# Multiple Regression Diagnostics and Simultaneous Multiple Regression Tutorial

Professor Christopher J. Schmank

2025-02-24

## Multiple Regression Diagnostics Demonstration

### **Applied Prompt**:

A local resident recently joined their a newly developed charitable giving program and is interested in finding out more about `prosocial behavior` among group members. Without knowing much about the literature, they developed a short survey to give to a nearby school district to find out.

```
**Research Question**: What influences the amount of money given to a program for community outreach.
```

Ultimately, they must report their findings to their program coordinator and a statistician, who is also on the board.

### **Variables in Data**:

`ID` — identifying number for each observation

`Belief` — belief that charitable giving has a positive effect on a scale of 1-10

`Need` — rating of perceived amount of need required in the community on a scale of 1-10

`Interest` — rating of level of interest in the project on a scale of 1-10

`Happy` — rating of happiness felt when making donations on a scale of 1-10

`Amount` — amount given from 0 - 10 dollars

### **For all variables**:

A value of `99` implies that a response was missing for that variable and case

### Load in Libraries

```
library(mice)
library(MVN)
library(lmtest)
library(apaTables)
library(psych)
library(jmv)
library(tidyverse)
```

### Load in Data

```
#Load in the data frame into an object `dat`

dat <- read.csv("RegDiagnostics2.csv")

#View(dat)
```

### Change Missing Data Labels

```
#Remove all `99` values from data frame and replace with `NA`s
# MAKE SURE YOU DO NOT HAVE AN ID SET TO 99 OR IT WILL REMOVE THIS CASE NUMBER!!

dat[dat=="99"] <- NA
```

## Exploratory Data Analysis

### Descriptive Statistics

```
# Descriptive statistics  (Default: No Removal of Missing Data)
describe(dat)
```

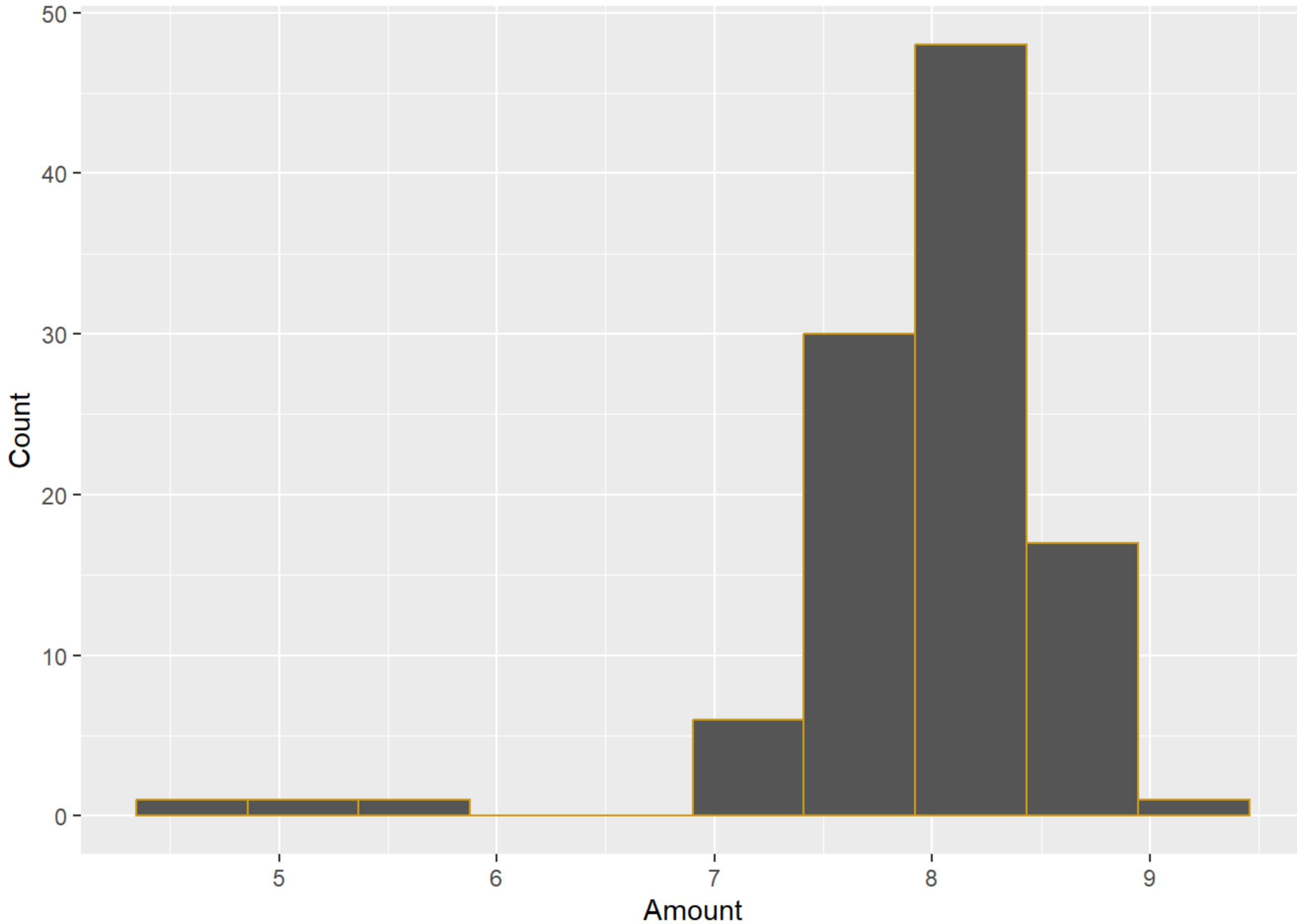
```
##          vars   n   mean    sd median trimmed   mad   min   max range  skew
## ID          1 106 152.50 30.74  152.5  152.50 39.29 100.0 205.0 105.0  0.00
## Amount      2 105   8.03  0.66    8.2    8.10  0.44   4.5   9.1   4.6 -2.74
## Need        3 104   5.83  1.77    5.9    5.90  2.08   1.7   9.5   7.8 -0.32
## Interest    4 104   6.14  0.75    6.2    6.16  0.89   4.4   7.5   3.1 -0.25
## Happy       5 106   8.84  0.85    9.1    8.92  0.59   3.4  10.0   6.6 -2.67
## Belief      6 104   7.53  0.38    7.5    7.53  0.30   6.1   8.3   2.2 -0.56
##          kurtosis   se
## ID          -1.23 2.99
## Amount      11.27 0.06
## Need        -0.71 0.17
## Interest    -0.65 0.07
## Happy       13.81 0.08
## Belief       1.63 0.04
```

```
# Descriptive statistics  (Removal of Missing Data)
describe(dat,
         na.rm=FALSE)
```

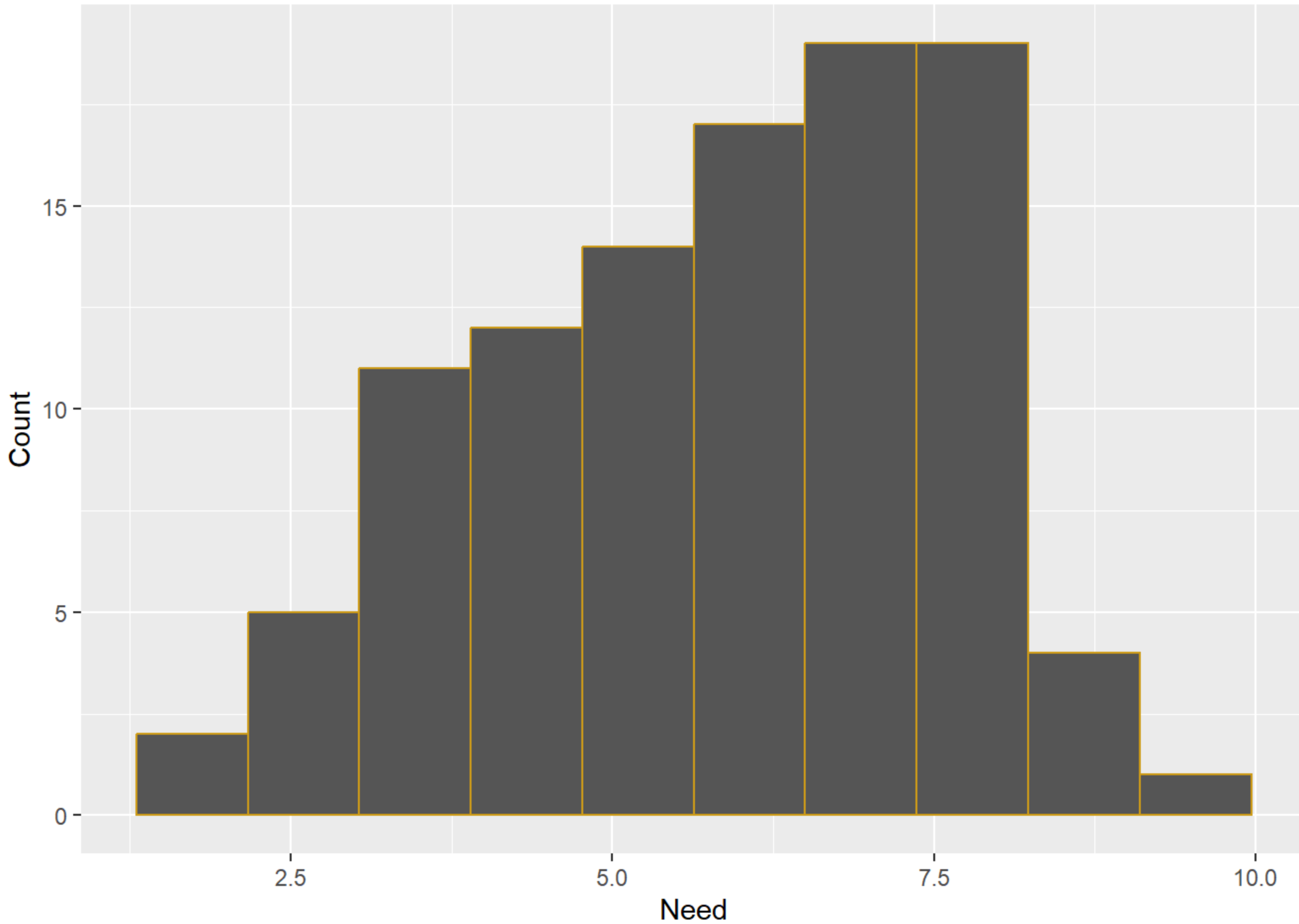
```
##          vars   n   mean    sd median trimmed   mad   min   max range  skew
## ID          1 100 153.46 30.51  153.5  153.62 38.55 100.0 205.0 105.0 -0.04
## Amount      2 100   8.06  0.60    8.2    8.11  0.44   4.5   9.1   4.6 -2.66
## Need        3 100   5.78  1.78    5.9    5.85  2.08   1.7   9.5   7.8 -0.28
## Interest    4 100   6.12  0.75    6.2    6.14  0.89   4.4   7.5   3.1 -0.25
## Happy       5 100   8.86  0.85    9.1    8.96  0.59   3.4  10.0   6.6 -2.84
## Belief      6 100   7.54  0.38    7.5    7.54  0.30   6.1   8.3   2.2 -0.61
##          kurtosis   se
## ID          -1.21 3.05
## Amount      12.74 0.06
## Need        -0.75 0.18
## Interest    -0.68 0.07
## Happy       14.99 0.08
## Belief       1.78 0.04
```

## Generate Univariate and Bivariate Visualizations

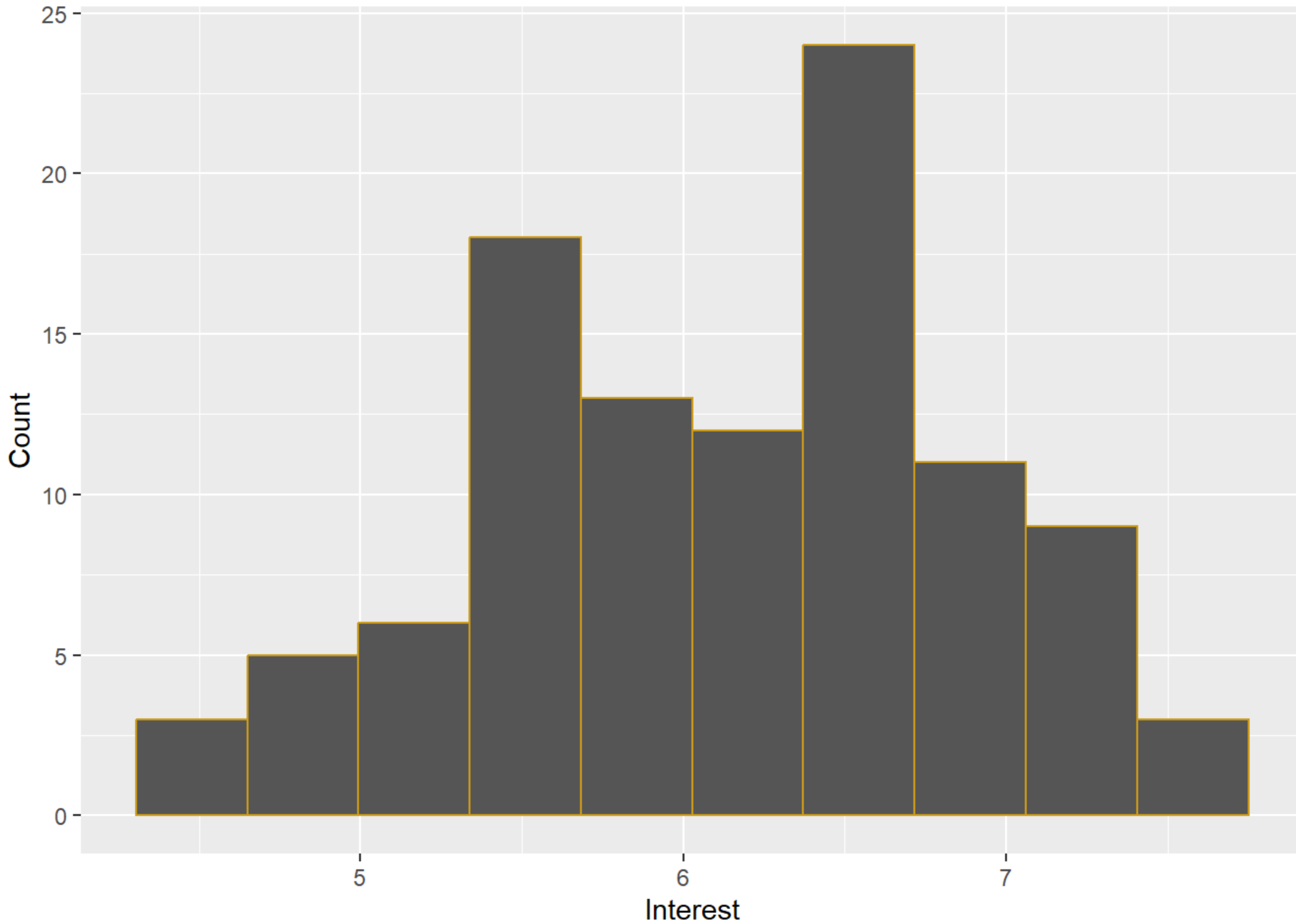
```
# Univariate Histograms of Continuous Variables

ggplot(data = dat,
       mapping = aes(x = Amount)) +
  geom_histogram(bins = 10, color = "goldenrod3") +
  labs(y = "Count", x = "Amount")
```



