Bunching Analysis

Karen Nershi

2025-04-08

Bunching anlaysis

The following code generates output for Tables 1-4 in the main body of the paper and Table 9 in the Supplementary Material. This output matches the code included in the file "Bunching Analysis.R."

```
################################################################################
################ PART 1: LOAD PACKAGES AND FUNCTIONS FOR ANLAYSIS ###############
################################################################################

################################################################################
### A. PACKAGES

# Packages used in analysis
pkg = c("tidyverse","knitr","kableExtra")

# Checks if installed; if not, install
if (length(setdiff(pkg, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(pkg, rownames(installed.packages())))
}

# Load packages
suppressMessages(suppressWarnings(lapply(pkg, library, character.only = TRUE)))
```

```
## [[1]]
##  [1] "lubridate" "forcats"   "stringr"   "dplyr"     "purrr"     "readr"
##  [7] "tidyr"     "tibble"    "ggplot2"   "tidyverse" "stats"     "graphics"
## [13] "grDevices" "utils"     "datasets"  "methods"   "base"
##
## [[2]]
##  [1] "knitr"     "lubridate" "forcats"   "stringr"   "dplyr"     "purrr"
##  [7] "readr"     "tidyr"     "tibble"    "ggplot2"   "tidyverse" "stats"
## [13] "graphics"  "grDevices" "utils"     "datasets"  "methods"   "base"
##
## [[3]]
##  [1] "kableExtra" "knitr"      "lubridate"  "forcats"    "stringr"
##  [6] "dplyr"      "purrr"      "readr"      "tidyr"      "tibble"
## [11] "ggplot2"    "tidyverse"  "stats"      "graphics"   "grDevices"
## [16] "utils"      "datasets"   "methods"    "base"
```

```
rm(list=ls())


################################################################################
```

```r
### B. FUNCTIONS

# Create function for estimating bunching

estimate_beta = function(thedata, z_vector, binv, zstar, bins_excl_l, bins_excl_r, rates) {
  z_vector = thedata$base

  # set parameters
  binwidth = 10 # 10 units for each bin
  bins_l = 25
  bins_r = 25
  poly = 3 # degree of polynomial
  n_boot = 1000 # number of iterations for caclulating bootstrapped standard errors
  rn = NA
  correct_above_zu = FALSE

  zmax <- zstar + (binwidth * bins_r)
  zmin <- zstar - (binwidth * bins_l)
  bins <- seq(zmin, zmax, by = binwidth)

  ##Cut the bins
  thebin <- cut(z_vector, bins, right = FALSE, labels = FALSE)
  thebin <- zmin + binwidth * (thebin - 1)
  thedata <- data.frame(z = z_vector, bin = thebin)
  thedata <- thedata %>% dplyr::group_by(bin) %>%
    dplyr::summarise(freq = n(), z = mean(z, na.rm = TRUE)) %>%
    dplyr::filter(!is.na(bin))
  thedata$freq_orig <- thedata$freq
  thedata <- as.data.frame(thedata)

  data_binned = thedata

  data_binned$z_rel = (data_binned$bin - zstar)/binwidth
  data_binned$zstar <- ifelse(data_binned$bin == zstar, 1,
                              0)
  extra_fe_vector <- ""

  polynomial_vector <- c()
  for (i in seq(poly)) {
    poly_varname <- paste0("poly_", i)
    data_binned[[poly_varname]] <- data_binned$z^i
    polynomial_vector <- c(polynomial_vector, poly_varname)
  }
  bins_excluded_all <- c()
  if (bins_excl_l != 0) {
    bins_excl_l_vector <- c()
    for (i in seq(bins_excl_l)) {
      bins_excl_l_varname <- paste0("bin_excl_l_", i)
      data_binned[[bins_excl_l_varname]] <- ifelse(data_binned$z_rel ==
                                                     -i, 1, 0)
      bins_excl_l_vector <- c(bins_excl_l_vector, bins_excl_l_varname)
    }
    bins_excluded_all <- c(bins_excluded_all, bins_excl_l_vector)
```

```
}
if (bins_excl_r != 0) {
  bins_excl_r_vector <- c()
  for (i in seq(bins_excl_r)) {
    bin_excl_r_varname <- paste0("bin_excl_r_", i)
    data_binned[[bin_excl_r_varname]] <- ifelse(data_binned$z_rel ==
                                                  i, 1, 0)
    bins_excl_r_vector <- c(bins_excl_r_vector, bin_excl_r_varname)
  }
  bins_excluded_all <- c(bins_excluded_all, bins_excl_r_vector)
}
if (length(bins_excluded_all) > 0) {
  data_binned$bunch_region <- rowSums(data_binned[, c("zstar",
                                                      bins_excluded_all)])
}
data_binned$bin_above_excluded <- ifelse(data_binned$bin >
                                           zstar, 1, 0)
rn_vector <- ""
rhs_vars <- c("zstar", extra_fe_vector, polynomial_vector,
              rn_vector, bins_excluded_all)
rhs_vars <- setdiff(rhs_vars, "")
model_formula <- stats::as.formula(paste0("freq", " ~ ",
                                          paste(rhs_vars, collapse = " +")))

data_forreg <- list(data_binned = data_binned, model_formula = model_formula)

#######
thedata = data_forreg$data_binned
themodelformula = data_forreg$model_formula
notch = FALSE
zD_bin = NA

# Define model outcomes
model_fit <- stats::lm(themodelformula, thedata)
coefficients <- summary(model_fit)$coefficients
residuals <- stats::residuals(model_fit)
thedata$cf <- stats::predict(model_fit, thedata)
thedata$cf <- thedata$cf - (thedata$zstar * coefficients["zstar",
                                                         "Estimate"])
bins_excluded_in_reg <- rownames(coefficients)[grepl("bin_excl",
                                                     rownames(coefficients))]
for (i in bins_excluded_in_reg) {
  thedata$cf <- thedata$cf - (thedata[[i]] * coefficients[i,
                                                          "Estimate"])
}
bins_zstar_zu <- sum(grepl("bin_excl_r", rownames(coefficients)))
bins_zl_zstar <- sum(grepl("bin_excl_l", rownames(coefficients))) +
  1
zstarvalue <- thedata$bin[thedata$zstar == 1]
zstarvalue = zstarvalue[1]
binwidthvalue <- thedata$bin[2] - thedata$bin[1]

#Excluded bins below threshold
```

```
thedata$zl_zstar <- ifelse(((thedata$bin >=
                              (zstarvalue -
                                 (binwidthvalue * (bins_zl_zstar - 1)))) &
                             (thedata$bin <= zstarvalue)),
                           1, 0)
thedata$zstar_zu <- ifelse((thedata$bin <=
                              zstarvalue + (binwidthvalue *
                                              bins_zstar_zu)) &
                             (thedata$bin > zstarvalue), 1, 0)
thedata$bunch_region <- ifelse(thedata$zl_zstar == 1, "zl_zstar",
                               ifelse(thedata$zstar_zu == 1,
                                      "zstar_zu", "outside_bunching"))
bunching_region_count <- thedata %>% dplyr::group_by(bunch_region) %>%
  dplyr::summarize(actual = sum(freq_orig), cf = sum(cf),
                   excess = actual - cf)

B_zl_zstar <- as.numeric(subset(bunching_region_count, bunch_region ==
                                  "zl_zstar", select = 'excess'))
B_zstar_zu <- as.numeric(subset(bunching_region_count, bunch_region ==
                                  "zstar_zu", select = "excess"))
if (is.na(B_zstar_zu)) {
  B_zstar_zu <- 0
}
bunching_excess <- B_zl_zstar + B_zstar_zu
