

Homework 4

Hanao Li, HL3202

About The Data

We will be working with a simulated data set related to social media sites. The data are stored in several files:

Profiles.csv: Information about the users with some fields from their profiles.

Connections.csv: Information about which users are connected to other users.

Registrations.csv: Information about history of the user’s account registrations (logins) over time.

Header The first row of the data set includes the column names, and each subsequent row includes one observation of values. Here is a selection of 20 lines from each data file:

Show

10

 entries

Search:

	id	density	age_group	gender	has_profile_photo	num_photos	date_created
1	0009g7fE	Suburban	18-24	F	1	41	2012-04-17
2	003fp4C4	Rural	35-44	M	1	10	2015-08-14
3	008Xfcs6	Rural	45-54	M	1	834	2016-03-17
4	00IKIVvN	Urban	18-24	F	1	927	2013-09-30
5	01sqmyAe	Suburban	55-64	M	1	1096	2013-01-14
6	02J5xPk5	Urban	18-24	M	1	78	2016-08-23
7	02KI5Mtk	Rural	18-24	M	1	515	2016-08-13
8	02sMiAYv	Suburban	45-54	F	0	1184	2015-05-15
9	02uJHCPG	Suburban	45-54	M	1	468	2015-11-17
10	0317C7rA	Suburban	25-34	F	1	400	2016-08-11

Showing 1 to 10 of 20 entries

Previous

1

2Next

Search:

	id	connection_id
1	0009g7fE	0LsKhl8z
2	0009g7fE	0MukfMFq
3	0009g7fE	0RaKGhQ3
4	0009g7fE	0RhatvnN

	id	connection_id
5	0009g7fE	0hJwTkNt
6	0009g7fE	1SI6BX3W
7	0009g7fE	1kmYujTc
8	0009g7fE	27PNx2Kz
9	0009g7fE	2IN0UsqQ
10	0009g7fE	2Ogw7nOx

Showing 1 to 10 of 20 entries

Previous 1 2 NextShow 10 entriesSearch:

	id	registration.time	original.registration.time
1	0009g7fE	2012-04-17T01:16:36Z	2012-04-17T01:16:36.638355Z
2	0009g7fE	2012-04-17T17:12:50Z	2012-04-17T17:12:50.185053Z
3	0009g7fE	2012-04-17T19:50:52Z	2012-04-17T19:50:52.626801Z
4	0009g7fE	2012-04-18T02:31:24Z	2012-04-18T02:31:24.478748Z
5	0009g7fE	2012-04-18T02:58:57Z	2012-04-18T02:58:57.193260Z
6	0009g7fE	2012-04-18T11:28:34Z	2012-04-18T11:28:34.407250Z
7	0009g7fE	2012-04-18T14:56:02Z	2012-04-18T14:56:02.268198Z
8	0009g7fE	2012-04-19T08:04:03Z	2012-04-19T08:04:03.395474Z
9	0009g7fE	2012-04-19T14:24:15Z	2012-04-19T14:24:15.157158Z
10	0009g7fE	2012-04-20T04:48:46Z	2012-04-20T04:48:46.500991Z

Showing 1 to 10 of 20 entries

Previous 1 2 Next

Here is a brief description of each variable across the three files:

Profiles Variables:

- **id**: A unique identifying string for each user.
- **density**: The type of area the user lives in, with categories of Urban, Suburban, and Rural areas.
- **gender**: female (F) or male (M).
- **has_profile_photo**: 1 if yes, 0 if no.
- **num_photos**: This is the number of photos the user has uploaded to the site.
- **date_created**: This is the date that the user first joined the site.

Connections Variables:

- **id**: A unique identifying string for each user.
- **connection_id**: This is the identifier of another user that the user listed under **id** is connected to.

This site chooses to use one-way connections. A user can connect to a second user's profile without requiring that the second user reciprocally connect to the first one. So, for any row in the Connections data, the user labeled with **id** is following the user labeled with **connection_id**. In some cases, pairs of users are mutually following each other, but this is by no means required. For mutual connections, the users will be coupled in two different rows in the two possible orders. Each connection for a single user is recorded in a separate row.

Registrations Variables:

- **id**: A unique identifying string for each user.
- **registration.time**: This is the date and time that a user registered by logging in to the site. Each registration for a user is recorded in a separate row.

