Progress Reports on ???

Wang Jiahe
February 29, 2024

Contents

1 Overview

This reports covers following aspects:

- Summarize of the theory, model, estimation methods and data used in AER16.
- Summarize of the theory, null hypothesis and test-statistics proposed by QE and how can it be applied to the validations of VaR and CoVaR models.
- Summarize of ???.
- Data descriptions and my empirical work. ???.

2 Summary of Notations and definitions

Table 1: Notations

Symbol	Definition
AER16	:
X^i	Institution i's loss, calculated as $-\frac{\Delta N^i_{t+1}}{N^i_t}$
N^i_t	institution i's net worth at time t
α^i_q	intercept of the quantile regression for estimating $VaR^i_{q,t}$
γ^i_q	coefficient of $\mathbf{M}_{t-1}$ in the quantile regression for estimating $VaR^i_{q,t}$
$\epsilon^i_{q,t}$	residuals in the quantile regression for estimating $VaR^i_{q,t}$
$\alpha^{system\|i}_q$	intercept of the quantile regression for estimating $CoVaR^{system\|i}_{q,t}$
$\gamma^{system\|i}_q$	coefficient of $\mathbf{M}_{t-1}$ in the quantile regression for estimating $CoVaR^{system\|i}_{q,t}$
$\beta^{system\|i}_q$	coefficient of X^i in the quantile regression for estimating $CoVaR^{system\|i}_{q,t}$
$\epsilon^{system\|i}_{q,t}$	residuals in the quantile regression for estimating $CoVaR^{system\|i}_{q,t}$
$\mathbf{M}_{t-1}$	vector of lagged state variables, 7 variables are used in AER16
VaR^i_q	Institution i's value at risk at confidence level q%
$VaR^i_{q,t}$	Adding t in the subscript means VaR is not stationary
$CoVaR^{i\|j}_q$	VaR of i conditional on some event of j
$CoVaR^{i\|j}_{q,t}$	Non-stationary $CoVaR^{i\|j}_q$
$\Delta^{\$}CoVaR$	$\Delta CoVaR \times Size^i$, $Size^{\$}$ is the size of i in dollar
$Stress - \Delta CoVaR$	$CoVaR^{system\|i}_q$, estimated using worst (1-q)% of state variable realizations.
$Network - \Delta CoVaR$	$CoVaR^{i\|j}_q$, i, j are both individual institutions.
$Exposure - \Delta CoVaR$	$CoVaR^{i\|system}_q$, i is an individual institution.
$Forward - \Delta CoVaR$	$\Delta^{Fwd}_h CoVaR^i_{q,t} = \hat{\alpha} + \hat{\mathbf{c}}\mathbf{M}_{t-h} + \hat{\mathbf{b}}\mathbf{X}^i_{t-h}$
QE22	:
$\mathbb{X}_t$	$\mathcal{F}_t$-adapted specific finite-dimensional conditioning random process.
$\mathcal{X}$	support or sample space of $\mathbb{X}_t$.
Y_t	loss, same meaning as X_t in AER16.
$\mathcal{F}_t$	filtration defined on the sample space, the information available at time t.
f_t	conditional q-quantile of Y_t given $\mathcal{F}_t$
θ	the parameter set in the VaR model f_t.
$\hat{\theta}_n$	empirical estimation of θ with n samples.
C_t	state variables, QE22's notation for M_t in AER16
$Q_q(Y_{t+1}\|\mathcal{F}_t)$	true one-period-ahead VaR at confidence level q in a general sense.
$Q_q(Y^{market}_{t+1}\|C_t, Y^{firm}_{t+1})$	QE22's notation of equation (7), one specific model of $Q_q(Y_{t+1}\|\mathcal{F}_t)$.
$\mathbb{E}[1_{\{Y_{t+1}\leq f_t\}} - q\|\mathcal{F}_t]$	conditional moment
$\hat{T}_n$	sup-t, test statistics used in testing if the conditional moment equals to 0.

3 $\Delta CoVaR$, What it is and How to estimates it.

3.1-3.4 introduce the $\Delta CoVaR$ definition, classifications, properties and estimation method used in AER16, 3.5 introduces a nonparametrical test proposed by QE22 and its usage in VaR and CoVaR.

3.1 Definition

In reference to AER16, instituion i's $\Delta CoVaR_q^{system|i}$ is defined as the change in the value at risk of the financial system conditional on institution i being under distress relative to its median state. It is a difference between two CoVaRs and Mathematically speaking, $CoVaR$ (or $CoVar_q^{system|C(x^i)}$) is defined as :

$$Pr(X^{system} \leq CoVar_q^{system|C(x^i)}|C(x^i)) = q\% \tag{1}$$

where X^{system} represents the loss of the financial system(calculated as $-\frac{\Delta N_{t+1}^i}{N_t^i}$ in AER16, N_t^i is institution i's net worth at time t), $C(X^i)$ represents some event of institution i(median state and distress state). Compared with financial system's VaR_q^{system}:

$$Pr(X^{system} \leq Var_q^{system}) = q\% \tag{2}$$

CoVaR is just a conditional VaR. For $\Delta CoVaR_q^{system|i}$, it's defined as:

$$\Delta CoVaR_q^{system|i} = CoVar_q^{system|X^i=VaR_q^i} - CoVar_q^{system|X^i=VaR_{50}^i} \tag{3}$$

Here the 50 is the "median state" mentioned in AER16, while q, the quantile level, is the "distress state".

