Store Random Effect Model

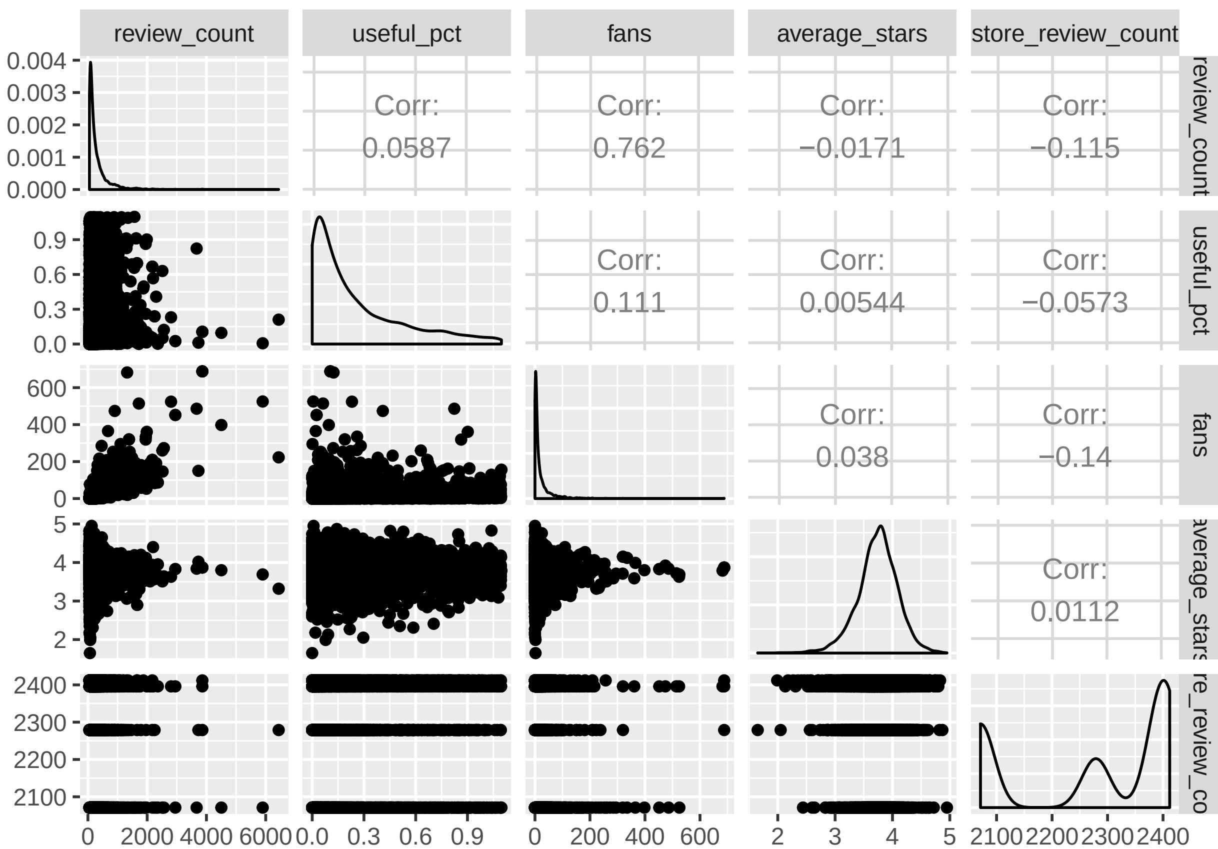
Jinghan Cui

12/1/2017

2. Store Random Effect Models

2.1 Variable Selection

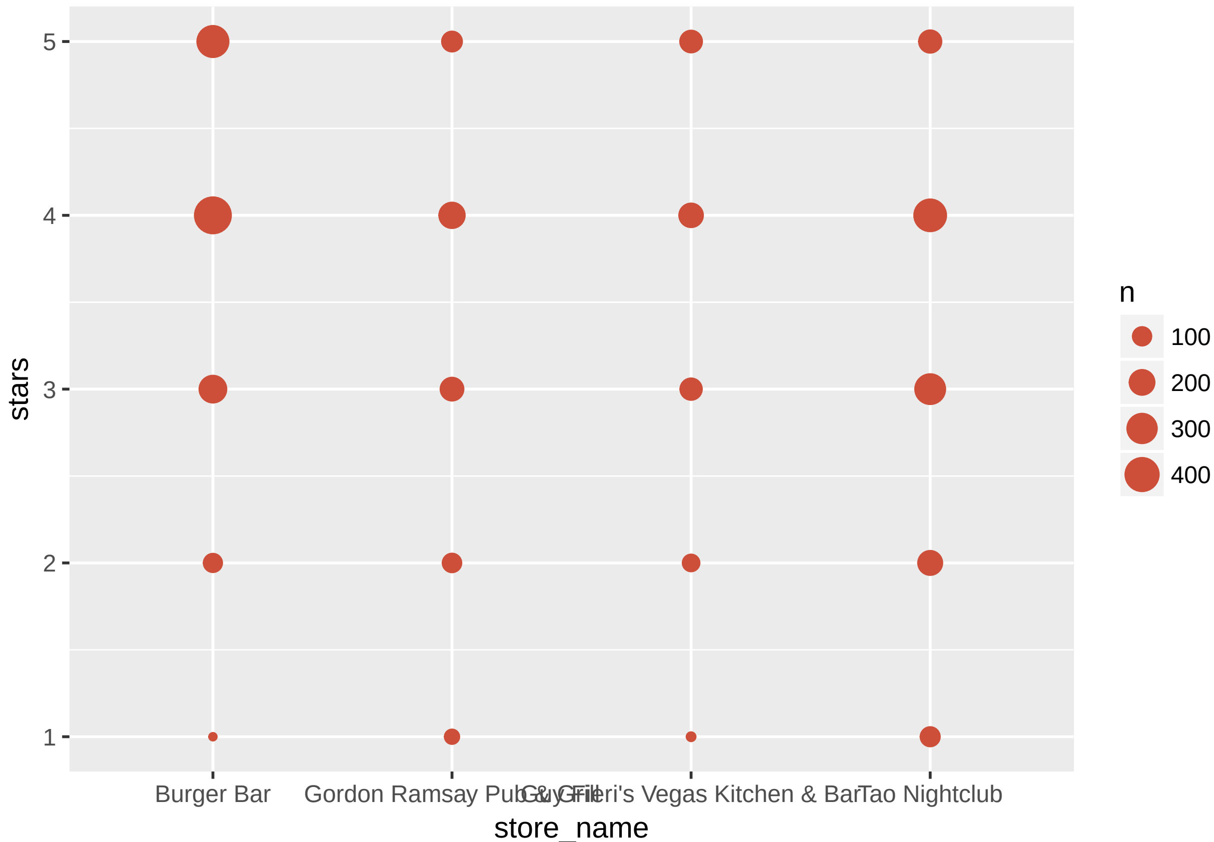
```r
sample_store_bars <- read.csv("sample_store_bars.csv")
# check correlations within predictor variables
ggpairs(sample_store_bars[, c("review_count","useful_pct","fans","average_stars", "store_review_count")]
```



