MathWorks
FINANCE CONFERENCE 2023

CRISK: Quantifying the Expected Capital Shortfall in a Climate Stress Scenario

October 11-12 | Online

Michael Robbins,
Columbia University

Arpit Narain,
MathWorks

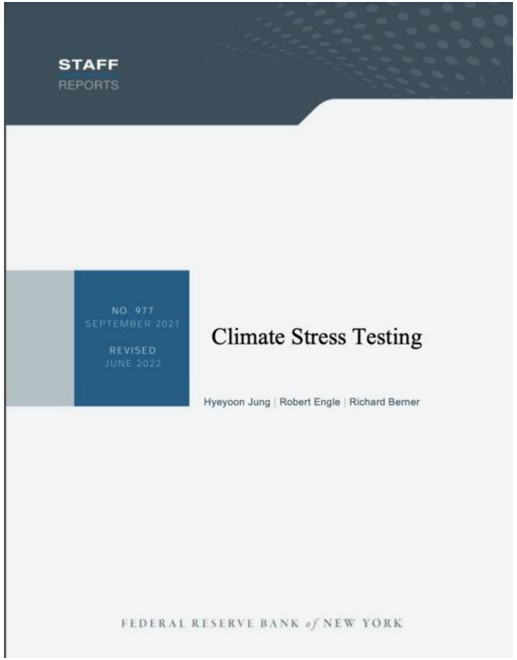
Content

- Introduction
- Overview of CRISK
- CRISK Implementation in MATLAB
- Conclusion

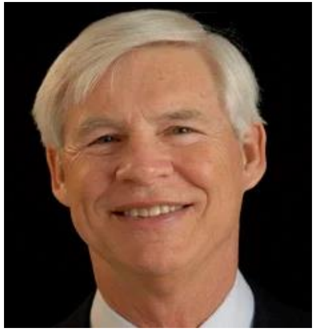
Introduction

- CRISK Paper
- Joint Project - MathWorks and Columbia

CRISK Paper



Hyeyoon Jung
Financial Research Economist, Federal Reserve



Robert Engle
Co-Director, Volatility & Risk Institute, NYU Stern
Won Nobel Prize in 2003 for GARCH model

Richard Berner
Co-Director, Volatility & Risk Institute, NYU Stern

Project Objective

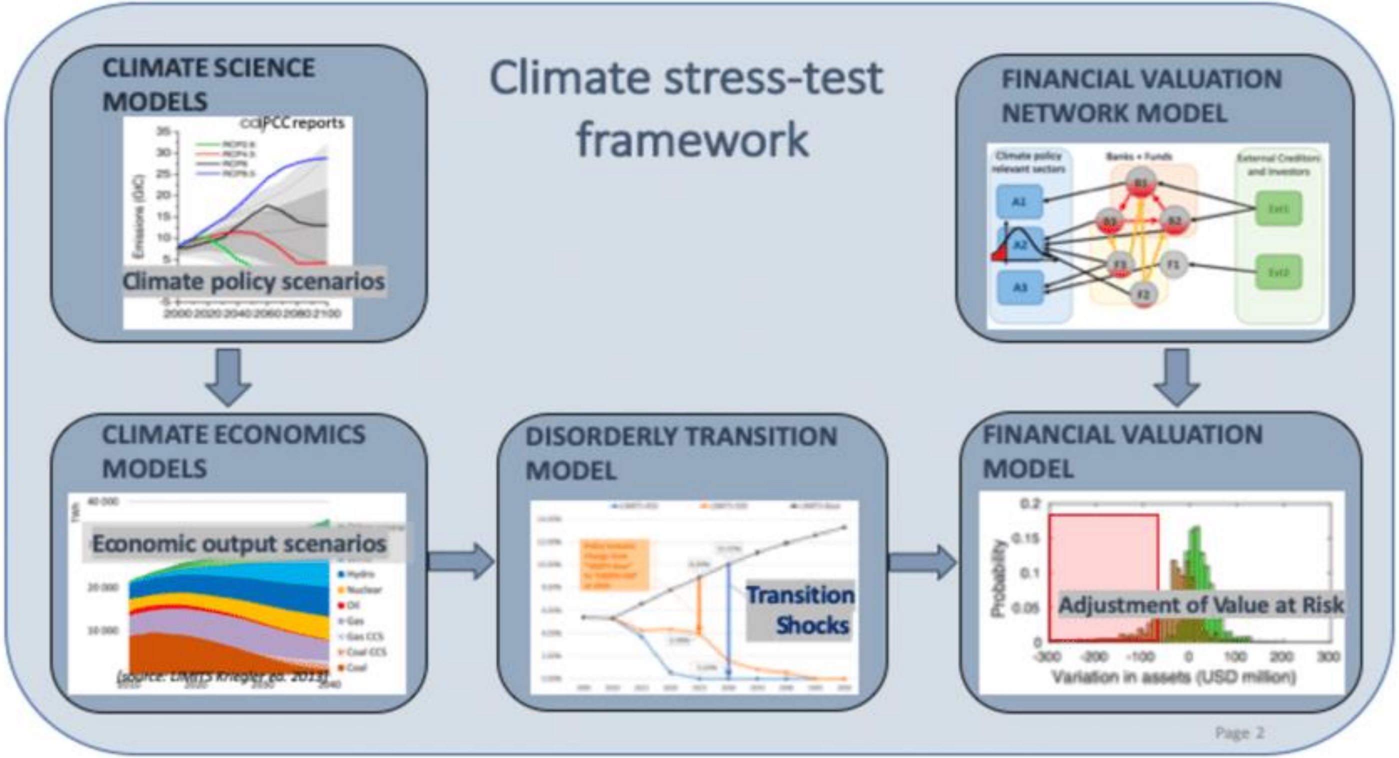
- Replicate the methodology in the paper
- Develop a user-friendly solution to enable dynamic analysis of financial institutions
- Leverage MathWorks technology for jurisdiction-specific customization, with easy production deployment and scaling across enterprises*

**Ongoing work*

Overview of CRISK

- Classic Climate Stress Testing Framework
- Classic Framework – Challenges and Solution
- CRISK Methodology Overview
- Implementation of CRISK by RBI

Classic Climate Stress Testing Framework



Classic Framework: Challenges, Solutions

Challenges		CRISK Solution
History may not capture current expectations	→	Climate beta
Perceptions change with time	→	Dynamic measurement
Lack of reliable data	→	Market data

Introduction to CRISK framework

Goal: Assess whether the banks are adequately capitalized to absorb transition and physical risks losses

What does CRISK aim to answer?	CRISK Framework

High level

How much capital does a firm have to raise if there is a climate stress scenario?

Detailed

Without market stress, when a stranded asset portfolio falls by **50%,** and a firm needs to maintain an **8%** of capital ratio to weather stress, what's the expected capital shortfall?

CRISK Framework

Leverage		Size	Risk
Prudential Ratio of Equity to Assets	Book Value of Debt	Market Value of Equity	Climate Beta

$$CRISK_{it} = k \cdot D_{it} - (1-k) \cdot W_{it} \cdot \exp\left(\beta_{it}^{Climate} \log(1-\theta)\right)$$

- Capital shortfall treated as a time series
- Relies only on publicly available data

Bloomberg yahoo! finance

CRISK Methodology

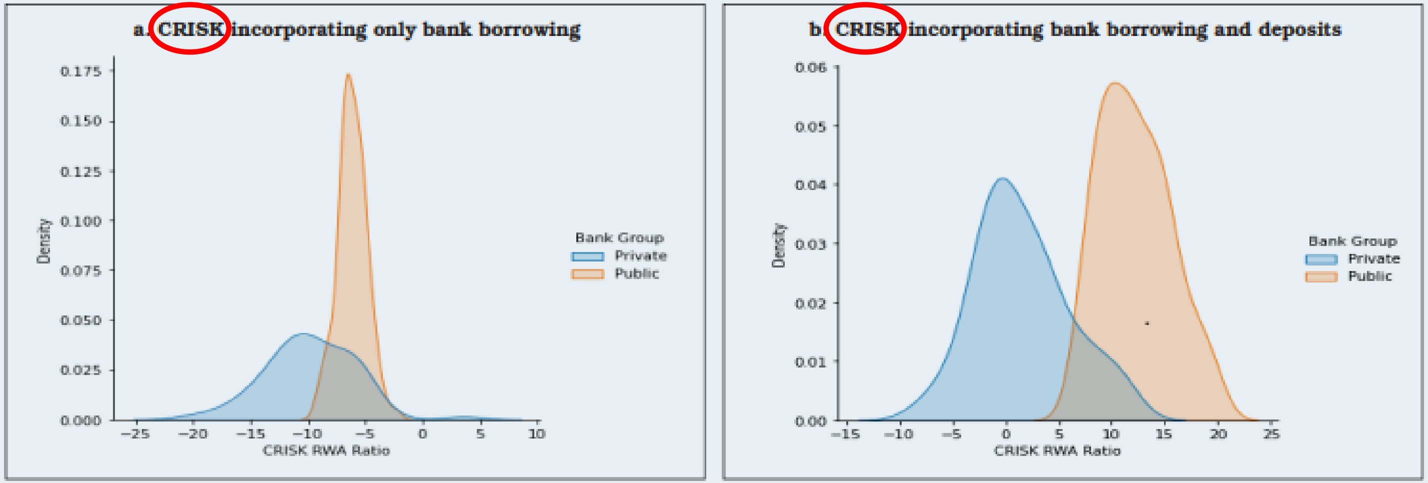
$$CF^{Str} = 0.3XLE + 0.7KOL - SPY$$

$$r_{it} = \beta_{it}^{Mkt} MKT_t + \beta_{it}^{Climate} CF_t + \varepsilon_{it}$$

$$CRISK_{it} = k \cdot D_{it} - (1-k) \cdot W_{it} \cdot \exp\left(\beta_{it}^{Climate} \log(1-\theta)\right)$$

