ANOVA, Regression and Logistic Regression

- ANOVA, Regression and Logistic Regression
  - Hypothesis Testing
  - Failing to Reject vs. Accept
  - P value
  - One Sample T-test
    - Verify if a population mean estimate is same as a **hypothesized value**, using a sample.
  - Two Sample T-test
    - Verify if two population have the same mean using samples from either.
    - Assumptions
  - PROC SGPLOT
    - Visualize relationships b/w cat.Predictors vs. cont.Response
  - PROC SGSCATTER
    - Visualize the relationships b/w cont.Predictors vs. cont.Responses
  - One Way ANOVA & PROC GLM
    - Detect effect on a cont.Reponse from the levels in a single cat.Predictor
    - Assumptions
  - POST HOC PairWise Comparison
    - Detect which level in a cat.Predictor level is causing effect in ANOVA
  - Pearson Correlation
    - Detect degree of linear relationship between variables
  - Linear Regression
    - Simple Linear Regression
      - Analyze the effect of a cont.Predictor on a cont.Response, also use the model to predict response.
    - N-Way ANOVA
      - Analyze the effect of more than one cat.Predictors on a cont.Response
    - Multiple Linear Regression
      - Analyze the effect of multiple cont.Predictors on a cont.Response
    - Model Selection for Linear Regression Models
      - Selecting the best models from all possible ones
    - Information Criterion
      - Measures for comparative model evaluation
    - Model Post Fitting
      - Steps to ensure the model assumptions are met before trusting the model
    - Outliers vs. Influential Observations
    - Detecting Collinearity
      - Collinearity leads to biases in the model
    - Predictive modelling with Linear Regression
      - Applying a selected model to a new dataset
  - Data Scoring with your model
    - Apply a model to predict response variables in a new dataset
  - Logistic Regression
    - Overall process
    - Model the relationship between a **binary response** vs. set of predictors (cont. or cat.)
    - PROC FREQ

        - Discover associations and evaluate classifier performance
    - Tests of association
    - Detecting Ordinal Associations
    - PROC LOGISTIC
        - Oversampling
        - Result interpretation
    - Interaction effects in Logistic regression
    - Predictive analysis/ Scoring using Logistic regression
- Data preparation
    - Missing data
    - Collapse the levels of categorical inputs.
    - Redundant Data
        - Step 1 - Variable reduction with Clustering
        - Step 2 - selecting a variable from each cluster
    - Variable screening
    - Automatic variable selection using PROC LOGISTIC
- Measuring Model Performance
    - Classifier Performance
        - ROC curve & AUC
        - Gains Chart and Lift Charts
        - Profit Prediction
        - Bayes Rule
        - Kolmogorov - Smirnov Statistic
    - Comparing models using plots

Hypothesis Testing

	H_0 is true	H_0 is false
Fail to Reject	correct inference	Type 2 Error P= β
Reject	Type 1 error P=α	Correct inference / Power= P= 1-β

Failing to Reject vs. Accept

If the results of a test supports the alternative hypothesis, then the null hypothesis can be rejected as false. However, if the data does not support the alternative hypothesis, this does not mean that the null hypothesis is true/accepted. All it means is that the null hypothesis has not been disproven—hence the term "failure to reject." A "failure to reject" a hypothesis should not be confused with acceptance.

P value

It measures the probability of observing a value as extreme as the one observed or more extreme, assuming that the H0 is true (H0 is usually that the means are same, so any difference is by chance). When the P value is low(significant), that means that the probability of observing the *effect size* (difference between the observed value and expected value) due to pure chance is low, ergo, the effect is not by chance.

One Sample T-test

Verify if a population mean estimate is same as a hypothesized value, using a sample.

Example: The mean price of a home in an area is $300,000. The hypothesized value is $300,000 and we use analysis on a sample to estimate the population mean.

It's used to check if our estimate of the population mean from a sample mean and standard error is accurate. A sample is taken and the sample mean is calculated. The null hypothesis is that the population mean is the statistically same as the sample mean, and the Ha is that they are different. Since the population std.dev is not known, its estimated using the *student's t distribution* which is similar to the normal distribution, but with a wider spread, and approaches the normal distribution when the sample size is large. The *T statistic* estimates how far the sample mean is from the hypothesized mean. The hypothesized mean is our assumption, like we assume the mean price of homes in the bat area is 900k, and this is what we are testing with our sample. If the H0 of pop.mean = sample.mean is true, then the T value will be small. Given a T value, we need to see the probability of observing such a T-statistic. If that probability is high (P > α), then we fail to reject the H0, i.e. the probability of observing such a T-statistic is high. If the P-value low, that means the probability of observing such a T-statistic is low, so our assumption of H0 is wrong, and we reject the H0.

```
PROC TTEST data=stat1.normtemp
                   plots(shownull)=interval
           H0=98.6;
    VAR bodytemp;
RUN;
```

- Here the data points to the dataset/sas table.
- H0 is the hypothesized mean
- Plots will plot an interval plot that shows the sample mean and the 95% confidence range for it, and `shownull` draws the hypothesized mean.

If the hypothesized mean lies outside the 95% confidence range, then the sample mean is very different from the hypo.mean. The T-stat is then expected to be large, and if the P-value for seeing such a large T-stat is low, then we know that H0 can be rejected, ergo the hypothesized mean is not the actual population mean. We should also ensure that the sample has a normal distribution and check this from the histogram and the normal QQ plot.

Two Sample T-test

Verify if two population have the same mean using samples from either.

Ex : Check if houses with heated garages have the same mean price as houses without heating in the garage. The two populations are demarcated from the same dataset using a categorical variable that denotes the availability of heating in the garage.

A two sample T test tries to see if two populations have the same mean, given a sample from each. This is unlike a one sample T-test where we are trying to see if the sample mean can accurately estimate the population mean.

Assumptions

The assumptions for 2-sample T-Tests are:

- The obs are independent (eg: no repeat measurements)
- Normally distributed population means (check the samples if they follow a normal distribution)
- The two populations should have equal variances.

To test for equality in variances, use the folded F test and compute the *F-statistic*. F statistic is the ratio of the max(sample variance) to the min(sample variance), and its always >= 1. The closer it is to 1, the lesser the variance. The P value for this F-Statistic will also be >= significance level showing that the variances of the two populations (H0) is true.

```
PROC TTEST data=stat1.german
           plots(shownull)=interval;
       CLASS group;
   VAR change;
RUN;
```

- The 2-Sample t test does not use the H0 parameter in the `PROC TTEST` .
- Instead, it has the `class` which should be a categorical variable with exactly two levels.
- The `VAR` as usual indicates the measurement we are interested in.

SAS will give two T-statistic values for the 2 sample t-test - *Pooled* and *Satterthwaite*. Evaluate the equal variances assumption using the F-statistic and the P value for the Fstatistic. *Pooled* T-statistic and its P value can be used if the F-statistic indicates equal variances in the populations. If it does not, the *Satterthwaite* T-Statistic compensates for the unequal variances and can be used instead.

