STA841 Final Project

Drew Jordan, Emily Shao

11/27/2016

Introduction:

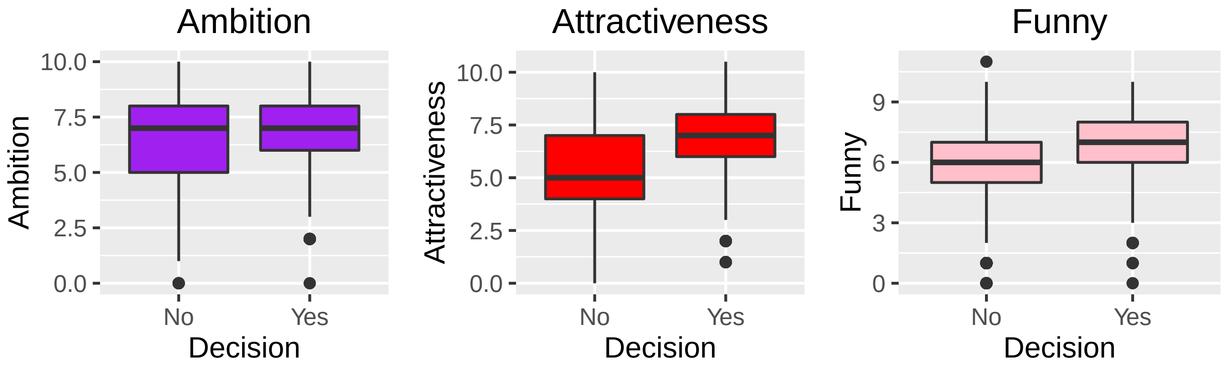
The dataset that we have chosen to analyze is the Speed Dating Experiment dataset available on Kaggle.com. The data was collected by Columbia Business School professors Ray Fisman and Sheena Iyengar from multiple speed dating sessions, in which students participated, conducted at Columbia University between 2002 and 2004. According to Kaggle, this is how the data was collected:

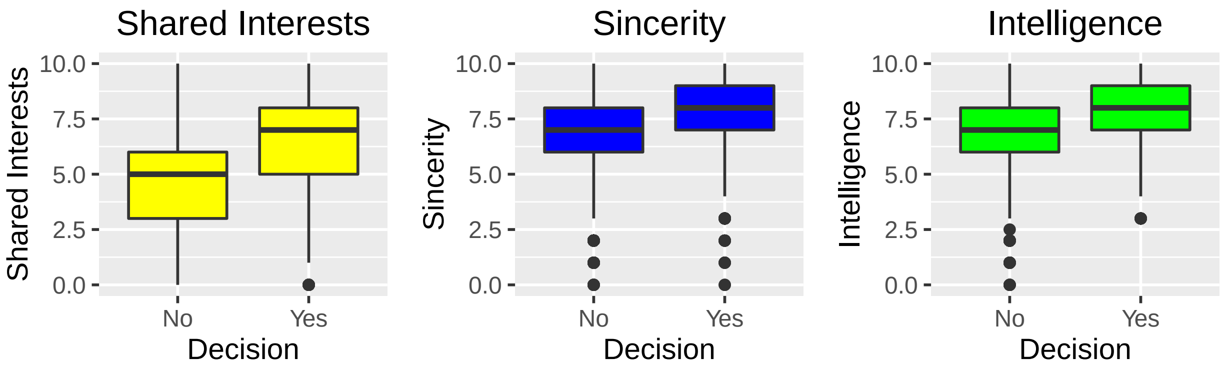
> During the events, the attendees would have a four minute "first date" with every other participant of the opposite sex. At the end of their four minutes, participants were asked if they would like to see their date again. They were also asked to rate their date on six attributes: Attractiveness, Sincerity, Intelligence, Fun, Ambition, and Shared Interests.
>
> The dataset also includes questionnaire data gathered from participants at different points in the process. These fields include: demographics, dating habits, self-perception across key attributes, beliefs on what others find valuable in a mate, and lifestyle information.

