Business 35150 **John H. Cochrane**

Problem Set 7

Part I Short answer questions on readings.

Note, if I don't provide it, state which table, figure, or exhibit backs up your point

1. Mitchell and Pulvino

   (a) Figure 3, Table IV. What accounts for the huge difference between VWRA and RAIM portfolio returns?

   (b) What is the meaning of the fact that the two betas differ in Table IV? What kind of security does this suggest you use to benchmark merger "arbitrage?"

   (c) (Thought question for class, no need to write or hand in.) Figure 4 suggests all the information is in about 4 big outliers. Does this worry you?

   (d) The $\alpha_{\text{Mkt High}}$ in Table IV is positive and significant. This is also where the bent line of Figure 4 intersects the vertical axis, the intercept of the bent line. So, do we conclude that Merger "Arbitrage" is profitable after accounting for its option-like component?

   (e) In Table VI 2159, the effect seems much stronger in cash transactions. Does this argue against their story; i.e. should the betas be the same any way you do it?

   (f) How is Figure 5 / Table VIII looks like a repeat of Figure 4 / Table IV. How are they different? What are the advantages and disadvantages of the two approaches?

2. Asness Et. Al.

   (a) Why, according to Asness & co., might hedge funds seem to have returns that are smoother and lower beta than in fact?

   (b) If returns were independent over time, how would adding lagged returns affect betas – increase, decrease, or stay the same as you add lags?

   (c) Overall, when we add lagged betas, do hedge fund betas seem to increase, decrease, or stay the same?

   (d) Are alphas meaningful as we add lagged betas? What happens to alphas as we add lagged betas?

   (e) What interpretation do Asness & co. give to the difference between up and down market betas? Can you think of a different interpretation?

   (f) A conceptual issue for class discussion, no answer required (p. 10, bottom left). Suppose a manger moves in and out of the market, according to some signal, and does so correctly. Is this alpha? Should we correct for such time-varying beta in our performance attribution?

3. Lamont and Thaler

   (a) There are three ways to buy Palm stock each with a different price. What are they?

   (b) L&T document that short costs keep you from exploiting different prices *But why, according to them, are prices wrong in the first place*?

   (c) Looking at Figure 1-4, and 3 in particular, do negative stubs seem to quickly converge? Does news seem to affect the stub?

   (d) How is "real world" shorting different from our frictionless textbook? Note two extra costs and risks

   (e) Was there a lot of shorting in the "overpriced" subsidiaries? Was there more or less than in the parents? Did the sub shorting increase or decrease over time?

(f) If we can't short, let's buy November (data of spinoff) puts, or create a synthetic short position in options markets. Will this work and if not why not?

(g) How do turnover and institutional ownership of Palm compare to that of 3Com? What conclusions do L&T draw from these facts?

(h) What happened to 3Com price during this episode? What conclusions to L&T draw?

4. Cochrane Stocks as Money

(a) According to Cochrane, which of money and bonds is like which one of 3Com/Palm ?

(b) How is the "overpricing" of money associated with

i. Turnover

ii. Supply

iii. Short sales constraints

iv. "Specialness" of the security (Palm, money); presence of substitutes

(c) Is turnover associated with "overpricing" for 3Com/Palm in the right direction?

(d) How much does a typical Palm investor lose by holding Palm, not 3Com? Is this "a lot" or "not much"?

(e) What's the point of Figure 5? (Short)

(f) Wait, monetary theory says you are willing to put up with low returns on money because there is no substitute. If you want to bet on Palm, why not buy 3Com or use options instead? (Point to evidence here in Table 1, Figure 7.)

(g) To Cochrane, the fact that 3Com *fell* is explained. How? What do Lamont and Thaler say about it?

(h) Cochrane thinks one crucial special feature marks "bubbles" in addition to "I wish I sold yesterday" price rises and declines. What is it?

Part II Computer

You're ready. It's time to form some portfolios.

We're going to replicate Carhart's results, and then extend the idea a bit. In doing so, you learn how to form portfolios the way Fama and French do.

Load the matlab workspace fund_factor_data from the class website. This has all you need; once you execute the command load fund_factor_data, all the variables are there in your workspace for you to play with. x is a 576x3668 matrix of percent monthly returns on equity mutual funds. dates is a corresponding 576x1 matrix with the dates for the return observations. If a fund is not operating in a given date, either has not been started yet or went out of business you will see a NaN (not a number) code instead of a return in x. The workspace also contains Fama French factors and a momentum factor rmrf, smb, hml, umd and rf for the same time period. The x returns are total, so your first step is rx = x-rf.

