Title

Using ADSB data for improved modelling of aircraft initial climb performance for ATC simulation.

Author

Michael Rowland, Visual Vectoring Pty Ltd

Abstract

For effective training outcomes, surveillance-based Air Traffic Control simulation requires accurate modelling of aircraft flight profiles, particularly of climb-rate performance at lower levels. The implementation of automated reporting of various flight parameters using ADSB has made available large volumes of data that may be used to check and revise existing performance models, such as those published by Eurocontrol. It is the aim of this paper to investigate the climb-rates of common jet aircraft types using data from inflight reports for comparison with published values, as well as to develop insight into the applicability of this type of data to other aspects of aircraft performance modelling. A sample of over ten million aircraft broadcasts, recorded during a 24-hour period in 2013, was filtered and arranged to generate a sequential group of reports for each of the 32,000 flights in the dataset, each representing the trajectory of an aircraft. Using data visualisation tools on these trajectories, the climb-rates of eight common aircraft types were compared with those published by Eurocontrol for these types. It was found that visualisations of ADSB data enabled an effective analysis of aircraft performance profiles in the vertical plane. Results further indicated that the Eurocontrol model overestimated the climb-rate to 5,000 feet for all types except the Boeing B744. It was concluded that the analysis of ADSB information has applicability in the modelling of climb-rates and other aspects of aircraft performance and the capability of improving the quality of Air Traffic Control simulation.

Introduction

The provision of high-fidelity Air Traffic Control (ATC) simulation training relies on accurate modelling of aircraft profiles (flight performance characteristics) to create realistic 4D trajectories (Sukhov *et al* 2005). The construction of accurate trajectories in the vertical plane presents the greatest challenge, due to the relative variability of climb and descent rates (Magill, 1996). For ATC training, it is particularly important to have correct modelling during the initial stages of flight, due to the higher density of traffic around aerodromes and increased traffic management complexity. (Gillet *et al* 2010).

Aircraft performance models are generated by either applying the physics of flight (kinetic modelling) or by using measurements of actual aircraft behaviour (kinematic modelling) (Nuic *et al*, 2005). The advent of Automatic Dependent Surveillance Broadcast (ADSB) technology provides access to vast quantities of recorded position, speed and altitude information which may be used to improve existing kinematic models, such as those published by the European Organisation for the Safety of Air Navigation (Eurocontrol).

It is the aim of this study to:

- Investigate initial climb-rates exhibited by the eight most flown passenger jet aircraft models undertaking commercial airline operations.
- Compare these rates with those published by Eurocontrol and commonly used for ATC simulation (Aircraft Performance Database. 2012) and
- Assess applicability of ADSB data to improve aircraft performance models in ATC simulation.

Data

The raw data for this observational study were provided by FlightRadar24 (FR24), a service that uses ADSB technology to record flight position and speed information (Flight Radar 24 AB, 2020). FR24 receives and records squits (individual ADSB broadcasts) from connected aircraft at intervals of approximately five seconds. For use in this paper, squits collected over a 24-hour period commencing at 12am Universal Coordinated Time (UTC) on 23 November 2013 were provided by FR23 in 16,947 Javascript Oriented Notation (JSON) files. Each file contained the received squits for a single five second period and, in total, there was a sample size of in excess of 10 million squits.

Each squit carried 18 elements, eight of which were relevant to climb profile analysis for this study or retained as they may be relevant for extended analysis in related work: The variables were:

- **Altitude** (numeric integer): the altitude of the aircraft in feet.
- **AircraftType** (nominal with multiple factors): the International Civil Aviation Organisation (ICAO) code for the aircraft model.
- **FlightNumber** (nominal with multiple factors): the identification for the flight, commonly the airline abbreviator and a numerical identifier for the flight.
- **Latitude** (numeric decimal): the latitude of the aircraft in degrees.
- **Longitude** (numeric decimal): the longitude of the aircraft in degrees.
- **Origin** (nominal with multiple factors): the code for the origin airport.
- **Speed** (numeric integer): the groundspeed of the aircraft in knots.
- **SquitDateTime** (numeric integer): the UTC time at which the squit was received.

It is noted that acquisition of ADSB data requires the carriage by the aircraft of compatible equipment and that squits are detectable by the FR24 network. Additionally, information on certain aircraft such as emergency flights and military operations may be blocked and not provided in these data.

Methods

The RStudio package (RStudio Team 2016) was used for sampling and analysis with techniques from in **R for Datascience** (Grolemund &Wickham 2020). All code is included in Appendix 1.

Pre-processing: Data-cleaning and Type Conversion

The JSON files were imported using the **rjson** package (Couture-Beil, A 2018) into individual data-frames, and then combined into a single data-frame containing all recorded squits.

Using the **dplyr** package (Wickham, H. *et al* 2019), the following steps were taken to clean and tidy the squit information:

- Variables were allocated appropriate names with the **rename()** function;
- A type conversion for **SquitDateTime,** which was provided as a date/time string, was completed. Initially a conversion to **POSIXct** type was made using the **substr()** and **as.POSIXct()** functions and the variable **TimeStamp** (numeric integer) was added for easy of reading.
- To ensure that only commercial flights were included, any squits for which there were missing or invalid values for **FlightNumber** were removed using the **filter()** function.
- Squits containing missing **AircraftType** or invalid **Speed** values were also removed.
- The unused variables were discarded using the **select()** function.