PROC SGPLOT

Visualize relationships b/w cat.Predictors vs. cont.Response

`PROC SGPLOT` is used to create box plots to visualize the relationships between categorical predictor variables and continuous response variables.

```
PROC SGPLOT data=stat1.garlic;
   VBOX bulbwt / category= fertilizer connect=mean;
RUN;
```

Here the `VBOX` option is used so a vertical box-plot is created using the response variable `bulbwt`. The categorical variables used for each box in the boxplot is the *fertilizer*. If *fertilizer* has 3 values, then there will be 3 Boxes in the blox-plot and they will be connected by their means because `connect=mean` option has been specified.

PROC SGSCATTER

Visualize the relationships b/w cont.Predictors vs. cont.Responses

Creates scatter plots. Can be used to create a whole panel of scatter plots that explore the relationship of the predictor variables to the response variables.

```
PROC SGSCATTER data=STAT1.ameshousing3;
    PLOT SalePrice*(Gr_Liv_Area Basement_Area Garage_Area Deck_Porch_Area
         Lot_Area Age_Sold Bedroom_AbvGr Total_Bathroom) / reg;
    TITLE "Associations of Interval Variables with Sale Price";
RUN;
```

The format for the `plot` directive is *response_variable* ***** (predictor1 predictor2 predictor3...)

One Way ANOVA & PROC GLM

Detect effect on a cont.Reponse from the levels in a single cat.Predictor

One way ANOVA is silimar to the 2-sample **T test**, in that we are trying to see if the population means of different populations are same or not, but unlike the 2-sample **T-Test**, the **One Way ANOVA can compare the means of more than 2 populations**. ANOVA compares the within-group variation to the between-group variation to evaluate if the populations have the same mean.

R^2 called the co-efficient of determination, is a measure of how much the predictor variables explain the variability in the data and is between 0 and 1. The closer it is to 1, the predictor variables explain a larger portion of the variability and the closer it is to 0 the lesser the predictor variables are at explaining the variability. So this can be seen as a measure of how well we have chosen our predictor variables, and how well they explain the effect.

In the ANOVA table the **F-value** indicates the amount of variability explained by the model and the larger the value, the more variability is explained by the model. The P-Value is the probability of getting an F-Value at least as extreme as thats computed by chance, and if its less than the significance level then we can reject the H0 of the population means being equal.

Assumptions

- The obs are independent (eg: no repeat measurements)
- Normally distributed error terms (check using Normal QQ plots)
- The error terms should have equal variances.

```
PROC GLM data=stat1.garlic plots=diagnostics;
 CLASS fertilizer;
 MODEL bulbwt = fertilizer;
 MEANS fertilizer / hovtest=levene;
RUN;
```

In `PROC GLM` , the `plots=diagnostics` option generates a panel of plots that lets us validate the normality of error terms assumptions. The `class` is the categorical variable. `model` specifies the response and predictor variables in the model. The `means` includes the option to run the `hovtest` (using levene's method in this case).

The `PROC GLM` displays the ANOVA table with the F-Value as well as the P value. If the P value is lesser than the significance level then we can say that the probability of seeing an F-value as extreme as the one computed, purely by chance is very small, and we can reject the H0 that the population means are equal (in other words, we reject the idea that the predictor valuable values do not affect the response variable). But before we can trust the P-value, we need to check the assumptions. The Normal QQ plots and the histograms give us an idea if the error terms are following a normal distribution.

The HOV test will produce a test statistic and a P-value which we can use to evaluate the H0 of equal variances in the error terms. This simply tells us that at least one of levels in the categorical predictor variable has a significant effect on the response variable. To know which level has this effect, we need the post-hoc tests to compare every category level in the predictor with each other or against a chosen reference level.

POST HOC PairWise Comparison

Detect which level in a cat.Predictor level is causing effect in ANOVA

If we reject the H0 in AVONA, and determine that at-least one of the predictor variable levels is significantly different from the others, we use the Post Hoc pairwise comparisons to see which one (or more) is different. Here we can use

- **Tukey's adjustment** to compare between every pair or
- **Dunnett's Adjustment** to compare each category against a single "control" group

The results of the post hoc analysis are graphed using a diffogram. The diffogram's diagonal line represents the case when there is no difference between the means. Each pairwise comparison is plotted and the confidence intervals are represented as lines. If the condifence interval line touches the diagonal, then the means are not significantly different. Control plots are another mechanism used for Dunnets's method where there is a control group. The Control plot has a shaded area and lines that represent the mean. if the line falls inside the shaded area, then the difference means for that variable compared to the control group is not significant.

```
LSMEANS fertilizer / pdiff=all adjust=tukey;
LSMEANS fertilizer / pdiff=control('4') adjust=dunnett;
```

Pearson Correlation

Detect degree of linear relationship between variables

The pearson correlation coefficient determines the strength of the linear association between two continuous variables - usually a predictor and a response. It can also be used to detect collinearity, or correlation between predictors. It is denoted by **r** ranges from -1 to +1. The closer the value is to -1, the stronger the negative linear association between the variables. The closer it is to +1, the stronger the positive association. 0 indicates that the variables are not associated.

Gotchas :

- Corelation does not imply causation.
- Pearson correlation can only measure **linear** associations. Associations can be non-linear as well.
- Outliners affect the correlation coefficients.

`PROC CORR` produces the pearson correlation coefficients and the p-values. You can use the `with` keyword to compare the variables in `var` to the variables in the `with`. Omitting the `with` statement will generate comparisons between all pairs of variables in the `var` statement. The `plots` can specify scatter plots with `nvars=all` to show all the variables. The proc produces the scatter-plots as well as the Pearson correlation coefficient to evaluate if there is a linear association between the variables.

```
PROC CORR data=STAT1.BodyFat2
          plots(only)=scatter(nvar=all ellipse=none);
   VAR &interval;
   WITH PctBodyFat2;
   ID Case;
   TITLE "Correlations and Scatter Plots";
RUN;
```

Linear Regression

Simple Linear Regression

Analyze the effect of a cont.Predictor on a cont.Response, also use the model to predict response.

The basis of linear regression should be familiar. The `PROC REG` is the SAS proc for linear regression. The proc will create an ANOVA table and compute the F-statistic and the P-value for it , with the H0 being that the response variable and the predictor do not have any association. So if the p-value is significant(< alpha), the we can reject this H0 (no association) and infer that there is a significant association between the predictor and the response variable.

The only thing that `proc reg` requires apart from the data set is the `model`. The `plots` produce the graphs that we can use to validate the test.

```
PROC REG data=stat1.bodyfat2
                 plots=all;
 model PctBodyFat2=Weight;
 run;
```

N-Way ANOVA

Analyze the effect of more than one cat.Predictors on a cont.Response

The reason to do an N-way (N = # of cat.Predictors) ANOVA rather than N * One-Way ANOVAs is that N-Way ANOVA also analyses the **interactions** between the N predictors. Also note that a One-Way ANOVA has one cat.Predictor with N levels, while an N-way ANOVA has N cat.Predictors, each with thier own levels.

