Week 6 – Credit Risk

Imagine this

- Your company is trying to penetrate a new market. To do so it acquires several smaller competitors for your products and services. As you acquire the companies (NewCo, collectively), you also acquire their customers. . . and now your customers' ability to pay you.
- Not only that, but you have also taken your NewCo's supply chain. Your company also has to contend with the credit worthiness of NewCo's vendors. If they default you don't get supplied, you can't produce, you can't fulfill your customers, they walk.
- Your CFO has handed you the job of organizing your accounts receivable, understanding your customers' paying patterns, and more importantly their defaulting patterns.

Think about this

1. What are the key business questions you should ask about your customers' paying / defaulting patterns?
2. What systematic approach might you use to manage customer and counterparty credit risk?

Thinking...

Some ideas

1. Key business questions might be

- What customers and counterparties default more often than others?
- If customer default what can we recover?
- What is the total exposure that we experience at any given point in time?
- How far can we go with customers that might default?

2. Managing credit risk

- Set up a scoring system to accept new customers and track existing customers
- Monitor transitions of customers from one rating notch to another
- Build in early warning indicators of customer and counterparty default
- Build a playbook to manage the otherwise insurgent and unanticipated credit events that can overtake your customers and counterparties

Whitman
SCHOOL *of* MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

Previously on Financial Analytics...

Topics we got to in the last sessions:

- Explored stylized fact of financial market data
- Learned just how insidious volatility really is
- Acquired new tools like `acf`, `pacf`, `ccf` to explore time series

This week We Will. . .

- Use actual transaction and credit migration data
- Write credit simulations using Markov chains
- Understand hazard rates, transition probabalities
- Predict default and generating loss distributions
- Analyze potential losses from the accounts receivable credit portfolio

Whitman
SCHOOL of MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

New customers!

Not so fast! Let's load the credit profiles of our newly acquired customers. Here is what was collected these past few years:

```r
firm.profile <- read.csv("data/creditfirmprofile.csv")
head(firm.profile)
```

```
##   id start.year default  wcTA  reTA ebitTA mktcapTL   sTA
## 1  1       2010       0 0.501 0.307  0.043    0.956 0.335
## 2  1       2011       0 0.550 0.320  0.050    1.060 0.330
## 3  1       2012       0 0.450 0.230  0.030    0.800 0.250
## 4  1       2013       0 0.310 0.190  0.030    0.390 0.250
## 5  1       2014       0 0.450 0.220  0.030    0.790 0.280
## 6  1       2015       0 0.460 0.220  0.030    1.290 0.320
```

Recorded for each of several years from 2006 through 2015 each firm's (customer's) indicator as to whether they defaulted or not (1 or 0).

```r
summary(firm.profile)
```

```
##        id          start.year      default           wcTA
##  Min.   :  1.0   Min.   :2006   Min.   :0.000   Min.   :-2.2400
##  1st Qu.:192.0   1st Qu.:2009   1st Qu.:0.000   1st Qu.: 0.0300
##  Median :358.0   Median :2011   Median :0.000   Median : 0.1200
##  Mean   :356.3   Mean   :2011   Mean   :0.018   Mean   : 0.1426
##  3rd Qu.:521.0   3rd Qu.:2013   3rd Qu.:0.000   3rd Qu.: 0.2400
##  Max.   :830.0   Max.   :2015   Max.   :1.000   Max.   : 0.7700
##       reTA              ebitTA            mktcapTL           sTA
##  Min.   :-3.3100   Min.   :-0.59000   Min.   : 0.020   Min.   :0.0400
##  1st Qu.: 0.0900   1st Qu.: 0.04000   1st Qu.: 0.620   1st Qu.:0.1700
##  Median : 0.2200   Median : 0.05000   Median : 1.140   Median :0.2600
##  Mean   : 0.2104   Mean   : 0.05181   Mean   : 1.954   Mean   :0.3036
##  3rd Qu.: 0.3700   3rd Qu.: 0.07000   3rd Qu.: 2.240   3rd Qu.:0.3700
##  Max.   : 1.6400   Max.   : 0.20000   Max.   :60.610   Max.   :5.0100
```


Several risk factors can contribute to credit risk:

1. Working Capital risk is measured by the Working Capital / Total Assets ratio `wcTA`.

When this ratio is zero, current assets are matched by current liabilities. When positive (negative), current assets are greater (lesser) than current liabilities. The risk is that there are very large current assets of low quality to feed revenue cash flow. Or the risk is that there are high current liabilities balances creating a hole in the cash conversion cycle and thus a possibility of lower than expected cash flow.

2. Internal Funding risk is measured by the Retained Earnings / Total Assets ratio `reTA`.

Retained Earnings measures the amount of net income plowed back into the organization. High ratios signal strong capability to fund projects with internally generated cash flow. The risk is that if the organization faces extreme changes in the market-place, there is not enough internally generated funding to manage the change.

3. Asset Profitability risk is measured by EBIT / Total Assets ratio.

- This is the return on assets used to value the organization. The risk is that

1. EBIT is too low or even negative and thus the assets are not productively reaching revenue markets or efficiently using the supply change, or both, all resulting in too low a cash flow to sustain operations and investor expectations, and
2. This metric falls short of investors' minimum required returns, and thus investors' expectations are dashed to the ground, they sell your stock, and with supply and demand simplicity your stock price falls, along with your equity-based compensation.

4. Capital Structure risk is measured by the Market Value of Equity / Total Liabilities ratio `mktcapTL`.

If this ratio deviates from industry norms, or if too low, then shareholder claims to cash flow for responding to market changes will be impaired. The risk is similar to internal funding risks, but carries the additional market perception that the organization is unwilling or unable to manage change.

5. Asset Efficiency risk is measured by the Sales / Total Assets ratio `sTA`.

If the ratio is too low, then the organization risks two things: 1. Ability to support sales with assets, and 2. The overburden of unproductive assets unable to support new projects through additions to retained earnings or in meeting liability commitments.

- Let's load customer credit migration data.
- This data records the start rating, end rating, and timing for each of 830 customers as their business, and the recession, affected them.

```
firm.migration <- read.csv("data/creditmigration.csv")
head(firm.migration)
```

```
##   id start.date start.rating end.date end.rating time start.year end.year
## 1  1      40541            6    42366          6 1825       2010     2015
## 2  2      41412            6    42366          6  954       2013     2015
## 3  3      40174            5    40479          6  305       2009     2010
## 4  3      40479            6    40905          5  426       2010     2011
## 5  3      40905            5    42366          5 1461       2011     2015
## 6  4      40905            5    41056          6  151       2011     2012
```

- Notice that the dates are given in number of days from January 1, 1900.
- Ratings are numerical.