Some sample entries from the dataset following this processing are displayed in Table 1.

#	Longitude	Latitude	Altitude	Speed	Aircraft Type	TimeStamp	Origin	Flight Number
1	30.94	-96.19	23230	460	B752	1385596803	SFO	UAL1204
2	33.61	-111.90	8930	232	A321	1385596803	PHX	AWE695
3	42.55	-72.25	18200	440	DC10	1385596803	BOS	FDX1026

Table 1: First three squits from pre-processed dataset

Creating Flight Threads: Data subset selection

For construction of individual flight threads (the sequence of squits representing a single flight) The pre-processed data were organised by grouping the ordered squits by each **FlightNumber**. This was achieved by using the **dplyr group_by()** and **arrange()** functions.

To remove aircraft that were taxying, landing and taking-off, the **filter()** function and the **lead()** and **lag()** values of **Altitude** were used to remove squits for which both adjacent values were zero. This method retained the first and the last squit in a flight thread to fix the exact departure and arrival time.

In addition, for performance analysis by the phase of flight and across different airlines, three additional variables were added using the **mutate()** function. They were:

- **Company** (nominal with multiple factors), containing the airline company, derived from the **FlightNumber** for future use in comparing performance between operators.
- **FlightPhase** (ternary nominal - Climb, Cruise or Descent): a descriptor of the phase of flight when the squit was broadcast, calculated from the relative magnitude of adjacent values of **Altitude**.
- **TimeSquitting** (numeric integer): Time in seconds since the initial squit in a flight thread. Where this initial squit has an **Altitude** of zero, this variable equates to the time since departure.

A sample of the first few squits (with extra variables) for a departing flight thread is shown in Table 2. It is noted that the subject aircraft has departed within good ADSB reception given that a squit has been received from lift-off every five seconds. This may not always be the case as discussed below.

Flight Number	Company	Aircraft Type	Origin	TimeSquitting	Altitude	Speed	FlightPhase
AAL1084	AAL	B738	MIA	0	0	168	Climb
AAL1084	AAL	B738	MIA	5	80	168	Climb
AAL1084	AAL	B738	MIA	10	480	164	Climb
AAL1084	AAL	B738	MIA	15	650	164	Climb
AAL1084	AAL	B738	MIA	20	850	168	Climb

Table 2: Derived Information on American Airlines flight 1084 departing Miami airport

Viewing Flight Threads: Exploratory Visualisation

To visually explore the climb performance by type, the flight thread squits for each **FlightNumber** were filtered to include only the climbing compnent based on the **FlightPhase** variable. Flight threads for each aircraft type were plotted using **geom_line()** from the ggplot2 package (Wickham 2016). The Eurocontrol model was overlaid using the **geom_segment()** function.

Comparison of Climb-Rates: Variable transformation and Grouped Data Summarisation

To ensure that the climb-rates of each flight were accurately represented for comparison with Eurocontrol data for each altitude band (Table 3), each **FlightThread** was filtered to eliminate those that exhibited features that would indicate poor ADSB coverage, observations that could result in skewed results in the calculation of climb-rate, or that did not exhibit continuous climb.

- Too few squits over the time period within the altitude band: At least four squits were required.
- Insufficient altitude coverage: Squitted **Altitudes** range must exceed 80% of the altitude band under analysis. For example, within the altitude band 0 to 5000 feet the maximum **Altitude** squitted must exceed the lowest **Altitude** squitted by at least 4000 feet.
- Excessive time (more than two minutes) between successive squits containing similar **Altitude** values (less than 500 feet), which indicate a levelling off period for which the climb-rate is not meaningful as it is not continuous.

Altitude Band (feet)	Climb-Rate (feet per minute)							
	Airbus A319	Airbus A320	Airbus A332	Boeing B737	Boeing B738	Boeing B744	Boeing B752	Boeing B772
0 to 5000	2500	2500	2000	3000	3000	1500	3500	3000
5000 to 15000	2200	2000	2500	2500	2000	1500	2500	2500
15000 to 24000	1500	1400	2200	2500	2000	1500	2000	2000
Above 24000	1000	1000	1500	1500	1500	1500	1500	1500

Table 3: Climb performance for various aircraft models published by Eurocontrol

A linear model of altitude against time was applied to calculate a climb-rate for each aircraft type within each Eurocontrol altitude band (Table 3), using the **lm()** function with default settings. These resulting climb-rates were mapped using the ggplot2 **geom_boxplot()** function.

Results and Discussion

Pre-processing

Following the initial cleaning and filtering of the data, there was a total of 32,275 flights contained within the data, which represents about 32% of the daily global airline flights in 2013 (International Air Transport Association 2014). After filtering, more than 100 flight threads were generated for each of the eight aircraft types.

