

Appendix 2: CODE

Contents

CODE CHUNK #1. INITIAL DATA PROCESSING	1
CODE CHUNK #2A. TRAVEL WEBSITES KEYWORDS CATEGORIZATION	5
CODE CHUNK #2B. TRAVEL WEBSITES WEB SCRAPER + UCLASSIFY.COM API	6
CODE CHUNK #3A. ADD PURCHASE DATA	8
CODE CHUNK #3B. ADD TRAVEL CATEGORIZATION	9
CODE CHUNK #4. AGGREGATE DATA	12
CODE CHUNK #5. SURVEY DATA PROCESSING	17
CODE CHUNK #6. MERGE SURVEY DATA AND PROCESSED BEHAVIORAL DATA	18
CODE CHUNK #8. FINAL ANALYSIS AND OLS TESTS	20

CODE CHUNK #1. INITIAL DATA PROCESSING

```
##### LOAD #####
load(file = "DesktopPV.Rdata")
load(file = "MobilePV.Rdata")
##### DESKTOP #####
library(urltools)
# GET THE HOST NAME (DOMAIN + SUBDOMAIN)
testD$host <- suffix_extract(domain(testD$url))[1]$host
visits <- testD
# GET USER IDS
uids <- sort(unique(visits$panelist_id))
# CREATE EMPTY LIST
chunks <- list()
# FILL IN THE LIST WITH DATA PER EACH ID AND CREATE
# MICROMOMENTS
for (ivar in 1:length(uids)) {
  workarr <- subset(visits, visits$panelist_id == uids[ivar])
  print(c("Start", ivar, nrow(workarr), as.character(Sys.time())))
  workarr <- workarr[order(workarr$used_at), ]
  mmid <- 1
  for (ivar2 in 1:(nrow(workarr) - 1)) {
    if (as.POSIXct(workarr$used_at[ivar2]) + workarr$active_second[ivar2] +
        300 < as.POSIXct(workarr$used_at[ivar2 + 1])) {
      workarr$mmid[ivar2] <- mmid
      mmid <- mmid + 1
    } else {
```

```

        workarr$mmid[ivar2] <- mmid
    }
}
if (as.POSIXct(workarr$used_at[nrow(workarr) - 1]) + workarr$active_second[nrow(workarr) -
1] + 300 < as.POSIXct(workarr$used_at[nrow(workarr)])) {
    workarr$mmid[nrow(workarr)] <- mmid
} else {
    workarr$mmid[nrow(workarr)] <- mmid - 1
}
chunks[[ivar]] <- workarr
print(c("End", ivar, nrow(workarr), as.character(Sys.time()))
rm(workarr)
}
# FIX OBS. 184
save(chunks, file = "chunks.Rdata")
##### Aggregate DESKTOP #####
# CREATE EMPTY LIST FOR THE AGGREGATED DATA ON MICROMOMENT
# LEVEL
aggchunks <- list()
# FILL IN THE LIST FOR THE AGGREGATED DATA ON MICROMOMENT
# LEVEL
for (forvar in 1:length(chunks)) {
    testdata <- NULL
    aggtest <- NULL
    testdata <- chunks[[forvar]]
    print(c("Start", nrow(testdata), forvar, as.character(Sys.time())))
    # NUMBER OF DOMAINS VISITS
    aggtest <- aggregate(testdata$host ~ testdata$mmid, data = testdata,
        FUN = function(x) length(unique(x)))
    colnames(aggtest) <- c("mmid", "domains")
    # NUMBER OF PAGEVIEWS VISITS
    aggtest$pageviews <- unlist(aggregate(testdata$url ~ testdata$mmid,
        data = testdata, FUN = length)[2])
    # SUM OF ACTIVE SECONDS
    aggtest$active_seconds <- unlist(aggregate(testdata$active_seconds ~
        testdata$mmid, data = testdata, FUN = sum, na.rm = TRUE)[2])
    # START OF THE MOMENT (MIN VALUE OF USED_AT)
    aggtest$start <- unlist(aggregate(testdata$used_at ~ testdata$mmid,
        data = testdata, function(x) min(x))[2])
    # END OF THE MOMENT (MAX VALUE OF SUM OF USED_AT AND ACTIVE
    # SECONDS OF THE LAST ACTIVITY BEFORE IDLE)
    aggtest$end <- unlist(aggregate(testdata$used_at ~ testdata$mmid,
        data = testdata, function(x) as.POSIXct(max(x)) +
            testdata$active_seconds[length(x)]))[2])
    # CONVERT TO DATE AND GET DIFFERENCE BETWEEN END AND START OF
    # THE MOMENT
    aggtest$start <- strptime(aggtest$start, format = "%Y-%m-%d %T")
    aggtest$end <- strptime(aggtest$end, format = "%Y-%m-%d %T")
    aggtest$length <- difftime(strptime(aggtest$end, format = "%Y-%m-%d %T"),
        strptime(aggtest$start, format = "%Y-%m-%d %T"), units = c("secs"))
    aggtest$app <- NA
    aggtest$device <- "desktop"
    aggchunks[[forvar]] <- aggtest
    print(c("End", nrow(testdata), forvar, as.character(Sys.time())))
    rm(testdata, aggtest)
}
save(aggchunks, file = "aggchunks.Rdata")

```

```
##### MOBILE #####
mobilevisits1 <- testM
# REPLACE EMPTY WITH NA
mobilevisits1[mobilevisits1 == ""] = NA
# REMOVE IDS OCCURRING LESS THAN 5 TIMES IN THE DATA
mobilevisits1 <- mobilevisits1[mobilevisits1$panelist_id %in%
  names(table(mobilevisits1$panelist_id))[table(mobilevisits1$panelist_id) >=
    5], ]
# GET THE HOST NAME (DOMAIN + SUBDOMAIN)
mobilevisits1$host <- suffix_extract(domain(mobilevisits1$url))[1]$host
# CONVERT DURATUION TO NUMERIC AND REMOVE '/N'
mobilevisits1$duration <- as.numeric(mobilevisits1$duration)
mobilevisits1$used_at <- as.character(mobilevisits1$used_at)
mobilevisits1$duration <- ifelse(is.na(mobilevisits1$duration),
  0, mobilevisits1$duration)
mobilevisits1$used_at <- strptime(mobilevisits1$used_at, format = "%Y-%m-%d %T")
# GET USER IDS
uidsm <- unique(mobilevisits1$panelist_id)
# CREATE EMPTY LIST
chunksm <- list()
# FILL IN THE LIST WITH DATA PER EACH ID AND CREATE
# MICROMOMENTS
for (ivar in 1:length(uidsm)) {
  workarr <- subset(mobilevisits1, mobilevisits1$panelist_id ==
    uidsm[ivar])
  print(c("Start", nrow(workarr), ivar, as.character(Sys.time())))
  if (ivar == 83) {
    workarr <- head(workarr, 75000)
  }
  workarr <- workarr[order(workarr$used_at), ]
  mmid <- 1
  for (ivar2 in 1:(nrow(workarr) - 1)) {
    if (as.POSIXct(workarr$used_at[ivar2]) + workarr$duration[ivar2] +
      300 < as.POSIXct(workarr$used_at[ivar2 + 1])) {
      workarr$mmid[ivar2] <- mmid
      mmid <- mmid + 1
    } else {
      workarr$mmid[ivar2] <- mmid
    }
  }
  if (as.POSIXct(workarr$used_at[nrow(workarr) - 1]) + workarr$duration[nrow(workarr) -
    1] + 300 < as.POSIXct(workarr$used_at[nrow(workarr)])) {
    workarr$mmid[nrow(workarr)] <- mmid
  } else {
    workarr$mmid[nrow(workarr)] <- mmid - 1
  }
  chunksm[[ivar]] <- workarr
  rm(workarr)
  print(c("End", ivar, as.character(Sys.time())))
}
save(chunksm, file = "chunksm.Rdata")
##### Aggregate MOBILE #####
aggchunksm <- list()
for (forvar in 1:length(chunksm)) {
  print(c("Start", forvar, as.character(Sys.time())))
  testdata <- NULL
  aggtest <- NULL
}
```

```

testdata <- chunksm[[forvar]]
testdata$used_at <- as.character(testdata$used_at)
aggtest <- aggregate(testdata$panelist_id ~ testdata$mmid,
  FUN = unique)
colnames(aggtest) <- c("mmid", "panelist_id")
aggtest <- aggtest[, c(2, 1)]
aggtest$host <- unlist(aggregate(host ~ panelist_id + mmid,
  data = testdata, FUN = function(x) if (any(!is.na(x))) {
    length(unique(x[which(!is.na(x))]))
  } else {
    x = NA
  }, na.action = na.pass)[3])
# NUMBER OF PAGEVIEWS VISITS
aggtest$pageviews <- unlist(aggregate(url ~ panelist_id +
  mmid, data = testdata, FUN = function(x) if (any(!is.na(x))) {
    length(x[which(!is.na(x))])
  } else {
    x = NA
  }, na.action = na.pass)[3])
# SUM OF ACTIVE SECONDS
aggtest$active_seconds <- unlist(aggregate(duration ~ panelist_id +
  mmid, data = testdata, FUN = sum, na.rm = TRUE)[3])
# START OF THE MOMENT (MIN VALUE OF USED AT)
aggtest$start <- unlist(aggregate(used_at ~ panelist_id +
  mmid, data = testdata, function(x) min(x))[3])
# END OF THE MOMENT (MAX VALUE OF USED_TILL I.E. SUM OF
# USED_AT AND ACTIVE SECONDS OF THE LAST ACTIVITY BEFORE
# IDLE)
aggtest$end <- unlist(aggregate(used_at ~ panelist_id + mmid,
  data = testdata, FUN = function(x) as.character(as.POSIXct(max(x)) +
    testdata$duration[length(x)]))[3])
# CONVERT TO DATE AND GET DIFFERENCE BETWEEN END AND START OF
# THE MOMENT
aggtest$start <- strptime(aggtest$start, format = "%Y-%m-%d %T")
aggtest$end <- strptime(aggtest$end, format = "%Y-%m-%d %T")
aggtest$length <- difftime(strptime(aggtest$end, format = "%Y-%m-%d %T"),
  strptime(aggtest$start, format = "%Y-%m-%d %T"), units = c("secs"))
aggtest$app <- unlist(aggregate(app_id ~ panelist_id + mmid,
  data = testdata, FUN = function(x) if (any(!is.na(x))) {
    length(unique(x[which(!is.na(x))]))
  } else {
    x = NA
  }, na.action = na.pass)[3])
aggtest$device <- "mobile"
aggchunksm[[forvar]] <- aggtest
rm(testdata, aggtest)
print(c("End", forvar, as.character(Sys.time())))
}
save(aggchunksm, file = "aggchunksm.Rdata")
# PRODUCT OF THE SCRIPT THE SCRIPT IS FED WITH THE RAW
# FULLDATASET THE RESULT OF THE SCRIPT IS A LIST WITH
# ELEMENTS CONTAINING THE INDIVIDUAL DATA PER EACH ID
# INCLUDING THE MICROMOMENTS

