

# Solutions to Ch3: Data Preparation

MATH 456 - Spring 2016

Right click the link and choose “save link as” to download the [\[RMD\]](#) file that created these solutions.

## Intial setup

It is good practice (and good coding form) to load all libraries and read in all data sets used in the document in the first code chunk. This is also where you would want to set any global options.

```
# Libraries
library(knitr); library(rmarkdown)
library(ggplot2); library(gridExtra)
library(car); library(readxl)
library(dplyr)

# Global options: don't show warnings or messages generated by code chunks
opts_chunk$set(warning=FALSE, message=FALSE)

# Read in all data
depress <- read.table("C:/GitHub/MATH456/data/Depress.txt", sep="\t", header=TRUE)
hiv <- read_excel("C:/GitHub/MATH456/data/Parhiv.xlsx")
```

## On Your Own: Data Wrangling

1. Using the depression data set, create a new variable that collapses the first three education levels.

```
depress$EDUCAT <- factor(depress$EDUCAT,
                        labels = c("<HS", "Some HS", "HS Grad", "Some college", "BS", "MS", "PhD"))
depress$EDUCAT2 <- recode(depress$EDUCAT, "'<HS' = 'Up to HS grad';
                                         'Some HS' = 'Up to HS grad';
                                         'HS Grad' = 'Up to HS grad'")
```

Confirm your recode by displaying a contingency table of the old variable `EDUCAT` against your new variable. Be sure to use the `useNA="always"` argument in the `table()` statement.

```
table(depress$EDUCAT, depress$EDUCAT2, useNA="always")
```

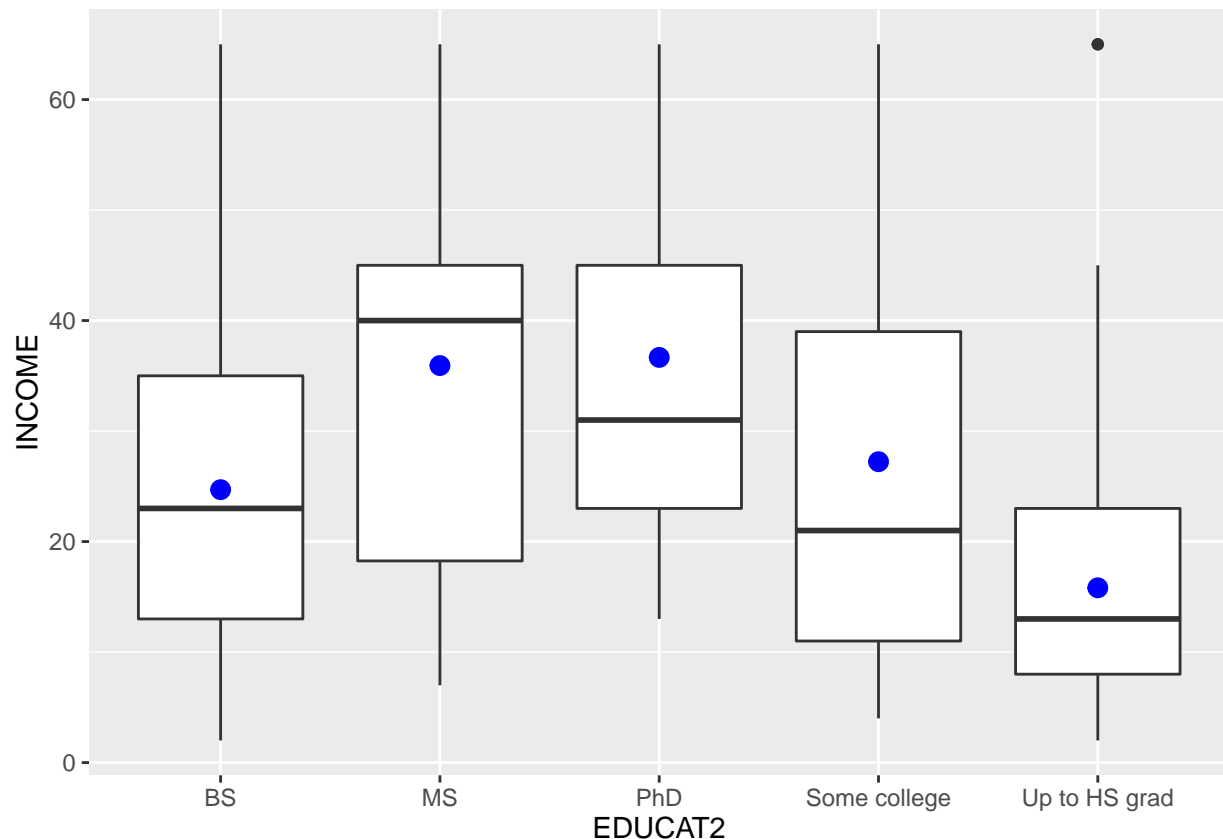
```
##
##           BS  MS PhD Some college Up to HS grad <NA>
## <HS         0   0  0             0             5   0
## Some HS     0   0  0             0             61   0
## HS Grad     0   0  0             0            114   0
## Some college 0   0  0            48             0   0
## BS         43   0  0             0             0   0
## MS          0  14  0             0             0   0
## PhD         0   0  9             0             0   0
## <NA>        0   0  0             0             0   0
```

