A Quantitative Primer on Investments with R

Dale W.R. Rosenthal[1]

30 May 2018

Q36

[1] info@q36llc.com

Introduction

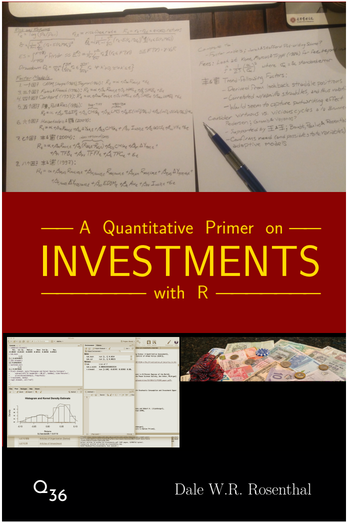
I did not come here just for free beer; I came here to sell you something.
Who am I?

- Head partner, Q_{36}: quant finance firm.
- PhD, Statistics, U. Chicago; BS, Elec. Eng., Cornell.
- Was: finance professor at UIC; now teach MSF class at Notre Dame.
- Presented @ ECB, FDIC, CFTC, BdF, BoFinland, NBSlovakia, RBNZ.
- 3 years: Proprietary algo trader, Equity Trading Lab, Morgan Stanley
- 5 years: Equity derivatives strategist, Long-Term Capital Management
- Interned at Goldman Sachs

I am here to sell you precious knowledge. Behold!

Q_{36}

Behold!



- A *truly* quantitative investments book
- A theory- *and* data-driven approach
- References latest research, issues
- *R* examples throughout the text

Q36

What Distinguishes the Book? Features

A number of features are included to help self-learning:

- Short quizzes and answers = quick feedback on comprehension.
- Many references w/DOIs and URLs = index to relevant research.
- *R* code throughout the book = see how to implement ideas.
- Slides at `www.q36llc.com/think/investments` = peek, refresher.
- `QPI` package (soon!) = wraps up useful code in book.
- Exercise solutions manual (soon!) = see how I analyzed the data.

What Distinguishes the Book? Content

A lot of content is less commonly covered:

- Modern markets;
- Efficiency + macroeconomy;
- Risk measures;
- Statistical modeling;
- Yield curves;
- Global investing and FX;
- Options in-depth;
- Credit;
- Structured finance + PE;
- Crises (LTCM*, Flash Crash, GFC, Euro Debt Crisis, China 2015).

Q_{36}

Cool Content

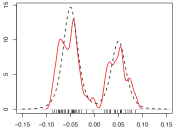
- In-text walk-throughs:
  - Risk measures: ES estimation horse race
  - Options: simulating mean-reverting prices; value of gold mine lease
  - CMO waterfall generation
- Exercises with data:
  - Has US economy become more allocatively efficient over time?
  - Compare inflation prediction methodologies
  - Infer PD, LGD from Greek, Venezuelan bond yields
  - Simulate a private equity fund cashflows
  - Finding closet index funds
- Prove a barter-based economy is likely doomed to fail.
- Yield curve movie
- What is happening with routine vs non-routine jobs?

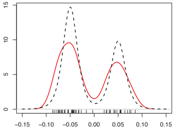
Risk Measure Horse Race

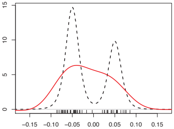
- Consider two distributions: two t's (bimodal) or t+gamma.

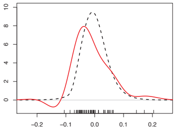
$$f_X(x) = \frac{3}{5} t_5(\mu = -0.05, \sigma = 0.02) + \frac{2}{5} t_5(\mu = 0.05, \sigma = 0.02), \quad (1)$$

$$f_Y(y) = \frac{1}{2} t_5(\mu = 0, \sigma = 0.05) + \frac{1}{2} \gamma_{k=5,\lambda=50}(y - 0.1). \quad (2)$$

- How do VaR and EX compare is computed different ways?
  - Empirically? Parametrically? (normal vs t?)
  - Kernel density estimator? 

or 

  - Cornish-Fisher? Edgeworth? 

or 

  - Extreme value theory? (GEV? GPD?)

Q_{36}

Risk Measure Horse Race: Walkthrough...

return to their prior levels. This is not hypothetical: Lefèvre (1923) details the manipulations and bear raids of the early 1900s; and, Goodwin (1987) discusses how Joseph Kennedy made his fortunes in such bear raids.

When somebody tries to exploit your weakness, you then leave the world of randomness and move into game theory. **Robust optimization** studies optimal behavior when under attack. Unfortunately, while robust optimization is an entire area of research, the findings tend to be consistent: any defense you mount will be very weak. Most results effectively say that if someone wants to hurt you, they probably can.

What this suggests is crucial. First, we should not be too reliant on extreme value theory or any other risk management methodology. Second, this implies there is a certain maximal trade size. If you grow beyond that size, you may be OK. However, in a crisis your death cannot be stopped; you will die exiting the trade, whether due to an inability to exit or due to price impact sealing your doom. Third, this is another reason to never publicize your trading strategy or positions.

8.7 From Density Measures to Risk Measures

With the above density estimates, we can try to estimate risk measures. To do this, we examine the two samples of data from above: one from a bimodal mixture of t-distributions and the other from a unimodal mixture of a t- and gamma distribution (shifted left to preserve mean 0):

$$f_X(x) = \frac{3}{5} t_5 \left(\frac{x+0.05}{0.02/\sqrt{5/3}} \right) \frac{\sqrt{5/3}}{0.02} + \frac{2}{5} t_5 \left(\frac{x-0.05}{0.02/\sqrt{5/3}} \right) \frac{\sqrt{5/3}}{0.02}, \tag{8.21}$$

$$f_Y(y) = \frac{1}{2} t_5 \left(\frac{y\sqrt{5/3}}{0.05} \right) \frac{\sqrt{5/3}}{0.05} + \frac{1}{2} \gamma_{k=5,\lambda=50}(y-0.1). \tag{8.22}$$

From these we sample 50 observations (as shown above in Section 8.5.3, see Figure 8.2). The resulting central moments are listed in Table 8.1. In the following subsections, we will see how to estimate risk measures (both good and bad) for these data. In the interest of brevity, I will only show the code for estimating the bimodal X risk measures.

