Data mining and data analysis for predicting tips in restaurants

Zeqi Ma,Meilin Li,Yanyan Feng

IEOR242 project

Abstract

The traditional way of observing tip and waiter is to analyze the social market. However, in reality, there are subjective consciousness and the situation that can not be quantified. So how should the waiter make the tip highest is a problem. In our project, we do the data imputation, deal with the outliers ,visualize and analyse the data, build the models to make prediction, we can quantify the factors that affect tips, and make suggestions on services according to the importance.

Keywords: Type your keywords here, separated by semicolons ;

1. Introduction

Tipping is a very common thing in the United States and many other countries. And those tips are also the major income sources for many waiters and waitresses. What's more, with tips paid by customers, restaurants are not required to pay the regular minimum wage. According to US department of labor, the US federal minimum wage for a tipped server is only $2.13 an hour. So paying tips will definitely release the pressure restaurants take, and boosts the economy.

Some people have investigated the topic of tips, but most of them start from the principle of market. In our project, we will focus our analysis on the servers' data, use visualization methods to find the relationship between them, and we attempt to weed through a list of 77 variables in order to isolate as well as quantify the effect that those factors have on the success of a server.

The purpose of this project is to develop a guide that will provide servers information on how they should behave to increase their tips. This information could be utilized by new restaurant as well as waiters or waitresses who would like to maximize the gratuity potential.

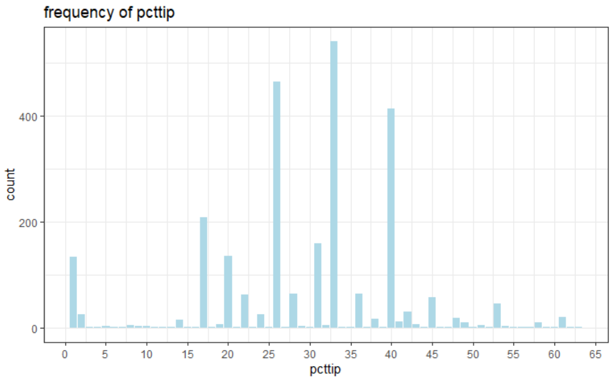
2. Data manipulation and analysis

The data comes from Professor William Michael Lynn of consumer behavior and marketing at the Cornell University School of Hotel Administration. Ithaca, NY, collected tipping information from more than 2,400 waiters and waitresses from all over the world. He used data to analyze the relationship between tips and customers from the perspective of the market. (http://www.tippingresearch.com/index.html)

The data has 77 variables, but generally there are four types of variables: (1)54 discrete or integer variables, (2)12 factor variables, (3)9 continuous or numerical variables, and (4)1 logical variable. These variables can be regrouped into three main categories. The first category describes the attitude. The second category describes the features of servers like hair or race. And the last category describes guests. We will process data from these three different perspectives.

To complete the objectives of this project of evaluating factors that affect tipping, some variables (remoteip, datercvd, submit_time, and more_mos) were removed from the dataset because they are unimportant and have little relationship with the target variables. These variables are not expected to contribute to the detecting of variables that affect tipping and thus removed.

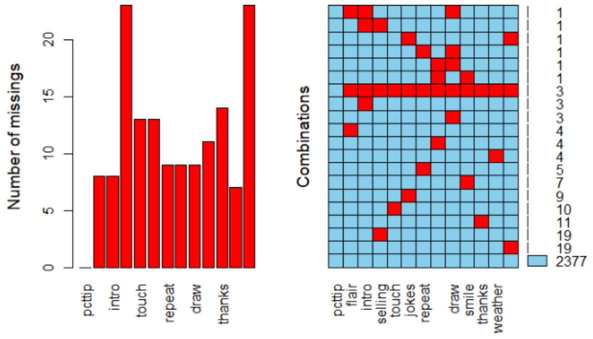
The plot below is the distribution of dependent variable: percentage of tip. It can be seen that there is a peak, as the vast majority of restaurants will give tip options, the most are 15, 17, 20, etc. In addition, for our purpose, the tip of 0 should be removed, since most of those are fast food restaurants, which are free of charge by default. Also, we set a maximum tip value, 30. If the tip is more than 30, it is considered that the waiter performs well or the data is an unrealizable data. So we will make them turn to 30



2.1. Variables describes the attitude

(1)EFFORT

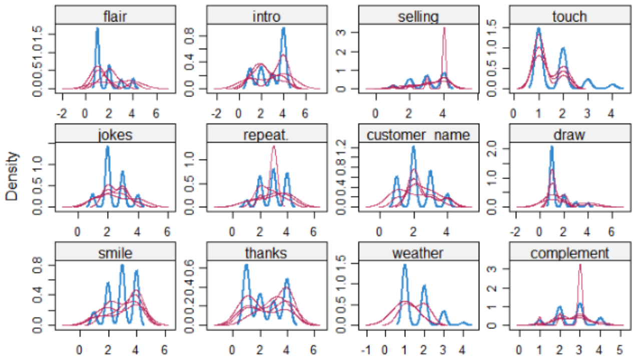
From the data set, there are about 12 variables to describe how much effort that a waiter/waitress put into work. As all of these kinds of values are factors from one to four representing the degree of effort, there are no outliers. Only the missing values should be focused on.



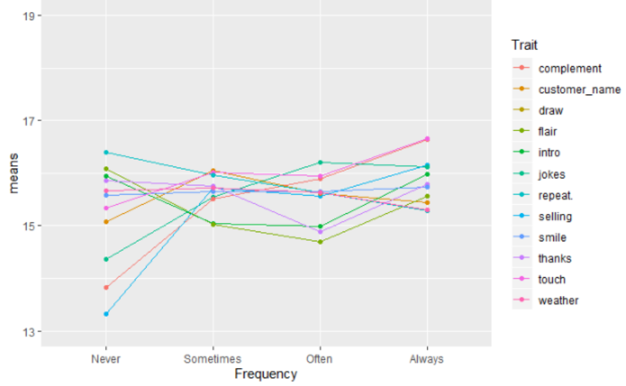
The distribution of the missing values is shown above.

From the plot, it is easier to be noticed that some of the variables are missing at random. Compared with deleting them directly or using mean values to replace, multiple imputation method seems much more reliable. The mice function in R can be used to achieve this goal. According to the valuable datasets, the random forest method is chosen to estimate the missing value by using chained equation approach. Then 5 completed dataset can be got after imputation, the best fitting one will be chosen as the final result.

The imputation result is shown as below. The blue lines are the distributions of the original data with missing values while the red lines are the 5 completed datasets after multiple imputation methods. We can see that most of the red lines are similar to the blue lines.

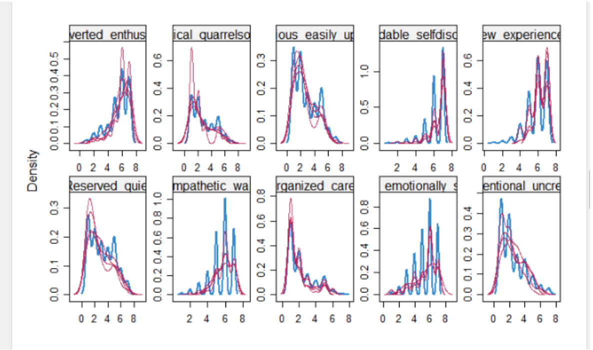


In order to interpret the relationship between the tips and these 13 features, line chart is drawn, the y axis is the mean of tips, the x axis is frequency. For example, in blue line, "never" means the waiter never sell things. As the plots shows, all of these features are associated with the response (tips)

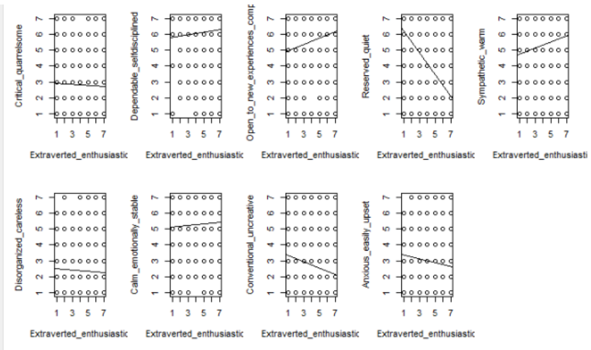


(2)PERSONALITY

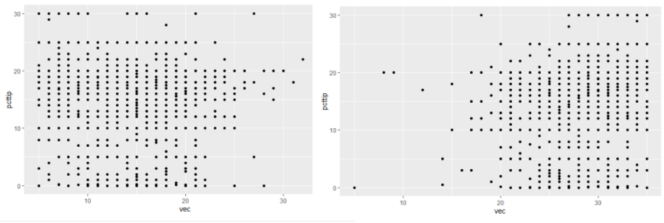
From rom the dataset, there are about 10 variables to describe the servers' personality. For all those kinds factors, values from one to seven are a measure of degree. The outliers are seldom. The method of dealing with the missing data is as the same as above (Effort). The result is shown below.