The first thing we'll do is to replicate Carhart in this longer data. Form Carhart portfolios, sorting on one year returns. Start with observation 12 (December of the first year). Find the funds that have 12 months of returns leading up to that date, i.e. have numbers and not NaNs continuously for observations 1:12. Find the mean return over the first year for each such fund.

Now, form an equally weighted portfolio (i.e. just average the returns across funds) of the 1/10th worst of these funds, the next 1/10th, etc. to the best 1/10th, for the *next* year. The result should be 12 months of returns on 10 portfolios, valid from observation 13 (January, year 2) to 24.

Some of your funds will disappear over the return year. If you ignore them, taking the portfolio return over the non-NaN return for each month, that has the same effect as taking the money from a dead fund and redistributing it equally to all the others. Now you know why we do equal-weight portfolios!

Repeat this procedure at observation 24 (December, year 2) and so forth, to create 10 portfolios sorted on the basis of previous year's return. (Obviously, you'll do this in a loop. See the programming hints at the end of the problem set!) Include in your results the winner-loser (10-1) portfolio return.

Now comes the easy part

1. Evaluate these mutual-fund portfolios:

   (a) Make a table of mean returns and standard errors ($\sigma/\sqrt{T}$) for the 11 portfolios, to verify that the winners keep winning etc.

   (b) Include in your table the mean monthly return of each portfolio for the prior (formation) year. This should give you an idea of how strong the continuation in returns from year to year is underlying the portfolio sorts. Are the prior-year returns a lot different from the subsequent year returns?

   (c) Continuing this theme, find in each of the 10 portfolios the fraction of funds in that portfolio that beat the market in the following year, and the fraction that do worse than the market in the following year. Again, I'm trying to emphasize that just because the *portfolio* has a high mean return, there is a lot of risk to the *funds in the portfolio.* (Note, these fractions are actually not that close to 50%. Well, facts are facts.)

   (d) Then find alphas and betas etc. from time-series regressions for all 11 portfolios (use your program from before or my tsregress2) and add these to the table. Do this for

      i. The CAPM

      ii. The four factor model. You should get results quite similar to Carhart's.

2. Repeat, but sort the funds on prior 5 year returns. This is the most common industry practice. $\sigma/\sqrt{T}$ is pretty disastrous on 1 year returns (and not so great on 5 year returns!) but if someone has skill, we are surely going to get more signal/noise, skill/luck looking at 5 year histories. If there's skill, we should see 5 year returns continue more than 1 year returns continued. Let's see how it works. Make the same table of mean returns, 4F alphas and betas, etc. but using 5 year returns to form portfolios, and compare to the results using a one-year sort.

3. Let's replicate the Fama and French alpha t statistic calculation. To do this, find the 4 factor alpha for each fund and its t statistic. (I used only funds with at least 6 months of data so there are enough data points to run a sensible FF regression.)

   (a) Then make a plot of the cumulative distribution of alpha t statistics: For each t statistic, plot the percentage of funds with an alpha t statistic less than or equal to this value. Include in your plot the cumulative distribution predicted by the normal (0,1)* distribution. This will end up looking like FF's S-shaped graph. See the programming hints.

   (b) Find and report how many funds there are at the 5% points and median – what is the value of the alpha t statistic such that 5%, 50%, and 95% of the funds have lower alpha t statistics. Find the normal distribution's predictions for these points and compare. Are there more or fewer winners above the 5% tail of the t distribution than there should be? Are there more or less losers below the 5% tail of the t distribution than there should be? (This is FF's table, but just the 5, 50, and 95% points.)

   (c) Also find and plot the actual distribution of alphas. We don't have a distribution to compare it to (without running the simulation), but it's interesting to look. What percentage of funds have, say 1% per month (12% per year!) reported alpha?

   Note: here you're plotting cumulative distributions, s-shaped pictures as in the Fama French paper. Those are easy to compute, but take some practice to interpret. Histograms or smoothed histograms (probability density rather than cumulative) are easier to interpret but harder to produce. Mentally, though, you should make the connection between cumulative distributions and

densities. It's a good idea to sketch the density implied by your cumulative distributions. There are techniques to do this with the computer,(smooth it and take a derivative) but the problem set is long enough already.

