Data Science for Actuaries (ACT6100)

Arthur Charpentier

Non-Supervisé # 4 (k plus proches voisins & imputation)

automne 2Q20

https://github.com/freakonometrics/ACT6100/

Missing Values & *k*-NN

There are 3 major types of missingness to be concerned about:

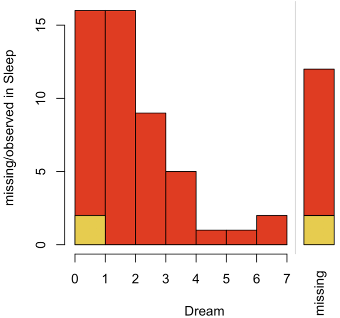
- Missing Completely at Random (MCAR) - the probability of missingness in a variable is the same for all units. Like randomly poking holes in a data set.
- Missing at Random (MAR) - the probability of missingness in a variable depends only on available information (in other predictors).
- Missing Not at Random (MNAR) - the probability of missingness depends on information that has not been recorded and this information also predicts the missing values.

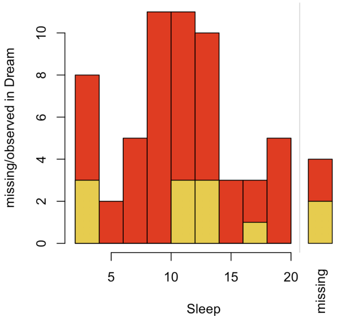
Missing Values & k-NN

via Allison & Chichetti (1976)

```
> library(VIM)
> x = sleep[, c("Dream", "
   Sleep")]
> summary(x)
      Dream            Sleep
 Min.   :0.000    Min.   : 2.60
 1st Qu.:0.900    1st Qu.: 8.05
 Median :1.800    Median :10.45
 Mean   :1.972    Mean   :10.53
 3rd Qu.:2.550    3rd Qu.:13.20
 Max.   :6.600    Max.   :19.90
 NA's   :12       NA's   :4
```

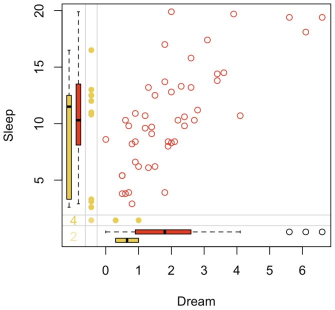
```
> histMiss(x)
> histMiss(x[,2:1])
```

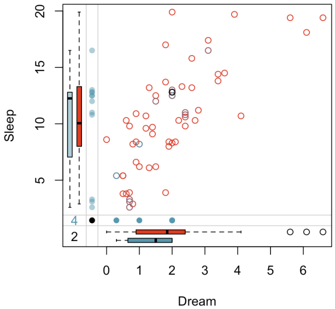




Missing Values & *k*-NN

```
> marginplot(x)
> x_imputed = kNN(x)
> marginplot(x_imputed,
    delimiter = "_imp")
> i=apply(x,1,function(x) sum(
    is.na(x)))
> cor(x[i==0,])
              Dream     Sleep
Dream  1.000000  0.727087
Sleep  0.727087  1.000000
> cor(x_imputed[,1:2])
               Dream      Sleep
Dream  1.0000000  0.7396222
Sleep  0.7396222  1.0000000
```





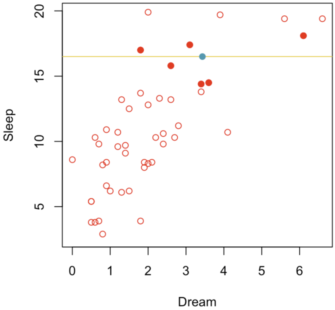
Missing Values & k-NN

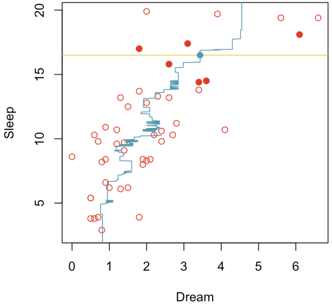
If $x_{i,2}$ is missing

$$\overline{x}_{i,2} = \frac{1}{5} \sum_{j \in V} x_{j,2}$$

$$V = \{j : \|x_{j,1} - x_{i,1}\| \leq k\}$$

```
> i = which(is.na(x[,2]))[1]
> xc=x[!is.na(x[,2]),]
> R=rank(abs(xc[,1]-x[i,1]),
    ties.method = "random")
> ic = which(R<=5)
> mean(xc[ic,2])
```





