# Markdown in Stata

## 1. What is markdown?

Markdown is a markup language that can be written using a plain text editor. It allows plain text content to be formatted with simple markup syntax.

## 2. Why should we study markdown?

* Markdown's syntax is easy to learn, and its feature is more powerful than plain text.
* It is easy to turn markdown into ppt,pdf and a word document.

## 3. What can we use markdown to do?

### 3.1 make a to-do list

* [ ] read a book
* [x] do homework
* [x] watch a movie

### 3.2 write formula

## 4. How to use markdown?

### 4.1 title

H1 :# Header 1

H2 :## Header 2

H3 :### Header 3

H4 :#### Header 4

H5 :##### Header 5

H6 :###### Header 6

### 4.2 Italic and bold

*ltalic* **bold**

## 5. How to combine markdown and stata?

### 5.1 Requirements

The command uses an external Markdown processor, John MacFarlane's **pandoc**, which can be downloaded for Linux, Mac or Windows from *pandoc.org/installing*. Generating Word documents requires Pandoc2.0 or higher. It also requires the Stata command **whereis**, available from SSC. This command is used to keep track of ancillary programs and is usually installed together with **markstat**. After downloading pandoc, you save the location of the executable in the whereis database by running the command *whereis pandoc* location.

If you want to generate PDF output you also need LaTeX, specifically **pdflatex**, which comes with MiKTeX on Windows, MacTeX on Macs or Live TeX on Linux. You save the location of the converter by running the command *whereis pdflatex* location. This is also used for Beamer presentations.