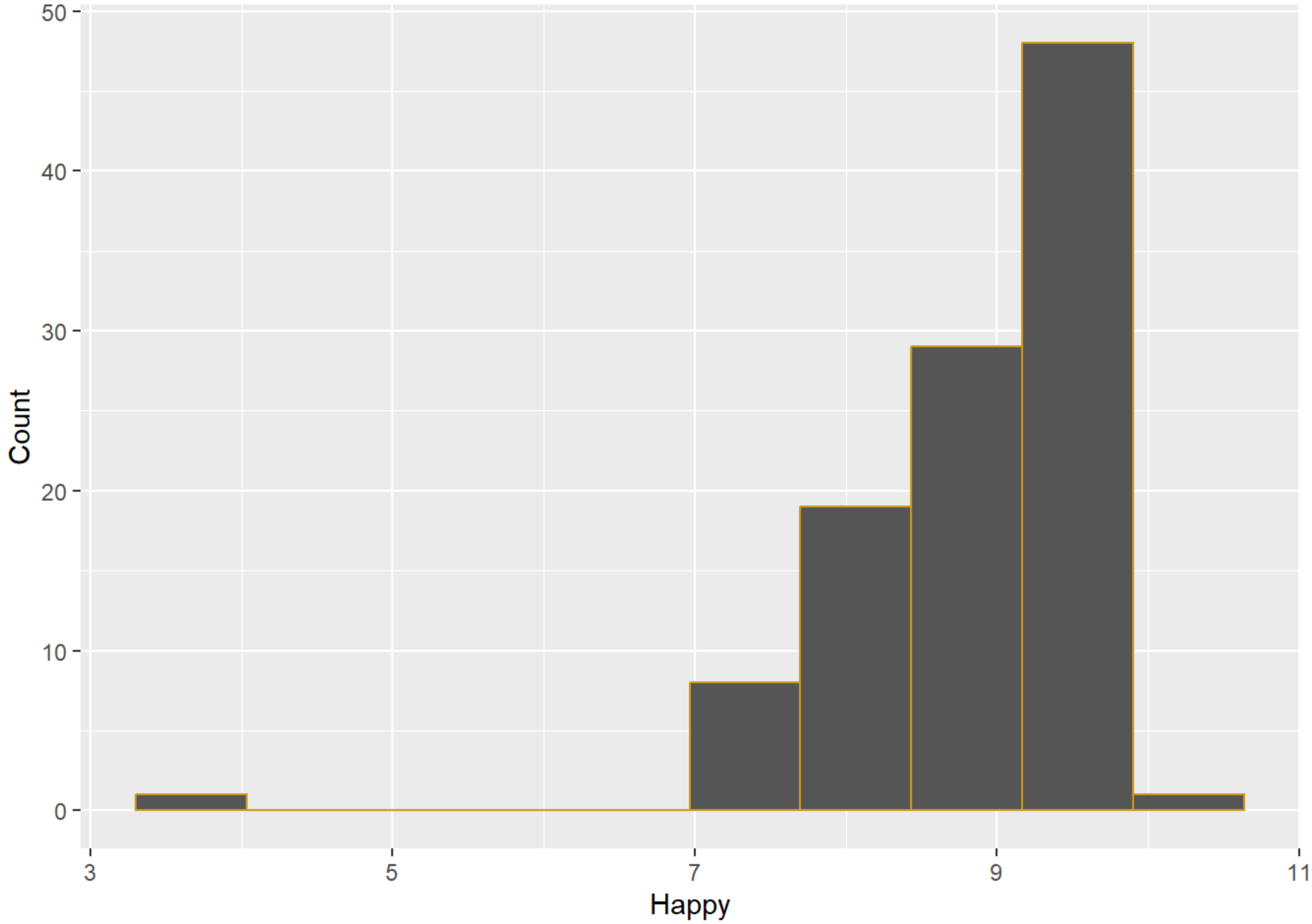

```
ggplot(data = dat,
       mapping = aes(x = Need)) +
  geom_histogram(bins = 10, color = "goldenrod3") +
  labs(y = "Count", x = "Need")
```



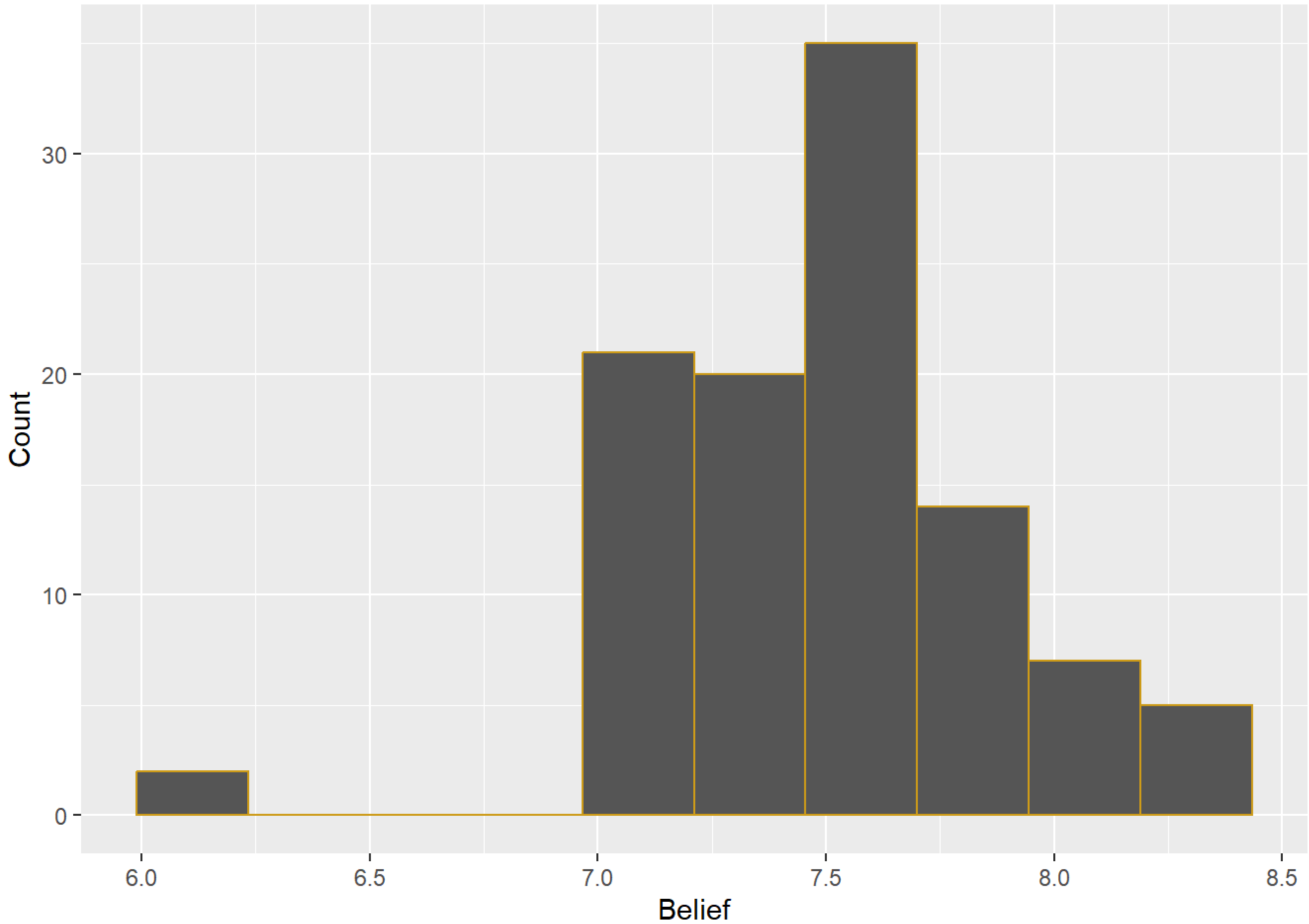

```
ggplot(data = dat,
       mapping = aes(x = Interest)) +
  geom_histogram(bins = 10, color = "goldenrod3") +
  labs(y = "Count", x = "Interest")
```



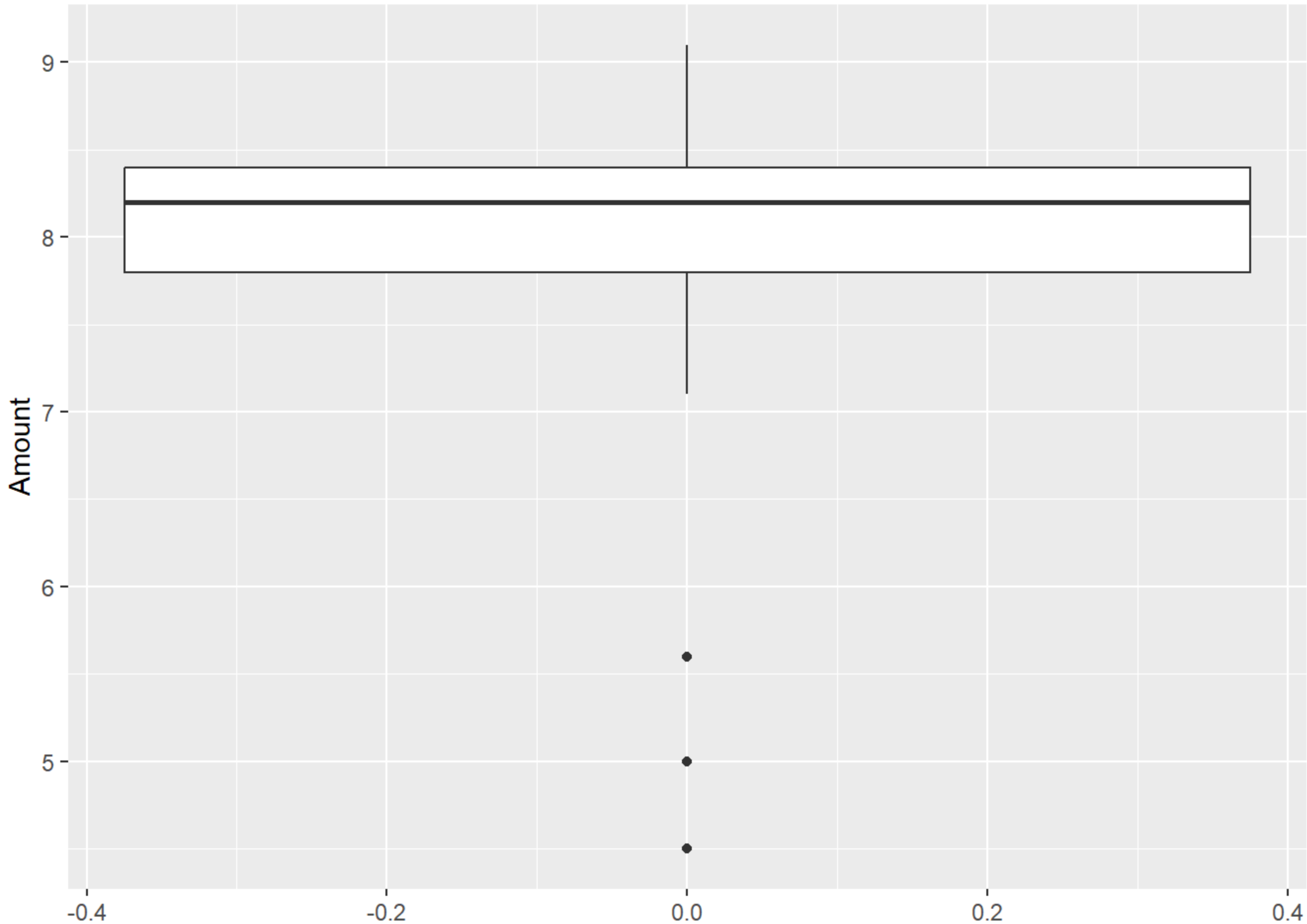

```
ggplot(data = dat,
       mapping = aes(x = Happy)) +
  geom_histogram(bins = 10, color = "goldenrod3") +
  labs(y = "Count", x = "Happy")
```



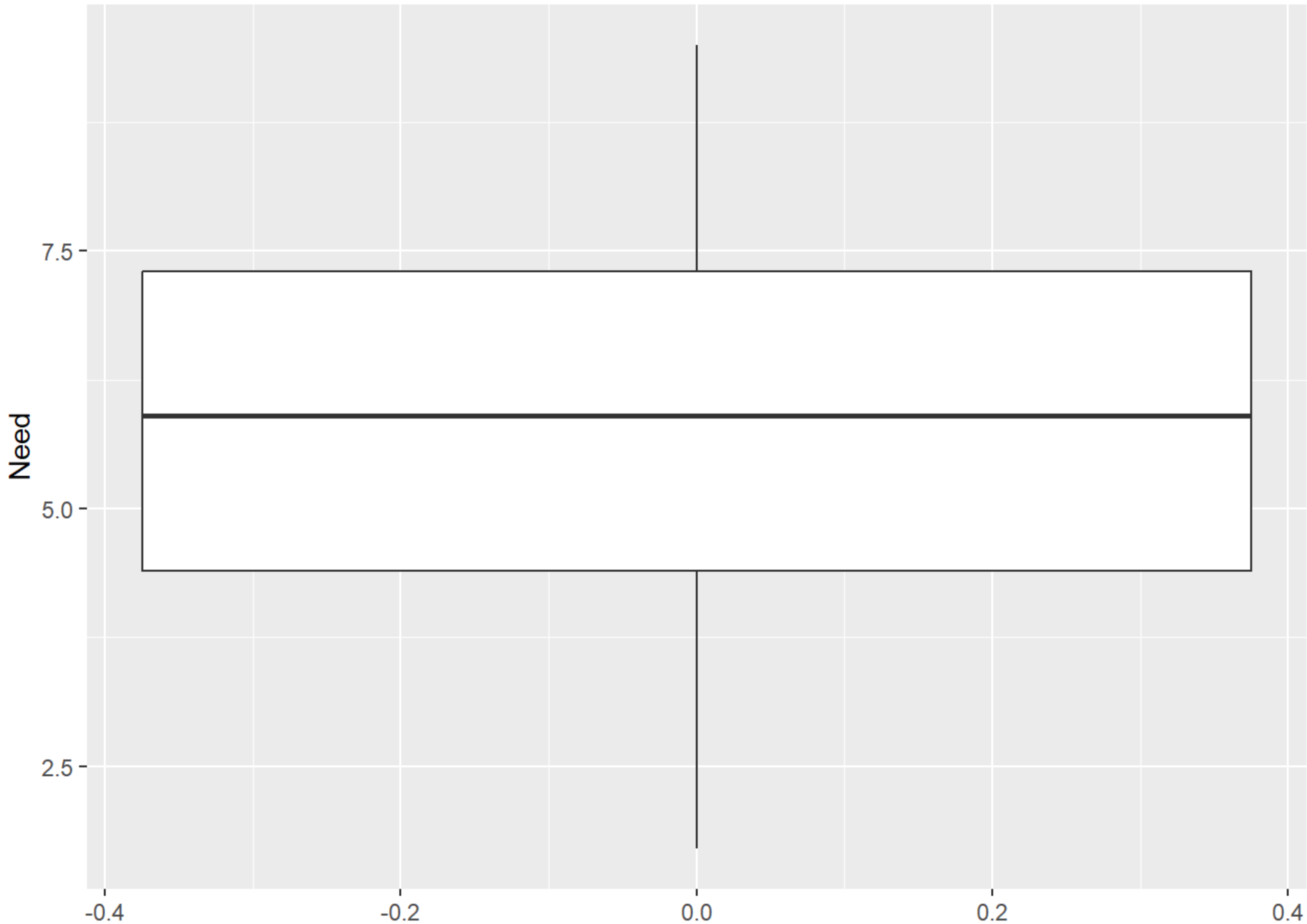

```
ggplot(data = dat,
       mapping = aes(x = Belief)) +
  geom_histogram(bins = 10, color = "goldenrod3") +
  labs(y = "Count", x = "Belief")
```



