Data Cleanse

RISK ID Column within the RISK and MITIGATION data sets

Find and Replace a number of scenarios:

Replace *dash* with (nothing)

Replace RISKRISK with Risk

Replace Risk with RISK

1st column, create the standard length of ID

=IF(LEN(F2)=7, F2, "RISK" & TEXT(MID(F2,5,LEN(A1)-4), "000"))

2nd column, make sure there are no spaces in error in the cells

=TRIM(F2)

3rd Column Add in the leading zeros if not already in the ID number

=IF(ISNUMBER(VALUE(F2)), "RISK" & TEXT(VALUE(F2), "000"), F2)

4th column amend any IDs that do not start with the prefix RISK

=IF(AND(LEFT(A1, 4)="RISK", LEN(A1)=7), A1, "RISK" & TEXT(VALUE(MID(A1,2,LEN(A1)-1)), "000"))

"RISK052" will remain unchanged because it already has "RISK" and the correct number of digits.

"R8" will become "RISK008"

"R72" will become "RISK072"

"R181" will remain "RISK181"

Copy and paste values for this column to use as the RISK ID can then delete the columns not required.

For best results a Python script could be ran to cleanse the entire data set:

import pandas as pd

import re

from spellchecker import SpellChecker

# Load the Excel file into a pandas DataFrame

def clean\_excel\_data(file\_path, output\_file):

# Load the spreadsheet

df = pd.read\_excel(file\_path)

# Initialize spell checker (optional: if you want to fix spelling errors automatically)

spell = SpellChecker()

# Function to clean individual entries in the DataFrame

def clean\_entry(entry):

if isinstance(entry, str):

# 1. Remove hyphens or any unwanted symbols

entry = entry.replace('-', '').replace('–', '') # Add any other symbols you want to remove

# 2. Split words that are not separated by a space (i.e., "wordword" -> "word word")

entry = re.sub(r'([a-z])([A-Z])', r'\1 \2', entry)

# 3. Correct spelling errors (optional, can be slow for large datasets)

# This can be customized further for specific word lists or custom dictionary

words = entry.split()

corrected\_words = [spell.correction(word) for word in words]

entry = ' '.join(corrected\_words)

return entry

# Apply the cleaning function to each cell in the DataFrame

df\_cleaned = df.applymap(clean\_entry)

# Save the cleaned DataFrame to a new Excel file

df\_cleaned.to\_excel(output\_file, index=False)

print(f"Data cleaned and saved to {output\_file}")

# Example usage

file\_path = 'input\_file.xlsx' # Path to your Excel file

output\_file = 'cleaned\_output.xlsx' # Path to save the cleaned file

clean\_excel\_data(file\_path, output\_file)