Experiment Default Effects Applications

Analysis July 19

Omar Vasquez Duque

1 Data cleaning

First experiment. July 13, 2023. Prolific. Sample of 300 people. Second experiment, July 14, 2023. Sample of 400 people.

1.1 var creation for analysis

1.2 var creation second exp

```
table(dfexp$choice)
```

```
##
##      Google        Bing DuckDuck Go      Ecosia       Yahoo
##         116           5          10           0           0
```

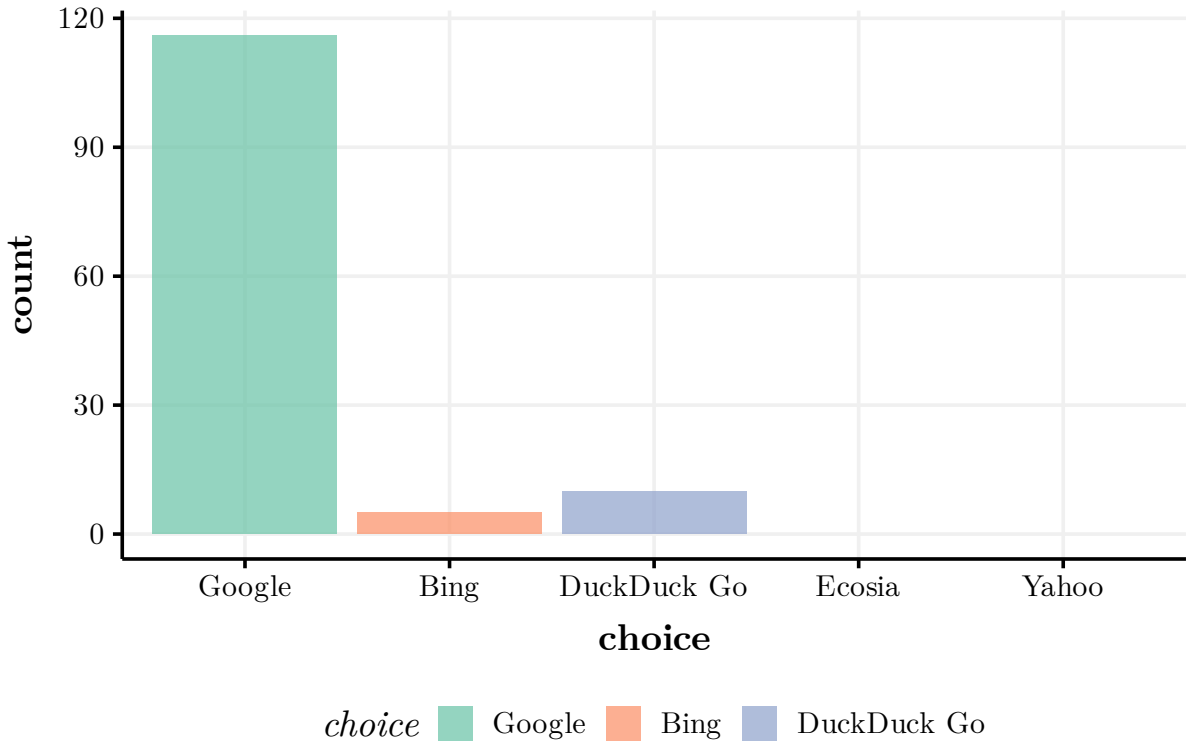
```
table(dfexp$assigned_bing)
```

```
##
##  0  1
## 95 31
```

2 Exploratory analysis

3 search engine choice plot

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```




```
# test = dfexp %>%
#   mutate(across(starts_with("used"), as.numeric))

#' *This worked!!! Saves a lot of time. Mind I should use "across" much more. And apply, lappl

# test = test %>%
#   mutate(nbing = used_bing_q1 + used_bing_q2 + used_bing_q3) # This works too!
```

3.1 creates dfs for comparison pA|D(A) vs pA|C

```
# Create dfs for comparison with forced choice

dfgoogle = dfexp %>%
  filter(forced_choice_s==1 | assigned_google==1)

dfbing = dfexp %>%
  filter(forced_choice_s==1 | assigned_bing==1)

dfyahoo = dfexp %>%
  filter(forced_choice_s==1 | assigned_yahoo==1)
```

```
dfddg = dfexp %>%
  filter(forced_choice_s==1 | assigned_duckduckgo==1)
```

```
library(jtools)
```

Mind there are two ways to operationalize the effect:

1) compared with choice-screen

2) compared with any other default

3.2 creates df for default condition

```
# DF for defaults in search
dfexp_def_s = dfexp %>%
  filter(choice_search==0)
```

4 Status Quo Effects Search

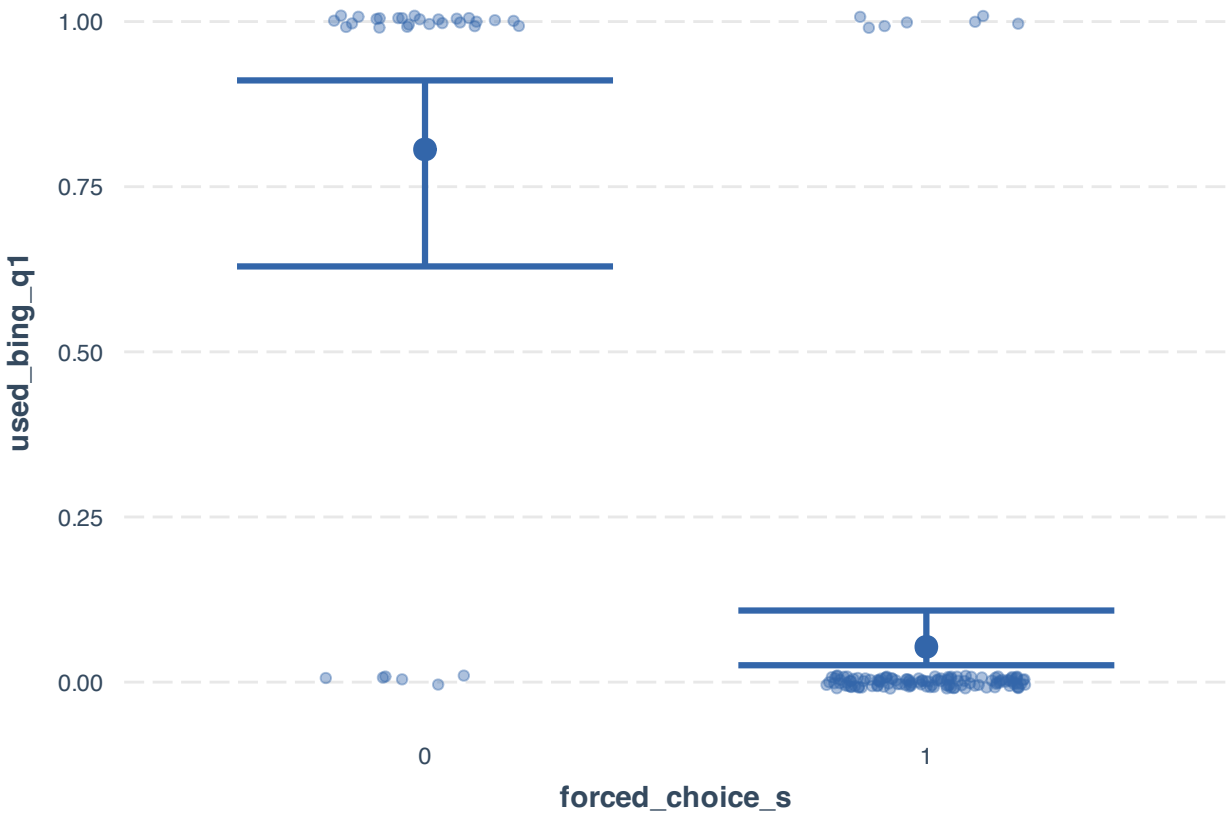
4.1 Bing

```
# Bing Q1
sqbingq1= glm(used_bing_q1 ~ forced_choice_s, data=dfbing, family ="binomial")

#check effect on quality

qualbing = lm(bing_qual ~ forced_choice_s, data = dfbing)

summary(qualbing)
```

```
##
## Call:
## lm(formula = bing_qual ~ forced_choice_s, data = dfbing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8929 -0.7689  0.0471  0.8012  2.2311
##
## Coefficients:
##                  Estimate Std. Error t value             Pr(>|t|)
## (Intercept)        3.5729     0.2024  17.657 < 0.0000000000000002 ***
## forced_choice_s1  -0.8040     0.2250  -3.573             0.000466 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.127 on 160 degrees of freedom
## Multiple R-squared:  0.0739, Adjusted R-squared:  0.06811
## F-statistic: 12.77 on 1 and 160 DF,  p-value: 0.0004664
```

```
plotbingq1 = effect_plot(sqbingq1, pred = "forced_choice_s", interval = TRUE, plot.points = TRU

plotbingq1
```



*Interesting. Status quo effect goes down. People start switching. I need longer questionnaire. Hopefully, 5 enough. If not, run final version with 7 questions? Not sure it's worth it

