Business Cycles and Predictability of Returns under Habit Formation

Rasmus M. Jensen & Morten B. Krogh*

ABSTRACT

We re-calibrate the model of Campbell and Cochrane (1999b) on up-to-date U.S. data and simulate 100,000 months of stock returns, we find that the model is able to generate returns with key-properties of empirical observed stock returns. Namely, that the predictability of stock returns is limited in expansions, whilst remaining sizeable in recessions. In addition, we provide results on the overall out-of-sample predictability of stock returns generated from an external habit model and estimate long-horizon regressions conditional on the state of the economy.

Keywords: Consumption-based asset pricing, Habit formation.

JEL classification: G11, G12, G17

*Jensen & Krogh are currently undertaking a Master's Degree at Aarhus School of Business and Social Sciences, Aarhus University. We thank our supervisor Stig Vinther Møller for making this project possible with passionate and inspirational guidance. In addition we would also like to thank Tom Engsted and Thomas Quistgaard Pedersen whose lectures has been a major source of inspiration throughout the process. The code written for this paper is based upon Campbell and Cochrane (1999b)'s GAUSS code and implemented in MatLab *release:* 2019b. Our code is available at `https://github.com/MortenBKrogh/Habit-Models-Advanced-Asset-Pricing`

I. Introduction

Predicting the equity premium has long been a topic of interest for researchers in asset pricing. Early attempts dates back to Dow (1920), who investigated the influence of dividend ratios. The field developed over time and the literature began to consider consumption as a key driver of asset prices and questions such as *»can a consumption-based approach contribute a theoretical explanation of the equity premium puzzle?«*, arose. The consumption based asset pricing setup takes the departure in the classical assumption of a representative agent with power utility and where consumption is log-normally distributed. This setup leads to several well-known puzzles, one of them the equity premium puzzle, which is the inability of general equilibrium models to match the empirical observed equity premium, which in the period 1889-1978 were around 6% Mehra and Prescott (1985), while standard models predicted around 1%.

By introducing a slow-moving habit to the basic power utility function, and thereby letting the relative risk aversion vary over time, Campbell and Cochrane (1999b) are able to match the empirical observed equity premium and several key-properties of the asset returns. Thus, they are able to model human behavior in the downturn of the business cycle were expected returns on stocks are seen to increase together with a higher risk aversion of investors.

In the model of Campbell and Cochrane (1999b) when consumption is approaching habit, the curvature of utility rises, increasing marginal utility, leading to falling prices of risky assets and increased expected returns, thus the model is able to solve the equity premium puzzle.

In this paper we follow the paper of Campbell and Cochrane (1999b). We introduce habits and assume that the consumption choice is restricted by an endowment received each period. We calibrate the model on US data spanning 1950-2018, thus extending Campbell and Cochrane (1999b)'s data period with around 25 years of data. This incorporates both the *»dot com bubble«* of 2000 and the *»Great Financial Crisis«* of 2007-2009, this provides the model with a more appropriate and nuanced information set, in contrast to relying upon 25 years old data for parameter calibration.

In the asset pricing literature, a consensus regarding unpredictability of stock-returns in expansions is present, the findings of (Henkel et al., 2011; Fama and French, 1989), are an example. In this paper we will not only replicate the results Campbell and Cochrane (1999b) on the extended information set, our central contribution to the literature of consumption based asset pricing is the examination of the Campbell and Cochrane (1999b) model's ability to predict stock market returns throughout the business cycle using the common dividend yield predictions of Rozeff (1984).

The analysis is conducted by simulating $100,000$ monthly observations of an economy, wherein we define simulated recessions using the surplus-consumption ratio of the model. A natural way to define the surplus-consumption threshold is to set it to the steady-state level of the model, however this yields that the simulated economy is in recession 37% of the time, which in comparison with empirical National Bureau of Economic Research (*NBER*) data is too much. Instead, we match the empirical recession data to that of our model by integrating over the theoretical density of the surplus-consumption ratio. This makes the model able to match both frequency and duration of *NBER*-recessions.

We estimate both in-sample and out-of-sample regressions based on the simulated data. The data is sub-sampled according to our recession indicator. We find evidence supporting the hypothesis that the external habit model is able to generate stock returns with properties similar to empirical stock returns. Using the dividend-price regression we find that excess stock returns are predictable in recessions with a slope and an intercept of the same magnitude as seen empirically. On a less pleasing note we find that in expansions stock returns generated from the model does still bear a linear relationship with the price-dividend ratio, which is not in-line with the findings of Henkel et al. (2011) and neither in our empirical counterpart-regression, however Perez-Quiros and Timmermann (2001) find predictability in expansions, while lesser in magnitude, is still non-negligible. Lastly, we estimate long-horizon business cycle regressions and obtain results similar to those of Campbell and Cochrane (1999b) and Campbell and Shiller (1988) both in recessions and expansions with R^2 values increasing over the horizon. We find that the R^2-value rises equally fast over the horizon for both recessions and expansions, however the short-run R^2 being much larger, by a factor of 2 or 3 depending on the choice of regressor, leads to much higher explanatory power in recessions, in line with the results of the literature, see for example Henkel et al. (2011).

A. Preliminaries

All regressions conducted in the analysis are based upon logged data. Furthermore all regressions are with excess-returns as the dependent variable, while the regressors are either the price-consumption ratio or the price-dividend ratio. In the model setup as presented below, the price-dividend ratio is a noisier remapping of the price-consumption ratio. Therefore when we mention price-dividend ratio it is interpreted as *either* the price-consumption ratio *or* the price-dividend ratio. In all the regressions the differences are explicit through their respective column specification, when we use "dividend claim" ("consumption claim") it refers to a regression with price-dividend ratio (price-consumption ratio) as the regressor.

We denote logged variables and parameters using minuscule letters while uppercase indicates level; or more precise: the exp of their log counterparts. This distinction is of some importance, as for example g denotes the log mean growth rate of the endowment of the agents, while G denotes the mean growth rate of the endowment of the agents and is thus not a level parameter as per say.

The paper is structured as follows, in section II the data processing for the paper is shortly described, in section III we present the habit model of Campbell and Cochrane (1999b). In IV we first show that we are able to replicate the results of Campbell and Cochrane (1999b), then the business cycle regressions results are presented and lastly in section V we conclude our findings.

II. Data

Data is used in the analysis for the calibration and for empirical regressions. All series used spans *Jan 1950-Dec 2018*. We use monthly consumption data from *Federal Reserve of St. Louis*, the indicator used for consumption is *consumer non-durables*. Data for construction of the dividend-yield *value-weighted* returns distributed by *WRDS* and published from the *CRSP*-dataset, note here that we exclude the *NASDAQ*-exchange due to deviating and unstable behavior of certain periods, returns series utilized thus spans the exchanges *AMEX* and *NYSE*. For construction of the risk-free rate we use the 30-day T-bill as an observable proxy. Lastly we use monthly business-cycle data from the National Bureau of Economic Research *NBER*, this allows for easy matching of the frequency and duration of model-implied recessions and the empirical business-cycle.

The data is used mainly for calibration and computations of unconditional moments and stylized facts regarding the U.S. economy. All data utilized is log-transformed, other relevant transformations of single series, will be noted throughout the paper as they are used, as will the computational usage of the data.

III. Model framework

The framework utilized is a direct application of the model by Campbell and Cochrane (1999b) in the following section a brief overview of the model and model dynamics are presented.

The model can be seen as an extension of the basic consumption-based asset pricing framework in that the standard CRRA-utility function is augmented with habit formation.

$$U(C_t, X_t) = \mathbb{E} \sum_{t=0}^{\infty} \delta^t \frac{(C_t - X_t)^{1-\gamma} - 1}{1 - \gamma} \tag{1}$$

Where X_t denotes the habit level, δ is the subjective discount factor of the agents. Campbell and Cochrane (1999b) proposes that the entire economy can be summarized by the state variable capturing the relationship between consumption and habit - the *surplus-consumption ratio* S_t.

$$S_t \equiv \frac{C_t - X_t}{C_t} \tag{2}$$

The surplus consumption ratio can thus be interpreted, as the consumption level above the habit level. This formulation ensures that the surplus consumption ratio does not go below 0, however as X_t approaches C_t the surplus consumption ratio converge to 0, which indicates an extreme case where the risk aversion in the economy diverges, this result follows directly from the expression of the local curvature of the utility function.

$$\frac{\gamma}{S_t} = -\frac{C_t U_{cc}(C_t, X_t)}{U_c(C_t, X_t)} \tag{3}$$

However, this result, while not well-behaved in extreme cases, implies a convenient feature of this model, namely that the degree of risk aversion increases (decreases) as the surplus consumption ratio declines (increases). The economic interpretation is that as the growth of wealth and by extension consumption, falls below the usual level, the risk aversion of the agents increases which is a well-known trait of agent behavior and the driving forces behind this phenomenon are studied more throughout in behavioral economics.

Following Campbell and Cochrane (1999b) the law of motion of log surplus consumption ratio is defined as a heteroscedastic autoregressive model of order 1,

$$s_{t+1}^a = (1 - \phi)\,\bar{s} + \phi s_t^a + \lambda\left(s_t^a\right)\left(c_{t+1}^a - c_t^a - g\right) \tag{4}$$

where $\lambda(s_t^a)$ is a sensitivity function whose functional form will be further specified below and $\bar{s}, \phi, g$ are parameters either to be matched or calibrated. We note that around the steady state the rational agents all behave equivalent to one another, and the superscript a denoting aggregate measures can be dropped. The last term of (4) follows from the assumed functional form of consumption growth,

$$\Delta c_{t+1} = g + v_{t+1}, \qquad v_{t+1} \overset{\mathcal{IID}}{\sim} \mathcal{N}\left(0, \sigma^2\right) \tag{5}$$

hence,

$$c_{t+1} - c_t - g = v_{t+1} \tag{6}$$

A. Stochastic discount factor/pricing kernel

Starting from the utility function of the rational agent,

$$U(C_t, X_t) = \mathbb{E} \sum_{t=0}^{\infty} \delta^t \frac{\left(C_t - X_t\right)^{1-\gamma} - 1}{1 - \gamma}$$

the first order condition with respect to consumption,

$$U_c\left(C_t, X_t\right) = \frac{\partial}{\partial C_t} U(C_t, X_t) = \left(C_t - X_t\right)^{-\gamma}$$

Substituting X_t from (2): $(-X_t = S_t)$,

$$U_c\left(C_t, X_t\right) = S_t^{-\gamma} C_t^{-\gamma} \tag{7}$$

The stochastic discount factor can then be expressed,

$$\begin{aligned} M_{t+1} &\equiv \delta \frac{u_c\left(C_{t+1}, X_{t+1}\right)}{u_c\left(C_t, X_t\right)} \\ &= \delta \left(\frac{S_{t+1} C_{t+1}}{S_t C_t}\right) \end{aligned} \tag{8}$$

Collecting previous expressions in the system and plugging into (8),

$$M_{t+1} = \delta G^{-\gamma} e^{-\gamma(s_{t+1} - s_t + v_{t+1})} = \delta G^{-\gamma} e^{-\gamma(\phi-1)(s_t - s) + [1+\lambda(s_t)]v_{t+1}\}} \tag{9}$$

Which is the pricing kernel utilized in this paper conditional moments of all variables of interest can be derived from this equation which is a function of the single state s_t.

B. *Risk-free Rate*

In the framework by Campbell and Cochrane (1999b) the risk-free rate is assumed to be constant, this follows from the definition of the risk-free rate,

$$R_t^f \equiv \frac{1}{\mathbb{E}_t\left[M_{t+1}\right]}$$

using equation (8) yields the log risk-free rate,

$$r_t^f = -\ln(\delta) + \gamma g - \underbrace{\gamma(1-\phi)(s_t - \bar{s})}_{A} - \underbrace{\frac{\gamma^2\sigma^2}{2}(1+\lambda(s_t))^2}_{B} \tag{10}$$

to make sure R_t^f is constant the functional form of $\lambda(s_t)$ is chosen such that the effects of intertemporal substitution (A) offsets the precautionary savings-term (B).

In addition the functional form of $\lambda(s_t)$ is chosen to satisfy two additional conditions the three conditions are then given:

1. Risk-free rate constant through time
2. Habit is predetermined at the steady-state
3. Habit is predetermined close to the steady-state

These conditions and expressions yields an expression for the steady-state surplus consumption ratio,

$$\bar{S} = \sigma\sqrt{\frac{\gamma}{1-\phi}} \tag{11}$$

Implying a predetermined habit level in the steady-state. In addition the sensitivity function is then specified,

$$\lambda(s_t) = \begin{cases} \frac{1}{\bar{S}}\sqrt{1-2(s_t-\bar{s})} - 1, & s_t \leq s_{\max} \\ 0, & s_t \geq s_{\max} \end{cases} \tag{12}$$

where $S_{\max}$ is the solution to

$$0 = \frac{1}{\bar{S}}\sqrt{1-2(s_{\max}-\bar{s})} - 1 \tag{13}$$

$$s_{\max} = \bar{s} + \frac{1}{2}\left(1-\bar{S}^2\right) \tag{14}$$

Now substituting these results into equation (10) yields a time-invariant risk-free rate as per construction.

$$r_t^f = -\ln(\delta) + \gamma g - \left(\frac{\gamma}{S}\right)^2 \frac{\sigma^2}{2} = -\ln(\delta) + \gamma g - \frac{\gamma}{2}(1-\phi) \tag{15}$$

Now to refine the framework one could implement a time-varying risk-free rate as suggested by Campbell and Cochrane (1999b) and implemented by Wachter (2005). A time-varying risk-free rate allows for the construction of the term-structure as a function of the state variable, this suggests that by correct specification the term-structure is predictable using the surplus consumption ratio.
However, Campbell and Cochrane (1999b) argues, that the risk-free rate in U.S. data exhibits limited variation furthermore extending the model with a time-varying risk-free rate has little to no effect on the results regarding the stock market. Therefore, the model specification used in this paper assumes a constant risk-free rate, consistent with the model in Campbell and Cochrane (1999b).

C. Pricing a Consumption Claim

The actual pricing relations in this part while very simple analytically are the most comprehensive part computationally and numerically, this follows from the fact that the price-dividend ratio is modelled as a function of the state-variable s_t are not observable and have to be solved on a grid, using the fixed point algorithm procedure, using the Gauss-Legendre quadrature procedure for numerical integration.

To clarify the economical procedure, we utilize the basic pricing relation that is,

$$\begin{aligned} 1 &= \mathbb{E}_t\left[M_{t+1}R_{t+1}\right] \\ R_{t+1} &\equiv \frac{P_{t+1}+D_{t+1}}{P_t} \end{aligned} \tag{16}$$

The functional $P_t/C_t(s_t)$ must then satisfy,

$$\frac{P_t}{C_t}(s_t) = E_t\left[M_{t+1}\frac{C_{t+1}}{C_t}\left[1+\frac{P_{t+1}}{C_{t+1}}(s_{t+1})\right]\right] \tag{17}$$

This functional is solved numerically over a grid of s_t spanning $(0 : s_{max}]$, to increase the precision of the estimation, the grid is an equally distributed linespace but augmented with more mass in the tails this allows the estimation to better capture the non-linearity in the tail of the distribution of the price-consumption ratio. The conditional expectation is then solved on a grid through the Gauss-Legendre procedure over the error term v_t. Being able to solve the conditional expectations allows us to essentially match the left-hand side and right-hand side for each point in the grid. To determine the price-consumption ratio in-between grid-points we use an interpolation procedure. This procedure in general is referred to as a fixed point algorithm, a brief overview of the actual algorithm used in this analysis can be seen in appendix B.

D. *Pricing a Dividend Claim*

The dividend claim process is solved in the same manner as in the previous section however the *iid* log-normal endowment process for dividend growth is constructed such that the innovations in dividend growth w_t, and in consumption v_t is correlated with magnitude ρ,

$$\Delta d_{t+1} = g + w_{t+1}, \qquad w_t \overset{\mathcal{IID}}{\sim} \mathcal{N}\left(0, \sigma_w^2\right), \qquad \text{corr}\left(w_t,\ v_t\right) = \rho \tag{18}$$

The functional price-dividend ratio is then calculated in the same manner as (17)

$$\frac{P_t}{D_t}\left(s_t\right) = E_t\left[M_{t+1}\frac{D_{t+1}}{D_t}\left[1 + \frac{P_{t+1}}{D_{t+1}}\left(s_{t+1}\right)\right]\right] \tag{19}$$

It is worth noting that Campbell and Cochrane (1999b) actually recommends that the relationship between dividend growth and consumption growth should in fact be co-integrated rather than correlated, however we will be closing the model with a correlational relationship rather than the co-integration relation proposed.

E. *Calibration of the model parameters*

We will utilize the analytical results presented by Campbell and Cochrane (1999b) and Engsted and Møller (2010)

To estimate the unconditional mean of the consumption growth, g, in the model we apply the expectations operator to equation (5),

$$\begin{aligned}\mathbb{E}\left[\Delta c_{t+1}\right] &= \mathbb{E}\left[g + v_{t+1}\right], \qquad v_t \overset{IID}{\sim} \mathcal{N}\left(0, \sigma\right), \forall t \\ &= g \end{aligned} \tag{20}$$

hence a consistent estimate of the true unconditional mean, in the model, is given as the sample mean of consumption growth. From this result follows by extension that the unconditional standard deviation is given by σ while the consistent estimate is simply the sample standard deviation of the consumption growth, the same result applies to dividend growth.

To determine the correlation between dividend growth and consumption ratio $\text{corr}(w_t, v_t)$ or ρ we follow the argumentation of Campbell and Cochrane (1999b), that correlation between dividend growth and consumption growth are especially tricky to calibrate as it is non-robust to horizon changes, hence we fix the correlation to the level chosen by Campbell and Cochrane (1999b), they use $\rho = 0.2$ but notes that with different horizons the value lies in-between (0.05-0.25).

To estimate the true autocorrelation-coefficient we use the first order sample autocorrelation function of the dividend-price ratio, i.e,

$$d_t - p_t = \alpha + \phi\left(d_{t-1} - p_{t-1}\right) + \varepsilon_t \tag{21}$$

These dynamics of the dividend yield are true only if the best forecast of future dividend yield can be retrieved from an AR(1) model, that this holds true in the model can be seen from equation (19), stating that the price-dividend ratio is a functional only of the state variable, hence inheriting the dynamics of only one state, the surplus consumption ratio. This hold for all t thus the effects of s_t on $d_{t+1} - p_{t+1}$ can be retrieved using the information contained in $d_t - p_t$.

