PIMA COUNTY MODEL FORMULAS *Revised 2/18/2020*

Single Family Model – Tax Years 2016 to Present

```
Parcel Value = CONVERT(NUMERIC(15,0),(
          ((isnull([value],1.00) *  blv  * power(convert(decimal(7,3),nbhd_pf),convert(decimal(21,15),site_exp))
        + ((( BASERATE * SQFT * power( (sqft/meanimpsqft) , bsize_exp)
                * ( 1 + case class when 0 then LM_ADJ when 1 then r1_adj
                        when 2 then r2_adj when 3 then r3_adj
                        when 4 then r4_adj when 5 then r5_adj
                        when 6 then r6_adj end  * 1)
                * power((case stories when 1 then 1.000
                        when 1.5 then STY_FAC15
                        when 2   then STY_FAC2
                        else 1.000 end),1)
                * power( (case when heat in ( '1','4','7' ) then HEATTA
                        when heat in ( '0','2','3','5','6','8' ) then HEATTB
                        when heat in ( '9' )                     then HEATNO
                        else 1.000 end) , 1)
                * power((case when cool in ( '0' )  then COOLAC
                        when cool in ( '1' )  then COOLEVAP
                        when cool in ( '2' )  then COOLWINDOW
                        when cool in ( '9' )  then COOLNO
                        else 1.000 end ),1)
                * ( 1 + case quality when 1 then GRADE1_ADJ
                        when 2 then GRADE2_ADJ
                        when 3 then GRADE34_ADJ
                        when 4 then GRADE34_ADJ else 0.000000  end *1)
                +(( GAR_RATE * case garage  when '1' then (garagecapacity +.0000)
                        when '3' then 0.0000000
                        when '6' then .50000* (garagecapacity+.0000)
                        when '9' then 0.0000000 else 0.000000 end
                + CARPORT_RATE * case garage  when '1' then 0.00000
                        when '3' then (garagecapacity+.0000)
                        when '6' then .50000* (garagecapacity +.0000)
                        when '9' then 0.0000000 else 0.000000 end
                + PATIO_RATE   *  case patio   when '1' then 0.00000
                        when '3' then (pationumber +.0000)
                        when '6' then .50000* (pationumber+.0000) else 0.000000 end
                + POOL_FAC *  case when poolarea >= 250 then 1.00000 else 0 end
                   )
                * (1+ .50000 * case class  when 0 then LM_ADJ when 1 then r1_adj
                        when 2 then r2_adj when 3 then r3_adj
                        when 4 then r4_adj when 5 then r5_adj
                        when 6 then r6_adj else 0 end *1 )
                * (1+ .50000 * case quality when 1 then GRADE1_ADJ
                        when 2 then GRADE2_ADJ
                        when 3 then GRADE34_ADJ
                        when 4 then GRADE34_ADJ else 0  end *1) )
                   )
                * power((1-sqrt(case when (CASE WHEN (taxyear-2)-[year] < 0 THEN 0
                        ELSE (taxyear-2)-[year] END) > 50 then 50
                        else (CASE WHEN (taxyear-2)-[year] < 0 THEN 0 ELSE (taxyear-2)-[year] END) end )
                        /( eqpctgood *100.00 )
                  ) , pgood_exp )  + enhancement
        )
                * ((  power(isnull([value],1.00),1) -1 ) * eqloc_adj + 1 )

                ) )
                 *isnull(adjustment,1.0)
                 + isnull(ccsaddfcv,0)
                 +isnull(nlrghou,0)
                )
                * isnull(trend,1.0)
                )
         )
```

Condominium Model – Tax Years 2016 to Present

```
Parcel Value = convert(dec(9,0), exp(Constant)
        * power(sqft,LN_SQFT)
        * power(1-(SQRT(AGE)/25),LNPCTGOOD25)
        * power(bGARNUM,case when GARAGECAPACITY > 4 then 4 else GARAGECAPACITY end )
        * power(bcarports,case when GARAGECAPACITY > 4 then 4 else GARAGECAPACITY end )
        * exp(case when evap_bin = 1 then EVAP else 0 end)
        * exp(case when story_15_bin=1 then STORY15 else 0 end )
        * exp(case when golf_bin = 1 then Golf else 0 end)
        * exp(case when traffic_bin =1 then TRAFFIC else 0 end)
        * exp(isnull(NBHDCOEF,0))
        * TREND
        )
```

Town Home Model – Tax Years 2016 to Present

```
Parcel Value = convert(dec(9,0), exp(constant)
        * power(sqft,ln_sqft)
        * power(1-(SQRT(AGE))/25,LNPCTGOOD25)
        * power(LSIZE10k/lsizemedian,LNLSIZE)
        * power(1+ bGARNUM,case
                when garage = 1 and GARAGECAPACITY  > 4 then 4
                when garage = 1 and GARAGECAPACITY <=4 then GARAGECAPACITY
                when garage = 6 and .5*GARAGECAPACITY > 4 then 4
                when garage = 6 and .5*GARAGECAPACITY <=4 then GARAGECAPACITY
                else 0
                end)
        * power(1+ bCARPORTS,case
                when garage = 3 and GARAGECAPACITY  > 4 then 4
                when garage = 3 and GARAGECAPACITY <=4 then GARAGECAPACITY
                when garage = 6 and .5*GARAGECAPACITY > 4 then 4
                when garage = 6 and .5*GARAGECAPACITY <=4 then GARAGECAPACITY
                else 0
                end  )
        * power(1+ bPatCov,case
                when PATIO = 3 then pationumber
                when PATIO = 6 then .5*PATIONUMBER
                else 0
                end)
        * EXP(CASE WHEN AZIS_BIN = 1 THEN AZIS ELSE 0 END)
        * EXP(CASE WHEN POOL_BIN = 1 THEN [POOL] ELSE 0 END)
        * EXP(CASE WHEN GOLF_BIN = 1 THEN GOLF ELSE 0 END)
        * EXP(CASE WHEN DETACHED_BIN = 1 THEN DETACHED ELSE 0 END)
        * EXP(CASE WHEN STORY_15_BIN = 1 THEN STORY15 ELSE 0 END)
        * EXP(CASE WHEN TRAFFIC_BIN = 1 THEN TRAFFIC ELSE 0 END)
        * EXP(isnull(nbhdcoef,0))
        * TREND
        )
```

