PSYCH308D - Data Analysis (DA03)

Brady C. Jackson

2025/04/19

Contents

---

1 Libraries

Load all requisite libraries here.

```r
# Load packages. Set messages and warnings to FALSE so I don't have to see the
# masking messages in the output.
library(jmv)        # for descriptive
library(ggplot2)
library(dplyr)
library(corrplot)    # For fancy covariance matrix plots
```

```r
library(apaTables)   # For Word formatted tables
library(car)         # for ncvTest (Breusch Pagan)
library(tidyverse)
library(jmv)         # for descriptives
library(ggplot2)
library(dplyr)
library(psych)
library(corrplot)    # For fancy covariance matrix plots
library(car)         # for ncvTest (Breusch Pagan)
library(stringr)     # for sub_str operations
library(Hmisc)       # for fun.dat substitution
library(see)         # for outliers analysis
library(magrittr)
library(foreign)
library(broom)
library(robmed)
library(mediation)   # For mediation analysis
library(multilevel)
library(GGally)
library(lsr)
library(car)
library(mvnTest)     # Multivariate Normality
library(lm.beta)
library(lavaan)      # Structural Equation Modeling
library(haven)
library(foreign)
library(parallel)
# library(AER)
library(janitor)      # Data cleaning
library(naniar)       # Data cleaning
library(performance)  # Data cleaning
library(mice)         # Data cleaning
```

2 Metadata

This section of code is to setup some general variables that we'll use throughout the code (e.g. figure colors, etc)

```r
# First we'll defines some meta-data to use in all of our plots so they're nice and clean
font_color = "#4F81BD"
grid_color_major = "#9BB7D9"
grid_color_minor = "#C8D7EA"
back_color = "gray95"
rb_colmap = colorRampPalette( c("firebrick", "grey86", "dodgerblue3") )(200)

# I'm going to try to save off my preferred ggplot theme combinations as a unqiue theme object that I c
# later in the code....totally unclear if ggplot works this way....
my_gg_theme = theme_minimal() +
              theme( plot.title = element_text(size = 12, face = "italic", color = font_color),
                     axis.title.x = element_text(color = font_color),
                     axis.title.y = element_text(color = font_color),
                     axis.text.x = element_text(color = font_color),
                     axis.text.y = element_text(color = font_color),
```

```
                        legend.title = element_text(color = font_color),
                        legend.text = element_text(color = font_color),
                        panel.grid.minor = element_line(color = grid_color_minor),
                        panel.grid.major = element_line(color = grid_color_major),
                        panel.background = element_rect(fill = back_color, color = font_color)
                      )
```

---

3 Part 0: Data Cleaning Prep

We're going to do some basic work here so we can get into the line-by-line cleaning tasks in the assignment a bit
smarter

3.1 Load the Data

```
# Load the assignment data from CSV
raw_dat = read.csv("./308D.DA3.Data.csv", na = c("", "NA", "-999", "na", "n/a", "N/A"))

# Rename columns to lower because why not
colnames(raw_dat) <- tolower( colnames(raw_dat) )

# Ensure that the numbers of each subject in the study are unique to prevent any duplicate data
# if the size of the unique-entries only is the same as the whole vector then there are no duplicate su
# NOTE: This fails if the colname of the subject ID is input wrong. So make sure you UPDATE the "test_c
#       entry below
test_colname = "x"

test_unique = ( length( unique( raw_dat[test_colname] ) ) == length(raw_dat[test_colname]))
if(!test_unique){
    print("WARNING: There are duplicate data entries in the raw data")
}else{
    print("No duplicate entries detected in raw data")
}
```

```
## [1] "No duplicate entries detected in raw data"
```

3.2 Name-Mapping

The names of the vars as given suck. We're going to remap them all.

```
# We're going to map the column names as given in the dataframe to a set we prefer.
raw_names <- c("x", "school", "subject", "average.exam.2.grade", "average.exam.1.grade", "school.year",
               "location", "interpersonal")
map_names <- c("idx", "schl_lvl", "sub", "avg_exm_2", "avg_exm_1", "schl_yr",
               "loc", "interp_skls")

# Loop through each raw name and apply the corresponding map_name
dummy_list <- list()
my_dat <- data.frame()
for(iii in 1:length(raw_names) ){

    # Extract paired names
```

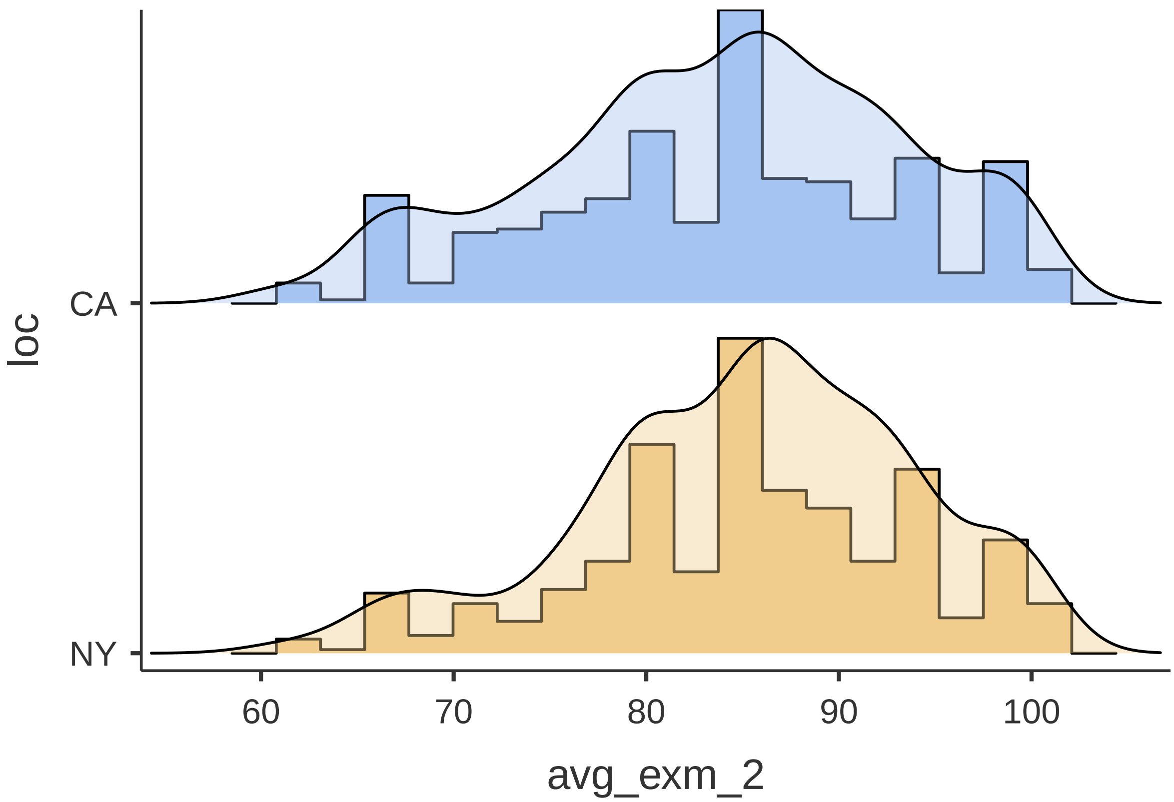
```r
    this_map <- map_names[iii]
    this_raw <- raw_names[iii]

    # Create a column in the new dataframe list named the name from map_names
    # We use a list because R sucks at dynamic binding
    dummy_list[[this_map]] <- raw_dat[[this_raw]]

}

# Convert the list to a dataframe
my_dat <- as.data.frame(dummy_list, stringsAsFactors = FALSE)
```

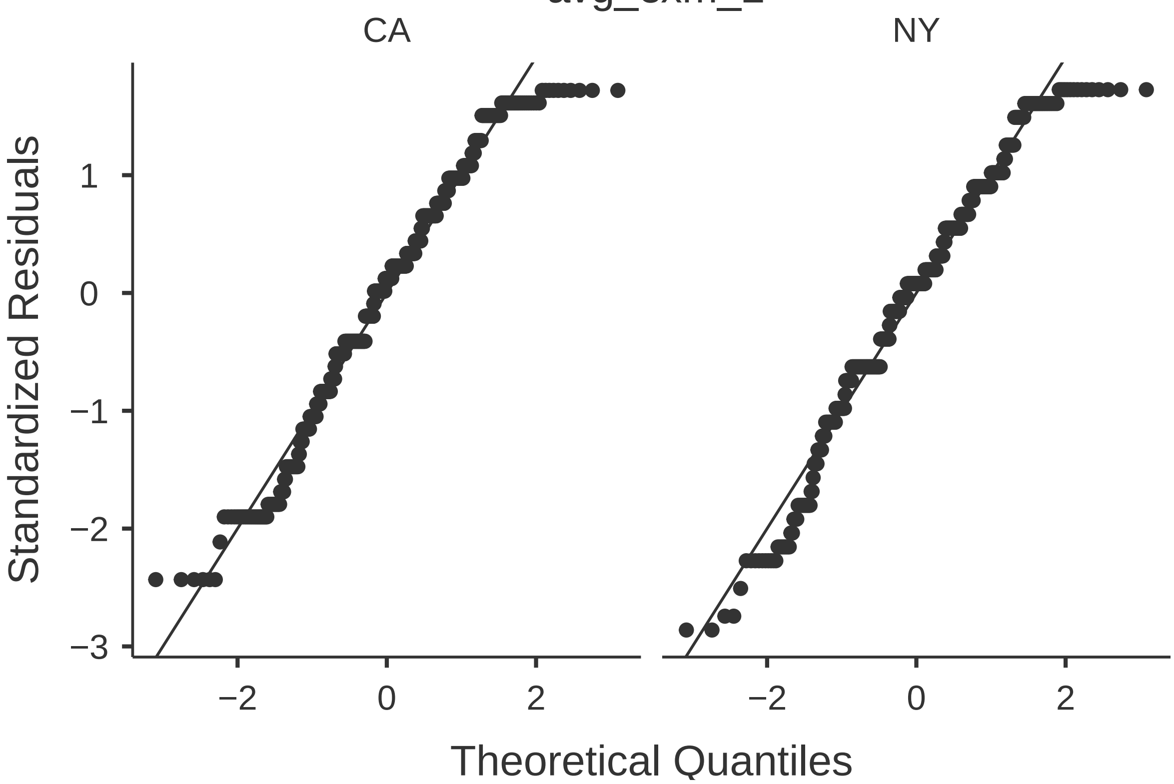
3.3 Descriptives

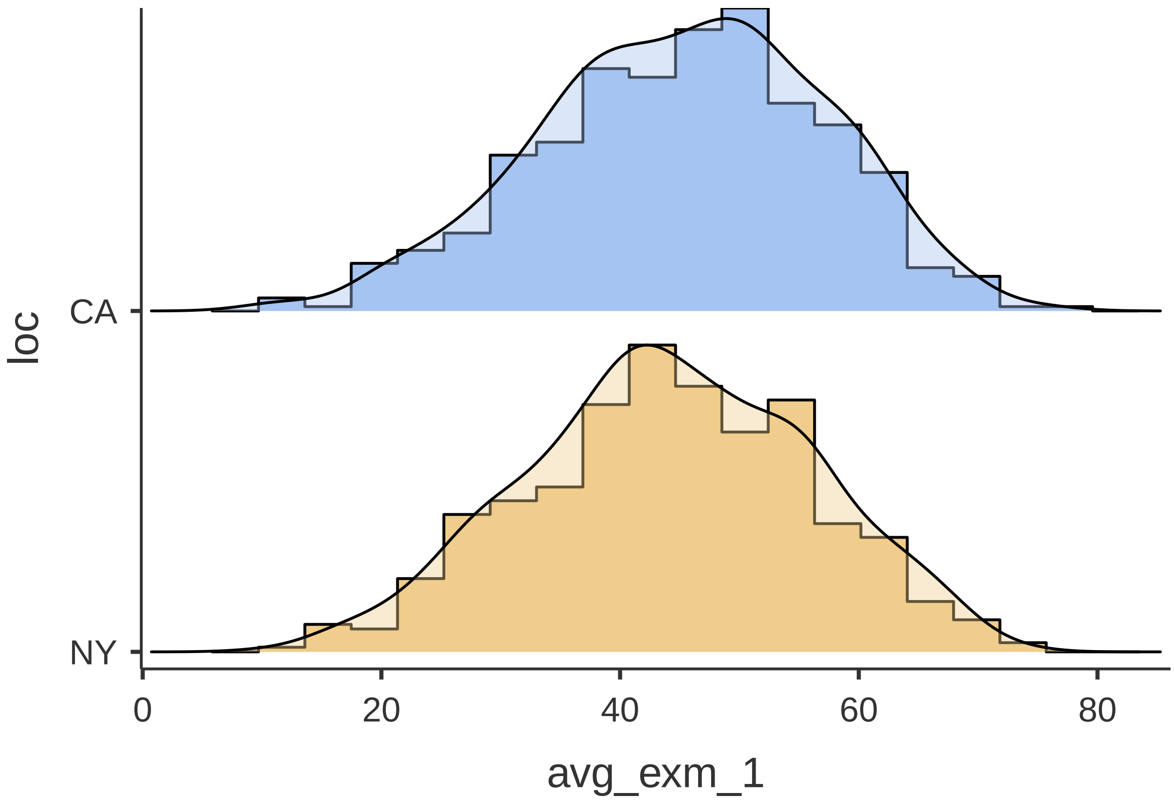
```r
# Names of numeric vars. There are only 3 of them.
cont_names = c("avg_exm_2", "avg_exm_1", "interp_skls")

# We're going to use some split descriptives to help us understand mising values quantities
loc_descr = jmv::descriptives( my_dat,
                               vars = cont_names[],
                               splitBy = "loc",
                               hist = TRUE,
                               dens = TRUE,
                               qq = TRUE,
                               sd = TRUE,
                               variance = TRUE,
                               se = TRUE,
                               missing = TRUE
                             )
print(loc_descr)
```

```
##
##  DESCRIPTIVES
##
##  Descriptives
##
##                           loc    avg_exm_2    avg_exm_1    interp_skls
##
##    N                      CA           510          510            511
##                           NY           486          482            486
##    Missing                CA             1            1              0
##                           NY             0            4              0
##    Mean                   CA      83.85098     45.30392       2.923679
##                           NY      85.32716     44.17427       2.948560
##    Std. error mean        CA     0.4158753    0.5259984     0.03818648
##                           NY     0.3856874    0.5413730     0.03971142
##    Median                 CA      85.00000     46.00000              3
##                           NY      86.00000     44.00000       3.000000
##    Standard deviation     CA      9.391786     11.87872      0.8632173
##                           NY      8.502635     11.88557      0.8754545
##    Variance               CA      88.20565     141.1039      0.7451441
##                           NY      72.29481     141.2669      0.7664206
##    Minimum                CA            61           10              1
```

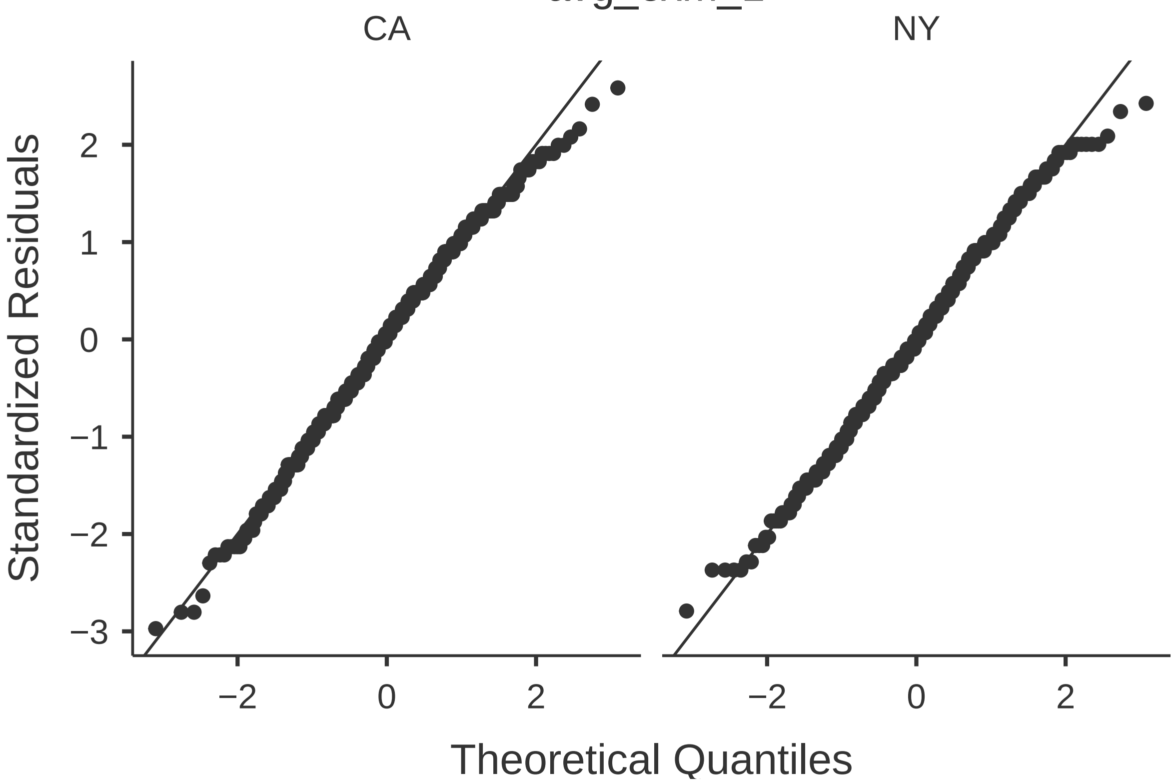
```
##                                NY          61          11              1
##    Maximum                     CA         100          76              7
##                                NY         100          73              7
##
```