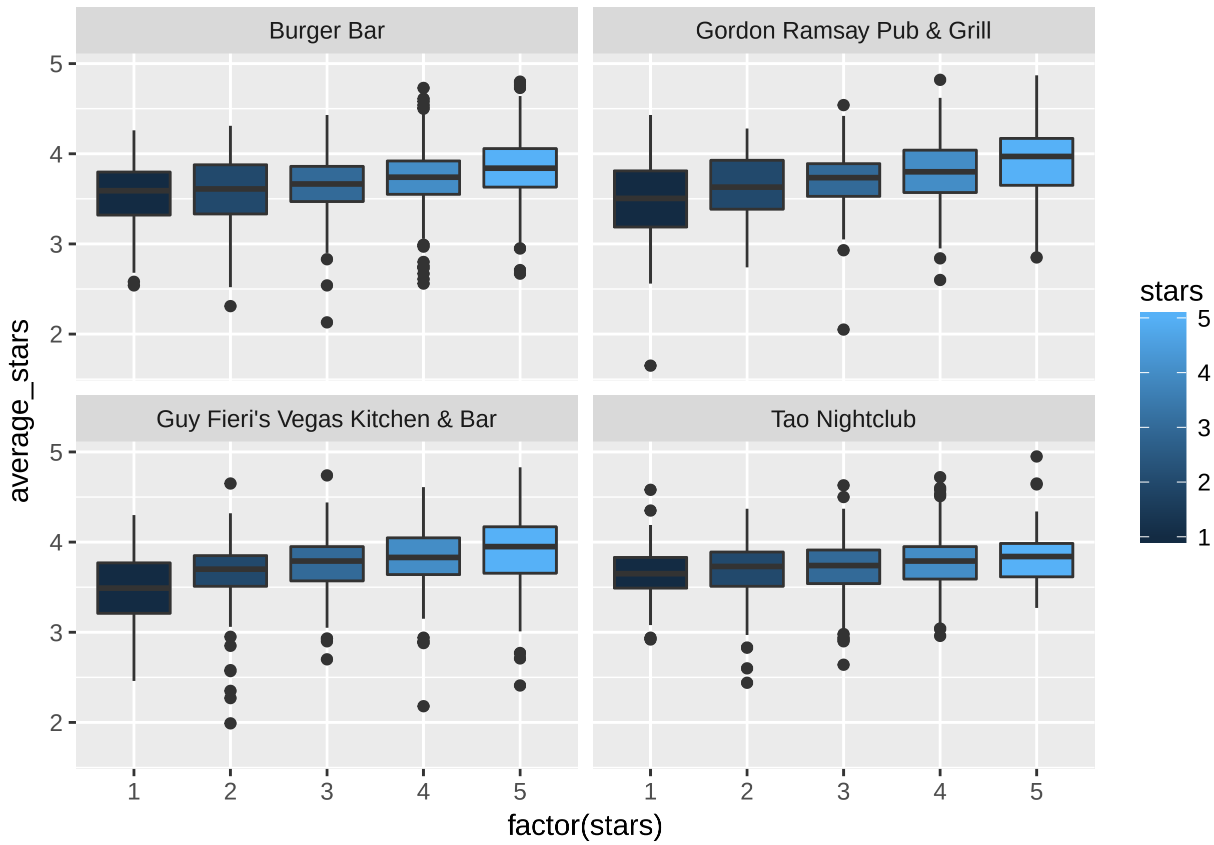

```r
# we may want to drop fans as it is highly correlated with review_count
```

2.2 EDA

```r
# the sampled restaurants have same store_stars while the plot shows a different distribution of reveiw
ggplot(sample_store_bars, aes(x = store_name, y = stars)) +
  stat_sum(aes(size = ..n.., group = 1), color = "tomato3")
```




```r
# There is a group=level of random effect among restaurants
ggplot(sample_store_bars, aes(x=factor(stars), y = average_stars, fill=stars)) +
  geom_boxplot() +
  facet_wrap(~store_name)
```



2.3 Linear Model

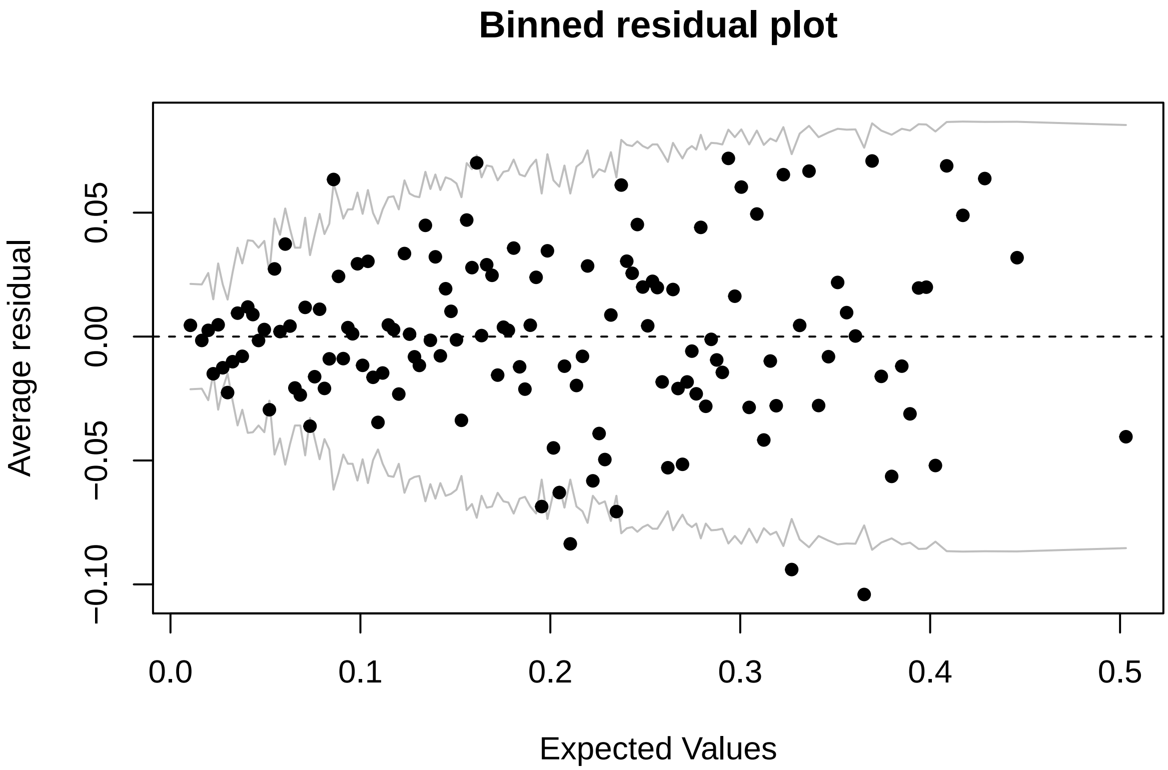
```r
# first fit a simple linear model
fit1 <- lm(stars~review_count_c + useful_pct + average_stars + store_stars + store_review_count_c + sto
display(fit1)
```

```
## lm(formula = stars ~ review_count_c + useful_pct + average_stars +
##     store_stars + store_review_count_c + store_name, data = sample_store_bars)
##                                      coef.est coef.se
## (Intercept)                          -1.45     0.28
## review_count_c                        0.01     0.02
## useful_pct                           -0.01     0.07
## average_stars                         0.78     0.05
## store_stars                           0.63     0.10
## store_review_count_c                 -0.05     0.18
## store_nameGordon Ramsay Pub & Grill -0.19     0.05
## ---
## n = 3557, k = 7
## residual sd = 1.10, R-Squared = 0.10
```

```r
# the linear model is not applicable to factor response
```

