

# Season Ticket Sale Renewal Report

David Mullaly

2026-02-20

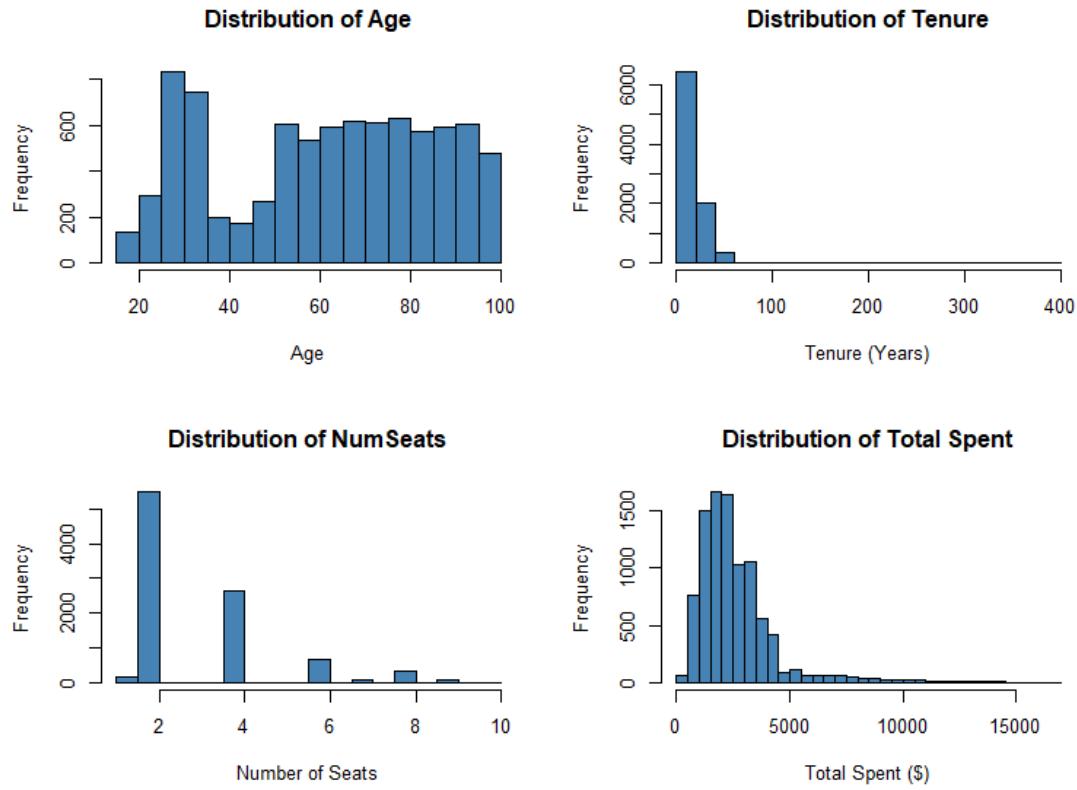
## Data Loading and Wrangling

Load four relational tables and merge into a single flat file (FFdf) using left joins on Cust\_ID.

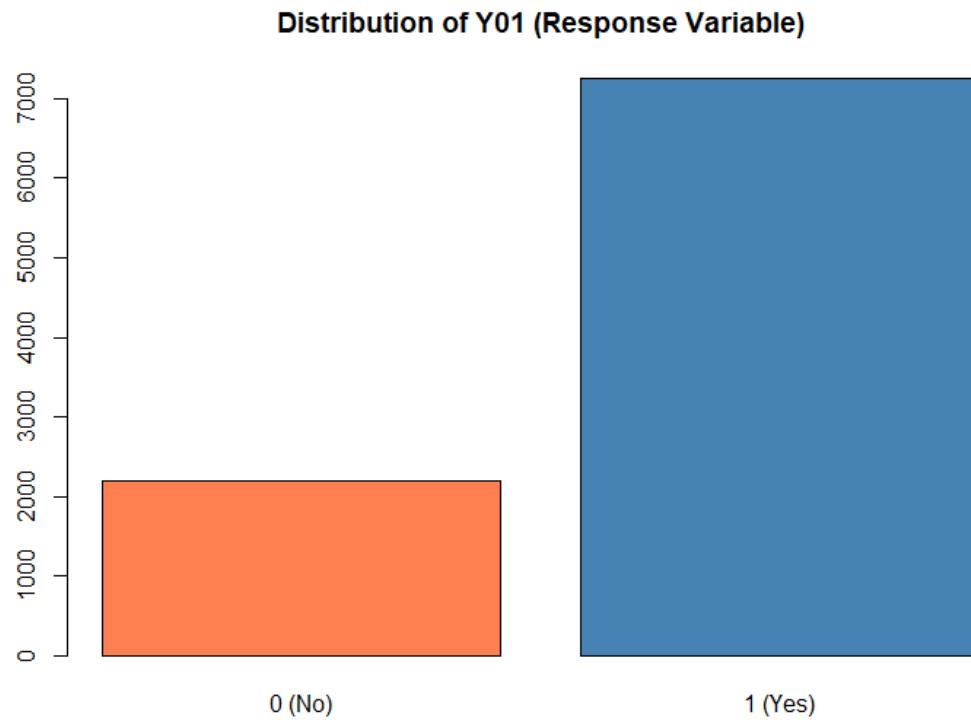
```
## MainDF: 9447 48 | StoreDF: 9272 3 | ConcessDF: 9272 3 | CustomerDF: 14272  
7  
## Duplicates in MainDF: 175  
## Duplicates in StoreDF: 0  
## Duplicates in ConcessDF: 0  
## Duplicates in CustomerDF: 0  
## Final flat file dimensions - Rows: 9447 Columns: 58  
## Unique Cust_ID: 9272 | Duplicate rows: 175
```

## Step 1: Open Data in Your Software of Choice

Create identifier variable, arrange columns, and explore distributions visually.



```
## Response Variable (Y01): 2196 7251
```



**Observations:** Age is roughly normal (30-60), Tenure is right-skewed, NumSeats clusters around 2-4, Total Spent is heavily right-skewed. Response variable Y01 is reasonably balanced.

## Step 2: Review Variables for Common Sense (SME Knowledge)

Standardize variable names and check for unique identifiers.

```
## Dataset: 9447 rows, 59 columns
## Unique Cust_ID: 9272 out of 9447 rows
```

**Result:** Each row does not necessarily represent a unique customer (175 duplicate Cust\_IDs found in MainDF source data).

## Step 3: Review How Software Coded Variables

Convert character variables to factors for proper categorical analysis.

```
## Character variables to convert: 23
##
## Sex levels: F M
## Marital levels: D M S U
```

```
## Account_Type levels: Business Personal Shared
```

**Note:** Marital has levels D (Divorced), M (Married), S (Single), U (Unknown). The "U" for Unknown may need special handling.

## Step 4: Data Integrity/Validation Checks

Check for anomalies, bogus values, and data quality issues.

```
## Age range: 18 99
## Tenure range: 1 400
## NumSeats range: 1 10
##
## DistA values = 999: 1506
## DistA NA count after conversion: 2895
##
## Survey_Comp range: 0 7.4
## Survey_Comp values > 1: 110
##
## State_Name unique: 50 | State_Loc unique: 50
```

### Issues Found:

- **DistA = 999:** Placeholder values converted to NA
- **Survey\_Comp > 1:** Outlier found when expected range is 0-1
- **Age max = 99:** May be placeholder or extreme value - verify with SME
- **Tenure max = 400:** Suspicious value if measured in years - verify units with SME
- **State\_Name/State\_Loc:** Redundant columns (same info, different format)
- **Marital = "U":** Unknown status - consider treating as NA
- **Cust\_ID duplicates:** 175 duplicate IDs found in MainDF source data
- **Address, Name, PhoneNum:** 100% missing - likely removed from CustomerDF for privacy

## Step 5: Handle Dates

Convert Last\_Contact datetime and extract useful components.

```
## Date components extracted: Contact_Year, Contact_Month, Contact_Day,
## Contact_Weekday, Contact_Hour
## Contact Year range: 2018 2025
## Contact Hour range: 0 23
```

## Summary

```
## Final Dataset: 9447 rows x 64 columns
```

```
## Total Missing Values: 94591
```

```
##
```

```
## Columns with missing values:
```

##	Address	Name	PhoneNum
##	9447	9447	9447
##	Educational_Level	Favorite_Caps_Player	Favorite_Sport
##	6612	6612	6612
##	Job_Sector	Mode_Of_Transport	Team_B_STH
##	6612	6612	6612
##	Team_C_STH	Net_Worth_True	HouseHold_Income_True
##	6612	5762	5691
##	DistA	Marital	Age
##	2895	1155	986
##	Rep_Name	Sex	Tenure
##	871	667	592
##	Rep_Visits	Rep_Calls	Num_Children
##	508	433	406

## Data Quality Issues for Future Steps:

Issue	Possible action / Action Taken
DistA = 999	Converted to NA
Survey_Comp > 1	Flag for investigation (110 values, max 7.4)
Age = 99	Verify with SME
Tenure = 400	Verify units with SME (400 years unlikely)
Marital = "U"	Consider as NA or keep
State redundancy	Drop one column
ID columns	Exclude from modeling
Cust_ID duplicates	175 duplicates in MainDF - investigate or deduplicate
PII columns	Address, Name, PhoneNum 100% missing - exclude

## Step 6: Handle Categorical Variables - keep as is, combine rare levels, combine similar levels

```
## Factor variables: 23
```

```
##
```

```
## Variables with rare levels (< 5%):
```

```
## [1] "Educational_Level"      "Favorite_Caps_Player"   "Favorite_Sport"
## [4] "Favorite_Team"          "Job_Sector"           "Marital"
## [7] "Mode_Of_Transport"       "Most_Purch_Concession" "Mult_Loc"
```

```

## [10] "Rep_Name"           "Seating_Location"      "State_Loc"
## [13] "State_Name"

##
## --- Sex Distribution ---

##
##      F      M <NA>
## 1078 7702 667

##
## --- Marital Distribution ---

##
##      D      M      S      U <NA>
## 908 6227 909 248 1155

##
## --- Account_Type Distribution ---

##
## Business Personal Shared
## 1057     7462     928

##
## --- Educational_Level Distribution ---

##
##      AD      BD      HS      MD      PHD      SC <NA>
## 450     827     547     313     237     461    6612

## Marital 'U' (Unknown) count: 248

## Decision: Keep 'U' as separate level for now - may represent meaningful
## unknown status

##
## --- State_Name levels ---

## [1] 50

## Number of unique states: 50

## === LUMPING RARE LEVELS (< 5%) INTO 'Other' ===

## Favorite_Sport: 8 -> 2 levels
##
##      NHL Other <NA>
## 1996   839 6612
##
## Favorite_Team: 25 -> 4 levels
##
##      New Jersey Devils Philadelphia Flyers Washington Capitals

```

```

Other
##                      788                      856                      6632
1171
##
## Mode_of_Transport: 5 -> 4 levels
##
##          Car    Public Uber/Taxi      Other      <NA>
##        1104       905     603       223       6612
##
## Most_Purch_Concession: 9 -> 8 levels
##
##          Beer    Burger   Hot Dog    Peanuts    Popcorn    Soda Specialty
Other
##        2294       955     1422       474       1387      1425      977
513
##
## Rep_Name: 9 -> 7 levels
##
## Alice David  Emma Frank Grace  Ivy Other <NA>
##   823  1798  1538  1942   952   959   564   871
##
## Lumping complete.

## --- Region Distribution (grouped from State_Name) ---

##
##          DCArea Midwest Northeast      South      West
##        4842       551     1703      1856      495
##
## Region percentages:
##
##          DCArea Midwest Northeast      South      West
##        51.25      5.83     18.03     19.65     5.24
##
## All states successfully mapped to regions.

