Felix Analysis 2

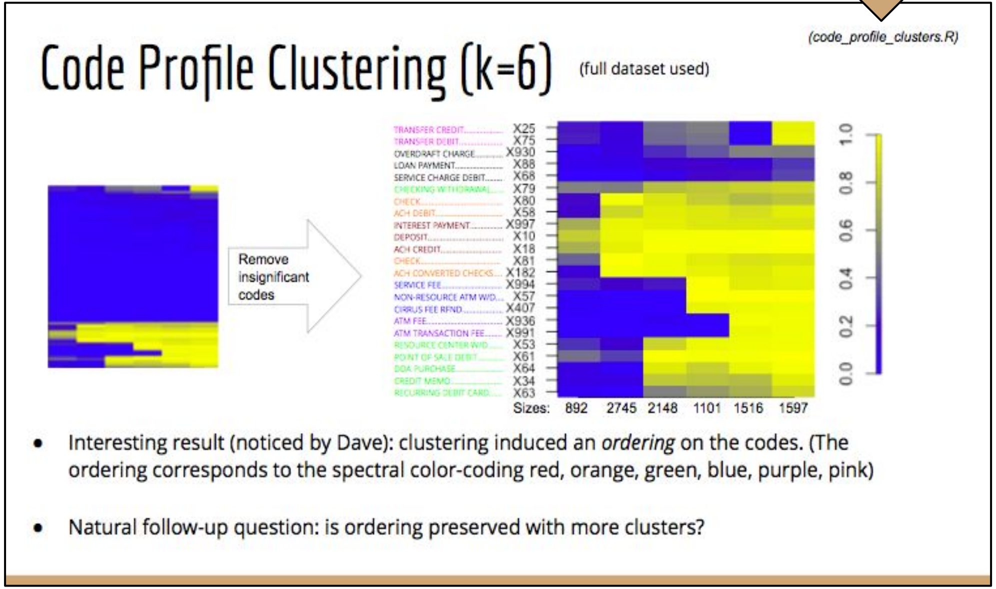
Justin Skycak

June 14, 2016

R code for Figures

program
name

- In my Github folder *felix_clustering*
- In my first set of slides, labeled upper-right corners of slides with the name of the program that can be used to generate the figures



Last Time

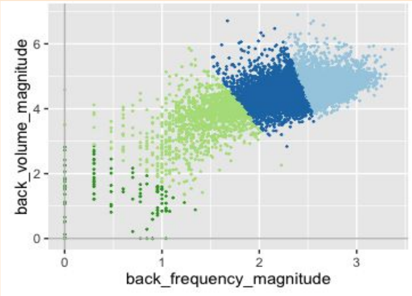
Frequency and *volume* yields good separation; net amount saved or spent does not

Churns are more likely to come from lower-activity accounts

When code usage profiles are clustered, the cluster centers reveal a sequential ordering

This Time

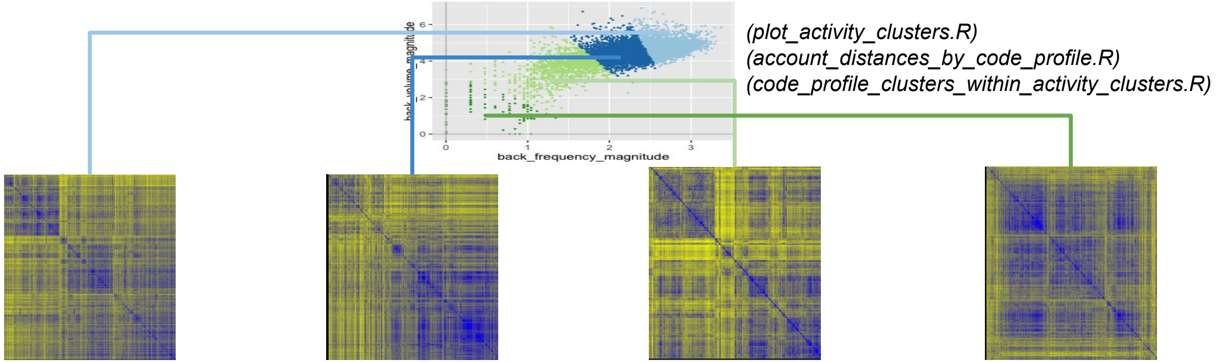
Will perform analysis with activity clusters based only on *frequency* and *volume*