The local utility curvature, γ, in the model is not calibrated as per se, instead we rely on existing literature such as Campbell and Cochrane (1999b) and fixes $\gamma = 2$. This parameter is different from the γ parameter of the *power-utility* model. In the external habit-model specified above γ scales the risk-aversion with the surplus consumption ratio, and is not the risk-aversion parameter itself. The risk-aversion of the model follows from equation (7), taking the second derivative and multiplying with $\frac{C}{U_C(C_t,S_t)}$ yields the relation

$$\frac{C_t U_{CC}}{U_C} = \frac{-\gamma C_t\left(C_t - X_t\right)^{-\gamma-1}}{\left(C_t - X_t\right)^{-\gamma}} = \frac{-\gamma C_t}{C_t - X_t} = \frac{\gamma}{S_t}$$

Where $S_t = (C_t - X_t)/C_t$. Already from this point it is clear that during periods of low S_t risk-aversion in the model will become very large, and as $S \to 0$, $\gamma/S \to \infty$.

The subjective discount factor is chosen as to match the risk-free rate reported in the *CRSP*-data. Recalling equation (15) allows for a closed form solution for subjective discount factor given an arbitrary calibration of the risk-free rate, the local utility curvature, the persistence coefficient and the growth rate of dividends/consumption,

$$\delta = \exp\left(\gamma g - \frac{1}{2}\left(\left(1-\phi\right)\gamma\right) - r^f\right) \tag{22}$$

that is when the model has been calibrated with all of the estimates above, there exists one unique solution to the δ parameter.

IV. Analysis

A. *Calibrated Model*

We calibrated the model according to the previous section the calibration is reported in Table I.

TABLE I
Parameters of the model

Parameter	Notation	Calibration	Campbell and Cochrane (1999b)
Calibrated			
Mean consumption growth	g	0.0134	0.0189
Standard deviation of Δc_t	σ	0.0152	0.0150
Standard deviation of Δd_t	σ_w	0.1256	0.1120
Log risk-free rate	r^f	0.0109	0.0094
Persistence parameter	ϕ	0.9008	0.8700
Assumed			
Coefficient of local curvature	γ	2.0000	2.0000
Correlation dividends/consumption	ρ	0.2000	0.2000
Implied			
Subjective discount factor	δ	0.9156	0.8900
Steady-state surplus consumption ratio	$\bar{S}$	0.0666	0.0570
Maximum surplus consumption ratio	$S_{\max}$	0.1096	0.0940

[1] All relevant parameters are annualized

[2] Calibrated parameters are estimated from data, assumed are chosen arbitrarily on the grounds of existing literature, while implied parameters are calculated from the calibrated/assumed parameters.

Calibrating the model to the extended data suggests a higher persistence parameter, higher volatility of dividend growth and lower consumption growth, the implied surplus consumption parameters suggest a slightly higher overall surplus consumption in comparison to Campbell and Cochrane (1999b). The difference likely stems from 25 years of extra data.

B. Simulation

From the calibrated model we simulate a chain of 100,000 monthly draws from the economy yielding 8,332 years of simulated time-series. We find in addition to the series that the procedure is extremely sensitive to the distribution of grid points. We use 10 equally distributed grid-points, and an additional 6 just below s_{max}, consistent with the approach of Campbell and Cochrane (1999b).

The simulated economy behaves well, according to most measures, that is it is able to explain the equity-premium puzzle, which is known to cause problems in many models including the widely used *power-utility*-model. The Campbell and Cochrane (1999b) model solves this by introducing time-varying risk-aversion this, however, causes the risk-aversion to diverge when the surplus-consumption ratio is close to 0, causing the model to generate risk-aversion amongst the agents in these periods to be implausibly high, as can be seen in Figure 5 in appendix A, reporting local utility-curvature coefficients as high as 1230, which is in sharp contrast to empirical estimates of risk aversion in the U.S. found to be between .75 and 2 found by Gandelman and Hernandez-Murillo (2015) from the standard *power-utility* model.

TABLE II
Simulated Moments

Statistic	Consumption Claim	Dividend Claim	CC99-Calibration Consumption Claim	CC99-Calibration Dividend Claim
$\mathbb{E}(\Delta c)$	0.0135	0.0117	0.019	0.0174
$\sigma(\Delta c)$	0.0124	0.103	0.0123	0.0915
$\mathbb{E}r^f$	0.0109	0.0109	0.0094	0.0094
$\mathbb{E}\left(r-f^f\right)/\sigma\left(r-r^f\right)$	0.376	0.231	0.436	0.318
$\mathbb{E}\left(R-R^f\right)/\sigma\left(R-R^f\right)$	0.417	0.311	0.476	0.389
$\mathbb{E}\left(r-r^f\right)$	0.0488	0.0453	0.0674	0.0647
$\sigma\left(r-r^f\right)$	0.13	0.197	0.154	0.197
$\mathbb{E}(p-c)$	3.11	3.16	2.89	2.92
$\sigma(p-c)$	0.262	0.29	0.277	0.294

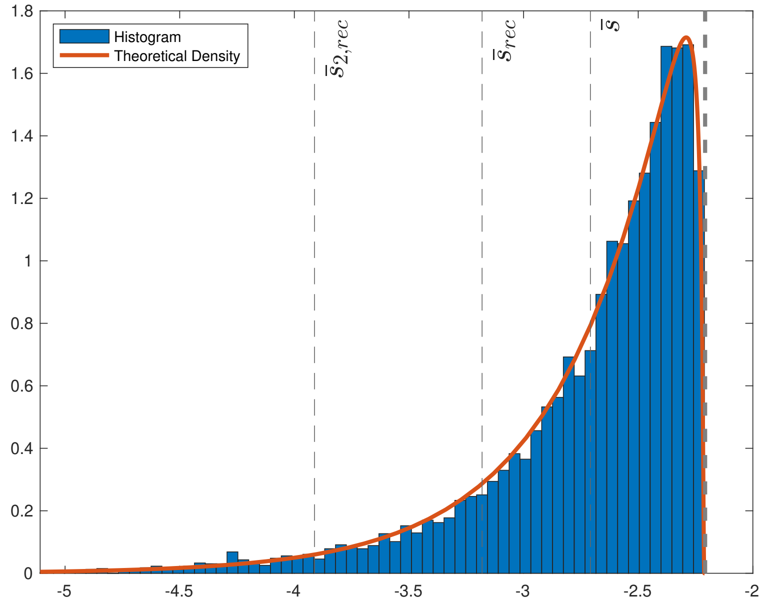
The risk-aversion shortcoming of this model is not the subject of our analysis, and hence are not further examined. The remaining series simulated from the model, and their moments of interest as reported in Table II are well behaved allowing us to examine the predictability of stock returns during times of simulated crisis.

Based on *NBER*-recession data, we find that the U.S. economy was in recession approximately 13.4% of the period spanning January 1950 until December 2018. In the model, a recession implies that present consumption is low relative to habit, that is the value of s_t is lower than the steady state value of surplus consumption $\bar{s}$ - integrating over the density of s_t yields that the simulated economy is in recession during 37% of all observations, which indicates that $\bar{s}$ might be misspecified. To correct $\bar{s}$ we match the empirical business cycle behavior by numerical optimization of the s_t density shown in Figure 1, such that the empirical and simulated economy is in recession roughly the same amount. The $\bar{s}$-value matching the empirical business cycle, throughout denoted $\bar{s}_{rec}$ *or* ($\bar{S}_{rec}$), is found to be -3.18 (0.0415). One additional recession threshold we denote as $\bar{s}_{2,rec}$ is chosen as to capture only the most extreme non-linearity of the relationship between the surplus consumption ratio and expected returns as presented in Figure 2(a).

TABLE III
Business Cycle, Simulated and historic

	Simulated			*Historic*
	$\bar{S}$	$\bar{S}_{REC}$	$\bar{S}_{2,REC}$	
Value	0.067	0.042	0.02	
Recession, %	36.92	13.41	2.83	13.41

Figure 1
Distribution of simulated s_t chain



Histogram of the simulated s_t or $\log(S_t)$, along with the continuous time density as derived analytically by Campbell and Cochrane (1999a).

From the constructed surplus consumption ratio threshold $\bar{s}_{rec}$ for recession, we construct an indicator dummy for recessions as

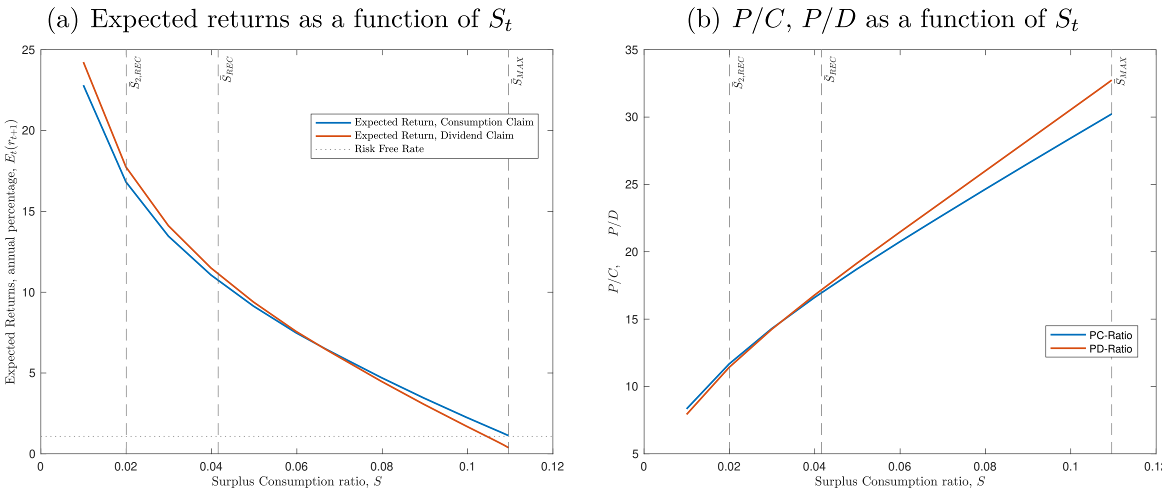
$$I_{rec} = \begin{cases} 1 & \text{if } \bar{s}_{rec} > s_t \\ 0 & \text{otherwise} \end{cases}$$

Having simulated an economy based upon habit formation, and justified our choice of recession periods of the simulated series, we will now investigate the following question:

Is the theoretical model of Campbell and Cochrane (1999b) able to generate stock returns predicTable by the price-consumption- or the price-dividend ratio in recessions, and unpredicTable using the same predictors in expansions as empirical evidence suggests?

Examining the expected returns as a function of the surplus consumption ratio, as in Figure 2(a). The model reveals a somewhat linear relationship for high values of the surplus consumption ratio, the relationship however is much more steep and nonlinear when S_t approaches S_{rec}, the goal is then to exploit this fact when predicting returns.

FIGURE 2
Functionals of surplus consumption ratio



From the functionals of surplus consumption ratio reported in Figure 2 some basic predictions can be made. We see that when the surplus consumption ratio is low, that is in recessionary times, expected returns increases, the intuition is that when times are bad, as consumption falls relative to habit levels, the risk aversion increases, people want a higher level of compensation per unit of risk - the risk premium (excess return) rises - note that this must hold true for all levels of S as the risk-free rate is fixed in the model.

Furthermore, the Figure implies that the regression coefficient of $p_t - d_t$ or equivalent $p_t - c_t$ should be negative, i.e., Lower price-dividend ratio implies higher values of $\mathbb{E}\left[r_t - r^f\right]$, both driven by the single state s_t.

C. Results

First we estimate the following regressions:

$$r_{t+1} - r^f = \alpha + \beta_{rec}\left(p_t - c_t\right) * I_{t,rec} + \beta_{exp}\left(p_t - c_t\right) * \left(1 - I_{t,rec}\right) + \varepsilon_{t+1} \tag{23}$$

$$r_{t+1} - r^f = \alpha + \beta_{rec}\left(p_t - d_t\right) * I_{t,rec} + \beta_{exp}\left(p_t - d_t\right) * \left(1 - I_{t,rec}\right) + \varepsilon_{t+1} \tag{24}$$

$$\begin{aligned} r_{t+1} - r^f = &\left(\alpha_{rec} + \beta_{rec}\left(p_t - c_t\right)\right) * I_{rec} \\ &+ \left(1 - I_{rec}\right)\left(\alpha_{exp} + \beta_{exp}\left(p_t - c_t\right)\right) + \varepsilon_{t+1} \end{aligned} \tag{25}$$

$$\begin{aligned} r_{t+1} - r^f = &\left(\alpha_{rec} + \beta_{rec}\left(p_t - d_t\right)\right) * I_{rec} \\ &+ \left(1 - I_{rec}\right)\left(\alpha_{exp} + \beta_{exp}\left(p_t - d_t\right)\right) + \varepsilon_{t+1} \end{aligned} \tag{26}$$

That is we use the recession indicator to infer the influence of the predictors in recessions and expansions respectively.

The regressions (25) and (26) are estimated over two steps; once for expansions, and once for recessions. With these two regressions, we estimate excess returns both with a constant intercept during all business cycle phases, and in addition where we allow the intercept to change over business-cycles, such that the slope is not the only thing driving predictability differences over recessions and expansions.

As a benchmark we estimate two additional regressions given by:

$$r_{t+1} - r^f = \alpha + \beta\,(p_t - c_t) + \varepsilon_{t+1}$$
$$r_{t+1} - r^f = \alpha + \beta\,(p_t - d_t) + \varepsilon_{t+1}$$

Note that these regressions do not take information about the business cycle into account, they do however produce negative coefficients in line with the results of Campbell and Shiller (1988). The benchmark result is shown in Table IV.

Table IV
Benchmark regressions

Dividend Claim			Consumption Claim		
Constant	β	$R^2, \%$	Constant	β	$R^2, \%$
0.08	-0.014	0.47	0.078	-0.013	0.83

[1] Regressions on 99,999 months of simulated data.

Table V reports results of the four regressions. The desired property of separability of predictability during phases of the business cycles gains support from this result. We see a significant increase in R^2, we find strong statistical evidence supporting rejection of the null-hypothesis for all estimates in all four regressions, this is however to be expected as a sample size of between 12,000 and 100,000 observations is quite large, and it is well established that as the sample size increases the asymptotic variance of the OLS-estimate decreases, and thus significance is inevitable as the sample size becomes quite large.

Table V
Regressions, $\bar{S}_{REC} = 0.041542$

Dependent variable: r^e_{t+1}						
	Dividend Claim			Consumption Claim		
Constant	0.09133	0.2174	0.1466	0.08802	0.1828	0.1451
β_{REC}	-0.01589	-0.04198		-0.01529	-0.03507	
β_{EXP}	-0.01547		-0.02497	-0.01496		-0.02488
R^2, %	0.48	1.9	0.58	0.84	2.2	1.2

[1] Expansionary observations: 85,950, recessionary observations: 12,220, full-sample (*including overlaps*) 99,999

In Table VI we estimate the empirical counterpart to this regression using the same data we used in the calibration of the model. We use the *30-day* T-bill as the risk-free rate and the *NBER U.S.* recession indicator, note that as to remain true to realistic empirical studies the risk-free rate is not assumed to be constant for the purpose of this regression.

Table VI
Empirical price-dividend regressions

Dependent variable: r^e_{t+1}					
Recession			Expansion		
Constant	β	R^2, %	Constant	β	R^2,%
0.191	-0.032	4.42	0.015	-0.00001	0.47
(0.0907)	(0.0159)		(0.0043)	(0.0000)	

[1] Brackets below estimates contains Newey and West (1987) corrected standard errors *1 lag*.

[2] Regressions on monthly data spanning Jan. 1950 - Dec. 2018.

The slope of the price-dividend ratio during recessions is of the same magnitude as the one estimated from the simulated economy (-0.035 vs. -0.032) same goes for the constant in recessions. This result follows from column 2 of the consumption claim part of Table V. The expansionary slope coefficient deviates quite a lot, we see how $\beta_{EXP} \approx 0$, implying that during expansions price dividend ratio has zero impact on excess returns. In the model however the correlation between excess returns and the price-dividend ratio is inevitable. We see this clearly from Figure 2, where even out of recessions the inverse almost linear relationship is quite strong.

To examine whether this results holds *out-of-sample*, we conduct a rolling-window forecast over the simulated observations. As the underlying s_t-chain leads to realistic economic business cycles, and thus non-continuous chains of recessions, we stack all recession periods in one sample, this is justified in the model as the state variable s_t behaves consistently, and the behavior of the remaining series behaves according to s_t, that is we have no structural changes over t in the model.

We use the squared forecast errors to estimate the R^2_{OoS} proposed by Campbell and Thompson (2005), the computed R^2_{OoS} is reported in Table VII, where the benchmark model is the prevailing mean. In the empirical literature the prevailing mean, has been shown to be quite difficult to out-perform when it comes to predicting returns, this and its widespread use as a benchmark is the reason we chose it as our benchmark model in the out-of-sample R^2-test.

Table VII
Out-of-sample R^2

	Dividend Claim		Consumption Claim	
	Expansions	Recessions	Expansions	Recessions
R^2_{OoS}, %	-1.22	1.04	-0.88	1.07

[1] Based upon 1-*period-ahead* rolling-window forecast with a window size of 120 months.

[2] Expansionary observations: 85,218, recessionary observations: 11,488

The interpretation is that the simple forecast of excess returns using the price-consumption ratio or equivalently price-dividend ratio compared to the no-predictability forecast, the prevailing mean forecast of excess returns, the simple regression performs better (positive sign) only in recessionary periods, while the mean-model is superior in times of economic prosperity in the model. For our hypothesis this result is good news, in times of expansions the price-dividend ratio does not predict excess stock returns out-of-sample better than the prevailing mean model, while the opposite remains true in recessions.

However from Figure 2(a) we see that the most extreme effect of S_t on expected returns kicks in at around $S_t = 0.02$, based on this fact we run additional regressions on the same form as above, but with a re-specified $\bar{S}_{REC} = 0.02$ and denote this $\bar{S}_{2,REC}$, the output of which is reported in Table VIII. This specification of $\bar{S}_{2,REC}$ implies that the simulated economy is in recession approximately 2.8% of the time. Corresponding to barely 3,000 observations in this case.

This specification indicates that recessions are defined now as only the most severe times of the simulated economy. The surplus consumption is extremely low, the threshold value of 0.02 indicates relative levels of risk aversion in these periods of at least 100, which is huge, leading to expected returns of above 17%. This gives us an idea of the state of the economy. The intuitive predictions based upon the model framework is that we should obtain similar results to those of Table V, however with the effect of price-dividend ratio on excess returns in recessions being much larger and even more predictability in recessions.

