Econ 250: Data Assignment #1

Lawrence D.W. Schmidt
University of Chicago

Due: February 1
Please hand in at the start of lecture on January 31st or at section/office hours on February 1st

1 Exercises with historical data

This section is intended to give you a brief review of how to compute nominal and real returns, means, variances, and covariances, and also to replicate some stylized facts about equity returns. You will be able to reproduce many of the results that appear in Chapter 10 of the textbook after working through these exercises. All of the data required to perform the relevant computations are available in the file *assigment_1_data.csv*. The data are collected from CRSP and the St. Louis Federal Reserve's FRED database and cover the sample period from 1926-2016. (We have normalized the nominal price level of the market index to $1 as of the end of 1925. The market index is a value-weighted portfolio of all stocks on the NYSE.)

Feel free to work with these data in the statistics package of your choosing, and **please attach a printout of your code to your problem set**. Also, please summarize the relevant output in a linear document so that we do not need to run the code to view your results. We will use the data to get some practice doing various computations related to learning about risk and return from the observed data.

(1) **Orders of magnitude of movements in prices over time**:

  (a) Make a plot with the nominal price of the market portfolio over time. Make a similar plot with the nominal dividend of the market. Is it more appropriate to use levels or a logarithmic scale on the vertical axis in these plots? Why? Which series appears to be more volatile?

  (b) Plot the ratio of prices to dividends. What is the key difference between this picture and your earlier plots?

(2) **Calculating returns**: Use the formulas we learned in class to compute a sequence of monthly *net* returns (i.e., subtract 1) for the market portfolio. Columns *rf*, *small*, and *large* give monthly net returns on a one month risk-free asset (T-bills), small stocks and large stocks, respectively.

   (a) Report the arithmetic mean and standard deviation of monthly net returns for each of these assets.

   (b) Please also compute monthly *excess* returns for the market, small stock, and large stock portfolios, respectively, then report the mean and standard deviation of monthly excess returns.

(3) **Adjusting for inflation**: Use the data from the CPI to compute monthly real returns on each of the assets: the market portfolio, T-bills, small stocks, and large stocks. Please report the arithmetic mean and standard deviation of monthly real returns for each asset. In addition, please make a histogram of the monthly distribution of monthly returns on each asset.

(4) **Compounding returns**: For each month in your sample with sufficient data available, compute real returns associated with a trading strategy which holds each asset for 12, 60, and 120 months, respectively. Report the arithmetic mean and standard deviation of these compounded returns in annualized units (multiply means by 12/horizon in months) and standard deviations by $\sqrt{12/\text{horizon in months}}$. Comment on any patterns you notice about the relationship between the annualized standard deviation and the investment horizon. Which asset appears most risky at a 10 year horizon?

(5) **Understanding portfolio drift and rebalancing**. A practical problem arises when forming portfolios in a dynamic context. Suppose, for simplicity, you have \$1 to invest now. For example, you make your initial investment and form a portfolio with w_1% in asset 1 and $(1 - w_1)$% weight in asset 2. Let's suppose for simplicity that asset 2 earns a constant risk-free rate in both periods. In period 1, you have a chance to *rebalance* the portfolio. In particular, you can take the proceeds from your period 1 return and reinvest it with the same weights as period 1. (In practice, this might be associated with tax consequences or transaction costs.) Alternatively, you can do nothing, and hold the initial investment through period 2. This is called a *buy-and-hold* strategy.

   (a) What are the dollar amounts reinvested in each asset at period 2 with the buy-and-hold strategy? What are the new portfolio weights (i.e., percentages invested in each asset) if I do not rebalance the portfolio? If I choose the rebalance, which asset am I buying and which asset am I selling if the excess return on asset 1 is negative?

(b) Suppose that I believe that, for some reason, the returns of asset 1 in periods 1 and 2 are negatively correlated (high returns in period 1 signal low returns in period 2 and vice versa). Can you provide a verbal intuition for why I might prefer to actively rebalance the portfolio in period 2 in this case?

(6) **Comparing portfolio returns with and without rebalancing**: Suppose I am considering two trading strategies: both of which invest 30% in the market, 20% in the risk-free asset, 30% in small stocks, and 20% in large stocks over a 5 year holding period. The first trading strategy actively rebalances the portfolio each month, while the second uses the buy and hold approach, never rebalancing the portfolio. Similar to the above question with compounding, report the mean and variance of returns over these holding periods. What do you notice? Also, please report the mean and variance of the fraction of the buy-and-hold portfolio's weight in risk-free asset at the end of the holding period.