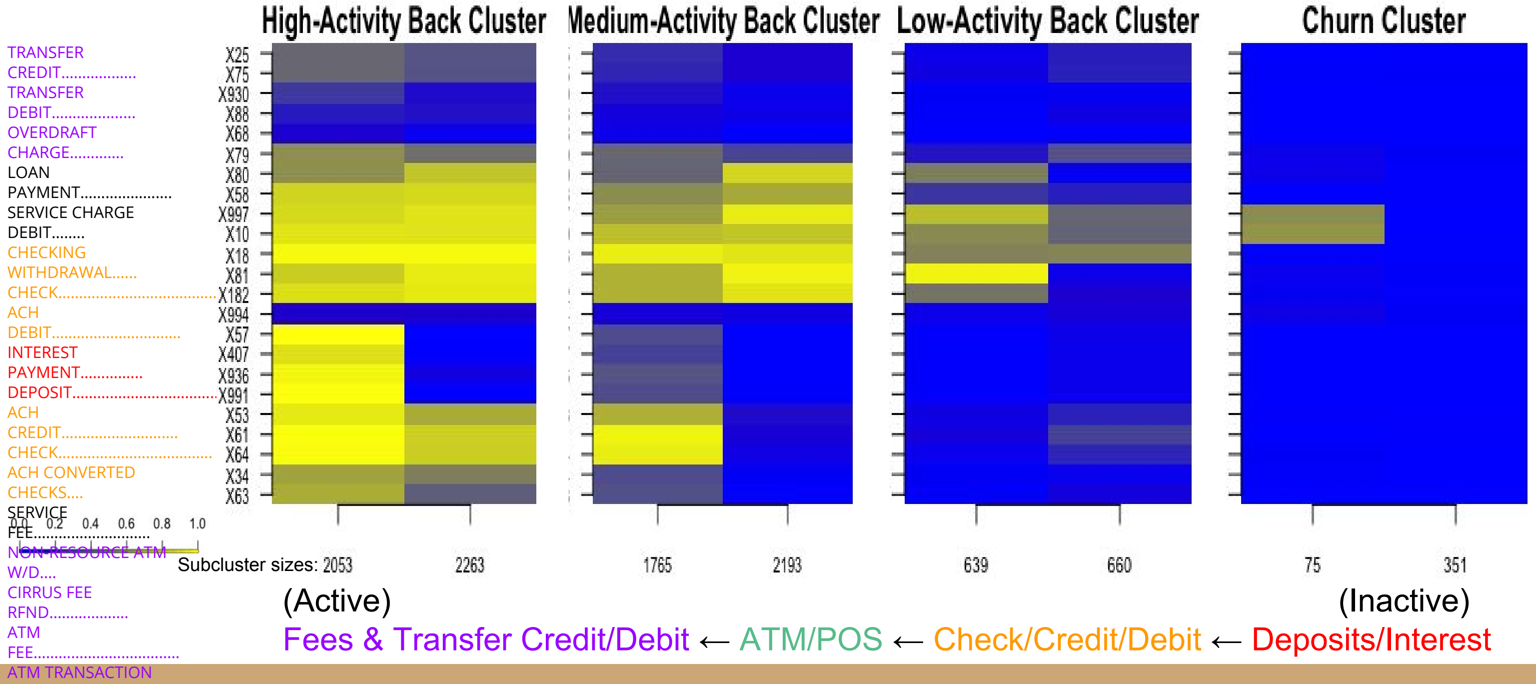


What's the "story" behind churns and code profile ordering?

Code Profiles of Activity Clusters

- Code orderings and activity level orderings appear synchronized



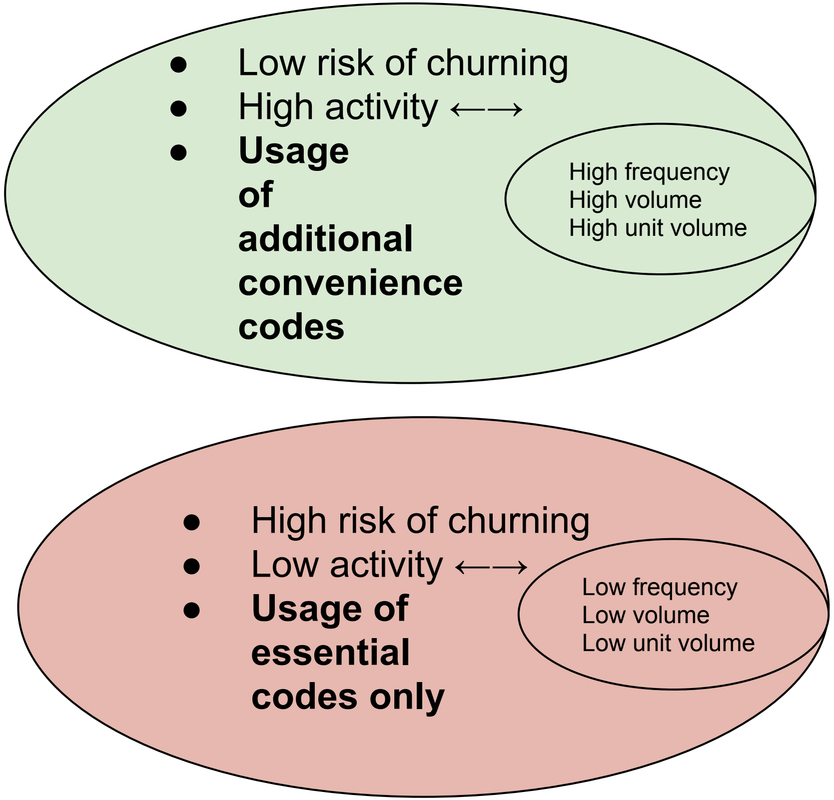


Code Profile Ordering reflects Hierarchy of Convenience

Last time, we saw that

- High or low frequency, volume, and unit volume all tend to occur together as high or low "activity"
- Activity level is inversely related to risk of churning

