EDA Class Data

Sriya Cheedella

September 24, 2019

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
alldata <- read.csv("C:/Users/Ujvala/Downloads/2019CMDA2014_ClassSurvey_Complete (1).csv", heade
r=TRUE)
curdata <- as.data.frame(cbind(alldata$Q47_1, alldata$Q21_1, alldata$Q22_1, alldata$Q23_1, allda
ta$Q24_1, alldata$Q25_1))
enddata <- curdata %>% filter_all(all_vars(!is.na(.)))
colnames(enddata) <- c("Excite", "CSLove", "StatLove", "MathLove", "DSLove", "VTLove")
```

Let's check the assumptions for multiple linear regression!

We should test for normality, but violating this condition is not a major issue so we will ignore it now for simplicity. Plus, after the cleaning the data we have a sample size of 21 so it's safe to assume normality with the Central Limit Theorem.

Let's see if there is linearity:

```
library(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

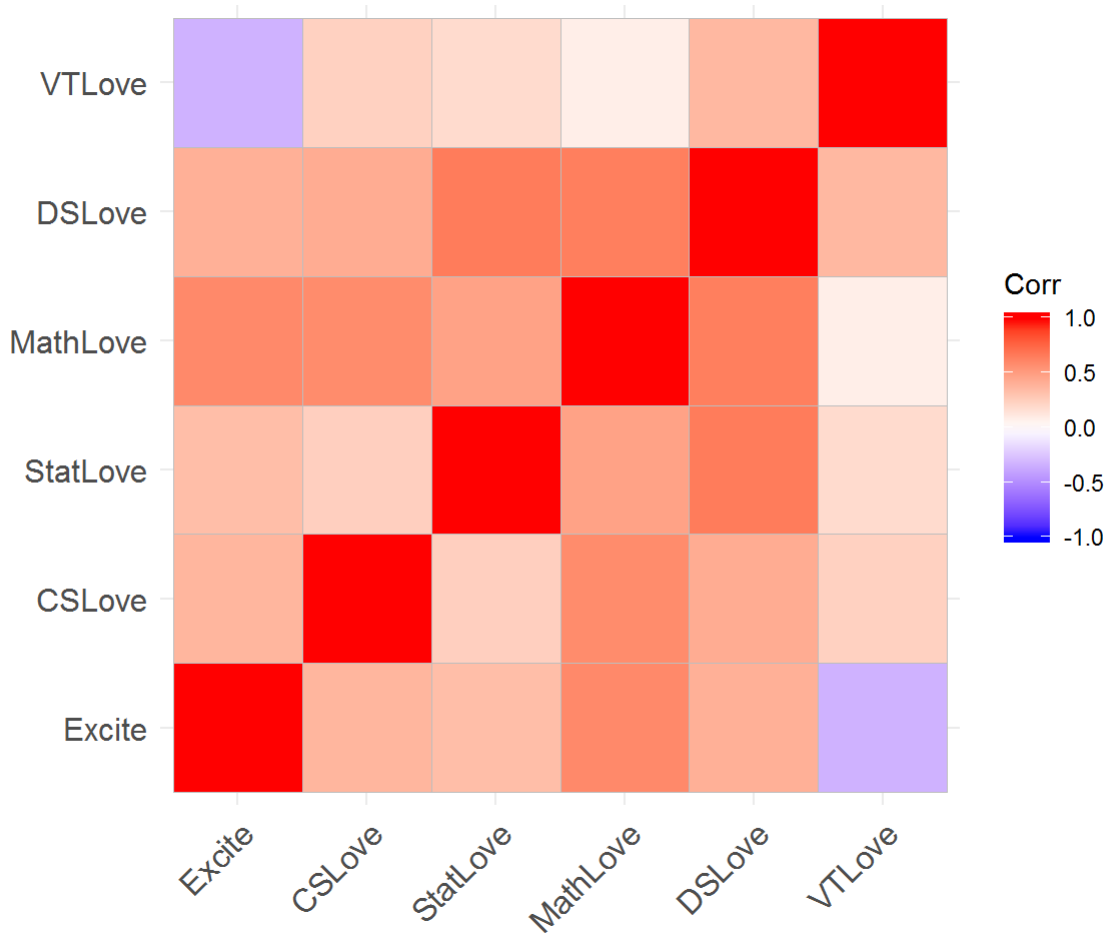
```
library(ggplot2)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 3.4.4
```