Table VIII
Regressions, $\bar{S}_{REC} = 0.02$

	Dependent variable: r^e_{t+1}					
	Dividend Claim			Consumption Claim		
Constant	0.07881	0.3536	0.09586	0.07811	0.3112	0.09567
β_{REC}	-0.01316	-0.07347		-0.0132	-0.06462	
β_{EXP}	-0.0133		-0.01626	-0.01323		-0.01631
R^2, %	0.47	3.3	0.45	0.86	3.8	0.86

[1] Expansionary observations: 96,435, recessionary observations: 2,972, full-sample (*including overlaps*) 99,999

D. *Long-run regressions*

Table IX presents long-run regression results of regressing log excess returns on log price-dividend ratio in artificial data produced by the model. The coefficients are negative translating to high prices leading to low expected returns. The magnitude of the coefficient are increasing linearly in horizons. Furthermore, we see that the R^2 values increase with the horizons, and even though the values are relatively low and unimpressive at short horizons, they increase to $\approx 55\%$ for the consumption claim and $\approx 26\%$ in case of the dividend claim, why is there such a big difference between the two? Recalling that the dividend claim returns contains noise from the dividend growth process, volatility of dividend growth is essentially amplified consumption-growth noise in the model, this explains the lower R^2 values. These results are also consistent with those of Campbell and Cochrane (1999b) and Campbell and

Shiller (1988).

Table IX
Long Run Regressions

Horizon, Years	β_{pc}	R^2_{pc}	β_{pd}	R^2_{pd}
1	-0.135	0.0736	-0.137	0.041
2	-0.273	0.1574	-0.278	0.086
3	-0.398	0.2324	-0.406	0.125
5	-0.617	0.3580	-0.628	0.187
7	-0.790	0.4525	-0.802	0.228
10	-0.979	0.5492	-0.987	0.259

[1] Regressand: additive excess stock returns over months from 12-120 months

Recall the aim of this paper is to clarify whether the external habit model proposed by Campbell and Cochrane (1999b) is able to predict excess returns during recessions as well as in expansions, therefore Table X shows long-run regression results in the same manner as Table IX, but the effects are divided according to the business cycle. That is, we have two samples one with data in recessions, another with data in expansionary periods. This can be interpreted as the economy always being in expansion or recession depending on the sample. We are aware that recessionary periods roughly lasts for 14 months according to *NBER* data, and thus it might provide little economic sense to assume an economy which is in permanent recession or permanent in expansion. However, to study the properties of excess returns in a theoretical experimental study we rely on this methodology, keep in mind the argumentation as before, in the simulation we did not introduce any structural brakes hence all simulated series behaves consistently throughout all business-cycles, i.e., in the model all recessions are alike.

First thing to note is that the coefficients still are negative, hence the results of e.g., Campbell and Shiller (1988) are still present. Second thing to note in Table X is how the magnitude of the coefficients have increased due to the business cycle split, and that is both in expansionary and recessionary periods. For the 1-year horizon with the price-consumption predictor, the magnitude of the coefficient has changed from -0.135 in the full-sample long-run regression from Table IX to -0.253 in expansions and to -0.342 in recessions. For the dividend claim the result is similar as expected.

The difference between the consumption- and dividend claim regressions in expansions is very small, thus at the 1-year horizon the difference is only present at the 3^{rd} decimal, 0.001, but increases a little through horizons. Turning to the recession pane of Table X we clearly see a different picture, where the size of the coefficients clearly is affected by the choice of pricing claim.

Table X
Long Run Regressions, Business Cycle Split, $\bar{S}_{REC}$

	Expansions				*Recessions*			
Horizon, Years	β_{pc}	R^2_{pc}	β_{pd}	R^2_{pd}	β_{pc}	R^2_{pc}	β_{pd}	R^2_{pd}
1	-0.253	0.1171	-0.254	0.0541	-0.342	0.2144	-0.406	0.1667
2	-0.440	0.2034	-0.436	0.0863	-0.581	0.3577	-0.687	0.2604
3	-0.597	0.2751	-0.583	0.1089	-0.735	0.4472	-0.890	0.3149
5	-0.821	0.3755	-0.794	0.1312	-0.947	0.5578	-1.123	0.3532
7	-0.968	0.4385	-0.926	0.1356	-1.072	0.6104	-1.268	0.3591
10	-1.099	0.4908	-1.044	0.1304	-1.165	0.6189	-1.386	0.3390

[1] Regressand: additive excess stock returns over months from 12-120 months

Looking at the explanatory power (R^2), in Table X, we note the same pattern, that they increase with horizons, as in full-sample long-run regression in Table IX and in Campbell and Cochrane (1999b).

We also see increasing R^2-values compared to the full-sample long-run regression in Table IX where they at the 1-year horizon were 0.0736 for the price-consumption claim and 0.041 for the price-dividend claim. In the expansions' pane of Table X the R^2 are 0.1171 and 0.0541, for consumption claim and dividend claim respectively, and in the recessions' pane the corresponding R^2-values are 0.2144 and 0.1667. That is a factor of 2 or 3 difference in 1-year ahead predictions over business cycles depending on the regressor of choice. Thus, by isolating the effects according to the business cycle, we are actually able to increase the R^2 both in expansions and in recessions at the 1-year horizon. Looking at the 10-year horizon, we note that for the price-consumption claim, the R^2 in expansions is lower than the relating number in Table IX, and in recessions the R^2 is higher, which supports the results of the literature of predictability in recessions and less predictability in expansions. That the magnitude of R^2 declines for expansions, while increasing in recessions, indicates that we successfully untangled the non-linearity of the long-run price-dividend ratio and excess stock returns.

A disadvantage of conditioning the information set on the business cycle is that it is not observed by the investor in advance, thus the investor does not know whether the economy will change state, or whether it will stay in one regime throughout the investors time horizon. A possible approach to this problem could be to utilize a *Hidden Markov*-model to predict the state of the economy in a forecasting exercise, however this will extend the investment decision with more uncertainty.

E. *Economic Interpretation*

The economic interpretation follows from the model intuition. We see how in times of recession the risk-aversion increases exponentially, this implies that as we set the threshold for recessions lower in the model the predictability of stock returns during recessions increases. That is the marginal contribution of predictability of the dividend yields in recessions decreases as surplus consumption relative to habit level increases. Using the model implied threshold of S leads to little to no difference in predictability over business cycles. The empirically implied threshold leads to statistically significant differences, while lowering the threshold even further magnifies the results.

The model takes into account that when we enter periods with expansionary tendencies, the price-dividend ratio is increasing people claim their yield, and consumption increases relative to the habit level. This leads to increasing prices of assets. As consumption increases relative to habit the utility of all agents are high, risk means less to the agents in this state of the economy hence the risk-premium they demand for holding risky assets declines.

On the contrary entering recessions, the price-dividend ratio declines, less yields leads to lower consumption amongst the agents, and by extension a lower consumption relative to habit. People exposed to relative losses, fearing further losses, increases the premium they demand for participating in a risky gamble; their risk-premium. This result is well-established, but notoriously difficult for models to match, often referred to as the equity-premium puzzle. Augmenting the simple *power-utility* model with external habit formation incorporates this feature, but with the cost of a badly behaved risk-aversion process; the simulated series of risk-aversion as can be seen in Figure 5 in Appendix A.

This brief economic interpretation of the model relationship, explains the results we find statistically, mainly that the price-dividend ratio is negatively correlated with the excess-returns (risk-premium). The magnitude by which the β_{REC} is lower than β_{EXP} is governed by the γ-coefficient governing the size of the relative risk-aversion. To find results more in-line with those found in Table VI, we would need the upper tail of Figure 2(a) to be essentially flat, indicating that the expected returns are in-sensitive to different values of the surplus-consumption ratio above the expansionary threshold, unfortunately the model is not able to accommodate this.

V. Conclusion

After re-calibrating the model of Campbell and Cochrane (1999b) and simulating from the re-calibrated model, we have shown how the model with external habit-formation is able to consistently generate returns with properties similar to returns observed in the real world. Returns generated from the model, even when we fix the risk-free rate, exhibits predictability only during recessions. Furthermore, the worse the recession the higher predictability from the price/dividend ratio. This result extends to the dividend yield following the reciprocal link between the two. It must be noted that the linear relationship between segments of the price-dividend ratio and excess returns remains sizable even when the surplus consumption ratio is very high, this seems not to be the case in empirical stock returns, where the co-linearity vanishes in expansions.

The estimates of the long-horizon regressions provides evidence that simulated returns are predictable during times of crisis in the model, and much less predictable during expansions, the latter being a bit more controversial than the former. As much of the predictability of empirical stock returns, seems to be concentrated during recessions, whilst some findings report none predictability in expansions, e.g. (Henkel et al., 2011), while others find statistical evidence supporting at least some predictability of stock returns in expansions, e.g. (Perez-Quiros and Timmermann, 2001). We find that both the magnitude of coefficients and R^2 rises significantly in recessions compared to expansions and also when we do not divide the sample according to the business cycle.

Goyal and Welch (2004) finds that returns are in no means predictable, this finding is in contrast to our and numerous other findings, we argue that this finding is driven by the fact that Goyal and Welch (2004) did not consider business cycles when examining returns. As the economy is most often in expansion, the unpredictability of returns in expansions suppresses the magnitude of predictability in recessions, thus landing the estimates of the full-sample closer to them of the expansionary sample only. Therefore, we find that it is of importance to incorporate business-cycle information into forecasts of excess-returns.

Our findings are not only good news, indeed we find that predictability increases drastically during recessions which might be of value for the typical investor. However, we see that the major increase of predictability occurs only during very dark times in the economy. Assuming that the S_t process generated by the model is a good representation of the true business cycle dynamics of the empirical economy, only during a very small fraction of time returns are (a bit) predictable and the dividend-price regression provides a better forecast than the prevailing mean model.

We chose to follow the constant *risk-free-rate* approach, that means that the *term-structure* is not considered and *bond-returns* are constant through maturities. Bond returns are thus not considered in this paper, we did however include the option to extend our study by adding the entirety of our MatLab-codes, where the option to deviate from constant interest rates and a flat term-structure are present.

REFERENCES

Campbell, John Y., and John Cochrane, 1999a, Appendix for ”by force of habit: A consumption-based explanation of aggregate stock market behavior”.

Campbell, John Y., and John H. Cochrane, 1999b, By force of habit: A consumption-based explanation of aggregate stock market behavior, *Journal of Political Economy* .

Campbell, John Y., and Robert J. Shiller, 1988, Stock Prices, Earnings and Expected Dividends, Cowles Foundation Discussion Papers 858, Cowles Foundation for Research in Economics, Yale University.

Campbell, John Y, and Samuel B Thompson, 2005, Predicting the equity premium out of sample: Can anything beat the historical average?, Working Paper 11468, National Bureau of Economic Research.

Dow, Charles Henry, 1920, Scientific stock speculation, *The Magazine of Wall Street* .

Engsted, Tom, and Stig V. Møller, 2010, An iterated GMM procedure for estimating the Campbell–Cochrane habit formation model, with an application to danish stock and bond returns, *International Journal of Finance & Economics* 15, 213–227.

Fama, Eugene F., and Kenneth R. French, 1989, Business conditions and expected returns on stocks and bonds, *Journal of Financial Economics* 25, 23 – 49.

Gandelman, Nestor, and Ruben Hernandez-Murillo, 2015, Risk Aversion at the Country Level, *Review* 97, 53–66.

Goyal, Amit, and Ivo Welch, 2004, A comprehensive look at the empirical performance of equity premium prediction, Working Paper 10483, National Bureau of Economic Research.

Henkel, Sam, J. Spencer Martin, and Federico Nardari, 2011, Time-varying short-horizon predictability, *Journal of Financial Economics* 99, 560–580.

Mehra, Rajnish, and Edward Prescott, 1985, The equity premium: A puzzle, *Journal of Monetary Economics* 15, 145–161.

Newey, Whitney K., and Kenneth D. West, 1987, A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix, *Econometrica* 55, 703–708.

Perez-Quiros, Gabriel, and Allan Timmermann, 2001, Business cycle asymmetries in stock returns: Evidence from higher order moments and conditional densities, *Journal of Econometrics* 103, 259–306.

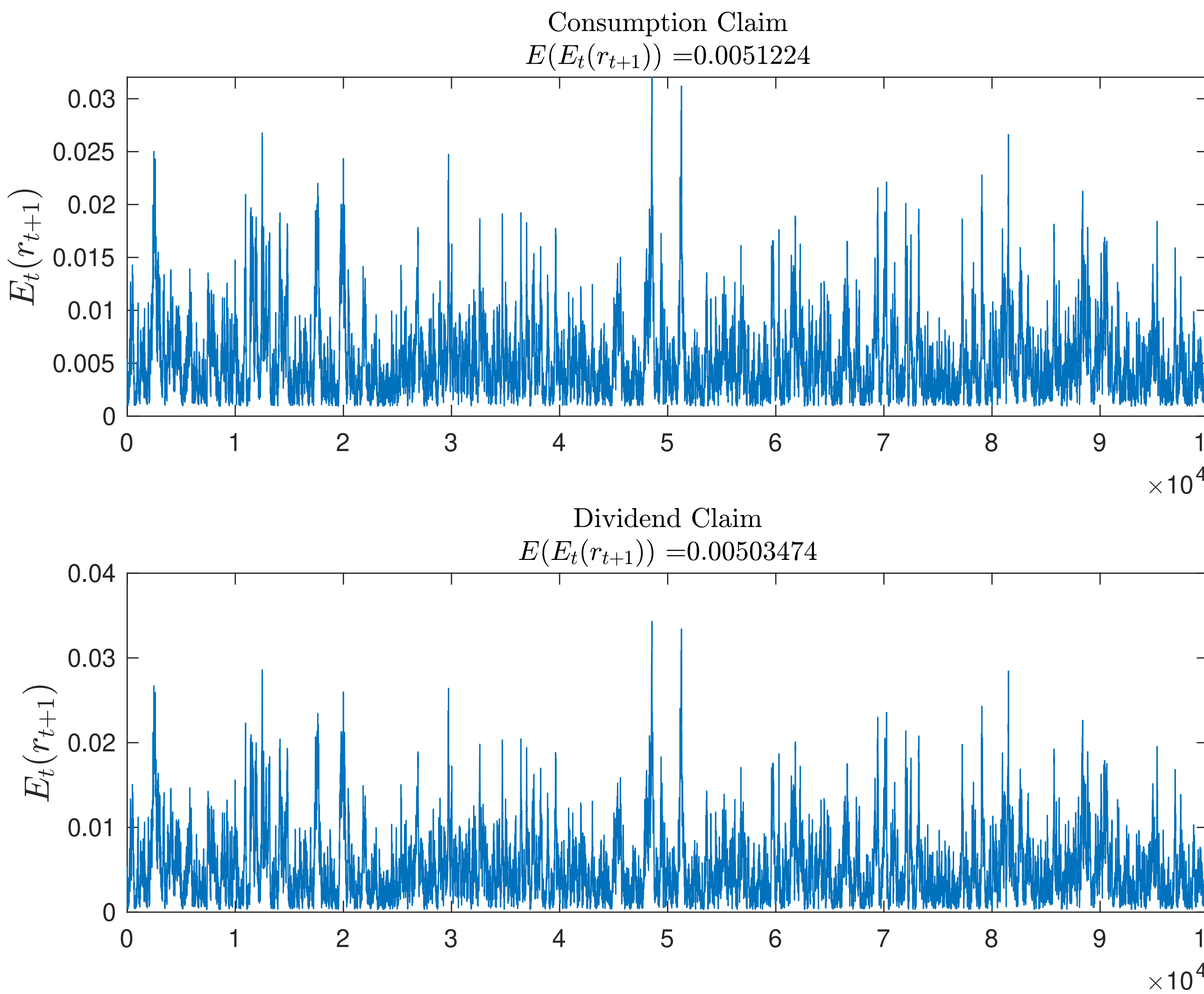
Rozeff, Michael S., 1984, Dividend yields are equity risk premiums, *The Journal of Portfolio Management* 11, 68–75.

Wachter, Jessica A., 2005, Solving models with external habit, *Finance Research Letters* .

Appendix A. Supplementary figures

Figure 3

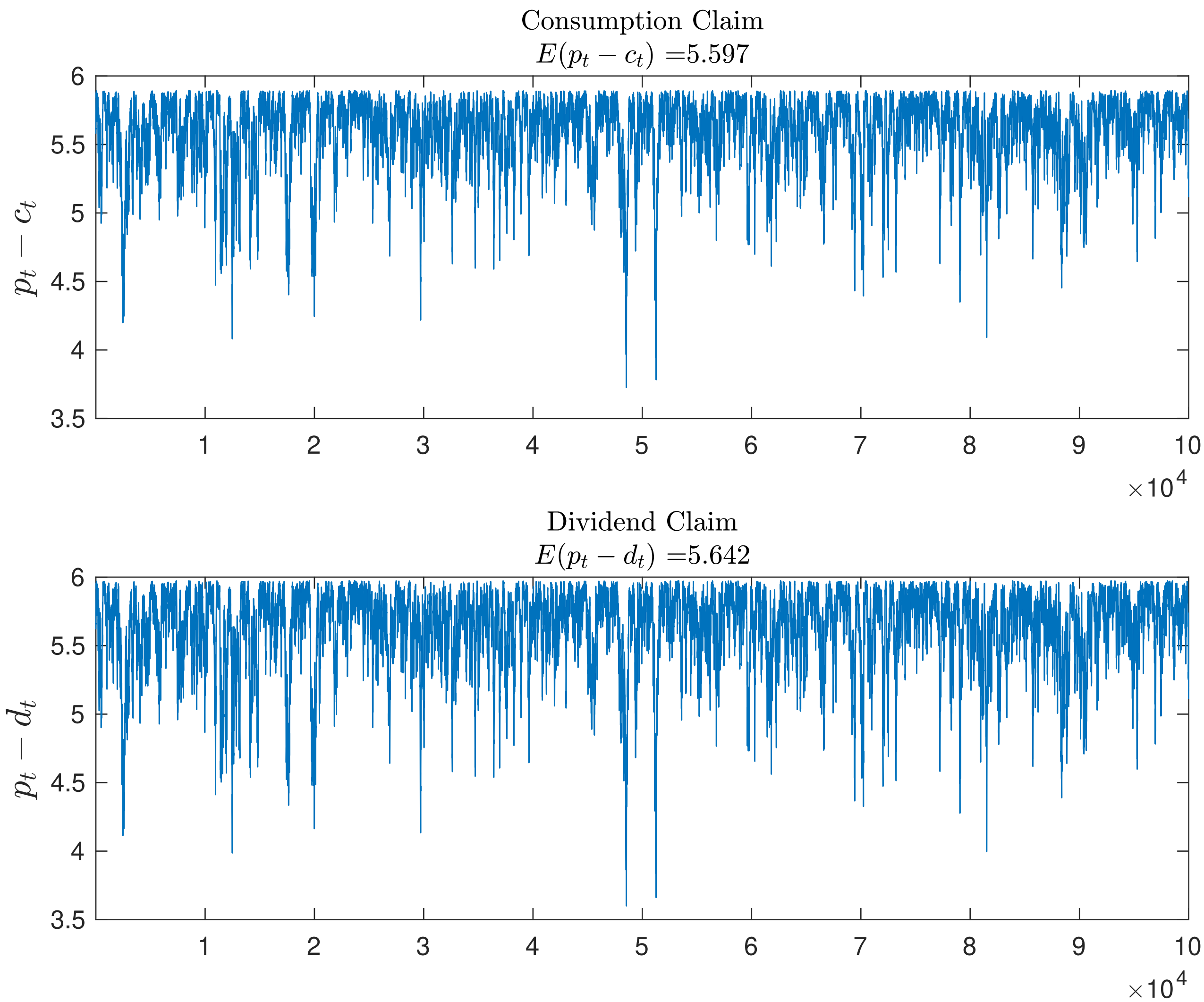
True $\mathbb{E}_t\left(r_{t+1}\right)$



This figure plots the true expected monthly returns of the model. That is for each $t \in \{1, 2, ..., 100000\}$ the model uses available information up until t to determine the conditional future mean of returns before $t+1$ realizes.

