1 Documentation of CMOD

1.1 Introduction

This paper provides a basic guide to running the program (sections 1.2 to 1.4) and sets out the main features of CMOD (section 1.5), including, where appropriate, guidance about the code and parameters used in the model. The full code of the program is accessible in a series of visual basic modules attached to the CMOD spreadsheet and as an attachment to this paper — and provide complete documentation of the program equations. The Commission has updated the version of CMOD provided with the draft report to make the program more user-friendly and to add additional features, such as generation of graphs.

1.2 Some preliminaries for running CMOD

CMOD is opened just like any other Excel spreadsheet. However, in order to run the program it is necessary to ensure that the Analysis Toolpak and Analysis Toolpak - VBA add-ins are activated. The add-ins are routinely supplied as a part of Excel. To activate, go to the Tools menu in Excel, and click Add-Ins (figure 1.1). Then tick the boxes 'Analysis Toolpak' and 'Analysis Toolpak –VBA'. If the add-ins are not listed, browse to locate them. If you see a message that indicates that they are not currently installed on the computer, tick yes to install them.

1.3 How to do calculations when *not* undertaking simulations

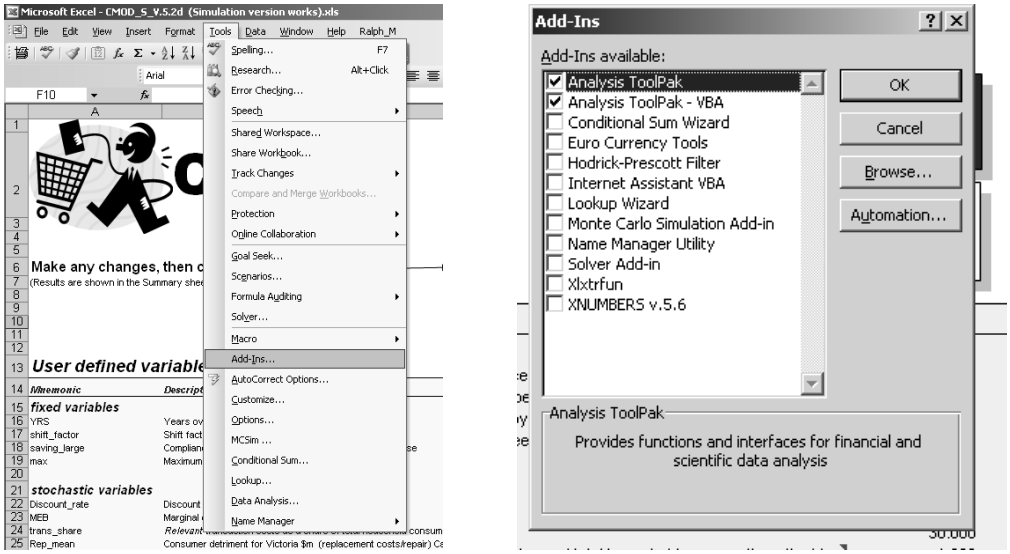
When opening CMOD, you should ensure you are in the '*Start here*' worksheet (labelled as the red tab at the bottom of the workbook). The screen appears as shown below in figure 1.2. If you are not interested in simulations at all, then the only figures of interest are those headed 'average values'. You can change these by clicking on the relevant cells and entering your new assumptions.

Once you have made the desired changes to the 'average values' column, click on the button '*Calculate values*'. The default setting ignores uncertainty and will

generate non-simulation or 'average' output values (figure 1.3). Simply click on OK to run in this mode.

(At any point the default set of values for the whole column headed average values can be restored by clicking on the '*Calculate values*' button and clicking the yes option for 'Restore values of parameters?')

Figure 1.1 **Installing the relevant add-ins**



The results will be calculated and the program will take you to the 'Summary' sheet (also labelled as a red tab). Some other (less important) result details are shown in the 'Some detailed results' sheet. You may also compare the new results with those produced by the default settings by examining the worksheets 'Main default results' and 'Deviations from default results'.

1.4 Simulation mode

The simulation mode of CMOD is considerably more complex.

A first step in using this mode is to understand why simulations are worth doing, how they work, and the distributional choices underpinning them. The assumptions underlying CMOD (and the parameters that embody these) are not certain. There are several commonly applied ways of seeing the effects of this uncertainty on outcomes, including:

- making different assumptions for each parameter one at a time, holding all others constant; and

- choosing all the high values (or all the low values) and considering the most extreme outcomes that are possible.

Figure 1.2 **CMOD opening screen**

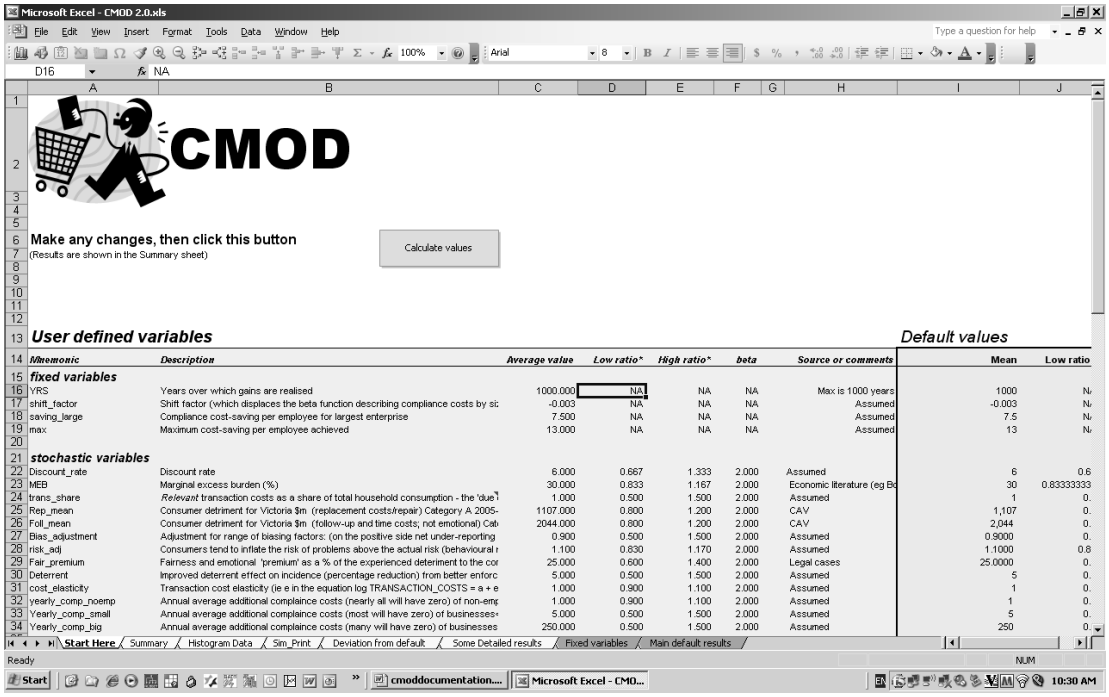
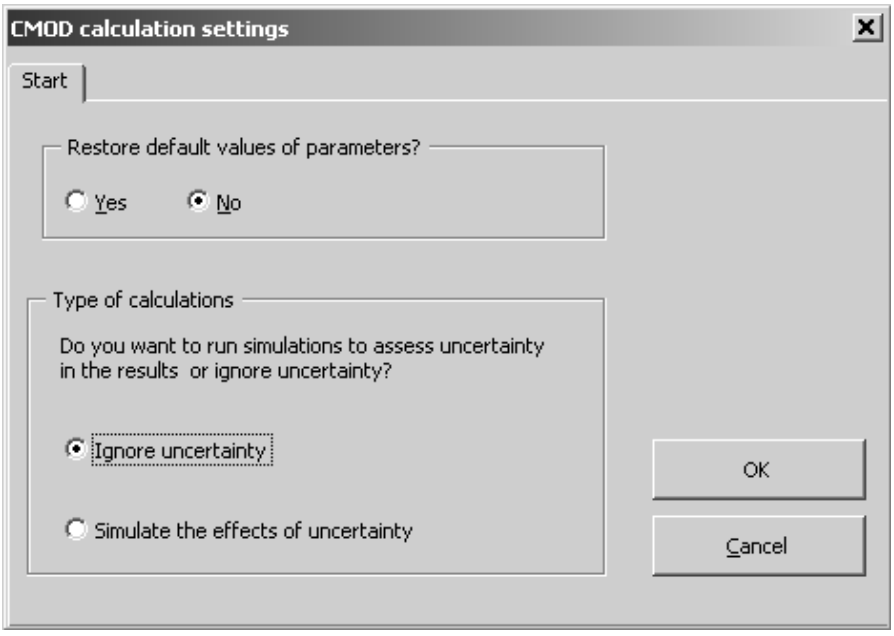


Figure 1.3 **CMOD run mode choices**

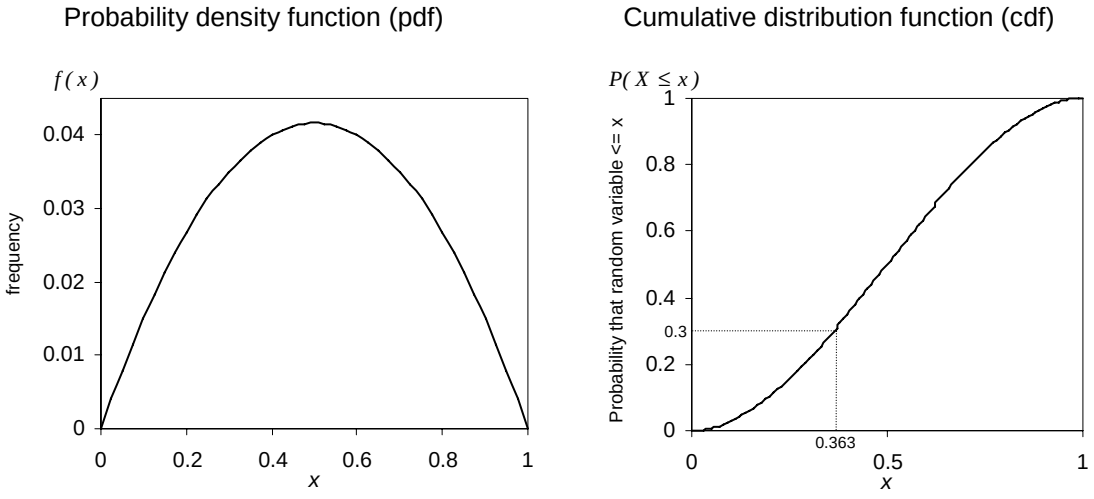


The problem with both approaches is that they give no indication of the likelihood of the outcomes. In particular, the latter approach gives a result that is highly

unlikely because there is a very low probability that all the parameters are simultaneously close to their lowest or highest values. Such an approach is usually only appropriate if the costs of extreme outcomes are very high or there are only a few discrete parameters underlying the outcomes of the model.

An alternative approach is to simulate the outcomes by drawing parameter values from plausible distributions generating them. The Commission has used a beta distribution for each parameter since this distribution is simple, but highly flexible. The shape of the distribution depends on two values, α and β. As α and β are varied, the beta distribution can be anything from a uniform distribution ($\alpha=1$, $\beta=1$) to a normal distribution ($\alpha=\beta \Rightarrow \infty$), and be either symmetric ($\alpha=\beta$) or asymmetric (among other possibilities). The default setting in CMOD is $\alpha=\beta=2$, which gives much more weight to outlying values than a normal distribution. The Commission has used this setting to give more weight to extreme values than the normal distribution, reflecting the significant uncertainty about parameter values. The characteristics of the beta distribution at $\alpha=\beta=2$ is shown in figure 1.4.

Figure 1.4 **The beta distribution**
$\alpha=\beta=2$



a The area under the pdf is one and the area between any two values gives the probability of finding x between these two values. The cdf gives the probability that a random variable distributed as a beta distribution is equal or less than a certain value. For example, there is a 30 percent chance that a variable distributed as a beta distribution bounded by zero and one will be less than or equal to 0.363. The inverse of the cdf gives the value of x associated with a given probability.

In each iteration of the simulation for each parameter (x), a number between 0 and 1 is drawn from a uniform distribution and the corresponding value of x is then derived using the inverse cumulative distribution function for the beta distribution. For example, suppose x was distributed as a beta distribution bounded between 0

and 1. Given a random draw from a uniform distribution of 0.3 (p=0.3), then $x = F^{-1}(p) = F^{-1}(0.3) = 0.363$ (figure 1.4) where F^{-1} denotes the inverse cdf.

Since the parameters used in CMOD are often not bounded between 0 and 1, the beta distribution has to be adapted to take account of this. Suppose that the observed parameter, x, is bounded between L and H. In that case, $x = L + F^{-1}(p) \times (H - L)$. These high and low values can be expressed as ratios of the average value of the parameter value ($\bar{x}$), so that $L = l \times \bar{x}$ and $H = h \times \bar{x}$ where l and h are the multiplicative factors that give L and H when multiplied by the mean. Consequently, $x = \bar{x} \times \{ l + F^{-1}(p) \times (h - l) \}$.

Now the average value of the beta distribution is $\alpha/(\alpha+\beta)$, so that $l + \alpha/(\alpha+\beta) \times (h - l) = 1$.

This implies that $\alpha = \frac{\beta(1-l)}{(h-1)}$ so that for given values of β, l, h it is possible to derive α. Accordingly, in CMOD, users enter four relevant factors for any random parameter — its average, the low and high *ratios* and the beta of its beta distribution.

Running in simulation mode

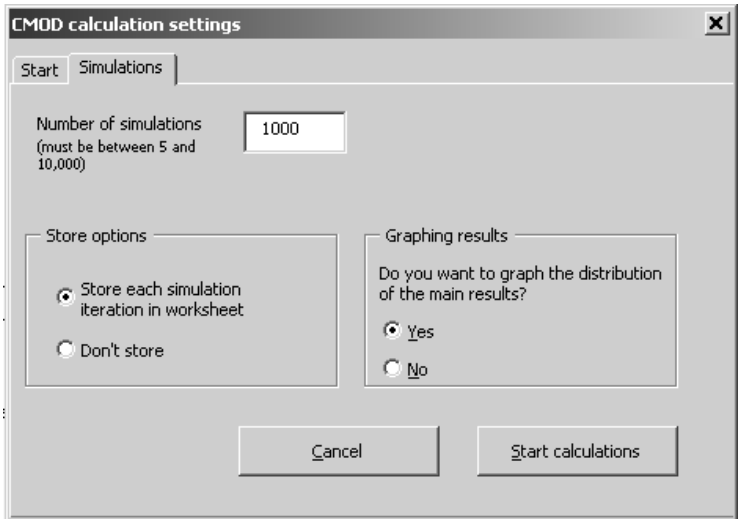
The simulation module allows users to specify the number of simulations, the ranges that stochastic parameters can take and the characteristics of the distribution generating the stochastic variables.

Users can modify the Commission's distributional assumptions in the 'Start here' worksheet of CMOD (as shown in figure 1.2).

In this case, you should make any changes to the columns C to F labelled consecutively as 'Average value', 'Low ratio', 'High ratio', and 'beta'. Using the above terminology these are $\bar{x}$, l, h and β for each relevant variable. (As before, you can restore all the default values by clicking on the '*Calculate values*' button and clicking the yes option for 'Restore values of parameters?'.)

Once you have made any desired changes to the 'average values' column, click on the button '*Calculate values*'. If you want to run simulations, click on the button adjacent to 'Simulate the effects of uncertainty'. Then a new screen will appear allowing various options for simulation modelling (figure 1.5). (If you decide you do not want to run simulations after all, you can always click on the '*Start*' tab and go back to the initial CMOD options.

Figure 1.5 **Simulation mode**



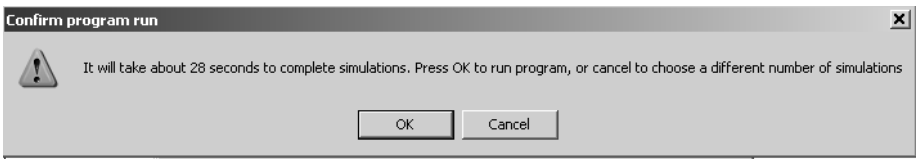
The default number of simulations is 1000. This will give reasonable approximations to the underlying distributions of all the relevant results. However, you can run anything between 5 and 10 000 simulation iterations by typing in a different number in the relevant text box. On the average PC, it will take about 30 seconds to run 1000 simulation iterations.

You can also choose whether to store the simulation results (storing results is the default) and whether to graph the resulting distributions of the key variables (again the default setting).

If you are satisfied with the settings then click the '*Start calculations*' button.

CMOD will tell you the rough duration of the calculations (figure 1.6). If that is too long you can click cancel in the relevant message box.

Figure 1.6 **Message box on duration of simulation run**



As before, the results will be calculated and the program will take you to the 'Summary' sheet (also labelled as a red tab). Table 5 will now be shown and will give the relevant statistics for the simulations (annuity values only). Individual simulation results ('Sim print) and graphs will be produced ('Histogram data') as separate worksheets, unless you have specified that you do not want them.

1.5 Key features of CMOD

CMOD consists of a series of modules that take account of the major impacts of consumer policy. The figure below, taken from chapter 14 of the report, indicates the channels through which consumer policy has impacts on the community, while table 1.1 lists all the major variables in the model.

