

# W203 Lab 1: Candidate Debt EDA

*Yulia and Mitch*

*January 27, 2018*

## 1. Introduction

### 1.1 Loading raw dataset

```
CandidateDebt <- read.csv("CandidateDebt.csv",
                          stringsAsFactors = FALSE)
str(CandidateDebt)
```

```
## 'data.frame': 1043 obs. of 28 variables:
## $ reportnumber : int 100495995 100496548 100498383 100495987 100496259 100496199 100496375 1
## $ origin : chr "B.3" "B.3" "B.3" "B.3" ...
## $ filerid : chr "RYU C 133" "THOMT 368" "FEY J 422" "STRAS 111" ...
## $ filertype : chr "Candidate" "Candidate" "Candidate" "Candidate" ...
## $ filename : chr "RYU CINDY S" "THOMAS TIMOTHY N JR" "FEY JACOB C" "STRACHAN STEVEN D" .
## $ firstname : chr "CINDY" "TIMOTHY" "JACOB" "STEVEN" ...
## $ middleinitial : chr "S" "N" "C" "D" ...
## $ lastname : chr "RYU" "THOMAS" "FEY" "STRACHAN" ...
## $ office : chr "STATE REPRESENTATIVE" "COUNTY COMMISSIONER" "STATE REPRESENTATIVE" "CO
## $ legislativedistrict: chr "STATE SENATOR" "STATE SENATOR" "STATE SENATOR" "STATE SENATOR" ...
## $ position : chr "1" "1" "1" "1" ...
## $ party : chr "" "" "" "" ...
## $ jurisdiction : chr "REPUBLICAN" "REPUBLICAN" "REPUBLICAN" "REPUBLICAN" ...
## $ jurisdictioncounty : chr "LEG DISTRICT 01 - SENATE" "LEG DISTRICT 01 - SENATE" "LEG DISTRICT 01 -
## $ jurisdictiontype : chr "KING" "KING" "KING" "KING" ...
## $ electionyear : chr "Legislative" "Legislative" "Legislative" "Legislative" ...
## $ amount : chr "2012" "2012" "2012" "2012" ...
## $ recordtype : chr "283.25" "283.25" "283.25" "283.25" ...
## $ fromdate : chr "DEBT" "DEBT" "DEBT" "DEBT" ...
## $ thrudate : chr "6/1/12" "6/1/12" "6/1/12" "6/1/12" ...
## $ debtdate : chr "7/16/12" "7/16/12" "7/16/12" "7/16/12" ...
## $ code : chr "7/3/12" "7/3/12" "7/3/12" "7/3/12" ...
## $ description : chr "" "" "" "" ...
## $ vendorname : chr "RE-ORDER TEE SHIRTS" "RE-ORDER TEE SHIRTS" "RE-ORDER TEE SHIRTS" "RE-O
## $ vendoraddress : chr "HICKEY GAYLE" "HICKEY GAYLE" "HICKEY GAYLE" "HICKEY GAYLE" ...
## $ vendorcity : chr "PO BOX 2749" "PO BOX 2749" "PO BOX 2749" "PO BOX 2749" ...
## $ vendorstate : chr "WOODINVILLE " "WOODINVILLE " "WOODINVILLE " "WOODINVILLE " ...
## $ vendorzip : chr "WA" "WA" "WA" "WA" ...
```

Problems with target variable *amount*:

```
table(CandidateDebt$amount)
```

```
##
## #N/A 2012
## 56 987
```

Resolution: shift column names:

```
# get column names from row data
var_names <- colnames(read.csv("CandidateDebt.csv", nrow = 1))

# insert column after "position" and remove last column
var_names_corrected <- c(var_names[1:grep("position", var_names)],
  "position2",
  var_names[(grep("position", var_names) + 1):(length(var_names) - 1)])
```

Re-loading raw data:

```
# reading the data with correct headers
CandidateDebt <- read.csv("CandidateDebt.csv",
  stringsAsFactors = FALSE,
  col.names = var_names_corrected)
rm(list = c("var_names", "var_names_corrected"))
```

Description of data set:

Blah Blah Blah

```
dim(CandidateDebt)
```

```
## [1] 1043 28
```

```
# Converting target variable to numeric
CandidateDebt$amount_num <- as.numeric(CandidateDebt$amount)
summary(CandidateDebt$amount_num)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.      NA's
##      3.24   283.25   300.00  1347.42  1210.50 19000.00      56
```

