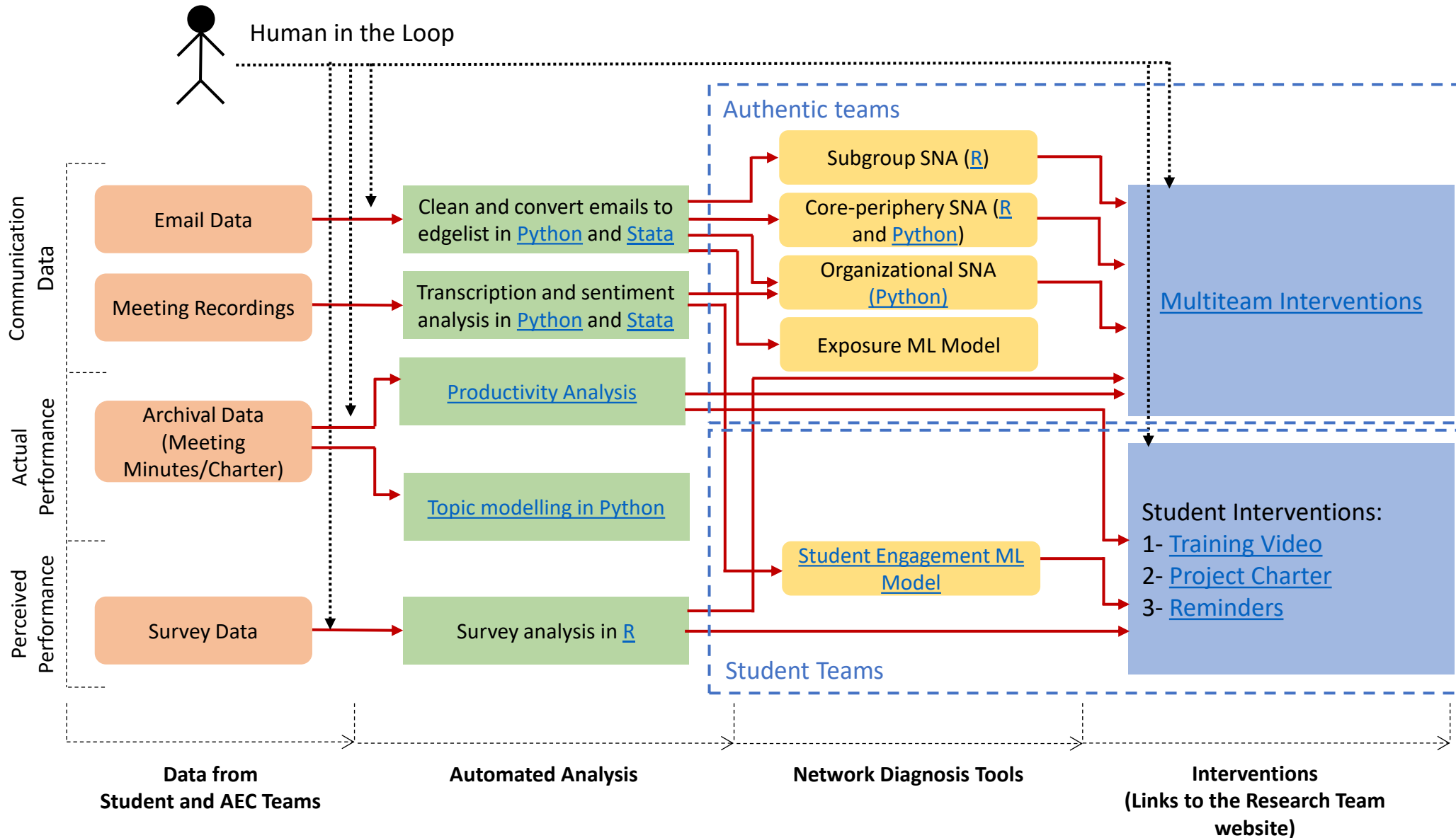


Current State of Practice Based on Our Work:

Semi-automated protocol to collect, clean, code, and process data to develop and implement interventions to small and multilevel project team systems.



This research was supported by the National Science Foundation through Grant No. 1928278. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the researchers and do not necessarily reflect the views of the National Science Foundation.