Just like the literal meaning of features. It is easier to predict that some of features are highly correlated with others. The one who is extraverted _enthusiastic is not likely to be reserved_quiet. The plot below shows the relationship between enthusiastic and other feature



From the plot, we can assume that if the feature has the same trend with "extraverted enthusiastic" then it can be defined as "positive" otherwise it will be defined as "negative ".



The left plot is the tips received by waiter or waitress with positive personality. On the contrary, the right is the graph of negative personality.

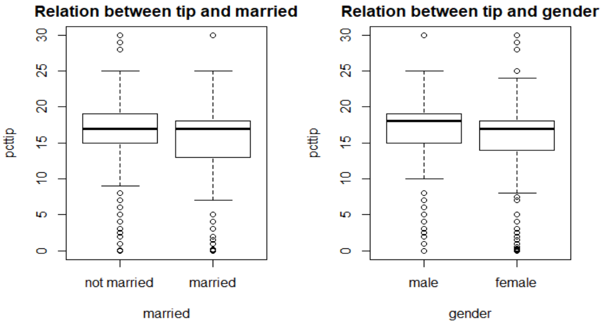
Due to the fact that there is high correlation among the features, it is not necessary to put all of them in the model

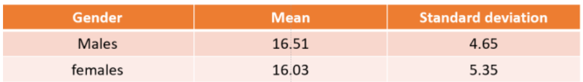
Other ten variables describes the attitude, such as the reserved_quiet, compare with above variables ,have no significant impact on the pcttip, so we will not bring these variables into the model.

2.2. Variables describes the servers

(1)SEX:

First some no meaning values needs to be removed. Compared with married or not, the range of tips for males is larger than the one for females and the median value is higher for males than for females. To determine if the difference between percentage tips is significant, t-test was used at 95% confidence interval. With a p-value of 0.003, we conclude that there is a significant difference between both genders and that the tip average for males is higher than for females. It will be worth adding the SEX feature to a model.



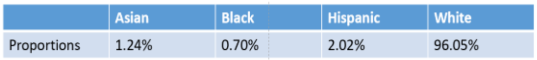


Gender	Mean	Standard deviation
Males	16.51	4.65
females	16.03	5.35

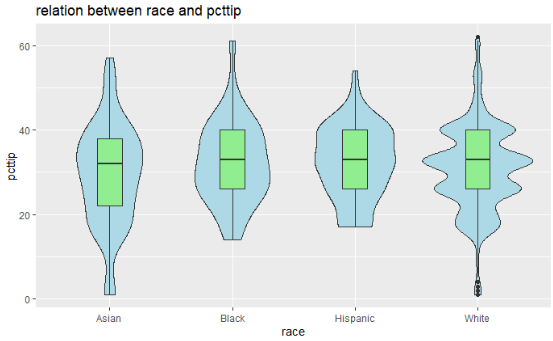
(2)RACE:

From the plot, the average tip for Black and Hispanic servers is much higher than for White. Based on common sense, there is a relationship between race and hair color. By looking at their correlation coefficients, It's better to keep the race and determine the missing value of the race by the color of the hair. At the same time, with more than 96% of the observations from the White population, conclusions will more likely be applicable to White waiters/waitresses.

It will be worth adding RACE to a model

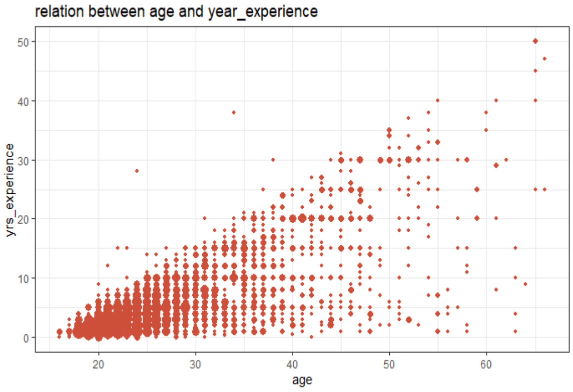


	Asian	Black	Hispanic	White
Proportions	1.24%	0.70%	2.02%	96.05%



(3) AGE,EXPERENCE

There are some servers reported erroneous values for either age or experience because this graph shows for instance someone 34 years old and having been server for 38 years. This graph also shows that most of the servers are between 13 and 30 years of age. To correct the dataset, some impossible data should be removed.

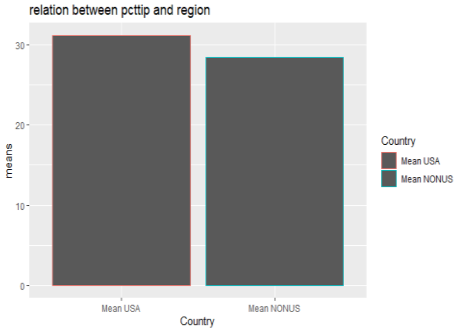


Other variables describes the server, such as the birth year, married, hair,compareed with above variables, have no significant impact on the tip so we will not bring these variables into the model.

2.3. *Variables describes the restaurants and guest*

(1)REGION

Tipping is not a common practice across countries. the average tip in the US is higher than the NONUS area; dinner's tips is more higher than others. It will be worth adding them to a model.



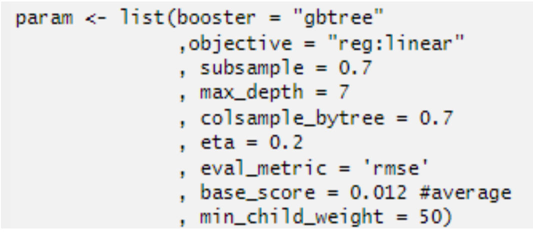
Other nineteen variables describe the restaurants and guest, such as the ethnic type of customers, the proportion of restaurants and the weather .After analysis,these variables have no significant impact on the tips, so we will not bring these variables into the model.

3. Model

Our final purpose is to predict discrete value of tip , about 27 features based on above analysis affect the response a lot . Before building model, we did type coercion to independent variables and make sure the type comes to "numeric" Then we considered to use ensemble models xgboost and random forest and one simple classifier SVM. The method to evaluate model is RMSE.

3.1. Xgboost

The Xgboost is an optimized distributed gradient boosting library designed to be highly efficient It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. The parameter we used to train model is shown as below.