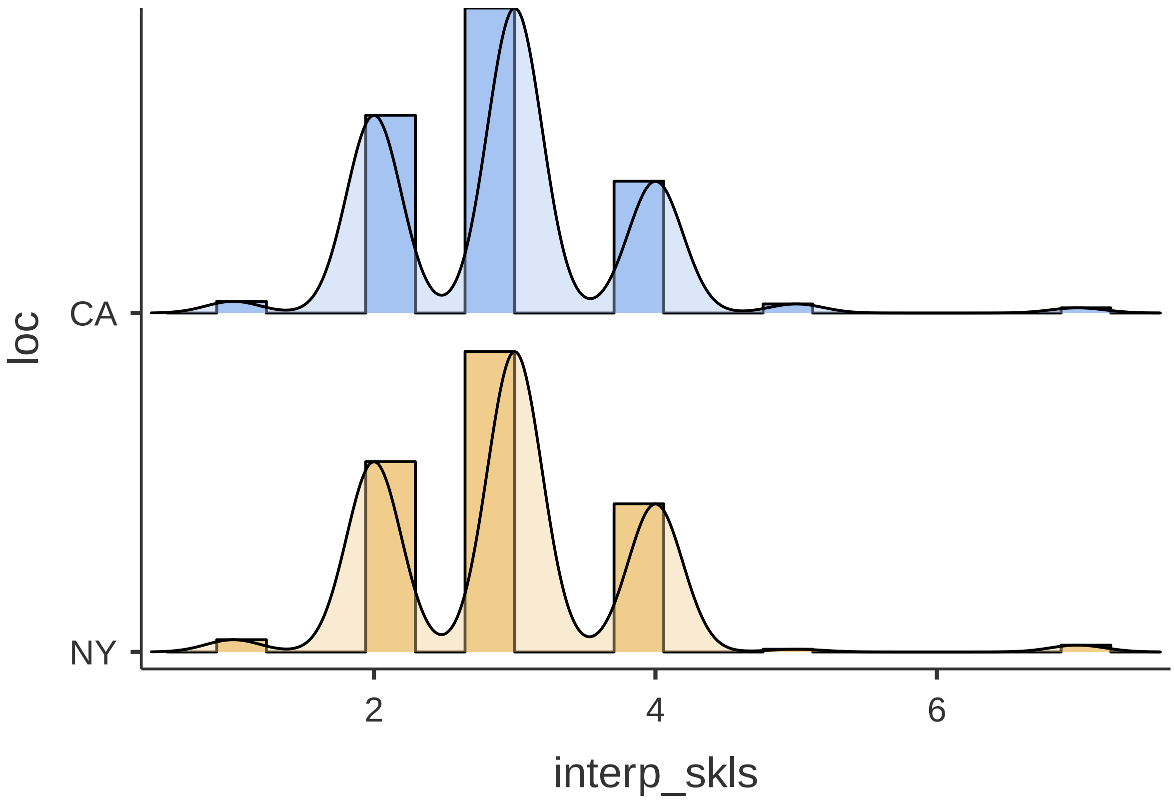


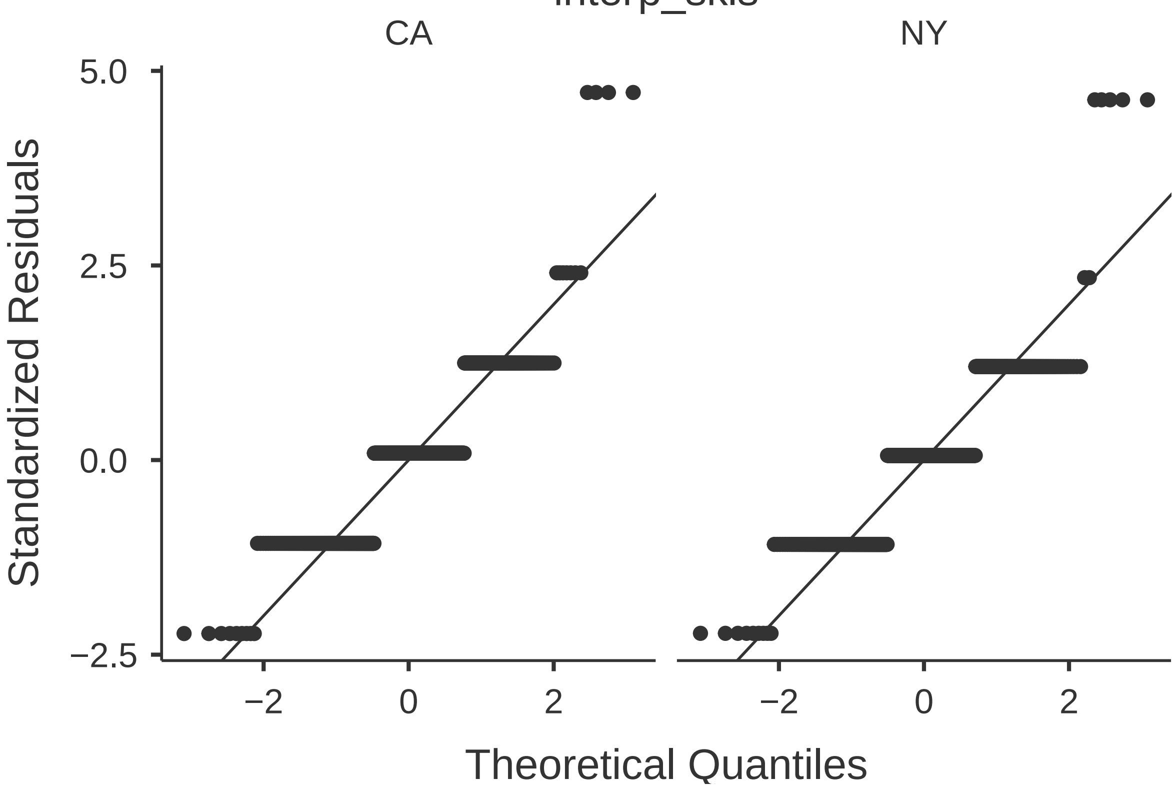


loc
CA
NY
0
20
40
60
80
avg_exm_1



CA
NY
Standardized Residuals
2
1
0
−1
−2
−3
−2
0
2
−2
0
2
Theoretical Quantiles


CA
NY
loc
2
4
6
interp_skls


CA
NY
5.0
2.5
0.0
−2.5
Standardized Residuals
−2
0
2
−2
0
2
Theoretical Quantiles

```
print("    ")
```

```
## [1] "    "
```

```r
print("------")
```

```
## [1] "------"
```

```r
print("    ")
```

```
## [1] "    "
```

```r
sub_descr = jmv::descriptives( my_dat,
                               vars = cont_names[],
                               splitBy = "sub",
                               hist = TRUE,
                               dens = TRUE,
                               qq = TRUE,
                               sd = TRUE,
                               variance = TRUE,
                               se = TRUE,
                               missing = TRUE
                              )
```

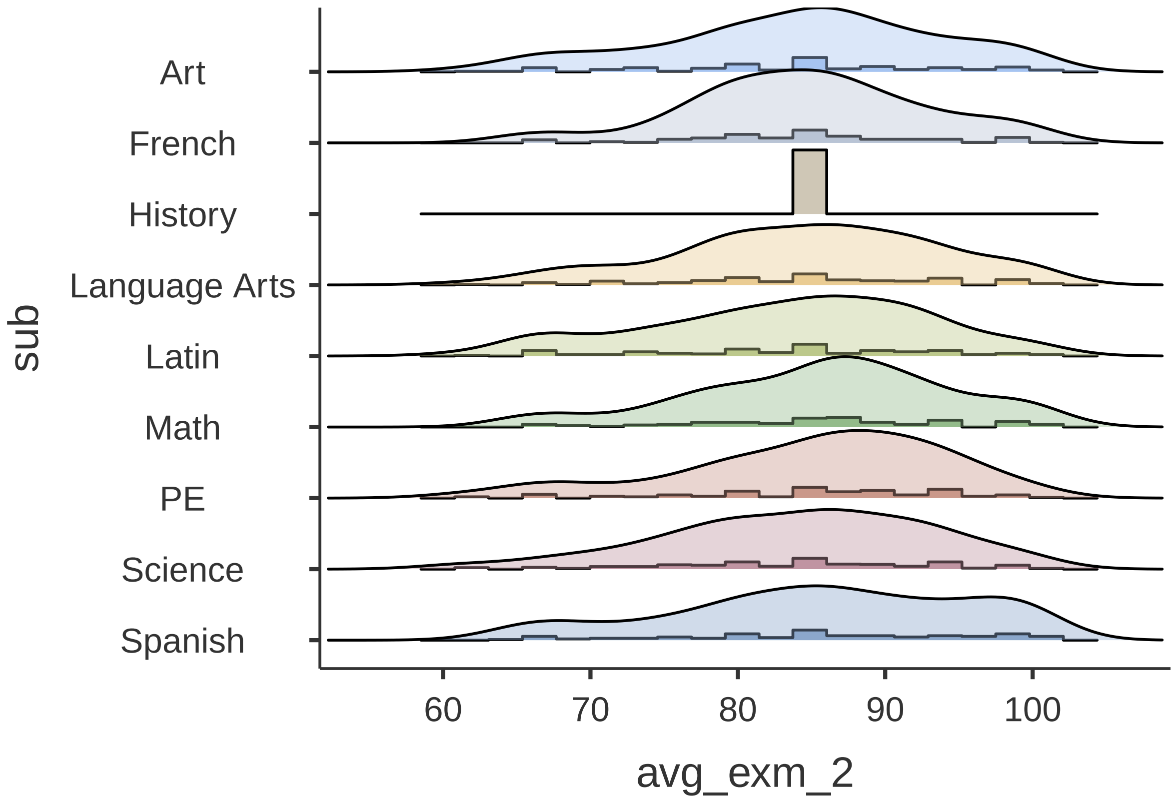
```
## Warning in qt(tCriticalValue, df = stats[["n"]] - 1): NaNs produced
## Warning in qt(tCriticalValue, df = stats[["n"]] - 1): NaNs produced
## Warning in qt(tCriticalValue, df = stats[["n"]] - 1): NaNs produced
```

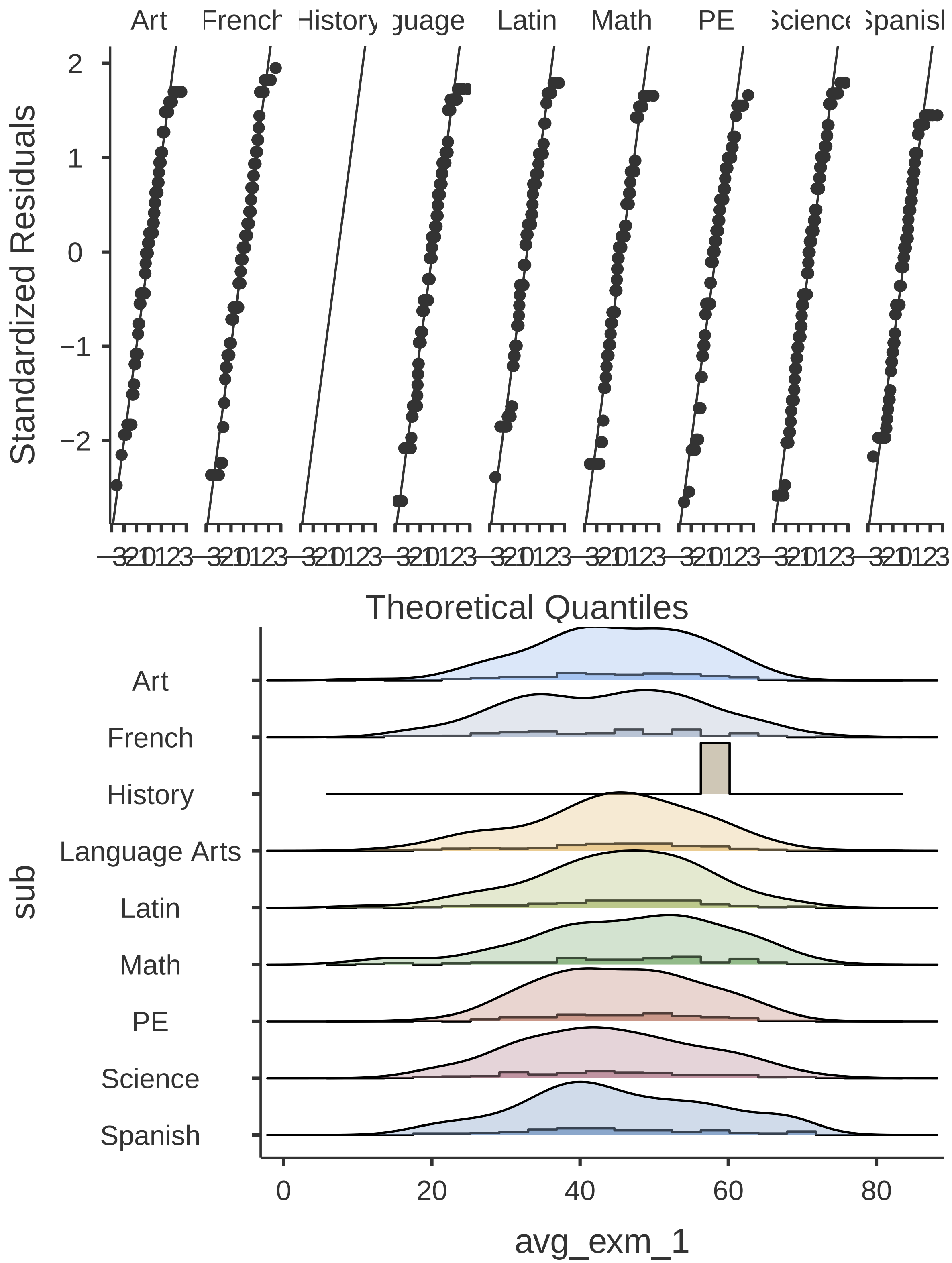
```r
print(sub_descr)
```

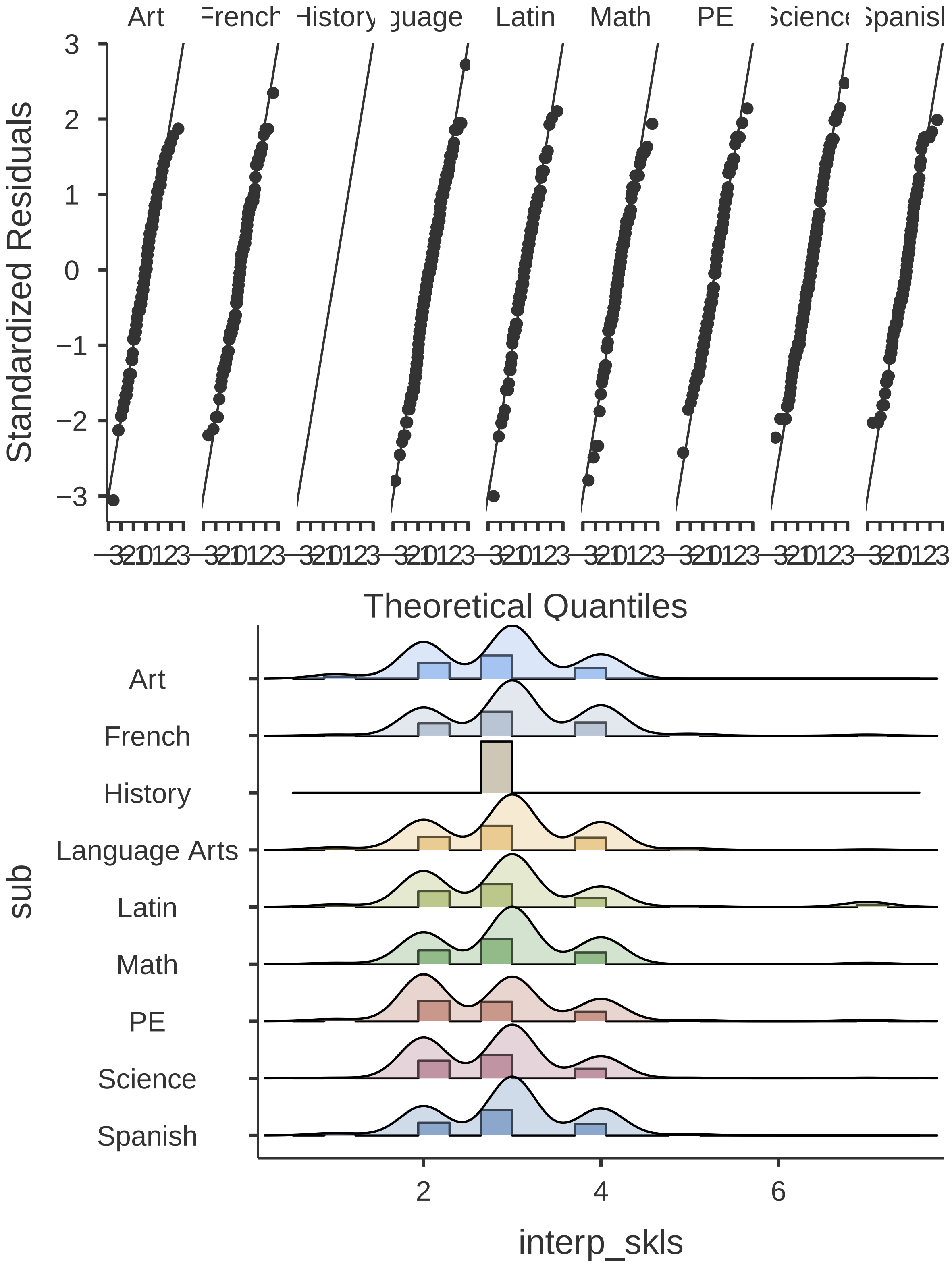
```
##
##  DESCRIPTIVES
##
##  Descriptives
##
##                             sub               avg_exm_2    avg_exm_1    interp_skls
##
##    N                        Art                     107          106            107
##                             French                  105          105            105
##                             History                   1            1              1
##                             Language Arts           215          215            216
##                             Latin                    92           92             92
##                             Math                     93           93             93
##                             PE                      101           99            101
##                             Science                 177          177            177
##                             Spanish                 101          100            101
##    Missing                  Art                       0            1              0
##                             French                    0            0              0
##                             History                   0            0              0
##                             Language Arts             1            1              0
##                             Latin                     0            0              0
##                             Math                      0            0              0
##                             PE                        0            2              0
##                             Science                   0            0              0
##                             Spanish                   0            1              0
##    Mean                     Art                84.12150     44.86792       2.822430
##                             French             84.62857     43.53333       3.076190
##                             History            86.00000     60.00000       3.000000
##                             Language Arts      84.57674     44.44651       2.981481
##                             Latin              83.28261     45.09783       3.021739
```

```
##                          Math            85.56989     46.60215      2.978495
##                          PE              84.97030     45.49495      2.811881
##                          Science         84.01695     43.96610      2.864407
##                          Spanish         85.58416     45.25000      2.960396
##     Std. error mean      Art            0.9042222     1.044222    0.07721694
##                          French         0.7694079     1.225497    0.08526541
##                          History
##                          Language Arts  0.6087903    0.7906091    0.05757180
##                          Latin          0.9734073     1.184292     0.1197009
##                          Math           0.9035277     1.359102    0.08780010
##                          PE             0.8993348     1.056766    0.09088440
##                          Science        0.6697800    0.9116710    0.06069115
##                          Spanish        0.9900762     1.295164    0.07697343
##     Median               Art                   85     45.00000             3
##                          French                85           45             3
##                          History               86           60             3
##                          Language Arts         86           45      3.000000
##                          Latin           85.00000     46.00000      3.000000
##                          Math                  87           48             3
##                          PE                    86           45             3
##                          Science               85           43             3
##                          Spanish               86     43.00000             3
##     Standard deviation   Art             9.353347     10.75093     0.7987382
##                          French          7.884085     12.55761     0.8737105
##                          History
##                          Language Arts   8.926618     11.59260     0.8461292
##                          Latin           9.336595     11.35933      1.148130
##                          Math            8.713305     13.10670     0.8467135
##                          PE              9.038203     10.51469     0.9133769
##                          Science         8.910843     12.12899     0.8074432
##                          Spanish         9.950143     12.95164     0.7735734
##     Variance             Art             87.48510     115.5824     0.6379827
##                          French          62.15879     157.6936     0.7633700
##                          History
##                          Language Arts   79.68450     134.3885     0.7159345
##                          Latin           87.17200     129.0343      1.318204
##                          Math            75.92169     171.7856     0.7169238
##                          PE              81.68911     110.5586     0.8342574
##                          Science         79.40312     147.1125     0.6519646
##                          Spanish         99.00535     167.7449     0.5984158
##     Minimum              Art                   61           12             1
##                          French                66           16             1
##                          History               86           60             3
##                          Language Arts         61           12             1
##                          Latin                 61           11             1
##                          Math                  66           10             1
##                          PE                    61           20             1
##                          Science               61           17             1
##                          Spanish               64           19             1
##     Maximum              Art                  100           65             4
##                          French               100           73             7
##                          History               86           60             3
##                          Language Arts        100           76             7
##                          Latin                100           69             7
```