Table 1.1 **Mnemonics and descriptions of initial variables in CMOD**

Mneumonic	*Description*
Fixed variables	
VIC_POP_16	Number of consumers Victoria 16 years plus
AUST_POP16	Number of consumers Australia 16 years plus
CON2005_06	Value of household consumer transactions (Australia $m 2005-06 current prices)
LR_con_sh	Long run average 1992-93 to 2006-07 Household consumption to GDP ratio (ratio)
LR_Total_Cons_Share	Long run average 1992-93 to 2006-07 Public & private consumption to GDP ratio (ratio)
Actions	Current Australia-wide estimated consumer policy court actions by regulators (number) 2005-06
Share_foreign	Share of equity owned by foeigners of those firms that are the subject of new consumer laws %
share_success	Success rate of new actions by regulators %
welfare_weight	Welfare weight factor
low_emp	Low employment range for businesses facing compliance savings (single proprietor businesses)
high_emp	Max employment at which compliance savings per employee are constant
alpha_emp	Parameter for function determining shape of compliance savings across firm size (alpha)
beta_emp	Second parameter for function determining shape of compliance savings across firm size (beta)
GDP_2006_07	GDP 2006-07 (2006-07 prices)
YRS	Years over which gains are realised
shift_factor	Shift factor (which displaces the beta function describing compliance costs by size)
saving_large	Compliance cost-saving per employee for largest enterprise
max	Maximum cost-saving per employee achieved

Figure 1.7 **Impacts of consumer policy**

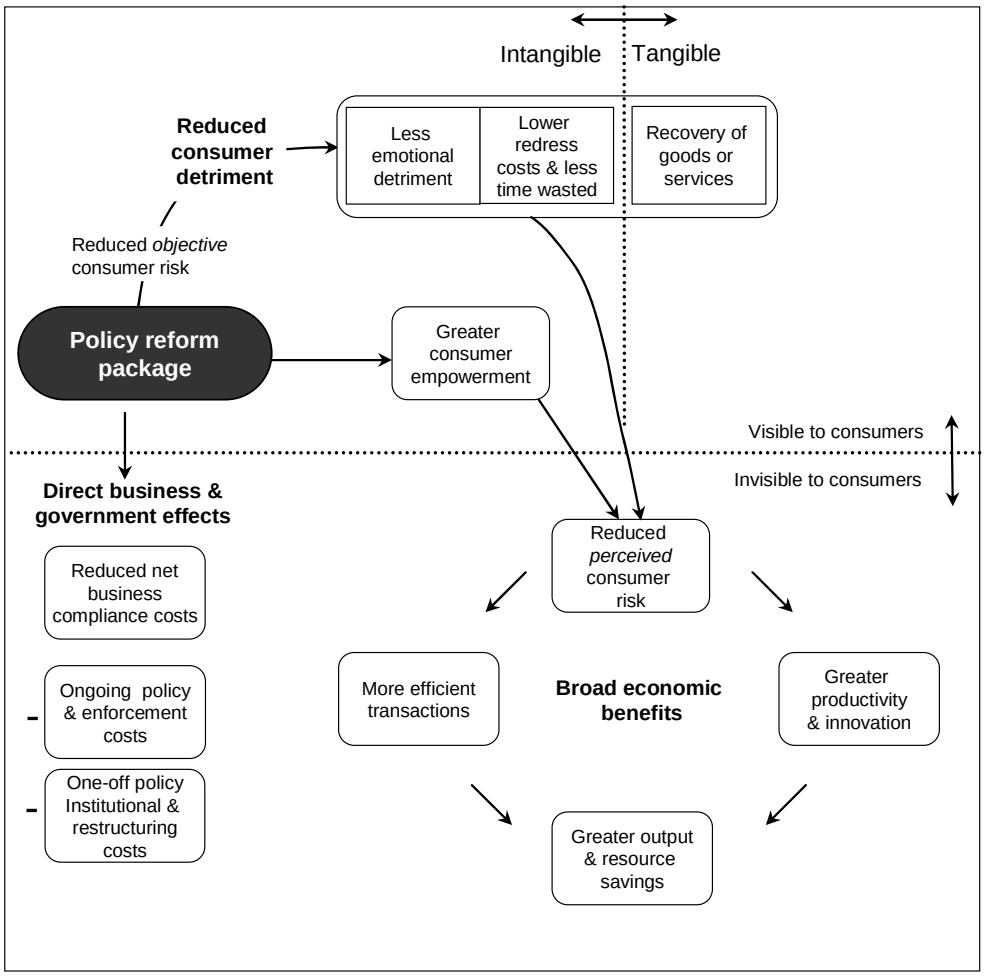


Table 1.1 **Continued**

Stochastic variables	
Discount_rate	Discount rate
MEB	Marginal excess burden (%)
trans_share	Relevant transaction costs as a share of total household consumption - the 'due diligence' costs of consumers (eg risk-based consumer search costs, brokers, advertising, information provision) % (ie ask the question, how many tangible and intangible cents per dollar are consumers willing to pay to reduce the risks of their purchases? eg time costs in searching on the internet; willingness to forgo a larger market to buy using eBay, paying for expert advice; etc)
Rep_mean	Consumer detriment for Victoria $m (replacement costs/repair) Category A 2005-06 (survey mean)
Foll_mean	Consumer detriment for Victoria $m (follow-up and time costs; not emotional) Category B 2005-06 (survey mean)
Bias_adjustment	Adjustment for range of biasing factors: (on the positive side net under-reporting of events covered and unreported unrevealed detriment, and on the negative side reporting of events that do not reflect misconduct or events from previous years)

Table 1.1 **Continued**

risk_adj	Consumers tend to inflate the risk of problems above the actual risk (behavioural research shows this, as does revealed behaviour with extended warranties)
Fair_premium	Fairness and emotional 'premium' as a % of the experienced deteriment to the consumer
Deterrent	Improved deterrent effect on incidence (percentage reduction) from better enforcement tools, unfair contracts, better empowerment, better institutions
cost_elasticity	Transaction cost elasticity (ie ε in the equation log TRANSACTION_COSTS = α + ε log RISKS + other variables)
yearly_comp_noemp	Annual average additional compliance costs (nearly all will have zero) of non-employing businesses
Yearly_comp_small	Annual average additional compliance costs (most will have zero) of businesses<100
Yearly_comp_big	Annual average additional compliance costs (many will have zero) of businesses>100
UnfairC_noemp	Setup average per firm costs associated with unfair contracts (no emp firms) $
UnfairC_small	Setup average per firm costs associated with unfair contracts (small firms) $
UnfairC_big	Setup average per firm costs associated with unfair contracts (big firms) $
Percent_actions	Litigation % increase (%)
Av_fine	Average fine/costs per guilty firm
False_positive	False positive rate (ie share of actions where penalty is applied where it should not)
Days_case	Days per case (legal/investigative, mediation etc) (will be weighted towards class actions and unfair contracts)
Cost_per_day	Cost per day of regulator/business in any contested case
Preparation_costs	Regulators' preparation costs per case (will be highly skewed - some very expensive, some cheap with a mean well above the mode)
Business_margin	The % difference between the costs of actions by business cf govt
POLICY_FT	FT staff increment associated with general policy development and admininstration (commencing in 2007-08)
POLICY_WAGE	Cost per staff member of new policy staff (including overheads) $
TEMPLATE_FT	FT person years per jurisdiction and negotiating new template arrangements and industry specific arrangements
Template_wage	Cost per staff member (average) for negotiating new template arrangements and industry specific arrangements, including overheads
OFT_ASIC_FT	Number of staff moves from OFTs to ASIC re credit
OFT_ASIC_WAGE	Salary costs per staff member involved in move from OFTs to ASIC, including overheads
Wastage	Share of annual staff time transitorily wasted in re-allocation of credit from States to ASIC (%)
LEGAL_ONEOFF_FT	Staff person years involved (all jurisdictions) in one-off legal & policy costs (legal advice re how credit should be integrated into the Corporations Act; wording of new unfair contracts law; new discosure requirements)

Table 1.1 **Continued**

LEGAL_ONEOFF_WAGE	Cost per staff member including overheads of one-off legal & policy costs $
Base_TFP_error	Base_TFP_Error = (Base_TFP less Max_TFP), where Base_TFP is the productivity growth at the initial level of consumer risk, and Max_TFP is the productivity growth rate when risk is zero
LowestTFP	Lowest TFP growth rate
Max_TFP	The maximum steady state TFP growth rate (associated with zero consumer risk)
Maxval_risk	Highest value of consumer risk before DTFP = TFPmin
multiplier_by_state	Parameters for determining the compliance savings as a function of the number of jurisdictions that a firm operates in
scale_compliance_firmsize	Scale_factor for compliance by business size
Nontransfer_sh	Non-transfer element of share that are repair/replacement costs (%)
Ongoing_Admin_Unfair	Ongoing unfair contracts regulatory staff (excluding those who are used to process legal cases)

The policy reform impacts perceived by consumers

The initial calculations use the data from the CAV survey of consumer detriment to derive a value for the risk that consumers face when making transactions.

The following calculates the gross value of detriment in Victoria based on the costs of replacement of goods and services (SV_Rep_Mean) and follow-up costs (SV_Foll_Mean). It (for the moment) ignores any psychic ('emotional') costs of detriment.

Vic_detriment = SV_Rep_Mean + SV_Foll_Mean

The Australian detriment is then estimated by grossing up the Victorian estimate using the adult population ratio.

Australian_detriment = Aust_pop16 / Vic_pop16 * Vic_detriment * SV_Bias_Adjustment

The additional factor (SV_Bias_Adjustment) takes account of any bias in the Victorian survey that may exaggerate or underestimate the value of the appropriate measure of detriment.

As also explored in chapter 14, the raw Victorian estimates may reflect consumer problems that should not be counted in the base level of consumer detriment relevant for the policy impact analysis in CMOD. Apparent consumer detriment can arise for a variety of reasons other than 'unfair' or illegal behaviour by businesses or sub-optimal behaviour by consumers — the focus of policy reforms in the Commission's report. Detriments can also arise from:

- the outcomes of decisions by suppliers about quality control or managing supply that actually assist consumers in general. For example, the cost of implementing a zero defect rate in consumer electronics would be prohibitive, pushing up prices of these goods beyond what consumers would be prepared to pay; and
- innocent errors and misunderstandings by business staff.

Were these sources of 'innocent detriment' to be fully included in the CAV survey then it would imply that the detriment associated with problematic business behaviour and sub-optimal consumer behaviour might be significantly smaller than the actually cited estimate. This would distort the base level of detriment and exaggerate the benefits of reform.

However, the detriment recorded by the CAV survey (shown as B in box 14.4 from chapter 14 of the report) is likely to exclude a significant proportion of such 'innocent detriment' (shown as C in the diagram in that box). In fact, it is possible that, overall, the CAV estimate for Victoria, and by inference, the estimated Australian estimate given by CMOD, may underestimate the base level of consumer detriment relevant to consumer policy reform. Nevertheless, in the default calculations of CMOD, the Commission has set SV_Bias_Adjustment at 0.9 (i.e. 90 percent).

The CAV survey provides qualitative account of the account of 'damages' (for example, emotional distress and the value people place on 'fair' outcomes), but does not provide quantitative estimates. These are hard to estimate, but a variety of indicators suggest they can be considerable. A base figure of 25 percent (SV_Fair_Premium) has been used to gross up the previous measure of detriment:

Experienced_Australian_detriment = Australian_detriment * (1 + SV_Fair_Premium / 100)

Since an important element of CMOD are the benefits derived from reducing consumer risk, measures of risk *before* and *after* policy change need to be calculated. The appropriate denominator for such a calculation is household consumption (from the National Accounts). Consumer *perceptions* of risk can be different from *objective* measures of risk, as documented in the behavioural results described in appendix B of the report. Such risk adjustment (SV_Risk_Adj) can be greater or smaller than unity, but in most contexts, it is likely that consumers exaggerate the objective risks they face (for example, as indicated by apparent over-insurance). A bias of 10 per cent has been used in the default scenarios.

Risk_before = Experienced_Australian_detriment / CON2005_06 * 100 * SV_Risk_Adj

The gross impacts of policy reform are based on an assumption about their percentage impact (SV_deterrent) on base level detriment:

Avoided_Aust_detriment = SV_Deterrent / 100 * Australian_detriment

In estimating the impacts of policy change, it is important to distinguish between consumers' perceptions of detriment and the actual community-wide detriment. Some of the apparent detriments perceived by particular consumers represent transfers from consumers to business owners (or through prices, to other consumers). The extent of transfer is given by the variable SV_Nontransfer_S, set at 50 percent in the base scenario. Transfers are only a feature of one aspect of consumer detriment (the replacement costs experienced by consumers). Clearly, there are no transfers implicit in the time and private costs of follow-up efforts by consumers (SV_Foll_Mean) — this is just a usage of real resources, not a monetary transfer. Accordingly, the net direct gain to consumers from policy reform (ignoring emotional detriments for the moment) can be calculated as:

FollowUp_share = SV_Foll_Mean / Vic_detriment * 100

Repair_share = 100 - FollowUp_share

Net_direct_gain = SV_Deterrent / 100 * Australian_detriment * (FollowUp_share / 100 + Repair_share / 100 * SV_Nontransfer_Sh / 100)

As with follow-up costs, the emotional costs experienced by consumers do not benefit businesses. Any reduction in these costs (Fairness_gain) should therefore also be added to the benefits obtained by consumers from policy change.

Fairness_gain = Avoided_Aust_detriment * SV_Fair_Premium / 100

Two measures of total detriment after policy change can then be calculated. The first is the one perceived by consumers and the other the net detriment relevant to Australians as a whole.

After_experienced = Avoided_Aust_detriment + Fairness_gain

Direct_detriment = Net_direct_gain + Fairness_gain

The perceived risk faced by consumers after policy change is calculated as a percentage reduction in the risk before policy change, using the impact factor, SV_Deterrent as above:

Risk_after = (1 - SV_Deterrent / 100) * Risk_before

Policy reform impacts invisible to consumers

Productivity growth

As discussed in chapter 14, there is likely to be an inverse relationship between the steady state productivity growth rate of the economy and the extent of consumer risk.

A simple power function was used to describe this relationship:

$$innov = \alpha - \beta r^{\gamma}$$

where *innov* is the productivity growth rate, r is the perceived consumer risk and the other factors are parameters. When risk is zero the productivity growth rate is at its highest attainable value (h), so that $\alpha = h$. Before any policy change, the steady state productivity growth (v) associated with the perceived risk ($\tilde{r}$), is just a little less than this. Using this association, the value of the beta parameter can be calculated as:

$$\beta = \frac{\alpha - v}{\tilde{r}^{\gamma}}$$

The lowest productivity growth (l) occurs at or above a maximum threshold risk rate (m), which, with substitution of the above equation for β, suggests:

$$l = \alpha - \beta m^{\gamma} = \alpha - (\alpha - v)\tilde{r}^{-\gamma} m^{\gamma} \Rightarrow \frac{(\alpha - l)}{(\alpha - v)} = \tilde{r}^{-\gamma} m^{\gamma} \text{ if } \alpha \neq v \Rightarrow$$

$$\gamma = \frac{ln(\frac{(\alpha - l)}{(\alpha - v)}}{(ln\, m - ln\, \tilde{r})} \quad \text{if } m \neq \tilde{r}$$

So, given assumed values of h, v, $\tilde{r}$ and m, it is possible to calculate α, β and γ. In that case, the post-reform steady state productivity growth rate (innov*) associated with the lower risk rate ($\hat{r}$) is:

$$innov^* = \alpha - \beta \hat{r}^{\gamma}$$

However, in some cases, the conditions that are required for well-behaved estimates of α, β and γ are absent.

In particular, it is conceivable that for certain choices of α, *l*, m, and $\tilde{r}$, $\gamma < 1$. This would lead to a convex innovation function, which would violate the reasonable

prior that there are diminishing marginal gains from reductions in risk. Now $\gamma < 1$ when:

$$\alpha > \left\{ l - \frac{vm}{\tilde{r}} \right\} \Big/ \left\{ 1 - \frac{m}{\tilde{r}} \right\}$$

This implies that this problem is just avoided when $\alpha = \left\{ l - \frac{vm}{\tilde{r}} \right\} \Big/ \left\{ 1 - \frac{m}{\tilde{r}} \right\}$, which implies resetting $h = \left\{ l - \frac{vm}{\tilde{r}} \right\} \Big/ \left\{ 1 - \frac{m}{\tilde{r}} \right\}$ when the above inequality is present. β and γ would be derived as before.

Another (very unlikely) case is where $\hat{r} = m$, in which case $innov^* = m$.

The CMOD equations that give effect to these various conditions are as follows. The base level productivity growth rate (base_TFP) is given as a difference (SV_Base_TFP_Error) from the maximum attainable productivity growth associated with zero risk (SV_Max_TFP).

base_TFP = SV_Max_TFP + SV_Base_TFP_Error

In the base case, SV_Max_TFP is set at 1.78 percent per annum, SV_Base_TFP_Error as -0.03 percent per annum, consistent with a base case productivity growth rate of 1.75 percent per year. This base case productivity growth rate was used by both the Australian Government Treasury and the Productivity Commission in their modelling of the impacts of population ageing in Australia (for example, the Intergenerational Reports). The remaining relevant values depend on what conditions hold, as described above:

If SV_Max_TFP > (SV_Lowesttfp - base_TFP * SV_Maxval_Risk / (Risk_before / 100)) / (1 - SV_Maxval_Risk / (Risk_before / 100)) Then

P_alpha = (SV_Lowesttfp - base_TFP * SV_Maxval_Risk / (Risk_before / 100)) / (1 - SV_Maxval_Risk / (Risk_before / 100))

where, as described above, P_alpha is α, SV_Max_TFP is h, SV_Lowesttfp is l, base_TFP is v, SV_Maxval_Risk is m, and Risk_before is $\tilde{r}$. If this condition does not hold (the usual case), then:

P_alpha = SV_Max_TFP.

In either case the estimate of γ (p_gamma) and β (p_beta) is respectively:

p_gamma = (Log((P_alpha - SV_Lowesttfp) / (P_alpha - base_TFP))) / (Log(SV_Maxval_Risk) - Log((Risk_before / 100)))

p_beta = (P_alpha - SV_Lowesttfp) / (SV_Maxval_Risk ^ p_gamma)

This implies that:

New_TFP = P_alpha - p_beta * (Risk_after / 100) ^ p_gamma

where New_TFP is innov* and $\hat{r}$ is the risk after policy reform (Risk_after).

The exception is when $\hat{r} \geq m$ (that is, Risk_after / 100 >= SV_Maxval_Risk) in which case:

New_TFP = SV_Lowesttfp

GDP and consumption estimates

Changes in GDP per capita (GDP/POP) can be readily decomposed:

$$\frac{(GDP_t/POP_t)}{(GDP_{t-1}/POP_{t-1})} = \frac{(L_t/POP_t)}{(L_{t-1}/POP_{t-1})} \times \frac{(GDP_t/L_t)}{(GDP_{t-1}/L_{t-1})} \Rightarrow y^* = (l^* + 1) \times (innov + 1) - 1$$

where L is labour inputs, y* is the percentage growth in GDP per capita, l^* is the percentage growth in labour input per capita and *innov* is the annual percentage growth rate in labour productivity. The former series is available from the Productivity Commission's model of future labour inputs (MoDEM), developed as part of the Commission's prior research into population ageing (Productivity Commission 2005) and further developed in recent modelling (Cuxson et al. 2007).

Similarly, GDP can be similarly decomposed as:

$$\frac{GDP_t}{GDP_{t-1}} = \frac{(GDP_t/POP_t)}{(GDP_{t-1}/POP_{t-1})} \times \frac{POP_t}{POP_{t-1}} \Rightarrow (GDP_t) = GDP_{t-1} \times (y^* + 1) \times (POP^* + 1)$$

where POP* is the percentage growth in population (also from MoDEM).

Accordingly, by combining existing forecasts of *l**, *POP** and with the estimates of *innov* discussed above, it is possible to derive a series of future GDP before and after the policy change. In turn, since consumption provides the appropriate measure of the value of future output to Australians, the GDP estimates are converted to consumption values using the long-run consumption share of GDP. In this case, consumption is appropriately measured as household and government consumption expenditure. The CMOD code that provides these forecasts is split into several modules. The first sets up initial values (noting that the GDP value in 2007-08 is the same both before and after the policy change since the policy is assumed to take effect in the following year):

```
GDPCAP_before(0)=100*((1+D_Labhours(0)/100)*(1+base_TFP/100)-1)  '2007-08
GDPCAP_after(0)=100*((1+D_Labhours(0)/100)*(1+base_TFP/100)-1)  '2007-08
GDP_before(0)=Gdp_2006_07*(GDPCAP_before(0)/100+1)*(D_Pop(0)/100+1)  '2007-08
GDP_after(0)=Gdp_2006_07*(GDPCAP_after(0)/100+1)*(D_Pop(0)/100+1)  '2007-08
CONS_before(0)=Lr_Con_Sh*GDP_before(0)  '2007-08
CONS_after(0)=Lr_Con_Sh*GDP_after(0)  '2007-08
GDPCAP_before(1)=100*((1+D_Labhours(1)/100)*(1+base_TFP/100)-1)   'ie2008-09
GDPCAP_after(1)=100*((1+D_Labhours(1)/100)*(1+New_TFP/100)-1)  '2008-09
GDP_before(1)=GDP_before(0)*(GDPCAP_before(1)/100+1)*(D_Pop(1)/100+1)   '2008-09
GDP_after(1)=GDP_after(0)*(GDPCAP_after(1)/100+1)*(D_Pop(1)/100+1)'2008-09
CONS_before(1)=Lr_Con_Sh*GDP_before(1)   '2008-09
CONS_after(1)=Lr_Con_Sh*GDP_after(1)  '2008-09
```

The second module then provides future values, using the base values of the GDP measures as their initial values.

```
For Period = 2009 To (2007 + Yrs)
    t=Period-2007
    GDPCAP_before(t)=100*((1+D_Labhours(t)/100)*(1+base_TFP/100)-1)
    GDPCAP_after(t)=100*((1+D_Labhours(t)/100)*(1+New_TFP/100)-1)
    GDP_before(t) = GDP_before(t-1)*(GDPCAP_before(t)/100+1)*(D_Pop(t)/100+1)
    GDP_after(t)=GDP_after(t-1)*(GDPCAP_after(t)/100+1)*(D_Pop(t)/100+1)
    CONS_before(t)=Lr_Con_Sh*GDP_before(t)
    CONS_after(t)=Lr_Con_Sh*GDP_after(t)
Next period
```

The future values of the difference in consumption stemming from the policy change are then discounted to give a present value of the gains.

