Calorie Burning Report

Methods and procedure

Data exploration

First let's load the data and look at the summary.

```
requirements = c("nlme", "effects", "pastecs", "lattice", "psych", "ggplot2", "GGally", "mice", "VIM",
if (length(setdiff(requirements, rownames(installed.packages()))) > 0) {
  invisible(install.packages(setdiff(requirements, rownames(installed.packages()))))
}
for (i in seq(1, length(requirements))) {
  invisible(library(requirements[i], character.only=T))
}
```

```
## Loading required package: carData
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
## Loading required package: boot
```

```
##
## Attaching package: 'lattice'
```

```
## The following object is masked from 'package:boot':
##
##     melanoma
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:boot':
##
##     logit
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## Loading required package: data.table
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:pastecs':
##
##     first, last
```

```
## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
```

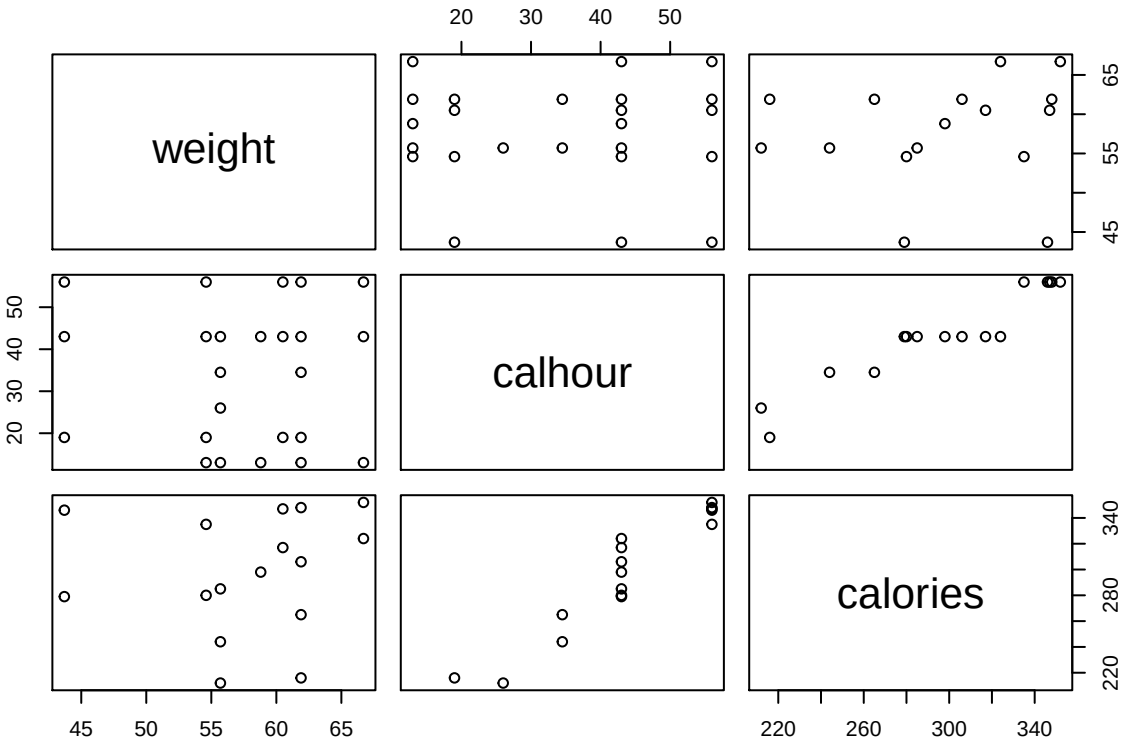
```
##
##           Please use the package to use the new (and old) GUI.

## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep

## Loading required package: Matrix

##
## Attaching package: 'lme4'

## The following object is masked from 'package:nlme':
##
##     lmList
```

```
options(digits=4)
muscledata = read.table("muscle-incomplete.txt", header=T, na.strings = "NA")
summary(muscledata)
```

```
##      weight        calhour        calories
##  Min.   :43.7   Min.   :13.0   Min.   :212
##  1st Qu.:54.6   1st Qu.:19.0   1st Qu.:276
##  Median :58.8   Median :38.8   Median :302
##  Mean   :57.5   Mean   :34.0   Mean   :297
##  3rd Qu.:61.9   3rd Qu.:43.0   3rd Qu.:338
##  Max.   :66.7   Max.   :56.0   Max.   :352
##                                NA's   :8
```

Here are some descriptive statistics.

```
plot(muscledata)
```



Let's check correlation between the weight of an individual and the intensity of their workout, as defined by the calories burnt per hour, using the Pearson correlation coefficient:

```
cor.test(muscledata$weight, muscledata$calhour, alternative="two.sided", method="pearson")
```

```
##
##  Pearson's product-moment correlation
##
## data:  muscledata$weight and muscledata$calhour
## t = -0.13, df = 22, p-value = 0.9
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4257  0.3807
## sample estimates:
##      cor
## -0.02687
```

The P value is way, way more than 0.05. We accept the null hypothesis, there seems to be no significant correlation.