*Different comparison: Pb|B compared with any other default

```
lm(data = dfexp, used_bing_q1 ~ assigned_bing)
```

```
##
## Call:
## lm(formula = used_bing_q1 ~ assigned_bing, data = dfexp)
##
## Coefficients:
```

```
##    (Intercept)  assigned_bing1
##        0.01053         0.79593
```

```
lm(data = dfexp, used_bing_q1 ~ assigned_bing + bing_qual + bing_prior) #It is working this ti
```

```
##
## Call:
## lm(formula = used_bing_q1 ~ assigned_bing + bing_qual + bing_prior,
##     data = dfexp)
##
## Coefficients:
##    (Intercept)  assigned_bing1       bing_qual      bing_prior
##       -0.19284         0.73993         0.07055         0.01190
```

```
bingq1 = glm(data = dfexp_def_s, used_bing_q1 ~ assigned_bing: bing_prior, family = "binomial")

summary(bingq1)
```

```
##
## Call:
## glm(formula = used_bing_q1 ~ assigned_bing:bing_prior, family = "binomial",
##     data = dfexp_def_s)
##
## Coefficients:
##                            Estimate Std. Error z value    Pr(>|z|)
## (Intercept)                 -1.9636     0.3775  -5.201 0.000000198 ***
## assigned_bing0:bing_prior   -1.7500     1.0802  -1.620       0.105
## assigned_bing1:bing_prior    4.1037     0.8375   4.900 0.000000958 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 128.29  on 125  degrees of freedom
## Residual deviance:  70.73  on 123  degrees of freedom
## AIC: 76.73
##
## Number of Fisher Scoring iterations: 6
```

```
typeof(dfexp$bing_qual)
```

```
## [1] "double"
```

```r
lm(data = dfexp, used_bing_q3 ~ assigned_bing)
```

```
##
## Call:
## lm(formula = used_bing_q3 ~ assigned_bing, data = dfexp)
##
## Coefficients:
##    (Intercept)  assigned_bing1
##        0.01053         0.66689
```

```r
#'* Second experiment*
lm(data = datexp, used_bing_q1 ~ assigned_bing)
```

```
##
## Call:
## lm(formula = used_bing_q1 ~ assigned_bing, data = datexp)
##
## Coefficients:
##    (Intercept)  assigned_bing1
##        0.03086         0.72710
```

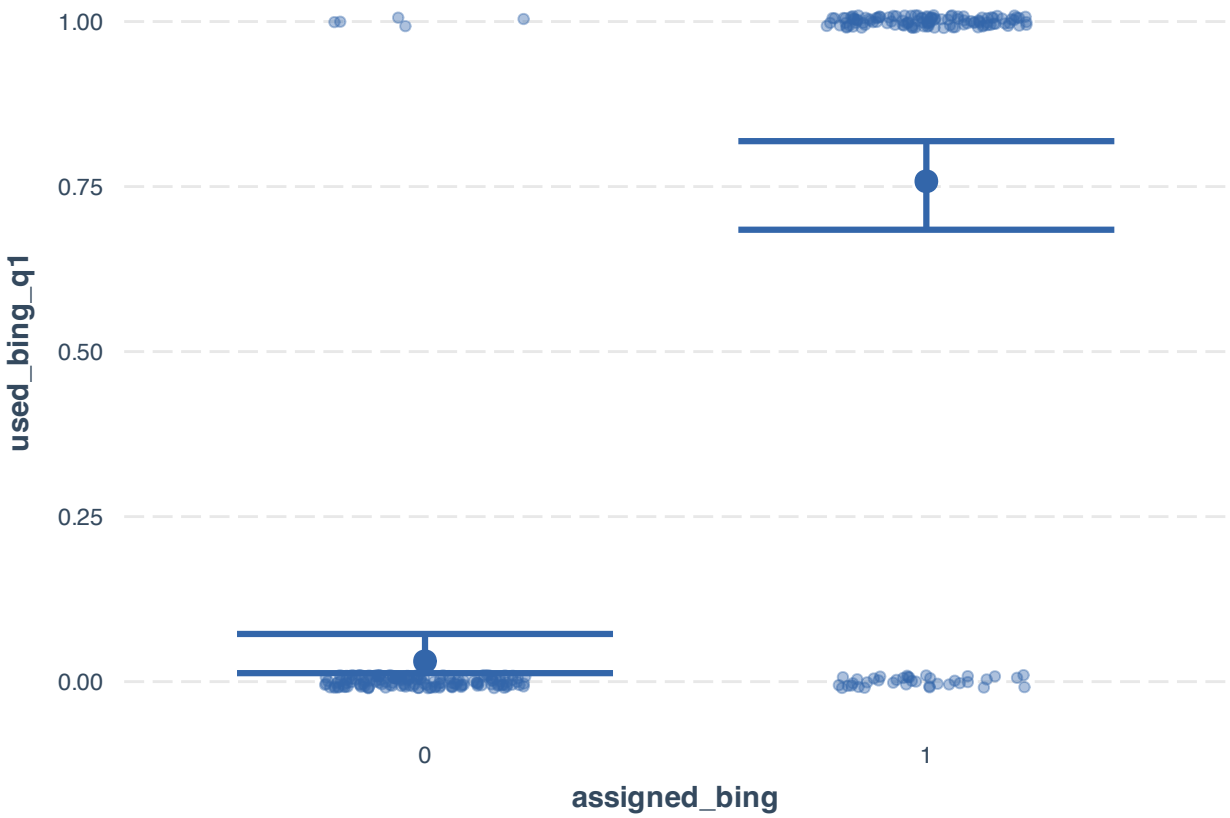
```r
e2.test = lm(data = datexp, used_bing_q1 ~ assigned_bing:bing_prior + bing_qual)
summary(e2.test) #there is an interaction effect. Prior use moderates the effect of the defaul
```

```
##
## Call:
## lm(formula = used_bing_q1 ~ assigned_bing:bing_prior + bing_qual,
##     data = datexp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.87099 -0.15981 -0.07405  0.16978  0.92608
##
## Coefficients:
##                             Estimate Std. Error t value            Pr(>|t|)
## (Intercept)                  0.05064    0.06301   0.804               0.4222
## bing_qual                    0.03988    0.01970   2.024               0.0439 *
## assigned_bing0:bing_prior -0.11669    0.05617  -2.078               0.0386 *
## assigned_bing1:bing_prior  0.62096    0.04936  12.581 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3547 on 310 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.4802, Adjusted R-squared:  0.4752
## F-statistic: 95.46 on 3 and 310 DF,  p-value: < 0.00000000000000022
```

```r
e2.bingq1 = glm(data = datexp, used_bing_q1 ~ assigned_bing + bing_prior, family = "binomial")

summary(e2.bingq1)
```

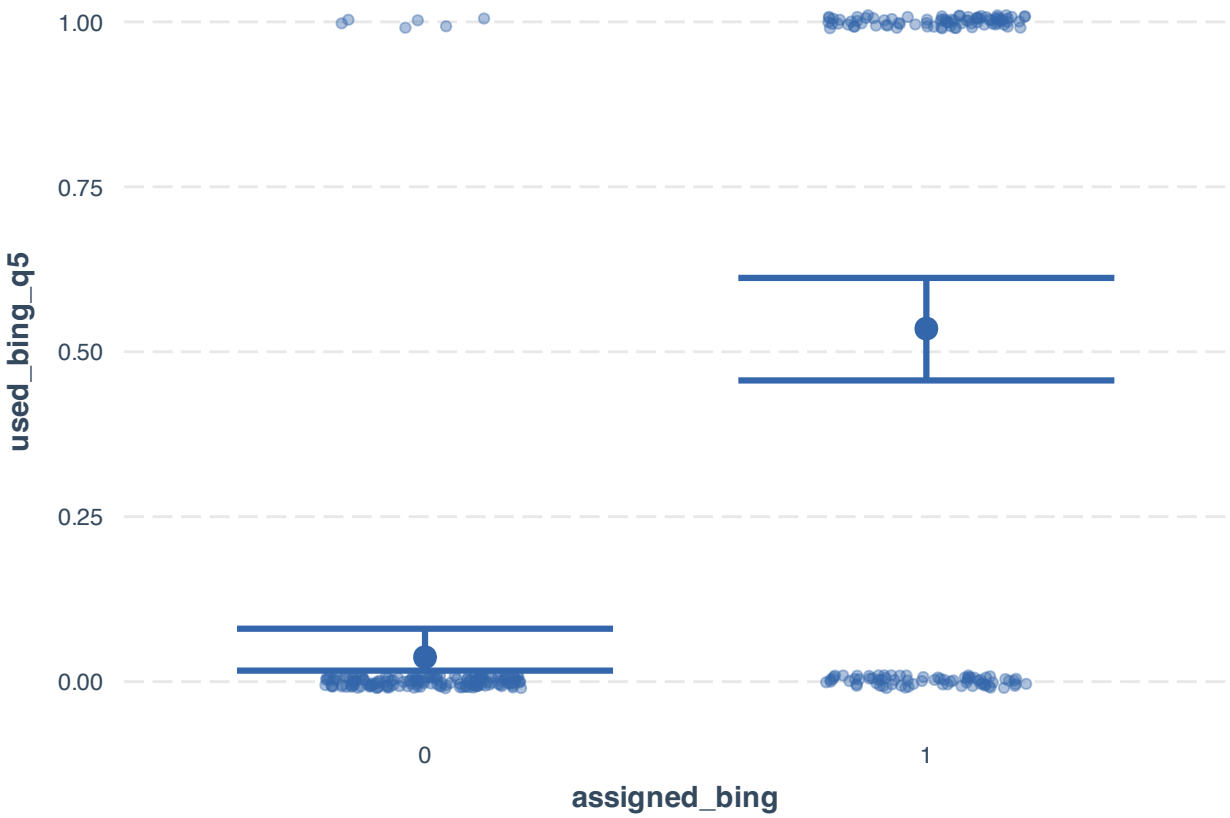
```
##
## Call:
## glm(formula = used_bing_q1 ~ assigned_bing + bing_prior, family = "binomial",
##     data = datexp)
##
## Coefficients:
##                Estimate Std. Error z value             Pr(>|z|)
## (Intercept)     -4.3456     0.5390  -8.062 0.000000000000000752 ***
## assigned_bing1   4.4281     0.5000   8.857 < 0.0000000000000002 ***
## bing_prior       1.5298     0.3801   4.024 0.000057136630600006 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 426.29  on 318  degrees of freedom
## Residual deviance: 201.80  on 316  degrees of freedom
## AIC: 207.8
##
## Number of Fisher Scoring iterations: 6
```

```r
#'*visualization*
# E2. Bing Q1
e2.sqbingq1= glm(used_bing_q1 ~ assigned_bing, data=datexp, family ="binomial")

e2.plotbingq1 = effect_plot(e2.sqbingq1, pred = "assigned_bing", interval = TRUE, plot.points =

e2.plotbingq1
```