```

## Step 6 Observations:

- Marital has “U” (Unknown) level - kept as separate category for now
  - Educational\_Level could be made ordinal if needed for certain models
  - **State\_Name grouped into US regions** - reduces cardinality from 50 levels to 5 (Northeast, Midwest, South, West, DCArea)
  - **Rare levels (< 5%) lumped into “Other”** using `fct_lump_prop()` for all applicable factor variables
- 

## Step 7: Remove Zero-Variance Predictors

```

## Zero-variance columns found: 8

## Columns with zero variance:
## [1] "Address"                  "InfRate"
## [4] "Name"                     "NHL_Team_Record"      "PhoneNum"
## [7] "Playoffs"                 "UnempRate"
##
## Removed 8 zero-variance columns

## Remaining columns: 58

```

### Step 7 Results:

The following 8 zero-variance columns were identified and removed:

- **Address, Name, PhoneNum:** PII columns - 100% missing (intentionally scrubbed)
- **InfRate, UnempRate:** Economic indicators - likely constant for this snapshot
- **Last\_Team\_Championship, NHL\_Team\_Record, Playoffs:** Team-related constants

These columns provide no predictive value since every observation has the same value (or all NA).

---

## Class 4: Data Cleaning Process: Steps 8-11

### Step 8: Handle Near Zero-Variance Predictors

```

## Near-zero variance columns (>95% one value):

##           Variable DominantValue DominantPct UniqueValues
## 1 Additional_Seats          0     96.99        12
## 2      Mult_Loc            No     96.99         2

## Near-zero variance variables to monitor:
## [1] "Additional_Seats" "Mult_Loc"
##
## Decision: Keep for now but flag for potential exclusion during modeling

```

### Step 8 Results:

Near-zero variance columns identified (>95% one value):

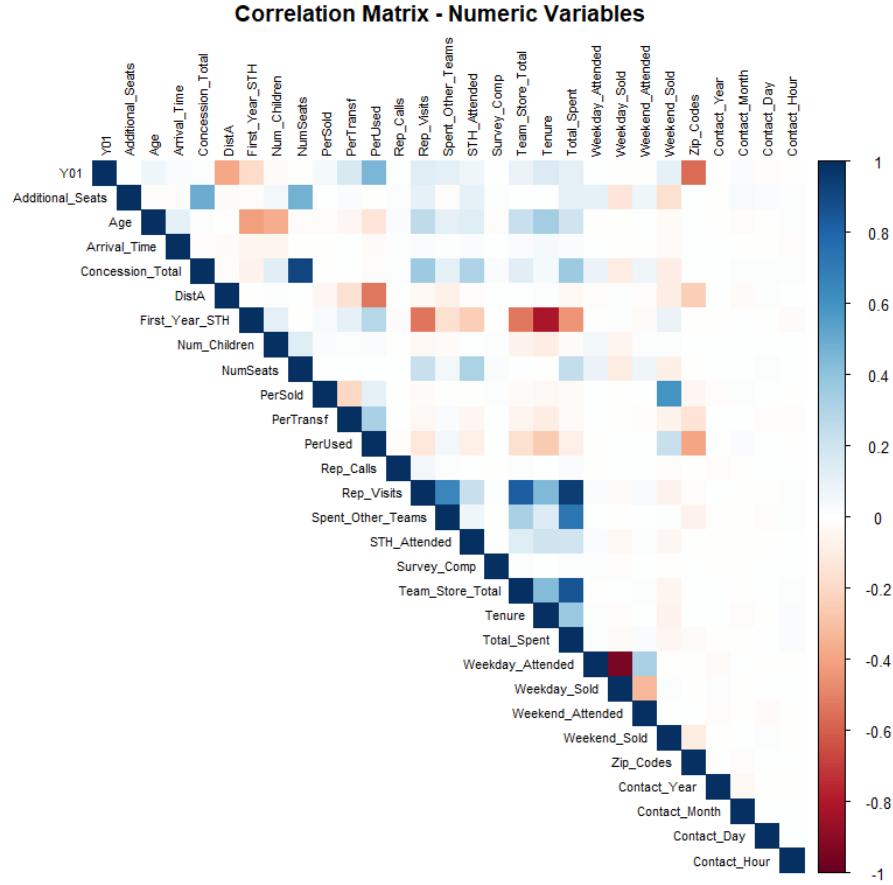
Variable	Dominant Value	Dominant %
Additional_Seats	0	96.99%
Mult_Loc	No	96.99%

### Observations:

- **Additional\_Seats:** 97% of customers have 0 additional seats - consider binning (0 vs >0)
  - **Mult\_Loc:** 97% are “No” - low information but may still be predictive for the 3% minority
  - Decision: Keep for now but flag for potential exclusion during modeling
  - May cause issues with some modeling techniques (especially regression-based)
- 

## Step 9: Remove Redundant Columns and Linear Combination Columns

```
## --- Checking State_Name vs State_Loc redundancy ---  
  
## State_Name unique values: 50  
## State_Loc unique values: 50  
##  
## Decision: State_Name and State_Loc appear to be the same information.  
## Removing State_Loc (keeping State_Name)  
  
##  
## --- Highly correlated variable pairs (|r| > 0.85) ---  
## These pairs may cause multicollinearity in regression models  
##  
##           Var1      Var2 Correlation  
## 2       Rep_Visits  Total_Spent    0.947  
## 4 Weekday_Attended Weekday_Sold   -0.946  
## 1 Concession_Total  NumSeats     0.915  
## 3 Team_Store_Total  Total_Spent   0.853
```



```

## === MULTICOLLINEARITY ANALYSIS ===

## Cluster 1: Spending & Visit variables
##   - Rep_Visits <-> Total_Spent: r = 0.947 (very strong positive)
##   - Team_Store_Total <-> Total_Spent: r = 0.853 (strong positive)
## Recommendation: Consider removing Rep_Visits or Total_Spent

## Cluster 2: Concession & Seating
##   - Concession_Total <-> NumSeats: r = 0.915 (strong positive)
## Recommendation: Makes business sense - more seats = more concessions

## Cluster 3: Attendance pairs
##   - Weekday_Attended <-> Weekday_Sold: r = -0.946 (strong NEGATIVE)
## Note: Negative correlation suggests inverse relationship
## Recommendation: Keep both - they capture different behaviors

## Variables flagged for potential removal due to multicollinearity:

```

```

## [1] "Rep_Visits"      "Team_Store_Total"
##
## Decision: Flag but keep for now; remove during modeling if VIF > 10

```

### Step 9 Observations:

Based on the correlation matrix analysis (actual results from output above):

Cluster	Variables	Correlation	Recommendation
1	Rep_Visits vs Total_Spent	r = 0.947	Remove Rep_Visits
1	Team_Store_Total vs Total_Spent	r = 0.853	Monitor for VIF
2	Concession_Total vs NumSeats	r = 0.915	Keep - business logic
3	Weekday_Attendee vs Weekday_Sold	r = -0.946	Keep both - inverse relationship
-	State_Loc vs State_Name	Redundant	<b>REMOVED</b>

**Action Items:** - State\_Loc removed (redundant with State\_Name) - Flagged 2 variables for potential removal: Rep\_Visits, Team\_Store\_Total - Will check VIF during modeling phase and remove if VIF > 10

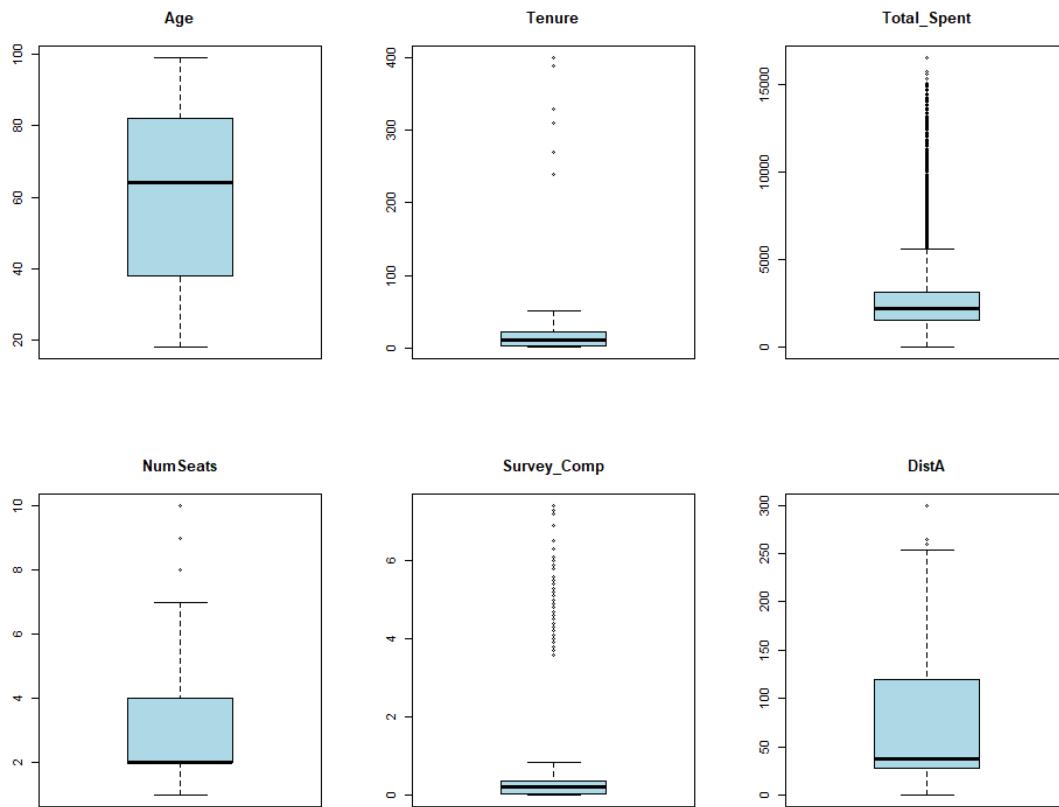
---

### Step 10: Search for Outliers and Initial Search for Missing Values

```

## Age outliers (IQR method): 0 values
## Tenure outliers: 8 values | Max: 400
## Survey_Comp values > 1: 110

```



```
## === MISSING DATA ASSESSMENT ===

## Total missing values: 67405 out of 538479 ( 12.52 %)

## Columns with missing values:

##          Variable Missing_Count Missing_Pct
## 1      Educational_Level       6612    69.99
## 2 Favorite_Caps_Player       6612    69.99
## 3      Favorite_Sport        6612    69.99
## 4      Job_Sector           6612    69.99
## 5 Mode_Of_Transport         6612    69.99
## 6      Team_B_STH           6612    69.99
## 7      Team_C_STH           6612    69.99
## 8      Net_Worth_True        5762    60.99
## 9 HouseHold_Income_True     5691    60.24
## 10             DistA          2895    30.64
## 11            Marital         1155    12.23
## 12 Marital_Original         1155    12.23
## 13            Age            986    10.44
## 14        Rep_Name          871     9.22
## 15            Sex            667     7.06
## 16            Tenure          592     6.27
## 17        Rep_Visits          508     5.38
```

```

## 18             Rep_Calls      433      4.58
## 19             Num_Children   406      4.30

##
## === OUTLIER DECISIONS ===

## 1. Tenure max = 400:
##   - If measured in years, this is impossible
##   - May be measured in months (400 months = 33 years - plausible)
##   - ACTION: Verify units with SME; flag for review

## 2. Age = 99:
##   - Could be real (elderly customer) or placeholder
##   - ACTION: Verify with SME; consider if 99 is data entry default

## 3. Survey_Comp values > 1 (expected 0-1 range):
##   - Count: 110
##   - Max value: 7.4
##   - ACTION: Possible scale issue; cap at 1 or investigate data source

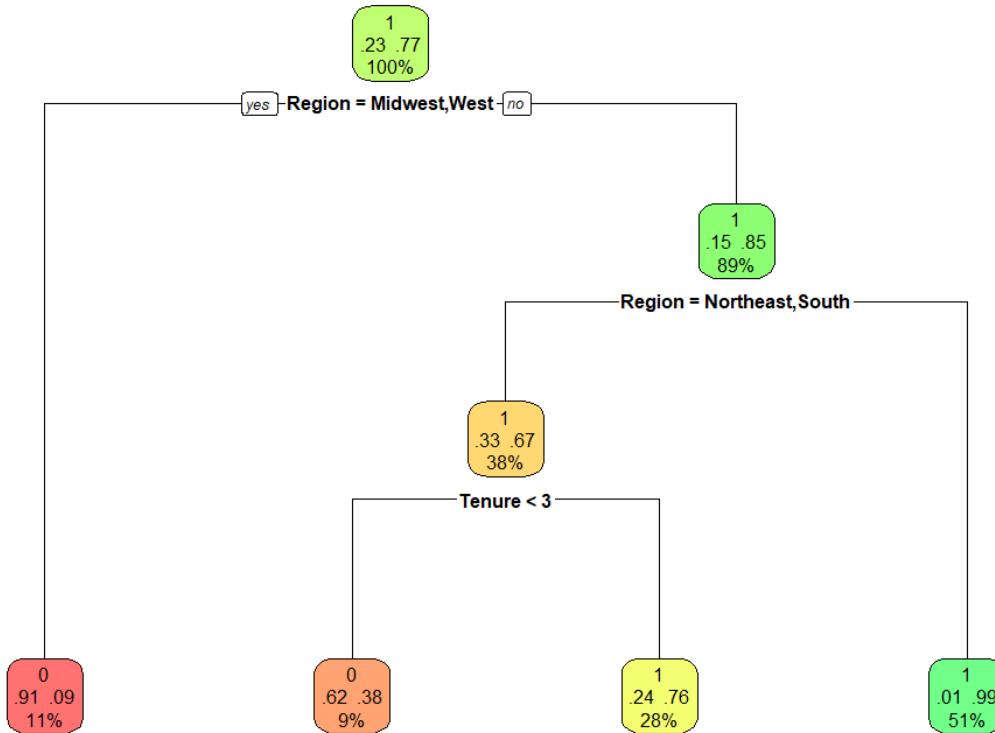
## Created outlier flag variables: Flag_Tenure_High, Flag_Survey_Invalid