Let's try plotting a simple linear model to this relationship.

```
weightCalhr = lm(weight~calhour, data=muscledata)
summary(weightCalhr)
```

```
##
## Call:
## lm(formula = weight ~ calhour, data = muscledata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14.00  -2.74   1.19   4.24   9.40
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  57.9102     3.2298   17.93  1.3e-14 ***
## calhour      -0.0108     0.0859   -0.13      0.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.74 on 22 degrees of freedom
## Multiple R-squared:  0.000722,   Adjusted R-squared:  -0.0447
## F-statistic: 0.0159 on 1 and 22 DF,  p-value: 0.901
```

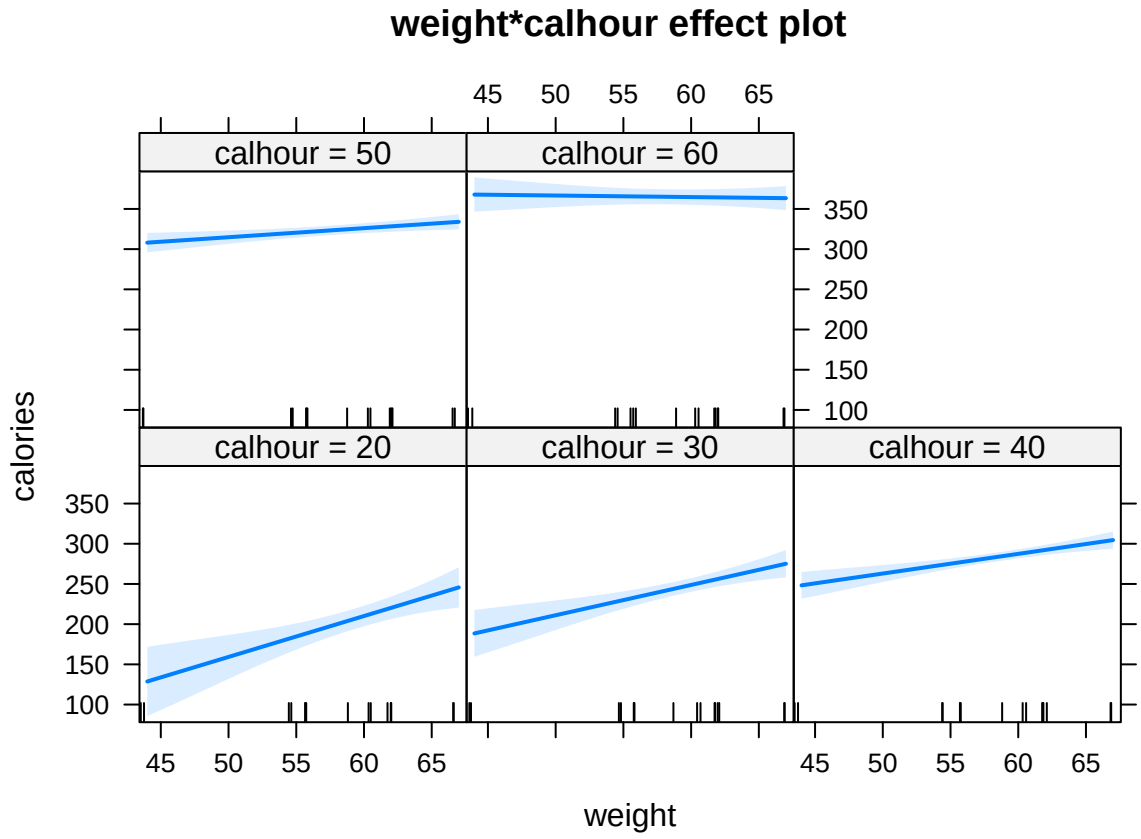
The linear model confirms our suspicion that there is no relationship here.

