

# Test des indicateurs HET

*JcB*

*09/11/2015*

## Contents

<b>1</b>	<b>Indicateurs Champagne Ardennes</b>	<b>1</b>
1.1	HET2 - Nombre de passages aux urgences (par jour) . . . . .	2
1.2	HET3: moyenne des durées de passage des patients hospitalisés à partir des urgences . . . . .	5
1.3	HET4: taux d'hospitalisation après passage aux urgences (nb d'hospitalisation / nb de passages)	6
1.4	HET5 - charge d'occupation à 15 heures / capacité d'accueil du service des urgences . . . . .	6
1.5	Synthèse . . . . .	7
1.6	Corrélation taux hospitalisation et nombre de passages, et durée de passage . . . . .	7
<b>2</b>	<b>Radar</b>	<b>10</b>
<b>3</b>	<b>Test de la fonction het.fr</b>	<b>19</b>

## 1 Indicateurs Champagne Ardennes

L'ORUCA a retenu 5 indicateurs:

- HET1: disponibilité en lit médecine + chir (source manuelle ou ROR)
- HET2: Le nombre de passages aux urgences à J-1
- HET3: moyenne des durées de passage des patients hospitalisés à partir des urgences J-1
- HET4: taux d'hospitalisation après passage aux urgences (nb d'hospitalisation / nb de passages) J-1
- HET5: charge d'occupation à 15 heures / capacité d'accueil du service des urgences (nb de patients pouvant être installés simultanément dans le service en salle d'examen et en zone d'attente organisée)

Résultats disponibles:

- synthèse régionale du radar de chaque établissement
- détail du radar avec tableau de données chiffrées pour chaque établissement
- détail de chaque indicateur pour chaque établissement avec courbe historique des 30 derniers jours (et moyenne lissée)

```
fichier <- "../..//DATA/data_test.Rda"
load(fichier) # dx

library(lubridate)
library(xts)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric
```

```
library(Rpu2)
```

```
## Loading required package: xtable
## Loading required package: openintro
## Please visit openintro.org for free statistics materials
##
## Attaching package: 'openintro'
##
## The following object is masked from 'package:datasets':
##
##      cars
##
## Loading required package: plotrix
```

```
source("duree_passage.R") # si console: source("Indicateurs/duree_passage.R")
```

```
# masquer cette ligne pour faire le calcul avec tous les établissements
dx <- dx[dx$FINESS == "Wis",]
```

```
# création d'un calendrier pour le période (nécessaire pour transformer en time serie xts)
x <- seq(min(as.Date(dx$ENTREE)), max(as.Date(dx$ENTREE)), 1)
```

## 1.1 HET2 - Nombre de passages aux urgences (par jour)

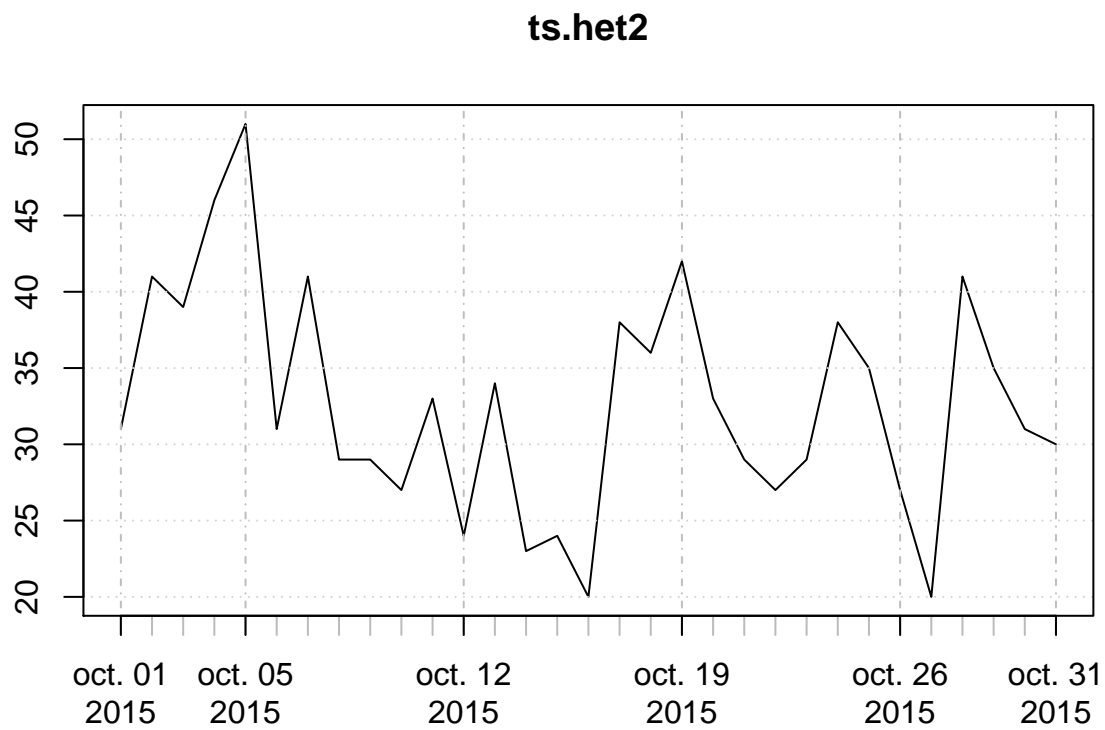
```
n.rpu.jour <- tapply(as.Date(dx$ENTREE), day(as.Date(dx$ENTREE)), length)
```

```
# transformation en time serie
x <- seq(min(as.Date(dx$ENTREE)), max(as.Date(dx$ENTREE)), 1)
ts.het2 <- xts(n.rpu.jour, order.by = x)
colnames(ts.het2) <- "HET2"
```

```
head(ts.het2)
```

```
##           HET2
## 2015-10-01  31
## 2015-10-02  41
## 2015-10-03  39
## 2015-10-04  46
## 2015-10-05  51
## 2015-10-06  31
```

```
plot(ts.het2)
```



```
# Répartition normale ?  
summary(n.rpu.jour)
```

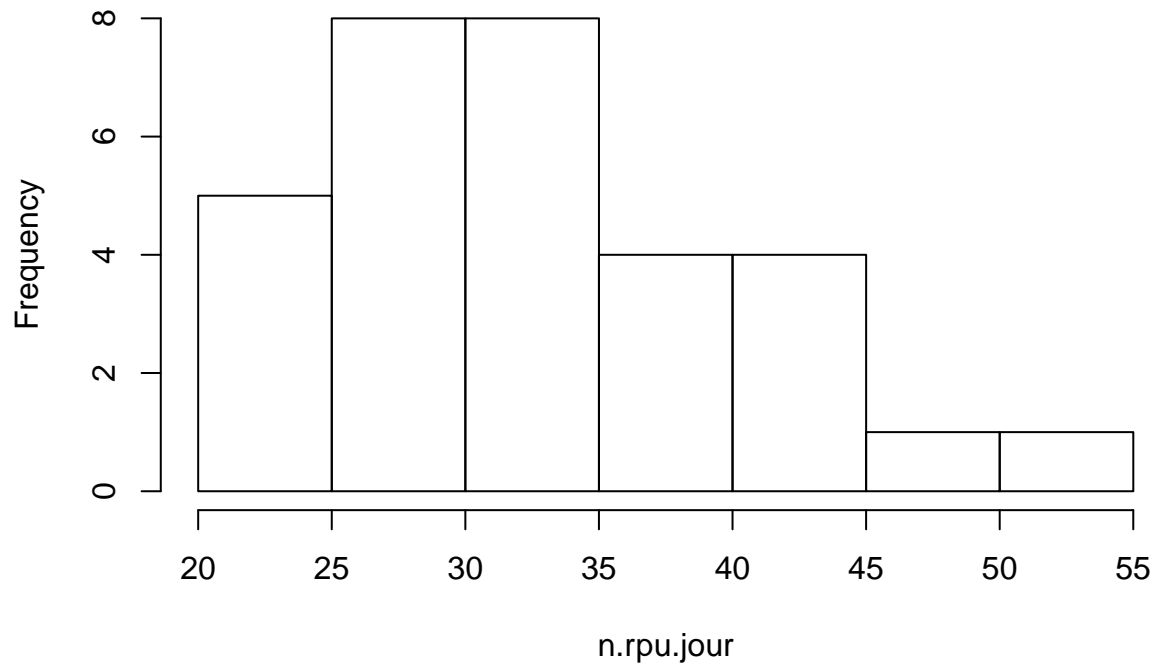
```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   
##  20.00  28.00   31.00   32.71  38.00   51.00
```

```
sd(n.rpu.jour)
```

```
## [1] 7.439505
```

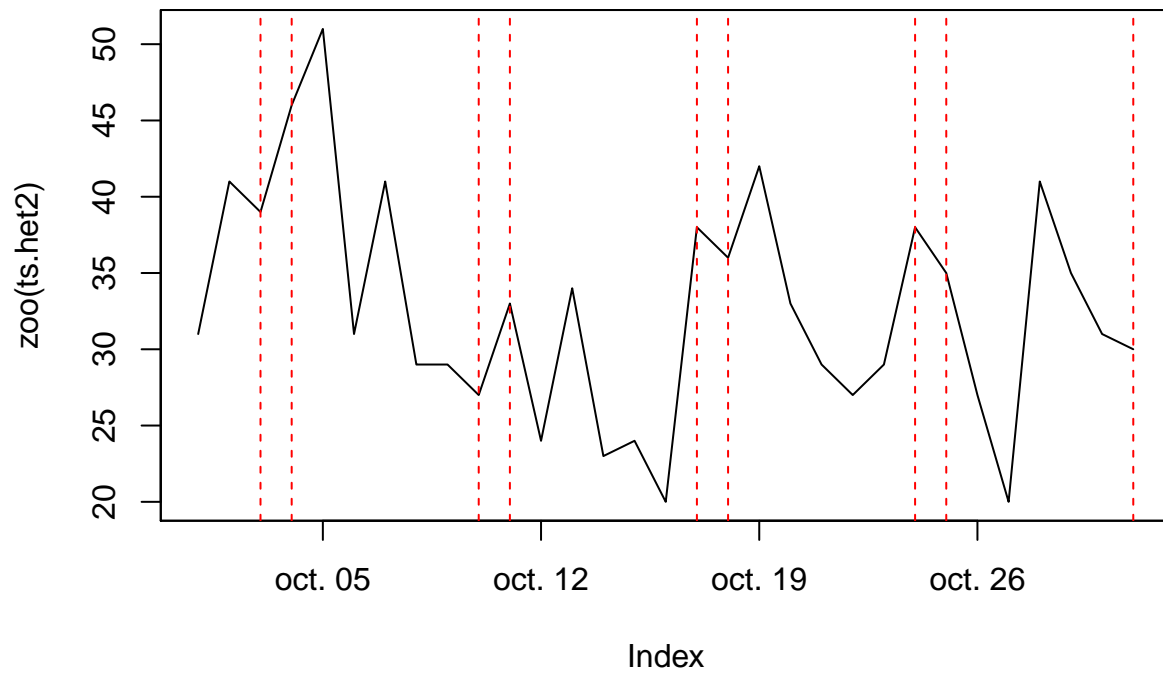
```
hist(n.rpu.jour)
```

## Histogram of n.rpu.jour



Graphe avec les WE: on utilise zoo car abline ne fonctionne pas avec xts ?

```
we <- x[wday(x) %in% c(1,7)]
plot(zoo(ts.het2))
abline(v = as.Date(we), lty = 2, col = "red")
```



