# Bank Revenues

## Multiple Linear Regression with Transformation

From *Building Better Models With JMP Pro*, Chapter 4, SAS Press (2015). Grayson, Gardner and Stephens.

# Bank Revenues

## Multiple Linear Regression with Transformation

Key ideas: The log transformation, stepwise regression, regression assumptions, residuals, Cook's D, interpreting model coefficients, singularity, Prediction Profiler, inverse transformations.

### Background

A bank wants to understand how customer banking habits contribute to revenues and profitability. The bank has customer age and bank account information, e.g., whether the customer has a savings account, whether the customer has received bank loans, and other indicators of account activity.

### The Task

We want to build a model that allows the bank to predict profitability for a given customer. A surrogate for customer profitability available in our data set is the **Total Revenue** a customer generates through their accounts and transactions. The resulting model will be used to forecast bank revenues and guide the bank in future marketing campaigns.

### The Data BankRevenue.jmp

The data set contains information on 7,420 bank customers:

**Rev_Total** Total revenue generated by the customer over a 6-month period.

**Bal_Tota** Total of all account balances, across all accounts held by the customer.

**Offer** An indicator of whether the customer has received a special promotional offer in the previous one-month period. Offer=1 if the offer was received, Offer=0 if it was not.

**AGE** The customer's age.

**CHQ** Indicator of debit card account activity. CHQ=0 is low (or zero) account activity, CHQ=1 is greater account activity.

**CARD** Indicator of credit card account activity. CARD=0 is low or zero account activity, CARD=1 is greater account activity.

**SAV1** Indicator of primary savings account activity. SAV1=0 is low or zero account activity, SAV1=1 is greater activity.

**LOAN** Indicator of personal loan account activity. LOAN=0 is low or zero account activity, LOAN=1 is greater activity.

**MORT** Indicator of mortgage account tier. MORT=0 is lower tier and less important to the bank's portfolio. MORT=1 is higher tier and indicates the account is more important to the bank's portfolio.

**INSUR** Indicator of insurance account activity. INSUR=0 is low or zero account activity, INSUR=1 is greater activity.

**PENS** Indicator or retirement savings (pension) account tier. PENS=0 is lower balance and less important to bank's portfolio. PENS=1 is higher tier and of more importance to the bank's portfolio.

**Check** Indicator of checking account activity. Check=0 is low or zero account activity, Check=1 is greater activity.

**CD** Indicator of certificate of deposit account tier. CD=0 is lower tier and of less importance to the bank's portfolio. CD=1 is higher tier and of more importance to the bank's portfolio.

**MM** Indicator of money market account activity. MM=0 is low or zero account activity, MM=1 is greater activity.

**Savings** Indicator of savings accounts (other than primary) activity. Savings=0 is low or zero account activity, Savings=1 is greater activity.

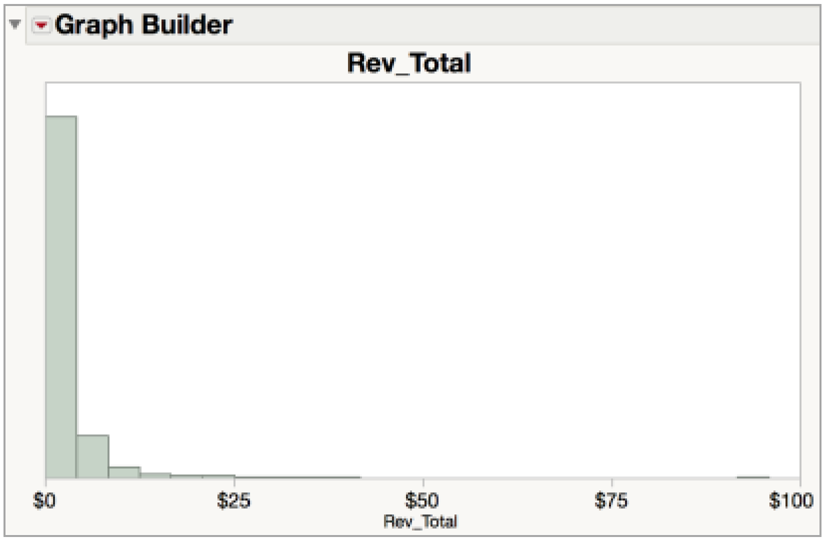
**AccountAge** Number of years as a customer of the bank.

## Prepare for Modeling

We begin by looking at the variable of interest**,** total revenue (**Rev_Total**) using **Graph > Graph Builder** (drag **Rev_Total** to the **X** zone, then click on the histogram icon above the graph frame). **Rev_Total** is highly skewed—a result that  is fairly typical of financial data (Exhibit 1).

> ***Note**: To explore the underlying shape of the distribution, select the **Grabber** (hand) tool from your toolbar, click on the graph (and hold) and drag the hand up and down. This changes the binning of values in the histogram and allows you to better see the details of the distribution.*

**Exhibit 1** Distribution of Total Revenue



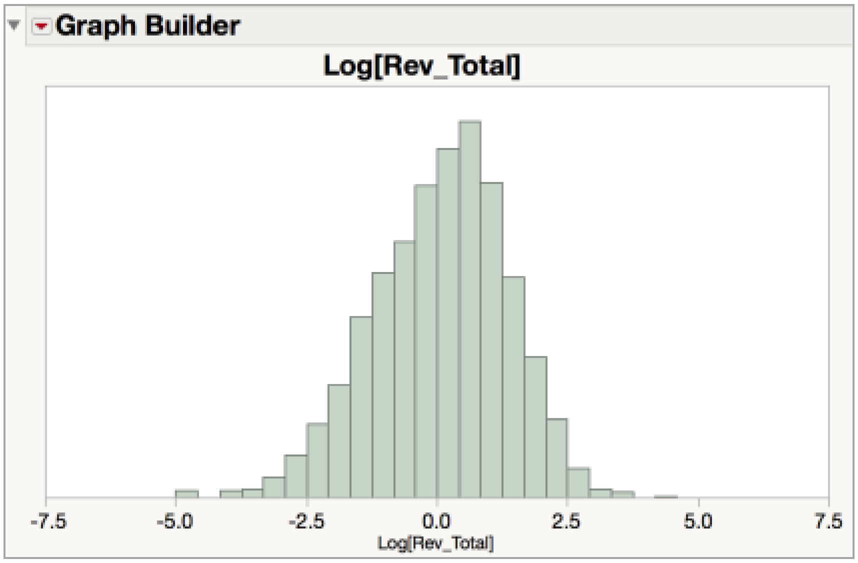
In regression situations, highly skewed data can result in a poorly fitting model. A transformation that can often be used to normalize highly skewed data, when all of the values are positive, is a log (natural logarithm) transformation (see Ramsey and Shafer, 2002, page 68).

We apply a log transformation to the **Rev_Total** variable directly in the **Graph Builder** and reexamine the distribution (Exhibit 2). To apply this transformation, right-click on the variable in the variable selection list and select **Transform > Log**. Then, to save the transformation to the data table, right-click on **Log(Rev_Total)** and select **Add to Data Table**.

This transformation gives us a much less skewed and more symmetric distribution; as such, we use **Log(Rev_Total)** for the rest of our analysis.

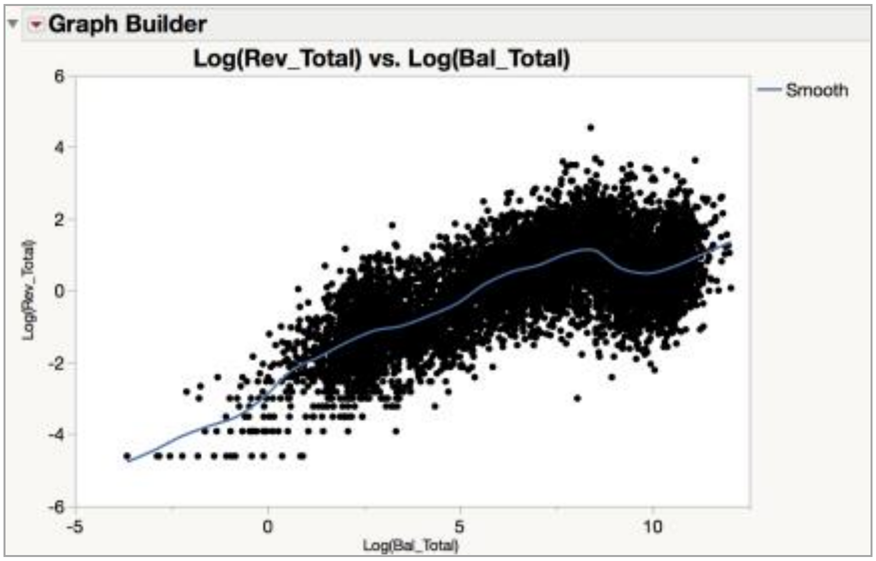
A similar examination of the total account balance (**Bal_Total**), which also has a skewed distribution, leads to the use of **Log(Bal_Total)** in our analyses.

