Regression Class 4 - Covid Update

Closing Topics

We're going to walk through a few closing topics that should help you in your analyses. First, load the following libraries:

```
library(tidyverse)
library(lubridate)
library(stringr)

# new libraries
library(sweep)

rmse <- function(error)
{
  sqrt(mean(error^2))
}

exp_smooth = function(x, alpha) {
  # Performs exponential smoothing by taking a weighted average of the
  # current and previous data points
  # x     : numeric vector
  # alpha : number between 0 and 1, weight assigned to current data value
  # Returns a numeric vector of the same length as x and values as the
  #       weighted averages of the (current, previous) consecutive pairs
  s = numeric(length(x) + 1) # make s 1 cell longer than x
  for (i in seq_along(s)) {
    if (i == 1) { # set the initial value of s the same as that of x
      s[i] = x[i]
    } else {
      # weight current value with alpha and previous value
      # with 1-alpha, and sum
      s[i] = alpha * x[i - 1] + (1 - alpha) * s[i - 1]
    }
  }
  s[-1] # drop the 1st element in s because it's extra
}
```

Interaction Terms

We've discussed different kinds of relationships between independent variables *(e.g., surrogate variables)*. In chapter 3 of ISL, the authors discuss how spending money on radio advertising actually increases the effectiveness of TV advertising, so that the slope term for TV should increase as radio increases. In marketing, this is known as a synergy. We model this effect simply by multiplying the terms together, as we'll see below.

The following data is from hospital stays, with age, sex, bmi, children, smoker, region and expenses for the stay. We want to build a model that can predict expenses based on the following independent variables. Let's take a look at the correlations:

```r
insurance <- read.csv("insurance.csv", stringsAsFactors = TRUE)
knitr::kable(cor(data.matrix(insurance))) %>%
  kable_styling(full_width = F, bootstrap_options = "striped", font_size = 9)
```

	age	sex	bmi	children	smoker	region	expenses
age	1.0000000	-0.0208559	0.1093410	0.0424690	-0.0250188	0.0021273	0.2990082
sex	-0.0208559	1.0000000	0.0463802	0.0171630	0.0761848	0.0045884	0.0572921
bmi	0.1093410	0.0463802	1.0000000	0.0126447	0.0039681	0.1574391	0.1985763
children	0.0424690	0.0171630	0.0126447	1.0000000	0.0076731	0.0165694	0.0679982
smoker	-0.0250188	0.0761848	0.0039681	0.0076731	1.0000000	-0.0021807	0.7872514
region	0.0021273	0.0045884	0.1574391	0.0165694	-0.0021807	1.0000000	-0.0062082
expenses	0.2990082	0.0572921	0.1985763	0.0679982	0.7872514	-0.0062082	1.0000000

So, if we cut it off at .2, that leaves age, smoker, and bmi *(close enough)* - the rest with minor effects. Let's create a training set with 60% of the data, and build a linear model using those variables:

```r
set.seed(13)
insurance = rowid_to_column(insurance, var="SampleID") # this creates a primary key for sampling
xTrain = sample_frac(insurance, .6)
xTest = anti_join(insurance, xTrain, by = "SampleID")
mod1 = lm(expenses ~ age +  bmi + sex + smoker + region,
           data = xTrain)
xTest$Pred = predict(mod1, xTest)
```

Check the rmse on the testset *(as a % of total expenses)*:

```r
rmse(xTest$expenses - predict(mod1, xTest))
```

```
## [1] 5764.583
```

About 9% error - not a bad start with this data. BTW, Another way to compare models is to measure the correlation between predicted and actual *(this comes in handy with irregular dependent variables)*.88% correlation here:

```r
cor(xTest$expenses, xTest$Pred)
```

```
## [1] 0.8801343
```