Let's try another one:

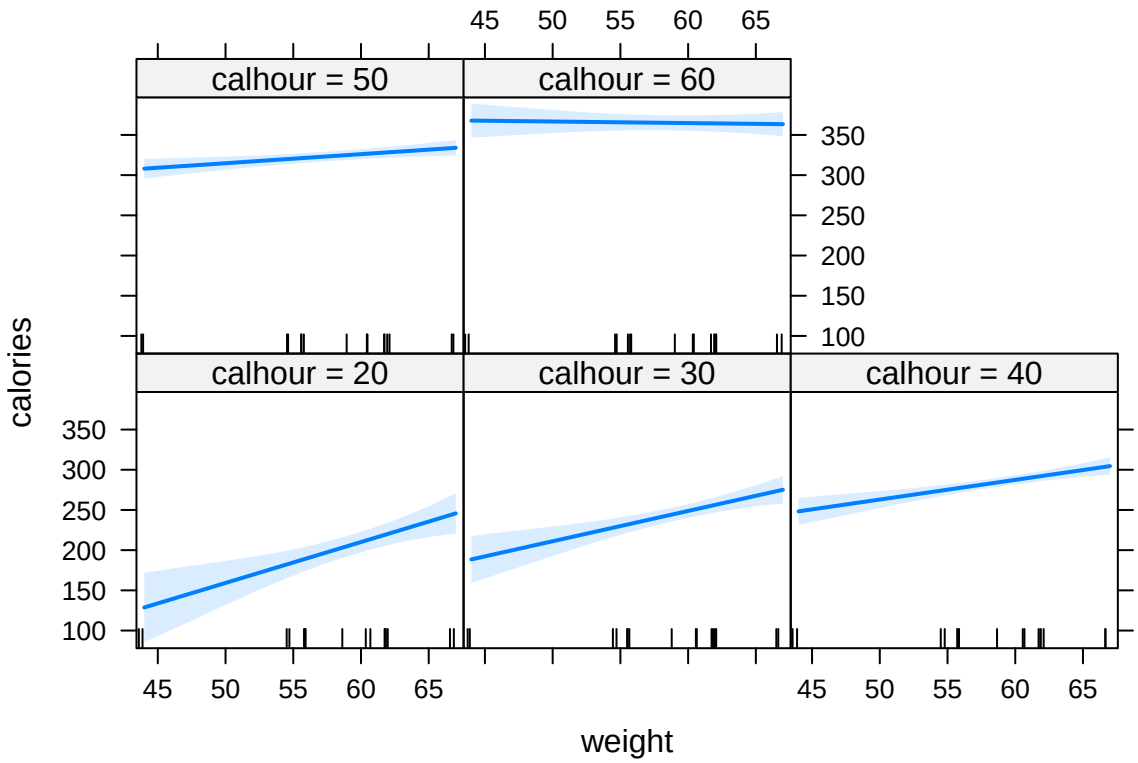
```
muscledata_edit = na.omit(muscledata)
muscledata.lm = lm(calories~weight+calhour+weight*calhour, data=muscledata_edit)
muscledata.lm.summary = summary(muscledata.lm)
muscledata.lm.summary
```

```
##
## Call:
## lm(formula = calories ~ weight + calhour + weight * calhour,
##     data = muscledata_edit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -12.48  -5.70  -1.04   2.39  16.95
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -330.884    124.674   -2.65  0.02102 *
## weight            7.728      2.106    3.67  0.00321 **
## calhour          11.787      2.548    4.63  0.00058 ***
## weight:calhour   -0.132      0.043   -3.07  0.00977 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.12 on 12 degrees of freedom
## Multiple R-squared:  0.968,  Adjusted R-squared:  0.96
## F-statistic:  123 on 3 and 12 DF,  p-value: 2.89e-09
```

```
plot(allEffects(muscledata.lm))
```




```
plot(allEffects(muscledata.lm))
```

