

Un modelo affine de Stock Prices

Catalina Bustillo, Francisco Cenzón, Ceferino Pettovello,
Gonzalo Silman, Eric Tulle
Universidad Torcuato Di Tella

July 17, 2021

1 Introducción

Es sabido que los modelos que se suelen utilizar para el pricing de stocks asumen log normalidad de dividendos, ya sea comenzando por *First Principals* (donde existe un agente representativo que maximiza su utilidad) o con argumentos de no arbitraje.

El modelo estándar que observamos parte de una ecuación de no arbitraje, que determina el precio del stock en el periodo t como el valor esperado de la suma del stock y los dividendos en $t+1$ descontado a la tasa ρ . Asumiendo que los dividendos son log normales (μ , σ^2), resolvemos el problema mediante una conjetura que asume el precio del stock como una ecuación lineal de los dividendos. Obtenemos como resultado un ratio de stock sobre dividendos constante, que al depender de ρ , hace que todo sea constante. Planteando la risk premia como el holding return (retorno por holdear el asset entre t y $t+1$) menos la tasa libre de riesgo, observamos que toda la variabilidad de la prima de riesgo esta explicada por fluctuaciones en la tasa.

Este resultado es inconsistente con lo que observamos en los datos, en particular en momentos de crisis. A su vez, consideramos poco representativo con la realidad el hecho de que los agentes descuenten el valor de las acciones de manera constante a lo largo de todos los periodos.

Con el fin de resolver el problema, presentamos un modelo afin para el pricing de stocks. La metodología proviene del paper de Casassus & Collin-Dufresne (2005), con la cual derivamos un implicit convenience yield que justifica un spot price afin.

Suponemos que el factor de descuento ρ tiene una forma funcional afin. Así, representamos mediante un Kernel el factor de descuento en base a los factores de riesgo. Lo que terminamos teniendo es un factor de descuento que modela la yield curve.

2 Modelo de determinación del precio de un stock

A continuación, presentamos un modelo estandar de determinación de stock pricing basado en argumentos de no arbitraje donde asumimos que la tasa de descuento es constante. Como mencionamos con anterioridad los únicos valores fundamentales en esta economía son los dividendos, donde se asume que el logaritmo de los mismos sigue un camino aleatorio.

La ecuación de no arbitraje que determina el precio del stock es:

$$S_t = E_t [e^{-\rho} (S_{t+1} + D_{t+1})], \quad (1)$$

donde S_t representa el precio del stock y es igual al valor descontado del stock futuro, mas el pago de dividendos.

El logaritmo de dividendos sigue un camino aleatorio

$$\ln(D_t) = \mu + \ln(D_{t-1}) + \varepsilon_t \quad (2)$$

Para resolver el problema de determinación de precio planteado en la ecuación (1), se propone la siguiente conjetura

$$S_t = K D_t.$$

Sustituyendo la conjetura en la ecuación (1) encontramos que

$$K D_t = E_t [e^{-\rho} (K D_{t+1} + D_{t+1})], \quad (3)$$

y usando las propiedades de log-normalidad obtenemos que

$$K D_t = e^{-\rho} (1 + K) e^{\mu + \frac{\sigma^2}{2}} D_t. \quad (4)$$

Esto nos permite obtener la siguiente ecuación

$$K = \frac{e^{\mu + \frac{\sigma^2}{2}}}{e^{\rho} - e^{\mu + \frac{\sigma^2}{2}}}, \quad (5)$$

lo cual implica que el ratio entre stocks y dividendos tiene la siguiente forma

$$\frac{S_t}{D_t} = \frac{e^{\mu + \frac{\sigma^2}{2}}}{e^{\rho} - e^{\mu + \frac{\sigma^2}{2}}}$$

Esta expresión que depende de μ , σ^2 y ρ es constante lo que está evidentemente en contra de los hechos estilizados.

En la figura (1) mostramos la evolución histórica de $\frac{S_t}{D_t}$ de 1908 a 2018.

Risk Premia

Otra propiedad no deseable de este modelo es que el “Holding Return” es constante (es relativamente fácil demostrarlo).

En la siguiente ecuación derivamos el Risk Premia para el modelo estandar,

$$E_t \frac{(S_{t+1} + D_{t+1}) - S_t}{S_t} - r_t = \frac{(1 + K)e^{\mu + \frac{\sigma^2}{2}}}{K} - 1 - r_t.$$

Por lo que el Risk Premia está conformado por una constante que llamamos el holding return $(\frac{(1+K)e^{\mu + \frac{\sigma^2}{2}}}{K} - 1)$ a la cual se le resta la tasa de interés (r_t) que representa el costo de **poner** la inversión en el Stock. De aquí podemos observar que la evolución del Risk Premia depende simplemente de las variaciones que surgen de la tasa de interés.

Eso lo hace poco atractivo porque tiene poco contenido informativo.

3 Un Modelo Afin de determinación de precios del stock

Como avanzamos en la sección anterior, el modelo que presenta como fundamentos de la determinación del stock falla en replicar los datos empíricos. Una de las razones por lo que esto sucede es que el descuento se toma como una constante, siendo que dicha tasa varía sustancialmente en el tiempo.

Las caídas del precio del stock pueden ser el resultado de una baja en los dividendos futuros o de una suba en la tasa de descuento.

Por lo tanto presentaremos un modelo donde los precios reaccionan a ambas variables.

A continuación presentamos la metodología de Casassus and Collin-Dufresne (2005) en derivar una *implicit convenience yield* en el precio spot. Si bien Casassus y Collin-Dufresne lo utilizan para derivar el precio de un futuro, nosotros usamos la estructura para avanzar en la determinación del precio spot.

3.1 Estructura del modelo

Asumimos que los factores de riesgo siguen un proceso autoregresivo de primer orden en el vector x_t .

$$x_{t+1} = \mu + \Phi x_t + \Gamma \eta_{t+1} \tag{6}$$

donde $E[\eta_{t+1}\eta'_{t+1}] = I$.

Es importante notar que la determinación de cuales son los factores de riesgo relevantes es parte del ejercicio, en nuestro análisis asumimos que x_t está compuesto por log de dividendos y la tasa real de interés, ya que movimientos en esta tasa pueden ser responsables de la variación del ratio de precios-dividendos.

Se asume que la tasa de corto plazo es una función afin de los factores de riesgo,

$$r_t = \delta_0 + \delta'_1 x_t.$$

Para el factor de descuento, utilizaremos el siguiente Kernel que es tradicional en la literatura. El mismo es función de la tasa de corto plazo y los factores de riesgo relevantes de la economía.

$$M_{t+1} = \exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t - \lambda_t' \eta_{t+1} \right) \quad (7)$$

Es importante resaltar que r_t es conocido a tiempo t , y que el riesgo se captura en el factor de descuento por la contribución de los shocks a las variables estado. Explícitamente, al elegir las variables estado estamos eligiendo cuales son los riesgos relevantes que afronta el individuo.

El modelo asume que el precio del riesgo es afin y, que a su vez, es una función de los factores de riesgo

$$\lambda_t = \lambda_0 + \Lambda_1 x_t$$

donde λ_0 es un vector de 2×1 y λ_1 una matriz de 2×2 .

Se propone que el logaritmo del stock es afin,

$$\log S_t = \gamma_0 + \gamma_1' x_t, \quad (8)$$

Y se propone lo mismo sobre el convenience yield:

$$\rho_t = \psi_0 + \psi_1' x_t. \quad (9)$$

donde $\gamma_0, \gamma_1', \psi_0$ y ψ_1' son parámetros a determinar dada la estructura del modelo.

