



**Department of Applied Mathematics and Statistics**

**I3-AMS-Bachelor Degree**

**Group: I3-AMS-04**

**Subject : Linear Regression**

**Topic : Analyze the factors that affect customer satisfaction in the airline industry using ordinal logistic regression**

**Member**

Ek Vong Panharith	e20200877
Tang Piseth	e20201634
Hon Ratana	e20201053
Sao Samarth	e20200084
Pheng Sothea	e20201734
Kry Senghort	e20200706

**Academic Year : 2022-2023**

## Report Model Selection:

Model Selection in linear regression is the process of choosing the most appropriate model among a set of candidate models for predicting the relationship between the dependent variable (response variable) and one or more independent variables (predictor variable). It involves evaluating different models based on their goodness of fit and selecting the one that best explains the data and minimizes the risk of overfitting. There are several common methods of model selection in linear regression, involving:

- Forward Selection
- Backward Selection
- Stepwise Selection

### I.) Forward Selection:

Forward selection is a common method for model selection in linear regression where variables are sequentially added to the model based on their statistical significance or other criteria. The basic idea is to start with an empty model and iteratively add one variable at a time, selecting the variable that improves the model the most until a stopping criterion is met.

Here's an example of forward selection in linear regression using R programming language:

```
# Load necessary libraries
library(alr4)

# Load the UBSprices data
data("UBSprices")

# Create an empty model
model <- lm(rice2009 ~ 1, data = UBSprices)

# Start with an intercept-only model
# Define a stopping criterion (e.g., based on p-value or AIC)
alpha <- 0.05 # Set the significance level
max_vars <- ncol(UBSprices) - 1 # Maximum number of variables to
include
stop_criterion <- FALSE # Initialize stopping criterion

# Perform forward selection
while (!stop_criterion) {
  # Initialize variables to keep track of best candidate variable
  best_candidate <- NULL
  best_pvalue <- Inf

  # Loop through each remaining predictor variable
  for (var in colnames(UBSprices)[-1]) {
    # Skip variable if it's already in the model
    if (var %in% colnames(model$coefficients)) next
    # Fit a model with the current candidate variable added to model
    Candidate_model <- lm(rice2009~.+UBSprices[[var]],data=UBSprices)
    # Extract p-value of the candidate variable
```

```

    pvalue <-
summary(candidate_model)$coefficients[rownames(summary(candidate_model)
$coefficients)==var,"Pr(>|t|)"]
    # Check if the p-value is better than the current best p-value
    if (pvalue < best_pvalue){
        best_candidate <- var
        best_pvalue <- pvalue
    }
}
    # Check if the best candidate variable meets the stopping criterion
    if (best_candidate == NULL || best_pvalue >alpha||
length(model$coefficients)-1 >= max_vars) {
        stop_criterion <- TRUE
    } else {
        # Add the best candidate variable to the model
        model <- update(model, formula = as.formula(paste0("~ . +",
best_candidate)))
    }
}
# Print the final model
print(model)

```

**Output:**

Call:

```
lm(formula = rice2009 ~ . + Iran + Pakistan + Bangladesh, data =
UBSprices)
```

Coefficients:

(Intercept)	Iran	Pakistan	Bangladesh
2.3546	0.6784	0.4208	0.4811

Residual standard error: 0.07022 on 67 degrees of freedom

Multiple R-squared: 0.9533, Adjusted R-squared: 0.9516

F-statistic: 557.7 on 4 and 67 DF, p-value: < 2.2e-16

In this example, the UBSprices dataset is loaded, and an empty model is created with only an intercept term. The final model selected by the forward selection process includes four variables: n, and Bangladesh, in addition to the intercept term. The coefficients of the variables in the model, the residual standard error, multiple R-squared, adjusted R-squared, and F-statistic with associated p-value are also displayed as part of the output. These statistics provide information about the goodness of fit and statistical significance of the model.

The forward selection loop iterates through each predictor variable not currently in the model, fits a model with the candidate variable added to the model, and selects the variable with the lowest p-value as the best candidate variable. The loop continues until a stopping criterion is met, which can be based on a predefined significance level (alpha) or a maximum number of variables (max\_vars) to include in the model. The final model is then printed, which includes the selected variables based on the forward selection process.

\* Note that the specific stopping criterion and selection criteria can be customized based on the specific requirements of the analysis and the dataset being used. It's important to validate the selection model using appropriate techniques, such as cross-validation, to ensure its generalizability to new data.

