Predictors of Intelligence for a College Entrance Exam Preparation Course

Daniel Pinedo

Psych 308c: Assignment 3

Below is my feedback on your manuscript. Please come see a TA if you'd like to review your manuscript f

Predictors of Intelligence for a College Entrance Exam Preparation Course

Standardized tests remain one important predictor for success in college. Sylvan Learning is a test preparation service that wants to find the biggest predictors of intelligence in order to evaluate their learning course to prepare students to take the ACT. A completed literature review indicated that the most important known predictors for intelligence are working memory, processing speed, and vocabulary. Sylvan learning center provided cross-sectional data testing these potential predictors of intelligence. The purpose of this study was to determine how working memory, processing speed, and vocabulary predict intelligence for our sample, with the model including all three predictors hypothesized to best predict intelligence.

Method

The present study used a correlational design. Data collection methods included archival data of intelligence, working memory, processing speed, vocabulary, and demographics.

Participants

Participant observations were 148 youth ages 14 to 18 in the archival dataset. Demographics included sex (female, *n* = 67; male *n* = 82), race (Latinx, *n* = 72; White, *n* = 67; no response, *n* = 9) and GPA (range was 2.26 to 3.86).

Measures

Each set of participant observations was assessed using the below measures. All measures were scored on a scale of 0 to 10.

Intelligence. Intelligence was assessed using Raven's Progressive Matrices.

Working Memory. Working memory was assessed using Letter Number Sequencing.

Processing Speed. Processing speed was assessed using Letter Comparison.

Vocabulary. Vocabulary was assessed using Peabody Picture Vocabulary Test.

Planned Analysis

The present study planned to use correlation, simple regression, and multiple regression to assess the relationships between predictors, as well as predictors and the outcome variable.

Results

Data analysis is in Appendix A. Observations (N = 148) were removed that had missing parameters (12 total, N = *136)* in the dataset. Analysis continued with tests of assumptions and inspection of histograms. Three univariate outliers were determined and removed that were greater than 3 *SD* from z-score mean (N = 133). Six multivariate outliers were determined and removed based on calculations of Cook's distance (N = 127), a measure of multivariate influence. Descriptive statistics are in Table 1. Data was verified to be normally distributed across all variables in the model as evidenced by skew for all variables being within a threshold of ± 3.00 (Table 1), and kurtosis being within a threshold of ± 10.00 (Table 1). The homoscedasticity assumption was confirmed using Breusch-Pagan test of non-constant variance, χ^2 (1) = 1.47, p = .225. The assumption of linearity appears to be met for working memory (r = .43, p < .001), processing speed (r = .19, p = .038), and vocabulary (r = .16, p = .075) when correlated with intelligence, as evidenced by viewing scatterplots with regression lines added.

Working memory and processing speed were significantly correlated with intelligence, while vocabulary was not (Table 2). The relationship between the outcome (intelligence) and potential predictors was further assessed through regression analyses. The best model fit for simple regression was indicated for working memory (β = .43, p < .001) which explained 19% of the variance in intelligence, F(1, 125) = 28.50, p < .001, R^2 = .19 (Table 3, Model 1). Adding processing speed (β = .06, p = .511) and vocabulary (β = .14, p = .092) to Model 1 did not account for additional significant variance, F(2, 123) = 1.70 , ΔR^2 = .02, p = .187 (Table 3, Model 2). Therefore, Model 1 is determined to be the best fit, indicating that working memory is the best predictor for intelligence.

Discussion

The purpose of the current project was to determine the best predictors for intelligence in order to create an optimal standardized test study program for Sylvan Learning. Correlation and regression analyses were used to test the hypothesis that working memory, processing speed, and vocabulary tests together predicted intelligence scores. The literature suggested that all three variables would predict intelligence, however our study indicated that working memory was tested to be the only significant predictor (Table 3).

Although working memory did predict a significant amount of variance in intelligence scores, processing speed and vocabulary both did not. This may be problematic because the tests Sylvan Learning is preparing high school students to take include both processing speed and vocabulary, such as the ACT or SAT. Furthermore, the entrance exams may not be measuring the same operational definition of intelligence that Sylvan Learning is testing. It could also be the case that other constructs may also predict additional unique variance for this specific intelligence test, such as stress levels, hours of sleep the night before the test, reading comprehension, and written communication. It is the recommendation of this analysis that Sylvan Learning test not only predictors for test scores of intelligence using cross-sectional data, but after implementing their program to include a pre- and post-test experimental study that assesses if the program accounted for an increase in overall test scores.

Table 1
Descriptive Statistics of Measures

Variable	Mean	SD	Median	Skew	Kurtosis
Intelligence	6.13	1.17	6.10	0.16	-0.63
Working Memory	8.35	1.05	8.40	-0.45	-0.40
Processing Speed	6.23	1.11	6.19	-0.10	-0.38
Vocabulary	7.95	1.20	7.92	-0.76	0.09

Table 2
Correlation Matrix for Measures Related to Intelligence

Variable	1	2	3	4
1. Intelligence	-	.43**	.19*	.16
2. Working Memory		-	.30***	.05
3. Processing Speed			-	.05
4. Vocabulary				-

Note. * $p < .05$, ** $p < .01$, *** $p < .001$.

Table 3
Hierarchical Regression Models Predicting Intelligence

Model	Variables	*B*	β	SE	R^2
Model 1	Working Memory	0.48	.43***	0.76	.19
Model 2	Processing Speed	0.06	.06	0.09	.21
	Vocabulary	0.13	.14	0.08	

Note. * $p < .05$, ** $p < .01$, *** $p < .001$.

Appendix A

Statistical Analysis in R

Daniel Pinedo

March 5, 2019

Prompt

You are hired by Sylvan Learning Center to investigate what best **predicts intelligence**. They want to incorporate this information into their ACT prep classes. The company hires you to complete a comprehensive literature review, reserach proposal, and expect a polished report back to them at their end-of-year meeting.

According to the **literature review**, a number of variables were related to intelligence. Among these variables included: **working memory**, **processing speed**, and **vocabulary**, as important predictors of intelligence. This being the case, you are given access to their database of collected information regarding student performance and a variety other measures. Please investigate and report back to Sylvan regarding the **most appropriate explantory model predicting intelligence for their sample of students**.

Measures: [all variables are on a scale of 0 to 10 unless otherwise noted]

intell: measure of intelligence (Raven's Progessive Matrices)
wm: measure of working memory (Letter Number Sequencing)
process: measure of processing speed (Letter Comparison)
vocab: measure of vocabulary (Peabody Picture Vocabulary Test)

Demographics:
Age: in years (open-text input).
Sex: self-reported.
Race: self-reported (NR = not reported).

Hypothesis:

H0: working memory, processing speed, vocabulary, and intelligence are not related

Ha: working memory, processing speed, and vocabulary predict intelligence

N = 148

Use the data in the file to investigate the relationships among these four measures and to predict intelligence from working memory, processing speed, and vocabulary. *Additionally, please be sure to incorporate learned procedures and data analysis techniques as appropriate.*

Initial Data Diagnosis

```
# Descriptives to get an overall view of data
desc <- descriptives(data = dat,
                     vars = c('intell', 'wm', 'process', 'vocab', 'age', 'Sex', 'Race'),
                     sd = TRUE,
                     range = TRUE,
                     skew = TRUE,
                     kurt = TRUE,
                     freq = TRUE) # for categorical variables
desc

##
##  DESCRIPTIVES
##
##  Descriptives
##  ---------------------------------------------------------------------------------------
##                          intell    wm      process    vocab     age      Sex    Race
##  ---------------------------------------------------------------------------------------
##    N                        144     144       146       146       148    148     148
##    Missing                    4       4         2         2         0      0       0
##    Mean                    5.97    8.32      6.23      7.87      16.4
##    Median                  6.10    8.41      6.19      7.92      16.4
##    Standard deviation      1.52    1.23      1.17      1.26     0.824
##    Range                   8.80    9.00      6.19      6.08      3.90
##    Minimum                0.800    1.00      3.33      3.50      14.5
##    Maximum                 9.60    10.0      9.52      9.58      18.4
##    Skewness              -0.673   -1.73    0.0935    -0.821    0.0562
##    Std. error skewness    0.202   0.202     0.201     0.201     0.199
##    Kurtosis                1.55    7.57   -0.0995     0.412    -0.475
##    Std. error kurtosis    0.401   0.401     0.399     0.399     0.396
##  ---------------------------------------------------------------------------------------
##
##
##  FREQUENCIES
```

```
##
##  Frequencies of Sex
##  -------------------------------------------------
##    Levels    Counts    % of Total    Cumulative %
##  -------------------------------------------------
##    Female        67          45.3            45.3
##    Male          81          54.7           100.0
##  -------------------------------------------------
##
##
##  Frequencies of Race
##  -------------------------------------------------
##    Levels    Counts    % of Total    Cumulative %
##  -------------------------------------------------
##    Latinx        72          48.6            48.6
##    NR             9           6.1            54.7
##    White         67          45.3           100.0
##  -------------------------------------------------
```

```
corr.test(dat[2:5]) # Prerequisite: outcome and predictor variables are measured on the continuous level
```

```
## Call:corr.test(x = dat[2:5])
## Correlation matrix
##         intell   wm process vocab
## intell    1.00 0.32    0.20  0.20
## wm        0.32 1.00    0.29  0.18
## process   0.20 0.29    1.00  0.09
## vocab     0.20 0.18    0.09  1.00
## Sample Size
##         intell  wm process vocab
## intell     144 140     142   142
## wm         140 144     142   142
## process    142 142     146   144
```

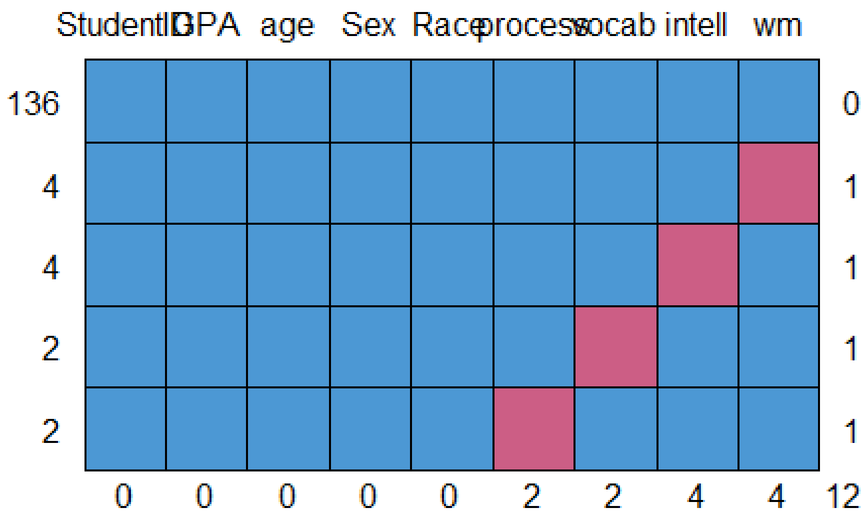
```
## vocab      142 142     144   146
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##         intell   wm process vocab
## intell    0.00 0.00    0.06  0.06
## wm        0.00 0.00    0.00  0.07
## process   0.02 0.00    0.00  0.28
## vocab     0.02 0.03    0.28  0.00
##
##  To see confidence intervals of the correlations, print with the short=FALSE option

#MISSING DATA --> Different N's and the line that indicates missing items indicates missing cases
#Running dim(dat) indicates 148 rows/observations
#Options: (1) delete list-wise (2) impute
```

Regression Diagnostics 1. Missing Data 2. Univariate a. Normality, b. Linearity and c. Outliers 3. Multivariate a. Normality and b.Outliers 4. Heteroscedsticity 5. Multi-collinearity 6. Linearity between outcome and predictor(s)