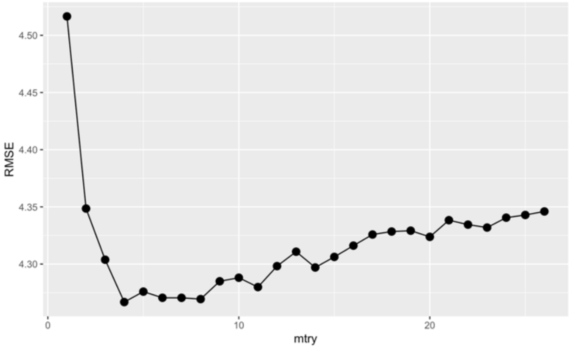


```
param <- list(booster = "gbtree"
              ,objective = "reg:linear"
              , subsample = 0.7
              , max_depth = 7
              , colsample_bytree = 0.7
              , eta = 0.2
              , eval_metric = 'rmse'
              , base_score = 0.012 #average
              , min_child_weight = 50)
```

3.2. Random forest

There are reasons we chose random forests to predict the tips. The main one is we have 26 predictors in total, and some of them, compared to to others, play more important role in predicting than others. Those variables are known as strong predictors. So without the random subset of features in random forest algorithm, we will obtain trees with similar structures, and those trees would also be highly correlated. Another one is, because our dataset is a large one, we need to choose a model with higher speed. Random forest is parallelizable, and will result in faster computation time compared to sequential models such as boosted mode*ls.*

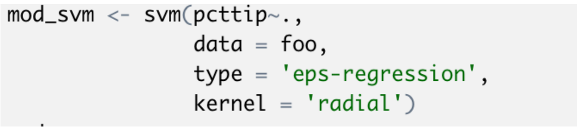
When using random forest model, we set the type of all the predictors into numeric. And delete all those missing values, since that is not applicable in random forest. And use train function to tune the model. We chose the metric RMSE and cross-validation fold 5. After tuning, we decide to set the parameter mtry as 4.



3.3. SVM

Reasons we choose SVM are as follows. One thing is just as the same reason we chose random forest. We have a big dataset and desperately need a model with less computation power. And SVM has this characteristic. Also, SVM models have generalization in practice, the risk of over-fitting is less.

The parameters selected are shown below.

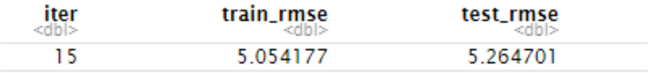


```
mod_svm <- svm(pcttip~.,
               data = foo,
               type = 'eps-regression',
               kernel = 'radial')
```

4. Results.

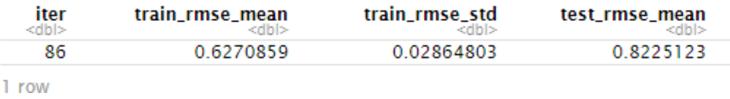
4.1. Xgboost

Only the predictors associated with effort (13 variables) are used to build model. The performance is shown as below.



iter	train_rmse	test_rmse
<dbl>	<dbl>	<dbl>
15	5.054177	5.264701

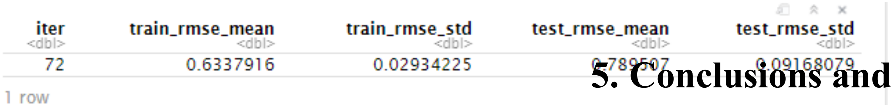
After adding two more variables associated with personality. The performance improves a lot.



iter	train_rmse_mean	train_rmse_std	test_rmse_mean
<dbl>	<dbl>	<dbl>	<dbl>
86	0.6270859	0.02864803	0.8225123

1 row

Then about predictors describing servers is also concerned (26 predictors). The performance improve s a little.

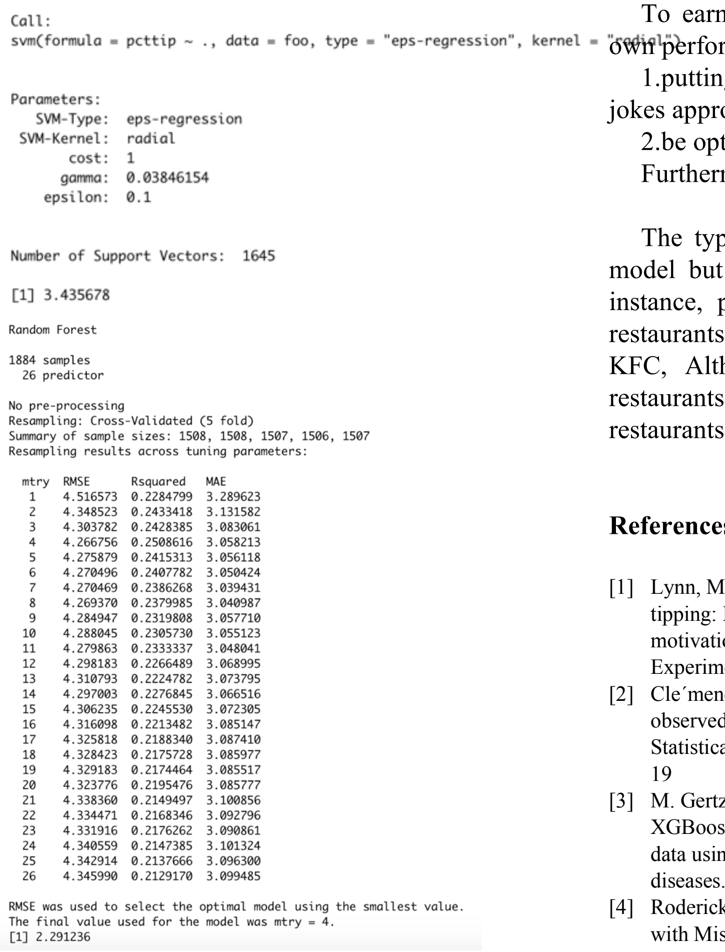


iter <dbl>	train_rmse_mean <dbl>	train_rmse_std <dbl>	test_rmse_mean <dbl>	test_rmse_std <dbl>
72	0.6337916	0.02934225	0.789507	0.09168079

1 row

4.2. Random Forest and SVM

Here is the prediction performance of the two models. From the graph below, we can see that compared with SVM, random forest generates lower testing RMSE. However, it also took more time to train the random forest model.



```
Call:
svm(formula = pcttip ~ ., data = foo, type = "eps-regression", kernel = "radial")


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.03846154
    epsilon:  0.1


Number of Support Vectors:  1645

[1] 3.435678

Random Forest

1884 samples
  26 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 1508, 1508, 1507, 1506, 1507
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
   1    4.516573  0.2284799  3.289623
   2    4.348523  0.2433418  3.131582
   3    4.303782  0.2428385  3.083061
   4    4.266756  0.2508616  3.058213
   5    4.275879  0.2415313  3.056118
   6    4.270496  0.2407782  3.050424
   7    4.270469  0.2386268  3.039431
   8    4.269370  0.2379985  3.040987
   9    4.284947  0.2319808  3.057710
  10    4.288045  0.2305730  3.055123
  11    4.279863  0.2333337  3.048041
  12    4.298183  0.2266489  3.068995
  13    4.310793  0.2224782  3.073795
  14    4.297003  0.2276845  3.066516
  15    4.306235  0.2245530  3.072305
  16    4.316098  0.2213482  3.085147
  17    4.325818  0.2188340  3.087410
  18    4.328423  0.2175728  3.085977
  19    4.329183  0.2174464  3.085517
  20    4.323776  0.2195476  3.085777
  21    4.338360  0.2149497  3.100856
  22    4.334471  0.2168346  3.092796
  23    4.331916  0.2176262  3.090861
  24    4.340559  0.2147385  3.101324
  25    4.342914  0.2137666  3.096300
  26    4.345990  0.2129170  3.099485

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 4.
[1] 2.291236
```

So by comparing these three models, our final model is xgboost with 26 predictors.

5. Conclusions and Futures

we have shown several things a server can control to improve their tip rate as well as other factors they may not be able to control that could affect their tip rate. Attitude does play a significant role in the success of a waiter and as we saw, a large amount of qualities one would classify as positive and negative actually do assist. External factors also have small effect. However, the age ,race ,or hair color are difficult to change.

To earn higher tip, servers should improve their own performance :

1.putting much more effort on their work like talk jokes appropriately

2.be optimism to life.

Furthermore, there are still some jobs can be done..

The types of restaurants are not included in our model but it will also be an important factor. For instance, people prefer to pay more tips in fancy restaurants but few tips in fast food restaurants like KFC, Although the data provided the names of restaurants,there is no proper way to classify restaurants according to their names.

References

[1] Lynn, M. (2015). Explanations for service gratuities and tipping: Evidence from individual differences in tipping motivations and tendencies. Journal of Behavioral and Experimental Economics, 55, 65-71.

