```
********************************************************************************
*  TITLE :      SAS GRAIN PRICE PROJECT
*
*  DESCRIPTION: Final project for BIOS 7400 with Xiao Song, UGA, Spring 2022.
*               Cleaning data for grain price analysis.
*
*-------------------------------------------------------------------------------
*  JOB NAME:    cleaning.SAS
*  LANGUAGE:    SAS v9.4 (on demand for academics)
*
*  NAME:        Zane Billings
*  DATE:        2022-04-20
*
*******************************************************************************;

FOOTNOTE "Job run by Zane Billings on &SYSDATE at &SYSTIME";

TITLE 'Grain Price Analysis';

OPTIONS NODATE LS=95 PS=42;

LIBNAME HOME '/home/u59465388/SAS-Grain-Prices';

*******************************************************************************;
* Macros;
*******************************************************************************;

* Variables for filtering the years to export in the cleaned dataset. I have
    them set to the min/max values in the dataset, but this allows for easier
    changing than specifying the years manually.;
%LET MINYEAR = 1866;
%LET MAXYEAR = 2021;

* Variable for controlling whether the following macro prints to the report.
    It is easier to toggle this in one place than to add or remove the macro
    calls later in the script.
    1: Prints first &PRINTN observations of the dataset and the descriptor
        portion as well.
    Any other value (preferably 0): does not print (indeed, the macro will
        not execute anything after the logical step).;
%LET VERBOSE = 1;
%LET PRINTN = 10;

* Macro for printing values and descriptor portion of data;
%MACRO DESCRIBE (DAT =, N = &PRINTN);
    %IF %EVAL(&VERBOSE = 1) %THEN %DO;
        PROC PRINT DATA = &DAT (OBS = &N) LABEL;
        RUN;

        PROC CONTENTS DATA = &DAT;
        RUN;
    %END;
%MEND;

*******************************************************************************;
* Data importing;
```

```
*******************************************************************************;

* Import the temperature anomaly data;
FILENAME NASATEMP "/home/u59465388/SAS-Grain-Prices/nasatemp.txt";
DATA TEMP;
    * Read in the NASA temperature data. The data starts at line 9.;
    INFILE NASATEMP FIRSTOBS = 9;

    * Bring the next line of the INFILE into the input buffer;
    INPUT @;

    * If the first detectable word (which should be the YEAR) is not a numeric
      digit, delete the row from the buffer, and thus do not import it.
      This skips the blank rows and repeated header rows.
      After DELETE is executed, return to the beginning of the data step.;
    IF NOTDIGIT(SCAN(_INFILE_, 1)) THEN DELETE;

    * If the YEAR is a number, import the current infile into the dataset;
    ELSE DO;
        * The data has missing values coded as '****', replace these with . so that
          SAS interprets them as missing correctly.;
        _INFILE_ = TRANSTRN(_INFILE_, "****", ".");
        * Read in only the first 13 columns.;
        INPUT YEAR JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC;
    END;

    * Get the yearly average, and then divide by 100 to make the units degrees C.
        Round to two decimal places.;
    TEMP = ROUND(MEAN(OF JAN -- DEC) / 100, 0.01);
    DROP JAN -- DEC;

    * Give information labels to the variables;
    LABEL
        YEAR = "Calendar year"
        TEMP = "Temperature diff. (deg. C)"
    ;
RUN;

%DESCRIBE(DAT = WORK.TEMP);

* Import the presidential party data;
FILENAME PRESI '/home/u59465388/SAS-Grain-Prices/presidential.csv';
DATA PRES;
    * Set length of variables to ensure character vars don't get cut off;
    LENGTH YEAR 4 PRES $ 20 PARTY $ 25;

    * Import CSV file, nothing complicated like the last file;
    INFILE PRESI DLM = ',' FIRSTOBS = 2;
    INPUT YEAR PRES $ PARTY $;

    * Abraham Lincoln and Andrew Johnson are listed as 'National Union' party
        members, but this isn't terribly useful. Historically, Abraham Lincoln
        was a Republican and Andrew Johnson was a Democrat, and the National Union
        coalition was a transitionary step. So I'll recode these two for simplicity.;
    IF PRES = "Abraham Lincoln" THEN PARTY = "Republican";
    ELSE IF PRES = "Andrew Johnson" THEN PARTY = "Democrat";

    * Add descriptive labels;
    LABEL
        YEAR = "Calendar year"
        PRES = "President name"
        PARTY = "President party"
    ;
RUN;

* The presidential data only goes through 2013, so we will have to manually
    input the 2013 - 2022 data and append that to the end.;
DATA PRES_END;
    LENGTH YEAR 4 PRES $ 20 PARTY $ 25;
    INPUT YEAR PRES $ PARTY $;
    LABEL
        YEAR = "Calendar year"
        PRES = "President name"
        PARTY = "President party"
    ;
    INFILE DATALINES DSD DLM = " ";
    DATALINES;
2014 "Barack Obama" "Democrat"
2015 "Barack Obama" "Democrat"
2016 "Barack Obama" "Democrat"
2017 "Donald Trump" "Republican"
2018 "Donald Trump" "Republican"
2019 "Donald Trump" "Republican"
2020 "Donald Trump" "Republican"
2021 "Joseph Biden" "Democrat"
2022 "Joseph Biden" "Democrat"
;
RUN;

* Now append the second dataset to the end of the first;
PROC APPEND BASE = WORK.PRES DATA = WORK.PRES_END;
RUN;

%DESCRIBE(DAT = WORK.PRES);

* Import the inflation data;
FILENAME INFL '/home/u59465388/SAS-Grain-Prices/inflation_data.csv';
DATA INFLATION;
    * Import CSV file, easy like the presidential data;
    INFILE INFL DLM = ',' FIRSTOBS = 2;
    INPUT YEAR VALUE INFL;

    * Create a new column for relative 'worth': 1 / value in 1886 dollars
      is the 'buying power' of $1 relative to an 1866 dollar.;
    PWR = ROUND(1 / VALUE, 0.01);

    * Assign descriptive lables;
    LABEL
        YEAR = 'Calendar year'
        VALUE = 'Adjusted value'
        INFL = 'Rate of inflation'
        PWR = 'Buying power'
    ;
RUN;

