Étude de productivité multifactorielle de *The Brattle Group*

Documents de soutien 2 à 7

Document de soutien 2

Brattle Controls Processing

Brattle

MEMORANDUM

TO Régie de l'énergie

FROM The Brattle Group

SUBJECT **Summary of Brattle Files**

DATE February 23, 2021

This file provides a summary of the files provided by The Brattle Group for the analysis contained in its report "Total Factor Productivity and the X-factor for Hydro-Quebec TransÉnergie" prepared for Hydro-Quebec TransÉnergie.

1. **Brattle TFP model [confidential]:**

This is an excel based model that calculates the growth of the TFP index for the US sample of transmission companies. All the data used for this analysis are contained within the model.

2. **Brattle HQT Data [not confidential]:**

This workbook contains all the data related to HQT. Calculations within this workbook are intended for the benchmarking analysis in which HQT is included among the sample of transmission companies.

3. **Brattle Benchmarking Analysis [not confidential]:**

This analysis pertains to the cost-based statistical benchmarking for the sample of transmission companies. There are three scripts here that were originally run in **Stata**. Each file applies to a specific cost:

a. Brattle Benchmarking Analysis_Total Costs: Applies to total costs.
b. Brattle Benchmarking Analysis_Capital: Applies to capital costs only.
c. Brattle Benchmarking Analysis_O&M: Applies to operations and maintenance (O&M) costs only.

4. **Brattle Benchmarking Results [not confidential]:**

Refers to three excel files that process results from the statistical benchmarking code. These are output files that are named as such in the Brattle Benchmarking Analysis code.

a. Brattle Benchmarking Results_Total Costs: Processes results for the total cost benchmarking code.
b. Brattle Benchmarking Results_Capital: Processes results for the capital cost benchmarking code.
c. Brattle Benchmarking Results_O&M: Processes results for the O&M cost benchmarking code.

5. **Brattle FERC Data Processing [not confidential]:**

This is a python script that converts the raw FERC Form 1 data from the FERC website, originally in the dbf format into usable excel files.

ReadMe - Summary of Files
Page 1 of 2

6. **Brattle Company Sample [not confidential]:**

This is an excel file, which provides a mapping of the naming conventions used for companies across two databases, the FERC and SNL. It is used as an input to the code that processes the control variables that are essential for the statistical benchmarking, described in further detail in the next section.

7. **Brattle Controls Processing [not confidential]:**

This refers to a collection of code files, originally run in **R**, that process the variables of interest, specifically the control variables that are essential to the statistical benchmarking analysis. The 8 files here help produce one excel file with all the necessary control variables, which is then used as an input file in the statistical benchmarking code.

a. **Brattle Controls Processing_Master:** This is the main script that runs all other scripts. This script extracts and cleans the substation capacity and transmission line data from the raw FERC Form 1 files, and outputs the business condition variables for the benchmarking study.
b. **Brattle Controls Processing_Read F1 Data:** Read FERC Form 1 Excel-based raw data and save as RDS files.
c. **Brattle Controls Processing_Read Company Sample List**: Read and clean list of companies to include in sample.
d. **Brattle Controls Processing_F1_1:** Clean the names of the list of companies to be used in other code.
e. **Brattle Controls Processing_F1_66:** Clean and summarize F1_66 data (substation capacity).
f. **Brattle Controls Processing_F1_74:** Clean and summarize F1_74 data (transmission lines).
g. **Brattle Controls Processing_Merge Datasets**: Merge dataframes created in F1_1, F1_66, and F1_74.
h. **Brattle Controls Processing_Outliers:** Manual adjustments to final dataset to account for outliers, data errors or inconsistencies.