Email Cleaning (Python and Stata)

```
#List of blacklist to exclude
keywords=set()
blackemails=set()
fp4=open('Subject Line Keywords (Black)-
Table 1.csv')
fp5=open('Blacklist E-Mail IDs-Table
1.csv')
reader4=csv.reader(fp4)
reader5=csv.reader(fp5)
for row in reader4:
    keywords.add(row[0].strip())
for row in reader5:

blackemails.add(row[0].strip().lower())
fp4.close()
fp5.close()
#Dictionary for mathcing current
collected emails with rosterID
fp1=open("Sheet1-Table 1.csv")
reader1=csv.reader(fp1)
next(reader1,None)
name_id={}
mail_id={}
for row in reader1:
    if row[1]:

name_id[row[1].strip().lower()]=row[2]

mail_id[row[11].strip().lower()]=row[2]
    if row[12]:
        mid=row[12].split(';')
        for i in mid:

mail_id[i.strip().lower()]=row[2]
fp1.close()
```

The first portion of the Python code

```
/*
clear all
set more off
set linesize 110
import excel using "C:\Users\vermi\Documents\Polluted\Hanzhe
Zhang\June2023\06.21-05.22.xlsx", firstrow

save "C:\Users\vermi\Documents\Polluted\Hanzhe
Zhang\June2023\06.21-05.22.dta", replace

import excel using "C:\Users\vermi\Documents\Polluted\Hanzhe
Zhang\June2023\07.21-04.23updated.xlsx", firstrow clear
*gen int date=dofc(Date)
*format date %tdNN/DD/CCYY
destring Senderid, generate(s)
destring Receiverid, generate(r)
rename Date date
rename Subject subject
rename Time time
drop Senderid Receiverid
merge m:m date s r subject using
"C:\Users\vermi\Documents\Polluted\Hanzhe
Zhang\June2023\06.21-05.22.dta"
tab _merge
drop _merge
bys _all: drop if _n>1
sort date time
export excel using "C:\Users\vermi\Documents\Polluted\Hanzhe
Zhang\June2023\06.21-04.23.xlsx", firstrow(variables) replace
*/
```

The first portion of the Stata code

Meeting Transcription & Sentiment Analysis (Python)

```
import torch
import torch.nn as nn

class RNNModel(nn.Module):
    def __init__(self, vocab_size,
                 input_dim, output_dim, n_class, n_layer,
                 rnn_type, device):
        super(RNNModel, self).__init__()
        self.input_dim = input_dim
        self.output_dim = output_dim

        # embedding layer
        self.emb =
nn.Embedding(vocab_size, input_dim)

        # rnn layer
        assert rnn_type in ['gru',
'lstm', 'rnn']
        self.rnn_type = rnn_type
        if rnn_type == 'gru':
            self.rnn = nn.GRU(input_dim,
output_dim, num_layers=n_layer,
batch_first=True)
        elif rnn_type == 'lstm':
            self.rnn =
nn.LSTM(input_dim, output_dim,
num_layers=n_layer, batch_first=True)
        else:
            self.rnn = nn.RNN(input_dim,
output_dim, num_layers=n_layer,
batch_first=True)
```

The first portion of the Python code

	A	B	C	E	F	G
1	Start time	End time	rosterID	Speaker	Transcription	Predicted Label
2	18.08	21.77	4	1	hello in welcome	G
3	22.52	23.22	26	49	oh i see	G
4	24.5	46.76	4	1	um really seen is	G
5	47.5	47.97	26	4	yes	G
6	49.8	50.74	26	49	okay	G
7	53.38	59.84	4	1	hum chef hyping i	G
8	60.47	66.35	26	4	yeah all the storm	G
9	68.49	82.48	4	1	um interior pre ca	G
10	87.26	88.34	24	8	there was some o	G
11	88.34	99.63	26	4	we <unk> we lost	G
12	100.72	121.11	4	1	so <unk> with rya	G
13	121.11	123.33	26	49	getting through th	G
14	125.55	138.34	4	1	um slab on grade	A
15	138.56	140.39	26	4	least space and	G
16	143.67	145.71	6	53	oh i got the least	G
17	145.71	145.95	26	49	[laughter]	O
18	145.95	150.09	26	4	<unk> what	O
19	155.33	168.84	4	1	interior framing o	G
20	169.53	170.57	26	4	they agreed	G
21	171.16	207.54	4	1	reiterate the sam	G
22	208.02	217.24	26	4	and i got reaching	G
23	218.24	220.51	4	1	or like ninety five	G
24	220.56	223.75	24	8	yeah you know i r	G
25	223.75	224.47	7	22	yeah	G