Our project will focus on the participants' initial ratings that they received from each partner on the 6 attributes as well as the demographic information of each participant. The demographic information we have for each participant includes gender, race, age, field of study, goals for participating in the speed dating event, dating tendencies, going-out tendencies, intended career, and the activities that the participants enjoy. Our ultimate goal is to discover what features of a participant make him/her more likely to receive a match (whether or not the participant's partner wants to see him/her again) from each of his/her partners on the night of the event and if these features are different between men and women.

Exploratory Data Analysis

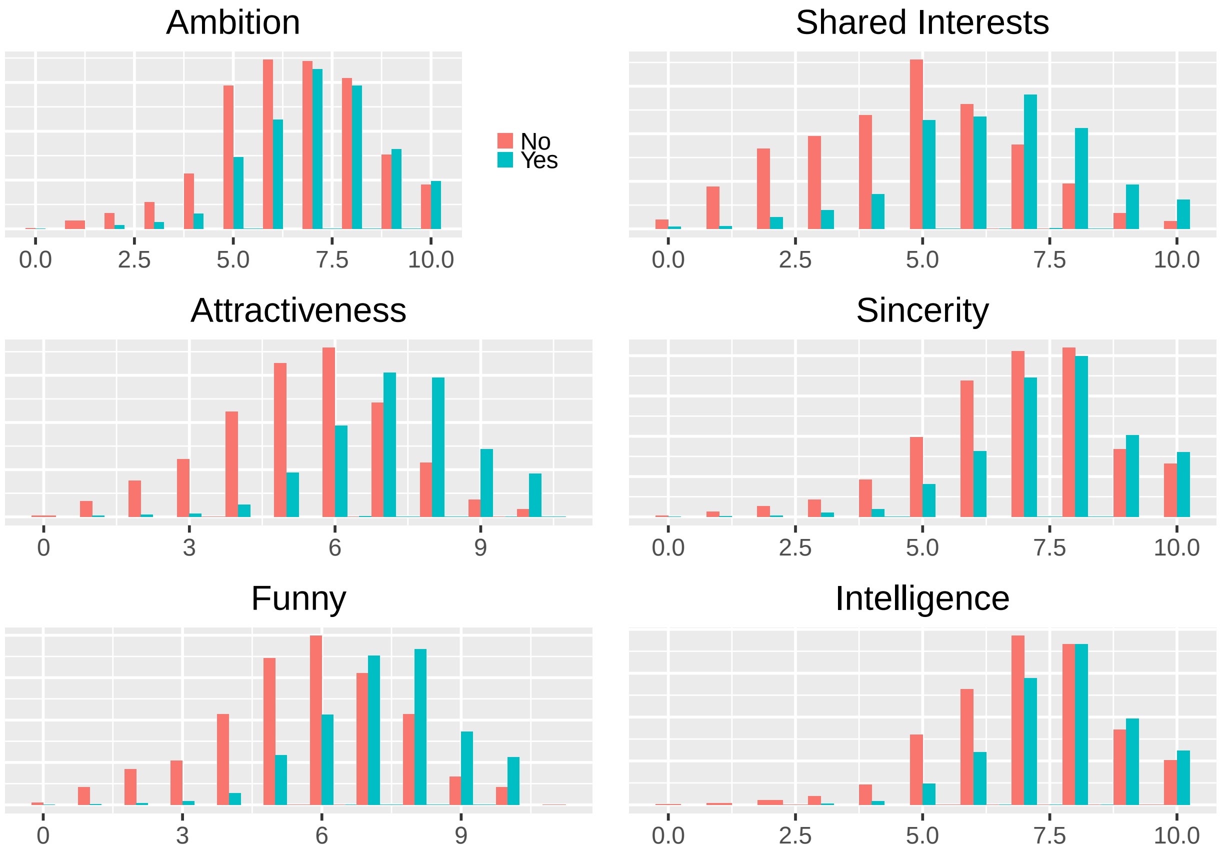
Before we fit any models, we first explore the relationships between individual covariates in our data and whether or not a person gets a match from his/her partner. Of primary interest to us is the relationship between how an individual is rated by his/her partner on the six attributes (Ambition, Funny, Shared Interests, Attractiveness, Intelligence, and Sincerity). These ratings are collected immediately after a speed date during the event for each person. Each attribute is rated on a 0-10 scale with 0 being the lowest rating and 10 being the highest rating. To ease the interpretation of our analysis, we treat these ratings as continuous variables. Below, we display boxplots for each attribute grouped by whether or not an individual's partner chose to match with him/her.





