Data Mining - Assignment 1

Liju Robin George

January 11, 2017

Data Description

The data that we have is of customers of direct marketing campaigns by a marketer. The marketer wishes to mine the data to learn what features or characteristics drive some customers to spend more than the others. These records include a customer's age (coded as young, middle, and old), gender (female/male), whether the customer owns or rents a home,is single or married, the location of the customer relative to the nearest brick-and-mortar (coded as far or close),the customer's salary, and how many children the customer has (between 0 and 3). The marketer also records the customer's past purchasing history (coded as low, medium, or high, or NA if the customer has not purchased anything in the past),the number of catalogs sent to that customer, and the amount of money the customer has spent.

1.Response and Predictor variables

```r
variable.names(train)
```

```
##  [1] "Age"         "Gender"      "OwnHome"     "Married"     "Location"
##  [6] "Salary"      "Children"    "History"     "Catalogs"    "AmountSpent"
```

As, discussed above, we need to explain the AmountSpent in terms of the customer's characteristics. Therefore, our response variable is: "AmountSpent" - Numerical Our predictors are: "Age" - categorical
"Gender" - Categorical
"OwnHome" - Categorical
"Married" - Categorical "Location" - Categorical
"Salary" - Numerical
"Children" - Numerical
"History" - Categorical
"Catalogs" - Numerical

Let us have a look at the descriptive statistics

```r
summary(train)
```

```
##      Age         Gender    OwnHome       Married     Location
##  Middle:508   Female:506   Own :516   Married:502   Close:710
##  Old   :205   Male  :494   Rent:484   Single :498   Far  :290
##  Young :287
##
##
##
##      Salary          Children       History       Catalogs
##  Min.   : 10100   Min.   :0.000   High  :255   Min.   : 6.00
##  1st Qu.: 29975   1st Qu.:0.000   Low   :230   1st Qu.: 6.00
##  Median : 53700   Median :1.000   Medium:212   Median :12.00
##  Mean   : 56104   Mean   :0.934   NA's  :303   Mean   :14.68
```

```
##  3rd Qu.: 77025   3rd Qu.:2.000                 3rd Qu.:18.00
##  Max.   :168800   Max.   :3.000                 Max.   :24.00
##   AmountSpent
##  Min.   :  38.0
##  1st Qu.: 488.2
##  Median : 962.0
##  Mean   :1216.8
##  3rd Qu.:1688.5
##  Max.   :6217.0
```

```r
dim(train)
```

```
## [1] 1000   10
```

2.Descriptive and Graphical Statistics

a. Cleaning the data

We can observe from the above summary that there is 303 NAs in the History variable. Let us go ahead and create a variable train.clean by converting NA to None and then to the lowest factor level in the History variable.

```r
train.clean = train
train.clean$History<- factor (train.clean$History)
train.clean$History<- as.character(train.clean$History)
train.clean$History[is.na(train.clean$History)] <- "None"

train.clean$History <- factor (train.clean$History,
                               levels=c("None","Low","Medium","High"))
levels(train.clean$History)
```

```
## [1] "None"   "Low"    "Medium" "High"
```

```r
attach(train.clean)
```

b. Data summary after cleaning

```r
summary(train.clean)
```

```
##      Age         Gender     OwnHome       Married     Location
##  Middle:508   Female:506   Own :516   Married:502   Close:710
##  Old   :205   Male  :494   Rent:484   Single :498   Far  :290
##  Young :287
##
##
##
##      Salary          Children       History       Catalogs
##  Min.   : 10100   Min.   :0.000   None  :303   Min.   : 6.00
##  1st Qu.: 29975   1st Qu.:0.000   Low   :230   1st Qu.: 6.00
```

```
##  Median : 53700   Median :1.000   Medium:212   Median :12.00
##  Mean   : 56104   Mean   :0.934   High  :255   Mean   :14.68
##  3rd Qu.: 77025   3rd Qu.:2.000                3rd Qu.:18.00
##  Max.   :168800   Max.   :3.000                Max.   :24.00
##   AmountSpent
##  Min.   :  38.0
##  1st Qu.: 488.2
##  Median : 962.0
##  Mean   :1216.8
##  3rd Qu.:1688.5
##  Max.   :6217.0
```

```
dim(train.clean)
```

```
## [1] 1000   10
```

Let us now look at the descriptive statistics for each of the numerical variables.

```
#Standard Deviation
standardDevs = c(Salary = sd(train.clean$Salary), Children = sd(train.clean$Children),
                 Catalogs = sd(train.clean$Catalogs), AmountSpent = sd(train.clean$AmountSpent))
standardDevs
```

```
##       Salary     Children     Catalogs  AmountSpent
## 30616.314826     1.051070     6.622895   961.068613
```

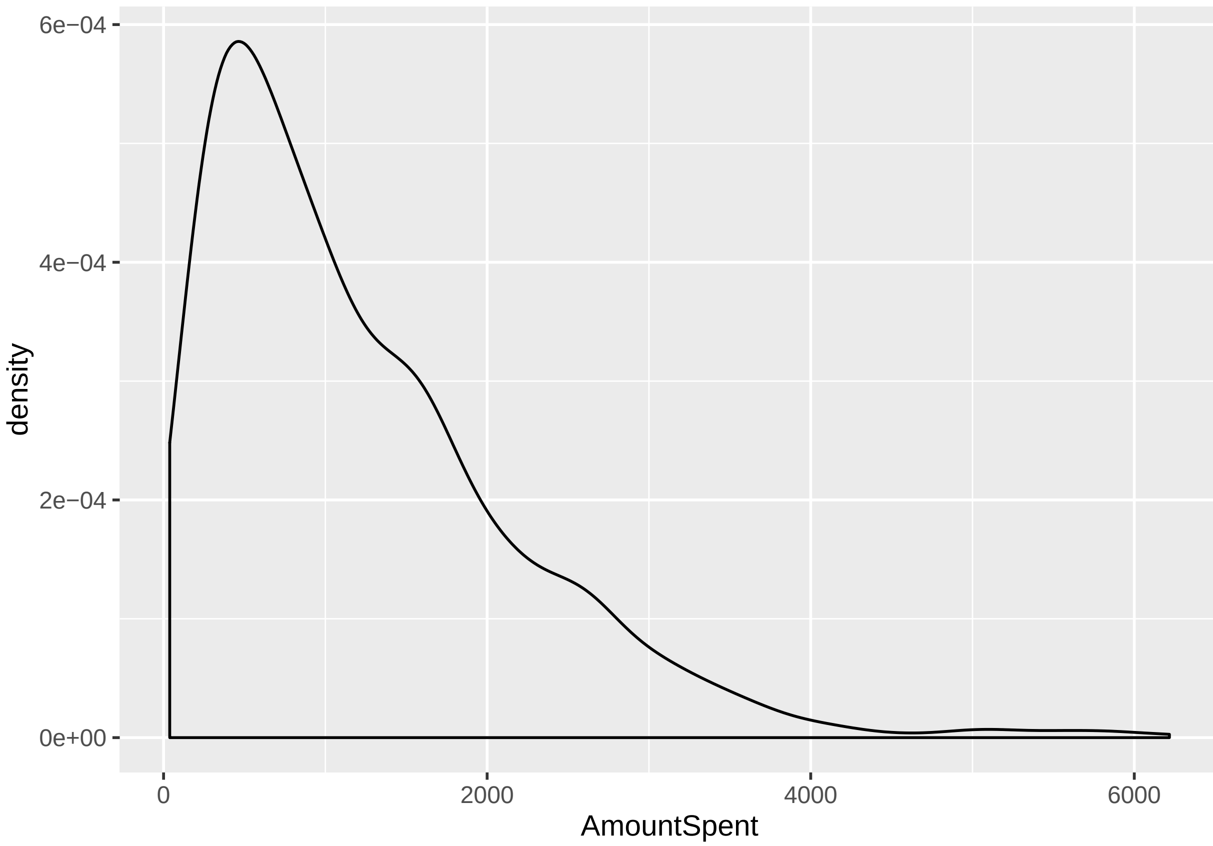
```
#Summary
totSummary = c(Salary = summary(train.clean$Salary), Children = summary(train.clean$Children),
               Catalogs = summary(train.clean$Catalogs), AmountSpent = summary(train.clean$AmountSpent))
totSummary
```

```
##         Salary.Min.      Salary.1st Qu.       Salary.Median
##           1.010e+04           2.998e+04           5.370e+04
##         Salary.Mean      Salary.3rd Qu.         Salary.Max.
##           5.610e+04           7.702e+04           1.688e+05
##       Children.Min.    Children.1st Qu.     Children.Median
##           0.000e+00           0.000e+00           1.000e+00
##       Children.Mean    Children.3rd Qu.       Children.Max.
##           9.340e-01           2.000e+00           3.000e+00
##       Catalogs.Min.    Catalogs.1st Qu.     Catalogs.Median
##           6.000e+00           6.000e+00           1.200e+01
##       Catalogs.Mean    Catalogs.3rd Qu.       Catalogs.Max.
##           1.468e+01           1.800e+01           2.400e+01
##    AmountSpent.Min. AmountSpent.1st Qu.  AmountSpent.Median
##           3.800e+01           4.882e+02           9.620e+02
##    AmountSpent.Mean AmountSpent.3rd Qu.    AmountSpent.Max.
##           1.217e+03           1.688e+03           6.217e+03
```

c. Density plots for AmountSpent and Salary

AmountSpent

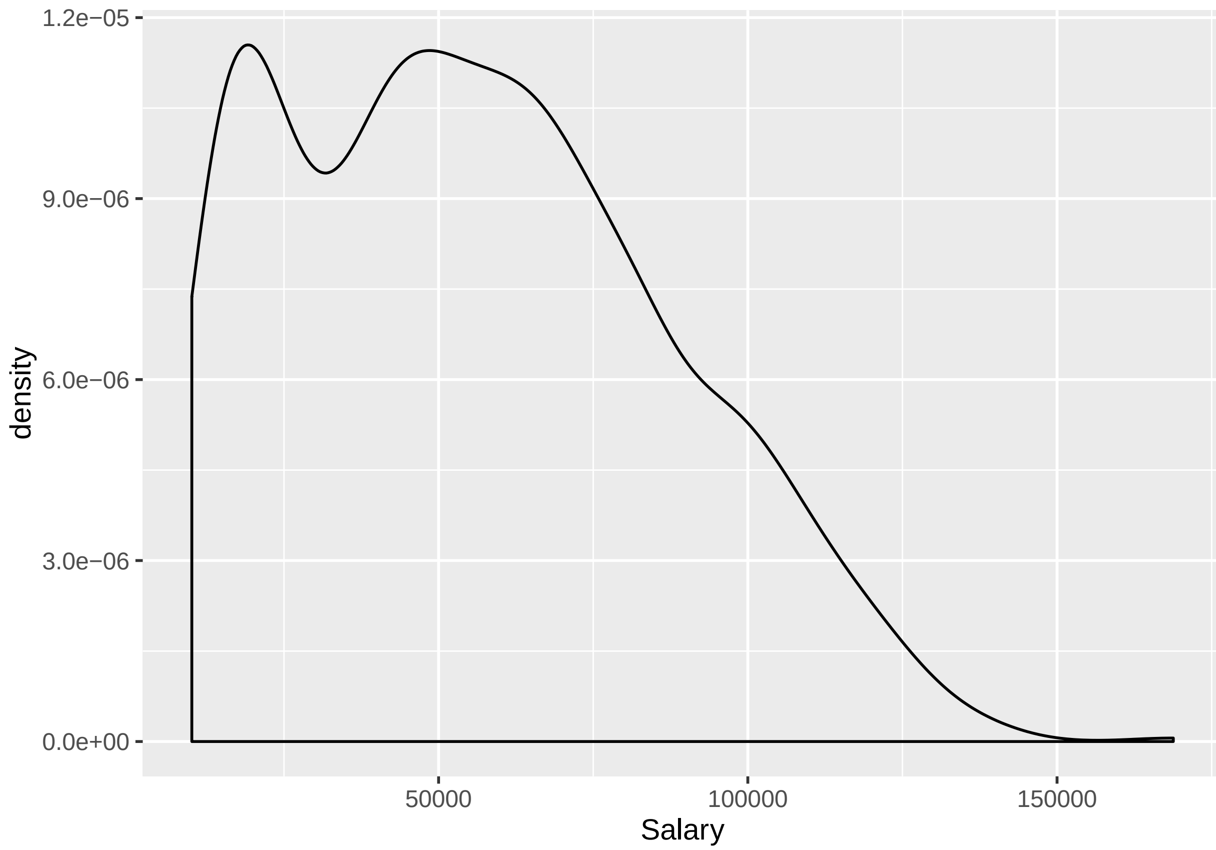
```r
library("ggplot2")

ggplot(train.clean, aes(x = AmountSpent)) + geom_density()
```



As we can see above the AmountSpent is quite right skewed, suggesting that the data may be quite inconsistent. Let us observe for Salary and see what distributon it follows

Salary

```r
ggplot(train.clean, aes(x = Salary)) + geom_density()
```



Again for Salary we see the data being right skewed. There are appropriate transformations that we can apply to better normalize these points. We shal first look at the correlation and scatterplots and take a call.

