**Code Example #1**

#########Analysis############

library(ggplot2)

library(directlabels)

library(margins)

library(reshape2)

library(scales)

library(openxlsx)

library(ggpubr)

#Unidimensional plot

**for** (a **in** statecycle) {

df <- leg.cand[leg.cand$election == a, ]

**name** <- paste(a,".png",sep='')

ggplot(df, aes(recipient\_cfscore, df$seat)) +

geom\_point(aes(**color** = party)) +

scale\_color\_manual(values = c("#429bf4","#f4414a","#686868")) +

theme\_minimal() +

ggsave(**name**, **device** = 'png',path = "/Users/Tom/OneDrive/MPhil/Thesis/Data/Bonica/Plots/")

}

statecycle <- unique(**all**.contribs$StateCycle)

#################################################################

################## INTER-CYCLE ANALYSIS ##################

#################################################################

##### A - DISTRIBUTION PLOTS #####

#Non-cumulative distributions plots of measures and legislature

**for** (i **in** statecycle) {

p <- ggplot(**data**=all.contribs[all.contribs$StateCycle == i,], aes(x= contributor\_cfscore, fill = id\_label, linetype= type)) +

geom\_density(alpha=0.25) +

labs(x = 'Ideological Position', y="Frequency")

  #Add plot labels, name, and save

p <- **direct**.label(p, "top.bumptwice")

**name** <- paste(i,"\_","density.png",sep='')

ggsave(**name**, **device** = 'png', plot = p, scale = 1.5, height = 4, width = 7, path = paste(dir,"/Plots/",sep=''))

rm(**name**)

}

##### Polarisation of legislatures #####

state.means <- **data**.frame(state = character(),

cycle = character(),

chamber = character(),

party = character(),

mean = double(),

stringsAsFactors = FALSE)

# Creates dataframe with mean legislative ideological contributions per party and chamber

**for** (i **in** c('2000','2002','2004','2006','2008','2010','2012')) {

**for** (j **in** c('AZ','CA','OR','WA', 'MA')) {

**for** (k **in** c('state:upper','state:lower')) {

**for** (l **in** c('100','200')) {

mn <- mean(**all**.contribs$contributor\_cfscore[**all**.contribs$cycle == i & all.contribs$recipient\_state == j & all.contribs$type == 'Legislature' & all.contribs$seat == k & all.contribs$recipient\_party == l],na.rm = TRUE)

row <- c(j,i,k,l, mn)

state.means[nrow(state.means)+1,] <- row

}

}

}

}

state.means$cycle <- as.integer(state.means$cycle)

state.means$mean <- as.double(state.means$mean)

state.means$abs <- abs(state.means$mean)

##Transform dataframe for results

state.means.pub <- dcast(state.means, state + cycle ~ party + chamber, value.var = "mean")

state.means.pub$cycle <- as.integer(state.means.pub$cycle)+2000

#Simple scatterplot over time

ggplot(state.means,

aes(x=cycle, y = mean, shape = chamber, color = party)) +

labs(title = "Mean Ideological Donation Score over Time", subtitle = "By state, year, party and chamber") +

geom\_point() +

geom\_smooth(method=lm) +

facet\_wrap(~state) +

scale\_color\_manual(values=c("blue","red")) +

scale\_x\_continuous(breaks=seq(2000,2012,2), labels=c("2000","2002","2004","2006","2008","2010","2012"))

##### Generates NCDF for legislative chambers, by party #####

**for** (i **in** statecycle) {

p1 <- ggplot(**data**=all.contribs[all.contribs$StateCycle == i & all.contribs$type == 'Legislature' & (all.contribs$recipient\_party == 100 | all.contribs$recipient\_party == 200),],

aes(x= contributor\_cfscore, fill = recipient\_party, linetype = id\_label)) +

geom\_density(alpha=0.25) +

scale\_fill\_manual(**name** = "Party", values=c("blue", "red")) +

scale\_linetype\_manual(**name** = "Chamber", values = c(1,2), labels = c("Democrat","Republican")) +

labs(x = 'Ideological Position', y="Frequency") +

xlim(-2,2)

**name** <- paste(i,"\_","leg\_parties.png",sep='')

ggsave(**name**, **device** = 'png', plot = p1, scale = 1.5, height = 4, width = 7, path = paste(dir,"/Plots/Legislature/",sep=''))

rm(**name**)

}

rm(p1)

### Issue area analysis

issues <- read.xlsx("Issue Descriptions.xlsx","Sheet1")

issues <- merge(measure.stats,issues,by=c("cycle","measure"))

xtabs( ~ extreme + broad, issues)

xtabs( ~ position + broad, issues)

chisq.test(issues$extreme,issues$broad)

GKtau(issues$extreme, issues$broad)

summary(factor(issues[issues$position=="Left",]$category))

summary(factor(issues[issues$position=="Centre",]$category))

summary(factor(issues[issues$position=="Right",]$category))

##### Multiplot function ######

#Credit: Cookbook-R.com: http://www.cookbook-r.com/Graphs/Multiple\_graphs\_on\_one\_page\_(ggplot2)/