## 1.2 HET3: moyenne des durées de passage des patients hospitalisés à partir des urgences

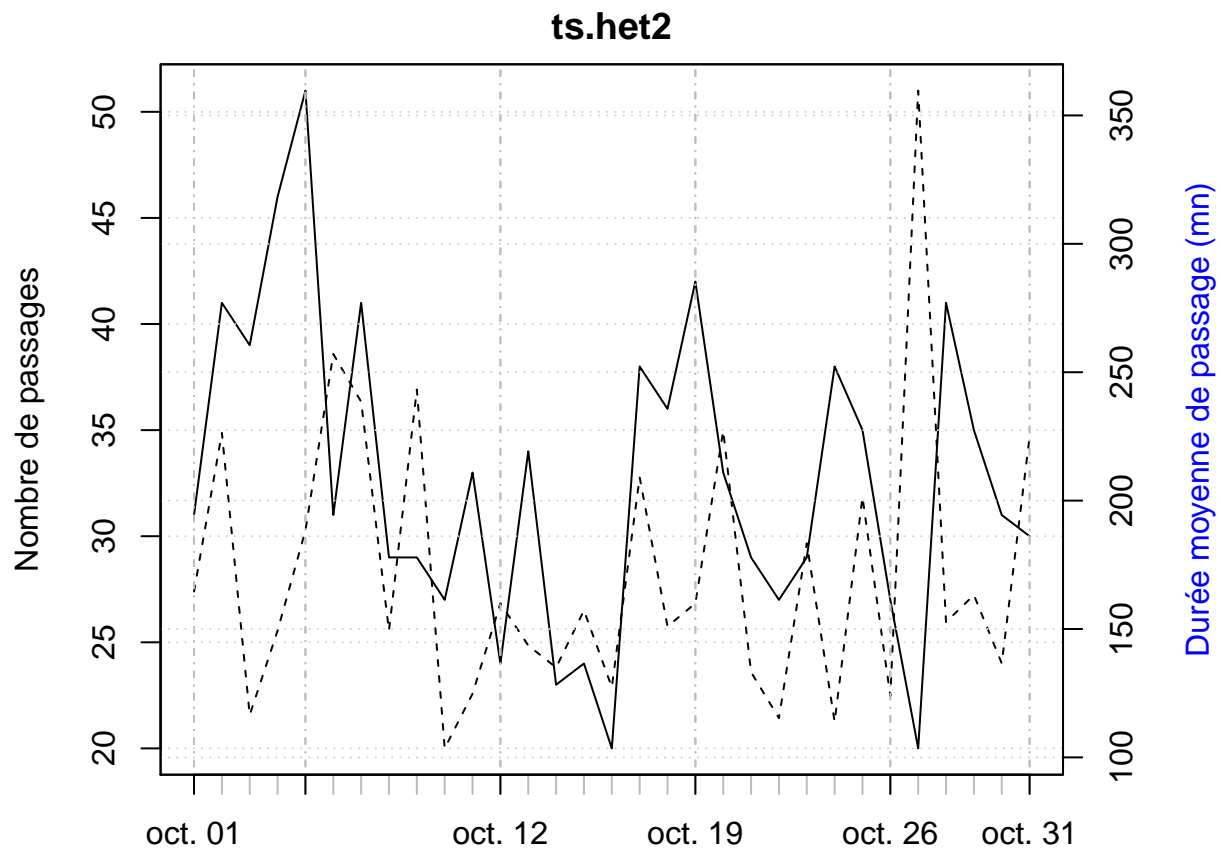
```
# sélectionne les enregistrements où le MODE_SORTIE correspond à une hospitalisation
hosp <- dx[!is.na(dx$MODE_SORTIE) & dx$MODE_SORTIE %in% c("Mutation", "Transfert"), ]

# durée de passage si hospitalisation
dp <- df.duree.pas(hosp, unit = "mins", mintime = 0, maxtime = 3)

# moyenne quotidienne
mean.dp <- tapply(dp$duree , day(as.Date(dp$ENTREE)), mean)

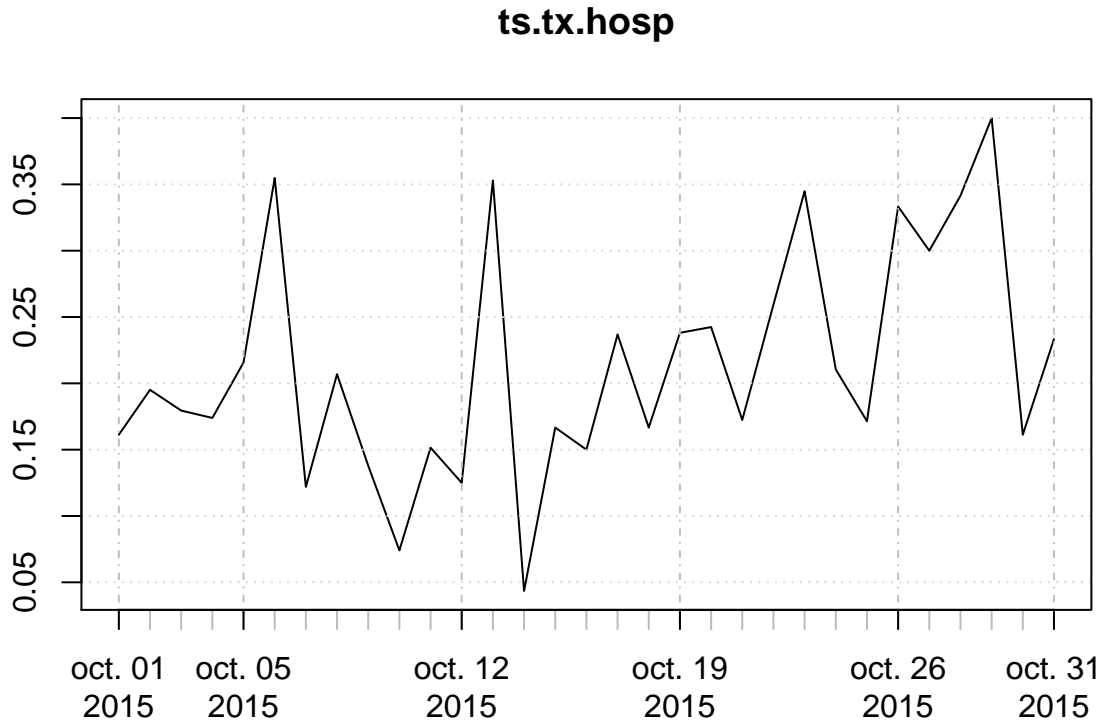
# transformation en time serie
ts.mean.dp <- xts(mean.dp, x)
colnames(ts.mean.dp) <- "HET3"

par(mar = c(2,4,2,5))
plot(ts.het2, ylab = "Nombre de passages")
par(new=TRUE)
plot(ts.mean.dp, xaxt="n",xlab="",ylab="", main = "", yaxt="n", lty = 2)
axis(4)
mtext("Durée moyenne de passage (mn)",side=4,line=3, col = "blue")
```



### 1.3 HET4: taux d'hospitalisation après passage aux urgences (nb d'hospitalisation / nb de passages)

```
n.hosp.jour <- tapply(as.Date(hosp$ENTREE), day(as.Date(hosp$ENTREE)), length)
tx.hosp <- n.hosp.jour / n.rpu.jour
ts.tx.hosp <- xts(tx.hosp, x)
colnames(ts.tx.hosp) <- "HET4"
plot(ts.tx.hosp)
```

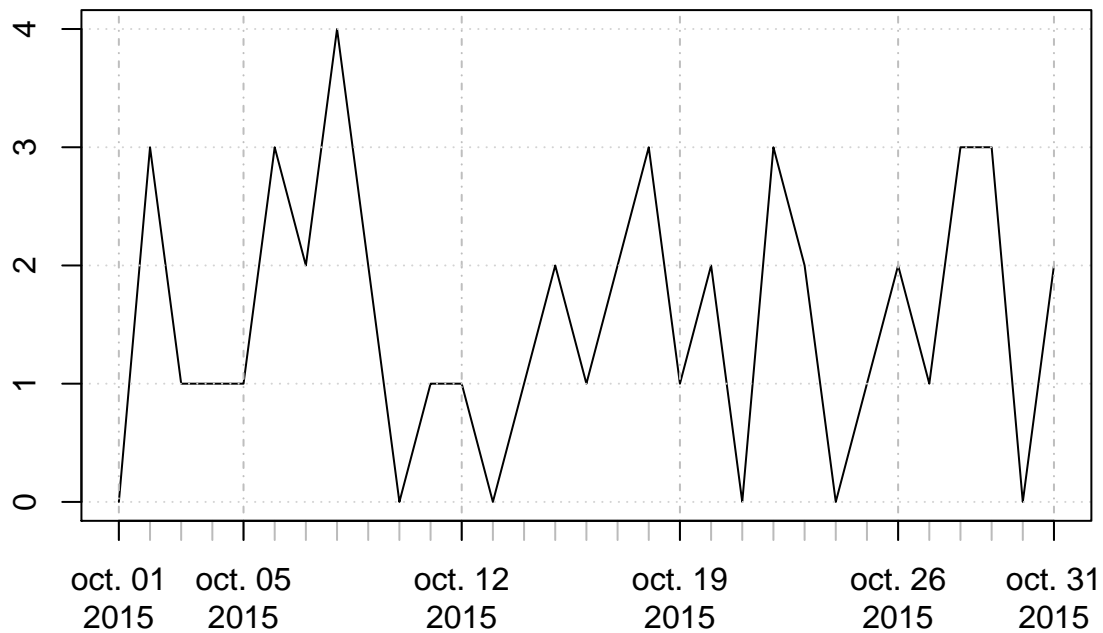


### 1.4 HET5 - charge d'occupation à 15 heures / capacité d'accueil du service des urgences

```
dp$present.a.15h <- is.present.at(dp)
# nombre moyen de patients présents à 15h tous les jours
n.p15 <- tapply(dp$present.a.15h, yday(as.Date(dp$ENTREE)), sum)

# Transformation en TS
ts.n.p15 <- xts(n.p15, x)
colnames(ts.n.p15) <- "HET5"
plot(ts.n.p15, main = "Nombre de patients présents au SU à 15 heures")
```