ReadMe - Summary of Files
Page 2 of 2

```
1  ************************************************
2  * Purpose: US Sample Transmission               *
3  *          Benchmarking Study - Total Costs     *
4  ************************************************
5
6      clear
7      set more off
8
9  ********************************************************************************
10 * SET DIRECTORY                                                                 *
11 ********************************************************************************
12
13     global DIR          /*insert file directory path here*/
14     global INPUT        "$DIR\Input"
15     global INT          "$DIR\Intermediate"
16     global OUTPUT       "$DIR\Output"
17
18     global km_conv      1.60934
19
20 ********************************************************************************
21 * DATA PROCESSING                                                               *
22 ********************************************************************************
23
24
25     *Import dataset with output variables from Brattle TFP Model
26     import excel "/*filepath*/\Brattle TFP Model.xlsx", sheet("Benchmarking for Stata") firstrow clear
27     save "$INT\Benchmarking_outputs.dta", replace
28
29     * Import dataset with business conditioning variables - output from the code for the business
   conditioning variables
30     import delimited "$INPUT\BusinessConditionVariables_10.02.2019.csv", clear
31     drop v1 id company_name
32     rename (snl_company_name year) (CompanyName Year)
33     duplicates drop
34     * checking for duplicate observations
35     bysort CompanyName Year: gen N = _N
36     * double check dropping once we have the final dataset
37     drop if N>1
38     drop N
39     save "$INT\Condition_vars.dta", replace
40
41     * Merge the two datasets to create a master
42     use "$INT\Benchmarking_outputs.dta", replace
43     merge 1:1 CompanyName Year using "$INT\Condition_vars.dta"
44     * keep matches
45     keep if _merge == 3
46     * drop variables
47     drop total_tx_line_length _merge
48
49     * process NA's in the dataset
50     local varlist "SystemPeakMW InputPriceIndex LengthofTransmissionLinesmi TotalEnergyMWh
   RatchetedSystemPeakMW RatchetedTotalEnergyMWh avg_subs_cap subs_count avg_volt_tx_lines perc_ug
   subscountperlinekm PercentTxPlant TotalCapitalCosts000 CapitalPriceIndexForBenchma
   TotalLaborCosts000 TotalMaterialsServicesCost TotalOMCosts000 OMPriceIndexForBenchmarkin"
51     foreach var of local varlist{
52
53         replace `var' = "" if `var' == "NA" | `var' == "Inf"
54         destring `var', replace
55
56     }
57
58     * create variables required for regressions
59     gen Total_inputcost = TotalInputCosts000*1000
```

```
60      gen Total_capitalcost = TotalCapitalCosts000*1000
61      gen Total_OMcost = TotalOMCosts000*1000
62      gen Real_cost = (TotalInputCosts000*1000)/InputPriceIndex
63      gen Real_capital_cost = (TotalCapitalCosts000*1000)/CapitalPriceIndexForBenchma
64      gen Real_OM_cost = (TotalOMCosts000*1000)/OMPriceIndexForBenchmarkin
65      gen Tx_line_km = LengthofTransmissionLinesmi * $km_conv
66      replace length_ug = length_ug * $km_conv
67
68      * drop observations with no cost data
69      drop if Real_cost == .
70      drop if Real_capital_cost == .
71      drop if Real_OM_cost == .
72      * drop observations with no business conditioning variables
73      drop if avg_subs_cap == . | subs_count == . | avg_volt_tx_lines == . | length_ug == .
74      * recalculate perc underground and substations per line km
75      replace perc_ug = length_ug/Tx_line_km
76      replace subscountperlinekm = subs_count/Tx_line_km
77
78      * generate log variables for regressions
79      local varlist "Total_inputcost Total_capitalcost Total_OMcost TotalEnergyMWh SystemPeakMW
    RatchetedSystemPeakMW RatchetedTotalEnergyMWh PercentTxPlant avg_subs_cap subs_count
    avg_volt_tx_lines length_ug perc_ug subscountperlinekm Real_cost Real_capital_cost Tx_line_km
    Real_OM_cost"
80      foreach var of local varlist{
81
82          gen ln_`var' = ln(`var')
83
84      }
85
86      save "$INT\Benchmarking_regr_data.dta", replace
87
88  **** Processing HQT Data - update path appropriately **********************
89      * import HQT Dataset
90      import excel "/*filepath*/\Brattle HQT Data.xlsx", sheet("Benchmarking Stata") firstrow clear
91      save "$INT\HQT Benchmarking data.dta", replace
92      * gen variables consistent with US dataset
93      gen Total_inputcost = TotalInputCostsM*1000000
94      gen Total_capitalcost = TotalCapitalCostsM*1000000
95      gen Total_OMcost = TotalOMCostsM*1000000
96      gen Real_cost = (Total_inputcost)/InputPriceIndex
97      gen Real_capital_cost = (Total_capitalcost)/CapitalPriceIndexForBenchma
98      gen Real_OM_cost = (Total_OMcost)/OMPriceIndexForBenchmarkin
99      gen Tx_line_km = LengthofTransmissionLinesk
100     gen TotalEnergyMWh = TotalEnergyGWh*1000