Implementation by RBI (the central bank of India)

Chart III.12: Climate Stress Test: PSBs *vis-à-vis* PVBs



Source: Authors' calculations.

$$\begin{aligned} Climate\ Risk\ Portfolio\ (CRP)\ Return \\ = 0.3 * \underline{NIFTY}\ ENERGY\ Index\ Return \\ + 0.7 * \underline{COAL\ India}\ Limited\ Return \\ - \underline{NIFTY}\ Index\ Return \end{aligned}$$

→

$$r_{it} = \beta_{it}^{Mkt} \underline{NIFTY}_t + \beta_{it}^{Climate} CRP_t + \varepsilon_{it}$$

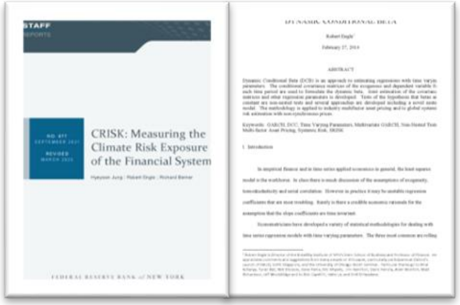
→

$$CRISK_{it} = k\ (D_{it}) - (1-k)\ E_{it} \exp(\beta_{it}^{Climate} \log(1-\theta))$$

Implementation

- Attribution
- Overview
- Adjustments
- Calculations
- Conclusions
- Appendix
  - Derivations
  - Code

Theory


STAFF REPORTS

CRISK: Measuring the Climate Risk Exposure of the Financial System

DYNAMIC CONDITIONAL BETA

ABSTRACT

Implementation

Hyeyoon Jung
Financial Research Economist, Federal Reserve

Robert Engle
Co-Director, Volatility & Risk Institute, NYU Stern
Won Nobel Prize in 2003 for GARCH model

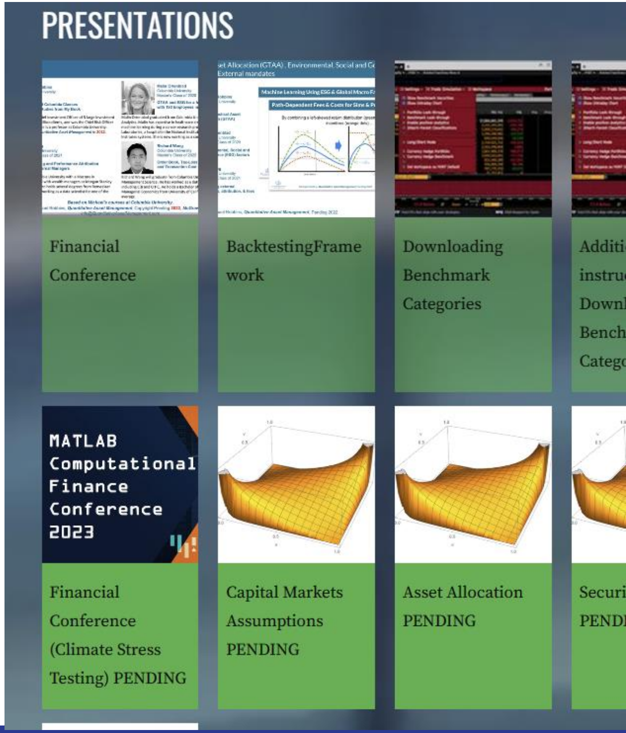
Richard Berner
Co-Director, Volatility & Risk Institute, NYU Stern

Alexandra Wang
MSOR, focus on Analytics
Computer Science and Business majors at UNC-Chapel Hill

Wenxiao Wu
MSOR, focus on risk management Financial Engineering major at Wuhan University

Clementine Mc Sweeny
MSOR, focus on finance
Master in Statistics, University of Paris
Engineering degree, Mines Nancy

Find this presentation video and slides at: quantitativeassetmanagement.com/webinar/



Implementation

- Attribution
- Overview
- Adjustments
- Calculations
- Conclusions
- Appendix
  - Derivations
  - Code

Capital Shortfall

Capital shortfall = capital reserves the bank needs - firm’s equity

$$CS_{it} = k(D_{it} + W_{it}) - W_{it}$$

The market value of equity

The book value of debt

The prudential ratio of equity to assets

HYEYOON JUNG

ROBERT ENGLE

RICHARD BERNER

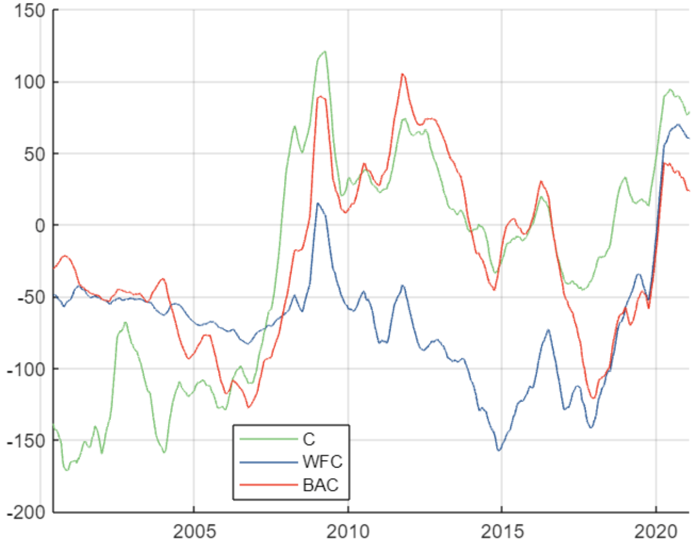
Implementation

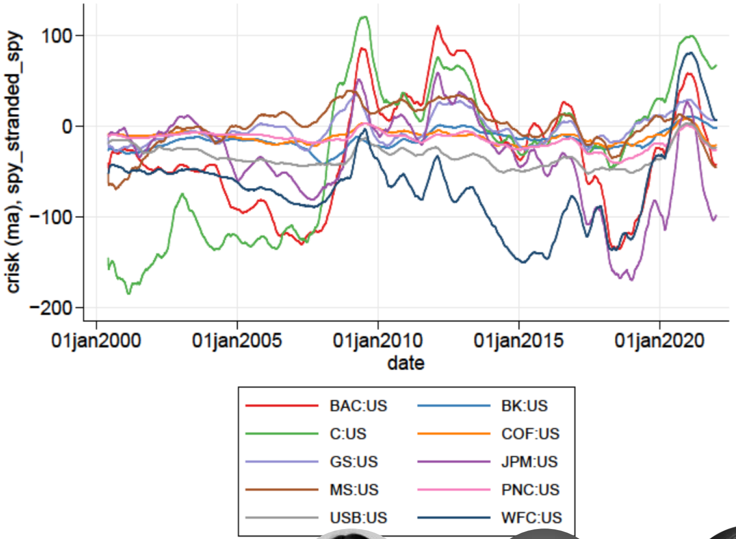
- Attribution
- Overview
- Adjustments
- Calculations
- Conclusions
- Appendix
  - Derivations
  - Code

Implementation

- Attribution
- Overview
- Adjustments
- Calculations
- Conclusions
- Appendix
    - Derivations
    - Code