## **II.) Backward Selection:**

Backward selection is a method used for feature selection in linear regression, where the model starts with a full model that includes all predictor variables and iteratively removes one variable at a time based on a specified criterion until a stopping criterion is met. The idea is to eliminate less significant variables from the model to obtain a more parsimonious model with a reduced number of predictor variables.

Here are the steps involved in backward selection:

- Step 1: Fit the full model - Start with a full model that includes all predictor variables.
- Step 2: Model fitting - Fit the full model using a regression function (e.g., `lm()` in R) and obtain the initial model fit.
- Step 3: Criterion selection - Choose a criterion to evaluate the model fit, such as Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), adjusted R-squared, or any other relevant criterion.
- Step 4: Variable elimination - Remove the least significant variable based on the chosen criterion. This can be done by comparing the criterion value of the current model with the criterion values of models with one variable removed at a time, and selecting the model with the smallest criterion value.
- Step 5: Model update - Update the model by removing the selected variable, and go back to step 3 to evaluate the new model fit.
- Step 6: Stopping criterion - Repeat steps 3, 4, and 5 until a stopping criterion is met. The stopping criterion could be a predefined number of iterations, reaching a specific criterion value, or when no further variable can be removed.
- Step 7: Final model - Once the stopping criterion is met, the final model is obtained with the selected subset of predictor variables.

Backward selection is a useful method for identifying the most important predictor variables in a linear regression model and building a simpler model with fewer variables, which can improve interpretability, reduce multicollinearity, and potentially improve model performance. However, it has limitations such as potential omission of important variables and dependence on the chosen criterion for variable selection, so careful consideration and validation of the final model is necessary.

Here's an example of backward selection code in R using the `step()` function:

```
# Load the necessary library
```

```

library(stats)
# Fit the initial full model
full_model <- lm(rice2009 ~ rice2003 + bread2003 + bigmac2003 +
rice2009 + bread2009 + bigmac2009, data = UBSprices)
# Perform backward selection
final_model <- step(full_model, direction = "backward")
Output:
Start:  AIC=-1501.84
rice2009 ~ rice2003 + bread2003 + bigmac2003 + rice2009 + bread2009
+ bigmac2009

```

	Df	Sum of Sq	RSS	AIC
- bigmac2003	1	0.0002435	0.0257	-1503.5
- bread2003	1	0.0005893	0.0260	-1503.3
- rice2003	1	0.0006659	0.0261	-1503.2
<none>			0.0255	-1501.8
- bread2009	1	0.0018731	0.0273	-1499.5
- bigmac2009	1	0.0023185	0.0278	-1498.0
- rice2009	1	0.0036019	0.0292	-1495.0

```

Step:  AIC=-1503.53
rice2009 ~ rice2003 + bread2003 + rice2009 + bread2009 + bigmac2009

```

	Df	Sum of Sq	RSS	AIC
- bread2003	1	0.0004896	0.0262	-1505.1
- rice2003	1	0.0006570	0.0264	-1504.9
<none>			0.0257	-1503.5
- bread2009	1	0.0018776	0.0276	-1501.2
- bigmac2009	1	0.0033948	0.0291	-1498.3
- rice2009	1	0.0043294	0.0300	-1496.4

```

Step:  AIC=-1505.07
rice2009 ~ rice2003 + rice2009 + bread2009 + bigmac2009

```

	Df	Sum of Sq	RSS	AIC
- rice2003	1	0.0007413	0.0270	-1506.3
<none>			0.0262	-1505.1
- bread2009	1	0.0018779	0.0281	-1503.1
- bigmac2009	1	0.0033244	0.0295	-1500.1
- rice2009	1	0.0043304	0.0305	-1498.4

```

Step:  AIC=-1506

```

The `step()` function in R performs backward selection, which starts with a full model that includes all the predictor variables (`rice2003`, `bread2003`, `bigmac2003`, `rice2009`, `bread2009`, `bigmac2009`) and then iteratively removes one variable at a time based on a specified criterion (e.g., AIC, BIC, adjusted R-squared) until the stopping criterion is met.

### III.) Stepwise Selection:

Stepwise selection is a method used for feature selection in linear regression that combines elements of both forward selection and backward selection. It starts with an empty model and iteratively adds or removes predictor variables based on specified criteria until a stopping criterion is met. Stepwise selection is a more flexible approach compared to forward and backward selection, as it allows for both addition and removal of variables at each step.