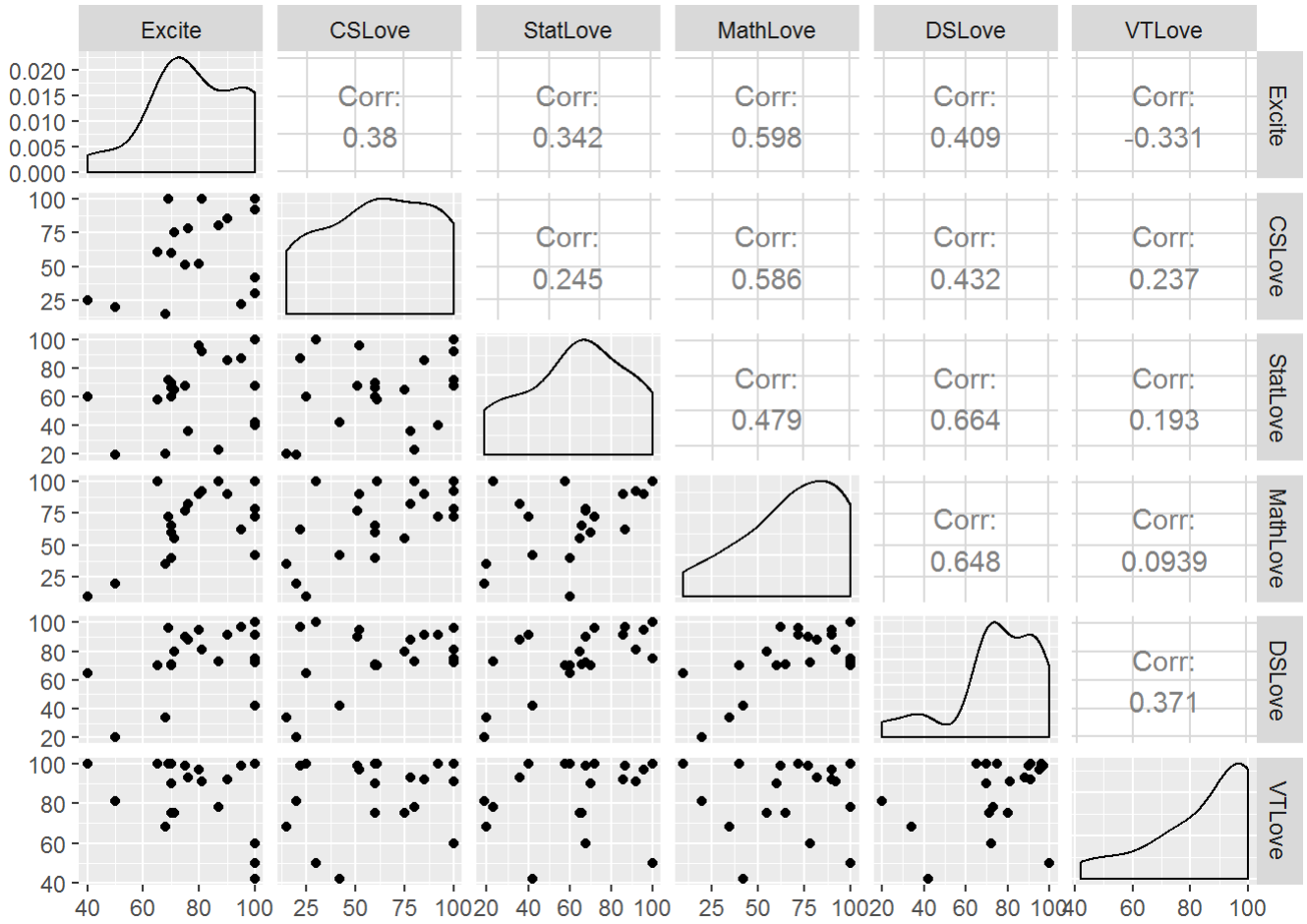
```
##
## Attaching package: 'GGally'
```

```
## The following object is masked from 'package:dplyr':
##
##     nasa
```

```
cor <- cor(enddata)
cormat <- cor_pmat(enddata)
ggcorrplot(cor)
```




```
ggpairs(enddata)
```



