**R Skills Test**

Create an R markdown within the ‘R Skills Test’ folder and name it ‘r\_skills\_answers.Rmd’. The rest of the tasks below should be completed in this markdown. For each task write R code that performs the necessary function. Clearly label each chunk with the task number. If you would like, briefly annotate where you feel necessary.

1. Read in the ‘lineages.csv’ file from the ‘lineages’ folder. Assign to ‘lineages\_df’.
2. Read in all csv files within the 'submissions' folder and bind them into a single dataframe. Assign to ‘submissions\_df’. Remove any empty rows if present. Assign to ‘submissions\_df\_0.
3. Transform the date values to a mm/dd/yyyy format. Account for any invalid dates. Assign to ‘submissions\_df\_1’
   1. Note: If the 4-digit year is at the end assume it is month followed by day then year. If the 4-digit year is at the beginning assume it is year followed by month then day
   2. Note: Any 5-digit numeric string is an excel serialized date. This should also be converted
4. Clean and transform the name values by performing the following and assign to ‘submissions\_df\_2’:
   1. Uppercase
   2. Remove whitespace
   3. Remove special characters
5. Merge ‘submissions\_df’ with ‘metadata\_df’ on the ‘LAB\_ACCESSION\_ID’ column. Ensure you keep all the data from ‘submissions\_df’. Assign to ‘merged\_df’.
6. Check whether the values within ‘SEQUENCE\_REASON’, ‘SEQUENCE\_STATUS’ and ‘PANGO\_LINEAGE’ columns of ‘merged\_df’ are valid. Refer to the data dictionary below for a list of valid values for these columns. Flag records with invalid values as you see fit.
7. Subset any records in ‘merged\_df’ that were flagged. Save these in a dataframe and write the dataframe to a csv file named ‘flag.csv’ in the output folder.
8. Save the final dataset (flagged records removed) as a csv called ‘final.csv’ in the output folder.
9. Create a summary table or plot of the final dataset. You can show or categorize whatever you find interesting.

Bonus Questions:

1. Output/knit the rmarkdown file to an html output to display your code and outputs below the text of each question.
2. In ‘merged\_df’ there was a record that was not able to be matched based on the LAB\_ACCESSION\_ID. Demonstrate how you would use fuzzy matching and the demographics such as date of birth to match instead.

**Data Dictionary:**

**submissions**

|  |  |  |
| --- | --- | --- |
| **Column** | **Description** | **Values** |
| LAB\_ACCESSION\_ID | Accession ID | - |
| GISAID\_ID | External Repository ID | - |
| SPECIMEN\_COLLECTION\_DATE | Date of collection | - |
| SUBMITTING\_LAB | Lab performing sequencing | - |
| SEQUENCE\_REASON | Reason for sequencing | SENTINEL SURVEILLANCE, OUTBREAK, CLINICAL |
| SEQUENCE\_STATUS | Status of sequencing | COMPLETE, FAILED |
| PANGO\_LINEAGE | Variant identified | Note: check against ‘lineage\_extract’ column within lineages.csv |
| FIRST\_NAME | Patient first name | - |
| LAST\_NAME | Patient last name | - |
| MIDDLE\_NAME | Patient middle name | - |
| DOB | Patient date of birth | - |
| ALTERNATIVE\_ID | Other ID used | - |

**patient\_metadata**

|  |  |  |
| --- | --- | --- |
| **Column** | **Description** | **Values** |
| CASE\_ID | Event ID | - |
| LAB\_ACCESSION\_ID | Accession ID | - |
| FIRST\_NAME | Patient first name | - |
| LAST\_NAME | Patient last name | - |
| DOB | Patient date of birth | - |

**lineages.csv**

|  |  |  |
| --- | --- | --- |
| **Column** | **Description** | **Values** |
| lineage\_extracted | SARS-CoV-2 Lineage/Variant | **-** |
| description | Origin information | **-** |
| status | Whether lineage is still in use | Active, Withdrawn |