**Exhibit 2** Transformed Total Revenue Using Log Transformation



The relationship between the log total revenue and log total account balance is shown in the scatterplot in Exhibit 3 (using **Graph Builder**, drag **Log(Rev_Total)** to the **Y** zone and drag **Log(Bal_Total)** to the **X** zone). The relationship appears to be nearly linear at lower account balances; higher account balances generally have higher revenues. This relationship, however, seems to change at higher account balances.

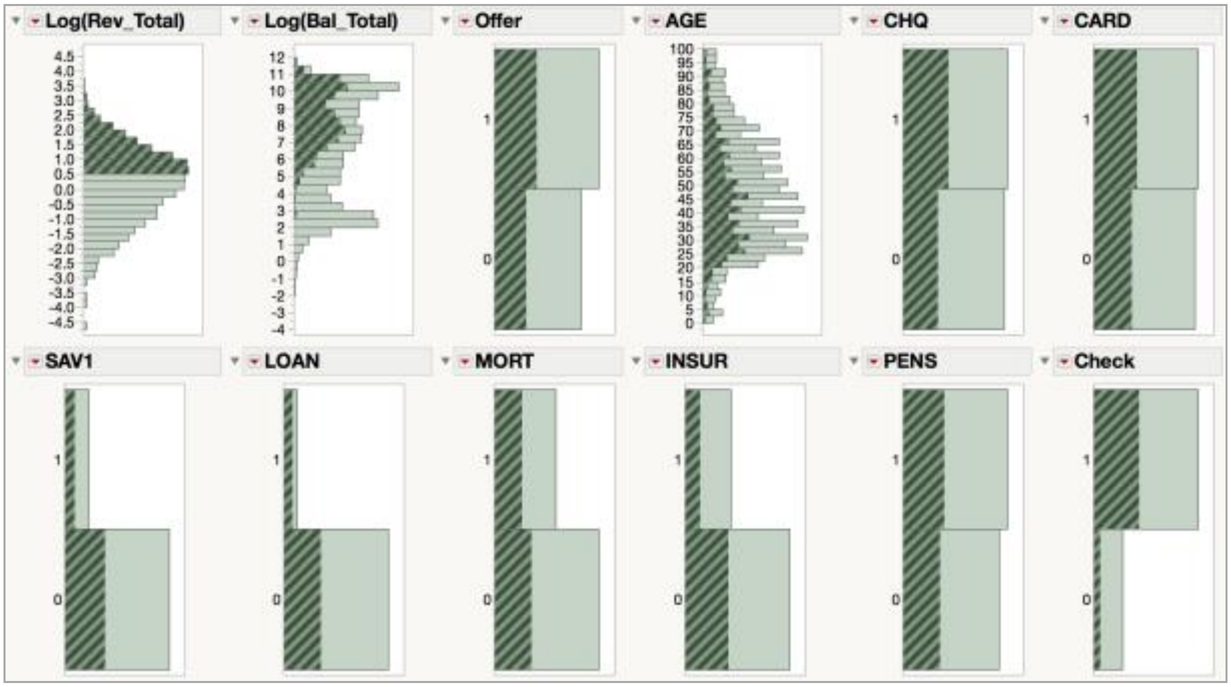
**Exhibit 3** Relationship between **Log(Rev_Total)** and **Log(Bal_Total)**



We now examine the other variables. We can see their distributions as well as their relationship to **Log(Rev_Total)**. Many of the variables are categorical, with two-levels. Higher revenue values are selected in Exhibit 4 (using the **Brush** tool on the toolbar); we can see this selection across the other variables in our data set. Other than total account balance, **Log(Bal_Total)**, there is no variable that stands out as being strongly related to revenue. The **Arrange In Rows** option under the top red triangle in the **Distribution** platform was used to generate Exhibit 4; not all variables are displayed.

Note that other graphical and analytic tools can be used to understand the data and explore potential relationships, such as **Fit Y by X** and **Graph Builder**. In addition, the **Data Filter** (under the **Rows** menu) and **Column Switcher** (under the top **red triangle > Script** in any output window) are dynamic tools that allow you to dive deeper into your data to explore variables of interest and investigate potential relationships. These tools, in addition to the **Columns Viewer** (under the **Cols** menu), can also be used to identify potential issues with data quality that will need to be addressed prior to modeling. We encourage you to explore the data using these tools on your own.

**Exhibit 4** Relationships between Transformed Variables and Other Variables

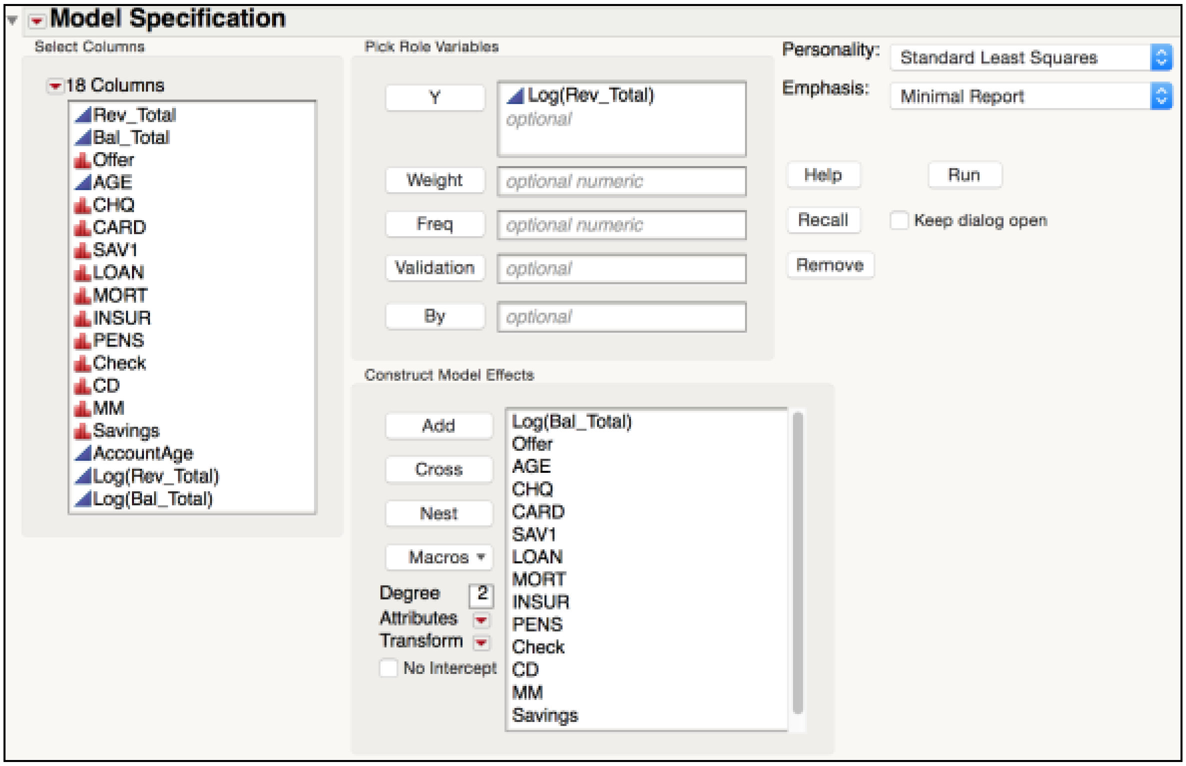


***Note**: Within JMP there are a number of preferences that can be set (under **File > Preferences** or **JMP > Preferences** on a Mac), and all JMP output is customizable with your mouse and keystrokes. Going forward, we periodically resize graphs and change axis scaling to better fit content on the page, and change marker sizes or colors to improve interpretability. We also turn off shaded table headings in output to provide a cleaner display (within **Preferences, Styles > Report Tables**).*

## Build the Model

We now build a regression model to predict **Log(Rev_Total)** using **Fit Model** (see Exhibit 5). The model effects are **Log(Bal_Total)** and the remaining 14 are potential predictor variables (**Offer** through **AccountAge**).

