```
* Encoding: UTF-8.
* Variable establishment is from analysis of Inventory and Descriptive Syntax results.
*Setup Variable Labels and Values.
variable labels parcel_id 'Parcel Number'.
variable labels zone 'Zoning'.
variable labels nbhd 'Neighborhood'.
variable labels parcel_size 'Parcel Size'.
variable labels num_stories 'Residential Stories'.
variable labels quality 'Quality Code'.
variable labels Quality_Grade 'Quality Grade'.
variable labels condition 'Condition Code'.
variable labels Condition_Grade 'Condition Grade'.
variable labels Age 'Age of Residence'.
variable labels Eff_Age 'Effective Age of Residence'.
variable labels living_area 'Total Living Area'.
variable labels MainFloor 'Main Floor Area'.
variable labels UpperFloor 'Upper Floor Area'.
variable labels FinBSMT 'Finished Basement Area'.
variable labels UnFinBSMT 'Unfinished Basement Area'.
variable labels blt_gar 'Built In Garage Area'.
variable labels att_gar 'Attached Garage Area'.
variable labels PlumbingFixtures 'Plumbing Fixtures'.
variable labels stove 'Wood Stoves'.
variable labels fireplace 'Masonry Fireplaces'.
variable labels Prefab_FP 'Prefabricated Fireplaces'.
variable labels roofing 'Roofing Material'.
variable labels ext_wall_typ 'Exterior Wall Type'.
variable labels masonry_trim 'Trim Material'.
variable labels ac 'Air Conditioning'.
variable labels num_bedrooms 'Number of Bedrooms'.
variable labels full_bath 'Full Baths'.
variable labels half_bath 'Half Baths'.
variable labels detatched_value 'Detached Value'.
variable labels land_view 'Land View'.
variable labels flood_plain 'Flood Plain'.
variable labels lot_shape 'Lot Shape'.
variable labels topography 'Topography'.
variable labels gas 'Natural Gas Hookup'.
variable labels street_finsh 'Street Finish'.
variable labels traffic 'Traffic Density'.
variable labels curbs 'Curbing'.
variable labels side_walk 'Sidewalk'.
variable labels assessed_land 'Previous Land Value'.
variable labels assessed_imp 'Previous Imp. Value'.
variable labels assessed_total 'Previous Assessed Total'.
variable labels sale_date 'Sale Date'.
variable labels sale_price 'Sale Price'.

add value labels Quality_Grade 1 'Low-Cost' 2 'Fair -' 3 'Fair' 4 'Fair +' 5 'Average' 6 'Average +' 7 'Good' 8 'Good
Plus' 9 'Very Good' 10 'Very Good +' 11 'Excellent' 12 ' Excellent +' 13 'Exceptional'.
add value labels Condition_Grade 1 'Very Poor' 2 'Poor' 3 'Fair' 4 'Average' 5 'Good' 6 'Very Good' 7 'Excellent'.
add value labels zone 'R1' 'Single-Family Residential' 'R2' 'Single-Family and 2 Family Residential' 'R3' 'Multifamily
Residential'.
```

```
add value labels roofing 'AC' 'Arch. Comp' 'CL' 'Clay Tile' 'CP' 'Comp' 'MT' 'Metal' 'OT' 'Mobile Metal' 'RR' 'Roll Roof'
'SH' 'Shake' 'SL' 'Slate' 'TG' 'Tar-Gravel' 'WS' 'Wood Shingle'.
add value labels ext_wall_typ 'AL' 'ALUMINUM' 'BV' 'BRICK-VENEER' 'CB' 'CONCRT-BLOCK' 'DR' 'DRYVIT' 'HB'
'HARDBOARD' 'LG' 'LOG' 'SH' 'SHINGLES' 'SN' 'STONE' 'SS' 'TEX1-11/PWD' 'ST' 'STUCCO' 'VN' 'VINYL' 'WD'
'WOOD-SIDING'.
add value labels masonry_trim 'B' 'Brick' 'N' 'None' 'S' 'Stone'.
add value labels ac 0 'No' 1 'Yes'.
add value labels land_view 'NV' 'No View'.
add value labels flood_plain 'N' 'None'.
add value labels lot_shape 'IR' 'Irregular' 'RC' 'Rectangular' 'SQ' 'Square' 'TF' 'Triangle Front' 'TR' Triangle Rear'.
add value labels topography 'L' 'Level'.
add value labels gas '0' 'No' '1' 'Yes'.
add value labels street_finsh 'P' 'Paved'.
add value labels traffic 'L' 'Light'.
add value labels curbs '1' 'Yes' '0' 'No'.
add value labels side_walk '1' 'Yes' '0' 'No'.

*select if property_use_category = 'Residential'.
select if parcel_size <= 5.0.

*Remove Pre L&I.
Select if Age < 48.

COMPUTE SPPSF=sale_price/living_area.
variable labels SPPSF 'Sale Price Per Square Foot'.
GRAPH/HISTOGRAM=sale_price.
GRAPH/HISTOGRAM=living_area.
GRAPH/HISTOGRAM=SPPSF.
GRAPH/HISTOGRAM=parcel_size.

GRAPH  /SCATTERPLOT(BIVAR)=living_area WITH sppsf  /MISSING=LISTWISE.
GRAPH  /SCATTERPLOT(BIVAR)=living_area WITH sale_price  /MISSING=LISTWISE.

COMPUTE total_area1 = living_area.
COMPUTE total_area2 = 0.
if (living_area >= 1250) total_area1 = 1250.
if (living_area >= 1250) total_area2 = living_area - 1250.

GRAPH  /SCATTERPLOT(BIVAR)=living_area WITH sppsf by condition_grade /MISSING=LISTWISE.