(7) **Comparing arithmetic versus geometric means**: Calculate the geometric mean of the real market, small cap portfolio, and risk-free returns, respectively. Please express you answers in annualized, net return units. Recall that the geometric mean is simply

$$G = \left[\prod_{t=1}^{T} R_t\right]^{\frac{12}{T}} - 1 = \exp\left[\frac{12}{T}\sum_{t=1}^{T}\log R_t\right] - 1.$$

For which of the two assets is the gap between the geometric and arithmetic mean measures largest? Provide the intuition for the result.

(8) **Statistical inference via monte carlo simulation**: A common assumption that we will make in parts of the course is that asset returns are independent and identically distributed (iid). Under such an assumption, a common way of conducting statistical inference is to sample *with replacement* from the observed history. This is sometimes called bootstrapping, monte carlo simulation, or nonparametric inference. Traders may often use these procedures when testing for the robustness of the performance of a given trading strategy to a few lucky draws.

While the econometric theory which formally justifies the use of the bootstrap is complex and well beyond the scope of this course, the basic procedure is extremely straight-forward. Here's how it works (suppose there are T months of data and K different columns in a Pandas dataframe named Y).

- Randomly select T independent integers between 0 and $T-1$. The Python statement *boot_rows = np.random.randint(0,T,T)* will return exactly this.

- Create a new, artificial dataset using the sampled rows. The Pandas syntax for this would be: *Y_boot = Y.iloc[boot_rows, :]*
- Perform your statistical procedure, just like you did on the real data, on this simulated time series and store the output, e.g., as an entry in a vector called *boot_stats*.
- Repeat a large number of times (e.g. 1000), which yields a *distribution of simulated statistics*. Its standard deviation is a nonparametric standard error estimate, and its 2.5th and 95th percentiles can be used to compute a 2-sided 95% confidence interval for the parameter of interest.

Later on in the class, we will use such a procedure a couple of times in lieu of calculating standard errors by pencil and paper. Bootstrapping can be an incredibly powerful tool to handle complex procedures in many situations.[1] So, let's try it out!

Just to get a feel for how this works, for this assignment, use the above procedure to add bootstrapped standard errors–which is just the standard deviation of the simulated statistics you kept track of–to the table above. The goal would be to produce a new table that looks like this (we often put the standard errors in parentheses as a convention), which should look very similar to the one we saw from lecture.

Statistic	Real stock return	Real risk-free return	Real small cap return	Real large cap return
Arithmetic mean				
BS std err of mean				
Volatility (std dev)				
BS std err of volatility				

How precise are our estimates of expected real returns? What is a 95% confidence interval for the expected real stock return? What about real small cap returns?

(9) **Correlation and covariance**: Please report the covariances and correlations between the real returns on the three equity portfolios (market, small caps, large caps) in this sample.

(10) **Stock returns and the real economy–first look**: A number of times in class, I have mentioned that stocks tend to do well in good macroeconomic environments and poorly in bad macroeconomic conditions. To get a flavor for this, I would like you to practice by running your first regression. We want to estimate the following equations by OLS:

$$Unemployment_{t+12} - Unemployment_t = \alpha_1 + \beta_1 \cdot r_{t+1} + \gamma Unemployment_t + \epsilon_{t+1:t+h},$$

[1] For instance, tests of the CAPM run several regressions in sequence. Standard errors are a pain to compute in these cases, but the bootstrap is easy.

$$\log(IndustrialProduction_{t+12}) - \log(IndustrialProduction_t) = \alpha_2 + \beta_2 \cdot r_{t+1} + \epsilon_{t+1:t+h},$$

where r_{t+1} is the real stock return in month $t+1$ and the unemployment rate and industrial production index are both available in the provided data. Please report β_1, β_2, and the R^2 of each of the two regressions. Finally, please comment on the sign of β_1 and β_2 and whether they are consistent with the argument I've made in class.