**Exhibit 5** Fit Model Dialog



Model Specification

Select Columns

18 Columns

Rev_Total
Bal_Total
Offer
AGE
CHQ
CARD
SAV1
LOAN
MORT
INSUR
PENS
Check
CD
MM
Savings
AccountAge
Log(Rev_Total)
Log(Bal_Total)

Pick Role Variables

Y: Log(Rev_Total)

Personality: Standard Least Squares

Emphasis: Minimal Report

Construct Model Effects

Degree: 2

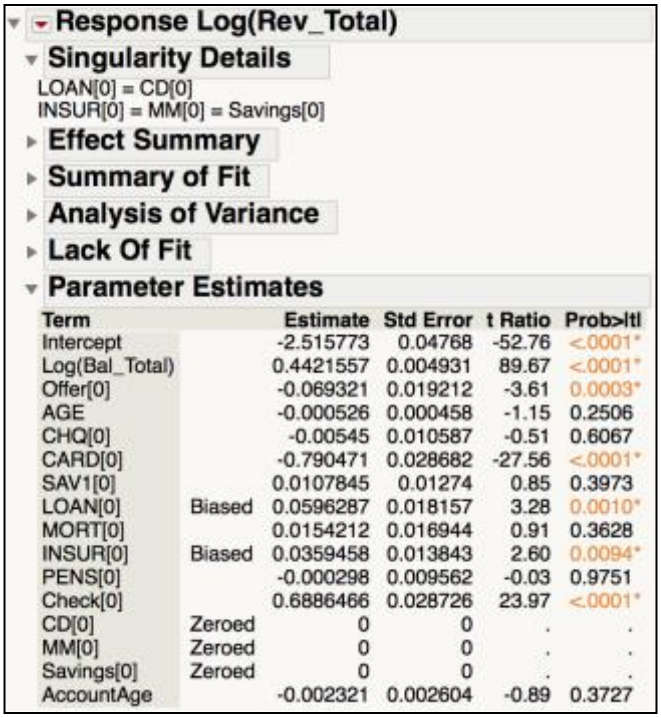
Log(Bal_Total)
Offer
AGE
CHQ
CARD
SAV1
LOAN
MORT
INSUR
PENS
Check
CD
MM
Savings

There are some immediate signs of trouble when we run this model (Exhibit 6). At the top of the **Fit Least Squares** window, we see some unexpected output, *Singularity Details*. This means that there are linear dependencies between predictor variables. The first row of this table, **LOAN[0] = CD[0]**, indicates that JMP cannot identify the difference between these two variables, **LOAN** and **CD**. The second line indicates that JMP cannot identify the difference between **INSUR**, **MM** and **Savings**.

The cause of this problem is illustrated in the **Distribution** output in Exhibit 7. The distributions of these three variables are identical. Every time **LOAN** = 1 (a customer has high loan activity), **MM** and **Savings** are also 1 (money market and savings activity are also high). Variables within each grouping are completely redundant to one another!

The result of this problem is seen in the parameter estimates table in Exhibit 6. JMP cannot estimate all of these coefficients, thus indicating that estimates for **LOAN** and **INSUR** are *biased*, and the estimates for **CD**, **MM** and **Savings** are *zeroed*. JMP can estimate some of the parameters for redundant variables (these estimates are biased), but not all (these estimates are zeroed). Whether variables appear as biased or zeroed depends entirely on the order in which they were entered into the model, i.e., those entered first into the model are displayed as biased.

**Exhibit 6** Fit Least Squares with Singularity



Response Log(Rev_Total)

Singularity Details

LOAN[0] = CD[0]
INSUR[0] = MM[0] = Savings[0]

Effect Summary

Summary of Fit

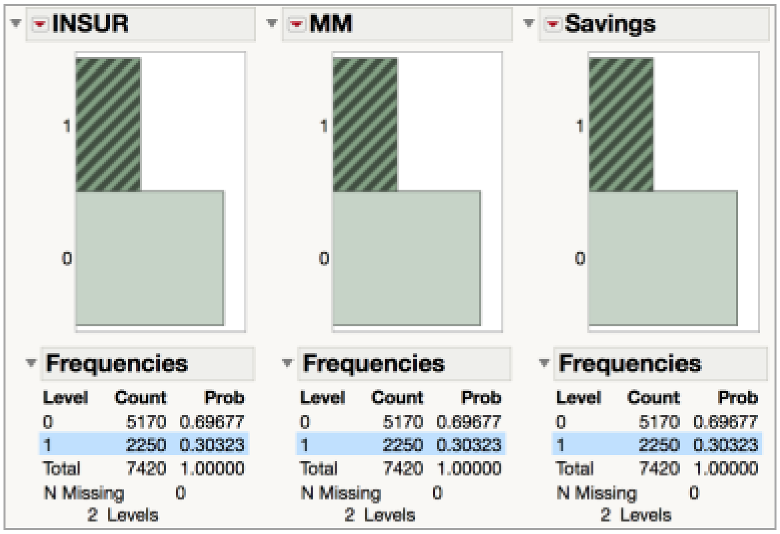
Analysis of Variance

Lack Of Fit

Parameter Estimates

| Term | | Estimate | Std Error | t Ratio | Prob>\|t\| |
|---|---|---|---|---|---|
| Intercept | | -2.515773 | 0.04768 | -52.76 | <.0001* |
| Log(Bal_Total) | | 0.4421557 | 0.004931 | 89.67 | <.0001* |
| Offer[0] | | -0.069321 | 0.019212 | -3.61 | 0.0003* |
| AGE | | -0.000526 | 0.000458 | -1.15 | 0.2506 |
| CHQ[0] | | -0.00545 | 0.010587 | -0.51 | 0.6067 |
| CARD[0] | | -0.790471 | 0.028682 | -27.56 | <.0001* |
| SAV1[0] | | 0.0107845 | 0.01274 | 0.85 | 0.3973 |
| LOAN[0] | Biased | 0.0596287 | 0.018157 | 3.28 | 0.0010* |
| MORT[0] | | 0.0154212 | 0.016944 | 0.91 | 0.3628 |
| INSUR[0] | Biased | 0.0359458 | 0.013843 | 2.60 | 0.0094* |
| PENS[0] | | -0.000298 | 0.009562 | -0.03 | 0.9751 |
| Check[0] | | 0.6886466 | 0.028726 | 23.97 | <.0001* |
| CD[0] | Zeroed | 0 | 0 | . | . |
| MM[0] | Zeroed | 0 | 0 | . | . |
| Savings[0] | Zeroed | 0 | 0 | . | . |
| AccountAge | | -0.002321 | 0.002604 | -0.89 | 0.3727 |

**Exhibit 7** Distributions of INSUR, MM, and Savings



INSUR – Frequencies

| Level | Count | Prob |
|---|---|---|
| 0 | 5170 | 0.69677 |
| 1 | 2250 | 0.30323 |
| Total | 7420 | 1.00000 |
| N Missing | 0 | |
| 2 Levels | | |

MM – Frequencies

| Level | Count | Prob |
|---|---|---|
| 0 | 5170 | 0.69677 |
| 1 | 2250 | 0.30323 |
| Total | 7420 | 1.00000 |
| N Missing | 0 | |
| 2 Levels | | |

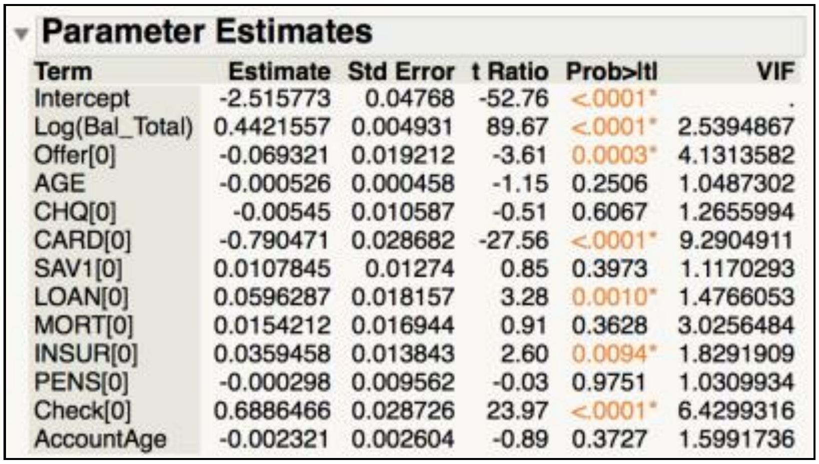
Savings – Frequencies

| Level | Count | Prob |
|---|---|---|
| 0 | 5170 | 0.69677 |
| 1 | 2250 | 0.30323 |
| Total | 7420 | 1.00000 |
| N Missing | 0 | |
| 2 Levels | | |

We refit the model without redundant variables. As of JMP 12, this can be done using the **Remove** button at the bottom of the **Effect Summary** table. We keep **LOAN** (and eliminate **CD**) and **INSUR** (eliminating **MM** and **Savings**). Note that this was an arbitrary decision: subject matter knowledge should guide the decision as to which redundant variables to remove (and which variables to keep in the model).

As we remove each variable (or term), the **Singularity Details** table updates, along with all other statistical outputs. JMP is now able to estimate coefficients for each of the parameters (Exhibit 8).

