Capstone: Analysis on 10-year BMW Sales trends in the U.S.

Stephanie Vaul

## Introduction

I'm going to analyze the BMW automotive sales trends in the USA over the last 10 years in order to determine which areas of the country and which model types display the highest sales volume. I will also look at how these trends vary between New and Used vehicles. In addition, I will try to identify a predictor for New sales.

My client is BMW North America, a company that my employer has partnered with over the last 10 years to provide customer retention marketing. Determining which model series do well in each region, will help BMW better focus their marketing efforts for various ad campaigns by leveraging existing trends, as well as being able to extrapolate those trends in other areas of similar makeup.

I plan to use dealership sales data that my company has been collecting for the past 10+ years, which we use to provide sales-to-service retention marketing, e.g. purchase thank you's, initial service reminders, etc. This data includes customer and vehicle information, along with the purchase date and type (new vs used). Some of the data includes demographic information, like birth date and gender. I will review the data to see if there is sufficient data in order to provide trend reports based on age &/or gender.

The proprietary sales data from most BMW centers in the U.S. going back 10 years includes the following:

* Region (BMW Region assigned to the selling dealership by BMW, of which there are four)
* Market (BMW Market assigned to the selling dealership by BMW)
* PurchDate
* VehYear
* VehModel
* VehECode (engine code assigned by BMW to categorize years & models)
* TradeVIN
* TradeVehYear
* TradeVehModel
* TradeVehECode
* NewUsed (New, Used, Other sale)
* PurchType (Lease, Cash, Finance)
* Gender (limited data provided here)

I will first review and clean the data so that we can properly work with the values. Then I'll segment the purchases by model series and type (new or used) in order to see both total and regional sales volumes. We will also take a look at purchase price information by model type and compare that with the sales volumes for each over the years. From there, I will include economic data gathered from Quandl that I can use to try to build a predictive sales model. This data will include information such as national unemployment, disposable income, loan rates, and overall automotive sales.

## The Data Set

To get started, let's set our working directory, clear our Environment, & load our libraries.

setwd("~/\_MyFiles/Data Science Workshop/Springboard\_DataScientist/Capstone Project")  
getwd()

## [1] "C:/Users/Stephaniev/Documents/\_MyFiles/Data Science Workshop/Springboard\_DataScientist/Capstone Project"

rm(list=ls())

library(devtools)  
library(stringr)  
library(tidyr)  
library(dplyr)  
library(ggplot2)  
library(scales)  
#install.packages("Hmisc")  
library(Hmisc)

Now let's read in our data set, which contains dealership, customer, and vehicle sales data for the "BMW" make. Once read in, we should take a look at the summary of this table. Since we don't want to reveal any sensitive data, we won't look at the head of the table until we've been able to reduce the data set to just the fields we need.

Sales <- read.table ("BMWSales.txt", header = TRUE, sep = "|", quote = "\"", stringsAsFactors = FALSE)

summary(Sales)

## RecNum BMWDlr BMWRegion BMWMarket   
## Min. : 1 Length:3926804 Length:3926804 Min. :10.00   
## 1st Qu.: 981702 Class :character Class :character 1st Qu.:16.00   
## Median :1963402 Mode :character Mode :character Median :26.00   
## Mean :1963402 Mean :26.47   
## 3rd Qu.:2945103 3rd Qu.:35.00   
## Max. :3926804 Max. :48.00   
## NA's :58157   
## DealerID DealerName DealerCity DealerState   
## Min. : 12 Length:3926804 Length:3926804 Length:3926804   
## 1st Qu.: 898 Class :character Class :character Class :character   
## Median :1013 Mode :character Mode :character Mode :character   
## Mean :1024   
## 3rd Qu.:1114   
## Max. :2616   
##   
## DealerZip VINID VIN Year   
## Length:3926804 Min. : 453 Length:3926804 Min. : 0   
## Class :character 1st Qu.:10044814 Class :character 1st Qu.:2007   
## Mode :character Median :14700210 Mode :character Median :2011   
## Mean :15005124 Mean :2010   
## 3rd Qu.:19612441 3rd Qu.:2013   
## Max. :26057070 Max. :2017   
## NA's :268   
## Make Model eCode   
## Length:3926804 Length:3926804 Length:3926804   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## NewUsed PurchDate PayType   
## Length:3926804 Length:3926804 Length:3926804   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## FinanceTerm PurchasePrice TradeVINID   
## Length:3926804 Min. : -4420 Min. : 26   
## Class :character 1st Qu.: 31500 1st Qu.: 8165448   
## Mode :character Median : 41200 Median :11098504   
## Mean : 45854 Mean :11967612   
## 3rd Qu.: 52365 3rd Qu.:15790270   
## Max. :10000000 Max. :26058534   
## NA's :509702 NA's :3217422   
## TradeVIN TradeYear TradeMake   
## Length:3926804 Min. : 0 Length:3926804   
## Class :character 1st Qu.:2004 Class :character   
## Mode :character Median :2007 Mode :character   
## Mean :2007   
## 3rd Qu.:2010   
## Max. :2029   
## NA's :3217610   
## TradeModel TradeECode CustID   
## Length:3926804 Length:3926804 Min. : 170999   
## Class :character Class :character 1st Qu.:12430611   
## Mode :character Mode :character Median :20091806   
## Mean :20123287   
## 3rd Qu.:27435784   
## Max. :35628131   
##   
## DMSNum CustFirstName CustLastName   
## Length:3926804 Length:3926804 Length:3926804   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## CustAddr1 CustAddr2 CustCity   
## Length:3926804 Length:3926804 Length:3926804   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## CustState CustZip DistFromDealer   
## Length:3926804 Length:3926804 Min. : 0.00   
## Class :character Class :character 1st Qu.: 6.00   
## Mode :character Mode :character Median : 12.00   
## Mean : 74.22   
## 3rd Qu.: 28.00   
## Max. :10339.00   
## NA's :40333   
## Gender BirthDate FixMe   
## Length:3926804 Length:3926804 Length:3926804   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
##

Now that we can see some information on all the fields, let's subset the data to eliminate any sales at non-BMW dealerships and remove the unnecessary fields. For example, we don't need to have any of the customer detail information, such as their name or street address. Since there would be no "New"" BMW sales records at non-BMW dealerships, we can eliminate those records. BMW is our client, so we want to focus on the sales - New or Used - at their dealerships. Once we've subset our data, we can take a look at the first few rows.

SalesSub <- subset(Sales, BMWDlr=="True", select = c(BMWRegion, BMWMarket, DealerID, DealerState, VINID, VIN, Year, Make, Model, eCode, PurchDate, NewUsed, PayType, FinanceTerm, PurchasePrice, TradeVINID, TradeVIN, TradeYear, TradeMake, TradeModel, TradeECode, CustID, CustState, Gender, BirthDate) )

head(SalesSub)

## BMWRegion BMWMarket DealerID DealerState VINID VIN Year  
## 1 EASTERN 15 40 PA 15698790 5UXZV4C54CL754821 2012  
## 2 EASTERN 15 40 PA 15985177 5UXFG2C59CL780025 2012  
## 3 EASTERN 15 40 PA 8574341 WBAHL83506DT06209 2006  
## 4 EASTERN 15 40 PA 18945332 WBAFU7C52DDU77149 2013  
## 5 EASTERN 15 40 PA 8137869 WBAVB735X7P101187 2007  
## 6 EASTERN 15 40 PA 18566362 WBA3B9G59DNR79004 2013  
## Make Model eCode PurchDate NewUsed PayType  
## 1 BMW X5 E70 2011-08-22 00:00:00 N C  
## 2 BMW X6 E71 2011-10-07 00:00:00 N C  
## 3 BMW 750i E65 2012-01-07 00:00:00 U F  
## 4 BMW 535 F07/F10/F11 2013-11-27 00:00:00 N L  
## 5 BMW 335i E90/91/92/93 2007-05-09 00:00:00 N C  
## 6 BMW 335 F30 2013-03-15 00:00:00 N F  
## FinanceTerm PurchasePrice TradeVINID TradeVIN TradeYear  
## 1 2011-09-22 00:00:00 46875.00 NA NA  
## 2 2011-11-07 00:00:00 62500.00 NA NA  
## 3 2017-01-07 00:00:00 27900.00 NA NA  
## 4 2016-11-27 00:00:00 60745.36 1710225 WBADD6327WGT94406 1998  
## 5 2007-06-09 00:00:00 43845.00 NA NA  
## 6 2016-03-15 00:00:00 51290.00 5838041 WBAVD33546KL52088 2006  
## TradeMake TradeModel TradeECode CustID CustState Gender BirthDate  
## 1 378290 PA M   
## 2 378290 PA M   
## 3 378378 PA M   
## 4 BMW 528i E39 378408 PA   
## 5 378425 FL M   
## 6 BMW 330xi E90/91/92/93 378540 PA

At this point, it looks like we'll have an issue with our date fields since they're currently formatted as characters. We need to set PurchDate, along FinanceTerm and BirthDate, to a date format before we can reduce the data to a 10 year window, in this case, between 1/1/2007 and 12/31/2016. Since the original data file was exported from a SQL database, the date fields were exported out to the millisecond ("2016-12-31 00:00:00.000") vs just out to the second ("2016-12-31 00:00:00"). Because of this, we need to use the following POSIXct format in order to convert them to a datetime. Once we've run the conversion, let's verify the format is correct.

SalesSub$PurchDate <- as.POSIXct(SalesSub$PurchDate, format = "%Y-%m-%d %H:%M:%OS")  
SalesSub$FinanceTerm <- as.POSIXct(SalesSub$FinanceTerm, format = "%Y-%m-%d %H:%M:%OS")  
SalesSub$BirthDate <- as.POSIXct(SalesSub$BirthDate, format = "%Y-%m-%d %H:%M:%OS")

SalesSub %>%   
 summarise(MinPurch = min(PurchDate),MaxPurch = max(PurchDate))

## MinPurch MaxPurch  
## 1 2005-01-01 3001-10-30

Since our dates look good, let's go ahead and drop the original Sales table since it's no longer needed. And we can now use the "PurchDate"" field to further subset our data down to the 10-year range we desire. In this case, let's just replace the "SalesSub"" table instead of creating a new version. We can then take another look at our minimum and maximum PurchDate range to confirm that it only includes the proper range.

rm(Sales)

SalesSub <- subset(SalesSub, as.POSIXct(PurchDate)>=as.POSIXct('2007-01-01 00:00:00.000') & as.POSIXct(PurchDate)<as.POSIXct('2017-01-01 00:00:00.000'), select = c(BMWRegion, BMWMarket, DealerID, DealerState, VINID, VIN, Year, Make, Model, eCode, PurchDate, NewUsed, PayType, FinanceTerm, PurchasePrice, TradeVINID, TradeVIN, TradeYear, TradeMake, TradeModel, TradeECode, CustID, CustState, Gender, BirthDate) )

SalesSub %>%   
 summarise(MinPurch = min(PurchDate),MaxPurch = max(PurchDate))

## MinPurch MaxPurch  
## 1 2007-01-01 2016-12-31

Before moving on, let's convert this data.frame into a dplyr table.