8.7.1 Empirical

The empirical distribution estimates are easy: find the α-quantile and the average of returns below that. For the bimodal data:

```
alpha <- 0.05  # probability mass in the loss tail
# sort the data; grab the lowest alpha fraction of the data
sorted.x.obs <- sort(x.obs)
frac.idx <- length(sorted.x.obs)*alpha
left.idx <- floor(frac.idx)
```


Distribution	Mean	Var.	Skew.	Ex. Kurt.
Bimodal t mix (f_X)	-0.0120	0.0029	0.3270	-1.3576
Unimodal t,γ mix (f_Y)	-0.0001	0.0037	1.3816	2.6460

Table 8.1: Sample central moments for observed data from the bimodal t mixture and the unimodal t+gamma mixture distributions. Note that the bimodal distribution has a negative excess kurtosis (kurtosis <3) while the unimodal distribution has positive skewness and positive excess kurtosis.

```
# VaR is linearly interpolated between points
rfrac <- 1-(frac.idx-left.idx)
VaR.emp <- sorted.x.obs[left.idx:(left.idx+1)] %*% c(1-rfrac, rfrac)

# ES is just the average of the observed data in the tail
es.emp <- mean(sorted.x.obs[1:left.idx])
```

8.7.2 Parametric

For the normal distribution, we can find the VaR using the inverse standard normal cdf; and, with a little work, we can find the expected shortfall using that and the standard normal density:

$$\alpha\text{-}VaR = \mu + \sigma\Phi^{-1}(\alpha) \tag{8.23}$$

$$\alpha\text{-}ES = \mu - \sigma\frac{\phi(\Phi^{-1}(\alpha))}{\alpha}. \tag{8.24}$$

This is easily implemented in R:

```
VaR.norm <- mean.obs + sqrt(var.obs)*qnorm(alpha)
es.norm <- mean.obs - sqrt(var.obs)*dnorm(qnorm(alpha))/alpha
```

If we want to instead use a t-distribution, we need to estimate the degrees of freedom ν and then use the formula:

$$\alpha\text{-}ES = \mu + \sigma\frac{f(F^{-1}(\alpha))}{\alpha} \cdot \frac{\nu - 2 + [F^{-1}(\alpha)]^2}{\nu - 1}. \tag{8.25}$$

Estimating the degrees of freedom for a t-distribution by maximum likelihood can run into issues: the likelihood function may be unbounded. To get around that, we can use the moment condition that the excess kurtosis for a t_ν distribution is $\frac{6}{\nu-2}$. Unfortunately, for the bimodal data, the excess kurtosis is negative — so a t-distribution fit is infeasible. For the unimodal data, we get an estimated degrees of freedom of $\hat{\nu} = 4.48$. Computing the value-at-risk and expected shortfall is easy for the unimodal mixture since the gamma contribute no mass to the left tail.

Q36

Risk Measure Horse Race: . . . Walkthrough. . .

8.7.3 Kernel Density

For the kernel density estimate, we have many x values beyond those observed, so we use those to integrate numerically:

```
# create the kernel density estimate
kde <- density(x.obs, kernel="gaussian", bw="SJ")

# now create numerical CDF
int.kde <- cumsum(kde$y)/sum(kde$y)

# Find where alpha is in loss tail
left.idx <- findInterval(alpha, int.kde)

# linearly interpolate to find VaR
rfrac <- (alpha-int.kde[left.idx])/(int.kde[left.idx+1]-int.kde[left.idx])
VaR.kde <- kde$x[left.idx:(left.idx+1)] %*% c(1-rfrac, rfrac)

# numerically integrate kernel density estimate to get ES (a bit hacky)
es.kde <- kde$x[1:left.idx] %*% kde$y[1:left.idx]/sum(kde$y)/alpha
```

8.7.4 Edgeworth

For the Edgeworth expansion, we can use the properties of the Hermite polynomials to easily find the integral for the expansion previously given:

$$\int_{-\infty}^{q} \hat{f}_r(z)dz = \Phi(q) - \phi(q)\left(\gamma\frac{q^2+1}{6} + (\kappa-3)\frac{q^3-3q}{24} + \gamma^2\frac{q^5-10q^3+15q}{72}\right). \tag{8.26}$$

We can also use the recursion $\int z^{n+1}\phi(z)dz = -z^n\phi(z) + n\int z^{n-1}\phi(z)dz$ to find the expectation of the Edgeworth expansion:

$$\int_{-\infty}^{q} z\hat{f}_r(z) = -\phi(q)\left(1 + \gamma\frac{q^3}{6} + (\kappa-3)\frac{q^4-2q^2-1}{24} + \gamma^2\frac{q^6-9q^4+9q^2+3}{72}\right). \tag{8.27}$$

Then we can just find the point where the area "under" the left tail of the standardized expansion is α:

$$\alpha\text{-}VaR = q_\alpha = \inf\{q | \int_{-\infty}^{q} \hat{f}_r(z)dz = \alpha\}. \tag{8.28}$$

In *R*, we can do this (remembering to correct for the location and scale changes) with these functions and shifting the distribution down to find where it equals α:

```
# this function returns the "CDF" of the Edgeworth expansion
int.left.edge <- function(q, mu, sigma, skew=0, exkurt=0) {
  z <- (q-mu)/sigma
  pnorm(z) - dnorm(z)*(skew*(z^2+1)/6 + exkurt*(z^3-3*z)/24
```



```
                                 + skew^2*(z^5-10*z^3+15*z)/72)
}

# returns the "conditional expectation" of the Edgeworth expansion
exp.left.edge <- function(q, mu, sigma, skew=0, exkurt=0) {
  z <- (q-mu)/sigma
  mu - sigma/alpha*dnorm(z)*(1 + skew*(z^3)/6 + exkurt*(z^4-2*z^2-1)/24
                               + skew^2*(z^6-9*z^4+9*z^2+3)/72)
}

# Find the point where the area under the left tail equals alpha
# Do this by shifting the integral down by alpha to find where it
# crosses zero (i.e. where the integral equals alpha)
int.left.edge.shifted <- function(q) {
  int.left.edge(q, mean.obs, sqrt(var.obs), skew.obs, exkurt.obs) - alpha
}
find.VaR <- uniroot(int.left.edge.shifted, check.conv = TRUE,
                    lower=-1, upper=mean.obs)

# The VaR is that point; the ES is the integral up to that point
VaR.edge.exact <-find.VaR$root
es.edge.exact <- exp.left.edge(VaR.edge.exact, mean.obs, sqrt(var.obs),
                               skew.obs, exkurt.obs)
```