CRISK Results from MATLAB Implementation





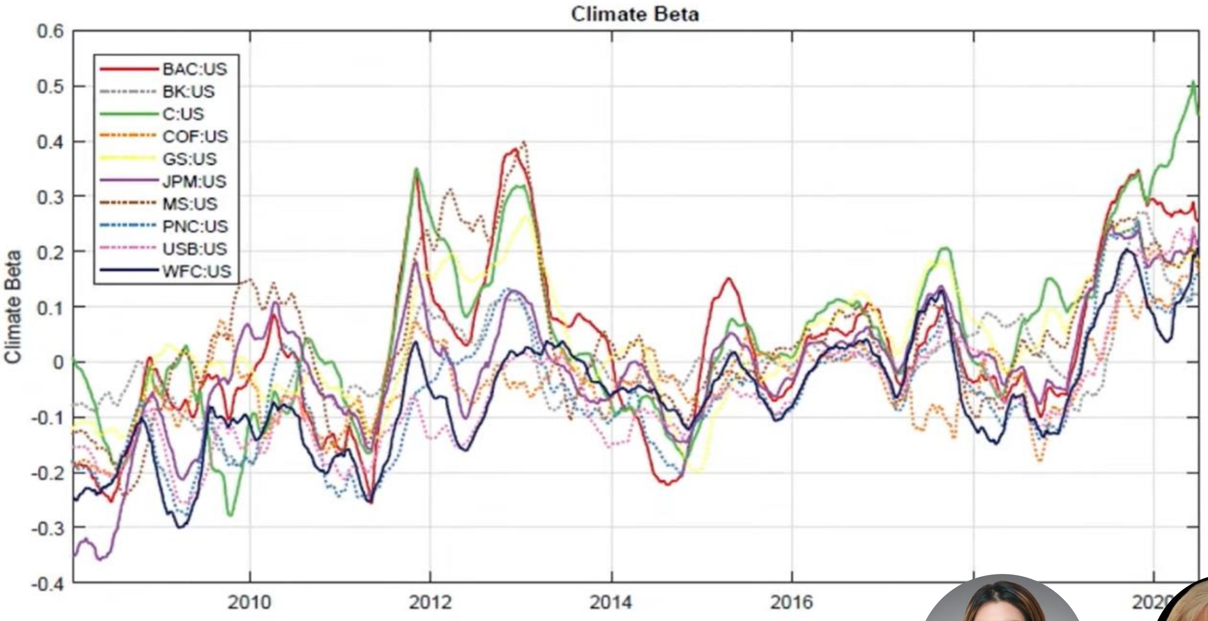

Alexandra Wang


Wenxiao Wu


Clementine Mc Sweeny

Implementation

Validation vs Results in The Paper



HYEYOON JUNG

ROBERT ENGLE

RICHARD BERNER

Impact and Takeaway

Result

Efficient and open-source implementation of climate stress testing procedure

Customize proxy for KOL

Customize for geography, e.g., India

CRISK code and logic will be publicized for financial institution clients to use through online documentation and official blogs

Selling Point of Paper

Federal Reserve Board

Simple procedure

Public data to Implement

Vlab maintained by NYU Stern

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

Why Matlab?

The MathWorks has about $1B year in revenue and 5,000 employees.
They sell a product that is similar to one offered for free.

I need answers, not tech. MATLAB gets me the answers.

They support me with advice and help building

Implementation

- Attribution
- Overview
- Adjustments
- Calculations
- Conclusions
- Appendix
    - Derivations
    - Code

CRISK Derivation

$$
\begin{aligned}
1 - LRMES_{it} &= E_t\left[1 + R^i_{t+1,t+h} \middle| \frac{P^{CF}_{t+h}}{P^{CF}_{t+1}} - 1 = -\theta, \frac{P^{Mkt}_{t+h}}{P^{Mkt}_{t+1}} - 1 = 0\right] \\
&= E_t\left[\exp\left(\sum_{j=1}^{h} r^i_{t+j}\right) \middle| \frac{P^{CF}_{t+h}}{P^{CF}_{t+1}} - 1 = -\theta, \frac{P^{Mkt}_{t+h}}{P^{Mkt}_{t+1}} - 1 = 0\right] \\
&= E_t\left[\exp\left(\sum_{j=1}^{h} \beta^{Mkt}_{i,t+j} r^{Mkt}_{t+j} + \beta^{Climate}_{i,t+j} r^{CF}_{t+j} + \varepsilon_{i,t+j}\right) \middle| \frac{P^{CF}_{t+h}}{P^{CF}_{t+1}} - 1 = -\theta, \frac{P^{Mkt}_{t+h}}{P^{Mkt}_{t+1}} - 1 = 0\right] \\
&= E_t\left[\exp\left(\beta^{Mkt}_{it} \log\left(\frac{P^{Mkt}_{t+1,t+j}}{P^{Mkt}_{t+1}}\right) + \beta^{CF}_{it} \log\left(\frac{P^{CF}_{t+1,t+j}}{P^{CF}_{t+1}}\right)\right) \middle| \frac{P^{CF}_{t+h}}{P^{CF}_{t+1}} - 1 = -\theta, \frac{P^{Mkt}_{t+h}}{P^{Mkt}_{t+1}} - 1 = 0\right] \\
&= \exp\left(\beta^{Climate}_{it} \log(1-\theta)\right)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
CRISK_{it} &= kD_{it} - (1-k)W_{it}\underbrace{\{1 + E_t[R^i_{t+1,t+h} | R^{CF}_{t+1,t+h} < C}_{1-LRMES_{it}} \\
&= kD_{it} - (1-k)W_{it}\exp\left(\beta^{Climate}_{it} \log(1-\theta)\right)
\end{aligned}
$$


HYEYOON JUNG


ROBERT ENGLE


RICHARD BERNER

DCB Model Estimation

$$r_{it} = \log(1 + R_{it}),\ r_{mt} = \log(1 + R_{mt}),\ r_{ct} = \log(1 + R_{ct})$$

Conditional on the information set $\mathcal{F}_{t-1}$, the return triple has a distribution $\mathcal{D}$ with zero mean and time-varying covariance:

$$\begin{bmatrix} r_{it} \\ r_{mt} \\ r_{ct} \end{bmatrix} \Bigg| \mathcal{F}_{t-1} \sim \mathcal{D}\left(\mathbf{0}, H_t = \begin{bmatrix} \sigma_{it}^2 & \rho_{imt}\sigma_{it}\sigma_{mt} & \rho_{ict}\sigma_{it}\sigma_{ct} \\ \rho_{imt}\sigma_{it}\sigma_{mt} & \sigma_{mt}^2 & \rho_{mct}\sigma_{mt}\sigma_{ct} \\ \rho_{ict}\sigma_{it}\sigma_{ct} & \rho_{mct}\sigma_{mt}\sigma_{ct} & \sigma_{ct}^2 \end{bmatrix}\right)$$

We use a GJR-GARCH volatility model and DCC correlation model. The GJR-GARCH model for volatility dynamics are:

$$\sigma_{it}^2 = \omega_{Vi} + \alpha_{Vi} r_{it-1}^2 + \gamma_{Vi} r_{it-1}^2 I_{i,t-1}^- + \beta_{Vi}\sigma_{it-1}^2, \tag{9}$$

$$\sigma_{mt}^2 = \omega_{Vm} + \alpha_{Vm} r_{mt-1}^2 + \gamma_{Vm} r_{mt-1}^2 I_{m,t-1}^- + \beta_{Vm}\sigma_{mt-1}^2, \tag{10}$$

$$\sigma_{ct}^2 = \omega_{Vc} + \alpha_{Vc} r_{ct-1}^2 + \gamma_{Vc} r_{ct-1}^2 I_{c,t-1}^- + \beta_{Vc}\sigma_{ct-1}^2 \tag{11}$$

where $I_{it}^- = 1$ if $r_{it} < 0$, $I_{mt}^- = 1$ if $r_{mt} < 0$, and $I_{ct}^- = 1$ if $r_{ct} < 0$.

The correlation of the volatility-adjusted returns $e_{it} = r_{it}/\sigma_{it}$, $e_{mt} = r_{mt}/\sigma_{mt}$, and $e_{ct} = r_{ct}/\sigma_{ct}$ is:

$$\text{Cor}\begin{pmatrix} \epsilon_{it} \\ \epsilon_{mt} \\ \epsilon_{ct} \end{pmatrix} = R_t = \begin{bmatrix} 1 & \rho_{imt} & \rho_{ict} \\ \rho_{imt} & 1 & \rho_{mct} \\ \rho_{ict} & \rho_{mct} & 1 \end{bmatrix} = \text{diag}(Q_{imct})^{-1/2}\, Q_{imct}\, \text{diag}(Q_{imct})$$

HYEYOON JUNG

ROBERT ENGLE

RICHARD BERNER

The DCC model specifies the dynamics of the pseudo-correlation matrix Q_{imct} as:

$$Q_{imct} = (1 - \alpha_{Ci} - \beta_{Ci})S_i + \alpha_{Ci} \begin{bmatrix} e_{it} \\ e_{mt} \\ e_{ct} \end{bmatrix} \begin{bmatrix} e_{it} \\ e_{mt} \\ e_{ct} \end{bmatrix}' + \beta_{Ci} Q_{imct-1} \tag{12}$$

where S_{it} is the unconditional correlation matrix of adjusted returns.
The market beta β_{it}^{Mkt} and the climate beta $\beta_{it}^{Climate}$ are:

$$\begin{bmatrix} \beta_{it}^{Mkt} \\ \beta_{it}^{Climate} \end{bmatrix} = \begin{bmatrix} \sigma_{mt}^2 & \rho_{mct}\sigma_{mt}\sigma_{ct} \\ \rho_{mct}\sigma_{mt}\sigma_{ct} & \sigma_{ct}^2 \end{bmatrix}^{-1} \begin{bmatrix} \rho_{imt}\sigma_{it}\sigma_{mt} \\ \rho_{ict}\sigma_{it}\sigma_{ct} \end{bmatrix} \tag{13}$$

Estimation procedure is as follows:

1. For each bank $i = 1 \cdots N$, estimate GARCH parameters and DCC parameters.

2. Take the median DCC parameters, $\alpha_{\bar{C}} = \text{median}(\alpha_{Ci})$ and $\beta_{\bar{C}} = \text{median}(\beta_{Ci})$.