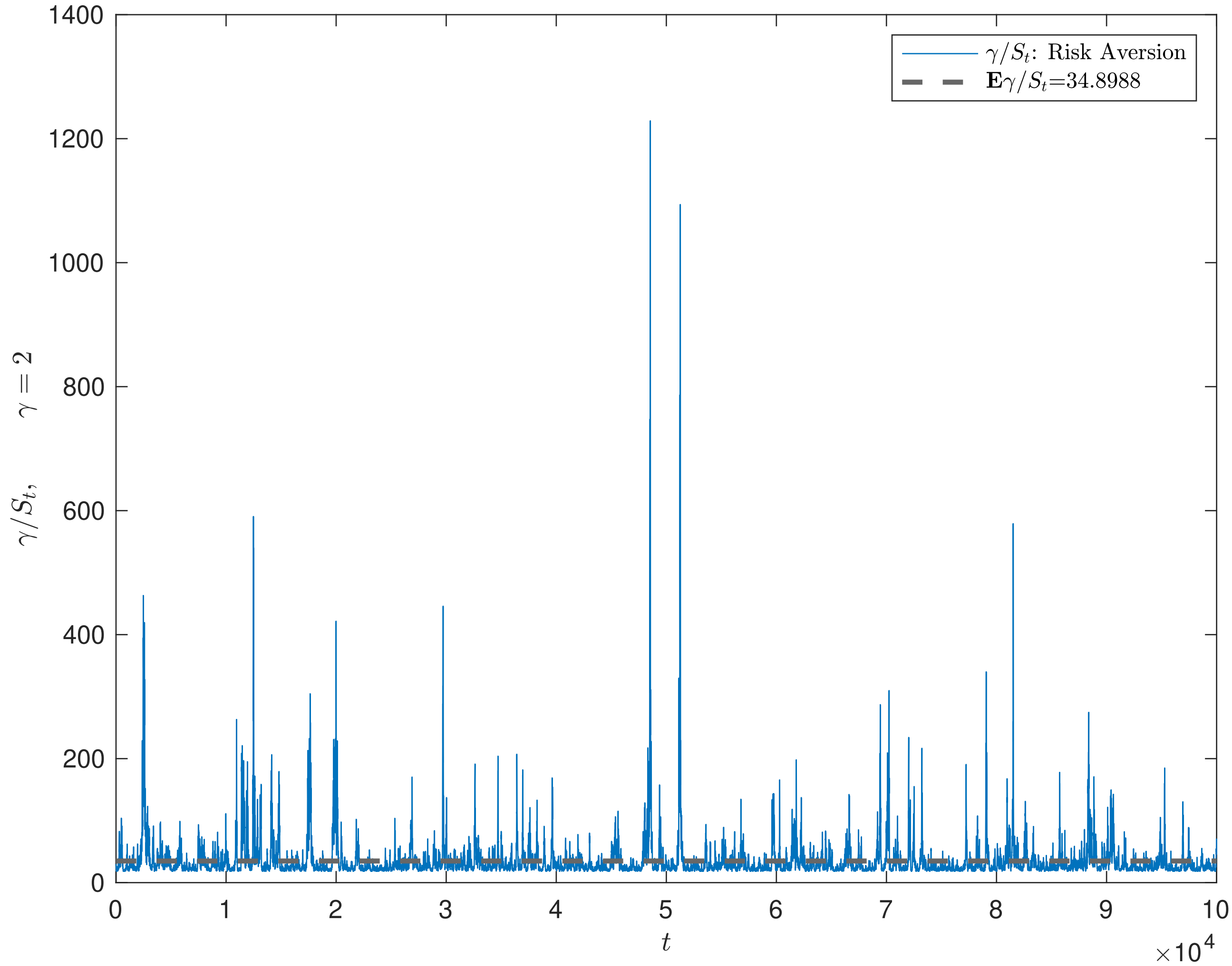
FIGURE 4

Chain of simulated monthly $p-d$, $p-c$



Note: extracted from the solution of the difference equation of price-consumption- and the price-dividend ratio, (17) and (19) respectively, as determined by the fixed point algorithm described in appendix B.

FIGURE 5
Simulated chain of risk aversion



The most extreme observation of γ/S_t around the 50,000 observation mark implies a value of $S = 0.00163$, while the mean implies $S = 0.0573$

Appendix B. Fixed-point Algorithm

Here we will present a brief overview of the solution method used to solve the model:

Output: log price-dividend (log price-consumption) functional

Input : Gridpoints, Calibration

initialize;

lnpd = 0;

erro = 1;

while $Err \geq 1e-6$ **do**

  for *i = 1:length(Gridpoints)* **do**

    s = Gridpoints(i);

    $newlnpc(i) = log\left(\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{\sqrt{v/\sigma}}{2}\right) * pcpdVal \;\; \mathrm{d}v\right)$

  end

  tv = max(abs((exp(newlnpc)-exp(lnpc))./exp(newlnpc)));

  lnpc = newlnpc;

  erro = max(tv);

end

return *lnpd*

Algorithm 1: Fixed-Point Algorithm for the model

Note that we truncate the infinite integral boundaries to $\pm 8\sigma$ (This boundary captures all of the mass under the PDF down to machine precision). Note also that the solution method is almost the same for both the price-dividend- and the price-consumption ratio. The difference lies in the *pcpdVal*-term denoting the value of $p-c$ or $p-d$ for a given v. We use the following piece of MatLab-code to specify *pcpdVal* for both $p-c$ and $p-d$:

```
if PD_Claim == 0  % Price-Consumption
   pcpdVal = delta * exp(g*(1-gamma))*exp(-gamma*(s1-s)).*...
          (1+exp(interp(s1,sg,lnpc)))'.*exp((1-gamma)*v);

elseif PD_Claim == 1 % Price-Dividend
   pcpdVal = delta * exp(g*(1-gamma))*exp(1/2 * (1-rhow^2) * sig_w^2 ) *...
         exp(-gamma * (s1 - s )) .* (1+exp(interp(s1,sg,lnpc)))' .* ...
         exp((rhow * sig_w/sig - gamma)*v);
end
```

sg denoting the gridpoints, s denoting the current gridpoint under evaluation, γ, σ, ρ_w, σ_w are the structural parameters, v denotes a draw from a normal, i.e. $v \sim \mathcal{N}(0, \sigma)$, `interp` is a function doing linear interpolation.

Appendix C. Code

Presented below is the entirety of our code, data files are not included here but are mainly used for the calibration part of the code. Hence it will be necessary to specify the calibration manually if one where to run the script outside our workspace, all parameters of our calibration are reported in table I.

Appendix A. Main Script and Regressions

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%      THIS FILE CAN ONLY RUN ON A WINDOWS MACHINE SINCE         %%
%%      WE USE THE GAUSLEGENDRE FUNCTION TO PERFORM THE NUMERICAL %%
%%      INTEGRATION, WHICH CALL THE COMPILED QUADLAB FUNCTION.    %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% This program is able to reproduce results of Campbell & Cochrane (1999)
% and recalibrate their model to an extended time frame. Then the program
% defines recessions based on NBER recession data, this allows us to
% divide the data sample in two parts respectively a recession sample and
% an expansion sample, which then is used to estimate two regressions to
% test predictability of excess returns in recessions as well as in
% expansions. The program is inspired by Campbell & Cochrane (1999) GAUSS
% code availible at John H. Cochrane's homepage:
% https://faculty.chicagobooth.edu/john.cochrane/
%
% For reproduction of our results in Jensen & Krogh (2019) the MAIN.m code
% needs to be run 4 times with preferences
%
% The first 2 runs use Campbell & Cochrane (1999) calibration and solves
% for both price-consumption and price-dividend.
%
% 1. Run
%       calib = 0, PD_Claim = 0
% 2. Run
%       calib = 0, PD_Claim = 1
%
% The next 2 runs uses Jensen & Krogh (2019)'s calibration (Extends the
% calibration period with 20 years) and then solves for the
% price-consumption and price-dividend.
%
% 3. Run
%       calib = 1, PD_Claim = 0
% 4. Run
%       calib = 1, PD_Claim = 1
%
% The program automatically saves the workspaces of the 4 runs such that
% the LoadData.m file can be run and changed independently of the MAIN.m
% file.

clear all
clc
```

```
format long
tic
addpath('Functions');
addpath('Data');
addpath('Workspaces');
addpath('Calibration');
addpath('Figures');
addpath('Tables');
%% Defining Globals
global g sig delta phi gamma S_bar s_bar S_max s_max tsc sg B maxcb ncalc ...
    bondsel rhow seedval verd debug ann lnpca con sig_w lnpda PD_Claim Regressions

%% Choices for solution methods
% Calibration Choice
calib=1;          % 0 - Campbell & Cochrane (1999)
                  % 1 - Krogh & Jensen (2019)

% Solution method:
PD_Claim = 1;     % 0 = Consumption Claim
                  % 1 = Dividend Claim
                  % Run with both = 0 and =1 before generating figures and
                  % tables and regressions
% Plots
Plots = 0;        % 0 = off
                  % 1 = on
                  % Set = 0 for the first two runs
% Update tables
Tables = 0;       % 0 = off
                  % 1 = on
% Regressions
Regressions = 0; % 0 = off
                  % 1 = on
%% Initialization
if calib == 0
    tsc = 12;
    g=0.0189/tsc;
    sig=0.015/sqrt(tsc);
    rf0=0.0094/tsc;
    phi=0.87^(1/tsc);
    gamma=2;
    B=0;
    verd=0;
    ann=0;
    sig_w = 0.112/sqrt(tsc);
    rhow = 0.2;
end

if calib == 1
    Pars = Model_Calibration;
    tsc = 12;
    g = Pars.g/tsc;
    sig = Pars.sigma/sqrt(tsc);
    rf0 = Pars.rf/tsc;
    phi=Pars.Phi^(1/tsc);
    gamma=2;
```

```
    rhow = 0.2;
    B=0;
    verd=0;
    ann=0;
    sig_w = Pars.sigma_w/sqrt(tsc);
end

PD_Claim_init = PD_Claim;
rho = (-1:.1:1);

S_bar=sig*sqrt(gamma/(1-phi-B/gamma));
s_bar = log(S_bar);
s_max = s_bar + (1-S_bar^2)/2;
S_max = exp(s_max);
delta=exp(gamma*g-.5*((1-phi)*gamma-B)-rf0); % Equation (12) in paper C&C- 1999.

szgrid=10; % +6, with ten we have a total of 16 gridpoints.

ncalc = 100000;             % Number of simulations
bondsel = [1 2 3 4 5 7 10 20]; % Maturity of bonds simulated
maxcb = max(bondsel);
seedval = 123;

chk = 1;
flag2 = 1; % 1 Simulation of yearly data, 0 of quarterly
con = 0;  % Interpolation

%% Grid def
sg = mkgrids(szgrid);
S=exp(sg);

%% PD- & PC-ratio
if Plots
 PD_Claim = 0;
 [output_lnpca ctrindx]=findlpc(sig,g,s_bar);
 PC_ratio=exp(output_lnpca);
 lnpca_pf=output_lnpca;

 PD_Claim = 1;
 [output_lnpda dtrindx]=findlpc(sig,g,s_bar);
 PD_ratio=exp(output_lnpda);
 lnpda_pf=output_lnpda;
end
clear lnpca lnpc
global lnpca lnpc
% reset PD_Claim to initial value we only changed it to make the plot
 PD_Claim = PD_Claim_init;

if PD_Claim == 0
    [output_lnpca ctrindx]=findlpc(sig,g,s_bar);
    lnpca = output_lnpca;
    lnpca_pf = output_lnpca;
else
    [output_lnpda dtrindx]=findlpc(sig,g,s_bar);
    lnpca = output_lnpda;
```

```
    lnpca_pf = output_lnpda;
end
%% Find expected returns and conditional deviations of consumption claim
verd=0;
% Fixed point method
[er_pf elnr_pf sdr_pf sdlnr_pf lnrf_pf lnrf1_pf lny_pf elnrcb_pf sdlnrcb_pf slpmv_pf] = finders(sg);

%% Adjustments of inputs for simulation
dc = 0;
%% Simulation of time-series
[alndctsim_pf astsim_pf alnpctsim_pf alnrtsim_pf alnrfsim_pf asdlnrtsim_pf ...
    alnchpsim_pf alnysim_pf aelnrcbsim_pf asdlnrcbsim_pf atesterfsim_pf aelnrtsim] ...
    =annvars(dc,lnpca_pf,er_pf,elnr_pf,sdr_pf,sdlnr_pf,elnrcb_pf,sdlnrcb_pf,lny_pf,lnrf1_pf);
%% Statistics of interest


if ann == 1
    Edc_pf = tsc*mean(alndctsim_pf);
    Stdc_pf = sqrt(tsc)*std(alndctsim_pf);
else
    Edc_pf = mean(alndctsim_pf);
    Stdc_pf = std(alndctsim_pf);
end

Erf_pf = mean(alnrfsim_pf); % mean log riskfree rate
Stdrf_pf = std(alnrfsim_pf); % sd log RF-rate
Erfinterp_pf = mean(atesterfsim_pf);
Stdrfinterp_pf = std(atesterfsim_pf);

exrett_pf = alnrtsim_pf - alnrfsim_pf; % Excess returns
exrettinterp_pf = alnrtsim_pf - atesterfsim_pf;

Shpr_pf = mean(exrett_pf)/std(exrett_pf); % Sharpe ratio of log returns
ShpR_pf = mean(exp(alnrtsim_pf)-exp(alnrfsim_pf))/std(exp(alnrtsim_pf)- exp(alnrfsim_pf));
Shprinterp_pf = mean(exrettinterp_pf)/std(exrettinterp_pf);
ShpRinterp_pf = mean(exp(alnrtsim_pf)- exp(atesterfsim_pf))/std(exp(alnrtsim_pf)-exp(atesterfsim_pf));
Eexrett_pf = mean(exrett_pf); % Mean excess log returns
Stdexrett_pf = std(exrett_pf); % SD excess log returns)
Eexrettinterp_pf = mean(exrettinterp_pf);
Stdexrettinterp_pf = std(exrettinterp_pf);
Ep_d_pf = mean(alnpctsim_pf); % Log price/consumption
Stdp_d_pf = std(alnpctsim_pf);

if PD_Claim == 0
    PC_Claim_Sim_mom = struct();
    PC_Claim_Sim_mom.MeanConsGrowth   = Edc_pf;
    PC_Claim_Sim_mom.StdConsGrowth    = Stdc_pf;
    PC_Claim_Sim_mom.MeanRiskFreeRate = Erfinterp_pf;
    PC_Claim_Sim_mom.StdRiskFreeRate  = Stdrfinterp_pf;
    PC_Claim_Sim_mom.logSharperatio   = Shprinterp_pf;
    PC_Claim_Sim_mom.Sharperatio      = ShpRinterp_pf;
    PC_Claim_Sim_mom.MeanExcessReturns = Eexrettinterp_pf;
    PC_Claim_Sim_mom.StdExcessReturns = Stdexrettinterp_pf;
    PC_Claim_Sim_mom.MeanPriceDividend = Ep_d_pf;
    PC_Claim_Sim_mom.StdPriceDividend = Stdp_d_pf;
```

```
    PC_Claim_Sim_mom.S_max            = S_max;
    PC_Claim_Sim_mom.S_bar            = S_bar;
    PC_Claim_Sim_mom.delta            = delta^tsc;
    PC_Claim_Sim_mom = struct2table(PC_Claim_Sim_mom);
    writetable(PC_Claim_Sim_mom)
elseif PD_Claim == 1
    PD_Claim_Sim_mom = struct();
    PD_Claim_Sim_mom.MeanDivGrowth    = Edc_pf;
    PD_Claim_Sim_mom.StdDivGrowth     = Stdc_pf;
    PD_Claim_Sim_mom.MeanRiskFreeRate = Erfinterp_pf;
    PD_Claim_Sim_mom.StdRiskFreeRate  = Stdrfinterp_pf;
    PD_Claim_Sim_mom.logSharperatio   = Shprinterp_pf;
    PD_Claim_Sim_mom.Sharperatio      = ShpRinterp_pf;
    PD_Claim_Sim_mom.MeanExcessReturns = Eexrettinterp_pf;
    PD_Claim_Sim_mom.StdExcessReturns = Stdexrettinterp_pf;
    PD_Claim_Sim_mom.MeanPriceDividend = Ep_d_pf;
    PD_Claim_Sim_mom.StdPriceDividend = Stdp_d_pf;
    PD_Claim_Sim_mom.S_max            = S_max;
    PD_Claim_Sim_mom.S_bar            = S_bar;
    PD_Claim_Sim_mom.delta            = delta^tsc;
    PD_Claim_Sim_mom = struct2table(PD_Claim_Sim_mom);
    writetable(PD_Claim_Sim_mom)
end
%% SDF Simulation
rng(24,'twister')
[stsim, vtsim lndctsim lnpctsim lnrtsim lnrfsim ertsim elnrtsim sdrtsim...
    sdlnrtsim elnrcbsim sdlnrcbsim lnysim lnrcbsim testerfsim]=...
    simvars(dc,lnpca_pf,er_pf,elnr_pf,sdr_pf,sdlnr_pf,elnrcb_pf,sdlnrcb_pf,lny_pf ,lnrf1_pf);
% Stochastic discount factor
SDFus = delta*exp(-g*gamma)*exp(-gamma*vtsim).*exp(- gamma*(stsim(2:length(stsim))...
    -stsim(1:length(stsim)-1)));
%% Data Table
if PD_Claim == 0
    PC_Claim_Sim_dat = struct();
    PC_Claim_Sim_dat.S_t        = astsim_pf;
    PC_Claim_Sim_dat.deltac     = alndctsim_pf;
    PC_Claim_Sim_dat.pcratio    = alnpctsim_pf;
    PC_Claim_Sim_dat.ExPostReturns = alnrtsim_pf;
    PC_Claim_Sim_dat.RiskFreeRate = alnrfsim_pf;
    PC_Claim_Sim_dat.Prices     = alnchpsim_pf;
    PC_Claim_Sim_dat.stdReturns = asdlnrtsim_pf;
    PC_Claim_Sim_dat = struct2table(PC_Claim_Sim_dat);
    writetable(PC_Claim_Sim_dat)
elseif PD_Claim == 1
    PD_Claim_Sim_dat = struct();
    PD_Claim_Sim_dat.S_t        = astsim_pf;
    PD_Claim_Sim_dat.deltac     = alndctsim_pf;
    PD_Claim_Sim_dat.pcratio    = alnpctsim_pf;
    PD_Claim_Sim_dat.ExPostReturns = alnrtsim_pf;
    PD_Claim_Sim_dat.RiskFreeRate = alnrfsim_pf;
    PD_Claim_Sim_dat.Prices     = alnchpsim_pf;
    PD_Claim_Sim_dat.stdReturns = asdlnrtsim_pf;
    PD_Claim_Sim_dat = struct2table(PD_Claim_Sim_dat);
    writetable(PD_Claim_Sim_dat)
end
```

