Analysis of Subscreen Package Example Code: PBC Dataset Implementation

July 10, 2025

This document explains a comprehensive R script that demonstrates the functionality of the **subscreen package** using the Primary Biliary Cholangitis (PBC) dataset. The code creates example data and showcases various subgroup analysis scenarios.

1 Data Setup and Preparation

1.1 Library Loading and Data Import

```r
library(survival)
library(dplyr)
utils::data(pbc, package = "survival")
```

The script begins by loading essential libraries and importing the famous PBC dataset from the `survival` package. PBC is a liver disease dataset commonly used in survival analysis research.

1.2 Categorical Variable Creation

The code systematically converts continuous variables into categorical ones using quantile-based cutoffs:

```r
pbc <- pbc %>%
  dplyr::mutate(
    ageg = dplyr::case_when(
      age <= quantile(pbc$age, 0.33, na.rm = TRUE) ~ "Low",
      age > quantile(pbc$age, 0.33, na.rm = TRUE) &
      age <= quantile(pbc$age, 0.66, na.rm = TRUE) ~ "Middle",
      age > quantile(pbc$age, 0.66, na.rm = TRUE) ~ "High",
      TRUE ~ "No data"
    ),
    # ... additional variables
  )
```

Variables created:

- **Age groups** (`ageg`): Low ($\leq$ 33rd percentile), Middle (33rd-66th), High ($>$ 66th)
- **Alkaline phosphatase** (`phosg`): Low/High (median split)
- **Albumin** (`albumg`): Low/Middle/High (tertiles)

- **AST** (astg): Low/Middle/High (tertiles)
- **Bilirubin** (bilig): Low/Middle/High (tertiles)
- **Cholesterol** (cholg): Low/High (median split)
- **Copper** (copperg): Low/Middle/High (tertiles)
- **Binary variables**: Ascites, Platelets, Spiders (Yes/No or Low/High)

2 Data Cleaning and Endpoint Creation

2.1 Treatment Group Filtering

```
pbcdat <- pbc[!is.na(pbc$trt), ]
```

This removes patients with missing treatment assignments, ensuring clean analysis groups.

2.2 Synthetic Endpoint Generation

```
set.seed(2006)
pbcdat$'event.pfs' <- sample(c(0, 1), dim(pbcdat)[1], replace = TRUE)
pbcdat$'timepfs' <- sample(1:5000, dim(pbcdat)[1], replace = TRUE)
pbcdat$'event.os' <- pbcdat$event # Use original event
pbcdat$'timeos' <- pbcdat$time    # Use original time
```

The script creates two types of survival endpoints:

- **PFS (Progression-Free Survival)**: Randomly generated for demonstration purposes
- **OS (Overall Survival)**: Uses the original PBC survival data for realistic analysis

3 Evaluation Function Definition

3.1 Hazard Ratio Calculation Function

```
hazardratio <- function(D) {
  HRpfs <- tryCatch(
    exp(coxph(Surv(D$timepfs, D$event.pfs) ~ D$trt)$coefficients[[1]]),
    warning = function(w) {NA}
  )
  HRpfs <- 1/HRpfs
  HR.pfs <- round(HRpfs, 2)
  HR.pfs[HR.pfs > 10] <- 10
  HR.pfs[HR.pfs < 0.00001] <- 0.00001

  # Similar calculation for OS...

  data.frame(HR.pfs, HR.os)
}
```

This function is **critical** for the subscreen analysis. It:

- Fits Cox proportional hazards models for both PFS and OS
- Calculates hazard ratios comparing treatments
- Inverts the HR (so HR $<$ 1 indicates treatment benefit)
- Includes robust error handling with `tryCatch`
- Caps extreme values between 0.00001 and 10
- **Must return a single-row data.frame** (subscreen package requirement)

4 Variable Importance Analysis

```r
importance <- subscreenvi(
  data = pbcdat,
  y = 'time',
  cens = 'status',
  trt = 'trt',
  x = c("sex", "ageg", "phosg", "albuming", "astg", "bilig",
        "cholg", "copperg", "ascitesg", "plateletg", "spidersg")
)
```

This calculates which variables are most important for predicting survival outcomes, helping prioritize which factors to examine in subgroup analysis.