cf_bunching <- sum(subset(bunching_region_count, bunch_region !=
                            "outside_bunching", select = "cf"))
bins_bunching <- sum(thedata$bunch_region %in% c("zl_zstar",
                                                 "zstar_zu"))
c0 <- cf_bunching/bins_bunching
b_estimate <- as.numeric(sprintf("%.3f", bunching_excess/c0))

##Calculate standard errors
data_for_boot <- thedata
model <- themodelformula

boot_results <- sapply(seq(1:n_boot), function(i) {
  data_for_boot$freq_orig <- data_for_boot$freq_orig +
    sample(residuals, replace = TRUE)
  data_for_boot$freq <- data_for_boot$freq_orig
  booted_firstpass <- bunching::fit_bunching(data_for_boot,
                                             model, binwidth, notch, zD_bin)
})
set.seed(99)
ans = list()
extract_me <- function(boot_results, ans) {
  thelist = seq(7, 12994, 13)
  for (i in thelist) {
    a = boot_results[[i]]
    ans <- append(ans, a)
  }
  return(ans)
}
```

```
  betas = extract_me(boot_results = boot_results, ans = ans)
  betas = unlist(betas, use.names=FALSE)
  b_sd <- stats::sd(betas, na.rm = TRUE)

  #####Create df
  info = cbind.data.frame(b_estimate, b_sd)
  names(info) = c('beta', 'standard error')
  info$threshold = zstar
  info <- info[,c(3,1:2)]
  return(info)
}

# Create function for rounding values in tables
rounding = function(x) {
  A = round(x,digits=3)
  return(A)
}

# Create function to measure value of excess trades in bunching region
excess_value = function(thedata, z_vector, binv, zstar, bins_excl_l, bins_excl_r, rates) {
  z_vector = thedata$base

  ##set certain defaults
  binwidth = 10
  bins_l = 25
  bins_r = 25
  poly = 3 # degree of polynomial
  n_boot = 1000 # number of iterations for caclulating bootstrapped standard errors

  zmax <- zstar + (binwidth * bins_r)
  zmin <- zstar - (binwidth * bins_l)
  bins <- seq(zmin, zmax, by = binwidth)

  ##Cut the bins
  thebin <- cut(z_vector, bins, right = FALSE, labels = FALSE)
  thebin <- zmin + binwidth * (thebin - 1)
  thedata <- data.frame(z = z_vector, bin = thebin)
  thedata <- thedata %>% dplyr::group_by(bin) %>% dplyr::summarise(freq = n(),
                                                                   z = mean(z, na.rm = TRUE))      %>% d
  thedata$freq_orig <- thedata$freq
  thedata <- as.data.frame(thedata)

  data_binned = thedata

  data_binned$z_rel = (data_binned$bin - zstar)/binwidth
  data_binned$zstar <- ifelse(data_binned$bin == zstar, 1,
                              0)

  extra_fe_vector <- ""
  polynomial_vector <- c()
  for (i in seq(poly)) {
    poly_varname <- paste0("poly_", i)
    data_binned[[poly_varname]] <- data_binned$z_rel^i
```

```
  polynomial_vector <- c(polynomial_vector, poly_varname)
}
bins_excluded_all <- c()
if (bins_excl_l != 0) {
  bins_excl_l_vector <- c()
  for (i in seq(bins_excl_l)) {
    bins_excl_l_varname <- paste0("bin_excl_l_", i)
    data_binned[[bins_excl_l_varname]] <- ifelse(data_binned$z_rel ==
                                                   -i, 1, 0)
    bins_excl_l_vector <- c(bins_excl_l_vector, bins_excl_l_varname)
  }
  bins_excluded_all <- c(bins_excluded_all, bins_excl_l_vector)
}
if (bins_excl_r != 0) {
  bins_excl_r_vector <- c()
  for (i in seq(bins_excl_r)) {
    bin_excl_r_varname <- paste0("bin_excl_r_", i)
    data_binned[[bin_excl_r_varname]] <- ifelse(data_binned$z_rel ==
                                                  i, 1, 0)
    bins_excl_r_vector <- c(bins_excl_r_vector, bin_excl_r_varname)
  }
  bins_excluded_all <- c(bins_excluded_all, bins_excl_r_vector)
}
if (length(bins_excluded_all) > 0) {
  data_binned$bunch_region <- rowSums(data_binned[, c("zstar",
                                                      bins_excluded_all)])
}
data_binned$bin_above_excluded <- ifelse(data_binned$bin >
                                           zstar, 1, 0)
rn_vector <- ""
rhs_vars <- c("zstar", extra_fe_vector, polynomial_vector,
              rn_vector, bins_excluded_all)
rhs_vars <- setdiff(rhs_vars, "")
model_formula <- stats::as.formula(paste0("freq", " ~ ",
                                          paste(rhs_vars, collapse = " +")))

data_forreg <- list(data_binned = data_binned, model_formula = model_formula)

#######
thedata = data_forreg$data_binned
themodelformula = data_forreg$model_formula
notch = FALSE
zD_bin = NA

#Get going
model_fit <- stats::lm(themodelformula, thedata)
coefficients <- summary(model_fit)$coefficients
residuals <- stats::residuals(model_fit)
thedata$cf <- stats::predict(model_fit, thedata)
thedata$cf <- thedata$cf - (thedata$zstar * coefficients["zstar",
                                                         "Estimate"])
bins_excluded_in_reg <- rownames(coefficients)[grepl("bin_excl",
                                                     rownames(coefficients))]
```

```
  for (i in bins_excluded_in_reg) {
    thedata$cf <- thedata$cf - (thedata[[i]] * coefficients[i,
                                                            "Estimate"])
  }
  bins_zstar_zu <- sum(grepl("bin_excl_r", rownames(coefficients)))
  bins_zl_zstar <- sum(grepl("bin_excl_l", rownames(coefficients))) +
    1
  zstarvalue <- thedata$bin[thedata$zstar == 1]
  zstarvalue = zstarvalue[1]
  binwidthvalue <- thedata$bin[2] - thedata$bin[1]
  thedata$zl_zstar <- ifelse(((thedata$bin >= (zstarvalue - (binwidthvalue *
                                                               (bins_zl_zstar - 1)))) & (thedata$bin <=
                               1, 0)
  thedata$zstar_zu <- ifelse((thedata$bin <= zstarvalue + (binwidthvalue *
                                                             bins_zstar_zu)) & (thedata$bin > zstarvalu
  thedata$bunch_region <- ifelse(thedata$zl_zstar == 1, "zl_zstar",
                                 ifelse(thedata$zstar_zu == 1, "zstar_zu", "outside_bunching"))
  sub = thedata[thedata$bunch_region=='zl_zstar',]
  sub$excess = sub$freq - sub$cf
  sub$excess_dollars = sub$excess*sub$bin
  a = sum(sub$excess_dollars)
  df = as.data.frame(list(a))
  names(df) = c("excess_bunching_fiat_value")
  return(df)
}


################################################################################
################## PART 2: ANALYSIS ############################################
################################################################################

################################################################################
### TABLE 1
################################################################################

# Set working directory to location of data files
# setwd("path to files")
setwd("/Users/knershi@middlebury.edu/Documents/Temporary/")

## 1. Import data
btc <-read.csv("reg_thresh_btc.csv")
eth <-read.csv("reg_thresh_eth.csv")

## Separate by threshold currency

# Bitcoin
btc.usd <- btc %>%
  filter(threshold_currency=="usd")
btc.eur <- btc %>%
  filter(threshold_currency=="eur")
btc.yen <- btc %>%
  filter(threshold_currency=="yen")
```