Recode confirmed. The 5 people with less than HS, 61 with some HS, and the 114 HS grads are now labeled *Up to HS grad* using the variable EDUCAT2.

## 2. What can you say about the relationship between Income and Educational level?

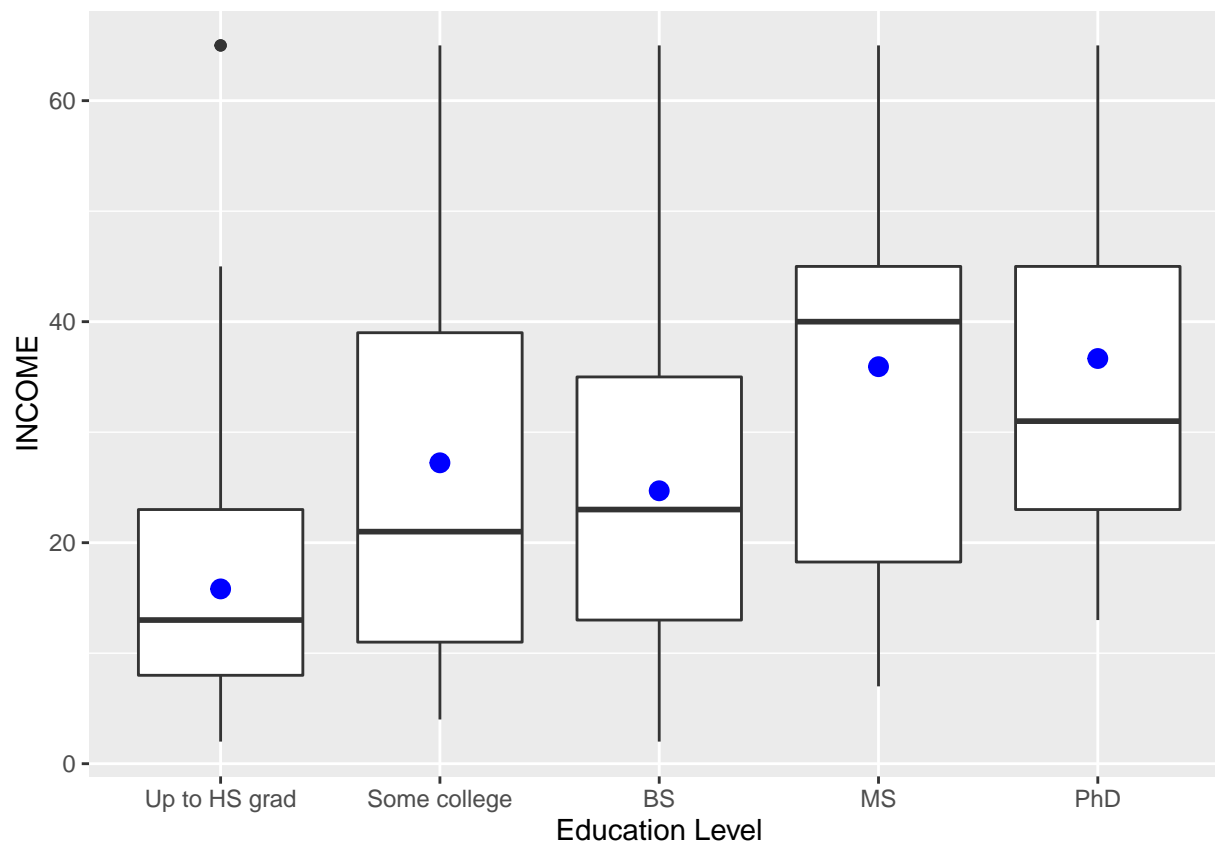
One common way to examine the distribution of a continuous variable `Income` across levels of a categorical variable `Education` is to create side by side boxplots. Using the code from the lecture notes that creates this plot across marital status, and updating the x variable from `MARITAL` to `EDUCAT2` we get the following plot.

```
qplot(y=INCOME, x=EDUCAT2, data=depress, geom="boxplot") +  
  stat_summary(fun.y=mean, colour="blue", size=3, geom="point")
```