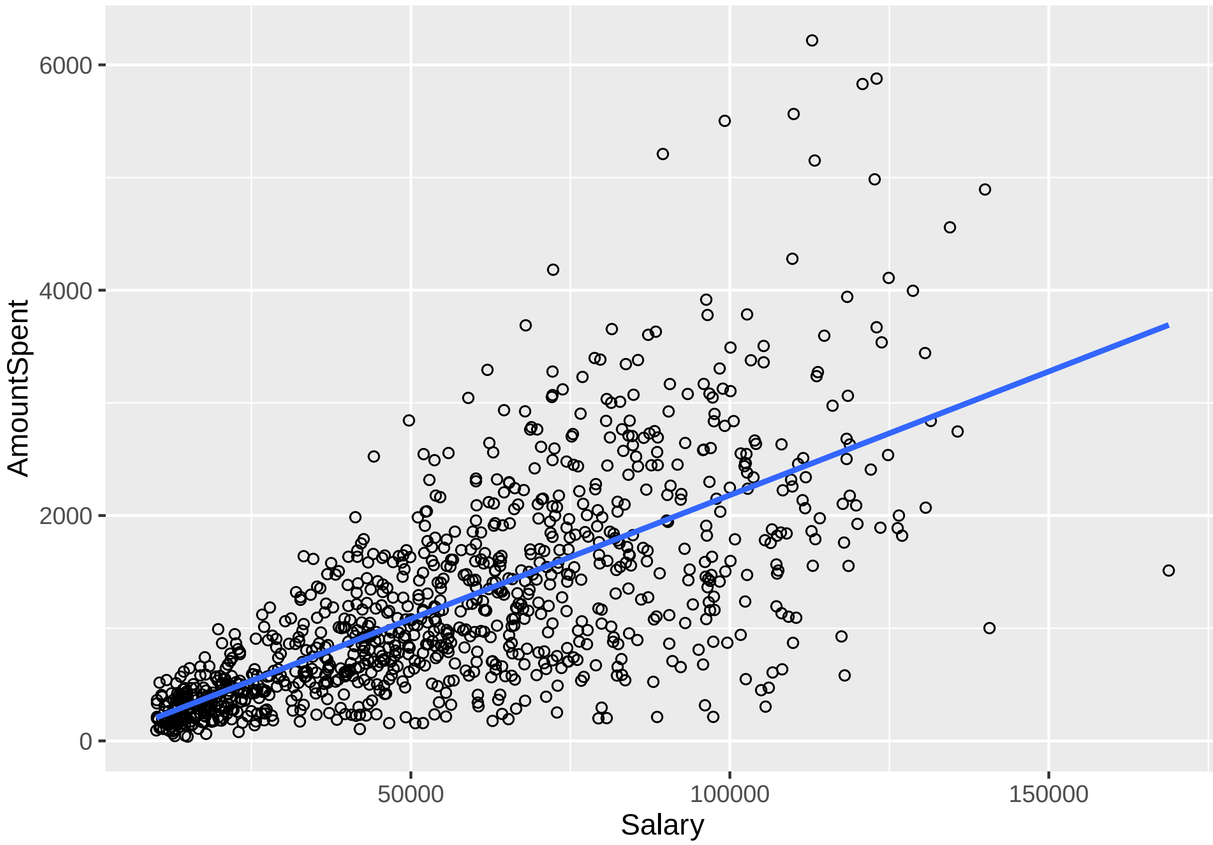
d. Correlation and Scatterplots for Numerical Variables

Let us first observe the correlation and scatterplots between AmountSpent and Salary

```
cor(train.clean$Salary, train.clean$AmountSpent)
```

```
## [1] 0.6995957
```

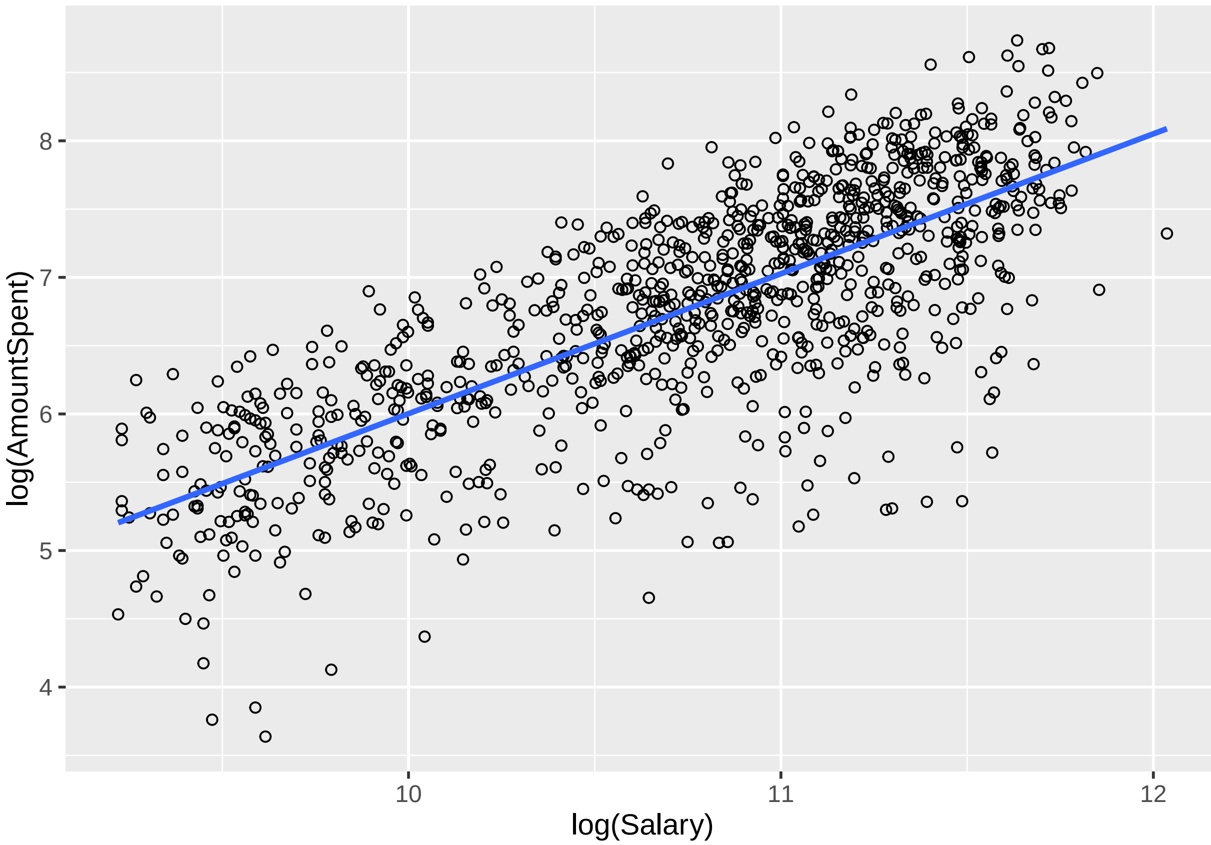
```
ggplot(train.clean, aes(x = Salary, y = AmountSpent)) +
  geom_point(shape=1) +  geom_smooth(method=lm, se=FALSE)
```



We can see above that there is a correlation of 0.70 between AmountSpent and Salary, which is good. This, seems right because, customers with higher salaries would tend to spend more But, we can also observe in the scatterplot that the relation seems funneled out.We also observed in the previous sections that both Salary and AmountSpent were having right skewed distributions. Putting these two together it means there is low vairance at lower salaries and high variance when salaries increase. This is problematic as we won't be able to make acurate predictions about our high paying customers.

Let us try to mitigate this problem by trying out log transforamtions on both sides. The log transformation will help to bring together the large data values while leaving the smaller values unchanged

```
ggplot(train.clean, aes(x = log(Salary), y = log(AmountSpent))) +
  geom_point(shape=1) +  geom_smooth(method=lm, se=FALSE)
```




```
cor(log(train.clean$Salary), log(train.clean$AmountSpent))
```

```
## [1] 0.7625987
```

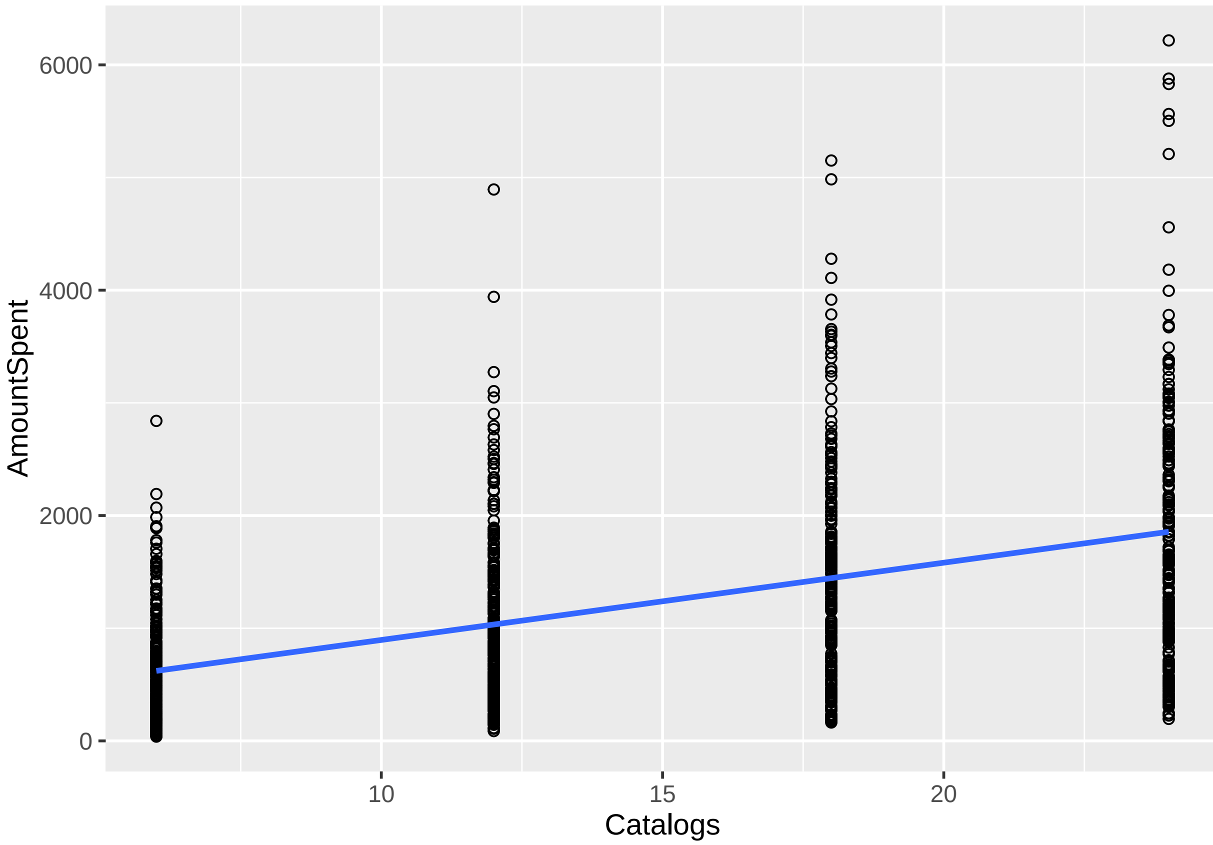
As we can see now the funnel effect at the higher end is reduced and the relationship seems more linear. This ensures that the variance is approximately same at all the levels. This may now enable us to predict the low spender's and high spender's effect at with similar accuracy. The correlation has also improved to 0.76.

Let us move on to AmountSpent vs Catalogs

```
cor(train.clean$Catalogs, train.clean$AmountSpent)
```

```
## [1] 0.4726499
```

```
ggplot(train.clean, aes(x = Catalogs, y = AmountSpent)) +
  geom_point(shape=1) +  geom_smooth(method=lm, se=FALSE)
```



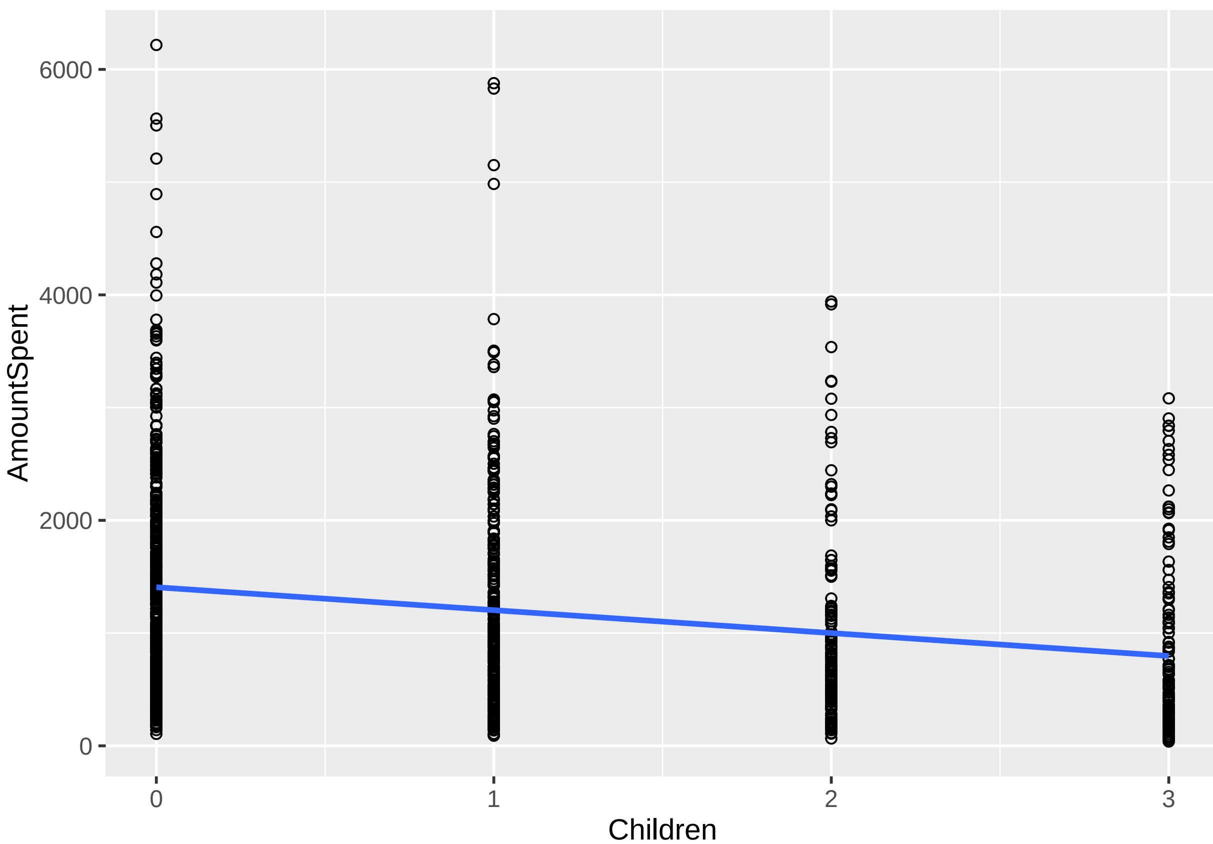
We see that catalogs have discrete values and for each level the scatterplot seems to be be linear. There is a positive correlation of 0.47. The point to note here is that as the catalogs are increased there is an increase in customers willing to spend more money. This can suggest that higher the number of catalogs being sent, the more chances of continued or higher purchasing.

