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/
- Language-agnostic parts of IPython ("Interactive Python") http://ipython.org/
- Provides interactive data science and scientific computing across ~40 programming languages
- **Ju**lia – **Pyt**hon – **R**

R User Interface Evolution

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

Comparisons using `?lm` help example

R Command Line

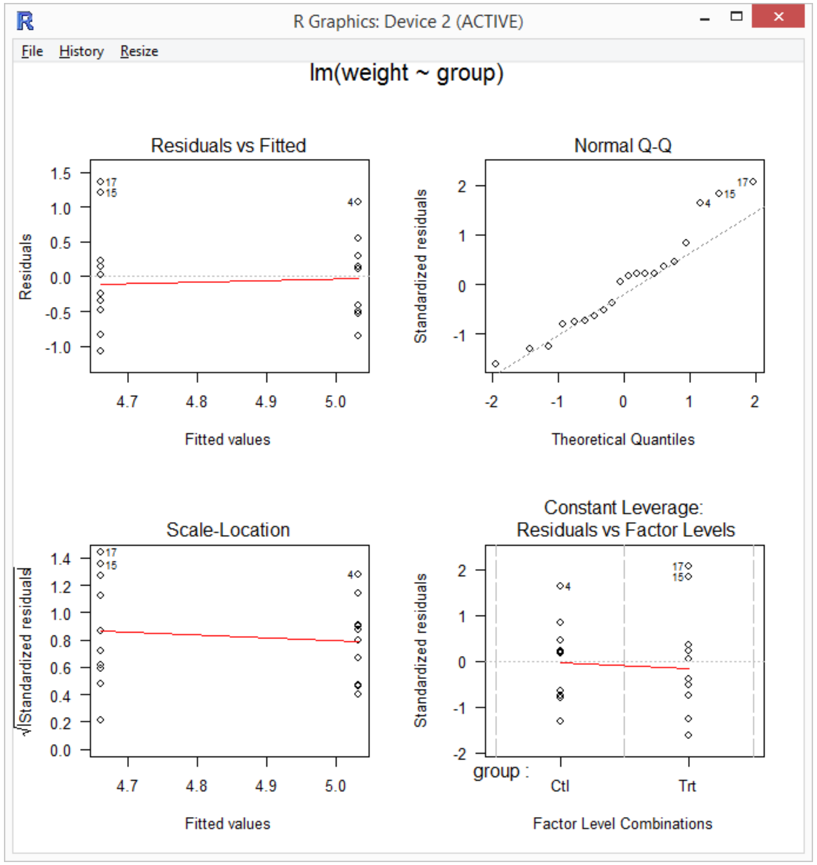
?lm

```
## 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(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
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)
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)
```

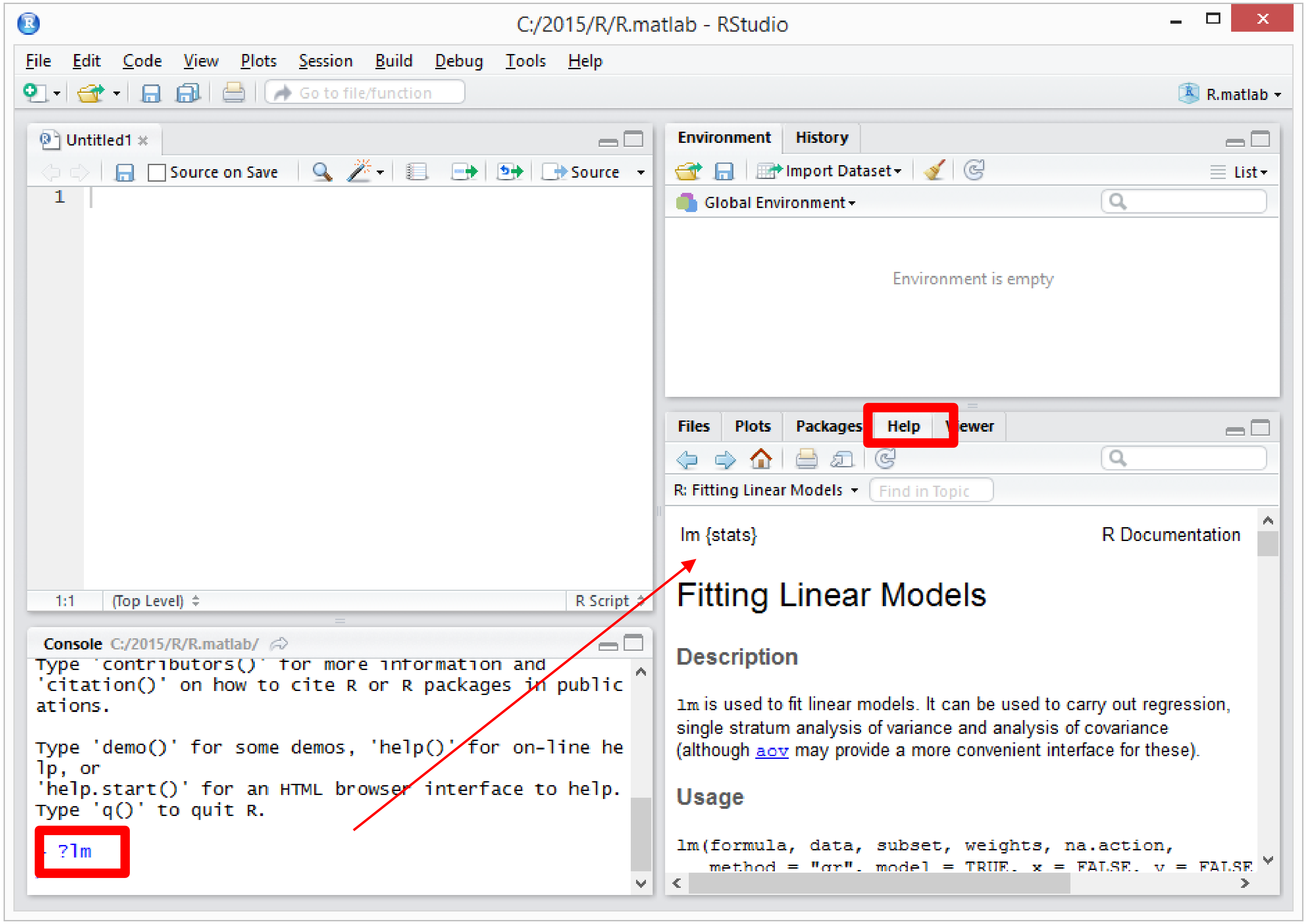
Copy and paste to R console window

R Command Line

```
> ## 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(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
> 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 0.6882 0.68820  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  0.0685  0.2462  1.3690

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
groupCtl   5.0320     0.2202   22.85 9.55e-15 ***
groupTrt   4.6610     0.2202   21.16 3.62e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6964 on 18 degrees of freedom
Multiple R-squared:  0.9818,    Adjusted R-squared:  0.9798
F-statistic: 485.1 on 2 and 18 DF,  p-value: < 2.2e-16

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



RStudio



```
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in public
ations.

