Lab 1: Candidate Debt in Washington State

Bradley Nott, Ernesto Del Valle and Zorian Apik

9/26/2018

Abstract:

In this report, we provide an Exploratory Data Analysis (EDA) of debt amounts spent by candidates in Washington State during the 2012 election cycle. We apply data cleaning techniques to rearrange the dataset that has been erroneously entered and analyze some of the key variables, including debt amounts over the course of the year, descriptions of debt, and jurisdiction types. We also analyze key relationships between party type and spending, debt accrued based on different positions, and go in depth on some of the most frequent debt types, including consulting, travel, and t-shirts.

Structure:

1) Introduction
2) Univariate Analysis of Key Variables
3) Analysis of Key Relationships
4) Analysis of Secondary Effects
5) Conclusion

1) Introduction:

For this Exploratory Data Analysis (EDA), we are members of the campaign committee for an upcoming election in Washington State and are interested in studying debt reported by candidates in the 2012 election, in order to better advise our candidate. Note the original prompt mentions monthly 2012 voter registration data in Oregon State as our source data set, however, the actual data we received covers debt reported by candidates in Washington State for the 2012 election. Hence, we‘ve decided to focus our efforts on the latter in order to explore debt reporting together with the associated factors provided in the source data, in the context of our campaign advisory efforts.

At a high level, the columns provided attempt to either (A) identify the candidate filing the report, (B) describe the corresponding election, or (C) describe the debt incurred from `amount` (our main variable of interest as "dependent" variable) to vendor characteristics. The following sub-sections 1a) and 1b) cover a more detailed structure of the data together with our wrangling process into a clean `raw_data` table that mimics our source data as close as possible and then into a `valid_data` table that subsets `raw_data` for ease of analysis ahead without any meaningful loss of information. In `valid_data` we also impute some simple variambles as summary of certain columns in `raw_data` that we consider useful ahead in our EDA.

1a) Source data description and wrangling into clean `raw_data` table:

We start by importing our corresponding dataset contained in the `CandidateDebt.csv` file. Upon initial exploration, we noticed both blanks and explicit NAs in the form of "#N/A" in the data, with the latter as a likely result of manipulation in Excel. As a result, we explicitly specify all blanks, "NA" and "#N/A" as `NA` at import in R's language, for easier handling and exploration ahead:

```
library(data.table)
read_path <- "/Users/ernesto/Documents/MIDS Berkeley/03. W203 Stats/Lab 1/CandidateDebt.csv"
#read_path <- "/Users/apik/Documents/MIDS/Lab1/CandidateDebt.csv"
raw_data <- fread(read_path, na.strings = c(getOption("datatable.na.strings",
                                                      c("NA", "#N/A", ""))))
rm(read_path)
```

Next up, we use both the 'str()and 'summary() functions to get a sense for what our data includes variable and content-wise. We note that there are 1,043 observations in our data set with 28 variables provided for each observation. Besides our source data in `CandidateDebt.csv` we also received the PDF file `CandidateDebt.pdf`, which contains descriptions of 29 variables. Based on these descriptions, we proceed to compare each column/variable with the description provided, in order to ensure that each column (1) actually displays the intended data as described in the PDF, and (2) is in the most convenient variable type for our remaining EDA ahead (e.g. what 'stringsmake sense asfactors' instead):

```
str(raw_data)
```

```
## Classes 'data.table' and 'data.frame':   1043 obs. of  28 variables:
##  $ reportnumber       : int  100495995 100496548 100498383 100495987 100496259 100496199 100496375 10
##  $ 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" ...
##  $ filername          : 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" "COU
##  $ legislativedistrict: chr  "STATE SENATOR" "STATE SENATOR" "STATE SENATOR" "STATE SENATOR" ...
##  $ position           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ party              : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ 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             : int  2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...
##  $ recordtype         : num  283 283 283 283 283 ...
##  $ 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  NA NA NA NA ...
##  $ 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" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
#summary(raw_data)  # Saving some space, for now think `str()` accomplishes what we want
```

Big picture columns fall into one of three categories where (A) they attempt to describe or identify the candidate from `reportnumber` through `lastname`, (B) they attempt to describe the election from `office` through `electionyear`, or (C) they attempt to provide characteristics of the debt incurred from `amount` itself through `vendorzip`.

The referenced PDF file lists 29 column descriptions, while the above summaries show 28 columns in place,

which point to some form of initial mismatch. After close inspection, it appears that the `id` column referenced in the PDF does not appear in the source data. Besides this, while some of the early candidate identification columns from `reportnumber` through `lastname` do appear to contain the data as intended and described, other columns detailing election or debt characteristics from `office` onwards appear to have shifted data across columns. As a result, we must assess each of the 28 columns present in the data set individually with (1) actual contents and (2) data types in mind. We do this to generate a "clean" file we name `raw_data`, a more reasonable starting point for analysis:

[1] `reportnumber` => (1) column indeed contains integer IDs for each individual form as described in PDF (see uniqueness test below) (2) leave as `int` variable type, appropriate for IDs

```
# Testing uniqueness of `reportnumber`:
length(unique(raw_data$reportnumber))==length(raw_data$reportnumber)
```

```
## [1] TRUE
```

[2] `origin` => (1) column shows same "B.3" key for all entries (2) turn `chr` variable type into a `factor` given reasonably finite distinct entries

```
# Testing distinct entries in `origin`:
summary(as.factor(raw_data$origin))
```

```
##  B.3
## 1043
```

```
# Since all entries the same turn into a `factor` data type:
raw_data$origin <- as.factor(raw_data$origin)
```

[3] `filerid` => (1) column does contain what appear to be unique IDs assigned to a candidate based off the candidate's name. There are 141 unique IDs/candidates if based off this metric. As a quick verification, we compared to unique concatenation of first/middle/lastname which showed 134 unique names. This could be an early indication of same name in some cases with different candidate IDs (unless IDs were tampered with), which means we may want to work with candidate IDs instead of names for candidate identification purposes (2) turn `chr` variable type into a `factor` given reasonably finite distinct entries, may help with candidate-level exploration ahead too

```
# Assessing how many unique `filerid` exist:
length(unique(raw_data$filerid))
```

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

```
# Comparing to first/middle/lastname concatenation as closest proxy:
length(unique(paste0(raw_data$firstname, raw_data$middleinitial, raw_data$lastname)))
```

```
## [1] 134
```

```
# Since entries reasonably finite and distinct turn into a `factor` data type:
raw_data$filerid <- as.factor(raw_data$filerid)
```

[4] `filertype` => (1) column shows same "Candidate" response for all entries (2) turn `chr` variable type into a `factor` given reasonably finite distinct entries

```
# Testing distinct entries in `filertype`:
summary(as.factor(raw_data$filertype))
```

```
## Candidate
##      1043
```

```
# Since all entries the same turn into a `factor` data type:
raw_data$filertype <- as.factor(raw_data$filertype)
```

[5] `filername` => (1) column indeed contains candidate names that appear to be a concatenation of other first/middle/lastname columns in "lastname firstname middleinitial" format. We test for uniqueness and whether it's "completely redundant" with the other three columns. There are 134 unique `filername` entries per our test which coincides with the concatenation of first/middle/lastname done in [3] under `filerid` (2) similar to `filerid`, turn `chr` variable type into a `factor` given reasonably finite distinct entries, may help with candidate-level exploration ahead

```
# Assessing how many unique `filername` exist:
length(unique(raw_data$filername))
```

```
## [1] 134
```

```
# Quick test of redundancy with other first/middle/lastname columns if put in same observed format:
sum(raw_data$filername==paste(raw_data$lastname, raw_data$firstname, raw_data$middleinitial))
```

```
## [1] 945
```

```
# Since entries reasonably finite and distinct turn into a `factor` data type:
raw_data$filername <- as.factor(raw_data$filername)
```

[6] `firstname` => (1) column indeed contains candidate first names as described in PDF (2) leave as `char` given different candidates can have the same first name and a `factor` would lose information lumping these together erroneously

[7] `middleinitial` => (1) column indeed contains middle initials as described in PDF (2) leave as `char` given different candidates can have the same middle initial, same as `firstname`

[8] `lastname` => (1) column indeed contains last names as described in PDF (2) leave as `char` given different candidates can have the same last name, same as `firstname`