Lastly let us look at Chidren vs AmountSpent

```
cor(train.clean$Children, train.clean$AmountSpent)
```

```
## [1] -0.2223082
```

```
ggplot(train.clean, aes(x = Children, y = AmountSpent)) +
  geom_point(shape=1) +  geom_smooth(method=lm, se=FALSE)
```



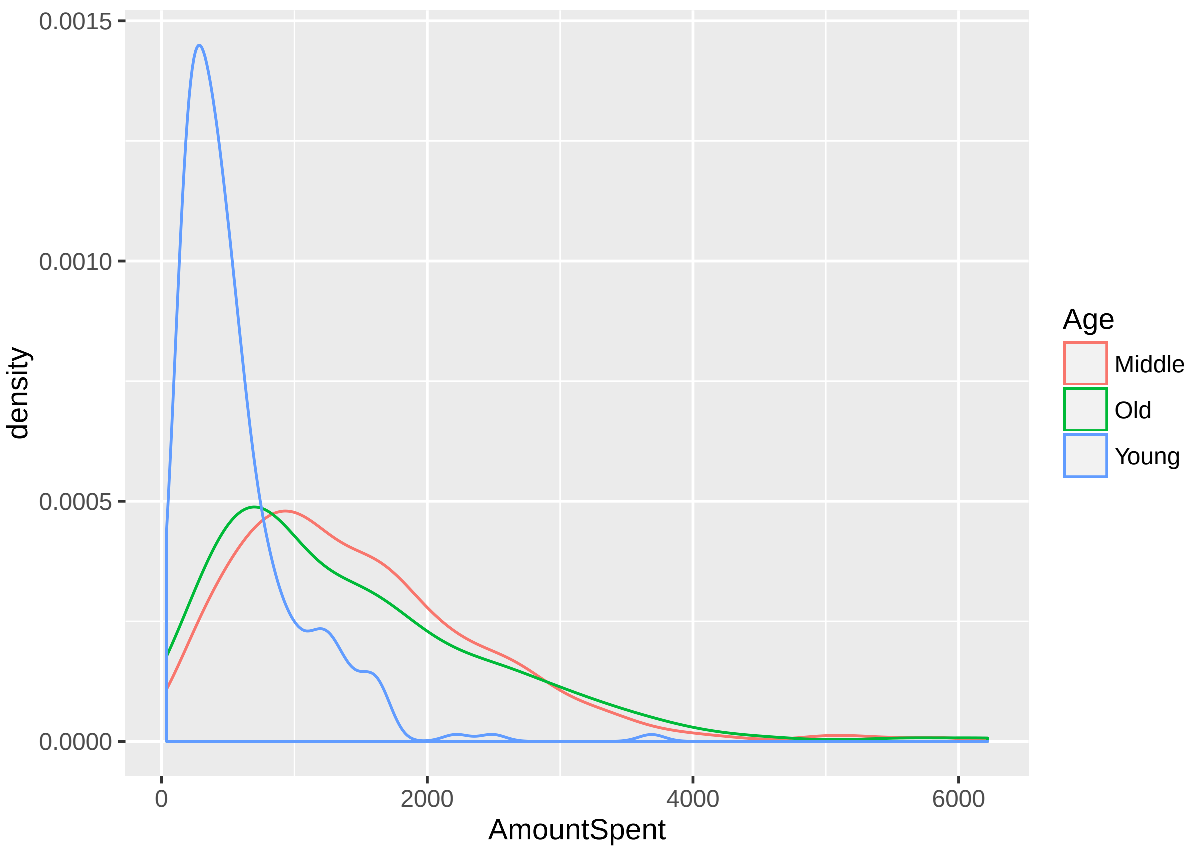
Here we see a negative correlation as the number of children go up. This is quite understandable, as more the children, there is less chance of people spending more. Customers with no children tend to make more as well as higher amount purchases. Does that mean single customers also make higher purchases? We shall see that later.

Let us now move on to see the density plots for the categorical variables against AmountSpent

e. Density plots of categorical

AmountSpent by Age

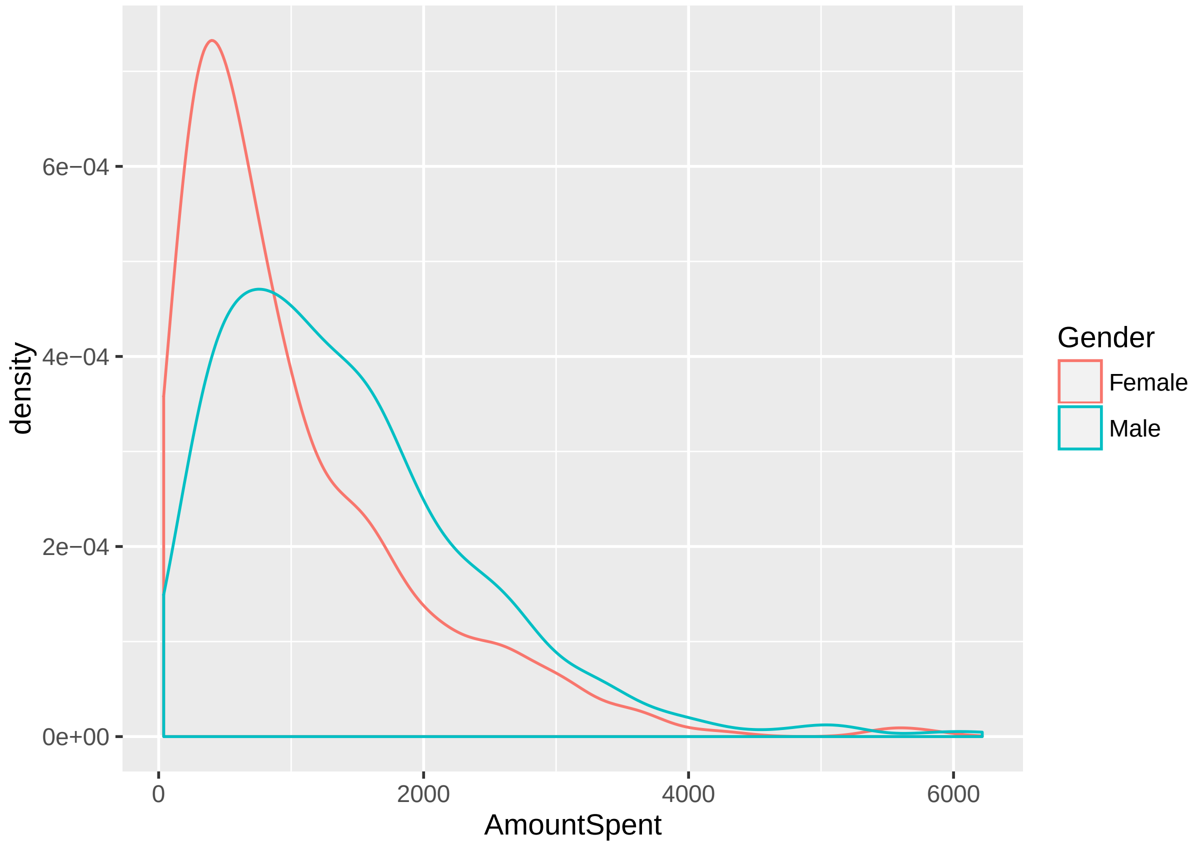
```
ggplot(train.clean, aes(x=AmountSpent, color=Age) )+
  geom_density(alpha = 0.5)
```



We see that the young customers make considerably higher purchases than middle aged or old customers. However, the AmountSpent of young customers tend to be of lower amounts. The middle aged and old customers have higher AmountSpent.

AmountSpent by Gender

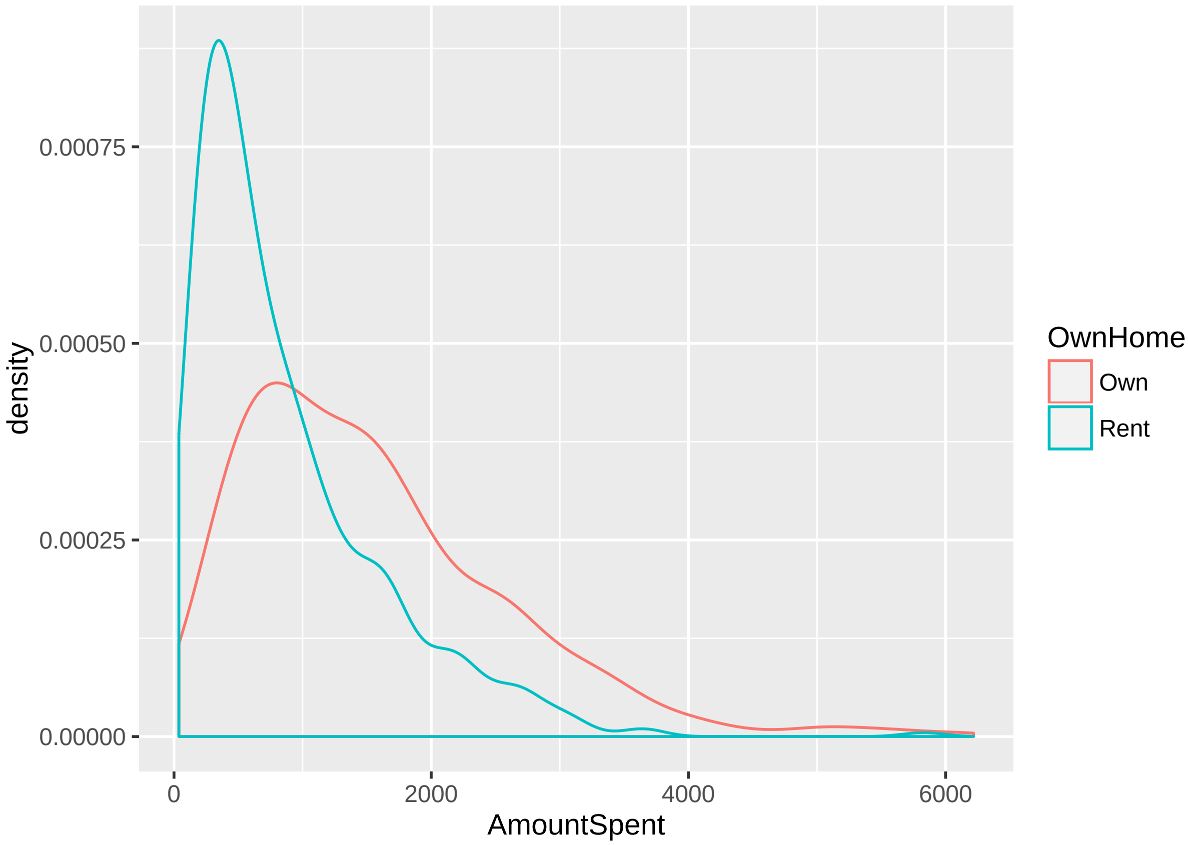
```
ggplot(train.clean, aes(x=AmountSpent, color=Gender) )+
  geom_density(alpha = 0.5)
```



The female customers make slightly higher purchases than male customers. Also, there is a slight increase in the AmountSpent by males. They seem to balance each other out.

AmountSpent by Own or Rented Home

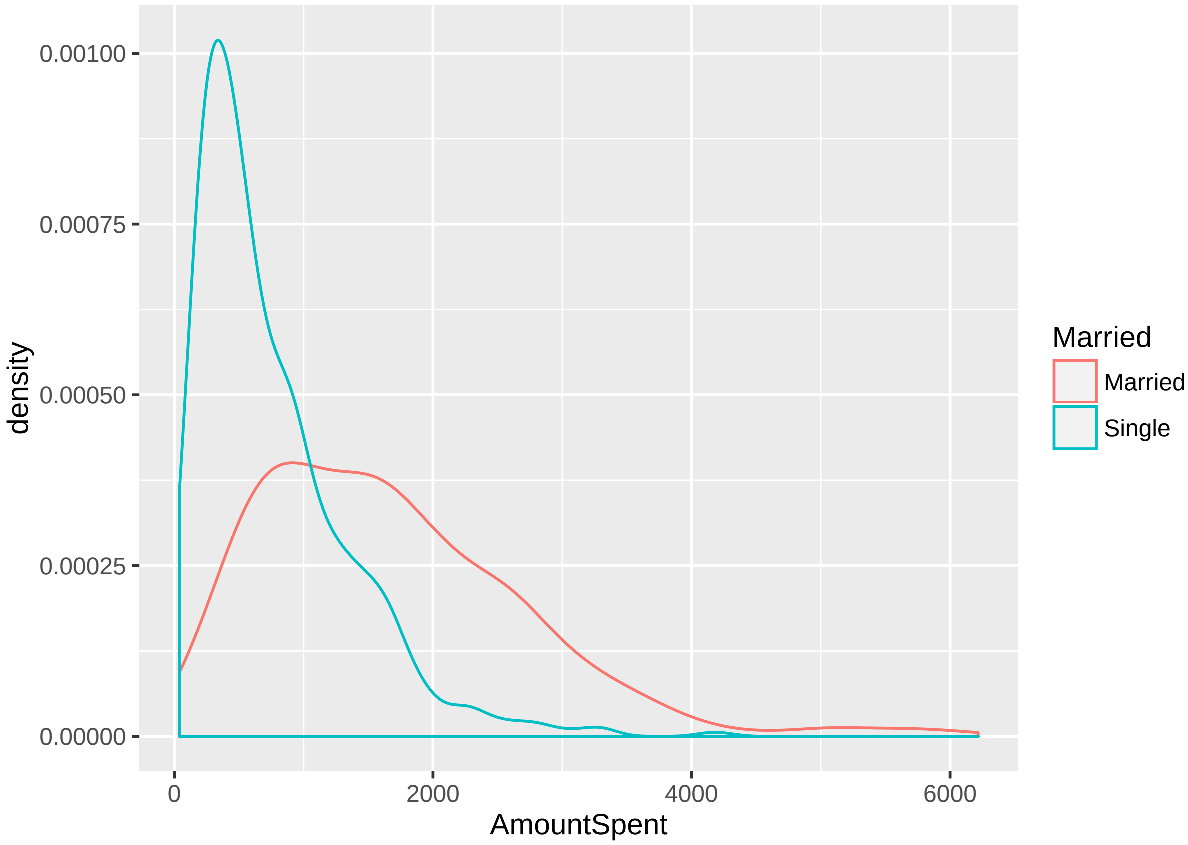
```
ggplot(train.clean, aes(x=AmountSpent, color=OwnHome) )+
  geom_density(alpha = 0.5)
```



Customers with own Home yet again spent higher amounts than customers with Rented homes. This pattern seem to signal some sort of financial stability. Groups with better financial stability tend to have higher AmountSpent but slightly less number of purchases than groups with lesser financial stability.

