Data Science for Business
Lecture #7
Understanding our Logistic Regression for the Freemium Exercise

Prof. Alan L. Montgomery

Carnegie Mellon University, Tepper School of Business

email: alanmontgomery@cmu.edu

Lecture Outline

Learning about Freemium adopters from Logistic Regression

Telling a story from Logistic Regression

How to cluster our logistic regression output

Freemium Exercise

Learning about Freemium adopters from Logistic Regression

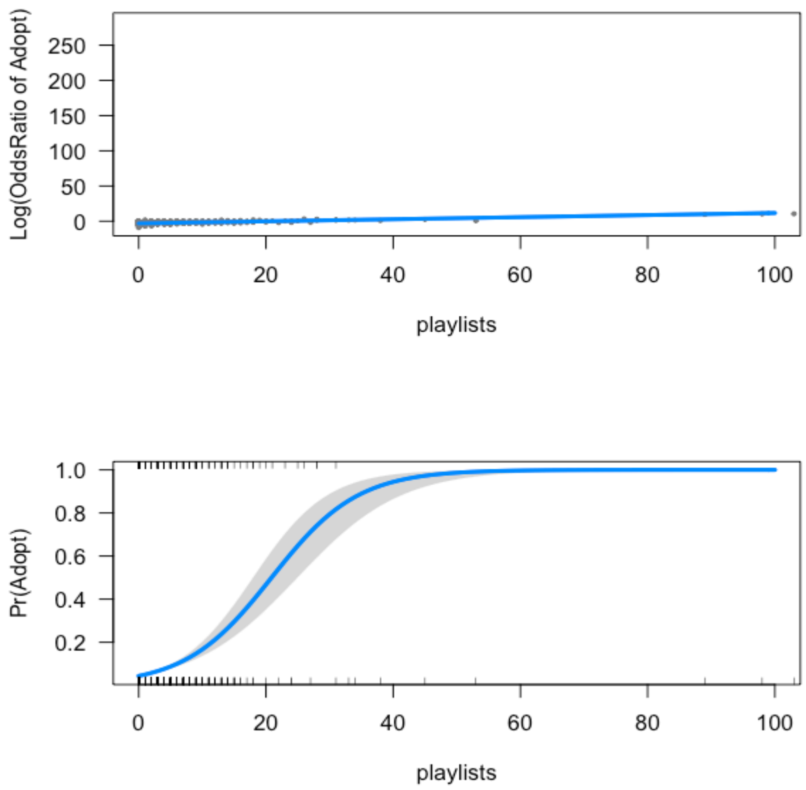
Results from a forward stepwise logistic regression

Warning: Do not copy and paste into your presentation!

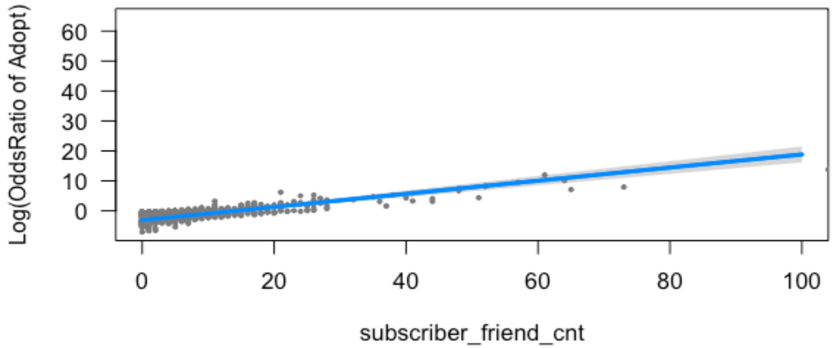
Suggestion: How can you illustrate your model or tell a story?

