

Présentation du Langage R

Pr Roch Giorgi

 roch.giorgi@univ-amu.fr

Généralités (1)

- S
 - ✓ Langage de programmation de très haut niveau
 - ✓ Environnement d'analyse des données et des graphiques
 - ✓ Conçu dans les années 1975-1976 (John Chambers)
 - ✓ Interpréteurs : S-PLUS (version commerciale), puis R (version libre)
- R
 - ✓ Logiciel libre mettant en œuvre le langage de programmation S
 - ✓ Projet GNU
 - ✓ Compile et fonctionne sur une grande variété de plateformes UNIX, Windows et Mac OS

Généralités (2)

- R est un logiciel de traitement statistique des données
 - ✓ Interprète des commandes
 - ✓ Dispose d'une bibliothèque « interne » de fonctions statistiques
 - ✓ Possibilité d'enrichir cette bibliothèque par des fonctions « externes » via le système des packages
 - publics, déposés sur le site CRAN, en téléchargement libre
 - personnels
 - ✓ Larges fonctionnalités graphiques
 - ✓ Selon les besoins, il est utilisable tel quel pour effectuer les analyses statistiques

Généralités (3)

- R est un langage de programmation
 - ✓ Création de script(s)
 - ✓ Langage interprété (😊 portabilité application ; 😞 lenteur)
 - Ce n'est pas un langage compilé (😊 rapidité ; 😞 portabilité)
 - Aime peu les boucles...
 - ✓ Données simples, données structurées
 - ✓ Opérations d'entrée / sortie
 - ✓ Branchements conditionnels
 - ✓ Boucles
 - ✓ Création de fonctions
 - ✓ Organisation du code en modules (fichier « .r »)
 - ✓ Distribution des modules (fichier « .r », package)

Langage Interprété – Langage Compilé (1)

- Langage interprété (R,...)
 - ✓ Requier un programme – l'interprète – pour exécuter une commande
- Langage compilé (C, C++,...)
 - ✓ Les programmes sont convertis en code machine par le compilateur, puis directement exécutés par l'ordinateur
- Le programme R
 - ✓ Interprète des commandes écrites dans le langage R
 - ✓ Exécute immédiatement des commandes, une à une et en séquence

Langage Interprété – Langage Compilé (2)

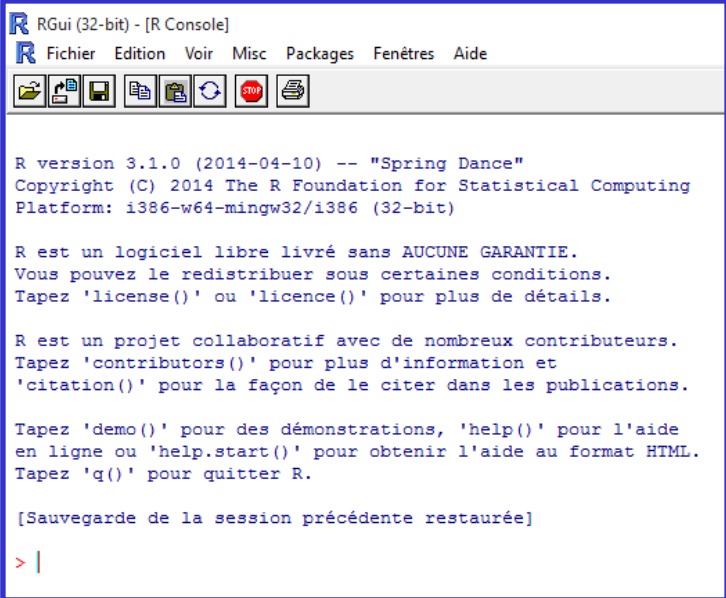
- R est un langage puissant pour les calculs mathématiques et statistiques
 - ✓ Basé sur la notion de vecteur, matrice, simplifiant les calculs mathématiques
 - Pour limiter le recours aux structures itératives (`for`, `while`,...)
 - ✓ Pas de typage ni déclaration obligatoire des variables
 - Obligatoire en C, C++
 - ✓ Programmes courts
 - Selon les besoins...
 - ✓ Ecriture rapide
 - Pour les analyses statistiques élémentaires
 - Mais on peut passer plus de temps à déboguer un programme que de temps passé à l'écrire (cela s'apprend !)