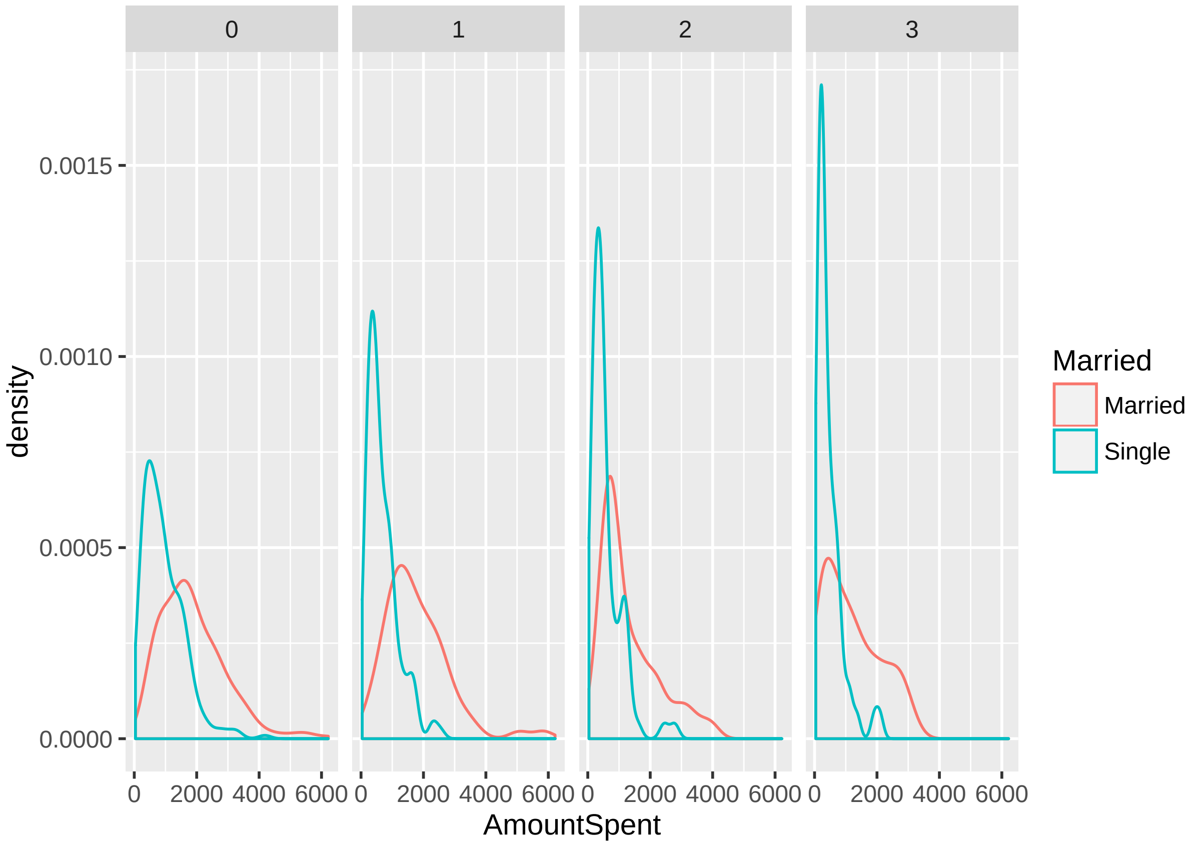
AmountSpent by Single vs Married

```
ggplot(train.clean, aes(x=AmountSpent, color=Married) )+
  geom_density(alpha = 0.5)
```



Yet again the same pattern as above. Single people may be living in rented homes and may be younger. At this point let us try and observe the patterns for married status and children with AmountSpent. A question we asked in the previou section. We can do so by adding a facet grid based on Children to our previous plot.

```
ggplot(train.clean, aes(x=AmountSpent, color=Married) )+
  geom_density(alpha = 0.5)+facet_grid(~Children)
```



Fascinatingly we get to observe that the single parent with 3 children or more tend to make many purchases of lesser Amount. The trend between Single and Married continues to remain at each level. However, as we move across levels the range of AmounSpent decreases and quantities increases. The high volume of single parents with 3 children purchases tends to be due to the factor of convenience associated with direct marketing.

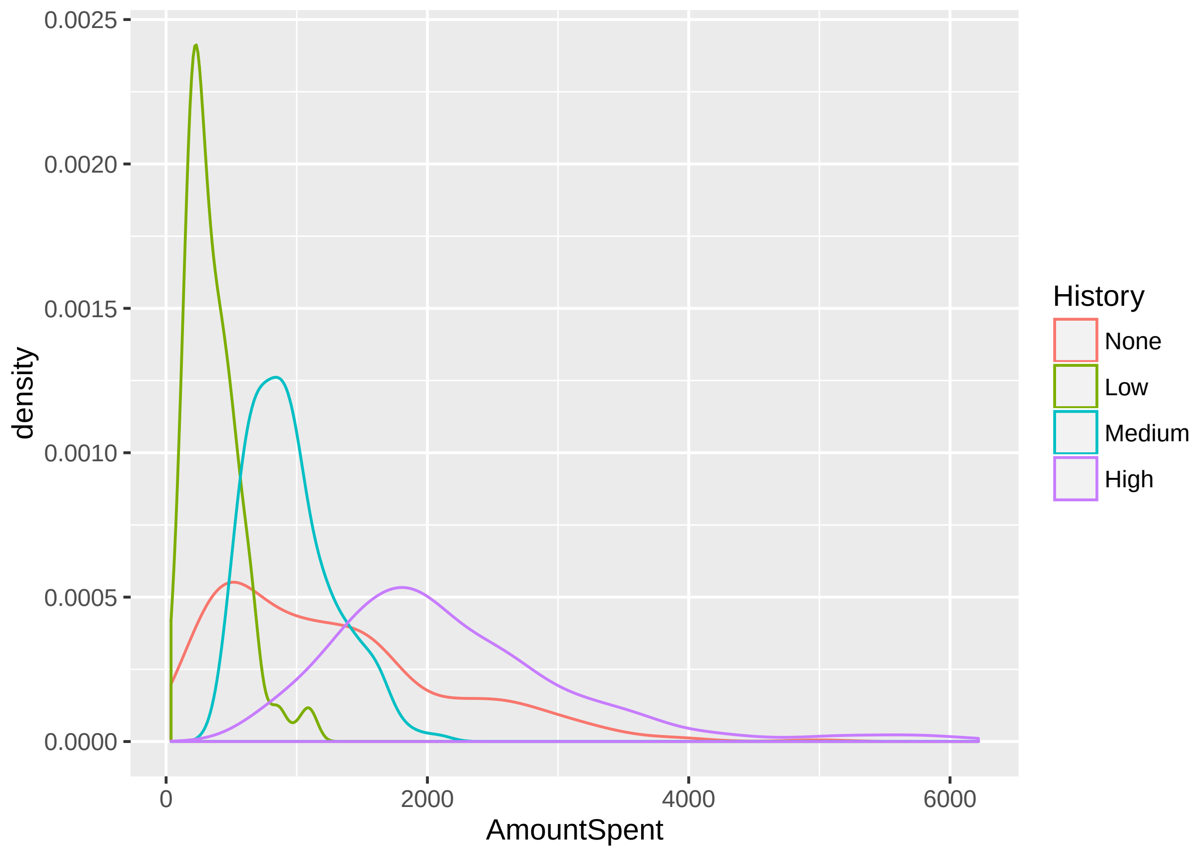
AmountSpent by Customers close and far

```
ggplot(train.clean, aes(x=AmountSpent, color=Location) )+
  geom_density(alpha = 0.5)
```

This trend seems to make sense. The customers living closer to brick and mortar stores tend to make lesser purchases of higher amounts, as it is always safe to go and check out quality in a store and make the purchase. This is not possible for far customers and thus the addition of some high AmountSpent.

AmountSpent by History of purchases

```
ggplot(train.clean, aes(x=AmountSpent, color=History) )+
  geom_density(alpha = 0.5)
```

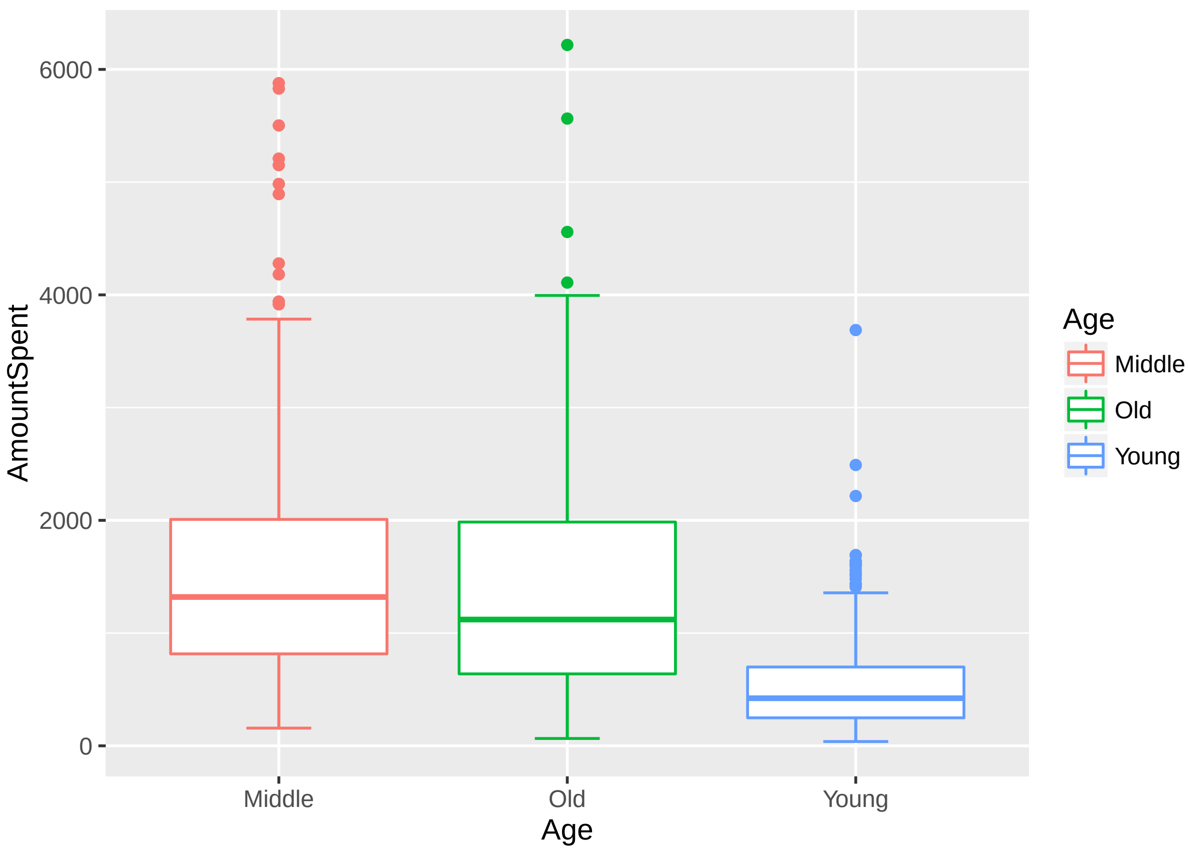


Here we see that most low history of purchases were for lower Amount Spent, highlighting one shot or a few more interactive customers.There is a low density of customers with higher history, more loyal customers and a medium range for customers with medium purchase history.The None(NA) values do not offer much information.

f. Mean Comaprison for categories in Categorical variables

Age vs AmountSpent

```
#Age vs AmountSpent
Age.Young = subset(train.clean, train.clean$Age=='Young')
Age.Middle =  subset(train.clean, train.clean$Age=='Middle')
Age.Old =  subset(train.clean, train.clean$Age=='Old')

ggplot(train.clean, aes(x = Age, y = AmountSpent, color=Age)) +
  stat_boxplot(geom = "errorbar", width = 0.3) + geom_boxplot()
```




```
mean(Age.Young$AmountSpent)
```

```
## [1] 558.6237
```

```
mean(Age.Middle$AmountSpent)
```

```
## [1] 1501.691
```

```
mean(Age.Old$AmountSpent)
```

```
## [1] 1432.127
```