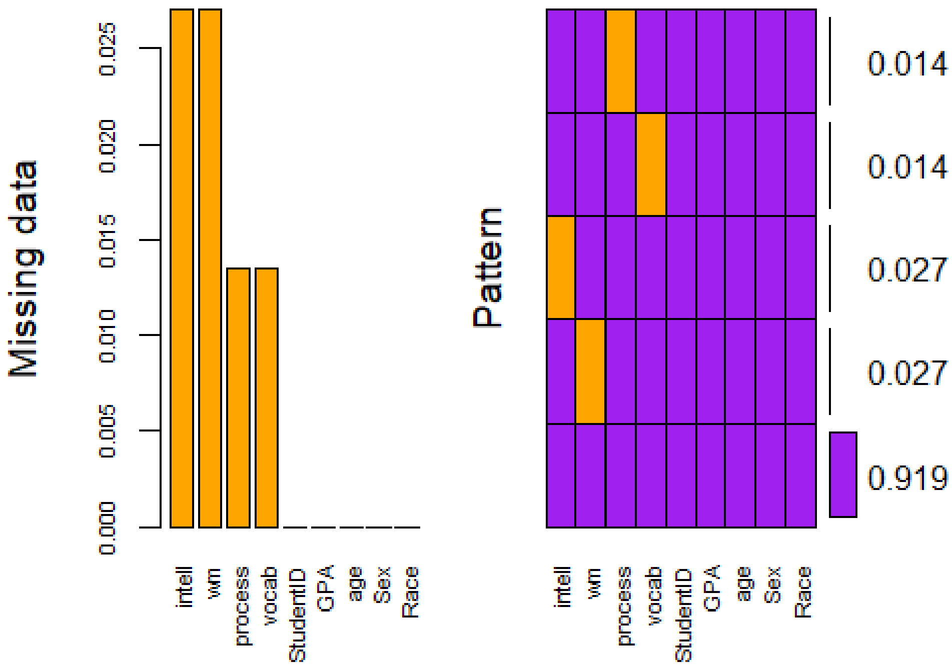
1. Missing Data

```
#check the pattern of missing data
dat[rowSums(is.na(dat)) > 0,]

##     StudentID intell    wm process vocab  GPA  age    Sex   Race
## 1           1    7.2 9.352   5.238    NA 3.17 17.0   Male  White
## 8           8     NA 8.908   8.095 5.417 2.55 16.6   Male  White
## 24         24     NA 8.089   5.714 6.667 3.01 18.2 Female Latinx
## 27         27    1.2 7.210      NA 6.667 3.68 16.4 Female  White
## 29         29    4.4    NA   6.905 6.667 3.37 17.9 Female  White
## 52         52    6.1    NA   6.190 7.500 2.96 16.9   Male Latinx
## 68         68     NA 8.352   6.190 7.917 3.17 16.3   Male  White
## 85         85    5.0    NA   5.714 8.333 2.76 17.4 Female Latinx
## 89         89    8.3 8.089   6.667    NA 2.96 15.3   Male  White
## 105       105     NA 7.216   6.429 8.750 3.18 16.8   Male     NR
## 113       113    3.8 7.387      NA 9.167 2.56 15.4   Male  White
## 132       132    6.7    NA   6.429 9.167 3.36 16.8   Male  White
```

```
md.pattern(dat)
```




```
##     StudentID GPA age Sex Race process vocab intell wm
## 136         1   1   1   1    1       1     1      1  1  0
## 4           1   1   1   1    1       1     1      1  0  1
## 4           1   1   1   1    1       1     1      0  1  1
## 2           1   1   1   1    1       1     0      1  1  1
## 2           1   1   1   1    1       0     1      1  1  1
##             0   0   0   0    0       2     2      4  4 12

mice_plot <-aggr(dat,
              col=c('purple', 'orange'),
              numbers = TRUE,
              sortVars = TRUE,
              labels = names(dat),
              cex.axis = .7,
              gap = 3,
              ylab = c("Missing data", "Pattern"))
```




```
##
## Variables sorted by number of missings:
##   Variable      Count
##     intell 0.02702703
##         wm 0.02702703
##    process 0.01351351
##      vocab 0.01351351
##  StudentID 0.00000000
##        GPA 0.00000000
##        age 0.00000000
##        Sex 0.00000000
##       Race 0.00000000
#orange bar chart is percentage missing from each variable --> no greater than 2.5% here
#purple and orange(missing) chart shows pattern of missing data --> no pattern here

# Option 1: Listwise deletion of missing data. New dataset is named "dat.no.NA"
dat.no.NA <- na.omit(dat)
```

```
#check descriptives again

desc_listwise <- descriptives(data = dat.no.NA,
                    vars = c('intell', 'wm', 'process', 'vocab', 'age', 'Sex', 'Race'),
                    sd = TRUE,
                    range = TRUE,
                    skew = TRUE,
                    kurt = TRUE,
                    freq = TRUE) # for categorical variables
desc_listwise

##
##  DESCRIPTIVES
##
##  Descriptives
##  ---------------------------------------------------------------------------------------
##                          intell    wm      process    vocab     age      Sex    Race
##  ---------------------------------------------------------------------------------------
##    N                        136      136       136       136       136    136     136
##    Missing                    0        0         0         0         0      0       0
##    Mean                    6.00     8.33      6.23      7.88      16.3
##    Median                  6.10     8.44      6.19      7.92      16.4
##    Standard deviation      1.47     1.25      1.19      1.27     0.815
##    Range                   8.80     9.00      6.19      6.08      3.90
##    Minimum                0.800     1.00      3.33      3.50      14.5
##    Maximum                 9.60     10.0      9.52      9.58      18.4
##    Skewness              -0.608    -1.76    0.0927    -0.864    0.0608
##    Std. error skewness    0.208    0.208     0.208     0.208     0.208
##    Kurtosis                1.62     7.53    -0.181     0.541    -0.472
##    Std. error kurtosis    0.413    0.413     0.413     0.413     0.413
##  ---------------------------------------------------------------------------------------
##
##
##  FREQUENCIES
```

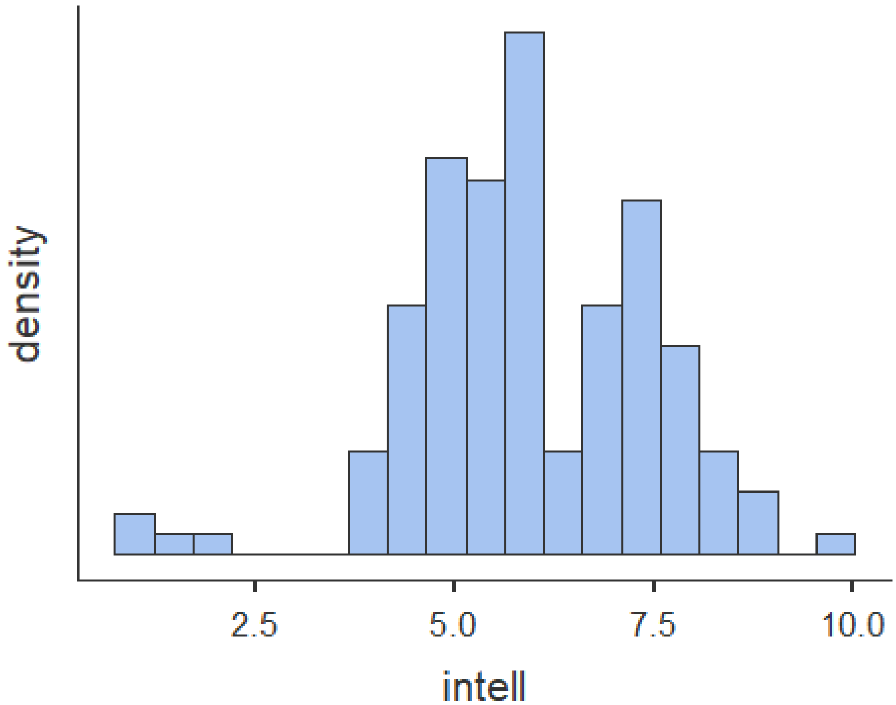
```
##
## Frequencies of Sex
## -------------------------------------------------
##   Levels    Counts    % of Total    Cumulative %
## -------------------------------------------------
##   Female        63          46.3            46.3
##   Male          73          53.7           100.0
## -------------------------------------------------
##
##
## Frequencies of Race
## -------------------------------------------------
##   Levels    Counts    % of Total    Cumulative %
## -------------------------------------------------
##   Latinx        69          50.7            50.7
##   NR             8           5.9            56.6
##   White         59          43.4           100.0
## -------------------------------------------------
```

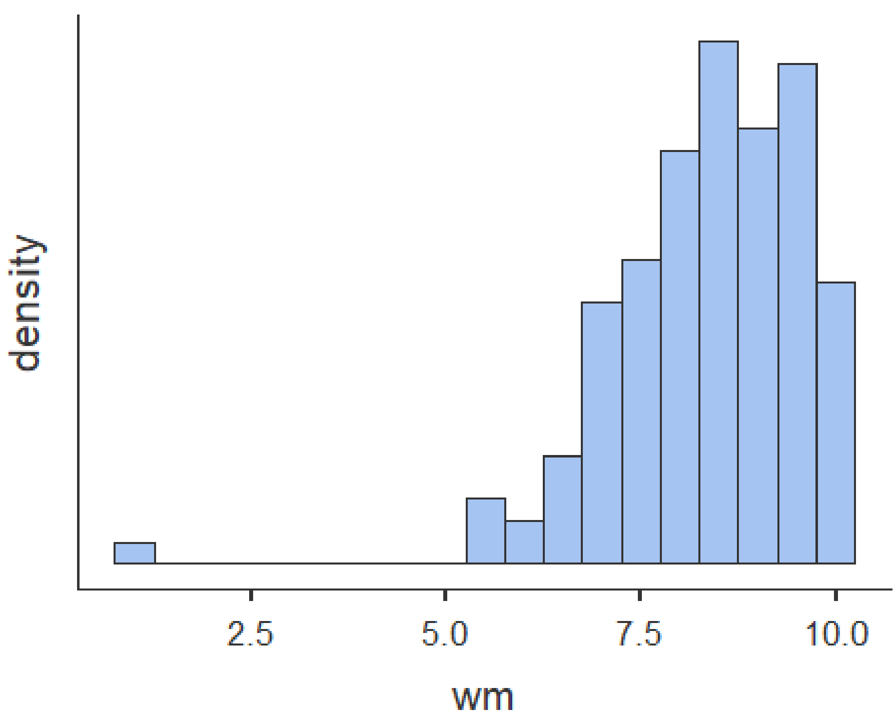
```
#N is all 136 (from 148) now and no missing data --> 12 observations removed (8%)
#Option 2: impute missing values. See Regression_Diagnostics.Rmd for how-to
#Big data set, can drop a few cases --> so going to continue on with more conservative "delete list-wise" data set
```