The first step is to visualize the interactions to make sure there are some. To do this you can use an `PROC sgplot` with a `vline` option.

```
proc SGPLOT data=stat1.drug;
        vline DrugDose / group=Disease response=BloodP stat=mean markers;
        format DrugDose dosefmt.;
RUN;
```

Here the VLine splits the data by a cat.Predictor (DrugDose) on the X-Axis. Then its grouped by disease, so for each dose the observation for each group is plotted. The `response` or the value to be plotted is the blood pressure, and the `stat` specifies what statistic to plot. If the plot has as similar pattern, then there may not be any interactions, however if atleast one group has a different plot then we can say that the response variable changes differently for atleast one pair of the cat.Predictor and group - there is an interaction.

Now we can do an N-way ANOVA :

```
PROC GLM data=stat1.drug plots=all;
Class drugdose disease;
model BloodP = drugdose disease drugdose*disease;
lsmeans DrugDose*Disease / slice=Disease;
format Drugdose dosefmt.;
store out=modelitems;
run;
quit;
```

Here the cat.Predictors we are using are drugdose and disease, in the `CLASS` . The `MODEL` includes the reponse variable and the predictors and the interaction term. `LSMEANS` will produce the lsmeans table, and here we are slicing it by the disease. This includes the F value for each ans corresponding P-value, which will tell us if the particular `slice` has a ststistically significant effect for each slice.

Multiple Linear Regression

Analyze the effect of multiple cont.Predictors on a cont.Response

A best practice in a two-way ANOVA is to plot the data to identify possible interactions between the variables. An interaction occurs when the difference between group means of one variable changes at different levels of another variable. This causes non-parallel lines in the interaction plot.

```
%LET var = Age Weight Height Neck  Abdomen Hip Thigh  Ankle Biceps Forearm Wrist;

PROC REG data=stat1.bodyfat2;
MODEL PctBodyfat2 = &var;
RUN;
QUIT;
```

In the example, the `PROC REG` is used like in the case of Linear Regression, but there are more terms in the model. This generates the standard ANOVA table for the overall model performance, and the P-value will indicate if there is a significant effect on the response by the predictors. If we have a lot of predictors, we should use the adjusted R^2 to account for the effect of additional terms. The table also gives the parameter estimates for all the terms and we can try to eliminate the terms with the highest P-value (least significant) to simplify the model without sacrificing accuracy. Every term removed will reduce the overall P-value, but the magnitude of the reduction will be small the more insignificant the term is (removing an highly effective term will increase the overall model P-value considerably)

Model Selection for Linear Regression Models

Selecting the best models from all possible ones

With *k* predictors, there will be 2^k models. So we need tools that can evaluate models and select effective models. Common Model selection approaches :

- All possible regressions
  - Computes all possible regression models and evaluates them
  - Practical only for a reasonable number of perdictors
  - SAS can return the best *n* models using the `BEST=` option. All 2^k models are still evaluated before the best *n* is selected.
- Step-wise selection
  - does not compute all possible models, and is faster and more reasonable for very large number of predictors.
    - **Forward** - model starts with 0 predictors
      - the F-statistic for ever variable not in the model is calculated, and the highest one if above a threshold is added to the model.
      - once added the predictor stays, even if subsequent predictors make it redundant.
    - **Backward** - model starts with all predictors
      - The F-statistic of every variable in the model is computed and the least significant is removed.
      - once removed the predictor stays out, even if adding it could have yeilded a better model.
    - **Step-wise** - starts with no predictors like the forward selection.
      - New predictors are added at every step by checking their F-statistic of the vars not in the model.

    - The existing predictors' F-statistic are also evaluated and the insignificant ones are dropped (like in backward selection)
    - Stepwise techniques don't take any collinearity in your model into account. Collinearity means that predictor variables in the same model are highly correlated.

Automatic model selection has issues and biases. Some are due to using the p-values and some are due to using the same sample to build the model and evaluate the model.

- **Overfitting the data** when the same sample is used to build the model and evaluate it, the model fits too well for the sample and becomes less significant for the population.
- divide the data set in to two and use one to build the model and the other to validate the model.
- This is impractical if the available sample size is low.
- Techniques like `k-fold cross validation` and `bootstrap method` can avoid overfitting with small sample sizes.

```
%LET interval=Gr_Liv_Area Basement_Area Garage_Area Deck_Porch_Area
         Lot_Area Age_Sold Bedroom_AbvGr Total_Bathroom ;

PROC GLMSELECT data=STAT1.ameshousing3 plots=all;
        STEPWISE: model SalePrice = &interval / showpvalues selection=stepwise details=
        TITLE "Stepwise Model Selection for SalePrice - SL 0.05";
run;
```

Information Criterion

Measures for comparative model evaluation

Information criterion are used to comparatively assess the fit of a model between many. They all try to point to the **model that explains the most variability in the data with the least predictors**. They all have a penalty to using more variables in the model and this penalty is what is different between them. ***Smaller values for information criteria is considered better.***

- AIC
- AICC (Corrected AIC) - Used for small sample sizes
- BIC
- SBC

Adjusted R^2 is similar to the Information Criteria in that, it penalizes the addition of terms. Since adding a term will never reduce R^2 value since the worst that could happen is that the term added has no effect on the response variable and hence the R^2 stays the same. The model becomes unnecessarily complex however, and this is why adjuisted R^2 will give a better estimate and balance the complexity of the model vs the variability explained by it.

Model Post Fitting

Steps to ensure the model assumptions are met before trusting the model

A model might have an impressive R^2 value, but that does not mean that the model is accurate, if we violated our model assumptions when we created it. The assumptions are :

1. The predictors and the response have a linear realtionship
    - Can be checked visually usig a scatterplot.
2. The errors (residuals - diff between the prediction by model and actual) are normally distributed.
    - Use residual plots. Generated by default by `PROC REG` .
    - Random scatter indicates that the residuals are normally distributed.
    - With multiple terms, the residual plot that does not appear random idicats the term that is violating the assumption.
3. The errors have equal variances at each value of the predictor
4. The errors are independent.

The pattern in the residual plots indicate the assumption that is being violated. - Curve / Quadratic shape -> indicates that the `linearity` assumption is violated. - Funnel shape -> `assumption of equal variances` of the error terms are violatied - Cycle/Sine Wave shape -> assumption of independent measurements is violated.

Outliers vs. Influential Observations

An Outlier is an unusual datapoint, whereas an Influential Observation is an Outlier that has a significant effect on the model. If removing an observation produces a significant change to the parameter estimates, then that is an influenatial observation.

Detection :

Both `PROC GLMSELECT` as well as `PROC REG` produces the effect plots that include the following plots.

