Adding New Variables into Corporate Tax Microsimulation Model

1. Include new variable being read from tax return into the “read” variables part of corprecords_variables.json.
   Example: We are reading in the “Opening balance of the written down value (WDV) on first block of assets in Plant and Machinery” (along with other variables from the Schedule that calculates depreciation).

```
    "PWR_DOWN_VAL_1ST_DAY_PY_15P": {
        "type": "float",
        "desc": "Opening WDV for Plant and Machinery for Block1",
        "form": {"2017": "ITR-6 Sch. DPM 3(i)"}
```

2. Include the function that uses this variable into corpfunction.py
   Note: if logic is peculiar to corporations then include in corpfunctions.py, otherwise include it in functions.py which has all functions for personal income tax.
   Example: Here we are calculating the depreciation deductions “dep_amt_pm1” for the first block of assets in Plant and Machinery and also the closing WDV “close_wdv_pm1”. We then use the depreciation deduction calculated to calculate the Income from business and profession.

```
@iterate_jit(nopython=True)
def depreciation_PM(dep_rate_pm1, PWR_DOWN_VAL_1ST_DAY_PY_15P,
                    PADDTNS_180_DAYS__MOR_PY_15P, PCR34_PY_15P,
                    PADDTNS_LESS_180_DAYS_15P, PCR7_PY_15P,
                    PEXP_INCURRD_TRF_ASSTS_15P, PCAP_GAINS_LOSS_SEC50_15P):
    dep_amt_pm1 = (PWR_DOWN_VAL_1ST_DAY_PY_15P +
PADDTNS_180_DAYS__MOR_PY_15P -
                   PCR34_PY_15P) * dep_rate_pm1
    dep_amt_pm1 += ((PADDTNS_LESS_180_DAYS_15P - PCR7_PY_15P) *
                    (dep_rate_pm1 / 2))
    close_wdv_pm1 = (PWR_DOWN_VAL_1ST_DAY_PY_15P +
                     PADDTNS_180_DAYS__MOR_PY_15P - PCR34_PY_15P +
                     PADDTNS_LESS_180_DAYS_15P - PCR7_PY_15P - dep_amt_pm1)
    cap_gain_pm1 = (PCR34_PY_15P + PCR7_PY_15P -
PWR_DOWN_VAL_1ST_DAY_PY_15P -
                    PADDTNS_180_DAYS__MOR_PY_15P - PEXP_INCURRD_TRF_ASSTS_15P -
                    PADDTNS_LESS_180_DAYS_15P)
    # Consider unusual cases when Capital Gains is negative and block DNE
    if (PCAP_GAINS_LOSS_SEC50_15P >= 0):
        cap_gain_pm1 = max(0.0, cap_gain_pm1)
    return (dep_amt_pm1, close_wdv_pm1)


@iterate_jit(nopython=True)
def corp_income_business_profession(dep_amt_pm1,
PRFT_GAIN_BP_OTHR_SPECLTV_BUS,
                                    PRFT_GAIN_BP_SPECLTV_BUS,
                        PRFT_GAIN_BP_SPCFD_BUS,
                        PRFT_GAIN_BP_INC_115BBF, Income_BP):
    """
    Compute Income from Business and Profession by adding the different
    sub-heads (i.e speculative, non-speculative, specified, patents, etc)
    """
    # TODO: when reading from schedule BP, calculate Income_BP from the read
    # TODO: variables of the schedule
    Income_BP = (PRFT_GAIN_BP_OTHR_SPECLTV_BUS +
    PRFT_GAIN_BP_SPECLTV_BUS +
    PRFT_GAIN_BP_SPCFD_BUS + PRFT_GAIN_BP_INC_115BBF - dep_amt_pm1)
    return Income_BP
```

