```
##############################################
# general structure
##############################################



#' things that trigger me to think that the structure should change:
  #' first indication was that the case is identified in multiple places: once in choose.summary, again in choose.test and presumably again in
visualise.results.
  #' then we pass the same stuff over and over that (maybe?) is only used to identify the case? if we knew the case, could we pass less?
  #' then there's the design object:
    #' to make it, we need to know the case; so it makes no sense that it's created by the user before the case is identified.
    #' on the other hand, we need some extra info that currently isn't given to the analyse_indicator function, so we cant make it in there (with
the current structure)
    #' but then again we need to create a unique design object for each analyse_indicator call to make sure it's created _after_ the missing data
for that indicator is removed
    #' woah cool you can use minimarkdown in rstudio comments *bold* _italic_ ~strikethrough?~ ok no strikethrough sad saaaad
    #' then again the design object contains almost all of the info we need for the hypothesis testing etc;
    #'

#' maybe it would make more sense to have
  #' one function that identifies the type to be called on top of analyse_indicator
  #'


#skeleton
analyse_indicator<-function(data, dependent.var, independent.var = NULL, hypothesis.type, design, do.for.each.unique.value.in.var = NULL){
##############################################
# do we have to pass the data AND the design object? wouldn't principle the design object would contain the data?
##############################################


##############################################
# why removed these input cleanup?
  # data <- data[!is.na(dependent.var),]
  # data <- data[!(dependent.var %in% c("NA", "N/A")),]
# oh i see there's a bug. fixed version:
  data <- data[!is.na(data[,dependent.var]),]
# you shouldn't have just removed that, but fixed it

##############################################

        # sanitise input
            if(nrow(data)==0){stop('provided data has no rows')}
            if(all(is.na(data[,dependent.var]))){stop(paste('variable', dependent.var, 'can\'t be all NA'))}
            if(all(is.na(data[,independent.var]))){stop(paste('variable', independent.var, 'can\'t be all NA'))}

        # get function for the appropriate summary statistic:  mean, median, percent, count..:
            summarise.result<- choose.summary(hypothesis.type, data, dependent.var, independent.var)
        # get function for the appropriate hypothesis test:
            test.hypothesis <- choose.test(hypothesis.type, data, dependent.var, independent.var)



        # do summary statistic:
            summary.result  <- summarise.result(dependent.var,independent.var, design)
        # do hypothesis test:
            hypothesis.test.result<- test.hypothesis(dependent.var,independent.var, design)

        # get ggplot object for the appropriate visualisation:
            visualisation <- visualise.results(summary.result, hypothesis.test.result)

##############################################
# YOU HAD RETURNED A NON EXISTENT VARIABLE CALLED `results`
##############################################
        return(list(
                    summary.result,
                    hypothesis.test.result,
                    visualisation
              ))

    }



##############################################
# the "choose" functions are mapping scales and should have been in the corresponding file. I moved them :)
##############################################


#would you like to test you function?
##############################################
# YES PLEASE BUT NOT HERE
# this is important because i want to be able to source() this file to load the functions without executing any of them.
# i moved this to the bottom of ./../tests.R.
# this is also important because i totally zero can not execute this script otherwise:
  # 1. i can't run this file because no data is loaded
  # 2. i can't call this file from another file because it would fail depending on whether some other variables from somewhere else exist
  # 3. now there's all the functions in ./scripts/..., and all the line by line executed code in ./tests.R
##############################################
```