|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Source Location** | **Process** | **Code** |
| Platform | User Created | Sorted through cleaned and merged SEC REG-CF filling data for form Cs. Further sorted data by companyName cells with startenginecapital. Then with this list, use Google and StartEngine website to find the funding page. All resulting scraped URLs were assigned StartEngine to the Platform variable. | merged\_dfse2$Platform <- 'StartEngine' |
| Platform\_Country | User Created | All portals are US based portals so US is simply assigned to this variable. | merged\_dfse2$Platform\_Country <- 'US' |
| nameOfIssuer.x | SEC | Simply a transfer from merged SEC fillings nameOfIssuer.x. |  |
| cik | SEC/ | Cik numbers originating from SEC fillings merging. Necessary to find cik on webpage to provide a consistent point for merging. Cik on webpage is obtained through a link to SEC documents page. The link is generally located in an a class in the li class jumbotron.financial-document. This contains all the document links. Html\_attr for href to extract all full link text. Grep to isolate the link that contains sec.gov/cgi-bin/browse-edgar in it. Str\_split to separate cik number from the rest of the link by CIK= and by &owner=. Grep to pick the section of the split that contains a digit. Gsub to remove heading zeros. Three entries are then manually filled in. | portaldata\_cikhtml <- html\_nodes(webpage,'li.jumbotron.financial-document a')  portaldata\_cik <- html\_attr(portaldata\_cikhtml, "href")  portaldata\_cik <- grep("www.sec.gov/cgi-bin/browse-edgar", portaldata\_cik, value=TRUE)  portaldata\_cik <- str\_split(portaldata\_cik, "CIK=", simplify=TRUE)  portaldata\_cik <- str\_split(portaldata\_cik, "&owner=", simplify=TRUE)  portaldata\_cik <- grep("[[:digit:]]",portaldata\_cik, value=TRUE)  portaldata\_cik <- portaldata\_cik[1]  portaldata\_cik <- gsub("(^|[^0-9])0+", "\\1", portaldata\_cik, perl = TRUE)  startenginedata\_df[162,"cik"] = 1741757  startenginedata\_df[144,"cik"] = 1740993  startenginedata\_df[6,"cik"] = 1688786 |
| stateOrCountry | SEC | Simply a transfer from merged SEC fillings stateOrCountry. |  |
| dateIncorporation | SEC | Simply a transfer from merged SEC fillings dateIncorporation. |  |
| securityOfferedType | SEC | Simply a transfer from merged SEC fillings securityOfferedType. |  |
| securityOfferedOtherDesc | SEC | Simply a transfer from merged SEC fillings securityOfferedOtherDesc. |  |
| inves |  | Information is located within a div class div.banner-stat.col-xs-12. Html\_text removes any html coding. Grep to select the string that contains investors. The first entry of this type is selected. Any \r or \n usually due to line breaks are removed. The word ‘investors’ is removed from string. Any extra spaces are removed. If-else statement inputs #N/A in case the information is missing. | portaldata\_inveshtml <- html\_nodes(webpage, 'div.banner-stat.col-xs-12')  portaldata\_inveshtml <- html\_text(portaldata\_inveshtml)  portaldata\_inveshtml <- grep("Investors", portaldata\_inveshtml, value=TRUE)  portaldata\_inves <- portaldata\_inveshtml[1]  portaldata\_inves <- gsub("[\r\n]", "", portaldata\_inves)  portaldata\_inves <- gsub("Investors", "", portaldata\_inves)  portaldata\_inves <- gsub(" ", "", portaldata\_inves)  portaldata\_inves <- ifelse(length(portaldata\_inves) == 0||portaldata\_inves == "", NA, portaldata\_inves) |