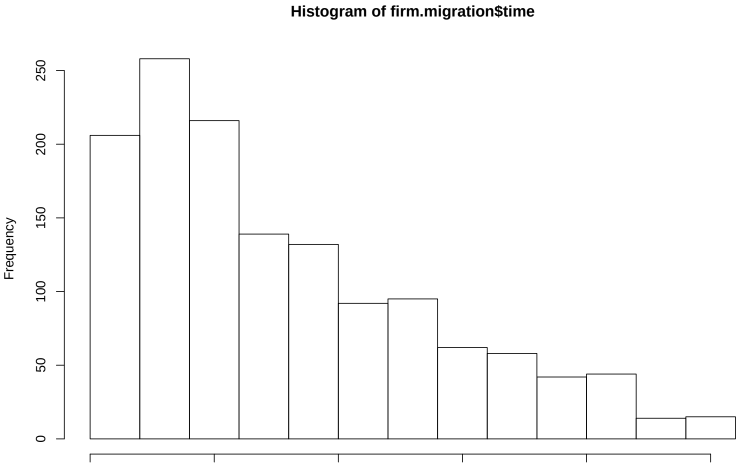
```r
summary(firm.migration)
```

```
##        id          start.date     start.rating      end.date    
##  Min.   :  1.0   Min.   :39951   Min.   :1.000   Min.   :39960  
##  1st Qu.:230.0   1st Qu.:40418   1st Qu.:3.000   1st Qu.:41170  
##  Median :430.0   Median :40905   Median :4.000   Median :42143  
##  Mean   :421.8   Mean   :40973   Mean   :4.041   Mean   :41765  
##  3rd Qu.:631.0   3rd Qu.:41421   3rd Qu.:5.000   3rd Qu.:42366  
##  Max.   :830.0   Max.   :42366   Max.   :8.000   Max.   :42366  
##  NA's   :1709    NA's   :1709    NA's   :1709    NA's   :1709   
##    end.rating         time          start.year      end.year   
##  Min.   :1.000   Min.   :   0.0   Min.   :2009   Min.   :2009  
##  1st Qu.:3.000   1st Qu.: 344.0   1st Qu.:2010   1st Qu.:2012  
##  Median :4.000   Median : 631.0   Median :2011   Median :2015  
##  Mean   :4.121   Mean   : 791.5   Mean   :2012   Mean   :2014  
##  3rd Qu.:5.000   3rd Qu.:1157.0   3rd Qu.:2013   3rd Qu.:2015  
##  Max.   :8.000   Max.   :2415.0   Max.   :2015   Max.   :2015  
##  NA's   :1709    NA's   :1709     NA's   :1709   NA's   :1709  
```

```r
firm.migration <- na.omit(firm.migration)
firm.migration$time <- as.numeric(firm.migration$time)
```

- An interesting metric is in `firm.migration$time`.
- This field has records of the difference between `end.date` and `start.date` in days between start ratings and end ratings.
- This duration is sometimes called "dwell time."

```
hist(firm.migration$time)
```



- Let's now merge the two credit files by starting year.
- This will ("inner") join the data so we can see what customer conditions might be consistent with a rating change and rating dwell time.
- The two keys are `id` and `start.year`.

```r
firm.credit <- merge(firm.profile, firm.migration,
    by = c("id", "start.year"))
head(firm.credit)
```

```
##    id start.year default  wcTA  reTA ebitTA mktcapTL   sTA start.date
## 1   1       2010       0 0.501 0.307  0.043    0.956 0.335      40541
## 2  10       2014       0 0.330 0.130  0.030    0.330 0.140      42001
## 3 100       2010       0 0.170 0.240  0.080    5.590 0.270      40510
## 4 100       2013       0 0.140 0.260  0.020    2.860 0.260      41635
## 5 106       2013       0 0.040 0.140  0.060    0.410 0.320      41605
## 6 106       2013       0 0.040 0.140  0.060    0.410 0.320      41421
##   start.rating end.date end.rating time end.year
## 1            6    42366          6 1825     2015
## 2            5    42366          5  365     2015
## 3            4    41635          3 1125     2013
## 4            3    42366          3  731     2015
## 5            5    42366          5  761     2015
## 6            6    41605          5  184     2013
```

```r
dim(firm.credit)
```

Try this

- The shape of `firm.migration$time` suggests a gamma or an exponential function. But before we go off on that goose chase, let's look at the inner-joined data to see potential differences in rating.
- Let's reuse this code from Week 2:

```
library(dplyr)

# 1: filter to keep one state. Not
# needed (yet...)
pvt_table <- firm.credit  # filter(firm.credit, xxx %in% 'NY')

# 2: set up data frame for by-group
# processing.
pvt_table <- group_by(pvt_table, default,
    end.rating)

# 3: calculate the three summary
# metrics
options(dplyr.width = Inf)  # to display all columns
pvt_table <- summarise(pvt_table, time.avg = mean(time)/365,
    ebitTA.avg = mean(ebitTA), sTA.avg = mean(sTA))
```

Run the code and comment.

Result

Display in a nice table:

```
knitr::kable(pvt_table)
```

default	end.rating	time.avg	ebitTA.avg	sTA.avg
0	1	3.021005	0.0491667	0.2633333
0	2	2.583007	0.0529762	0.3250000
0	3	2.468776	0.0520103	0.2904124
0	4	2.106548	0.0455495	0.2655495
0	5	1.912186	0.0501042	0.2883333
0	6	1.697821	0.0504886	0.3041477
0	7	1.672251	0.0494286	0.3308571
0	8	2.467123	0.0591667	0.3208333
1	1	2.413699	-0.0100000	0.4100000
1	2	2.497717	0.0216667	0.2683333
1	3	2.613699	0.0200000	0.2400000
1	6	2.242466	0.0350000	0.1150000
1	7	3.002740	0.0500000	0.3000000

- Defaulting (`default` $= 1$) firms have very low EBIT returns on Total Assets as well as low Sales to Total Assets... as expected.
- They also spent a lot of time (in 365 day years) in rating 7 – equivalent to a "C" rating at S&P.