Now, we see that code profile ordering reflects a "hierarchy of convenience" in the codes!



(Convenience codes) (Essential codes)
Fees & Transfer Credit/Debit ← ATM/POS ← Check/Credit/Debit ← Deposits/Interest

(high_activity_to_churn_code_profile_means.R)

Code Profiles of High-to-Churn Accounts

On average, high-activity accounts that didn't churn tended to use more convenience codes than high-activity accounts which churned.

TRANSFER CREDIT..................

TRANSFER DEBIT....................

CHECK..................................

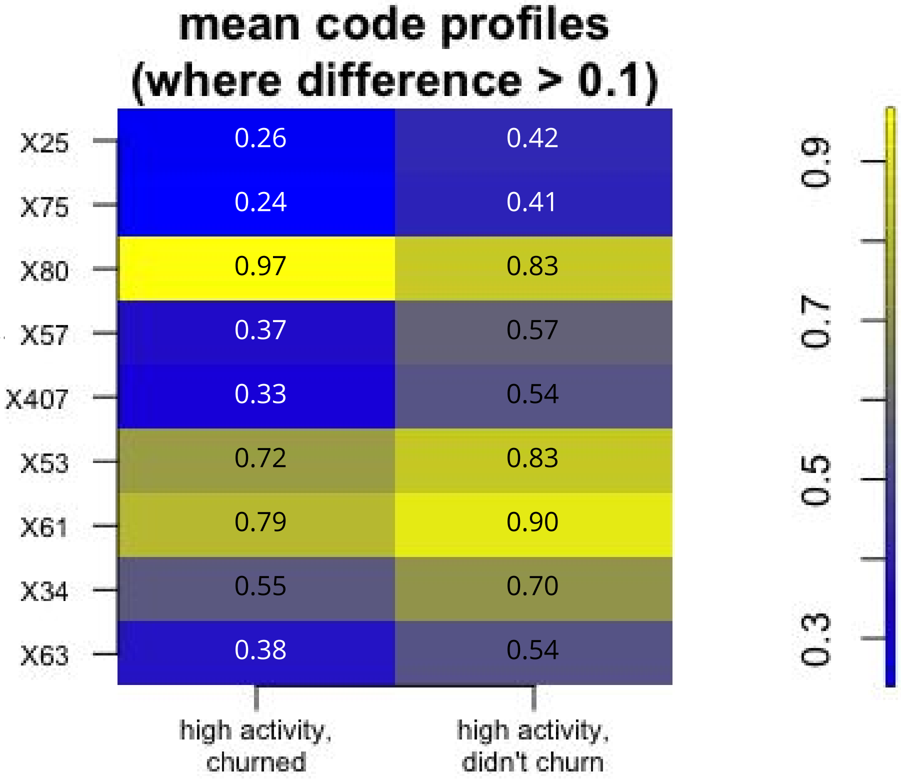
NON-RESOURCE ATM W/D....

CIRRUS FEE RFND..................

RESOURCE CENTER W/D.......