It is noted that there may be some bias in the flight thread selection because the ADSB network has best coverage around busy airports, for which operational restrictions may be imposed on the way the aircraft are flown. However, it is these airports at which ATC services are provided, the training of which requires the accurate performance modelling investigated by this study.

Flight Thread Visualisation

The blue lines in Figure 2 depict the flight thread trajectories for 116 Boeing B772 aircraft on climb to 30,000 feet. Most aircraft seem to be in continuous climb, with the straggling blue lines to the right representing those that, for some operational reason, were caused to level off at an intermediate level. The overlaid red line depicts the integrated climb trajectory as published by Eurocontrol (see Table 3). It is observed that, at lower levels, the Eurocontrol climb-rates are higher than those achieved by the sampled aircraft and that most aircraft take longer to reach 30,000 feet than predicted by Eurocontrol.

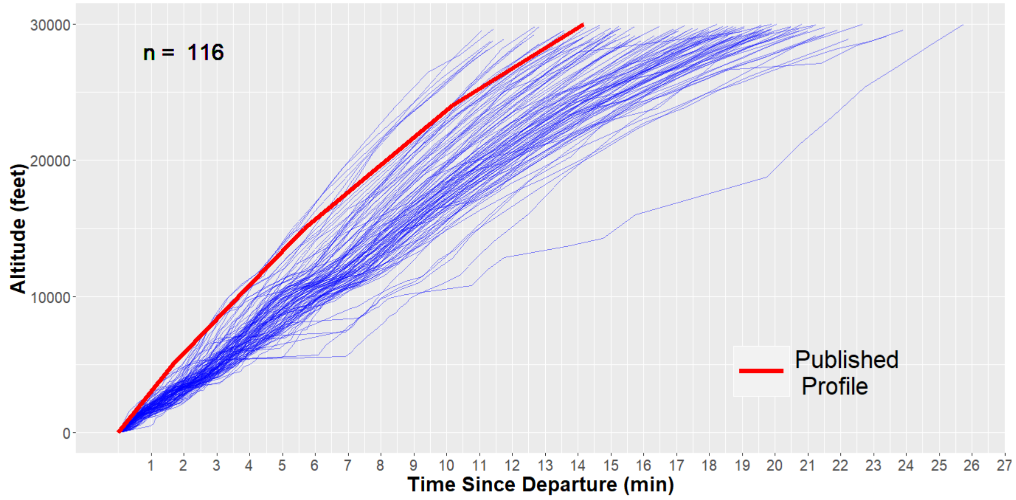

Figure 1: Boeing B772 Climb Trajectories to 30,000 feet

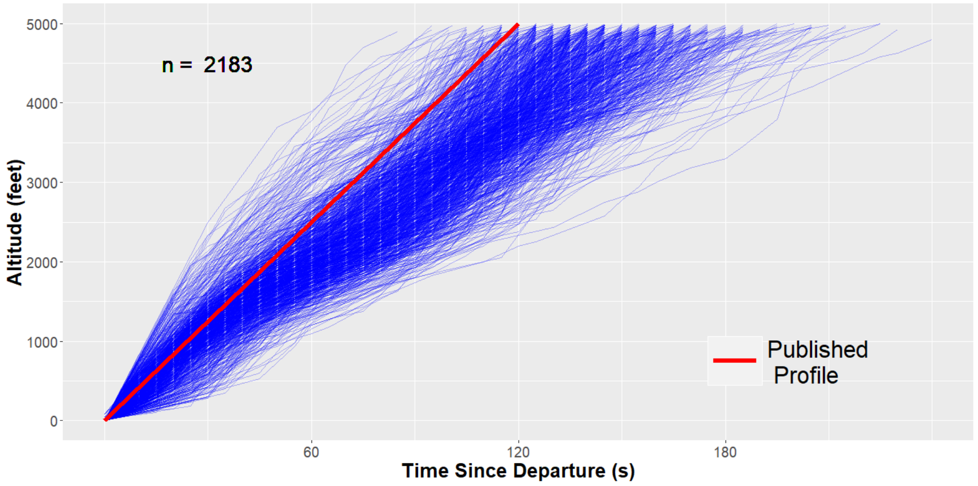

Figure 2: Airbus A320 Climb Trajectories to 5,000 feet

Focussing on the initial climb below 5,000 feet (Figure 2), it is evident that a majority of the 2183 Airbus A320 aircraft sampled did not achieve the 2,500 feet per minute rate published by Eurocontrol.

Climb-Rate Comparison

The climb-rates calculated using a linear model of squitted **Altitude** within the band below 5,000 feet is depicted in Figure 3 for each aircraft by type, along with the published value by Eurocontrol displayed as a red dot. Except for the Boeing B744 model, the Eurocontrol model overestimated and exceeded the third quartile for the climb-rate achieved for all other types. This supports the conclusion drawn from Figures 1 and 2 for the B772 and A320 types.