# Multiple plot function

#

# ggplot objects can be passed in ..., or to plotlist (as a list of ggplot objects)

# - cols: Number of columns in layout

# - layout: A matrix specifying the layout. If present, 'cols' is ignored.

#

# If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE),

# then plot 1 will go in the upper left, 2 will go in the upper right, and

# 3 will go all the way across the bottom.

#

multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {

library(grid)

  # Make a list from the ... arguments and plotlist

plots <- c(**list**(...), plotlist)

numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout

if (is.null(layout)) {

    # Make the panel

    # ncol: Number of columns of plots

    # nrow: Number of rows needed, calculated from # of cols

layout <- matrix(seq(1, cols \* ceiling(numPlots/cols)),

ncol = cols, nrow = ceiling(numPlots/cols))

}

if (numPlots==1) {

print(plots[[1]])

} else {

    # Set up the page

grid.newpage()

pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location

**for** (i **in** 1:numPlots) {

      # Get the i,j matrix positions of the regions that contain this subplot

matchidx <- as.**data**.frame(which(layout == i, arr.ind = TRUE))

print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,

layout.pos.col = matchidx$col))

}

}

}

**Code Example #2**

######################################################################

## ##

## PROJECT TITLE: SHOW ME THE MONEY ##

## PROJECT AUTHOR: THOMAS S. ROBINSON ##

## EMAIL: THOMAS.ROBINSON@POLITICS.OX.AC.UK ##

## ##

## DESCRIPTION: CONJOINT ANALYSIS OF INITIATIVE DATA ##

## ##

######################################################################

#### Packages ####

library(tidyverse)

library(here)

library(dotwhisker)

library(miceadds)

library(broom)

library(xtable)

library(stargazer)

here()

#### Init ####

## Not run:

source(here("code","conjoint\_functions.R"))

# cj\_format()

init\_df <- read\_csv(here("data", "initiative\_formatted.csv")) %>%

relevel\_factors(.) %>%

filter(!(average == "$1 million" & total == "$100,000 to $200,000"))

issues <- unique(init\_df$issue)

#### Descriptive statistics: vote ####

vote\_desc <- init\_df %>%

filter(camp == "Support") %>%

group\_by(issue) %>%

summarise(Vote = mean(vote))

rate\_desc <- init\_df %>%

group\_by(camp, issue) %>%

summarise(approval = mean(rate)) %>%

spread(**key** = camp, value = approval)

desc\_tabl <- vote\_desc %>%

cbind(rate\_desc[,c("Oppose","Support")]) %>%

mutate(issue = ifelse(issue == "marij","Marijuana legalisation",

ifelse(issue == "enviro","Environment tax",

ifelse(issue == "bond","Bond issuance",

ifelse(issue == "wage","Wage increase",NA))))) %>%

rename(Issue = issue) %>%

xtable(., caption = 'Subject support and rating of initiative campaigns',

label = "tab:init\_support",

digits = 2) %>%

print(., **file** = "tables/init\_suppport.tex",

**include**.rownames = FALSE)

## Check the statistical difference between approval for each issue

issue\_approval\_sig <- **data**.frame(Issue = rep(NA,4),

p = rep(NA,4))

**for** (i **in** 1:4) {

j <- issues[i]

init\_df\_tmp <- init\_df %>%

filter(issue == j)

issue\_approval\_sig[i,1] <- j

issue\_approval\_sig[i,2] <- t.test(init\_df\_tmp[init\_df\_tmp$camp == "Support",]$rate,

init\_df\_tmp[init\_df\_tmp$camp == "Oppose",]$rate)$p.value

}

rm(i,j, init\_df\_tmp)

#### Campaign side-specific models ####

init\_f\_vote <- function(df) {

if (!("issue" %in% names(df))) {

cluster\_glm(formula = vote ~ average + total + largest + prop + origin,

**data** = df, type = "logit",

cluster = "participantid")

} else {

cluster\_glm(formula = vote ~ average + total + largest + prop + origin + issue,

**data** = df, type = "logit",

cluster = "participantid")

}

}

init\_vote\_issues <- init\_df %>%

group\_by(issue) %>%

nest() %>%

mutate(model = map(data,init\_f\_vote)) %>%

unnest(model) %>%

mutate(issue = capitalize(issue))

## Frequency balance tests

summary\_stats\_init <- init\_df %>%

gather(average, largest, prop, **origin**, total, **key** = "attribute", value = "level") %>%

select(attribute, **level**, issue) %>%

group\_by(issue,attribute, **level**) %>%

summarise(**freq** = n()) %>%

ungroup()

total\_attributes\_init <- summary\_stats\_init %>%

group\_by(issue, attribute) %>%

summarise(total = sum(freq)) %>%

ungroup()

summary\_stats\_table\_init <- summary\_stats\_init %>%

left\_join(total\_attributes\_init, by = c("issue", "attribute")) %>%

mutate(proportion = freq/total) %>%

select(-**freq**, -total) %>%

spread(issue, proportion)

print(xtable(summary\_stats\_table\_init,

caption = "Balance test: proportion of times each attribute-level was displayed to participants in the initiative conjoint",

label = "tab:init\_bt\_level\_prop",

digits = 2),

**file** = "tables/init\_bt\_level\_proportions.tex")