tbl\_df(SalesSub)

## # A tibble: 3,479,515 × 25  
## BMWRegion BMWMarket DealerID DealerState VINID VIN  
## \* <chr> <int> <int> <chr> <int> <chr>  
## 1 EASTERN 15 40 PA 15698790 5UXZV4C54CL754821  
## 2 EASTERN 15 40 PA 15985177 5UXFG2C59CL780025  
## 3 EASTERN 15 40 PA 8574341 WBAHL83506DT06209  
## 4 EASTERN 15 40 PA 18945332 WBAFU7C52DDU77149  
## 5 EASTERN 15 40 PA 8137869 WBAVB735X7P101187  
## 6 EASTERN 15 40 PA 18566362 WBA3B9G59DNR79004  
## 7 EASTERN 15 40 PA 11706564 WBAHN835X6DT59882  
## 8 EASTERN 15 40 PA 22750487 WBA5A7C58FD628910  
## 9 EASTERN 15 40 PA 8839989 WBAVC935X7K035066  
## 10 EASTERN 15 40 PA 13795194 WBAPK5C54AA650407  
## # ... with 3,479,505 more rows, and 19 more variables: Year <int>,  
## # Make <chr>, Model <chr>, eCode <chr>, PurchDate <dttm>, NewUsed <chr>,  
## # PayType <chr>, FinanceTerm <dttm>, PurchasePrice <dbl>,  
## # TradeVINID <int>, TradeVIN <chr>, TradeYear <int>, TradeMake <chr>,  
## # TradeModel <chr>, TradeECode <chr>, CustID <int>, CustState <chr>,  
## # Gender <chr>, BirthDate <dttm>

Now, let's take a closer look at some of the other fields to see if anything needs to be corrected before we move on to the data analysis. Let's start with the Region since this is a field that we'll definitely want to utilize in our analysis.

##### BMW Region:

SalesSub %>%   
 group\_by(BMWRegion) %>%   
 summarise(cnt = n())

## # A tibble: 6 × 2  
## BMWRegion cnt  
## <chr> <int>  
## 1 CENTRAL 470960  
## 2 Eastern 119  
## 3 EASTERN 1079551  
## 4 SOUTHERN 942224  
## 5 Western 3945  
## 6 WESTERN 982716

We see that there are different cases. All four regions have values noted with all CAPS, but two also have values using mixed case. Since R reads these as six different regions, we'll need to make them all consistent in order to get back to the four actual regions. To do this, let's set those all to CAPS and verify our update when done.

SalesSub$BMWRegion <- toupper(SalesSub$BMWRegion)   
  
SalesSub %>%   
 group\_by(BMWRegion) %>%   
 summarise(cnt = n())

## # A tibble: 4 × 2  
## BMWRegion cnt  
## <chr> <int>  
## 1 CENTRAL 470960  
## 2 EASTERN 1079670  
## 3 SOUTHERN 942224  
## 4 WESTERN 986661

Let's move onto Gender next to see if we have enough data here to utilize for our analysis.

##### Gender:

SalesSub %>%   
 group\_by(Gender) %>%   
 summarise(Cnt = n()) %>%   
 arrange(desc(Cnt))

## # A tibble: 41 × 2  
## Gender Cnt  
## <chr> <int>  
## 1 2890357  
## 2 M 382453  
## 3 F 188916  
## 4 MALE 11833  
## 5 FEMALE 5858  
## 6 N 33  
## 7 1 5  
## 8 G 5  
## 9 . 4  
## 10 43 4  
## # ... with 31 more rows

We can see that the Gender field is filled with a lot more values than just Male/M/Female/F. While we may not be able to effectively utilize the Gender variable due to most records missing this field, let's go ahead and normalize it as well. We can set all the values of just "M"" to "MALE"" and just "F"" to "FEMALE".

SalesSub$Gender <- gsub('^M$', 'MALE', SalesSub$Gender)  
SalesSub$Gender <- gsub('^F$', 'FEMALE', SalesSub$Gender)

Once this is complete, we need to remove all the invalid data, anything other than just "MALE" or "FEMALE", so that it doesn't skew any possible analysis. Let's verify this once it's done.

SalesSub$Gender <- ifelse ( grepl("FEMALE|MALE", SalesSub$Gender), SalesSub$Gender, '' )  
  
SalesSub %>%   
 group\_by(Gender) %>%   
 summarise(Cnt = n())

## # A tibble: 3 × 2  
## Gender Cnt  
## <chr> <int>  
## 1 2890455  
## 2 FEMALE 194774  
## 3 MALE 394286

Now that BMWRegion and Gender look good, let's take a summarized look at many of the other variables to see if we need to clean up any additional fields at this time.

Let's review the outputs one at a time.

##### Market:

## # A tibble: 39 × 2  
## BMWMarket Cnt  
## <int> <int>  
## 1 10 132671  
## 2 11 106572  
## 3 12 141382  
## 4 13 153756  
## 5 14 127923  
## 6 15 99170  
## 7 16 130430  
## 8 17 53174  
## 9 18 58447  
## 10 19 76145  
## # ... with 29 more rows

* Market values are clean and all values are present.

##### Dealer State:

## # A tibble: 48 × 2  
## DealerState Cnt  
## <chr> <int>  
## 1 AK 3984  
## 2 AL 30676  
## 3 AR 9538  
## 4 AZ 66120  
## 5 CA 702772  
## 6 CO 53225  
## 7 CT 57316  
## 8 DE 6502  
## 9 FL 272131  
## 10 GA 125693  
## # ... with 38 more rows

* DealerState also looks clean with all values present. We see that 47 states and 1 territory (Puerto Rico, PR) are represented. Only the District of Columbia (DC), Montana (MT), North Dakota (ND), & Wyoming (WY) do not have a BMW dealership.

##### Customer State:

## # A tibble: 117 × 2  
## CustState Cnt  
## <chr> <int>  
## 1 9015  
## 2 - 1  
## 3 686  
## 4 00 15  
## 5 11 1  
## 6 28 3  
## 7 32 2  
## 8 99 4  
## 9 AA 13  
## 10 AB 243  
## # ... with 107 more rows

* CustState, on the other hand, includes 117 unique values - way more than just the 50 US states, 1 district, 1 territory, or military codes (AA, AE, AP). While we see some lower & mixed case values and some Canadian provinces, such as BC (British Columbia), we also see many invalid values. We most likely won't utilize this field in our analysis, so for now we'll leave it be. If we were to use it, we'd want to set it to Upper case and then remove or fix any that are not valid.

##### Model Year:

## # A tibble: 51 × 2  
## Year Cnt  
## <int> <int>  
## 1 6 1  
## 2 9 1  
## 3 11 1  
## 4 12 1  
## 5 201 1  
## 6 1842 1  
## 7 1849 1  
## 8 1971 1  
## 9 1979 1  
## 10 7 2  
## # ... with 41 more rows

* Similarly for the vehicle's Year, if we decide to utilize this field, we'll need to make some corrections first. We have several years that are only 1-2 digits. These would need to be changed to 4-digit years - for example, 6 is likely 2006 and 97 would be changed to 1997. While the year 0 could be 2000, it's also just as likely to be invalid so for these, we would need to just exclude them, along with any remaining missing or invalid values.

##### Model:

## # A tibble: 387 × 2  
## Model Cnt  
## <chr> <int>  
## 1 2193  
## 2 1 Series M Coupe 727  
## 3 128 15612  
## 4 128i 16615  
## 5 128I 1  
## 6 128i Convertible 8163  
## 7 128i Coupe 8270  
## 8 135 6894  
## 9 135i 11559  
## 10 135i Convertible 3238  
## # ... with 377 more rows

* Looking at the Model summary, we can see that we are missing value for nearly 2200 records and that there are 386 other unique values. Using this alone would make analysis difficult. If we want to utilize Model, we'll first need to normalize this into a new field where we mark all records where the model starts with a '1' as a '1-series', all that start with a '3' as a 3-series, etc. We'd have to be careful in doing this as we can see that not everything that starts with a '2' is a 2-series. For example, there are models listed as '2DR ....' or '4D ....'. In doing so, we still may find we are unable to categorize every model and would then need to exclude all of the missing or invalid records from analysis.

##### Vehicle eCode:

## # A tibble: 55 × 4  
## eCode Cnt MinModelYr MaxModelYr  
## <chr> <int> <int> <int>  
## 1 161405 NA NA  
## 2 E12 2 1980 1980  
## 3 E21 10 1980 2011  
## 4 E23 8 1979 1986  
## 5 E24 46 1981 1989  
## 6 E28 50 1983 1988  
## 7 E30 207 1984 1990  
## 8 E31 86 1991 1997  
## 9 E32 147 1987 1994  
## 10 E34 674 1989 1995  
## # ... with 45 more rows

* BMW uses what they call an Engine Code (eCode for short) that actually categorizes models and year into groups of similar body styles. Here we would see multiple eCodes for the 3-series vehicles. But instead of all 2007-2017 3-series, an eCode may include only certain 3-series and only for a smaller model year window. Since BMW does marketing based on eCodes, we may want to start here instead of normalizing our models. The downside here is that we don't directly see what models/years are included in each eCode. However, this would be relevant to our customer. The biggest issue we have with this field though is that we are missing the eCode value for 161k of our 3.4 million records. This is about 5.5% so we still have data on the majority of records and could proceed.

##### Pay Type:

## # A tibble: 4 × 2  
## PayType Cnt  
## <chr> <int>  
## 1 5726  
## 2 C 937642  
## 3 F 1501621  
## 4 L 1034526

* PayType looks pretty good as is. We're missing values for only 5.7k records and the remaining are divided into the three main pay types: C=Cash, F=Finance, & L=Lease. Right away we can see that the majority of buyers, 43%, finance their purchase, but leasing, 30%, and paying in cash, 27%, aren't too far behind.

##### NewUsed Flag:

## # A tibble: 10 × 2  
## NewUsed Cnt  
## <chr> <int>  
## 1 1463  
## 2 0 30  
## 3 D 204  
## 4 F 2505  
## 5 L 126  
## 6 M 193  
## 7 N 1986088  
## 8 O 42116  
## 9 U 1445957  
## 10 W 833

* NewUsed is used to designate if a vehicle purchased is either a New vehicle or a Used vehicle. But we are also seeing many other fields, such as O for Other, D for Demo, & F for Fleet. Some of the designations appear to be invalid or missing and others have unknown meaning - such as M & W. We could probably consider all non-New types as Used; however, since some of these may come with a New warranty and maintenance plan - being sold like a New vehicle type even though it's been used as a dealer demo, we'll exclude any sale not specifically designated as either New or Used.

## New & Used Sales Data

Let's now look at New vs Used sales for BMW vehicles across the Nation and within the four BMW regions. With this information, we can see if there are any national or regional trends that warrant further investigation. The first thing we'll do is subset our data to just include vehicle sales flagged as either New or Used as we discussed above. In this case, some of these Used vehicles may actually be Certified Pre-Owned (CPO) vehicles that have undergone additional inspections and include a warranty with purchase. Unfortunately, this information isn't available to us so we are unable to differentiate between these higher valued CPO vehicles and traditional Used vehicles.