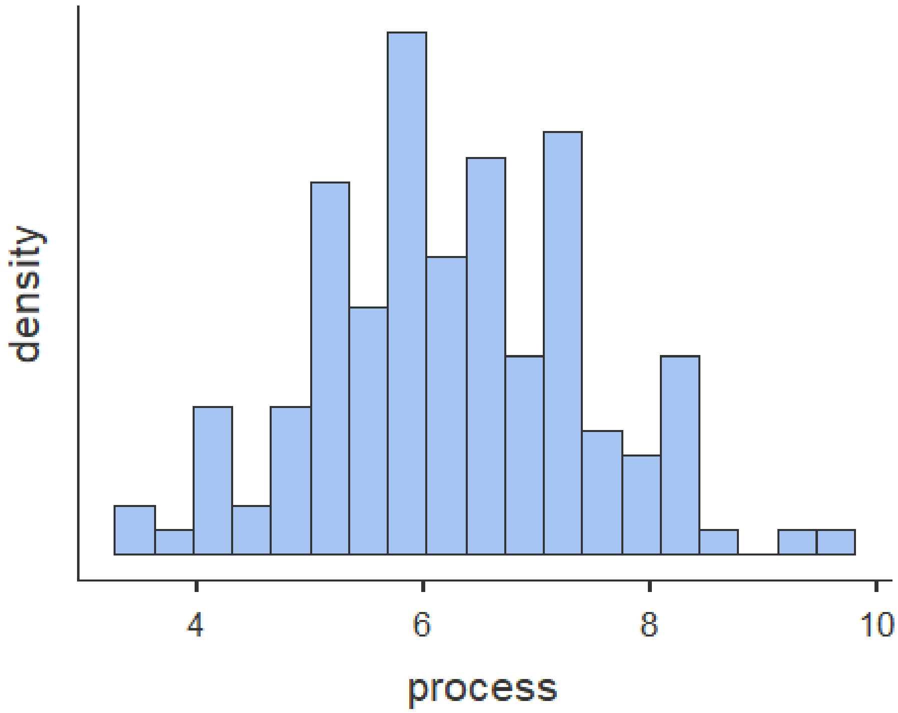
2a. Univariate Normality

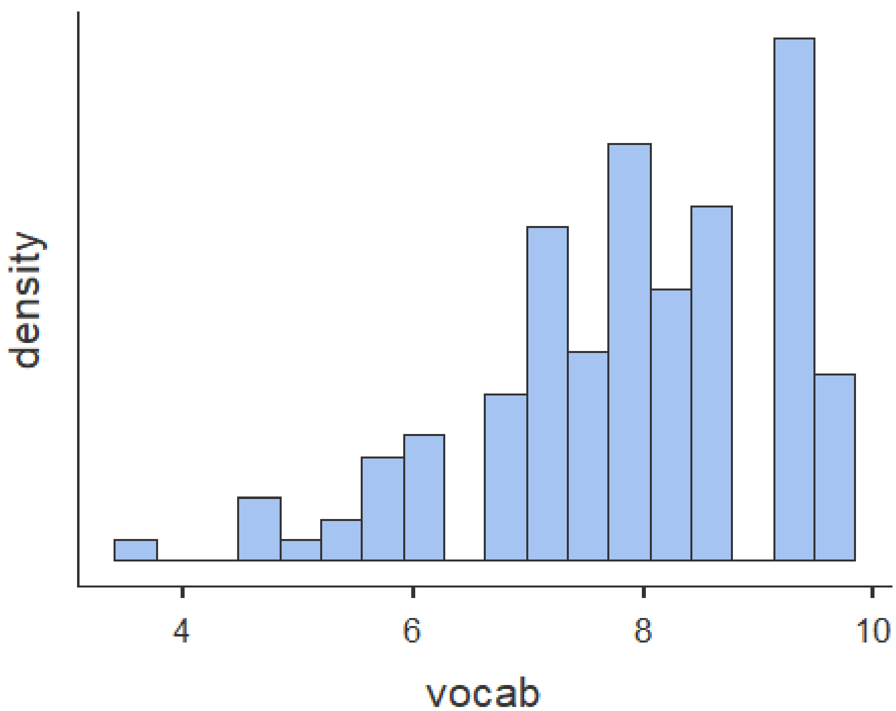
```
#ASSUMPTION: Normal Distribution for continuous variables X and Y (Intelligence) [i.e. histogram, skew +-3, kurtosis +-10]

desc_listwise.hist <- descriptives(data = dat.no.NA,
                          vars = c('intell', 'wm', 'process', 'vocab'),
                          sd = TRUE,
                          range = TRUE,
                          skew = TRUE,
                          kurt = TRUE,
```

```
                       hist = TRUE) # for visual inspection
desc_listwise.hist

##
## DESCRIPTIVES
##
## Descriptives
## -------------------------------------------------------------
##                        intell   wm      process   vocab
## -------------------------------------------------------------
##   N                    136      136     136       136
##   Missing              0        0       0         0
##   Mean                 6.00     8.33    6.23      7.88
##   Median               6.10     8.44    6.19      7.92
##   Standard deviation   1.47     1.25    1.19      1.27
##   Range                8.80     9.00    6.19      6.08
##   Minimum              0.800    1.00    3.33      3.50
##   Maximum              9.60     10.0    9.52      9.58
##   Skewness             -0.608   -1.76   0.0927    -0.864
##   Std. error skewness  0.208    0.208   0.208     0.208
##   Kurtosis             1.62     7.53    -0.181    0.541
##   Std. error kurtosis  0.413    0.413   0.413     0.413
## -------------------------------------------------------------
```


density
2.5
5.0
7.5
10.0
intell


density
2.5
5.0
7.5
10.0
wm



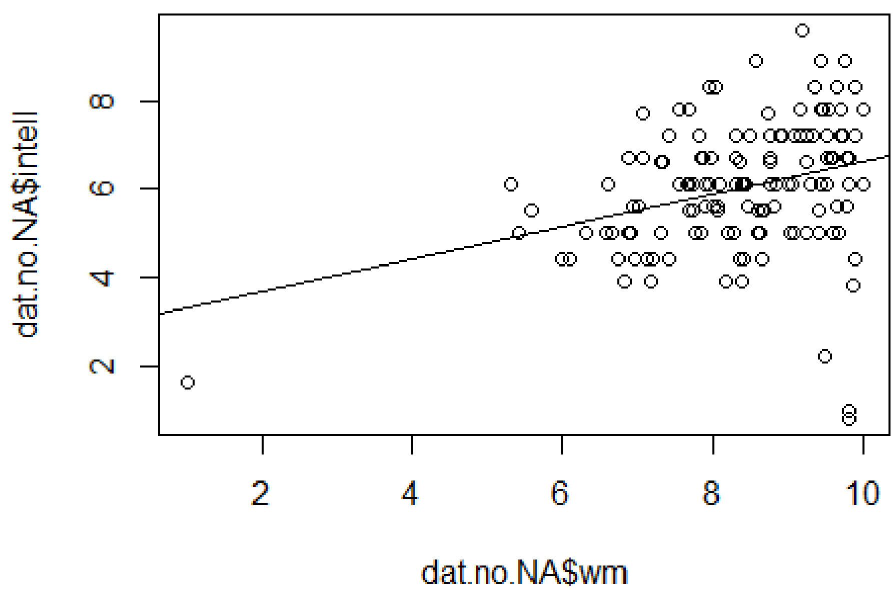



```
# Histogram for Intelligence (intell) is normal
   # Histogram for Working Memory (wm) is normal
```

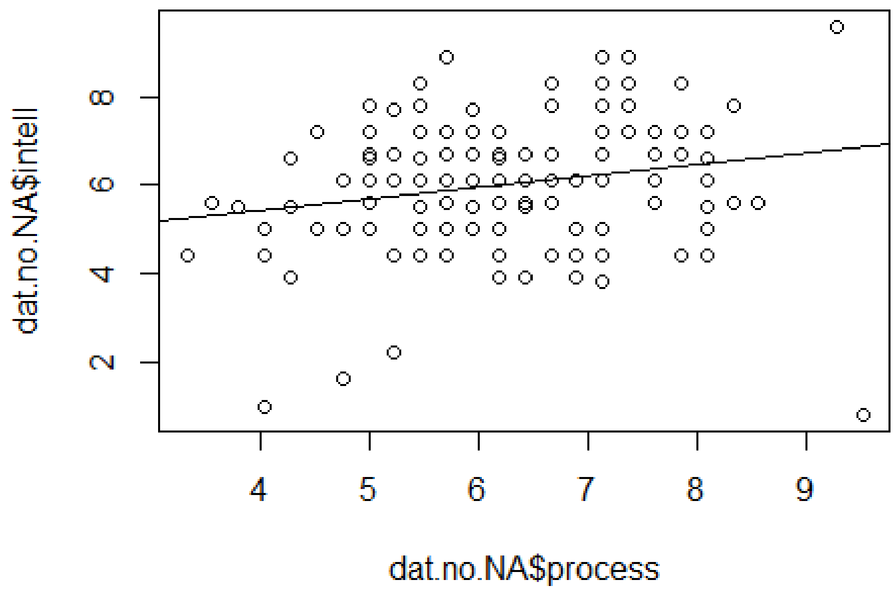
```
  # Histogram for Processing Speed (process) is normal
  # Histogram for Vocabulary (vocab) is normal
  # Skewness - ALL PASS
  # Kurtosis - ALL PASS

#Visual inspection indicates however that there may be outliers
 #Intelligence (intell) in negative tail
 #Working Memory (wm) in negative tail
 #Processing Speed  (process) has no outliers
 #Vocabulary (vocab) in Negative Tail
```

2b. Univariate Linearity

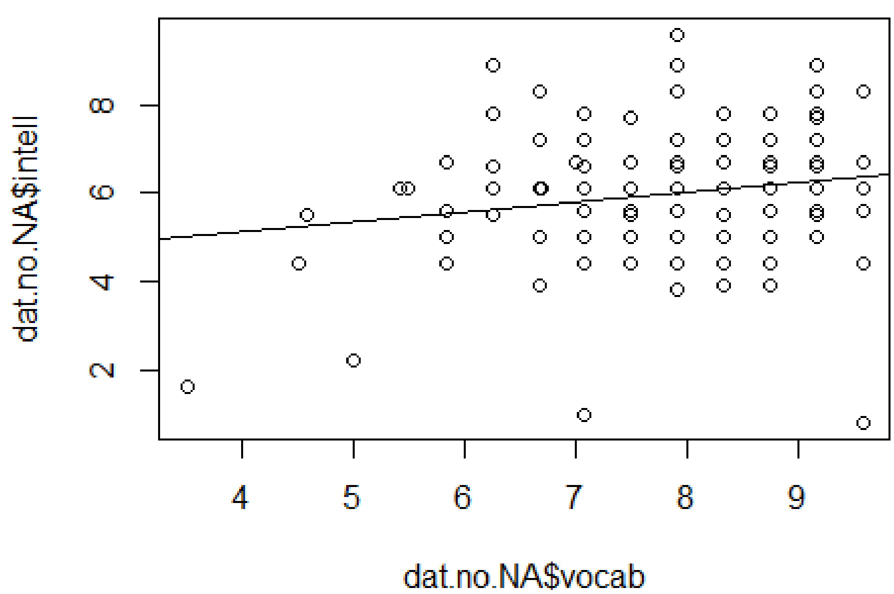
```
# Scatterplots [Assumption 2 and 3a]
plot(dat.no.NA$wm, dat.no.NA$intell, abline(lm(dat.no.NA$intell ~ dat.no.NA$wm)))
```




```
plot(dat.no.NA$process, dat.no.NA$intell, abline(lm(dat.no.NA$intell ~ dat.no.NA$process)))
```




```
plot(dat.no.NA$vocab, dat.no.NA$intell, abline(lm(dat.no.NA$intell ~ dat.no.NA$vocab)))
```