- **Outliers - Students residuals**
    - convert the residuals in to the same unit as the standard deviation.
    - Based on the distribution, 65% of values will be within 1 SD, 95% within 2 SD, and 99 %within 3 SDs.
    - If the student residual is 0-2 no issue(95% data falls in this range), 2-3 is unlikely and > 3 is definitely an outlier.
- **Influential Observations - CooksD, Rstudent, DFFits.**
    - **Rsudent**
        - calcualted as the diff in rtesiduals between the standard model and the model with that observation removed.
        - if the Student and Rstudent values are different or its > 3 that indicates the value is influential.
    - **CooksD**
        - used for explanatory analysis, for parameter estimates
        - It estimates the parameters with each obs eliminated and compares this with the parameters when all obs are included.
            - if there is a significant change in the parameters, thats an influential obs
    - **DFFits**
        - used for predictive models.
        - It generates the predicted value with two data models, the first with all obs, and the second with that obs removed
        - If there is a big difference betweent the predicted values, then the obs is influential.
    - **DFBETAS**

    - while Rstudent residuals/ Cook'sD/DFFits analyze the influence of an observarion, the DF betas lets you see which atttibure of feature in the observation is casuing this influence.
    - This is similar to the cooksd method and is calculated by estimating the coefficient for the variable with all the data and then substracting the the coefficient for the variable with the observation removed. This difference is divided by the standard err.
    - A large (>2) value for the dfbetas indicate that the variable has a large impact.

In SAS, you can use the `PROC GLMSELECT` to do step-wise selection of a model and then use `PROC REG` to evaluate the influential observations. `PROC GLMSELECT` produces a macro variable `_GLSIND` that stores the predictors selected for the final model in `PROC GLMSELECT`.

```
proc glmselect data=STAT1.ameshousing3 plots=all;
    STEPWISE: model SalePrice = &interval / selection=stepwise details=steps select=SL
    title "Stepwise Model Selection for SalePrice - SL 0.05";
run;
quit;

proc reg data=STAT1.ameshousing3
         plots(only label)=
              (RSTUDENTBYPREDICTED
               COOKSD
               DFFITS
               DFBETAS);
    SigLimit: model SalePrice = &_GLSIND;
    title 'SigLimit Model - Plots of Diagnostic Statistics';
run;
quit;
```

Detecting Collinearity

Collinearity leads to biases in the model

Collinearity or AutoCollinearity is when the predictors are closely related. This does not violate any test assumptions, however it leads to biases and incorrect P-values.

- Collinearity should be checked before model selection.
- Both `PROC CORR`, and `PROC REG` with `VIF` option can be used.
- `PROC CORR` will give the pearson correlation and the p-value. The closer it is to +/-1 the higher the correlation or collinearity.
    - Typically you would want to do something like

```
PROC CORR data=housingdata;
  VAR predictor1 predictor2;
  WITH suspect_predictor;
RUN;
```

    - this compares the predictors in the `VAR` with the one specified by `WITH` and generates the Pearson Correlation for each comparison. This tells us if any of the predictors in the `VAR` are correlated with the one in `WITH`

- Alternatively, `PROC REG` with the VIF option will report the variance inflation or `VIF` along with the parameter estimates. **If the VIF for any term is greater than 10, then the term is likely involved in collinearity.**

Predictive modelling with Linear Regression

Applying a selected model to a new dataset

Predcitive modelling start with **data partitioning**. The model is trained and the validation set is used to evaluate the model. Then the best model is chosen and applied to new data to predict the outcome.

Predictive modelling can be **Parametric** based on formulas like regression analysis or **Non-Parametric** which are based on rules like decision trees and random forests. Predictive modelling is always a balance between Overfitting and Underfitting - there is no perfect model.

- **Overfitting** - These are generally complex models that are too flexible and fits the random noise in the sample as well. This *leads to higher variance* when applied to a population. This generally happens when the modeller tries to maximize the R^2 value, since adding new variables and overfitting will never reduce the R^2

- **Underfitting** - The are models that are too simplistic and systematically fail to see the pattern in data. This leads to a model that *generates a biased prediction*.

- **Data Partitioning** - splits the data sample into two sets - training and validation. This is to honestly assess a model and its fit on data that was not present when the model was built, but from the same population. Partitioning can be done only when there is a sufficiently large sample size. If the sample size is small or moderate, other methods like cross validation can be used, which enables us to use the same data set for model building as well as validation without biases. `PROC GLMSELECT` can use honest assesment to select models and can work with a single input dataset and partition it or work with data that is already paritioned in to training and validation datasets.

```
PROC GLMSELECT data = <training_data> seed = <int>;
MODEL response = predictor_terms / showpvalues selection=stepwise(select=AIC choose=val
PARTITION FRACTION (TEST= .1 VALIDATE = .15);
RUN;
```

In this example, the data set is randomly partitioned in to 3 - Training(75%), validation(15%) and test(10%) sets. The `SEED` is an interger that will reproduce the same obs in the parititions if we need to recreate the exact partitions again. The Model statement options use the **stepwise** model selection. The AIC is used as the selection criteria (default selection criteria is AIC) and the model is chosen based on the best ASE (average squared err) value for the *validation* dataset.

Data Scoring with your model

Apply a model to predict response variables in a new dataset

During model building, we applied an algorithm to get a model. If this is a parametric model, then we have a formula. Scoring is the process of applying this formula on new data to predict a target variable.

Data may need to be transformed before it can be scored. Basically the same changes done to the traning and validation datasets need to be done here as well.

Scoring can be done 3 ways :

1. Use `PROC GLMSELECT` to generate the model and score data in one go
   - Inefficient - the model is ephemeral and if another data set needs to be scored, it needs to be generated again.
   - This is also slow if the training dataset is large, becasue the model is generated each time.
2. Use `STORE` in `PROC GLMSELECT` and then a `SCORE` in `PROC PLM`
   - Model generated by `PROC GLMSELECT` is stored in an item store, so its persistent.

```
PROC GLMSELECT data= <data> ;
 MODEL SalePrice = LotArea Home_Age / selection=stepwise details=summary;
 PARTITION FRACTION (validate=0.1 test=0.05);
 STORE out=housingdata;
RUN;
```

   - Then the `SCORE` statement is used in the subsequent `PROC PLM` to act on the Model in the *item store*

```
PROC PLM restore= <item_store>
  SCORE data= <new data that needs to be predicted> out= <dataset to save input
RUN;
```

   - Can use very large training dataset. No need to rebuild the model.
   - Not compatible with older versions of SAS
3. Use `STORE` in `PROC GLMSELECT` and then `CODE` in `PROC PLM` to generate scoring code, and finally `DATA` to run the code generated by `PROC PLM` .
   - same as above, but 3 steps .

```
PROC PLM restore= <item_store>
  CODE file= <path to create new .sas file>
RUN;

DATA <name of result dataset>
   set <Libref to input data to be predicted/scored>
```

```
    %include <path to .sas file created by CODE step>
RUN;
```

- No need to share the item store the generated code will contain everything needed to do scoring.
- compatible with older versions of SAS.

Logistic Regression

Logistic regression is used to predict the value of a binary response. Since a binary response is a categorical response with 2 levels, most of the categorical tools can be used.

Compared to Linear Regression, we cannot apply the tools and fitting techniques directly to a binary response variable, since the response values are not normally distributed (they resemble a sigmoidal distribution "S" shaped) and we cannot assume equal variances. In a linear regression the estimate we can predict is continuous and can be any value, but a binary response will always be 0/1. To solve these issues, we can predict the probability that a reponse will be 0/1 - this probabilaity is still bounded, and ranges from 0 to 1, but now its a continuous response variable. Aslo note that since we are estimating the probaility of the response, there is no observed probability - so we cannot use the *least squares method* like in linear regression. We can convert this probability to the logit scale or the log odds scale log(p/(1-p)), and we will have the logit(p) from -infinity to +infinity. Now we can use the linear regression methods and then convert the result back to probability scale.

