YouTube Video Scam Project Data Analysis

Elijah Bouma-Sims

2024-07-04

Introduction

This R Markdown file contains the analysis code for the paper "The Kids Are All Right: Investigating the Susceptibility of Teens and Adults to YouTube Giveaway Scams" by Elijah Bouma-Sims, Lily Klucinec, Mandy Lanyon, Lorrie Faith Cranor, Julie Downs. paper. Please see README.md for a descirption of all the files in this artifact.

Requirements

Running this notebook requires the packages "dplyr", "RVAideMemoire", "rstatix", "readxl", "rcompanion", "DescTools", and "stringr".

Code overview

In this section, we describe the structure of the code in "chronological" order.

The code begins with a setup section that loads packages/data, creates factors, etc. Data is pulled from the `data\df_analysis.xlsx` In the setup code, we also create a function called `stat_test`. This function will be used repeatedly throughout the code, so it's worth reviewing in detail. It is called to run statistical tests to check whether an outcome variable (specified by the label in `dep_var`) varies with respect to any of our potential explanatory variables (described in Table 1 of the paper). It runs the appropriate test and computes the appropriate effect size measure based on the type of the variables. Please see the function specification for a full ist of parameters.

After the setup code, we proceed to generate tables of describe statistics for demographic variables for both adult and teen participants. The output from these code blocks was used to generate table 2 from the paper.

The next section of the document contains all of the statistical testing code. Sub-sections are named based on the dependent variable being analyzed in a particular code block, with further subdivisions as appropriate. For example, the "Legit actions" subsection contains statistical testing results for users recommended actions in response to legit stimuli. The analysis for users reactions to the YouTube video and web video are under the subheadings "Legit video" and "Legit web" respectively.

In each statistical testing subsection, we run the `stat_test` function or the `cochran.qtest` function (for search result seleciton) to perform the statistical testing between the independent variables and the relevant dependent variable(s). If any results are significant, we use the "table" function to view how the dependent variable varies with the independent variable. Post-hoc tests run on a particular variable are listed under the appropriate heading for their dependent variable, and explicitly labeled as post-hoc results.

Finding key statistical testing results

The length of this file is necessary to document all of the statistical tests we performed, but it can make it difficult to find particular results. To ease navigation, statistical testing results that are highlighted in the paper will begin with the text **Paper Result**. This should allow you to search the document for everywhere the term "Paper Result" appears in order to jump to those sections explicitly discussed in the paper.

Aanlysis code

Setup

The following code imprts packages and loads the survey data (`df_merged`).

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("RVAideMemoire") # for cocran q
```

```
## *** Package RVAideMemoire v 0.9-83-7 ***
```

```
library("rstatix") # for cramer's v
```

```
##
## Attaching package: 'rstatix'

## The following object is masked from 'package:stats':
##
##     filter
```

```
library("readxl") # to load data
library("rcompanion") # for freeman theta
library("DescTools") # for CochranArmitageTest
library("stringr") # for str_split
data_path <- "data\\df_analysis.xlsx" # windows path
# load analysis data.
df_merged <- read_xlsx(data_path)
```

The following code creates new columns which are based upon the same question asked for Spotify and Roblox participants. For example, `risk_usedSpotify` and `risk_playedRoblox`, which are the results of a question asking how frequently the participant uses spotify/plays Roblox, are merged into a new column "risk_usedService""

```
# usedSpotify and playedRoblox
df_merged <- df_merged %>% mutate(risk_usedService = ifelse(is.na(risk_usedSpotify), ifelse(is.na(risk_p

# freeSpotify and freeRobux
df_merged <- df_merged %>% mutate(risk_freePremium = ifelse(is.na(risk_freeSpotify), ifelse(is.na(risk_f

# spotifyPremium and purchasedRobux
df_merged <- df_merged %>% mutate(risk_purchased = ifelse(is.na(risk_spotifyPremium), ifelse(is.na(risk_

# spotify_s and roblox_s
df_merged <- df_merged %>% mutate(risk_searchedBefore = ifelse(is.na(spotify_s), ifelse(is.na(roblox_s)
```

The following code turns the columns of categorical variables into factors for proper statistical analysis.

```
weekly_time_levels = c("None",  "Less than an hour", "1 to 5 hours", "5 to 10 hours", "10 to 15 hours",
usage_levels = c("Never", "Once or twice", "Three to five times", "More than five times")
income_levels = c("Less than $20,000", "$20,000 to $39,999", "$40,000 to $59,999", "$60,000 to $79,999"
rank_scam_levels = c("Definitely legitimate", "Probably legitimate", "I'm not sure", "Probably a scam",

# time factors
df_merged$time_overall <- factor(df_merged$time_overall, ordered = TRUE,levels=c("Less than 2 hours", "
df_merged$time_computer <- factor(df_merged$time_computer, ordered = TRUE,levels=weekly_time_levels)
df_merged$time_videos <- factor(df_merged$time_videos, ordered = TRUE,levels=weekly_time_levels)
df_merged$time_mobile <- factor(df_merged$time_mobile, ordered = TRUE,levels=weekly_time_levels)
df_merged$time_nonsocial <- factor(df_merged$time_nonsocial, ordered = TRUE,levels=weekly_time_levels)
df_merged$time_social <- factor(df_merged$time_social, ordered = TRUE,levels=weekly_time_levels)

# Potential experiential risk factors
df_merged$risk_coupons <- factor(df_merged$risk_coupons, ordered = TRUE,levels=usage_levels)

df_merged$risk_crypto <- factor(df_merged$risk_crypto, ordered = TRUE,levels=usage_levels)

df_merged$risk_investments <- factor(df_merged$risk_investments, ordered = TRUE,levels=usage_levels)

df_merged$risk_noRefund <- factor(df_merged$risk_noRefund, ordered = TRUE,levels=usage_levels)

df_merged$risk_onlineTasks <- factor(df_merged$risk_onlineTasks, ordered = TRUE,levels=usage_levels)

df_merged$risk_rebate <- factor(df_merged$risk_rebate, ordered = TRUE,levels=usage_levels)

df_merged$risk_usedSpotify <- factor(df_merged$risk_usedSpotify, ordered = TRUE,levels=usage_levels)

df_merged$risk_playedRoblox <- factor(df_merged$risk_playedRoblox, ordered = TRUE,levels=usage_levels)

df_merged$often_onlinetask <- factor(df_merged$often_onlinetask, ordered = TRUE, levels=usage_levels)

df_merged$risk_usedService <- factor(df_merged$risk_usedService, ordered = TRUE,levels=usage_levels)

# Ranking
df_merged$rank_legit <- factor(df_merged$rank_legit, levels=rank_scam_levels, ordered=TRUE)
df_merged$rank_scam <- factor(df_merged$rank_scam, levels=rank_scam_levels, ordered=TRUE)

# income
df_merged$income <- factor(df_merged$income, ordered = TRUE, levels = income_levels)

# Binary binary_gender
df_merged$binary_gender <- factor(df_merged$gender, levels = c("Male", "Female"))
```

The following code creates data frames for adult and teen data separately ("df_merged_adult`and`df_merged_teen"').

```
df_merged_adult <- df_merged %>% filter(adult == TRUE)
df_merged_teen <- df_merged %>% filter(adult == FALSE)
```

The following code defines the `stat_test` function. This function performs statistical testing between the variables listed in table 1 of the paper and the variable specified by the label in `dep_var`. The function returns the results in the form of a data frame containing the name of the independent variable, the name of the dependent variable, the name of the test which was run, and the appropriate effect size measure.

The parameters for the function are as follows:

1. The parameter `dep_var` specifies the dependent variable which we want to test. If the column specified by `dep_var` is one of the possible independent variables, the test for that indepndent variable is skipped.

2. The variable `condition_type` specifies if the dependent variable should be tested based on which `"legit"` stimuli the participant saw or which `"scam"` stimuli the participant saw. For example, when testing for differences in participants' actions with respect to scam websites, it only makes sense to test for significant differences between the different scam stimuli shown. If no comparison based on condition is necessary, the `test_condition` variable can be set to FALSE. The default value for `condition_type` is `"scam"` and the default value for `test_condition` is `TRUE`.

3. The parameter `df` specifies which dataframe should be used to perform the statistical test. The default value is `df_merged`.

4. The variable `stimuli_type` specifies whether or not comparisons should be restricted to only participants who saw particular type of stimuli (i.e., Roblox or Spotify related). Setting the value to `"roblox"` will only perform statistical tests with participants who saw Roblox stimuli. Setting the value to `"spotify"` wil only perform statistical tests with participants who saw Spotify stimuli. The default value, `"both"`, performs testing with the entire sample.

5. The variables `fisher_B` and `fisher_simulate_p` are passed through to the `fisher.test` parameters `B` and `simulate.p.value`. These variables are used to enable simulating Fisher's test using a Monte Carlo simulation with 10,000 replications. This is necessary due to the computational in feasibility of running Fisher's test on some larger contingency tables. See the documentation of "fisher.test" for more details.

```
stat_test <- function(dep_var, condition_type = "scam", df=df_merged, test_condition = TRUE, stimuli_ty
  raw_p = c()
  adjusted_p = c()
  indep_var_list = c()
  dep_var_list = c()
  test_list = c()
  effect_list = c()

  # Run fisher test dep_var v adult
  if(dep_var != "adult"){
    vs_adult <- fisher.test(table(df[[dep_var]], df$adult), B=fisher_B, simulate.p.value = fisher_simula
    raw_p = c(raw_p, vs_adult$p.value)
    indep_var_list = c(indep_var_list, "adult")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "fisher")
    effect_list = c(effect_list,cramer_v(table(df[[dep_var]], df$adult)))
  }
  # Run fisher test dep_var v  condition
  if(dep_var != "scam_condition" && dep_var != "legit_condition" && test_condition){
    if (condition_type == "scam"){
      vs_condition <- fisher.test(table(df[[dep_var]], df$scam_condition), B=fisher_B, simulate.p.value
      indep_var_list = c(indep_var_list, "scam_condition")
      effect_list = c(effect_list,cramer_v(table(df[[dep_var]], df$scam_condition)))
    }else if (condition_type == "legit"){
      vs_condition <- fisher.test(table(df[[dep_var]], df$legit_condition), B=fisher_B, simulate.p.valu
      indep_var_list = c(indep_var_list, "legit_condition")
      effect_list = c(effect_list,cramer_v(table(df[[dep_var]], df$legit_condition)))
    }
    if (condition_type == "scam" || condition_type == "legit"){
      raw_p = c(raw_p, vs_condition$p.value)
      dep_var_list = c(dep_var_list, dep_var)
```

```
      test_list = c(test_list, "fisher")
    }
  }
  if (dep_var != "time_overall"){
    # Run fisher test dep_var v time_overall
    vs_time_overall <- CochranArmitageTest(table(df[[dep_var]], df$time_overall))
    raw_p = c(raw_p, vs_time_overall$p.value)
    indep_var_list = c(indep_var_list, "time_overall")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$time_overall)))
  }