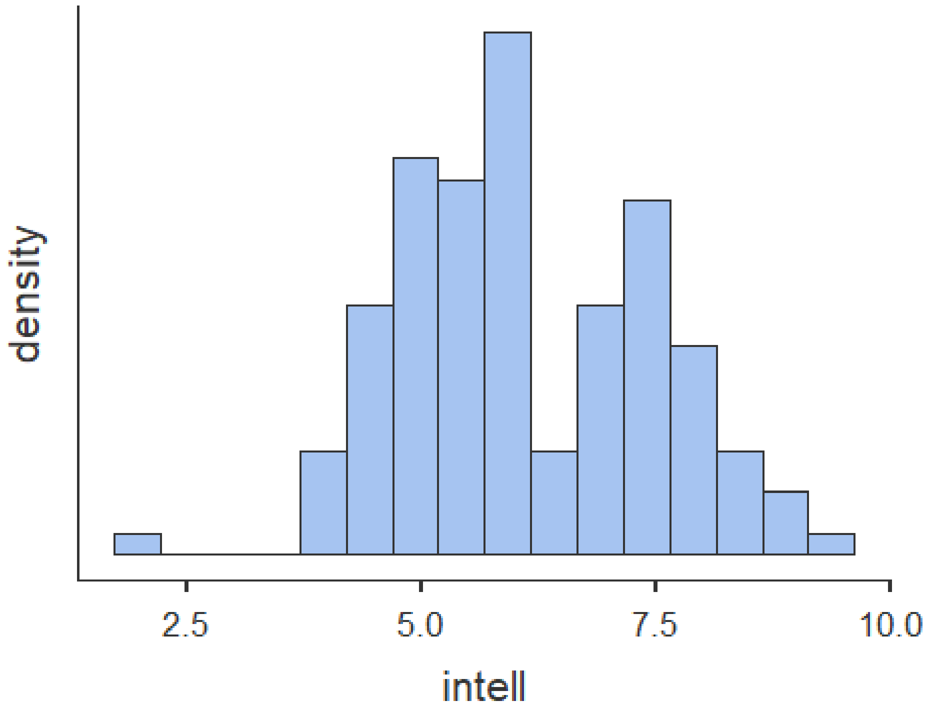
```
#visual inspection indicates a likely linear relationship and is consistent with visual inspection of histograms (step 2a) for outliers
```

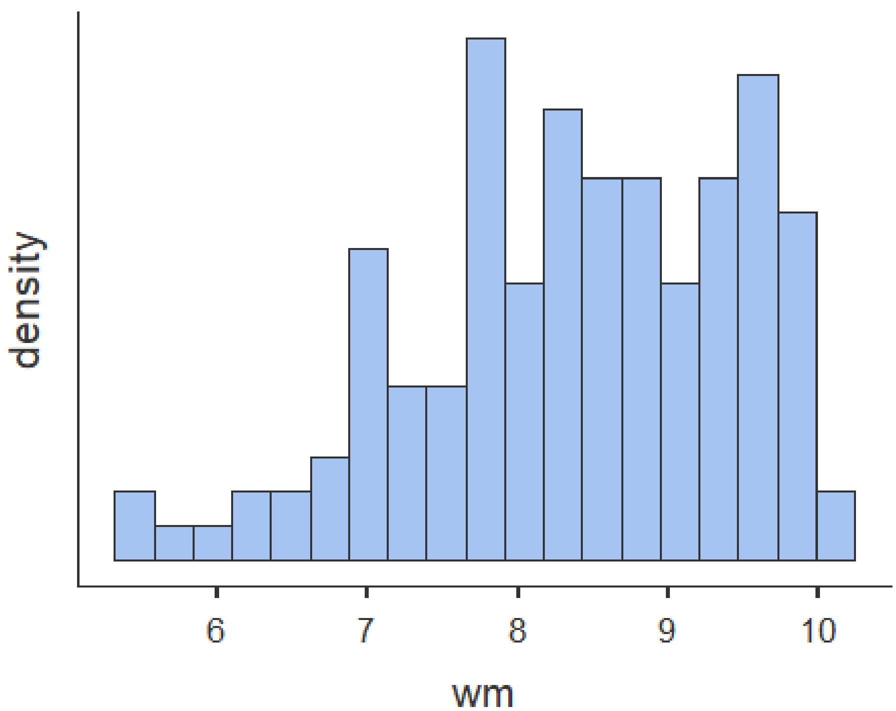
2c. Univariate Outliers

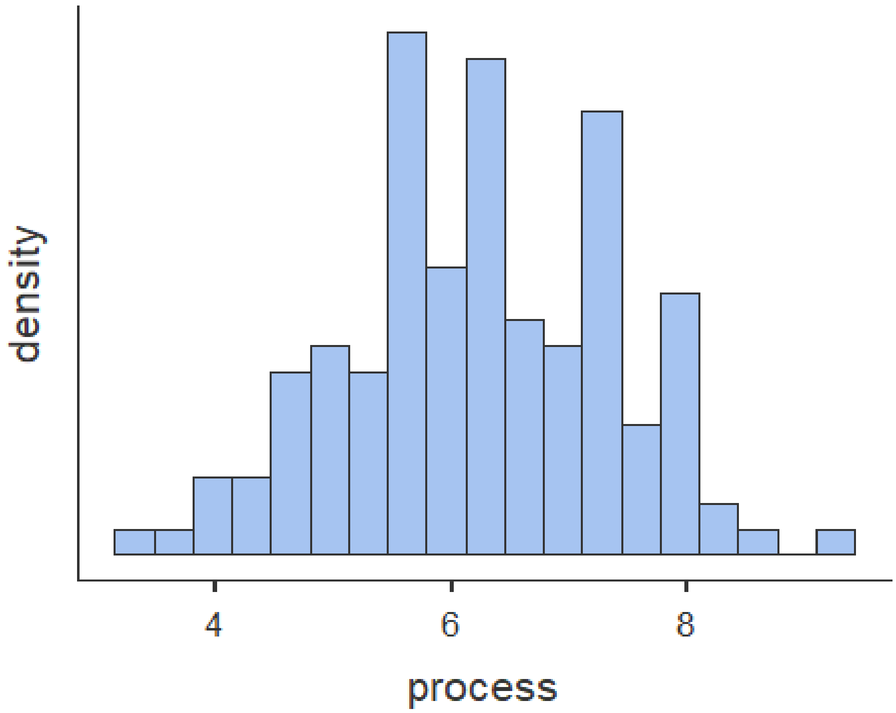
```
#Identify outliers
#scale() converts to z scores - "3" refers to standard deviations
dat.no.NA[abs(scale(dat.no.NA$intell)) > 3, ]

##     StudentID intell   wm process vocab  GPA  age    Sex  Race
## 2           2    1.6 1.00   4.762 3.500 2.95 16.7   Male White
## 33         33    1.0 9.81   4.048 7.083 3.24 16.5 Female White
## 148       148    0.8 9.81   9.524 9.583 2.73 15.4 Female White

dat.no.NA[abs(scale(dat.no.NA$wm)) > 3, ]

##   StudentID intell wm process vocab  GPA  age  Sex  Race
## 2         2    1.6  1   4.762   3.5 2.95 16.7 Male White

dat.no.NA[abs(scale(dat.no.NA$process)) > 3, ]

## [1] StudentID intell    wm        process   vocab     GPA       age
## [8] Sex       Race
## <0 rows> (or 0-length row.names)

dat.no.NA[abs(scale(dat.no.NA$vocab)) > 3, ]

##   StudentID intell wm process vocab  GPA  age  Sex  Race
## 2         2    1.6  1   4.762   3.5 2.95 16.7 Male White

#Intelligence (intell) has 3 univariate outliers
#Working Memory (wm) has 1 univariate outliers
#Processing Speed  (process) has 0 univariate outliers
#Vocabulary (vocab) has 1 univariate outlier
#There are a total of 3 independent observations that contain outliers

#Remove outliers - order here matters
#Order to remove matters - look up for loop for this ugly code
dat.no.uni <- dat.no.NA[!abs(scale(dat.no.NA$intell)) > 3, ]
```

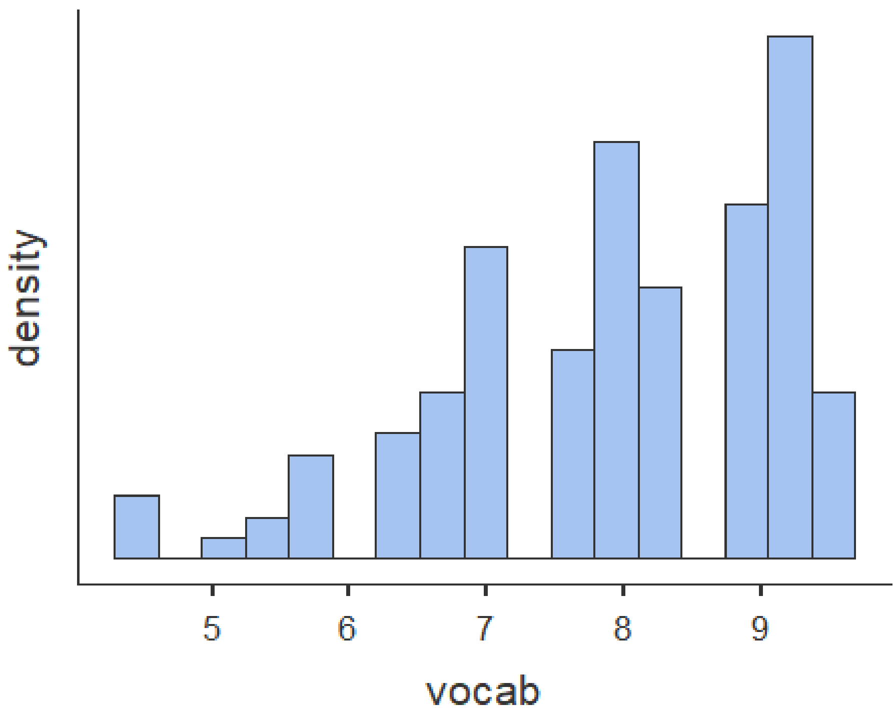
```
#Removed 3 cases that were outside +/-3 SD's for the variables

#Check descriptives for N and assumption of univariate normality in histograms, skew, and kurtosis
desc.no.uni <- descriptives(data = dat.no.uni,
                            vars = c('intell', 'wm', 'process', 'vocab'),
                             sd = TRUE,
                             range = TRUE,
                             skew = TRUE,
                             kurt = TRUE,
                             hist = TRUE) # for visual inspection
desc.no.uni
```

```
##
##  DESCRIPTIVES
##
##  Descriptives
##  -----------------------------------------------------------------
##                          intell     wm        process    vocab
##  -----------------------------------------------------------------
##    N                        133       133        133       133
##    Missing                    0         0          0         0
##    Mean                    6.11      8.37       6.23      7.91
##    Median                  6.10      8.43       6.19      7.92
##    Standard deviation      1.28      1.08       1.15      1.21
##    Range                   7.40      4.67       5.95      5.08
##    Minimum                 2.20      5.33       3.33      4.50
##    Maximum                 9.60      10.0       9.29      9.58
##    Skewness              0.0896    -0.545    -0.0147    -0.731
##    Std. error skewness    0.210     0.210      0.210     0.210
##    Kurtosis             -0.0299    -0.229     -0.301    0.0134
##    Std. error kurtosis    0.417     0.417      0.417     0.417
##  -----------------------------------------------------------------
```


density
2.5
5.0
7.5
10.0
intell


density
6
7
8
9
10
wm





 # Histogram for Intelligence (intell) is normal

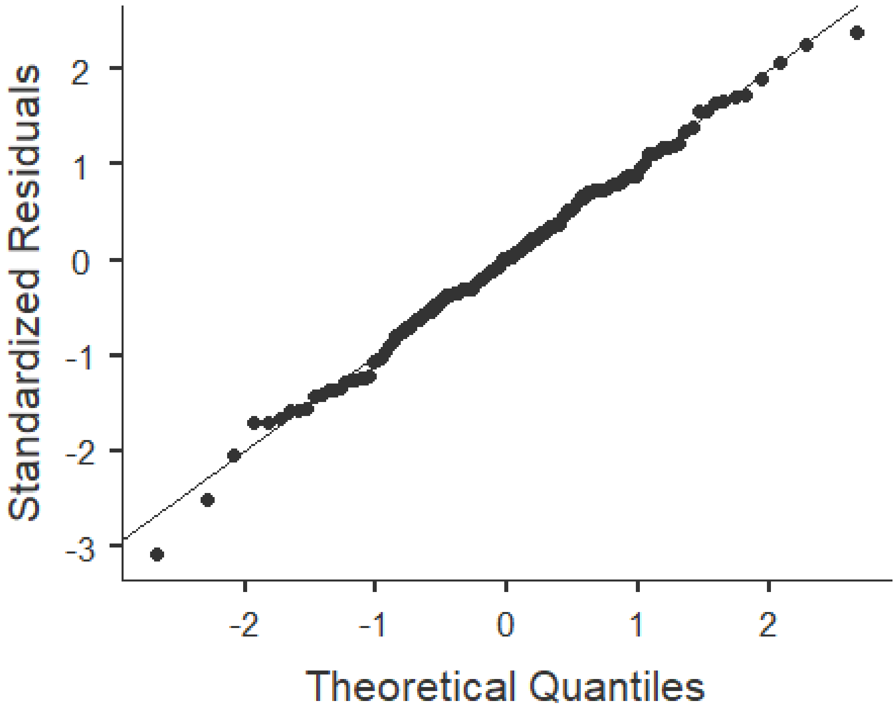
 # Histogram for Working Memory (wm) is normal