Langage Interprété – Langage Compilé (3)

- Les packages R peuvent contenir
 - ✓ Du code R
 - ✓ Du code compilé (C, C++, Fortran)
- Permet une optimisation en terme de temps de calcul

Interface

- R offre
 - ✓ Une interface graphique rudimentaire (Windows, Mac OS X)
 - ✓ Une invite de commande (seule offre sous Unix, Linux)



```
RGui (32-bit) - [R Console]
R Fichier Edition Voir Misc Packages Fenêtres Aide

R version 3.1.0 (2014-04-10) -- "Spring Dance"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

[Sauvegarde de la session précédente restaurée]

> |
```


Stratégie de Travail (1)

1. Approche « code virtuel et objets réels »

- ✓ Ecrire des expressions dans la ligne de commande pour une évaluation immédiate du code

```
> 4*2
```

```
[1] 8
```

```
>
```

- ✓ Création d'objets contenant un résultat et stockés en mémoire dans l'espace de travail en cours de R

```
> x <- 4*2
```

```
>
```


```
> x
```

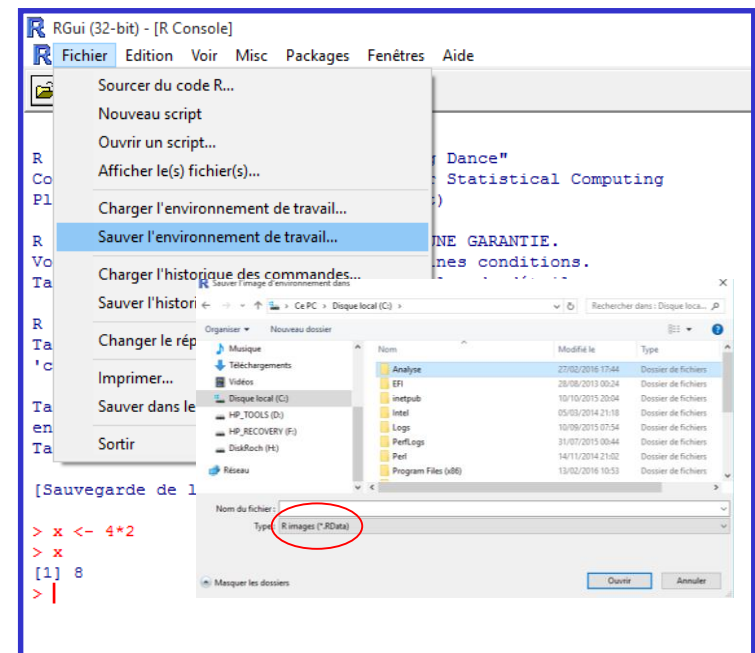
```
[1] 8
```

```
>
```

Stratégie de Travail (2)

1. Approche « code virtuel et objets réels »

- ✓ A la fin de la session de travail, sauvegarde d'une image de l'espace de travail sur le disque dur de l'ordinateur
- ✓ Conservation de ces objets pour une autre session de travail
- ✓ A l'aide de la souris 
- ✓ A l'aide d'une fonction R
`save.image()`
- ✓ Créant un fichier `.RData`
- ✓ Attention : être attentif à l'endroit où la sauvegarde est effectuée



Stratégie de Travail (3)