  # Run fisher test dep_var v time_computer
  if(dep_var != "time_videos"){
    vs_time_videos <- CochranArmitageTest(table(df[[dep_var]], df$time_videos))
    raw_p = c(raw_p, vs_time_videos$p.value)
    indep_var_list = c(indep_var_list, "time_videos")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$time_videos)))
  }
  # Run fisher test dep_var v time_mobile
  if(dep_var != "time_mobile"){
    vs_time_mobile <- CochranArmitageTest(table(df[[dep_var]], df$time_mobile))
    raw_p = c(raw_p, vs_time_mobile$p.value)
    indep_var_list = c(indep_var_list, "time_mobile")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$time_mobile)))
  }

  # Run fisher test dep_var v time_computer
  if(dep_var != "time_computer"){
    vs_time_computer <- CochranArmitageTest(table(df[[dep_var]], df$time_computer))
    raw_p = c(raw_p, vs_time_computer$p.value)
    indep_var_list = c(indep_var_list, "time_computer")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$time_computer)))
  }
  # Run fisher test dep_var v time_social
  if(dep_var != "time_social"){
    vs_time_social <- CochranArmitageTest(table(df[[dep_var]], df$time_social))
    raw_p = c(raw_p, vs_time_social$p.value)
    indep_var_list = c(indep_var_list, "time_social")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$time_social)))
  }
  # Run fisher test dep_var v time_nonsocial
  if(dep_var != "time_nonsocial"){
    vs_time_nonsocial <- CochranArmitageTest(table(df[[dep_var]], df$time_nonsocial))
```

```
  raw_p = c(raw_p, vs_time_nonsocial$p.value)
  indep_var_list = c(indep_var_list, "time_nonsocial")
  dep_var_list = c(dep_var_list, dep_var)
  test_list = c(test_list, "CochranArmitageTest")
  effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$time_nonsocial)))
}
# Run fisher test dep_var v risk_playedRoblox
if(dep_var != "risk_playedRoblox" && (stimuli_tyoe == "both" || stimuli_tyoe == "roblox")){
  vs_risk_playedRoblox <- fisher.test(table(df[[dep_var]], df$risk_playedRoblox))
  raw_p = c(raw_p, vs_risk_playedRoblox$p.value)
  indep_var_list = c(indep_var_list, "risk_playedRoblox")
  dep_var_list = c(dep_var_list, dep_var)
  test_list = c(test_list,"CochranArmitageTest")
  effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$risk_playedRoblox)))
}
# Run fisher test dep_var v risk_usedSpotify
if(dep_var != "risk_usedSpotify" && (stimuli_tyoe == "both" || stimuli_tyoe == "spotify")){
  vs_risk_usedSpotify <- fisher.test(table(df[[dep_var]], df$risk_usedSpotify))
  raw_p = c(raw_p, vs_risk_usedSpotify$p.value)
  indep_var_list = c(indep_var_list, "risk_usedSpotify")
  dep_var_list = c(dep_var_list, dep_var)
  test_list = c(test_list,"CochranArmitageTest")
  effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$risk_usedSpotify)))
}
# Run fisher test dep_var v risk_SpotifyPremium
if(dep_var != "risk_spotifyPremium" && (stimuli_tyoe == "both" || stimuli_tyoe == "spotify")){
  vs_risk_purchased <- fisher.test(table(df[[dep_var]], df$risk_spotifyPremium), B=fisher_B, simulate
  raw_p = c(raw_p, vs_risk_purchased$p.value)
  indep_var_list = c(indep_var_list, "risk_spotifyPremium")
  dep_var_list = c(dep_var_list, dep_var)
  test_list = c(test_list, "fisher")
  effect_list = c(effect_list,cramer_v(table(df[[dep_var]], df$risk_spotifyPremium)))
}
if(dep_var != "risk_puchasedRobux" && (stimuli_tyoe == "both" || stimuli_tyoe == "roblox")){
  vs_risk_purchased <- fisher.test(table(df[[dep_var]], df$risk_puchasedRobux), B=fisher_B, simulate.p
  raw_p = c(raw_p, vs_risk_purchased$p.value)
  indep_var_list = c(indep_var_list, "risk_puchasedRobux")
  dep_var_list = c(dep_var_list, dep_var)
  test_list = c(test_list, "fisher")
  effect_list = c(effect_list,cramer_v(table(df[[dep_var]], df$risk_puchasedRobux)))
}

if(dep_var != "risk_freeSpotify" && (stimuli_tyoe == "both" || stimuli_tyoe == "spotify")){
  vs_risk_freePremium <- fisher.test(table(df[[dep_var]], df$risk_freeSpotify), B=fisher_B, simulate.p
  raw_p = c(raw_p, vs_risk_freePremium$p.value)
  indep_var_list = c(indep_var_list, "risk_freeSpotify")
  dep_var_list = c(dep_var_list, dep_var)
  test_list = c(test_list, "fisher")
  effect_list = c(effect_list,cramer_v(table(df[[dep_var]], df$risk_freeSpotify)))
}
if(dep_var != "risk_freeRobux" && (stimuli_tyoe == "both" || stimuli_tyoe == "roblox")){
  vs_risk_freePremium <- fisher.test(table(df[[dep_var]], df$risk_freeRobux), B=fisher_B, simulate.p.
  raw_p = c(raw_p, vs_risk_freePremium$p.value)
```

```
    indep_var_list = c(indep_var_list, "risk_freeRobux")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "fisher")
    effect_list = c(effect_list,cramer_v(table(df[[dep_var]], df$risk_freeRobux)))
  }
  # Run fisher test dep_var v risk_noRefund
  if(dep_var != "risk_noRefund"){
    vs_risk_noRefund <- CochranArmitageTest(table(df[[dep_var]], df$risk_noRefund))
    raw_p = c(raw_p, vs_risk_noRefund$p.value)
    indep_var_list = c(indep_var_list, "risk_noRefund")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$risk_noRefund)))
  }
  # Run fisher test dep_var v risk_onlineTasks
  if(dep_var != "risk_onlineTasks"){
    vs_risk_onlineTasks <- CochranArmitageTest(table(df[[dep_var]], df$risk_onlineTasks))
    raw_p = c(raw_p, vs_risk_onlineTasks$p.value)
    indep_var_list = c(indep_var_list, "risk_onlineTasks")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$risk_onlineTasks)))
  }
  # Run fisher test dep_var v risk_crypto
  if(dep_var != "risk_crypto"){
    vs_risk_crypto <- CochranArmitageTest(table(df[[dep_var]], df$risk_crypto))
    raw_p = c(raw_p, vs_risk_crypto$p.value)
    indep_var_list = c(indep_var_list, "risk_crypto")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$risk_crypto)))
  }
  # Run fisher test dep_var v often_onlinetask
  if(dep_var != "often_onlinetask"){
    vs_often_onlinetask <- CochranArmitageTest(table(df[[dep_var]], df$often_onlinetask))
    raw_p = c(raw_p, vs_often_onlinetask$p.value)
    indep_var_list = c(indep_var_list, "often_onlinetask")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$often_onlinetask)))
  }
  # Run fisher test dep_var v income
  if(dep_var != "income"){
    vs_income <- CochranArmitageTest(table(df[[dep_var]], df$income))
    raw_p = c(raw_p, vs_income$p.value)
    indep_var_list = c(indep_var_list, "income")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "CochranArmitageTest")
    effect_list = c(effect_list,freemanTheta(table(df[[dep_var]], df$income)))
  }
  # Run fisher test dep_var v binary_gender
  if(dep_var != "binary_gender"){
    vs_binary_gender <- fisher.test(table(df[[dep_var]], df$binary_gender), B=fisher_B, simulate.p.valu
```

```r
    raw_p = c(raw_p, vs_binary_gender$p.value)
    indep_var_list = c(indep_var_list, "binary_gender")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "fisher")
    effect_list = c(effect_list,cramer_v(table(df[[dep_var]], df$binary_gender)))
  }
  # Run fisher test dep_var v community_type
  if(dep_var != "community_type"){
    vs_community_type <- fisher.test(table(df[[dep_var]], df$community_type), B=fisher_B, simulate.p.val
    raw_p = c(raw_p, vs_community_type$p.value)
    indep_var_list = c(indep_var_list, "community_type")
    dep_var_list = c(dep_var_list, dep_var)
    test_list = c(test_list, "fisher")
    effect_list = c(effect_list,cramer_v(table(df[[dep_var]], df$community_type)))
  }
  # adjust p values
  adjusted_p = p.adjust(raw_p, method = p.adjust.method)

  # compute significant results
  significant_list = c()
  for(i in adjusted_p){
    if (i < 0.05){
      significant_list = c(significant_list, TRUE)
    }else{
      significant_list = c(significant_list, FALSE)
    }
  }
  # return
  output_df = data.frame(independent=indep_var_list, dependent=dep_var_list, test=test_list, p.adjusted=
  return(output_df)
}
```

Descriptive statistics

The following code blocks produce tables describing the gender, community type, state, household income, and age for the entire sample and the teen/adult samples independently.

```r
print("Overall gender distribution")
```

```
## [1] "Overall gender distribution"
```

```r
prop.table(table(df_merged$gender, useNA = "always"))
```

```
##
##            Female              Male        Non-binary Prefer not to say
##       0.465517241       0.510344828       0.017241379       0.006896552
##              <NA>
##       0.000000000
```

```r
print("Adult gender distribution")
```

```
## [1] "Adult gender distribution"
```

```r
prop.table(table(df_merged_adult$gender, useNA = "always"))
```

```
##
```

```
##            Female              Male        Non-binary Prefer not to say
##       0.478048780       0.497560976       0.019512195       0.004878049
##              <NA>
##       0.000000000
```

```r
print("Teen gender distribution")
```

```
## [1] "Teen gender distribution"
```

```r
prop.table(table(df_merged_teen$gender, useNA = "always"))
```

```
##
##            Female              Male        Non-binary Prefer not to say
##        0.43529412        0.54117647        0.01176471        0.01176471
##              <NA>
##        0.00000000
```

```r
print("Overall community type distribution")
```

```
## [1] "Overall community type distribution"
```

```r
prop.table(table(df_merged$community_type, useNA = "always"))
```

```
##
##      Rural   Suburban      Urban       <NA>
## 0.15172414 0.51379310 0.31724138 0.01724138
```

```r
print("Adult community type distribution")
```

```
## [1] "Adult community type distribution"
```

```r
prop.table(table(df_merged_adult$community_type, useNA = "always"))
```

```
##
##     Rural  Suburban     Urban      <NA>
## 0.1853659 0.5121951 0.3024390 0.0000000
```

```r
print("Teen community type distribution")
```

```
## [1] "Teen community type distribution"
```

```r
prop.table(table(df_merged_teen$community_type, useNA = "always"))
```

```
##
##      Rural   Suburban      Urban       <NA>
## 0.07058824 0.51764706 0.35294118 0.05882353
```

```r
print("Overall state distribution")
```

```
## [1] "Overall state distribution"
```

```r
sort(prop.table(table(df_merged$state, useNA = "always")))
```

```
##
##             Arkansas          Connecticut District of Columbia
##          0.003448276          0.003448276          0.003448276
##                Maine              Montana             Nebraska
##          0.003448276          0.003448276          0.003448276
## Prefer not to answer         South Dakota        West Virginia
##          0.003448276          0.003448276          0.003448276
##             Delaware               Kansas             Kentucky
```