3. We now need to include any new variables created under the calculated or “calc” section in corprecords_variables.json
   Example: We include the two new variables “dep_amt_pm1” and “close_wdv_pm1” under the calculated or “calc” section

```
            "dep_amt_pm1": {
             "type": "float",
             "desc": "Depreciation allowance for Plant and Machinery Block1",
             "form": {"2017": "ITR-6 Schedule DPM 15(i)"}
            },
            "close_wdv_pm1": {
             "type": "float",
             "desc": "Closing Balance of WDV for Plant and Machinery Block1",
             "form": {"2017": "ITR-6 Schedule DPM 18(i)"}
          },
```

4. Incorporate any function that needs to be calculated for each record of the simulation into calculator.py.
   Example: we incorporate the newly created function corp_income_business_profession into calculator.py which tells the model to import these functions from corpfunctions.py

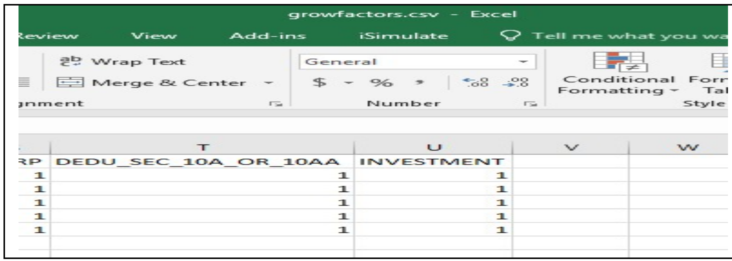
```
          from taxcalc.corpfunctions import (depreciation_PM,
          corp_income_business_profession,
                                    corp_GTI_before_set_off, GTI_and_losses,
                      cit_liability)
```

5. Further, ask calculator to compute this variable for corporecords in the “calc_all” function in calculator.py.
   Example: In this example we include the functions depreciation_PM(self.__policy, self.__corprecords) and corp_income_business_profession(self.__policy, self.__corprecords) to be calculate policy for every corprecord.

```
              # Corporate calculations
              net_rental_income(self.__policy, self.__corprecords)
              depreciation_PM((self.__policy, self.__corprecords)
              corp_income_business_profession(self.__policy, self.__corprecords)
              total_other_income(self.__policy, self.__corprecords)
```

```
                    current_year_losses(self.__policy, self.__corprecords)
```

6. Include the relevant variable in growfactors.csv if it needs to be blown up.
   Example: In this example all capital investment related variables are blown up by the same factor named – “INVESTMENT”



	T	U	V	W
RP	DEDU_SEC_10A_OR_10AA	INVESTMENT		
1	1	1		
1	1	1		
1	1	1		
1	1	1		
1	1	1		

7. Include this growfactor variable into growfactor.py
   Example: In this example we incorporate “INVESTMENT” into growfactor.py

```
                    VALID_NAMES = set(['CPI', 'SALARY', 'RENT', 'BP_NONSPECULATIVE',
                                'BP_SPECULATIVE', 'BP_SPECIFIED', 'BP_PATENT115BBF',
                                'STCG_APPRATE', 'OINCOME', 'DEDUCTIONS',
                                'DEDU_SEC_10A_OR_10AA', 'ST_CG_AMT_1',
                                'ST_CG_AMT_2', 'LT_CG_AMT_1', 'LT_CG_AMT_2',
                                'LOSSES_CY', 'LOSSES_BF', 'AGRI_INCOME', 'CORP',
                               'INVESTMENT'])
```