```

```

%DESCRIBE(DAT = WORK.INFLATION);

* Import the feed grains data. This is a complex and messy excel spreadsheet
  that is easy to manually view but difficult to use as actual data. For
  this project, I will only clean the first sheet.;
* In the current form, importing the data will be quite complicated and I think
  impossible using PROC IMPORT. So I opened the dataset in Excel and exported
  the sheet that I needed as a CSV file, which is what I'll import here.;
FILENAME FDGRN '/home/u59465388/SAS-Grain-Prices/fg-sheet1.csv';

DATA ALLGRNS;
  * Import the CSV file. The option DSD is necessary to read in consecutive
    delimiters as missing data, and the MISSOVER option is necessary as
    there are missing values at the end of lines, so the INPUT specification
    should be interpreted strictly.;
  INFILE FDGRN DLM = ',' FIRSTOBS = 9 DSD MISSOVER;

  * SAS doesn't like the missing values being denoted by ,, even with the DSD
    option, and has a hard time parsing the numeric values. So, I'll import
    all of the variables as character variables with silly names. The
    names are uninformative, but easy to use all together in SAS statements.
   Note that I have also included the trailing @ so I can check the next line
    for all blanks, and delete the line before being read if that is the case.;
  INPUT GRN $ YR $ V1 $ V2 $ V3 $ V4 $ V5 $ V6 $ @;

  * If the next line (@) is all missing, do not read it in;
  IF MISSING(YR) THEN DELETE;

  * The grain variable is only denoted once, and is missing for all other
    records in the time series. This part of the code saves the most recent
    non-missing value of GRN, and then uses it to fill in the value of
    all missing GRN values until it finds a new non-missing value.;
  IF NOT MISSING(GRN) THEN DO;
    TMP = GRN;
    RETAIN TMP;
  END;
  ELSE GRN = TMP;

  * Create a YEAR variable as the first four digits of the YR variable, which
    looks like ####/##. Use INPUT() to make this new variable numeric.;
  YEAR = INPUT(SUBSTR(YR, 1, 4), 4.);

  * Convert the imported character variables to numeric variables. Since SAS
    cannot modify variable types in place, we have to create two arrays. One
    array (_CHA) holds the placeholder character variables, and the second array
    (_NUM) holds the newly declared numeric variables with somewhat better
    names. Then we handle the missing character values explicitly to prevent SAS
    from complaining about the blanks, and use INPUT to parse the remaining
    values to numbers. We use the comma informat here since some of the
    numeric values have commas as place value separators.;
  ARRAY _CHA{6} $ V1 - V6;
  ARRAY _NUM{6} ACR HVT PRD YLD PCE LNR;
  DO I = 1 TO 6;
    IF MISSING(_CHA{I}) THEN _NUM{I} = .;
    ELSE _NUM{I} = INPUT(_CHA{I}, COMMA8.);
  END;

```

```

    * Compute the percent change from the previous year;
    PCT = ROUND(DIF(PCE) / LAG(PCE) * 100, 0.01);

    * Compute the log of the price;
    LPE = LOG10(PCE);

    * Drop all of the temporary and placeholder variables that we don't need in
        the cleaned dataset;
    DROP TMP YR V1 - V6 I;

    * Assign descriptive labels to the remaining useful variables.;
    LABEL
        GRN = "Grain commodity"
        YEAR = "Calendar year"
        ACR = "Acerage (M)"
        HVT = "Acres harvested (M)"
        PRD = "Bushels produced (M)"
        YLD = "Yield (bushels per acre)"
        PCE = "Price per bushel"
        LPE = "log10 price per bushel"
        LNR = "Loan rate per bushel"
        PCT = "Pct change in price"
    ;
RUN;

%DESCRIBE(DAT = WORK.ALLGRNS);

********************************************************************************;
* Data merging;
********************************************************************************;

* Next, we need to do a one-to-many merge of the four datasets by year. The
    grains dataset has up to four records for each year, so the other three
    datasets will need to be replicated.;

* First, we must sort all data sets by year. This macro will sort an arbitrary
    number of datasets. Note that it mutates currently existing datasets rather
    than assigning new names to the sorted datasets.;

%MACRO SORTALL (DAT = , BYVAR = );
    %LET N = %SYSFUNC(COUNTW(&DAT));
    %DO I = 1 %TO &N;
        PROC SORT DATA = %SCAN(&DAT, &I);
            BY &BYVAR;
        RUN;
    %END;
%MEND;

%SORTALL(
    DAT = ALLGRNS INFLATION PRES TEMP,
    BYVAR = YEAR
);

* Now we can do the actual merge. Only the records with admissible years
    (specified by the macro variables &MINYEAR and &MAXYEAR respectively)
    will be read in and included in the merge.;

```

```

DATA HOME.GRAINS;
    MERGE ALLGRNS INFLATION PRES TEMP;
    WHERE &MINYEAR <= YEAR <= &MAXYEAR;
    BY YEAR;
RUN;

PROC SORT DATA = HOME.GRAINS;
    BY GRN YEAR;
```