Business.analysis

```
yelp.business<- stream_in(file("/project/mssphw1/yelpmssp/business.json"),verbose = F)
yelp.photo<-stream_in(file("/project/mssphw1/yelpmssp/photo.json"),verbose = F)
yelp.photo<-flatten(yelp.photo)
business<-left_join(yelp.business, yelp.photo, by = "business_id")
attributes<-yelp.business$attributes
hours<-yelp.business$hours
```

```
###0<- business hasn't photo on yelp,1<- business has photo on yelp
business$has.photo[is.na(business$photo_id)==TRUE]<-0
business$has.photo[is.na(business$photo_id)==FALSE]<-1
business$has.photo<-as.factor(business$has.photo)
### 0 the business does not have photo, 1 the business has photo but is does not include caption, 2 the
business$has.caption<-rep(2,nrow(business))
business$has.caption[is.na(business$caption)==TRUE]<-0
business$has.caption[business$caption==""]<-1
business$has.caption<-as.factor(business$has.caption)
```

```
Mon.s<-rep('',nrow(hours))
Mon.e<-rep('',nrow(hours))
Tue.s<-rep('',nrow(hours))
Tue.e<-rep('',nrow(hours))
Wed.s<-rep('',nrow(hours))
Wed.e<-rep('',nrow(hours))
Thu.s<-rep('',nrow(hours))
Thu.e<-rep('',nrow(hours))
Fri.s<-rep('',nrow(hours))
Fri.e<-rep('',nrow(hours))
Sat.s<-rep('',nrow(hours))
Sat.e<-rep('',nrow(hours))
Sun.s<-rep('',nrow(hours))
Sun.e<-rep('',nrow(hours))
```

```
for(i in 1: nrow(hours)){
  Mon.s[i]<-strsplit(hours$Monday[i], "[-]")[[1]][1]
  Mon.e[i]<-strsplit(hours$Monday[i], "[-]")[[1]][2]
}

for(i in 1: nrow(hours)){
  Tue.s[i]<-strsplit(hours$Tuesday[i], "[-]")[[1]][1]
  Tue.e[i]<-strsplit(hours$Tuesday[i], "[-]")[[1]][2]
}

for(i in 1: nrow(hours)){