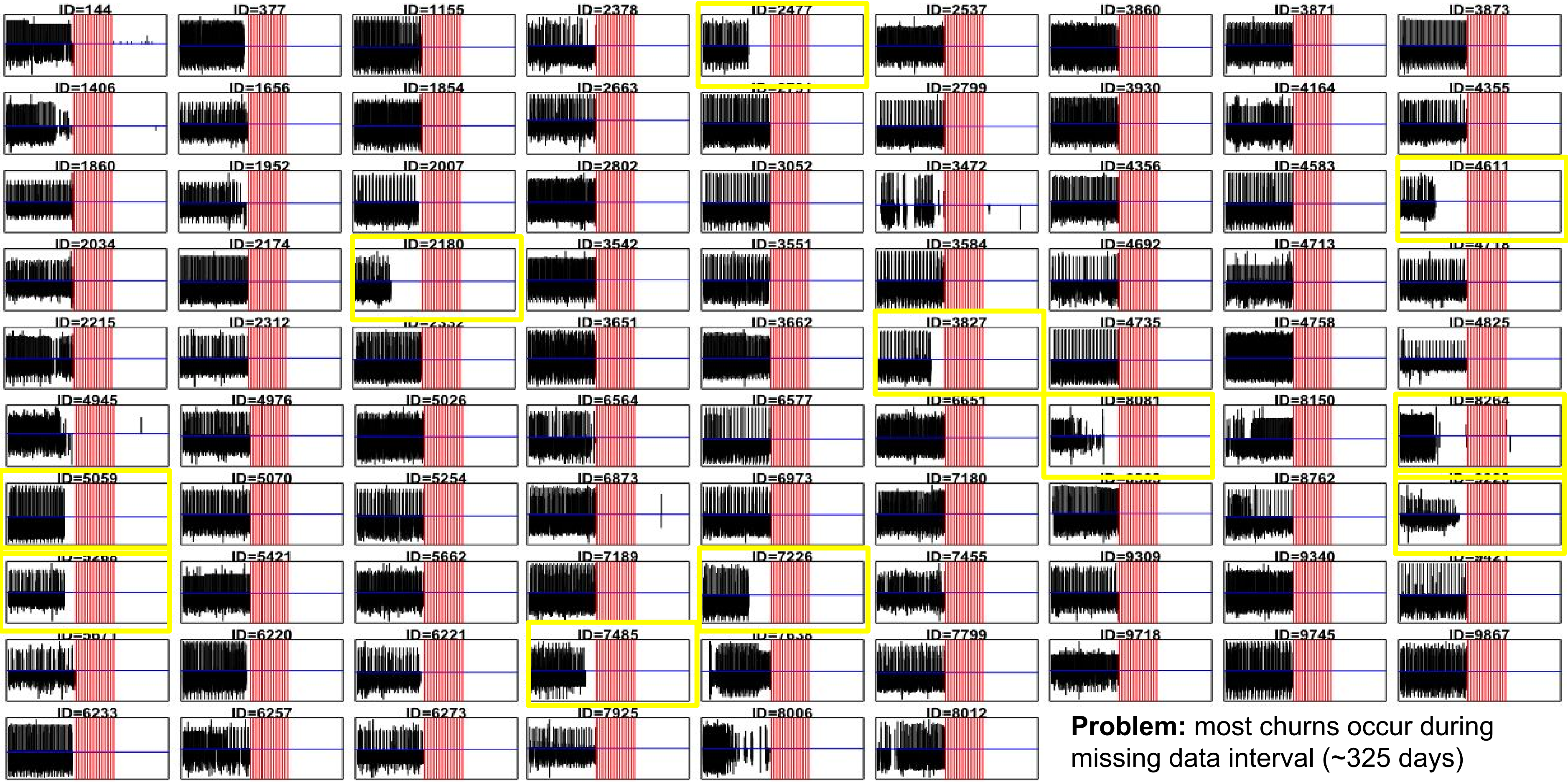
POINT OF SALE DEBIT............

CREDIT MEMO.......................



Time-Series of High-to-Churn Accounts (log scale)

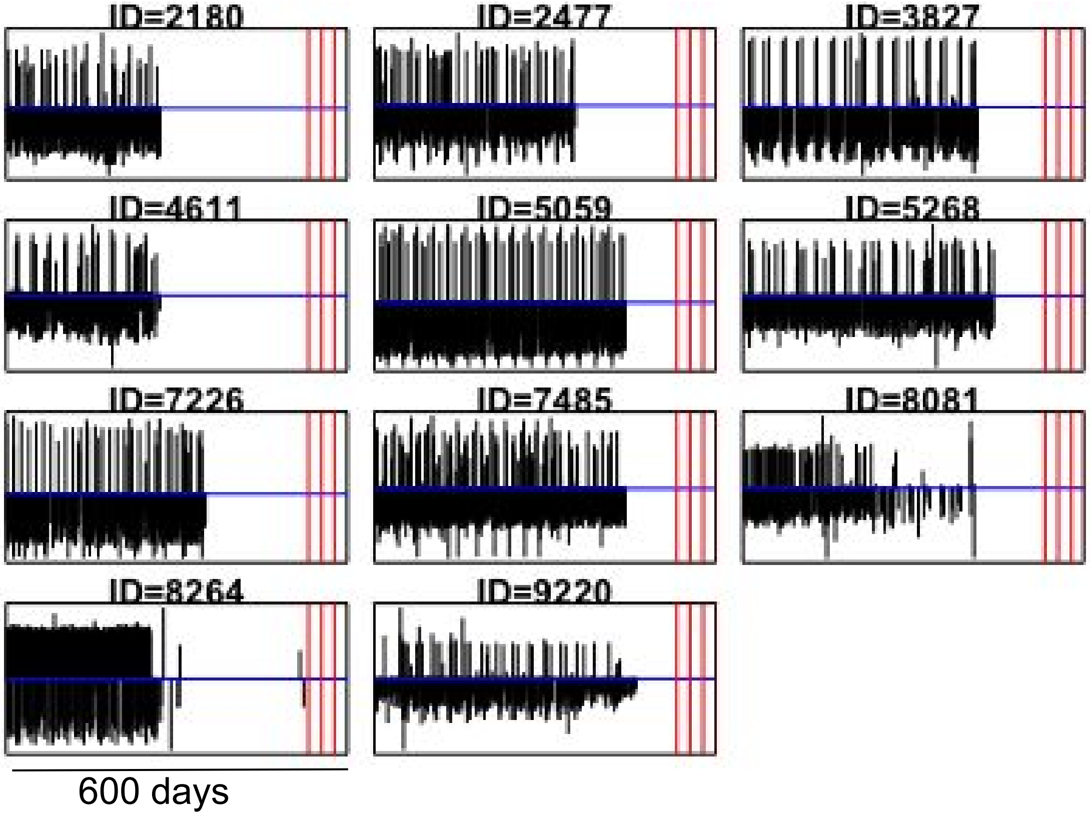
(high_activity_to_churn_time_series_plots.R)