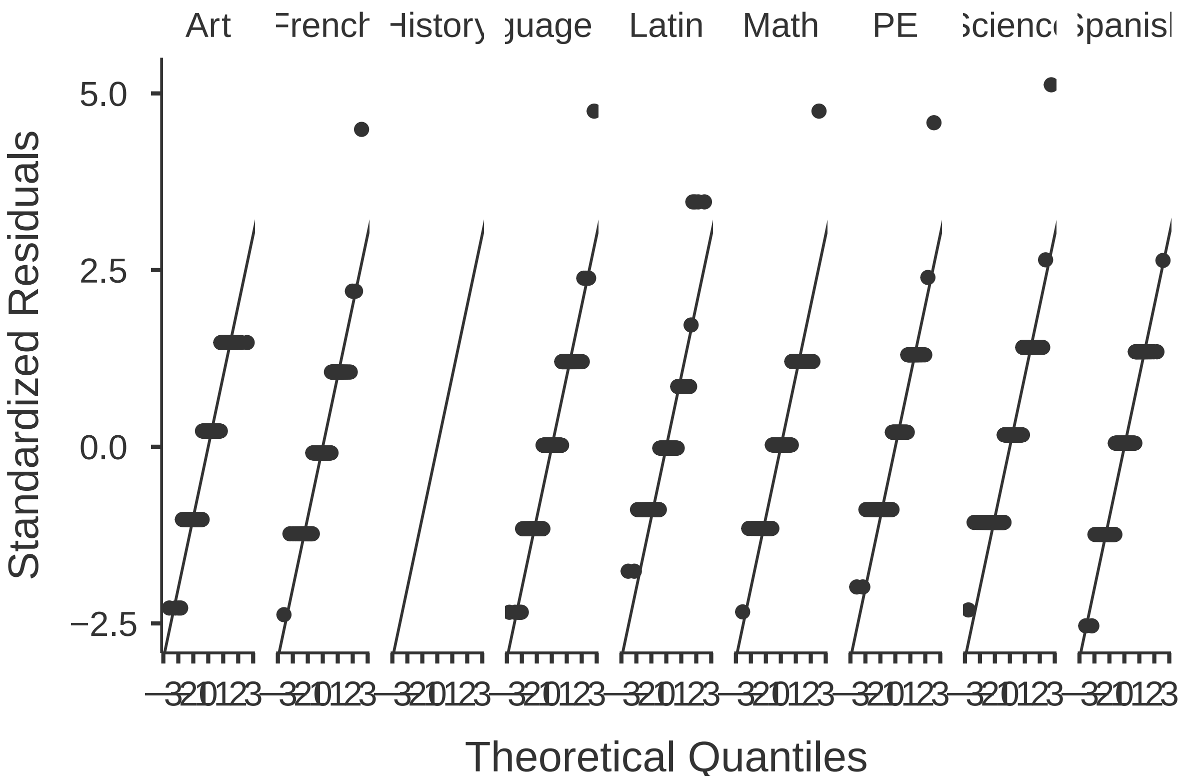
```
##                           Math                    100            72                7
##                           PE                      100            68                7
##                           Science                 100            74                7
##                           Spanish                 100            71                5
##
```




Art
French
History
Language Arts
Latin
Math
PE
Science
Spanish
Standardized Residuals
Theoretical Quantiles
sub
avg_exm_1
0
20
40
60
80



Art
French
History
Language Arts
Latin
Math
PE
Science
Spanish
Standardized Residuals
Theoretical Quantiles
sub
interp_skls




```r
print("   ")
```

```
## [1] "   "
```

```r
print("------")
```

```
## [1] "------"
```

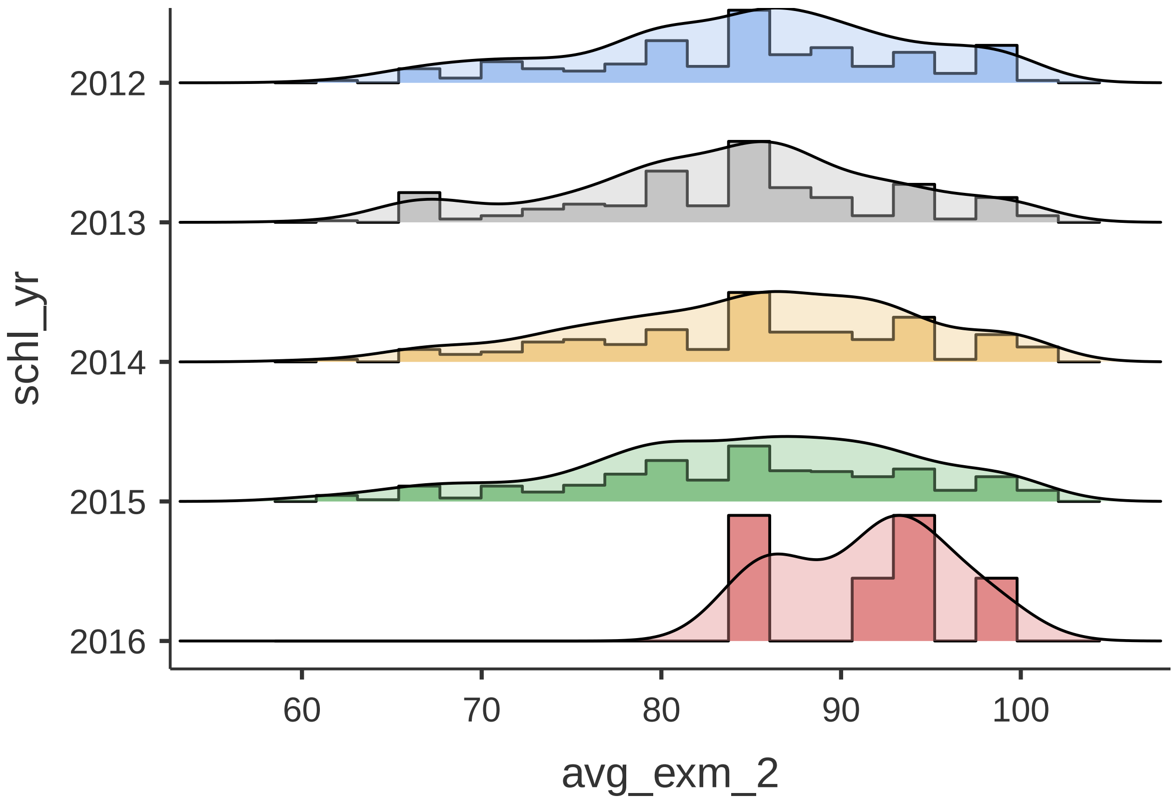
```r
print("    ")
```

```
## [1] "    "
```

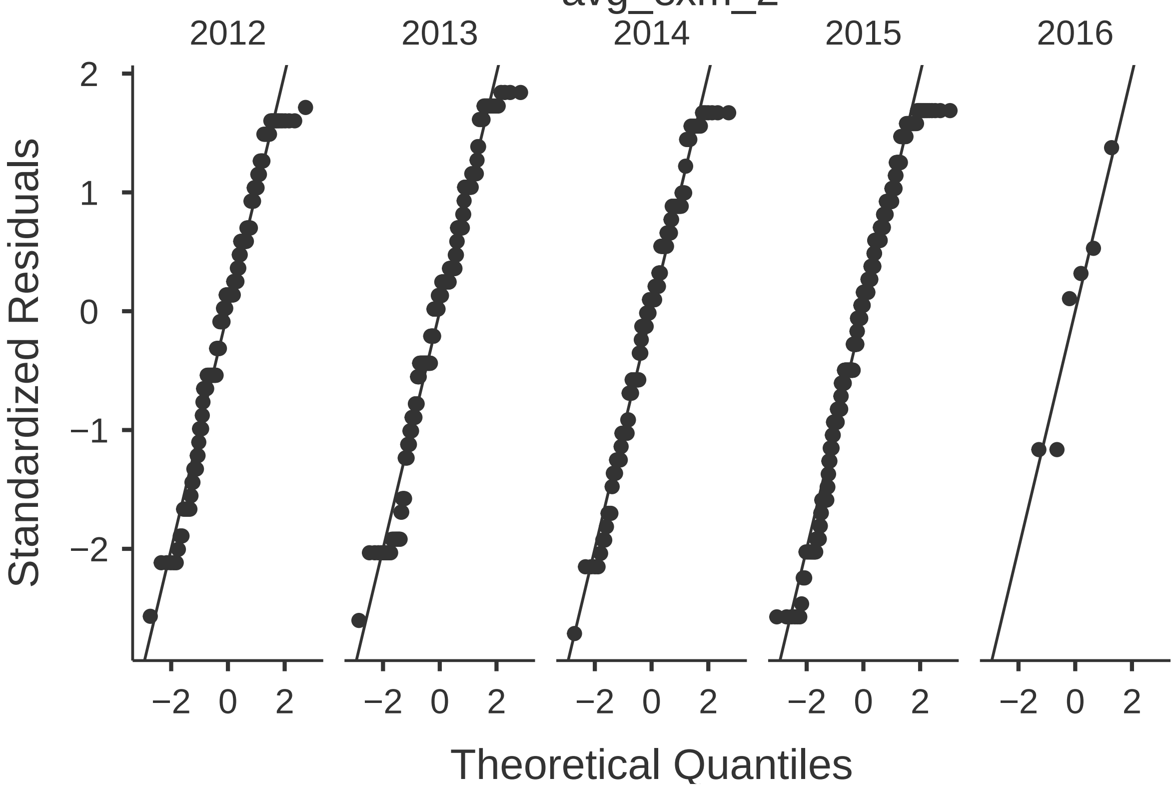
```r
yr_descr = jmv::descriptives( my_dat,
                              vars = cont_names[],
                              splitBy = "schl_yr",
                              hist = TRUE,
                              dens = TRUE,
                              qq = TRUE,
                              sd = TRUE,
                              variance = TRUE,
                              se = TRUE,
                              missing = TRUE
                            )
print(yr_descr)
```

```
##
##  DESCRIPTIVES
##
##  Descriptives
##
##                            schl_yr    avg_exm_2    avg_exm_1    interp_skls
##
```

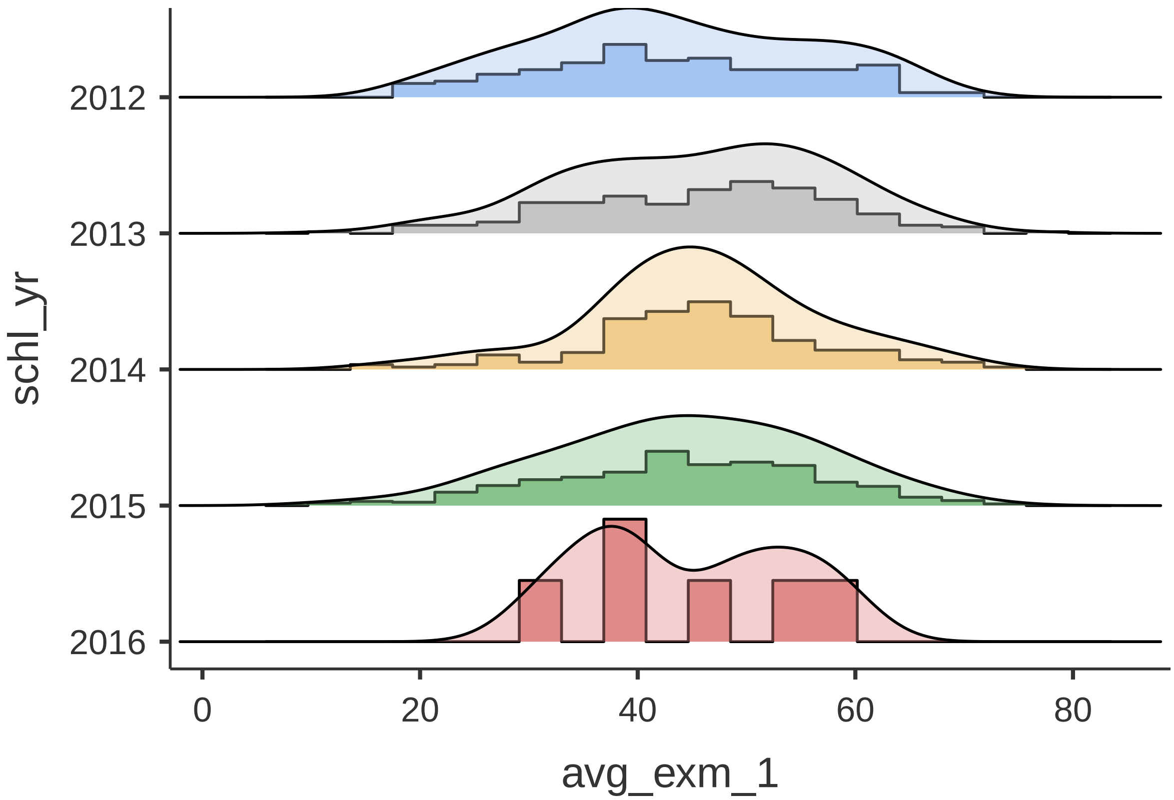
```
##    N                   2012       161        160        161
##                        2013       228        227        228
##                        2014       152        152        152
##                        2015       442        440        443
##                        2016         6          6          6
##    Missing             2012         0          1          0
##                        2013         0          1          0
##                        2014         0          0          0
##                        2015         1          3          0
##                        2016         0          0          0
##    Mean                2012  84.78261   43.31250   3.018634
##                        2013  83.84211   45.85022   3.039474
##                        2014  85.13816   45.98026   2.868421
##                        2015  84.54525   44.33864   2.882619
##                        2016  91.50000   44.66667   2.500000
##    Std. error mean     2012 0.6993062  0.9863928 0.07449209
##                        2013 0.5812578  0.7866784 0.05571253
##                        2014 0.7217462  0.8599403 0.07271811
##                        2015 0.4353860  0.5759222 0.04003870
##                        2016  1.927866   4.063387  0.3415650
##    Median              2012        86   42.00000          3
##                        2013  85.00000         47   3.000000
##                        2014  86.00000   46.00000   3.000000
##                        2015  86.00000   44.00000          3
##                        2016  92.50000   43.50000   2.000000
##    Standard deviation  2012  8.873201   12.47699  0.9451987
##                        2013  8.776801   11.85251  0.8412408
##                        2014  8.898284   10.60206  0.8965291
##                        2015  9.153465   12.08065  0.8427172
##                        2016  4.722288   9.953224  0.8366600
##    Variance            2012  78.73370   155.6753  0.8934006
##                        2013  77.03223   140.4819  0.7076861
##                        2014  79.17946   112.4036  0.8037644
##                        2015  83.78593   145.9420  0.7101723
##                        2016  22.30000   99.06667  0.7000000
##    Minimum             2012        62         19          1
##                        2013        61         12          1
##                        2014        61         16          1
##                        2015        61         10          1
##                        2016        86         32          2
##    Maximum             2012       100         69          7
##                        2013       100         76          7
##                        2014       100         72          7
##                        2015       100         74          7
##                        2016        98         58          4
##
```

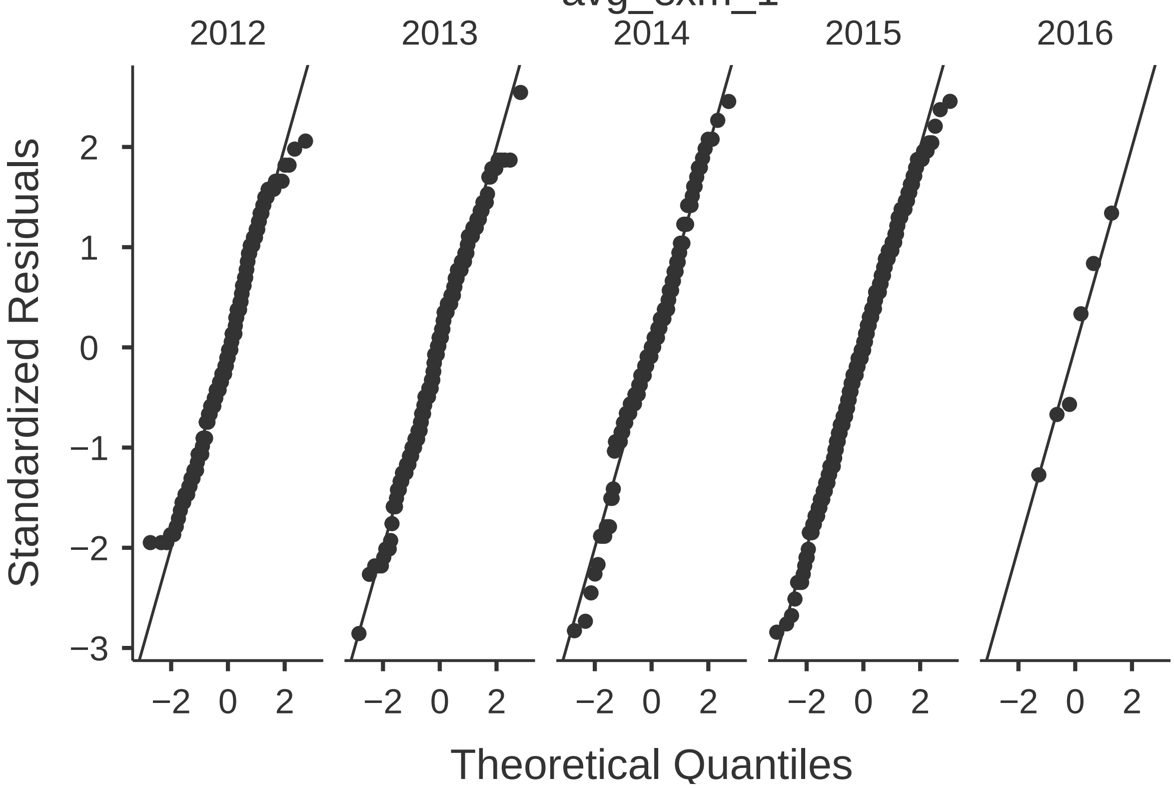


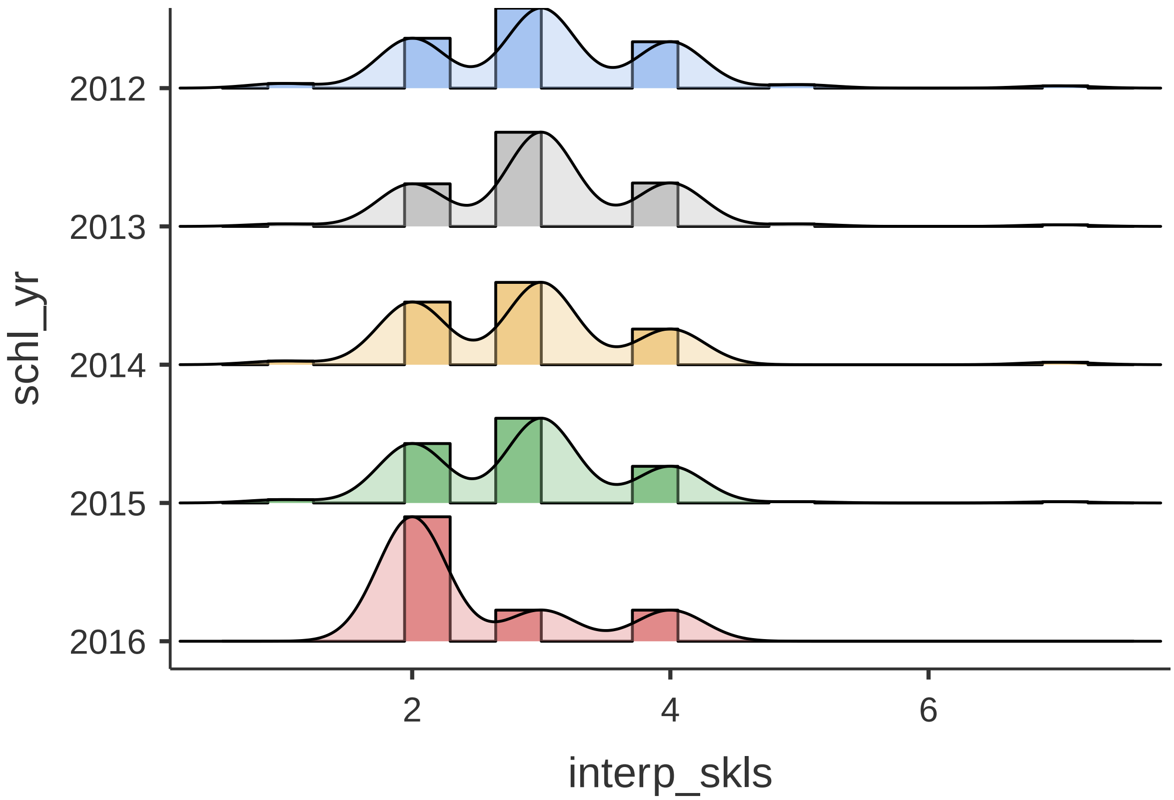
2012
2013
2014
2015
2016
schl_yr
60
70
80
90
100
avg_exm_2