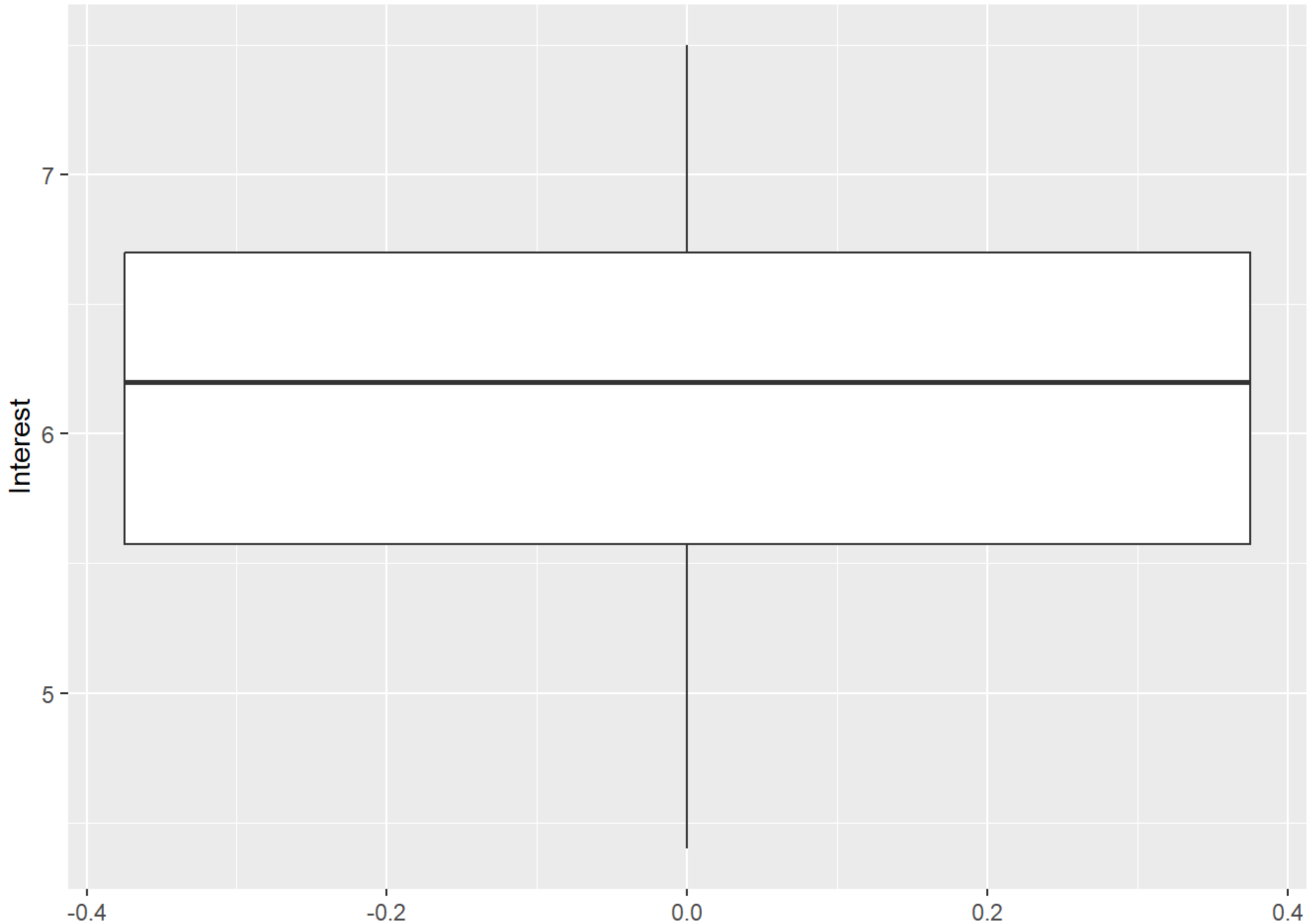

```
# Univariate Boxplots of Continuous Variables

ggplot(data = dat,
       mapping = aes(y = Amount)) +
  geom_boxplot() +
    labs(y = "Amount")
```



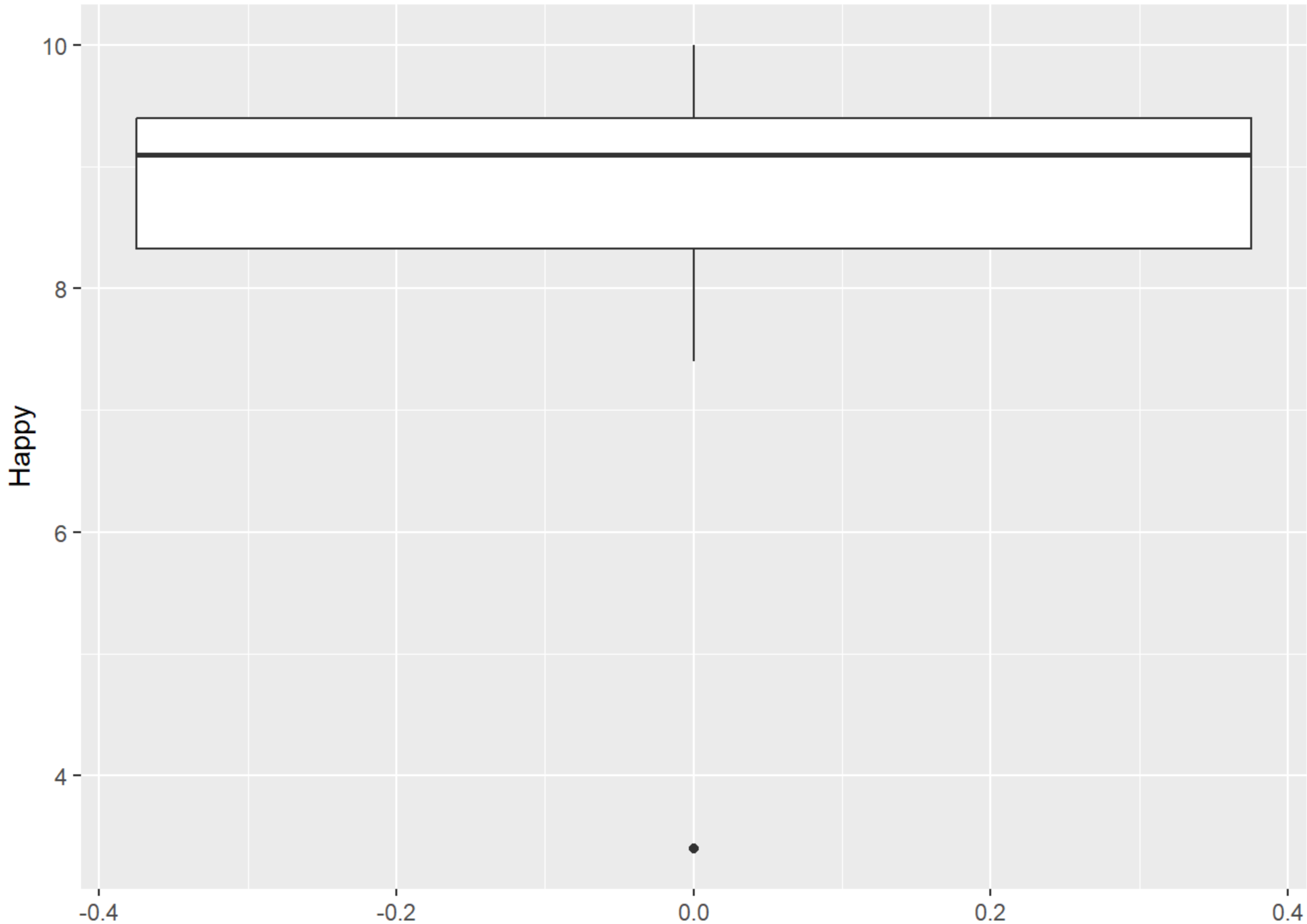

```
ggplot(data = dat,
       mapping = aes(y = Need)) +
  geom_boxplot() +
    labs(y = "Need")
```



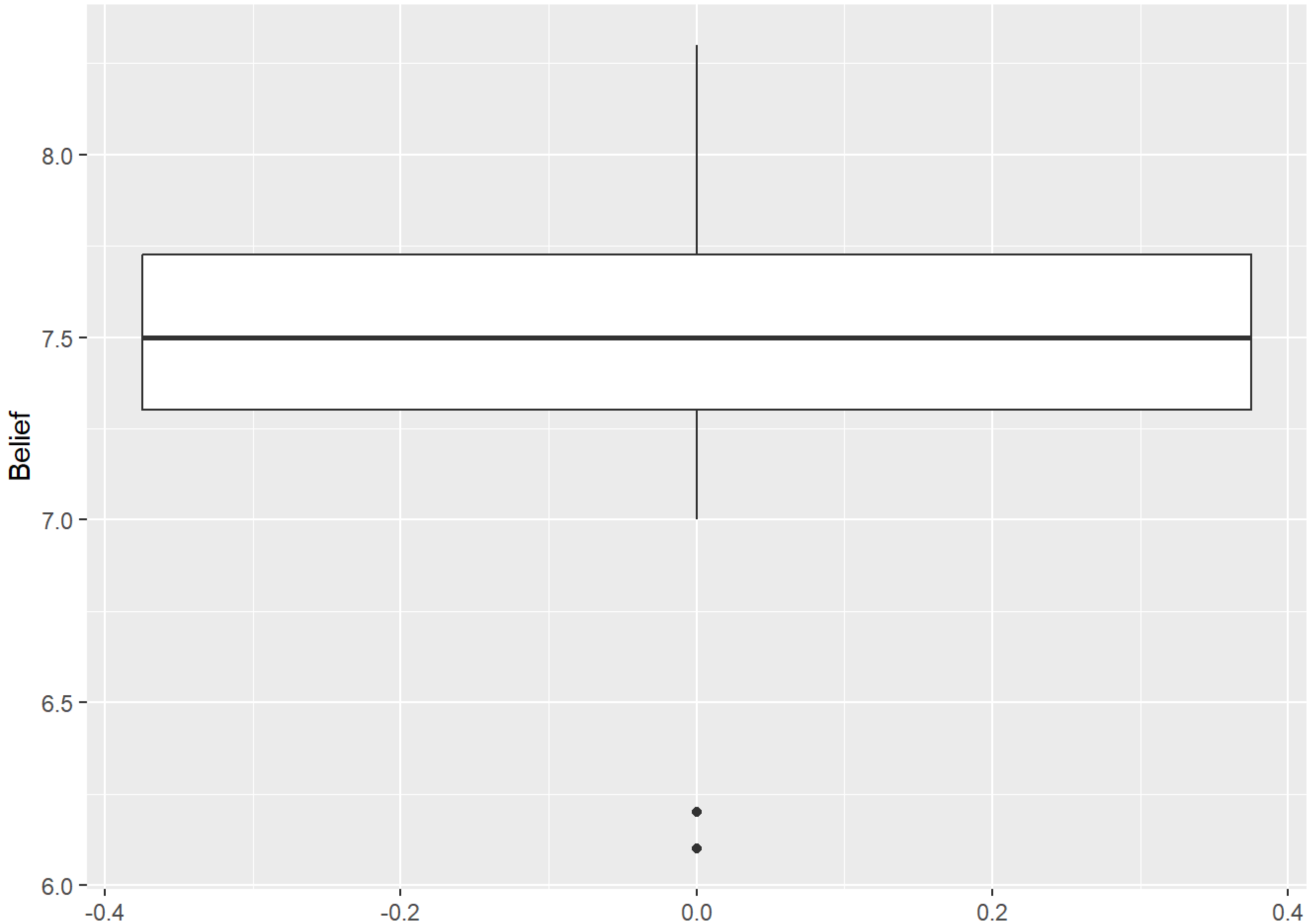

```
ggplot(data = dat,
       mapping = aes(y = Interest)) +
  geom_boxplot() +
    labs(y = "Interest")
```



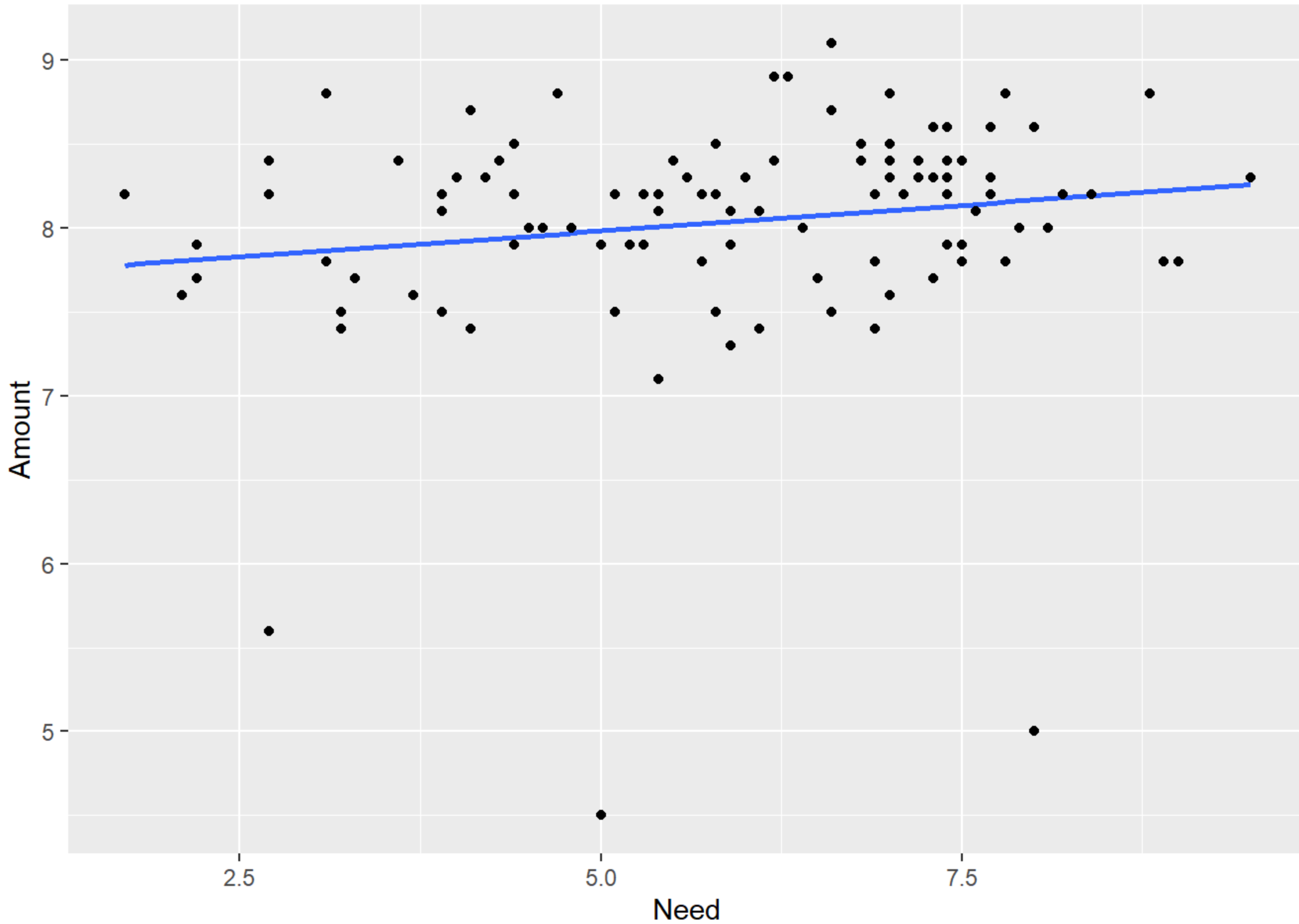

```
ggplot(data = dat,
       mapping = aes(y = Happy)) +
  geom_boxplot() +
    labs(y = "Happy")
```



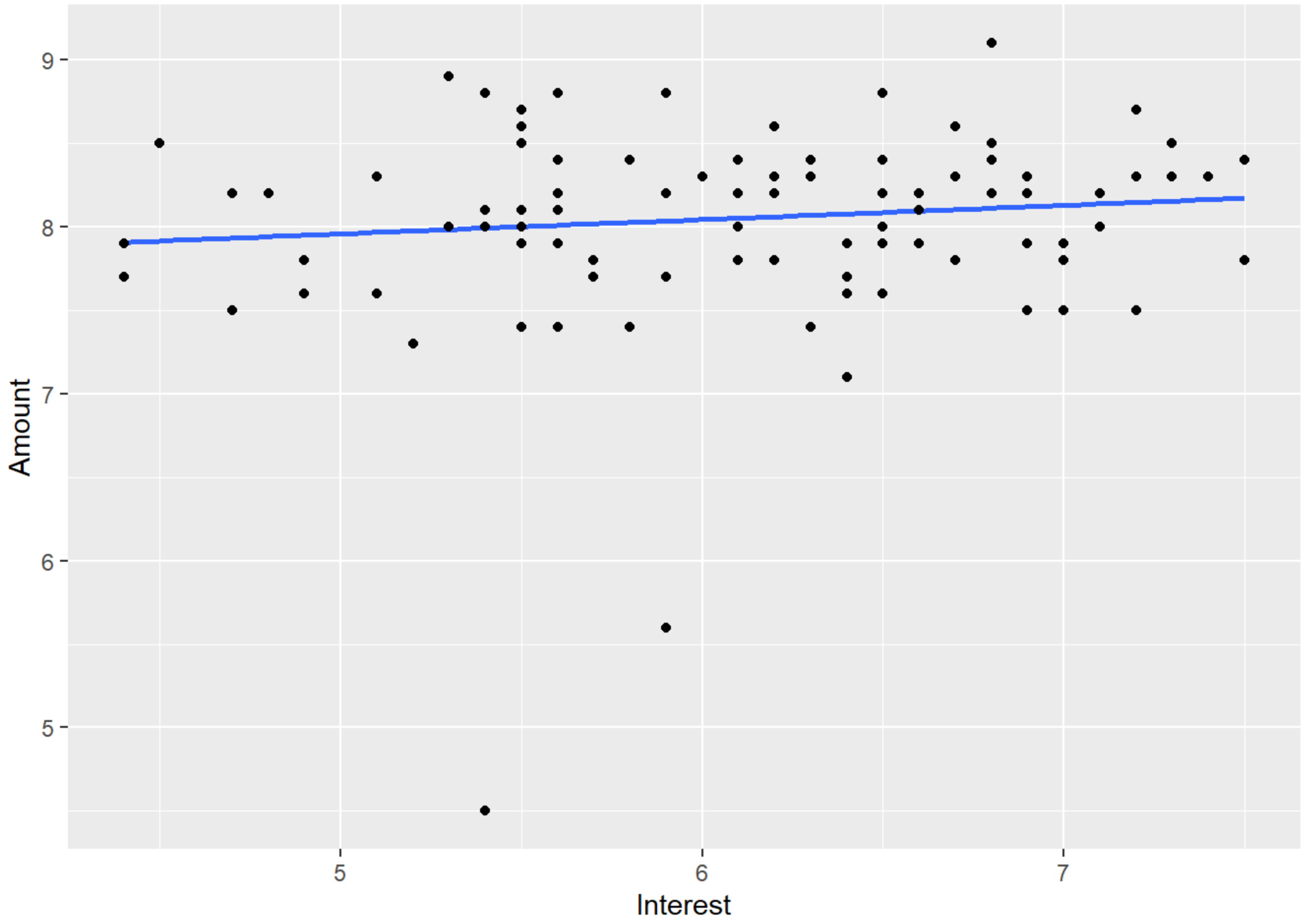

```
ggplot(data = dat,
       mapping = aes(y = Belief)) +
  geom_boxplot() +
    labs(y = "Belief")
```