## Nombre de patients présents au SU à 15 heures



### 1.5 Synthèse

```
a <- cbind(ts.het2, ts.mean.dp, ts.tx.hosp, ts.n.p15)
head(a)
```

```
##           HET2      HET3      HET4 HET5
## 2015-10-01    31 164.6000 0.1612903    0
## 2015-10-02    41 226.3750 0.1951220    3
## 2015-10-03    39 116.5000 0.1794872    1
## 2015-10-04    46 149.3750 0.1739130    1
## 2015-10-05    51 188.3636 0.2156863    1
## 2015-10-06    31 257.1000 0.3548387    3
```

```
a[1, ]
```

```
##           HET2 HET3      HET4 HET5
## 2015-10-01    31 164.6 0.1612903    0
```

```
#radial.plot(a[1, ], labels=ion.names,rp.type="p",main="Diagramme indicateurs HET", grid.unit="%",radia
```

### 1.6 Corrélation taux hospitalisation et nombre de passages, et durée de passage

```
# corrélation entre la durée moyenne de passage quotidienne et le nombre de présents à 15h
plot(mean.dp, n.p15, main = "Corrélation durée moyenne de passage quotidienne\n et le nombre de présents")
cor(mean.dp, n.p15)
```

```
## [1] 0.2490515
```

```
y <- lm(mean.dp ~ n.p15)
y
```

```
##
## Call:
## lm(formula = mean.dp ~ n.p15)
##
## Coefficients:
## (Intercept)      n.p15
##      153.24      12.29
```

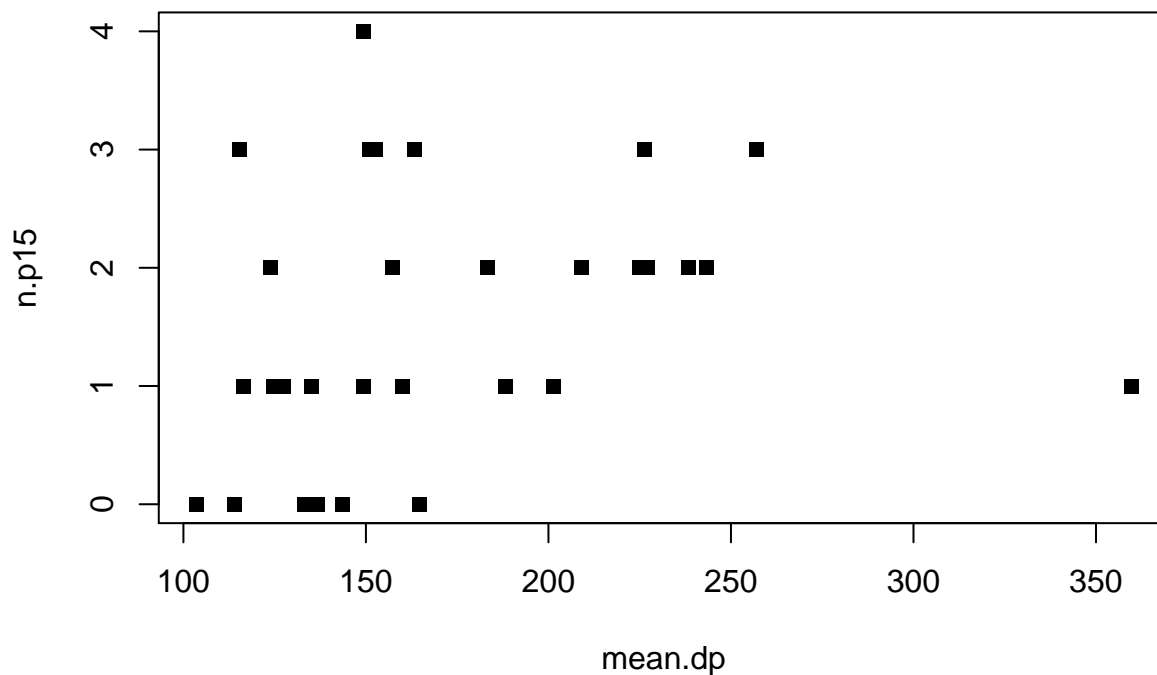
```
summary(y)
```

```
##
## Call:
## lm(formula = mean.dp ~ n.p15)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -74.81 -38.65 -16.15  33.55 194.14
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  153.238     16.865   9.086 5.54e-10 ***
## n.p15         12.287       8.873   1.385  0.177
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.46 on 29 degrees of freedom
## Multiple R-squared:  0.06203,    Adjusted R-squared:  0.02968
## F-statistic: 1.918 on 1 and 29 DF,  p-value: 0.1767
```

```
abline(y)
```



## Corrélation durée moyenne de passage quotidienne et le nombre de présents à 15h



```
# corrélation entre la durée moyenne de passage et le nombre total de passages
cor(mean.dp, n.rpu.jour)
```

```
## [1] -0.02570271
```

```
# corrélation entre taux hospitalisation et nombre de passages
cor(tx.hosp, n.rpu.jour)
```

```
## [1] 0.1218142
```

```
y <- lm(n.rpu.jour ~ tx.hosp)
y
```

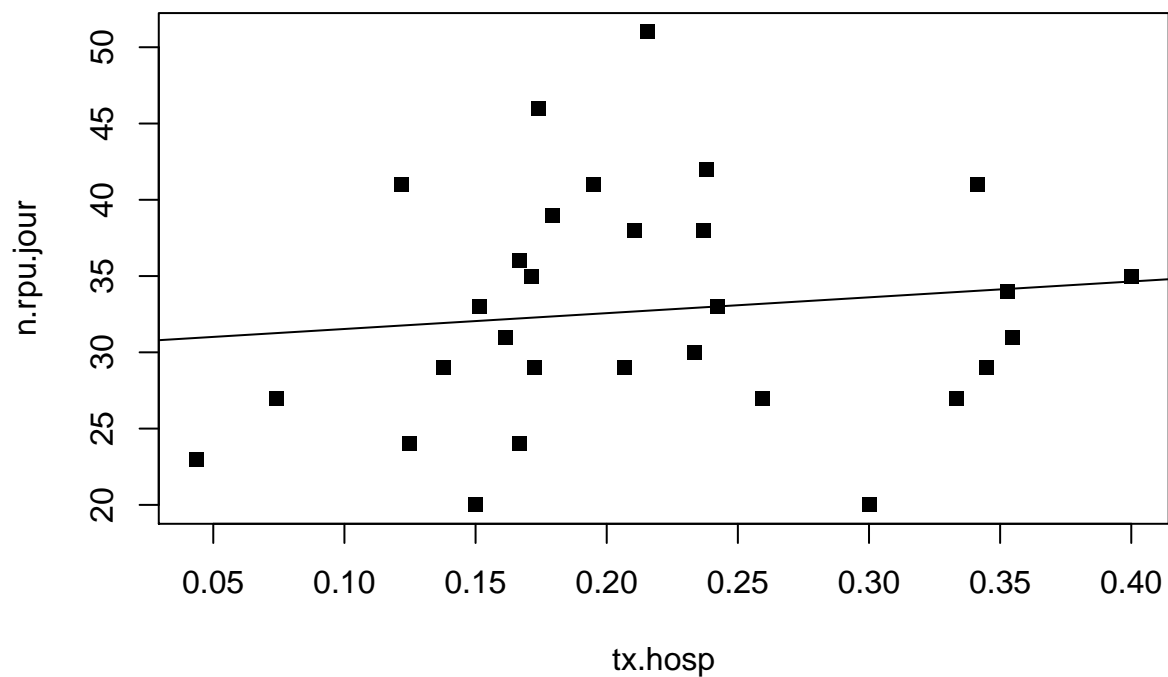
```
##
## Call:
## lm(formula = n.rpu.jour ~ tx.hosp)
##
## Coefficients:
## (Intercept)      tx.hosp
##      30.49         10.38
```

```
summary(y)
```

```
##
## Call:
```

```
## lm(formula = n.rpu.jour ~ tx.hosp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.606  -4.666  -1.166   5.186  18.269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   30.492     3.616    8.432 2.72e-09 ***
## tx.hosp       10.379     15.705    0.661   0.514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.51 on 29 degrees of freedom
## Multiple R-squared:  0.01484,    Adjusted R-squared:  -0.01913
## F-statistic: 0.4368 on 1 and 29 DF,  p-value: 0.5139
```

```
plot(tx.hosp, n.rpu.jour, col = "black", pch = 15)
abline(y)
```



## 2 Radar

```
source("../het.R")
a <- cbind(0, ts.het2, ts.mean.dp, ts.tx.hosp, ts.n.p15)
head(a)
```

```
##           X0 HET2      HET3      HET4 HET5
```

```
## 2015-10-01 0 31 164.6000 0.1612903 0
## 2015-10-02 0 41 226.3750 0.1951220 3
## 2015-10-03 0 39 116.5000 0.1794872 1
## 2015-10-04 0 46 149.3750 0.1739130 1
## 2015-10-05 0 51 188.3636 0.2156863 1
## 2015-10-06 0 31 257.1000 0.3548387 3
```