Vacant Residential Urban Land Model – Tax Year 2018

```
Parcel Value=CONVERT(numeric(12,0), CASE
        WHEN presentworth = 1 THEN ((
                ROUND(
                        (
                        (
                        (
        EXP(constant_coef)
        * POWER(EXP(ResMH_unsubd_coef), ResMH_unsubd)
        * POWER(EXP(Dev_subd_coef), Dev_subd)
        * POWER((asrsqft_lm/landsize_cntr), ln_lotsize_coef)
        * POWER((z/med_elev), ln_z_cntr_co_coef)
        * POWER(EXP(golf120ft_coef), golf120ft)
        * (CASE
                WHEN gated IS NOT NULL THEN POWER(EXP(gated_coef), gated)
                ELSE 1.0
                END)
        * (1-(POWER(EXP(FldCountyRatio_coef), FldCounty_ratio)*FldCounty))
        * (1-(POWER(EXP(FldDBRatio_coef), FldDB_ratio)*FldDB))
        * EXP(ISNULL([value],0))
                        )       * asrsqft_lm
                        )
                        * CASE
                        WHEN mp_PW IS NOT NULL AND presentworth = 1 THEN mp_PW_trend
                        WHEN presentworth = 0 AND assr_mp IS NOT NULL  and mp_trend is not null THEN mp_trend
                        when presentworth = 0 and mp is null and landsubarea IS NOT NULL
                        and subarea_trend is not null THEN subarea_trend
                        ELSE 1.0
                        END
                        ) /1000,0
                        )*1000
                        ) * ISNULL(valadj,1.00))/POWER(1.10,5)
        ELSE
                ((
                ROUND(
                (
                (
                (
                EXP(constant_coef)
                * POWER(EXP(ResMH_unsubd_coef), ResMH_unsubd)
                * POWER(EXP(Dev_subd_coef), Dev_subd)
                * POWER((asrsqft_lm/landsize_cntr), ln_lotsize_coef)
                * POWER((z/med_elev), ln_z_cntr_co_coef)
                * POWER(EXP(golf120ft_coef), golf120ft)
                * (CASE
                        WHEN gated IS NOT NULL THEN POWER(EXP(gated_coef), gated)
                        ELSE 1.0
                        END)
                * (1-(POWER(EXP(FldCountyRatio_coef), FldCounty_ratio)*FldCounty))
                * (1-(POWER(EXP(FldDBRatio_coef), FldDB_ratio)*FldDB))
                * EXP(ISNULL([value],0))
                )       * asrsqft_lm
                )
                )
                * CASE
                WHEN mp_PW IS NOT NULL AND presentworth = 1 THEN mp_PW_trend
                WHEN presentworth = 0 AND assr_mp IS NOT NULL  and mp_trend is not null THEN mp_trend
                when presentworth = 0 and mp is null and landsubarea IS NOT NULL
                and subarea_trend is not null THEN subarea_trend
                ELSE 1.0
                END
                ) /1000,0
                )*1000
                ) * ISNULL(valadj,1.00))
                END
        )
```

Vacant Residential Urban Land Model – Tax Year 2019

```
Parcel Value= CONVERT(numeric(12,0), CASE
                WHEN presentworth = 1 THEN ((
                ROUND(
                (
                (
                (
                EXP(constant) * POWER(EXP(ResMH_unsubd_COEF), ResMH_unsubd)
                * power(exp(resmh_subd_COEF),resmh_subd)
                * POWER(EXP(Dev_subd_COEF), Dev_subd)
                * POWER((asrsqft_lm/landsize_cntr), ln_lotsize_COEF)
                * POWER((z/med_elev), ln_z_cntr_co_COEF)
                * POWER(EXP(golf120ft_COEF), golf120ft)
                * (CASE
                WHEN gated IS NOT NULL THEN POWER(EXP(gated_COEF), gated)
                ELSE 1.0
                END)
                * (CASE WHEN FldCounty = 0 THEN 1
                                ELSE POWER((1-EXP(FLDCOUNTYRATIO_COEF)),FldCounty_ratio)
                                END)
                * (CASE WHEN Flddb = 0 THEN 1
                                ELSE POWER((1-EXP(FLDDBRATIO_COEF)),Flddb_ratio)
                                END)
                * exp(isnull([value],0))
                ) * asrlotsize
                )
                * CASE
                WHEN mp_PW IS NOT NULL AND presentworth = 1 THEN PW_trend
                WHEN presentworth = 0 AND assr_mp IS NOT NULL  and mp_trend is not null THEN mp_trend
                when presentworth = 0 and mp is null and landsubarea IS NOT NULL and subarea_trend is not null THEN
                subarea_trend
                ELSE 1.0
                END
                ) /1000,0
                )*1000
                ) * ISNULL(valadj,1.00))/POWER(1.10,5)
                ELSE
                ((
                ROUND(
                (
                (
                (
                EXP(constant)
                * POWER(EXP(ResMH_unsubd_COEF), ResMH_unsubd)
                * power(exp(resmh_subd_COEF),resmh_subd)
                * POWER(EXP(Dev_subd_COEF), Dev_subd)
                * POWER((asrsqft_lm/landsize_cntr), ln_lotsize_COEF)
                * POWER((z/med_elev), ln_z_cntr_co_COEF)
                * POWER(EXP(golf120ft_COEF), golf120ft)
                * (CASE
                WHEN gated IS NOT NULL THEN POWER(EXP(gated_COEF), gated)
                ELSE 1.0
                END)
                * (CASE WHEN FldCounty = 0 THEN 1
                                ELSE POWER((1-EXP(FLDCOUNTYRATIO_COEF)),FldCounty_ratio)
                                END)
                * (CASE WHEN Flddb = 0 THEN 1
                                ELSE POWER((1-EXP(FLDDBRATIO_COEF)),Flddb_ratio)
                                END)
                * exp(isnull([value],0))
                ) * asrlotsize
                )
                * CASE
                WHEN mp_PW IS NOT NULL AND presentworth = 1 THEN PW_trend
                WHEN presentworth = 0 AND assr_mp IS NOT NULL  and mp_trend is not null THEN mp_trend
                when presentworth = 0 and mp is null and landsubarea IS NOT NULL and subarea_trend is not null THEN
                subarea_trend
                ELSE 1.0
                END
                ) /1000,0
                )*1000
                ) * ISNULL(valadj,1.00))
                END
                )
```