Question 1: Classifying Connections

How often do users mutually follow each other, and how often are the connections one-way? We want to investigate this. For the investigation, we'll say that a two-way connection requires two one-way connections (two rows of data) but only counts once. Therefore, the number of overall connections (total one-way plus total two-way) will be less than the overall number of rows of data in the Connections file. With this in mind, answer these questions.

What percentage of all connections are one-way connections, and what percentage of all connections are two-way connections?

```
# Reverse datatable and combine
reverse <- connections[, .(id = connection_id, connection_id = id)]
combined.connections <- rbindlist(l = list(connections,
  reverse))
# Find connections and ids in common
overall.connections <- combined.connections[, .N, by = c(id.name,
  connection.id.name)]
# Remove unnecessary rows
overall.connections <- overall.connections[1:connections[,
  .N], ]

# Number of one way connections
overall.connections[N == 1, .N]
```

```
[1] 1204613
```

```
# Number of two way connections, For 1 two way
# connection, only count once here.
overall.connections[N == 2, .N/2]
```

```
[1] 1262170
```

```
# Percentage of one way connections
round(overall.connections[N == 1, .N]/(overall.connections[N ==
  1, .N] + overall.connections[N == 2, .N/2]) * 100, 1)
```

```
[1] 48.8
```

```
# Percentage of two way connections, Total Connections %
# = One Way % + Two Way %
round(overall.connections[N == 2, .N/2]/(overall.connections[N ==
1, .N] + overall.connections[N == 2, .N/2]) * 100, 1)
```

```
[1] 51.2
```

48.8% of all connections are one-way connections, and 51.2% of all connections are two-way connections.

Question 2: Recommending Connections

Which connections should we recommend to the user with id CLKcSSSC? One way is to find the unconnected users who are connected to users that user CLKcSSSC is also connected to. Create a table of all the users who satisfy all of the following criteria:

- have at least 30 connections in common with user CLKcSSSC's connections, and
- are not already connected with user CLKcSSSC.

The list should show the ids of the recommended users and the number of common connections they have with user CLKcSSSC. Order the list in decreasing order of mutual connections. Make sure not to include CLKcSSSC on the list of recommendations!

```
# Selected user's connection and users did not connect
# to the selected user
user.connected <- connections[id == the.id, ]
user.unconnected <- connections[connection.id.name != the.id,
]

merged <- merge(x = user.connected, y = user.unconnected,
  by = connection.id.name)
colnames(merged) <- c("connection_id", "user: CLKcSSSC",
  "users")
# Filter out users did not meet conditions
recommend.connections <- merged[, .N, by = "users"]
recommend.connections <- recommend.connections[N >= 30 &
  users != the.id, ]
setorderv(recommend.connections, cols = "N", order = -1)

datatable(recommend.connections)
```

Show entries

Search:

	users	N
1	KaBFI3mi	34
2	be4WGFRz	33
3	261Vpi2U	31
4	gfCgPhAM	31

	users	N
5	7KgwsKvw	31
6	YYEIHrg	31
7	NuNJRLuy	30
8	E2HSTCj6	30
9	L7gdqlhN	30
10	8XqINMTF	30

Showing 1 to 10 of 12 entries

Previous

1

2

Next

Question 3: Influential Connections

In social networks, some users are considered **influential**. They tend to have more connections, and their content can be widely viewed and shared. For our purposes, we will define the **influential users** as those who:

- Have at least 200 photos, and
- Have at least 150 connections.

Among all users (both influential and not so influential), how many users are connected to at least 250 **influential** users?

```
# Merge Profiles and Connections
profiles_connections <- merge(x = profiles, y = connections,
  by = id.name)
# Calculate followers
num_followers <- connections[, .(Num_Followers = .N), by = connection.id.name]
setnames(x = num_followers, old = "connection_id", new = "id")
# Filter with photo and number of followers conditions
influential_users <- merge(x = profiles, y = num_followers,
  by = id.name)
influential_users <- influential_users[num_photos >= 200 &
  Num_Followers >= 150]
# Users connect to influential users
desired_users <- connections[, get(connection.id.name) %in%
  influential_users[, get(id.name)]]
# Bind datatable and filter with 250 influential user
# condition
rbined_dat <- cbind(connections, desired_users)
connected_influential <- rbined_dat[desired_users == "TRUE",
  ]
connected_influential <- connected_influential[, .N, by = id.name]
connected_influential_250 <- connected_influential[N >=
  250, ]
connected_influential_250[, .N]
```

[1] 2804

Question 4: Early Utilizers

Starting from the time when the account for each user was created, what percentage of all users logged in at least 35 times during the first 7? Round your answer to 1 decimal point, e.g. 84.2%.