Transaction cost savings

CMOD has a simple approach to modelling transaction costs. Transaction costs borne by consumers (TransCosts) are a function of the risks they face (as well as an additional set of variables **x**):

$$\text{TransCosts} = \alpha \times \text{RISK}^{\varepsilon} \,.\, \mathbf{x}^{\beta}$$

If the **x** variables do not change, then transaction costs before and after a change in consumer risk is:

$$TransCosts_{before} = \alpha \times RISK_{before}^{\varepsilon}.\boldsymbol{x}^{\boldsymbol{\beta}} \quad and \quad TransCosts_{after} = \alpha \times RISK_{after}^{\varepsilon}.\boldsymbol{x}^{\boldsymbol{\beta}} \Rightarrow$$

$$TransCosts_{after} = TransCosts_{before} \times \left(\frac{RISK_{after}}{RISK_{before}} \right)^{\varepsilon}$$

In the first year of the policy change it is then possible to calculate the transaction share of household consumption and its value had no policy change taken place.

$$TransShare_{before} = \frac{TransCosts_{before}}{C_{before}} \quad and \quad TransShare_{after} = \frac{TransCosts_{before}}{C_{after}}$$

These ratios are then applied to subsequent values of household consumption to estimate savings in transactions costs over the long run. These savings are then discounted to give a present value.

1.6 Compliance and administrative cost effects

These play a much less important role in CMOD compared with the savings in detriment, innovation gains and transaction cost efficiencies. And unlike these three factors, policy reform reduces welfare in some instances, as it:

- increases some compliance costs (though there are net savings in compliance costs overall) and;
- introduces some new administrative costs associated with the change in the consumer policy regime.

CMOD has a comprehensive approach to such compliance and administrative costs (and savings). It:

- distinguishes between ongoing and transitional compliance and administrative costs;
- takes account of compliance cost savings associated with more national consumer laws (and from improvements in disclosure) for seven firm size categories and for eight different combinations of the extent of interstate activity by enterprises (eg only one state, two states etc). These savings are based on modest assumed gains that grow more significant in absolute size with enterprise employment and for firms that have a presence in more than one jurisdiction;
- estimates additional litigation costs for both governments and firms;
- considers the costs of regulatory actions taken by government are incorrect (false positive actions in courts);

- estimates costs associated with additional resourcing of regulators and the temporary disruption costs that occur when the regulation of credit and related matters are moved from the States to the Australian Government level;
- makes explicit assumptions about the administrative costs of unfair contracts (and their compliance burdens for firms); and
- takes account of the economic distortions arising from tax financing of the (small) additional expenditure by governments.

Most of the above features of compliance modelling of CMOD are described in table 1.1. However, one aspect, the compliance savings by firm size and jurisdictional presence, is less clearcut. In order to reduce the number of ad hoc assumptions underlying this feature of CMOD and to allow users to make different assumptions using just a few parameters, the Commission used a beta function to describe such cost savings. CMOD allows users to easily scale up or down the cost savings by altering the scale_compliance_firmsize variable in the 'Start Here' worksheet. It is also possible to vary the key characteristics of the underlying function describing costs by size and jurisdictional presence by removing the sheet protection on the worksheet labelled 'Fixed variables' (the password is cmod) and altering the variables high_emp, alpha_emp and beta_emp.

Key variables

The CMOD results are more sensitive to changes in some variables than others. The most important variables from this perspective are:

- The years over which the gains occur (YRS)
- The discount rate (Discount_rate)
- Transaction costs as a share of total household consumption (Trans_share)
- Bias in the CAV survey results (bias_adjustment)
- Consumer sensitivity to risk (risk_adj)
- The proportional increase in detriment arising from psychic costs (Fair_premium)
- The deterrent effects of policy reform on firm misconduct (Deterrent); and
- The steady state productivity growth rate in the base case (determined through Base_TFP_error)

1.7 Conclusion

It should be emphasised that CMOD provides an *experimental* assessment of the cost and benefits of consumer reform. It has the advantage that it sets out multiple detailed mechanisms through which reforms can act on the economy. As much as possible it provides a coherent structure in which assumptions sit.

Appendix: Full code of the model

Comments are preceded by a ‘

FRMCALC code

```
Public AvEmp As Range ' Note use of range object - and note first elemnt is {1,1}, not {0,0} as in conventional arrays in VBA
' The vector is the average employment size of enterprises of given employment ranges. There are 7 size groupings (non-employ, 1-4, 5-19, 20-49, 50-99, 100-199, 200+)
Public Enterprises As Range ' Note use of range object - and note first elemnt is {1,1}, not {0,0} as in conventional arrays in VBA
' The array is the number of enterprises by size and number of states they operate in. There are
' 8 jurisdictions from 1 to all; and 7 size groupings (non-employ, 1-4, 5-19, 20-49, 50-99, 100-199, 200+)

Public P_alpha, p_gamma, p_beta, New_TFP, TFP_BASE As Double
Public Yrs, iters, i, j, simdum, Badvar As Integer
Public Actions, Alpha_Emp, Aust_pop16, Beta, Beta_Emp, CON2005_06, Gdp_2006_07, High_Emp As Double
Public Low_Emp, Lr_Con_Sh, Max, Ratio_Austvic, Saving_Large, Share_Foreign, Share_Success As Double
Public Shift_Factor, Vic_pop16, Welfare_Weight As Double
Public No_Emp, Emp_Sm100, Emp_Gt100 As Double ' Defines variables for enterprise numbers for non-employers, smaller and larger businesses
Public LR_Total_Cons_Share As Double
Public base_risk As Double

'dimension endogenous variables
Public Vic_detriment  As Double
Public Australian_detriment As Double
Public Experienced_Australian_detriment As Double
Public Risk_before As Double
Public FollowUp_share As Double
Public Repair_share As Double
Public Avoided_Aust_detriment As Double
Public Net_direct_gain As Double
Public Fairness_gain As Double
Public After_experienced As Double
Public Direct_detriment As Double
Public Risk_after As Double
'*
'Dimension benefit and costs variables
Public Gross_gain_2008_09, Transaction_costs, Welfare_effect, a_value, Transact_dividend_rate As Double
Public emp_max, xmax, xsmall, scalar As Double
Public OneOff_compliance_burdens, Annuity_oneoff, Comply_ratio As Double
Public New_Actions, Transfer_fromOS, False_cases, Cost_false, Reg_costs, Bus_costs, litigation_costs, Policy_costs As Double
Public admin_ratio, Costs_negotiate, Credit_move_costs, One_off_legal_costs As Double
Public Oneoff_admin As Double
Public PV_Net_direct_Gain, PV_Innovation_gain, PV_Transact, PV_Comp_Savings, PV_Litigation_Policy_costs As Double
Public PV_Gross_Compliance_Savings, PV_Agency_Costs As Double
```

Public Net_Gain_2008_09, PV_Net_Gain, AN_Net_Gain, Net_Compliance_Gains, PV_Net_Compliance_Gains, AN_Net_Compliance_Gains As Double

Public Net_Ongoing_and_Oneoff_Comp_Savings, PV_Net_Ongoing_and_Oneoff_Comp_Savings, AN_Net_Ongoing_and_Oneoff_Comp_Savings As Double

Public Unfair_Admin_Costs As Double

Public states As Integer

Public AN_Net_direct_Gain, AN_Innovation_gain, AN_Transact, AN_Comp_Savings, AN_Litigation_Policy_costs As Double

'Dimension stochastic input variables

Public SD_Av_Fine, SD_Base_TFP_Error, SD_Bias_Adjustment, SD_Risk_Adj, SD_Business_Margin, SD_Cost_Elasticity, SD_Cost_Per_Day, SD_Days_Case As Double

Public SD_Deterrent, SD_Discount_Rate, SD_Fair_Premium, SD_False_Positive, SD_Foll_Mean, SD_Legal_Oneoff_Ft, SD_Legal_Oneoff_Wage As Double

Public SD_Lowesttfp, SD_Max_TFP, SD_Maxval_Risk, SD_Meb, SD_Multiplier_By_State, SD_Nontransfer_Sh, SD_Oft_Asic_Ft As Double

Public SD_Oft_Asic_Wage, SD_Percent_Actions, SD_Policy_Ft, SD_Policy_Wage, SD_Preparation_Costs, SD_Rep_Mean, SD_Scale_Compliance_Firmsize As Double

Public SD_Template_Ft, SD_Template_Wage, SD_Unfairc_Big, SD_Unfairc_Noemp, SD_Unfairc_Small As Double

Public SD_Wastage, SD_Yearly_Comp_Big, SD_Yearly_Comp_Noemp, SD_Yearly_Comp_Small, SD_Ongoing_Admin_Unfair, SD_Trans_share As Double

Public SV_Av_Fine, SV_Base_TFP_Error, SV_Bias_Adjustment, SV_Risk_Adj, SV_Business_Margin, SV_Cost_Elasticity, SV_Cost_Per_Day, SV_Days_Case As Double

Public SV_Deterrent, SV_Discount_Rate, SV_Fair_Premium, SV_False_Positive, SV_Foll_Mean, SV_Legal_Oneoff_Ft, SV_Legal_Oneoff_Wage As Double

Public SV_Lowesttfp, SV_Max_TFP, SV_Maxval_Risk, SV_Meb, SV_Multiplier_By_State, SV_Nontransfer_Sh, SV_Oft_Asic_Ft As Double

Public SV_Oft_Asic_Wage, SV_Percent_Actions, SV_Policy_Ft, SV_Policy_Wage, SV_Preparation_Costs, SV_Rep_Mean, SV_Scale_Compliance_Firmsize As Double

Public SV_Template_Ft, SV_Template_Wage, SV_Unfairc_Big, SV_Unfairc_Noemp, SV_Unfairc_Small As Double

Public SV_Wastage, SV_Yearly_Comp_Big, SV_Yearly_Comp_Noemp, SV_Yearly_Comp_Small, SV_Ongoing_Admin_Unfair, SV_Trans_share As Double

Public Lower_Av_Fine, Lower_Base_TFP_Error, Lower_Bias_Adjustment, Lower_Risk_Adj, Lower_Business_Margin, Lower_Cost_Elasticity, Lower_Cost_Per_Day, Lower_Days_Case As Double

Public Lower_Deterrent, Lower_Discount_Rate, Lower_Fair_Premium, Lower_False_Positive, Lower_Foll_Mean, Lower_Legal_Oneoff_Ft, Lower_Legal_Oneoff_Wage As Double

Public Lower_Lowesttfp, Lower_Max_TFP, Lower_Maxval_Risk, Lower_Meb, Lower_Multiplier_By_State, Lower_Nontransfer_Sh, Lower_Oft_Asic_Ft As Double

Public Lower_Oft_Asic_Wage, Lower_Percent_Actions, Lower_Policy_Ft, Lower_Policy_Wage, Lower_Preparation_Costs, Lower_Rep_Mean, Lower_Scale_Compliance_Firmsize As Double

Public Lower_Template_Ft, Lower_Template_Wage, Lower_Unfairc_Big, Lower_Unfairc_Noemp, Lower_Unfairc_Small As Double

Public Lower_Wastage, Lower_Yearly_Comp_Big, Lower_Yearly_Comp_Noemp, Lower_Yearly_Comp_Small, Lower_Ongoing_Admin_Unfair, Lower_Trans_share As Double

Public Upper_Av_Fine, Upper_Base_TFP_Error, Upper_Bias_Adjustment, Upper_Risk_Adj, Upper_Business_Margin, Upper_Cost_Elasticity, Upper_Cost_Per_Day, Upper_Days_Case As Double

Public Upper_Deterrent, Upper_Discount_Rate, Upper_Fair_Premium, Upper_False_Positive, Upper_Foll_Mean, Upper_Legal_Oneoff_Ft, Upper_Legal_Oneoff_Wage As Double

Public Upper_Lowesttfp, Upper_Max_TFP, Upper_Maxval_Risk, Upper_Meb, Upper_Multiplier_By_State, Upper_Nontransfer_Sh, Upper_Oft_Asic_Ft As Double

Public Upper_Oft_Asic_Wage, Upper_Percent_Actions, Upper_Policy_Ft, Upper_Policy_Wage, Upper_Preparation_Costs, Upper_Rep_Mean, Upper_Scale_Compliance_Firmsize As Double

