My deliverable presentation

Functions, Risk Scoring, Tasks
Tue, 2020-11-16
David

Agenda

D1 Functions

D2 Scoring

D3 Tasks

D1
Functions

sex = scoring explanation

	type	score_name	score_question	score_explanation
0	cis	r_score_1	Q2.1:'Please list the most significant climate...	1: less than 2 two hazards, 2: less than 5, 3:...
1	cis	r_score_2	Q2.1:'Please list the most significant climate...	1: less than 1 two hazards, 2: less than 2, 3:...
2	cis	r_score_3	Q2.1:'Please list the most significant climate...	mapped values for every single risk from 1 to ...
3	cis	r_score_4	Q2.1:'Please list the most significant climate...	mapped values for every single risk from 1 to ...
4	cos	r_score_1	QC2.3:'Have you identified any inherent climat...	1: no inherent risk, 5: inherent risk identified
5	cos	r_score_2	QC2.3a:'Provide details of risks identified wi...	mapped values for every single risk from 1 to ...
6	cos	r_score_3	QC2.3a:'Provide details of risks identified wi...	mapped values {"Transition risk": 0, "Physical...

	account_number	year	r_score_1	r_score_2	r_score_3	r_score_4	r_score_total
0	1093	2019	3.0	1.0	4.0	4.0	3.00
1	1184	2019	2.0	3.0	5.0	5.0	3.75
2	1184	2020	2.0	3.0	5.0	5.0	3.75
3	1499	2019	4.0	5.0	4.0	4.0	4.25
4	1499	2020	4.0	5.0	4.0	4.0	4.25
...	...	...	...	...	...	...	...
1289	848409	2020	1.0	2.0	4.0	4.0	2.75
1290	848474	2020	4.0	3.0	2.0	1.0	2.50
1291	848476	2020	2.0	3.0	5.0	5.0	3.75
1292	848478	2020	1.0	2.0	3.0	3.0	2.25
1293	73762	2019	NaN	NaN	1.0	1.0	1.00

cis = cities scores

	account_number	year	r_score_1	r_score_2	r_score_3	r_score_total
0	58	2018	5	NaN	5.0	5.0
1	58	2019	5	NaN	5.0	5.0
2	58	2020	5	1.0	NaN	3.0
3	64	2018	1	NaN	NaN	1.0
4	64	2019	1	NaN	NaN	1.0
...	...	...	...	...	...	...
2550	848215	2020	5	1.0	NaN	3.0
2551	848284	2020	5	2.0	NaN	3.5
2552	848285	2020	5	2.0	NaN	3.5
2553	848471	2020	5	3.0	NaN	4.0
2554	848541	2020	5	5.0	NaN	5.0

cos = corporates scores

```python
df = cor.copy()
e_type = "cos"                # cis or cos
score = "r_score_1"           # name of score
base_question = "C2.3"        # question to base score on
base_column = 0               # question to base score on

# select rows and get responses from dataframe
data = df.copy().query('question_number == @base_question & column_number == @base_column')

# provide corrosponding question context as variabel and output
q_string = print_question(data, base_question, [base_column])

# calculate scoring
response_map = {"No": 1, "Yes" : 5}                                    # set mapping dictionary
data[score] = data.response_answer.map(response_map)                   # map values to df
gob = data.groupby(["account_number", "year"], as_index=False)[score].sum()  # group multianswers
gob = gob.loc[:,["account_number", "year", score]]                     # select relevant columns

# add score to dataframe
cos = pd.merge(left=cos, right=gob, on=["account_number", "year"], how="outer")

# add explanation to dataframe
question = pd.Series(data=[
                        e_type,
                        score,
                        q_string,
                        "1: no inherent risk, 5: inherent risk identified"],
                    index=sex.columns)
sex = sex.append(question, ignore_index=True)
```

Risk Scoring

1. Cities
    a. Number of identified hazards
    b. Significant impact of hazards
    c. Current probability of hazards
    d. Current magnitude of hazards
2. Corporates
    a. Inherent risk identified
    b. Risk type mapping
    c. Transitional / physical risk

=> Results as uploaded to GitHub yesterday

D3

Todays Tasks

My Task Schedule

1. Update Risk Scoring / Bugfixing
2. Optional: Create Social or Opportunities Risk Scoring
3. t-SNE approach