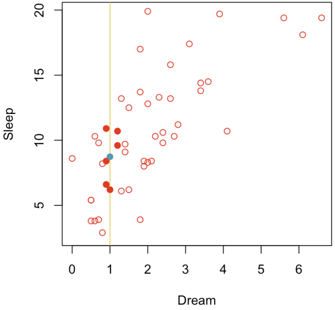
Missing Values & k-NN

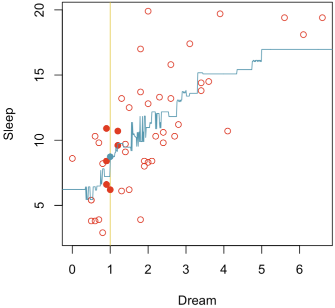
If $x_{i,1}$ is missing

$$\overline{x}_{i,1} = \frac{1}{5}\sum_{j\in V} x_{j,1}$$

$$V = \{j : \|x_{j,2} - x_{i,2}\| \leq k\}$$

```
> i = which(is.na(x[,1])[1]
> xc=x[!is.na(x[,1]),]
> R=rank(abs(xc[,2]-x[i,2]),
    ties.method = "random")
> ic = which(R<=5)
> mean(xc[ic,1])
```





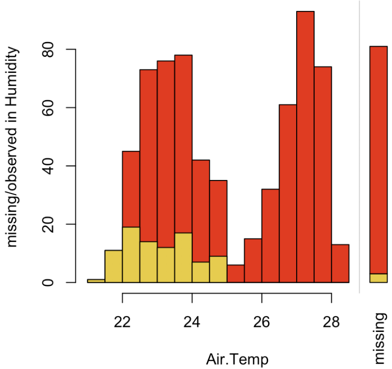
Missing Values & k-NN

```
> library(VIM)
> data(tao)
> y = tao[, c("Air.Temp", "Humidity")]
> summary(y)
    Air.Temp         Humidity
 Min.   :21.42   Min.   :71.60
 1st Qu.:23.26   1st Qu.:81.30
 Median :24.52   Median :85.20
 Mean   :25.03   Mean   :84.43
 3rd Qu.:27.08   3rd Qu.:88.10
 Max.   :28.50   Max.   :94.80
 NA's   :81      NA's   :93
```

Missing Values & k-NN

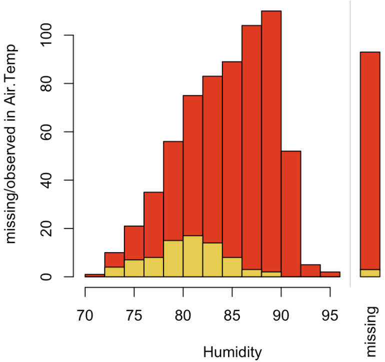
Missing humidity given the temperature

```
> y =tao[,c("Air.Temp", "Humidity")]
> histMiss(y)
```



Missing temperature given the humidity

```
> y = tao[,c("Humidity", "Air.Temp")]
> histMiss(y)
```



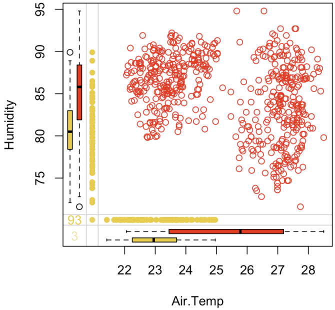
Missing Values & *k*-NN

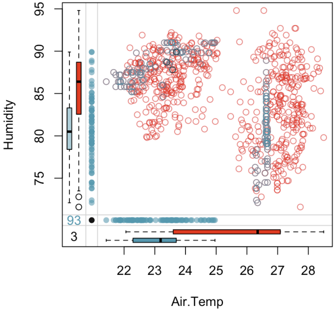
This package countains a *k*-Neareast Neighbors algorithm for imputation

```
> tao_kNN = kNN(tao, k = 5)
```

Imputation can be visualized using

```
vars = c("Air.Temp","Humidity","Air.Temp_imp","Humidity_imp")
marginplot(tao_kNN[,vars], delimiter="imp", alpha=0.6)
```





Missing Values & *k*-NN

This package countains a *k*-Neareast Neighbors algorithm for imputation

```
> tao_kNN = kNN(tao, k = 5)
```

Imputation can be visualized using

```
vars = c("Air.Temp","Humidity
    ","Air.Temp_imp","
    Humidity_imp")
marginplot(kNN(tao[vars[1:2]],
     k = 5), delimiter="imp",
    alpha=0.6)
```