‘Impute or not’ a story of missing Titanic data

Quick Overview

> *Missingness is an inherent characteristics of any dataset. In this tutorial I will attempt to determine missingness in the Titanic training dataset obtained from Kaggle (https://www.kaggle.com/c/titanic/download/train.csv). Through **visualizing, analysing and imputing** missing values with the help of VIM (https://cran.r-project.org/web/packages/VIMGUI/vignettes/VIM-Imputation.pdf), BaylorEdPsych and mvnmle (https://cran.r-project.org/web/packages/BaylorEdPsych/BaylorEdPsych.pdf), and mice (https://cran.r-project.org/web/packages/mice/mice.pdf) packages written for R ver 3.4; I will attempt to fill-in the missing values with approximated predicted values.*

Visualize the data

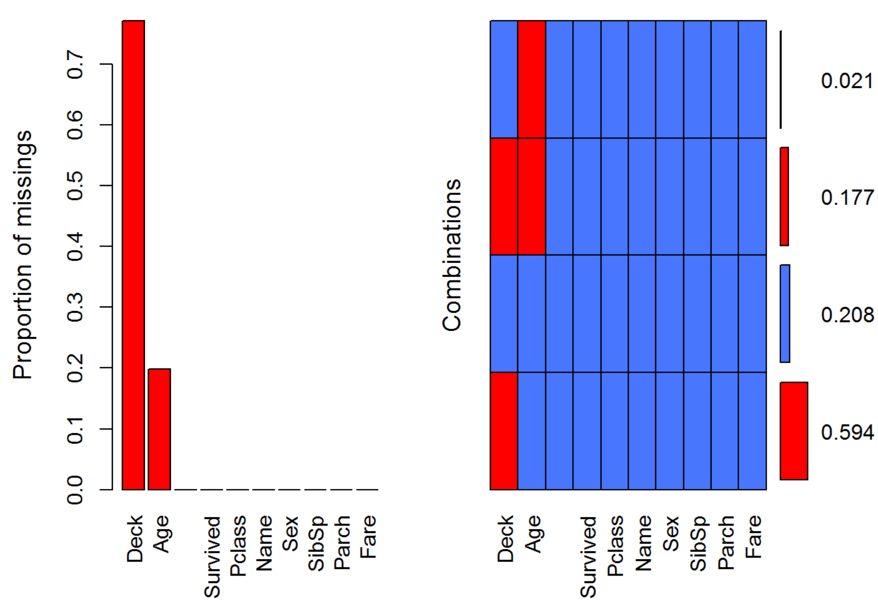
To start of let us read in the titanic data (*I saved it as train2*) and import the mice package. Next the *md.pattern* function will display a table with the missing values. Of the **total n=864** observations, there are n=177 missing values for **Age** variable and n=687 missing values for **Deck** variable.

```
train2<-read.csv("C:/Users/avi/Downloads/train2.csv")
library (mice)
md.pattern(train2)
```

```
##     PassengerId Survived Pclass Name Sex SibSp Parch Fare Age Deck
## 185           1        1      1    1   1     1     1    1   1    1   0
##  19           1        1      1    1   1     1     1    1   0    1   1
## 529           1        1      1    1   1     1     1    1   1    0   1
## 158           1        1      1    1   1     1     1    1   0    0   2
##               0        0      0    0   0     0     0    0 177  687 864
```

Below *aggr* function give us similar information as the above table, expect in a pretty visual produced by the VIM library. The red areas represents missing values in proportions, about **19% of Age is missing** and about **77% of Deck** variable is missing.

```
library(VIM)
aggr_plot<-aggr(train2, col=c('royalblue1','red1'), numbers=TRUE, sortVars=TRUE)
```



```
##
## Variables sorted by number of missings:
##     Variable     Count
##         Deck 0.7710438
##          Age 0.1986532
##  PassengerId 0.0000000
##     Survived 0.0000000
##       Pclass 0.0000000
##         Name 0.0000000
##          Sex 0.0000000
##        SibSp 0.0000000
##        Parch 0.0000000
##         Fare 0.0000000
```

MISSINGNESS ASSUMPTION

Next I am going to use the Little's test (https://cran.r-project.org/web/packages/BaylorEdPsych/BaylorEdPsych.pdf) to assess for missing completely at random (MCAR) assumption on the Age and Deck variables. MCAR assumption is satisfied if missing values are random and do not depend on the observed or missing values of y. (Little, 1988) (http://www.jstor.org.proxy.lib.duke.edu/stable/pdf/2290157.pdf)

$$(y_{obs}, y_{mis})$$

The Little's test is commonly used for checking the MCAR assumption; which, if found to be not significant, allows for rejection of the null hypothesis and it is safely assumed that ingnoring or dropping the missing values will not impact the analysis. Howerver, if the test is significant at p<.05 imputation is a resonable next step.