| raises |  | Target information is in a h2 node of a div class banner-stat.col-xs-12 node. Html\_text removes any html coding. Grep to select the string that contains a $. Select the first case that matches this condition. Any \r or \n usually due to line breaks are removed. Any spaces are removed. Node is then subset away from all empty nodes and properly selected. If-else statement inputs #N/A in case the information is missing. The dollar and plus signs are removed. The variable is changed to numeric to allow for comparisons. | portaldata\_raiseshtml <- html\_nodes(webpage, 'div.banner-stat.col-xs-12 h2')  portaldata\_raises <- html\_text(portaldata\_raiseshtml)  portaldata\_raises <- grep("\\$", portaldata\_raises, value=TRUE)  portaldata\_raises <- portaldata\_raises[1]  portaldata\_raises <- gsub("[\r\n]", " ", portaldata\_raises)  portaldata\_raises <- gsub(" ", "", portaldata\_raises)  portaldata\_raises <- ifelse(length(portaldata\_raises) == 0||portaldata\_raises == "", NA, portaldata\_raises)  startenginedata\_df$raises <- gsub("[\\$,\\+,]", "", startenginedata\_df$raises)  startenginedata\_df$raises <- as.numeric(startenginedata\_df$raises) |
| Campaign\_Duration | #N/A | No indication of Campaign Duration provided so this is left as #N/A | merged\_dfse2$Campaign\_Duration <- NA |
| deadlineDate | SEC | Simply a transfer from merged SEC fillings deadlineDate. |  |
| tags |  | Tag information is located in the same location so a xpath identifier is used. Html\_text removes any html coding. Remove any \r or \n usually due to line breaks. Remove any extra spaces. If-else statement inputs #N/A in case the information is missing. | portaldata\_lochtml <- html\_nodes(webpage,xpath='//\*[@id="the-profile"]/div[2]/div[2]/div[2]/h6/div[2]')  portaldata\_loc <- html\_text(portaldata\_lochtml)  portaldata\_loc <- gsub("[\r\n]", "", portaldata\_loc)  portaldata\_loc <- gsub("\\s+", " ", portaldata\_loc)  portaldata\_loc <- ifelse(length(portaldata\_loc) == 0||portaldata\_loc == "", NA, portaldata\_loc) |
| numberOfRounds | #N/A | No indication of which rounds are C-fillings and which are not so this variable is left as #N/A. | merged\_dfse2$numberOfRounds <- NA |
| zipCode | SEC | Simply a transfer from merged SEC fillings deadlineDate. |  |
| Survival | User Created | Survival is based off whether the listed website from SEC merged data is responsive or not. !http\_error (! = not) pings the website and returns 1 if there is no ping failure due to status code 400 or above. TryCatch is necessary to see the resulting errors and return any errors as 0. | merged\_dfse2$Survival <- sapply((merged\_dfse2$issuerWebsite), function(x){  tryCatch(  {  !http\_error(x, config(followlocation = 0L))  },  # ... but if an error occurs, tell me what happened:  error=function(error\_message) {  message("")  message("")  message(error\_message)  return(0)  }  )  }) |
| Successful\_Exit | User Created | \*incorrect variable\* Successful\_Exit is currently incorrectly defined as successful funding. Needs a rework. Here it is an if-else statement of whether thel raise listed is larger than the minimum offering variable. If minimum offering is less than total raise or variable is #N/A then 0 is the output, otherwise 1 is output. | merged\_dfse2$Successful\_Exit <- ifelse(merged\_dfse2$offeringAmount <= merged\_dfse2$raises, 1, 0)  merged\_dfse2$Successful\_Exit <- ifelse(is.na(merged\_dfse2$Successful\_Exit), 0, merged\_dfse2$Successful\_Exit) |
| issuerWebsite.x | SEC | A transfer from merged SEC fillings issuerWebsite. | merged\_dfse2$issuerWebsite <- as.character(merged\_dfse2$issuerWebsite)  merged\_dfse2$issuerWebsite <- gsub("https://", "" ,merged\_dfse2$issuerWebsite)  merged\_dfse2$issuerWebsite <- gsub("http://", "" ,merged\_dfse2$issuerWebsite)  merged\_dfse2$issuerWebsite[merged\_dfse2$issuerWebsite == "None"] <- NA |
