Reetika Sarkar – JMP Clinical

About Me

- PhD candidate (Year 4) in Computational Mathematics (Statistics) at UNC Greensboro
- Started at SAS in May 2024 as a Graduate Intern in JMP Clinical Group reporting to Mann Geoffrey
- Goal: Full-time Statistician/Developer role working in the Technology/Pharmaceutical Industry

Projects

- **Validation of JMP Clinical safety evaluation reports to enhance reliability of the software**
  - Reports validated: Adverse Events (AE), Medical History (MH), Concomitant Medications (CM), Acute Kidney Injury (AKI), and Drug Induced Liver injury (DILI)
  - Dataset used: Nicardipine (a drug used to treat hypertension)
- **Analysis of Vaccine Adverse Events Reporting Systems (VAERS) safety data using JMP Clinical 19**
- **SAS/JMP Courses and Trainings:**
  - SAS® Programming 1: Essentials
  - SAS® Programming 2: Data Manipulation Techniques
  - SAS® Programming for R Users
  - Statistical Thinking for Industrial Problem Solving (STIPS)
  - Introduction to the JMP Scripting Language (JSL)
- **Hands-on Training with Team**

Scan to view my LinkedIn profile

Contact Information

JMP, SAS email: Reetika.Sarkar@jmp.com | University email: rsarkar@uncg.edu

Personal email: rsarkar2@icloud.com

Double Programming and Unit Testing for Validation of JMP Clinical Safety Analysis Reports pertaining to Events, Findings, and Interventions

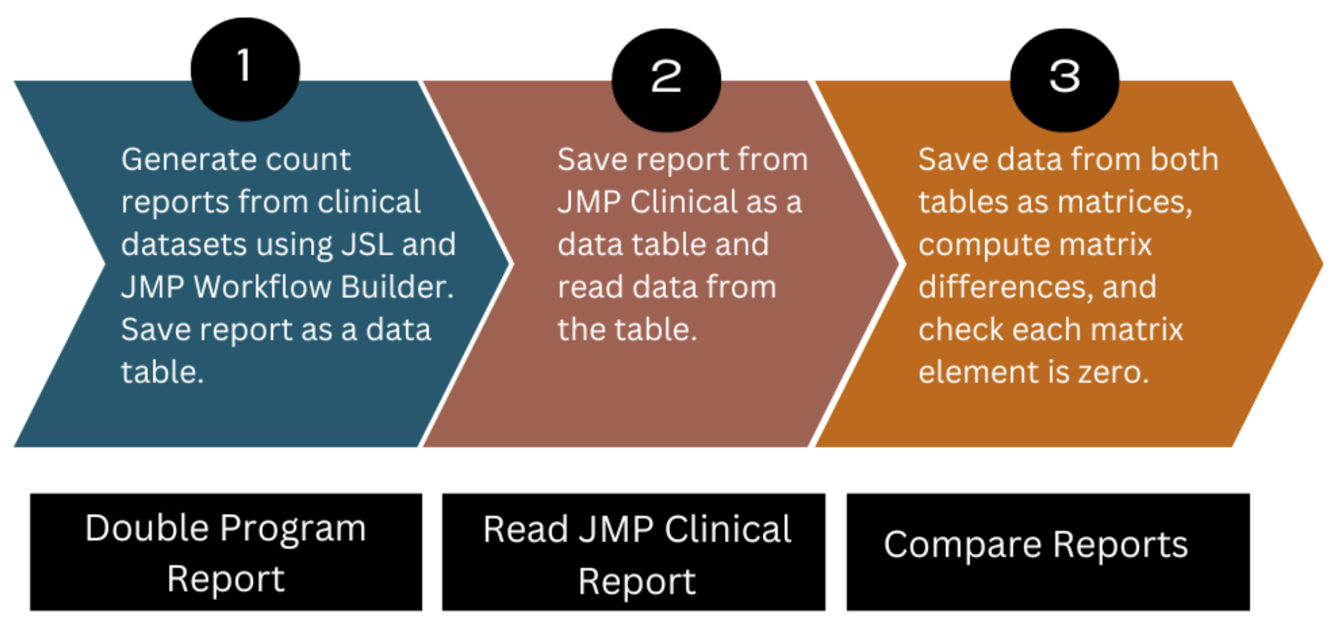
Challenge

Drug safety assessment in a clinical trial includes meticulous evaluation of a drug's potential side effects with the goal to ensure that the benefits outweigh the potential risks. It includes analysis of events, interventions, and findings data across treatment arms to report adverse findings. The goal of this project was to double program the reports mentioned above using JMP Scripting Language (JSL) and compare the findings to the reports generated in JMP Clinical.

Implications / Applications

The reports double programmed in this project would help in establishing a framework that validates the clinical safety reports generated using JMP Clinical. These reports would be made available to the users of JMP Clinical.