3. Compute β_{it}^{Mkt} and $\beta_{it}^{Climate}$ based on the median DCC parameters, $\alpha_{\bar{C}}$ and $\beta_{\bar{C}}$, and the volatility parameters.[30]

HYEYOON JUNG

ROBERT ENGLE

RICHARD BERNER

Implementation

- Attribution
- Overview
- Adjustments
- Calculations
- Conclusions
- Appendix
  - Derivations
  - Code

GJR

```
function [parameters, ll ,Ht, VCV, scores, diagnostics] =
            dcc(data,dataAsym,m,l,n,p,o,q,gjrType,method,composite,startingVals,options)
%GJR Create a GJR conditional variance model

…

%   Create a GJR(P,Q) conditional variance model by specifying either degrees P
%   and Q (short-hand syntax) or a list of parameter name-value arguments (long-hand
%   syntax). For conditional variances v(t) and innovations e(t), either syntax
%   creates a GJR(P,Q) model of the form
%
%   v(t) = c + g1*v(t-1) + ... + gP*v(t-P) + a1*e(t-1)^2 + ... + aQ*e(t-Q)^2
%            + L1*[e(t-1) < 0]*e(t-1)^2 + ... + LQ*[e(t-Q) < 0]*e(t-q)^2
%
%   In the above equation, terms such as [e(t-i) < 0] represent a logical indicator:
%   true if the i-th innovation is negative, and false otherwise.

…

% Copyright 2023 The MathWorks, Inc.
```

DCC

```
function [parameters, ll ,Ht, VCV, scores, diagnostics] =
           dcc(data,dataAsym,m,l,n,p,o,q,gjrType,method,composite,startingVals,options)
% Estimation of scalar DCC(m,n) and ADCC(m,l,n) multivarate volatility model with with TARCH(p,o,q)
% or GJRGARCH(p,o,q) conditional variances

…

%   The dynamics of a the correlations in a DCC model are:
%     3-stage:
%     Q(t) = R*(1-sum(a)-sum(b))-sum(g)*N + a(1)*e(t-1)'*e(t-1) + ... + a(m)*e(t-m)'*e(t-m)
%     + g(1)*v(t-1)'*v(t-1) + ... + g(l)*v(t-l)*v(t-l) + b(1)*Q(t-1) + ... + b(n)*Q(t-1)
%
%     2-stage
%     Q(t) = R.*scale + a(1)*e(t-1)'*e(t-1) + ... + a(m)*e(t-m)'*e(t-m)
%     + g(1)*v(t-1)'*v(t-1) + ... + g(l)*v(t-l)*v(t-l) + b(1)*Q(t-1) + ... + b(n)*Q(t-1)
%
%   where v(t,:) = e(t,;).*(e(t,:)<0) and s = sqrt((1-sum(a)-sum(b)-gScale*sum(g))) and scale = s*s'

…

% Copyright: Kevin Sheppard
% kevin.sheppard@economics.ox.ac.uk
% Revision: 1 Date: 4/13/2012
```

GARCHGJR

```
function [alpha,alphaSE,gamma,gammaSE,beta,betaSE,ConVar] = …
          GARCHGJR(i,bank_data,field)

    arguments
        i                (1,1) double
        bank_data        (1,:) struct
        field            (1,1) string = ""
    end

    bank_Ret = bank_data(i).logreturn.(field);

    Mdl     = gjr('GARCHLags',1,'ARCHLags',1,'LeverageLags',1);
    EstMdl  = estimate(Mdl,bank_Ret,'Display','off');
    sumMdl  = summarize(EstMdl);
    alpha   = sumMdl.Table{3,1};
    alphaSE = sumMdl.Table{3,2};
    beta    = sumMdl.Table{2,1};
    betaSE  = sumMdl.Table{2,2};
    gamma   = sumMdl.Table{4,1};
    gammaSE = sumMdl.Table{4,2};
    ConVar  = infer(EstMdl,bank_Ret);
end
```



Alexandra Wang


Wenxiao Wu


Clementine Mc Sweeny

GARCHDCC

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

```
function [alpha,beta,mkt_beta,cf_beta] = GARCHDCC(i,bank,US_market,bank_data)

arguments
    i                (1,1) double
    bank             (1,1) string
    US_market        (1,1) logical
    bank_data        (1,:) struct
end

if US_market
    bank_Ret = bank_data(i).logreturn.(bank);
    [~,~,~,~,~,~,BankConVar] = CRISK.beta.model.GARCHGJR(i,bank_data,bank);
    Bankstd = sqrt(BankConVar);
    Bank_adj_Ret = bank_Ret ./ Bankstd;
    Bank_adj_Ret = Bank_adj_Ret - mean(Bank_adj_Ret);

    % p: epsilon lag: shock
    % o: asymmetry lag: volatility increases/ decreases as price increases/ decreases
    % q: volatility lag: previous time periods' volatility
    % m: first model

    CF = bank_data(i).logreturn.CF;
    MKT = bank_data(i).logreturn.MKT;
    [~,~,~,~,~,~,CFConVar] = CRISK.beta.model.GARCHGJR(i,bank_data,'CF'); % GJR on CF
    CFstd = sqrt(CFConVar);
    CF_adj_Ret = CF ./ CFstd;
    CF_adj_Ret = CF_adj_Ret - mean(CF_adj_Ret);

    [~,~,~,~,~,~,MKTConVar] = CRISK.beta.model.GARCHGJR(i,bank_data,'MKT'); % GJR on MKT
    MKTstd = sqrt(MKTConVar);
    MKT_adj_Ret = MKT ./ MKTstd;
    MKT_adj_Ret = MKT_adj_Ret - mean(MKT_adj_Ret);

    data = [Bank_adj_Ret,MKT_adj_Ret,CF_adj_Ret];
    [parameters_dcc, ~, Ht]= CRISK.mfe.dcc(data,[],1,1,1);
```

```
% parameters
alpha = parameters_dcc(1,end-2);
beta = parameters_dcc(1,end);

% Correlation matrix
s = size(Ht);
corr_mat = Ht;
for i = 1:s(3)
    h = Ht(:,:,i);
    corr_mat(:,:,i) = corrcov(h);
end

coef = zeros(2,1,s(3));
mkt_beta = zeros(1,s(3));
cf_beta = zeros(1,s(3));
for i = 1:s(3)
    c = corr_mat(:,:,i);
    pmct = c(2,3);
    pimt = c(1,2);
    pict = c(1,3);
    bankvar = BankConVar(i);
    mkvar = MKTConVar(i);
    cfvar = CFConVar(i);
    d1 = [mkvar,pmct*sqrt(mkvar)*sqrt(cfvar); …
             pmct*sqrt(mkvar)*sqrt(cfvar),cfvar];
    d2 = [pimt*sqrt(bankvar)*sqrt(mkvar); …
             pict*sqrt(bankvar)*sqrt(cfvar)];
    coef(:,:,i) = d1\d2;
    mkt_beta(1,i) = coef(1,1,i);
    cf_beta(1,i) = coef(2,1,i);
end
```

```
else
    bank_Ret = bank_data(i).logreturn.(bank); % get time series of bank return
    [~,~,~,~,~,~,BankConVar] = CRISK.beta.model.GARCHGJR(i,bank_data,bank);
    Bankstd = sqrt(BankConVar);
    Bank_adj_Ret = bank_Ret ./ Bankstd;
    Bank_adj_Ret = Bank_adj_Ret - mean(Bank_adj_Ret);
    % p: epsilon lag: shock
    % o: asymmetry lag: volatility increases/ decreases as price up/down
    % q: volatility lag: previous time periods' volatility
    % m: first model
    CF = bank_data(i).logreturn.CF;
    MKT = bank_data(i).logreturn.MKT;
    [~,~,~,~,~,~,CFConVar] = CRISK.beta.model.GARCHGJR(i,bank_data,'CF');
    CFstd = sqrt(CFConVar);
    CF_adj_Ret = CF ./ CFstd;
    CF_adj_Ret = CF_adj_Ret - mean(CF_adj_Ret);
    [~,~,~,~,~,~,MKTConVar] = CRISK.beta.model.GARCHGJR(i,bank_data,'MKT');
    MKTstd = sqrt(MKTConVar);
    MKT_adj_Ret = MKT ./ MKTstd;
    MKT_adj_Ret = MKT_adj_Ret - mean(MKT_adj_Ret);
    data = [Bank_adj_Ret,MKT_adj_Ret,CF_adj_Ret];
    data2 = [Bank_adj_Ret(2:end,1),MKT_adj_Ret(1:end-1,1), …
             CF_adj_Ret(1:end-1,1)];
    [parameters_dcc, ~, Ht]= CRISK.mfe.dcc(data,[],1,1,1);
    [~, ~, Ht2]= CRISK.mfe.dcc(data2,[],1,1,1);

    % parameters
    alpha = parameters_dcc(1,end-2);
    beta = parameters_dcc(1,end);

    % Correlation matrix
    s = size(Ht);
    corr_mat = Ht;
    for i = 1:s(3)
        h = Ht(:,:,i);
        corr_mat(:,:,i) = corrcov(h);
    end

    s2 = size(Ht2);
    corr_mat2 = Ht2;
    for i = 1:s2(3)
        h2 = Ht2(:,:,i);
        corr_mat2(:,:,i) = corrcov(h2);
    end

    coef1 = zeros(2,1,s(3));
    mkt_beta1 = zeros(1,s(3));
    cf_beta1 = zeros(1,s(3));
    for i = 1:s(3)
        c = corr_mat(:,:,i);
        pmct = c(2,3);
        pimt = c(1,2);
        pict = c(1,3);
        bankvar = BankConVar(i);
        mkvar = MKTConVar(i);
        cfvar = CFConVar(i);
        d1 = [mkvar,pmct*sqrt(mkvar)*sqrt(cfvar);
              pmct*sqrt(mkvar)*sqrt(cfvar),cfvar];
        d2 = [pimt*sqrt(bankvar)*sqrt(mkvar);pict*sqrt(bankvar)*sqrt(cfvar)];
        coef1(:,:,i) = d1\d2;
        mkt_beta1(1,i) = coef1(1,1,i);
        cf_beta1(1,i) = coef1(2,1,i);
    end

    coef2 = zeros(2,1,s2(3));
    mkt_beta2 = zeros(1,s2(3));
    cf_beta2 = zeros(1,s2(3));
    for i = 1:s2(3)
        c2 = corr_mat(:,:,i);
        pmct2 = c2(2,3);
        pimt2 = c2(1,2);
        pict2 = c2(1,3);
        bankvar2 = BankConVar(i+1);
        mkvar2 = MKTConVar(i+1);
        cfvar2 = CFConVar(i+1);
        d12 = [mkvar2,pmct2*sqrt(mkvar2)*sqrt(cfvar2); …
               pmct2*sqrt(mkvar2)*sqrt(cfvar2),cfvar2];
        d22 = [pimt2*sqrt(bankvar2)*sqrt(mkvar2); …
               pict2*sqrt(bankvar2)*sqrt(cfvar2)];
        coef2(:,:,i) = d12\d22;
        mkt_beta2(1,i) = coef2(1,1,i);
        cf_beta2(1,i) = coef2(2,1,i);
    end

    mkt_beta = zeros(1,s2(3));
    cf_beta = zeros(1,s2(3));
    for i = 1:s2(3)
        mkt_beta(1,i) = mkt_beta2(1,i) + mkt_beta1(1,i+1);
        cf_beta(1,i) = cf_beta2(1,i) + cf_beta1(1,i+1);
    end
end
end
```