Let's go ahead and subset the New and Used only vehicles into their own table. We can also use this opportunity to whittle down our field list even further.

SalesNewUsed <-   
 subset (SalesSub, NewUsed=="N"|NewUsed=="U", select = c(BMWRegion, BMWMarket, VINID, Year, Make, Model, eCode, PurchDate, NewUsed, PayType, FinanceTerm, PurchasePrice, TradeVINID, TradeVIN, TradeYear, TradeMake, TradeModel, TradeECode) )

Now, all of our Purchase Dates are Month/Day/Year and over the course of 10-years this translates into a lot of data points. Because this would result in plots that are very difficult, if not impossible, to read and we don't really care about sales from one specific day vs another, we should pull apart our dates into separate purchase Month, Day, & Year columns. This will allow us to properly analyze the data.

SalesNewUsed <- separate(SalesNewUsed, PurchDate, c("PurYear","PurMo","PurDay"), remove=FALSE)

While this is great for looking at either annual or seasonal trends (if we group by Months only), if we want to group by purchase Year and Month, we'll need to go one step further and combine those two separate columns into one new combined column.

SalesNewUsed <- unite(SalesNewUsed, PurYearMo, c(PurYear,PurMo), sep="-", remove=FALSE)

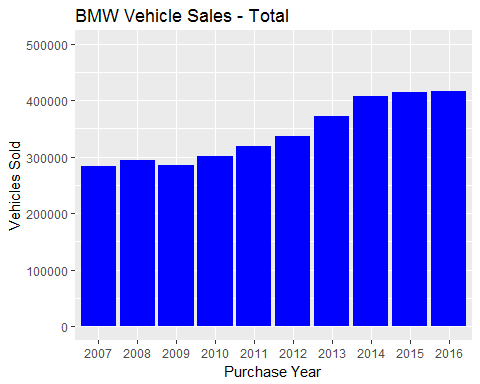
Here we can see the final result of our split and combining efforts.

head(SalesNewUsed)

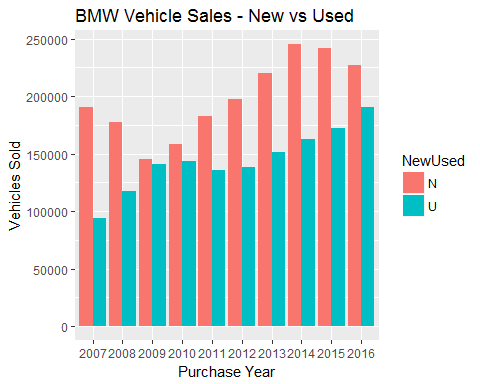
## BMWRegion BMWMarket VINID Year Make Model eCode PurchDate  
## 1 EASTERN 15 15698790 2012 BMW X5 E70 2011-08-22  
## 2 EASTERN 15 15985177 2012 BMW X6 E71 2011-10-07  
## 3 EASTERN 15 8574341 2006 BMW 750i E65 2012-01-07  
## 4 EASTERN 15 18945332 2013 BMW 535 F07/F10/F11 2013-11-27  
## 5 EASTERN 15 8137869 2007 BMW 335i E90/91/92/93 2007-05-09  
## 6 EASTERN 15 18566362 2013 BMW 335 F30 2013-03-15  
## PurYearMo PurYear PurMo PurDay NewUsed PayType FinanceTerm PurchasePrice  
## 1 2011-08 2011 08 22 N C 2011-09-22 46875.00  
## 2 2011-10 2011 10 07 N C 2011-11-07 62500.00  
## 3 2012-01 2012 01 07 U F 2017-01-07 27900.00  
## 4 2013-11 2013 11 27 N L 2016-11-27 60745.36  
## 5 2007-05 2007 05 09 N C 2007-06-09 43845.00  
## 6 2013-03 2013 03 15 N F 2016-03-15 51290.00  
## TradeVINID TradeVIN TradeYear TradeMake TradeModel TradeECode  
## 1 NA NA   
## 2 NA NA   
## 3 NA NA   
## 4 1710225 WBADD6327WGT94406 1998 BMW 528i E39  
## 5 NA NA   
## 6 5838041 WBAVD33546KL52088 2006 BMW 330xi E90/91/92/93

Now that we have that all taken care of, let's take a cursory look at the Total BMW Vehicle Sales, as well as the breakouts for the New (N) vs Used (U) categories.

options(scipen=3) #to show the y values as full numbers vs scientific (0e+00, 1e+05, etc)  
ggplot(SalesNewUsed, aes(x=PurYear)) +   
 geom\_histogram(stat="count", fill="blue") +  
 scale\_y\_continuous(limits = c(0,500000)) +  
 labs(title="BMW Vehicle Sales - Total", x="Purchase Year", y="Vehicles Sold")



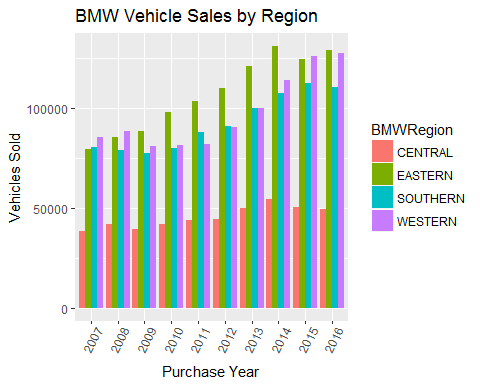
ggplot(SalesNewUsed, aes(x=PurYear, fill=NewUsed)) +   
 geom\_histogram(stat="count", position = "dodge") +  
 labs(title="BMW Vehicle Sales - New vs Used", x="Purchase Year", y="Vehicles Sold")



We can see that BMW sales climbed between 2011 and 2014, but have once again steadied out. In regards to New vs Used, in 2009 the sales of New and Used were almost equal with New sales at their lowest during this 10-year period. From that point on; however, Used sales were fairly steady with New sales climbing rapidly. It's only in the last year or two do we see that Used sales are climbing again and New are on a decline. We saw this same occurrence from 2007 to 2009. This could point to the economy valuing Used or perhaps that their CPO (Certified Pre-Owned) program has become more appealing in recent years and has started to impact their New vehicle sales. On the plus side, total sales are still at their highest.

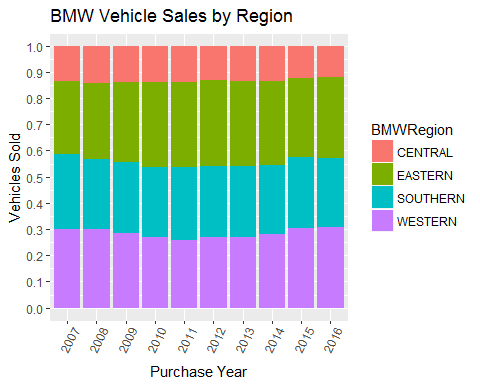
Let's now take a look at some Regional data. We start by looking at total sales by region.

ggplot(SalesNewUsed, aes(x=PurYear, fill=BMWRegion)) +   
 geom\_histogram(stat="count", position = "dodge") +  
 theme(axis.text.x = element\_text(angle=65, vjust=0.6)) +   
 labs(title="BMW Vehicle Sales by Region", x="Purchase Year", y="Vehicles Sold")



It's clear here that the Central region only accounts for a small portion of the sales. It's only around 50% of any of the other three Regions. Let's stack these bars to get an idea of each regions percentage of the annual total. Perhaps this will help us get a better visualization of how they compare.

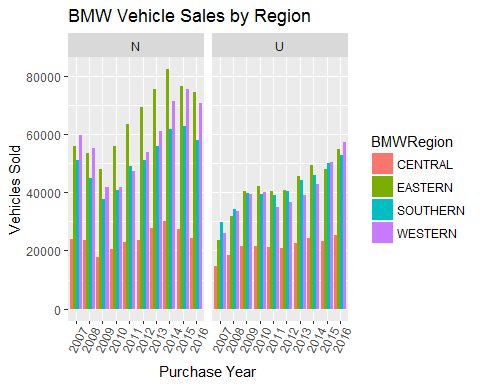
ggplot(SalesNewUsed, aes(x=PurYear, fill=BMWRegion)) +   
 geom\_histogram(stat="count", position="fill") + # vs "stack"  
 scale\_y\_continuous(breaks=seq(0, 1, .1)) + #c(0,.1, .2, .3, .4, .5, .6, .7, .8, .9, 1.00)) +  
 theme(axis.text.x = element\_text(angle=65, vjust=0.6)) +   
 labs(title="BMW Vehicle Sales by Region", x="Purchase Year", y="Vehicles Sold")



So looking at this graph, we can see that each region is fairly consistent with it's contribution to sales over the years. The Western & Southern regions appear to pull in between 25% and 30%, with the Eastern region typically pulling ahead slightly with 25-35% of sales. The Central region by far is the notable region with sales always under 15%. It's clear from this plot that it's this region that could benefit from more specific marketing efforts.

Now let's look at the Regional sales for New vs Used.

ggplot(SalesNewUsed, aes(x=PurYear, fill=BMWRegion)) +   
 geom\_histogram(stat="count", position = "dodge") +  
 facet\_grid(. ~ NewUsed) +  
 theme(axis.text.x = element\_text(angle=65, vjust=0.6)) +   
 labs(title="BMW Vehicle Sales by Region", x="Purchase Year", y="Vehicles Sold")



Here we can observe that even within each region, we're seeing a similar curve that we saw with each of the New and Used sales graphs. We see that the Eastern Region generally takes the lead in both categories. However, we can also see that the Eastern Region's lead over the Southern & Western regions in total sales between 2010 and 2014 can be accounted for in nearly all New purchases. Their edge over the Southern & Western regions in Used sales is very minor. The Eastern region saw a decline in their New sales between 2014 and 2015 though, while the Southern & Western regions continued to climb. 2016 continued to result in a decline in New sales for each region, but this was offset by the increase in Used sales for each. BMW should refocus their efforts on selling New vehicles to try to turn around their declining trend, while trying to leverage their recent increase in Used sales.

## Sales by eCode

Let's now take a look at how sales are performing at an eCode level. Since this analysis could affect BMW's manufacturing choices, we really only need to look at New sales for this. Used sales would be based on the supply from what customer's are trading in and BMW would have very little influence on that. If we can determine how long it takes for a customer to trade-in their old vehicle, then perhaps we can look at predicting what vehicles will be available for resale in the future. The impact on this in regards to BMW's sales or manufacturing would be minimal so for now, let's focus on the New sales only. We can also remove any records where the eCode is not present. I also observed that two records has an eCode of R56. These are actually MINI vehicles so they must have been miscoded. We will exclude them as well.