Basic review of tools and techniques for categorical response variables follows:

Overall process

1. Data Prep
    i. Deal with Missing data 2. `PROC STDIZE` to impute values 3. `PROC FASCLUS` to do cluster imputation
    ii. Reduce levels in Categorial variables
        a. `PROC CLUSTER` uses Greenacre's method
        b. Collapse levels with fewer obs and ones that have the least impact on the Chi-Squared
        c. R^2 indicates the proportion of the chi-squared remaining in each iteration compared to the original
        d. Plot the log(p-value) to see Ideal cluster count.
    iii. Reduce the overall input variables by reducing redundancy
        a. `PROC VARCLUS` to cluster variables that have similar effect on the target
        b. Select one variable from each cluster based on 1-R^2 value or SME input. 1-R^2 represents how the variable is corelated to within group to outside group.
        c. Pick variable that have a high correlation within group and least correlation outside its group.
    iv. Screen the variables
        a. We are trying to find poor performing variable, and not trying to find the most predictive ones.
        b. Look for non-linear relationships using Spearman co-efficient and Hoeffding value. Plot the spearman vs Hoeffding it should be relatively linear. Any thing that deviates from a

linear line can be eliminated (non-linear relationsip) and anything with a very high value for both spearman and heoffding can be eliminated (upper right quadrant for the plot)

c. For the non linear ones, discretize the continuous variables and plot the emperical logit plots. These may show a quadratic relationships, and thier binned percentile plots should be linear.

2. Run the clean inputs to `PROC LOGISTIC` Do automatic model selection using SCORE/STEPWISE/BACKWARD methods. If using Best Set method, after scoring the chi-squared values, generate fit statistics and select a model.
3. Select the model and valiadte using the valiadtion dataset.
4. Evaluate fit and see how well the odel generalizes.

Model the relationship between a binary response vs. set of predictors (cont. or cat.)

It models the probability of an outcome based on the predictor variables. In logistic regression, the dependent/response variable is always a categorical variable with two levels.

- Hypothesis tests are used to verify the realtionship between the the predictors and the response variable.
- A classifier model is built using the discovered realtionships with the predictors.
- The model is used to predict or classify new data in to one of the the two levels of the response variable.

PROC FREQ

Discover associations and evaluate classifier performance

`PROC FREQ` can display frequencies of the input variables or generate cross tabulation tables to discover associations between the categorical variables.

Tests of association

When the frequency analysis of the categorical varibles point to a difference in the distribution of the data, we need to make sure this distribution difference is not by chance, and there is actually an association.

- Chi-Squared Test, tests this association between categorical variables to make sure the differences in distribution are not by chance
- The Cramer's V statistic can be used to measure the strength of an association.

General notes:

- The Chi-Squared statistic measures the difference between the expected and the observed frequences in a cross tabulation. Expected counts assume no association - this is our H0.
- The greater the difference, the more the likely there is an association.
- The test generates a χ^2 statistic and it's p-value.
- The χ^2 statistic depends on the sample size and does not indicate the magnitude of the association. You can duplicate every obs and double the χ^2 statistic. So for very small samples, you may see the χ^2 statistic is small and the P-value is larger, but it can be artificailly changed by

duplicating the observations. In other words, the results point only to the existence of the association but not its strength.
- Cramer's V statistic measures the magnitude of the association.
- Cramer's V ranges from -1---0---+1 or 0---+1. The closer it is to 0, the weaker the association.

	male	**female**	**Row.total**
survived	(E=)O=2	(E=)O=10	12
perished	(E=)O=10	(E=)O=2	12
Col.Total	12	12	Total.Obs=**24**

The expected cell counts for H_0 = *There exists no associacation between survival and gender* are :

(Row.Tot * Col.Tot)/Obs.Tot . (12*12)/24 = 6

The χ^2 test can be performed by the `PROC FREQ` procedure.

```
PROC freq data=stat1.safety;
TABLES (Type Region Size Weight) * Unsafe / CHISQ;
run;
```

This creates cross tabulation reports with each of the predictors with the variable "Unsafe", and the `CHISQ` option for the `TABLES` statement performs the χ^2 test and produces the χ^2statistic and the P-values.

Detecting Ordinal Associations

Ordinal associations are cases where the predictor is an ordinal (category levels with a natuaral order : size=small/medium/big) and the association is ordinal in nature. The distribution of the response consistently increase/decrease across the predictor levels.

- Ordinal associations are tested using the **Mantel-Haenszel Chi-Squared** test, which is sensitive to the order of the predictor levels.
- The **Spearman corellation statistic** is used to measure the magnitude of the association. In SAS, its generated by the `CL` option (confidence limits) for `PROC FREQ` .

Just like the Pearson Chi-Squared test :

- It does not measure the magnitude of the association
- It simply evalutates if there is an associaton
- It is influenced by the sample size and for small sample sizes the value will be low with an insignificant P-value.
- Spearman's correlation statistic is not affected by sample size
- It ranges from -1----0---+1 with values near to -/+1 indicating a strong -/+ correlation and values close to 0 indicating a weak correlation.

```
PROC FREQ data=stat1.safety;
TABLES (Region Size )*Unsafe / chisq expected oddsratio cl;
```

```
RUN;
```

In the sample above, the `PROC FREQ` generates the cross tabulation tables for Region and Size with Unsafe. The options are

- `CHISQ` - chi-squared statistic
- `EXPECTED` - generate expected counts in the the cross tabulation tables
- `ODDSRATIO` - generate the odds ratio for the cross tabulation table.
- `CL` - generate confidence limits for stats, including the *Spearman Correlation*.

PROC LOGISTIC

`PROC LOGISTIC` is the procedure in SAS to fit a logistic regression model for a binary, ordinal or nominal response variable.

```
PROC LOGISTIC data= ameshousing_data plots(only)=(effect oddsratio);
   MODEL bonus_eligible(event='1') = basement_size / clodds=pl;
RUN;
```

The options are as follows :

- A `CLASS` statement can be specified to include cat.Predictors.
  - The referecen level in a cat.Predictor can be specified by the `ref` option. By default the reference level is the last level in alphanumerin order.
  - The paramterization (coding used for the levels) can be either
    - Effect coding - default - compares the difference between each level and the average with all levels.
    - referece cell coding - `param=ref` - compares the difference between each level and the referece level (default = last level in alpha numeric order)
- The `event` keyword specifies which reponse level probaility we are modelling. By default this is the first in the alphanumeric order, so usually the model predicts the value for 0.
- `clodds` - Confidence limit odds - can be:
  - PL - Profile Likelibood (conpute intensive, but works well with smaller sample sizes.)
  - Wald - Default value, and requires lesser computation, but not good for small sample sizes.

Oversampling

If we take a representative sample when modelling a rare event, then this may give us very few samples. Instead we can build a better (more obs) sample by, say choosing all the events in the population and a subset of the non-events. Now there is a larger proportion of the event in the sample, and the sample is biases. This is **oversampling**. The effect of oversampling needs to be adjusted after the model is built.

