Module 2C: Pricing Seasonality and Convenience Yield (Hay and Corn)

# Learning Outcomes

By working this module you should be able to:

* Describe the general trends in price, production and stocks in the U.S. hay market.
* Construct a dummy variable model with an interaction term, and interpret the meaning of the coefficients.
* Demonstrate that the long term average monthly prices in the U.S. hay market roughly conform to the saw-toothed theory of commodity pricing.
* Explain how seasonal pricing patterns are impacted by a reduction in carry over stocks in the U.S. hay market.
* Explain the concept of convenience yield and how this concept is used to enhance the intertemporal LOP.
* Explain how convenience yield can be used to achieve a realistic seasonal pricing pattern in an eight quarter (two year) storage simulation model.
* Conduct “what if” analysis with the simulation model to identify how price levels and pricing patters are impacted by changes in the initial level of inventory.

# Overview of this Module

In the previous module we used a simple model which predicts a saw-toothed pattern of prices over time. The theory tells us that the commodity’s price rises over time while storage is positive, and then rapidly drops when the market stocks out. If this pattern repeats itself for many years then the pricing pattern looks like a tooth side of a saw (i.e., angled up and then sharp down, angled up then sharp down, etc.). In real world commodity markets we should not expect to see an exact saw-toothed pattern because harvest typically does not take place at one defined point in time, and there is regional variation in where the commodity is produced. Nevertheless, we should still expect to see a rough saw-toothed pattern in prices assuming that the market stocks out.

Another important result from the previous module is that if the market does not stock out because stocks are carried from one crop year to the next then price does not drop with the arrival of the new harvest. Rather, when bridging from the old crop year to the new crop year price continues to rise according to unit cost of storage. This particular result leads to an important discrepancy between theory and observation. Specifically, real world markets for long storage commodities such as corn and wheat never fully stock out. As was just noted, the LOP tells us that if the market does not stock out then the commodity’s price must continue to rise. Clearly the price of any commodity cannot rise forever.

In the U.S. corn market, and in the markets for the major grains, stocks carried across years very rarely fall below 10 percent of that year’s production. Some of the carry over is term “pipeline stocks” because moving the commodity through the supply chain takes weeks and sometimes months. We are not interested in these pipeline stocks from an economics perspective because these stocks are largely viewed as exogenous. We are interested in the stocks which are carried from one year to the next in excess of pipeline stocks (i.e., strategic storage).

What we tend to observe in real world markets is that in years with relatively high stocks being carried over then we observe a relatively small (and sometimes zero) downward adjustment in price with the arrival of new harvest. In normal years with a more moderate amount of year over year carry over, we generally see a moderate downward adjustment in price with the arrival of new harvest. It is only when stocks are very low and only a small amount is being carried over across years do we see a large reduction in price with the arrival of new harvest. It is this last case that most closely resembles the saw-toothed pricing pattern which theory predicts happens only when there is a stock out.

This lack of consistency between what the theoretical LOP predicts and what we observe in real markets is an important problem. Economists have “solved” this problem by developing the concept of *convenience yield*. You can think of convenience yield as an implicit negative cost of storage. A merchant who is storing the commodity receives a flow of convenience from having the stocks on hand rather than having to source the commodity in the spot market if an unexpected sales situation arises. The fewer the stocks which are available in the market, the higher the level of convenience from having personal stocks on hand. If the convenience yield is larger than the actual cost of storage then the combined *carrying cost*, which is the sum of the actual cost of storage and the convenience yield, will take on a negative value. With a negative carrying cost the LOP theory predicts that price must decrease from one period to the next. Thus, a convenience yield which varies in size according to the level of stocks in the market can explain why storage is positive across a particular time period even though the price is expected to fall over that time period (e.g., spanning the end of the old crop year and the beginning of the new crop year).

We cannot observe or estimate convenience yield. For this reason we are able to treat it as a *fudge factor*. Specifically, suppose the cost of storing a commodity is $5 per tonne per month. Also suppose that we observe price at the beginning of the month equal to $100 per tonne and we expect price at the end of the month to be $90 per tonne. In a standard storage model the LOP would be violated if storage costs are positive and the level of storage is positive. In the enhanced LOP theory, convenience yield for that month must be $15 per tonne. Why? The net cost of storage which is referred to as the carrying cost is equal to $5 - $15 = -$10 per tonne. The LOP indicates that when storage is positive the expected future price minus the current price must equal the net cost of storage. The LOP is not violated because the price difference, 90 - 100, is indeed equal to the net cost of storage, -10.

With the revised LOP we can conclude that: (1) the monthly convenience yield must be larger than the monthly cost of storage if price is expected to decrease over the coming month; and (2) the convenience yield must be largest when the carry over stocks are the lowest because large price drops with the arrival of harvest are associated with low carry over stocks. In those cases where carry over stocks are large and the price does not fall with the arrival of a new harvest we can infer that the carrying cost is positive since the monthly convenience yield must be smaller than the monthly cost of storage.

The purpose of this module is to incorporate convenience yield into our theory of the intertemporal LOP. We use the enhanced theory to construct a pricing simulation model that allows for price decreases over time when storage is positive. The simulation model is calibrated to the U.S. corn market. This market was chosen because over the next several modules we will utilize the U.S corn market repeatedly to study the theory of commodity futures prices. When studying futures markets in these subsequent modules the concept of convenience yield is shown to play a very important role.

However, before we incorporate convenience yield into our LOP model and build the simulation model we should spend some time examining the saw-toothed pricing pattern in real world commodity prices. For this examination to be effective we need a commodity which has monthly prices and an observable stock level over a large number of years. Observing both price and stocks is important because this will allow us to examine how the saw-toothed pricing pattern is different for normal stock carry over years versus low stock carry over years. The commodity chosen for this exercise is the U.S. hay market because it ticks all of the above boxes. The next section provides a brief introduction to the U.S. hay market. Following this is an econometric examination of prices in the U.S. hay market.

# Saw-Toothed Pricing in the U.S. Hay Market

## Industry background

According to this 2018 [report](https://farmdocdaily.illinois.edu/2018/09/us-hay-market-over-the-last-100-years.html) hay is the third largest crop that is grown in the U.S. Between 1955 and 2005 alfalfa production exceeded the production of all other types of hay (e.g., sweet clover and brome grass). For the past 15 years this ranking between alfalfa and other types of hay has switched. Harvested acres of both alfalfa and other types of have been in a steady decline since the mid 1950s. Higher yields for both alfalfa and other types of hay have offset part of the decline in acreage but not by enough to stop an on-going decline in overall hay production.

Hay is used to feed livestock, especially during the months when livestock grazing is not feasible (e.g., late fall, winter and early spring). As livestock production gradually became more industrialized, livestock was increasingly fed year round in a feedlot rather than allowed to graze during the growing season and fed hay during the non-growing season. In a feedlot environment hay is fed along with high energy feeds such as corn, barley and oats. According to the above report an important reason for the decline in hay acres is that farmers find it more profitable to grow corn and other crops. This is problematic because from an environmental perspective hay is far superior to corn and most other crops, especially on light and easily eroded land. The U.S. Conservation Reserve Program (CRP) within which farmers are paid to not grow crops on environmentally sensitive land, currently has 5.3 million enrolled acres. Of these acres about half is devoted to hay production (see [here](https://www.fsa.usda.gov/programs-and-services/conservation-programs/conservation-reserve-program/)).

## Data on U.S. Hay Prices, Production and Stocks

Price, production and stock data for hay were extracted from the USDA Feed Grains [Database](https://www.ers.usda.gov/data-products/feed-grains-database/). Price data is monthly, production data is yearly and stocks data is bi-yearly (May and December). The data which spans from May of 1950 to July of 2021 consists of alfalfa and all other types of hay. Because of the large number of years it is important to account for on-going hay price inflation. One option is to do nothing, in which case the inflationary trend will end up in the intercept of the model when estimated in first differences. The second option is to remove inflation by dividing each monthly hay price by the monthly U.S. consumer price index (CPI). For this analysis where we will use dummies to isolate seasonality, the latter approach is best. This is because the seasonal effects will increase over time due to general inflation and a straight first difference specification is not sufficient to adjust the seasonal effects for inflation.

The monthly CPI used for deflating hay prices was downloaded from the U.S. Federal Reserve ([FRED](https://fred.stlouisfed.org/series/CPIAUCSL)) website. The base year for this CPI data is 1982. This means that the hay price for 1982 is the same with and without adjusting for inflation. As well, CPI adjusted hay prices for the years before (after) 1982 will be higher (lower) than the original prices. To give you a sense of the size of the CPI adjustment, the hay price is $22 per ton in the original data and $92.55 per ton in the CPI adjusted data. As well, the hay price is $185 per ton in the original data and $66.06 per ton in the CPI adjusted data.