Workflow

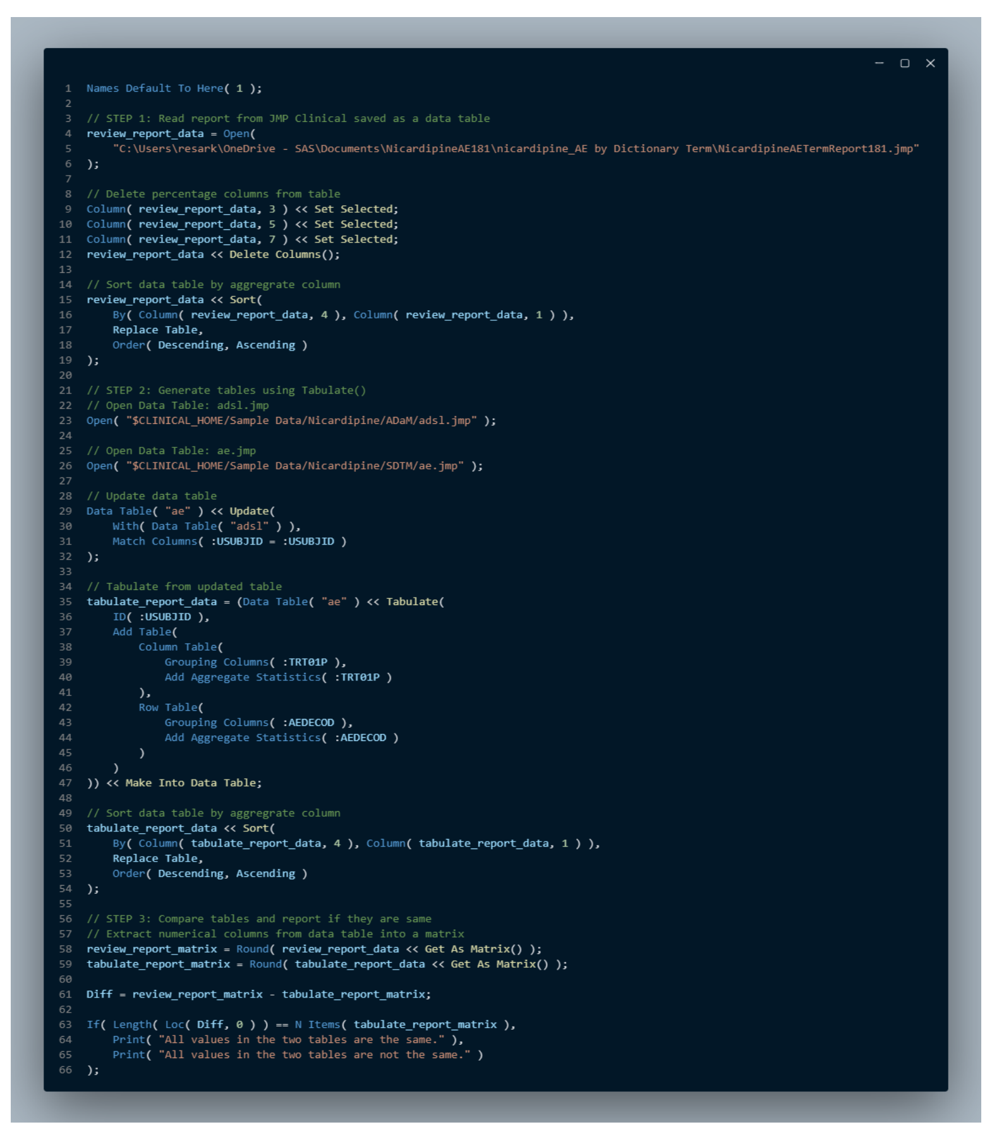


1. Generate count reports from clinical datasets using JSL and JMP Workflow Builder. Save report as a data table.
2. Save report from JMP Clinical as a data table and read data from the table.
3. Save data from both tables as matrices, compute matrix differences, and check each matrix element is zero.

JMP Functionalities and JSL Code Overview

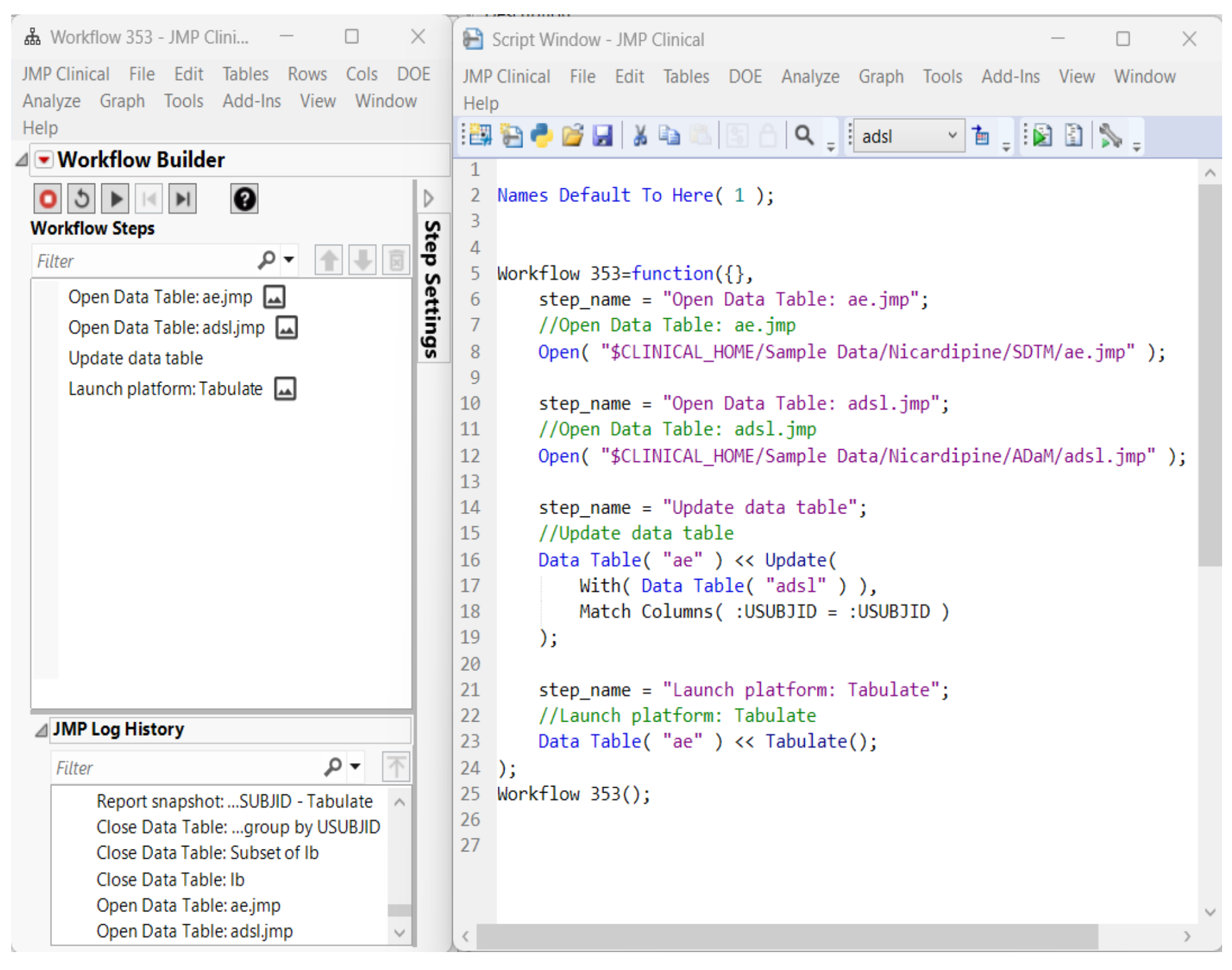
- Data Table: Open, Format, Tabulate, Subset, Summary, Transpose, Update, Get As Matrix, Select Matching Cells, Select Where, New Column, Set Selected Column, Delete Column, Exclude
- Statistical: Max
- Matrix: Loc
- Programming: Length, N Items, Show
- Conditional: If, For Each Row
- JMP Workflow Builder