Final product example

Meeting Transcription & Sentiment Analysis (Stata)

```

foreach team of numlist 5(1)31 {
import excel "_clean/FCU`team'.xlsx",
firstrow case(lower) clear

rename overallstartstamp start
rename overallendstamp end

foreach x in start end {
replace `x' = substr(`x',"H",":",.)
replace `x' = substr(`x',"M",":",.)
replace `x' = substr(`x',"Z",":",.)
}

split start, parse(:)
rename start1 h0
rename start2 m0
rename start3 s0

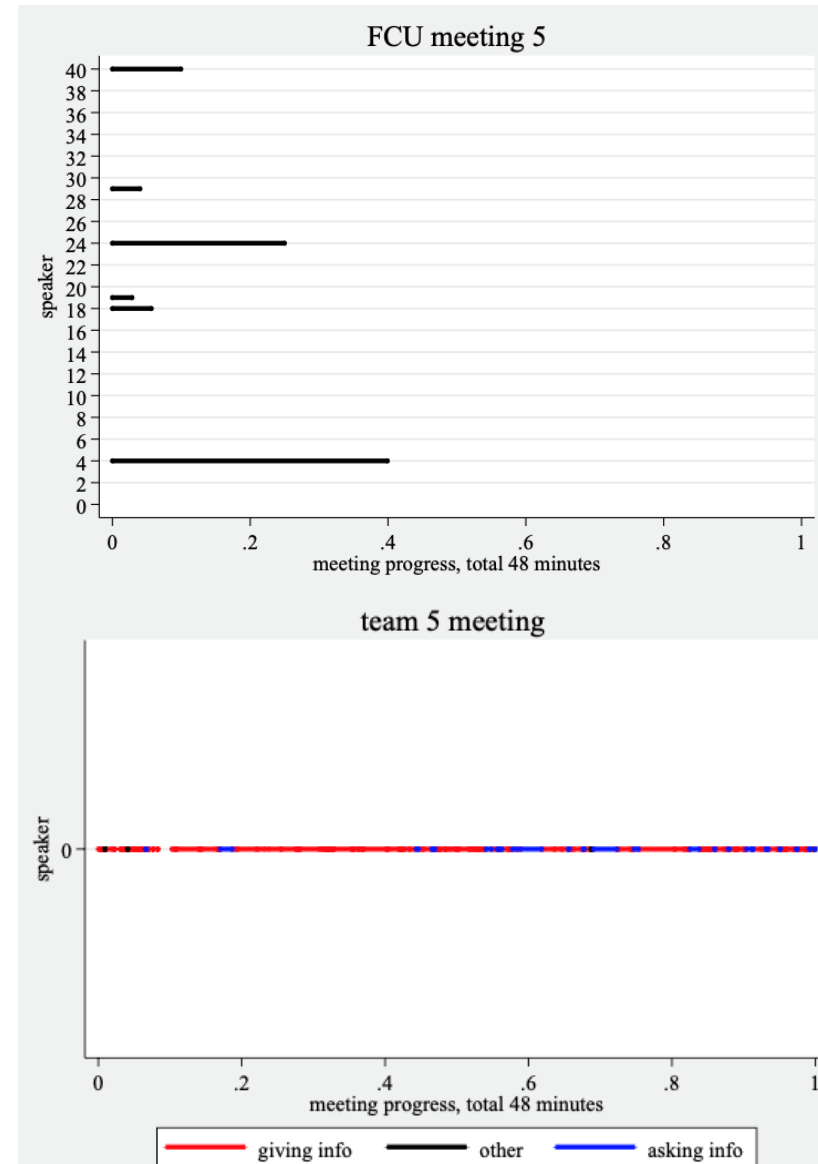
destring(h0), replace force
destring(m0), replace force
destring(s0), replace force

gen t0=3600*h0+60*m0+s0

split end, parse(:)
rename end1 h1
rename end2 m1
rename end3 s1

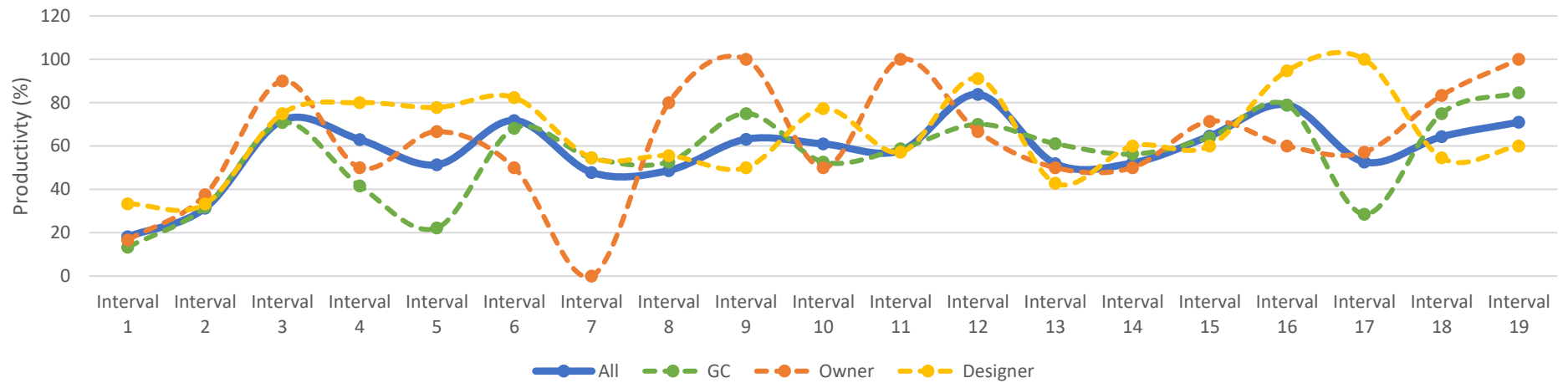
```