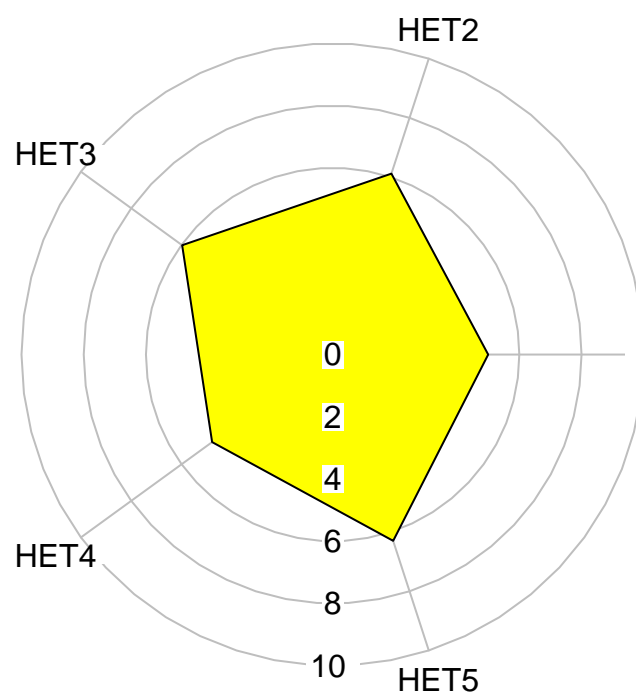
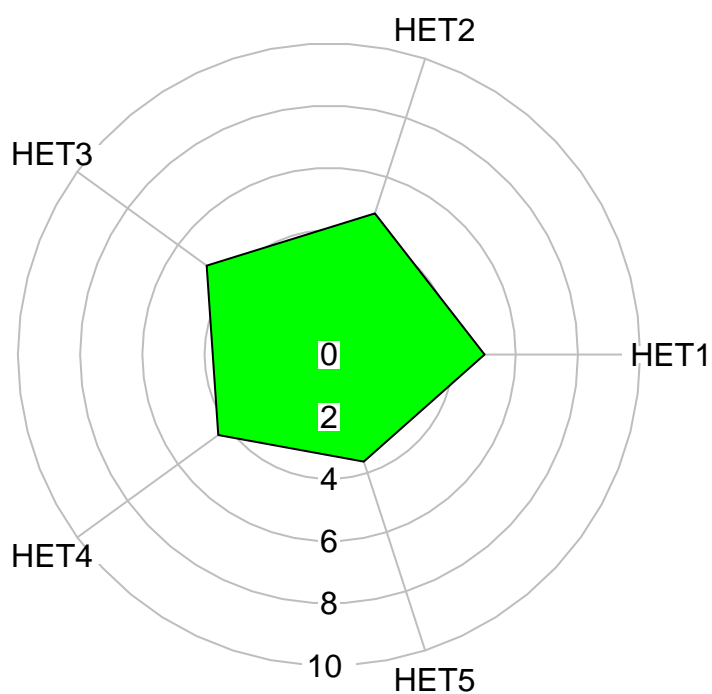
```
a[1, ]
```

```
##          X0 HET2  HET3      HET4 HET5
## 2015-10-01 0  31 164.6 0.1612903    0
```

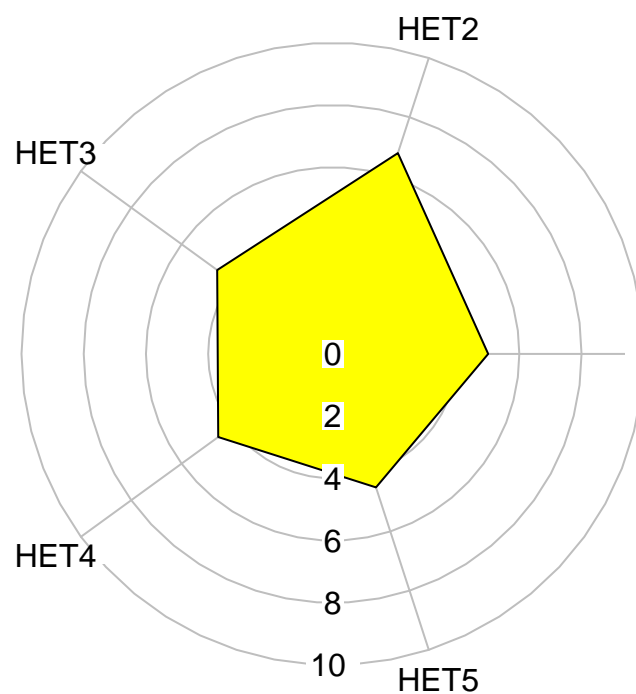
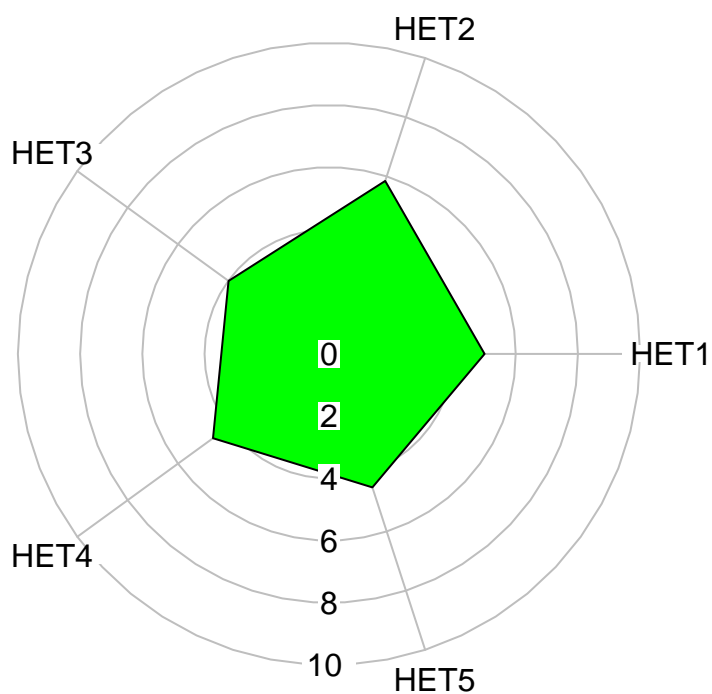
```
# normalisation sous forme de variable centrée et réduite. Par défaut, moyenne et sd sont calculés à pa
m <- 5
a[, 1] <- m # arbitraire faute de mieux
a[, 2] <- m + (a[, 2] - mean(n.rpu.jour)) / sd(n.rpu.jour)
a[, 3] <- m + (a[, 3] - mean(mean.dp)) / sd(mean.dp)
a[, 4] <- m + (a[, 4] - mean(tx.hosp)) / sd(tx.hosp)
a[, 5] <- m + (a[, 5] - mean(n.p15)) / sd(n.p15)

# indicateurs pour le mois d'octobre 2015
for(i in 1:30){
  radar.het(a[i,])
}
```

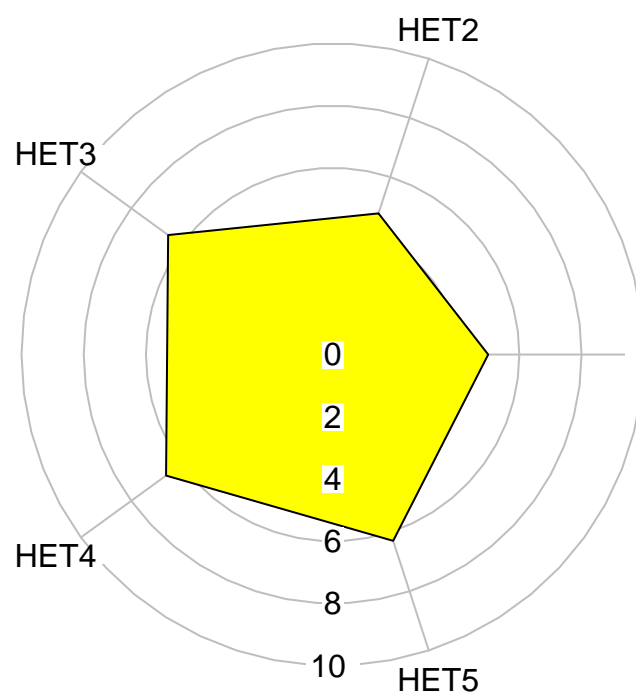
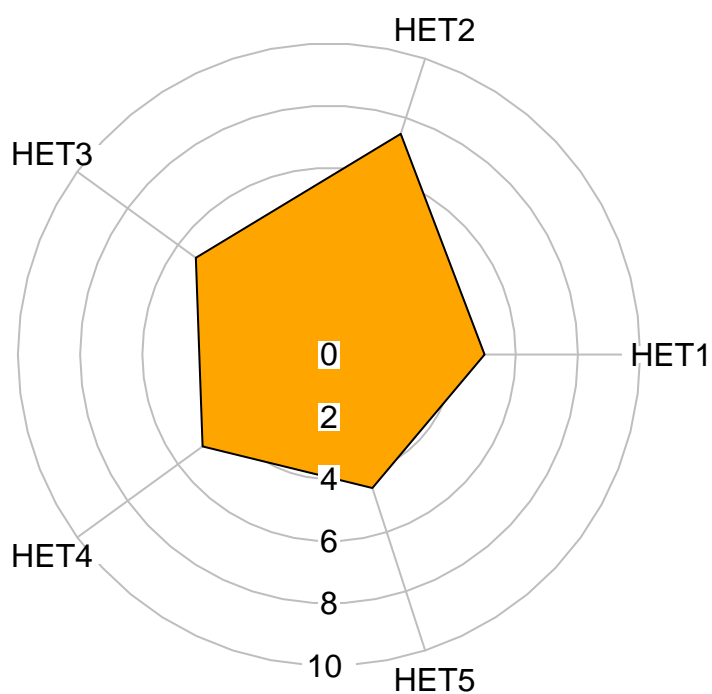
**2015-10-01 – Diagramme indicateurs HET 2015-10-02 – Diagramme indicateurs HET**