GRAPH  /SCATTERPLOT(BIVAR)=living_area WITH sale_price by condition_grade /MISSING=LISTWISE.

GRAPH  /SCATTERPLOT(BIVAR)=living_area WITH sppsf by quality_grade /MISSING=LISTWISE.
GRAPH  /SCATTERPLOT(BIVAR)=living_area WITH sale_price by quality_grade /MISSING=LISTWISE.

compute sqft_range = 0.
if (living_area >1000 and living_area <=1500) sqft_range = 1.
if (living_area >1500 and living_area <=2000) sqft_range = 2.
if (living_area >2000 and living_area <=2500) sqft_range = 3.
if (living_area >2500 and living_area <=3000) sqft_range = 4.
if (living_area >3000 and living_area <=3500) sqft_range = 5.
if (living_area >3500 and living_area <=4000) sqft_range = 6.
if (living_area >4000) sqft_range = 7.
```

```
variable labels sqft_range 'SQFT Range'.
add value labels sqft_range 0 '0 thru 1,000' 1 '1,000 thru 1,500' 2 '1,500 thru 2,000' 3 '2,000 thru 2,500' 4 '2,500 thru
 3,000' 5 '3,000 thru 3,500' 6 '3,500 thru 4,000' 7 '4,000 and up'.

compute age_range = 0.
if (age >10 and age <= 20) age_range = 1.
if (age >20 and age <= 30) age_range = 2.
if (age >30 and age <= 40) age_range = 3.
if (age >40 and age <= 50) age_range = 4.
if (age >50 and age <= 60) age_range = 5.
if (age >60 and age <= 70) age_range = 6.
if (age >70 and age <= 80) age_range = 7.
if (age >80 and age <= 90) age_range = 8.
if (age >90 and age <= 100) age_range = 9.
if (age >100 and age <=110) age_range = 10.
if (age >110) age_range = 11.

variable labels age_range 'Age Range'.
add value labels age_range 0 'Less than 10' 1 '10 thru 19' 2 '20 thru 29' 3 '30 thru 39' 4 '40 thru 49'  5 '50 thru 59' 6
'60 thru 69' 7 '70 thru 79' 8 '80 thru 89' 9 '90 thru 99' 10 '100 thru 109' 11 '110 and up'.

compute ln_living_area = ln(living_area).
compute ln_parcel_size = ln(parcel_size).

COMPUTE SAR=assessed_total/sale_price.
variable labels SAR 'Sales Assessment Ratio'.

EXAMINE sale_price by nbhd/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
MEANS TABLES=sale_price BY nbhd  /CELLS=MEAN MEDIAN COUNT STDDEV.
EXAMINE SPPSF by nbhd/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
MEANS TABLES=SPPSF BY nbhd  /CELLS=MEAN MEDIAN COUNT STDDEV.

STRING  model_nbhd (A4).
COMPUTE model_nbhd=nbhd.
if (nbhd = 'ZHLA') model_nbhd = 'ZH'.
if (nbhd = 'BU') model_nbhd = 'ZH'.
if (nbhd = 'NAKH') model_nbhd = 'NA'.
if (nbhd = 'OL') model_nbhd = 'SS'.

if (nbhd = 'PK') model_nbhd = '151'.
if (nbhd='WS') model_nbhd = '151'.
if (nbhd = 'GL') model_nbhd = 'GLR'.

if (nbhd = '262') model_nbhd = '261'.
if (nbhd = '372F') model_nbhd = '372S'.
if (nbhd = '372N') model_nbhd = '372'.

if (nbhd='TH1') model_nbhd = 'TH'.
if (nbhd='THCC') model_nbhd = 'TH'.
if (nbhd = 'YTL') model_nbhd = '151'.
if (nbhd = '152') model_nbhd = '151'.

if(nbhd = '485N') model_nbhd = '485'.
```

```
*Remove High and low Ratios (according to DOR >1.75 and <0.25.
compute DOROutlier = 1.
if (SAR > 1.75) DOROutlier = 0.
if (SAR < 0.25) DOROutlier = 0.

* Ratio Statistics prior to time adjustment.
RATIO STATISTICS assessed_total WITH sale_price/PRINT=COD MEAN MEDIAN PRB PRD WGTMEAN N.

RATIO STATISTICS assessed_total WITH sale_price BY model_nbhd (ASCENDING)/PRINT=COD MEAN MEDIAN
PRB PRD WGTMEAN N.
RATIO STATISTICS assessed_total WITH sale_price BY quality_grade (ASCENDING)/PRINT=COD MEAN MEDIAN
 PRB PRD WGTMEAN N.
RATIO STATISTICS assessed_total WITH sale_price BY condition_grade (ASCENDING)/PRINT=COD MEAN
MEDIAN PRB PRD WGTMEAN N.
RATIO STATISTICS assessed_total WITH sale_price BY sqft_range (ASCENDING)/PRINT=COD MEAN MEDIAN
PRB PRD WGTMEAN N.
RATIO STATISTICS assessed_total WITH sale_price BY age_range (ASCENDING)/PRINT=COD MEAN MEDIAN
PRB PRD WGTMEAN N.

compute smonth = xdate.month(sale_date).
compute syear = 2025 - xdate.year(sale_date).

COMPUTE MONTHS=SMONTH.
variable labels Months 'Months Prior to Apppraisal Date'.
if (SYEAR=2) MONTHS=SMONTH+12.
If (SYEAR=3) MONTHS=SMONTH+24.
FORMATS MONTHS(F4.0).
MEANS MONTHS BY SYEAR BY SMONTH.

GRAPH/SCATTER=months WITH SAR/Title="Sales Assessment Ratio Change Over Time".
