TA-on-Norwegian-Reports

May 26, 2022

[1]:
```
## Importing addons needed for our textual analysis.
```

[2]:
```
import PyPDF2   # PDF manipulation and editing
```

[3]:
```
import re       # Regular Expressions
```

[4]:
```
import sys      # Manipulate different parts of the Python runtime environment.
```

[5]:
```
import tabulate # Addon used for table creation
```

[6]:
```
import nltk     # Natural Language Processing Toolkit
```

[ ]:
```

```

[7]:
```
## Importing sub-addons needed for textual analysis.
```

[8]:
```
from tabulate import tabulate
```

[9]:
```
from PyPDF2 import PdfFileReader, PdfFileWriter
```

[10]:
```
from PyPDF2 import PdfFileMerger
```

[11]:
```
from nltk.corpus import wordnet
```

[12]:
```
from nltk.corpus import stopwords
```

[13]:
```
from nltk.probability import FreqDist
```

[14]:
```
from nltk.tokenize import sent_tokenize
```

[15]:
```
from nltk.tokenize import word_tokenize
```

[ ]:
```

```

[16]:
```
# Merging annual- and sustainability reports (for the key performance␣
↪indicators)
```

[ ]:
```

```

[17]:
```
#pdfs = [YOUR PATH", "YOUR PATH"]
```

[18]:
```
#merger = PdfFileMerger()
```

[19]:
```
#for pdf in pdfs:
 #   merger.append(pdf)
```

[20]:
```
#merger.write(r"YOUR PATH")
#merger.close()
```

[ ]:
```

```

[21]:
```
## Importing all annual reports (Aker ASA example)
```

[22]:
```
file2010 = open("/YOUR PATH/2010.pdf", "rb")
```

[23]:
```
file2011 = open("/YOUR PATH/2011.pdf", "rb")
```

[24]:
```
file2012 = open("/YOUR PATH/2012.pdf", "rb")
```

[25]:
```
file2013 = open("/YOUR PATH/2013.pdf", "rb")
```

[26]:
```
file2014 = open("/YOUR PATH/2014.pdf", "rb")
```

[27]:
```
file2015 = open("/YOUR PATH/2015.pdf", "rb")
```

[28]:
```
file2016 = open("/YOUR PATH/2016.pdf", "rb")
```

[29]:
```
file2017 = open("/YOUR PATH/2017.pdf", "rb")
```

[30]:
```
file2018 = open("/YOUR PATH/2018.pdf", "rb")
```

[31]:
```
file2019 = open("/YOUR PATH/2019.pdf", "rb")
```

[32]:
```
file2020 = open("/YOUR PATH/2020.pdf", "rb")
```

[ ]:
```

```

[33]:
```
## Importing annual reports
```

[34]:
```
filetext2010 = PyPDF2.PdfFileReader(file2010)
```

[35]:
```
filetext2011 = PyPDF2.PdfFileReader(file2011)
```

[36]:
```
filetext2012 = PyPDF2.PdfFileReader(file2012)
```

[37]:
```
filetext2013 = PyPDF2.PdfFileReader(file2013)
```

[38]:
```
filetext2014 = PyPDF2.PdfFileReader(file2014)
```

[39]:
```
filetext2015 = PyPDF2.PdfFileReader(file2015)
```

[40]:
```
filetext2016 = PyPDF2.PdfFileReader(file2016)
```

[41]:
```
filetext2017 = PyPDF2.PdfFileReader(file2017)
```

[42]:
```
filetext2018 = PyPDF2.PdfFileReader(file2018)
```

```
PdfReadWarning: Xref table not zero-indexed. ID numbers for objects will be
corrected. [pdf.py:1736]
```

[43]:
```
filetext2019 = PyPDF2.PdfFileReader(file2019)
```

[44]:
```
filetext2020 = PyPDF2.PdfFileReader(file2020)
```

[ ]:
```

```

[45]:
```
## Counting the number of pages in each annnual report
```

[46]:
```
pages2010 = filetext2010.numPages
```

[47]:
```
pages2011 = filetext2011.numPages
```

[48]:
```
pages2012 = filetext2012.numPages
```

[49]:
```
pages2013 = filetext2013.numPages
```

[50]:
```
pages2014 = filetext2014.numPages
```

[51]:
```
pages2015 = filetext2015.numPages
```

[52]:
```
pages2016 = filetext2016.numPages
```

[53]:
```
pages2017 = filetext2017.numPages
```

[54]:
```
pages2018 = filetext2018.numPages
```

[55]:
```
pages2019 = filetext2019.numPages
```

[56]:
```
pages2020 = filetext2020.numPages
```

[ ]:
```

```

[57]:
```
## Converting all PDFs into readable format
```

[58]:
```
count2010 = 0
```

[59]:
```
count2011 = 0
```

[60]:
```
count2012 = 0
```

[61]:
```
count2013 = 0
```

[62]:
```
count2014 = 0
```

[63]:
```
count2015 = 0
```

[64]:
```
count2016 = 0
```

[65]:
```
count2017 = 0
```

[66]:
```
count2018 = 0
```

[67]:
```
count2019 = 0
```

[68]:
```
count2020 = 0
```

[ ]:
```

```

[69]:
```
text2010 = ""
```

[70]:
```
text2011 = ""
```

[71]:
```
text2012 = ""
```

[72]:
```
text2013 = ""
```

[73]:
```
text2014 = ""
```

[74]:
```
text2015 = ""
```

[75]:
```
text2016 = ""
```

[76]:
```
text2017 = ""
```

[77]:
```
text2018 = ""
```

[78]:
```
text2019 = ""
```

[79]:
```
text2020 = ""
```

[ ]:
```

```

[80]:
```
while count2010 < pages2010:
    pageObj2010 = filetext2010.getPage(count2010)
```

```
    count2010 +=1
    text2010 +=pageObj2010.extractText()
```

[81]:
```
while count2011 < pages2011:
    pageObj2011 = filetext2011.getPage(count2011)
    count2011 +=1
    text2011 +=pageObj2011.extractText()
```

[82]:
```
while count2012 < pages2012:
    pageObj2012 = filetext2012.getPage(count2012)
    count2012 +=1
    text2012 +=pageObj2012.extractText()
```

[83]:
```
while count2013 < pages2013:
    pageObj2013 = filetext2013.getPage(count2013)
    count2013 +=1
    text2013 +=pageObj2013.extractText()
```

[84]:
```
while count2014 < pages2014:
    pageObj2014 = filetext2014.getPage(count2014)
    count2014 +=1
    text2014 +=pageObj2014.extractText()
```

[85]:
```
while count2015 < pages2015:
    pageObj2015 = filetext2015.getPage(count2015)
    count2015 +=1
    text2015 +=pageObj2015.extractText()
```

[86]:
```
while count2016 < pages2016:
    pageObj2016 = filetext2016.getPage(count2016)
    count2016 +=1
    text2016 +=pageObj2016.extractText()
```

[87]:
```
while count2017 < pages2017:
    pageObj2017 = filetext2017.getPage(count2017)
    count2017 +=1
    text2017 +=pageObj2017.extractText()
```

[88]:
```
while count2018 < pages2018:
    pageObj2018 = filetext2018.getPage(count2018)
    count2018 +=1
    text2018 +=pageObj2018.extractText()
```

[89]:
```
while count2019 < pages2019:
    pageObj2019 = filetext2019.getPage(count2019)
    count2019 +=1
    text2019 +=pageObj2019.extractText()
```

[90]:
```
while count2020 < pages2020:
    pageObj2020 = filetext2020.getPage(count2020)
    count2020 +=1
    text2020 +=pageObj2020.extractText()
```

[ ]:
```

```

[91]:
```
if text2010 !="":
    text2010 = text2010
```

[92]:
```
if text2011 !="":
    text2011 = text2011
```

[93]:
```
if text2012 !="":
    text2012 = text2012
```

[94]:
```
if text2013 !="":
    text2013 = text2013
```

[95]:
```
if text2014 !="":
    text2014 = text2014
```

[96]:
```
if text2015 !="":
    text2015 = text2015
```

[97]:
```
if text2016 !="":
    text2016 = text2016
```

[98]:
```
if text2017 !="":
    text2017 = text2017
```

[99]:
```
if text2018 !="":
    text2018 = text2018
```

[100]:
```
if text2019 !="":
    text2019 = text2019
```

[101]:
```
if text2020 !="":
    text2020 = text2020
```

[ ]:
```

```

[102]:
```
# Tokenizing: splitting up the entire text into tokens
```

[103]:
```
tokenized2010 = word_tokenize(text2010)
```

[104]:
```
tokenized2011 = word_tokenize(text2011)
```

[105]:
```
tokenized2012 = word_tokenize(text2012)
```

[106]:
```
tokenized2013 = word_tokenize(text2013)
```

[107]:
```
tokenized2014 = word_tokenize(text2014)
```

[108]:
```
tokenized2015 = word_tokenize(text2015)
```

[109]:
```
tokenized2016 = word_tokenize(text2016)
```

[110]:
```
tokenized2017 = word_tokenize(text2017)
```

[111]:
```
tokenized2018 = word_tokenize(text2018)
```

[112]:
```
tokenized2019 = word_tokenize(text2019)
```

[113]:
```
tokenized2020 = word_tokenize(text2020)
```

[ ]:
```

```

[114]:
```
# Setting the stopwords to English vocabulary
```

[115]:
```
sw = set(stopwords.words("english"))
```

[ ]:
```

```

[116]:
```
# Removing stop words
```

[117]:
```
filt_text2010 = [w for w in tokenized2010 if not w.lower() in sw]
```

[118]:
```
filt_text2011 = [w for w in tokenized2011 if not w.lower() in sw]
```

[119]:
```
filt_text2012 = [w for w in tokenized2012 if not w.lower() in sw]
```

[120]:
```
filt_text2013 = [w for w in tokenized2013 if not w.lower() in sw]
```

[121]:
```
filt_text2014 = [w for w in tokenized2014 if not w.lower() in sw]
```

[122]:
```
filt_text2015 = [w for w in tokenized2015 if not w.lower() in sw]
```

[123]:
```
filt_text2016 = [w for w in tokenized2016 if not w.lower() in sw]
```

[124]:
```
filt_text2017 = [w for w in tokenized2017 if not w.lower() in sw]
```

[125]:
```
filt_text2018 = [w for w in tokenized2018 if not w.lower() in sw]
```

[126]:
```
filt_text2019 = [w for w in tokenized2019 if not w.lower() in sw]
```

[127]:
```
filt_text2020 = [w for w in tokenized2020 if not w.lower() in sw]
```

[ ]:
```

```

[128]:
```
filt_text2010 = []
```

[129]:
```
filt_text2011 = []
```

[130]:
```
filt_text2012 = []
```

[131]:
```
filt_text2013 = []
```

[132]:
```
filt_text2014 = []
```

[133]:
```
filt_text2015 = []
```

[134]:
```
filt_text2016 = []
```

[135]:
```
filt_text2017 = []
```

[136]:
```
filt_text2018 = []
```

[137]:
```
filt_text2019 = []
```

[138]:
```
filt_text2020 = []
```

[ ]:
```

```

[139]:
```
for w in tokenized2010:
    if w not in sw:
        filt_text2010.append(w)
```

[140]:
```
for w in tokenized2011:
    if w not in sw:
        filt_text2011.append(w)
```

[141]:
```
for w in tokenized2012:
    if w not in sw:
        filt_text2012.append(w)
```

[142]:
```
for w in tokenized2013:
    if w not in sw:
        filt_text2013.append(w)
```

[143]:
```
for w in tokenized2014:
    if w not in sw:
        filt_text2014.append(w)
```

[144]:
```
for w in tokenized2015:
    if w not in sw:
        filt_text2015.append(w)
```

[145]:
```
for w in tokenized2016:
    if w not in sw:
        filt_text2016.append(w)
```

[146]:
```
for w in tokenized2017:
    if w not in sw:
        filt_text2017.append(w)
```

[147]:
```
for w in tokenized2018:
    if w not in sw:
        filt_text2018.append(w)
```

[148]:
```
for w in tokenized2019:
    if w not in sw:
        filt_text2019.append(w)
```

[149]:
```
for w in tokenized2020:
    if w not in sw:
        filt_text2020.append(w)
```

[ ]:
```

```

[150]:
```
# Removing year and special characters, and converting all text into small␣
↪lettercase
```

[151]:
```
removal = [",", "-", ".", ".", ":", ";", "(", ")", "[", "]", "~", "/", " ",␣
↪"%", "¥", "˜", "$", "*", "2010", "2011", "2012", "2013", "2014", "2015",␣
↪"2016", "2017", "2018", "2019", "2020", "2021", "2025", "2030", "2050"]
```