[2] Cle´mence Leyrat. Propensity score analysis with partially observed covariates: How should multiple imputation be used? Statistical Methods in Medical Research 2019, Vol. 28(1) 3–19

[3] M. Gertz; K. Große-Butenuth; W. Junge(2020); Using the XGBoost algorithm to classify neck and leg activity sensor data using on-farm health recordings for locomotor-associated diseases. Journal:Computers and Electronics in Agriculture.

[4] Roderick J. A. Little;Donald B. Rubin,Statistical Analysis with Missing Data,2rd ed.;Wiley-Interscience,2002

[5] Mesut Gumus,Mustafa S. Kiran . Crude oil price forecasting using XGBoost[C]//International Conference on Computer Science and Engineering.2017

APPENDIX

Code:

Data process:

~~~
#packages
```{r setup, include=FALSE}
library('ggplot2')
library('VIM')
library('dplyr')
library('readr')
library('stringr')
library('forcats')
library('lubridate')
library('data.table')
library('lattice')
library('MASS')
library('nnet')
library('mice')
~~~

#readdata
```{r}
train<-read.csv('tip.csv',row.names = 1)
glimpse(train)
```

#missing value
```{r}
library(VIM)
aggr(train,prop=F,numbers=T)
aggr(train,prop=F,plot=FALSE,numbers=T) #delete the most
```

#deleting and restructuring
```{r}
train<- subset(train,select=-c(submit_time,more_mos,hair_other,race_other))
train$pcttip <-as.numeric(as.character(train$pcttip))
train$sex<-factor(train$sex,levels=c(0,1),labels=c("male","female"))
a<-which(train$pcttip>30)
b<-which(train$pcttip<=0)
train<-train[-a,]
train<-train[-b,]
#train<-subset(train,train$pcttip!="NA")
train$sex[is.na(train$sex)]<-"female"
train$pcttip[is.na(train$pcttip)]<-17
train$race[train$race=="."|train$race==5]<-4
train$intro[train$intro==5|train$intro=="."]<-4
train$selling[train$selling==5|train$selling=="."]<-4
train$thanks[train$thanks==5|train$thanks=="."]<-1
train$complement[train$complement==5|train$complement=="."]<-3
train$customer_name[train$customer_name==5|train$customer_name=="."]<-2
train$jokes[train$jokes==5|train$jokes=="."]<-2
```

#total look
```{r}
train$pcttip<-as.numeric(train$pcttip)
ggplot(train,aes(x=pcttip, y = ..count..))+geom_bar(stat = 'count',fill='lightblue')+theme_bw()+labs(title="frequency of pcttip")+scale_x_continuous(breaks=seq(0, 200, 5))
qplot(train$pcttip,data=train)
```

#PART1 SERVERS
#sex,hair married
```{r}
test1<-train[,c("pcttip","hair","married","sex")]
test1$hair<-factor(test1$hair,levels=c(1,2,3),labels=c("black", "yellow","golden"))
test1$married<-factor(test1$married,levels=c(0,1),labels=c("not married", "married"))
test1$pcttip <-as.numeric(as.character(test1$pcttip))
par(mfrow=c(1,2))
#boxplot(pcttip ~ hair, data = test1, xlab = "",ylab = "pcttip", main = "Relation between tip and hair")
boxplot(pcttip ~ married, data = test1, xlab = "married",ylab = "pcttip", main = "Relation between tip and married")
boxplot(pcttip ~ sex, data = test1, xlab = "gender",ylab = "pcttip", main = "Relation between tip and gender")

#summary(filter(test1,sex=="male"))
#summary(filter(test1,sex=="female"))
qplot(sex,data=test1,type="histogram",scale="count")

train$sex[is.na(train$sex)]<-"female"

par(mfrow=c(1,2))
ggplot(test1,aes(x=sex,y=pcttip))+geom_boxplot(fill="cornflowerblue",color="black",notch = TRUE)+geom_point(position="jitter",color="blue",alpha=.5)+geom_rug(side="1",color="black")
ggplot(test1,aes(x=married,y=pcttip))+geom_boxplot(fill="cornflowerblue",color="black",notch = TRUE)+geom_point(position="jitter",color="blue",alpha=.5)+geom_rug(side="1",color="black")
```

#choose sex to take a t-test
```{r}
```

~~~
test1<-test1[,c('pcttip','sex')]
ttest1<-melt(test1,measure.var=1,preserve.na=F)
table1<-dcast(ttest1,sex~variable,fun=mean)
table2<-dcast(ttest1,sex~variable,fun=sd)
table1<-cbind(table1,sd=table2[,2])
#table1$sex=factor(table1$sex,levels = c("female","male"))
female <- filter(ttest1,test1$sex=="female")
male <- filter(ttest1,test1$sex=="male")
t.test(female$value,male$value,conf.level = 0.95)
```


#race
```{r}
#look for whites
test2<-train[,c('race','pcttip')]
test2$pcttip <-as.numeric(as.character(test2$pcttip))
test2$race<-factor(test2$race,levels =c(1,2,3,4),labels=c("Asian","Black","Hispanic","White"))
qplot(race,data=test2,type="histogram",scale="count",xlab="ethnical groups")
boxplot(pcttip ~ race, data = test2, xlab = "race",ylab = "pcttip", main = "Relation between tip and race")
mtcars$am <- factor(mtcars$am, levels=c(0,1), labels=c("Automatic", "Manual"))
mtcars$vs <- factor(mtcars$vs, levels=c(0,1), labels=c("V-Engine", "Straight Engine"))
mtcars$cyl <- factor(mtcars$cyl)
test2<-na.omit(test2)
ggplot(test2, aes(x=race, y=pcttip)) +geom_violin(fill="lightblue") +geom_boxplot(fill="lightgreen", width=.2)+labs(title="relation between race and pcttip")

```




#servers’ age and experience
```{r}
test3<-train[,c('sex','birth_yr','race','yrs_experience','pcttip','State')]
test3<-test3[test3$birth_yr %in% 1900:1995, ]
test3$yrs_experience<-as.numeric(as.character(test3$yrs_experience))
test3$birth_yr <-as.numeric(as.character(test3$birth_yr))
test3$pcttip <-as.numeric(as.character(test3$pcttip))
test3$age<-(2006-test3$birth_yr)
test3$age_exp_relation<-test3$age-test3$yrs_experience
ggplot(data=test3,aes(age,yrs_experience))+geom_count(color="tomato3",show.legend = F)+theme_bw()+labs(title="relation between age and year_experience")

#remove
#remove the point
test3<-test3[test3$age_exp_relation > 13 & !is.na(test3$age_exp_relation), ]
test3$race <-factor(test3$race,levels=c(1,2,3,4),labels=c("Asian","Black","Hispanic","White"))
qplot(age,pcttip,data = test3,geom="point")
qplot(yrs_experience,pcttip,data = test3,geom="point")
#进一步分析
qplot(yrs_experience,pcttip,data=test3,geom="point",xlab="years of
experience",ylab="percentage of tip",facets = race~sex)
qplot(age,pcttip,data=test3,geom="point",xlab="age",ylab="percentage of tip",facets =race~sex)
```



#state
```{r}
test4<-test3[,c('State','sex','pcttip','race','age')]
test4$State <-as.character(test4$State)
test4$State <-tolower(test4$State)
States <- c("alabama"="al", "alaska"="ak", "arizona"="az", "arkansas"="ar",
"california"="ca", "colorado"="co", "connecticut"="ct", "delaware"="de", "district of
columbia"="dc", "florida"="fl", "georgia"="ga", "hawaii"="hi", "idaho"="id",
"illinois"="il", "indiana"="in", "iowa"="ia", "kansas"="ks", "kentucky"="ky",
"louisiana"="la", "maine"="me","maryland"="md", "massachusetts"="ma", "michigan"="mi",
"minnesota"="mn", "mississippi"="ms", "missouri"="mo", "montana"="mt",
"nebraska"="ne", "nevada"="nv", "new hampshire"="nh", "new jersey"="nj", "new
mexico"="nm", "new york"="ny", "north carolina"="nc", "north dakota"="nd",
"ohio"="oh", "oklahoma"="ok", "oregon"="or", "pennsylvania"="pa", "rhode island"="ri",
"south carolina"="sc", "south dakota"="sd", "tennessee"="tn", "texas"="tx",
"utah"="ut", "vermont"="vt", "virginia"="va", "washington"="wa", "west virginia"="wv",
"wisconsin"="wi", "wyoming"="wy")
longstates <- test4$State %in% names(States)
test4$longstates<-longstates
test4$longstates<- test4$longstates*1
states2 <- c("al"="al","ak"="ak","az"="az","ar"="ar","ca"="ca","co"="co",
"ct"="ct","de"="de","dc"="dc","fl"="fl","ga"="ga","hi"="hi","id"="id","il"="il","in"="
in","ia"="ia","ks"="ks","ky"="ky","la"="la","me"="me","md"="md","ma"="ma","mi"="mi","m
n"="mn","ms"="ms","mo"="mo","mt"="mt","ne"="ne","nv"="nv","nh"="nh","nj"="nj","nm"="nm
","ny"="ny","nc"="nc","nd"="nd","oh"="oh","ok"="ok","or"="or","pw"="pw","pa"="pa","ri"
="ri","sc"="sc","sd"="sd","tn"="tn","tx"="tx","ut"="ut","vt"="vt","va"="va","wa"="wa",
"wv"="wv","wi"="wi","wy"="wy")
shortstates <- test4$State %in% names(states2)
test4$shortstates<-shortstates
test4$shortstates <- test4$shortstates*1
test4$USstates <- test4$longstates-test4$shortstates
test4$USstates <- (test4$USstates)^2
~~~