Vacant Residential Urban Land Model – Tax Year 2020 to present

```
Parcel Value= CONVERT(numeric(12,0), CASE
                WHEN presentworth = 1 THEN ((
                ROUND(
                (
                (
                (
                EXP(constant) * POWER(EXP(ResMH_unsubd_COEF), ResMH_unsubd)
                * power(exp(resmh_subd_COEF),resmh_subd)
                * POWER(EXP(Dev_subd_COEF), Dev_subd)
                * POWER((asrsqft_lm/landsize_cntr), ln_lotsize_COEF)
                * POWER((z/med_elev), ln_z_cntr_co_COEF)
                * POWER(EXP(golf120ft_COEF), golf120ft)
                * (CASE
                WHEN gated IS NOT NULL THEN POWER(EXP(gated_COEF), gated)
                ELSE 1.0
                END)
                * (CASE WHEN FldCounty = 0 THEN 1
                                ELSE POWER((1-EXP(FLDCOUNTYRATIO_COEF)),FldCounty_ratio)
                                END)
                * (CASE WHEN Flddb = 0 THEN 1
                                ELSE POWER((1-EXP(FLDDBRATIO_COEF)),Flddb_ratio)
                                END)
                * exp(isnull([value],0))
                ) * asrlotsize
                )
                * trend
                ) /1000,0
                )*1000
                ) * ISNULL(valadj,1.00))/POWER(1.10,5)
                ELSE
                ((
                ROUND(
                (
                (
                (
                EXP(constant)
                * POWER(EXP(ResMH_unsubd_COEF), ResMH_unsubd)
                * power(exp(resmh_subd_COEF),resmh_subd)
                * POWER(EXP(Dev_subd_COEF), Dev_subd)
                * POWER((asrsqft_lm/landsize_cntr), ln_lotsize_COEF)
                * POWER((z/med_elev), ln_z_cntr_co_COEF)
                * POWER(EXP(golf120ft_COEF), golf120ft)
                * (CASE
                WHEN gated IS NOT NULL THEN POWER(EXP(gated_COEF), gated)
                ELSE 1.0
                END)
                * (CASE WHEN FldCounty = 0 THEN 1
                                ELSE POWER((1-EXP(FLDCOUNTYRATIO_COEF)),FldCounty_ratio)
                                END)
                * (CASE WHEN Flddb = 0 THEN 1
                                ELSE POWER((1-EXP(FLDDBRATIO_COEF)),Flddb_ratio)
                                END)
                * exp(isnull([value],0))
                ) * asrlotsize
                )
                * trend
                ) /1000,0
                )*1000
                ) * ISNULL(valadj,1.00))
                END
                )
```

Vacant Residential Rural Land Model – Tax Year 2019

```
Parcel Value= CONVERT(numeric(12,0), CASE
                WHEN presentworth = 1 THEN ((
                ROUND(
                (
                (
                (
                EXP(constant) * POWER(EXP(Res_unsubd_coef), Res_unsubd)
                *power(exp(mh_unsubd_coef),mh_unsubd)
                * power(exp(resmh_subd_coef),resmh_subd)
                * POWER(EXP(Dev_subd_coef), Dev_subd)
                * POWER((asrlotsize/landsize_cntr), ln_lotsize_coef)
                * POWER((z/med_elev), ln_z_cntr_co_coef)
                * POWER(EXP(golf120ft_coef), golf120ft)
                * (CASE
                WHEN gated IS NOT NULL THEN POWER(EXP(gated_coef), gated)
                ELSE 1.0
                END)
                * (CASE WHEN FldCounty = 0 THEN 1
                                ELSE POWER((1-EXP(FLDCOUNTYRATIO_COEF)),FldCounty_ratio)
                                END)
                * (CASE WHEN Flddb = 0 THEN 1
                                ELSE POWER((1-EXP(FLDDBRATIO_coef)),Flddb_ratio)
                                END)
                * exp(isnull([value],0))
                ) * asrlotsize
                )
                * CASE
                WHEN mp_PW IS NOT NULL AND presentworth = 1 THEN mp_PW_trend
                WHEN presentworth = 0 AND assr_mp IS NOT NULL  and mp_trend is not null THEN mp_trend
                when presentworth = 0 and mp is null and landsubarea IS NOT NULL and subarea_trend is not null THEN
                subarea_trend
                ELSE 1.0
                END
                ) /1000,0
                )*1000
                ) * ISNULL(valadj,1.00))/POWER(1.10,5)
                ELSE
                ((
                ROUND(
                (
                (
                (
                EXP(constant)
                * POWER(EXP(Res_unsubd_coef), Res_unsubd)
                *power(exp(mh_unsubd_coef),mh_unsubd)
                * power(exp(resmh_subd_coef),resmh_subd)
                * POWER(EXP(Dev_subd_coef), Dev_subd)
                * POWER((asrlotsize/landsize_cntr),ln_lotsize_coef)
                * POWER((z/med_elev), ln_z_cntr_co_coef)
                * POWER(EXP(golf120ft_coef), golf120ft)
                * (CASE
                WHEN gated IS NOT NULL THEN POWER(EXP(gated_coef), gated)
                ELSE 1.0
                END)
                * (CASE WHEN FldCounty = 0 THEN 1
                                ELSE POWER((1-EXP(FLDCOUNTYRATIO_COEF)),FldCounty_ratio)
                                END)
                * (CASE WHEN Flddb = 0 THEN 1
                                ELSE POWER((1-EXP(FLDDBRATIO_COEF)),Flddb_ratio)
                                END)
                * exp(isnull([value],0))
                ) * asrlotsize
                )
                * CASE
                WHEN mp_PW IS NOT NULL AND presentworth = 1 THEN mp_PW_trend
                WHEN presentworth = 0 AND assr_mp IS NOT NULL  and mp_trend is not null THEN mp_trend
                when presentworth = 0 and mp is null and landsubarea IS NOT NULL and subarea_trend is not null THEN
                subarea_trend
                ELSE 1.0
                END
                ) /1000,0
                )*1000
                ) * ISNULL(valadj,1.00))
                END
                )
```