[ ]:
```

```

[152]:
```
filt_text2010 = [w for w in filt_text2010 if w not in removal]
```

[153]:
```
filt_text2011 = [w for w in filt_text2011 if w not in removal]
```

[154]:
```
filt_text2012 = [w for w in filt_text2012 if w not in removal]
```

[155]:
```
filt_text2013 = [w for w in filt_text2013 if w not in removal]
```

[156]:
```
filt_text2014 = [w for w in filt_text2014 if w not in removal]
```

[157]:
```
filt_text2015 = [w for w in filt_text2015 if w not in removal]
```

[158]:
```
filt_text2016 = [w for w in filt_text2016 if w not in removal]
```

[159]:
```
filt_text2017 = [w for w in filt_text2017 if w not in removal]
```

[160]:
```
filt_text2018 = [w for w in filt_text2018 if w not in removal]
```

[161]:
```
filt_text2018 = [w for w in filt_text2018 if w not in removal]
```

[162]:
```
filt_text2019 = [w for w in filt_text2019 if w not in removal]
```

[163]:
```
filt_text2020 = [w for w in filt_text2020 if w not in removal]
```

[ ]:
```

```

[164]:
```
new_text2010 = " ".join(filt_text2010)
```

[165]:
```
new_text2011 = " ".join(filt_text2011)
```

[166]:
```
new_text2012 = " ".join(filt_text2012)
```

[167]:
```
new_text2013 = " ".join(filt_text2013)
```

[168]:
```
new_text2014 = " ".join(filt_text2014)
```

[169]:
```
new_text2015 = " ".join(filt_text2015)
```

[170]:
```
new_text2016 = " ".join(filt_text2016)
```

[171]:
```
new_text2017 = " ".join(filt_text2017)
```

[172]:
```
new_text2018 = " ".join(filt_text2018)
```

[173]:
```
new_text2019 = " ".join(filt_text2019)
```

[174]:
```
new_text2020 = " ".join(filt_text2020)
```

[ ]:
```

```

[175]:
```
new_text2010 = re.sub(r'[^\w\s]', '', new_text2010)
```

[176]:
```
new_text2011 = re.sub(r'[^\w\s]', '', new_text2011)
```

[177]:
```
new_text2012 = re.sub(r'[^\w\s]', '', new_text2012)
```

[178]:
```
new_text2013 = re.sub(r'[^\w\s]', '', new_text2013)
```

[179]:
```
new_text2014 = re.sub(r'[^\w\s]', '', new_text2014)
```

[180]:
```
new_text2015 = re.sub(r'[^\w\s]', '', new_text2015)
```

[181]:
```
new_text2016 = re.sub(r'[^\w\s]', '', new_text2016)
```

[182]:
```
new_text2017 = re.sub(r'[^\w\s]', '', new_text2017)
```

[183]:
```
new_text2018 = re.sub(r'[^\w\s]', '', new_text2018)
```

[184]:
```
new_text2019 = re.sub(r'[^\w\s]', '', new_text2019)
```

[185]:
```
new_text2020 = re.sub(r'[^\w\s]', '', new_text2020)
```

[ ]:
```

```

[186]:
```
new_text2010 = new_text2010.lower()
```

[187]:
```
new_text2011 = new_text2011.lower()
```

[188]:
```
new_text2012 = new_text2012.lower()
```

[189]:
```
new_text2013 = new_text2013.lower()
```

[190]:
```
new_text2014 = new_text2014.lower()
```

[191]:
```
new_text2015 = new_text2015.lower()
```

[192]:
```
new_text2016 = new_text2016.lower()
```

[193]:
```
new_text2017 = new_text2017.lower()
```

[194]:
```
new_text2018 = new_text2018.lower()
```

[195]:
```
new_text2019 = new_text2019.lower()
```

[196]:
```
new_text2020 = new_text2020.lower()
```

[ ]:
```

```

[197]:
```
#print(new_text2020) # Controlling that the standard preprocessing procedures␣
↪works properly
```

[ ]:
```

```

[198]:
```
## Search Queries for Key Performance Indicators and Narrative Disclosure
```

[ ]:
```

```

[199]:
```
# Key Performance Indicators on the Impact on the External Environment
```

[200]:
```
Env_KPI = re.compile("(tonne|ton|tonnes|tons|kilo|kg)\s+(?:
↪\w+\s+){0,8}(GHG|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)\s+(?:\w+\s+){0,8}(\d+)",␣
↪flags=re.I)
```

[201]:
```
Env_KPI1 = re.compile("(tonne|ton|tonnes|tons|kilo|kg)\s+(?:
↪\w+\s+){0,8}(d+|%)\s+(?:\w+\s+){0,8}(GHG|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)", flags=re.I)
```

[202]:
```
Env_KPI2 = re.compile("(\d+)\s+(?:
↪\w+\s+){0,8}(tonne|ton|tonnes|tons|kilo|kg)\s+(?:\w+\s+){0,8}(GHG|CO2|CO␣
↪2|CO2e|CO 2E|N2O|carbon|greenhouse\s+gas|emissions|emission)", flags=re.I)
```

[203]:
```
Env_KPI3 = re.compile("(\d+)\s+(?:\w+\s+){0,8}(GHG|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)\s+(?:
↪\w+\s+){0,8}(tonne|ton|tonnes|tons|kilo|kg)", flags=re.I)
```

[204]:
```
Env_KPI4 = re.compile("(GHG|intesity|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)\s+(?:
↪\w+\s+){0,8}(tonne|ton|tonnes|tons|kilo|kg)\s+(?:\w+\s+){0,8}(\d+)",␣
↪flags=re.I)
```

[205]:
```
Env_KPI5 = re.compile("(GHG|intesity|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)\s+(?:
↪\w+\s+){0,8}(\d+)\s+(?:\w+\s+){0,8}(tonne|ton|tonnes|tons|kilo|kg)",␣
↪flags=re.I)
```

[206]:
```
Env_KPI6 = re.compile("(scope 1|scope 2|scope 3)\s+(?:
↪\w+\s+){0,8}(emission|emissions)\s+(?:
↪\w+\s+){0,8}(\d+|tonne|ton|tonnes|tons|kilo|kg)", flags=re.I)
```

[207]:
```
Env_KPI7 = re.compile("(\d+|tonne|ton|tonnes|tons|kilo|kg)\s+(?:
↪\w+\s+){0,8}(scope 1|scope 2|scope 3)\s+(?:
↪\w+\s+){0,8}(emission|emissions)", flags=re.I)
```

[208]:
```
Env_KPI8 = re.compile("(emission|emissions)\s+(?:\w+\s+){0,8}(scope 1|scope␣
↪2|scope 3)\s+(?:\w+\s+){0,8}(\d+|tonne|ton|tonnes|tons|kilo|kg)", flags=re.I)
```

[ ]:
```

```

[209]:
```
# Printing matches (can be adjusted for year and pattern)
```

[210]:
```
print(Env_KPI.findall(new_text2016), len(Env_KPI7.findall(new_text2016)))
```

```
[('tonnes', 'emissions', '1500')] 0
```

[ ]:
```

```

[211]:
```
# Aggregating all matches
```

[212]:
```
match_env2010 = len(Env_KPI8.findall(new_text2010)) + len(Env_KPI7.
↪findall(new_text2010)) + len(Env_KPI6.findall(new_text2010)) + len(Env_KPI5.
↪findall(new_text2010)) + len(Env_KPI4.findall(new_text2010)) + len(Env_KPI3.
↪findall(new_text2010)) + len(Env_KPI2.findall(new_text2010)) + len(Env_KPI1.
↪findall(new_text2010)) + len(Env_KPI.findall(new_text2010))
```

[213]:
```
match_env2011 = len(Env_KPI8.findall(new_text2011)) + len(Env_KPI7.
↪findall(new_text2011)) + len(Env_KPI6.findall(new_text2011)) + len(Env_KPI5.
↪findall(new_text2011)) + len(Env_KPI4.findall(new_text2011)) + len(Env_KPI3.
↪findall(new_text2011)) + len(Env_KPI2.findall(new_text2011)) + len(Env_KPI1.
↪findall(new_text2011)) + len(Env_KPI.findall(new_text2011))
```

[214]:
```
match_env2012 = len(Env_KPI8.findall(new_text2012)) + len(Env_KPI7.
↪findall(new_text2012)) + len(Env_KPI6.findall(new_text2012)) + len(Env_KPI5.
↪findall(new_text2012)) + len(Env_KPI4.findall(new_text2012)) + len(Env_KPI3.
↪findall(new_text2012)) + len(Env_KPI2.findall(new_text2012)) + len(Env_KPI1.
↪findall(new_text2012)) + len(Env_KPI.findall(new_text2012))
```

[215]:
```
match_env2013 = len(Env_KPI8.findall(new_text2013)) + len(Env_KPI7.
↪findall(new_text2013)) + len(Env_KPI6.findall(new_text2013)) + len(Env_KPI5.
↪findall(new_text2013)) + len(Env_KPI4.findall(new_text2013)) + len(Env_KPI3.
↪findall(new_text2013)) + len(Env_KPI2.findall(new_text2013)) + len(Env_KPI1.
↪findall(new_text2013)) + len(Env_KPI.findall(new_text2013))
```

[216]:
```
match_env2014 = len(Env_KPI8.findall(new_text2014)) + len(Env_KPI7.
↪findall(new_text2014)) + len(Env_KPI6.findall(new_text2014)) + len(Env_KPI5.
↪findall(new_text2014)) + len(Env_KPI4.findall(new_text2014)) + len(Env_KPI3.
↪findall(new_text2014)) + len(Env_KPI2.findall(new_text2014)) + len(Env_KPI1.
↪findall(new_text2014)) + len(Env_KPI.findall(new_text2014))
```

[217]:
```
match_env2015 = len(Env_KPI8.findall(new_text2015)) + len(Env_KPI7.
↪findall(new_text2015)) + len(Env_KPI6.findall(new_text2015)) + len(Env_KPI5.
↪findall(new_text2015)) + len(Env_KPI4.findall(new_text2015)) + len(Env_KPI3.
↪findall(new_text2015)) + len(Env_KPI2.findall(new_text2015)) + len(Env_KPI1.
↪findall(new_text2015)) + len(Env_KPI.findall(new_text2015))
```

[218]:
```
match_env2016 = len(Env_KPI8.findall(new_text2016)) + len(Env_KPI7.
↪findall(new_text2016)) + len(Env_KPI6.findall(new_text2016)) + len(Env_KPI5.
↪findall(new_text2016)) + len(Env_KPI4.findall(new_text2016)) + len(Env_KPI3.
↪findall(new_text2016)) + len(Env_KPI2.findall(new_text2016)) + len(Env_KPI1.
↪findall(new_text2016)) + len(Env_KPI.findall(new_text2016))
```

[219]:
```

```

```
match_env2017 = len(Env_KPI8.findall(new_text2017)) + len(Env_KPI7.
↪findall(new_text2017)) + len(Env_KPI6.findall(new_text2017)) + len(Env_KPI5.
↪findall(new_text2017)) + len(Env_KPI4.findall(new_text2017)) + len(Env_KPI3.
↪findall(new_text2017)) + len(Env_KPI2.findall(new_text2017)) + len(Env_KPI1.
↪findall(new_text2017)) + len(Env_KPI.findall(new_text2017))
```

[220]:
```
match_env2018 = len(Env_KPI8.findall(new_text2018)) + len(Env_KPI7.
↪findall(new_text2018)) + len(Env_KPI6.findall(new_text2018)) + len(Env_KPI5.
↪findall(new_text2018)) + len(Env_KPI4.findall(new_text2018)) + len(Env_KPI3.
↪findall(new_text2018)) + len(Env_KPI2.findall(new_text2018)) + len(Env_KPI1.
↪findall(new_text2018)) + len(Env_KPI.findall(new_text2018))
```

[221]:
```
match_env2019 = len(Env_KPI8.findall(new_text2019)) + len(Env_KPI7.
↪findall(new_text2019)) + len(Env_KPI6.findall(new_text2019)) + len(Env_KPI5.
↪findall(new_text2019)) + len(Env_KPI4.findall(new_text2019)) + len(Env_KPI3.
↪findall(new_text2019)) + len(Env_KPI2.findall(new_text2019)) + len(Env_KPI1.
↪findall(new_text2019)) + len(Env_KPI.findall(new_text2019))
```

[222]:
```
match_env2020 = len(Env_KPI8.findall(new_text2020)) + len(Env_KPI7.
↪findall(new_text2020)) + len(Env_KPI6.findall(new_text2020)) + len(Env_KPI5.
↪findall(new_text2020)) + len(Env_KPI4.findall(new_text2020)) + len(Env_KPI3.
↪findall(new_text2020)) + len(Env_KPI2.findall(new_text2020)) + len(Env_KPI1.
↪findall(new_text2020)) + len(Env_KPI.findall(new_text2020))
```

[ ]:
```

```

[223]:
```
# Creating a binary variable
```

[224]:
```
if (match_env2010 > 0):
    E_Match2010 = "1"
else:
    E_Match2010 = "0"
print(E_Match2010)
```

```
1
```

[225]:
```
if (match_env2011 > 0):
    E_Match2011 = "1"
else:
    E_Match2011 = "0"
print(E_Match2011)
```

```
0
```

[226]:
```
if (match_env2012 > 0):
    E_Match2012 = "1"
else:
    E_Match2012 = "0"
```

```
print(E_Match2012)
```

0

[227]:
```
if (match_env2013 > 0):
    E_Match2013 = "1"
else:
    E_Match2013 = "0"
print(E_Match2013)
```

1

[228]:
```
if (match_env2014 > 0):
    E_Match2014 = "1"
else:
    E_Match2014 = "0"
print(E_Match2014)
```

1

[229]:
```
if (match_env2015 > 0):
    E_Match2015 = "1"
else:
    E_Match2015 = "0"
print(E_Match2015)
```