*Distribution note: $\hat{\alpha}$ is the estimate for a fund, and $\sigma(\hat{\alpha}) = \sigma(\varepsilon)/\sqrt{T}$ is the standard error of the estimate. Thus, if $\hat{\alpha}$ is normal with mean zero and standard deviation $\sigma(\hat{\alpha})$, then $\hat{\alpha}/\sigma(\hat{\sigma})$ should be normal (0,1). The t statistic makes a little adjustment – we don't know $\sigma(\hat{\alpha})$ for sure, so variation in $\sigma(\hat{\alpha})$ in the denominator makes $\hat{\alpha}/\sigma(\hat{\alpha})$ a bit more volatile than it would be if we knew $\sigma(\hat{\alpha})$. As sample sizes get bigger, $\hat{\alpha}$ gets closer to normal, since it is a sample mean, and $\sigma(\hat{\alpha})$ gets closer to a constant, so the alpha t statistic approaches a N(0,1). So, Fama and French went to a LOT of work to compute a distribution that in the end is very close to normal. Compare their "sim" to a cumulative normal. But they got it really right.

Programming hints

Loops:

```
for date = 12:12:size(x,1)-12;
```

will step through the data on December only

NaN: Obviously, dealing with the NaN's is an issue. isnan is the key command. isnan(x) returns 1 where x is a NaN and 0 otherwise. For example

```
isnan([ 1 3 -5 NaN 10 NaN])
ans =
     0     0     0     1     0     1
```

Thus, ~isnan(x) delivers 1 where a matrix is a number and 0 where it isn't (~is the "not" operator). NaN's propagate through calculations. For example,

```
[NaN 2] + [ 2 4]
ans =
NaN 6
```

You should never be using the NaN, so if you find them in your result that's a sign you did something wrong and didn't avoid them.

The function nanmean() in the statistics toolbox will take a mean ignoring the NaN's. This is useful when taking portfolio average returns. nanmean(x,2) will take row means instead of column means just like mean(x,2)

Logical subscripts and find. You want to do something with only part of a data set, i.e. the part that is not NaN. One way to do this is with logical subscripts. For example,

```
>>>> x = [ 11 12 NaN 14 15 NaN 17 18]
x =
    11    12   NaN    14    15   NaN    17    18
>> y = ~isnan(x)
y =
     1     1     0     1     1     0     1     1
>> x(y)
ans =
    11    12    14    15    17    18
```

You see that x(y) pulled out the elements of x in the places where y is logical 1 (logical 1, the result of a true/false question, is different from real 1. If you try y = [1 1 0 1 1 0 1] here it won't work.) The find function (doc find for details) is also useful. Find tells you the indices of a matrix that meet some logical test,

```
>> y = find(~isnan(x))
y =
     1     2     4     5     7     8
>> x(find(~isnan(x)))
ans =
    11    12    14    15    17    18
```

Strategy There are lots of ways to attack this problem. For loops will be slower but perhaps much easier to program. Matrix operations are slicker but harder to get right and harder to debug.

I did it by matrices (of course). See the drawing at the end. First I found all the funds alive for the previous year with these NaN selectors. Then I attached the last year mean return to the next year's 12 monthly returns for each eligible fund. Then I have a 13 x many matrix [last year mean return; jan return; feb return; mar return....dec return], though some of the latter might be NaNs, and the columns track each eligible fund. Now we have to sort funds based on last return. I found useful the command