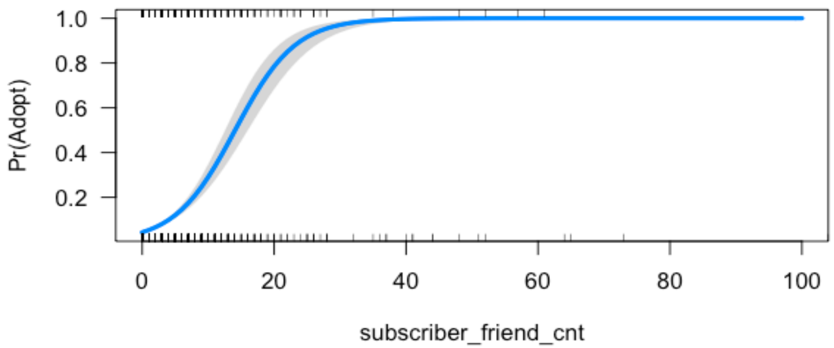
```
> summary(fwd)

Call:
glm(formula = adopter ~ lovedTracks + songsListened + subscriber_friend_cnt +
    age + male + good_country + playlists + friend_cnt + friend_country_cnt +
    avg_friend_age + subscriber_friend_cnt:age + good_country:playlists +
    subscriber_friend_cnt:playlists + lovedTracks:friend_cnt +
    friend_cnt:friend_country_cnt, family = "binomial", data = rfreemium[trainsample,
    crvarlist])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.9621  -0.3652  -0.3149  -0.2861   4.7057

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -4.556e+00  9.228e-02 -49.366  < 2e-16 ***
lovedTracks                      9.275e-04  5.750e-05  16.130  < 2e-16 ***
songsListened                    6.874e-06  4.936e-07  13.925  < 2e-16 ***
subscriber_friend_cnt            3.921e-01  3.306e-02  11.861  < 2e-16 ***
age                              2.808e-02  3.166e-03   8.868  < 2e-16 ***
male                             4.495e-01  4.440e-02  10.125  < 2e-16 ***
good_country                    -2.504e-01  4.522e-02  -5.536 3.09e-08 ***
playlists                        2.153e-01  1.874e-02  11.489  < 2e-16 ***
friend_cnt                       1.664e-03  7.974e-04   2.086   0.0369 *
friend_country_cnt               1.837e-02  4.405e-03   4.169 3.05e-05 ***
avg_friend_age                   2.200e-02  3.266e-03   6.736 1.63e-11 ***
subscriber_friend_cnt:age       -7.090e-03  1.003e-03  -7.069 1.56e-12 ***
good_country:playlists          -1.834e-01  1.977e-02  -9.277  < 2e-16 ***
subscriber_friend_cnt:playlists -1.604e-02  1.872e-03  -8.569  < 2e-16 ***
lovedTracks:friend_cnt          -3.828e-06  5.340e-07  -7.169 7.58e-13 ***
friend_cnt:friend_country_cnt   -6.914e-05  8.544e-06  -8.092 5.87e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 31734  on 64466  degrees of freedom
Residual deviance: 29300  on 64451  degrees of freedom
AIC: 29332

Number of Fisher Scoring iterations: 7
```

Visualize the relationship for playlists



Visualize the relationship for subscriber_friend_cnt





What do interactions in our logistic regression mean?

```
> summary(fwd)

Call:
glm(formula = adopter ~ lovedTracks + songsListened + subscriber_friend_cnt +
    age + male + good_country + playlists + friend_cnt + friend_country_cnt +
    avg_friend_age + subscriber_friend_cnt:age + good_country:playlists +
    subscriber_friend_cnt:playlists + lovedTracks:friend_cnt +
    friend_cnt:friend_country_cnt, family = "binomial", data = rfreemium[trainsample,
    crvarlist])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.9621  -0.3652  -0.3149  -0.2861   4.7057

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                    -4.556e+00  9.228e-02 -49.366  < 2e-16 ***
lovedTracks                     9.275e-04  5.750e-05  16.130  < 2e-16 ***
songsListened                   6.874e-06  4.936e-07  13.925  < 2e-16 ***
subscriber_friend_cnt           3.921e-01  3.306e-02  11.861  < 2e-16 ***
age                             2.808e-02  3.166e-03   8.868  < 2e-16 ***
male                            4.495e-01  4.440e-02  10.125  < 2e-16 ***
good_country                   -2.504e-01  4.522e-02  -5.536 3.09e-08 ***
playlists                       2.153e-01  1.874e-02  11.489  < 2e-16 ***
friend_cnt                      1.664e-03  7.974e-04   2.086   0.0369 *
friend_country_cnt              1.837e-02  4.405e-03   4.169 3.05e-05 ***
avg_friend_age                  2.200e-02  3.266e-03   6.736 1.63e-11 ***
subscriber_friend_cnt:age      -7.090e-03  1.003e-03  -7.069 1.56e-12 ***
good_country:playlists         -1.834e-01  1.977e-02  -9.277  < 2e-16 ***
subscriber_friend_cnt:playlists -1.604e-02  1.872e-03  -8.569  < 2e-16 ***
lovedTracks:friend_cnt         -3.828e-06  5.340e-07  -7.169 7.58e-13 ***
friend_cnt:friend_country_cnt  -6.914e-05  8.544e-06  -8.092 5.87e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 31734  on 64466  degrees of freedom
Residual deviance: 29300  on 64451  degrees of freedom
AIC: 29332

Number of Fisher Scoring iterations: 7
```

What does the R formula that include “subscriber_friend_cnt:playlists” *mean?*

- Remember when we did our stepwise regression we asked R to compute “adopter ~ .^2”, which means that we want R to compute *every* interaction and squared term in the model.
- Suppose we have a dataset with variables: y, x1, x2, then y~.^2 is equivalent to y=x1+x2+x1:x2+x1^2+x2^2, R automatically creates the independent variables x1^2, x2^2, and x1*x2 which is represented in the formula as x1:x2.