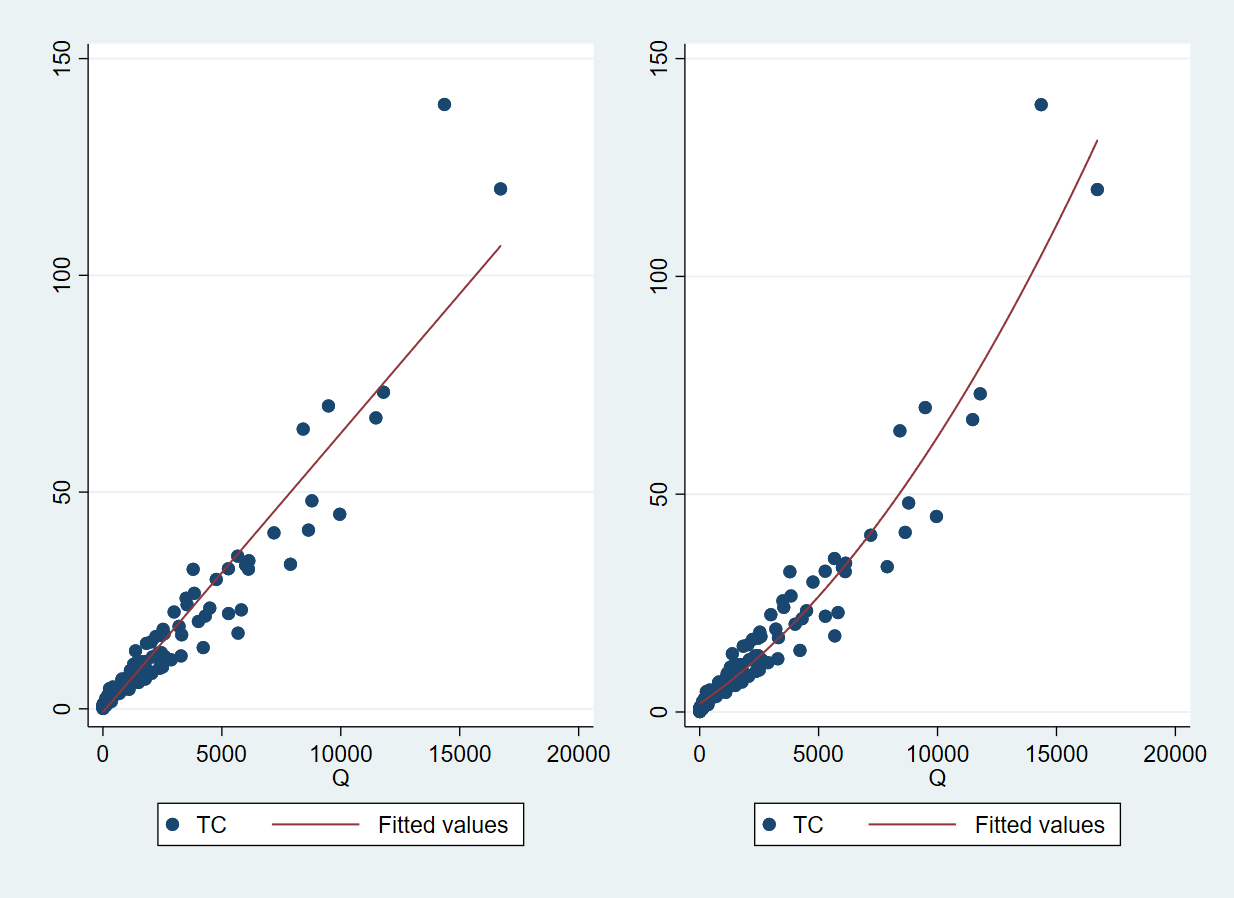
### 5.2 code

whereis pandoc "C:15.exe" whereis pdflatex "C:152.964.exe" markstat using filename [, pdf docx slides beamer mathjax bundle bibliography strict nodo nor keep]

## 6. example

### 6.1 Nerlove

. clear  
  
. import excel "C:\Users\Sabrina\Desktop\econometrics\nerlove.xls", firstrow cl  
> ear   
  
. la data "Nerlove 1963 paper"  
  
. describe  
  
Contains data  
 obs: 145 Nerlove 1963 paper  
 vars: 5   
 size: 4,060   
───────────────────────────────────────────────────────────────────────────────  
 storage display value  
variable name type format label variable label  
───────────────────────────────────────────────────────────────────────────────  
TC double %10.0g TC  
Q int %10.0g Q  
PL double %10.0g PL  
PF double %10.0g PF  
PK int %10.0g PK  
───────────────────────────────────────────────────────────────────────────────  
Sorted by:   
 Note: Dataset has changed since last saved.  
  