There are a lot of variables that aren't linearly associated, but we'll find the most significant variables and utilize the ones that make the best model.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.4.4
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
fitall <- lm(Excite ~ CSLove + StatLove + MathLove + DSLove + VTLove, data=enddata)
summary(fitall)
```

```
##
## Call:
## lm(formula = Excite ~ CSLove + StatLove + MathLove + DSLove +
##     VTLove, data = enddata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.849  -6.484  -2.319   3.317  22.778
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 81.11900   15.49162   5.236 0.000101 ***
## CSLove       0.10032    0.12729   0.788 0.442883
## StatLove     0.03938    0.15197   0.259 0.799067
## MathLove     0.22042    0.16383   1.345 0.198490
## DSLove       0.20812    0.22814   0.912 0.376071
## VTLove      -0.48873    0.18232  -2.681 0.017109 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.86 on 15 degrees of freedom
## Multiple R-squared:  0.5695, Adjusted R-squared:  0.426
## F-statistic: 3.968 on 5 and 15 DF,  p-value: 0.01712
```

```
fitnone <- lm(Excite ~ 1, data=enddata)
summary(fitnone)
```

```
##
## Call:
## lm(formula = Excite ~ 1, data = enddata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.905  -8.905  -2.905  16.095  21.095
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   78.905      3.704    21.3 3.23e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.97 on 20 degrees of freedom
```

