Extracting ICAO Carbon Calculator Data

18 July 2022

Contents

1 Overview

Environmental Sustainability has become a priority topic over the past months. Within PBWG and the bi-lateral work, discussions revolve around establishing an initial "look-up" table for converting operational inefficiencies to associated fuel burn or CO2 estimates (or other emissions).

Publicly available documentation on fuel burn / emissions is limited. ICAO maintains an ICAO Carbon Calculator that enables interested parties to inquire the fuel burn on an aerodrome-pair level. The supporting methodology is published in (ICAO 2018).

2 Understanding Reading pdf files with R

reading in the pdf file

- pdf_text() generates a list of all pages of the pdf file

```
icc_all <- pdftools::pdf_text(pdf_fn)
```

convert table page to a machine-readable table

```
page_17 <- icc_all[17]  # extract page from list

page_17 <- page_17 %>%
  read_lines() %>%      # use line breaks
  as_tibble()           # convert to tibble

page_17
```

```
## # A tibble: 54 x 1
##    value
##    <chr>
```

```
##  1 " Appendix C: ICAO Fuel Consumption Table"
##  2 "Equivalent"
##  3 "                                                                                ~
##  4 " Aircraft"
##  5 "  Code       125     250     500     750     1000    1500    2000    2500  ~
##  6 "   100       1296    2703    3788    5129     6427    8937   11373   13757 ~
##  7 "   141       1289    2754    3874    5258     6600    9199"
##  8 "   142       1289    2754    3874    5258     6600    9199   11725"
##  9 "   143       1324    2874    4105    5621     7100    9986"
## 10 "   146       1289    2754    3874    5258     6600    9199   11725"
## # ... with 44 more rows
```

we can skip the first rows

```
skip_rows <- 4
page_17 <- page_17 %>%
  filter(row_number() > skip_rows)

page_17
```

```
## # A tibble: 50 x 1
##    value
##    <chr>
##  1 "  Code       125     250     500     750     1000    1500    2000    2500  ~
##  2 "   100       1296    2703    3788    5129     6427    8937   11373   13757 ~
##  3 "   141       1289    2754    3874    5258     6600    9199"
##  4 "   142       1289    2754    3874    5258     6600    9199   11725"
##  5 "   143       1324    2874    4105    5621     7100    9986"
##  6 "   146       1289    2754    3874    5258     6600    9199   11725"
##  7 "   310       2628    5537    7790   10759    13658   19323   24876   30356 ~
##  8 "   313       2628    5537    7790   10759    13658   19323   24876   30356 ~
##  9 "   318       1488    3016    3925    5234     6482    8931   11335   13729 ~
## 10 "   319       1596    3259    4323    5830     7271   10026   12668   15233 ~
## # ... with 40 more rows
```

clean string and split columns

```
# table has IATA Code and then ranges up-to 8500NM
col_names <- c("CODE","125","250","500","750","1000","1500","2000","2500", "3000", "3500","4000","4500"

page_17 %>%
  mutate(
     value = trimws(value)
    ,value = gsub(pattern = "\\s+", replacement = "*", x = value)
    ) %>%
  separate(
    col = value
    , into = col_names
    , fill = "right"
    )
```

```
## # A tibble: 50 x 21
##    CODE  `125` `250` `500` `750` `1000` `1500` `2000` `2500` `3000` `3500`
##    <chr> <chr> <chr> <chr> <chr> <chr>  <chr>  <chr>  <chr>  <chr>  <chr>
##  1 Code  125   250   500   750   1000   1500   2000   2500   3000   3500
##  2 100   1296  2703  3788  5129  6427   8937   11373  13757  16104  <NA>
##  3 141   1289  2754  3874  5258  6600   9199   <NA>   <NA>   <NA>   <NA>
```

```
##  4 142   1289  2754  3874  5258  6600   9199   11725  <NA>   <NA>   <NA>
##  5 143   1324  2874  4105  5621  7100   9986   <NA>   <NA>   <NA>   <NA>
##  6 146   1289  2754  3874  5258  6600   9199   11725  <NA>   <NA>   <NA>
##  7 310   2628  5537  7790  10759 13658  19323  24876  30356  35784  41172
##  8 313   2628  5537  7790  10759 13658  19323  24876  30356  35784  41172
##  9 318   1488  3016  3925  5234  6482   8931   11335  13729  16130  <NA>
## 10 319   1596  3259  4323  5830  7271   10026  12668  15233  17741  20203
## # ... with 40 more rows, and 10 more variables: `4000` <chr>, `4500` <chr>,
## #   `5000` <chr>, `5500` <chr>, `6000` <chr>, `6500` <chr>, `7000` <chr>,
## #   `7500` <chr>, `8000` <chr>, `8500` <chr>
```