Now let's use the credit migration data

- To understand the probability of default as well as the probabilities of

1. Being in other ratings or
2. Migrating from one rating to another.

Whitman
SCHOOL of MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

It depends

- Most interesting examples in probability have a little dependence added in: "If it rained yesterday, what is the probability it rains today?"
- We can use this idea to generate weather patterns and probabilities for some time in the future.
- In market risk, we can use this idea to generate the persistence of consumption spending, inflation, and the impact on zero coupon bond yields.
- In credit, dependence can be seen in credit migration: if an account receivable was A rated this year, what are the odds this receivable be A rated next year?

Enter A.A. Markov

A. A. Markov

Suppose we have a sequence of T observations, $\{X_t\}_1^T$, that are dependent. In a time series, what happens next depends on what happened before:

$$p(X_1, X_2, ..., X_T) = p(X_1)p(X_2|X_1)...p(X_t|X_{t-1}, ..., X_1)$$

With Markov dependence each outcome `only` depends on the one that came before.

$$p(X_1, X_2, ..., X_T) = p(X_1)\prod_{s=2}^{T} p(X_s|X_{s-1})$$

We have already encountered this when we used the functions `acf` and `ccf` to explore very similar dependencies in macro-financial time series data. Markov dependence is equivalent to an `AR(1)` process (today = some part of yesterday plus some noise).

To generate a Markov chain, we

1. Set up the conditional distribution.
2. Draw the initial state of the chain.
3. For every additional draw, use the previous draw to inform the new one.

Now for a very simple (but oftentimes useful) credit model:

- If a receivable's issuer (a customer) last month was investment grade, then this month's chance of also being investment grade is 80%.
- If a receivable's issuer last month was **not** investment grade, then this month's chance of being investment grade is 20%.

Try this

- Simulate monthly for 5 years.
- Here we parameterize even the years and calculate the number of months in the simulation. We set up a dummy `investment.grade` variable with `NA` entries into which we deposit 60 dependent coin tosses using the binomial distribution.
- The probability of success (state = "Investment Grade") is overall 80% and is composed of a long run 20% (across the 60 months) plus a short run 60% (of the previous month).
- Again the similarity to an autoregressive process here with lags at 1 month and 60 months

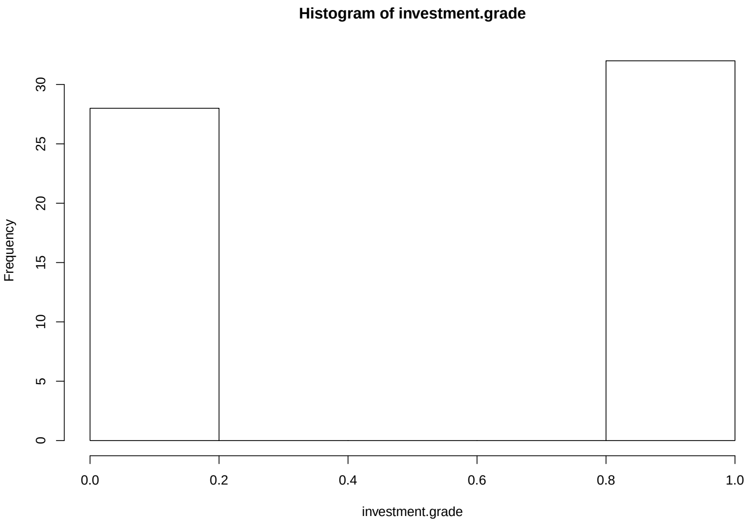
1. Run the following.
2. Look up `rbinom(n, size, prob)` (coin toss random draws) to see the syntax.
3. Comment on what happens when you set the `long.run` rate to 0% and the `short.run` rate to 80%.

```r
N.years <- 5
N.months <- N.years * 12
investment.grade <- rep(NA, N.months)
investment.grade[1] <- 1
long.run <- 0.5
short.run <- 0
for (month in 2:N.months) {
    investment.grade[month] <- rbinom(1,
        1, long.run + short.run * investment.grade[month -
            1])
}
hist(investment.grade)
```

- The `for (month in 2:N.months)` loop says "For each month starting at month 2, perform the tasks in the curly brackets (`{}`) until, and including, `N.months`"

Thinking...

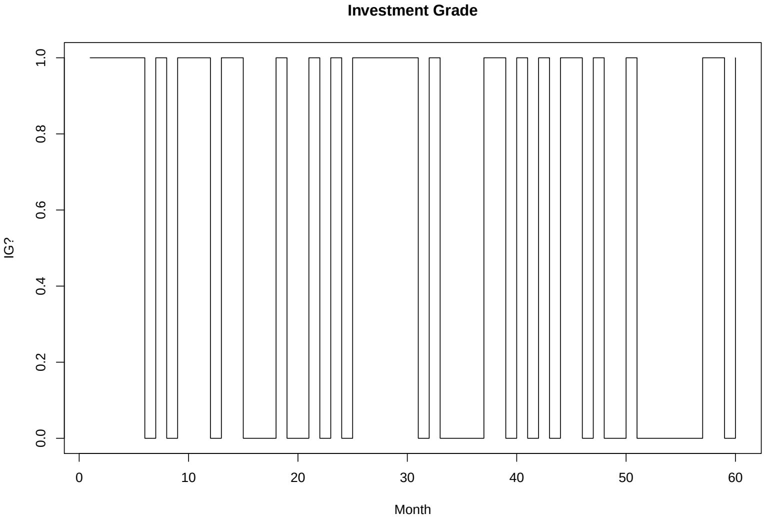
Some results



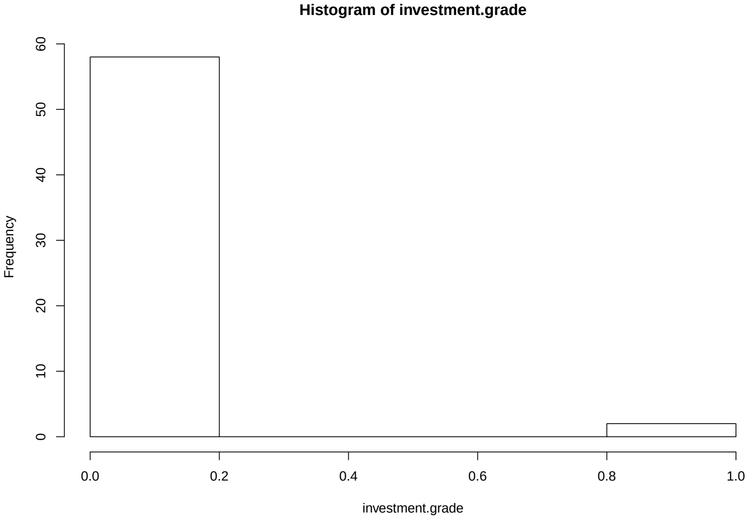
Almost evenly ditributed probabilities

And this as well...

```r
plot(investment.grade, main = "Investment Grade",
    xlab = "Month", ylab = "IG?", ty = "l")
```


