Software Design Documentation

Software Architecture:

First of all, this is a web application available at “[http://risk-mgmt-eevee.herokuapp.com](http://risk-mgmt-eevee.herokuapp.com/)”. There are three major components:

* HTML code:
  + For each sub-URL that belongs to the root-URL "[http://risk-mgmt-eevee.herokuapp.com](http://risk-mgmt-eevee.herokuapp.com/)", we need a HTML code saved in the "templates" folder. For example, for "<http://risk-mgmt-eevee.herokuapp.com/index>", we need "index.html" in the "templates" folder.
  + Bootstrap, a CSS/JS template, was used in creating the HTML code for this project.
  + "static" folder contains files to be loaded into the website, such as css and js files.
  + Javascript is used to make the website more user-friendly. For example, when a user click on one of the buttons for plotting and the page is redirected to the new page with generated plot, it automatically scrolls to the plot position.
  + The default inputs in the forms on the website are coded dynamically so that the last used parameters entered by the user become the new default inputs.
* Core Python script:
  + Flask, a microframework for Python, is used in connecting the Python code with the HTML code. The structure of "static" and "templates" folders is the Flask default required structure.
  + app.py is the python file powering the website. It will be discussed in more details in the following sections.
  + The application can also be used locally by running “python app.py”, provided all the required packages are installed.
  + Bokeh, a Python interactive visualization library that targets modern web browsers for presentation, is used for ploting. When a Bokeh plot is generated in the python back-end, it is coded into a JavaScript to get rendered in the website. If interested, visit the Bokeh documentation (<http://bokeh.pydata.org/en/latest/docs/user_guide.html>).
* Heroku deployment related files:
  + This project is deployed on Heroku, a cloud Platform-as-a-Service (PaaS) that is used as a web application deployment model. A general reference to Python app deployment using Heroku can be found here: <https://devcenter.heroku.com/articles/getting-started-with-python-o>.
  + The following buildpack (<https://github.com/kennethreitz/conda-buildpack>) is used as the buildpack for this Heroku app.
  + "conda-requirements.txt" and "requirements.txt" contain the required python packages. Heroku checks those files for what need to be installed in order for the app to run properly. It is important to list all the packages used in the core python script here, otherwise the application might not function.
  + "nomkl" in "conda-requirements.txt" prevents some slug size problem when deploying to heroku, as it will use the non-mkl optimized binaries, and won't download the mkl package. Otherwise, the slug size will exceed Heroku's limit.
  + "Procfile" and "runtime.txt" contain some default settings.

app.py components:

* Packages imports:
  + Some major packages include Pandas, Flask, and Bokeh.
* Function definitions:
  + The purposes of each function are commented in the app.py file.
* Main Flask app:
  + This is the body of the application. As the user click on different buttons on the website, different actions are executed in the back-end as written in this part and a new web page with newly generated results is rendered.

Data Flow:

* When the user enters his inputs on the website, they are transferred to the python back-end.
* Pandas-datareader package retrieves the required data from Yahoo Finance and load into a properly formatted pandas dataframe.
* The calculation methods take the user inputs and retrieved data and turn them into the result data of the user’s choosing.
* The result data are stored in .csv files for the user to download and used in Bokeh to create interactive plots for the user to view.