Day of Week Prediction Power of Expenditures Analysis

Nicholas Hartman Ponce

nicholas.paul.hartman@gmail.com

Github Repo

```
# This R markdown file contains the necessary markdown, commentary, and code to produce a knitr'd prese

# This file assumes that all files in the github repo are available in your R working directory.

# This analysis is intended to provide a demonstrative sample of my ability to conduct and narrate anal
```

Context

My wife and I recently discussed differences between weekday and weekend spending. I had already plotted average spend by day-of-week, but I began to ask myself whether the expenditure data held any predictive value.

As we are repatriating to the US in August, and I am currently seeking work in analysis in Providence, RI, and demonstrable analytical ability is helpful to that pursuit, a small investigation seemed appropriate - and so here we are.

Premise

My wife and I, both early/mid-career, white-collar professionals, have generally adhered to the Monday - Friday workweek conventional in the west. As most people struggle to spend money during employment hours (excepting generally small-dollar lunches, coffee, etc.), my hunch is our daily expenditures are lower during the week than at the weekend.

The question here then is whether I can predict either:

- Weekday from Weekend, or
- Which specific day of the week

using sum/count of our various expenditures for a given day.

Plan

To get to either A) a working predictive model or B) a determination that that would require more attributes, I'll:

- consider what's available
- decide what to export, and whether to aggregate before or after export
- export to .tld and then repackage/reformat the data in R
- explore/evaluate and plot the data
- index & split the data

- train a couple of models
- test, see if anything works
- summarize findings

Data, Broadly

I have recorded each of our US Dollar financial transactions from ~April 2014. The columns/attributes have multiplied over time, but the core of Date, Local Time to the quarter hour, Vendor, and Dr/Cr amounts have not changed.

While I have recorded all of our USD transactions, for this analysis I will be looking only at expenditures, excluding transfers between accounts, credit card payments, etc.

Also filtered out from the data will be expenditures that, for one reason or another, are clearly not relevant to DoW analysis (i.e. rent payments, loan payments, etc. which fall on a fixed calendar date, among other items).

Data, Basic Characteristics

The source data are ~7,900 rows (1 Apr 2014 through 12 Jul 2018) across 19 columns of USD transactions.

The transactions are only those in USD because we have spent the majority of our time living and working in the US using the USD, our non-USD expenditures comprise a small minority (both by count and sum) of our total expenditures, and, while I have plans for other uses of these data, tracking historical Point-in-Time exchange rates is not one of them.

A data dictionary for the source is provided as "SourceDataDictionary.txt". It uses column names as formatted in Excel, but the sequence of columns is unchanged from below. Here are the columns as dumped raw to R:

```r
## Read in precanned Column Names

AllColNames <- read.delim("AllColNames.txt")

AllColNames
```

```
##  [1] Date              DoW.M.1           Local.Time       
##  [4] Country           State             City             
##  [7] Zip.Code          Method.of.Payment Vendor           
## [10] Purchaser         Good.or.Service   Note.Comments    
## [13] Exp.d.            Reimb.            Recur.           
## [16] Traveling.        In.Pers.          Debit            
## [19] Credit           
## <0 rows> (or 0-length row.names)
```

For this analysis I've decided to limit use to four: Date, DoW, Debit, and Credit. Date will be used to aggregate, rather than for prediction (pretty easy to predict DoW if you know what the date is).

Data, Quality

While I'm certain the data are not 100% perfect, I am similarly certain that the error rate is negligibly small.

Informed by professional habits, I've been disciplined about data entry habits/controls (various account rec's, etc., personal month-end closes, etc.) such that errors, where they exist, are not material.

Known errors are generally something like, "I spent $15 in cash, but how much was at the Food Truck and how much was the bottle of water at the kiosk? Can't remember, put it all in as Food Truck."

There are a number of blank values across a handful of columns I'm working to backfill in separate efforts, however none of these columns are used in this analysis.

Initial Excel Decisions

As mentioned above, not all transactions in the data will be relevant and/or useful for this analysis, and the goal is to work from expenditure Sum-by-Day, not individual transactions. So there's some filtering and some aggregating to do.