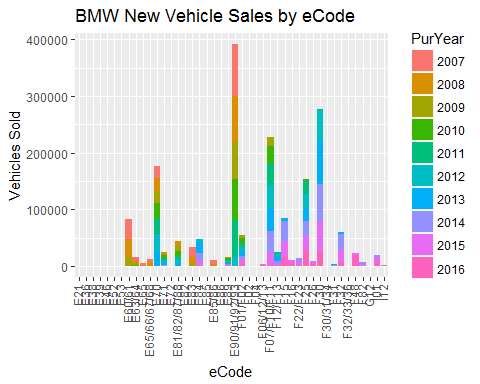
SalesNew <- subset(SalesNewUsed, NewUsed=='N'& eCode!="" & eCode!="R56")

SalesNew %>%  
 #filter(eCode!="") %>%  
 group\_by(eCode) %>%  
 summarise(cntSales = n()) %>%  
 arrange(cntSales)

## # A tibble: 44 × 2  
## eCode cntSales  
## <chr> <int>  
## 1 E21 2  
## 2 F02 2  
## 3 E52 7  
## 4 E36 11  
## 5 E38 16  
## 6 F04 65  
## 7 F32/33/36 117  
## 8 F30/31/34 119  
## 9 E39 123  
## 10 E72 166  
## # ... with 34 more rows

We can plot this data as is, showing each eCode and the total volume color-coded by purchase year. This can give us a visual on which eCodes are selling well now as well as in the past. We will also be able to see some eCodes that have dropped out of production this way.

options(scipen=3)  
ggplot(SalesNew, aes(x=eCode, fill=PurYear)) +  
 geom\_bar() +  
 theme(axis.text.x = element\_text(angle=90, vjust=.3, hjust=1)) +   
 labs(title="BMW New Vehicle Sales by eCode", x="eCode", y="Vehicles Sold")



So looking at this graph, it's clear that the E90/91/92/93 eCode has outsold all others during this time period. From experience, I know that these are 3-series vehicles. But we can also reference this website to get a look at these eCodes for BMW and MINI, referred to on the web page as "Chassis Code": [<http://www.burgertuning.com/BMW_chassis_codes.html>]. This shows us that these were produced from roughly 2005-2013. Since our data starts at 2007 purchases, we may see a few 2006's in there, but it should mostly comprise of 2007-2013 model years. Interestingly enough, we see that the 2nd highest bar belongs to the F30. This too consists of 3-series models, but this time ranging from 2012-present model years. We can see too by the coloring, that the F90-93's stopped around 2012 and the F30's started around then. So even without converting our models into model categories, we can see that the 3-series has, by far, outsold any other BMW series over the last 10-years. This is likely due to it's lower price point and therefore affordability to a larger audience. The 3rd largest bar belongs to the F07/F10/F11 grouping. We can find that these belong to the 5-series models manufactured for 2009-2016.The E70's come in 4th - these are their 2007-2013 X5 models, which belong to the luxury SUV class of vehicles. This eCode was then replaced by the F15, representing the 2014-present X5's. That particular bar actually comes in 6th. The last notable bar (5th tallest) belongs to the F25's, which are the 2010-present X3's - also an SUV, but priced more moderately. The E83 bar, which is quite small (< 50,000), represents the earlier (2004-2010) X3's. Again, our data will represent mostly 2007-2010's in this count.

In order to compare complete model year ranges in all four categories, we need to find the earlier 5-series vehicles. Looking this up on the website, shows us that the E60/61 category included 5-series model up until 2010, so there was some cross-over with the F07/10/11's. Let's add those counts into our chart too. Scanning through the site's codes, I see that the F18's are also a 5-series sedan, but with a long wheel base. We should include these counts, but it appears we have none. And if we really want to make sure we're including everything, we can grab the E53's as well. These are 2000-2006 X5's and since we have counts for this eCode, albeit only 717, we can include them too.

|  |  |  |  |
| --- | --- | --- | --- |
| eCode Groups | Model Years^ | Category Description | Total New Sales Count |
| E90/91/92/93 & F30 | 2005-2016 | 3-series vehicles | 670080 (*392370+277710*) |
| F07/F10/F11 & E60/61 | 2003-2016 | 5-series vehicles | 310819 (*227741+83078*) |
| E70, F15 & E53 | 2000-2016 | X5 SUV's | 262452 (*176578+85157+717*) |
| F25 & E83 | 2004-2016 | X3 SUV's | 154102 |

^ Since our data is based on New sales purchased from 2007 to 2016, we may have some sales of model years older than 2007, but not many. For the most part, these counts represent 2007+ model years.

Now that we have all the counts for the top categories summarized, we can make some observations. The best-selling 3-series outsold the next highest category, the 5-series, by more than double! It's also interesting to point out here that while in the sedan/coupe category, the more moderately priced 3-series beat the higher priced 5-series, but in the SUV lineup, the more expensive X5's are the clear winner over the X3's with 63% of their SUV sales.

We can also take a look at the "Purchase Price" field for each of these categories. In order to get statistics for each group, we first need to create a new field to house those categories. Let's call this new field "VehCategory" & set these to "3-Series", "5-Series", "X5 SUV's", "X3 SUV's", and all remaining eCodes to "Others".

SalesNew <- SalesNew %>%   
 mutate(VehCategory=eCode)  
   
SalesNew <- SalesNew %>%   
 mutate(VehCategory=ifelse(VehCategory=="E90/91/92/93"|VehCategory=="F30","3-Series", VehCategory)) %>%   
 mutate(VehCategory=ifelse(VehCategory=="F07/F10/F11"|VehCategory=="E60/61","5-Series", VehCategory)) %>%   
 mutate(VehCategory=ifelse(VehCategory=="E70"|VehCategory=="F15"|VehCategory=="E53","X5 SUV's",VehCategory)) %>%  
 mutate(VehCategory=ifelse(VehCategory=="F25"|VehCategory=="E83","X3 SUV's",VehCategory)) %>%  
 mutate(VehCategory=ifelse(VehCategory!="3-Series" & VehCategory!="5-Series" & VehCategory!="X5 SUV's"& VehCategory!="X3 SUV's","Others",VehCategory))   
  
head(SalesNew)

## BMWRegion BMWMarket VINID Year Make Model eCode PurchDate  
## 1 EASTERN 15 15698790 2012 BMW X5 E70 2011-08-22  
## 2 EASTERN 15 15985177 2012 BMW X6 E71 2011-10-07  
## 3 EASTERN 15 18945332 2013 BMW 535 F07/F10/F11 2013-11-27  
## 4 EASTERN 15 8137869 2007 BMW 335i E90/91/92/93 2007-05-09  
## 5 EASTERN 15 18566362 2013 BMW 335 F30 2013-03-15  
## 6 EASTERN 15 22750487 2015 BMW 528 F07/F10/F11 2015-06-30  
## PurYearMo PurYear PurMo PurDay NewUsed PayType FinanceTerm PurchasePrice  
## 1 2011-08 2011 08 22 N C 2011-09-22 46875.00  
## 2 2011-10 2011 10 07 N C 2011-11-07 62500.00  
## 3 2013-11 2013 11 27 N L 2016-11-27 60745.36  
## 4 2007-05 2007 05 09 N C 2007-06-09 43845.00  
## 5 2013-03 2013 03 15 N F 2016-03-15 51290.00  
## 6 2015-06 2015 06 30 N L 2018-06-30 54150.00  
## TradeVINID TradeVIN TradeYear TradeMake TradeModel TradeECode  
## 1 NA NA   
## 2 NA NA   
## 3 1710225 WBADD6327WGT94406 1998 BMW 528i E39  
## 4 NA NA   
## 5 5838041 WBAVD33546KL52088 2006 BMW 330xi E90/91/92/93  
## 6 NA NA   
## VehCategory  
## 1 X5 SUV's  
## 2 Others  
## 3 5-Series  
## 4 3-Series  
## 5 3-Series  
## 6 5-Series

So now that we have our VehCategory field, let's check out both the average (mean) price, as well as the median (middle) price points. In order to do this correctly, we need to filter out all records without a purchase price. At this time, we can also exclude the "Others" category since it contains multiple categories & therefore the price comparisons won't be meaningful.

SalesNew %>%  
 filter(!is.na(PurchasePrice), VehCategory != "Others") %>%  
 group\_by(VehCategory) %>%  
 summarise(AvgPurPrice=mean(PurchasePrice), MedianPurPrice=median(PurchasePrice), SalesCnt=n())

## # A tibble: 4 × 4  
## VehCategory AvgPurPrice MedianPurPrice SalesCnt  
## <chr> <dbl> <dbl> <int>  
## 1 3-Series 48263.48 41990.00 559485  
## 2 5-Series 60708.66 55370.00 257054  
## 3 X3 SUV's 46661.37 45605.27 162102  
## 4 X5 SUV's 63522.59 59075.05 226351

This confirms for us that our 3-Series and X3's are at the lower price points and the 5-Series and X5's are about $13,000 higher. But given we should review this data first to see if we have any outliers that should be removed. So let's make a new table first excluding the "Others" and those without a price.

SalesNewPop <- SalesNew %>%  
 filter(VehCategory != "Others",!is.na(PurchasePrice))

Then we can review the minimum, mean, and maximum pricing for each category.

SalesNewPop %>%  
 group\_by(VehCategory) %>%  
 summarise(minPrice=min(PurchasePrice), PriceMean=mean(PurchasePrice), maxPrice=max(PurchasePrice))

## # A tibble: 4 × 4  
## VehCategory minPrice PriceMean maxPrice  
## <chr> <dbl> <dbl> <dbl>  
## 1 3-Series -0.91 48263.48 7822000  
## 2 5-Series -4420.00 60708.66 10000000  
## 3 X3 SUV's -0.91 46661.37 5130466  
## 4 X5 SUV's -0.91 63522.59 7846000

Well, it certainly looks like we have some bad data points remaining. We have minimums that are less than $0. Since these are new vehicles, we would expect to see a decent price point here. Now it's possible that some dealers are removing the price of a trade-in and only including the difference or they're entering a down payment or monthly payment instead. These will skew our data so we want to be sure to remove anything that is too low. With BMW, it's safe to say anything under $25,000 is too low, but to be sure, we can search this site for the Original MSRP on each category for the earliest model year: [<http://usnews.rankingsandreviews.com/cars-trucks/bmw/3-series/2007>] $32,400 - $40,800; $32,700 - $49,500 for 2008 [<http://usnews.rankingsandreviews.com/cars-trucks/bmw/5-series/2008>] $44,600 - $58,800 [<http://usnews.rankingsandreviews.com/cars-trucks/bmw/x3/2007>] $38,000 - $38,000 [<http://usnews.rankingsandreviews.com/cars-trucks/bmw/x5/2007>] $45,900 - $54,500; $46,200 - $54,800 for 2008

Unfortunately, they had no listings for 2007 5-series, but looking at the the 2007 to 2008 pricing on the 3-series and X5's, we can see that these would have only gone up by about $300, so we can assume the 2007 5-series started at $44,300. Now we also know that consumers don't always pay the MSRP so we don't want to use these as our minimums, but rather as a reference point. Let's go with 25% below MSRP as our starting point.