So, now let's consider the interaction term. Can you guess what it is? How about people who are overweight and smoke - do you think that's a combination that might increase hospital expenses? Let's test a series of models: 1. the simple linear model we just created, 2. a model that considers the interaction effect of bmi and smoking, and lastly, we add a polynomial term to emphasize age:

```r
mod2 <- lm(expenses ~ age + bmi + sex + smoker, data = xTrain)
rmse(xTest$expenses - predict(mod2, xTest))
```

```
## [1] 5754.592
```

```r
mod3 <- lm(expenses ~ age + bmi + sex + smoker + bmi*smoker, data = xTrain)
rmse(xTest$expenses - predict(mod3, xTest))
```

```
## [1] 4565.688
```

```r
mod4 <- lm(expenses ~ age + bmi + sex + smoker + bmi*smoker + I(age^2), data = xTrain)
rmse(xTest$expenses - predict(mod4, xTest))
```

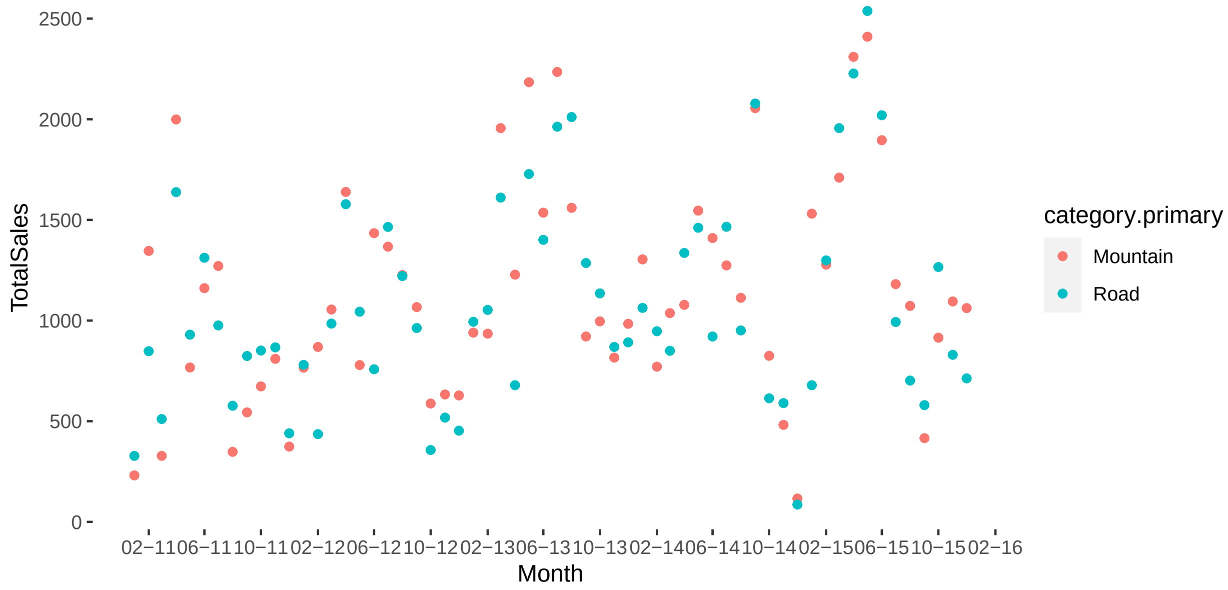
```
## [1] 4541.453
```

So, the interaction term significantly improved the model *(the polynomial improved a little, but not really worth it)*. Just some advice: Before using interaction, be sure you can define the relationship in logical terms.