Example Code for AE Report



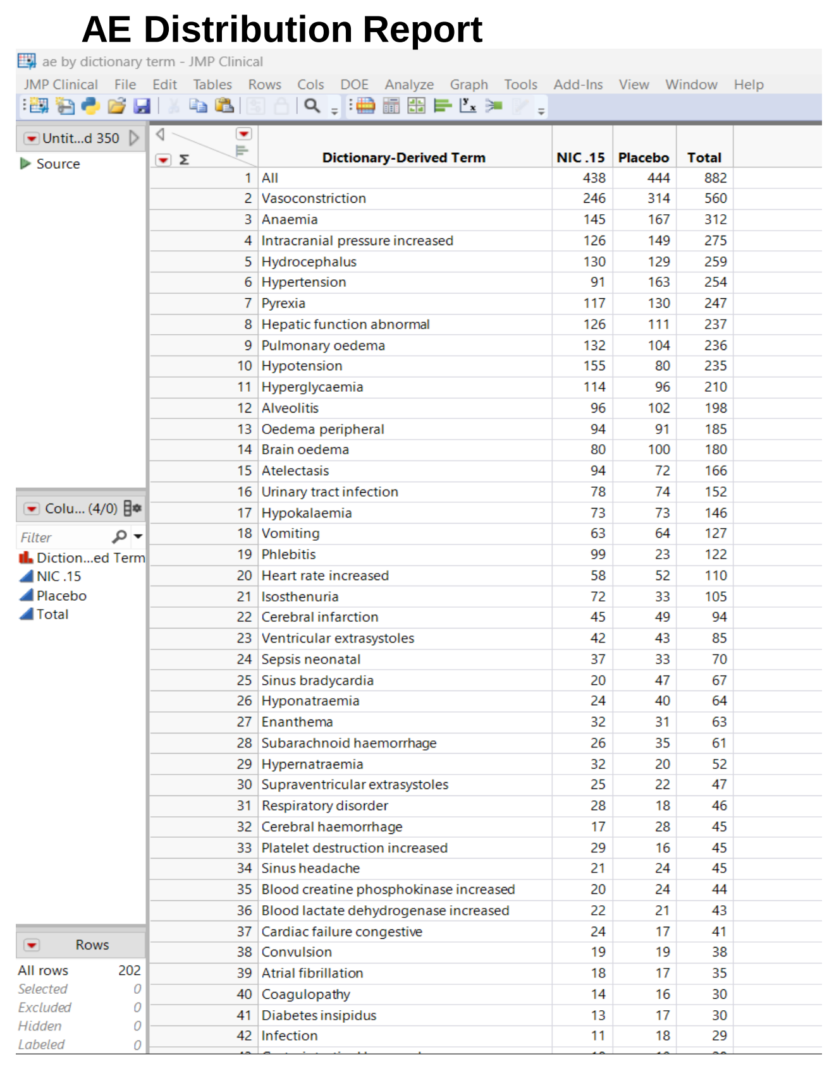
```
Names Default To Here( 1 );

// STEP 1: Read report from JMP Clinical saved as a data table
review_report_data = Open(
    "C:\Users\resark\OneDrive - SAS\Documents\NicardipineAE181\nicardipine_AE by Dictionary Term\NicardipineAETermReport181.jmp"
);

// Delete percentage columns from table
Column( review_report_data, 3 ) << Set Selected;
Column( review_report_data, 5 ) << Set Selected;
Column( review_report_data, 7 ) << Set Selected;
review_report_data << Delete Columns();

// Sort data table by aggregate column
review_report_data << Sort(
    By( Column( review_report_data, 4 ), Column( review_report_data, 1 ) ),
    Replace Table,
    Order( Descending, Ascending )
);

// STEP 2: Generate tables using Tabulate()
// Open Data Table: adsl.jmp
Open( "$CLINICAL_HOME/Sample Data/Nicardipine/ADaM/adsl.jmp" );

// Open Data Table: ae.jmp
Open( "$CLINICAL_HOME/Sample Data/Nicardipine/SDTM/ae.jmp" );

// Update data table
Data Table( "ae" ) << Update(
    With( Data Table( "adsl" ) ),
    Match Columns( :USUBJID = :USUBJID )
);

// Tabulate from updated table
tabulate_report_data = (Data Table( "ae" ) << Tabulate(
    ID( :USUBJID ),
    Add Table(
        Column Table(
            Grouping Columns( :TRT01P ),
            Add Aggregate Statistics( :TRT01P )
        ),
        Row Table(
            Grouping Columns( :AEDECOD ),
            Add Aggregate Statistics( :AEDECOD )
        )
    )
)) << Make Into Data Table;

// Sort data table by aggregate column
tabulate_report_data << Sort(
    By( Column( tabulate_report_data, 4 ), Column( tabulate_report_data, 1 ) ),
    Replace Table,
    Order( Descending, Ascending )
);

// STEP 3: Compare tables and report if they are same
// Extract numerical columns from data table into a matrix
review_report_matrix = Round( review_report_data << Get As Matrix() );
tabulate_report_matrix = Round( tabulate_report_data << Get As Matrix() );

Diff = review_report_matrix - tabulate_report_matrix;

If( Length( Loc( Diff, 0 ) ) == N Items( tabulate_report_matrix ),
    Print( "All values in the two tables are the same." ),
    Print( "All values in the two tables are not the same." )
);
```