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

How to calculate CRISK ?

Table of Contents

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

1. Importing and Preprocessing data

1) Data import

a) Importing climate factor indicators

- Import historical prices for **3 ETFs (XLE, KOL, SPY)**, **banks** and **1 market index** (ACWI) from Jan 01 2000 to Jan 31 2021.

```
XLE = readtimetable("climate_factor_indicators.xlsx","Sheet","XLE"); % Import Prices for XLE
KOL = readtimetable("climate_factor_indicators.xlsx","Sheet","KOL"); % Import Prices for KOL
SPY = readtimetable("climate_factor_indicators.xlsx","Sheet","SPY"); % Import Prices for SPY
```

b) Importing banks data

```
ACWI = readtimetable("market_data.xlsx","Sheet","ACWI"); % Import Prices for ACWI
banks_info = readtable("market_data.xlsx","Sheet","banks_info") % get information of all the banks
```

banks_info = 12×4 table

	bank	Country	ticker	code
1	'Citi'	'US'	'C'	'C:US'
2	'Wells Fargo'	'US'	'WFC'	'WFC:US'
3	'BofA Securities'	'US'	'BAC'	'BAC:US'
4	'JP Morgan'	'US'	'JPM'	'JPM:US'
5	'BNP Paribas'	'France'	'BNP'	'BNP:FP'
6	'Societe Generale'	'France'	'GLE'	'GLE:FP'
7	'HSBC'	'UK'	'HSBA'	'HSBA:LN'
8	'Barclays'	'UK'	'BARC'	'BARC:LN'
9	'Mitsubishi UFJ Financial Group Inc'	'Japan'	'8306.T'	'8306.T:JP'
10	'Mizuho Financial'	'Japan'	'8411.T'	'8411.T:JP'
11	'RBC Capital Markets'	'Canada'	'RY'	'RY:CN'

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

	bank	Country	ticker	code
12	'TD Securities'	'Canada'	'TD'	'TD:CN'

c) Merging market index and climate index and displaying the raw data

```
ETF_data = synchronize(XLE,KOL,SPY);
figure;
stackedplot(ETF_data)
grid on
title("ETF closed prices")
```

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

2) Data Preprocessing

a) Preprocess KOL and fill missing data by using returns of top 4 coal companies

- For KOL which started in 2008 and was liquidated in 2020, import historical prices for top 4 coal companies as ... ere we use BHP, Rio, CSUAY, NGLOY)



Alexandra Wang
Wenxiao Wu
Clementine Mc Sweeny

```
KOL_ret = CRISK.CF.utils.preprocessKOL(KOL);
```

2. Calculate Climate Risk Factor (CF)

- **Climate risk factor:** We use ***stranded asset portfolio return*** as a market-based proxy measure for ***transition risk*** as studies suggest that climate risks are priced in the equity market

$$CF^{Str} = 0.3XLE + 0.7KOL - SPY$$

It consists of ***a long position in the stranded asset index*** comprised of 30% in Energy Select Sector SPDR ETF (XLE) and 70% in VanEck Vectors Coal ETF (KOL), and ***a short position*** in SPDR S&P 500 ETF Trust (SPY).

- **Comparisons in statistics:**

A.1 Return Data

	count	mean	sd	min	max
SPY	5206	0.0002	0.0123	-0.1159	0.1356
ACWI	5206	0.0002	0.0123	-0.1190	0.1170
0.7KOL+0.3XLE	5206	-0.0002	0.0197	-0.1819	0.1233
0.7KOL+0.3XLE-SPY	5206	-0.0004	0.0139	-0.1259	0.0901

Table A.1: Market Returns and Climate Factors Summary Statistics Daily log returns for June 2000 – Dec 2021.

stats2 = 5x4 table

		XLE	KOL	SPY	0.7KOL+0.3XLE-SPY
1	count	5197	5197	5197	5197
2	mean	4.4210e-05	1.8743e-04	1.1402e-04	-3.6220e-05
3	sd	0.0182	0.0122	0.0231	0.0144
4	min	-0.2154	-0.1386	-0.3232	-0.1297
5	max	0.1510	0.1092	0.2993	0.1644

1) Calculate Climate Risk Factor (CF) using daily returns

a. Computing Climate Factor with our Indicators

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

- Concatenate timetables of 2 ETFs
- Calculate daily returns using historical prices

```
ETF = synchronize(XLE,SPY);
ETF_Return = tick2ret(ETF);
ETF_Return = synchronize(ETF_Return,KOL_ret);
```

b. Selecting daily returns

```
isLogReturn = false;
[ETF_Return_CF, stats] = CRISK.CF.computeClimateFactor(ETF_Return,isLogReturn);
```