GRAPH/SCATTER=sale_date WITH SAR/Title="Sales Assessment Ratio Change Over Time".
GRAPH/HISTOGRAM=SAR.

COMPUTE LN_SAR=LN(SAR).
variable labels LN_SAR 'Natural Log of SAR'.
GRAPH/SCATTER=MONTHS WITH LN_SAR/Title="Natural Log of Sales Assessment Ratio Change Over Time".
GRAPH/SCATTER=sale_date WITH LN_SAR/Title="Natural Log of Sales Assessment Ratio Change Over Time".

REGRESSION
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10) TOLERANCE(.0001)
  /DEPENDENT LN_SAR
  /METHOD=STEPWISE MONTHS.

COMPUTE END1=11.
COMPUTE END2=18.
COMPUTE END3=28.
COMPUTE MONTHS1=MONTHS.
IF (MONTHS>END1) MONTHS1=END1.
COMPUTE MONTHS2=0.
IF (MONTHS>END1) MONTHS2=MONTHS-END1.
IF (MONTHS>END2) MONTHS2=END2-END1.
```

```
COMPUTE MONTHS3=0.
IF (MONTHS>END2) MONTHS3=MONTHS-END2.
IF (MONTHS>END3) MONTHS3=END3-END2.
COMPUTE MONTHS4=0.
IF (MONTHS>END3) MONTHS4=MONTHS-END3.
MEANS MONTHS1 MONTHS2 MONTHS3 MONTHS4 BY MONTHS/CELLS=MEAN.

REGRESSION
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10) TOLERANCE(.0001)
  /DEPENDENT LN_SAR
  /METHOD=STEPWISE MONTHS1 MONTHS2 MONTHS3 MONTHS4.

COMPUTE r1 = 0.000.
COMPUTE r2 = 0.01164903.
COMPUTE r3 = 0.000.
COMPUTE r4 = 0.000.
formats r1(f6.3).
formats r2(f6.3).
formats r3(f6.3).
formats r4(f6.3).
COMPUTE PRICE_INDEX=(1+r1)**MONTHS1 * (1+r2)**MONTHS2* (1+r3)**MONTHS3* (1+r4)**MONTHS4.
GRAPH/SCATTER=MONTHS WITH PRICE_INDEX.

MEANS PRICE_INDEX BY MONTHS.
COMPUTE END_INDEX=1.000.
COMPUTE TAF=PRICE_INDEX/END_INDEX.
variable labels TAF 'Time Adjustment Factor'.
GRAPH/SCATTER=MONTHS WITH TAF.
MEANS TAF BY MONTHS.

*TIME-ADJUSTED PRICES.
COMPUTE TASP=sale_price*TAF.
variable labels TASP 'Time Adjusted Sales Price'.
DES sale_price TASP TAF.
COMPUTE TASAR=assessed_total/TASP.
variable labels TASAR 'Time Adjusted Sales Assessment Ratio'.

GRAPH/SCATTER=MONTHS WITH TASAR.

GRAPH/HISTOGRAM=sale_price.
GRAPH/HISTOGRAM=TASP.
GRAPH/HISTOGRAM=SAR.
GRAPH/HISTOGRAM=TASAR.

* Ratio Outputs after time Adjustment..
RATIO STATISTICS assessed_total WITH TASP/PRINT=COD MEAN MEDIAN PRB PRD WGTMEAN N.

RATIO STATISTICS assessed_total WITH TASP BY model_nbhd (ASCENDING)/PRINT=COD MEAN MEDIAN PRB
PRD WGTMEAN N.
RATIO STATISTICS assessed_total WITH TASP BY quality_grade (ASCENDING)/PRINT=COD MEAN MEDIAN PRE
 PRD WGTMEAN N.
RATIO STATISTICS assessed_total WITH TASP BY condition_grade (ASCENDING)/PRINT=COD MEAN MEDIAN
```

```
PRB PRD WGTMEAN N.
RATIO STATISTICS assessed_total WITH TASP BY sqft_range (ASCENDING)/PRINT=COD MEAN MEDIAN PRB
PRD WGTMEAN N.
RATIO STATISTICS assessed_total WITH TASP BY age_range (ASCENDING)/PRINT=COD MEAN MEDIAN PRB
PRD WGTMEAN N.

COMPUTE TASPPSF= (TASP - market_land - detatched_value)/living_area.
variable labels TASPPSF 'Time Adjusted Sales Price Per Sqft'.
GRAPH/HIST=TASPPSF.
GRAPH/SCATTER=living_area WITH TASPPSF.

*LIVING AREA AND quality_grade.
GRAPH/SCATTER=living_area WITH TASPPSF by quality_grade.
EXAMINE TASPPSF BY quality_grade/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
MEANS TABLES=living_area BY quality_grade  /CELLS=MEAN MEDIAN COUNT STDDEV.
MEANS TABLES=TASPPSF BY quality_grade  /CELLS=MEAN MEDIAN COUNT STDDEV.
*Grade 6 appears to be median grade and should not be included in analysis.
COMPUTE GRADE1_SF=0.
IF (quality_grade=1) GRADE1_SF = living_area.
COMPUTE GRADE2_SF=0.
IF (quality_grade=2) GRADE2_SF = living_area.
COMPUTE GRADE3_SF=0.
IF (quality_grade=3) GRADE3_SF = living_area.
COMPUTE GRADE4_SF=0.
IF (quality_grade=4) GRADE4_SF = living_area.
COMPUTE GRADE5_SF=0.
IF (quality_grade=5) GRADE5_SF = living_area.
COMPUTE GRADE6_SF=0.
IF (quality_grade=6) GRADE6_SF = living_area./**BASE QUALITY.