Traditional Time Series with Exponential Smoothing

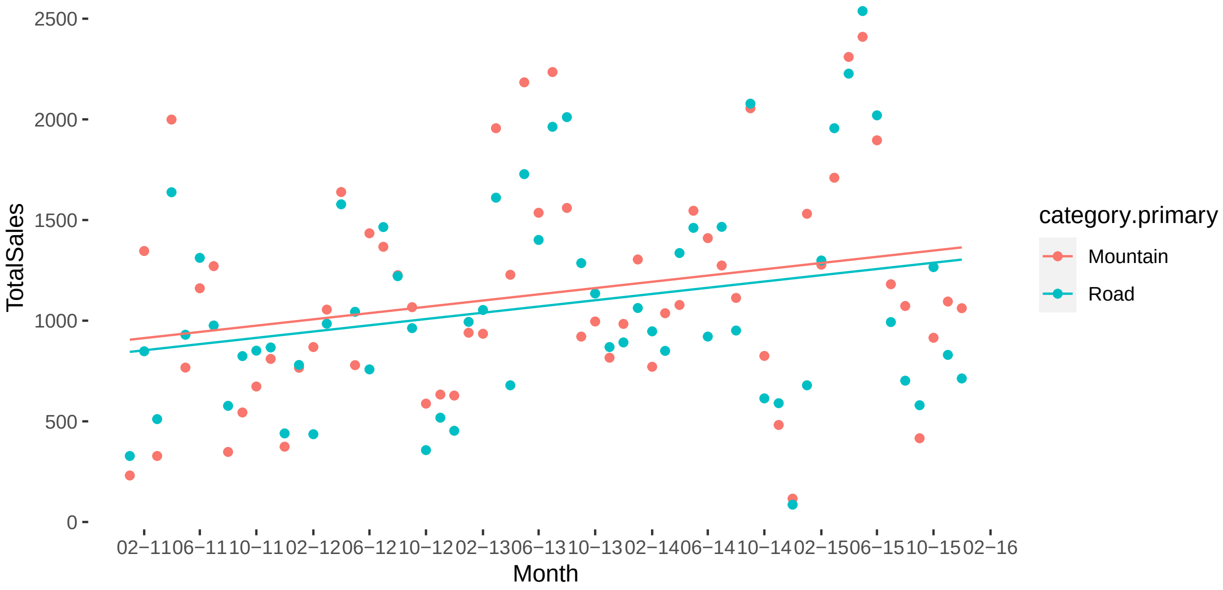
I have to preface this section with an opinion: if all you know about a process is that time has passed, then you need to gather more data. Time series are usually of limited value, but just so you know, here's the basic idea. Load the data form bike_sales *(sweep package)*, and group by Month:

```
dfBikeSales = bike_sales

dfBikeSalesMo = dfBikeSales %>% mutate(Month = floor_date(order.date, "month")) %>%
  group_by(category.primary, Month) %>% summarise(TotalSales = sum(order.line))

p1 = ggplot(dfBikeSalesMo, aes(Month, TotalSales, color = category.primary)) + geom_point() +
  scale_x_date(breaks = "4 month", date_labels = "%m-%y" ) +
  theme(
    panel.background = element_rect(fill = "white")
  )
p1
```