3 Extracting ICAO Carbon Calculator Fuel Burn Estimates

3.1 Apendix C

wrap this into a function

```
col_names <- c("CODE","125","250","500","750","1000","1500","2000","2500", "3000", "3500","4000","4500"

extract_icc_range_table <- function(.pdf_page, .skip_rows = 0, .col_names = col_names){
  range_table <- .pdf_page %>%
    read_lines() %>%      # use line breaks
    as_tibble() %>%
    filter(row_number() > skip_rows) %>%
    mutate(
       value = trimws(value)
      ,value = gsub(pattern = "\\s+", replacement = "*", x = value)
      ) %>%
    separate(
      col = value
      , into = col_names
      , fill = "right"
      ) %>%
    # filter all rows where each range column has no value
    filter(!if_all(col_names[2]:col_names[length(col_names)], is.na))
  return(range_table)
}
```

ICC document lists fuel consumption tables on page 17 through 23

```
page_nbrs <-  17:23
fuel_stages <- page_nbrs %>%
  purrr::map_dfr(.f = ~ icc_all[.x] %>% extract_icc_range_table) %>%
  filter(CODE != "Code")
```

write out fuel_stages table for future use

write_csv(fuel_stages, "./data-analytic/ICC-FuelConsumption-v11-2018.csv")

3.2 Apendix B - Aircraft Mapping

Follow same logic to read in mapping tables. Note that there are multiple Aircraft-Equivalent mappings per page.

```
extract_aircraft_mappings <- function(.pdf_page){
  mapping <- .pdf_page %>%
```

```
    read_lines() %>%
    as_tibble() %>%
    filter(row_number() > row_number()[grepl("^Aircraft", value)]) %>%
    mutate(
         value = trimws(value)
        ,value = gsub(pattern = "\\s+", replacement = "*", x = value)
    ) %>% separate(
        col = value
        , into = c("Aircraft","Equivalent","Aircraft_2","Equivalent_2","Aircraft_3","Equivalent_3","Air
        , fill = "right"
    )
  tmp1 <- mapping[,1:2]
  tmp2 <- mapping[,3:4]; names(tmp2) =  c("Aircraft","Equivalent")
  tmp3 <- mapping[,5:6]; names(tmp3) =  c("Aircraft","Equivalent")
  tmp4 <- mapping[,7:8]; names(tmp4) =  c("Aircraft","Equivalent")
  mapping <- bind_rows(tmp1, tmp2, tmp3, tmp4) %>%
    # remove any incomplete combination / mapping (e.g. NAs, spurious page number)
    filter(!if_any(1:2, is.na))
  return(mapping)
}
```

```
page_nbrs <-  14:16
ac_equi_map <- page_nbrs %>%
  purrr::map_dfr(.f = ~ icc_all[.x] %>% extract_aircraft_mappings )
```

write_csv(fuel_stages, "./data-analytic/ICC-AircraftMapping-v11-2018.csv")

3.3 ICAO Aircraft Type Designator to IATA Type Code Mapping

The ICC documentation uses - unfortunately - IATA Type Codes. Let's hack together a mapping table on top of our handwork for the ACERT look-up.

```
# get something useful from wikipedia
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 4.0.5
```

```
##
## Attaching package: 'rvest'
```

```
## The following object is masked from 'package:readr':
##
##     guess_encoding
```

```
url <- "https://en.wikipedia.org/wiki/List_of_aircraft_type_designators"

wiki_html  <- read_html(url)
wiki_nodes <- wiki_html %>%
  html_nodes(css = "table")

# only one table in nodes, but to be sure we fetch #1
wiki_tbl   <- wiki_nodes %>%
  nth(1) %>%
  html_table()

names(wiki_tbl) <- c("ICAO", "IATA", "MODEL")
```

```
# another source
url2 <- "https://www.avcodes.co.uk/acrtypes.asp"

avcodes   <- read_html(url2)
avc_nodes <- avcodes %>%
  html_nodes("table")
avc_tbl   <- avc_nodes %>% first() %>% html_table()

names(avc_tbl) <- c("IATA", "ICAO_AVCodes", "Manufacturer-Type-Model_AVCodes", "WTC")
```

```
# another source
url3 <- "https://stringfixer.com/nl/List_of_aircraft_type_designators"

stringfix  <- read_html(url3)
sfix_nodes <- stringfix %>% html_nodes("table")
sfix_tbl   <- sfix_nodes %>% first() %>% html_table()

names(sfix_tbl) <- c("ICAO_sfix", "IATA_sfix", "MODEL_sfix")
```