```

CODE CHUNK #2A. TRAVEL WEBSITES KEYWORDS CATEGORIZATION

```
### APPLY DOMAIN VS. PATH
library(plyr)
##### DOMAINS FROM DEKSTOP AND MOBILE #####
# GET UNIQUE DOMAINS AND FREQ
domains_D <- count(df = testD, vars = "host")
domains_M <- count(df = testM, vars = "host")
# MERGE AND SUM FREQ
domains_full <- merge(domains_D, domains_M, by = "host", all = TRUE)
domains_full$freq <- rowSums(domains_full[c("freq.x", "freq.y")],
  na.rm = TRUE)
# FIND KEYWORDS AND ASSING CLASSIFY
# KEYWORDS
travelwebskeywords <- c("airbnb", "hotel", "hotels", "flight",
  "flights", "air", "booking", "expedia", "trivago", "travel",
  "tourism", "trip", "vacation", "vakant", "reis", "orbitz",
  "kayak", "weekendje", "vliegtickets", "vliegen", "villa")
for (web1 in travelwebskeywords) {
  domains_full[web1] <- ifelse(grepl(pattern = web1, x = domains_full$host,
    ignore.case = TRUE), 1, 0)
}
domains_full$book <- ifelse(grepl(pattern = "book\\.", x = domains_full$host,
  ignore.case = TRUE), 1, 0)
domains_full[grepl(pattern = "facebook", x = domains_full$host,
  ignore.case = TRUE), c("book")] <- 0
domains_full[grepl(pattern = "solitaire", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "hair", x = domains_full$host, ignore.case = TRUE),
  c("air")] <- 0
domains_full[grepl(pattern = "fair", x = domains_full$host, ignore.case = TRUE),
  c("air")] <- 0
domains_full[grepl(pattern = "millionaire", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "billionaire", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "questionnaire", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "airfry", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "repair", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "clair", x = domains_full$host, ignore.case = TRUE),
  c("air")] <- 0
domains_full[grepl(pattern = "nuclair", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "airmax", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "primaire", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "flair", x = domains_full$host, ignore.case = TRUE),
  c("air")] <- 0
domains_full[grepl(pattern = "miljonair", x = domains_full$host,
  ignore.case = TRUE), c("air")] <- 0
```

```

domains_full[grepl(pattern = "ipad", x = domains_full$host, ignore.case = TRUE),
c("air")] <- 0
domains_full[grepl(pattern = "airbag", x = domains_full$host,
ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "airco", x = domains_full$host, ignore.case = TRUE),
c("air")] <- 0
domains_full[grepl(pattern = "airfilter", x = domains_full$host,
ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "puzzel", x = domains_full$host,
ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "polair", x = domains_full$host,
ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "aupair", x = domains_full$host,
ignore.case = TRUE), c("air")] <- 0
domains_full[grepl(pattern = "kayako", x = domains_full$host,
ignore.case = TRUE), c("kayak")] <- 0
# CREATE IS TRAVEL
domains_full$IsTravel <- apply(domains_full[5:ncol(domains_full)],
1, FUN = function(x) if (any(x == 1)) {
1
} else {
0
})
##### APPS #####
Appsused <- count(testM, "app_name")
for (app1 in travelwebskeywords) {
Appsused[app1] <- ifelse(grepl(pattern = app1, x = Appsused$app_name,
ignore.case = TRUE), 1, 0)
}
Appsused[grepl(pattern = "solitaire", x = Appsused$app_name, ignore.case = TRUE),
c("air")] <- 0
Appsused$TravelApp <- apply(Appsused[travelwebskeywords], 1,
FUN = function(x) if (any(x == 1)) {
1
} else {
0
})
# PRODUCT OF THE SCRIPT THE SCRIPT TAKES THE UNIQUE DOMAINS
# VISITED VIA DESKTOP AND MOBILE ALSO THE APPS THE RESULT OF
# THE SCRIPT IS A DATAFRAME WITH CATEGORIES PER EACH DOMAIN
# BASED ON KEYWORDS

```

CODE CHUNK #2B. TRAVEL WEBSITES WEB SCRAPER + UCLASSIFY.COM API

```

### ABOVE 10 LABRARIES
library(httr)
library(jsonlite)
library(stringr)
library(Hmisc)
# AGG OVER HOST

```

```

testclassifytest1 <- aggregate(fullwebvisits$host, by = list(fullwebvisits$host),
  length)
# RENAME
colnames(testclassifytest1) <- c("url", "freq")
# testclassifytest2 <- subset(testclassifytest1, freq>10)
testclassify <- subset(testclassifytest1, freq > 10)
# GET COLNAMES IN VECTOR
cnames <- c("Arts and Entertainment", "Autos", "Businss Finance",
  "Celebrity", "College", "Cooking", "Dating and Romance",
  "Exercise", "Fashion and Beauty", "Games", "Health", "Home Improvement",
  "News", "Parents and Family", "Technology", "Travel")
# CREATE COLNAME FOR EACH NAME OF COLNAME VECTOR
for (i in cnames) {
  testclassify[, i] <- NA
}
for (item in 1:nrow(testclassify)) {
  # for (item in 1:50) {
  # TRY TO OPEN CONNECTION
  try(fullhtml <- GET(paste0("http://", testclassify$url[item])))
  if (exists("fullhtml")) {
    # SCRAPE THE TEXT
    contenthtml <- content(fullhtml, "text")
    try(onlytest <- htmlParse(contenthtml, asText = TRUE))
    # try ( plaintext <- xpathSApply(onlytest, '//p', xmlValue) )
    try(plaintext <- xpathSApply(onlytest, "//text()[not(ancestor::script)][not(ancestor::style)][not(ancestor::a)]", xmlValue))
    text <- paste(plaintext, collapse = "")
    text <- str_replace_all(text, "[\r\n\t]", " ")
    text <- str_replace_all(text, "[[:punct:]]", "")
    text <- str_replace_all(text, " +", "+")
    # IF THERE IS TEXT
    if (text != "") {
      # ENCODE AS URL
      encodedurlstring <- URLencode(text, reserved = FALSE,
        repeated = FALSE)
      encodedurlstring <- str_replace_all(encodedurlstring,
        "%2", "+")
      apiurlstring <- "https://api.uclassify.com/v1/ephraimbaro/Categories/classify/?readKey=70rn6c"
      fullurlstring <- paste0(apiurlstring, encodedurlstring,
        collapse = "")
      # SEND IT TO THE UCLASSIFY API
      try(test1 <- fromJSON(fullurlstring))
      # FILL THE DATA IN DATAFRAME
      testclassify[item, cnames] <- data.frame(matrix(unlist(test1),
        nrow = 1, byrow = T))
      print(item)
    }
    rm(fullhtml)
  }
}
# GET THE HIGHEST CATEGORY
testclassify$class <- unlist(apply(testclassify[5:20], 1, function(x) if (!is.na(x[1]) &&
  x[1] != 0.0625) {
  which(x == max(x))
} else {
  17
}))

```



```

cnames2 <- c(cnames, "NO category")
# GET CATEGORY NAME
testclassify$class1 <- unlist(lapply(testclassify$class, function(x) if (is.numeric(x)) {
  cnames2[as.numeric(x)]
})))
# REMOVE SCIENTIFIC NOTATION options(scipen=999)
# save(testclassify, file =
# 'testclassify_2907_until_48k.Rdata')
# PERCENTAGE VECTOR NAMES
percentagecols <- paste0("P", cnames)
# PERCENTAGE CREATE EMPTY COLUMNS
for (i in percentagecols) testclassify[, i] <- NA
# CREATE COLUMNS IN PERCENTAGES
for (ivar in 1:16) {
  shift = sum(ivar, 4)
  testclassify[percentagecols[ivar]] <- round(testclassify[shift]/rowSums(testclassify[5:20]),
    3)
}
describe(testclassify$class1)
class48k <- head(testclassify, 47818)
save(class48k, file = "class48k.Rdata")
# PRODUCT OF THE SCRIPT THE SCRIPT TAKES THE UNIQUE DOMAINS
# VISITED VIA DESKTOP AND MOBILE ALSO THE APPS IT VISITS THE
# DOMAIN AND SCRAPES DOWN THE INFORMATION STORES IT LOCALLY
# AND SENDS IT OUT TO UCLASSIFY.COM THE RESULT OF THE SCRIPT
# IS A DATAFRAME WITH CATEGORIES PER EACH DOMAIN BASED

```

CODE CHUNK #3A. ADD PURCHASE DATA

```

#### MERGE PURCHASE INTO DESKTOP CHUNKS
# LOAD PURCHASE
library(lubridate)
purchase <- read.csv("gdselect-pageviews-20160101-20160731.csv",
  stringsAsFactors = FALSE)
for (ivar1 in 1:length(uids)) {
  cid <- uids[ivar1]
  workarr <- chunks[[ivar1]]
  workarr$purchase <- NULL
  purchworkarr <- subset(purchase, purchase$ID == cid)
  print(c("start", nrow(workarr), nrow(purchworkarr), ivar1,
    as.character(Sys.time()))))
  if (nrow(purchworkarr) >= 1) {
    purchworkarr <- purchworkarr[c("url", "used_at")]
    purchworkarr$used_at <- as.numeric(ymd_hms(purchworkarr$used_at))
    purchworkarr <- ddply(purchworkarr, "url", numcolwise(min))
    purchworkarr$url <- NULL
    purchworkarr$purchase <- 1
    purchworkarr$used_at <- as.character(as.POSIXct(purchworkarr$used_at,
      origin = "1970-01-01 00:00.00"))
    workarr <- merge(workarr, purchworkarr, by.x = "used_at",
      by.y = "used_at", all.x = TRUE)
  }
}

```



```

    chunks[[ivar1]] <- workarr
  }
  print(c("end", nrow(workarr), nrow(purchworkarr), ivar1,
    as.character(Sys.time())))
  rm(workarr)
}
save(chunks, file = "chunks.Rdata")
#### MERGE PURCHASE INTO MOBILE CHUNKS
# LOAD PURCHASE
purchasem <- read.csv("gdselect-mobile-pageviews-20160101-20160731.csv",
  stringsAsFactors = FALSE)
for (ivar1 in 1:length(uidsm)) {
  cid <- uidsm[ivar1]
  workarr <- chunksm[[ivar1]]
  workarr$purchase <- NULL
  purchworkarr <- subset(purchasem, purchasem$ID == cid)
  print(c("start", nrow(workarr), nrow(purchworkarr), ivar1,
    as.character(Sys.time())))
  if (nrow(purchworkarr) >= 1) {
    purchworkarr <- purchworkarr[c("url", "used_at")]
    purchworkarr$used_at <- as.numeric(ymd_hms(purchworkarr$used_at))
    purchworkarr <- ddply(purchworkarr, "url", numcolwise(min))
    purchworkarr$url <- NULL
    purchworkarr$purchase <- 1
    purchworkarr$used_at <- as.character(as.POSIXct(purchworkarr$used_at,
      origin = "1970-01-01 00:00:00"))
    workarr$used_at <- as.character(workarr$used_at)
    workarr <- merge(workarr, purchworkarr, by.x = "used_at",
      by.y = "used_at", all.x = TRUE)
    chunksm[[ivar1]] <- workarr
  }
  print(c("end", nrow(workarr), nrow(purchworkarr), ivar1,
    as.character(Sys.time())))
  rm(workarr)
}