```
%% Construction of Indicator of recession
% Load NBER Recession data from 1854-12-01 to 2019-10-01
% The USREC.csv is monthly observed.
% For updated data see

NBER_REC = importdata('USREC.csv');

% Define period yyyy-mm-dd
from = '1950-01-01';
to   = '2018-12-01';

% find indexes
idx_from = find(NBER_REC.textdata(:,1)==string(from)) - 1;
idx_to   = find(NBER_REC.textdata(:,1)==string(to)) - 1;

% Calculate percentage of the time the economy is in recession
rec_emp_percentage = sum(NBER_REC.data(idx_from:idx_to,1)) / length(NBER_REC.data(idx_from:idx_to,1));
% define recession dummey as when s_t < s_bar
rec_sim_ss = NaN(length(astsim_pf), 1);
for i = 1:length(astsim_pf)
    if astsim_pf(i) < s_bar
        rec_sim_ss(i) = 1;
    else
        rec_sim_ss(i) = 0;
    end
end
rec_sim_ss_percentage = sum(rec_sim_ss(:)==1) / length(rec_sim_ss);
%% Matching the empirical density
Rec_s_bar = fzero(@(x) (integral(@q_s,-Inf,x) - rec_emp_percentage), s_bar-0.9);
%% Redefining recession periods in the simulation
% such that the frequency of recession in the simulation corresponds to the
% empirical frequency of recessions:
% Recession s_t < Rec_s_bar
rec_sim_ss = NaN(length(astsim_pf), 1);
for i = 1:length(astsim_pf)
    if astsim_pf(i) < Rec_s_bar
        rec_sim_ss(i) = 1;
    else
        rec_sim_ss(i) = 0;
    end
end
%% Finish
if Plots == 1
    Figures_CC1998;
end

if Tables == 1
    Table_Generator;
end

if calib == 1
    if PD_Claim == 0
        save('Workspaces/PC_Claim_workspace');
    else
        save('Workspaces/PD_Claim_workspace');
```

```
    end
else
    if PD_Claim == 0
        save('Workspaces/CC_PC_Claim_workspace');
    else
        save('Workspaces/CC_PD_Claim_workspace');
    end
end

%%
%load gong
%audioplayer(y,Fs);
%play(ans)
toc
```

Appendix B. Functions called by main script

```
function [sg] = mkgrids(szgrid)
global s_max s_bar
a = exp(s_max)*1;
b = (exp(s_max)*1)/(szgrid+1);
    sg = 0:b:a;
    sg = log(sg(2:end))';

     if max(sg == s_bar) == 0    % Making sure s_bar will be on the grid.
        sg = cat(1,sg,s_bar);
        sg = sort(sg);
    end
    if max(sg == s_max) == 0
        sg = cat(1,sg,s_max);
        sg = sort(sg);
    end

    % Put more density at the beginning of the grid to improve iteration
    % during the fixed point procedure.%
    % ---------------------------------------------------------------------- %
    idens= s_max-[0.01:0.01:0.04]';
    sg=cat(1,sg,idens);
    sg = sort(sg); % Sorting the values present in the grid
end
```

```
function [lnpca, ctrindx]=findlpc(sig,g,s_bar)
% This is the procedure that will calculate the fixed point P / C. %
% ---------------------------------------------------------------------- %
global s sg lnpc delta gamma phi B debug
%% S_bar
%Now we need to find the index of the value of s_bar to be used in
% graphs and other statistics
if max(sg == s_bar) == 1
ctrindx = find(sg == s_bar);
else
disp ('ERROR: The stationary value of log (S) is not in the grid');
end
%% Function value vectors P / C or P / D
lnpca = zeros(size(sg,1),1); % We are starting with PC or PD = 1
```

```
lnpc = lnpca;
newlnpc = lnpc;
%% Loop: find ln (P / C) from the grid of s
iter = 1;
erro = 1;
while iter < 10000 && erro > 1e-6
for i=1:size(sg,1)
s = sg(i);
if exp(-log(delta)+gamma*g-(gamma*(1-phi)-B)/2-B*(s-s_bar)) < g
disp('\t Attention: Rf < g \n');
fprintf('value of st: %g',s);
end
% Generate the log of the variable interest rate in time
newlnpc(i)=log(GaussLegendre(@pdint,abs(sig)*(-8),abs(sig)*8,40) );
debug;
end
tv = max(abs((exp(newlnpc)-exp(lnpc))./exp(newlnpc)));
lnpc = newlnpc;
erro = max(tv);
iter = iter + 1;
end
lnpca = lnpc;
end
```

Note here that an additional file is required! Available here: `http://www.holoborodko.com/pavel/numerical-methods/numerical-integration/`
This is a `.mex` file calling a `c++`-function, greatly increasing the computational efficiency of the integration process. The built-in numerical integrator of MatLab can be used if preferred though.

```
function [ anss, x, w ] = GaussLegendre(f,a,b,n,tol,varargin)
% GAUSSLEGENDRE(f,a,b,n,tol) Fast and precise Gauss-Legendre quadrature.
%
% Approximate definite integral of a function f(x) on the interval [a,b]
% using n-points high precision Gauss-Legendre Quadrature.
%
% Abscissas and weights are calculated with prescribed tolerance or used
% pre-calculated with high precision.
%
% Example 1:   >>GaussLegendre(@cos,-pi/2,pi/2,1024)
%                ans =
%                     2.000000000000000
%
% Example 2:   >>f=inline('cos(x)');
%              >>GaussLegendre(f,-pi/2,pi/2,1024)
%                ans =
%                     2.000000000000000


% Prepare function for sending to MEX
f = fcnchk(f,'vectorized');

% Use default number of nodes and tolerance
```

```
if nargin < 4, n = 256;
end
if nargin < 5, tol = eps * 1e+3;
end

% Calculate integral by ultra-fast native compiled
% code in MEX

if nargout <= 1, anss = quadlab(20,f,a,b,n,tol);
else           [anss,x,w] = quadlab(20,f,a,b,n,tol);
end

end
```

```
function [inside] = pdint(v)
% Will create a new normal density according to the innovations of v {t + 1}
% to integrate P / C functional.
% ---------------------------------------------------------------------- %
global sig debug
inside = (1/(sig*(2*pi)^(.5)))*exp(-.5*(v/sig).^2).*pdmotor(v);
debug(:,3)=inside';
end
```

```
function [inside] = pdmotor(v)
% Procedure to be used for numeric integration when calculating
% the fixed point. It has as argument only v ~ N (0, sig). Returns VALUE of
% P / C or P / D for each iteration over the current value of s {t} in each
% iteration.
% ---------------------------------------------------------------------- %
global delta g gamma s sg lnpc debug PD_Claim rhow sig sig_w

s1=strans(s,v);

if PD_Claim == 0 % Calculating the Consumption Claim P / C ratio
   inside = delta * exp(g*(1-gamma))*exp(-gamma*(s1-s)).*...
            (1+exp(interp(s1,sg,lnpc)))'.*exp((1-gamma)*v);

elseif PD_Claim == 1 % Calculating the Dividend Claim P / D ratio
   inside = delta * exp(g*(1-gamma))*exp(1/2 * (1-rhow^2) * sig_w^2 ) *...
           exp(-gamma * (s1 - s )) .* (1+exp(interp(s1,sg,lnpc)))' .* ...
           exp((rhow * sig_w/sig - gamma)*v);
end
debug(:,2)=inside';
end
```

```
function [news]=strans(s,v)
% Strans procedure                                                 %
% This function will return the value of s {t + 1} = log (S {t + 1}) %
% s {t + 1} = (1-phi) * s_bar + phi * s {t} + lambda (s {t}) * v {t + 1}; %
global s_bar phi debug

news = (1-phi)*s_bar + phi*s + lambda(s) * v;

debug(:,1)=news';
end
```

```
function [fofs indx] = interp(sv,x,fx)
% ---------------------------------------------------------------------------%
% Interpolation procedure of s distribution                          %
% sv -> vector of any values where f (s) is to be generated.         %
% x -> vector of grid points of s. It must be monotonic increasing. %
% fx -> vector of current values of f (x) in the grid.               %
% Find local slope and intercept to use on% grid                     %
% log (S) such that returns f (x) = a + b * x                        %
% ---------------------------------------------------------------------------%

if isempty(min(find(fx == 0))) == 0
   fofs = 0*sv’; else
   T= size(x,1);

   if x(2) < x(1)
      disp(’Not monotonically increasing’);
   end

   if size(sv,2) > 1 && size(sv,1) == 1 sv=sv’;
      chk = 1;

   elseif size(sv,2) > 1 && size(sv,1) > 1
      disp(’ERROR: Not Same size’);

   elseif size(sv,2) == 1 && size(sv,1) > 1
      chk = 0;

   elseif size(sv,2) == 1 && size(sv,1) == 1
      chk = 2;

   end

   gradf = (fx(2:T)-fx(1:T-1))./(x(2:T)-x(1:T-1));
   const = cat(1,(fx(1:T-1)-gradf.*x(1:T-1)),(fx(T)-gradf(T-1)*x(T)));
   const = cat(1,fx(1)-gradf(1)*x(1),const);
   slope = cat(1,gradf(1),gradf);
   slope = cat(1,slope,gradf(T-1));

  % We are interested in finding the smallest vector index (sv (i) -x) so
  % we get an index vector that contains these indexes and we can arrange
  % monotonically the slope and intercept vectors.

   indx = zeros(size(sv,1),1);

   for i = 1 : size(sv,1)

      if sv(i)> x(1)
         indx(i)= max(find((sv(i)-x) > 0))+1;
      else
         indx(i)=1;
      end
   end

   fofs = const(indx)+ slope(indx).*sv;
   if chk == 0
```

```
        fofs = fofs';
    end
end
end
```

```
function [er elnr sdr sdlnr lnrf lnrf1 lny elnrcb sdlnrcb slpmv] = finders(sg)

%Procedure that calculates expected returns on consumer assets. elenos will
% provide E (R), SD (R), lnrf, the mean boundary curvature of the variance
% given by the variable (slpmv) and integrate the Sharpe ratio of the
% consumer asset vector.                                              %
% ---------------------------------------------------------------------- %

global g gamma sig phi s maxcb s_bar delta tsc lnpcb matur PD_Claim

% Medium-Variance Boundary Slope                        %
%                                                                 %
% slpmv = (exp((gamma*sig)^2.*(1+lambda(sg)).^2)-1).^.5           %
% ---------------------------------------------------------------------- %

slpmv = (exp((gamma*sig)^2*(1+lambda(sg)).^2)-1).^(0.5);

%% Term interest rate structure given by:                     %
%                                                                 %
% lnrf = -ln(delta) + gamma*g -                                   %
% gamma*(1-phi)*(s{t}-s_bar)-.5((gamma*sig)^2)*(1+lambda(s{t}))^2
% - B*(sg - s_bar) --> Vari?vel no estado
% ---------------------------------------------------------------------- %

lnrf = -log(delta) + gamma*g - gamma*(1-phi)*(sg-s_bar)...
    - 0.5*(gamma*sig*(1+lambda(sg))).^2;

%% Bonds
% Matrix of all bond prices. Your dimension will be N(sg) x (maxcb*tsc)
lnpcb = [];
lnpcb(:,1) = -lnrf;

lnp = zeros(size(sg,1),1);

for j = 2:(maxcb*tsc)
    for i = 1:length(sg)
        s = sg(i);
        lnp(i) = log(GaussLegendre(@intpcb,abs(sig)*(-8),abs(sig)*8,40));
    end
    lnpcb = cat(2,lnpcb,lnp);
end

% Yields
lny = - ...
    lnpcb./kron(ones(size(sg,1),1),linspace(1/tsc,(maxcb*tsc)/tsc,(maxcb*tsc)));

%% Expected Returns and Standard Deviations                        %
% ---------------------------------------------------------------------- %

lnrf1 = zeros(size(sg,1),1);
```

```
er = zeros(size(sg,1),1);
elnr = zeros(size(sg,1),1);
sdr = zeros(size(sg,1),1);
sdlnr = zeros(size(sg,1),1);
elnrcb = zeros(size(sg,1),maxcb*tsc);   % zero-coupon bonds %
sdlnrcb = zeros(size(sg,1),maxcb*tsc);

for i=1:size(sg,1)
    s = sg(i);


    lnrf1(i)= - log(GaussLegendre(@intemrs,abs(sig)*(-8),abs(sig)*8,40));
    if PD_Claim == 0
    er(i)= GaussLegendre(@inter,abs(sig)*(-8),abs(sig)*8,40);
    sdr(i) = GaussLegendre(@inter2,abs(sig)*(-8),abs(sig)*8,40);
    else
    er(i) = GaussLegendre(@interd,abs(sig)*(-8),abs(sig)*8,40);
    sdr(i) = GaussLegendre(@inter2d,abs(sig)*(-8),abs(sig)*8,40);
    end
    elnr(i)= GaussLegendre(@intelnr,abs(sig)*(-8),abs(sig)*8,40);
    sdr(i) = (sdr(i) - er(i).^2).^(.5);
    sdlnr(i) = GaussLegendre(@intelnr2,abs(sig)*(-8),abs(sig)*8,40);

    % Bonds
    matur = maxcb*tsc; elnrcb(i,1) = lnrf(i);

    for k = 2:matur
        elnrcb(i,k) = GaussLegendre(@intelnrcb,abs(sig)*(-8),abs(sig)*8, 40);
        sdlnrcb(i,k) = GaussLegendre(@intelnr2,abs(sig)*(-8),abs(sig)*8, 40);
        sdlnrcb(i,k) = (sdlnrcb(i,k) - elnrcb(i,k).^2).^(.5);
    end

end
end
```

```
function [sg] = mkgrids(szgrid,flag)
% Will build s grid efficiently %
% ---------------------------------------------------------------------- %
global s_max S_max s_bar
if flag == 0
    sg = linspace(0,S_max,szgrid);
    aux = [(sg(szgrid)-.01) (sg(szgrid)-.02) (sg(szgrid)-.03) (sg(szgrid)-.04)];
    sg = cat(2,sg,aux);
    sg = sort(sg);
    sg = log(sg(2:size(sg,2)))';

    if max(sg == s_bar) == 0    % Making sure s_bar will be on the grid.
        sg = cat(1,sg,s_bar);
        sg = sort(sg);
    end
    if max(sg == s_max) == 0
        sg = cat(1,sg,s_max);
        sg = sort(sg);
    end
```

```
    % Put more density at the beginning of the grid to improve iteration
    % during the fixed point procedure.%
    % ------------------------------------------------------------------------ %
    idens=log([.0005 0.0015 .0025 .0035 .0045])’;
    sg=cat(1,sg,idens);
    sg = sort(sg); % Sorting the values present in the grid
end
% Grid 3 da Wachter (2005)
if flag == 1
    sg = linspace(0,S_max,szgrid);
    sg = log(sg(2:size(sg,2)))’;
    if max(sg == s_bar) == 0
        sg = cat(1,sg,s_bar);
        sg = sort(sg);
    end
    if max(sg == s_max) == 0
        sg = cat(1,sg,s_max);
        sg = sort(sg);
    end
    u=min(sg);
    aux = linspace(-300,u,200)’;
    sg = cat(1,sg,aux);
    sg = sort(sg);
end
end
```

```
function out = intpcb(v)

% Function that provides the price of bonds for each maturity.     %

global s sg lnpcb sig

out = pdf(’norm’,v,0,sig).*mrsinsd(v).*...
    exp(interp(strans(s,v),sg,lnpcb(:,size(lnpcb,2))))’;

end
```

```
function [out]=intemrs(v)

% It will return the marginal rate of substitution in such a way that it
% can be used for fixed point integration.                           %
% ------------------------------------------------------------------------ %

global sig
out = (1/(sig*(2*pi)^(.5)))*exp(-.5*(v/sig).^2).*mrsinsd(v);
end
```

```
function [out]=mrsinsd(v)
% Returns the marginal rate of intertemporal substitution in the model. %
% -------------------------------------------------------------------------
global delta g gamma s
out = delta*exp(-gamma*g)*exp(-gamma*v).*exp(-gamma*(strans(s,v)-s));
end
```

```
function [out] = inter(v)
```

```
% Provides the expected return value according to a normal distribution.
% ------------------------------------------------------------------------ %

global sig
out = (1/(sig*(2*pi)^(.5)))*exp(-.5*(v/sig).^2).*erinsd(v);
end
```

```
function [er]=erinsd(v)
% Procedure for calculating consumption claim returns.              %
% ------------------------------------------------------------------------ %

global sg s lnpca g
er=((1+exp(interp(strans(s,v),sg,lnpca)))'./(ones(size(v))...
*exp(interp(s,sg,lnpca)))).*exp(g+v);

end
```

```
function [out] = inter2(v)

% Expected variance returns consumption claim
% ------------------------------------------------------------------------ %

global sig
out = (1/(sig*(2*pi)^(.5)))*exp(-.5*(v/sig).^2).*erinsd(v).^2;
end
```

```
function [inside] = interd(v)
% ------------------------------------------------------------------------%
% Numerical integration expected returns P/D                      %
% ------------------------------------------------------------------------%
inside = internorm(v) .* erdinsd(v);
end
```