To run Little's test, I will install BaylorEdPsych and mvnmle package. LittleMCAR (i.e., Little test) will gives us chi-square statistics with degree of freedom and p-value. In this case the p value is 0, indicating MCAR assumtions are not met, which also implies that missingness is not random and we will have to use impution techniques to approximate missing values.

```
library(BaylorEdPsych)
library(mvnmle)
```

> LittleMCAR(train2)

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Chi-square	df	p-value
543.8286	26	0

Imputing missing values

Lets impute the missing values in Age and Deck variables, using the *rf random forest imputation method*; in the Multivariate Imputation by Chained Equations, MICE library. I decided to use Random Forest method becasue, it can handle both continious and categorial variable.

I decided to pick m=20 for imputation size, the conclusion was based on Rubin's formula (https://books.google.com/books?id=cNvTIOLs_WMC&printsec=frontcover&dq=1.+Rubin+DB.+Multiple+Imputation+for+Nonresponse+in+Surveys.+John+Wiley+%26+Sons;+New+York:+1987.+pp. for relative efficiency:

$$1/(1+F/M)$$

where F is the fraction of missing information and M is the number of imputations.

> ***Note: it took approximately 45 min to run the imputation on Window 10 icore7, go grab few cups of coffee or tea:)***

```
tempData2 <- mice(train2,m=20, method= 'rf', seed=500)
```

```
##
##  iter imp variable
##   1   1  Age  Deck
##   1   2  Age  Deck
##   1   3  Age  Deck
##   1   4  Age  Deck
##   1   5  Age  Deck
##   1   6  Age  Deck
##   1   7  Age  Deck
##   1   8  Age  Deck
##   1   9  Age  Deck
##   1   10  Age  Deck
##   1   11  Age  Deck
##   1   12  Age  Deck
##   1   13  Age  Deck
##   1   14  Age  Deck
##   1   15  Age  Deck
##   1   16  Age  Deck
##   1   17  Age  Deck
##   1   18  Age  Deck
##   1   19  Age  Deck
##   1   20  Age  Deck
##   2   1  Age  Deck
##   2   2  Age  Deck
##   2   3  Age  Deck
##   2   4  Age  Deck
##   2   5  Age  Deck
##   2   6  Age  Deck
##   2   7  Age  Deck
##   2   8  Age  Deck
##   2   9  Age  Deck
##   2   10  Age  Deck
##   2   11  Age  Deck
##   2   12  Age  Deck
##   2   13  Age  Deck
##   2   14  Age  Deck
##   2   15  Age  Deck
##   2   16  Age  Deck
##   2   17  Age  Deck
##   2   18  Age  Deck
##   2   19  Age  Deck
##   2   20  Age  Deck
##   3   1  Age  Deck
##   3   2  Age  Deck
##   3   3  Age  Deck
##   3   4  Age  Deck
##   3   5  Age  Deck
##   3   6  Age  Deck
##   3   7  Age  Deck
##   3   8  Age  Deck
##   3   9  Age  Deck
##   3   10  Age  Deck
##   3   11  Age  Deck
##   3   12  Age  Deck
##   3   13  Age  Deck
##   3   14  Age  Deck
##   3   15  Age  Deck
##   3   16  Age  Deck
##   3   17  Age  Deck
##   3   18  Age  Deck
##   3   19  Age  Deck
##   3   20  Age  Deck
##   4   1  Age  Deck
##   4   2  Age  Deck
##   4   3  Age  Deck
##   4   4  Age  Deck
##   4   5  Age  Deck
##   4   6  Age  Deck
##   4   7  Age  Deck
##   4   8  Age  Deck
##   4   9  Age  Deck
##   4   10  Age  Deck
##   4   11  Age  Deck
##   4   12  Age  Deck
##   4   13  Age  Deck
##   4   14  Age  Deck
##   4   15  Age  Deck
##   4   16  Age  Deck
##   4   17  Age  Deck
##   4   18  Age  Deck
##   4   19  Age  Deck
```