```

CODE CHUNK #3B. ADD TRAVEL CATEGORIZATION

```

#### MERGE DOMAINS FROM SCRAPER AND KEYWORDS
domains_full_classified <- merge(domains_full, class48k, by.x = "host",
  by.y = "url", all = TRUE)
domains_full_classified$class_full <- ifelse(is.na(domains_full_classified$class),
  0, domains_full_classified$class)
domains_full_classified$Class_Travel <- ifelse(domains_full_classified$class_full ==
  16, 1, 0)
domains_full_classified$Class_Travel <- ifelse(domains_full_classified$IsTravel ==
  1, 1, domains_full_classified$Class_Travel)
domain_travel <- domains_full_classified[c("host", "freq", "Class_Travel")]
# FINAL LIST OF DOMAINS
save(domain_travel, file = "final_domain_list.Rdata")
#### MERGE TRAVEL INTO DESKTOP CHUNKS

```

```

# MERGING TRAVEL DOMAINS WITH THE CHUNKS
for (ivar in 1:length(chunks)) {
  # GET FROM THE LIST
  workarr <- chunks[[ivar]]
  print(c("start", nrow(workarr), ivar, as.character(Sys.time())))
  # MERGE
  workarr <- merge(workarr, domain_travel[c("host", "Class_Travel")],
    by.x = "host", by.y = "host", all.x = TRUE)
  # BACK TO THE LIST
  chunks[[ivar]] <- workarr
  print(c("end", nrow(workarr), ivar, as.character(Sys.time())))
  rm(workarr)
}

save(chunks, file = "chunks.Rdata")
for (ivar in 1:length(chunks)) {
  # GET FROM THE LIST
  workarr <- chunks[[ivar]]
  aggarr <- aggchunks[[ivar]]
  print(c("start", nrow(workarr), nrow(aggarr), ivar, as.character(Sys.time())))
  # WAS TRAVEL MOMENT
  aggarr$travel <- unlist(aggregate(Class_Travel ~ mmid, data = workarr,
    FUN = function(x) if (sum(x, na.rm = TRUE) > 0) {
      1
    } else {
      0
    }, na.action = na.pass)[2])
  # Travel Domains
  aggarr$traveldomain <- unlist(merge(aggarr, subset(aggregate(host ~
    Class_Travel + mmid, data = workarr, FUN = function(x) length(unique(x))),
    Class_Travel == 1)[c("mmid", "host")], by = "mmid", all.x = TRUE)[c("host")])
  # NUMBER OF TRAVEL PAGEVIEWS
  aggarr$travelPV <- unlist(aggregate(Class_Travel ~ mmid,
    data = workarr, FUN = function(x) if (any(!is.na(x))) {
      length(which(x == 1))
    } else {
      0
    }, na.action = na.pass)[2])
  # BACK TO THE LIST
  aggchunks[[ivar]] <- aggarr
  print(c("end", nrow(workarr), nrow(aggarr), ivar, as.character(Sys.time())))
  rm(workarr)
  rm(aggarr)
}

save(aggchunks, file = "aggchunks.Rdata")
#### MERGE TRAVEL INTO MOBILE CHUNKS
# MERGING TRAVEL DOMAINS AND APPS WITH THE CHUNKS
for (ivar in 1:length(chunksm)) {
  # GET FROM THE LIST
  workarr <- chunksm[[ivar]]
  print(c("start", nrow(workarr), ivar, as.character(Sys.time())))
  # MERGE
  workarr <- merge(workarr, domain_travel[c("host", "Class_Travel")],
    by.x = "host", by.y = "host", all.x = TRUE)
  workarr <- merge(workarr, Appused[c("app_name", "TravelApp")],
    by = "app_name", all.x = TRUE)
  # BACK TO THE LIST
  chunksm[[ivar]] <- workarr

```

```

print(c("end", nrow(workarr), ivar, as.character(Sys.time())))
rm(workarr)
}
save(chunksm, file = "chunksm.Rdata")
#
for (ivar in 1:length(chunksm)) {
  print(c("start", ivar, as.character(Sys.time())))
  # GET FROM THE LIST
  workarr <- chunksm[[ivar]]
  aggarr <- aggchunksm[[ivar]]
  # WAS TRAVEL MOMENT aggarr$travel <- unlist( aggregate(travel
  # ~ mmid, data=workarr, FUN = function(x) if(sum(x, na.rm =
  # TRUE)>0){1}else{0}, na.action = na.pass)[2] ) Travel
  # Domains aggarr$traveldomain <- unlist(merge(aggarr,
  # subset(aggregate( host ~ travel + mmid, data=workarr, FUN =
  # function(x) length(unique(x)) ),
  # travel==1)[c('mmid','host')], by = 'mmid', all.x =
  # TRUE)[c('host')]) NUMBER OF TRAVEL PAGEVIEWS
  # aggarr$travelPV <- unlist( aggregate(travel ~ mmid,
  # data=workarr, FUN =
  # function(x)if(any(!is.na(x))){length(which(x==1))}else{0},
  # na.action = na.pass)[2] )
  # WAS TRAVEL WEB MOMENT
  aggarr$travel_mm_w <- unlist(aggregate(Class_Travel ~ mmid,
    data = workarr, FUN = function(x) if (any(x[which(!is.na(x))] >
    0)) {
      1
    } else {
      0
    }, na.action = na.pass)[2])
  aggarr$travel_mm_a <- unlist(aggregate(TravelApp ~ mmid,
    data = workarr, FUN = function(x) if (any(x[which(!is.na(x))] >
    0)) {
      1
    } else {
      0
    }, na.action = na.pass)[2])
  aggarr$travel <- ifelse(rowSums(aggarr[c("travel_mm_w", "travel_mm_a")]) >
    0, 1, 0)
  # NUMBER OF TRAVEL PAGEVIEWS aggarr$travelPV <- unlist(
  # aggregate(Class_Travel ~ mmid, data=workarr, FUN =
  # function(x)if(any(!is.na(x))){length(which(x==1))}else{0},
  # na.action = na.pass)[2] ) WAS TRAVEL APP MOMENT
  # aggarr$travelapp <- unlist( aggregate(TravelApp ~ mmid,
  # data=workarr, FUN = function(x)if(sum(x, na.rm =
  # TRUE)>0){1}else{0}, na.action = na.pass)[2] ) Travel
  # Domains DIRTY PATCH BUT FUCK THOSE PPL
  if (ivar != 52 & ivar != 75 & ivar != 76 & ivar != 80 & ivar !=
  94) {
    aggarr$traveldomain <- unlist(merge(aggarr, subset(aggregate(host ~
      Class_Travel + mmid, data = workarr, FUN = function(x) length(unique(x))),
      Class_Travel == 1)[c("mmid", "host")], by = "mmid",
      all.x = TRUE)[c("host.y")])
  } else {
    aggarr$traveldomain <- NA
  }
  # NUMBER OF TRAVEL PAGEVIEWS

```

```

aggarr$travelPV <- unlist(aggregate(Class_Travel ~ mmid,
  data = workarr, FUN = function(x) if (any(!is.na(x))) {
    length(which(x == 1))
  } else {
    0
  }, na.action = na.pass)[2])
# NUMBER OF TRAVEL APPS
aggarr$travel_app <- unlist(merge(aggarr, subset(aggregate(app_name ~
  TravelApp + mmid, data = workarr, FUN = function(x) length(unique(x))),
  TravelApp == 1)[c("mmid", "app_name")], by = "mmid",
  all.x = TRUE)[c("app_name")])
# BACK TO THE LIST
aggchunksm[[ivar]] <- aggarr
print(c("end", ivar, as.character(Sys.time())))
rm(aggarr, workarr)
}
save(aggchunksm, file = "aggchunksm.Rdata")

```

CODE CHUNK #4. AGGREGATE DATA

```

##### AGGREGATE DESKTOP #####
# FAUTY ONES
aggchunks[[431]]$end[which(is.na(aggchunks[[431]]$end))] <- as.POSIXct("2016-03-08 23:59:20")
aggchunks[[325]]$end[which(is.na(aggchunks[[325]]$end))] <- as.POSIXct("2016-01-13 23:58:14")
aggchunks[[224]]$end[which(is.na(aggchunks[[224]]$end))] <- as.POSIXct("2016-02-18 23:56:08")
aggchunks[[431]]$length[which(is.na(aggchunks[[431]]$length))] <- difftime(strptime(aggchunks[[431]]$end[526],
  format = "%Y-%m-%d %T"), strptime(aggchunks[[431]]$start[526],
  format = "%Y-%m-%d %T"), units = c("secs"))
aggchunks[[325]]$length[which(is.na(aggchunks[[325]]$length))] <- difftime(strptime(aggchunks[[325]]$end[91],
  format = "%Y-%m-%d %T"), strptime(aggchunks[[325]]$start[91],
  format = "%Y-%m-%d %T"), units = c("secs"))
aggchunks[[224]]$length[which(is.na(aggchunks[[224]]$length))] <- difftime(strptime(aggchunks[[224]]$end[516],
  format = "%Y-%m-%d %T"), strptime(aggchunks[[224]]$start[516],
  format = "%Y-%m-%d %T"), units = c("secs"))
aggall <- data.frame()
for (ivar in 1:length(uids)) {
  uid <- uids[ivar]
  aggarr <- aggchunks[[ivar]]
  aggarr$id <- uid
  aggall <- rbind(aggall, aggarr)
}
fullagginfo <- aggregate(url ~ panelist_id, data = testD, FUN = length)
colnames(fullagginfo) <- c("ID", "obs")
# DOMAINS
fullagginfo$domains <- unlist(aggregate(host ~ panelist_id, data = testD,
  FUN = function(x) length(unique(x)))[2])
# PAGEVIEWS fullagginfo$domains_PV <- unlist(aggregate(url ~
# panelist_id, data = testD, FUN = function(x)
# length(unique(x)))[2])
fullagginfo$domains_PV <- unlist(aggregate(url ~ panelist_id,
  data = testD, FUN = function(x) length(x))[2])