JMP Workflow Builder



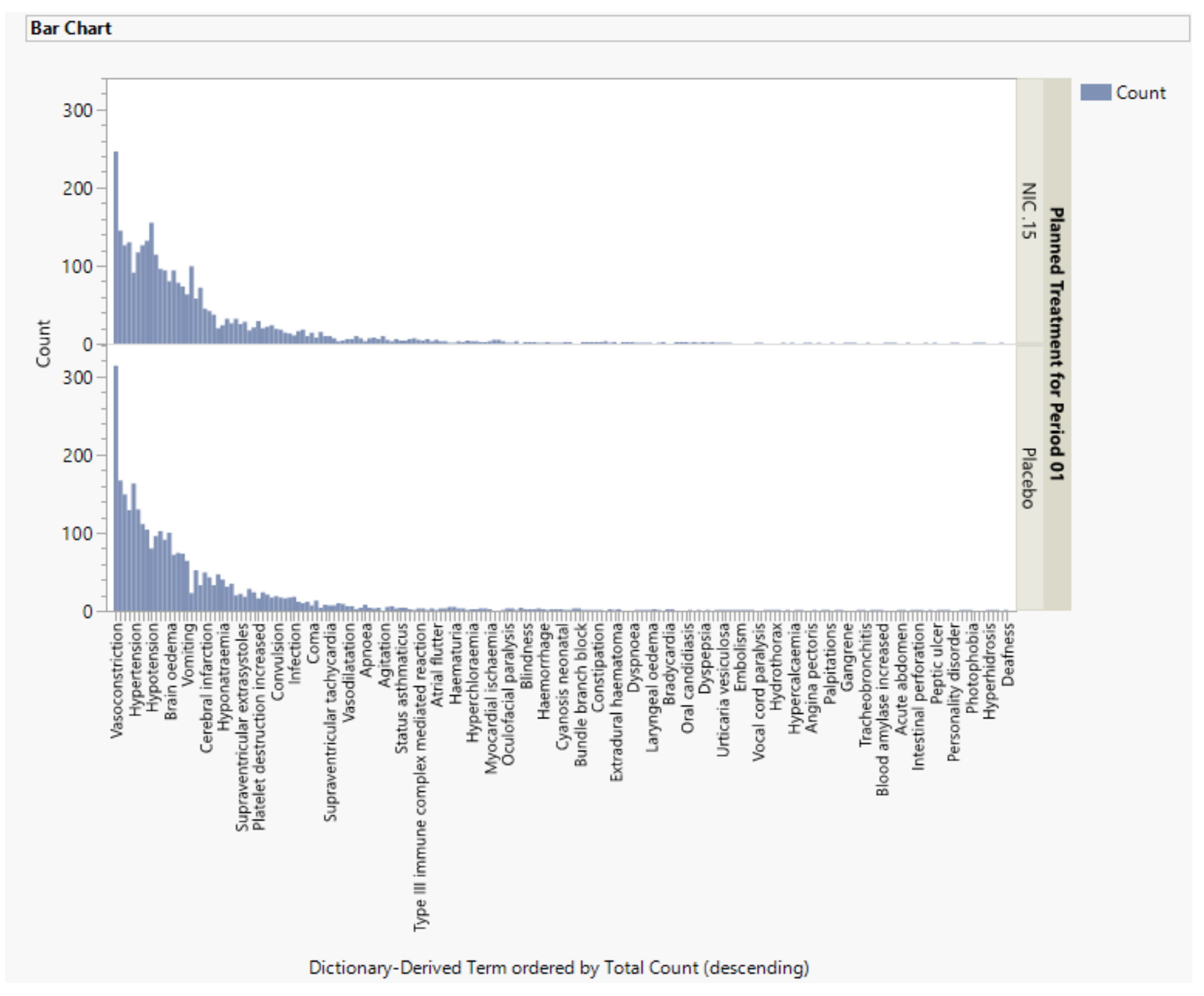
```
Names Default To Here( 1 );


Workflow 353=function({},
    step_name = "Open Data Table: ae.jmp";
    //Open Data Table: ae.jmp
    Open( "$CLINICAL_HOME/Sample Data/Nicardipine/SDTM/ae.jmp" );

    step_name = "Open Data Table: adsl.jmp";
    //Open Data Table: adsl.jmp
    Open( "$CLINICAL_HOME/Sample Data/Nicardipine/ADaM/adsl.jmp" );

    step_name = "Update data table";
    //Update data table
    Data Table( "ae" ) << Update(
        With( Data Table( "adsl" ) ),
        Match Columns( :USUBJID = :USUBJID )
    );

    step_name = "Launch platform: Tabulate";
    //Launch platform: Tabulate
    Data Table( "ae" ) << Tabulate();
);
Workflow 353();
```