Table 1: Bitcoin USD Results

threshold	beta	standard error	t_statistic	count_transactions
490	0.484	2.113	0.229	137822
990	2.732	0.775	3.524	161605
1490	-0.780	0.298	2.618	74794

[a] N exchanges = 2: N trading pairs = 5.

```
# Ethereum
eth.usd <- eth %>%
  filter(threshold_currency=="usd")
eth.eur <- eth %>%
  filter(threshold_currency=="eur")
eth.yen <- eth %>%
  filter(threshold_currency=="yen")

## 2. Bitcoin analysis

## A. USD

b_range1 = btc.usd[(btc.usd$base >= 250 & btc.usd$base < 750),]
b_range2 = btc.usd[(btc.usd$base >= 750 & btc.usd$base < 1250),]
b_range3 = btc.usd[(btc.usd$base >= 1250 & btc.usd$base < 1750),]

##Get estimates
set.seed(99)
b_r1 = estimate_beta(btc.usd, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
b_r2 = estimate_beta(btc.usd, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
b_r3 = estimate_beta(btc.usd, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

b_results = rbind.data.frame(b_r1,b_r2,b_r3)

b_results$t_statistic = abs(b_results$beta)/b_results$`standard error`
b_results[,c(2:4)] = sapply(b_results[,c(2:4)],rounding)
b_transactions = c(count(b_range1),count(b_range2),count(b_range3))
b_transactions = unlist(b_transactions)
b_results$count_transactions = b_transactions

# Total exchanges and trading pairs
b_N_exchanges <- sum(length(unique(btc.usd$exchange)))
b_N_trading_pairs <- sum(length(unique(paste0(btc.usd$exchange,btc.usd$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote1b <- paste0("N exchanges = ",b_N_exchanges,": N trading pairs = ",b_N_trading_pairs,".")

# Display Table
kable(b_results, caption = "Bitcoin USD Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote1b)
```

```
## B. EUR
```

Table 2: Bitcoin EUR Results

threshold	beta	standard error	t_statistic	count_transactions
490	0.001	0.234	0.004	1302271
990	2.143	0.760	2.821	1180226
1490	0.023	0.139	0.166	399601

[a] N exchanges = 1: N trading pairs = 3.

```
b2_range1 = btc.eur[(btc.eur$base >= 250 & btc.eur$base < 750),]
b2_rangb2 = btc.eur[(btc.eur$base >= 750 & btc.eur$base < 1250),]
b2_range3 = btc.eur[(btc.eur$base >= 1250 & btc.eur$base < 1750),]

##Get estimates
b2_r1 = estimate_beta(btc.eur, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
b2_r2 = estimate_beta(btc.eur, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
b2_r3 = estimate_beta(btc.eur, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

b2_results = rbind.data.frame(b2_r1,b2_r2,b2_r3)

b2_results$t_statistic = abs(b2_results$beta)/b2_results$`standard error`
b2_results[,c(2:4)] = sapply(b2_results[,c(2:4)],rounding)
b2_transactions = c(count(b2_range1),count(b2_rangb2),count(b2_range3))
b2_transactions = unlist(b2_transactions)
b2_results$count_transactions = b2_transactions

# Total exchanges and trading pairs
b2_N_exchanges <- sum(length(unique(btc.eur$exchange)))
b2_N_trading_pairs <- sum(length(unique(paste0(btc.eur$exchange,btc.eur$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote2 <- paste0("N exchanges = ",b2_N_exchanges,": N trading pairs = ",b2_N_trading_pairs,".")

# Display Table
kable(b2_results, caption = "Bitcoin EUR Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote2)
```

```
## C. YEN

b3_range1 = btc.yen[(btc.yen$base >= 250 & btc.yen$base < 750),]
b3_rangb3 = btc.yen[(btc.yen$base >= 750 & btc.yen$base < 1250),]
b3_range3 = btc.yen[(btc.yen$base >= 1250 & btc.yen$base < 1750),]

##Get estimates
b3_r1 = estimate_beta(btc.yen, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
b3_r2 = estimate_beta(btc.yen, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
b3_r3 = estimate_beta(btc.yen, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

b3_results = rbind.data.frame(b3_r1,b3_r2,b3_r3)

b3_results$t_statistic = abs(b3_results$beta)/b3_results$`standard error`
b3_results[,c(2:4)] = sapply(b3_results[,c(2:4)],rounding)
```

Table 3: Bitcoin YEN Results

threshold	beta	standard error	t_statistic	count_transactions
490	7.292	4.628	1.576	5007865
990	8.431	2.260	3.730	2431032
1490	28.255	154.470	0.183	804864