[9] `office` => (1) this is where columns matching headers per descriptions in PDF actually gets interesting. The columns with headers `office`, `legislativedistrict` and `position` appear to have swapped data resulting in mismatches between header names and the actual data in the column. Upon further inspection through the below code we conclude that (A) `office` actually contains the `position` data, (B) that `legislativedistrict` actually contains the `office` data, and (C) that `position` actually contains the `legislativedistrict` data. We therefore go ahead and swap these accordingly to match the order in the PDF and for data to correspond to its respective headers (2) post-swap, turn `char` variable type into a `factor` given distinct and reasonable finite entries for `office`. We do this for all three columns, given all of them satisfy these conditions for conversion into `factor`:

```
# Summarizing original source contents in `office`, `legislativedistrict` and `position`:
summary(as.factor(raw_data$office))
```

```
##         APPEALS COURT JUDGE            ATTORNEY GENERAL
##                           4                          36
##             COUNTY ASSESSOR         COUNTY COMMISSIONER
##                          19                          72
##       COUNTY COUNCIL MEMBER              COUNTY SHERIFF
##                          15                          17
##                    GOVERNOR   PUBLIC LANDS COMMISSIONER
##                          48                          38
## PUBLIC UTILITY COMMISSIONER          SECRETARY OF STATE
##                           8                          11
##               STATE AUDITOR        STATE REPRESENTATIVE
##                           7                         535
##               STATE SENATOR STATE SUPREME COURT JUSTICE
##                         144                          28
##             STATE TREASURER        SUPERIOR COURT JUDGE
##                          28                          33
```

```
summary(as.factor(raw_data$legislativedistrict))
```

```
##            ATTORNEY GENERAL             COUNTY ASSESSOR
##                          49                           2
##         COUNTY COMMISSIONER       COUNTY COUNCIL MEMBER
##                          44                           3
##              COUNTY SHERIFF                    GOVERNOR
##                           8                         101
##   PUBLIC LANDS COMMISSIONER PUBLIC UTILITY COMMISSIONER
##                          21                           2
##               STATE AUDITOR        STATE REPRESENTATIVE
##                           4                         384
##               STATE SENATOR STATE SUPREME COURT JUSTICE
##                         305                          12
##             STATE TREASURER        SUPERIOR COURT JUDGE
##                          28                          24
##                       NA's
##                          56
```

```
summary(as.factor(raw_data$position))
```

```
##    1    3    5    8   11   14   21   22   23   24   25   26   27   28   29
##  246    8    3    8   43    2   55    1    3    5    4   37    6   21    2
##   32   34   36   37   40   41   43   44   45   46   47   48 NA's
##    6    7    9    1   22   10   72    1   68   18   13   18  354
```

```
# Swapping in order to match PDF descriptions:
temp_store <- raw_data$office
raw_data$office <- raw_data$legislativedistrict
raw_data$legislativedistrict <- raw_data$position
raw_data$position <- temp_store
rm(temp_store)
# Checking if we got desired results with no data loss post-swap:
summary(as.factor(raw_data$office))
```

```
##            ATTORNEY GENERAL             COUNTY ASSESSOR
##                          49                           2
##         COUNTY COMMISSIONER       COUNTY COUNCIL MEMBER
##                          44                           3
##              COUNTY SHERIFF                    GOVERNOR
##                           8                         101
##   PUBLIC LANDS COMMISSIONER PUBLIC UTILITY COMMISSIONER
##                          21                           2
##               STATE AUDITOR        STATE REPRESENTATIVE
##                           4                         384
##               STATE SENATOR STATE SUPREME COURT JUSTICE
##                         305                          12
##             STATE TREASURER        SUPERIOR COURT JUDGE
##                          28                          24
##                       NA's
##                          56
```

```
summary(as.factor(raw_data$legislativedistrict))
```

```
##    1    3    5    8   11   14   21   22   23   24   25   26   27   28   29
##  246    8    3    8   43    2   55    1    3    5    4   37    6   21    2
```

```
##   32   34   36   37   40   41   43   44   45   46   47   48 NA's
##    6    7    9    1   22   10   72    1   68   18   13   18  354
```

```
summary(as.factor(raw_data$position))
```

```
##          APPEALS COURT JUDGE                    ATTORNEY GENERAL
##                            4                                  36
##              COUNTY ASSESSOR                  COUNTY COMMISSIONER
##                           19                                  72
##        COUNTY COUNCIL MEMBER                      COUNTY SHERIFF
##                           15                                  17
##                     GOVERNOR           PUBLIC LANDS COMMISSIONER
##                           48                                  38
## PUBLIC UTILITY COMMISSIONER                   SECRETARY OF STATE
##                            8                                  11
##               STATE AUDITOR                 STATE REPRESENTATIVE
##                            7                                 535
##               STATE SENATOR          STATE SUPREME COURT JUSTICE
##                          144                                  28
##              STATE TREASURER                 SUPERIOR COURT JUDGE
##                           28                                  33
```

```
# Post-swap, turn all three columns into factors, given reasonably finite and distinct outcomes:
raw_data$office <- as.factor(raw_data$office)
raw_data$legislativedistrict <- as.factor(raw_data$legislativedistrict)
raw_data$position <- as.factor(raw_data$position)
```

[10] `legislativedistrict` => (1) see details in [9] `office` above, column now contains what PDF describes as there are ~48 legislative districts in Washington. There are 354 unpopulated districts marked as NA (2) see details in [9] `office` above, already converted from `char` to `factor`

[11] `position` => (1) see details in [9] `office` above, column now contains what PDF describes with 16 positions corresponding to 14 identified offices in `office` (2) see details in [9] `office` above, already converted from `char` to `factor`

[12] `party` => (1) column does not contain what header implies, but rather integers with sporadic "1s" and "2s" for at least ~75% of the populated data (3rd quantile) and with 574 blank or missing entries, which is more than half of our dataset with 1,043 rows. Additionally, the party data appears to be populated in the following column titled `jurisdiction`. This one-column offset appears to prevail up to the end of our columns, with the second-to-last header `vendorstate`'s data included under the last column titled `vendorzip`. As such, we go ahead and solve this by doing the following (A) keep headers to match variable order provided in PDF as this will avoid confusion, (B) store this apparently random data currently under `party` as a new column titled `random` at the end of our data table, (C) shift all data but not headers left to match headers, (D) this implies the last column's data, titled `vendorzip` is missing, so we assign `NA` values to it for now. This allows us to not lose information, just reordeing what is already present, and to continue evaluating our data table entries under a more reasonable starting point (i.e. a data order that matches headers more closely) (2) once adjusted per the above, we convert variable type from `char` to `factor` as well, given finite existence of political parties and the (clear) distinction among these

```
# What `party` contains at import:
summary(as.factor(raw_data$party))
```

```
##    1    2    3    4    8    9   12   19   40 NA's
##  105  324    2    4    3    9    1   16    5  574
```

```
# Showing offset, how `jurisdiction` contains what `party` should and at end `vendorzip` (last column) i
summary(as.factor(raw_data$jurisdiction))
```

```
##     DEMOCRAT  INDEPENDENT NON PARTISAN   REPUBLICAN         NA's
##          638            2           48          299           56
```

```
summary(as.factor(raw_data$vendorzip))
```

```
##   CA   DC   TX   WA NA's
##   10  100    5  847   81
```

```
# Fixing offset, preserving data currently under `party` as `random` at the end of our table, and turni
random <- raw_data$party
raw_data <- cbind(raw_data, random)
raw_data[, 12:27] <- raw_data[, 13:28]
raw_data$vendorzip <- NA
rm(random)
# Converting `party` into a `factor` post rearrangement described above:
raw_data$party <- as.factor(raw_data$party)
```

[13:15] `jurisdiction`, `jurisdictioncounty` and `jurisdictiontype` => (1) post rearrangement in [12] `party`, columns now contain what PDF describes. There are 52 distinct jurisdictions, 15 jurisdiction counties and 5 jurisdiction types, respectively (2) turn all three `chr` variables into `factor` too given distinct finite set of inputs

```
# Assessing how many unique `jurisdiction`, `jurisdictioncounty` and `jurisdictiontype` exist:
length(unique(raw_data$jurisdiction))
```

```
## [1] 52
```

```
length(unique(raw_data$jurisdictioncounty))
```

```
## [1] 15
```

```
length(unique(raw_data$jurisdictiontype))
```

```
## [1] 5
```

```
# Since entries reasonably finite and distinct turn into a `factor` data type:
raw_data$jurisdiction <- as.factor(raw_data$jurisdiction)
raw_data$jurisdictioncounty <- as.factor(raw_data$jurisdictioncounty)
raw_data$jurisdictiontype <- as.factor(raw_data$jurisdictiontype)
```

[16] `electionyear` => (1) same as others, post rearrangement in [12] `party`, column now contains what PDF describes, all same 2012 year or NA. If we want to restrict to 2012 we may have to omit NA entries ahead (2) turn `chr` variable type into a `factor` given all 2012 or NA

```
# Assessing how many unique `electionyear` exist:
length(unique(raw_data$electionyear))
```

```
## [1] 2
```

```
# Since entries reasonably finite and distinct turn into a `factor` data type:
raw_data$electionyear <- as.factor(raw_data$electionyear)
```

[17] `amount` => (1) same as others, post rearrangement in [12] `party`, column now contains what PDF describes, debt incurred on orders placed (2) interestingly enough there's a reasonably finite set of entries here too even though we would have expected it to be more continuous. For instance, the $283.25 entry occurs 241 times. However, we'll keep as `num` data type for analysis

```
# Assessing how many unique `amount` exist:
length(unique(raw_data$amount))
```

```
## [1] 124
```

```
# View of amounts present (save space by just showing most often one's frequency):
#summary(as.factor(raw_data$amount))
sum(raw_data$amount==283.25, na.rm=TRUE)
```

```
## [1] 241
```

[18] `recordtype` => (1) same as others, post rearrangement in [12] `party`, column now contains what PDF describes, all "DEBT" or NA, where NAs may coincide with those seen under `electionyear` actually (2) turn `chr` variable type into a `factor` given all "DEBT or NA

```
# Assessing how many unique `recordtype` exist:
length(unique(raw_data$recordtype))
```

```
## [1] 2
```

```
# Since entries reasonably finite and distinct turn into a `factor` data type:
raw_data$recordtype <- as.factor(raw_data$recordtype)
```