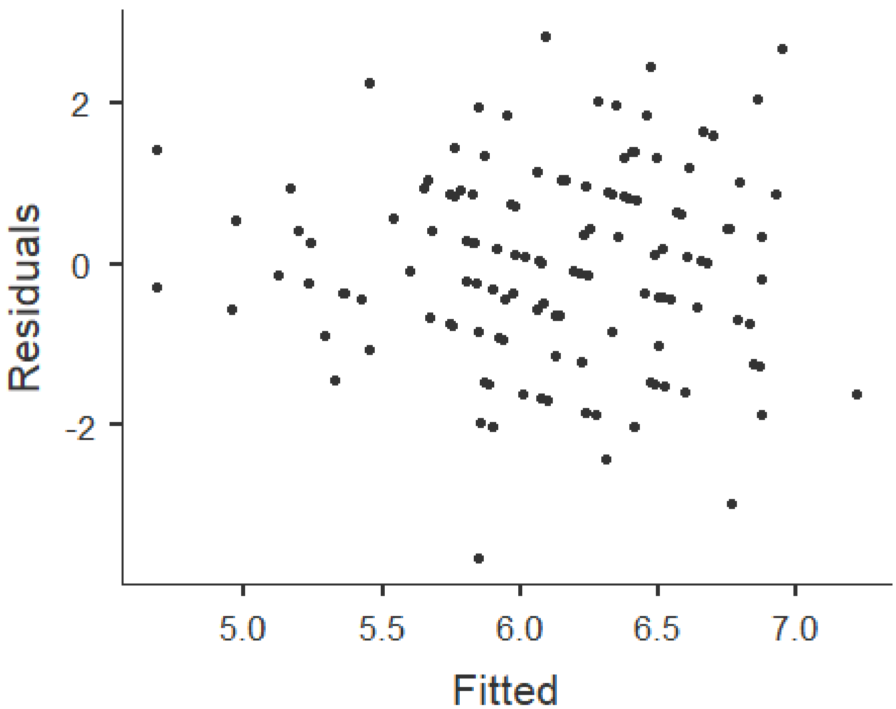
```
    # Histogram for Processing Speed (process) is normal
    # Histogram for Vocabulary (vocab) is normal
    # Skewness - ALL PASS
    # Kurtosis - ALL PASS

# N is now 133 after removing 3 independent cases with univariate outliers,
    #was 136 after removing 8 observations with missing parameters
    #was 148 originally

# everything is now within range of normal distribution
# if this did not fix the problem, square root or log transform may help - See
Regression_Diagnostics.Rmd for how-to
```

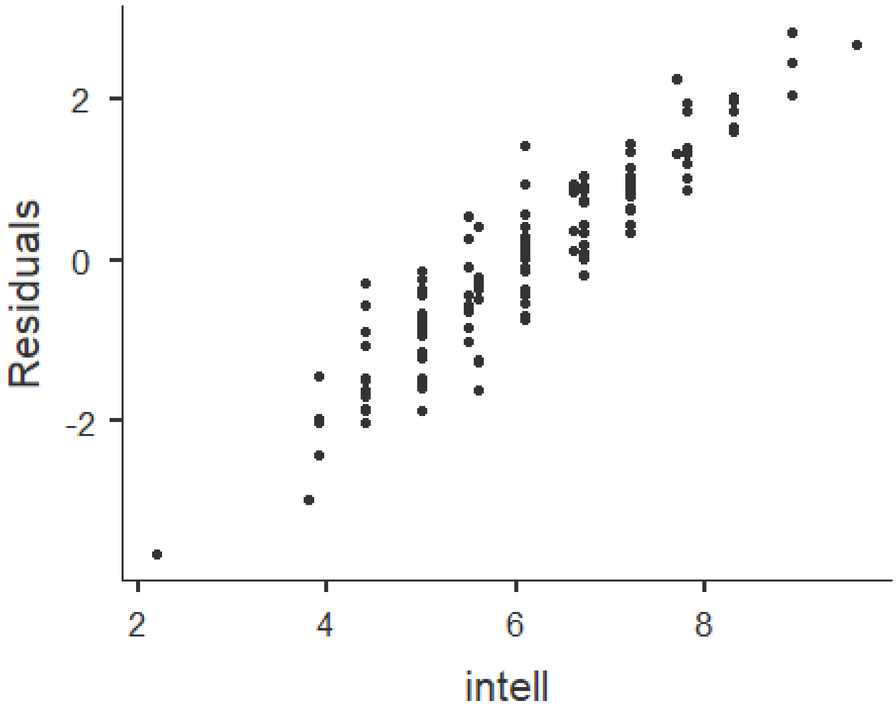
3a. Multivariate Normality

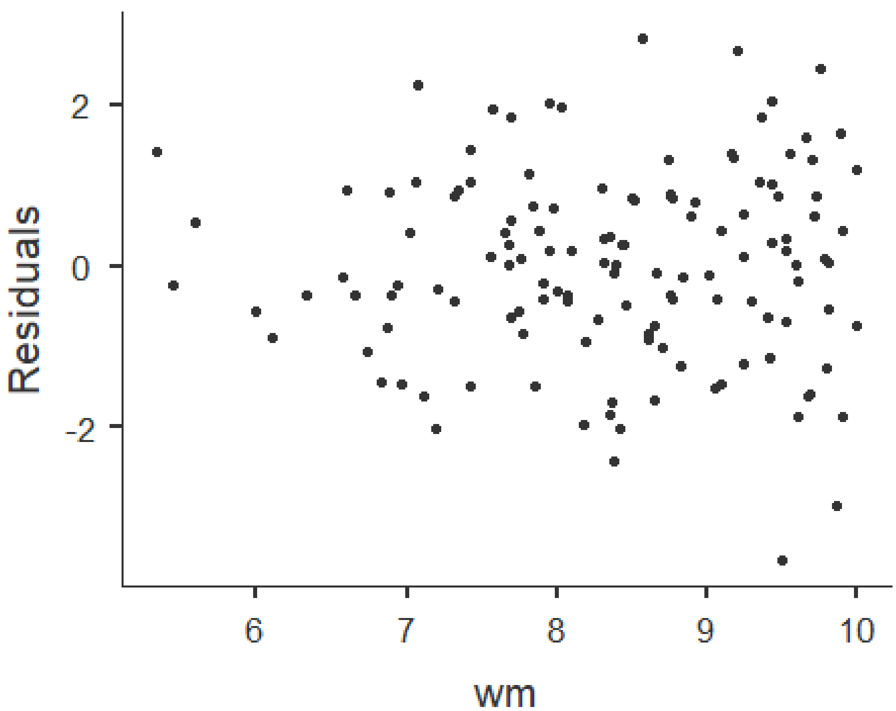
```
#look at residuals and the Q-Q plot
#Observe Leverage (Mahalanobis' Distance) + Discrepancy (= Influence; Cook's Distance)

model.multi_norm <- linReg(data = dat.no.uni,
              dep = 'intell',
              covs = c('wm', 'process', 'vocab'),
              blocks = list(c('wm', 'process', 'vocab')),
              modelTest = TRUE,
              r2Adj = TRUE,
              stdEst = TRUE,
              ciStdEst = TRUE,
              qqPlot = TRUE,  ##QQ plot
              resPlots = TRUE) ##residuals plot

model.multi_norm

##
##  LINEAR REGRESSION
##
##  Model Fit Measures
```

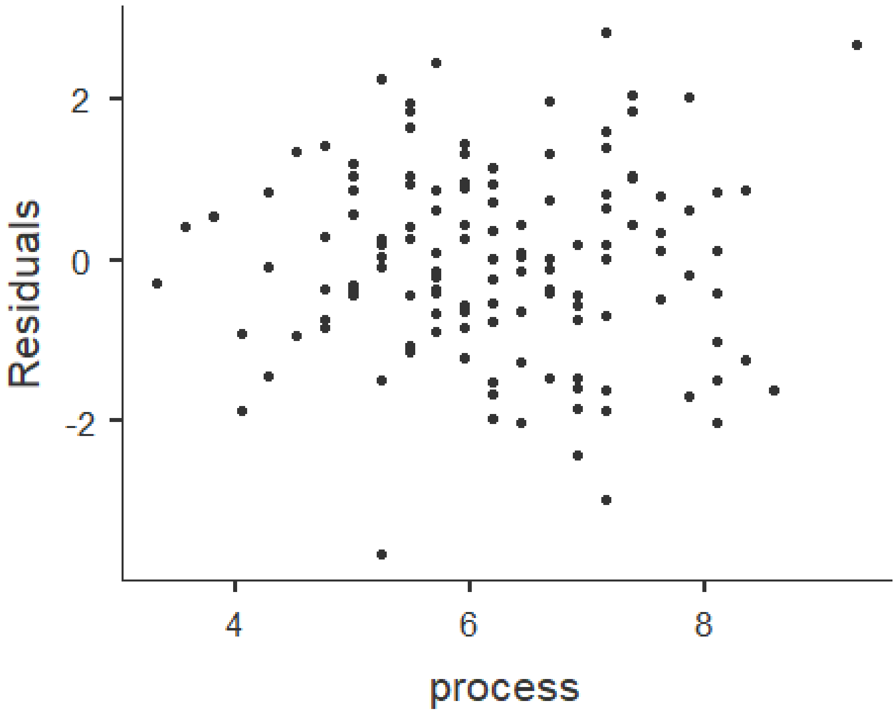
```
## ---------------------------------------------------------------------------
##   Model    R       R²       Adjusted R²    F      df1    df2    p
## ---------------------------------------------------------------------------
##       1    0.392    0.154         0.134    7.83     3    129    < .001
## ---------------------------------------------------------------------------
##
##
## MODEL SPECIFIC RESULTS
##
## MODEL 1
##
## Model Coefficients
## ------------------------------------------------------------------------------------------------
##   Predictor    Estimate    SE       t       p       Stand. Estimate    Lower      Upper
## ------------------------------------------------------------------------------------------------
##   Intercept       1.060    1.0819    0.980    0.329
##   wm            0.322    0.1008    3.194    0.002            0.271     0.10303    0.439
##   process         0.184    0.0948    1.940    0.055            0.165    -0.00327    0.332
##   vocab          0.154    0.0860    1.788    0.076            0.145    -0.01548    0.306
## ------------------------------------------------------------------------------------------------
##
##
## ASSUMPTION CHECKS
```