Now, do the same for Q5

```
e2.sqbingq5= glm(used_bing_q5 ~ assigned_bing, data=datexp, family ="binomial")

e2.plotbingq5 = effect_plot(e2.sqbingq5, pred = "assigned_bing", interval = TRUE, plot.points =

e2.plotbingq5
```



See how this effect affects Google

4.2 Video had no effect. Probably everyone pais attention. Check number of correct answers

See if video had an effect... It did not work out. No difference

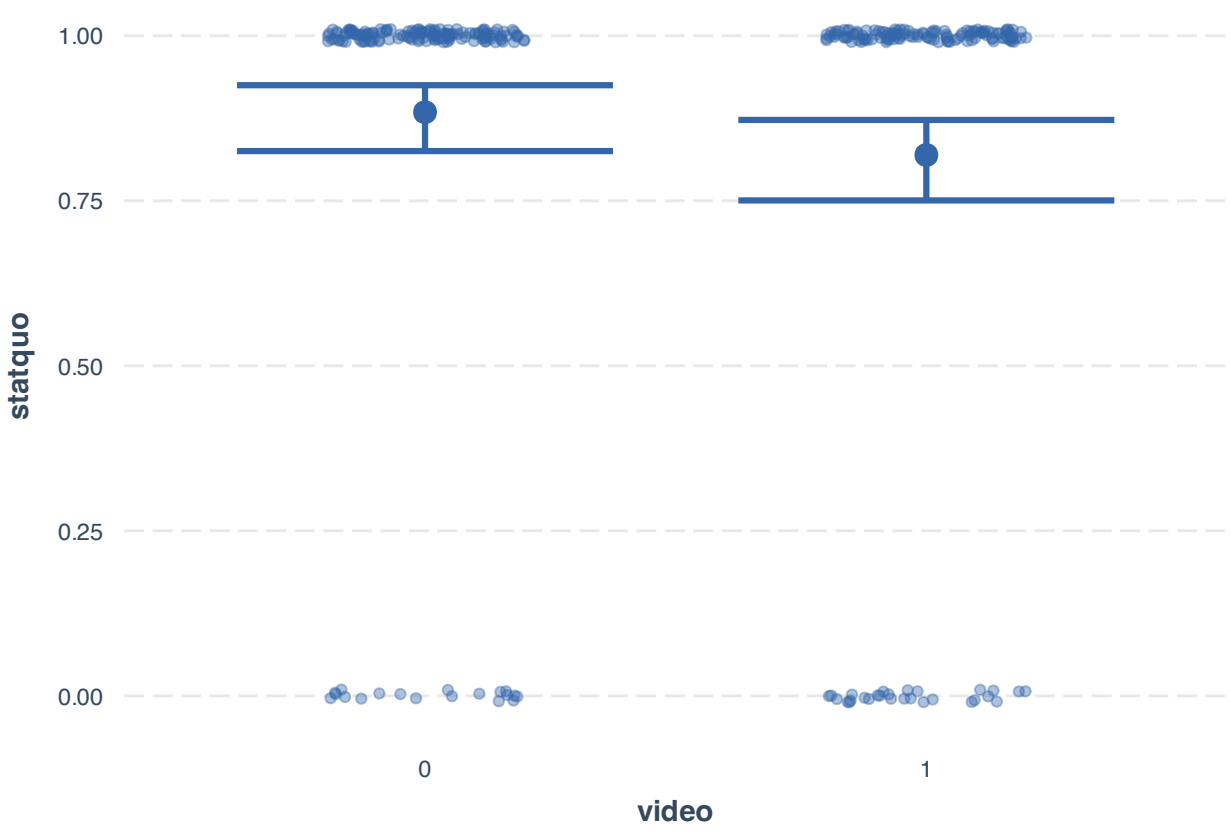
```
# Creates status quo variable (for Q1)
datexp = datexp %>%
  mutate(statquo = ifelse(assigned_bing ==1 & used_bing_q1==1 | assigned_google ==1 & used_goo
         video = as.factor(video))

mean(datexp$statquo)
```

```
## [1] 0.8526646
```

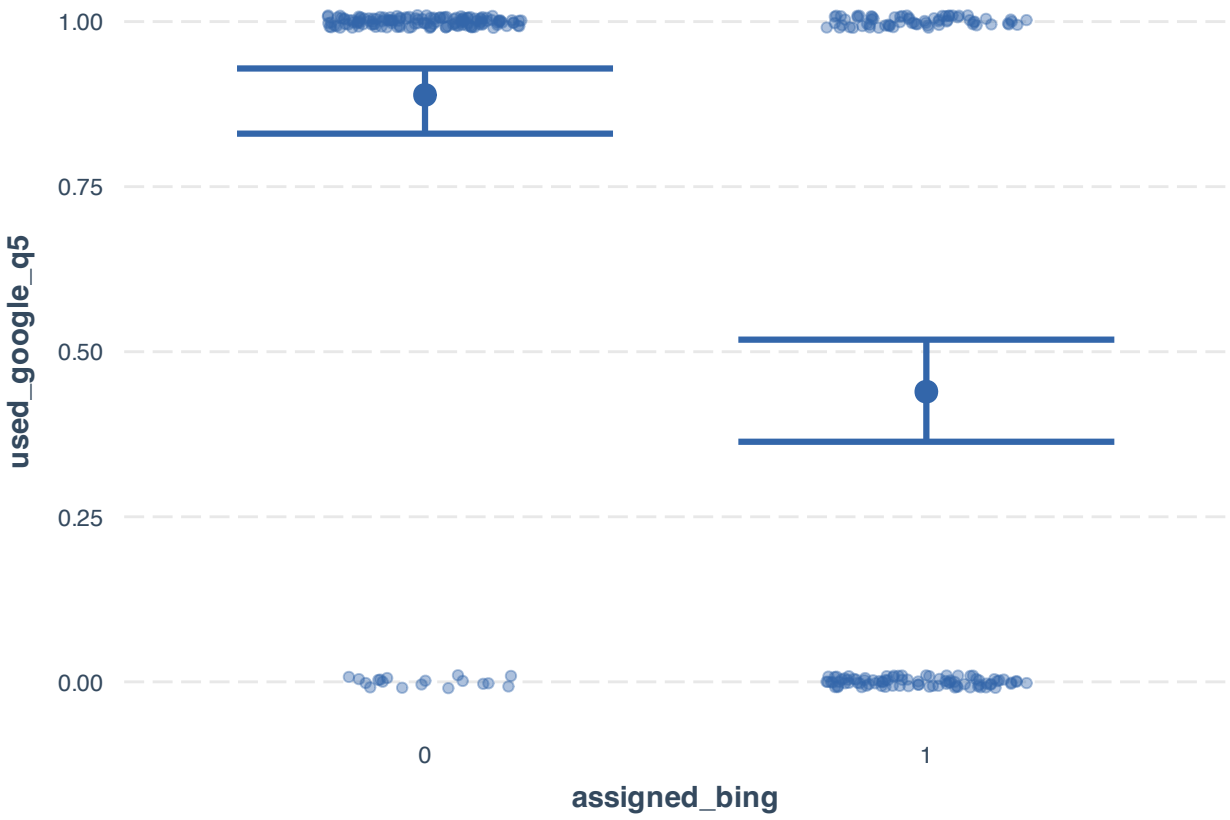
```
datbing = datexp %>%
  filter(assigned_bing==1) # New DF to compare just those assigned to Bing

glmvideo = glm(data=datexp, formula=statquo ~ video, family = "binomial")

summary(glmvideo)
```

```
##
## Call:
## glm(formula = statquo ~ video, family = "binomial", data = datexp)
##
## Coefficients:
##             Estimate Std. Error z value            Pr(>|z|)
## (Intercept)   2.0323     0.2440    8.33 <0.0000000000000002 ***
## video1       -0.5203     0.3211   -1.62               0.105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 266.72  on 318  degrees of freedom
## Residual deviance: 264.05  on 317  degrees of freedom
## AIC: 268.05
##
## Number of Fisher Scoring iterations: 4
```

```
glm.video = effect_plot(glmvideo, pred = "video", interval = TRUE, plot.points = TRUE, jitter =

glm.video
```