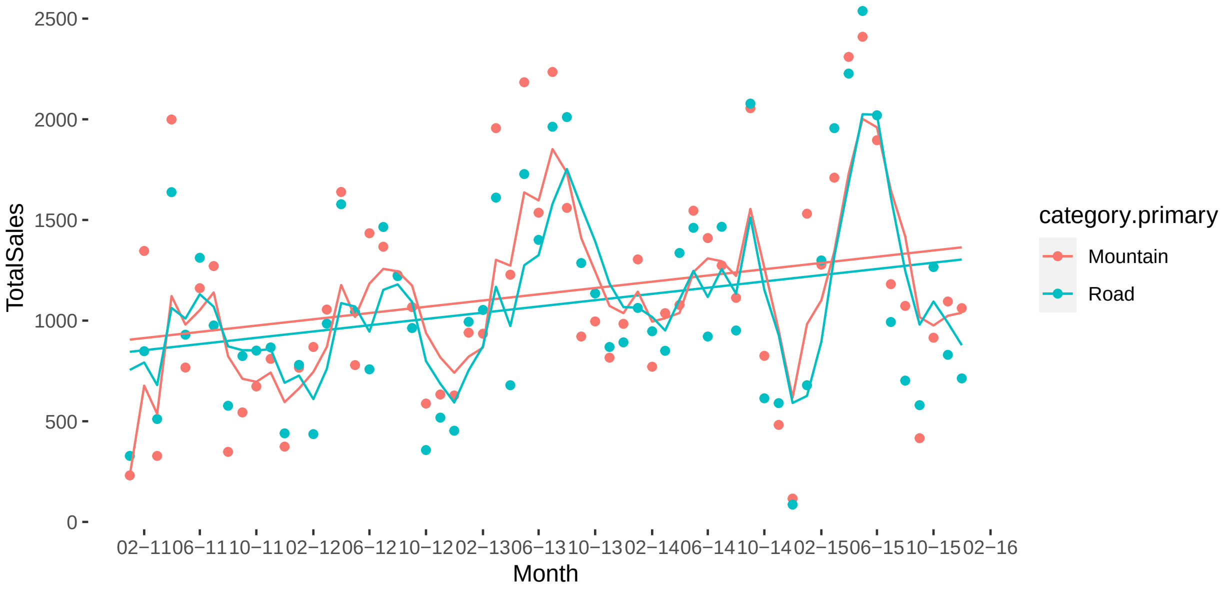
Let's look at a linear model. Just from a visual, I think we can say it's not a great fit:

```
mod1 = lm(TotalSales ~ category.primary + Month, dfBikeSalesMo)
dfBikeSalesMo$lmPred = predict(mod1, dfBikeSalesMo)

p1 = p1 + geom_line(data = dfBikeSalesMo, aes(Month, lmPred, color = category.primary))
p1
```



We're going to take another approach. The function we loaded was an exponential smoothing function *(which exponentially decreases weight as we move backwards in a moving average)*. It's similar to the local regression in 2 dimensions.

```
dfBikeSalesMo$ESPred = round(exp_smooth(dfBikeSalesMo$TotalSales, 0.4), 0)
# .4 = weighting of most recent period

p1 = p1 + geom_line(data = dfBikeSalesMo, aes(Month, ESPred, color = category.primary))
p1
```

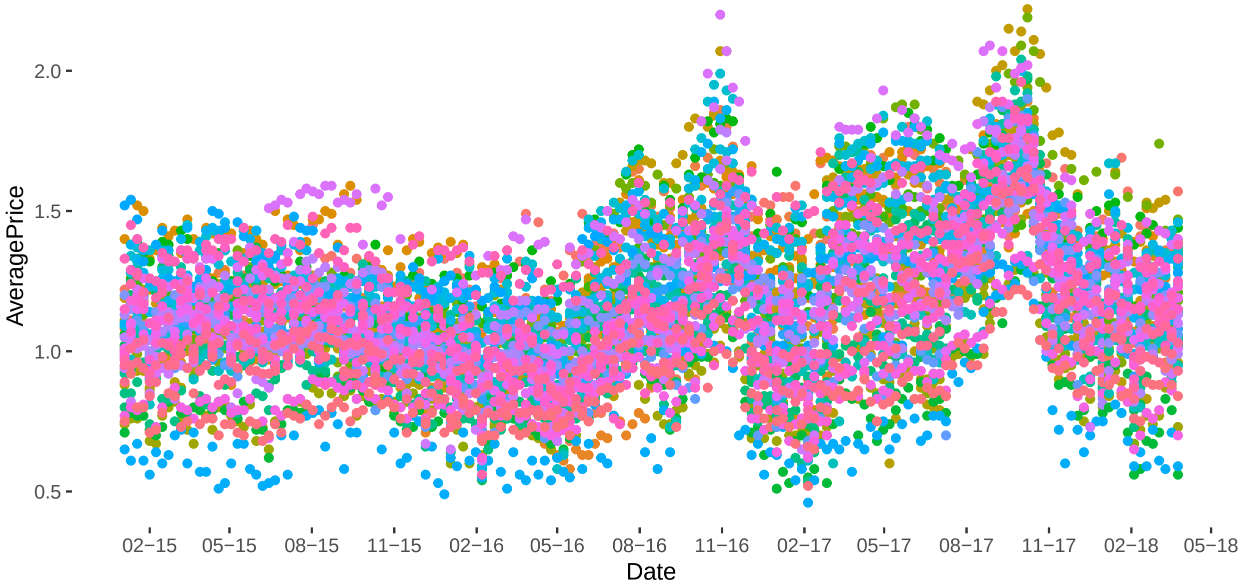


Again, this *looks* like a good fit, and this method does pick up on seasonal trends, but time-based trends are rarely enough for forecasting. Let's look at another dataset.