Of course, we could also approximate that numerically with the Edgeworth expansion we created:

```
# Alternately, just use numeric integration on the Edgeworth
# expansion points we got from the moments

# create numeric "CDF"
int.edge <- cumsum(edge.expand$approx)/sum(edge.expand$approx)

# linearly interpolate to find VaR
left.idx <- which(int.edge>alpha)[1] - 1
rfrac <- (alpha-int.edge[left.idx])/(int.edge[left.idx+1]-int.edge[left.idx])
VaR.edge.num <- edge.expand$y[left.idx:(left.idx+1)] %*% c(1-rfrac, rfrac)

# numerically integrate Edgeworth expansion to get ES
es.edge.num <- edge.expand$y[1:left.idx] %*% edge.expand$approx[1:left.idx]/
    sum(edge.expand$approx)/alpha
```

8.7.5 Modified Cornish-Fisher

If we use the modified Cornish-Fisher expansion in `PerformanceAnalytics`, finding the VaR and expected shortfall is easy:

```
VaR.cornfish <- VaR(x.obs, p=0.05, method="modified", mu=mean.obs, sigma=var.obs,
                    m3=skew.obs, m4=exkurt.obs)
es.cornfish <- ES(x.obs, p=0.05, method="modified", mu=mean.obs, sigma=var.obs,
                  m3=skew.obs, m4=exkurt.obs)
```

8.7.6 Extreme Value Theory

While extreme value theory is mostly used for estimating expected worst-case returns, we can also use it to estimate VaR and expected shortfall. This is easy with the `evir` *R* package.

Q_{36}

Risk Measure Horse Race: . . . Walkthrough


```
library(evir)

#evir package looks at upper tail, so flip returns
x.neg.obs <- -x.obs

# First, work with the generalized extreme value distribution
gev.model <- gev(x.neg.obs, block=23)
q.gev <- function(p) {
  -1*qgev(1-p, xi=gev.model$par.ests[["xi"]],
                mu=gev.model$par.ests[["mu"]],
                sigma=gev.model$par.ests[["sigma"]])
}
VaR.gev <- q.gev(alpha)  # use evir function
# a bit hacky: numerically integrate CDF to get ES
es.gev <- mean(q.gev(seq(0.001, alpha, 0.001)))

# Next, work with the generalized Pareto distribution
gpd.model <- gpd(x.neg.obs, nextremes=5)
riskmeasures(gpd.model, c(0.95))
```

For the bimodal data, the generalized extreme value distribution and generalized Pareto distribution calculate tail indices of $\hat{\xi}_{GEV} = -2.16$ and $\hat{\xi} = -1.26$. For the unimodal data, we get estimated tail indices of $\hat{\xi}_{GEV} = -0.04$ and $\hat{\xi} = -0.36$.

8.7.7 Comparison of Methods

We can now compare the various methods to the true VaR and expected shortfall. The true values are given by:

```
# now create the exact CDF
t.df <- 5
p.t <- function(q) {
  z1 <- (q+0.05)/desired.sd*t5.sd
  z2 <- (q-0.05)/desired.sd*t5.sd
  pt(z1, df=5)*3/5 + pt(z2, df=5)*2/5
}

# shift CDF down to find the point where alpha is in the loss tail
p.t.quantile <- uniroot(function(q) p.t(q)-alpha, check.conv=TRUE,
                        lower=-1, upper=mean.obs)
VaR.true <- p.t.quantile$root

# function finds ES for a t-distribution to left of quantile q
es.t <- function(q) {
  -dt(qt(q, df=t.df), df=t.df)/q*(t.df-2+qt(q, df=t.df)^2)/(t.df-1)
}

# ES calculation is crusty: we found point for alpha in loss tail.
# Now find ES for both t-distributions; then find average of them
q1 <- pt((VaR.true+0.05)/desired.sd*t5.sd, df=t.df)
q2 <- pt((VaR.true-0.05)/desired.sd*t5.sd, df=t.df)
es.true <- (3/5*es.t(q1) + 2/5*es.t(q2))*desired.sd/t5.sd - 0.01
```

Table 8.2 lists the various methodologies and their values for the 5%-VaR and 5%-expected shortfall. Obviously, for only 50 observations there will be a lot of noise in any estimates. Furthermore, estimating risk measures for a mixture of t-distributions is probably one of the more challenging tasks: while the tail behavior is fat-tailed, the kurtosis is low due to the bimodality. This is why many of the methods based

on unimodal densities (all but the kernel density estimator) overestimate the risk. With most financial returns being unimodal, these methods are far less likely to overestimate the risk.

	Bimodal t mix		Unimodal $t+\gamma$ mix	
Methodology	5%-VaR	5%-ES	5%-VaR	5%-ES
True	-7.51%	-8.83%	-5.72%	-8.72%
Empirical Distribution	-8.15%	-8.50%	-7.81%	-9.56%
Parametric – Normal	-10.14%	-12.41%	-10.10%	-12.64%
Parametric – t	—	—	-9.96%	-11.32%
Kernel Density Estimate	-8.59%	-9.35%	-8.61%	-10.03%
Edgeworth – Exact	-9.40%	-12.54%	-5.82%	-22.23%
Edgeworth – Numeric	-9.96%	-10.41%	-7.85%	-6.50%
Modified Cornish-Fisher	-9.72%	-10.46%	-7.24%	-11.18%
Generalized Extreme Value	-8.65%	-8.65%	-11.34%	-13.05%
Generalized Pareto	-8.11%	-8.41%	-7.52%	-8.95%

Table 8.2: Comparison of 5%-value-at-risk and 5%-expected shortfall as computed using various methods. The data are sampled from two mixture distributions: $[\frac{3}{5}t_5(x+0.05) + \frac{2}{5}t_5(x-0.05)] \cdot \frac{0.01}{\sqrt{5/3}}$ and $\frac{1}{2}t_5 + \frac{1}{2}(Gamma(5,50)(x-0.1))$. Given that these have low kurtosis but heavy tails, they are challenging distributions to fit. We can note that the empirical and extreme value theory-based Generalized Pareto methods worked well.

If we look at these calculations, a few salient details stand out. First, the empirical distribution does not do poorly for a small dataset. The methods that are based on unimodal densities (parametric normal, Edgeworth, Cornish-Fisher) can overstate the risk when handling a bimodal distribution. The Edgeworth-based expansions can also yield very strange results like the expected shortfall being less that the value-at-risk for the numerical integration. (Since the expected shortfall is the average of returns beyond value-at-risk, this should never happen.) This is caused by the negative tail behavior in the left tail for the unimodal data. Were we to integrate over a much larger range, that would go away; however, that makes calculating expectations a manual process. The Cornish-Fisher results are much more stable, part of why they feature in the `PerformanceAnalytics` *R* package.

Finally, the extreme value theory-based estimates also do very well. This is a bit surprising since there are multiple levels of approximation to get at the tail behavior. However, since those methods are designed to model tail behavior, the intermediate approximations seem justified.