  Wed.s[i]<-strsplit(hours$Wednesday[i], "[-]")[[1]][1]
  Wed.e[i]<-strsplit(hours$Wednesday[i], "[-]")[[1]][2]
}
```

```
for(i in 1: nrow(hours)){
  Thu.s[i]<-strsplit(hours$Thursday[i], "[-]")[[1]][1]
  Thu.e[i]<-strsplit(hours$Thursday[i], "[-]")[[1]][2]
}

for(i in 1: nrow(hours)){
  Fri.s[i]<-strsplit(hours$Friday[i], "[-]")[[1]][1]
  Fri.e[i]<-strsplit(hours$Friday[i], "[-]")[[1]][2]
}

for(i in 1: nrow(hours)){
  Sat.s[i]<-strsplit(hours$Saturday[i], "[-]")[[1]][1]
  Sat.e[i]<-strsplit(hours$Saturday[i], "[-]")[[1]][2]
}

for(i in 1: nrow(hours)){
  Sun.s[i]<-strsplit(hours$Sunday[i], "[-]")[[1]][1]
  Sun.e[i]<-strsplit(hours$Sunday[i], "[-]")[[1]][2]
}
```

```
###creat a function to get the start time and end time.  Using star time to subtract end time,
##I can get working hour

getTime =  function(weekday){
  time<-rep(0,nrow(hours))
  for(i in 1 : nrow(hours) ){
  if(!is.null(weekday[i])){
    h<-strsplit(weekday[i] , "[:]")[[1]][1]
    m<-strsplit(weekday[i] , "[:]")[[1]][1]
    h<-as.numeric(h)
    m<-as.numeric(m)
    time[i]<-h+m/60
  }else{
    time[i]<-NA
  }
  }
  return(time)
}
```

```
Mon_s<-getTime(Mon.s)
Mon_e<-getTime(Mon.e)
Mon<-Mon_e-Mon_s
Mon[Mon==0]<-24

Tue_s<-getTime(Tue.s)
Tue_e<-getTime(Tue.e)
Tue<-Tue_e-Tue_s
Tue[Tue==0]<-24

Wed_s<-getTime(Wed.s)
Wed_e<-getTime(Wed.e)
Wed<-Wed_e-Wed_s
Wed[Wed==0]<-24
```

```
Thu_s<-getTime(Thu.s)
Thu_e<-getTime(Thu.e)
Thu<-Thu_e-Thu_s
Thu[Thu==0]<-24

Fri_s<-getTime(Fri.s)
Fri_e<-getTime(Fri.e)
Fri<-Fri_e-Fri_s
Fri[Fri==0]<-24

Sat_s<-getTime(Sat.s)
Sat_e<-getTime(Sat.e)
Sat<-Sat_e-Sat_s
Sat[Sat==0]<-24