After coding the standard preliminaries we can read the data into R in the usual way.

rm(list = ls())   
graphics.off()  
pacman::p\_load(here, dplyr, ggplot2)  
hay\_data <- read.csv(here("Data", "usda\_hay.csv"), header=TRUE, sep=",", stringsAsFactors = FALSE)   
head(hay\_data)

## year month price stckMay stckDec production  
## 1 1950 May 92.55364 14599 67676 103820  
## 2 1950 Jun 87.10218 14599 67676 103820  
## 3 1950 Jul 82.26007 14599 67676 103820  
## 4 1950 Aug 83.47107 14599 67676 103820  
## 5 1950 Sep 83.40181 14599 67676 103820  
## 6 1950 Oct 84.08163 14599 67676 103820

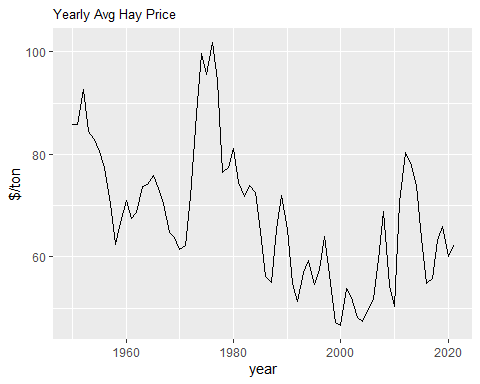
The first few rows of the imported data shows that the annual production data repeats for each of the 12 months. As well, there is a column for May stocks and another column for December stocks, both of which also repeat for each of the 12 months.

There is too much data to plot monthly prices, and so prices are averaged into annual data for the purpose of graphing. The next chunk of code deletes the *month* column in the imported data and then performs the averaging for each of the columns.

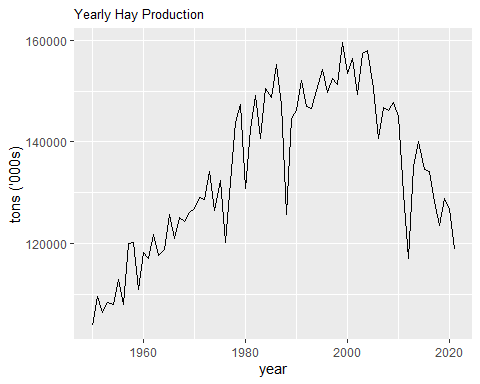
hay\_data2 <- select(hay\_data, -"month")  
  
mean\_data <- hay\_data2 %>%  
 group\_by(year) %>%  
 summarise\_all(list(mean))

We can now plot average annual price, annual production, annual May stocks and Annual December stocks.

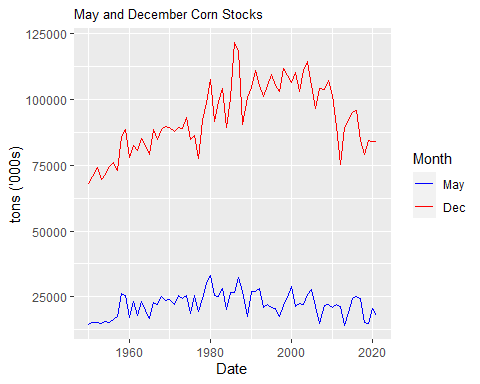
plot\_price <- ggplot(mean\_data, aes(x = year, y = price)) +   
 geom\_line() +   
 labs(title = "Yearly Avg Hay Price", y= "$/ton") +   
 theme(plot.title = element\_text(size=10))  
plot\_price



plot\_production <- ggplot(mean\_data, aes(x = year, y = production)) +   
 geom\_line() +   
 labs(title = "Yearly Hay Production", y= "tons ('000s)") +   
 theme(plot.title = element\_text(size=10))  
plot\_production



plot\_stocks <- ggplot(mean\_data, aes(x = year)) +   
 geom\_line(aes(y = stckMay, color = "May")) +   
 geom\_line(aes(y = stckDec, color = "Dec")) +   
 labs(title = "May and December Corn Stocks", y = "tons ('000s)", x = "Date") +  
 theme(plot.title = element\_text(size=10)) +  
 scale\_color\_manual(name = "Month", values = c("May" = "blue", "Dec" = "red"))  
plot\_stocks



The first graph shows the inflated-adjusted (real) price of hay beginning in 1950. Notice that the longer term pricing pattern is similar to the longer term pricing pattern of the major crops. Specifically, the price of hay rose steadily throughout the inflationary period of the 1970s, and then began a long term decline throughout the 1980s and 1990s. For the past two decades prices have strengthened but at a relatively slow pace.

The second graph shows annual production. This graph is consistent with the discussion about hay production in the previous section. Up until about the year 2000 increasing yields more than offset the declining harvested acres, and so overall production increased. After 2000 the yield increase did not keep pace and hay production dropped and continues to decline. The production graph shows large reductions in production in the late 1980s and around 2012. These sharp reductions are very likely the result of a severe drought rather than a sharp drop in hay acres. This claim can be investigated by reading about major U.S. droughts on [Wikipedia](https://en.wikipedia.org/wiki/Droughts_in_the_United_States).

The third graph shows the annual May and December hay stocks. The top schedule shows December stocks, which corresponds to the beginning of the livestock feeding season. The bottom schedule shows May stocks, which corresponds to the end of the livestock feeding season. The rising December stocks up until the year 2000 reflects the rising production levels over this period of time. Conversely, the falling December stocks reflect the falling production levels after 2000.

Hay is storable but the cost of storing it for more than one season is relatively high. This is because hay must be covered for long term storage to prevent excessive deterioration from rain. It is likely for this reason that the May stocks do not follow the same trends as the December stocks. The variation in May stocks is relatively low but as will be shown below the variation is large enough to identify very different seasonal pricing patterns for a low stock carry over year versus a normal stock carry over year.

The May stocks play an important role in the analysis and so it is useful to examine the summary statistics.

summary(mean\_data$stckMay)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 14156 18384 22081 22071 25358 33192

We see that the average level of stocks is about 22 million tons (keep in mind that the units of measure for the Q variables are thousands of tons). The previous production chart shows that in recent years production was about 120 million tonnes. This means that May stocks constitute about 18 percent of annual production. We can’t say for sure that the full 18 percent is carried over to the next crop year but 18 percent provides a reasonable estimate. This is because by May most livestock are not longer fed hay and new hay production is near at hand.

An 18 percent carry over rate is similar to the stock carry over rate for the major U.S. grains. The previous table shows that the first quartile is about 18 million tons. This means that 25 percent of the May stock observations were below 18 million tons. In the analysis below 18 million is used as the threshold for defining a “low stock” year. Even though 18 million is not much below the average of 22 million, we will see that there is a large difference in the seasonal pattern of prices for a normal year versus a low stock year.

## Dummy Variables with Interaction Terms

In our study of the U.S. potato CPI in Module 1E we used a set of 11 monthly dummies (December was omitted and serves as the reference point) for measuring seasonality in potato prices. We will do the same thing here except now we will use conditional monthly dummies in additional to the regular unconditional dummies. Specifically, we will create an indicator variable which takes on a value of one if May stocks are low (i.e., less than the first quartile, which is approximately 18 million tons) and a value of zero otherwise. We will then create a new set of 11 interaction variables, where each variable is the product of the regular monthly dummy and the indicator variable. This will become clearer with a formal specification of the enhanced model.

The enhanced dummy variable model can be written in levels as

In this equation the variable is an indicator variable which takes on a value of 1 if year stocks are low and a value of 0 otherwise.

The Appendix provides a detailed discussion about how to interpret the estimated coefficients of a dummy variable model with interaction terms. By way of a summary, is the long run average price in December of a normal year, and is the long average price in December of a low-stock year. Similarly, is the long run average price in month of a normal year, and is the long average price in month of a low stock year.

Let and denote the long average price in month in a normal stock year and a low stock year, respectively. Using the expressions from above it can be shown that

and

or

These equations tell us that is the difference in the long run average price in month and December, both in a normal year. Similarly, can be interpreted as the as the difference in the low versus normal stocks price gap where the gap is defined as the long run average price difference between month and December. Using the last equation, can also be interpreted as the month to month difference in the price gap where the gap is the long run average price difference for low versus normal stocks.

The previous plot reveals a high degree of non-stationarity in the price of hay. It is therefore necessary to estimate the model in first differences as follows:

It is important to understand that even though this equation is estimated in first difference format, the interpretation of the estimated coefficients remain the same as the case where the model was estimated in levels. See the Appendix for a more detailed discussion about estimating the model in first differences.