~~~~
    test4US<-test4
    test4US<-test4US[test4US$USstates %in% 1,]
    test4NONUS<-test4
    test4NONUS<-test4NONUS[test4NONUS$USstates %in% 0,]
    meanUS<-mean(test4US$pcttip)
    meanNONUS<-mean(test4NONUS$pcttip)
    means<- c(meanUS, meanNONUS)
    Country = c("Mean USA","Mean NONUS")
    Country<-factor(Country, level=c("Mean USA","Mean NONUS"))
    df<-data.frame(Country,means)
    #画图
    ggplot(data=df,aes(x=Country,y=means,color=Country))+geom_bar(stat =
"identity")+labs(title="relation between pcttip and region")
    #qplot(Country, means, geom="histogram", data=df, ylim=c(0,17), ylab="Mean % tip")
    ```

    #trait
    ```{r}
    test6<-
train[,c("pcttip","intro","selling","jokes","customer_name","thanks","complement","sex","race","
asian_prop","black_prop","hispanic_prop","white_prop")]
    test6$pcttip <-as.numeric(as.character(test6$pcttip))
    test6$sex<-factor(test6$sex, levels=c(0,1),labels=c("male","female"))
    qplot(intro,data=test6,type ="histogram", breaks=seq(1, 4, by=0.1), scale="count",
xlab="Intro", main="Introduce themselves(1-Never, 4-Always)")
    qplot(selling,data=test6, type="histogram", breaks=seq(1, 4, by=0.1), scale="count",
xlab="Selling", main="Suggestive Selling(1-Never, 4-Always)")
    qplot(jokes,data=test6, type="histogram", breaks=seq(1, 4, by=0.1), scale="count",
xlab="Jokes", main="Jokes(1-Never, 4-Always)")
    qplot(customer_name,data=test6, type="histogram", breaks=seq(1, 4, by=0.1),
scale="count",xlab ="Customer Name", main="Customer Name(1-Never, 4-Always)")
    qplot(complement,data=test6, type="histogram", breaks=seq(1, 4, by=0.1),
scale="count",xlab ="Complement", main="Complement(1-Never, 4-Always)")
    qplot(thanks,data=test6, type="histogram", breaks=seq(1, 4, by=0.1), scale="count",
xlab="Thanks", main="Thanks(1-Never, 4-Always)")
    ```

    #trait relation
    ```{r}
    traitUS<-train[,c("pcttip","intro","selling","jokes","customer_name","thanks",
    "complement","sex", "race","asian_prop","black_prop","hispanic_prop","white_prop")]
    traitUSm<-melt(traitUS,measure.var=1,preserve.na=FALSE)
    options(digits=3)
    t1<-dcast(traitUSm,intro~variable,mean)
    t2<-dcast(traitUSm,selling~variable,mean)
    t3<-dcast(traitUSm,jokes~variable,mean)
    t4<-dcast(traitUSm,customer_name~variable,mean)
    t5<-dcast(traitUSm,complement~variable,mean)
    t6<-dcast(traitUSm,thanks~variable,mean)
    t1$intro <- factor(t1$intro, levels=c(1,2,3,4), labels=c("Never","Sometimes","Often",
    "Always"))
    t2$selling <- factor(t2$selling, levels=c(1,2,3,4), labels=c("Never","Sometimes"
    ,"Often","Always"))
    t3$jokes<- factor(t3$jokes, levels=c(1,2,3,4), labels=c("Never","Sometimes"
    ,"Often","Always"))
    t4$customer_name <- factor(t4$customer_name , levels=c(1,2,3,4),
    labels=c("Never","Sometimes"
    ,"Often","Always"))
    t5$complement <- factor(t5$complement, levels=c(1,2,3,4), labels=c("Never","Sometimes"
    ,"Often","Always"))
    t6$thanks <- factor(t6$thanks, levels=c(1,2,3,4), labels=c("Never","Sometimes"
    ,"Often","Always"))
    intro<-t1$pcttip
    selling<-t2$pcttip
    jokes<-t3$pcttip
    customer_name<-t4$pcttip
    complement<-t5$pcttip
    thanks<-t6$pcttip
    # Create a vector to assign the labels (selected for trait frequencies above) to each of the six
traits
    # and factor them
    Frequency <- rep(c("Never","Sometimes","Often","Always"),6)
    Frequency <- factor(Frequency,level=c("Never","Sometimes","Often","Always"))
    # Create a vector to assign the trait names to each of the four frequencies and factor them
    Trait <- rep(c("intro","selling","jokes","customer_name","complement","thanks"),4)
    Trait <-
    factor(Trait,level=c("intro","selling","jokes","customer_name","complement","thanks"))
    # Create vectors to repeat each trait four times, each for four different frequencies and
combine
    # these vectors into a matrix to include all traits
    a<-rep("intro",4)
    b<-rep("selling",4)
    c<-rep("jokes",4)
    d<-rep("customer_name",4)
    e<-rep("complement",4)
    f<-rep("thanks",4)
    Trait<-c(a,b,c,d,e,f)
    df<-data.frame(Frequency,Trait,means =
c(intro,selling,jokes,customer_name,complement,thanks))
    ggplot(df,aes(x=Frequency,y=means,colour=Trait,group=Trait))+geom_line()

    intro_sellingUS<-dcast(traitUSm,intro+selling+sex~variable,mean)

    introm<-intro_sellingUS$male$intro
    sellingm<-intro_sellingUS$male$selling
    #is_tipm<-intro_sellingUS$male$pcttip
~~~~