Problem: most churns occur during missing data interval (~325 days)

Time Series of High-to-Churn Accounts

These are the only *observable* churns (i.e. they occurred before the missing data interval)

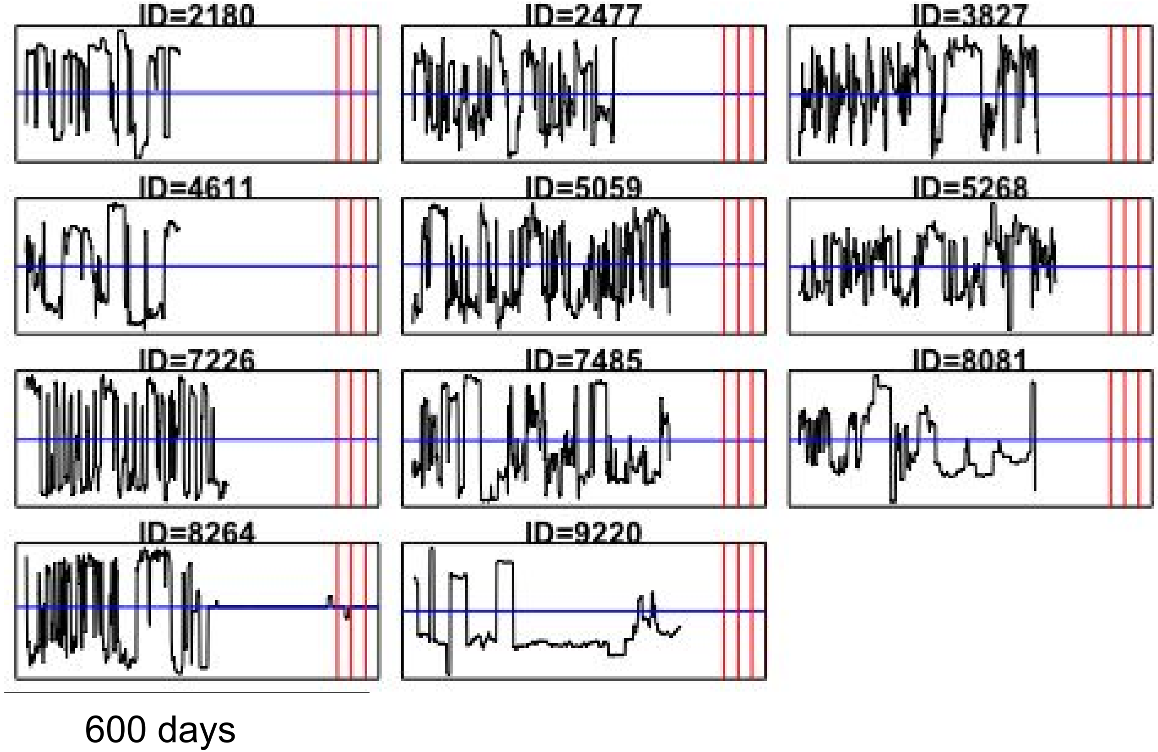


(high_activity_to_churn_time_series_plots.R)

SMA Time Series of High-to-Churn Accounts

30-day simple moving averages (still log scale)