- Oversampling leads to the intercept being higher. This difference is the Offset.
- To correct the bias, the offset is applied to reduce the intercept
- The offset only affects the intercept and not the other parameter estimates
- The Pi-1 value of the population - the proportion of events in the population need to be known to compute the offset.

- Oversampling does not affect the sensitivity or specificity, so it also does not affect the ROC curve.
- It does affect the PV+ and PV-, so the gains chart and lift charts are affected

Result interpretation

In the output, always check the '**Probability Modelled**', make sure the model is modelling the desired outcome.

- Model Convergence - make sure that the model converges, without this the results cannot be trusted.
- Model Fit statistics
  - **AIC** - Penalizes the number of predictors, but not sample size
  - **SC/SBC** - Larger penalty for #predictors, and also adjusts for sample size. This metric favors the most parsimonious models.
  - **-2Log L** - -2* Log(Likelihood) - the value depends on the number of predictors, so this cannot be used to compare models with varying number of predictors.
  - **Testing global H0** - all regression co-effs are 0.
    - Use the **likelihood ratio** rather than the wald test to verify the gloabl H0. We can say that at least one regression co-eff is != 0 if the likelihood ratio's P-value is significant.
  - Analysis of Maximum Likelihood estimates - This table tells us which regression co-effs are significant, based on the Wald Chi-Squared test and its P-value. Compared to the Type-3 Analysis of effects, this is more detailed since this breaks down the analysis by the levels in the catergorical predictors.
  - Association of predicted probabilities with observations
    - This is a goodness of fit measure that used the model to evaluate the dataset itself.
    - All possible pair-wise combinations of the obs from either of the two binary response var is created.
    - The model is used to see if the model would have predicted the outcome correctly. ie, the predicted probability of the desired outcome is higher. This is also called **concordant**.
    - When the model's predicted probability for an outcome does not match the actual outcome, its called **discordant**.
    - When the model predicts equal probability of either outcome, it is a **tie**.
    - More concordant pairs and less ties and discordant pairs in a model makes it a better model.
    - The following statistics can be calculated.
      - Somer's D, Gamma, Tau-A, C (Concordant statistic)
      - Larger values for these statistics indicate a better fitting model.
- **The Odds Ratio** table shows the odds ratio and the confidence limits of the odds ratio should not include 1 to be significant.
  - Odds ratios depend on the `UNITS` statement in the `PROC LOGISTIC`. If a `UNITS` statement is provided, then the default option should also be provided because any predictors without an explicit `UNITS` set will be exacluded from the set.
  - Odds ratios measure how much a one unit change in the predictor changes the odds of the response (p(response)/1-p(response)).
  - The confidence limits of the the Odds Ratio has 1, then the predictor is not significant or does not change the odds of the response significantly.
  - Ex: in a `MODEL genuine_product(event='1') = price`, If the odds ratio for a $100 increase in price is 1.07, that means that there is a 7% increase in the odds that the item is genuine

for every $100 increase in price. Odds ratio does not show the change in logit or probability (though these can be calculated).

- **Analysis of maximum likelihood estimates**
  - This table shows the parameter estimates for the model
  - The Chi-squares values and the p-values are also shown
  - The insignificant predictors will have an insignificant P value
  - The most powerful predictors are found by looking at the **Standardized Estimate**column, where the rank order of the absolute value of the standardized estimate will yeild this order.

Sample :

```
PROC LOGISTIC data=stat1.safety plots(only)=(effect oddsratio);
class region(ref='Asia') size(ref='3') / param=ref;
model Unsafe(event='1') = region size weight / clodds=pl;
run;
```

The sample above does the following :

- data is from the safety dataset.
- Plots requests the effect plot (sigmoidal) and the odds ratio plot.
- CLASS identifies the region and size as cat.Predictors.
  - The `ref=` arg sets the reference level for each cat.predictor
  - The `param=ref` option specifies that reference cell coding should be used.
- Model speficies that the response variable is *Unsafe* and that we are modelling the event where Unsafe = 1.
  - The `CLODDS=PL` requests Profile Likelihood confidence limits (and also makes the oddsratio plot available for the plot option in the proc)

Interaction effects in Logistic regression

The core ideas are the same as that of linear regression.

- An interaction occurs when one predictor is affected by another, in other words, when the presence or absence of one predictor influences the the model parameter esitmates significantly.
- We can use **stepwise**, **forward** or **backward** selection methods.

`PROC LOGISTIC` can:

- do automatic model selction when the `selection` option is provided.
- model interactions by using the `|` between the model predictors that need to be analyzed for interactions.
  - `MODEL response = cat1 | cat2 | cat3 @2`
  - Indicates that the interactions between each pair of cat predictors need to be analyzed and `@2` indicates that only 2-factor interactions are analyzed.

Predictive analysis/ Scoring using Logistic regression

Using a logistic regression model to predict or score a new dataset is done similar to lineat regression.

- The model generated by `PROC LOGISTIC` is saved to an item store using the `STORE out=` statement.
- `PROC PLM` is then used with the `restore=` option to read the item store and then
  - `SCORE data= out=` statement can be used to score in modern SAS versions
  - CODE statement is used to generate the scoring code.

Data preparation

Missing data

- Complete case analysis - default in many procedures.
  - Leads to bias in the sample
  - `PROC LOGISTIC` with a `SCORE` will not score a new record that has a missing value
- **Imputing**
  - `PROC STDIZE` can impute values in a dataset based on the method specified
    - `REPONLY` replaces the missing values with imputed values
    - `REPLACE` replaces the missing values with 0
    - `METHOD` selects an imputation method - mean median etc.
    - `DATA` - input dataset
    - `OUT` - output dataset
  - Impute with median or other measures - cont.vars
    - simple but does not take in to account the effect of other vars
      - ex: when imputing the value of a home, rather than the median of all home prices, its better to impute the median of the homes in the same zipcode.
    - median is resistant to outliers
    - use when there is missing <= 50% of cases
    - If more than 50% of the values are missing,m then perhaps avoid that variable
  - Cluster Imputation - cont.vars
    - divides the dataset in to separate clusters based on other variables and imputed values based on the measures for the cluster/subset.
    - ex: the home value imputation example above
    - `PROC FASCLUS` does cluster imputation in SAS
      - It creates clusters based on the parameters in the VAR statement

Collapse the levels of categorical inputs.

- Cases where there are to many levels in a categorial variable ex: postcode.
- Create a new level in the cat.var that represent the missing values
- Quasi-Complete separation
  - This occurs when the values for a categorical var is tha same value in all the obs. Ex: The number of japanese speakers in a remote village in africa might be 0 for all observations. cat.var might be = languages known, with levels- English, Swahili, and Japanese.