```
##   4   20  Age  Deck
##   5   1  Age  Deck
##   5   2  Age  Deck
##   5   3  Age  Deck
##   5   4  Age  Deck
##   5   5  Age  Deck
##   5   6  Age  Deck
##   5   7  Age  Deck
##   5   8  Age  Deck
##   5   9  Age  Deck
##   5   10  Age  Deck
##   5   11  Age  Deck
##   5   12  Age  Deck
##   5   13  Age  Deck
##   5   14  Age  Deck
##   5   15  Age  Deck
##   5   16  Age  Deck
##   5   17  Age  Deck
##   5   18  Age  Deck
##   5   19  Age  Deck
##   5   20  Age  Deck
```

```
modelFit2 <- with(tempData2,lm(Survived~Age+Pclass+SibSp+Parch+Fare+Deck))
summary(pool(modelFit2))
```

```
##                       est           se          t        df     Pr(>|t|)
## (Intercept)  0.9617464088 0.1285819275  7.4796391 102.51801 2.628764e-11
## Age         -0.0067739159 0.0013381110 -5.0622975 220.19233 8.739481e-07
## Pclass      -0.2087157178 0.0275100308 -7.5868951 191.52312 1.388223e-12
## SibSp       -0.0426322766 0.0164272388 -2.5952186 465.46232 9.751562e-03
## Parch        0.0441344520 0.0214222005  2.0602203 731.83608 3.973048e-02
## Fare         0.0008234633 0.0004135462  1.9912245 419.61975 4.710416e-02
## Deck2        0.0974177147 0.1221475163  0.7975415  45.29043 4.292983e-01
## Deck3        0.0339439635 0.1035303635  0.3278648  68.73248 7.440105e-01
## Deck4        0.1276645675 0.1048453105  1.2176469  79.39773 2.269646e-01
## Deck5        0.1363932423 0.1226430476  1.1121156  43.37863 2.722141e-01
## Deck6        0.1132983250 0.1219226250  0.9292642  62.41046 3.563326e-01
## Deck7        0.0927709918 0.1724184014  0.5380574  68.46442 5.922823e-01
## Deck8       -0.2654135310 0.3769910741 -0.7040313 151.80085 4.824918e-01
##                    lo 95        hi 95 nmis        fmi     lambda
## (Intercept)  7.067202e-01  1.216772585   NA 0.39883352 0.38721879
## Age         -9.411060e-03 -0.004136772  177 0.24709658 0.24028890
## Pclass      -2.629773e-01 -0.154454173    0 0.27163610 0.26406959
## SibSp       -7.491301e-02 -0.010351543    0 0.13018325 0.12645383
## Parch        2.078157e-03  0.086190747    0 0.05746369 0.05489139
## Fare         1.058311e-05  0.001636343    0 0.14553925 0.14147639
## Deck2       -1.485565e-01  0.343391905   NA 0.62034921 0.60394622
## Deck3       -1.726078e-01  0.240495730   NA 0.49852060 0.48413765
## Deck4       -8.100855e-02  0.336337684   NA 0.46069699 0.44728112
## Deck5       -1.108776e-01  0.383664103   NA 0.63399001 0.61749514
## Deck6       -1.303896e-01  0.356986299   NA 0.52504535 0.51006503
## Deck7       -2.512423e-01  0.436784301   NA 0.49957557 0.48516749
## Deck8       -1.010240e+00  0.479413286   NA 0.31537975 0.30641880
```

To check if the model can repoduce similar results, I will set higher seed value and re-run it.

```
tempData2 <- mice(train2,m=20, method= 'rf', seed=245836)
```

