TEND_Group_Testing

Me

2023-05-27

R Markdown

Setting Working Directory

```
## here() starts at C:/Users/adcre/OneDrive/Documents/Desktop_RStudio
```


```
## here() starts at C:/Users/adcre/OneDrive/Documents/Desktop_RStudio
```


```
## [1] "C:/Users/adcre/OneDrive/Documents/Desktop_RStudio"
```


```
## [1] "C:/Users/adcre/OneDrive/Documents/Desktop_RStudio"
```


Adding libraries/packages

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
##
## Attaching package: 'psych'
##
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
##
##
##
## Please cite as:
##
##
##  Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##
##  R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

Load and Prepare Data

```
Brain_Data <- read.csv("Brain_Data.csv", header=T, sep=",", na.strings=c("NA", "888", "999"))
Behavioral_Data<-read.csv("Bx_Data.csv", header=T, sep=",", na.strings=c("NA", "888", "999"))

# Merge the data sets
DATA <- merge(Brain_Data, Behavioral_Data, "ID", all=T)
View(DATA)

# The most common form of data is factors, numbers, strings and characters. They are all understood by
DATA$ID<- as.factor(DATA$ID)
DATA$Age<- as.numeric(DATA$Age_at_Bx)

# It will be a very good habit to not only check what form your data or columns are represented as, it
```

Start Exploring

```
# First we will use colnames() to see the names of the variables of our data set
# Assign the colnames to a variable through a data frame, to be accessed later saves this as a data fra
Column_indexes <- data.frame(colnames(DATA))
str(Column_indexes)
```

```
## 'data.frame':    39 obs. of  1 variable:
##  $ colnames.DATA.: chr  "ID" "Group.x" "RSFC1" "RSFC2" ...
```

```
# which() allows us to know the index a specific variable, ex "Gender" is our 10th column variable
which(colnames(DATA) == "Gender")
```

```
## [1] 10
```

mutate()

```
# coming from dplyr, the primary purpose of mutate() is to add new columns to a data frame based on cal
```