3.2 Classifications

The definition of $\Delta CoVaR_q^{system|i}$ shows that it is a directional measure, by which means $\Delta CoVaR_q^{system|i}$ is not necessary equal to $\Delta CoVaR_q^{i|system}$. The conditioning radically changes the interpretation of the systemic risk measure. This report considers the direction of $\Delta CoVaR_q^{system|i}$, which quantifies the incremental change in systemic risk when institution i is in distress relative to its median state. And for other types of $\Delta CoVaR$, below gives their definitions:

1. *Exposure*-$\Delta CoVaR$: $\Delta CoVaR_q^{i|system}$, the reverse condition of $\Delta CoVaR_q^{system|i}$, which reveals the institution's risks when exposed to financial crisis.

2. *Network*-$\Delta CoVaR$: $\Delta CoVaR_q^{i|j}$, which refers to the tail-risk dependency between two individual institutions.

3. *Stress*-$\Delta CoVaR$: $\Delta CoVaR_{q,t}^{system|i}$, estimated by substituting worst $(1-q)\%$ of state variable realizations into the fitted model of $\Delta CoVaR_{q,t}^{system|i}$.

3.3 Properties

$\Delta CoVaR$ has following properties:

1. Independent of the business model of the conditional institution. Since $\Delta CoVaR$ is using VaR as condition instead of a particular return level, the probability of the conditional institution i's loss exceeding VaR_q^i is $(1-q)\%$ and is always $(1-q)\%$ when changing the conditional institution.

2. Measures the tail-dependency between two random return variables.

3. A reduced-form measure. $\Delta CoVaR$ is a statistical tail-dependency measure and does not necessarily correctly capture externalities or spillover effects. And it is not deduced from a structural model.

4. Clone Property. after splitting one large individually systemic institution into n smaller clones, the CoVaR of the large institution (in return space) is exactly the same as the CoVaRs of the n clones.

5. Systemic As a Herd. Consider a large number of small financial institutions that are exposed to the same factors (because they hold similar positions and are funded in a similar way). Only one of these institutions falling into distress will not necessarily cause a systemic crisis. However, if the distress is due to a common factor, then the other institutions will also be in distress. Overall, the set of institutions is systemic as a herd. Each individual institution's corisk measure should capture this notion of being systemic as a herd, even in the absence of a direct causal link. The $\Delta CoVaR$ measure achieves exactly that.

6. Endogeneity. each institution's $\Delta CoVaR$ is endogenous and depends on other institutions' risk-taking.

3.4 Estimations

AER16 used quantile regression to estimate $\Delta CoVaR$ and there are more methods to be applied like GARCH model. Before all of that, the first step is the estimation of VaR and CoVaR.

Considering there are two entities (the financial system and the conditional institution), we need first estimate the conditional institution i's VaR. If the data generating process of institution i's returns (or losses) is stationary, we can simply use the $q\%$ quantile of the historical returns as the VaR. But stationarity does not hold. For time-varying VaR, AER16 estimate VaRs and CoVaRs as a function of state variables, allowing us to model the evolution of the joint distributions over time. By using $CoVaR^i_{q,t}$ and $VaR^i_{q,t}$ we mean VaR and CoVaR are time-varying and is estimated using qunatile regression with a vector of lagged state variables $\mathbf{M}_{t-1}$ as follows:

1. First we run a quantile regression of the conditional institution i's loss X^i_t at quantile level q with the vector of lagged state variables $\mathbf{M}_{t-1}$:

$$X^i_t = \alpha^i_q + \gamma^i_q \mathbf{M}_{t-1} + \epsilon^i_{q,t} \tag{4}$$

2. According to the definition of VaR (how much a set of investments might lose with a given probability), the estimated coefficients $\hat{\alpha}^i_q$, $\hat{\gamma}^i_q$ can be directly used to estimate $VaR^i_{q,t}$:

$$VaR^i_{q,t} = \hat{\alpha}^i_q + \hat{\gamma}^i_q \mathbf{M}_{t-1} \tag{5}$$

3. Recall that CoVaR is just a conditional VaR, all we need to estimate CoVaR is just adding X^i_t into above equations:

$$X^{system|i}_t = \alpha^{system|i}_q + \gamma^{system|i}_q \mathbf{M}_{t-1} + \beta^{system|i}_q X^i_t + \epsilon^{system|i}_{q,t} \tag{6}$$

$$CoVaR^{system|i}_{q,t} = \hat{\alpha}^{system|i}_q + \hat{\gamma}^{system|i}_q \mathbf{M}_{t-1} + \hat{\beta}^{system|i}_q VaR^i_{q,t} \tag{7}$$