~~~~
#df <- data.frame(introm=factor(introm),sellingm=factor(sellingm),is_tipm)
qplot(sellingm,is_tipm,data=df,geom="line")
```

#customers
#ethnicity of customers
```{r}
test8<-train[,c('asian_prop','black_prop','hispanic_prop','white_prop','pcttip')]
test8$asian_prop <-as.numeric(as.character(test8$asian_prop))
test8$black_prop <-as.numeric(as.character(test8$black_prop))
test8$hispanic_prop<-as.numeric(as.character(test8$hispanic_prop))
test8$white_prop <-as.numeric(as.character(test8$white_prop))
test8$pcttip <-as.numeric(as.character(test8$pcttip))
test8[test8>100]<-100
ggplot(test8, aes(x=asian_prop, y=pcttip))+geom_point()
ggplot(test8, aes(x=white_prop, y=pcttip))+geom_point()
ggplot(test8, aes(x=black_prop, y=pcttip))+geom_point()
ggplot(test8, aes(x=hispanic_prop, y=pcttip))+geom_point()
```
#when
```{r}
test9<-train[,c('pcttip','breakfast','lunch','dinner','late_night')]
test9$pcttip<-as.numeric(test9$pcttip)
ttest9<-melt(test9,measure.var=1,preserve.na=FALSE)
ttest9$value<-as.numeric(ttest9$value)
t1<-dcast(ttest9,breakfast~variable,mean)
t2<-dcast(ttest9,lunch~variable,mean)
t3<-dcast(ttest9,dinner~variable,mean)
t4<-dcast(ttest9,late_night~variable,mean)
mean<-c(t1[2,2],t2[2,2],t3[2,2],t4[2,2])
wh<-c("breakfast","lunch","dinner","latenight")
ch<-c(rep("yes",4))
df1<-as.data.frame(cbind(mean,wh,ch))
df1$mean<-as.numeric(as.character(df1$mean))
ggplot(df1, aes(x=wh,y=mean)) + geom_point()
```

#dinner
```{r}
test9<-train[,c('pcttip','breakfast','dinner','late_night')]

test1<-test1[,c('pcttip','sex')]
ttest1<-melt(test1,measure.var=1,preserve.na=F)
table1<-dcast(ttest1,sex~variable,fun=mean)
table2<-dcast(ttest1,sex~variable,fun=sd)
table1<-cbind(table1,sd=table2[,2])
#table1$sex=factor(table1$sex,levels = c("female","male"))
female <- filter(ttest1,test1$sex=="female")
male <- filter(ttest1,test1$sex=="male")
t.test(female$value,male$value,conf.level = 0.95)
```

#busy
```{r}
test10<-train[,c("pcttip","busy")]
qplot(busy,data=test10,type="histogram",scale="count")
test10[test10=="."]<-2
test10$pcttip <-as.numeric(as.character(test10$pcttip))
ttest10<-melt(test10,measure.var=1,preserve.na=FALSE)
ttest10$value<-as.numeric(ttest10$value)
#missing value
ttest10[is.na(ttest10)] <- 10
t1<-dcast(ttest10,busy~variable,mean)
barplot(t1[,2],names.arg = t1[,1],xlab="",ylab="pcttip",col="lightblue",
main="relationship between tip and busy",border="red")
```
#effect
```{r}

test18<-train[,c("pcttip","effect_sz")]
qplot(effect_sz,data=test18,type="histogram",scale="count")
test18[test18=="."]<-3
test18$pcttip <-as.numeric(as.character(test18$pcttip))
ttest18<-melt(test18,measure.var=1,preserve.na=FALSE)
ttest18$value<-as.numeric(ttest18$value)
#missing value
t1<-dcast(ttest18,effect_sz~variable,mean)
barplot(t1[,2],names.arg = t1[,1],xlab="",ylab="pcttip",col="lightblue",
main="relationship between tip and effect_sz",border="red")

```

#kind1
```{r}
    test11<-
train[,c("pcttip","Men","Women","Teenagers","Young_Adults","Middle_Aged_Adults","Elderly
_Adults")]
    test11$Men<-factor(test11$Men,levels=c(1,2,3,99),labels=c("Below Average", "Average",
"Above Average", "Don't Know"))
    test11$Women<-factor(test11$Women, levels=c(1,2,3,99),labels=c("Below Average",
"Average", "Above Average", "Don't Know"))
    test11$Teenagers<-factor(test11$Teenagers,levels=c(1,2,3,99),labels=c("Below Average",
"Average", "Above Average", "Don'tKnow"))
    test11$Young_Adults<-factor(test11$Young_Adults,levels=c(1,2,3,99), labels=c("Below
Average", "Average", "Above Average", "Don't Know"))
~~~~

~~~~
test11$Middle_Aged_Adults<-factor(test11$Middle_Aged_Adults,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test11$Elderly_Adults<-factor(test11$Elderly_Adults,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test11$pcttip <-as.numeric(as.character(test11$pcttip))
par(mfrow=c(2,3))
boxplot(pcttip ~ Men, data = test11, xlab = "",ylab = "pcttip", main = "Relation between tip and Men")
boxplot(pcttip ~ Women, data = test11, xlab = "",ylab = "pcttip", main = "Relation between tip and Women")
boxplot(pcttip ~ Teenagers, data = test11, xlab = "",ylab = "pcttip", main = "Relation between tip and Teenage")
boxplot(pcttip ~ Young_Adults, data = test11, xlab = "",ylab = "pcttip", main = "Relation between tip and young_adults")
boxplot(pcttip ~ Middle_Aged_Adults, data = test11, xlab = "",ylab = "pcttip", main = "Relation between tip and middle_age_adult")
boxplot(pcttip ~ Elderly_Adults, data = test11, xlab = "",ylab = "pcttip", main = "Relation between tip and elderly_adulta")

```

#kind2
```{r}
test12<-train[,c("pcttip","Couples","onetops","kids","Business_People")]
test12$Couples<-factor(test12$Couples,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test12$onetops<-factor(test12$onetops, levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test12$kids<-factor(test12$kids,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don'tKnow"))
test12$Business_People<-factor(test12$Business_People,levels=c(1,2,3,99), labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test12$pcttip <-as.numeric(as.character(test12$pcttip))
par(mfrow=c(1,4))
boxplot(pcttip ~ Couples, data = test12, xlab = "",ylab = "pcttip", main = "Relation between tip and Couples")
boxplot(pcttip ~ onetops, data = test12, xlab = "",ylab = "pcttip", main = "Relation between tip and onetops")
boxplot(pcttip ~ kids, data = test12, xlab = "",ylab = "pcttip", main = "Relation between tip and kids")
boxplot(pcttip ~ Business_People, data = test12, xlab = "",ylab = "pcttip", main = "Relation between tip and Business_People")

```

#kind3
```{r}
test13<-train[,c("pcttip","Regulars","First_Timers")]
test13$Regulars<-factor(test13$Regulars,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test13$First_Timers<-factor(test13$First_Timers,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test13$pcttip <-as.numeric(as.character(test13$pcttip))
par(mfrow=c(1,2))
boxplot(pcttip ~ Regulars, data = test13, xlab = "",ylab = "pcttip", main = "Relation between tip and Rugulars")
boxplot(pcttip ~ First_Timers, data = test13, xlab = "",ylab = "pcttip", main = "Relation between tip and First_Timers")
```

#kind4
```{r}
test14<-train[,c("pcttip","Cash_Customers","Charge_Customers")]
test14$Cash_Customers<-factor(test14$Cash_Customers,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test14$Charge_Customers<-factor(test14$Charge_Customers,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test14$pcttip <-as.numeric(as.character(test14$pcttip))
par(mfrow=c(1,2))
boxplot(pcttip ~ Cash_Customers, data = test14, xlab = "",ylab = "pcttip", main = "Relation between tip and Cash_Customers")
boxplot(pcttip ~ Charge_Customers, data = test14, xlab = "",ylab = "pcttip", main = "Relation between tip and Charge_Customers")
```

#kind5
```{r}
test15<-train[,c("pcttip","Asians","Blacks","Hispanics","Whites","Foreigners")]
test15$Asians<-factor(test15$Asians,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test15$Blacks<-factor(test15$Blacks, levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test15$Hispanics<-factor(test15$Hispanics,levels=c(1,2,3,99),labels=c("Below Average", "Average", "Above Average", "Don'tKnow"))
test15$Whites<-factor(test15$Whites,levels=c(1,2,3,99), labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test15$Foreigners<-factor(test15$Foreigners,levels=c(1,2,3,99), labels=c("Below Average", "Average", "Above Average", "Don't Know"))
test15$pcttip <-as.numeric(as.character(test15$pcttip))
par(mfrow=c(1,5))
~~~~