. list TC Q  
  
 ┌─────────────────┐  
 │ TC Q │  
 ├─────────────────┤  
 1. │ .082 2 │  
 2. │ .661 3 │  
 3. │ .99 4 │  
 4. │ .315 4 │  
 5. │ .197 5 │  
 ├─────────────────┤  
 6. │ .098 9 │  
 7. │ .949 11 │  
 8. │ .675 13 │  
 9. │ .525 13 │  
 10. │ .501 22 │  
 ├─────────────────┤  
 11. │ 1.194 25 │  
 12. │ .67 25 │  
 13. │ .349 35 │  
 14. │ .423 39 │  
 15. │ .501 43 │  
 ├─────────────────┤  
 16. │ .55 63 │  
 17. │ .795 68 │  
 18. │ .664 81 │  
 19. │ .705 84 │  
 20. │ .903 73 │  
 ├─────────────────┤  
 21. │ 1.504 99 │  
 22. │ 1.615 101 │  
 23. │ 1.127 119 │  
 24. │ .718 120 │  
 25. │ 2.414 122 │  
 ├─────────────────┤  
 26. │ 1.13 130 │  
 27. │ .992 138 │  
 28. │ 1.554 149 │  
 29. │ 1.225 196 │  
 30. │ 1.565 197 │  
 ├─────────────────┤  
 31. │ 1.936 209 │  
 32. │ 3.154 214 │  
 33. │ 2.599 220 │  
 34. │ 3.298 234 │  
 35. │ 2.441 235 │  
 ├─────────────────┤  
 36. │ 2.031 253 │  
 37. │ 4.666 279 │  
 38. │ 1.834 290 │  
 39. │ 2.072 290 │  
 40. │ 2.039 295 │  
 ├─────────────────┤  
 41. │ 3.398 299 │  
 42. │ 3.083 324 │  
 43. │ 2.344 333 │  
 44. │ 2.382 338 │  
 45. │ 2.657 353 │  
 ├─────────────────┤  
 46. │ 1.705 353 │  
 47. │ 3.23 416 │  
 48. │ 5.049 420 │  
 49. │ 3.814 456 │  
 50. │ 4.58 484 │  
 ├─────────────────┤  
 51. │ 4.358 516 │  
 52. │ 4.714 550 │  
 53. │ 4.357 563 │  
 54. │ 3.919 566 │  
 55. │ 3.442 592 │  
 ├─────────────────┤  
 56. │ 4.898 671 │  
 57. │ 3.584 696 │  
 58. │ 5.535 719 │  
 59. │ 4.406 742 │  
 60. │ 4.289 795 │  
 ├─────────────────┤  
 61. │ 6.731 800 │  
 62. │ 6.895 808 │  
 63. │ 5.112 811 │  
 64. │ 5.141 855 │  
 65. │ 5.72 860 │  
 ├─────────────────┤  
 66. │ 4.691 909 │  
 67. │ 6.832 913 │  
 68. │ 4.813 924 │  
 69. │ 6.754 984 │  
 70. │ 5.127 991 │  
 ├─────────────────┤  
 71. │ 6.388 1000 │  
 72. │ 4.509 1098 │  
 73. │ 7.185 1109 │  
 74. │ 6.8 1118 │  
 75. │ 7.743 1122 │  
 ├─────────────────┤  
 76. │ 7.968 1137 │  
 77. │ 8.858 1156 │  
 78. │ 8.588 1166 │  
 79. │ 6.449 1170 │  
 80. │ 8.488 1215 │  
 ├─────────────────┤  
 81. │ 8.877 1279 │  
 82. │ 10.274 1291 │  
 83. │ 6.024 1290 │  
 84. │ 8.258 1331 │  
 85. │ 13.376 1373 │  
 ├─────────────────┤  
 86. │ 10.69 1420 │  
 87. │ 8.308 1474 │  
 88. │ 6.082 1497 │  
 89. │ 9.284 1545 │  
 90. │ 10.879 1649 │  
 ├─────────────────┤  
 91. │ 8.477 1668 │  
 92. │ 6.877 1782 │  
 93. │ 15.106 1831 │  
 94. │ 8.031 1833 │  
 95. │ 8.082 1838 │  
 ├─────────────────┤  
 96. │ 10.866 1787 │  
 97. │ 8.596 1918 │  
 98. │ 8.673 1930 │  
 99. │ 15.437 2028 │  
100. │ 8.211 2057 │  
 ├─────────────────┤  
101. │ 11.982 2084 │  
102. │ 16.674 2226 │  
103. │ 12.62 2304 │  
104. │ 12.905 2341 │  
105. │ 11.615 2353 │  
 ├─────────────────┤  
106. │ 9.321 2367 │  
107. │ 12.962 2451 │  
108. │ 16.932 2457 │  
109. │ 9.648 2507 │  
110. │ 18.35 2530 │  
 ├─────────────────┤  
111. │ 17.333 2576 │  
112. │ 12.015 2607 │  
113. │ 11.32 2870 │  
114. │ 22.337 2993 │  
115. │ 19.035 3202 │  
 ├─────────────────┤  
116. │ 12.205 3286 │  
117. │ 17.078 3312 │  
118. │ 25.528 3498 │  
119. │ 24.021 3538 │  
120. │ 32.197 3794 │  
 ├─────────────────┤  
121. │ 26.652 3841 │  
122. │ 20.164 4014 │  
123. │ 14.132 4217 │  
124. │ 21.41 4305 │  
125. │ 23.244 4494 │  
 ├─────────────────┤  
126. │ 29.845 4764 │  
127. │ 32.318 5277 │  
128. │ 21.988 5283 │  
129. │ 35.229 5668 │  
130. │ 17.467 5681 │  
 ├─────────────────┤  
131. │ 22.828 5819 │  
132. │ 33.154 6000 │  
133. │ 32.228 6119 │  
134. │ 34.168 6136 │  
135. │ 40.594 7193 │  
 ├─────────────────┤  
136. │ 33.354 7886 │  
137. │ 64.542 8419 │  
138. │ 41.238 8642 │  
139. │ 47.993 8787 │  
140. │ 69.878 9484 │  
 ├─────────────────┤  
141. │ 44.894 9956 │  
142. │ 67.12 11477 │  
143. │ 73.05 11796 │  
144. │ 139.422 14359 │  
145. │ 119.939 16719 │  
 └─────────────────┘  
  