**Exhibit 8** Fit Least Squares Parameter Estimates without Singularity, Showing VIFs



Parameter Estimates

| Term | Estimate | Std Error | t Ratio | Prob>\|t\| | VIF |
|---|---|---|---|---|---|
| Intercept | -2.515773 | 0.04768 | -52.76 | <.0001* | . |
| Log(Bal_Total) | 0.4421557 | 0.004931 | 89.67 | <.0001* | 2.5394867 |
| Offer[0] | -0.069321 | 0.019212 | -3.61 | 0.0003* | 4.1313582 |
| AGE | -0.000526 | 0.000458 | -1.15 | 0.2506 | 1.0487302 |
| CHQ[0] | -0.00545 | 0.010587 | -0.51 | 0.6067 | 1.2655994 |
| CARD[0] | -0.790471 | 0.028682 | -27.56 | <.0001* | 9.2904911 |
| SAV1[0] | 0.0107845 | 0.01274 | 0.85 | 0.3973 | 1.1170293 |
| LOAN[0] | 0.0596287 | 0.018157 | 3.28 | 0.0010* | 1.4766053 |
| MORT[0] | 0.0154212 | 0.016944 | 0.91 | 0.3628 | 3.0256484 |
| INSUR[0] | 0.0359458 | 0.013843 | 2.60 | 0.0094* | 1.8291909 |
| PENS[0] | -0.000298 | 0.009562 | -0.03 | 0.9751 | 1.0309934 |
| Check[0] | 0.6886466 | 0.028726 | 23.97 | <.0001* | 6.4299316 |
| AccountAge | -0.002321 | 0.002604 | -0.89 | 0.3727 | 1.5991736 |

### A Bit About Multicollinearity

Before proceeding, we check to make sure that there is no substantial correlation, or *multicollinearity*, between our predictors. When two or more predictors are correlated with one another, it is difficult to determine which of the correlated predictors are most important. In addition, model coefficients and standard errors may be inflated. A statistical measure of the degree of multicollinearity is VIF, or *Variance Inflation Factor*. As a general rule of thumb, a VIF for a predictor greater than 10 indicates that multicollinearity is a problem that should be addressed (Neter, 1996). In some cases, eliminating one of the correlated terms from the model can resolve the issue. In severe cases, other techniques may be required[1].

To display VIFs, right-click on the **Parameter Estimates** table and select **Columns > VIF**. A quick check of the VIFs indicates that multicollinearity is not a serious issue; the largest VIF is 9.29 (see the last column in Exhibit 8).

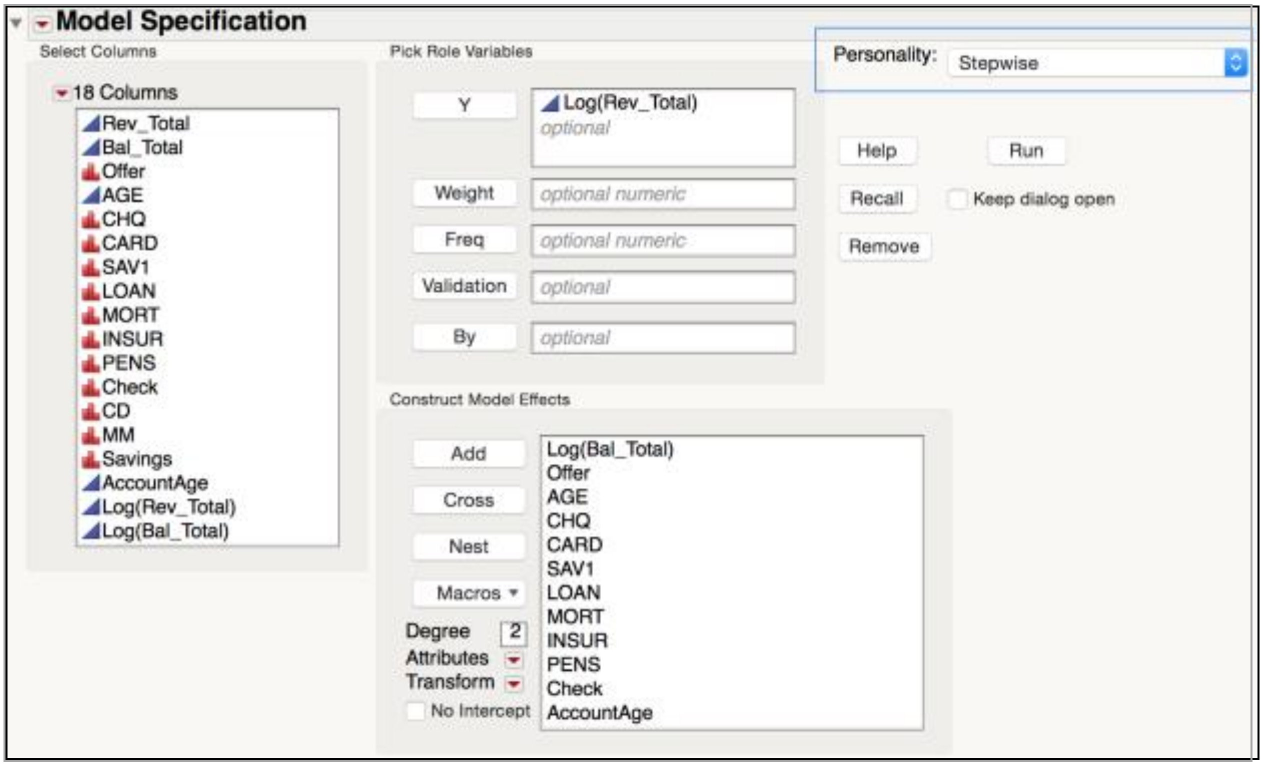
### Fitting a Model Using Stepwise Regression

Since we have 12 remaining potential predictor variables, we use stepwise regression to help with variable selection. We return to the Fit Model platform and select **Stepwise** from the **Personality** list (to return to the Fit Model dialog, click on the top red triangle in the Least Squares output window and select **Model Dialog**).

We again use **Log(Rev_Total)** as our Y variable, and use the 12 remaining predictor variables as model effects (see Exhibit 9). Click **Run** to launch the Stepwise platform.

[1] See Building Better Models with JMP Pro, Chapter 4, for additional information on multicollinearity and VIF, and techniques for addressing severe multicollinearity.

**Exhibit 9** Fit Model Dialog with Stepwise Personality



Model Specification

Select Columns

18 Columns

Rev_Total
Bal_Total
Offer
AGE
CHQ
CARD
SAV1
LOAN
MORT
INSUR
PENS
Check
CD
MM
Savings
AccountAge
Log(Rev_Total)
Log(Bal_Total)

Pick Role Variables

Y: Log(Rev_Total)
Weight: optional numeric
Freq: optional numeric
Validation: optional
By: optional

Personality: Stepwise

□ Keep dialog open

Construct Model Effects

Log(Bal_Total)
Offer
AGE
CHQ
CARD
SAV1
LOAN
MORT
INSUR
PENS
Check
AccountAge

Degree 2

□ No Intercept

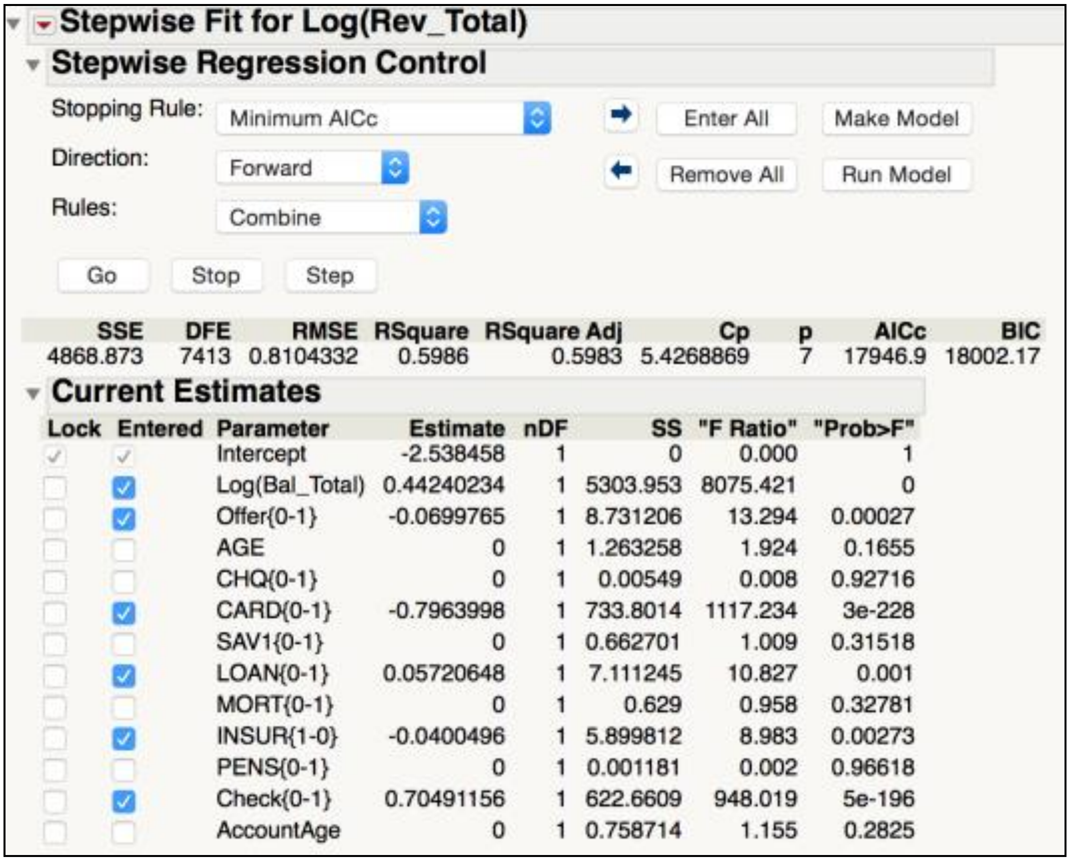
Stepwise regression provides a number of stopping rules for selecting the best subset of variables for our model. The default rule is **Minimum BIC**, or minimum *Bayesian Information Criterion*. The **Direction**, which is set to **Forward** by default, indicates that variables will be added to the model one at a time. After you click **Go**, the model with the smallest BIC statistic is selected.