It's not strictly necessary to do either in one or the other of Excel or R. Seeing as the pre-export data already live in Excel with easily GUI-filtered columns, it'd probably be easiest to just do that pre-export.

As far as aggregation, I could either pivot my way there and then export, or use a simple script. In R, the aggregation code would look something like:

```r
## Read in disaggregated Transactions
Demo <- read.delim("DataExp1.txt")

## Change class on Date and DoW
Demo[,1] <- as.factor(Demo[,1])
Demo[,2] <- as.factor(Demo[,2])

## Replace nulls with 0
Demo[is.na(Demo)] <- 0

## First net Debits and Credits
Demo$Net <- Demo$Debit + Demo$Credit

## Aggregate transaction lines to Sum-by-Date
AggDemo <- aggregate(x = Demo$Net, by = list(DoW = Demo$DoW.M.1, DatesList = Demo$Date), FUN = "sum")

## Clean up names
names(AggDemo) <- c("DoW","Date","NetExp")

## Review
head(AggDemo)
```

```
##   DoW  Date   NetExp
## 1   2 41730 -1952.21
## 2   3 41731  -465.12
## 3   4 41732   -39.99
## 4   5 41733  1587.63
## 5   6 41734   -75.43
## 6   7 41735   -46.33
```

That's not terrible, but this kind of work is so well developed in Excel, no point in reinventing the wheel. I'll export the filtered, pivoted/aggregated data.

Actually, just before exporting, I can even quickly and easily handle the Workday/Weekend day column I'll need with with a simple IF formula column:

=IF($DoW < 6$, TRUE, FALSE)

Where DoW in Excel is the relevant cell reference.

Great, so now we've got what we want out of Excel (DataExp2.txt). Let's see what we can find out.

Preliminary Analysis

We'll begin by bringing every thing, cleaning/reformatting it, and tidying up the environment as we go:

```
# Load Data
RawImport <- read.delim("DataExp2.txt", header = TRUE)

# Clean Data
CleanImport <- RawImport

CleanImport[,1] <- as.factor(CleanImport[,1])
CleanImport[,2] <- as.factor(CleanImport[,2])

CleanImport[,5] <- CleanImport[,5]*-1

colnames(CleanImport)[1] <- "Date"
colnames(CleanImport)[2] <- "DoW"
colnames(CleanImport)[5] <- "Exp"

drops <- c("Date")
CleanImport <- CleanImport[ , !(names(CleanImport) %in% drops)]

remove(RawImport,drops)

head(CleanImport)
```

```
##   DoW Weekday Count   Exp
## 1   2    TRUE     2 25.66
## 2   3    TRUE     1  5.17
## 3   4    TRUE     4 39.99
## 4   5    TRUE     3 69.56
## 5   6   FALSE     6 75.43
## 6   7   FALSE     5 46.33
```