3.2 Evaluación del Precio del Stock

En el presente modelo la condición de no arbitraje implica que el precio del stock satisface:

$$S_t = E_t [M_{t+1} (S_{t+1} + D_{t+1})]. \quad (10)$$

donde ahora la tasa de descuento es una variable aleatoria M_t y D_t es el dividendo. Podemos escribir esta ecuación en términos de la “convenience yield neta” de la siguiente manera:

$$\begin{aligned} S_t &= E_t [M_{t+1} (S_{t+1} + D_{t+1})] \\ &= E_t \left[M_{t+1} \left(1 + \frac{D_{t+1}}{S_{t+1}} \right) S_{t+1} \right] \\ &\equiv E_t [M_{t+1} \exp(\rho_{t+1}) S_{t+1}] \end{aligned}$$

donde $1 + \frac{D_{t+1}}{S_{t+1}} \approx \exp \rho_{t+1}$, y ρ_{t+1} es la expresión afin antes descrita.

Ahora hallamos las restricciones que impone el modelo sobre los parámetros sustituyendo las expresiones de precios y de convenience yield para que la ecuación de pricing previa sea consistente con el spot price de

equilibrio que satisface (8). La ecuación de Stock pricing queda de la siguiente manera:

$$\begin{aligned}\exp(\gamma_0 + \gamma_1' x_t) &= E_t \left[\exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t - \lambda_t' \eta_{t+1} \right) \exp(\rho_{t+1}) \exp(\gamma_0 + \gamma_1' x_{t+1}) \right] \\ &= \exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t + \psi_0 + \gamma_0 + (\psi_1 + \gamma_1)' (\mu + \Phi x_t) \right) E_t \left[\exp \left([\Gamma' (\psi_1 + \gamma_1) - \lambda_t]' \eta_{t+1} \right) \right]\end{aligned}$$

Notamos que el único componente aleatorio es η_{t+1} , que es log normal, por lo que resolviendo la expresión del valor esperado simplemente tendremos que igualar coeficientes en esa ecuación.

Por lo tanto, considerando este término obtenemos,

$$\begin{aligned}E_t \left[\exp \left([\Gamma' (\psi_1 + \gamma_1) - \lambda_t]' \eta_{t+1} \right) \right] &= \exp \left(\frac{1}{2} [\Gamma' (\psi_1 + \gamma_1) - \lambda_t]' [\Gamma' (\psi_1 + \gamma_1) - \lambda_t] \right) \\ &= \exp \left(\frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1) - (\psi_1 + \gamma_1)' \Gamma \lambda_t + \frac{1}{2} \lambda_t' \lambda_t \right)\end{aligned}$$

Sustituyendo nos queda:

$$\begin{aligned}\exp(\gamma_0 + \gamma_1' x_t) &= \exp \left(\begin{array}{c} -r_t - \frac{1}{2} \lambda_t' \lambda_t + \psi_0 + \gamma_0 + (\psi_1 + \gamma_1)' (\mu + \Phi x_t) \\ + \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1) - (\psi_1 + \gamma_1)' \Gamma \lambda_t + \frac{1}{2} \lambda_t' \lambda_t \end{array} \right) \\ &= \exp \left(\begin{array}{c} -\delta_0 + \psi_0 + \gamma_0 + (\psi_1 + \gamma_1)' (\mu - \Gamma \lambda_0) + \frac{(\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)}{2} \\ + [(\psi_1 + \gamma_1)' (\Phi - \Gamma \Lambda_1) - \delta_1'] x_t \end{array} \right)\end{aligned}$$

Tomando logaritmo e igualando coeficientes por las variables que multiplican a x_t :

$$\gamma_1' = (\psi_1 + \gamma_1)' (\Phi - \Gamma \Lambda_1) - \delta_1',$$

$$\psi_1 + \gamma_1 = [(\Phi - \Gamma \Lambda_1)']^{-1} (\gamma_1 + \delta_1),$$

o

$$\psi_1 = [(\Phi - \Gamma \Lambda_1)']^{-1} (\gamma_1 + \delta_1) - \gamma_1. \quad (11)$$

Definiendo ϕ^* como la matriz ajustada por el riesgo,

$$\Phi^* = \Phi - \Gamma \Lambda_1,$$

podemos encontrar ψ , como una función de ϕ^* , γ_1 y δ_1

$$\gamma_1' = (\psi_1 + \gamma_1)' (\Phi^*) - \delta_1'$$

$$\psi_1 + \gamma_1 = [(\Phi^*)']^{-1} (\gamma_1 + \delta_1)$$

o

$$\psi_1 = [(\Phi^*)']^{-1} (\gamma_1 + \delta_1) - \gamma_1 \quad (12)$$

Esta expresión relaciona las expresiones que entran en la solución afin del convenience yield y del logaritmo del Spot Price.

Análogamente, para la constante:

$$\delta_0 = \psi_0 + (\psi_1 + \gamma_1)' (\mu - \Gamma \lambda_0) + \frac{(\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)}{2}$$

o

$$\psi_0 = \delta_0 - (\psi_1 + \gamma_1)' (\mu - \Gamma \lambda_0) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1). \quad (13)$$

Si definimos el drift neutral al riesgo μ^* ,

$$\mu^* = \mu - \Gamma \lambda_0,$$

Podemos operar para encontrar la ecuación relevante para ψ_0 .

$$\delta_0 = \psi_0 + (\psi_1 + \gamma_1)' (\mu^*) + \frac{(\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)}{2}$$

o

$$\psi_0 = \delta_0 - (\psi_1 + \gamma_1)' (\mu^*) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1). \quad (14)$$

Si $\delta_0 = 0$ y $\delta_1 = J = [0, 1]'$

$$\psi_1 = [(\Phi^*)']^{-1} (\gamma_1 + J) - \gamma_1 \quad (15)$$

$$\gamma_1' = (\psi_1 + \gamma_1)' (\Phi^*) - J'$$

$$\psi_0 = -(\psi_1 + \gamma_1)' (\mu^*) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1). \quad (16)$$

Esto nos muestra las restricciones que tienen que tener los parámetros para la estimación del modelo conceptual:

3.3 Estimación del modelo

El modelo que vamos a estimar consta de 3 ecuaciones (17), (18) y (19). La primera ecuación, como adelantamos, es la de la evolución de los factores de riesgo que asumimos ser el logaritmo de dividendos y la tasa real de interés. Las otras ecuaciones son las de determinación del precio del stock y del convenience yield, ambas aumentadas por los errores de medición que están incorrelacionados entre sí e incorrelacionados con el stock en los factores de riesgo.

Estas ecuaciones están estimadas conjuntamente por máxima verosimilitud, donde la estructura del modelo impone restricciones entre los parámetros de las ecuaciones. Por lo que varios parámetros del modelo no son libres y están sujetos a las restricciones antes descritas.

$$x_{t+1} = \mu + \Phi x_t + \Gamma \eta_{t+1} \quad (17)$$

$$\log S_t = \gamma_0 + \gamma'_1 x_t + u_{1t} \quad (18)$$

$$\ln\left(1 + \frac{D_t}{S_t}\right) = \psi_0 + \psi'_1 x_t + u_{2t} \quad (19)$$

Dado que ψ_1 y Φ^* son parámetros libres, derivamos las expresiones para γ_1 y ψ_0 ,

$$\begin{aligned} \gamma'_1 &= (\psi_1 + \gamma_1)' \Phi^* - J' \\ \gamma'_1 (I - \Phi^*) &= \psi'_1 \Phi^* - J' \\ \gamma'_1 &= (\psi'_1 \Phi^* - J') (I - \Phi^*)^{-1}, \end{aligned}$$

$$\psi_0 = -(\psi_1 + \gamma_1)' (\mu^*) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1). \quad (20)$$

Adicionalmente, para que el modelo esté identificado, imponemos que ϕ^* es diagonal y μ^* tiene un solo valor distinto de cero.