2.4 Multinomial logistic regression

```
# then try a logistic model
fit.multi <- vglm(ordered(stars) ~ review_count_c + useful_pct + average_stars + store_stars + store_rev
binnedplot(fitted(fit.multi,type="response"), resid(fit.multi, type="response"))
```




```
summary(fit.multi)
```

```
##
## Call:
## vglm(formula = ordered(stars) ~ review_count_c + useful_pct +
##     average_stars + store_stars + store_review_count_c, family = cumulative,
##     data = sample_store_bars)
##
##
## Pearson residuals:
##                   Min      1Q  Median      3Q    Max
## logit(P[Y<=1]) -1.397 -0.2128 -0.1435 -0.1036 14.845
## logit(P[Y<=2]) -1.955 -0.4217 -0.2346 -0.1511  6.097
## logit(P[Y<=3]) -2.828 -0.8527 -0.3057  0.6940  2.706
## logit(P[Y<=4]) -7.165  0.1401  0.2893  0.6787  1.453
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept):1           6.922752   0.961479   7.200 6.02e-13 ***
## (Intercept):2           6.741039   0.636552  10.590  < 2e-16 ***
## (Intercept):3           7.550981   0.542885  13.909  < 2e-16 ***
## (Intercept):4           9.754805   0.688867  14.161  < 2e-16 ***
## review_count_c:1       -0.379460   0.113018  -3.358 0.000786 ***
```

```
## review_count_c:2       -0.176477   0.054966  -3.211 0.001324 **
## review_count_c:3        0.002632   0.036293   0.073 0.942184
## review_count_c:4        0.203165   0.058520   3.472 0.000517 ***
## useful_pct:1            0.183097   0.251011   0.729 0.465733
## useful_pct:2            0.231817   0.159106   1.457 0.145118
## useful_pct:3           -0.026199   0.131195  -0.200 0.841717
## useful_pct:4           -0.146526   0.155985  -0.939 0.347545
## average_stars:1        -1.428095   0.175801  -8.123 4.53e-16 ***
## average_stars:2        -1.327946   0.120683 -11.004  < 2e-16 ***
## average_stars:3        -1.253683   0.103778 -12.080  < 2e-16 ***
## average_stars:4        -1.282887   0.126482 -10.143  < 2e-16 ***
## store_stars:1          -1.422883   0.433315      NA       NA
## store_stars:2          -1.455448   0.253982  -5.731 1.00e-08 ***
## store_stars:3          -1.317580   0.197089  -6.685 2.31e-11 ***
## store_stars:4          -0.667522   0.220400  -3.029 0.002456 **
## store_review_count_c:1  0.194802   0.659072   0.296 0.767559
## store_review_count_c:2  0.707569   0.401722   1.761 0.078181 .
## store_review_count_c:3  0.603777   0.328978   1.835 0.066460 .
## store_review_count_c:4 -0.473470   0.389816  -1.215 0.224520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of linear predictors:  4
##
## Names of linear predictors:
## logit(P[Y<=1]), logit(P[Y<=2]), logit(P[Y<=3]), logit(P[Y<=4])
##
## Residual deviance: 10148.42 on 14204 degrees of freedom
##
## Log-likelihood: -5074.21 on 14204 degrees of freedom
##
## Number of iterations: 5
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):4', 'average_stars:1', 'average_stars:2', 'store_stars:1', 'store_stars:2'
##
## Exponentiated coefficients:
##       review_count_c:1       review_count_c:2       review_count_c:3
##              0.6842306              0.8382179              1.0026356
##       review_count_c:4           useful_pct:1           useful_pct:2
##              1.2252751              1.2009311              1.2608885
##           useful_pct:3           useful_pct:4        average_stars:1
##              0.9741408              0.8637031              0.2397653
##        average_stars:2        average_stars:3        average_stars:4
##              0.2650209              0.2854516              0.2772359
##          store_stars:1          store_stars:2          store_stars:3
##              0.2410182              0.2332959              0.2677826
##          store_stars:4 store_review_count_c:1 store_review_count_c:2
##              0.5129782              1.2150700              2.0290517
## store_review_count_c:3 store_review_count_c:4
##              1.8290141              0.6228374
```

```
# so far the residuals look good but I still try the other models to see whether we can find a better o
```

2.5 Multilevel Model

```r
# not including store_review_count because we sample the data by store review count
fit2 <- glmer(stars ~ review_count_c + useful_pct + average_stars + store_stars + store_review_count_c +
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge with max|grad| = 0.00151669 (tol =
## 0.001, component 1)
```

```r
summary(fit2)
```

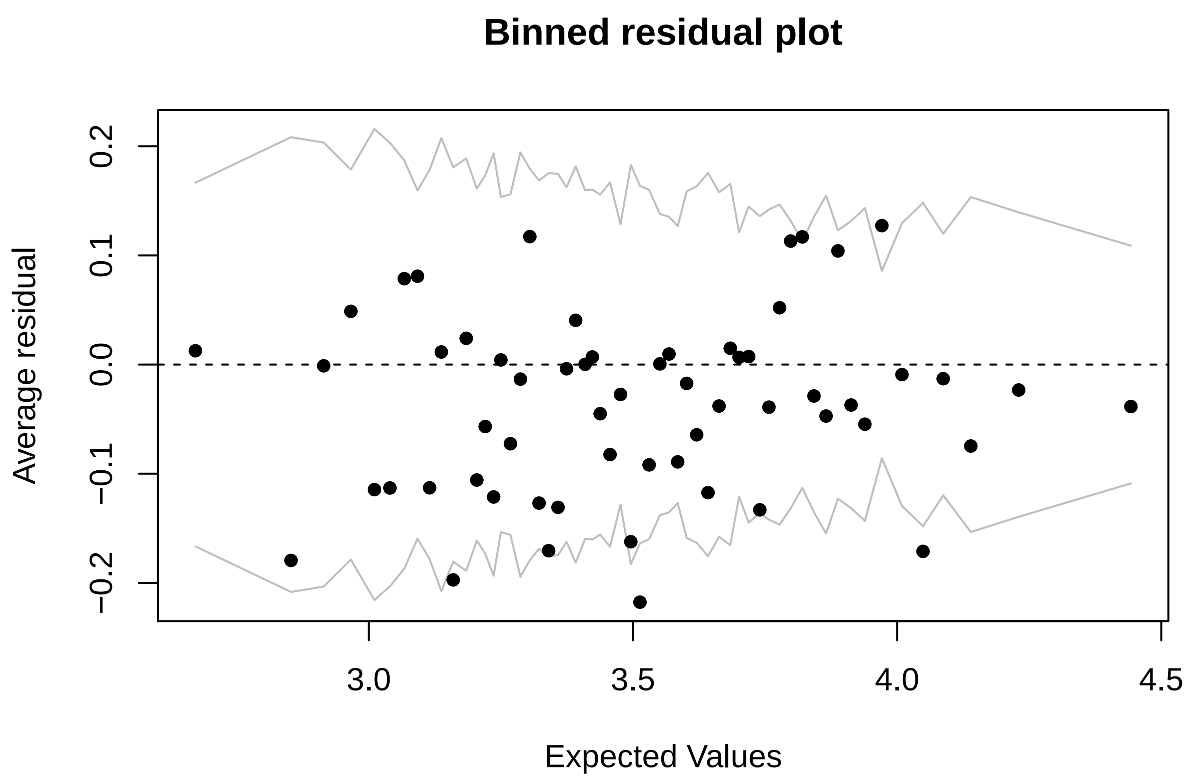
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula:
## stars ~ review_count_c + useful_pct + average_stars + store_stars +
##     store_review_count_c + (1 | store_name)
##    Data: sample_store_bars
##
##      AIC      BIC   logLik deviance df.resid
##  12347.2  12390.4  -6166.6  12333.2     3550
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.58310 -0.36698  0.07056  0.43841  1.52006
##
## Random effects:
##  Groups     Name        Variance  Std.Dev.
##  store_name (Intercept) 1.212e-05 0.003482
## Number of obs: 3557, groups:  store_name, 4
##
## Fixed effects:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.187383   0.138427  -1.354 0.175845
## review_count_c        0.005105   0.009435   0.541 0.588480
## useful_pct           -0.002584   0.033721  -0.077 0.938929
## average_stars         0.224936   0.025777   8.726  < 2e-16 ***
## store_stars           0.181946   0.051694   3.520 0.000432 ***
## store_review_count_c -0.016624   0.088675  -0.187 0.851291
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) rvw_c_ usfl_p avrg_s str_st
## reviw_cnt_c -0.101
## useful_pct  -0.091 -0.052
## averag_strs -0.683  0.030 -0.005
## store_stars  0.088 -0.064 -0.004  0.049
## str_rvw_cn_ -0.390  0.104  0.025 -0.051 -0.896
## convergence code: 0
## Model failed to converge with max|grad| = 0.00151669 (tol = 0.001, component 1)
```