101     gen RatchetedTotalEnergyMWh = RatchetedTotalEnergyGWh*1000
102     gen CompanyName = "HQT"
103     gen Companytoconsider = 1
104     drop TotalLaborCostsM MaterialsServicesExpense TotalCapitalCostsM TotalOMCostsM TotalInputCostsM
    GrowthofInputPriceIndex LengthofTransmissionLinesk TotalEnergyGWh RatchetedTotalEnergyGWh
105     order Companytoconsider CompanyName, first
106
107     * generate log variables
108     local varlist "Total_inputcost Total_capitalcost Total_OMcost TotalEnergyMWh SystemPeakMW
    RatchetedSystemPeakMW RatchetedTotalEnergyMWh PercentTxPlant avg_subs_cap subs_count
    avg_volt_tx_lines length_ug perc_ug subscountperlinekm Real_cost Real_capital_cost Real_OM_cost
    Tx_line_km"
109     foreach var of local varlist{
110
111         gen ln_`var' = ln(`var')
112
113     }
114
115     save "$INT\HQT Benchmarking data_processed.dta", replace
```

```
116
117 ***** Append HQT to US dataset ************************************************
118
119     use "$INT\Benchmarking_regr_data.dta", replace
120     * keep the same set of companies from the TFP study
121     keep if Companytoconsider == 1
122     append using "$INT\HQT Benchmarking data_processed.dta"
123     gen HQT = (CompanyName == "HQT")
124     save "$INT\Benchmarking_regr_data_wHQT.dta", replace
125
126 *******************************************************************************
127 * REGRESSION ANALYSIS                                                         *
128 *******************************************************************************
129
130     use "$INT\Benchmarking_regr_data_wHQT.dta", replace
131
132     * drop companies
133     drop if CompanyName == "Indiana Michigan Power Company"
134
135     encode USRegionCode, gen(USRegion0)
136     encode HWRegion, gen(HWRegion0)
137     encode NERCRegionCode, gen(NERCRegion0)
138
139     * declare data as panel
140     encode CompanyName, gen(CompanyName0)
141     xtset CompanyName0 Year, yearly
142
143     * create time trend
144     preserve
145         keep Year
146         duplicates drop
147         sort Year
148         gen trend = _n
149         save "$INT\time_trend.dta", replace
150     restore
151
152     * merge trend
153     merge m:1 Year using "$INT\time_trend.dta"
154     keep if _merge == 3
155     drop _merge
156     gen trend_sq = trend * trend
157
158
159     global ylist    ln_Real_cost
160     global xlist3   ln_Tx_line_km ln_RatchetedSystemPeakMW ln_TotalEnergyMWh PercentTxPlant
    subscountperlinekm avg_volt_tx_lines avg_subs_cap perc_ug trend
161
162
163     local varlist "fe re"
164
165     foreach var of local varlist{
166
167         * Reg 3: Log Log US Sample with energy
168         xtreg $ylist $xlist3 if HQT == 0, `var' cluster(CompanyName0)
169         est store `var'_3
170
171     }
172
173
174     local varlist "fe re"
175
176     foreach var of local varlist{
177
```

```
        * Reg 3: Log Log US Sample + HQT with energy
        xtreg $ylist $xlist3, `var' cluster(CompanyName0)
        est store `var'_HQT_3

    }


*******************************************************************************
* BENCHMARKING FOR HQT                                                         *
*******************************************************************************

    * Predict for fixed effects
    est restore fe_HQT_3
    predict yhat_fe_HQT_3, xbu
    global rmse_fe_HQT_3 = e(rmse)
    gen rmse_factor_fe = $rmse_fe_HQT_3 * $rmse_fe_HQT_3
    * generate Y hat for a log-log model
    gen cost_hat_fe_HQT_3 = exp(yhat_fe_HQT_3)*exp(rmse_factor_fe/2)
    gen diff_fe = ln(Real_cost/cost_hat_fe_HQT_3)

    * Predict for random effects
    est restore re_HQT_3
    predict yhat_re_HQT_3, xbu
    global rmse_re_HQT_3 = e(rmse)
    gen rmse_factor_re = $rmse_re_HQT_3 * $rmse_re_HQT_3
    * generate Y hat for a log-log model
    gen cost_hat_re_HQT_3 = exp(yhat_re_HQT_3)*exp(rmse_factor_re/2)
    gen diff_re = ln(Real_cost/cost_hat_re_HQT_3)

    * saving the summary stats of % diff for all companies and years
    local varlist "mean sd min max"
    foreach var of local varlist{

        preserve
            collapse (`var') diff_*, by(Year)
            export excel using "$OUTPUT\Brattle Benchmarking Results_TotalCosts.xlsx", sheet("percent_diff_`var'") sheetreplace firstrow(variables)
        restore

    }

    * Extract percent differences for HQT
    preserve
        keep if HQT == 1
        keep CompanyName Year InputPriceIndex Total_inputcost Real_cost cost_hat_fe_HQT_3 diff_fe cost_hat_re_HQT_3 diff_re
        export excel using "$OUTPUT\Brattle Benchmarking Results_TotalCosts.xlsx", sheet("HQT percent diff") sheetreplace firstrow(variables)
    restore

    * Export percent differences for full sample
    preserve
        keep CompanyName Year InputPriceIndex Total_inputcost Real_cost cost_hat_fe_HQT_3 diff_fe cost_hat_re_HQT_3 diff_re
        export excel using "$OUTPUT\Brattle Benchmarking Results_TotalCosts", sheet("Full Sample percent diff") sheetmodify firstrow(variables)
    restore