1

[230]:
```
if (match_env2016 > 0):
    E_Match2016 = "1"
else:
    E_Match2016 = "0"
print(E_Match2016)
```

1

[231]:
```
if (match_env2017 > 0):
    E_Match2017 = "1"
else:
    E_Match2017 = "0"
print(E_Match2017)
```

1

[232]:
```
if (match_env2018 > 0):
    E_Match2018 = "1"
else:
    E_Match2018 = "0"
print(E_Match2018)
```

1

[233]:
```
if (match_env2019 > 0):
    E_Match2019 = "1"
else:
    E_Match2019 = "0"
print(E_Match2019)
```

```
1
```

[234]:
```
if (match_env2020 > 0):
    E_Match2020 = "1"
else:
    E_Match2020 = "0"
print(E_Match2020)
```

```
1
```

[ ]:
```
```

[235]:
```
## Key Performance Indicators on Gender Equality
```

[236]:
```
Gen_KPI = re.compile("(female|gender|woman|women|sex)\s+(?:
↪\w+\s+){0,8}(board|director|directors|executive|executives|manager|managers|management|mane
↪:\w+\s+){0,8}(\d+|one|two|three|four|five|six|seven|eigth|nine)", flags=re.I)
```

[237]:
```
Gen_KPI1 = re.compile("(\d+|one|two|three|four|five|six|seven|eigth|nine)\s+(?:
↪\w+\s+){0,8}(board|director|directors|executive|executives|manager|managers|management|mane
↪:\w+\s+){0,8}(female|gender|woman|women|sex)", flags=re.I)
```

[238]:
```
Gen_KPI2 = re.compile("(female|gender|woman|women|sex)\s+(?:
↪\w+\s+){0,8}(\d+|%|one|two|three|four|five|six|seven|eigth|nine)\s+(?:
↪\w+\s+){0,8}(board|director|directors|executive|executives|manager|managers|management|mane
↪flags=re.I)
```

[239]:
```
Gen_KPI3 = re.
↪compile("(board|director|directors|executive|executives|manager|managers|management|maneger
↪:\w+\s+){0,8}(\d+|%|one|two|three|four|five|six|seven|eigth|nine)\s+(?:
↪\w+\s+){0,8}(female|gender|woman|women|sex)", flags=re.I)
```

[240]:
```
Gen_KPI4 = re.compile("(\d+|%|one|two|three|four|five|six|seven|eigth|nine)\s+(?
↪:\w+\s+){0,8}(female|gender|woman|women|sex)\s+(?:
↪\w+\s+){0,8}(board|director|directors|executive|executives|manager|managers|management|mane
↪flags=re.I)
```

[241]:
```
Gen_KPI5 = re.
↪compile("(board|director|directors|executive|executives|manager|managers|management|maneger
↪:\w+\s+){0,8}(female|gender|woman|women|sex)\s+(?:
↪\w+\s+){0,8}(\d+|%|one|two|three|four|five|six|seven|eigth|nine)", flags=re.
↪I)
```

[242]:
```
Gen_KPI6 = re.compile("(gender|sex|female|women|woman)\s+(?:
↪\w+\s+){0,8}(split|diversity|ratio|distribution|composition|percentage|breakdown)\s+(?
↪:\w+\s+){0,8}(\d+|%)", flags=re.I)
```

[243]:
```
Gen_KPI7 = re.compile("(gender|sex|female|women|woman)\s+(?:
↪\w+\s+){0,8}(\d+)\s+(?:
↪\w+\s+){0,8}(split|diversity|ratio|distribution|composition|percentage|breakdown)",␣
↪flags=re.I)
```

[244]:
```
Gen_KPI8 = re.
↪compile("(split|diversity|ratio|distribution|composition|percentage|breakdown)\s+(?
↪:\w+\s+){0,8}(gender|sex|female|women|woman)\s+(?:\w+\s+){0,8}(\d+|%)",␣
↪flags=re.I)
```

[245]:
```
Gen_KPI9 = re.
↪compile("(split|diversity|ratio|distribution|composition|percentage|breakdown)\s+(?
↪:\w+\s+){0,8}(\d+|%)\s+(?:\w+\s+){0,8}(gender|sex|female|women|woman)",␣
↪flags=re.I)
```

[246]:
```
Gen_KPI10 = re.compile("(\d+|%)\s+(?:
↪\w+\s+){0,8}(split|diversity|ratio|distribution|composition|percentage|breakdown)\s+(?
↪:\w+\s+){0,8}(gender|sex|female|women|woman)", flags=re.I)
```

[247]:
```
Gen_KPI11 = re.compile("(\d+|%)\s+(?:
↪\w+\s+){0,8}(gender|sex|female|women|woman)\s+(?:
↪\w+\s+){0,8}(split|diversity|ratio|distribution|composition|percentage|breakdown)",␣
↪flags=re.I)
```

[ ]:
```

```

[248]:
```
# Printing matches (year by year)
```

[249]:
```
print(Gen_KPI.findall(new_text2011), len(Gen_KPI.findall(new_text2011)))
```

```
[] 0
```

[250]:
```
print(Gen_KPI1.findall(new_text2011), len(Gen_KPI1.findall(new_text2011)))
```

```
[] 0
```

[251]:
```
print(Gen_KPI2.findall(new_text2011), len(Gen_KPI2.findall(new_text2011)))
```

```
[] 0
```

[252]:
```
print(Gen_KPI3.findall(new_text2010), len(Gen_KPI3.findall(new_text2010)))
```

```
[('board', 'seven', 'women'), ('board', '14', 'woman')] 2
```

[253]:
```
print(Gen_KPI4.findall(new_text2011), len(Gen_KPI4.findall(new_text2011)))
```

```
[] 0
```

[254]:
```
print(Gen_KPI5.findall(new_text2017), len(Gen_KPI5.findall(new_text2017)))
```

```
[('board', 'women', 'three'), ('employee', 'women', '20152016'), ('management',
'women', '485486484456467447')] 3
```

[255]:
```
print(Gen_KPI6.findall(new_text2011), len(Gen_KPI6.findall(new_text2011)))
```

```
[] 0
```

[256]:
```
print(Gen_KPI7.findall(new_text2011), len(Gen_KPI7.findall(new_text2011)))
```

```
[] 0
```

[257]:
```
print(Gen_KPI8.findall(new_text2011), len(Gen_KPI8.findall(new_text2011)))
```

```
[] 0
```

[258]:
```
print(Gen_KPI9.findall(new_text2011), len(Gen_KPI9.findall(new_text2011)))
```

```
[] 0
```

[259]:
```
print(Gen_KPI10.findall(new_text2019), len(Gen_KPI10.findall(new_text2019)))
```

```
[('40', 'percentage', 'women'), ('355', 'percentage', 'women'), ('51544018',
'diversity', 'women')] 3
```

[260]:
```
print(Gen_KPI11.findall(new_text2011), len(Gen_KPI11.findall(new_text2011)))
```

```
[] 0
```

[ ]:
```

```

[261]:
```
# Aggregating all matches
```

[262]:
```
match_gen2010 = len(Gen_KPI11.findall(new_text2010)) + len(Gen_KPI10.
↪findall(new_text2010)) + len(Gen_KPI9.findall(new_text2010)) + len(Gen_KPI8.
↪findall(new_text2010))  + len(Gen_KPI7.findall(new_text2010)) + len(Gen_KPI6.
↪findall(new_text2010)) + len(Gen_KPI5.findall(new_text2010)) + len(Gen_KPI4.
↪findall(new_text2010)) + len(Gen_KPI3.findall(new_text2010)) + len(Gen_KPI2.
↪findall(new_text2010)) + len(Gen_KPI1.findall(new_text2010)) + len(Gen_KPI.
↪findall(new_text2010))
```

[263]:
```
match_gen2011 = len(Gen_KPI11.findall(new_text2011)) + len(Gen_KPI10.
↪findall(new_text2011)) + len(Gen_KPI9.findall(new_text2011)) + len(Gen_KPI8.
↪findall(new_text2011))  + len(Gen_KPI7.findall(new_text2011)) + len(Gen_KPI6.
↪findall(new_text2011)) + len(Gen_KPI5.findall(new_text2011)) + len(Gen_KPI4.
↪findall(new_text2011)) + len(Gen_KPI3.findall(new_text2011)) + len(Gen_KPI2.
↪findall(new_text2011)) + len(Gen_KPI1.findall(new_text2011)) + len(Gen_KPI.
↪findall(new_text2011))
```

[264]:
```
match_gen2012 = len(Gen_KPI11.findall(new_text2012)) + len(Gen_KPI10.
↪findall(new_text2012)) + len(Gen_KPI9.findall(new_text2012)) + len(Gen_KPI8.
↪findall(new_text2012))  + len(Gen_KPI7.findall(new_text2012)) + len(Gen_KPI6.
↪findall(new_text2012)) + len(Gen_KPI5.findall(new_text2012)) + len(Gen_KPI4.
↪findall(new_text2012)) + len(Gen_KPI3.findall(new_text2012)) + len(Gen_KPI2.
↪findall(new_text2012)) + len(Gen_KPI1.findall(new_text2012)) + len(Gen_KPI.
↪findall(new_text2012))
```

[265]:
```
match_gen2013 = len(Gen_KPI11.findall(new_text2013)) + len(Gen_KPI10.
↪findall(new_text2013)) + len(Gen_KPI9.findall(new_text2013)) + len(Gen_KPI8.
↪findall(new_text2013))  + len(Gen_KPI7.findall(new_text2013)) + len(Gen_KPI6.
↪findall(new_text2013)) + len(Gen_KPI5.findall(new_text2013)) + len(Gen_KPI4.
↪findall(new_text2013)) + len(Gen_KPI3.findall(new_text2013)) + len(Gen_KPI2.
↪findall(new_text2013)) + len(Gen_KPI1.findall(new_text2013)) + len(Gen_KPI.
↪findall(new_text2013))
```

[266]:
```
match_gen2014 = len(Gen_KPI11.findall(new_text2014)) + len(Gen_KPI10.
↪findall(new_text2014)) + len(Gen_KPI9.findall(new_text2014)) + len(Gen_KPI8.
↪findall(new_text2014))  + len(Gen_KPI7.findall(new_text2014)) + len(Gen_KPI6.
↪findall(new_text2014)) + len(Gen_KPI5.findall(new_text2014)) + len(Gen_KPI4.
↪findall(new_text2014)) + len(Gen_KPI3.findall(new_text2014)) + len(Gen_KPI2.
↪findall(new_text2014)) + len(Gen_KPI1.findall(new_text2014)) + len(Gen_KPI.
↪findall(new_text2014))
```

[267]:
```
match_gen2015 = len(Gen_KPI11.findall(new_text2015)) + len(Gen_KPI10.
↪findall(new_text2015)) + len(Gen_KPI9.findall(new_text2015)) + len(Gen_KPI8.
↪findall(new_text2015))  + len(Gen_KPI7.findall(new_text2015)) + len(Gen_KPI6.
↪findall(new_text2015)) + len(Gen_KPI5.findall(new_text2015)) + len(Gen_KPI4.
↪findall(new_text2015)) + len(Gen_KPI3.findall(new_text2015)) + len(Gen_KPI2.
↪findall(new_text2015)) + len(Gen_KPI1.findall(new_text2015)) + len(Gen_KPI.
↪findall(new_text2015))
```

[268]:
```
match_gen2016 = len(Gen_KPI11.findall(new_text2016)) + len(Gen_KPI10.
↪findall(new_text2016)) + len(Gen_KPI9.findall(new_text2016)) + len(Gen_KPI8.
↪findall(new_text2016))  + len(Gen_KPI7.findall(new_text2016)) + len(Gen_KPI6.
↪findall(new_text2016)) + len(Gen_KPI5.findall(new_text2016)) + len(Gen_KPI4.
↪findall(new_text2016)) + len(Gen_KPI3.findall(new_text2016)) + len(Gen_KPI2.
↪findall(new_text2016)) + len(Gen_KPI1.findall(new_text2016)) + len(Gen_KPI.
↪findall(new_text2016))
```

[269]:
```
match_gen2017 = len(Gen_KPI11.findall(new_text2017)) + len(Gen_KPI10.
↪findall(new_text2017)) + len(Gen_KPI9.findall(new_text2017)) + len(Gen_KPI8.
↪findall(new_text2017))  + len(Gen_KPI7.findall(new_text2017)) + len(Gen_KPI6.
↪findall(new_text2017)) + len(Gen_KPI5.findall(new_text2017)) + len(Gen_KPI4.
↪findall(new_text2017)) + len(Gen_KPI3.findall(new_text2017)) + len(Gen_KPI2.
↪findall(new_text2017)) + len(Gen_KPI1.findall(new_text2017)) + len(Gen_KPI.
↪findall(new_text2017))
```

[270]:
```
match_gen2018 = len(Gen_KPI11.findall(new_text2018)) + len(Gen_KPI10.
↪findall(new_text2018)) + len(Gen_KPI9.findall(new_text2018)) + len(Gen_KPI8.
↪findall(new_text2018))  + len(Gen_KPI7.findall(new_text2018)) + len(Gen_KPI6.
↪findall(new_text2018)) + len(Gen_KPI5.findall(new_text2018)) + len(Gen_KPI4.
↪findall(new_text2018)) + len(Gen_KPI3.findall(new_text2018)) + len(Gen_KPI2.
↪findall(new_text2018)) + len(Gen_KPI1.findall(new_text2018)) + len(Gen_KPI.
↪findall(new_text2018))
```