Vacant Residential Rural Land Model – Tax Year 2020 to present

```
Parcel Value= CONVERT(numeric(12,0), CASE
                WHEN presentworth = 1 THEN ((
                ROUND(
                (
                (
                (
                EXP(constant) * POWER(EXP(Res_unsubd_coef), Res_unsubd)
                *power(exp(mh_unsubd_coef),mh_unsubd)
                * power(exp(resmh_subd_coef),resmh_subd)
                * POWER(EXP(Dev_subd_coef), Dev_subd)
                * POWER((asrlotsize/landsize_cntr), ln_lotsize_coef)
                * POWER((z/med_elev), ln_z_cntr_co_coef)
                * POWER(EXP(golf120ft_coef), golf120ft)
                * (CASE
                WHEN gated IS NOT NULL THEN POWER(EXP(gated_coef), gated)
                ELSE 1.0
                END)
                * (CASE WHEN FldCounty = 0 THEN 1
                                ELSE POWER((1-EXP(FLDCOUNTYRATIO_COEF)),FldCounty_ratio)
                                END)
                * (CASE WHEN Flddb = 0 THEN 1
                                ELSE POWER((1-EXP(FLDDBRATIO_coef)),Flddb_ratio)
                                END)
                * exp(isnull([value],0))
                ) * asrlotsize
                )
                * trend
                ) /1000,0
                )*1000
                ) * ISNULL(valadj,1.00))/POWER(1.10,5)
                ELSE
                ((
                ROUND(
                (
                (
                (
                EXP(constant)
                * POWER(EXP(Res_unsubd_coef), Res_unsubd)
                *power(exp(mh_unsubd_coef),mh_unsubd)
                * power(exp(resmh_subd_coef),resmh_subd)
                * POWER(EXP(Dev_subd_coef), Dev_subd)
                * POWER((asrlotsize/landsize_cntr),ln_lotsize_coef)
                * POWER((z/med_elev), ln_z_cntr_co_coef)
                * POWER(EXP(golf120ft_coef), golf120ft)
                * (CASE
                WHEN gated IS NOT NULL THEN POWER(EXP(gated_coef), gated)
                ELSE 1.0
                END)
                * (CASE WHEN FldCounty = 0 THEN 1
                                ELSE POWER((1-EXP(FLDCOUNTYRATIO_COEF)),FldCounty_ratio)
                                END)
                * (CASE WHEN Flddb = 0 THEN 1
                                ELSE POWER((1-EXP(FLDDBRATIO_COEF)),Flddb_ratio)
                                END)
                * exp(isnull([value],0))
                ) * asrlotsize
                )
                * trend
                ) /1000,0
                )*1000
                ) * ISNULL(valadj,1.00))
                END
                )
```

Industrial Model – Tax Years 2016 and 2017

```
Economic Unit Value = convert(numeric(6,0),(constant
                        * size_factor
                        * coverage_factor
                        * PERCENT_GOOD
                        * OFFICE_PCT_ADJ
                        * PCT_METAL_ADJ
                        * height_factor))
                        * totsqft
```

Industrial Model – Tax Year 2018

```
Economic Unit Value = convert(numeric(6,0),(constant
                        * size_factor
                        * coverage_factor
                        * PERCENT_GOOD
                        * OFFICE_PCT_ADJ
                        * PCT_METAL_ADJ
                        * height_factor
                        * whse_condo_adj))
                        * totsqft
```

Industrial Model – Tax Years 2019 to Present

```
Economic Unit Value = convert(dec(6,2),exp(constant)
                        *power(capped_sizefactor50, ln_capped_sizefactor50_coef)
                         *power(coverage,ln_coverage_coef)
                         *power(pct_good,ln_pctgood_coef)
                         *power(exp(office_pct05_coef),office_pct05)
                         *power(exp(pctmetal_coef),pctmetal)
                         *power(exp(pctframe_coef),pctframe)
                         *power(storyht_factor,ln_storyht_factor_coef)
                         *power(exp(whse_condo_coef),whse_condo)
                         *power(exp(indus_nopark_coef),indus_nopark)
                         *power(exp(indus_park_coef),indus_park)
                         *power(exp(distr_terminal_coef),distr_terminal))
                         *totsqft
```

Multi-Family Model – Tax Years 2016 to 2020

```
Economic Unit Value = exp(constant)
                        * power(unitsize_800,unitsize_coef)
                        * power(land_multiplier,lnland_mult_coef)
                        * power(age30_coef,age30)
                        * exp(case when units_3 = 1 then units3_coef else 0 end)
                        * exp(case when units_4_5 = 1 then units45_coef else 0 end)
                        * exp(case when units_6up = 1 then units6up_coef else 0 end)
                        * exp(nbhd_coef)
                        * TotalUnits
                        * isnull(trend_sa,1.000)
```