```r
print(fit2, corr = FALSE)
```

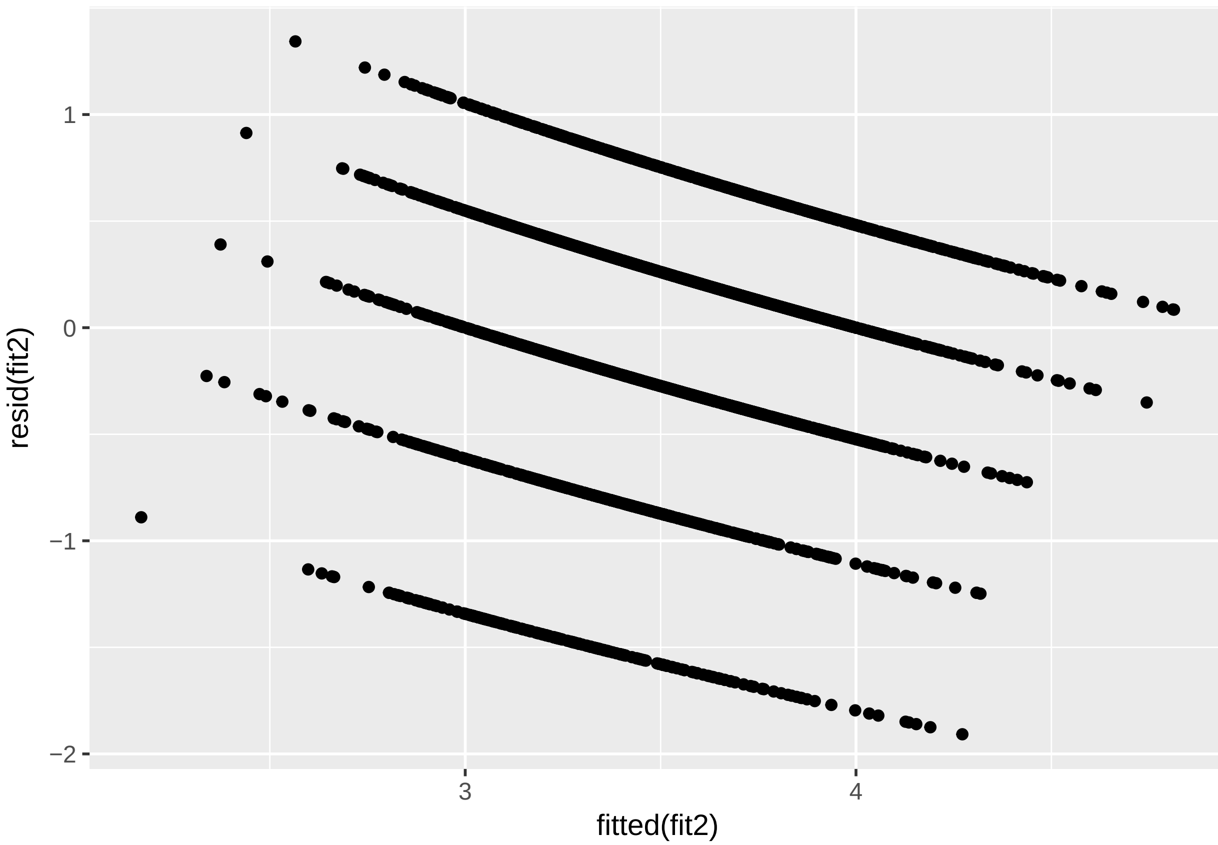
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
```

```
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula:
## stars ~ review_count_c + useful_pct + average_stars + store_stars +
##     store_review_count_c + (1 | store_name)
##    Data: sample_store_bars
##       AIC       BIC    logLik  deviance  df.resid
## 12347.183 12390.420 -6166.591 12333.183      3550
## Random effects:
##  Groups     Name        Std.Dev.
##  store_name (Intercept) 0.003482
## Number of obs: 3557, groups:  store_name, 4
## Fixed Effects:
##          (Intercept)        review_count_c            useful_pct
##            -0.187383              0.005105             -0.002584
##        average_stars           store_stars  store_review_count_c
##             0.224936              0.181946             -0.016624
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

```r
# binned residual plot
binnedplot(fitted(fit2), resid(fit2))
```




```r
# ggplot residual plot
ggplot(fit2, aes(x=fitted(fit2), y=resid(fit2))) +
  geom_point()
```




```
# most of the residuals are random distributed, it looks the multilevel model is better than the logist
se2 <- sqrt(diag(vcov(fit2)))
# table of estimates with 95% CI
(tab2 <- cbind(Est = fixef(fit2), LL = fixef(fit2) - 1.96 * se2, UL = fixef(fit2) + 1.96 * se2))
```

```
##                                Est          LL         UL
## (Intercept)           -0.187382536 -0.45869879 0.08393372
## review_count_c         0.005104877 -0.01338831 0.02359806
## useful_pct            -0.002583547 -0.06867646 0.06350936
## average_stars          0.224936112  0.17441310 0.27545912
## store_stars            0.181945527  0.08062481 0.28326624
## store_review_count_c  -0.016624036 -0.19042690 0.15717883
```

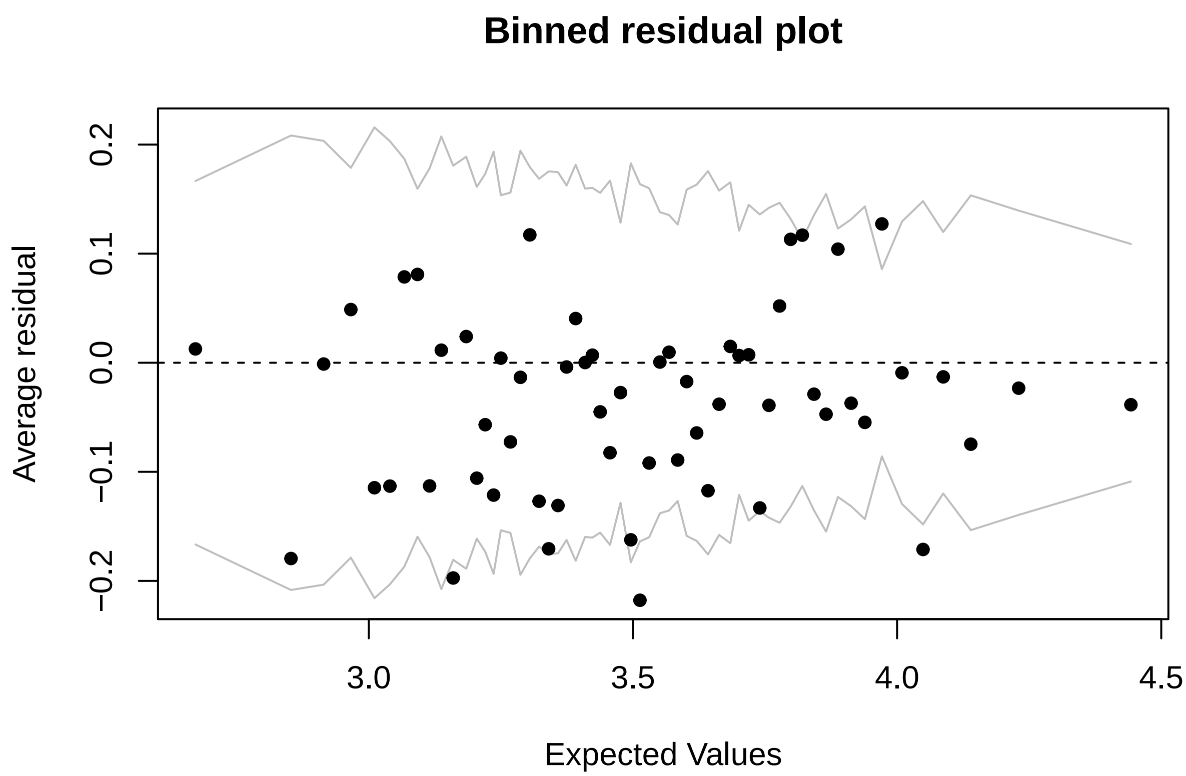
```
# useful_pct and Store_review_count have prediction intercal aross zero
```