Based on these boxplots, it appears that the ratings for Funny, Shared Interests, Attractiveness, Intelligence, and Sincerity each have positive relationships with the decision made by an individual's partner. These preliminary results match our general intuition given that we expect these attributes to be important when determining the overall attractiveness of a potential date. Surprisingly, there does not seem to be any relationship between a participant's ambition and his/her overall dating attractiveness.

Additionally, we create segmented barplots for each of the 6 attributes to show the proportion within each rating level that received matches.



We can see that for every attribute besides ambition and perhaps sincerity, the higher the score, the higher the proportion of matches that are received. This agrees with the box plots.

Methods:

The primary method we will use to determine which features of a partner are most highly associated with receiving a match is logistic regression. We use logistic regression because we are interested in predicting the probability that a particular individual receives a match from his/her partner given the ratings that he/she receives at the time of the speed dating event and his/her demographic information. We choose to use logistic regression over any other binary response regression method because of the easy interpretation of logistic regression coefficients. These interpretations of the regression coefficients are as follows:

- For continuous covariates, a unit increase results in an increase in the log-odds of the response occurring equal to the coefficient.
- For categorical covariates, the coefficient corresponds to the log-odds ratio of the response occurring between the value of the covariate and a baseline level of the covariate.

Both of these interpretations can easily be converted back to a probability of the response occurring given the covariates.

Because our data contains 37 variables that we believe may be associated with a participant receiving a match, we will use forward selection to reduce the number of variables that are included in our final model. This will reduce the model's complexity while increasing the interpretability of the selected covariates. The forward selection scheme that we implement is as follows:

1. Fit an intercept-only model on whether or not a participant receives a match from his/her partner.
2. Fit a model for each individual excluded covariate, and use an ANOVA $X^2 - test$ to test for the significance of each model versus the intercept-only model.
3. Add the covariate with the lowest associated p-value from the ANOVA $X^2 - test$ to the best model we are constructing.
4. Fit a model for each excluded covariate added to the previous best model, and use an ANOVA $X^2 - test$ to test for the significance of each model versus the previous best model.
5. Repeat 3. and 4. until we reach a point in which adding any additional variables no longer produces a significant change in the best model according to the ANOVA $X^2 - test$.

We build models using this method for men and women individually to aid in the comparison of the preferences between genders.

Results

The resulting logistic regression model for the men in our data (so the preferences of women) is shown in Table 1. We see that the best model for the preferences of women according to our forward selection process includes women's ratings of men in attractiveness, sense of humor, ambition, and shared interests. It also includes women's perceived probability that men also match them and a categorical variable outlining the frequency in which men go on dates. The "date" variable breaks down as follows: 1 = Several times a week, 2 = Twice a week, 3 = Once a week, 4 = Twice a month, 5 = Once a month, 6 = Several times a year, and 7 = Almost never.

The model for women (the preferences of men) is shown in Table 2. We can see that several variables are shared between when examining the preferences of men and the preferences of women. For both genders, a partner's attractiveness (attr_o), sense of humor (fun_o), and ambition (amb_o) are associated with the probability that an individual receives a match. The perceived probability that a partner will also match with

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-5.0632	0.4635	-10.92	0.0000
attr_o	0.4122	0.0299	13.76	0.0000
fun_o	0.2435	0.0335	7.27	0.0000
amb_o	-0.1522	0.0306	-4.97	0.0000
prob_o	0.1378	0.0231	5.97	0.0000
shar_o	0.2353	0.0278	8.46	0.0000
date2	-0.1126	0.4280	-0.26	0.7925
date3	-0.5708	0.3992	-1.43	0.1528
date4	-0.5553	0.3883	-1.43	0.1527
date5	-0.7692	0.3906	-1.97	0.0489
date6	-0.7885	0.3901	-2.02	0.0432
date7	-0.7802	0.3981	-1.96	0.0500

Table 1: Preferences of Women

an individual (prob_o) is associated as well. It is interesting to note that all of these variables are positively associated with the probability of receiving a match, but ambition is negatively associated. Additionally, all of the variable's magnitudes in the model are similar except men appear to place greater emphasis on the attractiveness of women than vice-versa.