On the other hand, we clearly have data points that are much too high, such as the $10,000,000 5-series. It's possible that some of the others were just missing their decimal points and are could be fixed by dividing those values by 10; however, since we don't want to assume this is the case, we should probably just exclude these as well. We also know that there is variation in the different model types within each category and a customer could have chosen to add on several upgrades. We can use the same website to check out the 2016 MSRP's for each. 3-Series: $33,150 - $63,500 5-Series: $50,200 - $94,100 X3: $38,950 - $46,800 X5: $54,700 - $98,800

So we can see that the ranges for 2016 models have increased significantly over the available ranges in the 2007 models. For our upper end, let's do a similar 25% over the high end.

SalesNewPop <- SalesNewPop %>%  
 filter( (VehCategory=="3-Series" & PurchasePrice >= 24300 & PurchasePrice <= 79375) |   
 (VehCategory=="5-Series" & PurchasePrice >= 33225 & PurchasePrice <= 117625) |   
 (VehCategory=="X3 SUV's" & PurchasePrice >= 28500 & PurchasePrice <= 58500) |   
 (VehCategory=="X5 SUV's" & PurchasePrice >= 34425 & PurchasePrice <= 123500) )

## BMWRegion BMWMarket VINID Year Make Model eCode PurchDate  
## 1 EASTERN 15 15698790 2012 BMW X5 E70 2011-08-22  
## 2 EASTERN 15 18945332 2013 BMW 535 F07/F10/F11 2013-11-27  
## 3 EASTERN 15 8137869 2007 BMW 335i E90/91/92/93 2007-05-09  
## 4 EASTERN 15 18566362 2013 BMW 335 F30 2013-03-15  
## 5 EASTERN 15 22750487 2015 BMW 528 F07/F10/F11 2015-06-30  
## 6 EASTERN 15 8839989 2007 BMW 328xi E90/91/92/93 2007-08-31  
## PurYearMo PurYear PurMo PurDay NewUsed PayType FinanceTerm PurchasePrice  
## 1 2011-08 2011 08 22 N C 2011-09-22 46875.00  
## 2 2013-11 2013 11 27 N L 2016-11-27 60745.36  
## 3 2007-05 2007 05 09 N C 2007-06-09 43845.00  
## 4 2013-03 2013 03 15 N F 2016-03-15 51290.00  
## 5 2015-06 2015 06 30 N L 2018-06-30 54150.00  
## 6 2007-08 2007 08 31 N L 2010-08-31 36600.00  
## TradeVINID TradeVIN TradeYear TradeMake TradeModel TradeECode  
## 1 NA NA   
## 2 1710225 WBADD6327WGT94406 1998 BMW 528i E39  
## 3 NA NA   
## 4 5838041 WBAVD33546KL52088 2006 BMW 330xi E90/91/92/93  
## 5 NA NA   
## 6 4531494 5UXFA13505LY05831 2005 BMW X5 3.0i E53  
## VehCategory  
## 1 X5 SUV's  
## 2 5-Series  
## 3 3-Series  
## 4 3-Series  
## 5 5-Series  
## 6 3-Series

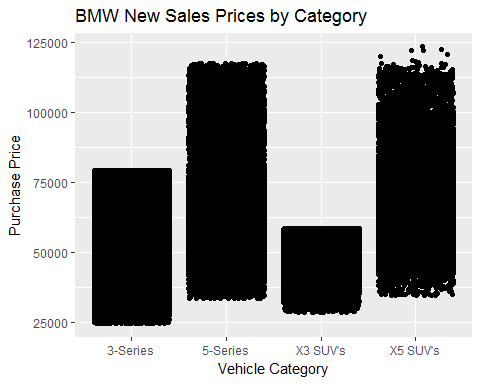
Now that we've removed our outliers, let's recheck our values.

SalesNewPop %>%  
 group\_by(VehCategory) %>%  
 summarise(AvgPurPrice=mean(PurchasePrice), MedianPurPrice=median(PurchasePrice), SalesCnt=n())

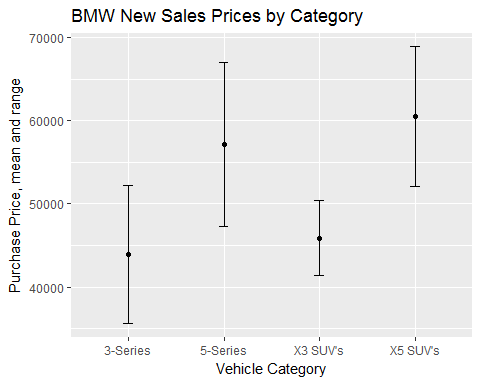
## # A tibble: 4 × 4  
## VehCategory AvgPurPrice MedianPurPrice SalesCnt  
## <chr> <dbl> <dbl> <int>  
## 1 3-Series 43893.12 42100.00 545886  
## 2 5-Series 57105.03 55545.00 251249  
## 3 X3 SUV's 45873.98 45659.49 157950  
## 4 X5 SUV's 60483.31 59215.00 222693

So we see that the values changed a little, more so with the Mean than with the Median, which is to be expected. But from here we can plot the sales prices to get a better idea of how they range within each category, as well as across the categories.

ggplot(SalesNewPop, aes(x=VehCategory, y=PurchasePrice)) +  
 geom\_point(position = position\_jitter(0.4)) +  
 stat\_summary(fun.y=mean, geom="point") +   
 labs(title="BMW New Sales Prices by Category", x="Vehicle Category", y="Purchase Price")



ggplot(SalesNewPop, aes(x=VehCategory, y=PurchasePrice)) +  
 stat\_summary(fun.y=mean, geom="point") +   
 stat\_summary(fun.data = mean\_sdl, fun.args = list(mult=1), geom="errorbar", width = 0.1) +   
 labs(title="BMW New Sales Prices by Category", x="Vehicle Category", y="Purchase Price, mean and range")

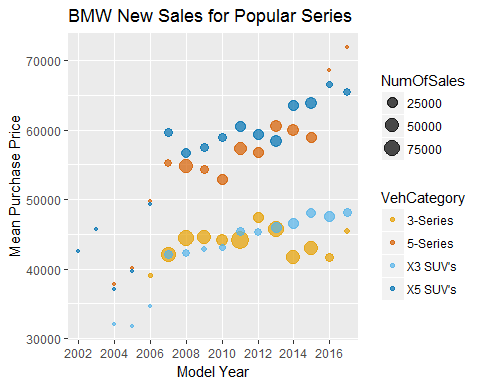


Our first plot is very solid with more variance on the end points, especially evident with the X5's. The 2nd plot may be a bit easier for us to interpret since it better points out the Means along their ranges. But we are looking at a 10-year period here and as we saw above, that price point grew a lot over the last 10 years. Perhaps we'd be better breaking each of these up by Model Year. We can do this by first pulling out the data summaries that we may want to look at. Since we're using vehicle year, we should also make sure that those values are valid by excluding anything out of normal ranges, in this case I used 1980-2017. We can also exclude the "Others" category since we don't want to compare their pricing.

SalesNewPopSumm <- SalesNewPop %>%   
 filter(Year>=1980 & Year<=2017 & VehCategory!="Others") %>%   
 group\_by(VehCategory, Year) %>%   
 summarise(NumOfSales=n(), MinPrice=min(PurchasePrice), MedianPrice=median(PurchasePrice), AvgPrice=mean(PurchasePrice), MaxPrice=max(PurchasePrice))

Now let's take a closer look at how the sales counts compare to the mean purchase price for each model year and for each vehicle category.

ggplot(SalesNewPopSumm, aes(x=Year, y=AvgPrice, size=NumOfSales, colour=VehCategory)) +  
 geom\_point(alpha=.7) +   
 scale\_x\_discrete(limits=seq(2002, 2017, by = 2)) +   
 scale\_colour\_manual(values=c("#E69F00", "#D55E00","#56B4E9","#0072B2")) +   
 labs(title="BMW New Sales for Popular Series", x="Model Year", y="Mean Purchase Price")



So here we can now see that over the last 10-years, the average price relative to the volume of sales for each model year of our four most popular vehicle categories. The orange colors represent the sedans/coupes & the blue colors are the SUV's. Now since our data represents Jan 1st 2007 through Dec 31st 2016 sales, we see that the number of sales for models prior to 2007 or after 2016 are relatively small. This is to be expected given our New sales date ranges, so our primary focus should be on the 2007 to 2016 model years, but this still can give us an idea of the prices at which the other model years sold. The older model years could have been heavily discounted though since they're selling past their intended model year roll-out, so we don't want to put too much stock in those prices.

It is interesting to note the drop in the mean pricing of the sedan/coupes (orange colors) from 2013 to 2014 model years. It's quite significant, especially in the 3-series. Perhaps they dropped specific models within those series. We do see that for 2016, the 5-series price jumped considerably and the number of sales dropped with it. The decrease in sales for the 3-series could be offset by the increase in sales of the X3 SUV - indicating a move of consumer preference from sedans/coupes to SUV's. We can see that the X3's have done a fairly stead job of increasing sales. Since we learned earlier that in 2016 BMW's New sales went down, with Used sales making up the loss, it's not a surprise to see the smaller points appear in 2016.

Another interesting point is the large volume of sales for the 2011 3-series. The points before and after are fairly similar, but this one is much larger. Perhaps the majority of New sales increase from 2010 to 2011 can be explained by a push of these models by BMW or perhaps by the large jump in price of the 5-series models during that same time.

### Side Note on the eCodes by PurYear

If we were to look back at our initial count summary, we have eCodes with sales in only the hundreds, while there are many others that are over 100,000. Of course, this is grouping all 10-years together so this summary won't give us an idea of how the counts have changed over time. In order to do that, we need to consider the purchase year as well. Let's group this out by eCode and PurYear.

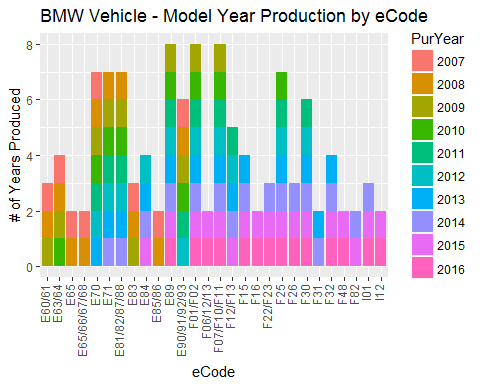
SalesNew %>%  
 group\_by(eCode, PurYear) %>%  
 summarise(cntSales = n()) %>%  
 arrange(PurYear, cntSales)

## Source: local data frame [249 x 3]  
## Groups: eCode [44]  
##   
## eCode PurYear cntSales  
## <chr> <chr> <int>  
## 1 F15 2007 1  
## 2 F26 2007 1  
## 3 F30 2007 1  
## 4 F48 2007 1  
## 5 E81/82/87/88 2007 2  
## 6 E52 2007 4  
## 7 E38 2007 6  
## 8 E39 2007 68  
## 9 E86 2007 221  
## 10 E85 2007 267  
## # ... with 239 more rows

So now we can see that we still have some eCodes that had very low sales volumes within each year. In order to produce a cleaner plot, we can ignore the records where the counts for that eCode and PurYear are under 900 and focus on those that had higher sales volumes. I've chosen 900 since for most years we go from one eCode with no more than say 500 or 600 and then jump to the next with closer to 2000. There was one year where we had counts at 955 so I want to include them.