It also bears repeating that value-at-risk is not a good risk measure. It is shown here to get a flavor for the variation in VaR estimates; however, this should not be taken as an endorsement of VaR. If that is not crystal clear, please re-read Section 8.1 to see how badly value-at-risk can fail.

Q36

Risk Measure Horse Race: Results

Methodology	Bimodal t mix 5%-VaR	Bimodal t mix 5%-ES	Unimodal $t+\gamma$ mix 5%-VaR	Unimodal $t+\gamma$ mix 5%-ES
True	-7.51%	-8.83%	-5.72%	-8.72%
Empirical Distribution	-8.15%	-8.50%	-7.81%	-9.56%
Parametric – Normal	-10.14%	-12.41%	-10.10%	-12.64%
Parametric – t	—	—	-9.96%	-11.32%
Kernel Density Estimate	-8.59%	-9.35%	-8.61%	-10.03%
Edgeworth – Exact	-9.40%	-12.54%	-5.82%	-22.23%
Edgeworth – Numeric	-9.96%	-10.41%	-7.85%	-6.50%
Modified Cornish-Fisher	-9.72%	-10.46%	-7.24%	-11.18%
Generalized Extreme Value	-8.65%	-8.65%	-11.34%	-13.05%
Generalized Pareto	-8.11%	-8.41%	-7.52%	-8.95%

Q_{36}

Valuing a Gold Mine

- Say we want to value a ten-year gold mine lease.
  - 10k oz/year; gold: \$1300/oz, 30% vol; extraction: \$1k/oz; WACC=8%, r_f=3%.
- Extraction decisions made annually? quarterly?
- What methodology to use?
  - Discount cashflows of expected prices from futures curve.
  - Simulate extraction options with GBM prices.
  - Simulate extraction options with mean-reverting (O-U) prices.
  - Price extraction options using Black-Scholes.
  - Price extraction options w/Black-Scholes mean-reverting approximation.

Q_{36}

Valuing a Gold Mine: Walkthrough. . .

gold. When we create this example, we have to be very careful about when decisions are made and when we get paid for extraction.

Assumptions

In this case, we make the extraction decision at the start of a period based on prices which we lock in by hedging. We then invest firm capital (incurring one period of cost-of-capital charges) and collect the revenue at the end of the period. Since we make decisions at the start of each period, the first-period decision (and value) will be the same for any method; however, subsequent period valuations will differ.

As gold prices rise, extraction comes from increasingly expensive mines. Assume the mine we lease can produce 10,000 oz per year and extraction costs \$1000/oz.[17] Also assume cash gold is \$1300/oz and futures out to one year are \$1300/oz before declining linearly to \$1200/oz at two years out and beyond. Gold volatility is 30%, our weighted average cost of capital is 8%, and the risk-free rate r_f is 3%. Finally, for simplicity assume the convenience yield is equal to storage costs.

We also need to consider what process gold prices follow. We could assume they follow geometric Brownian motion. However, this choice is questionable for many commodities since competition pushes prices toward mean extraction costs: when prices rise above mean extraction costs, more firms extract for a profit. This drives prices down until enough firms cease profitable extraction. Thus we could also assume gold prices follow an autoregressive (aka mean-reverting or Ornstein-Uhlenbeck) process. The only trick: as prices rise, the average extraction costs rise. We will assume the short-term equilibrium is the futures price.

Extraction Process

The extraction process has a few steps governing how valuation is done. Therefore, we walk-through the extraction decisions and some ideas about their valuation.

At time $t = 0$ we see futures for delivery in a year at \$1300/oz versus extraction costs of \$1000/oz. We lock that price in and run the mine incurring the weighted average cost of capital which is used to discount the \$300/oz profit. We assume no uncertainty in the amount of gold produced.[18] This value is the same for all methods.

The risk-neutral present-valued expected profit at time $t = 1$ is given by the one-year call option struck at \$1000/oz with underlier at \$1300/oz. That would lead us to lock in prices and incur extraction costs to realize the expected profit at $t = 2$, so we discount the option value back one period for the extraction over $t \in [1, 2)$ using the weighted average cost of capital. The DCF valuation assumes we

[17] Typical extraction costs are \$600–\$700/oz with more expensive mines in Asia and Africa.
[18] This is not accurate, but it would complicate this example.

will be able to lock in the \$200/oz profit and incurs two periods of discounting at the weighted average cost of capital because we have committed to extraction.[19]

The decision at $t = 2$ and afterward is more complicated. We have a futures curve that suggests the underlier will be at \$1200/oz. That would be our equilibrium price. So we price the decision with a one-year option over $t \in [1, 2)$ with the underlier assumed to be at \$1200/oz and the strike again at \$1000/oz. Is this realistic? We could enter into a forward contract on this option, so. . . yes. However, you can start to see how unclear the funding and valuation structures can be. We then run the mine, incur one period of capital costs, but we have a forward on the option — so we need to discount that at the risk-free rate. (We need not commit capital for the forward.) For the DCF valuation, we assume we will run the mine for a \$200/oz profit. We commit to that and incur three periods of WACC discounting.

The decision at $t = 3$ is still based off of the futures curve. Since curves get illiquid as we go out farther, we will price this as an option with the underlier at \$1200 and the strike at \$1000. If we assume a geometric Brownian motion price process, this would be a two-year option over $t \in [1, 3)$; if we assume an Ornstein-Uhlenbeck price process, a one-year option will yield a more accurate price than the two-year option. We use either of these to get the risk-neutral expected profit present-valued to time $t = 1$. Then we lock in prices, run the mine, and incur a period of capital costs. Again, we have a forward that requires a period of risk-free discounting for $t = 1$ to $t = 0$. The periods from here on are similar except with more risk-free discounting if we use the one-year-option approximation.

DCF Gold Mine Lease Valuation

If we price the lease with a DCF approach and make extraction decisions annually or quarterly, we get that the mine lease is worth:

$$V_{DCF,ann} = \frac{\$1300 - \$1000}{1.08} + \sum_{t=2}^{10} \frac{\$1200 - \$1000}{1.08^t} = \$14.3 \text{ mn or,} \tag{21.71}$$

$$\begin{aligned} V_{DCF,qtr} &= \sum_{t=1}^{4} \frac{\$1300 - \$1000}{1.08^{t/4}} + \sum_{t=5}^{8} \frac{\$1300 - 25(i-4) - \$1000}{1.08^{t/4}} \\ &+ \sum_{t=9}^{40} \frac{\$1200 - \$1000}{1.08^{t/4}} = \$15.1 \text{ mn.} \end{aligned} \tag{21.72}$$

Simulated Gold Mine Lease Valuation

We can easily simulate the gold price process for either a geometric Brownian motion or an Ornstein-Uhlenbeck process. The code to do this is straightforward:

[19] Remember that the options are not discounted at the WACC because they are priced by arbitrage with the underlier and a risk-free bond.