Beyond the shared variables from the two models, the women's preferences model produces two additional variables: a positive association in shared interests (shar_o) and a negative association in the dating tendencies of men (date). The men's preferences model produces on additional significant variable: a slightly negative association with women's interest in watching sports on television (tvsports). For all variables in both models with the exception of the date variable, the interpretations of the coefficients are that for each unit increase in the variable, the log-odds of receiving a match increases by the amount of the coefficient. For the date variable, each coefficient represents the log-odds ratio between the specified level and the baseline level (date1). Finally, we find that there is not significant overdispersion in either model.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-6.4852	0.2668	-24.31	0.0000
attr_o	0.7048	0.0323	21.81	0.0000
fun_o	0.2244	0.0309	7.26	0.0000
amb_o	-0.1589	0.0299	-5.32	0.0000
prob_o	0.2953	0.0227	12.98	0.0000
tvsports	-0.0347	0.0156	-2.22	0.0262

Table 2: Prefences of Men

Conclusions

Following our methodology for forward model selection, we have created separate models summarizing the preferences of men and women using the Speed Dating dataset from Columbia University. We have shown that both men and women value the attractiveness and humors of their partners, while actually devaluing ambitious partners. We have also shown that women prefer men that go on more dates and that share their interests. Men appear to prefer women that are less interested in watching sports. These findings generally match our intuition about what makes people more atractive to date.

There are some possible limitations to our analysis. Since our model is based on data that was collected from an experiment, weaknesses of our models could arise from the data itself and our model assumptions. First, our models may not generalize well to the entire population of men and women who participate in a speed dating event. Since these models were fit to a representative Columbia University graduate student population in 2002-2004, we cannot claim that our models generalize perfectly to anyone who is between 18 to 55 years old (range of age in this experiment). This is because the life styles of graduate students and professionals may affect their preferences when picking romantic partners. Second, our methodology only considers a few interactions between variables that may affect a person's decision, while there might be additional interactions we have neglected that could be significant in our models. We only considered interaction terms that could be interpretted sensibly. Third, our models do not take into account potential correlation between our explanatory variables. Additionally, while each observation is treated independently in our models, each person was rated multiple times (once for every speed date he/she went on), which suggests that some observations might be correlated. Our model also does not address each individual's preferences in selecting a romantic partner. Fourth, we treat the ratings for each of the 6 attributes as continuous variables when they can only take values from 0-10. We may get a better fit if we treat those variables as ordered categorical variables. We chose to treat them as continuous because that eases the interpretation of their coefficients. For future analysis, we may consider a linear mixed model to address the issues raised from our models.

APPENDIX:

```
knitr::opts_chunk$set(echo = TRUE)

# Multiple plot function
#
# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)
# - cols:   Number of columns in layout
# - layout: A matrix specifying the layout. If present, 'cols' is ignored.
#
# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),
# then plot 1 will go in the upper left, 2 will go in the upper right, and
# 3 will go all the way across the bottom.
#
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                    ncol = cols, nrow = ceiling(numPlots/cols))
  }

 if (numPlots==1) {
    print(plots[[1]])

  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))
    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

```
library(data.table)
library(dplyr)
library(MASS)
library(ggplot2)
library(xtable)
options(xtable.comment = FALSE)

dating=fread("Speed Dating Data.csv", verbose = FALSE)
# New variable to label dec_o as yes and no
dating$dec_o_word = ifelse(dating$dec_o == 1, "Yes", "No")
```