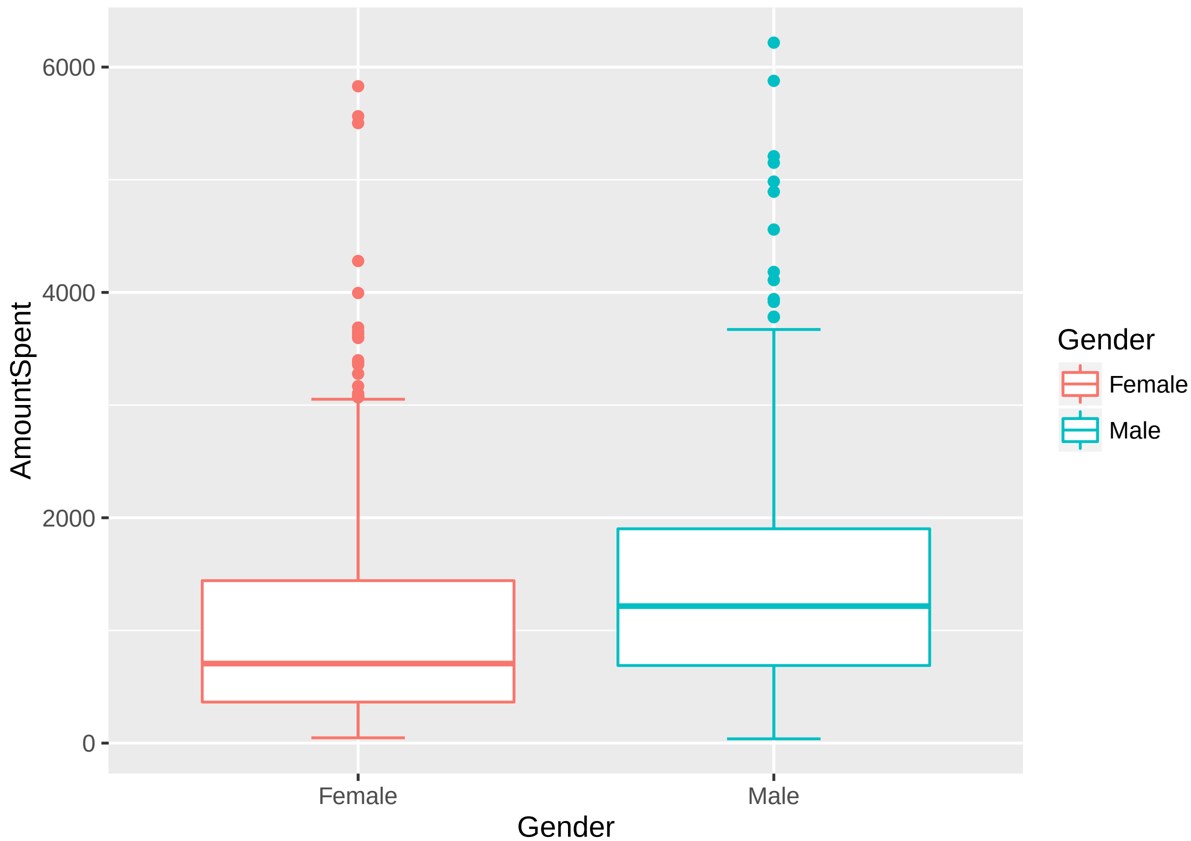
```
#ANOVA test
oneway.test(train.clean$AmountSpent ~ train.clean$Age, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  train.clean$AmountSpent and train.clean$Age
## F = 208.08, num df = 2.00, denom df = 477.07, p-value < 2.2e-16
```

As can be seen the mean of young vs the middle and old is low. This is also seen by extremley low p values for the one way anova test. There is a statistically significant difference.

Gender vs AmountSpent

```r
#Gender vs AmountSpent
Gender.Female = subset(train.clean, train.clean$Gender=='Female')
Gender.Male =  subset(train.clean, train.clean$Gender=='Male')

ggplot(train.clean, aes(x = Gender, y = AmountSpent, color=Gender)) +
  stat_boxplot(geom = "errorbar", width = 0.3) + geom_boxplot()
```




```r
mean(Gender.Female$AmountSpent)
```

```
## [1] 1025.34
```

```r
mean(Gender.Male$AmountSpent)
```

```
## [1] 1412.85
```

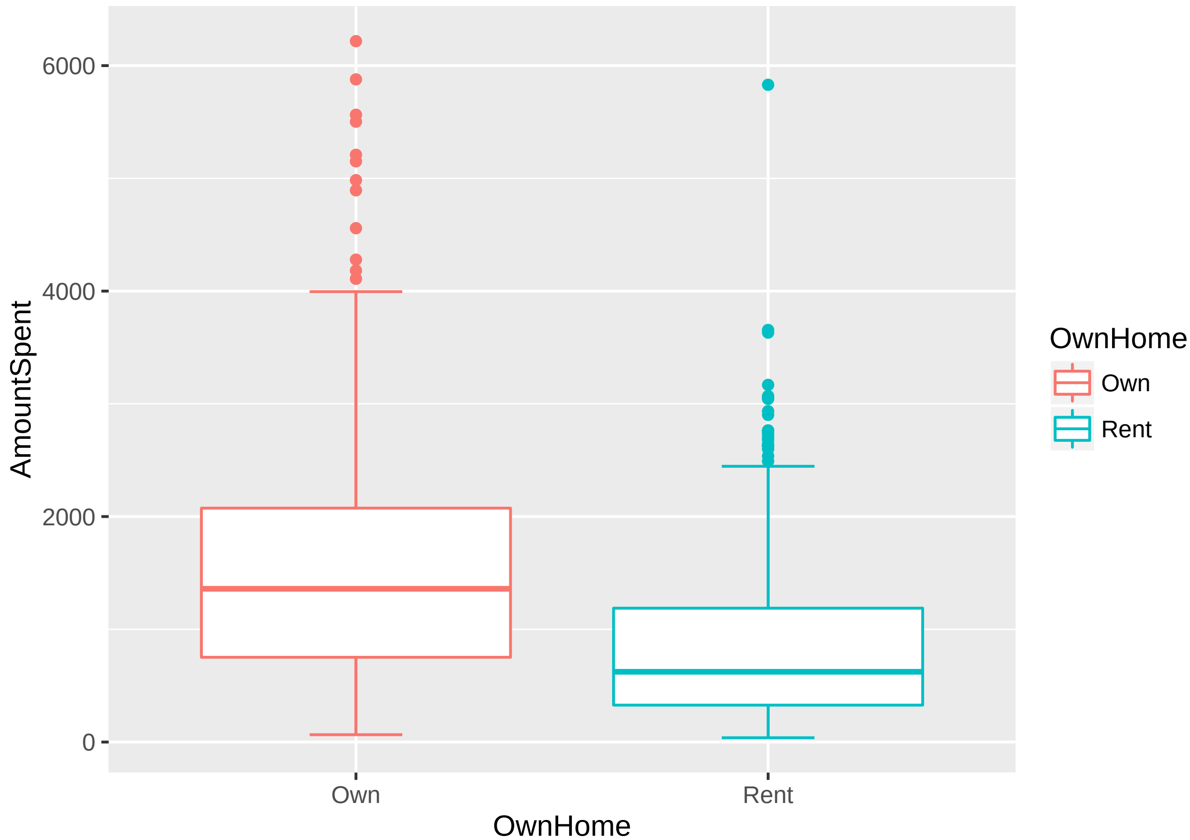
```r
#ANOVA test
oneway.test(train.clean$AmountSpent ~ train.clean$Gender, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  train.clean$AmountSpent and train.clean$Gender
## F = 42.252, num df = 1.00, denom df = 989.98, p-value = 1.27e-10
```

There is also a difference in means of males vs females. Male has a higher mean for AmountSpent.

OwnHome vs AmountSpent

```
#OwnHome vs AmountSpent
OwnHome.Own = subset(train.clean, train.clean$OwnHome=='Own')
OwnHome.Rent = subset(train.clean, train.clean$OwnHome=='Rent')

ggplot(train.clean, aes(x = OwnHome, y = AmountSpent, color=OwnHome)) +
  stat_boxplot(geom = "errorbar", width = 0.3) + geom_boxplot()
```




```
mean(OwnHome.Rent$AmountSpent)
```

```
## [1] 868.8264
```

```
mean(OwnHome.Own$AmountSpent)
```

```
## [1] 1543.136
```

```
#ANOVA test
oneway.test(train.clean$AmountSpent ~ train.clean$OwnHome, var.equal = FALSE)
```

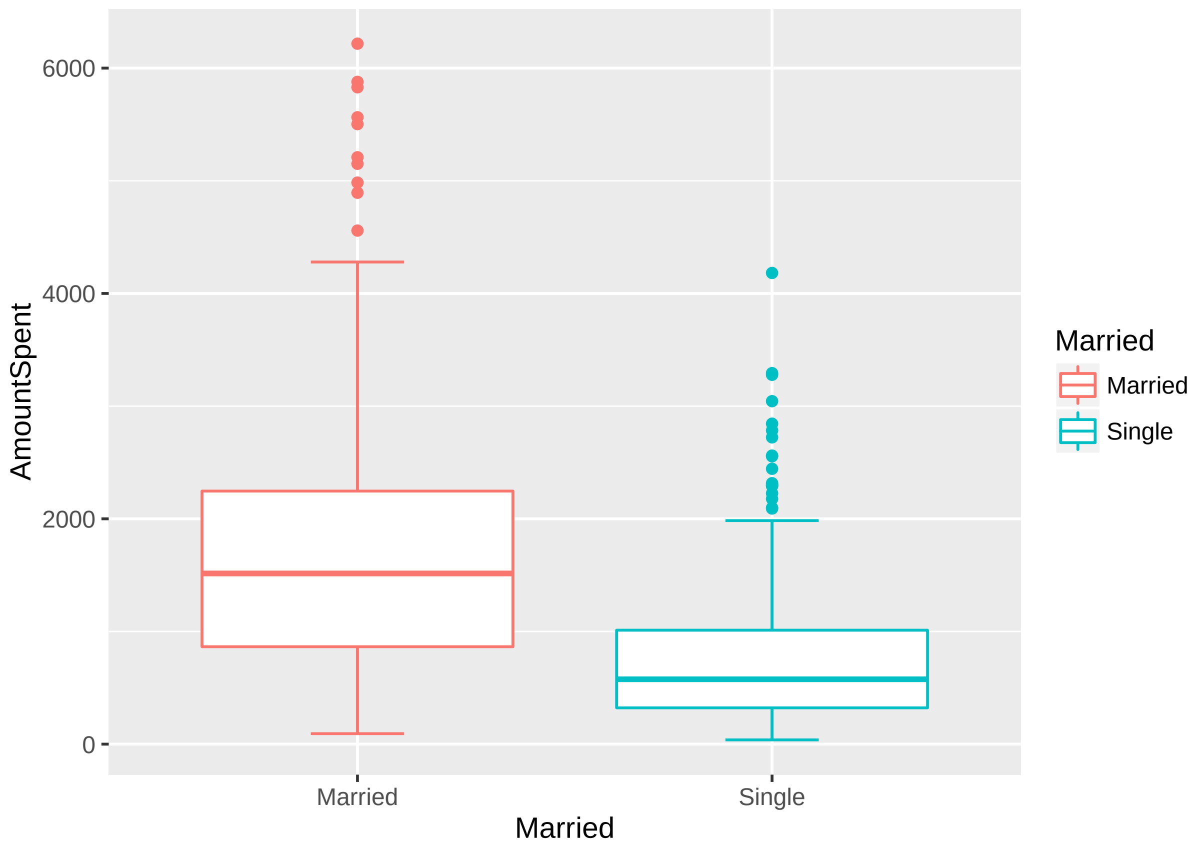
```
##
##  One-way analysis of means (not assuming equal variances)
```

```
##
## data:  train.clean$AmountSpent and train.clean$OwnHome
## F = 142.98, num df = 1.00, denom df = 934.01, p-value < 2.2e-16
```

As seen above there is a statistcally significant difference in mean of own home and rented home on AmountSpent.

Married vs AmountSpent

```
#Married vs AmountSpent
Married.Single = subset(train.clean, train.clean$Married=='Single')
Married.Married = subset(train.clean, train.clean$Married=='Married')

ggplot(train.clean, aes(x = Married, y = AmountSpent, color=Married)) +
  stat_boxplot(geom = "errorbar", width = 0.3) + geom_boxplot()
```




```
mean(Married.Single$AmountSpent)
```

```
## [1] 757.8133
```

```
mean(Married.Married$AmountSpent)
```

```
## [1] 1672.07
```

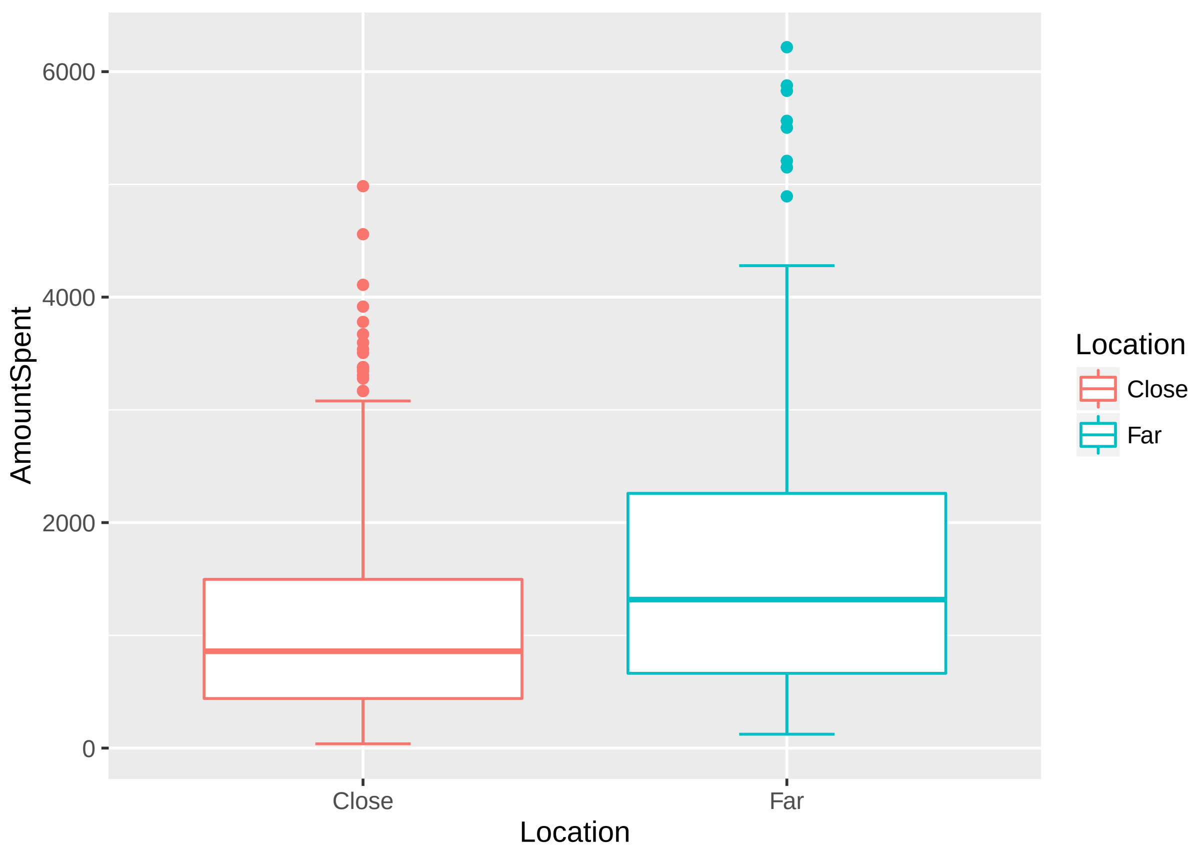
```r
#ANOVA test
oneway.test(train.clean$AmountSpent ~ train.clean$Married, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  train.clean$AmountSpent and train.clean$Married
## F = 293.37, num df = 1.00, denom df = 797.33, p-value < 2.2e-16
```

