EDA Homework 2

Elliot Williams

Working with Missing Data

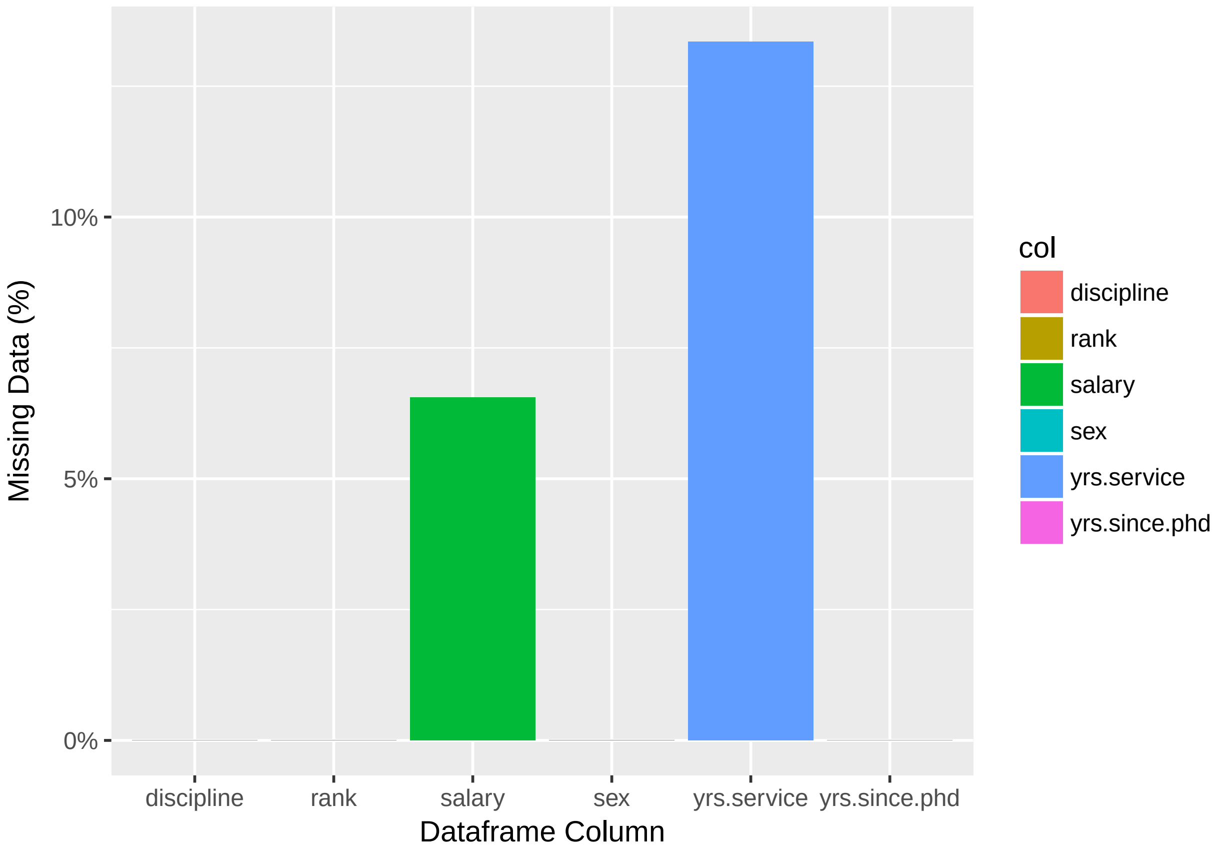
Load the Data

```
library(dplyr)
load("prof_salary.Rdata")
head(prof_salary)
```

```
##         rank discipline yrs.since.phd yrs.service  sex salary
## 1       Prof          B            19          18 Male 139750
## 2       Prof          B            20          16 Male 173200
## 3   AsstProf          B             4           3 Male  79750
## 4       Prof          B            45          NA Male 115000
## 5       Prof          B            40          NA Male     NA
## 6  AssocProf          B             6           6 Male  97000
```

What percentage of the data is missing for each variable?

```
library(ggplot2)

missing_pct <- colMeans(is.na(prof_salary))
df <- data.frame(missing_pct, stringsAsFactors=FALSE)
df$col <- rownames(df)

ggplot(df, aes(x=col, y=missing_pct, fill = col)) + geom_bar(stat="identity") +
  scale_y_continuous(labels=scales::percent) + labs(
    x = "Dataframe Column",
    y = "Missing Data (%)"
  )
```