Great! Now we've got one, clean R data frame:

```
dim(CleanImport)
```

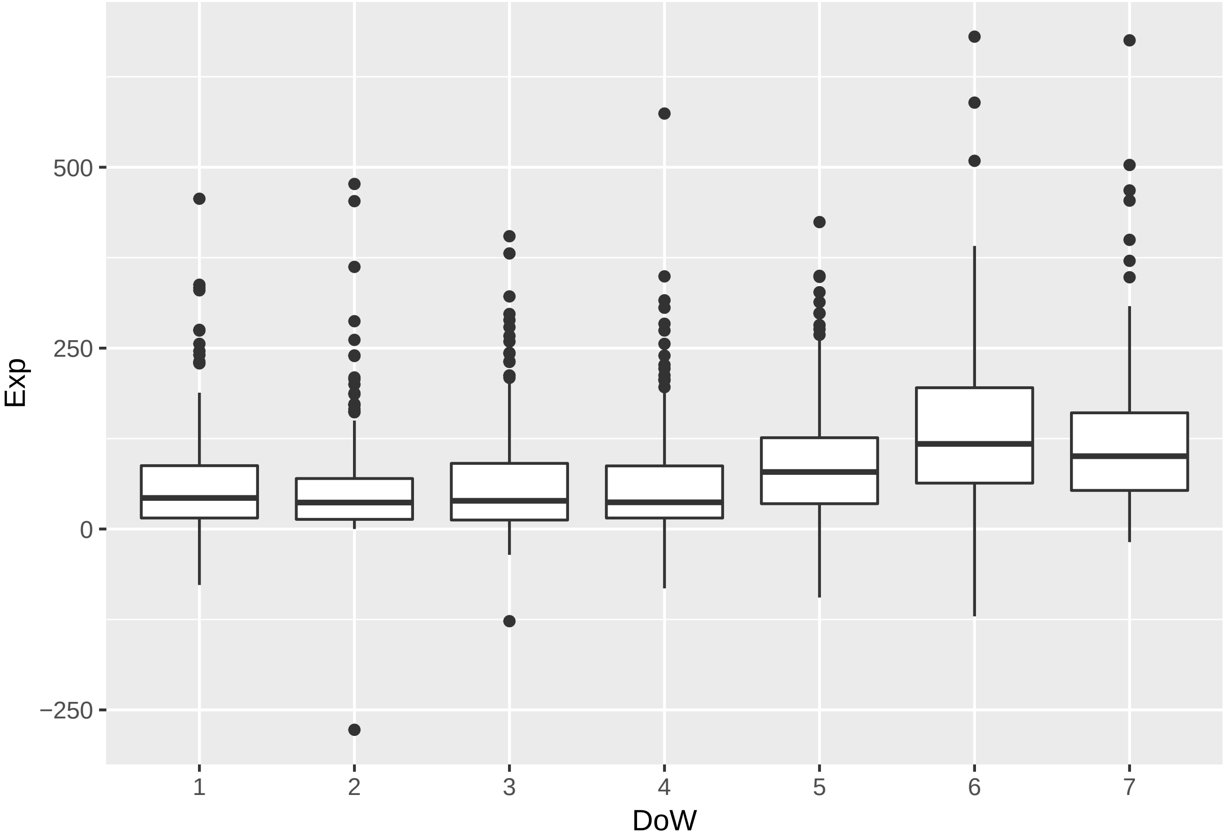
```
## [1] 1561    4
```

```
summary(CleanImport)
```

```
##  DoW      Weekday            Count             Exp
##  1:222   Mode :logical   Min.   : 1.000   Min.   :-277.47
##  2:224   FALSE:446       1st Qu.: 2.000   1st Qu.:  23.01
##  3:223   TRUE :1115      Median : 3.000   Median :  61.79
##  4:223                   Mean   : 3.675   Mean   :  84.66
##  5:223                    3rd Qu.: 5.000   3rd Qu.: 121.30
##  6:223                    Max.   :23.000   Max.   : 680.45
##  7:223
```

4 columns of 1,561 rows to work with. Let's make a quick visual inspection of the data, starting with Net Expenditure by Day of the Week:

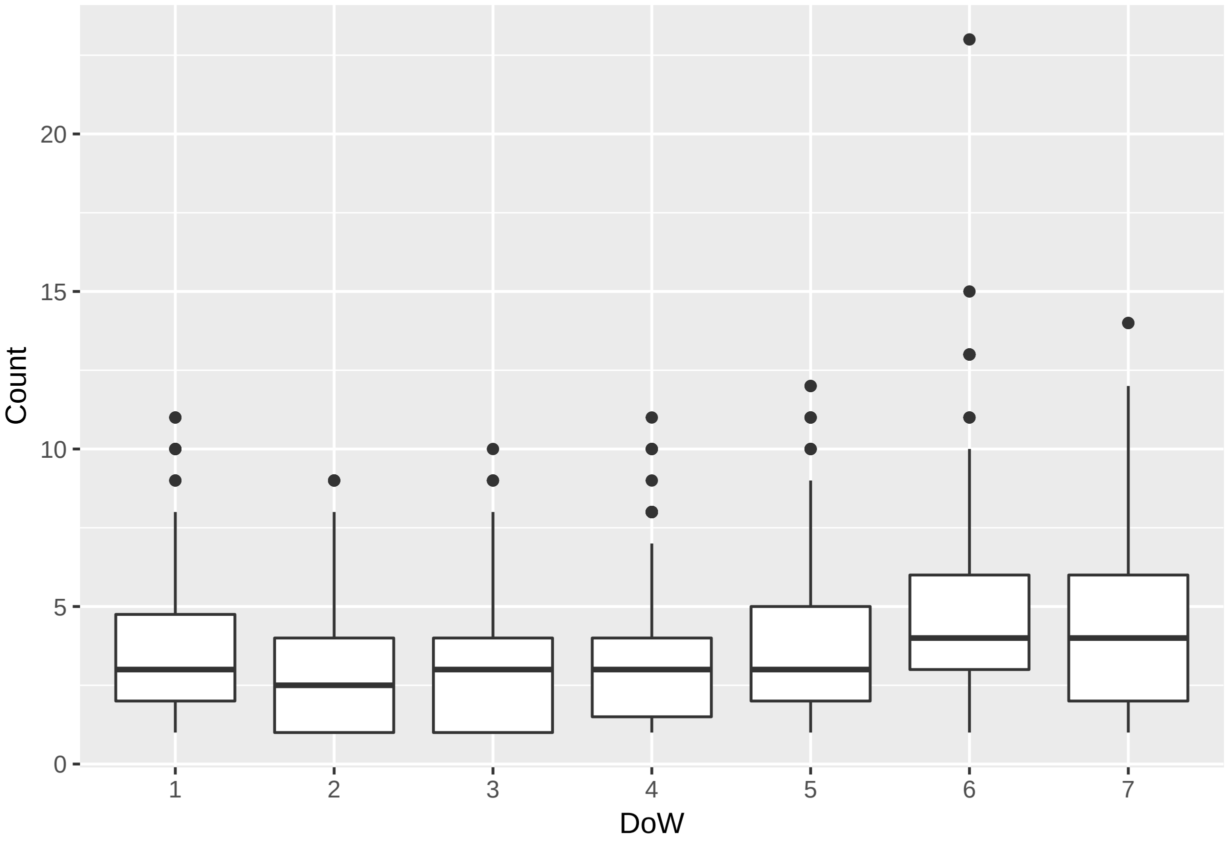
```
ExpByDoWBox <- ggplot(CleanImport, aes(x = DoW, y = Exp)) + geom_boxplot()
ExpByDoWBox
```



So definitely skewed, lots of outliers, but not really surprising, since it's intuitive that there would be many fewer sub-zero expenditure days than high-spend days.

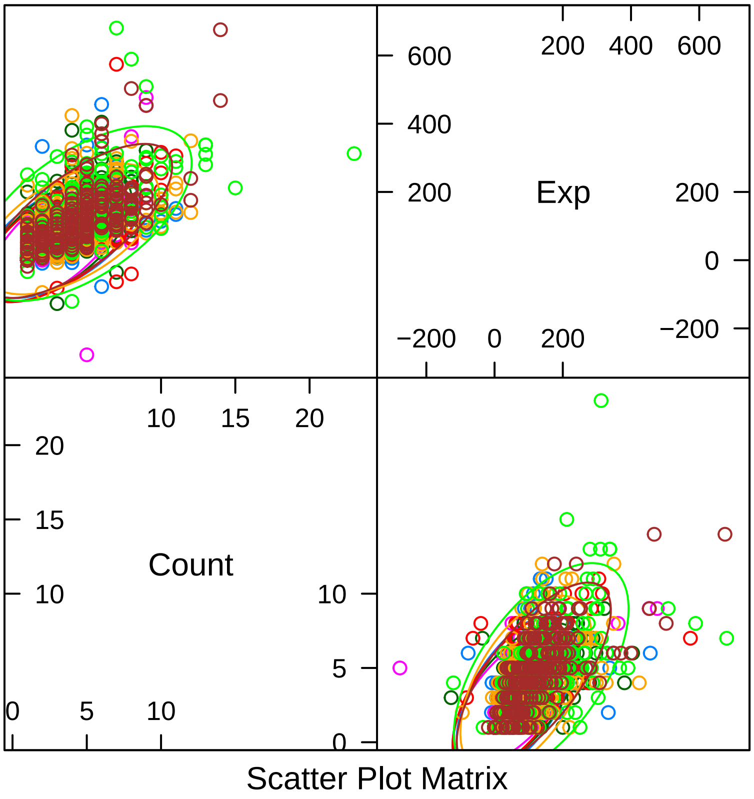
Some cause for prediction concern for the weekdays; Monday through Thursday's boxes are pretty similar. On the other hand, some cause for hope on the Weekday/Weekend front, although Friday could throw a wrench in that. Let's take a look at transaction counts:

```
CountByDoWBox <- ggplot(CleanImport, aes(x = DoW, y = Count)) + geom_boxplot()
CountByDoWBox
```