To understand this interaction is to look for all the places where playlists occurs in our parameters. We can write our score and focus on the terms that involve playlists:

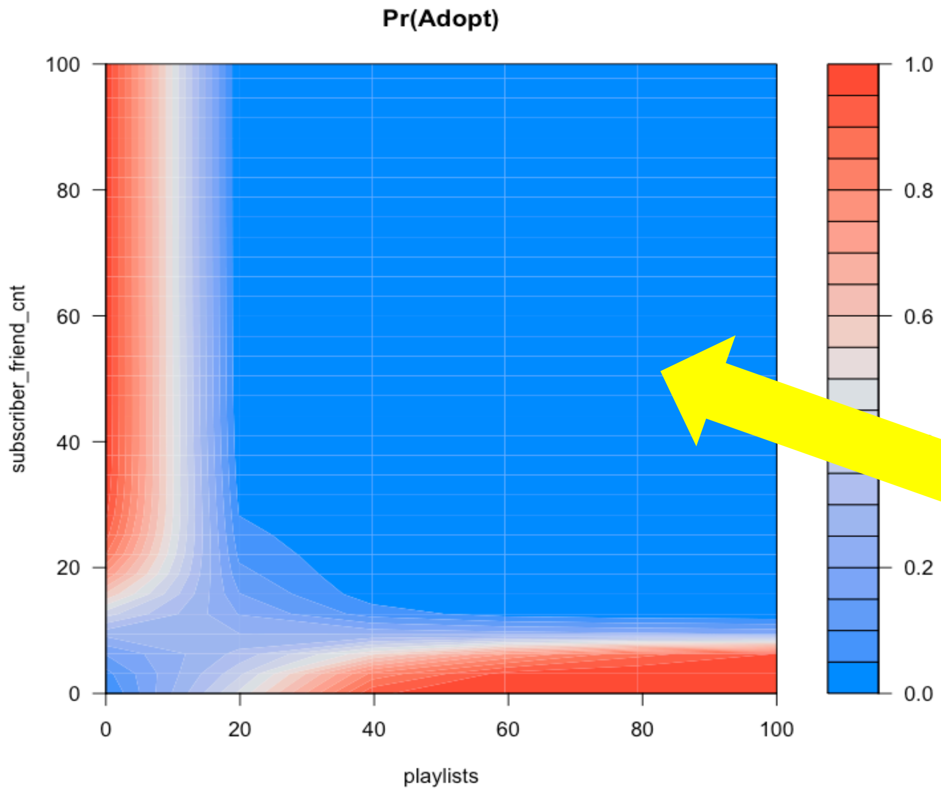
$$Score = \ldots + 0.2153\, playlists - .01834\ goodcountry \times playlists$$
$$- .01604\ subscriber\_friend\_cnt \times playlists + \ldots$$

Let’s rewrite these other terms to see that the interactions are modifying the effect of playlists:

$$Score = \ldots + (0.2153 - .01834\ goodcountry - .01604\ subscriber\_friend\_cnt) \times playlists + \ldots$$

For example, if the user comes from goodcountry=1 and has subscriber_friend_cnt=1 then the playlist coefficient is 0.18092. Our interaction term tells us how the effect of playlists are modified by other variables, notice the effect of playlists is lessened as these other variables increase.

Illustrating the Interaction between playlists and subscriber_friend_cnt with visreg



Notice that positive effect on playlists means that the more playlists the greater the probability of adopting

However, as the negative interaction between playlists and subscriber_friend_cnt means that as one increases the other will decrease

Notice that the area on the diagonal shows very little adoption. In other words, if a users has many subscriber friends and many playlists, we do not expect the user to adopt.

How do we know which variables are important?

Look at the coefficients

- Problem: The coefficients are on different scales
- Solution: Standardize, but how?

Compare p-values (or Z-values)

- Small p-values (or large absolute Z-values) correspond with those coefficients that have the biggest impact on the fit of our model

Ask the model to compute a counterfactual:

- What would happen if a value was increased by one standardized unit – what would be the most important variable?
- Problem: What is a good way to standardize a unit?
- Solution: Compute the standard deviation of the variable from the original dataset.

Understanding Calculations of Importance
Why exp and stddev?

Suppose we change x what is the effect?

$$\ln\left(\frac{\Pr(y_i=1)}{1-\Pr(y_i=1)}\right)=\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}+\cdots+\beta_p x_{pi}$$

If we change x_{ji} to $x_{ji}{}'$ then we can state the change in the odds

$$\ln\left(\frac{\Pr(y_i=1)}{1-\Pr(y_i=1)}\,|\,x_{ji}{}'\right)-\ln\left(\frac{\Pr(y_i=1)}{1-\Pr(y_i=1)}\,|\,x_{ji}\right)=\beta_j\left(x_{ji}{}'-x_{ji}\right)$$

So the impact in the natural units of the odds ratio is:

$$\exp\left\{\beta_j\left(x_{ji}{}'-x_{ji}\right)\right\}=\exp\left\{\beta_j\right\}^{\left(x_{ji}{}'-x_{ji}\right)}$$

We are interested in a one standard deviation increase in variable j and we care about the magnitude of the effect relative to the base. The base odds is 1, so compute importance of variable j as:

$$Importance_j=\left|1\text{-}\exp\left\{\beta_j\right\}^{\left(stddev(x_j)\right)}\right|$$