Q36

Valuing a Gold Mine: …Walkthrough…


```
oz.per.year <- 10000
rf <- 0.03  # 3% interest rates
wacc <- 0.08  # 8% weighted average cost of capital
num.sims <- 100000 # 100,000 simulations
s0 <- 1300  # gold price at t=0
K <- 1000  # cost of extraction
sigma <- 0.3  # volatility of gold log-returns
lambda <- 1  # rate of mean reversion; 1 => half-life = ln(2) = 0.69 years

## Annual extraction decisions at start of each year (t=0,...,9)
T.list <- 1:9  # 1--9 year options; years 2:8 are 1 year fwd
s.bar.list <- c(1300, rep(1200, 8))  # 1300/oz, then 1200/oz

# gold price follows geometric Brownian motion
sum.val <- (s0-K)/(1+wacc)  # time t=0 decision
for (i in 1:9) {
    T <- T.list[i]
    z <- rnorm(num.sims) # simulated Z_T values
    s.sim <- s0*exp((rf-sigma^2/2)*T+sigma*sqrt(T)*z)
    if (T>1) s.sim <- s.sim*s.bar.list[i]/s0
    opt.terminal.val.sim <- pmax(s.sim - K, 0)
    opt.value <- mean(opt.terminal.val.sim)*exp(-rf*T)
    sum.val <- sum.val + opt.value/(1+wacc)
}
print(sum.val*oz.per.year)

## gold price follows Ornstein-Uhlenbeck mean reversion
sum.val <- (s0-K)/(1+wacc)
for (i in 1:9) {
    T <- T.list[i]
    s.bar <- s.bar.list[i]
    s.meanrevert.sim <- s.bar + exp(-lambda*T)*(s0-s.bar) + sigma*s0*
        sqrt((1-exp(-2*lambda*T))/(2*lambda))*z
    opt.terminal.val.meanrevert.sim <- pmax(s.meanrevert.sim - K, 0)
    opt.meanrevert.value <- mean(opt.terminal.val.meanrevert.sim)*
        exp(-rf*T)
    sum.val <- sum.val + opt.meanrevert.value/(1+wacc)
}
print(sum.val*oz.per.year)

## Quarterly extract decisions at start of each quarter
T.list <- (1:39)/4 # 1--39 quarter options
s.bar.list <- c(rep(1300, 4), 1275, 1250, 1225, rep(1200, 32))

# gold price follows geometric Brownian motion
sum.val <- (s0-K)/(1+wacc)^0.25
for (i in 1:39) {
    T <- T.list[i]
    z <- rnorm(num.sims) # simulated Z_T values
    s.sim <- s0*exp(sigma*sqrt(T)*z)
    opt.terminal.val.sim <- pmax(s.sim - K, 0)
    opt.value <- mean(opt.terminal.val.sim)*exp(-rf*T)
    sum.val <- sum.val + opt.value/(1+wacc)^0.25
}
print(sum.val*oz.per.year/4)

# gold price follows Ornstein-Uhlenbeck mean reversion
sum.val <- (s0-K)/(1+wacc)^0.25
for (i in 1:39) {
    T <- T.list[i]
    s.bar <- s.bar.list[i]
    s.meanrevert.sim <- s.bar + exp(-lambda*T)*(s0-s.bar) + sigma*s0*
        sqrt((1-exp(-2*lambda*T))/(2*lambda))*z
    opt.terminal.val.meanrevert.sim <- pmax(s.meanrevert.sim - K, 0)
```



```
    opt.meanrevert.value <- mean(opt.terminal.val.meanrevert.sim)*exp(-rf*T)
    sum.val <- sum.val + opt.meanrevert.value/(1+wacc)^0.25
}
print(sum.val*oz.per.year/4)
```

If we run these simulations, we get some crazy prices for the geometric Brownian motion processes: \$41.7 mn for annual extraction decisions and \$60.5 mn for quarterly extraction decisions. The Ornstein-Uhlenbeck processes yield more reasonable and believable valuations: \$20.8 mn for annual extraction decisions and \$21.9 mn for quarterly extraction decisions.

Real Options Approach to Gold Mine Lease Valuation

If we price the mine lease with options, we again consider both the geometric Brownian motion-implied price and the approximation for a mean-reverting process. The code to do this is also easy:

```
bs.call <- function(S,K,rf,T,sigma) {
    d1 <- (log(S/K)+(rf+sigma^2/2)*T)/(sigma*sqrt(T))
    S*pnorm(d1)-K*exp(-rf*T)*pnorm(d1-sigma*sqrt(T))
}

oz.per.year <- 10000
rf <- 0.03  # 3% interest rates
wacc <- 0.08  # 8% weighted average cost of capital
s0 <- 1300
s0.lt <- 1200
K <- 1000
sigma <- 0.3

## annual real options approach, no mean reversion
oz.per.year*((s0-K)/(1+wacc) + bs.call(s0,K,rf,1,sigma)/(1+wacc) +
            sum(bs.call(s0.lt,K,rf,1:8,sigma)/(1+wacc)*exp(-rf)))

## annual real options, assuming mean reversion (approximation)
oz.per.year*((s0-K)/(1+wacc) + bs.call(s0,K,rf,1,sigma)/(1+wacc) +
            sum(bs.call(1200,1000,0.03,1,0.3)/(1+wacc)*exp(-rf*(1:8))))

## quarterly real options, no mean reversion
oz.per.year/4*((s0-K)/(1+wacc)^0.25
    + sum(bs.call(1300,1000,0.03,(1:4)/4,0.3)/(1+wacc)^0.25)
    + sum(bs.call(1300-25*(1:3),1000,0.03,0.25,0.3)/(1+wacc)^0.25*exp(-rf*(5:7)/4))
    + sum(bs.call(1200,1000,0.03,(1:32)/4,0.3)/(1+wacc)^0.25*exp(-rf*(8:39)/4)))

## quarterly real options, assuming mean reversion (approximation)
oz.per.year/4*((s0-K)/(1+wacc)^0.25
    + sum(bs.call(1300,1000,0.03,0.25,0.3)/(1+wacc)^0.25*exp(-rf*(0:3)/4))
    + sum(bs.call(1300-25*(1:4),1000,0.03,0.25,0.3)/(1+wacc)^0.25*exp(-rf*(4:7)/4))
    + sum(bs.call(1200,1000,0.03,0.25,0.3)/(1+wacc)^0.25*exp(-rf*(8:39)/4)))
```

For the real option approaches discussed, we (again) get crazy prices for the standard options formulas — although less so since some of the options are shorter-term or start from two years out. For the standard geometric Brownian motion real options approach, we get a value of \$37.7 mn for annual extraction decisions and \$33.3 mn for quarterly extraction decisions. The mean-reverting approximation real options approach again yields more reasonable and believable valuations: \$23.9 mn for annual extraction decisions and \$20.3 mn for quarterly extraction decisions.