```

```

# MICROMOMENTS
fullagginfo$mm <- unlist(aggregate(mmid ~ id, data = aggall,
  FUN = function(x) length(x))[2])
# TRAVEL DOMAINS
fullagginfo$travel_domains <- unlist(aggregate(traveldomain ~
  id, data = aggall, FUN = function(x) if (any(!is.na(x))) {
    sum(x[which(!is.na(x))])
  } else {
    0
  }, na.action = na.pass)[2])
# TRAVEL PAGEVIEWS
fullagginfo$travel_domains_PV <- unlist(aggregate(travelPV ~
  id, data = aggall, FUN = function(x) if (any(!is.na(x))) {
    sum(x[which(!is.na(x))])
  } else {
    0
  }, na.action = na.pass)[2])
# TRAVEL MICORMOMENTS
fullagginfo$travel_mm <- unlist(aggregate(travel ~ id, data = aggall,
  FUN = function(x) sum(x))[2])
for (ivar in 1:length(uids)) {
  print(c(ivar, as.character(Sys.time())))
  fullagginfo$total_time[ivar] <- unlist(aggregate(Class_Travel ~
    panelist_id, data = chunks[[ivar]], FUN = function(x) sum(chunks[[ivar]]$active_seconds))[2])
  fullagginfo$total_time_nottravel[ivar] <- unlist(aggregate(Class_Travel ~
    panelist_id, data = chunks[[ivar]], FUN = function(x) sum(chunks[[ivar]]$active_seconds[which(x ==
    0)]))[2])
  fullagginfo$total_time_travel[ivar] <- unlist(aggregate(Class_Travel ~
    panelist_id, data = chunks[[ivar]], FUN = function(x) sum(chunks[[ivar]]$active_seconds[which(x !=
    0)]))[2])
  fullagginfo$total_time2[[ivar]] <- as.numeric(sum(aggchunks[[ivar]]$length))
  fullagginfo$total_time_nottravel2[ivar] <- as.numeric(sum(aggchunks[[ivar]]$length[which(aggchunks[[ivar]]
    0)]))
  fullagginfo$total_time_travel2[ivar] <- as.numeric(sum(aggchunks[[ivar]]$length[which(aggchunks[[ivar]]
    1)]))
}
for (ivar in 1:length(uids)) {
  print(c(ivar, as.character(Sys.time())))
  fullagginfo$f_date[ivar] <- as.character(min(aggchunks[[ivar]]$start))
  fullagginfo$l_date[ivar] <- as.character(max(aggchunks[[ivar]]$end))
  fullagginfo$purchase[ivar] <- sum(chunks[[ivar]]$purchase,
    na.rm = TRUE)
}
fullagginfo$days_act <- round(difftime(fullagginfo$l_date, fullagginfo$f_date,
  units = "d"))
fullagginfo$log_travel_mm <- log(fullagginfo$travel_mm)
fullagginfo$log_travel_domains <- log(fullagginfo$travel_domains)
fullagginfo$log_travel_domains_PV <- log(fullagginfo$travel_domains_PV)
fullagginfo$log_total_time_travel <- log(fullagginfo$total_time_travel)
fullagginfo$log_purchase <- log(fullagginfo$purchase)
fullagginfo$d_purchase <- ifelse(fullagginfo$purchase < mean(fullagginfo$purchase),
  0, 1)
fullagginfo$log_mm <- log(fullagginfo$mm)
fullagginfo$log_domains <- log(fullagginfo$domains)
fullagginfo$log_domains_PV <- log(fullagginfo$domains_PV)
fullagginfo$log_time <- log(fullagginfo$total_time)
fullagginfo$log_time2 <- log(fullagginfo$total_time2)

```

```

fullagginfo$log_total_time_travel2 <- log(fullagginfo$total_time_travel2)
fullagginfo$log_purchase <- ifelse(fullagginfo$log_purchase ==
  "-Inf", 0, fullagginfo$log_purchase)
fullagginfo$device <- 1
fullagginfo$act_mm <- ifelse(fullagginfo$mm < mean(fullagginfo$mm),
  0, 1)
fullagginfo$act_total_time <- ifelse(fullagginfo$total_time <
  mean(fullagginfo$total_time), 0, 1)
fullagginfo$act_total_time2 <- ifelse(fullagginfo$total_time2 <
  mean(fullagginfo$total_time2, na.rm = TRUE), 0, 1)
fullagginfo$act_domains <- ifelse(fullagginfo$domains < mean(fullagginfo$domains),
  0, 1)
fullagginfo$act_domains_PV <- ifelse(fullagginfo$domains_PV <
  mean(fullagginfo$domains_PV), 0, 1)
fullagginfo$share_mm <- (fullagginfo$travel_mm/as.numeric(fullagginfo$days_act))/(fullagginfo$mm/as.numeric(fullagginfo$days_act))
fullagginfo$share_total_time <- (fullagginfo$total_time_travel/as.numeric(fullagginfo$days_act))/(fullagginfo$mm/as.numeric(fullagginfo$days_act))
fullagginfo$share_total_time2 <- (fullagginfo$total_time_travel2/as.numeric(fullagginfo$days_act))/(fullagginfo$mm/as.numeric(fullagginfo$days_act))
fullagginfo$share_domains <- (fullagginfo$travel_domains/as.numeric(fullagginfo$days_act))/(fullagginfo$domains/as.numeric(fullagginfo$days_act))
fullagginfo$share_domains_PV <- (fullagginfo$travel_domains_PV/as.numeric(fullagginfo$days_act))/(fullagginfo$domains_PV/as.numeric(fullagginfo$days_act))
save(fullagginfo, file = "fullagginfo.Rdata")
##### AGGREGATE MOBILE #####
aggallm <- data.frame()
for (ivar in 1:length(uidsm)) {
  uid <- uidsm[ivar]
  aggarr <- aggchunksm[[ivar]]
  aggarr$id <- uid
  aggallm <- rbind(aggallm, aggarr)
}
fullagginfom <- aggregate(url ~ panelist_id, data = mobilevisits1,
  FUN = length, na.action = na.pass)
colnames(fullagginfom) <- c("ID", "obs")
# DOMAINS
fullagginfom$domains <- unlist(aggregate(host ~ panelist_id,
  data = mobilevisits1, FUN = function(x) length(unique(x)),
  na.action = na.pass)[2])
# PAGEVIEWS fullagginfom$domains_PV <- unlist(aggregate(url
# ~ panelist_id, data = mobilevisits1, FUN = function(x)
# length(unique(x)) , na.action = na.pass)[2])
fullagginfom$domains_PV <- unlist(aggregate(url ~ panelist_id,
  data = mobilevisits1, FUN = function(x) length(x), na.action = na.pass)[2])
# MICROMOMENTS
fullagginfom$mm <- unlist(aggregate(mmid ~ id, data = aggallm,
  FUN = function(x) length(x), na.action = na.pass)[2])
# TRAVEL DOMAINS
fullagginfom$travel_domains <- unlist(aggregate(travel ~ id,
  data = aggallm, FUN = function(x) if (any(!is.na(x))) {
    sum(x[which(!is.na(x))])
  } else {
    0
  }, na.action = na.pass)[2])
# TRAVEL PAGEVIEWS
fullagginfom$travel_domains_PV <- unlist(aggregate(travelPV ~
  id, data = aggallm, FUN = function(x) if (any(!is.na(x))) {
    sum(x[which(!is.na(x))])
  } else {
    0
  }, na.action = na.pass)[2])

```

TRAVEL MICORMOMENTS

```

fullagginfom$travel_mm_web <- unlist(aggregate(travel_mm_w ~
  id, data = aggallm, FUN = function(x) if (any(!is.na(x))) {
    sum(x[which(!is.na(x))])
  } else {
    0
  }, na.action = na.pass)[2])
fullagginfom$travel_mm_app <- unlist(aggregate(travel_mm_a ~
  id, data = aggallm, FUN = function(x) if (any(!is.na(x))) {
    sum(x[which(!is.na(x))])
  } else {
    0
  }, na.action = na.pass)[2])
fullagginfom$travel_mm <- rowSums(fullagginfom[c("travel_mm_web",
  "travel_mm_app")], na.rm = TRUE)
fullagginfom <- fullagginfom[match(uidsm, fullagginfom$ID), ]
for (ivar in 1:length(uidsm)) {
  print(c(ivar, as.character(Sys.time())))
  fullagginfom$total_time[[ivar]] <- as.numeric(sum(aggchunksm[[ivar]]$active_seconds))
  fullagginfom$total_time_nottravel[[ivar]] <- as.numeric(sum(aggchunksm[[ivar]]$active_seconds[which(aggchunksm[[ivar]]$active_seconds == 0)]))
  fullagginfom$total_time_travel[[ivar]] <- as.numeric(sum(aggchunksm[[ivar]]$active_seconds[which(aggchunksm[[ivar]]$active_seconds != 0)]))
  fullagginfom$total_time2[[ivar]] <- as.numeric(sum(aggchunksm[[ivar]]$length))
  fullagginfom$total_time_nottravel2[[ivar]] <- as.numeric(sum(aggchunksm[[ivar]]$length[which(aggchunksm[[ivar]]$length == 0)]))
  fullagginfom$total_time_travel2[[ivar]] <- as.numeric(sum(aggchunksm[[ivar]]$length[which(aggchunksm[[ivar]]$length != 0)]))
}
# aggregate(travel ~ panelist_id, data = aggchunksm[[1]], FUN
# = function(x) sum(
# as.numeric(aggchunksm[[1]]$active_seconds) ) )
# aggregate(travel ~ panelist_id, data = aggchunksm[[1]], FUN
# = function(x) sum( as.numeric(aggchunksm[[1]]$length) ) )
# fullagginfom$total_time_travel <- rowSums(
# fullagginfom[c('total_time_travel_web',
# 'total_time_travel_app')], na.rm = TRUE)
fullagginfom$app <- unlist(aggregate(app_name ~ panelist_id,
  data = mobilevisits1, FUN = function(x) length(unique(x)),
  na.action = na.pass)[2])
for (ivar in 1:length(uidsm)) {
  print(c(ivar, as.character(Sys.time())))
  fullagginfom$f_date[[ivar]] <- as.character(min(aggchunksm[[ivar]]$start))
  fullagginfom$l_date[[ivar]] <- as.character(max(aggchunksm[[ivar]]$end))
  fullagginfom$purchase[[ivar]] <- sum(chunksm[[ivar]]$purchase,
    na.rm = TRUE)
}
fullagginfom$days_act <- round(difftime(fullagginfom$l_date,
  fullagginfom$f_date, units = "d"))
fullagginfom$log_travel_mm <- log(fullagginfom$travel_mm)
fullagginfom$log_travel_domains <- log(fullagginfom$travel_domains)
fullagginfom$log_travel_domains_PV <- log(fullagginfom$travel_domains_PV)
fullagginfom$log_total_time_travel <- log(fullagginfom$total_time_travel)
fullagginfom$log_travel_mm <- ifelse(fullagginfom$log_travel_mm ==
  "-Inf", 0, fullagginfom$log_travel_mm)
fullagginfom$log_travel_domains <- ifelse(fullagginfom$log_travel_domains ==
  "-Inf", 0, fullagginfom$log_travel_domains)