Building Simulators and Clustering with Logistic Regression to Tell a Better Story

How do we tell a story from this data?

```
> summary(fwd)

Call:
glm(formula = adopter ~ lovedTracks + songsListened + subscriber_friend_cnt +
    age + male + good_country + playlists + friend_cnt + friend_country_cnt +
    avg_friend_age + subscriber_friend_cnt:age + good_country:playlists +
    subscriber_friend_cnt:playlists + lovedTracks:friend_cnt +
    friend_cnt:friend_country_cnt, family = "binomial", data = rfreemium[trainsample,
    crvarlist])

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.9621  -0.3652  -0.3149  -0.2861   4.7057

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -4.556e+00  9.228e-02 -49.366  < 2e-16 ***
lovedTracks                      9.275e-04  5.750e-05  16.130  < 2e-16 ***
songsListened                    6.874e-06  4.936e-07  13.925  < 2e-16 ***
subscriber_friend_cnt            3.921e-01  3.306e-02  11.861  < 2e-16 ***
age                              2.808e-02  3.166e-03   8.868  < 2e-16 ***
male                             4.495e-01  4.440e-02  10.125  < 2e-16 ***
good_country                    -2.504e-01  4.522e-02  -5.536 3.09e-08 ***
playlists                        2.153e-01  1.874e-02  11.489  < 2e-16 ***
friend_cnt                       1.664e-03  7.974e-04   2.086   0.0369 *
friend_country_cnt               1.837e-02  4.405e-03   4.169 3.05e-05 ***
avg_friend_age                   2.200e-02  3.266e-03   6.736 1.63e-11 ***
subscriber_friend_cnt:age       -7.090e-03  1.003e-03  -7.069 1.56e-12 ***
good_country:playlists          -1.834e-01  1.977e-02  -9.277  < 2e-16 ***
subscriber_friend_cnt:playlists -1.604e-02  1.872e-03  -8.569  < 2e-16 ***
lovedTracks:friend_cnt          -3.828e-06  5.340e-07  -7.169 7.58e-13 ***
friend_cnt:friend_country_cnt   -6.914e-05  8.544e-06  -8.092 5.87e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 31734  on 64466  degrees of freedom
Residual deviance: 29300  on 64451  degrees of freedom
AIC: 29332

Number of Fisher Scoring iterations: 7
```

Extract Data Into Spreadsheet

	A	B	C	D	E	F	G	H	I
1		rn	Estimate	Std. Error	z value	Pr(>|z|)	meandata	sddata	userdata
2	1	(Intercept)	-4.5556506	0.09228335	-49.365901	0	1	0	1
3	2	lovedTracks	0.00092747	5.75E-05	16.1301231	1.57E-58	78.0015822	305.429401	420
4	3	songsListene	6.87E-06	4.94E-07	13.9252467	4.45E-44	12934.6135	25483.6632	22403
5	4	subscriber_fı	0.39206401	0.03305565	11.8607266	1.89E-32	0.3366994	2.31357166	1
6	5	age	0.02807996	0.00316631	8.8683647	7.42E-19	24.3744319	4.94264743	34
7	6	male	0.44950159	0.04439708	10.1245765	4.30E-24	0.62367736	0.38628242	0
8	7	good_countr	-0.2503658	0.04522447	-5.536069	3.09E-08	0.36692536	0.38366481	0
9	8	playlists	0.21530428	0.01874009	11.4889655	1.50E-30	0.5449765	7.82243525	27
10	9	friend_cnt	0.00166372	0.00079743	2.08634504	0.03694738	12.3498069	49.1193501	16
11	10	friend_count	0.01836654	0.00440499	4.16948143	3.05E-05	2.80228644	5.02288058	11
12	11	avg_friend_a	0.02199917	0.00326597	6.73587511	1.63E-11	24.5945253	5.11833992	32
13	12	subscriber_fı	-0.0070905	0.00100305	-7.0689097	1.56E-12	8.78296001	82.5032288	34
14	13	good_countr	-0.1834274	0.01977252	-9.2768846	1.75E-20	0.19594076	1.33911363	0
15	14	subscriber_fı	-0.0160377	0.00187161	-8.5689415	1.04E-17	0.39525649	10.4327184	27
16	15	lovedTracks:f	-3.83E-06	5.34E-07	-7.1686076	7.58E-13	3532.89365	61910.939	6720
17	16	friend_cnt:fr	-6.91E-05	8.54E-06	-8.0920236	5.87E-16	215.958072	4106.72477	176

