```r
# THIS FUNCTION WAS DEFINED TWICE
# I THINK YOU USED THE OTHER ONE BUT
# choose.test <- function(hypothesis.type, variable.types, crit = NULL, paired = NULL){
#
#   typestring <- paste(c("TYPE",hypothesis.type,variable.types, paired), collapse = "_")
#   # typestring="TYPE_direct_reporting_simple_random_numeric"
#   TYPE_group_difference_categorical_categorical <- hypothesis_test_chisquare
#   # TYPE_group_difference_numeric_categorical <- hypothesis_test_difference_in_means
#   # TYPE_limit_numeric <- hypothesis_test_one_sample_z_num
#   # TYPE_limit_categorical <- hypothesis_test_one_sample_z_cat
#   TYPE_direct_reporting_numeric <- confidence_intervals_num
#   # TYPE_direct_reporting_categorical <- confidence_intervals_cat
#   # TYPE_change_over_time_numeric <- hypothesis_test_difference_in_means #paired or unpaires
#   # TYPE_change_over_time_categorical <- hypothesis_test_chisquare
#   # TYPE_correlation_numeric_numeric <- hypothesis_test_regression
#   # TYPE_correlation_categorical_numeric <- hypothesis_test_logistic_regression #warn: categorical variable must be binary (ensure)
#   #
#
#   return(get(typestring))
#
# }

find.data.types <- function(data.dependent.var, independent.var = NULL) {



  data.type.dep = c()
  data.type.indep = c()

  if(question_is_categorical(data.dependent.var) == T){
    data.type.dep = "categorical"
  }
  if(question_is_numeric(data.dependent.var) == T){
    data.type.dep = "numeric"
  }
  if(question_is_categorical(independent.var) == T){
    data.type.indep = "categorical"
  }
  if(question_is_numeric(independent.var) == T){
    data.type.indep = "numeric"
  }
  # fixed the underscore etc. issue
  variable.types <- paste(c(data.type.dep, data.type.indep), collapse="_")
  return(variable.types)
}

# Function that finds out what analysis case you have
# each analysis case can be mapped to a single, appropriate..
  # summary statist
  # hypothesis test
  # visualisation.
# It depends on
  # the types and number of dependent and indepenendent variables
  # the hypothesis type
  # the sampling strategy

analysis_case<-function(data,
                        hypothesis.type,
                        dependent.var = dependent.var,
                        independent.var = independent.var){


}

# repetition between those two functions!
# there should be a generic way to map from analysis parameters to a case, and from a case to a test, and from a case to a stat

# Function that chooses which case your lovely combination of variables falls
# if it throws an error at the variable_type level, consider changing it to the right type
# data$dependen.var <- as.numeric(sub(",", ".", as.character(data$dependent.var)))
choose.summary <- function(hypothesis.type = hypothesis.type,
                           data = data,
                           dependent.var = dependent.var,
                           independent.var = independent.var,
                           paired = NULL) {

  variable.type <- paste0(reachR:::variable_type(dependent.var), "_", reachR:::variable_type(independent.var))
  typestring <- paste(c("TYPE",hypothesis.type,variable.type, paired), collapse = "_")
  TYPE_difference_in_groups_categorical_categorical <- hypothesis_test_chisquare
  return(get(typestring)) }

# To be combined with the choose.summary function above so 1) chooses the case 2) determines the relevant
# summary statistics and tests and visual representations for this case
choose.test <- function(hypothesis.type = hypothesis.type,
                        data = data,
                        dependent.var = dependent.var,
                        independent.var = independent.var,
                        paired = NULL) {
  variable.type <- paste0(reachR:::variable_type(dependent.var), "_", reachR:::variable_type(independent.var))
  typestring <- paste(c("TYPE",hypothesis.type,variable.type, paired), collapse = "_")
  TYPE_difference_in_groups_categorical_categorical <- hypothesis_test_chisquare
  #TYPE_group_difference_numeric_categorical <- hypothesis_test_difference_in_means
  #TYPE_limit_numeric <- hypothesis_test_one_sample_z_num
  #TYPE_limit_categorical <- hypothesis_test_one_sample_z_cat
  return(get(typestring)) }
```