- It becomes a perfect predictor.
- These levels need to be collapsed, so that the number of cat.vars is reduced. We can use `PROC CLUSTER` for this.
- `PROC CLUSTER` implements Greenacre's method. It tries to collapse category levels while minimizing the drop on the chi-squared value. It first collapses redundant category levels, and then collapses the category level that has very low values. It's an interative process and when each iteration is processed if we plot the log(p-value) for chi-squared versus the number of clusters, we will see a U-shaped pattern. The bottom of the curve indicates the # of clusters that give good P-values (any less number of clusters and we have a significant loss in P-value.) after a point at which the loss of the informaion is so great that the model is useless.
- The loss of information at each iteration is mesured by the drop in the Chi-Squared value and in the output the proportion of the chi-square that remains with each collapse is marked as the R^2. So to get the Chi-Square of a model, (overall_chi-square * R^2)

Redundant Data

- Redundant variables is not related to the target variable - these destabilize the parameter estimates, overfitting the data, compute power and effort.
- Variable reduction using variable clustering
  i. Identify variable clusters - corelated with eachother, but not with others
  ii. Select a variable from each cluster.

Step 1 - Variable reduction with Clustering

- `PROC VARCLUS` - Iterative Principal Components Analysis
  - Iterative process called **Divisive Clustering**
  - At each stage, a PCA is done and the second eigen value is checked if its above a threshold. the default is 1, and 0.7 is a good recommended value as well. If the second eigen value is large, then that menas there are at least 2 principal compone ts thats responsible for the variability
  - If the value is larger than cutoff, then the set is split in to two clusters PCA are done on each.
  - Smaller cutoff yeilds more clusters. when a cluster has only one variable, then the eigen value will be 0.
  - Example

```
PROC VARCLUS DATA= <data> MAXEIGEN= <cutoff, default=1 > ;
  VAR &vars_to_include_in_model;
RUN;
```

  - The `SHORT` option is used to suppress detailed output

- The `VAR` statement contains the numeric variable. if absent all numeric variables in the input are considered. Cat.Vars are not used by default, and if they need tob used, they need to be coded as dummy variables, using a `DO` loop

Step 2 - selecting a variable from each cluster

The step of variable selection from each cluster is subjective. You could use :

- 1-R^2 value. Lower values are better as indicate the highest corelation within the cluster and the lowest correlatin with other clusters.
- selection based on cost, or subject matter expertise.

Variable screening

- Variable screening can further reduce the number of inputs becasue `PROC LOGISTIC` uses the full model.
- After variable clustering, which identifies and collapses the correlated variables, the input screening method with univariate statistics - `PROC CORR` - can be used to detect irrelevant variables and non linear relationships between the variables.
- Spearman's correlation - measures the non linear, but monotonic (Y never decreases when X increses, but rate of change in Y can change)relationships.
    - Interpertation is similar to Pearson correlation, value ranges from -1 ---- 0 ---- +1. Values near +/- 1 indicate strong relationships and values around 0 indicate a weaker relationship
- Hoeffding correlation statistic can detect non-linear relationships as well. When the function is non-monotonic.
    - The value ranges from -0.5 --- 0 --- +1, and values around the edges indicate more stronger relationship and the values around 0 indicate a weaker relationship
- When a variable has a low rank on Spearman, and a High Rank on Hoeffding, that tell us that there is likely a nonlinear relation ship between that variable and the target.
    - A weak spearman correlation indicates a weak monotonic relationship, while a strong Hoeffding indicates a strong relationship including a nonlinear relationship, so the high Hoeffding must be attributed to a non linear relationship.
    - We can plot emperical logit plots to visualize the non linear realtionships or bin or discretize the values to account for this.
        - When binning, the smoothing of the logit plots is determined by the number of bins. The fewer bins there are, the smoother the curve.

Automatic variable selection using PROC LOGISTIC

- `PROC LOGISTIC` can use the same `selection=` option as `PROC GLMSELECT` and `PROC REG`
    - selection is based on the wald chi-squared test statistic
    - **Backward selection**
        - `selection = BACKWARD SLSTAY= (0..1, def=0.5)`
        - *starts with all variables* in model and iteratively drops them based on the wald chi-squared test and the `SLSTAY` param.

        - The variable with the largest P-value that is greater than `SLSTAY` is removed.
        - Backwards methods are better at excluding spurious inputs that can happen in forward methods.
        - Backwards methods suffer from quasi-complete separation (perform category level collapse to address) and multicollinearity. Inputs once excluded cannot be re-added.
    - **Stepwise**
        - `selection = STEPWISE SLENTRY=(0..1|def=0.5) SLSTAY= (0..1, def=0.5)`
        - *start with no parameters* and add parameters to the model based on the wald chi-squared test result and value for `SLENTRY` .
        - at every iteration existing variables are re-evaluated and ejected from the model if they no longer meet the criteria based the the `SLSTAY`
        - Stepwise selection does not work well in the case when there is multicollinearity in the variables. This can happen if the data preparation step does not address the redundant variables properly (using variable clustering)
    - **Best/All subset selection**
        - `selection=SCORE best= 1`
            - `best` selects the n best models that have the same number of variables.
            - So the 1 best model with 1 var, 1 best model with 2 vars.. until 1 best model with k vars (all vars).
        - Most comprehensive method
        - This method does not support `CLASS` baed selection, so dummay variables need to be created manually
        - computes all combinations of the variables and **rank orders their chi-squared score**.
        - Results in evaluating 2^k models for k variables and is very expesive, for large number of variables.
        - This calculates the the chi-squared value for all the combinations, but does not actually select a model. The chi-squared values never decreases in the models that are output from `PROC LOGISTIC` . So we need to perform fit statistics for finding out the most parsimonious model.
        - This uses the same techniques as linear regression to penalize the addition of variables to the model and generate fit statistics like AIC, AICC and BIC, and SBC and Brier Score. Brier score is a method to measure the delta between the predicted and
    - **Default selection is NONE**
    - Performance characteristics
        - Best-subset method performance follows a quadratic curve. Its fast for small number of inputs (<60), but when the #inputs increase, the time increases quadratically
        - FAST backward selection exhibits a linear increase in the time as the number of variables increase. its predictable
        - Stepwise is the poorest performing since the model has to be refit at each step. It follows a quadratic curve.

Measuring Model Performance

Optimism principle : If you use the same data to test a classifier model as the data you used to fit the model, then you will get better results that are misleading and you will end up overfitting or underfitting the data.

To avoid this split the data in to two or more datasets :

- Training dataset : used to fit the model
- Validation dataset : used to test and comapre models. Have atleast 10 events per each input variable
- Test dataset : used to do a final test on the selected model.

Stratification - is the process to ensure that the datasets are divided in to the Training and Validations sets that contain an approxinmately equal proprtion of the target event. Ex: to make sure that the if the 10% of the obs in Training data contains target, the 10% of the validation dataset also contains the target. We can use `PROC SURVEYSELECT` to split the dataset and do stratification.

```
PROC SURVEYSELECT data= <dataset>
                  seed=555
                  samprate=.667
                  out= <output_dataset>
                  outall;
 STRATA <var_to_use_for_stratification>
 RUN;
```

Rare target events : when the target event is rare, we can use bootstraping or k-fold cross validation to generate the training and validation datasets. We could also use oversampling and then adjust for the oversampling by using an offset

- **Bootstraping** is repeated sampling with repleacement. Multiple samples are taken with the response variable and models fitting is done and the model statistics are averages across the models.
- **K-Fold cross validation** - is partitioning the data into k partitions. then models are trained using k-1 parts and validated on the one part that was not used for training. It does not give the final model - the final model is created by fitting the data to the entire dataset.