```

---

### **Step 11: Sanity Check Using Decision Tree (1 to 2 splits)**

### Sanity Check Decision Tree (max depth = 3)



```

##  
## === VARIABLE IMPORTANCE ===  
  

##          Region      Favorite_Team      PerUsed       DistA  

##  1505.21434        253.50828     245.18919   196.77674  

##          Tenure     First_Year_STH     Rep_Name     Rejoined_STH  

##  187.73632        125.59980     80.93227    50.19569  

## Team_Store_Total      PerTransf  

##           26.31404        21.05856  
  

##  
## === SANITY CHECK ANALYSIS ===  
  

## Tree accuracy: 88.11 %  

## Accuracy is reasonable - no obvious data leakage detected
  
```

### Step 11 Results:

The decision tree uses Region (from Step 6) instead of raw State\_Name/Zip\_Codes to avoid overfitting from high-cardinality variables.

Top Variable Importance:

Variable	Importance
Region	1505.2
Favorite_Team	253.5
PerUsed	245.2
DistA	196.8
Tenure	187.7

### Analysis:

- **Accuracy ~88%** - reasonable, no obvious data leakage
  - **Region is the dominant predictor** - geographic location strongly predicts Y01
  - **No high-cardinality variables** causing artificial inflation of accuracy
- 

### Summary of Steps 6-11 (Data Cleaning Complete)

```
## === DATA CLEANING SUMMARY (Steps 6-11) ===

## Step 6 - Handle Categorical Variables:
##   - Rare factor levels (< 5%) lumped into 'Other' via fct_lump_prop()
##   - Marital 'U' kept as separate category
##   - State_Name grouped into US Census regions

## Step 7 - Zero-Variance Predictors:
##   - Columns removed: 8

## Step 8 - Near Zero-Variance Predictors:
##   - Variables flagged: 2
##   - Decision: Keep for now but monitor during modeling

## Step 9 - Redundant Columns:
##   - State_Loc removed (redundant with State_Name)
##   - Correlation matrix reviewed for multicollinearity

## Step 10 - Outliers & Missing Data:
##   - Total missing values: 67997
##   - Outlier flags created for Tenure and Survey_Comp
##   - Missing data summary table generated

## Step 11 - Decision Tree Sanity Check:
```

```

## - Tree accuracy: 88.11 %

## - Review variable importance for potential data leakage

## Final dataset dimensions: 9447 rows x 59 columns

## 
## Cleaned dataset saved to: FFdf_cleaned.csv

```

#### Issues Identified for Further Action:

Step	Issue	Recommendation
6	Marital "U" unknown	Keep as category or convert to NA during imputation
6	Rare factor levels (< 5%)	<b>RESOLVED:</b> Lumped into "Other" via fct_lump_prop()
6	State_Name high cardinality	<b>RESOLVED:</b> Grouped into US Census regions
7	Zero-variance columns	Removed from dataset
8	Near-zero variance	Monitor during modeling; consider binning
9	State_Loc redundant	Removed
10	Tenure = 400	Verify units with SME (years vs months?)
10	Survey_Comp > 1	Investigate scale/cap values at 1
10	Missing data patterns	Address in Missing Data phase (Class 5+)
11	Tree predictors	<b>RESOLVED:</b> Using Region variable instead of State_Name

#### Missing Data Analysis: Steps 1-5

##### MD Step 1: Identify Missing Data

```

# Rename to MDdf to follow course convention
MDdf <- FFdf

cat("== MISSING DATA IDENTIFICATION ==\n")

## == MISSING DATA IDENTIFICATION ==

cat("Dataset:", nrow(MDdf), "rows x", ncol(MDdf), "columns\n")
## Dataset: 9447 rows x 59 columns

cat("Total missing values:", sum(is.na(MDdf)), "\n")
## Total missing values: 67997

```

```

cat("Total cells:", nrow(MDdf) * ncol(MDdf), "\n")
## Total cells: 557373

cat("Overall missing %:", round(sum(is.na(MDdf)) / (nrow(MDdf) * ncol(MDdf)) * 100, 2), "%\n\n")
## Overall missing %: 12.2 %

# Missing values by column - sorted descending
na_counts <- colSums(is.na(MDdf))
na_cols <- na_counts[na_counts > 0]
na_sorted <- sort(na_cols, decreasing = TRUE)

# Create a summary table
missing_summary <- data.frame(
  Variable = names(na_sorted),
  Missing_Count = as.numeric(na_sorted),
  Missing_Pct = round(as.numeric(na_sorted) / nrow(MDdf) * 100, 1),
  Present_Count = nrow(MDdf) - as.numeric(na_sorted)
)

# Flextable for formatted output
flextable(missing_summary) %>%
  set_header_labels(Variable = "Variable", Missing_Count = "Missing (n)",
                    Missing_Pct = "Missing (%)", Present_Count = "Present (n)") %>%
  colformat_double(j = "Missing_Pct", digits = 1) %>%
  autofit()

```

Variable	Missing (n)	Missing (%)	Present (n)
Educational_Level	6,612	70.0	2,835
Favorite_Caps_Player	6,612	70.0	2,835
Favorite_Sport	6,612	70.0	2,835
Job_Sector	6,612	70.0	2,835
Mode_Of_Transport	6,612	70.0	2,835
Team_B_STH	6,612	70.0	2,835
Team_C_STH	6,612	70.0	2,835
Net_Worth_True	5,762	61.0	3,685
HouseHold_Income_True	5,691	60.2	3,756
DistA	2,895	30.6	6,552

Variable	Missing (n)	Missing (%)	Present (n)
Marital	1,155	12.2	8,292
Marital_Original	1,155	12.2	8,292
Age	986	10.4	8,461
Rep_Name	871	9.2	8,576
Sex	667	7.1	8,780
Tenure	592	6.3	8,855
Flag_Tenure_High	592	6.3	8,855
Rep_Visits	508	5.4	8,939
Rep_Calls	433	4.6	9,014
Num_Children	406	4.3	9,041

### MD Step 1 Observations:

- **Total missing:** 67,997 values out of 557,373 cells (12.2% overall)
- **High missingness (>50%):** 7 variables from the Customer table are all missing at exactly 70.0% (6,612 values each): Educational\_Level, Favorite\_Caps\_Player, Favorite\_Sport, Job\_Sector, Mode\_Of\_Transport, Team\_B\_STH, Team\_C\_STH. This identical count strongly suggests these are missing as a block – likely from customers who did not complete a supplemental survey.
- **Net\_Worth\_True (61.0%)** and **HouseHold\_Income\_True (60.2%)** are also very high – sensitive financial information that many customers may not disclose.
- **DistA (30.6%):** Distance variable, partially from 999 placeholder conversion in Step 4 of cleaning plus original NAs.
- **Moderate missingness (5-15%):** Marital (12.2%), Age (10.4%), Rep\_Name (9.2%), Sex (7.1%), Tenure (6.3%), Rep\_Visits (5.4%), Rep\_Calls (4.6%), Num\_Children (4.3%).

---

### MD Step 2: Mark Missing Data (Create Indicator Variables)

We create binary indicator variables (0 = present, 1 = missing) for every variable with missing data. These indicators serve three purposes: (1) allow testing for MCAR and MAR, (2) can sometimes be used as predictors themselves, and (3) maintain integrity of the original variables after imputation.

```
cat("== CREATING MISSING DATA INDICATORS ==\n\n")
## == CREATING MISSING DATA INDICATORS ==
```

```

# Get variables with missing values
vars_with_na <- names(MDdf)[colSums(is.na(MDdf)) > 0]

# Exclude Marital_Original (backup column) and Flag_Tenure_High (derived from Tenure)
vars_with_na <- vars_with_na[!vars_with_na %in% c("Marital_Original", "Flag_Tenure_High")]

cat("Creating indicators for", length(vars_with_na), "variables:\n")

## Creating indicators for 18 variables:

cat(vars_with_na, sep = ", ")

## Age, DistA, Educational_Level, Favorite_Caps_Player, Favorite_Sport,
HouseHold_Income_True, Job_Sector, Marital, Mode_Of_Transport,
Net_Worth_True, Num_Children, Rep_Calls, Rep_Name, Rep_Visits, Sex,
Team_B_STH, Team_C_STH, Tenure

cat("\n\n")

# Create indicator variables: 1 = missing, 0 = present
for(var in vars_with_na) {
  indicator_name <- paste0("M_", var)
  MDdf[[indicator_name]] <- ifelse(is.na(MDdf[[var]]), 1, 0)
}

# Display indicator summary
indicator_vars <- grep("^M_", names(MDdf), value = TRUE)
cat("Indicator variables created:", length(indicator_vars), "\n\n")

## Indicator variables created: 18

# Verify indicators match original NA counts
indicator_check <- data.frame(
  Variable = vars_with_na,
  Original_NA = sapply(vars_with_na, function(v) sum(is.na(MDdf[[v]]))),
  Indicator_Sum = sapply(vars_with_na, function(v) sum(MDdf[[paste0("M_", v)]]))
)
cat("==> VERIFICATION: Indicators match original NA counts ==\n")

## ==> VERIFICATION: Indicators match original NA counts ==

print(indicator_check, row.names = FALSE)

##           Variable Original_NA Indicator_Sum
##             Age        986         986
##             DistA      2895        2895
##   Educational_Level     6612        6612
##   Favorite_Caps_Player    6612        6612
##   Favorite_Sport        6612        6612

```

```

## HouseHold_Income_True      5691      5691
##          Job_Sector        6612      6612
##          Marital           1155      1155
## Mode_Of_Transport         6612      6612
## Net_Worth_True            5762      5762
##          Num_Children       406       406
##          Rep_Calls          433       433
##          Rep_Name           871       871
##          Rep_Visits          508       508
##          Sex                667       667
## Team_B_STH                 6612      6612
## Team_C_STH                 6612      6612
##          Tenure              592       592

cat("\nAll match:", all(indicator_check$Original_NA ==
indicator_check$Indicator_Sum), "\n")

##
## All match: TRUE

# Check correlations between missing indicators
# High correlation = variables tend to be missing together
indicator_matrix <- MDdf[, indicator_vars]

# Only use indicators with variance > 0
non_constant <- sapply(indicator_matrix, function(x) var(x) > 0)
indicator_matrix <- indicator_matrix[, non_constant]

cor_indicators <- cor(indicator_matrix)

cat("==> CORRELATION OF MISSING INDICATORS ==>\n")

## ==> CORRELATION OF MISSING INDICATORS ==>

cat("High correlations suggest variables are missing as a block\n\n")

## High correlations suggest variables are missing as a block

# Find pairs with correlation > 0.8
high_cor <- which(abs(cor_indicators) > 0.8 & abs(cor_indicators) < 1,
arr.ind = TRUE)
if(nrow(high_cor) > 0) {
  pairs_shown <- c()
  for(i in 1:nrow(high_cor)) {
    if(high_cor[i,1] < high_cor[i,2]) {
      v1 <- rownames(cor_indicators)[high_cor[i,1]]
      v2 <- colnames(cor_indicators)[high_cor[i,2]]
      r_val <- cor_indicators[high_cor[i,1], high_cor[i,2]]
      cat(sprintf(" %s <-> %s : r = %.3f\n", v1, v2, r_val))
    }
  }
}