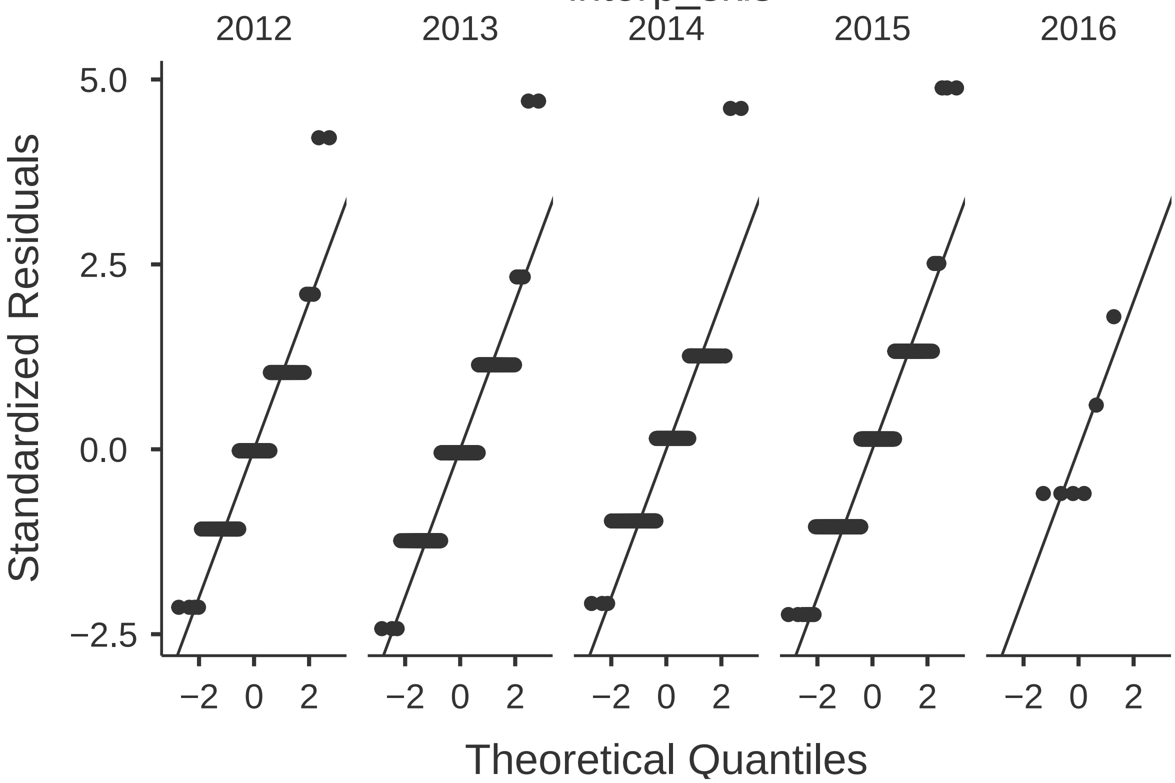


2012
2013
2014
2015
2016
2
1
0
−1
−2
Standardized Residuals
−2 0 2
Theoretical Quantiles


schl_yr
2012
2013
2014
2015
2016
0
20
40
60
80
avg_exm_1


2012
2013
2014
2015
2016
Standardized Residuals
2
1
0
−1
−2
−3
−2 0 2
Theoretical Quantiles


2012
2013
2014
2015
2016
schl_yr
2
4
6
interp_skls


2012
2013
2014
2015
2016
5.0
2.5
0.0
−2.5
Standardized Residuals
−2 0 2
Theoretical Quantiles

```
print("   ")
```

```
## [1] "   "
```

```r
print("------")
```

```
## [1] "------"
```

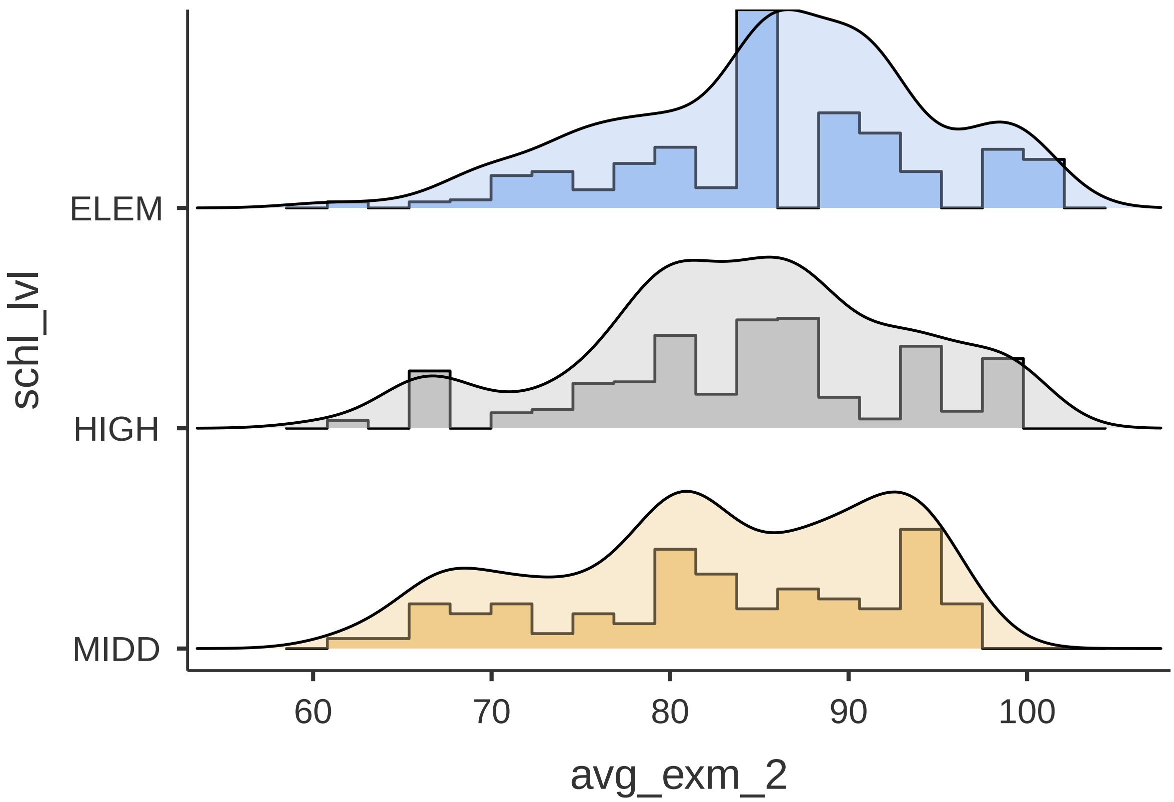
```r
print("    ")
```

```
## [1] "    "
```

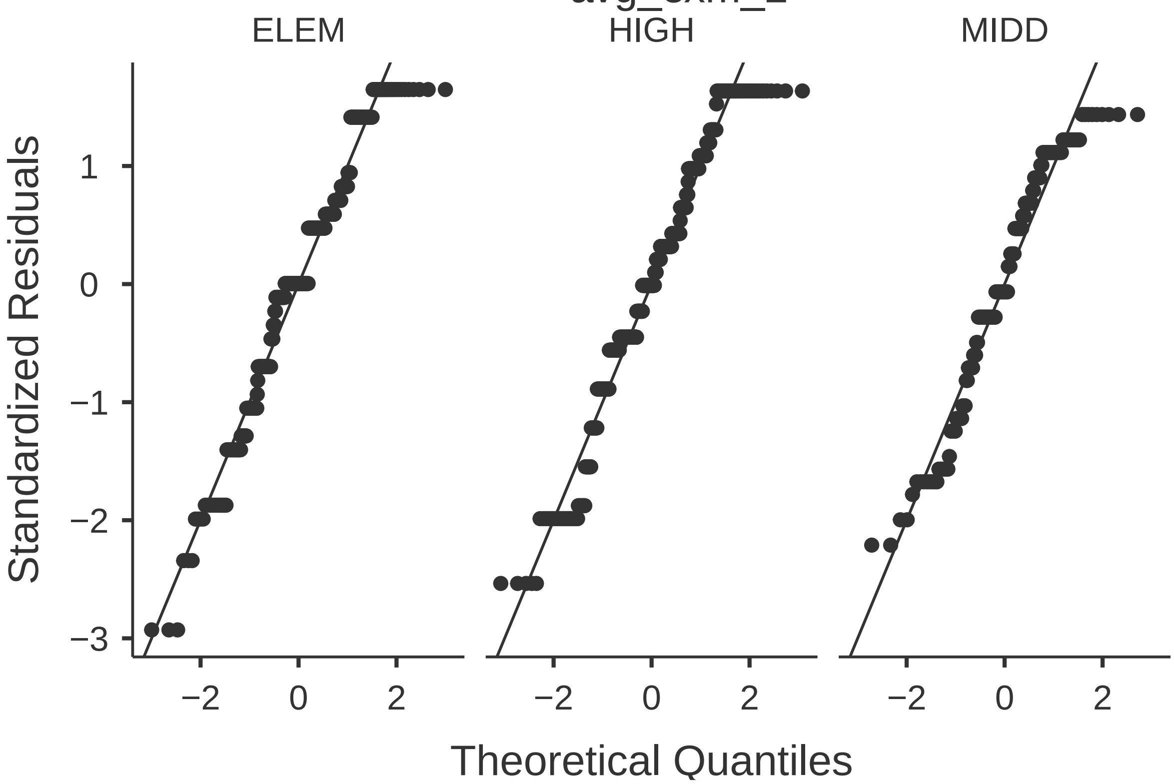
```r
lvl_descr = jmv::descriptives( my_dat,
                               vars = cont_names[],
                               splitBy = "schl_lvl",
                               hist = TRUE,
                               dens = TRUE,
                               qq = TRUE,
                               sd = TRUE,
                               variance = TRUE,
                               se = TRUE,
                               missing = TRUE
                             )
print(lvl_descr)
```

```
##
##  DESCRIPTIVES
##
##  Descriptives
##
##                           schl_lvl    avg_exm_2    avg_exm_1    interp_skls
##
##    N                      ELEM              368          367            368
##                           HIGH              481          478            481
##                           MIDD              150          150            151
##    Missing                ELEM                0            1              0
##                           HIGH                0            3              0
##                           MIDD                1            1              0
##    Mean                   ELEM         85.95924     44.74387       2.978261
##                           HIGH         84.09979     45.20921       2.918919
##                           MIDD         82.61333     43.35333       2.887417
##    Std. error mean        ELEM        0.4442833    0.6245972     0.04680166
##                           HIGH        0.4154585    0.5361304     0.03768221
##                           MIDD        0.7613591    0.9873081     0.07549282
##    Median                 ELEM         86.00000           45       3.000000
##                           HIGH               84     45.50000              3
##                           MIDD         82.00000     42.50000              3
##    Standard deviation     ELEM         8.522832     11.96556      0.8978116
##                           HIGH         9.111715     11.72153      0.8264354
##                           MIDD         9.324707     12.09201      0.9276713
##    Variance               ELEM         72.63866     143.1747      0.8060656
##                           HIGH         83.02335     137.3943      0.6829955
##                           MIDD         86.95016     146.2166      0.8605740
##    Minimum                ELEM               61           10              1
##                           HIGH               61           11              1
##                           MIDD               62           16              1
##    Maximum                ELEM              100           76              7
##                           HIGH               99           70              7
##                           MIDD               96           68              7
```

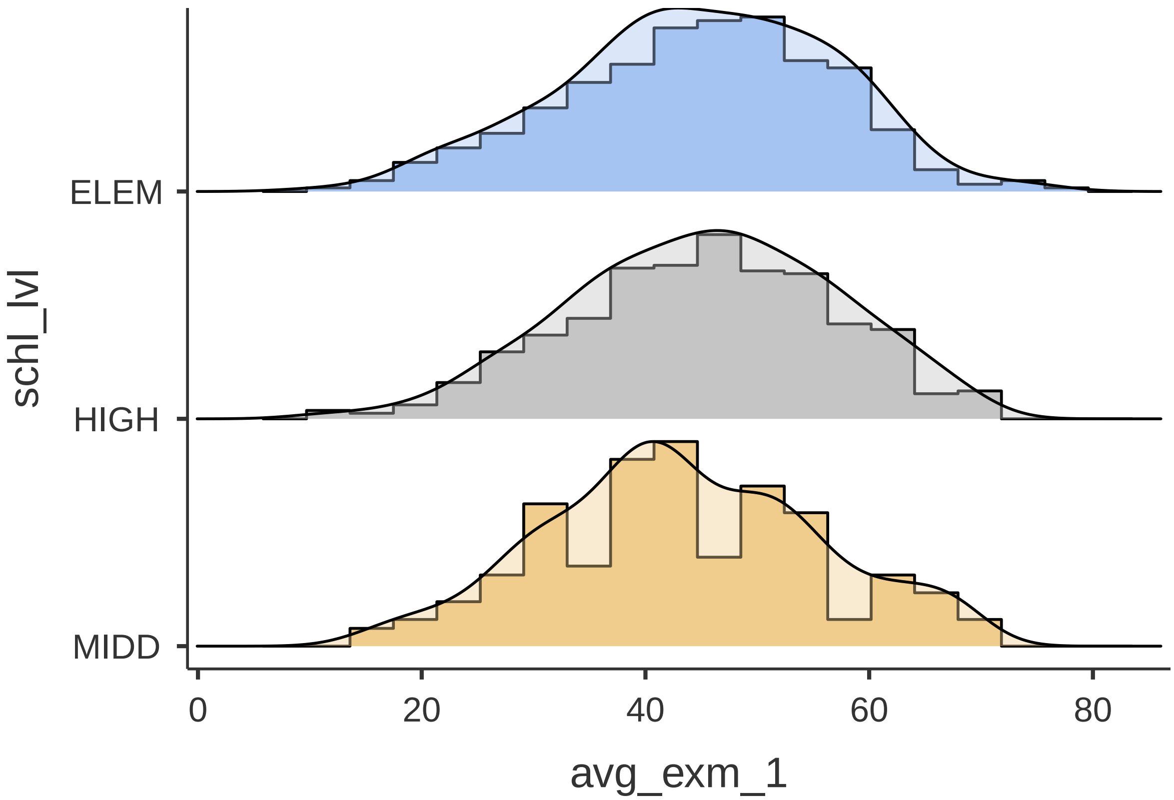
##

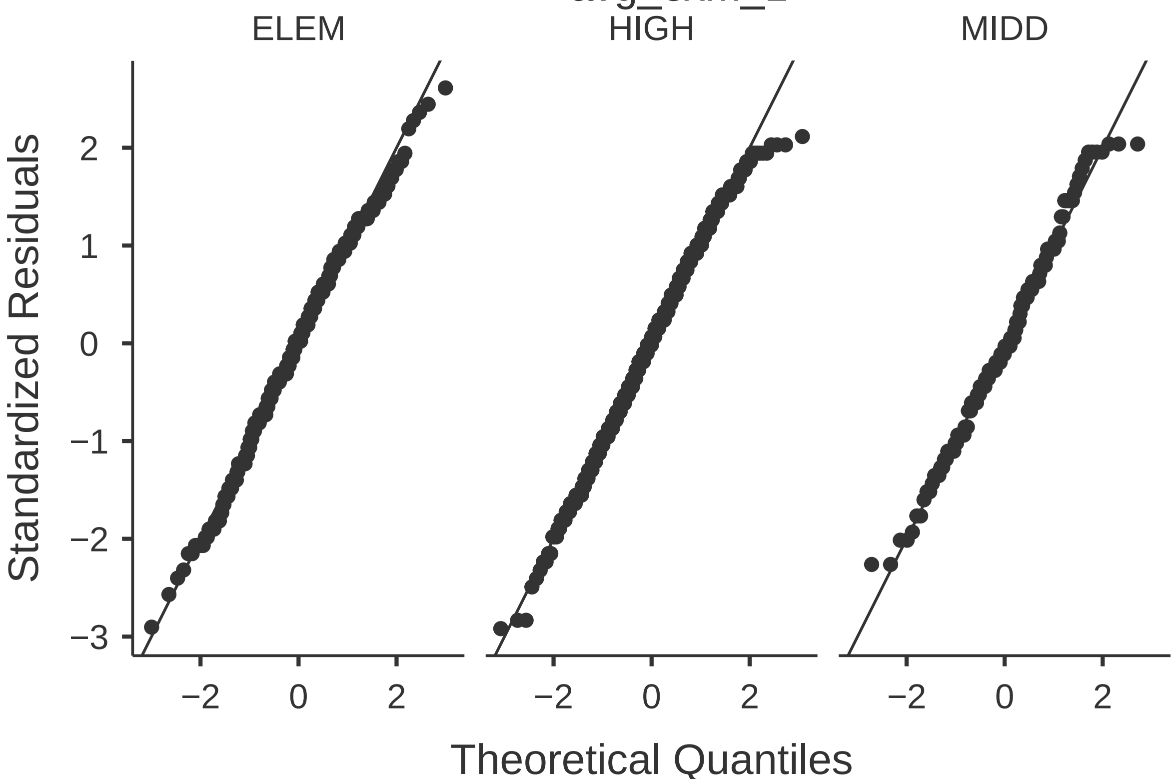


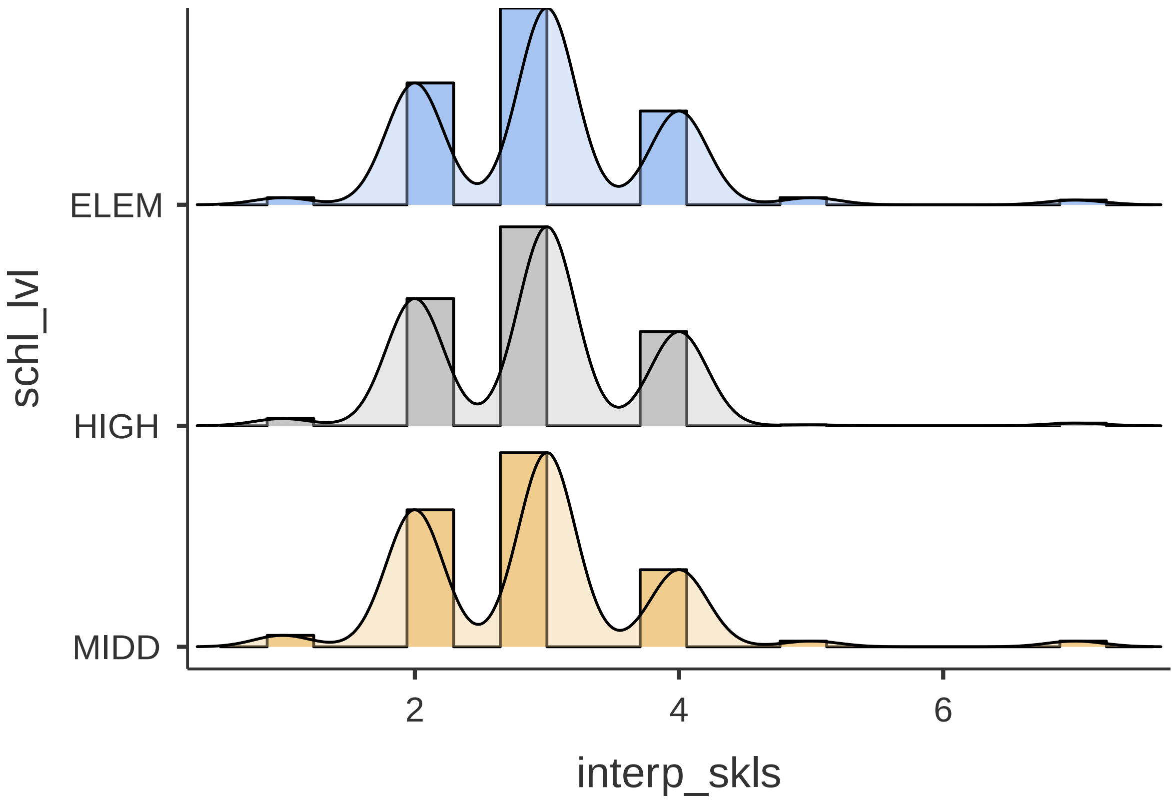
schl_lvl
ELEM
HIGH
MIDD
60
70
80
90
100
avg_exm_2