## Building the Data Set

To further construct our data set we need to create the low stock carry over indicator variable (*StckDum*). To do this we need to identify the first quartile of the May stock variable over the 1950 - 2021 sample period. We can then create a new column in the monthly data set within which the indicator takes on a value of one if the current level of May stocks is below the first quartile of long term annual May stocks and zero otherwise.

The first step is use the *quantile()* function and the May stocks column of the annual data which was used for graphing (i.e., *mean\_data*) to determine the lower threshold for defining low stocks

(quant\_cut = quantile(mean\_data$stckMay, 0.25))

## 25%   
## 18384

We can now use the *quant\_cut* variable to create the indicator variable in the original monthly data frame, *hay\_data*: 1 if May stocks are less than the threshold and zero otherwise. The new *stckDum* column is added to *mean\_data* as follows:

hay\_data <- hay\_data %>%  
 mutate(stckDum = ifelse(stckMay<quant\_cut,1,0) )  
head(hay\_data)

## year month price stckMay stckDec production stckDum  
## 1 1950 May 92.55364 14599 67676 103820 1  
## 2 1950 Jun 87.10218 14599 67676 103820 1  
## 3 1950 Jul 82.26007 14599 67676 103820 1  
## 4 1950 Aug 83.47107 14599 67676 103820 1  
## 5 1950 Sep 83.40181 14599 67676 103820 1  
## 6 1950 Oct 84.08163 14599 67676 103820 1

The procedure for adding monthly dummies to the current data frame is somewhat tedious but it is easy to understand:

hay\_data <- hay\_data %>%   
 mutate(d1 = ifelse(month=="Jan",1,0),   
 d2 = ifelse(month=="Feb",1,0),  
 d3 = ifelse(month=="Mar",1,0),  
 d4 = ifelse(month=="Apr",1,0),  
 d5 = ifelse(month=="May",1,0),  
 d6 = ifelse(month=="Jun",1,0),  
 d7 = ifelse(month=="Jul",1,0),  
 d8 = ifelse(month=="Aug",1,0),  
 d9 = ifelse(month=="Sep",1,0),  
 d10 = ifelse(month=="Oct",1,0),  
 d11 = ifelse(month=="Nov",1,0))  
head(hay\_data, 15)

## year month price stckMay stckDec production stckDum d1 d2 d3 d4 d5 d6 d7  
## 1 1950 May 92.55364 14599 67676 103820 1 0 0 0 0 1 0 0  
## 2 1950 Jun 87.10218 14599 67676 103820 1 0 0 0 0 0 1 0  
## 3 1950 Jul 82.26007 14599 67676 103820 1 0 0 0 0 0 0 1  
## 4 1950 Aug 83.47107 14599 67676 103820 1 0 0 0 0 0 0 0  
## 5 1950 Sep 83.40181 14599 67676 103820 1 0 0 0 0 0 0 0  
## 6 1950 Oct 84.08163 14599 67676 103820 1 0 0 0 0 0 0 0  
## 7 1950 Nov 86.17886 14599 67676 103820 1 0 0 0 0 0 0 0  
## 8 1950 Dec 87.26982 14599 67676 103820 1 0 0 0 0 0 0 0  
## 9 1951 Jan 89.04649 15232 70657 109502 1 1 0 0 0 0 0 0  
## 10 1951 Feb 89.81804 15232 70657 109502 1 0 1 0 0 0 0 0  
## 11 1951 Mar 89.25811 15232 70657 109502 1 0 0 1 0 0 0 0  
## 12 1951 Apr 89.12037 15232 70657 109502 1 0 0 0 1 0 0 0  
## 13 1951 May 88.11081 15232 70657 109502 1 0 0 0 0 1 0 0  
## 14 1951 Jun 83.30120 15232 70657 109502 1 0 0 0 0 0 1 0  
## 15 1951 Jul 77.96218 15232 70657 109502 1 0 0 0 0 0 0 1  
## d8 d9 d10 d11  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 1 0 0 0  
## 5 0 1 0 0  
## 6 0 0 1 0  
## 7 0 0 0 1  
## 8 0 0 0 0  
## 9 0 0 0 0  
## 10 0 0 0 0  
## 11 0 0 0 0  
## 12 0 0 0 0  
## 13 0 0 0 0  
## 14 0 0 0 0  
## 15 0 0 0 0

In a similar way the interaction variables can be created and added to the data frame:

hay\_data <- hay\_data %>%   
 mutate(i0 = stckDum,   
 i1 = d1\*stckDum,  
 i2 = d2\*stckDum,  
 i3 = d3\*stckDum,  
 i4 = d4\*stckDum,  
 i5 = d5\*stckDum,  
 i6 = d6\*stckDum,  
 i7 = d7\*stckDum,  
 i8 = d8\*stckDum,  
 i9 = d9\*stckDum,  
 i10 = d10\*stckDum,  
 i11 = d11\*stckDum)  
head(hay\_data, 15)

## year month price stckMay stckDec production stckDum d1 d2 d3 d4 d5 d6 d7  
## 1 1950 May 92.55364 14599 67676 103820 1 0 0 0 0 1 0 0  
## 2 1950 Jun 87.10218 14599 67676 103820 1 0 0 0 0 0 1 0  
## 3 1950 Jul 82.26007 14599 67676 103820 1 0 0 0 0 0 0 1  
## 4 1950 Aug 83.47107 14599 67676 103820 1 0 0 0 0 0 0 0  
## 5 1950 Sep 83.40181 14599 67676 103820 1 0 0 0 0 0 0 0  
## 6 1950 Oct 84.08163 14599 67676 103820 1 0 0 0 0 0 0 0  
## 7 1950 Nov 86.17886 14599 67676 103820 1 0 0 0 0 0 0 0  
## 8 1950 Dec 87.26982 14599 67676 103820 1 0 0 0 0 0 0 0  
## 9 1951 Jan 89.04649 15232 70657 109502 1 1 0 0 0 0 0 0  
## 10 1951 Feb 89.81804 15232 70657 109502 1 0 1 0 0 0 0 0  
## 11 1951 Mar 89.25811 15232 70657 109502 1 0 0 1 0 0 0 0  
## 12 1951 Apr 89.12037 15232 70657 109502 1 0 0 0 1 0 0 0  
## 13 1951 May 88.11081 15232 70657 109502 1 0 0 0 0 1 0 0  
## 14 1951 Jun 83.30120 15232 70657 109502 1 0 0 0 0 0 1 0  
## 15 1951 Jul 77.96218 15232 70657 109502 1 0 0 0 0 0 0 1  
## d8 d9 d10 d11 i0 i1 i2 i3 i4 i5 i6 i7 i8 i9 i10 i11  
## 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0  
## 2 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0  
## 3 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0  
## 4 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0  
## 5 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0  
## 6 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0  
## 7 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1  
## 8 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0  
## 9 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0  
## 10 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0  
## 11 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0  
## 12 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0  
## 13 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0  
## 14 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0  
## 15 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0

The next step is to add the first differences of all variables to the current data frame. At the end of using the *mutate()* function to add the differenced variable, the top row is removed because there is no useable data in the first row.

diff\_data <- hay\_data %>%  
 mutate(price\_diff = price - lag(price),   
 d1Diff = d1 - lag(d1),   
 d2Diff = d2 - lag(d2),   
 d3Diff = d3 - lag(d3),  
 d4Diff = d4 - lag(d4),  
 d5Diff = d5 - lag(d5),  
 d6Diff = d6 - lag(d6),  
 d7Diff = d7 - lag(d7),  
 d8Diff = d8 - lag(d8),  
 d9Diff = d9 - lag(d9),  
 d10Diff = d10 - lag(d10),  
 d11Diff = d11 - lag(d11),  
 i0Diff = i0 - lag(i0),  
 i1Diff = i1 - lag(i1),  
 i2Diff = i2 - lag(i2),  
 i3Diff = i3 - lag(i3),  
 i4Diff = i4 - lag(i4),  
 i5Diff = i5 - lag(i5),  
 i6Diff = i6 - lag(i6),  
 i7Diff = i7 - lag(i7),  
 i8Diff = i8 - lag(i8),  
 i9Diff = i9 - lag(i9),  
 i10Diff = i10 - lag(i10),  
 i11Diff = i11 - lag(i11)) %>%  
 slice(-1) #drops first row

## Estimated Model Without Interaction Variables

We will begin our econometric analysis by estimating the simple version of the model (i.e., the one without the interaction variables). We don’t want to estimate an intercept and so we add a 0 to the end of our variable list. The coefficient estimates are stored in a vector for the purpose of graphical analysis.

regP\_diff <- lm(price\_diff ~ d1Diff + d2Diff + d3Diff + d4Diff + d5Diff + d6Diff + d7Diff + d8Diff + d9Diff + d10Diff + d11Diff + 0, data = diff\_data)  
summary(regP\_diff)