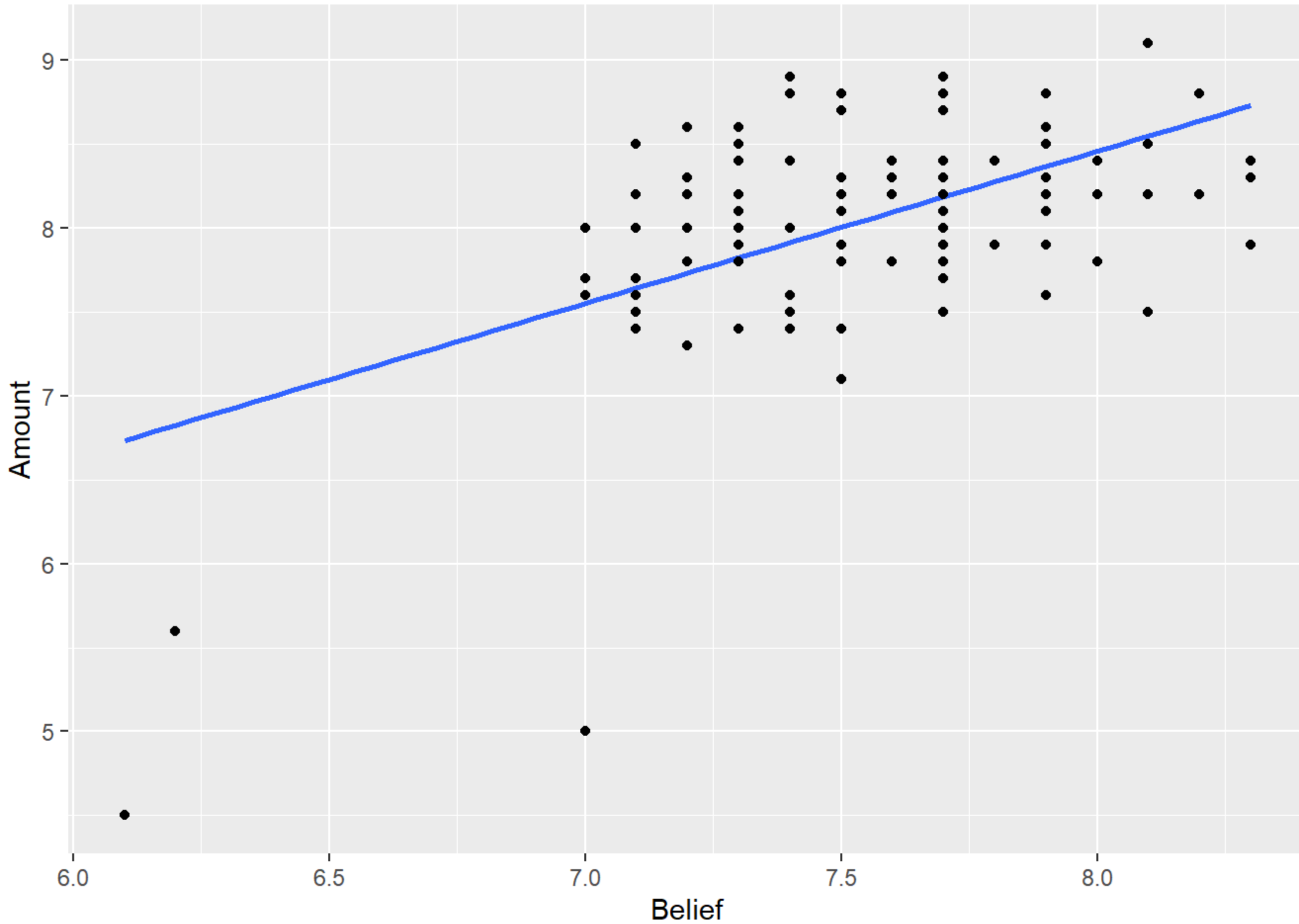

```
# Bivariate Scatterplots of Amount with Predictors
ggplot(data = dat,
       mapping = aes(y = Amount, x = Need)) +
  geom_smooth(method = "lm", formula = y~x, se=FALSE) +
  geom_point() +
  labs(y = "Amount", x = "Need")
```



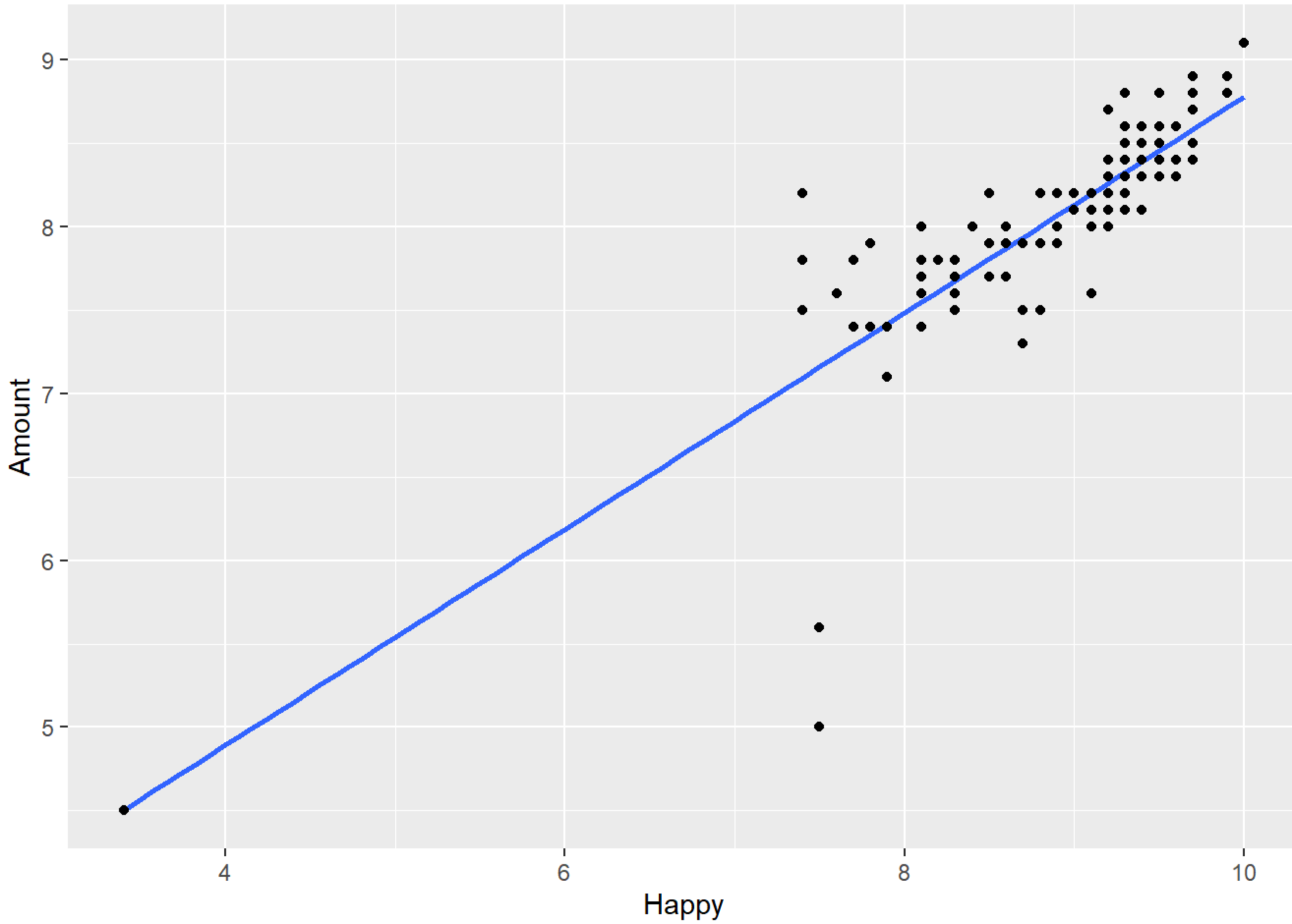

```
ggplot(data = dat,
       mapping = aes(y = Amount, x = Interest)) +
  geom_smooth(method = "lm", formula = y~x, se=FALSE) +
  geom_point() +
  labs(y = "Amount", x = "Interest")
```




```
ggplot(data = dat,
       mapping = aes(y = Amount, x = Belief)) +
  geom_smooth(method = "lm", formula = y~x, se=FALSE) +
  geom_point() +
  labs(y = "Amount", x = "Belief")
```




```
ggplot(data = dat,
       mapping = aes(y = Amount, x = Happy)) +
  geom_smooth(method = "lm", formula = y~x, se=FALSE) +
  geom_point() +
  labs(y = "Amount", x = "Happy")
```



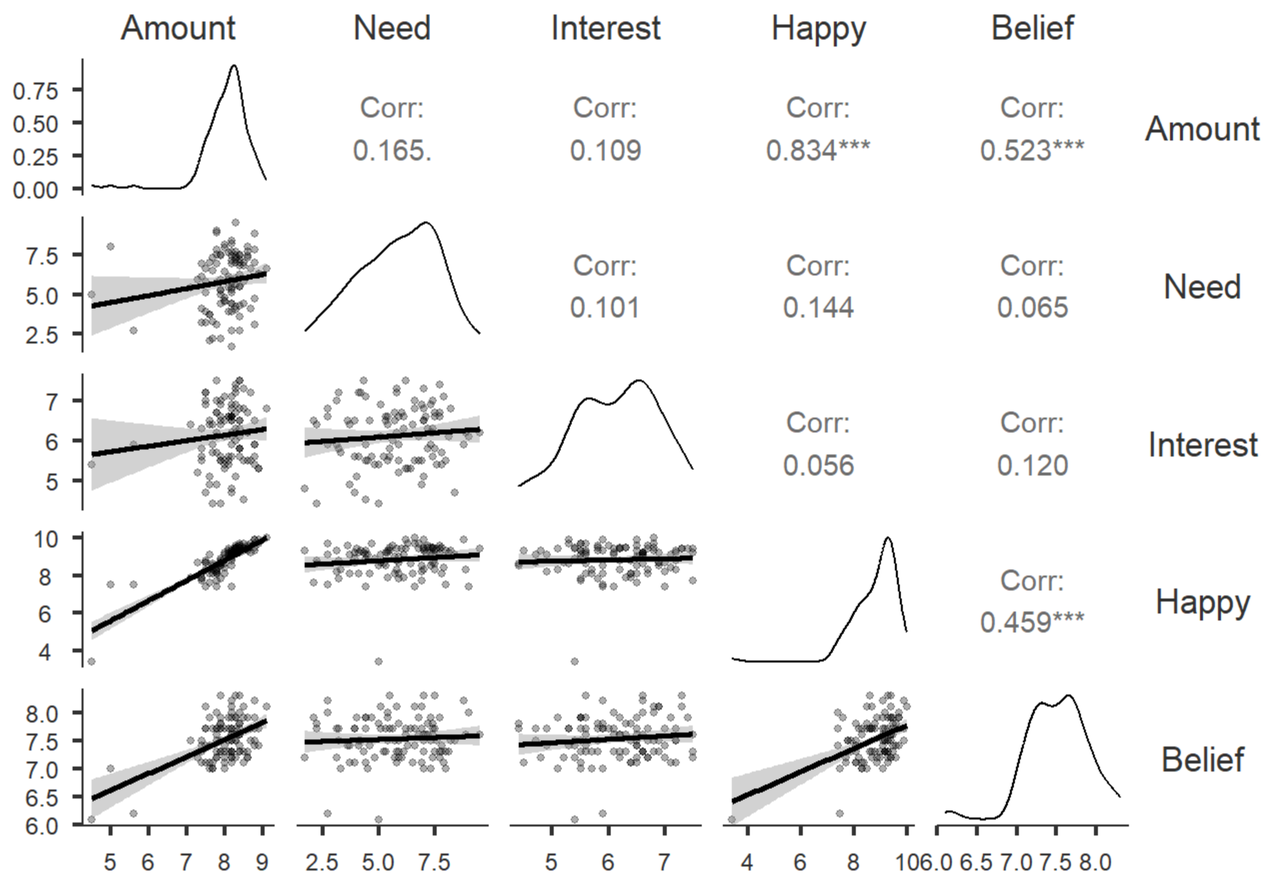

## Generate Correlations Among Variables

```
# `psych` Correlation analyses (Listwise Deletion)
cor_listwise <- corr.test(dat[2:6], use="complete")
print(cor_listwise, short=FALSE)
```

```
## Call:corr.test(x = dat[2:6], use = "complete")
## Correlation matrix
##          Amount Need Interest Happy Belief
## Amount     1.00 0.25     0.12  0.87   0.52
## Need       0.25 1.00     0.10  0.16   0.08
## Interest   0.12 0.10     1.00  0.08   0.15
## Happy      0.87 0.16     0.08  1.00   0.45
## Belief     0.52 0.08     0.15  0.45   1.00
## Sample Size
## [1] 102
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##          Amount Need Interest Happy Belief
## Amount     0.00 0.09     0.89  0.00   0.00
## Need       0.01 0.00     0.95  0.63   0.95
## Interest   0.22 0.32     0.00  0.95   0.68
## Happy      0.00 0.11     0.42  0.00   0.00
## Belief     0.00 0.41     0.14  0.00   0.00
##
##  Confidence intervals based upon normal theory.  To get bootstrapped values, try cor.ci
##             raw.lower raw.r raw.upper raw.p lower.adj upper.adj
## Amont-Need       0.05  0.25      0.42  0.01     -0.02      0.48
## Amont-Intrs     -0.07  0.12      0.31  0.22     -0.13      0.36
## Amont-Happy      0.81  0.87      0.91  0.00      0.78      0.92
## Amont-Belif      0.36  0.52      0.65  0.00      0.29      0.69
## Need-Intrs      -0.10  0.10      0.29  0.32     -0.14      0.33
## Need-Happy      -0.03  0.16      0.34  0.11     -0.10      0.40
## Need-Belif      -0.11  0.08      0.27  0.41     -0.14      0.30
## Intrs-Happy     -0.12  0.08      0.27  0.42     -0.12      0.27
## Intrs-Belif     -0.05  0.15      0.33  0.14     -0.11      0.39
## Happy-Belif      0.28  0.45      0.59  0.00      0.21      0.64
```

```
# `jmv` Correlation analyses with Plots
corrMatrix(dat[2:6],
           ci = TRUE,
           plots = TRUE,
           plotDens = TRUE,
           plotStats = TRUE)
```

```
##
##  CORRELATION MATRIX
##
##  Correlation Matrix
##  ─────────────────────────────────────────────────────────────────────────────────────────────────
##                                Amount        Need          Interest     Happy        Belief
##  ─────────────────────────────────────────────────────────────────────────────────────────────────
##    Amount      Pearson's r              —
##                df                       —
##                p-value                  —
##                95% CI Upper             —
##                95% CI Lower             —
##
##    Need        Pearson's r      0.1645276             —
##                df                     101             —
##                p-value          0.0967671             —
##                95% CI Upper     0.3470037             —
##                95% CI Lower    -0.0299506             —
##
##    Interest    Pearson's r      0.1090415     0.1011541             —
##                df                     101           100             —
##                p-value          0.2728962     0.3117232             —
##                95% CI Upper     0.2963133     0.2899256             —
##                95% CI Lower    -0.0863044    -0.0951934             —
##
##    Happy       Pearson's r      0.8341407     0.1436484     0.0557898            —
##                df                     103           102           102            —
##                p-value         < .0000001     0.1457294     0.5737694            —
##                95% CI Upper     0.8844102     0.3271850     0.2457377            —
##                95% CI Lower     0.7647387    -0.0503323    -0.1382842            —
##
##    Belief      Pearson's r      0.5227888     0.0646370     0.1204032    0.4592751            —
##                df                     101           100           101          102            —
##                p-value         < .0000001     0.5186476     0.2257197    0.0000009            —
##                95% CI Upper     0.6505007     0.2558952     0.3067796    0.5988908            —
##                95% CI Lower     0.3663263    -0.1314908    -0.0748659    0.2925645            —
##  ─────────────────────────────────────────────────────────────────────────────────────────────────
```