```



```

fullagginfom$log_travel_domains_PV <- ifelse(fullagginfom$log_travel_domains_PV ==
  "-Inf", 0, fullagginfom$log_travel_domains_PV)
fullagginfom$log_total_time_travel <- ifelse(fullagginfom$log_total_time_travel ==
  "-Inf", 0, fullagginfom$log_total_time_travel)
fullagginfom$log_mm <- log(fullagginfom$mm)
fullagginfom$log_domains <- log(fullagginfom$domains)
fullagginfom$log_domains_PV <- log(fullagginfom$domains_PV)
fullagginfom$log_time <- log(fullagginfom$total_time)
fullagginfom$log_time2 <- log(fullagginfom$total_time2)
fullagginfom$log_total_time_travel2 <- log(fullagginfom$total_time_travel2)
fullagginfom$log_mm <- ifelse(fullagginfom$log_mm == "-Inf",
  0, fullagginfom$log_mm)
fullagginfom$log_domains <- ifelse(fullagginfom$log_domains ==
  "-Inf", 0, fullagginfom$log_domains)
fullagginfom$log_domains_PV <- ifelse(fullagginfom$log_domains_PV ==
  "-Inf", 0, fullagginfom$log_domains_PV)
fullagginfom$log_time <- ifelse(fullagginfom$log_time == "-Inf",
  0, fullagginfom$log_time)
fullagginfom$log_total_time_travel2 <- ifelse(fullagginfom$log_total_time_travel2 ==
  "-Inf", 0, fullagginfom$log_total_time_travel2)
fullagginfom$device <- 2
fullagginfom$act_mm <- ifelse(fullagginfom$mm < mean(fullagginfom$mm),
  0, 1)
fullagginfom$act_total_time <- ifelse(fullagginfom$total_time <
  mean(fullagginfom$total_time), 0, 1)
fullagginfom$act_total_time2 <- ifelse(fullagginfom$total_time2 <
  mean(fullagginfom$total_time2, na.rm = TRUE), 0, 1)
fullagginfom$act_domains <- ifelse(fullagginfom$domains < mean(fullagginfom$domains),
  0, 1)
fullagginfom$act_domains_PV <- ifelse(fullagginfom$domains_PV <
  mean(fullagginfom$domains_PV), 0, 1)
save(fullagginfom, file = "fullagginfom.Rdata")
##### AGGREGATE DESKTOP AND MOBILE ####
commonvarnames <- colnames(fullagginfo)[c(1:9, 11:12, 14, 17:18,
  30)]
library(plyr)
# fullagginfo_all <- rbind(fullagginfo[c(
# colnames(fullagginfo)[c(1:8,11,19,20,9,12,14)] )],
# fullagginfom[c(
# colnames(fullagginfom)[c(1:7,10,16,26,20,11,26,28)] )] )
fullagginfo_all <- rbind(fullagginfo[commonvarnames], fullagginfom[commonvarnames])
fullagginfo_all <- ddply(fullagginfo_all, "ID", numcolwise(sum))
fullagginfo_all_datearr <- merge(aggregate(f_date ~ ID, data = rbind(fullagginfo[c(colnames(fullagginfo)[c(
  15:16)])], fullagginfom[c(colnames(fullagginfom)[c(1, 18:19)])]),
  min), aggregate(l_date ~ ID, data = rbind(fullagginfo[c(colnames(fullagginfo)[c(1,
  15:16)])], fullagginfom[c(colnames(fullagginfom)[c(1, 18:19)])]),
  max), by = "ID", all = TRUE)
fullagginfo_all <- merge(fullagginfo_all, fullagginfo_all_datearr,
  by = "ID", all = TRUE)
fullagginfo_all$days_act <- round(difftime(fullagginfo_all$l_date,
  fullagginfo_all$f_date, units = "d"))
fullagginfo_all$log_travel_mm <- log(fullagginfo_all$travel_mm)
fullagginfo_all$log_travel_domains <- log(fullagginfo_all$travel_domains)
fullagginfo_all$log_travel_domains_PV <- log(fullagginfo_all$travel_domains_PV)
fullagginfo_all$log_total_time_travel <- log(fullagginfo_all$total_time_travel)
fullagginfo_all$log_total_time_travel2 <- log(fullagginfo_all$total_time_travel2)
fullagginfo_all$log_purchase <- log(fullagginfo_all$purchase)

```

```

fullagginfo_all$d_purchase <- ifelse(fullagginfo_all$d_purchase <
  mean(fullagginfo_all$d_purchase), 0, 1)
fullagginfo_all$log_mm <- log(fullagginfo_all$mm)
fullagginfo_all$log_domains <- log(fullagginfo_all$domains)
fullagginfo_all$log_domains_PV <- log(fullagginfo_all$domains_PV)
fullagginfo_all$log_time <- log(fullagginfo_all$total_time)
fullagginfo_all$log_time2 <- log(fullagginfo_all$total_time2)
fullagginfo_all$log_travel_mm <- ifelse(fullagginfo_all$log_travel_mm ==
  "-Inf", 0, fullagginfo_all$log_travel_mm)
fullagginfo_all$log_travel_domains <- ifelse(fullagginfo_all$log_travel_domains ==
  "-Inf", 0, fullagginfo_all$log_travel_domains)
fullagginfo_all$log_travel_domains_PV <- ifelse(fullagginfo_all$log_travel_domains_PV ==
  "-Inf", 0, fullagginfo_all$log_travel_domains_PV)
fullagginfo_all$log_total_time_travel <- ifelse(fullagginfo_all$log_total_time_travel ==
  "-Inf", 0, fullagginfo_all$log_total_time_travel)
fullagginfo_all$log_total_time_travel2 <- ifelse(fullagginfo_all$log_total_time_travel2 ==
  "-Inf", 0, fullagginfo_all$log_total_time_travel2)
fullagginfo_all$log_purchase <- ifelse(fullagginfo_all$log_purchase ==
  "-Inf", 0, fullagginfo_all$log_purchase)
fullagginfo_all$log_mm <- ifelse(fullagginfo_all$log_mm == "-Inf",
  0, fullagginfo_all$log_mm)
fullagginfo_all$log_domains <- ifelse(fullagginfo_all$log_domains ==
  "-Inf", 0, fullagginfo_all$log_domains)
fullagginfo_all$log_domains_PV <- ifelse(fullagginfo_all$log_domains_PV ==
  "-Inf", 0, fullagginfo_all$log_domains_PV)
fullagginfo_all$log_time <- ifelse(fullagginfo_all$log_time ==
  "-Inf", 0, fullagginfo_all$log_time)
fullagginfo_all$log_time2 <- ifelse(fullagginfo_all$log_time ==
  "-Inf", 0, fullagginfo_all$log_time2)
fullagginfo_all$act_mm <- ifelse(fullagginfo_all$mm < mean(fullagginfo_all$mm),
  0, 1)
fullagginfo_all$act_total_time <- ifelse(fullagginfo_all$total_time <
  mean(fullagginfo_all$total_time), 0, 1)
fullagginfo_all$act_total_time2 <- ifelse(fullagginfo_all$total_time2 <
  mean(fullagginfo_all$total_time2, na.rm = TRUE), 0, 1)
fullagginfo_all$act_domains <- ifelse(fullagginfo_all$domains <
  mean(fullagginfo_all$domains), 0, 1)
fullagginfo_all$act_domains_PV <- ifelse(fullagginfo_all$domains_PV <
  mean(fullagginfo_all$domains_PV), 0, 1)

```

CODE CHUNK #5. SURVEY DATA PROCESSING

```

rawsurveycompletes1$Q7new <- ifelse(rawsurveycompletes1$Q7 >
  3, 1, 0)
rawsurveycompletes1$Q9new <- ifelse(rawsurveycompletes1$Q8 ==
  2, 0, rawsurveycompletes1$Q9)
rawsurveycompletes1$dRA <- ifelse(rawsurveycompletes1$RA == "seeking",
  1, 0)
rawsurveycompletes1$dUA <- ifelse(rawsurveycompletes1$UA == "seeking",
  1, 0)
which(rawsurveycompletes1$dRA == 1) == which(rawsurveycompletes1$RA ==

```

```

"averse")
which(rawsurveycompletes1$dRA == 0) == which(rawsurveycompletes1$RA ==
"seeking")
which(rawsurveycompletes1$dUA == 1) == which(rawsurveycompletes1$UA ==
"averse")
which(rawsurveycompletes1$dUA == 0) == which(rawsurveycompletes1$UA ==
"seeking")
rawsurveycompletes1$RAUA <- ifelse(rawsurveycompletes1$RA ==
"averse" & rawsurveycompletes1$UA == "averse", "RAUA", ifelse(rawsurveycompletes1$RA ==
"averse" & rawsurveycompletes1$UA == "seeking", "RAUS", ifelse(rawsurveycompletes1$RA ==
"seeking" & rawsurveycompletes1$UA == "averse", "RSUA", ifelse(rawsurveycompletes1$RA ==
"seeking" & rawsurveycompletes1$UA == "seeking", "RSUS",
"NA"))))
rawsurveycompletes1$D3new <- ifelse(as.numeric(rawsurveycompletes1$D3) <
5, 0, 1)
rawsurveycompletes1$Q18new <- ifelse(rawsurveycompletes1$Q18.1.value !=
"{null}", rawsurveycompletes1$Q18.1.value, round(mean(as.numeric(rawsurveycompletes1$Q18.1.value),
na.rm = TRUE), digits = 0))
rawsurveycompletes1$Q18new[242] <- 400
rawsurveycompletes1$Q18new[396] <- 600
rawsurveycompletes1$Q18new[402] <- 400
which(is.numeric(as.numeric(rawsurveycompletes1$Q18.1.value)))
summary(as.factor(rawsurveycompletes1$Q18Anew2))
rawsurveycompletes1$Q18Anew2 <- ifelse(as.numeric(rawsurveycompletes1$Q18new) >
mean(as.numeric(rawsurveycompletes1$Q18new)), 1, 0)
##### FACTOR analysis #####
RAUAdata <- rawsurveycompletes1[, 58:64]
RAUAdata <- as.data.frame(lapply(RAUAdata, as.numeric))
RAUAFactors <- factanal(RAUAdata, factors = 2, rotation = "varimax")
print(RAUAFactors, digits = 2, cutoff = 0.4, sort = FALSE)
BIG5data <- fullagginfo_sdata[c(paste0("Q15_", 1:11))]
BIG5data <- as.data.frame(lapply(BIG5data, as.numeric))
BIG5factors <- factanal(BIG5data, factors = 5, rotation = "varimax")
print(BIG5factors, digits = 2, cutoff = 0.4, sort = FALSE)

```

CODE CHUNK #6. MERGE SURVEY DATA AND PROCESSED BEHAVIORAL DATA

```

library(stargazer)
library(lmtest)
##### LOAD SURVEY DATA #####
load(file = "~/trailspot-channel/export/final_files/final_survey_completes.Rdata")
##### DESKTOP #####
fullagginfo_sdata <- merge(fullagginfo, rawsurveycompletes1,
by.x = "ID", by.y = "ID")
fullagginfo_sdata <- subset(fullagginfo_sdata, obs > 1000 & QC ==
0)
fullagginfo_sdata <- fullagginfo_sdata[-which(duplicated(fullagginfo_sdata$ID)),
]
finalRA <- aggregate(fullagginfo_sdata[2], by = list(RA = fullagginfo_sdata$RA),