Here too we see the means of Single vs Married is statistically significantly different based on AmountSpent

Location vs AmountSpent

```r
#Location vs AmountSpent
Location.Close = subset(train.clean, train.clean$Location=='Close')
Location.Far = subset(train.clean, train.clean$Location=='Far')

ggplot(train.clean, aes(x = Location, y = AmountSpent, color=Location)) +
  stat_boxplot(geom = "errorbar", width = 0.3) + geom_boxplot()
```




```r
mean(Location.Close$AmountSpent)
```

```
## [1] 1061.686
```

```r
mean(Location.Far$AmountSpent)
```

```
## [1] 1596.459
```

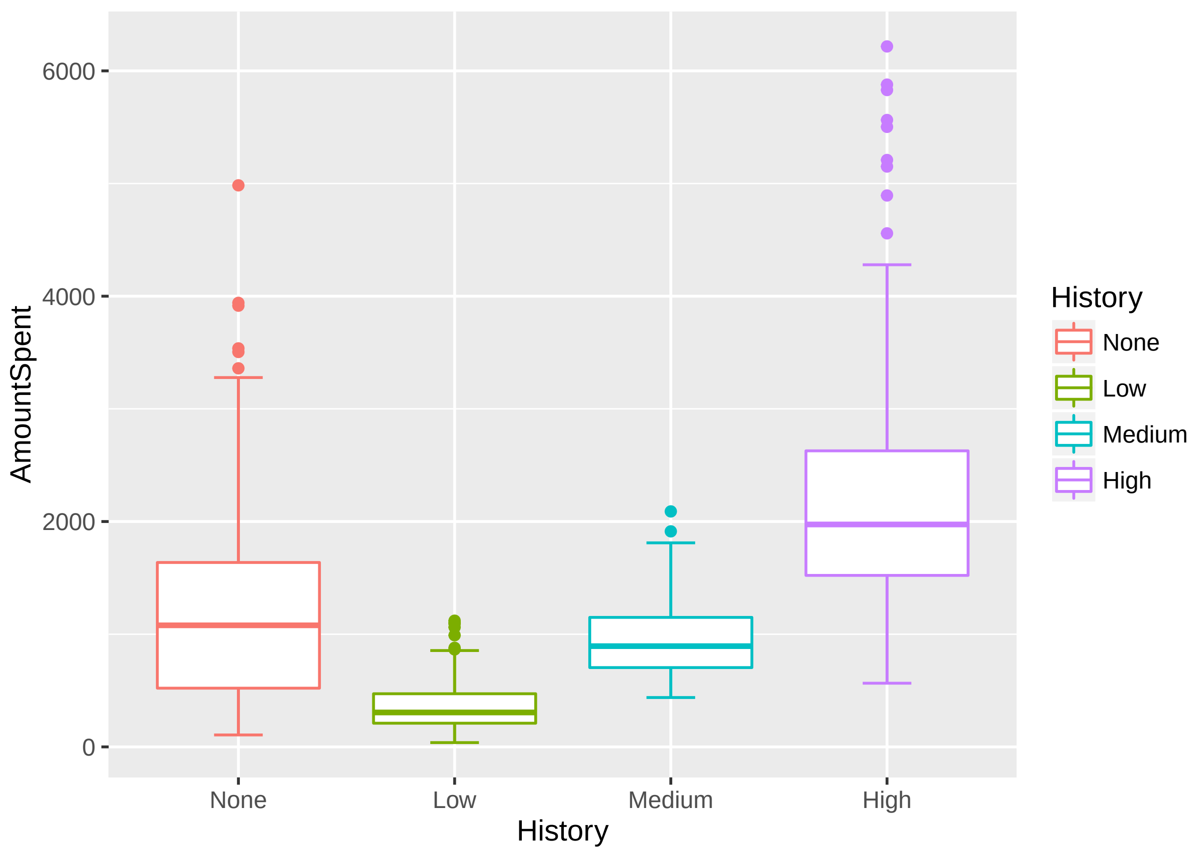
```r
#ANOVA test
oneway.test(train.clean$AmountSpent ~ train.clean$Location, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  train.clean$AmountSpent and train.clean$Location
## F = 50.185, num df = 1.00, denom df = 404.96, p-value = 6.239e-12
```

Same trend here too. The means are statistically and significantly different.

History vs AmountSpent

```r
#History vs AmountSpent
History.Low = subset(train.clean, train.clean$History=='Low')
History.Medium = subset(train.clean, train.clean$History=='Medium')
History.High = subset(train.clean, train.clean$History=='High')

ggplot(train.clean, aes(x = History, y = AmountSpent, color=History)) +
  stat_boxplot(geom = "errorbar", width = 0.3) + geom_boxplot()
```



```r
mean(History.Low$AmountSpent)
```

```
## [1] 357.087
```

```r
mean(History.Medium$AmountSpent)
```

```
## [1] 950.4009
```

```r
mean(History.High$AmountSpent)
```

```
## [1] 2186.137
```

```r
#ANOVA test
oneway.test(train.clean$AmountSpent ~ train.clean$History, var.equal = FALSE)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  train.clean$AmountSpent and train.clean$History
## F = 470.8, num df = 3.00, denom df = 507.08, p-value < 2.2e-16
```

The above trend shows extreme leaps in mean. This is also confirmed by the extremely low p values in the anova test.

3. Regression, Prediction and Modelling

a. Simple Regression using all the predictors:

Let us now look into using a simple regression mode using all the predictors and see what we get

```r
fitFull = lm(AmountSpent ~ Catalogs + Salary  + Children + History +
               Age + Gender + Location + Married + OwnHome, data=train.clean)

summary(fitFull)
```

```
##
## Call:
## lm(formula = AmountSpent ~ Catalogs + Salary + Children + History +
##     Age + Gender + Location + Married + OwnHome, data = train.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1711.44  -292.41   -17.56   237.87  2876.91
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -278.75674  101.88340  -2.736  0.00633 **
## Catalogs        41.86880    2.45796  17.034  < 2e-16 ***
## Salary           0.01920    0.00103  18.652  < 2e-16 ***
```

```
## Children      -162.73555   18.00348  -9.039  < 2e-16 ***
## HistoryLow    -359.88752   46.71128  -7.705 3.19e-14 ***
## HistoryMedium -411.40232   44.86553  -9.170  < 2e-16 ***
## HistoryHigh     -6.99218   51.32915  -0.136  0.89167
## AgeOld          63.36828   47.79586   1.326  0.18521
## AgeYoung         8.90120   49.70059   0.179  0.85790
## GenderMale     -46.99837   32.85192  -1.431  0.15286
## LocationFar    436.50575   35.92138  12.152  < 2e-16 ***
## MarriedSingle   32.74314   44.54067   0.735  0.46244
## OwnHomeRent    -16.63382   36.64327  -0.454  0.64997
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 485.7 on 987 degrees of freedom
## Multiple R-squared:  0.7476, Adjusted R-squared:  0.7446
## F-statistic: 243.7 on 12 and 987 DF,  p-value: < 2.2e-16
```

```r
model.mse = mean(residuals(fitFull)^2)
model.mse
```

```
## [1] 232860.9
```

```r
rmse = sqrt(model.mse)
rmse
```

```
## [1] 482.5567
```

We see that we have a considerable model with R squared of 74.7% and Adjusted R-squared of 74% . We also see significance of Catalogs, Salary, Children, History and Loaction variables. The model is statistically significant owing to the extremely low p-value. However, there is high residual error, and high RMSE (RMSE taken without cross-validation). These values show that the model is useful(Good R-squared and significant) but the performance is low(High RMSE) and the data does not fit the model very well(high residual error).

The results can be interpredeted as:

1. A unit increase in Catalogs influences a 41.86880 increase in AmountSpent, controlling for all other predictors.
2. A unit increase in Salary influences a 0.01920 increase in AmountSpent, controlling for all other predictors.
3. A unit increase in Children influences a -162.73555 decrease in AmountSpent, controlling for all other predictors.
4. A unit increase in HistoryLow referenced on None(NA) influences a -359.88752 decrease in AmountSpent, controlling for all other predictors.
5. A unit increase in HistoryMedium referenced on None(NA) influences a -411.40232 decrease in AmountSpent, controlling for all other predictors.
6. A unit increase in LocationFar referenced on LocationClose influences a 436.50575 increase in AmountSpent, controlling for all other predictors.

The high p-values of Age, Gender, Married, OwnHome, HistoryHigh suggests that none of them are linearly related to AmountSpent controlling for all other variables.

Together all these predictors account for 74.7% of the variance in AmountSpent across customers.

b. Model Selection

Let us now try to combine various predictors and build linear as well as non linear models and use out-of-sample evaluation using leave one out cross-validation to select the best model.

1. AmoountSpent with Salary:

```
fitSal = lm(AmountSpent ~ Salary, data=train.clean)
summary(fitSal)
```

```
##
## Call:
## lm(formula = AmountSpent ~ Salary, data = train.clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2179.7  -315.2   -53.5   279.7  3752.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15.31783   45.37416  -0.338    0.736
## Salary        0.02196    0.00071  30.930   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 687.1 on 998 degrees of freedom
## Multiple R-squared:  0.4894, Adjusted R-squared:  0.4889
## F-statistic: 956.7 on 1 and 998 DF,  p-value: < 2.2e-16
```

```
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~  Salary , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] -0.02136584
```

```
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 688.2645
```

We did see before that there is a strong correlation between Salary and AmountSpent (0.70). It makes sense to see what we get with this basic model. However we do see that we have a less R-squared, High residual error and also high RMSE for the model. This is not great. Let us now try a variant. We will try to add a log transformation and see the result, as we saw before that log reduces the high funneling at the high ends.

```r
fitSalLog = lm(log(AmountSpent) ~ log(Salary), data=train.clean)
summary(fitSalLog)
```

```
##
## Call:
## lm(formula = log(AmountSpent) ~ log(Salary), data = train.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.16381 -0.28005  0.07409  0.40193  1.11970
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.24279    0.29630  -14.32   <2e-16 ***
## log(Salary)  1.02449    0.02751   37.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5666 on 998 degrees of freedom
## Multiple R-squared:  0.5816, Adjusted R-squared:  0.5811
## F-statistic:  1387 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(log(AmountSpent) ~  log(Salary) , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] 1209.998
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 1544.658
```

The above model improves the residual error dramtically and we also see an increase in R-squared but, the RMSE is terrible. Seems the good result is due to overfitting and we don't want such a model.

We also saw a good correlation with Catalogs.

```r
fitCat = lm(AmountSpent ~ Catalogs, data=train.clean)
summary(fitCat)
```

```
##
```

```
## Call:
## lm(formula = AmountSpent ~ Catalogs, data = train.clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1660.9  -536.2  -135.1   399.8  4361.1
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  209.766     65.194   3.218  0.00133 **
## Catalogs      68.588      4.048  16.944  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 847.4 on 998 degrees of freedom
## Multiple R-squared:  0.2234, Adjusted R-squared:  0.2226
## F-statistic: 287.1 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~ Catalogs , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] -0.01272292
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 848.2697
```

The scores are extremely poor! High RMSE, low R-squared, High residual error.

Let us now try a combination of all the numerical variabes

```r
fitNum = lm(AmountSpent ~ Catalogs + Children + Salary, data=train.clean)
summary(fitNum)
```

```
##
## Call:
## lm(formula = AmountSpent ~ Catalogs + Children + Salary, data = train.clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1775.9  -348.7   -38.7   255.5  3211.3
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.428e+02  5.372e+01  -8.242 5.29e-16 ***
## Catalogs     4.770e+01  2.755e+00  17.310  < 2e-16 ***
## Children    -1.987e+02  1.709e+01 -11.628  < 2e-16 ***
## Salary       2.041e-02  5.929e-04  34.417  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 562.5 on 996 degrees of freedom
## Multiple R-squared:  0.6584, Adjusted R-squared:  0.6574
## F-statistic:   640 on 3 and 996 DF,  p-value: < 2.2e-16
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~ Catalogs + Children + Salary , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] -0.01977371
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 564.2606
```

Yes, we do have a better model now. The R-squared explains 66% of the variance. Decreased RMSE and Residual error, but still they are considerably high. Let us try to improve further.

Based on the significance values of our full model on all the predictors which we performed in the previous section, let us try to add the sinificant variables. Note, here we are going to use a combination of numerical as well as catgorical variables. We are also not going to consider interactions between the variables even if any exists.

```r
fitComb = lm(AmountSpent ~ Catalogs + Children + Salary +
               History + Location, data=train.clean)
summary(fitComb)
```

```
##
## Call:
## lm(formula = AmountSpent ~ Catalogs + Children + Salary + History +
##     Location, data = train.clean)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1651.7  -287.9   -11.6   239.7  2913.2
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.465e+02  5.611e+01  -4.392 1.24e-05 ***
## Catalogs       4.165e+01  2.453e+00  16.979  < 2e-16 ***
## Children      -1.694e+02  1.665e+01 -10.179  < 2e-16 ***
## Salary         1.871e-02  6.791e-04  27.551  < 2e-16 ***
## HistoryLow    -3.491e+02  4.628e+01  -7.542 1.04e-13 ***
## HistoryMedium -4.080e+02  4.383e+01  -9.310  < 2e-16 ***
## HistoryHigh    1.875e+00  5.110e+01   0.037    0.971
## LocationFar    4.363e+02  3.589e+01  12.156  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 485.9 on 992 degrees of freedom
## Multiple R-squared:  0.7462, Adjusted R-squared:  0.7444
## F-statistic: 416.6 on 7 and 992 DF,  p-value: < 2.2e-16
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~ Catalogs + Children + Salary + History +
            Location, data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] -0.1315771
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 488.3369
```

Yes, we do have a better model. The R-squared is 75%. Still, even though there is a decrease in residual error and RMSE, they are still high.