AE Distribution Report

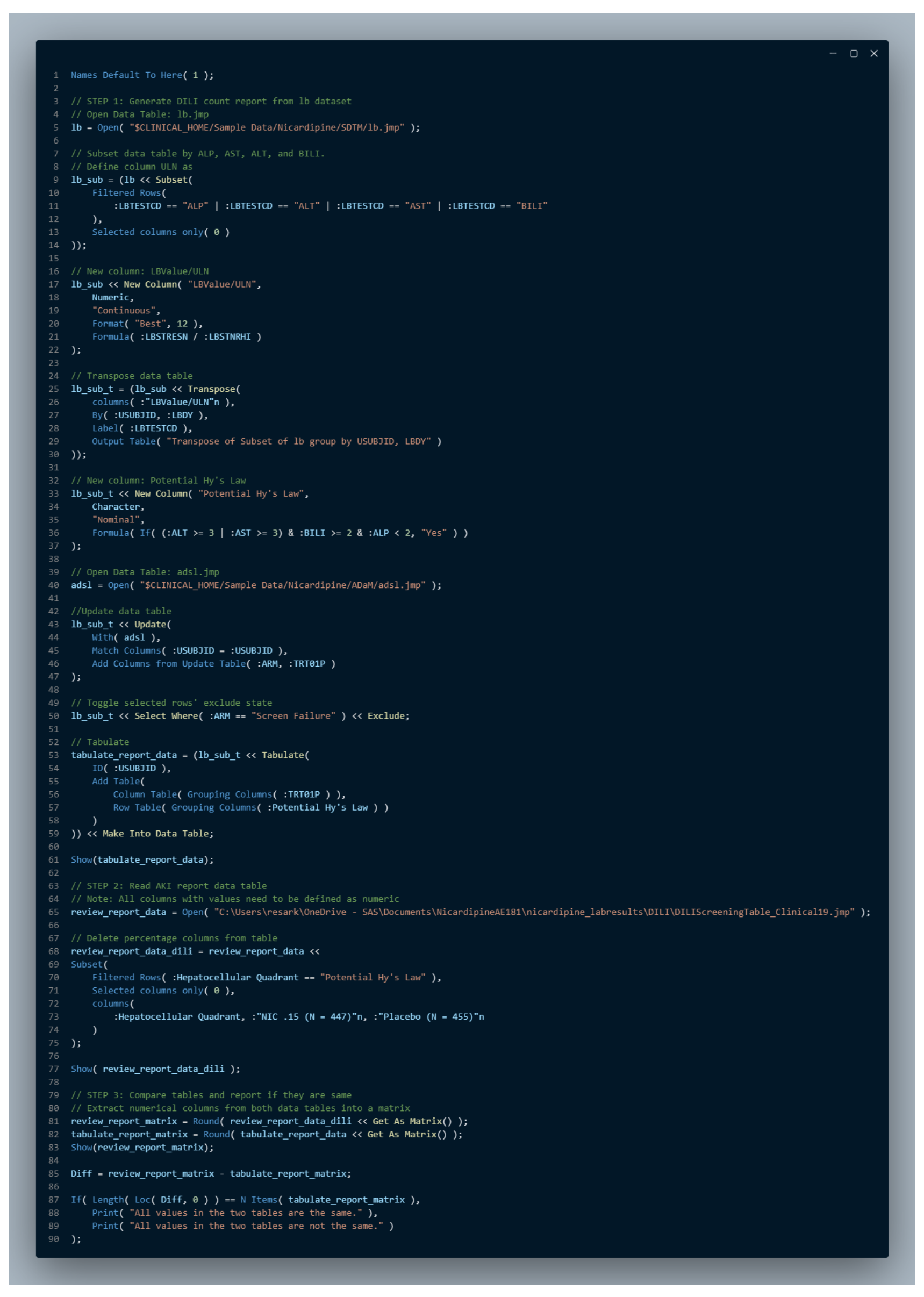


	Dictionary-Derived Term	NIC .15	Placebo	Total
1	All	438	444	882
2	Vasoconstriction	246	314	560
3	Anaemia	145	167	312
4	Intracranial pressure increased	126	149	275
5	Hydrocephalus	130	129	259
6	Hypertension	91	163	254
7	Pyrexia	117	130	247
8	Hepatic function abnormal	126	111	237
9	Pulmonary oedema	132	104	236
10	Hypotension	155	80	235
11	Hyperglycaemia	114	96	210
12	Alveolitis	96	102	198
13	Oedema peripheral	94	91	185
14	Brain oedema	80	100	180
15	Atelectasis	94	72	166
16	Urinary tract infection	78	74	152
17	Hypokalaemia	73	73	146
18	Vomiting	63	64	127
19	Phlebitis	99	23	122
20	Heart rate increased	58	52	110
21	Isosthenuria	72	33	105
22	Cerebral infarction	45	49	94
23	Ventricular extrasystoles	42	43	85
24	Sepsis neonatal	37	33	70
25	Sinus bradycardia	20	47	67
26	Hyponatraemia	24	40	64
27	Enanthema	32	31	63
28	Subarachnoid haemorrhage	26	35	61
29	Hypernatraemia	32	20	52
30	Supraventricular extrasystoles	25	22	47
31	Respiratory disorder	28	18	46
32	Cerebral haemorrhage	17	28	45
33	Platelet destruction increased	29	16	45
34	Sinus headache	21	24	45
35	Blood creatine phosphokinase increased	20	24	44
36	Blood lactate dehydrogenase increased	22	21	43
37	Cardiac failure congestive	24	17	41
38	Convulsion	19	19	38
39	Atrial fibrillation	18	17	35
40	Coagulopathy	14	16	30
41	Diabetes insipidus	13	17	30
42	Infection	11	18	29

JMP Clinical AE Distribution visualization

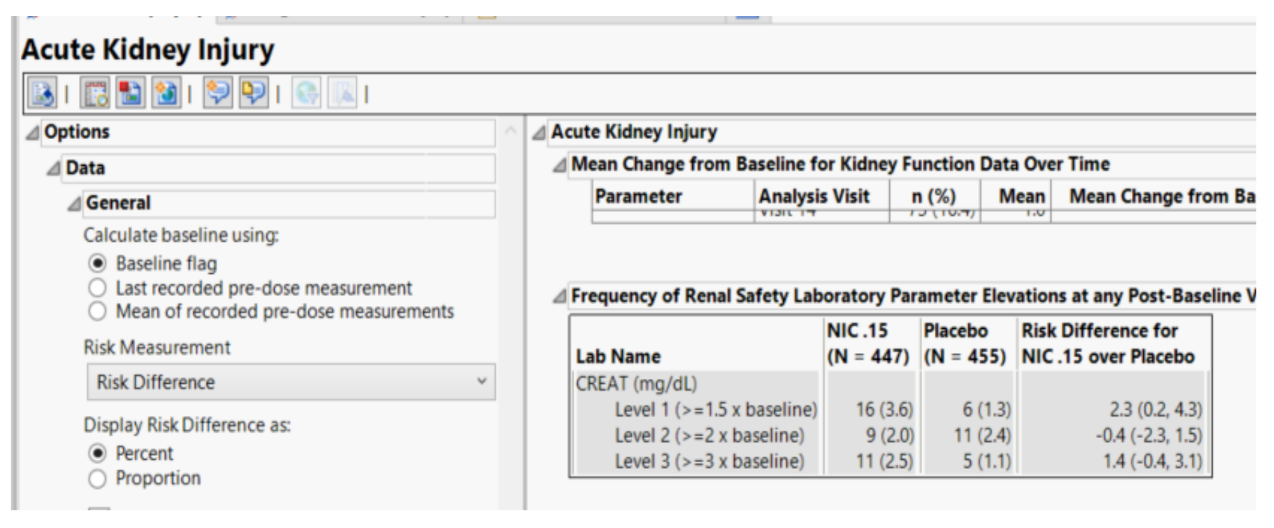


Example Code for AKI/DILI Report



```
Names Default To Here( 1 );

// STEP 1: Generate DILI count report from lb dataset
// Open Data Table: lb.jmp
lb = Open( "$CLINICAL_HOME/Sample Data/Nicardipine/SDTM/lb.jmp" );

// Subset data table by ALP, AST, ALT, and BILI.
// Define column ULN as
lb_sub = (lb << Subset(
    Filtered Rows(
        :LBTESTCD == "ALP" | :LBTESTCD == "ALT" | :LBTESTCD == "AST" | :LBTESTCD == "BILI"
    ),
    Selected columns only( 0 )
));

// New column: LBValue/ULN
lb_sub << New Column( "LBValue/ULN",
    Numeric,
    "Continuous",
    Format( "Best", 12 ),
    Formula( :LBSTRESN / :LBSTNRHI )
);

// Transpose data table