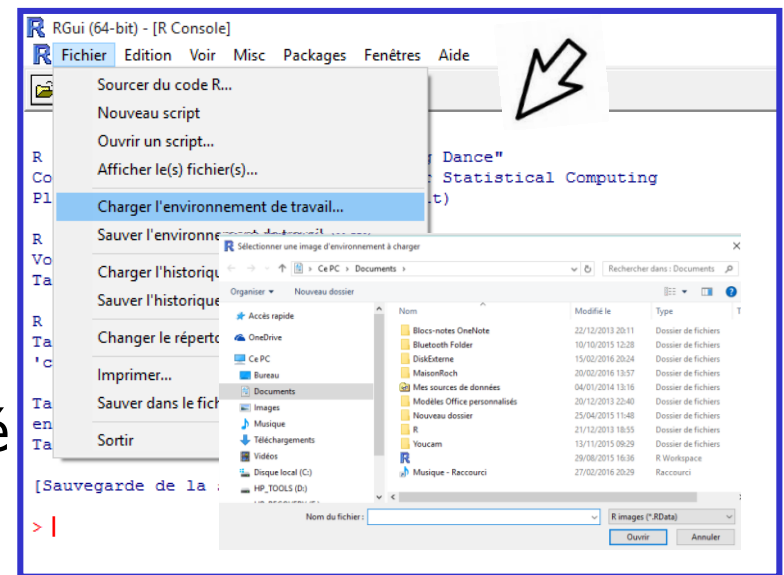
1. Approche « code virtuel et objets réels »

✓ Avantage

- On peut charger ultérieurement le fichier `.RData` créé (souris ; fonction `load.image()`)
- Travailler avec les objets qu'il contient

✓ Inconvénient

- Le code ayant permis de créer les objets n'est pas sauvegardé
- On ne sait pas comment les valeurs contenues dans les objets ont été obtenues



Stratégie de Travail (4)

2. Approche « code réel et objets virtuels »

- ✓ Le code servant à créer les objets, effectuer l'analyse est sauvegardé dans un ou des fichiers de script
- ✓ Un script contient les expressions R, le code de fonctions personnelles
- ✓ Un script est un fichier ayant l'extension `.r` (par convention)

```
# Exemple de contenu d'un fichier de script
# Fichier dans : c:\stat\calcul.r
x <- c(2, 4, 2, 8, 1, 9, 10)
y <- length(x)
sum(x)/y
```

Quel est le calcul effectué en utilisant ce script ?

Stratégie de Travail (5)

2. Approche « code réel et objets virtuels »

✓ Avantage

- L'analyse peut être refaite à tout moment, par n'importe qui, à l'aide du code contenu dans le fichier de script
- Les objets sont recréés à besoin
- Simple copier/coller du code du fichier de script dans l'invite de commande de R, ou utilisation de fonction R (`source()`), ou utilisation d'éditeurs appropriés (Tinn-R, R Studio)

✓ Inconvénient

- Si programme long, que tous les objets utiles ont été créés et toutes les actions importantes ont été effectuées (par exemple, nettoyage d'une grande base) → enregistrer le `.RData`

Exemple

```
> # Fichier de script dans : c:\stat\calcul.r  
> source("c:/stat/calcul.r")  
> x
```

Quelle est la valeur retournée ?

```
> # Fichier de script dans : c:\stat\calcul.r  
> source("c:/stat/calcul.r")  
> z <- sum(x)/y  
> save.image("c:/stat/calcul.RData")
```

Quelle est la modification apportée dans le fichier de script calcul.r ?
Comment récupérer la valeur contenue dans l'objet z une fois la session de travail terminée ?

Editeurs de Texte – Editeurs de Code R (1)

- Lorsque l'on travail avec des fichiers de script
 - ✓ Recommandé !
- Editeurs de texte
 - ✓ Création, édition, lecture/sauvegarde de fichiers textes
 - ✓ Texte pur
 - Pas d'information, de présentation et de mise en forme concernant la syntaxe utilisée, le nom des fonctions, le type d'action,...
 - ✓ Editeur quel qu'il soit
 - bloc-notes sous Windows
 - éditeur de script dans R
 - ...