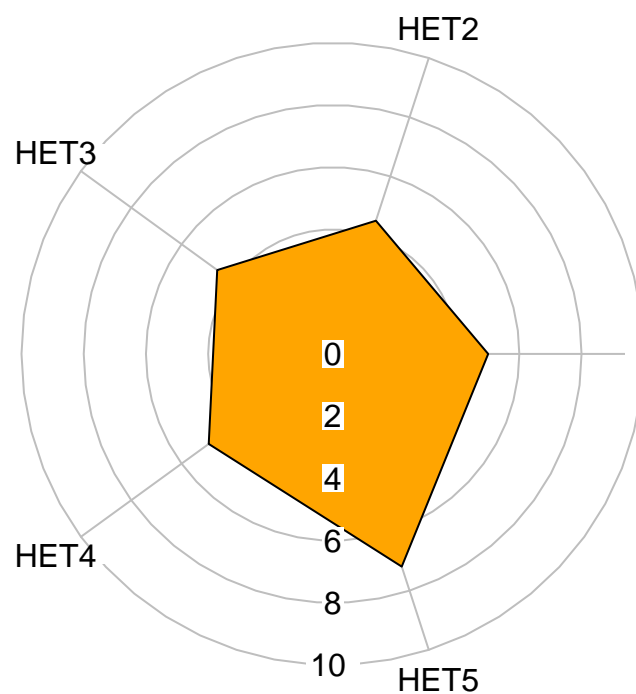
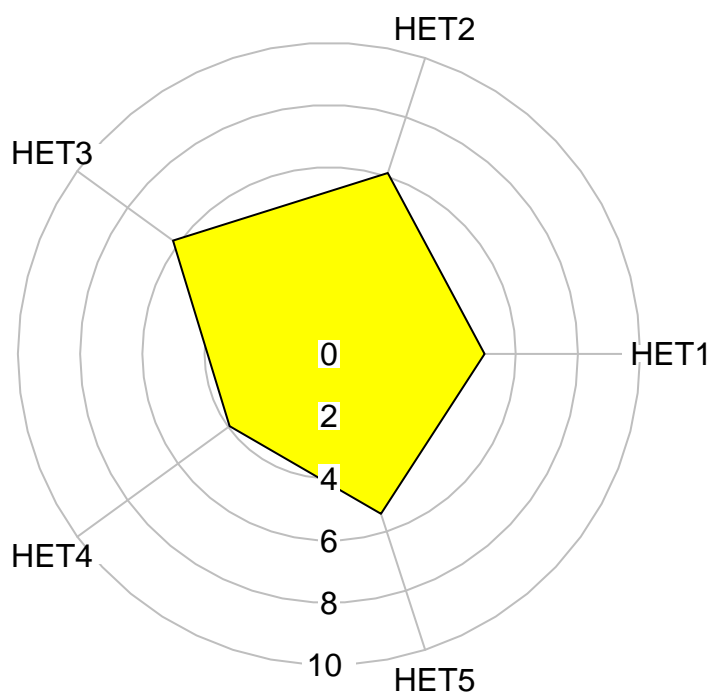
**2015-10-03 – Diagramme indicateurs HET 2015-10-04 – Diagramme indicateurs HET**



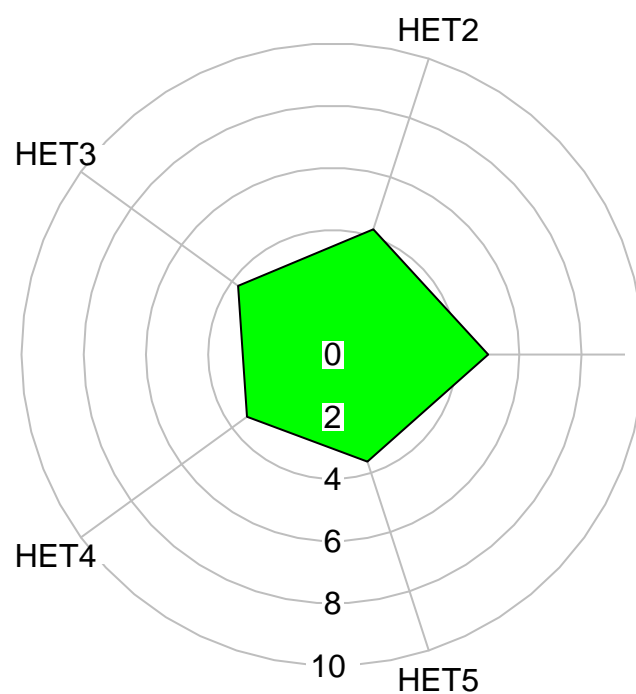
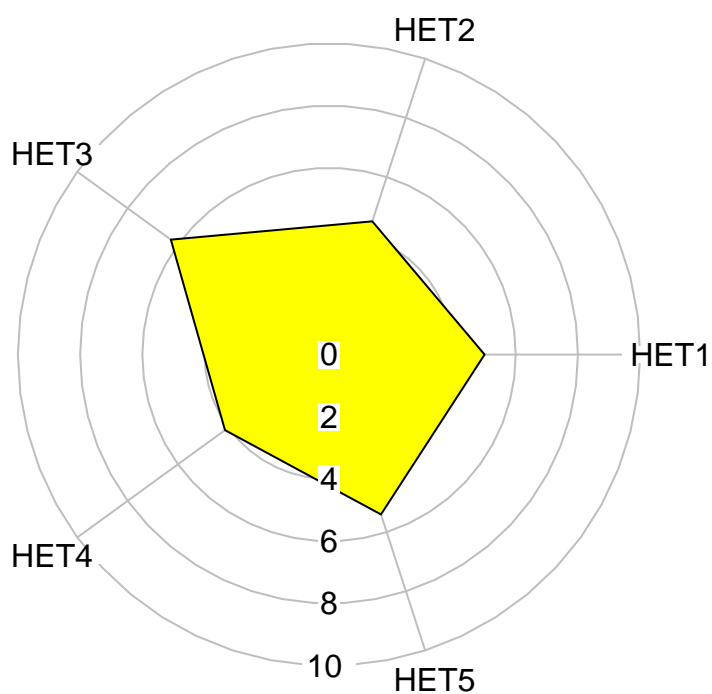
2015-10-05 – Diagramme indicateurs HET 2015-10-06 – Diagramme indicateurs HET



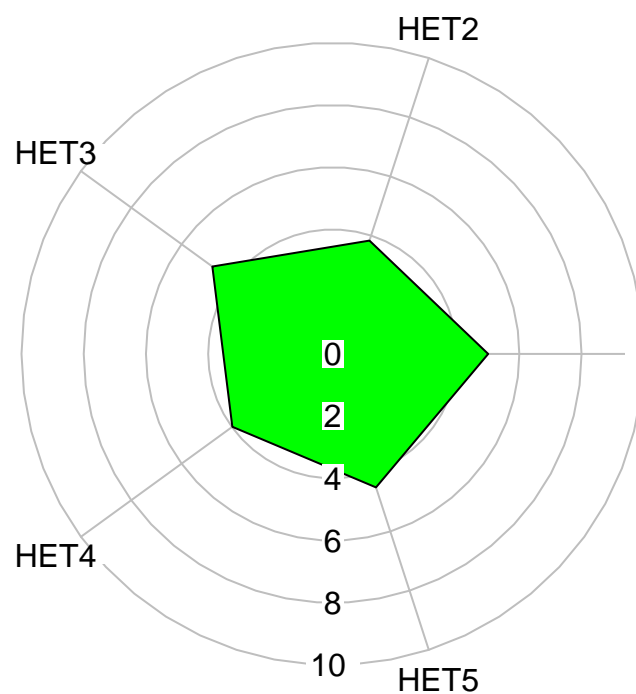
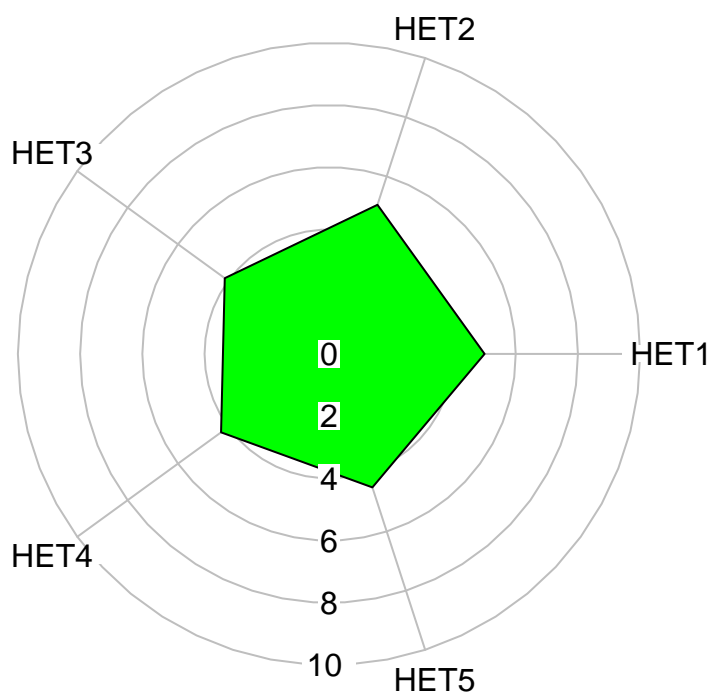
2015-10-07 – Diagramme indicateurs HET 2015-10-08 – Diagramme indicateurs HET



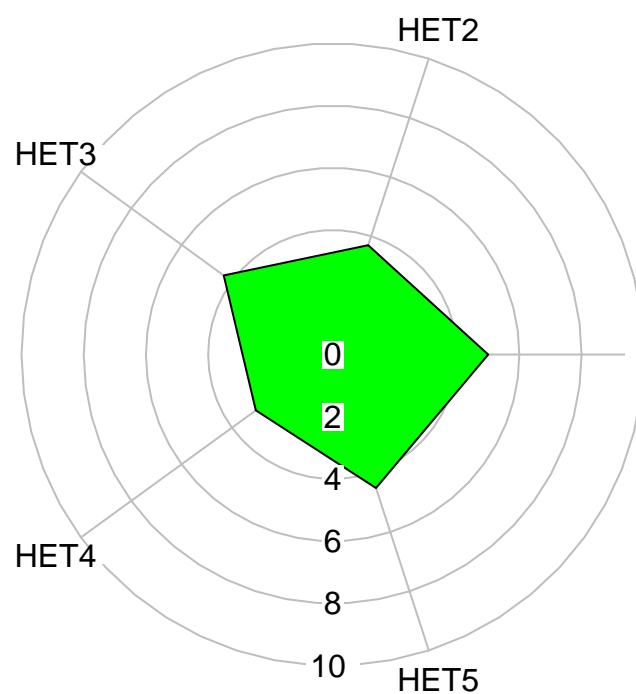
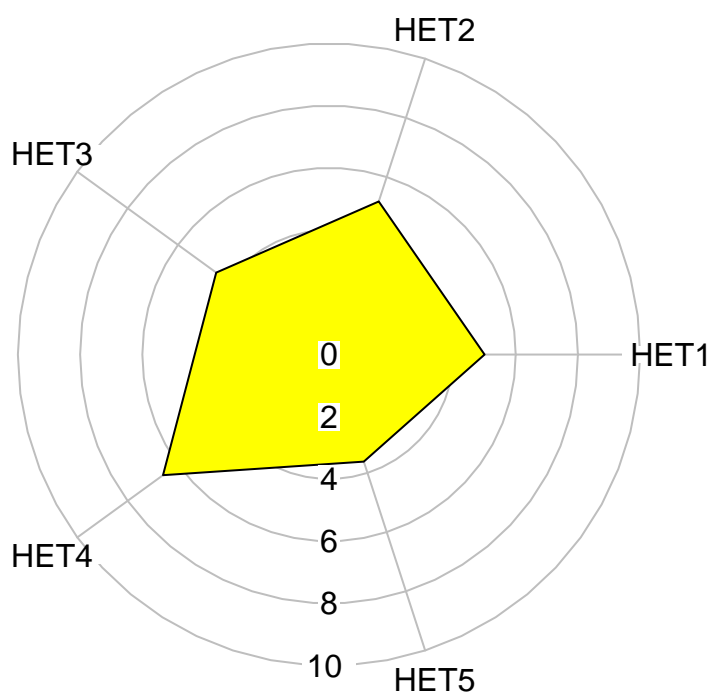
2015-10-09 – Diagramme indicateurs HET 2015-10-10 – Diagramme indicateurs HET