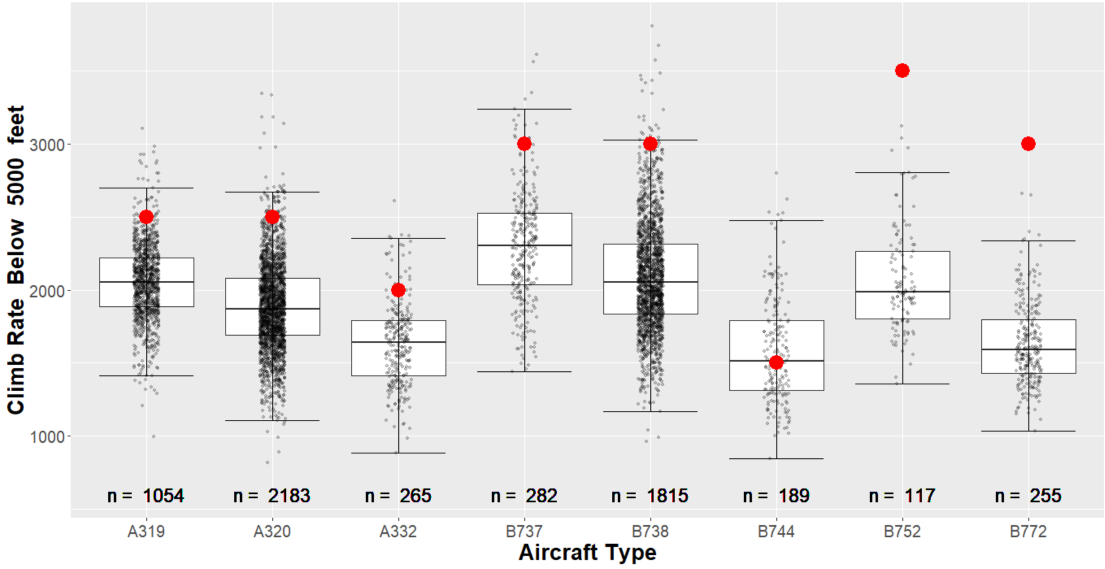

Figure 3: Climb-rate to 5,000 feet by Aircraft Type

Other Applications

Whilst Figures 1 and 2 provide an effective means of displaying trajectories, it is likely that denser plot involving a higher number of flight threads would likely diminish usefulness. Under this consideration, flight threads for climb and descent against time and distance (integrating the **Speed** variable) would be instructive.

The climb-rate analysis (Figure 3) in this study was limited to the 0 to 5,000 feet band, but initial work was done on the other bands (see Appendix 3). The analysis could be extended by further grouping by **Company** and **Origin** which would be highly useful for investigation of the potential bias in the sample and also for location specific simulation. In addition, the techniques would be highly applicable to examination of descent rates, turn rates (using the **Track** variable), airspeed and acceleration (using the **Speed, Latitude** and **Longitude** variables).

Conclusions

The construction of flight thread and climb-rate visualisations provided valuable insight into the movement of aircraft in the vertical plane and enabled a comparison between the recorded ADSB climb-rates and the published Eurocontrol values.

It was found that that the published climb-rate for climb below 5,000 feet was overestimated for seven of the eight aircraft types investigated. Given potential biases associated with the collection method relying on ADSB coverage, it would be instructive to conduct further investigations on any variability in climb-rate that might occur between airlines and between aircraft from different departure airports.

The analysis, visualisation and the underlying data manipulation techniques developed for this study have an applicability to climb-rate comparison in other altitude bands and to other areas of aircraft performance modelling, with the potential to greatly improve fidelity in ATC simulation training.

References

Aircraft Performance Database. (2012). Retrieved February 21, 2020, from https://contentzone.eurocontrol.int/aircraftperformance/

Couture-Beil, A (2018). rjson: JSON for R. R package version 0.2.20, https://CRAN.R-project.org/package=rjson

FlightRadar24 AB, (2020), Live Flight Tracker, Retrieved 20 February 2020, from https://www.flightradar24.com/about/

Gillet, S., Nuic, A., & Mouillet, V. (2010). Enhancement in realism of ATC simulations by improving aircraft behaviour models. *29th Digital Avionics Systems Conference*. doi: 10.1109/dasc.2010.5655482

Grolemund, G., & Wickham, H. (2018). R for Data Science. Retrieved February 2, 2020, from https://r4ds.had.co.nz/

International Air Transport Association. (2014). *Annual Review*. Doha.

Magill, S., (1996) On the Vertical Speeds of Airways Traffic, *Journal of Navigation*, vol 49 no. 1

Nuic, A., Poinsot, C., Iagaru, M., Gallo, E., Navarro, F., & Querejeta, C. (2005). Advanced Aircraft Performance Modeling for ATM: Enhancements to the Bada Model. *24th Digital Avionics Systems Conference*. doi: 10.1109/dasc.2005.1563320

RStudio Team (2016), RStudio: Integrated Development for R. RStudio, Inc., Boston, MA http://www.rstudio.com/

Sukhov, A., Swierstra, S., & Nuic, A. (2005) Aircraft Performance Modeling for Air Traffic Management Applications. Retrieved from http://www.atmseminar.org/seminarContent/seminar5/papers/p_039_DS.pdf

Wickham, H., François, R., Henry, L., & Müller, K. (2019). dplyr: A Grammar of Data Manipulation. R package version 0.8.3. https://CRAN.R-project.org/package=dplyr