Sun_s<-getTime(Sun.s)
Sun_e<-getTime(Sun.e)
Sun<-Sun_e-Sun_s
Sun[Sun==0]<-24
```

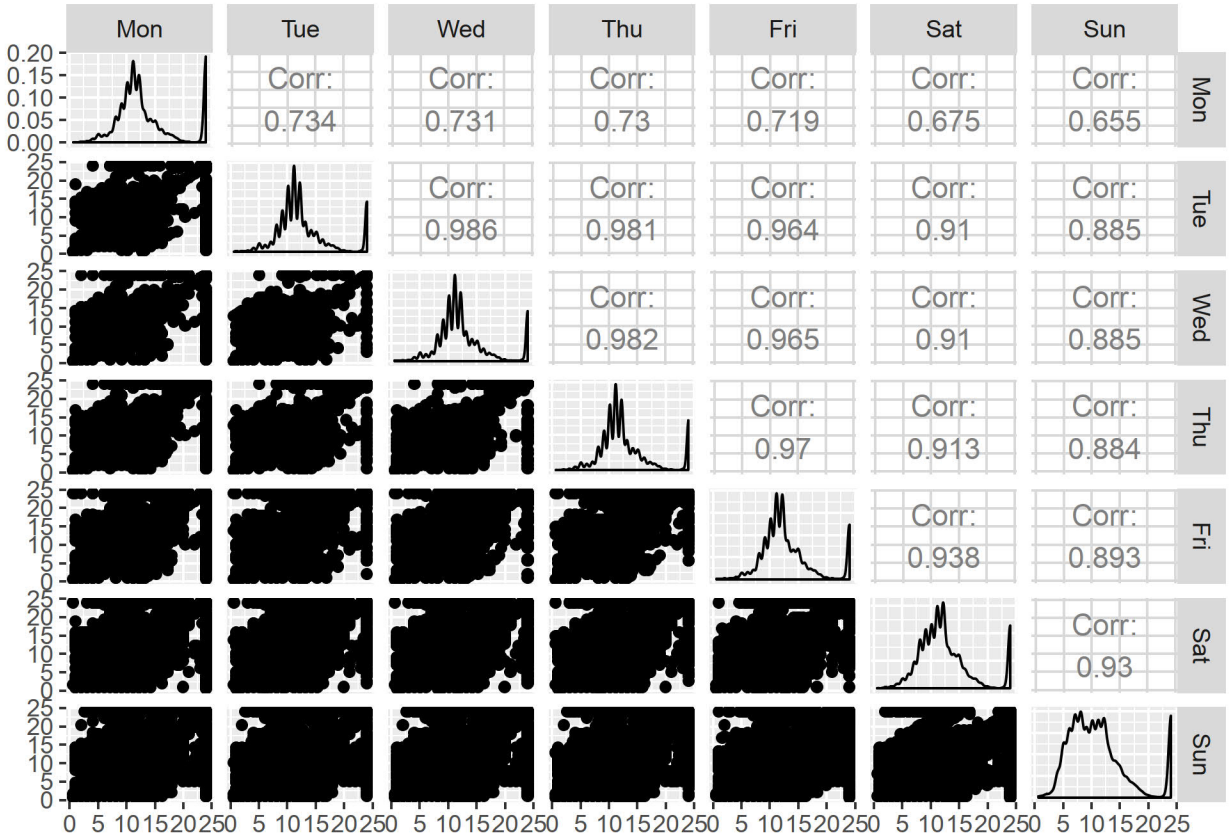
```
branch.tab.h<-yelp.business%>%count(name,sort = TRUE)
```

```
Data.Star<-data.frame(yelp.business$review_count,yelp.business$state,yelp.business$name,yelp.business$s
Data.Star<-na.omit(Data.Star)
```

```
Data.Star<-merge(Data.Star,branch.tab.h,by.x='yelp.business.name',by.y='name')
colnames(Data.Star)[colnames(Data.Star)=="yelp.business.name"] <- "name"
colnames(Data.Star)[colnames(Data.Star)=="n"] <- "Num_branch"
Data.Star$Mon<-ifelse(Data.Star$Mon<0,Data.Star$Mon+24,Data.Star$Mon)
Data.Star$Tue<-ifelse(Data.Star$Tue<0,Data.Star$Tue+24,Data.Star$Tue)
Data.Star$Wed<-ifelse(Data.Star$Wed<0,Data.Star$Wed+24,Data.Star$Wed)
Data.Star$Thu<-ifelse(Data.Star$Thu<0,Data.Star$Thu+24,Data.Star$Thu)
Data.Star$Fri<-ifelse(Data.Star$Fri<0,Data.Star$Fri+24,Data.Star$Fri)
Data.Star$Sat<-ifelse(Data.Star$Sat<0,Data.Star$Sat+24,Data.Star$Sat)
Data.Star$Sun<-ifelse(Data.Star$Sun<0,Data.Star$Sun+24,Data.Star$Sun)
Data.Star$yelp.business.stars<-as.numeric(Data.Star$yelp.business.stars)
Data.Star$is.High[Data.Star$yelp.business.stars>=4]<-1
Data.Star$is.High[Data.Star$yelp.business.stars<4]<-0
Data.Star$is.High<-as.factor(Data.Star$is.High)
```

```
X<-Data.Star[,5:11]

ggpairs(X)
```




```
Data.Star$Mean.Hours <- rowSums( Data.Star[,5:11] )/7
```

```
Data.Star%>%count(yelp.business.state)
```

```