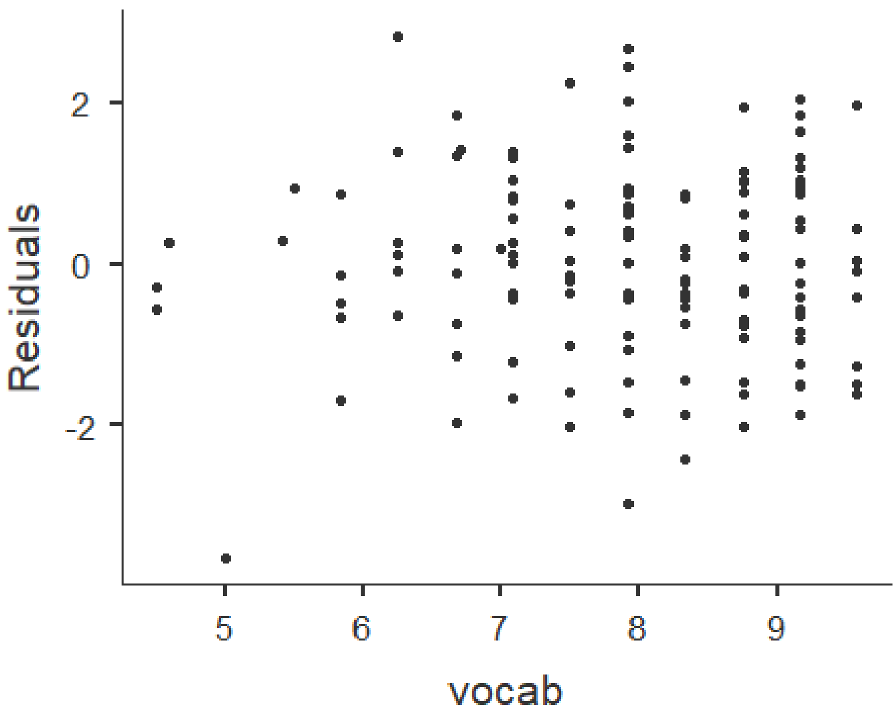

Standardized Residuals
2
1
0
-1
-2
-3
-2
-1
0
1
2
Theoretical Quantiles


Residuals
2
0
-2
5.0
5.5
6.0
6.5
7.0
Fitted


Residuals
2
0
-2
2
4
6
8
intell


Residuals
2
0
-2
6
7
8
9
10
wm





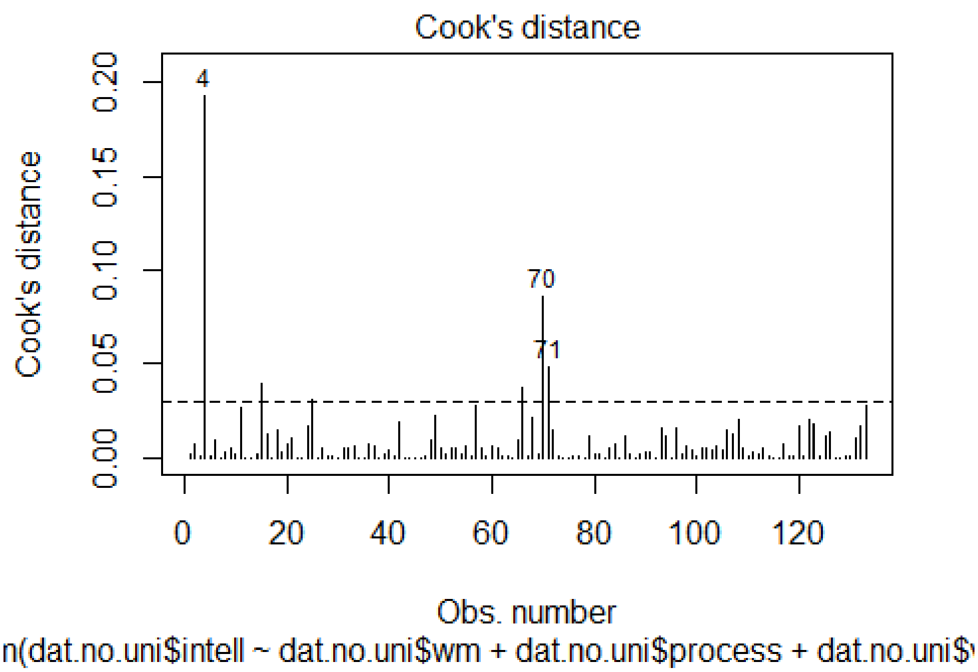

```
#Alternate not using jvm library
#model <- lm(Amount ~ Belief + Need, data = dat.no.uni)
```

```
#plot(model)

#inspection of plots of predictors vs residuals indicates likely multivariate normality, but possible heteroscadasticity
#inspection of theoretical quantiles vs standardized residuals indicates a possible problem with multivariate distance and leverage
#as such, Cook's distance - a measure of influence - will be used to test for multivariate normality
#for Mahalanobis' Distance (leverage only), see Regression_Diagnostics.Rmd for how-to
```

3b. Multivariate Outliers

```
#Check and remove multivariate outliers based on Cook's distance (CD)
#CD = Influence = Leverage + Discrepancy (Discrepancy = how much an observation deviates from the overall pattern of the model)

#create model
model.cook <- lm(dat.no.uni$intell ~ dat.no.uni$wm + dat.no.uni$process + dat.no.uni$vocab)
model.cook

##
## Call:
## lm(formula = dat.no.uni$intell ~ dat.no.uni$wm + dat.no.uni$process +
##     dat.no.uni$vocab)
##
## Coefficients:
##        (Intercept)       dat.no.uni$wm  dat.no.uni$process
##             1.0597              0.3219              0.1839
##   dat.no.uni$vocab
##             0.1537

summary(model.cook)

##
## Call:
## lm(formula = dat.no.uni$intell ~ dat.no.uni$wm + dat.no.uni$process +
```

```
##     dat.no.uni$vocab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6496 -0.7296  0.0236  0.8555  2.8079
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.0597     1.0819   0.980  0.32916
## dat.no.uni$wm        0.3219     0.1008   3.194  0.00177 **
## dat.no.uni$process   0.1839     0.0948   1.940  0.05457 .
## dat.no.uni$vocab     0.1537     0.0860   1.788  0.07616 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.194 on 129 degrees of freedom
## Multiple R-squared:  0.154,  Adjusted R-squared:  0.1343
## F-statistic: 7.826 on 3 and 129 DF,  p-value: 7.708e-05

#find cook's distance for that model
dat.no.uni$cook <- cooks.distance(model.cook)

#create the cutoff [> 4/N]
cook.cutoff <- 4/nrow(dat.no.uni)
cook.cutoff

## [1] 0.03007519

# 4/133 --> cutoff = .03

#plot it out
plot(model.cook, which = 4, cook.levels = cook.cutoff)

#Add a cutoff line
abline(h = cook.cutoff, lty = 2)
```




```
#Show and remove all outliers above your cutoff line

dat.no.uni[(dat.no.uni$cook) > cook.cutoff,]

##    StudentID intell    wm process vocab  GPA  age    Sex   Race       cook
## 6          6    2.2 9.500   5.238 5.000 2.97 16.1   Male  White 0.19243975
## 18        18    8.9 8.568   7.143 6.250 3.14 16.0   Male Latinx 0.04022736
## 31        31    6.1 5.333   4.762 6.700 2.96 16.3 Female  White 0.03066145
## 75        75    3.8 9.867   7.143 7.917 3.35 15.5 Female Latinx 0.03793013
## 79        79    9.6 9.200   9.286 7.917 3.04 18.2 Female Latinx 0.08610298
## 80        80    4.4 9.905   4.048 8.333 3.46 15.8 Female Latinx 0.04870741

dat.final <- dat.no.uni[!(dat.no.uni$cook) > cook.cutoff,]

#N is now 127 after removing 6 multivariate outlier observations
  #was 133 after removing 3 univariate outlier obervations,
  #was 136 after removing 8 observations with missing parameters
  #was 148 originally (total 21 observations removed from orginal dataset - 14%)
```

4. Heteroscedasticity

```
#Breusch-Pagan test
#H0 = no change in variance across residuals.
model.breusch_pagan <- lm(dat.final$intell ~ dat.final$wm + dat.final$process +
dat.final$vocab)
ncvTest(model.breusch_pagan)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1.470042, Df = 1, p = 0.22534

#not significant = homoscedastic
#If violated use Box-cox transformation [boxcox(model)] in library MASS
```

5. Multi-collinearity

```
#Tolerance = 1 - R squared --> for our purpose < .4 is bad
#VIF = 1/Tolerance ---> for our purpose > 2.5 is bad
#Small VIF values (or higher Tolerance values) indicates low correlation among variables under
ideal conditions

#Multicollinearity occurs when two or more predictors in the model are correlated and provide
redundant information about the response. Multicollinearity is measured by variance inflation
factors (VIF) and tolerance. If VIF value exceeds 4.0, or tolerance less than 0.2 then there is a
problem with multicollinearity according to Hair et al. (2010).

model.wm_process_vocab <- linReg(data = dat.final,
             dep = 'intell',
             cov = c('wm', 'process', 'vocab'),
             blocks = list(c('wm', 'process', 'vocab')),
             modelTest = TRUE,
             r2Adj = TRUE,
             stdEst = TRUE,
             ciStdEst = TRUE,
```

```
            collin = TRUE) #this line does the thing
model.wm_process_vocab

##
## LINEAR REGRESSION
##
## Model Fit Measures
## -----------------------------------------------------------------------
##   Model    R        R²       Adjusted R²    F       df1    df2    p
## -----------------------------------------------------------------------
##       1    0.455    0.207          0.188    10.7      3    123    < .001
## -----------------------------------------------------------------------
##
##
## MODEL SPECIFIC RESULTS
##
## MODEL 1
##
## Model Coefficients
## ------------------------------------------------------------------------------------------
##   Predictor    Estimate    SE        t        p         Stand. Estimate    Lower      Upper
## ------------------------------------------------------------------------------------------
##   Intercept     0.8774    1.0066    0.872    0.385
##   wm            0.4574    0.0946    4.837    < .001            0.4075     0.2407     0.574
##   process       0.0587    0.0892    0.659    0.511             0.0555    -0.1113     0.222
##   vocab         0.1340    0.0789    1.698    0.092             0.1365    -0.0226     0.296
## ------------------------------------------------------------------------------------------
##
##
## ASSUMPTION CHECKS
##
## Collinearity Statistics
## --------------------------------
##             VIF     Tolerance
```

```
## --------------------------------
##   wm        1.10       0.908
##   process    1.10       0.908
##   vocab      1.00       0.997
## --------------------------------

#Tolerance for all variables indicates low/no multicollinearity
```