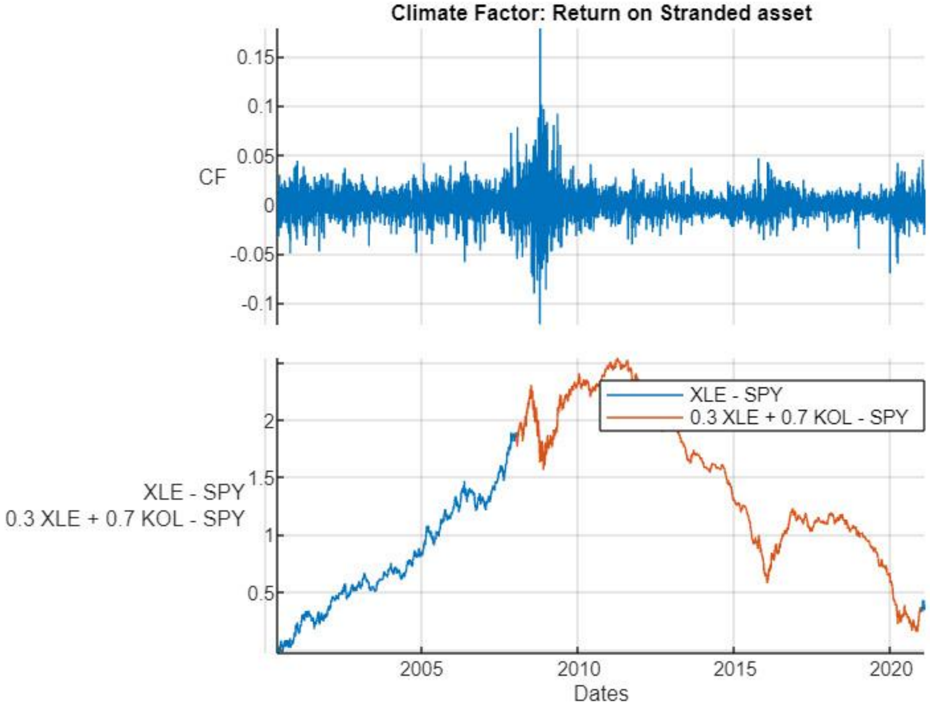
c. Plotting Climate Factor daily returns

```
CRISK.CF.utils.plotClimateFactor(ETF_Return_CF, KOL);
```

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny



d. Displaying statistic table

```
stats
```

stats = 5x4 table

	XLE	KOL	SPY	0.7KOL+0.3XLE-SPY
1 count	5197	5197	5197	5197

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

	XLE	KOL	SPY	0.7KOL+0.3XLE-SPY
2 mean	2.0863e-04	2.6125e-04	3.7966e-04	6.7097e-05
3 sd	0.0181	0.0121	0.0230	0.0144
4 min	-0.1938	-0.1294	-0.2761	-0.1217
5 max	0.1630	0.1154	0.3489	0.1787

2) Calculate Climate Risk Factor (CF) using daily log returns

a. Selecting daily log returns

```
isLogReturn = true;
[ETF_LogReturn_CF, stats_LogReturn] = CRISK.CF.computeClimateFactor(ETF_Return,isLogReturn);
```

b. Displaying statistic table

```
stats_LogReturn
```

stats_LogReturn = 5x4 table

	XLE	KOL	SPY	0.7KOL+0.3XLE-SPY
1 count	5197	5197	5197	5197
2 mean	4.4210e-05	1.8743e-04	1.1402e-04	-3.6220e-05
3 sd	0.0182	0.0122	0.0231	0.0144
4 min	-0.2154	-0.1386	-0.3232	-0.1297
5 max	0.1510	0.1092	0.2993	0.1644

3. Betas Estimation

This section will show how to estimate Betas using three different techniques:

- Fixed Beta Estimation
- Rolling Windows Beta Estimation
- Dynamic Conditional Beta (DCB) Estimation

1) Data Preparation

a) Build Matrix for Calculating Beta

- Include predictors (MKT, CF) in one matrix.

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

Using daily return

```
features = CRISK.beta.utils.getBetaFeatures(ETF_Return_CF)
```

features = 5197×2 timetable

	Dates	MKT	CF
1	01-Jun-2000	0.0079	0.0105
2	02-Jun-2000	0.0365	-0.0327
3	05-Jun-2000	-0.0099	0.0097
4	06-Jun-2000	-0.0057	0.0074
5	07-Jun-2000	0	0.0128
6	08-Jun-2000	0.0060	-0.0043
7	09-Jun-2000	0	0.0016
8	12-Jun-2000	-0.0036	-0.0044
9	13-Jun-2000	-0.0147	0.0299
10	14-Jun-2000	0.0237	-0.0223
11	15-Jun-2000	-0.0008	0.0067
12	16-Jun-2000	0.0013	0.0189
13	19-Jun-2000	-0.0124	-0.0056
14	20-Jun-2000	0.0117	-0.0245

⋮

Using daily log return

```
features_log = CRISK.beta.utils.getBetaFeatures(ETF_LogReturn_CF)
```

features_log = 5197×2 timetable

	Dates	MKT	CF
1	01-Jun-2000	0.0079	0.0104
2	02-Jun-2000	0.0359	-0.0333
3	05-Jun-2000	-0.0099	0.0096
4	06-Jun-2000	-0.0057	0.0074
5	07-Jun-2000	0	0.0127
6	08-Jun-2000	0.0059	-0.0044

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

	Dates	MKT	CF
7	09-Jun-2000	0	0.0016
8	12-Jun-2000	-0.0036	-0.0044
9	13-Jun-2000	-0.0148	0.0295
10	14-Jun-2000	0.0235	-0.0225
11	15-Jun-2000	-0.0008	0.0067
12	16-Jun-2000	0.0013	0.0187
13	19-Jun-2000	-0.0125	-0.0056
14	20-Jun-2000	0.0117	-0.0248

⋮

b) Compute returns for banks

- Calculate simple daily returns and daily log returns separately for each bank.

```
bank_return = CRISK.beta.utils.getBankReturns(banks_info, features);
bank_return_table = struct2table(bank_return)
```

bank_return_table = 12×3 table

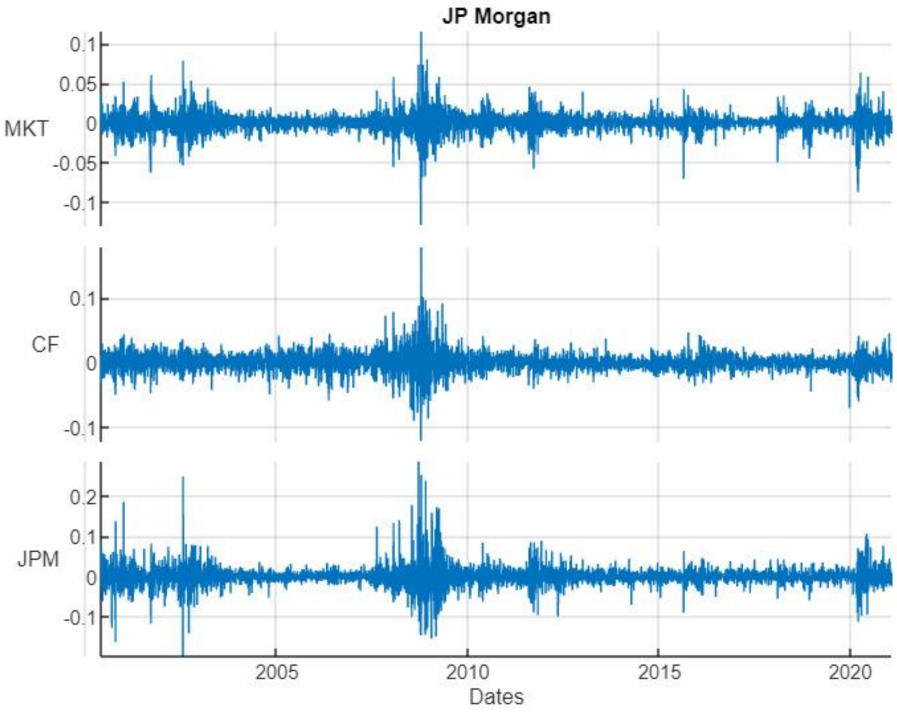
	bank	return	logreturn
1	'Citi'	5197×3 timetable	5197×3 timetable
2	'Wells Fargo'	5197×3 timetable	5197×3 timetable
3	'BofA Securities'	5197×3 timetable	5197×3 timetable
4	'JP Morgan'	5197×3 timetable	5197×3 timetable
5	'BNP Paribas'	5139×3 timetable	5139×3 timetable
6	'Societe Generale'	5124×3 timetable	5124×3 timetable
7	'HSBC'	5133×3 timetable	5133×3 timetable
8	'Barclays'	5197×3 timetable	5197×3 timetable
9	'Mitsubishi UFJ Financial Group Inc'	3661×3 timetable	3661×3 timetable
10	'Mizuho Financial'	4304×3 timetable	4304×3 timetable
11	'RBC Capital Markets'	5053×3 timetable	5053×3 timetable
12	'TD Securities'	5053×3 timetable	5053×3 timetable

- Plotting bank return



Alexandra Wang
Wenxiao Wu
Clementine Mc Sweeny

```
bank_index = 4;
stackedplot(bank_return(bank_index).return)
grid on
title(bank_return_table{bank_index,1})
```



2) Fixed Beta Estimation - using defined function fixed_beta

What does beta and gamma represent?

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

- The beta and gamma in this regression reflect the sensitivity of bank i to broad market declines and to climate deterioration. One would expect that banks with many loans to the fossil fuel industry will be more sensitive to CF than average and will have positive γ.

What data is used for MKT and CF?

- MKT denotes return on market and SPY is used. For CF, the return on the stranded asset portfolio CF_Str is used.
- Full sample period is 01/01/2000 - 01/31/2021 and post-crisis sample period is 01/01/2010 - 01/31/2021. We will calculate beta for both periods.
- According to paper, **Standard errors** are Newey-West adjusted with optimally selected number of lags.