```
# Plot for attractiveness
attr_plot = ggplot(data = dating, aes(x = dec_o_word, y = attr_o)) +
  geom_boxplot(fill = "red") +
  xlab("Decision") + ylab("Attractiveness") +
  ggtitle("Attractiveness")
# Plot for sincerity
sinc_plot = ggplot(data = dating, aes(x = dec_o_word, y = sinc_o, group = dec_o)) +
  geom_boxplot(fill = "blue") +
  xlab("Decision") + ylab("Sincerity") +
  ggtitle("Sincerity")
# Plot for intelligence
intel_plot = ggplot(data = dating, aes(x = dec_o_word, y = intel_o, group = dec_o)) +
  geom_boxplot(fill = "green") +
  xlab("Decision") + ylab("Intelligence") +
  ggtitle("Intelligence")
# Plot for ambition
amb_plot = ggplot(data = dating, aes(x = dec_o_word, y = amb_o, group = dec_o)) +
  geom_boxplot(fill = "purple") +
  xlab("Decision") + ylab("Ambition") +
  ggtitle("Ambition")
# Plot for shared interests
shar_plot = ggplot(data = dating, aes(x = dec_o_word, y = shar_o, group = dec_o)) +
  geom_boxplot(fill = "yellow") +
  xlab("Decision") + ylab("Shared Interests") +
  ggtitle("Shared Interests")
# Plot for funny
fun_plot = ggplot(data = dating, aes(x = dec_o_word, y = fun_o, group = dec_o)) +
  geom_boxplot(fill = "pink") +
  xlab("Decision") + ylab("Funny") +
  ggtitle("Funny")
```

```r
# Barplots showing the number of people who get a match
fun_bar = ggplot(data = dating, aes(x = fun_o, fill = dec_o_word)) +
  geom_bar(position = "dodge", width = .5) +
  xlab("Funny") + ylab("Count") +
  ggtitle("Funny") +
  guides(fill = FALSE) +
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.x=element_blank())
sinc_bar = ggplot(data = dating, aes(x = sinc_o, fill = dec_o_word)) +
  geom_bar(position = "dodge", width = .5) +
  xlab("Sincerity") + ylab("Count") +
  ggtitle("Sincerity") +
  guides(fill = FALSE) +
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.x=element_blank())
amb_bar = ggplot(data = dating, aes(x = amb_o, fill = dec_o_word)) +
  geom_bar(position = "dodge", width = .5) +
  ylab("Count") + ggtitle("Ambition") +
  guides(fill = guide_legend(title = NULL)) +
  theme(legend.key.size = unit(.25, "cm"),
        axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.x=element_blank())
intel_bar = ggplot(data = dating, aes(x = intel_o, fill = dec_o_word)) +
  geom_bar(position = "dodge", width = .5) +
  xlab("Intelligence") + ylab("Count") +
  ggtitle("Intelligence") +
  guides(fill = FALSE) +
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.x=element_blank())
shar_bar = ggplot(data = dating, aes(x = shar_o, fill = dec_o_word)) +
  geom_bar(position = "dodge", width = .5) +
  xlab("Shared Interests") + ylab("Count") +
  ggtitle("Shared Interests") +
  guides(fill = FALSE) +
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.x=element_blank())
attr_bar = ggplot(data = dating, aes(x = attr_o, fill = dec_o_word)) +
  geom_bar(position = "dodge", width = .5) +
  xlab("Attractiveness") + ylab("Count") +
  ggtitle("Attractiveness") +
  guides(fill = FALSE) +
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank(),
        axis.title.x=element_blank())

# Plotting them all
multiplot(amb_plot, attr_plot, fun_plot, cols = 3)
```

```
multiplot(shar_plot, sinc_plot, intel_plot, cols = 3)
```

```
multiplot(amb_bar, attr_bar, fun_bar,
          shar_bar, sinc_bar, intel_bar, cols = 2)
```