The first portion of the Stata code



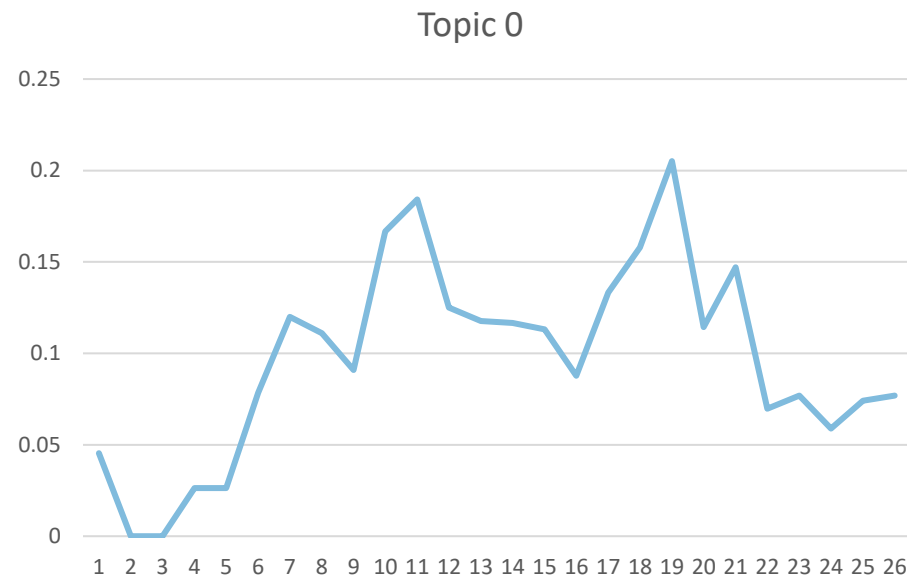
Final product example

Productivity Analysis



Productivity Analysis

Topic Modeling



Topics Discussed: Top keywords for each category

Final product example: Topic 0 Across meetings

Survey Analysis (R)

```

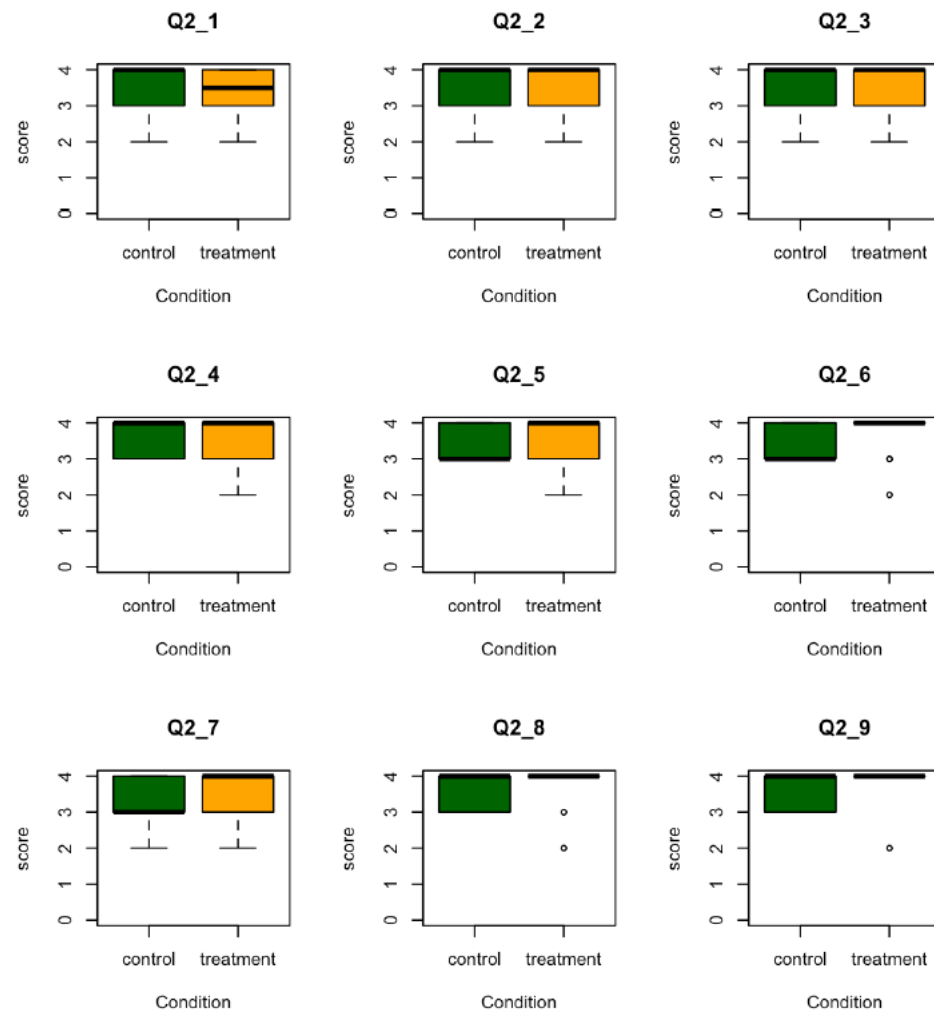
----
title: "survey and convo"
author: "Shimeñg Dāi"
date: "4/19/2022"
output: html_document
----

```{r setup, include=FALSE}
setwd("~/Desktop/Michigan State/Second
semester/RA work/QAP/Lop")
library(stringr)
library(readxl)
library(tidyverse)

```{r}
file_list <-
list.files("~/Desktop/Michigan
State/Second semester/RA work/QAP/Lop",
pattern = "\\..xlsx", full.names = TRUE)
# \\..txt means "literally" names ending
in .txt
file_list

# Read survey data
```{r}
survey 1 data <-
read_xcxl("~/Desktop/Michigan
State/Second semester/RA
work/QAP/Loop/survey/survey1_edgelist.xl
sx")
survey 2 data <-
read_xcxl("~/Desktop/Michigan
State/Second semester/RA
work/QAP/Loop/survey/survey2_edgelist.xl
sx")

```



The first portion of the R code

Final product example

# Subgroup SNA (R)

```
library(kliqfindr)
library(dplyr)
#install.packages("readr")
library(readr)