Let's Improve the Formatting and compute Importance

	A	B	C	D	E	F	G	H	I	J	K
1		rn	Estimate	Std. Error	z value	Pr(>|z|)	meandata	sddata	userdata		importance
2	1	(Intercept)	-4.556	0.092	-49.366	0.000	1.000	0.000	1		0.0
3	2	lovedTracks	0.001	0.000	16.130	0.000	78.002	305.429	420		0.3
4	3	songsListened	0.000	0.000	13.925	0.000	12934.613	25483.663	22403		0.2
5	4	subscriber_friend_cnt	0.392	0.033	11.861	0.000	0.337	2.314	1		1.5
6	5	age	0.028	0.003	8.868	0.000	24.374	4.943	34		0.1
7	6	male	0.450	0.044	10.125	0.000	0.624	0.386	0		0.2
8	7	good_country	-0.250	0.045	-5.536	0.000	0.367	0.384	0		0.1
9	8	playlists	0.215	0.019	11.489	0.000	0.545	7.822	27		4.4
10	9	friend_cnt	0.002	0.001	2.086	0.037	12.350	49.119	16		0.1
11	10	friend_country_cnt	0.018	0.004	4.169	0.000	2.802	5.023	11		0.1
12	11	avg_friend_age	0.022	0.003	6.736	0.000	24.595	5.118	32		0.1
13	12	subscriber_friend_cnt:age	-0.007	0.001	-7.069	0.000	8.783	82.503	34		0.4
14	13	good_country:playlists	-0.183	0.020	-9.277	0.000	0.196	1.339	0		0.2
15	14	subscriber_friend_cnt:playlists	-0.016	0.002	-8.569	0.000	0.395	10.433	27		0.2
16	15	lovedTracks:friend_cnt	0.000	0.000	-7.169	0.000	3532.894	61910.939	6720		0.2
17	16	friend_cnt:friend_country_cnt	0.000	0.000	-8.092	0.000	215.958	4106.725	176		0.2

Suggested story…

Try using findings from a model like a logistic regression to say…

- "An increase of 7.8 playlists (or one standard deviation) over the average of .55 results in a 4.4 increase in the odds ratio”
- “An increase of 2.3 subscriber_friend_cnt (or one standard deviation) over the average of 0.34 results in a 1.50 increase in the odds ratio”

More simply

- “8 more playlists quadruples our odds”
- “2 more subscriber friends double our odds”
- The goal is to tell a story like adopters help us attract their friends or free subscribers who are engaged are most likely to convert.

The consequence is that we can take someone with a odds of subscribing is 1 to 13 (7% probability)

- “8 more playlists gives them odds of about 1 to 3 (30% probability)”
- “2 more subscriber friends gives them odds of about 1 to 7 (14% probability)”

Caution: We have interactions

- Playlists interacts negatively with good_country and subscriber_friend_cnt, so playlists isn’t as strong for those from good countries, nor for those with a lot of subscribe friends – It is still positive just not as strong.

An important assumption in your analysis is causation (e.g., if we increase LovedTracks then more will subscribe). Our predictive model is correlational, so remember that you are making an important conjecture here.

Compute the Score for this User
What is the probability to convert?

	A	B	C	G	I	J	K
1		rn	Estimate	meandata	userdata		score
2	1	(Intercept)	-4.556	1.000	1		-4.556
3	2	lovedTracks	0.001	78.002	420		0.390
4	3	songsListened	0.000	12934.613	22403		0.154
5	4	subscriber_friend_cnt	0.392	0.337	1		0.392
6	5	age	0.028	24.374	34		0.955
7	6	male	0.450	0.624	0		0.000
8	7	good_country	-0.250	0.367	0		0.000
9	8	playlists	0.215	0.545	27		5.813
10	9	friend_cnt	0.002	12.350	16		0.027
11	10	friend_country_cnt	0.018	2.802	11		0.202
12	11	avg_friend_age	0.022	24.595	32		0.704
13	12	subscriber_friend_cnt:age	-0.007	8.783	34		-0.241
14	13	good_country:playlists	-0.183	0.196	0		0.000
15	14	subscriber_friend_cnt:playlists	-0.016	0.395	27		-0.433
16	15	lovedTracks:friend_cnt	0.000	3532.894	6720		-0.026
17	16	friend_cnt:friend_country_cnt	0.000	215.958	176		-0.012
18							
19						score=	3.369
20						prob=	97%

Freemium Logistic Regression
Understanding our Score

$$\Pr(Adopter) = \frac{\exp\{Score\}}{1+\exp\{Score\}} \Rightarrow$$

$$Score =$$

$$-4.5$$

Heavy Users

$$+.2 \times Playlists$$

With Lots of Subscriber Friends (who are younger)

$$+.4 \times SubscriberFriendCnt$$

$$-.007 \times SubscriberFriendCnt * Age$$

Older & Male

$$+.4 \times Male$$

$$+.03 \times Age$$

$$+\cdots$$

Express Score as difference from mean

Why is this user likely to adopt?