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

> ?lm
```

lm {stats} R Documentation

Fitting Linear Models

Description

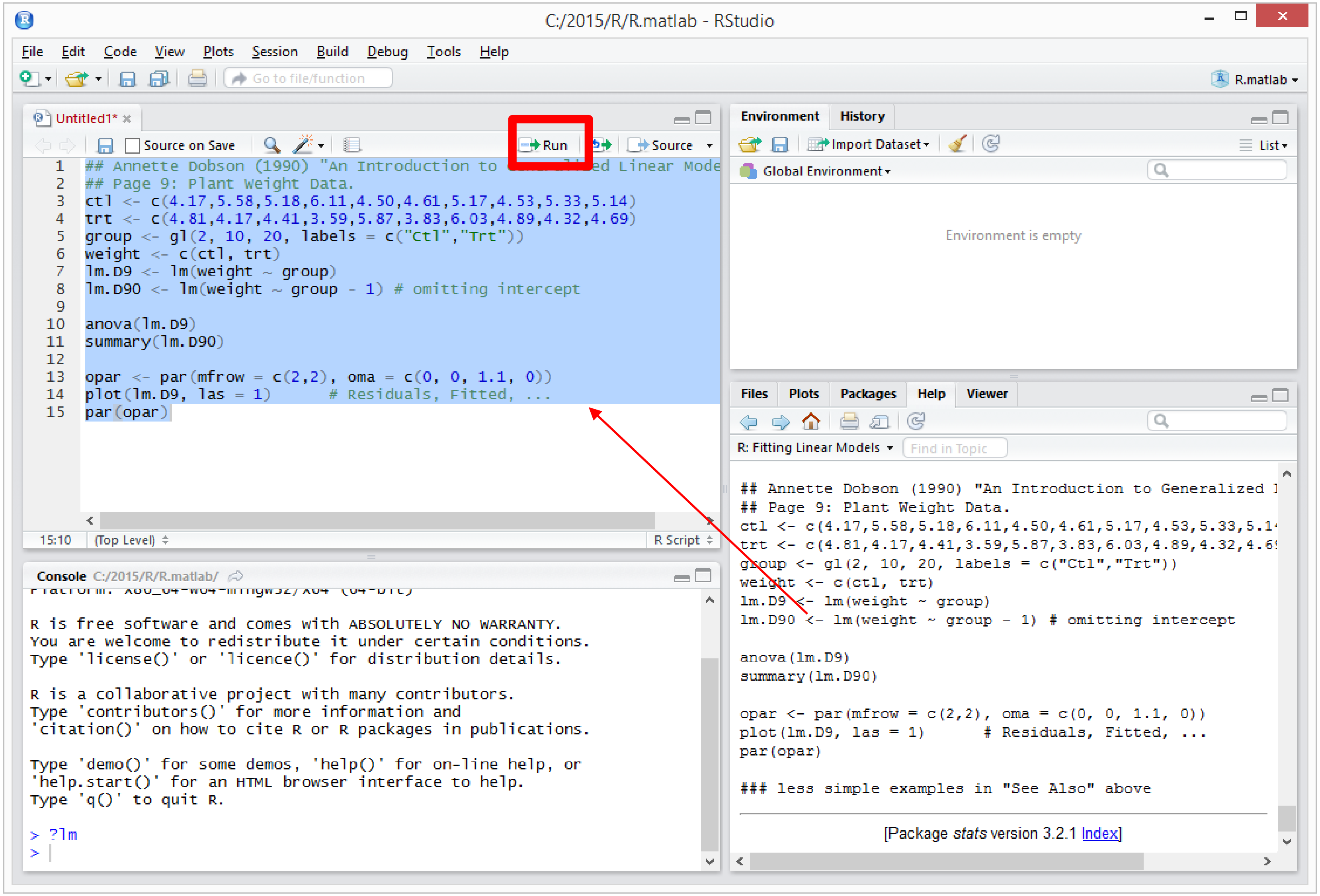
`lm` is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although `aov` may provide a more convenient interface for these).

Usage

```
lm(formula, data, subset, weights, na.action,
   method = "qr", model = TRUE, x = FALSE, y = FALSE
```

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

RStudio



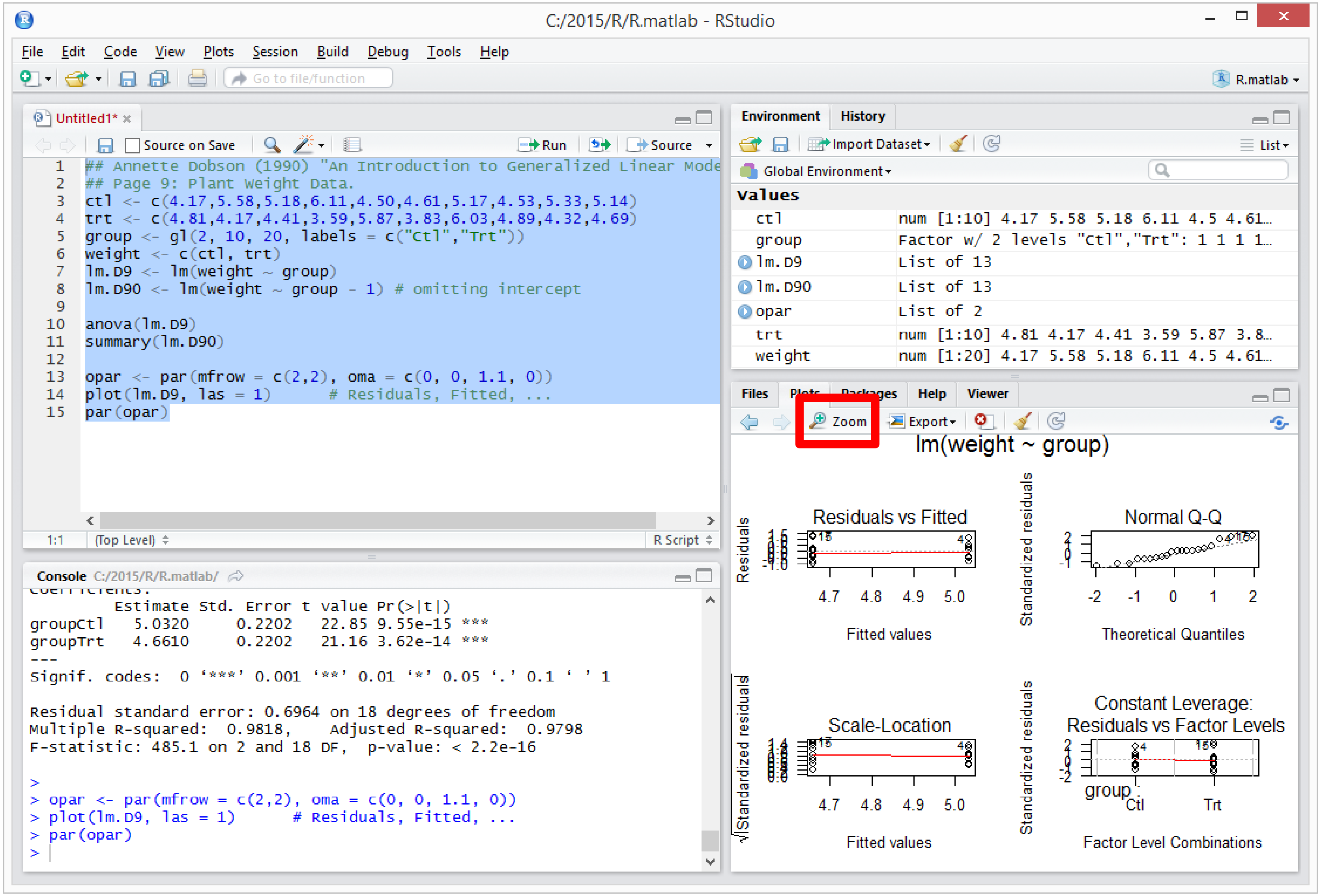
```
## Annette Dobson (1990) "An Introduction to Generalized Linear Mode
## Page 9: Plant Weight Data.
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
lm.D90 <- lm(weight ~ group - 1) # omitting intercept

anova(lm.D9)
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)
```

```
Platform: x86_64-w64-mingw32/x64 (64-bit)

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.