Wickham, H. (2016) ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. https://ggplot2.tidyverse.org

Appendix 1: R Code

```r
setwd("C:/Users/micha/Dropbox/Documents/trajectorydata")
library(dplyr)
library(ggplot2)
library(tidyr)
#################### Preprocessing ####################
path <- "C:/Users/micha/Dropbox/Documents/trajectorydata "
AllSquitFiles <- list.files(path, pattern=NULL, all.files=FALSE, full.names=TRUE)
SquitDataFrameList <- list()
# Create a dataframe list (time ~ 8 hours)
for (i in seq_along(AllSquitFiles)) {
  message(i)
  SquitDataLoadedDataFrame <- fromJSON(file = AllSquitFiles[i]) %>%
    lapply(function(appliedfunction)
    {
      # Convert each group to a data frame.
      data.frame(matrix(unlist(appliedfunction), ncol=18, byrow=T))
    })
  SquitDataFrameList[[i]] <- do.call(rbind, SquitDataLoadedDataFrame)
}
#combine the list of dataframes into one (time ~ 6 hours)
SquitData_UnPreprocessed = do.call(what = rbind, args = SquitDataFrameList)
# Save dataframe for future analysis
write.csv(SquitData_UnPreprocessed, "SquitData_UnProcessed.csv", row.names = F)
# Import Dataset
Squit_Loaded <- read.csv(file.choose(), header = T, na.strings = c(""), stringsAsFactors = F)
# Explore Dataset
str(Squit_Loaded) # 10,246,187 squits
# Name variables, filter to remove NA, and select rel;evant variables
Squit_Raw <- Squit_Loaded %>%
  rename(AdsbCode = X1,Longitude = X2, Latitude = X3, Track = X4, Altitude = X5, Speed = X6,   AircraftType =
X9,Registration = X10, SquitDateTime = X11, Origin = X12,Destination = X13, FlightNumber  = X17)
# Type conversion of TimeStamp provided as YYYYMMDDHHmmSS
d <- Squit_Raw$SquitDateTime
Squit_Raw$DateString <- (as.POSIXct(paste(substr(d,1,4),"/",substr(d, 5,6), "/", substr(d, 7,8), " ", substr(d,
9,10), ":", substr(d, 11,12), ":", substr(d, 13,14), sep = ""), tz = "GMT"))
# For ease of reading, convert to integer
Squit_Raw$TimeStamp <- as.integer(Squit_Raw$DateString)
# Number of individual aircraft
Squit_Raw %>% group_by(Registration) %>% summarize() %>% nrow()
# Number of flights
Squit_Raw %>% filter(!is.na(FlightNumber)) %>% group_by(FlightNumber) %>% summarize() %>%nrow()
# calculate percentage of flights based on IATA values
IATAFlightPerDay2013 <- 99700
Squit_Raw %>% group_by(FlightNumber) %>% summarize() %>% nrow() / IATAFlightPerDay2013 * 100
# remove squits containing missing values
Squit_Raw <- Squit_Raw %>%
  filter(Speed >= 0, !is.na(FlightNumber), !is.na(AircraftType), !is.na(Origin), !is.na(Destination)) %>%
  select(AdsbCode, Longitude, Latitude, Track, Altitude, Speed, AircraftType, Registration, TimeStamp, Origin,
Destination, FlightNumber)

#################### Creating Flight Threads ####################
# Arrange squits into a flight thread and remove taxying squits
SquitData <- Squit_Raw %>% arrange(FlightNumber, TimeStamp) %>%
  group_by(FlightNumber) %>% filter (Altitude > 0 | lead(Altitude) > 0 | lag(Altitude) > 0)
# add additional variables
ClimbDescentThreshold <- 500
SquitData <- mutate(SquitData,
 AltitudeBand = as.integer(floor(Altitude/1000) * 1000), # for modelling, a band is defined
 AltitudeRate = ifelse(is.na(lag(Altitude)),(lead(Altitude) - Altitude)/(lead(TimeStamp) - TimeStamp) * 60,
 (Altitude - lag(Altitude))/(TimeStamp - lag(TimeStamp)) * 60),  # Instant altitude rate
 SquitInterval = ifelse(is.na(lag(TimeStamp)), 0,
  TimeStamp - lag(TimeStamp)), # seconds since previous squit,  # Instant altitude rate
                    TimeSquitting = TimeStamp - min(TimeStamp),
                    PreviousAltitude = lag(Altitude), # altitude from previous squit
                    Company = ifelse(is.na(FlightNumber), NA, # company from first three characters
                                      ifelse(nchar(FlightNumber) < 3, NA, substr(FlightNumber, 1, 3))),
FlightPhase = ifelse(is.na(AltitudeRate), NA,
              ifelse(AltitudeRate < -ClimbDescentThreshold, "Descent",
              ifelse(AltitudeRate > ClimbDescentThreshold, "Climb",  "Cruise")))) # Flight Phase : # Display the