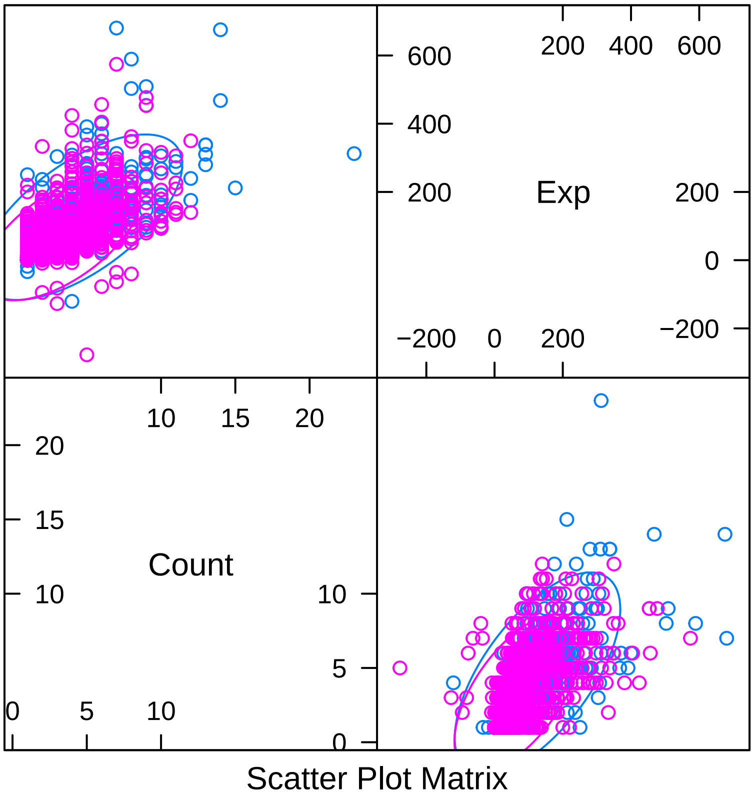
Hmm. So still a bit of a difference between the week and the weekend, but neither substantial nor really distinct. Let's try one other approach:

```
x <- as.matrix(CleanImport[,c(3:4)])
y <- factor(CleanImport$DoW)
featurePlot(x, y, plot = "ellipse")
```



Still not much separation. One last look:

```
x <- as.matrix(CleanImport[,c(3:4)])
y <- factor(CleanImport$Weekday)
featurePlot(x, y, plot = "ellipse")
```