```
bestfit = stepAIC(fitnone, direction = "both", scope = list(upper=fitall, lower=fitnone))
```

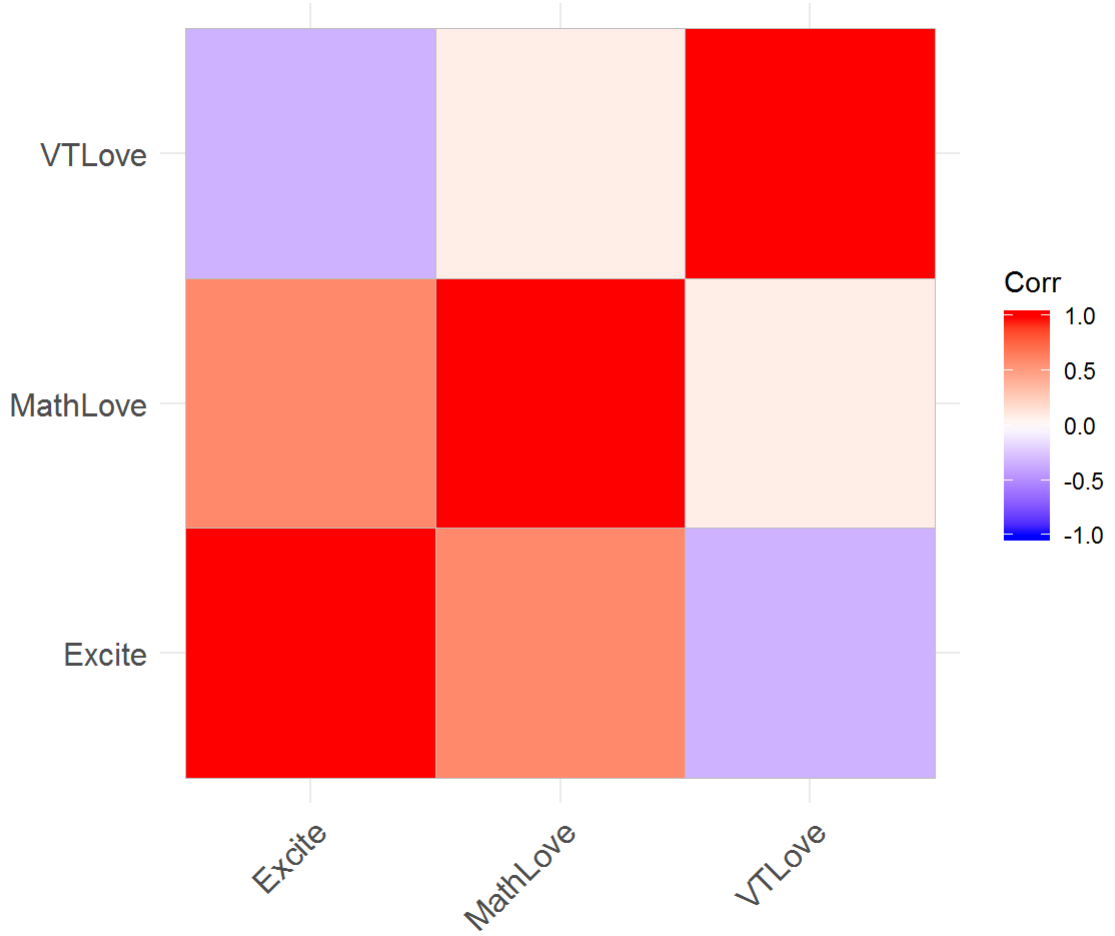
```
## Start:  AIC=119.9
## Excite ~ 1
##
##            Df Sum of Sq    RSS    AIC
## + MathLove  1   2059.59 3702.2 112.61
## + DSLove    1    964.01 4797.8 118.06
## + CSLove    1    834.05 4927.8 118.62
## + StatLove  1    674.44 5087.4 119.29
## + VTLove    1    630.90 5130.9 119.47
## <none>                  5761.8 119.90
##
## Step:  AIC=112.62
## Excite ~ MathLove
##
##            Df Sum of Sq    RSS    AIC
## + VTLove    1    870.90 2831.3 108.98
## <none>                  3702.2 112.61
## + StatLove  1     23.03 3679.2 114.48
## + CSLove    1      7.96 3694.3 114.57
## + DSLove    1      4.66 3697.6 114.59
## - MathLove  1   2059.59 5761.8 119.90
##
## Step:  AIC=108.98
## Excite ~ MathLove + VTLove
##
##            Df Sum of Sq    RSS    AIC
## <none>                  2831.3 108.98
## + DSLove    1    242.61 2588.7 109.10
## + StatLove  1     98.84 2732.5 110.24
## + CSLove    1     94.65 2736.7 110.27
## - VTLove    1    870.90 3702.2 112.61
## - MathLove  1   2299.59 5130.9 119.47
```