Public Upper_Template_Ft, Upper_Template_Wage, Upper_Unfairc_Big, Upper_Unfairc_Noemp, Upper_Unfairc_Small As Double

```
Public Upper_Wastage, Upper_Yearly_Comp_Big, Upper_Yearly_Comp_Noemp, Upper_Yearly_Comp_Small,
    Upper_Ongoing_Admin_Unfair, Upper_Trans_share As Double
Public Beta_Av_Fine, Beta_Base_TFP_Error, Beta_Bias_Adjustment, Beta_Risk_Adj, Beta_Business_Margin,
    Beta_Cost_Elasticity, Beta_Cost_Per_Day, Beta_Days_Case As Double
Public Beta_Deterrent, Beta_Discount_Rate, Beta_Fair_Premium, Beta_False_Positive, Beta_Foll_Mean,
    Beta_Legal_Oneoff_Ft, Beta_Legal_Oneoff_Wage  As Double
Public Beta_Lowesttfp, Beta_Max_TFP, Beta_Maxval_Risk, Beta_Meb, Beta_Multiplier_By_State,
    Beta_Nontransfer_Sh, Beta_Oft_Asic_Ft  As Double
Public Beta_Oft_Asic_Wage, Beta_Percent_Actions, Beta_Policy_Ft, Beta_Policy_Wage, Beta_Preparation_Costs,
    Beta_Rep_Mean, Beta_Scale_Compliance_Firmsize  As Double
Public Beta_Template_Ft, Beta_Template_Wage, Beta_Unfairc_Big, Beta_Unfairc_Noemp, Beta_Unfairc_Small As
    Double
Public Beta_Wastage, Beta_Yearly_Comp_Big, Beta_Yearly_Comp_Noemp, Beta_Yearly_Comp_Small,
    Beta_Ongoing_Admin_Unfair, Beta_Trans_share As Double
Public alpha_Av_Fine, alpha_Max_TFP, alpha_Bias_Adjustment, alpha_Business_Margin, alpha_Cost_Elasticity,
    alpha_Cost_Per_Day, alpha_Days_Case, alpha_Deterrent, alpha_Discount_Rate, alpha_Fair_Premium,
    alpha_False_Positive, alpha_Foll_Mean, alpha_Legal_Oneoff_Ft, alpha_Legal_Oneoff_Wage, alpha_Lowesttfp,
    alpha_Base_TFP_Error, alpha_Maxval_Risk, alpha_Meb, alpha_Multiplier_By_State, alpha_Nontransfer_Sh,
    alpha_Oft_Asic_Ft, alpha_Oft_Asic_Wage, alpha_Percent_Actions, alpha_Policy_Ft, alpha_Policy_Wage,
    alpha_Preparation_Costs, alpha_Rep_Mean, alpha_Risk_Adj, alpha_Scale_Compliance_Firmsize,
    alpha_Template_Ft, alpha_Template_Wage, alpha_Unfairc_Big, alpha_Unfairc_Noemp, alpha_Unfairc_Small,
    alpha_Wastage, alpha_Yearly_Comp_Big, alpha_Yearly_Comp_Noemp, alpha_Yearly_Comp_Small,
    alpha_Ongoing_Admin_Unfair, alpha_Trans_share As Double

Sub Dimvar()
'* arrays
Dim D_Labhours() As Double ' creates dynamic array
Dim D_Pop() As Double
Dim YR_Net_direct_gain(), Innovation_gain(), Transact(), Comp_Savings(), Litigation_Policy_costs() As Double
Dim Gross_compliance_savings() As Double
Dim agency_costs() As Double
End Sub

Private Sub CmdCancel_Click()
Unload Me
End Sub

Private Sub CmdCancel2_Click()
Unload Me
End Sub

Private Sub CmdOK_Click()
If OptignoreUn.Value = True Then
    Call killsheets
    Call BasicCalc
    Call NoSim
Else:
    MultiPageCalc.page2.Visible = True
    MultiPageCalc.Value = 1
End If


End Sub
```

```
Private Sub CmdStartCalc_Click()

'deletes old histogram data and sim print so new data can overwrite
Call killsheets

' now call key procedures
Call BasicCalc
Call sim
End Sub

Sub NoSim()
Sheets("Summary").Select
Range("A74:G83").Select
Selection.ClearContents
Range("A75:G83").Select
Selection.Interior.ColorIndex = xlNone
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Application.Goto Reference:="summary_sheet"

CmdCancel2 = True
End Sub

Private Sub FrameRestore_Click()
End Sub

Private Sub Lbl_graph_Click()
End Sub

Private Sub MultiPageCalc_Change()
End Sub

Private Sub OptButYesRestore_Click()
'
'Restores default values
FrmWarning.Show
End Sub

Private Sub OptDontShow_Click()
End Sub

Private Sub OptNo_Click()
End Sub
```

```
Private Sub OptShow_Click()
End Sub

Private Sub OptYes_Click()
End Sub

Private Sub SimGraph_Click()
End Sub


Private Sub UserForm_Click()
Call Dimvar
End Sub

Sub Hist(M As Long, arr As Variant, rows As Variant, PrintCellStart As Integer) ' this undertakes the calculations for the
    histogram
Dim i As Long, j As Long
Dim Length As Single
ReDim breaks(M) As Single
ReDim freq(M) As Single
For j = 1 To M
    freq(j) = 0
Next j
highbound = Application.Max(arr)
lowbound = Application.Min(arr)
Length = (highbound - lowbound) / M

For j = 1 To M
    breaks(j) = lowbound + Length * j
Next j

For i = 0 To rows - 1
    If (arr(i) <= breaks(1)) Then freq(1) = freq(1) + 1
    If (arr(i) >= breaks(M - 1)) Then freq(M) = freq(M) + 1
    For j = 2 To M - 1
        If (arr(i) > breaks(j - 1) And arr(i) <= breaks(j)) Then freq(j) = freq(j) + 1
    Next j
Next i

Application.ScreenUpdating = False
PrintCellStart = PrintCellStart * 2 - 1
For i = 1 To M
    Cells(i + 1, PrintCellStart) = breaks(i)
    Cells(i + 1, 1 + PrintCellStart) = freq(i)
Next i
End Sub

Sub ArrangeMyCharts()
Dim iChart As Long
Dim nCharts As Long
```

```
Dim dTop As Double
Dim dLeft As Double
Dim dHeight As Double
Dim dWidth As Double
Dim nColumns As Long

dTop = 180  ' top of first row of charts
dLeft = 20' left of first column of charts
dHeight = 225  ' height of all charts
dWidth = 250   ' width of all charts
nColumns = 3   ' number of columns of charts
nCharts = ActiveSheet.ChartObjects.Count
Application.ScreenUpdating = False

For iChart = 1 To nCharts
With ActiveSheet.ChartObjects(iChart)
    .Height = dHeight
    .Width = dWidth
    .Top = dTop + Int((iChart - 1) / nColumns) * dHeight
    .Left = dLeft + ((iChart - 1) Mod nColumns) * dWidth
End With

With ActiveChart.ChartArea
    .Border.Weight = 2
    .Border.LineStyle = 0
    .Interior.ColorIndex = xlAutomatic
    .Font.Size = 8
    .Font.name = "Arial"
End With
Next

End Sub

Function wkshExists(name As String) As Boolean
Dim wksh As Worksheet
On Error GoTo errs
Set wksh = Worksheets(name)
wkshExists = True
Exit Function
errs:
wkshExists = False
End Function

Function Graph_add(name As String, cell_ref As String)
'
Application.ScreenUpdating = False
ActiveWorkbook.Names.Add name:=name, RefersToR1C1:= _
"='Histogram Data'!" & cell_ref

Charts.Add
```

```
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData Source:=Sheets("Histogram Data").Range(name), _
PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).name = "='Histogram Data'!R1C2"
ActiveChart.Location Where:=xlLocationAsObject, name:="Histogram Data"
With ActiveChart
    .Legend.Delete
    .HasTitle = False
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = name
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "frequency"
End With

With ActiveChart.Axes(xlValue)
    .MajorGridlines.Delete
    .MinimumScale = 0
    .MaximumScaleIsAuto = True
    .MinorUnitIsAuto = True
    .MajorUnitIsAuto = True
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
End With

With ActiveChart.PlotArea.Border
    .ColorIndex = 1
    .Weight = xlHairline
    .LineStyle = xlContinuous
End With

ActiveChart.PlotArea.Interior.ColorIndex = xlNone

With ActiveChart.SeriesCollection(1)
    .Border.ColorIndex = 1
    .Border.Weight = xlHairline
    .Border.LineStyle = xlContinuous
    .MarkerBackgroundColorIndex = xlNone
    .MarkerForegroundColorIndex = xlNone
    .MarkerStyle = xlNone
    .Smooth = True
    .MarkerSize = 3
End With

With ActiveChart.Parent
    .Left = 100
    .Width = 250
    .Top = 150
    .Height = 225
```

```
End With

With ActiveChart.ChartArea
    .Border.Weight = 2
    .Border.LineStyle = 0
    .Interior.ColorIndex = xlAutomatic
    .Font.Size = 8
    .Font.name = "Arial"
End With

ActiveChart.ChartArea.Select
End Function

Private Sub killsheets()
If wkshExists("histogram data") Then
    Application.DisplayAlerts = False
    Sheets("histogram data").Select
    ActiveWindow.SelectedSheets.Delete
    ActiveWorkbook.Names("freq_det").Delete
    ActiveWorkbook.Names("freq_trans").Delete
    ActiveWorkbook.Names("freq_innov").Delete
    ActiveWorkbook.Names("freq_comply").Delete
    ActiveWorkbook.Names("freq_admin").Delete
    ActiveWorkbook.Names("freq_netgain").Delete
    Application.DisplayAlerts = True
End If

If wkshExists("sim_print") Then
    Application.DisplayAlerts = False
    Sheets("sim_print").Select
    ActiveWindow.SelectedSheets.Delete
    Application.DisplayAlerts = True
End If
End Sub

Public Sub BasicCalc()
Actions = Range("Actions").Value
Alpha_Emp = Range("alpha_emp").Value
Aust_pop16 = Range("AUST_POP16").Value
Beta_Emp = Range("beta_emp").Value
CON2005_06 = Range("CON2005_06").Value
Gdp_2006_07 = Range("GDP_2006_07").Value
High_Emp = Range("high_emp").Value
Low_Emp = Range("low_emp").Value
LR_Total_Cons_Share = Range("LR_Total_Cons_Share").Value
Lr_Con_Sh = Range("LR_con_sh").Value
Max = Range("Max").Value
Saving_Large = Range("saving_large").Value
Share_Foreign = Range("Share_foreign").Value
Share_Success = Range("share_success").Value
```

```
Shift_Factor = Range("shift_factor").Value
Vic_pop16 = Range("VIC_POP16").Value
Welfare_Weight = Range("welfare_weight").Value
Yrs = Range("Yrs").Value
Ratio_Austvic = Aust_pop16 / Vic_pop16
No_Emp = Range("No_emp").Value
Emp_Sm100 = Range("Emp_Sm100").Value
Emp_Gt100 = Range("Emp_Gt100").Value

ReDim D_Labhours(Yrs + 1) As Double ' ie yrs rows (since row zero is defined as the first)
ReDim D_Pop(Yrs + 1) As Double

Set Enterprises = Range("ENT_NEMP:ENT_emp200_allStates")
Set AvEmp = Range("Small_avemp:Large_avemp")

For Period = 2008 To (2008 + Yrs) ' IE "yrs" IS THE NUMBER OF YEARS, STARTING WITH 2008, WHICH IS THE
    2007-08 fiscal year
    i = Period - 2008 'IE RESET THE COUNTER TO ZERO FOR THE FIRST ELEMENT OF THE VECTOR
    D_Labhours(i) = Worksheets("Fixed Variables").Cells(i + 48, 2).Value
    D_Pop(i) = Worksheets("Fixed Variables").Cells(i + 48, 3).Value
Next Period

'*********************************************************************
' read in stochastic variables mean values and limits in simulation
'*********************************************************************
SV_Av_Fine = Range("Av_fine").Value
SV_Base_TFP_Error = Range("Base_TFP_Error").Value
SV_Bias_Adjustment = Range("Bias_adjustment").Value
SV_Business_Margin = Range("Business_margin").Value
SV_Cost_Elasticity = Range("cost_elasticity").Value
SV_Cost_Per_Day = Range("Cost_per_day").Value
SV_Days_Case = Range("Days_case").Value
SV_Deterrent = Range("Deterrent").Value
SV_Discount_Rate = Range("Discount_rate").Value
SV_Fair_Premium = Range("Fair_premium").Value
SV_False_Positive = Range("False_positive").Value
SV_Foll_Mean = Range("Foll_mean").Value
SV_Legal_Oneoff_Ft = Range("LEGAL_ONEOFF_FT").Value
SV_Legal_Oneoff_Wage = Range("LEGAL_ONEOFF_WAGE").Value
SV_Lowesttfp = Range("LowestTFP").Value
SV_Max_TFP = Range("Max_TFP").Value
SV_Maxval_Risk = Range("Maxval_risk").Value
SV_Meb = Range("MEB").Value
SV_Multiplier_By_State = Range("multiplier_by_state").Value
SV_Nontransfer_Sh = Range("Nontransfer_sh").Value
SV_Oft_Asic_Ft = Range("OFT_ASIC_FT").Value
SV_Oft_Asic_Wage = Range("OFT_ASIC_WAGE").Value
SV_Percent_Actions = Range("Percent_actions").Value
SV_Policy_Ft = Range("POLICY_FT").Value
SV_Policy_Wage = Range("POLICY_WAGE").Value
```

```
SV_Preparation_Costs = Range("Preparation_costs").Value
SV_Rep_Mean = Range("Rep_mean").Value
SV_Risk_Adj = Range("Risk_Adj").Value
SV_Scale_Compliance_Firmsize = Range("scale_compliance_firmsize").Value
SV_Template_Ft = Range("TEMPLATE_FT").Value
SV_Template_Wage = Range("Template_wage").Value
SV_Unfairc_Big = Range("UnfairC_big").Value
SV_Unfairc_Noemp = Range("UnfairC_noemp").Value
SV_Unfairc_Small = Range("UnfairC_small").Value
SV_Wastage = Range("Wastage").Value
SV_Yearly_Comp_Big = Range("Yearly_Comp_Big").Value
SV_Yearly_Comp_Noemp = Range("yearly_comp_noemp").Value
SV_Yearly_Comp_Small = Range("Yearly_comp_small").Value
SV_Ongoing_Admin_Unfair = Range("Ongoing_Admin_Unfair").Value
SV_Trans_share = Range("Trans_share").Value

Lower_Av_Fine = Range("Lower_Av_fine").Value
Lower_Base_TFP_Error = Range("Lower_Base_TFP_Error").Value
Lower_Bias_Adjustment = Range("Lower_Bias_adjustment").Value
Lower_Business_Margin = Range("Lower_Business_margin").Value
Lower_Cost_Elasticity = Range("Lower_cost_elasticity").Value
Lower_Cost_Per_Day = Range("Lower_Cost_per_day").Value
Lower_Days_Case = Range("Lower_Days_case").Value
Lower_Deterrent = Range("Lower_Deterrent").Value
Lower_Discount_Rate = Range("Lower_Discount_rate").Value
Lower_Fair_Premium = Range("Lower_Fair_premium").Value
Lower_False_Positive = Range("Lower_False_positive").Value
Lower_Foll_Mean = Range("Lower_Foll_mean").Value
Lower_Legal_Oneoff_Ft = Range("Lower_LEGAL_ONEOFF_FT").Value
Lower_Legal_Oneoff_Wage = Range("Lower_LEGAL_ONEOFF_WAGE").Value
Lower_Lowesttfp = Range("Lower_LowestTFP").Value
Lower_Max_TFP = Range("Lower_Max_TFP").Value
Lower_Maxval_Risk = Range("Lower_Maxval_risk").Value
Lower_Meb = Range("Lower_MEB").Value
Lower_Multiplier_By_State = Range("Lower_multiplier_by_state").Value
Lower_Nontransfer_Sh = Range("Lower_Nontransfer_sh").Value
Lower_Oft_Asic_Ft = Range("Lower_OFT_ASIC_FT").Value
Lower_Oft_Asic_Wage = Range("Lower_OFT_ASIC_WAGE").Value
Lower_Percent_Actions = Range("Lower_Percent_actions").Value
Lower_Policy_Ft = Range("Lower_POLICY_FT").Value
Lower_Policy_Wage = Range("Lower_POLICY_WAGE").Value
Lower_Preparation_Costs = Range("Lower_Preparation_costs").Value
Lower_Rep_Mean = Range("Lower_Rep_mean").Value
Lower_Risk_Adj = Range("Lower_Risk_Adj").Value
Lower_Scale_Compliance_Firmsize = Range("Lower_scale_compliance_firmsize").Value
Lower_Template_Ft = Range("Lower_TEMPLATE_FT").Value
Lower_Template_Wage = Range("Lower_Template_wage").Value
Lower_Unfairc_Big = Range("Lower_UnfairC_big").Value
Lower_Unfairc_Noemp = Range("Lower_UnfairC_noemp").Value
Lower_Unfairc_Small = Range("Lower_UnfairC_small").Value
```

```
Lower_Wastage = Range("Lower_Wastage").Value
Lower_Yearly_Comp_Big = Range("Lower_Yearly_Comp_Big").Value
Lower_Yearly_Comp_Noemp = Range("Lower_yearly_comp_noemp").Value
Lower_Yearly_Comp_Small = Range("Lower_Yearly_comp_small").Value
Lower_Ongoing_Admin_Unfair = Range("Lower_Ongoing_Admin_Unfair").Value
Lower_Trans_share = Range("Lower_Trans_share").Value

Upper_Av_Fine = Range("Upper_Av_fine").Value
Upper_Base_TFP_Error = Range("Upper_Base_TFP_Error").Value
Upper_Bias_Adjustment = Range("Upper_Bias_adjustment").Value
Upper_Business_Margin = Range("Upper_Business_margin").Value
Upper_Cost_Elasticity = Range("Upper_cost_elasticity").Value
Upper_Cost_Per_Day = Range("Upper_Cost_per_day").Value
Upper_Days_Case = Range("Upper_Days_case").Value
Upper_Deterrent = Range("Upper_Deterrent").Value
Upper_Discount_Rate = Range("Upper_Discount_rate").Value
Upper_Fair_Premium = Range("Upper_Fair_premium").Value
Upper_False_Positive = Range("Upper_False_positive").Value
Upper_Foll_Mean = Range("Upper_Foll_mean").Value
Upper_Legal_Oneoff_Ft = Range("Upper_LEGAL_ONEOFF_FT").Value
Upper_Legal_Oneoff_Wage = Range("Upper_LEGAL_ONEOFF_WAGE").Value
Upper_Lowesttfp = Range("Upper_LowestTFP").Value
Upper_Max_TFP = Range("Upper_Max_TFP").Value
Upper_Maxval_Risk = Range("Upper_Maxval_risk").Value
Upper_Meb = Range("Upper_MEB").Value
Upper_Multiplier_By_State = Range("Upper_multiplier_by_state").Value
Upper_Nontransfer_Sh = Range("Upper_Nontransfer_sh").Value
Upper_Oft_Asic_Ft = Range("Upper_OFT_ASIC_FT").Value
Upper_Oft_Asic_Wage = Range("Upper_OFT_ASIC_WAGE").Value
Upper_Percent_Actions = Range("Upper_Percent_actions").Value
Upper_Policy_Ft = Range("Upper_POLICY_FT").Value
Upper_Policy_Wage = Range("Upper_POLICY_WAGE").Value
Upper_Preparation_Costs = Range("Upper_Preparation_costs").Value
Upper_Rep_Mean = Range("Upper_Rep_mean").Value
Upper_Risk_Adj = Range("Upper_Risk_Adj").Value
Upper_Scale_Compliance_Firmsize = Range("Upper_scale_compliance_firmsize").Value
Upper_Template_Ft = Range("Upper_TEMPLATE_FT").Value
Upper_Template_Wage = Range("Upper_Template_wage").Value
Upper_Unfairc_Big = Range("Upper_UnfairC_big").Value
Upper_Unfairc_Noemp = Range("Upper_UnfairC_noemp").Value
Upper_Unfairc_Small = Range("Upper_UnfairC_small").Value
Upper_Wastage = Range("Upper_Wastage").Value
Upper_Yearly_Comp_Big = Range("Upper_Yearly_Comp_Big").Value
Upper_Yearly_Comp_Noemp = Range("Upper_yearly_comp_noemp").Value
Upper_Yearly_Comp_Small = Range("Upper_Yearly_comp_small").Value
Upper_Ongoing_Admin_Unfair = Range("Upper_Ongoing_Admin_Unfair").Value
Upper_Trans_share = Range("Upper_Trans_share").Value

Beta_Av_Fine = Range("Beta_Av_fine").Value
Beta_Base_TFP_Error = Range("Beta_Base_TFP_Error").Value
```

Beta_Bias_Adjustment = Range("Beta_Bias_adjustment").Value
Beta_Business_Margin = Range("Beta_Business_margin").Value
Beta_Cost_Elasticity = Range("Beta_cost_elasticity").Value
Beta_Cost_Per_Day = Range("Beta_Cost_per_day").Value
Beta_Days_Case = Range("Beta_Days_case").Value
Beta_Deterrent = Range("Beta_Deterrent").Value
Beta_Discount_Rate = Range("Beta_Discount_rate").Value
Beta_Fair_Premium = Range("Beta_Fair_premium").Value
Beta_False_Positive = Range("Beta_False_positive").Value
Beta_Foll_Mean = Range("Beta_Foll_mean").Value
Beta_Legal_Oneoff_Ft = Range("Beta_LEGAL_ONEOFF_FT").Value
Beta_Legal_Oneoff_Wage = Range("Beta_LEGAL_ONEOFF_WAGE").Value
Beta_Lowesttfp = Range("Beta_LowestTFP").Value
Beta_Max_TFP = Range("Beta_Max_TFP").Value
Beta_Maxval_Risk = Range("Beta_Maxval_risk").Value
Beta_Meb = Range("Beta_MEB").Value
Beta_Multiplier_By_State = Range("Beta_multiplier_by_state").Value
Beta_Nontransfer_Sh = Range("Beta_Nontransfer_sh").Value
Beta_Oft_Asic_Ft = Range("Beta_OFT_ASIC_FT").Value
Beta_Oft_Asic_Wage = Range("Beta_OFT_ASIC_WAGE").Value
Beta_Percent_Actions = Range("Beta_Percent_actions").Value
Beta_Policy_Ft = Range("Beta_POLICY_FT").Value
Beta_Policy_Wage = Range("Beta_POLICY_WAGE").Value
Beta_Preparation_Costs = Range("Beta_Preparation_costs").Value
Beta_Rep_Mean = Range("Beta_Rep_mean").Value
Beta_Risk_Adj = Range("Beta_Risk_Adj").Value
Beta_Scale_Compliance_Firmsize = Range("Beta_scale_compliance_firmsize").Value
Beta_Template_Ft = Range("Beta_TEMPLATE_FT").Value
Beta_Template_Wage = Range("Beta_Template_wage").Value
Beta_Unfairc_Big = Range("Beta_UnfairC_big").Value
Beta_Unfairc_Noemp = Range("Beta_UnfairC_noemp").Value
Beta_Unfairc_Small = Range("Beta_UnfairC_small").Value
Beta_Wastage = Range("Beta_Wastage").Value
Beta_Yearly_Comp_Big = Range("Beta_Yearly_Comp_Big").Value
Beta_Yearly_Comp_Noemp = Range("Beta_yearly_comp_noemp").Value
Beta_Yearly_Comp_Small = Range("Beta_Yearly_comp_small").Value
Beta_Ongoing_Admin_Unfair = Range("Beta_Ongoing_Admin_Unfair").Value
Beta_Trans_share = Range("Beta_Trans_share").Value