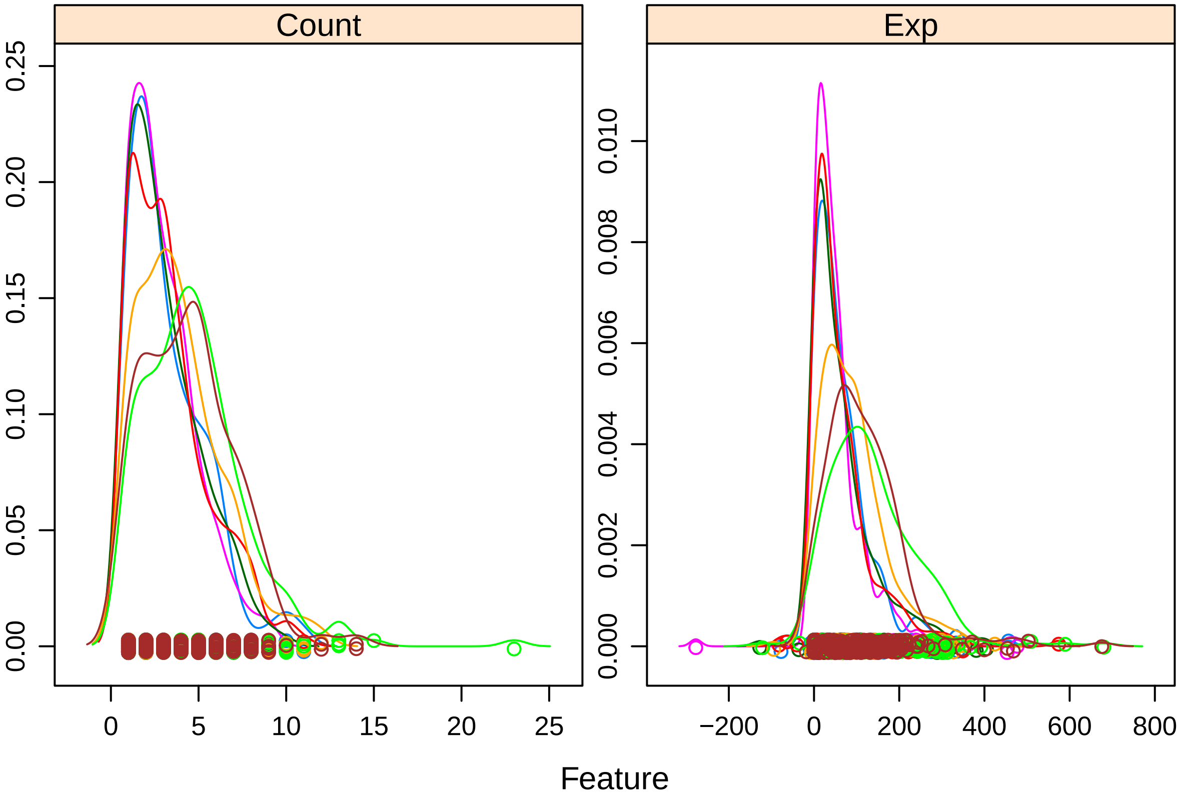
Uf, not looking good just on these variables. We can see some of the higher Weekend values (blue) scattering out to the top/right, but we've still got a lot of overlap with Weekdays.

Statistical Review

An ANOVA on the data (Count/Exp vs. DoW) seem the next logical step, but we should probably check the distributions first before settling on that. Let's look at this another way to get a sense how off of Normal the distributions are:

```
## Arrange Data for Plotting
x <- as.matrix(CleanImport[,c(3:4)])
y <- factor(CleanImport$DoW)

## Plot
featurePlot(x, y, plot = "density",
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free"))
            )
```