Investment Grade
IG?
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
Month

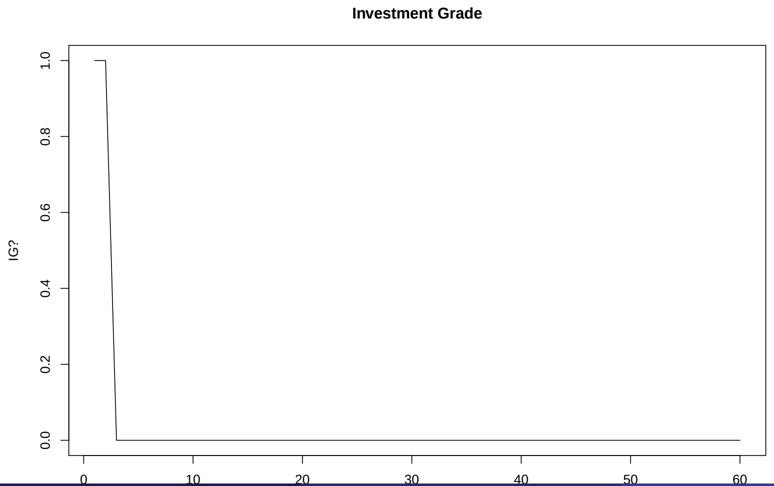
Now to look at a scenario with long-run = 0.0, and short-run = 0.80



Much different now with more probability concentrated in lower end of investment grade scale.

Next plot the up and down transitions between investment grade and not-investment grade using lines to connect up to down transitions. Now this looks more like a bull and bear graph.

```
plot(investment.grade, main = "Investment Grade",
    xlab = "Month", ylab = "IG?", ty = "l")
```



Whitman
SCHOOL of MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

In our (very) crude credit model...

... transitions are represented as a matrix: Q_{ij} is $P(X_t = j | X_{t-1} = i)$ where, i is the start state ("Investment Grade"), and j is the end state ("not-Investment Grade"). Here is a transition matrix to encapsulate this data.

```r
(transition.matrix <- matrix(c(0.8, 0.2,
    0.2, 0.8), nrow = 2))
```

```
##      [,1] [,2]
## [1,]  0.8  0.2
## [2,]  0.2  0.8
```

This function will nicely simulate this and more general random Markov chains.

```
rmarkovchain <- function(n.sim, transition.matrix,
    start = sample(1:nrow(transition.matrix),
        1)) {
    result <- rep(NA, n.sim)
    result[1] <- start
    for (t in 2:n.sim) result[t] <- sample(ncol(transition.matrix),
        1, prob = transition.matrix[result[t -
            1], ])
    return(result)
}
```


Try this

1. Run a 1000 trial Markov chain with the 2-state `transition.matrix` and save to a variable called 'markov.sim'.
2. Use the `table` function to calculate how many 1s and 2s are simulated.

Thinking...

Results

Many trials (and tribulations...)

```
markov.sim <- rmarkovchain(1000, transition.matrix)
head(markov.sim)
```

```
## [1] 1 1 2 2 2 2
```

Tabulate

```
ones <- which(markov.sim[-1000] == 1)
twos <- which(markov.sim[-1000] == 2)
state.one <- signif(table(markov.sim[ones +
    1])/length(ones), 3)
state.two <- signif(table(markov.sim[twos +
    1])/length(twos), 3)
(transition.matrix.sim <- rbind(state.one,
    state.two))
```

```
##                1     2
## state.one 0.810 0.190
## state.two 0.189 0.811
```

- Pretty close to `transition.matrix`.
- Law of large numbers would say we converge to these values.
- `which` sets up two indexes to find where the 1s and 2s are in `markov.sim`.
- `signif` with 3 means use 3 significant digits
- `table` tabulates the number of 1 states and 2 states simulated.

Whitman
SCHOOL of MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

- Next let's develop an approach to estimating transition probabilities using observed rates of credit migration from one rating to another.
- These could be simulated... if we are not very sure of future trends.

Whitman
SCHOOL *of* MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

Generating some hazards

- Let's set up a more realistic situation.
- Suppose we have annual hazard rates λ for each of four in-house credit ratings for a portfolio of accounts receivable. (We can use a pivot table to get these rates...)
- Suppose that we know the number of accounts that transition from one rating (start state) to another rating (end state) in a unit of time (here a year).
- We define the hazard rate λ as the number of accounts that migrate from one rating to another rating (per year) divided by the number of all the accounts that year at that starting rating.

- Suppose we have N ratings which in Markov-speak are states of being in a credit rating.
- Transition probabilities are summarized by a N row $\times$ N column "generator" matrix $\Lambda = (\lambda_{ij})$. Here the row index is $i = 1...N$ starting credit rating states and the column index is $j = 1...N$ ending credit rating states for a customer.
- Over any small (and getting ever smaller...) time step of duration dt the probability of a transition from rating i to j is given approximately by $\lambda_{ij} dt$.
- The probability of moving to any other rating from the starting rating is $\Sigma_{i \neq j}$ so that staying at rating i is given by $1 - \Sigma_{i \neq j} \lambda_{ij}$ all along the diagonal of the matrix.

A little more formally, and usefully, we can define for transitions from a starting state rating i to an ending state rating j over an interval of time $s = [0, t]$, the hazard rate for $i \neq j$,

$$\lambda_{ij} = \frac{N_{ij}}{\int_0^t Y_i(s)ds},$$

where N_{ij} is the number of accounts that made the transition from one rating i to another, different rating j, and Y_i is the number of accounts rated in the starting state rating i at the beginning of the time interval $(t, t + dt)$.