```
# List of predictors
predictors = c("int_corr","samerace","race_o", "attr_o", "sinc_o", "intel_o", "fun_o", "amb_o",
               "shar_o", "prob_o", "met_o", "race", "field_cd", "goal", "date", "go_out",
               "career_c","sports","tvsports","exercise", "dining", "museums", "art", "hiking",
               "gaming","clubbing","reading","tv", "theater","movies","concerts", "music",
               "shopping","yoga","exphappy","expnum", "age_diff")

dating=fread("Speed Dating Data.csv") %>%
  filter(gender == 1)
dating$age_diff = dating$age - dating$age_o
dating=as.data.frame(dating[complete.cases(as.data.frame(dating)[c(predictors,"dec_o")])])
dating$samerace=as.factor(dating$samerace)
dating$race=as.factor(dating$race)
dating$race_o=as.factor(dating$race_o)
dating$field_cd=as.factor(dating$field_cd)
dating$goal = as.factor(dating$goal)
dating$date = as.factor(dating$date)
dating$go_out = as.factor(dating$go_out)


# Fitting the null model
model_null = glm(data = dating, dec_o ~ 1, family = binomial(link = "logit"))

significance = 0

# Baseline formula
base = "dec_o~ 1"


# Storage for best p_values
best_pvalues = c(100)
# Storage for best models
best_models = list(model_null)
# Storage for chosen model predictors
preds_model = c()


# While loop through forward selection until our new p-value is worse than our old one
while(significance < .05 & length(preds_model) < length(predictors))
{
  # Storage for models
  models = list()
  # Storage for p_values
  p_values = c()

  summary=list()

  # Looping through each combination of predictors to find the best model
  # Using forward selection
  for(i in 1:(length(predictors)))
  {
    if(predictors[i] %in% preds_model)
    {
      next
    }
    # Formula for glm
    formula_tmp = formula(paste(base, "+", predictors[i]))
    # Fitting logistic regression for this formula
    models[[i]] = glm(formula_tmp, data = dating, family = binomial(link = "logit"))


    # Pearson residuals
    PR = residuals(models[[i]], type = "pearson")
    # Checking for overdispersion
    overdispersion = sum(PR^2) / models[[i]]$df.res
    if(abs(overdispersion - 1) > .1)
    {
    #Refitting logistic regression including the dispersion paramater
      summary[[i]] = summary(glm(formula_tmp, data = dating, family = binomial(link = "logit")),
                            dispersion = overdispersion)
    }
    # Extracting the p-value from the ANOVA table
    p_values[i] = anova(best_models[[length(best_models)]],models[[i]],
                       test = "Chisq")$`Pr(>Chi)`[2]
  }

  # Extracting the best predictor
  preds_best = predictors[p_values == min(p_values, na.rm = TRUE)]
  preds_best = preds_best[is.na(preds_best) == FALSE]
  preds_model = c(preds_model, preds_best)

  # Updating best_pvalues
  best_pvalues = c(best_pvalues, min(p_values, na.rm = TRUE))

  # Storing the best model
  best_models = list(best_models, models[[which.min(p_values)]])

  # Updating base formula
  base = paste(base, "+", preds_best[length(preds_best)])

  # Checking the significance of our model
  significance = best_pvalues[length(best_pvalues)]
}

if(best_pvalues[length(best_pvalues)] > .05)
```

```
{
  preds_model = preds_model[-length(preds_model)]
}

f = paste("dec_o ~")
# Final Formula
for(i in 1:length(preds_model))
{
  f = paste(f, "+", preds_model[i])
}
f = formula(f)

# Fitting final glm
model_final = glm(f, data = dating, family = binomial(link = "logit"))
# summary(model_final)
```

```
predictors = c("attr_o", "fun_o", "amb_o",
               "prob_o", "shar_o", "date")

dating=fread("Speed Dating Data.csv") %>%
  filter(gender == 1)
dating=as.data.frame(dating[complete.cases(as.data.frame(dating)[c(predictors,"dec_o")])])
dating$date = as.factor(dating$date)

model_final = glm(dec_o ~ attr_o + fun_o + amb_o + prob_o + shar_o + date,
                  data = dating, family = binomial(link = "logit"))
# summary(model_final)

overdispersion_f = sum(residuals(model_final, type = "pearson")^2) / model_final$df.residual

xtable(model_final, caption = "Preferences of Women")
```

```
# List of predictors
predictors = c("int_corr","samerace","race_o", "attr_o", "sinc_o", "intel_o", "fun_o", "amb_o",
               "shar_o", "prob_o", "met_o", "race", "field_cd", "goal", "date", "go_out",
               "career_c","sports","tvsports","exercise", "dining", "museums", "art", "hiking",
               "gaming","clubbing","reading","tv", "theater","movies","concerts", "music",
               "shopping","yoga","exphappy","expnum", "age_diff")