Hints: Within the **lubridate** library, you can use the function **days** to add a specified number of days to the registration times. The first week ends before (less than) the user's first registration time plus 7 days. The registration that occurred when the account was created counts toward the overall total for this period.

```
# Merge Profile and Registration
profiles_registrations <- merge(x = profiles, y = registrations,
  by = id.name)
# Find the minimum time and minimum time + 7
profiles_registrations[, `:=`(first.registration.time, min(registration.time)),
  by = id.name]
profiles_registrations[, `:=`(first.7.days, first.registration.time +
  days(x = 7)), by = id.name]
# Filter condition
all_utilizers <- profiles_registrations[registration.time <=
  first.7.days, .N, by = id.name]
early_utilizers <- all_utilizers[N >= 35, ]
# The answer varies with where you put *100
round(early_utilizers[, .N]/profiles[, .N] * 100, 1)
```

```
[1] 29.8
```

```
round(early_utilizers[, .N] * 100/profiles[, .N], 1)
```

```
[1] 29.9
```

Question 5: Imbalanced Connections

What percentage of users have at least 100 more followers than the number of users that they are following? Round the answer to 1 decimal place, e.g. 84.2%.

```
# Find connections and followers
num_following <- connections[, .(Num_Following = .N), by = id.name]
num_followers <- connections[, .(Num_Followers = .N), by = connection.id.name]
setnames(x = num_followers, old = "connection_id", new = "id")
# Merge tables
connections_counts <- merge(x = num_following, y = num_followers,
  by = id.name)
# Calculate imbalanced connections
imbalanced_connections <- connections_counts[, .(Followers_Minus_Following = Num_Followers -
  Num_Following), by = id.name]
imbalanced_connections_100 <- imbalanced_connections[Followers_Minus_Following >=
  100, ]

round(imbalanced_connections_100[, .N]/profiles[, .N] *
  100, 1)
```

```
[1] 32.8
```

Question 6: Active Users

What percentage of unique users in the sample were active (with at least 1 registration) between 00:00:00 of January 1st, 2017 and 23:59:59 on January 7th, 2017? Round the percentage to 1 decimal place, e.g. 84.2%

Hint: For any given date in character format (e.g. "1999-07-01"), you can calculate a date in the future with the **as.Date** function: `as.Date("1999-07-01") + 3` would result in "1999-07-04".

```
# Find active users
active_users <- profiles_registrations[registration.time >=
  as.Date("2017-01-01 00:00:00") & registration.time <=
  as.Date("2017-01-07 23:59:59"), ]

active_user_list <- active_users[, .N, by = id.name]
round(active_user_list[, .N]/profiles[, .N] * 100, 1)
```

```
[1] 8.6
```

Question 7: Burning the Midnight Oil

Across all days, what percentage of all registrations occur between the hours of 00:00:00 and 05:59:59, inclusive of both endpoints? Round your answer to 1 decimal place, e.g. 84.2%. **Hint:** Use the `hour()` function to classify the time of day.

```
# all.registrations <-
# hour(profiles.plus.registrations$first.registration.time)
midnight_registrations <- profiles_registrations[hour(registration.time) >=
  0 & hour(registration.time) < 6, ]

round(midnight_registrations[, .N]/profiles_registrations[,
  .N] * 100, 1)
```

```
[1] 25
```

Question 8: Retention Rates

What percentage of users were retained at 183 days (half a year)? To answer this question, we will use a 7 day window. Any user who had at least one registration in the period of time that was at least 183 days and less than 190 days from their first registration would be considered retained. Round your answer to 1 decimal place, e.g. 84.2%.

Note: The evaluation window would begin at exactly 183 days after the first registration. This period lasts for 7 days. This window would include the left end-point but not the right end-point. The registration times are listed in the data set rounded to the nearest second. If the user had at least 1 registration during this window, the user would be considered retained at 183 days (approximately 6 months).

