Using Interactive Jupyter Notebooks with R

Earl F Glynn
Kansas City R Users Group
2015-12-05

http://earlglynn.github.io/kc-r-users-jupyter/
Using Interactive Jupyter Notebooks with R

- What is Jupyter?
- R User Interface Evolution
  - Command Line
  - RStudio
  - RStudio with Markdown
  - Jupyter Notebook
- Jupyter Markdown Cells
- Jupyter Code Cells
- Installation of Jupyter
What is Jupyter?

- [http://jupyter.org/](http://jupyter.org/)
- Provides interactive data science and scientific computing across ~40 programming languages
- **Julia** – **Python** – **R**
R User Interface Evolution

• R Command Line
• RStudio
• RStudio with Markdown
• Jupyter Notebook

Comparisons using \texttt{?lm} help example
R Command Line

?lm

```r
## Annette Dobson (1990) "An Introduction to Generalized Linear Models". 
## Page 9: Plant Weight Data.
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14)
trt <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03, 4.89, 4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)

lm.D9 <- lm(weight ~ group)

lm.D90 <- lm(weight ~ group - 1) # omitting intercept

anova(lm.D9)
summary(lm.D90)

opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))
plot(lm.D9, las = 1) # Residuals, Fitted, ...

par(opar)
```

Copy and paste to R console window
> ## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
> ## Page 9: Plant Weight Data.
> ctl <- c(4.17,5.58,5.18,4.1,5.99,6.52,5.68,4.81,4.62,5.21,4.36,5.22,6.07,4.56,4.24,5.34,5.32,5.56,5.26,5.16)
> trt <- c(4.01,4.17,4.41,5.59,5.87,3.83,6.03,4.69,4.32,4.69)
> group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
> weight <- c(ctl, trt)
> lm.D9 <- lm(weight ~ group)
> lm.D90 <- lm(weight ~ group - 1) # omitting intercept
> anova(lm.D9)

Analysis of Variance Table

Response: weight

DF  Sum Sq  Mean Sq  F value  Pr(>F)
   group        1   0.6862  0.68620  1.4191 0.249
Residuals  18  8.7292 0.48496
> summary(lm.D90)

Call:
  lm(formula = weight ~ group - 1)

Residuals:
   Min     1Q Median     3Q    Max
-1.0710 -0.4938  0.0605  0.2462  1.3680

Coefficients:  Estimate Std. Error   t value  Pr(>|t|)
   groupCtl  5.0320    0.2202   22.889   9.58e-15 ***
   groupTrt  4.6610    0.2202   21.168  3.62e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6966 on 18 degrees of freedom
Multiple R-squared:  0.9967,   Adjusted R-squared:  0.9963
F-statistic:  465.1 on 2 and 18 DF,  p-value: < 2.2e-16

> par <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))
> plot(lm.D9, las = 1)  # Residuals, fitted, ...
> par(opar)
RStudio

https://www.rstudio.com/products/RStudio/
RStudio

```r
# Annette Dobson (1990) "An Introduction to Generalized Linear Models"
# Page 9: Plant Weight Data.
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.00, 4.08, 5.22, 5.88, 4.14)
trt <- c(4.81, 4.17, 4.41, 5.63, 6.87, 5.33, 6.08, 4.24, 4.58, 5.83)
group <- gl(2, 10, 20, labels = c("ctl", "trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
summary(lm.D9)

anova(lm.D9)
summary(lm.D9)

par(opar)

# Residuals, Fitted, ...
plot(lm.D9, las = 1) # Residuals, Fitted, ...
par(opar)
```

---

R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under certain conditions. Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
RStudio
RStudio
RStudio with Markdown

Markdown Basics:  [http://rmarkdown.rstudio.com/authoring_basics.html](http://rmarkdown.rstudio.com/authoring_basics.html)
RStudio with Markdown

Output to HTML, PDF, Word.
Graphics output included.

**lm help example**


```r
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.50, 4.61, 5.17, 4.53, 5.33, 5.14)
trt <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.33, 6.03, 4.89, 4.32, 4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
lm.D90 <- lm(weight ~ group - 1)  # omitting intercept
anova(lm.D9)
```

```r
### Analysis of Variance Table
### Response: weight
### Df Sum Sq Mean Sq F value Pr(>F)
### group 1 0.6882 0.68820 1.4191 0.249
### Residuals 18 8.7292 0.48496
```

```r
summary(lm.D9)
```

```r
### Call
```
Jupyter Notebook

From command window in working directory, start Jupyter notebook server:

```
jupyter notebook
```
Jupyter Notebook
Jupyter Notebook

Each Jupyter cell contains Markdown or the equivalent of a Code “chunk” in RStudio.
Jupyter Notebook