Multi-Family Model – Tax Years 2021

```
Economic Unit Value = exp(constant)
                        * power(unitsize_800,unitsize_coef)
                        * power(land_multiplier,lnland_mult_coef)
                        * power(exp(age30_coef),age30)
                        * exp(case when units_3 = 1 then units3_coef else 0 end)
                        * exp(case when units_4_5 = 1 then units45_coef else 0 end)
                        * exp(case when units_6up = 1 then units6up_coef else 0 end)
                        * power(exp(ac_coef),ac)
                        * power(exp(evap_coef),evap)
                        * power(exp(coolnone_coef),coolnone)
                        * power(exp(nocovpkg_coef),nocovpkg)
                        * exp(nbhd_coef)
                        * TotalUnits
                        * isnull(trend_sa,1.000)
```

Multi-Family Mixed Model – Tax Years 2016 to 2020

[mixed Economic Units containing Single-family units]

```
Economic Unit Value = (exp(constant)
                        * power(unitsize_800,unitsize_coef)
                        * power(land_multiplier,lnland_mult_coef)
                        * power(age30_coef,age30)
                        * exp(case when units_3 = 1 then units3_coef else 0 end)
                        * exp(case when units_4_5 = 1 then units45_coef else 0 end)
                        * exp(case when units_6up = 1 then units6up_coef else 0 end)
                        * exp(nbhd_coef)
                        * TotalUnits
                        * isnull(trend_sa,1.000))
                        + case when sfr_totsqft > 0 then
                                convert(dec(12,0),(sfr_totsqft * (.75*sfr_valsqft)))
                                else 0 end
```

Multi-Family Mixed Model – Tax Year 2021

[mixed Economic Units containing Single-family units]

```
Economic Unit Value = (exp(constant)
                        * power(unitsize_800,unitsize_coef)
                        * power(land_multiplier,lnland_mult_coef)
                        * power(exp(age30_coef),age30)
                        * exp(case when units_3 = 1 then units3_coef else 0 end)
                        * exp(case when units_4_5 = 1 then units45_coef else 0 end)
                        * exp(case when units_6up = 1 then units6up_coef else 0 end)
                        * power(exp(ac_coef),ac)
                        * power(exp(evap_coef),evap)
                        * power(exp(coolnone_coef),coolnone)
                        * power(exp(nocovpkg_coef),nocovpkg)
                        * exp(nbhd_coef)
                        * TotalUnits
                        * isnull(trend_sa,1.000))
                        + case when sfr_totsqft > 0 then
                                convert(dec(12,0),(sfr_totsqft * (.75*sfr_valsqft)))
                                else 0 end
```

Apartment Model – Tax Year 2017

[Economic Units containing 25 or more apartments]

```
Economic Unit Value = CONVERT(numeric(15,0),((rent_constant
                        + (rent_st_pct_units * st_pct_units)
                        + (rent_bb1_pct_units * bb1_pct_units)
                        + (rent_bb2_pct_units * bb2_pct_units)
                        + (rent_bb3_pct_units * bb3_pct_units)
                        + (rent_bb4_pct_units * bb4_pct_units)
                        + (rent_unitsize_cntr * unitsize_cntr)
                        + (rent_apt_condo * apt_condo)
                        + (rent_elevator * elevator)
                        + (rent_sqrtage50*sqrt_age_50)
                        + (rent_garages_unit*gar_units)
                        + (rent_covspaces_unit*cpt_units)
                        + (rent_mstr_mtr*mstr_mtr)
                        + (rent_mstr_boiler*mstr_boiler)
                        + (rent_utilities_pd_owner*utilities_owner_pd)
                        + (rent_onestory*onestory)
                        + (rent_stories4plus*stories4plus)
                        + (rent_age_restrict*agerestrict)
                        + (rent_restrict*restricted_rent)
                        + (rent_gvt_subsidized*GVT_SUBSIDIZED)
                        + (rent_int_amenities*interior_amenities)
                        + (rent_ext_amenities*exterior_amenities)
                        + (rent_security_gate*security_gate)
                        + (rent_heat_pool*heated_pool)
                        + CASE nlr_ecodist
                                WHEN 1 THEN rent_d1
                                WHEN 2 THEN rent_d2
                                WHEN 3 THEN rent_d3
                                WHEN 4 THEN rent_d4
                                WHEN 5 THEN rent_d5
                                WHEN 6 THEN rent_d6
                                WHEN 7 THEN rent_d7
                                WHEN 8 THEN rent_d8
                                WHEN 9 THEN rent_d9
                                WHEN 10 THEN rent_d10
                                WHEN 11 THEN rent_d11
                                WHEN 12 THEN rent_d12
                                WHEN 13 THEN rent_d13
                                WHEN 14 THEN rent_d14
                                WHEN 15 THEN rent_d15
                                WHEN 16 THEN rent_d16
                                WHEN 17 THEN rent_d17
                                WHEN 18 THEN rent_d18
                                WHEN 19 THEN rent_d19
                                WHEN 20 THEN rent_d20
                                WHEN 30 THEN rent_d30
                                WHEN 40 THEN rent_d40
                                ELSE 0
                                END)
```

```
* (1-
(vac_constant
+ (vac_low_income*low_income)
+ (vac_age30*age_30)
+ CASE x.nlr_ecodist
        WHEN 1 THEN vac_d1
        WHEN 2 THEN vac_d2
        WHEN 3 THEN vac_d3
        WHEN 4 THEN vac_d4
        WHEN 5 THEN vac_d5
        WHEN 6 THEN vac_d6
        WHEN 7 THEN vac_d7
        WHEN 8 THEN vac_d8
        WHEN 9 THEN vac_d9
        WHEN 10 THEN vac_d10
        WHEN 11 THEN vac_d11
        WHEN 12 THEN vac_d12
        WHEN 13 THEN vac_d13
        WHEN 14 THEN vac_d14
        WHEN 15 THEN vac_d15
        WHEN 16 THEN vac_d16
        WHEN 17 THEN vac_d17
        WHEN 18 THEN vac_d18
        WHEN 19 THEN vac_d19
        WHEN 20 THEN vac_d20
        WHEN 30 THEN vac_d30
        WHEN 40 THEN vac_d40
        ELSE 0
        END)
        )
)
*
(egim_constant
* POWER(egim_low_income,low_income)
* POWER(egim_age_p75,age_P75)
* CASE nlr_ecodist
        WHEN 1 THEN egim_d1
        WHEN 2 THEN egim_d2
        WHEN 3 THEN egim_d3
        WHEN 4 THEN egim_d4
        WHEN 5 THEN egim_d5
        WHEN 6 THEN egim_d6
        WHEN 7 THEN egim_d7
        WHEN 8 THEN egim_d8
        WHEN 9 THEN egim_d9
        WHEN 10 THEN egim_d10
        WHEN 11 THEN egim_d11
        WHEN 12 THEN egim_d12
        WHEN 13 THEN egim_d13
        WHEN 14 THEN egim_d14
        WHEN 15 THEN egim_d15
        WHEN 16 THEN egim_d16
        WHEN 17 THEN egim_d17
        WHEN 18 THEN egim_d18
        WHEN 19 THEN egim_d19
        WHEN 20 THEN egim_d20
        WHEN 30 THEN egim_d30
        WHEN 40 THEN egim_d40
        ELSE 0
        END)
        * 12
        * Ratio * CCSUnits)
```