```
# another way of doing multilevel model
fit3 <- glmer(stars ~ review_count_c + useful_pct + average_stars + store_stars + store_review_count_c +
summary(fit3)
```

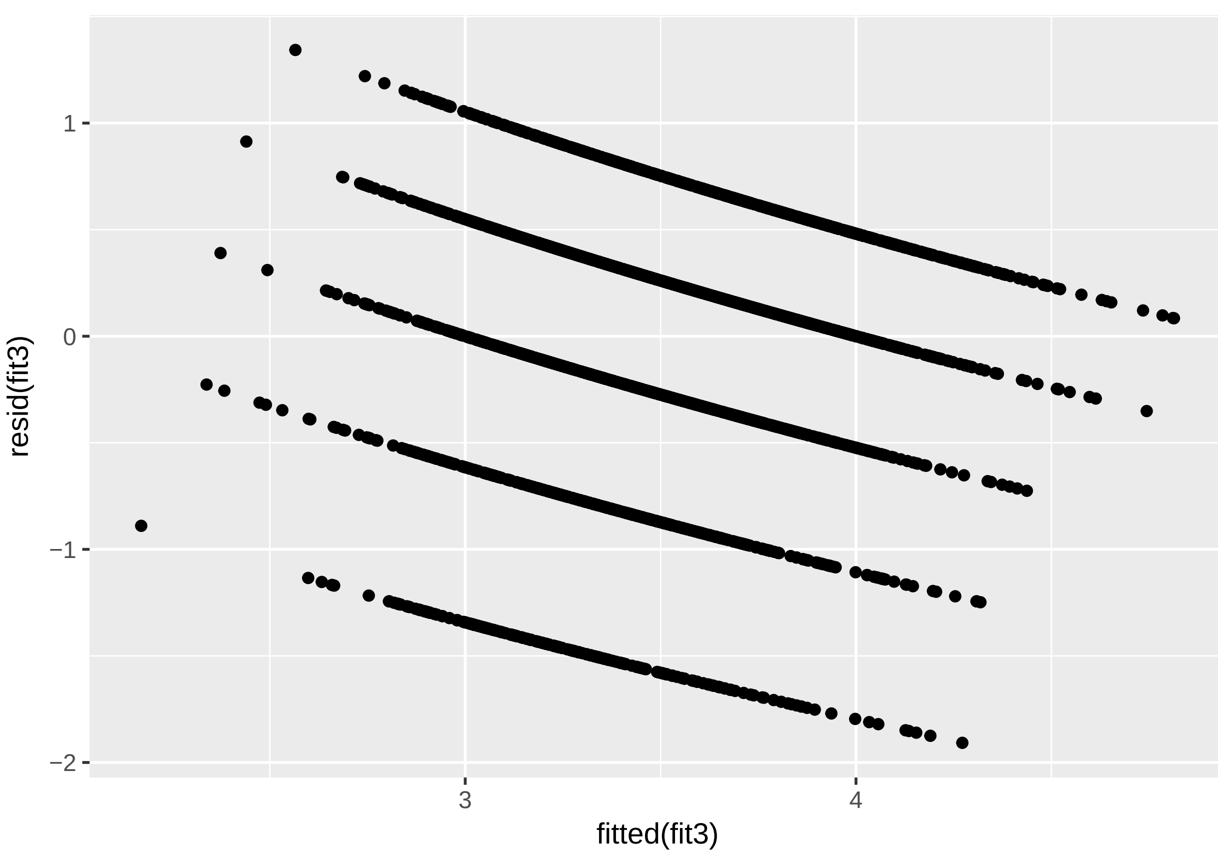
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula:
## stars ~ review_count_c + useful_pct + average_stars + store_stars +
##     store_review_count_c + (1 | store_name)
##    Data: sample_store_bars
## Control: glmerControl(optimizer = "bobyqa")
##
```

```
##      AIC      BIC   logLik deviance df.resid
##  12347.2  12390.4  -6166.6  12333.2     3550
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.58310 -0.36698  0.07056  0.43841  1.52006
##
## Random effects:
##  Groups     Name        Variance  Std.Dev.
##  store_name (Intercept) 1.214e-05 0.003484
## Number of obs: 3557, groups:  store_name, 4
##
## Fixed effects:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.187386   0.138433  -1.354 0.175859
## review_count_c       0.005105   0.009435   0.541 0.588451
## useful_pct          -0.002582   0.033721  -0.077 0.938968
## average_stars        0.224935   0.025777   8.726  < 2e-16 ***
## store_stars          0.181946   0.051694   3.520 0.000432 ***
## store_review_count_c -0.016622   0.088677  -0.187 0.851310
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) rvw_c_ usfl_p avrg_s str_st
## reviw_cnt_c -0.101
## useful_pct  -0.091 -0.052
## averag_strs -0.683  0.030 -0.005
## store_stars  0.088 -0.064 -0.004  0.049
## str_rvw_cn_ -0.390  0.104  0.025 -0.051 -0.896
```

```
binnedplot(fitted(fit3), resid(fit3))
```




```r
ggplot(fit3, aes(fitted(fit3), resid(fit3))) +
  geom_point()
```

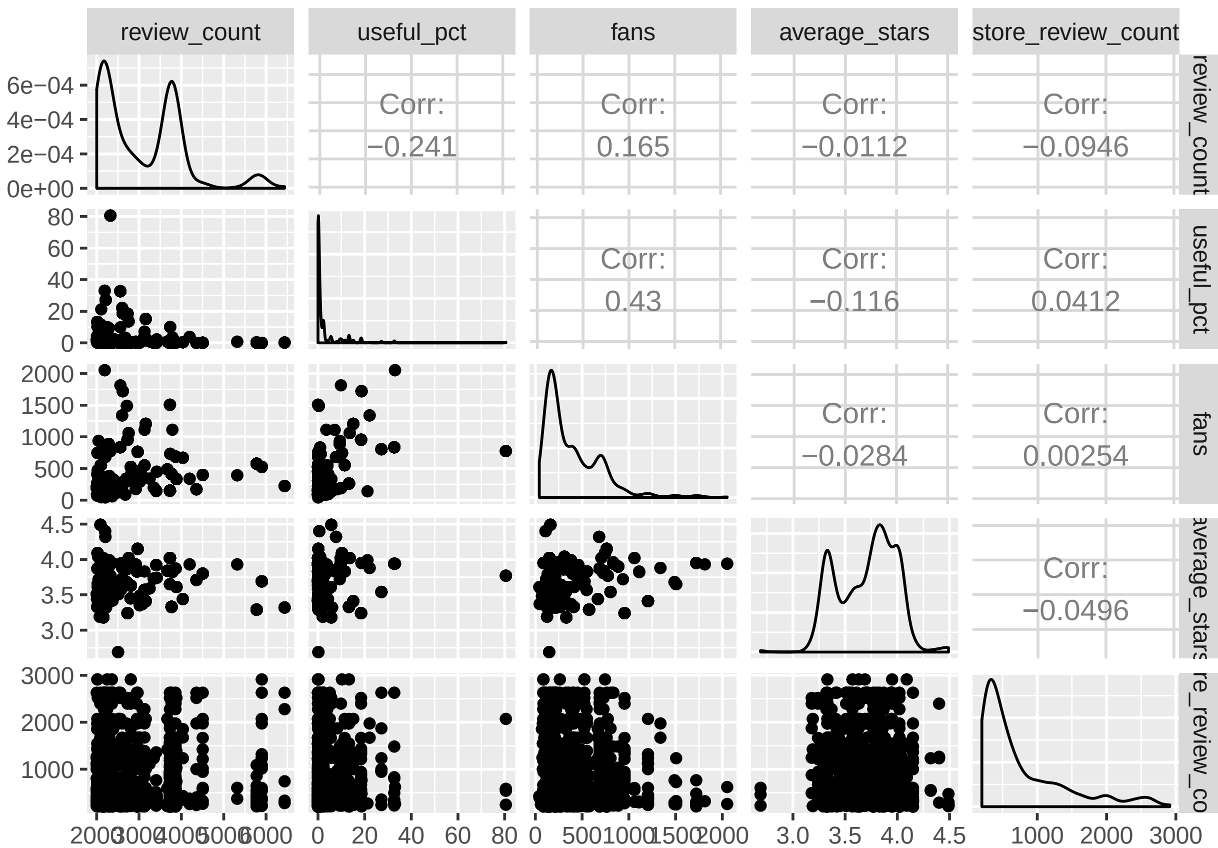


```
# still not showing a good fit
```