4. Then calculate the differences between $CoVaR^i_{q,t}$ and $CoVaR^i_{50,t}$ to get $\Delta CoVaR^{system|i}_{q,t}$:

$$\Delta CoVaR^{system|i}_{q,t} = CoVaR^i_{q,t} - CoVaR^i_{50,t} = \hat{\beta}^{system|i}_q (VaR^i_{q,t} - VaR^i_{50,t}) \tag{8}$$

Following contents will use $\Delta CoVaR^i_{q,t}$ as shortcut for $\Delta CoVaR^{system|i}_{q,t}$ since we only considers systematic risks conditioned on individual institutions. For state variables in $\mathbf{M}_{t-1}$, AER16 used 7 state variables that are:

1. known to capture time variation in the conditional moments of asset returns.
2. liquid.
3. tractable.

AER16 emphasized that these variables should not be interpreted as systematic risk factors, but rather as variables that condition the mean and volatility of the risk measures. 7 state variables used in AER16 are:

1. **Yld3M** : The change in the three-month yield from the Federal Reserve Board's H.15 release.
2. **TERM** : The change in the slope of the yield curve, measured by the spread between the composite long-term bond yield and the three-month bill rate obtained from the Federal Reserve Board's H.15 release.
3. **TED** : A short-term TED spread, defined as the difference between the three-month LIBOR rate and the three-month secondary market Treasury bill rate.
4. **Credit** : The change in the credit spread between Moody's Baa-rated bonds and the ten-year Treasury rate from the Federal Reserve Board's H.15 release.
5. **MktRet** : The weekly market return computed from the S&P500.
6. **Housing** : The weekly real estate sector return in excess of the market financial sector return (from the real estate companies with SIC code 65–66)
7. **MktSD** : Equity volatility, which is computed as the 22-day rolling standard deviation of the daily CRSP equity market return.

Above variables can also be named as macrostate variables compared with a news set of institution-specific variables used in the constructing of forward-$\Delta^{\$}CoVaR$.

Forward-$\Delta^{\$}CoVaR$ tries to predict estimated $\Delta CoVaR$ using lagged institution specific factors ($\mathbf{X}^i_{t-h}$) and state variables ($\mathbf{M}_{t-h}$) assuming following equation exists:

$$\Delta^{\$}CoVaR^i_{q,t} = \alpha + \mathbf{c}\mathbf{M}_{t-h} + \mathbf{b}\mathbf{X}^i_{t-h} + \eta^i_t \tag{9}$$

Where $\Delta^{\$}CoVaR = \Delta CoVaR \times Size^i$, (Size is the dollar size of institution i), $\mathbf{X}^i_{t-h}$ is the vector of characteristics for institution i, $\mathbf{M}_{t-h}$ is the vector of macrostate variables lagged hquarters, and η^i_t is an error term. The h-quarter ahead prediction $\Delta^{\$}CoVaR^i_{q,t}$ is denoted as:

$$\Delta^{Fwd}_h CoVaR^i_{q,t} = \hat{\alpha} + \hat{\mathbf{c}}\mathbf{M}_{t-h} + \hat{\mathbf{b}}\mathbf{X}^i_{t-h} \tag{10}$$

3.5 Model Validation Test

AER16 used a linear quantile model to estimate VaR and CoVaR as shown in equation (5) and (7), and directly reported t-statistic results without reporting model validation tests, which leaves a question that whether the model structure is correct. QE22 proposed a nonparametrical test by estimating the "conditional moment function" via series regression and test whether it is identically zero using uniform functional inference. To explain what "conditional moment function" mean in QE22 and how to conduct the test, we first need to introduce the notations used in QE22:

- Y_t, loss, same meaning as X_t in AER16.
- $\mathcal{F}_t$, filtration defined on the sample space, the information available at time t.
- f_t, the assumed conditional q-quantile of Y_t given $\mathcal{F}_t$, also a general representation of the parametric models used for estimating CoVaR.

- θ, the parameter set in the VaR model f_t.
- $\hat{\theta}_n$, empirical estimation of θ with n samples.
- C_t, state variables, QE22's notation for M_t in AER16
- $Q_q(Y_{t+1}|\mathcal{F}_t)$, true one-period-ahead VaR at confidence level q in a general sense.
- $Q_q(Y_{t+1}^{market}|C_t, Y_{t+1}^{firm})$, QE22's notation of equation (7), one specific model of $Q_q(Y_{t+1}|\mathcal{F}_t)$.
- $\mathbb{E}[1_{\{Y_{t+1}\leq f_t\}} - q|\mathcal{F}_t]$, conditional moment
- X_t, $\mathcal{F}_t$-adapted specific finite-dimensional conditioning random process, since AER used X_t, $\mathbb{X}_t$ will be used instead.
- $\mathcal{X}$, support or sample space of $\mathbb{X}_t$.

