

A.) Use the git log command to output a report showing all commits by author Mike McQuaid during August 2017 in this format: author name (author date) and commit message. Chapter 2.3 in the Git Book has instructions and options you can use for formatting and filtering log output.

Git command used:

```
git log --author="Mike McQuaid" --since="2017-08-01" --before="2017-09-01"
--pretty=format:"%an (%ad) %s"
```

---

B.) Rank the activity of the top 10 most active authors by number of commits during 2016.

Git command used:

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2016-01-01"
--before="2016-12-31" > ~/gitlog.csv
```

Explanation:

Prettify git log, isolate dates from 01/01/2016 to 12/31/2016 and download them

all to a CSV file. Edit CSV file on Google Drive (easier to isolate unique values here), isolate unique names, count their appearances, and voala. Here is

your top 10 most active authors in 2016:

Name:	Commits:
Mike McQuaid	1152
Markus Reiter	298
Xu Cheng	234
Martin Afanasjew	219
Dominyk Tiller	210
Alyssa Ross	109
ilovezfs	79
Josh Hagins	72
Baptiste Fontaine	64
Misty De Meo	63

OR (Lazy Way)

```
git shortlog -sn --since="2016-01-01" --before="2016-12-31"
```

---

C.) Report on the total number of commits made each year since the repository started in 2009.

Git commands used:

(2009) - 610 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2009-01-01"
--before="2009-12-31" > ~/brewlog2009.csv
```

(2010) - 540 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2010-01-01"
--before="2010-12-31" > ~/brewlog2010.csv
```

(2011) - 555 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2011-01-01"
--before="2011-12-31" > ~/brewlog2011.csv
```

(2012) - 1170 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2012-01-01"
--before="2012-12-31" > ~/brewlog2012.csv
```

(2013) - 1886 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2013-01-01"
--before="2013-12-31" > ~/brewlog2013.csv
```

(2014) - 2068 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2014-01-01"
--before="2014-12-31" > ~/brewlog2014.csv
```

(2015) - 1770 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2015-01-01"
--before="2015-12-31" > ~/brewlog2015.csv
```

(2016) - 3066 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2016-01-01"
--before="2016-12-31" > ~/brewlog2016.csv
```

(2017) - 3075 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2017-01-01"
--before="2017-12-31" > ~/brewlog2017.csv
```

(2018) - 2800 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2018-01-01"
--before="2018-12-31" > ~/brewlog2018.csv
```

(2019) - 1432 commits

```
git log --date=local --pretty="%h, %an, %ar : %s" --since="2019-01-01"
--before="2019-12-31" > ~/brewlog2019.csv
```

And then, last line should be number of commits that has been made

OR (Lazy way)