Apartment Model - Tax Year 2018

[Economic Units containing 25 or more apartments]

```
Economic Unit Value = CONVERT(numeric(15,0),
                        ((rent_constant
                        + (rent_st_pct_units * st_pct_units)
                        + (rent_bb1_pct_units * bb1_pct_units)
                        + (rent_bb2_pct_units * bb2_pct_units)
                        + (rent_bb3_pct_units * bb3_pct_units)
                        + (rent_bb4_pct_units * bb4_pct_units)
                        + (rent_st_unitsize_cntr * st_unitsize_cntr)
                        + (rent_bb1_unitsize_cntr * bb1_unitsize_cntr)
                        + (rent_bb2_unitsize_cntr * bb2_unitsize_cntr)
                        + (rent_bb3_unitsize_cntr * bb3_unitsize_cntr)
                        + (rent_bb4_unitsize_cntr * bb4_unitsize_cntr)
                        + (rent_elevator * elevator)
                        + (rent_sqrtage50*sqrt_age_50)
                        + (rent_garages_unit*gar_units)
                        + (rent_covspaces_unit*cpt_units)
                        + (rent_mstr_mtr*mstr_mtr)
                        + (rent_mstr_boiler*mstr_boiler)
                        + (rent_utilities_pd_owner*utilities_owner_pd)
                        + (rent_onestory*onestory)
                        + (rent_stories4plus*stories4plus)
                        + (rent_int_amenities*interior_amenities)
                        + (rent_ext_amenities*exterior_amenities)
                        + (rent_security_gate*security_gate)
                        + (rent_heat_pool*heated_pool)
                        + CASE nlr_ecodist
                                WHEN 1 THEN rent_d1
                                WHEN 2 THEN rent_d2
                                WHEN 3 THEN rent_d3
                                WHEN 4 THEN rent_d4
                                WHEN 5 THEN rent_d5
                                WHEN 6 THEN rent_d6
                                WHEN 7 THEN rent_d7
                                WHEN 8 THEN rent_d8
                                WHEN 9 THEN rent_d9
                                WHEN 10 THEN rent_d10
                                WHEN 11 THEN rent_d11
                                WHEN 12 THEN rent_d12
                                WHEN 13 THEN rent_d13
                                WHEN 14 THEN rent_d14
                                WHEN 15 THEN rent_d15
                                WHEN 16 THEN rent_d16
                                WHEN 17 THEN rent_d17
                                WHEN 18 THEN rent_d18
                                WHEN 19 THEN rent_d19
                                WHEN 20 THEN rent_d20
                                WHEN 30 THEN rent_d30
                                WHEN 40 THEN rent_d40
                                ELSE 0
                                END)
```

```
* (1-
(vac_constant
+ (vac_age30*age_30)
+ CASE nlr_ecodist
        WHEN 1 THEN vac_d1
        WHEN 2 THEN vac_d2
        WHEN 3 THEN vac_d3
        WHEN 4 THEN vac_d4
        WHEN 5 THEN vac_d5
        WHEN 6 THEN vac_d6
        WHEN 7 THEN vac_d7
        WHEN 8 THEN vac_d8
        WHEN 9 THEN vac_d9
        WHEN 10 THEN vac_d10
        WHEN 11 THEN vac_d11
        WHEN 12 THEN vac_d12
        WHEN 13 THEN vac_d13
        WHEN 14 THEN vac_d14
        WHEN 15 THEN vac_d15
        WHEN 16 THEN vac_d16
        WHEN 17 THEN vac_d17
        WHEN 18 THEN vac_d18
        WHEN 19 THEN vac_d19
        WHEN 20 THEN vac_d20
        WHEN 30 THEN vac_d30
        WHEN 40 THEN vac_d40
        ELSE 0
        END)
        )
)
*
(egim_constant
* POWER(egim_age_p75,age_P75)
* CASE nlr_ecodist
        WHEN 1 THEN egim_d1
        WHEN 2 THEN egim_d2
        WHEN 3 THEN egim_d3
        WHEN 4 THEN egim_d4
        WHEN 5 THEN egim_d5
        WHEN 6 THEN egim_d6
        WHEN 7 THEN egim_d7
        WHEN 8 THEN egim_d8
        WHEN 9 THEN egim_d9
        WHEN 10 THEN egim_d10
        WHEN 11 THEN egim_d11
        WHEN 12 THEN egim_d12
        WHEN 13 THEN egim_d13
        WHEN 14 THEN egim_d14
        WHEN 15 THEN egim_d15
        WHEN 16 THEN egim_d16
        WHEN 17 THEN egim_d17
        WHEN 18 THEN egim_d18
        WHEN 19 THEN egim_d19
        WHEN 20 THEN egim_d20
        WHEN 30 THEN egim_d30
        WHEN 40 THEN egim_d40
        ELSE 0
        END)
        * 12
        * Ratio * CCSUnits
)
```