ECodePlot <- SalesNew %>%  
 group\_by(eCode, PurYear) %>%  
 summarise(cntSales = n()) %>%  
 arrange(PurYear, cntSales) %>%  
 filter(cntSales >= 900)

options(scipen=3)  
ggplot(ECodePlot, aes(x=eCode, fill=PurYear)) +  
 geom\_bar() +  
 theme(axis.text.x = element\_text(angle=90, vjust=0.2, hjust=1)) +   
 labs(title="BMW Vehicle - Model Year Production by eCode", x="eCode", y="# of Years Produced")



This plot shows us which years the most popular eCodes were produced.

## Predicting 'New' Sales for 2017

Now let's see if we can use our data to predict 2017 New sales. First we need to create a new data set to use for our predictions. We'll stick to only New sales again, but let's keep in the "Others" category. I'm also going to limit this data to just model years of 2006 and newer since the calendar year of 2007 would have seen the release of the new 2008's, any 2006's here would have been old inventory. I'm primarily interested in total sales, but in case we want to dig deeper, I'm grouping this into many categories using VehCategory, BMWRegion, as well as PurYearMo, and just PurYear. From there, I can further summarize the data to just look at just Total, just Regional, or just VehCategory statistics & use those tables from my modelling.

SalesNewForPredModel <-   
 SalesNew %>%   
 filter(Year >= 2007) %>%   
 group\_by(PurYear, PurYearMo, BMWRegion, VehCategory) %>%   
 summarise(SalesCnt=n())  
head(SalesNewForPredModel)

## Source: local data frame [6 x 5]  
## Groups: PurYear, PurYearMo, BMWRegion [2]  
##   
## PurYear PurYearMo BMWRegion VehCategory SalesCnt  
## <chr> <chr> <chr> <chr> <int>  
## 1 2007 2007-01 CENTRAL 3-Series 516  
## 2 2007 2007-01 CENTRAL 5-Series 284  
## 3 2007 2007-01 CENTRAL Others 104  
## 4 2007 2007-01 CENTRAL X3 SUV's 186  
## 5 2007 2007-01 CENTRAL X5 SUV's 226  
## 6 2007 2007-01 EASTERN 3-Series 1688

Now we did gather some Purchase Price summary data earlier, but I'm not going to include any purchase price data since, as we saw before, it varies greatly within each vehicle category and our pricing data wasn't consistent enough to use as a factor in determining sales counts.

Now we can get our sales data into the summary level we want to model first and then add in the economic data. The main thing we're looking for here is just overall BMW sales. So we can re-summarize our table 'SalesNewForPredModel' to be grouped by YearMo. Instead of counting the sales, we now need to sum the already grouped counts.

SalesForModelByYearMo <-   
 SalesNewForPredModel %>%   
 group\_by (PurYearMo) %>%   
 summarise(SalesCnt=sum(SalesCnt))

So in addition to our sales counts, we'll also need to add some economic factors in order to help create a model that can be used to predict sales. These will be all of our Independent variables. To do this, I've pulled data from Quandl from the "Federal Reserve Economic Data" databases [<https://www.quandl.com/data/FRED-Federal-Reserve-Economic-Data?keyword>=] and compiled them into a single csv since all values are based on the first of each month. This will make it easier to manage the data and combine it with our sales data.

EconData <- read.csv("EconomicData.csv", stringsAsFactors = FALSE, as.is=TRUE, colClasses=c("Year"="character", "Mo"="character"))

Now that we have our data imported, let's review the fields it contains.

* PrimeRate - the U.S. Prime Interest Rate posted by a majority of top-25 commercial banks
* Auto48MoRate - the Finance Rate on Consumer Installment Loans at Commercial Banks for NEW AUTO 48 month loans
* CPI\_ALL - Consumer Price Index for all urban consumers, all items (including food and energy), seasonally adjusted
* RealDispIncoBillions - Real Disposable Personal Income in Billions of Dollars, seasonally adjusted annual rate
* PersConsBillions - Personal Consumption Expenditures in Billions of Dollars, seasonally adjusted annual rate
* PerSavRate - Personal Savings Rate, personal savings as a percent of disposable income, seasonally adjusted annual rate
* UnempThousands - Unemployment, Thousands of Persons, age 16yrs or older, seasonally adjusted
* USAutoSalesThousUnadj - Motor Vehicle Retail Sales: Domestic & Foreign Autos, Thousands of Units, NOT seasonally adjusted annual rate
* USAutoSalesMillionsAdj - Motor Vehicle Retail Sales: Domestic & Foreign Autos, Millions of Units, seasonally adjusted annual rate

We have two versions of the U.S. Automotive sales. One is seasonally adjusted and the other is not. Which do we choose? Well, since our BMW New Sales counts are NOT seasonally adjusted and we want to be able to compare the total U.S. Auto sales with the BMW vehicle sales, we should keep them the same.

Let's take a quick look at our two tables before joining them so that we make sure we have the correct fields being matched.

head(EconData)

## DATE Year Mo YearMo PrimeRate Auto48MoRate CPI\_ALL  
## 1 12/1/2016 2016 12 2016-12 3.5 4.45 243.032  
## 2 11/1/2016 2016 11 2016-11 3.5 4.45 242.348  
## 3 10/1/2016 2016 10 2016-10 3.5 4.25 241.863  
## 4 9/1/2016 2016 9 2016-09 3.5 4.25 241.002  
## 5 8/1/2016 2016 8 2016-08 3.5 4.25 240.301  
## 6 7/1/2016 2016 7 2016-07 3.5 4.33 239.810  
## RealDispIncoBillions PersConsBillions PerSavRate UnempThousands  
## 1 12791.1 13032.1 5.4 7529  
## 2 12772.2 12969.0 5.6 7409  
## 3 12770.5 12940.2 5.7 7740  
## 4 12742.7 12894.9 5.6 7904  
## 5 12726.5 12804.7 6.0 7853  
## 6 12717.8 12796.8 5.8 7749  
## USAutoSalesThousUnadj USAutoSalesMillionsAdj  
## 1 599.4 6.971  
## 2 509.3 6.880  
## 3 506.4 6.812  
## 4 555.5 7.000  
## 5 579.9 6.540  
## 6 590.7 6.885

head(SalesForModelByYearMo)

## # A tibble: 6 × 2  
## PurYearMo SalesCnt  
## <chr> <int>  
## 1 2007-01 12324  
## 2 2007-02 12286  
## 3 2007-03 16974  
## 4 2007-04 15885  
## 5 2007-05 17384  
## 6 2007-06 15771

In this case, we want to match the Sales data's 'PurYearMo' to the Econ data's 'YearMo'. We can do a left join on these to grab everything from our Sales table and any matches to our Econ table. We'll put this into it's own table as well & see what returns.

SalesEconForModelByYearMo <-  
 left\_join(SalesForModelByYearMo,EconData,by=c("PurYearMo" = "YearMo")) # will match S.a to E.b & only keep S.a  
  
head(SalesEconForModelByYearMo)

## # A tibble: 6 × 14  
## PurYearMo SalesCnt DATE Year Mo PrimeRate Auto48MoRate CPI\_ALL  
## <chr> <int> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 2007-01 12324 1/1/2007 2007 1 8.25 7.92 203.437  
## 2 2007-02 12286 2/1/2007 2007 2 8.25 7.74 204.226  
## 3 2007-03 16974 3/1/2007 2007 3 8.25 7.74 205.288  
## 4 2007-04 15885 4/1/2007 2007 4 8.25 7.74 205.904  
## 5 2007-05 17384 5/1/2007 2007 5 8.25 7.92 206.755  
## 6 2007-06 15771 6/1/2007 2007 6 8.25 7.92 207.234  
## # ... with 6 more variables: RealDispIncoBillions <dbl>,  
## # PersConsBillions <dbl>, PerSavRate <dbl>, UnempThousands <int>,  
## # USAutoSalesThousUnadj <dbl>, USAutoSalesMillionsAdj <dbl>

Well, we don't want to utilize the individual 'DATE', 'Year', or 'Mo' fields from the Econ data as these would not be independent variables, plus we already have the 'PurYearMo' field. So let's remove those before we build our model.

SalesEconForModelByYearMo <-   
 select(SalesEconForModelByYearMo, -(DATE:Mo))

Alright, we're now ready to start on our model. We should start by splitting the data into a 'training' and 'test' data set. We'll do this by Purchase Year of less than 2014 for the training (7 years total) and then greater than or equal to 2014 for the testing data (3 years total).

SalesTrain <- subset(SalesEconForModelByYearMo, PurYearMo<="2013-12")  
SalesTrain

## # A tibble: 84 × 11  
## PurYearMo SalesCnt PrimeRate Auto48MoRate CPI\_ALL RealDispIncoBillions  
## <chr> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 2007-01 12324 8.25 7.92 203.437 10759.9  
## 2 2007-02 12286 8.25 7.74 204.226 10788.0  
## 3 2007-03 16974 8.25 7.74 205.288 10820.6  
## 4 2007-04 15885 8.25 7.74 205.904 10818.3  
## 5 2007-05 17384 8.25 7.92 206.755 10813.0  
## 6 2007-06 15771 8.25 7.92 207.234 10800.7  
## 7 2007-07 16170 8.25 7.92 207.603 10825.1  
## 8 2007-08 17878 8.25 7.82 207.667 10831.6  
## 9 2007-09 13009 8.25 7.82 208.547 10859.2  
## 10 2007-10 15804 7.75 7.82 209.190 10839.4  
## # ... with 74 more rows, and 5 more variables: PersConsBillions <dbl>,  
## # PerSavRate <dbl>, UnempThousands <int>, USAutoSalesThousUnadj <dbl>,  
## # USAutoSalesMillionsAdj <dbl>

SalesTest <- subset(SalesEconForModelByYearMo, PurYearMo>="2014-01")  
SalesTest

## # A tibble: 36 × 11  
## PurYearMo SalesCnt PrimeRate Auto48MoRate CPI\_ALL RealDispIncoBillions  
## <chr> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 2014-01 14009 3.25 4.42 235.436 11658.2  
## 2 2014-02 14263 3.25 4.23 235.621 11723.9  
## 3 2014-03 20729 3.25 4.23 235.897 11793.9  
## 4 2014-04 20903 3.25 4.23 236.495 11826.5  
## 5 2014-05 22567 3.25 4.50 236.803 11875.4  
## 6 2014-06 20616 3.25 4.50 237.016 11932.1  
## 7 2014-07 19114 3.25 4.50 237.259 11955.2  
## 8 2014-08 21028 3.25 4.16 237.163 12009.6  
## 9 2014-09 18594 3.25 4.16 237.510 12026.7  
## 10 2014-10 20603 3.25 4.16 237.651 12080.1  
## # ... with 26 more rows, and 5 more variables: PersConsBillions <dbl>,  
## # PerSavRate <dbl>, UnempThousands <int>, USAutoSalesThousUnadj <dbl>,  
## # USAutoSalesMillionsAdj <dbl>