```
function [inside] = erdinsd(v)
global g s sg rhow sig sig_w lnpca
% ------------------------------------------------------------------------%
% Numerical integration expected returns P/D                        %
% ------------------------------------------------------------------------%
inside = (1 + exp(interp(strans(s,v),sg,lnpca)'))...
                         ./exp(interp( s, sg, lnpca)') .* exp(g) .*...
              exp(rhow.* sig_w./ sig .* v).* exp(1/2 * (1- rhow^2)* sig_w^2);
end
```

```
function [out] = inter2d(v)

% Expected variance returns consumption claim
% ------------------------------------------------------------------------ %

global sig g rhow sig_w lnpca sg
out = internorm(v) .* erd2ind(v);
end
```

```
function [inside] = internorm(v)
global sig
```

```
% ------------------------------------------------------------------------%
% Density of a normal                                                     %
% ------------------------------------------------------------------------%
inside = 1/((2*pi)^(1/2) .* sig) .* exp(-(v .^2)/(2 * sig ^2));
end
```

```
function [out] = erd2ind(v)

% Expected variance returns consumption claim
% ----------------------------------------------------------------------- %

global sig g rhow sig_w lnpca sg s
out = (1 + exp(interp(strans(s,v), sg, lnpca)))...
                         ./exp(interp(s,sg,lnpca)) .* exp(g) .*...
              exp( rhow.* sig_w./ sig.*v).*exp((1- rhow^2) * sig_w^2)^2;

end
```

```
function [out] = intelnr(v)

% Provides the expected return value according to a normal distribution.
% ----------------------------------------------------------------------- %

global sig
out = (1/(sig*(2*pi)^(.5)))*exp(-.5*(v/sig).^2).*log(erinsd(v));
end
```

```
function [out] = intelnr2(v)

% Provides the expected return value according to a normal distribution.
% ----------------------------------------------------------------------- %

global sig
out = (1/(sig*(2*pi)^(.5)))*exp(-.5*(v/sig).^2).*log(erinsd(v)).^2;
end
```

```
function out = intercb(v)
% Integrating the expected returns of public securities or        %
% the termstructure                                               %

global sig

out = pdf('norm',v,0,sig).*log(ercbin(v));

end
```

```
function out = ercbin(v)
%      Expected Returns of the Term Structure                          %
global s sg lnpcb matur
out = exp(interp(strans(s,v),sg,lnpcb(:,matur-1))) ./ ...
   exp(interp(s,sg,lnpcb(:,matur)));
end
```

```
function [alndctsim astsim alnpctsim alnrtsim alnrfsim asdlnrtsim alnchpsim ...
   alnysim aelnrcbsim asdlnrcbsim atesterf
       aelnrtsim]=annvars(dc,lnpc,er,elnr,sdr,sdlnr,elnrcb,sdlnrcb,lny,lnrf1)
```

```
% Annualising and preparing data from the simulation "simvars.m". Returns
% various series of interest. Returns, PC and DC ratio, std, bond returns
% etc.

global tsc bondsel ann ncalc
% Simulating series
[stsim vtsim lndctsim lnpctsim lnrtsim lnrfsim ertsim elnrtsim sdrtsim...
    sdlnrtsim elnrcbsim sdlnrcbsim lnysim lnrcbsim
         testerf]=simvars(dc,lnpc,er,elnr,sdr,sdlnr,elnrcb,sdlnrcb,lny,lnrf1);

T = size(stsim,1);

%% Consumption
if ann == 1
    alndctsim=lndctsim;
else
    alnctsim = cumsum(lndctsim);
    % Monthly logs
    alnctsim = log(chgfreq(exp(alnctsim),tsc,tsc,0));
    alndctsim = alnctsim(2:size(alnctsim,1))-alnctsim(1:(size(alnctsim,1)- 1));

end

%% s_t
if T > 1
    astsim = chgfreq(stsim(2:T),1,tsc,0);
    astsim = astsim(2:size(astsim,1));
end

%% P/C Ratio
if size(lnpctsim,1) > 1
    alnpctsim = chgfreq(lnpctsim(2:T),1,tsc,0)-log(tsc);
    alnpctsim = alnpctsim(2:size(alnpctsim,1));
end

%% Yearly Returns
if size(lnrtsim,1) > 1
    alnrtsim = chgfreq(lnrtsim,tsc,tsc,0);
    alnrtsim = alnrtsim(2:size(alnrtsim,1));
end

% Expected returns:
if size(elnrtsim,1) > 1
    aelnrtsim = chgfreq(elnrtsim,tsc,tsc,0);
    aelnrtsim = aelnrtsim(2:size(aelnrtsim,1));
end

% Risk free rate
if size(lnrfsim,1) > 1
    alnrfsim = chgfreq(lnrfsim(1:T-1),tsc,tsc,0);
    alnrfsim = alnrfsim(2:size(alnrfsim,1));
end
% Interpolation
if size(testerf,1) > 1
    atesterf = chgfreq(testerf(1:T-1),tsc,tsc,0);
```

```
    atesterf = atesterf(2:size(atesterf,1));
end
%% Conditional deviation of returns
if size(sdlnrtsim,1) > 1
    asdlnrtsim = chgfreq(sdlnrtsim,tsc,tsc,0);
    asdlnrtsim = asdlnrtsim(2:size(asdlnrtsim,1));
end
%% Price evolution
if size(lnpctsim,1) > 1
    lnchpsim = lnpctsim(2:T)-lnpctsim(1:T-1)+lndctsim;
    alnchpsim = chgfreq(lnchpsim,tsc,tsc,0);
    alnchpsim = alnchpsim(2:size(alnchpsim,1));
end
%% Yields
if size(lnysim,1) > 1
    for i=1:length(bondsel)+1
     alnysim(:,i) = chgfreq(lnysim(1:T-1,i),tsc,tsc,0);
    end
    alnysim = alnysim(2:size(alnysim,1),:);
end
%% Bonds
% Mean returns
if size(elnrcbsim,1) > 1
    for i=1:length(bondsel)+1
       aelnrcbsim(:,i) = chgfreq(elnrcbsim(1:T-1,i),tsc,tsc,0);
    end

    aelnrcbsim = aelnrcbsim(2:size(aelnrcbsim,1),:);
end

% Deviations
if size(sdlnrcbsim,1) > 1
    for i=1:length(bondsel)
        asdlnrcbsim(:,i) = chgfreq(sdlnrcbsim(1:T-1,i+1),tsc,tsc,0);
    end

    asdlnrcbsim = asdlnrcbsim(2:size(asdlnrcbsim,1),:);
end
end
```

```
%{
     USAGE:

        chgfreq(returns, horizon , freqency, offset);

     1) offset = 0, frequency = 1:

     Takes monthly end-of month returns, in units log(R) so they may be added.
     returns monthly observations of
     k month overlapping log returns, dated as of last date.
     e.g  if r(t) is the return to end of month t, the program takes

     r(1)        0                         = ro(1)
     r(2)        0                         = ro(2)
     ..          ..
```

```
     r(k)   -->   r(1)+r(2)..+r(k) = ro(k)
     r(k+1)       r(2)+r(3)..+r(k+1) = ro(k+1)
     ..           ..                             ..
     r(T)         r(T-k+1)+..+r(T) = ro(T)

     NOTE returns SUMS not averages.

     2) frequency > 1, offset .ne. 0:

     samples every frequency points, starting at the frequency + offest'th.
     e.g. for freq = 3, o = 2,

     ro(1)   -->   rf(1)
     ro(2)
     ro(3)
     -----
     ro(4)   -->   rf(2)
     ro(5)
     ro(6)
       ..

    EXAMPLE: from monthly data:
    to create quarterly data, with Q1 return = jan+feb+march,
    use horizon = 3; freqency = 3, offset = 0;
    to create quarterly data, with Q1 return = nov+dec+jan,
    use horizon = 3; frequency = 3, offset = 2;
    to create quarterly observations of annual returns, with Q1 = feb+..+jan,
    use horizon = 12, frequency = 3, offset = 2;
    to create quarterly data, with Q1 data = march;
    use horizon = 1, frequency = 3, offset = 0;

    from quarterly data:
    to create annual data
    use horizon = 4, frequency = 4, offset = 0;
    to create quarterly observations of annual averages
    use horizon = 4, frequency = 1, offset = 0
%}
function [ro] = chgfreq(rm,k,f,o)
T = size(rm,1);
ro = rm;

if k > 1
    bigr = rm(1:T-k+1,:);
    i = 1;
    while i <= k-1
        bigr = bigr+rm(1+i:T-k+1+i,:);
        i = i+1;
    end
    ro = cat(1,(-99*ones(k-1,1))*ones(1,size(bigr,2)),bigr);
end

if f > 1
    mask = zeros(size(ro,1),1);
    i = 1;
    while i <= size(ro,1)
```

```
        if (f*i-o) <= size(ro,1)
           mask(f*i-o) = 1;
        end
        i = i+1;
    end
end
ro = selif(ro,mask);
end
```

```
function [stsim vtsim lndctsim lnpctsim lnrtsim lnrfsim ertsim elnrtsim sdrtsim...
    sdlnrtsim elnrcbsim sdlnrcbsim lnysim lnrcbsim testerf
        erd]=simvars(dc,lnpca,er,elnr,sdr,sdlnr,elnrcb,sdlnrcb,lny,lnrf1)

%
% This routine simulates the most important time-series of this model %
% from a chosen calibration                                           %
% Simulating:                                                         %
% - s=log(S);                                                         %
% - P/C;                                                              %
% - R{t+1};                                                           %
% - E{t}[R{t+1}];                                                     %
% - SD{t}[R{t+1}];                                                    %
% - Rf{t+1};                                                          %
% - corr(Rf{t+1},Cons_t)                                              %
% - Bonds                                                             %
%                                                                     %

global ncalc gamma sig sig_w g phi delta s_max s_bar sg maxcb tsc bondsel...
    PD_Claim rhow g

%% initialization
if dc == 0
    T=ncalc;
    rng(24,'twister');
    vtsim = sig*randn(T,1);
    wtsim = rhow * sig_w / sig * vtsim + sig_w * (1 - rhow ^2) ^ 0.5 * randn(T,1);
    if PD_Claim == 0
        lndctsim = g + vtsim;
    else
        lndctsim = g + wtsim;
    end
else
    if min(dc) <= 0

        disp ('simvars: You entered the consumption growth log.');
        disp ('You need to enter consumption growth data');
        disp ('gross, ie neither log nor net growth.');

    end
    T = length(dc);
    lndctsim = log(dc);
end

%% Simulation of log(S_t)

```

```
stsim = zeros(T+1,1);

stsim(1) = s_bar;        % Starting the process at SS

% if PD_Claim == 0
    for i=2:T+1
        if strans(stsim(i-1),vtsim(i-1)) <= s_max
            stsim(i) = strans(stsim(i-1),vtsim(i-1));
        else
            stsim(i)=(1-phi)*s_bar+phi*stsim(i-1);
        end
    end
% else
%     for i=2:T+1
%         if strans(stsim(i-1),wtsim(i-1)) <= s_max
%             stsim(i) = strans(stsim(i-1),vtsim(i-1));
%         else
%             stsim(i)=(1-phi)*s_bar+phi*stsim(i-1);
%         end
%     end
% end
%% PC ratio                                         %
lnpctsim = interp(stsim,sg,lnpca)';
%% ex-post Returns                                                    %
%                                                                     %
%                       R = (C'/C){(1+(P/C)')/(P/C)}                  %
% ------------------------------------------------------------------- %
lnrtsim = log(1+exp(lnpctsim(2:T+1))) - lnpctsim(1:T) + lndctsim;
    %%
%% potential time varying RF-rate

lnrfsim = -log(delta) + gamma*g - gamma*(1-phi)*(stsim-s_bar)...
    - 0.5*(gamma*sig*(1+lambda(stsim))).^2;

%% Expected Returns and Expected deviations

testerf = interp(stsim,sg,lnrf1)';
ertsim = interp(stsim,sg,er)';
elnrtsim = interp(stsim,sg,elnr)';
sdrtsim = interp(stsim,sg,sdr)';
sdlnrtsim = interp(stsim,sg,sdlnr)';

%% Treasury Bill return 90 days

elnrcbsim = interp(stsim,sg,elnrcb(:,1))'; % Expected Returns on Tbill
sdlnrcbsim = interp(stsim,sg,sdlnrcb(:,1))';
lnysim = interp(stsim,sg,lny(:,1))';   % Bond yields
lny2sim = zeros(size(lnysim,1),1);

for i = 2:(maxcb*tsc)
    if find(i == bondsel*tsc)

        lnysim = cat(2,lnysim,interp(stsim,sg,lny(:,i))');
        lny2sim = cat(2,lny2sim,interp(stsim,sg,lny(:,i-1))');
        elnrcbsim = cat(2,elnrcbsim,interp(stsim,sg,elnrcb(:,i))');
```

```
        sdlnrcbsim = cat(2,sdlnrcbsim,interp(stsim,sg,sdlnrcb(:,i))');

    end
end

% Returns on bonds with maturities = [1 2 4 8 12 16 20]

lnrcbsim = cat(1,0,lnysim(1:T-1,1));
for i = 2:length(bondsel)
    lnrcbsim = cat(2,lnrcbsim,cat(1,0,(-lny2sim(2:T,i)*((bondsel(i)- 1/tsc))+...
        lnysim(1:T-1,i)*bondsel(i))/tsc));
end
end
```

```
function [stsim vtsim lndctsim lnrfsim]=simulacorr(rho)

global ncalc gamma sig g phi delta B s_bar seedval

%% Simulation of shocks from the correlationcoefficient rho

T=ncalc;
randn('seed',seedval);
x = sig*randn(T,1);
y = sig*randn(T,1);

vtsim = rho*x + sqrt(1-rho^2)*y;
lndctsim = g + vtsim;

%% Simulation log state (Surplus consumption ratio)

stsim = zeros(T+1,1);

stsim(1) = s_bar;

for i=2:T+1

    stsim(i) = strans(stsim(i-1),vtsim(i-1));

end

%% Time variying log RF-rate

lnrfsim = -log(delta)+gamma*g-(gamma*(1-phi)-B)/2-B .*(stsim-s_bar);

end
```

```
function [y]=lambda(s)
% function Lambda
global S_bar s_bar s_max verd
if verd == 0
    if (double(s) <= s_max)
        y = (1 / S_bar).*sqrt(max(0 , 1-2.*(s-s_bar)))-1;
    else
        y = 0;
    end
```

```
elseif verd == 1
   y=(1-S_bar)/S_bar;
end
end
```

```
function [y]=lambda_Helper(s)
% function Lambda
global S_bar s_bar s_max verd
       y = (1 / S_bar).*sqrt(max(0 , 1-2.*(s-s_bar)))-1;
end
```

```
function xs=selif(x,t)
% SELIF
% SELIF(x,t) selects the elements of x for which t=1
% x: NxK matrix, t: Nx1 matrix of 0's and 1's
nt=repmat(t,1,size(x',1));
nx=x.*nt;
xs=nx(nx~=0);
col=size(x',1);
xs=reshape(xs,size(xs,1)/col,col);
```

```
function [q] = q_s(s)
%Density of s (surplus consumption) continous time
global s_max
q = z_s(s)./integral(@z_s,-Inf,s_max);
end
```

```
function [z] = z_s(s)
% Functional Z of s, for the theoretical density of S
global gamma S_bar sig
lnZ = -gamma .* S_bar^2 .* ((S_bar^(-2)-1)./(lambda(s))+3.*lambda(s)+(lambda(s).^2)./2) - (gamma .* (3 .*
    S_bar^2-1) +2) .* log(lambda(s))-2*log(sig);
z = exp(lnZ);
end
```

```
function results=nwest(y,x,nlag)
% PURPOSE: computes Newey-West adjusted heteroscedastic-serial
%          consistent Least-squares Regression
%---------------------------------------------------
% USAGE: results = nwest(y,x,nlag)
% where: y = dependent variable vector (nobs x 1)
%        x = independent variables matrix (nobs x nvar)
%     nlag = lag length to use
%---------------------------------------------------
% RETURNS: a structure
%        results.meth = 'newlyw'
%        results.beta = bhat
%        results.tstat = t-stats
%        results.yhat = yhat
%        results.resid = residuals
%        results.sige = e'*e/(n-k)
%        results.rsqr = rsquared
%        results.rbar = rbar-squared
%        results.dw   = Durbin-Watson Statistic
%        results.nobs = nobs
```

```
%       results.nvar = nvars
%       results.y    = y data vector
% --------------------------------------------------
% SEE ALSO: nwest_d, prt(results), plt(results)
%---------------------------------------------------
% References: Gallant, R. (1987),
%  "Nonlinear Statistical Models," pp.137-139.
%---------------------------------------------------

% written by:
% James P. LeSage, Dept of Economics
% University of Toledo
% 2801 W. Bancroft St,
% Toledo, OH 43606
% % jlesage@spatial-econometrics.com



if (nargin ~= 3); error('Wrong # of arguments to nwest'); end;

[nobs nvar] = size(x);

results.meth   = 'nwest';
results.y      = y;
results.nobs   = nobs;
results.nvar   = nvar;

xpxi = inv(x'*x);
results.beta   = xpxi*(x'*y);
results.yhat   = x*results.beta;
results.resid  = y - results.yhat;
sigu = results.resid'*results.resid;
results.sige   = sigu/(nobs-nvar);

%pd = fitdist(Y, 'normal');
%results.LLH    = pd.NLogL;


% perform Newey-West correction
emat = [];
for i=1:nvar;
emat = [emat
       results.resid'];
end;

    hhat=emat.*x';
    G=zeros(nvar,nvar); w=zeros(2*nlag+1,1);
    a=0;

    while a~=nlag+1
        ga=zeros(nvar,nvar);
        w(nlag+1+a,1)=(nlag+1-a)/(nlag+1);
        za=hhat(:,(a+1):nobs)*hhat(:,1:nobs-a)';
          if a==0;
           ga=ga+za;
```

```
          else
           ga=ga+za+za';
          end;
        G=G+w(nlag+1+a,1)*ga;
        a=a+1;
    end % end of while