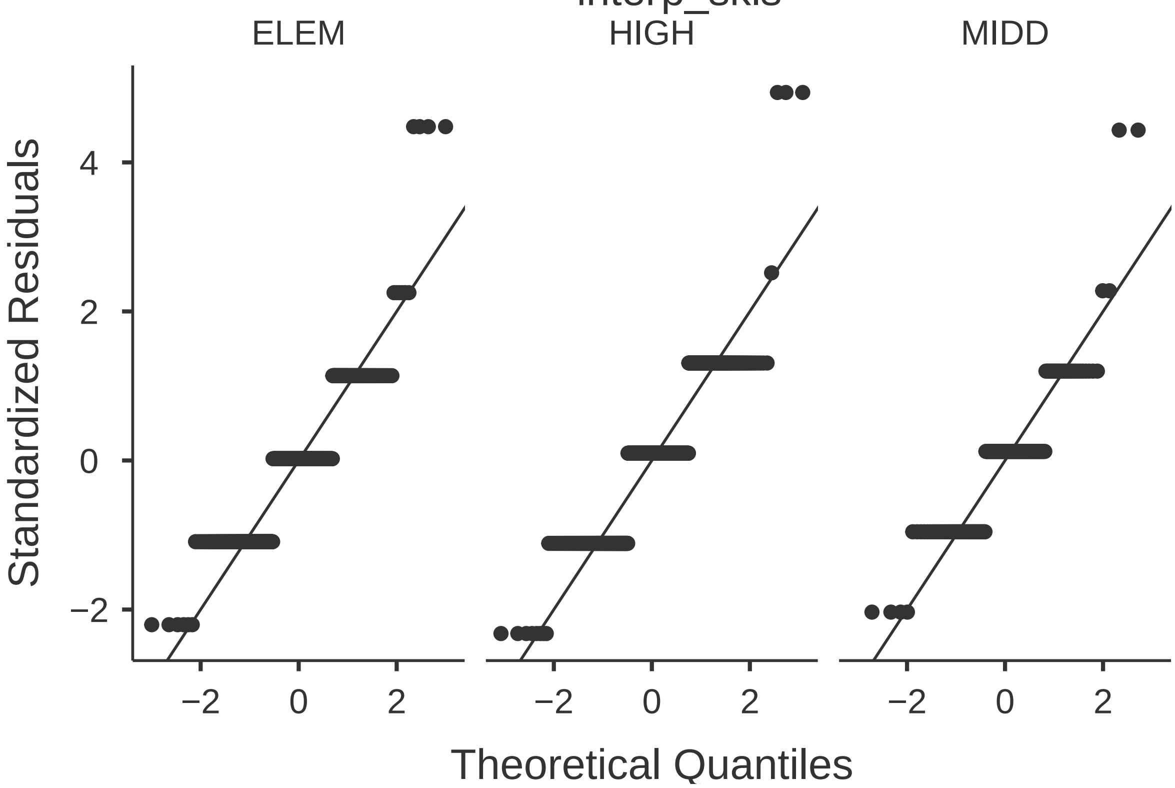


ELEM
HIGH
MIDD
Standardized Residuals
1
0
−1
−2
−3
−2
0
2
Theoretical Quantiles


ELEM
HIGH
MIDD
schl_lvl
0
20
40
60
80
avg_exm_1


ELEM
HIGH
MIDD
Standardized Residuals
2
1
0
−1
−2
−3
−2
0
2
Theoretical Quantiles


ELEM
HIGH
MIDD
schl_lvl
2
4
6
interp_skls


ELEM
HIGH
MIDD
Standardized Residuals
4
2
0
−2
−2
0
2
Theoretical Quantiles

```
print("   ")
```

```
## [1] "   "
```

```r
print("------")
```

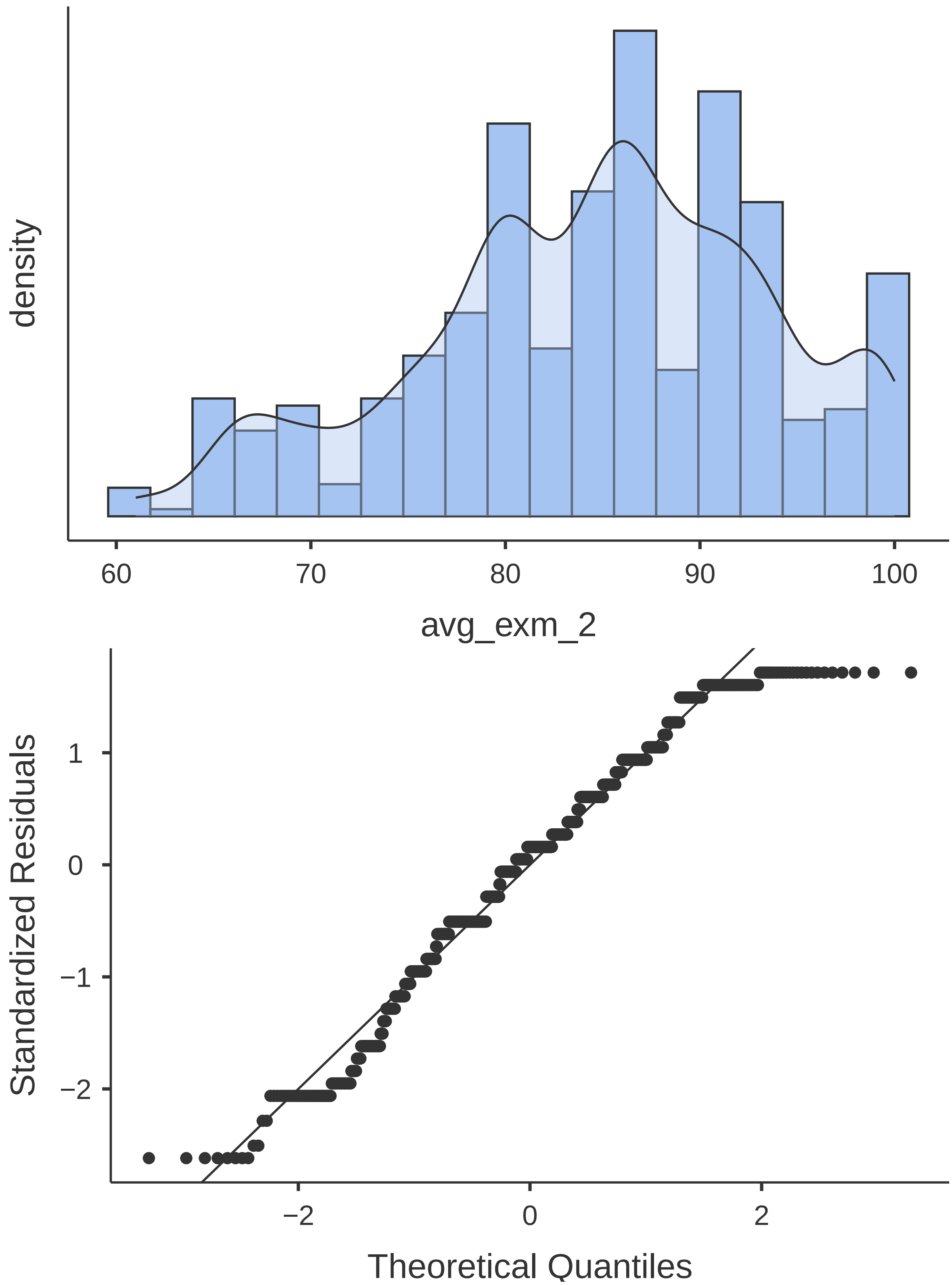
```
## [1] "------"
```

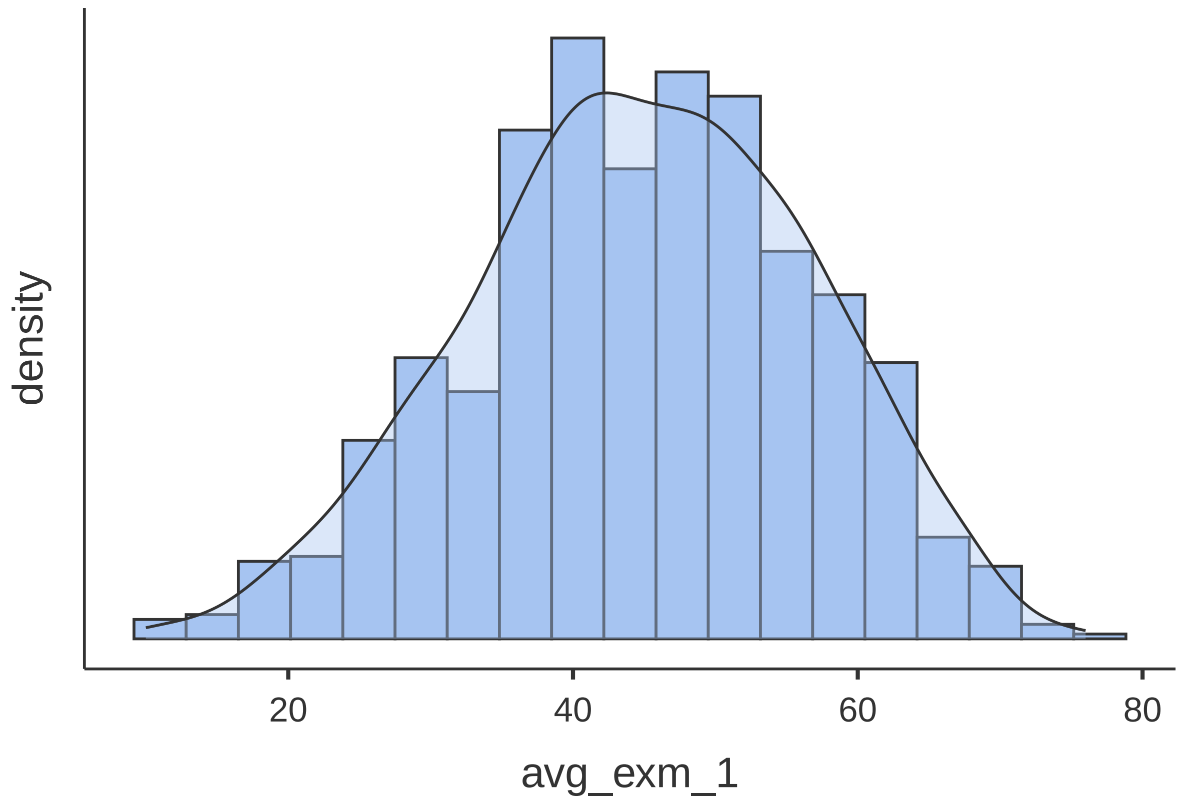
```r
print("    ")
```

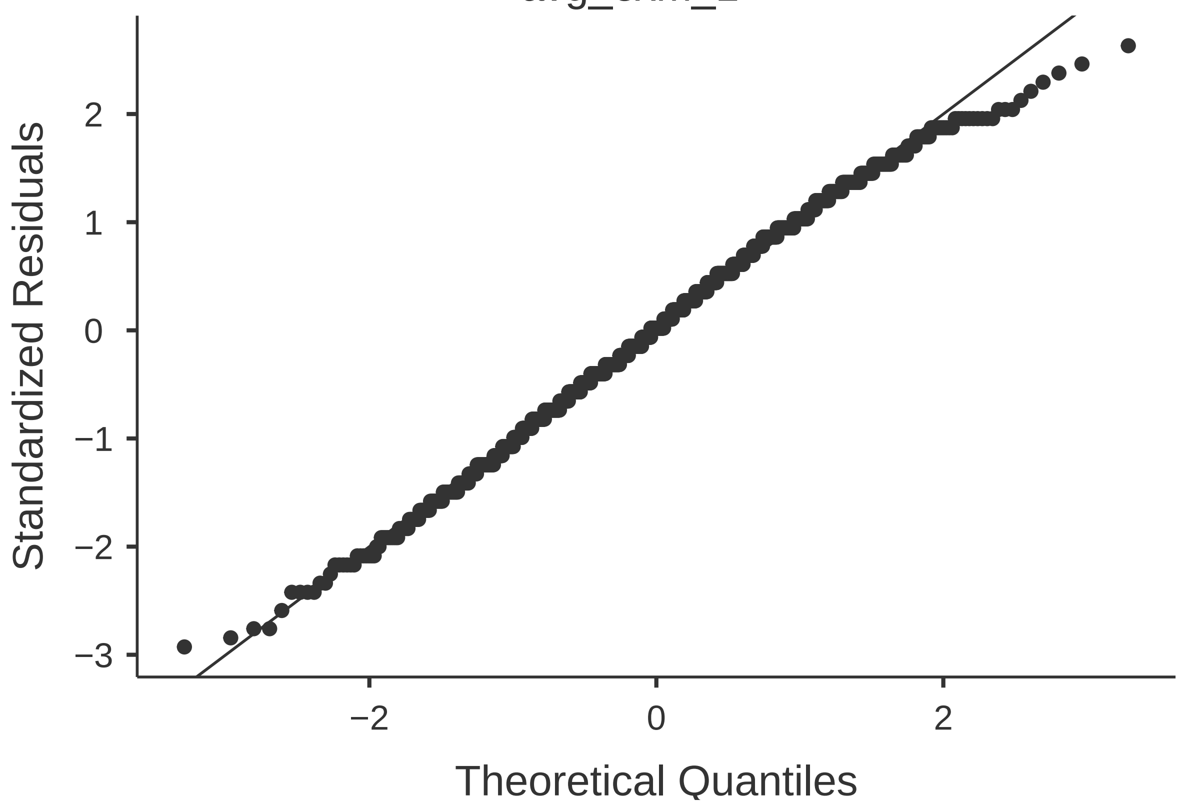
```
## [1] "    "
```

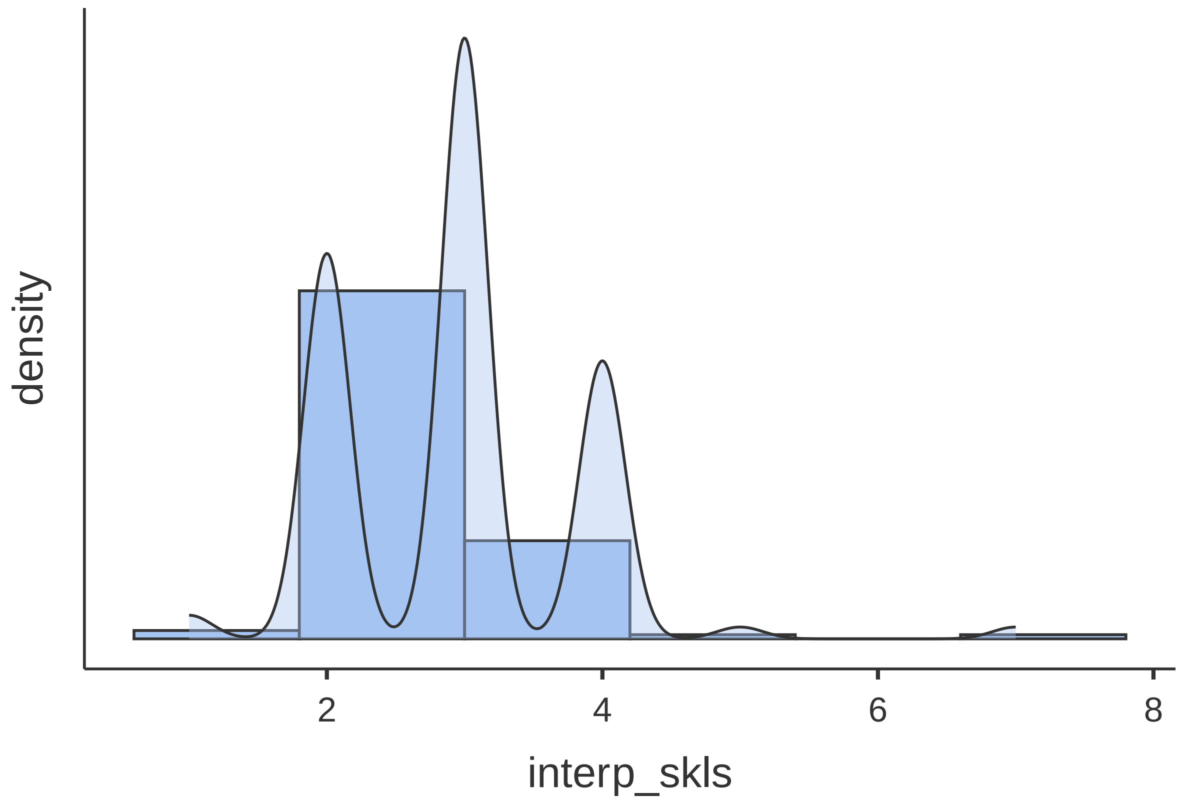
```r
lvl_descr = jmv::descriptives( my_dat,
                               vars = cont_names[],
                               hist = TRUE,
                               dens = TRUE,
                               qq = TRUE,
                               sd = TRUE,
                               variance = TRUE,
                               se = TRUE,
                               missing = TRUE
                             )
print(lvl_descr)
```

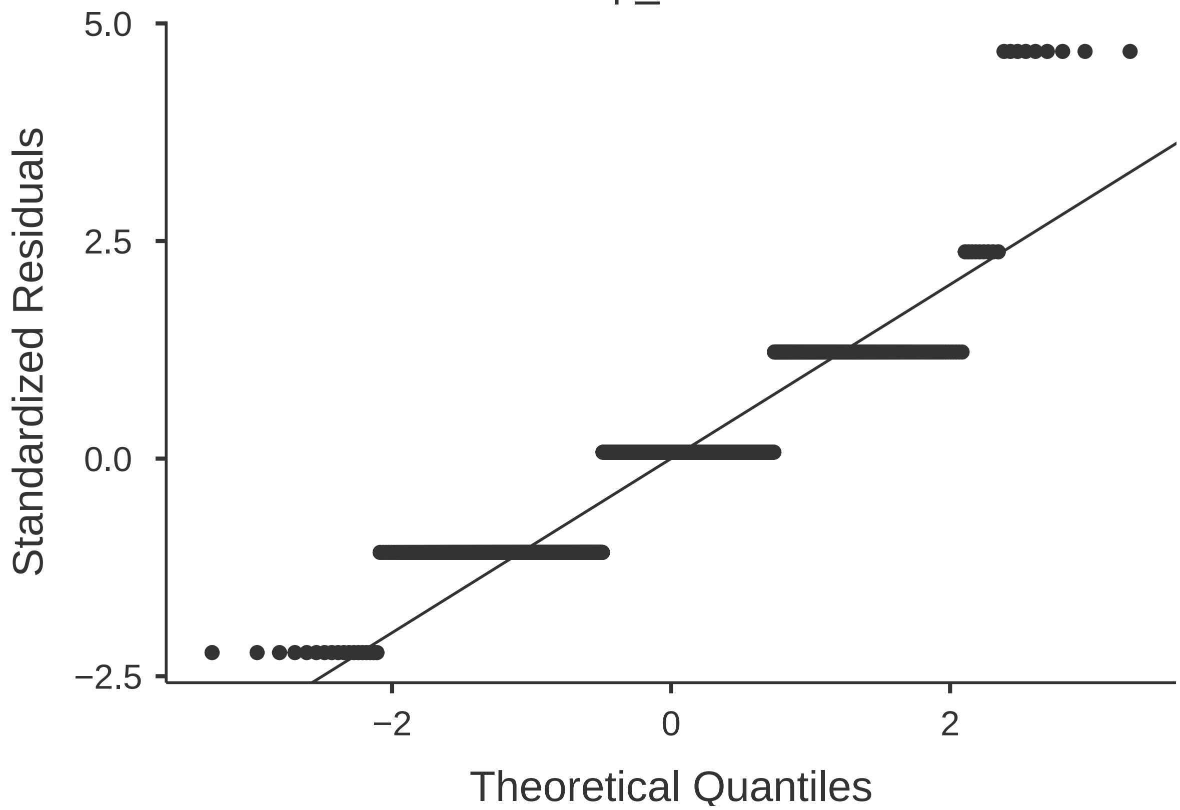
```
##
##  DESCRIPTIVES
##
##  Descriptives
##
##                         avg_exm_2    avg_exm_1    interp_skls
##
##    N                          999          995           1000
##    Missing                      1            5              0
##    Mean                  84.56156     44.75779       2.936000
##    Std. error mean      0.2847790    0.3763944     0.02747105
##    Median                      86           45       3.000000
##    Standard deviation    9.001000     11.87284      0.8687109
##    Variance              81.01800     140.9644      0.7546587
##    Minimum                     61           10              1
##    Maximum                    100           76              7
##
```


density
60
70
80
90
100
avg_exm_2
Standardized Residuals
1
0
−1
−2
−2
0
2
Theoretical Quantiles


density
20
40
60
80
avg_exm_1


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


density
2
4
6
8
interp_skls


5.0
2.5
0.0
−2.5
Standardized Residuals
−2
0
2
Theoretical Quantiles

4 Part 1: Data Cleaning Questions / Tasks

Download the dataset posted on canvas called "308A.DA3.Data.csv" and create an RMarkdown file.
This DA consists of three categories of tasks for you to complete – data cleaning (complete in RStudio), data querying (complete in RStudio and respond to questions below), and a code investigation (respond below).
Upload both a word document with your completed questions and your knitted RMarkdown file in either word or pdf format.