8. In corprecords.py we need to include this variable and its growfactor variable to allow the variable to be estimated for subsequent years. We incorporate this in the “blowup” method.
   Example: we pull-in the growfactor for “INVESTMENT” from the grofactor.csv file and use it to blow up the capital asset variables for the next year.

```
                    GF_DEDUCTION_10AA = self.gfactors.factor_value('DEDU_SEC_10A_OR_10AA',
                                                     year)
                    GF_NET_AGRC_INCOME = self.gfactors.factor_value('AGRI_INCOME', year)
                   GF_INVESTMENT = self.gfactors.factor_value('INVESTMENT', year)


*****
                    self.PWR_DOWN_VAL_1ST_DAY_PY_15P *= GF_INVESTMENT
                    self.PADDTNS_180_DAYS__MOR_PY_15P *= GF_INVESTMENT
                    self.PCR34_PY_15P *= GF_INVESTMENT
                    self.PADDTNS_LESS_180_DAYS_15P *= GF_INVESTMENT
                    self.PCR7_PY_15P *= GF_INVESTMENT
                    self.PEXP_INCURRD_TRF_ASSTS_15P *= GF_INVESTMENT
                   self.PCAP_GAINS_LOSS_SEC50_15P *= GF_INVESTMENT
```

9. We need to do the same when we deal with panel data. We incorporate these growfactors in the “extract_panel_year” function in corprecords.py.
   Example:

```
                    BF_DEDUCTION_10AA = blowup_data['DEDUCT_SEC_10A_OR_10AA']
                    BF_NET_AGRC_INC = blowup_data['NET_AGRC_INCOME']
```

```
                BF_INVESTMENT = blowup_data['INVESTMENT']

*****
                temp = data1['PWR_DOWN_VAL_1ST_DAY_PY_15P']
                data1['PWR_DOWN_VAL_1ST_DAY_PY_15P'] = temp * BF_INVESTMENT
                temp = data1['PADDTNS_180_DAYS__MOR_PY_15P']
                data1['PADDTNS_180_DAYS__MOR_PY_15P'] = temp * BF_INVESTMENT
                temp = data1['PCR34_PY_15P']
                data1['PCR34_PY_15P'] = temp * BF_INVESTMENT
                temp = data1['PADDTNS_LESS_180_DAYS_15P']
                data1['PADDTNS_LESS_180_DAYS_15P'] = temp * BF_INVESTMENT
                temp = data1['PCR7_PY_15P']
                data1['PCR7_PY_15P'] = temp * BF_INVESTMENT
                temp = data1['PEXP_INCURRD_TRF_ASSTS_15P']
                data1['PEXP_INCURRD_TRF_ASSTS_15P'] = temp * BF_INVESTMENT
                temp = data1['PCAP_GAINS_LOSS_SEC50_15P']
                data1['PCAP_GAINS_LOSS_SEC50_15P'] = temp * BF_INVESTMENT
```

10. We need to incorporate variables that are carried between panel years such as WDV of capital assets, losses, etc. We include this in corprecords.py in the function “increment_panel_year” function.
    Example: we include the variable being carried forward which in this case is “close_wdv_pm1” the closing WDV of the first block of assets of Plant and Machinery. We then ensure that the opening WDV for the next year is the closing WDV of the previous year. This is done by setting data2['PWR_DOWN_VAL_1ST_DAY_PY_15P'] = np.where(to_update, data2['close_wdv_pm1'], data2['PWR_DOWN_VAL_1ST_DAY_PY_15P']) where data2 relates to the subsequent year.

```
                carryforward_df = pd.DataFrame({'ID_NO': self.ID_NO,
                                     'newloss1': self.newloss1,
                                     'newloss2': self.newloss2,
                                     'newloss3': self.newloss3,
                                     'newloss4': self.newloss4,
                                     'newloss5': self.newloss5,
                                     'newloss6': self.newloss6,
                                     'newloss7': self.newloss7,
                                     'newloss8': self.newloss8,
                                    'close_wdv_pm1': self.close_wdv_pm1})

*****

                                    data2['LOSS_LAG7'])
                                    data2['LOSS_LAG8'] = np.where(to_update, data2['newloss8'],
                                    data2['LOSS_LAG8'])
                                    temp = np.where(to_update, data2['close_wdv_pm1'],
                                    data2['PWR_DOWN_VAL_1ST_DAY_PY_15P'])
                                    data2['PWR_DOWN_VAL_1ST_DAY_PY_15P'] = temp
                                    data3 = data2[to_keep]
                                   self._read_data(data3)
```