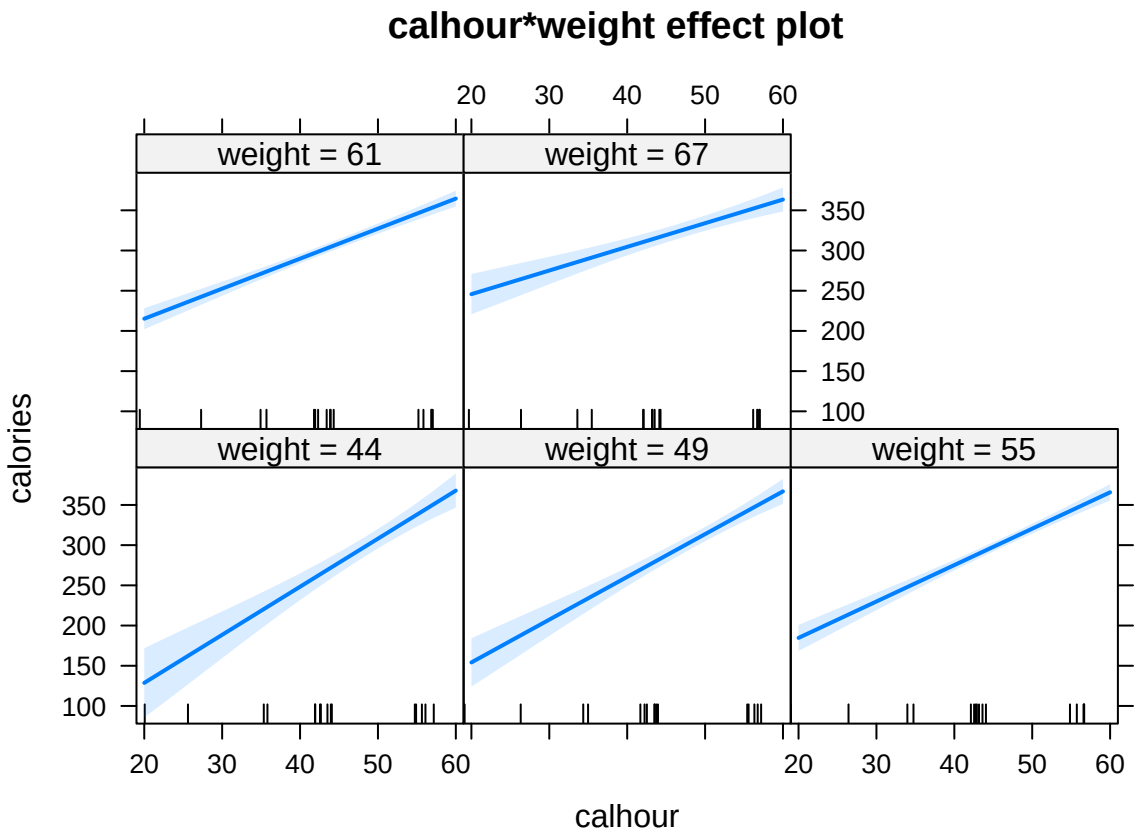


There seems to be a relationship between calhour and missing data. Clearly, there is a Missing at Random (MAR) mechanism involved - P(Missing) depends only on the observed values of calhour.

```
muscledata.lm.2 = lm(calories~calhour+weight+calhour*weight, data=muscledata_edit)
summary(muscledata.lm.2)
```