COMPUTE GRADE7_SF=0.
IF (quality_grade=7) GRADE7_SF = living_area.
COMPUTE GRADE8_SF=0.
IF (quality_grade=8) GRADE8_SF = living_area.
COMPUTE GRADE9_SF=0.
IF (quality_grade=9) GRADE9_SF = living_area.
COMPUTE GRADE10_SF=0.
IF (quality_grade=10) GRADE10_SF = living_area.
COMPUTE GRADE11_SF=0.
IF (quality_grade=11) GRADE11_SF = living_area.
COMPUTE GRADE12_SF=0.
IF (quality_grade=12) GRADE12_SF = living_area.
COMPUTE GRADE13_SF=0.
IF (quality_grade=13) GRADE13_SF = living_area.

MEANS GRADE1_SF GRADE2_SF GRADE3_SF GRADE4_SF GRADE5_SF GRADE6_SF GRADE7_SF
GRADE8_SF GRADE9_SF GRADE10_SF GRADE11_SF GRADE12_SF GRADE13_SF BY
quality_grade/CELLS=MEAN.

*LIVING AREA AND condition_grade.
GRAPH/SCATTER=living_area WITH TASPPSF by condition_grade.
EXAMINE TASPPSF BY condition_grade/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
MEANS TABLES=living_area BY condition_grade  /CELLS=MEAN MEDIAN COUNT STDDEV.
MEANS TABLES=TASP BY condition_grade  /CELLS=MEAN MEDIAN COUNT STDDEV.
```

```
MEANS TABLES=TASPPSF BY condition_grade  /CELLS=MEAN MEDIAN COUNT STDDEV.
* Choosing Good (CONDITION GRADE 5) as default condition_grade by living area and TASP.
COMPUTE COND1_SF=0.
IF (condition_grade=1) COND1_SF = living_area.
COMPUTE COND2_SF=0.
IF (condition_grade=2) COND2_SF = living_area.
COMPUTE COND3_SF=0.
IF (condition_grade=3) COND3_SF = living_area.
COMPUTE COND4_SF=0.
IF (condition_grade=4) COND4_SF = living_area./** BASE CONDITION.
COMPUTE COND5_SF=0.
IF (condition_grade=5) COND5_SF = living_area.
COMPUTE COND6_SF=0.
IF (condition_grade=6) COND6_SF = living_area.
COMPUTE COND7_SF=0.
IF (condition_grade=7) COND7_SF = living_area.
MEANS COND1_SF COND2_SF COND3_SF COND4_SF COND5_SF COND6_SF COND7_SF by
condition_grade/CELLS=MEAN.

*Create binaries for exterior wall type.
EXAMINE tasp BY ext_wall_typ/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
MEANS TABLES=TASPPSF By ext_wall_typ /CELLS=MEAN MEDIAN COUNT STDDEV.
compute ext_wall_AL = 0.
compute ext_wall_BV = 0.
compute ext_wall_CB = 0.
compute ext_wall_DR = 0.
compute ext_wall_HB = 0.
compute ext_wall_LG = 0.
compute ext_wall_SH = 0.
compute ext_wall_SN = 0.
compute ext_wall_SS = 0.
compute ext_wall_ST = 0.
compute ext_wall_MT = 0.
compute ext_wall_VN = 0.
compute ext_wall_WD = 0.
if (ext_wall_typ = 'AL') ext_wall_AL = 1.
if (ext_wall_typ = 'BV') ext_wall_BV = 1.
if (ext_wall_typ = 'CB') ext_wall_CB = 1.
if (ext_wall_typ = 'DR') ext_wall_DR = 1.
if (ext_wall_typ = 'HB') ext_wall_HB = 1.
if (ext_wall_typ = 'LG') ext_wall_LG = 1.
if (ext_wall_typ = 'SH') ext_wall_SH = 1.
if (ext_wall_typ = 'SN') ext_wall_SN = 1.
if (ext_wall_typ = 'SS') ext_wall_SS = 1.
if (ext_wall_typ = 'ST') ext_wall_ST = 1.
if (ext_wall_typ = 'MT') ext_wall_MT = 1.
if (ext_wall_typ = 'VN') ext_wall_VN = 1.
if (ext_wall_typ = 'WD') ext_wall_WD = 1.

*Create binaries for roofing.
EXAMINE tasppsf BY roofing/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
compute roofing_AC = 0.
compute roofing_CL = 0.
```

```
compute roofing_CP = 0.
compute roofing_MT = 0.
compute roofing_OT = 0.
compute roofing_RR = 0.
compute roofing_SH = 0.
compute roofing_SL = 0.
compute roofing_TG = 0.
compute roofing_WS = 0.
compute roofing_E = 0.
if(roofing='AC') roofing_AC = living_area + att_gar.
if(roofing='CL') roofing_CL = living_area + att_gar.
if(roofing='CP') roofing_CP = living_area + att_gar.
if(roofing='C') roofing_CP = living_area + att_gar.
if(roofing='MT') roofing_MT = living_area + att_gar.
if(roofing='OT') roofing_OT = living_area + att_gar.
if(roofing='M') roofing_OT = living_area + att_gar.
if(roofing='RR') roofing_RR = living_area + att_gar.
if(roofing='SH') roofing_SH = living_area + att_gar.
if(roofing='SL') roofing_SL = living_area + att_gar.
if(roofing='TG') roofing_TG = living_area + att_gar.
if(roofing='WS') roofing_WS = living_area + att_gar.
if(roofing='E') roofing_E = living_area + att_gar.

*Create binaries for masonry_trim.
EXAMINE tasp BY masonry_trim/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
COMPUTE masonry_brick=0.
IF (masonry_trim="B") masonry_brick=1.