Set Working Directory
setwd("C:/Users/Jordan/Desktop/Hasan_3_3
0_22/RawEdgelist/")

Writes list file version of edge list
-- required for kliqfinder
filename<-"Edgelis_Test"
file<-paste(filename, ".csv", sep="")

imports edgelist
ties_zerosincluded <- read.csv(file,
header=T, as.is=T)

removes edges with weight 0
ties <- ties_zerosincluded %>%
filter(edge.weight != 0)

write edgelist to list file
write_delim(ties, "myedge.list", delim="
", col_names=F)
```

## The first portion of the R code



# Core-Periphery SNA (R)

```

{##with igraph

library(tidyverse)
library(readxl)
library(igraph)

emails <-
read_excel("C:/Users/moore/Desktop/School/Econ/FOW/Circle/Emails&Intervals.xlsx")

roster <-
read_excel("C:/Users/moore/Desktop/School/Econ/FOW/Roster/roster2.xlsx")
roster <- rename(roster, id = "0", Tier = "Tiers", Role = "Roles")

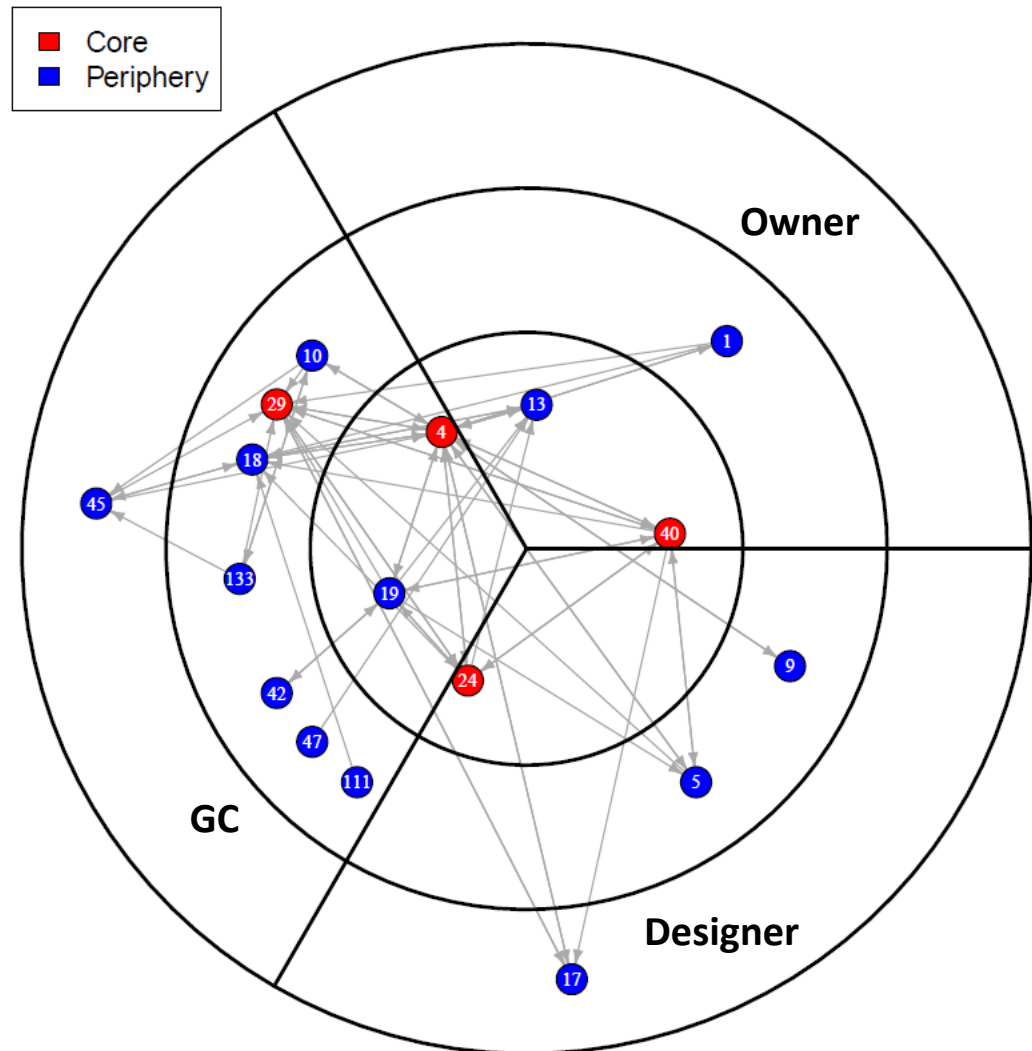
sheets <- excel_sheets(path =
"C:/Users/moore/Desktop/School/Econ/FOW/Circle/Emails&Intervals.xlsx")