From the definition of CoVaR, for a specific model f_t to be correct, the conditional moment must equal to 0:

$$\mathbb{E}[1_{\{Y_{t+1}\leq f_t\}} - q|\mathcal{F}_t] = 0 \tag{11}$$

So a consistent specification test can be obtained by estimating the conditional expectation function on the left-hand side of equation (9) and testing whether it is identically zero. Li and Liao (2020) proposed a uniform nonparametric inference method based on series regression for general time-series data. They estimated the conditional moment function by regressing $1_{\{Y_{t+1}\leq f_t\}} - q$ on an asymptotically growing number of approximating functions of $\mathbb{X}_t$. However, in a realistically calibrated Monte Carlo experiment (see Section 3 in QE22), such an asymptotic approximation works well only for quantiles near the middle of the distribution, while it suffers from substantial size distortion for quantiles in the tails. QE22 overcome this issue by proposing a novel bootstrap method for computing the critical values for test statistic.

Following QE22, for a VaR model to be correct, one must satisfies the implied conditional restrictions:

$$\mathbb{E}[1_{\{Y_{t+1}\leq f_t(\theta^\star)\}} - q|\mathcal{F}_t] = 0 \tag{12}$$

where the $\theta^\star$ is the probability limit of $\hat{\theta}_n$ ($\hat{\theta}_n \xrightarrow{P} \theta^*$). Above equation implies :

$$\mathbb{E}[1_{\{Y_{t+1}\leq f_t(\theta^\star)\}} - q|\mathbb{X}_t^\star = x] = 0, \text{for all } x \in \mathcal{X} \tag{13}$$

And that's the null hypothesis. The test statistics is "sup-t":

$$\hat{T}_n = sup_{x\in\mathcal{X}} \frac{n^{1/2}}{\hat{\sigma}_n(x)} \tag{14}$$

where $\hat{\sigma}_n(x)$ is standard error function and defined as $\hat{\sigma}_n(x) = (P(\hat{x})^T\hat{\Sigma}_n P(\hat{x}))^{1/2}$, $P(\hat{x})$ is a series of approximation functions : $P(x) = (p_1(x), ..., p_{m_n}(x))^T$ (mth-order Legendre polynomials).

QE22 used bootstrap procedure to compute critical values and the detailed mathematics behind can be found in QE22 part 2.3. The test will be used in empirical part.

4 ???, improvement ???

This part is deleted.

5 Data

AER' data includes:

1. All publicly traded US financial institutions(commercial banks, broker-dealers, insurance companies, and real estate companies), collected from WRDS CRSP; Stock / Security Files / Daily Stock File with COMPUSTAT SIC codes between 60 and 67 for the period from January 1971 to June 2013. Then merge daily data into weekly.

2. Macro variables data. Collected from Federal Reserve Board's H.15, British Bankers' Association, British Bankers' Association.(FR Y-9C, H.15, Treasury, FF)

3. Institution specific variables for estimating $forward - \Delta CoVaR$. Collected from CRSP/-Compustat Merged; Fundamentals Quarterly

QE needs extra condition variables and that paper chosed two uncertainty measures:

- Economic policy uncertainty index proposed by Baker, Bloom, and Davis (2016). Monthly, collected from https://www.policyuncertainty.com

- financial and macro uncertainty indexes proposed by Jurado, Ludvigson, and Ng (2015). Monthly, collected from https://www.sydneyludvigson.com/macro-and-financial-uncertainty-indexes

While this reports collected the same data as AER16 and extends the data range to today, which leads to a total of 23GB of files. After some data cleaning and preprocessing as what AER16 did:

1. start with daily equity data from CRSP for all financial institutions with two-digit COMPUSTAT SIC codes between 60 and 67 inclusive, indexed by PERMNO. Banks correspond to SIC codes 60, 61, and 6712; insurance companies correspond to SIC codes 63–64, real estate companies correspond to SIC codes 65–6, and broker-dealers are SIC code 67 (except for the bank holding companies, 6712).

2. keep only ordinary common shares (which exclude certificates, e.g., American Depositary Receipts (ADRs), Shares of Beneficial Interest (SBIs), Real Estate Investment Trusts (REITs), etc.) and drop daily equity observations with missing or negative prices or missing returns.

3. Daily data are then collapsed to weekly frequency and merged with quarterly balance sheet data from the CRSP/COMPUSTAT quarterly dataset. The quarterly data are filtered to remove leverage and book-to-market ratios less than zero and greater than 100. Apply 1 percent and 99 percent truncation to the maturity mismatch variable

The final data set is then around 2 GB. The rest is deleted in this part........

6 Empirical results and ???

This part is deleted ....

