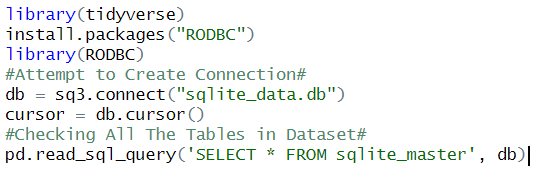
*Supply Chain Analyst Case Study – Using Data Analysis Techniques to Determine Investment Strategy with Historical Foreign Grain Markets*

In this case study, I have just been hired by a US Venture Capital firm as a data scientist. The company is considering foreign grain markets to help meet its supply chain requirement for its recent investments in the microbrewery and microdistillery industry, which is involved with the production and distribution of crafts beer and spirits.

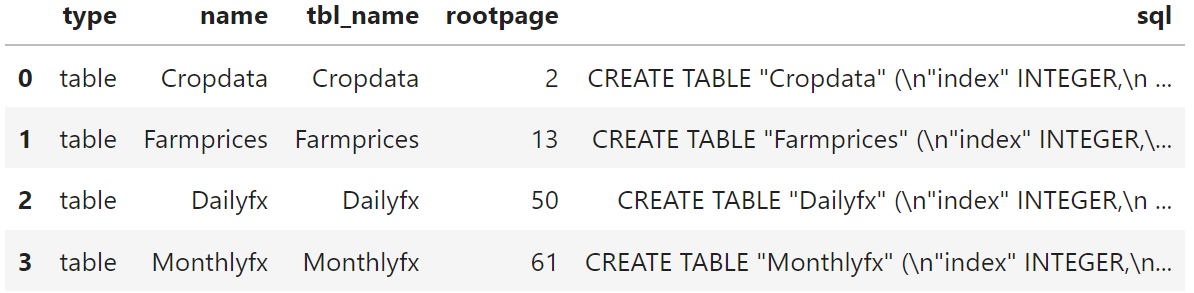
My first task is to provide a high level analysis of crop production in Canada. At the US Venture Capital firm, my key stakeholders want to under the current and historical performance of certain crop types in terms of supply and price. At the moment, my key stakeholders are mainly interested in a macro-view of Canada’s crop farming industry and how it relates to the relative value of the Canadian and US dollar.

When tackling this business problem, my first exercise would be to retrieve data relevant to the analysis task. Upon doing some research, I come across three datasets that detail Canadian principal crops, farm product prices, and the Bank of Canada daily average exchange rates. The Canadian principal crops dataset contains agricultural production measures for the principal crops grown in Canada, including a breakdown by province and territory, for each year from 1908-2020. The farm product prices dataset contains monthly average farm product prices for Canadian crops and livestock by province and territory from 1980-2020. The Bank of Canada daily average exchange rates dataset contains the daily average exchange rates for multiple foreign currencies. However, it only includes the latest four years of data, and the rates are published once each business day by 16:30 ET. Further access and details to the dataset will be provided at the end of this case study.

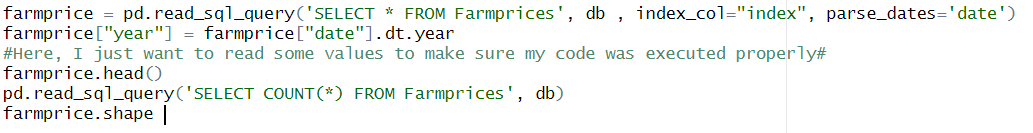
Before beginning the project, the first thing I want to do is to establish a connection to the Db2 database. From there, I know I can create tables for my data using the RODBC package in R. So, I run the following code in R:



In doing so, I confirm the that dataset was imported correctly. Likewise, I can directly view my dataset by checking all the tables with the final part of the previous code. The table displays the following:



Next, the first problem I am tasked with from my supervisors is to figure out how many records are in the farm price dataset. To determine this, I run the following R code:

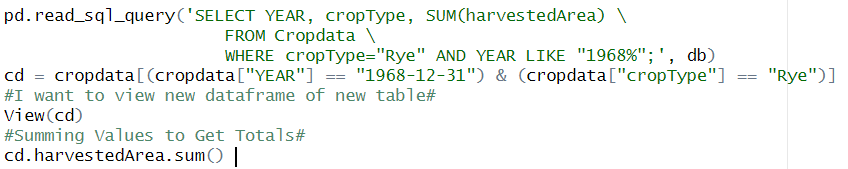


Ultimately, I determine that there are 2,678 records in the farm price dataset. Next, I am tasked to determine which provinces are included in the farm prices dataset. To figure that out, I run the following code in R:



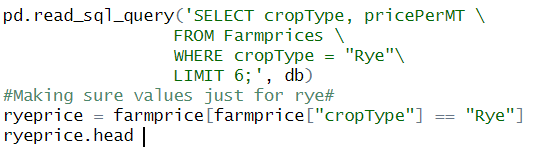
In the code above, I make sure to include the distinct function as I do not want to count the geography characteristics multiple times. After running this code, I determine that the that the provinces included in the farm prices dataset include Alberta and Saskatchewan.