~~~
boxplot(pcttip ~ Asians, data = test15, xlab = "",ylab = "pcttip", main = "Relation between tip and Asians")
boxplot(pcttip ~ Blacks, data = test15, xlab = "",ylab = "pcttip", main = "Relation between tip and Blacks")
boxplot(pcttip ~ Hispanics, data = test15, xlab = "",ylab = "pcttip", main = "Relation between tip and Hispanics")
boxplot(pcttip ~ Whites, data = test15, xlab = "",ylab = "pcttip", main = "Relation between tip and Whites")
boxplot(pcttip ~ Foreigners, data = test15, xlab = "",ylab = "pcttip", main = "Relation between tip and Foreigners")
```

#kind6
```{r}
test17<-train[,c("pcttip","Extraverted_enthusiastic","Critical_quarrelsome","Anxious_easily_upset","Dependable_selfdisciplined","Open_to_new_experiences_complex","Reserved_quiet","Sympathetic_warm","Disorganized_careless","Calm_emotionally_stable","Conventional_uncreative")]
test17$pcttip <-as.numeric(as.character(test17$pcttip))
par(mfrow=c(2,5))

boxplot(pcttip ~ Extraverted_enthusiastic, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Extraverted_enthusiastic")
boxplot(pcttip ~ Critical_quarrelsome, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Critical_quarrelsome")
boxplot(pcttip ~ Anxious_easily_upset, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Anxious_easily_upset")
boxplot(pcttip ~ Dependable_selfdisciplined, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Dependable_selfdisciplined")
boxplot(pcttip ~ Open_to_new_experiences_complex, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Open_to_new_experiences_complex")
boxplot(pcttip ~ Reserved_quiet, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Reserved_quiet")
boxplot(pcttip ~ Sympathetic_warm, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Sympathetic_warm")

boxplot(pcttip ~ Disorganized_careless, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Disorganized_careless")
boxplot(pcttip ~ Calm_emotionally_stable, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Calm_emotionally_stable")
boxplot(pcttip ~ Conventional_uncreative, data = test17, xlab = "",ylab = "pcttip", main = "Relation between tip and Conventional_uncreative")
```
~~~

MODEL

~~~
```{r}
dat<-read.csv('tip.csv')
Train<-dat
Train<-unique(Train)
a<-which(Train$pcttip>30)
b<-which(Train$pcttip<=0)
Train[a,23]=30
Train[b,23]=0
data<-subset(Train,Train$pcttip!="NA")
q<-data
d<-cbind(data$pcttip,data$flair,data$intro,data$selling,data$touch,data$jokes,data$repeat.,data$customer_name,data$draw,data$smile,data$thanks,data$weather,data$complement)
colnames(d) <- c("pcttip","flair","intro","selling","touch","jokes","repeat","customer_name","draw","smile","thanks","weather","complement")
d<-as.data.frame(d)
aggr(d,prop=FALSE,numbers=TRUE)
d$pcttip<- NULL
a<-which((rowSums(x = is.na(x = d)) == ncol(x = d)))
pc<-q[-a,]
Data<-d[!(rowSums(x = is.na(x = d)) == ncol(x = d)),]
pctip<-pc$pcttip
data1<-data.frame(Data,pctip)
dd<-data.frame(Data,pctip)
data1$pctip<-cut(data1$pctip,c(0,10,15,20,30),c("1-10","11-15","16-20","21-30"))
library(lattice)
library(MASS)
library(nnet)
library(mice) #前三个包是mice的基础
imp=mice(data1,m=4,method="rf") #4重插补，即生成4个无缺失数据集
f_data<-complete(imp)

summary(imp)

densityplot(imp)
names(f_data)[13]<-c("pcttip")
names(dd)[13]<-c("pcttip")
f_data$pcttip<-dd$pcttip

library(ggplot2)
ggplot(dd,aes(x=dd$pcttip)) +
geom_histogram(colour="black",fill = "blue", bins = 60, position="stack", show.legend = TRUE)
min(f_data$intro)
pc[26:28]<-f_data[1:3]
pc[30:38]<-f_data[4:12]

```
~~~

~~~~
```{r}
attitude<-pc[,c("pcttip","Extraverted_enthusiastic","Critical_quarrelsome","Anxious_easily_upset","Dependable_selfdisciplined","Open_to_new_experiences_complex","Reserved_quiet","Sympathetic_warm","Disorganized_careless","Calm_emotionally_stable","Conventional_uncreative")]

imp=mice(attitude,m=4,method="rf") #4重插补，即生成4个无缺失数据集
f_data1<-complete(imp)
summary(imp)
densityplot(imp)
names(f_data1)[1]<-c("pcttip")
names(dd)[13]<-c("pcttip")
f_data1$pcttip<-dd$pcttip
grp_df<-f_data1%>%group_by(Extraverted_enthusiastic)%>%summarise(avg_dur=mean(Critical_quarrelsome),Depend=mean(Dependable_selfdisciplined),Open=mean(Open_to_new_experiences_complex),reser=mean(Reserved_quiet),sym=mean(Sympathetic_warm),dis=mean(Disorganized_careless),clam=mean(Calm_emotionally_stable),conv=mean(Conventional_uncreative),Anxious_easily_upset=mean(Anxious_easily_upset))

par(mfrow=c(2,5))
plot(Critical_quarrelsome ~ Extraverted_enthusiastic, data=f_data1)
z <- lm(Critical_quarrelsome ~ Extraverted_enthusiastic, data = f_data1)
abline(z)
plot( Dependable_selfdisciplined~ Extraverted_enthusiastic, data=f_data1)
z <- lm( Dependable_selfdisciplined ~ Extraverted_enthusiastic, data = f_data1)
abline(z)
plot(Open_to_new_experiences_complex ~ Extraverted_enthusiastic, data=f_data1)
z <- lm(Open_to_new_experiences_complex ~ Extraverted_enthusiastic, data = f_data1)
abline(z)
plot(Reserved_quiet ~ Extraverted_enthusiastic, data=f_data1)
z <- lm(Reserved_quiet ~ Extraverted_enthusiastic, data = f_data1)
abline(z)
plot(Sympathetic_warm ~ Extraverted_enthusiastic, data=f_data1)
z <- lm(Sympathetic_warm ~ Extraverted_enthusiastic, data = f_data1)
abline(z)
plot(Disorganized_careless ~ Extraverted_enthusiastic, data=f_data1)
z <- lm(Disorganized_careless ~ Extraverted_enthusiastic, data = f_data1)
abline(z)
plot(Calm_emotionally_stable ~ Extraverted_enthusiastic, data=f_data1)
z <- lm(Calm_emotionally_stable ~ Extraverted_enthusiastic, data = f_data1)
abline(z)
plot(Conventional_uncreative ~ Extraverted_enthusiastic, data=f_data1)
z <- lm(Conventional_uncreative ~ Extraverted_enthusiastic, data = f_data1)
abline(z)
plot(Anxious_easily_upset ~ Extraverted_enthusiastic, data=f_data1)
z <- lm(Anxious_easily_upset ~ Extraverted_enthusiastic, data = f_data1)
abline(z)

regression<-lm(pcttip~., data=f_data1)
summary(regression)

pos<-cbind(f_data1$Extraverted_enthusiastic,f_data1$Dependable_selfdisciplined,f_data1$Open_to_new_experiences_complex,f_data1$Calm_emotionally_stable,f_data1$Sympathetic_warm)
vec<-apply(pos,1,sum)
ggplot(f_data1,aes(x=vec,y=pcttip)) +geom_point()

neg<-cbind(f_data1$Critical_quarrelsome,f_data1$Anxious_easily_upset,f_data1$Reserved_quiet,f_data1$Disorganized_careless,f_data1$Conventional_uncreative)
vec<-apply(neg,1,sum)
ggplot(f_data1,aes(x=vec,y=pcttip)) +geom_point()

pc[,c("pcttip","Extraverted_enthusiastic","Critical_quarrelsome","Anxious_easily_upset","Dependable_selfdisciplined","Open_to_new_experiences_complex","Reserved_quiet","Sympathetic_warm","Disorganized_careless","Calm_emotionally_stable","Conventional_uncreative")]<-f_data1

```


