Print

Practice: Using the Binary Logistic Regression Task to Perform Backward Elimination; Using PROC PLM to Generate Predictions

The insurance company wants to model the relationship between three of a car's characteristics, weight, size, and region of manufacture, and its safety rating. Use the Binary Logistic Regression task and backward elimination. Start with a model using only main effects. The **safety** data set contains the data about vehicle safety.

1. Use the Binary Logistic Regression task to fit a multiple logistic regression model with **Unsafe** as the response variable and **Weight**, **Size**, and **Region** as the predictor variables.
    1. Use the EVENT= option to model the probability of Below Average safety scores.
    2. Specify **Size** and **Region** as classification variables and use reference cell coding. Specify *1* (small cars) as the reference level for **Size** and *Asia* as the reference level for **Region**.
    3. Add a UNITS statement with -1 as the unit for **Weight** so that you can see the odds ratio for lighter cars over heavier cars.
    4. Add a STORE statement to save the analysis results as **isSafe**.
    5. Request any relevant plots.
    6. Run the task and view the results.

    1. In the Navigation pane, select **Tasks and Utilities**.
    2. Expand **Tasks**.
    3. Expand **Statistics** and open the **Binary Logistic Regression** task.
    4. Select the **stat1.safety** table.
    5. Assign **Unsafe** to the Response role, and use the Event of interest drop-down list to specify **1**.
    6. Assign **Size** and **Region** to the Classification variables role.
    7. Expand the **Parameterization of Effects** property and use the Coding drop-down list to select **Reference coding**.
    8. Assign **Weight** to the Continuous variables role.
    9. On the MODEL tab, verify that **Main effects model** is selected.
    10. On the SELECTION tab, use the Selection method drop-down list to choose **Backward elimination**.
    11. On the OPTIONS tab, in the Select statistics to display drop-down list, select **Default and additional statistics**.
    12. Expand the **Parameter Estimates** property. In the Confidence intervals for odds ratios drop-down list, select **Based on profile likelihood**.
    13. Expand **PLOTS**, and in the Select plots to display drop-down list, select **Default and additional plots**.
    14. Select **Effect plot** and **Odds ratio plot**.
    15. Modify the code to specify specific levels of each class variable to use as reference levels. On the CODE tab, click the **Edit SAS code** icon.
    16. In the CLASS statement, add the options **(REF='1')** immediately after **Size** and **(REF='Asia')** immediately after **Region**.
    17. Add the statement **units Weight= -1;** after the MODEL statement.
    18. Add the statement **store isSafe;** after the UNITS statement.
    19. Click **Run**.

    Here are the results.

2. Which terms appear in the final model?

    Only **Size** appears in the final model.

3. If you compare these results with those from the previous practice (a model fit with only one variable, **Region**), do you think that this is a better model?

   Comparing the model fit statistics, you see that the AIC (92.629) and SC (100.322) are both smaller in the model fit by the backward elimination method, 119.854 and 124.982, respectively. This indicates that the **Size**-only model is doing better than the **Region**-only model.

   Using the c statistics, you can also see improvement beyond the **Region**-only model, that is, 0.818 in this model compared wtih 0.598 in the previous model.

4. Using the final model that was chosen by backward elimination, and using the STORE statement, generate predictive probabilities for the cars in the following DATA step:

```
data checkSafety;
   length Region $9.;
   input Weight Size Region $ 5-13;
   datalines;
   4 1 N America
   3 1 Asia
   5 3 Asia
   5 2 N America
         ;
run;
```

```
proc plm restore=isSafe;
   score data=checkSafety out=scored_cars / ILINK;
   title 'Safety Predictions using PROC PLM';
run;

proc print data=scored_cars;
run;
```

   Here are the results.

Hide Solution