3. User as Random Effect

3.1 Variable Selection

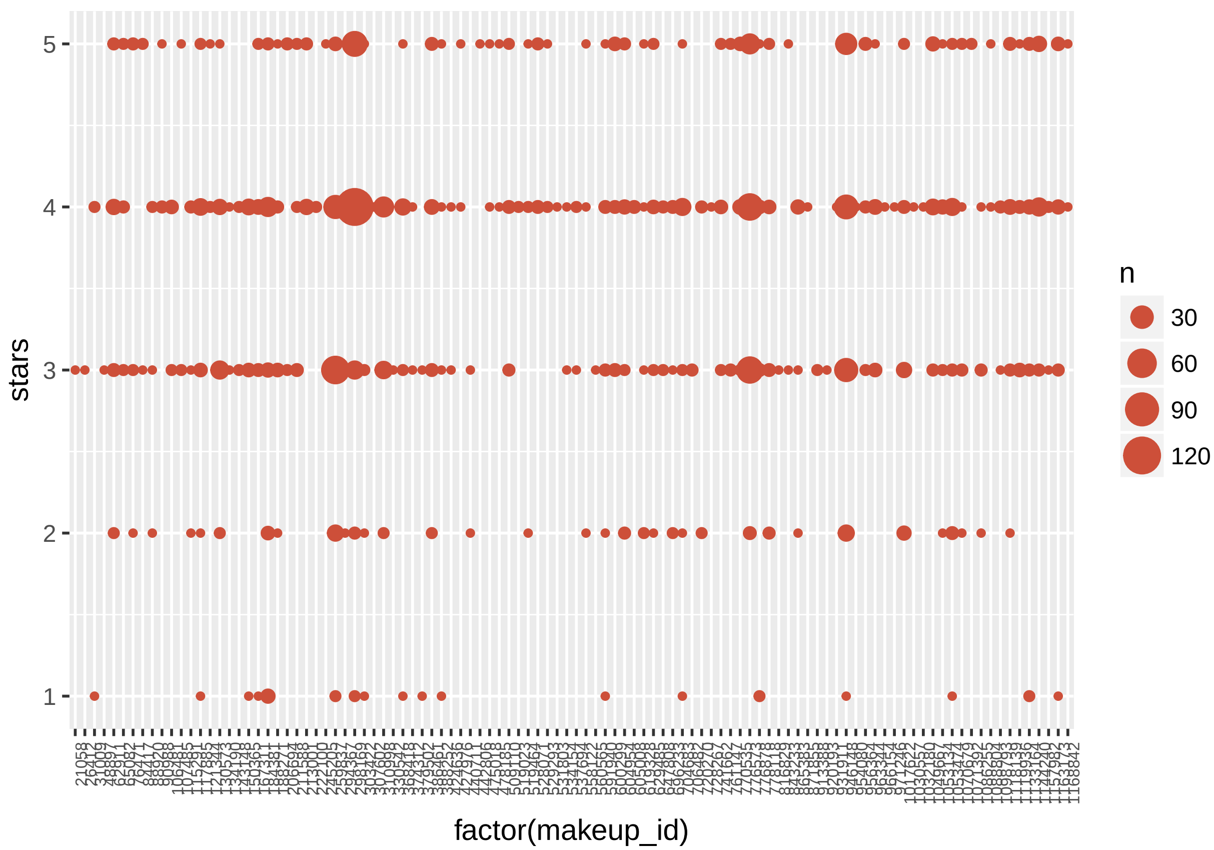
```
sample_user_bars <- read.csv("sample_user_bars.csv")
# check correlations within predictor variables
ggpairs(sample_user_bars[, c("review_count","useful_pct","fans","average_stars", "store_review_count")]
```




```
# we may want to drop fans as it is highly correlated with review_count, as well as useful_pct
```

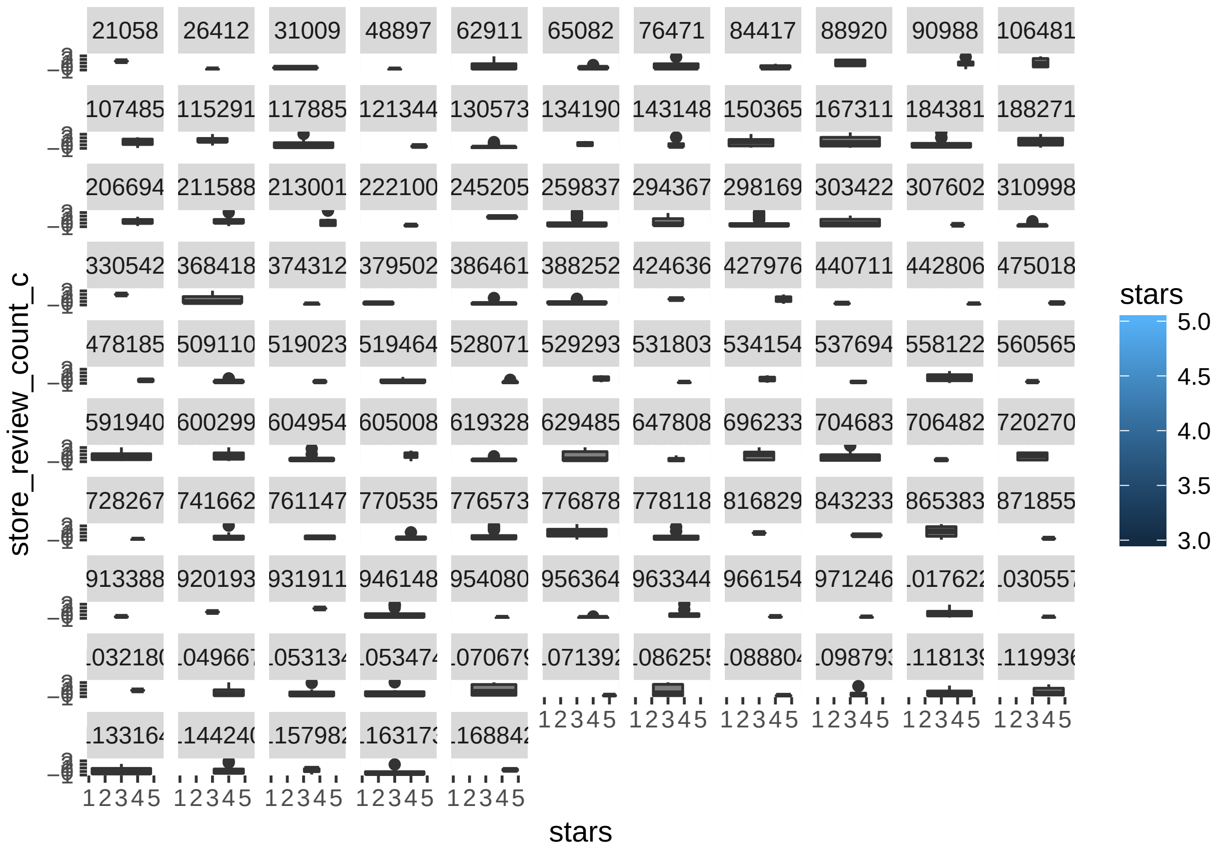
3.2 EDA

```
# most of the user give scores of 4 or 5 to bars
ggplot(sample_user_bars, aes(x = factor(makeup_id), y = stars)) +
  stat_sum(aes(size = ..n.., group = 1), color = "tomato3") +
  theme(axis.text.x=element_text(angle=90, hjust=1, size = 6))
```