| sm | #N/A | No area for social media coded on portal so this is left as #N/A | merged\_dfse2$sm <- NA |
| Appointments | #N/A | No indication of Appointments provided so this is left as #N/A | merged\_dfse2$Appointments <- NA |
| docs |  | All attached file links are contained in the a class of the li class jumbotron.financial-document node. Html\_attr extracts all the href or actual link text. All the document attachment links are combined into a single cell. If-else statement inputs #N/A in case the information is missing. | portaldata\_docshtml <- html\_nodes(webpage,'li.jumbotron.financial-document a')  portaldata\_docs <- html\_attr(portaldata\_docshtml, "href")  portaldata\_docs <- paste(portaldata\_docs, collapse = " ")  portaldata\_docs <- ifelse(length(portaldata\_docs) == 0||portaldata\_docs == "", NA, portaldata\_docs) |
| pitchv |  | The video link is generally contained in a javascript so a xpath identifier is used. Str\_split to isolate the text between video\_url and video\_transcript and select the correct string corresponding string fragment. Gsub to remove quotes, colons, and commas. If-else statement inputs #N/A in case the information is missing. | portaldata\_pitchvhtml <- html\_nodes(webpage, xpath='/html/body/script[8]')  portaldata\_pitchv <- str\_split(portaldata\_pitchvhtml, "video\_url", simplify=TRUE)  portaldata\_pitchv <- portaldata\_pitchv[2]  portaldata\_pitchv <- str\_split(portaldata\_pitchv, "video\_transcript", simplify=TRUE)  portaldata\_pitchv <- portaldata\_pitchv[1]  portaldata\_pitchv <- gsub('"', "", portaldata\_pitchv)  portaldata\_pitchv <- gsub(':', "", portaldata\_pitchv)  portaldata\_pitchv <- gsub(',', "", portaldata\_pitchv)  portaldata\_pitchv <- ifelse(length(portaldata\_pitchv) == 0||portaldata\_pitchv == "", NA, portaldata\_pitchv) |
| manag |  | Information is located in a javascript so a xpath identifier is used. Gsub to remove quotations. Str\_split to isolate the sections of the text between full\_name and twitter\_url. Grep to select the sections using ‘bio:’ as the identifier. Paste to combine all strings to a single string. Gsub to remove dashes, rnrn, title:, bio:, colons, u0026, and extra spaces. If-else statement inputs #N/A in case the information is missing. | portaldata\_managhtml <- html\_nodes(webpage,xpath='/html/body/script[8]')  portaldata\_managhtml <- gsub('"',"", portaldata\_managhtml)  portaldata\_manag <- str\_split(portaldata\_managhtml, "(full\_name)", simplify=TRUE)  portaldata\_manag <- str\_split(portaldata\_manag, "(twitter\_url)", simplify=TRUE)  portaldata\_manag <- grep("bio:", portaldata\_manag, value=TRUE)  portaldata\_manag <- paste(portaldata\_manag, collapse = " ")  portaldata\_manag <- gsub("\\", "", portaldata\_manag, fixed=TRUE)  portaldata\_manag <- gsub("rnrn", "", portaldata\_manag)  portaldata\_manag <- gsub("title:", " ", portaldata\_manag)  portaldata\_manag <- gsub("bio:", " ", portaldata\_manag)  portaldata\_manag <- gsub(":", "", portaldata\_manag)  portaldata\_manag <- gsub("u0026", "", portaldata\_manag)  portaldata\_manag <- gsub("\\s+", " ", portaldata\_manag)  portaldata\_manag <- ifelse(length(portaldata\_manag) == 0||portaldata\_manag == "", NA, portaldata\_manag) |
| offeringAmount | SEC | Simply a transfer from merged SEC fillings offeringAmount. |  |
| percent | #N/A | No relevant information so this variable assigned NA. | merged\_dfse2$percent <- NA |
| PercentEquityAvail | #N/A | No relevant information so this variable assigned NA. | merged\_dfse2$PercentEquityAvail <- NA |