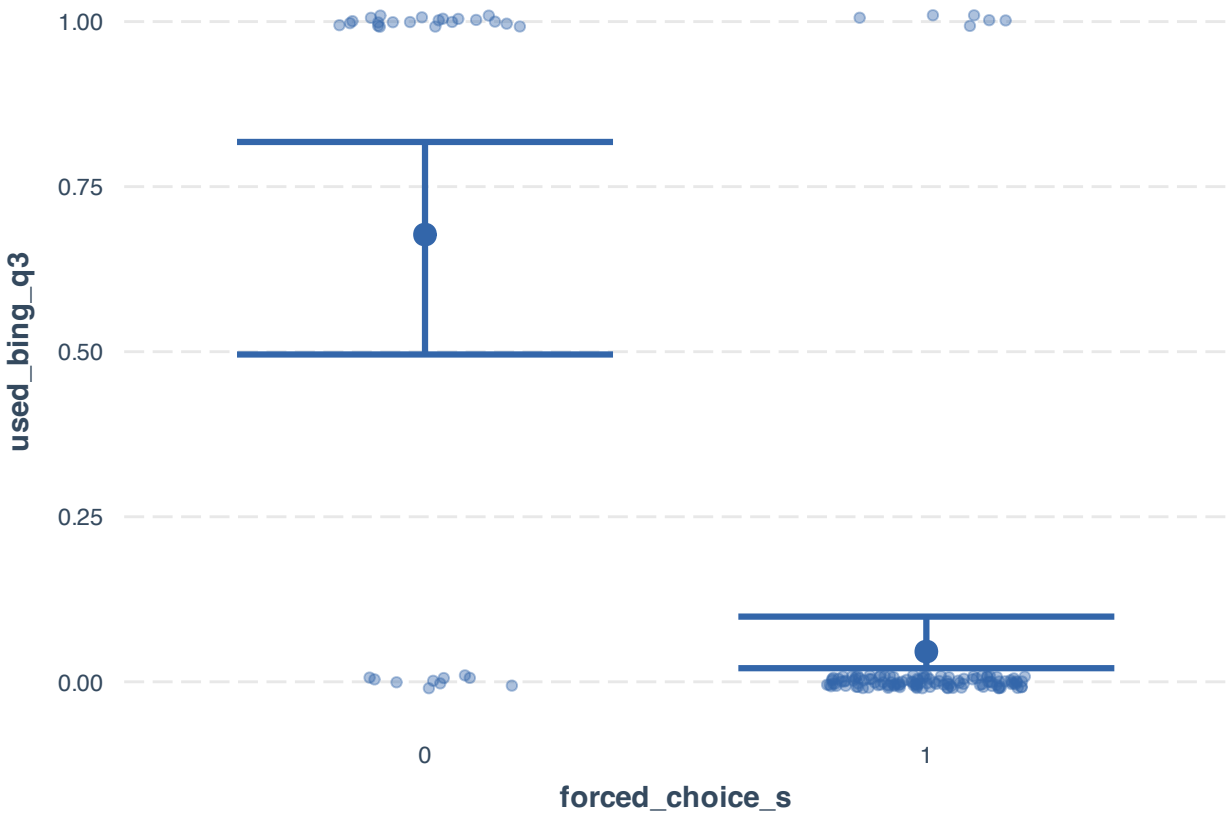


```
e2.googleq5= glm(used_google_q5 ~ assigned_bing, data=datexp, family ="binomial")

e2.plotgoogleq5 = effect_plot(e2.googleq5, pred = "assigned_bing", interval = TRUE, plot.points

e2.plotgoogleq5
```



Substantial effect! lowers use of Google!

```
#Bing Q3

sqbingq3= glm(used_bing_q3 ~ forced_choice_s, data=dfbing, family ="binomial")

plotbingq3 = effect_plot(sqbingq3, pred = "forced_choice_s", interval = TRUE, plot.points = TRU

plotbingq3
```