- The $\int$ sign is again (remember the term structure of interest rates) the cumulative sum of the product of the number of accounts times the time an account is in a transition from one credit rating state to another (the ds = a change in times).
- Using this formula, the math to generate ratings migration probabilities is fairly straightforward with a couple of kinks. We will use the `expm` package to calculate the matrix exponent.
- The probability of transitions is $P = exp(\Lambda)$. We can think of probabilities as discounts. They range from 0 to 1 like the present value of $1. Each rate , $\lambda_{ij} dt$, across a segment of time dt is just like a forward interest rate in form as it is the change in probability (present value) from one date to another t to $t + dt$.

- First suppose we already know the hazard rates for each starting state rating. We will assign the `D.rating` hazard rate as a 0 throughout as this is the last state; that is, there is no transition from this rating to another rating.
- Then create a λ matrix by concatenating the rows of hazard rates, and see that the diagonals are zero.
- By definition, if an account stays in a rating, the diagonal must be the negative of the row sum of this matrix, where we use the apply function on the first, the row, dimension of the matrix. We put the negative row sum into the diagonal and now we have a proper hazard, or also called generator, matrix.
- Now just raise the hazard matrix to the exponent power. The result is the probability transition matrix.

Whitman
SCHOOL of MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE

Now for the future

- Decision makers now want to use this model to look into the future. Using the hazard rates to develop policies for our accounts receivable, and ultimately customer and counterparty (e.g., vendors) relationships.
- Let's use our credit migration data set to estimate hazard rates and then estimate transition probabilities. Once we have these we can use default probabilities, along with exposures, to calculate Value at Risk and Expected Shortfall amounts. Then we will know really how risky our acquired customers are.

Read in the credit migration data set and set factors and numeric types for the columns of data. Also get rid of any `NA`'s.

```
output.Transitions <- read.csv("data/creditmigration.csv")
output.Transitions$start.rating <- as.factor(output.Transitions$start.rating)
output.Transitions$end.rating <- as.factor(output.Transitions$end.rating)
output.Transitions$time <- as.numeric(output.Transitions$time)
transition.Events <- na.omit(output.Transitions)
```

Let's inspect the data.

```r
head(transition.Events, n = 5)
```

```
##   id start.date start.rating end.date end.rating time start.year end.year
## 1  1      40541            6    42366          6 1825       2010     2015
## 2  2      41412            6    42366          6  954       2013     2015
## 3  3      40174            5    40479          6  305       2009     2010
## 4  3      40479            6    40905          5  426       2010     2011
## 5  3      40905            5    42366          5 1461       2011     2015
```

```r
dim(transition.Events)
```

```
## [1] 1373    8
```

```r
summary(transition.Events)
```

```
##        id          start.date      start.rating    end.date    
##  Min.   :  1.0   Min.   :39951   3      :351   Min.   :39960  
##  1st Qu.:230.0   1st Qu.:40418   4      :348   1st Qu.:41170  
##  Median :430.0   Median :40905   5      :212   Median :42143  
##  Mean   :421.8   Mean   :40973   6      :189   Mean   :41765  
##  3rd Qu.:631.0   3rd Qu.:41421   2      :179   3rd Qu.:42366  
##  Max.   :830.0   Max.   :42366   7      : 61   Max.   :42366  
##                                  (Other): 33                  
##    end.rating       time          start.year      end.year   
##  3      :333   Min.   :   0.0   Min.   :2009   Min.   :2009  
##  4      :332   1st Qu.: 344.0   1st Qu.:2010   1st Qu.:2012  
##  5      :209   Median : 631.0   Median :2011   Median :2015  
##  6      :190   Mean   : 791.5   Mean   :2012   Mean   :2014  
##  2      :177   3rd Qu.:1157.0   3rd Qu.:2013   3rd Qu.:2015  
##  7      : 82   Max.   :2415.0   Max.   :2015   Max.   :2015  
##  (Other): 50
```

Build we must

The work flow to build a hazard rate estimation model from data.

- Tabulate the count of transitions from each rating to other rating and to itself.
- Calculate the time spent from each start rating to all end ratings.
- Divide counts by time spent.

These steps will calculate the hazard rate formula:

$$\lambda_{ij} = \frac{N_{ij}}{\int_0^t Y_i(s)ds}$$

Now we begin to estimate:

```r
(Nij.table <- table(transition.Events$start.rating,
    transition.Events$end.rating))
```

```
##
##       1   2   3   4   5   6   7   8
##   1  17   2   1   0   0   0   0   0
##   2   9 146  24   0   0   0   0   0
##   3   0  29 271  49   2   0   0   0
##   4   0   0  35 251  51   8   3   0
##   5   0   0   2  32 125  50   3   0
##   6   0   0   0   0  28 116  40   5
##   7   0   0   0   0   2  14  33  12
##   8   0   0   0   0   1   2   3   7
```

Calculate the dwell time in transitions from the 8 start ratings (all 8 of them...)

```
(RiskSet <- by(transition.Events$time,
    transition.Events$start.rating, sum))
```

```
## transition.Events$start.rating: 1
## [1] 25683
## --------------------------------------------------------
## transition.Events$start.rating: 2
## [1] 172285
## --------------------------------------------------------
## transition.Events$start.rating: 3
## [1] 318679
## --------------------------------------------------------
## transition.Events$start.rating: 4
## [1] 279219
## --------------------------------------------------------
## transition.Events$start.rating: 5
## [1] 129886
## --------------------------------------------------------
## transition.Events$start.rating: 6
## [1] 120696
## --------------------------------------------------------
## transition.Events$start.rating: 7
## [1] 29737
## --------------------------------------------------------
## transition.Events$start.rating: 8
## [1] 10577
```