```
formula(bestfit)
```

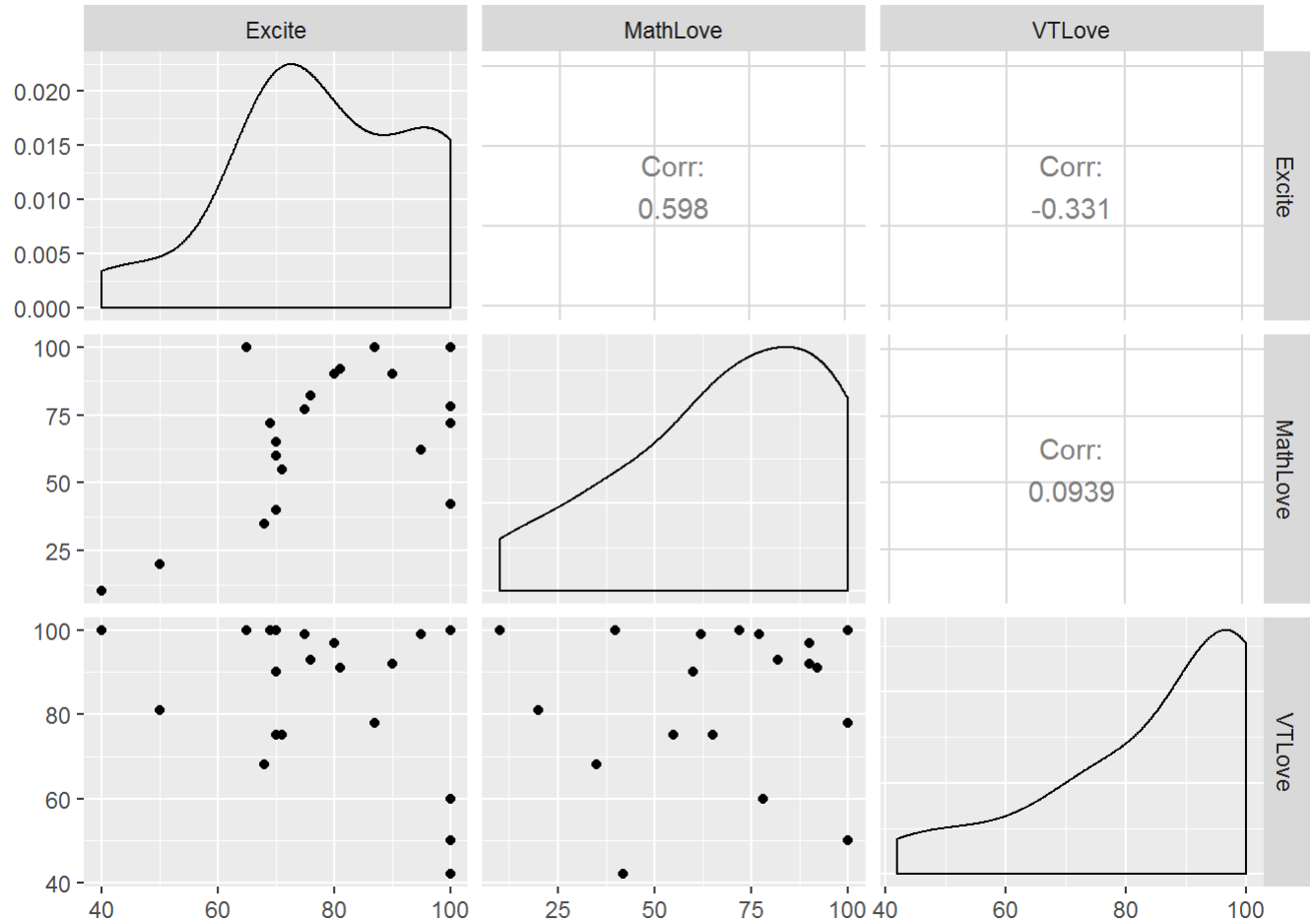
```
## Excite ~ MathLove + VTLove
```

We now see that how much students like math and Virginia Tech correlate best with how excited students are for this class. We will now create a new correlation matrix with those variables for simplicity.

```
sigdata <- enddata %>% dplyr::select(Excite, MathLove, VTLove)
sigcor <- cor(sigdata)
sigcormat <- cor_pmat(sigdata)
ggcorrplot(sigcor)
```




```
ggpairs(sigdata)
```



Even though the correlations aren't significant, these are the best we can use so we will continue with this model.

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 3.4.4
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(bestfit)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  bestfit
## BP = 1.0202, df = 2, p-value = 0.6004
```

With a p-value of 0.6004, there is slight evidence of heteroscedacity but we will continue regardless since it doesn't seem too significant.

Normally we would check for serial correlation between points, but each person's response for the questions are independent of each other and each person's liking for the particular subject are independent as well. So we can assume the errors aren't related and one observation does not increase the probability of another observation.

The test for serial correlation should only be done on time dependent data which this data isn't. It also shouldn't be used on data that hasn't been cleaned since it excludes data points that may be vital.