Another common rule, which works in a similar manner, is **Minimum AICc** (*Akaike's Information Criterion,* with a correction for small sample sizes). Both of these rules (Minimum BIC and Minimum AICc) attempt to explain the relationship between predictors and a response without building models that are overly complex in terms of the number of predictors. Since different criteria are used to determine when to stop adding terms to the model, these stopping rules may lead to different "best" models (see Burnham, 2002).

For this example, we use the **Minimum AICc** stopping rule (we revisit the Minimum BIC stopping rule in an exercise). After clicking **Go**, the model with the smallest AICc statistic is selected.

Stepwise selects six variables for the model. These are checked under **Current Estimates** in Exhibit 10. Note that when using AICc (or BIC), resulting models may include terms that are not significant. This is because both AICc and BIC build models based on *important effects* (i.e., effects that explain the relationship between the response and predictors) rather than searching for *significant effects* (see Burhnam, 2002). However, in this example, all six selected variables have low *p*-values.

**Exhibit 10** Stepwise Regression Dialog with Model Variables Selected



Stepwise Fit for Log(Rev_Total)

Stepwise Regression Control

Stopping Rule: Minimum AICc

Direction: Forward

Rules: Combine

| SSE | DFE | RMSE | RSquare | RSquare Adj | Cp | p | AICc | BIC |
|---|---|---|---|---|---|---|---|---|
| 4868.873 | 7413 | 0.8104332 | 0.5986 | 0.5983 | 5.4268869 | 7 | 17946.9 | 18002.17 |

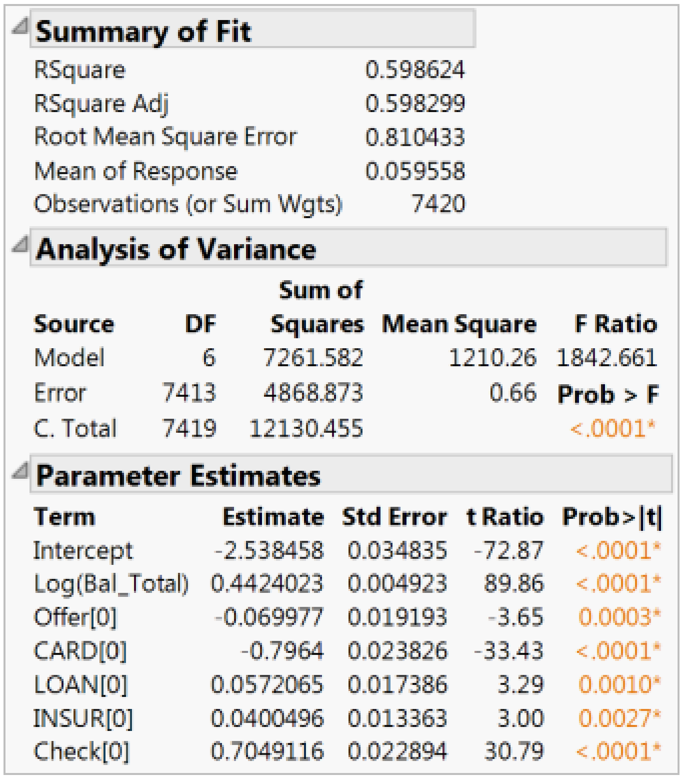
Current Estimates

| Lock | Entered | Parameter | Estimate | nDF | SS | "F Ratio" | "Prob>F" |
|---|---|---|---|---|---|---|---|
| ✓ | ✓ | Intercept | -2.538458 | 1 | 0 | 0.000 | 1 |
| □ | ☑ | Log(Bal_Total) | 0.44240234 | 1 | 5303.953 | 8075.421 | 0 |
| □ | ☑ | Offer{0-1} | -0.0699765 | 1 | 8.731206 | 13.294 | 0.00027 |
| □ | □ | AGE | 0 | 1 | 1.263258 | 1.924 | 0.1655 |
| □ | □ | CHQ{0-1} | 0 | 1 | 0.00549 | 0.008 | 0.92716 |
| □ | ☑ | CARD{0-1} | -0.7963998 | 1 | 733.8014 | 1117.234 | 3e-228 |
| □ | □ | SAV1{0-1} | 0 | 1 | 0.662701 | 1.009 | 0.31518 |
| □ | ☑ | LOAN{0-1} | 0.05720648 | 1 | 7.111245 | 10.827 | 0.001 |
| □ | □ | MORT{0-1} | 0 | 1 | 0.629 | 0.958 | 0.32781 |
| □ | ☑ | INSUR{1-0} | -0.0400496 | 1 | 5.899812 | 8.983 | 0.00273 |
| □ | □ | PENS{0-1} | 0 | 1 | 0.001181 | 0.002 | 0.96618 |
| □ | ☑ | Check{0-1} | 0.70491156 | 1 | 622.6609 | 948.019 | 5e-196 |
| □ | □ | AccountAge | 0 | 1 | 0.758714 | 1.155 | 0.2825 |

We now run this model (click **Run Model**) and explore the results (see Exhibit 11).

As expected, the overall model is significant with a *p*-value < .0001, as are all of the terms in the model. The R Square is 0.5986, indicating that our model explains nearly 60% of variation in the response.

**Exhibit 11** Model Results, Reduced Model



**Summary of Fit**

| | |
|---|---|
| RSquare | 0.598624 |
| RSquare Adj | 0.598299 |
| Root Mean Square Error | 0.810433 |
| Mean of Response | 0.059558 |
| Observations (or Sum Wgts) | 7420 |

**Analysis of Variance**

| | | Sum of | | |
|---|---|---|---|---|
| **Source** | **DF** | **Squares** | **Mean Square** | **F Ratio** |
| Model | 6 | 7261.582 | 1210.26 | 1842.661 |
| Error | 7413 | 4868.873 | 0.66 | **Prob > F** |
| C. Total | 7419 | 12130.455 | | <.0001* |

**Parameter Estimates**

| **Term** | **Estimate** | **Std Error** | **t Ratio** | **Prob>\|t\|** |
|---|---|---|---|---|
| Intercept | -2.538458 | 0.034835 | -72.87 | <.0001* |
| Log(Bal_Total) | 0.4424023 | 0.004923 | 89.86 | <.0001* |
| Offer[0] | -0.069977 | 0.019193 | -3.65 | 0.0003* |
| CARD[0] | -0.7964 | 0.023826 | -33.43 | <.0001* |
| LOAN[0] | 0.0572065 | 0.017386 | 3.29 | 0.0010* |
| INSUR[0] | 0.0400496 | 0.013363 | 3.00 | 0.0027* |
| Check[0] | 0.7049116 | 0.022894 | 30.79 | <.0001* |

Before interpreting the results of the regression model, we check that the regression assumptions are met; namely, that our model errors are independent, have equal variance, and are normally distributed. Another key assumption is that the relationship between our response and the predictors is linear (i.e., that there isn't an underlying non-linear relationship that we've missed).