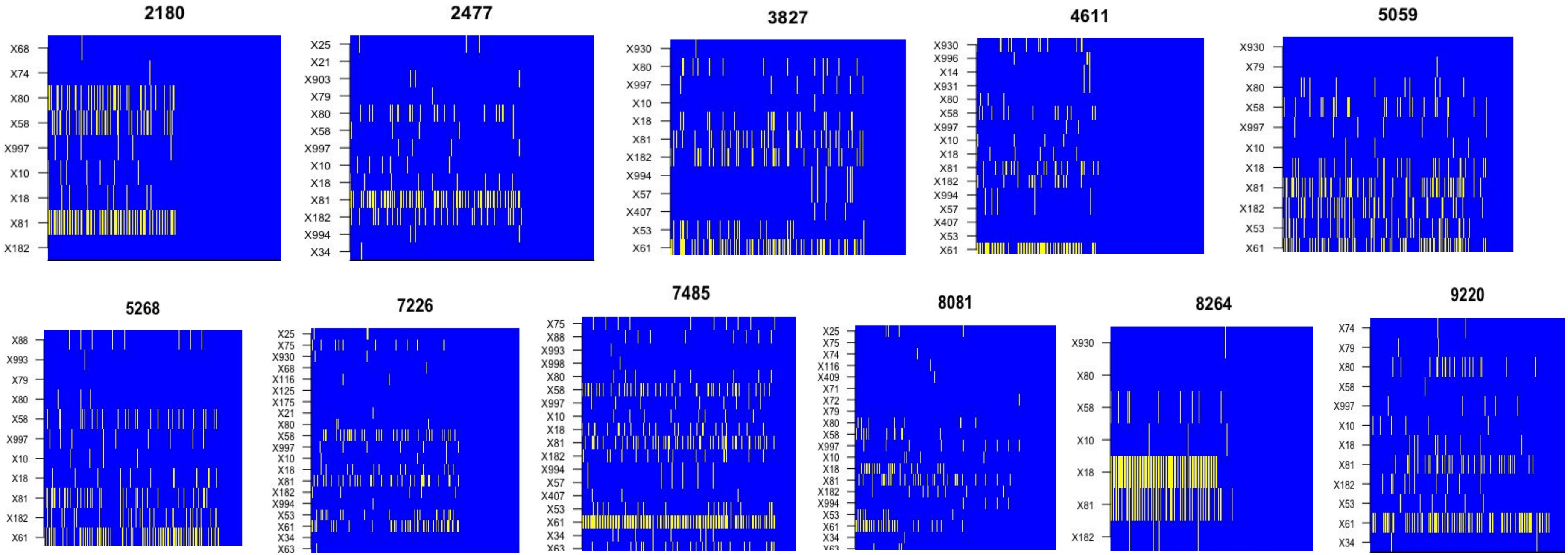
Unable to find any consistent “telltale signs” preceding a churn



(high_activity_to_churn_time_series_plots.R)

Code Use Raster Plots of High-to-Churn Accounts

Vertical axis corresponds to transaction codes, horizontal axis corresponds to time



Again, unable to find any consistent “telltale signs” preceding a churn

(high_activity_to_churn_code_profile_means.R)

Code Profiles of Medium-to-Churn Accounts

Again: on average, medium-activity accounts that didn’t churn tended to use more convenience codes than medium-activity accounts which churned.

CHECKING WITHDRAWAL............

CHECK...

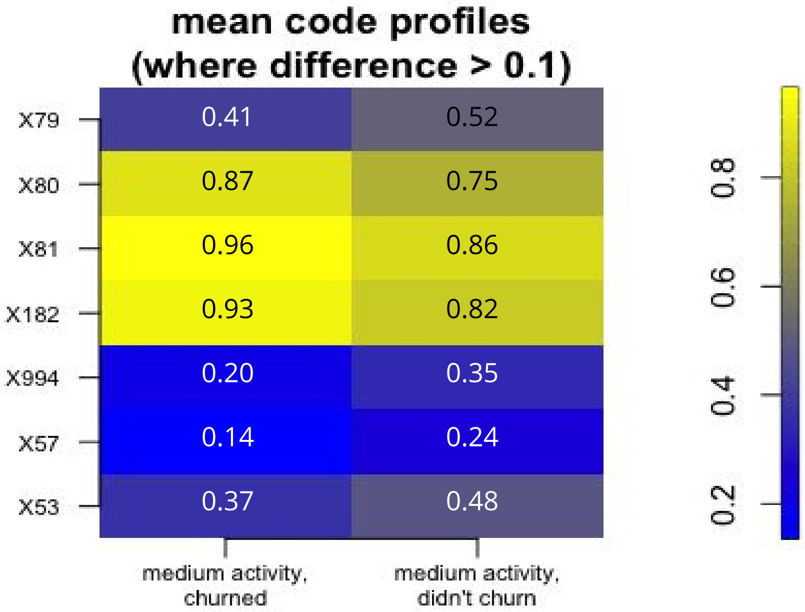
CHECK...

ACH CONVERTED CHECKS..........

SERVICE FEE.................................