```
(2009) - 610 commits
git rev-list --count HEAD --since="2009-01-01" --before="2009-12-31"
(2010) - 540 commits
git rev-list --count HEAD --since="2010-01-01" --before="2010-12-31"
(2011) - 555 commits
git rev-list --count HEAD --since="2011-01-01" --before="2011-12-31"
(2012) - 1170 commits
git rev-list --count HEAD --since="2012-01-01" --before="2012-12-31"
(2013) - 1886 commits
git rev-list --count HEAD --since="2013-01-01" --before="2013-12-31"
(2014) - 2068 commits
git rev-list --count HEAD --since="2014-01-01" --before="2014-12-31"
(2015) - 1770 commits
git rev-list --count HEAD --since="2015-01-01" --before="2015-12-31"
(2016) - 3066 commits
git rev-list --count HEAD --since="2016-01-01" --before="2016-12-31"
(2017) - 3075 commits
git rev-list --count HEAD --since="2017-01-01" --before="2017-12-31"
(2018) - 2800 commits
git rev-list --count HEAD --since="2018-01-01" --before="2018-12-31"
(2019) - 1432 commits
git rev-list --count HEAD --since="2019-01-01" --before="2019-12-31"
```

IF merges should not be included, add an extention: --all --no-merges  
i.e. git rev-list --count HEAD --since="2009-01-01" --before="2009-12-31"  
--all --no-merges

---

D.) Generate a list of the top 10 contributors (authors-the most # lines of code) to the project during the first 6 months of 2017 and identify how many files they modified and how many lines of code they inserted and deleted. Rank them based on number of insertions. (Hint: use the --shortstat option to get the raw data). Run the same report for calendar year 2016. Explain briefly how you compiled the list.

Git command used:  
(2017)

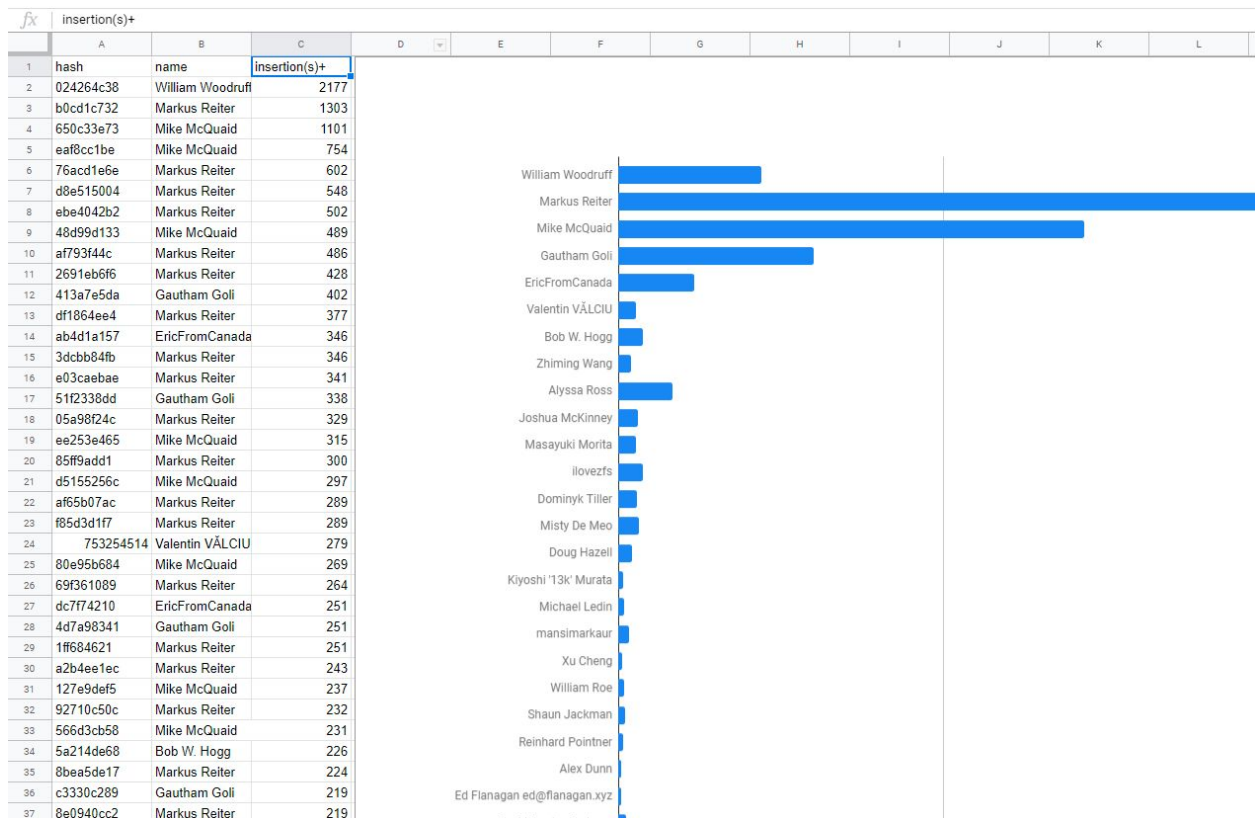
```
git log --since="2017-01-01" --before="2017-06-30" --pretty="@%h,"
--shortstat | tr "\n" " " | tr "@" "\n" > ~/stats.csv
git log --pretty="%an" --since="2017-01-01" --before="2017-06-30" >
~/names.csv
```

Top 10 Contributors (Based on Insertion):

Name: Insertions:

```
Markus Reiter      20823
Mike McQuaid      7173
Gautham Goli      3006
William Woodruff  2206
EricFromCanada    1172
Alyssa Ross       839
Bob W. Hogg       384
Ilovezfs          375
Misty De Meo      315
Joshua McKinney   304
```

Explanation:



A graph that aggregates all the name values and sums the amount of insertions. Markus Reiter being the obvious winner here and some other obvious outliers, like: Mike, Gautham, William, EricFromCanada, and Alyssa

Ross. The other I had to spot manually. A separate graph like this was also done with File Changes and Deletion.

Git command used:

(2016)

```
git log --since="2016-01-01" --before="2016-12-31" --pretty="@%h,"  
--shortstat | tr "\n" " " | tr "@" "\n" > ~/stats.csv  
git log --pretty="%an" --since="2016-01-01" --before="2016-12-31" >  
~/names.csv
```

Top 10 Contributors (Based on Insertion):

Name:	Insertions:
AnastasiaSulyagina	22415
Markus Reiter	20197
Mike McQuaid	12118
William Woodruff	4697
Martin Afanasjew	37989
Xu Cheng	3620
Baptiste Fontaine	3374
Alyssa Ross	2207
Andrew Janke	1645
Vlad Shablinsky	1129

Explanation:

Same process as before.

---