‘_____’

alpha_Av_Fine = Beta_Av_Fine * (1# - Lower_Av_Fine) / (Upper_Av_Fine - 1#)
alpha_Max_TFP = Beta_Max_TFP * (1# - Lower_Max_TFP) / (Upper_Max_TFP - 1#)
alpha_Bias_Adjustment = Beta_Bias_Adjustment * (1# - Lower_Bias_Adjustment) / (Upper_Bias_Adjustment - 1#)
alpha_Business_Margin = Beta_Business_Margin * (1# - Lower_Business_Margin) / (Upper_Business_Margin - 1#)
alpha_Cost_Elasticity = Beta_Cost_Elasticity * (1# - Lower_Cost_Elasticity) / (Upper_Cost_Elasticity - 1#)
alpha_Cost_Per_Day = Beta_Cost_Per_Day * (1# - Lower_Cost_Per_Day) / (Upper_Cost_Per_Day - 1#)
alpha_Days_Case = Beta_Days_Case * (1# - Lower_Days_Case) / (Upper_Days_Case - 1#)
alpha_Deterrent = Beta_Deterrent * (1# - Lower_Deterrent) / (Upper_Deterrent - 1#)
alpha_Discount_Rate = Beta_Discount_Rate * (1# - Lower_Discount_Rate) / (Upper_Discount_Rate - 1#)
alpha_Fair_Premium = Beta_Fair_Premium * (1# - Lower_Fair_Premium) / (Upper_Fair_Premium - 1#)

alpha_False_Positive = Beta_False_Positive * (1# - Lower_False_Positive) / (Upper_False_Positive - 1#)

alpha_Foll_Mean = Beta_Foll_Mean * (1# - Lower_Foll_Mean) / (Upper_Foll_Mean - 1#)

alpha_Legal_Oneoff_Ft = Beta_Legal_Oneoff_Ft * (1# - Lower_Legal_Oneoff_Ft) / (Upper_Legal_Oneoff_Ft - 1#)

alpha_Legal_Oneoff_Wage = Beta_Legal_Oneoff_Wage * (1# - Lower_Legal_Oneoff_Wage) / (Upper_Legal_Oneoff_Wage - 1#)

alpha_Lowesttfp = Beta_Lowesttfp * (1# - Lower_Lowesttfp) / (Upper_Lowesttfp - 1#)

alpha_Base_TFP_Error = Beta_Base_TFP_Error * (1# - Lower_Base_TFP_Error) / (Upper_Base_TFP_Error - 1#)

alpha_Maxval_Risk = Beta_Maxval_Risk * (1# - Lower_Maxval_Risk) / (Upper_Maxval_Risk - 1#)

alpha_Meb = Beta_Meb * (1# - Lower_Meb) / (Upper_Meb - 1#)

alpha_Multiplier_By_State = Beta_Multiplier_By_State * (1# - Lower_Multiplier_By_State) / (Upper_Multiplier_By_State - 1#)

alpha_Nontransfer_Sh = Beta_Nontransfer_Sh * (1# - Lower_Nontransfer_Sh) / (Upper_Nontransfer_Sh - 1#)

alpha_Oft_Asic_Ft = Beta_Oft_Asic_Ft * (1# - Lower_Oft_Asic_Ft) / (Upper_Oft_Asic_Ft - 1#)

alpha_Oft_Asic_Wage = Beta_Oft_Asic_Wage * (1# - Lower_Oft_Asic_Wage) / (Upper_Oft_Asic_Wage - 1#)

alpha_Percent_Actions = Beta_Percent_Actions * (1# - Lower_Percent_Actions) / (Upper_Percent_Actions - 1#)

alpha_Policy_Ft = Beta_Policy_Ft * (1# - Lower_Policy_Ft) / (Upper_Policy_Ft - 1#)

alpha_Policy_Wage = Beta_Policy_Wage * (1# - Lower_Policy_Wage) / (Upper_Policy_Wage - 1#)

alpha_Preparation_Costs = Beta_Preparation_Costs * (1# - Lower_Preparation_Costs) / (Upper_Preparation_Costs - 1#)

alpha_Rep_Mean = Beta_Rep_Mean * (1# - Lower_Rep_Mean) / (Upper_Rep_Mean - 1#)

alpha_Risk_Adj = Beta_Risk_Adj * (1# - Lower_Risk_Adj) / (Upper_Risk_Adj - 1#)

alpha_Scale_Compliance_Firmsize = Beta_Scale_Compliance_Firmsize * (1# - Lower_Scale_Compliance_Firmsize) / (Upper_Scale_Compliance_Firmsize - 1#)

alpha_Template_Ft = Beta_Template_Ft * (1# - Lower_Template_Ft) / (Upper_Template_Ft - 1#)

alpha_Template_Wage = Beta_Template_Wage * (1# - Lower_Template_Wage) / (Upper_Template_Wage - 1#)

alpha_Unfairc_Big = Beta_Unfairc_Big * (1# - Lower_Unfairc_Big) / (Upper_Unfairc_Big - 1#)

alpha_Unfairc_Noemp = Beta_Unfairc_Noemp * (1# - Lower_Unfairc_Noemp) / (Upper_Unfairc_Noemp - 1#)

alpha_Unfairc_Small = Beta_Unfairc_Small * (1# - Lower_Unfairc_Small) / (Upper_Unfairc_Small - 1#)

alpha_Wastage = Beta_Wastage * (1# - Lower_Wastage) / (Upper_Wastage - 1#)

alpha_Yearly_Comp_Big = Beta_Yearly_Comp_Big * (1# - Lower_Yearly_Comp_Big) / (Upper_Yearly_Comp_Big - 1#)

alpha_Yearly_Comp_Noemp = Beta_Yearly_Comp_Noemp * (1# - Lower_Yearly_Comp_Noemp) / (Upper_Yearly_Comp_Noemp - 1#)

alpha_Yearly_Comp_Small = Beta_Yearly_Comp_Small * (1# - Lower_Yearly_Comp_Small) / (Upper_Yearly_Comp_Small - 1#)

alpha_Ongoing_Admin_Unfair = Beta_Ongoing_Admin_Unfair * (1# - Lower_Ongoing_Admin_Unfair) / (Upper_Ongoing_Admin_Unfair - 1#)

alpha_Trans_share = Beta_Trans_share * (1# - Lower_Trans_share) / (Upper_Trans_share - 1#)

'Start Basic calculations

Vic_detriment = SV_Rep_Mean + SV_Foll_Mean

Australian_detriment = Aust_pop16 / Vic_pop16 * Vic_detriment * SV_Bias_Adjustment ' takes account of any biases in the Victorian survey (under & over)

Experienced_Australian_detriment = Australian_detriment * (1# + SV_Fair_Premium / 100#) ' takes account of 'damages' (eg emotional distress, which is not counted in the CAV survey)

Risk_before = Experienced_Australian_detriment / CON2005_06 * 100# * SV_Risk_Adj 'Calculates the risk perceived by consumers BEFORE policy change

FollowUp_share = SV_Foll_Mean / Vic_detriment * 100#

Repair_share = 100# - FollowUp_share

Avoided_Aust_detriment = SV_Deterrent / 100# * Australian_detriment ' gross gain to consumers (what they get, but ignoring transfers)

Net_direct_gain = SV_Deterrent / 100# * Australian_detriment * (FollowUp_share / 100# + Repair_share / 100# * SV_Nontransfer_Sh / 100#) ' take account of transfers for welfare gain, but ignores indirect effects of fairness gain

```
Fairness_gain = Avoided_Aust_detriment * SV_Fair_Premium / 100# ' calculates the fairness gain
After_experienced = Avoided_Aust_detriment + Fairness_gain ' ie the gross measure of the gain to consumers not dealing with transfers - consumers' perceptions of risk do not take into account transfers
Direct_detriment = Net_direct_gain + Fairness_gain ' ie the real economy-wide benefits of reduced detriment, ignoring transfers but including emotional damages avoided
Risk_after = (1# - SV_Deterrent / 100#) * Risk_before ' Perceived consumer risk AFTER policy change (ie Before risk less gain = actual new risk)
base_risk = Risk_before
'*****************************************************************
'TFP and growth - determining the new productivity growth rate
'*****************************************************************

base_TFP = SV_Max_TFP + SV_Base_TFP_Error
If SV_Max_TFP > (SV_Lowesttfp - base_TFP * SV_Maxval_Risk / (Risk_before / 100#)) / (1# - SV_Maxval_Risk / (Risk_before / 100#)) Then
    P_alpha = (SV_Lowesttfp - base_TFP * SV_Maxval_Risk / (Risk_before / 100#)) / (1# - SV_Maxval_Risk / (Risk_before / 100#))
Else
    P_alpha = SV_Max_TFP
End If
p_gamma = (Log((P_alpha - SV_Lowesttfp) / (P_alpha - base_TFP))) / (Log(SV_Maxval_Risk) - Log((Risk_before / 100#)))
p_beta = (P_alpha - SV_Lowesttfp) / (SV_Maxval_Risk ^ p_gamma)

If Risk_after / 100# < SV_Maxval_Risk Then
    New_TFP = P_alpha - p_beta * (Risk_after / 100#) ^ p_gamma
Else
    New_TFP = SV_Lowesttfp
End If

'*********************************************************************
'2007-08 and 2008-09 values of household consumption and GDP in 2006-07 prices
'*********************************************************************
'Dimmension growth variables
Dim GDPCAP_before(), GDPCAP_after(), GDP_before(), GDP_after(), CONS_before(), CONS_after() As Double
ReDim GDPCAP_before(Yrs + 1), GDPCAP_after(Yrs + 1), GDP_before(Yrs + 1), GDP_after(Yrs + 1), CONS_before(Yrs + 1), CONS_after(Yrs + 1) As Double

GDPCAP_before(0) = 100# * ((1# + D_Labhours(0) / 100#) * (1# + base_TFP / 100#) - 1#) ' ie 2007-08 'growth in GDPCAP
GDPCAP_after(0) = 100# * ((1# + D_Labhours(0) / 100#) * (1# + base_TFP / 100#) - 1#) ' 2007-08 (same as before- value since intervention starts tyo have effect in 2008-09)
GDP_before(0) = Gdp_2006_07 * (GDPCAP_before(0) / 100# + 1#) * (D_Pop(0) / 100# + 1#) '2007-08
GDP_after(0) = Gdp_2006_07 * (GDPCAP_after(0) / 100# + 1#) * (D_Pop(0) / 100# + 1#) '2007-08
CONS_before(0) = Lr_Con_Sh * GDP_before(0) '2007-08
CONS_after(0) = Lr_Con_Sh * GDP_after(0)  '2007-08
GDPCAP_before(1) = 100 * ((1# + D_Labhours(1) / 100#) * (1# + base_TFP / 100#) - 1#) ' ie 2008-09
GDPCAP_after(1) = 100 * ((1# + D_Labhours(1) / 100#) * (1# + New_TFP / 100#) - 1#) ' 2008-09
GDP_before(1) = GDP_before(0) * (GDPCAP_before(1) / 100# + 1#) * (D_Pop(1) / 100# + 1#) '2008-09
GDP_after(1) = GDP_after(0) * (GDPCAP_after(1) / 100# + 1) * (D_Pop(1) / 100# + 1#) '2008-09
CONS_before(1) = Lr_Con_Sh * GDP_before(1) '2008-09
CONS_after(1) = Lr_Con_Sh * GDP_after(1)  '2008-09
```

```
'************************************
'Gains in 2008-09 in 2006-07 prices *
'************************************
'ReDimmension benefit and costs variables
ReDim YR_Net_direct_gain(Yrs + 1), Innovation_gain(Yrs + 1), Transact(Yrs + 1), Comp_Savings(Yrs + 1),
    Litigation_Policy_costs(Yrs + 1) As Double
ReDim Gross_compliance_savings(Yrs + 1) As Double
ReDim agency_costs(Yrs + 1) As Double

con2008_09 = CONS_after(1) ' household consumption 2008-09 (2006-07 prices)
YR_Net_direct_gain(1) = Direct_detriment / CON2005_06 * con2008_09 ' Realised consumer benefit from redcued
    detriment for 2008-09 (2006-07 prices)
Welfare_effect = Welfare_Weight * YR_Net_direct_gain(1) ' Welfare impact taking into account distributional weights
Innovation_gain(1) = (GDP_after(1) - GDP_before(1)) * LR_Total_Cons_Share
Gross_gain_2008_09 = 1# / SV_Risk_Adj * (Risk_before - Risk_after) / 100# * con2008_09 'Gross gain to consumers
    from reduced detriment (exluding transfers) in 2008-09 (2006-07 prices)

'[A] Transaction efficiency benefits 2008-09 gains in 2006-07 prices
Transaction_costs = SV_Trans_share / 100# * con2008_09 ' here using the expected value of trans_share
Transact(1) = -Transaction_costs * (((Risk_after / Risk_before) ^ SV_Cost_Elasticity) - 1#)
Transact_dividend_rate = Transact(1) / con2008_09 * 100#

'[B] Compliance savings by firm size
'[B1] determine (endogenous) parameters for function describing savings by enterprise employment size
emp_max = ((Alpha_Emp - 1#) * (1# - Shift_Factor) - (Beta_Emp - 1#) * Shift_Factor) / (Alpha_Emp + Beta_Emp - 2#)
    * High_Emp
xmax = ((Alpha_Emp - 1#) * (1# - Shift_Factor) - (Beta_Emp - 1#) * Shift_Factor) / (Alpha_Emp + Beta_Emp - 2#)
xsmall = 1# / High_Emp
scalar = (Max - Saving_Large) / (((xmax + Shift_Factor) ^ (Alpha_Emp - 1#)) * ((1# - Shift_Factor - xmax) ^ (Beta_Emp
    - 1#)))


'[B2] Ongoing compliance savings by business size

'dimension some fixed arrays
Dim Scale_savings(8) As Double
Dim Compliance_savings_by_Size(8, 7) As Double

Gross_compliance_savings(1) = 0#
For states = 1 To 8
    Scale_savings(states) = 1# + SV_Multiplier_By_State * Log(states) ' sets a scale factor that increases the compliance
    gain as a log function of the number of states a business operates in
    For empsize = 1 To 7
        Compliance_savings_by_Size(states, empsize) = Scale_savings(states) * ((AvEmp(empsize) * Saving_Large +
        scalar * ((AvEmp(empsize) / High_Emp + Shift_Factor) ^ (Alpha_Emp - 1#)) * (1# - (AvEmp(empsize) /
        High_Emp + Shift_Factor)) ^ (Beta_Emp - 1#)) ^ SV_Scale_Compliance_Firmsize)
        Gross_compliance_savings(1) = Gross_compliance_savings(1) + Compliance_savings_by_Size(states,
        empsize) * Enterprises(states, empsize) / 1000000#
    Next empsize
Next states
```

```
'[B3] Ongoing and one-off incremental agency and compliance burdens for business
agency_costs(1) = (No_Emp * SV_Yearly_Comp_Noemp + Emp_Sm100 * SV_Yearly_Comp_Small + Emp_Gt100 * SV_Yearly_Comp_Big) / 1000000#
OneOff_compliance_burdens = (No_Emp * SV_Unfairc_Noemp + Emp_Sm100 * SV_Unfairc_Small + Emp_Gt100 * SV_Unfairc_Big) / 1000000#
Annuity_oneoff = SV_Discount_Rate / (100# + SV_Discount_Rate) * OneOff_compliance_burdens
Comp_Savings(1) = Gross_compliance_savings(1) - agency_costs(1) ' ongoing net compliance savings
Comply_ratio = (Gross_compliance_savings(1) - agency_costs(1)) / con2008_09 * 100# 'ongoing net compliance savings as a share of consumption
Net_Ongoing_and_Oneoff_Comp_Savings = Gross_compliance_savings(1) - agency_costs(1) - OneOff_compliance_burdens ' Total net compliance savings

'[C] Ongoing litigation costs and admin commencing in 2007-08 (2006-07 prices)

'[C1] One-off admin costs
Costs_negotiate = SV_Template_Ft * 9# * SV_Template_Wage * (1# + SV_Meb / 100#) / 1000000# 'Costs of negotation (including meb)
Credit_move_costs = SV_Wastage / 100# * SV_Oft_Asic_Wage * SV_Oft_Asic_Ft / 1000000#  'Costs of credit move (a resource waste - not needing to be financed through new taxes - hence no meb)
One_off_legal_costs = SV_Legal_Oneoff_Ft * SV_Legal_Oneoff_Wage * (1# + SV_Meb / 100#) / 1000000# 'One-off legal & policy costs (legal advice re how credit should be integrated into the Corporations Act; wording of new unfair contracts law; new discosure requirements)
Oneoff_admin = One_off_legal_costs + Credit_move_costs + Costs_negotiate

'[C2] Ongoing litigation costs and admin commencing in 2007-08 (2006-07 prices)
New_Actions = Round(SV_Percent_Actions / 100# * Actions, 0#) 'Businesses subject to new court-based enforcement measures per year (civil penalties & including class actions)
Transfer_fromOS = Share_Foreign / 100# * Share_Success / 100# * SV_Av_Fine * New_Actions / 1000000#  'Fines collected from O/S firms (net gain) $m
False_cases = Round(SV_False_Positive / 100# * New_Actions, 0) 'False positive cases
Cost_false = False_cases * 100000# / 1000000# 'External cost of false cases = $1,000,000 per case (unfairness, excessivley risk averse behaviour by all firms) $m
Reg_costs = (SV_Preparation_Costs + SV_Days_Case * SV_Cost_Per_Day) * New_Actions * (1# + SV_Meb / 100#) / 1000000#  'Regulator's costs associated with new penalties (including court) and MEB financing $m
Bus_costs = (1# + SV_Business_Margin / 100#) * (SV_Preparation_Costs + SV_Days_Case * SV_Cost_Per_Day) * New_Actions / 1000000# 'Business' own costs $m 2007-08, 2006-007 prices
litigation_costs = (Bus_costs + Reg_costs + Cost_false - Transfer_fromOS)
Policy_costs = SV_Policy_Ft * SV_Policy_Wage / 1000000# 'Ongoing policy costs associated with new changes
Unfair_Admin_Costs = SV_Ongoing_Admin_Unfair * SV_Oft_Asic_Wage / 1000000#
Litigation_Policy_costs(1) = (litigation_costs + Policy_costs + Unfair_Admin_Costs) + Oneoff_admin 'All legal costs by firms and govt, all policy and admin costs, including re-structuring costs
admin_ratio = (Unfair_Admin_Costs + litigation_costs + Policy_costs) / con2008_09 * 100# 'Only ONGOING litigation & admin costs as a share of consumption

'****************************************
'PV and Anuity Gains in 2006-07 prices *
'****************************************
PV_Net_direct_Gain = YR_Net_direct_gain(1)
PV_Innovation_gain = Innovation_gain(1)
PV_Transact = Transact(1)
PV_Gross_Compliance_Savings = Gross_compliance_savings(1)
PV_Agency_Costs = agency_costs(1)
PV_Comp_Savings = Comp_Savings(1)
```

```
PV_Litigation_Policy_costs = Litigation_Policy_costs(1)

For Period = 2009 To (2007 + Yrs) ' IE "Yrs" IS THE NUMBER OF YEARS, STARTING WITH 2009, WHICH IS THE 2008-09 fiscal year
    T = Period - 2007 'This ensures that VARIABLE(1), which is the value for 2008-09 is not overwritten
    GDPCAP_before(T) = 100# * ((1# + D_Labhours(T) / 100#) * (1# + base_TFP / 100#) - 1#)
    GDPCAP_after(T) = 100# * ((1# + D_Labhours(T) / 100#) * (1# + New_TFP / 100#) - 1#)
    GDP_before(T) = GDP_before(T - 1) * (GDPCAP_before(T) / 100# + 1#) * (D_Pop(T) / 100# + 1#) 'GDP BEFORE policy in 2006-07 prices
    GDP_after(T) = GDP_after(T - 1) * (GDPCAP_after(T) / 100# + 1#) * (D_Pop(T) / 100# + 1#) 'GDP AFTER policy in 2006-07 prices
    CONS_before(T) = Lr_Con_Sh * GDP_before(T) 'household consumption BEFORE policy in 2006-07 prices
    CONS_after(T) = Lr_Con_Sh * GDP_after(T) 'household consumption AFTER policy in 2006-07 prices
    YR_Net_direct_gain(T) = (Direct_detriment / CON2005_06) * CONS_after(T) / ((1# + SV_Discount_Rate / 100#) ^ (T - 1))
    Innovation_gain(T) = LR_Total_Cons_Share * (GDP_after(T) - GDP_before(T)) / ((1# + SV_Discount_Rate / 100#) ^ (T - 1))
    Transact(T) = Transact_dividend_rate / 100# * CONS_after(T) / ((1# + SV_Discount_Rate / 100#) ^ (T - 1))
    Gross_compliance_savings(T) = (Gross_compliance_savings(1) / con2008_09) * CONS_after(T) / ((1# + SV_Discount_Rate / 100#) ^ (T - 1))
    agency_costs(T) = (agency_costs(1) / con2008_09) * CONS_after(T) / ((1# + SV_Discount_Rate / 100#) ^ (T - 1))
    Comp_Savings(T) = (Comply_ratio / 100# * CONS_after(T)) / ((1# + SV_Discount_Rate / 100#) ^ (T - 1))
    Litigation_Policy_costs(T) = (admin_ratio / 100# * CONS_after(T)) / ((1# + SV_Discount_Rate / 100#) ^ (T - 1))
    PV_Net_direct_Gain = PV_Net_direct_Gain + YR_Net_direct_gain(T)
    PV_Innovation_gain = PV_Innovation_gain + Innovation_gain(T)
    PV_Transact = PV_Transact + Transact(T)
    PV_Gross_Compliance_Savings = PV_Gross_Compliance_Savings + Gross_compliance_savings(T)
    PV_Agency_Costs = PV_Agency_Costs + agency_costs(T)
    PV_Comp_Savings = PV_Comp_Savings + Comp_Savings(T)
    PV_Litigation_Policy_costs = PV_Litigation_Policy_costs + Litigation_Policy_costs(T)
Next Period
PV_NET_ONGOING_Compliance_Savings = PV_Gross_Compliance_Savings - PV_Agency_Costs
PV_Net_Ongoing_and_Oneoff_Comp_Savings = PV_Gross_Compliance_Savings - PV_Agency_Costs - OneOff_compliance_burdens

'Derive annuity values
AN_Net_direct_Gain = PV_Net_direct_Gain * SV_Discount_Rate / 100#
AN_Innovation_gain = PV_Innovation_gain * SV_Discount_Rate / 100#
AN_Transact = PV_Transact * SV_Discount_Rate / 100#
AN_Net_Ongoing_and_Oneoff_Comp_Savings = PV_Net_Ongoing_and_Oneoff_Comp_Savings * SV_Discount_Rate / 100#
AN_Litigation_Policy_costs = PV_Litigation_Policy_costs * SV_Discount_Rate / 100#
AN_Gross_compliance_savings = PV_Gross_Compliance_Savings * SV_Discount_Rate / 100#
AN_agency_costs = PV_Agency_Costs * SV_Discount_Rate / 100#
AN_NET_ONGOING_Compliance_Savings = PV_NET_ONGOING_Compliance_Savings * SV_Discount_Rate / 100#
AN_OneOff_compliance_burdens = OneOff_compliance_burdens * SV_Discount_Rate / 100#

'*************************************************
'Summary values (2006-07 prices)  *
'*************************************************
```

```
'net gains
Net_Gain_2008_09 = YR_Net_direct_gain(1) + Transact(1) + Innovation_gain(1) +
    Net_Ongoing_and_Oneoff_Comp_Savings - Litigation_Policy_costs(1)
PV_Net_Gain = PV_Net_direct_Gain + PV_Transact + PV_Innovation_gain +
    PV_Net_Ongoing_and_Oneoff_Comp_Savings - PV_Litigation_Policy_costs
AN_Net_Gain = AN_Net_direct_Gain + AN_Transact + AN_Innovation_gain +
    AN_Net_Ongoing_and_Oneoff_Comp_Savings - AN_Litigation_Policy_costs

'net compliance gains (should match with above - but a good check)

Net_Compliance_Gains = Gross_compliance_savings(1) - agency_costs(1) - OneOff_compliance_burdens
PV_Net_Compliance_Gains = PV_Gross_Compliance_Savings - PV_Agency_Costs - OneOff_compliance_burdens
AN_Net_Compliance_Gains = AN_Gross_compliance_savings - AN_agency_costs - AN_OneOff_compliance_burdens



Application.ScreenUpdating = False

'Table 1: print results BASED ON MEANS
Worksheets("Summary").Cells(7, 2).Value = YR_Net_direct_gain(1)
Worksheets("Summary").Cells(8, 2).Value = Transact(1)
Worksheets("Summary").Cells(9, 2).Value = Innovation_gain(1)
Worksheets("Summary").Cells(10, 2).Value = Net_Ongoing_and_Oneoff_Comp_Savings
Worksheets("Summary").Cells(11, 2).Value = Litigation_Policy_costs(1)

Worksheets("Summary").Cells(7, 3).Value = PV_Net_direct_Gain
Worksheets("Summary").Cells(8, 3).Value = PV_Transact
Worksheets("Summary").Cells(9, 3).Value = PV_Innovation_gain
Worksheets("Summary").Cells(10, 3).Value = PV_Net_Ongoing_and_Oneoff_Comp_Savings
Worksheets("Summary").Cells(11, 3).Value = PV_Litigation_Policy_costs



Worksheets("Summary").Cells(7, 4).Value = AN_Net_direct_Gain
Worksheets("Summary").Cells(8, 4).Value = AN_Transact
Worksheets("Summary").Cells(9, 4).Value = AN_Innovation_gain
Worksheets("Summary").Cells(10, 4).Value = AN_Net_Ongoing_and_Oneoff_Comp_Savings
Worksheets("Summary").Cells(11, 4).Value = AN_Litigation_Policy_costs

Worksheets("Summary").Cells(12, 2).Value = Net_Gain_2008_09
Worksheets("Summary").Cells(12, 3).Value = PV_Net_Gain
Worksheets("Summary").Cells(12, 4).Value = AN_Net_Gain

'Table 2: Print results
Worksheets("Some detailed results").Cells(5, 2).Value = Vic_detriment
Worksheets("Some detailed results").Cells(6, 2).Value = Aust_pop16 / Vic_pop16 * Vic_detriment
Worksheets("Some detailed results").Cells(7, 2).Value = Aust_pop16 / Vic_pop16 * Vic_detriment *
    SV_Bias_Adjustment  ' takes account of possible bias in the Victorian survey (under & over) from exclusions and
    additions that are out of scope go the Commission's work
Worksheets("Some detailed results").Cells(8, 2).Value = Experienced_Australian_detriment ' takes account of 'damages'
    (eg emotional distress, which is not counted in the CAV survey
```

```
Worksheets("Some detailed results").Cells(9, 2).Value = Experienced_Australian_detriment / CON2005_06 * 100#
Worksheets("Some detailed results").Cells(10, 2).Value = Risk_before
Worksheets("Some detailed results").Cells(11, 2).Value = CON2005_06

Worksheets("Some detailed results").Cells(14, 2).Value = SV_Risk_Adj * (Experienced_Australian_detriment -
    After_experienced) / CON2005_06 * 100#
Worksheets("Some detailed results").Cells(16, 2).Value = Avoided_Aust_detriment
Worksheets("Some detailed results").Cells(17, 2).Value = Net_direct_gain
Worksheets("Some detailed results").Cells(18, 2).Value = After_experienced
Worksheets("Some detailed results").Cells(19, 2).Value = Direct_detriment
Worksheets("Some detailed results").Cells(20, 2).Value = con2008_09
Worksheets("Some detailed results").Cells(21, 2).Value = After_experienced / CON2005_06 * con2008_09
Worksheets("Some detailed results").Cells(22, 2).Value = Direct_detriment / CON2005_06 * con2008_09

Worksheets("Summary").Cells(28, 2).Value = Risk_after
Worksheets("Summary").Cells(29, 2).Value = Gross_gain_2008_09
Worksheets("Summary").Cells(30, 2).Value = YR_Net_direct_gain(1)
Worksheets("Summary").Cells(31, 2).Value = PV_Net_direct_Gain
Worksheets("Summary").Cells(32, 2).Value = AN_Net_direct_Gain
Worksheets("Summary").Cells(33, 2).Value = Welfare_effect

Worksheets("Summary").Cells(36, 2).Value = Transaction_costs
Worksheets("Summary").Cells(37, 2).Value = Transact(1)
Worksheets("Summary").Cells(38, 2).Value = PV_Transact
Worksheets("Summary").Cells(39, 2).Value = AN_Transact

Worksheets("Summary").Cells(42, 2).Value = New_TFP
Worksheets("Summary").Cells(43, 2).Value = Innovation_gain(1)
Worksheets("Summary").Cells(44, 2).Value = PV_Innovation_gain
Worksheets("Summary").Cells(45, 2).Value = AN_Innovation_gain

Worksheets("Summary").Cells(52, 2).Value = Gross_compliance_savings(1)
Worksheets("Summary").Cells(53, 2).Value = agency_costs(1)
Worksheets("Summary").Cells(54, 2).Value = Comp_Savings(1)
Worksheets("Summary").Cells(55, 2).Value = OneOff_compliance_burdens

Worksheets("Summary").Cells(52, 3).Value = PV_Gross_Compliance_Savings
Worksheets("Summary").Cells(53, 3).Value = PV_Agency_Costs
Worksheets("Summary").Cells(54, 3).Value = PV_NET_ONGOING_Compliance_Savings
Worksheets("Summary").Cells(55, 3).Value = OneOff_compliance_burdens

Worksheets("Summary").Cells(52, 4).Value = AN_Gross_compliance_savings
Worksheets("Summary").Cells(53, 4).Value = AN_agency_costs
Worksheets("Summary").Cells(54, 4).Value = AN_NET_ONGOING_Compliance_Savings
Worksheets("Summary").Cells(55, 4).Value = AN_OneOff_compliance_burdens

Worksheets("Summary").Cells(56, 2).Value = Net_Compliance_Gains
Worksheets("Summary").Cells(56, 3).Value = PV_Net_Compliance_Gains
Worksheets("Summary").Cells(56, 4).Value = AN_Net_Compliance_Gains
```

```
Worksheets("Summary").Cells(61, 2).Value = litigation_costs
Worksheets("Summary").Cells(62, 2).Value = Policy_costs
Worksheets("Summary").Cells(63, 2).Value = Oneoff_admin
Worksheets("Summary").Cells(64, 2).Value = Litigation_Policy_costs(1)
Worksheets("Summary").Cells(65, 2).Value = PV_Litigation_Policy_costs
Worksheets("Summary").Cells(66, 2).Value = AN_Litigation_Policy_costs

End Sub

Private Sub sim()
' SIMULATION PART
'displays alphas and betas
Dim BetaVarName(40) As String
Dim Betaf(40, 2) As Double

Betaf(1, 1) = alpha_Av_Fine
Betaf(2, 1) = alpha_Max_TFP
Betaf(3, 1) = alpha_Bias_Adjustment
Betaf(4, 1) = alpha_Business_Margin
Betaf(5, 1) = alpha_Cost_Elasticity
Betaf(6, 1) = alpha_Cost_Per_Day
Betaf(7, 1) = alpha_Days_Case
Betaf(8, 1) = alpha_Deterrent
Betaf(9, 1) = alpha_Discount_Rate
Betaf(10, 1) = alpha_Fair_Premium
Betaf(11, 1) = alpha_False_Positive
Betaf(12, 1) = alpha_Foll_Mean
Betaf(13, 1) = alpha_Legal_Oneoff_Ft
Betaf(14, 1) = alpha_Legal_Oneoff_Wage
Betaf(15, 1) = alpha_Lowesttfp
Betaf(16, 1) = alpha_Base_TFP_Error
Betaf(17, 1) = alpha_Maxval_Risk
Betaf(18, 1) = alpha_Meb
Betaf(19, 1) = alpha_Multiplier_By_State
Betaf(20, 1) = alpha_Nontransfer_Sh
Betaf(21, 1) = alpha_Oft_Asic_Ft
Betaf(22, 1) = alpha_Oft_Asic_Wage
Betaf(23, 1) = alpha_Percent_Actions
Betaf(24, 1) = alpha_Policy_Ft
Betaf(25, 1) = alpha_Policy_Wage
Betaf(26, 1) = alpha_Preparation_Costs
Betaf(27, 1) = alpha_Rep_Mean
Betaf(28, 1) = alpha_Risk_Adj
Betaf(29, 1) = alpha_Scale_Compliance_Firmsize
Betaf(30, 1) = alpha_Template_Ft
Betaf(31, 1) = alpha_Template_Wage
Betaf(32, 1) = alpha_Unfairc_Big
Betaf(33, 1) = alpha_Unfairc_Noemp
Betaf(34, 1) = alpha_Unfairc_Small
Betaf(35, 1) = alpha_Wastage
```