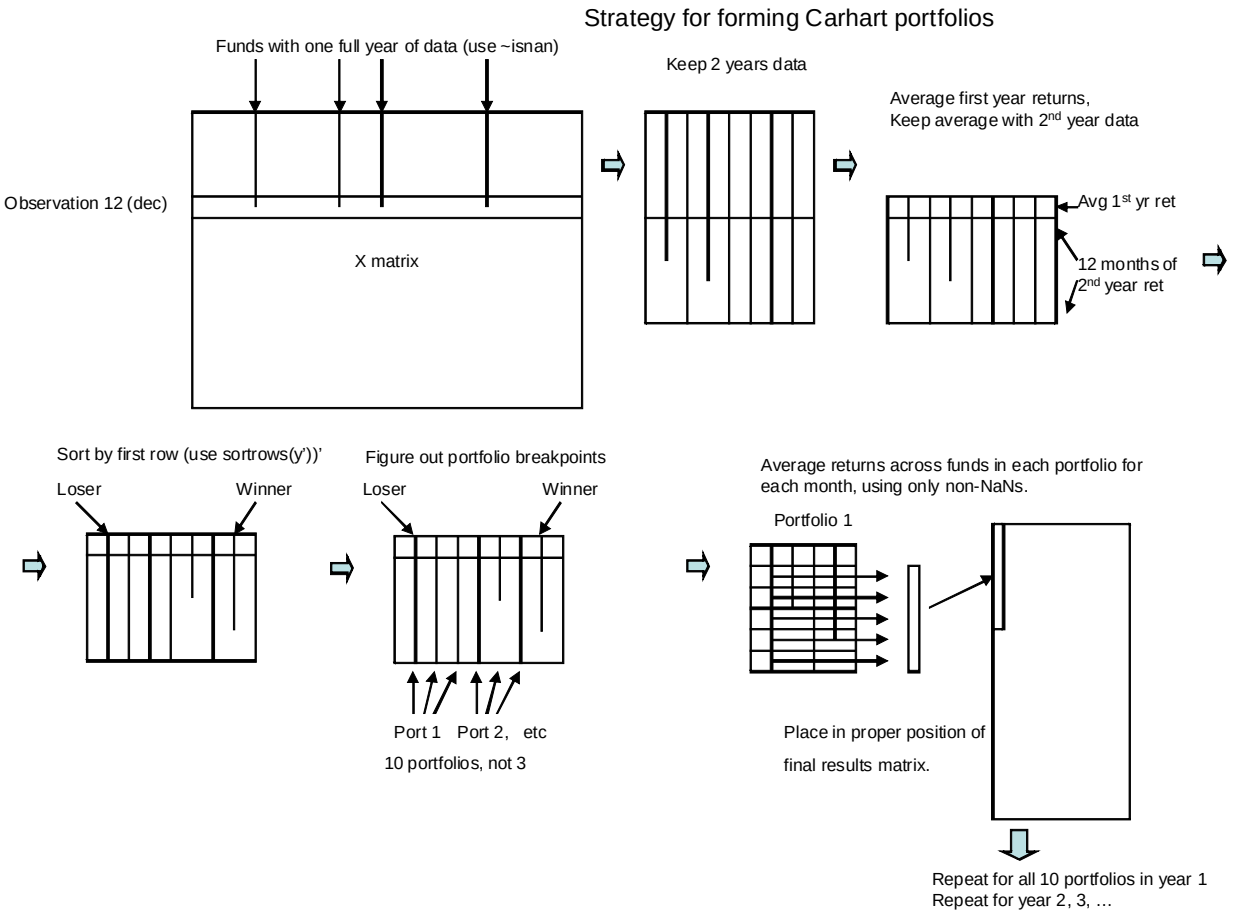
```
sortrows(matrix,column).
```

This sorts the rows of the matrix according to the value in the specified column. Since I want to sort the *columns* of my 13 x large matrix based on the first *row,* I did

```
sortrows(matrix',column)'.
```

Now I have a 13 x large matrix with losers on the left and winners on the right. Then I find the breakpoints. For example, if I have 1000 funds (13 x 1000 matrix), I want funds and column indices 1-100 in the first portfolio, 101-200 in the second portfolio, etc. Since the number of funds is not divisible by 10, you won't have exactly the same number in each portfolio. That's ok. All I have to do is find the average return *across funds* for each month in each portfolio, and only averaging over funds that don't have NaN returns. I wasn't clever enough to do this in a matrix, so I start with for i = 1:10 across portfolios and then for t = 1:12 over months and just took the average of all non-NaN returns in each portfolio in each month. Put them in the right slot of a big (dates x 10 portfolios) matrix and you're done. Remember mean(x) works on the columns of x. mean(x') or mean(x,2) lets you take a mean across rows. You can use the nanmean function to ignore the nans).

Obviously, forming the fund portfolios is the hard part, though once you've done one the others are minor variations. The rest is just tsregress2, so doing the three variations though interesting is not such a big programming deal.



For the Fama French question, you can find the valid data for each fund with

```
indx = find(~isnan(x(:,i)))
```

Then a regression with left and right hand variables

```
x(indx,i),f(indx)
```

will just run over the valid data. (I think matlab's regression command automatically does this. My tsregress does not.) Make sure to run the regression only when there are more than 6 data points!

Making cumulative distribution plots is easy. First sort the alphats from lowest to highest with sort(alphat). Then if you just plot

```
N = size(alphat,1);
plot(alphat,100*(1:N)/N);
```

You'll see a beautiful cumulative density. The median is

```
alphat(floor(N/2))
```

and so on.