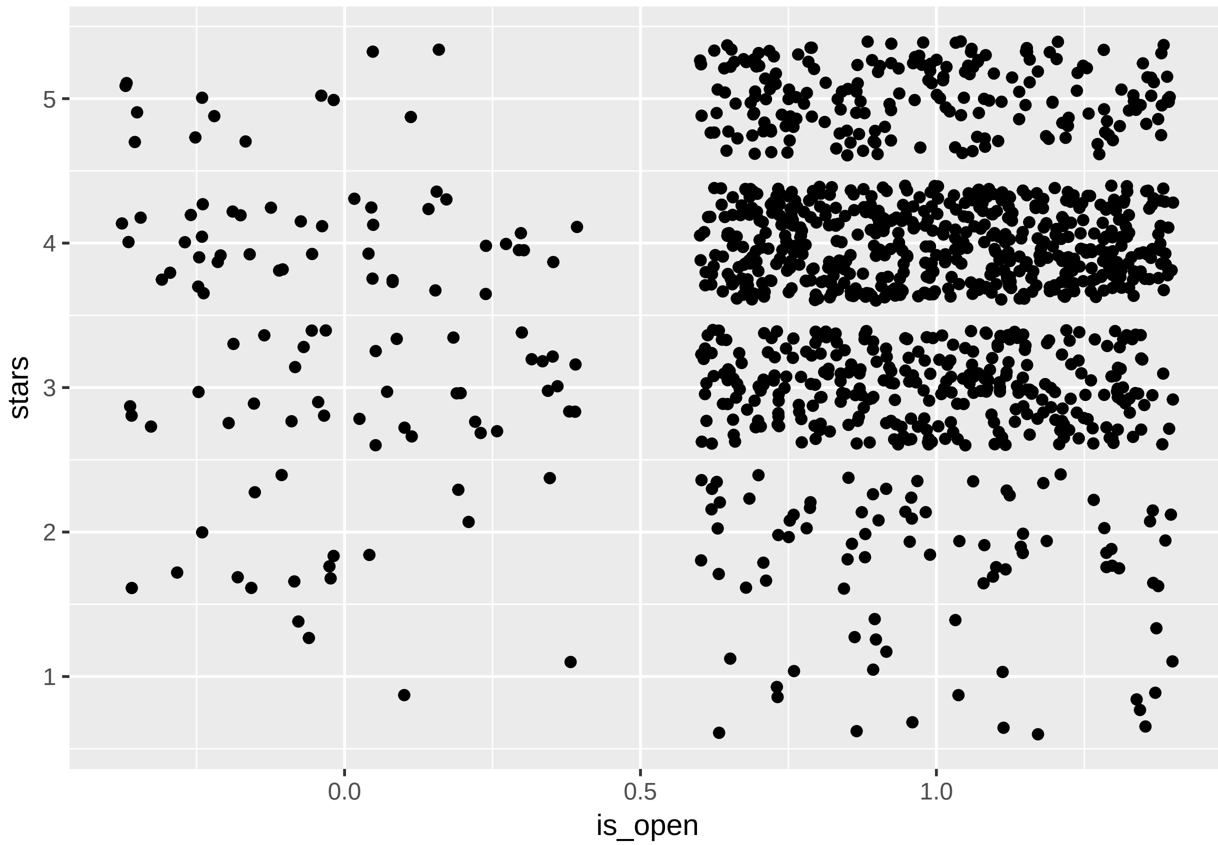



```r
# There is a individual-level of random effect among users
ggplot(sample_user_bars, aes(x=stars, y =store_review_count_c, fill = stars)) +
  geom_boxplot() +
  facet_wrap(~factor(makeup_id))
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```




```r
# 1 means the store is open, 0 means the store closed.
ggplot(sample_user_bars, aes(is_open, stars)) +
  geom_jitter()
```




```
# we would include is_open as a level predictor as well.
```

3.3 Multilevel Model

```
sample_user_bars$makeup_id <- factor(sample_user_bars$makeup_id)
fit <- glmer(factor(stars) ~ review_count_c + average_stars + store_review_count_c + (1|makeup_id) + (1
summary(fit)
```

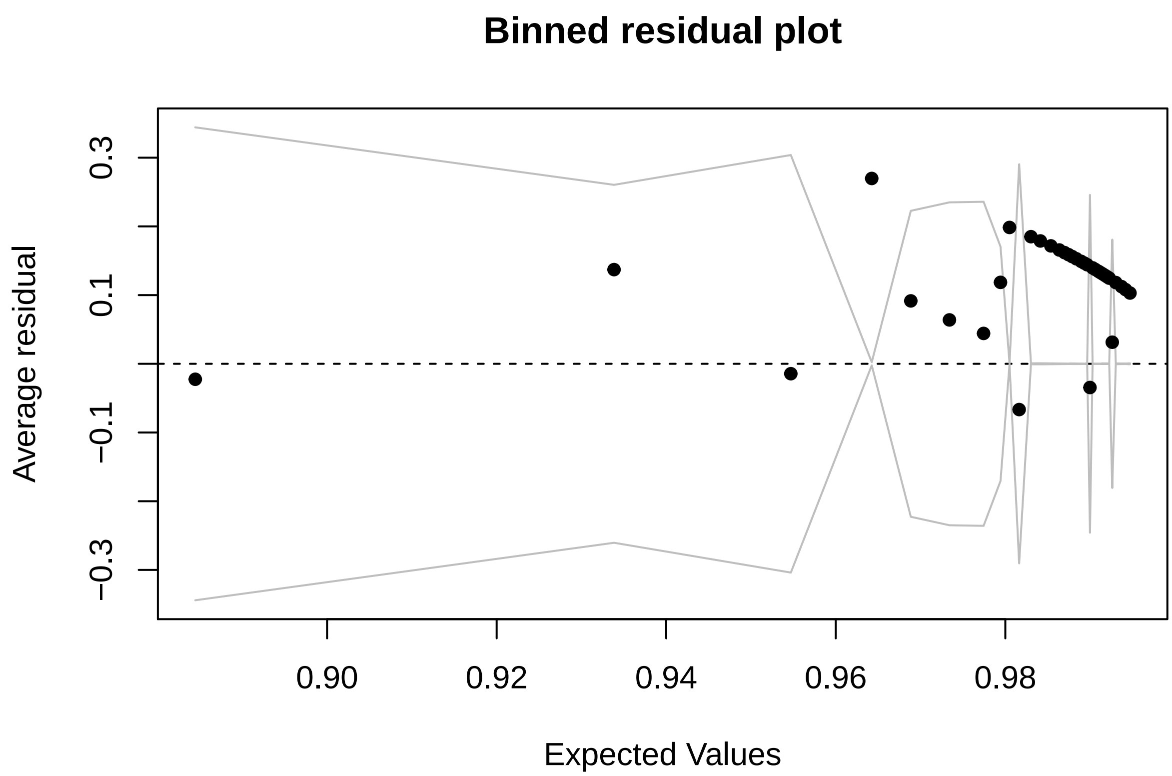
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## factor(stars) ~ review_count_c + average_stars + store_review_count_c +
##     (1 | makeup_id) + (1 | is_open)
##    Data: sample_user_bars
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##    255.8    286.5   -121.9    243.8     1227
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -11.5555   0.0920   0.1063   0.1370   0.5084
##
## Random effects:
##  Groups    Name        Variance Std.Dev.
```

```
##  makeup_id (Intercept) 0.8502   0.9221
##  is_open   (Intercept) 0.0000   0.0000
## Number of obs: 1233, groups:  makeup_id, 104; is_open, 2
##
## Fixed effects:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           1.98380    3.79223   0.523  0.60089
## review_count_c        0.07517    0.11488   0.654  0.51288
## average_stars         0.48316    0.98686   0.490  0.62442
## store_review_count_c -0.40710    0.15599  -2.610  0.00906 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) rvw_c_ avrg_s
## reviw_cnt_c -0.231
## averag_strs -0.975  0.033
## str_rvw_cn_ -0.038  0.019  0.011
```

```
print(fit, corr = FALSE)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## factor(stars) ~ review_count_c + average_stars + store_review_count_c +
##     (1 | makeup_id) + (1 | is_open)
##    Data: sample_user_bars
##       AIC       BIC    logLik  deviance  df.resid
##  255.8038  286.5071 -121.9019  243.8038      1227
## Random effects:
##  Groups    Name        Std.Dev.
##  makeup_id (Intercept) 0.9221
##  is_open   (Intercept) 0.0000
## Number of obs: 1233, groups:  makeup_id, 104; is_open, 2
## Fixed Effects:
##          (Intercept)        review_count_c         average_stars
##              1.98380               0.07517               0.48316
## store_review_count_c
##             -0.40710
```