Betaf(36, 1) = alpha_Yearly_Comp_Big
Betaf(37, 1) = alpha_Yearly_Comp_Noemp
Betaf(38, 1) = alpha_Yearly_Comp_Small
Betaf(39, 1) = alpha_Ongoing_Admin_Unfair
Betaf(40, 1) = alpha_Trans_share
Betaf(1, 2) = Beta_Av_Fine
Betaf(2, 2) = Beta_Max_TFP
Betaf(3, 2) = Beta_Bias_Adjustment
Betaf(4, 2) = Beta_Business_Margin
Betaf(5, 2) = Beta_Cost_Elasticity
Betaf(6, 2) = Beta_Cost_Per_Day
Betaf(7, 2) = Beta_Days_Case
Betaf(8, 2) = Beta_Deterrent
Betaf(9, 2) = Beta_Discount_Rate
Betaf(10, 2) = Beta_Fair_Premium
Betaf(11, 2) = Beta_False_Positive
Betaf(12, 2) = Beta_Foll_Mean
Betaf(13, 2) = Beta_Legal_Oneoff_Ft
Betaf(14, 2) = Beta_Legal_Oneoff_Wage
Betaf(15, 2) = Beta_Lowesttfp
Betaf(16, 2) = Beta_Base_TFP_Error
Betaf(17, 2) = Beta_Maxval_Risk
Betaf(18, 2) = Beta_Meb
Betaf(19, 2) = Beta_Multiplier_By_State
Betaf(20, 2) = Beta_Nontransfer_Sh
Betaf(21, 2) = Beta_Oft_Asic_Ft
Betaf(22, 2) = Beta_Oft_Asic_Wage
Betaf(23, 2) = Beta_Percent_Actions
Betaf(24, 2) = Beta_Policy_Ft
Betaf(25, 2) = Beta_Policy_Wage
Betaf(26, 2) = Beta_Preparation_Costs
Betaf(27, 2) = Beta_Rep_Mean
Betaf(28, 2) = Beta_Risk_Adj
Betaf(29, 2) = Beta_Scale_Compliance_Firmsize
Betaf(30, 2) = Beta_Template_Ft
Betaf(31, 2) = Beta_Template_Wage
Betaf(32, 2) = Beta_Unfairc_Big
Betaf(33, 2) = Beta_Unfairc_Noemp
Betaf(34, 2) = Beta_Unfairc_Small
Betaf(35, 2) = Beta_Wastage
Betaf(36, 2) = Beta_Yearly_Comp_Big
Betaf(37, 2) = Beta_Yearly_Comp_Noemp
Betaf(38, 2) = Beta_Yearly_Comp_Small
Betaf(39, 2) = Beta_Ongoing_Admin_Unfair
Betaf(40, 2) = Beta_Trans_share

BetaVarName(1) = "Av_Fine"
BetaVarName(2) = "Max_TFP"
BetaVarName(3) = "Bias_Adjustment"
BetaVarName(4) = "Business_Margin"

```
BetaVarName(5) = "Cost_Elasticity"
BetaVarName(6) = "Cost_Per_Day"
BetaVarName(7) = "Days_Case"
BetaVarName(8) = "Deterrent"
BetaVarName(9) = "Discount_Rate"
BetaVarName(10) = "Fair_Premium"
BetaVarName(11) = "False_Positive"
BetaVarName(12) = "Foll_Mean"
BetaVarName(13) = "Legal_Oneoff_Ft"
BetaVarName(14) = "Legal_Oneoff_Wage"
BetaVarName(15) = "Lowesttfp"
BetaVarName(16) = "Base_TFP_Error"
BetaVarName(17) = "Maxval_Risk"
BetaVarName(18) = "Meb"
BetaVarName(19) = "Multiplier_By_State"
BetaVarName(20) = "Nontransfer_Sh"
BetaVarName(21) = "Oft_Asic_Ft"
BetaVarName(22) = "Oft_Asic_Wage"
BetaVarName(23) = "Percent_Actions"
BetaVarName(24) = "Policy_Ft"
BetaVarName(25) = "Policy_Wage"
BetaVarName(26) = "Preparation_Costs"
BetaVarName(27) = "Rep_Mean"
BetaVarName(28) = "Risk_Adj"
BetaVarName(29) = "Scale_Compliance_Firmsize"
BetaVarName(30) = "Template_Ft"
BetaVarName(31) = "Template_Wage"
BetaVarName(32) = "Unfairc_Big"
BetaVarName(33) = "Unfairc_Noemp"
BetaVarName(34) = "Unfairc_Small"
BetaVarName(35) = "Wastage"
BetaVarName(36) = "Yearly_Comp_Big"
BetaVarName(37) = "Yearly_Comp_Noemp"
BetaVarName(38) = "Yearly_Comp_Small"
BetaVarName(39) = "Ongoing_Admin_Unfair"
BetaVarName(40) = "Trans_share"

For i = 1 To 40
    For j = 1 To 2
      Worksheets("Some detailed results").Cells(26 + i, j + 1).Value = Betaf(i, j)
      If Betaf(i, j) < 0 Then
          MsgBox ("The variable " & BetaVarName(i) & " gives a negative alpha or beta. Amend beta value or range!")
          End
      End If
    Next j
Next i

Dim time As Double

iters = Val(TxtSim)
```

```
If iters = False Then
    CmdCancel = True
ElseIf iters > 10000 Then
    MsgBox ("Really! Despite prompting, you have asked for more than 10000 iterations")
    End
ElseIf iters < 5 Then
    MsgBox ("Oops! You have asked for less than 5 iterations")
    End
Else
units = " seconds"
time = 3 + Int(iters * 0.025)
time_value = Application.WorksheetFunction.Text(time, "0")

If time > 60 Then
    time = Round(time / 60, 1)
    units = " minutes"
    time_value = Application.WorksheetFunction.Text(time, "0.0")
End If

Answer = MsgBox("It will take about " & time_value & units & " to complete simulations. Press OK to run program, or
    cancel to choose a different number of simulations", _
vbOKCancel + vbExclamation, "Confirm program run")
If Answer = vbCancel Then Exit Sub ' the sub will end if the user selects the CANCEL-button

End If

Dim SIM_AN_Net_direct_Gain(), SIM_AN_NET_Gain(), SIM_AN_Transact(), SIM_AN_Innovation_gain(),
    SIM_AN_Comp_Savings() As Double
Dim SIM_AN_LITPCY_Costs() As Double
ReDim SIM_AN_Net_direct_Gain(iters - 1), SIM_AN_NET_Gain(iters - 1), SIM_AN_Transact(iters - 1),
    SIM_AN_Innovation_gain(iters - 1), SIM_Net_CMPLC_gains(iters - 1) As Double
ReDim SIM_AN_LITPCY_Costs(iters - 1) As Double
ReDim D_Labhours(Yrs + 1) As Double ' ie yrs rows (since row zero is defined as the first)
ReDim D_Pop(Yrs + 1) As Double

For Period = 2008 To (2008 + Yrs) ' IE "yrs" IS THE NUMBER OF YEARS, STARTING WITH 2008, WHICH IS THE
    2007-08 fiscal year
    i = Period - 2008 'IE RESET THE COUNTER TO ZERO FOR THE FIRST ELEMENT OF THE VECTOR
    D_Labhours(i) = Worksheets("Fixed Variables").Cells(i + 48, 2).Value
    D_Pop(i) = Worksheets("Fixed Variables").Cells(i + 48, 3).Value
Next Period

For i = 1 To iters
    'SD_ denotes a draw for a stochastic variable
    'Draw variables from a beta distribution
    Randomize ' seeds random generator with each simulation
    'Now take random draws from a beta distribution for each variable (SD = stochastic draw)
    SD_Av_Fine = SV_Av_Fine * (Application.BetaInv(Rnd, alpha_Av_Fine, Beta_Av_Fine, Lower_Av_Fine,
    Upper_Av_Fine))
    SD_Max_TFP = SV_Max_TFP * (Application.BetaInv(Rnd, alpha_Max_TFP, Beta_Max_TFP, Lower_Max_TFP,
    Upper_Max_TFP))
```

SD_Bias_Adjustment = SV_Bias_Adjustment * (Application.BetaInv(Rnd, alpha_Bias_Adjustment, Beta_Bias_Adjustment, Lower_Bias_Adjustment, Upper_Bias_Adjustment))

SD_Business_Margin = SV_Business_Margin * (Application.BetaInv(Rnd, alpha_Business_Margin, Beta_Business_Margin, Lower_Business_Margin, Upper_Business_Margin))

SD_Cost_Elasticity = SV_Cost_Elasticity * (Application.BetaInv(Rnd, alpha_Cost_Elasticity, Beta_Cost_Elasticity, Lower_Cost_Elasticity, Upper_Cost_Elasticity))