In the next task, I am determining how many hectares of rye were harvested in Canada in 1969. To figure this out, I run the following code in R:

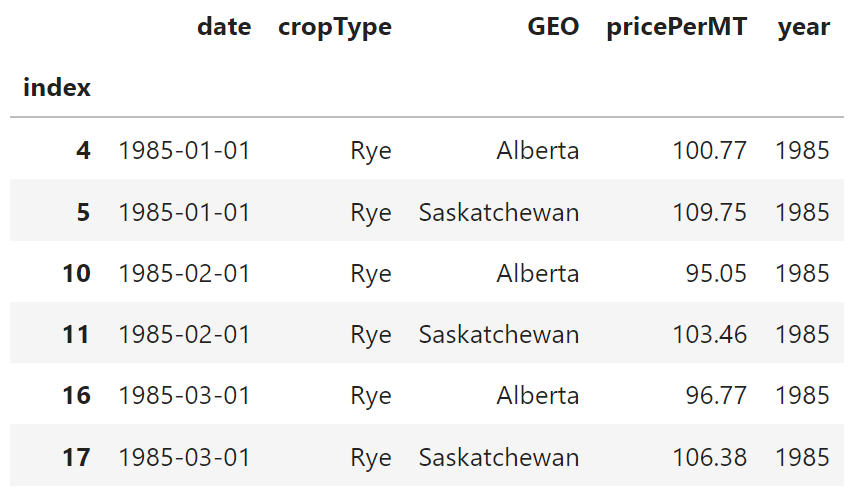


After running the code, I determine that the numbers of hectares of rye that were harvested in Canada in 1969 is 476,100. Likewise, the number of hectares of rye that were harvested in Canada in 1969 with respect to just Saskatchewan and Alberta can be calculated through simple subtraction. This total is 202,000.

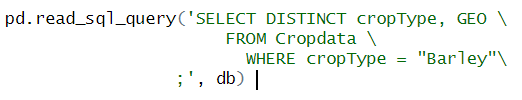
For my next task, I have been asked to query and display the first six rows of the farm prices table for just Rye. To determine this, I run the following code in R:



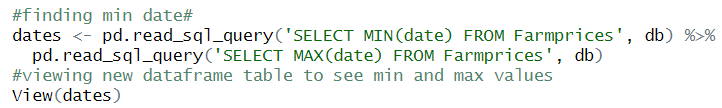
In the code above, it is important to note that I used the == to make sure my values were exactly equal to only rye values. Similarly, the head command automatically shows the first six values. Ultimately, when executing this code, I pull up the following table to answer the original problem or question.



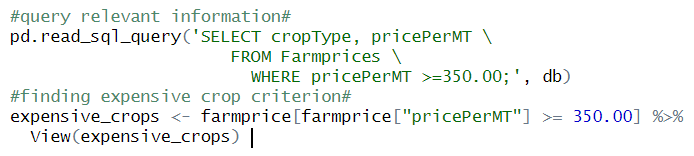
Next, I am asked to determine which provinces grew Barley. To determine this, I run the following code in R:



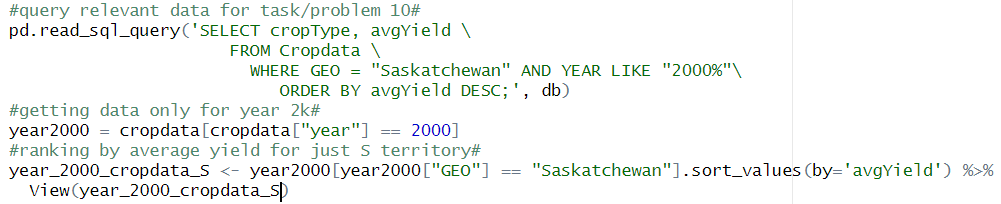
In executing this code, I determine that the provinces which grew barley include Alberta, Saskatchewan, and the rest of greater Canada. Upon doing some supplemental research, apparently, Canada just hosts a great environment to grow barley. For my next task, I am asked to find the first and last dates for the farm prices data. To determine this, I draft and execute the following code in R:



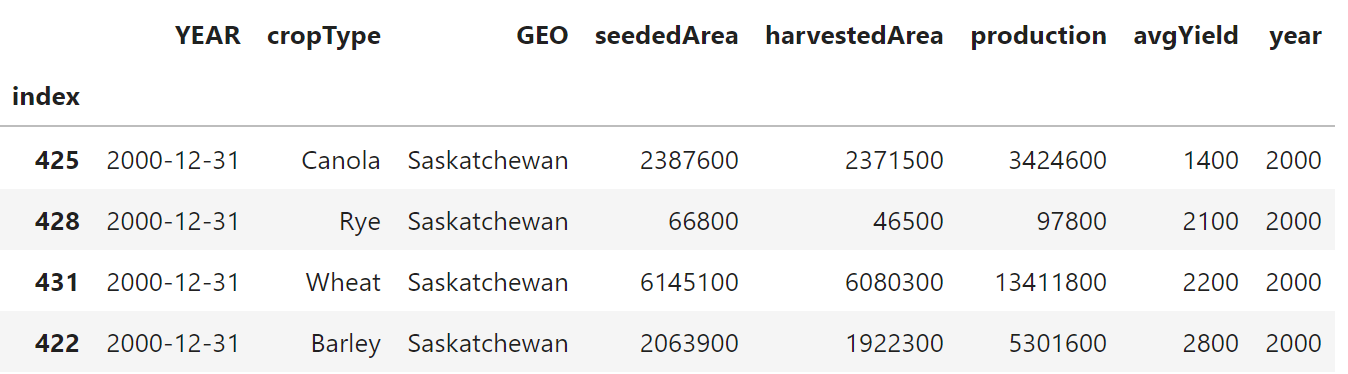
Upon executing this code, I find that the first date in the farm prices data is January 1, 1985. Likewise, I find that the last date in the farm prices data is December 1, 2020. Next, I am tasked to figure out which crops have ever reached a farm price greater than or equal to $350 per metric ton. To determine this, I draft and execute the following code in R:



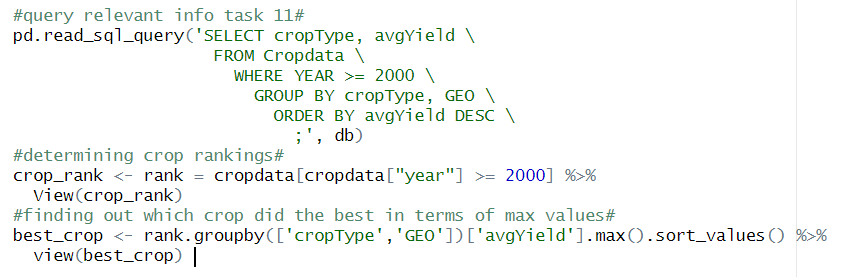
Upon executing this code, I determine that Canola is the only crop that has reached or exceeded $350 per metric ton. For my next task, I have been asked to rank the crop types harvested in Saskatchewan in the year 2000 by their average yield. Similarly, I am to determine which crops and provinces had the highest average yield since the year 2000. To determine this information, I draft and execute the following code in R:



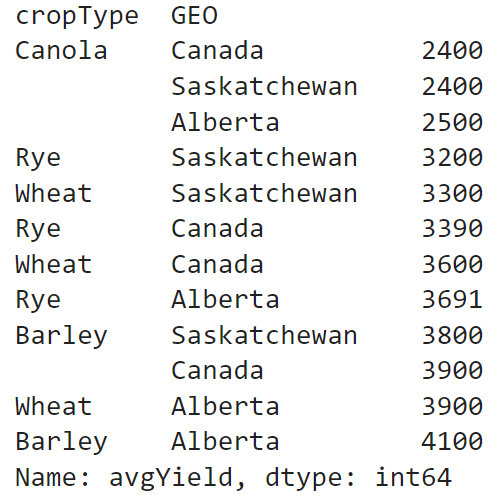
After executing the code above, I view the table of the data frame I created. The output is described below:



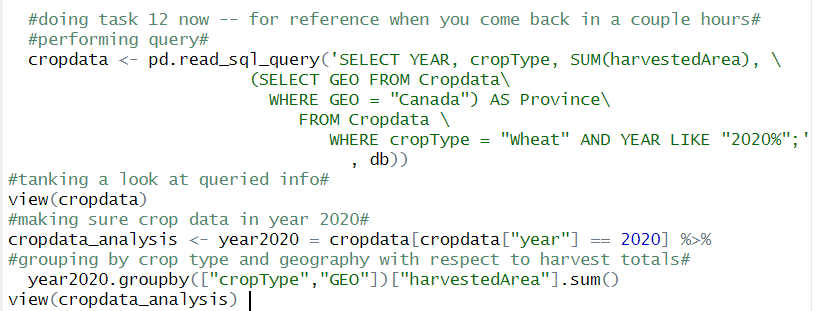
For my next task, I have been asked to rank the crops and geographics by their average yield, measured in KG per hectare, since the year 2000. Likewise, I am to determine which crop and province had the highest average yield since the year 2000. To provide insight to these questions, I draft and execute the following code in R:



After executing the code above, it returns my best crop table, which is pictured below.