## # A tibble: 28 x 2
##    yelp.business.state     n
##    <fct>               <int>
##  1 AB                   3575
##  2 AL                      2
##  3 AR                      1
##  4 AZ                  22346
##  5 BC                      1
##  6 CA                      6
##  7 CT                      2
##  8 FL                      2
##  9 GA                      1
## 10 IL                    808
## # ... with 18 more rows
```

```
Data.Star.C<-Data.Star %>%
  filter(!grepl("AL", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("CA", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("CT", yelp.business.state))
```

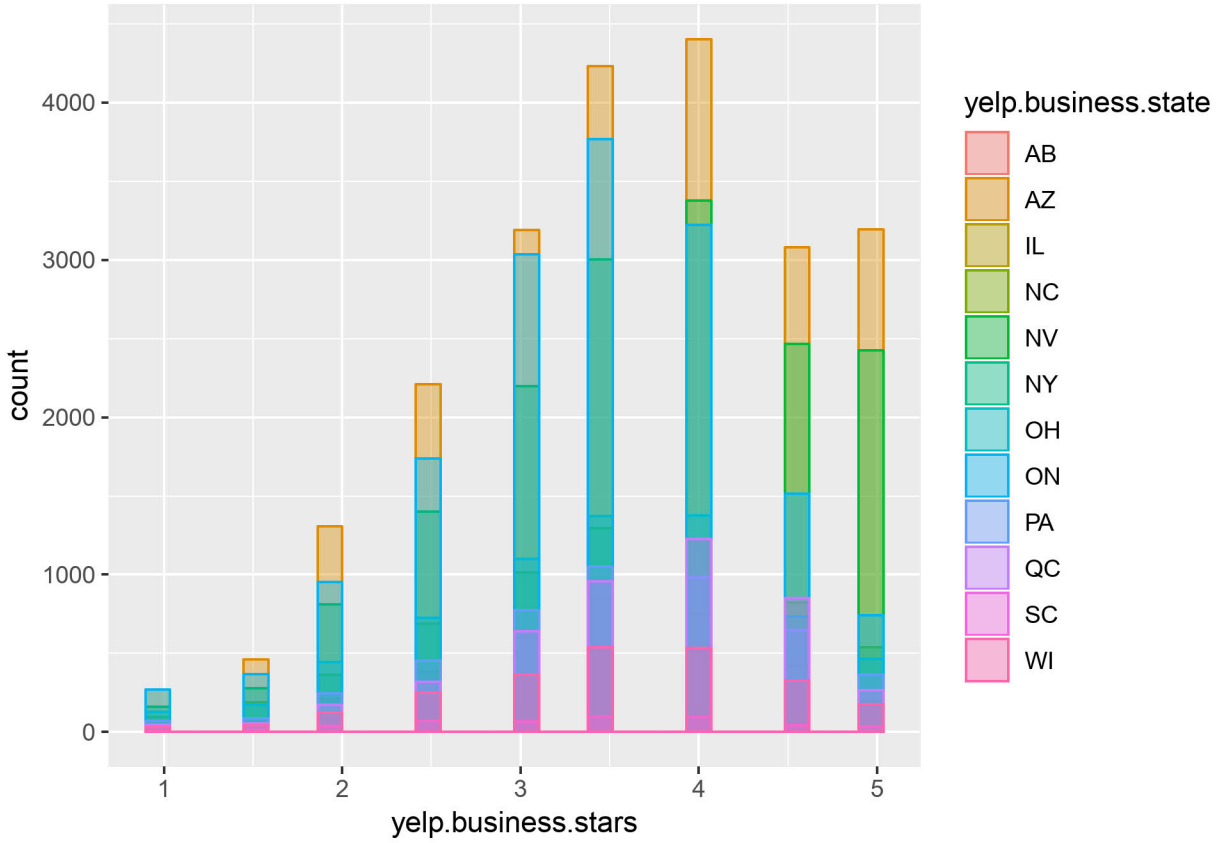
```r
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("GA", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("NE", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("AR", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("BC", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("FL", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("NJ", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("NM", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("TN", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("VA", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("VT", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("WA", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("TX", yelp.business.state))
Data.Star.C<-Data.Star.C %>%
  filter(!grepl("XGM", yelp.business.state))

show.state<-Data.Star.C%>%count(yelp.business.state)
head(show.state)
```

```
## # A tibble: 6 x 2
##   yelp.business.state     n
##   <fct>               <int>
## 1 AB                   3575
## 2 AZ                  22346
## 3 IL                    808
## 4 NC                   6217
## 5 NV                  16116
## 6 NY                     11
```