Here are the steps involved in stepwise selection:

- Step 1: Start with an empty model - Begin with an empty model that does not include any predictor variables.
- Step 2: Model fitting - Fit the empty model using a regression function (e.g., `lm()` in R) and obtain the initial model fit.
- Step 3: Criterion selection - Choose a criterion to evaluate the model fit, such as Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), adjusted R-squared, or any other relevant criterion.
- Step 4: Forward selection - Add one predictor variable to the model based on the chosen criterion. This can be done by considering all available predictor variables and selecting the variable that improves the model fit the most according to the chosen criterion.
- Step 5: Backward selection - Remove one predictor variable from the model based on the chosen criterion. This can be done by considering all variables in the current model and selecting the variable that has the least impact on the model fit according to the chosen criterion.
- Step 6: Model update - Update the model by either adding or removing the selected variable, and go back to step 3 to evaluate the new model fit.
- Step 7: Stopping criterion - Repeat steps 3, 4, 5, and 6 until a stopping criterion is met. The stopping criterion could be a predefined number of iterations, reaching a specific criterion value, or when no further variable can be added or removed.
- Step 8: Final model - Once the stopping criterion is met, the final model is obtained with the selected subset of predictor variables.

Stepwise selection allows for a more flexible approach to feature selection in linear regression, as it combines forward and backward selection methods and allows for addition or removal of variables at each step. However, it also has limitations such as potential omission of important variables, sensitivity to the chosen criterion for variable selection, and potential overfitting if not carefully validated. Therefore, caution and validation of the final model are necessary when using stepwise selection.

Consider an example of stepwise selection for feature selection in a linear regression problem using R programming language. Suppose we have a dataset with multiple predictor variables (`x1`, `x2`, `x3`, `x4`, etc.)

and a target variable (y), and we want to perform stepwise selection to select the best subset of predictor variables for our linear regression model.

Here's an example implementation of stepwise selection in R using the built-in mtcars dataset:

```
# Load the mtcars dataset
```

```
library('datasets')
```

```
data(mtcars)
```

```
mtcars
```

```
# Fit the initial model with an empty set of predictor variables
```

```
initial_model <- lm(mpg ~ 1, data = mtcars)
```

```
# Define the criterion for model evaluation (e.g., AIC)
```

```
criterion <- AIC
```

```
# Stepwise selection
```

```
while (TRUE) {
```

```
  # Perform forward selection
```

```
  forward_model <- step(initial_model, scope = list(lower =  
formula(initial_model), upper = formula(mpg ~ .)), direction =  
"forward", criterion = criterion)
```

```
  # Perform backward selection
```

```
  backward_model <- step(initial_model, scope = list(lower =  
formula(initial_model), upper = formula(mpg ~ .)), direction =  
"backward", criterion = criterion)
```

```
  # Compare the models
```

```
  if (criterion(forward_model) < criterion(backward_model)) {
```

```
    # If forward model has a lower criterion value, update the initial  
model with the forward model
```

```
    initial_model <- forward_model
```

```
    print("Forward Selection:")
```

```
    print(summary(initial_model))
```

```
  } else if (criterion(backward_model) < criterion(forward_model)) {
```

```
    # If backward model has a lower criterion value, update the initial  
model with the backward model
```

```
    initial_model <- backward_model
```

```
    print("Backward Selection:")
```

```
    print(summary(initial_model))
```

```
  } else {
```

```
    # If neither model improves the criterion value, break the loop
```

```
    break
```

```
  }
```

```
}
```

```
# Final selected model
```

```
final_model <- initial_model
```

```
print("Final Selected Model:")
```

```
print(summary(final_model))
```

**Output:**

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

```
Start: AIC=115.94
```

```
mpg ~ 1
```

```
Start: AIC=115.94
```

```
mpg ~ 1
```

```
"Final Selected Model:"
```

```
Call:
```



```
lm(formula = mpg ~ 1, data = mtcars)
Residuals:
    Min       1Q   Median       3Q      Max
-9.6906 -4.6656 -0.8906  2.7094 13.8094
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   20.091     1.065   18.86  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 6.027 on 31 degrees of freedom
```

#### IV.) Confounder:

A confounder, also known as a confounding variable or a confounding factor, is an extraneous variable that is related to both the independent variable and the outcome variable in a research study. It can introduce bias into the study results by distorting the true relationship between the independent variable and the outcome variable. Confounding can occur when there is an association between the independent variable and the outcome variable, but this association is not due to a causal relationship, but rather due to the presence of a third variable that is related to both. Confounding variables can distort the true effect of the independent variable on the outcome variable, leading to inaccurate or misleading conclusions.