[19:21] `fromdate`, `thrudate` and `debtdate` => (1) same as others, post rearrangement in [12] `party`, these columns now contain what PDF describes. Data shows reporting start date, reporting end date and debt incurrence date, respectively (2) turn `chr` variable types into explicit date formats as `POSIXct`

```
# Turning columns into actual date formats, assign "1/1/71" to NA first for easier conversion (uniform
raw_data$fromdate[is.na(raw_data$fromdate)] <- "1/1/71"
raw_data$fromdate <- as.POSIXct(strptime(raw_data$fromdate[grepl(".*/.*/.*",
  raw_data$fromdate)==TRUE], "%m/%d/%y"), format = "%Y-%m-%d")
raw_data$fromdate[year(raw_data$fromdate)==1971] <- NA
raw_data$thrudate[is.na(raw_data$thrudate)] <- "1/1/71"
raw_data$thrudate <- as.POSIXct(strptime(raw_data$thrudate[grepl(".*/.*/.*",
  raw_data$thrudate)==TRUE], "%m/%d/%y"), format = "%Y-%m-%d")
raw_data$thrudate[year(raw_data$thrudate)==1971] <- NA
raw_data$debtdate[is.na(raw_data$debtdate)] <- "1/1/71"
raw_data$debtdate <- as.POSIXct(strptime(raw_data$debtdate[grepl(".*/.*/.*",
  raw_data$debtdate)==TRUE], "%m/%d/%y"), format = "%Y-%m-%d")
raw_data$debtdate[year(raw_data$debtdate)==1971] <- NA
```

[22] `code` => (1) same as others, post rearrangement in [12] `party`, column now contains what PDF describes, type of debt (2) turn `chr` variable into `factor` given only three debt types. About 2/3 of this column (666 entries) are NA

```
# View of debt types present under `code`:
summary(as.factor(raw_data$code))
```

```
##            Fundraising    Management Services Operation and Overhead
##                      5                     10                    362
##                   NA's
##                    666
```

```
# Since entries reasonably finite and distinct turn into a `factor` data type:
raw_data$code <- as.factor(raw_data$code)
```

[23] `description` => (1) same, post rearrangement in [12] `party`, column now contains what PDF describes, debt description (2) turn `chr` variable into a `factor` given there are 106 different descriptions. 241 entires, matching the repreated amount $283.25 correspond to "RE-ORDER TEE SHIRTS"

```
# Counting debt descriptions present under `description`:
length(levels(as.factor(raw_data$description)))
```

```
## [1] 105
```

```
# Since entries reasonably finite and distinct turn into a `factor` data type, interesting top descript
raw_data$description <- as.factor(raw_data$description)
```

[24:27] `vendorname`, `vendoraddress`, `vendorcity` and `vendorstate` => (1) same as others, post rearrangement in [12] `party`, columns now contain what PDF describes. These columns now contain vendors' names, addresses, cities and states, respectively. There's 75 different vendors under 79 distinct addresses distributed into 29 cities in 4 states (excluding NAs). For `vendorstate` we would have expected all or most as "WA". However, we actually encountered some entries for "CA", "DC" (wrong Washington!) and "TX" besides some NAs. We may want to adjust for this ahead if we're restricting the analysis to the fabulous Washington State. We will later impute an `out_of_state` column with this in mind (2) only so many entries for each of these, so turn `chr` types into `factor` too

```
# Counting vendor names and addresses present under `vendorname` and `vendoraddress`:
length(levels(as.factor(raw_data$vendorname)))
```

```
## [1] 75
```

```
length(levels(as.factor(raw_data$vendoraddress)))
```

```
## [1] 78
```

```
# View of vendor cities and states present under `vendorcity` and `vendorstate`:
summary(as.factor(raw_data$vendorcity))
```

```
## BAINBRIDGE ISLAND           BELFAIR  CITY OF INDUSTRY            DALLAS
##                 3                13                 1                 5
##     FRIDAY HARBOR       GIG HARBOR          ISSAQUAH         KENNEWICK
##                 5                 5                 1                 2
##          KIRKLAND     MERCER ISLAND     MOUNTAIN VIEW        OAK HARBOR
##                38                 1                 3                 6
##           OLYMPIA      PORT ORCHARD     PORT TOWNSEND          PUYALLUP
##                20                10                 1                 5
##        SACRAMENTO          SAN JOSE      SANTA MONICA           SEATTLE
##                 3                 1                 3               452
##            SEQUIM           SHELTON           SPOKANE    SPOKANE VALLEY
##                 3                 2                 2                 1
##            TACOMA          TUMWATER  UNIVERSITY PLACE        WASHINGTON
##                24                 1                11               100
##       WOODINVILLE              NA's
##               241                80
```

```
summary(as.factor(raw_data$vendorstate))
```

```
##   CA   DC   TX   WA NA's
##   10  100    5  847   81
```

```
# Since entries reasonably finite and distinct for all turn them into `factor` data:
raw_data$vendorname <- as.factor(raw_data$vendorname)
raw_data$vendoraddress <- as.factor(raw_data$vendoraddress)
raw_data$vendorcity <- as.factor(raw_data$vendorcity)
raw_data$vendorstate <- as.factor(raw_data$vendorstate)
```

[28] `vendorzip` => (1) this piece of data appears absent from the original data set as nothing resembled zip codes. Hence, in the re-arranging done under [12] `party` above we decided to turn it into NAs (2) nothing to transform variable-wise then. Would be `int` or `factor` if present

We're done cleaning our source data as `raw_data`. To conclude this section, we just verify we've obtained the desired formats for all columns by calling the `str()` function again and then discuss how to avoid redundancy,

subsetting `raw_data` into `valid_data` ahead:

```
str(raw_data)
```

```
## Classes 'data.table' and 'data.frame':   1043 obs. of  29 variables:
##  $ reportnumber       : int  100495995 100496548 100498383 100495987 100496259 100496199 100496375 10
##  $ origin             : Factor w/ 1 level "B.3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ filerid            : Factor w/ 141 levels "ASHAK  359","BILLA2 203",..: 110 129 30 122 56 105 93 8
##  $ filertype          : Factor w/ 1 level "Candidate": 1 1 1 1 1 1 1 1 1 1 ...
##  $ filername          : Factor w/ 134 levels "ASHABRANER KARIN L",..: 105 124 31 117 56 99 86 82 70 1
##  $ firstname          : chr  "CINDY" "TIMOTHY" "JACOB" "STEVEN" ...
##  $ middleinitial      : chr  "S" "N" "C" "D" ...
##  $ lastname           : chr  "RYU" "THOMAS" "FEY" "STRACHAN" ...
##  $ office             : Factor w/ 14 levels "ATTORNEY GENERAL",..: 11 11 11 11 11 11 11 11 11 11 ...
##  $ legislativedistrict: Factor w/ 27 levels "1","3","5","8",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ position           : Factor w/ 16 levels "APPEALS COURT JUDGE",..: 12 4 12 6 7 12 4 12 12 12 ...
##  $ party              : Factor w/ 4 levels "DEMOCRAT","INDEPENDENT",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ jurisdiction       : Factor w/ 51 levels "ATTORNEY GENERAL, OFFICE OF",..: 10 10 10 10 10 10 10 10
##  $ jurisdictioncounty : Factor w/ 14 levels "BENTON","CLALLAM",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ jurisdictiontype   : Factor w/ 4 levels "Judicial","Legislative",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ electionyear       : Factor w/ 1 level "2012": 1 1 1 1 1 1 1 1 1 1 ...
##  $ amount             : num  283 283 283 283 283 ...
##  $ recordtype         : Factor w/ 1 level "DEBT": 1 1 1 1 1 1 1 1 1 1 ...
##  $ fromdate           : POSIXct, format: "2012-06-01" "2012-06-01" ...
##  $ thrudate           : POSIXct, format: "2012-07-16" "2012-07-16" ...
##  $ debtdate           : POSIXct, format: "2012-07-03" "2012-07-03" ...
##  $ code               : Factor w/ 3 levels "Fundraising",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ description        : Factor w/ 105 levels "$750 PER MONTH THROUGH OCTOBER",..: 72 72 72 72 72 72 7
##  $ vendorname         : Factor w/ 75 levels "ABBOT TAYLOR",..: 26 26 26 26 26 26 26 26 26 26 ...
##  $ vendoraddress      : Factor w/ 78 levels "10 SABLE COURT",..: 67 67 67 67 67 67 67 67 67 67 ...
##  $ vendorcity         : Factor w/ 29 levels "BAINBRIDGE ISLAND",..: 29 29 29 29 29 29 29 29 29 29 ..
##  $ vendorstate        : Factor w/ 4 levels "CA","DC","TX",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ vendorzip          : chr  NA NA NA NA ...
##  $ random             : int  NA NA NA NA NA NA NA NA NA NA ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

1b) Subsetting `raw_data` into `valid_data`, imputing some potentially useful columns:

Now that we have arrived at a clean `raw_data` data table, we can focus on segmenting into the most useful columns without losing relevant information for our exploratory analysis. For instance, `filerid` is a strong representation of all `filername`, `firstname`, `middleinitial` and `lastname` given it derives from those columns but as an ID column we can be more certain of its uniqueness. We deem it as the best candidate identifier as there's 141 unique values in it vs. the concatenation of names at 134. At the very least we are preserving the same categories. Another area of simplification across all types of columns (whether candidate, election or debt describing) relates to columns where we have the same entry in the entire column, such as `electionyear`==“2012”, `origin`==“B.3”, `filertype`==“Candidate” and `recordtype`==“DEBT”. Finally, there's 56 entries/rows that have most data deleted appearing as NAs. We don't think deleting these rows in our dataset with 1,043 entries otherwise (987 post deletion) impacts EDA results, so we go ahead and omit those in `valid_data`. The following code segments `raw_data` into `valid_data` based on these guidelines:

```
# Subsetting into columns of interest that avoid redundancy:
valid_data <- subset(raw_data, select=c("reportnumber", "filerid", "office",
  "legislativedistrict", "position", "party", "jurisdictiontype", "amount",
  "debtdate", "code", "description", "vendorname", "vendorstate"))
```

```r
# Deleting 56 rows containing invalid data as NAs across several fields
#   (including `amount`, our key variable of interest):
valid_data <- valid_data[!is.na(valid_data$amount)]
```

After the aforementioned adjustment we selected 13 columns from the original 29 in `raw_data` once clean (or 28 before any wrangling) without any meaningful loss of information. Before proceeding with our EDA, we consider it helpful to summarize some of these columns in a different way, that is, to impute some summary columns at this point. For instance, we consider it helpful to impute a `debtmonth` and `debtyear` columns off `debtdate` in order to study seasonality across the year of spending by candidates and also the build up (or whatever pattern) of spending into November 2012 as a mini time-series. We also consider it helpful to impute a `t_shirts` logical column given 241 entries appear related to that same item under the same vendor and with the same $283.25 amount. Last, we noticed some vendors are located in other states. Hence, we also want to impute a binary column indicating or telling apart in-state from out-of-state spending, naming it as `out_of_state` of type logical:

```r
# Imputing columns of interest as summaries of existing columns in subset:
debtmonth <- as.factor(as.numeric(month(valid_data$debtdate)))
debtyear <- as.factor(as.numeric(year(valid_data$debtdate)))
t_shirts <- as.logical(ifelse(valid_data$vendorname=="HICKEY GAYLE", 1, 0))
out_of_state <- as.logical(ifelse(valid_data$vendorstate=="WA", 0, 1))

# Inserting imputed columns into `valid_data` and removing object/cleaning up:
valid_data <- cbind(valid_data[, 1:9], debtmonth, debtyear, valid_data[, 10:12],
                    t_shirts, valid_data[, 13], out_of_state)
rm(debtmonth, debtyear, t_shirts, out_of_state)

# Summarizing latest structure of `valid_data` to double-check it is our desired
#   data table for EDA:
str(valid_data)
```

```
## Classes 'data.table' and 'data.frame':   987 obs. of  17 variables:
##  $ reportnumber       : int  100495995 100496548 100498383 100495987 100496259 100496199 100496375 10
##  $ filerid            : Factor w/ 141 levels "ASHAK  359","BILLA2 203",..: 110 129 30 122 56 105 93 8
##  $ office             : Factor w/ 14 levels "ATTORNEY GENERAL",..: 11 11 11 11 11 11 11 11 11 11 ...
##  $ legislativedistrict: Factor w/ 27 levels "1","3","5","8",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ position           : Factor w/ 16 levels "APPEALS COURT JUDGE",..: 12 4 12 6 7 12 4 12 12 12 ...
##  $ party              : Factor w/ 4 levels "DEMOCRAT","INDEPENDENT",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ jurisdictiontype   : Factor w/ 4 levels "Judicial","Legislative",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ amount             : num  283 283 283 283 283 ...
##  $ debtdate           : POSIXct, format: "2012-07-03" "2012-07-03" ...
##  $ debtmonth          : Factor w/ 12 levels "1","2","3","4",..: 7 7 7 7 7 7 7 7 7 7 ...
##  $ debtyear           : Factor w/ 5 levels "2008","2009",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ code               : Factor w/ 3 levels "Fundraising",..: NA NA NA NA NA NA NA NA NA NA ...
##  $ description        : Factor w/ 105 levels "$750 PER MONTH THROUGH OCTOBER",..: 72 72 72 72 72 72 7
##  $ vendorname         : Factor w/ 75 levels "ABBOT TAYLOR",..: 26 26 26 26 26 26 26 26 26 26 ...
##  $ t_shirts           : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ vendorstate        : Factor w/ 4 levels "CA","DC","TX",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ out_of_state       : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

2) Univariate Analysis of Key Variables

2a) Amount:

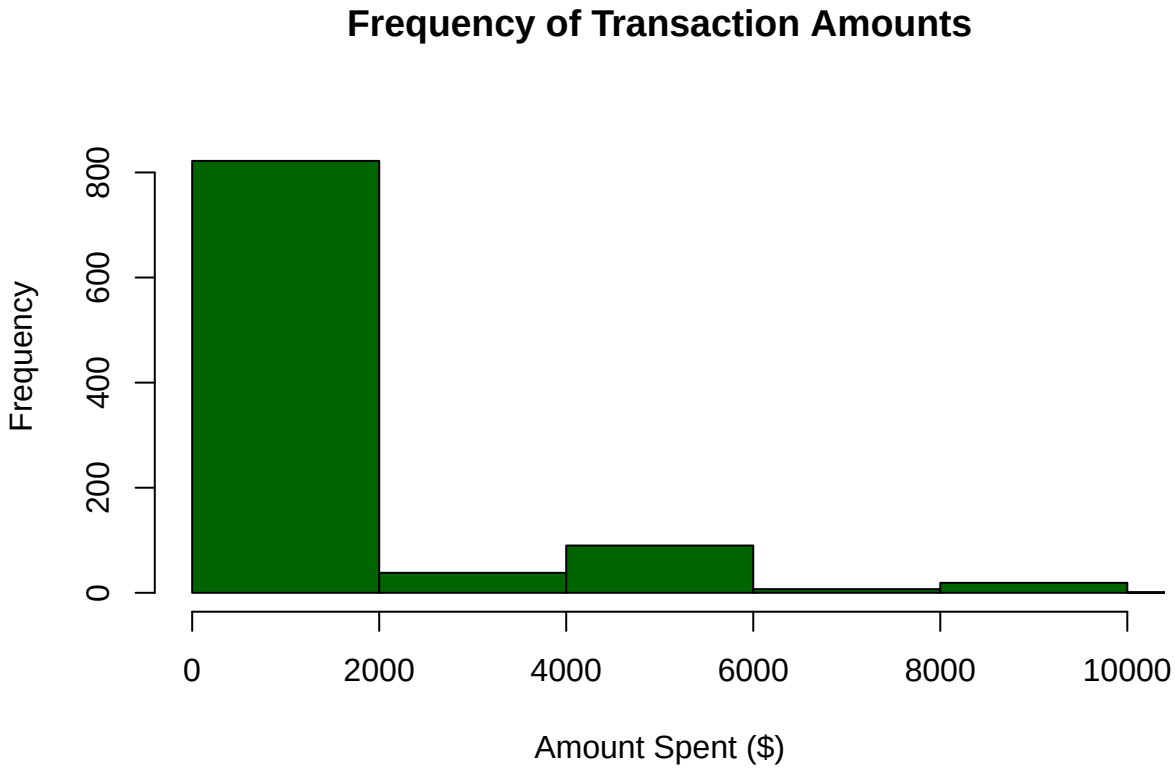
We observe first the `amount` debt recordings for the 2012 candidates:

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

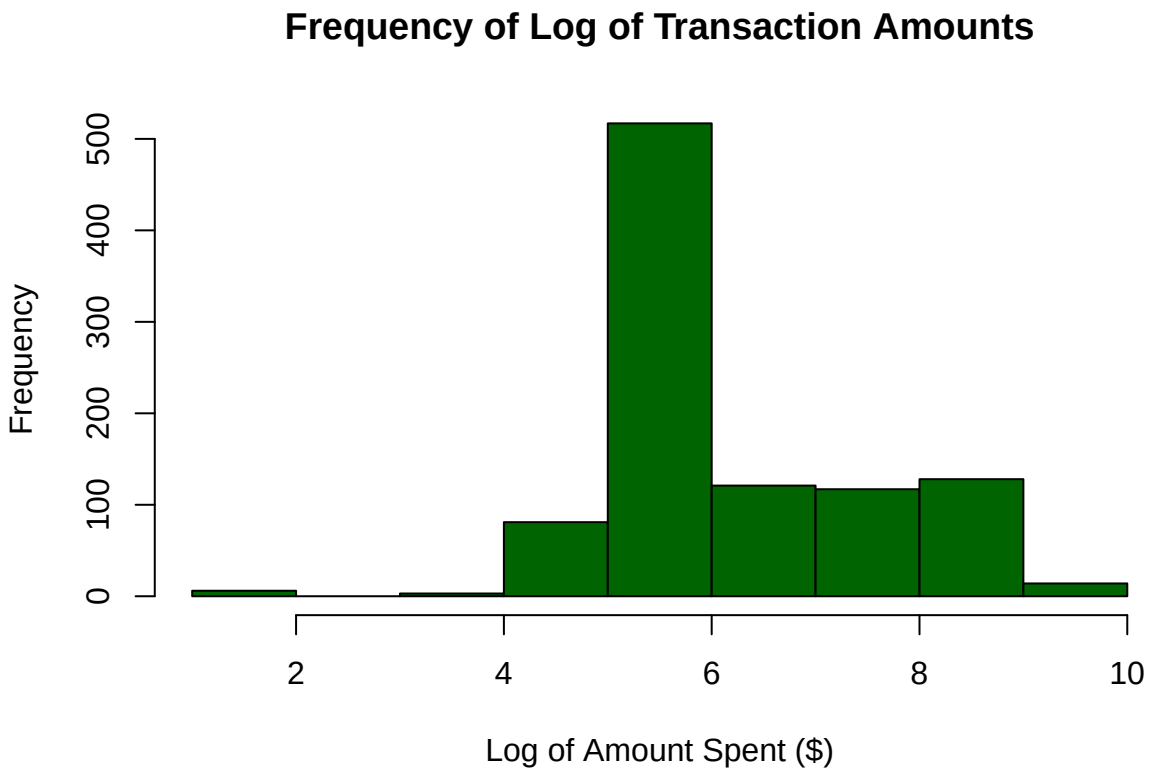
```
(getmode(amount))
```

```
## [1] 283.25
```

