In this crosstabulation table of Titanic data, what evidence indicates a possible association between the variables **Age** and **Survived**?

Table of Age by Survived			
Age	Survived		
Frequency Percent Row Pct Col Pct	no	yes	Total
adult	1438	654	2092
	65.33	29.71	95.05
	68.74	31.26	
	96.51	91.98	
child	52	57	109
	2.36	2.59	4.95
	47.71	52.29	
	3.49	8.02	
Total	1490	711	2201
	67.70	32.30	100.00

- *a.* The frequency statistics indicate that the values of each variable are equally distributed across levels.
- *b.* The row percentages indicate that the distribution of **Survived** changes when the value of **Age** changes.
- *c.* The column percentages indicate that most of the people on the boat were children.

Correct.
To see a possible association, you look at the row percentages. The percent of children who survived (52.29) is much higher than the percent of adults who survived (31.26).

In this PROC FREQ step, write a TABLES statement to create the following frequency and crosstabulation tables for categorical variables in the **entertainment.movies** data set.

- frequency tables for **Type** and **Rating**
- crosstabulation table for **Type** (as the row variable) by **Rating** (as the column variable)

```
proc freq data=entertainment.movies;
   [                    ];
run;
```

The following TABLES statement will create the specified tables:

```
tables Type Rating Type*Rating;
```

In a crosstabulation table request, you specify an asterisk between the names of the variables that you want to appear in the table. The first variable represents the rows, and the second variable represents the columns.

The following crosstabulation table shows Titanic survivors by **Gender**. What is the odds ratio of survival of **females** to **males**?

Table of Gender by Survived			
Gender	Survived		
Frequency Row Pct	no	yes	Total
female	126	344	470
	26.81	73.19	
male	1364	367	1731
	78.80	21.20	
Total	1490	711	2201

- *a.* 2.73
- *b.* 0.27
- *c.* 10.1
- *d.* 1.07

Correct.
The odds of survival for **females** are (344/470) / (126/470) = 2.73. The odds of survival for **males** are (367/1731) / (1364/1731) = 0.27. The odds ratio of survival of **females** to **males** is 2.73/.27 = 10.1. The odds of a **female** surviving are 10 times higher than the odds of a **male** surviving.

Which of the following tends to occur when a sample size decreases? Select all that apply.

- [ ] *a.* The chi-square value increases.
- [x] *b.* The *p*-value increases.
- [ ] *c.* Cramer's V increases.
- [ ] *d.* The odds ratio increases.
- [x] *e.* The width of the CI for the odds ratio increases.

Correct.
When your sample size decreases, your chi-square value decreases, causing both the *p*-value and CI width to increase because the hypothesis test becomes less significant. Cramer’s V statistic and odds ratios are not affected by sample size.

Suppose you're testing for an ordinal association between people's income and their body mass index (BMI). The levels of the **Income** and **BMI** variables are shown in the table below. The Mantel-Haenszel chi-square *p*-value is 0.01 and the Spearman correlation statistic is 0.253. What can you conclude about the association between **Income** and **BMI**?

Variable	Levels
Income	1_low
	2_medium
	3_high
BMI	1_underweight
	2_normal
	3_overweight
	4_obese

- *a.* There is a negative, ordinal association.
- *b.* There is a positive, ordinal association.
- *c.* There is no ordinal association.
- *d.* You can't conclude anything. The results are invalid because the variable values are not ordered correctly.

Correct.
The *p*-value is significant (less than 0.05) so there is an ordinal association. The Spearman correlation statistic is positive, so the ordinal association is positive.

In this PROC FREQ step, write a TABLES statement that displays the crosstabulation of the ordinal variables **Size** and **Severity**. Along with the default output, you want to display the Mantel-Haenszel chi-square test of association and the Spearman correlation statistic with confidence bounds.

```
proc freq data=weather.storms;
    [                    ];
run;
```

The following TABLES statement will create the specified tables:

```
tables Size*Severity / chisq measures cl;
```

Note: The options can appear in any order.

What are the lower and upper bounds for a logit?

- *a.* lower bound=0, upper bound=1

- *b.* lower bound=0, no upper bound
- *c.* no lower bound, no upper bound
- *d.* no lower bound, upper bound=1

Correct.
A probability is bounded by 0 and 1. The logit of the probability transforms the probability into a linear function, which has no lower or upper bounds.

You're modeling the relationship between the variables **Gender** (with the levels *female* and *male*) and **Survived** (with the levels *yes* and *no*). How do you interpret the odds ratio in the output from this PROC LOGISTIC program?

```
proc logistic data=stat1.titanic plots(only)=(effect);
   class Gender (param=ref);
   model Survived (event='yes')=Gender;
run;
```

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
Gender female vs male	10.147	8.027	12.827

- *a.* The odds of females surviving were 10 times the odds of males surviving.
- *b.* The odds of males surviving were 10 times the odds of females surviving.
- *c.* The probability of a female surviving was 10%.
- *d.* Females aboard the Titanic had a 10% survival rate.

Correct.
In the output, the odds ratio of survival for females to males is 10.147. This means that the odds of females surviving were 10 times the odds of males surviving.

The variable **Income** has the values *High*, *Low*, and *Medium*. You've parameterized the variable with reference cell coding using the default reference level. For which value of **Income** do both design variables have the value *0*?

- ○ a. *High*
- ○ b. *Low*
- ◉ c. *Medium*

Correct.
The design variables have the value *0* for the *Medium* level, the default reference level, because it comes last in alphanumeric order.

Suppose you want to fit a logistic regression model to explain the relationship between two variables in the **Titanic** data set: **Age** (which has the values *adult* and *child*) and **Survived** (which has the values *no* and *yes*). To complete this PROC LOGISTIC step, write a CLASS statement (if you think it's necessary) and a MODEL statement that models the probability of survival. Specify reference cell coding and the reference level that PROC LOGISTIC selects by default.

```
proc logistic data=stat1.titanic;
    [                    ];
    [                    ];
run;
```

The following statements correctly complete the PROC LOGISTIC step:

```
class Age (ref='child') / param=ref;

model Survived(event='yes')=Age;
```

Alternatively:

```
class Age (param=ref ref='child');

model Survived(event='yes')=Age;
```

The CLASS statement is necessary because the predictor is categorical. You're specifying the level of **Age** that PROC LOGISTIC will select by default for reference cell coding, so the code will generate the same results with or without the REF= option.

In the MODEL statement, add **Gender** as a main effect and specify the backward elimination method.

```
        proc logistic data=stat1.titanic;
           class Age (param=ref ref='child')
                 Gender (param=ref ref='female')
           model Survived(event='yes')= Age
[                    ]
                 [                    ];
        run;
```

The completed MODEL statement is shown below:

```
model Survived(event='yes')=Age Gender
      / selection=backward;
```

In the MODEL statement, specify the predictor variables **Age**, **Gender**, and **Class**. Indicate that you want to include an interaction between **Gender** and **Class** in addition to the three main effects.

```
        proc logistic data=stat1.titanic;
           class Age (param=ref ref='child')
                 Gender (param=ref ref='female')
```

```
                    Class (param=ref ref='crew');
              model Survived(event='yes')=
                                          ;
                    / selection=backward;
           run;
```

The completed MODEL statement is shown below. You place a bar operator only between the variables that you want to include in an interaction.

```
model Survived(event='yes')=Age Gender | Class
      / selection=backward;
```

Alternate Solution: The bar operator saves typing, but is not required for identifying interactions.

```
model Survived(event=’yes’)=Age Gender Class
Gender*Class
      / selection=backward;
```