[a] N exchanges = 4: N trading pairs = 6.

```
b3_transactions = c(count(b3_range1),count(b3_rangb3),count(b3_range3))
b3_transactions = unlist(b3_transactions)
b3_results$count_transactions = b3_transactions

# Total exchanges and trading pairs
b3_N_exchanges <- sum(length(unique(btc.yen$exchange)))
b3_N_trading_pairs <- sum(length(unique(paste0(btc.yen$exchange,btc.yen$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote3 <- paste0("N exchanges = ",b3_N_exchanges,": N trading pairs = ",b3_N_trading_pairs,".")

# Display Table
kable(b3_results, caption = "Bitcoin YEN Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote3)
```

```
## 4. Ethereum analysis

## A. USD

e_range1 = eth.usd[(eth.usd$base >= 250 & eth.usd$base < 750),]
e_range2 = eth.usd[(eth.usd$base >= 750 & eth.usd$base < 1250),]
e_range3 = eth.usd[(eth.usd$base >= 1250 & eth.usd$base < 1750),]

##Get estimates
e_r1 = estimate_beta(eth.usd, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
e_r2 = estimate_beta(eth.usd, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
e_r3 = estimate_beta(eth.usd, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

e_results = rbind.data.frame(e_r1,e_r2,e_r3)

e_results$t_statistic = abs(e_results$beta)/e_results$`standard error`
e_results[,c(2:4)] = sapply(e_results[,c(2:4)],rounding)
e_transactions = c(count(e_range1),count(e_range2),count(e_range3))
e_transactions = unlist(e_transactions)
e_results$count_transactions = e_transactions

# Total exchanges and trading pairs
e_N_exchanges <- sum(length(unique(eth.usd$exchange)))
e_N_trading_pairs <- sum(length(unique(paste0(eth.usd$exchange,eth.usd$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote1 <- paste0("N exchanges = ",e_N_exchanges,": N trading pairs = ",e_N_trading_pairs,".")
```

Table 4: Ethereum USD Results

threshold	beta	standard error	t_statistic	count_transactions
490	0.148	0.790	0.187	129240
990	2.641	0.636	4.151	63760
1490	-0.144	1.240	0.116	23049

[a] N exchanges = 2: N trading pairs = 2.

```
# Display Table
kable(e_results, caption = "Ethereum USD Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote1)
```

```
## B. EUR

e2_range1 = eth.eur[(eth.eur$base >= 250 & eth.eur$base < 750),]
e2_range2 = eth.eur[(eth.eur$base >= 750 & eth.eur$base < 1250),]
e2_range3 = eth.eur[(eth.eur$base >= 1250 & eth.eur$base < 1750),]

##Get estimates
e2_r1 = estimate_beta(eth.eur, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
e2_r2 = estimate_beta(eth.eur, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
e2_r3 = estimate_beta(eth.eur, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

e2_results = rbind.data.frame(e2_r1,e2_r2,e2_r3)

e2_results$t_statistic = abs(e2_results$beta)/e2_results$`standard error`
e2_results[,c(2:4)] = sapply(e2_results[,c(2:4)],rounding)
e2_transactions = c(count(e2_range1),count(e2_range2),count(e2_range3))
e2_transactions = unlist(e2_transactions)
e2_results$count_transactions = e2_transactions

# Total exchanges and trading pairs
e2_N_exchanges <- sum(length(unique(eth.eur$exchange)))
e2_N_trading_pairs <- sum(length(unique(paste0(eth.eur$exchange,eth.eur$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote2 <- paste0("N exchanges = ",e2_N_exchanges,": N trading pairs = ",e2_N_trading_pairs,".")

# Display Table
kable(e2_results, caption = "Ethereum EUR Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote2)
```

```
## C. YEN

e3_range1 = eth.yen[(eth.yen$base >= 250 & eth.yen$base < 750),]
e3_range2 = eth.yen[(eth.yen$base >= 750 & eth.yen$base < 1250),]
e3_range3 = eth.yen[(eth.yen$base >= 1250 & eth.yen$base < 1750),]

##Get estimates
e3_r1 = estimate_beta(eth.yen, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
```

Table 5: Ethereum EUR Results

threshold	beta	standard error	t_statistic	count_transactions
490	0.033	0.385	0.086	1103214
990	2.045	0.464	4.409	879677
1490	0.236	0.332	0.710	365390

[a] N exchanges = 2: N trading pairs = 4.

Table 6: Ethereum YEN Results

threshold	beta	standard error	t_statistic	count_transactions
490	5.240	1.536	3.411	143507
990	-1.311	0.929	1.412	59978
1490	2.102	0.995	2.113	45270