Notice carefully now that if you read left to right, education level does not strictly increase. Specifically notice that the categories are displayed in alphabetical order, not in any order that makes reasonable sense. This can be adjusted by specifying the ordering of the levels of the factor variable. The code for this solution was found on [http://www.cookbook-r.com/Manipulating\\_data/Changing\\_the\\_order\\_of\\_levels\\_of\\_a\\_factor/](http://www.cookbook-r.com/Manipulating_data/Changing_the_order_of_levels_of_a_factor/)

```
depress$EDUCAT2 <- factor(depress$EDUCAT2,  
  levels=c("Up to HS grad", "Some college", "BS", "MS", "PhD"))  
qplot(y=INCOME, x=EDUCAT2, data=depress, geom="boxplot", xlab="Education Level") +  
  stat_summary(fun.y=mean, colour="blue", size=3, geom="point")
```



Note that this reordering could have been done in a single step when we first converted the `EDUCAT` variable to a factor variable, but not the new `EDUCAT2` variable.

```
# Example code to create a factor variable and define the levels at the same time.
depress$EDUCAT <- factor(depress$EDUCAT,
  labels = c("<HS", "Some HS", "HS Grad", "Some college", "BS", "MS", "PhD"),
  levels = c("<HS", "Some HS", "HS Grad", "Some college", "BS", "MS", "PhD"))
```

Now it is clear to see that as the amount of education increases so does the mean (blue dots) and median income levels. There is a potential outlier with over \$60k annual income but with no more than a HS diploma. This individual record should be examined in any analysis to determine if it is an influential point.

3. **Determine if any variables in the depression data set have observations that do not fall within the ranges given in the codebook. If there are any, decide what to do with those values and implement your decision.** There are a lot of variables in the Depression data set, so (for me) the easiest way to visually confirm that all values are within the expected range is to do a summary of the entire data set. This produces a lot of output, but I can then go through each variable one by one, and cross-check the data against what is written in the codebook.