```
##          0.006896552          0.006896552          0.006896552
##        Massachusetts         Rhode Island                 Utah
##          0.006896552          0.006896552          0.006896552
##              Arizona             Colorado                 Iowa
##          0.010344828          0.010344828          0.010344828
##            Louisiana        New Hampshire             Missouri
##          0.010344828          0.010344828          0.013793103
##               Nevada             Oklahoma             Maryland
##          0.013793103          0.013793103          0.017241379
##           New Jersey               Oregon       South Carolina
##          0.017241379          0.017241379          0.017241379
##            Tennessee              Alabama              Indiana
##          0.017241379          0.020689655          0.020689655
##            Wisconsin                 <NA>             Michigan
##          0.020689655          0.020689655          0.027586207
##              Georgia                 Ohio             Virginia
##          0.031034483          0.031034483          0.037931034
##       North Carolina           Washington             Illinois
##          0.044827586          0.044827586          0.051724138
##             New York         Pennsylvania                Texas
##          0.055172414          0.055172414          0.068965517
##              Florida           California
##          0.093103448          0.124137931
```

```r
print("Adult state distribution")
```

```
## [1] "Adult state distribution"
```

```r
sort(prop.table(table(df_merged_adult$state, useNA = "always")))
```

```
## 
##              Arizona             Arkansas          Connecticut
##          0.004878049          0.004878049          0.004878049
##               Kansas                Maine              Montana
##          0.004878049          0.004878049          0.004878049
##             Nebraska Prefer not to answer         South Dakota
##          0.004878049          0.004878049          0.004878049
##                 <NA>             Colorado             Delaware
##          0.004878049          0.009756098          0.009756098
##             Kentucky        Massachusetts             Oklahoma
##          0.009756098          0.009756098          0.009756098
##         Rhode Island                 Utah                 Iowa
##          0.009756098          0.009756098          0.014634146
##            Louisiana               Nevada        New Hampshire
##          0.014634146          0.014634146          0.014634146
##             Missouri           New Jersey               Oregon
##          0.019512195          0.019512195          0.019512195
##            Tennessee           Washington              Alabama
##          0.019512195          0.019512195          0.024390244
##             Maryland         Pennsylvania            Wisconsin
##          0.024390244          0.024390244          0.024390244
##              Indiana             Michigan              Georgia
##          0.029268293          0.029268293          0.039024390
##             Illinois       North Carolina                 Ohio
##          0.039024390          0.039024390          0.043902439
```

```
##           Virginia            New York               Texas
##        0.043902439         0.073170732         0.082926829
##            Florida          California
##        0.087804878         0.121951220
```

```r
print("Teen state distribution")
```

```
## [1] "Teen state distribution"
```

```r
sort(prop.table(table(df_merged_teen$state, useNA = "always")))
```

```
##
##              Alabama            Colorado District of Columbia
##           0.01176471          0.01176471          0.01176471
##              Georgia              Kansas               Nevada
##           0.01176471          0.01176471          0.01176471
##           New Jersey            New York               Oregon
##           0.01176471          0.01176471          0.01176471
##            Tennessee       West Virginia            Wisconsin
##           0.01176471          0.01176471          0.01176471
##              Arizona            Michigan             Oklahoma
##           0.02352941          0.02352941          0.02352941
##             Virginia               Texas       North Carolina
##           0.02352941          0.03529412          0.05882353
##       South Carolina                <NA>             Illinois
##           0.05882353          0.05882353          0.08235294
##              Florida          Washington           California
##           0.10588235          0.10588235          0.12941176
##         Pennsylvania
##           0.12941176
```

```r
print("Overall income distribution")
```

```
## [1] "Overall income distribution"
```

```r
prop.table(table(df_merged$income, useNA = "always"))
```

```
##
##    Less than $20,000   $20,000 to $39,999   $40,000 to $59,999
##           0.07931034          0.16551724           0.17931034
##   $60,000 to $79,999   $80,000 to $99,999 $100,000 to $149,999
##           0.12758621          0.08275862           0.15862069
##        Over $150,000                 <NA>
##           0.12413793          0.08275862
```

```r
print("Adult income distribution")
```

```
## [1] "Adult income distribution"
```

```r
prop.table(table(df_merged_adult$income, useNA = "always"))
```

```
##
##    Less than $20,000   $20,000 to $39,999   $40,000 to $59,999
##           0.09756098          0.18536585           0.17560976
##   $60,000 to $79,999   $80,000 to $99,999 $100,000 to $149,999
##           0.16097561          0.09268293           0.15609756
##        Over $150,000                 <NA>
##           0.10243902          0.02926829
```

```r
print("Teen income distribution")
```

```
## [1] "Teen income distribution"
```

```r
prop.table(table(df_merged_teen$income, useNA = "always"))
```

```
##
##    Less than $20,000   $20,000 to $39,999   $40,000 to $59,999
##           0.03529412           0.11764706           0.18823529
##   $60,000 to $79,999   $80,000 to $99,999 $100,000 to $149,999
##           0.04705882           0.05882353           0.16470588
##       Over $150,000                 <NA>
##           0.17647059           0.21176471
```

```r
df_merged_adult <- df_merged_adult %>% mutate(age_cat = case_when(
                         age >= 18 & age <= 24 ~ "18 to 24",
                         age >= 25 & age <= 34 ~ "25 to 34",
                         age >= 35 & age <= 44 ~ "35 to 44",
                         age >= 45 & age <= 54 ~ "45 to 54",
                         age >= 55 ~ "55+",
                         TRUE ~ NA_character_))
print("Adult age distribution")
```

```
## [1] "Adult age distribution"
```

```r
prop.table(table(df_merged_adult$age_cat))
```

```
##
##  18 to 24  25 to 34  35 to 44  45 to 54       55+
## 0.1512195 0.2048780 0.2195122 0.1609756 0.2634146
```

```r
print("Teen age distribution")
```

```
## [1] "Teen age distribution"
```

```r
prop.table(table(df_merged_teen$age))
```

```
##
##        13        14        15        16        17
## 0.1058824 0.1882353 0.3176471 0.2000000 0.1882353
```

The following code blocks produce tables describing the various aspects of behavior asked about at the beginning of the survey for the entire sample and the teen/adult samples independently. This includes devices used in the last week, social media services used in the last week, hours per day spent on digital entertainment, etc.

```r
# unlist(str_split(*)) splits comma separated values
print("Overall social media services used in the last week distribution")
```

```
## [1] "Overall social media services used in the last week distribution"
```

```r
table(unlist(str_split(df_merged$social_media, ",")))/nrow(df_merged)
```

```
##
##                BeReal                Discord               Facebook
##            0.04482759             0.30000000             0.54482759
##             Instagram Other (please specify)              Pinterest
##            0.65172414             0.02758621             0.28275862
##                Reddit               Snapchat                 TikTok
```

```
##             0.52413793             0.30689655             0.50344828
##                Twitter                YouTube
##             0.37586207             1.00000000
```

```r
print("Teens social media services used in the last week distribution")
```

```
## [1] "Teens social media services used in the last week distribution"
```

```r
table(unlist(str_split(df_merged_teen$social_media, ",")))/nrow(df_merged_teen)
```

```
##
##                 BeReal                Discord               Facebook
##             0.12941176             0.30588235             0.31764706
##              Instagram Other (please specify)              Pinterest
##             0.63529412             0.01176471             0.31764706
##                 Reddit               Snapchat                 TikTok
##             0.22352941             0.45882353             0.61176471
##                Twitter                YouTube
##             0.23529412             1.00000000
```

```r
print("Adults social media services used in the last week distribution ")
```

```
## [1] "Adults social media services used in the last week distribution "
```

```r
table(unlist(str_split(df_merged_adult$social_media, ",")))/nrow(df_merged_adult)
```

```
##
##                 BeReal                Discord               Facebook
##            0.009756098            0.297560976            0.639024390
##              Instagram Other (please specify)              Pinterest
##            0.658536585            0.034146341            0.268292683
##                 Reddit               Snapchat                 TikTok
##            0.648780488            0.243902439            0.458536585
##                Twitter                YouTube
##            0.434146341            1.000000000
```

```r
print("Overall devices used in the last week distribution")
```

```
## [1] "Overall devices used in the last week distribution"
```

```r
table(trimws(unlist(str_split(df_merged$devices, ","))))/nrow(df_merged)
```

```
##
##                Game console Laptop or desktop computer
##                 0.344827586                0.806896552
##      Other (please specify)                 Smartphone
##                 0.006896552                0.941379310
##                  Smartwatch                     Tablet
##                 0.106896552                0.355172414
##                  Television    Virtual Reality Devices
##                 0.693103448                0.037931034
```

```r
print("Teens devices used in the last week distribution")
```

```
## [1] "Teens devices used in the last week distribution"
```

```r
table(trimws(unlist(str_split(df_merged_teen$devices, ","))))/nrow(df_merged_teen)
```

```
##
##                Game console Laptop or desktop computer
```

```
##              0.35294118                 0.62352941
##              Smartphone                 Smartwatch
##              0.88235294                 0.11764706
##                  Tablet                 Television
##              0.43529412                 0.63529412
##    Virtual Reality Devices
##              0.07058824
```

```r
print("Adults devices used in the last week distribution")
```

```
## [1] "Adults devices used in the last week distribution"
```

```r
table(trimws(unlist(str_split(df_merged_adult$devices, ","))))/nrow(df_merged_adult)
```

```
##
##            Game console Laptop or desktop computer
##             0.341463415                0.882926829
##     Other (please specify)              Smartphone
##             0.009756098                0.965853659
##              Smartwatch                     Tablet
##             0.102439024                0.321951220
##              Television    Virtual Reality Devices
##             0.717073171                0.024390244
```

```r
print("Overall distribution of hours per day on digital entertainment")
```

```
## [1] "Overall distribution of hours per day on digital entertainment"
```

```r
prop.table(table(df_merged$time_overall, useNA = "always"))
```

```
##
## Less than 2 hours      2 to 4 hours      4 to 8 hours More than 8 hours
##        0.06206897        0.40689655        0.40000000        0.13103448
##              <NA>
##        0.00000000
```

```r
print("Adult distribution of hours per day on digital entertainment")
```

```
## [1] "Adult distribution of hours per day on digital entertainment"
```

```r
prop.table(table(df_merged_adult$time_overall, useNA = "always"))
```

```
##
## Less than 2 hours      2 to 4 hours      4 to 8 hours More than 8 hours
##        0.05853659        0.42926829        0.37560976        0.13658537
##              <NA>
##        0.00000000
```

```r
print("Teen distribution of hours per day on digital entertainment")
```

```
## [1] "Teen distribution of hours per day on digital entertainment"
```

```r
prop.table(table(df_merged_teen$time_overall, useNA = "always"))
```

```
##
## Less than 2 hours      2 to 4 hours      4 to 8 hours More than 8 hours
##        0.07058824        0.35294118        0.45882353        0.11764706
##              <NA>
##        0.00000000
```

```r
print("Overall distribution of time on watching online videos per week")
```