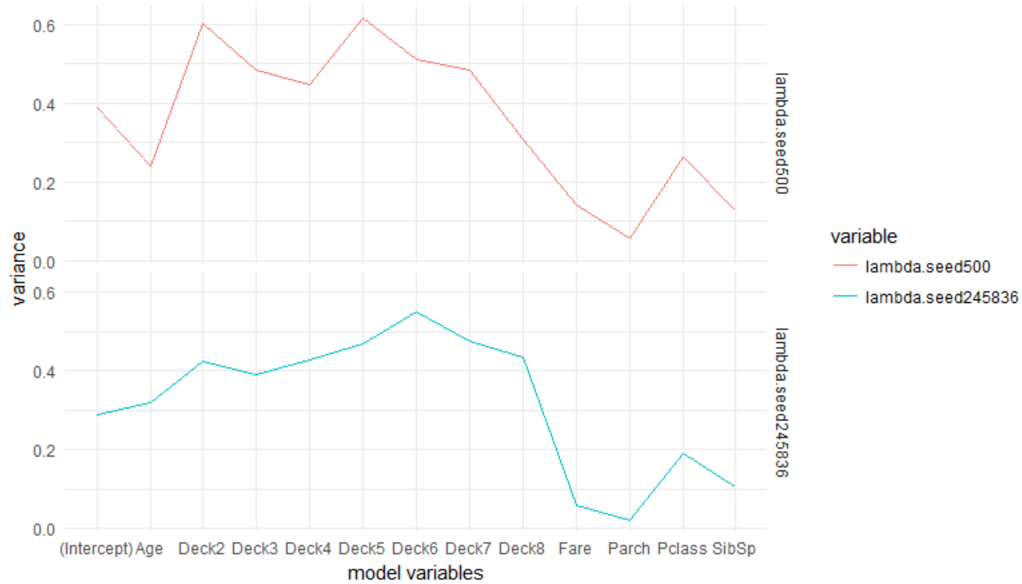
```
##
##  iter imp variable
##   1   1  Age  Deck
##   1   2  Age  Deck
##   1   3  Age  Deck
##   1   4  Age  Deck
##   1   5  Age  Deck
##   1   6  Age  Deck
##   1   7  Age  Deck
##   1   8  Age  Deck
##   1   9  Age  Deck
##   1   10  Age  Deck
##   1   11  Age  Deck
##   1   12  Age  Deck
##   1   13  Age  Deck
##   1   14  Age  Deck
##   1   15  Age  Deck
##   1   16  Age  Deck
##   1   17  Age  Deck
##   1   18  Age  Deck
##   1   19  Age  Deck
##   1   20  Age  Deck
##   2   1  Age  Deck
##   2   2  Age  Deck
##   2   3  Age  Deck
##   2   4  Age  Deck
##   2   5  Age  Deck
##   2   6  Age  Deck
##   2   7  Age  Deck
##   2   8  Age  Deck
##   2   9  Age  Deck
##   2   10  Age  Deck
##   2   11  Age  Deck
##   2   12  Age  Deck
##   2   13  Age  Deck
##   2   14  Age  Deck
##   2   15  Age  Deck
##   2   16  Age  Deck
##   2   17  Age  Deck
##   2   18  Age  Deck
##   2   19  Age  Deck
##   2   20  Age  Deck
##   3   1  Age  Deck
##   3   2  Age  Deck
##   3   3  Age  Deck
##   3   4  Age  Deck
##   3   5  Age  Deck
##   3   6  Age  Deck
##   3   7  Age  Deck
##   3   8  Age  Deck
##   3   9  Age  Deck
##   3   10  Age  Deck
##   3   11  Age  Deck
##   3   12  Age  Deck
##   3   13  Age  Deck
##   3   14  Age  Deck
##   3   15  Age  Deck
##   3   16  Age  Deck
##   3   17  Age  Deck
##   3   18  Age  Deck
##   3   19  Age  Deck
##   3   20  Age  Deck
##   4   1  Age  Deck
##   4   2  Age  Deck
##   4   3  Age  Deck
##   4   4  Age  Deck
##   4   5  Age  Deck
##   4   6  Age  Deck
##   4   7  Age  Deck
##   4   8  Age  Deck
##   4   9  Age  Deck
##   4   10  Age  Deck
##   4   11  Age  Deck
##   4   12  Age  Deck
##   4   13  Age  Deck
##   4   14  Age  Deck
##   4   15  Age  Deck
##   4   16  Age  Deck
##   4   17  Age  Deck
##   4   18  Age  Deck
##   4   19  Age  Deck
```

```
##   4   20  Age  Deck
##   5   1  Age  Deck
##   5   2  Age  Deck
##   5   3  Age  Deck
##   5   4  Age  Deck
##   5   5  Age  Deck
##   5   6  Age  Deck
##   5   7  Age  Deck
##   5   8  Age  Deck
##   5   9  Age  Deck
##   5   10  Age  Deck
##   5   11  Age  Deck
##   5   12  Age  Deck
##   5   13  Age  Deck
##   5   14  Age  Deck
##   5   15  Age  Deck
##   5   16  Age  Deck
##   5   17  Age  Deck
##   5   18  Age  Deck
##   5   19  Age  Deck
##   5   20  Age  Deck
```