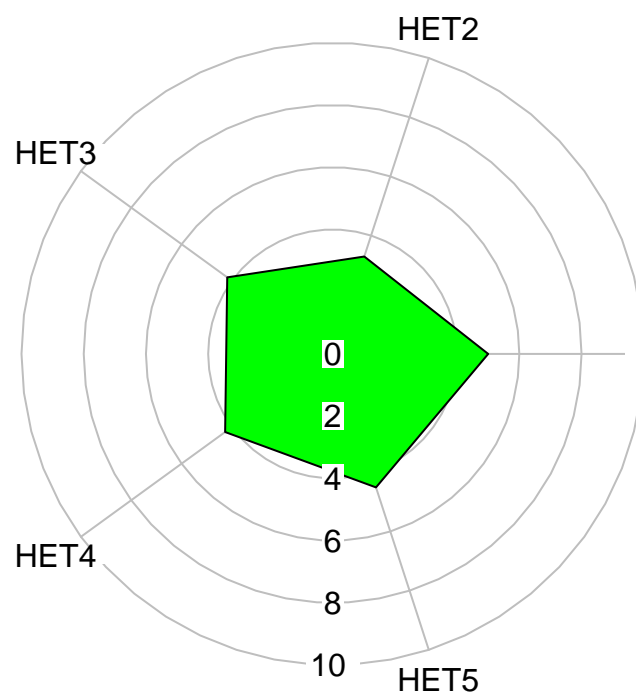
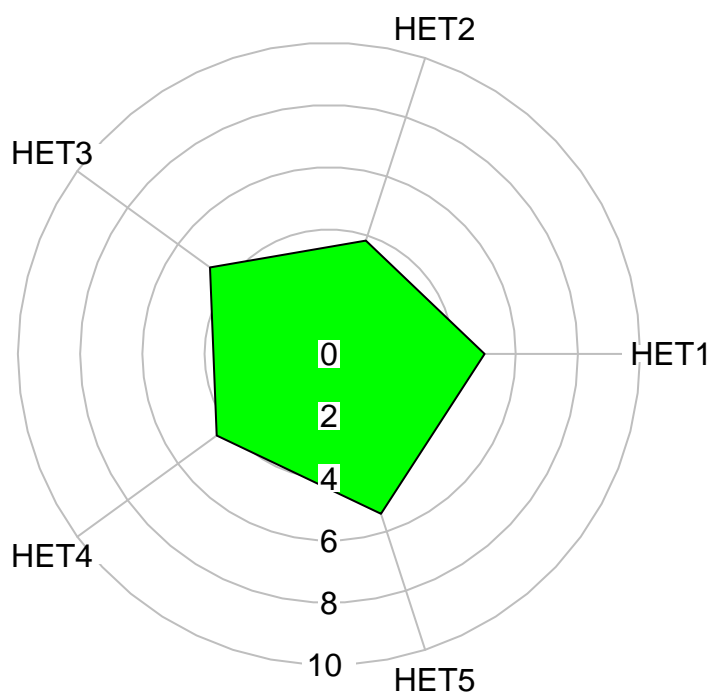
2015-10-11 – Diagramme indicateurs HET 2015-10-12 – Diagramme indicateurs HET



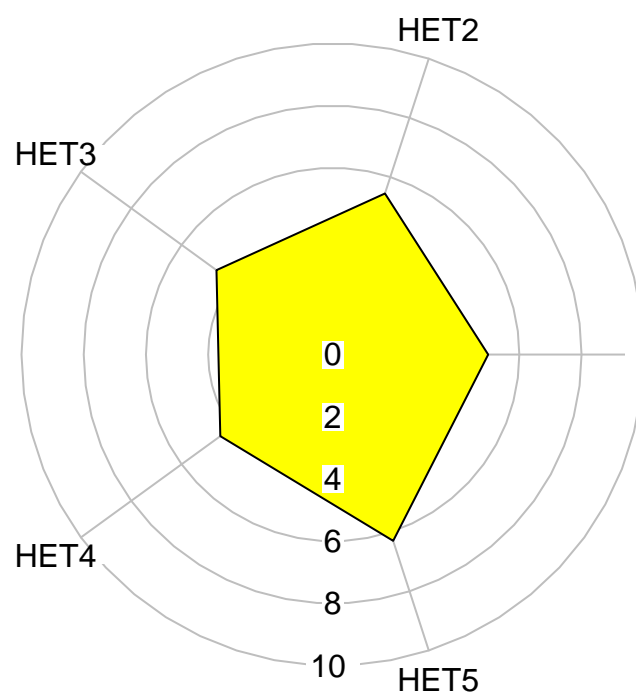
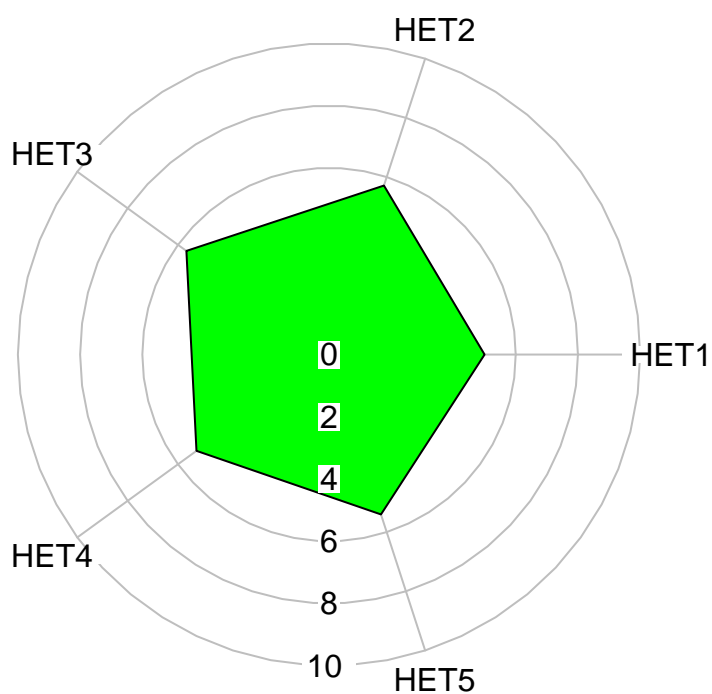
2015-10-13 – Diagramme indicateurs HET 2015-10-14 – Diagramme indicateurs HET



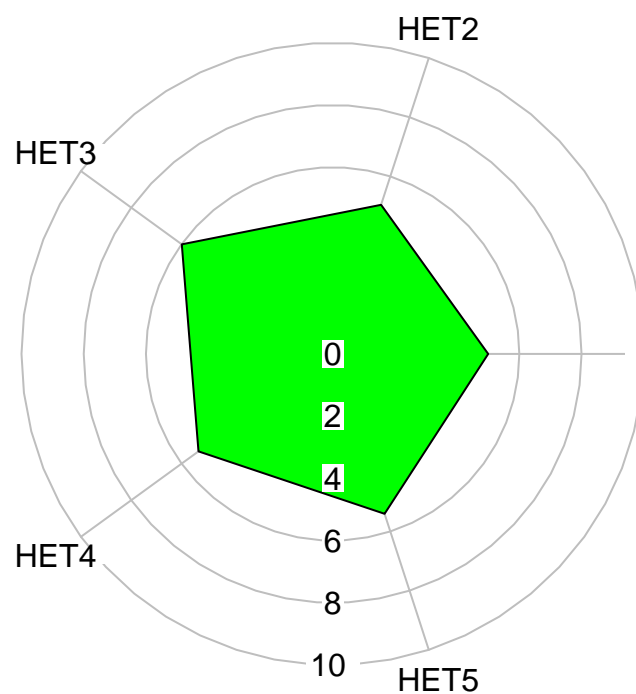
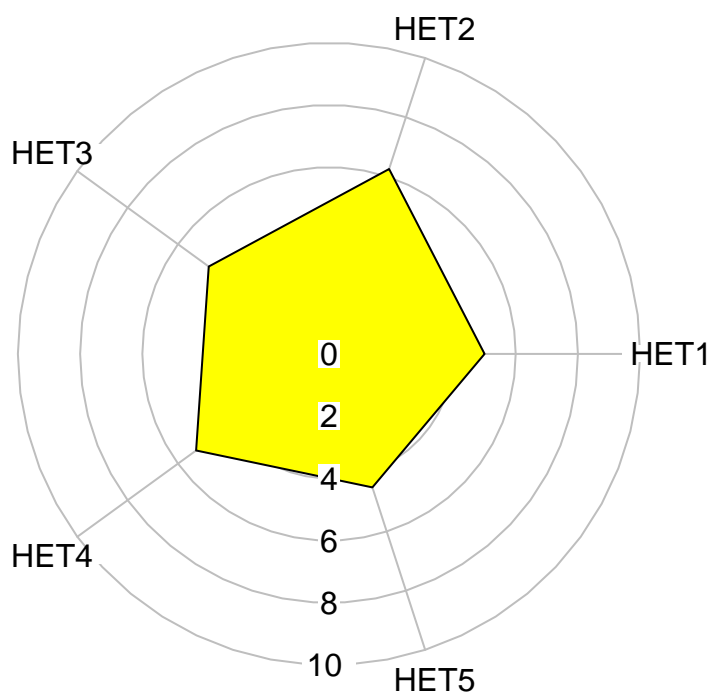
2015-10-15 – Diagramme indicateurs HET 2015-10-16 – Diagramme indicateurs HET