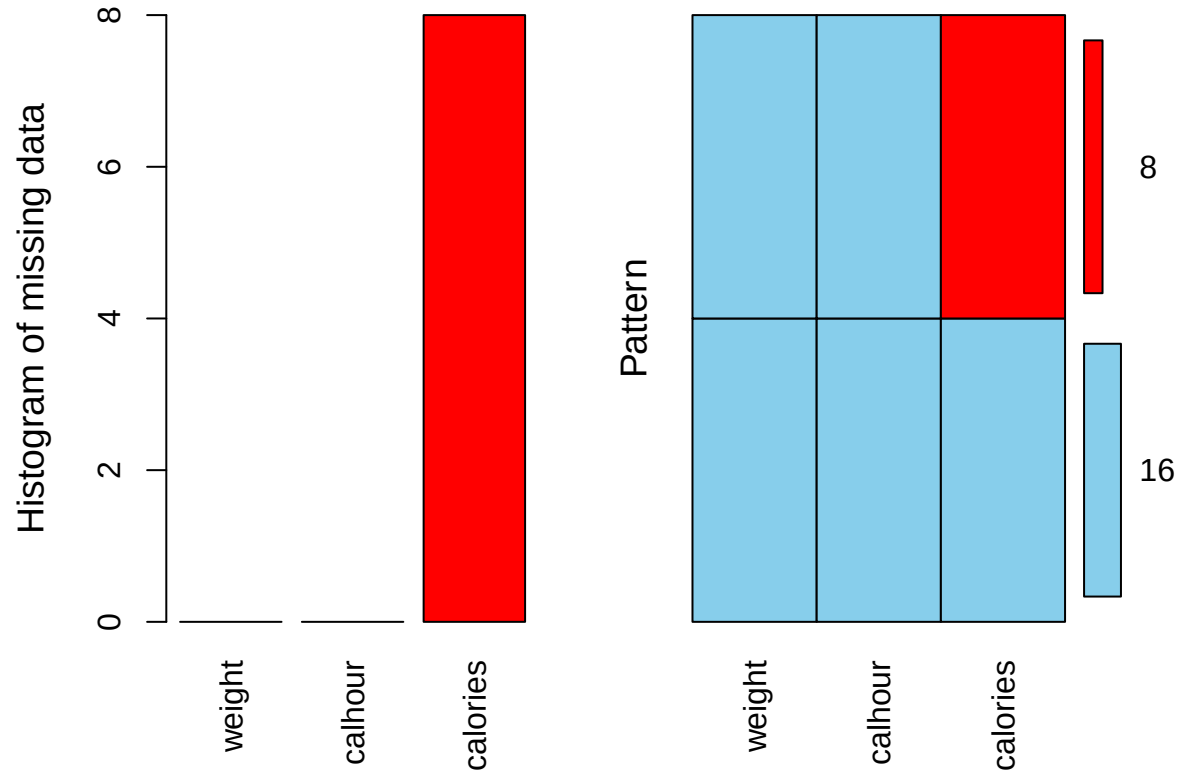
```
##
## Call:
## lm(formula = calories ~ calhour + weight + calhour * weight,
##     data = muscledata_edit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -12.48  -5.70  -1.04   2.39  16.95
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -330.884    124.674   -2.65  0.02102 *
## calhour          11.787      2.548    4.63  0.00058 ***
## weight            7.728      2.106    3.67  0.00321 **
## calhour:weight   -0.132      0.043   -3.07  0.00977 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.12 on 12 degrees of freedom
## Multiple R-squared:  0.968,  Adjusted R-squared:  0.96
## F-statistic:  123 on 3 and 12 DF,  p-value: 2.89e-09
```

```
plot(allEffects(muscledata.lm.2))
```

