Illustrating imputation of missing data using the Titanic dataset

Steve Simon

2024-05-07

Multiple imputation is a way to properly account for missing values without causing bias. Simpler forms of imputation, such as replacing missing values with the mean of the non-missing values, can produce serious problems. You might think that ignoring the missingness and relying just on records with complete data for all key variables would be acceptable, but this also can produce serious problems. I want to illustrate a simple example of multiple imputation using data on mortality from the Titanic.

Here is a brief description of this dataset, taken from the data dictionary on my github site.

The Titanic was a large cruise ship, the biggest of its kind in 1912. It was thought to be unsinkable, but when it set sail from England to America in its maiden voyage, it struck an iceberg and sank, killing many of the passengers and crew. You can get fairly good data on the characteristics of passengers who died and compare them to those that survived. The data indicate a strong effect due to age and gender, representing a philosophy of "women and children first" that held during the boarding of life boats.

Here are the first few rows of data.

```
##                                          Name PClass   Age    Sex
## 1                  Allen, Miss Elisabeth Walton    1st 29.00 female
## 2                   Allison, Miss Helen Loraine    1st  2.00 female
## 3           Allison, Mr Hudson Joshua Creighton    1st 30.00   male
## 4 Allison, Mrs Hudson JC (Bessie Waldo Daniels)    1st 25.00 female
## 5                 Allison, Master Hudson Trevor    1st  0.92   male
## 6                           Anderson, Mr Harry    1st 47.00   male
##   Survived
## 1        1
## 2        0
## 3        0
## 4        0
```

```
## 5        1
## 6        1
```

I have hidden the R code up to this point, as it is mundane and not of great interest. I will show the R code and output for the rest of the analysis.

Notice the large number of missing values for age. The first three passengers with missing ages are

```
missing_rows <- which(is.na(ti0$Age))[1:3]
ti0 %>% slice(missing_rows)
```

```
##                           Name PClass Age    Sex Survived
## 1 Aubert, Mrs Leontine Pauline    1st  NA female        1
## 2     Barkworth, Mr Algernon H    1st  NA   male        1
## 3           Baumann, Mr John D    1st  NA   male        0
```

Create multiply imputed values for age. The default is to use every variable in the dataset other than age to impute the value of age. You don't want to use the name of the passenger, of course, so be sure to drop it before the imputation step.

```
ti1 <- ti0
ti1$i_female <- as.numeric(ti1$Sex=="female")
ti1 <- ti1[ , c("PClass", "Age", "Survived", "i_female")]
ti_mice <- mice(ti1)
```

```
##
##  iter imp variable
##   1   1  Age
##   1   2  Age
##   1   3  Age
##   1   4  Age
##   1   5  Age
##   2   1  Age
##   2   2  Age
##   2   3  Age
##   2   4  Age
##   2   5  Age
##   3   1  Age
```

```
##   3   2  Age
##   3   3  Age
##   3   4  Age
##   3   5  Age
##   4   1  Age
##   4   2  Age
##   4   3  Age
##   4   4  Age
##   4   5  Age
##   5   1  Age
##   5   2  Age
##   5   3  Age
##   5   4  Age
##   5   5  Age
```

```
## Warning: Number of logged events: 1
```

The mice function creates a complex object. Go ahead and explore the various components, but be fore-warned that this is messy. You can extract simple parts of the imputation with various functions. The complete function shows the complete datasets with the imputed values. Here are the first three rows where age was imputed the first time.

```
ti_mice %>%
  complete(1) %>%
  slice(missing_rows)
```

```
##   PClass Age Survived i_female
## 1    1st  60        1        1
## 2    1st  39        1        0
## 3    1st  17        0        0
```

Look at the imputations for the second, third, etc. times. Notice that the values change with each imputation.

```
ti_mice %>%
  complete(2) %>%
  slice(missing_rows)
```

```
##   PClass Age Survived i_female
## 1    1st  24        1        1
## 2    1st  23        1        0
## 3    1st  37        0        0
```

```
ti_mice %>%
  complete(3) %>%
  slice(missing_rows)
```

```
##   PClass Age Survived i_female
## 1    1st  18        1        1
## 2    1st  35        1        0
## 3    1st  32        0        0
```

```
ti_mice %>%
  complete(4) %>%
  slice(missing_rows)
```

```
##   PClass Age Survived i_female
## 1    1st  28        1        1
## 2    1st  27        1        0
## 3    1st  48        0        0
```

```
ti_mice %>%
  complete(5) %>%
  slice(missing_rows)
```

```
##   PClass Age Survived i_female
## 1    1st   3        1        1
## 2    1st  21        1        0
## 3    1st  31        0        0
```

Use the with function of the mice package to fit a model to apply a particular analysis to each of the five imputed datasets.

```
ti_with <- with(
  ti_mice,
  glm(
    Survived~PClass+rcs(Age, 3)+i_female,
    family=binomial))
```