[a] N exchanges = 3: N trading pairs = 3.

```
e3_r2 = estimate_beta(eth.yen, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
e3_r3 = estimate_beta(eth.yen, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

e3_results = rbind.data.frame(e3_r1,e3_r2,e3_r3)

e3_results$t_statistic = abs(e3_results$beta)/e3_results$`standard error`
e3_results[,c(2:4)] = sapply(e3_results[,c(2:4)],rounding)
e3_transactions = c(count(e3_range1),count(e3_range2),count(e3_range3))
e3_transactions = unlist(e3_transactions)
e3_results$count_transactions = e3_transactions

# Total exchanges and trading pairs
e3_N_exchanges <- sum(length(unique(eth.yen$exchange)))
e3_N_trading_pairs <- sum(length(unique(paste0(eth.yen$exchange,eth.yen$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote3 <- paste0("N exchanges = ",e3_N_exchanges,": N trading pairs = ",e3_N_trading_pairs,".")

# Display Table
kable(e3_results, caption = "Ethereum YEN Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote3)
```

```
################################################################################
### TABLE 4
################################################################################

## 1. Estimate values

### Bitcoin

## USD
btc_dollars <- excess_value(btc.usd, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
print(btc_dollars)
```

```
##   excess_bunching_fiat_value
## 1                    8626426
```

```
total_dollar_val_in_range_btc <- btc.usd %>%
  filter(base >= 750 & base < 1250) %>%
  summarise(sum_trade_val = sum(base))

## Percentage of trades in dollar value
round(unname(btc_dollars/total_dollar_val_in_range_btc),2)
```

```
##
## 1 0.05
```

```
## EUR
btc_euros <- excess_value(btc.eur, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)

## Average eur-usd exchange rate between July 1, 2020 - August 31, 2020 = 1.18461
# Source: ECBeuropa.eu
btc_euros_in_dollars <- btc_euros*1.1637
print(btc_euros_in_dollars)
```

```
##   excess_bunching_fiat_value
## 1                   52653878
```

```
total_euro_val_in_range_btc <- btc.eur %>%
  filter(base >= 750 & base < 1250) %>%
  summarise(sum_trade_val = sum(base))

## Percentage of trades in dollar value
round(unname(btc_euros/total_euro_val_in_range_btc),2)
```

```
##
## 1 0.04
```

```
## YEN
btc_yen <- excess_value(btc.yen, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)

## Average yen-usd exchange rate (Open) between July 1, 2020 - August 31, 2020 = 0.0093
# Source: Yahoo finance
btc_yen_in_dollars <- btc_yen*100*0.0093
print(btc_yen_in_dollars)
```

```
##   excess_bunching_fiat_value
## 1                  246777354
```

```
total_yen_val_in_range_btc <- btc.yen %>%
  filter(base >= 750 & base < 1250) %>%
  summarise(sum_trade_val = sum(base))

## Percentage of trades in dollar value
round(unname(btc_yen/total_yen_val_in_range_btc),2)
```

```
##
## 1 0.11
```

```
### Ethereum

## USD
```

```r
eth_dollars <- excess_value(eth.usd, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
print(eth_dollars)
```

```
##   excess_bunching_fiat_value
## 1                    3187479
```

```r
total_dollar_val_in_range_eth <- eth.usd %>%
  filter(base >= 750 & base < 1250) %>%
  summarise(sum_trade_val = sum(base))

## Percentage of trades in dollar value
round(unname(eth_dollars/total_dollar_val_in_range_eth),2)
```

```
##
## 1 0.05
```

```r
## EUR
eth_euros <- excess_value(eth.eur, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)

## Average eur-usd exchange rate between July 1, 2020 - August 31, 2020 = 1.18461
# Source: ECBeuropa.eu
eth_euros_in_dollars <- eth_euros*1.1637
print(eth_euros_in_dollars)
```

```
##   excess_bunching_fiat_value
## 1                   35447929
```

```r
total_euro_val_in_range_eth <- eth.eur %>%
  filter(base >= 750 & base < 1250) %>%
  summarise(sum_trade_val = sum(base))

## Percentage of trades in dollar value
round(unname(eth_euros/total_euro_val_in_range_eth),2)
```

```
##
## 1 0.04
```

```r
##############################################################################
### TABLE 9
##############################################################################

# To avoid confusion, remove all model objects created for Table 1
# Define a list of prefixes you want to remove
prefixes_to_remove <- c("b_", "b2_", "b3_",
                        "e_", "e2_", "e3_",
                        "my_")

# Create a pattern that matches any of these prefixes at the beginning of object names
pattern <- paste0("^(", paste(prefixes_to_remove, collapse="|"), ")")

# Remove objects matching the pattern
rm(list=ls(pattern=pattern))

## 3. Bitcoin analysis

## A. USD
```

Table 7: Bitcoin USD Results

threshold	beta	standard error	t_statistic	count_transactions
490	1.425	1.733	0.822	137822
990	1.664	0.566	2.938	161605
1490	-0.077	0.271	0.284	74794

[a] N exchanges = 2: N trading pairs = 5.

```
b_range1 = btc.usd[(btc.usd$base >= 250 & btc.usd$base < 750),]
b_range2 = btc.usd[(btc.usd$base >= 750 & btc.usd$base < 1250),]
b_range3 = btc.usd[(btc.usd$base >= 1250 & btc.usd$base < 1750),]

##Get estimates
b_r1 = estimate_beta(btc.usd, z_vector, binv, zstar=490, bins_excl_l=6, bins_excl_r=0)
b_r2 = estimate_beta(btc.usd, z_vector, binv, zstar=990, bins_excl_l=6, bins_excl_r=0)
b_r3 = estimate_beta(btc.usd, z_vector, binv, zstar=1490, bins_excl_l=6, bins_excl_r=0)

b_results = rbind.data.frame(b_r1,b_r2,b_r3)

b_results$t_statistic = abs(b_results$beta)/b_results$`standard error`
b_results[,c(2:4)] = sapply(b_results[,c(2:4)],rounding)
b_transactions = c(count(b_range1),count(b_range2),count(b_range3))
b_transactions = unlist(b_transactions)
b_results$count_transactions = b_transactions

# Total exchanges and trading pairs
b_N_exchanges <- sum(length(unique(btc.usd$exchange)))
b_N_trading_pairs <- sum(length(unique(paste0(btc.usd$exchange,btc.usd$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote1b <- paste0("N exchanges = ",b_N_exchanges,": N trading pairs = ",b_N_trading_pairs,".")

# Display Table
kable(b_results, caption = "Bitcoin USD Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote1b)
```

```
## B. EUR

b2_range1 = btc.eur[(btc.eur$base >= 250 & btc.eur$base < 750),]
b2_rangb2 = btc.eur[(btc.eur$base >= 750 & btc.eur$base < 1250),]
b2_range3 = btc.eur[(btc.eur$base >= 1250 & btc.eur$base < 1750),]

##Get estimates
b2_r1 = estimate_beta(btc.eur, z_vector, binv, zstar=490, bins_excl_l=6, bins_excl_r=0)
b2_r2 = estimate_beta(btc.eur, z_vector, binv, zstar=990, bins_excl_l=6, bins_excl_r=0)
b2_r3 = estimate_beta(btc.eur, z_vector, binv, zstar=1490, bins_excl_l=6, bins_excl_r=0)

b2_results = rbind.data.frame(b2_r1,b2_r2,b2_r3)

b2_results$t_statistic = abs(b2_results$beta)/b2_results$`standard error`
```

Table 8: Bitcoin EUR Results

threshold	beta	standard error	t_statistic	count_transactions
490	0.446	0.188	2.376	1302271
990	0.520	0.564	0.922	1180226
1490	0.022	0.107	0.205	399601