Q_{36}

Valuing a Gold Mine: ...Walkthrough

Summary

Comparing these, we can get a better impression of how much we believe each and the uncertainty of valuation. Table 21.1 shows all of the different valuation results.

Methodology	Annual	Quarterly
DCF	\$14.3	\$15.1
Simulated, GBM	\$41.7	\$60.5
Simulated, OU	\$20.8	\$21.9
Real Options, GBM	\$37.7	\$33.3
Real Options, OU Approx.	\$23.9	\$20.3

Table 21.1: Comparison of valuations in millions from various methodologies for valuing a ten-year gold mine lease. The simulated Ornstein-Uhlenbeck and real options approach Ornstein-Uhlenbeck approximation results are the closer to what we believe given prior assumptions than the geometric Brownian motion and DCF approaches.

Given that the simulated Ornstein-Uhlenbeck and the real options approach with the Ornstein-Uhlenbeck approximation results are the closest to what we believe for the gold price process, these results are the most believable. Given that we would likely make quarterly extraction decisions instead of annually, this suggests the mine lease is worth about \$22 mn. Note that this is almost 50% greater than the DCF valuation.

If we had production constraints or other complicated conditions on the lease, we would need to use simulation, a tree, or stochastic optimization. Also note that if gold were below \$1000/oz at any point on the futures curve, the DCF valuation would be even more flawed.

21.8 Summary

For those interested in learning more about options, I highly recommend McDonald (2010) and Luenberger (2013). For information on commodity options, Geman (2005) is superb. Also, Derman and Miller (2016) is a good text covering volatility surfaces and recent modeling advances. To dig into the probability theory behind pricing contingent claims, Etheridge (2002) and Steele (2001) are delightful introductions.

For those wanting to be a full-blown $\mathbb{Q}$-quant, Wichura (2006) (if you can find a copy) is *by far* the best text on measure theory available.[20] If that is not available, Øksendahl (2013), Billingsley (1996) (the 3rd, not later editions), and Karatzas and Shreve (1991) are OK substitutes.

[20] Wichura's suggestions to Billingsley eliminated an entire chapter of unneeded development in the second edition of Billingsley (1996).

$\mathbb{Q}_{36}$

Valuing a Gold Mine: Results

Methodology	Annual	Quarterly
DCF	\$14.3	\$15.1
Simulated, GBM	\$41.7	\$60.5
Simulated, OU	\$20.8	\$21.9
Real Options, GBM	\$37.7	\$33.3
Real Options, OU Approx.	\$23.9	\$20.3

Valuations in USD millions

Q_{36}

Collateralized Mortgage Obligation Waterfall

- Say we have a \$30 mn CMO; tranches: 78% A, 8% B, 9% C, 5% Z.
  - A gets \$24 mn of principal, B and C get \$3 mn of principal.
- We would like to see cashflows under various prepayment assumptions
- Typically, benchmark with PSA prepayments speeds.
  - PSA 100: prepays rise to 6% at 30 months, 6% thereafter.
  - PSA 200: Prepays at twice the PSA 100 speed.

Q_{36}

CMO Waterfall: Walkthrough...

rates fall, they tend to decline after a while as the pool becomes increasingly full of borrowers who will not or cannot prepay.

23.3 Securitization

While the above are effectively forms of securitization, we typically reserve the term **securitization** for the reallocation of principal, interest, prepayments, and losses among investors in SPV-issued securities (and not covered bonds). This is done by the SPV issuing different securities against the debt in its pool.

23.3.1 Terminology

When an SPV creates a number of different securities from mortgages, we often refer to it as a **collateralized mortgage obligation** (CMO). The birth and flowering of this market was largely driven by Salomon Brothers and is discussed in the amusing classic *Liar's Poker*.

A CMO might do something simple like separate the interest and principal to offer investors interest-only and principal-only instruments. However, a CMO might also offer some investors more protection from prepayments or defaults by shifting that risk to other investors willing to bear it for the potential of high returns.

The main technologies a CMO uses for reducing and shifting risk are diversification, **tranching** — a way of slicing up the portfolio and reapportioning risks and cashflows to different slices, and subordination of tranches.

We define a tranche by an **attachment point** (the fraction of debt subordinate to that tranche) and **detachment point** (which defines the subordination for the tranche and its subordinates. The difference between the detachment and attachment points also defines the "thickness" or fraction of notional impairment that the tranche can absorb. Together, the tranche definitions define the CMO's capital structure.

According to Chaudhary (2006), CMOs have five common structure archetypes:

1. **pass-through**: send shares of payments out to all investors;
2. **sequential**: tranches are paid principal sequentially;
3. **amortization classes**: tranches receive principal according to assumed prepayment rates;
4. **IO/PO**: tranches receive interest- or principal-only; and,
5. **indexed**: tranches receive interest based on an index, often a floating rate.

Obviously, these can be combined. In many of these cases, creating one type of desired tranche requires creating another **support tranche** which may have the opposite risk or which absorbs risk to protect the desired tranche. The division of these tranches is largely dictated by market demand. For example, there is a lot of demand for short- to intermediate-term debt yielding more than government bonds at low risk, less demand for longer-term debt at risk of prepayment, and little

demand for risky debt which may pay off handsomely or receive little interest at all if interest rates drop over the life of the pool. An example is illuminating.

23.3.2 Example CMO

Suppose we have a sequential CMO funded with the previously-mentioned $30 mn pool of 30-year fixed-rate US mortgages — and that they are conforming mortgages securitized by a government-sponsored entity. Since they are backed by that entity and, implicitly, by the US government, we (for this example) ignore default risk.

We create a set of tranches A, B, C, and Z with seniority in that order: A is senior to B is senior to C is senior to Z: $A > B > C > Z$. The tranche structure is given in Table 23.1. Typically, the thinner tranches between A and Z are referred to as **mezzanine** tranches; and, the senior tranches sacrifice interest to fund the Z tranche which provides protection against prepayments.