```
## [1] "Overall distribution of time on watching online videos per week"
```

```r
prop.table(table(df_merged$time_videos, useNA = "always"))
```

```
##
##               None  Less than an hour       1 to 5 hours      5 to 10 hours
##        0.000000000        0.034482759        0.313793103        0.265517241
##     10 to 15 hours     15 to 20 hours More than 20 hours               <NA>
##        0.151724138        0.096551724        0.134482759        0.003448276
```

```r
print("Adult distribution of time on watching online videos per week")
```

```
## [1] "Adult distribution of time on watching online videos per week"
```

```r
prop.table(table(df_merged_adult$time_videos, useNA = "always"))
```

```
##
##               None  Less than an hour       1 to 5 hours      5 to 10 hours
##         0.00000000         0.04878049         0.34146341         0.25853659
##     10 to 15 hours     15 to 20 hours More than 20 hours               <NA>
##         0.12195122         0.09268293         0.13658537         0.00000000
```

```r
print("Teen distribution of time on watching online videos per week")
```

```
## [1] "Teen distribution of time on watching online videos per week"
```

```r
prop.table(table(df_merged_teen$time_videos, useNA = "always"))
```

```
##
##               None  Less than an hour       1 to 5 hours      5 to 10 hours
##         0.00000000         0.00000000         0.24705882         0.28235294
##     10 to 15 hours     15 to 20 hours More than 20 hours               <NA>
##         0.22352941         0.10588235         0.12941176         0.01176471
```

```r
print("Overall distribution of frequency of playing robux")
```

```
## [1] "Overall distribution of frequency of playing robux"
```

```r
prop.table(table(df_merged$risk_playedRoblox))
```

```
##
##                Never        Once or twice  Three to five times
##           0.57142857           0.20000000           0.06428571
## More than five times
##           0.16428571
```

```r
print("Adult distribution of frequency of playing robux")
```

```
## [1] "Adult distribution of frequency of playing robux"
```

```r
prop.table(table(df_merged_adult$risk_playedRoblox))
```

```
##
##                Never        Once or twice  Three to five times
##           0.72916667           0.15625000           0.04166667
## More than five times
##           0.07291667
```

```r
print("Teen distribution of frequency of playing robux")
```

```
## [1] "Teen distribution of frequency of playing robux"
```

```r
prop.table(table(df_merged_teen$risk_playedRoblox))
```

```
##
##                Never        Once or twice  Three to five times
##            0.2272727            0.2954545            0.1136364
## More than five times
##            0.3636364
```

```r
print("Overall distribution of frequency of using Spotify")
```

```
## [1] "Overall distribution of frequency of using Spotify"
```

```r
prop.table(table(df_merged$risk_usedSpotify))
```

```
##
##                Never        Once or twice  Three to five times
##           0.08000000           0.24000000           0.09333333
## More than five times
##           0.58666667
```

```r
print("Adult distribution of frequency of using Spotify")
```

```
## [1] "Adult distribution of frequency of using Spotify"
```

```r
prop.table(table(df_merged_adult$risk_usedSpotify))
```

```
##
##                Never        Once or twice  Three to five times
##           0.11009174           0.20183486           0.05504587
## More than five times
##           0.63302752
```

```r
print("Teen distribution of frequency of using Spotify")
```

```
## [1] "Teen distribution of frequency of using Spotify"
```

```r
prop.table(table(df_merged_teen$risk_usedSpotify))
```

```
##
##                Never        Once or twice  Three to five times
##            0.0000000            0.3414634            0.1951220
## More than five times
##            0.4634146
```

Statistical testing

Adult vs Teen comparisons

The following code tests which potential independent variables are associated with whether a participant is a teen or an adult

```r
adult_results <- stat_test(dep_var ="adult", condition_type = "scam")
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Paper Result: The following code prints results that are significantly associated with being an adult/teen. These results are discussed in section 5A of the paper

```
adult_results %>% filter(significant == TRUE)
```

```
##          independent dependent                 test   p.adjusted        p.raw
## 1        time_mobile     adult CochranArmitageTest 3.439816e-02 1.375926e-02
## 2  risk_playedRoblox     adult CochranArmitageTest 5.037379e-07 5.037379e-08
## 3    risk_usedSpotify    adult CochranArmitageTest 4.488214e-03 1.570875e-03
## 4 risk_puchasedRobux     adult               fisher 4.488214e-03 1.546996e-03
## 5      risk_noRefund     adult CochranArmitageTest 2.802389e-03 7.005972e-04
## 6   risk_onlineTasks     adult CochranArmitageTest 2.235614e-35 1.117807e-36
## 7        risk_crypto     adult CochranArmitageTest 4.754589e-04 9.509178e-05
## 8   often_onlinetask     adult CochranArmitageTest 2.745084e-06 4.117626e-07
## 9             income     adult CochranArmitageTest 4.825138e-02 2.171312e-02
##   effect_size significant
## 1    0.240000        TRUE
## 2    0.549000        TRUE
## 3    0.090000        TRUE
## 4    0.265165        TRUE
## 5    0.222000        TRUE
## 6    0.824000        TRUE
## 7    0.203000        TRUE
## 8    0.386000        TRUE
## 9    0.178000        TRUE
```

The following code prints cross tabs for significant variables. Each row represents either Teens or Adults. Each column represents a value of the tested variable. For the rows: 0 = teens, 1 = adults.

```
print("Age vs.  Frequency of use of Roblox")
```

```
## [1] "Age vs.  Frequency of use of Roblox"
```

```
print(prop.table(table(df_merged$adult, df_merged$risk_playedRoblox),1))
```

```
##
##          Never Once or twice Three to five times More than five times
##   0 0.22727273    0.29545455          0.11363636           0.36363636
##   1 0.72916667    0.15625000          0.04166667           0.07291667
```

```
print("Age vs. Freuqnecy of use of Spotify")
```

```
## [1] "Age vs. Freuqnecy of use of Spotify"
```

```
print(prop.table(table(df_merged$adult, df_merged$risk_usedSpotify),1))
```

```
##
##          Never Once or twice Three to five times More than five times
##   0 0.00000000    0.34146341          0.19512195           0.46341463
##   1 0.11009174    0.20183486          0.05504587           0.63302752
```

```
print("Age vs. Purchased Roblox")
```

```
## [1] "Age vs. Purchased Roblox"
```

```
print(prop.table(table(df_merged$adult, df_merged$risk_puchasedRobux),1))
```

```
##
##             0         1
##   0 0.6136364 0.3863636
```

```
##   1 0.8645833 0.1354167
```

```r
print("Age vs. Frequency of playing mobile phone games")
```

```
## [1] "Age vs. Frequency of playing mobile phone games"
```

```r
prop.table(table(df_merged$adult, df_merged$time_mobile),1)
```

```
##
##           None Less than an hour 1 to 5 hours 5 to 10 hours 10 to 15 hours
##   0 0.08235294        0.22352941   0.37647059    0.25882353     0.03529412
##   1 0.27804878        0.25853659   0.23414634    0.12682927     0.05365854
##
##     15 to 20 hours More than 20 hours
##   0     0.01176471         0.01176471
##   1     0.02926829         0.01951220
```

```r
print("Age vs. Frequency of shopping with no refund")
```

```
## [1] "Age vs. Frequency of shopping with no refund"
```

```r
prop.table(table(df_merged$adult, df_merged$risk_noRefund),1)
```

```
##
##          Never Once or twice Three to five times More than five times
##   0 0.62352941    0.35294118          0.02352941           0.00000000
##   1 0.41951220    0.49756098          0.04390244           0.03902439
```

```r
print("Age vs. Frequency of doing online tasks for money")
```

```
## [1] "Age vs. Frequency of doing online tasks for money"
```

```r
prop.table(table(df_merged$adult, df_merged$risk_onlineTasks),1)
```

```
##
##          Never Once or twice Three to five times More than five times
##   0 0.35294118    0.48235294          0.09411765           0.07058824
##   1 0.01463415    0.09756098          0.05853659           0.82926829
```

```r
print("Age vs. Frequency of online tasks for money without being paid")
```

```
## [1] "Age vs. Frequency of online tasks for money without being paid"
```

```r
prop.table(table(df_merged$adult, df_merged$often_onlinetask),1)
```

```
##
##          Never Once or twice Three to five times More than five times
##   0 0.68235294    0.22352941          0.05882353           0.03529412
##   1 0.32195122    0.40975610          0.12195122           0.14634146
```

```r
print("Age vs. Frequency of purchasing crypto assets")
```

```
## [1] "Age vs. Frequency of purchasing crypto assets"
```

```r
prop.table(table(df_merged$adult, df_merged$risk_crypto),1)
```

```
##
##          Never Once or twice Three to five times More than five times
##   0 0.82352941    0.15294118          0.02352941           0.00000000
##   1 0.65196078    0.14215686          0.07843137           0.12745098
```

```r
# income
print("Age vs. Household Income")
```

```
## [1] "Age vs. Household Income"
```

```r
prop.table(table(df_merged$adult, df_merged$income, useNA = "always"),1)
```

```
##
##        Less than $20,000 $20,000 to $39,999 $40,000 to $59,999
##   0           0.03529412         0.11764706         0.18823529
##   1           0.09756098         0.18536585         0.17560976
##   <NA>
##
##        $60,000 to $79,999 $80,000 to $99,999 $100,000 to $149,999 Over $150,000
##   0            0.04705882         0.05882353           0.16470588    0.17647059
##   1            0.16097561         0.09268293           0.15609756    0.10243902
##   <NA>
##
##              <NA>
##   0    0.21176471
##   1    0.02926829
##   <NA>
```

Gender comparisons

The following code tests which potential independent variables are associated with binary gender. Sample size was insufficient to include non-binary individuals in this analysis. These comparisons are post-hoc.

```r
binary_gender_results <- stat_test(dep_var ="binary_gender", condition_type = "scam")
```

Paper Result: The following code prints results that are significantly associated with binary gender These results are discussed in section 5D of the paper

```r
binary_gender_results %>% filter(significant == TRUE)
```

```
##     independent     dependent                test   p.adjusted        p.raw
## 1   time_overall binary_gender CochranArmitageTest 1.610318e-02 2.415477e-03
## 2    time_videos binary_gender CochranArmitageTest 1.913492e-02 3.826983e-03
## 3  time_computer binary_gender CochranArmitageTest 2.524232e-09 1.262116e-10
## 4 time_nonsocial binary_gender CochranArmitageTest 2.273868e-02 5.684670e-03
## 5    risk_crypto binary_gender CochranArmitageTest 1.444870e-03 1.444870e-04
##   effect_size significant
## 1       0.200        TRUE
## 2       0.192        TRUE
## 3       0.419        TRUE
## 4       0.179        TRUE
## 5       0.215        TRUE
```

