1. Prepare data/models and check

Changes to data layers

My most common checks

```
hist()
summary()
dim()
table()
plot(old_data, new_data)
```

Questions

Does range/distribution seem reasonable?
Does the number of NA values make sense?
Correct sample size (i.e., length of data)?

Common Problems

Be especially neurotic after joins (check sample size, NAs)!
Compare distribution of NA values between old/new data!

A GoogleVis plot is good for visualizing multiple years of data

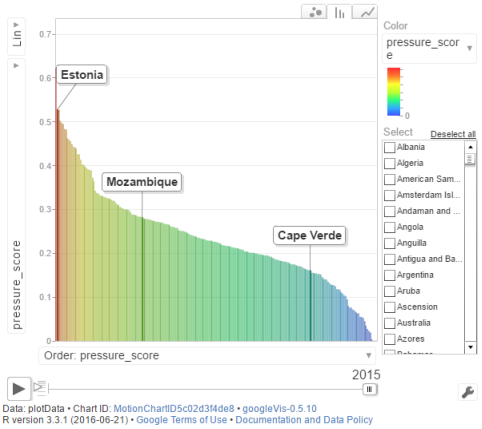
Arrange data like this

```
> head(plotData)
          rgn_nam year pressure_score
1    Cocos Islands 1993    0.015672246
2 Christmas Island 1993    0.000000000
3   Norfolk Island 1993    0.000000000
4 Macquarie Island 1993    0.119013228
5     New Caledonia 1993    0.001883531
6          Vanuatu 1993    0.000000000
```

```
library(googleVis)

Motion = gvisMotionChart(plotData,
                         idvar="rgn_nam",
                         timevar="year")
plot(Motion)

print(Motion, file =
file.path(save_loc, 'slr.html'))
```



Changes to functions

Put a `browser()` at start of the function in functions.R. This will stop the calculations at this point, then walk through the function to make sure everything is going well.

2. Run OHIcore functions and visualize scores

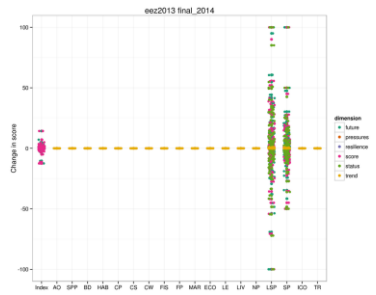
Check

1. Did the files you expect to change actually change?
2. Check the scores diff file to see if changes make sense.
3. Review `warning()` messages.

Future

Current visualization is bad at detecting changes to missing data (which is a common mistake). Make a function to do this.

How do scores change across goals?



Currently in ohicore, but needs to be updated for ohi+ repositories

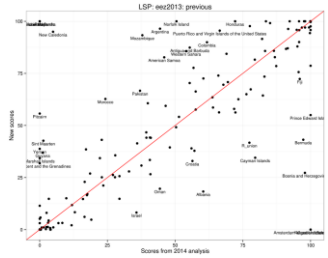
```
changePlot(repo = "ohi-global",
           scenario = "eez2013",
           commit = "previous",    # can also use sha, e.g., "e30e7a4"
           fileSave = "name_of_plot")
```

saves interactive html figure to: changePlot_figures

To find sha values:
Can use git2r package: `git2r::commits(git2r::repository(repo2))`
Can also just look it up on github.com

Can read csv files from previous commits in Github (git2r package):
read_git_csv(org/repo, sha, path)
read_git_csv("OHI-Science/ohi-global", "9627676", "eez2014/scores.csv"

How do scores change within a goal/dimension?

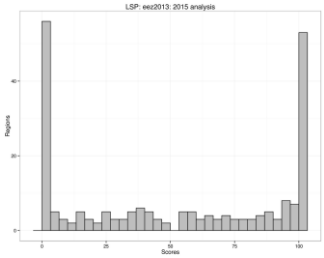


Currently in ohiprep/src/R/VisGlobal.R (planning to move to ohicore)

Regions >1 sd from mean difference are labeled

```
scatterPlot(repo = "ohi-global",
scenario = "eez2013",
commit = "previous",
goal = "LSP",
dim = "status",
fileSave = "LSPstatus")
```

Distribution of scores within a goal/distribution



Currently in ohiprep/src/R/VisGlobal.R (planning to move to ohicore)

```
goalHistogram(repo = "ohi-global",
scenario = "eez2013",
goal="LSP",
dim="score",
fileSave="LSP_trend_data_update")
```

Checklist for ohi assessment

- [ ] Update "layer" spreadsheet
- [ ] Pull ohi-global and ohiprep
- [ ] Check that branch is correct
- [ ] Run Calculate_scores_all.R
- [ ] Check diffs of scores.csv to see if changes make sense
- [ ] Visualize data
- [ ] Commit/Push ohi-global
- [ ] Post changes on github for review

3. Document and share

I have an ongoing github issue to: 1) document changes to data/models, 2) describe effects on scores, 3) describe outliers and explain why they are outliers.