[a] N exchanges = 1: N trading pairs = 3.

```
b2_results[,c(2:4)] = sapply(b2_results[,c(2:4)],rounding)
b2_transactions = c(count(b2_range1),count(b2_rangb2),count(b2_range3))
b2_transactions = unlist(b2_transactions)
b2_results$count_transactions = b2_transactions

# Total exchanges and trading pairs
b2_N_exchanges <- sum(length(unique(btc.eur$exchange)))
b2_N_trading_pairs <- sum(length(unique(paste0(btc.eur$exchange,btc.eur$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote2 <- paste0("N exchanges = ",b2_N_exchanges,": N trading pairs = ",b2_N_trading_pairs,".")

# Display Table
kable(b2_results, caption = "Bitcoin EUR Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote2)
```

```
## C. YEN

b3_range1 = btc.yen[(btc.yen$base >= 250 & btc.yen$base < 750),]
b3_rangb3 = btc.yen[(btc.yen$base >= 750 & btc.yen$base < 1250),]
b3_range3 = btc.yen[(btc.yen$base >= 1250 & btc.yen$base < 1750),]

##Get estimates
b3_r1 = estimate_beta(btc.yen, z_vector, binv, zstar=490, bins_excl_l=6, bins_excl_r=0)
b3_r2 = estimate_beta(btc.yen, z_vector, binv, zstar=990, bins_excl_l=6, bins_excl_r=0)
b3_r3 = estimate_beta(btc.yen, z_vector, binv, zstar=1490, bins_excl_l=6, bins_excl_r=0)

b3_results = rbind.data.frame(b3_r1,b3_r2,b3_r3)

b3_results$t_statistic = abs(b3_results$beta)/b3_results$`standard error`
b3_results[,c(2:4)] = sapply(b3_results[,c(2:4)],rounding)
b3_transactions = c(count(b3_range1),count(b3_rangb3),count(b3_range3))
b3_transactions = unlist(b3_transactions)
b3_results$count_transactions = b3_transactions

# Total exchanges and trading pairs
b3_N_exchanges <- sum(length(unique(btc.yen$exchange)))
b3_N_trading_pairs <- sum(length(unique(paste0(btc.yen$exchange,btc.yen$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote3 <- paste0("N exchanges = ",b3_N_exchanges,": N trading pairs = ",b3_N_trading_pairs,".")

# Display Table
```

Table 9: Bitcoin YEN Results

threshold	beta	standard error	t_statistic	count_transactions
490	5.288	2.798	1.890	5007865
990	6.706	1.510	4.440	2431032
1490	14.578	10.499	1.388	804864

[a] N exchanges = 4: N trading pairs = 6.

```
kable(b3_results, caption = "Bitcoin YEN Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote3)
```

```
## 4. Ethereum analysis

## A. USD

e_range1 = eth.usd[(eth.usd$base >= 250 & eth.usd$base < 750),]
e_range2 = eth.usd[(eth.usd$base >= 750 & eth.usd$base < 1250),]
e_range3 = eth.usd[(eth.usd$base >= 1250 & eth.usd$base < 1750),]

##Get estimates
e_r1 = estimate_beta(eth.usd, z_vector, binv, zstar=490, bins_excl_l=6, bins_excl_r=0)
e_r2 = estimate_beta(eth.usd, z_vector, binv, zstar=990, bins_excl_l=6, bins_excl_r=0)
e_r3 = estimate_beta(eth.usd, z_vector, binv, zstar=1490, bins_excl_l=6, bins_excl_r=0)

e_results = rbind.data.frame(e_r1,e_r2,e_r3)

e_results$t_statistic = abs(e_results$beta)/e_results$`standard error`
e_results[,c(2:4)] = sapply(e_results[,c(2:4)],rounding)
e_transactions = c(count(e_range1),count(e_range2),count(e_range3))
e_transactions = unlist(e_transactions)
e_results$count_transactions = e_transactions

# Total exchanges and trading pairs
e_N_exchanges <- sum(length(unique(eth.usd$exchange)))
e_N_trading_pairs <- sum(length(unique(paste0(eth.usd$exchange,eth.usd$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote1 <- paste0("N exchanges = ",e_N_exchanges,": N trading pairs = ",e_N_trading_pairs,".")

# Display Table
kable(e_results, caption = "Ethereum USD Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote1)
```

```
## B. EUR

e2_range1 = eth.eur[(eth.eur$base >= 250 & eth.eur$base < 750),]
e2_range2 = eth.eur[(eth.eur$base >= 750 & eth.eur$base < 1250),]
e2_range3 = eth.eur[(eth.eur$base >= 1250 & eth.eur$base < 1750),]

##Get estimates
```

Table 10: Ethereum USD Results

threshold	beta	standard error	t_statistic	count_transactions
490	0.195	0.597	0.327	129240
990	2.446	0.471	5.195	63760
1490	0.419	0.989	0.424	23049

[a] N exchanges = 2: N trading pairs = 2.

Table 11: Ethereum EUR Results

threshold	beta	standard error	t_statistic	count_transactions
490	0.039	0.288	0.135	1103214
990	1.133	0.360	3.149	879677
1490	0.223	0.247	0.902	365390

[a] N exchanges = 2: N trading pairs = 4.

```
e2_r1 = estimate_beta(eth.eur, z_vector, binv, zstar=490, bins_excl_l=6, bins_excl_r=0)
e2_r2 = estimate_beta(eth.eur, z_vector, binv, zstar=990, bins_excl_l=6, bins_excl_r=0)
e2_r3 = estimate_beta(eth.eur, z_vector, binv, zstar=1490, bins_excl_l=6, bins_excl_r=0)

e2_results = rbind.data.frame(e2_r1,e2_r2,e2_r3)

e2_results$t_statistic = abs(e2_results$beta)/e2_results$`standard error`
e2_results[,c(2:4)] = sapply(e2_results[,c(2:4)],rounding)
e2_transactions = c(count(e2_range1),count(e2_range2),count(e2_range3))
e2_transactions = unlist(e2_transactions)
e2_results$count_transactions = e2_transactions

# Total exchanges and trading pairs
e2_N_exchanges <- sum(length(unique(eth.eur$exchange)))
e2_N_trading_pairs <- sum(length(unique(paste0(eth.eur$exchange,eth.eur$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote2 <- paste0("N exchanges = ",e2_N_exchanges,": N trading pairs = ",e2_N_trading_pairs,".")

# Display Table
kable(e2_results, caption = "Ethereum EUR Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote2)
```

```
## C. YEN

e3_range1 = eth.yen[(eth.yen$base >= 250 & eth.yen$base < 750),]
e3_range2 = eth.yen[(eth.yen$base >= 750 & eth.yen$base < 1250),]
e3_range3 = eth.yen[(eth.yen$base >= 1250 & eth.yen$base < 1750),]

##Get estimates
e3_r1 = estimate_beta(eth.yen, z_vector, binv, zstar=490, bins_excl_l=6, bins_excl_r=0)
e3_r2 = estimate_beta(eth.yen, z_vector, binv, zstar=990, bins_excl_l=6, bins_excl_r=0)
e3_r3 = estimate_beta(eth.yen, z_vector, binv, zstar=1490, bins_excl_l=6, bins_excl_r=0)
```

Table 12: Ethereum YEN Results

threshold	beta	standard error	t_statistic	count_transactions
490	-1.214	1.201	1.011	143507
990	-1.105	0.763	1.448	59978
1490	1.475	0.666	2.216	45270