> ?lm
> 
```

```
## Annette Dobson (1990) "An Introduction to Generalized L
## Page 9: Plant Weight Data.
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.1
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.6
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)

### less simple examples in "See Also" above
```

[Package *stats* version 3.2.1 Index]

RStudio

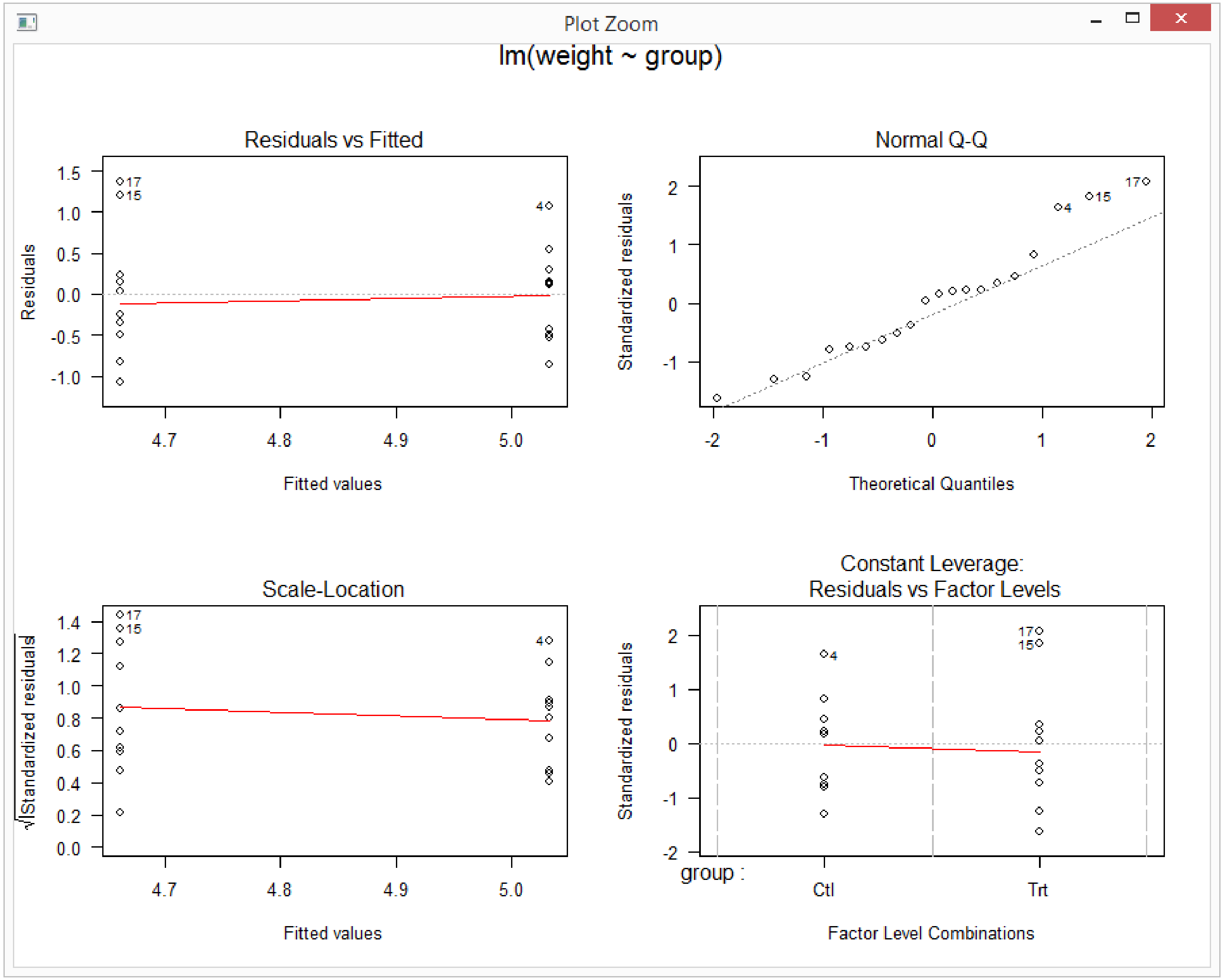


```r
## Annette Dobson (1990) "An Introduction to Generalized Linear Mode
## Page 9: Plant Weight Data.
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
lm.D90 <- lm(weight ~ group - 1) # omitting intercept

anova(lm.D9)
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)
```

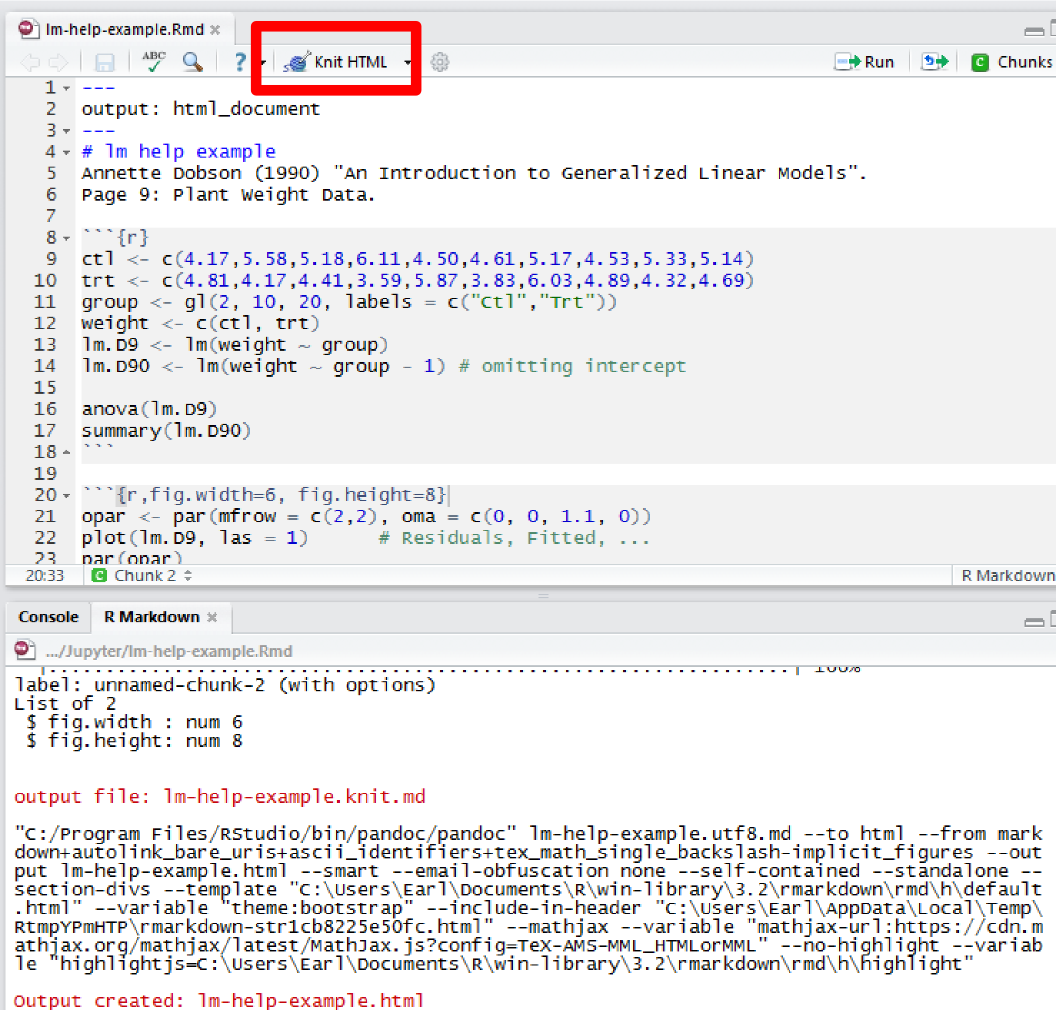
```
ctl      num [1:10] 4.17 5.58 5.18 6.11 4.5 4.61...
group    Factor w/ 2 levels "Ctl","Trt": 1 1 1 1...
lm.D9    List of 13
lm.D90   List of 13
opar     List of 2
trt      num [1:10] 4.81 4.17 4.41 3.59 5.87 3.8...
weight   num [1:20] 4.17 5.58 5.18 6.11 4.5 4.61...
```

```
Coefficients:
         Estimate Std. Error t value Pr(>|t|)
groupCtl   5.0320     0.2202   22.85 9.55e-15 ***
groupTrt   4.6610     0.2202   21.16 3.62e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6964 on 18 degrees of freedom
Multiple R-squared:  0.9818,    Adjusted R-squared:  0.9798
F-statistic: 485.1 on 2 and 18 DF,  p-value: < 2.2e-16

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

RStudio


Plot Zoom
lm(weight ~ group)
Residuals vs Fitted
Residuals
1.5
1.0
0.5
0.0
-0.5
-1.0
17
15
4
4.7
4.8
4.9
5.0
Fitted values
Normal Q-Q
Standardized residuals
2
1
0
-1
17
15
4
-2
-1
0
1
2
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
17
15
4
4.7
4.8
4.9
5.0
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
2
1
0
-1
-2
4
17
15
group :
Ctl
Trt
Factor Level Combinations