##   
## Call:  
## lm(formula = price\_diff ~ d1Diff + d2Diff + d3Diff + d4Diff +   
## d5Diff + d6Diff + d7Diff + d8Diff + d9Diff + d10Diff + d11Diff +   
## 0, data = diff\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.1557 -1.3362 -0.0353 1.1572 18.7633   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## d1Diff 0.5385 0.2880 1.870 0.06183 .   
## d2Diff 1.1569 0.3883 2.980 0.00297 \*\*   
## d3Diff 1.1586 0.4511 2.568 0.01039 \*   
## d4Diff 2.4256 0.4911 4.940 9.44e-07 \*\*\*  
## d5Diff 4.8592 0.5135 9.463 < 2e-16 \*\*\*  
## d6Diff 0.2507 0.5207 0.481 0.63036   
## d7Diff -1.6049 0.5135 -3.126 0.00184 \*\*   
## d8Diff -1.4315 0.4911 -2.915 0.00365 \*\*   
## d9Diff -0.8339 0.4511 -1.849 0.06487 .   
## d10Diff -0.1768 0.3883 -0.455 0.64904   
## d11Diff -0.5174 0.2880 -1.797 0.07274 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.535 on 842 degrees of freedom  
## Multiple R-squared: 0.311, Adjusted R-squared: 0.302   
## F-statistic: 34.55 on 11 and 842 DF, p-value: < 2.2e-16

matrix\_coef1 <- summary(regP\_diff)$coefficients  
coeff1 <- as.data.frame(matrix\_coef1[,1])  
colnames(coeff1) <- "dum"  
coeff1

## dum  
## d1Diff 0.5385143  
## d2Diff 1.1569003  
## d3Diff 1.1586057  
## d4Diff 2.4256484  
## d5Diff 4.8591714  
## d6Diff 0.2506579  
## d7Diff -1.6049427  
## d8Diff -1.4315036  
## d9Diff -0.8339005  
## d10Diff -0.1767678  
## d11Diff -0.5174031

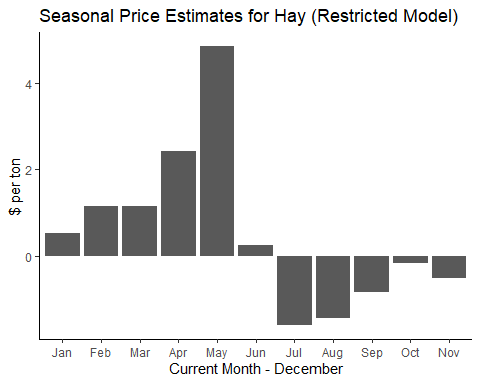
The regression results reveal strong pricing seasonality. Notice that all of the month dummies are statistically significant except for June and October. The value of 0.302 indicates that 30 percent of the monthly pricing variation is due to recurring seasonal effects. A more robust model which allowed pricing seasonality to change over time is likely to have an even higher value. In other words, the current model assumes that the seasonal price differences are the same in 2021 as they were in 1950, and this is a rather strong assumption.

Lets graph the estimated coefficients for the 11 dummy variables. To do this we need to first create a set of X axis labels and bind them to our set of coefficient estimates. The labels are R factor variables, and R insists on ordering them alphabetically rather than they way they are listed. To prevent this from happening we use the *levels* function within the *factor* function as follows.

label <- factor(c("Jan","Feb","Mar", "Apr","May","Jun","Jul","Aug","Sep","Oct","Nov"),  
 levels = c("Jan","Feb","Mar", "Apr","May","Jun","Jul","Aug","Sep","Oct","Nov"))  
  
coeff1 <- cbind(coeff1, label)

Now we can build the column graph.

plot1 <- ggplot(coeff1, aes(x=label, y=dum)) +  
 geom\_bar(stat = "identity") +  
 theme\_classic() +  
 labs(title = "Seasonal Price Estimates for Hay (Restricted Model)",  
 x = "Current Month - December",  
 y = "$ per ton")  
  
plot1



In this graph we can interpret the height of the column for month as the long term average difference in the month price and the December price with no restrictions on stock carry over. The full set of prices roughly conform to the saw-toothed pricing pattern, which was emphasized in the previous module. Storage of hay begins in July and the price increases gradually through the hay storage season (August to November) and the livestock feeding season (December to April). When the hay harvest begins the following June, the price drops sharply with the arrival of new stocks. This is precisely what is predicted by the saw-toothed theory of intertemporal prices for a storable commodity which is produced just once each year. If the intertemporal LOP holds then the rate of increase in the prices as shown in the previous plot is our estimate of the monthly cost of storing hay.

## Estimated Model with Interaction Terms

Let’s re-estimate the model but this time with the interaction variables included. After estimating the model we will separately graph the estimated dummy coefficients and the sum of the estimated dummy coefficients and the interaction variable coefficients so we can more effectively see how low stocks affects the seasonality in hay prices.

We begin by estimating the unrestricted model and saving the estimated coefficients.

regP\_diff2 <- lm(price\_diff ~ d1Diff + d2Diff + d3Diff + d4Diff + d5Diff + d6Diff + d7Diff + d8Diff + d9Diff + d10Diff + d11Diff   
 + i0Diff + i1Diff + i2Diff + i3Diff + i4Diff + i5Diff + i6Diff + i7Diff + i8Diff + i9Diff + i10Diff + i11Diff + 0, data = diff\_data)  
summary(regP\_diff2)

##   
## Call:  
## lm(formula = price\_diff ~ d1Diff + d2Diff + d3Diff + d4Diff +   
## d5Diff + d6Diff + d7Diff + d8Diff + d9Diff + d10Diff + d11Diff +   
## i0Diff + i1Diff + i2Diff + i3Diff + i4Diff + i5Diff + i6Diff +   
## i7Diff + i8Diff + i9Diff + i10Diff + i11Diff + 0, data = diff\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.1696 -1.2480 -0.0491 1.1833 17.7121   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## d1Diff 0.233324 0.338078 0.690 0.490293   
## d2Diff 0.908008 0.443149 2.049 0.040777 \*   
## d3Diff 0.853178 0.509873 1.673 0.094642 .   
## d4Diff 2.254689 0.552327 4.082 4.89e-05 \*\*\*  
## d5Diff 5.739406 0.575903 9.966 < 2e-16 \*\*\*  
## d6Diff 1.144830 0.582896 1.964 0.049859 \*   
## d7Diff -0.397481 0.573914 -0.693 0.488768   
## d8Diff -0.513621 0.548172 -0.937 0.349046   
## d9Diff -0.144423 0.503102 -0.287 0.774134   
## d10Diff 0.473451 0.432708 1.094 0.274203   
## d11Diff -0.321322 0.320726 -1.002 0.316705   
## i0Diff -0.003416 0.702738 -0.005 0.996122   
## i1Diff 1.274411 0.793732 1.606 0.108743   
## i2Diff 1.039080 0.972508 1.068 0.285626   
## i3Diff 1.274992 1.089296 1.170 0.242147   
## i4Diff 0.713184 1.162924 0.613 0.539867   
## i5Diff -3.677304 1.201353 -3.061 0.002277 \*\*   
## i6Diff -3.733248 1.208298 -3.090 0.002071 \*\*   
## i7Diff -5.041900 1.184936 -4.255 2.33e-05 \*\*\*  
## i8Diff -3.832685 1.128360 -3.397 0.000715 \*\*\*  
## i9Diff -2.878965 1.033127 -2.787 0.005447 \*\*   
## i10Diff -2.715207 0.886868 -3.062 0.002273 \*\*   
## i11Diff -0.818722 0.656317 -1.247 0.212585   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.461 on 830 degrees of freedom  
## Multiple R-squared: 0.3599, Adjusted R-squared: 0.3422   
## F-statistic: 20.29 on 23 and 830 DF, p-value: < 2.2e-16

matrix\_coef2 <- summary(regP\_diff2)$coefficients  
coeff2 <- as.data.frame(matrix\_coef2[,1])  
coeff2

## matrix\_coef2[, 1]  
## d1Diff 0.233324384  
## d2Diff 0.908007774  
## d3Diff 0.853178052  
## d4Diff 2.254688748  
## d5Diff 5.739405990  
## d6Diff 1.144829986  
## d7Diff -0.397480719  
## d8Diff -0.513621220  
## d9Diff -0.144422919  
## d10Diff 0.473450755  
## d11Diff -0.321321818  
## i0Diff -0.003416271  
## i1Diff 1.274410767  
## i2Diff 1.039080422  
## i3Diff 1.274991942  
## i4Diff 0.713184129  
## i5Diff -3.677303517  
## i6Diff -3.733248031  
## i7Diff -5.041899923  
## i8Diff -3.832684913  
## i9Diff -2.878964897  
## i10Diff -2.715206636  
## i11Diff -0.818721647

For this unrestricted regression there are fewer statistically significant coefficients on the 11 dummies (now only February through June are significant). However, the estimated coefficients for the May through October interaction variables are highly statistically significant. Notice also that the value has risen from 0.302 in the restricted model to 0.359 in the current unrestricted model. A F test could be used to determine if the inclusion of the interaction terms is statistically significant as a whole. Based on the increase in the it is very likely that the interaction variables as a whole are statistically important.