The dataset contains data regarding average grades for Exam 1 and 2 for various classes,
each case is classified by school level (elem, midd, high), subject, year, and location.

4.1 Handle Missing Data for all Variables

Question #1 of Data Cleaning Section. NOTE: We loaded the data with the following code snippet (above):
`raw_dat = read.csv("./308D.DA3.Data.csv", na = c("", "NA", "-999", "na", "n/a, "N/A"))` ….
so all values in raw_dat (and therefore my_dat) should be na if they were missing.

We look for unique values in all non-numeric variables in our dataset, and we check for non-numeric values in our
numeric variables to confirm this worked as expected.

```
# We already confirmed that "x" as loaded had 1000 unique entries in it in our load data section, so we
# handle that column

# Look for unique entries in each of the categorical columns. This will help us understand how creative
# data in each column is. We can spot weird missing data or poorly formatted entries this way.
uni_lvl  <- unique(my_dat$schl_lvl)
uni_sub <- unique(my_dat$sub)
uni_yr   <- unique(my_dat$schl_yr)
uni_loc  <- unique(my_dat$loc)

# Print unique values to see if we have any permutations like CA, ca, and CA in loc
print(uni_lvl)
```

```
## [1] "ELEM" "MIDD" "HIGH"
```

```
print(uni_sub)
```

```
##  [1] "Math"          "History"       "Science"       "French"
##  [5] "Language Arts" "Art"           "Spanish"       "PE"
##  [9] "Latin"         NA
```

```
print(uni_yr)
```

```
## [1] 2015 2013 2012 2014   NA 2016
```

```
print(uni_loc)
```

```
## [1] "CA" "NY" NA
```

```
# Output indicates no weird duplicate formats so we don't need to mess with that. Yay!

## EXPLANATION - WHY
# Per our descriptives we have:
#  1x datapoint in the "history" subject, so that should be removed (it's a non-useful level in subject
#  5 missing exam 1 scores and 1 missing exam 2 scores and no missing interpersonal skills data.
#  The largest missing:sample_size ratio we have, if we slice the data by categories, is 2 missing scor
#    for 99 PE samples
```

```r
#
#  So given all that, our power shouldn't be severely damaged if we just drop all rows missing data in
#   (99 isn't a big sample but difference between 99 and 97 isn't much)
my_dat_with_missing <- my_dat
my_dat <- na.omit(my_dat)

# Toss our one dumb history datapoint
my_dat <- my_dat[my_dat$sub != "History", ]

# Print data loss metric
data_loss = 1 - ( length(my_dat$idx) / length(my_dat_with_missing$idx) )
cat( paste("Total Percentage of data lost due to dropping missing: ", 100*data_loss, "%\n\n", sep="") )
```

```
## Total Percentage of data lost due to dropping missing: 2.6%
```

4.2 Convert Categorical Variables

Convert School Level, Subject, Year, and Location to categorical variables

```r
# We convert all categorical variables to factors and then dummy code each

# We convert all 4x to factors
my_dat$schl_lvl <- as.factor(my_dat$schl_lvl)
my_dat$sub      <- as.factor(my_dat$sub)
my_dat$schl_yr  <- as.factor(my_dat$schl_yr)
my_dat$loc      <- as.factor(my_dat$loc)

# ---
# Dummy Code School Level with Elem as the reference
my_dat$lvl_mid_refd_elem  <- as.numeric(my_dat$schl_lvl)
my_dat$lvl_high_refd_elem <- as.numeric(my_dat$schl_lvl)

# Midd
my_dat$lvl_mid_refd_elem[my_dat$schl_lvl == "ELEM"]  <- 0
my_dat$lvl_mid_refd_elem[my_dat$schl_lvl == "MIDD"]  <- 1
my_dat$lvl_mid_refd_elem[my_dat$schl_lvl == "HIGH"]  <- 0

# High
my_dat$lvl_high_refd_elem[my_dat$schl_lvl == "ELEM"]  <- 0
my_dat$lvl_high_refd_elem[my_dat$schl_lvl == "MIDD"]  <- 0
my_dat$lvl_high_refd_elem[my_dat$schl_lvl == "HIGH"]  <- 1

# Move Category to front of DCs
my_dat <- my_dat %>% relocate(schl_lvl, .before=lvl_mid_refd_elem)

# ---
# Dummy Code subject with math as the reference - Leave history out since we dropped it
# my_dat$sub_hist_refd_math  <- as.numeric(my_dat$sub)
my_dat$sub_sci_refd_math   <- as.numeric(my_dat$sub)
my_dat$sub_frnch_refd_math <- as.numeric(my_dat$sub)
my_dat$sub_lang_refd_math  <- as.numeric(my_dat$sub)
my_dat$sub_art_refd_math   <- as.numeric(my_dat$sub)
my_dat$sub_span_refd_math  <- as.numeric(my_dat$sub)
my_dat$sub_pe_refd_math    <- as.numeric(my_dat$sub)
```

```r
my_dat$sub_latn_refd_math  <- as.numeric(my_dat$sub)

# # History
# my_dat$sub_hist_refd_math[my_dat$sub == "Math"]          <- 0
# my_dat$sub_hist_refd_math[my_dat$sub == "History"]       <- 1
# my_dat$sub_hist_refd_math[my_dat$sub == "Science"]       <- 0
# my_dat$sub_hist_refd_math[my_dat$sub == "French"]        <- 0
# my_dat$sub_hist_refd_math[my_dat$sub == "Language Arts"] <- 0
# my_dat$sub_hist_refd_math[my_dat$sub == "Art"]           <- 0
# my_dat$sub_hist_refd_math[my_dat$sub == "Spanish"]       <- 0
# my_dat$sub_hist_refd_math[my_dat$sub == "PE"]            <- 0
# my_dat$sub_hist_refd_math[my_dat$sub == "Latin"]         <- 0
# my_dat$sub_hist_refd_math[is.na(my_dat$sub)]            <- NA

# Science
my_dat$sub_sci_refd_math[my_dat$sub == "Math"]          <- 0
my_dat$sub_sci_refd_math[my_dat$sub == "History"]       <- 0
my_dat$sub_sci_refd_math[my_dat$sub == "Science"]       <- 1
my_dat$sub_sci_refd_math[my_dat$sub == "French"]        <- 0
my_dat$sub_sci_refd_math[my_dat$sub == "Language Arts"] <- 0
my_dat$sub_sci_refd_math[my_dat$sub == "Art"]           <- 0
my_dat$sub_sci_refd_math[my_dat$sub == "Spanish"]       <- 0
my_dat$sub_sci_refd_math[my_dat$sub == "PE"]            <- 0
my_dat$sub_sci_refd_math[my_dat$sub == "Latin"]         <- 0
my_dat$sub_sci_refd_math[is.na(my_dat$sub)]            <- NA

# French
my_dat$sub_frnch_refd_math[my_dat$sub == "Math"]          <- 0
my_dat$sub_frnch_refd_math[my_dat$sub == "History"]       <- 0
my_dat$sub_frnch_refd_math[my_dat$sub == "Science"]       <- 0
my_dat$sub_frnch_refd_math[my_dat$sub == "French"]        <- 1
my_dat$sub_frnch_refd_math[my_dat$sub == "Language Arts"] <- 0
my_dat$sub_frnch_refd_math[my_dat$sub == "Art"]           <- 0
my_dat$sub_frnch_refd_math[my_dat$sub == "Spanish"]       <- 0
my_dat$sub_frnch_refd_math[my_dat$sub == "PE"]            <- 0
my_dat$sub_frnch_refd_math[my_dat$sub == "Latin"]         <- 0
my_dat$sub_frnch_refd_math[is.na(my_dat$sub)]            <- NA

# Language Arts
my_dat$sub_lang_refd_math[my_dat$sub == "Math"]          <- 0
my_dat$sub_lang_refd_math[my_dat$sub == "History"]       <- 0
my_dat$sub_lang_refd_math[my_dat$sub == "Science"]       <- 0
my_dat$sub_lang_refd_math[my_dat$sub == "French"]        <- 0
my_dat$sub_lang_refd_math[my_dat$sub == "Language Arts"] <- 1
my_dat$sub_lang_refd_math[my_dat$sub == "Art"]           <- 0
my_dat$sub_lang_refd_math[my_dat$sub == "Spanish"]       <- 0
my_dat$sub_lang_refd_math[my_dat$sub == "PE"]            <- 0
my_dat$sub_lang_refd_math[my_dat$sub == "Latin"]         <- 0
my_dat$sub_lang_refd_math[is.na(my_dat$sub)]            <- NA

# Art
my_dat$sub_art_refd_math[my_dat$sub == "Math"]          <- 0
my_dat$sub_art_refd_math[my_dat$sub == "History"]       <- 0
```

```r
my_dat$sub_art_refd_math[my_dat$sub == "Science"]       <- 0
my_dat$sub_art_refd_math[my_dat$sub == "French"]        <- 0
my_dat$sub_art_refd_math[my_dat$sub == "Language Arts"] <- 0
my_dat$sub_art_refd_math[my_dat$sub == "Art"]           <- 1
my_dat$sub_art_refd_math[my_dat$sub == "Spanish"]       <- 0
my_dat$sub_art_refd_math[my_dat$sub == "PE"]            <- 0
my_dat$sub_art_refd_math[my_dat$sub == "Latin"]         <- 0
my_dat$sub_art_refd_math[is.na(my_dat$sub)]            <- NA

# Spanish
my_dat$sub_span_refd_math[my_dat$sub == "Math"]          <- 0
my_dat$sub_span_refd_math[my_dat$sub == "History"]       <- 0
my_dat$sub_span_refd_math[my_dat$sub == "Science"]       <- 0
my_dat$sub_span_refd_math[my_dat$sub == "French"]        <- 0
my_dat$sub_span_refd_math[my_dat$sub == "Language Arts"] <- 0
my_dat$sub_span_refd_math[my_dat$sub == "Art"]           <- 0
my_dat$sub_span_refd_math[my_dat$sub == "Spanish"]       <- 1
my_dat$sub_span_refd_math[my_dat$sub == "PE"]            <- 0
my_dat$sub_span_refd_math[my_dat$sub == "Latin"]         <- 0
my_dat$sub_span_refd_math[is.na(my_dat$sub)]            <- NA

# PE
my_dat$sub_pe_refd_math[my_dat$sub == "Math"]          <- 0
my_dat$sub_pe_refd_math[my_dat$sub == "History"]       <- 0
my_dat$sub_pe_refd_math[my_dat$sub == "Science"]       <- 0
my_dat$sub_pe_refd_math[my_dat$sub == "French"]        <- 0
my_dat$sub_pe_refd_math[my_dat$sub == "Language Arts"] <- 0
my_dat$sub_pe_refd_math[my_dat$sub == "Art"]           <- 0
my_dat$sub_pe_refd_math[my_dat$sub == "Spanish"]       <- 0
my_dat$sub_pe_refd_math[my_dat$sub == "PE"]            <- 1
my_dat$sub_pe_refd_math[my_dat$sub == "Latin"]         <- 0
my_dat$sub_pe_refd_math[is.na(my_dat$sub)]            <- NA

# Latin
my_dat$sub_latn_refd_math[my_dat$sub == "Math"]          <- 0
my_dat$sub_latn_refd_math[my_dat$sub == "History"]       <- 1
my_dat$sub_latn_refd_math[my_dat$sub == "Science"]       <- 0
my_dat$sub_latn_refd_math[my_dat$sub == "French"]        <- 0
my_dat$sub_latn_refd_math[my_dat$sub == "Language Arts"] <- 0
my_dat$sub_latn_refd_math[my_dat$sub == "Art"]           <- 0
my_dat$sub_latn_refd_math[my_dat$sub == "Spanish"]       <- 0
my_dat$sub_latn_refd_math[my_dat$sub == "PE"]            <- 0
my_dat$sub_latn_refd_math[my_dat$sub == "Latin"]         <- 1
my_dat$sub_latn_refd_math[is.na(my_dat$sub)]            <- NA

# Move Category to front of DCs
# my_dat <- my_dat %>% relocate(sub, .before=sub_hist_refd_math)
my_dat <- my_dat %>% relocate(sub, .before=sub_sci_refd_math )

# ---
# Dummy Code yr with (shouldn't be a categorical) with 2012 as the reference
my_dat$yr_2013_refd_2012 <- as.numeric(my_dat$schl_yr)
my_dat$yr_2014_refd_2012 <- as.numeric(my_dat$schl_yr)
```

```r
my_dat$yr_2015_refd_2012 <- as.numeric(my_dat$schl_yr)
my_dat$yr_2016_refd_2012 <- as.numeric(my_dat$schl_yr)

# 2013
my_dat$yr_2013_refd_2012[my_dat$schl_yr == "2012"] <- 0
my_dat$yr_2013_refd_2012[my_dat$schl_yr == "2013"] <- 1
my_dat$yr_2013_refd_2012[my_dat$schl_yr == "2014"] <- 0
my_dat$yr_2013_refd_2012[my_dat$schl_yr == "2015"] <- 0
my_dat$yr_2013_refd_2012[my_dat$schl_yr == "2016"] <- 0
my_dat$yr_2013_refd_2012[is.na(my_dat$schl_yr)]   <- NA

# 2014
my_dat$yr_2014_refd_2012[my_dat$schl_yr == "2012"] <- 0
my_dat$yr_2014_refd_2012[my_dat$schl_yr == "2013"] <- 0
my_dat$yr_2014_refd_2012[my_dat$schl_yr == "2014"] <- 1
my_dat$yr_2014_refd_2012[my_dat$schl_yr == "2015"] <- 0
my_dat$yr_2014_refd_2012[my_dat$schl_yr == "2016"] <- 0
my_dat$yr_2014_refd_2012[is.na(my_dat$schl_yr)]   <- NA

# 2015
my_dat$yr_2015_refd_2012[my_dat$schl_yr == "2012"] <- 0
my_dat$yr_2015_refd_2012[my_dat$schl_yr == "2013"] <- 0
my_dat$yr_2015_refd_2012[my_dat$schl_yr == "2014"] <- 0
my_dat$yr_2015_refd_2012[my_dat$schl_yr == "2015"] <- 1
my_dat$yr_2015_refd_2012[my_dat$schl_yr == "2016"] <- 0
my_dat$yr_2015_refd_2012[is.na(my_dat$schl_yr)]   <- NA

# 2016
my_dat$yr_2016_refd_2012[my_dat$schl_yr == "2012"] <- 0
my_dat$yr_2016_refd_2012[my_dat$schl_yr == "2013"] <- 0
my_dat$yr_2016_refd_2012[my_dat$schl_yr == "2014"] <- 0
my_dat$yr_2016_refd_2012[my_dat$schl_yr == "2015"] <- 0
my_dat$yr_2016_refd_2012[my_dat$schl_yr == "2016"] <- 1
my_dat$yr_2016_refd_2012[is.na(my_dat$schl_yr)]   <- NA

# Move Category to front of DCs
my_dat <- my_dat %>% relocate(schl_yr, .before=yr_2013_refd_2012)

# ---
# Dummy Code Location with California as the reference.
my_dat$loc_ny_refd_ca  <- as.numeric(my_dat$loc)

# NY - Handle NA cases
my_dat$loc_ny_refd_ca[my_dat$loc == "CA"] <- 0
my_dat$loc_ny_refd_ca[my_dat$loc == "NY"] <- 1
my_dat$loc_ny_refd_ca[is.na(my_dat$loc)] <- NA

# Move Category to front of DCs
my_dat <- my_dat %>% relocate(loc, .before=loc_ny_refd_ca)
```

4.3 Rename the Variables "exam1" and "exam2"

```
# I like my names better
my_dat <- my_dat %>% dplyr::rename(exam1 = avg_exm_1)
my_dat <- my_dat %>% dplyr::rename(exam2 = avg_exm_2)
```

4.4 Check the Alpha for "exam1" and "exam2"

Check the Alpha for "exam1" and "exam2" to see if we can make a composite score.

```
# Using column names slice so I don't have to pay attention to what order columns are in
cronbachs_alpha( my_dat[c("exam1", "exam2")] )
```

```
## [1] 0.06939527
```

4.5 Combine Exam Grades for Each Classes

Create 1 variable for exam grade for each class (average of the two)

```
# I'm assuming this means to create a column, exam_mean, which just row-wise means exam1 and exam2. (i.
#  a unique exam_mean value). But this could also be interpreted as meaning across both exams for each
#  (i.e. every "math" exam_mean would be equal). I don't see how the latter is helpful so I'm going to
my_dat$exam_mean <- rowMeans(my_dat[ , c("exam1", "exam2")])
```

4.6 Reorder the Columns

Reorder the Columns so all categories (level, subject, year, location) are listed first, followed by Interpersonal, Exam 1, Exam 2, and average Exam

```
# First we move Interpersonal to the back, then all the exams
my_dat <- my_dat %>% relocate(interp_skls, .after=last_col())
my_dat <- my_dat %>% relocate(exam1, .after=last_col())
my_dat <- my_dat %>% relocate(exam2, .after=last_col())
my_dat <- my_dat %>% relocate(exam_mean, .after=last_col())
```