For each firm i:

$$r_{it} = \alpha + \beta_i \cdot \mathrm{MKT}_t + \gamma_i \cdot \mathrm{CF}_t + \varepsilon_{it}$$

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

Bank	Ticker	CF	tstatCF	MKT	tstatMKT	CONS	tstatCONS	Rsq	N
BankofAmericaCorp	BAC	0.09	1.98	1.54	13.8	−0.0001	−0.34	0.46	5,444
CitigroupInc	C	0.07	1.63	1.67	16.98	−0.0005	−1.9	0.47	5,444
WellsFargoCo	WFC	0.05	1.19	1.29	12.42	0	0.06	0.45	5,444
BankofNewYorkMellonCorpThe	BK	0.04	1.16	1.35	19.22	−0.0001	−0.78	0.51	5,444
PNCFinancialServicesGroupIncThe	PNC	0.01	0.22	1.25	12.81	0.0001	0.74	0.43	5,444
CapitalOneFinancialCorp	COF	0	−0.08	1.59	18.33	0	−0.16	0.43	5,444
USBancorp	USB	−0.02	−0.53	1.15	15.25	0.0001	0.57	0.43	5,444
GoldmanSachsGroupIncThe	GS	−0.03	−0.93	1.37	29.19	0	0.16	0.53	5,444
MorganStanley	MS	−0.05	−1.19	1.82	16.61	−0.0002	−0.9	0.55	5,444
JPMorganChaseCo	JPM	−0.05	−1.25	1.47	20	0	0.25	0.56	5,444

Table C.1: Large Banks, SPY

Bank	Ticker	CF	tstatCF	MKT	tstatMKT	CONS	tstatCONS	Rsq	N
CitigroupInc	C	0.3	5.1	1.53	26.6	−0.0003	−1.16	0.61	2,832
BankofAmericaCorp	BAC	0.24	4.7	1.47	25.09	−0.0003	−0.86	0.55	2,832
MorganStanley	MS	0.23	4.89	1.53	26.79	−0.0002	−0.89	0.6	2,832
JPMorganChaseCo	JPM	0.18	4.01	1.27	35.75	0	0.02	0.62	2,832
CapitalOneFinancialCorp	COF	0.16	2.7	1.38	18	−0.0002	−0.64	0.52	2,832
GoldmanSachsGroupIncThe	GS	0.15	3.86	1.25	31.64	−0.0003	−1.23	0.57	2,832
BankofNewYorkMellonCorpThe	BK	0.14	3.5	1.15	31.74	−0.0003	−1.41	0.55	2,832
WellsFargoCo	WFC	0.13	2.13	1.27	24	−0.0004	−1.63	0.57	2,832
PNCFinancialServicesGroupIncThe	PNC	0.11	2.35	1.22	21.27	−0.0001	−0.33	0.58	2,832
USBancorp	USB	0.09	1.77	1.15	21.62	−0.0002	−1.03	0.58	2,832

Table C.2: Large Banks, SPY, Post-crisis

a) Set time range: full time and post-crisis period

```
timeframe = "post-crisis";
```

b) Fixed beta for all banks

```
stats = CRISK.beta.computeFixedBeta(banks_info,bank_return,timeframe)
```

stats = 12x10 table

	Bank	Ticker	CF	tstatCF	MKT	tstatMKT	CONS
1	"CII"	"C"	0.2757	10.1024	1.5363	55.4352	-9.2920e-05

	Bank	Ticker	CF	tstatCF	MKT	tstatMKT	CONS
2	"Wells Fargo"	"WFC"	0.1143	4.9035	1.2157	51.3493	-2.9668e-04
3	"BofA Securities"	"BAC"	0.2393	8.0575	1.5122	50.1509	-5.4397e-05
4	"JP Morgan"	"JPM"	0.1908	8.7231	1.2454	56.0688	1.0158e-04
5	"BNP Paribas"	"BNP"	0.2288	5.5530	1.0030	24.0046	-1.7101e-04
6	"Societe Generale"	"GLE"	0.2617	5.6385	1.0063	21.3820	-3.4352e-04
7	"HSBC"	"HSBA"	0.1261	5.5276	0.4916	21.2062	-2.5345e-04
8	"Barclays"	"BARC"	0.4911	12.9877	1.5277	39.7902	-3.0990e-04
9	"Mitsubishi UFJ Financial Group Inc"	"8306.T"	0.0169	0.4567	0.4713	12.7314	2.4351e-05
10	"Mizuho Financial"	"8411.T"	0.0382	1.1593	0.3333	10.0848	3.0992e-05
11	"RBC Capital Markets"	"RY"	0.1146	7.2653	0.6587	40.9994	7.7299e-05
12	"TD Securities"	"TD"	0.1152	7.5272	0.6570	42.1606	1.4108e-04

3) Rolling Window Beta Estimation - using defined function rolling_beta

- 252-day rolling window regressions from June 2000 to December 2021.

Why do we use 252-day?

- Because it's the number of days the stock market is open in one year.

Why do we use rolling window beta instead of fixed beta?

- In finance, nothing remains constant across time and that is why we use to report moving averages etc. Thus, it makes total sense to define a rolling window for monitoring the market beta and to see how it evolves across time.

What are our findings using rolling window beta estimation?

- We confirm that climate betas became significant in recent times based on rolling-window regressions

How is rolling window conducted?

- Instead of having one beta generated from all market return and climate factor data, we have a beta for each da regression of 252 days of return data before that day. For you to fully understand it, here's a YouTube video that calculate rolling window beta on Excel: https://www.youtube.com/watch?v=UwcjZXd3nVk



Alexandra Wang

Wenxiao Wu

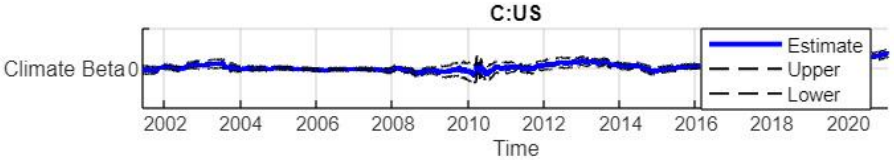
Clementine Mc Sweeny

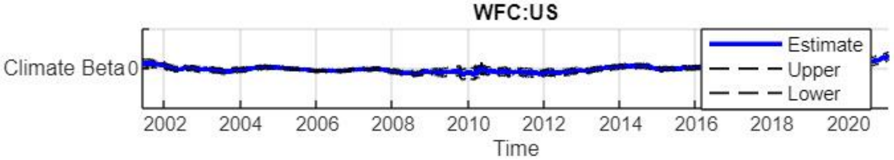
a) Setting country zone

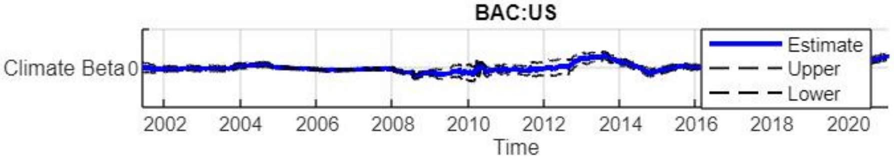
```
country = "US";
```

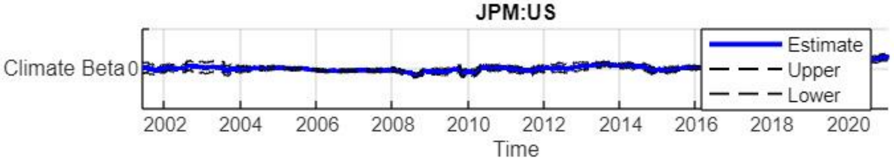
b) Rolling Beta estimation

```
verbose = true;
stats = CRISK.beta.computeRollingWindowBeta(banks_info,bank_return,country,verbose)
```









Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

stats = 4×9 table

Bank	Ticker	CF	tstatCF	MKT	tstatMKT	CONS	tstatCONS
"Citi"	"C"	1×4946 double	1×4946 double	1×4946 double	1×4946 double	1×4946 double	1×4946 double
"Wells Fargo"	"WFC"	1×4946 double	1×4946 double	1×4946 double	1×4946 double	1×4946 double	1×4946 double
"BofA Securities"	"BAC"	1×4946 double	1×4946 double	1×4946 double	1×4946 double	1×4946 double	1×4946 double
"JP Morgan"	"JPM"	1×4946 double	1×4946 double	1×4946 double	1×4946 double	1×4946 double	1×4946 double

4) DCB Model Estimation

- We use the Dynamic Conditional Beta (DCB) model to estimate the time-varying climate betas on a daily basis, which ***allows volatility and correlation of returns to vary*** over time.