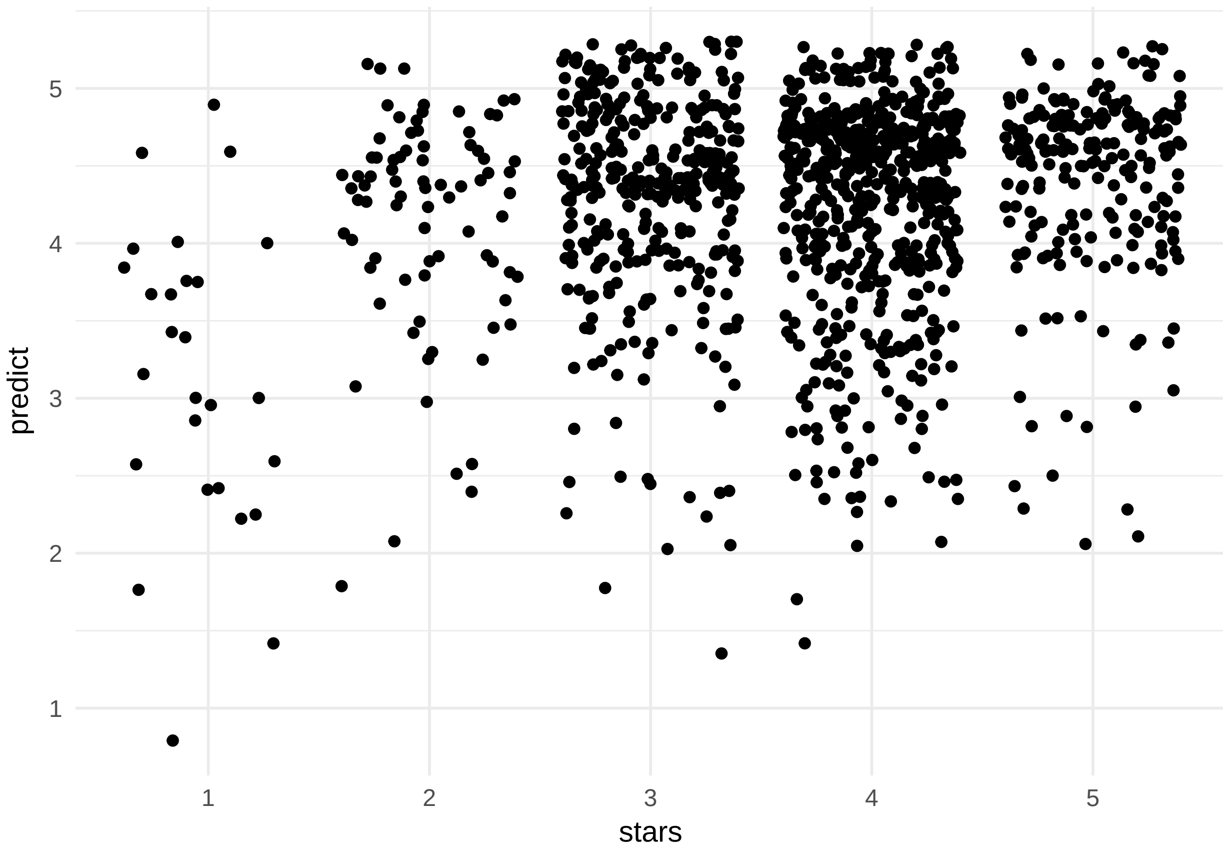
```
# our model is improved by including individual random effect
binnedplot(fitted(fit), resid(fit))
```



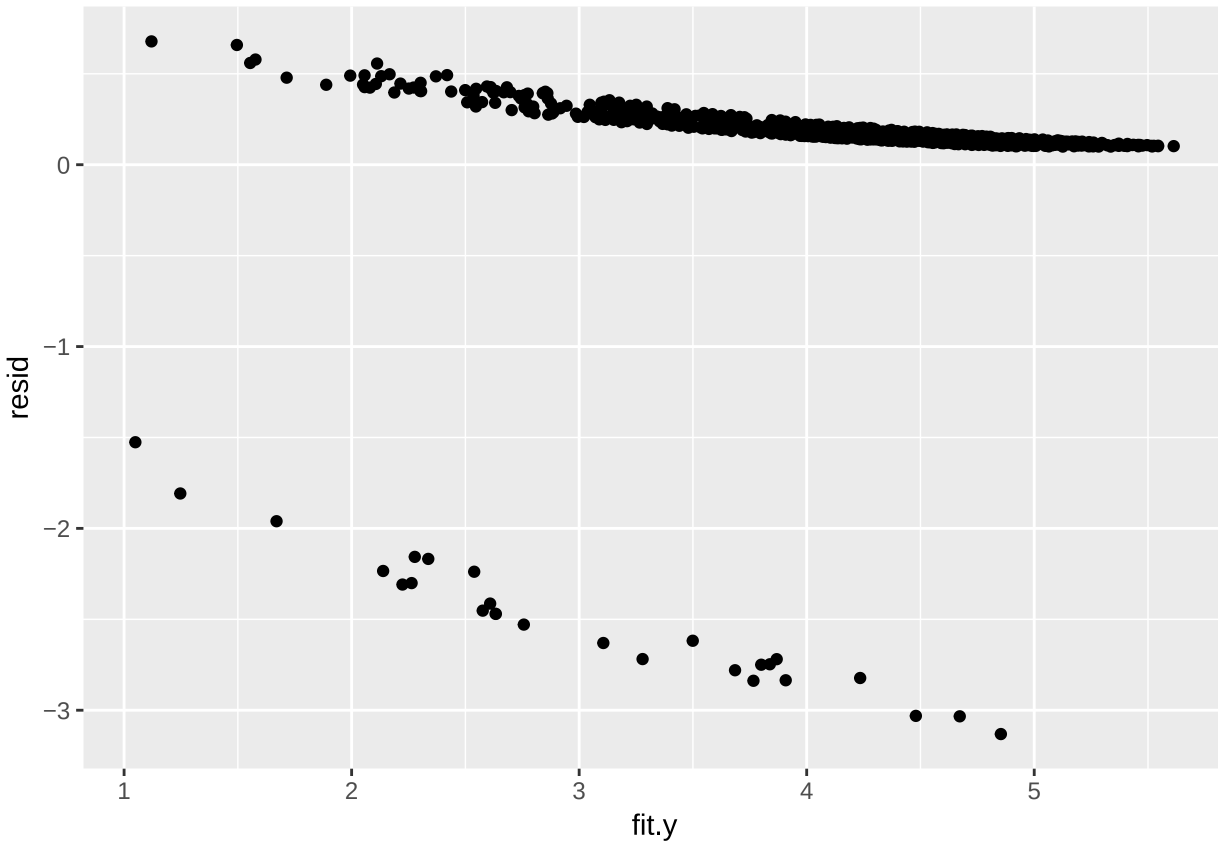

```
se2 <- sqrt(diag(vcov(fit2)))
# table of estimates with 95% CI
(tab <- cbind(Est = fixef(fit2), LL = fixef(fit2) - 1.96 * se2, UL = fixef(fit2) + 1.96 * se2))
```

```
##                              Est          LL         UL
## (Intercept)          -0.187382536 -0.45869879 0.08393372
## review_count_c        0.005104877 -0.01338831 0.02359806
## useful_pct           -0.002583547 -0.06867646 0.06350936
## average_stars         0.224936112  0.17441310 0.27545912
## store_stars           0.181945527  0.08062481 0.28326624
## store_review_count_c -0.016624036 -0.19042690 0.15717883
```

```
# fitted value v.s. residuals
fit.x <- fit@frame$`factor(stars)`
fit.y <- predict(fit)
fit.X <- data.frame(fit.x, fit.y, "resid" = resid(fit))
# predict v.s observed
# the model give more accurate results when the observed score is around 4
ggplot(fit.X, aes(fit.x, fit.y)) +
    geom_point(position = position_jitter(width = .4)) +
    geom_smooth(method = "loess",
                se = FALSE) +
    theme_minimal() +
    labs(x = "stars", y = "predict")
```




```r
# residual plots shows the negative residuals are really high
ggplot(fit.X, aes(fit.y, resid)) +
  geom_point(position=position_jitter(width=.4))
```



Discussion

Althgough from the summary of our model that the coefficients are significant, the prediction plots show otherwise. I think this is due to the limitation of data size. My laptop is not able to process such a large size of levels. There are millions of users and business in the dataset and we build the model by sampling.