Risk Premia:

A continuación determinamos la prima de riesgo. A diferencia del modleo anterior, el holding return varía en el tiempo.

Computamos el log-holding return esperado para los stock prices en términos de una estructura afin y notamos que el (log) return esperado a tiempo t de tener un Stock por un período, dada la estructura del modelo, está dada por:

$$\begin{aligned} E_t \left(\log \frac{S_{t+1} + D_{t+1}}{S_t} \right) &= E_t \log(\exp(\rho_{t+1}) S_{t+1}) - \log S_t \\ &= E_t(\psi_0 + \psi'_1 x_{t+1} + \gamma_0 + \gamma'_1 x_{t+1}) - \gamma_0 + \gamma'_1 x_t \end{aligned}$$

Y sustituyendo por la expresión que determina $\log(S_t)$ encontramos que:

$$\begin{aligned} E_t \log(S_{t+1} + D_{t+1}) - \log S_t &= E_t(\psi_0 + \psi'_1 x_{t+1} + \gamma_0 + \gamma'_1 x_{t+1}) - (\gamma_0 + \gamma'_1 x_t) \\ &= \psi_0 + (\psi'_1 + \gamma'_1) E_t [x_{t+1}] - \gamma'_1 x_t \end{aligned}$$

Ahora, para determinar el Risk Premia (exceso de retorno de mantener un bono sobre la tasa libre de riesgo), simplemente utilizamos que $r_t = \delta_0 + \delta'_1 x_t$ y operamos para obtener:

$$\begin{aligned}
E_t \log(S_{t+1} + D_{t+1}) - \log S_t - r_t &= E_t(\psi_0 + \psi'_1 x_{t+1} + \gamma_0 + \gamma'_1 x_{t+1}) - (\gamma_0 + \delta_0 + (\gamma'_1 + \delta'_1) x_t) \\
&= \psi_0 - \delta_0 + (\psi'_1 + \gamma'_1) \mu + (\psi'_1 + \gamma'_1) \Phi x_t - (\gamma'_1 + \delta'_1) x_t
\end{aligned}$$

Por último, usando que,

$$(\gamma_1 + \delta_1)' = (\psi_1 + \gamma_1)' \Phi^*,$$

$$\begin{aligned}
E_t \log(S_{t+1} + D_{t+1}) - \log S_t - r_t &= \psi_0 - \delta_0 + (\psi'_1 + \gamma'_1) \mu + (\psi'_1 + \gamma'_1) \Phi x_t - (\psi_1 + \gamma_1)' \Phi^* x_t \\
&\quad \psi_0 - \delta_0 + (\psi'_1 + \gamma'_1) \mu + (\psi'_1 + \gamma'_1) (\Phi - \Phi^*) x_t
\end{aligned}$$

y utilizando que, $\psi_0 = \delta_0 - (\psi_1 + \gamma_1)' (\mu^*) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)$

donde,

$$\begin{aligned}
\lambda_t &= \lambda_0 + \Lambda_1 x_t \\
\lambda_0 &= \Gamma^{-1} (\mu - \mu^*) \\
\Lambda_1 &= \Gamma^{-1} (\Phi - \Phi^*)
\end{aligned}$$

$$\begin{aligned}
E_t \log(S_{t+1} + D_{t+1}) - \log S_t - r_t &= -\frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1) + (\psi'_1 + \gamma'_1) (\mu - \mu^*) + (\psi'_1 + \gamma'_1) (\Phi - \Phi^*) x_t \\
&= -\frac{1}{2 (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1) + (\psi'_1 + \gamma'_1) \Gamma \lambda_t}
\end{aligned}$$

Esta última expresión nos muestra que el Risk Premia es por construcción una función de los factores de riesgo y se mueve con ellos.

En la parte experimental del trabajo evaluaremos mediante su comparación con variables de la economía cuán razonable son las primas presentadas por los modelos y si hay evidencia a favor de alguna de las mismas.

4 Análisis empírico

En primer lugar, tenemos los gráficos que muestran el modelo estándar y su comparación con los datos. El primer plot nos muestra en rojo la serie de dividendos multiplicada por el factor constante K derivado en la ecuación (5). Como podemos ver sobrevalúa el precio del stock hasta mediados de los 90' y los subvalúa en el período siguiente.

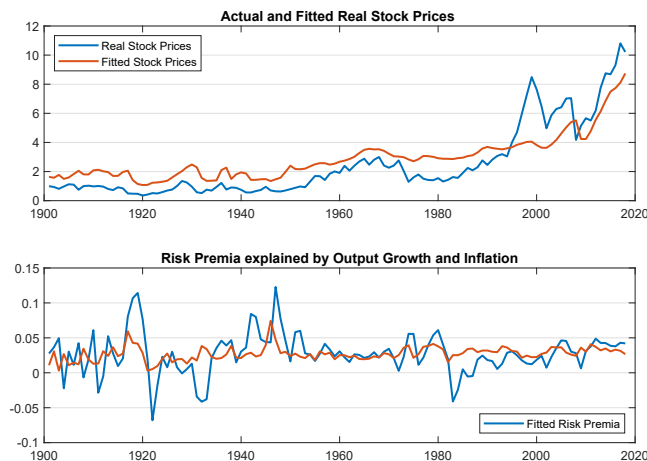


Figure 1: Modelo Estándar

El segundo panel de la figura 1 muestra la prima de riesgo derivada del modelo, cuya variación es heredada simplemente de los movimientos de la tasa de interés, y a su vez muestra la curva predicha de regresar dicha prima contra el spread de tasas (10 y 1 años) rezagadas y crecimiento del producto. Si miramos el R^2 de este modelo, que mide cuánto explican las variables independientes la variabilidad de la variable dependiente, vemos que nos da 0.1181.

A continuación presentamos los resultados obtenidos tanto en bondad de ajuste como en la evolución de la prima de riesgo del modelo afin. La figura 2 consta de 3 paneles pues también mostramos la evolución del convenience yield.

El risk premia no está simplemente explicado por la tasa de interés sino que se mueve por otros factores de riesgo. En nuestro caso, los dividendos futuros y la convenience yield.

En este caso, el R^2 nos da un valor de 0.2025.

El primer panel muestra la evolución del precio del stock y aquella predicha por el modelo donde encontramos que no incurre en errores sistematicos y solo falla drasticamente ne los finales de los 90. Algunos autores atribuyen dichos errores a burbujas especulativas que no han sido modeladas.

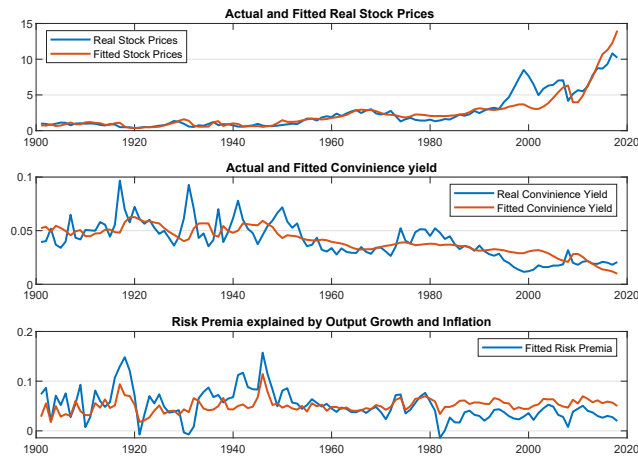


Figure 2: Modelo Afin

Con respecto al convenience yield vemos que el modelo muestra una forma suavizada del mismo.

Finalmente mostramos la evolución de la prima de riesgo predicha por el modelo conjuntamente con el predicho de regresar dicha prima contra el spread rezagado y la tasa de crecimiento.