7 Codes for reproduction

7.1 Stata codes for preprocessing

```
// CoVaR Estimation -- Main File -- covar_main.do

// This file is the main do-file for CoVaR estimation.
// to perform different stages of the process

// ******************* SET GLOBAL VARIABLES *********************
//   This section needs to be run EVERY TIME any part
//   of the covar code is run
// ***************************************************************

clear
set more off
version 11

adopath + "~/ado/plus"

global meDate = "12311925"
global firstDate = "01011971"
global newestDate = "06302013"
global asofDate = "06302013"
global osDate = "12312006"
global beginCrisis = "07012007"
global endCrisis = "01012009"

//SET quarter OF LAST GOOD QTRLY COMPUSTAT DATA
global realQMax = tq(2013q2)
global minYears = 5

//PATHS
global rootPath = "~/CoVaR"
global scratchPath = "~/CoVaR/Scratch"
global bhcPath =  "~/CoVaR/bhc"
global svData = "~/CoVaR/Data/statevars/"

//INPUT FILES--see Documentation
global bhcFile = "/CoVaR/bhc/Output/bhcFile"
global crspFile = "/CoVaR/Data/wrds_crsp_25_2013_alt"
global sicFile = "/CoVaR/Data/sic_codes_sept_2013"
global delistFile = "/CoVaR/Data/delist"
global compFile = "/CoVaR/Data/crspcomp612013q3big"

// The rootpath should be on a backed up directory
// from one run of the program to another.

global levMin = 0                          //minimum leverage
global levMax = 100                        //maximum leverage
global bmMin = 0                           //minimum book to market
global bmMax = 100                         //maximum book to market
global quantList = "95 99"                 //list of quantiles over which to estimate CoVaR
global garchIterations = 50                //maximum number of iterations
global archIterations = 100                //maximum number of iterations
global neweyLags = 5                       //newey-west lags
global horizonList = "8 4 1"               //forecast horizons (in quarters)

//Initialize Programs
do "$rootPath/DO Files/covar_qest.do"
```

```
do "$rootPath/DO Files/covar_gen_wkly_panel.do"
do "$rootPath/DO Files/covar_sim1.do"
do "$rootPath/DO Files/covar_sim2.do"
do "$rootPath/DO Files/covar_qest_sim.do"

// ****************************************************************
//      USE OUTPUT PERMNO LISTS FROM THIS SECTION TO PULL
//          COMP FILES (by hand) FROM WRDS INTERFACE
// ****************************************************************

// The method is to get all firms w/ COMPUSTAT SIC codes in 6000-7000
// First we pull the permnos from CRSP/COMPUSTAT that match this constraint...

do "$rootPath/DO Files/covar_getCOMPPermnoList.do" "$newestDate"
do "$rootPath/DO Files/covar_getCOMPPermnoList.do" "$osDate"

//  In principal there should be different files for each of the 2
//  dates, but for the dates in question, the permno lists have
//  historically been the same; this is why there has only been
//  the need for 1 COMP file for all dates

// ****************************************************************
//                   Pull CRSP File by hand
// ****************************************************************

// use COMPPermnoList (see Documentation)

// ****************************************************************
//      USE OUTPUT PERMNO LISTS FROM THIS SECTION TO PULL
//          CRSP FILES (by hand) FROM WRDS INTERFACE
// ****************************************************************

// ...Then pull everything in this list from CRSP
// Then pull the balance sheet data from CRSP/COMPUSTAT, then do data cleaning

do "$rootPath/DO Files/covar_getCRSPPermnoList.do" "$newestDate"
do "$rootPath/DO Files/covar_getCRSPPermnoList.do" "$osDate"

//  In principal there should be different files for each of the 2
//  dates, but for the dates in question, the permno lists have
//  historically been the same; this is why there has only been
//  the need for 1 CRSP file for all dates
// ****************************************************************
//                   Pull CRSP File by hand
// ****************************************************************

// use CRSPPermnoList (see Documentation)

// ****************************************************************
//                   DATA COLLECTION & ASSEMBLY
// ****************************************************************

// Get and clean Daily CRSP Data
do "$rootPath/DO Files/covar_cleanCRSP.do" "$newestDate"
//keep only data to this date variable

// Get and clean Quarterly COMPUSTAT Data (from CRSP/COMPUSTAT merged)
do "$rootPath/DO Files/covar_cleanCOMP.do" "$newestDate"
//keep only data up to this date variable

// Run merger adjustment and preestimation to create datasets on which to estimate CoVaR.
```

```
// We want to create some out-of-sample merger adjusted datasets.
// On these datasets, we will allow the data to go past the last valid merger adjustment
// and see if the out of sample forecasts generate good future predictions

//Run data assembly for full in-sample analysis
do "$rootPath/DO Files/covar_mergers.do" "$newestDate" "$newestDate"
do "$rootPath/DO Files/covar_preest.do" "$newestDate" "$newestDate" "$firstDate" "all_firms" "ret"

do "$rootPath/DO Files/covar_mergers.do" "$osDate" "$newestDate"
do "$rootPath/DO Files/covar_preest.do" "$osDate" "$newestDate" "$firstDate" "all_firms" "ret"

//Assemble State Variables
do "$rootPath/DO Files/covar_statevars.do" "$osDate"
do "$rootPath/DO Files/covar_statevars.do" "$newestDate"

// ***************************************************************
//                      CoVaR ESTIMATION
// ***************************************************************

// Estimate CoVaR - Quantile Regressions

foreach quant in $quantList{
     covar_qest "$scratchPath/all_firms_m${newestDate}_d${newestDate}" "$rootPath/Intermed/conditionf
     sort permno week
     save "$scratchPath/wklyCoVaR_p`quant'_m${newestDate}_d${newestDate}", replace
}

genWklyPanel, in_name("wklyCoVaR") out_name("CoVaR_Weekly") mergeDate(${newestDate}) endDate(${newest

clear
set more off
foreach quant in $quantList{
     covar_qest "$scratchPath/all_firms_m${osDate}_d${newestDate}" "$rootPath/Intermed/conditionfacto
     sort permno week
     save "$scratchPath/wklyCoVaR_p`quant'_m${osDate}_d${osDate}", replace
}

genWklyPanel, in_name("wklyCoVaR") out_name("CoVaR_Weekly") mergeDate(${osDate}) endDate(${osDate}) c

foreach quant in $quantList{
     covar_qest "$scratchPath/all_firms_m${newestDate}_d${newestDate}" "$rootPath/Intermed/conditionf
     sort permno week
     save "$scratchPath/wklyCoVaR_altSys_p`quant'_m${newestDate}_d${newestDate}", replace
}

genWklyPanel, in_name("wklyCoVaR_altSys") out_name("CoVaR_Weekly_altSys") mergeDate(${newestDate}) en


// *******************************************************************
//                      TSTATS and PVALUES
// *******************************************************************

do "$rootPath/DO Files/covar_qest_tstats.do"
foreach quant in $quantList{
     covar_qest_tstats "$scratchPath/all_firms_m${newestDate}_d${newestDate}" "$rootPath/Intermed/con