The following code prints cross tabs for significant variables. Each row represents either men or women. Each column represents a value of the tested variable.

```r
print("Gender vs.Time spent on digital entertainment")
```

```
## [1] "Gender vs.Time spent on digital entertainment"
```

```r
prop.table(table(df_merged$binary_gender, df_merged$time_overall))
```

```
##
```

```
##          Less than 2 hours 2 to 4 hours 4 to 8 hours More than 8 hours
##   Male          0.02826855   0.17314488   0.22968198        0.09187279
##   Female        0.03180212   0.23674912   0.16961131        0.03886926
```

```r
print("Gender vs.Time spent watching online videos per week")
```

```
## [1] "Gender vs.Time spent watching online videos per week"
```

```r
prop.table(table(df_merged$binary_gender, df_merged$time_videos), 1)
```

```
##
##                None Less than an hour 1 to 5 hours 5 to 10 hours 10 to 15 hours
##   Male   0.00000000        0.01351351   0.27702703    0.25675676     0.16891892
##   Female 0.00000000        0.05970149   0.35074627    0.28358209     0.14179104
##
##          15 to 20 hours More than 20 hours
##   Male       0.11486486         0.16891892
##   Female     0.05970149         0.10447761
```

```r
print("Gender vs.Time spent on computer/console games")
```

```
## [1] "Gender vs.Time spent on computer/console games"
```

```r
prop.table(table(df_merged$binary_gender, df_merged$time_computer),1)
```

```
##
##                 None Less than an hour 1 to 5 hours 5 to 10 hours
##   Male   0.189189189       0.121621622  0.216216216   0.189189189
##   Female 0.407407407       0.207407407  0.244444444   0.088888889
##
##          10 to 15 hours 15 to 20 hours More than 20 hours
##   Male      0.128378378    0.033783784        0.121621622
##   Female    0.037037037    0.007407407        0.007407407
```

```r
print("Gender vs.time spent on non-social media websites per week")
```

```
## [1] "Gender vs.time spent on non-social media websites per week"
```

```r
prop.table(table(df_merged$binary_gender, df_merged$time_nonsocial), 1)
```

```
##
##                 None Less than an hour 1 to 5 hours 5 to 10 hours
##   Male   0.013513514       0.189189189  0.493243243   0.155405405
##   Female 0.029629630       0.251851852  0.555555556   0.125925926
##
##          10 to 15 hours 15 to 20 hours More than 20 hours
##   Male      0.121621622    0.027027027        0.000000000
##   Female    0.007407407    0.014814815        0.014814815
```

```r
print("Gender vs.Frequency of purchasing crypto assets")
```

```
## [1] "Gender vs.Frequency of purchasing crypto assets"
```

```r
prop.table(table(df_merged$binary_gender, df_merged$risk_crypto), 1)
```

```
##
##               Never Once or twice Three to five times More than five times
##   Male   0.60544218    0.16326531          0.10204082           0.12925170
##   Female 0.80740741    0.11851852          0.02222222           0.05185185
```

Experience searching for Free Robux comparisons

The following code tests which potential independent variables are significantly associated with experience searching for "Free Roblox robux" or something similar

```
roblox_s_results <- stat_test("roblox_s", stimuli_tyoe = "roblox",condition_type = "neither")
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Paper Result: The following code prints results for variables that are significantly associated with experience previously searching for "Free Roblox robux" or something similar. These results are discussed in section 5C and 5D of the paper.

```
roblox_s_results %>% filter(significant == TRUE,)
```

```
##          independent dependent                test   p.adjusted        p.raw
## 1              adult  roblox_s              fisher 8.227889e-05 5.368282e-06
## 2  risk_playedRoblox  roblox_s CochranArmitageTest 8.227889e-05 1.451980e-05
## 3 risk_puchasedRobux  roblox_s              fisher 2.060745e-04 4.848811e-05
## 4     risk_freeRobux  roblox_s              fisher 9.460568e-03 2.782520e-03
## 5   risk_onlineTasks  roblox_s CochranArmitageTest 8.227889e-05 1.000558e-05
##   effect_size significant
## 1   0.3849957        TRUE
## 2   0.5450000        TRUE
## 3   0.3557067        TRUE
## 4   0.2645712        TRUE
## 5   0.5200000        TRUE
```

The following code prints cross tabs for significant variables. Each row represents people reported previously searching for "Free Roblox robux" or something similar vs. those who had not searched. 0 = those who had not searched, 1 = those who had searched.

```
print("Searching for Free Robux vs. Age")
```

```
## [1] "Searching for Free Robux vs. Age"
```

```
prop.table(table(df_merged$adult, df_merged$roblox_s),1)
```

```
##
##             0         1
##   0 0.6136364 0.3863636
##   1 0.9375000 0.0625000
```

```
print("Searching for Free Robux vs.Frequency of playing Roblox")
```

```
## [1] "Searching for Free Robux vs.Frequency of playing Roblox"
```

```
table(df_merged$roblox_s, df_merged$risk_playedRoblox)
```

```
##
##     Never Once or twice Three to five times More than five times
##   0    75            24                   7                   11
##   1     5             4                   2                   12
```

```r
# two times or fewer: 108 total, 9 say they searched = 9/108 = 8.3%
# Three times or more: 32 total, 14 say they searched 14/32 = 0.4375
```

```r
print("Searching for Free Robux vs.Purchasing Robux")
```

```
## [1] "Searching for Free Robux vs.Purchasing Robux"
```

```r
prop.table(table(df_merged$roblox_s, df_merged$risk_puchasedRobux),2)
```

```
##
##              0          1
##   0 0.90909091 0.56666667
##   1 0.09090909 0.43333333
```

```r
print("Searching for Free Robux vs.Receiving Free Robux ")
```

```
## [1] "Searching for Free Robux vs.Receiving Free Robux "
```

```r
prop.table(table(df_merged$roblox_s, df_merged$risk_freeRobux ),2)
```

```
##
##             0         1
##   0 0.8682171 0.4545455
##   1 0.1317829 0.5454545
```

```r
print("Searching for Free Robux vs.Frequency of doing online tasks")
```

```
## [1] "Searching for Free Robux vs.Frequency of doing online tasks"
```

```r
prop.table(table(df_merged$roblox_s, df_merged$risk_onlineTasks ),2)
```

```
##
##          Never Once or twice Three to five times More than five times
##   0 0.71428571    0.58823529          0.90000000           0.95121951
##   1 0.28571429    0.41176471          0.10000000           0.04878049
```

Experience searching for Free Spotify comparisons

The following code tests which potential independent variables are significantly associated with previously searching for "Free Spotify Premium" or something similar.

```r
spotify_s_results <- stat_test("spotify_s", stimuli_tyoe = "spotify", condition_type = "neither")
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Paper Result: The following code shows that none of the tested variables varied significantly

```r
spotify_s_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Free Robux search liklihood comparisons

The following code tests which potential independent variables are significantly associated with a liklihood of searching for "Free Roblox Robux" or something similar

```r
# Bin to likely vs unlikely
df_merged <- df_merged %>% mutate(rbolox_s_likleihood_bool = ifelse(roblox_s_likliehood == "Somewhat lil
roblox_s_likliehood_results <- stat_test("rbolox_s_likleihood_bool", stimuli_tyoe = "roblox")
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

The following code shows that none of the tested variables varied significantly

```r
roblox_s_likliehood_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Free Spotify search liklihood comparisons

The following code tests which potential independent variables are significantly associated with a likelihood of searching for "Free Spotify Premium" or something similar

```r
# Bin to likely vs unlikely
df_merged <- df_merged %>% mutate(spotify_s_liklihood_bool = ifelse(spotify_s_liklihood == "Somewhat lil
spotify_s_liklihood_results <- stat_test("spotify_s_liklihood_bool", stimuli_tyoe = "spotify")
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

The following code shows that none of the tested variables varied significantly

```r
spotify_s_liklihood_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Scam Ranking comparions

The following code tests which potential independent variables are significantly associated with correctly identifying the scam stimuli

```r
rank_scam_results <- stat_test(dep_var ="rank_scam_bool", condition_type = "scam")
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Paper Result: The following code prints results for variables that are significantly associated with correctly identifying scam stimuli. These results are discussed in section 5D

```r
rank_scam_results %>% filter(significant == TRUE)
```

```
##       independent      dependent                test  p.adjusted        p.raw
## 1 scam_condition rank_scam_bool              fisher 0.002099790 0.0000999900
## 2   time_overall rank_scam_bool CochranArmitageTest 0.003518548 0.0005026497
## 3  time_computer rank_scam_bool CochranArmitageTest 0.022079107 0.0042055442
## 4  binary_gender rank_scam_bool              fisher 0.003381953 0.0003220907
##   effect_size significant
## 1   0.3108626        TRUE
## 2   0.3220000        TRUE
## 3   0.2750000        TRUE
## 4   0.2230187        TRUE
```

The following code prints cross tables for significant results. The rows represent whether or not scam video was correctly identified. 0 = participant identified the video as legit or selected I don't know, 1 = participant identified the video as a scam.

```r
print("Correct scam identification vs. condition")
```

```
## [1] "Correct scam identification vs. condition"
```

```r
prop.table(table(df_merged$rank_scam_bool, df_merged$scam_condition), 2)
```

```
##
##            sr1        sr2        sr3        ss1        ss2        ss3
##   0 0.31707317 0.24324324 0.02500000 0.02173913 0.23255814 0.11627907
##   1 0.68292683 0.75675676 0.97500000 0.97826087 0.76744186 0.88372093
```

```r
print("Scam ranking vs. condition")
```

```
## [1] "Scam ranking vs. condition"
```

```r
prop.table(table(df_merged$rank_scam, df_merged$scam_condition ),2)
```

```
##
##                               sr1        sr2        sr3        ss1        ss2
##   Definitely legitimate 0.02439024 0.00000000 0.00000000 0.00000000 0.00000000
##   Probably legitimate   0.14634146 0.10810811 0.00000000 0.02173913 0.13953488
##   I'm not sure          0.14634146 0.13513514 0.02500000 0.00000000 0.09302326
##   Probably a scam       0.46341463 0.48648649 0.30000000 0.34782609 0.44186047
##   Definitely a scam     0.21951220 0.27027027 0.67500000 0.63043478 0.32558140
##
##                               ss3
##   Definitely legitimate 0.02325581
##   Probably legitimate   0.09302326
##   I'm not sure          0.00000000
##   Probably a scam       0.48837209
```

```
##   Definitely a scam     0.39534884
```

```r
print("Correct scam identification vs. gender")
```

```
## [1] "Correct scam identification vs. gender"
```

```r
prop.table(table(df_merged$rank_scam_bool, df_merged$binary_gender), 2)
```

```
##
##          Male    Female
##   0 0.0720000 0.2416667
##   1 0.9280000 0.7583333
```

```r
print("Scam ranking vs. gender")
```

```
## [1] "Scam ranking vs. gender"
```

```r
table(df_merged$rank_scam, df_merged$binary_gender)
```

```
##
##                         Male Female
##   Definitely legitimate    0      2
##   Probably legitimate      6     14
##   I'm not sure             3     13
##   Probably a scam         54     49
##   Definitely a scam       62     42
```

```r
# 29 women did not identify scam. 13 of these selected I'm not sure. 13/29 = 0.4482759
# 9 men did not identify scam. 3 of these selected I'm not sure. 3/9 = 0.333333
```

```r
print("Correct scam identification vs. time on computer/console games")
```

```
## [1] "Correct scam identification vs. time on computer/console games"
```

```r
prop.table(table(df_merged$time_computer, df_merged$rank_scam_bool),1)
```

```
##
##                               0          1
##   None               0.20512821 0.79487179
##   Less than an hour  0.30952381 0.69047619
##   1 to 5 hours       0.07142857 0.92857143
##   5 to 10 hours      0.08823529 0.91176471
##   10 to 15 hours     0.14285714 0.85714286
##   15 to 20 hours     0.00000000 1.00000000
##   More than 20 hours 0.00000000 1.00000000
```