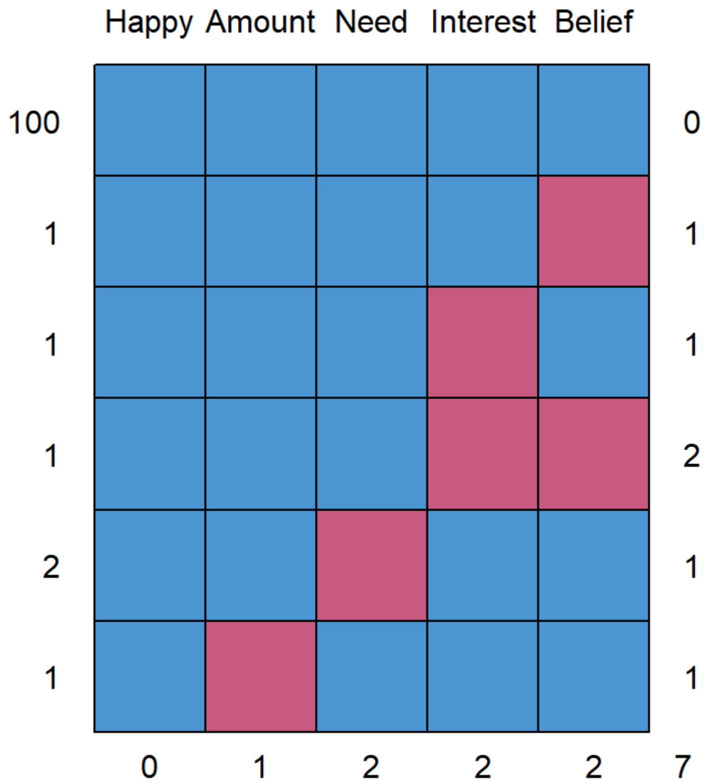



## Assess Missing Data/Pattern

```
# Calculate Amount of Missing Data
sum(is.na(dat[2:6]))/prod(dim(dat[2:6]))
```

```
## [1] 0.01320755
```

```
# Assess Pattern of Missing Data
md.pattern(dat[2:6])
```




```
##     Happy Amount Need Interest Belief
## 100     1      1    1        1      1 0
## 1       1      1    1        1      0 1
## 1       1      1    1        0      1 1
## 1       1      1    1        0      0 2
## 2       1      1    0        1      1 1
## 1       1      0    1        1      1 1
##         0      1    2        2      2 7
```

```
# Due to small amount of missing data we will use listwise deletion and proceed
```

### List-wise deletion of missing data

```
# Listwise deletion results in 7 cases removed from data
# Will create new data object to compare analyses later
dat_no_NA <- na.omit(dat)
```

## Assessment of Univariate Outliers

```
#Identify outliers
dat_no_NA[abs(scale(dat_no_NA$Belief)) > 3.29, ]
```

```
##    ID Amount Need Interest Happy Belief
## 1 100    4.5  5.0      5.4   3.4    6.1
## 6 105    5.6  2.7      5.9   7.5    6.2
```

```
dat_no_NA[abs(scale(dat_no_NA$Need)) > 3.29, ]
```

```
## [1] ID       Amount   Need     Interest Happy    Belief
## <0 rows> (or 0-length row.names)
```

```
dat_no_NA[abs(scale(dat_no_NA$Interest)) > 3.29, ]
```

```
## [1] ID       Amount   Need     Interest Happy    Belief
## <0 rows> (or 0-length row.names)
```

```
dat_no_NA[abs(scale(dat_no_NA$Happy)) > 3.29, ]
```

```
##    ID Amount Need Interest Happy Belief
## 1 100    4.5    5      5.4   3.4    6.1
```

```
dat_no_NA[abs(scale(dat_no_NA$Amount)) > 3.29, ]
```

```
##    ID Amount Need Interest Happy Belief
## 1 100    4.5  5.0      5.4   3.4    6.1
## 6 105    5.6  2.7      5.9   7.5    6.2
```

```
#Belief has 2 univariate outliers (IDs 100 and 105)
#Need has 0
#Interest has 0
#Happy has 1 univariate outlier  (ID 100)
#Amount has 2 univariate outliers  (IDs 100 and 105)
```

### Remove Univariate Outliers

```
# Step needs to be conducted sequentially as shown below (can be expanded with larger datasets)
dat.no.uni1 <- dat_no_NA[!abs(scale(dat_no_NA$Belief)) > 3.29, ]
dat.no.uni2 <- dat.no.uni1[!abs(scale(dat.no.uni1$Need)) > 3.29, ]
dat.no.uni3 <- dat.no.uni2[!abs(scale(dat.no.uni2$Interest)) > 3.29, ]
dat.no.uni4 <- dat.no.uni3[!abs(scale(dat.no.uni3$Happy)) > 3.29, ]
dat_no_NA_UNI <- dat.no.uni4[!abs(scale(dat.no.uni4$Amount)) > 3.29, ]
```

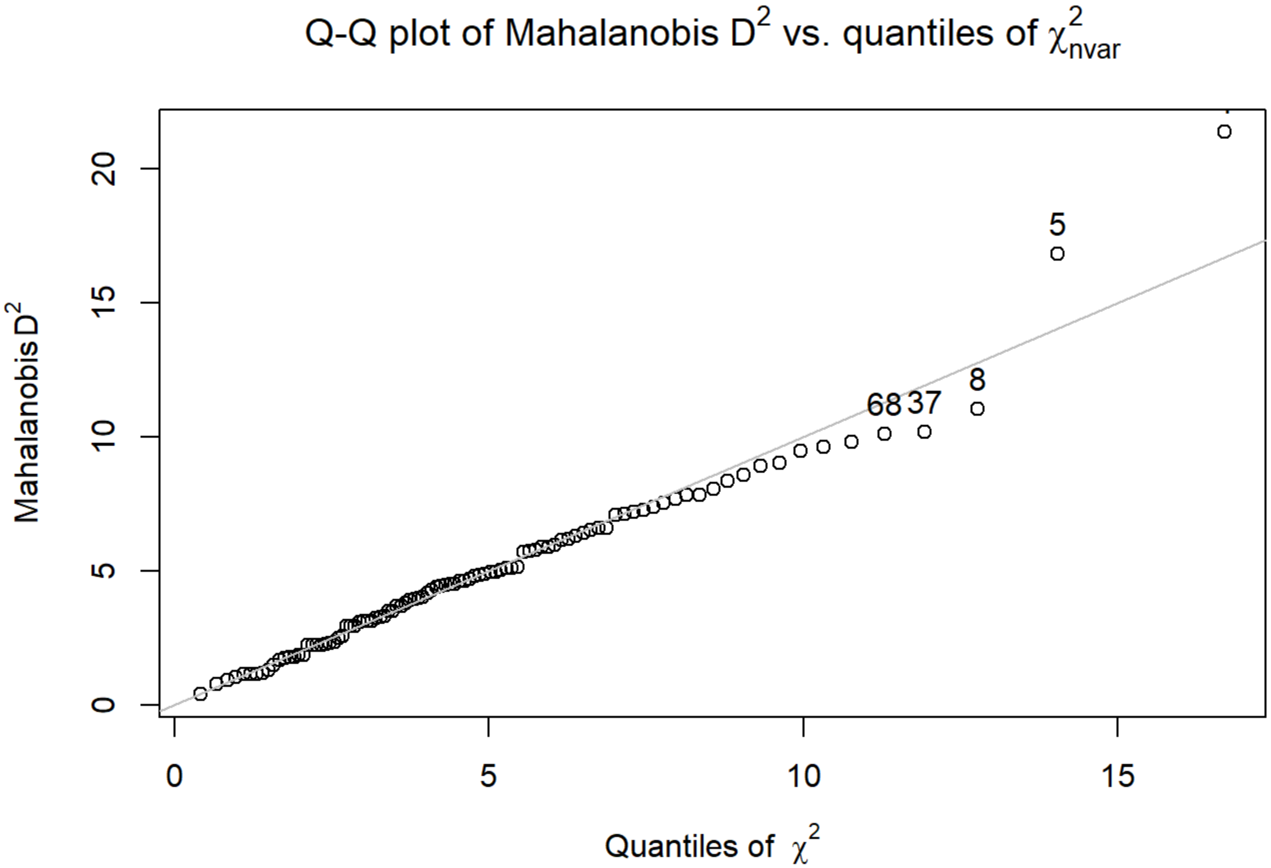
### Reassess Univariate Normality After Univariate Outlier Removal

```
describe(dat_no_NA_UNI[2:6])
```

```
##          vars  n mean   sd median trimmed  mad min  max range  skew kurtosis
## Amount      1 98 8.12 0.41   8.20    8.12 0.44 7.1  9.1   2.0 -0.11    -0.49
## Need        2 98 5.82 1.77   5.90    5.89 2.08 1.7  9.5   7.8 -0.30    -0.71
## Interest    3 98 6.13 0.75   6.20    6.15 0.89 4.4  7.5   3.1 -0.28    -0.67
## Happy       4 98 8.93 0.63   9.10    8.98 0.59 7.4 10.0   2.6 -0.64    -0.49
## Belief      5 98 7.56 0.33   7.55    7.55 0.37 7.0  8.3   1.3  0.28    -0.68
##            se
## Amount   0.04
## Need     0.18
## Interest 0.08
## Happy    0.06
## Belief   0.03
```

### Multivariate Outliers - Mahalanobis and Cook’s Distances

```
# Mahalanobis Distance
dat_no_NA_UNI$mahal <- outlier(dat_no_NA_UNI[2:6])
```




```
dat_no_NA_UNI[abs(scale(dat_no_NA_UNI$mahal)) > 3.29, ]
```

```
##    ID Amount Need Interest Happy Belief    mahal
## 4 103    8.2  4.4      5.9   7.4    7.7 21.35014
## 5 104    7.8  9.0      6.1   7.4    8.0 16.80466
```

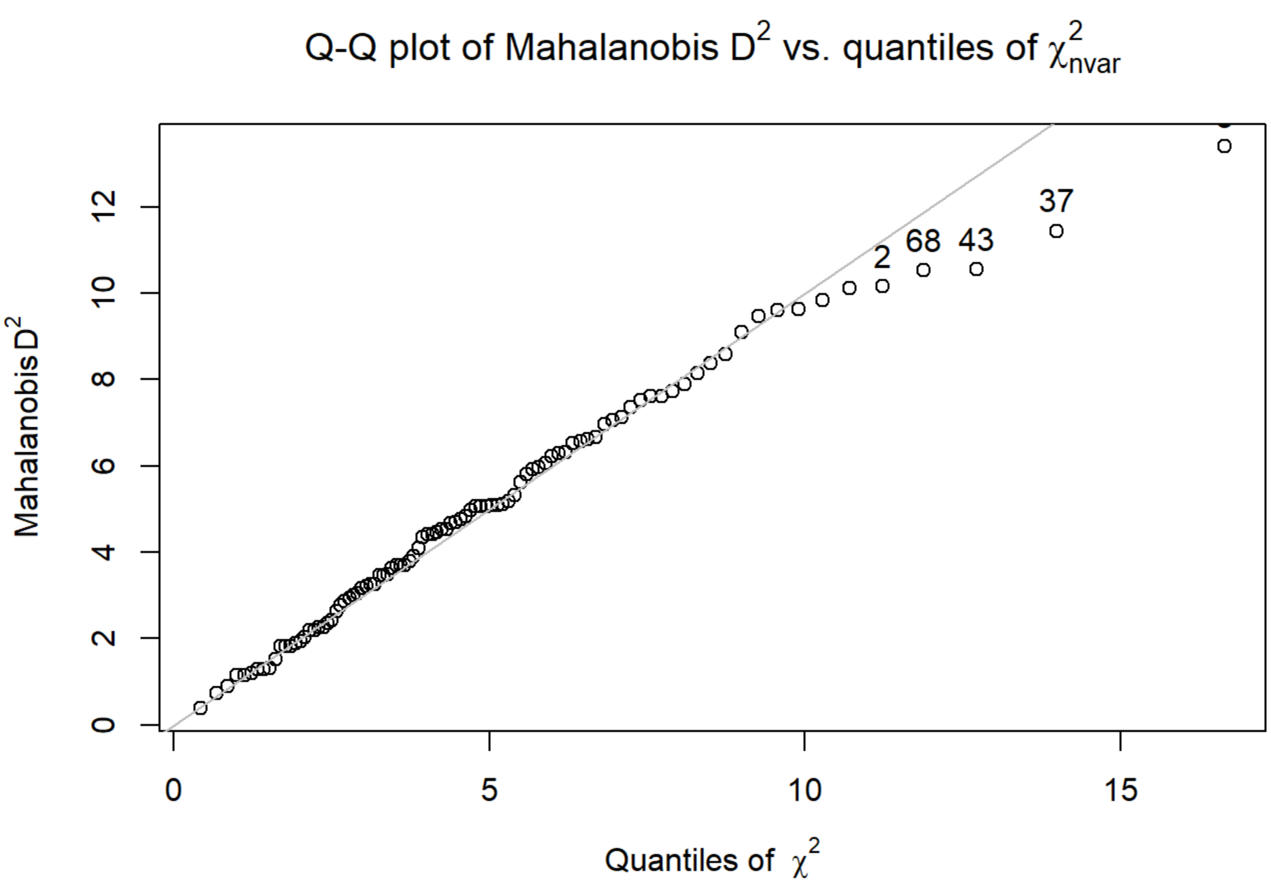
```
## 2 Multivariate Outliers Identified (IDs 103 and 104)

#Cook's Distance
lm<-lm(Amount~Belief + Need + Interest + Happy, data = dat_no_NA_UNI)

#Generate Cook's Distance
dat_no_NA_UNI$cooks <- cooks.distance(lm)

dat_no_NA_UNI[abs(scale(dat_no_NA_UNI$cooks)) > 3.29, ]
```

```
##    ID Amount Need Interest Happy Belief    mahal     cooks
## 4 103    8.2  4.4      5.9   7.4    7.7 21.35014 0.2715843
```

```
## 1 Multivariate Outlier Identified (ID 103)

#Remove multivariate outliers
dat_no_NA_UMO <- dat_no_NA_UNI[!abs(scale(dat_no_NA_UNI$mahal)) > 3.29,]
```

### Multivariate Outlier Iteration Checking Code

```
# Mahalanobis Distance
dat_no_NA_UMO$mahal <- outlier(dat_no_NA_UMO[2:6])
```




```
dat_no_NA_UMO[abs(scale(dat_no_NA_UMO$mahal)) > 3.29, ]
```

```
## [1] ID       Amount   Need     Interest Happy    Belief   mahal    cooks
## <0 rows> (or 0-length row.names)
```

```
# 0 Multivariate Outliers Identified

#Cook's Distance
lm<-lm(Amount~Belief + Need + Interest + Happy, data = dat_no_NA_UMO)

#Generate Cook's Distance
dat_no_NA_UMO$cooks <- cooks.distance(lm)

dat_no_NA_UMO[abs(scale(dat_no_NA_UMO$cooks)) > 3.29, ]
```

```
##     ID Amount Need Interest Happy Belief    mahal      cooks
## 43 142    7.5  3.2      7.2   8.8    7.7 10.56072 0.07604036
```

```
## Multivariate Outliers Identified IDs #1[142], #2[140,150], #3[111,136], #4[107,138],
## #5[101,173], #6[113,131,162], #7[102,205], #8[197], #9[187], #10[166]

#Remove multivariate outliers
dat_no_NA_UMO <- dat_no_NA_UMO[!abs(scale(dat_no_NA_UMO$cooks)) > 3.29,]
```