```{r}
test1<-pc
#glimpse(train)
```

```{r}
#aggr(test1,prop=F,plot=FALSE,numbers=T) #delete the most
#aggr(test1,prop=F,numbers=T)
#cor(na.omit(test1))
test1$hair_other<-str_to_lower(test1$hair_other)
test1$hair[test1$hair_other=='black']<-4
test1$race[test1$race==5|test1$race==30]<-NA
test1$hair[test1$hair==30]<-NA
test1$race<-factor(test1$race,levels=c(1,2,3,4),labels=c("Asian","Black","Hispanic","White"))
test1$hair<-factor(test1$hair,levels=c(1,2,3,4),labels=c("yellow", "brown","blond","black"))
table(paste(test1$hair,test1$race))
if(is.na(test1$race)==TRUE){test1$race[test1$hair=='black']<-'Asian'}
if(is.na(test1$race)==TRUE){test1$race[test1$hair=='blond'|test1$hair=='brown'|test1$hair=='yellow']<-'White'}
if(is.na(test1$hair)==TRUE){test1$hair[test1$race=='Asian']<-'black'}
if(is.na(test1$hair)==TRUE){test1$hair[test1$race=='White'|test1$race=='Black'|test1$race=='Hispanic']<-'brown'}
~~~~

~~~
test1$sex[test1$sex==0|test1$sex==30|is.na(test1$sex)]<-1
#train$sex<-factor(train$sex,levels=c(0,1),labels=c("male","female"))
```

```{r}
test1$yrs_experience<-as.numeric(as.character(test1$yrs_experience))
test1$birth_yr <-as.numeric(as.character(test1$birth_yr))
test1$age<-(2006-test1$birth_yr)
test1$age_exp_relation<-test1$age-test1$yrs_experience
test1<-test1[test1$age_exp_relation > 13 & !is.na(test1$age_exp_relation), ]
test1<-test1[test1$age<100, ]
qplot(age,yrs_experience,data=test1,geom="point")
```


```{r}
test1$hair[is.na(test1$hair)]<-'brown'
test1$race[is.na(test1$race)]<-'White'
qplot(age,married,data=test1)
test1$married[test1$married==2]<-NA
for (i in 1:nrow(test1)) {
  if(is.na(test1$married[i]==TRUE)){test1$married[i]<-rbinom(1, 1, 1/2)}
}

test1<-test1[,c(-78,-75)]
write.csv(test1,"whywhy.csv")
```



```{r}
test2<-read.csv('whywhy.csv')
test2$State <-as.character(test2$State)
test2$State <-tolower(test2$State)
States <- c("alabama"="al", "alaska"="ak", "arizona"="az", "arkansas"="ar",
"california"="ca", "colorado"="co", "connecticut"="ct", "delaware"="de", "district of
columbia"="dc", "florida"="fl", "georgia"="ga", "hawaii"="hi", "idaho"="id",
"illinois"="il", "indiana"="in", "iowa"="ia", "kansas"="ks", "kentucky"="ky",
"louisiana"="la",             "maine"="me","maryland"="md",             "massachusetts"="ma",
"michigan"="mi",
"minnesota"="mn", "mississippi"="ms", "missouri"="mo", "montana"="mt",
"nebraska"="ne", "nevada"="nv", "new hampshire"="nh", "new jersey"="nj", "new
mexico"="nm", "new york"="ny", "north carolina"="nc", "north dakota"="nd",
"ohio"="oh", "oklahoma"="ok", "oregon"="or", "pennsylvania"="pa", "rhode island"="ri",
"south carolina"="sc", "south dakota"="sd", "tennessee"="tn", "texas"="tx",
"utah"="ut", "vermont"="vt", "virginia"="va", "washington"="wa", "west virginia"="wv",
"wisconsin"="wi", "wyoming"="wy")
longstates <- test2$State %in% names(States)
test2$longstates<-longstates
test2$longstates<- test2$longstates*1
states2 <- c("al"="al","ak"="ak","az"="az","ar"="ar","ca"="ca","co"="co",
"ct"="ct","de"="de","dc"="dc","fl"="fl","ga"="ga","hi"="hi","id"="id","il"="il","in"="
in","ia"="ia","ks"="ks","ky"="ky","la"="la","me"="me","md"="md","ma"="ma","mi"="mi"
,"m
n"="mn","ms"="ms","mo"="mo","mt"="mt","ne"="ne","nv"="nv","nh"="nh","nj"="nj","nm
"="nm
","ny"="ny","nc"="nc","nd"="nd","oh"="oh","ok"="ok","or"="or","pw"="pw","pa"="pa","r
i"
="ri","sc"="sc","sd"="sd","tn"="tn","tx"="tx","ut"="ut","vt"="vt","va"="va","wa"="wa",
"wv"="wv","wi"="wi","wy"="wy")
shortstates <- test2$State %in% names(states2)
test2$shortstates<-shortstates
test2$shortstates <- test2$shortstates*1
test2$USstates <- test2$longstates-test2$shortstates
test2$USstates <- (test2$USstates)^2
test2US<-test2
test2US<-test2US[test2US$USstates %in% 1,]
test2NONUS<-test2
test2NONUS<-test2NONUS[test2NONUS$USstates %in% 0,]
meanUS<-mean(test2US$pcttip)
meanNONUS<-mean(test2NONUS$pcttip)
means<- c(meanUS, meanNONUS)
Country = c("Mean USA","Mean NONUS")
Country<-factor(Country, level=c("Mean USA","Mean NONUS"))
df<-data.frame(Country,means)
#画图
ggplot(data=df,aes(x=Country,y=means,color=Country))+geom_bar(stat = "identity")
#qplot(Country, means, geom="histogram", data=df, ylim=c(0,17), ylab="Mean % tip")
test2<-test2[ ,c(-81,-80,-1,-2,-4,-6,-7,-8,-9)]
test2<-test2[ ,c(-2,-3,-4,-5,-6)]
write.csv(test2,"whywhywhy.csv")
table(test2$married)
```

```{r}
library("dplyr")
library("stringr")
library("ggplot2")
library("VIM")
library("Rmisc")
library("Matrix")
library("xgboost")
library("caret")
library("lubridate")
```
~~~

~~~~
```{r}

train<-read.csv("features.csv")
train$hair<-as.integer(train$hair)
#train$race<-as.integer(train$race)

```

```{r}

set.seed(4321)
trainIndex <- createDataPartition(train$pcttip, p = 0.8, list = FALSE, times = 1)
train <- train[trainIndex,]
valid <- train[-trainIndex,]
foo <- train
bar <- valid
dtrain <- xgb.DMatrix(as.matrix(foo),label = train$pcttip)
dtest <- xgb.DMatrix(as.matrix(bar),label = valid$pcttip)
```

```{r}



param <- list(booster = "gbtree"
              ,objective = "reg:linear"
              , subsample = 0.7
              , max_depth = 7
              , colsample_bytree = 0.7
              , eta = 0.2
              , eval_metric = 'rmse'
              , base_score = 0.012 #average
              , min_child_weight = 50)
foldsCV <- createFolds(f_data$pcttip, k=7, list=TRUE, returnTrain=FALSE)

xgb_cv <- xgb.cv(dtrain,
                params=param,
                nrounds=100,
                prediction=TRUE,
                maximize=FALSE,
                folds=foldsCV,
                early_stopping_rounds = 30,
                print_every_n = 5
)
print(xgb_cv$evaluation_log[which.min(xgb_cv$evaluation_log$test_rms)])

```



```{r}
foo <- na.omit(foo)
bar<-na.omit(bar)
mod_rf = train(pcttip ~ .,
               data = foo,method = "rf",
               tuneGrid = data.frame(mtry=1:26),
               trControl = trainControl(method="cv", number=5),
               metric = "RMSE")
mod_rf

ggplot(mod_rf$results, aes(x=mtry, y=RMSE)) +
  geom_point(size=3) +
  xlab("mtry") + geom_line()
mod_rf$bestTune
final<- mod_rf$finalModel


pred_rf <- predict(final, newdata = bar, type = "response")
RMSE(pred_rf, bar$pcttip)
```

```{r}
library(e1071)
mod_svm <- svm(pcttip~.,
               data = foo,
               type = 'eps-regression',
               kernel = 'radial')
mod_svm
pred_svm = predict(mod_svm, newdata = bar)
RMSE(pred_svm, bar$pcttip)
```
~~~~