[271]:
```
match_gen2019 = len(Gen_KPI11.findall(new_text2019)) + len(Gen_KPI10.
↪findall(new_text2019)) + len(Gen_KPI9.findall(new_text2019)) + len(Gen_KPI8.
↪findall(new_text2019))  + len(Gen_KPI7.findall(new_text2019)) + len(Gen_KPI6.
↪findall(new_text2019)) + len(Gen_KPI5.findall(new_text2019)) + len(Gen_KPI4.
↪findall(new_text2019)) + len(Gen_KPI3.findall(new_text2019)) + len(Gen_KPI2.
↪findall(new_text2019)) + len(Gen_KPI1.findall(new_text2019)) + len(Gen_KPI.
↪findall(new_text2019))
```

[272]:
```
match_gen2020 = len(Gen_KPI11.findall(new_text2020)) + len(Gen_KPI10.
↪findall(new_text2020)) + len(Gen_KPI9.findall(new_text2020)) + len(Gen_KPI8.
↪findall(new_text2020))  + len(Gen_KPI7.findall(new_text2020)) + len(Gen_KPI6.
↪findall(new_text2020)) + len(Gen_KPI5.findall(new_text2020)) + len(Gen_KPI4.
↪findall(new_text2020)) + len(Gen_KPI3.findall(new_text2020)) + len(Gen_KPI2.
↪findall(new_text2020)) + len(Gen_KPI1.findall(new_text2020)) + len(Gen_KPI.
↪findall(new_text2020))
```

[ ]:
```

```

[273]:
```
# Creating binary variable
```

[274]:
```
if (match_gen2010 > 0):
    G_Match2010 = "1"
else:
    G_Match2010 = "0"
print(G_Match2010)
```

```
1
```

[275]:
```
if (match_gen2011 > 0):
    G_Match2011 = "1"
else:
    G_Match2011 = "0"
print(G_Match2011)
```

```
0
```

[276]:
```
if (match_gen2012 > 0):
    G_Match2012 = "1"
else:
    G_Match2012 = "0"
```

```
print(G_Match2012)
```

0

[277]:
```
if (match_gen2013 > 0):
    G_Match2013 = "1"
else:
    G_Match2013 = "0"
print(G_Match2013)
```

1

[278]:
```
if (match_gen2014 > 0):
    G_Match2014 = "1"
else:
    G_Match2014 = "0"
print(G_Match2014)
```

1

[279]:
```
if (match_gen2015 > 0):
    G_Match2015 = "1"
else:
    G_Match2015 = "0"
print(G_Match2015)
```

1

[280]:
```
if (match_gen2016 > 0):
    G_Match2016 = "1"
else:
    G_Match2016 = "0"
print(G_Match2016)
```

1

[281]:
```
if (match_gen2017 > 0):
    G_Match2017 = "1"
else:
    G_Match2017 = "0"
print(G_Match2017)
```

1

[282]:
```
if (match_gen2018 > 0):
    G_Match2018 = "1"
else:
    G_Match2018 = "0"
print(G_Match2018)
```

1

[283]:
```
if (match_gen2019 > 0):
    G_Match2019 = "1"
else:
    G_Match2019 = "0"
print(G_Match2019)
```

```
1
```

[284]:
```
if (match_gen2020 > 0):
    G_Match2020 = "1"
else:
    G_Match2020 = "0"
print(G_Match2020)
```

```
1
```

[ ]:
```

```

[285]:
```
## Key Performance Indicators on Work Environment
```

[286]:
```
HSE_KPI = re.compile("(number|total|reported|reportable)\s+(?:
↪\w+\s+){0,8}(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sick
↪absence|sick leave)\s+(?:\w+\s+){0,8}(\d+|no|none)", flags=re.I)
```

[287]:
```
HSE_KPI1 = re.compile("(\d+|no|none)\s+(?:
↪\w+\s+){0,8}(number|total|reported|reportable)\s+(?:
↪\w+\s+){0,8}(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sick
↪absence|sick leave)", flags=re.I)
```

[288]:
```
HSE_KPI2 = re.
↪compile("(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sicknes
↪absence|sick leave)\s+(?:\w+\s+){0,8}(\d+|no|none)", flags=re.I)
```

[289]:
```
HSE_KPI3 = re.compile("(number|total|reported|reportable)\s+(?:
↪\w+\s+){0,8}(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sick
↪absence|sick leave)", flags=re.I)
```

[290]:
```
HSE_KPI4 = re.
↪compile("(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sicknes
↪absence|sick leave)\s+(?:\w+\s+){0,8}(number|total|reported|reportable)",␣
↪flags=re.I)
```

[291]:
```
HSE_KPI5 = re.
↪compile("(TRI|OSHA|TCIR|TRIR|TRIFR|TRCF|AIFR|AFR|LTI|LTIF|LTIR|LTIFR|LWDR)\s+(?
↪:\w+\s+){0,8}(\d+)", flags=re.I)
```

[292]:
```

```

```
HSE_KPI6 = re.compile("(\d+)\s+(?:
↪\w+\s+){0,8}(TRI|OSHA|TCIR|TRIR|TRIFR|TRCF|AIFR|AFR|LTI|LTIF|LTIR|LTIFR)",␣
↪flags=re.I)
```

[ ]:

[293]:
```
# Printing matches (year by year)
```

[294]:
```
print(HSE_KPI.findall(new_text2013), len(HSE_KPI.findall(new_text2013)))
```

```
[('reported', 'sickness absence', '46'), ('number', 'injuries', '36')] 2
```

[295]:
```
print(HSE_KPI1.findall(new_text2013), len(HSE_KPI1.findall(new_text2013)))
```

```
[('5', 'reported', 'injuries'), ('1', 'reported', 'sickness absence'), ('49',
'number', 'injuries')] 3
```

[296]:
```
print(HSE_KPI2.findall(new_text2013), len(HSE_KPI2.findall(new_text2013)))
```

```
[('sickness absence', '46'), ('sickness absence', '54'), ('sickness absence',
'43'), ('sickness absence', 'no'), ('injuries', '36')] 5
```

[297]:
```
print(HSE_KPI3.findall(new_text2013), len(HSE_KPI2.findall(new_text2013)))
```

```
[('reported', 'injuries'), ('reported', 'sickness absence'), ('number',
'injuries'), ('total', 'injuries'), ('number', 'accidents')] 5
```

[298]:
```
print(HSE_KPI4.findall(new_text2013), len(HSE_KPI2.findall(new_text2013)))
```

```
[] 5
```

[299]:
```
print(HSE_KPI5.findall(new_text2013), len(HSE_KPI5.findall(new_text2013)))
```

```
[('lwdr', '20092012')] 1
```

[300]:
```
print(HSE_KPI6.findall(new_text2013), len(HSE_KPI5.findall(new_text2013)))
```

```
[] 1
```

[ ]:

[301]:
```
# Aggregating all matches
```

[302]:
```
match_HSE2010 = len(HSE_KPI6.findall(new_text2010)) + len(HSE_KPI5.
↪findall(new_text2010)) + len(HSE_KPI4.findall(new_text2010)) + len(HSE_KPI3.
↪findall(new_text2010)) + len(HSE_KPI2.findall(new_text2010)) + len(HSE_KPI1.
↪findall(new_text2010)) + len(HSE_KPI.findall(new_text2010))
```

[303]:

```
match_HSE2011 = len(HSE_KPI6.findall(new_text2011)) + len(HSE_KPI5.
↪findall(new_text2011)) + len(HSE_KPI4.findall(new_text2011)) + len(HSE_KPI3.
↪findall(new_text2011)) + len(HSE_KPI2.findall(new_text2011)) + len(HSE_KPI1.
↪findall(new_text2011)) + len(HSE_KPI.findall(new_text2011))
```

[304]:
```
match_HSE2012 = len(HSE_KPI6.findall(new_text2012)) + len(HSE_KPI5.
↪findall(new_text2012)) + len(HSE_KPI4.findall(new_text2012)) + len(HSE_KPI3.
↪findall(new_text2012)) + len(HSE_KPI2.findall(new_text2012)) + len(HSE_KPI1.
↪findall(new_text2012)) + len(HSE_KPI.findall(new_text2012))
```

[305]:
```
match_HSE2013 = len(HSE_KPI6.findall(new_text2013)) + len(HSE_KPI5.
↪findall(new_text2013)) + len(HSE_KPI4.findall(new_text2013)) + len(HSE_KPI3.
↪findall(new_text2013)) + len(HSE_KPI2.findall(new_text2013)) + len(HSE_KPI1.
↪findall(new_text2013)) + len(HSE_KPI.findall(new_text2013))
```

[306]:
```
match_HSE2014 = len(HSE_KPI6.findall(new_text2014)) + len(HSE_KPI5.
↪findall(new_text2014)) + len(HSE_KPI4.findall(new_text2014)) + len(HSE_KPI3.
↪findall(new_text2014)) + len(HSE_KPI2.findall(new_text2014)) + len(HSE_KPI1.
↪findall(new_text2014)) + len(HSE_KPI.findall(new_text2014))
```

[307]:
```
match_HSE2015 = len(HSE_KPI6.findall(new_text2015)) + len(HSE_KPI5.
↪findall(new_text2015)) + len(HSE_KPI4.findall(new_text2015)) + len(HSE_KPI3.
↪findall(new_text2015)) + len(HSE_KPI2.findall(new_text2015)) + len(HSE_KPI1.
↪findall(new_text2015)) + len(HSE_KPI.findall(new_text2015))
```

[308]:
```
match_HSE2016 = len(HSE_KPI6.findall(new_text2016)) + len(HSE_KPI5.
↪findall(new_text2016)) + len(HSE_KPI4.findall(new_text2016)) + len(HSE_KPI3.
↪findall(new_text2016)) + len(HSE_KPI2.findall(new_text2016)) + len(HSE_KPI1.
↪findall(new_text2016)) + len(HSE_KPI.findall(new_text2016))
```

[309]:
```
match_HSE2017 = len(HSE_KPI6.findall(new_text2017)) + len(HSE_KPI5.
↪findall(new_text2017)) + len(HSE_KPI4.findall(new_text2017)) + len(HSE_KPI3.
↪findall(new_text2017)) + len(HSE_KPI2.findall(new_text2017)) + len(HSE_KPI1.
↪findall(new_text2017)) + len(HSE_KPI.findall(new_text2017))
```

[310]:
```
match_HSE2018 = len(HSE_KPI6.findall(new_text2018)) + len(HSE_KPI5.
↪findall(new_text2018)) + len(HSE_KPI4.findall(new_text2018)) + len(HSE_KPI3.
↪findall(new_text2018)) + len(HSE_KPI2.findall(new_text2018)) + len(HSE_KPI1.
↪findall(new_text2018)) + len(HSE_KPI.findall(new_text2018))
```

[311]:
```
match_HSE2019 = len(HSE_KPI6.findall(new_text2019)) + len(HSE_KPI5.
↪findall(new_text2019)) + len(HSE_KPI4.findall(new_text2019)) + len(HSE_KPI3.
↪findall(new_text2019)) + len(HSE_KPI2.findall(new_text2019)) + len(HSE_KPI1.
↪findall(new_text2019)) + len(HSE_KPI.findall(new_text2019))
```

[312]:
```

```

```
match_HSE2020 = len(HSE_KPI6.findall(new_text2020)) + len(HSE_KPI5.
↪findall(new_text2020)) + len(HSE_KPI4.findall(new_text2020)) + len(HSE_KPI3.
↪findall(new_text2020)) + len(HSE_KPI2.findall(new_text2020)) + len(HSE_KPI1.
↪findall(new_text2020)) + len(HSE_KPI.findall(new_text2020))
```

[ ]:

[313]:
```
# Creating binary variable
```

[314]:
```
if (match_HSE2010 > 0):
    H_Match2010 = "1"
else:
    H_Match2010 = "0"
print(H_Match2010)
```

```
1
```

[315]:
```
if (match_HSE2011 > 0):
    H_Match2011 = "1"
else:
    H_Match2011 = "0"
print(H_Match2011)
```

```
1
```

[316]:
```
if (match_HSE2012 > 0):
    H_Match2012 = "1"
else:
    H_Match2012 = "0"
print(H_Match2012)
```

```
1
```

[317]:
```
if (match_HSE2013 > 0):
    H_Match2013 = "1"
else:
    H_Match2013 = "0"
print(H_Match2013)
```

```
1
```

[318]:
```
if (match_HSE2014 > 0):
    H_Match2014 = "1"
else:
    H_Match2014 = "0"
print(H_Match2014)
```

```
1
```

[319]:
```
if (match_HSE2015 > 0):
    H_Match2015 = "1"
else:
    H_Match2015 = "0"
print(H_Match2015)
```

```
1
```

[320]:
```
if (match_HSE2016 > 0):
    H_Match2016 = "1"
else:
    H_Match2016 = "0"
print(H_Match2016)
```

```
1
```

[321]:
```
if (match_HSE2017 > 0):
    H_Match2017 = "1"
else:
    H_Match2017 = "0"
print(H_Match2017)
```

```
1
```