Apartment Model - Tax Year 2019

[Economic Units containing 25 or more apartments]

```
Economic Unit Value = CONVERT(numeric(15,0),
        (((rent_constant
                + (rent_st_pct_units * st_pct_units)
                + (rent_bb1_pct_units * bb1_pct_units)
                + (rent_bb2_pct_units * bb2_pct_units)
                + (rent_bb3_pct_units * bb3_pct_units)
                + (rent_bb4_pct_units * bb4_pct_units)
                + (rent_st_unitsize_cntr * st_unitsize_cntr)
                + (rent_bb1_unitsize_cntr * bb1_unitsize_cntr)
                + (rent_bb2_unitsize_cntr * bb2_unitsize_cntr)
                + (rent_bb3_unitsize_cntr * bb3_unitsize_cntr)
                + (rent_bb4_unitsize_cntr * bb4_unitsize_cntr)
                + (rent_elevator * elevator)
                + (rent_sqrtage50* sqrt_age_50)
                + (rent_garages_unit*gar_units)
                + (rent_covspaces_unit*cpt_units)
                + (rent_mstr_mtr*mstr_mtr)
                + (rent_mstr_boiler*mstr_boiler)
                + (rent_onestory*onestory)
                + (rent_stories4plus*stories4plus)
                + (rent_int_amenities*interior_amenities)
                + (rent_ext_amenities*exterior_amenities)
                + (rent_security_gate*security_gate)
                + (rent_heat_pool*heated_pool)
                + CASE nlr_ecodist
                        WHEN 1 THEN rent_d1
                        WHEN 2 THEN rent_d2
                        WHEN 3 THEN rent_d3
                        WHEN 4 THEN rent_d4
                        WHEN 5 THEN rent_d5
                        WHEN 6 THEN rent_d6
                        WHEN 7 THEN rent_d7
                        WHEN 8 THEN rent_d8
                        WHEN 9 THEN rent_d9
                        WHEN 10 THEN rent_d10
                        WHEN 11 THEN rent_d11
                        WHEN 12 THEN rent_d12
                        WHEN 13 THEN rent_d13
                        WHEN 14 THEN rent_d14
                        WHEN 15 THEN rent_d15
                        WHEN 16 THEN rent_d16
                        WHEN 17 THEN rent_d17
                        WHEN 18 THEN rent_d18
                        WHEN 19 THEN rent_d19
                        WHEN 20 THEN rent_d20
                        WHEN 30 THEN rent_d30
                        WHEN 40 THEN rent_d40
                        ELSE 0
                        END)
```

```
* (1-(vac_constant
+ (vac_age30*age_30)
+ CASE nlr_ecodist
        WHEN 1 THEN vac_d1
        WHEN 2 THEN vac_d2
        WHEN 3 THEN vac_d3
        WHEN 4 THEN vac_d4
        WHEN 5 THEN vac_d5
        WHEN 6 THEN vac_d6
        WHEN 7 THEN vac_d7
        WHEN 8 THEN vac_d8
        WHEN 9 THEN vac_d9
        WHEN 10 THEN vac_d10
        WHEN 11 THEN vac_d11
        WHEN 12 THEN vac_d12
        WHEN 13 THEN vac_d13
        WHEN 14 THEN vac_d14
        WHEN 15 THEN vac_d15
        WHEN 16 THEN vac_d16
        WHEN 17 THEN vac_d17
        WHEN 18 THEN vac_d18
        WHEN 19 THEN vac_d19
        WHEN 20 THEN vac_d20
        WHEN 30 THEN vac_d30
        WHEN 40 THEN vac_d40
        ELSE 0
        END)
)
)
*
(exp(egim_constant)
* POWER(exp(egim_age_p75),age_P75)
* CASE nlr_ecodist
        WHEN 1 THEN exp(egim_d1)
        WHEN 2 THEN exp(egim_d2)
        WHEN 3 THEN exp(egim_d3)
        WHEN 4 THEN exp(egim_d4)
        WHEN 5 THEN exp(egim_d5)
        WHEN 6 THEN exp(egim_d6)
        WHEN 7 THEN exp(egim_d7)
        WHEN 8 THEN exp(egim_d8)
        WHEN 9 THEN exp(egim_d9)
        WHEN 10 THEN exp(egim_d10)
        WHEN 11 THEN exp(egim_d11)
        WHEN 12 THEN exp(egim_d12)
        WHEN 13 THEN exp(egim_d13)
        WHEN 14 THEN exp(egim_d14)
        WHEN 15 THEN exp(egim_d15)
        WHEN 16 THEN exp(egim_d16)
        WHEN 17 THEN exp(egim_d17)
        WHEN 18 THEN exp(egim_d18)
        WHEN 19 THEN exp(egim_d19)
        WHEN 20 THEN exp(egim_d20)
        WHEN 30 THEN exp(egim_d30)
        WHEN 40 THEN exp(egim_d40)
        ELSE 0
        END)
*12
* Ratio)
* ccsunits)
```