- Using the levels in start and end ratings (1 to 8 each), build a matrix of start ratings each to all of the end ratings by replicating columns of times...
- Divide by 365.25 (to get the leap year in ...) and convert to annual times.

```
I.levels <- levels(transition.Events$start.rating)
J.levels <- levels(transition.Events$end.rating)
(Ni.matrix <- matrix(nrow = length(I.levels),
    ncol = length(J.levels), as.numeric(RiskSet)/365,
    byrow = FALSE))
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,]  70.36438  70.36438  70.36438  70.36438  70.36438  70.36438  70.36438
## [2,] 472.01370 472.01370 472.01370 472.01370 472.01370 472.01370 472.01370
## [3,] 873.09315 873.09315 873.09315 873.09315 873.09315 873.09315 873.09315
## [4,] 764.98356 764.98356 764.98356 764.98356 764.98356 764.98356 764.98356
## [5,] 355.85205 355.85205 355.85205 355.85205 355.85205 355.85205 355.85205
## [6,] 330.67397 330.67397 330.67397 330.67397 330.67397 330.67397 330.67397
## [7,]  81.47123  81.47123  81.47123  81.47123  81.47123  81.47123  81.47123
## [8,]  28.97808  28.97808  28.97808  28.97808  28.97808  28.97808  28.97808
##           [,8]
## [1,]  70.36438
## [2,] 472.01370
## [3,] 873.09315
## [4,] 764.98356
## [5,] 355.85205
## [6,] 330.67397
## [7,]  81.47123
## [8,]  28.97808
```

- The `Nij.table` tabulates the count of transitions from start to end.
- The `Ni.matrix` gives us the row sums of transition times (“spell” or “dwelling time”) for each starting state in years (and fractions thereof. . .).

Now we can estimate a simple hazard rate matrix. This looks a bit like the formula:

```
(lambda.hat <- Nij.table/Ni.matrix)
```

```
##
##              1           2           3           4           5
##   1 0.241599502 0.028423471 0.014211735 0.000000000 0.000000000
##   2 0.019067243 0.309313057 0.050845982 0.000000000 0.000000000
##   3 0.000000000 0.033215242 0.310390707 0.056122305 0.002290706
##   4 0.000000000 0.000000000 0.045752617 0.328111626 0.066668099
##   5 0.000000000 0.000000000 0.005620313 0.089925011 0.351269575
##   6 0.000000000 0.000000000 0.000000000 0.000000000 0.084675548
##   7 0.000000000 0.000000000 0.000000000 0.000000000 0.024548542
##   8 0.000000000 0.000000000 0.000000000 0.000000000 0.034508840
##
##              6           7           8
##   1 0.000000000 0.000000000 0.000000000
##   2 0.000000000 0.000000000 0.000000000
##   3 0.000000000 0.000000000 0.000000000
##   4 0.010457741 0.003921653 0.000000000
##   5 0.140507830 0.008430470 0.000000000
##   6 0.350798701 0.120965069 0.015120634
##   7 0.171839796 0.405050947 0.147291253
##   8 0.069017680 0.103526520 0.241561880
```


Back to the definition of the λ hazard rate matrix: the diagonal of a generator matrix is the negative of the sum of off-diagonal elements row by row.

```r
# Add default row and correct
# diagonal
lambda.hat <- lambda.hat[-8, ]
lambda.hat.diag <- rep(0, dim(lambda.hat)[2])
lambda.hat <- rbind(lambda.hat, lambda.hat.diag)
diag(lambda.hat) <- lambda.hat.diag
rowsums <- apply(lambda.hat, 1, sum)
diag(lambda.hat) <- -rowsums
# check for valid generator
apply(lambda.hat, 1, sum)
```

```
##               1               2               3               4
##   -1.734723e-18   -3.469447e-18    1.734723e-18    1.127570e-17
##               5               6               7 lambda.hat.diag
##   -2.602085e-18   -6.938894e-18   -6.938894e-18    0.000000e+00
```

```r
dim(lambda.hat)
```

```
## [1] 8 8
```

- The `apply` statement calculates the row sums of the `lambda.hat` matrix.
- If the sums are zero, then we correctly placed the diagonals.

```
lambda.hat
```

```
##                           1           2            3           4
## 1               -0.04263521  0.02842347  0.014211735  0.00000000
## 2                0.01906724 -0.06991323  0.050845982  0.00000000
## 3                0.00000000  0.03321524 -0.091628253  0.05612230
## 4                0.00000000  0.00000000  0.045752617 -0.12680011
## 5                0.00000000  0.00000000  0.005620313  0.08992501
## 6                0.00000000  0.00000000  0.000000000  0.00000000
## 7                0.00000000  0.00000000  0.000000000  0.00000000
## lambda.hat.diag  0.00000000  0.00000000  0.000000000  0.00000000
##                            5           6            7          8
## 1                0.000000000  0.00000000  0.000000000 0.00000000
## 2                0.000000000  0.00000000  0.000000000 0.00000000
## 3                0.002290706  0.00000000  0.000000000 0.00000000
## 4                0.066668099  0.01045774  0.003921653 0.00000000
## 5               -0.244483624  0.14050783  0.008430470 0.00000000
## 6                0.084675548 -0.22076125  0.120965069 0.01512063
## 7                0.024548542  0.17183980 -0.343679591 0.14729125
## lambda.hat.diag  0.000000000  0.00000000  0.000000000 0.00000000
```

Try this

Now generate the transition matrix using the matrix exponent function. Use the `Matrix` package to calculate the matrix exponent

```
require(Matrix)
P.hat <- expm(lambda.hat)
```

Look up the so-called matrix exponent in help (`??expm`) as well. You could also use the `expm` package.

Thinking. . .

Results

We calculate P.hat, the estimator of the probability of transitions from start.rating to end.rating.

```
# The matrix exponential Annual
# transition probabilities
require(Matrix)
(P.hat <- as.matrix(expm(lambda.hat)))
```