RStudio with Markdown



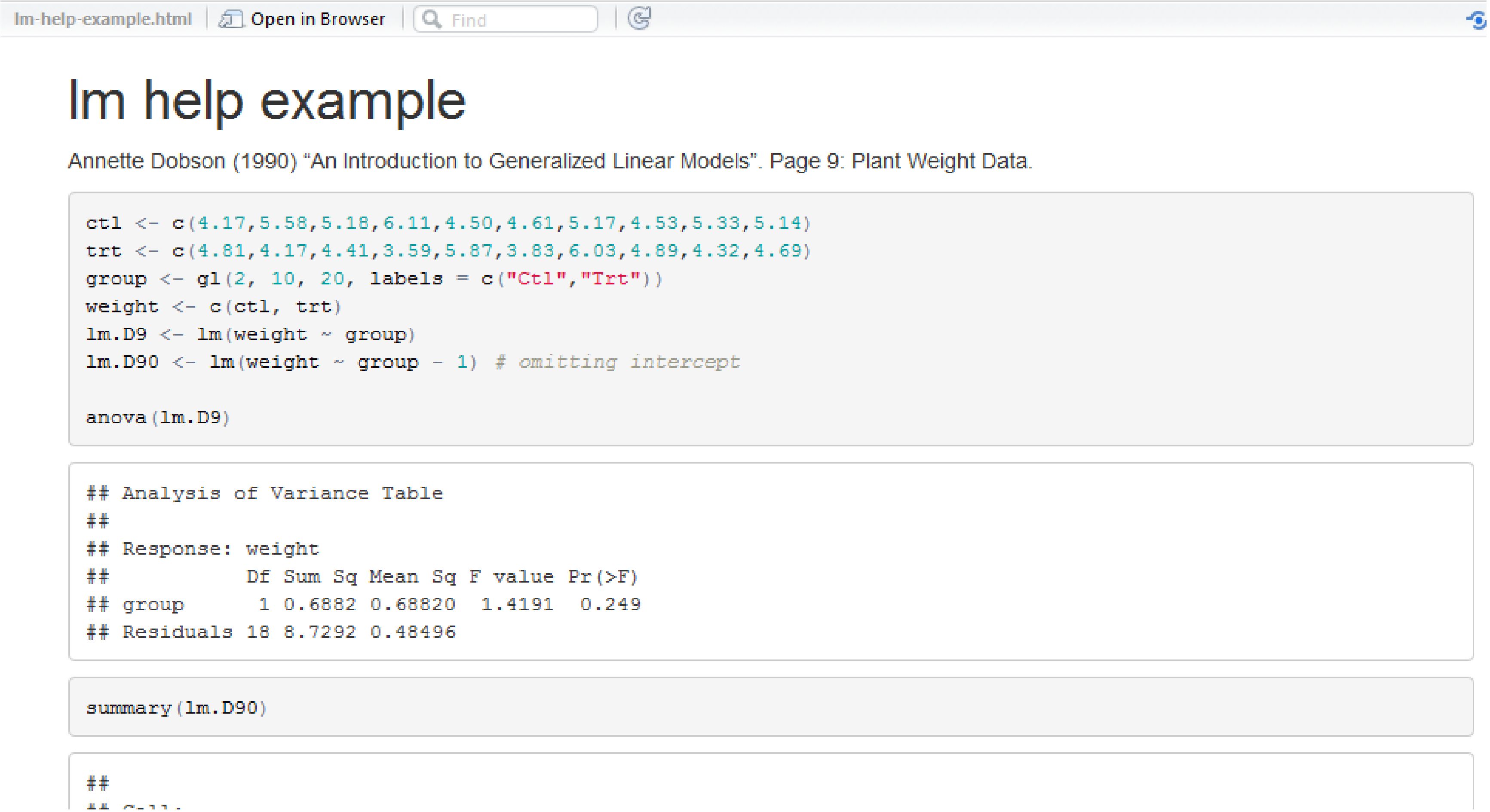
~~~
---
output: html_document
---
# lm help example
Annette Dobson (1990) "An Introduction to Generalized Linear Models".
Page 9: Plant Weight Data.

```{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(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
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)
summary(lm.D90)
```

```{r,fig.width=6, fig.height=8}
opar <- par(mfrow = c(2,2), oma = c(0, 0, 1.1, 0))
plot(lm.D9, las = 1)      # Residuals, Fitted, ...
par(opar)
~~~

```
label: unnamed-chunk-2 (with options)
List of 2
 $ fig.width : num 6
 $ fig.height: num 8


output file: lm-help-example.knit.md

"C:/Program Files/RStudio/bin/pandoc/pandoc" lm-help-example.utf8.md --to html --from mark
down+autolink_bare_uris+ascii_identifiers+tex_math_single_backslash-implicit_figures --out
put lm-help-example.html --smart --email-obfuscation none --self-contained --standalone --
section-divs --template "C:\Users\Earl\Documents\R\win-library\3.2\rmarkdown\rmd\h\default
.html" --variable "theme:bootstrap" --include-in-header "C:\Users\Earl\AppData\Local\Temp\
RtmpYPmHTP\rmarkdown-str1cb8225e50fc.html" --mathjax --variable "mathjax-url:https://cdn.m
athjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML" --no-highlight --variab
le "highlightjs=C:\Users\Earl\Documents\R\win-library\3.2\rmarkdown\rmd\h\highlight"

Output created: lm-help-example.html
```

Markdown Basics: http://rmarkdown.rstudio.com/authoring_basics.html

RStudio with Markdown

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



lm help example

Annette Dobson (1990) “An Introduction to Generalized Linear Models”. Page 9: Plant Weight Data.

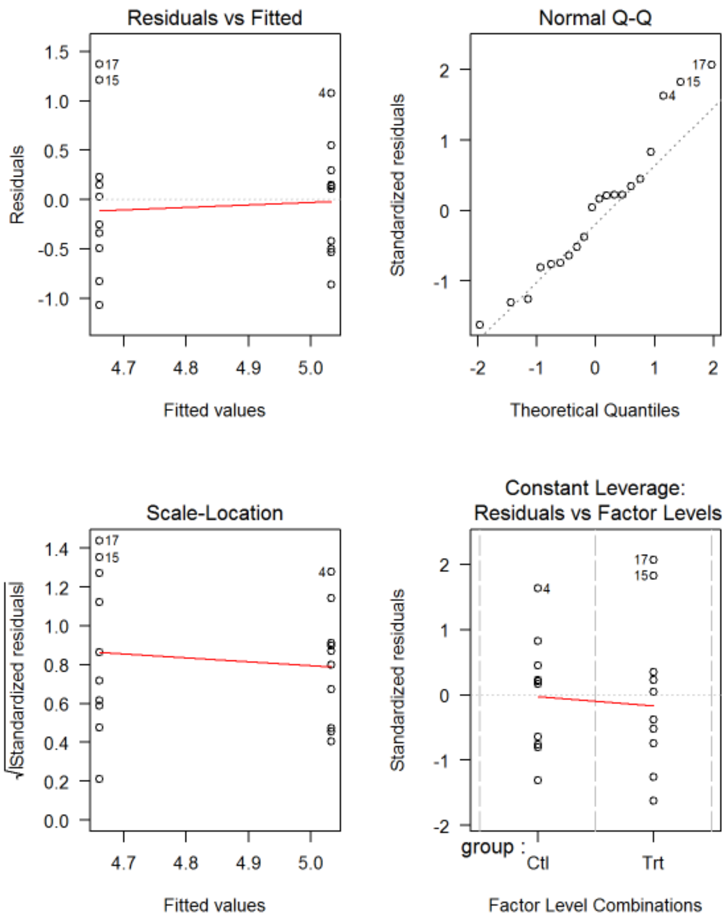
```
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
lm.D90 <- lm(weight ~ group - 1) # omitting intercept