[322]:
```
if (match_HSE2018 > 0):
    H_Match2018 = "1"
else:
    H_Match2018 = "0"
print(H_Match2018)
```

```
1
```

[323]:
```
if (match_HSE2019 > 0):
    H_Match2019 = "1"
else:
    H_Match2019 = "0"
print(H_Match2019)
```

```
1
```

[324]:
```
if (match_HSE2020 > 0):
    H_Match2020 = "1"
else:
    H_Match2020 = "0"
print(H_Match2020)
```

```
1
```

[ ]:
```

```

[325]:
```
## Narrative Disclosure Measures
```

[ ]:
```

```

[326]:
```
# Outside-In Impact
```

[327]:
```
E_Out = re.compile("(climate|planet|transition|physical)\s+(?:
↪\w+\s+){0,8}(risk|risks|impact|impacts)\s+(?:\w+\s+\D){0,5}", flags=re.I)
```

[ ]:
```

```

[328]:
```
# Printing matches
```

[329]:
```
print(E_Out.findall(new_text2020), len(E_Out.findall(new_text2020)))
```

```
[('climate', 'risk'), ('climate', 'impacts'), ('climate', 'risk'), ('climate',
'risk'), ('climate', 'risk'), ('climate', 'impact'), ('climate', 'risk'),
('climate', 'impact'), ('climate', 'impact'), ('climate', 'risk'), ('climate',
'impact'), ('climate', 'impact'), ('climate', 'risk'), ('climate', 'impact'),
('climate', 'impact'), ('climate', 'impacts'), ('climate', 'impact'),
('climate', 'impact'), ('climate', 'impact'), ('climate', 'risk'), ('climate',
'impact'), ('climate', 'impact'), ('climate', 'impact'), ('planet', 'impact'),
('climate', 'impact'), ('climate', 'impact'), ('climate', 'impact'), ('climate',
'impact'), ('climate', 'impact'), ('climate', 'impact'), ('climate', 'impact'),
('climate', 'impact'), ('climate', 'impact'), ('climate', 'risk'), ('climate',
'risk')] 35
```

[ ]:
```

```

[330]:
```
# Counting number of matches (converted to binary variable later)
```

[331]:
```
OutIn2010 = len(E_Out.findall(new_text2010))
```

[332]:
```
OutIn2011 = len(E_Out.findall(new_text2011))
```

[333]:
```
OutIn2012 = len(E_Out.findall(new_text2012))
```

[334]:
```
OutIn2013 = len(E_Out.findall(new_text2013))
```

[335]:
```
OutIn2014 = len(E_Out.findall(new_text2014))
```

[336]:
```
OutIn2015 = len(E_Out.findall(new_text2015))
```

[337]:
```
OutIn2016 = len(E_Out.findall(new_text2016))
```

[338]:
```
OutIn2017 = len(E_Out.findall(new_text2017))
```

[339]:
```
OutIn2018 = len(E_Out.findall(new_text2018))
```

[340]:
```
OutIn2019 = len(E_Out.findall(new_text2019))
```

[341]:
```
OutIn2020 = len(E_Out.findall(new_text2020))
```

[ ]:
```

```

[342]:
```
# Narrative Sustainability Disclosure
```

[ ]:
```

```

[343]:
```
# Environmental Dimension
```

[344]:
```
Nar_E = re.compile("(sustainable|sustainability|corporate␣
↪responsibility|renewable|ESG|global␣
↪warming|ecology|environment|environmental|paris agreement|kyoto␣
↪protocol|biodiversity|biofuel|biofuels|deforestation|reforestation|emission|emissions|recyc
↪development goals|ozone depletion|contamination|radioactive|hazardous|CO2|CO␣
↪2|GHG|nitrogen|asbestos|CSR|carbon|greenhouse|climate|net-zero|net zero|zero␣
↪waste|sectoral decarbonization|SDA|GRI|TCFD|circular economy|taxonomy|scope␣
↪1|scope 2|scope 3|footprint|un environment␣
↪programme|divestment|transition|toxic)", flags=re.I)
```

[345]:
```
# Printing matches (year by year)
```

[346]:
```
#print(Nar_E.findall(new_text2020), len(Nar_E.findall(new_text2020)))
```

[347]:
```
# Computing the Ratio ESG Dictionary (E) / Report Length
```

[348]:
```
NarEDS2010 = (len(Nar_E.findall(new_text2010)) / len(filt_text2010)) * 100
```

[349]:
```
NarEDS2011 = (len(Nar_E.findall(new_text2011)) / len(filt_text2011)) * 100
```

[350]:
```
NarEDS2012 = (len(Nar_E.findall(new_text2012)) / len(filt_text2012)) * 100
```

[351]:
```
NarEDS2013 = (len(Nar_E.findall(new_text2013)) / len(filt_text2013)) * 100
```

[352]:
```
NarEDS2014 = (len(Nar_E.findall(new_text2014)) / len(filt_text2014)) * 100
```

[353]:
```
NarEDS2015 = (len(Nar_E.findall(new_text2015)) / len(filt_text2015)) * 100
```

[354]:
```
NarEDS2016 = (len(Nar_E.findall(new_text2016)) / len(filt_text2016)) * 100
```

[355]:
```
NarEDS2017 = (len(Nar_E.findall(new_text2017)) / len(filt_text2017)) * 100
```

[356]:
```
NarEDS2018 = (len(Nar_E.findall(new_text2018)) / len(filt_text2018)) * 100
```

[357]:
```
NarEDS2019 = (len(Nar_E.findall(new_text2019)) / len(filt_text2019)) * 100
```

[358]:
```
NarEDS2020 = (len(Nar_E.findall(new_text2020)) / len(filt_text2020)) * 100
```

[359]:
```
print('Narrative ESG score (Environmental):', NarEDS2010)
```

```
Narrative ESG score (Environmental): 0.8103927152550715
```

[ ]:
```

```

[360]:
```
# Social Dimensions
```

[361]:
```
Nar_S = re.
↪compile("(gender|genders|transgender|female|woman|women|sex|ethnicity|ethnic|ethnicities|mi
↪rights|labor rights|equality|equal pay|pay gap|wage␣
↪gap|discrimination|discriminate|discriminated|harassment|harass|charity|donate|donation|don
↪absence|sick leave|fairtrade|fair trade)", flags=re.I)
```

[362]:
```
# Printing matches (year by year)
```

[363]:
```
#print(Nar_S.findall(new_text2014), len(Nar_S.findall(new_text2014)))
```

[364]:
```
# Computing the Ratio ESG Dictionary (S) / Report Length
```

[365]:
```
NarSDS2010 = (len(Nar_S.findall(new_text2010)) / len(filt_text2010)) * 100
```

[366]:
```
NarSDS2011 = (len(Nar_S.findall(new_text2011)) / len(filt_text2011)) * 100
```

[367]:
```
NarSDS2012 = (len(Nar_S.findall(new_text2012)) / len(filt_text2012)) * 100
```

[368]:
```
NarSDS2013 = (len(Nar_S.findall(new_text2013)) / len(filt_text2013)) * 100
```

[369]:
```
NarSDS2014 = (len(Nar_S.findall(new_text2014)) / len(filt_text2014)) * 100
```

[370]:
```
NarSDS2015 = (len(Nar_S.findall(new_text2015)) / len(filt_text2015)) * 100
```

[371]:
```
NarSDS2016 = (len(Nar_S.findall(new_text2016)) / len(filt_text2016)) * 100
```

[372]:
```
NarSDS2017 = (len(Nar_S.findall(new_text2017)) / len(filt_text2017)) * 100
```

[373]:
```
NarSDS2018 = (len(Nar_S.findall(new_text2018)) / len(filt_text2018)) * 100
```

[374]:
```
NarSDS2019 = (len(Nar_S.findall(new_text2019)) / len(filt_text2019)) * 100
```

[375]:
```
NarSDS2020 = (len(Nar_S.findall(new_text2020)) / len(filt_text2020)) * 100
```

[376]:
```
print('Narrative ESG score (Social):', NarSDS2020)
```

```
Narrative ESG score (Social): 0.8281174771304767
```

[ ]:
```

```

[377]:
```
# Governance Dimension
```

[378]:
```
Nar_G = re.
↪compile("(governance|conduct|misconduct|audit|audited|auditing|control|controls|oversee|ove
↪structure|board member|board␣
↪members|composition|independence|independent|succession|tenure|vacancies|vacancy|nomination
↪ethics|business ethic|ethics|ethical)", flags = re.I)
```

[379]:
```
# Printing matches
```

[380]:
```
#print(Nar_G.findall(new_text2019), len(Nar_G.findall(new_text2019)))
```

[381]:
```
# Computing the Ratio ESG Dictionary (G) / Report Length
```

[382]:
```
NarGDS2010 = (len(Nar_G.findall(new_text2010)) / len(filt_text2010)) * 100
```

[383]:
```
NarGDS2011 = (len(Nar_G.findall(new_text2011)) / len(filt_text2011)) * 100
```

[384]:
```
NarGDS2012 = (len(Nar_G.findall(new_text2012)) / len(filt_text2012)) * 100
```

[385]:
```
NarGDS2013 = (len(Nar_G.findall(new_text2013)) / len(filt_text2013)) * 100
```

[386]:
```
NarGDS2014 = (len(Nar_G.findall(new_text2014)) / len(filt_text2014)) * 100
```

[387]:
```
NarGDS2015 = (len(Nar_G.findall(new_text2015)) / len(filt_text2015)) * 100
```

[388]:
```
NarGDS2016 = (len(Nar_G.findall(new_text2016)) / len(filt_text2016)) * 100
```

[389]:
```
NarGDS2017 = (len(Nar_G.findall(new_text2017)) / len(filt_text2017)) * 100
```

[390]:
```
NarGDS2018 = (len(Nar_G.findall(new_text2018)) / len(filt_text2018)) * 100
```

[391]:
```
NarGDS2019 = (len(Nar_G.findall(new_text2019)) / len(filt_text2019)) * 100
```

[392]:
```
NarGDS2020 = (len(Nar_G.findall(new_text2020)) / len(filt_text2020)) * 100
```

[393]:
```
print('Narrative ESG Score (Governance):', NarGDS2020)
```

```
Narrative ESG Score (Governance): 2.121191278629892
```

[ ]:
```

```

[394]:
```
# Computing the Aggregate Narrative Sustainability Disclosure
```

[395]:
```
NarDS2010 = ( (len(Nar_E.findall(new_text2010)) + len(Nar_S.
↪findall(new_text2010)) + len(Nar_G.findall(new_text2010))) /␣
↪len(filt_text2010) ) * 100
```

[396]:
```
NarDS2011 = ( (len(Nar_E.findall(new_text2011)) + len(Nar_S.
↪findall(new_text2011)) + len(Nar_G.findall(new_text2011))) /␣
↪len(filt_text2011) ) * 100
```

[397]:
```
NarDS2012 = ( (len(Nar_E.findall(new_text2012)) + len(Nar_S.
↪findall(new_text2012)) + len(Nar_G.findall(new_text2012))) /␣
↪len(filt_text2012) ) * 100
```

[398]:
```
NarDS2013 = ( (len(Nar_E.findall(new_text2013)) + len(Nar_S.
↪findall(new_text2013)) + len(Nar_G.findall(new_text2013))) /␣
↪len(filt_text2013) ) * 100
```

[399]:
```
NarDS2014 = ( (len(Nar_E.findall(new_text2014)) + len(Nar_S.
↪findall(new_text2014)) + len(Nar_G.findall(new_text2014))) /␣
↪len(filt_text2014) ) * 100
```

[400]:
```
NarDS2015 = ( (len(Nar_E.findall(new_text2015)) + len(Nar_S.
↪findall(new_text2015)) + len(Nar_G.findall(new_text2015))) /␣
↪len(filt_text2015) ) * 100
```

[401]:
```
NarDS2016 = ( (len(Nar_E.findall(new_text2016)) + len(Nar_S.
↪findall(new_text2016)) + len(Nar_G.findall(new_text2016))) /␣
↪len(filt_text2016) ) * 100
```

[402]:
```
NarDS2017 = ( (len(Nar_E.findall(new_text2017)) + len(Nar_S.
↪findall(new_text2017)) + len(Nar_G.findall(new_text2017))) /␣
↪len(filt_text2017) ) * 100
```

[403]:
```
NarDS2018 = ( (len(Nar_E.findall(new_text2018)) + len(Nar_S.
↪findall(new_text2018)) + len(Nar_G.findall(new_text2018))) /␣
↪len(filt_text2018) ) * 100
```

[404]:
```
NarDS2019 = ( (len(Nar_E.findall(new_text2019)) + len(Nar_S.
↪findall(new_text2019)) + len(Nar_G.findall(new_text2019))) /␣
↪len(filt_text2019) ) * 100
```

[405]:
```
NarDS2020 = ( (len(Nar_E.findall(new_text2020)) + len(Nar_S.
↪findall(new_text2020)) + len(Nar_G.findall(new_text2020))) /␣
↪len(filt_text2020) ) * 100
```