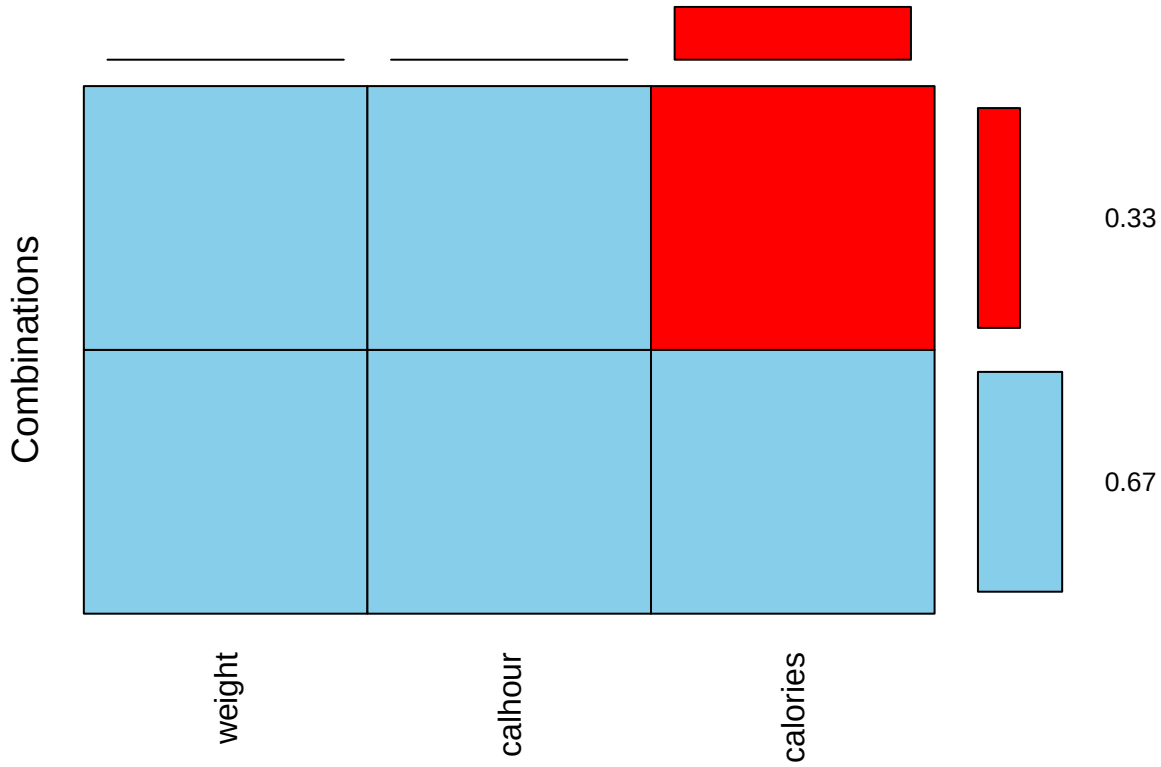


Missing data exploration

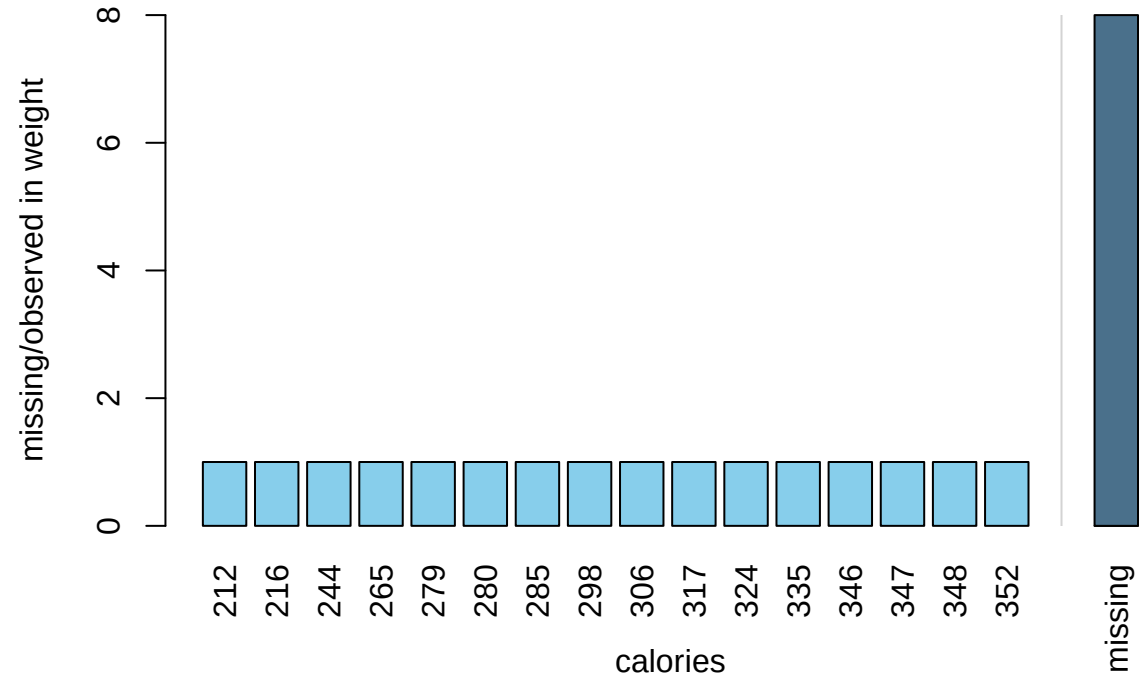
```
aggr(muscledata, numbers = TRUE, prop = FALSE, ylab = c("Histogram of missing data", "Pattern"))
```