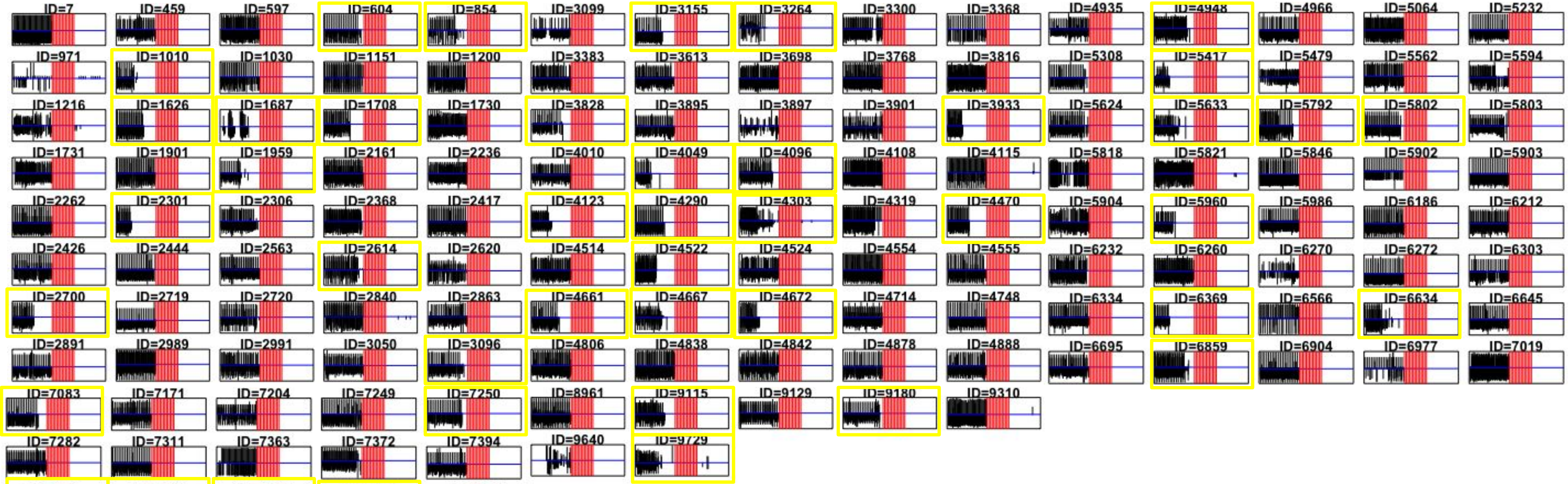
NON-RESOURCE ATM W/D.........

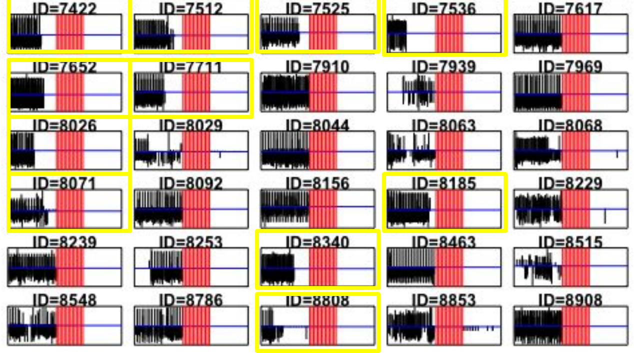
RESOURCE CENTER W/D..........



Time-Series of Medium-to-Churn Accounts (log scale)

(high_activity_to_churn_time_series_plots.R)