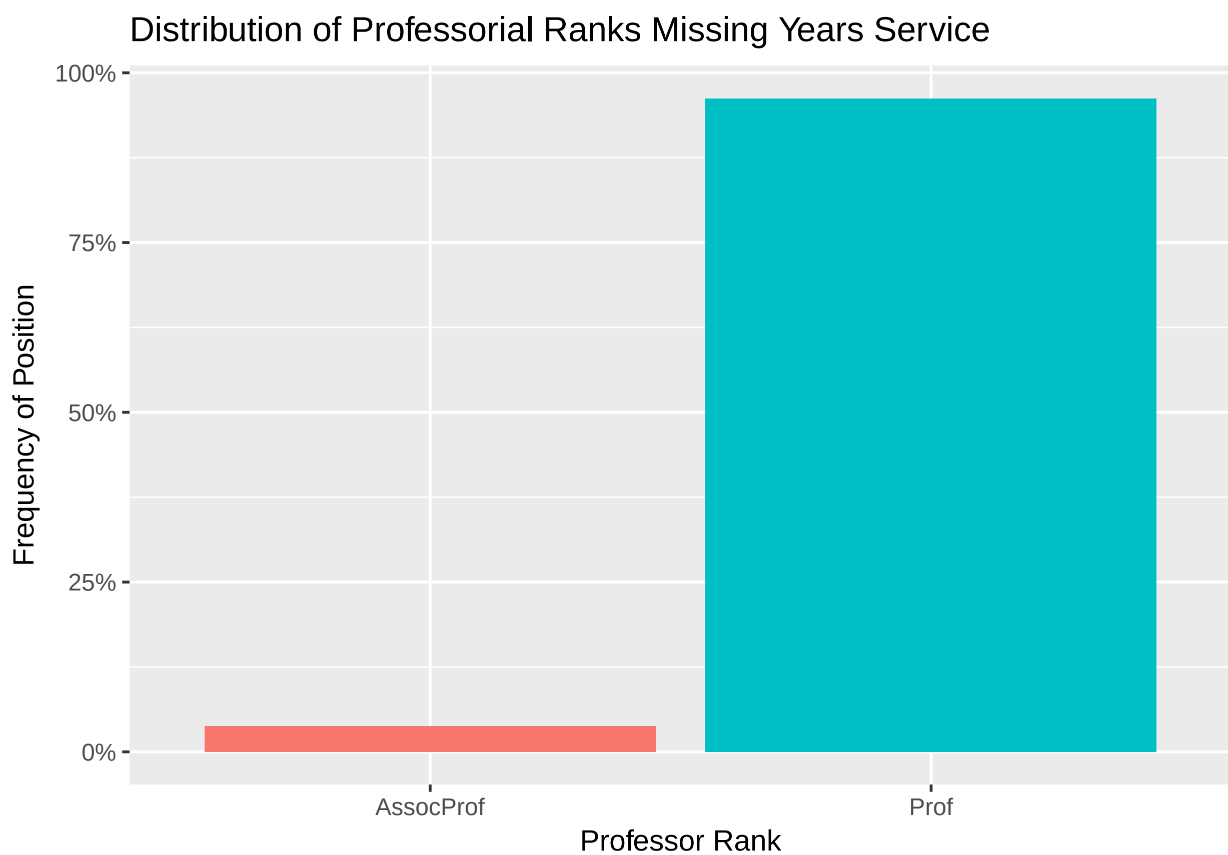
What are the patterns of missing data?

As we can see, full professors are by far the most likely to not fill out the years of service and salary fields.

```
library(gridExtra)

df2 <- prof_salary %>%
  filter(is.na(salary))

df3 <- prof_salary %>%
  filter(is.na(yrs.service))

plotRankDist <- function(df, label) {
  ggplot(df, aes(x=rank, fill=rank)) +
  geom_bar(aes(y = (..count..)/sum(..count..)))  +
  theme(legend.position = "none") +
  labs(x="Professor Rank", y="Frequency of Position",
       title=label) +
  scale_y_continuous(label=scales::percent)
}

plotRankDist(prof_salary, "Full Distribution of Professorial Ranks")
```


```
plotRankDist(df2, "Distribution of Professorial Ranks Missing Salary")
```


```
plotRankDist(df3, "Distribution of Professorial Ranks Missing Years Service")
```



Additionally, we can see that all of the professors who didn't fill out both of those fields were male.

```
df_missing <- prof_salary %>%
  group_by(sex) %>%
  summarize(`Missing Salary Percentage` = mean(is.na(salary)),
            `Missing Years Service Percentage` = mean(is.na(yrs.service)))

df_missing
```

```
## # A tibble: 2 x 3
##   sex    `Missing Salary Percentage` `Missing Years Service Percentage`
##   <fct>                        <dbl>                              <dbl>
## 1 Female                      0                                  0
## 2 Male                        0.0726                             0.148
```

And, we can see that the professors who don't fill out the salary and years of service data usually have a significantly higher amount time since their PhD than professors in the dataset overall.

```
mean(prof_salary$yrs.since.phd)
```

```
## [1] 22.31486
```

```
mean(df2$yrs.since.phd)
```

```
## [1] 39.92308
```

```
mean(df3$yrs.since.phd)
```

```
## [1] 39.43396
```