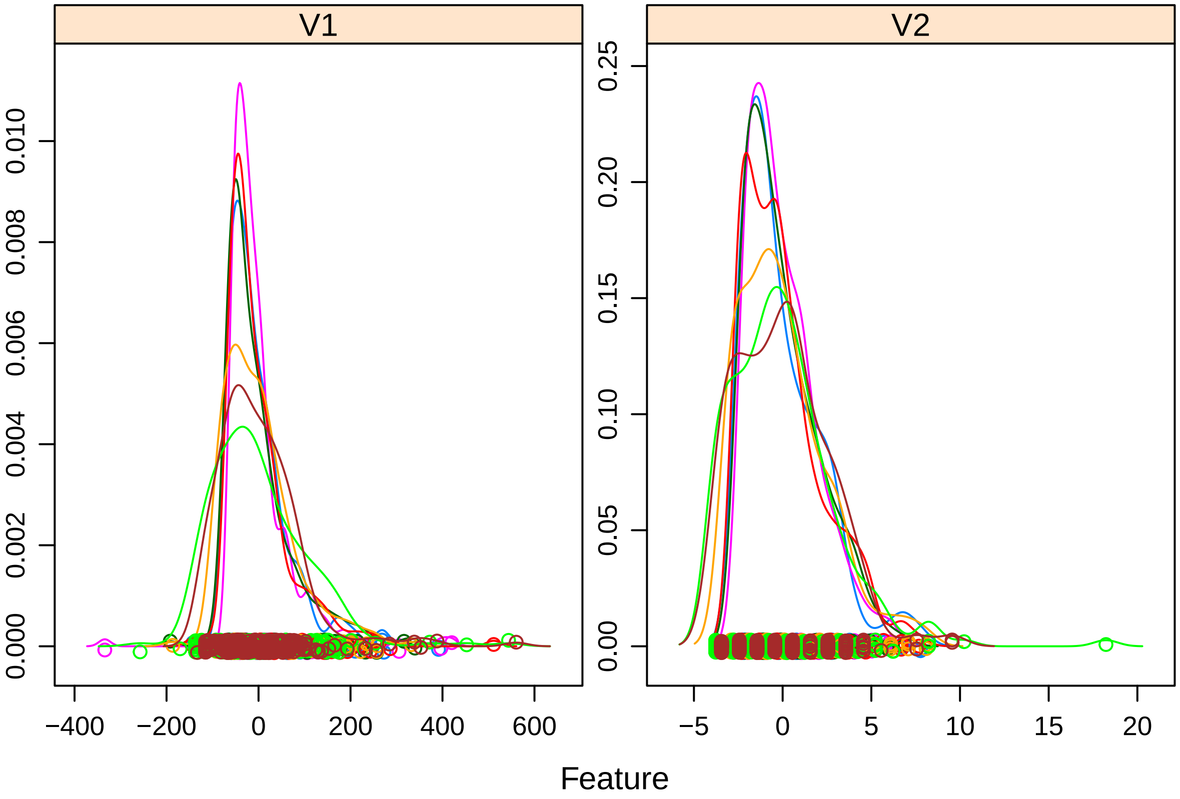
So we can see the skew, but it's not multi-modal or otherwise bizarrely shaped. Let's get a sense of the distribution of the residuals:

```
## Setup Residuals Matrix
Residuals <- matrix(nrow = 1561, ncol = 2)

## Populate Residuals
Residuals[,1] <- resid(aov(Exp ~ DoW, CleanImport))
Residuals[,2] <- resid(aov(Count ~ DoW, CleanImport))

## Tee-up DoW factors
Variable <- factor(CleanImport$DoW)

## Plot Residuals
featurePlot(Residuals, Variable, plot = "density",
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")))
```



Better, but clearly not normal. Some skew, and some inconsistent peaks.

So with non-Normal residuals distributions, one-way ANOVA evaluation of the DoW factors is out. There are a variety of other ways to test whether there's any difference in the samples given non-normality. I'll use Kruskal-Wallis testing:

```
kruskal.test(Count ~ DoW, CleanImport)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Count by DoW
## Kruskal-Wallis chi-squared = 107, df = 6, p-value < 2.2e-16
```

```
kruskal.test(Exp ~ DoW, CleanImport)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Exp by DoW
## Kruskal-Wallis chi-squared = 206.83, df = 6, p-value < 2.2e-16
```

So it looks like at least one DoW group is statistically distinct from the others for both Count and Net Expenditure. That's something!