lb_sub_t = (lb_sub << Transpose(
    columns( :"LBValue/ULN"n ),
    By( :USUBJID, :LBDY ),
    Label( :LBTESTCD ),
    Output Table( "Transpose of Subset of lb group by USUBJID, LBDY" )
));

// New column: Potential Hy's Law
lb_sub_t << New Column( "Potential Hy's Law",
    Character,
    "Nominal",
    Formula( If( (:ALT >= 3 | :AST >= 3) & :BILI >= 2 & :ALP < 2, "Yes" ) )
);

// Open Data Table: adsl.jmp
adsl = Open( "$CLINICAL_HOME/Sample Data/Nicardipine/ADaM/adsl.jmp" );

//Update data table
lb_sub_t << Update(
    With( adsl ),
    Match Columns( :USUBJID = :USUBJID ),
    Add Columns from Update Table( :ARM, :TRT01P )
);

// Toggle selected rows' exclude state
lb_sub_t << Select Where( :ARM == "Screen Failure" ) << Exclude;

// Tabulate
tabulate_report_data = (lb_sub_t << Tabulate(
    ID( :USUBJID ),
    Add Table(
        Column Table( Grouping Columns( :TRT01P ) ),
        Row Table( Grouping Columns( :Potential Hy's Law ) )
    )
)) << Make Into Data Table;

Show(tabulate_report_data);

// STEP 2: Read AKI report data table
// Note: All columns with values need to be defined as numeric
review_report_data = Open( "C:\Users\resark\OneDrive - SAS\Documents\NicardipineAE181\nicardipine_labresults\DILI\DILIScreeningTable_Clinical19.jmp" );

// Delete percentage columns from table
review_report_data_dili = review_report_data <<
Subset(
    Filtered Rows( :Hepatocellular Quadrant == "Potential Hy's Law" ),
    Selected columns only( 0 ),
    columns(
        :Hepatocellular Quadrant, :"NIC .15 (N = 447)"n, :"Placebo (N = 455)"n
    )
);

Show( review_report_data_dili );

// STEP 3: Compare tables and report if they are same
// Extract numerical columns from both data tables into a matrix
review_report_matrix = Round( review_report_data_dili << Get As Matrix() );
tabulate_report_matrix = Round( tabulate_report_data << Get As Matrix() );
Show(review_report_matrix);

Diff = review_report_matrix - tabulate_report_matrix;

If( Length( Loc( Diff, 0 ) ) == N Items( tabulate_report_matrix ),
    Print( "All values in the two tables are the same." ),
    Print( "All values in the two tables are not the same." )
);
```