Post-hoc comparisons **Paper Result** Gender is associated with amount of time spent daily on digital entertainment activities. The following code tests whether time spent on digital entertainment activities is a significant predictor of ranking success when controlling for gender. From this, we see that time remains a weakly significant predictor for women ($p_{uncorrected} < 0.006$, $\theta = 0.181$) but not men ($p_{uncorrected} = 0.156$)

```r
# bin to men and women separately
df_merged_men <- df_merged %>% filter(binary_gender == "Male")
df_merged_women <- df_merged %>% filter(binary_gender == "Female")
print("Test scam ranking vs. time overall with just men")
```

```
## [1] "Test scam ranking vs. time overall with just men"
```

```r
CochranArmitageTest(table(df_merged_men$time_overall, df_merged_men$rank_scam_bool))
```

```
##
##  Cochran-Armitage test for trend
##
## data:  table(df_merged_men$time_overall, df_merged_men$rank_scam_bool)
## Z = -1.4926, dim = 4, p-value = 0.1355
## alternative hypothesis: two.sided
```

```
print("Test scam ranking vs. time overall with just women")
```

```
## [1] "Test scam ranking vs. time overall with just women"
```

```
CochranArmitageTest(table(df_merged_women$time_overall, df_merged_women$rank_scam_bool))
```

```
##
##  Cochran-Armitage test for trend
##
## data:  table(df_merged_women$time_overall, df_merged_women$rank_scam_bool)
## Z = -2.7868, dim = 4, p-value = 0.005323
## alternative hypothesis: two.sided
```

```
freemanTheta(table(df_merged_women$time_overall, df_merged_women$rank_scam_bool))
```

```
## Freeman.theta
##         0.181
```

Paper Result Gender is associated with amount of time spent weekly on console/computer games. The following code tests whether time spent on console/comptuer games is a significant predictor of ranking success when controlling for gender. From this, we see that time remains a not a significant predictor for just women ($p_{uncorrected} = 0.159$) or just men($p_{uncorrected} = 0.275$)

```
# Check if difference exists with just men
print("Test scam ranking vs. time on console/computer games with just men")
```

```
## [1] "Test scam ranking vs. time on console/computer games with just men"
```

```
CochranArmitageTest(table(df_merged_men$time_computer, df_merged_men$rank_scam_bool))
```

```
##
##  Cochran-Armitage test for trend
##
## data:  table(df_merged_men$time_computer, df_merged_men$rank_scam_bool)
## Z = -1.0908, dim = 7, p-value = 0.2754
## alternative hypothesis: two.sided
```

```
print("Test scam ranking vs. time on console/computer games with just women")
```

```
## [1] "Test scam ranking vs. time on console/computer games with just women"
```

```
CochranArmitageTest(table(df_merged_women$time_computer, df_merged_women$rank_scam_bool))
```

```
##
##  Cochran-Armitage test for trend
##
## data:  table(df_merged_women$time_computer, df_merged_women$rank_scam_bool)
## Z = -1.4095, dim = 7, p-value = 0.1587
## alternative hypothesis: two.sided
```

Legit Ranking comparions

The following code tests which potential independent variables are significantly associated with correctly identifying the legit stimuli

```
rank_legit_results <- stat_test(dep_var ="rank_legit_bool", condition_type = "legit")
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

The following code prints results for variables that are significantly associated with correctly identifying legit stimuli

```
rank_legit_results %>% filter(significant == TRUE)
```

```
##        independent       dependent                test p.adjusted       p.raw
## 1  legit_condition rank_legit_bool              fisher 0.01049895 0.000499950
## 2 risk_onlineTasks rank_legit_bool CochranArmitageTest 0.01344661 0.001280629
##   effect_size significant
## 1   0.2954397        TRUE
## 2   0.1880000        TRUE
```

The following code prints cross tables for significant results. The rows represent whether or not legit video was correctly identified. 0 = participant identified the video as a scam or selected I don't know, 1 = participant identified the video as legit.

```
print("Correct legit identification vs. condition")
```

```
## [1] "Correct legit identification vs. condition"
```

```
prop.table(table(df_merged$rank_legit_bool,df_merged$legit_condition),2)
```

```
##
##           lr1       lr2       lr3       ls1       ls2       ls3
##   0 0.2777778 0.3750000 0.6190476 0.3255814 0.6222222 0.6136364
##   1 0.7222222 0.6250000 0.3809524 0.6744186 0.3777778 0.3863636
```

```
print("Legit ranking vs. condition")
```

```
## [1] "Legit ranking vs. condition"
```

```
prop.table(table(df_merged$rank_legit,df_merged$legit_condition),2)
```

```
##
##                                lr1        lr2        lr3        ls1        ls2
##   Definitely legitimate 0.22222222 0.15000000 0.11904762 0.39534884 0.06666667
##   Probably legitimate   0.50000000 0.47500000 0.26190476 0.27906977 0.31111111
##   I'm not sure          0.00000000 0.10000000 0.09523810 0.09302326 0.28888889
##   Probably a scam       0.25000000 0.20000000 0.33333333 0.20930233 0.24444444
##   Definitely a scam     0.02777778 0.07500000 0.19047619 0.02325581 0.08888889
##
##                                ls3
##   Definitely legitimate 0.06818182
##   Probably legitimate   0.31818182
##   I'm not sure          0.11363636
##   Probably a scam       0.27272727
##   Definitely a scam     0.22727273
```

```
print("Correct legit identification vs. frequency of doing online tasks")
```

```
## [1] "Correct legit identification vs. frequency of doing online tasks"
```

```r
prop.table(table(df_merged$rank_legit_bool, df_merged$risk_onlineTasks), 2)
```

```
##
##         Never Once or twice Three to five times More than five times
##   0 0.8500000     0.5102041           0.5625000            0.4181818
##   1 0.1500000     0.4897959           0.4375000            0.5818182
```

```r
print("Legit ranking vs. frequency of doing online tasks")
```

```
## [1] "Legit ranking vs. frequency of doing online tasks"
```

```r
prop.table(table(df_merged$rank_legit, df_merged$risk_onlineTasks), 2)
```

```
##
##                              Never Once or twice Three to five times
##   Definitely legitimate 0.00000000    0.04081633          0.06250000
##   Probably legitimate   0.15000000    0.44897959          0.37500000
##   I'm not sure          0.10000000    0.12244898          0.06250000
##   Probably a scam       0.45000000    0.26530612          0.25000000
##   Definitely a scam     0.30000000    0.12244898          0.25000000
##
##                         More than five times
##   Definitely legitimate           0.23636364
##   Probably legitimate             0.34545455
##   I'm not sure                    0.12727273
##   Probably a scam                 0.22424242
##   Definitely a scam               0.06666667
```

Legit action comparisons

Legit Youtube video comparisons The following code splits the list of actions that the user recommended in response to the legit youtube videos into columns of booleans, with each boolean indicating whether or not an action was selected.For example, `Exit.the.video.without.doing.anything` is `TRUE` if the partiicpant recommended that their friend exit the video.

```r
splitup <- sapply(unlist(df_merged$legit), strsplit, ',')
headnames <- unique(unlist(splitup))
mat <- t(unname(sapply(splitup, function(x) headnames %in% x)))
colnames(mat) <- headnames
df_legit_actions <- data.frame(df_merged, mat)
```

Each of the following subsections runs the statistical testing for a particular action.

Exit The following code tests which potential independent variables are significantly associated with recommending exiting the legit YouTube video.

```r
legit_exit_results <- stat_test(dep_var = "Exit.the.video.without.doing.anything", df=df_legit_actions,
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Paper Result: The following code prints results for variables that are significantly associated with recommending to Exit the legit YouTube video. The fact that condition is significantly associated with the rate of exiting the legit YouTube video is presented in Figure 3a of the paper.

```
legit_exit_results %>% filter(significant == TRUE)
```

```
##       independent                            dependent   test p.adjusted
## 1 legit_condition Exit.the.video.without.doing.anything fisher 0.00209979
##       p.raw effect_size significant
## 1 9.999e-05   0.3296086        TRUE
```

The following code prints the rate of recommending to exit the legit YouTube video by condition.

```
prop.table(table(df_legit_actions$Exit.the.video.without.doing.anything, df_legit_actions$legit_conditi
```

```
##
##                lr1        lr2        lr3        ls1        ls2        ls3
##   FALSE 0.84090909 0.87500000 0.79166667 0.95833333 0.79245283 0.53061224
##   TRUE  0.15909091 0.12500000 0.20833333 0.04166667 0.20754717 0.46938776
```

Search to learn more The following code tests which potential independent variables are significantly associated with recommending searching to learn more about the legit YouTube video.

```
legit_search_results <- stat_test(dep_var = "Search.online.to.learn.more.about.what.the.video.describes
```

Paper Result: The following code prints results for variables that are significantly associated with recommending to search for more information about the legit YouTube video. The fact that condition is significantly associated the rate of searching for more information about the legit YouTube video is presented in Figure 3a of the paper.

```
legit_search_results %>% filter(significant == TRUE)
```

```
##       independent                                                dependent
## 1 legit_condition Search.online.to.learn.more.about.what.the.video.describes
##     test p.adjusted     p.raw effect_size significant
## 1 fisher  0.0209979 0.0009999   0.2596229        TRUE
```

The following code prints the rate of recommending to search to learn more about the legit YouTube video by condition.

```
prop.table(table(df_legit_actions$Search.online.to.learn.more.about.what.the.video.describes, df_legit_a
```

```
##
##               lr1       lr2       lr3       ls1       ls2       ls3
##   FALSE 0.4318182 0.4166667 0.6250000 0.6041667 0.6226415 0.7959184
##   TRUE  0.5681818 0.5833333 0.3750000 0.3958333 0.3773585 0.2040816
```

Report the video The following code tests which potential independent variables are significantly associated with recommending to report the legit YouTube video.

```
legit_report_results <- stat_test(dep_var = "Report.the.video.to.YouTube", df=df_legit_actions, conditi
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Paper Result: The following code prints results for variables that are significantly associated with recommending to report the legit YouTube video. This result is discussed in section 5D of the paper.

```
legit_report_results %>% filter(significant == TRUE)
```

```
##       independent                   dependent                test p.adjusted
## 1 risk_onlineTasks Report.the.video.to.YouTube CochranArmitageTest 0.02156373
##         p.raw effect_size significant
## 1 0.001026844       0.448        TRUE
```

The following code prints the rate of recommending to report the legit YouTube video by frequency of doing online tasks.