anova(lm.D9)
```

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

```
summary(lm.D90)
```

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

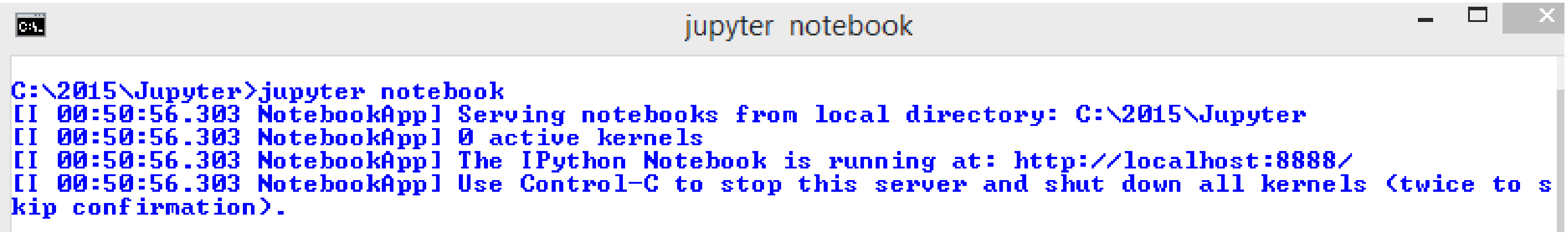
RStudio with Markdown



Jupyter Notebook

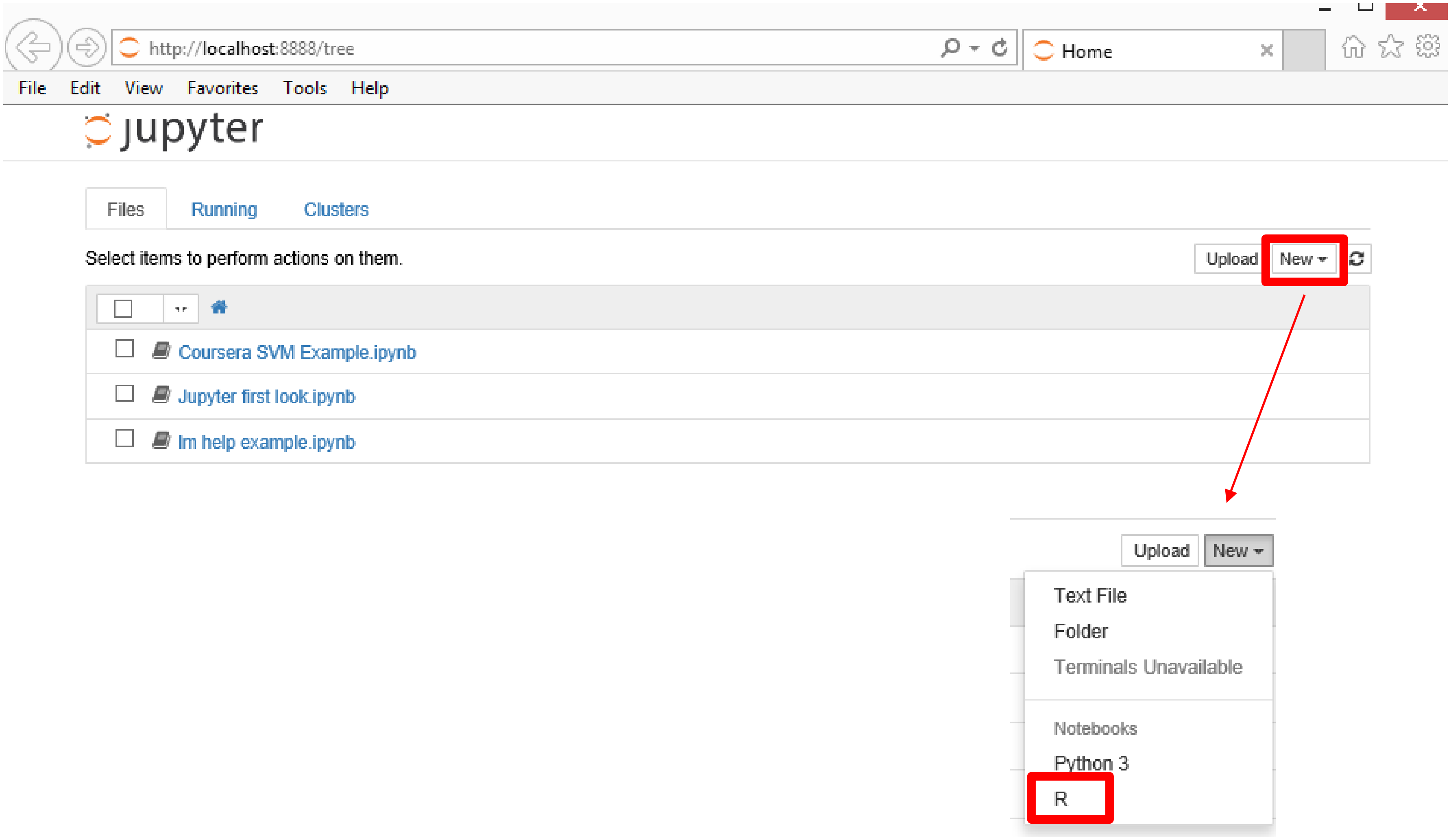
From command window in working directory, start Jupyter notebook server:

```
jupyter notebook
```



```
C:\2015\Jupyter>jupyter notebook
[I 00:50:56.303 NotebookApp] Serving notebooks from local directory: C:\2015\Jupyter
[I 00:50:56.303 NotebookApp] 0 active kernels
[I 00:50:56.303 NotebookApp] The IPython Notebook is running at: http://localhost:8888/
[I 00:50:56.303 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to s
kip confirmation).
```

Jupyter Notebook



http://localhost:8888/tree

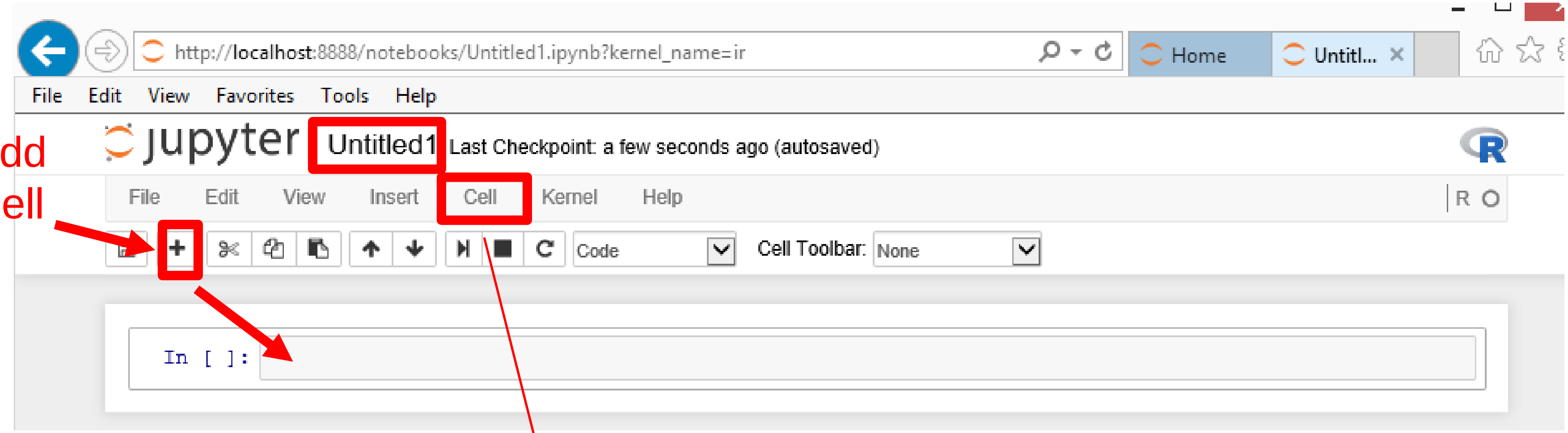
Select items to perform actions on them.

□ Coursera SVM Example.ipynb

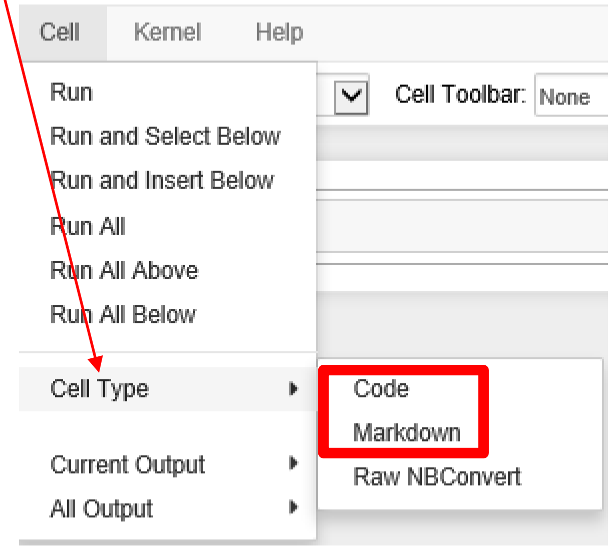
□ Jupyter first look.ipynb

□ lm help example.ipynb

Jupyter Notebook

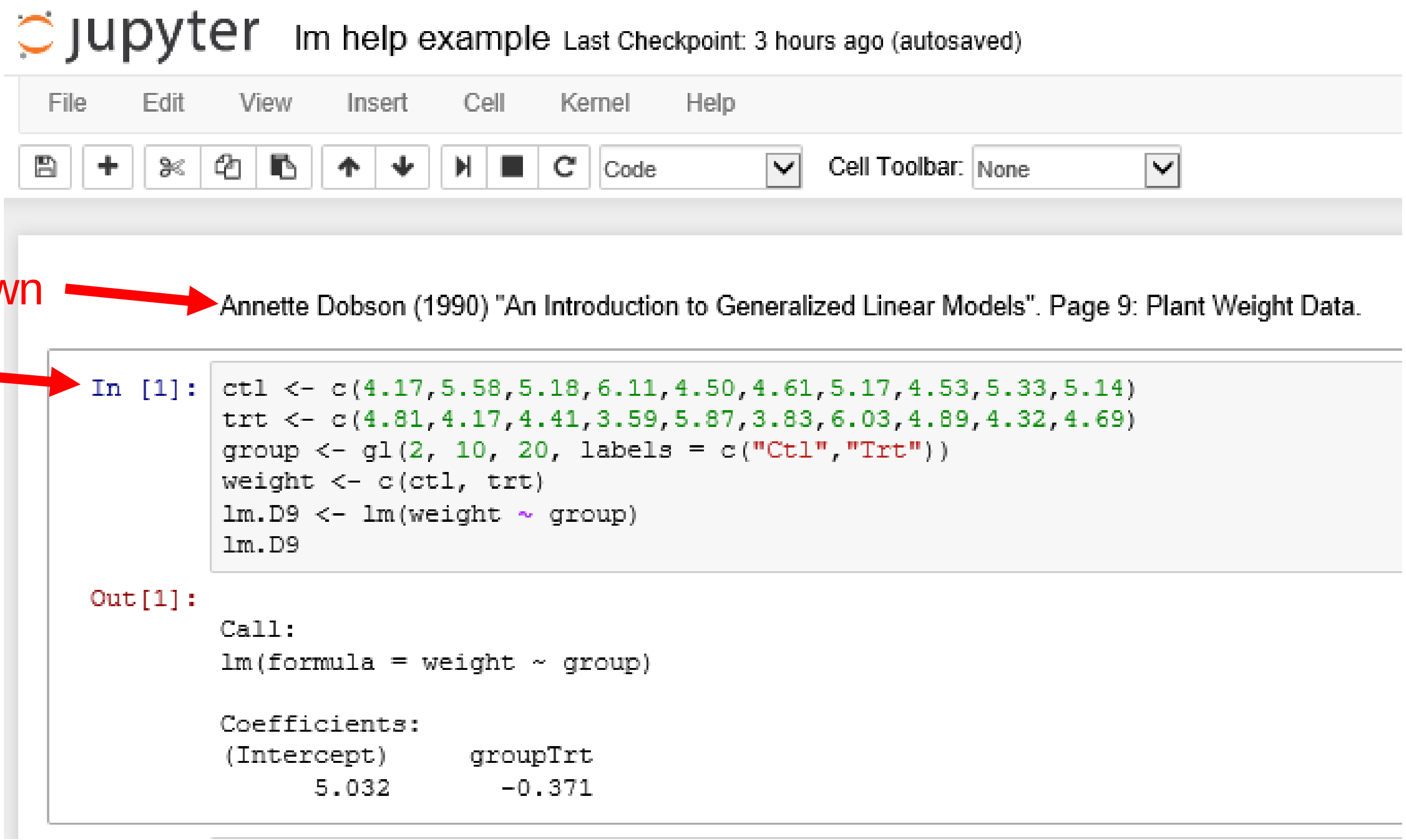


Add Cell



Each Jupyter cell contains Markdown or the equivalent of a Code "chunk" in RStudio

Jupyter Notebook



lm help example Last Checkpoint: 3 hours ago (autosaved)

Code

Cell Toolbar: None

Markdown

Annette Dobson (1990) "An Introduction to Generalized Linear Models". Page 9: Plant Weight Data.

Code

In [1]:

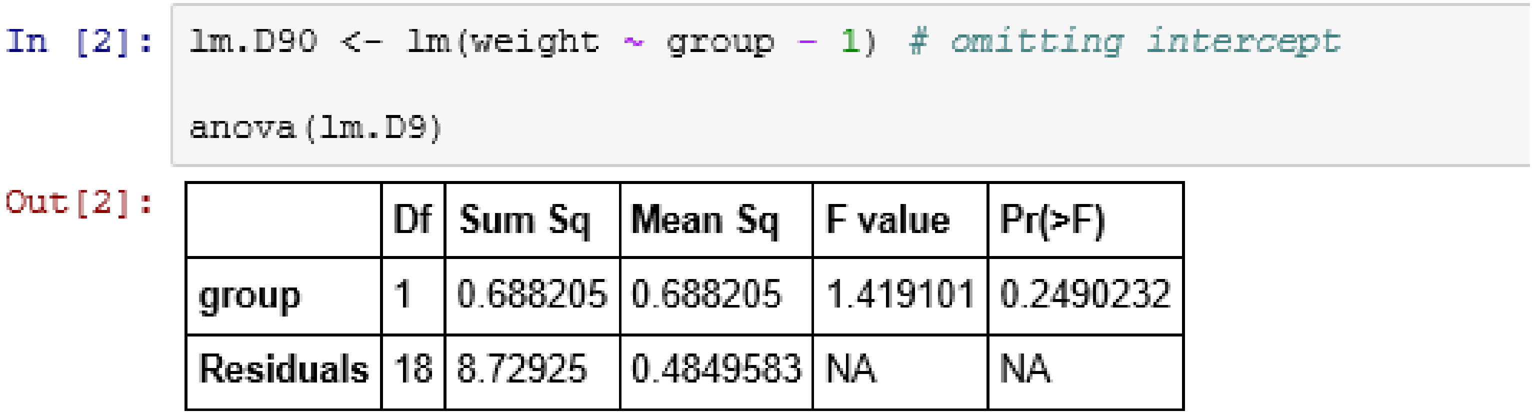
```
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
lm.D9
```

Out[1]:

```
Call:
lm(formula = weight ~ group)

