full}})(q - p)}$$

It follows an F distribution with $df_1 = q - p$, $df_2 = n - q$ degree of freedom. The p-value is found as the upper-tail area from F as usual.

In R, we can use *anova*(model1, model2) to conduct a partial F -test. We pass the reduced model first, then the complex model.

Code:

```
redu_model = lm(norm_price~norm_volume+norm_supply, data = df)
complex_model = lm(norm_price~norm_volume*norm_supply, data = df)
anova(redu_model, complex_model)
```

In the complex model, instead of the main effect, we add interaction term of these two. And the result shows that this additional interaction term do provide a statistically significant improvement in fit because the p-value is pretty small. We reject the null.

The number of variables in the reduced model is $p = 2$, and the number of variables in the complex model is $q = 3$ (two main effects and a interaction term). Hence we can compute the DF (degree of freedom), $DF = q - p = 3 - 2 = 1$. And that is shown in the second row of the table, i.e., column DF.

From the result, statistics $F = 15.717$, $df_1 = 1$, $df_2 = 2569$.

Again, we use the partial F-test to verify if the complex model provides an improvement in fit.

Disadvantage: It can be difficult to manage if you have many different models to fit when you have many explanatory variables. Forward Selection can help us to deal with this problem.

Table 3: The Partial F-test result

	Res.DF	**RSS**	**DF**	**F**	**p-value**
Model 1	2570	1029.8			
Model 2	2569	1023.6	1	15.717	7.556e-05 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Model 1: norm_price = norm_volume + norm_supply
Model 2: norm_price = norm_volume * norm_supply

3.2.2 Forward Selection

Also called forward elimination.

Step 1:

It starts with an intercept-only model, and then perform a series of independent tests to determine which of your explanatory variables significantly improves the GOF.

Step 2:

Then you add that term and execute the series of tests again for all remaining terms to determine which of those would further improve the fit.

Step 3:

The loop stops until there's no term can further improve the GOF.

Table below shows the result of intercept-only model.

Table 4: Intercept-only model

	Coef	**Se**	**t-value**	**p-value**
Intercept	-2.036e-16	1.971e-02	0	1

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Now we use *add*1() to find the most helpful variable to be added to our intercept-only model.

Table 5: Forward selection

	DF	**Sum of Sq**	**RSS**	**ACI**	**F-value**	**p-value**
none			2572.0	1.0		
norm_supply	1	1173.3	1398.7	-1564.4	2156.8	< 2.2e-16 ***
norm_volume	1	1301.8	1270.2	-1812.3	2635.0	< 2.2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

From we want to add the one with "three stars". Let's add norm_ supply first. Then you keep doing this. The R code is below.

Code:

```
model1 = lm(norm\_price~1, data = df)
summary(model1)

add1(model1, scope = .~.+norm\_supply+norm\_volume, test = 'F')

model2 = update(model1, formula = .~. + norm\_supply)
summary(model2)

add1(model2, scope = .~.+norm\_volume, test = 'F')
```

3.2.3 Backward Selection

As a reverse of the Forward Selection, Backward Selection starts with your full/complex model, and systematically drops terms.

In R, we instead of *add*1(), we use *drop*1()for backward selection and conduct the partial F-test and *update*(). In contrast with the adding terms in forward selection, here we are going to drop those terms DO NOT result in a statistically significant detriment to GOF. We drop those terms do not have "stars" in the result of *drop*1().

Code:

```
model1 = lm(norm_price~norm_volume*norm_supply, data = df)
summary(model1)
drop1(model1, test = 'F')
# use update(model1, .~.-norm_volume*norm_supply) to drop a term.
# Since this interaction term is significant, we don't want to drop it.
```

3.2.4 Stepwise AIC Selection

The partial F-test is the most common test-based model selection method.

Besides it, we also have a criterion-based approach. One of them is known as Akaike's

Information Criterion, or AIC for short.

For a given lnear model, AIC is calculated as follows:

$$AIC = -2 \times \mathscr{L} + 2 \times (p + 2),$$

where $\mathscr{L}$ is a measure of GOF named the log-likelihood, and p is the number of regression parameters in the model, excluding the overall intercept (explanatory variables). A smaller values of AIC refer to more parsimonious models.

NOTE: The value of AIC has no direct interpretation and is useful ONLY when you compare it against the AIC of another model. Hence, you can decide on which term to add or drop based on the change of the AIC values, instead of focusing exclusively on the significance of the change via the F-test when you conduct a forward or backward selection.