We would like to have the estimated coefficients for the 11 dummies and the 11 interaction variables in separate columns rather than one long column. We can filter out the estimated dummy coefficients and the interaction variable coefficients and then add names to the columns as follows:

coeff2\_A <- coeff2 %>% slice(1:11)  
coeff2\_A <- coeff2\_A   
colnames(coeff2\_A) <- "dum"  
  
coeff2\_B <- coeff2 %>% slice(12:22)  
colnames(coeff2\_B) <- "inter"

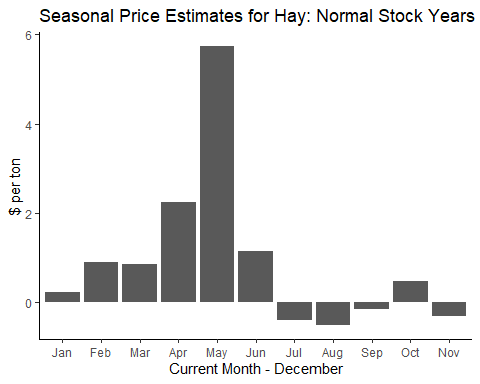
Let’s bind the two sets of estimates into a new data frame. Within this new data frame let’s create a new column called *dum\_plus\_inter* which is the sum of the dummy variable and the interaction term (i.e., ).

coeff2 <- cbind(coeff2\_A,coeff2\_B)  
  
coeff2 <- coeff2 %>%   
 mutate(dum\_plus\_inter = dum + inter )   
coeff2

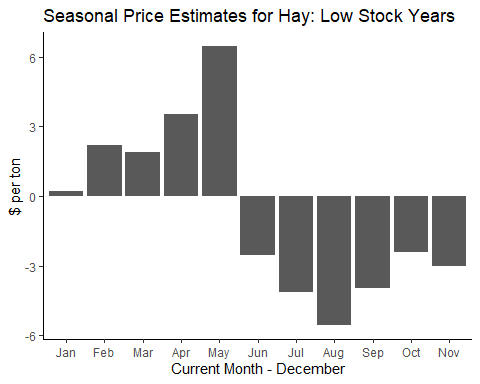
## dum inter dum\_plus\_inter  
## d1Diff 0.2333244 -0.003416271 0.2299081  
## d2Diff 0.9080078 1.274410767 2.1824185  
## d3Diff 0.8531781 1.039080422 1.8922585  
## d4Diff 2.2546887 1.274991942 3.5296807  
## d5Diff 5.7394060 0.713184129 6.4525901  
## d6Diff 1.1448300 -3.677303517 -2.5324735  
## d7Diff -0.3974807 -3.733248031 -4.1307288  
## d8Diff -0.5136212 -5.041899923 -5.5555211  
## d9Diff -0.1444229 -3.832684913 -3.9771078  
## d10Diff 0.4734508 -2.878964897 -2.4055141  
## d11Diff -0.3213218 -2.715206636 -3.0365285

We can now create the pair of graphs for the unrestricted model. The first graph will show the estimated coefficients for the dummy variables, which reflects seasonality in a normal year. The second graph will show the sum of the estimated coefficients for the dummy variables and the interaction variables. This latter graph reflects seasonality in a low stock year. Keep in mind that the first graph will be different from the previous graph, which shows the pricing seasonality in all years.

plot2A <- ggplot(coeff2, aes(x=label, y=dum)) +  
 geom\_bar(stat = "identity") +  
 theme\_classic() +  
 labs(title = "Seasonal Price Estimates for Hay: Normal Stock Years",  
 x = "Current Month - December",  
 y = "$ per ton")  
  
plot2A



plot2B <- ggplot(coeff2, aes(x=label, y=dum\_plus\_inter)) +  
 geom\_bar(stat = "identity") +  
 theme\_classic() +  
 labs(title = "Seasonal Price Estimates for Hay: Low Stock Years",  
 x = "Current Month - December",  
 y = "$ per ton")  
  
plot2B



Both graphs have the same general saw-toothed properties of the previous graph. Interestingly, the price rise during the August to November storage season is much lower with normal stocks than with low stocks. The opposite result was expected because the monthly cost of storage should be higher with the higher stocks. One possibility is that the actual cost of storing hay is quite low since hay bales are often stacked in a field and covered with plastic. Perhaps the price rises much more quickly in a low stock year because the market is sending a signal that more hay should be stored until the following spring in case the feeding season is longer than expected when stocks are low.

Another important difference between the two graphs is that in a normal year, the price drop with the onset of a new hay harvest returns the price to approximately the same level that the price cycle started at in the previous year, and price drop is over relatively quickly (most of the price drop takes place in June and July), With low stocks, the price drop is much more sread out and pronounced, starting in June and going through to August. Why is hay being store when the price is falling over a several month period? This is where the theory of convenience yield comes into play. when stocks are low very little is carried over because convenience yield is high. As new stocks become available the price continues to drop until September, and only then does the price begin to rise. I expect you are feeling the frustration of many economists who feel that the current theory of pricing does a poor job predicting seasonal pricing patterns and explaining comparisons such as what is being done here.

# A Formal Model of Convenience Yield

Imagine in the days prior to debit and credit cards your parents’ decision regarding how much money to withdraw from the bank in order to pay for their purchases for the coming week. It is likely that they would take out more money than what they expected to use in case they needed to make an unexpected purchase. By withdrawing an amount larger than what they expected to use, they were willing to give up the interest earnings in order to have the convenience of cash on hand when an unexpected purchase is optimal. The more inconvenient it is to obtain the additional cash, the greater the convenience yield they received from having the extra cash on hand.

The situation is similar for merchants and processors. They will want to keep more inventory on hand than what they expect to use. The implicit benefit that merchants and processors receive from having inventory on hand rather than having to buy it in the spot market is called *convenience yield*. Convenience yield, which is equivalent to a negative cost of storage, is larger when stocks are scarce and smaller when stocks are plentiful. This makes sense because when stocks are scare it is more costly to search and find stocks in the spot market when an unexpected order comes along.

## Combining Convenience Yield and Storage Costs

Our goal is to create an integrated model of storage costs and convenience yield. In the previous module the marginal physical cost of a unit of the stored commodity was assumed to be constant at level , and the capital cost of storage was assumed to equal where is the rate of interest and is the commodity’s price. The rate of interest is currently very low and so we will simplify by assuming . However, the assumption that the unit cost of storage does not depend on aggregate stocks in the market is not very realistic. We expect a higher unit cost storage with higher stocks because congestion from the excess stocks will require high-cost storage options to be utilized.

With these additional assumptions, the marginal cost of storage in period can be expressed as where is a measure of aggregate stocks in the market. The marginal convenience yield can be expressed as . The negative slope of this function shows that marginal convenience yield is lower when stocks are larger. The net cost of storage is given by . We subtract because it is a benefit, which is equivalent to a negative cost. We call this net cost of storage the commodity’s *carrying cost*

Similar to the previous module, the intertemporal LOP tells us that when storage is positive then the price increase must equal the carrying cost. A merchant is indifferent between selling the commodity immediately and receiving price , or storing the commodity for one period and then selling it for a net price of . Indifference implies . We can rearrange this to obtain a revised LOP expression:

It is important to recognize that a negative value for , which emerges when convenience yield is larger than the cost of storage, results in price decreasing rather than increasing over time. This possibility is central to the modeling of commodity prices because it allows for positive stock carry over and a gradual downward price adjustment from one crop year to the next rather than the classic saw-tooth outcome.

Let and . If we substitute this pair of expressions together with and into the previous pricing equation we obtain the final LOP expression:

## Pricing Dynamics

The goal is to generate a pricing pattern which allows for stock carry over and a gradual price reduction when transitioning from one crop year to the next. The model that we will use consists of two crop years, each with four quarters. A crop year begins with harvest and so Q1 corresponds to the September to November quarter. As is typical for most U.S. and Canadian crops, price falls in both the summer quarter when pre-harvest stocks are at their lowest, and in the fall quarter as harvest gradually replenishes the stock pile.

If the price falls in Q4 then . We know that because storage costs are always higher and the convenience yield is always lower when stocks are higher. It must therefore be the case that . In Q1 when stocks are largest and convenience yield is smallest, we see that takes on its largest value, and thus the price increases are the largest. As stocks diminish as we progress from Q1 to Q2 to Q3 the value of remains positive but it is getting smaller and smaller. Thus, the price increases are weakening. By the time we hit Q4 the convenience yield is larger than the storage costs, which means that and the price decreases instead of increases.