        V=xpxi*G*xpxi;
        nwerr= sqrt(diag(V));

results.tstat = results.beta./nwerr; % Newey-West t-statistics
results.nwerr = nwerr;
ym = y - ones(nobs,1)*mean(y);
rsqr1 = sigu;
rsqr2 = ym'*ym;
results.rsqr = 1.0 - rsqr1/rsqr2; % r-squared
rsqr1 = rsqr1/(nobs-nvar);
rsqr2 = rsqr2/(nobs-1.0);
results.rbar = 1 - (rsqr1/rsqr2); % rbar-squared
ediff = results.resid(2:nobs) - results.resid(1:nobs-1);
results.dw = diag((ediff'*ediff)./(sigu))'; % durbin-watson
```

Appendix C. Analysis code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% RUNNING THIS FILE PRODUCES MOST THE TABLES AND FIGURES IN THE ARTICLE
% JENSEN & KROGH (2019).
% The program relies on running the MAIN.m file according to the
% description in that file. This is because this program is working upon
% the 4 workspaces saved when running the MAIN.m program.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
addpath('Functions');
addpath('Data');
addpath('Workspaces');
addpath('Calibration');
addpath('Figures');
addpath('Tables');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Loads simulated data and performs regressions %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear
clc
opts.Colors    = get(groot,'defaultAxesColorOrder');

%%
Save_Figures = 0;       % 0 = dont save
                        % 1 = save

load('PC_Claim_workspace','s_bar','s_max',...
        'verd','S_bar','sig','gamma','S','astsim_pf','alnrtsim_pf','stsim');

%% Matching the empirical receession probability with the models density of s_t
NBER_REC = importdata('USREC.csv');
% Define period yyyy-mm-dd
```

```
from = '1950-01-01';
to   = '2018-12-01';

% find indexes
idx_from = find(NBER_REC.textdata(:,1)==string(from)) - 1;
idx_to   = find(NBER_REC.textdata(:,1)==string(to)) - 1;

% Calculate percentage of the time the economy is in recession
rec_emp_percentage = sum(NBER_REC.data(idx_from:idx_to,1)) / length(NBER_REC.data(idx_from:idx_to,1));
rec_emp_percentagetotal = sum(NBER_REC.data(1:end,1)) / length(NBER_REC.data(1:end,1));

labeledMatrix = bwlabel(NBER_REC.data(idx_from:idx_to,1));
measurements = regionprops(labeledMatrix, 'Area');
RecessionLengths = [measurements.Area];
mean(RecessionLengths)/12;
%%
s_bar_2 = log(0.02); % Recession specification by figure
Rec_s_bar = fzero(@(x) (integral(@q_s,-Inf,x) - rec_emp_percentage), s_bar-0.1);
Model_Rec = integral(@q_s,-Inf,s_bar);
Model_Rec_2 = integral(@q_s,-Inf,s_bar_2);
Match_Rec = integral(@q_s,-Inf,Rec_s_bar);
%%
load('PC_Claim_workspace','astsim_pf');astsim = astsim_pf;
[heights location] = hist(astsim, 75);
width = location(2) - location(1);
heights = heights / (size(astsim, 1) * width);
%%
warning('off','all'); % fplot doesnt like the integral functions
figure;
barplot = bar(location, heights,'hist');
barplot.FaceColor = [0, 0.4470, 0.7410];
hold on
fplot(@q_s, [min(log(S)-0.5) s_max+0.25],'Color',[0.8500, 0.3250, 0.0980],'LineWidth',2.5);%title('Stationary
    Distribution of s');
hold on
xline(Rec_s_bar,'--','$\bar{s}_{rec}$','Interpreter','latex','FontSize',18);
hold on
xline(s_bar,'--','$\bar{s}$','Interpreter','latex','FontSize',18);
xline(s_bar_2,'--','$\bar{s}_{2,rec}$','Interpreter','latex','FontSize',18);
xlim([min(log(S))-0.5 -2]);
legend('Histogram','Theoretical Density','Location','northwest')
hold off
if Save_Figures
saveas(gcf,'../Figures/DistributionS_t','epsc')
end
%% Risk aversion plot
RA = gamma./exp(stsim);
mRA = mean(RA);
figure;
plot(RA);ylabel('$\gamma/S_t,\qquad$ $\gamma=2$','Interpreter','latex');
xlabel('$t$','interpreter','latex');
yline(mean(RA),'--','LineWidth',2);
legend('$\gamma/S_t$: Risk Aversion',['$\mathbf{E}\gamma/S_t$=',num2str(mRA)],'interpreter','latex')
xlim([0 100000]);
if Save_Figures
```

```
saveas(gcf,'../Figures/RA','epsc')
end
max(RA)
%% Redefining recession periods in the simulation
% such that the frequency of recession in the simulation corresponds to the
% empirical frequency of recessions:
% Recession s_t < Rec_s_bar
load('PC_Claim_workspace','stsim');astsim = stsim;
rec_sim_ss = NaN(length(astsim), 1);
for i = 1:length(astsim)
    if astsim(i) < Rec_s_bar
        rec_sim_ss(i) = 1;
    else
        rec_sim_ss(i) = 0;
    end
end
rec_sim_ss_percentage = sum(rec_sim_ss(:)==1) / length(rec_sim_ss);
%%
labeledMatrix = bwlabel(rec_sim_ss);
measurements = regionprops(labeledMatrix, 'Area');
RecessionLengths = [measurements.Area];
mean(RecessionLengths)
%%
    load('PD_Claim_workspace','s_bar','s_max',...
        'verd','S_bar','sig','gamma','S','stsim','lnrtsim','lnpctsim','Erfinterp_pf');
    Erfinterp_pf = Erfinterp_pf./12;
    PD_regress = lnpctsim(2:end,1);          % PD
    lnrtsim_PD = lnrtsim;
    load('PC_Claim_workspace','lnrtsim','lnpctsim')
    lnrtsim_PC = lnrtsim;
    PC_regress = lnpctsim(2:end,1);          % PC
    rec_sim_ss = rec_sim_ss(2:end,1);
%%
rfr  = Erfinterp_pf;                    % Risk free rate
Erets_PC = lnrtsim_PC - rfr;            % Excess Returns PC
Erets_PD = lnrtsim_PD - rfr;
h    =  1;                                % Forecast Horizon 0 = in-sample regression
yPC  = Erets_PC(1+h:end,1);               % Regressand PC
yPD  = Erets_PD(1+h:end,1);               % Regressand PD
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%   No business cycle regressions %%%
%%% r_(t+h) = alpha + beta p/d_t + eps %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
x   = [ones(length(PD_regress(1:end-h,1)), 1),...
    PD_regress(1:end-h,1)];
regPDnorec = nwest(yPD,x,0);

x   = [ones(length(PC_regress(1:end-h,1)), 1), ...
    PC_regress(1:end-h,1)];
regPCnorec = nwest(yPC,x,0);

regsNB = [regPCnorec regPDnorec];
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%                   Business cycle regressions               %%%
%%% r_(t+h) = alpha + beta_1 p/d_t*I_rec + beta_2(1-I_rec)p/d_t + eps %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                         Full Business cycle                        %
x   = [ones(length(PD_regress(1:end-h,1)), 1), ...
    rec_sim_ss(1:end-h,:) .* PD_regress(1:end-h,1), ...
    (1-rec_sim_ss(1:end-h,:)) .* PD_regress(1:end-h,1)];
regPDrec = nwest(yPD,x,0); %% Full BC <- PD

x   = [ones(length(PC_regress(1:end-h,1)), 1), ...
    rec_sim_ss(1:end-h,:) .* PC_regress(1:end-h,1), ...
    (1-rec_sim_ss(1:end-h,:)) .* PC_regress(1:end-h,1)];
regPCrec = nwest(yPC,x,0); %% Full BC <- PC

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                         Split Business cycle                       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lower_Sbar = 0; %% Table purposes only do not change

retsHRecPD = yPD .* rec_sim_ss(2:end); %% Excess Returns Recession
retsHExpPD = yPD .* (1-rec_sim_ss(2:end)); %% Exceess Returns Expansions
retsHRecPC = yPC .* rec_sim_ss(2:end); %% Excess Returns Recession
retsHExpPC = yPC .* (1-rec_sim_ss(2:end)); %% Exceess Returns Expansions

PDRegHRec = rec_sim_ss(1:end-h,:) .* PD_regress(1:end-h,1);
PDRegHExp = (1 - rec_sim_ss(1:end-h,:)) .* PD_regress(1:end-h,1);
PCRegHRec = rec_sim_ss(1:end-h,:) .* PD_regress(1:end-h,1);
PCRegHExp = (1 - rec_sim_ss(1:end-h,:)) .* PC_regress(1:end-h,1);

a = [retsHRecPD, PDRegHRec];
a = a(all(a,2),:);
ExcRetsRec = a(:,1);               %% <- Excess Returns Recessions only
ExRetsRecPDFC = ExcRetsRec;
PDrecHR = [ones(size(a,1), 1) a(:,2)]; %% <- PD recession
regPDrec1 = nwest(ExcRetsRec,PDrecHR,0);

a = [retsHExpPD, PDRegHExp];
a = a(all(a,2),:);
ExRetsExp = a(:,1); %% <- Excess Returns Expansions only
ExRetsExpPDFC = ExRetsExp;
PDexpHR  = [ones(size(a,1),1) a(:,2)]; %% <- PD Expansion
regPDexp1 = nwest(ExRetsExp,PDexpHR,0);

a = [retsHExpPC, PCRegHExp];
a = a(all(a,2),:);
ExRetsExp = a(:,1);
ExRetsExpPCFC = ExRetsExp;
PCExpHR  = [ones(size(a,1),1) a(:,2)];
regPCexp1 = nwest(ExRetsExp,PCExpHR,0);

a = [retsHRecPC, PCRegHRec];
a = a(all(a,2),:);
ExRecRets = a(:,1);
ExRetsRecPCFC = ExRecRets;
PCrecHR  = [ones(size(a,1),1) a(:,2)];
```

```matlab
regPCrec1 = nwest(ExRecRets,PCrecHR,0);

regs1 = [regPCrec regPDrec regPCrec1 regPCexp1 regPDrec1 regPDexp1];
if Save_Figures
RegressionTable2;
end
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                   Split Business cycle lower s bar               %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load('PD_Claim_workspace','s_bar','s_max',...
       'verd','S_bar','sig','gamma','S','stsim','lnrtsim','lnpctsim','Erfinterp_pf');
    Erfinterp_pf = Erfinterp_pf./12;
    PD_regress  = lnpctsim(2:end,1);          % PD
    lnrtsimPD   = lnrtsim;
    load('PC_Claim_workspace','lnrtsim','lnpctsim')
    alnrtsim_pf = lnrtsim;
    PC_regress = lnpctsim(2:end,1);           % PC
    lnrtsimPC  = lnrtsim;
    h=1;
rfr  = Erfinterp_pf;                 % Risk free rate
retsPC = lnrtsimPC - rfr;            % Excess Returns
retsPD = lnrtsimPD - rfr;
h    =  1;                           % Forecast Horizon 0 = in-sample regression
yPC  =  retsPC(1+h:end,1);           % Regressand
yPD  =  retsPD(1+h:end,1);
rec_sim_02 = zeros(size(stsim,1),1);
for i = 1:size(stsim,1)
    if stsim(i) < log(0.02)
        rec_sim_02(i) = 1;
    else
        rec_sim_02(i) = 0;
    end
end
rec_sim_02 = rec_sim_02(2:end);
%%
lower_Sbar = 1;
s_bar_2 = log(0.02);
x   = [ones(length(yPD), 1), ...
    rec_sim_02(1:end-h,:) .* PD_regress(1:end-h,1), ...
    (1-rec_sim_02(1:end-h,:)) .* PD_regress(1:end-h,1)];
regPDrec = nwest(yPD,x,0); %% Full BC <- PD

x   = [ones(length(yPC), 1), ...
    rec_sim_02(1:end-h,:) .* PC_regress(1:end-h,1), ...
    (1-rec_sim_02(1:end-h,:)) .* PC_regress(1:end-h,1)];
regPCrec = nwest(yPC,x,0); %% Full BC <- PC


retsHRecPC = retsPC(1+h:end) .* rec_sim_02(1+h:end); %% Excess Returns Recession
retsHExpPC = retsPC(1+h:end) .* (1-rec_sim_02(1+h:end)); %% Excess Returns Expansions
retsHRecPD = retsPD(1+h:end) .* rec_sim_02(1+h:end); %% Excess Returns Recession
retsHExpPD = retsPD(1+h:end) .* (1-rec_sim_02(1+h:end)); %% Excess Returns Expansions

PDRegHRec = rec_sim_02(1:end-h,:) .* PD_regress(1:end-h,1);
```

```
PDRegHExp = (1 - rec_sim_02(1:end-h,:)) .* PD_regress(1:end-h,1);
PCRegHRec = rec_sim_02(1:end-h,:) .* PD_regress(1:end-h,1);
PCRegHExp = (1 - rec_sim_02(1:end-h,:)) .* PC_regress(1:end-h,1);

a = [retsHRecPD, PDRegHRec];
a = a(all(a,2),:);
ExcRetsRec = a(:,1);               %% <- Excess Returns Recessions only
PDrecHR1 = [ones(size(a,1), 1) a(:,2)]; %% <- PD recession
regPDrec1 = nwest(ExcRetsRec,PDrecHR1,0);

a = [retsHExpPD, PDRegHExp];
a = a(all(a,2),:);
ExRetsExp = a(:,1);                 %% <- Excess Returns Expansions only
PDexpHR1  = [ones(size(a,1),1) a(:,2)]; %% <- PD Expansion
regPDexp1 = nwest(ExRetsExp,PDexpHR1,0);

a = [retsHExpPC, PCRegHExp];
a = a(all(a,2),:);
ExRetsExp = a(:,1);
PCExpHR1  = [ones(size(a,1),1) a(:,2)];
regPCexp1 = nwest(ExRetsExp,PCExpHR1,0);

a = [retsHRecPC, PCRegHRec];
a = a(all(a,2),:);
ExRecRets = a(:,1);
PCrecHR1  = [ones(size(a,1),1) a(:,2)];
regPCrec1 = nwest(ExRecRets,PCrecHR1,0);

regs2 = [regPCrec regPDrec regPCrec1 regPCexp1 regPDrec1 regPDexp1];
if Save_Figures
RegressionTable2;
end
%%
load('PD_Claim_workspace','elnrtsim','tsc'); ExpRetsPD = elnrtsim;
load('PC_Claim_workspace','elnrtsim'); ExpRetsPC = elnrtsim;
figure;
subplot(2,1,1)
plot(ExpRetsPC);title({'Consumption Claim', ['$E( E_t (r_{t+1}) )$
    =',num2str(mean(ExpRetsPC),6)]},'Interpreter','latex');
xlim([0 100000]);
ylabel('$ E_t (r_{t+1})$','FontSize',14,'interpreter','latex');
subplot(2,1,2)
plot(ExpRetsPD);title({'Dividend Claim', ['$E( E_t (r_{t+1}) )$
    =',num2str(mean(ExpRetsPD),6)]},'Interpreter','latex');
ylabel('$ E_t (r_{t+1})$','FontSize',14,'interpreter','latex');
xlim([0 100000]);
if Save_Figures
saveas(gcf,'../Figures/Excess_Rets','epsc');
end
%% Persistence s_t
x = astsim_pf(1:end-1,:);
x1 = astsim_pf(2:end,:);
autocorrX = x\x1;
%%
load('PD_Claim_workspace','lnpctsim'); PDratio = lnpctsim;
```

```
load('PC_Claim_workspace','lnpctsim'); PCratio = lnpctsim;
name = '../Figures/PCPDMonthly_chain';
subplot(2,1,1)
plot(PCratio);ylabel('$p_t-c_t$','FontSize',14,'Interpreter','latex');
title({'Consumption Claim', ['$E(p_t-c_t)$ =',num2str(mean(PCratio),4)]},'Interpreter','latex');
xlim([1 100000]);
subplot(2,1,2)
plot(PDratio);ylabel('$p_t-d_t$','FontSize',14,'Interpreter','latex');
xlim([1 100000]);
title({'Dividend Claim', ['$E(p_t-d_t)$ =',num2str(mean(PDratio),4)]},'Interpreter','latex');
if Save_Figures
saveas(gcf,name,'epsc');
end
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%   Long run regressions based on simulated data %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
load('PC_Claim_workspace','Erfinterp_pf','lnrtsim','lnpctsim');
rfr  = Erfinterp_pf;
retsPC = lnrtsim - (rfr/4);
PCrat = lnpctsim(2:end);
load('PD_Claim_workspace','lnrtsim','lnpctsim')
PDrat = lnpctsim(2:end);
retsPD = lnrtsim - (rfr/4);
j = 1;
T = length(PCrat);
Ta = T;
h= [1 2 3 5 7 10] * 12;
while j <= size(h,2)
k = h(1,j);
xC = [ones(T-k+1,1), PCrat(1:T-k+1)];
yC  = retsPC(1:T-k+1);%-rfr;
xD = [ones(T-k+1,1) PDrat(1:T-k+1)];
yD = retsPD(1:T-k+1);
    i = 2;
while i <= k
 yC = yC + retsPC(i:T-k+i);
 yD = yD + retsPD(i:T-k+i);
 i = i+1;
end
b = xC\yC;
bmat(:,j) = b;
R2(:,j) = (std(xC * b)/ std(yC) )^2;
bd = xD\yD;
bdmat(:,j) = bd;
R2d(:,j) = (std(xD * bd)/ std(yD) )^2;
j = j+1;
end

tab = [bmat(2,:)', R2', bdmat(2,:)', R2d']
names = split(char(num2str(h/12,1)));
varnames = split(['$\beta_{pc}$ ', '$R^2_{pc}$ ','$\beta_{pd}$ ','$R^2_{od}$'])'
tab = array2table(tab,'Rownames',names,'VariableNames',varnames)
%% Moments Table
load('PD_Claim_workspace','Edc_pf', 'Stdc_pf', 'Erfinterp_pf', 'Shprinterp_pf', 'ShpRinterp_pf',
```

```
    'Eexrettinterp_pf', 'Stdexrettinterp_pf', 'Ep_d_pf', 'Stdp_d_pf');
PD_edc = Edc_pf;
PD_stdc = Stdc_pf;
PD_rfr = Erfinterp_pf;
PD_shrp = Shprinterp_pf;
PD_SHRP = ShpRinterp_pf;
PD_Eer = Eexrettinterp_pf;
PD_Stder = Stdexrettinterp_pf;
PD_Epd = Ep_d_pf;
PD_StdPD = Stdp_d_pf;
load('PC_Claim_workspace','Edc_pf', 'Stdc_pf', 'Erfinterp_pf', 'Shprinterp_pf', 'ShpRinterp_pf',
    'Eexrettinterp_pf', 'Stdexrettinterp_pf', 'Ep_d_pf', 'Stdp_d_pf');
PC_edc = Edc_pf;
PC_stdc = Stdc_pf;
PC_rfr = Erfinterp_pf;
PC_shrp = Shprinterp_pf;
PC_SHRP = ShpRinterp_pf;