At this point let us try some variable selection. We can use STEPAIC to perform the same. Let us try to perform variable selection on the full model.

```r
library(MASS)

fitFull = lm(AmountSpent ~ Catalogs + Salary  + Children +
               History + Age+ Gender+ Location+ Married+ OwnHome, data=train.clean)
stepAIC(fitFull, direction="backward")
```

```
## Start:  AIC=12384.2
## AmountSpent ~ Catalogs + Salary + Children + History + Age +
##     Gender + Location + Married + OwnHome
##
##            Df Sum of Sq       RSS   AIC
## - Age       2    443097 233304046 12382
## - OwnHome   1     48616 232909565 12382
## - Married   1    127499 232988448 12383
## <none>                  232860949 12384
## - Gender    1    482863 233343812 12384
## - Children  1  19276638 252137587 12462
## - History   3  28426404 261287353 12493
## - Location  1  34838025 267698974 12522
## - Catalogs  1  68455782 301316731 12640
## - Salary    1  82083034 314943983 12684
##
## Step:  AIC=12382.1
## AmountSpent ~ Catalogs + Salary + Children + History + Gender +
##     Location + Married + OwnHome
##
##            Df Sum of Sq       RSS   AIC
## - Married   1     55318 233359364 12380
## - OwnHome   1    147202 233451248 12381
## <none>                  233304046 12382
## - Gender    1    664803 233968849 12383
## - Children  1  24879626 258183672 12481
## - History   3  28889456 262193501 12493
## - Location  1  35011045 268315091 12520
## - Catalogs  1  68392954 301697000 12637
## - Salary    1 107651722 340955767 12760
##
## Step:  AIC=12380.33
## AmountSpent ~ Catalogs + Salary + Children + History + Gender +
##     Location + OwnHome
##
##            Df Sum of Sq       RSS   AIC
## - OwnHome   1    162809 233522173 12379
## <none>                  233359364 12380
## - Gender    1    634446 233993810 12381
## - Children  1  24825054 258184418 12479
## - History   3  29027254 262386618 12492
## - Location  1  34961973 268321337 12518
## - Catalogs  1  68354217 301713581 12635
## - Salary    1 153879921 387239285 12885
##
## Step:  AIC=12379.03
## AmountSpent ~ Catalogs + Salary + Children + History + Gender +
##     Location
##
##            Df Sum of Sq       RSS   AIC
## <none>                  233522173 12379
## - Gender    1    670888 234193061 12380
## - Children  1  24994947 258517120 12479
## - History   3  29194376 262716549 12491
```

```
## - Location  1  34842146 268364319 12516
## - Catalogs  1  68330846 301853019 12634
## - Salary    1 177237435 410759607 12942
```

```
##
## Call:
## lm(formula = AmountSpent ~ Catalogs + Salary + Children + History +
##     Gender + Location, data = train.clean)
##
## Coefficients:
##   (Intercept)       Catalogs         Salary       Children     HistoryLow
##    -228.41947       41.74594        0.01892     -171.98225     -355.02137
## HistoryMedium    HistoryHigh     GenderMale    LocationFar
##    -408.77777        0.03510      -54.28354      436.04608
```

Interestingly we see the addition of Gender in the variable selection. Let us include Gender into our predictors.

```
fitNonPoly = lm(AmountSpent ~ Catalogs + Salary + Children + History + Gender + Location, data=train.cl
summary(fitNonPoly)
```

```
##
## Call:
## lm(formula = AmountSpent ~ Catalogs + Salary + Children + History +
##     Gender + Location, data = train.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1649.78  -286.23   -16.99   241.88  2925.47
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.284e+02  5.707e+01  -4.002 6.74e-05 ***
## Catalogs       4.175e+01  2.452e+00  17.029  < 2e-16 ***
## Salary         1.892e-02  6.899e-04  27.425  < 2e-16 ***
## Children      -1.720e+02  1.670e+01 -10.299  < 2e-16 ***
## HistoryLow    -3.550e+02  4.637e+01  -7.656 4.55e-14 ***
## HistoryMedium -4.088e+02  4.379e+01  -9.335  < 2e-16 ***
## HistoryHigh    3.510e-02  5.106e+01   0.001   0.9995
## GenderMale    -5.428e+01  3.217e+01  -1.687   0.0919 .
## LocationFar    4.360e+02  3.586e+01  12.160  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 485.4 on 991 degrees of freedom
## Multiple R-squared:  0.7469, Adjusted R-squared:  0.7449
## F-statistic: 365.6 on 8 and 991 DF,  p-value: < 2.2e-16
```

```
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~ Catalogs + Children + Salary +
```

```r
            History + Location + Gender, data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] -0.1361469
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 488.1256
```

We have a slight improvement.

Let us now try polynomial regression on this model. I have tried different combination and for various degrees for each variable. I will only show the best model I got from them.

```r
poly.fitFinal <- lm(AmountSpent ~ poly(Salary, degree = 6)
                    +poly(Catalogs, degree = 3)+ poly(Children, degree = 3)+ History +
                      Gender + Location, data = train.clean)
summary(poly.fitFinal)
```

```
##
## Call:
## lm(formula = AmountSpent ~ poly(Salary, degree = 6) + poly(Catalogs,
##     degree = 3) + poly(Children, degree = 3) + History + Gender +
##     Location, data = train.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1320.17  -287.91   -19.61   233.50  2856.13
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1283.25      34.12  37.606  < 2e-16 ***
## poly(Salary, degree = 6)1   18218.75     673.77  27.040  < 2e-16 ***
## poly(Salary, degree = 6)2     107.11     519.67   0.206 0.836744
## poly(Salary, degree = 6)3      79.73     500.88   0.159 0.873565
## poly(Salary, degree = 6)4   -1848.59     490.71  -3.767 0.000175 ***
## poly(Salary, degree = 6)5   -1578.19     485.95  -3.248 0.001203 **
## poly(Salary, degree = 6)6    -479.65     487.06  -0.985 0.324965
## poly(Catalogs, degree = 3)1  8630.90     511.05  16.889  < 2e-16 ***
## poly(Catalogs, degree = 3)2   656.90     485.71   1.352 0.176541
## poly(Catalogs, degree = 3)3  -534.43     485.39  -1.101 0.271156
## poly(Children, degree = 3)1 -5745.23     552.97 -10.390  < 2e-16 ***
## poly(Children, degree = 3)2  -280.47     489.98  -0.572 0.567173
## poly(Children, degree = 3)3    28.99     488.61   0.059 0.952698
## HistoryLow                   -356.20      47.42  -7.512 1.31e-13 ***
## HistoryMedium                -416.85      46.02  -9.059  < 2e-16 ***
```

```
## HistoryHigh                      17.08      51.45   0.332 0.740052
## GenderMale                      -49.47      32.08  -1.542 0.123371
## LocationFar                     427.24      35.67  11.978  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 480.7 on 982 degrees of freedom
## Multiple R-squared:  0.7541, Adjusted R-squared:  0.7498
## F-statistic: 177.1 on 17 and 982 DF,  p-value: < 2.2e-16
```

```
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~   poly(Salary, degree = 6) + poly(Catalogs, degree = 3)+
            Location  + poly(Children, degree = 2) + Age+
            History  , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] 0.4718715
```

```
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 487.0383
```

This sees a slight increased RMSE, higher residual and less R-squared. Now let us try to perform a log transformation on Salary and AmountSpent and evaluate.

```
poly.fitFinalLog <- lm(log(AmountSpent) ~ poly(log(Salary), degree = 6) +
                         poly(Catalogs, degree = 3)+ poly(Children, degree = 2)+
                         History + Age + Location, data = train.clean)
summary(poly.fitFinalLog)
```

```
##
## Call:
## lm(formula = log(AmountSpent) ~ poly(log(Salary), degree = 6) +
##     poly(Catalogs, degree = 3) + poly(Children, degree = 2) +
##     History + Age + Location, data = train.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90959 -0.20238  0.00723  0.21076  0.99261
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                    6.90582    0.02342 294.905  < 2e-16 ***
## poly(log(Salary), degree = 6)1 16.92354    0.54035  31.320  < 2e-16 ***
## poly(log(Salary), degree = 6)2 -0.09158    0.33597  -0.273 0.785240
## poly(log(Salary), degree = 6)3  0.91548    0.32561   2.812 0.005028 **
## poly(log(Salary), degree = 6)4  0.37129    0.32082   1.157 0.247421
## poly(log(Salary), degree = 6)5 -0.37837    0.31266  -1.210 0.226514
## poly(log(Salary), degree = 6)6 -0.18888    0.31405  -0.601 0.547690
## poly(Catalogs, degree = 3)1     7.95526    0.32830  24.232  < 2e-16 ***
## poly(Catalogs, degree = 3)2    -1.13978    0.31130  -3.661 0.000264 ***
## poly(Catalogs, degree = 3)3    -0.15736    0.31144  -0.505 0.613475
## poly(Children, degree = 2)1    -6.00088    0.38095 -15.752  < 2e-16 ***
## poly(Children, degree = 2)2    -0.85666    0.32486  -2.637 0.008497 **
## HistoryLow                     -0.58375    0.03098 -18.846  < 2e-16 ***
## HistoryMedium                  -0.30653    0.02951 -10.387  < 2e-16 ***
## HistoryHigh                    -0.12177    0.03311  -3.677 0.000248 ***
## AgeOld                          0.03318    0.03015   1.100 0.271408
## AgeYoung                       -0.03681    0.03312  -1.111 0.266708
## LocationFar                     0.34629    0.02293  15.100  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3086 on 982 degrees of freedom
## Multiple R-squared:  0.8779, Adjusted R-squared:  0.8758
## F-statistic: 415.2 on 17 and 982 DF,  p-value: < 2.2e-16
```

```
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(log(AmountSpent) ~   poly(log(Salary), degree = 6) +
            poly(Catalogs, degree = 3)+ Location  + poly(Children, degree = 2) +
            History + Age , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] 1209.997
```

```
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 1544.516
```

We see a dramatic decrease in the residual error a much better R-squre of 87%. This seems to be like a good model but once we see the RMSE after cross validation, the values are terrible signalling overfitting the data.

We can thus see the best model so far is below.

RMSE before cross-validation: 475.96

RMSE after cross-validation: 487.03

R-squared: 75%

Residual error: 480.3

```r
poly.fitFinal <- lm(AmountSpent ~ poly(Salary, degree = 6)
                    +poly(Catalogs, degree = 3)+ poly(Children, degree = 2)+ History +
                      Age + Location, data = train.clean)
summary(poly.fitFinal)
```

```
##
## Call:
## lm(formula = AmountSpent ~ poly(Salary, degree = 6) + poly(Catalogs,
##     degree = 3) + poly(Children, degree = 2) + History + Age +
##     Location, data = train.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1320.13  -287.87   -12.56   232.41  2793.47
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1245.5074    36.3563  34.258  < 2e-16 ***
## poly(Salary, degree = 6)1  18018.0316   803.1452  22.434  < 2e-16 ***
## poly(Salary, degree = 6)2    137.2666   568.5149   0.241 0.809259
## poly(Salary, degree = 6)3     27.2633   504.7164   0.054 0.956932
## poly(Salary, degree = 6)4  -1866.8418   489.3479  -3.815 0.000145 ***
## poly(Salary, degree = 6)5  -1612.8699   488.2711  -3.303 0.000990 ***
## poly(Salary, degree = 6)6   -470.0979   491.3284  -0.957 0.338909
## poly(Catalogs, degree = 3)1  8639.2666   510.4177  16.926  < 2e-16 ***
## poly(Catalogs, degree = 3)2   658.9493   484.8095   1.359 0.174398
## poly(Catalogs, degree = 3)3  -568.4930   484.7564  -1.173 0.241185
## poly(Children, degree = 2)1 -5246.3243   587.7801  -8.926  < 2e-16 ***
## poly(Children, degree = 2)2  -577.2816   505.7376  -1.141 0.253955
## HistoryLow                  -360.2834    47.7584  -7.544 1.04e-13 ***
## HistoryMedium               -424.7619    45.8218  -9.270  < 2e-16 ***
## HistoryHigh                   10.4041    51.5702   0.202 0.840156
## AgeOld                        86.8035    46.8161   1.854 0.064019 .
## AgeYoung                       0.8889    51.3104   0.017 0.986182
## LocationFar                  425.7739    35.6401  11.946  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 480.3 on 982 degrees of freedom
```

```
## Multiple R-squared:  0.7545, Adjusted R-squared:  0.7502
## F-statistic: 177.5 on 17 and 982 DF,  p-value: < 2.2e-16
```

```r
model.mse = mean(residuals(poly.fitFinal)^2)
model.mse
```

```
## [1] 226543.3
```

```r
rmse = sqrt(model.mse)
rmse
```

```
## [1] 475.9656
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~   poly(Salary, degree = 6) + poly(Catalogs, degree = 3)+
            Location  + poly(Children, degree = 2) + Age+
            History  , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] 0.4718715
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 487.0383
```

c. Most important predictor

It shouldn't come as a surprise that Salary is the most important predictor for this regression. We have seen a strong correlation and also a good score in STEPAIC. Let us now try to find out how good it is: We take our best model and remove a predictor at a time an observe the statistics.