Notice now that `SEX` is still being treated as numeric with values 1 and 2, but `EDUCAT` now is being displayed properly as a categorical variable with our specified labels.

```
summary(depress)
```

```
##          ID          SEX          AGE          MARITAL
```

```

## Min.      : 1.00    Min.      :1.000    Min.      : 9.00    Min.      :1.000
## 1st Qu.: 74.25    1st Qu.:1.000    1st Qu.:28.00    1st Qu.:2.000
## Median :147.50    Median :2.000    Median :42.50    Median :2.000
## Mean   :147.50    Mean   :1.622    Mean   :44.38    Mean   :2.374
## 3rd Qu.:220.75    3rd Qu.:2.000    3rd Qu.:59.00    3rd Qu.:3.000
## Max.    :294.00    Max.    :2.000    Max.    :89.00    Max.    :5.000
##
##          EDUCAT      EMPLOY      INCOME      RELIG
## <HS      : 5      Min.      :1.000    Min.      : 2.00    Min.      :1.000
## Some HS   : 61    1st Qu.:1.000    1st Qu.: 9.00    1st Qu.:1.000
## HS Grad   :114    Median :1.000    Median :15.00    Median :1.000
## Some college: 48    Mean   :2.109    Mean   :20.57    Mean   :1.983
## BS        : 43    3rd Qu.:3.000    3rd Qu.:28.00    3rd Qu.:3.000
## MS        : 14    Max.    :7.000    Max.    :65.00    Max.    :6.000
## PhD       : 9
##          C1          C2          C3          C4
## Min.      :0.0000    Min.      :0.000    Min.      :0.0000    Min.      :0.0000
## 1st Qu.:0.0000    1st Qu.:0.000    1st Qu.:0.0000    1st Qu.:0.0000
## Median :0.0000    Median :0.000    Median :0.0000    Median :0.0000
## Mean   :0.3639    Mean   :0.568    Mean   :0.5442    Mean   :0.1939
## 3rd Qu.:0.0000    3rd Qu.:1.000    3rd Qu.:1.0000    3rd Qu.:0.0000
## Max.    :3.0000    Max.    :3.000    Max.    :3.0000    Max.    :3.0000
##
##          C5          C6          C7          C8
## Min.      :0.000    Min.      :0.0000    Min.      :0.0000    Min.      :0.0000
## 1st Qu.:0.000    1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.:0.0000
## Median :0.000    Median :0.0000    Median :0.0000    Median :0.0000
## Mean   :0.551    Mean   :0.2483    Mean   :0.2449    Mean   :0.3503
## 3rd Qu.:1.000    3rd Qu.:0.0000    3rd Qu.:0.0000    3rd Qu.:0.0000
## Max.    :3.000    Max.    :3.0000    Max.    :3.0000    Max.    :3.0000
##
##          C9          C10         C11         C12
## Min.      :0.000    Min.      :0.0000    Min.      :0.0000    Min.      :0.0000
## 1st Qu.:0.000    1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.:0.0000
## Median :0.000    Median :0.0000    Median :0.0000    Median :0.0000
## Mean   :0.568    Mean   :0.4626    Mean   :0.3605    Mean   :0.5136
## 3rd Qu.:1.000    3rd Qu.:1.0000    3rd Qu.:1.0000    3rd Qu.:1.0000
## Max.    :3.000    Max.    :3.0000    Max.    :3.0000    Max.    :3.0000
##
##          C13         C14         C15         C16
## Min.      :0.0000    Min.      :0.0000    Min.      :0.0000    Min.      :0.0000
## 1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.:0.0000
## Median :0.0000    Median :0.0000    Median :0.0000    Median :1.0000
## Mean   :0.3401    Mean   :0.7211    Mean   :0.6735    Mean   :0.7483
## 3rd Qu.:0.0000    3rd Qu.:1.0000    3rd Qu.:1.0000    3rd Qu.:1.0000
## Max.    :3.0000    Max.    :3.0000    Max.    :3.0000    Max.    :3.0000
##
##          C17         C18         C19         C20
## Min.      :0.000    Min.      :0.0000    Min.      :0.0000    Min.      :0.0000
## 1st Qu.:0.000    1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.:0.0000
## Median :0.000    Median :0.0000    Median :0.0000    Median :0.0000
## Mean   :0.619    Mean   :0.3095    Mean   :0.2551    Mean   :0.2483
## 3rd Qu.:1.000    3rd Qu.:0.0000    3rd Qu.:0.0000    3rd Qu.:0.0000
## Max.    :3.000    Max.    :3.0000    Max.    :3.0000    Max.    :3.0000

```

```
##
##      CESD      CASES      DRINK      HEALTH
## Min.   : 0.000   Min.   :0.0000   Min.   :1.000   Min.   :1.000
## 1st Qu.: 3.000   1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:1.000
## Median : 7.000   Median :0.0000   Median :1.000   Median :2.000
## Mean   : 8.884   Mean   :0.1701   Mean   :1.204   Mean   :1.772
## 3rd Qu.:12.000   3rd Qu.:0.0000   3rd Qu.:1.000   3rd Qu.:2.000
## Max.   :47.000   Max.   :1.0000   Max.   :2.000   Max.   :4.000
##
##      REGDOC      TREAT      BEDDAYS      ACUTEILL
## Min.   :1.000   Min.   :1.000   Min.   :0.0000   Min.   :0.0000
## 1st Qu.:1.000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.0000
## Median :1.000   Median :1.000   Median :0.0000   Median :0.0000
## Mean   :1.187   Mean   :1.497   Mean   :0.2143   Mean   :0.2959
## 3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:0.0000   3rd Qu.:1.0000
## Max.   :2.000   Max.   :2.000   Max.   :1.0000   Max.   :1.0000
##
##      CHRONILL      EDUCAT2
## Min.   :0.0000   Up to HS grad:180
## 1st Qu.:0.0000   Some college : 48
## Median :1.0000   BS           : 43
## Mean   :0.5068   MS           : 14
## 3rd Qu.:1.0000   PhD          : 9
## Max.   :1.0000
##
```