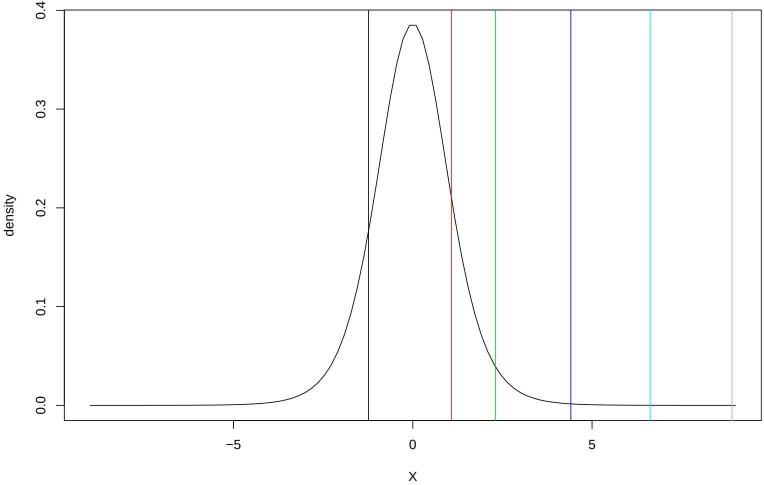
```
##                           1            2            3            4
## 1              9.585197e-01 2.709947e-02 1.397624e-02 0.0003788324
## 2              1.803109e-02 9.335132e-01 4.706561e-02 0.0013011506
## 3              2.959594e-04 3.066299e-02 9.143909e-01 0.0504916113
## 4              4.457478e-06 6.926947e-04 4.125191e-02 0.8846163306
## 5              6.280975e-07 1.017064e-04 6.540354e-03 0.0751252452
## 6              1.270261e-08 2.695916e-06 2.501585e-04 0.0031761461
## 7              3.983395e-09 8.602456e-07 8.202232e-05 0.0010478460
## lambda.hat.diag 0.000000e+00 0.000000e+00 0.000000e+00 0.0000000000
##                           5            6            7            8
## 1              2.293462e-05 2.240489e-06 5.793996e-07 2.965535e-08
## 2              7.902795e-05 7.777942e-06 2.011272e-06 1.035575e-07
## 3              3.560428e-03 4.688817e-04 1.209161e-04 8.270560e-06
## 4              5.605507e-02 1.299617e-02 3.999003e-03 3.843666e-04
## 5              7.904359e-01 1.130189e-01 1.299505e-02 1.782205e-03
## 6              6.867445e-02 8.147624e-01 9.206368e-02 2.107050e-02
## 7              2.399937e-02 1.317139e-01 7.170047e-01 1.261513e-01
## lambda.hat.diag 0.000000e+00 0.000000e+00 0.000000e+00 1.000000e+00
```

Most of our accounts will most probably stay in their initial ratings. We will be most interested in ratings 6 and 7 and their transitions to rating 8, default.

What does that mean?

Now let's replot the nondefaulting state distributions with thresholds using the estimated transition probabilities.

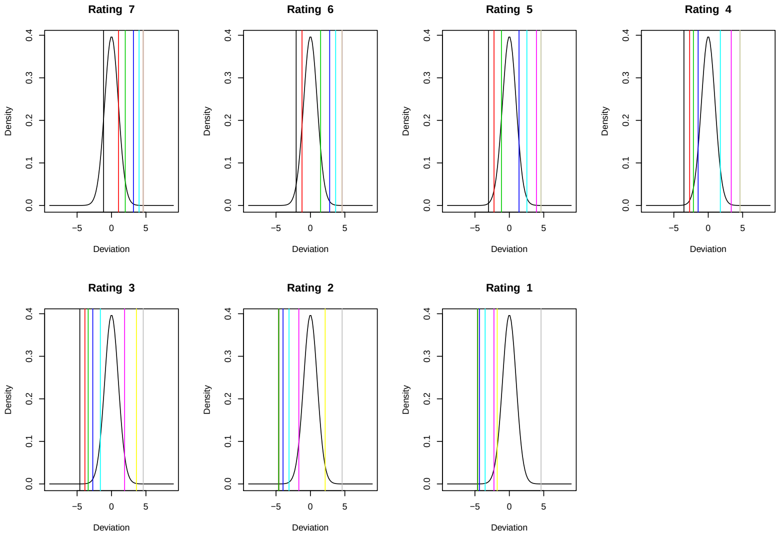
```
P.reverse <- P.hat[8:1, 8:1]  # Use P.hat now
P.reverse <- P.reverse[-1, ]  #without rating 8 state transitions
# select the 7th rating transition
# probabilities
seven.probs <- P.reverse[1, ]
seven.cumprobs <- cumsum(seven.probs)
seven.cumprobs <- pmin(0.99999, pmax(1e-05,
    cumsum(seven.probs)))
seven.thresholds <- qt(seven.cumprobs,
    8)
plot(seq(from = -9, to = 9, length = 100),
    dt(seq(from = -9, to = 9, length = 100),
        df = 8), type = "l", xlab = "X",
    ylab = "density")
abline(v = seven.thresholds, col = 1:length(seven.thresholds))
```


0.4
0.3
0.2
0.1
0.0
density
−5
0
5
X

- We pulled out only the first row that now contains rating 7, as these accounts might be on a watch list.

Let's look at all of the ratings row by row using apply with the cusum function.

```
cum.probs <- t(apply(P.reverse, 1, function(v) {
    pmin(0.99999, pmax(1e-05, cumsum(v)))
}))
all.thresholds <- qt(cum.probs, 64)  #Use Student-t with 16 degrees of freedom
opa <- par(mfrow = c(2, 4))
for (j in 1:nrow(all.thresholds)) {
    plot(seq(from = -9, to = 9, length = 100),
        dt(seq(from = -9, to = 9, length = 100),
            64), type = "l", xlab = "Deviation",
        ylab = "Density", main = paste("Rating ",
            rownames(all.thresholds)[j]))
    abline(v = all.thresholds[j, ], col = 1:length(all.thresholds[j,
        ]))
}
par(opa)
```


Rating 7
Rating 6
Rating 5
Rating 4
Rating 3
Rating 2
Rating 1
Density
Deviation
0.0
0.1
0.2
0.3
0.4
−5
0
5

- `cum.probs` is adjusted for 0 and 1 as these might produce `NaN`s and stop the action. Notice the use of `pmin` and `pmax` to perform element by element (*p*arallel minimum and maximum) operations.
- This just goes to show it is hard to be rated a 6 or a 7. They are the riskiest of all. Watch list anyone?

Whitman

SCHOOL of MANAGEMENT

SYRACUSE UNIVERSITY

MBA@SYRACUSE

Now for the finale

- We will now use a technique that can be used with any risk category.
- The question on the table is: how can we generate a loss distribution for credit risk with so much variation across ratings?
- A loss distribution is composed of two elements, frequency and severity.
- Frequency asks the question how often and related to that question, how likely.
- Severity asks how big a loss.

- For operational risk frequency will be modeled by a Poisson distribution with an average arrival rate of any loss above a threshold.
- Severity will be modeled using Beta, Gamma, Exponential, Weibull, Normal, Log-Normal, Student-t, or extreme value distributions.
- For credit risk we can model some further components: loss given default (i.e., recovery) and exposure at default for severity, and probability of default for frequency.
- By the by: our transition probabilities are counting distirubtion and have as their basis the Poisson distirbution.
- Dwelling times AND the computation of transition probabilities are used to model the transition probabilities.