1. Removing Salary

```r
poly.fitBestModel <- lm(AmountSpent ~ poly(Catalogs, degree = 3)+
                          poly(Children, degree = 2)+ History +
                        Age + Location, data = train.clean)
summary(poly.fitBestModel)
```

```
##
## Call:
## lm(formula = AmountSpent ~ poly(Catalogs, degree = 3) + poly(Children,
##     degree = 2) + History + Age + Location, data = train.clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1557.6  -355.8   -70.2   270.3  3681.0
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1445.99      43.48  33.255  < 2e-16 ***
## poly(Catalogs, degree = 3)1  8653.43     645.82  13.399  < 2e-16 ***
## poly(Catalogs, degree = 3)2  -145.50     613.51  -0.237   0.8126
## poly(Catalogs, degree = 3)3    24.50     613.90   0.040   0.9682
## poly(Children, degree = 2)1 -1500.50     718.30  -2.089   0.0370 *
## poly(Children, degree = 2)2  -379.16     634.33  -0.598   0.5502
## HistoryLow                   -699.51      55.64 -12.572  < 2e-16 ***
## HistoryMedium                -450.12      56.30  -7.996 3.58e-15 ***
## HistoryHigh                   515.23      58.47   8.812  < 2e-16 ***
## AgeOld                       -109.48      57.28  -1.911   0.0563 .
## AgeYoung                     -556.09      50.92 -10.922  < 2e-16 ***
## LocationFar                   268.12      44.23   6.061 1.92e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 610 on 988 degrees of freedom
## Multiple R-squared:  0.6016, Adjusted R-squared:  0.5972
## F-statistic: 135.6 on 11 and 988 DF,  p-value: < 2.2e-16
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~   poly(Catalogs, degree = 3)+
            Location  + poly(Children, degree = 2) + Age+
            History  , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] -0.105746
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 613.3017
```

Removing Salary has a high impact on the R-squared it falls from 75% to 60% also the residual error jumps to 610 from 480. The RMSE also shoots to 613 after cross validation

2. Removing Catalogs

```
poly.fitBestModel <- lm(AmountSpent ~ poly(Salary, degree = 6)+
                          poly(Children, degree = 2)+ History +
                      Age + Location, data = train.clean)
summary(poly.fitBestModel)
```

```
##
## Call:
## lm(formula = AmountSpent ~ poly(Salary, degree = 6) + poly(Children,
##     degree = 2) + History + Age + Location, data = train.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1339.14  -299.03   -37.28   233.64  3020.53
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1210.64      41.24  29.354  < 2e-16 ***
## poly(Salary, degree = 6)1    17848.16     911.02  19.591  < 2e-16 ***
## poly(Salary, degree = 6)2      -22.11     645.56  -0.034  0.97268
## poly(Salary, degree = 6)3     -125.33     573.36  -0.219  0.82702
## poly(Salary, degree = 6)4    -2440.98     553.29  -4.412 1.14e-05 ***
## poly(Salary, degree = 6)5    -1458.60     554.59  -2.630  0.00867 **
## poly(Salary, degree = 6)6     -316.77     558.13  -0.568  0.57047
## poly(Children, degree = 2)1 -5534.32     666.94  -8.298 3.46e-16 ***
## poly(Children, degree = 2)2  -415.74     573.92  -0.724  0.46900
## HistoryLow                    -407.88      54.12  -7.537 1.09e-13 ***
## HistoryMedium                 -387.73      51.87  -7.475 1.71e-13 ***
## HistoryHigh                    173.83      57.51   3.023  0.00257 **
## AgeOld                          53.44      53.14   1.006  0.31480
## AgeYoung                       -35.82      58.24  -0.615  0.53867
## LocationFar                    472.90      40.38  11.712  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 545.8 on 985 degrees of freedom
## Multiple R-squared:  0.682,  Adjusted R-squared:  0.6774
## F-statistic: 150.9 on 14 and 985 DF,  p-value: < 2.2e-16
```

```
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~   poly(Salary, degree = 6)+
            Location  + poly(Children, degree = 2) + Age+
            History  , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] -1.009636
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 551.8345
```

Removing Catalogs also shows an impact. However the impact is lower than removing Salary.

3. Removing Location

```r
poly.fitBestModel <- lm(AmountSpent ~ poly(Salary, degree = 6)+
                          poly(Children, degree = 2)+ History +
                        Age + poly(Catalogs, degree = 3), data = train.clean)
summary(poly.fitBestModel)
```

```
##
## Call:
## lm(formula = AmountSpent ~ poly(Salary, degree = 6) + poly(Children,
##     degree = 2) + History + Age + poly(Catalogs, degree = 3),
##     data = train.clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1463.2  -308.3   -41.3   245.3  3068.5
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1354.963     37.634  36.004  < 2e-16 ***
## poly(Salary, degree = 6)1   16046.453    840.758  19.086  < 2e-16 ***
## poly(Salary, degree = 6)2     580.449    606.821   0.957  0.33903
## poly(Salary, degree = 6)3      98.900    539.836   0.183  0.85468
## poly(Salary, degree = 6)4   -2046.300    523.189  -3.911 9.81e-05 ***
## poly(Salary, degree = 6)5   -1693.053    522.234  -3.242  0.00123 **
## poly(Salary, degree = 6)6    -738.791    525.003  -1.407  0.15968
## poly(Children, degree = 2)1 -4196.917    621.663  -6.751 2.51e-11 ***
## poly(Children, degree = 2)2  -684.091    540.882  -1.265  0.20625
## HistoryLow                   -449.576     50.456  -8.910  < 2e-16 ***
## HistoryMedium                -437.978     48.999  -8.938  < 2e-16 ***
## HistoryHigh                   156.504     53.589   2.920  0.00358 **
## AgeOld                         94.412     50.073   1.885  0.05966 .
## AgeYoung                       -4.187     54.883  -0.076  0.93920
## poly(Catalogs, degree = 3)1  9111.152    544.335  16.738  < 2e-16 ***
## poly(Catalogs, degree = 3)2   662.950    518.581   1.278  0.20141
## poly(Catalogs, degree = 3)3  -628.685    518.496  -1.213  0.22561
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 513.8 on 983 degrees of freedom
## Multiple R-squared:  0.7188, Adjusted R-squared:  0.7142
## F-statistic:   157 on 16 and 983 DF,  p-value: < 2.2e-16
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~   poly(Salary, degree = 6)+
            poly(Catalogs, degree = 3)  + poly(Children, degree = 2) + Age+
            History  , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] 3.355892
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 532.6443
```

Yet again the we have a considerably lower model but better than the one in which Salary or Catalogs was removed.

4. Removing Children

```r
poly.fitBestModel <- lm(AmountSpent ~ poly(Salary, degree = 6)+
                          Location+ History +
                      Age + poly(Catalogs, degree = 3), data = train.clean)
summary(poly.fitBestModel)
```

```
##
## Call:
## lm(formula = AmountSpent ~ poly(Salary, degree = 6) + Location +
##     History + Age + poly(Catalogs, degree = 3), data = train.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1589.56  -311.59   -27.56   258.36  2859.26
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1221.860     37.537  32.551  < 2e-16 ***
## poly(Salary, degree = 6)1  16042.968    804.672  19.937  < 2e-16 ***
## poly(Salary, degree = 6)2    394.478    587.669   0.671 0.502213
## poly(Salary, degree = 6)3    120.437    522.792   0.230 0.817849
## poly(Salary, degree = 6)4  -1899.924    509.112  -3.732 0.000201 ***
## poly(Salary, degree = 6)5  -1486.436    505.949  -2.938 0.003381 **
## poly(Salary, degree = 6)6   -651.950    510.755  -1.276 0.202100
## LocationFar                  378.547     36.668  10.324  < 2e-16 ***
```

```
## HistoryLow                      -485.907     47.581 -10.212  < 2e-16 ***
## HistoryMedium                   -402.185     47.409  -8.483  < 2e-16 ***
## HistoryHigh                      147.522     51.245   2.879 0.004078 **
## AgeOld                           205.523     44.570   4.611 4.53e-06 ***
## AgeYoung                           8.371     53.203   0.157 0.875012
## poly(Catalogs, degree = 3)1  8778.042    530.759  16.539  < 2e-16 ***
## poly(Catalogs, degree = 3)2   429.304    503.531   0.853 0.394096
## poly(Catalogs, degree = 3)3  -565.484    504.144  -1.122 0.262276
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 499.8 on 984 degrees of freedom
## Multiple R-squared:  0.7336, Adjusted R-squared:  0.7296
## F-statistic: 180.7 on 15 and 984 DF,  p-value: < 2.2e-16
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~   poly(Salary, degree = 6)+
            poly(Catalogs, degree = 3)  + Location + Age+
            History  , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] 2.811774
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 516.4043
```

Further reduced impact than our cases before

5. Removing History

```r
poly.fitBestModel <- lm(AmountSpent ~ poly(Salary, degree = 6)+
                          Location+ poly(Children, degree = 2) +
                        Age + poly(Catalogs, degree = 3), data = train.clean)
summary(poly.fitBestModel)
```

```
##
## Call:
## lm(formula = AmountSpent ~ poly(Salary, degree = 6) + Location +
##     poly(Children, degree = 2) + Age + poly(Catalogs, degree = 3),
##     data = train.clean)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1622.55  -317.97   -26.36   223.16  2754.09
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 1042.62      28.91  36.058  < 2e-16 ***
## poly(Salary, degree = 6)1   20742.68     689.33  30.091  < 2e-16 ***
## poly(Salary, degree = 6)2     168.36     565.30   0.298  0.76590
## poly(Salary, degree = 6)3    -780.23     516.96  -1.509  0.13155
## poly(Salary, degree = 6)4   -1263.63     513.29  -2.462  0.01399 *
## poly(Salary, degree = 6)5   -1641.28     513.77  -3.195  0.00144 **
## poly(Salary, degree = 6)6    -640.05     516.34  -1.240  0.21542
## LocationFar                   498.56      36.02  13.842  < 2e-16 ***
## poly(Children, degree = 2)1 -6656.56     564.85 -11.785  < 2e-16 ***
## poly(Children, degree = 2)2  -216.34     531.34  -0.407  0.68398
## AgeOld                         64.59      49.50   1.305  0.19218
## AgeYoung                       56.89      54.08   1.052  0.29310
## poly(Catalogs, degree = 3)1  8927.61     531.10  16.810  < 2e-16 ***
## poly(Catalogs, degree = 3)2   919.60     513.31   1.792  0.07351 .
## poly(Catalogs, degree = 3)3  -778.36     511.39  -1.522  0.12832
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 509.8 on 985 degrees of freedom
## Multiple R-squared:  0.7225, Adjusted R-squared:  0.7186
## F-statistic: 183.2 on 14 and 985 DF,  p-value: < 2.2e-16
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~   poly(Salary, degree = 6)+
            poly(Catalogs, degree = 3)  + Location + Age+
            poly(Children, degree = 2)  , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
me
```

```
## [1] 2.783248
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 524.2416
```

6. Removing Age

```r
poly.fitBestModel <- lm(AmountSpent ~ poly(Salary, degree = 6)+
                          Location+ poly(Children, degree = 2) +
                      History + poly(Catalogs, degree = 3), data = train.clean)
summary(poly.fitBestModel)
```

```
##
## Call:
## lm(formula = AmountSpent ~ poly(Salary, degree = 6) + Location +
##     poly(Children, degree = 2) + History + poly(Catalogs, degree = 3),
##     data = train.clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1333.83  -289.71   -16.14   227.22  2841.66
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  1256.99      29.43  42.707  < 2e-16 ***
## poly(Salary, degree = 6)1   18020.20     661.32  27.249  < 2e-16 ***
## poly(Salary, degree = 6)2     164.63     518.42   0.318 0.750886
## poly(Salary, degree = 6)3      72.26     500.58   0.144 0.885255
## poly(Salary, degree = 6)4   -1878.78     489.71  -3.836 0.000133 ***
## poly(Salary, degree = 6)5   -1561.51     485.85  -3.214 0.001352 **
## poly(Salary, degree = 6)6    -481.49     486.57  -0.990 0.322631
## LocationFar                   427.12      35.67  11.974  < 2e-16 ***
## poly(Children, degree = 2)1 -5665.44     550.59 -10.290  < 2e-16 ***
## poly(Children, degree = 2)2  -325.28     489.20  -0.665 0.506257
## HistoryLow                   -352.12      47.31  -7.443 2.15e-13 ***
## HistoryMedium                -415.60      45.58  -9.118  < 2e-16 ***
## HistoryHigh                    19.64      51.42   0.382 0.702545
## poly(Catalogs, degree = 3)1  8611.28     510.39  16.872  < 2e-16 ***
## poly(Catalogs, degree = 3)2   668.15     485.22   1.377 0.168825
## poly(Catalogs, degree = 3)3  -547.69     485.08  -1.129 0.259141
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 480.8 on 984 degrees of freedom
## Multiple R-squared:  0.7535, Adjusted R-squared:  0.7497
## F-statistic: 200.5 on 15 and 984 DF,  p-value: < 2.2e-16
```

```r
n = length(train.clean$AmountSpent)
error = dim(n)
for (k in 1:n) {
  train1 = c(1:n)
  train2 = train1[train1!=k] ## pick elements that are different from k
  m2 = lm(AmountSpent ~   poly(Salary, degree = 6)+
            poly(Catalogs, degree = 3)  + Location + History+
            poly(Children, degree = 2)  , data=train.clean[train2 ,])
  pred = predict(m2, newdat=train.clean[-train2 ,])
  obs = train.clean$AmountSpent[-train2]
  error[k] = obs-pred
}
me=mean(error)
```

```r
me
```

```
## [1] 0.9476679
```

```r
rmse=sqrt(mean(error^2))
rmse
```

```
## [1] 488.1441
```

Removing Age has the least impact.

Thus we can see confirm that Salary is the most important predictor followed by Catalogs and then the others.

Conclusion

This assignment showed a glimpse of how we can go over a step of process in understanding our data and interpreting it. This understanding can be further enhanced by trying to uncover patterns and valuable information which is not obvious. Summarising:

1. Initial anaysis involves understanding various variables and the values they hold. Highlighting missing values.
2. Finding correlation or any other relations between variables
3. Analysing a descriptive statistics of the variables
4. Performing various comparisons using plots, tables etc.
5. Building a baseine model and checking model fit, accuracy and performance.
6. Improving on the model using variable selection, cross-validation etc
7. Identifying importance of each variable.
8. Continuously imrpoving the model using alternative apporaches like non-linear, parametric, mixed -model etc.

References:

1. Oneway Test - https://ww2.coastal.edu/kingw/statistics/R-tutorials/oneway.html
2. Factors - http://www.ats.ucla.edu/stat/r/modules/dummy_vars.htm
3. Regression - http://tutorials.iq.harvard.edu/R/Rstatistics/Rstatistics.html