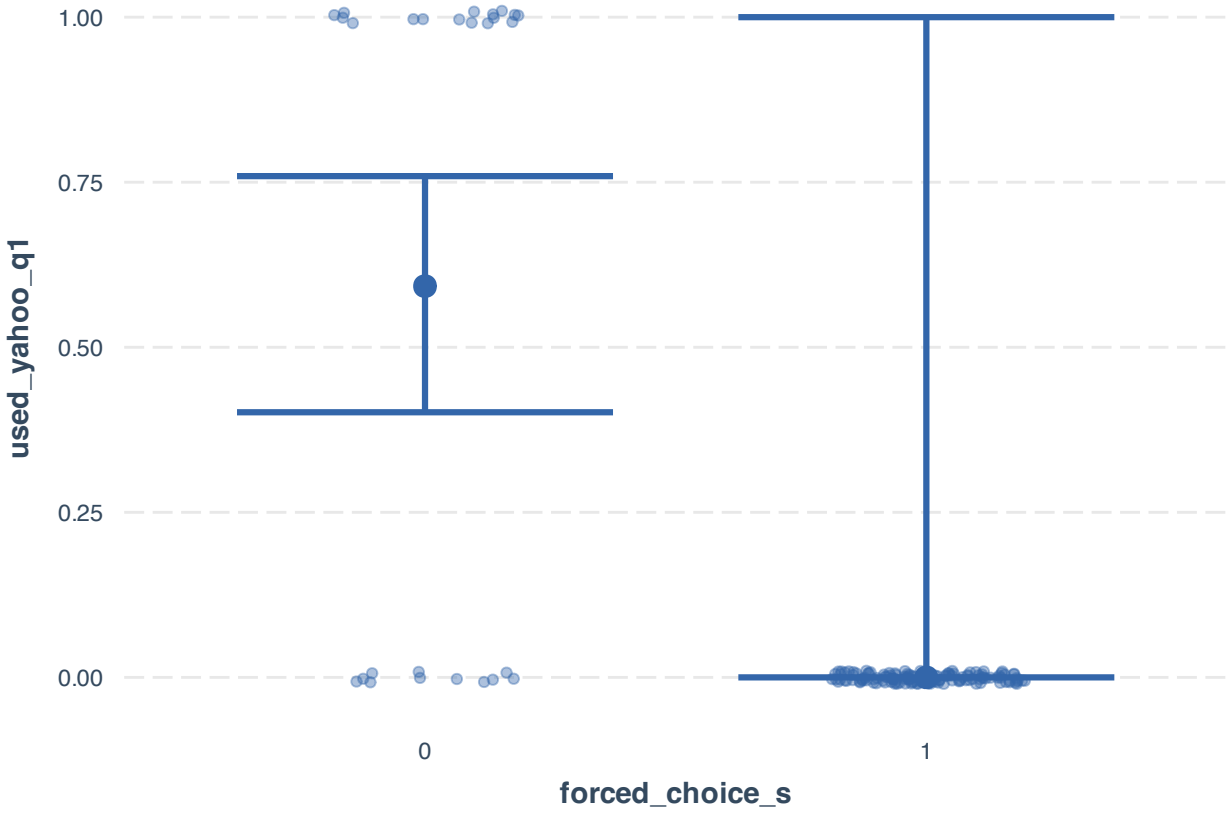


4.3 Yahoo

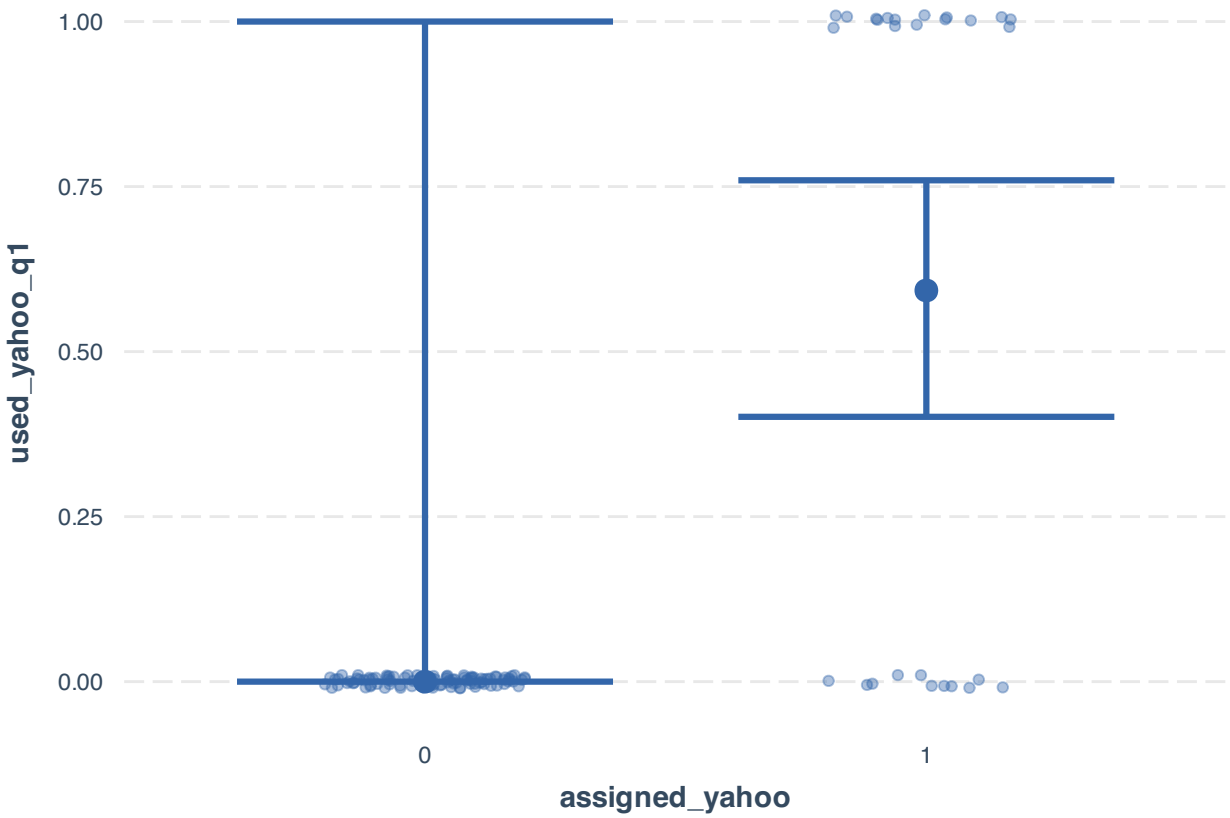
Check status quo effects for Yahoo. There is no variance, so just report counts

```
sqyahooq1= glm(used_yahoo_q1 ~ forced_choice_s, data=dfyahoo, family ="binomial")

plotyahooq1 = effect_plot(sqyahooq1, pred = "forced_choice_s", interval = TRUE, plot.points = T

#Now check for default data: pY|Y vs pY|Y`
sqdefyahooq1 = glm(used_yahoo_q1 ~ assigned_yahoo, data=dfexp_def_s, family ="binomial")
plotsqdefyahooq1 = effect_plot(sqdefyahooq1, pred = "assigned_yahoo", interval = TRUE, plot.poi
# Check q3
sqdefyahooq3 = glm(used_yahoo_q3 ~ assigned_yahoo, data=dfexp_def_s, family ="binomial")
plotsqdefyahooq3 = effect_plot(sqdefyahooq3, pred = "assigned_yahoo", interval = TRUE, plot.poi
```

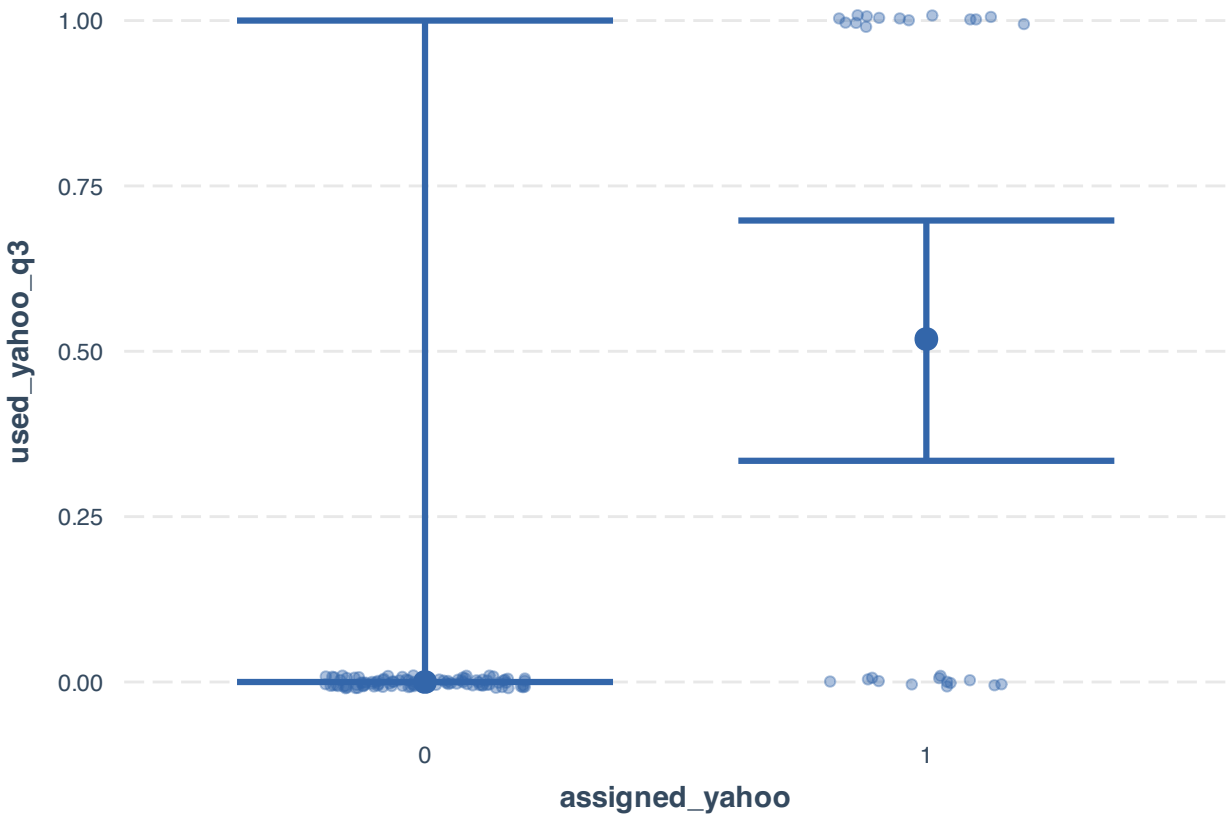
```
plotyahooq1
```




```
plotsqdefyahooq1
```