2 Extra credit: Simulating a life-cycle portfolio choice strategy

Here is an optional (but fun!) exercise to give you a flavor for the types of things that you can accomplish with monte carlo simulations. It turns out that a few lines of code can provide a pretty realistic picture of the potential outcomes associated with real-life portfolio problems. These types of simulation exercises are frequently conducted by financial planners to try and answer questions like: "Am I saving enough for retirement?" A couple of for loops is enough to handle these sorts of questions!

To answer these questions, please use the historical real returns on stocks and T-bills that you calculated for your answers to the first question. Consider the perspective of a person who is just graduating college at age 25. She plans to retire at age 65, which is $40 \times 12 = 480$ months later. Her real, monthly income after graduating (period 0) is Y_0= \$5,000 = \$60,000/12, which will grow deterministically at a rate of 0.5% per month, so $Y_t = \$5000 \times 1.005^t$. Upon graduation she has no debt and no assets. She plans to save s% of her paycheck each month to invest, and she plans to invest ω% in stocks and $1 - \omega$% in bonds each period, rebalancing the portfolio monthly. Therefore her budget constraint is

$$\underbrace{W_{t+1}}_{\substack{\text{financial wealth at}\\ \text{the start of period t+1}}} = \underbrace{(W_t + sY_t)}_{\substack{\text{invested wealth at}\\ \text{the end of period t}}} \underbrace{(\omega R_{s,t+1} + (1-\omega)R_{f,t+1})}_{\text{return on investment}}.$$

Our objective will be to characterize the distribution of wealth at the point of retirement associated with a given trading strategy. To do this, we will randomly select a sample of 480 pairs of $(R_{s,t+1}, R_{f,t+1})$ (with replacement, as above), then update wealth iteratively using the budget constraint, starting from an initial condition of 0 at time 0. The final observation, W_{480} gives us an expression for real wealth at retirement. This is a particular realization of terminal wealth, which we want to store in a vector (analogous to the *boot_stats* vector described above). Then, we draw a new set of potential returns and repeat the exercise. Once you've done this, say 1000, 5000, or 10000 times, make a histogram of the distribution of terminal wealth. Try making histograms with the following combinations of parameters:

$(s, \omega) \in \{(10\%, 50\%), (10\%, 100\%), (15\%, 50\%), (50\%, 100\%), (20\%, 0\%)\}$.

To clarify, the code you would write for this exercise involves two loops. There is an inner loop, which iterates over time periods to update the investor's wealth dynamically. Then, there is an outer loop which simulates multiple sample paths of wealth.

Another thing to note: it's very easy to play with the trading strategies. For example, you can pretty easily modify the profile of labor income, make savings rates and/or portfolio weights vary with age. Working these distributions by hand is really tedious, but designing a fairly realistic simulation actually turns out to be pretty easy!

3 Extra credit: Calculating prices of Arrow-Debreu securities with incomplete markets

Before answering the question, please read the following background information.

In this part of the problem set, you will get some hands on experience backing out the prices of Arrow-Debreu securities, given an $(N \times 1)$ vector of prices $\mathbf{p}$, a $(S \times 1)$ vector of probabilities π, and a $(N \times S)$ matrix of payoffs $\mathbf{X}$, as defined as in class.

Let's start with the simplest possible case. When $S = N$ and there are no redundant assets (i.e., $rank(\mathbf{X}) = N$), then we can use the formula from lecture:

$$p_j = \sum_{s=1}^{S} q_s X_{js}, \qquad \text{for } j = 1, \ldots, N$$

which we can write in matrix form as $\mathbf{p} = \mathbf{X}\mathbf{q}$. Thus, we can solve for $\mathbf{q} = \mathbf{X}^{-1}\mathbf{p}$.

Next, suppose that $N > S$, but $rank(\mathbf{X}) = S$.[2] Then, the law of one price *still* implies that $\mathbf{p} = \mathbf{X}\mathbf{q}$. If there are no arbitrage opportunities, then we should still be able to find a $(N \times 1)$ vector $\mathbf{q}$ which satisfies $\mathbf{p} = \mathbf{X}\mathbf{q}$. We can find this $\mathbf{q}$ by running a linear regression of $\mathbf{p_j}$ on $\mathbf{X_j}$ for $j = 1, \ldots, N$. No arbitrage implies that the pricing errors (regression residuals) should be exactly zero, even when N is much larger than S, and $\mathbf{q} > 0$.