It would be handy to know where the dividing lines were if we're going to make anything useful of this. Since we've rejected the null for both measures, let's run a Conover-Iman test to see where we see pair-wise differences:

```
conover.test(x = CleanImport$Count, g = CleanImport$DoW, list = TRUE, altp = TRUE)
```

```
##   Kruskal-Wallis rank sum test
```

```
##
## data: x and group
## Kruskal-Wallis chi-squared = 107.0008, df = 6, p-value = 0
##
##
##                            Comparison of x by group
##                                 (No adjustment)
## Col Mean-|
## Row Mean |          1          2          3          4          5          6
## ---------+------------------------------------------------------------------
##        2 |   0.960818
##          |     0.3368
##          |
##        3 |   0.459047  -0.501823
##          |     0.6463     0.6159
##          |
##        4 |  -0.432781  -1.395655  -0.892833
##          |     0.6652     0.1630     0.3721
##          |
##        5 |  -3.282485  -4.251755  -3.745744  -2.852911
##          |    0.0011*    0.0000*    0.0002*    0.0044*
##          |
##        6 |  -6.570664  -7.547315  -7.037624  -6.144790  -3.291879
##          |    0.0000*    0.0000*    0.0000*    0.0000*    0.0010*
##          |
##        7 |  -5.647977  -6.622557  -6.113899  -5.221065  -2.368154   0.923724
##          |    0.0000*    0.0000*    0.0000*    0.0000*    0.0180*     0.3558
##
##
## List of pairwise comparisons: t statistic (p-value)
## ---------------------------
## 1 - 2 :  0.960818 (0.3368)
## 1 - 3 :  0.459047 (0.6463)
## 2 - 3 : -0.501823 (0.6159)
## 1 - 4 : -0.432781 (0.6652)
## 2 - 4 : -1.395655 (0.1630)
## 3 - 4 : -0.892833 (0.3721)
## 1 - 5 : -3.282485 (0.0011)*
## 2 - 5 : -4.251755 (0.0000)*
## 3 - 5 : -3.745744 (0.0002)*
## 4 - 5 : -2.852911 (0.0044)*
## 1 - 6 : -6.570664 (0.0000)*
## 2 - 6 : -7.547315 (0.0000)*
## 3 - 6 : -7.037624 (0.0000)*
## 4 - 6 : -6.144790 (0.0000)*
## 5 - 6 : -3.291879 (0.0010)*
## 1 - 7 : -5.647977 (0.0000)*
## 2 - 7 : -6.622557 (0.0000)*
## 3 - 7 : -6.113899 (0.0000)*
## 4 - 7 : -5.221065 (0.0000)*
## 5 - 7 : -2.368154 (0.0180)*
## 6 - 7 :  0.923724 (0.3558)
##
## alpha = 0.05
## Reject Ho if p <= alpha
```

```
conover.test(x = CleanImport$Exp, g = CleanImport$DoW, list = TRUE, altp = TRUE)
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 206.8286, df = 6, p-value = 0
##
##
##                            Comparison of x by group
##                                 (No adjustment)
## Col Mean-|
## Row Mean |          1          2          3          4          5          6
## ---------+------------------------------------------------------------------
##        2 |   1.280544
##          |     0.2005
##          |
##        3 |   0.417398  -0.863653
##          |     0.6764     0.3879
##          |
##        4 |   0.544620  -0.736146   0.127364
##          |     0.5861     0.4618     0.8987
##          |
##        5 |  -4.585194  -5.877475  -5.008223  -5.135587
##          |    0.0000*    0.0000*    0.0000*    0.0000*
##          |
##        6 |  -9.567370  -10.87083  -9.996006  -10.12337  -4.987783
##          |    0.0000*    0.0000*    0.0000*    0.0000*    0.0000*
##          |
##        7 |  -7.643015  -8.942160  -8.069485  -8.196850  -3.061262   1.926521
##          |    0.0000*    0.0000*    0.0000*    0.0000*    0.0022*     0.0542
##
##
## List of pairwise comparisons: t statistic (p-value)
## ---------------------------
## 1 - 2 :  1.280544 (0.2005)
## 1 - 3 :  0.417398 (0.6764)
## 2 - 3 : -0.863653 (0.3879)
## 1 - 4 :  0.544620 (0.5861)
## 2 - 4 : -0.736146 (0.4618)
## 3 - 4 :  0.127364 (0.8987)
## 1 - 5 : -4.585194 (0.0000)*
## 2 - 5 : -5.877475 (0.0000)*
## 3 - 5 : -5.008223 (0.0000)*
## 4 - 5 : -5.135587 (0.0000)*
## 1 - 6 : -9.567370 (0.0000)*
## 2 - 6 : -10.87083 (0.0000)*
## 3 - 6 : -9.996006 (0.0000)*
## 4 - 6 : -10.12337 (0.0000)*
## 5 - 6 : -4.987783 (0.0000)*
## 1 - 7 : -7.643015 (0.0000)*
## 2 - 7 : -8.942160 (0.0000)*
## 3 - 7 : -8.069485 (0.0000)*
## 4 - 7 : -8.196850 (0.0000)*
## 5 - 7 : -3.061262 (0.0022)*
## 6 - 7 :  1.926521 (0.0542)
##
## alpha = 0.05
## Reject Ho if p <= alpha
```