Editeurs de Texte – Editeurs de Code R (2)

- Editeurs de code R

- ✓ Création, édition, lecture/sauvegarde de fichiers textes

- ✓ Texte pour « programmeur »

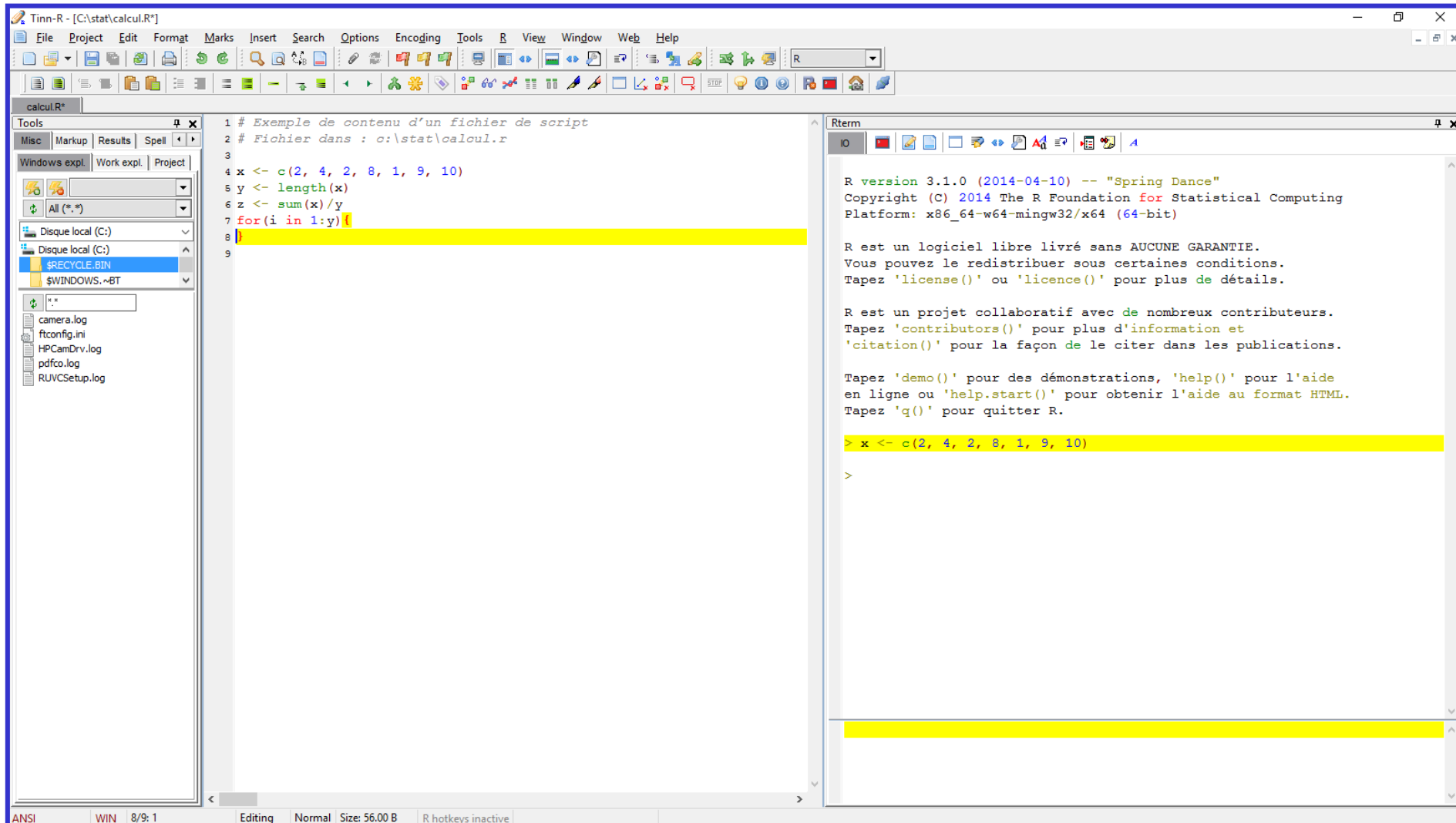
- Reconnait la syntaxe d'un langage de programmation, assiste à sa mise en forme (indentation automatique du code, coloration syntaxique), reconnaît des fonctions, manipulation d'objets,...

```
1 # Exemple de contenu d'un fichier de script
2 # Fichier dans : c:\stat\calcul.r
3
4 x <- c(2, 4, 2, 8, 1, 9, 10)
5 y <- length(x)
6 z <- sum(x)/y
7 for(i in 1:y){
8 }
```