Last Exercise

If we want a model that **explains** responses for reasons other than the passage of time, then we need to use a multidimensional model. Let's take a look at avocado prices based on region, type *(organic or conventional)* and volume. Load the data and let's clean it up a bit *(filter out organic and focus on conventional)*, and then take a look by month *(regions in color)*:

```r
avocado <- read_csv("avocado.csv")
avocado$Date <- mdy(avocado$Date)
colnames(avocado)[4] <- "Volume"
avocado <- filter(avocado, type == "conventional")

p <- ggplot(avocado, aes(Date, AveragePrice, color = region)) +
  geom_point() +
  scale_x_date(date_breaks= "3 months", date_labels = "%m-%y") +
  theme(
    panel.background = element_rect(fill = "white"),
    legend.position="none"
  )

p
```

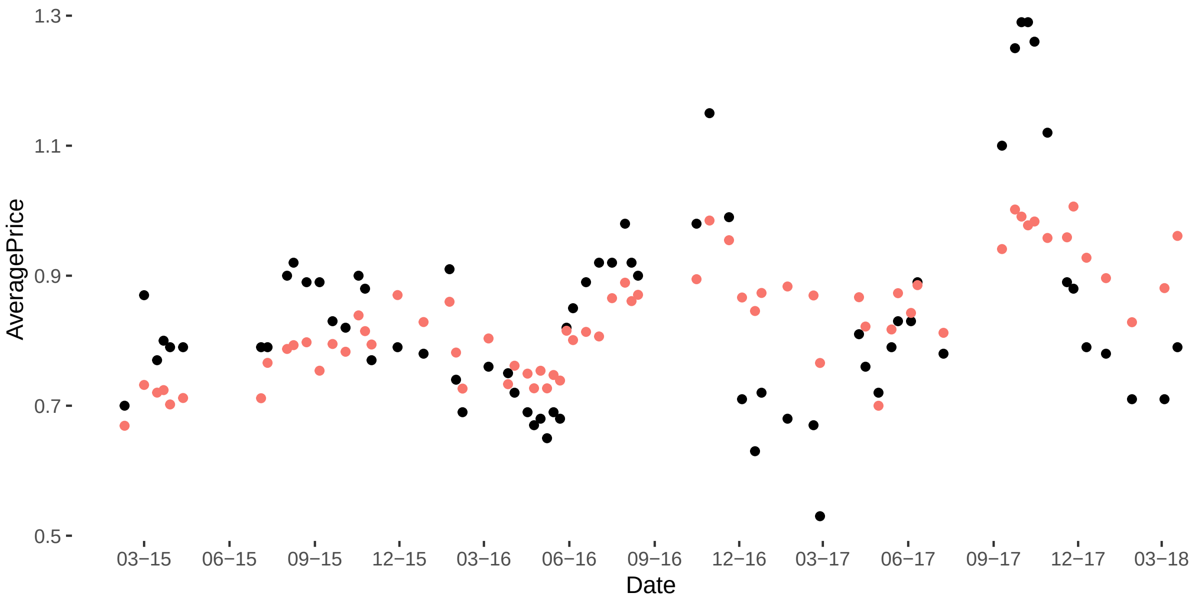


```r
avocado <- rowid_to_column(avocado, var="SampleID")
TrainSet = sample_frac(avocado, .6)
TestSet = anti_join(avocado, TrainSet, by = "SampleID")
```

Now, let's break this into training and test, and from the TestSet. we'll pull just the Houston region for testing, just for fun. I've created 2 polynomial variables here: $Volume^2$ and $Volume^3$. If you look at the shape of the data, its not really a quadratic shape - more of a cubic function. Then, we predict based on our model for the Houston region.