Tranche	Attach	Detach	Principal Owed	Claims Interest On
A	22%	100%	$24.0 mn	$23.4 mn
B	14%	22%	$3.0 mn	$2.4 mn
C	5%	14%	$3.0 mn	$2.7 mn
Z	0%	5%	$0.0 mn	$1.5 mn

Table 23.1: Tranche structure for example sequential CMO. The senior tranches sacrifice some interest to fund the Z tranche which protects senior tranches from prepayments.

Typically, tranche names mirror bond ratings (as above); however, the names should not be conflated with ratings.[10] Furthermore, the naming of the Z tranche (aka "toxic waste") is puzzling. Some say it is because it does not get paid until near the end of its life making it like a zero-coupon bond; others say the Z tranche is named because the Z tranche absorbs the first prepayments (giving it the highest prepayment risk) and Z is as far from A as possible (connoting the worst rating).

The payment structure is often defined by the **waterfall**: the ordering and structure which dictates how tranches participate in principal and interest payments. A diagram showing the intuition for how tranches work and the cashflow waterfall is shown in Figure 23.5.

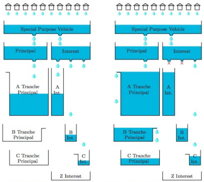
Figures 23.6 and 23.7 show the CMO at maturity with no prepayments and prepayment of some mortgages. Note that prepayments do not affect the cashflows received by senior tranches in this case; however, they do impose a loss on the Z tranche — since it loses out on curtailed interest payments. Prepayments normally hasten the repayment of senior tranches and, if they are severe enough, can affect the mezzanine or even A tranche.

There is another subtlety only hinted at in these diagrams: the Z tranche foregoes

[10] Credit rating documents even note that a "AAA" or "A" tranche rating is not equivalent to a "AAA" or "A" bond rating. This raises the question: why use the same rating names if the ratings should not be conflated?

Q36

Valuing a Gold Mine: . . . Walkthrough. . .



Before A tranche repaid After A and B tranches repaid

Figure 23.5: Cashflow "waterfall" as sequential CMO tranches are repaid in order.

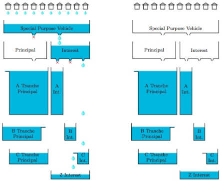
interest until the senior tranches are repaid — but it then gets all of the remaining interest (if any). In effect, the Z tranche swaps its earlier interest payments for the later interest and principal payments of the senior tranches. Not only that, but those early interest payments are used to repay principal on the senior tranches.

While these colorful plots are intuitive, they are not how most structurers and mortgage analysts look at CMOs. When they discuss the waterfall, it is often defined in terms of a computer language that specifies the reapportionment and prioritization of cashflows for each tranche. Also typical is to show a **waterfall plot**: a plot of cashflows expected across time for each tranche under some assumption of prepayment rates. The plots use stacked areas versus time (as in Figure 23.4) to show when various tranches are paid off.

The code for allocating payments to tranches is a bit involved. Below is the *R* code for our example sequential CMO. The waterfall plot based on this code is shown in Figure 23.8. While waterfall plots typically show interest plotted on top, that is more confusing and so I have done otherwise here.

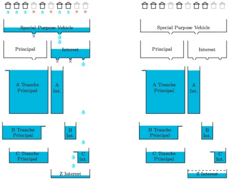
```
principal.start <- 30e6  # pool principal
n.peryr <- 12  # monthly = 12 payments per year
T <- 30   # 30-year mortgage
r <- 0.0625  # 6.25%
t <- 1:(n.peryr*T)

## tranche info
attach.pts <- list(a=0.22, b=0.14, c=0.05, z=0)
```

After all tranche principal repaid If no prepayments, all interest paid

Figure 23.6: Cashflow "waterfall" for sequential CMO ending with no prepayments.



After all tranche principal repaid With prepayments, Z tranche takes loss

Figure 23.7: Cashflow "waterfall" for sequential CMO ending with some prepayments.