```
prop.table(table(df_legit_actions$Report.the.video.to.YouTube, df_legit_actions$risk_onlineTasks),2)
```

```
##
##             Never Once or twice Three to five times More than five times
##   FALSE 0.90909091    0.86885246          1.00000000           0.98295455
##   TRUE  0.09090909    0.13114754          0.00000000           0.01704545
```

Post-hoc result

Paper Result Being an adult/teen is associated with frequency of doing online tasks. While being an adult/teen was not found to be a significant predictor of reporting legit YouTube videos, it was a significant predictor for reporting scam YouTube videos. For this reason, the following code tests whether frequency of doing online tasks is a significant predictor of ranking success when controlling for age. From this, we see that frequency of online tasks is not a significant predictor when looking at adults ($p_{uncorrected} = 0.1049$) or teens alone ($p_{uncorrected} = 0.4185$) This result is discussed in section 5D of the paper.

```
df_legit_actions_adult<-df_legit_actions %>% filter(adult == TRUE)
df_legit_actions_teen<-df_legit_actions %>% filter(adult == FALSE)
print("Test reporting legit YouTube videos vs. frequency of doing online tasks with just adults")
```

```
## [1] "Test reporting legit YouTube videos vs. frequency of doing online tasks with just adults"
```

```
CochranArmitageTest(table(df_legit_actions_adult$risk_onlineTasks, df_legit_actions_adult$Report.the.vi
```

```
##
##  Cochran-Armitage test for trend
##
## data:  table(df_legit_actions_adult$risk_onlineTasks, df_legit_actions_adult$Report.the.video.to.You
## Z = 1.6217, dim = 4, p-value = 0.1049
## alternative hypothesis: two.sided
```

```
print("Test reporting legit YouTube videos vs. frequency of doing online tasks with just teens")
```

```
## [1] "Test reporting legit YouTube videos vs. frequency of doing online tasks with just teens"
```

```r
CochranArmitageTest(table(df_legit_actions_teen$risk_onlineTasks, df_legit_actions_teen$Report.the.vide
```

```
##
##  Cochran-Armitage test for trend
##
## data:  table(df_legit_actions_teen$risk_onlineTasks, df_legit_actions_teen$Report.the.video.to.YouTu
## Z = 0.80895, dim = 4, p-value = 0.4185
## alternative hypothesis: two.sided
```

Visit the website The following code tests which potential independent variables are significantly associated with recommending to visit the website from the legit YouTube video.

```r
legit_visit_results <- stat_test(dep_var = "Visit.the.website.s..shown.in.the.video", df=df_legit_actio
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Paper Result: The following code prints results for variables that are significantly associated with recommending to visit the website shown in legit YouTube video. The fact that condition is significantly associated the rate of vising the website shown in the legit YouTube video is presented in Figure 3a of the paper. The other results are not discussed due to a lack of space.

```r
legit_visit_results %>% filter(significant == TRUE)
```

```
##        independent                               dependent                test
## 1            adult Visit.the.website.s..shown.in.the.video              fisher
## 2  legit_condition Visit.the.website.s..shown.in.the.video              fisher
## 3      time_mobile Visit.the.website.s..shown.in.the.video CochranArmitageTest
## 4 risk_onlineTasks Visit.the.website.s..shown.in.the.video CochranArmitageTest
##    p.adjusted        p.raw effect_size significant
## 1 0.002557643 0.0002435851   0.2060506        TRUE
## 2 0.002099790 0.0000999900   0.3512431        TRUE
## 3 0.007221967 0.0013328743   0.1660000        TRUE
## 4 0.007221967 0.0013756128   0.1850000        TRUE
```

The following code prints cross tables for significant results. The rows represent whether or not the user recommended visiting the website shown in the legit YouTube video.

```r
print("Visit legit website vs. teen (0)/adult(1)")
```

```
## [1] "Visit legit website vs. teen (0)/adult(1)"
```

```r
prop.table(table(df_legit_actions$Visit.the.website.s..shown.in.the.video, df_legit_actions$adult),2)
```

```
##
##                 0         1
##   FALSE 0.7529412 0.5219512
##   TRUE  0.2470588 0.4780488
```

```r
print("Visit legit website vs. condition")
```

```
## [1] "Visit legit website vs. condition"
```

```r
prop.table(table(df_legit_actions$Visit.the.website.s..shown.in.the.video, df_legit_actions$legit_condi
```

```
##
##               lr1       lr2       lr3       ls1       ls2       ls3
##   FALSE 0.5909091 0.5208333 0.7291667 0.2500000 0.6603774 0.7755102
##   TRUE  0.4090909 0.4791667 0.2708333 0.7500000 0.3396226 0.2244898
```

```r
print("Visit legit website vs. time spent playing mobile games")
```

```
## [1] "Visit legit website vs. time spent playing mobile games"
```

```r
prop.table(table(df_legit_actions$time_mobile, df_legit_actions$Visit.the.website.s..shown.in.the.video
```

```
##
##                          FALSE      TRUE
##   None               0.6406250 0.3593750
##   Less than an hour  0.6111111 0.3888889
##   1 to 5 hours       0.6750000 0.3250000
##   5 to 10 hours      0.5416667 0.4583333
##   10 to 15 hours     0.2857143 0.7142857
##   15 to 20 hours     0.2857143 0.7142857
##   More than 20 hours 0.0000000 1.0000000
```

```r
print("online tasks vs visit website")
```

```
## [1] "online tasks vs visit website"
```

```r
prop.table(table(df_legit_actions$risk_onlineTasks, df_legit_actions$Visit.the.website.s..shown.in.the.v
```

```
##
##                          FALSE      TRUE
##   Never               0.8181818 0.1818182
##   Once or twice       0.6557377 0.3442623
##   Three to five times 0.5500000 0.4500000
##   More than five times 0.5284091 0.4715909
```

Post-hoc tests

Being an adult/teen is associated with time spent playing mobile games per week. The following code determines whether time spent playing mobile games remains a significant predictor when controlling for age. From this, we see that time spent on mobile games remains a significant predictor of visiting the website shown in the legit YouTube video, even when looking at just adults ($p_{uncorrected} = 0.014$) or teens ($p_{uncorrected} < 0.001$).

```r
df_legit_actions_adult<-df_legit_actions %>% filter(adult == TRUE)
df_legit_actions_teen<-df_legit_actions %>% filter(adult == FALSE)
print("Test visit website vs. time on mobile games with just adults")
```

```
## [1] "Test visit website vs. time on mobile games with just adults"
```

```r
CochranArmitageTest(table(df_legit_actions_adult$time_mobile, df_legit_actions_adult$Visit.the.website.s
```

```
##
##  Cochran-Armitage test for trend
##
## data:  table(df_legit_actions_adult$time_mobile, df_legit_actions_adult$Visit.the.website.s..shown.in
## Z = -2.464, dim = 7, p-value = 0.01374
## alternative hypothesis: two.sided
```

```r
print("Test visit website vs. time on mobile games with just teens")
```

```
## [1] "Test visit website vs. time on mobile games with just teens"
```

```r
CochranArmitageTest(table(df_legit_actions_teen$time_mobile, df_legit_actions_teen$Visit.the.website.s.
```

```
##
##  Cochran-Armitage test for trend
```

```
##
## data:  table(df_legit_actions_teen$time_mobile, df_legit_actions_teen$Visit.the.website.s..shown.in.t
## Z = -3.9432, dim = 7, p-value = 8.041e-05
## alternative hypothesis: two.sided
```

Being an adult/teen is associated with frequency of doing online tasks. The following code determines whether frequency of doing online tasks remains a significant predictor when controlling for age. From this, we see that frequency of doing online tasks remains weakly significant predictor of visiting the website shown in the legit YouTube video only for teens alone ($p_{uncorrected} = 0.012$), but not adults alone ($p_{uncorrected} = 0.4441$)

```
print("online tasks vs visit website for adults")

## [1] "online tasks vs visit website for adults"

CochranArmitageTest(table(df_legit_actions_adult$risk_onlineTasks, df_legit_actions_adult$Visit.the.webs

##
##  Cochran-Armitage test for trend
##
## data:  table(df_legit_actions_adult$risk_onlineTasks, df_legit_actions_adult$Visit.the.website.s..sho
## Z = 0.7653, dim = 4, p-value = 0.4441
## alternative hypothesis: two.sided

print("online tasks vs visit website for teens")

## [1] "online tasks vs visit website for teens"

CochranArmitageTest(table(df_legit_actions_teen$risk_onlineTasks, df_legit_actions_teen$Visit.the.websi

##
##  Cochran-Armitage test for trend
##
## data:  table(df_legit_actions_teen$risk_onlineTasks, df_legit_actions_teen$Visit.the.website.s..shown
## Z = -2.5182, dim = 4, p-value = 0.01179
## alternative hypothesis: two.sided
```

Look at the comments The following code tests which potential independent variables are significantly associated with recommending to look at the comments on the legit YouTube video.

```
legit_lookatcomments_results <- stat_test(dep_var = "Look.at.the.comments.on.the.video", df=df_legit_ac
```

The following code shows that none of the tested variables were significantly associated with looking at the comments on the legit YouTube video

```
legit_lookatcomments_results %>% filter(significant == TRUE)

## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Leave a comment The following code tests which potential independent variables are significantly associated with recommending to leave a comment on the legit YouTube video

```
legit_leaveacomment_results <- stat_test(dep_var = "Leave.a.comment.on.the.video..please.specify.", df=d

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
```

```
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

The following code shows that none of the tested variables were significantly associated with recommending to leave a comment on the legit YouTube video

```
legit_leaveacomment_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

other The following code tests which potential independent variables are significantly associated with recommending a different action in response to the legit YouTube video

```
legit_other_results <- stat_test(dep_var = "Other..please.specify.", df=df_legit_actions, condition_typ
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

The following code shows that none of the tested variables were significantly associated with recommending a different action in response to the legit YouTube video

```r
legit_other_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Legit Web video cmpariosons The following code splits the list of actions that the user recommended in response to the legit website videos into columns of booleans, with each boolean indicating whether or not an action was selected.For example, `Exit.from.the.website.without.doing.anything` is `TRUE` if the partiicpant recommended that their friend exit the website.

```r
splitup <- sapply(unlist(df_merged$lgt_web), strsplit, ',')
headnames <- unique(unlist(splitup))
mat <- t(unname(sapply(splitup, function(x) headnames %in% x)))
colnames(mat) <- headnames
df_lgt_web_actions <- data.frame(df_merged, mat)
```

Exit The following code tests which potential independent variables are significantly associated with recommending to exit the legit website

```r
lgt_web_exit_results <- stat_test(dep_var = "Exit.from.the.website.without.doing.anything", df=df_lgt_we
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

The following code shows that none of the tested variables were significantly associated with exiting the legit website.

```r
lgt_web_exit_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Search online The following code tests which potential independent variables are significantly associated with recommending to search to learn more about the legit website

```r
lgt_web_search_results <- stat_test(dep_var = "Search.online.to.learn.more.about.the.website", df=df_lg
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

The following code prints results for variables that are significantly associated with recommending to search to learn more about the legit website.

```r
lgt_web_search_results %>% filter(significant == TRUE)
```

```
##        independent                                     dependent
## 1 risk_usedSpotify Search.online.to.learn.more.about.the.website
##                  test p.adjusted        p.raw effect_size significant
## 1 CochranArmitageTest 0.01302861 0.0006204101      0.0748        TRUE
```

The following code prints the table comparing the rate of recommending to search to learn more about the legit website based on level of usage of Spotify

```
prop.table(table(df_lgt_web_actions$Search.online.to.learn.more.about.the.website, df_lgt_web_actions$ri
```

```
##
##             Never Once or twice Three to five times More than five times
##   FALSE 1.0000000     0.6666667           0.3571429            0.7954545
##   TRUE  0.0000000     0.3333333           0.6428571            0.2045455
```

Follow all instructions The following code tests which potential independent variables are significantly associated with recommending to follow the instructions from the legit video.

```
lgt_web_follow_results <- stat_test(dep_var = "Follow.all.of.the.instructions.in.the.video.to.complete.t
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

The following code shows that none of the tested variables were significantly associated with following the instrucitons on the legit website.