```r
star.training<-Data.Star.C[1:40000,]
star.testing<-Data.Star.C[40001:80000,]
```

```r
ggplot(Data.Star.C, aes(x = yelp.business.stars)) +
  geom_histogram(aes(color = yelp.business.state, fill = yelp.business.state),
                 position = "identity", bins = 30, alpha = 0.4)
```




```
Data.Star.C$Mean.Hours.z<-(Data.Star.C$Mean.Hours-mean(Data.Star.C$Mean.Hours))/sd(Data.Star.C$Mean.Hour
star.training<-Data.Star.C[1:40000,]
star.testing<-Data.Star.C[40001:80000,]
```

```
Stars.mod <- glmer(formula = is.High ~ Mean.Hours.z + Num_branch +
    (1  | yelp.business.state), data = star.training, family = binomial(link = "logit"))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.00141567
## (tol = 0.001, component 1)
```

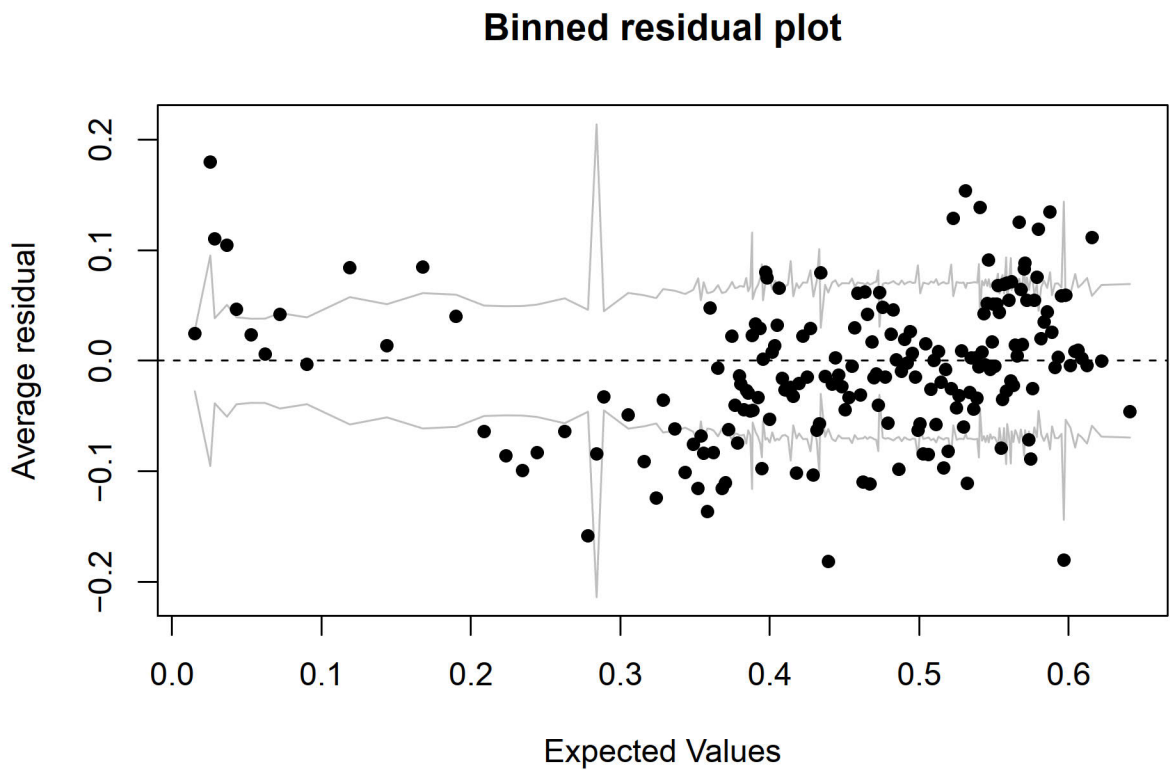
```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unide
##  - Rescale variables?
```

```
summary(Stars.mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: is.High ~ Mean.Hours.z + Num_branch + (1 | yelp.business.state)
##    Data: star.training
##
##      AIC      BIC   logLik deviance df.resid
##  51883.1  51917.5 -25937.5  51875.1    39996
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4411 -0.9183 -0.4632  0.9489  7.1391
##
## Random effects:
##  Groups              Name        Variance Std.Dev.
##  yelp.business.state (Intercept) 0.05168  0.2273
## Number of obs: 40000, groups:  yelp.business.state, 12
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.0752698  0.0709715  -1.061    0.289