```

```

    data = fullagginfo_sdata, FUN = mean)
finalUA <- aggregate(fullagginfo_sdata[2], by = list(UA = fullagginfo_sdata$UA),
    data = fullagginfo_sdata, FUN = mean)
finalRAUA <- aggregate(fullagginfo_sdata[2], by = list(RA = fullagginfo_sdata$RA,
    UA = fullagginfo_sdata$UA), data = fullagginfo_sdata, FUN = mean)
for (ivar in 2:11) {
    finalRA[ivar] <- aggregate(fullagginfo_sdata[ivar], by = list(RA = fullagginfo_sdata$RA),
        data = fullagginfo_sdata, FUN = mean, na.rm = TRUE)[2]
    finalUA[ivar] <- aggregate(fullagginfo_sdata[ivar], by = list(UA = fullagginfo_sdata$UA),
        data = fullagginfo_sdata, FUN = mean, na.rm = TRUE)[2]
    finalRAUA[ivar + 1] <- aggregate(fullagginfo_sdata[ivar],
        by = list(RA = fullagginfo_sdata$RA, UA = fullagginfo_sdata$UA),
        data = fullagginfo_sdata, FUN = mean, na.rm = TRUE)[3]
}
mean(fullagginfo_sdata$days_act)
fullagginfo_sdata$l_date[209]
fullagginfo_sdata$f_date[209]
fullagginfo_sdata$days_act[209]
##### MOBILE #####
fullagginfom_sdata <- merge(fullagginfom, rawsurveycompletes1,
    by.x = "ID", by.y = "ID")
fullagginfom_sdata <- subset(fullagginfom_sdata, QC == 0)
fullagginfom_sdata <- fullagginfom_sdata[-which(duplicated(fullagginfom_sdata$ID)),
    ]
finalRA_m <- aggregate(fullagginfom_sdata[2], by = list(RA = fullagginfom_sdata$RA),
    data = fullagginfom_sdata, FUN = mean)
finalUA_m <- aggregate(fullagginfom_sdata[2], by = list(UA = fullagginfom_sdata$UA),
    data = fullagginfom_sdata, FUN = mean)
finalRAUA_m <- aggregate(fullagginfom_sdata[2], by = list(RA = fullagginfom_sdata$RA,
    UA = fullagginfom_sdata$UA), data = fullagginfom_sdata, FUN = mean)
for (ivar in 2:17) {
    finalRA_m[ivar] <- aggregate(fullagginfom_sdata[ivar], by = list(RA = fullagginfom_sdata$RA),
        data = fullagginfom_sdata, FUN = mean, na.rm = TRUE)[2]
    finalUA_m[ivar] <- aggregate(fullagginfom_sdata[ivar], by = list(UA = fullagginfom_sdata$UA),
        data = fullagginfom_sdata, FUN = mean, na.rm = TRUE)[2]
    finalRAUA_m[ivar + 1] <- aggregate(fullagginfom_sdata[ivar],
        by = list(RA = fullagginfom_sdata$RA, UA = fullagginfom_sdata$UA),
        data = fullagginfom_sdata, FUN = mean, na.rm = TRUE)[3]
}
ftable(fullagginfom_sdata$RA, fullagginfom_sdata$UA)
##### DESKTOP AND MOBILE #####
fullagginfo_all_sdata <- merge(fullagginfo_all, rawsurveycompletes1,
    by.x = "ID", by.y = "ID")
fullagginfo_all_sdata <- subset(fullagginfo_all_sdata, QC ==
    0 & device != 2)
fullagginfo_all_sdata <- fullagginfo_all_sdata[-which(duplicated(fullagginfo_all_sdata$ID)),
    ]
finalRA_all <- aggregate(fullagginfo_all_sdata[2], by = list(RA = fullagginfo_all_sdata$RA),
    data = fullagginfo_all_sdata, FUN = mean)
finalUA_all <- aggregate(fullagginfo_all_sdata[2], by = list(UA = fullagginfo_all_sdata$UA),
    data = fullagginfo_all_sdata, FUN = mean)
finalRAUA_all <- aggregate(fullagginfo_all_sdata[2], by = list(RA = fullagginfo_all_sdata$RA,
    UA = fullagginfo_all_sdata$UA), data = fullagginfo_all_sdata,
    FUN = mean)
for (ivar in 2:13) {
    finalRA_all[ivar] <- aggregate(fullagginfo_all_sdata[ivar],
        by = list(RA = fullagginfo_all_sdata$RA), data = fullagginfo_all_sdata,

```

```

    FUN = mean, na.rm = TRUE)[2]
finalUA_all[ivar] <- aggregate(fullagginfo_all_sdata[ivar],
    by = list(UA = fullagginfo_all_sdata$UA), data = fullagginfo_all_sdata,
    FUN = mean, na.rm = TRUE)[2]
finalRAUA_all[ivar + 1] <- aggregate(fullagginfo_all_sdata[ivar],
    by = list(RA = fullagginfo_all_sdata$RA, UA = fullagginfo_all_sdata$UA),
    data = fullagginfo_all_sdata, FUN = mean, na.rm = TRUE)[3]
}

```

CODE CHUNK #8. FINAL ANALYSIS AND OLS TESTS

```

library(stargazer)
library(sandwich)
library(lmtest)

#####
#####          RESTRICTED MODEL          #####
#####

#####
##### INDEPENDENT VARIABLE LIST          #####
#####

listvars <- c( "I(RA == 'seeking')", "I(UA == 'seeking')", "I(RA == 'seeking')*I(UA == 'seeking')", "days_act" )

core_ind_var_set <- paste0( listvars , collapse = " + " )

#RESTRICTED
ind_var_set_r <- paste0( c(core_ind_var_set), collapse = " + " )

#####
#####          DESKTOP          #####
#####

ols <- list()
ols[["restricted_desktop_1"]] <- lm( paste( "log_travel_mm" , "~", ind_var_set_r, "+", "log_mm" ) )
ols[["restricted_desktop_2"]] <- lm( paste( "log_travel_domains" , "~", ind_var_set_r, "+", "log_dom" ) )
ols[["restricted_desktop_3"]] <- lm( paste( "log_travel_domains_PV" , "~", ind_var_set_r, "+", "log_dom_PV" ) )
ols[["restricted_desktop_4"]] <- lm( paste( "log_total_time_travel" , "~", ind_var_set_r, "+", "log_time" ) )
ols[["restricted_desktop_5"]] <- lm( paste( "log_total_time_travel2" , "~", ind_var_set_r, "+", "log_time2" ) )
ols[["restricted_desktop_6"]] <- lm( paste( "travel_mm" , "~", ind_var_set_r, "+", "mm" ) )
ols[["restricted_desktop_7"]] <- lm( paste( "travel_domains_PV" , "~", ind_var_set_r, "+", "domains_PV" ) )
ols[["restricted_desktop_8"]] <- lm( paste( "travel_domains" , "~", ind_var_set_r, "+", "domains" ) )
ols[["restricted_desktop_9"]] <- lm( paste( "total_time_travel" , "~", ind_var_set_r, "+", "total_time" ) )
ols[["restricted_desktop_10"]] <- lm( paste( "total_time_travel2" , "~", ind_var_set_r, "+", "total_time2" ) )
ols[["restricted_desktop_11"]] <- lm( paste( "log_travel_mm" , "~", ind_var_set_r, "+", "act_mm" ) )
ols[["restricted_desktop_12"]] <- lm( paste( "log_travel_domains" , "~", ind_var_set_r, "+", "act_dom" ) )
ols[["restricted_desktop_13"]] <- lm( paste( "log_travel_domains_PV" , "~", ind_var_set_r, "+", "act_dom_PV" ) )

```

```

ols[["restricted_desktop_14"]] <- lm( paste( "log_total_time_travel" , "~", ind_var_set_r, "+", "act_tot
ols[["restricted_desktop_15"]] <- lm( paste( "log_total_time_travel2" , "~", ind_var_set_r, "+", "act_tot
ols[["restricted_desktop_16"]] <- lm( paste( "travel_mm" , "~", ind_var_set_r, "+", "act_mm
ols[["restricted_desktop_17"]] <- lm( paste( "travel_domains_PV" , "~", ind_var_set_r, "+", "act_dor
ols[["restricted_desktop_18"]] <- lm( paste( "travel_domains" , "~", ind_var_set_r, "+", "act_dor
ols[["restricted_desktop_19"]] <- lm( paste( "total_time_travel" , "~", ind_var_set_r, "+", "act_tot
ols[["restricted_desktop_20"]] <- lm( paste( "total_time_travel2" , "~", ind_var_set_r, "+", "act_tot

table_tests <- list()
table_tests[["restricted_desktop"]] <- as.data.frame( paste0( "REST:", c("log(MM)", "log(TD)", "log(TDPV)",
colnames(table_tests[["restricted_desktop"]]) <- 'data_name'

for (ivar in 1:nrow(table_tests[["restricted_desktop"]])) {
  table_tests[["restricted_desktop"]][ivar] <- iver
  table_tests[["restricted_desktop"]][ivar]$Residual_Standard_Err[ivar] <- summary( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$F.Stat[ivar] <- summary( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$NumDF[ivar] <- summary( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$FDenDF[ivar] <- summary( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$R.Sq[ivar] <- summary( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$Adj.R.Sq[ivar] <- summary( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$Shapiro_Wilk_Stat[ivar] <- shapiro.test( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$Shapiro_Wilk_P.val[ivar] <- shapiro.test( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$Reset_Stat[ivar] <- resettest( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$Reset_P.val[ivar] <- resettest( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$BP_Stat[ivar] <- bptest( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$BP_P.val[ivar] <- bptest( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$Wald.F.Stat[ivar] <- waldtest( ols[[paste0("restricted
  table_tests[["restricted_desktop"]][ivar]$Wald.P.Stat[ivar] <- waldtest( ols[[paste0("restricted

}

stargazer( ols[["restricted_desktop_1" ]],
  ols[["restricted_desktop_2" ]],
  ols[["restricted_desktop_3" ]],
  ols[["restricted_desktop_4" ]],
  ols[["restricted_desktop_5" ]],
  ols[["restricted_desktop_6" ]],
  ols[["restricted_desktop_7" ]],
  ols[["restricted_desktop_8" ]],
  ols[["restricted_desktop_9" ]],
  ols[["restricted_desktop_10"]],
  ols[["restricted_desktop_11"]],
  ols[["restricted_desktop_12"]],
  ols[["restricted_desktop_13"]],
  ols[["restricted_desktop_14"]],
  ols[["restricted_desktop_15"]],
  ols[["restricted_desktop_16"]],
  ols[["restricted_desktop_17"]],
  ols[["restricted_desktop_18"]],
  ols[["restricted_desktop_19"]],
  ols[["restricted_desktop_20"]],
  title="Results Desktop Restricted",
  align=TRUE,