[a] N exchanges = 3: N trading pairs = 3.

```
e3_results = rbind.data.frame(e3_r1,e3_r2,e3_r3)

e3_results$t_statistic = abs(e3_results$beta)/e3_results$`standard error`
e3_results[,c(2:4)] = sapply(e3_results[,c(2:4)],rounding)
e3_transactions = c(count(e3_range1),count(e3_range2),count(e3_range3))
e3_transactions = unlist(e3_transactions)
e3_results$count_transactions = e3_transactions

# Total exchanges and trading pairs
e3_N_exchanges <- sum(length(unique(eth.yen$exchange)))
e3_N_trading_pairs <- sum(length(unique(paste0(eth.yen$exchange,eth.yen$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote3 <- paste0("N exchanges = ",e3_N_exchanges,": N trading pairs = ",e3_N_trading_pairs,".")

# Display Table
kable(e3_results, caption = "Ethereum YEN Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote3)
```

```
################################################################################
### TABLE 2
################################################################################

# To avoid confusion, remove all model objects created for previous models.
# Define a list of prefixes you want to remove
prefixes_to_remove <- c("b_", "b2_", "b3_",
                        "e_", "e2_", "e3_",
                        "btc","eth",
                        "my_")

# Create a pattern that matches any of these prefixes at the beginning of object names
pattern <- paste0("^(", paste(prefixes_to_remove, collapse="|"), ")")

# Remove objects matching the pattern
rm(list=ls(pattern=pattern))

## 1. Import data
file_name = paste0("/Users/knershi@middlebury.edu/Documents/Temporary/","unreg_btc.csv")
btc <-read.csv(file_name)

file_name2 = paste0("/Users/knershi@middlebury.edu/Documents/Temporary/","unreg_eth.csv")
eth <-read.csv(file_name2)
```

```r
## Separate by threshold currency

# Bitcoin
btc.usd <- btc %>%
  filter(threshold_currency=="usd")
btc.eur <- btc %>%
  filter(threshold_currency=="eur")

# Ethereum
eth.usd <- eth %>%
  filter(threshold_currency=="usd")
eth.eur <- eth %>%
  filter(threshold_currency=="eur")

## 2. Bitcoin analysis

## A. USD

b_range1 = btc.usd[(btc.usd$base >= 250 & btc.usd$base < 750),]
b_range2 = btc.usd[(btc.usd$base >= 750 & btc.usd$base < 1250),]
b_range3 = btc.usd[(btc.usd$base >= 1250 & btc.usd$base < 1750),]

##Get estimates
b_r1 = estimate_beta(btc.usd, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
b_r2 = estimate_beta(btc.usd, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
b_r3 = estimate_beta(btc.usd, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

b_results = rbind.data.frame(b_r1,b_r2,b_r3)

b_results$t_statistic = abs(b_results$beta)/b_results$`standard error`
b_results[,c(2:4)] = sapply(b_results[,c(2:4)],rounding)
b_transactions = c(count(b_range1),count(b_range2),count(b_range3))
b_transactions = unlist(b_transactions)
b_results$count_transactions = b_transactions

# Total exchanges and trading pairs
b_N_exchanges <- sum(length(unique(btc.usd$exchange)))
b_N_trading_pairs <- sum(length(unique(paste0(btc.usd$exchange,btc.usd$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote1b <- paste0("N exchanges = ",b_N_exchanges,": N trading pairs = ",b_N_trading_pairs,".")

# Display Table
kable(b_results, caption = "Bitcoin USD Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote1b)
```

```r
## B. EUR

b2_range1 = btc.eur[(btc.eur$base >= 250 & btc.eur$base < 750),]
b2_rangb2 = btc.eur[(btc.eur$base >= 750 & btc.eur$base < 1250),]
b2_range3 = btc.eur[(btc.eur$base >= 1250 & btc.eur$base < 1750),]
```

Table 13: Bitcoin USD Results

threshold	beta	standard error	t_statistic	count_transactions
490	3.511	0.641	5.473	229213
990	-0.936	1.253	0.747	94209
1490	0.099	0.672	0.147	50773

[a] N exchanges = 4: N trading pairs = 4.

Table 14: Bitcoin EUR Results

threshold	beta	standard error	t_statistic	count_transactions
490	-0.762	0.836	0.912	25642
990	-1.875	2.079	0.902	20588
1490	-0.487	0.672	0.725	10537

[a] N exchanges = 1: N trading pairs = 1.

```
##Get estimates
b2_r1 = estimate_beta(btc.eur, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
b2_r2 = estimate_beta(btc.eur, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
b2_r3 = estimate_beta(btc.eur, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

b2_results = rbind.data.frame(b2_r1,b2_r2,b2_r3)

b2_results$t_statistic = abs(b2_results$beta)/b2_results$`standard error`
b2_results[,c(2:4)] = sapply(b2_results[,c(2:4)],rounding)
b2_transactions = c(count(b2_range1),count(b2_rangb2),count(b2_range3))
b2_transactions = unlist(b2_transactions)
b2_results$count_transactions = b2_transactions

# Total exchanges and trading pairs
b2_N_exchanges <- sum(length(unique(btc.eur$exchange)))
b2_N_trading_pairs <- sum(length(unique(paste0(btc.eur$exchange,btc.eur$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote2 <- paste0("N exchanges = ",b2_N_exchanges,": N trading pairs = ",b2_N_trading_pairs,".")

# Display Table
kable(b2_results, caption = "Bitcoin EUR Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote2)
```

```
## 4. Ethereum analysis

## A. USD

e_range1 = eth.usd[(eth.usd$base >= 250 & eth.usd$base < 750),]
e_range2 = eth.usd[(eth.usd$base >= 750 & eth.usd$base < 1250),]
e_range3 = eth.usd[(eth.usd$base >= 1250 & eth.usd$base < 1750),]

##Get estimates
```

Table 15: Ethereum USD Results

threshold	beta	standard error	t_statistic	count_transactions
490	2.188	0.993	2.203	100537
990	-0.922	1.462	0.631	33876
1490	-0.271	1.202	0.226	18710