Right away, at the end of the first line I notice that RELIG has a max value of 6, when there is no 6th category according to the codebook.

```
table(depress$RELIG)
```

```
##
##    1    2    3    4    6
## 155   51   30   56    2
```

Without any other information to indicate what the correct response should have been, these values are set to missing.

The easiest way to perform a recode when there is only two options (or one simple logical statement) is to use the `ifelse()` function. This has three pieces: `ifelse(logical statement, value if TRUE, value if FALSE)`.

```
depress$RELIG <- ifelse(depress$RELIG == 6, NA, depress$RELIG)
```

Let's break down each piece to help you understand the logic.

```
depress$RELIG<-
```

```
ifelse(depress$RELIG == 6,
```

```
NA,
```

```
depress$RELIG)
```

This line of code says that if the religion variable is 6 (logical statement), then assign the value of this variable to be missing (value if the logical statement is TRUE), otherwise don't change the variable from its current value (value if the logical statement is FALSE).

Then, always, confirm your recodes. It worked because the 2 cases that were under category 6 are now set as NA (missing).

```
table(depress$RELIG, useNA="always")
```

```
##
##      1      2      3      4 <NA>
## 155    51    30    56      2
```

#### 4. Update the Parental HIV data set by creating all the subscales listed at the bottom of the codebook.

I will use this space to show many different ways to approach this task. The methods are not numbered by any real characteristic. For each scale in the list I just thought of a method that would work easily for the variable at hand, that I had not already demonstrated. Some methods are more advanced than others. You will find the method(s) that work best for you.

```
rpb02 <- recode(hiv$PB02, '1=4; 2=3; 3=2; 4=1') # using recode()
table(rpb02, hiv$PB02, useNA="always")
```

**Method 1: Reverse code all sub items that require it by making new variables. Then apply the mean function across that matrix (cbind) of variables.**

```
##
## rpb02      1      2      3      4 <NA>
##      1      0      0      0 26      0
##      2      0      0 38      0      0
##      3      0 56      0      0      0
##      4    131      0      0      0      0
##    <NA>      0      0      0      0      1
```

```
rpb04 <- 5-hiv$PB04 #easier way to flip a scale
rpb14 <- 5-hiv$PB14
rpb16 <- 5-hiv$PB16
rpb18 <- 5-hiv$PB18
rpb24 <- 5-hiv$PB24
```

```
hiv$parent_care <- apply(cbind(hiv$PB01, rpb02, rpb04, hiv$PB05, hiv$PB06, hiv$PB11, hiv$PB12,
                              rpb14, rpb16, hiv$PB17, rpb18, rpb24), 1, mean)
# confirm recode
summary(hiv$parent_care)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
## 1.417   2.917   3.417   3.277   3.750   4.000      1
```

```
hiv$parent_overprotection <- apply(cbind(5-hiv$PB03, 5-hiv$PB07, hiv$PB08, hiv$PB09, hiv$PB10,
                                         hiv$PB13, 5-hiv$PB15, hiv$PB19, hiv$PB20, 5-hiv$PB21,
                                         5-hiv$PB22, hiv$PB23, 5-hiv$PB25), 1, mean)
summary(hiv$parent_overprotection)
```

Method 2: Reverse code sub-items at the time of combining (cbind).