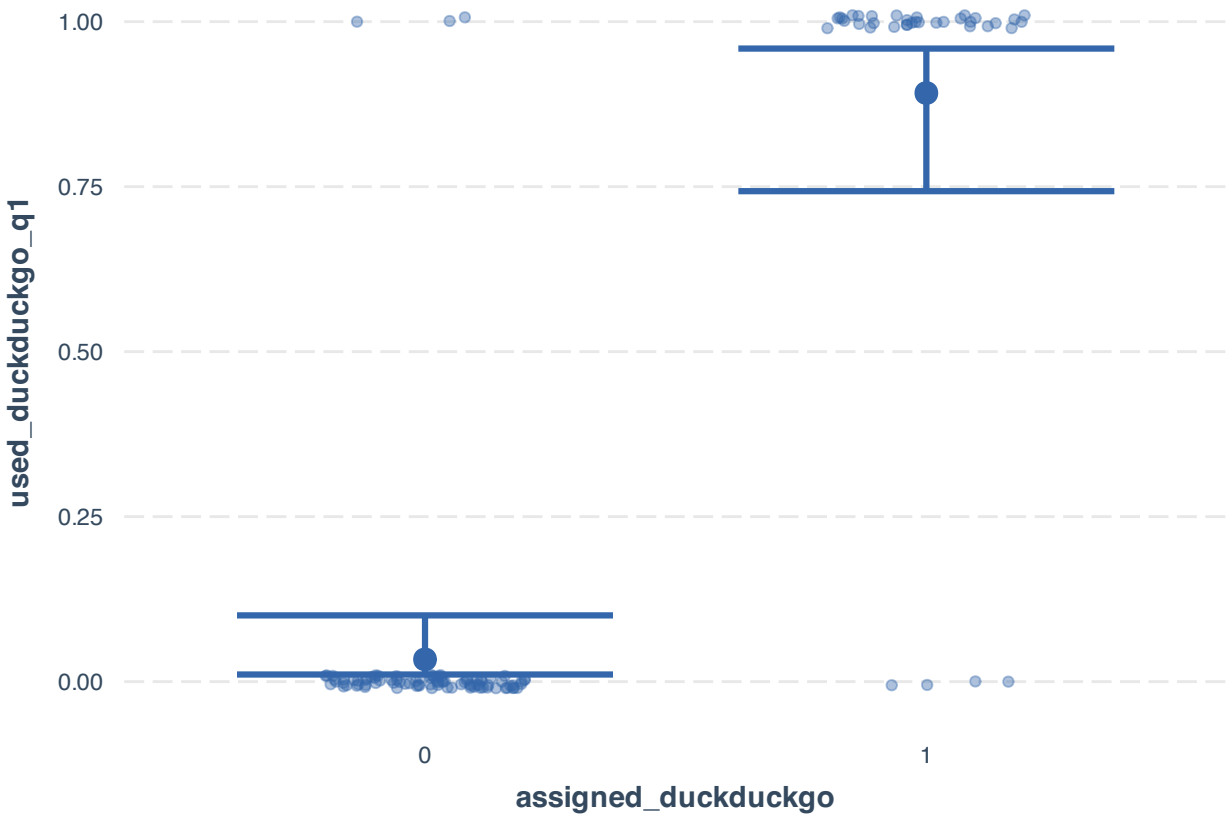

```
plotsqdefyahooq3
```



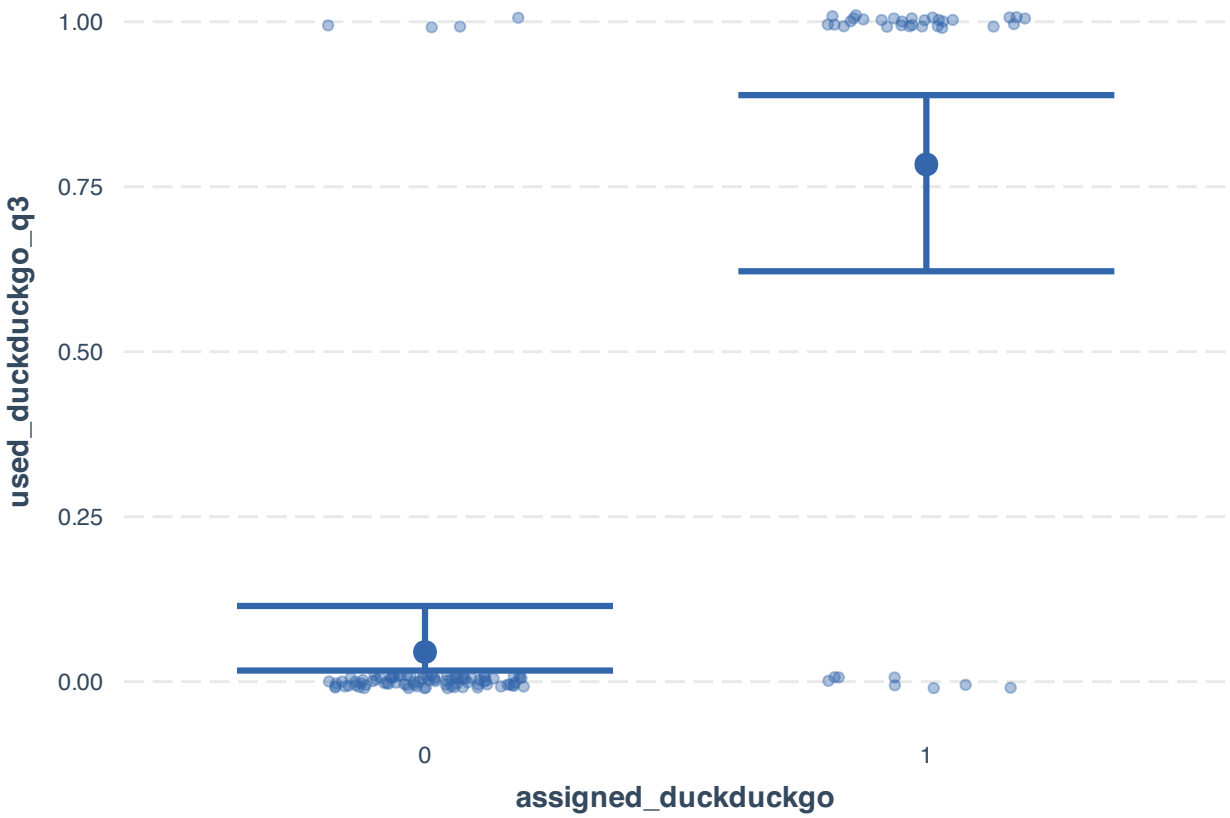
4.4 DuckDuck Go

Do the same with DuckDuckGo

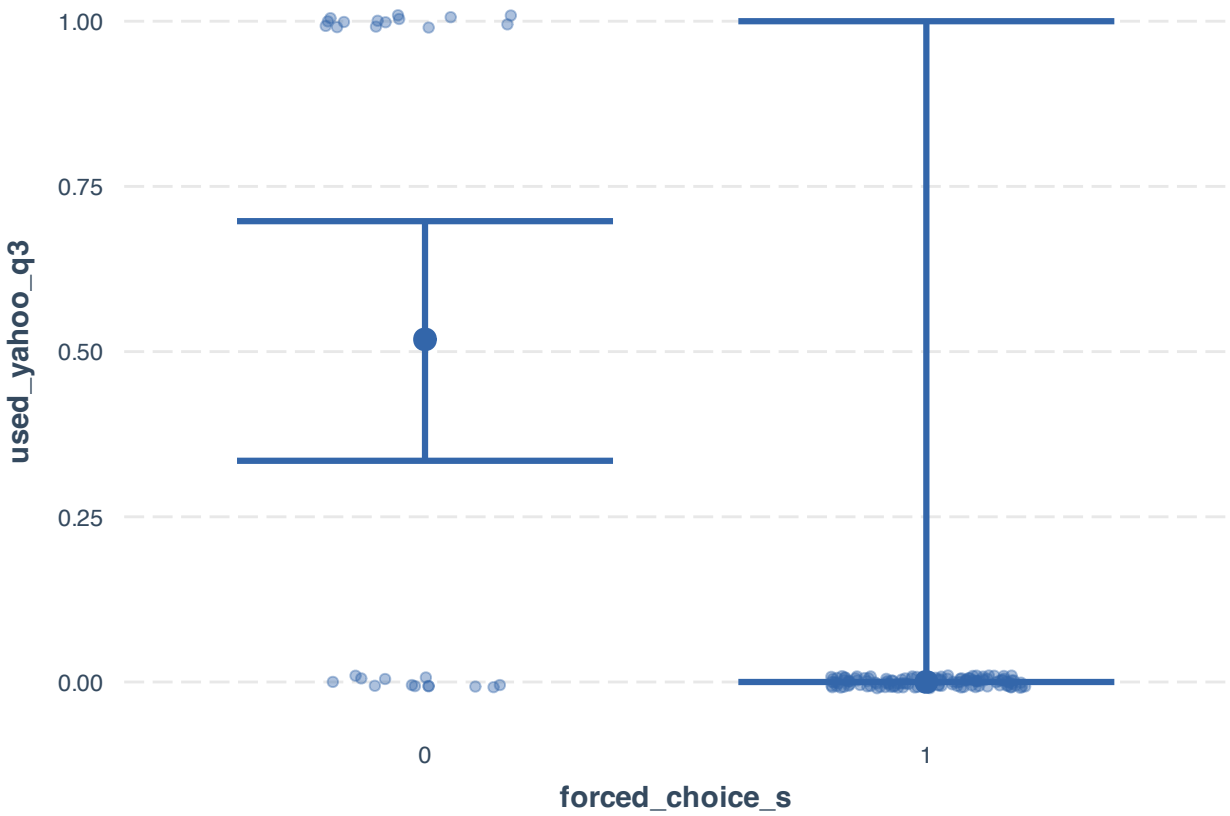
```
sqdefddg1 = glm(used_duckduckgo_q1 ~ assigned_duckduckgo, data=dfexp_def_s, family ="binomial"

plotsqdefddg1 = effect_plot(sqdefddg1, pred = "assigned_duckduckgo", interval = TRUE, plot.poi

plotsqdefddg1
```




```
# Check q3

sqdefddgq3 = glm(used_duckduckgo_q3 ~ assigned_duckduckgo, data=dfexp_def_s, family ="binomial"

plotsqdefddgq3 = effect_plot(sqdefddgq3, pred = "assigned_duckduckgo", interval = TRUE, plot.p

plotsqdefddgq3
```




```
sqyahooq3= glm(used_yahoo_q3 ~ forced_choice_s, data=dfyahoo, family ="binomial")

plotyahooq3 = effect_plot(sqyahooq3, pred = "forced_choice_s", interval = TRUE, plot.points = T

plotyahooq3
```