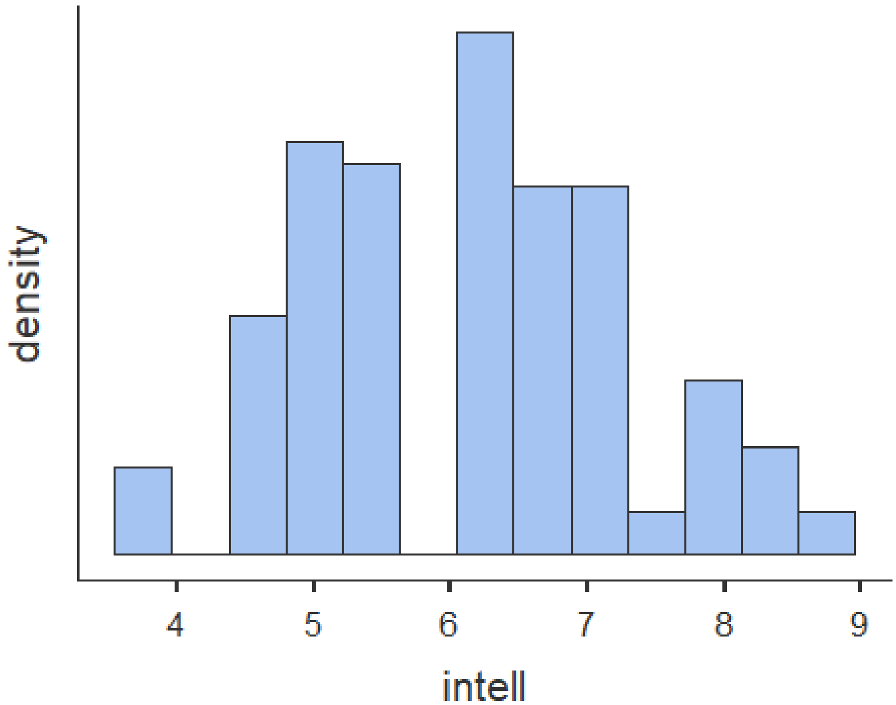
Data Analysis 1. Descriptive Statistics 2. Correlations 3. Center Data (if useful) 4. Simple Regression 5. Hierarchical Model Comparison 6. Visualization

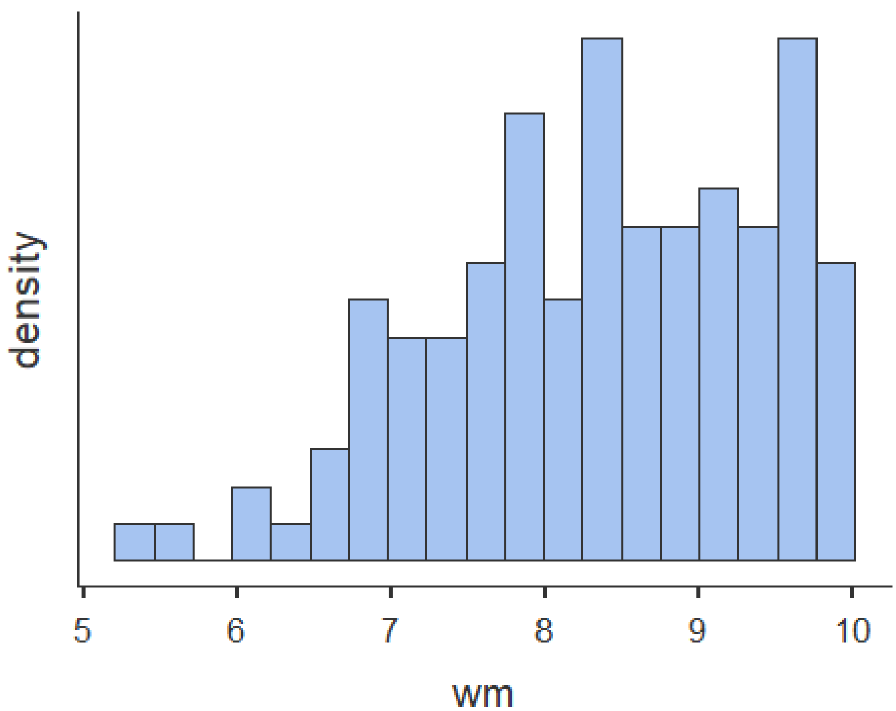
1. Descriptive Statistics

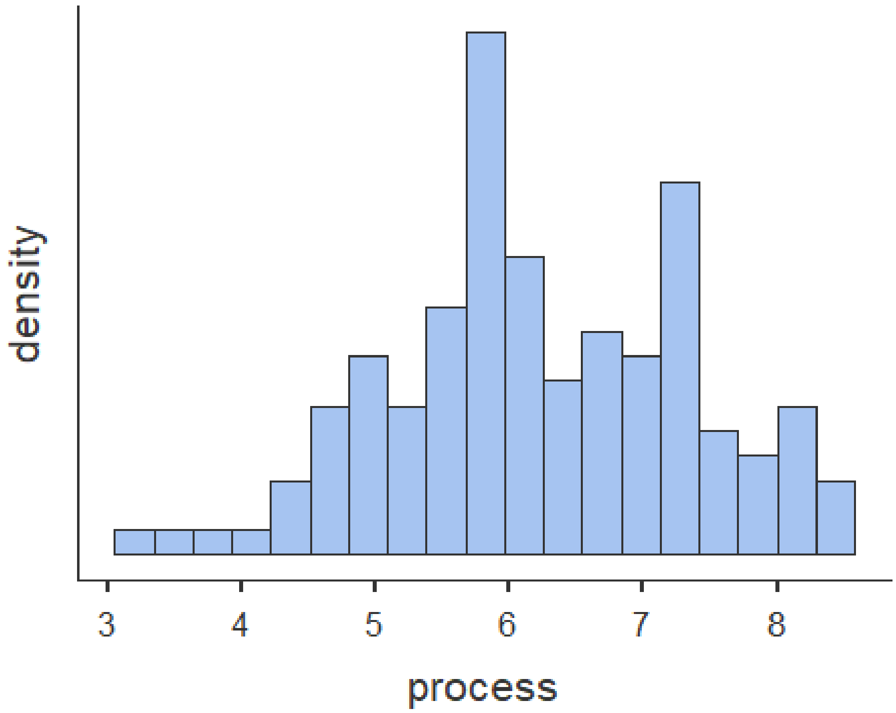
```
#Prerequisite: predictors and outcome all measured on continuous level
#Assumptions:
 #1. Normal Distribution for X and Y (Product) [i.e. histogram, skew +-3, kurtosis +-10]
  # Histograms observed are normal
  # Skewness - ALL PASS
  # Kurtosis - ALL PASS
  # Observations with missing parameters were removed (see Diagnostics)
  # univariate outliers were removed (see Diagnostics)
  # multivariate outliers were removed (see Diagnostics)

 #2. Linear Relationship beween X and Y
  # Visual inspection of scatterplot and prediction model line in Diagnostics 2b. indicate a linear relationship
 #3. Homoscedasticity - OK (see Diagnostics)
 #4. Multicollearity -diagnostics completed - OK (see Diagnostics)

#N is now 127 after removing 6 multivariate outlier observations
  #was 133 after removing 3 univariate outlier obervations,
  #was 136 after removing 8 observations with missing parameters
  #was 148 originally (total 21 observations removed from orginal dataset - 14%)

desc.final <- descriptives(data = dat.final,
                    vars = c('intell', 'wm', 'process', 'vocab', 'age', 'Sex', 'Race'),
```

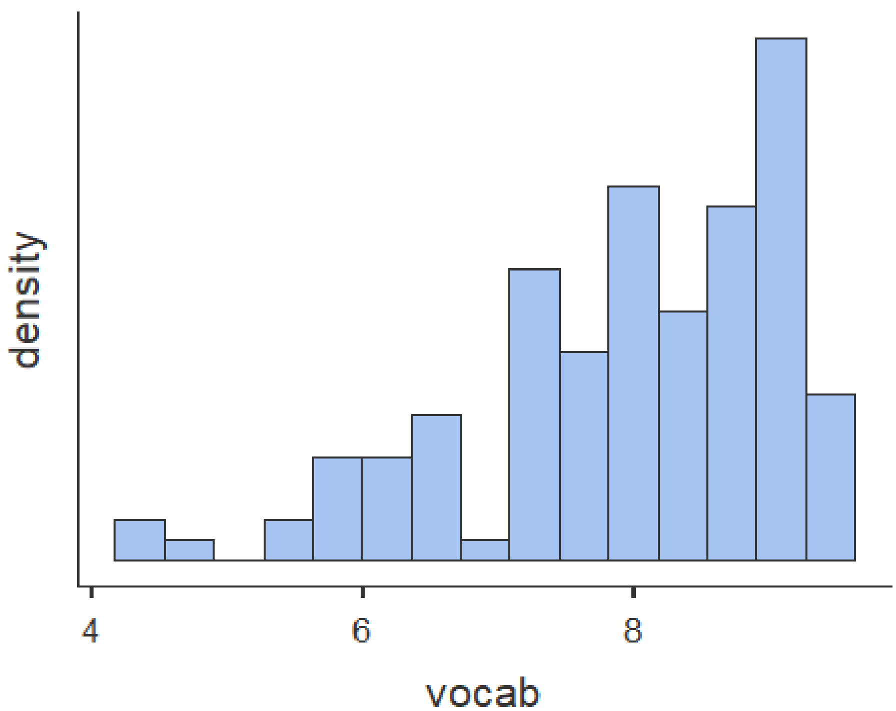
```
                hist = TRUE,
                sd = TRUE,
                range = TRUE,
                skew = TRUE,
                kurt = TRUE,
                freq = TRUE)
desc.final

##
## DESCRIPTIVES
##
## Descriptives
## ---------------------------------------------------------------------------
##                        intell   wm      process   vocab    age      Sex   Race
## ---------------------------------------------------------------------------
##   N                      127      127     127       127      127     127   127
##   Missing                  0        0       0         0        0       0     0
##   Mean                   6.13     8.35    6.23      7.95     16.3
##   Median                 6.10     8.40    6.19      7.92     16.4
##   Standard deviation     1.17     1.05    1.11      1.20     0.817
##   Range                  5.00     4.57    5.24      5.08     3.90
##   Minimum                3.90     5.43    3.33      4.50     14.5
##   Maximum                8.90     10.0    8.57      9.58     18.4
##   Skewness               0.155   -0.445  -0.104    -0.755   -0.0234
##   Std. error skewness    0.215    0.215   0.215     0.215    0.215
##   Kurtosis              -0.633   -0.404  -0.383     0.0909  -0.552
##   Std. error kurtosis    0.427    0.427   0.427     0.427    0.427
## ---------------------------------------------------------------------------
##
##
## FREQUENCIES
##
## Frequencies of Sex
## ------------------------------------------
```

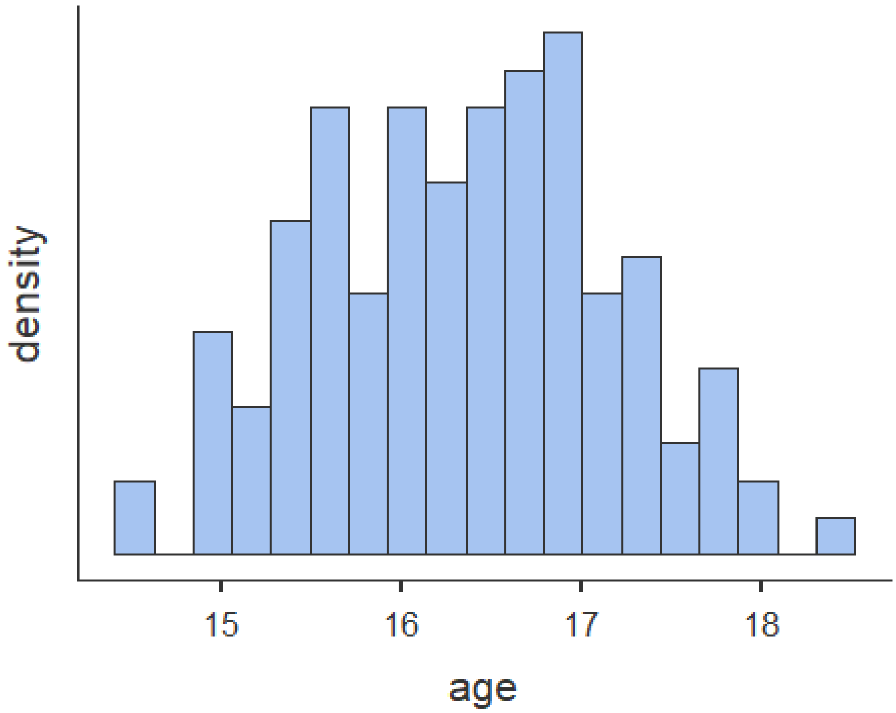
```
##   Levels   Counts   % of Total   Cumulative %
##  -----------------------------------------------
##   Female     57        44.9          44.9
##   Male       70        55.1         100.0
##  -----------------------------------------------
##
##
##  Frequencies of Race
##  -----------------------------------------------
##   Levels   Counts   % of Total   Cumulative %
##  -----------------------------------------------
##   Latinx     65        51.2          51.2
##   NR          8         6.3          57.5
##   White      54        42.5         100.0
##  -----------------------------------------------
```


density
4
5
6
7
8
9
intell


density
5
6
7
8
9
10
wm


density
3
4
5
6
7
8
process


density
4
6
8
vocab



2. Correlations

```
# Correlations of predictor and outcome variables
cortable <- corrMatrix(data = dat.final,
                       vars = c('intell', 'wm', 'process', 'vocab'),
                       flag = TRUE)
cortable

##
##  CORRELATION MATRIX
##
##  Correlation Matrix
##  ----------------------------------------------------------------
##                              intell    wm       process    vocab
##  ----------------------------------------------------------------
##    intell     Pearson's r         ☐     0.431      0.185     0.158
##               p-value             ☐    < .001      0.038     0.075
##
##    wm         Pearson's r                    ☐     0.302     0.047
##               p-value                        ☐    < .001     0.598
```

```
##
##   process   Pearson's r          ☐   0.046
##             p-value              ☐   0.608
##
##   vocab     Pearson's r          ☐
##             p-value              ☐
## ---------------------------------------------------------------
##   Note. * p < .05, ** p < .01, *** p < .001
```