## 1.2 Exploring rows with missing debt data

```
# creating flag for missing values (1 for missing)
CandidateDebt$missing_amount <- ifelse(is.na(CandidateDebt$amount_num), 1, 0)
table(CandidateDebt$missing_amount)
```

```
##
##  0  1
## 987 56
```

While exploring 56 rows with missing data, we discovered that those rows are missing data in all columns except filer name and office they run for. Good news is we are losing only one candidate if we exclude those 56 rows from the analysis. No unique values of *office* variable are among 56 rows.

```
# number of of unique filer ids (candidates in full dataset)
length(unique(CandidateDebt$filerid))
```

```
## [1] 141
```

```
# number of unique filer ids (candidates) in data set without 56 rows with missing data:
length(unique(CandidateDebt[CandidateDebt$missing_amount == 0,]$filerid))
```

```
## [1] 140
```

```
# number of of unique values of office (candidates in full dataset)
length(unique(CandidateDebt$office))
```

```
## [1] 16
```

```
# number of unique values of office in data set without 56 rows with missing data:
length(unique(CandidateDebt[CandidateDebt$missing_amount == 0,]$office))
```

```
## [1] 16
```

```
# converting dates from character to dates
```

```
CandidateDebt$fromdate <- as.Date(CandidateDebt$fromdate, format = "%m/%d/%y")
```

```
CandidateDebt$thrudate <- as.Date(CandidateDebt$thrudate, format = "%m/%d/%y")
```

```
CandidateDebt$debtdate <- as.Date(CandidateDebt$debtdate, format = "%m/%d/%y")
```

### 1.3 Creating analytic dataset

Exclude variables:

- origin (one value = B.3)
- filertype (one value = Candidate)
- filename, firstname, middleinitial, lastname (will use filerid as a candidate identifier)
- position and position2 (values are not clear and were messed up in raw data)
- electionyear (one value = 2012)
- recordtype (one value = DEBT)

```
# creating a vector of variables to keep for analysis
```

```
keep_vars <- c("reportnumber", "filerid", "filename", "office", "legislativedistrict",
  "party", "jurisdiction", "jurisdictioncounty", "jurisdictiontype",
  "amount_num", "fromdate", "thrudate", "debtdate", "code", "description",
  "vendorname", "vendoraddress", "vendorcite", "vendorstate")
```

```
# removing 56 rows with missing data
```

```
CandidateDebtSub <- CandidateDebt[CandidateDebt$missing_amount == 0,]
```

```
CandidateDebtSub <- CandidateDebtSub[keep_vars]
```

```
rm(keep_vars)
```

Looking at main analytic dataset:

```
summary(CandidateDebtSub)
```

```
##   reportnumber      filerid      filename
##   Min.   :100346104   Length:987   Length:987
##   1st Qu.:100446276   Class :character   Class :character
##   Median :100471547   Mode  :character   Mode  :character
##   Mean    :100466089
##   3rd Qu.:100494036
##   Max.    :100599472
##   office      legislativedistrict  party
##   Length:987   Length:987          Length:987
##   Class :character   Class :character   Class :character
##   Mode  :character   Mode  :character   Mode  :character
##
##
##   jurisdiction      jurisdictioncounty jurisdictiontype
##   Length:987        Length:987          Length:987
##   Class :character   Class :character   Class :character
```

```
## Mode :character Mode :character Mode :character
##
##
##
## amount_num fromdate thrudate
## Min. : 3.24 Min. :2009-10-01 Min. :2009-10-31
## 1st Qu.: 283.25 1st Qu.:2011-10-01 1st Qu.:2011-10-31
## Median : 300.00 Median :2012-02-01 Median :2012-02-29
## Mean : 1347.42 Mean :2011-12-19 Mean :2012-01-20
## 3rd Qu.: 1210.50 3rd Qu.:2012-06-01 3rd Qu.:2012-07-16
## Max. :19000.00 Max. :2012-08-01 Max. :2012-08-31
## debtdate code description
## Min. :2008-10-29 Length:987 Length:987
## 1st Qu.:2011-07-03 Class :character Class :character
## Median :2012-02-29 Mode :character Mode :character
## Mean :2011-12-13
## 3rd Qu.:2012-07-03
## Max. :2012-08-31
## vendorname vendoraddress vendorcity
## Length:987 Length:987 Length:987
## Class :character Class :character Class :character
## Mode :character Mode :character Mode :character
##
##
##
## vendorstate
## Length:987
## Class :character
## Mode :character
##
##
##
```

```
# checking for presense of missing values
sum(is.na(CandidateDebtSub))
```

```
## [1] 0
```