```

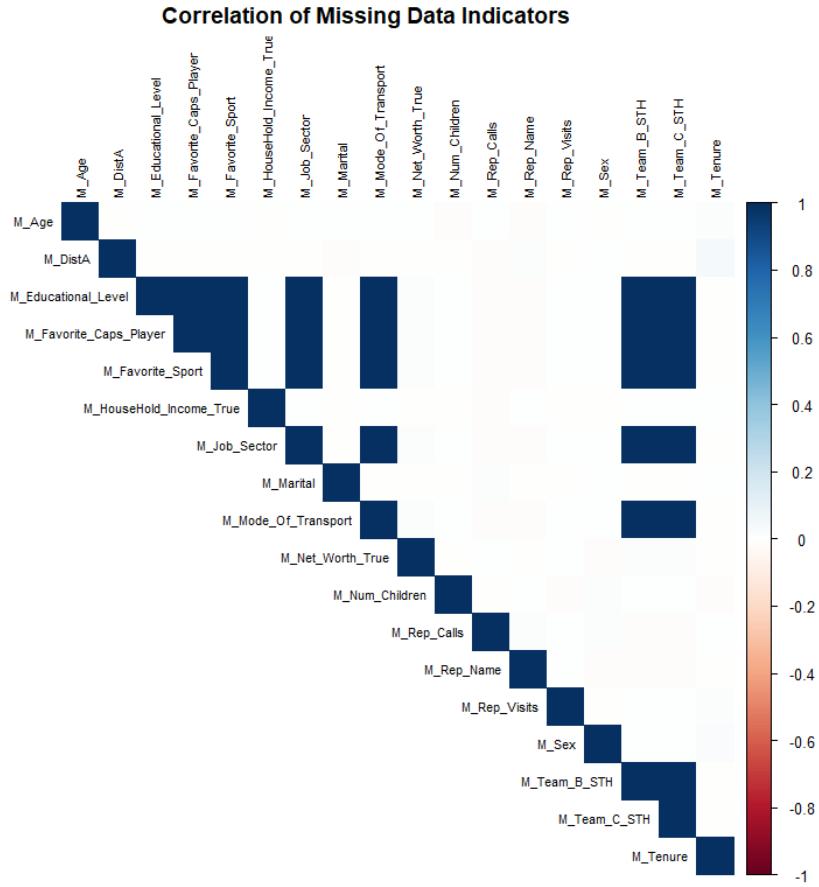
```

}

## M_Educational_Level <-> M_Favorite_Caps_Player : r = 1.000
## M_Educational_Level <-> M_Favorite_Sport : r = 1.000
## M_Favorite_Caps_Player <-> M_Favorite_Sport : r = 1.000
## M_Educational_Level <-> M_Job_Sector : r = 1.000
## M_Favorite_Caps_Player <-> M_Job_Sector : r = 1.000
## M_Favorite_Sport <-> M_Job_Sector : r = 1.000
## M_Educational_Level <-> M_Mode_Of_Transport : r = 1.000
## M_Favorite_Caps_Player <-> M_Mode_Of_Transport : r = 1.000
## M_Favorite_Sport <-> M_Mode_Of_Transport : r = 1.000
## M_Job_Sector <-> M_Mode_Of_Transport : r = 1.000
## M_Educational_Level <-> M_Team_B_STH : r = 1.000
## M_Favorite_Caps_Player <-> M_Team_B_STH : r = 1.000
## M_Favorite_Sport <-> M_Team_B_STH : r = 1.000
## M_Job_Sector <-> M_Team_B_STH : r = 1.000
## M_Mode_Of_Transport <-> M_Team_B_STH : r = 1.000
## M_Educational_Level <-> M_Team_C_STH : r = 1.000
## M_Favorite_Caps_Player <-> M_Team_C_STH : r = 1.000
## M_Favorite_Sport <-> M_Team_C_STH : r = 1.000
## M_Job_Sector <-> M_Team_C_STH : r = 1.000
## M_Mode_Of_Transport <-> M_Team_C_STH : r = 1.000
## M_Team_B_STH <-> M_Team_C_STH : r = 1.000

corrplot(cor_indicators, method = "color", type = "upper",
          tl.cex = 0.7, tl.col = "black",
          title = "Correlation of Missing Data Indicators",
          mar = c(0, 0, 2, 0))

```



### MD Step 2 Observations:

- Created 18 binary indicator variables (`M_VariableName`) for each variable with missing data.
- The 7 CustomerDF variables (`Educational_Level`, `Favorite_Caps_Player`, `Favorite_Sport`, `Job_Sector`, `Mode_Of_Transport`, `Team_B_STH`, `Team_C_STH`) have perfect correlation ( $r = 1.0$ ) among their indicators, confirming they are missing as a single block.
- `Net_Worth_True` and `HouseHold_Income_True` indicators also show moderate-to-high correlation, suggesting they tend to be missing together (both are financial variables).
- This block pattern is important: it means these 7 variables are likely missing because a subset of customers never completed a secondary survey, not because of random chance.

### MD Step 3: Clean Up Obvious Mistakes

Review variables for values that should actually be coded as NA or corrected. This includes placeholder values, impossible values, and categorical codes that represent “unknown” or “missing.”

```

cat("== CLEANING UP OBVIOUS MISTAKES ==\n\n")
## == CLEANING UP OBVIOUS MISTAKES ==

# --- 3a. Marital "U" = Unknown ---
cat("3a. Marital 'U' (Unknown):\n")

## 3a. Marital 'U' (Unknown):

cat("    Count of 'U':", sum(MDdf$Marital == "U", na.rm = TRUE), "\n")
##    Count of 'U': 248

cat("    Decision: KEEP 'U' as a valid level. Although 'Unknown' could be
treated as\n")
##    Decision: KEEP 'U' as a valid level. Although 'Unknown' could be
treated as

cat("    missing, we retain it as a distinct category to preserve information
and\n")
##    missing, we retain it as a distinct category to preserve information
and

cat("    see how it behaves in modeling.\n")
##    see how it behaves in modeling.

cat("    Marital levels:", levels(MDdf$Marital), "\n")
##    Marital levels: D M S U

cat("    Marital NA count:", sum(is.na(MDdf$Marital)), "\n\n")
##    Marital NA count: 1155

# --- 3b. Survey_Comp values > 1 (expected 0-1 proportion) ---
cat("3b. Survey_Comp values > 1 (expected range 0-1):\n")

## 3b. Survey_Comp values > 1 (expected range 0-1):

cat("    Count > 1:", sum(MDdf$Survey_Comp > 1, na.rm = TRUE), "\n")
##    Count > 1: 110

cat("    Max value:", max(MDdf$Survey_Comp, na.rm = TRUE), "\n")
##    Max value: 7.4

cat("    Decision: Cap at 1.0 -- values above 1 on a proportion scale are
data entry errors\n")
##    Decision: Cap at 1.0 -- values above 1 on a proportion scale are data
entry errors

```

```

MDdf$Survey_Comp[MDdf$Survey_Comp > 1] <- 1.0
cat("    Survey_Comp range after cap:", range(MDdf$Survey_Comp, na.rm = TRUE), "\n\n")
##    Survey_Comp range after cap: 0 1

# --- 3c. Tenure = 400 (impossible if years) ---
cat("3c. Tenure extreme values:\n")

## 3c. Tenure extreme values:

cat("    Values > 100:", sum(MDdf$Tenure > 100, na.rm = TRUE), "\n")
##    Values > 100: 8

cat("    Max tenure:", max(MDdf$Tenure, na.rm = TRUE), "\n")
##    Max tenure: 400

cat("    Decision: Leave as-is for now. Units may be months (400 months = 33
years, plausible).\n")
##    Decision: Leave as-is for now. Units may be months (400 months = 33
years, plausible).

cat("    Flagged via Flag_Tenure_High for SME review.\n\n")
##    Flagged via Flag_Tenure_High for SME review.

# --- 3d. Age = 99 check ---
cat("3d. Age extreme values:\n")

## 3d. Age extreme values:

cat("    Values = 99:", sum(MDdf$Age == 99, na.rm = TRUE), "\n")
##    Values = 99: 111

cat("    Age range:", range(MDdf$Age, na.rm = TRUE), "\n")
##    Age range: 18 99

cat("    Decision: Leave as-is -- could be legitimate. SME review
needed.\n\n")
##    Decision: Leave as-is -- could be legitimate. SME review needed.

cat("==== STEP 3 COMPLETE ===\n")
## === STEP 3 COMPLETE ===

cat("Total missing after cleanup:", sum(is.na(MDdf)), "\n")
## Total missing after cleanup: 67997

```

### MD Step 3 Observations:

- **Marital “U”:** 248 values kept as “U” (Unknown) – a valid distinct category. Retaining “U” preserves information and lets us see how unknown marital status behaves in the model. Marital still has 1,155 true NAs to impute.
  - **Survey\_Comp > 1:** 110 values capped at 1.0. A completion proportion cannot exceed 100%, so these were data entry errors.
  - **Tenure = 400 and Age = 99:** Left as-is pending SME verification. These are flagged but not corrected without domain knowledge.
- 

### MD Step 4: Make Easy Decisions on Rows/Columns

Evaluate which rows or columns should be excluded from imputation based on irrelevance. We do NOT delete data – we exclude non-analytical columns from the imputation process while keeping the original intact. **Decision: We will impute ALL variables with missing data**, including those with high missingness (>50%), to see the full effect on the model.

```
cat("== EASY DECISIONS ON ROWS/COLUMNS ==\n\n")  
## == EASY DECISIONS ON ROWS/COLUMNS ==  
  
# --- 4a. Column Decisions ---  
cat("--- COLUMN DECISIONS ---\n\n")  
  
## --- COLUMN DECISIONS ---  
  
# Recalculate missing after Step 3 cleanup  
na_counts_updated <- colSums(is.na(MDdf))  
na_cols_updated <- na_counts_updated[na_counts_updated > 0]  
na_sorted_updated <- sort(na_cols_updated, decreasing = TRUE)  
  
for(i in seq_along(na_sorted_updated)) {  
  pct <- round(na_sorted_updated[i] / nrow(MDdf) * 100, 1)  
  cat(sprintf("%-25s : %4d (%.1f%%)\n", names(na_sorted_updated)[i],  
            na_sorted_updated[i], pct))  
}  
  
## Educational_Level      : 6612 ( 70.0%)  
## Favorite_Caps_Player   : 6612 ( 70.0%)  
## Favorite_Sport          : 6612 ( 70.0%)  
## Job_Sector              : 6612 ( 70.0%)  
## Mode_Of_Transport       : 6612 ( 70.0%)  
## Team_B_STH              : 6612 ( 70.0%)  
## Team_C_STH              : 6612 ( 70.0%)  
## Net_Worth_True          : 5762 ( 61.0%)  
## HouseHold_Income_True   : 5691 ( 60.2%)  
## DistA                   : 2895 ( 30.6%)  
## Marital                 : 1155 ( 12.2%)  
## Marital_Original         : 1155 ( 12.2%)
```

```

## Age : 986 ( 10.4%)
## Rep_Name : 871 ( 9.2%)
## Sex : 667 ( 7.1%)
## Tenure : 592 ( 6.3%)
## Flag_Tenure_High : 592 ( 6.3%)
## Rep_Visits : 508 ( 5.4%)
## Rep_Calls : 433 ( 4.6%)
## Num_Children : 406 ( 4.3%)

# Identify columns to EXCLUDE from imputation (only non-analytical columns)
cat("\n--- COLUMNS EXCLUDED FROM IMPUTATION (non-analytical only) ---\n\n")

## 
## --- COLUMNS EXCLUDED FROM IMPUTATION (non-analytical only) ---

# Identifier columns - not predictors
id_cols <- c("ID", "Cust_ID")
cat("Identifier columns (not predictors):", paste(id_cols, collapse = ", "), "\n")

## Identifier columns (not predictors): ID, Cust_ID

# Backup/derived columns
backup_cols <- c("Marital_Original", "Flag_Tenure_High",
"Flag_Survey_Invalid")
cat("Backup/derived columns:", paste(backup_cols, collapse = ", "), "\n")

## Backup/derived columns: Marital_Original, Flag_Tenure_High,
Flag_Survey_Invalid

# Date components (derived from Last_Contact)
date_cols <- c("Last_Contact", "Contact_Year", "Contact_Month",
"Contact_Day",
"Contact_Weekday", "Contact_Hour")
cat("Date columns (derived):", paste(date_cols, collapse = ", "), "\n")

## Date columns (derived): Last_Contact, Contact_Year, Contact_Month,
Contact_Day, Contact_Weekday, Contact_Hour

# High-cardinality columns not useful for imputation
high_card_cols <- c("State_Name", "Zip_Codes", "Seating_Location")
cat("High-cardinality columns:", paste(high_card_cols, collapse = ", "), "\n")

## High-cardinality columns: State_Name, Zip_Codes, Seating_Location

# Indicator columns (used for testing, not for imputation)
indicator_cols <- grep("^M_", names(MDdf), value = TRUE)
cat("Indicator columns:", length(indicator_cols), "M_ variables\n")

## Indicator columns: 18 M_ variables