emails <-
multiplesheets("C:/Users/moore/Desktop/School/Econ/FOW/Circle/Emails&Intervals.xlsx")

setwd("C:/Users/moore/Desktop/School/Econ/FOW/Circle/Intervals/")

```

The first portion of the R code



Final product example

# Core-periphery SNA (Python)

```
import numpy as np
import cpnet
import networkx as nx
import matplotlib.pyplot as plt
import io
import pandas as pd
from pathlib import Path
import os
from networkx.drawing.nx_pydot import
graphviz_layout

filepath =
r"C:\Users\Jordan\Desktop\Hasan_3_30_22\
RawEdgelist"

filename = "Edgelist_Test"

file =
os.path.join(filepath, filename+".csv")

print(file)

Convert loaded edgelist into a graph
object
Data = open(file, "r")

insert name of uploaded CSV here
next(Data, None) # skips the first line
in the input file
Graphype = nx.Graph()

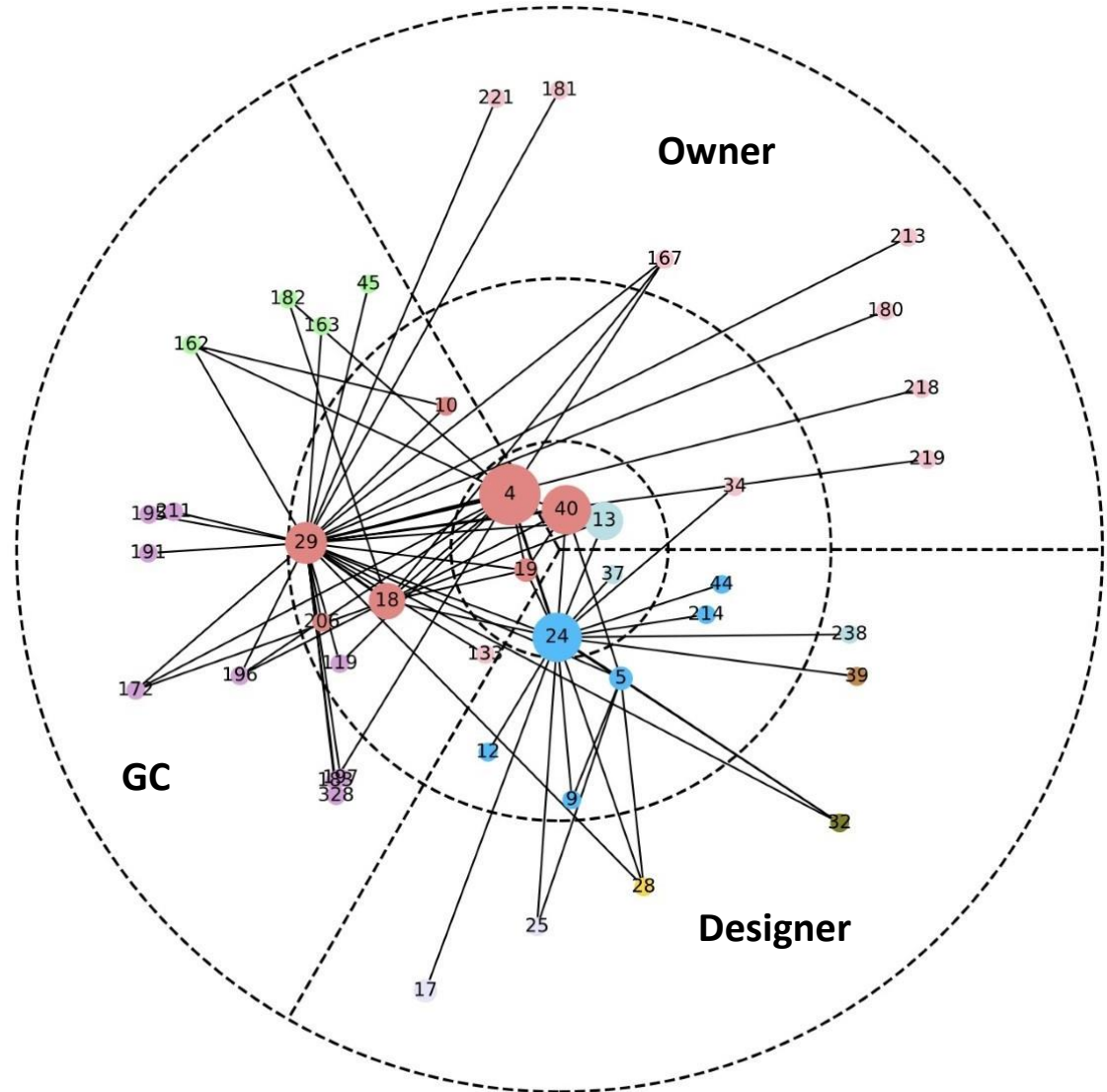
G = nx.parse_edgelist(Data,
delimiter=',', create_using=Graphype,
```

## The first portion of the R code

# Organizational SNA (Python)