## 1.4 Evaluating data quality

Calculating number of unique values per candidate for campaign related variable

```
aggr_office <- aggregate(amount_num ~ filerid + office, data = CandidateDebtSub, sum)
aggr_office <- aggregate(office ~ filerid, data = aggr_office, length)

aggr_legdis <- aggregate(amount_num ~ filerid + legislativedistrict, data = CandidateDebtSub, sum)
aggr_legdis <- aggregate(legislativedistrict ~ filerid, data = aggr_legdis, length)

aggr_party <- aggregate(amount_num ~ filerid + party, data = CandidateDebtSub, sum)
aggr_party <- aggregate(party ~ filerid, data = aggr_party, length)

aggr_jur <- aggregate(amount_num ~ filerid + jurisdiction, data = CandidateDebtSub, sum)
aggr_jur <- aggregate(jurisdiction ~ filerid, data = aggr_jur, length)

aggr_jurc <- aggregate(amount_num ~ filerid + jurisdictioncounty, data = CandidateDebtSub, sum)
```

```

aggr_jurc <- aggregate(jurisdictioncounty ~ filerid, data = aggr_jurc, length)

aggr_jurt <- aggregate(amount_num ~ filerid + jurisdictiontype, data = CandidateDebtSub, sum)
aggr_jurt <- aggregate(jurisdictiontype ~ filerid, data = aggr_jurt, length)

aggr_comb <- cbind(aggr_office,
                  aggr_legdis[,2],
                  aggr_party[,2],
                  aggr_jur[,2],
                  aggr_jurc[,2],
                  aggr_jurt[,2])

colnames(aggr_comb) <- c("filerid", "office", "legislativedistrict", "party", "jurisdiction",
                        "jurisdictioncounty", "jurisdictiontype")
rm(list = c("aggr_office", "aggr_legdis", "aggr_party", "aggr_jur", "aggr_jurc", "aggr_jurt"))

#sapply(aggr_comb[, -1], table)
summary(aggr_comb[, -1])

```

```

##      office  legislativedistrict      party      jurisdiction
##  Min.   :1  Min.   :1.000      Min.   :1.000  Min.   : 1.000
##  1st Qu.:1  1st Qu.:1.000      1st Qu.:1.000  1st Qu.: 2.000
##  Median :1  Median :3.000      Median :2.000  Median : 3.000
##  Mean   :1  Mean   :2.943      Mean   :1.836  Mean   : 4.457
##  3rd Qu.:1  3rd Qu.:4.000      3rd Qu.:2.000  3rd Qu.: 6.250
##  Max.   :1  Max.   :8.000      Max.   :3.000  Max.   :14.000
##  jurisdictioncounty jurisdictiontype
##  Min.   :1.000      Min.   :1.000
##  1st Qu.:1.000      1st Qu.:1.000
##  Median :3.000      Median :2.000
##  Mean   :2.693      Mean   :2.057
##  3rd Qu.:4.000      3rd Qu.:3.000
##  Max.   :6.000      Max.   :4.000

```

Based on the above, we thing all but *office* variables are unreliable

```

# creating flag variables for candidates with more than 1 unique value
aggr_comb$legdist_mult <- ifelse(aggr_comb$legislativedistrict > 1, 1, 0)
aggr_comb$party_mult <- ifelse(aggr_comb$party > 1, 1, 0)
aggr_comb$jur_mult <- ifelse(aggr_comb$jurisdiction > 1, 1, 0)
aggr_comb$jurc_mult <- ifelse(aggr_comb$jurisdictioncounty > 1, 1, 0)
aggr_comb$jurt_mult <- ifelse(aggr_comb$jurisdictiontype > 1, 1, 0)
aggr_comb$mult <- aggr_comb$legdist_mult + aggr_comb$party_mult + aggr_comb$jur_mult +
  aggr_comb$jurc_mult + aggr_comb$jurt_mult
table(aggr_comb$mult)

```

```

##
##  0  1  2  3  4  5
## 34  2  3  4 15 82

```

Only 34 candidates with “clean” data

```

# adding this flag variable to the main data set
CandidateDebtSub <- merge(CandidateDebtSub, aggr_comb[, c("filerid", "mult")], by = "filerid")
rm(aggr_comb)