[ ]:
```

```

[406]:
```
# Printing the score for the respective years
```

[407]:
```
print(NarDS2010, NarDS2011, NarDS2012, NarDS2013, NarDS2014, NarDS2015,␣
↪NarDS2016, NarDS2019, NarDS2020)
```

```
3.350560770664136 0.6534474988733664 0.46460781634326315 3.393511605809692
3.2513786764705883 3.966247668195052 4.061280531868767 4.595619015724658
4.821514547080267
```

[ ]:

[408]:
```
# Creating table with overview of data
```

[ ]:

[409]:
```
output_env = [[E_Match2010, E_Match2011, E_Match2012, E_Match2013, E_Match2014,␣
 ↪E_Match2015, E_Match2016, E_Match2017, E_Match2018, E_Match2019,␣
 ↪E_Match2020]]
```

[410]:
```
print("="*100)
print(tabulate(output_env, headers=["E2010", "E2011", "E2012", "E2013",␣
 ↪"E2014", "E2015", "E2016", "E2017", "E2018", "E2019", "E2020"]))
print("-"*100)
```

```
====================================================================================
====================
  E2010    E2011    E2012    E2013    E2014    E2015    E2016    E2017    E2018
E2019    E2020
-------  -------  -------  -------  -------  -------  -------  -------  -------
-------  -------
      1        0        0        1        1        1        1        1        1
1        1
------------------------------------------------------------------------------------
--------------------
```

[411]:
```
output_gen = [[G_Match2010, G_Match2011, G_Match2012, G_Match2013, G_Match2014,␣
 ↪G_Match2015, G_Match2016, G_Match2017, G_Match2018, G_Match2019,␣
 ↪G_Match2020]]
```

[412]:
```
print("="*100)
print(tabulate(output_gen, headers=["G2010", "G2011", "G2012", "G2013",␣
 ↪"G2014", "G2015", "G2016", "G2017", "G2018", "G2019", "G2020"]))
print("-"*100)
```

```
====================================================================================
====================
  G2010    G2011    G2012    G2013    G2014    G2015    G2016    G2017    G2018
G2019    G2020
-------  -------  -------  -------  -------  -------  -------  -------  -------
-------  -------
      1        0        0        1        1        1        1        1        1
1        1
------------------------------------------------------------------------------------
--------------------
```

[413]:
```
output_hse = [[H_Match2010, H_Match2011, H_Match2012, H_Match2013, H_Match2014,␣
↪H_Match2015, H_Match2016, H_Match2017, H_Match2018, H_Match2019,␣
↪H_Match2020]]
```

[414]:
```
print("="*100)
print(tabulate(output_hse, headers=["H2010", "H2011", "H2012", "H2013",␣
↪"H2014", "H2015", "H2016", "H2017", "H2018", "H2019", "H2020"]))
print("-"*100)
```

```
====================================================================================
====================
  H2010    H2011    H2012    H2013    H2014    H2015    H2016    H2017    H2018
H2019    H2020
-------  -------  -------  -------  -------  -------  -------  -------  -------
-------  -------
      1        1        1        1        1        1        1        1        1
1        1
------------------------------------------------------------------------------------
--------------------
```

[415]:
```
output_nar = [[NarDS2010, NarDS2011, NarDS2012, NarDS2013, NarDS2014,␣
↪NarDS2015, NarDS2016, NarDS2017, NarDS2018, NarDS2019, NarDS2020]]
```

[416]:
```
print("="*100)
print(tabulate(output_nar, headers=["2010", "2011", "2012", "2013", "2014",␣
↪"2015", "2016", "2017", "2018", "2019", "2020"]))
print("-"*100)
```

```
====================================================================================
====================
   2010      2011      2012     2013     2014     2015     2016     2017    2018
2019     2020
-------  --------  --------  -------  -------  -------  -------  -------  ------
-------  -------
3.35056  0.653447  0.464608  3.39351  3.25138  3.96625  4.06128  4.47165  4.4435
4.59562  4.82151
------------------------------------------------------------------------------------
--------------------
```

[417]:
```
output_oi = [[OutIn2010, OutIn2011, OutIn2012, OutIn2013, OutIn2014, OutIn2015,␣
↪OutIn2016, OutIn2017, OutIn2018, OutIn2019, OutIn2020]]
```

[418]:
```
print("="*100)
print(tabulate(output_oi, headers=["2010", "2011", "2012", "2013", "2014",␣
↪"2015", "2016", "2017", "2018", "2019", "2020"]))
print("-"*100)
```

```
====================================================================================
====================
```

```
  2010     2011     2012     2013     2014     2015     2016     2017     2018     2019
2020
------  ------  ------  ------  ------  ------  ------  ------  ------  ------
------
     0        0        0        2        1        1        3        9       17       36
35
--------------------------------------------------------------------------------
--------------------
```

TA-on-10K

May 26, 2022

[1]:
```
#pip install -U sec-edgar-downloader
```

[2]:
```
#pip install future
```

[ ]:
```

```

[3]:
```
# Importing neccessary packages and toolkits
```

[4]:
```
import re # Regular Expressions
```

[5]:
```
import nltk # Natural Language Processing Toolkit
```

[6]:
```
from nltk.tokenize import sent_tokenize
```

[7]:
```
from nltk.tokenize import word_tokenize
```

[8]:
```
from sec_edgar_downloader import Downloader # Download 10-K from EDGAR
```

[9]:
```
import tabulate # Addon used for table creation
```

[10]:
```
import codecs
```

[11]:
```
from __future__ import division, unicode_literals
```

[12]:
```
from bs4 import BeautifulSoup
```

[13]:
```
from nltk.corpus import stopwords
```

[14]:
```
from tabulate import tabulate
```

[ ]:
```

```

[15]:
```
# Defining path
```

[16]:
```
#dl = Downloader("/YOUR PATH") # Downloading the reports
```

[17]:
```
# Importing 10-K for a specific firm over a specified period using ticker
```

[18]:
```
#dl.get("10-K", "GNSS", after="2010-01-01", before="2021-12-31")
```

[ ]:
```

```

[19]:
```
# Importing 10-K (Computer Task Group Inc. example)
```

[20]:
```
file2010 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[21]:
```
file2011 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[22]:
```
file2012 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[23]:
```
file2013 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[24]:
```
file2014 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[25]:
```
file2015 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[26]:
```
file2016 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[27]:
```
file2017 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[28]:
```
file2018 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[29]:
```
file2019 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[30]:
```
file2020 = codecs.open("/YOUR PATH/filing-details.html", "r", 'utf-8')
```

[31]:
```
document2010 = BeautifulSoup(file2010.read()).get_text()
```

[32]:
```
document2011 = BeautifulSoup(file2011.read()).get_text()
```

[33]:
```
document2012 = BeautifulSoup(file2012.read()).get_text()
```

[34]:
```
document2013 = BeautifulSoup(file2013.read()).get_text()
```

[35]:
```
document2014 = BeautifulSoup(file2014.read()).get_text()
```

[36]:
```
document2015 = BeautifulSoup(file2015.read()).get_text()
```

[37]:
```
document2016 = BeautifulSoup(file2016.read()).get_text()
```

[38]:
```
document2017 = BeautifulSoup(file2017.read()).get_text()
```

[39]:
```
document2018 = BeautifulSoup(file2018.read()).get_text()
```

[40]:
```
document2019 = BeautifulSoup(file2019.read()).get_text()
```

[41]:
```
document2020 = BeautifulSoup(file2020.read()).get_text()
```

[ ]:
```

```

[42]:
```
# Tokenizing: splitting up the entire text into tokens
```

[43]:
```
tokenized2010 = word_tokenize(document2010)
```

[44]:
```
tokenized2011 = word_tokenize(document2011)
```

[45]:
```
tokenized2012 = word_tokenize(document2012)
```

[46]:
```
tokenized2013 = word_tokenize(document2013)
```

[47]:
```
tokenized2014 = word_tokenize(document2014)
```

[48]:
```
tokenized2015 = word_tokenize(document2015)
```

[49]:
```
tokenized2016 = word_tokenize(document2016)
```

[50]:
```
tokenized2017 = word_tokenize(document2017)
```

[51]:
```
tokenized2018 = word_tokenize(document2018)
```

[52]:
```
tokenized2019 = word_tokenize(document2019)
```

[53]:
```
tokenized2020 = word_tokenize(document2020)
```

[ ]:
```

```

[54]:
```
# Setting the stopwords to English vocabulary
```

[55]:
```
sw = set(stopwords.words("english"))
```

[ ]:
```

```

[56]:
```
# Removing stopwords
```

[57]:
```
filt_text2010 = [w for w in tokenized2010 if not w.lower() in sw]
```

[58]:
```
filt_text2011 = [w for w in tokenized2011 if not w.lower() in sw]
```

[59]:
```
filt_text2012 = [w for w in tokenized2012 if not w.lower() in sw]
```

[60]:
```
filt_text2013 = [w for w in tokenized2013 if not w.lower() in sw]
```

[61]:
```
filt_text2014 = [w for w in tokenized2014 if not w.lower() in sw]
```

[62]:
```
filt_text2015 = [w for w in tokenized2015 if not w.lower() in sw]
```

[63]:
```
filt_text2016 = [w for w in tokenized2016 if not w.lower() in sw]
```

[64]:
```
filt_text2017 = [w for w in tokenized2017 if not w.lower() in sw]
```

[65]:
```
filt_text2018 = [w for w in tokenized2018 if not w.lower() in sw]
```

[66]:
```
filt_text2019 = [w for w in tokenized2019 if not w.lower() in sw]
```

[67]:
```
filt_text2020 = [w for w in tokenized2020 if not w.lower() in sw]
```

[68]:
```
filt_text2010 = []
```

[69]:
```
filt_text2011 = []
```

[70]:
```
filt_text2012 = []
```

[71]:
```
filt_text2013 = []
```

[72]:
```
filt_text2014 = []
```

[73]:
```
filt_text2015 = []
```

[74]:
```
filt_text2016 = []
```

[75]:
```
filt_text2017 = []
```

[76]:
```
filt_text2018 = []
```

[77]:
```
filt_text2019 = []
```

[78]:
```
filt_text2020 = []
```

[79]:
```
for w in tokenized2010:
    if w not in sw:
        filt_text2010.append(w)
```

[80]:
```
for w in tokenized2011:
    if w not in sw:
        filt_text2011.append(w)
```

[81]:
```
for w in tokenized2012:
    if w not in sw:
        filt_text2012.append(w)
```

[82]:
```
for w in tokenized2013:
    if w not in sw:
```

```
        filt_text2013.append(w)
```

[83]:
```
for w in tokenized2014:
    if w not in sw:
        filt_text2014.append(w)
```

[84]:
```
for w in tokenized2015:
    if w not in sw:
        filt_text2015.append(w)
```

[85]:
```
for w in tokenized2016:
    if w not in sw:
        filt_text2016.append(w)
```

[86]:
```
for w in tokenized2017:
    if w not in sw:
        filt_text2017.append(w)
```

[87]:
```
for w in tokenized2018:
    if w not in sw:
        filt_text2018.append(w)
```

[88]:
```
for w in tokenized2019:
    if w not in sw:
        filt_text2019.append(w)
```

[89]:
```
for w in tokenized2020:
    if w not in sw:
        filt_text2020.append(w)
```

[ ]:
```

```

[90]:
```
# Removing years and special characters, and converting all text into small␣
↪lettercase
```

[91]:
```
removal = ["2010","2011", "2012", "2013", "2014", "2015", "2016", "2017",␣
↪"2018", "2019", "2020", "2021", "2025", "2030", "2050"]
```

[ ]:
```

```

[92]:
```
filt_text2010 = [w for w in filt_text2010 if w not in removal]
```

[93]:
```
filt_text2011 = [w for w in filt_text2011 if w not in removal]
```

[94]:
```
filt_text2012 = [w for w in filt_text2012 if w not in removal]
```

[95]:
```
filt_text2013 = [w for w in filt_text2013 if w not in removal]
```

[96]:
```
filt_text2014 = [w for w in filt_text2014 if w not in removal]
```

[97]:
```
filt_text2015 = [w for w in filt_text2015 if w not in removal]
```

[98]:
```
filt_text2016 = [w for w in filt_text2016 if w not in removal]
```

[99]:
```
filt_text2017 = [w for w in filt_text2017 if w not in removal]
```

[100]:
```
filt_text2018 = [w for w in filt_text2018 if w not in removal]
```

[101]:
```
filt_text2019 = [w for w in filt_text2019 if w not in removal]
```

[102]:
```
filt_text2020 = [w for w in filt_text2020 if w not in removal]
```

[ ]:
```

```

[103]:
```
new_text2010 = " ".join(filt_text2010)
```

[104]:
```
new_text2011 = " ".join(filt_text2011)
```

[105]:
```
new_text2012 = " ".join(filt_text2012)
```

[106]:
```
new_text2013 = " ".join(filt_text2013)
```

[107]:
```
new_text2014 = " ".join(filt_text2014)
```

[108]:
```
new_text2015 = " ".join(filt_text2015)
```

[109]:
```
new_text2016 = " ".join(filt_text2016)
```

[110]:
```
new_text2017 = " ".join(filt_text2017)
```

[111]:
```
new_text2018 = " ".join(filt_text2018)
```

[112]:
```
new_text2019 = " ".join(filt_text2019)
```

[113]:
```
new_text2020 = " ".join(filt_text2020)
```

[ ]:
```

```

[114]:
```
new_text2010 = new_text2010.lower()
```

[115]:
```
new_text2011 = new_text2011.lower()
```

[116]:
```
new_text2012 = new_text2012.lower()
```

[117]:
```
new_text2013 = new_text2013.lower()
```

[118]:
```
new_text2014 = new_text2014.lower()
```

[119]:
```
new_text2015 = new_text2015.lower()
```

[120]:
```
new_text2016 = new_text2016.lower()
```

[121]:
```
new_text2017 = new_text2017.lower()
```

[122]:
```
new_text2018 = new_text2018.lower()
```

[123]:
```
new_text2019 = new_text2019.lower()
```

[124]:
```
new_text2020 = new_text2020.lower()
```

[ ]:
```

```

[125]:
```
new_text2010 = re.sub(r'[^\w\s]', '', new_text2010)
```

[126]:
```
new_text2011 = re.sub(r'[^\w\s]', '', new_text2011)
```

[127]:
```
new_text2012 = re.sub(r'[^\w\s]', '', new_text2012)
```

[128]:
```
new_text2013 = re.sub(r'[^\w\s]', '', new_text2013)
```

[129]:
```
new_text2014 = re.sub(r'[^\w\s]', '', new_text2014)
```

[130]:
```
new_text2015 = re.sub(r'[^\w\s]', '', new_text2015)
```

[131]:
```
new_text2016 = re.sub(r'[^\w\s]', '', new_text2016)
```

[132]:
```
new_text2017 = re.sub(r'[^\w\s]', '', new_text2017)
```

[133]:
```
new_text2018 = re.sub(r'[^\w\s]', '', new_text2018)
```

[134]:
```
new_text2019 = re.sub(r'[^\w\s]', '', new_text2019)
```

[135]:
```
new_text2020 = re.sub(r'[^\w\s]', '', new_text2020)
```

[136]:
```
# print(new_text2014) # Controlling that the standard preprocessing procedures␣
↪work properly
```