COMPUTE masonry_stone=0.
IF (masonry_trim="S") masonry_stone=1.
COMPUTE masonry_none=0.
IF (masonry_trim="N") masonry_none=1.
IF (masonry_trim = "") masonry_none = 1.

EXAMINE tasp BY num_stories/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
EXAMINE TASPPSF BY num_stories/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.

GRAPH/SCATTER=PlumbingFixtures with TASP.
EXAMINE tasp BY full_bath/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
EXAMINE tasp BY num_bedrooms/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.

*/ Create nbhd variables with binary to get a flat $ adjustment from regression.
EXAMINE tasp by model_nbhd/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
MEANS TABLES=TASP BY model_nbhd  /CELLS=MEAN MEDIAN COUNT STDDEV.
EXAMINE TASPPSF by model_nbhd/PLOT=BOXPLOT/STAT=NONE/NOTOTAL.
MEANS TABLES=TASPPSF BY model_nbhd  /CELLS=MEAN MEDIAN COUNT STDDEV.

*/ Create nbhd variables with living area to get a $ per sqft adjustment from regression.
compute nbhd_151_sqft = 0.
compute nbhd_153_sqft = 0.
compute nbhd_260_sqft = 0.
compute nbhd_261_sqft = 0.
compute nbhd_263_sqft = 0.
compute nbhd_370_sqft = 0.
```

```
compute nbhd_371G_sqft = 0.
compute nbhd_371R_sqft = 0.
compute nbhd_371S_sqft = 0.
compute nbhd_372_sqft = 0.
compute nbhd_372S_sqft = 0.
compute nbhd_481_sqft = 0.
compute nbhd_482_sqft = 0.
compute nbhd_483_sqft = 0.
compute nbhd_484_sqft = 0.
compute nbhd_485_sqft = 0.
compute nbhd_485O_sqft = 0.
compute nbhd_485P_sqft = 0.
compute nbhd_GLR_sqft = 0.
compute nbhd_GLV_sqft = 0.
compute nbhd_GR_sqft = 0.
compute nbhd_GV_sqft = 0.
compute nbhd_HA_sqft = 0.
compute nbhd_MB_sqft = 0.
compute nbhd_MX_sqft = 0.
compute nbhd_NA_sqft = 0.
compute nbhd_SL_sqft = 0.
compute nbhd_SLN_sqft = 0.
compute nbhd_SLV_sqft = 0.
compute nbhd_SS_sqft = 0.
compute nbhd_SSSH_sqft = 0.
compute nbhd_TC_sqft = 0.
compute nbhd_TH_sqft = 0.
compute nbhd_TP_sqft = 0.
compute nbhd_UG_sqft = 0.
compute nbhd_UGR_sqft = 0.
compute nbhd_UGW_sqft = 0.
compute nbhd_WA_sqft = 0.
compute nbhd_WV1_sqft = 0.
compute nbhd_WV2_sqft = 0.
compute nbhd_WVAT_sqft = 0.
compute nbhd_YC1_sqft = 0.
compute nbhd_YC2_sqft = 0.
compute nbhd_YE1_sqft = 0.
compute nbhd_YN1_sqft = 0.
compute nbhd_YN2_sqft = 0.
compute nbhd_YSE_sqft = 0.
compute nbhd_YSW_sqft = 0.
compute nbhd_ZH_sqft = 0.

if (model_nbhd = '151') nbhd_151_sqft = living_area.
if (model_nbhd = '153') nbhd_153_sqft = living_area.
if (model_nbhd = '260') nbhd_260_sqft = living_area.
if (model_nbhd = '261') nbhd_261_sqft = living_area.
if (model_nbhd = '263') nbhd_263_sqft = living_area.
if (model_nbhd = '370') nbhd_370_sqft = living_area.
if (model_nbhd = '371G') nbhd_371G_sqft = living_area./*This is a baseline Nbhd.
if (model_nbhd = '371R') nbhd_371R_sqft = living_area.
if (model_nbhd = '371S') nbhd_371S_sqft = living_area.
```

```
if (model_nbhd = '372') nbhd_372_sqft = living_area.
if (model_nbhd = '372S') nbhd_372S_sqft = living_area.
if (model_nbhd = '481') nbhd_481_sqft = living_area.
if (model_nbhd = '482') nbhd_482_sqft = living_area.
if (model_nbhd = '483') nbhd_483_sqft = living_area.
if (model_nbhd = '484') nbhd_484_sqft = living_area.
if (model_nbhd = '485') nbhd_485_sqft = living_area.
if (model_nbhd = '485O') nbhd_485O_sqft = living_area.
if (model_nbhd = '485P') nbhd_485P_sqft = living_area.
if (model_nbhd = 'GLR') nbhd_GLR_sqft = living_area.
if (model_nbhd = 'GLV') nbhd_GLV_sqft = living_area.
if (model_nbhd = 'GR') nbhd_GR_sqft = living_area./*This is a baseline Nbhd,
if (model_nbhd = 'GV') nbhd_GV_sqft = living_area.
if (model_nbhd = 'HA') nbhd_HA_sqft = living_area.
if (model_nbhd = 'MB') nbhd_MB_sqft = living_area.
if (model_nbhd = 'MX') nbhd_MX_sqft = living_area.
if (model_nbhd = 'NA') nbhd_NA_sqft = living_area.
if (model_nbhd = 'SL') nbhd_SL_sqft = living_area.
if (model_nbhd = 'SLN') nbhd_SLN_sqft = living_area.
if (model_nbhd = 'SLV') nbhd_SLV_sqft = living_area.
if (model_nbhd = 'SS') nbhd_SS_sqft = living_area.
if (model_nbhd = 'SSSH') nbhd_SSSH_sqft = living_area.