## Mean.Hours.z -0.1679631  0.0105596 -15.906   <2e-16 ***
## Num_branch   -0.0140473  0.0004072 -34.499   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Mn.Hr.
## Mean.Hors.z  0.014
## Num_branch  -0.055 -0.075
## convergence code: 0
## Model failed to converge with max|grad| = 0.00141567 (tol = 0.001, component 1)
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
```

```
binnedplot(fitted(Stars.mod),residuals(Stars.mod,type="response"))
```




```
predicted<-predict(Stars.mod, newdata = star.testing)
range(predicted)
```

```
## [1] -15.8795840   0.7268014
```

```
thresholds<-0.4
confusionMatrix(star.testing$is.High, predicted, threshold = thresholds)
```

```
##       0     1
## 0 21795 16536
## 1   652  1017
```

```
coef(Stars.mod)
```

```
## $yelp.business.state
##     (Intercept) Mean.Hours.z  Num_branch
## AB -0.291406209   -0.1679631 -0.01404729
## AZ  0.157226179   -0.1679631 -0.01404729
## IL -0.080426393   -0.1679631 -0.01404729
## NC -0.152475004   -0.1679631 -0.01404729
## NV  0.316038123   -0.1679631 -0.01404729
## NY -0.116407412   -0.1679631 -0.01404729
## OH -0.179013146   -0.1679631 -0.01404729
## ON -0.496273511   -0.1679631 -0.01404729
```

```
## PA -0.003229975   -0.1679631 -0.01404729
## QC  0.135607350   -0.1679631 -0.01404729
## SC -0.222939298   -0.1679631 -0.01404729
## WI  0.032053143   -0.1679631 -0.01404729
##
## attr(,"class")
## [1] "coef.mer"
```

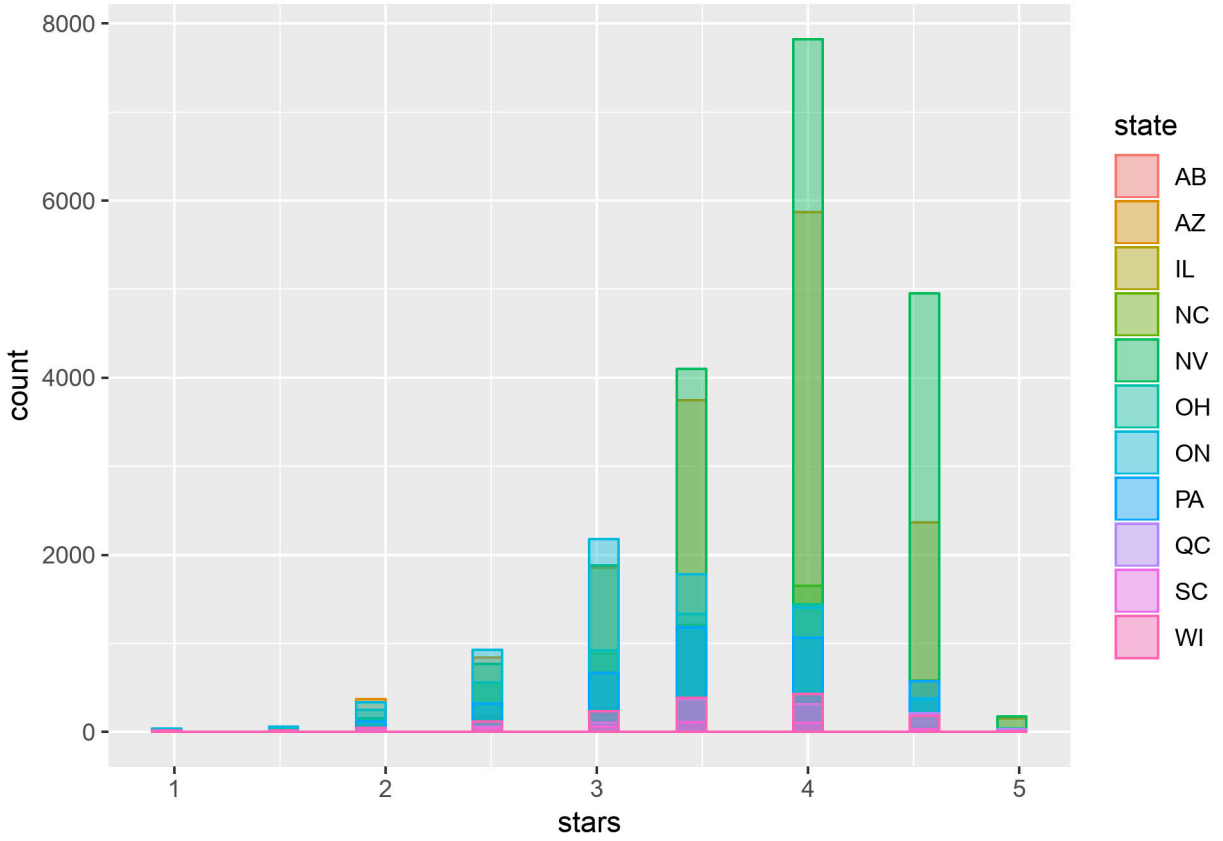
```
branch.tab<-business %>%count(name)
business<-merge(business,branch.tab,'name')
names(business)[names(business) == "n"] <- "Branch"
```

```
getUSR<-str_count(business$categories, pattern = "American|Restaurants")