[ ]:
```

```

[137]:
```
## Search Queries for Key Performance Indicators and Narrative Disclosure
```

[ ]:
```

```

[138]:
```
# Key Performance Indicators on the Impact on the External Environment
```

[139]:
```
Env_KPI = re.compile("(tonne|ton|tonnes|tons|kilo|kg)\s+(?:
↪\w+\s+){0,8}(GHG|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)\s+(?:\w+\s+){0,8}(\d+)",␣
↪flags=re.I)
```

[140]:
```
Env_KPI1 = re.compile("(tonne|ton|tonnes|tons|kilo|kg)\s+(?:
↪\w+\s+){0,8}(d+|%)\s+(?:\w+\s+){0,8}(GHG|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)", flags=re.I)
```

[141]:
```
Env_KPI2 = re.compile("(\d+)\s+(?:
↪\w+\s+){0,8}(tonne|ton|tonnes|tons|kilo|kg)\s+(?:\w+\s+){0,8}(GHG|CO2|CO␣
↪2|CO2e|CO 2E|N2O|carbon|greenhouse\s+gas|emissions|emission)", flags=re.I)
```

[142]:
```
Env_KPI3 = re.compile("(\d+)\s+(?:\w+\s+){0,8}(GHG|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)\s+(?:
↪\w+\s+){0,8}(tonne|ton|tonnes|tons|kilo|kg)", flags=re.I)
```

[143]:
```
Env_KPI4 = re.compile("(GHG|intesity|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)\s+(?:
↪\w+\s+){0,8}(tonne|ton|tonnes|tons|kilo|kg)\s+(?:\w+\s+){0,8}(\d+)",␣
↪flags=re.I)
```

[144]:
```
Env_KPI5 = re.compile("(GHG|intesity|CO2|CO 2|CO2e|CO␣
↪2E|N2O|carbon|greenhouse\s+gas|emissions|emission)\s+(?:
↪\w+\s+){0,8}(\d+)\s+(?:\w+\s+){0,8}(tonne|ton|tonnes|tons|kilo|kg)",␣
↪flags=re.I)
```

[145]:
```
Env_KPI6 = re.compile("(scope 1|scope 2|scope 3)\s+(?:
↪\w+\s+){0,8}(emission|emissions)\s+(?:
↪\w+\s+){0,8}(\d+|tonne|ton|tonnes|tons|kilo|kg)", flags=re.I)
```

[146]:
```
Env_KPI7 = re.compile("(\d+|tonne|ton|tonnes|tons|kilo|kg)\s+(?:
↪\w+\s+){0,8}(scope 1|scope 2|scope 3)\s+(?:
↪\w+\s+){0,8}(emission|emissions)", flags=re.I)
```

[147]:
```
Env_KPI8 = re.compile("(emission|emissions)\s+(?:\w+\s+){0,8}(scope 1|scope␣
↪2|scope 3)\s+(?:\w+\s+){0,8}(\d+|tonne|ton|tonnes|tons|kilo|kg)", flags=re.I)
```

[ ]:
```

```

[148]:
```
# Printing matches (can be adjusted for year and pattern)
```

[149]:
```
print(Env_KPI.findall(new_text2016), len(Env_KPI7.findall(new_text2016)))
```

```
[] 0
```

[ ]:
```

```

[150]:
```
# Aggregating all matches
```

[151]:
```
match_env2010 = len(Env_KPI8.findall(new_text2010)) + len(Env_KPI7.
↪findall(new_text2010)) + len(Env_KPI6.findall(new_text2010)) + len(Env_KPI5.
↪findall(new_text2010)) + len(Env_KPI4.findall(new_text2010)) + len(Env_KPI3.
↪findall(new_text2010)) + len(Env_KPI2.findall(new_text2010)) + len(Env_KPI1.
↪findall(new_text2010)) + len(Env_KPI.findall(new_text2010))
```

[152]:
```
match_env2011 = len(Env_KPI8.findall(new_text2011)) + len(Env_KPI7.
↪findall(new_text2011)) + len(Env_KPI6.findall(new_text2011)) + len(Env_KPI5.
↪findall(new_text2011)) + len(Env_KPI4.findall(new_text2011)) + len(Env_KPI3.
↪findall(new_text2011)) + len(Env_KPI2.findall(new_text2011)) + len(Env_KPI1.
↪findall(new_text2011)) + len(Env_KPI.findall(new_text2011))
```

[153]:
```
match_env2012 = len(Env_KPI8.findall(new_text2012)) + len(Env_KPI7.
↪findall(new_text2012)) + len(Env_KPI6.findall(new_text2012)) + len(Env_KPI5.
↪findall(new_text2012)) + len(Env_KPI4.findall(new_text2012)) + len(Env_KPI3.
↪findall(new_text2012)) + len(Env_KPI2.findall(new_text2012)) + len(Env_KPI1.
↪findall(new_text2012)) + len(Env_KPI.findall(new_text2012))
```

[154]:
```
match_env2013 = len(Env_KPI8.findall(new_text2013)) + len(Env_KPI7.
↪findall(new_text2013)) + len(Env_KPI6.findall(new_text2013)) + len(Env_KPI5.
↪findall(new_text2013)) + len(Env_KPI4.findall(new_text2013)) + len(Env_KPI3.
↪findall(new_text2013)) + len(Env_KPI2.findall(new_text2013)) + len(Env_KPI1.
↪findall(new_text2013)) + len(Env_KPI.findall(new_text2013))
```

[155]:
```
match_env2014 = len(Env_KPI8.findall(new_text2014)) + len(Env_KPI7.
↪findall(new_text2014)) + len(Env_KPI6.findall(new_text2014)) + len(Env_KPI5.
↪findall(new_text2014)) + len(Env_KPI4.findall(new_text2014)) + len(Env_KPI3.
↪findall(new_text2014)) + len(Env_KPI2.findall(new_text2014)) + len(Env_KPI1.
↪findall(new_text2014)) + len(Env_KPI.findall(new_text2014))
```

[156]:
```
match_env2015 = len(Env_KPI8.findall(new_text2015)) + len(Env_KPI7.
↪findall(new_text2015)) + len(Env_KPI6.findall(new_text2015)) + len(Env_KPI5.
↪findall(new_text2015)) + len(Env_KPI4.findall(new_text2015)) + len(Env_KPI3.
↪findall(new_text2015)) + len(Env_KPI2.findall(new_text2015)) + len(Env_KPI1.
↪findall(new_text2015)) + len(Env_KPI.findall(new_text2015))
```

[157]:
```
match_env2016 = len(Env_KPI8.findall(new_text2016)) + len(Env_KPI7.
↪findall(new_text2016)) + len(Env_KPI6.findall(new_text2016)) + len(Env_KPI5.
↪findall(new_text2016)) + len(Env_KPI4.findall(new_text2016)) + len(Env_KPI3.
↪findall(new_text2016)) + len(Env_KPI2.findall(new_text2016)) + len(Env_KPI1.
↪findall(new_text2016)) + len(Env_KPI.findall(new_text2016))
```

[158]:
```

```

```
match_env2017 = len(Env_KPI8.findall(new_text2017)) + len(Env_KPI7.
↪findall(new_text2017)) + len(Env_KPI6.findall(new_text2017)) + len(Env_KPI5.
↪findall(new_text2017)) + len(Env_KPI4.findall(new_text2017)) + len(Env_KPI3.
↪findall(new_text2017)) + len(Env_KPI2.findall(new_text2017)) + len(Env_KPI1.
↪findall(new_text2017)) + len(Env_KPI.findall(new_text2017))
```

[159]:
```
match_env2018 = len(Env_KPI8.findall(new_text2018)) + len(Env_KPI7.
↪findall(new_text2018)) + len(Env_KPI6.findall(new_text2018)) + len(Env_KPI5.
↪findall(new_text2018)) + len(Env_KPI4.findall(new_text2018)) + len(Env_KPI3.
↪findall(new_text2018)) + len(Env_KPI2.findall(new_text2018)) + len(Env_KPI1.
↪findall(new_text2018)) + len(Env_KPI.findall(new_text2018))
```

[160]:
```
match_env2019 = len(Env_KPI8.findall(new_text2019)) + len(Env_KPI7.
↪findall(new_text2019)) + len(Env_KPI6.findall(new_text2019)) + len(Env_KPI5.
↪findall(new_text2019)) + len(Env_KPI4.findall(new_text2019)) + len(Env_KPI3.
↪findall(new_text2019)) + len(Env_KPI2.findall(new_text2019)) + len(Env_KPI1.
↪findall(new_text2019)) + len(Env_KPI.findall(new_text2019))
```

[161]:
```
match_env2020 = len(Env_KPI8.findall(new_text2020)) + len(Env_KPI7.
↪findall(new_text2020)) + len(Env_KPI6.findall(new_text2020)) + len(Env_KPI5.
↪findall(new_text2020)) + len(Env_KPI4.findall(new_text2020)) + len(Env_KPI3.
↪findall(new_text2020)) + len(Env_KPI2.findall(new_text2020)) + len(Env_KPI1.
↪findall(new_text2020)) + len(Env_KPI.findall(new_text2020))
```

[ ]:
```

```

[162]:
```
# Creating a binary variable
```

[163]:
```
if (match_env2010 > 0):
    E_Match2010 = "1"
else:
    E_Match2010 = "0"
print(E_Match2010)
```

```
0
```

[164]:
```
if (match_env2011 > 0):
    E_Match2011 = "1"
else:
    E_Match2011 = "0"
print(E_Match2011)
```

```
0
```

[165]:
```
if (match_env2012 > 0):
    E_Match2012 = "1"
else:
    E_Match2012 = "0"
```

```
print(E_Match2012)
```

0

[166]:
```
if (match_env2013 > 0):
    E_Match2013 = "1"
else:
    E_Match2013 = "0"
print(E_Match2013)
```

0

[167]:
```
if (match_env2014 > 0):
    E_Match2014 = "1"
else:
    E_Match2014 = "0"
print(E_Match2014)
```

0

[168]:
```
if (match_env2015 > 0):
    E_Match2015 = "1"
else:
    E_Match2015 = "0"
print(E_Match2015)
```

0

[169]:
```
if (match_env2016 > 0):
    E_Match2016 = "1"
else:
    E_Match2016 = "0"
print(E_Match2016)
```

0

[170]:
```
if (match_env2017 > 0):
    E_Match2017 = "1"
else:
    E_Match2017 = "0"
print(E_Match2017)
```

0

[171]:
```
if (match_env2018 > 0):
    E_Match2018 = "1"
else:
    E_Match2018 = "0"
print(E_Match2018)
```

0

[172]:
```
if (match_env2019 > 0):
    E_Match2019 = "1"
else:
    E_Match2019 = "0"
print(E_Match2019)
```

```
0
```

[173]:
```
if (match_env2020 > 0):
    E_Match2020 = "1"
else:
    E_Match2020 = "0"
print(E_Match2020)
```

```
0
```

[ ]:
```
```

[174]:
```
# Key Performance Indicators on Gender Equality
```

[175]:
```
Gen_KPI = re.compile("(female|gender|woman|women|sex)\s+(?:
↪\w+\s+){0,8}(board|director|directors|executive|executives|manager|managers|management|mane
↪:\w+\s+){0,8}(\d+|one|two|three|four|five|six|seven|eigth|nine)", flags=re.I)
```