}


// *******************************************************************
//                      ASSEMBLE QUARTERLY DATA
// *******************************************************************
```

```
do "$rootPath/DO Files/covar_gen_mkt_betas.do"
do "$rootPath/DO Files/covar_gen_vol.do"

do "$rootPath/DO Files/covar_gen_fcast_vars.do" "$osDate" "$newestDate"
do "$rootPath/DO Files/covar_gen_fcast_vars.do" "$newestDate" "$newestDate"

//Assemble quarterly panel up to out-of-sample date
do "$rootPath/DO Files/covar_gen_qtrly_panel.do" "${osDate}" "${newestDate}" "CoVaR_Weekly_m${osDate

//Assemble quarterly panel for total sample period
do "$rootPath/DO Files/covar_gen_qtrly_panel.do" "${newestDate}" "${newestDate}" "CoVaR_Weekly_m${new

//Assemble quarterly panel for total sample period (alternate system returns)
do "$rootPath/DO Files/covar_gen_qtrly_panel.do" "${newestDate}" "${newestDate}" "CoVaR_Weekly_altSys

do "$rootPath/DO Files/covar_additional_variables.do" "CoVaR_Qtrly_m${osDate}_d${osDate}" "Final_CoVa
do "$rootPath/DO Files/covar_additional_variables.do" "CoVaR_Qtrly_m${newestDate}_d${newestDate}" "Fi
do "$rootPath/DO Files/covar_additional_variables.do" "CoVaR_Qtrly_altSys_m${newestDate}_d${newestDat

// *******************************************************************
//                      Dataset to Distribute
// *******************************************************************

use "$rootPath/Output/Final_CoVaR_Qtrly_m${newestDate}_d${newestDate}", clear
keep permno tic conm qtr cCoVaR_p95 cCoVaR_p99 cVaR_p95 cVaR_p99
order permno tic conm qtr
drop if cVaR_p95 ==.

tsset permno qtr
tsfill

capture mkdir "$rootPath/Output/for_Distribution"

save "$rootPath/Output/for_Distribution/CoVaR_qtrly", replace

// *********************************************************************
//                           GARCH COVAR
// *********************************************************************

do "$rootPath/DO Files/covar_garchest.do" "$newestDate" "$newestDate"

do "$rootPath/DO Files/covar_gen_qtrly_panel.do" "$newestDate" "$newestDate" "garchPanel_m${newestDat

do "$rootPath/DO Files/covar_additional_variables.do" "garch_CoVaR_Qtrly_m${newestDate}_d${newestDate

// *******************************************************************
//                       Panel Regregressions
// *******************************************************************

//Tables 5,6,8, and 9
do "$rootPath/DO Files/covar_panelreg.do"

// *******************************************************************
//      Cristofferson Test of Conditional Coverage for VaR
// *******************************************************************

//Table 10
do "$rootPath/DO Files/covar_cristofferson_test.do"

// *******************************************************************
```

```
//                    Simulation (part 1)
// ************************************************************************

// Run part 1 of the simulation; then run the matlab program "CoVaR_sim.m"
// using the number of simulations output by this section; then run part 2.

// Simulation using Wells Fargo--the most important financial firm with a long history.
// The idea is to use system returns and firm returns to estimate the parameters in Appendix A and th
// randomly generated independent errors (Normal, Laplace, Student) to simulate 1000 alternate "syste
// Then using Wells Fargo returns and these new system returns, estimate 1000 CoVaR series with which
// an empirical distribution of CoVaR for every time period. Since we know the data generating proces
// the "true" CoVaR to compare the simulated CoVaRs against

set more off

use "$scratchPath/all_firms_m${newestDate}_d${newestDate}", clear

//use equity returns for the simulation
local retD = "reteq"

gen temp_ret = `retD'
gen temp_ret0 = `retD'0
gen temp_sysret = sys`retD'_i

drop ret* sysret*
rename temp_ret ret
rename temp_ret0 ret0
rename temp_sysret sysret_i

keep if permno == 38703 //Wells Fargo
save "${scratchPath}/wf0", replace

//estimate parameters of data generating process in appendix A
covar_sim1 "${scratchPath}/wf0" "$rootPath/Intermed/conditionfactors_stress_d${newestDate}" "${firstD

save "${scratchPath}/wf1", replace

local N = _N
di "RUN MATLAB SIMULATION WITH THIS MANY ERRORS: `N'000"

// ************************************************************************
//    Matlab Interlude -- Run this between simulation part 1 and 2
//                (if it hasn't been done already)
// ************************************************************************

//                rootPath/Simulation/CoVaR_sim.m

// ************************************************************************
//                    Simulation (part 2)
// ************************************************************************

clear
set more off

//stack 1000 instances of the Wells Fargo Data
forvalues x = 1/1000{
        append using "${scratchPath}/wf1"
        replace ID = `x' if _n > `N'*(`x'-1)
}
gen sim_id = _n
sort sim_id
```

```
save "${scratchPath}/wf2", replace

//read in and append matlab generated errors (arbitrarily partitioned into 100 files)
forvalues i = 1/100{
        insheet using ${rootPath}/Data/simulation/CoVaR_sim_`i'.csv, clear
        sort sim_id
        save ${scratchPath}/CoVaR_sim_`i', replace
}

clear

forvalues i = 1/100{
        append using ${scratchPath}/CoVaR_sim_`i'
        sort sim_id
}
save ${scratchPath}/CoVaR_sim, replace

use "${scratchPath}/wf2", replace

merge sim_id using ${scratchPath}/CoVaR_sim
drop _merge*
sort ID week
save "${scratchPath}/wf3", replace

//simulate 1000 "system return" variables
covar_sim2 "${scratchPath}/wf3"
save "${scratchPath}/wf4", replace

//And now that the system returns have been simulated, estimate CoVaR
foreach error in "norm" "lap" "stud" {

     local quant = "95"

     covar_qest_sim "${scratchPath}/wf4_`error'" "$rootPath/Intermed/conditionfactors_stress_d${newes

     sort permno week
     keep week ID cCoVaR_p95 *95_ret B_ret cVaR_p95 cmed
     rename cCoVaR_p95 simCoVaR_`error'
     sort ID week
     capture drop _merge*

     save "${scratchPath}/wf4_`error'_`quant'", replace
}

// ************************************************************************
//                             Figures
// ************************************************************************

do "$rootPath/DO Files/figures_1.do"

do "$rootPath/DO Files/figures_2.do"

do "$rootPath/DO Files/figures_3.do"

do "$rootPath/DO Files/figures_4.do"

do "$rootPath/DO Files/figures_6.do"

do "$rootPath/DO Files/figures_sim.do"
```

```
//This one takes a while...
do "$rootPath/DO Files/figure5_table7_VaR.do"

/***********************************************************************
                            SUMMARY TABLES
***********************************************************************/

do "$rootPath/DO Files/tables_1_4.do"

/***********************************************************************
                              END OF CODE
***********************************************************************/
```