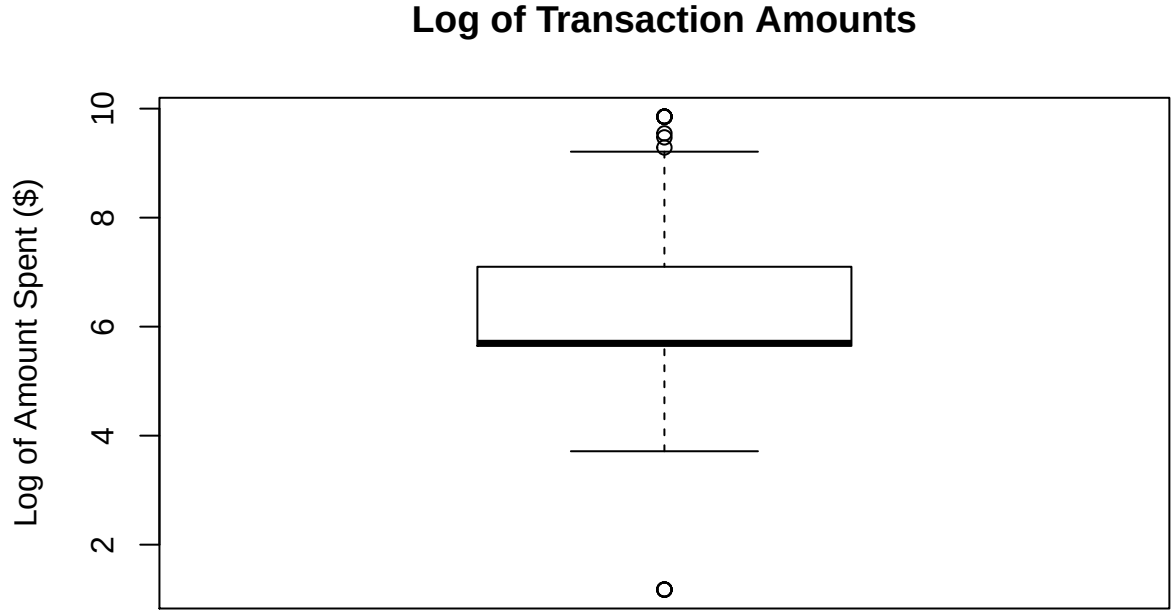
We can see that we will be expecting a rightly skewed histogram, as the mean transaction amount ($1347.42) is much higher than the median ($300.0). We also see that the mode ($283.25) is equal to the 1st Quartile, which would imply that the majority of values below the mean must have been $283.25. We will see later on that this in fact is true, as there were hundreds of orders for campaign t-shirts that were valued at exactly $283.25.



We see that the data is indeed skewed right, showing that the data does have a lower bound. Taking the log of the amounts improves the visualization.



This gives us a much nicer distribution to observe. In this histogram, we can clearly see that the area where log(x) falls between 5 and 6 (i.e. \$148-\$403) shows the most frequency, and we can conclude that most debt amounts do fall in this range. Finally, we observe these log amounts as a boxplot.



Again, we can see the median is hugging the first quartile value and we can observe several higher bound outliers compared to the one lower bound outlier, which we know is valued at \$3.24 (min value from summary). These higher amounts contribute to the data being rightly skewed.

2b) Description Frequency:

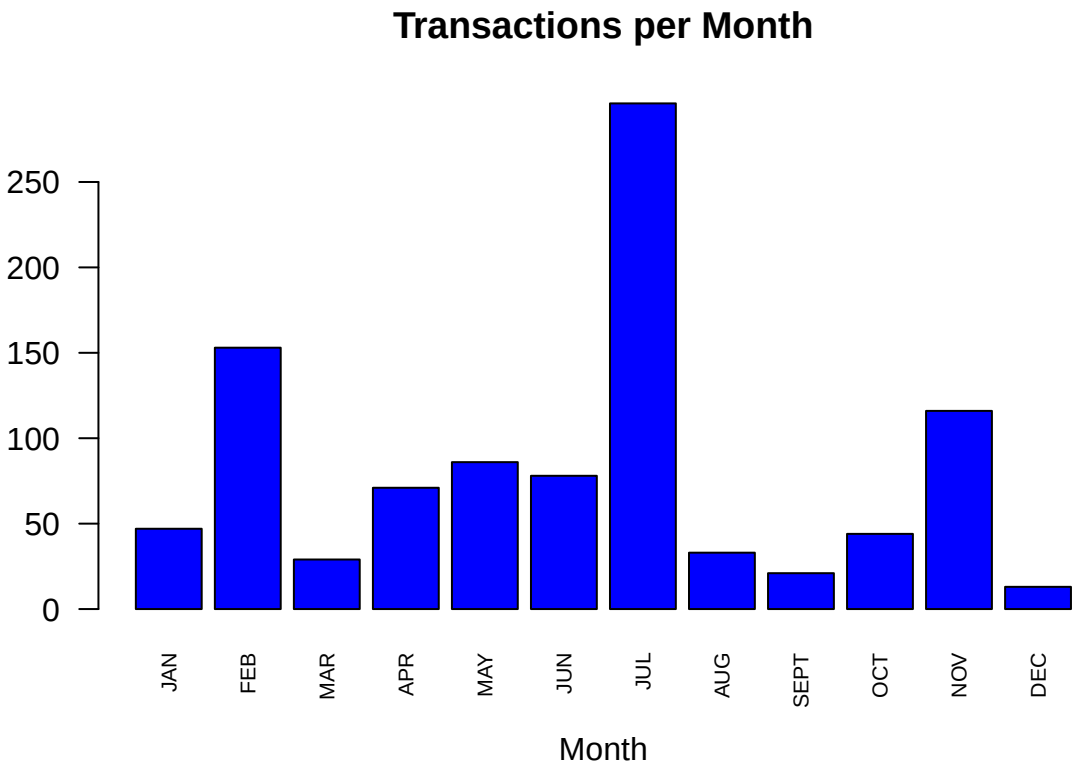
We also observed the `description` of the candidate debt transactions:

Table 1: Top 10 Transactions

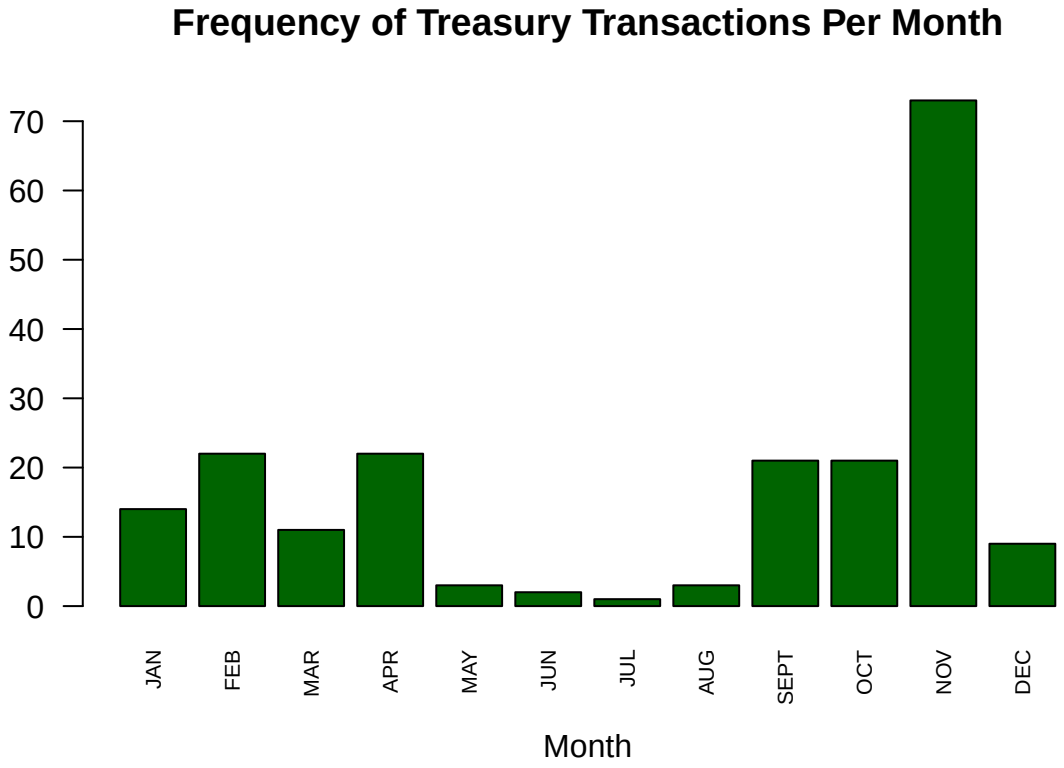
	description	%_transactions
72	RE-ORDER TEE SHIRTS	24.42%
28	CONSULTING/TRAVEL	8.61%
2	ACCOUNTING/COMPLIANCE	7.80%
62	NOVEMBER TREASURY	5.88%
50	JUNE FUNDRAISING	3.34%
81	REIMB. MTG EXP (MERCATO)	3.14%
64	OCTOBER TREASURY	2.13%
103	WIN BONUS	2.13%
87	SEPTEMBER TREASURY	1.93%
8	APRIL TREASURY	1.82%