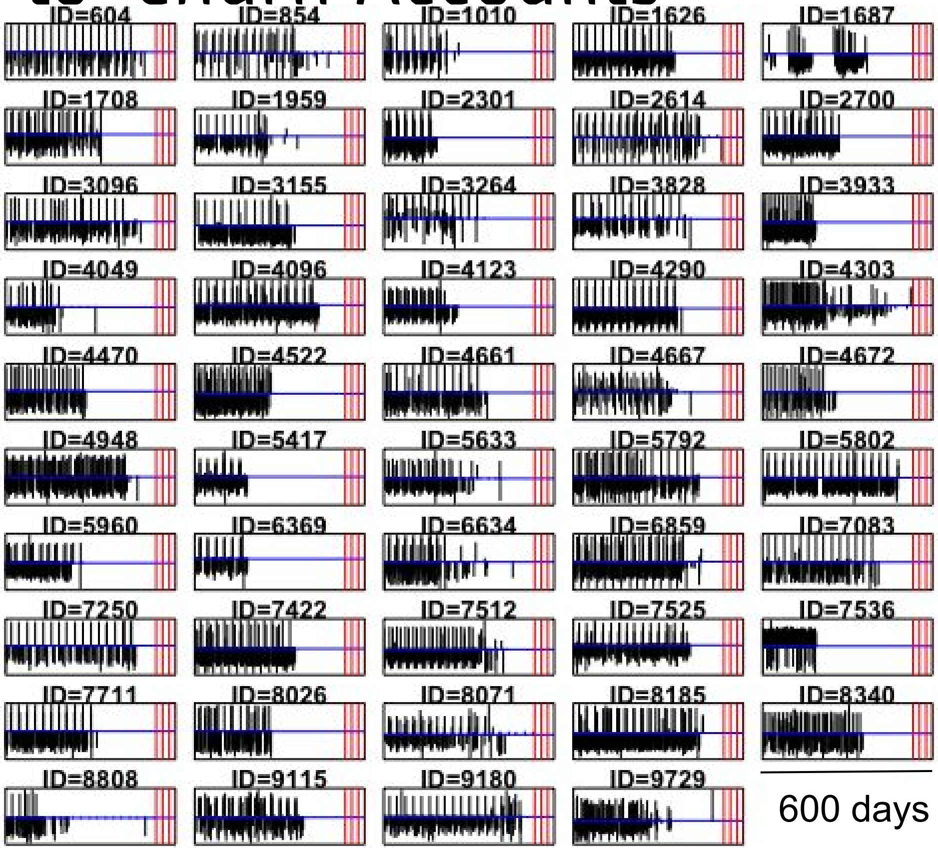


- Again, many churns occurred during the missing data interval. But this time there is a substantial number of observable churns.
- Some high-activity accounts that churned early were probably classified as medium-activity accounts

(high_activity_to_churn_time_series_plots.R)

Time Series of Medium-to-Churn Accounts

These are the only *observable* churns (i.e. they occurred before the missing data interval)



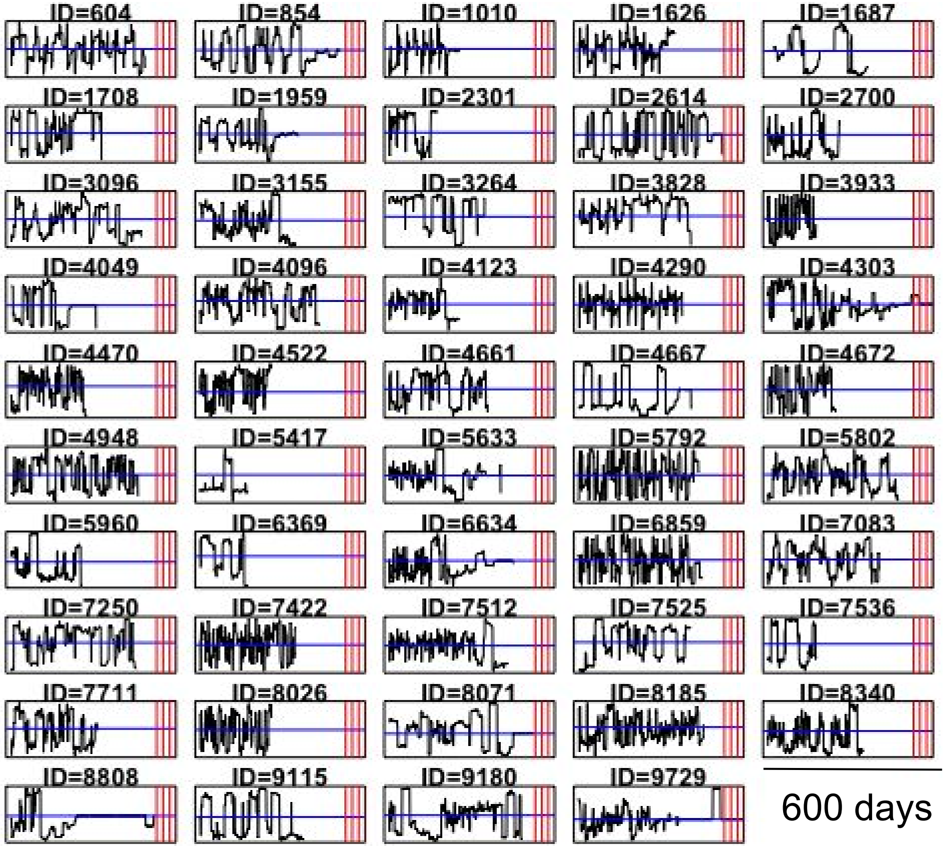
(high_activity_to_churn_time_series_plots.R)

SMA Time Series of Medium-to-Churn Accounts

30-day simple moving averages (still log scale)

Unable to find any consistent “telltale signs” preceding a churn

Won’t show the code raster plots; there are a lot of them and they don’t reveal any consistent “telltale signs” preceding a churn



Next Steps

- Just because I can't see any consistent "telltale signs" preceding a churn doesn't mean there aren't any.
- Is this a job for neural nets? I can envision training a convnet to classify accounts into "churn imminent" and "no churn imminent" categories based on windows of a code-amount plot, similar to the way Spotify has trained a convnet to classify songs into genres based on their time-frequency plots (http://benanne.github.io/2014/08/05/spotify-cnns.html).
- Caveat: This is pretty heavy machinery I haven't implemented convnets before, so it could take a while.
- Otherwise - **is there an alternative way to extract value from the data, other than predicting churns?**