5 Apéndice

5.1

$$\begin{aligned}
\exp(\gamma_0 + \gamma_1' x_t) &= E_t \left[\exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t - \lambda_t' \eta_{t+1} \right) \exp(\rho_{t+1}) \exp(\gamma_0 + \gamma_1' x_{t+1}) \right] \\
&= E_t \left[\exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t - \lambda_t' \eta_{t+1} + \rho_{t+1} + \gamma_0 + \gamma_1' x_{t+1} \right) \right] \\
&= E_t \left[\exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t - \lambda_t' \eta_{t+1} + \psi_0 + \psi_1' x_{t+1} + \gamma_0 + \gamma_1' x_{t+1} \right) \right] \\
&= \exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t + \psi_0 + \gamma_0 \right) E_t \left[\exp \left(-\lambda_t' \eta_{t+1} + (\psi_1 + \gamma_1)' x_{t+1} \right) \right] \\
&= \exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t + \psi_0 + \gamma_0 \right) E_t \left[\exp \left(-\lambda_t' \eta_{t+1} + (\psi_1 + \gamma_1)' (\mu + \Phi x_t + \Gamma \eta_{t+1}) \right) \right] \\
&= \exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t + \psi_0 + \gamma_0 + (\psi_1 + \gamma_1)' (\mu + \Phi x_t) \right) E_t \left[\exp \left(-\lambda_t' \eta_{t+1} + (\psi_1 + \gamma_1)' \Gamma \eta_{t+1} \right) \right] \\
&= \exp \left(-r_t - \frac{1}{2} \lambda_t' \lambda_t + \psi_0 + \gamma_0 + (\psi_1 + \gamma_1)' (\mu + \Phi x_t) \right) E_t \left[\exp \left([\Gamma' (\psi_1 + \gamma_1) - \lambda_t]' \eta_{t+1} \right) \right]
\end{aligned}$$

Por lo tanto,

$$\begin{aligned}
E_t \left[\exp \left([\Gamma' (\psi_1 + \gamma_1) - \lambda_t]' \eta_{t+1} \right) \right] &= \exp \left(\frac{1}{2} [\Gamma' (\psi_1 + \gamma_1) - \lambda_t]' [\Gamma' (\psi_1 + \gamma_1) - \lambda_t] \right) \\
&= \exp \left(\frac{1}{2} [(\psi_1 + \gamma_1)' \Gamma - \lambda_t]' [\Gamma' (\psi_1 + \gamma_1) - \lambda_t] \right) \\
&= \exp \left(\frac{1}{2} [(\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1) - \lambda_t' \Gamma' (\psi_1 + \gamma_1) - (\psi_1 + \gamma_1)' \Gamma \lambda_t + \lambda_t' \lambda_t] \right) \\
&= \exp \left(\frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1) - (\psi_1 + \gamma_1)' \Gamma \lambda_t + \frac{1}{2} \lambda_t' \lambda_t \right)
\end{aligned}$$

Susituyendo nos queda:

$$\begin{aligned}
\exp(\gamma_0 + \gamma_1' x_t) &= \exp \left(\begin{array}{c} -r_t - \frac{1}{2} \lambda_t' \lambda_t + \psi_0 + \gamma_0 + (\psi_1 + \gamma_1)' (\mu + \Phi x_t) \\ + \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1) - (\psi_1 + \gamma_1)' \Gamma \lambda_t + \frac{1}{2} \lambda_t' \lambda_t \end{array} \right) \\
&= \exp \left(-r_t + \psi_0 + \gamma_0 + (\psi_1 + \gamma_1)' (\mu + \Phi x_t) + \frac{(\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)}{2} - (\psi_1 + \gamma_1)' \Gamma \lambda_t \right) \\
&= \exp \left(\begin{array}{c} -\delta_0 - \delta_1' x_t + \psi_0 + \gamma_0 + (\psi_1 + \gamma_1)' \mu + (\psi_1 + \gamma_1)' \Phi x_t \\ + \frac{(\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)}{2} - (\psi_1 + \gamma_1)' \Gamma \lambda_0 - (\psi_1 + \gamma_1)' \Gamma \Lambda_1 x_t \end{array} \right) \\
&= \exp \left(\begin{array}{c} -\delta_0 + \psi_0 + \gamma_0 + (\psi_1 + \gamma_1)' (\mu - \Gamma \lambda_0) + \frac{(\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)}{2} \\ + [(\psi_1 + \gamma_1)' (\Phi - \Gamma \Lambda_1) - \delta_1'] x_t \end{array} \right)
\end{aligned}$$

Tomando logaritmo e igualando coeficientes por las variables que multiplican a x_t :

$$\gamma_1' = (\psi_1 + \gamma_1)' (\Phi - \Gamma \Lambda_1) - \delta_1'$$

$$(\gamma_1 + \delta_1)' = (\psi_1 + \gamma_1)' (\Phi - \Gamma \Lambda_1)$$

$$\begin{aligned}\gamma_1 + \delta_1 &= (\Phi - \Gamma\Lambda_1)' (\psi_1 + \gamma_1) \\ \psi_1 + \gamma_1 &= [(\Phi - \Gamma\Lambda_1)']^{-1} (\gamma_1 + \delta_1)\end{aligned}$$

o

$$\psi_1 = [(\Phi - \Gamma\Lambda_1)']^{-1} (\gamma_1 + \delta_1) - \gamma_1 \quad (21)$$

Definimos ϕ^* como la matriz ajustada por el riesgo,

$$\Phi^* = \Phi - \Gamma\Lambda_1$$

Sustituyendo nos queda:

$$\begin{aligned}\gamma_1' &= (\psi_1 + \gamma_1)' (\Phi^*) - \delta_1' \\ (\gamma_1 + \delta_1)' &= (\psi_1 + \gamma_1)' (\Phi^*) \\ \gamma_1 + \delta_1 &= (\Phi^*)' (\psi_1 + \gamma_1) \\ \psi_1 + \gamma_1 &= [(\Phi^*)']^{-1} (\gamma_1 + \delta_1)\end{aligned}$$

o

$$\psi_1 = [(\Phi^*)']^{-1} (\gamma_1 + \delta_1) - \gamma_1 \quad (22)$$

Análogamente, para la constante:

$$\delta_0 = \psi_0 + (\psi_1 + \gamma_1)' (\mu - \Gamma\lambda_0) + \frac{(\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)}{2}$$

o

$$\psi_0 = \delta_0 - (\psi_1 + \gamma_1)' (\mu - \Gamma\lambda_0) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1). \quad (23)$$

Si definimos el drift neutral al riesgo μ^* ,

$$\mu^* = \mu - \Gamma\lambda_0,$$

Podemos operar para encontrar la ecuación relevante para ϕ_0 .

$$\delta_0 = \psi_0 + (\psi_1 + \gamma_1)' (\mu^*) + \frac{(\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)}{2}$$

o

$$\psi_0 = \delta_0 - (\psi_1 + \gamma_1)' (\mu^*) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1). \quad (24)$$

Si $\delta_0 = 0$ y $\delta_1 = J = [0, 1]'$

$$\psi_1 = [(\Phi^*)']^{-1} (\gamma_1 + J) - \gamma_1 \quad (25)$$

$$\gamma_1' = (\psi_1 + \gamma_1)' (\Phi^*) - J'$$

$$\psi_0 = -(\psi_1 + \gamma_1)' (\mu^*) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1). \quad (26)$$

5.2

$$x_{t+1} = \mu + \Phi x_t + \Gamma \eta_{t+1} \quad (27)$$

$$\log S_t = \gamma_0 + \gamma_1' x_t + u_{1t} \quad (28)$$

$$\ln\left(1 + \frac{D_t}{S_t}\right) = \psi_0 + \psi_1' x_t + u_{2t} \quad (29)$$