	A	B	C	G	I	J	K	L	M	N
1		rn	Estimate	meandata	userdata		score		userdata-mean	score
2	1	(Intercept)	-4.556	1.000	1		-4.556			-2.792
3	2	lovedTracks	0.001	78.002	420		0.390		341.998	0.317
4	3	songsListened	0.000	12934.613	22403		0.154		9468.387	0.065
5	4	subscriber_friend_cnt	0.392	0.337	1		0.392		0.663	0.260
6	5	age	0.028	24.374	34		0.955		9.626	0.270
7	6	male	0.450	0.624	0		0.000		-0.624	-0.280
8	7	good_country	-0.250	0.367	0		0.000		-0.367	0.092
9	8	playlists	0.215	0.545	27		5.813		26.455	5.696
10	9	friend_cnt	0.002	12.350	16		0.027		3.650	0.006
11	10	friend_country_cnt	0.018	2.802	11		0.202		8.198	0.151
12	11	avg_friend_age	0.022	24.595	32		0.704		7.405	0.163
13	12	subscriber_friend_cnt:age	-0.007	8.783	34		-0.241		25.217	-0.179
14	13	good_country:playlists	-0.183	0.196	0		0.000		-0.196	0.036
15	14	subscriber_friend_cnt:playlists	-0.016	0.395	27		-0.433		26.605	-0.427
16	15	lovedTracks:friend_cnt	0.000	3532.894	6720		-0.026		3187.106	-0.012
17	16	friend_cnt:friend_country_cnt	0.000	215.958	176		-0.012		-39.958	0.003
18										
19						score=	3.369		score=	3.369
20						prob=	97%		prob=	97%

What if… lovedTracks doubles? Playlists go to 0? We have a female aged 50?

Understanding Calculations

If we want to understand the contribution of each value relative to the average person:

$$\ln\left(\frac{\Pr(y_i=1)}{1-\Pr(y_i=1)}\right)=\beta_0+\beta_1 x_{1i}+\beta_2 x_{2i}+\cdots+\beta_p x_{pi}$$

We can add and subtract the average of each variable (μ_j):

$$=\beta_0+\beta_1\left(x_{1i}-\mu_1+\mu_1\right)+\beta_2\left(x_{2i}-\mu_2+\mu_2\right)+\cdots+\beta_p\left(x_{pi}-\mu_p+\mu_p\right)$$

$$=\beta_0+\beta_1\left(x_{1i}-\mu_1\right)+\beta_1\mu_1+\beta_2\left(x_{2i}-\mu_2\right)+\beta_2\mu_2+\cdots+\beta_p\left(x_{pi}-\mu_p\right)+\beta_p\mu_p$$

$$=\left(\beta_0+\beta_1\mu_1+\beta_2\mu_2+\cdots+\beta_p\mu_p\right)+\beta_1\left(x_{1i}-\mu_1\right)+\beta_2\left(x_{2i}-\mu_2\right)+\cdots+\beta_p\left(x_{pi}-\mu_p\right)$$

After rearranging the terms we can think of the Log-OddsRatio as the relative contribution of each value compared to its mean plus a constant:

$$=\beta_0^*+\beta_1\left(x_{1i}-\mu_1\right)+\beta_2\left(x_{2i}-\mu_2\right)+\cdots+\beta_p\left(x_{pi}-\mu_p\right)$$

How to cluster our logistic regression output

Unlike decision trees we do not get “clusters” from the logistic regression output

Our Logistic Regression Helps Classify

But are all users with same probability the same?

Notice these users all have similar probabilities of converting, but one has 384 lovedTracks (user 1) and the other has 0 (user 6). Are they really the same?

```
> head(round(cbind(userpred,xmodeldata),2))
  userpred (Intercept) lovedTracks songsListened subscriber_friend_cnt   age male good_country avg_friend_age
1     0.08           1         348          8414                     1 24.39 0.00         1.00          30.29
2     0.04           1           0          1943                     0 24.39 0.62         0.37          30.50
3     0.03           1         194          9687                     0 22.00 0.00         1.00          22.57
4     0.06           1          12         26863                     0 31.00 0.00         0.00          24.61
5     0.04           1           0           187                     0 24.39 0.62         0.37          24.61
6     0.07           1           0             0                     0 35.00 0.00         0.00          28.00
  friend_country_cnt friend_cnt playlists shouts_Missing tenure good_country_Missing avg_friend_male
1                 14         20         1              0     59                    0            0.74
2                  1          3         0              0     34                    1            0.33
3                  1          8         1              0     59                    0            0.43
4                  0          0         0              0     55                    0            0.63
5                  1          1         0              0     52                    1            1.00
6                  2          2         0              0     35                    0            1.00
  avg_friend_male_Missing
1                       0
2                       0
3                       0
4                       1
5                       0
6                       0
```

How could we use cluster analysis to group users so that each group has similar reasons for converting?