Unlike RStudio/knitr, no special syntax for code chunk. Enter “Ctrl-Enter” to execute code in cell interactively. Out[1] is the R output here from cell In[1].
In [2]:
   :   `lm.D90 <- lm(weight ~ group - 1)`  # omitting intercept

   :   `anova(lm.D90)`

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>1</td>
<td>0.688205</td>
<td>0.688205</td>
<td>1.419101</td>
<td>0.2490232</td>
</tr>
<tr>
<td>Residuals</td>
<td>18</td>
<td>8.72925</td>
<td>0.4849583</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>
Jupyter Notebook

```r
options(repr.plot.width=6, repr.plot.height=6)
opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))
plot(lm.D9, las = 1)  # Residuals, Fitted, ...
par(opar)
```

- **Residuals vs Fitted**
  - Residuals vs Fitted values.

- **Normal Q-Q**
  - Normal Quantiles vs Theoretical Quantiles.

- **Scale-Location**
  - Standardized Residuals vs Fitted values.

- **Constant Leverage: Residuals vs Factor Levels**
  - Residuals vs Factor Level Combinations.
## Statistical Properties of Co-occurrence Matrix

[Miyamoto 2008](http://www.inf.ethz.ch/personal/markusp/teaching/18-799B-CMU-spring05/material/eizan-tad.pdf) describes computation of a number of statistical properties of the co-occurrence matrix first, followed by computation of the Haralick texture features.

\[ R = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P(i,j) \]  

A sum of all elements of co-occurrence frequency matrix

\[ p(i,j) = \frac{P(i,j)}{R} \]  

An element of co-occurrence probability matrix

\[ p_x(i) = \sum_{j=1}^{N_g} p(i,j) \]  

The \( i^{th} \) entry in the marginal-probability matrix obtained by summing the rows of \( p(i,j) \).

\[ p_y(j) = \sum_{i=1}^{N_g} p(i,j) \]  

The \( j^{th} \) entry in the marginal-probability matrix obtained by summing the columns of \( p(i,j) \).
Statistical Properties of Co-occurrence Matrix

Miyamoto 2003 describes computation of a number of statistical properties of the co-occurrence matrix first, followed by computation of the Haralick texture features.

\[ R = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} p(i,j) = \text{sum of all elements of co-occurrence frequency matrix} \]

\[ p(i,j) = \frac{p_{ij}}{R} = \text{co-occurrence probability matrix} \]

\[ p_x(i) = \sum_{j=1}^{N_y} p(i,j) = i^{th} \text{ entry in the marginal probability matrix obtained by summing the rows of } p(i,j) \]

\[ p_y(j) = \sum_{i=1}^{N_x} p(i,j) = j^{th} \text{ entry in the marginal probability matrix obtained by summing the columns of } p(i,j) \]
Jupyter Code Cells

Online Examples:
http://earlglynn.github.io/kc-r-users-jupyter/

• Jupyter First Look

• Im help example
• Co-occurrence Matrix

• Exploring Kaggle Facial Keypoints Detection Data
Installation of Jupyter

Perhaps easiest:
Install Anaconda Python from Continuum Analytics

https://www.continuum.io/downloads

• Python 3.5, Windows 64-bit graphical installer
• Package List: http://docs.continuum.io/anaconda/pkg-docs
  – Includes: numpy, scipy, scikit-learn, matplotlib, ...

Installation of Jupyter

From command prompt:

• Conda: `conda update conda`
• Jupyter: `conda install jupyter`
• R Essentials: `conda install -c r r-essentials`
• R Kernel:
  `conda install -c r ipython-notebook r-irkernel`

http://irkernel.github.io/installation/
https://www.continuum.io/blog/developer/jupyter-and-conda-r
R Packages Used by Jupyter

```
In [1]: .libPaths()
Out[1]: "C:/Users/Earl/Documents/R/win-library/3.1"  "C:/Anaconda3/R/library"
```

```
In [2]: library()
```

Packages in library 'C:/Anaconda3/R/library':

- base: The R Base Package
- base64enc: Tools for base64 encoding
- boot: Bootstrap Functions (Originally by Angelo Canty for S)
- class: Functions for Classification
- cluster: Cluster Analysis Extended Rousseeuw et al.
- codetools: Code Analysis Tools for R
- compiler: The R Compiler Package
- datasets: The R Datasets Package

...
Installation of Jupyter

Kernels for other languages:
https://github.com/ipython/ipython/wiki/IPython-kernels-for-other-languages
Take Home Message

Jupyter is a great way to use R interactively to document the steps in a data analysis project.

Jupyter’s interactive approach is better (IMHO) than the batch processing by RStudio/knitr to document reproducible results.