You can implement stepwise AIC selection yourself by using either add1 or drop1 at each stage, but forfunately R provides the built-in step function for you.

Progress:

You can start with any model you like. Here, let's start with the intercept-only model.

Code:

```
model1 = lm(norm_price~1, data = df)
stepwise_model_selection = step(model1,
                                scope = .~. + norm_volume*norm_supply
)
summary(stepwise_model_selection)
```

R would return a list of results, each block of them starts with a regression model and its AIC value, and presents the change of AIC by adding or deleting a term $(+, -)$.

The final model is stored, and you can check it using *sumamry*().

Note:

The AIC is sometimes criticized for a tendency to err on the side of more complexity and higher p-values. To balance this, you can increase the penalizing effect by increasing the multiplicative contribution of $p+2$ on the RHS. Now is $2 \times (p+2)$, you can try $2.5 \times (p+2)$, etc.

Criterion-based measures are incredibly useful when you have models that aren't nested (ruling out the partial F-test).

3.2.5 BIC selction algorithm

Besides AIC, the corrected AIC (AIC_c) and the BIC (Bayesian Information Criterion) are alternatives to AIC. They impose heavier penalties on complexity than the default AIC.

Also, the adjusted R^2 is another selection algorithm. Recall, R^2 does not penalize the complexity, but, here, the adjusted R^2 does incorporate a penalty for complexity relative to the sample size n.

$$\overline{R}^2 = 1 - \frac{(1-R^2)(n-1)}{n-p-1},$$

where p is the number of predictor terms. Monitoring $\overline{R}^2$ can be useful as a quick check between nested models – a higher value points to a preferred model.

3.3 Residual Diagnostics

We are going to cover the methods that are essential for determining the validity of your model, model diagnostics. The goal is to ensure that your model is valid and accurately represents the relationships in your data.

3.3.1 Inspecting and Interpreting Residuals

We can plot a scatterplot of the observed-minus-fitted raw residuals. By assumption, they should appear randomly around zero. This plot can also be used to detect heteroscedasticity.

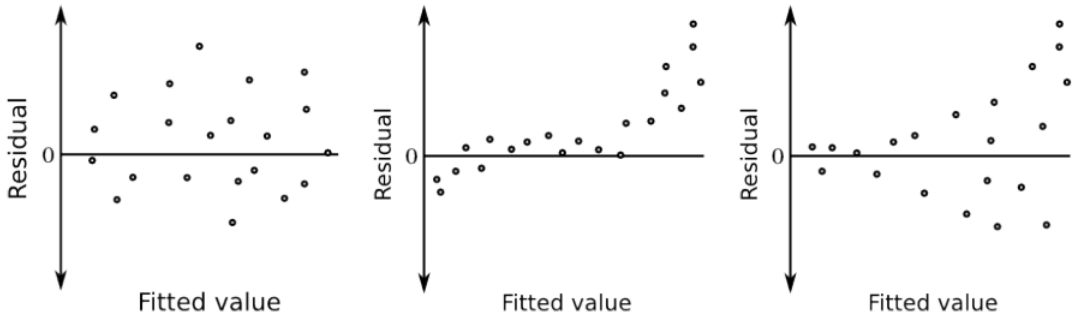


Figure 22-1: Three impressions of a hypothetical residuals versus fitted diagnostic plot from a linear regression: random (left), systematic (middle), and heteroscedastic (right)

The figure above shows the scatterplot of the residuals. The one on the left shows a homoscedasticity, good. The one on the right is a heteroscedasticity result.

Note:

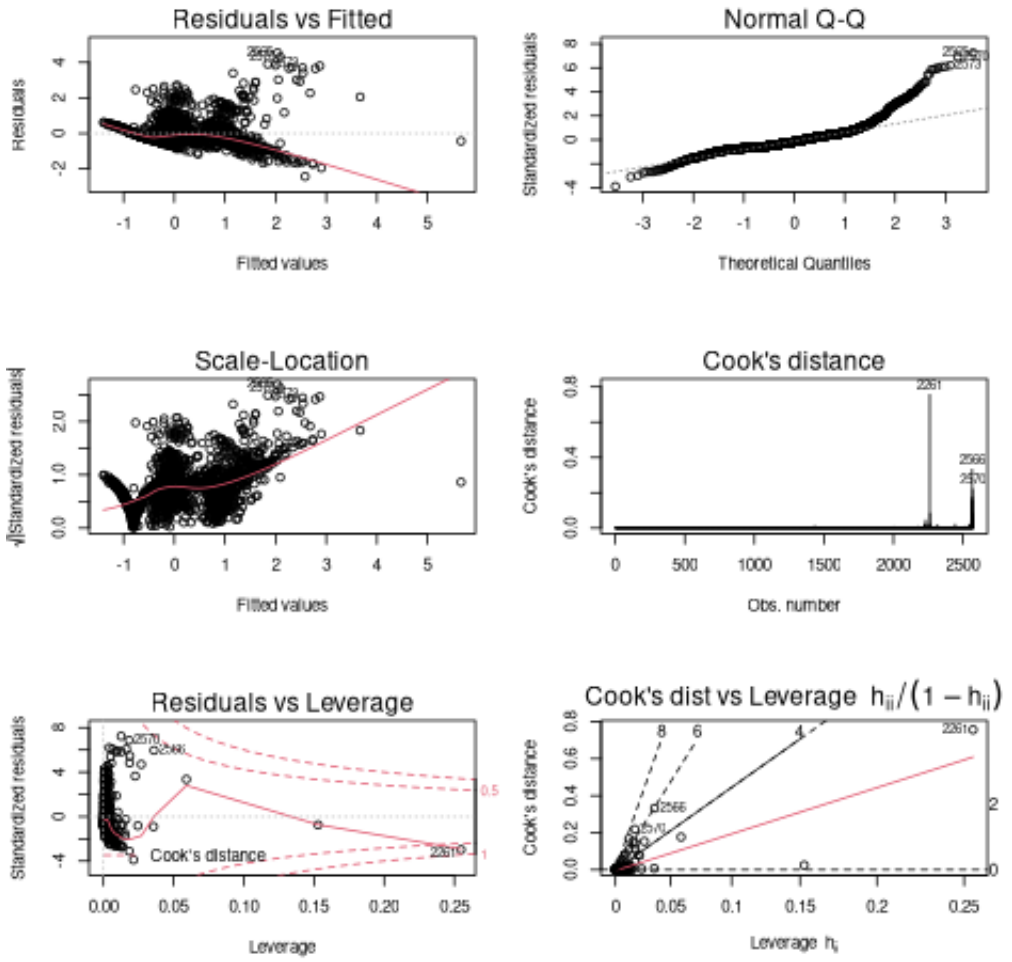
Even if your result does not like the figure on the left, you can still improve your model by adding explanatory variable, changing the treatment of a categorical variable, or performing nonlinear transformations of certain continuous variables to reduce nonlinearity.

In R, you can use plot() function on a lm() object, it can produce six types of diagnostic plot of the fit. You can manually select a particular plot specifying which = i argument, where i stands for the i^{th} plot.

Code:

```
model1 = lm(norm_price~norm_supply*norm_volume, data = df)
png('figures/residual_diagnostic_plots.png')
par(mfrow = c(2,3))
plot(model1, which = 1)
plot(model1, which = 2)
plot(model1, which = 3)
plot(model1, which = 4)
plot(model1, which = 5)
plot(model1, which = 6)
dev.off()
```

Figure 1: Six types of diagnostic plots.



The scale-location plot provides

$$\left| \frac{e_i}{\widehat{\sigma}\{1-h_{ii}\}^{\frac{1}{2}}} \right|^{\frac{1}{2}}$$

It is used to reveal tends in the size of the departure of each data point from its fitted value.

It is much more useful than the raw residuals in detecting things such as heteroscedasticity.

3.3.2 Assessing Normality

You can use a normal QQ plot (pass $which = 2$) to check the normality.

```
model1 = lm(norm_price~norm_supply*norm_volume, data = df)
```

```
plot(model1, which = 2)
```

There are also other ways to test for normality, such as Shapiro-Wilk test. The null is that the data are normally distributed. We can use *shapiro.test*() in R to implement this test.

Code:

```
model1 = lm(norm_price~norm_volume*norm_supply, df)
shapiro.test(rstandard(model1))
#          Shapiro-Wilk normality test
#
# data:  rstandard(model1)
# W = 0.84736, p-value < 2.2e-16
# Here, we reject the null, hence, the data are not normally distributed.
```

3.3.3 Illustrating outliers, leverage, and influence

Leverage:

The extremity of the values of the explanatory variable. A high-leverage point is an observation with explanatory values extreme enough to potentially significantly affect the slopes or trends in the fitted model.

An outlier can have a high or low leverage.

Influence:

An obs with high leverage that DOES affect the estimated trends is deemed influential.