Controlling for confounders is an important step in research design and analysis to ensure that the observed relationship between the independent variable and the outcome variable is not influenced by the presence of confounding variables. This can be done through various techniques, such as randomization in experimental studies, matching in observational studies, or statistical adjustment using regression analysis or other statistical methods.

It is essential to identify and account for potential confounding variables in research studies to obtain valid and reliable results and to draw accurate conclusions about the relationship between the independent variable and the outcome variable. Failure to adequately account for confounding variables can lead to erroneous conclusions and may compromise the validity of research findings.

#### Here is an example to illustrate the concept of confounding.

Suppose we are interested in investigating the relationship between coffee consumption and the risk of developing heart disease. We conduct a study where we collect data on coffee consumption (in cups per day) and the occurrence of heart disease (yes or no) from a sample of individuals. The results show that there is a significant association between coffee consumption and heart disease risk, with higher coffee consumption being associated with a higher risk of heart disease.

However, upon further examination, we realize that there is a potential confounding variable that may be influencing this relationship - age. Older individuals may be more likely to consume more coffee and also have a higher risk of heart disease due to age-related factors. In this case, age is a confounding variable because it is related to both the independent variable (coffee consumption) and the outcome variable (heart disease risk).

To account for this confounding variable, we can use statistical techniques to control for age. One approach is to stratify the data by age groups (e.g., <40 years, 40-60 years, >60 years) and separately

analyze the association between coffee consumption and heart disease risk within each age group. Another approach is to use statistical regression analysis to adjust for age as a covariate in the analysis, which can help to isolate the independent effect of coffee consumption on heart disease risk after accounting for the influence of age.

By controlling for the confounding variable of age, we can obtain a more accurate estimate of the true relationship between coffee consumption and heart disease risk, and draw more valid conclusions about the potential association between the two variables. This illustrates the importance of identifying and accounting for confounding variables in research studies to ensure the validity of the findings.

## V.) Collider:

In statistics, a collider is a variable that can induce an association between two other variables when conditioning on it. This is also known as the "collider bias" or the "selection bias." In other words, a collider is a variable that becomes a common effect of two or more variables in a causal network when conditioning on it, leading to spurious associations or incorrect inferences.

To illustrate the concept of a collider, consider a simple example using the variables "A", "B", and "C". Let's assume "A" and "B" are independent variables, and "C" is the outcome variable. If "A" and "B" are not directly related to each other, but both are related to "C", conditioning on "C" can create a spurious association between "A" and "B". This is because "C" acts as a collider, inducing an association between "A" and "B" when conditioned on, even if they are not directly related.

Here's an example in R to further illustrate the concept of a collider:

```
# Generate a synthetic dataset
set.seed(123)
n <- 1000
A <- rnorm(n)
B <- rnorm(n)
C <- A + B + rnorm(n)

# Perform a simple linear regression of A on C
model1 <- lm(A ~ C)
summary(model1)

# Perform a simple linear regression of B on C
model2 <- lm(B ~ C)
summary(model2)

# Perform a multiple linear regression of A and B on C
model3 <- lm(A + B ~ C)
summary(model3)

Output:
Call:
lm(formula = A + B ~ C)

Residuals:
      Min       1Q   Median       3Q      Max
-2.54926 -0.57268  0.01004  0.58013  2.27329

Coefficients:
```

