

Exemples de code permettant la création de graphiques avec les packages ‘ggplot2’ et ‘data.table’

Pierre Tocquin

24/02/2018

Lecture et traitement des données tabulaires (fichier de type *csv*)

Lecture des données, méthode `data.table`

```
data1 <- fread(input = "student1/resulat.csv")
data2 <- fread(input = "student2/gravi.csv")
```

Mise en forme des données, méthode `data.table`

```
data1.summarized <- data1[,
  list(diff=longueur[jour==1]-longueur[jour==0]),
  by=list(plante, lumière)][,
  list(mean=mean(diff), sd=sd(diff)),
  by=lumière]

data2.summarized <- data2[time == 1,
  list(direction=mean(angle-180), ecarttype=sd(angle-180)),
  by=list(Position, NPA)]
```

Graphiques de type ‘BarPlot’

Exemple 1

```
# Définition des paramètres principaux du graphique
g <- ggplot(data = data2.summarized, mapping = aes(x = Position, y = direction, fill = NPA))

# Création du graphique
g.basic <- g +
  # choix d'un barplot
  geom_bar(stat = "identity", position = position_dodge(), color = "black") +
  # ajout des écart-types
  geom_errorbar(width=.25, aes(ymin=direction-ecarttype, ymax=direction),
  # re-positionnement correct des barres d'erreur
  position=position_dodge(.9))
g.basic

# Amélioration du graphique
g.improved <- g.basic +
  # re-définition des couleurs des barres
```

```

scale_fill_manual(values = c("deepskyblue", "orange")) +
# re-définition des étiquettes de l'axe x
scale_x_discrete(labels=c("90°", "0°")) +
ylab("Direction de croissance \npar rapport à l'axe vertical (°)") + # Titre axe y
xlab("Orientation de la boîte (°)") # Titre axe x

g.improved

```

Exemple 2

```

# Définition des paramètres principaux du graphique
g <- ggplot(data = data1.summarized,
  mapping = aes(x = lumière, y = mean, fill = lumière))

# Création du graphique de type 'barplot'
g.basic <- g +
  geom_bar(stat = "identity", position = position_dodge()) + # choix d'un barplot
  geom_errorbar(width=.25, aes(ymin=mean-sd, ymax=mean+sd)) # ajout des écart-types

g.basic

# Amélioration du graphique
g.improved <- g.basic +
  # re-définition des couleurs des barres
  scale_fill_manual(values = c("white","blue", "black", "red", "darkred")) +
  # re-définition des étiquettes de l'axe x
  scale_x_discrete(labels=c("Blanc", "450nm", "Obscurité", "660nm", "750nm")) +
  ylab("Croissance (cm/jour)") + # Titre axe y
  xlab("Conditions lumineuses") + # Titre axe x
  guides(fill=FALSE) # Suppression de la légende

g.improved

```

Graphiques de type 'X-Y, lignes'

Exemple

```

# Définition des paramètres principaux du graphique
# Notez que l'argument 'fill' des barplots est devenu 'color'
g <- ggplot(data = data.per.day, mapping = aes(x = jour, y = mean, color = lumière))

# Création du graphique de type 'X-Y, lignes'
g.basic <- g +
  geom_line() + # choix d'un graphique x/y de type ligne
  geom_errorbar(width=.05, aes(ymin=mean-sd, ymax=mean+sd)) # ajout des écart-types

g.basic

# Amélioration du graphique

```

```

g.improved <- g.basic +
  # re-définition des couleurs des barres
  scale_color_manual(values = c("white", "blue", "black", "red", "darkred")) +
  scale_x_continuous(breaks = c(0,1)) + # re-définition des étiquettes de l'axe x
  ylab("Croissance (cm/jour)") + # Titre axe y
  xlab("Jours") # Titre axe x

g.improved

```

Graphiques sur plusieurs panneaux, nombre de variables est = 3

Il est parfois nécessaire de fixer le type des colonnes qui servent à trier ou regrouper les données sur `factor`. Dans le cas présent si vous ne faites pas cela pour la colonne `stimulus`, elle ne pourra pas vous servir à paramétrer la couleur des points via l'argument `fill` de `ggplot`. Une façon de faire est de fixer le type de colonne lors de la lecture `fread` à l'aide de l'argument `colClasses`. Notez que `ggplot2` considère automatiquement les types 'string' comme des facteurs, raison pour laquelle, dans le cas présent, le problème se pose uniquement pour la colonne 'stimulus' dont les valeurs 0 et 90 sont considérés comme des 'numeric' (ce qu'ils sont) et non automatiquement interprétés comme des facteurs par `ggplot2`.

```

data4 <- fread(input = "student3/fulldata.csv",
  colClasses = c("factor", "factor", "factor", "factor", "numeric"))

# Définition des paramètres principaux du graphique
# Notez ici l'utilisation de l'argument 'group'
g <- ggplot(data = data4, mapping = aes(x = stimulus, y = angle, fill = stimulus))

# Création du graphique
g.basic <- g +
  geom_dotplot(binaxis = "y", stackdir = "center", binwidth = 7)

# Séparation du graphique en panneaux sur base du génotype et du traitement
g.facet <- g.basic +
  # le paramètre labeller sert au renommage
  facet_grid(. ~ Genotype + traitement, labeller = labeller(Genotype = c(GUS = "WT", PIN = "pin3-4")))

# Amélioration du graphique
g.improved <- g.facet +
  scale_x_discrete(labels=c("90°C", "0°C")) + # re-définition des étiquettes de l'axe x
  ylab("Courbure (°)") + # Titre axe y
  xlab("Orientation de la boîte (°)") + # Titre axe x
  guides(fill=FALSE) # Suppression de la légende

g.improved

```