### Checking Model Assumptions

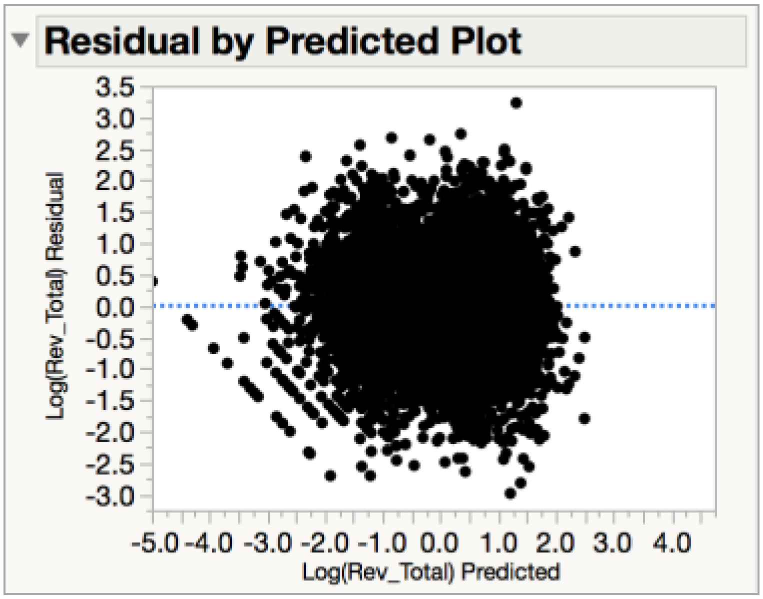
Variation in the *residuals* (which is another word for the errors) shows us variation in the response that is not explained by the model that we have fit. Plots of residuals can be used to verify that our assumptions about model errors were correct. If our model assumptions are met, points should be randomly scattered above and below the center line (zero), with no obvious pattern (just a cloud of seemingly random points). An obvious pattern, such as curvature, would be an indication that our current model is inadequately explaining the relationship between the predictors and response, that certain observations are influencing our model, or that something is missing from the model.

Since our data have been amassed from over 7,400 different customers, we have some assurance that the independence assumption is met. The residual versus predicted value plot (Exhibit 12) shows some diagonal striations in the lower left corner (under the red triangle for the response, select **Row Diagnostics > Plot Residual by Predicted**).

To explore these values, we use the **lasso tool** on the toolbar to select the observations (select the lasso tool, then draw a circle around the points), go to the data table, and then use the **F7** function key to scroll through selected observations in the data table. The first strip on the left corresponds to revenue $0.01, while the second corresponds to revenue $0.02. This is the result of the fact that there are many customers who generate little, if any, revenue for the bank.

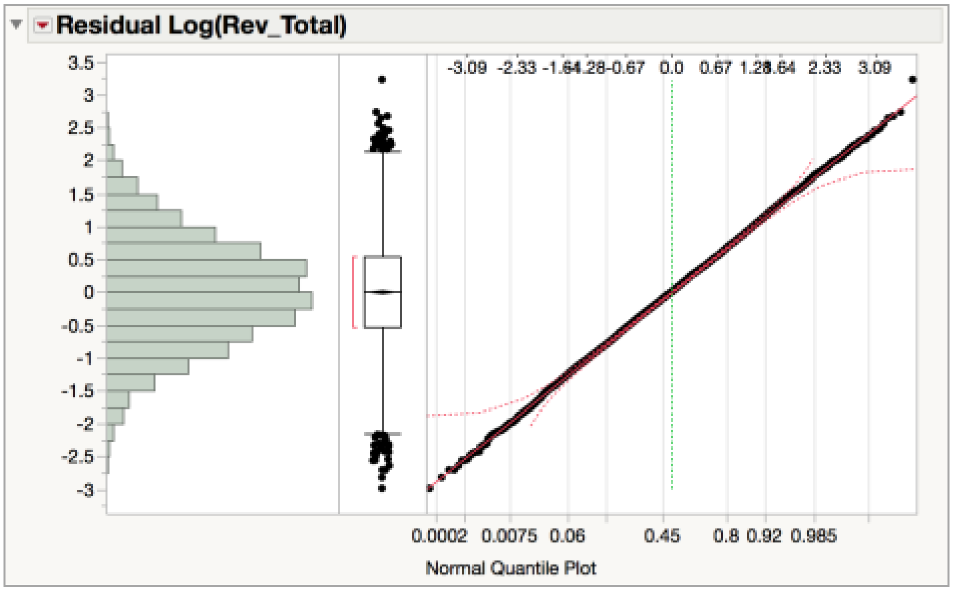
There appear to be two clusters or groupings of points (above and below **Log(Rev_Total) Predicted** = 0). Otherwise, the residual plot shows no unusual patterns and points appear randomly scattered on either side of the center line (zero).

**Exhibit 12** Residual versus Predicteds



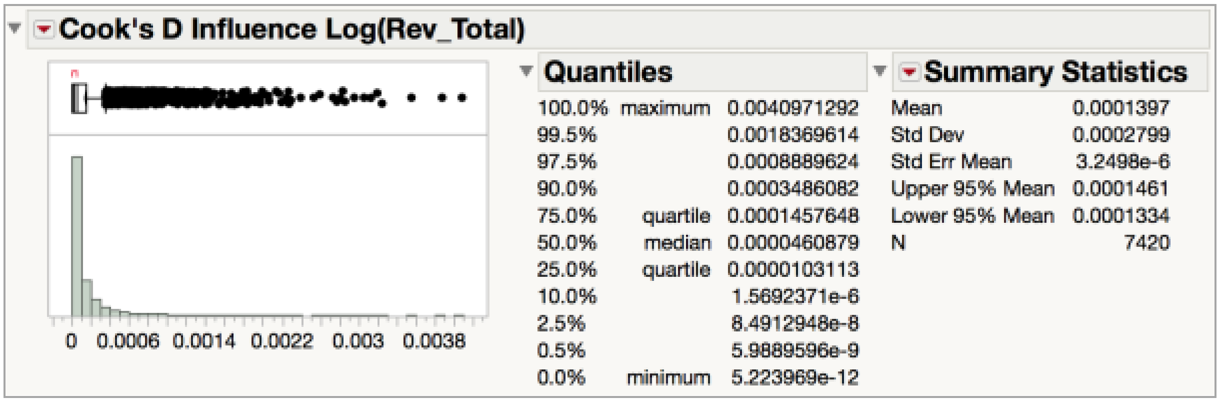
For further exploration of the regression assumptions, we save residuals to the data table. We also save Cook's D values (under the red triangle, select **Save Columns > Residuals** and **Cook's D** – these values are saved in two new columns in the data table)**.** We use the **Distribution** platform to generate a histogram of the residuals, and then use a red triangle option to add a normal quantile plot (Exhibit 13). These plots provide evidence that the normality assumption has been met.

**Exhibit 13** Distribution of Residuals with Normal Quantile Plot



We also see (in Exhibit 13) that there are no serious outliers. A quick peek at Cook's D values, again using the **Distribution** platform, confirms that there are no highly influential observations (see Exhibit 14). A high Cook's D value (>1) for a particular observation indicates that  model predictions with and without that observation are different. All values are very low (none of the values are >1). As such, we can conclude that no single point is exerting too much influence over our model.

**Exhibit 14**  Checking Assumptions with Cook's D



Cook's D Influence Log(Rev_Total)

Quantiles

| | | |
|---|---|---|
| 100.0% | maximum | 0.0040971292 |
| 99.5% | | 0.0018369614 |
| 97.5% | | 0.0008889624 |
| 90.0% | | 0.0003486082 |
| 75.0% | quartile | 0.0001457648 |
| 50.0% | median | 0.0000460879 |
| 25.0% | quartile | 0.0000103113 |
| 10.0% | | 1.5692371e-6 |
| 2.5% | | 8.4912948e-8 |
| 0.5% | | 5.9889596e-9 |
| 0.0% | minimum | 5.223969e-12 |

Summary Statistics

| | |
|---|---|
| Mean | 0.0001397 |
| Std Dev | 0.0002799 |
| Std Err Mean | 3.2498e-6 |
| Upper 95% Mean | 0.0001461 |
| Lower 95% Mean | 0.0001334 |
| N | 7420 |