```
modelFit2 <- with(tempData2,lm(Survived~Age+Pclass+SibSp+Parch+Fare+Deck))
summary(pool(modelFit2))
```

```
##                       est           se          t        df     Pr(>|t|)
## (Intercept)  0.9922096929 0.1184804604  8.3744584 167.34091 2.131628e-14
## Age         -0.0070602395 0.0014159265 -4.9863036 141.81045 1.770326e-06
## Pclass      -0.2137837799 0.0263171364 -8.1233679 300.70302 1.199041e-14
## SibSp       -0.0426289510 0.0162717734 -2.6198098 536.31767 9.047078e-03
## Parch        0.0450752908 0.0210959838  2.1366764 838.00487 3.291335e-02
## Fare         0.0007891985 0.0003956438  1.9947199 714.48498 4.645291e-02
## Deck2        0.0813493309 0.1000859065  0.8127951  87.61244 4.185388e-01
## Deck3        0.0277050171 0.0936606849  0.2958020 101.10650 7.679880e-01
## Deck4        0.1172520198 0.1009580740  1.1613932  86.62652 2.486722e-01
## Deck5        0.1122658806 0.1038915513  1.0806065  72.71244 2.834434e-01
## Deck6        0.0784496074 0.1264399899  0.6204493  54.01860 5.375710e-01
## Deck7        0.1309519637 0.1710990234  0.7653578  71.36568 4.465823e-01
## Deck8       -0.1698740807 0.3731505089 -0.4552428  83.22713 6.501200e-01
##                    lo 95        hi 95 nmis        fmi     lambda
## (Intercept)  7.583006e-01  1.226118747   NA 0.29657386 0.28821669
## Age         -9.859291e-03 -0.004261188  177 0.32893987 0.31954196
## Pclass      -2.655729e-01 -0.161994698    0 0.19607428 0.19074502
## SibSp       -7.459317e-02 -0.010664727    0 0.10902234 0.10570595
## Parch        3.668118e-03  0.086482464    0 0.02449995 0.02217457
## Fare         1.243513e-05  0.001565962    0 0.06205801 0.05943617
## Deck2       -1.175626e-01  0.280261302   NA 0.43610604 0.42337885
## Deck3       -1.580902e-01  0.213500249   NA 0.40202796 0.39031524
## Deck4       -8.342528e-02  0.317929323   NA 0.43888336 0.42607635
## Deck5       -9.480348e-02  0.319335240   NA 0.48350240 0.46948855
## Deck6       -1.750453e-01  0.331944525   NA 0.56660871 0.55085435
## Deck7       -2.101795e-01  0.472083435   NA 0.48845334 0.47431551
## Deck8       -9.120255e-01  0.572277320   NA 0.44881014 0.43572196
```