Of the 987 total transactions recorded, almost 25% were for T-Shirts. This is not to say that T-Shirts, as a category, grossed the largest amount of debt. However, it does say a lot about where candidates are spending their money to spread the word about their campaign the most. We can also see that Consulting/Travel is the next most frequent transaction. This also would make sense as candidates, especially those running for statewide positions, would often be traveling around the state for speeches and appearances. With most U.S. elections taking place in Novemeber, spending naturally ramps up as election day gets closer. This would explain why Treasury transactions during the months of September, October, and November appear in the top 10 most frequent records.

2c) Month Frequency:

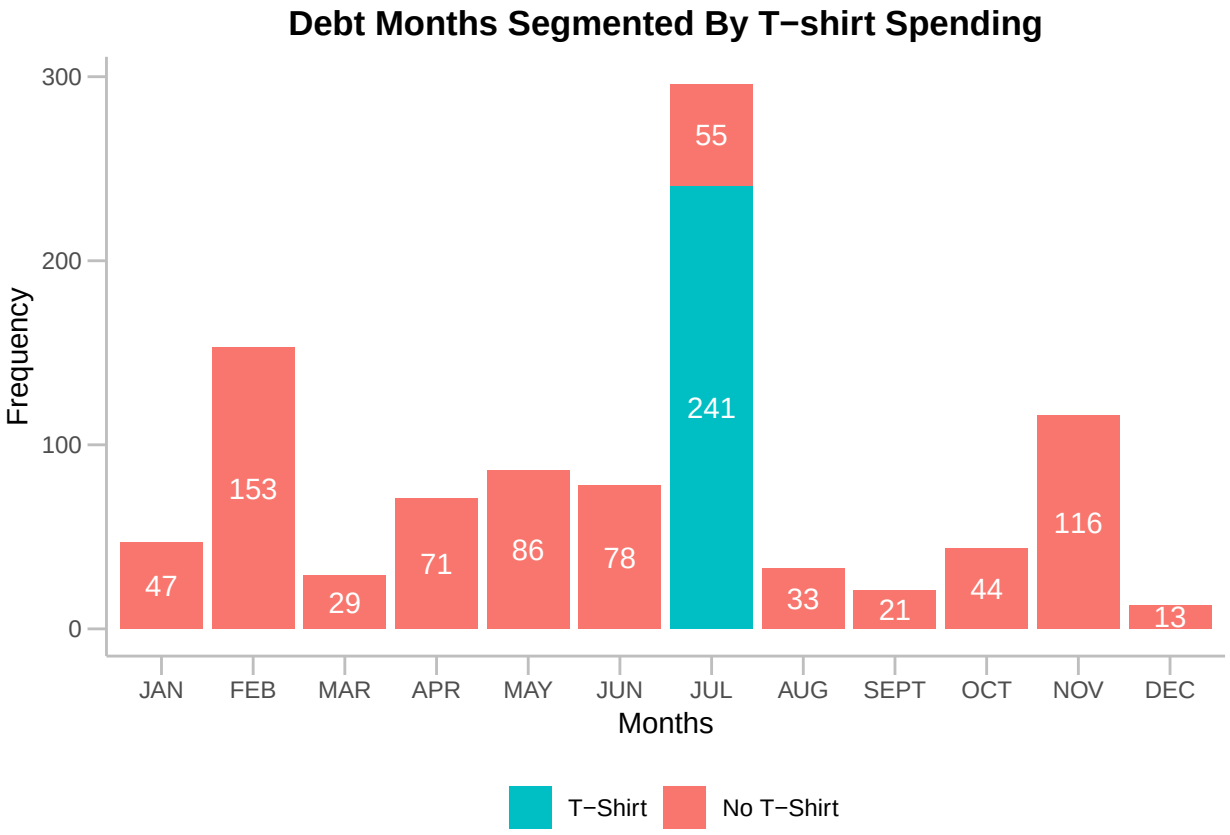


We can see that majority of debt records were recorded during the month of July. We mapped this barplot using `debtmonth` as our month indicator, which accounts for the month this debt was recorded. We recognized t-shirt debt was skewing this data to show that July had the most transactions, and thus decided to look specifically into monthly Treasury Debt. This category may indicate loans candidates took out in the specific month.



We can see a steady ramp up in Treasury debt as November approaches, as November has the most transactions by far of any other month. We also see a spike in January and February, which can be attributed to the primary elections that take place for some of the candidates during these months.

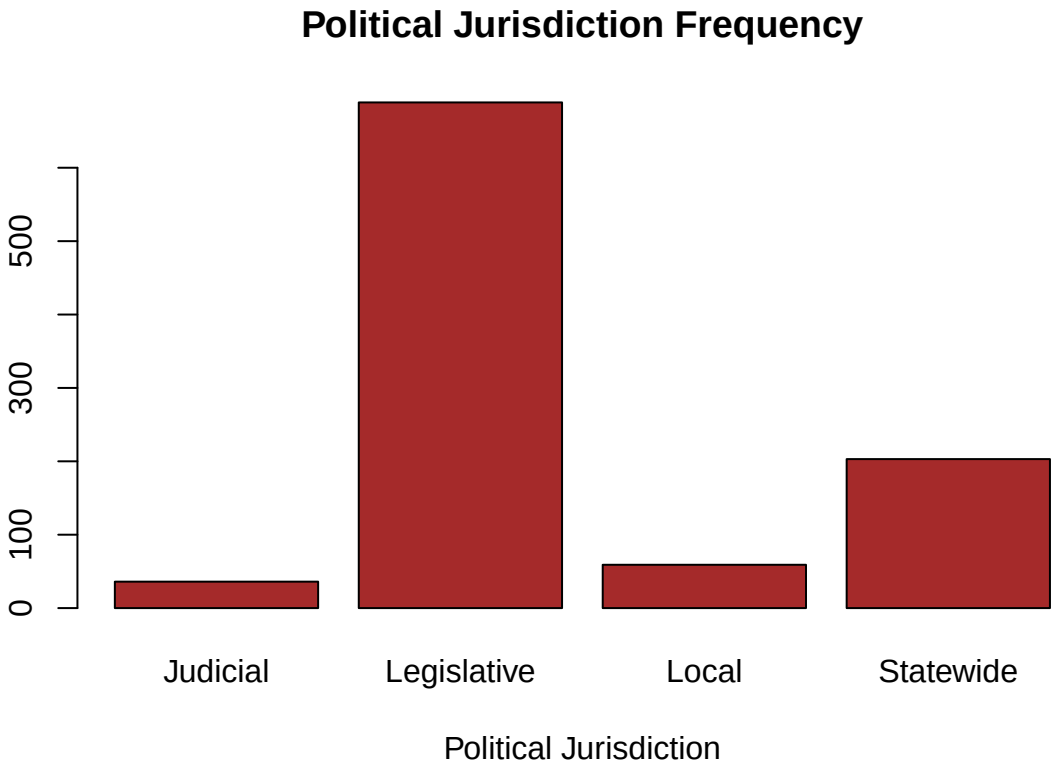
The unusual spending on T-Shirts we observed was quite interesting, and we decided to also look into what months T-Shirt sales were most prevalent.



Interestingly enough, all 241 T-Shirt transactions were made during the month of July. While this is normally the hottest month of the year for most states, it is still unusal that every candidate decided to purchase t-shirts during the same month, and not for example in April or May, in preparation for the summer.

2d) Jurisdiction Type:

Our final univariate analysis was on transactions for candidates with different Jurisdiction Types.



We can see here that candidates running for office in the Legislative Jurisdiction type accrued the most debt transactions by a large margin. While we can't make the assumption that candidates running for Legislative offices (State Senate, State Rep) spend more on their campaigns than those running for Statewide office (Attorney General, Governer), we do notice that the frequency of debt transactions is most likely related to the number of unique candidates running for these legislative offices:

```
length(unique(valid_data[valid_data$jurisdiction == "Legislative"]$filerid))
```

```
## [1] 124
```

```
length(unique(valid_data[valid_data$jurisdiction == "Statewide"]$filerid))
```

```
## [1] 87
```

```
length(unique(valid_data[valid_data$jurisdiction == "Judicial"]$filerid))
```

```
## [1] 31
```

```
length(unique(valid_data[valid_data$jurisdiction == "Local"]$filerid))
```

```
## [1] 46
```

A direct relationship can be seen between the number of unique candidates and the frequency of debt recorded. More unique candidates (Legislative having the most) relates to more transactions, while lower number of unique candidates (Judicial with the least) relates to less number of transactions.