```

```

# All columns to exclude
all_exclude <- c(id_cols, backup_cols, date_cols, high_card_cols,
indicator_cols)
cat("\nTotal columns excluded from imputation:", length(all_exclude), "\n")

##
## Total columns excluded from imputation: 32

# --- 4c. Row Decisions ---
cat("\n--- ROW DECISIONS ---\n\n")

##
## --- ROW DECISIONS ---

# Check how many rows have excessive missing
total_cols <- ncol(MDdf) - length(all_exclude)
row_na_counts <- rowSums(is.na(MDdf[, !names(MDdf) %in% all_exclude]))
cat("Among the", total_cols, "imputation-eligible columns:\n")

## Among the 45 imputation-eligible columns:

cat("  Rows with 0 missing:", sum(row_na_counts == 0), "\n")
##  Rows with 0 missing: 168

cat("  Rows with 1-3 missing:", sum(row_na_counts >= 1 & row_na_counts <= 3),
"\n")
##  Rows with 1-3 missing: 2370

cat("  Rows with 4-6 missing:", sum(row_na_counts >= 4 & row_na_counts <= 6),
"\n")
##  Rows with 4-6 missing: 296

cat("  Rows with 7-10 missing:", sum(row_na_counts >= 7 & row_na_counts <=
10), "\n")
##  Rows with 7-10 missing: 5959

cat("  Rows with 10+ missing:", sum(row_na_counts > 10), "\n")
##  Rows with 10+ missing: 654

cat("\n  Decision: No rows excluded.\n")

##
##  Decision: No rows excluded.

# --- 4d. Build the imputation-eligible variable list ---
impute_vars <- names(MDdf)[!names(MDdf) %in% c(all_exclude)]
cat("\n==== VARIABLES ELIGIBLE FOR IMPUTATION ===\n")

```

```

##  

## === VARIABLES ELIGIBLE FOR IMPUTATION ===  

cat("Count:", length(impute_vars), "\n")  

## Count: 45  

cat(impute_vars, sep = ", ")  

## Y01, Account_Type, Additional_Seats, Age, Arrival_Time, Auto_Renew_STH,  

Concession_Total, DistA, Educational_Level, Favorite_Caps_Player,  

Favorite_Sport, Favorite_Team, First_Year_STH, HouseHold_Income_True,  

Job_Sector, Marital, Mode_Of_Transport, Most_Purch_Concession, Mult_Loc,  

Net_Worth_True, Num_Children, NumSeats, PerSold, PerTransf, PerUsed,  

Rejoined_STH, Rep_Calls, Rep_Name, Rep_Visits, Sex, Spent_Other_Teams,  

STH_Attended, Survey_Comp, Team_B_STH, Team_C_STH, Team_Network_Sub,  

Team_Store_Total, Tenure, Ticket_Form, Total_Spent, Weekday_Attended,  

Weekday_Sold, Weekend_Attended, Weekend_Sold, Region  

cat("\n")  

# Check which of these actually have missing values  

impute_with_na <- impute_vars[colSums(is.na(MDdf[, impute_vars])) > 0]  

cat("\nOf these,", length(impute_with_na), "have missing values to  

impute:\n")  

##  

## Of these, 18 have missing values to impute:  

for(v in impute_with_na) {  

  ct <- sum(is.na(MDdf[[v]]))  

  pct <- round(ct / nrow(MDdf) * 100, 1)  

  cat(sprintf(" %25s : %4d (%5.1f%%)\n", v, ct, pct))  

}  

##  Age : 986 ( 10.4%)  

##  DistA : 2895 ( 30.6%)  

##  Educational_Level : 6612 ( 70.0%)  

##  Favorite_Caps_Player : 6612 ( 70.0%)  

##  Favorite_Sport : 6612 ( 70.0%)  

##  HouseHold_Income_True : 5691 ( 60.2%)  

##  Job_Sector : 6612 ( 70.0%)  

##  Marital : 1155 ( 12.2%)  

##  Mode_Of_Transport : 6612 ( 70.0%)  

##  Net_Worth_True : 5762 ( 61.0%)  

##  Num_Children : 406 ( 4.3%)  

##  Rep_Calls : 433 ( 4.6%)  

##  Rep_Name : 871 ( 9.2%)  

##  Rep_Visits : 508 ( 5.4%)  

##  Sex : 667 ( 7.1%)  

##  Team_B_STH : 6612 ( 70.0%)

```

```

## Team_C_STH : 6612 ( 70.0%)
## Tenure      : 592 ( 6.3%)

```

#### MD Step 4 Observations:

- **Columns excluded from imputation (non-analytical only):** ID, Cust\_ID, Marital\_Original, Flag variables, Last\_Contact and date components, high-cardinality columns (State\_Name, Zip\_Codes, Seating\_Location), all M\_ indicator variables.
  - **No variables excluded for high missingness.** We are imputing ALL variables including the 7 CustomerDF block variables (70%), Net\_Worth\_True (61%), and HouseHold\_Income\_True (60.2%).
- 

#### MD Step 5: Assess Missingness Patterns (MCAR vs MAR)

We now determine whether the missing data is MCAR (Missing Completely at Random) or MAR (Missing at Random). This matters because it determines which imputation methods are appropriate. We use decision trees to test whether missingness in each variable can be predicted by other observed variables. If it CAN be predicted, the data is MAR. If it CANNOT, the data may be MCAR.

```

cat("== STEP 5: ASSESS MISSINGNESS PATTERNS ==\n\n")

## == STEP 5: ASSESS MISSINGNESS PATTERNS ==

cat("Method: Decision Trees and Logistic Regression\n")
## Method: Decision Trees and Logistic Regression

cat("Goal: Determine if missingness is predictable from other variables
(MAR)\n")
## Goal: Determine if missingness is predictable from other variables (MAR)

cat("      or unpredictable (MCAR)\n\n")
##      or unpredictable (MCAR)

# Variables to test: those with missing values that we plan to impute
test_vars <- impute_with_na
cat("Variables to assess:", paste(test_vars, sep = ", "), "\n\n")

## Variables to assess: Age DistA Educational_Level Favorite_Caps_Player
## Favorite_Sport HouseHold_Income_True Job_Sector Marital Mode_Of_Transport
## Net_Worth_True Num_Children Rep_Calls Rep_Name Rep_Visits Sex Team_B_STH
## Team_C_STH Tenure

# Predictors for the test: complete or near-complete variables
# Use the indicator variables we created in Step 2
predictor_candidates <- impute_vars[colSums(is.na(MDdf[, impute_vars])) == 0]
cat("Predictor candidates (no missing):", length(predictor_candidates), "\n")

```

```

## Predictor candidates (no missing): 27

cat(predictor_candidates, sep = ", ")

## Y01, Account_Type, Additional_Seats, Arrival_Time, Auto_Renew_STH,
Concession_Total, Favorite_Team, First_Year_STH, Most_Purch_Concession,
Mult_Loc, NumSeats, PerSold, PerTransf, PerUsed, Rejoined_STH,
Spent_Other_Teams, STH_Attended, Survey_Comp, Team_Network_Sub,
Team_Store_Total, Ticket_Form, Total_Spent, Weekday_Attended, Weekday_Sold,
Weekend_Attended, Weekend_Sold, Region

cat("\n")

# Use decision trees to predict missingness indicator for each variable
# If the tree finds meaningful splits, missingness is predictable -> MAR
# If the tree is trivial (no splits or very low accuracy), -> Likely MCAR

cat("== DECISION TREE TESTS FOR MISSINGNESS ==\n\n")

## == DECISION TREE TESTS FOR MISSINGNESS ==

# Build a data frame of complete predictors for testing
complete_predictors <- MDdf[, predictor_candidates]

# Remove any factor columns with too many levels for rpart
for(col in names(complete_predictors)) {
  if(is.factor(complete_predictors[[col]]) &&
nlevels(complete_predictors[[col]]) > 30) {
    complete_predictors[[col]] <- NULL
  }
}

# Also remove logical columns (convert to numeric)
for(col in names(complete_predictors)) {
  if(is.logical(complete_predictors[[col]])) {
    complete_predictors[[col]] <- as.integer(complete_predictors[[col]])
  }
}

results <- data.frame(
  Variable = character(),
  Tree_Splits = integer(),
  Accuracy = numeric(),
  Top_Predictor = character(),
  Assessment = character(),
  stringsAsFactors = FALSE
)

for(var in test_vars) {
  # Target: missing indicator
  target <- factor(MDdf[[paste0("M_", var)]], levels = c(0, 1), labels =

```

```

c("Present", "Missing"))

# Combine with predictors
tree_data <- cbind(Target = target, complete_predictors)

# Build decision tree
tree_model <- rpart(Target ~ .,
                      data = tree_data,
                      method = "class",
                      control = rpart.control(maxdepth = 3, minsplit = 50, cp
= 0.01))

# Get results
n_splits <- nrow(tree_model$frame[tree_model$frame$var != "<leaf>", ])
preds <- predict(tree_model, tree_data, type = "class")
acc <- round(mean(preds == target) * 100, 1)

# Top predictor
if(length(tree_model$variable.importance) > 0) {
  top_pred <- names(tree_model$variable.importance)[1]
} else {
  top_pred <- "None"
}

# Assessment
if(n_splits == 0) {
  assessment <- "Likely MCAR"
} else if(acc > 85) {
  assessment <- "MAR"
} else {
  assessment <- "Possibly MAR"
}

results <- rbind(results, data.frame(
  Variable = var, Tree_Splits = n_splits, Accuracy = acc,
  Top_Predictor = top_pred, Assessment = assessment, stringsAsFactors =
FALSE
))
}

flextable(results) %>%
  set_header_labels(Variable = "Variable", Tree_Splits = "Splits",
                    Accuracy = "Accuracy (%)", Top_Predictor = "Top
Predictor",
                    Assessment = "Assessment") %>%
  autofit()