| Valuation | #N/A | No relevant information so this variable assigned NA. | merged\_dfse2$Valuation <- NA |
| desc |  | Description under company name. Relevant node is the h4 class desc node in the div class splash-info.text-left.no-percent node. Html\_text() then removes html coding. Gsub removes any \r or \n usually due to line breaks and to remove any extra spacing. If-else statement inputs #N/A in case the information is missing. | portaldata\_deschtml <- html\_nodes(webpage, 'div.splash-info.text-left.no-percent h4.desc')  portaldata\_desc <- html\_text(portaldata\_deschtml)  portaldata\_desc <- gsub("[\r\n]", "", portaldata\_desc)  portaldata\_desc <- gsub("\\s+", " ", portaldata\_desc)  portaldata\_desc <- ifelse(length(portaldata\_desc) == 0||portaldata\_desc == "", NA, portaldata\_desc) |
| TaxRelief | #N/A | Tax Relief is not applicable to US portals so this is left as #N/A. | merged\_dfse2$TaxRelief <- NA |
|  | **Extra Data Begins Here** |  |  |
| uniqueid | User Created | A number assigned from 2001 to 1999 to signify companies are from StartEngine. 70 pages were scraped so a vector from 2001 to 2186 was assigned to each funding page. | startenginedata\_df$uniqueid <- c(2001:2186) |
| accession\_number | SEC | Simply a transfer from merged SEC fillings accession\_number. |  |
| compensationAmount | SEC | Simply a transfer from merged SEC fillings compensationAmount. |  |
| financialInterest | SEC | Simply a transfer from merged SEC fillings financialInterest. |  |
| noOfSecurityOffered | SEC | Simply a transfer from merged SEC fillings noOfSecurityOffered. |  |
| price | SEC | Simply a transfer from merged SEC fillings price. |  |
| priceDeterminationMethod | SEC | Simply a transfer from merged SEC fillings priceDeterminationMethod. |  |
| overSubscriptionAccepted | SEC | Simply a transfer from merged SEC fillings overSubscriptionAccepted. |  |
| overSubscriptionAllocationType | SEC | Simply a transfer from merged SEC fillings overSubscriptionAllocationType. |  |
| descOverSubscription | SEC | Simply a transfer from merged SEC fillings descOverSubscription. |  |
| maximumOfferingAmount | SEC | Simply a transfer from merged SEC fillings maximumOfferingAmount. |  |
| currentEmployees | SEC | Simply a transfer from merged SEC fillings currentEmployees. |  |
| totalAssetMostRecentFiscalYear | SEC | Simply a transfer from merged SEC fillings totalAssetMostRecentFiscalYear. |  |
| totalAssetPriorFiscalYear | SEC | Simply a transfer from merged SEC fillings totalAssetPriorFiscalYear. |  |
| cashEquiMostRecentFiscalYear | SEC | Simply a transfer from merged SEC fillings cashEquiMostRecentFiscalYear. |  |
| cashEquiPriorFiscalYear | SEC | Simply a transfer from merged SEC fillings cashEquiPriorFiscalYear. |  |
| actReceivedMostRecentFiscalYear | SEC | Simply a transfer from merged SEC fillings actReceivedMostRecentFiscalYear. |  |
| actReceivedPriorFiscalYear | SEC | Simply a transfer from merged SEC fillings actReceivedPriorFiscalYear. |  |
| shortTermDebtMostRecentFiscalYear | SEC | Simply a transfer from merged SEC fillings shortTermDebtMostRecentFiscalYear. |  |
| shortTermDebtPriorFiscalYear | SEC | Simply a transfer from merged SEC fillings shortTermDebtPriorFiscalYear. |  |
| longTermDebtMostRecentFiscalYear | SEC | Simply a transfer from merged SEC fillings longTermDebtMostRecentFiscalYear. |  |
| longTermDebtPriorFiscalYear | SEC | Simply a transfer from merged SEC fillings longTermDebtPriorFiscalYear. |  |
| revenueMostRecentFiscalYear | SEC | Simply a transfer from merged SEC fillings revenueMostRecentFiscalYear. |  |