## Rename Data Object for Additional Assumption Checks

```
dat_final <- dat_no_NA_UMO

# Remove extra datasets
rm(dat.no.uni1,
   dat.no.uni2,
   dat.no.uni3,
   dat.no.uni4,
   dat_no_NA,
   dat_no_NA_UMO,
   dat_no_NA_UNI)
```

## Multivariate Normality

```r
# Mardia's Test of Multivariate Normality
mardia <- mvn(dat_final[2:6],
              mvnTest = "mardia",
              desc = FALSE)

# Henze-Zirkler's Test of Multivariate Normality
hz <- mvn(dat_final[2:6],
          mvnTest = "hz",
          desc = FALSE)

# Energy Test of Multivariate Normality
energy <- mvn(dat_final[2:6],
              mvnTest = "energy",
              desc = FALSE)

# Doornik-Hansen's Test of Multivariate Normality
dh <- mvn(dat_final[2:6],
          mvnTest = "dh",
          desc = FALSE)

mardia$multivariateNormality
```

```
##              Test         Statistic           p value Result
## 1 Mardia Skewness  33.9618014701476 0.518086248634635    YES
## 2 Mardia Kurtosis -1.17166969389339 0.241329695001204    YES
## 3             MVN              <NA>              <NA>    YES
```

```r
hz$multivariateNormality
```

```
##            Test       HZ     p value MVN
## 1 Henze-Zirkler 1.127156 0.000575138  NO
```

```r
energy$multivariateNormality
```

```
##          Test Statistic p value MVN
## 1 E-statistic  1.447145   0.001  NO
```

```r
dh$multivariateNormality
```

```
##             Test        E df      p value MVN
## 1 Doornik-Hansen 53.28651 10 6.587082e-08  NO
```

## Homoscedasticity

```
bplm<-lm(Amount~Belief+Need+Interest+Happy, data = dat_final[2:6])

#Breusch-Pagan test
bptest(bplm, studentize=FALSE)
```

```
##
##  Breusch-Pagan test
##
## data:  bplm
## BP = 5.7009, df = 4, p-value = 0.2226
```

```
bptest(bplm, studentize=TRUE)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  bplm
## BP = 4.0224, df = 4, p-value = 0.403
```

## Multicollinearity

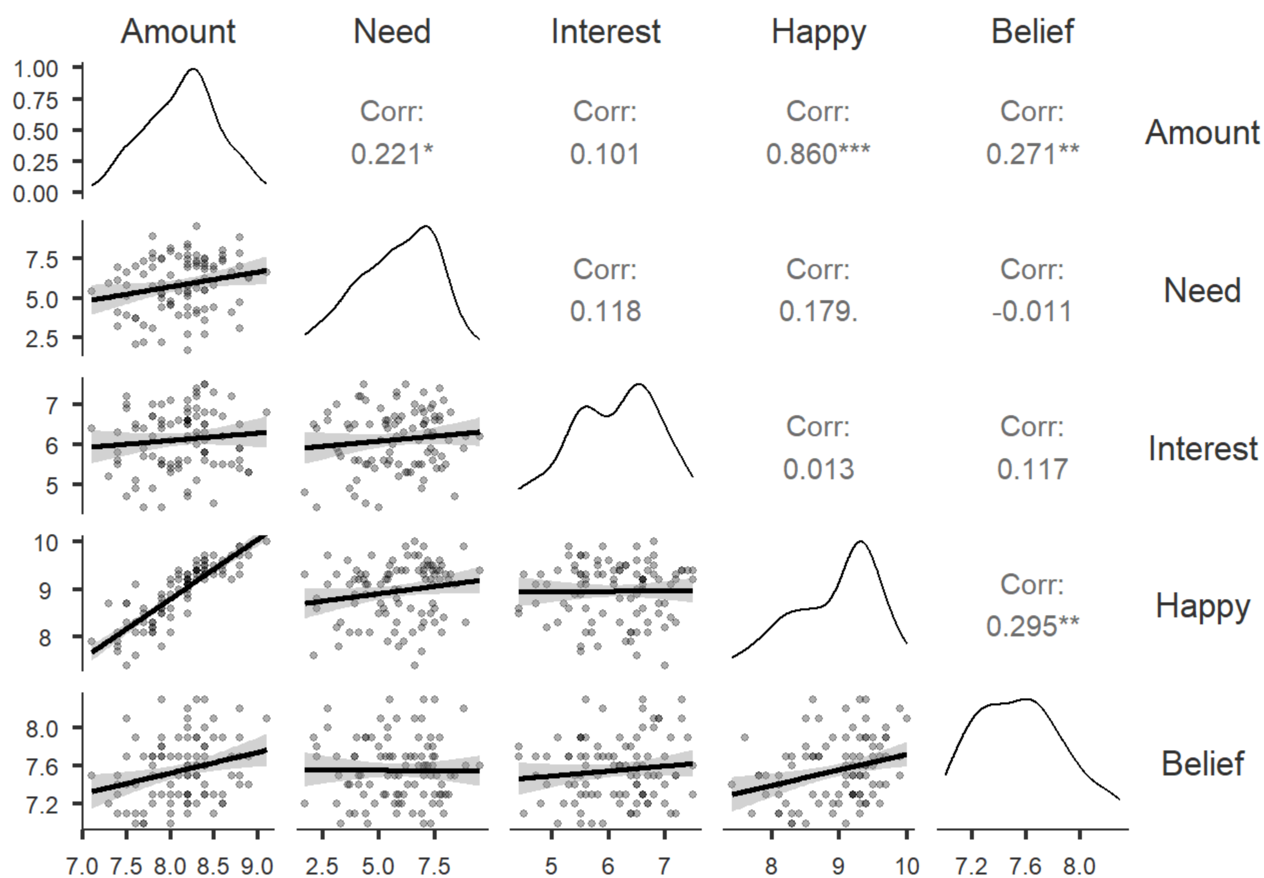
```
#VIF and Tolerance
collin <- linReg(data = dat_final,
                 dep = Amount,
                 cov = c(Belief, Need, Interest, Happy),
                 blocks = list(list('Belief','Need','Interest','Happy')),
                 r=FALSE,
                 r2=FALSE,
                 collin = TRUE)
collin
```

```
##
##  LINEAR REGRESSION
##
##  MODEL SPECIFIC RESULTS
##
##  MODEL 1
##
##  Model Coefficients - Amount
##  ──────────────────────────────────────────────────────────────────────
##    Predictor    Estimate      SE            t             p
##  ──────────────────────────────────────────────────────────────────────
##    Intercept    2.45654948    0.54450316     4.5115431     0.0000194
##    Belief       0.01671975    0.06935396     0.2410785     0.8100427
##    Need         0.01429280    0.01285053     1.1122346     0.2689994
##    Interest     0.04465911    0.02928590     1.5249354     0.1307817
##    Happy        0.57908575    0.03854724    15.0227545    < .0000001
##  ──────────────────────────────────────────────────────────────────────
##
##
##  ASSUMPTION CHECKS
##
##  Collinearity Statistics
##  ─────────────────────────────────────
##                VIF         Tolerance
##  ─────────────────────────────────────
##    Belief      1.118742    0.8938611
##    Need        1.054752    0.9480898
##    Interest    1.031024    0.9699099
##    Happy       1.139251    0.8777698
##  ─────────────────────────────────────
```

```
corrMatrix(dat_final[2:6],
           ci = TRUE)
```

```
## 
##  CORRELATION MATRIX
## 
##  Correlation Matrix
##  ────────────────────────────────────────────────────────────────────────────────────────────────
##                                    Amount        Need          Interest      Happy        Belief    
##  ────────────────────────────────────────────────────────────────────────────────────────────────
##    Amount      Pearson's r                 —                                                        
##                df                          —                                                        
##                p-value                     —                                                        
##                95% CI Upper                —                                                        
##                95% CI Lower                —                                                        
##                                                                                                     
##    Need        Pearson's r         0.2206200             —                                          
##                df                         93             —                                          
##                p-value             0.0316806             —                                          
##                95% CI Upper        0.4041909             —                                          
##                95% CI Lower        0.0199647             —                                          
##                                                                                                     
##    Interest    Pearson's r         0.1010147     0.1179069             —                            
##                df                         93            93             —                            
##                p-value             0.3300415     0.2551309             —                            
##                95% CI Upper        0.2965209     0.3120349             —                            
##                95% CI Lower       -0.1026174    -0.0856720             —                            
##                                                                                                     
##    Happy       Pearson's r         0.8597107     0.1789614     0.0128636            —               
##                df                         93            93            93            —               
##                p-value            < .0000001     0.0826971     0.9015336            —               
##                95% CI Upper        0.9045274     0.3672580     0.2138522            —               
##                95% CI Lower        0.7961083    -0.0234266    -0.1891698            —               
##                                                                                                     
##    Belief      Pearson's r         0.2714395    -0.0106415     0.1172407    0.2951672            —  
##                df                         93            93            93           93            —  
##                p-value             0.0077957     0.9184786     0.2578443    0.0036862            —  
##                95% CI Upper        0.4484493     0.1913118     0.3114251    0.4688206            —  
##                95% CI Lower        0.0739416    -0.2117304    -0.0863425    0.0995460            —  
##  ────────────────────────────────────────────────────────────────────────────────────────────────
```

```
corrMatrix(dat_final[2:6],
           ci = TRUE,
           plots = TRUE,
           plotDens = TRUE,
           plotStats = TRUE)$plot
```




### Multiple Regression Cleaned vs. Uncleaned Data

```
# Multiple regression not clean
linReg(data = dat,
       dep = 'Amount',
       covs = c('Belief', 'Need', 'Interest', 'Happy'),
       blocks = list(list('Belief', 'Need', 'Interest', 'Happy')),
       r2Adj = TRUE,
       modelTest=TRUE,
       ci = TRUE,
       stdEst = TRUE,
       ciStdEst = TRUE)
```

```
##
##  LINEAR REGRESSION
##
##  Model Fit Measures
##  ───────────────────────────────────────────────────────────────────────────────────────────
##    Model    R            R²           Adjusted R²    F           df1    df2    p
##  ───────────────────────────────────────────────────────────────────────────────────────────
##        1    0.8887089    0.7898036      0.7809532    89.23955      4     95    < .0000001
##  ───────────────────────────────────────────────────────────────────────────────────────────
##    Note. Models estimated using sample size of N=100
##
##
##  MODEL SPECIFIC RESULTS
##
##  MODEL 1
##
##  Model Coefficients - Amount
##  ────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
───────────────────────────
##    Predictor    Estimate      SE            Lower           Upper          t             p             Stand. Estimate
Lower          Upper
##  ────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
───────────────────────────
##    Intercept    0.98615474    0.57877510    -0.162859108    2.13516858     1.7038652     0.0916753
##    Belief       0.24442770    0.08326544     0.079124896    0.40973050     2.9355239     0.0041761         0.15592356
0.050474786    0.2613723
##    Need         0.03529352    0.01613011     0.003271213    0.06731583     2.1880527     0.0311183         0.10470208
0.009704411    0.1996998
##    Interest     0.02051693    0.03833814    -0.055593905    0.09662776     0.5351571     0.5937917         0.02555862
-0.069255181    0.1203724
##    Happy        0.55325668    0.03768777     0.478437005    0.62807636    14.6800065    < .0000001         0.78109548
0.675464013    0.8867269
##  ────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
───────────────────────────
```

```
#Multiple regression clean
linReg(data = dat_final,
       dep = 'Amount',
       covs = c('Belief', 'Need', 'Interest', 'Happy'),
       blocks = list(list('Belief', 'Need', 'Interest', 'Happy')),
       r2Adj = TRUE,
       modelTest=TRUE,
       ci = TRUE,
       stdEst = TRUE,
       ciStdEst = TRUE)
```

```
##
##  LINEAR REGRESSION
##
##  Model Fit Measures
##  ───────────────────────────────────────────────────────────────────────────────────────────
##    Model    R            R²           Adjusted R²    F           df1    df2    p
##  ───────────────────────────────────────────────────────────────────────────────────────────
##        1    0.8664207    0.7506848      0.7396041    67.74719      4     90    < .0000001
##  ───────────────────────────────────────────────────────────────────────────────────────────
##    Note. Models estimated using sample size of N=95
##
##
##  MODEL SPECIFIC RESULTS
##
##  MODEL 1
##
##  Model Coefficients - Amount
##  ──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
─────────────────────────
##    Predictor    Estimate      SE            Lower         Upper         t             p             Stand. Estimate    L
ower          Upper
##  ──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
─────────────────────────
##    Intercept    2.45654948    0.54450316     1.37479892    3.53830004     4.5115431     0.0000194
##    Belief       0.01671975    0.06935396    -0.12106399    0.15450348     0.2410785     0.8100427         0.01342074    -
0.09717665    0.1240181
##    Need         0.01429280    0.01285053    -0.01123701    0.03982262     1.1122346     0.2689994         0.06012081    -
0.04726704    0.1675087
##    Interest     0.04465911    0.02928590    -0.01352245    0.10284066     1.5249354     0.1307817         0.08149649    -
0.02467653    0.1876695
##    Happy        0.57908575    0.03854724     0.50250492    0.65566657    15.0227545    < .0000001         0.84394165
0.73233513    0.9555482
##  ──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
─────────────────────────
```

## Model Pruning and Model Comparisons/Parsimonious Models

```r
#Simple Bivariate Regression (Amount ~ Belief)
lm1 <- lm(Amount~Belief, data = dat_final)

#Simple Standardized Bivariate Regression (Amount ~ Belief)
zlm1 <- lm(scale(Amount) ~ scale(Belief), data = dat_final)

#Call for standardized regression coefficients/summary
summary(zlm1)
```

```
##
## Call:
## lm(formula = scale(Amount) ~ scale(Belief), data = dat_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4444 -0.6748  0.1282  0.5759  1.9865
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.875e-16  9.928e-02    0.00   1.0000
## scale(Belief)  2.714e-01  9.980e-02    2.72   0.0078 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9676 on 93 degrees of freedom
## Multiple R-squared:  0.07368,    Adjusted R-squared:  0.06372
## F-statistic: 7.397 on 1 and 93 DF,  p-value: 0.007796
```

```r
#Call for standardized regression confidence intervals
confint(zlm1)
```

```
##                     2.5 %    97.5 %
## (Intercept)   -0.19714112 0.1971411
## scale(Belief)  0.07325257 0.4696265
```

```r
#Multiple Linear Regression
lm2 <- lm(Amount~Belief+Need, data = dat_final)

#Standardized Multiple Regression
zlm2 <- lm(scale(Amount) ~ scale(Belief) + scale(Need), data = dat_final)

#Call for standardized MR coefficients/summary
summary(zlm2)
```

```
##
## Call:
## lm(formula = scale(Amount) ~ scale(Belief) + scale(Need), data = dat_final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.38828 -0.65393  0.03925  0.61153  1.92773
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.334e-16  9.708e-02   0.000  1.00000
## scale(Belief)  2.738e-01  9.760e-02   2.805  0.00613 **
## scale(Need)    2.235e-01  9.760e-02   2.290  0.02430 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9463 on 92 degrees of freedom
## Multiple R-squared:  0.1236, Adjusted R-squared:  0.1046
## F-statistic:  6.49 on 2 and 92 DF,  p-value: 0.002309
```

```r
summary(lm2)
```

```
##
## Call:
## lm(formula = Amount ~ Belief + Need, data = dat_final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.98845 -0.27064  0.01625  0.25310  0.79784
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.24303    0.93098   5.632 1.93e-07 ***
## Belief       0.34113    0.12160   2.805  0.00613 **
## Need         0.05314    0.02320   2.290  0.02430 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3916 on 92 degrees of freedom
## Multiple R-squared:  0.1236, Adjusted R-squared:  0.1046
## F-statistic:  6.49 on 2 and 92 DF,  p-value: 0.002309
```

```
#Call for standardized MR confidence intervals
confint(zlm2)
```

```
##                    2.5 %    97.5 %
## (Intercept)   -0.19281789 0.1928179
## scale(Belief)  0.07996649 0.4676701
## scale(Need)    0.02968202 0.4173856
```

```
confint(lm2)
```

```
##                   2.5 %     97.5 %
## (Intercept) 3.394036695 7.09203114
## Belief      0.099623346 0.58262973
## Need        0.007056446 0.09922702
```

