```
* Aggregated Data Scripts
* Murphy John;


*******************************************************************************
* Process Data Script
* Murphy John
* 2025-04-07
* This script loads, processes, and compiles the data used in this project.
*******************************************************************************;

title "Setup";

** footnote;
footnote "Data processing script run on &SYSDATE at &SYSTIME.";

** establish library;
libname mylib "/home/u63984496/BIOS7400/final-project";
*******************************************************************************;

title "Data processing";

title2 "Avocado Data";

* load data;
proc import datafile="/home/u63984496/BIOS7400/final-project/avocado.csv"
    out=work.raw_avo
    dbms=csv
    replace;
    guessingrows=MAX;
run;

* data processing;
data work.clean_avo;
    * read raw avocado data;
    * rename select variables;
    set work.raw_avo(rename = (
        AveragePrice = avgprice
        'Total Volume'n = totvol
        '4046'n = totsm
        '4225'n = totlg
        '4770'n = totxl
        'Total Bags'n = totbags
        'Small Bags'n = totbags_sm
        'Large Bags'n = totbags_lg
        'XLarge Bags'n = totbags_xl
        ));

    * seperate date by month and year;
    * create a month year variable;
    month = put(date, monname.);
    month_num = month(date);
    month = strip(propcase(month));
    date = mdy(month_num, 1, year);

    * keep only specififc regions;
    if region not in (
        "California",
        "West",
        "Northeast",
        "SouthCentral",
        "Southeast",
        "GreatLakes",
        "MidSouth",
        "Plains")
        then delete;

    drop VAR1;
run;

* group by year, month, region, and type;
proc sql;
    create table work.avo_group as
    select
        year,
        month,
```

```
        month_num,
        date,
        region,
        type,
        mean(avgprice) as avgprice format=8.2,
        sum(totvol) as totvol,
        sum(totsm) as totsm,
        sum(totlg) as totlg,
        sum(totxl) as totxl,
        sum(totbags) as totbags,
        sum(totbags_sm) as totbags_sm,
        sum(totbags_lg) as totbags_lg,
        sum(totbags_xl) as totbags_xl
    from work.clean_avo
    group by date, region, type;
quit;

* sort by date and remove duplicate obs;
proc sort data=work.avo_group nodupkey out=work.dat_avo;
    by date region type;
run;

* print first 10 obs;
proc print data=work.dat_avo(obs=10);
run;

title2 "Temperature Data";

** load data;
filename raw_temp '/home/u63984496/BIOS7400/final-project/temp.txt';
data dat_temp;
    * read raw temp data;
    infile raw_temp;

    * use absolute input pointer control;
    input @;

    * delete non-numeric values;
    if notdigit(scan(_infile_, 1)) then delete;

    * create year and month columns;
    else input year January February March April May June July August September October November December;

    * keep only years 2015 - 2018;
    if year < 2015 or year > 2018 then delete;

    * temperatures are in 0.01 degrees C. convert to actual degrees C;
    * pivot longer to create a month/year column and temp column;
    length month $9;
    array col{12} January February March April May June July August September October November December;
        do i = 1 to 12;
            temp = round(col{i} / 100, 0.01);
            month = vname(col{i});
            output;
        end;
    month = strip(propcase(month));

    * keep year month temp cols only;
    keep year month temp

run;

* print first 10 obs;
proc print data=work.dat_temp(obs=10);
run;

title2 "President Data";