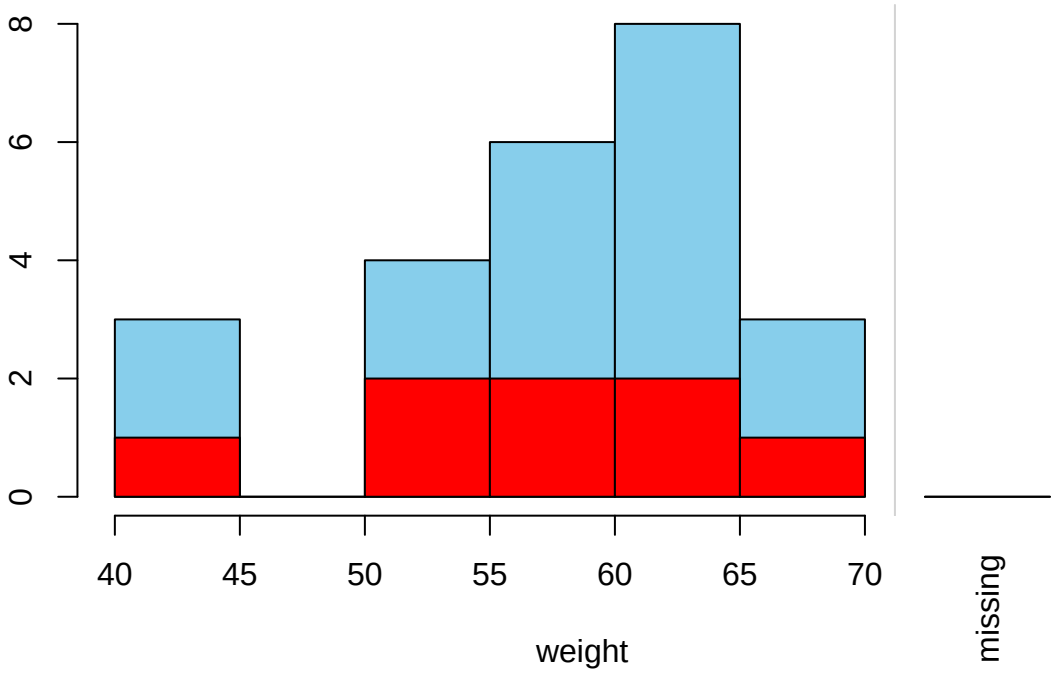

```
aggr(muscledata, combined=TRUE, numbers = TRUE, prop = TRUE, cex.numbers=0.87, varheight = FALSE)
```




```
barMiss(muscledata[,c("calories","weight")])
```




```
histMiss(muscledata)
```



Complete case analysis

Multiple imputation analysis

```
invisible(capture.output(muscledata.imp <- mice(muscledata, meth = c("", "", "pmm"), m=100)))
invisible(muscledata.fit <- with(data=muscledata.imp, exp=glm(calories~weight+calhour+weight*calhour)))
MI.matrix<-matrix(0,100,4)
```

```
for(k in 1:100){
  invisible(MI.matrix[k,]<-coefficients(muscledata.fit$analyses[[k]]))
  }
MI.results=data.frame(Intercept=MI.matrix[,1], weight=MI.matrix[,2],calhour=MI.matrix[,3], interaction=M
head(MI.results[1:10,])
```

```
##   Intercept    weight calhour interaction
## 1     62.73  2.505985  3.9151    -0.02922
## 2    434.05 -4.277493 -3.4074     0.10508
## 3    302.41 -1.765560 -0.8238     0.05531
## 4    234.21 -0.459291  0.5340     0.02927
## 5    341.09 -2.246851 -1.5632     0.06422
## 6    189.42 -0.006038  1.3029     0.02281
```

```
muscledata.est <- pool(muscledata.fit)
summary(muscledata.est)
```

```
##                      est        se       t    df Pr(>|t|)     lo 95
## (Intercept)    182.72156 174.50617 1.04708 8.090   0.3253 -218.9137
## weight           0.24289   2.99608 0.08107 8.091   0.9374   -6.6525
## calhour          1.56453   3.83792 0.40765 9.907   0.6922   -6.9978
## weight:calhour   0.01531   0.06587 0.23249 9.905   0.8209   -0.1316
##                    hi 95 nmis    fmi lambda
## (Intercept)    584.3568   NA 0.6278 0.5459
## weight           7.1383    0 0.6277 0.5458
## calhour         10.1269    0 0.5322 0.4465
## weight:calhour   0.1623   NA 0.5323 0.4465
```

IPW analysis

```
muscledata$r<-as.numeric(!is.na(muscledata$calories))
head(muscledata)
```

```
##   weight calhour calories r
## 1   43.7      19       NA 0
## 2   43.7      43      279 1
## 3   43.7      56      346 1
## 4   54.6      13       NA 0
## 5   54.6      19       NA 0
## 6   54.6      43      280 1
```

```
muscledata.ipw.glm<-glm(r ~ weight+calhour+weight*calhour, data=muscledata,family=binomial)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(muscledata.ipw.glm)
```