#### Direct Model Comparison

```
#Model Comparison
anova(lm1, lm2)
```

```
## Analysis of Variance Table
##
## Model 1: Amount ~ Belief
## Model 2: Amount ~ Belief + Need
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1     93 14.915
## 2     92 14.111  1   0.80446 5.245 0.0243 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(zlm1, zlm2)
```

```
## Analysis of Variance Table
##
## Model 1: scale(Amount) ~ scale(Belief)
## Model 2: scale(Amount) ~ scale(Belief) + scale(Need)
##   Res.Df    RSS Df Sum of Sq     F Pr(>F)
## 1     93 87.074
## 2     92 82.378  1    4.6964 5.245 0.0243 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

## Run and Visualize Final Model

```
linReg(data = dat_final,
       dep = 'Amount',
       covs = c('Belief', 'Need'),
       blocks = list(list('Belief', 'Need')),
       r2Adj = TRUE,
       modelTest=TRUE,
       ci = TRUE,
       stdEst = TRUE,
       ciStdEst = TRUE)
```

```
##
##  LINEAR REGRESSION
##
##  Model Fit Measures
##  ─────────────────────────────────────────────────────────────────────────────────────────
##    Model    R            R²           Adjusted R²    F           df1    df2    p
##  ─────────────────────────────────────────────────────────────────────────────────────────
##        1    0.3516264    0.1236411      0.1045898    6.489912      2     92    0.0023087
##  ─────────────────────────────────────────────────────────────────────────────────────────
##    Note. Models estimated using sample size of N=95
##
##
##  MODEL SPECIFIC RESULTS
##
##  MODEL 1
##
##  Model Coefficients - Amount
##  ────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
─────────────────────
##    Predictor    Estimate      SE            Lower          Upper         t            p            Stand. Estimate    Lowe
r       Upper
##  ────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
─────────────────────
##    Intercept    5.24303392    0.93097526    3.394036695    7.09203114    5.631765    0.0000002
##    Belief       0.34112654    0.12159753    0.099623346    0.58262973    2.805374    0.0061327          0.2738183    0.07
996649    0.4676701
##    Need         0.05314174    0.02320407    0.007056446    0.09922702    2.290190    0.0242958          0.2235338    0.02
968202    0.4173856
##  ────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────
─────────────────────
```

```
set.seed(13)
apa.reg.boot.table(lm2,
                   number.samples = 1000)
```

```
##
##
## apa.reg.boot.table is a beta version.
## Block 1: Generating 1000 bootstrap samples
```

```
## Warning: The `x` argument of `as_tibble.matrix()` must have unique column names if
## `.name_repair` is omitted as of tibble 2.0.0.
## ℹ Using compatibility `.name_repair`.
## ℹ The deprecated feature was likely used in the apaTables package.
##   Please report the issue at <https://github.com/dstanley4/apaTables/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

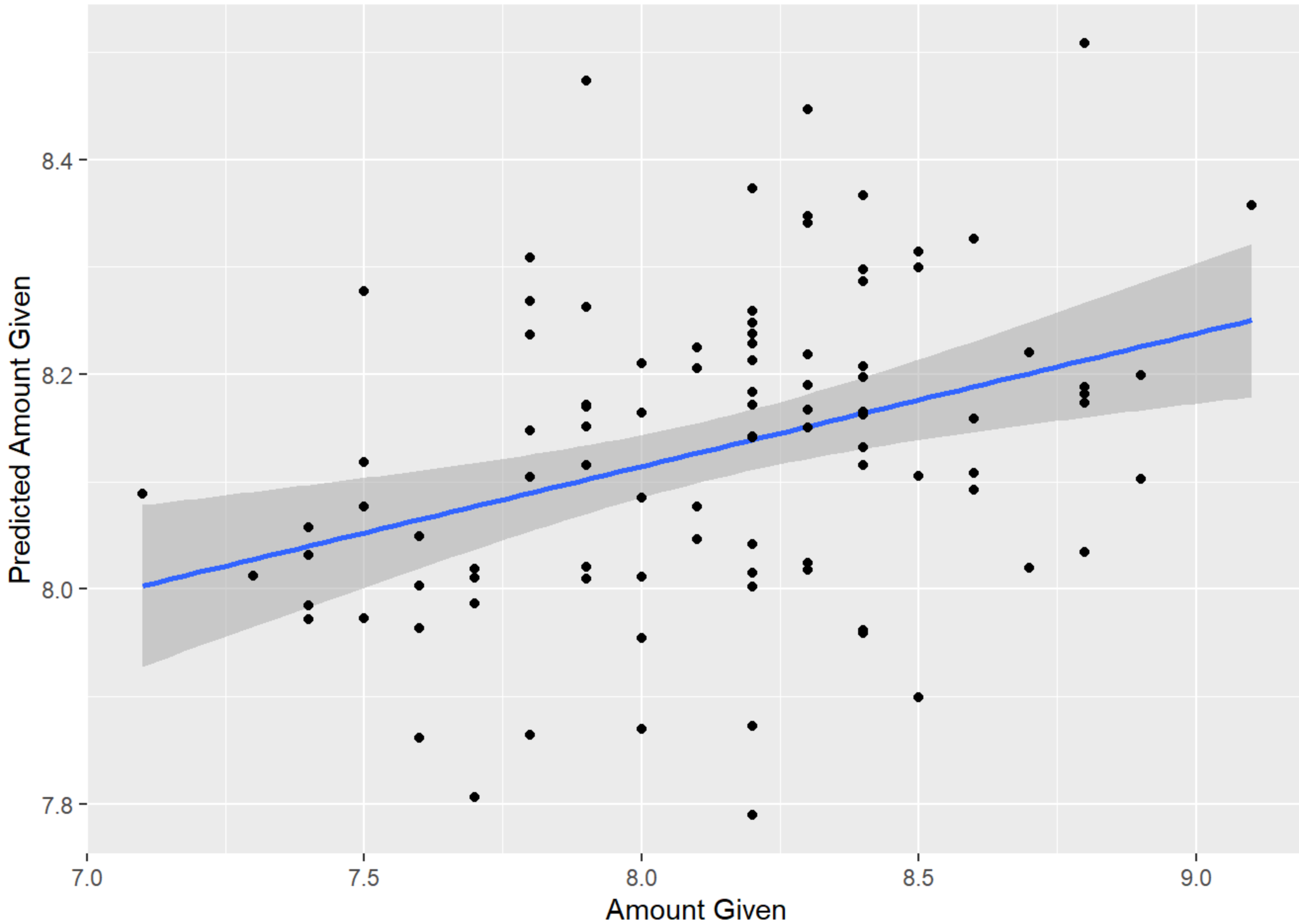
```
## Bootstrap for Delta RSQ in progress
```

```
##
##
## Regression results using Amount as the criterion
##
##
##    Predictor      b     b_95%_CI beta  beta_95%_CI sr2 sr2_95%_CI     r
##  (Intercept) 5.24** [3.44, 6.97]
##       Belief 0.34** [0.12, 0.57] 0.27 [0.10, 0.44] .07 [.01, .19] .27**
##         Need  0.05* [0.01, 0.09] 0.22 [0.04, 0.38] .05 [.00, .15]  .22*
##
##
##
##             Fit
##
##
##
##      R2 = .124**
##  95% CI[.04,.26]
##
##
## Note. A significant b-weight indicates the beta-weight and semi-partial correlation are also significant.
## b represents unstandardized regression weights. beta indicates the standardized regression weights.
## sr2 represents the semi-partial correlation squared. r represents the zero-order correlation.
## Square brackets are used to enclose the lower and upper limits of a confidence interval.
## * indicates p < .05. ** indicates p < .01.
##
```

```
apa.cor.table(dat_final[2:6])
```

```
##
##
## Means, standard deviations, and correlations with confidence intervals
##
##
##   Variable    M    SD   1           2           3           4
##   1. Amount   8.13 0.41
##
##   2. Need     5.83 1.74 .22*
##                         [.02, .40]
##
##   3. Interest 6.12 0.76 .10         .12
##                         [-.10, .30] [-.09, .31]
##
##   4. Happy    8.96 0.60 .86**       .18         .01
##                         [.80, .90]  [-.02, .37] [-.19, .21]
##
##   5. Belief   7.56 0.33 .27**       -.01        .12         .30**
##                         [.07, .45]  [-.21, .19] [-.09, .31] [.10, .47]
##
##
## Note. M and SD are used to represent mean and standard deviation, respectively.
## Values in square brackets indicate the 95% confidence interval.
## The confidence interval is a plausible range of population correlations
## that could have caused the sample correlation (Cumming, 2014).
##  * indicates p < .05. ** indicates p < .01.
##
```

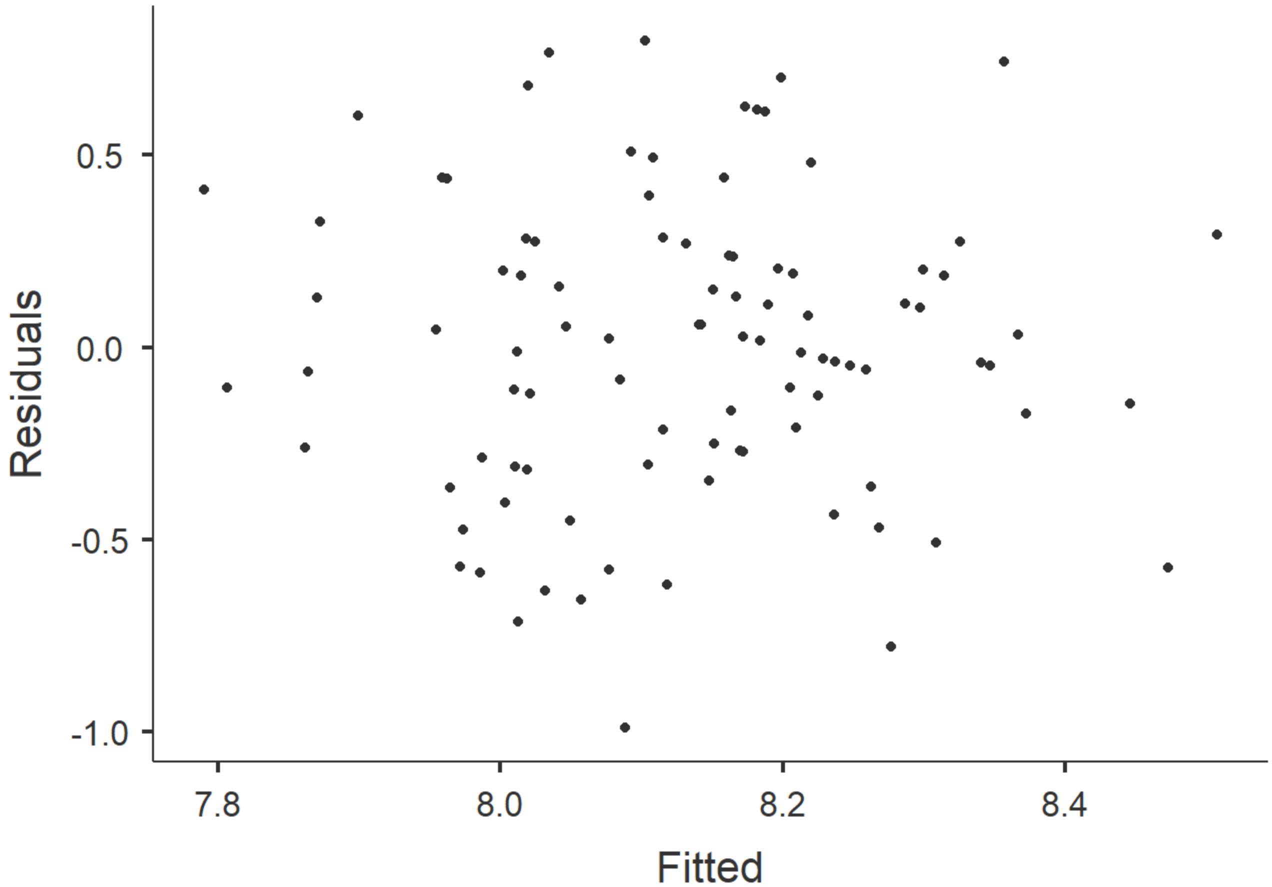
```
# create predicted values from three predictors and save in object
dat_final$predictedF <- fitted(lm2)
dat_final$residuals <- resid(lm2)

# plot predicted line
ggplot(data = dat_final,
       mapping = aes(x = Amount, y = predictedF)) +
  geom_smooth(method = "lm", formula = y ~ x) +
  geom_point() +
  labs(x = "Amount Given", y = "Predicted Amount Given")
```