Clustering Logistic Regression Predictions

First, notice we do not want to cluster the data (already did that) since it does not *optimize* out ability to predict adopters. Second, we do not want to simply sort consumers, since we may get very different reasons why users adopt

Idea: Weight the raw data by the contribution each variable gives to the score (e.g., multiply the observation by coefficient)

Using Data compute the "New Data" or wmodeldata to use for cluster

Original data:

```
> xmodeldata[1,1:4]
  (Intercept)    lovedTracks    songsListened subscriber_friend_cnt
            1            348             8414                     1
```

Parameters:

```
> parm[1:4]
  (Intercept)    lovedTracks    songsListened subscriber_friend_cnt
-4.284221e+00   7.053991e-04     8.030343e-06          8.810018e-02
```

New data:

```
> wuserdata[1,1:4]
  (Intercept)    lovedTracks    songsListened subscriber_friend_cnt
  -4.28422100     0.24547889       0.06756731            0.08810018
```

0.2547889
= 348 x 0.0007054

Example of the data that we are clustering
Weighs the original data by their contribution to the score which measures why users adopt

The first 6 rows of our original and weighted data:

```
> head(round(cbind(userpred,xmodeldata[,1:4]),2))
  userpred (Intercept) lovedTracks songsListened subscriber_friend_cnt
1     0.08           1         348          8414                     1
2     0.04           1           0          1943                     0
3     0.03           1         194          9687                     0
4     0.06           1          12         26863                     0
5     0.04           1           0           187                     0
6     0.07           1           0             0                     0
> head(round(cbind(userpred,wuserdata[,1:4]),2))
  userpred (Intercept) lovedTracks songsListened subscriber_friend_cnt
1     0.08       -4.28        0.25          0.07                  0.09
2     0.04       -4.28        0.00          0.02                  0.00
3     0.03       -4.28        0.14          0.08                  0.00
4     0.06       -4.28        0.01          0.22                  0.00
5     0.04       -4.28        0.00          0.00                  0.00
6     0.07       -4.28        0.00          0.00                  0.00
```

Newvar = Oldvar x Coef

Previously when we clustered our observations, we standardized the data (mean=0, stddev=1), now we are weighting the variables by the (standardized) contribution to the log of the odds-ratio

Perform a k-Means Cluster Analysis
What are the inputs and outputs?

```
# cluster the users into groups
ncluster=10    # number of clusters
set.seed(612490)   # make sure we get the same solution
grpA=kmeans(wuserdata,ncluster,nstart=50)  # add nstart=50 to choose 50 different random seeds
```

Input:

- wuserdata (107,213 users x 16 variables)
- k=10 (how many clusters)

Output:

- grpA$cluster (assignment of each of the 107,213 users to one of the 10 clusters)
- grpA$centers (averages of each of the 16 variables for the 10 clusters)

How will a k-means analysis help us?

What do the clusters mean?
Is this a good solution?

R^2 is 86%

Total of 107,213 observations

How does Pr(Convert) change across clusters?

```
> # what is the rsquare
> grpA$betweenss/grpA$totss
[1] 0.8787008
> # distribution of observations to the clusters
> table(grpA$cluster)

    1     2     3     4     5     6     7     8     9    10
 1674     3 13754 20794     1     8   195    28 69994   762
> describeBy(userpred,group=grpA$cluster,mat=TRUE,digits=2)[,c("item","n","mean","sd","min","max")]
     item     n mean   sd  min  max
X11     1  1674 0.29 0.16 0.01 1.00
X12     2     3 0.15 0.25 0.00 0.44
X13     3 13754 0.10 0.06 0.03 0.99
X14     4 20794 0.04 0.02 0.02 0.83
X15     5     1 1.00   NA 1.00 1.00
X16     6     8 0.67 0.34 0.07 1.00
X17     7   195 0.59 0.30 0.00 1.00
X18     8    28 0.64 0.44 0.00 1.00
X19     9 69994 0.06 0.03 0.02 0.82
X110   10   762 0.35 0.23 0.00 1.00
```

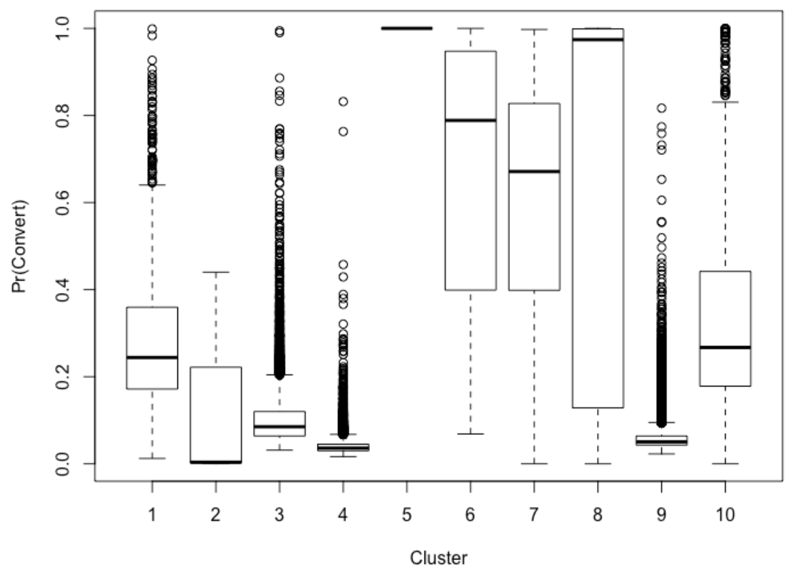
What do the clusters mean?
How do we interpret each cluster?