```
lgt_web_follow_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Register for website The following code tests which potential independent variables are significantly associated with recommending to register for the legit video.

```
lgt_web_register_results <- stat_test(dep_var = "Register.for.the.website", df=df_lgt_web_actions, condi
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Paper result: The following code prints results for variables that are significantly associated with recommending to register for the legit website. The fact that condition is significantly associated with recommending to register for the website is shown in figure 3b.

```
lgt_web_register_results %>% filter(significant == TRUE)
```

```
##        independent                dependent                test p.adjusted
## 1  legit_condition Register.for.the.website              fisher 0.01049895
## 2 risk_onlineTasks Register.for.the.website CochranArmitageTest 0.02164421
##         p.raw effect_size significant
## 1 0.000499950   0.2771629        TRUE
## 2 0.002061353   0.1970000        TRUE
```

The following code prints cross tables for significant results. The rows represent whether or not the user recommended visiting the website shown in the legit YouTube video.

```
print("Register for the legit website vs legit condition ")
```

```
## [1] "Register for the legit website vs legit condition "
```

```
prop.table(table(df_lgt_web_actions$Register.for.the.website, df_lgt_web_actions$legit_condition),2)
```

```
##
##               lr1       lr2       lr3       ls1       ls2       ls3
##   FALSE 0.5454545 0.5625000 0.7291667 0.6666667 0.8113208 0.8979592
##   TRUE  0.4545455 0.4375000 0.2708333 0.3333333 0.1886792 0.1020408
```

```r
print("Rgister for legit website vs. rate of doing online tasks")
```

```
## [1] "Rgister for legit website vs. rate of doing online tasks"
```

```r
prop.table(table(df_lgt_web_actions$Register.for.the.website, df_lgt_web_actions$risk_onlineTasks),2)
```

```
##
##             Never Once or twice Three to five times More than five times
##   FALSE 0.8787879     0.7868852           0.7000000            0.6477273
##   TRUE  0.1212121     0.2131148           0.3000000            0.3522727
```

Ask for help The following code combines the option presented to adults ("Ask a knowledgeable friend for help") with the option presented ot teens ("Ask a parent for help")

```r
df_lgt_web_actions$Ask.for.help <- df_lgt_web_actions$Ask.a.knowledgeable.friend.for.help | df_lgt_web_a
```

The following code tests which potential independent variables are significantly associated with recommending to ask for help when viewing the legit website

```r
lgt_web_ask_for_help_results <- stat_test(dep_var = "Ask.for.help", df=df_lgt_web_actions, condition_typ
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Paper result: The following code prints results for variables that are significantly associated with asking for help when viewing the legit website. The fact that teens were more likely to recommend asking for hlep is discussed in seciton 5C

```r
lgt_web_ask_for_help_results %>% filter(significant == TRUE)
```

```
##   independent    dependent   test p.adjusted       p.raw effect_size
## 1       adult Ask.for.help fisher 0.02494114 0.001187673   0.1916494
##   significant
## 1        TRUE
```

The following code prints the rate of recommending asking for help between teens (0) and adults (1). The rows represent recommending asking for help or not.

```r
prop.table(table(df_lgt_web_actions$Ask.for.help, df_lgt_web_actions$adult),2)
```

```
##
##                  0          1
##   FALSE 0.77647059 0.92195122
##   TRUE  0.22352941 0.07804878
```

Other The following code tests which potential independent variables are significantly associated with recommending another action when viewing the legit website

```r
lgt_other_results <- stat_test(dep_var = "Other..please.specify.", df=df_lgt_web_actions)
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

The following code shows that none of the tested variables were significantly associated with recommending another action when viewing the legit website.

```
lgt_other_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Scam action comparsons

Scam YouTube video comparisons The following code splits the list of actions that the user recommended in response to the legit youtube videos into columns of booleans, with each boolean indicating whether or not an action was selected.For example, `Exit.the.video.without.doing.anything` is `TRUE` if the partiicpant recommended that their friend exit the video.

```
splitup <- sapply(unlist(df_merged$scam), strsplit, ',')
headnames <- unique(unlist(splitup))
mat <- t(unname(sapply(splitup, function(x) headnames %in% x)))
colnames(mat) <- headnames
df_scam_actions <- data.frame(df_merged, mat)
```

Paper result: The following code calculates the proportion of participants who both recommended visiting the scam website alongside another information gathering action. These proportions are discussed in section 5B of the paper.

```
print("Proportion of participants who recommended to visit the website AND look at the comments")
```

```
## [1] "Proportion of participants who recommended to visit the website AND look at the comments"
```

```
nrow(filter(df_scam_actions, Visit.the.website.s..shown.in.the.video &  (Look.at.the.comments.on.the.vi
```

```
## [1] 0.6
```

```
print("Proportion of participants who recommended to visit the website AND search online to learn more"
```

```
## [1] "Proportion of participants who recommended to visit the website AND search online to learn more
```

```
nrow(filter(df_scam_actions, Visit.the.website.s..shown.in.the.video &  (Search.online.to.learn.more.ab
```

```
## [1] 0.3714286
```

Exit run stat tests

```
scam_exit_results <- stat_test(dep_var = "Exit.the.video.without.doing.anything", df=df_scam_actions)
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Filter to significant results

```r
scam_exit_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Search to learn more run stat tests

```r
scam_search_results <- stat_test(dep_var = "Search.online.to.learn.more.about.what.the.video.describes"
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Filter to significant results

```r
scam_search_results %>% filter(significant == TRUE)
```

```
##     independent                                                  dependent
## 1 scam_condition Search.online.to.learn.more.about.what.the.video.describes
##     test p.adjusted      p.raw effect_size significant
## 1 fisher 0.01469853 0.00069993    0.300931        TRUE
```

Look at signfiicant cross tabs

```r
prop.table(table(df_scam_actions$scam_condition , df_scam_actions$Search.online.to.learn.more.about.wha
```

```
##
##           FALSE      TRUE
##   sr1 0.5106383 0.4893617
##   sr2 0.7555556 0.2444444
##   sr3 0.8541667 0.1458333
##   ss1 0.8846154 0.1153846
##   ss2 0.7500000 0.2500000
##   ss3 0.8600000 0.1400000
```

Report the video run stat tests

```r
scam_report_results <- stat_test(dep_var = "Report.the.video.to.YouTube", df=df_scam_actions)
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Filter to significant results

```r
scam_report_results %>% filter(significant == TRUE)
```

```
##       independent                   dependent                 test   p.adjusted
## 1           adult Report.the.video.to.YouTube               fisher 0.0005506151
## 2 risk_onlineTasks Report.the.video.to.YouTube CochranArmitageTest 0.0153465311
##          p.raw effect_size significant
```

```
## 1 2.621977e-05   0.2499186        TRUE
## 2 1.461574e-03   0.3380000        TRUE
```

Look at significant crosstabs

```r
print(" adult vs. report videos")
```

```
## [1] " adult vs. report videos"
```

```r
prop.table(table(df_scam_actions$adult, df_scam_actions$Report.the.video.to.YouTube),1)
```

```
##
##          FALSE       TRUE
##   0 0.78823529 0.21176471
##   1 0.95609756 0.04390244
```

```r
print("online tasks vs. report videos")
```

```
## [1] "online tasks vs. report videos"
```

```r
prop.table(table(df_scam_actions$risk_onlineTasks, df_scam_actions$Report.the.video.to.YouTube),1)
```

```
##
##                            FALSE       TRUE
##   Never                0.87878788 0.12121212
##   Once or twice        0.78688525 0.21311475
##   Three to five times  0.90000000 0.10000000
##   More than five times 0.95454545 0.04545455
```

```r
# Check if result continues if we look at just adults or teens alone
df_scam_actions_adult <- df_scam_actions %>% filter(adult == TRUE)
df_scam_actions_teen <- df_scam_actions %>% filter(adult == FALSE)
CochranArmitageTest(table(df_scam_actions_adult$Report.the.video.to.YouTube, df_scam_actions_adult$risk_
```

```
##
##  Cochran-Armitage test for trend
##
## data:  table(df_scam_actions_adult$Report.the.video.to.YouTube, df_scam_actions_adult$risk_onlineTas
## Z = 0.15648, dim = 4, p-value = 0.8757
## alternative hypothesis: two.sided
```

```r
CochranArmitageTest(table(df_scam_actions_teen$Report.the.video.to.YouTube, df_scam_actions_teen$risk_o
```

```
##
##  Cochran-Armitage test for trend
##
## data:  table(df_scam_actions_teen$Report.the.video.to.YouTube, df_scam_actions_teen$risk_onlineTasks
## Z = -0.35076, dim = 4, p-value = 0.7258
## alternative hypothesis: two.sided
```

Visit the website

run stat tests

```r
scam_visit_results <- stat_test(dep_var = "Visit.the.website.s..shown.in.the.video", df=df_scam_actions
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Filter to significant results

```
scam_visit_results %>% filter(significant == TRUE)
```

```
##      independent                                 dependent   test p.adjusted
## 1 scam_condition Visit.the.website.s..shown.in.the.video fisher 0.00419958
##        p.raw effect_size significant
## 1 0.00019998    0.289071        TRUE
```

Look at significant crosstabs

```
prop.table(table(df_scam_actions$scam_condition, df_scam_actions$Visit.the.website.s..shown.in.the.vide
```

```
##
##            FALSE       TRUE
##   sr1 0.70212766 0.29787234
##   sr2 0.88888889 0.11111111
##   sr3 0.97916667 0.02083333
##   ss1 0.98076923 0.01923077
##   ss2 0.87500000 0.12500000
##   ss3 0.84000000 0.16000000
```

Leave a comments run stat tests

```
scam_leavecomments_results <- stat_test(dep_var = "Leave.a.comment.on.the.video..please.specify.", df=d
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Filter to significant results

```
scam_leavecomments_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

Look at comments run stat tests

```
scam_lookcomments_results <- stat_test(dep_var = "Look.at.the.comments.on.the.video", df=df_scam_actions
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Filter to significant results

```
scam_lookcomments_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```

other run stat tests

```
scam_other_results <- stat_test(dep_var = "Other..please.specify.", df=df_scam_actions)
```

```
## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect

## Warning in stats::chisq.test(x, y, correct = correct, ...): Chi-squared
## approximation may be incorrect
```

Filter to significant results

```
scam_other_results %>% filter(significant == TRUE)
```

```
## [1] independent dependent   test        p.adjusted  p.raw       effect_size
## [7] significant
## <0 rows> (or 0-length row.names)
```