```
library(gridExtra)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine() masks gridExtra::combine()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(rmarkdown)
library(mosaic)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggformula
```

```
## Loading required package: ggstance
```

```
##
## Attaching package: 'ggstance'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following objects are masked from 'package:dplyr':
##
```

```
##     count, do, tally
```

```
## The following object is masked from 'package:purrr':
##
##     cross
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(broom)
library(knitr)
library(foreign)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(tableone)
library(readstata13)
library(AER)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following objects are masked from 'package:mosaic':
##
##     deltaMethod, logit
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: survival
```

```
library(plm)
```

```
##
## Attaching package: 'plm'
```

```
## The following object is masked from 'package:mosaic':
##
##     r.squared
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```
data <- read.csv("training_dataset150000.csv", header = T)
data$yelping_since <- as.Date(data$yelping_since)
data$years_yelping <- (Sys.Date() - data$yelping_since) / 365

data$total_compliments_received <- data$num_hot_compliment + data$num_cool_compliment + data$num_cute_c
  data$num_funny_compliment + data$num_list_compliment + data$num_note_compliment + data$num_photos_com
  data$num_profile_compliment + data$num_plain_compliment + data$num_writer_compliment + data$num_more_
data$total_compliments_given <- data$useful + data$funny + data$cool

everything <- glm(ever_elite ~ score + review_count + fans + average_stars + years_yelping + num_hot_co
    num_profile_compliment + num_cute_compliment + num_list_compliment + num_note_compliment + num_plai
    num_cool_compliment + num_funny_compliment + num_writer_compliment + num_photos_compliment + funny
    family = "binomial", data = data)
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
kable(tidy(everything))
```

term	estimate	std.error	statistic	p.value
(Intercept)	-4.272995e+15	1.558503e+06	-2741730612	0
score	-8.176757e+12	1.167760e+04	-700208637	0
review_count	1.147776e+13	4.347297e+03	2640206255	0
fans	2.777124e+10	6.542055e+01	424503272	0
average_stars	8.412324e+14	1.512903e+05	5560386322	0
years_yelping	-6.337189e+12	6.998165e+04	-90555013	0
num_hot_compliment	3.298523e+12	1.194928e+04	276043728	0
num_more_compliment	1.181449e+14	9.897912e+04	1193634270	0
num_profile_compliment	-5.243630e+13	8.002152e+04	-655277507	0
num_cute_compliment	-2.854204e+13	6.698357e+04	-426105016	0
num_list_compliment	3.337597e+13	1.352304e+05	246808219	0
num_note_compliment	-3.282697e+12	1.015279e+04	-323329460	0
num_plain_compliment	-8.187738e+12	6.384274e+03	-1282485306	0
num_cool_compliment	1.002870e+13	9.111463e+03	1100668736	0
num_writer_compliment	8.653894e+12	2.196694e+04	393950831	0
num_photos_compliment	-4.298204e+12	5.095313e+03	-843560406	0
funny	-1.755651e+12	5.356193e+03	-327779618	0

term	estimate	std.error	statistic	p.value
useful	-3.360055e+12	3.470789e+03	-968095584	0
cool	3.100749e+13	2.967000e+04	1045079072	0
friends	1.512660e+12	4.196225e+03	360481179	0

```
smaller <- glm(ever_elite ~ score + review_count + fans + average_stars + years_yelping + total_complime
    total_compliments_received + friends, family = "binomial", data = data)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
kable(tidy(smaller))
```

term	estimate	std.error	statistic	p.value
(Intercept)	-4.1184859	0.1688817	-24.386805	0.0000000
score	-0.0185642	0.0010639	-17.449798	0.0000000
review_count	0.0236431	0.0004504	52.497742	0.0000000
fans	0.0000776	0.0000044	17.719083	0.0000000
average_stars	0.4126102	0.0264123	15.621892	0.0000000
years_yelping	-0.0171222	0.0076540	-2.237032	0.0252842
total_compliments_given	-0.0019019	0.0003567	-5.332629	0.0000001
total_compliments_received	0.0005216	0.0003997	1.304855	0.1919421
friends	0.0056643	0.0004384	12.919983	0.0000000

```
subset <- glm(ever_elite ~ score+ review_count + fans + average_stars + years_yelping + total_compliment
                          friends, family = "binomial", data = data)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
kable(tidy(subset))
```

term	estimate	std.error	statistic	p.value
(Intercept)	-4.1188368	0.1689018	-24.385981	0.0000000
score	-0.0185899	0.0010637	-17.476071	0.0000000
review_count	0.0235893	0.0004486	52.581873	0.0000000
fans	0.0000778	0.0000044	17.788797	0.0000000
average_stars	0.4129042	0.0264146	15.631668	0.0000000
years_yelping	-0.0166854	0.0076457	-2.182329	0.0290852
total_compliments_given	-0.0016636	0.0003094	-5.376560	0.0000001
friends	0.0056131	0.0004372	12.838952	0.0000000

```
# VIF used, none over 10
```