I plot the amount of variance from the two models (seed 500 and seed 245836), to visualize if there are any difference. They both produced similar predicted missing values.

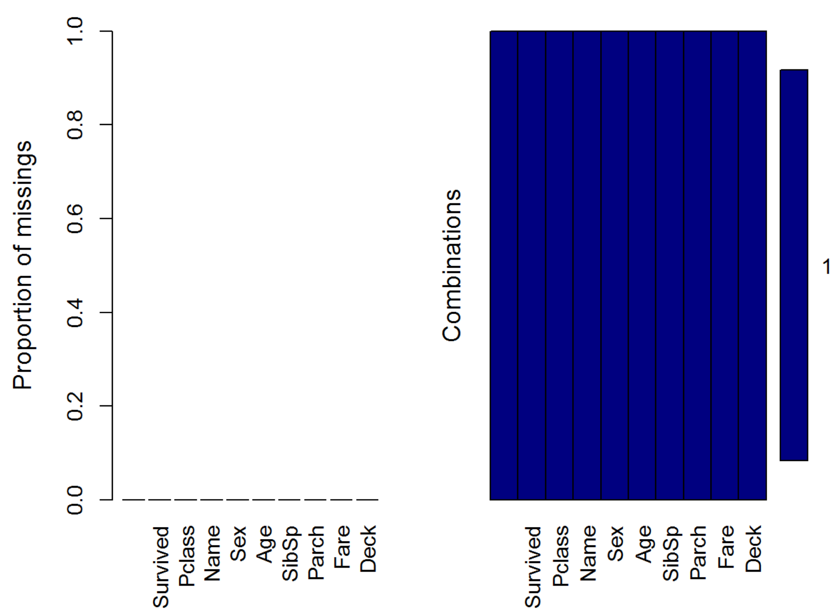


caption

If we visualize the complete data, we will see there are none. Great!

```
completeData2 <- complete(tempData2,1)


library(VIM)
aggr_plot<-aggr(completeData2, col=c('navyblue','red'), numbers=TRUE, sortVars=TRUE)
```




```
##
##  Variables sorted by number of missings:
##     Variable Count
##  PassengerId     0
##     Survived     0
##       Pclass     0
##         Name     0
##          Sex     0
##          Age     0
##        SibSp     0
##        Parch     0
##         Fare     0
##         Deck     0
```

Next lets run a Linear model of original data with missing values (the model will ignore the missing values) so we can plot it.

```
modelFit <-lm(Survived~Age+Pclass+SibSp+Parch+Fare+Deck, data=train2)
summary(modelFit)
```

```
##
## Call:
## lm(formula = Survived ~ Age + Pclass + SibSp + Parch + Fare +
##     Deck, data = train2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9716 -0.4480  0.1742  0.3331  0.7293
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.9795878  0.2146113   4.564  9.5e-06 ***
## Age         -0.0087251  0.0023828  -3.662 0.000333 ***
## Pclass      -0.0353452  0.1099506  -0.321 0.748249
## SibSp        0.0572773  0.0575666   0.995 0.321147
## Parch       -0.0661380  0.0510955  -1.294 0.197263
## Fare         0.0008168  0.0005499   1.485 0.139307
## DeckB        0.0414517  0.1522697   0.272 0.785775
## DeckC       -0.1315700  0.1504536  -0.874 0.383071
## DeckD        0.0925135  0.1551713   0.596 0.551824
## DeckE        0.0947487  0.1582973   0.599 0.550261
## DeckF       -0.0997679  0.2388778  -0.418 0.676722
## DeckG       -0.2019162  0.3453193  -0.585 0.559500
## DeckT       -0.5806090  0.4694428  -1.237 0.217846
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4509 on 172 degrees of freedom
##   (706 observations deleted due to missingness)
## Multiple R-squared:  0.1374, Adjusted R-squared:  0.07726
## F-statistic: 2.284 on 12 and 172 DF,  p-value: 0.01021
```

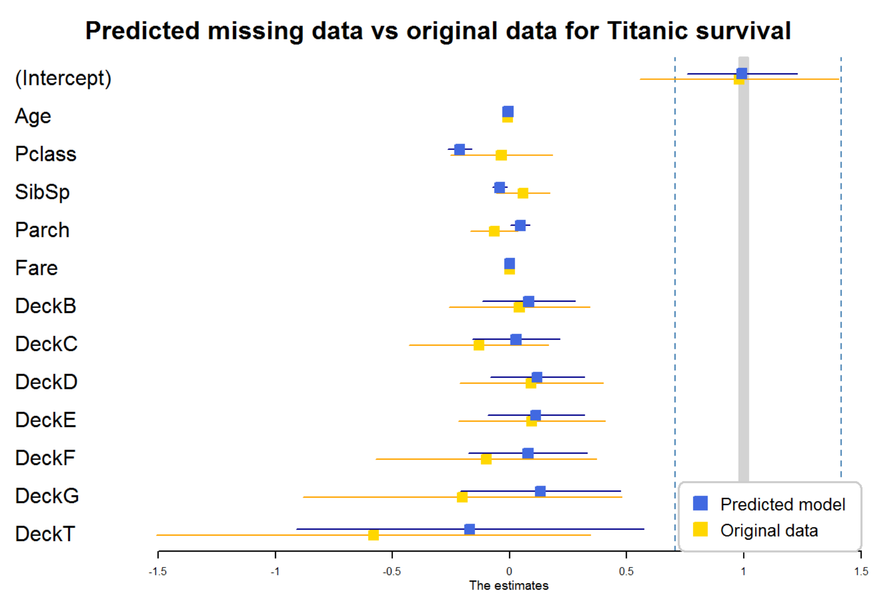
```
confint.lm(modelFit)
```

```
##                     2.5 %       97.5 %
## (Intercept)  0.5559768518  1.403198706
## Age         -0.0134284718 -0.004021705
## Pclass      -0.2523714838  0.181681172
## SibSp       -0.0563506951  0.170905243
## Parch       -0.1669929812  0.034717029
## Fare        -0.0002686887  0.001902229
## DeckB       -0.2591061167  0.342009550
## DeckC       -0.4285432867  0.165403189
## DeckD       -0.2137718106  0.398798754
## DeckE       -0.2177068166  0.407204255
## DeckF       -0.5712773998  0.371741630
## DeckG       -0.8835254718  0.479693108
## DeckT       -1.5072196504  0.346001688
```