```

```
1  ************************************************
2  * Purpose: US Sample Transmission               *
3  *          Benchmarking Study - Capital Costs    *
4  ************************************************
5
6      clear
7      set more off
8
9  *******************************************************************************
10 * SET DIRECTORY                                                                *
11 *******************************************************************************
12
13     global DIR          /*insert file directory path here*/
14     global INPUT        "$DIR\Input"
15     global INT          "$DIR\Intermediate"
16     global OUTPUT       "$DIR\Output"
17
18     global km_conv      1.60934
19
20 *******************************************************************************
21 * REGRESSION ANALYSIS                                                          *
22 *******************************************************************************
23
24     * import regression dataset from the total cost benchmarking do file
25     use "$INT\Benchmarking_regr_data_wHQT.dta", replace
26
27     * drop companies
28     drop if CompanyName == "Indiana Michigan Power Company"
29
30     encode USRegionCode, gen(USRegion0)
31     encode HWRegion, gen(HWRegion0)
32     encode NERCRegionCode, gen(NERCRegion0)
33
34     * declare data as panel
35     encode CompanyName, gen(CompanyName0)
36     xtset CompanyName0 Year, yearly
37
38     * create time trend
39     preserve
40         keep Year
41         duplicates drop
42         sort Year
43         gen trend = _n
44         save "$INT\time_trend.dta", replace
45     restore
46
47     * merge trend
48     merge m:1 Year using "$INT\time_trend.dta"
49     keep if _merge == 3
50     drop _merge
51     gen trend_sq = trend * trend
52
53
54     global ylist    ln_Real_capital_cost
55     global xlist3   ln_Tx_line_km ln_RatchetedSystemPeakMW ln_TotalEnergyMWh PercentTxPlant
   subscountperlinekm avg_volt_tx_lines avg_subs_cap perc_ug trend
56
57
58     local varlist "fe re"
59
60     foreach var of local varlist{
61
62         * Reg 3: Log Log US Sample with energy
```

```
63          xtreg $ylist $xlist3 if HQT == 0, `var' cluster(CompanyName0)
64          est store `var'_3
65
66      }
67
68
69      local varlist "fe re"
70
71      foreach var of local varlist{
72
73          * Reg 3: Log Log US Sample + HQT  with energy
74          xtreg $ylist $xlist3, `var' cluster(CompanyName0)
75          est store `var'_HQT_3
76
77      }
78
79
80  ********************************************************************************
81  * BENCHMARKING FOR HQT                                                         *
82  ********************************************************************************
83
84      * Predict for fixed effects
85      est restore fe_HQT_3
86      predict yhat_fe_HQT_3, xbu
87      global rmse_fe_HQT_3 = e(rmse)
88      gen rmse_factor_fe = $rmse_fe_HQT_3 * $rmse_fe_HQT_3
89      * generate Y hat for a log-log model
90      gen cost_hat_fe_HQT_3 = exp(yhat_fe_HQT_3)*exp(rmse_factor_fe/2)
91      gen diff_fe = ln(Real_capital_cost/cost_hat_fe_HQT_3)
92
93      * Predict for random effects
94      est restore re_HQT_3
95      predict yhat_re_HQT_3, xbu
96      global rmse_re_HQT_3 = e(rmse)
97      gen rmse_factor_re = $rmse_re_HQT_3 * $rmse_re_HQT_3
98      * generate Y hat for a log-log model
99      gen cost_hat_re_HQT_3 = exp(yhat_re_HQT_3)*exp(rmse_factor_re/2)
100     gen diff_re = ln(Real_capital_cost/cost_hat_re_HQT_3)
101
102     * saving the summary stats of % diff for all companies and years
103     local varlist "mean sd min max"
104     foreach var of local varlist{
105
106         preserve
107             collapse (`var') diff_*, by(Year)
108             export excel using "$OUTPUT\Brattle Benchmarking Results_Capital.xlsx", sheet(
    "percent_diff_`var'") sheetreplace firstrow(variables)
109         restore
110
111     }
112
113     * Extract percent differences for HQT
114     preserve
115         keep if HQT == 1
116         keep CompanyName Year InputPriceIndex Total_inputcost Real_cost cost_hat_fe_HQT_3 diff_fe
    cost_hat_re_HQT_3 diff_re
117         export excel using "$OUTPUT\Brattle Benchmarking Results_Capital.xlsx", sheet("HQT percent
    diff") sheetreplace firstrow(variables)
118     restore
119
120     * Export percent differences for full sample
121     preserve
122         keep CompanyName Year InputPriceIndex Total_inputcost Real_cost cost_hat_fe_HQT_3 diff_fe
```

```
cost_hat_re_HQT_3 diff_re
123         export excel using "$OUTPUT\Brattle Benchmarking Results_Capital.xlsx", sheet("Full Sample
percent diff") sheetmodify firstrow(variables)
124     restore
125
```

```
1  ************************************************
2  * Purpose: US Sample Transmission               *
3  *          Benchmarking Study - O&M Costs       *
4  ************************************************
5
6      clear
7      set more off
8
9  *******************************************************************************
10 * SET DIRECTORY                                                                *
11 *******************************************************************************
12
13     global DIR          /*insert file directory path here*/
14     global INPUT        "$DIR\Input"
15     global INT          "$DIR\Intermediate"
16     global OUTPUT       "$DIR\Output"
17
18     global km_conv      1.60934
19
20 *******************************************************************************
21 * REGRESSION ANALYSIS                                                          *
22 *******************************************************************************
23
24     * import regression dataset from the total cost benchmarking do file