3) Analysis of Key Relationships:

In order to better understand campaign characteristics that influenced the target variable `amount` we will examine some key variable relationships.

3a) `filerid` and `amount`:

First we will look at some candidates in terms of their debt `amount`. Let's conduct a simple overview to see the top 5 candidates in terms of their total debt `amount` and number of transactions:

	filerid	total_debt	num_transactions
44	HABIC 004	41581.18	12
12	CLIBJ 040	33116.03	16
74	MCAUR 011	32318.24	8
45	HAMID 528	32256.70	11
10	CHOPF 103	31795.49	23
119	SPRIL 033	30035.65	19

To put that in perspective, the median candidate total debt is around $6849.9. The median is an appropriate measure of typical debt becasue the distribution is skewed due to larger debt amounts.

If you recall the boxplot of individual transaction amounts (section 2a) that we talked about earlier, we see how there are some outliers, so it is reasonable to assume those are associated with some outlier candidates. If we consider outliers to be any total debt amount above $Q3 + 1.5 \times IQR$, the mean outlier debt (2 candidates) is $37348.6 and the mean non-outlier debt (remaining 138 candidates) is $9095.7 candidates. So the 2 outliers spent, on average, $28253 more than non-outliers, but only account for $74697.21 of the total candidate debt of $1329908.

Interestingly, some of the top expenses of all candidates, and specifically these outliers, are consulting expenses. More on consulting in section 3h.

3b) `office` and `amount`:

We need to know more about a candidate than just their total debt. A natural next step is to group candidates by the `office` they were seeking and then look at debt totals.

The `office` variable is supposed to explain what office a candidate was seeking. Unfortunately grouping with this variable is problematic as numerous candidates have multiple offices listed throughout their records. This results in less informative groupings so we need another approach.

3c) `position` and `amount`:

The `position` variable is very similar to the `office` variable as it also provides descriptive information about the job a candidate was seeking. It has the advantage of being a unique value for each candidate so we will use it for our groupings.

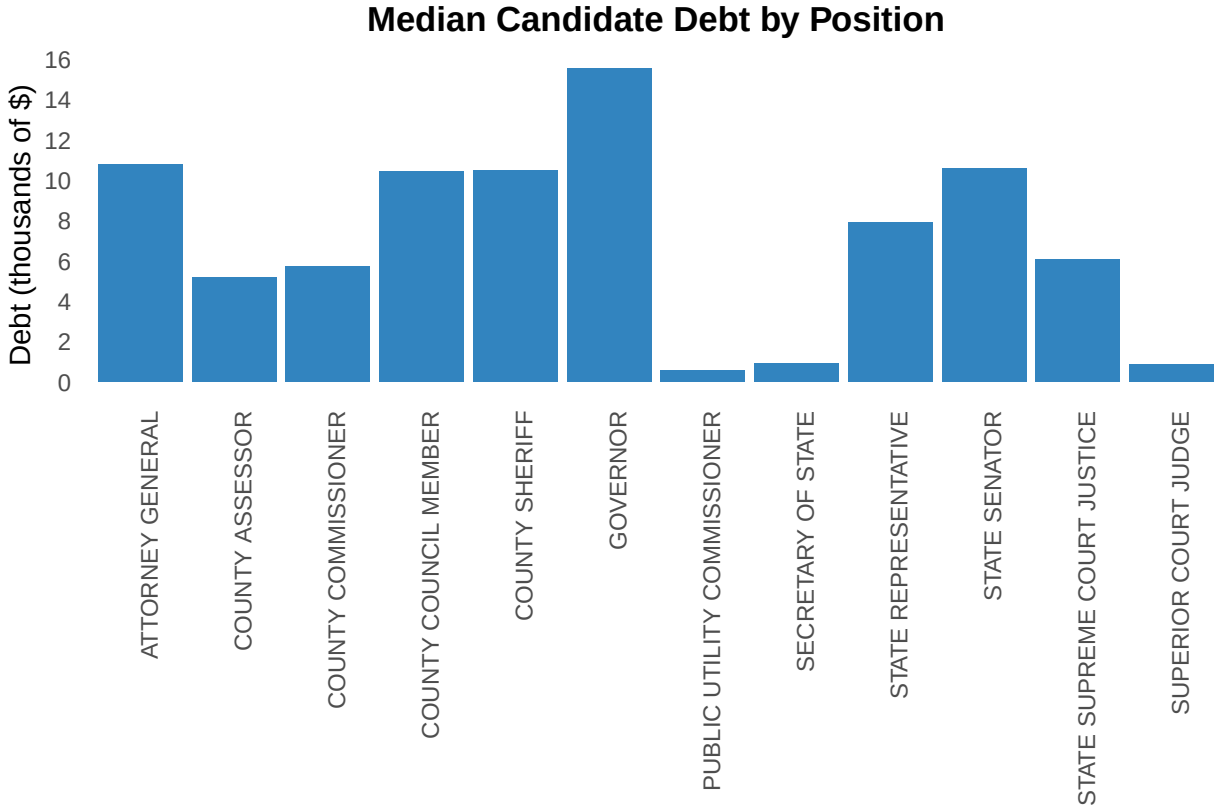
First, lets group by position and look at total debt and candidate median debt for the top 5 positions with the highest total debt `amount`:

	position	total	candidate_med	num_candidates
12	STATE REPRESENTATIVE	684100.80	7886.64	67
4	COUNTY COMMISSIONER	162983.31	5738.93	17
13	STATE SENATOR	141181.27	10603.60	13
16	SUPERIOR COURT JUDGE	53680.40	888.23	13
14	STATE SUPREME COURT JUSTICE	48972.72	6065.44	5
2	ATTORNEY GENERAL	47825.13	10765.42	4

We can see that median debt seems related to the position, and not necessarily related to how many candidates are campaigning for similar positions. This is quite clear if we look at the 5 positions with the least number of candidates:

	position	total	candidate_med	num_candidates
1	APPEALS COURT JUDGE	15741.56	15741.56	1
8	PUBLIC LANDS COMMISSIONER	20386.55	20386.55	1
11	STATE AUDITOR	10369.37	10369.37	1
15	STATE TREASURER	17909.91	17909.91	1
6	COUNTY SHERIFF	21012.02	10506.01	2
3	COUNTY ASSESSOR	11421.64	5175.56	3

Note that some positions with very few candidates are associated with higher debt. For a clearer picture of the required costs associated with a postion, we should limit the assessment to positions with multiple candidates:



This relationship could serve as a very useful first estimate for how much a candidate can anticipate spending given the position they are seeking.

3d) `legislativedistrict` and `amount`:

Initially, grouping by legislative district seems like a useful way to assess candidate debt by geographic region. Unfortunately the `legislativedisctrict` variable is not a unique value for each candidate.

With multiple legislative district values appearing across the records for the same candidate, and no clear explanation for what this means, it is hard to justify using this variable to group candidates by district in order to understand district-specific campaign costs.

3e) `party` and `amount`:

An inspection of the `party` variable reveals that throughout the data set 63 candidates have 2 different parties indicated, and 27 candidates have 3 different parties indicated. We have no explanation for what this means. Our assumption is that multiple party labels for the same candidate indicate that candidate switched their party affiliation during the year.

If this assumption is wrong, then grouping by `party` will provide an incorrect assessment of debt `amount`. We will conduct the grouping anyway with the caveat that checking this assumption, with additional information from the creators of this data set, would be necessary to clear up the ambiguity:

	party	total	candidate_med
1	DEMOCRAT	1153600.79	8347.57
4	REPUBLICAN	124508.50	849.75
3	NON PARTISAN	51696.07	592.50
2	INDEPENDENT	102.88	51.44

Given our assumption, it appears that candidates with a Democratic party affiliation spent significatnly more than other candidates. A comparison of the total debt amounts reveals that 86.7% of total debt is attributed to the Democratic candidates, and they spent $977293.3 more than the other parties combined.

3f) `jurisdictiontype` and `amount`:

Another way to group debt records is by the `jurisdictiontype` variable. While the `position` variable already provides a good solution for grouping, `jurisdictiontype` has only four levels which should make comparisons easier to understand:

	jurisdictiontype	total	pct_of_total	candidate_med
4	Statewide	876892.09	65.9	9124.25
2	Legislative	371291.60	27.9	2119.81
3	Local	58938.91	4.4	671.25
1	Judicial	22785.64	1.7	376.50

As we can see candidates seeking offices with Statewide or Legislative jurisdiction types make up the majority of the candidate debt (approx. 66% and 28% respecively).