Coefficients:
(Intercept)     groupTrt
      5.032       -0.371
```

Unlike RStudio/knittr, 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].

Jupyter Notebook



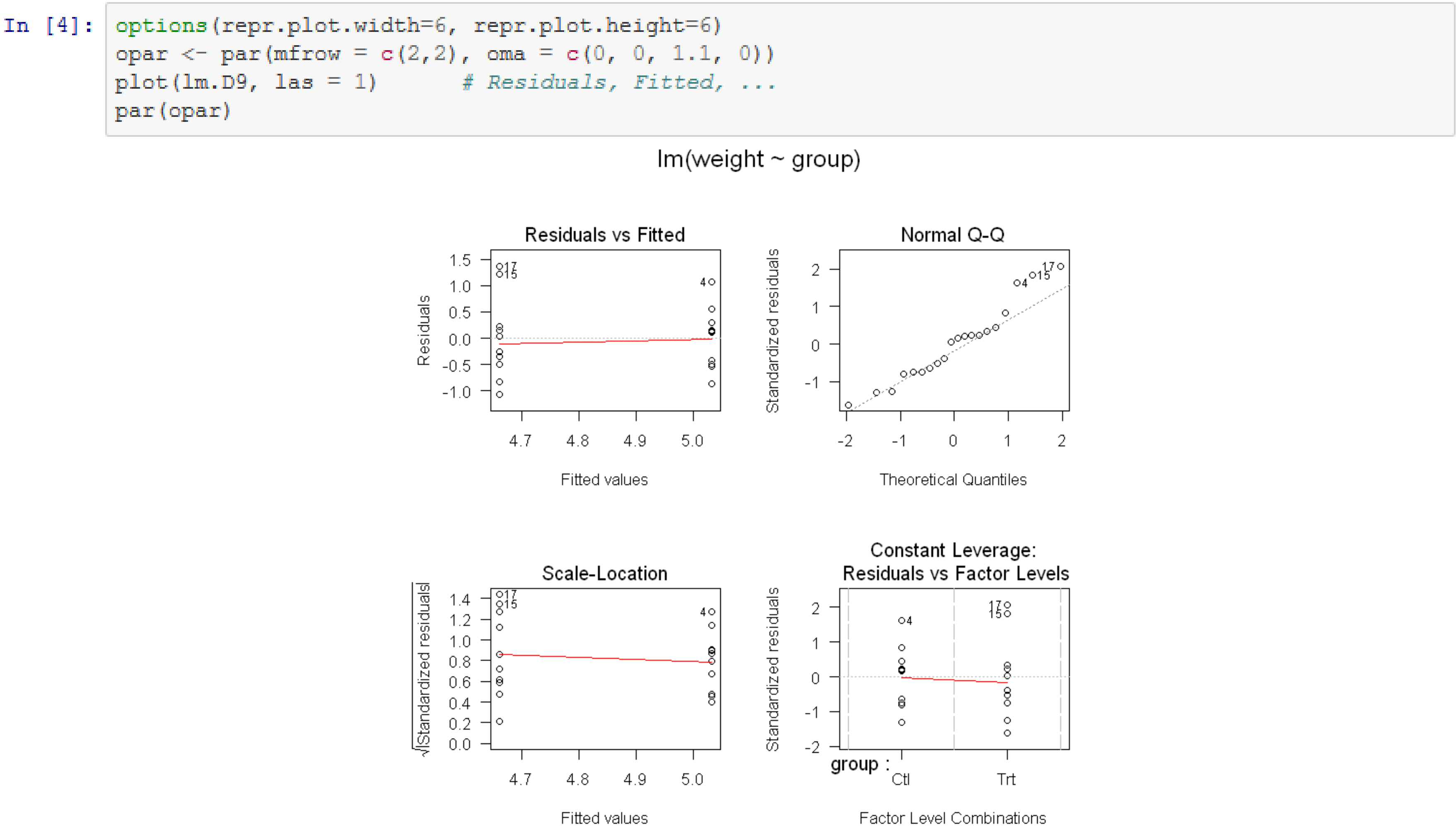
In [2]:

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

anova(lm.D9)
```

Out[2]:

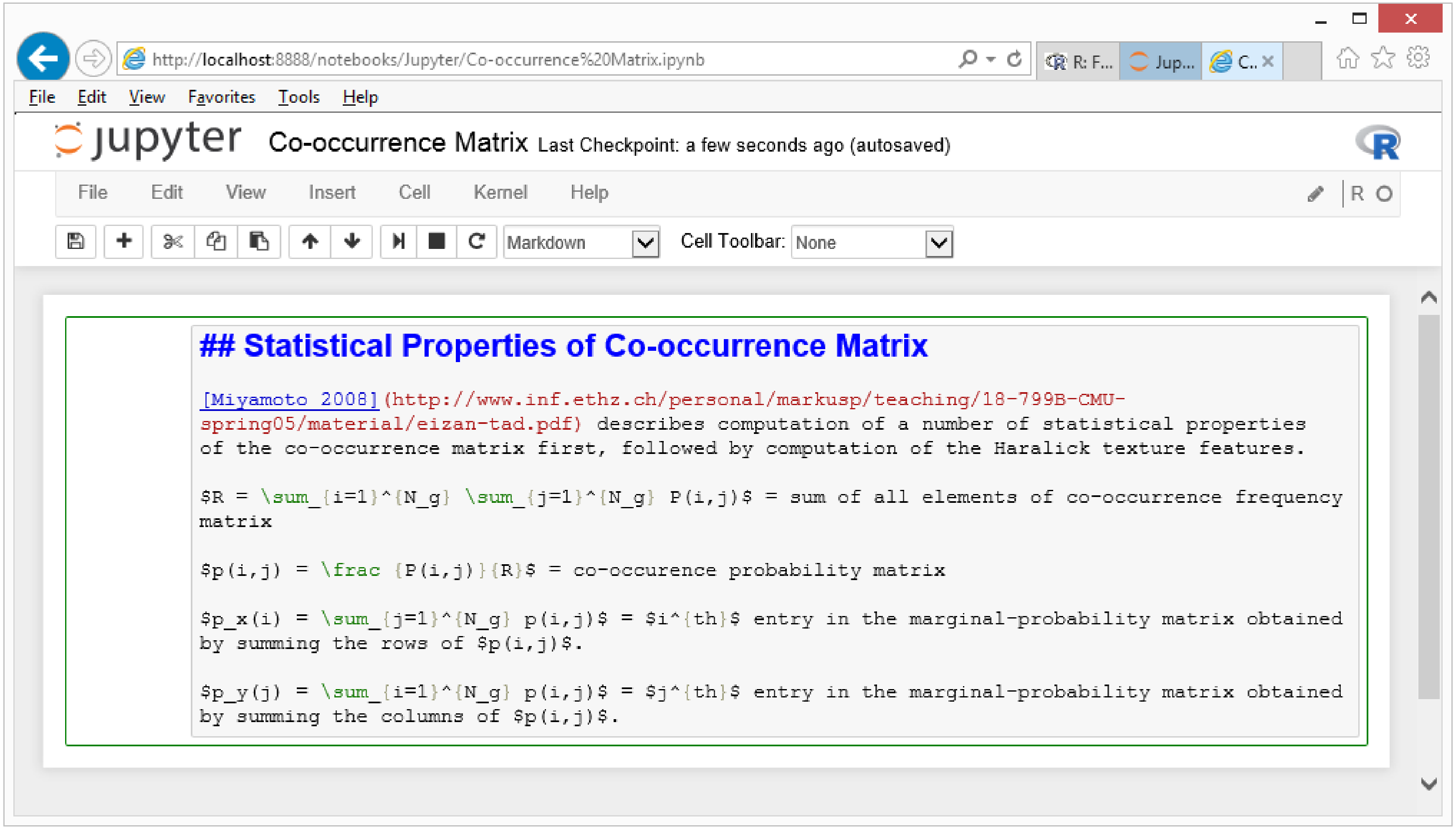
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
group	1	0.688205	0.688205	1.419101	0.2490232
Residuals	18	8.72925	0.4849583	NA	NA

Jupyter Notebook



```
In [4]: 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)
```