25     use "$INT\Benchmarking_regr_data_wHQT.dta", replace
26
27     * drop companies
28     drop if CompanyName == "Indiana Michigan Power Company"
29
30     encode USRegionCode, gen(USRegion0)
31     encode HWRegion, gen(HWRegion0)
32     encode NERCRegionCode, gen(NERCRegion0)
33
34     * declare data as panel
35     encode CompanyName, gen(CompanyName0)
36     xtset CompanyName0 Year, yearly
37
38     * create time trend
39     preserve
40         keep Year
41         duplicates drop
42         sort Year
43         gen trend = _n
44         save "$INT\time_trend.dta", replace
45     restore
46
47     * merge trend
48     merge m:1 Year using "$INT\time_trend.dta"
49     keep if _merge == 3
50     drop _merge
51     gen trend_sq = trend * trend
52
53
54     global ylist    ln_Real_OM_cost
55     global xlist3   ln_Tx_line_km ln_RatchetedSystemPeakMW ln_TotalEnergyMWh PercentTxPlant
   subscountperlinekm avg_volt_tx_lines avg_subs_cap perc_ug trend
56
57
58     local varlist "fe re"
59
60     foreach var of local varlist{
61
62         * Reg 3: Log Log US Sample with energy
```

```
63          xtreg $ylist $xlist3 if HQT == 0, `var' cluster(CompanyName0)
64          est store `var'_3
65
66      }
67
68
69      local varlist "fe re"
70
71      foreach var of local varlist{
72
73          * Reg 3: Log Log HQT with energy
74          xtreg $ylist $xlist3, `var' cluster(CompanyName0)
75          est store `var'_HQT_3
76
77      }
78
79
80  ********************************************************************************
81  * BENCHMARKING FOR HQT                                                         *
82  ********************************************************************************
83
84      * Predict for fixed effects
85      est restore fe_HQT_3
86      predict yhat_fe_HQT_3, xbu
87      global rmse_fe_HQT_3 = e(rmse)
88      gen rmse_factor_fe = $rmse_fe_HQT_3 * $rmse_fe_HQT_3
89      * generate Y hat for a log-log model
90      gen cost_hat_fe_HQT_3 = exp(yhat_fe_HQT_3)*exp(rmse_factor_fe/2)
91      gen diff_fe = ln(Real_OM_cost/cost_hat_fe_HQT_3)
92
93      * Predict for random effects
94      est restore re_HQT_3
95      predict yhat_re_HQT_3, xbu
96      global rmse_re_HQT_3 = e(rmse)
97      gen rmse_factor_re = $rmse_re_HQT_3 * $rmse_re_HQT_3
98      * generate Y hat for a log-log model
99      gen cost_hat_re_HQT_3 = exp(yhat_re_HQT_3)*exp(rmse_factor_re/2)
100     gen diff_re = ln(Real_OM_cost/cost_hat_re_HQT_3)
101
102     * saving the summary stats of % diff for all companies and years
103     local varlist "mean sd min max"
104     foreach var of local varlist{
105
106         preserve
107             collapse (`var') diff_*, by(Year)
108             export excel using "$OUTPUT\Brattle Benchmarking Results_O&M.xlsx", sheet(
    "percent_diff_`var'") sheetreplace firstrow(variables)
109         restore
110
111     }
112
113     * Extract percent differences for HQT
114     preserve
115         keep if HQT == 1
116         keep CompanyName Year InputPriceIndex Total_inputcost Real_OM_cost cost_hat_fe_HQT_3 diff_fe
    cost_hat_re_HQT_3 diff_re
117         export excel using "$OUTPUT\Brattle Benchmarking Results_O&M.xlsx", sheet("HQT percent diff")
     sheetreplace firstrow(variables)
118     restore
119
120     * Export percent differences for full sample
121     preserve
122         keep CompanyName Year InputPriceIndex Total_inputcost Real_OM_cost cost_hat_fe_HQT_3 diff_fe
```

```
    cost_hat_re_HQT_3 diff_re
123         export excel using "$OUTPUT\Brattle Benchmarking Results_O&M.xlsx", sheet("Full Sample
    percent diff") sheetmodify firstrow(variables)
124     restore
125
```

```
################################################################################
### THE BRATTLE GROUP
### PURPOSE: Convert dbf files in the FERC Form 1 raw data into Excel-based files
################################################################################

### Import packages
from dbfread import DBF
import os
import pandas as pd

### Define path
path = '' # ADD PATH TO FOLDER CONTAINING FERC FORM 1 FILES FOR A GIVEN YEAR

### Create list of names of FERC Form 1 DBF files to import
files = []
for filename in os.listdir(path):
   if filename.endswith('.DBF'):
      files.append(filename)

### Import files into dictionary
d = {}
files.sort()
for i in files:
   try:
      table = DBF(i, load = True)
   except Exception:
      print(i,'  ', 'failed filename')
   table_df = pd.DataFrame.from_dict(table)
   d[i] = table_df

### Write dataframes into Excel sheets
writer = pd.ExcelWriter('', engine='xlsxwriter') # ADD FILE PATH AND NAME

for i in files:
   try:
      d[i].to_excel(writer, sheet_name = i)
   except Exception:
      print(i,'  ', 'failed filename')

### Close the Pandas Excel writer and save the Excel file
writer.save()
```

```
################################################################################
###   THE BRATTLE GROUP
###   PURPOSE: This is the main script that runs all other scripts.
###   This script extracts and cleans the substation capacity and transmisssion
###   line data from the raw FERC Form 1 files, and outputs the business
###   condition variables for the benchmarking study
################################################################################


# Notes on business condition variables:
# - Avg substation capacity (File F1_66)
# -     Number of transmission substations per kM of transmission line = Number of substations (File F1_66) / Total
km tx lines (File F1_74)
# - Avg. voltage of tx lines (File F1_74)
# - (% of underground lines (4) U.G.) (File F1_74)


### Load Packages ###
if (!require("pacman")) install.packages("pacman")
pacman::p_load(tidyverse,
               rio,
               magrittr,
               zoo,