2015-10-17 – Diagramme indicateurs HET 2015-10-18 – Diagramme indicateurs HET

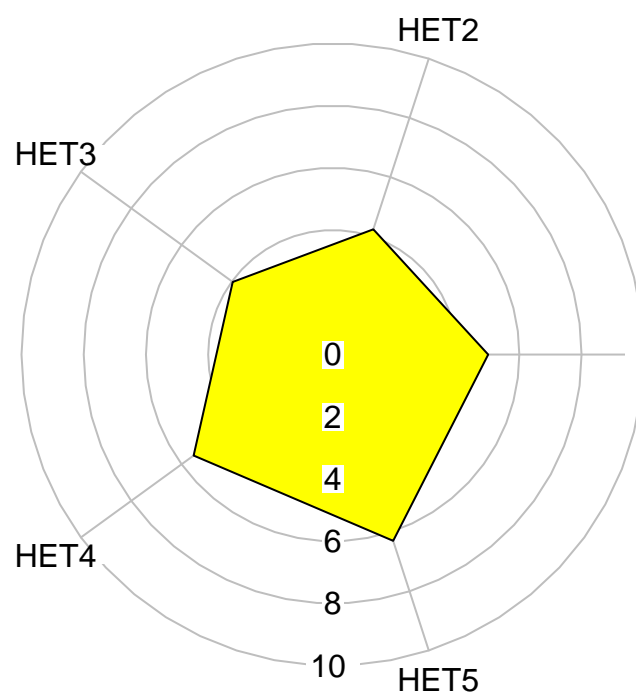
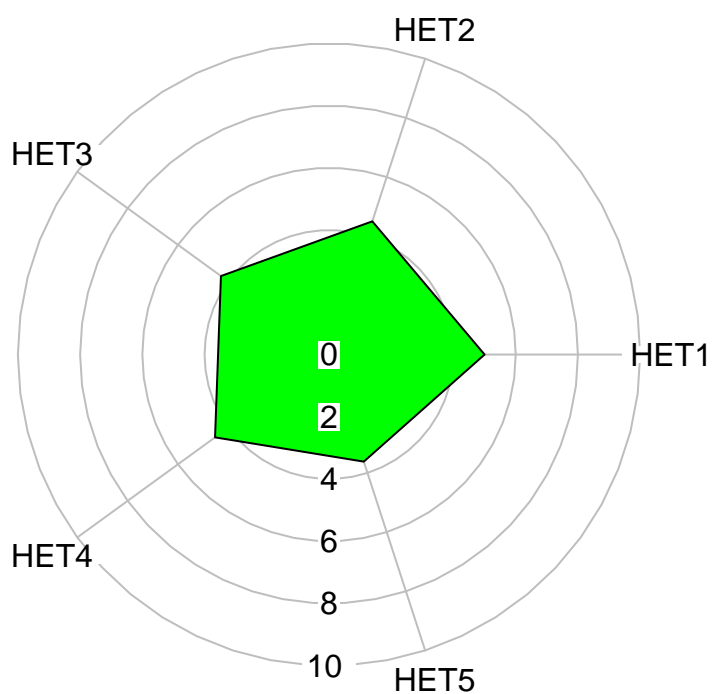


2015-10-19 – Diagramme indicateurs HET 2015-10-20 – Diagramme indicateurs HET

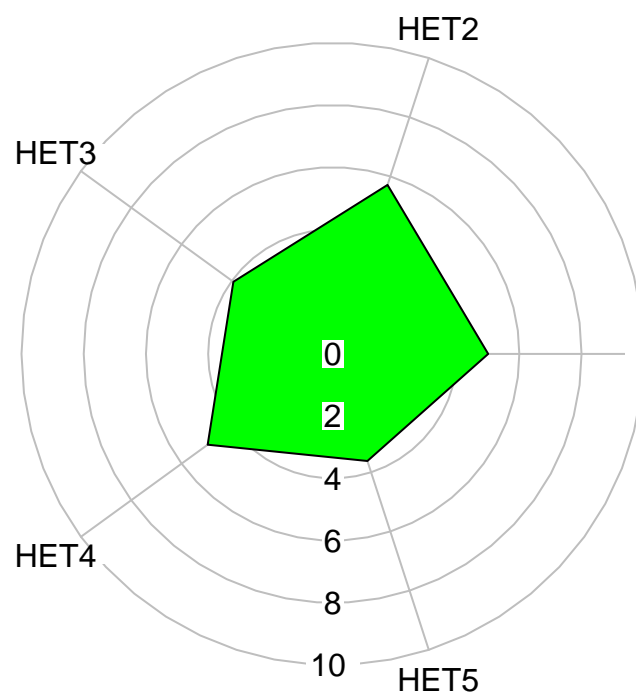
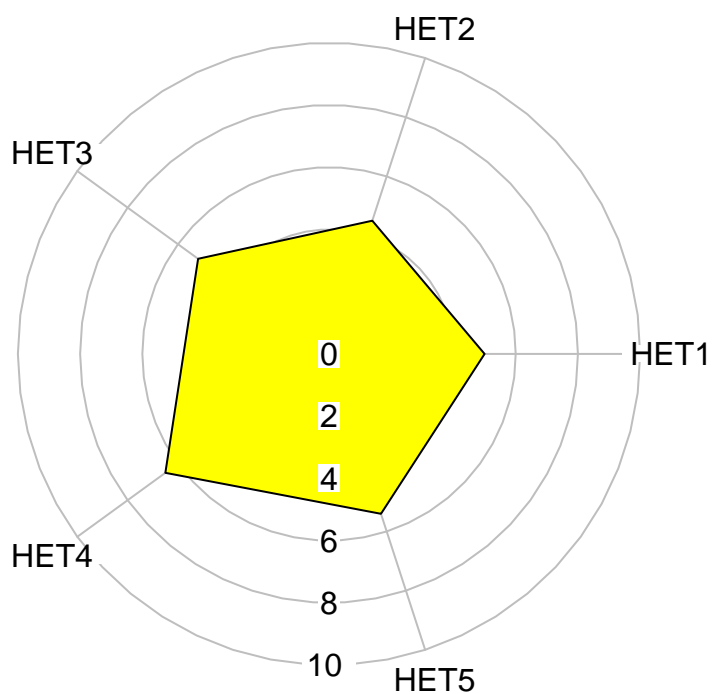




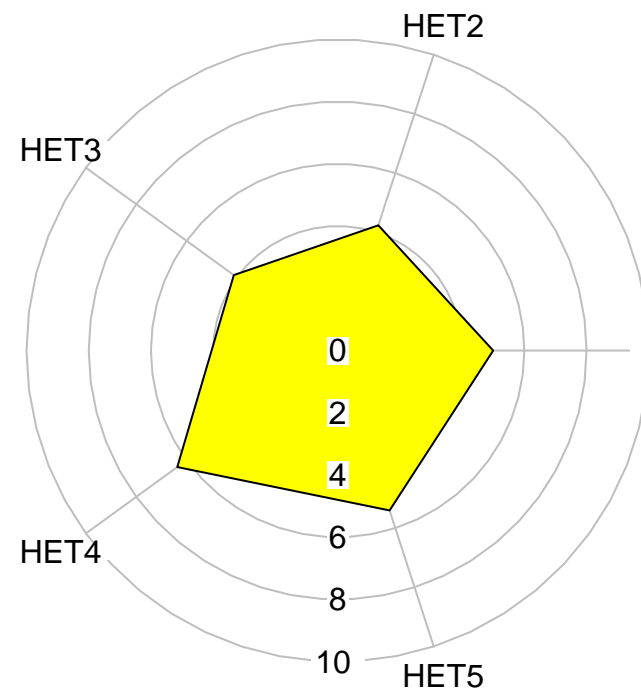
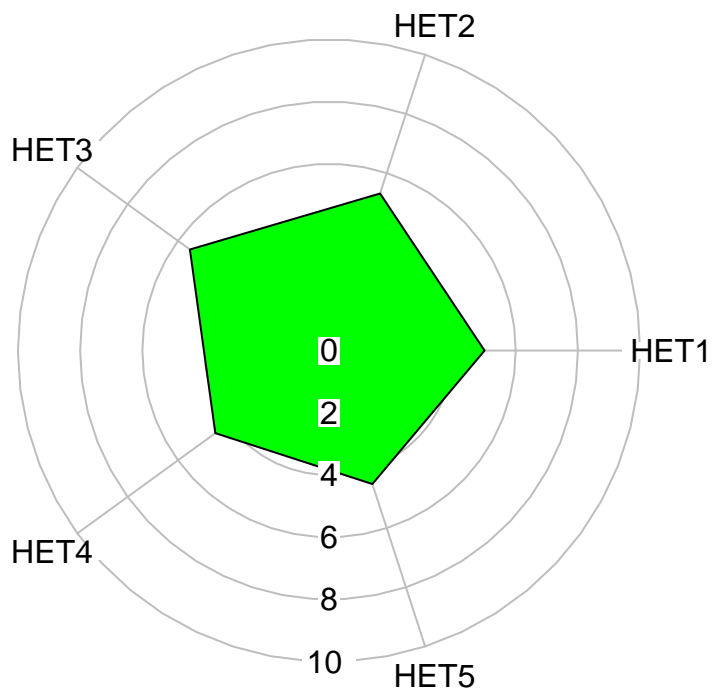
2015-10-21 – Diagramme indicateurs HET 2015-10-22 – Diagramme indicateurs HET