combine all - for the time being

```
rq <-  wiki_tbl %>% left_join(avc_tbl %>% rename(ICAO = ICAO_AVCodes)) %>% left_join(sfix_tbl %>% renam
```

```
## Joining, by = c("ICAO", "IATA")
## Joining, by = "ICAO"
```

write_csv(rq, "./data-analytic/Aircraft-ICAO-IATA-Types.csv")

3.4 Augment ICC Fuel Consumption

Appendix B recognises "equivalent" emitting aircraft types. Thus, we have to augment the look up to account for these additional aircraft.

```
fuel_stages2 <- fuel_stages %>%
  mutate(VERSION = "ICC-v11-2018 (direct)")

which_equiv <- ac_equi_map %>%
  mutate(CHECK = Aircraft != Equivalent) %>%
  filter(CHECK == TRUE)

which_equiv <- which_equiv %>% select(-CHECK) %>%
  inner_join(fuel_stages %>% rename(Equivalent = CODE)) %>%
  mutate(VERSION = paste0("ICC-v11-2018 (equiv: ", Equivalent, ")")) %>%
  rename(CODE = Aircraft)
```

```
## Joining, by = "Equivalent"
```

```
which_equiv <- which_equiv %>%
  filter(CODE != "787")     # double accounting of 787

fuel_stages2 <- bind_rows(fuel_stages2, which_equiv) %>%
  rename(IATA = CODE, EQUIVALENT = Equivalent)
```

```
fuel_stages2 <- fuel_stages2 %>%
  left_join(avc_tbl %>% rename(ICAO = ICAO_AVCodes))
```

```
## Joining, by = "IATA"
```

```
fuel_stages2 <- fuel_stages2 %>% select(ICAO, IATA, everything())
```

```
missin_icao <- fuel_stages2 %>%
  filter(ICAO == "" | is.na(ICAO))
```

fix by hand - check that we only have unique ICAO !

> missin_icao %>% left_join(sfix_tbl, by = c("IATA"="IATA_sfix")) %>% select(ICAO_sfix, ICAO, IATA, everything()) %>% filter(!is.na(ICAO_sfix)) # A tibble: 5 x 28 ICAO_sfix ICAO IATA 125 250 500 750 1000 1500 2000 2500 3000 3500 4000 4500 5000 5500 1 A388 NA 388 5851 12016 17623 24940 32211 46695 61160 75638 90143 104681 119255 133865 148512 163196 2 A321 NA 32B 1909 3925 5270 7157 8970 12456 15818 19094 22308 NA NA NA NA NA 3 A158 NA A58 1543 3087 4064 5306 6478 NA NA NA NA NA NA NA NA NA 4 E75L NA E7W 1113 2240 2989 3953 4890 6725 NA NA NA NA NA NA NA NA 5 E50P NA EP1 290 580 764 997 1218 NA NA NA NA NA NA NA NA NA

write_csv(fuel_stages2, "./data-analytic/ICC-FuelConsumption-v11-2018-augmented.csv")

There exists already a A321, E75L. Fixed by editing the csv (outside R).

4 Conclusion

This document summarises the data preparatory steps for generating a look-up table for PBWG and the bi-regional comparison work. It is based on the extraction of fuel burn estimates from the ICAO Carbon Calculator Methodology document, v11 2018.

The data has been extracted from the respective Appendix C. With Appendix B, the lookup has been expanded for a set of aircraft types that show - in accordance with the methodology document - similar fuel burn characteristics.

The comparison work is based on ICAO Aircraft Type Designators (ICAO 2022).

== initial lookup table!

References

ICAO. 2018. "ICAO Carbon Emissions Calculation Methodology." ICAO.
———. 2022. "Aircraft Type Designators, Doc 8643."