               dplyr,
               lubridate,
               stringr,
               readxl,
               BrattleExtras,
               SciViews,
               ggplot2,
               openxlsx)


### Set working directory ###
setwd("") ### ADD PATH TO WORKING DIRECTORY


### Clear up workspace ###
rm(list=ls(all = T))
gc()


### prevent scientific notation ###
options(scipen=999)




### 1 --- Read excel-formatted data and save data as RDS file
source("./Code/1 - Read F1 Data.R")


### 2.a --- Read list of companies in sample
source("./Code/2 - Read Company Sample List.R")


### 2.b --- Read table with Tx line length data from SNL
TxLineLength_SNL <- read_csv("X:/6000/6705_HQT_TFP/Analysis/FERC Form 1 Data/Business Condition
Variables/Input/Tx_Line_Length.csv")


### 3 --- Read and clean list of companies to include

for (year in c(1994:2019)) {

  # read RDS file
  list_all <- readRDS(paste0("./Intermediate/", year, "_data.rds"))


  ### Clean F1_1 data: Company name and ID
  source("./Code/3a - F1_1.R")


  ### Clean F1_66 data: Substation Capacity
  source("./Code/3b - F1_66.R")


  ### Clean F1_74 data: Tx line
  source("./Code/3c - F1_74.R")


  ### Merge datasets from F1_1, F1_66 and F1_74
  source("./Code/3d - Merge Datasets.R")
```

```
} # end for-loop


### Sort by company name and by year
combined_total %<>% arrange(SNL_Company_Name, year)


### 4 --- Dealing with outliers:
source("./Code/4 - Outliers.R")



### Export dataframe
write.csv(combined_total, paste0("")) ### ADD FILE PATH AND NAME
```

```
################################################################################
###   THE BRATTLE GROUP
###   PURPOSE: Read FERC Form 1 Excel-based raw data and save as RDS files
################################################################################


# Names of tabs containing relevant data
tab_names <- c("F1_1.DBF", "F1_66.DBF", "F1_74.DBF")

# Path to folder where raw data (files above) are saved
path_name <- "" # ADD PATH TO FOLDER


# Read excel files for each year and save as RDS
for (year in c(1994:2019)) {

  # Read excel files (F1_1, F1_66, F1_74)
  list_all <- lapply(tab_names, function(x) read_excel(path = paste0(path_name, "/f1_", year,".xlsx"), sheet = x))

  # Save data as RDS list
  saveRDS(list_all, paste0("./Intermediate/", year, "_data.rds"))

  # check
  print(year)

}
```

```
################################################################################
###   THE BRATTLE GROUP
###   PURPOSE: Read and clean list of companies to include in sample
################################################################################

# Read list of utilities included in sample
company_list <- read_csv("./Input/Brattle Company_Sample.csv")


# Clean company_list dataframe
company_list %<>%

  # exclude utilities labeled as include = "No"
  filter(Include != "No") %>%

  # format ID as integer
  mutate(ID = as.integer(ID)) %>%

  # keep relevant columns only
  select(SNL_Company_Name, Include, FERC_Company_Name, ID)
```

```
################################################################################
###   THE BRATTLE GROUP
###   PURPOSE: Clean F1_1 data (list of companies)
################################################################################


### Clean F1_1 data (list of companies)
F1_1 <- as.data.frame(list_all[1]) %>%

  # keep relevant columns only
  select(RESPONDENT, RESPONDEN2) %>%

  # rename columns
  rename(ID = RESPONDENT, Company_Name = RESPONDEN2) %>%

  # merge with company_list
  right_join(company_list, by = "ID") %>%

  # remove irrelevant columns
  select(-FERC_Company_Name)
```

```
################################################################################
###   THE BRATTLE GROUP
###   PURPOSE: Clean and summarize F1_66 data (substation capacity)
################################################################################


######### Clean F1_66 data #############

### F1_66 - Substation data - Notes ###
# Business condition variables:
# - Avg substation capacity
# - Number of substations

# Convert to dataframe
F1_66 <- as.data.frame(list_all[2]) %>%

  # include companies in company_list only
  filter(RESPONDENT %in% unique(F1_1$ID))



# Keep transmission substations only. Filtering steps:

# Keep if substation characteristic contains t or T
F1_66_included <- F1_66[grepl("t|T", F1_66$SUBSTN_CHA),]

# Exclude if substation characteristic contains distribution-related names
F1_66_included <- F1_66_included[!grepl("dist|DIST|Dist|DS| D|D-", F1_66_included$SUBSTN_CHA),]

# Remove if substation characteristic = NAs
F1_66_included %<>% filter(is.na(SUBSTN_CHA) == FALSE)

# Exclude if substation characteristic contains "total"
F1_66_included <- F1_66_included[!grepl("TOTAL|Total|Tot", F1_66_included$SUBSTN_CHA),]

# Exclude if substation capacity = 0
F1_66_included %<>% filter(SUBSTN_CAP != 0)

# Exclude the following items:
F1_66_included <- subset(F1_66_included, !(SUBSTN_CHA %in% c('"   (ATTENDED)',
                                                             '(A) WHITE PLAINS "',
                                                             "14kV system",
                                                             "5  PLANT STEP-UP (A",
                                                             #"Attended",
                                                             "Gen. Plant-Attended",
                                                             "Gen. Plant-Unattend",
                                                             "Gen./Attended",
                                                             "Gen./Unattended",
                                                             "Generating Plant",
                                                             "included in total",