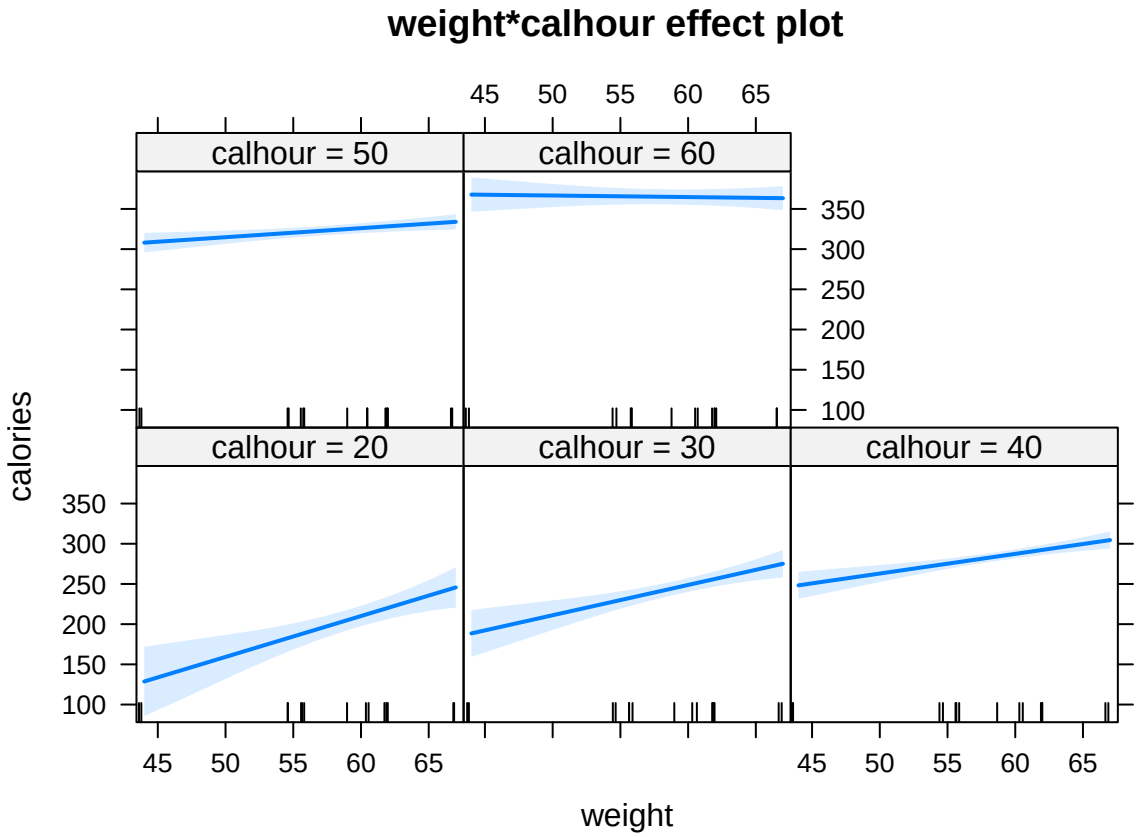
```
##
## Call:
## glm(formula = r ~ weight + calhour + weight * calhour, family = binomial,
##     data = muscledata)
##
## Deviance Residuals:
```

```
##       Min         1Q     Median         3Q        Max
## -1.01e-04  -2.00e-08   2.00e-08   2.00e-08   9.76e-05
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.79e+03   6.47e+05       0        1
## weight          2.09e+01   8.83e+03       0        1
## calhour         6.19e+00   1.37e+04       0        1
## weight:calhour  3.37e-01   2.67e+02       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3.0553e+01  on 23  degrees of freedom
## Residual deviance: 2.0732e-08  on 20  degrees of freedom
## AIC: 8
##
## Number of Fisher Scoring iterations: 25
```

```
muscledata$w<-1/fitted(muscledata.ipw.glm)
muscledata.results.ipw<- glm(calories~weight+calhour+weight*calhour, data=muscledata, weights=muscledata
summary(muscledata.results.ipw)
```

```
##
## Call:
## glm(formula = calories ~ weight + calhour + weight * calhour,
##     data = muscledata, weights = muscledata$w)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -12.48   -5.70   -1.04    2.39   16.95
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -330.884    124.674   -2.65  0.02102 *
## weight            7.728      2.106    3.67  0.00321 **
## calhour          11.787      2.548    4.63  0.00058 ***
## weight:calhour   -0.132      0.043   -3.07  0.00977 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 83.14)
##
##     Null deviance: 31557.75  on 15  degrees of freedom
## Residual deviance:   997.67  on 12  degrees of freedom
##   (8 observations deleted due to missingness)
## AIC: 121.5
##
## Number of Fisher Scoring iterations: 2
```

```
plot(allEffects(muscledata.results.ipw))
```



Discussion

Conclusion