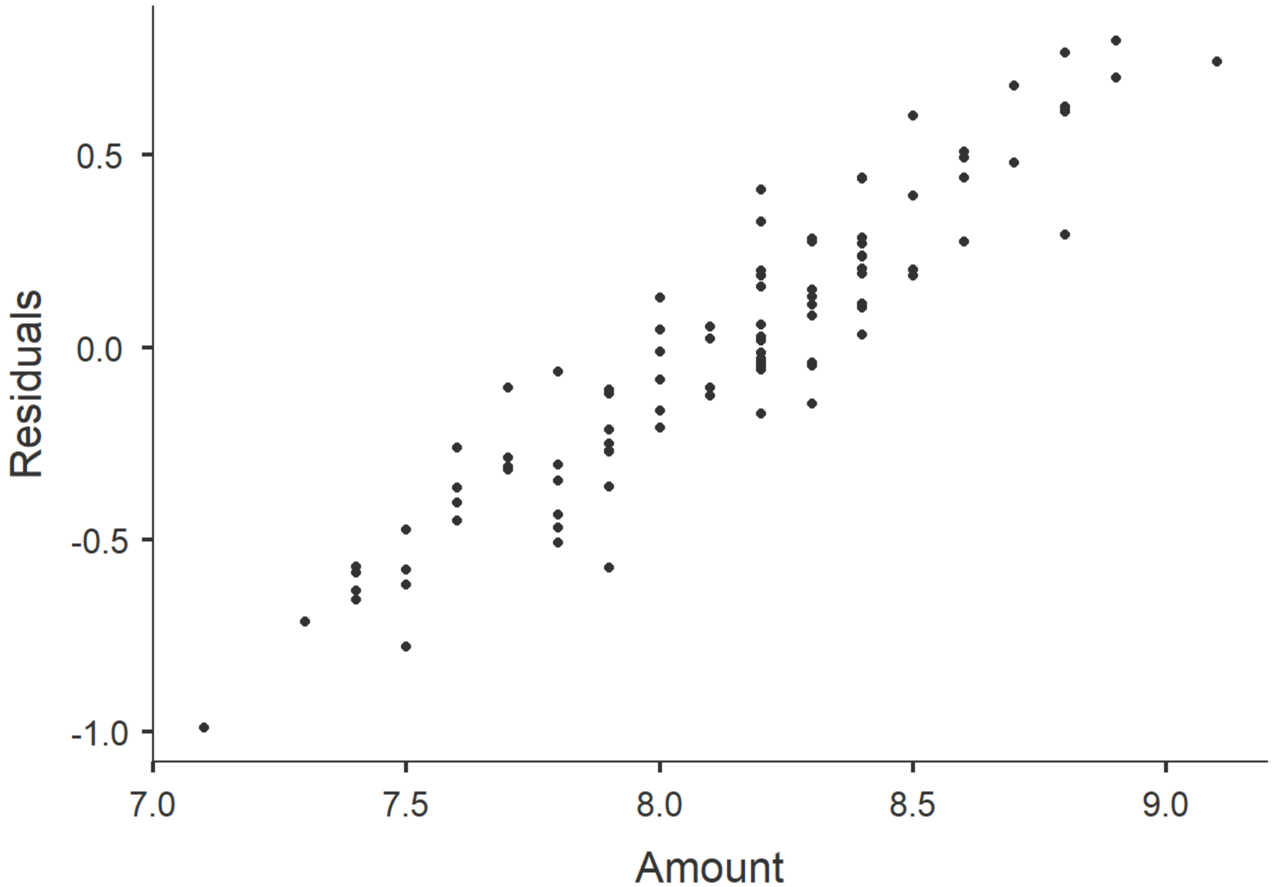

### Checking residual plots

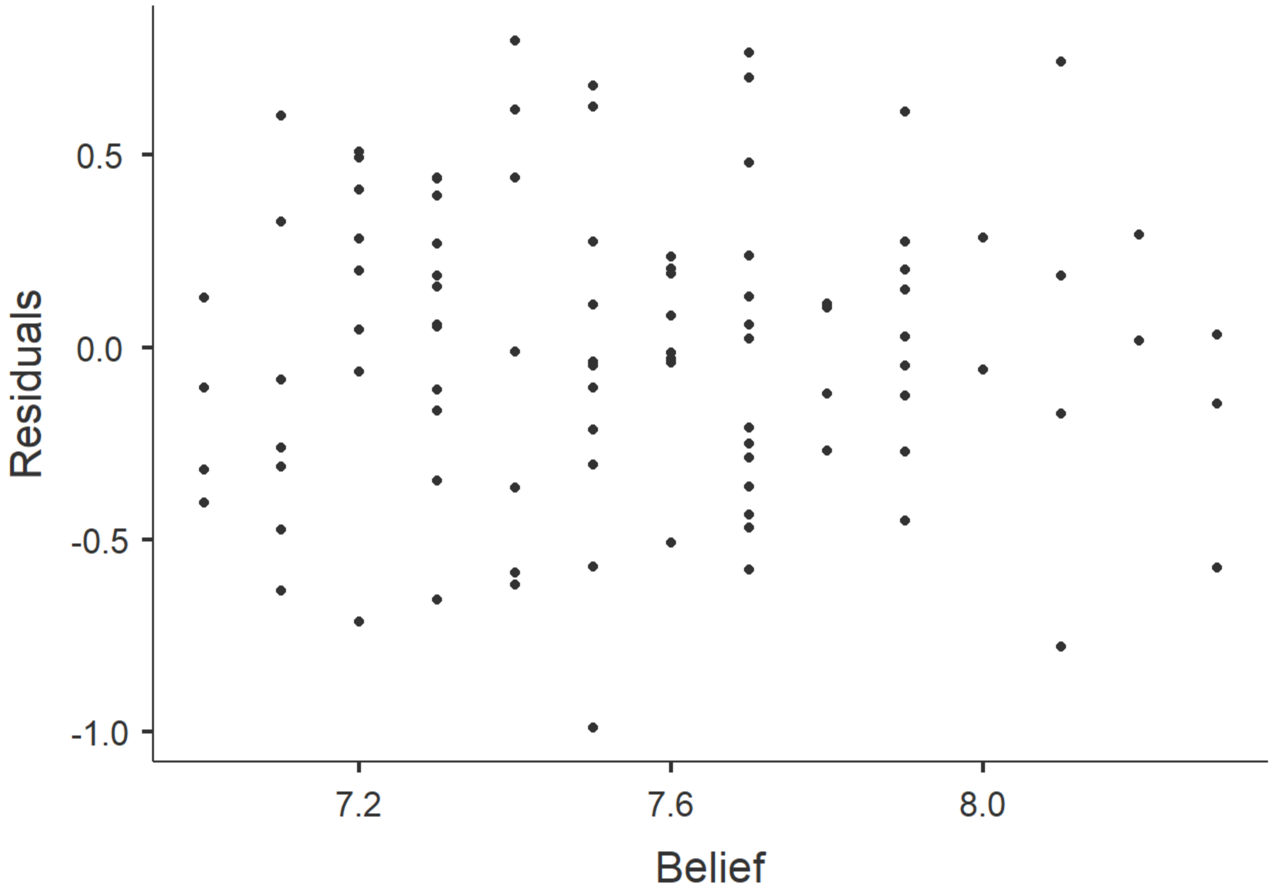
```
linReg(data = dat_final,
       dep = Amount,
       covs = vars(Belief,Need),
       blocks = list(list('Belief','Need')),
       modelTest = FALSE,
       r=FALSE,
       r2=FALSE,
       resPlots=TRUE)
```

```
##
##  LINEAR REGRESSION
##
##  MODEL SPECIFIC RESULTS
##
##  MODEL 1
##
##  Model Coefficients - Amount
##  ───────────────────────────────────────────────────────────────────
##    Predictor    Estimate      SE            t           p
##  ───────────────────────────────────────────────────────────────────
##    Intercept    5.24303392    0.93097526    5.631765    0.0000002
##    Belief       0.34112654    0.12159753    2.805374    0.0061327
##    Need         0.05314174    0.02320407    2.290190    0.0242958
##  ───────────────────────────────────────────────────────────────────
##
##
##  ASSUMPTION CHECKS
```

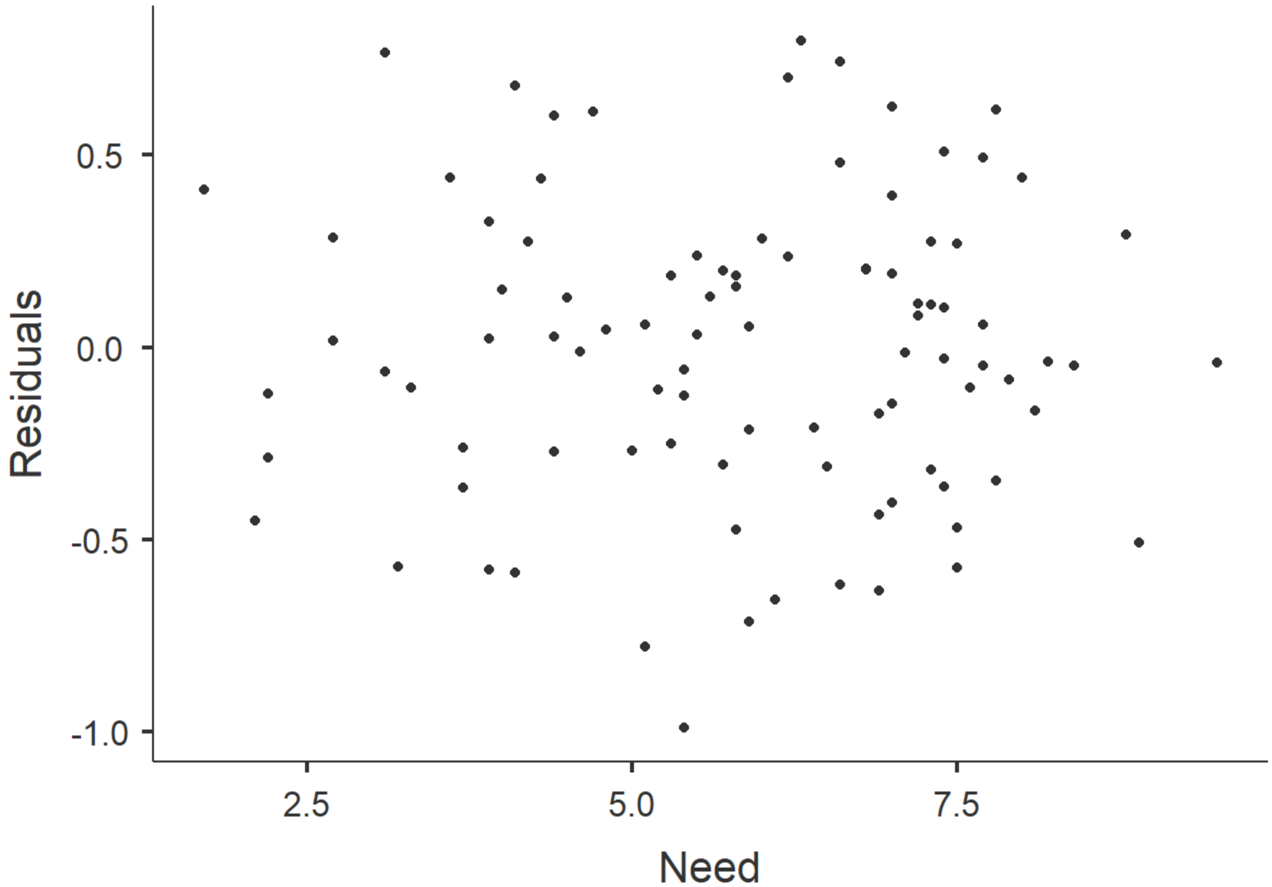


Residuals
0.5
0.0
-0.5
-1.0
7.8
8.0
8.2
8.4
Fitted

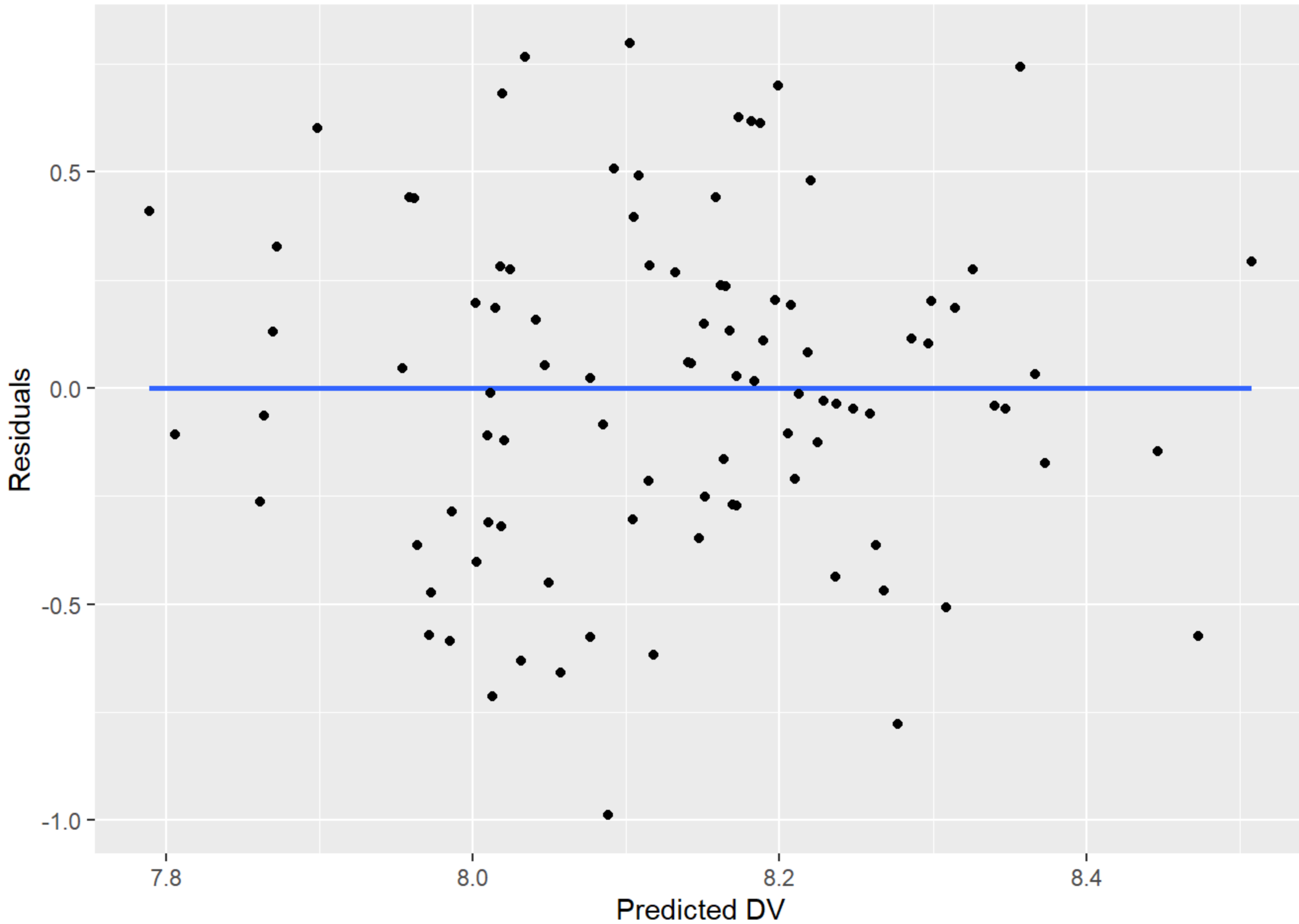

0.5
0.0
-0.5
-1.0
Residuals
7.0
7.5
8.0
8.5
9.0
Amount



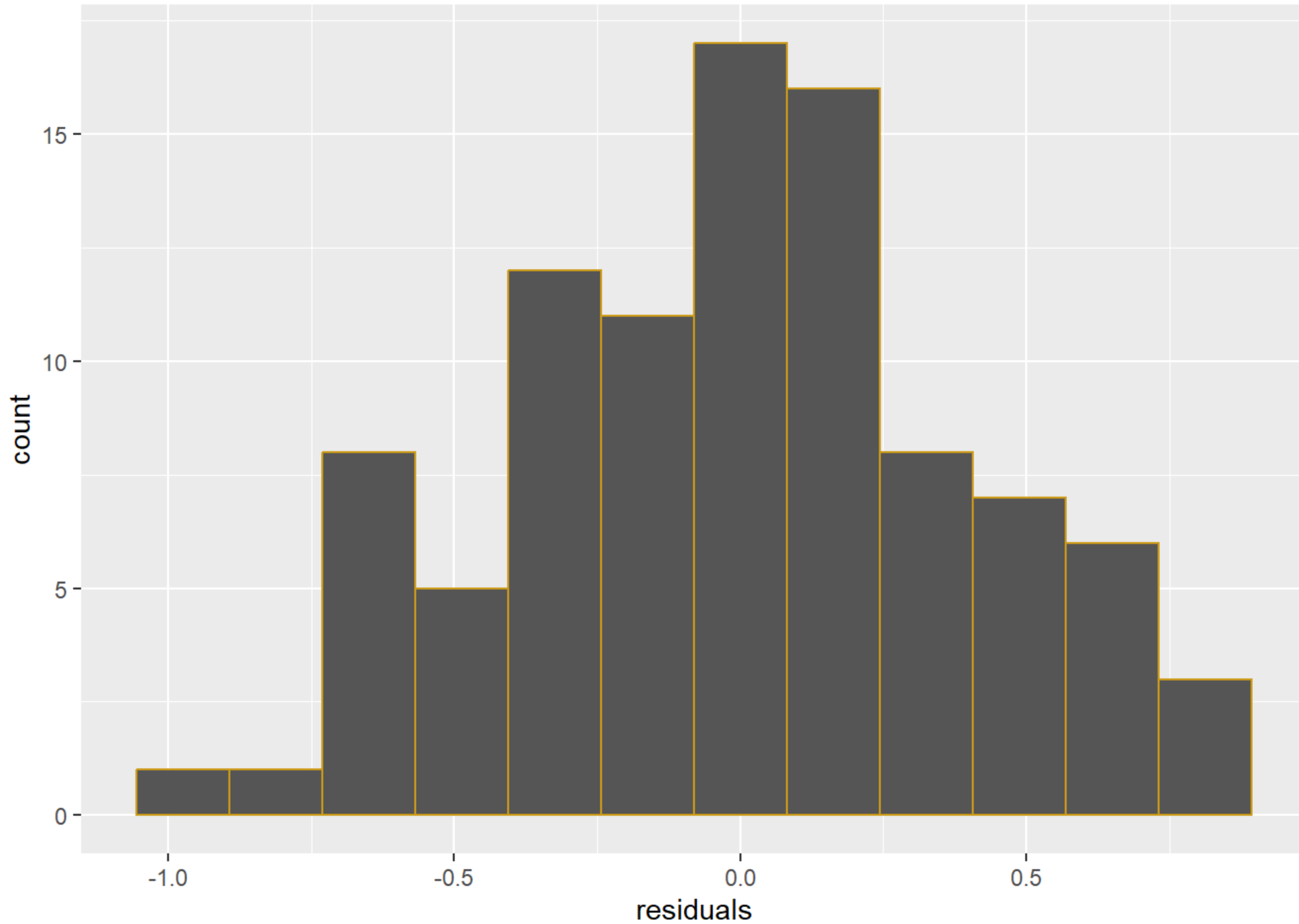
Residuals
0.5
0.0
-0.5
-1.0
7.2
7.6
8.0
Belief




```
ggplot(data = dat_final,
       mapping = aes(x = predictedF, y = residuals)) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
  geom_point() +
  labs(x = "Predicted DV", y = "Residuals")
```




```
ggplot(data = dat_final,
       mapping = aes(x = residuals)) +
  geom_histogram(bins = 12, color = "goldenrod3")
```




```
describe(dat_final$residuals)
```

```
##    vars  n mean   sd median trimmed  mad   min max range  skew kurtosis   se
## X1    1 95    0 0.39   0.02       0 0.39 -0.99 0.8  1.79 -0.06    -0.49 0.04
```