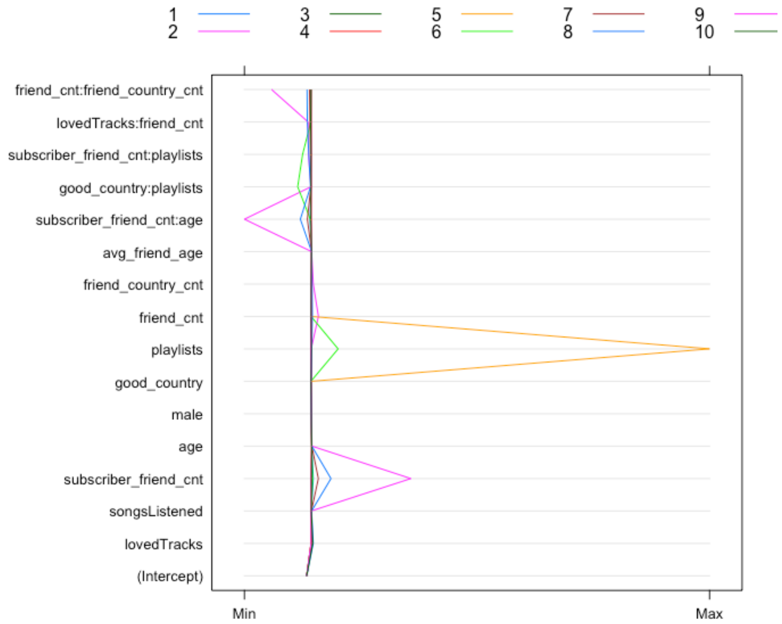
The centers matrix from kmeans gives us the means of each cluster.

```
> round(t(grpA$centers),2)   # transpose so the clusters are in the columns and variables in the rows
```

	1	2	3	4	5	6	7	8	9	10
(Intercept)	-4.56	-4.56	-4.56	-4.56	-4.56	-4.56	-4.56	-4.56	-4.56	-4.56
lovedTracks	0.33	0.15	0.11	0.05	0.00	0.96	0.68	2.30	0.04	1.82
songsListened	0.32	0.10	0.19	0.06	0.01	0.34	0.38	0.57	0.07	0.32
subscriber_friend_cnt	2.33	104.81	0.53	0.00	0.00	1.72	7.67	20.91	0.00	0.43
age	0.73	1.00	0.70	0.64	0.68	0.74	0.79	0.83	0.69	0.72
male	0.27	0.45	0.28	0.00	0.28	0.20	0.29	0.34	0.36	0.32
good_country	-0.09	-0.17	-0.09	-0.09	0.00	-0.15	-0.11	-0.16	-0.09	-0.10
playlists	0.21	0.14	0.13	0.12	418.34	28.42	0.45	0.75	0.09	0.43
friend_cnt	0.20	7.75	0.05	0.02	0.02	0.07	0.70	1.42	0.01	0.07
friend_country_cnt	0.36	2.35	0.12	0.05	0.02	0.21	0.76	1.09	0.03	0.16
avg_friend_age	0.58	0.58	0.57	0.52	0.68	0.63	0.61	0.64	0.54	0.56
subscriber_friend_cnt:age	-1.10	-69.94	-0.24	0.00	0.00	-0.78	-3.87	-11.21	0.00	-0.20
good_country:playlists	-0.08	-0.06	-0.04	-0.04	0.00	-14.01	-0.15	-0.55	-0.03	-0.17
subscriber_friend_cnt:playlists	-0.10	-2.75	-0.01	0.00	0.00	-8.85	-0.67	-2.39	0.00	-0.04
lovedTracks:friend_cnt	-0.18	-3.13	-0.02	0.00	0.00	-0.22	-1.16	-3.90	0.00	-0.35
friend_cnt:friend_country_cnt	-0.25	-41.44	-0.03	0.00	0.00	-0.06	-1.57	-4.16	0.00	-0.06

Tell a story about "Customer Segments" from our 10 Cluster solution





Our 10 Customer Segments

	1	2	3	4	5	6	7	8	9	10
Pr(Convert)	29%	15%	10%	4%	100%	67%	59%	64%	6%	35%
Size	1,674	3	13,754	20,794	1	8	195	28	69,994	762
Positives	Subscriber friend Count	Lots of subscriber friends	More subscribers friends and male than average		Heaviest playlist user	Heavy playlist users	Subscriber friend Count	Subscriber friend count		Loved Tracks
Negatives	Subscriber friends matter less when old	Older subscribers with friends, Lots of foreign friends	Older subscribers with friends			Playlists matter less when from good countries, Inactive friends	Friends are young			
Summary	Very connected	Small	More subscriber friends	About average	Small	Small			Slightly above average	Most [illegible]ive

Summary

Logistic Regression generates a "score" that can be used to predict the probability of an upgrade. Unlike k-Means this is the "optimal" score if we want to identify upgrade.

Cluster Analysis is really helpful when we want to find a story and/or reduce the dimensionality of a dataset

Combining methods helps us find insights