04 Exploratory Analysis

Fabian Blasch

05/28/2022

1 Load Data

```
# source AUX
source("./../Misc/Auxilliary.R")

# load data
auctions <- readRDS("./../../Data/Bid Tab RDS/Bid_Tabs.RDS")
```

2 Required Data and Missigness

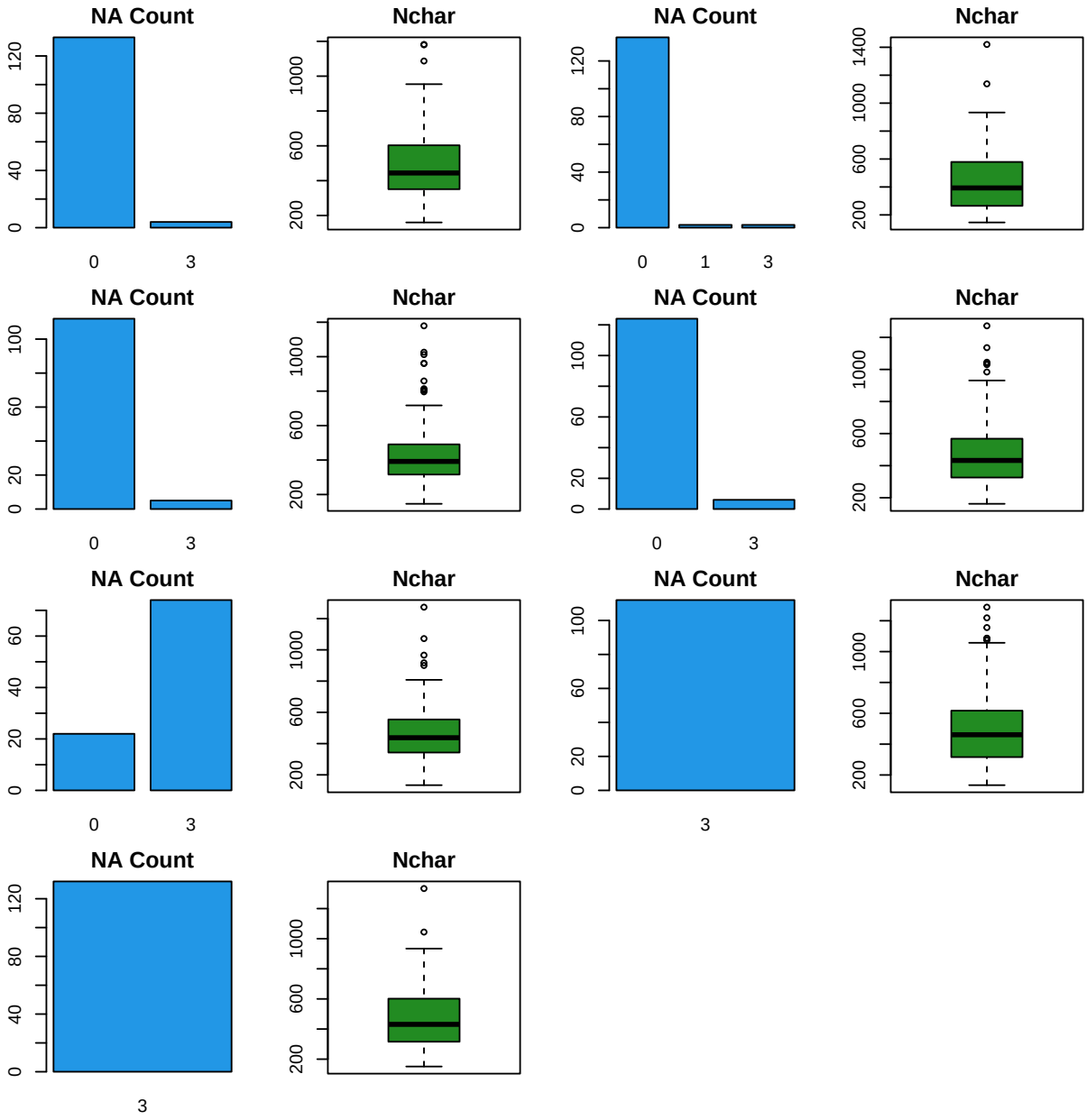
Firstly, we want to find the number of auctions that feature all characteristics required for the estimation procedure outlined in Krasnokutskaya 2012, i.e., letting date, completion time, location, tasks involved (description), identity of all bidders, their bids and an engineer's estimate.

```
# number of auctions available
sapply(auctions, length) |> sum()
```

```
## [1] 865
```

```
# align plots
par(mfrow = c(4, 4), mar = c(2, 2, 2, 2) + 0.1)

# over different years
invis.lapply(auctions, \(x){

  # over project ID
  lapply(x, \(y){

    # check
    NAs <- y$Text[c("Letting Date", "Contract Time", "Counties")] |> is.na()

    # no NAs
    na_check <- all(!NAs)

    # Description length
    Dlen <- y$Text["Contract Description"] |> nchar()

    # required min char count
    dlen_check <- Dlen > 200

    # Check for both conditions
    check <- na_check & dlen_check

    # return
```

```
    list("NAs" = NAs, "Description" = Dlen,
         "NAcheck" = na_check, "Dlcheck" = dlen_check,
         "Check" = check)

  }) -> tmp

  # display missigness in the three variables
  barplot(table(do.call(cbind, lapply(tmp, "[[", 1)) |> colSums()), xlab = "",
       ylab = "NAs", pch = 19, col = 4, main = "NA Count")

  # display nchar
  boxplot(sapply(tmp, "[[", 2), main = "Nchar", xlab = "", col = "forestgreen")

  # return
  return(tmp)

}) -> filter_dat
```