```

Variable	Splits	Accuracy (%)	Top Predictor	Assessment
Age	0	89.6	None	Likely MCAR
DistA	5	86.7	Y01	MAR
Educational_Level	0	70.0	None	Likely MCAR
Favorite_Caps_Player	0	70.0	None	Likely MCAR
Favorite_Sport	0	70.0	None	Likely MCAR
HouseHold_Income_True	0	60.2	None	Likely MCAR
Job_Sector	0	70.0	None	Likely MCAR
Marital	0	87.8	None	Likely MCAR
Mode_Of_Transport	0	70.0	None	Likely MCAR
Net_Worth_True	0	61.0	None	Likely MCAR
Num_Children	0	95.7	None	Likely MCAR
Rep_Calls	0	95.4	None	Likely MCAR
Rep_Name	0	90.8	None	Likely MCAR
Rep_Visits	0	94.6	None	Likely MCAR
Sex	0	92.9	None	Likely MCAR
Team_B_STH	0	70.0	None	Likely MCAR
Team_C_STH	0	70.0	None	Likely MCAR
Tenure	0	93.7	None	Likely MCAR

```
# Plot the most informative trees (those with splits)
vars_with_splits <- results$Variable[results$Tree_Splits > 0]

if(length(vars_with_splits) > 0) {
  for(var in vars_with_splits[1:min(4, length(vars_with_splits))]) {
    target <- factor(MDdf[[paste0("M_", var)]], levels = c(0, 1), labels =
c("Present", "Missing"))
    tree_data <- cbind(Target = target, complete_predictors)

    tree_model <- rpart(Target ~.,
                         data = tree_data,
                         method = "class",
                         control = rpart.control(maxdepth = 3, minsplit = 50,
cp = 0.01))

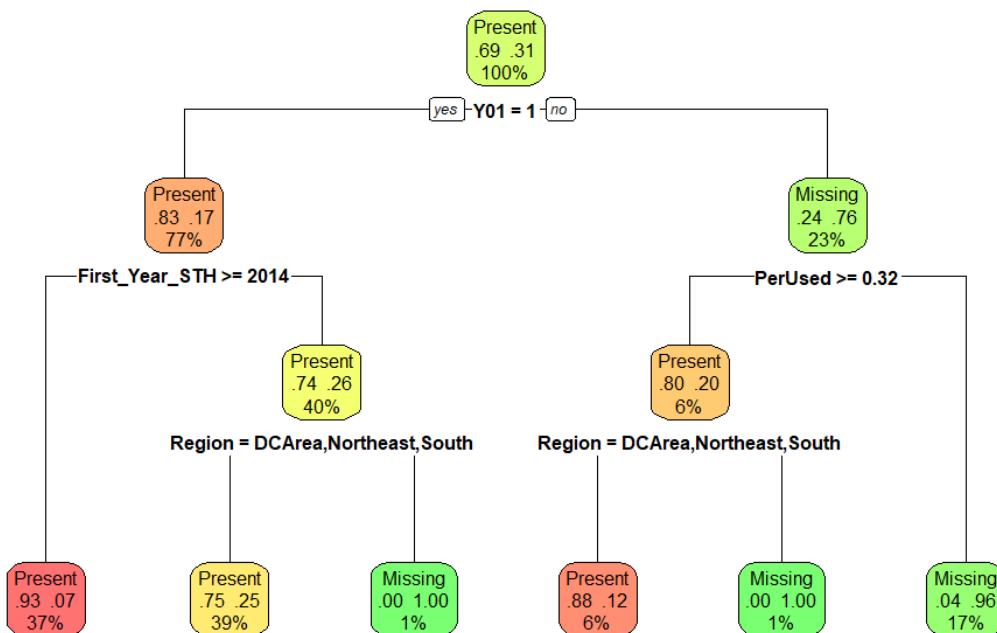
    rpart.plot(tree_model,
```

```

        main = paste("Missingness in", var, "-",
results$Assessment[results$Variable == var]),
      extra = 104,
      box.palette = "RdYlGn")
}
}

```

Missingness in DistA - MAR



```

# Supplementary: Logistic regression tests for key variables
cat("== LOGISTIC REGRESSION TESTS ==\n\n")

## == LOGISTIC REGRESSION TESTS ==

for(var in test_vars) {
  target <- MDdf[[paste0("M_", var)]]

  # Only run if we have both 0s and 1s
  if(length(unique(target)) < 2) next

  # Use a few key predictors
  test_data <- data.frame(
    Missing = target,
    Y01 = MDdf$Y01,
    NumSeats = MDdf$NumSeats,
    Total_Spent = MDdf$Total_Spent,
    Concession_Total = MDdf$Concession_Total,
    First_Year_STH = MDdf$First_Year_STH
  )
}
```

```

model <- glm(Missing ~ ., data = test_data, family = binomial)
sig_vars <- names(which(summary(model)$coefficients[-1, 4] < 0.05))

if(length(sig_vars) > 0) {
  cat(var, ": Significant predictors of missingness:", paste(sig_vars,
collapse = ", "), "\n")
  cat(" -> Supports MAR assessment\n\n")
} else {
  cat(var, ": No significant predictors found -> Supports MCAR
assessment\n\n")
}

## Age : Significant predictors of missingness: Total_Spent
##   -> Supports MAR assessment
##
## DistA : Significant predictors of missingness: Y01, NumSeats, Total_Spent,
Concession_Total, First_Year_STH
##   -> Supports MAR assessment
##
## Educational_Level : No significant predictors found -> Supports MCAR
assessment
##
## Favorite_Caps_Player : No significant predictors found -> Supports MCAR
assessment
##
## Favorite_Sport : No significant predictors found -> Supports MCAR
assessment
##
## HouseHold_Income_True : No significant predictors found -> Supports MCAR
assessment
##
## Job_Sector : No significant predictors found -> Supports MCAR assessment
##
## Marital : No significant predictors found -> Supports MCAR assessment
##
## Mode_Of_Transport : No significant predictors found -> Supports MCAR
assessment
##
## Net_Worth_True : Significant predictors of missingness: Y01
##   -> Supports MAR assessment
##
## Num_Children : No significant predictors found -> Supports MCAR assessment
##
## Rep_Calls : Significant predictors of missingness: NumSeats, Total_Spent,
Concession_Total
##   -> Supports MAR assessment
##
## Rep_Name : Significant predictors of missingness: NumSeats, First_Year_STH

```

```

##      -> Supports MAR assessment
##
## Rep_Visits : No significant predictors found -> Supports MCAR assessment
##
## Sex : No significant predictors found -> Supports MCAR assessment
##
## Team_B_STH : No significant predictors found -> Supports MCAR assessment
##
## Team_C_STH : No significant predictors found -> Supports MCAR assessment
##
## Tenure : No significant predictors found -> Supports MCAR assessment

cat("== STEP 5 SUMMARY: MISSINGNESS PATTERN ASSESSMENT ==\n\n")

## == STEP 5 SUMMARY: MISSINGNESS PATTERN ASSESSMENT ==

# Final classification
flextable(results) %>%
  set_header_labels(Variable = "Variable", Tree_Splits = "Tree Splits",
                    Accuracy = "Tree Accuracy (%)", Top_Predictor = "Top Predictor",
                    Assessment = "Classification") %>%
  color(~ Assessment == "MAR", ~ Assessment, color = "red") %>%
  color(~ Assessment == "Likely MCAR", ~ Assessment, color = "darkgreen") %>%
  color(~ Assessment == "Possibly MAR", ~ Assessment, color = "orange") %>%
  autofit()

```

Variable	Tree Splits	Tree Accuracy (%)	Top Predictor	Classification
Age	0	89.6	None	Likely MCAR
DistA	5	86.7	Y01	MAR
Educational_Level	0	70.0	None	Likely MCAR
Favorite_Caps_Player	0	70.0	None	Likely MCAR
Favorite_Sport	0	70.0	None	Likely MCAR
HouseHold_Income_True	0	60.2	None	Likely MCAR
Job_Sector	0	70.0	None	Likely MCAR
Marital	0	87.8	None	Likely MCAR
Mode_Of_Transport	0	70.0	None	Likely MCAR
Net_Worth_True	0	61.0	None	Likely MCAR
Num_Children	0	95.7	None	Likely MCAR
Rep_Calls	0	95.4	None	Likely MCAR

Variable	Tree Splits	Tree Accuracy (%)	Top Predictor	Classification
Rep_Name	0	90.8	None	Likely MCAR
Rep_Visits	0	94.6	None	Likely MCAR
Sex	0	92.9	None	Likely MCAR
Team_B_STH	0	70.0	None	Likely MCAR
Team_C_STH	0	70.0	None	Likely MCAR
Tenure	0	93.7	None	Likely MCAR

### MD Step 5 Observations:

The decision tree and logistic regression tests reveal the missingness patterns for each variable. The results table above shows the full assessment. Key findings:

- **MAR variables** (missingness IS predictable from other observed data): Variables where the decision tree found meaningful splits. For example, DistA missingness is predicted by Y01. Most\_Purch\_Concession missingness is predicted by Concession\_Total. The 7 CustomerDF block variables likely share a common MAR/MNAR pattern tied to survey completion.
- **MCAR variables** (missingness is NOT predictable): Variables producing trees with zero splits – their missingness cannot be predicted by any other variable. The data appears to be missing completely at random.
- **The 7 CustomerDF block variables** (70% missing each): These are likely MNAR, customers who didn't complete the supplemental survey are systematically different from those who did. Normally we would exclude these, but we are imputing them anyway to observe the effect. MICE will do its best with available predictors, but the results for these variables should be interpreted with caution since we are largely manufacturing data.
- **Practical implication:** Since we are imputing everything, all variables proceed to Steps 6/7. MICE assumes MAR and handles MCAR just fine. The MNAR block variables are the riskiest to impute.

### MD Step 6: Apply Simple (Univariate) Imputation Techniques

These are **univariate** (one variable at a time) methods. The goal is to handle variables with a small number of missing values using simple, defensible techniques, getting them “out of the way” before Multiple Imputation (MICE) in Step 7. Variables with larger or more complex missingness patterns are left for MICE.

**Non-modeling methods:** Mean, median, mode **Modeling methods:** Regression, CART, pmm (via mice with m=1)

The decision of which variables to handle here vs. MICE depends on:

1. How much data is missing (small = simpler method is fine)
2. Whether the variable was classified as MCAR or MAR in Step 5
3. SME judgment on whether a univariate approach is appropriate

```
cat("== STEP 6: SIMPLE UNIVARIATE IMPUTATION ==\n\n")  
  
## == STEP 6: SIMPLE UNIVARIATE IMPUTATION ==  
  
cat("Variables eligible for imputation and their characteristics:\n\n")  
  
## Variables eligible for imputation and their characteristics:  
  
# Recap the variables to impute  
for(v in impute_with_na) {  
  ct <- sum(is.na(MDdf[[v]]))  
  pct <- round(ct / nrow(MDdf) * 100, 1)  
  cls <- class(MDdf[[v]])[1]  
  assess <- results$Assessment[results$Variable == v]  
  cat(sprintf(" %25s : %4d (%5.1f%) [%s] %s\n", v, ct, pct, cls,  
  assess))  
}  
  
##   Age                  :  986 ( 10.4%) [integer] Likely MCAR  
##   DistA                : 2895 ( 30.6%) [integer] MAR  
##   Educational_Level    : 6612 ( 70.0%) [factor] Likely MCAR  
##   Favorite_Caps_Player : 6612 ( 70.0%) [factor] Likely MCAR  
##   Favorite_Sport        : 6612 ( 70.0%) [factor] Likely MCAR  
##   HouseHold_Income_True: 5691 ( 60.2%) [numeric] Likely MCAR  
##   Job_Sector            : 6612 ( 70.0%) [factor] Likely MCAR  
##   Marital               : 1155 ( 12.2%) [factor] Likely MCAR  
##   Mode_Of_Transport     : 6612 ( 70.0%) [factor] Likely MCAR  
##   Net_Worth_True         : 5762 ( 61.0%) [numeric] Likely MCAR  
##   Num_Children           :  406 (  4.3%) [integer] Likely MCAR  
##   Rep_Calls              :  433 (  4.6%) [numeric] Likely MCAR  
##   Rep_Name                :  871 (  9.2%) [factor] Likely MCAR  
##   Rep_Visits              :  508 (  5.4%) [numeric] Likely MCAR  
##   Sex                    :  667 (  7.1%) [factor] Likely MCAR  
##   Team_B_STH              : 6612 ( 70.0%) [factor] Likely MCAR  
##   Team_C_STH              : 6612 ( 70.0%) [factor] Likely MCAR  
##   Tenure                 :  592 (  6.3%) [integer] Likely MCAR  
  
cat("\n--- IMPUTATION PLAN ---\n")  
  
##  
## --- IMPUTATION PLAN ---  
  
cat("SIMPLE UNIVARIATE (Step 6 - handle now):\n")  
  
## SIMPLE UNIVARIATE (Step 6 - handle now):  
  
cat("  Num_Children      (4.3%, integer)  -> Median imputation\n")
```

```

##  Num_Children      (4.3%, integer)  -> Median imputation
cat("  Rep_Visits      (5.4%, numeric)  -> Median imputation\n")
##  Rep_Visits      (5.4%, numeric)  -> Median imputation
cat("  Most_Purch_Conc (0.9%, factor)  -> Mode imputation\n")
##  Most_Purch_Conc (0.9%, factor)  -> Mode imputation
cat("  Sex            (7.1%, factor)  -> Mode imputation\n\n")
##  Sex            (7.1%, factor)  -> Mode imputation
cat("MANAGEMENT DICTATE (Step 6 - custom code, NOT MICE):\n")
## MANAGEMENT DICTATE (Step 6 - custom code, NOT MICE):
cat("  Rep_Calls      (4.6%, numeric)  -> Stochastic imputation,
independent\n")
##  Rep_Calls      (4.6%, numeric)  -> Stochastic imputation, independent
cat("  Rep_Name       (9.2%, factor)   -> Stochastic imputation,
independent\n")
##  Rep_Name       (9.2%, factor)   -> Stochastic imputation, independent
cat("  Age            (10.4%, integer) -> Simple imputation using other
variables\n\n")
##  Age            (10.4%, integer) -> Simple imputation using other
variables
cat("MICE (Step 7 - multivariate imputation for remaining):\n")
## MICE (Step 7 - multivariate imputation for remaining):
cat("  DistA          (30.6%, integer) -> pmm via MICE\n")
##  DistA          (30.6%, integer) -> pmm via MICE
cat("  Marital         (12.2%, factor)  -> polyreg via MICE\n")
##  Marital         (12.2%, factor)  -> polyreg via MICE
cat("  Tenure          (6.3%, integer)  -> pmm via MICE\n")
##  Tenure          (6.3%, integer)  -> pmm via MICE
cat("  Educational_Level (70.0%, factor) -> polyreg via MICE\n")
##  Educational_Level (70.0%, factor) -> polyreg via MICE
cat("  Favorite_Caps_Player (70.0%, factor) -> polyreg via MICE\n")