JMP Clinical AKI Report



Acute Kidney Injury

Calculate baseline using: [✓ Baseline flag] Last recorded pre-dose measurement / Mean of recorded pre-dose measurements

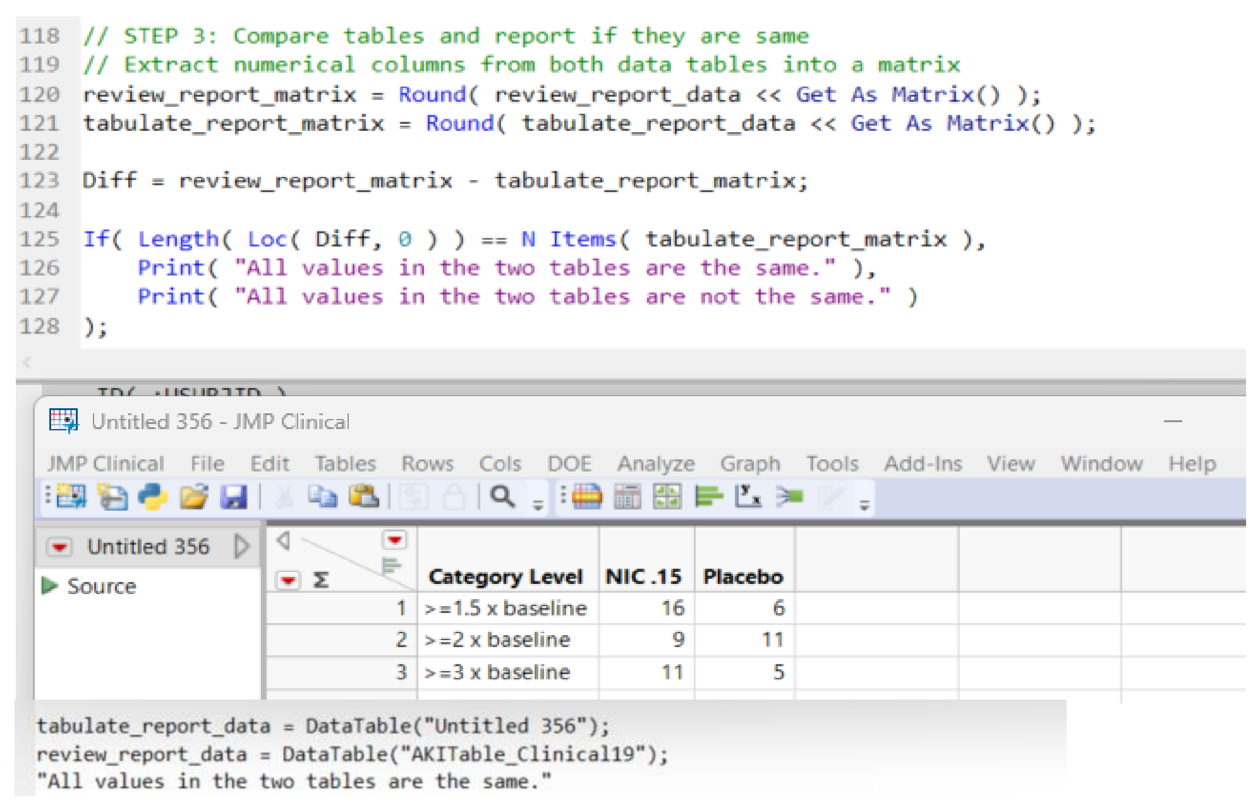
Risk Measurement: Risk Difference

Display Risk Difference as: [✓ Percent] Proportion

Frequency of Renal Safety Laboratory Parameter Elevations at any Post-Baseline V

Lab Name	NIC .15 (N = 447)	Placebo (N = 455)	Risk Difference for NIC .15 over Placebo
CREAT (mg/dL)			
Level 1 (>=1.5 x baseline)	16 (3.6)	6 (1.3)	2.3 (0.2, 4.3)
Level 2 (>=2 x baseline)	9 (2.0)	11 (2.4)	-0.4 (-2.3, 1.5)
Level 3 (>=3 x baseline)	11 (2.5)	5 (1.1)	1.4 (-0.4, 3.1)