Dado que ψ_1 y Φ^* son parámetros libres, derivamos la expresión para γ_1' y ψ_0 .

$$\begin{aligned} \gamma_1' &= (\psi_1 + \gamma_1)' \Phi^* - J' \\ \gamma_1' (I - \Phi^*) &= \psi_1' \Phi^* - J' \\ \gamma_1' &= (\psi_1' \Phi^* - J') (I - \Phi^*)^{-1} \end{aligned}$$

$$\psi_0 = -(\psi_1 + \gamma_1)' (\mu^*) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1). \quad (30)$$

Log-holding return

$$\begin{aligned} E_t \left(\log \frac{S_{t+1} + D_{t+1}}{S_t} \right) &= E_t \log(\exp(\rho_{t+1}) S_{t+1}) - \log S_t \\ &= E_t (\log(\exp(\rho_{t+1})) + \log(S_{t+1})) - \log S_t \\ &= E_t (\psi_0 + \psi_1' x_{t+1} + \gamma_0 + \gamma_1' x_{t+1}) - \gamma_0 + \gamma_1' x_t \end{aligned}$$

Y sustituyendo por la expresión que determina $\log(S_t)$ encontramos que:

$$\begin{aligned} E_t \log(S_{t+1} + D_{t+1}) - \log S_t &= E_t (\psi_0 + \psi_1' x_{t+1} + \gamma_0 + \gamma_1' x_{t+1}) - (\gamma_0 + \gamma_1' x_t) \\ &= \psi_0 + (\psi_1' + \gamma_1') E_t [x_{t+1}] - \gamma_1' x_t \end{aligned}$$

Utilizamos que $r_t = \delta_0 + \delta_1' x_t$ y operamos para obtener:

$$\begin{aligned}
E_t \log(S_{t+1} + D_{t+1}) - \log S_t - r_t &= E_t(\psi_0 + \psi'_1 x_{t+1} + \gamma_0 + \gamma'_1 x_{t+1}) - (\gamma_0 + \delta_0 + (\gamma'_1 + \delta'_1) x_t) \\
&= \psi_0 - \delta_0 + (\psi'_1 + \gamma'_1) E_t[x_{t+1}] - (\gamma'_1 + \delta'_1) x_t \\
&= \psi_0 - \delta_0 + (\psi'_1 + \gamma'_1) \mu + (\psi'_1 + \gamma'_1) \Phi x_t - (\gamma'_1 + \delta'_1) x_t
\end{aligned}$$

Por último, usando que:

$$(\gamma_1 + \delta_1)' = (\psi_1 + \gamma_1)' \Phi^*$$

$$\begin{aligned}
E_t \log(S_{t+1} + D_{t+1}) - \log S_t - r_t &= \psi_0 - \delta_0 + (\psi'_1 + \gamma'_1) \mu + (\psi'_1 + \gamma'_1) \Phi x_t - (\psi_1 + \gamma_1)' \Phi^* x_t \\
&\quad \psi_0 - \delta_0 + (\psi'_1 + \gamma'_1) \mu + (\psi'_1 + \gamma'_1) (\Phi - \Phi^*) x_t
\end{aligned}$$

y utilizando que,

$$\psi_0 = \delta_0 - (\psi_1 + \gamma_1)' (\mu^*) - \frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1)$$

donde,

$$\begin{aligned}
\lambda_t &= \lambda_0 + \Lambda_1 x_t \\
\lambda_0 &= \Gamma^{-1} (\mu - \mu^*) \\
\Lambda_1 &= \Gamma^{-1} (\Phi - \Phi^*)
\end{aligned}$$

$$\begin{aligned}
E_t \log(S_{t+1} + D_{t+1}) - \log S_t - r_t &= -\frac{1}{2} (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1) + (\psi'_1 + \gamma'_1) (\mu - \mu^*) + (\psi'_1 + \gamma'_1) (\Phi - \Phi^*) x_t \\
&= \frac{1}{2 (\psi_1 + \gamma_1)' \Gamma \Gamma' (\psi_1 + \gamma_1) + (\psi'_1 + \gamma'_1) \Gamma \lambda_t}
\end{aligned}$$

5.3

MODELO ESTÁNDAR

```

1 % THIS PROGRAM ESTIMATES THE YIELD CURVE USING THE DYNAMIC NELSON AND
2 % SIEGEL AND IMPOSING NON-ARBITRAGE RESTRICTIONS. NON-DIAGONAL MODEL.
3 %
4 % The program creates a file "DNS_YieldCurve_NoArbitrage.mat" with the
5 % inputs to be used by the Futures block of the estimation
6
7 clear all; close all

```

```

8 tic
9
10 % Estimate parameters?
11 info.estimate = 1; % >0 yes; =0 no
12 info.initval = 1; % >0 if use old estimate as initial guess, =0 otherwise
13 info.start = 1; % First observation to compute likelihood
14
15 % Optimization method:
16 info.do_anneal = 1; % Do simulated annealing; >0 if yes, =0 if not
17 info.do_simplex = 0; % Do Nelder-Meade algorithm? >0 if yes, =0 if no
18 info.do_qnewton = 0; %Do quasi-newton optimization?; >0 if yes, =0 if no
19
20 info.opt_type = 3; % Type of quasi-newton algorithm: =1 uses fminunc; =2 uses
    fminsearch; =3 uses uncmin; =4 uses csminwel; 5 fmincon
21
22
23 info.do_stderr = 1; % >0 if do asymptotic standard errors; =0 if no
24 info.sderr_type = 4; % Type of standard errors to compute:
25 % = 1 use fdhess.m with forward difference
26 % = 2 use fdhess.m with centered difference
27 % = 3 use fdhess2.m (from gauss) with forward difference
28 % = 4 use outer product of gradient
29
30
31
32
33
34
35 info.plotfigs = 1; % >0 if plot figures, =0 if not.
36
37 % Choose sample size: fullsample or shortsample. Short sample is used to
38 % initialize the recursive estimation to perform the forecast exercise.
39 info.sample = 1; % >0 is full sample; =0 is short sample
40
41 % INFORMATION FOR OPTIMIZERS:
42 % Options for simulated annealing:
43 info.options_sa.Verbosity = 2; % Show intermediate information
44 info.options_sa.InitTemp = 1; % Initial temperature
45 info.options_sa.StopTemp = 1e-15; % Stop temperature criterium
46
47 % Options for fminunc:
48 info.option_fmunc = optimset('Largescale','off','Display','iter','MaxFunEvals',5000,
    ...
49 'Maxiter',2000,'TolX',1e-10,'TolFun',1e-10,'Diagnostics'
    , 'on');

```

```

50 % Options for fminsearch:
51 info.options_fms = optimset('Display','iter','MaxFunEvals',5000,'Maxiter',5000);
52
53 % -----
54 fprintf('\n Date and Hour: %s.\n ', datestr(now))
55 fprintf('\n ESTIMATING Present value an Affine Model WITH NON-ARBITRAGE
    RESTRICTIONS \n ')
56 fprintf('
    ----- \n '
    )
57
58 % Load data
59 %load interest_rates_data; % These rates are annualized and in percentage terms
60
61
62 load DATASP; % Int10Y Int1Y CONVYIELD REALDIV REALINT REALSP
63 load DATA2.txt; % Slope and Growth
64
65 yy = DATASP;
66 yy(:,1) = yy(:,1)/100; %pongo en tasa
67 yy(:,2) = yy(:,2)/100; %pongo en tasa
68
69
70 if info.sample > 0 % Use all data
71     namefile = 'SPPRICES1_NoArbitrage.mat';
72     % Set initial and final dates of sample: (T=length(datosinf) is full sample:
73     T0 = 1;
74     T = length(DATASP);
75 else
76     namefile = 'SPPRICES1_NoArbitrage.mat';
77     T0 = 1;
78     Tend = 459; % This is the last day for which we have data on future prices
79     T = Tend-12*4; % To have 4 years to perform the forecast exercise
80
81 end
82
83 yyy = [ DATA2(1:T-1,1) DATA2(2:T,2) ]; % Usar el slope lagged y growth
84
85
86
87 t0 = 1900; % Initial date
88 t1 = 2018; % Final date
89 time = ( t0 : 1 : t1 )';
90
91