Let's take these top juridiction types and break them down by types of expenses. First, the top 5 most-frequent expense types (statewide):

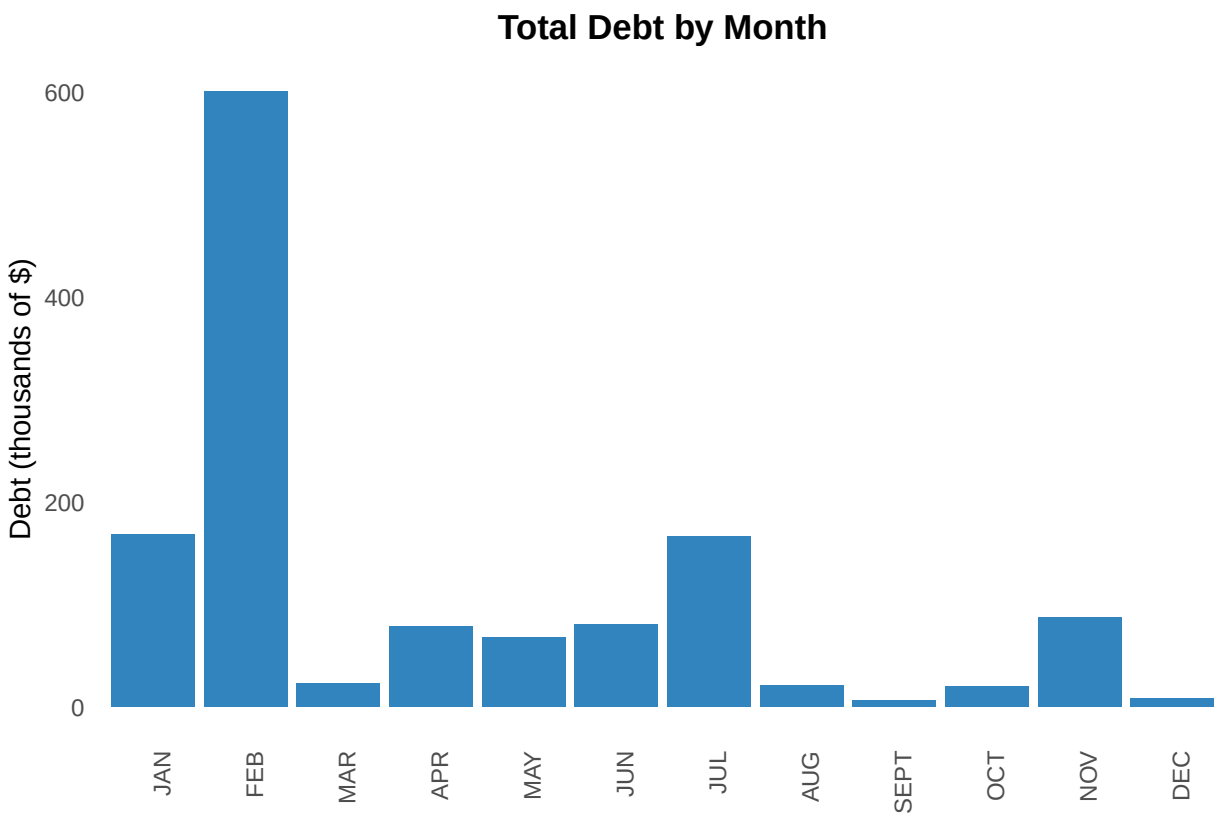
	expense (statewide jurisdiction)	Freq
28	CONSULTING/TRAVEL	85
2	ACCOUNTING/COMPLIANCE	45
17	CARRY FORWARD DEBT	16
40	EST. NOVEMBER TREASURY	14
20	CONSULTING	12
31	DATA SERVICES	5

What about those competing for jobs with a legislative jurisdiction type? Let's look at those top 5 most-frequent expense types:

	expense (legislative jurisdiction)	Freq
72	RE-ORDER TEE SHIRTS	241
62	NOVEMBER TREASURY	58
50	JUNE FUNDRAISING	33
81	REIMB. MTG EXP (MERCATO)	31
2	ACCOUNTING/COMPLIANCE	30

Note that the big difference between these groupings is consulting-related expenses. Consulting appears to be a significant part of campaign strategy for candidates seeking jobs with a statewide jurisdiction.

3g) `debtmonth` and `amount`:



February was by far the biggest month for candidate debt. Total debt for February was $600697.4. As can be seen below, consulting expenses dominated for the month of February:

	Feb expense	Freq
6	CONSULTING/TRAVEL	84
12	TREASURY - FEBRUARY	18
3	CARRY FORWARD DEBT	16
8	JANUARY SERVICES	13
1	$750 PER MONTH THROUGH OCTOBER	6
7	FEBRUARY TREASURY	4

3h) Consulting spend:

Given that consulting expenses seem to dominate much of this data set let's explore how consuling money is spent. When consulting expenses of all types are combined the total debt amount across all candidates is $707748.8. Below are the top candidates in terms of consulting spend:

	filerid	amount
25	HABIC 004	38000.00
26	HAMID 528	29140.28
7	CHOPF 103	24139.68
12	DUNNR 059	24070.14
33	JONEM 073	24070.14
67	WAGEJ 498	23210.42

Looks like Cyrus Habib, who was elected to the Washington State House in 2012, took the lead.

Another broader way of looking at consulting is grouping by the `position` variable, and then sorting by the total amount spent on consulting:

	position	amount	candidate_med	pct_of_total	num_candidates
11	STATE REPRESENTATIVE	346258.12	10140.28	48.9	34
4	COUNTY COMMISSIONER	134484.20	7605.21	19.0	12
12	STATE SENATOR	55268.50	5358.08	7.8	7
13	STATE SUPREME COURT JUSTICE	39155.99	18070.14	5.5	3
14	SUPERIOR COURT JUDGE	34210.42	5070.14	4.8	3
5	COUNTY COUNCIL MEMBER	25350.70	10140.28	3.6	3

We see that nearly 68% of consulting debt is accounted for by candidates seeking State Representative and County Comissioner positions.

Now that we have a better idea for how candidates for a particular position spend the money on consulting, let's look at what vendors are receiving that money:

	Vendor Name	Freq	vendor_state
3	HIRSCHBERG STRATEGIES INC.	84	DC
5	NEW PARTNERS CONSULTING INC.	8	DC
12	WINPOWER STRATEGIES	7	WA
13	WINPOWER STRATEGIES INC	6	WA
6	NEWMAN PARTNERS	5	WA
9	SOUTH COVE STRATEGIES	3	WA

HIRSCHBERG STRATEGIES INC. tops the list with $425,891.8, a whopping 60% of all consulting spend. They are located out-of-state in Washington DC.

In fact, 4 out of 14 consulting vendors were out-of-state. They accounted for 86% ($610391.8) of total consulting debt

3i) `t_shirts`:

Earlier we noticed an interesting trend in the candidate transactions related to 241 t-shirt orders in July 2012. Let's examine those orders in more detail.

If we subset the data according to those transactions the total debt is $68263.25. Checking this subset for unique candidate names we find 88 candidates are responsible for the 241 orders. Additionalyy, all the political party affiliations were Republican.

Another interesting feature of these transactions is that while some candidates have multiple transactions, the dates and amounts are the same for all 241 transactions. So why do some candidates have multiple records? We really have no way of knowing. Consequently we will keep all records, but remain skeptical about the accuracy of total amount.

Finally, we note that the same vendor, "Hickey Gayle", located in Washington state, was used for all 241 transactions.

4) Analysis of Secondary Effects:

Secondary variables that might have a confounding effect on the relationships that we have explored include details behind how `party` and `jurisdictiontype` are tallied. The reason for this is that candidates appear in more than one category when we would generally expect them to have unique groupings, in particular when dealing with one specific election cycle. Explaining how candidates can be tagged in more than one category could provide insight on how to separate the more direct effect between belonging to the associated categories of these fields and the resulting debt `amount`. This would account for the potentially counfounding effect at play.

Furthermore, our data set appears heavily levered (~25%) towards a couple of particular types of expenditure (i.e. `t_shirts` and consulting). Understanding elements that could be specific to this election cycle in particular in relation to these categories could also add important explanatory power as it relates to `amount`.

We also do not have a good way to gage geographic coverage of a position beyond broad local/state characterizations. A more granular understanding of the "prestige", potentially, as in reach a candidate perceives in a position could also add in separating effects of what drives spending.

Lastly, redundancy between categories makes it harder to target a specific variable for analysis. For example, State Representative appears both as an `office` and a `position` while varying over all `jurisdictiontype` entries (Judicial, Statewide, Local, Legislative). Without additional information, if candidates wanted to analyze State Representative debt activity, it would be difficult to know where to start data analysis from as this position has been gathered over multiple categories.

5) Conclusion:

This data set has given us some very useful insights into how specific campaign characteristics relate to candidate debt.

The most practical insight is that we now have a set of estimates for the median candiate campaign debts associated with certain positions. This provies a useful starting point for forecasting campaign financial requirements. To refine that estiamte, we also understand the breakdown of those debts in terms of individual transaction types. This can assist our campaign in structuring our campaign strategy to match, and potentially exceed, the requirements that similar candidates had in previous election years.

In terms of specific strategic planning, we know that consulting was important to candidates in this data set. However, we do not know why. We do not know what specific services those consulting firms provided, or the impact of the recommendations they provided. Therefore, we strongly recommend investigating the services

these consulting firms provide and assessing how their various approaches could benefit our strategic goals given our cost constraints.

We also recommend researching who won the 2012 elections and trying to model their campaigns based on how they decided to spend their money.

End of Report