| revenuePriorFiscalYear | SEC | Simply a transfer from merged SEC fillings revenuePriorFiscalYear. |  |
| costGoodsSoldMostRecentFiscalYear | SEC | Simply a transfer from merged SEC fillings costGoodsSoldMostRecentFiscalYear. |  |
| costGoodsSoldPriorFiscalYear | SEC | Simply a transfer from merged SEC fillings costGoodsSoldPriorFiscalYear. |  |
| taxesPaidMostRecentFiscalYear | SEC | Simply a transfer from merged SEC fillings taxesPaidMostRecentFiscalYear. |  |
| taxesPaidPriorFiscalYear | SEC | Simply a transfer from merged SEC fillings taxesPaidPriorFiscalYear. |  |
| netIncomeMostRecentFiscalYear | SEC | Simply a transfer from merged SEC fillings netIncomeMostRecentFiscalYear. |  |
| netIncomePriorFiscalYear | SEC | Simply a transfer from merged SEC fillings netIncomePriorFiscalYear. |  |
| isAmendment | SEC | Simply a transfer from merged SEC fillings isAmendment. |  |
| progressUpdate | SEC | Simply a transfer from merged SEC fillings progressUpdate. |  |
| natureOfAmendment | SEC | Simply a transfer from merged SEC fillings natureOfAmendment. |  |
| legalStatusForm | SEC | Simply a transfer from merged SEC fillings legalStatusForm. |  |
| legalStatusOtherDesc | SEC | Simply a transfer from merged SEC fillings legalStatusOtherDesc. |  |
| jurisdictionOrganization | SEC | Simply a transfer from merged SEC fillings jurisdictionOrganization. |  |
| street1 | SEC | Simply a transfer from merged SEC fillings street1. |  |
| street2 | SEC | Simply a transfer from merged SEC fillings street2. |  |
| city | SEC | Simply a transfer from merged SEC fillings city. |  |
| companyName | SEC | Simply a transfer from merged SEC fillings companyName. |  |
| commissionCik | SEC | Simply a transfer from merged SEC fillings commissionCik. |  |
| commissionFileNumber | SEC | Simply a transfer from merged SEC fillings commissionFileNumber. |  |
| crdNumber | SEC | Simply a transfer from merged SEC fillings crdNumber. |  |
| submission\_type | SEC | Simply a transfer from merged SEC fillings submission\_type. |  |
| filing\_date | SEC | Simply a transfer from merged SEC fillings filing\_date. |  |
| file\_number | SEC | Simply a transfer from merged SEC fillings file\_number. |  |
| period | SEC | Simply a transfer from merged SEC fillings period. |  |
| source | User Created | Source variable created through combine function relative to their naming in SECdfMerge.R. | merged\_df <- combine(sec2016\_q2, sec2016\_q3, sec2016\_q4, sec2017\_q1, sec2017\_q2, sec2017\_q3, sec2017\_q4, sec2018\_q1, sec2018\_q2, sec2018\_q3) |
| location | User Created | Location variable created for Google geocoding purposes. Punctuation removal to remove incompatible ‘#’s within street addresses. Addresses converted to character variables and then pasted together to a single variable. | merged\_df1.07$street1 <- gsub("[[:punct:][:blank:]]+", " ", merged\_df1.07$street1)  merged\_df1.07$street1 <- as.character(merged\_df1.07$street1)  merged\_df1.07$city <- as.character(merged\_df1.07$city)  merged\_df1.07$stateOrCountry <- as.character(merged\_df1.07$stateOrCountry)  merged\_df1.07$location <- paste0(merged\_df1.07$street1, ", ", merged\_df1.07$city, merged\_df1.07$stateOrCountry, merged\_df1.07$zipCode) |
| lon | User Created | This is the Google geocoded longitude coordinates. Google geocoding using location variable and then merged back into SEC merged file. | register\_google(key = "AIzaSyD2\_43O9jK2A6u44CLmqe6XEYUtx2R48HE", second\_limit = "50")  geo <- geocode(location = merged\_df1.07$location, output="latlon", source="google")  secgeo1 <- geo  merged\_df1.07$lon <- secgeo1$lon |