```

```
initial squits for AAL1084
SquitData %>% filter(FlightNumber == "AAL1084") %>% head()

# ungroup data for further analysis
SquitData <- ungroup(SquitData)


# Convert flight phase to a factor
SquitData$FlightPhase = as.factor(SquitData$FlightPhase)

# Most prevalent companies
SquitData %>% group_by(Company) %>% summarize(N = n()) %>%
  arrange(-N) %>% select(Company, N) %>% head(20)

# Aircraft types under analysis
AircraftTypes <-  c("A320", "B738","A319", "B737", "B752", "A332", "B772", "B744")

# Store the Eurocontrol climb profiles published by Eurocontrol for each of the aircraft types
ClimbRateTo5000 <- c(2500,3000,2500, 3000, 3500, 2000, 3000, 1500)  # Rate blw 5000 for A320….
ClimbRateTo15000 <- c(2000,2000,2200, 2500, 2500, 2500, 2500, 1500) # Rate 5000-15000 for A320.
ClimbRateTo24000 <- c(1400,2000,1500, 2500, 2000, 2200, 2000, 1500) # Climb Rate 15000-24000
ClimbRateAbv24000 <- c(1000,1500,1000, 1500, 1500, 1500, 1500, 1500) # Climb Rate above 24000

# Calculate the line segments for display of the Eurocontrol data to a ceiling of 35000
x1 <- 0
y <- c(0, 5000, 15000, 24000, 30000)
# The x values are dependent on the previous climb rate/s
x2 <- x1 + (y[2] - y[1]) / ClimbRateTo5000
x3 <- x2 + (y[3] - y[2]) / ClimbRateTo15000
x4 <- x3 + (y[4] - y[3]) / ClimbRateTo24000
x5 <- x4 + (y[5] - y[4]) / ClimbRateAbv24000

cVector <- 1:8 # Loop increment
myplots <- vector('list', length = 8) # Storage for plots

# Constants for filtering
MaximumAltitude <- 30000
MaximumTimePeriod <- 60 * 30
MinimumAltitudeForFlight <- 100
MaximumAltitudeForFlight <- 28000
MaximumSquitInterval <- 200
MaximumAltitudeRate <- 5000

# Visualisarion of flight threads: Loop through for all eight types to obtain the plots
for (c in cVector)
{
  message(c)
  myplots[[c]] <- local({
    c <- c  # Ensure persistent copy of c is retained in this loop
    ThisAircraft <- AircraftTypes[c] # Text of the aircraft type
    # Generate a subset of the data for climbing aircraft of type ThisAircraft, grouped by FlightNumber
    ThisSquitData <- SquitData %>%
      filter(Altitude <= MaximumAltitude, FlightPhase == "Climb", AircraftType == ThisAircraft,
             !is.na(Company), !is.na(Origin)) %>%
      group_by(FlightNumber, Registration) %>% filter(TimeSquitting < MaximumTimePeriod) %>%
      mutate(TimeIncrementMinutes = TimeSquitting / 60, MinAltitude = min(Altitude),
             MaxAltitude = max(Altitude), IsGoAround = (Altitude > 5000  & TimeSquitting < 400 ),
             SquitIntervalClimb = ifelse(is.na(lag(TimeStamp)), 0,
                                    TimeStamp - lag(TimeStamp))) %>% # seconds since previous squit,  # Instant
altitude rate
            # IsLongSquit = (SquitInterval > 300)) %>%
      filter (MinAltitude < MinimumAltitudeForFlight, MaxAltitude > MaximumAltitudeForFlight,
              sum(SquitIntervalClimb > 300) == 0, sum(TimeSquitting < 60 & AltitudeRate >  MaximumAltitudeRate)
== 0)

    print(ThisSquitData %>% filter(Altitude>4000, TimeSquitting < 60))

    NumberFlights <- ThisSquitData %>% summarize() %>% nrow()
    currentplot <- ggplot(data = ThisSquitData, aes(x = TimeIncrementMinutes, y = Altitude, group =
FlightNumber)) +
      geom_line(alpha = 0.45, colour = "blue") +
      scale_x_continuous(breaks=1:30) +
      geom_segment(aes(x = x1, y = y[1], xend = x2[c], yend = y[2], colour = "red"), size = 2.2) +
      geom_segment(aes(x = x2[c], y = y[2], xend = x3[c], yend = y[3], colour = "red"), size = 2.2) +
      geom_segment(aes(x = x3[c], y = y[3], xend = x4[c], yend = y[4], colour = "red"), size = 2.2) +
      geom_segment(aes(x = x4[c], y = y[4], xend = x5[c], yend = y[5], colour = "red"), size = 2.2) +
      geom_text(aes(x=2, y=28000, label=paste("n = ", NumberFlights)), size = 8) +
      theme(legend.text=element_text(size=25), legend.position = c(0.8, 0.2),
            legend.background =  element_rect(fill = NA), legend.key.width = unit(2,"cm"))  +