Again, the object created here is complex. You can extract the individual analyses fairly easily. Here is the analysis of the first imputed dataset.

```
ti_with$analyses[[1]]
```

```
##
## Call:  glm(formula = Survived ~ PClass + rcs(Age, 3) + i_female, family = binomial
##
## Coefficients:
##     (Intercept)         PClass2nd         PClass3rd    rcs(Age, 3)Age
##         1.74344          -1.01365          -2.30009          -0.08354
## rcs(Age, 3)Age'          i_female
##         0.06150           2.49171
##
## Degrees of Freedom: 1312 Total (i.e. Null);  1307 Residual
## Null Deviance:           1688
## Residual Deviance: 1130      AIC: 1142
```

The key variable here is i_female, which shows how much different the log odds of survival are between men and women.

Notice that the estimate for sexmale changes from one analysis to another, though not by much.

```
ti_with$analyses[[2]]
```

```
##
## Call:  glm(formula = Survived ~ PClass + rcs(Age, 3) + i_female, family = binomial
##
## Coefficients:
##     (Intercept)         PClass2nd         PClass3rd    rcs(Age, 3)Age
##         0.93884          -1.04663          -2.33136          -0.04927
```

```
## rcs(Age, 3)Age'         i_female
##         0.02058          2.47974
##
## Degrees of Freedom: 1312 Total (i.e. Null);  1307 Residual
## Null Deviance:           1688
## Residual Deviance: 1167      AIC: 1179
```

```
ti_with$analyses[[3]]
```

```
##
## Call:  glm(formula = Survived ~ PClass + rcs(Age, 3) + i_female, family = binomial
##
## Coefficients:
##     (Intercept)        PClass2nd        PClass3rd   rcs(Age, 3)Age
##         1.73209         -0.96838         -2.18861         -0.08233
## rcs(Age, 3)Age'         i_female
##         0.07460          2.41404
##
## Degrees of Freedom: 1312 Total (i.e. Null);  1307 Residual
## Null Deviance:           1688
## Residual Deviance: 1149      AIC: 1161
```

```
ti_with$analyses[[4]]
```

```
##
## Call:  glm(formula = Survived ~ PClass + rcs(Age, 3) + i_female, family = binomial
##
## Coefficients:
##     (Intercept)        PClass2nd        PClass3rd   rcs(Age, 3)Age
##         0.71786         -0.99987         -2.36536         -0.03671
## rcs(Age, 3)Age'         i_female
##         0.01279          2.35473
##
## Degrees of Freedom: 1312 Total (i.e. Null);  1307 Residual
## Null Deviance:           1688
## Residual Deviance: 1173      AIC: 1185
```

```
ti_with$analyses[[5]]
```

```
##
## Call:  glm(formula = Survived ~ PClass + rcs(Age, 3) + i_female, family = binomial
##
## Coefficients:
##     (Intercept)        PClass2nd        PClass3rd   rcs(Age, 3)Age
##         0.44374         -0.87842         -2.18108         -0.03541
## rcs(Age, 3)Age'         i_female
##         0.04429          2.39806
##
## Degrees of Freedom: 1312 Total (i.e. Null);  1307 Residual
## Null Deviance:           1688
## Residual Deviance: 1188      AIC: 1200
```

Combine all these analyses with the pool function.

```
ti_pool <- pool(ti_with)
ti_summary <- summary(ti_pool)
ti_summary
```

```
##              term    estimate  std.error  statistic         df
## 1     (Intercept)  1.11519494 0.72644969   1.535130   6.035133
## 2       PClass2nd -0.98139084 0.21817393  -4.498204 283.401534
## 3       PClass3rd -2.27329763 0.21515276 -10.565970 107.396225
## 4  rcs(Age, 3)Age -0.05744986 0.02887569  -1.989558   5.783234
## 5 rcs(Age, 3)Age'  0.04275140 0.03332546   1.282845   7.057096
## 6        i_female  2.42765706 0.16719551  14.519870 169.311495
##        p.value
## 1 1.753695e-01
## 2 1.000789e-05
## 3 0.000000e+00
## 4 9.556300e-02
## 5 2.400694e-01
## 6 0.000000e+00
```

To complete things, compute the odds ratio and confidence interval.

```r
female <- ti_summary$term=="i_female"
log_or <- ti_summary[female, "estimate"]
se <- ti_summary[female, "std.error"]
or <- round(exp(log_or), 1)
lo <- round(exp(log_or-1.96*se), 1)
hi <- round(exp(log_or+1.96*se), 1)
glue("Odds ratio for females is {or}, 95% CI ({lo},{hi})")
```

```
## Odds ratio for females is 11.3, 95% CI (8.2,15.7)
```