```



```

92 info.nobs = T-T0+1; % Number of observations
93 info.t0   = t0;    % Initial observation
94 info.t1   = t1;    % Last observation
95
96 fprintf('\n Information of this run: \n ')
97 disp(info)
98
99 XX  = [ log(yy(:,4)) yy(:,5)]' ; % Log(dividends and real interest rate
100 SP  = yy(:,6); % Real stock prices
101 DY  = yy(:,3); % 1Convinience yield
102
103
104
105 % Initialize parameters:
106 if info.initval > 0 % check if initialize using old estimate (must be in file '
    estimates_noswitch.mat'
107
108     if exist(namefile,'file')==2 % check if file exists
109         load(namefile);
110     else
111         fprintf('File %s does not exist. Set info.initval = 0.',namefile);
112         error(' ')
113     end
114
115 else
116
117
118 %% Estimate parameters of observable factors running a VAR to have a good initial
    guess
119
120 % Now run a VAR using level, slope, curvature, and the 1-year rate;
121
122 VARoutput = var_ls( XX, 1 ); % VAR with 1 lag
123
124 % Intercept and coefficient of the VAR
125 phi_guess= VARoutput.coef{1};
126 MU_guess = VARoutput.const;
127 % Estimate of cholesky decomposition of covariance matrix
128 Gamma_guess = chol(VARoutput.Omega,'lower');
129
130 param = zeros(17,1);
131
132 % Parameter phiSTAR
133 param(1:2) = [ 1.0147 6.0970 ];
134

```

```

135 % Parameter mustar
136 param(3) = -0.01;
137
138 % Parameter PSI1:
139 param(4:5) = [-0.025175 -0.023180]'; % PSI of the convenience yield
140
141
142 % Parameter gama0:
143 param(6) = 0.5313597;
144
145 % Constants state variables:
146 param(7:8) = [0.002981 0.003131]';
147
148 % Matrix coefficients on lagged values (VAR for state variables - Non-diagonal
149 matrix)
150 param(9:12) = [ 0.996534 0.121849 -0.001120 0.555087 ];
151
152 % Matrix with lower triangular Cholesky decomposition of covar matrix
153 param(13:14) = [ 10 5 ]';
154 param(15) = 1;
155
156 % Square root of variances of measurement errors
157 param(16:17) = [10 10]';
158
159
160 % Initialize log-likelihood value
161 loglik = 9999999;
162
163 end
164
165 % Perform estimation:
166 [ loglik_out, param_out, StdErr, gamma1, psi0, fittersp, fitconvyield ] = mle_na1(
167 param, yy, info );
168
169 % Check if new estimates are better than old estimates and save if so
170 % update parameters and save.
171
172 % % TRANSFORMATION TO EXPORT IN RIGHT UNITS
173 param_export = param_out;
174
175 J = [0 , 1]';
176 phiistar = zeros(2);
177 phistar(1,1) = param_export(1);
178 phistar(2,2) = param_export(2);

```

```

178
179 StdErrphistar = zeros(2);
180 StdErrphistar(1,1) = StdErr(1);
181 StdErrphistar(2,2) = StdErr(2);
182
183
184 mustar = [param_export(3) 0 ]' ;
185 StdErrmustar = [StdErr(3) 0 ];
186
187
188
189 psi1 = param_export(4:5);
190 StdErrpsi1 = StdErr(4:5);
191
192
193 gamma0 = 10*param_export(6);
194 StdErrgamma0 = 10*StdErr(6);
195
196
197 % Constant (state equation)
198 mu = param_export(7:8);
199 StdErrmu = StdErr(7:8);
200
201 % Matrix on lagged values (state equation)
202 phi = reshape([param_export(9) param_export(10) param_export(11) param_export
(12)]',2,2);
203 StdErrphi = reshape([StdErr(9) StdErr(10) StdErr(11) StdErr(12)]',2,2);
204
205 % Covariance matrix (state equation):
206 % Cholesky decomposition of covariance matrix of state equation
207 Qc = zeros(2);
208 StdErrQc = zeros(2);
209
210 Qc(1,1) = param_export(13)/10;
211 StdErrQc(1,1) = StdErr(13)/10;
212
213 Qc(2,1) = param_export(14)/10000000000;
214 StdErrQc(2,1) = StdErr(13)/10000000000;
215
216 Qc(2,2) = param_export(15)/100;
217
218 StdErrQc(2,2) = StdErr(15)/100;
219
220
221 % Covariance matrix of state equation

```

```

222     Q = Qc*Qc';
223
224
225
226     % STD deviation of the measurement errors
227     SS = diag([10*param_export(16) param_export(17)/100] ');
228
229     StdErrSS = diag([10*StdErr(16) StdErr(17)/100] ');
230
231
232
233
234
235
236     if loglik_out <= loglik
237         loglik      = loglik_out;
238         param       = param_out;
239         info_est    = info;
240     end
241
242     fprintf('\n Log-likelihood value : %.12g \n ', -loglik)
243
244     %PARAMETERS
245
246     fprintf('\n Parameters and Standard errors. \n ')
247     fprintf('(Note: Std Err = -99 means that we did not compute them.) \n \n ')%
248
249
250     fprintf('\n phistar = \n'); disp(phistar);
251     fprintf('\n StdErr(phistar)= \n'); disp(StdErrphistar);
252
253     %
254     fprintf('\n mustar= \n'); disp(mustar);
255     fprintf('\n StdErr(mustar)= \n'); disp(StdErrmustar);
256
257     fprintf(['\n gamma0 = ', num2str(gamma0), ' (', num2str(StdErrgamma0),')\n'])
258
259     %
260     fprintf('\n gamma1 = \n'); disp(gamma1);
261
262     fprintf('\n psi0 = \n'); disp(psi0);
263
264     fprintf('\n psi1 = \n'); disp(psi1);
265     fprintf('\n StdErr(psi1)= \n'); disp(StdErrpsi1);
266

```

```

267
268
269 fprintf('\n mu= \n'); disp(mu);
270 fprintf('\n StdErr(mu)= \n'); disp(StdErrmu);
271
272
273 fprintf('\n phi= \n'); disp(phi);
274 fprintf('\n StdErr(phi)= \n'); disp(StdErrphi);
275
276
277 % Cholesky dec of covariance matrix
278
279
280 fprintf('\n Qc= \n'); disp(Qc);
281 fprintf('\n StdErr(Qc)= \n'); disp(StdErrQc);
282 if info.plotfigs>0
283
284
285 % Price of the risk LAMDA(t) and riskpremia
286
287 QClambdat = (mu-mustar)+ (phi-phistar)*XX(:,2:T);
288
289 riskpremaia = (psi1+ gamma1)'*QClambdat -0.5*(psi1+ gamma1)'*Q*(psi1+ gamma1);
290
291
292
293
294 XS = [ones(T-1,1) yyy];
295 YS = riskpremaia';
296
297 BETTA = inv(XS'*XS)*XS'*YS;
298 FITT = XS*BETTA;
299 Res = YS -XS*BETTA;
300
301 Sighat = sqrt(((Res'*Res)/((T-1)-2))*diag(inv(XS'*XS)));
302
303 tva = BETTA./Sighat;
304
305 RCA = 1 -((Res'*Res)/sum((YS-mean(YS)).^2));
306
307
308 disp('Coeficientes de Regresar Riesgo del modelo contra constante slope rezagado y
Crecimiento')
309 disp(BETTA)
310