Now let's look at all our variables and see if we can spot in multi-collinearity among them.

cor(select(SalesTrain, -PurYearMo))

## SalesCnt PrimeRate Auto48MoRate CPI\_ALL  
## SalesCnt 1.00000000 0.03770861 -0.4199656 0.4604143  
## PrimeRate 0.03770861 1.00000000 0.7289035 -0.7005669  
## Auto48MoRate -0.41996560 0.72890352 1.0000000 -0.9741863  
## CPI\_ALL 0.46041432 -0.70056688 -0.9741863 1.0000000  
## RealDispIncoBillions 0.52542741 -0.57053310 -0.9005976 0.8987171  
## PersConsBillions 0.54403512 -0.55606891 -0.9570066 0.9748796  
## PerSavRate 0.05308314 -0.74389154 -0.5495429 0.5117481  
## UnempThousands -0.26528192 -0.87529141 -0.4527487 0.4010180  
## USAutoSalesThousUnadj 0.66167023 0.33001419 -0.1551218 0.1643954  
## USAutoSalesMillionsAdj 0.59470553 0.43591588 -0.1853008 0.2218669  
## RealDispIncoBillions PersConsBillions PerSavRate  
## SalesCnt 0.5254274 0.5440351 0.05308314  
## PrimeRate -0.5705331 -0.5560689 -0.74389154  
## Auto48MoRate -0.9005976 -0.9570066 -0.54954293  
## CPI\_ALL 0.8987171 0.9748796 0.51174813  
## RealDispIncoBillions 1.0000000 0.9147430 0.67264987  
## PersConsBillions 0.9147430 1.0000000 0.40145827  
## PerSavRate 0.6726499 0.4014583 1.00000000  
## UnempThousands 0.2731259 0.2441747 0.63611215  
## USAutoSalesThousUnadj 0.2550706 0.2811284 -0.19443210  
## USAutoSalesMillionsAdj 0.3009778 0.3739440 -0.31935460  
## UnempThousands USAutoSalesThousUnadj  
## SalesCnt -0.2652819 0.6616702  
## PrimeRate -0.8752914 0.3300142  
## Auto48MoRate -0.4527487 -0.1551218  
## CPI\_ALL 0.4010180 0.1643954  
## RealDispIncoBillions 0.2731259 0.2550706  
## PersConsBillions 0.2441747 0.2811284  
## PerSavRate 0.6361122 -0.1944321  
## UnempThousands 1.0000000 -0.4944644  
## USAutoSalesThousUnadj -0.4944644 1.0000000  
## USAutoSalesMillionsAdj -0.6394982 0.8029357  
## USAutoSalesMillionsAdj  
## SalesCnt 0.5947055  
## PrimeRate 0.4359159  
## Auto48MoRate -0.1853008  
## CPI\_ALL 0.2218669  
## RealDispIncoBillions 0.3009778  
## PersConsBillions 0.3739440  
## PerSavRate -0.3193546  
## UnempThousands -0.6394982  
## USAutoSalesThousUnadj 0.8029357  
## USAutoSalesMillionsAdj 1.0000000

One may have thought that the Prime Rate and the 48-Month New Automotive Loan Rate would have been highly correlated. While they are pretty high, 0.72, the Prime Rate appears to correlate higher with the Unemployment Level, 0.87, and the Auto Loan Rate has extremely high correlations with both the Consumer Price Index (CPI\_ALL), -0.97, and Personal Consumption, 0.96. And those two have a correlation of 0.97. Since we're dealing with New Sales and we learned before that 43% of the BMW buyers financed their purchase, this is a variable we want to keep in. We may be able to remove the other too. As expected, Real Disposable Income and Personal Consumption are highly correlated at 0.91. Interestingly enough, our SalesCnt doesn't have a high correlation with any of the variables.

Now that we've taken a quick look at correlation, we can run a linear regression model using all our variables. The SalesCnt will be our Dependent variable and the rest our Independent variables.

lin1 <- lm(SalesCnt ~ PrimeRate + Auto48MoRate + CPI\_ALL + RealDispIncoBillions + PersConsBillions + PerSavRate + UnempThousands + USAutoSalesThousUnadj, data=SalesTrain)  
summary(lin1)

##   
## Call:  
## lm(formula = SalesCnt ~ PrimeRate + Auto48MoRate + CPI\_ALL +   
## RealDispIncoBillions + PersConsBillions + PerSavRate + UnempThousands +   
## USAutoSalesThousUnadj, data = SalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4965.2 -1298.3 -19.4 1014.4 6956.6   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -208854.0459 126226.1129 -1.655 0.102   
## PrimeRate 24.9353 653.9288 0.038 0.970   
## Auto48MoRate 704.2680 1608.7188 0.438 0.663   
## CPI\_ALL 617.6526 497.4873 1.242 0.218   
## RealDispIncoBillions 19.7808 17.6698 1.119 0.267   
## PersConsBillions -12.9547 15.3797 -0.842 0.402   
## PerSavRate -2063.9684 2162.9012 -0.954 0.343   
## UnempThousands 0.0865 0.2474 0.350 0.728   
## USAutoSalesThousUnadj 12.8902 3.0332 4.250 0.0000608 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2120 on 75 degrees of freedom  
## Multiple R-squared: 0.6173, Adjusted R-squared: 0.5765   
## F-statistic: 15.12 on 8 and 75 DF, p-value: 5.764e-13

Wow! So it looks like the only significant variable is the overall U.S. automotive sales. Perhaps we have too many other variables, so before we rely solely on this variable, let's see what happens if we start removing some of them. We'll start by removing some of the variables that were highly correlated. We discussed how the Auto Loan Rate has extremely high correlations with both the Consumer Price Index (CPI\_ALL), -0.97, and Personal Consumption, 0.96. So let's remove both the CPI and the Personal Consumption variables. Disposable Income was also quite high with all three of those, around 90%, but for now, we'll keep it in.

lin2 <- lm(SalesCnt ~ PrimeRate + Auto48MoRate + RealDispIncoBillions + PerSavRate + UnempThousands + USAutoSalesThousUnadj, data=SalesTrain)  
summary(lin2)

##   
## Call:  
## lm(formula = SalesCnt ~ PrimeRate + Auto48MoRate + RealDispIncoBillions +   
## PerSavRate + UnempThousands + USAutoSalesThousUnadj, data = SalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4827.2 -1319.5 48.2 977.8 7120.3   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -93437.15970 45593.68706 -2.049 0.0438 \*   
## PrimeRate -318.56474 538.52566 -0.592 0.5559   
## Auto48MoRate 1050.44972 1105.10538 0.951 0.3448   
## RealDispIncoBillions 8.98763 3.70868 2.423 0.0177 \*   
## PerSavRate -759.12122 472.00673 -1.608 0.1119   
## UnempThousands -0.04747 0.22648 -0.210 0.8345   
## USAutoSalesThousUnadj 13.56330 2.89536 4.685 0.0000119 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2130 on 77 degrees of freedom  
## Multiple R-squared: 0.6033, Adjusted R-squared: 0.5724   
## F-statistic: 19.52 on 6 and 77 DF, p-value: 1.048e-13

So it's interesting to note here that our Adjusted R-squared value dropped ever so slightly from 0.5765 to 0.5724, but we have picked up two slightly significant variables - the Intercept and the Disposable Income. But I'm still curious as to what would happen if we remove the Disposable Income since it does have a higher correlation with the Auto Loan Rate.

lin3 <- lm(SalesCnt ~ PrimeRate + Auto48MoRate + PerSavRate + UnempThousands + USAutoSalesThousUnadj, data=SalesTrain)  
summary(lin3)

##   
## Call:  
## lm(formula = SalesCnt ~ PrimeRate + Auto48MoRate + PerSavRate +   
## UnempThousands + USAutoSalesThousUnadj, data = SalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4710.6 -1480.8 -194.8 1113.6 7697.3   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 16487.2015 4757.3026 3.466 0.000863 \*\*\*  
## PrimeRate 286.3567 491.8605 0.582 0.562117   
## Auto48MoRate -1422.7578 436.9918 -3.256 0.001674 \*\*   
## PerSavRate 242.0764 235.2947 1.029 0.306743   
## UnempThousands -0.2295 0.2202 -1.042 0.300510   
## USAutoSalesThousUnadj 13.6380 2.9843 4.570 0.000018 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2196 on 78 degrees of freedom  
## Multiple R-squared: 0.573, Adjusted R-squared: 0.5456   
## F-statistic: 20.94 on 5 and 78 DF, p-value: 3.302e-13

So with Disposable Income removed, we saw no changes in our values, which means that it was a good choice for removal in this particular case, but perhaps the Auto Loan Rate is actually less significant than the Disposable Income. Let's see what happens when we keep our USAutoSales and add just the AutoLoan or just the DispIncome.

lin4 <- lm(SalesCnt ~ RealDispIncoBillions + USAutoSalesThousUnadj, data=SalesTrain)  
summary(lin4)

##   
## Call:  
## lm(formula = SalesCnt ~ RealDispIncoBillions + USAutoSalesThousUnadj,   
## data = SalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4945.5 -1185.7 -197.8 1138.0 7027.7   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -37441.0378 8290.4981 -4.516 2.12e-05 \*\*\*  
## RealDispIncoBillions 3.8681 0.7606 5.085 2.32e-06 \*\*\*  
## USAutoSalesThousUnadj 16.6997 2.2197 7.524 6.47e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2153 on 81 degrees of freedom  
## Multiple R-squared: 0.5739, Adjusted R-squared: 0.5633   
## F-statistic: 54.54 on 2 and 81 DF, p-value: 9.926e-16

lin5 <- lm(SalesCnt ~ Auto48MoRate + USAutoSalesThousUnadj, data=SalesTrain)  
summary(lin5)

##   
## Call:  
## lm(formula = SalesCnt ~ Auto48MoRate + USAutoSalesThousUnadj,   
## data = SalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4636 -1427 -372 1107 8934   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 10942.777 2002.792 5.464 5.01e-07 \*\*\*  
## Auto48MoRate -950.576 222.773 -4.267 5.34e-05 \*\*\*  
## USAutoSalesThousUnadj 18.086 2.255 8.021 6.82e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2234 on 81 degrees of freedom  
## Multiple R-squared: 0.541, Adjusted R-squared: 0.5297   
## F-statistic: 47.73 on 2 and 81 DF, p-value: 2.013e-14

Alright, we can see that both show as very significant, but we have a higher Adjusted R-squared value when we keep the Disposable Income. And this makes intuitive sense since we'd expect sales to increase when people have more disposable income. And while auto loan rates could still play a part in some people's decision to buy or not, it won't in many others. So what happens if we run our model with just these three?

lin6 <- lm(SalesCnt ~ Auto48MoRate + RealDispIncoBillions + USAutoSalesThousUnadj, data=SalesTrain)  
summary(lin6)

##   
## Call:  
## lm(formula = SalesCnt ~ Auto48MoRate + RealDispIncoBillions +   
## USAutoSalesThousUnadj, data = SalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4952.8 -1229.4 -176.9 1118.9 7034.0   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -44817.711 22284.095 -2.011 0.0477 \*   
## Auto48MoRate 177.906 498.419 0.357 0.7221   
## RealDispIncoBillions 4.436 1.766 2.512 0.0140 \*   
## USAutoSalesThousUnadj 16.556 2.268 7.301 1.87e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2164 on 80 degrees of freedom  
## Multiple R-squared: 0.5745, Adjusted R-squared: 0.5586   
## F-statistic: 36.01 on 3 and 80 DF, p-value: 7.86e-15

The Auto Rate becomes insignificant despite the fact that our Adjusted R-squared increased by nearly 0.03 points.