- Let's look at our top 20 firms and suppose we have these exposures.
- Probabilities of default are derived from the transition probabilities we just calculated.

```
# Specify portfolio characteristics
n.accounts <- 20  # Use even number please
exposure.accounts <- c(5, 5, 5, 5, 10,
    10, 10, 10, 20, 20, 20, 20, 30, 30,
    30, 30, 40, 40, 40, 40)
probability.accounts <- c(rep(P.hat[7,
    8], n.accounts/2), rep(P.hat[6, 8],
    n.accounts/2))
```

Enter Laplace

Pierre-Simon Laplace

A hugely useful tool for finance and risk is the Laplace transform. Let's formally define this as the integral (again think cumulative sum):

$$L(f(t)) = \int_0^\infty e^{-st} f(t) dt = f(s)$$

where $f(t)$ is a monotonic, piecewise differentiable function, say the cash flow from an asset, or a cumulative density function . To make this "real" for us we can calculate (or look up on a standard table of transforms)

$$L\{1\} = \int_0^\infty e^{-st} 1 dt = \frac{1}{s}$$

If 1 is a cash flow today $t = 0$, then $L\{1\}$ can be interpreted as the present value of \$1 at s rate of return in perpetuity. Laplace transforms are thus to a financial economist a present value calculation. They map the time series of cash flows, returns, exposures, into rates of return.

In our context we are trying to meld receivables account exposures, the rate of recovery if a default occurs, and the probability of default we worked so hard to calculate using the Markov chain probabilities.

For our purposes we need to calculate for m exposures the Laplace transform of the sum of losses convolved with probabilities of default:

$$\sum_{0}^{m} pd_i \, lgd_i \, S_i$$

where pd is the probability of default, lgd is the loss given default, and S is exposure in monetary units. In what follows `lgd` is the "loss given default," typically one minus the recovery rate in the event of default. Here we assume perfect recovery, even our attorney's fees.

This function effectively computes the cumulative loss given the probability of default, raw account exposures, and the loss given default. It does so without ever specifying the underlying distributions of the probabilities, loss given default, or exposures.

```
laplace.transform <- function(t, pd,
    exposure, lgd = rep(1, length(exposure))) {
    output <- rep(NA, length(t))
    for (i in 1:length(t)) output[i] <- exp(sum(log(1 -
        pd * (1 - exp(-exposure * lgd *
            t[i])))))
    output
}
```

It's technical...

- We can evaluate the Laplace transform at $s = i$ (where $i = sqrt{-1}$, the imaginary number) to produce the loss distribution's characteristic function.
- The loss distribution's characteristic function encapsulates all of the information about loss: means, standard deviations, skewness, kurtosis, quantiles,..., all of it.
- When we do that, we can then calculate the "fast"" Fourier transform (FFT) of the loss distribution characteristic function to recover the loss distribution itself.
- This is an often a more computationally efficient alternative to Monte Carlo simulation.
- Note below that in `R` we must divide the FFT output by the number of exposures (plus 1 to get the factor of 2 necessary for efficient operation of the FFT).

```r
N <- sum(exposure.accounts) + 1  # Exposure sum as a multiple of two
t <- 2 * pi * (0:(N - 1))/N  # Setting up a grid of t's
loss.transform <- laplace.transform(-t *
    (0+1i), probability.accounts, exposure.accounts)  # 1i is the imaginary number
loss.fft <- round(Re(fft(loss.transform)),
    digits = 20)  # Back to Real numbers
sum(loss.fft)
```

```
## [1] 421
```

```r
loss.probs <- loss.fft/N
loss.probs.1 <- loss.probs[(0:20) * 5 +
    1]
loss.q <- quantile(loss.probs.1, 0.99)
loss.es <- loss.probs.1[loss.probs.1 >
    loss.q]
(VaR <- loss.q * N)
```

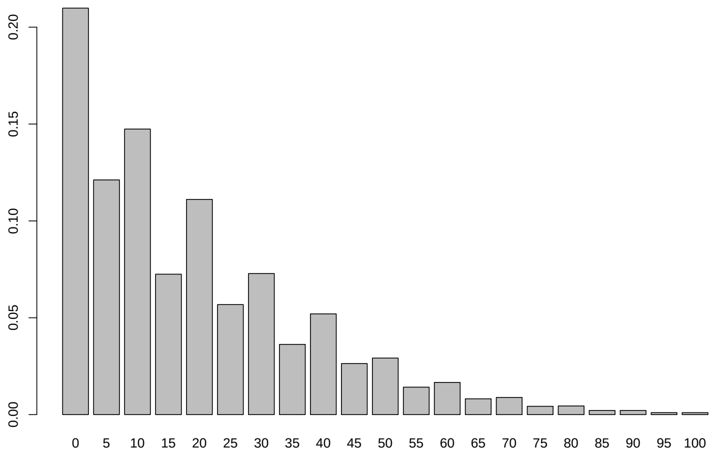
```
##      99%
## 83.08391
```

```r
(ES <- loss.es * N)
```

```
## [1] 88.34024
```

- We can use this same technique when we try to aggregate across all exposures in the organization.
- The last two statements using the `quantile` statement calculate the amount of capital we need to cover at least a 1% loss on this portfolio of accounts receivable.

```r
barplot(loss.probs.1, names.arg = paste((0:20) *
    5))
```



Whitman

SCHOOL *of* MANAGEMENT

SYRACUSE UNIVERSITY

MBA@SYRACUSE

The wrap

- We learned a lot of credit maths: Markov, transition probabilities, hazard rates, Laplace, and Fourier.
- We just built a rating model that produced data driven risk thresholds.
- We used these probabilities to generate an aggregate credit loss distribution.
- We also learned some math and some more R and definitely some more finance.

To Prepare for the live session:

1. What are the top 3 key learnings for you from this week?
2. What pieces are still a mystery?
3. What parts would you like more practice on?
4. Please review the assignment for this week. What questions would you like addressed in the live session?

Thanks!

Whitman
SCHOOL of MANAGEMENT
SYRACUSE UNIVERSITY

MBA@SYRACUSE