```

```

311 disp('t-values de la Regresion')
312 disp(tva)
313
314 disp('R^2 de la Regresion')
315 disp(RCA)
316
317
318
319
320 figure
321 title('Affine Pricing Model')
322
323 subplot(3,1,1)
324 plot(time(2:T),[SP(2:T) fittrrsp], 'linewidth',1.5);
325 title('Actual and Fitted Real Stock Prices ')
326 legend({'Real Stock Prices', 'Fitted Stock Prices'}, 'Location', 'Northwest')
327
328 subplot(3,1,2)
329 plot(time(2:T), [DY(2:T) fitconvyield ], 'linewidth',1.5);
330 title('Actual and Fitted Convinience yield')
331 legend({'Real Convinience Yield', 'Fitted Convinience Yield'}, 'Location', '
    northeast')
332
333 subplot(3,1,3)
334 plot(time(2:T), [ riskpremaia' FITT ], 'linewidth',1.5);
335 title('Risk Premia explained by Output Growth and Inflation')
336 legend({'Risk Premia'}, 'Location', 'northeast')
337
338
339 pause;
340
341 end
342
343
344
345
346
347
348
349 % Now plot factor loadings
350 % mtau = 1:180; % To plot factor loadings
351 % slope = (1 - exp(-lambda*mtau))./(lambda*mtau);
352 % curve = slope - exp(-lambda*mtau);
353 %
354 % figure

```

```

355 %     plot(mtau,slope,'b-',mtau,curve,'b--','Linewidth',1.5)
356 %     legend('Loading on \beta_{2t}','Loading on \beta_{3t}')
357 %     legend('boxoff')
358 %     title('Factor loadings: slope and curvature')
359 %     xlabel('Maturity \tau (in months)')
360 %
361
362
363 YXS = [ riskpremaia' Data2(2:T,1) Data2(2:T,2) ];
364
365
366
367 % function results = vare(y,nlag,x)
368 % PURPOSE: performs vector autoregressive estimation
369 %-----
370 % USAGE: result = vare(y,nlag,x)
371 % where:  y      = an (nobs x neqs) matrix of y-vectors
372 %         nlag   = the lag length
373 %         x      = optional matrix of variables (nobs x nx)
374 %                 (NOTE: constant vector automatically included)
375
376 % PURPOSE: An example of using vare, pgranger, prt_var,plt_var
377 %           to estimate a vector autoregressive model
378 %           report Granger-causality results, print and plot
379 %-----
380 % USAGE: vare_d
381 %-----
382
383 % y = YXS;
384 % nlag = 1; % number of lags in var-model
385 %
386 % % estimate the model
387 % results = vare(y,nlag);
388 %
389 % vnames = ['Risk premia ',
390 %          ' Slope      ',
391 %          ' Growth     '];
392 %
393 %
394 % prt(results,vnames);
395 % cutoff = 0.1;
396 % pgranger(results,vnames,cutoff);
397 % plt(results,vnames);
398
399 % nperiods = 48;

```

```

400 %
401 % Vnames = char('Risk premia ', 'Slope ', 'Growth ');
402 %
403 % [m1 , m2] = irf(results, nperiods, 'o1', Vnames);
404
405
406
407
408
409
410
411
412
413
414 % SAVE RESULTS
415 save(namefile, 'param', 'StdErr', 'loglik')
416 disp(['I saved the updated estimates in the file ', namefile])

```

MODELO AFIN

```

1 % THIS PROGRAM ESTIMATES THE YIELD CURVE USING THE DYNAMIC NELSON AND
2 % SIEGEL AND IMPOSING NON-ARBITRAGE RESTRICTIONS. NON-DIAGONAL MODEL.
3 %
4 % The program creates a file "DNS_YieldCurve_NoArbitrage.mat" with the
5 % inputs to be used by the Futures block of the estimation
6
7 clear all; close all
8 tic
9
10 % Estimate parameters?
11 info.estimate = 1; % >0 yes; =0 no
12 info.initval = 0; % >0 if use old estimate as initial guess, =0 otherwise
13 info.start = 1; % First observation to compute likelihood
14
15 % Optimization method:
16 info.do_anneal = 1; % Do simulated annealing; >0 if yes, =0 if not
17 info.do_simplex = 0; % Do Nelder-Meade algorithm? >0 if yes, =0 if no
18 info.do_qnewton = 0; %Do quasi-newton optimization?; >0 if yes, =0 if no
19
20 info.opt_type = 3; % Type of quasi-newton algorithm: =1 uses fminunc; =2 uses
    fminsearch; =3 uses uncmin; =4 uses csminwel; 5 fmincon
21
22
23 info.do_stderr = 1; % >0 if do asymptotic standard errors; =0 if no
24 info.sderr_type = 4; % Type of standard errors to compute:
25 % = 1 use fdhess.m with forward difference
26 % = 2 use fdhess.m with centered difference

```



```

27         % = 3 use fdhess2.m (from gauss) with forward difference
28         % = 4 use outer product of gradient
29
30
31
32
33
34
35 info.plotfigs    = 1; % >0 if plot figures, =0 if not.
36
37 % Choose sample size: fullsample or shortsample. Short sample is used to
38 % initialize the recursive estimation to perform the forecast exercise.
39 info.sample      = 1; % >0 is full sample; =0 is short sample
40
41 % INFORMATION FOR OPTIMIZERS:
42 % Options for simulated annealing:
43 info.options_sa.Verboesity = 2;          % Show intermediate information
44 info.options_sa.InitTemp   = 1;          % Initial temperature
45 info.options_sa.StopTemp   = 1e-15;     % Stop temperature criterium
46
47 % Options for fminunc:
48 info.option_fmunc = optimset('Largescale','off','Display','iter','MaxFunEvals',5000,
49     ...
50     'Maxiter',2000,'TolX',1e-10,'TolFun',1e-10,'Diagnostics',
51     'on');
52
53 % Options for fminsearch:
54 info.options_fmsearch = optimset('Display','iter','MaxFunEvals',5000,'Maxiter',5000);
55
56 % -----
57 fprintf('\n Date and Hour: %s.\n ', datestr(now))
58 fprintf('\n ESTIMATING Present value an Affine Model WITH NON-ARBITRAGE
59     RESTRICTIONS \n ')
60 fprintf('
61     ----- \n '
62     )
63
64 % Load data
65 %load interest_rates_data; % These rates are annualized and in percentage terms
66
67
68 load DATASP; % Int10Y Int1Y CONVYIELD REALDIV REALINT REALSP
69 load Data2.txt; % Slope and Growth
70
71
72 yy = DATASP;
73 yy(:,1) = yy(:,1)/100; %pongo en tasa