According to the table above, Alberta barley performed the best. For my next task, I have asked to use a subquery to determine how much wheat was harvested in Canada in the most recent year of the data. To determine this information, I draft and execute the following code in R:



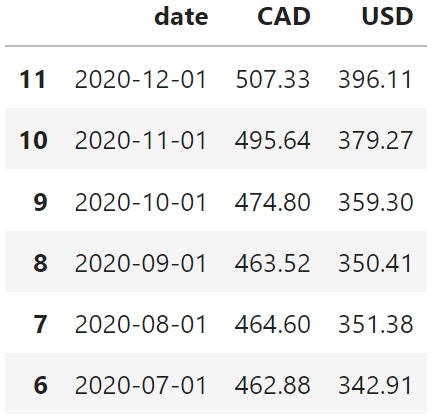
Upon executing the code and viewing the new data frame I created for analysis purpose, it returns the following table:



So, using this table created, I can see how much wheat was harvested in Canada in the most recent year of the data. For my final task, I have been asked to use an implicit inner join to calculate the monthly price per metric ton of canola grown in Saskatchewan in both Canadian and US dollars. Likewise, I am to calculate and display this information with respect to the most recent six months of the data. To perform this task, I draft and execute the following code in R:



Upon executing the code and viewing the final task data table, it prints out the following:



---------------------------------------------------------------------------------------------------------------------

**Conclusion:**

In this case study, I demonstrated my proficiency in R as well as SQL. Similarly, I acted effectively as a supply chain analyst. During my analysis, I performed a variety of tasks to inform the key stakeholders about foreign grain markets. This insight is helpful as the US Venture Capital firm is considering foreign grain markets to help meet its supply chain requirement for its recent investment in the microbrewery and microdistillery industry. Using these key insights I found throughout my analysis, the key stakeholders can be better informed when making an executive decision with respect to supply chain concerns.

---------------------------------------------------------------------------------------------------------------------

**Full Code in R:**

library(tidyverse)

install.packages("RODBC")

library(RODBC)

#Attempt to Create Connection#

db = sq3.connect("sqlite\_data.db")

cursor = db.cursor()

#Checking All The Tables in Dataset#

pd.read\_sql\_query('SELECT \* FROM sqlite\_master', db)

farmprice = pd.read\_sql\_query('SELECT \* FROM Farmprices', db , index\_col="index", parse\_dates='date')

farmprice["year"] = farmprice["date"].dt.year

#Here, I just want to read some values to make sure my code was executed properly#

farmprice.head()

pd.read\_sql\_query('SELECT COUNT(\*) FROM Farmprices', db)

farmprice.shape

pd.read\_sql\_query('SELECT DISTINCT GEO FROM Farmprices', db)

farmprice.GEO.unique()

pd.read\_sql\_query('SELECT YEAR, cropType, SUM(harvestedArea) \

FROM Cropdata \

WHERE cropType="Rye" AND YEAR LIKE "1968%";', db)

cd = cropdata[(cropdata["YEAR"] == "1968-12-31") & (cropdata["cropType"] == "Rye")]

#I want to view new dataframe of new table#

View(cd)

#Summing Values to Get Totals#

cd.harvestedArea.sum()

pd.read\_sql\_query('SELECT cropType, pricePerMT \

FROM Farmprices \

WHERE cropType = "Rye"\

LIMIT 6;', db)

#Making sure values just for rye#

ryeprice = farmprice[farmprice["cropType"] == "Rye"]

ryeprice.head

pd.read\_sql\_query('SELECT DISTINCT cropType, GEO \

FROM Cropdata \

WHERE cropType = "Barley"\

;', db)

#finding min date#

dates <- pd.read\_sql\_query('SELECT MIN(date) FROM Farmprices', db) %>%

pd.read\_sql\_query('SELECT MAX(date) FROM Farmprices', db)

#viewing new dataframe table to see min and max values

View(dates)

#query relevant information#

pd.read\_sql\_query('SELECT cropType, pricePerMT \

FROM Farmprices \

WHERE pricePerMT >=350.00;', db)