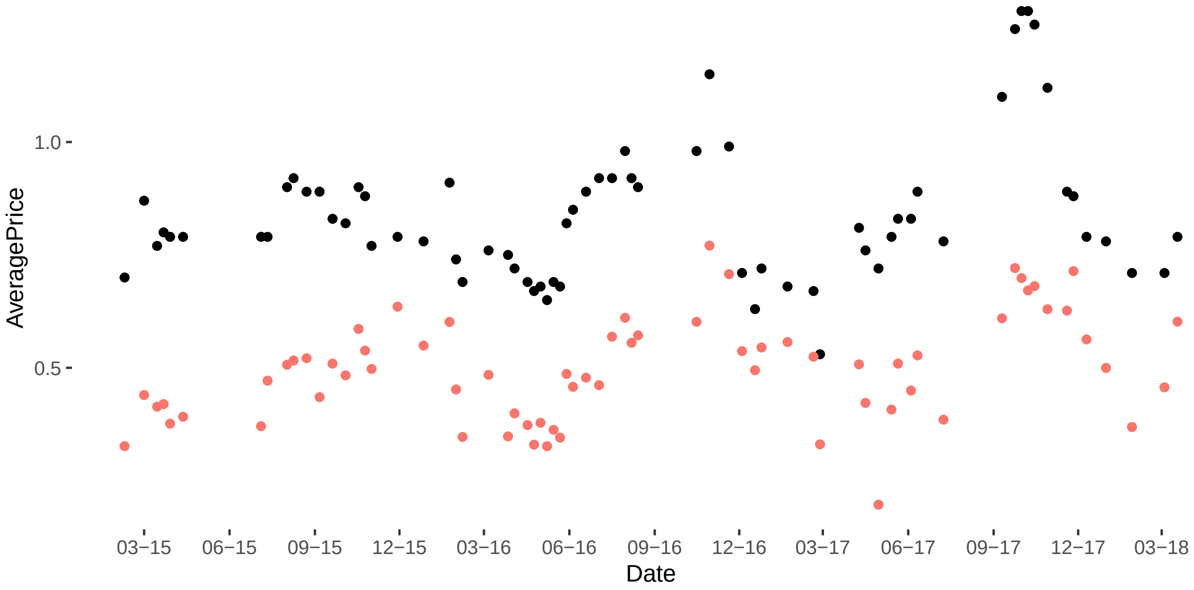
```r
avocadoHou = select(TestSet, AveragePrice, Date, region, Volume) %>% filter(region == "Houston")

HouMod <- lm(AveragePrice ~ Date + region + Volume +I(Volume^2) + I(Volume^3), data = TrainSet)

avocadoHou$yHat <- predict(HouMod, avocadoHou)
```

```
p <- ggplot(avocadoHou, aes(Date, AveragePrice)) +
  geom_point() +
  geom_point(aes(Date, yHat, color = 'red')) +
  scale_x_date(date_breaks= "3 months", date_labels = "%m-%y") +
  theme(
    panel.background = element_rect(fill = "white"),
    legend.position="none")
p
```



The difference here is: there are economic drivers in our model. Let's say that Volume doubles:

```
newTest =  avocadoHou %>% mutate(Volume = Volume * 2)

newTest$yHat <- predict(HouMod, newTest)

p <- ggplot(newTest, aes(Date, AveragePrice)) +
  geom_point() +
  geom_point(aes(Date, yHat, color = 'red')) +
  scale_x_date(date_breaks= "3 months", date_labels = "%m-%y") +
  theme(
    panel.background = element_rect(fill = "white"),
    legend.position="none"
  )
p
```



Notice the drop in prices… So volume is a dimension with value for forecasting *(Econ 101)*

Thanks

I've really enjoyed getting to know everyone in the class, and I wish you all the best in your career. Stay in touch!