```

```

67 yy(:,2) = yy(:,2)/100; %pongo en tasa
68
69
70 if info.sample > 0 % Use all data
71     namefile = 'SPPRICES2.mat';
72     % Set initial and final dates of sample: (T=length(datosinf) is full sample:
73     T0 = 1;
74     T = length(DATASP);
75 else
76     namefile = 'SPPRICES2.mat';
77     T0 = 1;
78     Tend = 459; % This is the last day for which we have data on future prices
79     T = Tend-12*4; % To have 4 years to perform the forecast exercise
80
81 end
82
83
84
85
86 t0 = 1900; % Initial date
87 t1 = 2018; % Final date
88 time = ( t0 : 1 : t1 )';
89
90
91 info.nobs = T-T0+1; % Number of observations
92 info.t0 = t0; % Initial observation
93 info.t1 = t1; % Last observation
94
95 fprintf('\n Information of this run: \n ')
96 disp(info)
97
98 XX = [ log(yy(:,4)) ] ; % Log(dividends and real interest rate
99 SP = yy(:,6); % Real stock prices
100 DY = yy(:,3); % 1Convinience yield
101
102
103 XS = [ones(T-1,1)];
104 YS = XX(2:T) - XX(1:T-1);
105
106 BETTA1 = inv(XS'*XS)*XS'*YS;
107 FITT1 = XS*BETTA1;
108 Res1 = YS -XS*BETTA1;
109
110 Sighat1 = sqrt(((Res1'*Res1)/((T-1)-1))*diag(inv(XS'*XS)));
111

```

```

112
113 yyy = [ Data2(1:T-1,1) Data2(2:T,2) ]; % Usar el slope lagged (ya es lagged) y
      growth
114
115
116
117
118
119 % Initialize parameters:
120 if info.initval > 0 % check if initialize using old estimate (must be in file '
      estimates_noswitch.mat'
121
122     if exist(namefile,'file')==2 % check if file exists
123         load(namefile);
124     else
125         fprintf('File %s does not exist. Set info.initval = 0.',namefile);
126         error(' ');
127     end
128
129 else
130
131
132 %% Estimate parameters of observable factors running a VAR to have a good initial
      guess
133
134 % Now run a VAR using level, slope, curvature, and the 1-year rate;
135
136
137 param = zeros(3,1);
138
139     % Parameter mu
140     param(1) = BETTA1;
141
142     % Parameter sigD
143     param(2) = Sighat1;
144
145
146     % Square root of variances of measurement errors
147     param(3) = [10]';
148
149
150     % Initialize log-likelihood value
151     loglik = 9999999;
152
153 end

```

```

154
155 % Perform estimation:
156 [ loglik_out, param_out, StdErr, K, fittrsp ] = mle_na2( param, yy, info );
157
158 % Check if new estimates are better than old estimates and save if so
159 % update parameters and save.
160
161 %% TRANSFORMATION TO EXPORT IN RIGHT UNITS
162 param_export = param_out;
163
164
165
166 % Constant (state equation)
167 muD = param_export(1);
168 StdErrmuD = StdErr(1);
169
170
171
172 SigD = param_export(2);
173 StdErrSigD = StdErr(2);
174
175
176
177 % STD deviation of the measurement errors
178 SigPD = param_export(3);
179
180 StdErrSigPD = StdErr(3);
181
182
183
184
185
186
187 if loglik_out <= loglik
188     loglik     = loglik_out;
189     param     = param_out;
190     info_est  = info;
191 end
192
193 fprintf('\n Log-likelihood value : %.12g \n ', -loglik)
194
195 %PARAMETERS
196
197 fprintf('\n Parameters and Standard errors. \n ')
198 fprintf('(Note: Std Err = -99 means that we did not compute them.) \n \n ')%

```

```

199
200
201 fprintf('\n muD= \n'); disp(muD);
202 fprintf('\n StdErr(mu)= \n'); disp(StdErrmuD);
203
204
205 fprintf('\n SigD= \n'); disp(SigD);
206 fprintf('\n StdErr(SigD)= \n'); disp(StdErrSigD);
207
208
209 fprintf('\n SigPD= \n'); disp(SigPD);
210 fprintf('\n StdErr(SigPD)= \n'); disp(StdErrSigPD);
211
212 if info.plotfigs>0
213
214
215 % RISK PREMIA
216
217
218 riskpremaia = ((1 + K)/K)*exp(muD + 0.5*(SigD^2))-1 - yy(1:T-1,5);
219
220
221
222
223 XS = [ones(T-1,1) yyy];
224 YS = riskpremaia;
225
226 BETTA = inv(XS'*XS)*XS'*YS;
227 FITT = XS*BETTA;
228 Res = YS -XS*BETTA;
229
230 Sighat = sqrt(((Res'*Res)/((T-1)-2))*diag(inv(XS'*XS)));
231
232 tva = BETTA./Sighat;
233
234 RCA = 1 -((Res'*Res)/sum((YS-mean(YS)).^2));
235
236
237 disp('Coeficientes de Regresar Riesgo del modelo contra constante slope rezagado y
      Crecimiento')
238 disp(BETTA)
239
240 disp('t-values de la Regresion')
241 disp(tva)
242

```

```

243 disp('R^2 de la Regresion')
244 disp(RCA)
245
246
247
248
249 figure
250 subplot(2,1,1)
251 plot(time(2:T),[SP(2:T) fittrsp], 'linewidth',1.5);
252 title('Actual and Fitted Real Stock Prices ')
253 legend({'Real Stock Prices', 'Fitted Stock Prices'}, 'Location', 'northwest')
254
255 subplot(2,1,2)
256 plot(time(2:T), [ riskpremaia FITT ], 'linewidth',1.5);
257 title('Risk Premia explained by Output Growth and Inflation')
258 legend({'Fitted Risk Premia'}, 'Location', 'southeast')
259
260 pause;
261
262 end
263 %
264 %
265 %
266 %
267 %
268 %
269 %
270 % % % Now plot factor loadings
271 % % mtau = 1:180; % To plot factor loadings
272 % % slope = (1 - exp(-lambda*mtau))./(lambda*mtau);
273 % % curve = slope - exp(-lambda*mtau);
274 % %
275 % % figure
276 % % plot(mtau,slope, 'b-', mtau, curve, 'b--', 'Linewidth',1.5)
277 % % legend('Loading on \beta_{2t}', 'Loading on \beta_{3t}')
278 % % legend('boxoff')
279 % % title('Factor loadings: slope and curvature')
280 % % xlabel('Maturity \tau (in months)')
281 % %
282 %
283 %
284 % YXS = [ riskpremaia' Data2(2:T,1) Data2(2:T,2) ];
285 %
286 %
287 %

```

```

288 %% function results = vare(y,nlag,x)
289 %% PURPOSE: performs vector autogressive estimation
290 %%-----
291 %% USAGE: result = vare(y,nlag,x)
292 %% where:   y   = an (nobs x neqs) matrix of y-vectors
293 %%          nlag = the lag length
294 %%          x   = optional matrix of variables (nobs x nx)
295 %%              (NOTE: constant vector automatically included)
296 %%
297 %% PURPOSE: An example of using vare, pgranger, prt_var,plt_var
298 %%           to estimate a vector autoregressive model
299 %%           report Granger-causality results, print and plot
300 %%-----
301 %% USAGE: vare_d
302 %%-----
303 %%
304 %% y = YXS;
305 %% nlag = 1; % number of lags in var-model
306 %%
307 %% % estimate the model
308 %% results = vare(y,nlag);
309 %%
310 %% vnames = ['Risk premia ',
311 %%           ' Slope      ',
312 %%           ' Growth     '];
313 %%
314 %%
315 %% prt(results,vnames);
316 %% cutoff = 0.1;
317 %% pgranger(results,vnames,cutoff);
318 %% plt(results,vnames);
319 %%
320 %% nperiods = 48;
321 %%
322 %% Vnames = char('Risk premia ', 'Slope      ', 'Growth     ');
323 %%
324 %% [m1 , m2] = irf(results, nperiods, 'o1', Vnames);
325 %%
326 %%
327 %%
328 %%
329 %%
330 %%
331 %%
332 %%

```

```
333 %  
334 %  
335 % % SAVE RESULTS  
336 % save(namefile,'param','StdErr','loglik')  
337 % disp(['I saved the updated estimates in the file ', namefile])
```