```



```

    type = "text",
    #covariate.labels = c("Risk Seeking", "Uncertainty Seeking", "days in panel", "RS * US"),
    dep.var.labels = c("log(MM)", "log(TD)", "log(TDPV)", "log(TT)", "log(TL)", "MM", "TD", "TDPV", "TT", "TL"),
    report=('vc*p'),
    no.space = TRUE
  )
)

stargazer(table_tests[["restricted_desktop"]], type = "text", summary=FALSE)

#####
#####          MOBILE          #####
#####

#FULL
#REMOVE Q11_5 bc all the same values

surv_ind_var_set_m <- paste0( fulllist[-which(fulllist=="Q11_5")] , collapse = " + " )
ind_var_set_m <- paste0( c(core_ind_var_set, surv_ind_var_set_m), collapse = " + " )

ols[["restricted_mobile_1"]] <- lm( paste( "log_travel_mm" , "~", ind_var_set_r, "+", "log_mm"
ols[["restricted_mobile_2"]] <- lm( paste( "log_travel_domains" , "~", ind_var_set_r, "+", "log_doma
ols[["restricted_mobile_3"]] <- lm( paste( "log_travel_domains_PV" , "~", ind_var_set_r, "+", "log_doma
ols[["restricted_mobile_4"]] <- lm( paste( "log_total_time_travel" , "~", ind_var_set_r, "+", "log_time
ols[["restricted_mobile_5"]] <- lm( paste( "log_total_time_travel2" , "~", ind_var_set_r, "+", "log_time
ols[["restricted_mobile_6"]] <- lm( paste( "travel_mm" , "~", ind_var_set_r, "+", "mm"
ols[["restricted_mobile_7"]] <- lm( paste( "travel_domains_PV" , "~", ind_var_set_r, "+", "domains
ols[["restricted_mobile_8"]] <- lm( paste( "travel_domains" , "~", ind_var_set_r, "+", "domains
ols[["restricted_mobile_9"]] <- lm( paste( "total_time_travel" , "~", ind_var_set_r, "+", "total_t
ols[["restricted_mobile_10"]] <- lm( paste( "total_time_travel2" , "~", ind_var_set_r, "+", "total_t
ols[["restricted_mobile_11"]] <- lm( paste( "log_travel_mm" , "~", ind_var_set_r, "+", "act_mm"
ols[["restricted_mobile_12"]] <- lm( paste( "log_travel_domains" , "~", ind_var_set_r, "+", "act_doma
ols[["restricted_mobile_13"]] <- lm( paste( "log_travel_domains_PV" , "~", ind_var_set_r, "+", "act_doma
ols[["restricted_mobile_14"]] <- lm( paste( "log_total_time_travel" , "~", ind_var_set_r, "+", "act_tota
ols[["restricted_mobile_15"]] <- lm( paste( "log_total_time_travel2" , "~", ind_var_set_r, "+", "act_tota
ols[["restricted_mobile_16"]] <- lm( paste( "travel_mm" , "~", ind_var_set_r, "+", "act_mm"
ols[["restricted_mobile_17"]] <- lm( paste( "travel_domains_PV" , "~", ind_var_set_r, "+", "act_doma
ols[["restricted_mobile_18"]] <- lm( paste( "travel_domains" , "~", ind_var_set_r, "+", "act_doma
ols[["restricted_mobile_19"]] <- lm( paste( "total_time_travel" , "~", ind_var_set_r, "+", "act_tota
ols[["restricted_mobile_20"]] <- lm( paste( "total_time_travel2" , "~", ind_var_set_r, "+", "act_tota

table_tests[["restricted_mobile"]] <- as.data.frame( paste0( "REST:", c("log(MM)", "log(TD)", "log(TDPV)", "log(TT)", "log(TL)", "MM", "TD", "TDPV", "TT", "TL"),
colnames(table_tests[["restricted_mobile"]]) <- 'data_name'

for (ivar in 1:nrow(table_tests[["restricted_mobile"]])) {
  table_tests[["restricted_mobile"]]$name [ivar] <- ivar
  table_tests[["restricted_mobile"]]$Residual_Standard_Err[ivar] <- summary( ols[[paste0("restricted_r
  table_tests[["restricted_mobile"]]$F.Stat [ivar] <- summary( ols[[paste0("restricted_r
  table_tests[["restricted_mobile"]]$NumDF [ivar] <- summary( ols[[paste0("restricted_r
  table_tests[["restricted_mobile"]]$FDenDF [ivar] <- summary( ols[[paste0("restricted_r

```



```

ols[["restricted_combined_5" ]] <- lm( paste( "log_total_time_travel2" , "~", ind_var_set_r, "+", "log_t
ols[["restricted_combined_6" ]] <- lm( paste( "travel_mm" , "~", ind_var_set_r, "+", "mm"
ols[["restricted_combined_7" ]] <- lm( paste( "travel_domains_PV" , "~", ind_var_set_r, "+", "domain
ols[["restricted_combined_8" ]] <- lm( paste( "travel_domains" , "~", ind_var_set_r, "+", "domain
ols[["restricted_combined_9" ]] <- lm( paste( "total_time_travel" , "~", ind_var_set_r, "+", "total
ols[["restricted_combined_10" ]] <- lm( paste( "total_time_travel2" , "~", ind_var_set_r, "+", "total
ols[["restricted_combined_11" ]] <- lm( paste( "log_travel_mm" , "~", ind_var_set_r, "+", "act_mm
ols[["restricted_combined_12" ]] <- lm( paste( "log_travel_domains" , "~", ind_var_set_r, "+", "act_d
ols[["restricted_combined_13" ]] <- lm( paste( "log_travel_domains_PV" , "~", ind_var_set_r, "+", "act_d
ols[["restricted_combined_14" ]] <- lm( paste( "log_total_time_travel" , "~", ind_var_set_r, "+", "act_t
ols[["restricted_combined_15" ]] <- lm( paste( "log_total_time_travel2" , "~", ind_var_set_r, "+", "act_t
ols[["restricted_combined_16" ]] <- lm( paste( "travel_mm" , "~", ind_var_set_r, "+", "act_mm
ols[["restricted_combined_17" ]] <- lm( paste( "travel_domains_PV" , "~", ind_var_set_r, "+", "act_d
ols[["restricted_combined_18" ]] <- lm( paste( "travel_domains" , "~", ind_var_set_r, "+", "act_d
ols[["restricted_combined_19" ]] <- lm( paste( "total_time_travel" , "~", ind_var_set_r, "+", "act_t
ols[["restricted_combined_20" ]] <- lm( paste( "total_time_travel2" , "~", ind_var_set_r, "+", "act_t

```

```

table_tests[["restricted_combined"]] <- as.data.frame( paste0( "REST:", c("log(MM)", "log(TD)", "log(TDPV)"
colnames(table_tests[["restricted_combined"]]) <- 'data_name'

```

```

for (ivar in 1:nrow(table_tests[["restricted_combined"]])) {

```

table_tests[["restricted_combined"]]\$name	[ivar] <- ivar	
table_tests[["restricted_combined"]]\$Residual_Standard_Err	[ivar] <- summary(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$F.Stat	[ivar] <- summary(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$NumDF	[ivar] <- summary(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$FDenDF	[ivar] <- summary(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$R.Sq	[ivar] <- summary(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$Adj.R.Sq	[ivar] <- summary(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$Shapiro_Wilk_Stat	[ivar] <- shapiro.test(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$Shapiro_Wilk_P.val	[ivar] <- shapiro.test(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$Reset_Stat	[ivar] <- resettest(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$Reset_P.val	[ivar] <- resettest(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$BP_Stat	[ivar] <- bptest(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$BP_P.val	[ivar] <- bptest(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$Wald.F.Stat	[ivar] <- waldtest(ols[[paste0("restricted_
table_tests[["restricted_combined"]]\$Wald.P.Stat	[ivar] <- waldtest(ols[[paste0("restricted_

```

}

```

```

stargazer( ols[["restricted_combined_1" ]],
            ols[["restricted_combined_2" ]],
            ols[["restricted_combined_3" ]],
            ols[["restricted_combined_4" ]],
            ols[["restricted_combined_5" ]],
            ols[["restricted_combined_6" ]],
            ols[["restricted_combined_7" ]],
            ols[["restricted_combined_8" ]],
            ols[["restricted_combined_9" ]],
            ols[["restricted_combined_10"]],
            ols[["restricted_combined_11"]],
            ols[["restricted_combined_12"]],

```

```

      ols[["restricted_combined_13"]],
      ols[["restricted_combined_14"]],
      ols[["restricted_combined_15"]],
      ols[["restricted_combined_16"]],
      ols[["restricted_combined_17"]],
      ols[["restricted_combined_18"]],
      ols[["restricted_combined_19"]],
      ols[["restricted_combined_20"]],
      title="Results Combined Restricted",
      align=TRUE,
      type = "text",
      #covariate.labels = c("Risk Seeking", "Uncertainty Seeking", "days in panel", "RS * US"),
      dep.var.labels = c("log(MM)", "log(TD)", "log(TDPV)", "log(TT)", "log(TL)", "MM", "TD", "TDPV", "TT", "TL"),
      report=('vc*p'),
      no.space = TRUE
    )

stargazer(table_tests[["restricted_combined"]], type = "text", summary=FALSE)

#####
##### Summary ALL RESTRICTED MODELS #####
#####

stargazer(ols[["restricted_desktop_11"]],
  ols[["restricted_desktop_12"]],
  ols[["restricted_desktop_13"]],
  ols[["restricted_desktop_14"]],
  ols[["restricted_desktop_15"]],
  ols[["restricted_mobile_11"]],
  ols[["restricted_mobile_12"]],
  ols[["restricted_mobile_13"]],
  ols[["restricted_mobile_14"]],
  ols[["restricted_mobile_15"]],
  ols[["restricted_combined_11"]],
  ols[["restricted_combined_12"]],
  ols[["restricted_combined_13"]],
  ols[["restricted_combined_14"]],
  ols[["restricted_combined_15"]],
  title="Results Restricted Desktop, Restricted Mobile, Restricted Desktop AND Mobile",
  align=TRUE,
  type = "text",
  #covariate.labels = c("Risk Seeking", "Uncertainty Seeking", "days in panel", "RS * US"),
  dep.var.labels = c( paste("DESK", c("log(MM)", "log(TD)", "log(TDPV)", "log(TT)", "log(TL)")), paste("MOB", c("log(MM)", "log(TD)", "log(TDPV)", "log(TT)", "log(TL)")), paste("COMB", c("log(MM)", "log(TD)", "log(TDPV)", "log(TT)", "log(TL)")) ),
  report=('vc*p'),
  omit.stat = "all"
)

table_tests[["final_restricted"]] <- rbind(table_tests[["restricted_desktop"]],
                                           table_tests[["restricted_mobile"]],
                                           table_tests[["restricted_combined"]])

table_tests[["final_restricted"]]$data_name <- c( paste0( "DESKTOP:", table_tests[["restricted_desktop"]]$data_name ),
  paste0( "MOBILE:", table_tests[["restricted_mobile"]]$data_name ),
  paste0( "COMBINED:", table_tests[["restricted_combined"]]$data_name ) )