5 Subgroup Analysis Scenarios

The code demonstrates multiple subgroup analysis configurations:

5.1 Scenario 1: Basic Individual Factor Analysis

```r
results <- subscreencalc(
  data = pbcdat,
  eval_function = hazardratio,
  subjectid = "id",
  factors = c("sex", "ageg", "phosg", ...),
  use_complement = FALSE,
  factorial = FALSE
)
```

Configuration:

- Analyzes each factor individually
- No complement groups
- No factorial combinations

5.2 Scenario 2: Factorial Combination Analysis

```r
results_factorial_true <- subscreencalc(
  ...,
  factorial = TRUE
)
```

Configuration:

- Considers combinations of factors (e.g., ”Male + High Age”)
- Explores interaction effects between variables

5.3 Scenario 3: Complement Group Analysis

```r
results_complement_true <- subscreencalc(
  ...,
  use_complement = TRUE
)
```

Configuration:

- Analyzes the ”opposite” of each subgroup
- Example: if analyzing ”Males”, also analyzes ”Females”
- Provides comprehensive coverage of patient population

5.4 Scenario 4: Full Analysis with Funnel Plot

```r
results_factorial_complement_true <- subscreenfunnel(
  data = pbcdat,
  H = results_factorial_complement_true1,
  eval_function = hazardratio,
  min_start = 15,
  n_support_points = 25,
  nperm = 1500,
  alpha = c(0.05, 0.1),
  treat = "trt",
  endpoints = c("timepfs", "event.pfs", "timeos", "event.os")
)
```

This is the **key function** that creates funnel plots to identify subgroups with unusually good or bad treatment effects. It:

- Uses permutation testing (1500 permutations)
- Tests at multiple significance levels (5% and 10%)
- Considers both PFS and OS endpoints
- Identifies statistically significant subgroup effects
- Creates visual funnel plots for interpretation

6 Data Persistence

```r
save(results, file = "data/results.rda")
save(results_factorial_true, file = "data/results_factorial_true.rda")
save(results_factorial_complement_true, file = "data/results_factorial_
    complement_true.rda")
save(results_complement_true, file = "data/results_complement_true.rda"
    )
save(importance, file = "data/importance.rda")
```

All analysis results are saved for future use and comparison.

7 What This Code Demonstrates

This comprehensive example script shows how to:

1. **Prepare clinical trial data** for subgroup analysis
2. **Create categorical variables** from continuous measurements
3. **Define evaluation functions** that calculate treatment effects
4. **Run variable importance analysis** to prioritize factors
5. **Perform subgroup screening** under different scenarios
6. **Create funnel plots** to identify promising subgroups
7. **Handle multiple endpoints** simultaneously
8. **Save and organize results** for further analysis

8 Clinical Context and Applications

In real clinical trial analysis, this type of comprehensive subgroup screening would help researchers:

8.1 Primary Applications

- **Identify patient subgroups** who benefit more or less from treatment
- **Prioritize biomarkers** for further investigation
- **Design future trials** targeting specific patient populations
- **Support regulatory submissions** with robust subgroup evidence

8.2 Statistical Advantages

- **Multiple testing correction** through permutation methods
- **Visual interpretation** via funnel plots
- **Comprehensive coverage** of patient subgroups
- **Robust error handling** for real-world data challenges

9 Dataset Appropriateness

The PBC dataset serves as an excellent example because it has the typical structure of clinical trial data:

- Treatment assignments (`trt`)
- Survival times (`time`)
- Event indicators (`event`)
- Multiple baseline covariates (demographics, lab values, clinical factors)
- Realistic sample size and missing data patterns

This makes it an ideal teaching dataset for demonstrating subgroup analysis methodology in a clinically relevant context.