*** In 2015 and 2016, Barack Obama of the democratic party was president of the US.
*** In 2017 and 2018, Donald Trump of the republican party was president of the US.;

* establish data;
data dat_pres;
    length year 4 president $ 20 pres_party $ 25;
    input year president pres_party;
    infile datalines dsd dlm = " ";
    datalines;
```

```
2015 "Barack Obama" "Democratic"
2016 "Barack Obama" "Democratic"
2017 "Donald Trump" "Republican"
2018 "Donald Trump" "Republican"
;
run;

* print;
proc print data=work.dat_pres;
run;
*******************************************************************************;

title "Data merging";
* sql can handle many-to-one merging;
* save to mylib;
proc sql;
    create table work.dat_merge as
    select
        a.*,
        b.*,
        c.*
    from work.dat_avo as a
    inner join work.dat_temp as b
        on a.year = b.year and a.month = b.month
    inner join work.dat_pres as c
        on a.year = c.year;
quit;

* add labels to variables;
data mylib.dat;
    set work.dat_merge;
    label
        year = "Year"
        month = "Month Name"
        month_num = "Month Number"
        date = "Date of observation- only month and years are known"
        region = "City or region of the observation"
        type = "Type of farming method"
        avgprice = "Average price of a single avocado"
        totvol = "Total Number of avocados sold"
        totsm = "Total number of avocados with PLU 4046 (small) sold"
        totlg = "Total number of avocados with PLU 4225 (large) sold"
        totxl = "Total number of avocados with PLU 4770 (xlarge) sold"
        totbags = "Total number of bags sold"
        totbags_sm = "Total number of PLU 4046 (small) bags sold"
        totbags_lg = "Total number of PLU 4225 (large) bags sold"
        totbags_xl = "Total number of PLU 4770 (xlarge) bags sold"
        temp = "Temperature difference (degress C)"
        president = "Name of current U.S. president"
        pres_party = "Poliical Party of current U.S. president";
run;
*******************************************************************************;

title "Print data";

* print first 10 obs;
proc print data=mylib.dat(obs=10);
run;

* get frequency tables;
proc freq data=mylib.dat;
    tables year month region type pres_party;
run;

* describe dataset;
proc contents data=mylib.dat;
run;

* END OF PROCESSING SCRIPT;

*******************************************************************************
* Analysis Script
* Murphy John
* 2025-04-21
* This script performs the main analyses of this project.
*******************************************************************************;
```

```
title "Setup";

** footnote;
footnote "Analysis script run on &SYSDATE at &SYSTIME.";

** establish library;
libname mylib "/home/u63984496/BIOS7400/final-project";

** set graphics ods;
ods graphics on / width=8in height=4in;
*******************************************************************************;

title "Exploratory analysis";

* plot the outcome of interest, avgprice;
** sort data by type, date, and region;
proc sort data=mylib.dat;
    by type date region;
run;

** plot price over time by type stratified by region;
proc sgpanel data=mylib.dat;
    panelby type;
    series x = date y = avgprice / group = region;
run;

** plot price by pres_party and type stratified by region;
proc sgpanel data=mylib.dat;
    panelby region type;
    hbox avgprice / group = pres_party;
run;

** plot price by temp and type stratified by region;
proc sgpanel data=mylib.dat;
    panelby type;
    scatter x = temp y = avgprice / group = region;
run;
*******************************************************************************;

* univariable analysis;
** get means for each type;
proc univariate data=mylib.dat plots;
    var avgprice;
    class type;
run;

* bivariable analysis;
** check correlations of numerical variables;
proc corr pearson data=mylib.dat;
    var temp avgprice;
    by type;
run;

** create macro to check means of categorical variables;
%macro means(var);
proc means data = mylib.dat;
    var avgprice;
    class &var;
    by type;
run;
%mend means;

* run;
%means(month_num);
%means(year);
%means(region);
%means(pres_party);
*******************************************************************************;

title "Regression Model fits";

* create train and test data;