Linear Regression

```r
# For Listwise Deletion
prof_salary.listwise_del <- prof_salary %>%
  filter(is.na(salary) != TRUE, is.na(yrs.service) != TRUE)
model.listwise_del <- lm(salary ~ ., prof_salary.listwise_del)
summary(model.listwise_del)
```

```
##
## Call:
## lm(formula = salary ~ ., data = prof_salary.listwise_del)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66155 -11855  -1098   9693  82357
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   65953.6     4047.0  16.297  < 2e-16 ***
## rankAssocProf 13245.8     3629.9   3.649 0.000305 ***
## rankProf      44564.6     3845.4  11.589  < 2e-16 ***
## disciplineB   16525.9     2171.9   7.609 2.79e-13 ***
## yrs.since.phd   312.3      246.1   1.269 0.205320
## yrs.service    -257.0      217.2  -1.183 0.237574
## sexMale        3866.7     3326.3   1.162 0.245870
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19360 on 337 degrees of freedom
## Multiple R-squared:  0.5393, Adjusted R-squared:  0.5311
## F-statistic: 65.75 on 6 and 337 DF,  p-value: < 2.2e-16
```

```r
# For kNN imputation

library(VIM)

prof_salary.knn <- kNN(prof_salary, imp_var = F)
model.knn <- lm(salary ~ ., prof_salary.knn)
summary(model.knn)
```

```
##
## Call:
## lm(formula = salary ~ ., data = prof_salary.knn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -64455 -13064  -1258  11290  99482
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   68002.4     4271.0  15.922  < 2e-16 ***
## rankAssocProf 14395.1     3866.3   3.723 0.000226 ***
## rankProf      46033.6     3937.4  11.691  < 2e-16 ***
## disciplineB   13949.1     2178.1   6.404 4.35e-10 ***
```

```
## yrs.since.phd    125.7      232.5   0.541 0.588932
## yrs.service     -131.8      217.4  -0.607 0.544490
## sexMale         4178.2     3600.1   1.161 0.246520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21020 on 390 degrees of freedom
## Multiple R-squared:  0.4697, Adjusted R-squared:  0.4616
## F-statistic: 57.58 on 6 and 390 DF,  p-value: < 2.2e-16
```

```r
# For missForest imputation

library(missForest)

prof_salary.missForest <- missForest(prof_salary)$ximp
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
##   missForest iteration 4 in progress...done!
##   missForest iteration 5 in progress...done!
```

```r
model.missForest <- lm(salary ~ ., prof_salary.missForest)
summary(model.missForest)
```

```
##
## Call:
## lm(formula = salary ~ ., data = prof_salary.missForest)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -63108 -12655  -1604   9342  99707
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    68254.5     4173.7  16.354  < 2e-16 ***
## rankAssocProf  14529.4     3788.6   3.835 0.000146 ***
## rankProf       47293.3     3855.6  12.266  < 2e-16 ***
## disciplineB    14614.2     2133.3   6.850 2.88e-11 ***
## yrs.since.phd     39.2      227.2   0.173 0.863112
## yrs.service     -114.3      222.7  -0.513 0.608036
## sexMale         3844.6     3527.7   1.090 0.276463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20600 on 390 degrees of freedom
## Multiple R-squared:  0.4816, Adjusted R-squared:  0.4736
## F-statistic: 60.39 on 6 and 390 DF,  p-value: < 2.2e-16
```

So, we can see that the model that performs best (according to r-squared value being closest to the actual model) is the model that uses kNN imputation.

I chose to use r-squared as a metric for imputation selection because the statistically significant variables for each of the models were the same.

Part II: Basic Linear Algebra

Question 10: A is a 2x2 matrix. C is a 3x2 matrix. X is a 2x1 vector.

Question 11+:

```
a <- matrix(c(2,2,1,2), ncol=2, nrow=2)
b <- matrix(c(4,0,0,8), ncol=2, nrow=2)
c <- matrix(c(2, 3,4,1,1,1), ncol=2, nrow=3)
d <- matrix(c(1, -1, -1/2, 1), ncol=2, nrow=2)
x <- matrix(c(1,2), ncol=1, nrow=2)

a + b
```

```
##      [,1] [,2]
## [1,]    6    1
## [2,]    2   10
```

```
b + a
```

```
##      [,1] [,2]
## [1,]    6    1
## [2,]    2   10
```

```
a %*% x
```

```
##      [,1]
## [1,]    4
## [2,]    6
```

```
# b %*% c --> dimension mismatch, so impossible
c %*% b
```

```
##      [,1] [,2]
## [1,]    8    8
## [2,]   12    8
## [3,]   16    8
```

```
# inverse of a
solve(a) #so yes, d is the inverse of a
```

```
##      [,1] [,2]
## [1,]    1 -0.5
## [2,]   -1  1.0
```

```
# transpose of c
t(c)
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    4
## [2,]    1    1    1
```

```
# Recall that asymmetric matrix is equal to its transpose
b == t(b) # --> so, b is symmetric
```

```
##      [,1] [,2]
## [1,] TRUE TRUE
## [2,] TRUE TRUE
```

```
# Principal Diagonal of B
diag(b)
```

```
## [1] 4 8
```

```r
# Trace of B
sum(diag(b))
```

```
## [1] 12
```

```r
# If A*E = B, A^-1 * A * E = A^-1 * B = E
# the `solve` function does this for us
e <- solve(a, b)
e
```

```
##      [,1] [,2]
## [1,]    4   -4
## [2,]   -4    8
```