The plots are aligned by year from left to right and from top to bottom. We observe that the NA count for the variables of interest is 100% for 2021 and 2020. In 2019 some of th auctions seem to carry the required information but most do not. Accordingly, the subset of the data that we will pursue our estimation with will be limited to auctions from 2015 until 2019. Further, we observe that the character count of the description is quite consistent over time, with the median being very close to 400 characters for all years observed. In order to remove descriptions that are not particularly informative observations with a character count below 200 will be removed.

3 Subset of Auctions for Estimation

```
# fetch index for subsetting

# over years
```

```r
lapply(filter_dat, \(x){

  # over project ID
  sapply(x, "[[", "Check")

}) -> ind_check

# subset
Map(\(au, ind) au[ind], auctions, ind_check) -> auctions_checked

# number of remaining auctions
sapply(auctions_checked, length) |> sum()
```

```
## [1] 478
```

```r
# remove empty lists
auctions_checked[c("2020", "2021")] <- NULL
```

4 Further Data Cleaning

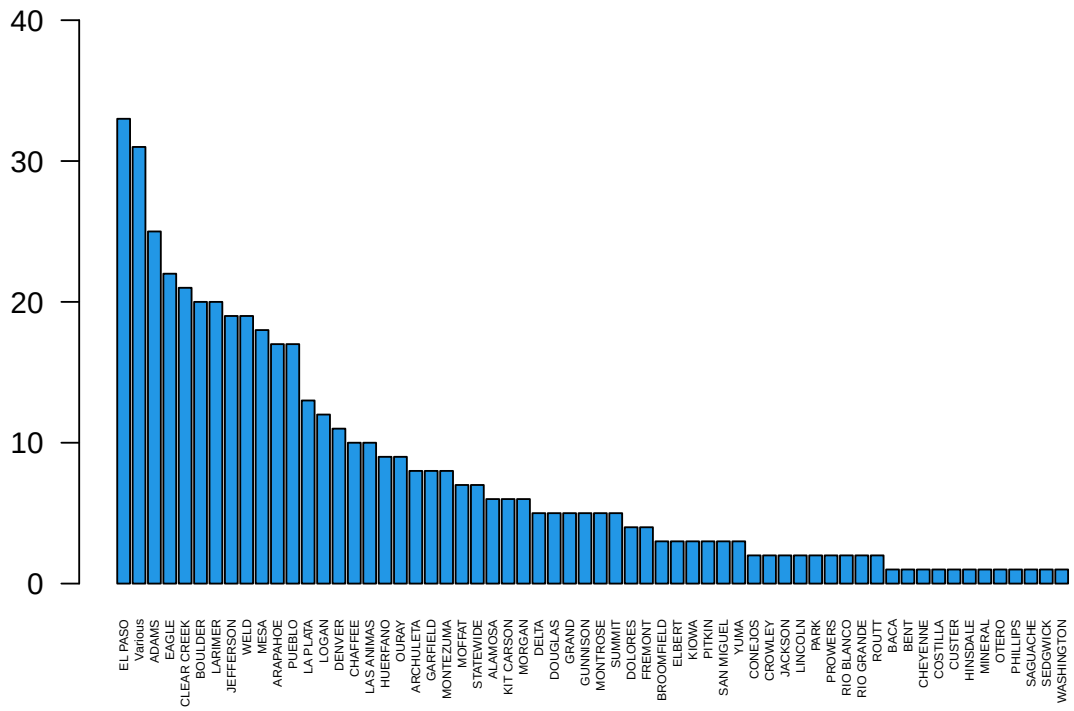
4.1 Counties

The county data that was scraped via *pdftools* still suffers to some textual impurities. Further, counties are split into sub-regions, accordingly to reduce the levels of the factor from 133 to 62, the county specific regions are removed.

```r
# pull counties from all auctions
counties <- pull_varT(auctions_checked, "Counties")

# factor levels
counties |> table() |> length()
```

```
## [1] 133
```

```r
# remove in county regional separation
counties_alt <- strsplit(counties, ",", fixed = TRUE) |> sapply(FUN = "[[", 1)

# result
barplot(sort(table(counties_alt), decreasing = TRUE), las = 2, cex.names = 0.4,
        ylim = c(0, 40), col = 4)
```




```r
# apply transformation to all auctions in the list
auctions_checked <- alt_varT(auctions_checked, "Counties",
                             Fun = \(x) strsplit(x, ",", fixed = TRUE) |>
                             sapply(FUN = "[[", 1))
```