```

```

#table_tests[["final_restricted"]] <- table_tests[["final_restricted"]][order(table_tests[["final_restricted"]][1:12], type = "text", summary=FALSE, rownames = FALSE)
#stargazer(table_tests[["final_restricted"]][1:12], type = "text", summary=FALSE, rownames = FALSE)
stargazer(table_tests[["final_restricted"]][c(11:15,31:35,51:55),], type = "text", summary=FALSE, rownames = FALSE)
#apply(table_tests[["final_restricted"]][1:12],2, FUN = function(x) paste0(x , sep = ","))

```

```

#####
#####          FULL MODEL          #####
#####

```

```

#####
##### INDEPENDENT VARIABLE LIST #####
#####

```

```

fulllist <- c(
  #SCREEN
  #REPORTED NUMBER OF TRIPS
  'factor( S1 )',
  #REPORTED PRIMARY BOOKING
  'factor( S2 )',
  #ROLE IN DECISION MAKING
  'factor( S3 )',
  # BUSINESS and LEASURE vs. LEASURE (NOT ENOUGH)
  #'factor( S4 )',
  #DEMO,
  #AGE
  #'as.numeric(D1)',
  #AGE SQ
  #'D2' ,
  'poly( as.numeric(D1),degree = 2 )',
  #INCOME
  #'factor(D3)' ,
  #INCOME DUMMY
  #'factor(D3new)' ,
  #TRIPCHAR,
  'factor(Q1)',
  'factor(Q1)*factor(Q5_5)',
  #'as.numeric( Q2)',
  'as.numeric(Q4)',
  #SOURCE,
  'factor(Q5_1)',
  'factor(Q5_2)',
  'factor(Q5_3)',
  'factor(Q5_4)',
  'factor(Q5_5)',
  'factor(Q5_6)',
  #'factor(Q5Anew)',
  #BOUGHT WHAT,
  'factor( Q6.1 )',
  'factor( Q6.2 )',
  'factor( Q6.3 )',
  #AWAY FROM HOME ,
  #'as.numeric(Q7)' ,
  'factor( Q7new )',
  #VISITED BEFORE,
  'factor( Q8 )',

```

```

#as.numeric(Q9),
#'poly( as.numeric(Q9new),degree = 2 )',
'as.numeric(Q10)',
#KIDS, V relatives, Stayed at hotel, airbnb, grouptrip,
'factor( Q11_1 )' ,
'factor( Q11_2 )' ,
'factor( Q11_3 )' ,
'factor( Q11_4 )' ,
'factor( Q11_5 )' ,
# as.numeric(Q18.1.value),
#poly( as.numeric(Q18new),2),
#as.numeric( Q18new),
#'as.numeric( Q18Anew2 )',
'factor( 0 )',
'factor( C )',
'factor( E )',
'factor( A )',
'factor( N )'
)

listvars <- c( "I(RA == 'seeking')", "I(UA == 'seeking')", "I(RA == 'seeking')*I(UA == 'seeking')", "days_act"

core_ind_var_set <- paste0( listvars , collapse = " + " )
surv_ind_var_set <- paste0( fulllist , collapse = " + " )

#RESTRICTED
ind_var_set_r <- paste0( c(core_ind_var_set), collapse = " + " )
#FULL
ind_var_set_f <- paste0( c(core_ind_var_set, surv_ind_var_set), collapse = " + " )

stepwise_select <- list()

#####
#####      DESKTOP      #####
#####
stepwise_select[["final_desktop_1"]] <-
stepwise_select[["final_desktop_2"]] <-
stepwise_select[["final_desktop_3"]] <-
stepwise_select[["final_desktop_4"]] <-
stepwise_select[["final_desktop_5"]] <-

#####
#####      MOBILE      #####
#####
step( lm(
  scope = list(lower = as.formula(paste( "log_t
  upper = as.formula(paste( "log_t
step( lm(
  scope = list(lower = as.formula(paste( "log_t
  upper = as.formula(paste( "log_t
step( lm(
  scope = list(lower = as.formula(paste( "log_t
  upper = as.formula(paste( "log_t
step( lm(
  scope = list(lower = as.formula(paste( "log_t
  upper = as.formula(paste( "log_t
step( lm(
  scope = list(lower = as.formula(paste( "log_t
  upper = as.formula(paste( "log_t

```

```

surv_ind_var_set_m <- paste0( fulllist[-which(fulllist=="factor( Q11_5 )")] , collapse = " + " )
ind_var_set_m <- paste0( c(core_ind_var_set, surv_ind_var_set_m), collapse = " + " )

stepwise_select[["final_mobile_1"]] <-
stepwise_select[["final_mobile_2"]] <-
stepwise_select[["final_mobile_3"]] <-
stepwise_select[["final_mobile_4"]] <-
stepwise_select[["final_mobile_5"]] <-

#####
#####          COMBINED          #####
#####

stepwise_select[["final_combined_1"]] <-
stepwise_select[["final_combined_2"]] <-
stepwise_select[["final_combined_3"]] <-
stepwise_select[["final_combined_4"]] <-
stepwise_select[["final_combined_5"]] <-

stargazer(lm(stepwise_select[["final_desktop_1"]]) ,
           lm(stepwise_select[["final_desktop_2"]]) ,
           lm(stepwise_select[["final_desktop_3"]]) ,
           lm(stepwise_select[["final_desktop_4"]]) ,
           lm(stepwise_select[["final_desktop_5"]]) ,
           lm(stepwise_select[["final_mobile_1"]]) ,
           lm(stepwise_select[["final_mobile_2"]]) ,
           lm(stepwise_select[["final_mobile_3"]]) ,
           lm(stepwise_select[["final_mobile_4"]]) ,
           lm(stepwise_select[["final_mobile_5"]]) ,
           lm(stepwise_select[["final_combined_1"]]) ,
           lm(stepwise_select[["final_combined_2"]]) ,
           lm(stepwise_select[["final_combined_3"]]) ,
           lm(stepwise_select[["final_combined_4"]]) ,
           lm(stepwise_select[["final_combined_5"]]) ,
           column.labels = c("Desktop", "Mobile", "Combined"),
           column.separate = c(5, 5, 5),

```

```

#dep.var.labels = c( paste("D", c("log(MM)", "log(TD)", "log(TDPV)", "log(TT)", "log(TL)")),paste
dep.var.labels = c( paste("D. Model#", 1:5),paste("M. Model#", 1:5),paste("C. Model#", 1:5) ),
multicolumn = FALSE,
omit.stat = "all",
covariate.labels = c("Risk Seeking","Uncertainty Seeking","Risk(x)Uncertainty: Seeking",
                     "Days active","D Purchase","D Micromoments","D Domains","D PageViews","D Time
                     "S1. x1 travel related purchase","S1. x2 travel related purchase","S1. x3+tr
                     "Q1. Country: Asia (Base Europe)","Q1. Country: North America (Base Europe)"
                     "Q4. Planning horizon (Weeks)",
                     "Q5.2. Used advice of friends or relatives","Q5.3. Used tourist information o
                     "Q6.1. Purchased online: Transport","Q6.2. Purchased online: Accommodation",
                     "Q7. Trip longer than 3 nights",
                     "Q8. Visited before", "Q10. Numer of people",
                     "Q11.1. Children (N)","Q11.2. Visited Friends/Relatives (N)","Q11.3. Stayed a
                     "Openness to experience (low)", "Extraversion (low)", "Neuroticism (low)", "A
                     "Asia x Travel Agent","North America x Travel Agent","South America x Travel

),
order = c(1,2,42,3,4,5,6,12,7,8,
          9:11,13:21,23,
          27,24,28,22,
          25,29,26,
          30:32,
          33,37,34,36,35
),
type="text")

final_results <- list()

for(ivar1 in c("final_desktop_","final_mobile_","final_combined_")){
  for(ivar in 1:5){

    print(paste0(ivar1,ivar))
    final_results[[ paste0(ivar1,ivar) ]]$dataname <- paste0(ivar1,ivar)

    final_results[[ paste0(ivar1,ivar) ]]$Residual_Standard_Err[ paste0(ivar1,ivar) ] <- summary(
    final_results[[ paste0(ivar1,ivar) ]]$F.Stat [ paste0(ivar1,ivar) ] <- summary(
    final_results[[ paste0(ivar1,ivar) ]]$NumDF [ paste0(ivar1,ivar) ] <- summary(
    final_results[[ paste0(ivar1,ivar) ]]$FDenDF [ paste0(ivar1,ivar) ] <- summary(
    final_results[[ paste0(ivar1,ivar) ]]$R.Sq [ paste0(ivar1,ivar) ] <- summary(
    final_results[[ paste0(ivar1,ivar) ]]$Adj.R.Sq [ paste0(ivar1,ivar) ] <- summary(
    final_results[[ paste0(ivar1,ivar) ]]$Shapiro_Wilk_Stat [ paste0(ivar1,ivar) ] <- shapiro.test(
    final_results[[ paste0(ivar1,ivar) ]]$Shapiro_Wilk_P.val [ paste0(ivar1,ivar) ] <- shapiro.test(
    final_results[[ paste0(ivar1,ivar) ]]$Reset_Stat [ paste0(ivar1,ivar) ] <- resettest(
    final_results[[ paste0(ivar1,ivar) ]]$Reset_P.val [ paste0(ivar1,ivar) ] <- resettest(
    final_results[[ paste0(ivar1,ivar) ]]$BP_Stat [ paste0(ivar1,ivar) ] <- bptest(
    final_results[[ paste0(ivar1,ivar) ]]$BP_P.val [ paste0(ivar1,ivar) ] <- bptest(
    try(final_results[[ paste0(ivar1,ivar) ]]$Wald.F.Stat [ paste0(ivar1,ivar) ] <- waldtest(
    try(final_results[[ paste0(ivar1,ivar) ]]$Wald.P.Stat [ paste0(ivar1,ivar) ] <- waldtest(

```



```
}  
}  
  
final_results[["all"]] <- rbind(final_results[["final_desktop_1"]],  
                                final_results[["final_desktop_2"]],  
                                final_results[["final_desktop_3"]],  
                                final_results[["final_desktop_4"]],  
                                final_results[["final_desktop_5"]],  
                                final_results[["final_mobile_1"]],  
                                final_results[["final_mobile_2"]],  
                                final_results[["final_mobile_3"]],  
                                final_results[["final_mobile_4"]],  
                                final_results[["final_mobile_5"]],  
                                final_results[["final_combined_1"]],  
                                final_results[["final_combined_2"]],  
                                final_results[["final_combined_3"]],  
                                final_results[["final_combined_4"]],  
                                final_results[["final_combined_5"]])  
  
stargazer(final_results[["all"]], type="text" )
```