if (model_nbhd = 'TC') nbhd_TC_sqft = living_area.
if (model_nbhd = 'TH') nbhd_TH_sqft = living_area./*This is a baseline Nbhd.
if (model_nbhd = 'TP') nbhd_TP_sqft = living_area.
if (model_nbhd = 'UG') nbhd_UG_sqft = living_area.
if (model_nbhd = 'UGR') nbhd_UGR_sqft = living_area.
if (model_nbhd = 'UGW') nbhd_UGW_sqft = living_area.
if (model_nbhd = 'WA') nbhd_WA_sqft = living_area.
if (model_nbhd = 'WV1') nbhd_WV1_sqft = living_area.
if (model_nbhd = 'WV2') nbhd_WV2_sqft = living_area./*This is a baseline Nbhd.
if (model_nbhd = 'WVAT') nbhd_WVAT_sqft = living_area.
if (model_nbhd = 'YC1') nbhd_YC1_sqft = living_area.
if (model_nbhd = 'YC2') nbhd_YC2_sqft = living_area.
if (model_nbhd = 'YE1') nbhd_YE1_sqft = living_area.
if (model_nbhd = 'YN1') nbhd_YN1_sqft = living_area.
if (model_nbhd = 'YN2') nbhd_YN2_sqft = living_area.
if (model_nbhd = 'YSE') nbhd_YSE_sqft = living_area.
if (model_nbhd = 'YSW') nbhd_YSW_sqft = living_area.
if (model_nbhd = 'ZH') nbhd_ZH_sqft = living_area.

*Curbs and other adjustments to residence, not land.
compute curb_inf = 0.
if (curbs = "Y") curb_inf = 1.

compute sidewalk_inf = 0.
if (side_walk = "Y") sidewalk_inf = 1.

compute gas_inf = 0.
if (gas = "Y") gas_inf = 1.

compute ac_sf = ac * living_area.
```

```
*AGE AND EFFECTIVE AGE.
GRAPH/SCATTER=Age WITH TASP.
GRAPH/SCATTER=Eff_Age WITH TASP.

*Choosing Age instead of Age.
COMPUTE AgeEND1=25.
COMPUTE AgeEND2=39.
COMPUTE AgeEND3=120.
Compute Age1 = Age.
IF (Age>AgeEND1) Age1=AgeEND1.
COMPUTE Age2=0.
IF (Age>AgeEND1) Age2=Age-AgeEND1.
IF (Age>AgeEND2) Age2=AgeEND2-AgeEND1.
COMPUTE Age3=0.
IF (Age>AgeEND2) Age3=Age-AgeEND2.
MEANS Age1 Age2 Age3 BY AGE/CELLS=MEAN.


**REMOVE Detatched_Value from TASP.
COMPUTE TASP_Minus_Det = TASP - detatched_value.
COMPUTE TASP_Land = TASP - detatched_value - market_land.

*Normal Regression.
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10) TOLERANCE(.0001)
  /NOORIGIN
  /DEPENDENT  TASP_Land
  /METHOD=BACKWARD GRADE1_SF GRADE2_SF GRADE3_SF GRADE4_SF GRADE5_SF GRADE6_SF GRAD
_SF GRADE8_SF
     GRADE9_SF GRADE10_SF GRADE11_SF GRADE12_SF GRADE13_SF
     COND1_SF COND2_SF COND3_SF COND4_SF COND5_SF COND6_SF COND7_SF
     nbhd_151_sqft nbhd_153_sqft nbhd_260_sqft nbhd_261_sqft nbhd_263_sqft nbhd_370_sqft nbhd_371R_sqft
 nbhd_371S_sqft nbhd_372_sqft nbhd_372S_sqft
     nbhd_481_sqft nbhd_482_sqft nbhd_483_sqft nbhd_484_sqft nbhd_485_sqft nbhd_485O_sqft nbhd_485P_sq
ft nbhd_GLR_sqft nbhd_GLV_sqft nbhd_GV_sqft
     nbhd_HA_sqft nbhd_MB_sqft nbhd_MX_sqft nbhd_NA_sqft nbhd_SL_sqft nbhd_SLN_sqft nbhd_SS_sqft nbhd_
TC_sqft nbhd_TP_sqft nbhd_UG_sqft
     nbhd_UGR_sqft nbhd_UGW_sqft nbhd_WA_sqft nbhd_WV1_sqft nbhd_YC1_sqft nbhd_YE1_sqft nbhd_YN1_sq
ft nbhd_YN2_sqft nbhd_YSE_sqft nbhd_ZH_sqft
     roofing_CP roofing_MT roofing_OT
     living_area total_area1 total_area2 MainFloor FinBSMT att_gar ac_sf Age1 Age2 AGE3
     num_bedrooms PlumbingFixtures fireplace ext_wall_AL ext_wall_HB ext_wall_SS ext_wall_VN ext_wall_WD
     full_bath half_bath
     Carport Covered_Porch Open_Porch Enclose_Porch Screen_Porch Breezeway WoodDeck Patio .

*Normal Regression Stepwise.