dating=fread("Speed Dating Data.csv") %>%
  filter(gender == 0)
dating$age_diff = dating$age - dating$age_o
dating=as.data.frame(dating[complete.cases(as.data.frame(dating)[c(predictors,"dec_o")])])
dating$samerace=as.factor(dating$samerace)
dating$race=as.factor(dating$race)
dating$race_o=as.factor(dating$race_o)
dating$field_cd=as.factor(dating$field_cd)
dating$goal = as.factor(dating$goal)
dating$date = as.factor(dating$date)
dating$go_out = as.factor(dating$go_out)

# Adding interaction terms
predictors = c(predictors, "age:goal", "age:go_out")
```

```
# Fitting the null model
model_null = glm(data = dating, dec_o ~ 1, family = binomial(link = "logit"))

significance = 0

# Baseline formula
base = "dec_o~ 1"


# Storage for best p_values
best_pvalues = c(100)
# Storage for best models
best_models = list(model_null)
# Storage for chosen model predictors
preds_model = c()


# While loop through forward selection until our new p-value is worse than our old one
while(significance < .05 & length(preds_model) < length(predictors))
{
  # Storage for models
  models = list()
  # Storage for p_values
  p_values = c()

  summary=list()

  # Looping through each combination of predictors to find the best model
  # Using forward selection
  for(i in 1:(length(predictors)))
  {
    if(predictors[i] %in% preds_model)
    {
      next
    }
    # Formula for glm
    formula_tmp = formula(paste(base, "+", predictors[i]))
    # Fitting logistic regression for this formula
    models[[i]] = glm(formula_tmp, data = dating, family = binomial(link = "logit"))


    # Pearson residuals
    PR = residuals(models[[i]], type = "pearson")
    # Checking for overdispersion
    overdispersion = sum(PR^2) / models[[i]]$df.res
    if(abs(overdispersion - 1) > .1)
    {
    #Refitting logistic regression including the dispersion paramater
      summary[[i]] = summary(glm(formula_tmp, data = dating, family = binomial(link = "logit")),
                            dispersion = overdispersion)
    }
    # Extracting the p-value from the ANOVA table
    p_values[i] = anova(best_models[[length(best_models)]],models[[i]],
                       test = "Chisq")$`Pr(>Chi)`[2]
  }

  # Extracting the best predictor
  preds_best = predictors[p_values == min(p_values, na.rm = TRUE)]
  preds_best = preds_best[is.na(preds_best) == FALSE]
  preds_model = c(preds_model, preds_best)

  # Updating best_pvalues
  best_pvalues = c(best_pvalues, min(p_values, na.rm = TRUE))

  # Storing the best model
  best_models = list(best_models, models[[which.min(p_values)]])

  # Updating base formula
  base = paste(base, "+", preds_best[length(preds_best)])

  # Checking the significance of our model
  significance = best_pvalues[length(best_pvalues)]
}

if(best_pvalues[length(best_pvalues)] > .05)
{
  preds_model = preds_model[-length(preds_model)]
}

f = paste("dec_o ~")
# Final Formula
for(i in 1:length(preds_model))
{
  f = paste(f, "+", preds_model[i])
}
f = formula(f)

# Fitting final glm
model_final = glm(f, data = dating, family = binomial(link = "logit"))
# summary(model_final)
```

```
predictors = c("attr_o", "fun_o", "amb_o",
               "prob_o", "tvsports")

dating=fread("Speed Dating Data.csv") %>%
  filter(gender == 0)
dating=as.data.frame(dating[complete.cases(as.data.frame(dating)[c(predictors,"dec_o")])])

model_final = glm(dec_o ~ attr_o + fun_o + amb_o + prob_o + tvsports,
                  data = dating, family = binomial(link = "logit"))

overdispersion_m = sum(residuals(model_final, type = "pearson")^2) / model_final$df.residual

xtable(model_final, caption = "Prefences of Men")
```