Classifier Performance

Classifier performance is an indication of how well our model performs with respect to known data. Any logistic model (predicting a binary target) will have some False Positives, False Negatives, True Positives and True Negatives. A model that minimizes the false positives and false negatives is a better model.

Consider the following confusion matrix (table with predicted binary reponses and actual binary responses)

	Predicted True	**Predicted False**	**Total**
Actual True	100 (TP)	8 (FN)	108
Actual False	2 (FP)	50 (TN)	52
Total	102	58	160

- **True Positive (TP)** - The model predicted a positive result, and the result as infact positive
- **False Positive (FP)** - The model predicted a positive result, but the actual result as negaive. Disease model predits a patient will have a disease but he does not.
- **True Negative (TN)** - The model predicts a negative result and the result is infact negative.

- **False Negative (FN)** - The model predicts a negative result, but the actual result is positive. Disease model predicts no disease, but the person actually is diseased.
- **Accuracy**: Overall, how often is the classifier correct?
  - (TP+TN)/total = (100+50)/160 = 0.94
- **Misclassification Rate**: Overall, how often is it wrong?
  - (FP+FN)/total = (8+2)/160 = 0.06
  - equivalent to 1 minus Accuracy also known as "Error Rate"
- **Sensitivity / True Positive Rate**: When it's actually yes, how often does it predict yes?
  - TP/actual yes = 100/108 = 0.93
  - also known as "Recall"
- **False Positive Rate**: When it's actually no, how often does it predict yes?
  - FP/actual no = 2/52 = 0.04
- **Specificity / True Negative Rate**: When it's actually no, how often does it predict no?
  - TN/actual no = 50/52 = 0.96
  - equivalent to 1 minus False Positive Rate
- **Positive Predicted value or PV+ / Precision**: When it predicts yes, how often is it correct?
  - TP/predicted yes = 100/102 = 0.98
- **Prevalence**: How often does the yes condition actually occur in our sample?
  - actual yes/total = 102/160 = 0.63

Cutoff - Each cutoff will generate a different confusion matrix. When the cutoff is low, we classify more obs as positive - so the sensitivity is higher (we catch more of the true positives) at the cost of specificity (we inadvertedly catch a lot of False positives as well.)

ROC curve & AUC

ROC Curve is plotted between the TPR/Sensitivity (y-axis) vs. False Positive Rate/ 1-Sepcificity (x-axis). The curve shows the sensitivity and false positive rates for the whole range of cutoff values (0-1). You set a thereshold to minimize False Positives, or maximize true poistives. If the model is very discriminating, then we would have a good TPR for a low threshold of FPR. The AUC is the are under th curve and ranges from .5 (diagonal) to 1 (totally hugs the left and top edges), and the larger the area, the more discrininating the model. So better models have a ROC that hugs the top left corner.

Gains Chart and Lift Charts

- visual aid to measure the efficetivness of a model. Compares the reponses with a model to without a model.
- Maps the positive predicted value - PV+ (TP/Pred.+) - to the depth. Depth is the percentage of Positives selected by the cutoff. Ex: if the cutoff is 0, then all values are predicted positive - 100%. If the cutoff is 1, then all values are predicted as negative - percentage of positives selected = 0%.
- So that the cutoff increases - model becomes more selective, the depth decreases. You are targeting fewer but surer observations.
- For a good predictive model the positive predictive value increases as he depth decreases.
- Lift Charts - they are similar to the gains chart and tells us how much the the prediction model is better than random chance.
- This looks similar to the gains chart, and tells us that the smaller percentage of cases we target the higher out lift will be when using a good model.
- Lift and cumulative gains charts differ in only the Y-axis scale. Cumulative gains is plotted with the Positive Predicted Value to the depth, while Lift is plotted with the PV+/(random chance). So the

lift number gives you the factor by which the model improves the reponse.

Profit Prediction

Optimal cutoff for the logistic regression model is usually based on the profit. Use a profit matrix to find the optimal cutoff. If the cost of an offer is $1 and return is $100, for every predicted positive, we spend $1, and from those, if the actual response is positive, we make $100, netting a profit of $99. If the predicted positive is a false positive, we lose $1.

	Predicted True	Predicted False
Actual True	$99 TPProfit	0
Actual False	$-1 FPLoss	0

When combined with the actual confusion matrix for varisous cutoofs, we can see the net profit. This will be (TP * TPProfit) - (FP*FPLoss)

Bayes Rule

To find the most profitable cutoff across all possible cutoffs, the the Bayes rule

p = 1/1+[(TPProfit- FNProfit) / (TrueNegativeProfit - FPLoss)]

in the above case , it's

p = 1/1+[(99 - 0) / (0 - -1)]

p = 1/1+ (99/+1) = 1/100 = 0.01

In cases where the profit information is not known, the central cutoff is used. this is the meeting point of the Sensitivity and Specificity - the value at wich they meet (the point at which they are equal)

The average profit can be graphed against either the cutoff (to see how setting a cutoff will affect the average profit) or the depth (the graph will show a curve that maps profit as the depath increases, and after a point the the cost with diminish the returns)

Kolmogorov - Smirnov Statistic

Is a measure of the overall performance of a model. Its frequently used in financial analysis.

One way to check the predictive power is to see the discriminating power of the model. To compare two samples, you can use the 2-sample T test. however, the T test assumes the normality of the variances, which cannot be guantanteed when we are discussing the distribution of events vs non events. They will have different variances - the alternative is to use the KS-Statistic.

SAS uses the `D-Statistic` in the `PROC NPAR1WAY` with the `EDF` option. The higher value of D-stat, the better the model. The KS-Test tests the shape, variance and central tendency of the distribution. For model evaluation central tendency is the most important.

The Wilcoxon-Mann-Whitney test is a better test for central tendency, and its value is the same as that of the C-Statistic (area under the ROC curve or AUC) in the `PROC LOGISTIC` .

Both the KS-Statistic and the C-Statistic are unaffected by oversampling because teh emperical cumulative distribution function is unchanged when each case represents more than one case in the population.

```
PROC NPAR1WAY edf data= <dataset>;
  CLASS binary_target;
  VAR Probabilities;
RUN;
```

Comparing models using plots

A popular way to compare models is to score the data with many models in increasing complexinty and then choosing the most parsimonious one. You could plot the fit statistics of each model in increasing complexity, for the training and validation datasets. When we do this we notice that the fit and complexity increase initially as valid terms are added to the model and then after a point the fit does not improve with increasing complexity (the point at which the new terms added do not seem to have much predictive power). Interestingly, after this point the fit may improve for the traning dataset and the same model the fit drops for the validation dataset - a clear indication that the model is being overfit and generalization being lost.

This difference/divergence in the curves is called **shrinkage** and we can select model that minimize or limit shrinkage to strike a balance between the generalization, predictive power and the complexity of the model.