| lat | User Created | This is the Google geocoded latitude coordinates. Google geocoding using location variable and then merged back into SEC merged file. | register\_google(key = "AIzaSyD2\_43O9jK2A6u44CLmqe6XEYUtx2R48HE", second\_limit = "50")  geo <- geocode(location = merged\_df1.07$location, output="latlon", source="google")  secgeo1 <- geo  merged\_df1.07$lat <- secgeo1$lat |
| nameonportal |  | Name of company listed on portal. Information located in the h3 node of a div class splash-info.text-left.no-percent node. Html\_text to remove html coding. Gsub to remove any \r or \n usually due to line breaks and to remove any extra spacing. Tolower to change everything to lower case. Gsub to remove ‘the’, ‘inc’, ‘llc.’ and any punctuation and spacing. | portaldata\_namehtml <- html\_nodes(webpage, 'div.splash-info.text-left.no-percent h3')  portaldata\_name <- html\_text(portaldata\_namehtml)  portaldata\_name <- gsub("[\r\n]", " ", portaldata\_name)  portaldata\_name <- gsub("\\s+", " ", portaldata\_name)  portaldata\_name <- lapply(portaldata\_name, tolower)  portaldata\_name <- gsub("the", "", portaldata\_name)  portaldata\_name <- gsub("inc", "", portaldata\_name)  portaldata\_name <- gsub("llc.", "", portaldata\_name)  portaldata\_name <- gsub("[[:punct:]]", "", portaldata\_name)  portaldata\_name <- gsub(" ", "", portaldata\_name) |
| loc |  | Company location coded in the same location so a xpath identifier is used. Html\_text() to remove html coding. Gsub to remove any \r or \n usually due to line breaks and to remove extra spacing. If-else statement inputs #N/A in case the information is missing. | portaldata\_lochtml <- html\_nodes(webpage,xpath='//\*[@id="the-profile"]/div[2]/div[2]/div[2]/h6/div[2]')  portaldata\_loc <- html\_text(portaldata\_lochtml)  portaldata\_loc <- gsub("[\r\n]", "", portaldata\_loc)  portaldata\_loc <- gsub("\\s+", " ", portaldata\_loc)  portaldata\_loc <- ifelse(length(portaldata\_loc) == 0||portaldata\_loc == "", NA, portaldata\_loc) |
| pitcht |  | \*NOTE: highly subject to change. Collects a lot of text which possibly overloads cell limit. This currently contains all text information of webpage. Relevant node is the div class campaign-story node. Html\_text() then removes html coding. Gsub to remove any \r or \n usually due to line breaks, to remove extra spaces, and to remove extra dashes. Str\_split to isolate text data from other coding and then to select that string section. If-else statement inputs #N/A in case the information is missing. | portaldata\_pitchthtml <- html\_nodes(webpage, 'div.campaign-story')  portaldata\_pitcht <- html\_text(portaldata\_pitchthtml)  portaldata\_pitcht <- gsub("[\r\n]", "", portaldata\_pitcht)  portaldata\_pitcht <- gsub("\\s+", " ", portaldata\_pitcht)  portaldata\_pitcht <- gsub("\\\\", "", portaldata\_pitcht, fixed=TRUE)  portaldata\_pitcht <- str\_split(portaldata\_pitcht, "@media screen", simplify=TRUE)  portaldata\_pitcht <- portaldata\_pitcht[1]  portaldata\_pitcht <- ifelse(length(portaldata\_pitcht) == 0||portaldata\_pitcht == "", NA, portaldata\_pitcht) |
| raising |  | Currently raising information docated in a div class action-tabs.clearfix.show-ng node . Html\_text removes html coding. Grep to select the string that contains Invest Now.. If-else statement inputs 0 if such a node doesn’t exist and inputs 1 if otherwise. | raisinghtml <- html\_nodes(webpage, 'div.action-tabs.clearfix.show-ng')  raising <- html\_text(raisinghtml)  raising <- grep("Invest Now", raising, value=TRUE)  raising <- ifelse(length(raising) == 0||raising == "", 0, 1) |

**Sample Webpage for Reference Below:**

                     