Summary of Preparing Data

CAS Action Overview

When you program using CAS, there are multiple ways to interact with the CAS server to process data. You can use:

- CAS-enabled PROCS,
- traditional DATA step programs,
- and open-source languages.

All of these techniques leverage the CAS API to convert native language elements to CAS actions.

Modifying Tables

- Update rows in a table:

```
table.update /
    table={castable},
    set={
          var="column-name", value="expression"},
          <{var-n={"column-name-n", value-n="expression"},>
     } ...;
```

- Copy one table to another:

```
table.copyTable /
    table={castable},
    casout={casout-table};
```

Preparing Data

- Add a format to a computed column:

```
...
table = {...
          computedVars={
                              name="variable-name",
                              format="string",
                              label="string"...},
                              { ... }
} ...
```

- Create a calculated column:

```
table={...
      computedVarsProgram="expression(s)"
```

```
};
```

- Convert a character value to numeric:

```
INPUTN(source, informat)
```

- Run DATA step code in CAS using the CAS action:

```
dataStep.runCode /
    code="string";
```

- Alter CAS table metadata:

```
table.alterTable /
      caslib="string", name="table-name",
      rename="string",
      label="string",
      drop={column-names},
      keep={column-names},
      columns={
                  {AlterTableColumn-1}
                  <,{AlterTableColumn-n}, ...>
     };
```

- Impute missing values:

```
dataPreprocess.impute /
      table={castable},
      casout={casouttable},
      inputs={
                {column-to-impute-1}
               <,{column-to-impute-n}, ...>
      }
      <copyAllVars=FALSE | TRUE,>
      <methodInterval="impute-method",>
      <methodNominal="impute-method">;
```

- Transpose a CAS table:

```
transpose.transpose /
      table={ name="table-name", caslib="caslib", groupBy="column"},
      casOut={casouttable},
      label="string",
      transpose={"column-name-1" <,"column-name-n" ...>},
      ID={"column-name-1" <,"column-name-n" ...>};
```

- Calculate quantiles, high and low whiskers, and outliers:

```
percentile.boxPlot /
      table={castable},
```

inputs={column-names},
casOut={*casouttable*},
<, additional parameters},

- Execute SQL in CAS:

fedSQL.execDirect / query="*sql-query*";