[a] N exchanges = 3: N trading pairs = 3.

```
e_r1 = estimate_beta(eth.usd, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
e_r2 = estimate_beta(eth.usd, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
e_r3 = estimate_beta(eth.usd, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

e_results = rbind.data.frame(e_r1,e_r2,e_r3)

e_results$t_statistic = abs(e_results$beta)/e_results$`standard error`
e_results[,c(2:4)] = sapply(e_results[,c(2:4)],rounding)
e_transactions = c(count(e_range1),count(e_range2),count(e_range3))
e_transactions = unlist(e_transactions)
e_results$count_transactions = e_transactions

# Total exchanges and trading pairs
e_N_exchanges <- sum(length(unique(eth.usd$exchange)))
e_N_trading_pairs <- sum(length(unique(paste0(eth.usd$exchange,eth.usd$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote1 <- paste0("N exchanges = ",e_N_exchanges,": N trading pairs = ",e_N_trading_pairs,".")

# Display Table
kable(e_results, caption = "Ethereum USD Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote1)
```

```
## B. EUR

e2_range1 = eth.eur[(eth.eur$base >= 250 & eth.eur$base < 750),]
e2_range2 = eth.eur[(eth.eur$base >= 750 & eth.eur$base < 1250),]
e2_range3 = eth.eur[(eth.eur$base >= 1250 & eth.eur$base < 1750),]

##Get estimates
e2_r1 = estimate_beta(eth.eur, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
e2_r2 = estimate_beta(eth.eur, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
e2_r3 = estimate_beta(eth.eur, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

e2_results = rbind.data.frame(e2_r1,e2_r2,e2_r3)

e2_results$t_statistic = abs(e2_results$beta)/e2_results$`standard error`
e2_results[,c(2:4)] = sapply(e2_results[,c(2:4)],rounding)
e2_transactions = c(count(e2_range1),count(e2_range2),count(e2_range3))
e2_transactions = unlist(e2_transactions)
e2_results$count_transactions = e2_transactions

# Total exchanges and trading pairs
```

Table 16: Ethereum EUR Results

threshold	beta	standard error	t_statistic	count_transactions
490	0.872	2.204	0.396	64118
990	-0.595	0.493	1.207	30360
1490	2.858	2.066	1.383	9381

[a] N exchanges = 2: N trading pairs = 2.

```
e2_N_exchanges <- sum(length(unique(eth.eur$exchange)))
e2_N_trading_pairs <- sum(length(unique(paste0(eth.eur$exchange,eth.eur$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote2 <- paste0("N exchanges = ",e2_N_exchanges,": N trading pairs = ",e2_N_trading_pairs,".")

# Display Table
kable(e2_results, caption = "Ethereum EUR Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote2)
```

```
################################################################################
### TABLE 3
################################################################################

# To avoid confusion, remove all model objects created for previous models.
# Define a list of prefixes you want to remove
prefixes_to_remove <- c("b_", "b2_", "b3_",
                        "e_", "e2_", "e3_",
                        "btc","eth",
                        "my_")

# Create a pattern that matches any of these prefixes at the beginning of object names
pattern <- paste0("^(", paste(prefixes_to_remove, collapse="|"), ")")

# Remove objects matching the pattern
rm(list=ls(pattern=pattern))

## 1. Import data
df <-read.csv("full.reg.without.threshold.csv")

## Separate by cryptocurrency
df.btc <- df %>%
  filter(base_cryptocurrency=="btc")
df.eth <- df %>%
  filter(base_cryptocurrency=="eth")

## 2. Bitcoin analysis

# Transactions in range
b_range1 = df.btc[(df.btc$base >= 250 & df.btc$base < 750),]
b_range2 = df.btc[(df.btc$base >= 750 & df.btc$base < 1250),]
b_range3 = df.btc[(df.btc$base >= 1250 & df.btc$base < 1750),]
```

Table 17: Bitcoin Results

threshold	beta	standard error	t_statistic	count_transactions
490	3.326	1.518	2.190	228834
990	0.748	1.648	0.454	203284
1490	-0.530	0.279	1.902	83937

[a] N exchanges = 4: N trading pairs = 4.

```
##Get estimates
set.seed(99)
b_r1 = estimate_beta(df.btc, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
b_r2 = estimate_beta(df.btc, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
b_r3 = estimate_beta(df.btc, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)
b_results = rbind.data.frame(b_r1,b_r2,b_r3)

b_results$t_statistic = abs(b_results$beta)/b_results$`standard error`
b_results[,c(2:4)] = sapply(b_results[,c(2:4)],rounding)
b_transactions = c(count(b_range1),count(b_range2),count(b_range3))
b_transactions = unlist(b_transactions)
b_results$count_transactions = b_transactions

# Total exchanges and trading pairs
b_N_exchanges <- sum(length(unique(df.btc$exchange)))
b_N_trading_pairs <- sum(length(unique(paste0(df.btc$exchange,df.btc$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote <- paste0("N exchanges = ",b_N_exchanges,": N trading pairs = ",b_N_trading_pairs,".")

# Display Table
kable(b_results, caption = "Bitcoin Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote)
```

```
## 3. Ethereum analysis

#Transactions in range
e_range1 = df.eth[(df.eth$base >= 250 & df.eth$base < 750),]
e_range2 = df.eth[(df.eth$base >= 750 & df.eth$base < 1250),]
e_range3 = df.eth[(df.eth$base >= 1250 & df.eth$base < 1750),]

##Get estimates
e_r1 = estimate_beta(df.eth, z_vector, binv, zstar=490, bins_excl_l=9, bins_excl_r=0)
e_r2 = estimate_beta(df.eth, z_vector, binv, zstar=990, bins_excl_l=9, bins_excl_r=0)
e_r3 = estimate_beta(df.eth, z_vector, binv, zstar=1490, bins_excl_l=9, bins_excl_r=0)

e_results = rbind.data.frame(e_r1,e_r2,e_r3)

e_results$t_statistic = abs(e_results$beta)/e_results$`standard error`
e_results[,c(2:4)] = sapply(e_results[,c(2:4)],rounding)
e_transactions = c(count(e_range1),count(e_range2),count(e_range3))
e_transactions = unlist(e_transactions)
e_results$count_transactions = e_transactions
```

Table 18: Ethereum Results

threshold	beta	standard error	t_statistic	count_transactions
490	-2.010	2.202	0.913	92375
990	-4.915	2.483	1.979	35262
1490	0.030	0.797	0.038	10130

[a] N exchanges = 2: N trading pairs = 2.

```
# Total exchanges and trading pairs
e_N_exchanges <- sum(length(unique(df.eth$exchange)))
e_N_trading_pairs <- sum(length(unique(paste0(df.eth$exchange,df.eth$s))))

# Create a footnote showing the number of exchanges and trading pairs
my_footnote2 <- paste0("N exchanges = ",e_N_exchanges,": N trading pairs = ",e_N_trading_pairs,".")

# Display Table
kable(e_results, caption = "Ethereum Results", format = "latex", booktabs = T) %>%
  kable_styling() %>%
  add_footnote(my_footnote2)
```