. hist Q, width(1000) frequency  
(bin=17, start=2, width=1000)  
  
. kdensity Q  
  
. scatter TC Q  
  
. gen n=\_n  
  
. scatter TC Q, mlabel(n) mlabpos(6)  
  
. twoway (scatter TC Q)(lfit TC Q)  
  
. graph save scatter1  
(file scatter1.gph saved)  
  
. twoway (scatter TC Q)(qfit TC Q)  
  
. graph save scatter2  
(file scatter2.gph saved)  
  
. graph combine scatter1.gph scatter2.gph  
  
. graph export scatter3.png  
(file scatter3.png written in PNG format)



scatter

### 6.2 a new one

Let us read the fuel efficiency data that ships with Stata

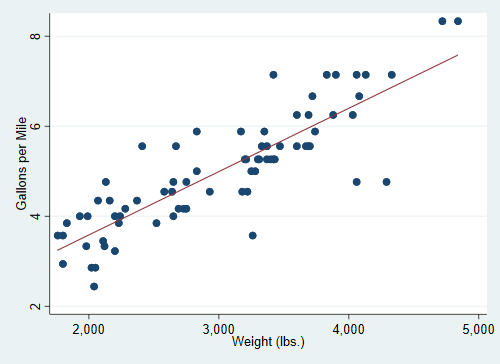
. sysuse auto, clear   
(1978 Automobile Data)

To study how fuel efficiency depends on weight it is useful to transform the dependent variable from "miles per gallon" to "gallons per 100 miles"

. gen gphm = 100/mpg

We then obtain a fairly linear relationship

. twoway scatter gphm weight || lfit gphm weight ///  
> , ytitle(Gallons per Mile) legend(off)   
  
. graph export auto.png, width(500) replace   
(file auto.png written in PNG format)

 The regression equation estimated by OLS is

. regress gphm weight   
  
 Source │ SS df MS Number of obs = 74  
─────────────┼────────────────────────────────── F(1, 72) = 194.71  
 Model │ 87.2964969 1 87.2964969 Prob > F = 0.0000  
 Residual │ 32.2797639 72 .448330054 R-squared = 0.7300  
─────────────┼────────────────────────────────── Adj R-squared = 0.7263  
 Total │ 119.576261 73 1.63803097 Root MSE = .66957  
  
─────────────┬────────────────────────────────────────────────────────────────  
 gphm │ Coef. Std. Err. t P>|t| [95% Conf. Interval]  
─────────────┼────────────────────────────────────────────────────────────────  
 weight │ .001407 .0001008 13.95 0.000 .001206 .0016081  
 \_cons │ .7707669 .3142571 2.45 0.017 .1443069 1.397227  
─────────────┴────────────────────────────────────────────────────────────────

Thus, a car that weighs 1,000 pounds more than another requires on average an extra 1.4 gallons to travel 100 miles. That's all for now!