Double programmed AKI Report tabulation and comparison



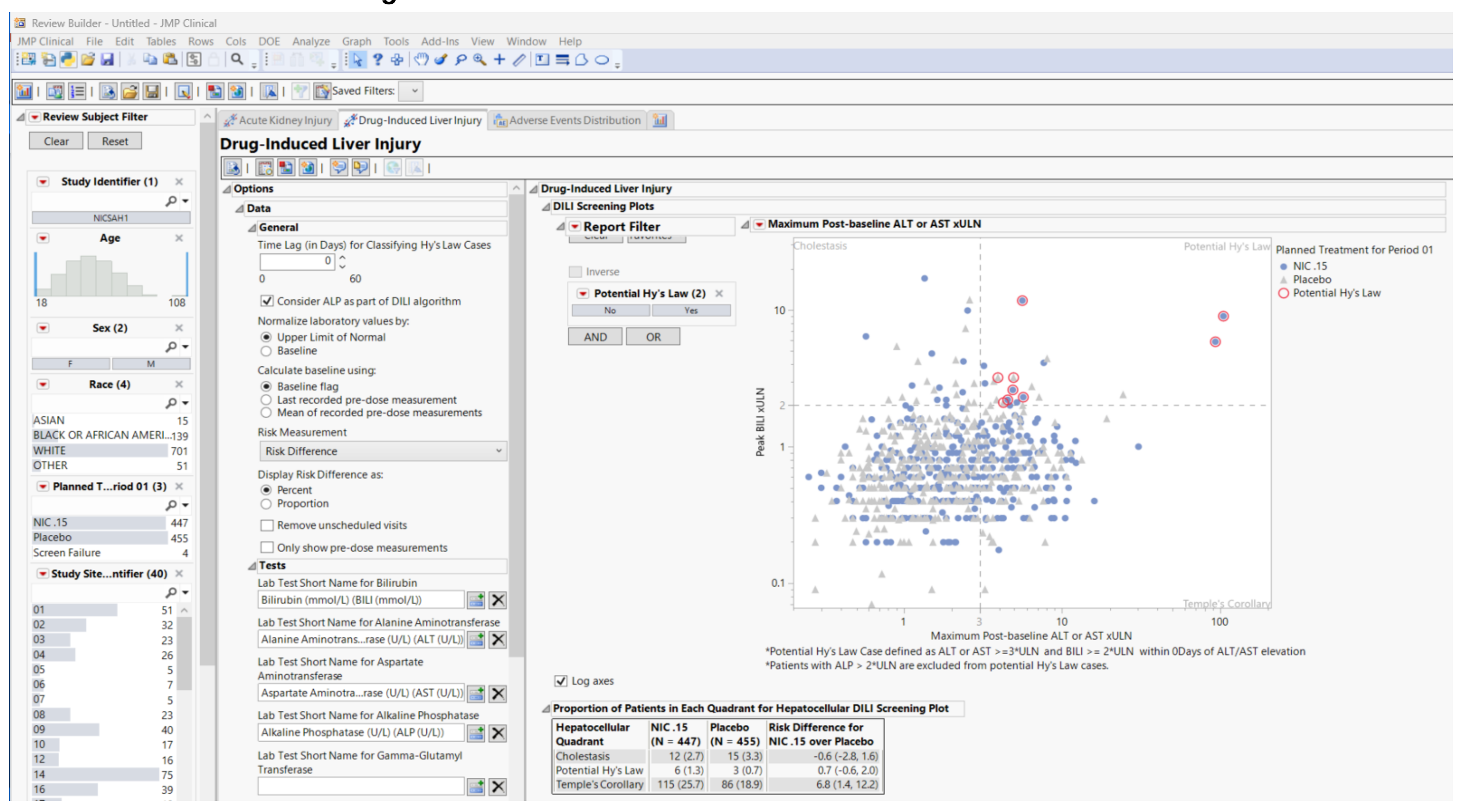
```
// STEP 3: Compare tables and report if they are same
// Extract numerical columns from both data tables into a matrix
review_report_matrix = Round( review_report_data << Get As Matrix() );
tabulate_report_matrix = Round( tabulate_report_data << Get As Matrix() );

Diff = review_report_matrix - tabulate_report_matrix;

If( Length( Loc( Diff, 0 ) ) == N Items( tabulate_report_matrix ),
    Print( "All values in the two tables are the same." ),
    Print( "All values in the two tables are not the same." )
);
```

Untitled 356 - JMP Clinical

	Category Level	NIC .15	Placebo
1	>=1.5 x baseline	16	6
2	>=2 x baseline	9	11
3	>=3 x baseline	11	5

```
tabulate_report_data = DataTable("Untitled 356");
review_report_data = DataTable("AKITable_Clinical19");
"All values in the two tables are the same."
```

JMP Clinical DILI Screening Plots



Review Builder - Untitled - JMP Clinical
Drug-Induced Liver Injury
DILI Screening Plots
Maximum Post-baseline ALT or AST xULN
Peak BILI xULN
Potential Hy's Law
Temple's Corollary
Cholestasis
Planned Treatment for Period 01
NIC .15
Placebo
Proportion of Patients in Each Quadrant for Hepatocellular DILI Screening Plot