GJR_GARCH: alpha (Coefficients for ARCH), gamma(Coefficients for Leverage effect), beta (Coefficients for GARCH)

	Bank	alpha	alphaSE	gamma	gammaSE	beta	betaSE
1	'C:US'	0.0321	0.0046	0.0938	0.0063	0.9182	0.0030
2	'WFC:US'	0.1000	0.0075	0.0249	0.0073	0.8800	0.0044
3	'BAC:US'	0.0560	0.0060	0.0925	0.0079	0.8977	0.0044
4	'JPM:US'	0.0372	0.0050	0.1141	0.0064	0.9037	0.0045
5	'BNP:FP'	0.0107	0.0045	0.1137	0.0081	0.9269	0.0041
6	'GLE:FP'	0.0365	0.0036	0.0889	0.0076	0.9152	0.0036
7	'HSBA:LN'	0.0281	0.0035	0.0371	0.0044	0.9509	0.0024
8	'BARC:LN'	0.0740	0.0030	0.0617	0.0052	0.8940	0.0042
9	'8306.T:JP'	0.0365	0.0049	0.0469	0.0071	0.9315	0.0042
10	'8411.T:JP'	0.0701	0.0055	0.0428	0.0068	0.9085	0.0040
11	'RY:CN'	0.0631	0.0073	0.0951	0.0089	0.8768	0.0066
12	'TD:CN'	0.0684	0.0074	0.0921	0.0087	0.8779	0.0060

Alexandra Wang
Wenxiao Wu
Clementine Mc Sweeny

Bank	alpha	alphaSE	gamma	gammaSE	beta	betaSE
BAC:US	0.0452	0.0128	0.0904	0.0206	0.9061	0.0198
BK:US	0.0327	0.0344	0.1337	0.0312	0.885	0.0359
C:US	0.0514	0.012	0.099	0.0186	0.8952	0.016
COF:US	0.0483	0.0194	0.0881	0.0302	0.897	0.0247
GS:US	0.0447	0.0202	0.0633	0.0261	0.9129	0.0271
JPM:US	0.037	0.013	0.1511	0.0258	0.8776	0.0222
MS:US	0.0427	0.0125	0.1011	0.0198	0.8991	0.0164
PNC:US	0.0582	0.0202	0.1807	0.0545	0.8379	0.0471
USB:US	0.0348	0.0178	0.1188	0.0209	0.9007	0.0249
WFC:US	0.0452	0.0178	0.1183	0.0322	0.8909	0.0306

Table E.1: Volatility Parameters

a) GJR GARCH

```
stats = CRISK.beta.computeDynamicBetaGJR(banks_info,bank_return)
```

stats = 12x7 table

	Bank	alpha	alphaSE	gamma	gammaSE	beta	betaSE
1	'C:US'	0.0314	0.0046	0.0940	0.0063	0.9185	0.0030
2	'WFC:US'	0.0506	0.0045	0.0706	0.0047	0.9061	0.0036
3	'BAC:US'	0.0581	0.0064	0.0969	0.0082	0.8934	0.0047
4	'JPM:US'	0.0365	0.0050	0.1144	0.0064	0.9036	0.0046
5	'BNP:FP'	0.0096	0.0046	0.1183	0.0083	0.9258	0.0041
6	'GLE:FP'	0.0368	0.0037	0.0901	0.0077	0.9138	0.0037
7	'HSBA:LN'	0.0772	0.0066	0.0388	0.0072	0.9024	0.0046
8	'BARC:LN'	0.0745	0.0031	0.0653	0.0054	0.8918	0.0043
9	'8306.T:JP'	0.0365	0.0049	0.0469	0.0071	0.9315	0.0042
10	'8411.T:JP'	0.0701	0.0055	0.0428	0.0068	0.9065	0.0040
11	'RY:CN'	0.0588	0.0073	0.0976	0.0089	0.8781	0.0066
12	'TD:CN'	0.0711	0.0076	0.0930	0.0089	0.8733	0.0062

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

b) DCC GARCH

```
do_training = false;
if do_training
    [stats_dcc,MKT,CF] = CRISK.beta.computeDynamicBetaDCC(banks_info,bank_return);
```

- alpha (Coefficients for ARCH)
- beta (Coefficients for GARCH)

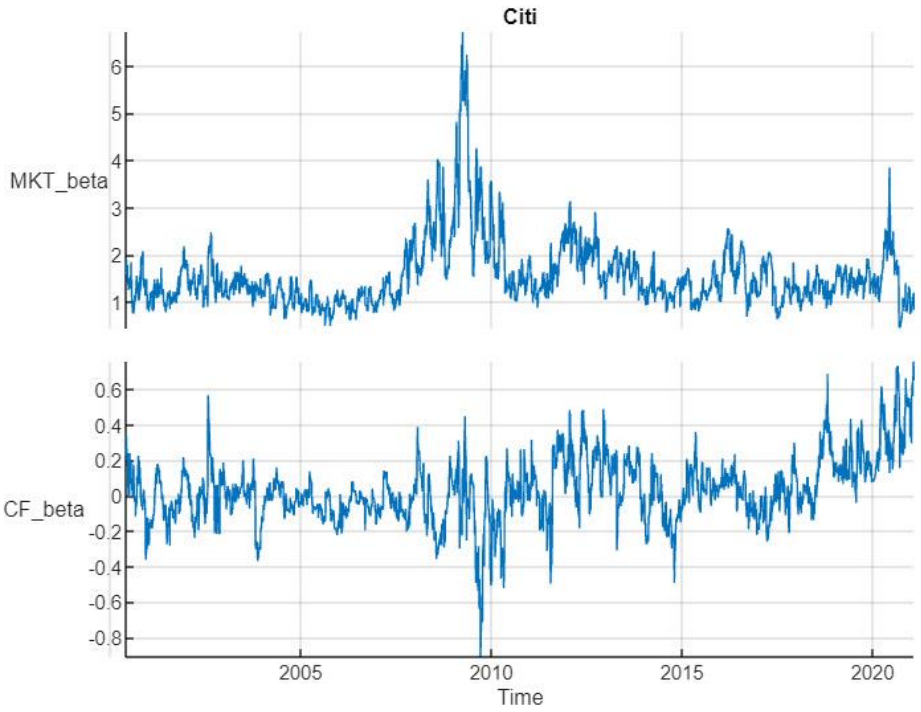
```
    stats_dcc = struct2table(stats_dcc);
    proj = currentProject;
    save(fullfile(proj.RootFolder,"data","dcc_data"),"CF","MKT","stats_dcc");
else
    proj = currentProject;
    load(fullfile(proj.RootFolder,"data","dcc_data"))
end
```

c) Plot bank market and climate beta

```
bank_index = 1;% choose the bank data you wanna plot
CRISK.beta.utils.plotMKTandCFBeta(bank_index,MKT,CF,bank_return_table);
```



Alexandra Wang
Wenxiao Wu
Clementine Mc Sweeny



4. Compute CRISK

1) Import climate beta, market beta, debt and equity

To implement CRISK, we used the quartely equity and debt values fron 2000 to 2011 found on Yahoo Finance. We used climate betas calculate in an other file.

```
equity = readtimetable('debt_equity.xlsx','Sheet','equity');
```

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

```
debt = readtimetable('debt_equity.xlsx','Sheet','debt');
tickers = ["C","WFC","BAC"];
```

2) Concatenate tables for each bank

We use the following formula to calculate CRISK:

$$\mathrm{CRISK}_{\mathrm{it}} = k \cdot D_{\mathrm{it}} - (1-k) \cdot W_{\mathrm{it}} \cdot \exp\left(\beta_{\mathrm{it}}^{\mathrm{Climate}} \log(1-\theta)\right)$$

Where:

- D : book value of debt
- W : market value of equity
- k = 8%
- θ = 50%

In the following sections, we plotted the charts we obtained for **`Citi`**, **`Well Fargo`** and **`Bank of America`**.

In this following code, we implement the above fomula for CRISK.

```
crisk = CRISK.computeCRISK(equity,debt,CF,tickers)
```

```
crisk = 1×5 struct
```

Fields	bank	data	CRISK
1	'C'	5197×3 time...	5197×1 time...
2	'WFC'	5197×3 time...	5197×1 time...
3	'BAC'	5197×3 time...	5197×1 time...
4	'year'	5197×1 time...	5197×0 time...
5	'quarter'	5197×1 time...	5197×0 time...

CRISK for **`Citi`**

```
CRISK_Citi = crisk(1).CRISK
```

```
CRISK_Citi = 5197×1 timetable
```

	Time	risk
1	01-Jun-2000	-100.8500
2	02-Jun-2000	-88.1003
3	05-Jun-2000	-95.1533

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny

	Time	risk
4	06-Jun-2000	-97.3144
5	07-Jun-2000	-98.2041
6	08-Jun-2000	-102.4971
7	09-Jun-2000	-103.1047
8	12-Jun-2000	-104.3401
9	13-Jun-2000	-104.3133
10	14-Jun-2000	-100.8866
11	15-Jun-2000	-102.9171
12	16-Jun-2000	-99.5011
13	19-Jun-2000	-122.5365
14	20-Jun-2000	-112.0939

⋮

3) Plot the results

Here are some of the results obtained by the authors:

Alexandra Wang

Wenxiao Wu

Clementine Mc Sweeny