                                                             "INDUSTRIAL",
                                                             "Plant   Attended",
                                                             "Plant   Unattended",
                                                             "Plant  Attended",
                                                             "PLANT STEP-UP (A)",
                                                             "Production - (A)",
                                                             "Production - (U)",
                                                             "Sta. Aux. Attended",
                                                             "Switch RC-U",
                                                             "Switch U",
                                                             "Utility",
                                                             "(A) Washington St.",
                                                             "(A) Westchester Misc",
                                                             "4 INTERCONNECTION",
                                                             "CONTROLLED",
                                                             "DIstribution*",
                                                             "DMistrib. Unattended",
                                                             "Gen/Unattended",
                                                             "Generating Plant",
                                                             "Industrial",
                                                             "Manhattan",
                                                             "Plant Attended",
                                                             "Step-Down-**",
                                                             "Step-up",
                                                             "Step-Up",
                                                             "Step-Up**",
                                                             "Unit 1 Aux.",
                                                             "Unit 1 Scrubber",
                                                             "Unit 1 Start-Up",
                                                             "Unit 1 Step-Up",
                                                             "Unit 2 Aux.",
                                                             "Unit 2 Scrubber",
                                                             "Unit 2 Step-UP",
                                                             "Unit 3 Aux. A",
                                                             "Unit 3 Aux. B",
```

```
                                                    "Unit 3 Start-Up",
                                                    "Unit 4 Aux. A",
                                                    "Unit 4 Aux. B",
                                                    "Unit 4 Step-Up",
                                                    "Unit Spare",
                                                    "Unit3 Step-Up",
                                                    "Unit4 Start-Up",
                                                    "Westchester",
                                                    "(A) Washington St. .",
                                                    "(A) Westchester Misc",
                                                    "4  PLANT STEP-UP (A)",
                                                    "Didtribution - U",
                                                    "Disctribution-U",
                                                    "DIstribution - U",
                                                    "DIstribution*",
                                                    "Ditribution - U",
                                                    "DMistrib. Unattended",
                                                    "Gen/Unattended",
                                                    "Generating Plant"

)) )

#"Single Customer"
#Network
#Network - U



# Calculate business variables
F1_66_SubsData <- F1_66_included %>%
  group_by(RESPONDENT) %>%
  summarize(

    # Calculate avg. substation capacity
    avg_subs_cap = mean(SUBSTN_CAP),

    # of substations
    subs_count   = n()) %>%

  # If avg. subs. capacity > 10,000, generally units are off by a factor of 1000. Fix that:
  mutate(avg_subs_cap = ifelse(avg_subs_cap > 10000, avg_subs_cap / 1000, avg_subs_cap))
```

```
################################################################################
###   THE BRATTLE GROUP
###   PURPOSE: Clean and summarize F1_74 data (tx lines)
################################################################################


######### Clean F1_74 data #############
### F74 - Tx line data - Notes ###
# Business condition variables:
# - Avg. voltage of tx lines
# - (% of underground lines (4) U.G.)
# - Total km tx lines

# Convert to dataframe
F1_74 <- as.data.frame(list_all[3])[-1] %>% # remove first col

  # include companies in company_list only
  filter(RESPONDENT %in% unique(F1_1$ID))



# Calculate total km tx lines:
# - From 1994-1997, total found in rows where DESIGNATION == NA
# - From 1998-2018, total found in rows where DESIGNATION == "TOTAL"

if(year < 1998) {

  F1_74_TxLineLength <- F1_74 %>%

    # Find totals for each utility
    filter(is.na(DESIGNATIO) == TRUE & is.na(DESIGNATI2) == TRUE & LENGTH_DSG > 0 & is.na(STRUCTURE) == TRUE &
(is.na(VOLTAGE_OP) == TRUE | VOLTAGE_OP == 0))

} else {


  F1_74_TxLineLength <- F1_74 %>%

    # Find totals for each utility
    filter(DESIGNATIO == "TOTAL" & is.na(DESIGNATI2) == TRUE & LENGTH_DSG > 0)

}


F1_74_TxLineLength <- F1_74_TxLineLength %>%