Compare with other defaults. Do chi-squared test.

```
yahooq1 = glm(data = dfexp_def_s, used_yahoo_q1 ~ assigned_yahoo + yahoo_prior, family = "bino

summary(yahooq1)
```

```
##
## Call:
## glm(formula = used_yahoo_q1 ~ assigned_yahoo + yahoo_prior, family = "binomial",
##     data = dfexp_def_s)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -21.897   2886.766  -0.008    0.994
## assigned_yahoo1   21.763   2886.766   0.008    0.994
## yahoo_prior        1.232      0.844   1.460    0.144
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 95.915  on 125  degrees of freedom
## Residual deviance: 34.224  on 123  degrees of freedom
## AIC: 40.224
##
```

```
## Number of Fisher Scoring iterations: 20
```

```
lm(data = dfexp_def_s, used_yahoo_q1 ~ assigned_yahoo + yahoo_prior + yahoo_qual)
```

```
##
## Call:
## lm(formula = used_yahoo_q1 ~ assigned_yahoo + yahoo_prior + yahoo_qual,
##     data = dfexp_def_s)
##
## Coefficients:
##     (Intercept)  assigned_yahoo1      yahoo_prior       yahoo_qual
##        -0.10781          0.56504          0.07431          0.03965
```

4.5 Big regression general analysis status quo effect

There si something wrong with the default data set. Showing NAs in assigned to a

```
dfexp_def_s = dfexp_def_s %>%
  mutate(statquo_sq1 = ifelse(assigned_bing==1 & used_bing_q1==1 | assigned_google==1 & used_g
         statquo_sq3= ifelse(assigned_bing==1 & used_bing_q3==1 | assigned_google==1 & used_goo
         prior_use = ifelse(assigned_google==1 & google_prior==1 | assigned_bing==1 & bing_prio
         explor_sq1 = ifelse(str_length(se_beatles)>1, 1,0),
         age = 2023 - as.numeric(q32),
         age_c = case_when(age<30 ~ "18-29",
                           age >29 & age <40 ~ "30-39",
                           age >39 & age <50 ~ "40-49",
                           age>49 & age <60 ~ "50-59",
                           age>59 ~ "60 +"))


lm(data=dfexp_def_s, formula= statquo_sq1 ~  prior_use)
```

```
##
## Call:
## lm(formula = statquo_sq1 ~ prior_use, data = dfexp_def_s)
##
## Coefficients:
## (Intercept)    prior_use
##      0.6000       0.3186
```

```
lm(data=dfexp_def_s, formula= statquo_sq1 ~ prior_use + age_c)
```

```
##
## Call:
## lm(formula = statquo_sq1 ~ prior_use + age_c, data = dfexp_def_s)
```

```
##
## Coefficients:
## (Intercept)    prior_use   age_c30-39   age_c40-49   age_c50-59    age_c60 +
##     0.64683      0.31029     -0.05514     -0.02280     -0.08007     -0.10985
```

Data defaults without Google

```
dfexp_def_s2 = dfexp_def_s %>%
  filter(assigned_google==0)
```

```
lm(data=dfexp_def_s2, formula= statquo_sq1 ~ explor_sq1 + prior_use)
```

```
##
## Call:
## lm(formula = statquo_sq1 ~ explor_sq1 + prior_use, data = dfexp_def_s2)
##
## Coefficients:
## (Intercept)   explor_sq1    prior_use
##      0.6154       0.1091       0.2755
```

```
glmsqs1 = glm(data=dfexp_def_s2, formula= statquo_sq1 ~ explor_sq1 + prior_use, family = "bino

summary(glmsqs1)
```

```
##
## Call:
## glm(formula = statquo_sq1 ~ explor_sq1 + prior_use, family = "binomial",
##     data = dfexp_def_s2)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.4700     0.3291   1.428  0.15330
## explor_sq1    13.4660  1455.3976   0.009  0.99262
## prior_use      1.6301     0.5435   2.999  0.00271 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 100.365  on 94  degrees of freedom
## Residual deviance:  89.877  on 92  degrees of freedom
## AIC: 95.877
##
## Number of Fisher Scoring iterations: 14
```

Interesting coefficients, qualitatively. Now, run glm because p>1

```r
mean(dfexp_def_s$statquo_sq1)
```

```
## [1] 0.8174603
```

4.6 Check if those who use the default use another one too!

This is important. q1 only a few use more than one, q3 many more explore.

```r
# Create var for analysis

dfexp_def_s = dfexp_def_s %>%
  mutate(se_assigned = case_when(assigned_google==1 ~ "Google",
                                 assigned_bing==1 ~ "Bing",
                                 assigned_duckduckgo==1 ~"DuckDuck Go",
                                 assigned_yahoo==1 ~ "Yahoo"))
```

Code below irrelevant. q1 only one person used more than one

```r
# glm_just_one= glm(just_one_sq1 ~ se_assigned, data=dfexp_def_s, family ="binomial")
#
# plot_just_one_sq1 = effect_plot(glm_just_one, pred = "se_assigned", interval = TRUE, plot.po
#
# plot_just_one_sq1

table(dfexp_def_s$se_assigned, dfexp_def_s$just_one_sq1)
```

```
##
##                0  1
##   Bing         1 30
##   DuckDuck Go  0 37
##   Google       0 31
##   Yahoo        0 27
```

```r
table(dfexp_def_s$se_assigned, dfexp_def_s$just_one_sq2)
```

```
##
##                0  1
##   Bing         4 27
##   DuckDuck Go  1 36
##   Google       1 30
##   Yahoo        2 25
```

```
table(dfexp_def_s$se_assigned, dfexp_def_s$just_one_sq3)
```

```
##
##                0  1
##   Bing         2 29
##   DuckDuck Go  2 35
##   Google       0 31
##   Yahoo        1 26
```

Code above is not what I need. The relevant variable is stat_quo_qi * just_one_qi

```
dfexp_def_s = dfexp_def_s %>%
  mutate(just_default_sq1 = ifelse(statquo_sq1==1 & just_one_sq1==1,1,0),
         just_default_sq3 = ifelse(statquo_sq3==1 & just_one_sq3==1,1,0))
```

```
table(dfexp_def_s$se_assigned, dfexp_def_s$just_default_sq1)
```

```
##
##                0  1
##   Bing         7 24
##   DuckDuck Go  4 33
##   Google       2 29
##   Yahoo       11 16
```

```
table(dfexp_def_s$se_assigned, dfexp_def_s$just_default_sq3)
```

```
##
##                0  1
##   Bing        11 20
##   DuckDuck Go 10 27
##   Google       2 29
##   Yahoo       14 13
```

4.6.1 Now with the second experiment

```
datexp = datexp %>%
  mutate(se_assigned = case_when(assigned_google==1 ~ "Google",
                                 assigned_bing==1 ~ "Bing"),
         statquo_sq1 = ifelse(assigned_bing==1 & used_bing_q1==1 | assigned_google==1 & used_g
         statquo_sq2 = ifelse(assigned_bing==1 & used_bing_q2==1 | assigned_google==1 & used_g
         statquo_sq3= ifelse(assigned_bing==1 & used_bing_q3==1 | assigned_google==1 & used_goo
         statquo_sq4 = ifelse(assigned_bing==1 & used_bing_q4==1 | assigned_google==1 & used_g
         statquo_sq5 = ifelse(assigned_bing==1 & used_bing_q5==1 | assigned_google==1 & used_g
```

```
        prior_use = ifelse(assigned_google==1 & google_prior==1 | assigned_bing==1 & bing_pri
        just_default_sq1 = ifelse(statquo_sq1==1 & just_one_sq1==1,1,0),
        just_default_sq2 = ifelse(statquo_sq2==1 & just_one_sq3==1,1,0),
        just_default_sq3 = ifelse(statquo_sq3==1 & just_one_sq3==1,1,0),
        just_default_sq4 = ifelse(statquo_sq4==1 & just_one_sq3==1,1,0),
        just_default_sq5 = ifelse(statquo_sq5==1 & just_one_sq3==1,1,0))
```

```
table(datexp$se_assigned, datexp$just_default_sq1)
```

```
##
##             0   1
##   Bing     45 112
##   Google    9 153
```

```
table(datexp$se_assigned, datexp$just_default_sq2)
```

```
##
##             0   1
##   Bing     67  90
##   Google   12 150
```

```
table(datexp$se_assigned, datexp$just_default_sq3)
```

```
##
##             0   1
##   Bing     78  79
##   Google    9 153
```

```
table(datexp$se_assigned, datexp$just_default_sq4)
```

```
##
##             0   1
##   Bing     82  75
##   Google   12 150
```

```
table(datexp$se_assigned, datexp$just_default_sq5)
```

```
##
##             0   1
##   Bing     84  73
##   Google   18 144
```

Interesting result. More people tend to use Bing exclusively over time.