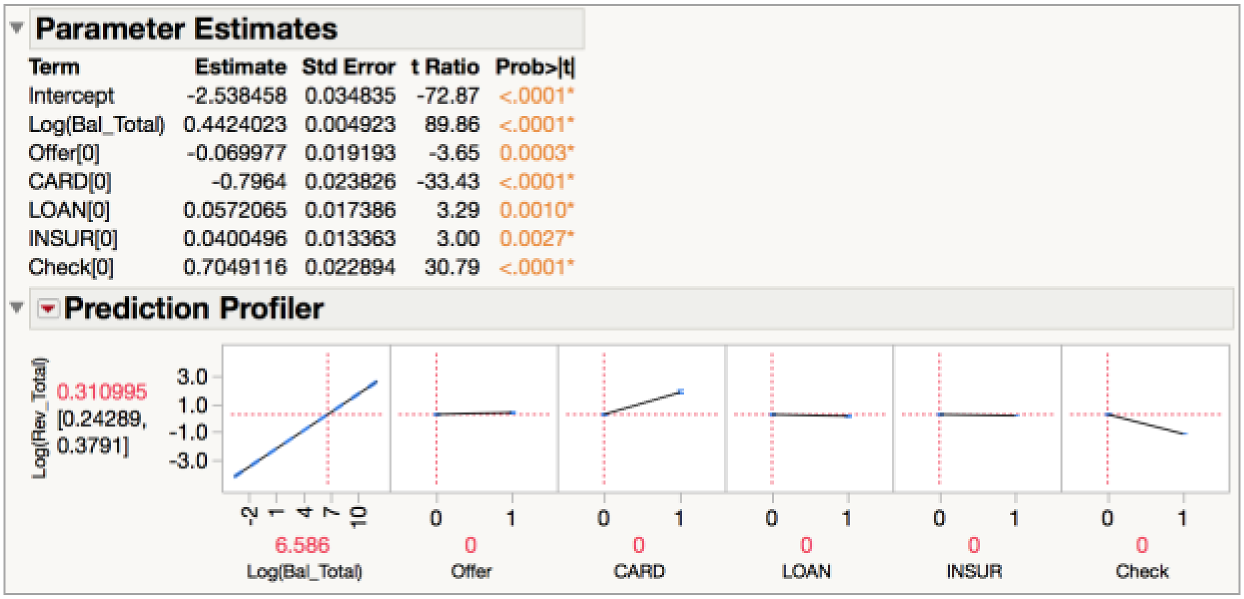
**Interpreting Our Regression Model**

After investigating residuals and looking at Cook's D values, we have confidence that the regression assumptions have been satisfied. Our final model, shown in Exhibit 15, includes the following variables:

- The total account balance (**Log(Bal_Total)**)
- Whether the customer received a promotional offer (**Offer**)
- Credit card activity (**CARD**)
- Personal loan account activity (**LOAN**)
- Insurance account activity (**INSUR**)
- Checking account activity (**Check**)

All significant variables except **Log(Bal_Total)** are binary categorical variables. For the continuous predictor, **Log(Bal_Total)**, the coefficient in the parameter estimates table (top in Exhibit 15) indicates how revenues change as the account balance changes. A positive coefficient indicates that revenues increase on average as account balances increase. The coefficient value itself is somewhat difficult to interpret because it reflects the transformation of **Rev_Total** to **Log(Rev_Total)**.

**Exhibit 15** Exploring the Reduced Model with the Prediction Profiler



Parameter Estimates

| Term | Estimate | Std Error | t Ratio | Prob>\|t\| |
|---|---|---|---|---|
| Intercept | -2.538458 | 0.034835 | -72.87 | <.0001* |
| Log(Bal_Total) | 0.4424023 | 0.004923 | 89.86 | <.0001* |
| Offer[0] | -0.069977 | 0.019193 | -3.65 | 0.0003* |
| CARD[0] | -0.7964 | 0.023826 | -33.43 | <.0001* |
| LOAN[0] | 0.0572065 | 0.017386 | 3.29 | 0.0010* |
| INSUR[0] | 0.0400496 | 0.013363 | 3.00 | 0.0027* |
| Check[0] | 0.7049116 | 0.022894 | 30.79 | <.0001* |

Prediction Profiler

For each of the two-level categorical predictors, parameter estimates show how the average response changes at the low level of each predictor. For example, the coefficient for **CARD[0]** is negative 0.7964, indicating that log revenues are 0.7964 lower on average if credit card activity is low, and 0.7964 higher on average if credit card activity is high. The coefficients for **LOAN**, **Check** and **INSUR** are all positive, indicating that low activity in these three accounts leads to higher revenues.

**Note:** When fitting regression models in JMP, two-level categorical predictors are automatically transformed into coded indicator variables using a -1/+1 coding scheme. The parameter estimate is reported for the lowest level or value of the predictor. In this example, **CARD** is a nominal predictor with levels with 0 and 1. The term in the reduced model is represented as **CARD[0]**, and the parameter estimate is -0.7964 (see Exhibit 15). The estimate for **CARD[1]**, which is not reported, is +0.7964. To display both estimates, select **Expanded Estimates** from the **top red triangle > Estimates**.

Many statistical software packages require dummy coding of categorical predictors using a 0/1 “dummy” or “indicator” coding scheme. This coding is done prior to fitting the model, and results in different parameter estimates and a different interpretation of these estimates. For example, the parameter estimate for **CARD**, using 0/1 dummy coding, is 1.5928 instead of -0.7964. The sign is different, and the estimate is exactly twice the magnitude. To confirm this, change the modeling type for **CARD** to **Continuous** (to tell JMP to use dummy coding) and refit the reduced model shown in Exhibit 4.28. Note that, although the parameter estimates are different, the two coding schemes produce identical model predictions.

To view the indicator-coded version of parameter estimates in the **Fit Least Squares** output, select **Indicator Parameterization Estimates** from the **top red triangle > Estimates**. Further details of how JMP transforms categorical factors can be found in the Statistical Details section of the book *Fitting Linear Models* (under **Help > Books**).

### Using the Prediction Profiler

The **Prediction Profiler** (see the bottom of Exhibit 15) can help us understand how changes in the values of predictor variable impact **Log(Rev_Total)**. To turn on the Profiler, select **Factor Profiling > Profiler** from the red triangle for the variable.

The Profiler shows the predicted response (on the far left) at specified values of each of the predictor values (given at the bottom). Initial values for the predictors are predictor averages, and vertical red lines are drawn at these values. The starting value for the response is also the overall average (the mean **Log(Rev_Total)** in this example), and the bracketed values are the 95% confidence interval for the average. The slopes of the lines for each predictor indicate whether predicted **Log(Rev_Total)** will increase or decrease if predictor value increases, assuming that other predictor values are held constant.

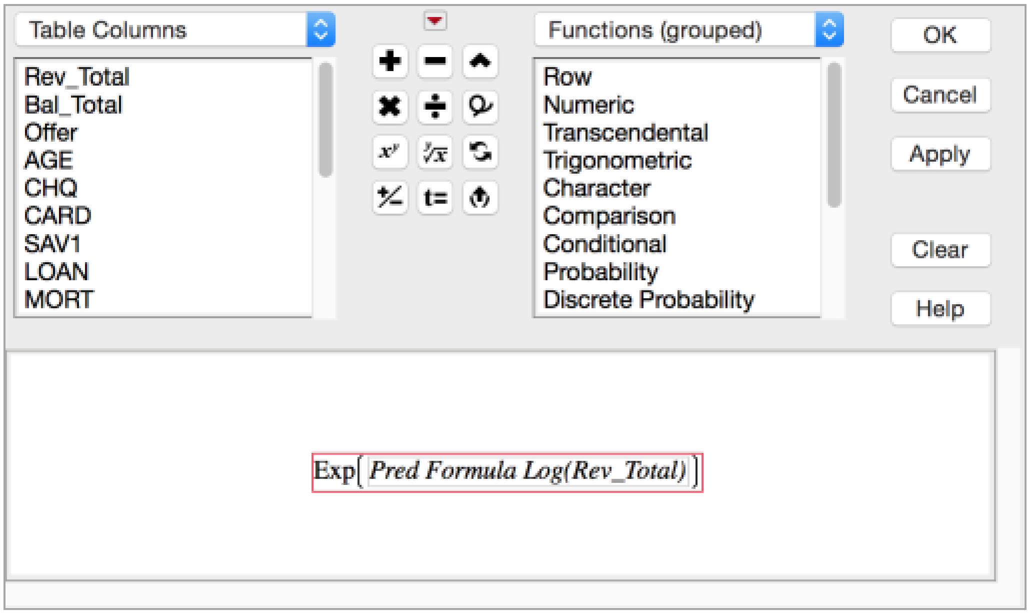
Clearly, **Log(Bal_Total)** has a large positive effect on the response (i.e., the slope of the Profiler line is steep). Three predictors, **Offer**, **LOAN** and **INSUR**, while significant, have a relatively small effect on the response (the profile lines are relatively flat.

To show the predicted values for each bank customer, the prediction equation (the formula) can be saved to the data table (red triangle, **Save Columns > Prediction Formula**). Unfortunately, these are the log predicted values, which are difficult to interpret.

The inverse transformation (in this case the *exponential*, or *Exp* function) can be used to examine predicted values on the original scale. To apply this transformation, create a new column in the data table (we've named this column **Pred Rev_Total).** Then, right-click on the column and select **Formula** to open the **Formula Editor**, and use the **Transcendental > Exp** function from the **Functions (grouped)** list (see Exhibit 16).

Note that this formula can also be created using a shortcut. Simply right-click on the saved prediction formula column, and select **New Formula Column > Transform > Exp**. JMP will create a new column with the stored formula shown in Exhibit 16.