SD_Cost_Per_Day = SV_Cost_Per_Day * (Application.BetaInv(Rnd, alpha_Cost_Per_Day, Beta_Cost_Per_Day, Lower_Cost_Per_Day, Upper_Cost_Per_Day))

SD_Days_Case = SV_Days_Case * (Application.BetaInv(Rnd, alpha_Days_Case, Beta_Days_Case, Lower_Days_Case, Upper_Days_Case))

SD_Deterrent = SV_Deterrent * (Application.BetaInv(Rnd, alpha_Deterrent, Beta_Deterrent, Lower_Deterrent, Upper_Deterrent))

SD_Discount_Rate = SV_Discount_Rate * (Application.BetaInv(Rnd, alpha_Discount_Rate, Beta_Discount_Rate, Lower_Discount_Rate, Upper_Discount_Rate))

SD_Fair_Premium = SV_Fair_Premium * (Application.BetaInv(Rnd, alpha_Fair_Premium, Beta_Fair_Premium, Lower_Fair_Premium, Upper_Fair_Premium))

SD_False_Positive = SV_False_Positive * (Application.BetaInv(Rnd, alpha_False_Positive, Beta_False_Positive, Lower_False_Positive, Upper_False_Positive))

SD_Foll_Mean = SV_Foll_Mean * (Application.BetaInv(Rnd, alpha_Foll_Mean, Beta_Foll_Mean, Lower_Foll_Mean, Upper_Foll_Mean))

SD_Legal_Oneoff_Ft = SV_Legal_Oneoff_Ft * (Application.BetaInv(Rnd, alpha_Legal_Oneoff_Ft, Beta_Legal_Oneoff_Ft, Lower_Legal_Oneoff_Ft, Upper_Legal_Oneoff_Ft))

SD_Legal_Oneoff_Wage = SV_Legal_Oneoff_Wage * (Application.BetaInv(Rnd, alpha_Legal_Oneoff_Wage, Beta_Legal_Oneoff_Wage, Lower_Legal_Oneoff_Wage, Upper_Legal_Oneoff_Wage))

SD_Lowesttfp = SV_Lowesttfp * (Application.BetaInv(Rnd, alpha_Lowesttfp, Beta_Lowesttfp, Lower_Lowesttfp, Upper_Lowesttfp))

SD_Base_TFP_Error = (Application.BetaInv(Rnd, alpha_Base_TFP_Error, Beta_Base_TFP_Error, SV_Base_TFP_Error * Lower_Base_TFP_Error, SV_Base_TFP_Error * Upper_Base_TFP_Error))

SD_Maxval_Risk = SV_Maxval_Risk * (Application.BetaInv(Rnd, alpha_Maxval_Risk, Beta_Maxval_Risk, Lower_Maxval_Risk, Upper_Maxval_Risk))

SD_Meb = SV_Meb * (Application.BetaInv(Rnd, alpha_Meb, Beta_Meb, Lower_Meb, Upper_Meb))

SD_Multiplier_By_State = SV_Multiplier_By_State * (Application.BetaInv(Rnd, alpha_Multiplier_By_State, Beta_Multiplier_By_State, Lower_Multiplier_By_State, Upper_Multiplier_By_State))

SD_Nontransfer_Sh = SV_Nontransfer_Sh * (Application.BetaInv(Rnd, alpha_Nontransfer_Sh, Beta_Nontransfer_Sh, Lower_Nontransfer_Sh, Upper_Nontransfer_Sh))

SD_Oft_Asic_Ft = SV_Oft_Asic_Ft * (Application.BetaInv(Rnd, alpha_Oft_Asic_Ft, Beta_Oft_Asic_Ft, Lower_Oft_Asic_Ft, Upper_Oft_Asic_Ft))

SD_Oft_Asic_Wage = SV_Oft_Asic_Wage * (Application.BetaInv(Rnd, alpha_Oft_Asic_Wage, Beta_Oft_Asic_Wage, Lower_Oft_Asic_Wage, Upper_Oft_Asic_Wage))

SD_Percent_Actions = SV_Percent_Actions * (Application.BetaInv(Rnd, alpha_Percent_Actions, Beta_Percent_Actions, Lower_Percent_Actions, Upper_Percent_Actions))

SD_Policy_Ft = SV_Policy_Ft * (Application.BetaInv(Rnd, alpha_Policy_Ft, Beta_Policy_Ft, Lower_Policy_Ft, Upper_Policy_Ft))

SD_Policy_Wage = SV_Policy_Wage * (Application.BetaInv(Rnd, alpha_Policy_Wage, Beta_Policy_Wage, Lower_Policy_Wage, Upper_Policy_Wage))

SD_Preparation_Costs = SV_Preparation_Costs * (Application.BetaInv(Rnd, alpha_Preparation_Costs, Beta_Preparation_Costs, Lower_Preparation_Costs, Upper_Preparation_Costs))

SD_Rep_Mean = SV_Rep_Mean * (Application.BetaInv(Rnd, alpha_Rep_Mean, Beta_Rep_Mean, Lower_Rep_Mean, Upper_Rep_Mean))

SD_Risk_Adj = SV_Risk_Adj * (Application.BetaInv(Rnd, alpha_Risk_Adj, Beta_Risk_Adj, Lower_Risk_Adj, Upper_Risk_Adj))

SD_Scale_Compliance_Firmsize = SV_Scale_Compliance_Firmsize * (Application.BetaInv(Rnd, alpha_Scale_Compliance_Firmsize, Beta_Scale_Compliance_Firmsize, Lower_Scale_Compliance_Firmsize, Upper_Scale_Compliance_Firmsize))