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##    1.000   1.769   2.154   2.117   2.385   4.000         2
```

```
# Find the column numbers whose variable names start with the string BSI
bsi.columns <- grep("^BSI", names(hiv))
# Apply the function mean() row-wise (1) across the column numbers found above.
hiv$BSI_overall <- apply(hiv[,bsi.columns], 1, mean)
# confirm that some numbers were created and that the variable is not fully missing,
# and all values are in the appropriate range of 0 to 4.
summary(hiv$BSI_overall)
```

Method 3: Take the row-wise mean across columns with a variable name that starts with BSI

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##    0.0000   0.2075   0.3962   0.6517   1.0380   3.3020         3
```

```
hiv$BSI_somat <- apply(hiv[,c("BSI02", "BSI07", "BSI23", "BSI29", "BSI30", "BSI33", "BSI37")], 1, mean)
summary(hiv$BSI_somat)
```

Method 4: Take the row-wise mean across specified columns using the variable names

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##    0.0000   0.0000   0.1429   0.4166   0.5714   3.4290         1
```

```
hiv <- hiv %>%
  rowwise() %>%
  mutate(BSI_obcomp = mean(c(BSI05, BSI15, BSI26, BSI27, BSI32, BSI36)))
summary(hiv$BSI_obcomp)
```

Method 5: Using the mutate function, rowwise, in dplyr

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##    0.0000   0.1667   0.5000   0.7961   1.3330   4.0000         1
```

```

hiv <- hiv %>%
  rowwise() %>%
  mutate(BSI_interp = mean(c(BSI20, BSI21, BSI22, BSI42)),
         BSI_depress = mean(c(BSI09, BSI16, BSI17, BSI18 ,BSI35, BSI50)),
         BSI_anxiety = mean(c(BSI01, BSI12, BSI19, BSI38 ,BSI45, BSI49)),
         BSI_hostil = mean(c(BSI06 ,BSI13, BSI40, BSI41, BSI46)),
         BSI_phobic = mean(c(BSI08, BSI28, BSI31, BSI43, BSI47)),
         BSI_paranoid = mean(c(BSI04 ,BSI10 ,BSI24, BSI48 ,BSI51)),
         BSI_psycho = mean(c(BSI03, BSI14, BSI34, BSI44, BSI53))
  )
summary(cbind(hiv$BSI_interp, hiv$BSI_depress, hiv$BSI_anxiety, hiv$BSI_hostil, hiv$BSI_phobic,
              hiv$BSI_paranoid, hiv$BSI_psycho))

```

Method 6: Using Method #5 but for *all* the remaining variables.

```

##      V1      V2      V3      V4
## Min.   :0.0000 Min.   :0.0000 Min.   :0.0000 Min.   :0.0000
## 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.:0.2000
## Median :0.5000 Median :0.3333 Median :0.3333 Median :0.6000
## Mean   :0.7192 Mean   :0.6359 Mean   :0.5186 Mean   :0.9675
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.8333 3rd Qu.:1.6000
## Max.   :4.0000 Max.   :3.8333 Max.   :3.3333 Max.   :4.0000
##
##      NA's :3      NA's :1
##      V5      V6      V7
## Min.   :0.0000 Min.   :0.0000 Min.   :0.00
## 1st Qu.:0.0000 1st Qu.:0.2000 1st Qu.:0.00
## Median :0.2000 Median :0.8000 Median :0.40
## Mean   :0.4421 Mean   :0.9778 Mean   :0.54
## 3rd Qu.:0.6000 3rd Qu.:1.4000 3rd Qu.:0.80
## Max.   :3.6000 Max.   :3.6000 Max.   :3.20
##
##      NA's :2

```

The column names here are listed as V1-V7, which is fine. This was just for the `summary()` information. Nothing that we are going to keep. You can confirm that the variables were created with the names you intended to create using the `names()` function.

```
names(hiv)
```

```

##      [1] "ID"      "AGE"
##      [3] "GENDER"  "LIVWITH"
##      [5] "SIBLINGS" "JOBMO"
##      [7] "EDUMO"   "HOWREL"
##      [9] "ATTSERV" "NGHB1"
##     [11] "NGHB2"   "NGHB3"
##     [13] "NGHB4"   "NGHB5"
##     [15] "NGHB6"   "NGHB7"
##     [17] "NGHB8"   "NGHB9"
##     [19] "NGHB10"  "NGHB11"
##     [21] "MONFOOD" "FINSIT"
##     [23] "ETHN"    "AGESMOKE"
##     [25] "SMOKEP3M" "AGEALC"

