Observations of data fields

CPM

1-6

Airline designator: can find all possible airlines (EY, AA, BA, CX…)

Flight number: any number

Date: 1<=date<=31; not sure about month, year, and GMT

Aircraft registration:

1. The first 1-3 characters represent country codes. We can find all possible cases. See [Wikipedia](https://en.wikipedia.org/wiki/List_of_aircraft_registration_prefixes).
2. The rest vary across countries, so it’s hard to find any rules. For UAE: format is A6+AAA-ZZZ

Departure/destination/IMP stations: can find all possible airports.

7. ULD information

ULD bay designation:’

1. Each line has a unique code; repeating code is very likely an error.
2. Typically takes the form of X or X+L/R/P, where X is a letter (A-Z except I&O) or number (1-99). Codes containing “I,O” or ending with a letter other than “L,P,R” are likely to be incorrect.
3. L and R simply means left and right; therefore, two lines

ULD type code:

1. First letter (type): cannot be “I,O,T”; most common ones include “A,D,M,P,Q,R”
2. Second letter (base size): cannot be “C,D,I,O,T,U”; might be a number
3. Third letter (contour size): cannot be “I,O,Q,R,S,T”
4. Relationship between the 3-letter code and ULD positions: see the spreadsheet

\*Note: letter 2 is the most important factor to determine ULD positions. Blank cells mean it can be any letter. The data is gathered from multiple unofficial websites, which contain some discrepancy between each other, so this chart represents the data that most websites agree with and can be used in our project.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ULD pos | letter 1 | letter 2 | letter 3 | CONCAT | Max gross weight/kg | Max tare weight/kg |
| LD2 |  | P | E | ?PE | 1225 | 92 |
| LD1 |  | V |  | ?V? | 1588 | 170 |
| A | K | C | AKC |
| LD3 | X | C | XKC | 1588 | 181 |
|  | E | ?KE |
|  | N | ?KN |
| LD3-45 |  | H | ?KH |
|  | W | ?KW |
| LD9 |  | A | P | ?AP | 4625 |  |
| LD26 |  | F | ?AF | 4625 |  |
| LD29 |  | U | ?AU | 4625 |  |
| LD7 |  | A | ?AA | 4626 | 105 |
|  | D | ?AD |
|  | Y | ?AY |
|  | Z | ?AZ |
|  | W | ?AW |
| P | D | PAD |
| P | .number. | P | P.num.P |
| P | M | C | PMC |
|  | J | ?MJ |
| LD39 |  | U | ?MU | 5100 |  |
| M1 | A | A | AMA | 6804 |  |
| M1H |  | D | ?MD | 6800 |  |
| LD6 | A | W | C | AWC | 3175 | 230 |
| LD11 | R | B | RWB | 3175 |  |
|  | L | P | ?LP |  |
|  | A | ?LA |  |
| LD6 |  | F | ?LF | 3175 | 230 |
| LD4 | A | B | ALB | 2449 | 120 |
| M2 | A | G |  | AG? |  |  |
| M6 | P |  | PG? |  |  |
| V |  |  | V?? |  |  |
| LD8 |  | Q |  | ?Q? | 2449 |  |
| Special Positions | | | | |  |  |
| Demi | S |  |  | S?? |  |  |
|  | Y |  | ?Y? |  |  |
| Horse stall | H | M | A | HMA | 3500 | 1310 |
| Sheep and goat pen | K | A | KMA |  |  |

Weight: Each ULD type has a maximum weight. We might detect errors if the weight is too much, but might not be able to suggest any correction.

Load types: can find some typical values [(codes includes letter=load type)] [C=cargo, B=baggage, X=empty…]

Volume code: 0-9

Pallet and contour code format: P1P/P6P + Q6/Q7/X5 (2\*3=6 possible cases allowed by “aaa/mm”)

e.g., P1P/Q6 and P6P/X5; Condition: only lines of ULD types starting with P can have this value!

8.Bulk load information:

Designation of compartment: 0-99

8.2-8.5 included in the above section

Number of pieces: PCS+0-9999

Available volume information: VRf

Supplementary information: SI

Rules for auto-correction:

1. Only suggest correction for the data which we can find all possibilities (airports, airlines, ULD types, etc.). Otherwise, tell the user of why data has error (invalid format etc., date>31, etc.)
2. For the data being almost identical across each line of message (e.g., unloading stations, IMP codes, etc.,), list each one that appears. These are more likely the correct cases compared to others when correcting error.

Therefore, when there are multiple versions of corrected data, the one appearing the list takes precedence. However, do let the users aware of other possible corrections.

1. For data of fixed length (airports, airlines, etc.):
   1. If the error data is the correct length:
      1. Switch each pair of adjacent characters (FJK >> JFK/FKJ; YE >> EY). If some of these match possible cases, they might be correct. Otherwise:
      2. Compute the difference (as Rule 5) between the error data and possible cases. If one case is significantly more similar than others, it might be correct.
   2. If the error data is one character longer: consider all possible combinations of cutting one character off (JFKL >> JFK/JFL/JKL/FKL). If some match possible cases, they might be correct. Otherwise, don’t auto-correct.
   3. If the error data is one character shorter: if every character appear in order in some possible cases (JK/JF/FK >> JFK), they might be correct. Otherwise, don’t auto-correct.
   4. If the data is too long/short, or if there’s no reasonably close match, don’t correct them
2. For data of various length (load category codes, etc.):

Directly compute the difference (as Rule 5)

1. Possible ways of computing difference in characters:
   1. Ignore keyboard locations. If two string is the same length, count the number of differences in characters. Otherwise, the missing or redundant character counts 1 difference. (JFK-JFL=1; JFK-JFKJ=1; JFK-JF=1)
   2. Assign 2D coordinates for each key on keyboards. For strings of same length, compute the average square difference in coordinates corresponding to each character. Otherwise, the character too long or too short counts maximal difference in coordinates.

To be continued. Please suggest improvements!

**Given data**  Falling within possible

cases/reasonable ranges?

If not

Does the field contain

fixed-length (n characters) values?

Yes No

Is given data also

of length n? How many characters

No longer/shorter

Yes

1 -1 +/-n|n>1

Swapping each adjacent character pairs,

can it match any correct value?

Deleting each character at a time, Can the n-1 partially match

can the rest match any of the any correct values?

correct values?

Compute the “distance” between given data and

other possible values

Find k nearest neighbors. Is the closest match significantly

closer than the other k-1?

Has the possible correct value appeared

many times elsewhere?

How to deal with the error?

Keep the error and Suggest several possible Auto-correct the error

warn the user corrections