$\mathbb{Q}_{36}$

CMO Waterfall: . . . Walkthrough


```
principal <- list(a=24e6,b=3e6,c=3e6,z=0)
interestbase <- list(a=(1-attach.pts$a)*principal.start,
                     b=(attach.pts$a-attach.pts$b)*principal.start,
                     c=(attach.pts$b-attach.pts$c)*principal.start,
                     z=(attach.pts$c-attach.pts$z)*principal.start)

## calculate level monthly payment
pmt.level <- principal.start*r/(n.peryr*(1-1/(1+r/n.peryr)^(n.peryr*T)))

## 100% PSA prepayment: rate up to 6% annual at 30 months
psa.speed <- 100  # 100% PSA
max.rate <- psa.speed/100*0.06
seasoned.month <- 30  # at 30 months, PSA says mortgages are seasoned
prepay.rate <- c(seq(max.rate/seasoned.month, max.rate, max.rate/seasoned.month),
                 rep(max.rate, n.peryr*T-seasoned.month))
prepay.monthly.survival <- (1-prepay.rate)^(1/n.peryr)
prepay.monthly.rate <- 1-prepay.monthly.survival

## In production code, this would be more elegant; however,
## a for loop shows the intuition better than other approaches
### Use fullword.variables for pool payments...
num.pmts <- n.peryr*T
principal.left <- c(principal.start, rep(0,num.pmts-1))
pmt.perperiod <- c(pmt.level, rep(0,num.pmts-1))
pmt.interest <- rep(0, num.pmts)
pmt.principal <- rep(0, num.pmts)
pmt.prepays <- rep(0, num.pmts)
### ... and use abbr.vars (abbreviated) for tranche payments
pmt.prin.a <- pmt.prin.b <- pmt.prin.c <- rep(0, num.pmts)
pmt.int.a <- pmt.int.b <- pmt.int.c <- pmt.int.z <- rep(0, num.pmts)
stop.loop <- FALSE
for (i in 1:num.pmts) {
    ## determine pool cashflows
    pmt.interest[i] <- principal.left[i]*r/n.peryr
    pmt.principal[i] <- pmt.perperiod[i] - pmt.interest[i]
    if (pmt.principal[i] > principal.left[i]) {
        ## do not repay more than principal remaining
        pmt.principal[i] <- principal.left[i]
        ## stop the loop; pool is paid off!
        stop.loop <- TRUE
    } else {
        ## do not prepay more than principal remaining after normal payment
        pmt.prepays[i] <- (principal.left[i]-pmt.principal[i])*prepay.monthly.rate[i]
        if ((pmt.principal[i] + pmt.prepays[i]) > principal.left[i]) {
            pmt.prepays[i] <- principal.left[i] - pmt.principal[i]
            ## stop the loop; pool is paid off!
            stop.loop <- TRUE
        }
    }
    ## allocate payments to tranches
    prin.to.allocate <- pmt.principal[i]+pmt.prepays[i]
    prinleft.a <- principal$a - sum(pmt.prin.a)
    prinleft.b <- principal$b - sum(pmt.prin.b)
    prinleft.c <- principal$c - sum(pmt.prin.c)
    ## Scale interest payments by fraction not repaid*interestbase
    ## Remaining interest will get allocated to principal repayment
    ## and then deferrred interest on the z tranche
    pmt.int.a[i] <- prinleft.a/principal$a*interestbase$a*r/n.peryr
    pmt.int.b[i] <- prinleft.b/principal$b*interestbase$b*r/n.peryr
    pmt.int.c[i] <- prinleft.c/principal$c*interestbase$c*r/n.peryr
    prin.to.allocate <- prin.to.allocate + pmt.interest[i] -
        pmt.int.a[i] - pmt.int.b[i] - pmt.int.c[i]
    ## A is first in line for principal (Andropov is the name to say)
    if (prinleft.a > 0) {
        pmt.prin.a[i] <- min(prinleft.a, prin.to.allocate)
```



```
        prin.to.allocate <- prin.to.allocate - pmt.prin.a[i]
    }
    ## The letter B is next in line (for principal, Brezhnev's dead so he'll do just fine)
    if (prinleft.b > 0 & prin.to.allocate > 0) {
        pmt.prin.b[i] <- min(prinleft.b, prin.to.allocate)
        prin.to.allocate <- prin.to.allocate - pmt.prin.b[i]
    }
    ## The letter C is next for principal (C is for Chernenko)
    if (prinleft.c > 0 & prin.to.allocate > 0) {
        pmt.prin.c[i] <- min(prinleft.c, prin.to.allocate)
        prin.to.allocate <- prin.to.allocate - pmt.prin.c[i]
    }
    ## Finally, if any cashflows are left, they go to the Z tranche
    if (prin.to.allocate > 0)
        pmt.int.z[i] <- prin.to.allocate

    ## prepare for next iteration of loop
    if (i < num.pmts) {
        principal.left[i+1] <- principal.left[i] - pmt.principal[i] - pmt.prepays[i]
        pmt.perperiod[i+1] <- pmt.perperiod[i]*prepay.monthly.survival[i]
    }

    if (stop.loop)
        break
}

areaplot(t,data.frame(pmt.int.c,pmt.int.b,pmt.int.a,
                      pmt.prin.a,pmt.prin.b,pmt.prin.c,pmt.int.z),
         col=c("green","yellow","lightblue","blue","orange","darkgreen","red"),
         ylab="", border=FALSE)
text(100, 150000, "A", cex=2, col="white")
text(210, 60000, "B", cex=2)
text(255, 40000, "C", cex=2, col="white")
text(310, 20000, "Z", cex=2)
```

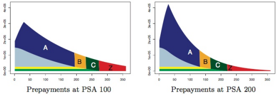
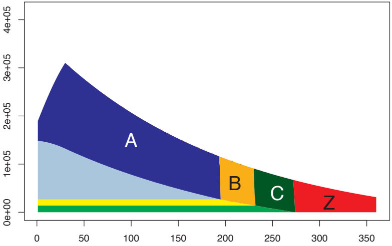
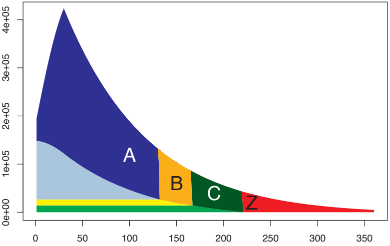


Figure 23.8: Waterfall plot for the example sequential CMO created from \$30 mn of 30-year (360-month) fixed-rate mortgages with prepayments at PSA 100 (left) and PSA 200 (right). Tranche A is shown in blue colors; tranche B in orange colors; and tranche C in green colors. Dark colors are principal; light colors are interest. The Z tranche (all interest) is shown in red and is the last tranche to be repaid. Prepayments clearly shift the maturities of the senior tranches and diminish the value of the Z tranche.

We can see from looking at the waterfall plots that for prepayments at PSA 100, the A tranche is paid off after about 16 years; the B tranche is paid off a little more than 3 years after the A tranche; and, the C tranche is paid off about 4 years after

CMO Waterfall: Results



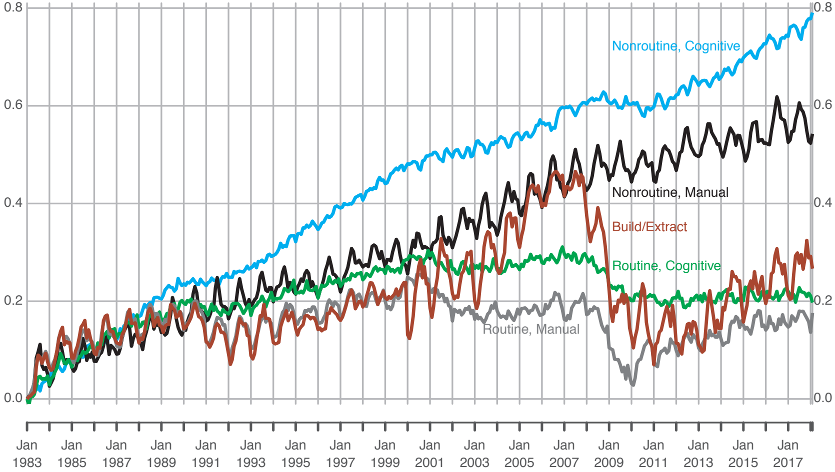


- Can see interest (lighter colors) and principal payments.
- Note how prepayments affect Z tranche (no principal, interest-only).

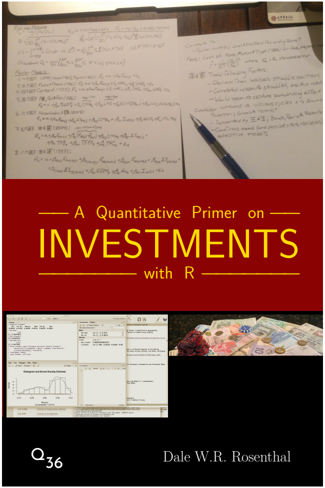
Q_{36}

Why We Need to Improve Finance Education

- Many people learn finance as a collection of facts, a few formulas.
- Theory is crucial, explains how and why the world changes.
- Implementation is also key, but must automate to justify salary.



Q_{36}



- Buy the book! Amazon, B&N, etc.
- Come to the book launch! Email firm (info@q36llc.com)
- Check out the slides: `www.q36llc.com/think/investments`

Q36