The above LOP price equation is one of three key equations which governing how prices change over time. The general stock adjustment equation is where is the level of new production (i.e., harvest) in period and is the level of consumption in period . For the case of quarterly data, the full stock adjustment equation can be written as

In the above equation represents starting stocks, represents the level of stocks which are typically carried over from year to year (i.e., long-run carry over) and is a measure of the short run demand for year 2 stocks to be carried into year 3 net of . These three variables, , and , together with the two harvest variables, and , are exogenous in the model. An exogenous variable means that we must attach values to these variables from outside the model rather than calculating equilibrium values from within the model.

The final equation for simulating prices over the eight quarters is the inverse demand for the commodity by the processor. We will use the standard linear demand schedule, . The model has seven LOP equations, eight stock adjustment equations and eight demand equations for a total of 23 equations. There are also 23 variables to be solved for: eight quarterly prices (), eight quarterly consumption levels () and seven ending stocks (). There are only seven stock variables because as you can see FROM above the value of is determined by . After assigning values to the model parameters and the exogenous variables we can solve this system of equations and then recover the equilibrium prices, which we are interested in analyzing.

## Data

We will simulate eight quarterly corn prices, which means two full years beginning with Q1 in the fall of year 1 and ending with Q8 in the summer of year 2. Data from the USDA Feed Grains Database reveals that average corn harvested acres, yield per harvested acre and beginning stocks for the most recent five crop years (2015/16 - 2019/20) was 82.91 million acres, 173.4 bushels per acre and 2.015 billion bushels, respectively. Multiplying the five year average acreage and yield gives the five year average production of 14.38 billion bushels. If these estimates are used as the long term average it follows that for the base case. Similarly, if the five year average level of stocks is viewed as normal pipeline stocks it follows that for the base case.

The two parameters from the demand equation are set to and . These parameters ensure that the average price across all eight quarters is $3.628/bu, which is very close to the $3.648/bu average farm gate price for 2016 - 2020. Moreover, the demand elasticity, which is calculated with the simulated $3.628/bu average quarterly price and the simulated 3.595 billion bushel average quarterly consumption, is equal to -0.288. This simulated elasticity is reasonably close to the -0.2 corn demand elasticity estimate which was reported in the literature.

Data on storage costs and convenience yields are not available and so it is not possible to directly estimate values for and . The chosen values, and , are those which achieve a reasonably close match between the seasonal pattern of the simulated prices and real-world prices. More is said about this below.

Let’s formally assign these values to the model parameters and exogenous variables. The full set of values is placed in a vector titled *v*.

a <- 16.21  
b <- 3.50  
m0 <- -0.22  
m1 <- 0.03  
S0 <- 2.015  
H1 <- 14.38  
S\_bar <- 2.015  
v <- c(a, b, m0, m1, S0, H1, S\_bar)

## Simulation

We could program R to solve the systems of 23 equations and 23 variables and then recover the eight equilibrium quarterly prices. However, to tie in better with the next module it is useful to specify the equilibrium prices as linear functions of and . These two variables are chosen because it is natural to view (i.e., year 2 harvest) as a random variable in Q1 through Q4, and to view (i.e., long term net demand for stocks) as a random variable in all eight quarters. The USDA provides a forecast for both of these variables and the properties of these forecasts will be used when using the model to generate random prices.

The desired equation is

In this equation the (tilde) on the and variables indicate that they are random as described above.

There are eight values for each of , and , which means that we need 24 values to generate the set of eight quarterly prices. The complex set of equations which are required to obtain the 24 values are contained in a R function titled “get\_delta” (this function is stored in file titled “price\_function.R”). We can call this function into our program and feed into it the parameter vector *v*. The function will return the 24 delta values, which are stored in a matrix titled “del”.

source(here("Code", "price\_function.R"))  
del <- get\_delta(v)  
del

## del0 del1 del2  
## [1,] 9.545454 -0.4212615 0.4005162  
## [2,] 9.760180 -0.4248723 0.4039492  
## [3,] 9.919621 -0.4321249 0.4108446  
## [4,] 10.025144 -0.4430814 0.4212615  
## [5,] 10.077655 -0.4578358 0.4352893  
## [6,] 10.077603 -0.4465144 0.4530481  
## [7,] 10.024987 -0.4390203 0.4746902  
## [8,] 9.919357 -0.4352893 0.5004010

The next step is to use these 24 values together with to generate the eight quarterly prices. The following function does the necessary calculations:

get\_price <- function(H5,D) {  
 Price <- del[,1] + del[,2]\*H5 + del[,3]\*D   
}

If we assume and , which implies that year 2 harvest and long term net demand are both normal, then the prices which are generated will equal the base case prices. We first assign these values to and and then call the above pricing function:

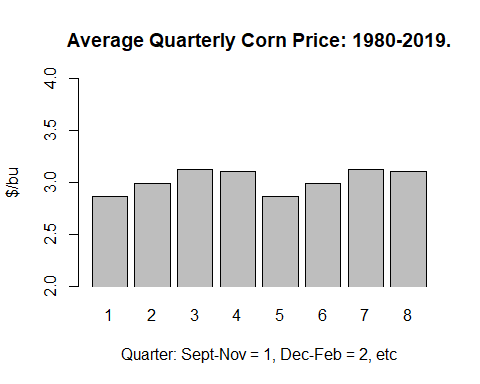
H5 <- 14.38  
D <- 0  
P <- get\_price(H5,D)  
P

## [1] 3.487714 3.650515 3.705664 3.653634 3.493977 3.656725 3.711874 3.659897

## Comparing Simulated and Real-World Corn Prices

It is of interest to compare the simulated quarterly prices of corn with the real world long term average (1980 - 2019) quarterly prices of corn (taken from Vercammen 2021). A plot of these long term quarterly average prices can be generated using the following code.

historic <- c(2.866, 2.993, 3.123, 3.105, 2.866, 2.993, 3.123, 3.105)  
barplot(historic,names.arg = c(1,2,3,4,5,6,7,8), main="Average Quarterly Corn Price: 1980-2019.",  
 xlab="Quarter: Sept-Nov = 1, Dec-Feb = 2, etc", ylab="$/bu",  
 beside=TRUE, ylim=c(2, 4), xpd = FALSE)



Notice that pricing pattern for the long term averages looks similar to what we observed for U.S. hay prices. Prices are lowest in the harvest quarters (Q1 and Q5) and rise during the winter and spring quarters (Q2, Q3, Q6, Q8). In the summer quarter (Q4 and Q8) the price begins to decline. Theory suggests that it is the particularly high convenience yield in Q4 which is responsible for the price drop before the arrival of harvest.

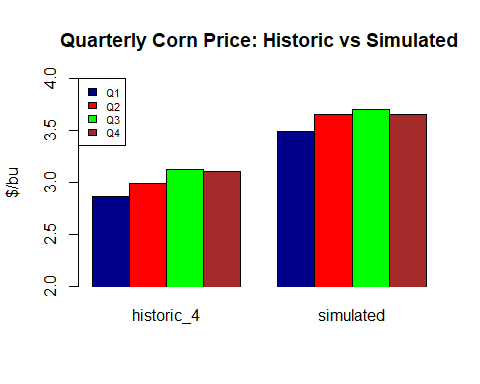
To compare the simulated and real-world prices it is useful to consider Q1 through Q4 only since the pattern in year 2 is quite similar to the pattern in year 1. The following code creates the *all\_price* matrix by binding together the two individual price series.

historic\_4 <- historic[1:4]  
simulated <- P[1:4]  
all\_price <- cbind(historic\_4,simulated)  
rownames(all\_price) <- c("Q1","Q2","Q3","Q4")  
all\_price

## historic\_4 simulated  
## Q1 2.866 3.487714  
## Q2 2.993 3.650515  
## Q3 3.123 3.705664  
## Q4 3.105 3.653634

We can now plot the pair of prices series using:

barplot(all\_price, main="Quarterly Corn Price: Historic vs Simulated",ylab="$/bu",  
 col=c("darkblue","red", "green", "brown"),  
 legend = rownames(all\_price), args.legend = list(x = "topleft", cex = .7), beside=TRUE, ylim=c(2, 4), xpd = FALSE)



Notice that the simulated prices are higher because the model has been calibrated to the average corn prices in the 2016 - 2020 period, whereas the long term average prices utilize the much longer 1980 - 2019 period. Despite the different levels of pricing the seasonal pattern is quite similar for the two pricing outcomes. This suggests that that despite its simplicity the calibrated pricing model is well suited to analyzing seasonal prices in the U.S. corn market.