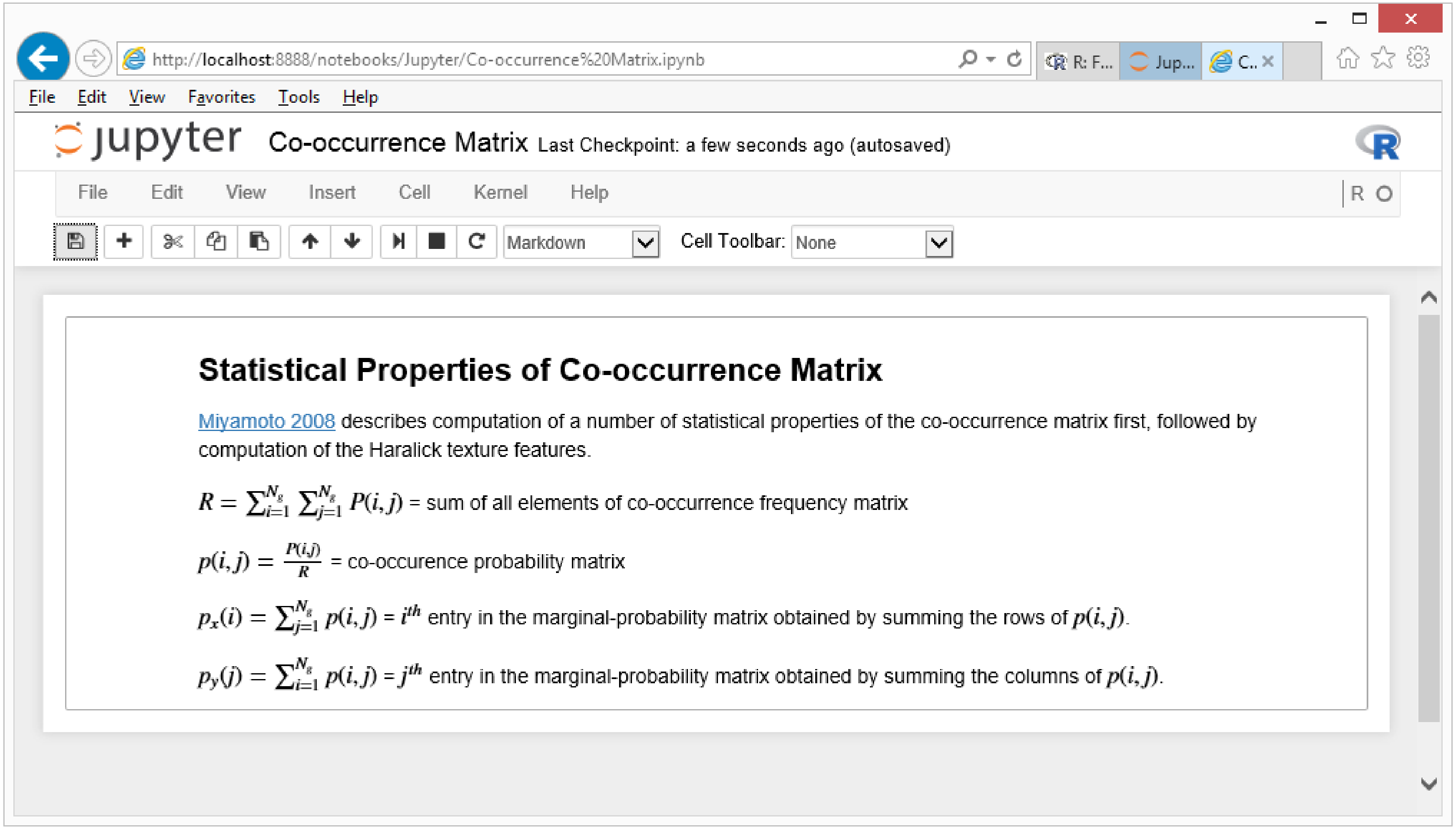
Jupyter Markdown Cells



```
## 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.

$R = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P(i,j)$ = sum of all elements of co-occurrence frequency
matrix

$p(i,j) = \frac {P(i,j)}{R}$ = co-occurence probability matrix

$p_x(i) = \sum_{j=1}^{N_g} p(i,j)$ = $i^{th}$ entry in the marginal-probability matrix obtained
by summing the rows of $p(i,j)$.

$p_y(j) = \sum_{i=1}^{N_g} p(i,j)$ = $j^{th}$ entry in the marginal-probability matrix obtained
by summing the columns of $p(i,j)$.
```

Markdown example including inline LaTeX equations. *Ctrl-Enter* to render.

Jupyter Markdown Cells



Statistical Properties of Co-occurrence Matrix

Miyamoto 2008 describes computation of a number of statistical properties of the co-occurrence matrix first, followed by computation of the Haralick texture features.

$R = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P(i,j)$ = sum of all elements of co-occurrence frequency matrix

$p(i,j) = \frac{P(i,j)}{R}$ = co-occurence probability matrix

$p_x(i) = \sum_{j=1}^{N_g} p(i,j)$ = i^{th} entry in the marginal-probability matrix obtained by summing the rows of $p(i,j)$.

$p_y(j) = \sum_{i=1}^{N_g} p(i,j)$ = j^{th} 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

- lm 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:

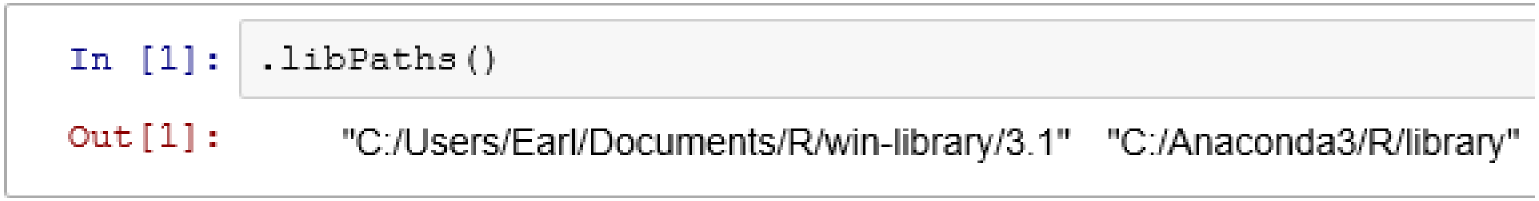
  ```
  conda install -c r r-essentials
  ```
- R Kernel:

  ```
  conda install -c r ipython-notebook r-irkernel
  ```

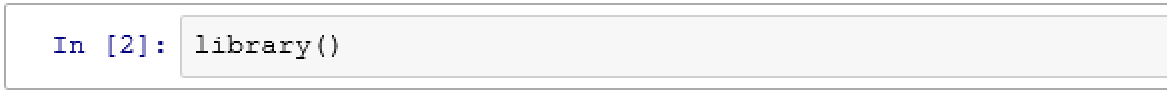
  http://irkernel.github.io/installation/

  https://www.continuum.io/blog/developer/jupyter-and-conda-r

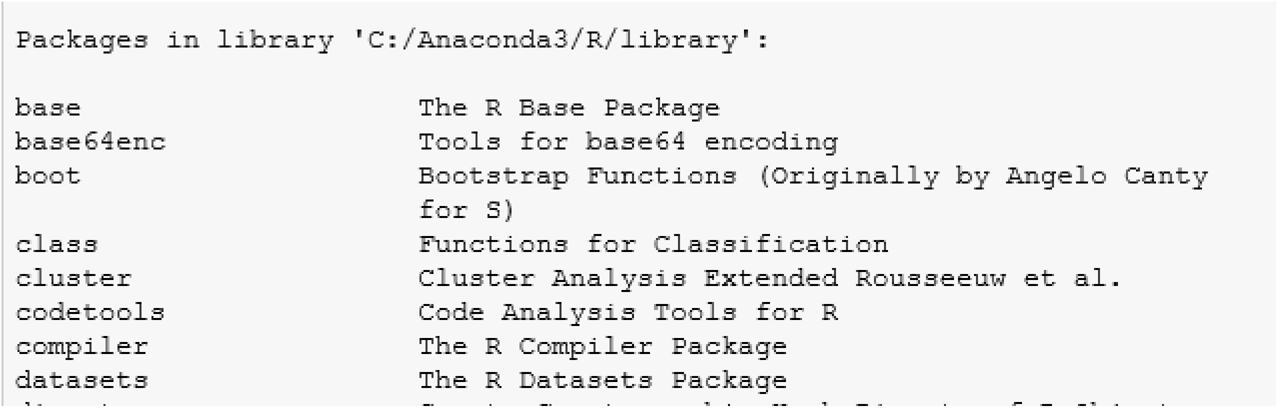
R Packages Used by Jupyter



```
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.