4.7 Construct Reverse Codes

There was an error in qualtrics and the scores for Interpersonal skills were not set up with reverse coding. Reverse code the Interpersonal scores using R.

```
# Per our descriptives, interp skills runs 1 to 7, so we need to map 1 to 7, 7 to 1, and etc. in the ap
my_dat_rev_code <- my_dat
my_dat$interp_skls <- dplyr::recode(my_dat$interp_skls, '1'=7, '2'=6, '3'=5, '4'=4, '5'=3, '6'=2, '7'=1
```

4.8 Standardize the Exam and Interpersonal Scores

Standardize the Exam and Interpersonal Scores for ease of comparison.

```
# Save unstandardized data for reference
my_dat_unstd <- my_dat

# Scale the 4x numeric vars we have
my_dat$interp_skls <- scale( my_dat$interp_skls, center = TRUE, scale = TRUE )[,1]
my_dat$exam1 <- scale( my_dat$exam1, center = TRUE, scale = TRUE )[,1]
my_dat$exam2 <- scale( my_dat$exam2, center = TRUE, scale = TRUE )[,1]
my_dat$exam_mean <- scale( my_dat$exam_mean, center = TRUE, scale = TRUE )[,1]
```

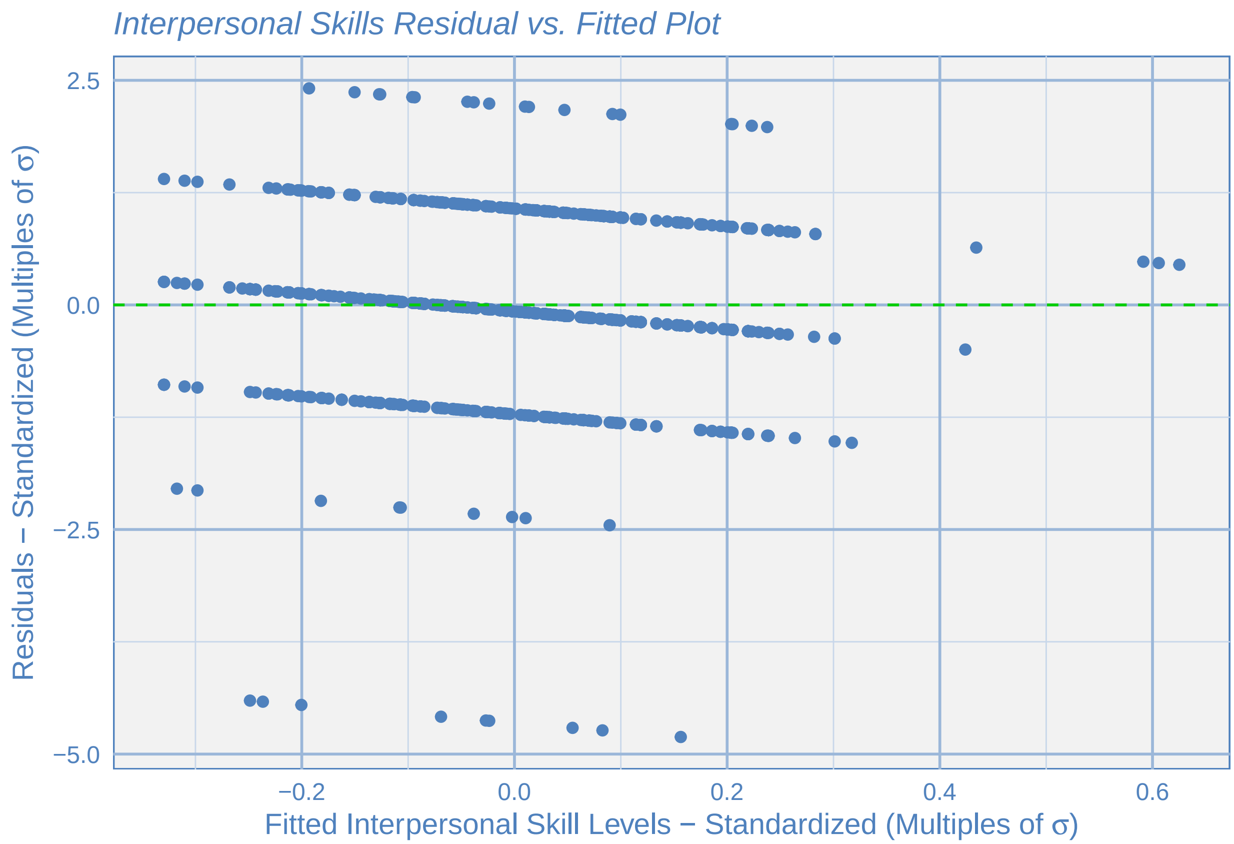
4.9 Dummy Code Location

I already did this in the "create categorical variables" section above.

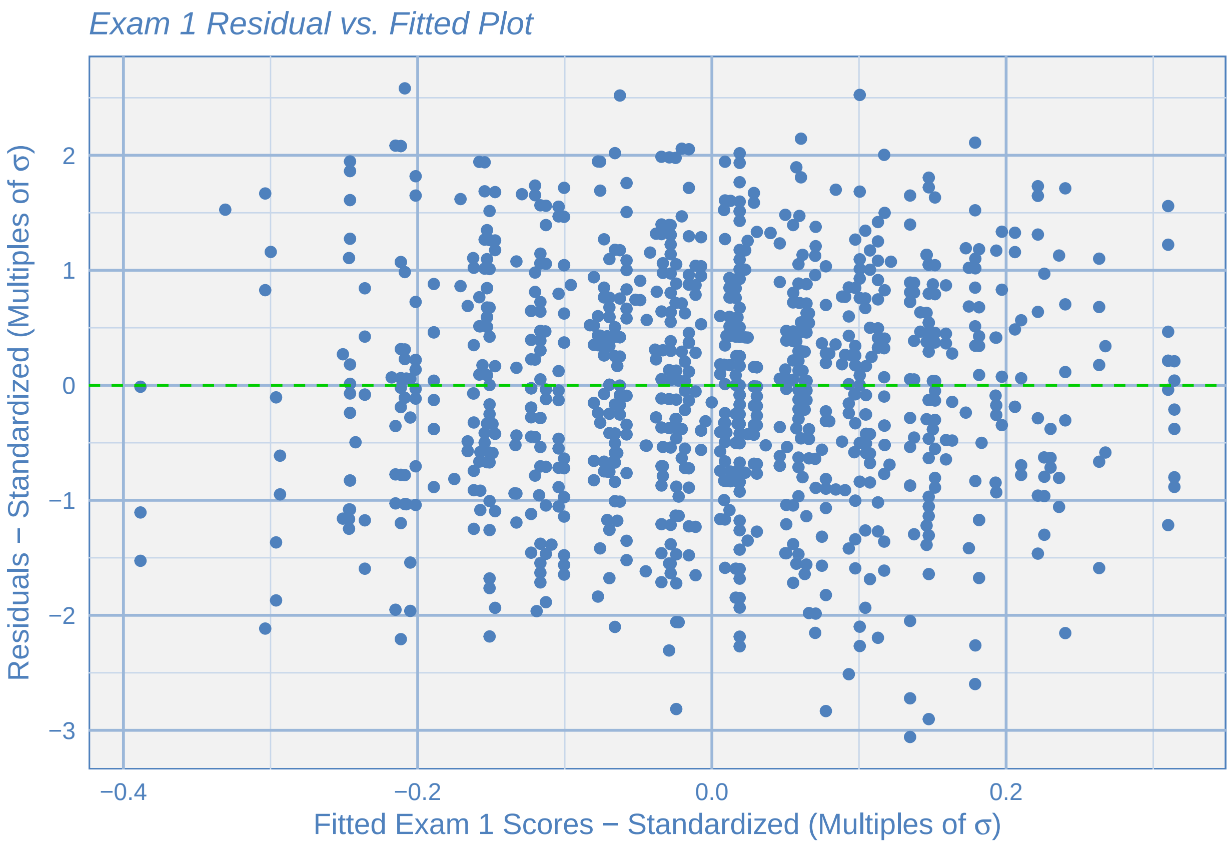
4.10 Detect Outliers and Handle Accordingly.

```
# Specify the fully saturated model for each of the 4x numeric outcome variables (exam scores and interp
mod_interp    <- lm( interp_skls ~ lvl_mid_refd_elem  +
                                   lvl_high_refd_elem  +
                                   sub_sci_refd_math   +
                                   sub_frnch_refd_math +
                                   sub_lang_refd_math  +
                                   sub_art_refd_math   +
                                   sub_span_refd_math  +
                                   sub_pe_refd_math    +
                                   sub_latn_refd_math  +
                                   yr_2013_refd_2012   +
                                   yr_2014_refd_2012   +
                                   yr_2015_refd_2012   +
                                   yr_2016_refd_2012   +
                                   loc_ny_refd_ca,
                     data = my_dat
                   )
mod_exam1     <- lm( exam1 ~ lvl_mid_refd_elem  +
                             lvl_high_refd_elem  +
                             sub_sci_refd_math   +
                             sub_frnch_refd_math +
                             sub_lang_refd_math  +
                             sub_art_refd_math   +
                             sub_span_refd_math  +
                             sub_pe_refd_math    +
                             sub_latn_refd_math  +
                             yr_2013_refd_2012   +
                             yr_2014_refd_2012   +
                             yr_2015_refd_2012   +
                             yr_2016_refd_2012   +
                             loc_ny_refd_ca,
                     data = my_dat
                   )
mod_exam2     <- lm( exam2 ~ lvl_mid_refd_elem  +
                             lvl_high_refd_elem  +
                             sub_sci_refd_math   +
                             sub_frnch_refd_math +
                             sub_lang_refd_math  +
                             sub_art_refd_math   +
                             sub_span_refd_math  +
                             sub_pe_refd_math    +
                             sub_latn_refd_math  +
                             yr_2013_refd_2012   +
                             yr_2014_refd_2012   +
                             yr_2015_refd_2012   +
                             yr_2016_refd_2012   +
                             loc_ny_refd_ca,
                     data = my_dat
```

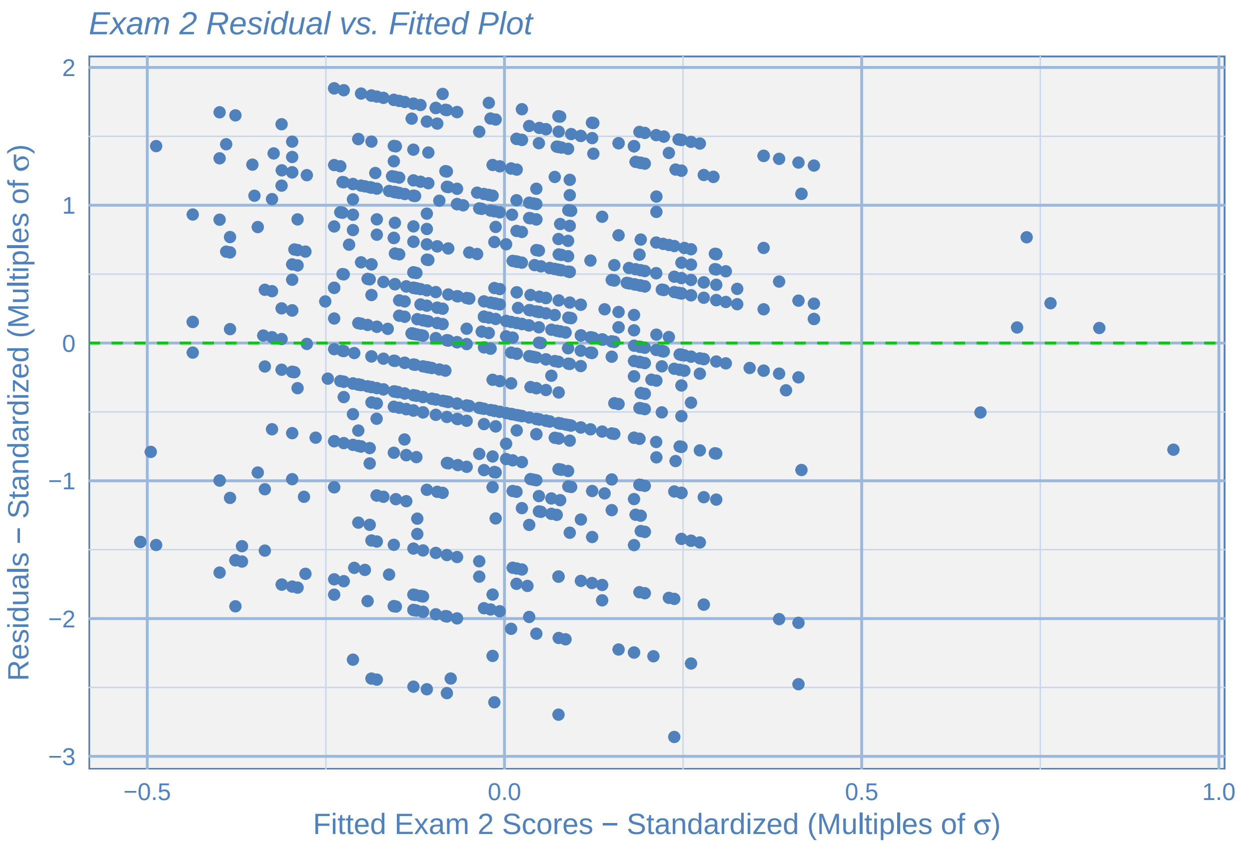
```
                   )
mod_exam_mean <- lm( exam_mean ~ lvl_mid_refd_elem  +
                                 lvl_high_refd_elem  +
                                 sub_sci_refd_math   +
                                 sub_frnch_refd_math +
                                 sub_lang_refd_math  +
                                 sub_art_refd_math   +
                                 sub_span_refd_math  +
                                 sub_pe_refd_math    +
                                 sub_latn_refd_math  +
                                 yr_2013_refd_2012   +
                                 yr_2014_refd_2012   +
                                 yr_2015_refd_2012   +
                                 yr_2016_refd_2012   +
                                 loc_ny_refd_ca,
                     data = my_dat
                   )
#----
# Brady way of Plotting outliers because I'm a bit extra

# Interp Skills per Category
# Since we're using standardized data units will be in Standard Deviations
interp_skls_sd_fig = ggplot( mod_interp,
                             aes(.fitted , .resid )
                            )

interp_skls_sd_fig +
    geom_point(col = font_color) +
    geom_hline(yintercept=0, col="green3", linetype="dashed") +
    xlab(expression( "Fitted Interpersonal Skill Levels - Standardized (Multiples of " * sigma * ")" ) )
    ylab(expression( "Residuals - Standardized (Multiples of " * sigma * ")" ) )  +
    ggtitle("Interpersonal Skills Residual vs. Fitted Plot") +
    my_gg_theme
```



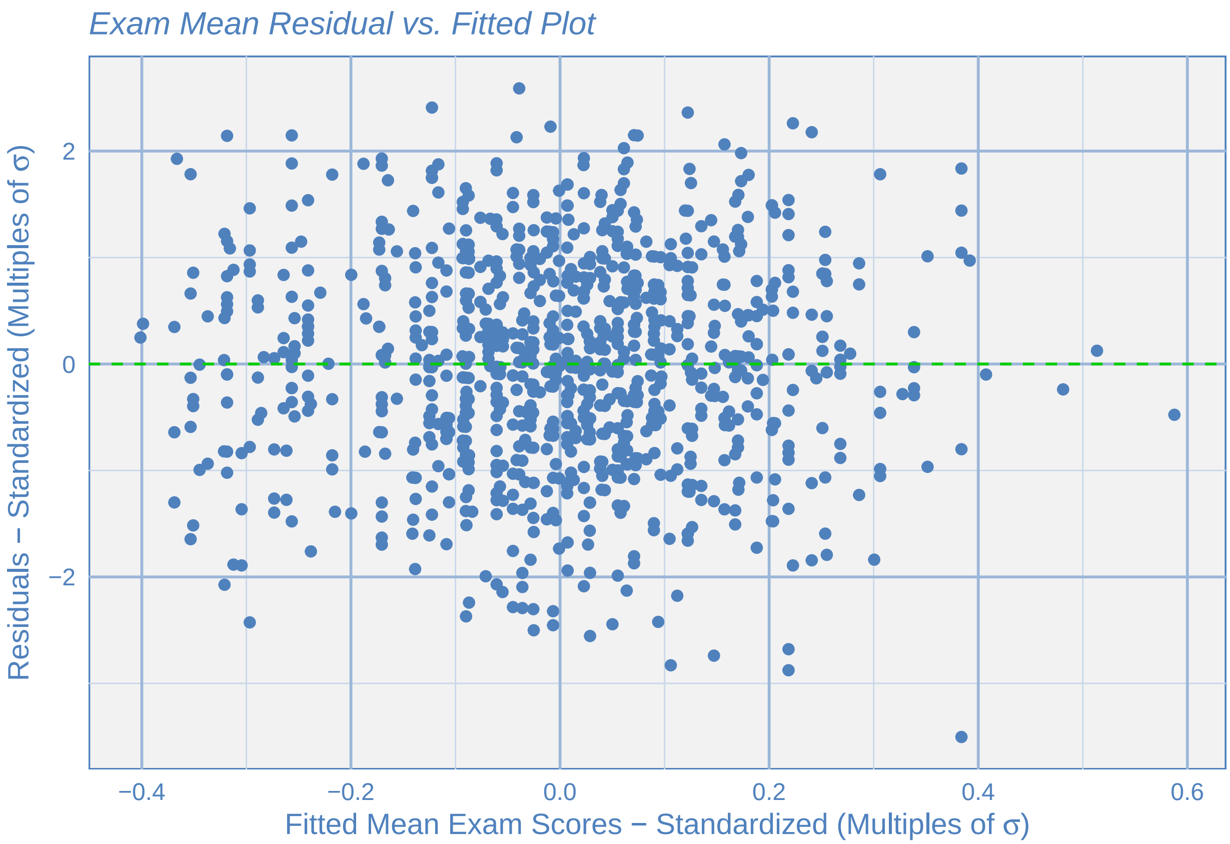

```
# Exam 1 Skills per Category
# Since we're using standardized data units will be in Standard Deviations
exam1_sd_fig = ggplot( mod_exam1,
                       aes(.fitted , .resid )
                          )

exam1_sd_fig +
    geom_point(col = font_color) +
    geom_hline(yintercept=0, col="green3", linetype="dashed") +
    xlab(expression( "Fitted Exam 1 Scores - Standardized (Multiples of " * sigma * ")" ) ) +
    ylab(expression( "Residuals - Standardized (Multiples of " * sigma * ")" ) )  +
    ggtitle("Exam 1 Residual vs. Fitted Plot") +
    my_gg_theme
```