Finally, let's consider the case with incomplete markets (i.e., $rank(\mathbf{X}) < S$, usually $N < S$ also). In this case, there is not a unique SDF that is consistent with the absence of arbitrage. However, one can show that there is a unique vector of Arrow-Debreu securities which is consistent with the absence of arbitrage and is *in the space of traded payoffs*. In other words, we can find a unique

[2] To calculate the rank of a matrix X in python, use the command *numpy.linalg.matrix_rank(X)*.

$(S \times 1)$ vector $\mathbf{q}^* \in \{\mathbf{X}'\beta \text{ for some } \beta \in \mathbb{R}^N\}$.[3] Geometrically, $\mathbf{q}^*$ is the *projection* of $\mathbf{q}$ onto the space of traded payoffs. Plugging in this assumption about $\mathbf{q}^*$, we get $\mathbf{p} = \mathbf{X}\mathbf{q}^* = \mathbf{X}\mathbf{X}'\beta$, which is a set of N equations in N unknowns.

If $\mathbf{X}\mathbf{X}'$ is nonsingular, then $\beta = [\mathbf{X}\mathbf{X}']^{-1}\mathbf{p}$. Otherwise, we may need to trim some redundant assets before calculating the formulas. Suppose that we identify $\mathbf{Z}$, which is a $(N^* \times S)$ matrix of non-redundant payoffs, where N^* is the number of non-redundant assets (linearly independent rows of $\mathbf{X}$).[4] Then, we can assume without loss of generality that $\mathbf{q}^* = \mathbf{Z}'\beta^*$. Then, in the absence of arbitrage, we get the following regression-like equation that should be satisfied (again, with no pricing errors):

$$\underset{(N\times 1)}{\mathbf{p}} = \underset{(N\times S)}{\mathbf{X}} \; \underset{(S\times N^*)}{\mathbf{Z}'} \; \underset{(N^*\times 1)}{\beta^*}.$$

Again, we can find β^* by running a regression of the price vector on $\mathbf{X}\mathbf{Z}'$, and $\mathbf{Z}'\beta^*$ is the unique vector of A-D prices which is spanned by the set of traded assets.[5]

Next, please answer the following questions (you will probably want to use Matlab):

(1) Suppose that

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} 1 \\ 0.75 \\ 2.5 \\ 3.25 \\ 6.5 \end{bmatrix}, \text{ and } \quad \pi = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}.$$

Explain, in words (not math), why the market is incomplete. What types of insurance would an agent be unable to buy in this setting?

(2) Use the algorithm described above, calculate the vector of Arrow-Debreu prices $\mathbf{q}^*$ defined above. Verify that the price vector is consistent with the lack of arbitrage opportunities. What is the implied SDF $\mathbf{M}^*$?

(3) What is the risk-free rate consistent with the absence of arbitrage and your estimate of $\mathbf{q}^*$?

[3] If the matrix notation is confusing, this just says that the A-D price in state s is a linear combination of individual asset payoffs ($\sum_{j=1}^{N} \beta_j X_{js}$).

[4] I've written a simple Python function called 'trim_redundant_rows' which can eliminate redundant rows from a matrix $\mathbf{X}$. This function uses the same linear algebra techniques that a standard regression package will use to identify and drop variables which are linearly dependent. However, in the example below, it should be easy to visually identify and exclude redundant rows.

[5] Note that it is the *fitted value* from a regression of $\mathbf{p}$ on $\mathbf{X}\mathbf{X}'$ that is unique, not necessarily the parameter vector β. If $rank(\mathbf{X}\mathbf{X}') < N$, then there will be multiple choices of $\mathbf{Z}$ (i.e., choices of a basis for the space of traded payoffs) which will all yield the same vector of A-D prices $\mathbf{q}^*$.

Is it unique or could it change if we received more information? Explain your reasoning.

(4) Suppose I add a new security, with payoff vector $(0, 0, 0, 1)'$, to the market, and its price is \$1. Now, are markets complete? Find the new vector of A-D prices $\mathbf{q}$.

(5) Recall that, in lecture, we argued that two assets which have the same covariance with the stochastic discount factor should have the same prices. Therefore, since prices have not changed, the difference between the SDFs found in parts ii and iv should be uncorrelated with the original set of payoffs. Verify that $E(M - M^*, X_3) = 0$ for $X_3 = (0, 1, 1, 1)$.