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03190    0.02574   1.239   0.215
C            0.69359    0.01451  47.796  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 0.8138 on 998 degrees of freedom
Multiple R-squared:  0.696,    Adjusted R-squared:  0.6957
F-statistic: 2284 on 1 and 998 DF,  p-value: < 2.2e-16

```

## VI.) Colliders VS Confounders:

Confounding and colliding variables are related concepts in statistics and causal inference, but they have some key differences:

- **Definition:** Confounding refers to a situation where an extraneous variable, called a confounder, affects the relationship between an exposure variable and an outcome variable, leading to a spurious association. Colliding, on the other hand, refers to a situation where a variable acts as a common effect of two or more other variables, and conditioning on this variable induces an association between the other variables, even if they are not directly related.
- **Role:** Confounders are extraneous variables that can distort the true relationship between an exposure and an outcome variable, leading to confounding bias. Colliders, on the other hand, are variables that can induce spurious associations between other variables when conditioned on, leading to collider bias.
- **Mechanism:** Confounding occurs when a third variable is associated with both the exposure and the outcome, and it acts as a common cause of both, leading to an incorrect inference of the true relationship between the exposure and the outcome. Colliding, on the other hand, occurs when a variable is a common effect of two or more other variables, and conditioning on it induces an association between the other variables, even if they are not directly related.
- **Handling:** Confounding can be addressed through various methods, such as stratification, matching, or statistical techniques like regression adjustment or propensity score methods. Colliders can also be addressed using similar methods, but the appropriate approach depends on the specific context and research question.

In summary, confounding and colliding variables are both sources of bias in statistical analyses, but they differ in their underlying mechanisms and how they affect the relationship between variables. It's important to carefully consider and appropriately handle both confounders and colliders in statistical analyses to obtain valid and reliable results.

## VII.) Collapsibility:

Collapsibility is a concept in statistics that refers to the property of a measure of association or effect size remaining unchanged or "collapsing" when stratified or conditioned on another variable. In other words, a measure of association or effect size is collapsible if its value does not change when additional variables are controlled for or conditioned on.

Collapsibility is an important property of some statistical measures, particularly in the context of confounding and Simpson's paradox. Simpson's paradox is a phenomenon where the direction or magnitude of an association between two variables can change when a third variable is included in the analysis, leading to potentially misleading conclusions. Collapsibility helps to determine whether an observed association between two variables is spurious due to confounding or Simpson's paradox, or whether it reflects a true relationship between the variables.

Measures of association or effect size that are collapsible are considered to be more reliable and robust, as they are not influenced by additional variables that may be controlled for or conditioned on. Examples of collapsible measures include the odds ratio in logistic regression, the relative risk in epidemiological studies, and the partial correlation coefficient in multivariate regression.

It's important to consider the concept of collapsibility when interpreting statistical results and drawing conclusions from research studies, as failure to account for collapsibility can lead to erroneous or misleading conclusions.

Here are a few example of collapsibility concept:

A study investigating the relationship between smoking status (binary variable: smoking or non-smoking) and the risk of developing lung cancer (binary variable: yes or no), while controlling for a potential confounding variable, age (continuous variable).

Suppose we have the following data:

```
# Data
```

```
Smoking <- c(1, 1, 0, 0, 1, 1, 0, 0)
```

```
# Smoking status (1 = smoking, 0 = non-smoking)
```

```
LungCancer <- c(1, 1, 0, 0, 1, 1, 0, 0)
```

```
# Lung cancer status (1 = yes, 0 = no)
```

```
Age <- c(60, 55, 65, 50, 70, 75, 80, 85) # Age in years
```

We can calculate the crude odds ratio (OR) of lung cancer for smoking status without considering age:

```
# Crude Odds Ratio (without considering age)
```

```
crude_OR <- (sum(Smoking == 1 & LungCancer == 1) * sum(Smoking == 0 & LungCancer == 0)) /
```

```
(sum(Smoking == 0 & LungCancer == 1) * sum(Smoking == 1 & LungCancer == 0))
```

Now, let's calculate the adjusted odds ratio (AOR) of lung cancer for smoking status, adjusting for age using logistic regression:

```
# Logistic Regression Model (adjusting for age)
```

```
model <- glm(LungCancer ~ Smoking + Age, family = binomial())
```

```
# Adjusted Odds Ratio (controlling for age)
```

```
adjusted_OR <- exp(coef(model)["Smoking"])
```

```
# print result:
```

```
print(crud_OR)
```

```
print(mdoel)
```

```
print(adjusted_OR)
```

**Output:**

```
> print(crude OR)
```

```
[1] Inf
```

```
> print(model)
```

```
Call: glm(formula = LungCancer ~ Smoking + Age, family = binomial())
```

```
Coefficients:
```

```
(Intercept)      Smoking          Age
```

```
-2.457e+01    4.913e+01    5.897e-15
```

```
Degrees of Freedom: 7 Total (i.e. Null); 5 Residual
```

```
Null Deviance:      11.09
```

```
Residual Deviance: 3.429e-10    AIC: 6
```

```
> print(adjusted OR)
```

```
Smoking
```

```
2.17679e+21
```

In this example, the crude OR is the measure of association without considering age, while the adjusted OR is the measure of association after adjusting for age, taking into account the potential confounding effect of age. If the crude and adjusted ORs are similar, it indicates that the association between smoking and lung cancer is not confounded by age and the measure of association is collapsible. However, if the crude and adjusted ORs are substantially different, it suggests that age is a confounding variable and the association between smoking and lung cancer may be spurious without accounting for age.

\* Note: This is a simplified example for illustrative purposes and may not reflect the complexities of real-world data and statistical analysis. Proper statistical methods, including appropriate adjustment for potential confounding variables, should be applied in actual research studies.