## Verifying the LOP

It is a good idea to verify that the model is working as intended. First, we should check that the LOP equation, , holds for all eight quarters. Second, we should check that consumption of the inventory across all eight quarters is equal to . In other words, all stocks are fully accounted for. Both of these checks are conducted in the Appendix

## What If Analysis

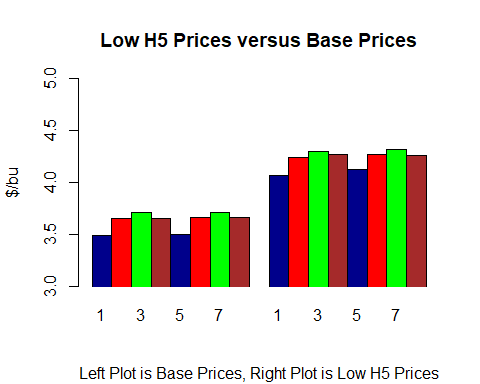
Now that we have the model built we can do “what if” analysis. An obvious “what if” is how does the set of 8 prices respond to a change in the size of the year 2 harvest (e.g., a smaller value for due to fewer planted acres)? For example, suppose , which is below the level of year 1 harvest, . Generating a new set of eight quarterly prices requires using the pricing function, “get\_price”, with the revised value for .

H5 <- 13  
D <- 0  
P\_rev <- get\_price(H5,D)  
P\_rev

## [1] 4.069054 4.236839 4.301997 4.265086 4.125790 4.272915 4.317722 4.260596

To graph this revised set of prices together with the base set of prices we first combine the two price series into a single matrix and then generate a side-by-side bar chart similar to what what shown in the previous section.

all\_price2 <- cbind(P,P\_rev)  
  
barplot(all\_price2, main="Low H5 Prices versus Base Prices",ylab="$/bu", sub="Left Plot is Base Prices, Right Plot is Low H5 Prices", names.arg=c(1,2,3,4,5,6,7,8,1,2,3,4,5,6,7,8),  
 col=c("darkblue","red", "green", "brown"),  
 beside=TRUE, ylim=c(3, 5), xpd = FALSE)



The chart shows that the lower year 2 harvest raises all eight prices by roughly the same amount. This is an important pricing property of storable commodities. A supply or demand shock in the current year will impact prices in future years by roughly the same amount. This is because merchants are continually shifting the level of stocks being carried through time in order to maximize the average selling price of the commodity.

When prices adjust in response to a supply or demand shock such as the one implied by the previous chart the impact on the eight prices is similar but not identical. This is because the shock will typically change the level of stocks, which in turn affects storage costs and convenient yield. The pricing impacts are calculated as follows:

impacts <- P\_rev - P  
impacts

## [1] 0.5813409 0.5863238 0.5963324 0.6114524 0.6318134 0.6161899 0.6058481  
## [8] 0.6006992

We see that the reduction in the year 2 harvest increased the Q1 price by $0.581/bu and raised the Q4 price by $0.611/bu. This outcome emerges because with the smaller year 2 harvest, more is stored in year 1 and this raises the marginal cost of storage. We know that price must rise at a rate equal to the marginal cost of storage. Thus the price increase in Q4 is larger than the price increase in Q1. The rate of pricing increase is very small but it is nevertheless important to consider because thinking through these what if scenarios helps us to improve our economic intuition regarding pricing dynamics.

There are other what if scenarios that can be examined, and most have predictable outcomes. For example, increasing the intercept value of the quarterly inverse demand schedule will raise the full set of prices. Similarly, raising the demand for stocks that leave year 2 or reducing the amount of stocks which enter year 1 will also raise the full set of prices because there is less corn available for consumption in years one and two.

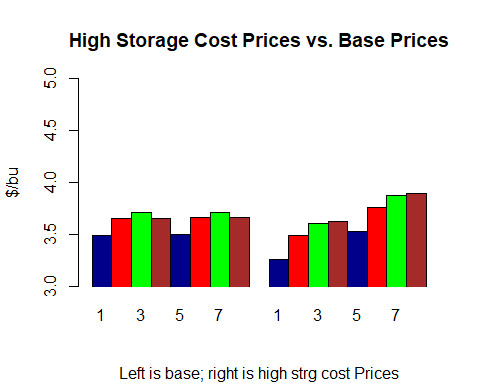
What happens to the set of prices if the cost of storage increases. In particular suppose that *m* increases from its base case value of -0.22 t0 -0.15. To make this change we need to update the various values which are used inside the pricing function. Specifically, after we change the value of , or any other parameter except of and , we need to call the *get\_delta(v)* function to generate the revised values for the various parameters. The following code will accomplish this task.

m0 <- -0.15 # base value = -0.22  
v <- c(a, b, m0, m1, S0, H1, S\_bar)  
H5 <- 14.38 # base value = 14.38  
D <- 0  
del <- get\_delta(v)  
P\_rev2 <- get\_price(H5,D)  
P\_rev2

## [1] 3.254649 3.485453 3.607190 3.620902 3.526708 3.755200 3.876937 3.892961

Using the same procedure as before, we can bind the new results and the base case results together, and then plot the pair of price series.

all\_price3 <- cbind(P,P\_rev2)  
  
barplot(all\_price3, main="High Storage Cost Prices vs. Base Prices",ylab="$/bu", sub="Left is base; right is high strg cost Prices", names.arg=c(1,2,3,4,5,6,7,8,1,2,3,4,5,6,7,8),  
 col=c("darkblue","red", "green", "brown"),  
 beside=TRUE, ylim=c(3, 5), xpd = FALSE)



A comparison of the two plots reveals how the higher cost of storage has impacted both the level of prices and seasonal nature of prices. It is clear that the year 1 prices are higher (especially in Q1 and Q2) and the year 2 prices are lower (especially in Q7 and q8). This outcome is expected because with the higher cost there will be lower storage both within each year and across the two years. These two impacts combined mean that the consumption of corn will be much higher in Q1 and Q2, and this forces the Q1 and Q2 prices down. Similarly, due to the lower volume of storage, less will be available for consumption in Q7 and Q8. This reduced supply will result in relatively high prices in Q7 and Q8.

## Conclusion

This module should be viewed as a bridge between our study of pricing fundamentals in Module 1 and our study of commodity futures in Module 2. We now have a model that generates prices with fairly realistic seasonal patterns. In the next module (2C) we assume that forward looking traders understand the seasonality in spot prices and as a result the set of futures prices at a particular point in time (i.e., the forward curve) will reflect this seasonality. We will define basis as the spot price minus the futures price. If the spot market is located in Chicago where futures contracts are settled, then the basis will also reflect the seasonality in spot prices. If the spot market is located outside of Chicago then me must also account for transportation costs. The properties of the basis is important when we turn our attention to hedging because the profitability and risk sharing properties of the hedge are strongly connected to the seasonality in the basis.

Another module will focus on the connection between convenience yield and slope of the forward curve: contango (forward slope) and backwardation (negative slop). The slope of the forward curve is important for institutional traders who rely on “roll yield” in their long-only positions to generate profits. As you can see, there is no shortage of interesting and important topics which we will be examining in the next few modules.

# Appendix I

The purpose of this Appendix is to provide a detailed discussion about estimating dummy variables and first difference models.

When a dummy variable is included in an econometric model, the estimated coefficient on the dummy is interpreted as the difference between the conditional mean of the dependent variable for those observations when the dummy takes on a value of 1 versus 0. When data is non-stationary, we take the first difference to make the data stationary before proceeding with econometric estimation. In the October 14th class exercise, we had both types of differences. There was some confusion about the relationship between the two types of differences. The purpose this note is clear up the confusion regarding the two types of differences.

I will refer to the first type as “dummy difference” and the second type as “stationarity difference”.

The bottom line is as follows:

* The “dummy difference” method must be used in all models that use either an indicator variable such as the one we used for high stocks (equivalent to a single dummy variable) or a larger set of dummies (e.g., quarterly or monthly dummies).
* The “stationarity difference” method is optional, and is only used if the underlying data is non stationary (i.e., the mean value of the dependent variable is not constant over time).
* Time series models of commodity prices often do not use dummy variables and so the dummy difference method is not required. Time series models are often non-stationary and so the stationary difference method is commonly used.
* The one case where time series models of commodity prices do use dummy variables is when the goal is to estimate seasonality. In most cases the data which has seasonality is also non stationary, and so we must use both the dummy difference and the stationary difference method.
* Regardless of whether you estimate the model in levels (i.e., without taking the first difference to ensure the data is stationary) or in first difference format, the estimated coefficients have the same interpretation.
* Even though the interpretation of the estimated coefficients are the same, the value of the estimated coefficients are typically somewhat different when you estimate the model in levels versus first differences. The less stationary the data, the greater the difference between the two sets of estimates.