Below I plotted the coefficients and 95% confidences interval from both the predicted and the original data in a forest plot. The results are very comparable, with predicted values showing tigher confidence intervals.

```
library(forestplot)

#data for forest plot
test_data <- data.frame(coef1=c(0.9922096929, -0.0070602395,-0.2137837799,-0.0426289510,0.0450752908,0.0007891985,0.08134933
09,0.0277050171,0.1172520198,0.1122658806,0.0784496074,0.1309519637,-0.1698740807),
                        coef2=c(0.9795878,-0.0087251,-0.0353452,0.0572773,-0.0661380,0.0008168,0.0414517,-0.1315700,0.092513
5,0.0947487,-0.0997679,-0.2019162,-0.580609),
                        low1=c(7.583006e-01,-9.859291e-03,-2.655729e-01,-7.459317e-02,3.668118e-03,1.243513e-05,-1.175626e-0
1,-1.580902e-01,-8.342528e-02,-9.480348e-02,-1.750453e-01,-2.101795e-01,-9.120255e-01),
                        low2=c(0.5559768518,-0.0134284718,-0.2523714838,-0.0563506951,-0.1669929812,-0.0002686887,-0.2591061
167,-0.4285432867,-0.2137718106,-0.2177068166,-0.5712773998,-0.8835254718,-1.5072196504),
                        high1=c(1.226118747,-0.004261188,-0.161994698,-0.010664727,0.086482464,0.001565962,0.280261302,0.213
500249,0.317929323,0.319335240,0.331944525,0.472083435,0.572277320),
                        high2=c(1.403198706,-0.004021705,0.181681172,0.170905243,0.034717029,0.001902229,0.342009550,0.16540
3189,0.398798754,0.407204255,0.371741630,0.479693108,0.346001688))
col_no <- grep("coef", colnames(test_data))
row_names <- list(
  list("(Intercept)", "Age", "Pclass", "SibSp","Parch","Fare","DeckB","DeckC","DeckD","DeckE","DeckF","DeckG","DeckT")
)
coef <- with(test_data, cbind(coef1, coef2))
low <- with(test_data, cbind(low1, low2))
high <- with(test_data, cbind(high1, high2))
forestplot(row_names, coef, low, high,
           title="Predicted missing data vs original data for Titanic survival",
           zero = c(0.98, 1.02),
           grid = structure(c(2^-.5, 2^.5), gp = gpar(col = "steelblue", lty=2)),
           boxsize=0.25,
           col=fpColors(box=c("royalblue", "gold"),
                        line=c("darkblue", "orange"),
                        summary=c("darkblue", "red")),
           xlab="The estimates",
           new_page = TRUE,
           legend=c("Predicted model", "Original data"),
           legend_args = fpLegend(pos = list("bottomright"),
                                  r = unit(.1, "snpc"),
                                  gp = gpar(col="#CCCCCC", lwd=1.5)))
```



Thats is the end of the tutorial, now we can use the complete dataset for predicting survival for Titanic passengers.

Reference

Roderick J. A. Little. (1988). A Test of Missing Completely at Random for Multivariate Data with Missing Values. Journal of the American Statistical Association, 83(404), 1198-1202. doi:10.2307/2290157 (doi:10.2307/2290157)

RUBIN, D.B. (1978). Multiple imputations in sample surveys - a phenomenological Bayesian approach to nonresponse. Proceedings of the Survey Research Methods Section of the American Statistical Association, 20- 34. Also in Imputation and Editin E of Faulty or Missin E Survey Data, U.S. Dept. of Commerce, 1-23.

Back to top