business$isAmericanR<-ifelse(getUSR>=2,business$isAmericanR<-1,business$isAmericanR<-0)
business$is.High[business$stars>=4]<-1
business$is.High[business$stars<4]<-0
business.A<-subset(business, isAmericanR==1)
```

```
business.A<-business.A %>%
  filter(!grepl("FL", state))
business.A<-business.A %>%
  filter(!grepl("NM", state))
business.A<-business.A %>%
  filter(!grepl("NY", state))
business.A<-business.A %>%
  filter(!grepl("VA", state))
```

```
ggplot(business.A, aes(x = stars)) +
  geom_histogram(aes(color = state, fill = state),
                 position = "identity", bins = 30, alpha = 0.4)
```




```
train<-sample.split(business.A$is.High,SplitRatio = 0.65 )
train.A<-subset(business.A, train == T)
test.A<-subset(business.A, train == F)
```

```
mod.A<-glmer(is.High~ Branch + review_count + has.photo + (1 | state), data = train.A, family = binomial
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge with max|grad| = 0.0832075
## (tol = 0.001, component 1)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?;Model is nearly unidentifiable: large eigenvalue ratio
##  - Rescale variables?
```

```
summary(mod.A)
```

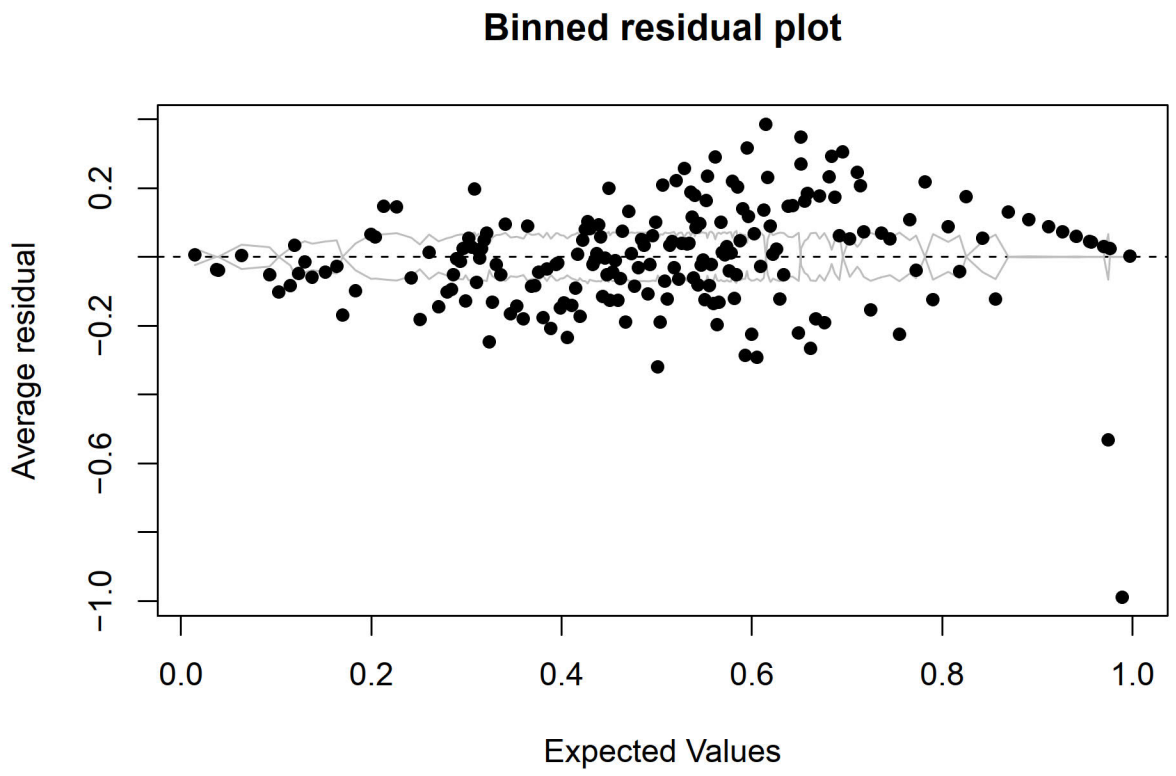
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: is.High ~ Branch + review_count + has.photo + (1 | state)
##    Data: train.A
##
##      AIC      BIC   logLik deviance df.resid
##  47179.4  47222.3 -23584.7  47169.4    39553
```

```
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.5135  -0.8501   0.0534   0.8521   9.9712
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  state  (Intercept) 0.06352  0.252