```

```
# counting number of unique offices among those 34 candidates
length(unique(CandidateDebtSub$office[CandidateDebtSub$mult == 0]))
```

```
## [1] 9
```

```
# counting number of unique parties/offices among those 34 candidates
```

```
aggr_party <- aggregate(amount_num ~ filerid + party + office, data = CandidateDebtSub[CandidateDebtSub$mult == 0,],
table(aggr_party$office, aggr_party$party))
```

```
##
```

```
##           DEMOCRAT NON PARTISAN REPUBLICAN
## ATTORNEY GENERAL           0           0           1
## COUNTY COMMISSIONER        3           2           0
## GOVERNOR                   0           0           1
## PUBLIC UTILITY COMMISSIONER 0           0           2
## SECRETARY OF STATE          0           0           1
## STATE REPRESENTATIVE        4           2           9
## STATE SENATOR               0           0           2
## STATE SUPREME COURT JUSTICE  1           0           0
## SUPERIOR COURT JUDGE        3           1           2
```

```
rm(aggr_party)
```

Based on the above, only “State Representative” and “Superior Court Judge” had representatives of two major parties. This is suspect. Hence, we will exclude the following 5 variables from the analysis: *legislativedistrict*, *party*, *jurisdiction*, *jurisdictioncounty*, *jurisdictiontype*

## 1.5 Creating extra variables

Processing date variables

```
summary(CandidateDebtSub$debtdate)
```

```
##           Min.           1st Qu.           Median           Mean           3rd Qu.
## "2008-10-29" "2011-07-03" "2012-02-29" "2011-12-13" "2012-07-03"
##           Max.
## "2012-08-31"
```

```
summary(CandidateDebtSub$fromdate)
```

```
##           Min.           1st Qu.           Median           Mean           3rd Qu.
## "2009-10-01" "2011-10-01" "2012-02-01" "2011-12-19" "2012-06-01"
##           Max.
## "2012-08-01"
```

```
summary(CandidateDebtSub$thrudate)
```

```
##           Min.           1st Qu.           Median           Mean           3rd Qu.
## "2009-10-31" "2011-10-31" "2012-02-29" "2012-01-20" "2012-07-16"
##           Max.
## "2012-08-31"
```

Based on the above we will assume that the election was in August 2012

```
# Number of months before election the debt occurred
```

```
CandidateDebtSub$weeksindebt <-
```

```
  round(difftime(max(CandidateDebtSub$debtdate), CandidateDebtSub$debtdate, units = "weeks"))
```

```
CandidateDebtSub$monthsindebt <-
```

```

round(CandidateDebtSub$weeksindebt / 52 * 12)
CandidateDebtSub$monthsindebt <-
  as.numeric(CandidateDebtSub$monthsindebt)
# capping months at 13 months (for exploratory reasons)
CandidateDebtSub$monthsindebt_cap <-
  ifelse(CandidateDebtSub$monthsindebt > 12, 13, CandidateDebtSub$monthsindebt)
summary(CandidateDebtSub$monthsindebt)

```

```

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.000   2.000   6.000   8.583  14.000  46.000

```

```
summary(CandidateDebtSub$monthsindebt_cap)
```

```

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.00   2.00   6.00   6.73  13.00  13.00

```

Recoding debt *description* variable to make it more digestable

```

creditcard <- c("AM EX", "AMERICAN EXPRESS", "AMERICAN EXPRESS LOWES", "AMEX",
               "CITI MASTERCARD", "MASTERCARD", "VISA", "CAPITOL ONE",
               "MASTER CARD")
consulting <- c("CONSULTING", "JANUARY SERVICES", "$750 PER MONTH THROUGH OCTOBER",
               "AUGUST CONSULTING", "CONSULTING ESTIMATE", "CONSULTING/PHOTOGRAPHY",
               "CONSULTING/TRAVEL", "MAY CONSULTING SERVICES", "MONTHLY CONSULTING FEE",
               "RETAINER", "APRIL RETAINER")
swag <- c("RE-ORDER TEE SHIRTS", "BUMPER STICKERS/FLYERS", "CONSULTING/YARD SIGNS",
          "YARD SIGNS", "OFFICE SUPPLIES/ WATER FOR KICKOFF")