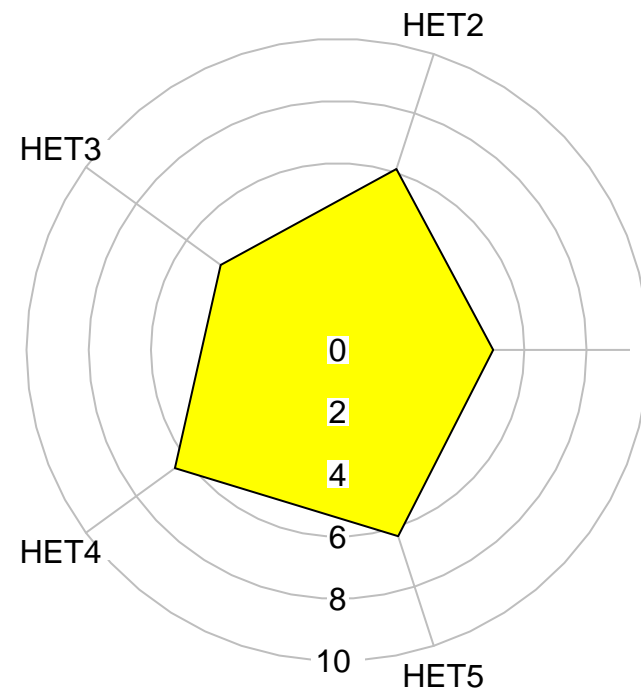
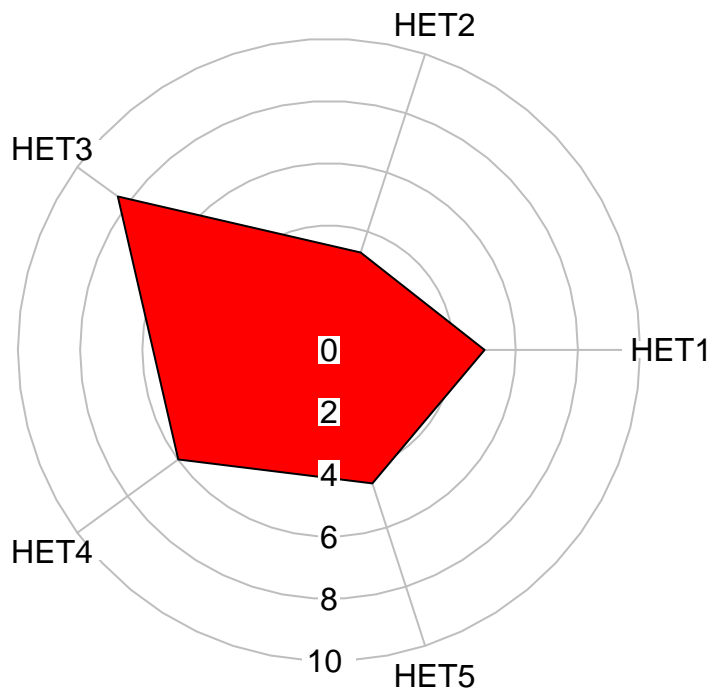
2015-10-23 – Diagramme indicateurs HET 2015-10-24 – Diagramme indicateurs HET



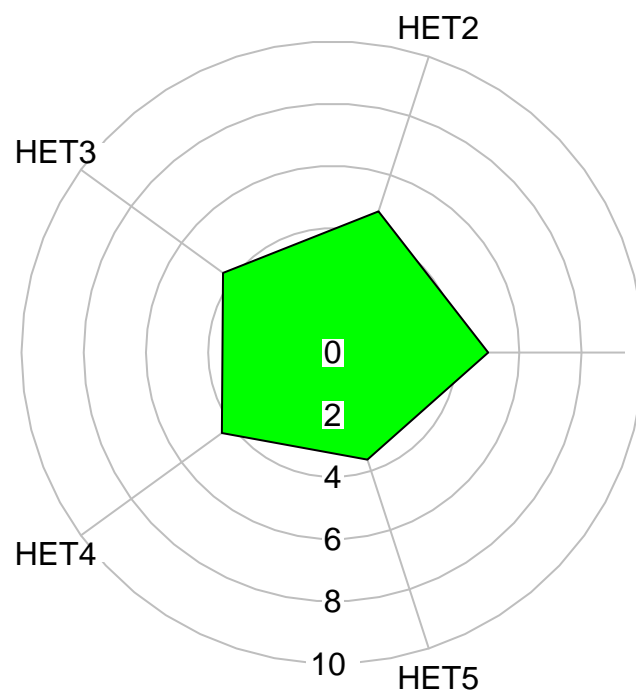
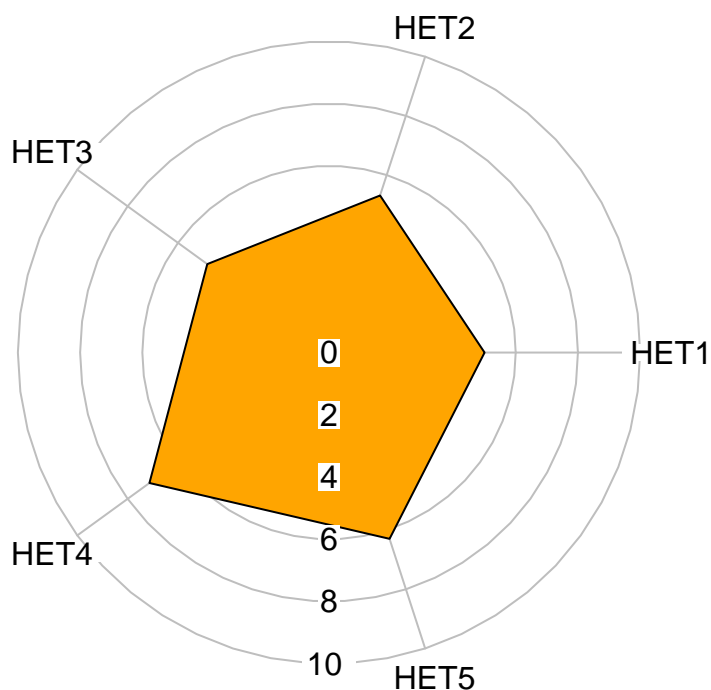
2015-10-25 – Diagramme indicateurs HET 2015-10-26 – Diagramme indicateurs HET



2015-10-27 – Diagramme indicateurs HET 2015-10-28 – Diagramme indicateurs HET



## 2015-10-29 – Diagramme indicateurs HET 2015-10-30 – Diagramme indicateurs HET



### 3 Test de la fonction het.fr

Cette fonction crée un dataframe de type Xts à partir d'un dataframe RPU en extrayant les indicateurs HET. Au final on obtient un dataframe dont chaque ligne correspond à une date et 5 colonnes correspondant au 5 indicateurs:

```
xt <- het.df(dx)
head(xt)
```

```
##           X0 HET2      HET3      HET4 HET5
## 2015-10-01  0   31 164.6000 0.1612903    0
## 2015-10-02  0   41 226.3750 0.1951220    3
## 2015-10-03  0   39 116.5000 0.1794872    1
## 2015-10-04  0   46 149.3750 0.1739130    1
## 2015-10-05  0   51 188.3636 0.2156863    1
## 2015-10-06  0   31 257.1000 0.3548387    3
```

A partir de ce dataframe on peut:

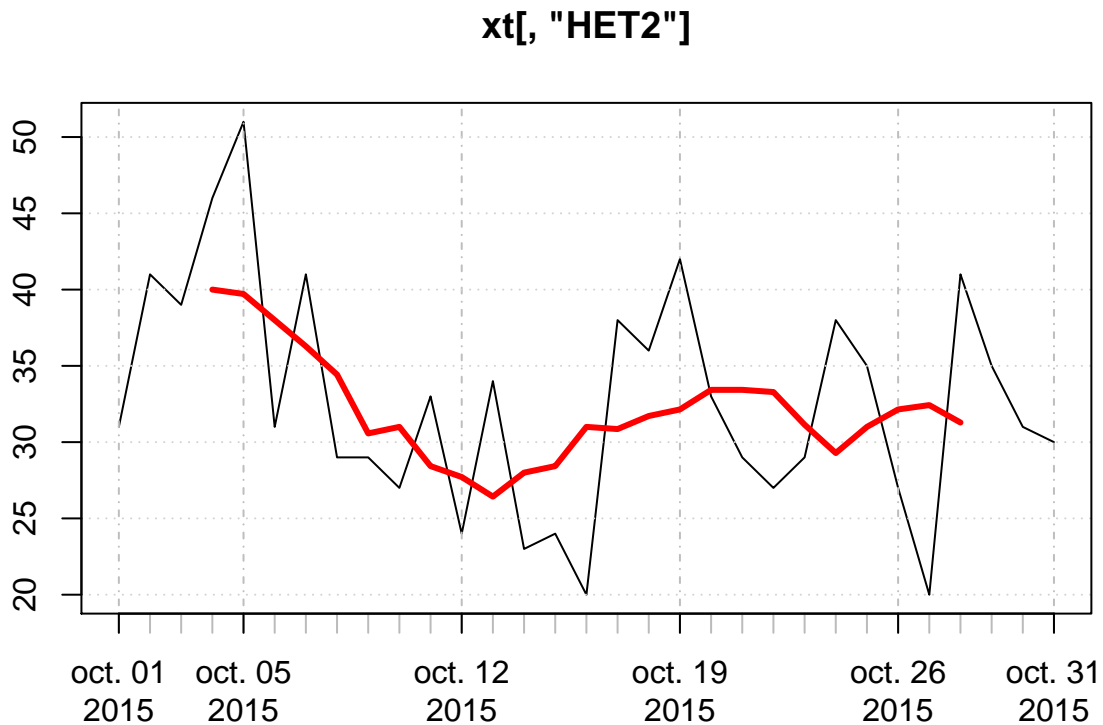
- récupérer la date du jour

```
index(head(xt))
```

```
## [1] "2015-10-01" "2015-10-02" "2015-10-03" "2015-10-04" "2015-10-05"
## [6] "2015-10-06"
```

- tracer la courbe de tendance pour un indicateur

```
plot(xt[, "HET2"])
lines(rollmean(xt[, "HET2"], 7), col = "red", lwd = 3)
```



- créer le polygone correspondant après avoir réduit-centré la variable: A FAIRE

```
# normalisation sous forme de variable centrée et réduite. Par défaut, moyenne et sd sont calculés à pa
m <- 5
xt[, 1] <- m # arbitraire faute de mieux
xt[, 2] <- m + (xt[, 2] - mean(n.rpu.jour)) / sd(n.rpu.jour)
xt[, 3] <- m + (xt[, 3] - mean(mean.dp)) / sd(mean.dp)
xt[, 4] <- m + (xt[, 4] - mean(tx.hosp)) / sd(tx.hosp)
xt[, 5] <- m + (xt[, 5] - mean(n.p15)) / sd(n.p15)

head(xt)
```

```
##          X0      HET2      HET3      HET4      HET5
## 2015-10-01  5  4.770189  4.861409  4.400482  3.618345
## 2015-10-02  5  6.114365  5.978744  4.787962  6.295302
## 2015-10-03  5  5.845530  3.991417  4.608894  4.510664
## 2015-10-04  5  6.786453  4.586033  4.545052  4.510664
## 2015-10-05  5  7.458540  5.291226  5.023490  4.510664
## 2015-10-06  5  4.770189  6.534471  6.617231  6.295302
```

```
radar.het(xt[1,])
```

## 2015-10-01 – Diagramme indicateurs HET