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10) TOLERANCE(.0001)
  /NOORIGIN
```

```
  /DEPENDENT  TASP_Land
  /METHOD=STEPWISE GRADE1_SF GRADE2_SF GRADE3_SF GRADE4_SF GRADE5_SF GRADE6_SF GRADE
SF GRADE8_SF
     GRADE9_SF GRADE10_SF GRADE11_SF GRADE12_SF GRADE13_SF
     COND1_SF COND2_SF COND3_SF COND4_SF COND5_SF COND6_SF COND7_SF
     nbhd_151_sqft nbhd_153_sqft nbhd_260_sqft nbhd_261_sqft nbhd_263_sqft nbhd_370_sqft nbhd_371R_sqft
 nbhd_371S_sqft nbhd_372_sqft nbhd_372S_sqft
     nbhd_481_sqft nbhd_482_sqft nbhd_483_sqft nbhd_484_sqft nbhd_485_sqft nbhd_485O_sqft nbhd_485P_sq
ft nbhd_GLR_sqft nbhd_GLV_sqft nbhd_GV_sqft
     nbhd_HA_sqft nbhd_MB_sqft nbhd_MX_sqft nbhd_NA_sqft nbhd_SL_sqft nbhd_SLN_sqft nbhd_SS_sqft nbhd_
TC_sqft nbhd_TP_sqft nbhd_UG_sqft
     nbhd_UGR_sqft nbhd_UGW_sqft nbhd_WA_sqft nbhd_WV1_sqft nbhd_YC1_sqft nbhd_YE1_sqft nbhd_YN1_sq
ft nbhd_YN2_sqft nbhd_YSE_sqft nbhd_ZH_sqft
     roofing_CP roofing_MT roofing_OT
     living_area total_area1 total_area2 MainFloor FinBSMT att_gar ac_sf Age1 Age2 AGE3
     num_bedrooms PlumbingFixtures fireplace ext_wall_AL ext_wall_HB ext_wall_SS ext_wall_VN ext_wall_WD
     masonry_brick masonry_none full_bath half_bath
     Carport Covered_Porch Open_Porch Enclose_Porch Screen_Porch Breezeway WoodDeck Patio.


*Backward Regression produces better results.

*FInal Model.
REGRESSION
  /MISSING LISTWISE
  /STATISTICS COEFF OUTS R ANOVA
  /CRITERIA=PIN(.05) POUT(.10) TOLERANCE(.0001)
  /NOORIGIN
  /DEPENDENT TASP_Land
  /METHOD=ENTER COND1_SF COND2_SF COND3_SF COND6_SF
  nbhd_260_sqft nbhd_261_sqft nbhd_371R_sqft nbhd_482_sqft nbhd_484_sqft nbhd_485_sqft nbhd_485O_sqft
 nbhd_GLR_sqft
  nbhd_GLV_sqft nbhd_GV_sqft nbhd_HA_sqft nbhd_SLN_sqft nbhd_SS_sqft roofing_MT
  MainFloor ac_sf Age2 PlumbingFixtures ext_wall_AL ext_wall_SS  Covered_Porch
  /SAVE PRED (market_pred).

MEANS TABLES=market_pred /CELLS=MIN MAX MEDIAN MEAN COUNT STDDEV.

compute market_total_pred = market_pred + market_land + detatched_value.
compute Pred_Market_Ratio = market_total_pred/TASP.

compute PredDOROutlier = 0.
if (Pred_Market_Ratio > 1.75) PredDOROutlier = 1.
if (Pred_Market_Ratio < 0.25) PredDOROutlier = 1.

compute RatioFilter = 1.
if (PredDOROutlier = 1) RatioFilter = 0.
if (DOROutlier = 0) RatioFilter = 0.

filter by RatioFilter.

RATIO STATISTICS market_total_pred WITH TASP /PRINT=COD MEAN MEDIAN PRB PRD WGTMEAN N.
```

```
RATIO STATISTICS market_total_pred WITH TASP BY model_nbhd (ASCENDING)/PRINT=COD MEAN MEDIAN
PRD WGTMEAN N.
if (model_nbhd = '151') market_total_pred = market_total_pred *1.0081.
if (model_nbhd = '153') market_total_pred = market_total_pred *1.0695.
if (model_nbhd = '260') market_total_pred = market_total_pred *1.0331.
if (model_nbhd = '261') market_total_pred = market_total_pred *1.1013.
if (model_nbhd = '263') market_total_pred = market_total_pred *1.1062.
if (model_nbhd = '370') market_total_pred = market_total_pred *1.0352.
if (model_nbhd = '371G') market_total_pred = market_total_pred *1.073.
if (model_nbhd = '371R') market_total_pred = market_total_pred *0.9881.
if (model_nbhd = '371S') market_total_pred = market_total_pred *1.0661.
if (model_nbhd = '372') market_total_pred = market_total_pred *1.0893.
if (model_nbhd = '372S') market_total_pred = market_total_pred *1.2092.
if (model_nbhd = '481') market_total_pred = market_total_pred *1.0121.
if (model_nbhd = '482') market_total_pred = market_total_pred *0.9785.
if (model_nbhd = '483') market_total_pred = market_total_pred *1.007.
if (model_nbhd = '484') market_total_pred = market_total_pred *0.9775.
if (model_nbhd = '485') market_total_pred = market_total_pred *0.9823.
if (model_nbhd = '485O') market_total_pred = market_total_pred *0.9901.
if (model_nbhd = '485P') market_total_pred = market_total_pred *0.9116.
if (model_nbhd = 'GLR') market_total_pred = market_total_pred *0.9852.
if (model_nbhd = 'GLV') market_total_pred = market_total_pred *1.
if (model_nbhd = 'GR') market_total_pred = market_total_pred *0.8969.
if (model_nbhd = 'GV') market_total_pred = market_total_pred *1.1416.
if (model_nbhd = 'HA') market_total_pred = market_total_pred *1.
if (model_nbhd = 'MB') market_total_pred = market_total_pred *0.9515.
if (model_nbhd = 'MX') market_total_pred = market_total_pred *1.0101.
if (model_nbhd = 'NA') market_total_pred = market_total_pred *0.996.
if (model_nbhd = 'SL') market_total_pred = market_total_pred *1.0893.
if (model_nbhd = 'SLN') market_total_pred = market_total_pred *1.0142.