      theme(plot.title = element_text(hjust = 0.5)) +
      theme(axis.text=element_text(size=15), axis.title=element_text(size=20,face="bold")) +
      labs(title=paste(ThisAircraft, " Climb Profile to 30,000 feet"),
           x = "Time Since Departure (min)", y = "Altitude (feet)", color = "") +
      scale_color_manual(labels = c("Published \n Profile"), values = c("red"))
    print (currentplot)
  })
}
#################### Visualisation of Climb Rates
pVector <- 1:4 # Loop increment
myPplots <- vector('list', length = 4) # Storage for plots
MinimumAltitudeBand <- c(1, 5000, 15000, 24000)
MinimumAltitudeString <- c("Below ", "5000 to ", "15000 to ", "24000 to ")
MaximumAltitudeBand <- c(5000, 15000, 24000, 30000)
NumberAircraft <- c(0, 0, 0, 0)
ClimbrateMatrix <- rbind(ClimbRateTo5000, ClimbRateTo15000, ClimbRateTo24000, ClimbRateAbv24000)
MaximumTimePeriod <- c(6, 10, 10, 10) * 60  #five minutes to reach 5000 feet
MaximumLag <- c(60, 120, 120, 120)
MinimumAltitudeCovered <- (MaximumAltitudeBand - MinimumAltitudeBand) * .8
for (p in pVector)
{
  message(p)
  myPplots[[p]] <- local({
    p <- p  # Ensure persistent copy of c is retained in this loop
    # Generate a subset of the data for climbing aircraft of type ThisAircraft, grouped by FlightNumber
    SdClimb <- SquitData %>%
      filter(Altitude < MaximumAltitudeBand[p], Altitude > MinimumAltitudeBand[p], FlightPhase == "Climb",
AircraftType %in% AircraftTypes) %>%
      group_by(FlightNumber, AircraftType, Company) %>%
      filter ((TimeStamp - min(TimeStamp)) <  MaximumTimePeriod[p], max(TimeStamp - lag(TimeStamp), na.rm = T) <
MaximumLag[p])  %>%
      summarize(MinAlt = min(Altitude), MaxAlt = max(Altitude),
                TimeInt = max(TimeStamp) - min(TimeStamp),
                ClimbRate =  60 * ((MaxAlt - MinAlt)/ TimeInt),
                ClimbRateModelled = lm(Altitude ~ TimeSquitting)$coeff[2] * 60, N = n()) %>%
      filter((MaxAlt - MinAlt) > MinimumAltitudeCovered[p], N > 4)
    for (i in 1:8)
    {
      NumberAircraft[i] <- SdClimb %>% filter(AircraftType == AircraftTypes[i]) %>% summarize() %>% nrow()
    }
      currentplot <- ggplot(SdClimb, aes(x=AircraftType, y = ClimbRateModelled)) +
      geom_boxplot(outlier.shape = NA) +
      stat_boxplot(geom ='errorbar') +
      geom_jitter(width = 0.1, alpha = 0.2, size = 0.9) +
      geom_point(aes(x="A320", y = ClimbrateMatrix[p,1]), colour="red", size = 6) +
      geom_point(aes(x="B738", y = ClimbrateMatrix[p,2]), colour="red", size = 6) +
      geom_point(aes(x="A319", y = ClimbrateMatrix[p,3]), colour="red", size = 6) +
      geom_point(aes(x="B737", y = ClimbrateMatrix[p,4]), colour="red", size = 6) +
      geom_point(aes(x="B752", y = ClimbrateMatrix[p,5]), colour="red", size = 6) +
      geom_point(aes(x="A332", y = ClimbrateMatrix[p,6]), colour="red", size = 6) +
      geom_point(aes(x="B772", y = ClimbrateMatrix[p,7]), colour="red", size = 6) +
      geom_point(aes(x="B744", y = ClimbrateMatrix[p,8]), colour="red", size = 6) +
      theme(axis.text=element_text(size=15), axis.title=element_text(size=20,face="bold")) +
      labs(x = "Aircraft Type", y = paste("Climb Rate ", MinimumAltitudeString[p], MaximumAltitudeBand[p], "
feet")) +
      geom_text(aes(x="A320", y=600, label=paste("n = ", NumberAircraft[1])), size = 6) +
      geom_text(aes(x="B738", y=600, label=paste("n = ", NumberAircraft[2])), size = 6) +
      geom_text(aes(x="A319", y=600, label=paste("n = ", NumberAircraft[3])), size = 6) +
      geom_text(aes(x="B737", y=600, label=paste("n = ", NumberAircraft[4])), size = 6) +
      geom_text(aes(x="B752", y=600, label=paste("n = ", NumberAircraft[5])), size = 6) +
      geom_text(aes(x="A332", y=600, label=paste("n = ", NumberAircraft[6])), size = 6) +
      geom_text(aes(x="B772", y=600, label=paste("n = ", NumberAircraft[7])), size = 6) +
      geom_text(aes(x="B744", y=600, label=paste("n = ", NumberAircraft[8])), size = 6)
    print (currentplot)
  })}
```