PC_Eer = Eexrettinterp_pf;
PC_Stder = Stdexrettinterp_pf;
PC_Epd = Ep_d_pf;
PC_StdPD = Stdp_d_pf;
load('CC_PC_Claim_workspace','Edc_pf', 'Stdc_pf', 'Erfinterp_pf', 'Shprinterp_pf', 'ShpRinterp_pf',
    'Eexrettinterp_pf', 'Stdexrettinterp_pf', 'Ep_d_pf', 'Stdp_d_pf');
CC_PC_edc = Edc_pf;
CC_PC_stdc = Stdc_pf;
CC_PC_rfr = Erfinterp_pf;
CC_PC_shrp = Shprinterp_pf;
CC_PC_SHRP = ShpRinterp_pf;
CC_PC_Eer = Eexrettinterp_pf;
CC_PC_Stder = Stdexrettinterp_pf;
CC_PC_Epd = Ep_d_pf;
CC_PC_StdPD = Stdp_d_pf;
load('CC_PD_Claim_workspace','Edc_pf', 'Stdc_pf', 'Erfinterp_pf', 'Shprinterp_pf', 'ShpRinterp_pf',
    'Eexrettinterp_pf', 'Stdexrettinterp_pf', 'Ep_d_pf', 'Stdp_d_pf');
CC_PD_edc = Edc_pf;
CC_PD_stdc = Stdc_pf;
CC_PD_rfr = Erfinterp_pf;
CC_PD_shrp = Shprinterp_pf;
CC_PD_SHRP = ShpRinterp_pf;
CC_PD_Eer = Eexrettinterp_pf;
CC_PD_Stder = Stdexrettinterp_pf;
CC_PD_Epd = Ep_d_pf;
CC_PD_StdPD = Stdp_d_pf;
if Save_Figures
Simulatedmom;
end
%%
load('PC_Claim_workspace','stsim')
rec_sim_ss = NaN(length(stsim), 1);
for i = 1:length(stsim)
   if stsim(i) < Rec_s_bar
      rec_sim_ss(i) = 1;
   else
      rec_sim_ss(i) = 0;
   end
```

```
end
rec_sim_ss_percentage = sum(rec_sim_ss(:)==1) / length(rec_sim_ss);
%%
labeledMatrix = bwlabel(rec_sim_ss);
measurements = regionprops(labeledMatrix, 'Area');
RecessionLengths = [measurements.Area];
mean(RecessionLengths)/12;
%% FORECAST 1-period Expanding-Window
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                    Forecast Measures                         %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SampleSize = 0.05; % startng point 0 = full data [0:1[
WindowSize = 120; % 240 = 20 years
%%%                       Recessions                             %%%
init = SampleSize * size(ExRetsRecPDFC,1)+1;
MaxFC  = size(ExRetsRecPDFC,1)-WindowSize-1;
% LHS, Excess returns at time t+1
% RHS, ratios t
j = 1;
for i = init:MaxFC
    b1 = PDrecHR(i:i+WindowSize-1,:)\ExRetsRecPDFC(i:i+WindowSize-1,1);
    fit = b1(1) * PDrecHR(i+WindowSize,1) + b1(2) * PDrecHR(i+WindowSize,2);
    PDRecError(j,1) = ExRetsRecPDFC(i+WindowSize,:) - fit;
    PDRecMeanError(j,1) = ExRetsRecPDFC(i+WindowSize,:) - mean(ExRetsRecPDFC(i:i+WindowSize-1,:));

    b1 = PCrecHR(i:i+WindowSize-1,:)\ExRetsRecPCFC(i:i+WindowSize-1,1);
    fit = b1(1) * PCrecHR(i+WindowSize,1)+b1(2) * PCrecHR(i+WindowSize,2);
    PCRecError(j,1) = ExRetsRecPCFC(i+WindowSize,:) - fit;
    PCRecMeanError(j,1) = ExRetsRecPCFC(i+WindowSize,:) - mean(ExRetsRecPCFC(i:i+WindowSize-1,:));

    j=j+1;
end
%% Expansions
Init = SampleSize * size(ExRetsExpPDFC,1)+1;
MaxFC  = size(ExRetsExpPDFC,1)-WindowSize-1;
j = 1;
for i = init:MaxFC
    b1 = PDexpHR(i:i+WindowSize-1,:)\ExRetsExpPDFC(i:i+WindowSize-1,1);
    fit = b1(1) * PDexpHR(i+WindowSize,1) + b1(2) * PDexpHR(i+WindowSize,2);
    PDExpError(j,1) = ExRetsExpPDFC(i+WindowSize,:) - fit;
    PDExpMeanError(j,1) = ExRetsExpPDFC(i+WindowSize,:) - mean(ExRetsExpPDFC(i:i+WindowSize-1,:));

    b1 = PCExpHR(i:i+WindowSize-1,:)\ExRetsExpPCFC(i:i+WindowSize-1,1);
    fit = b1(1) * PCExpHR(i+WindowSize,1) + b1(2) * PCExpHR(i+WindowSize,2);
    PCExpError(j,1) = ExRetsExpPCFC(i+WindowSize,:) - fit;
    PCExpMeanError(j,1) = ExRetsExpPCFC(i+WindowSize,:) - mean(ExRetsExpPCFC(i:i+WindowSize-1,:));

    j=j+1;
end
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                     R^2 OOS                                      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
OoSR2ExpPD = 1-sum(PDExpError.^2)/sum(PDExpMeanError.^2);
OoSR2ExpPC = 1-sum(PCExpError.^2)/sum(PCExpMeanError.^2);
```

```
OoSR2recPD = 1-sum(PDRecError.^2)/sum(PDRecMeanError.^2);
OoSR2recPC = 1-sum(PCRecError.^2)/sum(PCRecMeanError.^2);
OoSR2 = [OoSR2recPC OoSR2recPD OoSR2ExpPC OoSR2ExpPD];
```

```
addpath('Data');
addpath('Calibration');
addpath('Functions');
clear
%%
datfile = importdata('All_returns_market_Monthly.csv');
%datfile = importdata('All_returns_market_Annual.csv');
cal = datfile.data(:,1); % Dates
r  = datfile.data(:,2); % Returns
rx  = datfile.data(:,3); % Returns less dividinds
%%
dp =  (1+r)./(1+rx) - 1; % Div. Yield
dp = smoothdata(dp,'movmean',3);
dp = dp.^(-1);
dd = [NaN ; dp(2:end)./dp(1:end-1).*(1+rx(2:end))-1]; % Div. Growth
%%
RR = importdata('Rets30dayTBILL.csv');
Bond = RR.data(:,2); % Returns on 90 day T-bill
Infl = RR.data(:,3); % Inflation rate measured as CPI
RFR = Bond - Infl; % Real Risk free rate
re = r-RFR; % Excess Returns
%%
RecData = importdata('USREC_Period.csv');
Rec = RecData.data(:,1);
%%
T = length(dp);
dpRec = dp .* Rec;
reRec = re .* Rec;

a = [dpRec(1:end-1), reRec(2:end)];
a = a(all(a,2),:);

rhv = [ones(length(a),1), log(1 + a(:,1))];
lhv = log(1+ a(:,2));
Rec_Reg = nwest(lhv,rhv,2);
prt_reg(Rec_Reg)
%%
dpExp = dp .* (1-Rec);
reExp = re .* (1-Rec);

a = [dpExp(1:end-1), log(1+reExp(2:end))];
a = a(all(a,2),:);

rhv = [ones(length(a),1), a(:,1)];
lhv = log(1+a(:,2));
Exp_Reg = nwest(lhv,rhv,2);
%%
prt_reg(Exp_Reg)
prt_reg(Rec_Reg)
```

```
%% Table 1 - Calibrated Model
```

```
name = string(['Tables/Table_1_Calib_', num2str(calib),'_PD_', num2str(PD_Claim), '.tex']);
if isfile(name)
delete name;
end

%%
diary(name);
diary on

disp('\begin{table}[H]');
disp('\centering');
disp('\begin{threeparttable}[b]');
disp(['\caption{Parameters of the model, with calibration = ', num2str(calib),', and using claim = ', num2str(PD_Claim),'}']);
disp(['\label{tab:ModelCalib_',num2str(calib),'_',num2str(PD_Claim),'}']);
disp('\begin{tabular}{@{}ll@{\hspace{1.5cm}}ll@{}}');
disp('\toprule');
disp(' & Parameter                          & Notation       & Value   \\ \midrule') ;
disp('\multicolumn{4}{l}{\textit{Calibrated}}                          \\');
disp([' & Mean consumption growth           & $g$            & $', num2str(Edc_pf), '$ \\']);
disp([' & Standard deviation of $\Delta c_t$ & $\sigma$      & $', num2str(Stdc_pf),'$ \\']);
disp([' & Standard deviation of $\Delta d_t$ & $\sigma_w$    & $', num2str(Stdc_pf), '$ \\']);
disp([' & Log risk-free rate                & $r^f$          & $', num2str(Erfinterp_pf),'$ \\']);
disp([' & Persistence parameter             & $\phi$         & $', num2str(phi),'$ \\']);
disp([' \multicolumn{4}{l}{\textit{Assumed}}                           \\']);
disp([' & Coefficient of Risk Aversion      & $\gamma$       & $', num2str(gamma),'$ \\']);
disp([' & Correlation dividends/consumption & $\rho$         & $', num2str(phi),'$ \\']);
disp(['\multicolumn{4}{l}{\textit{Implied}}                            \\']);
disp([' & Subjective discount factor        & $\delta$       & $', num2str(delta),'$ \\']);
disp(['& Steady-state surplus consumption ratio & $\Bar{S}$ & $', num2str(S_bar),'$ \\']);
disp([' & Maximum surplus consumption ratio & $S_{\text{max}}$ & $', num2str(S_max),'$ \\ \bottomrule']);
disp('\end{tabular}');
disp('\begin{tablenotes}');
disp('\footnotesize{\item [1] All relevant parameters are annualized');
disp('            \item [2] Calibrated parameters are estimated from data, assumed are chosen arbitrarily on the grounds of existing literature, while implied parameters are calculated from the calibrated/assumed parameters.}');
disp('\end{tablenotes}');
disp('\end{threeparttable}');
disp('\end{table}');

diary off

% %% Table 2 - Data Properties
% name = string(['Tables/Table_2_Calib_', num2str(calib),'_PD_', num2str(PD_Claim), '.tex']);
% if isfile(name)
% delete name;
% end
%
% %%
% diary(name);
% diary on
% disp(['\begin{table}[H]']);
% disp(['\centering']);
% disp(['\caption{Data Properties calibration = ', num2str(calib),' claim = ', num2str(PD_Claim),'}']);
```

```matlab
% disp(['\label{tab:Data_props_', num2str(calib),'_',num2str(PD_Claim),'}']);
% disp(['\begin{tabular}{@{}l@{\hspace{1.5cm}}l@{\hspace{1.5cm}}l@{}}']);
% disp(['\toprule']);
% disp([' & \textit{Simulated} & \textit{Historic} \\ \midrule']);
% disp(['$\mathbb{E}\left[r_t- r^f_t\right]$& $',                              num2str(Eexrettinterp_pf),'$ & $', num2str(0.0927),'$       \\']);
% disp(['$\sigma\left(r_t - r^f_t \right)$ & $',                              num2str(Stdexrettinterp_pf),'$ & $', num2str(0.1670),'$       \\']);
% disp(['$\mathbb{E}\left[r_t- r^f_t\right] / \sigma\left(r_t - r^f_t,\right)$ & $', num2str(Shprinterp_pf),'$ & $', num2str(0.5548),'$ \\ \bottomrule']);
% disp(['\end{tabular}']);
% disp(['\end{table}']);
% diary off


%% Table 3 - Simulated Moments
name = string(['../Tables/Moments.tex']);
if isfile(name)
delete(name);
end

%%
momPC = table2array(readtable('PC_Claim_Sim_mom.txt'));
momPD = table2array(readtable('PD_Claim_Sim_mom.txt'));
diary(name);
diary on
disp(['\begin{tabular}{@{}llllllllll@{}}']);
disp(['\toprule ']);
disp([' & $\mathbb{E}\Delta d$ & $\sigma_{\Delta d}$ & $\mathbb{E}r^f$ & $\mathbb{E}r^m/\sigma _{r^m}$ & $\mathbb{E}R^m/\sigma _{R^m}$ & $\mathbb{E}r^m$ & $\sigma_{r^m}$ & $\mathbb{E}d-p$ & $\sigma_{d-p}$ \\ ']);
disp(['\midrule ']);
disp(['\multicolumn{10}{l}{$P/D$}\\']);
disp([' &', num2str(momPD(1,1)),'&', num2str(momPD(1,2)),'& ', num2str(momPD(1,3)),' & ', num2str(momPD(1,5)),' & ', num2str(momPD(1,6)),' & ', num2str(momPD(1,7)),' & ', num2str(momPD(1,8)),' & ', num2str(momPD(1,9)),' & ', num2str(momPD(1,10)),' \\ ']);
disp(['\multicolumn{10}{l}{$P/C$}\\']);
disp([' &', num2str(momPC(1,1)),'&', num2str(momPC(1,2)),'& ', num2str(momPC(1,3)),' & ', num2str(momPC(1,5)),' & ', num2str(momPC(1,6)),' & ', num2str(momPC(1,7)),' & ', num2str(momPC(1,8)),' & ', num2str(momPC(1,9)),' & ', num2str(momPC(1,10)),' \\ ']);
disp(['\bottomrule ']);
disp(['\end{tabular}']);
diary off
```

```matlab
global Regressions
%% Figures
clear
% Figure 7 in CC1998
load('PC_Claim_workspace')

Sample = 1:500;

figure;
subplot(2,1,1)
scatter(lnrtsim(Sample)*1e2,lndctsim(Sample)*1e2);title("Monthly Returns vs. consumption growth");
```

```
subplot(2,1,2)
scatter(alnrtsim_pf(Sample)*1e2,alndctsim_pf(Sample)*1e2);title("Annual Returns vs. consumption growth");
hold off
%saveas(gcf,string(['Figures/Figure_7_CC_1998_Calib_', num2str(calib),'_PD_', num2str(PD_Claim), '.eps']),'eps2c');


%%
load('PD_Claim_workspace','output_lnpda','S','tsc');PD_ratio = output_lnpda;
load('PC_Claim_workspace','output_lnpca');PC_ratio = output_lnpca;
%%
figure;
plot(S,exp(PC_ratio)/tsc,'LineWidth',1.5);% Annulized P/C-curve
hold on;
plot(S,exp(PD_ratio)/tsc,'LineWidth',1.5); % Annulized P/D-curve
ylabel('$P/C,\qquad P/D$','Interpreter','latex');
xlabel('Surplus Consumption ratio, $S$','Interpreter','latex');
xline(exp(Rec_s_bar),'--','$\bar{S}_{REC}$','Interpreter','latex');
xline(S_max,'--','$\bar{S}_{MAX}$','Interpreter','latex');
xline(0.02,'--','$\bar{S}_{2,REC}$','Interpreter','latex');
legend('PC-Ratio', 'PD-Ratio','Location','best')
hold off;
saveas(gcf,string(['../Figures/PC_PD_Ratio']),'eps2c');
%%
load('PD_Claim_workspace','elnr_pf','lnrf_pf');lnrPD = elnr_pf;
load('PC_Claim_workspace','elnr_pf');lnrPC = elnr_pf;
figure;
plot(S,lnrPC *tsc*100,'LineWidth',1.5);
hold on
plot(S,lnrPD*tsc*100,'LineWidth',1.5);
yline(mean(lnrf_pf)*tsc*100,':');
xline(exp(Rec_s_bar),'--','$\bar{S}_{REC}$','Interpreter','latex');
xline(S_max,'--','$\bar{S}_{MAX}$','Interpreter','latex');
xline(0.02,'--','$\bar{S}_{2,REC}$','Interpreter','latex');
ylabel('Expected Returns, annual percentage, $E_t ( r_{t+1} )$','Interpreter','latex');
xlabel('Surplus Consumption ratio, $S$','Interpreter','latex');
legend('Expected Return, Consumption Claim','Expected Return, Dividend Claim','Risk Free Rate','Interpreter','latex','Location','best');
saveas(gcf,string(['../Figures/ErPCPD']),'eps2c');
```

```
function [Coefficients] = Model_Calibration
Coefficients = struct;
%% Risk free rate
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% The Risk free rate is calculated from annual returns on %%%
%%% 90 day T-bills less the inflation rate. Using CRSP index %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RR = importdata('Bonds_INF.csv');
Bond = RR.data(:,2);
Infl = RR.data(:,4);
Rbond = Bond - Infl;
RFR = log(Rbond + 1); % Risk free rate calib;
meanRFR = mean(RFR);
Coefficients.rf = meanRFR;
clearvars -except Coefficients
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% the mean of the risk free rate 1950-2019 is found to be .0109 %%%
%%% that is slightly above 1%                              %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Persistence coefficient
RetAM = importdata('All_returns_market_Monthly.csv');
R =  1+RetAM.data(:,2);
Rx =  1+RetAM.data(:,3);
pd = log(R./Rx-1);
pd_t  = pd(1:end-1,:); % t
pd_t1 = pd(2:end,:); % t+1
AR1 = pd_t1\[ones(size(pd_t,1),1) pd_t];
Coefficients.Phi = AR1(2)^12;
%clearvars -except Coefficients R Rx
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Autocorrelation of price/dividend ratio is found to be .9008 %%%
%%% that is slightly above .87                             %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Consumption growth moments
%url = 'https://fred.stlouisfed.org/';
%c = fred(url);
%startdate = '01/01/1950';
%enddate = floor(now);
%cons = fetch(c, 'A796RX0Q048SBEA',startdate,enddate); % Real Consumption non-durable goods
%consdat = cons.Data(:,2);
%writematrix(consdat);
dat = readtable('consdat.txt');
dat = table2array(dat);
dat = log(dat);
diffDat = dat(2:end) - dat(1:end-1);
g = mean(diffDat)*4;
Coefficients.g = g;
Coefficients.sigma = std(diffDat) * sqrt(4);
clearvars -except Coefficients
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Multiplying with 4 as we use quarterly data here, g is found to be %%%
%%% .0134 or 1.34% while sigma_v is found to be 1.52% (.0152)   %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Standard deviation of dividend growth
RetAM = importdata('Annual_rets_ADAX_NYSE.csv');
RetAM = RetAM.data(:,[2 3])+1;
dp = RetAM(:,1)./RetAM(:,2)-1; % DP-ratio
DD = dp(2:end)./dp(1:end-1).*RetAM(2:end,2);
sigma_w = std(DD);
Coefficients.sigma_w = sigma_w;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Annual Data on returns yields a sigma_w f .1256 or 12.56%   %%%
%%% Note here not very robust to frequency changes              %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```