```



##	[27]	"AGEMAR"	"FRNDS"
##	[29]	"SCHOOL"	"LIKESCH"
##	[31]	"HOOKEY"	"NHOOKEY"
##	[33]	"HMONTH"	"PB01"
##	[35]	"PB02"	"PB03"
##	[37]	"PB04"	"PB05"
##	[39]	"PB06"	"PB07"
##	[41]	"PB08"	"PB09"
##	[43]	"PB10"	"PB11"
##	[45]	"PB12"	"PB13"
##	[47]	"PB14"	"PB15"
##	[49]	"PB16"	"PB17"
##	[51]	"PB18"	"PB19"
##	[53]	"PB20"	"PB21"
##	[55]	"PB22"	"PB23"
##	[57]	"PB24"	"PB25"
##	[59]	"BSI01"	"BSI02"
##	[61]	"BSI03"	"BSI04"
##	[63]	"BSI05"	"BSI06"
##	[65]	"BSI07"	"BSI08"
##	[67]	"BSI09"	"BSI10"
##	[69]	"BSI11"	"BSI12"
##	[71]	"BSI13"	"BSI14"
##	[73]	"BSI15"	"BSI16"
##	[75]	"BSI17"	"BSI18"
##	[77]	"BSI19"	"BSI20"
##	[79]	"BSI21"	"BSI22"
##	[81]	"BSI23"	"BSI24"
##	[83]	"BSI25"	"BSI26"
##	[85]	"BSI27"	"BSI28"
##	[87]	"BSI29"	"BSI30"
##	[89]	"BSI31"	"BSI32"
##	[91]	"BSI33"	"BSI34"
##	[93]	"BSI35"	"BSI36"
##	[95]	"BSI37"	"BSI38"
##	[97]	"BSI39"	"BSI40"
##	[99]	"BSI41"	"BSI42"
##	[101]	"BSI43"	"BSI44"
##	[103]	"BSI45"	"BSI46"
##	[105]	"BSI47"	"BSI48"
##	[107]	"BSI49"	"BSI50"
##	[109]	"BSI51"	"BSI52"
##	[111]	"BSI53"	"parent_care"
##	[113]	"parent_overprotection"	"BSI_overall"
##	[115]	"BSI_somat"	"BSI_obcomp"
##	[117]	"BSI_interp"	"BSI_depress"
##	[119]	"BSI_anxiety"	"BSI_hostil"
##	[121]	"BSI_phobic"	"BSI_paranoid"
##	[123]	"BSI_psycho"	

See, all the subscales have been appended to the end of the data set in columns 112 through 123.

Use the `write.table()` function to write this data set as a tab-delimited text file using the current date in the file name.

```
write.table(hiv, "C:/GitHub/MATH456/data/PARHIV_013116.txt", sep="\t", row.names=FALSE, col.names=FALSE)
```

## Session Info

This document was compiled on 2016-02-01 16:40:08 and with the following system information:

```
sessionInfo()
```

```
## R version 3.2.2 (2015-08-14)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] dplyr_0.4.3      readxl_0.1.0    car_2.1-1      gridExtra_2.0.0
## [5] ggplot2_2.0.0    rmarkdown_0.8.1 knitr_1.11
##
## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.1      formatR_1.2.1    nloptr_1.0.4
## [4] plyr_1.8.3       tools_3.2.2      digest_0.6.8
## [7] lme4_1.1-10      evaluate_0.8      gtable_0.1.2
## [10] nlme_3.1-121     lattice_0.20-33  mgcv_1.8-7
## [13] Matrix_1.2-2     DBI_0.3.1        yaml_2.1.13
## [16] parallel_3.2.2   SparseM_1.7      stringr_1.0.0
## [19] MatrixModels_0.4-1 grid_3.2.2        nnet_7.3-10
## [22] R6_2.1.1         minqa_1.2.4      magrittr_1.5
## [25] scales_0.3.0     htmltools_0.2.6  MASS_7.3-43
## [28] splines_3.2.2    assertthat_0.1   pbkrtest_0.4-4
## [31] colorspace_1.2-6 labeling_0.3      quantreg_5.19
## [34] stringi_0.5-5    lazyeval_0.1.10  munsell_0.4.2
```