5 Results Weather Apps

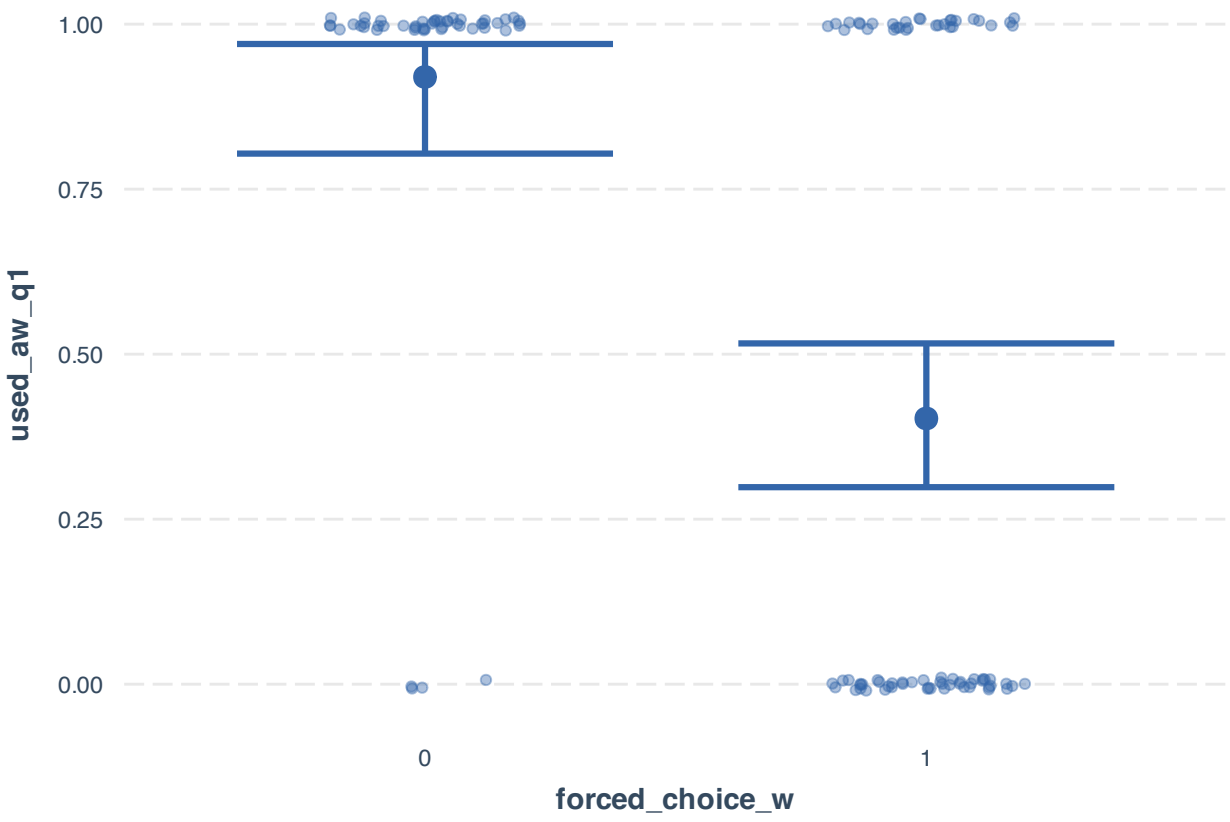
```
# The same for weather apps part

dfexp2 = dfexp %>%
  mutate(chose_wc = if_else(choice_screen_2 == 1, 1, 0, NULL),
         chose_aw = if_else(choice_screen_2 == 2,1,0, NULL),
         chose_wu = if_else(choice_screen_2 == 3, 1,0, NULL),

         choice_wa = factor(case_when(chose_wc == 1 ~ "Weather Channel",
                                      chose_aw == 1 ~ "AccuWeather",
                                      chose_wu == 1 ~ "Weather Underground"),
                                    levels = c("Weather Channel",
                                               "AccuWeather",
                                               "WeatherUnderground")),

         assigned_wc = ifelse(weather_do %in% "default_wc", 1, 0),
         assigned_aw = ifelse(weather_do %in% "default_aw", 1, 0),
         assigned_wu = ifelse(weather_do %in% "default_wu", 1, 0),

         used_wc_q1 = if_else(grepl("1", app_austin), 1, 0, NULL),
         used_aw_q1 = if_else(grepl("2", app_austin),1,0,NULL),
         used_wu_q1 = if_else(grepl("3", app_austin),1,0,NULL),
         used_other_q1 = if_else(grepl("4", app_austin),1,0,NULL),

         used_wc_q2 = if_else(grepl("1", app_cambridge), 1, 0, NULL),
         used_aw_q2 = if_else(grepl("2", app_cambridge),1,0,NULL),
         used_wu_q2 = if_else(grepl("3", app_cambridge),1,0,NULL),

         used_wc_q3 = if_else(grepl("1", app_cupertino), 1, 0, NULL),
         used_aw_q3 = if_else(grepl("2", app_cupertino),1,0,NULL),
         used_wu_q3 = if_else(grepl("3", app_cupertino),1,0,NULL),

         ac_qual = q31_1,
         aw_qual = q31_2,
         wu_qual = q31_3,

         ac_prior = ifelse(grepl("1", q54), 1, 0),
         aw_pior = ifelse(grepl("2", q54),1, 0),
         wu_prior = ifelse(grepl("3", q54), 1, 0),
         other_prior = ifelse(grepl("4", q54), 1, 0))
```

```
dfaw = dfexp2 %>%
  filter(forced_choice_w==1| assigned_aw==1)
```

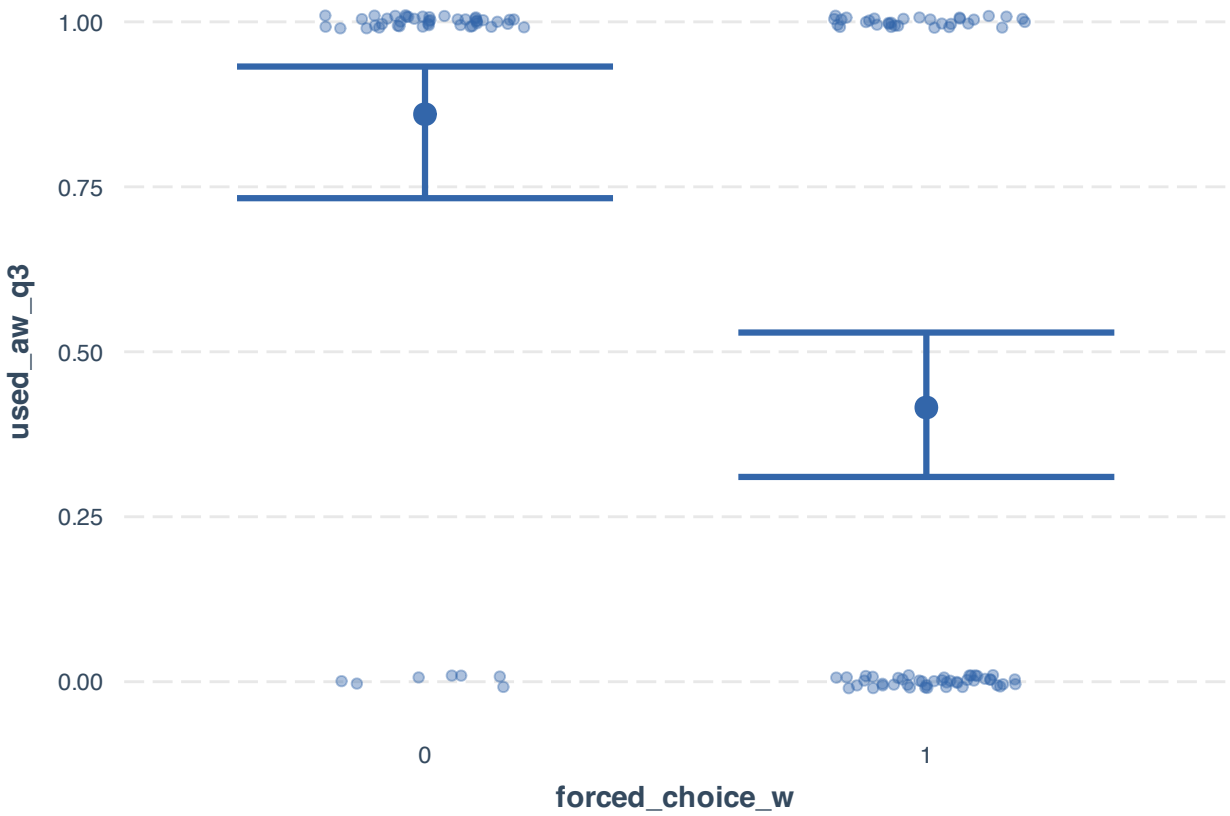
One plot for sq q1 for each (a1_q1, a2_q1, a3_q1)

Another for sq q3 for each (a1_q3, a2_q3, a3_q3)

```
#AccuWeather
sqawq1= glm(used_aw_q1 ~ forced_choice_w, data=dfaw, family ="binomial")

plotawq1 = effect_plot(sqawq1, pred = "forced_choice_w", interval = TRUE, plot.points = TRUE, j

plotawq1
```




```
sqawq3= glm(used_aw_q3 ~ forced_choice_w, data=dfaw, family ="binomial")

plotawq3 = effect_plot(sqawq3, pred = "forced_choice_w", interval = TRUE, plot.points = TRUE, j

plotawq3
```




```
# Convert vars for analysis

test = dfexp2 %>%
  mutate(across(starts_with("used"), as.numeric))

#' *This worked!!! Saves a lot of time. Mind I should use "across" much more. And apply, lappl

test = test %>%
  mutate(nbing = used_bing_q1 + used_bing_q2 + used_bing_q3) # This works too!
```

Effect_plot to show correlations browsers/s.engines controlling for changes in OSX market shares