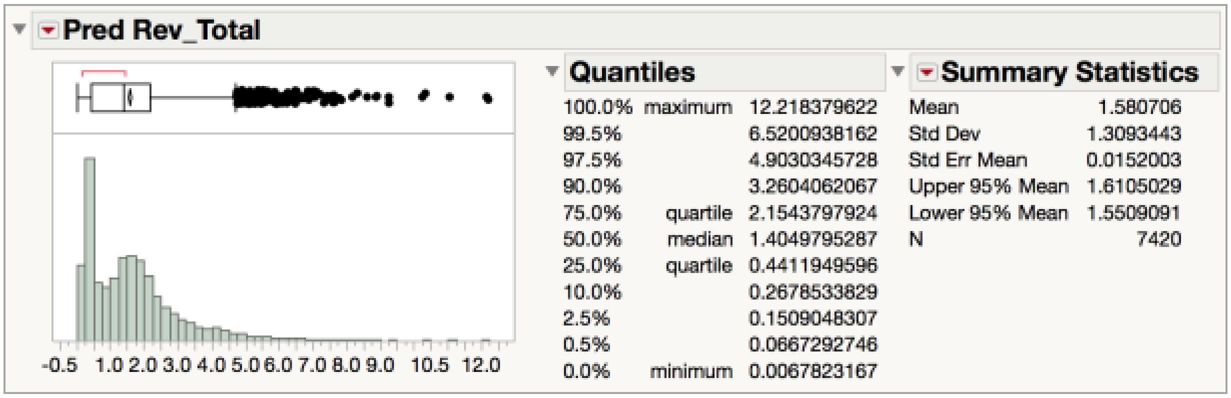
**Exhibit 16** Transforming Predicted Log(Rev_Total) to Predicted Rev_Total



Exp[*Pred Formula Log(Rev_Total)*]

Now we can explore the distribution of these values using **Distribution** or the **Graph Builder** (see Exhibit 17).

**Exhibit 17** Distribution of Predicted Rev_Total



Quantiles

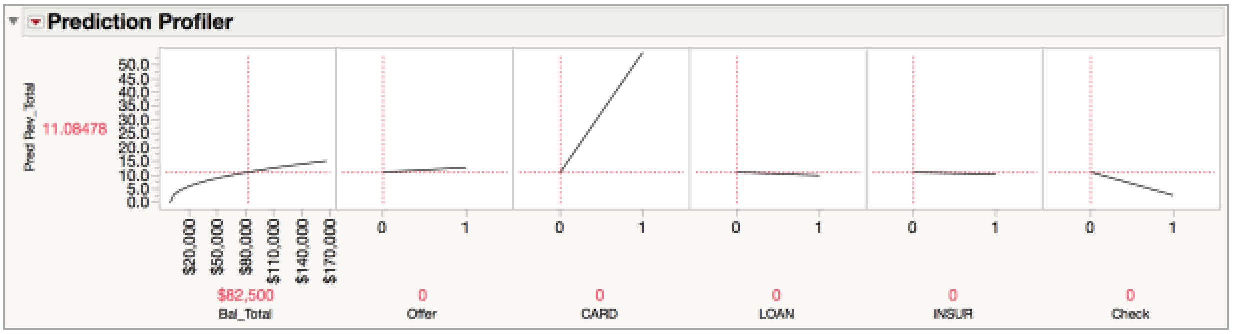
| | | |
|---|---|---|
| 100.0% | maximum | 12.218379622 |
| 99.5% | | 6.5200938162 |
| 97.5% | | 4.9030345728 |
| 90.0% | | 3.2604062067 |
| 75.0% | quartile | 2.1543797924 |
| 50.0% | median | 1.4049795287 |
| 25.0% | quartile | 0.4411949596 |
| 10.0% | | 0.2678533829 |
| 2.5% | | 0.1509048307 |
| 0.5% | | 0.0667292746 |
| 0.0% | minimum | 0.0067823167 |

Summary Statistics

| | |
|---|---|
| Mean | 1.580706 |
| Std Dev | 1.3093443 |
| Std Err Mean | 0.0152003 |
| Upper 95% Mean | 1.6105029 |
| Lower 95% Mean | 1.5509091 |
| N | 7420 |

We can also explore the formula itself using **Graph > Profiler** (Exhibit 18). Select the transformed prediction formula as the **Y, Prediction Formula**, and check the **Expand Intermediate Formulas** box to drill down to the original saved prediction formula. Now, we can readily see and explore the impact of changes to each of the variables on the predicted revenues in the original scale.

**Exhibit 18** Prediction Profiler for Predicted Rev_Total



## Summary

### Insights

It is clear that high account balance customers and those who use their credit cards frequently generate more revenue. What is curious is that high checking account usage seems to indicate lower revenue, and that customers with higher activity on loan and insurance accounts have lower predicted revenue on average.

### Implications

Was the promotional offer successful? That is, did it lead to increased revenue? For a customer maintaining an account balance of $82,500, with low credit card, loan, insurance and checking account activity, the promotional offer increased revenues from $11.08 to $12.75 on average. If this same customer had high credit card activity instead of low, the predicted revenue increased from $54.5 to $62.7. (Click and drag the vertical red lines in the prediction profiler to see how the predicted response changes as you chance values of the predictors). However, this analysis does not determine return on investment. Further information would need to be gathered to determine the cost of the promotional offer program and to examine the increased revenue relative to that cost.

### JMP Features and Hints

The **Graph Builder** was used to explore the shape of the response distribution and to dynamically apply a log transformation. The **Distribution** platform was used to explore distribution shapes and potential relationships between the variables. A least squares regression model was fit using **Fit Model**, and **Stepwise Regression** was used to reduce the model. Residuals were used to explore model assumptions, and Cook's D values were saved to check for influential observations.

Since the model was created to predict the Log of the response, the prediction formula was saved to the data table and the inverse transformation (Exp) was applied to the predicted values. The **Prediction Profiler** was then used to explore and understand the relationships between the predictors and the response, in the original units (revenue, rather than log revenue).

## Exercises

**Exercise 1**: In this exercise we use the BankRevenue.jmp data.

Fit a full model to **Log(Rev_Total)** using **Log(Bal_Total)** and the other variables as model effects (using main effects only). Note, you may need to recreate these columns. Use the Minimum BIC stopping rule and stepwise regression to build your model.

a. Compare your reduced model to that obtained using Minimum AICc in this chapter. Describe the differences in terms of the variables in the model and key statistics (adjusted R Square, RMSE, and other statistics provided).
b. Which is the "better" model? Why? Does one model do a better job of predicting the response than the other? Explain

**Exercise 2**: Continue with the BankRevenue.jmp data.

Instead of fitting a model using the transformed variables, fit a model using the original (untransformed) variables. Use **Rev_Total** as the response, and **Bal_Total** and the other variables as model effects. Use stepwise and your preferred stopping rule to build the model.

a. Restate the model assumptions presented earlier in this case.
b. Use the tools covered in this chapter to check model assumptions. Which tools should you use to check these assumptions? Explain how each tool helps check assumptions.
c. Explain why the model assumptions are or are not met.
d. Does it make sense to use this model to make predictions? Why or why not?

**Exercise 3**: Use the BostonHousing.jmp data set from the **Sample Data Library** (under the **Help** menu) for this exercise. The response of interest, **mvalue**, is the median value of homes for towns in the Boston area in the 1970s.

a. Use the tools such as the **Columns Viewer** (from the **Cols** menu), **Analyze > Fit Y by X**, and the **Graph Builder** (from the **Graph** menu) to explore the data.

    i. Are there any potential data quality issues (other than the fact that the data are from the 1970s)? Determine what actions, if any, should be taken to address data quality issues that you identify. Document what you discover, and any steps you take to address these issues.

    ii. Do there appear to be any relationships between the predictors and the response? Describe what you observe.

b. Fit a model to **mvalue** using only **chas** and **rooms**. Recall that **rooms** is the number of rooms (rooms) and **chas** is a dummy variable (**chas=**1 indicates the town tracks the Charles River).

    i. Write down the equation for this model.

    ii. Interpret the coefficients for **chas[0]** and rooms.

    iii. What is the predicted **mvalue** for a home that tracks the Charles River and has 6 rooms?

c. Fit a model to **mvalue** using all of the other variables as model effects. Use the Minimum BIC stopping rule and stepwise regression to build your model. How many terms are in the final model? Which terms are not included in the model?

d. Check model assumptions. Are model assumptions met? Explain.

e. How would a realtor, selling homes in the Boston area (in the same time period), use this model? How would a potential home buyer use this model?

**SAS Institute Inc. World Headquarters +1 919 677 8000**

JMP is a software solution from SAS. To learn more about SAS, visit **www.sas.com**
For JMP sales in the US and Canada, call 877 594 6567 or go to **www.jmp.com**