Apartment Model - Tax Year 2020 to present

[Economic Units containing 25 or more apartments]

```
Economic Unit Value = CONVERT(numeric(15,0),
        (((rent_constant
                + (rent_st_pct_units * st_pct_units)
                + (rent_bb1_pct_units * bb1_pct_units)
                + (rent_bb2_pct_units * bb2_pct_units)
                + (rent_bb3_pct_units * bb3_pct_units)
                + (rent_bb4_pct_units * bb4_pct_units)
                + (rent_st_unitsize_cntr * st_unitsize_cntr)
                + (rent_bb1_unitsize_cntr * bb1_unitsize_cntr)
                + (rent_bb2_unitsize_cntr * bb2_unitsize_cntr)
                + (rent_bb3_unitsize_cntr * bb3_unitsize_cntr)
                + (rent_bb4_unitsize_cntr * bb4_unitsize_cntr)
                + (rent_apt_home * apt_home)
                + (rent_elevator * elevator)
                + (rent_sqrtage50* sqrt_age_50)
                + (rent_garages_unit*gar_units)
                + (rent_covspaces_unit*cpt_units)
                + (rent_mstr_mtr*mstr_mtr)
                + (rent_mstr_boiler*mstr_boiler)
                + (rent_onestory*onestory)
                + (rent_stories4plus*stories4plus)
                + (rent_int_amenities*interior_amenities)
                + (rent_ext_amenities*exterior_amenities)
                + (rent_security_gate*security_gate)
                + (rent_heat_pool*heated_pool)
                + CASE nlr_ecodist
                        WHEN 1 THEN rent_d1
                        WHEN 2 THEN rent_d2
                        WHEN 3 THEN rent_d3
                        WHEN 4 THEN rent_d4
                        WHEN 5 THEN rent_d5
                        WHEN 6 THEN rent_d6
                        WHEN 7 THEN rent_d7
                        WHEN 8 THEN rent_d8
                        WHEN 9 THEN rent_d9
                        WHEN 10 THEN rent_d10
                        WHEN 11 THEN rent_d11
                        WHEN 12 THEN rent_d12
                        WHEN 13 THEN rent_d13
                        WHEN 14 THEN rent_d14
                        WHEN 15 THEN rent_d15
                        WHEN 16 THEN rent_d16
                        WHEN 17 THEN rent_d17
                        WHEN 18 THEN rent_d18
                        WHEN 19 THEN rent_d19
                        WHEN 20 THEN rent_d20
                        WHEN 30 THEN rent_d30
                        WHEN 40 THEN rent_d40
                        ELSE 0
                        END)
```

```
* (1-(vac_constant
+ (vac_age30*age_30)
+ CASE nlr_ecodist
        WHEN 1 THEN vac_d1
        WHEN 2 THEN vac_d2
        WHEN 3 THEN vac_d3
        WHEN 4 THEN vac_d4
        WHEN 5 THEN vac_d5
        WHEN 6 THEN vac_d6
        WHEN 7 THEN vac_d7
        WHEN 8 THEN vac_d8
        WHEN 9 THEN vac_d9
        WHEN 10 THEN vac_d10
        WHEN 11 THEN vac_d11
        WHEN 12 THEN vac_d12
        WHEN 13 THEN vac_d13
        WHEN 14 THEN vac_d14
        WHEN 15 THEN vac_d15
        WHEN 16 THEN vac_d16
        WHEN 17 THEN vac_d17
        WHEN 18 THEN vac_d18
        WHEN 19 THEN vac_d19
        WHEN 20 THEN vac_d20
        WHEN 30 THEN vac_d30
        WHEN 40 THEN vac_d40
        ELSE 0
        END)
)
)
*
(exp(egim_constant)
* POWER(exp(egim_age_p75),age_P75)
* CASE nlr_ecodist
        WHEN 1 THEN exp(egim_d1)
        WHEN 2 THEN exp(egim_d2)
        WHEN 3 THEN exp(egim_d3)
        WHEN 4 THEN exp(egim_d4)
        WHEN 5 THEN exp(egim_d5)
        WHEN 6 THEN exp(egim_d6)
        WHEN 7 THEN exp(egim_d7)
        WHEN 8 THEN exp(egim_d8)
        WHEN 9 THEN exp(egim_d9)
        WHEN 10 THEN exp(egim_d10)
        WHEN 11 THEN exp(egim_d11)
        WHEN 12 THEN exp(egim_d12)
        WHEN 13 THEN exp(egim_d13)
        WHEN 14 THEN exp(egim_d14)
        WHEN 15 THEN exp(egim_d15)
        WHEN 16 THEN exp(egim_d16)
        WHEN 17 THEN exp(egim_d17)
        WHEN 18 THEN exp(egim_d18)
        WHEN 19 THEN exp(egim_d19)
        WHEN 20 THEN exp(egim_d20)
        WHEN 30 THEN exp(egim_d30)
        WHEN 40 THEN exp(egim_d40)
        ELSE 0
        END)
*12
* Ratio)
* ccsunits)
```

Office Medical Model – Tax Year 2020 to present

```
Economic Unit Value = convert(dec(6,2),exp(constant)
              *power(capped_sizefactor15,ln_size2_capped_coef)
              *power(capped_coverage,ln_covcapped20_coef)
              *power(pct_good,ln_pctgood_coef)
              *power(exp(MC_BIN_coef),MC_BIN)
              *power(exp(MMT_BIN_coef),MMT_BIN)
              *power(exp(MST_BIN_coef),MST_BIN)
              *power(exp(OC_BIN_coef),OC_BIN)
              *power(exp(OSF_BIN_coef),OSF_BIN)
              *power(exp(OST_BIN_coef),OST_BIN)
              *power(exp(OMT_BIN_coef),OMT_BIN)
              *power(exp(OFFICESA01_coef),OFFICESA01)
              *power(exp(OFFICESA02_coef),OFFICESA02)
              *power(exp(OFFICESA03_coef),OFFICESA03)
              *power(exp(OFFICESA04_coef),OFFICESA04)
              *power(exp(OFFICESA05_coef),OFFICESA05)
              *power(exp(OFFICESA06_coef),OFFICESA06)
              *power(exp(OFFICESA07_coef),OFFICESA07)
              *power(exp(OFFICESA08_coef),OFFICESA08)
              *power(exp(OFFICESA09_coef),OFFICESA09))
              *totsqft
```