We will now conduct multiple linear regression and interpret the results!

```
summary(bestfit)
```

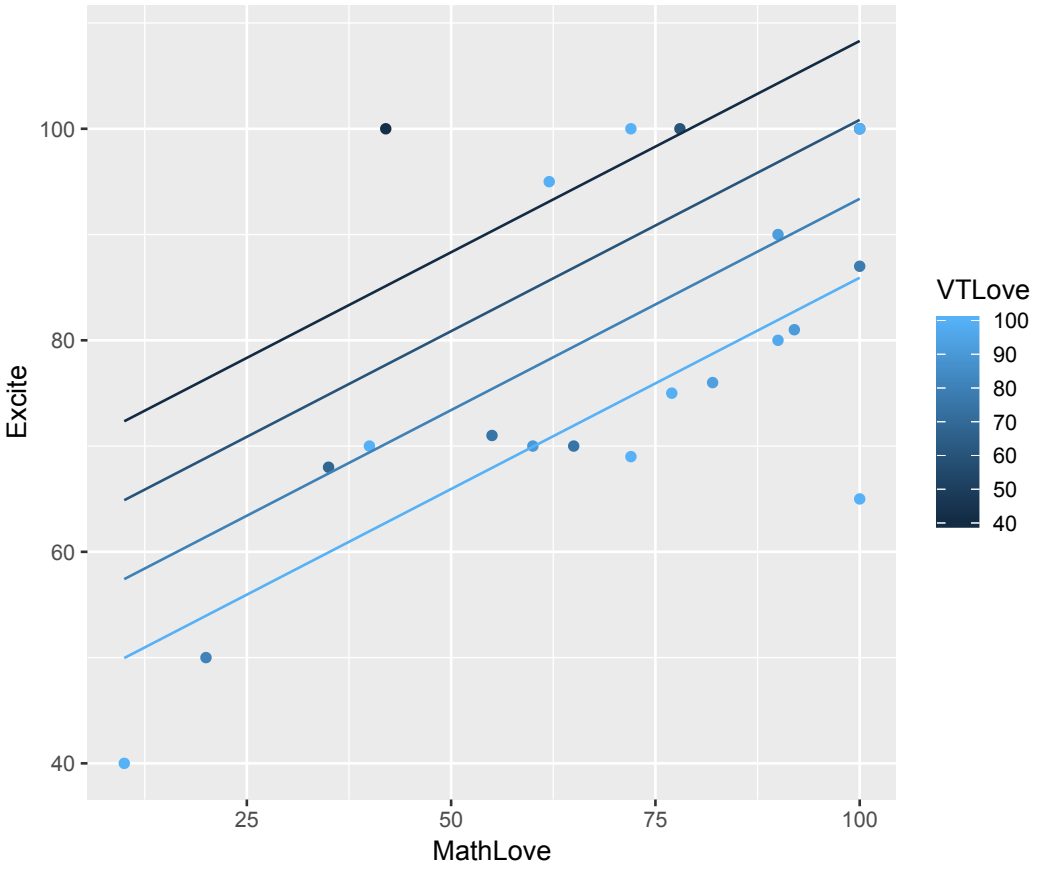
```
##
## Call:
## lm(formula = Excite ~ MathLove + VTLove, data = enddata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.916  -6.264  -3.886   7.950  25.271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  83.2733    14.9475   5.571 2.74e-05 ***
## MathLove      0.3995     0.1045   3.824  0.00124 **
## VTLove       -0.3731     0.1586  -2.353  0.03019 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.54 on 18 degrees of freedom
## Multiple R-squared:  0.5086, Adjusted R-squared:  0.454
## F-statistic: 9.315 on 2 and 18 DF,  p-value: 0.001671
```

```
library(ggiraphExtra)
```

```
## Warning: package 'ggiraphExtra' was built under R version 3.4.4
```

```
ggPredict(bestfit, se=FALSE, interactive = TRUE)
```

```
## Warning: package 'gdtools' was built under R version 3.4.4
```



The p-values for each variable are significant (less than 0.05) so there is no evidence of multicollinearity which is good! The R^2 value is not very high (0.454) but it's considerably decent since data regularly isn't super clean.

The equation is Excite = 83.2733 + 0.3995(MathLove) - 0.03731(VTLove). When there is 0 liking towards for math and VT the excitement is 83.2733 which is fairly high! But the more someone likes Virginia Tech, they are slightly less excited about this class while the more a student likes math they are signficantly more excited for this class.