#finding expensive crop criterion#

expensive\_crops <- farmprice[farmprice["pricePerMT"] >= 350.00] %>%

View(expensive\_crops)

#query relevant data for task/problem 10#

pd.read\_sql\_query('SELECT cropType, avgYield \

FROM Cropdata \

WHERE GEO = "Saskatchewan" AND YEAR LIKE "2000%"\

ORDER BY avgYield DESC;', db)

#getting data only for year 2k#

year2000 = cropdata[cropdata["year"] == 2000]

#ranking by average yield for just S territory#

year\_2000\_cropdata\_S <- year2000[year2000["GEO"] == "Saskatchewan"].sort\_values(by='avgYield') %>%

View(year\_2000\_cropdata\_S)

#query relevant info task 11#

pd.read\_sql\_query('SELECT cropType, avgYield \

FROM Cropdata \

WHERE YEAR >= 2000 \

GROUP BY cropType, GEO \

ORDER BY avgYield DESC \

;', db)

#determining crop rankings#

crop\_rank <- rank = cropdata[cropdata["year"] >= 2000] %>%

View(crop\_rank)

#finding out which crop did the best in terms of max values#

best\_crop <- rank.groupby(['cropType','GEO'])['avgYield'].max().sort\_values() %>%

view(best\_crop)

#doing task 12 now -- for reference when you come back in a couple hours#

#performing query#

cropdata <- pd.read\_sql\_query('SELECT YEAR, cropType, SUM(harvestedArea), \

(SELECT GEO FROM Cropdata\

WHERE GEO = "Canada") AS Province\

FROM Cropdata \

WHERE cropType = "Wheat" AND YEAR LIKE "2020%";'

, db))

#tanking a look at queried info#

view(cropdata)

#making sure crop data in year 2020#

cropdata\_analysis <- year2020 = cropdata[cropdata["year"] == 2020] %>%

#grouping by crop type and geography with respect to harvest totals#

year2020.groupby(["cropType","GEO"])["harvestedArea"].sum()

view(cropdata\_analysis)

#performing query for days#

dailyfx = pd.read\_sql\_query('SELECT \* FROM Dailyfx', db , index\_col='index', parse\_dates='date')

#checking if data looks alright#

head(dailyfx)

#performing query for months#

monthlyfx = pd.read\_sql\_query('SELECT \* FROM Monthlyfx', db , index\_col='index', parse\_dates='date')

#checking if data looks alright#

head(monthlyfx)

#performing query for dates & currencies#

pd.read\_sql\_query('SELECT a.date, (a.pricePerMT\*1) AS CAD, (a.pricePerMT/b.FXUSDCAD) AS USD \

FROM Farmprices AS a \

INNER JOIN Monthlyfx AS b USING(date) \

WHERE a.GEO = "Saskatchewan" AND a.date LIKE "2020%" AND CropType = "Canola"\

ORDER BY a.date DESC \ ', db)

#performing query for dates, crop type, and prices#

pd.read\_sql\_query('SELECT a.date, a.cropType, a.priceperMT \

FROM Farmprices AS a, Monthlyfx AS b\

WHERE a.GEO = "Saskatchewan" AND a.date=b.date\

GROUP BY a.date\

ORDER BY a.date DESC \ ', db)

#making sure data just focused on canola/canola analaysis #

canola = farmprice[farmprice["cropType"] == "Canola"]

#sorting canola for just S territory3

canolasas = canola[canola["GEO"] == "Saskatchewan"]

#sorting data for just year 2020#

canolasas2020 = canolasas[canolasas["year"] == 2020]

#attempt to join queried tables#

df = pd.merge(left=canolasas2020, right=monthlyfx, on="date", how="inner")

#check if inner join successful#

view(df)

#check -- continuing analysis with respect to currency figures#

#calculating Canadian currency figures#

df["CAD"] = df["pricePerMT"] \* 1.00

#calculating US currency figures

df["USD"] = df["CAD"] / df["FXUSDCAD"]

#creating table for analysis#

final\_task <- df[["date","CAD","USD"]].tail(6).sort\_values(by="date", ascending=False) %>%

view(final\_task)

---------------------------------------------------------------------------------------------------------------------

**Access to the Datasets:**

* <https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMRP0203ENSkillsNetwork890-2022-01-01&pid=3210035901>
* <https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMRP0203ENSkillsNetwork890-2022-01-01&pid=3210007701>
* <https://www.bankofcanada.ca/rates/exchange/daily-exchange-rates/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMRP0203ENSkillsNetwork890-2022-01-01>