** randomly select 80 percent of the data for the training and reserve the remainder for testing;
proc surveyselect data=mylib.dat
    out=dat_select
    samprate=0.8
    outall
```

```
    seed=333;
run;

** create seperate data sets;
data dat_train dat_test;
    set dat_select;
    if selected then output dat_train;
    else output dat_test;
run;


* simple linear regression;
** avgprice by type, only;
proc glm data=dat_train plots=all;
    class type (ref='conventional');
    model avgprice = type;
run;

*** rmse = 0.251153, rsq = 0.485679;

* bivariable regression with interaction;
** model avgprice by type, covariate, and interaction;
** covariates are month_num, year, temp, region, pres_party;

* write macro to fit the numerical covariate model;
%macro bivariable_num(covariate);
    proc glm data=dat_train plots=all;
        class type (ref='conventional');
        model avgprice = type &covariate type*&covariate;
    run;

%mend bivariable_num;

* write macro to fit the categorical covariate models;
%macro bivariable_cat(covariate);
    proc glm data=dat_train plots=all;
        class type (ref='conventional') &covariate;
        model avgprice = type &covariate type*&covariate;
    run;

%mend bivariable_cat;

* run;

%bivariable_num(temp);
*** rmse = 0.234790, rsq = 0.552579;

%bivariable_cat(region);
*** rmse = 0.202991, rsq = 0.673291;

%bivariable_cat(pres_party);
*** rmse = 0.244407, rsq = 0.515176;

%bivariable_cat(year);
*** rmse = 0.239643, rsq = 0.538200 ;

%bivariable_cat(month_num);
*** rmse = 0.231129, rsq = 0.586450;

* create table of metrics by hand;
data stats;
    length Covariate $30;
    infile datalines dsd truncover;
    input Covariate :$30. R_Square RMSE;
    datalines;
"Type only",0.485679,0.251153
"Temperature",0.552579,0.234790
"Region",0.673291,0.202991
"Presidential Party",0.515176,0.244407
"Year",0.538200,0.239643
"Month",0.586450,0.231129
;
run;

proc print data=stats noobs;
    var Covariate RMSE R_Square;
run;
```

```
*** All three bivariable models improve the fit above the univariable model as measued by the rmse and r-squared.
*** The bivariable model with region has the lowest rmse and highest r-squared.;

* full model;
** avgprice by type, temp, region, pres_party and interactions;
proc glm data=dat_train plots=all;
        class pres_party type (ref='conventional') region month_num year;
        model avgprice = type temp region pres_party month_num year
              type*temp type*region type*pres_party type*month_num type*year;
run;
*** rmse = 0.145137, rsq = 0.844828;

* reduced model;
** the interaction between type and temp, month_num, and pres_party have small Type III SS and large p-values.
** Reduce the model by removing these terms.;
proc glm data=dat_train plots=all alpha=0.05;
        class type (ref='conventional') region pres_party month_num year;
        model avgprice = type temp region pres_party month_num year
              type*region;
        store out=final_model;
run;
quit;
*** rmse = 0.152099, rsq = 0.823078;

*** The reduced model is very similar in terms of fit to the full model but does not include unnessecary interaction terms.
*** Select this as the final model.;
*******************************************************************************;

title 'Model Evaluation';
* use the final model to generate predictions on the test data;
proc plm restore=final_model;
    score data=dat_test out=predictions predicted;
run;

* compute rmse and r-squared;
** get residuals;
data eval;
    set predictions;
    resid = avgprice - predicted;
    sq_resid = resid**2;
run;

* compute mean;
proc means data=eval noprint;
    var avgprice predicted sq_resid;
    output out=metrics
        mean(avgprice)=mean_y
        sum(sq_resid)=ss_res
        n=samples;
run;

* compute metrics;
data results;
    set metrics;
    ss_total = 0;
    do i = 1 to samples;
        set eval point=i nobs=n;
        ss_total + (avgprice - mean_y)**2;
    end;
    rmse = sqrt(ss_res / samples);
    rsq = 1 - (ss_res / ss_total);
    keep rmse rsq;
run;

proc print data=results;
    title "RMSE and R-squared on Test Data";
run;

* residuals vs fitted plot;
proc sgplot data=eval;
    scatter x=predicted y=resid / markerattrs=(symbol=circlefilled color=black);
    refline 0 / axis=y lineattrs=(color=red pattern=shortdash);
    xaxis label="Fitted Values";
```