  # Keep relevant columns (Respondent ID, Total length of Tx lines)
  select(RESPONDENT, LENGTH_DSG, LENGTH_ANO) %>%

  # remove duplicate rows
  distinct() %>%

  # sum LENGTH_DSG + LENGTH_ANO
  mutate(Total_Tx_Line_Length = LENGTH_DSG + LENGTH_ANO) %>%

  # delete unused columns
  select(-LENGTH_DSG, -LENGTH_ANO) %>%

  # remove rows with NAs in Tx_Line_Length
  filter(is.na(Total_Tx_Line_Length) == FALSE)



### if there is more than 1 row for a utility --> pick max

# sort by id and reverse of abs(value)
F1_74_TxLineLength <- F1_74_TxLineLength[order(F1_74_TxLineLength$RESPONDENT, -
abs(F1_74_TxLineLength$Total_Tx_Line_Length) ), ]

# take the first row within each id
F1_74_TxLineLength <- F1_74_TxLineLength[!duplicated(F1_74_TxLineLength$RESPONDENT), ]




### Avg. voltage of tx lines

# if voltage is in V instead of kV, then convert to kV
F1_74$VOLTAGE_OP <- ifelse(F1_74$VOLTAGE_OP > 10000,
                                    F1_74$VOLTAGE_OP / 1000,  # convert from V to kV
                                    F1_74$VOLTAGE_OP)


# Calculate avg. voltage of tx lines
F1_74_AvgVolt <- F1_74 %>%
```

```
  group_by(RESPONDENT) %>%

  # remove zeros; also excluding values < 0
  filter(VOLTAGE_OP > 0) %>%

  # Calculate avg. voltage of tx lines
  summarize(avg_volt_tx_lines = mean(VOLTAGE_OP))
```

```
### Calculate % of underground lines

# Keep if line structure description contains u or U or 4
F1_74_UG <- F1_74[grepl("U|u|4", F1_74$STRUCTURE),]

# Exclude if line structure description contains... (strings with "u" that are not "underground" related)
F1_74_UG <- F1_74_UG[!grepl("Various|various|Construction|Alu|Guy|ALUM|SUB TO SUB 1
S|Unknown|VARIOUS|CONDUIT|Included|NO STRUCTURES|Submarine|STEEL|Steel|ST|steel|1-|2-|3-|(1)|(2)|(3)|P-|Z-|B-|H-|Wood
- U|3,4", F1_74_UG$STRUCTURE),]

# Exclude if tx line length > 0
F1_74_UG %<>% filter(LENGTH_DSG > 0)


#  % of underground lines (4) U.G.
F1_74_UG_grouped <- F1_74_UG %>%

  group_by(RESPONDENT) %>%

  # calculate length of UG lines
  summarize(length_UG = sum(LENGTH_DSG))
```

```
################################################################################
###   THE BRATTLE GROUP
###   PURPOSE: Merge F1_1, F1_66, F1_74 dataframes
################################################################################


# Merge all datasets
combined <- F1_1 %>%

  # merge substation avg capacity and length
  left_join(F1_66_SubsData, by = c("ID" = "RESPONDENT")) %>%

  # merge tx line avg voltage
  left_join(F1_74_AvgVolt, by = c("ID" = "RESPONDENT")) %>%

  # merge tx line total length
  left_join(F1_74_TxLineLength, by = c("ID" = "RESPONDENT")) %>%

  # merge tx line UG length
  left_join(F1_74_UG_grouped, by = c("ID" = "RESPONDENT")) %>%


  mutate(
    # convert UG length NAs to 0s
    length_UG = ifelse(is.na(length_UG) == TRUE, 0, length_UG),

    # calculate % of underground lines
    perc_UG = length_UG / Total_Tx_Line_Length,

    # number of substations / Total km tx lines
    SubsCountPerLineKm = subs_count / Total_Tx_Line_Length,

    # add column with year
    year = year) %>%


  # merge tx line length SNL dataset
  left_join(TxLineLength_SNL, by = c("SNL_Company_Name" = "Company Name", "year" = "Year"))




# combine "combined" dataframe with other years
if(year == 1994) {

  # create a new dataframe to bind dataframes for other years
  combined_total <- combined

} else {

  # bind combined to combined_total
  combined_total <- rbind(combined, combined_total)

}
```

```
################################################################################
###   THE BRATTLE GROUP
###   PURPOSE: Manual adjustments to final dataset to account for outliers,
###   data errors or inconsistencies
################################################################################


# Gulf Power Company - adjust entries manually as data not reported in dataset but available through SNL
combined_total$avg_volt_tx_lines[combined_total$SNL_Company_Name == "Gulf Power Company" & combined_total$year ==
2015] <- 230
combined_total$Total_Tx_Line_Length[combined_total$SNL_Company_Name == "Gulf Power Company" & combined_total$year ==
2015] <- 1701
combined_total$perc_UG[combined_total$SNL_Company_Name == "Gulf Power Company" & combined_total$year == 2015] <- 0
combined_total$SubsCountPerLineKm[combined_total$SNL_Company_Name == "Gulf Power Company" & combined_total$year ==
2015] <- 16/1701


### To remove from dataset

# Wisconsin Public Service Corporation, 2001-2019
combined_total <- combined_total[!(combined_total$SNL_Company_Name == "Wisconsin Public Service Corporation" &
                                   combined_total$year %in% seq(2001,2019)),]

# Wisconsin Electric Power Company, 2001-2019
combined_total <- combined_total[!(combined_total$SNL_Company_Name == "Wisconsin Electric Power Company" &
                                    combined_total$year %in% seq(2001,2019)),]

# Toledo Edison Company, 2000-2019
combined_total <- combined_total[!(combined_total$SNL_Company_Name == "Toledo Edison Company" &
                                    combined_total$year %in% seq(2000,2019)),]

# Pennsylvania Power Company, 2000-2019
combined_total <- combined_total[!(combined_total$SNL_Company_Name == "Pennsylvania Power Company" &
                                    combined_total$year %in% seq(2000,2019)),]

# Ohio Edison Company, 2000-2019
combined_total <- combined_total[!(combined_total$SNL_Company_Name == "Ohio Edison Company" &
                                    combined_total$year %in% seq(2000,2019)),]
```

Documents déposés en version électronique

Documents de soutien 3 à 7

3 - Brattle HQT data

4 - Brattle company sample

5 - Brattle benchmarking results O&M

6 - Brattle benchmarking results capital

7 - Brattle benchmarking results total costs