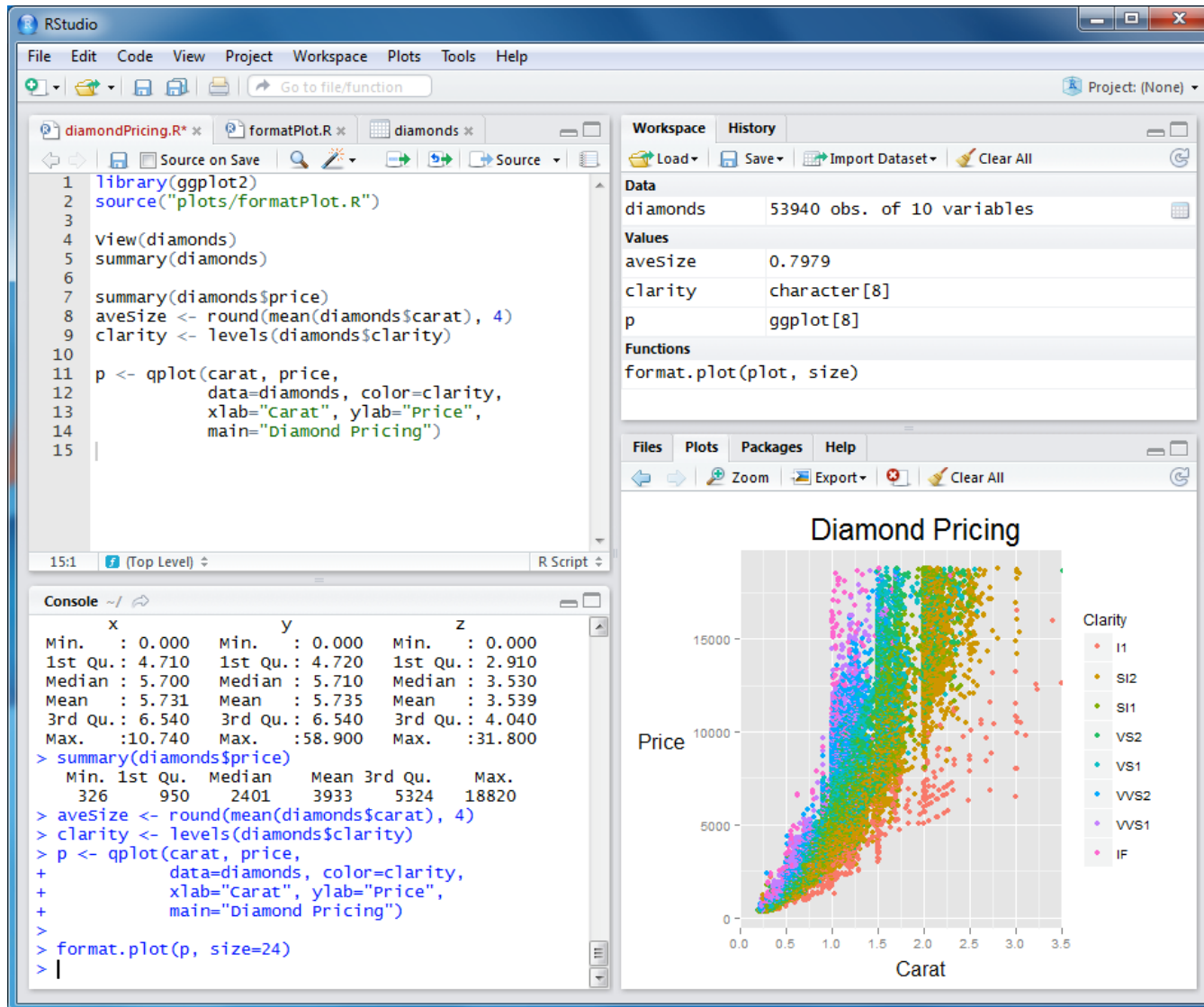
- ✓ Exemples

- Tinn-R (<https://sourceforge.net/projects/tinn-r/>)
- Rstudio (<https://www.rstudio.com/>)
- Notepad++ associé à NppToR
- ...

Editeurs de Texte pour Programmeur R (1)



Editeurs de Texte pour Programmeur R (2)



The screenshot displays the RStudio interface. The top-left pane shows the R script editor with the following code:

```
1 library(ggplot2)
2 source("plots/formatPlot.R")
3
4 view(diamonds)
5 summary(diamonds)
6
7 summary(diamonds$price)
8 aveSize <- round(mean(diamonds$carat), 4)
9 clarity <- levels(diamonds$clarity)
10
11 p <- qplot(carat, price,
12           data=diamonds, color=clarity,
13           xlab="Carat", ylab="Price",
14           main="Diamond Pricing")
15
```

The bottom-left pane shows the console output:

```
Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
Median : 5.700   Median : 5.710   Median : 3.530
Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  326   950   2401   3933   5324  18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+           data=diamonds, color=clarity,
+           xlab="Carat", ylab="Price",
+           main="Diamond Pricing")
>
> format.plot(p, size=24)
> |
```

The right-hand side of the interface shows the Workspace pane with the 'diamonds' data frame (53940 observations) and the 'p' ggplot object. Below it, the Plots pane displays a scatter plot titled 'Diamond Pricing' showing Price (Y-axis, 0 to 15000) versus Carat (X-axis, 0.0 to 3.5). The points are colored by Clarity, with a legend on the right listing categories: I1, SI2, SI1, VS2, VS1, VVS2, VVS1, and IF.