3. Center data (if useful)

```
# Center only predictor variables
# c = x - M
# Centering only changes the intercept for regression equation
  # Centering means, on average (instead of zero) across all predictor variables Y intercept is [coefficient for X units]
# Center predictors wm, process, vocab
dat.final$wm.centered <- dat.final$wm - mean(dat.final$wm)
dat.final$process.centered <- dat.final$process - mean(dat.final$process)
dat.final$vocab.centered <- dat.final$vocab - mean(dat.final$vocab)

#NOT USEFUL - We will not center data for models of these predictors, as negative predicted values would not make much sense for a test with no possible score below zero.
```

4. Simple Regression

```
# Simple regression
# R = correlation between observed scores and predicted scores
# R squared = percentage of variance explained
# t = Estimate / SE
# df1 = k = number of predictors
# df2 = N - k - 1 [k is number of predictors]
# H0: B0 = 0; H0; R squared = 0

model.wm <- linReg(data = dat.final,
             dep = 'intell',
```

```
            covs = c('wm'),
            blocks = list('wm'),
            modelTest = TRUE,
            stdEst = TRUE,
            ci = TRUE)
model.wm #1 fit

##
##  LINEAR REGRESSION
##
##  Model Fit Measures
##  ---------------------------------------------------------
##    Model    R        R²       F       df1    df2    p
##  ---------------------------------------------------------
##        1    0.431    0.185    28.5      1    125    < .001
##  ---------------------------------------------------------
##
##
##  MODEL SPECIFIC RESULTS
##
##  MODEL 1
##
##  Model Coefficients
##  ------------------------------------------------------------------------------------------
##    Predictor    Estimate    SE        Lower    Upper    t       p         Stand. Estimate
##  ------------------------------------------------------------------------------------------
##    Intercept       2.091    0.7624    0.582    3.600    2.74     0.007
##    wm              0.483    0.0906    0.304    0.663    5.34    < .001              0.431
##  ------------------------------------------------------------------------------------------

model.process <- linReg(data = dat.final,
            dep = 'intell',
            covs = c('process'),
            blocks = list('process'),
```

```
            modelTest = TRUE,
            stdEst = TRUE,
            ci = TRUE)
model.process #2 fit

##
## LINEAR REGRESSION
##
## Model Fit Measures
## ---------------------------------------------------------
##   Model   R       R²       F      df1   df2   p
## ---------------------------------------------------------
##      1   0.185   0.0341   4.42     1   125   0.038
## ---------------------------------------------------------
##
##
## MODEL SPECIFIC RESULTS
##
## MODEL 1
##
## Model Coefficients
## -----------------------------------------------------------------------------------------
##   Predictor   Estimate   SE       Lower    Upper   t      p        Stand. Estimate
## -----------------------------------------------------------------------------------------
##   Intercept      4.910   0.5884   3.7458   6.075   8.35   < .001
##   process        0.196   0.0930   0.0114   0.380   2.10    0.038             0.185
## -----------------------------------------------------------------------------------------

model.vocab <- linReg(data = dat.final,
            dep = 'intell',
            covs = c('vocab'),
            blocks = list('vocab'),
            modelTest = TRUE,
            stdEst = TRUE,
```

```
           ci = TRUE)
model.vocab #3 fit

## 
## LINEAR REGRESSION
## 
## Model Fit Measures
## ----------------------------------------------------------
##   Model   R       R²       F      df1   df2   p
## ----------------------------------------------------------
##     1   0.158   0.0251   3.21     1   125   0.075
## ----------------------------------------------------------
## 
## 
## MODEL SPECIFIC RESULTS
## 
## MODEL 1
## 
## Model Coefficients
## -------------------------------------------------------------------------------------------
##   Predictor   Estimate   SE       Lower     Upper    t      p        Stand. Estimate
## -------------------------------------------------------------------------------------------
##   Intercept     4.892   0.6969    3.5127   6.271   7.02   < .001
##   vocab         0.155   0.0867   -0.0162   0.327   1.79    0.075            0.158
## -------------------------------------------------------------------------------------------
```

5. Hierarchical Model Comparison

```
# Model comparison
# H0 = delta of R squared = 0
compare <- linReg(data = dat.final,
            dep = 'intell',
            covs = c('wm', 'process', 'vocab'),
            blocks = list(
                  list('wm'),
```

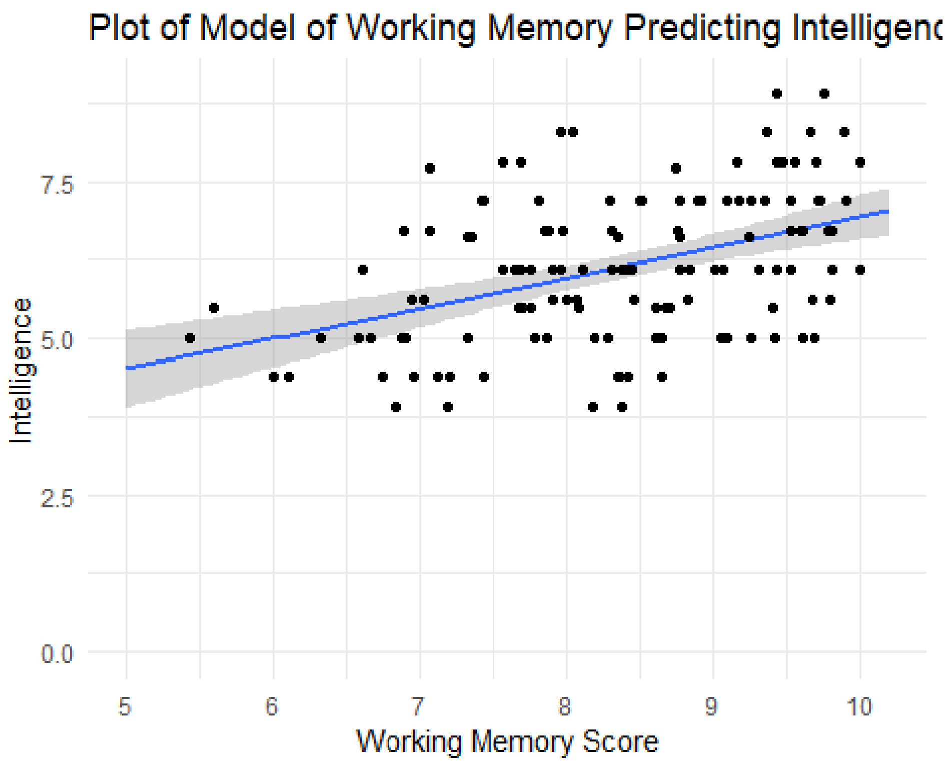
```
                    list('vocab', 'process')),
             modelTest = TRUE,
             stdEst = TRUE,
             ci = TRUE)
compare

##
## LINEAR REGRESSION
##
## Model Fit Measures
## --------------------------------------------------------
##   Model    R        R²       F       df1    df2    p
## --------------------------------------------------------
##       1    0.431    0.185    28.5      1    125    < .001
##       2    0.455    0.207    10.7      3    123    < .001
## --------------------------------------------------------
##
##
## Model Comparisons
## -------------------------------------------------------------
##   Model          Model    <U+0394>R²      F       df1    df2    p
## -------------------------------------------------------------
##       1    -        2    0.0219    1.70      2    123    0.187
## -------------------------------------------------------------
##
##
## MODEL SPECIFIC RESULTS
##
## MODEL 1
##
## Model Coefficients
## -------------------------------------------------------------------------------------
##   Predictor    Estimate    SE       Lower    Upper    t      p        Stand. Estimate
## -------------------------------------------------------------------------------------
```

```
##   Intercept      2.091    0.7624    0.582    3.600    2.74     0.007
##   wm            0.483    0.0906    0.304    0.663    5.34    < .001            0.431
##  --------------------------------------------------------------------------------------
##
##
##  MODEL 2
##
##  Model Coefficients
##  -----------------------------------------------------------------------------------------
##   Predictor    Estimate    SE        Lower      Upper    t       p        Stand. Estimate
##  -----------------------------------------------------------------------------------------
##   Intercept     0.8774    1.0066    -1.1151    2.870    0.872    0.385
##   wm           0.4574    0.0946    0.2702    0.645    4.837    < .001            0.4075
##   vocab        0.1340    0.0789    -0.0222    0.290    1.698    0.092            0.1365
##   process      0.0587    0.0892    -0.1178    0.235    0.659    0.511            0.0555
##  -----------------------------------------------------------------------------------------

#simple regression model with wm compared with nested movel adding vocab + process
#simple model is best fit overall
```

6. Visualization

```
# plotting a simple regression model based on:
 # Model 1: intell ~ wm.centered

# create linear model
model.final <- lm(intell ~ wm, data = dat.final)
summary(model.final)

##
## Call:
## lm(formula = intell ~ wm, data = dat.final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.24118 -0.75361 -0.04263  0.79818  2.36475
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.0911     0.7624   2.743  0.00699 **
## wm            0.4834     0.0906   5.335 4.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.064 on 125 degrees of freedom
## Multiple R-squared:  0.1855, Adjusted R-squared:  0.179
## F-statistic: 28.47 on 1 and 125 DF,  p-value: 4.319e-07

model_p <- ggpredict(model.final, full.data = TRUE, pretty = TRUE) #for multiple regression,
add terms = c("v1"", "v2", "vn")

# plot predicted line - for multiple regression, change to aes(x, predicted)
plot <- ggplot(model.final, aes(y = intell, x = wm)) +
    geom_smooth(method = "lm", se = TRUE, fullrange = TRUE) + scale_x_continuous(limits
= c(5, 10.2)) +
    scale_y_continuous(limits = c(0, 9)) + xlab("Working Memory Score") + ggtitle("Plot of
Model of Working Memory Predicting Intelligence") + ylab("Intelligence") + geom_point() +
theme_minimal()

plot
```


Plot of Model of Working Memory Predicting Intelligenc
7.5
5.0
2.5
0.0
Intelligence
5
6
7
8
9
10
Working Memory Score