## Number of obs: 39558, groups:  state, 11
##
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.445e-01  8.809e-02  -9.587   <2e-16 ***
## Branch       -3.027e-03  8.110e-05 -37.321   <2e-16 ***
## review_count  1.207e-03  2.762e-05  43.718   <2e-16 ***
## has.photo1    5.763e-01  4.428e-02  13.015   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Branch rvw_cn
## Branch      -0.004
## review_cont  0.006 -0.172
## has.photo1  -0.437 -0.100 -0.133
## convergence code: 0
## Model failed to converge with max|grad| = 0.0832075 (tol = 0.001, component 1)
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
## Model is nearly unidentifiable: large eigenvalue ratio
##  - Rescale variables?
```

```
coef(mod.A)
```

```
## $state
##    (Intercept)       Branch review_count has.photo1
## AB  -1.0613259 -0.003026878  0.001207397  0.5762651
## AZ  -0.4270280 -0.003026878  0.001207397  0.5762651
## IL  -0.8167867 -0.003026878  0.001207397  0.5762651
## NC  -0.7881398 -0.003026878  0.001207397  0.5762651
## NV  -0.5629569 -0.003026878  0.001207397  0.5762651
## OH  -0.9232496 -0.003026878  0.001207397  0.5762651
## ON  -1.3769473 -0.003026878  0.001207397  0.5762651
## PA  -0.8705263 -0.003026878  0.001207397  0.5762651
## QC  -0.6586359 -0.003026878  0.001207397  0.5762651
## SC  -0.9769095 -0.003026878  0.001207397  0.5762651
## WI  -0.8233585 -0.003026878  0.001207397  0.5762651
##
## attr(,"class")
## [1] "coef.mer"
```

```
binnedplot(fitted(mod.A),residuals(mod.A,type="response"))
```




```
predicted.A<-predict(mod.A, newdata = train.A,type = 'response' )


confusionMatrix(train.A$is.High, predicted.A, threshold = 0.5)
```

```
##        0     1
## 0 12958  5625
## 1  6585 14390
```