if (model_nbhd = 'SS') market_total_pred = market_total_pred *1.0965.
if (model_nbhd = 'TC') market_total_pred = market_total_pred *1.087.
if (model_nbhd = 'TH') market_total_pred = market_total_pred *0.9551.
if (model_nbhd = 'TP') market_total_pred = market_total_pred *1.004.
if (model_nbhd = 'UG') market_total_pred = market_total_pred *0.9443.
if (model_nbhd = 'UGW') market_total_pred = market_total_pred *1.227.
if (model_nbhd = 'WA') market_total_pred = market_total_pred *1.0549.
if (model_nbhd = 'WV1') market_total_pred = market_total_pred *0.9833.
if (model_nbhd = 'WV2') market_total_pred = market_total_pred *1.006.
if (model_nbhd = 'YC1') market_total_pred = market_total_pred *1.0417.
if (model_nbhd = 'YE1') market_total_pred = market_total_pred *0.9515.
if (model_nbhd = 'YN2') market_total_pred = market_total_pred *0.9921.
if (model_nbhd = 'YSE') market_total_pred = market_total_pred *0.8772.
if (model_nbhd = 'ZH') market_total_pred = market_total_pred *0.9872.
RATIO STATISTICS market_total_pred WITH TASP BY model_nbhd (ASCENDING)/PRINT=COD MEAN MEDIAN
PRD WGTMEAN N.

RATIO STATISTICS market_total_pred WITH TASP BY quality_grade (ASCENDING)/PRINT=COD MEAN MEDIAN
PRD WGTMEAN N.
if (quality_grade = 2) market_total_pred = market_total_pred *1.0977.
if (quality_grade = 3) market_total_pred = market_total_pred *1.0776.
if (quality_grade = 4) market_total_pred = market_total_pred *1.
if (quality_grade = 5) market_total_pred = market_total_pred *1.
```

```
if (quality_grade = 6) market_total_pred = market_total_pred *1.0352.
RATIO STATISTICS market_total_pred WITH TASP BY quality_grade (ASCENDING)/PRINT=COD MEAN MEDIAN
PRD WGTMEAN N.

RATIO STATISTICS market_total_pred WITH TASP BY condition_grade (ASCENDING)/PRINT=COD MEAN MEDIAN
 PRD WGTMEAN N.
if (condition_grade = 1) market_total_pred = market_total_pred *1.
if (condition_grade = 2) market_total_pred = market_total_pred *1.0893.
if (condition_grade = 3) market_total_pred = market_total_pred *0.9891.
if (condition_grade = 4) market_total_pred = market_total_pred *1.
if (condition_grade = 5) market_total_pred = market_total_pred *1.
if (condition_grade = 6) market_total_pred = market_total_pred *0.997.
if (condition_grade = 7) market_total_pred = market_total_pred *1.006.
RATIO STATISTICS market_total_pred WITH TASP BY condition_grade (ASCENDING)/PRINT=COD MEAN MEDIAN
 PRD WGTMEAN N.

RATIO STATISTICS market_total_pred WITH TASP BY sqft_range (ASCENDING)/PRINT=COD MEAN MEDIAN PRD
 WGTMEAN N.
if (sqft_range = 0) market_total_pred = market_total_pred *1.0091.
if (sqft_range = 1) market_total_pred = market_total_pred *1.
if (sqft_range = 2) market_total_pred = market_total_pred *1.003.
if (sqft_range = 3) market_total_pred = market_total_pred *0.9804.
if (sqft_range = 4) market_total_pred = market_total_pred *0.997.
RATIO STATISTICS market_total_pred WITH TASP BY sqft_range (ASCENDING)/PRINT=COD MEAN MEDIAN PRD
 WGTMEAN N.

RATIO STATISTICS market_total_pred WITH TASP BY age_range (ASCENDING)/PRINT=COD MEAN MEDIAN PRD
 WGTMEAN N.
if (age_range = 0) market_total_pred = market_total_pred *1.007.
if (age_range = 1) market_total_pred = market_total_pred *0.997.
if (age_range = 2) market_total_pred = market_total_pred *1.001.
if (age_range = 3) market_total_pred = market_total_pred *1.
if (age_range = 4) market_total_pred = market_total_pred *0.9881.
RATIO STATISTICS market_total_pred WITH TASP BY age_range (ASCENDING)/PRINT=COD MEAN MEDIAN PRD
 WGTMEAN N.

compute Pred_Market_Ratio = market_total_pred/TASP.
GRAPH/HISTOGRAM=Pred_Market_Ratio.

EXAMINE VARIABLES=Pred_Market_Ratio /PLOT BOXPLOT STEMLEAF /COMPARE GROUPS
/PERCENTILES(25,50,75) HAVERAGE /STATISTICS DESCRIPTIVES /CINTERVAL 95 /MISSING LISTWISE
/NOTOTAL.

* GIven a median of 0.9998 and an interquartile range of 0.1483 the Extreme Outlier Threshold is 0.4449 (3.0 X
IQR).
*This gives us a lower bound of .5551 (say .6) and an upper bound of 1.4449 (say 1.5).
compute finalOutlier = 0.
if (Pred_Market_Ratio > 1.5000) finalOutlier = 1.
if (Pred_Market_Ratio < 0.6000) finalOutlier = 1.
if (finalOutlier = 1) RatioFilter = 0.
Filter by RatioFilter.

*After Market Adjustments.
```

```
RATIO STATISTICS market_total_pred WITH TASP/PRINT=COD MEAN MEDIAN PRB PRD WGTMEAN COV N.
MEANS TABLES=market_total_pred /CELLS=MIN MAX MEDIAN MEAN COUNT STDDEV.
```