## Dummy Difference Only

Suppose the price series is stationary and so there is no need to difference the data to make the series stationary. Also suppose we have monthly data over several years. We are interested in estimating the long-term average monthly price (i.e., the seasonality).

Let denote the price in month of year . Suppose we estimate the following model:

In this case our estimate of is the long-term average price across all months and years. The data is stationary by assumption, and so our model is well specified (i.e., is indeed constant).

Suppose we believed that prices were on average higher in the first six months than the last six months (i.e., there exists bi-annual seasonality). To estimate this model we might consider including two dummy variables: and . We assume that takes on a value of 1 if the current month as indicated on the subscript is one of the first six months of the year, and zero otherwise. Similarly, We assume that takes on a value of 1 if the current month as indicated by the subscript is one of the last six months of the year, and zero otherwise.

We can write the above equation with the two dummies included as follows:

If you wrote out the right hand side variables starting in a January (remembering that is multiplied by a “1”), the data would look like

1 1 0  
1 1 0  
1 1 0  
1 1 0  
1 1 0  
1 1 0  
1 0 1  
1 0 1  
1 0 1  
1 0 1  
1 0 1  
1 0 1  
1 1 0  
etc

If you were able to estimate this model then your estimate of would continue to be your long term average price across all months and years. Now is the difference between the long term average price for just the first six months versus all months. Similarly, is the difference between the long term average price for just the last six months versus all months.

If you put the above data into the X matrix, and the prices into a Y matrix you could attempt to estimate the three s using the familiar formula: . However, regardless of whether you are using Excel, Stata or R, you will receive an error message telling you that there is a singularity in your data. This occurs because if you sum the values in the last two columns of the above data set, the resulting column is a vector of 1s. The first column is also a vector of 1s. This is where the singularity problem arises. The X matrix cannot be inverted when one column can be expressed as a linear combination of two or more other columns.

One way to proceed is keep both dummies and estimate the model without a constant. In this case the estimated is the long term average price for the first six months, and the estimated is the long term average price for the last six months. This approach is seldom used in applied work. Rather, it is conventional to keep the intercept and to drop one of the dummies.

Let’s drop the second dummy, and so now our model simplifies to

This model can be estimated but how should we interpret our estimates of and ? In this case is the long run average price when (i.e., the last six months) and is the long run average price when (i.e., the first six months). If we subtract the first from the latter we are left with being the long run average price when (i.e., the first six months) minus the long run average price when (i.e., the last six months). More simply stated, is a measure of how much higher the long run average price is during the first six months versus the last six months.

Consider now the model that you estimated during the October 14th lecture. Let denote the year and denote the quarter. If you want to estimate the model with quarterly dummies, then for the reasons discussed above, one dummy must be excluded from the model. If we exclude the quarter 4 dummy then the model becomes

In this case we interpret as the long term average price for Q4 since this was the quarter which was excluded from the analysis. Similarly, we interpret , and as the long term average prices in quarters 1, 2 and 3, respectively, minus the long term average price in Q4. Thus, is the seasonal price difference relative to the omitted quarter. If you choose to exclude a different quarter then your estimates of the , and coefficients will change.

In the analysis of commodity prices, the time period with the lowest price (e.g, harvest) is commonly excluded. This makes it possible to interpret a particular values as the cost of storing the commodity from harvest to the period . Why? Because according to the LOP, the average long term monthly price increase will equal the monthly cost of storage. Thus the cumulative price increase must equal the cumulative storage cost.

## Stationarity Difference Only

Suppose there was no seasonality in the data and so there is no need to include monthly or quarterly dummies to control for the seasonality. However, suppose price is non stationary, which means that the mean of the price series is not constant over time. In particular, assume where

In other words, the error term of the pricing function follows a random walk with drift. The price series, , must be made stationary before we can use it in most econometric analysis.

The standard way to make a price series stationary is to take the first difference. If you are working with quarterly data, this means subtracting the price in the previous quarter from the price in the current quarter. That is, . If we substitute in the expression for we get

However, an earlier equation shows that . This means that

It can now be seen that the series of price differences is stationary because there are no terms on the right hand side which causes the mean value to drift over time.

## Combine Dummy Difference and Stationary Difference

Suppose our price series is non stationary as described above, and we want to use dummy variables to estimate seasonality. This means that we must take the first difference as in the previous section to make the price series stationary. As well, we must exclude the Q4 dummy and then interpret the estimated coefficient on the dummy as the difference in the long term average price for quarter and quarter 4.

We begin with

After substituting in

The first difference of this equation can be expressed as

How should the estimated coefficients of this model be interpreted? In the previous model was the long term average price in Q4. However, has disappeared from the equation. As well, the dummies in brackets now take on three possible values: -1, 0 and 1. Does this mean that we can no longer interpret as the difference between the long run average price in quarter and quarter 4? It is shown in the Github reading that the answer is no; continues to have the same interpretation as before.

The one remaining issue to discuss is the effects of long term price inflation on the estimates. Suppose price increases by a constant $2 per tonne per quarter over a large number of quarters due to economy wide inflation. If we estimate the above model, our estimate of will be 2 because the coefficient is an estimate of the long term average price drift over time. The problem is that the seasonal price differences will become amplified over time due to the inflation, and the dummy variables do not reflect this growth. This means that the estimates of will pick up the average level of seasonality over the long time period. This average level is likely to be much smaller than the current level of seasonality due to on going price inflation. This was the situation encountered when estimating seasonality in U.S. hay prices in the first half of Module 2C.

There are different ways to model growth in seasonality over time. However, for our purposes it is best to eliminate the inflation before estimating the dummy variable model. The most common way to do this is to divide each price by the consumer price index (CPI). If the commodity’s price is inflating at the same rate as the CPI then and the above model becomes

## Interaction Variables

The inclusion of a high-stocks indicator variable means that the dummy variable equation expressed in levels can be written as

The quarterly indicator variable, takes on a value of 0 if the year in question has normal stocks, and a value of 1 if it has high stocks.

How can we interpret the coefficients. If the three dummies and the indicator all take on a value of zero, we are left with . This must mean that is the long term average price in Q4 of a normal stock year. If the three dummies take on a value of zero and the indicator takes on a value of 1, we are left with . This must meant that is the long term average price in Q4 of a high stock year. It now follows that is the long term average price in quarter of a normal year, and is the long term average price in quarter of a high stock year.

We can use the previous expressions to write

Thus, is the quarter versus quarter 4 difference in the price gap, where the gap is the long term average price difference between a high stock year and a normal year. The previous expression can be rearranged and rewritten as

This means that can also be interpreted as the high stock versus normal stock difference in the price gap, where the gap is the long term average price difference between quarter and quarter 4.

Similar to the model without interaction terms, this new model can be written and estimated in first differences to ensure that the price data is stationary:

In the student exercise and Module 2C the drift parameter, , is assumed to equal zero.

Despite this rather complicated equation, the estimates of the and coefficients remain the same as in the previous case where the model was specified in levels.

# Appendix II

The purpose of this Appendix is to demonstrate the LOP and the stock adjustment equation hold for the base case set of eight simulated prices. The verification begins by substituting the equilibrium price into the inverse demand schedule to obtain the quarterly consumption, , Then substitute quarterly consumption into the stock adjustment equation to derive the quarterly stock levels, . Finally, substitute the derived values of and the equilibrium prices into the LOP equation, , to verify that the equation holds.

m0 <- -0.22 # base value = -0.22  
v <- c(a, b, m0, m1, S0, H1, S\_bar)  
H5 <- 14.38 # base value = 14.38  
D <- 0  
del <- get\_delta(v)  
P\_chck <- get\_price(H5,D)  
  
  
X <- a/b - 1/b\*P\_chck  
S <- rep(0, times = 8)  
S[1] <- S0 + H1 - X[1]  
S[2] <- S[1] - X[2]  
S[3] <- S[2] - X[3]  
S[4] <- S[3] - X[4]  
S[5] <- S[4] + H5 - X[5]  
S[6] <- S[5] - X[6]  
S[7] <- S[6] - X[7]  
S[8] <- S[7] - X[8]  
S

## [1] 12.760061 9.171637 5.598970 2.011436 12.758287 9.171637 5.600744  
## [8] 2.015000

lop <- rep(0, times = 8)  
for(i in 1:7)  
{  
 lop[i+1] <- P[i+1]-P[i]-(m0+m1\*S[i])  
}  
lop

## [1] 0.000000e+00 -3.247402e-15 -1.193490e-15 3.635980e-15 -4.996004e-15  
## [6] 8.049117e-16 5.828671e-16 2.137179e-15

The outcome that all LOP check values are zero confirms that all eight quarterly prices are consistent with the intertemporal LOP.