Appendix 3: Additional Plots

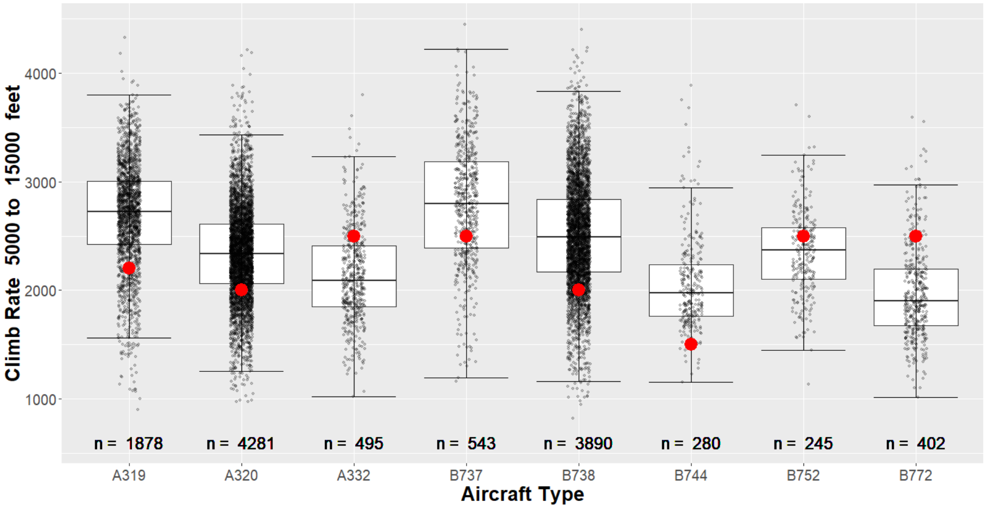

Figure A1: Climb rate 5,000 to 15,000 feet by Aircraft Type

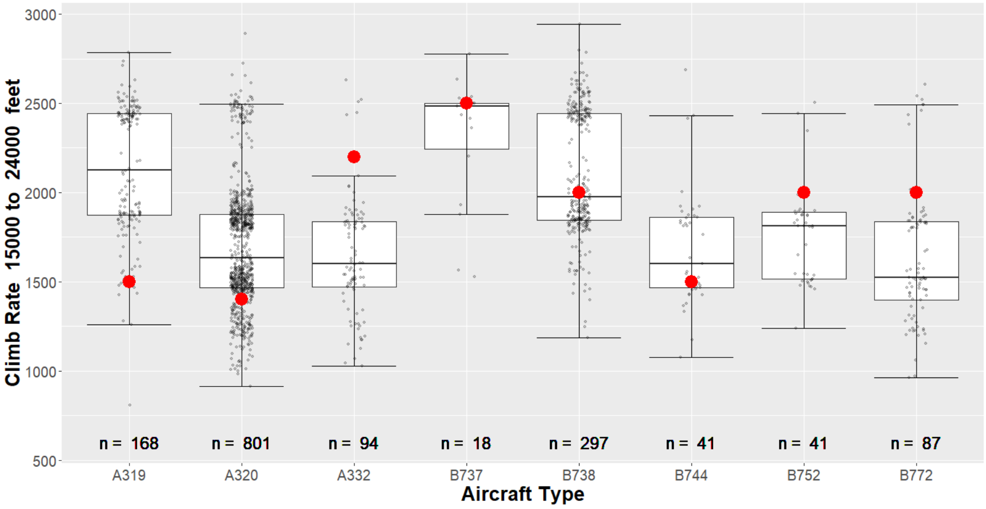

Figure A2: Climb rate 15,000 to 24,000 feet by Aircraft Type

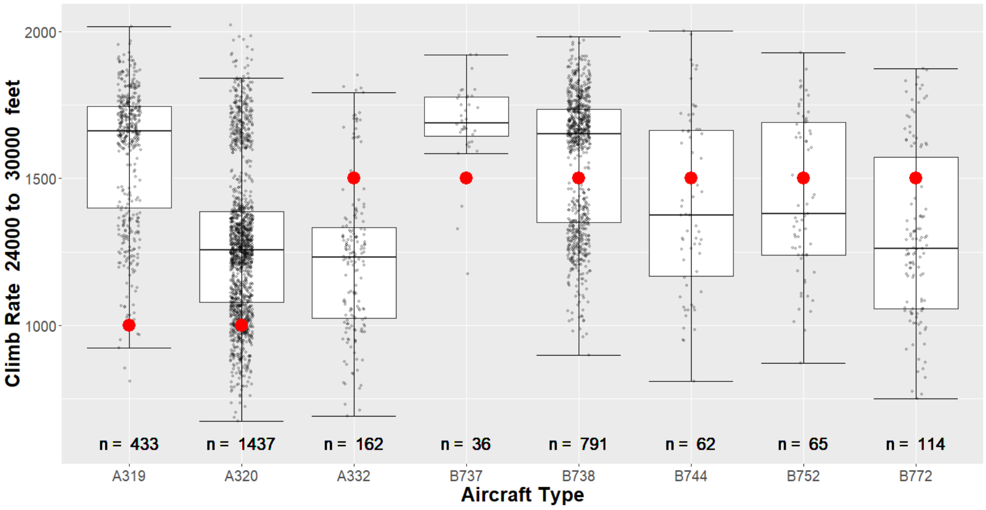

Figure A3: Climb rate above 24,000 feet by Aircraft Type