

```
#####CHI SQUARED
#function that executes the literal chi squared test

#####
# consistent naming strategy?

# naming should be in line with what side of each mapping we're on: analysis parameters VS. cases VS. statistical tests etc.
# that's why group_difference_categorical_categorical() is such a terrible function(name). It actually does a chi squared test. The mapping from
case to test happens between fun name and fun content, while really that should happen in the mapping function.

#####

do.a.chi.sq <- function(independent.var = independent.var,
                        dependent.var = data.dependent.var,
                        design,
                        na.rm = TRUE){
  formula_string<-paste0("~",independent.var, "+", dependent.var)
  chisq <- svychisq (formula(formula_string), design, na.rm = TRUE)
  results$test.results <- c(chisq$statistic, chisq$p.value)
  results$test.parameters <- c(chisq$parameter, chisq$method)
  return(results)
}

#function that regroups all procedures for group_difference_categorical_categorical

#####
# comment not readable. what does this fun do? no idea; there's also no comments inside so it's WORSE! HARG! JALSIFEJLIEWJSFKf\@SKLNj~OP@j\fp:j!
# actually i changed my mind screw the comments we don't need the comments.
# ' The real problem is the name of the function, more specifically it's a shit name.
# ' It doesnt tell me at all what it does. like, 0. The best guess from the name would be that it does ANYTHING todo with the case, but it
doesnt
# ' we could do that but that would mean to structure the whole thing like identify case -> call function that does everything for that case.
# ' but our current structure is better than that
# ' the names of the functions used _inside_ this function are great, so i can read what it does easily, so need no comments here either.
# ' i thought confints should go into one place with the summary stat functions, or in it's own place?
#####

group_difference_categorical_categorical<- function(dependent.var, independent.var, design){
  results <- list()
  chisq.results <- do.a.chi.sq(independent.var = independent.var, dependent.var = dependent.var, design = design)
  confints <- percent_with_confints(independent.var = independent.var, dependent.var = dependent.var, design = design)
  results <- list(chisq.results, confints)
  return(results)}

#####
# assuming we have a fun like this for each summary stat option
# and they all come with an associated error stuff, maybe this should be called:
# summarise_[NAMEOFSTAT]() or something
# summarise_percent
# well i guess i went full circle to aggregate_median etc.
# essentially same thing so makes sense
#####
percent_with_confints <- function(independent.var = independent.var,
                                dependent.var = data.dependent.var,
                                design,
                                na.rm = TRUE){
  formula_string<-paste0("~",independent.var, "+", dependent.var)
  f.table <- svytable(formula(formula_string), design)
  formula_err <- paste0("~", dependent.var, sep = "")
  by <- paste0(" ~", independent.var, sep = "")
  error_bars <- svyby(formula(formula_err), formula(by), design, na.rm = T, svymean)
  names_df <- sapply(rownames(f.table), paste0, colnames(f.table))
  results$names <- c(names_df[,1], names_df[,2])
  results$numbers <- as.numeric(c(prop.table(f.table, 1)[1,], prop.table(f.table, 1)[2,]))
  results$se <- as.numeric(c(error_bars[,grep("se.", names(error_bars))][1,], error_bars[,grep("se.", names(error_bars))][2,]))
  results$min <- results$numbers - results$se
  results$max <- results$numbers + results$se
  return(results)
}

# underscores or dots in names?
#####
# if i ever used "a" or "the" in a variable name then i didn't mean it
#####
visualise.a.chisq <- function(){
  test_name <- hypothesis.test.results$test.parameters[[3]]
  p_value <- hypothesis.test.results$test.results[[2]]

  chart <- reach_style_barchart(group = summary.result$names,
                              percent = summary.result$numbers,
                              error_min = summary.result$min,
                              error_max = summary.result$max)

  chart + geom_text(aes(x =4,
                        y = 2,
                        label= paste0("To determine ", hypothesis.type, "\n", test_name, "\n"
                                      , " returned a p value of ", round(p_value,6))),
                    size=3,
                    family="Arial Narrow",
                    col='#000000',
                    hjust=0,
                    vjust=0.5)}

#####ONE SAMPLE Z tEST
hypothesis_test_one_sample_z_num <- function(data.dependent.var, crit, design, data = data) {
  svytesttest(data[[dependentvar]]~data[[independent.var]], design = design, family = quasibinomial())
}

##### DIRECT REPORTING NUMERICAL
confidence_intervals_num <- function(dependentvar, design, data = data){
  summary <- svymean(data[[dependentvar]], design, na.rm = T)
```

```
confint(svymean(data[[dependentvar]] + data[[dependentvar]], design, na.rm = T), level = 0.95)  
}
```