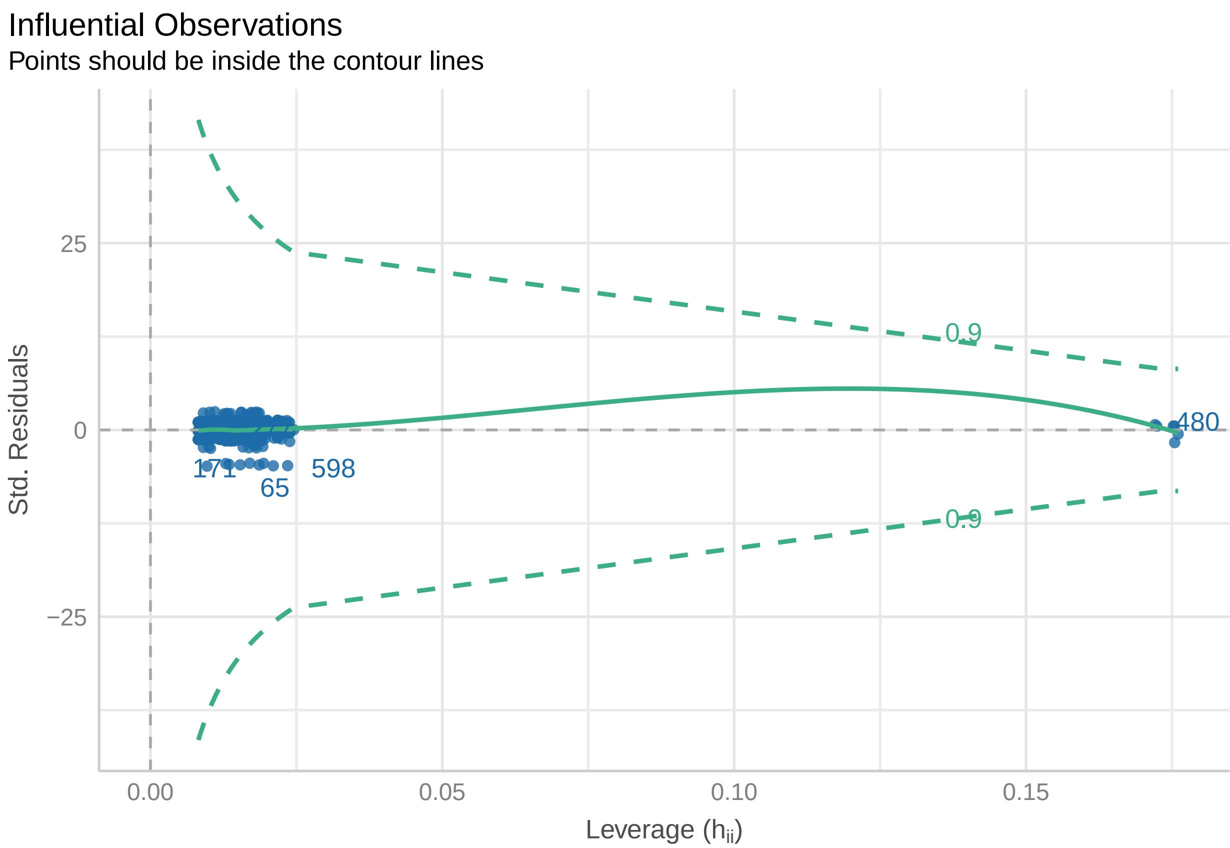

```r
# Exam 2 Skills per Category
# Since we're using standardized data units will be in Standard Deviations
exam2_sd_fig = ggplot( mod_exam2,
                             aes(.fitted , .resid )
                            )

exam2_sd_fig +
    geom_point(col = font_color) +
    geom_hline(yintercept=0, col="green3", linetype="dashed") +
    xlab(expression( "Fitted Exam 2 Scores - Standardized (Multiples of " * sigma * ")" ) ) +
    ylab(expression( "Residuals - Standardized (Multiples of " * sigma * ")" ) )  +
    ggtitle("Exam 2 Residual vs. Fitted Plot") +
    my_gg_theme
```



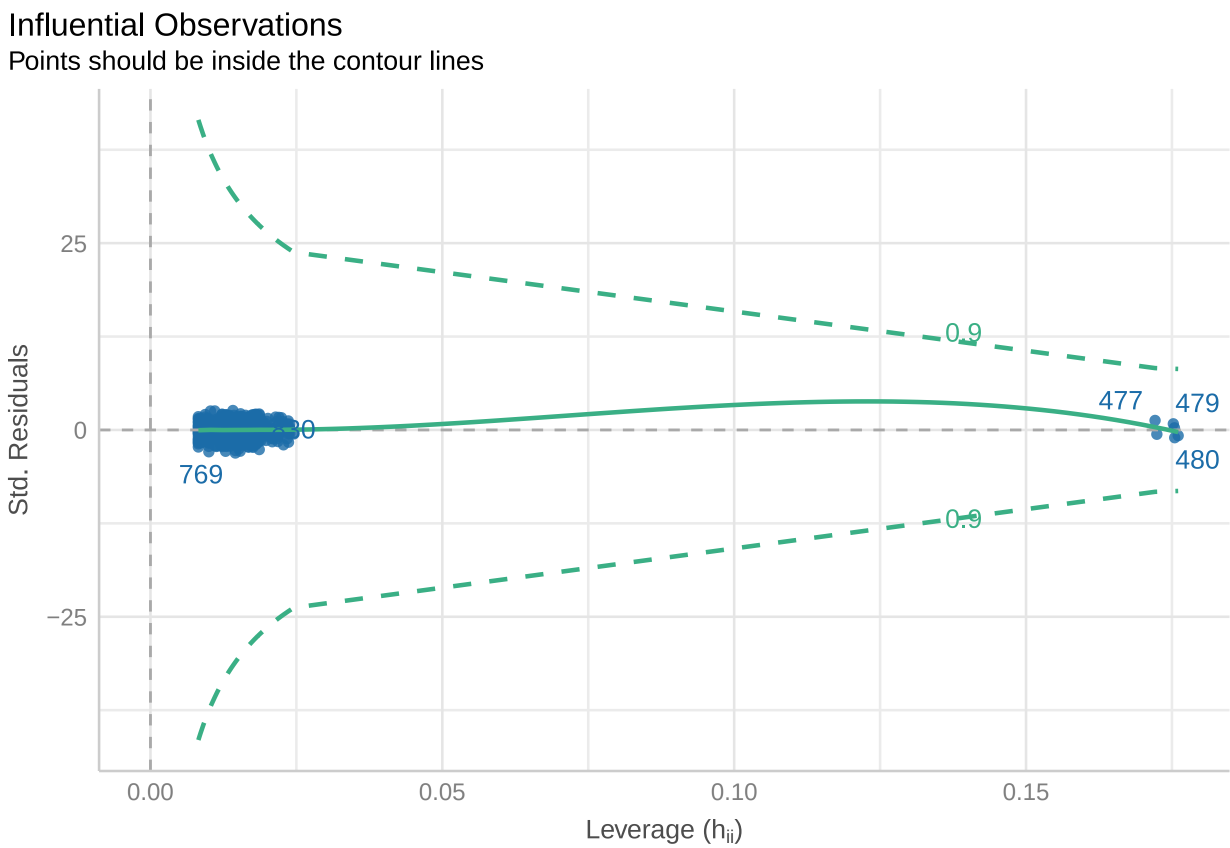

```
# Exam Mean Skills per Category
# Since we're using standardized data units will be in Standard Deviations
exam_m_sd_fig = ggplot( mod_exam_mean,
                              aes(.fitted , .resid )
                            )

exam_m_sd_fig +
    geom_point(col = font_color) +
    geom_hline(yintercept=0, col="green3", linetype="dashed") +
    xlab(expression( "Fitted Mean Exam Scores - Standardized (Multiples of " * sigma * ")" ) ) +
    ylab(expression( "Residuals - Standardized (Multiples of " * sigma * ")" ) )  +
    ggtitle("Exam Mean Residual vs. Fitted Plot") +
    my_gg_theme
```



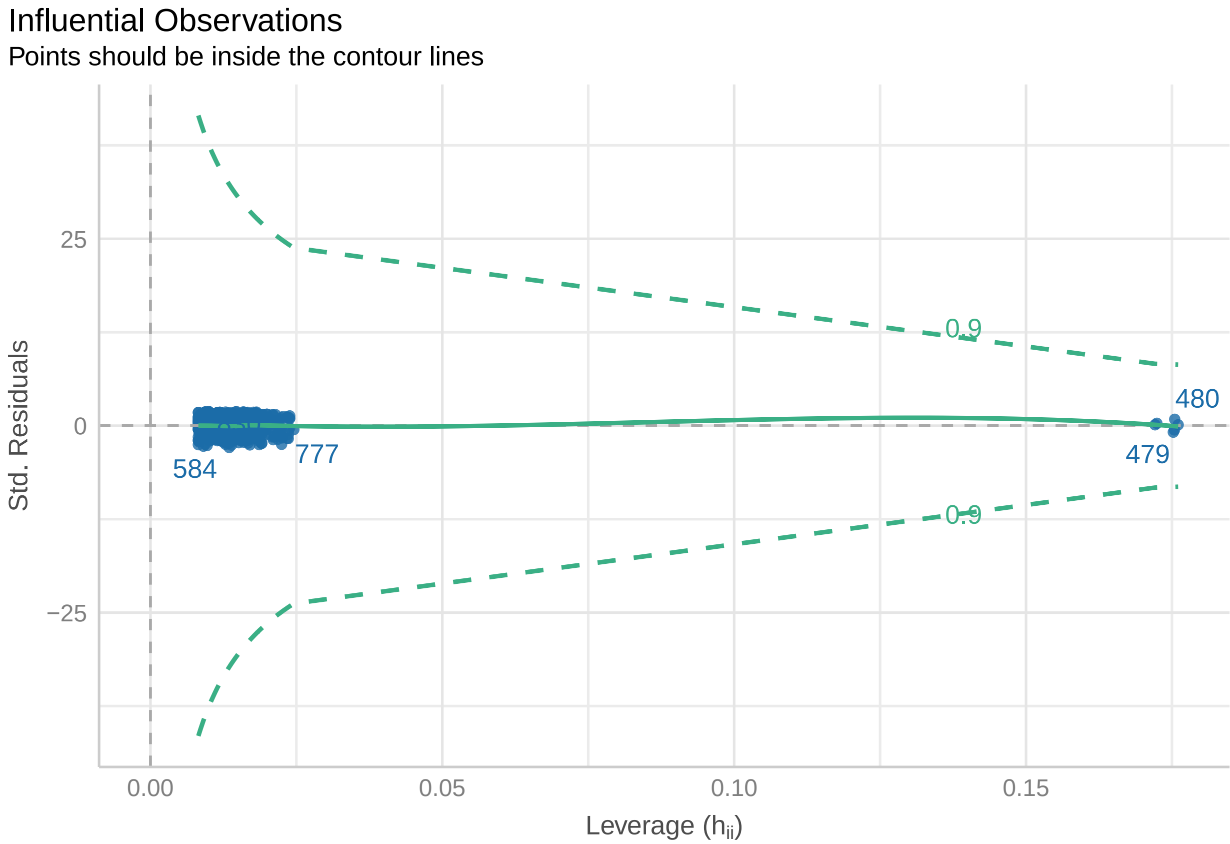

```r
# ----
# Dr. Diaz method of plotting outliers:

# Outlier plots for all 4x vars
interp_outliers <- check_outliers(mod_interp)
plot(interp_outliers, type = "dots")
```



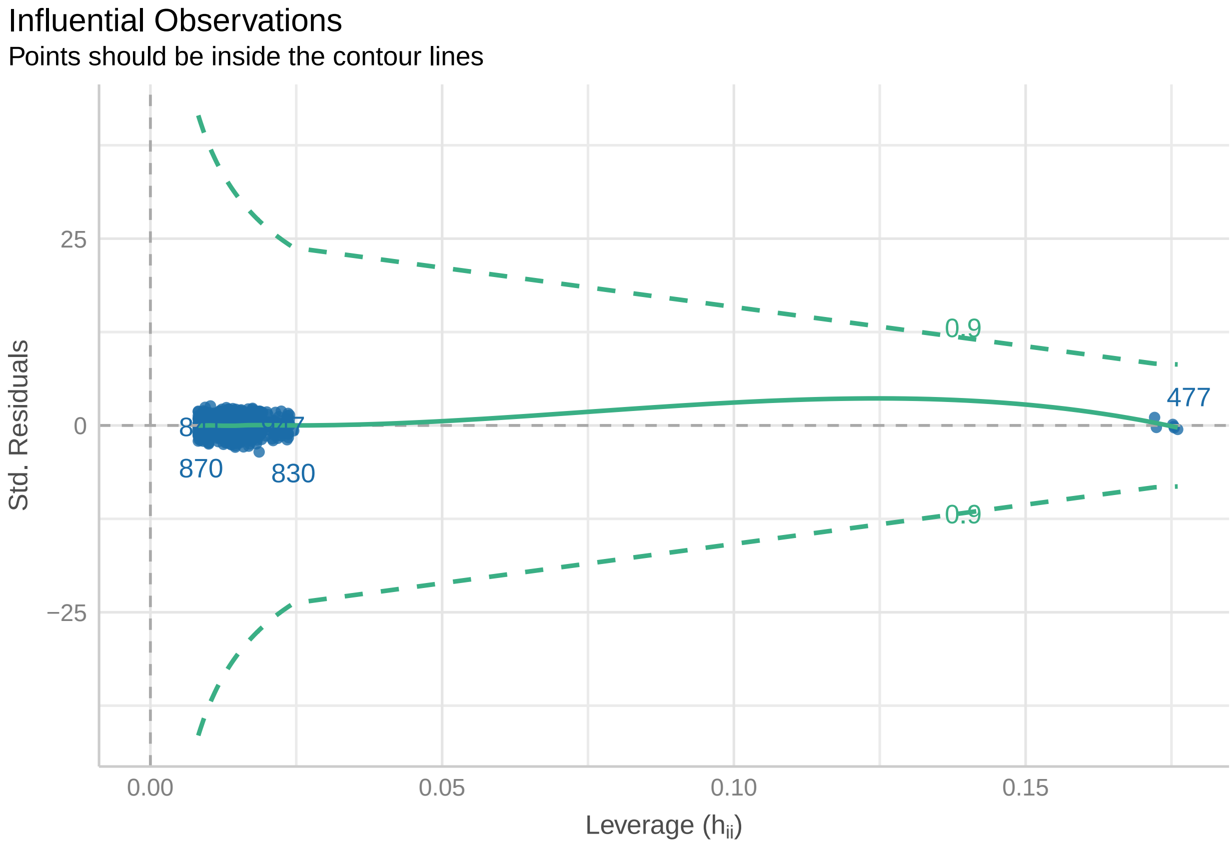

```r
exam1_outliers <- check_outliers(mod_exam1)
plot(exam1_outliers, type = "dots")
```




```r
exam2_outliers <- check_outliers(mod_exam2)
plot(exam2_outliers, type = "dots")
```




```r
exam_mean_outliers <- check_outliers(mod_exam_mean)
plot(exam_mean_outliers, type = "dots")
```




```
#----
#identify multivariate outliers (alpha = 0.001)
# library(performance)

# BJ_Note:Looks above 99% malaanobis distance (1 - .01). Can change to 95% w/ (1 - .05), etc.
#         Increased percentile (e.g. 99%) finds fewer outliers. Decreased percentile (e.g. 95%) finds m
# We look for outliers in all 3 unadjusted numeric variables space (exam 1, exam 2, and interp). We don
#  exam_mean as that's a composite of 1 and 2.
# Used 95% threshold
cont_names <- c("interp_skls", "exam1", "exam2")
out_multi.05 <- check_outliers( my_dat[cont_names],
                                method = "mahalanobis",
                                threshold = stats::qchisq( p = 1 - 0.05, df = ncol( my_dat[cont_names] 


                              )
out_multi.05
```

```
## 36 outliers detected: cases 65, 81, 171, 193, 251, 277, 308, 321, 384,
##   407, 437, 445, 492, 584, 598, 659, 673, 704, 735, 753, 769, 808, 827,
##   830, 839, 841, 846, 859, 866, 870, 883, 901, 916, 942, 947, 958.
## - Based on the following method and threshold: mahalanobis (8).
## - For variables: interp_skls, exam1, exam2.
```

```
# Unstandardized outliers, just to see if its different at all
out_multi.05_unst <- check_outliers( my_dat_unstd[cont_names],
                                     method = "mahalanobis",
                                    threshold = stats::qchisq( p = 1 - 0.05, df = ncol( my_dat_unstd[co
```

```r
                              )
out_multi.05_unst
```

```
## 36 outliers detected: cases 65, 81, 171, 193, 251, 277, 308, 321, 384,
##   407, 437, 445, 492, 584, 598, 659, 673, 704, 735, 753, 769, 808, 827,
##   830, 839, 841, 846, 859, 866, 870, 883, 901, 916, 942, 947, 958.
## - Based on the following method and threshold: mahalanobis (8).
## - For variables: interp_skls, exam1, exam2.
```

```r
# remove outliers
my_dat_with_outliers <- my_dat
my_clean_dat         <- my_dat[!out_multi.05,]

# Remove outliers from unstandardized data too
my_dat_unstd_with_outliers <- my_dat_unstd
my_clean_dat_unstd         <- my_dat_unstd[!out_multi.05,]

# my_dat_unstd_2 <- my_dat_unstd[!out_multi.05_unst,]

# Sample size = 938 (removed 36 outliers whose mahalanobis exceeded the 95% percentile)
```

4.11 Check the Alpha for "exam1" and "exam2" Without Outliers

Check the Alpha for "exam1" and "exam2" a second time with the outliers removed to see if they're any better…

```r
# Using column names slice so I don't have to pay attention to what order columns are in
cronbachs_alpha( my_clean_dat[c("exam1", "exam2")] )
```

```
## [1] 0.04778261
```

```r
cronbachs_alpha( my_clean_dat_unstd[c("exam1", "exam2")] )
```

```
## [1] 0.0459635
```

```r
# cronbachs_alpha( my_dat_unstd_2[c("exam1", "exam2")] )
```

5 Part 2: Queries

5.1 What is the average overall grade for each level of school?

```r
# NOTE: I'm assuming I'm supposed to use my cleaned data for these

# For Elem
elem_avg_grade <- mean( my_clean_dat_unstd$exam_mean[my_clean_dat_unstd$schl_lvl == "ELEM"] )
cat( paste("Average Elementary School Grade: ", elem_avg_grade, "\n\n", sep="") )
```

```
## Average Elementary School Grade: 65.7827988338192
```

```r
# For Middle
mid_avg_grade <- mean( my_clean_dat_unstd$exam_mean[my_clean_dat_unstd$schl_lvl == "MIDD"] )
cat( paste("Average Middle School Grade: ", mid_avg_grade, "\n\n", sep="") )
```

```
## Average Middle School Grade: 63.1443661971831
```

```r
# For High
high_avg_grade <- mean( my_clean_dat_unstd$exam_mean[my_clean_dat_unstd$schl_lvl == "HIGH"] )
cat( paste("Average High School Grade: ", high_avg_grade, "\n\n", sep="") )
```

```
## Average High School Grade: 64.8885209713024
```

5.2 What is the average exam 2 grade for math classes?

```r
# NOTE: I'm assuming I'm supposed to use my cleaned data for these

math_avg_exam2 <- mean( my_clean_dat_unstd$exam2[my_clean_dat_unstd$sub == "Math"] )
cat( paste("Average Math Exam 2 Grade: ", math_avg_exam2, "\n\n", sep="") )
```

```
## Average Math Exam 2 Grade: 85.7865168539326
```

5.3 Calculate the overall average exam grade for all classes.

```r
# NOTE: I'm assuming I'm supposed to use my cleaned data for these

overall_avg_exam <- mean( my_clean_dat_unstd$exam_mean )
cat( paste("Average Overall Exam Grade: ", overall_avg_exam, "\n\n", sep="") )
```

```
## Average Overall Exam Grade: 64.9514925373134
```

5.4 Create a new data frame with only classes from CA.

What is the average exam 1 score?

```r
# NOTE: I'm assuming I'm supposed to use my cleaned data for these
cali_clean_da_unstd <- my_clean_dat_unstd[my_clean_dat_unstd$loc == "CA", ]
cali_avg_exam1 <- mean( cali_clean_da_unstd$exam1 )
cat( paste("Average Cali Exam 1: ", cali_avg_exam1, "\n\n", sep="") )
```

```
## Average Cali Exam 1: 45.6979166666667
```