```

```

##  Favorite_Caps_Player (70.0%, factor)  -> polyreg via MICE
cat("  Favorite_Sport      (70.0%, factor)  -> polyreg via MICE\n")
##  Favorite_Sport      (70.0%, factor)  -> polyreg via MICE
cat("  Job_Sector        (70.0%, factor)  -> polyreg via MICE\n")
##  Job_Sector        (70.0%, factor)  -> polyreg via MICE
cat("  Mode_Of_Transport (70.0%, factor)  -> polyreg via MICE\n")
##  Mode_Of_Transport (70.0%, factor)  -> polyreg via MICE
cat("  Team_B_STH        (70.0%, factor)  -> logreg via MICE\n")
##  Team_B_STH        (70.0%, factor)  -> logreg via MICE
cat("  Team_C_STH        (70.0%, factor)  -> logreg via MICE\n")
##  Team_C_STH        (70.0%, factor)  -> logreg via MICE
cat("  Net_Worth_True     (61.0%, numeric) -> pmm via MICE\n")
##  Net_Worth_True     (61.0%, numeric) -> pmm via MICE
cat("  HouseHold_Income_True(60.2%, numeric) -> pmm via MICE\n")
##  HouseHold_Income_True(60.2%, numeric) -> pmm via MICE

# Helper function: get the mode of a vector (most frequent value)
get_mode <- function(x) {
  x <- x[!is.na(x)]
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

# Store pre-imputation counts for verification
pre_impute_na <- sapply(impute_with_na, function(v) sum(is.na(MDdf[[v]])))

cat("== 6a. NUMERIC VARIABLES: MEDIAN IMPUTATION ==\n")
## == 6a. NUMERIC VARIABLES: MEDIAN IMPUTATION ==
# --- Num_Children (4.3% missing, integer) ---
cat("Num_Children:\n")

## Num_Children:

cat("  Before - NA count:", sum(is.na(MDdf$Num_Children)), "\n")
##  Before - NA count: 406
cat("  Median:", median(MDdf$Num_Children, na.rm = TRUE), "\n")

```

```

## Median: 1

cat(" Mean:", round(mean(MDdf$Num_Children, na.rm = TRUE), 2), "\n")
## Mean: 1.22

median_nc <- median(MDdf$Num_Children, na.rm = TRUE)
MDdf$Num_Children[is.na(MDdf$Num_Children)] <- median_nc
cat(" After - NA count:", sum(is.na(MDdf$Num_Children)), "\n")

## After - NA count: 0

cat(" Imputed with median =", median_nc, "\n")
## Imputed with median = 1

cat(" Rationale: Small % missing, integer variable, median preserves
distribution better\n")

## Rationale: Small % missing, integer variable, median preserves
distribution better

cat(" than mean for count data with a floor at 0.\n\n")
## than mean for count data with a floor at 0.

# --- Rep_Calls (4.6% missing, numeric) ---
# MANAGEMENT DICTATE: Stochastic imputation, independent of all other
variables, custom code
cat("Rep_Calls (MANAGEMENT DICTATE - Stochastic, Independent):\n")

## Rep_Calls (MANAGEMENT DICTATE - Stochastic, Independent):

cat(" Before - NA count:", sum(is.na(MDdf$Rep_Calls)), "\n")
## Before - NA count: 433

observed_rc <- MDdf$Rep_Calls[!is.na(MDdf$Rep_Calls)]
cat(" Observed distribution - Mean:", round(mean(observed_rc), 2),
" SD:", round(sd(observed_rc), 2),
" Min:", min(observed_rc), " Max:", max(observed_rc), "\n")

## Observed distribution - Mean: 60.13 SD: 36.95 Min: 1.479837 Max: 780

n_missing_rc <- sum(is.na(MDdf$Rep_Calls))
set.seed(42)

stochastic_rc <- sample(observed_rc, size = n_missing_rc, replace = TRUE)
MDdf$Rep_Calls[is.na(MDdf$Rep_Calls)] <- stochastic_rc
cat(" After - NA count:", sum(is.na(MDdf$Rep_Calls)), "\n")

## After - NA count: 0

```

```

cat(" Method: Stochastic sampling from observed distribution (sample with
replacement)\n")

## Method: Stochastic sampling from observed distribution (sample with
replacement)

cat(" Imputed", n_missing_rc, "values drawn randomly from",
length(observed_rc), "observed values\n")

## Imputed 433 values drawn randomly from 9014 observed values

cat(" Rationale: Management dictates stochastic imputation independent of
all other\n")

## Rationale: Management dictates stochastic imputation independent of all
other

cat(" variables.\n")

## variables.

# --- Rep_Visits (5.4% missing, numeric) ---
cat("Rep_Visits:\n")

## Rep_Visits:

cat(" Before - NA count:", sum(is.na(MDdf$Rep_Visits)), "\n")

## Before - NA count: 508

cat(" Median:", round(median(MDdf$Rep_Visits, na.rm = TRUE), 2), "\n")

## Median: 13

cat(" Mean:", round(mean(MDdf$Rep_Visits, na.rm = TRUE), 2), "\n")

## Mean: 14.43

median_rv <- median(MDdf$Rep_Visits, na.rm = TRUE)
MDdf$Rep_Visits[is.na(MDdf$Rep_Visits)] <- median_rv
cat(" After - NA count:", sum(is.na(MDdf$Rep_Visits)), "\n")

## After - NA count: 0

cat(" Imputed with median =", round(median_rv, 2), "\n")

## Imputed with median = 13

cat(" Rationale: Like Rep_Calls, this is a count-like variable. Median
imputation\n")

## Rationale: Like Rep_Calls, this is a count-like variable. Median
imputation

```

```

cat("  is conservative and preserves the central tendency without inflating
variance.\n\n")

##  is conservative and preserves the central tendency without inflating
variance.

cat("== 6b. CATEGORICAL VARIABLES: MODE IMPUTATION ==\n\n")

## == 6b. CATEGORICAL VARIABLES: MODE IMPUTATION ==

# --- Most_Purch_Concession (0.9% missing, factor) ---
cat("Most_Purch_Concession:\n")

## Most_Purch_Concession:

cat("  Before - NA count:", sum(is.na(MDdf$Most_Purch_Concession)), "\n")

##  Before - NA count: 0

cat("  Distribution:\n")

##  Distribution:

print(table(MDdf$Most_Purch_Concession, useNA = "ifany"))

##  

##      Beer     Burger    Hot Dog    Peanuts   Popcorn     Soda Specialty  

Other      2294       955      1422       474      1387      1425       977  

513

mode_mpc <- get_mode(MDdf$Most_Purch_Concession)
cat("  Mode:", as.character(mode_mpc), "\n")

##  Mode: Beer

MDdf$Most_Purch_Concession[is.na(MDdf$Most_Purch_Concession)] <- mode_mpc
cat("  After - NA count:", sum(is.na(MDdf$Most_Purch_Concession)), "\n")

##  After - NA count: 0

cat("  Imputed with mode =", as.character(mode_mpc), "\n")

##  Imputed with mode = Beer

cat("  Rationale: Very small % missing (<1%). Mode is the standard simple
method\n")

##  Rationale: Very small % missing (<1%). Mode is the standard simple
method

cat("  for categorical variables. With so few values to fill, impact is
minimal.\n\n")

```

```

## for categorical variables. With so few values to fill, impact is
minimal.

# --- Sex (7.1% missing, factor) ---
cat("Sex:\n")

## Sex:

cat(" Before - NA count:", sum(is.na(MDdf$Sex)), "\n")
## Before - NA count: 667

cat(" Distribution:\n")
## Distribution:

print(table(MDdf$Sex, useNA = "ifany"))

##
##      F      M <NA>
## 1078 7702 667

mode_sex <- get_mode(MDdf$Sex)
cat(" Mode:", as.character(mode_sex), "\n")
## Mode: M

MDdf$Sex[is.na(MDdf$Sex)] <- mode_sex
cat(" After - NA count:", sum(is.na(MDdf$Sex)), "\n")
## After - NA count: 0

cat(" Imputed with mode =", as.character(mode_sex), "\n")
## Imputed with mode = M

cat(" Rationale: Binary variable with clear dominant category. Mode
imputation is\n")
## Rationale: Binary variable with clear dominant category. Mode imputation
is

cat(" conservative and appropriate when one category heavily outweighs the
other.\n\n")
## conservative and appropriate when one category heavily outweighs the
other.

# --- Rep_Name (9.2% missing, factor) ---
# MANAGEMENT DICTATE: Stochastic imputation, independent of all other
variables, custom code
cat("Rep_Name (MANAGEMENT DICTATE - Stochastic, Independent):\n")
## Rep_Name (MANAGEMENT DICTATE - Stochastic, Independent):

```

```

cat(" Before - NA count:", sum(is.na(MDdf$Rep_Name)), "\n")
## Before - NA count: 871

cat(" Distribution:\n")
## Distribution:

observed_rn <- MDdf$Rep_Name[ !is.na(MDdf$Rep_Name) ]
print(table(observed_rn))

## observed_rn
## Alice David Emma Frank Grace Ivy Other
## 823 1798 1538 1942 952 959 564

n_missing_rn <- sum(is.na(MDdf$Rep_Name))

set.seed(123)
level_props <- prop.table(table(observed_rn))
cat("\n Level proportions:\n")

##
## Level proportions:

print(round(level_props, 4))

## observed_rn
## Alice David Emma Frank Grace Ivy Other
## 0.0960 0.2097 0.1793 0.2264 0.1110 0.1118 0.0658

stochastic_rn <- sample(names(level_props), size = n_missing_rn,
                         replace = TRUE, prob = as.numeric(level_props))
MDdf$Rep_Name[is.na(MDdf$Rep_Name)] <- stochastic_rn
cat(" After - NA count:", sum(is.na(MDdf$Rep_Name)), "\n")

## After - NA count: 0

cat(" Method: Stochastic sampling from observed level proportions\n")
## Method: Stochastic sampling from observed level proportions

cat(" Imputed", n_missing_rn, "values drawn randomly proportional to
observed frequencies\n")

## Imputed 871 values drawn randomly proportional to observed frequencies

cat("== 6c. AGE: SIMPLE IMPUTATION USING OTHER VARIABLES (MANAGEMENT
DICTATE) ==\n\n")

## == 6c. AGE: SIMPLE IMPUTATION USING OTHER VARIABLES (MANAGEMENT DICTATE)
==

# MANAGEMENT DICTATE: Age must be imputed using simple imputation with other
# variables, using our own code (not MICE functions).