7.2 Python codes for preprocessing

Generating full training set

```
#========================================================
file_path = 'D:/RA/WRDS_Data/DailyStock.csv'

chunks = pd.read_csv(file_path, usecols=['date', 'SICCD', 'PERMNO', 'TICKER', 'SHRCLS
filtered_chunks = []

for chunk in chunks:
    filtered_chunks.append(chunk)

filtered_data = pd.concat(filtered_chunks)

filtered_data.to_csv('D:/RA/WRDS_Data/filtered_data.csv', index=False)

data = data.dropna(subset=['SICCD'])
data = data[data['SICCD'] != 'Z']
data['SICCD'] = data['SICCD'].astype(int)

data = data[(data['SICCD'] >= 6000) & (data['SICCD'] <= 6799)]

data = data[['date', 'TICKER', 'RET']]
data = data.pivot_table(index='date', columns='TICKER', values='RET', aggfunc='first'
```

Sample 100 institutions for testing

```
sample = data['1977':'2013'].dropna(axis=1, how='all')
sample = data['1977':'2013'].dropna(axis=1, how='all')
nan_percentage = sample.isna().mean()

columns_to_drop = nan_percentage[nan_percentage > 0.5].index

sample.drop(columns=columns_to_drop, inplace=True)

sample = sample.iloc[:, :100].replace('C', np.nan).replace('B', np.nan).astype(float)
```

7.3 R codes for ??? estimation

This part is deleted ....