Source : <http://rprogramming.net/download-and-install-rstudio/>

Environnement de Travail

- Dossier par défaut dans lequel R
 - ✓ Recherche des fichiers de script, des données
 - ✓ Sauvegarde l'environnement (espace) de travail dans le fichier `.RData` (donc les objets créés lors de la session en cours)
- Déterminé au lancement de R

- Fonctions utiles
 - ✓ `getwd()` : affiche le répertoire de travail en cours
 - ✓ `ls()` : liste les objets contenus dans l'espace de travail en cours
 - ✓ `setwd("monRepertoire")` : Change le répertoire de travail

Aide et Documentation

- Aide en ligne

- ✓ `?NomFonction` # `?mean`

- ✓ `help(NomFonction)` # `help(mean)`

- ✓ `??NomNonExact` # `??mea`

- ✓ `help.search("NomRecherche")` # `help.search("mea")`

- ✓ `RSiteSearch("NomRecherche")` # `RSiteSearch("mea")`

- Documentation

- ✓ Depuis R : menu Aide → Manuels (en PDF)

- ✓ Ouvrages

- ✓ Sur Internet +++

Session de Travail

- Démarrer une session R
- Ouvrir son éditeur de texte / de code R (couplé à l'ouverture de la session R selon son choix)
- Créer / Ouvrir un fichier de script
- Charger l'espace de travail spécifique, si nécessaire
- Développer son code R
 - ✓ Vérification au fur et à mesure
 - ✓ Sauvegardes
- Sauvegarder son espace de travail spécifique, si nécessaire
- Quitter (éditeur, logiciel R)

Pour en Savoir Plus...

- Effectuer l'exemple donné en Annexe A de Venables et coll. An Introduction to R
<https://cran.r-project.org/doc/manuals/R-intro.html#A-sample-session>
- Introduction à la programmation en R. Vincent Goulet
https://cran.r-project.org/doc/contrib/Goulet_introduction_programmation_R.pdf