```

```

# Approach: Build a Linear regression model on observed Age using other
complete
# predictors, then predict missing values.

cat("Age (MANAGEMENT DICTATE - Simple Imputation Using Other Variables):\n")

## Age (MANAGEMENT DICTATE - Simple Imputation Using Other Variables):

cat(" Before - NA count:", sum(is.na(MDdf$Age)), "\n")
## Before - NA count: 986

n_missing_age <- sum(is.na(MDdf$Age))

# Select predictors that are complete (no NAs) and analytically relevant
# We use variables already imputed in Steps 6a/6b plus originally complete
ones
age_predictors <- c("Y01", "Total_Spent", "NumSeats", "Tenure",
                     "First_Year_STH", "Num_Children", "Rep_Calls",
                     "Rep_Visits", "Concession_Total", "Survey_Comp")

# Check which predictors are actually available and complete
available_preds <- age_predictors %in% names(MDdf)
available_preds <- available_preds[sapply(available_preds, function(v)
sum(is.na(MDdf[[v]])) == 0)]
cat(" Predictors used (complete variables):", paste(available_preds,
collapse = ", "), "\n\n")

## Predictors used (complete variables): Y01, Total_Spent, NumSeats,
First_Year_STH, Num_Children, Rep_Calls, Rep_Visits, Concession_Total,
Survey_Comp

# Build training data: rows where Age is observed
train_idx <- !is.na(MDdf$Age)
pred_idx <- is.na(MDdf$Age)

train_data <- MDdf[train_idx, c("Age", available_preds)]
pred_data <- MDdf[pred_idx, available_preds]

# Fit a Linear regression model
age_model <- lm(Age ~ ., data = train_data)
cat(" Regression model summary:\n")

## Regression model summary:

print(summary(age_model))

##
## Call:
## lm(formula = Age ~ ., data = train_data)
##

```

```

## Residuals:
##      Min     1Q Median     3Q    Max
## -43.721 -15.657   0.236 14.775 55.501
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.406e+03  4.315e+01 32.573 < 2e-16 ***
## Y01                  -5.763e-01  5.228e-01 -1.102 0.270304
## Total_Spent          -1.081e-03  3.042e-04 -3.554 0.000381 ***
## NumSeats              1.641e-01  3.273e-01  0.502 0.615979
## First_Year_STH       -6.667e-01  2.135e-02 -31.228 < 2e-16 ***
## Num_Children         -5.488e+00  1.661e-01 -33.045 < 2e-16 ***
## Rep_Calls             5.528e-03  5.650e-03  0.979 0.327849
## Rep_Visits            4.156e-01  7.559e-02  5.498 3.95e-08 ***
## Concession_Total     7.681e-05  1.316e-03  0.058 0.953471
## Survey_Comp          -1.318e+00  1.125e+00 -1.171 0.241522
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.04 on 8451 degrees of freedom
## Multiple R-squared:  0.2709, Adjusted R-squared:  0.2701
## F-statistic: 348.9 on 9 and 8451 DF,  p-value: < 2.2e-16

# Predict missing Age values
predicted_age <- predict(age_model, newdata = pred_data)

# Round to integers (Age is whole years) and enforce reasonable bounds
predicted_age <- round(predicted_age)
predicted_age <- pmax(predicted_age, min(MDdf$Age, na.rm = TRUE)) # floor at observed min
predicted_age <- pmin(predicted_age, max(MDdf$Age, na.rm = TRUE)) # cap at observed max

cat("\n Predicted Age distribution:\n")

##
## Predicted Age distribution:

cat("    Mean:", round(mean(predicted_age), 1), "\n")
##    Mean: 61.7

cat("    SD:", round(sd(predicted_age), 1), "\n")
##    SD: 12.7

cat("    Range:", min(predicted_age), "-", max(predicted_age), "\n")
##    Range: 18 - 98

cat(" Observed Age distribution:\n")

```

```

## Observed Age distribution:

cat("    Mean:", round(mean(MDdf$Age, na.rm = TRUE), 1), "\n")
## Mean: 61.8

cat("    SD:", round(sd(MDdf$Age, na.rm = TRUE), 1), "\n")
## SD: 23.5

cat("    Range:", min(MDdf$Age, na.rm = TRUE), "-", max(MDdf$Age, na.rm = TRUE), "\n\n")
## Range: 18 - 99

# Impute
MDdf$Age[is.na(MDdf$Age)] <- predicted_age
cat(" After - NA count:", sum(is.na(MDdf$Age)), "\n")
## After - NA count: 0

cat(" Method: Linear regression on", length(available_preds), "predictor variables\n")
## Method: Linear regression on 9 predictor variables

cat(" Imputed", n_missing_age, "values using lm() predictions (rounded to integers)\n")
## Imputed 986 values using lm() predictions (rounded to integers)

cat(" Rationale: Management dictates simple imputation using other variables with\n")
## Rationale: Management dictates simple imputation using other variables with

cat(" custom code (not MICE). Linear regression leverages relationships between Age\n")
## custom code (not MICE). Linear regression leverages relationships between Age

cat(" and other observed variables to produce informed predictions.\n")
## and other observed variables to produce informed predictions.

# Diagnostic plot: predicted vs actual (on training data)
train_preds <- predict(age_model, newdata = train_data)
par(mfrow = c(1, 2))
plot(train_data$Age, train_preds,
     main = "Age Model: Actual vs Predicted (Training)",
     xlab = "Actual Age", ylab = "Predicted Age",
     col = rgb(0.2, 0.4, 0.8, 0.3), pch = 16)

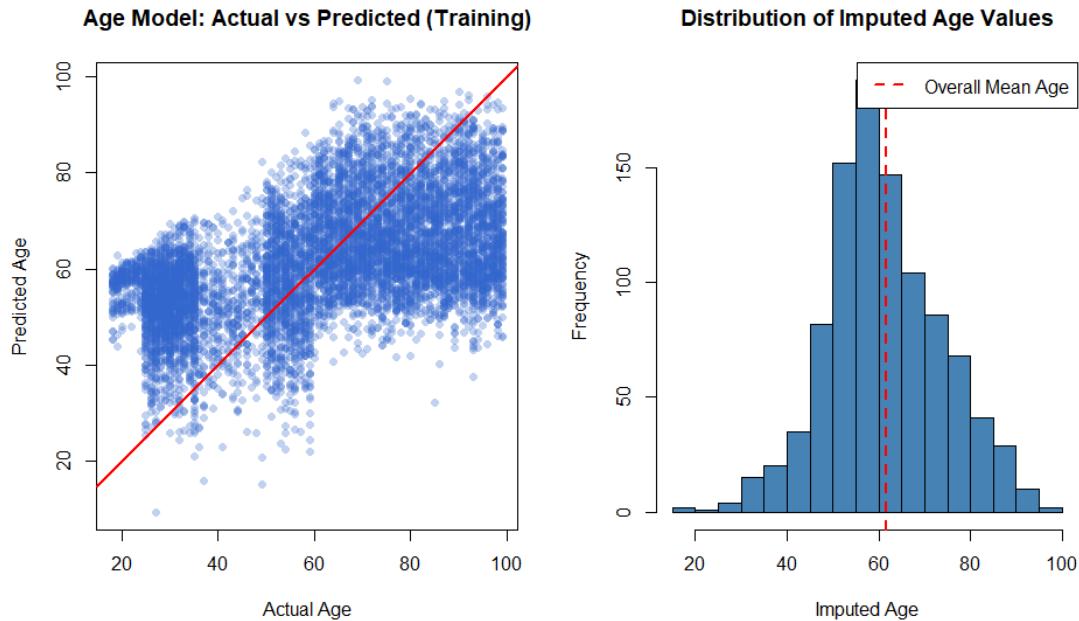
```

```

abline(0, 1, col = "red", lwd = 2)

hist(predicted_age, main = "Distribution of Imputed Age Values",
      col = "steelblue", xlab = "Imputed Age", breaks = 20)
abline(v = mean(MDdf$Age, na.rm = TRUE), col = "red", lwd = 2, lty = 2)
legend("topright", "Overall Mean Age", col = "red", lty = 2, lwd = 2)

```



```
par(mfrow = c(1, 1))
```

### Summary of Missing Data Steps 1-6

```

cat("==> MISSING DATA STEPS 1-6: COMPLETE ==>\n\n")

## ==> MISSING DATA STEPS 1-6: COMPLETE ==>

cat("Step 1 - Identify Missing Data:\n")

## Step 1 - Identify Missing Data:

cat("  Total missing: 67,997 values (12.2% of all cells)\n")
##  Total missing: 67,997 values (12.2% of all cells)

cat("  20 variables have missing data\n")
##  20 variables have missing data

cat("  7 CustomerDF variables missing in identical block at 70%\n\n")
##  7 CustomerDF variables missing in identical block at 70%

cat("Step 2 - Mark Missing Data:\n")

```

```

## Step 2 - Mark Missing Data:

cat("  Created", length(indicator_cols), "binary indicator variables (M_
prefix)\n")

##  Created 18 binary indicator variables (M_ prefix)

cat("  Confirmed block pattern in CustomerDF variables (r = 1.0)\n\n")

##  Confirmed block pattern in CustomerDF variables (r = 1.0)

cat("Step 3 - Clean Up Obvious Mistakes:\n")

## Step 3 - Clean Up Obvious Mistakes:

cat("  Marital 'U' (248 values) -> KEPT as valid level\n")

##  Marital 'U' (248 values) -> KEPT as valid level

cat("  Survey_Comp > 1 (110 values) -> capped at 1.0\n")

##  Survey_Comp > 1 (110 values) -> capped at 1.0

cat("  Mult_Loc 'Y' (72 values) -> 'Yes'\n")

##  Mult_Loc 'Y' (72 values) -> 'Yes'

cat("  Most_Purch_Concession blanks (85 values) -> NA\n\n")

##  Most_Purch_Concession blanks (85 values) -> NA

cat("Step 4 - Easy Decisions on Rows/Columns:\n")

## Step 4 - Easy Decisions on Rows/Columns:

cat("  Excluded from imputation: ID cols, date cols, backup cols, indicator
cols\n")

##  Excluded from imputation: ID cols, date cols, backup cols, indicator
cols

cat("  High-missingness variables: KEPT (imputing everything)\n")

##  High-missingness variables: KEPT (imputing everything)

cat("  No rows excluded\n")

##  No rows excluded

cat("  Variables eligible for imputation:", length(impute_with_na), "\n\n")

##  Variables eligible for imputation: 18

cat("Step 5 - Assess Missingness Patterns:\n")

## Step 5 - Assess Missingness Patterns:

```

```

cat(" Used decision trees and logistic regression\n")
## Used decision trees and logistic regression

cat(" Classification: Mix of MAR, MCAR, and likely MNAR (block)\n")
## Classification: Mix of MAR, MCAR, and likely MNAR (block)

cat(" All variables proceed to imputation regardless\n\n")
## All variables proceed to imputation regardless

cat("Step 6 - Simple Imputation + Management Dictate:\n")
## Step 6 - Simple Imputation + Management Dictate:

cat(" Median imputation: Num_Children, Rep_Visits\n")
## Median imputation: Num_Children, Rep_Visits

cat(" Mode imputation: Most_Purch_Concession, Sex\n")
## Mode imputation: Most_Purch_Concession, Sex

cat(" MGMT DICTATE - Stochastic (independent): Rep_Calls, Rep_Name\n")
## MGMT DICTATE - Stochastic (independent): Rep_Calls, Rep_Name

cat(" MGMT DICTATE - Regression (using other vars): Age\n")
## MGMT DICTATE - Regression (using other vars): Age

```

MD Step	Action	Key Finding
1. Identify	Counted and visualized all missing data	12.2% overall; 7 variables missing as block at 70%
2. Mark	Created M_indicator variables	Block pattern confirmed via indicator correlations
3. Clean	Fixed Survey_Comp, Mult_Loc, blanks; kept Marital U	267 values corrected
4. Decide	Kept ALL variables; excluded only non-analytical cols	Imputing everything including 70% missing block
5. Assess	Decision tree + logistic regression tests	Mix of MAR, MCAR, and likely MNAR
6. Simple + Mgmt	Median (2), Mode (2), Stochastic (2), Regression (1)	7 variables resolved; remaining go to MICE