```
{
 "cells": [
 {
 "cell_type": "markdown",
 "id": "vocal-charleston",
 "metadata": {},
 "source": [
 "## For New data"
]
 },
 {
 "cell_type": "code",
 "execution count": 1,
 "id": "minimal-commerce",
 "metadata": {},
 "outputs": [],
 "source": [
 "import pandas as pd\n",
 "import networkx as nx\n",
 "import matplotlib.pyplot as plt\n",
 "import numpy as np"
]
 },
 {
 "cell_type": "code",
 "execution count": 2,
 "id": "according-gates",
 "metadata": {},
 "outputs": [],
 "source": [
 "user data =
pd.read_csv('./data/Directory3.csv',
usecols=[\"ID\", \"General Group (For
PieSociogram)\", \"Tier Level\",
\"General Expertise\", 'Interval 2
Duration'], dtype=str)"
]
 }
]
}
```

The first portion of the Python code



Final product example

# Student Engagement ML Model

```
import pandas as pd
import sys

import torch
from torch_geometric.data import Data

edgelist_dir = 'data/edgelist'
engage_dir = 'data/engage'
measure_dir = 'data/measure'

teams = list('ABCDEFGHIJKMNOPQRST')

data = {}
node_dict = {}

invalid_num = 0

for team in teams:
 data[team] = []
 node_dict[team] = {}
 convo_file = edgelist_dir +
'/convo_' + team + '.xlsx'
 engage_file = engage_dir +
'/engage_' + team + '.xlsx'
 measure_file = measure_dir +
'/measure_' + team + '.xlsx'

 convo_df = pd.read_excel(convo_file)
 engage_df =
pd.read_excel(engage_file)
 measure_df =
pd.read_excel(measure_file)

 team_size =
measure_df.teamsize.values[0]
 for i in range(team_size):
```

**The first portion of the Python code**

```
import sys

import pandas as pd
import torch
from read_data import data
from torch_geometric.datasets import
TUDataset

torch.manual_seed(1234)

dataset =
TUDataset(root='data/TUDataset',
name='MUTAG')
print()
print(f'Dataset: {dataset}:')
print('=====')
print(f'Number of graphs:
{len(dataset)}')
print(f'Number of features:
{dataset.num_features}')
print(f'Number of classes:
{dataset.num_classes}')
data = dataset[0] # Get the first
graph object.
print()
print(data)
print('=====')
print('=====')
Gather some statistics about the
first graph.
print("edge index:")
data.edge_index.shape)
print("edge attr:")
data.edge_attr.shape)
print(f'Number of nodes:
{data.num_nodes}')
print(f'Number of edges:
{data.num_edges}')
print(f'Average node degree:
{data.num_edges / data.num_nodes:.2f}')
print(f'Has isolated nodes:
```

**The first portion of the Python code**