7.4 MATLAB codes for model validation test

Implementation of Test

```
function results = Specification_Test_Bootstrap(Moment, CondVar, m, bootstrp_n)

% INPUTS:
%        Moment: The moment condition
%        CondVar: Conditioning variables
%        m: Order of the polynomial
%        bootstrp_n: Number of bootstrap samples


    n = length(Moment);

%% Create P(X)
    PX = NaN(n, m);
    [CondVar, Index] = sort(CondVar);
    Moment = Moment(Index);
    [~,CondVar_rank] = ismember(CondVar,unique(CondVar));
    CondVar_scaled = 2*(CondVar_rank - min(CondVar_rank))/(max(CondVar_rank) - min(Co
    for ind_k = 1:m
            PX(:, ind_k) = legendreP(ind_k-1, CondVar_scaled);
    end

%% Calculate the supT stat
    betahat = (PX'*PX)\(PX'*Moment);

    u = Moment - PX*betahat;

    unPX = u.*PX;
    unPX_demean = unPX - mean(unPX, 1);
    A_n = unPX_demean'*unPX_demean/n;
    Q_n = 1/n*(PX'*PX);

    Sig_n =  Q_n\A_n/Q_n;

    That = (-1)*inf;
    That_vector = NaN(n, 1);
    for ind_n = 1:n
       PX_tmp = PX(ind_n, :);
       sigma_n = sqrt(PX_tmp * Sig_n *PX_tmp');
       T_tmp = abs(sqrt(n)*PX_tmp*betahat/sigma_n);
       That_vector(ind_n) = T_tmp;
       if That < T_tmp
          That = T_tmp;
       end
    end


%% Compute the Critical Value using Bootstrap
    That_save = NaN(bootstrp_n, 1);
```

```
    parfor ind_bootstr = 1:bootstrp_n

        rng(ind_bootstr*n)

        IDs = randsample(n,n, true);
        PX_bstr = PX(IDs, :);
        Moment_bstr = Moment(IDs, :);

        betahat_bstr = (PX_bstr'*PX_bstr)\(PX_bstr'*Moment_bstr);

        u = Moment_bstr - PX_bstr*betahat_bstr;

        unPX = u.*PX_bstr;
        unPX_demean = unPX - mean(unPX, 1);
        A_n = unPX_demean'*unPX_demean/n;
        Q_n = 1/n*(PX_bstr'*PX_bstr);

        Sig_n =  Q_n\A_n/Q_n;

        That_save(ind_bootstr) = (-1)*inf;
        for ind_n = 1:n
            PX_tmp = PX_bstr(ind_n, :);
            sigma_n = sqrt(PX_tmp * Sig_n *PX_tmp');
            T_tmp = abs(sqrt(n)*PX_tmp*(betahat_bstr-betahat)/sigma_n);
            if That_save(ind_bootstr) < T_tmp
                That_save(ind_bootstr) = T_tmp;
            end
        end

    end
    cv_save = prctile(That_save, 95);

    results.cv = cv_save;
    results.That = That;
    results.That_save = That_save;
    if That > cv_save
        results.decision = 0;
    else
        results.decision = 1;
    end

end

```

Testing CoVaR

```

clear; clc; close all;

load RiskMeasures
load CoVaR_credit
load EPU_Historical
```

```matlab

return_loss = double(return_loss);

Observed = ~isnan(CoVaR95);

Ret = return_loss(2:end, 2:end); % First column is the market loss
Ret_Sys = return_loss(2:end, 1);

CondVar = Historical_EPU(2:end, 2);

m_vector = 1:10;
bootstrp_n = 1000;

results_CoVaR95 = NaN(length(m_vector), size(CoVaR95, 2));
results_CoVaR99 = NaN(length(m_vector), size(CoVaR95, 2));
for ind_firm = 1:size(CoVaR95, 2)
    for m = m_vector

            Observed_tmp = Observed(:, ind_firm);

            Moment_CoVaR95 = (Ret_Sys(Observed_tmp)  <= CoVaR95(Observed_tmp, ind_firm
            Moment_CoVaR99 = (Ret_Sys(Observed_tmp)  <= CoVaR99(Observed_tmp, ind_firm

            CondVar_tmp = CondVar(Observed_tmp);

            results = Specification_Test_Bootstrap(Moment_CoVaR95, CondVar_tmp, m, boo
            results_CoVaR95(m, ind_firm) = results.decision;

            results = Specification_Test_Bootstrap(Moment_CoVaR99, CondVar_tmp, m, boo
            results_CoVaR99(m, ind_firm) = results.decision;

    end

end
clearvars -except results_*

save OUTPUT_CoVaR_credit

```