SD_Template_Ft = SV_Template_Ft * (Application.BetaInv(Rnd, alpha_Template_Ft, Beta_Template_Ft, Lower_Template_Ft, Upper_Template_Ft))

```
SD_Template_Wage = SV_Template_Wage * (Application.BetaInv(Rnd, alpha_Template_Wage,
Beta_Template_Wage, Lower_Template_Wage, Upper_Template_Wage))
SD_Unfairc_Big = SV_Unfairc_Big * (Application.BetaInv(Rnd, alpha_Unfairc_Big, Beta_Unfairc_Big,
Lower_Unfairc_Big, Upper_Unfairc_Big))
SD_Unfairc_Noemp = SV_Unfairc_Noemp * (Application.BetaInv(Rnd, alpha_Unfairc_Noemp,
Beta_Unfairc_Noemp, Lower_Unfairc_Noemp, Upper_Unfairc_Noemp))
SD_Unfairc_Small = SV_Unfairc_Small * (Application.BetaInv(Rnd, alpha_Unfairc_Small, Beta_Unfairc_Small,
Lower_Unfairc_Small, Upper_Unfairc_Small))
SD_Wastage = SV_Wastage * (Application.BetaInv(Rnd, alpha_Wastage, Beta_Wastage, Lower_Wastage,
Upper_Wastage))
SD_Yearly_Comp_Big = SV_Yearly_Comp_Big * (Application.BetaInv(Rnd, alpha_Yearly_Comp_Big,
Beta_Yearly_Comp_Big, Lower_Yearly_Comp_Big, Upper_Yearly_Comp_Big))
SD_Yearly_Comp_Noemp = SV_Yearly_Comp_Noemp * (Application.BetaInv(Rnd, alpha_Yearly_Comp_Noemp,
Beta_Yearly_Comp_Noemp, Lower_Yearly_Comp_Noemp, Upper_Yearly_Comp_Noemp))
SD_Yearly_Comp_Small = SV_Yearly_Comp_Small * (Application.BetaInv(Rnd, alpha_Yearly_Comp_Small,
Beta_Yearly_Comp_Small, Lower_Yearly_Comp_Small, Upper_Yearly_Comp_Small))
SD_Ongoing_Admin_Unfair = SV_Ongoing_Admin_Unfair * (Application.BetaInv(Rnd,
alpha_Ongoing_Admin_Unfair, Beta_Ongoing_Admin_Unfair, Lower_Ongoing_Admin_Unfair,
Upper_Ongoing_Admin_Unfair))
SD_Trans_share = SV_Trans_share * (Application.BetaInv(Rnd, alpha_Trans_share, Beta_Trans_share,
Lower_Trans_share, Upper_Trans_share))


'Start Basic calculations

Vic_detriment = SD_Rep_Mean + SD_Foll_Mean
Australian_detriment = Aust_pop16 / Vic_pop16 * Vic_detriment * SD_Bias_Adjustment ' takes account of any
biases in the Victorian survey (under & over)
Experienced_Australian_detriment = Australian_detriment * (1# + SD_Fair_Premium / 100#) ' takes account of
'damages' (eg emotional distress, which is not counted in the CAV survey)
Risk_before = SD_Risk_Adj * Experienced_Australian_detriment / CON2005_06 * 100# 'Calculates the risk
perceived by consumers BEFORE policy change
FollowUp_share = SD_Foll_Mean / Vic_detriment * 100#
Repair_share = 100# - FollowUp_share
Avoided_Aust_detriment = SD_Deterrent / 100# * Australian_detriment ' gross gain to consumers (what they
perceive, but ignoring transfers)
Net_direct_gain = SD_Deterrent / 100# * Australian_detriment * (FollowUp_share / 100# + Repair_share / 100# *
SD_Nontransfer_Sh / 100#) ' take account of transfers for welfare gain, but ignores indirect effects of fairness gain
Fairness_gain = Avoided_Aust_detriment * SD_Fair_Premium / 100# ' calculates the fairness gain
After_experienced = Avoided_Aust_detriment + Fairness_gain ' ie the gross measure not dealing with transfers -
consumers' perceptions of risk do not take into account transfers
Direct_detriment = Net_direct_gain + Fairness_gain ' ie the real economy-wide benefits of reduced detriment,
ignoring transfers but including emotional damages avoided
Risk_after = (1 - SD_Deterrent / 100#) * Risk_before ' Perceived consumer risk AFTER policy change (ie Before risk
less gain = actual new risk)

'*****************************************************************
'TFP and growth - determining the new productivity growth rate
'*****************************************************************

base_TFP = SD_Max_TFP + SD_Base_TFP_Error
If SD_Max_TFP > (SD_Lowesttfp - base_TFP * SD_Maxval_Risk / (Risk_before / 100#)) / (1# - SD_Maxval_Risk /
(Risk_before / 100#)) Then
```

```
        P_alpha = (SD_Lowesttfp - base_TFP * SD_Maxval_Risk / (Risk_before / 100#)) / (1# - SD_Maxval_Risk /
        (Risk_before / 100#))
    Else
        P_alpha = SD_Max_TFP
    End If
    p_gamma = (Log((P_alpha - SD_Lowesttfp) / (P_alpha - base_TFP))) / (Log(SD_Maxval_Risk) - Log((Risk_before /
    100#)))
    p_beta = (P_alpha - SD_Lowesttfp) / (SD_Maxval_Risk ^ p_gamma)

    If Risk_after / 100# < SD_Maxval_Risk Then
        New_TFP = P_alpha - p_beta * (Risk_after / 100#) ^ p_gamma
    Else
        New_TFP = SD_Lowesttfp
    End If

'*********************************************************************
'2007-08 values of household consumption and GDP in 2006-07 prices
'*********************************************************************
'Dimmension growth variables
    ReDim GDPCAP_before(Yrs + 1), GDPCAP_after(Yrs + 1), GDP_before(Yrs + 1), GDP_after(Yrs + 1),
    CONS_before(Yrs + 1), CONS_after(Yrs + 1) As Double

    GDPCAP_before(0) = 100# * ((1# + D_Labhours(0) / 100#) * (1# + base_TFP / 100#) - 1#) ' ie 2007-08
    GDPCAP_after(0) = 100# * ((1# + D_Labhours(0) / 100#) * (1# + base_TFP / 100#) - 1#) ' 2007-08
    GDP_before(0) = Gdp_2006_07 * (GDPCAP_before(0) / 100# + 1#) * (D_Pop(0) / 100# + 1#)
    GDP_after(0) = Gdp_2006_07 * (GDPCAP_after(0) / 100# + 1#) * (D_Pop(0) / 100# + 1#)
    CONS_before(0) = Lr_Con_Sh * GDP_before(0)
    CONS_after(0) = Lr_Con_Sh * GDP_after(0)

    GDPCAP_before(1) = 100# * ((1# + D_Labhours(1) / 100#) * (1# + base_TFP / 100#) - 1#) ' ie 2008-09
    GDPCAP_after(1) = 100# * ((1# + D_Labhours(1) / 100#) * (1# + New_TFP / 100#) - 1#) ' 2008-09
    GDP_before(1) = GDP_before(0) * (GDPCAP_before(1) / 100# + 1#) * (D_Pop(1) / 100# + 1#)
    GDP_after(1) = GDP_after(0) * (GDPCAP_after(1) / 100# + 1#) * (D_Pop(1) / 100# + 1#)
    CONS_before(1) = Lr_Con_Sh * GDP_before(1)
    CONS_after(1) = Lr_Con_Sh * GDP_after(1)

'************************************
'Gains in 2007-08 in 2006-07 prices *
'************************************
'Dimmension benefit and costs variables
    ReDim YR_Net_direct_gain(Yrs + 1), Innovation_gain(Yrs + 1), Transact(Yrs + 1), Comp_Savings(Yrs + 1),
    Litigation_Policy_costs(Yrs + 1) As Double
    ReDim Gross_compliance_savings(Yrs + 1) As Double
    ReDim agency_costs(Yrs + 1) As Double

    con2008_09 = CONS_after(1) ' household consumption 2008-09 after policy  (2006-07 prices)
    YR_Net_direct_gain(1) = Direct_detriment / CON2005_06 * con2008_09 ' Detriment benefit for 2007-08 (2006-07
    prices)
    Welfare_effect = Welfare_Weight * YR_Net_direct_gain(1) ' Welfare impact taking into account distributional
    weights
    Innovation_gain(1) = LR_Total_Cons_Share * (GDP_after(1) - GDP_before(1))
```

```
Gross_gain_2008_09 = 1# / SD_Risk_Adj * (Risk_before - Risk_after) / 100# * con2008_09 'Gross gain to consumers from reduced detriment (exluding transfers) in 2007-08 (2006-07 prices)

'[A] Transaction efficiency benefits 2008-09 gains in 2006-07 prices
Dim u_error As Double
u_error = SD_Trans_share - SV_Trans_share ' this is the random error from the expected value of trans_share
a_value = Log(SV_Trans_share / 100#) - SD_Cost_Elasticity * Log(base_risk / 100#)   'This gives an alpha value that means that the expected value at base risk is the base trans_share (the base is the non-simulation case)
Trans_share = Exp(a_value + SD_Cost_Elasticity * Log(Risk_before / 100#)) * (1# + u_error) ' This is the before policy change value of the trans_share in the simulation
Transaction_costs = Trans_share * con2008_09
Transact(1) = con2008_09 * Trans_share * (1# - (Risk_after / Risk_before) ^ SD_Cost_Elasticity)
Transact_dividend_rate = Transact(1) / con2008_09 * 100#


'[B] Compliance savings by firm size
'[B1] determine (endogenous) parameters for function describing savings by enterprise employment size
emp_max = ((Alpha_Emp - 1#) * (1# - Shift_Factor) - (Beta_Emp - 1#) * Shift_Factor) / (Alpha_Emp + Beta_Emp - 2#) * High_Emp
xmax = ((Alpha_Emp - 1#) * (1# - Shift_Factor) - (Beta_Emp - 1#) * Shift_Factor) / (Alpha_Emp + Beta_Emp - 2#)
xsmall = 1# / High_Emp
scalar = (Max - Saving_Large) / (((xmax + Shift_Factor) ^ (Alpha_Emp - 1#)) * ((1# - Shift_Factor - xmax) ^ (Beta_Emp - 1#)))


'[B2] Ongoing compliance savings by business size 2008-09

'dimension some fixed arrays
Dim Scale_savings(8) As Double
Dim Compliance_savings_by_Size(8, 7) As Double

Gross_compliance_savings(1) = 0#
For states = 1 To 8
    Scale_savings(states) = 1# + SD_Multiplier_By_State * Log(states) ' sets a scale factor that increases the compliance gain as a log function of the number of states a business operates in
    For empsize = 1 To 7
        Compliance_savings_by_Size(states, empsize) = Scale_savings(states) * ((AvEmp(empsize) * Saving_Large + scalar * ((AvEmp(empsize) / High_Emp + Shift_Factor) ^ (Alpha_Emp - 1#)) * (1# - (AvEmp(empsize) / High_Emp + Shift_Factor)) ^ (Beta_Emp - 1#)) ^ SD_Scale_Compliance_Firmsize)
        Gross_compliance_savings(1) = Gross_compliance_savings(1) + Compliance_savings_by_Size(states, empsize) * Enterprises(states, empsize) / 1000000#
    Next empsize
Next states

'[B3] Ongoing and one-off incremental agency and compliance burdens for business
agency_costs(1) = (No_Emp * SD_Yearly_Comp_Noemp + Emp_Sm100 * SD_Yearly_Comp_Small + Emp_Gt100 * SD_Yearly_Comp_Big) / 1000000#
OneOff_compliance_burdens = (No_Emp * SD_Unfairc_Noemp + Emp_Sm100 * SD_Unfairc_Small + Emp_Gt100 * SD_Unfairc_Big) / 1000000#
Annuity_oneoff = SD_Discount_Rate / (100# + SD_Discount_Rate) * OneOff_compliance_burdens
Comp_Savings(1) = Gross_compliance_savings(1) - agency_costs(1) ' ongoing net compliance savings
```

```
Comply_ratio = (Gross_compliance_savings(1) - agency_costs(1)) / con2008_09 * 100# 'ongoing net compliance savings as a share of consumption
Net_Ongoing_and_Oneoff_Comp_Savings = Gross_compliance_savings(1) - agency_costs(1) - OneOff_compliance_burdens ' Total net compliance savings


'[C] Ongoing litigation costs and admin commencing in 2008-09 (2006-07 prices)

'[C1] One-off admin costs
Costs_negotiate = SD_Template_Ft * 9# * SD_Template_Wage * (1# + SD_Meb / 100#) / 1000000# 'Costs of negotation (including meb)
Credit_move_costs = SD_Wastage / 100# * SD_Oft_Asic_Wage * SD_Oft_Asic_Ft / 1000000#  'Costs of credit move (a resource waste - not needing to be financed through new taxes - hence no meb)
One_off_legal_costs = SD_Legal_Oneoff_Ft * SD_Legal_Oneoff_Wage * (1# + SD_Meb / 100#) / 1000000# 'One-off legal & policy costs (legal advice re how credit should be integrated into the Corporations Act; wording of new unfair contracts law; new discosure requirements)
Oneoff_admin = One_off_legal_costs + Credit_move_costs + Costs_negotiate

'[C2] Ongoing litigation costs and admin commencing in 2008-09 (2006-07 prices)
New_Actions = Round(SV_Percent_Actions / 100# * Actions, 0#) 'Businesses subject to new court-based enforcement measures per year (civil penalties & including class actions)
Transfer_fromOS = Share_Foreign / 100# * Share_Success / 100# * SD_Av_Fine * New_Actions / 1000000  'Fines collected from O/S firms (net gain) $m
False_cases = Round(SV_False_Positive / 100# * New_Actions, 0) 'False positive cases
Cost_false = False_cases * 100000# / 1000000# 'External cost of false cases = $1,000,000 per case (unfairness, excessivley risk averse behaviour by all firms) $m
Reg_costs = (SV_Preparation_Costs + SD_Days_Case * SD_Cost_Per_Day) * New_Actions * (1# + SD_Meb / 100#) / 1000000#  'Regulator's costs associated with new penalties (including court) and MEB financing $m
Bus_costs = (1# + SD_Business_Margin / 100#) * (SV_Preparation_Costs + SD_Days_Case * SD_Cost_Per_Day) * New_Actions / 1000000# 'Business' own costs $m 2007-08, 2006-007 prices
litigation_costs = (Bus_costs + Reg_costs + Cost_false - Transfer_fromOS)
Policy_costs = SD_Policy_Ft * SD_Policy_Wage / 1000000# 'Ongoing policy costs associated with new changes
Unfair_Admin_Costs = SD_Ongoing_Admin_Unfair * SD_Oft_Asic_Wage / 1000000#
Litigation_Policy_costs(1) = (litigation_costs + Policy_costs + Unfair_Admin_Costs) + Oneoff_admin 'All legal costs by firms and govt, all policy and admin costs, including re-structuring costs
admin_ratio = (Unfair_Admin_Costs + litigation_costs + Policy_costs) / con2008_09 * 100# 'Only ONGOING litigation & admin costs as a share of consumption


'****************************************
'PV and Anuity Gains in 2006-07 prices *
'****************************************

PV_Net_direct_Gain = YR_Net_direct_gain(1)
PV_Innovation_gain = Innovation_gain(1)
PV_Transact = Transact(1)
PV_Gross_Compliance_Savings = Gross_compliance_savings(1)
PV_Agency_Costs = agency_costs(1)
PV_Comp_Savings = Comp_Savings(1)
PV_Litigation_Policy_costs = Litigation_Policy_costs(1)

For Period = 2009 To (2007 + Yrs) ' IE "Yrs" IS THE NUMBER OF YEARS, STARTING WITH 2009, WHICH IS THE 2008-09 fiscal year
```

```
        T = Period - 2007 'This ensures that VARIABLE(0), which is the value for 2007-08 is not overwritten
        GDPCAP_before(T) = 100# * ((1# + D_Labhours(T) / 100#) * (1# + base_TFP / 100#) - 1#)
        GDPCAP_after(T) = 100# * ((1# + D_Labhours(T) / 100#) * (1# + New_TFP / 100#) - 1#)
        GDP_before(T) = GDP_before(T - 1) * (GDPCAP_before(T) / 100# + 1#) * (D_Pop(T) / 100# + 1#) 'GDP
BEFORE policy in 2006-07 prices
        GDP_after(T) = GDP_after(T - 1) * (GDPCAP_after(T) / 100# + 1#) * (D_Pop(T) / 100# + 1#) 'GDP AFTER
policy in 2006-07 prices
        CONS_before(T) = Lr_Con_Sh * GDP_before(T) 'household consumption BEFORE policy in 2006-07 prices
        CONS_after(T) = Lr_Con_Sh * GDP_after(T) 'household consumption AFTER policy in 2006-07 prices
        YR_Net_direct_gain(T) = (Direct_detriment / CON2005_06) * CONS_after(T) / ((1# + SD_Discount_Rate /
100#) ^ (T - 1))
        Innovation_gain(T) = LR_Total_Cons_Share * (GDP_after(T) - GDP_before(T)) / ((1# + SD_Discount_Rate /
100#) ^ (T - 1))
        Transact(T) = Transact_dividend_rate / 100# * CONS_after(T) / ((1# + SD_Discount_Rate / 100#) ^ (T - 1))
        Gross_compliance_savings(T) = (Gross_compliance_savings(1) / con2008_09) * CONS_after(T) / ((1# +
SD_Discount_Rate / 100#) ^ (T - 1))
        agency_costs(T) = (agency_costs(1) / con2008_09) * CONS_after(T) / ((1# + SD_Discount_Rate / 100#) ^ (T -
1))
        Comp_Savings(T) = (Comply_ratio / 100# * CONS_after(T)) / ((1# + SD_Discount_Rate / 100#) ^ (T - 1))
        Litigation_Policy_costs(T) = (admin_ratio / 100# * CONS_after(T)) / ((1# + SD_Discount_Rate / 100#) ^ (T - 1))
        PV_Net_direct_Gain = PV_Net_direct_Gain + YR_Net_direct_gain(T)
        PV_Innovation_gain = PV_Innovation_gain + Innovation_gain(T)
        PV_Transact = PV_Transact + Transact(T)
        PV_Gross_Compliance_Savings = PV_Gross_Compliance_Savings + Gross_compliance_savings(T)
        PV_Agency_Costs = PV_Agency_Costs + agency_costs(T)
        PV_Comp_Savings = PV_Comp_Savings + Comp_Savings(T)
        PV_Litigation_Policy_costs = PV_Litigation_Policy_costs + Litigation_Policy_costs(T)
    Next Period
    PV_NET_ONGOING_Compliance_Savings = PV_Gross_Compliance_Savings - PV_Agency_Costs
    PV_Net_Ongoing_and_Oneoff_Comp_Savings = PV_Gross_Compliance_Savings - PV_Agency_Costs -
OneOff_compliance_burdens


'Derive annuity values

    AN_Net_direct_Gain = PV_Net_direct_Gain * SD_Discount_Rate / 100#
    AN_Innovation_gain = PV_Innovation_gain * SD_Discount_Rate / 100#
    AN_Transact = PV_Transact * SD_Discount_Rate / 100#
    AN_Net_Ongoing_and_Oneoff_Comp_Savings = PV_Net_Ongoing_and_Oneoff_Comp_Savings *
SD_Discount_Rate / 100#
    AN_Litigation_Policy_costs = PV_Litigation_Policy_costs * SD_Discount_Rate / 100#
    AN_Gross_compliance_savings = PV_Gross_Compliance_Savings * SD_Discount_Rate / 100#
    AN_agency_costs = PV_Agency_Costs * SD_Discount_Rate / 100#
    AN_NET_ONGOING_Compliance_Savings = PV_NET_ONGOING_Compliance_Savings * SD_Discount_Rate /
100#
    AN_OneOff_compliance_burdens = OneOff_compliance_burdens * SD_Discount_Rate / 100#

'************************************************
'Summary values (2006-07 prices)  *
'************************************************

'net gains
```

```
    Net_Gain_2007_08 = YR_Net_direct_gain(1) + Transact(1) + Innovation_gain(1) +
    Net_Ongoing_and_Oneoff_Comp_Savings - Litigation_Policy_costs(1)
    PV_Net_Gain = PV_Net_direct_Gain + PV_Transact + PV_Innovation_gain +
    PV_Net_Ongoing_and_Oneoff_Comp_Savings - PV_Litigation_Policy_costs
    AN_Net_Gain = AN_Net_direct_Gain + AN_Transact + AN_Innovation_gain +
    AN_Net_Ongoing_and_Oneoff_Comp_Savings - AN_Litigation_Policy_costs

'net compliance gains should match with above - but a good check

    Net_Compliance_Gains = Gross_compliance_savings(1) - agency_costs(1) - OneOff_compliance_burdens
    PV_Net_Compliance_Gains = PV_Gross_Compliance_Savings - PV_Agency_Costs - OneOff_compliance_burdens
    AN_Net_Compliance_Gains = AN_Gross_compliance_savings - AN_agency_costs -
    AN_OneOff_compliance_burdens

'store simulation results

    SIM_AN_Net_direct_Gain(i - 1) = AN_Net_direct_Gain
    SIM_AN_Transact(i - 1) = AN_Transact
    SIM_AN_Innovation_gain(i - 1) = AN_Innovation_gain
    SIM_Net_CMPLC_gains(i - 1) = AN_Net_Ongoing_and_Oneoff_Comp_Savings
    SIM_AN_LITPCY_Costs(i - 1) = AN_Litigation_Policy_costs
    SIM_AN_NET_Gain(i - 1) = AN_Net_Gain

Next i

Dim End_Val As Integer
If iters > 10000 Then
    endval = 10000 ' this stops you printing 100,000 observations if the settings for sims are changed and  a very large
    simulation is run
Else: End_Val = iters
End If

Application.ScreenUpdating = False

If OptShow Then
    Sheets("Summary").Select
    ActiveWorkbook.Worksheets.Add(After:=ActiveSheet).name = "Sim_Print"
    Sheets("Sim_Print").Select
    For i = 1 To End_Val
        Worksheets("Sim_Print").Cells(1, 1).Value = "Draw number'"
        Worksheets("Sim_Print").Cells(1, 2).Value = "Net direct gain for consumers - reduced detriment'"
        Worksheets("Sim_Print").Cells(1, 3).Value = "Transaction efficiencies'"
        Worksheets("Sim_Print").Cells(1, 4).Value = "Innovation gains'"
        Worksheets("Sim_Print").Cells(1, 5).Value = "Net savings in compliance costs for business'"
        Worksheets("Sim_Print").Cells(1, 6).Value = "Administrative and litigation costs'"
        Worksheets("Sim_Print").Cells(1, 7).Value = "The net economic benefit'"
        Worksheets("Sim_Print").Cells(1 + i, 1).Value = i
        Worksheets("Sim_Print").Cells(1 + i, 2).Value = SIM_AN_Net_direct_Gain(i - 1)
        Worksheets("Sim_Print").Cells(1 + i, 3).Value = SIM_AN_Transact(i - 1)
        Worksheets("Sim_Print").Cells(1 + i, 4).Value = SIM_AN_Innovation_gain(i - 1)
        Worksheets("Sim_Print").Cells(1 + i, 5).Value = SIM_Net_CMPLC_gains(i - 1)
```

```
        Worksheets("Sim_Print").Cells(1 + i, 6).Value = SIM_AN_LITPCY_Costs(i - 1)
        Worksheets("Sim_Print").Cells(1 + i, 7).Value = SIM_AN_NET_Gain(i - 1)
    Next i
End If

Sheets("summary").Select
Range("a75:G83").Select
With Selection.Interior
    .ColorIndex = 36
    .Pattern = xlSolid
End With


With Selection
    .Borders(xlEdgeLeft).LineStyle = xlContinuous
    .Borders(xlEdgeLeft).Weight = xlMedium
    .Borders(xlEdgeLeft).ColorIndex = xlAutomatic
    .Borders(xlEdgeTop).LineStyle = xlContinuous
    .Borders(xlEdgeTop).Weight = xlMedium
    .Borders(xlEdgeTop).ColorIndex = xlAutomatic
    .Borders(xlEdgeBottom).LineStyle = xlContinuous
    .Borders(xlEdgeBottom).Weight = xlMedium
    .Borders(xlEdgeBottom).ColorIndex = xlAutomatic
    .Borders(xlEdgeRight).LineStyle = xlContinuous
    .Borders(xlEdgeRight).Weight = xlMedium
    .Borders(xlEdgeRight).ColorIndex = xlAutomatic
    .Borders(xlInsideVertical).LineStyle = xlNone
    .Borders(xlInsideHorizontal).LineStyle = xlNone
End With


Worksheets("Summary").Cells(74, 1).Value = "Table 5: Simulation Results for annuity values"
Worksheets("Summary").Cells(74, 1).Font.Size = 12
Worksheets("Summary").Cells(75, 2).Value = "Results are in $ million"
Worksheets("Summary").Cells(76, 2).Value = "Average"
Worksheets("Summary").Cells(76, 3).Value = "StDev"
Worksheets("Summary").Cells(76, 4).Value = "Max"
Worksheets("Summary").Cells(76, 5).Value = "Min"
Worksheets("Summary").Cells(76, 6).Value = "5% tail"
Worksheets("Summary").Cells(76, 7).Value = "95% tail"
Worksheets("Summary").Cells(78, 1).Value = "Net avoided direct detriment for consumers"
Worksheets("Summary").Cells(79, 1).Value = "Gains from reduced transaction costs"
Worksheets("Summary").Cells(80, 1).Value = "Gains from increased innovation"
Worksheets("Summary").Cells(81, 1).Value = "Net gains from reduced business compliance costs"
Worksheets("Summary").Cells(82, 1).Value = "All legal and administrative costs"
Worksheets("Summary").Cells(83, 1).Value = "Net gain"

Worksheets("Summary").Cells(78, 2).Value = Application.Average(SIM_AN_Net_direct_Gain)
Worksheets("Summary").Cells(79, 2).Value = Application.Average(SIM_AN_Transact)
Worksheets("Summary").Cells(80, 2).Value = Application.Average(SIM_AN_Innovation_gain)
```

```
Worksheets("Summary").Cells(81, 2).Value = Application.Average(SIM_Net_CMPLC_gains)
Worksheets("Summary").Cells(82, 2).Value = Application.Average(SIM_AN_LITPCY_Costs)
Worksheets("Summary").Cells(83, 2).Value = Application.Average(SIM_AN_NET_Gain)

Worksheets("Summary").Cells(78, 3).Value = Application.StDev(SIM_AN_Net_direct_Gain)
Worksheets("Summary").Cells(79, 3).Value = Application.StDev(SIM_AN_Transact)
Worksheets("Summary").Cells(80, 3).Value = Application.StDev(SIM_AN_Innovation_gain)
Worksheets("Summary").Cells(81, 3).Value = Application.StDev(SIM_Net_CMPLC_gains)
Worksheets("Summary").Cells(82, 3).Value = Application.StDev(SIM_AN_LITPCY_Costs)
Worksheets("Summary").Cells(83, 3).Value = Application.StDev(SIM_AN_NET_Gain)

Worksheets("Summary").Cells(78, 4).Value = Application.Max(SIM_AN_Net_direct_Gain)
Worksheets("Summary").Cells(79, 4).Value = Application.Max(SIM_AN_Transact)
Worksheets("Summary").Cells(80, 4).Value = Application.Max(SIM_AN_Innovation_gain)
Worksheets("Summary").Cells(81, 4).Value = Application.Max(SIM_Net_CMPLC_gains)
Worksheets("Summary").Cells(82, 4).Value = Application.Max(SIM_AN_LITPCY_Costs)
Worksheets("Summary").Cells(83, 4).Value = Application.Max(SIM_AN_NET_Gain)

Worksheets("Summary").Cells(78, 5).Value = Application.Percentile(SIM_AN_Net_direct_Gain, (1# / iters))
Worksheets("Summary").Cells(79, 5).Value = Application.Percentile(SIM_AN_Transact, (1# / iters))
Worksheets("Summary").Cells(80, 5).Value = Application.Percentile(SIM_AN_Innovation_gain, (1# / iters))
Worksheets("Summary").Cells(81, 5).Value = Application.Percentile(SIM_Net_CMPLC_gains, (1# / iters))
Worksheets("Summary").Cells(82, 5).Value = Application.Percentile(SIM_AN_LITPCY_Costs, (1# / iters))
Worksheets("Summary").Cells(83, 5).Value = Application.Percentile(SIM_AN_NET_Gain, (1# / iters))

Worksheets("Summary").Cells(78, 6).Value = Application.Percentile(SIM_AN_Net_direct_Gain, 0.05)
Worksheets("Summary").Cells(79, 6).Value = Application.Percentile(SIM_AN_Transact, 0.05)
Worksheets("Summary").Cells(80, 6).Value = Application.Percentile(SIM_AN_Innovation_gain, 0.05)
Worksheets("Summary").Cells(81, 6).Value = Application.Percentile(SIM_Net_CMPLC_gains, 0.05)
Worksheets("Summary").Cells(82, 6).Value = Application.Percentile(SIM_AN_LITPCY_Costs, 0.05)
Worksheets("Summary").Cells(83, 6).Value = Application.Percentile(SIM_AN_NET_Gain, 0.05)

Worksheets("Summary").Cells(78, 7).Value = Application.Percentile(SIM_AN_Net_direct_Gain, 0.95)
Worksheets("Summary").Cells(79, 7).Value = Application.Percentile(SIM_AN_Transact, 0.95)
Worksheets("Summary").Cells(80, 7).Value = Application.Percentile(SIM_AN_Innovation_gain, 0.95)
Worksheets("Summary").Cells(81, 7).Value = Application.Percentile(SIM_Net_CMPLC_gains, 0.95)
Worksheets("Summary").Cells(82, 7).Value = Application.Percentile(SIM_AN_LITPCY_Costs, 0.95)
Worksheets("Summary").Cells(83, 7).Value = Application.Percentile(SIM_AN_NET_Gain, 0.95)

'*******
 If OptYes Then
    Sheets("Summary").Select
    ActiveWorkbook.Worksheets.Add(After:=ActiveSheet).name = "Histogram Data"
    Sheets("Histogram Data").Select

    Worksheets("Histogram Data").Cells(1, 1).Value = "Upper bin values for detriment"
    Worksheets("Histogram Data").Cells(1, 3).Value = "Upper bin values for transaction efficiencies"
    Worksheets("Histogram Data").Cells(1, 5).Value = "Upper bin values for innovation gains"
    Worksheets("Histogram Data").Cells(1, 7).Value = "Upper bin values for reduced business compliance costs"
    Worksheets("Histogram Data").Cells(1, 9).Value = "Upper bin values for all legal and administrative costs"
```

```
    Worksheets("Histogram Data").Cells(1, 11).Value = "Upper bin values for net gains"

    Worksheets("Histogram Data").Cells(1, 2).Value = "Net avoided direct detriment for consumers"
    Worksheets("Histogram Data").Cells(1, 4).Value = "Gains from reduced transaction costs"
    Worksheets("Histogram Data").Cells(1, 6).Value = "Gains from increased innovation"
    Worksheets("Histogram Data").Cells(1, 8).Value = "Net gains from reduced business compliance costs"
    Worksheets("Histogram Data").Cells(1, 10).Value = "All legal and administrative costs"
    Worksheets("Histogram Data").Cells(1, 12).Value = "Net gain"

    Call Hist(10, SIM_AN_Net_direct_Gain, iters, 1)
    Call Hist(10, SIM_AN_Transact, iters, 2)
    Call Hist(10, SIM_AN_Innovation_gain, iters, 3)
    Call Hist(10, SIM_Net_CMPLC_gains, iters, 4)
    Call Hist(10, SIM_AN_LITPCY_Costs, iters, 5)
    Call Hist(10, SIM_AN_NET_Gain, iters, 6)

    Range("A2:L11").Select
    Selection.NumberFormat = "0"

    Range("A1:L11").Select
    With Selection.Font
        .name = "Arial"
        .Size = 8
        .ColorIndex = xlAutomatic
    End With

     'graph results
    'ActiveWorkbook.Names.Add name:="freq_det", RefersToR1C1:= _
    '"='Histogram Data'!R2C1:R11C2"
    Application.ScreenUpdating = False
    Call Graph_add("freq_det", "R2C1:R11C2")
    Call Graph_add("freq_trans", "R2C3:R11C4")
    Call Graph_add("freq_innov", "R2C5:R11C6")
    Call Graph_add("freq_comply", "R2C7:R11C8")
    Call Graph_add("freq_admin", "R2C9:R11C10")
    Call Graph_add("freq_netgain", "R2C11:R11C12")
    Call ArrangeMyCharts
End If

Application.Goto Reference:="sim_table"
CmdCancel = True
End Sub
```

FrmWarning code

```
Private Sub CmdBut_cancel_Click()
   Unload Me
```

```
    FrmCalc.OptButYesRestore = False
    FrmCalc.OptButNoRestore = True
End Sub

Private Sub CmdBut_no_Click()
    Unload Me
    FrmCalc.OptButNoRestore = True
End Sub

Private Sub CmdBut_Yes_Click()
    Range("I16:L61").Select
    Selection.Copy
    Range("c16:f61").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False
    Unload Me
End Sub

Private Sub UserForm_Click()
End Sub
```

Module 1 code

```
Sub RunProgram()
'
FrmCalc.Show
Application.ScreenUpdating = False
End Sub
```

References

Productivity Commission 2005, *Economic Implications of an Ageing Australia*, Research Report, Canberra (including technical paper no. 1).

Cuxson, G., Huo, S. and Fry, J. 2007, *A User's Guide to the Modified Demographic and Economic Model*, Productivity Commission Technical Paper, Melbourne.