Final Project:

Installation

1. Open a command prompt in the folder, and type 'pip install –r requirements.txt' to make sure that all the dependencies are installed

2. To run the file, on a command prompt type 'python RiskManagement.py'

3. Please note: This script makes use of fix_yahoo_finance, which may bug out during execution and fetch blank dataframes. Please re-run the program

Explanation:

1. **Download data for last 10 years for all the 30 stocks of the Dow Jones Industrial Average (ignore survivorship bias, and unless you have access to a point-in-time database, simply download the data for the current set of DJIA index constituents.)**
   NOTE: I've only downloaded for the past 4 years. This is because a lot of Data takes lots of time to process, and takes upward of 10 minutes or so on a PC with less RAM to create say, stop loss portfolio. Therefore, as a sample scenario, I've opted to keep the data small, as it is more about KPI rather than generating a large portfolio.

```
def generate_portfolio():
    # replacing visa with At&t as visa started trading only in 2008
    DJI_constituents = ['MT','JPM','MSFT','DIS','BA','MCD',
                        'TRV','MMM','NKE','UNH','CSCO','JNJ',
                        'T','INTC','KO','PG','MRK','IBM',
                        'DD','UTX','PFE','CAT','GE','VZ',
                        'HD','AXP','GS','AAPL','XOM','CVX']
    close = pdr.get_data_yahoo(DJI_constituents, start='2013-09-01', end='2016-09-01', as_panel=False).fillna(0)
    position = pd.DataFrame(data=0,index=close.index,columns=close.columns)
    portfolio = pd.Panel({'price':close,'pos':position})
    return portfolio
```

2. **In addition, download data for Futures contract of the DJIA for the same period of time (consider the nearest month contract always)**

```
def get_hist_futures(future_code, start_date, end_date):
    DJIA_Futures = quandl.get(future_code,
                              authtoken="M9yAZGcQVxrQKRr6WYjw",
                              # authtoken="cvQViZ3mh8gkuANqgTc_",
                              start_date=start_date,
                              end_date=end_date)
    return DJIA_Futures
```

Futures data is downloaded from Quandl

3. **For each of the cases outlined below, (unless specifically mentioned) randomly select a basket of 10 stocks from the list and create an equal weighted portfolio**
   An equally weighted portfolio with random weights is created

```
def equal_weighted_portfolio(portfolio):
    """

    :type portfolio: panel
    """
    random_list = []
    while len(np.unique(random_list)) != 10:
        random_list = np.random.random_integers(0,29,10)
    for stock in random_list:
        portfolio['pos'].iloc[:,stock] = round(1000/ portfolio['price'].iloc[0,stock])
    return portfolio
```

4. **Risk-return profile calculations should include calculations for the following KPIs**
   The find_kpi function creates a capital amount, where capital amount is price into position, a pnl (profit and loss), a cumulative pnl and a return. The following KPIs are then calculated:

```
monthly = ret.resample('M').sum().mean().fillna(0)
print("\tMonthly returns average : ", monthly.mean())

individual_month = ret.resample('M').mean().fillna(0)
print("\tPositive Monthly return percentage :",
      (individual_month[individual_month>0]/
       individual_month[individual_month>0]).fillna(0).mean().mean())

yearly = ret.resample('A').sum().mean().fillna(0)
print("\tYearly returns average : ", yearly.mean())

monthly_high = cumpnl.resample('M').sum().max().fillna(0)
monthly_low = cumpnl.resample('M').sum().min().fillna(0)
print("\tMax monthly Drawdown : ",(monthly_high-monthly_low).max())

alltime_high = cumpnl.sum().max()
alltime_low = cumpnl.sum().min()
print("\tMax Drawdown : ",(alltime_high-alltime_low).max())

water = (cumpnl.sum().max()-cumpnl.sum()).fillna(0).sum()
earth = (cumpnl.sum() - cumpnl.sum().min()).fillna(0).sum()
print("\tLake ratio : ", water/earth)

total_returns = ret.sum().sum()
negative_returns = abs(ret[ret<0]).sum().sum()
print("\tGain to pain ratio : ",total_returns/negative_returns)
```

5. **Risk Management by Stop Loss**
   i) **Estimate the risk-return profile of such a long-only portfolio over the period of last 10 years**
   We estimate the KPI of the equally weighted portfolio

```
('\tMonthly returns average : ', 0.009484302992541258)
('\tPositive Monthly return percentage :', 0.033045977011494254)
('\tYearly returns average : ', 0.0853587269328713)
('\tMax monthly Drawdown : ', '22726.619929999983')
('\tMax Drawdown : ', '335274.541364')
('\tLake ratio : ', 49.18680841240304)
('\tGain to pain ratio : ', 0.10190581593076344)
```

ii) **Now consider a modified portfolio, where position in any stock is fully liquidated if the price drop 20% from its nearest 6-month high and its position is filled by another randomly selected stock from the rest of the group**

The following code snippet does the aforementioned question

```
#dataframe to indicate price is too low and thus time to switch
switch = price[price/max_price < 0.8].fillna(0)
```

This is within the generate_stop_loss() function

iii) **Estimate the risk-return profile of such a modified portfolio over the same period. Compare and contrast the results with the simple long-only portfolio**

```
Printing KPIs for stop-loss portfolio :
('\tMonthly returns average : ', 0.007768594220888319)
('\tPositive Monthly return percentage :', 0.035759897828863345)
('\tYearly returns average : ', 0.06991734798799484)
('\tMax monthly Drawdown : ', '14450.23976899999')
('\tMax Drawdown : ', '235297.91669500014')
('\tLake ratio : ', 44.61197901550544)
('\tGain to pain ratio : ', 0.08953119042006957)
```

The monthly average returns of the equally weighted portfolio are slightly better than that of the stop loss one, however, positive monthly return of the stop loss portfolio are better. The yearly returns of the equally weighted portfolio seem to be better. The Monthly drawdown of the stop loss portfolio is better than the equally weighted one. The G2P ratio of equally weighted portfolio seems to be slightly better, along with lake ratio.