



# Fugitive - vim plugin for git

Matthew Citron

# Getting Started

- Fugitive allows **all** git file operations within vim
- Much improved interface
- Everything is reversible within vim buffer!
- Download from <https://github.com/tpope/vim-fugitive> (can use e.g. pathogen to install)
- Nice tutorial: <http://vimcasts.org/episodes/fugitive-vim---a-complement-to-command-line-git/>

# Basic Commands

git	fugitive	action
:Git add %	:Gwrite	Stage the current file to the index
:Git checkout %	:Gread	Revert current file to last checked in version
:Git rm %	:Gremove	Delete the current file and the corresponding Vim buffer
:Git mv %	:Gmove	Rename the current file and the corresponding Vim buffer

# Basic Commands - git status

- Gives an interactive version of the git status

## Command

- add/reset file (works in visual mode too)

<Enter> open current file in the window below

p run `git add -patch` for current file

C invoke :Gcommit

# Example

## Edit File and run Gstatus

```
1 # On branch DMBranch
2 # Changed but not updated:
3 #   (use "git add <file>..." to update what will be committed)
4 #   (use "git checkout -- <file>..." to discard changes in working directory)
5 #
6 #       modified:   analysis/statsCode/optimiseBinning.py
7 #
8 # Untracked files:
9 #   (use "git add <file>..." to include in what will be committed)
10 #
11 #       analysis/.___afs10D8
12 #       analysis/.___afsE2FA
13 #       analysis/AlphaTModule/PlotModule2012.py
14 #       analysis/combine400_900.root.dot
15 #       analysis/copyFile.C
16 #       analysis/dataCards/
17 #       analysis/dmOutput/
```

# Example

Stage using '-'

```
1 # On branch DMBbranch
2 # Changes to be committed:
3 #   (use "git reset HEAD <file>..." to unstage)
4 #
5 #       modified:   analysis/statsCode/optimiseBinning.py
6 #
7 # Untracked files:
8 #   (use "git add <file>..." to include in what will be committed)
9 #
10 #    analysis/.__afs10D8
11 #    analysis/.__afsE2FA
12 #    analysis/AlphaTModule/PlotModule2012.py
13 #    analysis/combine400_900.root.dot
14 #    analysis/copyFile.C
15 #    analysis/dataCards/
16 #    analysis/dmOutput/
```

# Example

Edit by pressing <enter> and discard changes

```
1 # On branch DMBranch
2 # Untracked files:
3 #   (use "git add <file>..." to include in what will be committed)
4 #
5 #   analysis/.__afs10D8
6 #   analysis/.__afsE2FA
7 #   analysis/AlphaTModule/PlotModule2012.py
8 #   analysis/combine400_900.root.dot
9 #   analysis/copyFile.C
10 #   analysis/dataCards/
11 #   analysis/dmOutput/
```

# Nice Commands - Gdiff

Run vimdiff between file and staged version

```
+ 1 +-- 54 lines: from collections import namedtuple,
55         print
56     else:
57         verboseprint = lambda *a: None      # do->
58         return verboseprint
59
60 def main(nToys, defaultRange, writeSysts, shapes, ver>
61
62     from countsFromBins import countsFromBinDict,>
63     import commonHC
64     from configStat import importArguments
65     from getHistoFromFile import getHistoFromFile
66     import ROOT as r
+ 67 +--148 lines: r.gROOT.SetBatch(True)-----+
+ 1 +-- 54 lines: from collections import namedtupl
55         print
56     else:
57         verboseprint = lambda *a: None      # >
58         return verboseprint
59
60 def main(nToys, defaultRange, writeSysts, shapes,>
61     print "test"
62
63     from countsFromBins import countsFromBinDi>
64     import commonHC
65     from configStat import importArguments
66     from getHistoFromFile import getHistoFromF>
67     import ROOT as r
+ 68 +--148 lines: r.gROOT.SetBatch(True)-----+
```

:Gdiff <branch> will vimdiff current version with that on another branch

Can use with merge conflicts to get 3 way diff!



# Nice Commands - Gdiff

Run vimdiff between file and staged version

```
+ 1 +-- 54 lines: from collections import namedtuple, + 1 +-- 54 lines: from collections import namedtupl
55         print
56     else:
57         verboseprint = lambda *a: None # do->
58     return verboseprint
59
60 def main(nToys, defaultRange, writeSysts, shapes, ver>
61
62     from countsFromBins import countsFromBinDict,>
63     import commonHC
64     from configStat import importArguments
65     from getHistoFromFile import getHistoFromFile
66     import ROOT as r
+ 67 +--148 lines: r.gROOT.SetBatch(True)-----+ 68 +--148 lines: r.gROOT.SetBatch(True)-----
```

Use **diffput** (dp) and **diffget** (dg)  
to choose changes to discard/keep

# Nice Commands - Gblame

Git blame to find who was the last to edit each line

```
df3412bd analysis/cumulative.py 11 (Matthew > 13 def parse_args():
df3412bd analysis/cumulative.py 12 (Matthew > 14     parser = optparse.OptionParser()
6dbc00e6 analysis/cumulative.py 17 (Matthew > 15     parser.add_option('-d', '--defaultRange', action="store_true",>
7623e28e analysis/statsCode/optimiseBinning.py 16 (Matthew > 16     parser.add_option('-n', '--nToys', help='Number of pseudo-expe>
40ea75d7 analysis/cumulative.py 20 (Matthew > 17     parser.add_option('-S', '--writeSysfs', help='-S 0 for no syst>
7ae1c4ad analysis/optimiseBinning.py 22 (Stefano Casasso > 18     parser.add_option('-s', '--shapes', action = "store_true", help>
7623e28e analysis/statsCode/optimiseBinning.py 19 (Matthew > 19     parser.add_option('-v', '--verboseArg', action = "store_true",>
aaaa9a1c analysis/optimiseBinning.py 26 (Matthew > 20     parser.add_option('-o', '--outDir', default="output_optimisat>
7623e28e analysis/statsCode/optimiseBinning.py 21 (Matthew > 21     parser.add_option('-c', '--correlation', default = "none", help>
7623e28e analysis/statsCode/optimiseBinning.py 22 (Matthew > 22     parser.add_option('--cowan', action="store_true", help="Use >
7623e28e analysis/statsCode/optimiseBinning.py 23 (Matthew > 23     parser.add_option('--cardPerCategory', action="store_true", >
7623e28e analysis/statsCode/optimiseBinning.py 24 (Matthew > 24     parser.add_option('-l', '--logUni', action="store_true", help>
7623e28e analysis/statsCode/optimiseBinning.py 25 (Matthew > 25     parser.add_option('--controlMult', type = float, default = 1>
7623e28e analysis/statsCode/optimiseBinning.py 26 (Matthew > 26     parser.add_option('--noWriteSignalModels', action="store_tru>
6c7203ac analysis/statsCode/optimiseBinning.py 37 (Matthew > 27     parser.add_option('--signalModel', help="signal model to run>
aaaa9a1c analysis/optimiseBinning.py 27 (Matthew > 28     parser.add_option('-f', '--force', action="store_true", help=>
```

Can use to find bugs/apportion blame to idiotic colleagues

# Summary

- Many more commands (Ggrep, Glog....)
- Makes dealing with merge conflicts or finding differences much nicer
- Changes are made on the vim buffer
  - Gread (unlike git checkout —) is reversible!
- Add `%{fugitive#statusline()}` to 'statusline' to get current branch on the statusline