```

```

CandidateDebtSub$description_aggr[grepl("TREASURY", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "TREASURY"
CandidateDebtSub$description_aggr[grepl("CAMPAIGN", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "CAMPAIGN MANAGEMENT"
CandidateDebtSub$description_aggr[grepl("FUND", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "FUNDRAISING"
CandidateDebtSub$description_aggr[grepl("CARRY FORWARD", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "CARRY FORWARD"
CandidateDebtSub$description_aggr[grepl("REIMB", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "REIMBURSEMENT"
CandidateDebtSub$description_aggr[grepl("ACCOUNTING", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "ACCOUNTING"
CandidateDebtSub$description_aggr[grepl("BONUS", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "BONUS"
CandidateDebtSub$description_aggr[grepl("DESIGN", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "DESIGN/PRINT"
CandidateDebtSub$description_aggr[grepl("PRINT", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "DESIGN/PRINT"
CandidateDebtSub$description_aggr[grepl("POLLING", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "POLLING"
CandidateDebtSub$description_aggr[grepl("CREDIT", CandidateDebtSub$description, ignore.case = TRUE)] <-
  "CREDIT CARD"
CandidateDebtSub$description_aggr[CandidateDebtSub$vendorname %in% creditcard] <-
  "CREDIT CARD"
CandidateDebtSub$description_aggr[CandidateDebtSub$description %in% consulting] <-
  "CONSULTING"
CandidateDebtSub$description_aggr[CandidateDebtSub$description %in% swag] <-

```

```

"SWAG"
CandidateDebtSub$description_aggr[grepl("MAIL", CandidateDebtSub$description, ignore.case = TRUE)] <-
"MAIL"
CandidateDebtSub$description_aggr[grepl("POSTAGE", CandidateDebtSub$description, ignore.case = TRUE)] <-
"MAIL"
CandidateDebtSub$description_aggr[grepl("STAMPS", CandidateDebtSub$description, ignore.case = TRUE)] <-
"MAIL"
CandidateDebtSub$description_aggr[grepl("DATA", CandidateDebtSub$description, ignore.case = TRUE)] <-
"DATA/TECH/AD"
CandidateDebtSub$description_aggr[grepl("DISPLAY", CandidateDebtSub$description, ignore.case = TRUE)] <-
"DATA/TECH/AD"
CandidateDebtSub$description_aggr[grepl("WEB", CandidateDebtSub$description, ignore.case = TRUE)] <-
"DATA/TECH/AD"
CandidateDebtSub$description_aggr[grepl("ADVERTISEMENT", CandidateDebtSub$description, ignore.case = TRUE)] <-
"DATA/TECH/AD"
CandidateDebtSub$description_aggr[grepl("COMPUTER", CandidateDebtSub$description, ignore.case = TRUE)] <-
"DATA/TECH/AD"
CandidateDebtSub$description_aggr[is.na(CandidateDebtSub$description_aggr)] <- "OTHER"

rm(list = c("creditcard", "consulting", "swag"))
table(CandidateDebtSub$description_aggr)

```

```

##
##      ACCOUNTING      BONUS CAMPAIGN MANAGEMENT
##           79           22           10
##    CARRY FORWARD    CONSULTING    CREDIT CARD
##           17           130           42
##    DATA/TECH/AD    DESIGN/PRINT    FUNDRAISING
##           30           36           45
##           MAIL      OTHER      POLLING
##           14           24           5
##    REIMBURSEMENT    SWAG      TREASURY
##           54           261           218

```

```
#table(CandidateDebtSub$description[CandidateDebtSub$description_aggr == "OTHER"])
```

```

aggr_descr <- aggregate(amount_num ~ description_aggr, data = CandidateDebtSub, sum)
aggr_descr[order(-aggr_descr$amount_num),]

```

```

##      description_aggr amount_num
## 5      CONSULTING 706613.68
## 4    CARRY FORWARD 132400.93
## 1      ACCOUNTING 94592.75
## 14      SWAG 85218.14
## 9    FUNDRAISING 64764.36
## 15     TREASURY 56146.29
## 10      MAIL 35683.97
## 2      BONUS 35500.00
## 8    DESIGN/PRINT 31081.60
## 6    CREDIT CARD 21186.69
## 12     POLLING 20000.00
## 11      OTHER 14924.79
## 7    DATA/TECH/AD 13540.00
## 3  CAMPAIGN MANAGEMENT 11517.20
## 13    REIMBURSEMENT 6737.84

```



```
rm(aggr_descr)
```

Now we are ready to explore!

```
save(CandidateDebtSub, file = "CandidateDebtSub.RData")
```