So, it looks like we have a good base model using just the Auto Sales and the Disposable Income, but let's see if we can improve upon it by adding in some of the other variables one at a time. I'll leave out the two values that were very highly correlated with the auto loan sales though since we'd expect to see a similar reaction as we did when we added the Auto Loan Rate.

lin7 <- lm(SalesCnt ~ PrimeRate + RealDispIncoBillions + USAutoSalesThousUnadj, data=SalesTrain)  
summary(lin7)

##   
## Call:  
## lm(formula = SalesCnt ~ PrimeRate + RealDispIncoBillions + USAutoSalesThousUnadj,   
## data = SalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4945.7 -1164.4 -123.8 1068.6 7038.9   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -50060.417 11943.579 -4.191 0.00007098 \*\*\*  
## PrimeRate 295.769 202.787 1.459 0.149   
## RealDispIncoBillions 5.002 1.084 4.614 0.00001481 \*\*\*  
## USAutoSalesThousUnadj 14.296 2.752 5.194 0.00000153 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2138 on 80 degrees of freedom  
## Multiple R-squared: 0.5849, Adjusted R-squared: 0.5693   
## F-statistic: 37.57 on 3 and 80 DF, p-value: 2.957e-15

lin8 <- lm(SalesCnt ~ PerSavRate + RealDispIncoBillions + USAutoSalesThousUnadj, data=SalesTrain)  
summary(lin8)

##   
## Call:  
## lm(formula = SalesCnt ~ PerSavRate + RealDispIncoBillions + USAutoSalesThousUnadj,   
## data = SalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4855.0 -1332.3 -55.1 860.7 7553.6   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -53625.570 11264.748 -4.760 0.000008457 \*\*\*  
## PerSavRate -490.755 236.547 -2.075 0.0412 \*   
## RealDispIncoBillions 5.685 1.150 4.943 0.000004161 \*\*\*  
## USAutoSalesThousUnadj 14.012 2.532 5.534 0.000000384 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2110 on 80 degrees of freedom  
## Multiple R-squared: 0.5956, Adjusted R-squared: 0.5805   
## F-statistic: 39.28 on 3 and 80 DF, p-value: 1.048e-15

lin9 <- lm(SalesCnt ~ UnempThousands + RealDispIncoBillions + USAutoSalesThousUnadj, data=SalesTrain)  
summary(lin9)

##   
## Call:  
## lm(formula = SalesCnt ~ UnempThousands + RealDispIncoBillions +   
## USAutoSalesThousUnadj, data = SalesTrain)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4865.5 -1269.9 -51.2 1024.1 7081.3   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -41260.5899 8525.7687 -4.840 0.000006227 \*\*\*  
## UnempThousands -0.1818 0.1104 -1.647 0.104   
## RealDispIncoBillions 4.5374 0.8554 5.304 0.000000982 \*\*\*  
## USAutoSalesThousUnadj 13.9403 2.7627 5.046 0.000002768 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2130 on 80 degrees of freedom  
## Multiple R-squared: 0.5878, Adjusted R-squared: 0.5724   
## F-statistic: 38.03 on 3 and 80 DF, p-value: 2.232e-15

Alright, so both Unemployment and Prime Rate increase our Adjusted R-squared, but neither proves to be significant. The Personal Savings Rate, on the other hand, is showing to be somewhat significant and increases our Adjusted R-squared value to 0.58 - the highest of all previous models, so we'll keep that one in.

SSE <- sum(lin8$residuals^2)  
SSE

## [1] 356220374

RMSE <- sqrt(SSE/nrow(SalesTrain))  
RMSE

## [1] 2059.301

AvgSalesMo <- mean(SalesTrain$SalesCnt)  
AvgSalesMo

## [1] 15060.96

RMSE/AvgSalesMo

## [1] 0.136731

So on average, we make an error of 2059 sales in a month, a 13% swing in either direction. This doesn't seem to be the best predictive model, but it is the best one we have based on the data we have available to us. With that in mind, let's use our model to see if we can use it to predict sales. For this, we'll use our testing data set.

SalesPredict <- predict(lin8, newdata = SalesTest)   
SalesPredict

## 1 2 3 4 5 6 7 8   
## 16780.48 18310.03 21164.34 20115.16 21952.69 20969.07 20849.54 22448.06   
## 9 10 11 12 13 14 15 16   
## 19877.29 20445.76 20735.12 21954.83 20038.31 20702.98 23102.19 22323.58   
## 17 18 19 20 21 22 23 24   
## 23994.86 22908.86 22898.70 23380.60 22512.41 22615.97 21777.64 23517.19   
## 25 26 27 28 29 30 31 32   
## 21281.44 22540.42 24160.59 23743.17 24094.64 24069.05 24110.49 23910.46   
## 33 34 35 36   
## 23856.97 23277.94 23377.31 24845.43

So this output list shows the sales predictions for each of the months in our test data set (Jan 2014 through Dec 2016), but how well did our model do? We can calculate the out-of-sample R-squared value. When we have that, we can compare it to the RMSE we calculated for our model above. Ideally the error would not have changed significantly.

pSSE <- sum((SalesPredict - SalesTest$SalesCnt)^2) # predictions - test actuals  
pSST <- sum((mean(SalesTrain$SalesCnt)-SalesTest$SalesCnt)^2) #avg train actuals - test actuals  
pR2 <- 1 - pSSE/pSST  
pRMSE <- sqrt(pSSE/nrow(SalesTest))  
pRMSE

## [1] 8496.76

RMSE

## [1] 2059.301

Based on our test data set from the last three years, our RMSE has quadrupled & we now have a 56% spread of error! This means that our linear regression model from our testing set is not a good fit to predict the future sales of BMW's. Unfortunately, it doesn't look like any of the data we have is a good way to predict future BMW sales. We may be able to build a decent model using data from all years from three of the regions and using the fourth region as our testing data. However, our goal was to try to predict future sales, and that approach would not do that for us.

## Pitfalls of the data set

* Not all BMW stores have provided access to their Sales data so the analysis is limited by dealership participation.
* The PurchasePrice could vary from dealer to dealer in how they're entering the values. If it includes the sales price negotiated for the vehicle or if trade-in's are removed from the total. This could also vary based on the data management software the dealership uses. This makes the values calculated less reliable as a true sales price, but by eliminating most of the outliers, we should still get a decent representation of pricing.
* We do not have enough demographic data on the customers - gender, birth date, etc - to use this as part of our analysis or prediction model.

## Additional Areas for Exploration

* Break down trends by BMW Market
  + We saw that many of the Regional trends mirrored the National, but what about within each Market? We could dig into this further to see if the same holds true or if there's more variablilty from Market to Market within each Region.
* Vehicle Categories by Gender &/or Age (using birth date)
  + Even though we have very limited Gender data, and likely birth date date, we could still disect the data we do have and see if there are any trends or preferences within each gender or in age groups. Which vehicle categories appeal to the younger market and which to the older? Are females buying more of the SUV's than males?
* Repeat vs First-time Buyers
  + Here we can make good use of the trade VIN data. We can look at what make they traded to see if it was an older BMW, indicating brand loyalty, or another make, indicated a converted & possibly first-time owner. We could also look beyond the trade VIN data, by checking out the number of distinct VINs owned by each CustomerID. This would be able to get us a minimum number of VINs, but not necessarily the whole picture. Due to possible changes in the system that a dealer uses to manage their database, the CustomerIDs can change over time. This means that a single customer may actually have more than one CustomerID and be seen as 2 or more unique customers. More elaborate customer matching, based on name & address, could be done to try to remedy some of these issue, but there is still room for error based on how a name or address is typed into the system. For example, "Jim" vs "James" or "1234 Main Street" vs "1234 Main St.". Using CustomerID will also not allow us to see if the same customer purchases at more than one BMW dealership since these values are unique to each dealership. So while we can try to view this within the scope of each dealership, we'd have a harder time getting true values since we wouldn't be able to accurately apply this to all BMW dealerships as a whole.
* Length of Vehicle Ownership
  + If we have a subsequent sales transaction on the vehicle or the VIN being listed by the same customer as being traded in, then we'd have a firm length of ownership. If there is no additional sales record though, we'd have a difficult time ascertaining this data. We could try to turn to the service history, which our company also collects, to see when they last serviced it. But many dealers use different CustomerID's for sales than they do for service so matching these would be difficult. And as mentioned above, even the CustomerID's within the service history at a single dealership can be dicey. We'd also have to take into consideration that even if we can match service data with a sales records, we still wouldn't know their true ownership time because we wouldn't know if they still own the vehicle and perhaps are servicing it elsewhere, or have since sold it, either privately or traded it in for competing make. Because of these unknowns, it would be difficult to get accurate ownership time frames unless we had multiple sales records on hand.

## Summary

In conclusion, while we've been able to explore our data and visualize trends in our sales data, we have been unsuccessfully been able to predict sales using any of the economic values we've obtained. Ideally, we would have had access to personal information about each of the buyers, such as income, occupation, home value, ethnicity, gender, and birth date, that may have been able to lead us to a better model. We may have found that what affected the nation as a whole, may not have affected the BMW customers as much. Due to the higher price points of the BMW lineup and the fact that a large number of buyers (27%) paid in cash, we can conclude that, in general, BMW buyers are probably upper class with a higher income than the national average.

We can also see that the BMW sales are trending up as a whole, although fairly stagnant at the moment. This could suggest that sales will start to pick up again in 2017 or 2018. The New sales have been on a slight decline in recent years, but we also saw a similar drop in the first few years of our analysis, after which their sales rose for six straight years. Since we don't have data prior to 2007 in our graphs, we don't see how long that downward trend continued before the rise, but it does appear that their New sales are cyclical. We should expect to see them rise again in the next few years. Used sales trends were less volatile than the New. They only had one year where the counts declined, and even then, only sightly, so we can predict that Used sales will continue to rise. We also learned that the 3-series models have been, by far, BMW's biggest seller. To boost volume of New sales, they may want to put their marketing focus here. And while in the sedan/coupes category, the moderately priced version won out, in the SUV category, customers are opting for the more expensive models. BMW could explore upselling their existing 3-series or X3 customers to the higher priced 5-series and X5's. Or they could look at increased marketing of their SUV lineup to owners of other SUV brands.

We still have many more areas for exploration with our data, which could lead to more insights into the trends happening in the BMW market. We could also continue to explore other ways to try to predict future values, perhaps finding one that will prove more useful.