Hint: You may use the **days()** function to add time to a user's initial registration time.

```
# Calculate retention period
retention_dat <- merge(x = profiles, y = registrations,
  by = id.name)
retention_dat[, `:=`(first_registration_time, min(registration.time)),
  by = id.name]
retention_dat[, `:=`(starting_period, first_registration_time +
  days(x = 183)), by = id.name]
retention_dat[, `:=`(ending_period, first_registration_time +
  days(x = 190)), by = id.name]
# Filter out users
retained_users <- retention_dat[registration.time >= starting_period &
  registration.time < ending_period, .N, by = id.name]

round(retained_users[, .N]/profiles[, .N] * 100, 1)
```

```
[1] 30.2
```

Question 9: False Positive Rates

In the previous question, we estimated the rate of retention at 6 months using a 7-day window for evaluation. What is the rate of false positives for the 7-day window? In other words, what percentage of users who were considered not retained at 6 months using a 7-day window later had a registration? Round the results to 2 decimal places, e.g. 84.23%.

```
# User ID considered retained
users_retained <- retention_dat[, get(id.name) %in% retained_users[,
  get(id.name)]]
# Users who were not considered retained but had a late
# registration
retention_rbind <- cbind(retention_dat, users_retained)
false_positive_users <- retention_rbind[users_retained ==
  "FALSE" & registration.time > ending_period, .N, by = id.name]

round(false_positive_users[, .N]/(profiles[, .N] - retained_users[,
  .N]) * 100, 2)
```

```
[1] 0.04
```

Question 10: Modeling Retention

Build a logistic regression model for retention at 6 months. Classify users as retained at 6 months if they have any account registrations at times at least 183 days after their account was created. Include the following variables:

- density
- age_group
- gender
- num_photos (categories: 0-24, 25-49, 50-99, 100-249, 250-499, 500+) (current status)
- average daily registrations in the first week. (To simplify matters, let this be the total number of registrations in the first week divided by 7, regardless of whether the user's retention truly lasted 7 days or not.)

- number of connections the user currently has
- number of users currently connected to this user

Display the odds ratios, confidence intervals for the odds ratios, and p-values for the coefficients, rounded to 3 digits. Then briefly comment on the results.

```
# Add retained variable
retained_users10 <- retention_dat[registration.time >= starting_period,
  .N, by = id.name]
model_dat <- profiles[, `:=`(Retained, get(id.name) %in%
  retained_users10[, get(id.name)])]
# Cut number of photos
cuts.photo <- c(25, 50, 100, 250, 500, 1500)
model_dat[, `:=`(Num_photos, cut2(x = num_photos, cuts = cuts.photo))]
# Add number of followers
model_dat <- merge(x = model_dat, y = num_followers, by = id.name)
# Add number of connections
model_dat <- merge(x = model_dat, y = num_following, by = id.name)
# Remove unnecessary columns
model_dat[, `:=`(c("has_profile_photo", "num_photos", "date_created"),
  NULL)]
# Add average daily registration
model_date <- registrations
model_date[, `:=`(first_regis, min(registration.time)),
  by = id.name]
model_date[, `:=`(first_week, min(registration.time) + days(x = 7)),
  by = id.name]
average_date <- model_date[registration.time >= first_regis &
  registration.time <= first_week, .(Average_Registration = .N/7),
  by = id.name]
model_dat <- merge(x = model_dat, y = average_date, by = id.name)
# Correct str
model_dat$density <- as.factor(model_dat$density)
model_dat$age_group <- as.factor(model_dat$age_group)
model_dat$gender <- as.factor(model_dat$gender)
# Model
glm_mod <- glm(Retained ~ ., family = binomial, data = model_dat[,
  -1])
result <- round(cbind(exp(cbind(coef(glm_mod), confint(glm_mod))),
  coef(summary(glm_mod))[, 4]), 3)
colnames(result) <- c("Odds Ratio", "2.5%", "97.5%", "P-value")
datatable(result)
```

Show entries

Search:

	Odds Ratio	2.5%	97.5%	P-value
(Intercept)	0.012	0.008	0.018	0
densitySuburban	0.551	0.453	0.671	0
densityUrban	0.272	0.223	0.332	0
age_group25-34	0.864	0.669	1.117	0.265
age_group35-44	0.692	0.544	0.881	0.003
age_group45-54	0.465	0.368	0.589	0

	Odds Ratio	2.5%	97.5%	P-value
age_group55-64	0.304	0.232	0.399	0
age_group65+	0.628	0.467	0.844	0.002
genderM	0.716	0.618	0.83	0
Num_photos[25, 50)	0.978	0.724	1.321	0.883

Showing 1 to 10 of 17 entries

From the results, we could see that density variables, age group of 45-54 and age group of 55-64, genderm, number of following are and average registrations are very significant while all the photo vaariables did little contribution to the data. Average registration has the highest odds ratio among other variables.