[176]:
```
Gen_KPI1 = re.compile("(\d+|one|two|three|four|five|six|seven|eigth|nine)\s+(?:
↪\w+\s+){0,8}(board|director|directors|executive|executives|manager|managers|management|mane
↪:\w+\s+){0,8}(female|gender|woman|women|sex)", flags=re.I)
```

[177]:
```
Gen_KPI2 = re.compile("(female|gender|woman|women|sex)\s+(?:
↪\w+\s+){0,8}(\d+|%|one|two|three|four|five|six|seven|eigth|nine)\s+(?:
↪\w+\s+){0,8}(board|director|directors|executive|executives|manager|managers|management|mane
↪flags=re.I)
```

[178]:
```
Gen_KPI3 = re.
↪compile("(board|director|directors|executive|executives|manager|managers|management|maneger
↪:\w+\s+){0,8}(\d+|%|one|two|three|four|five|six|seven|eigth|nine)\s+(?:
↪\w+\s+){0,8}(female|gender|woman|women|sex)", flags=re.I)
```

[179]:
```
Gen_KPI4 = re.compile("(\d+|%|one|two|three|four|five|six|seven|eigth|nine)\s+(?
↪:\w+\s+){0,8}(female|gender|woman|women|sex)\s+(?:
↪\w+\s+){0,8}(board|director|directors|executive|executives|manager|managers|management|mane
↪flags=re.I)
```

[180]:
```
Gen_KPI5 = re.
↪compile("(board|director|directors|executive|executives|manager|managers|management|maneger
↪:\w+\s+){0,8}(female|gender|woman|women|sex)\s+(?:
↪\w+\s+){0,8}(\d+|%|one|two|three|four|five|six|seven|eigth|nine)", flags=re.
↪I)
```

[181]:
```
Gen_KPI6 = re.compile("(gender|sex|female|women|woman)\s+(?:
↪\w+\s+){0,8}(split|diversity|ratio|distribution|composition|percentage|breakdown)\s+(?
↪:\w+\s+){0,8}(\d+|%)", flags=re.I)
```

[182]:
```
Gen_KPI7 = re.compile("(gender|sex|female|women|woman)\s+(?:
↪\w+\s+){0,8}(\d+)\s+(?:
↪\w+\s+){0,8}(split|diversity|ratio|distribution|composition|percentage|breakdown)",␣
↪flags=re.I)
```

[183]:
```
Gen_KPI8 = re.
↪compile("(split|diversity|ratio|distribution|composition|percentage|breakdown)\s+(?
↪:\w+\s+){0,8}(gender|sex|female|women|woman)\s+(?:\w+\s+){0,8}(\d+|%)",␣
↪flags=re.I)
```

[184]:
```
Gen_KPI9 = re.
↪compile("(split|diversity|ratio|distribution|composition|percentage|breakdown)\s+(?
↪:\w+\s+){0,8}(\d+|%)\s+(?:\w+\s+){0,8}(gender|sex|female|women|woman)",␣
↪flags=re.I)
```

[185]:
```
Gen_KPI10 = re.compile("(\d+|%)\s+(?:
↪\w+\s+){0,8}(split|diversity|ratio|distribution|composition|percentage|breakdown)\s+(?
↪:\w+\s+){0,8}(gender|sex|female|women|woman)", flags=re.I)
```

[186]:
```
Gen_KPI11 = re.compile("(\d+|%)\s+(?:
↪\w+\s+){0,8}(gender|sex|female|women|woman)\s+(?:
↪\w+\s+){0,8}(split|diversity|ratio|distribution|composition|percentage|breakdown)",␣
↪flags=re.I)
```

[ ]:
```

```

[187]:
```
# Printing matches (can be adjusted for year and pattern)
```

[188]:
```
print(Gen_KPI.findall(new_text2010))
```

```
[]
```

[ ]:
```

```

[189]:
```
# Aggregating all matches
```

[190]:
```
match_gen2010 = len(Gen_KPI11.findall(new_text2010)) + len(Gen_KPI10.
↪findall(new_text2010)) + len(Gen_KPI9.findall(new_text2010)) + len(Gen_KPI8.
↪findall(new_text2010))  + len(Gen_KPI7.findall(new_text2010)) + len(Gen_KPI6.
↪findall(new_text2010)) + len(Gen_KPI5.findall(new_text2010)) + len(Gen_KPI4.
↪findall(new_text2010)) + len(Gen_KPI3.findall(new_text2010)) + len(Gen_KPI2.
↪findall(new_text2010)) + len(Gen_KPI1.findall(new_text2010)) + len(Gen_KPI.
↪findall(new_text2010))
```

[191]:
```
match_gen2011 = len(Gen_KPI11.findall(new_text2011)) + len(Gen_KPI10.
↪findall(new_text2011)) + len(Gen_KPI9.findall(new_text2011)) + len(Gen_KPI8.
↪findall(new_text2011))  + len(Gen_KPI7.findall(new_text2011)) + len(Gen_KPI6.
↪findall(new_text2011)) + len(Gen_KPI5.findall(new_text2011)) + len(Gen_KPI4.
↪findall(new_text2011)) + len(Gen_KPI3.findall(new_text2011)) + len(Gen_KPI2.
↪findall(new_text2011)) + len(Gen_KPI1.findall(new_text2011)) + len(Gen_KPI.
↪findall(new_text2011))
```

[192]:
```
match_gen2012 = len(Gen_KPI11.findall(new_text2012)) + len(Gen_KPI10.
↪findall(new_text2012)) + len(Gen_KPI9.findall(new_text2012)) + len(Gen_KPI8.
↪findall(new_text2012))  + len(Gen_KPI7.findall(new_text2012)) + len(Gen_KPI6.
↪findall(new_text2012)) + len(Gen_KPI5.findall(new_text2012)) + len(Gen_KPI4.
↪findall(new_text2012)) + len(Gen_KPI3.findall(new_text2012)) + len(Gen_KPI2.
↪findall(new_text2012)) + len(Gen_KPI1.findall(new_text2012)) + len(Gen_KPI.
↪findall(new_text2012))
```

[193]:
```
match_gen2013 = len(Gen_KPI11.findall(new_text2013)) + len(Gen_KPI10.
↪findall(new_text2013)) + len(Gen_KPI9.findall(new_text2013)) + len(Gen_KPI8.
↪findall(new_text2013))  + len(Gen_KPI7.findall(new_text2013)) + len(Gen_KPI6.
↪findall(new_text2013)) + len(Gen_KPI5.findall(new_text2013)) + len(Gen_KPI4.
↪findall(new_text2013)) + len(Gen_KPI3.findall(new_text2013)) + len(Gen_KPI2.
↪findall(new_text2013)) + len(Gen_KPI1.findall(new_text2013)) + len(Gen_KPI.
↪findall(new_text2013))
```

[194]:
```
match_gen2014 = len(Gen_KPI11.findall(new_text2014)) + len(Gen_KPI10.
↪findall(new_text2014)) + len(Gen_KPI9.findall(new_text2014)) + len(Gen_KPI8.
↪findall(new_text2014))  + len(Gen_KPI7.findall(new_text2014)) + len(Gen_KPI6.
↪findall(new_text2014)) + len(Gen_KPI5.findall(new_text2014)) + len(Gen_KPI4.
↪findall(new_text2014)) + len(Gen_KPI3.findall(new_text2014)) + len(Gen_KPI2.
↪findall(new_text2014)) + len(Gen_KPI1.findall(new_text2014)) + len(Gen_KPI.
↪findall(new_text2014))
```

[195]:
```
match_gen2015 = len(Gen_KPI11.findall(new_text2015)) + len(Gen_KPI10.
↪findall(new_text2015)) + len(Gen_KPI9.findall(new_text2015)) + len(Gen_KPI8.
↪findall(new_text2015))  + len(Gen_KPI7.findall(new_text2015)) + len(Gen_KPI6.
↪findall(new_text2015)) + len(Gen_KPI5.findall(new_text2015)) + len(Gen_KPI4.
↪findall(new_text2015)) + len(Gen_KPI3.findall(new_text2015)) + len(Gen_KPI2.
↪findall(new_text2015)) + len(Gen_KPI1.findall(new_text2015)) + len(Gen_KPI.
↪findall(new_text2015))
```

[196]:
```
match_gen2016 = len(Gen_KPI11.findall(new_text2016)) + len(Gen_KPI10.
↪findall(new_text2016)) + len(Gen_KPI9.findall(new_text2016)) + len(Gen_KPI8.
↪findall(new_text2016))  + len(Gen_KPI7.findall(new_text2016)) + len(Gen_KPI6.
↪findall(new_text2016)) + len(Gen_KPI5.findall(new_text2016)) + len(Gen_KPI4.
↪findall(new_text2016)) + len(Gen_KPI3.findall(new_text2016)) + len(Gen_KPI2.
↪findall(new_text2016)) + len(Gen_KPI1.findall(new_text2016)) + len(Gen_KPI.
↪findall(new_text2016))
```

[197]:
```
match_gen2017 = len(Gen_KPI11.findall(new_text2017)) + len(Gen_KPI10.
↪findall(new_text2017)) + len(Gen_KPI9.findall(new_text2017)) + len(Gen_KPI8.
↪findall(new_text2017))  + len(Gen_KPI7.findall(new_text2017)) + len(Gen_KPI6.
↪findall(new_text2017)) + len(Gen_KPI5.findall(new_text2017)) + len(Gen_KPI4.
↪findall(new_text2017)) + len(Gen_KPI3.findall(new_text2017)) + len(Gen_KPI2.
↪findall(new_text2017)) + len(Gen_KPI1.findall(new_text2017)) + len(Gen_KPI.
↪findall(new_text2017))
```

[198]:
```
match_gen2018 = len(Gen_KPI11.findall(new_text2018)) + len(Gen_KPI10.
↪findall(new_text2018)) + len(Gen_KPI9.findall(new_text2018)) + len(Gen_KPI8.
↪findall(new_text2018))  + len(Gen_KPI7.findall(new_text2018)) + len(Gen_KPI6.
↪findall(new_text2018)) + len(Gen_KPI5.findall(new_text2018)) + len(Gen_KPI4.
↪findall(new_text2018)) + len(Gen_KPI3.findall(new_text2018)) + len(Gen_KPI2.
↪findall(new_text2018)) + len(Gen_KPI1.findall(new_text2018)) + len(Gen_KPI.
↪findall(new_text2018))
```

[199]:
```
match_gen2019 = len(Gen_KPI11.findall(new_text2019)) + len(Gen_KPI10.
↪findall(new_text2019)) + len(Gen_KPI9.findall(new_text2019)) + len(Gen_KPI8.
↪findall(new_text2019))  + len(Gen_KPI7.findall(new_text2019)) + len(Gen_KPI6.
↪findall(new_text2019)) + len(Gen_KPI5.findall(new_text2019)) + len(Gen_KPI4.
↪findall(new_text2019)) + len(Gen_KPI3.findall(new_text2019)) + len(Gen_KPI2.
↪findall(new_text2019)) + len(Gen_KPI1.findall(new_text2019)) + len(Gen_KPI.
↪findall(new_text2019))
```

[200]:
```
match_gen2020 = len(Gen_KPI11.findall(new_text2020)) + len(Gen_KPI10.
↪findall(new_text2020)) + len(Gen_KPI9.findall(new_text2020)) + len(Gen_KPI8.
↪findall(new_text2020))  + len(Gen_KPI7.findall(new_text2020)) + len(Gen_KPI6.
↪findall(new_text2020)) + len(Gen_KPI5.findall(new_text2020)) + len(Gen_KPI4.
↪findall(new_text2020)) + len(Gen_KPI3.findall(new_text2020)) + len(Gen_KPI2.
↪findall(new_text2020)) + len(Gen_KPI1.findall(new_text2020)) + len(Gen_KPI.
↪findall(new_text2020))
```

[ ]:
```

```

[201]:
```
# Creating a binary variable
```

[202]:
```
if (match_gen2010 > 0):
    G_Match2010 = "1"
else:
    G_Match2010 = "0"
print(G_Match2010)
```

```
0
```

[203]:
```
if (match_gen2011 > 0):
    G_Match2011 = "1"
else:
    G_Match2011 = "0"
```

```
print(G_Match2011)
```

0

[204]:
```
if (match_gen2012 > 0):
    G_Match2012 = "1"
else:
    G_Match2012 = "0"
print(G_Match2012)
```

0

[205]:
```
if (match_gen2013 > 0):
    G_Match2013 = "1"
else:
    G_Match2013 = "0"
print(G_Match2013)
```

0

[206]:
```
if (match_gen2014 > 0):
    G_Match2014 = "1"
else:
    G_Match2014 = "0"
print(G_Match2014)
```

0

[207]:
```
if (match_gen2015 > 0):
    G_Match2015 = "1"
else:
    G_Match2015 = "0"
print(G_Match2015)
```

0

[208]:
```
if (match_gen2016 > 0):
    G_Match2016 = "1"
else:
    G_Match2016 = "0"
print(G_Match2016)
```

0

[209]:
```
if (match_gen2017 > 0):
    G_Match2017 = "1"
else:
    G_Match2017 = "0"
print(G_Match2017)
```

0

[210]:
```
if (match_gen2018 > 0):
    G_Match2018 = "1"
else:
    G_Match2018 = "0"
print(G_Match2018)
```

```
0
```

[211]:
```
if (match_gen2019 > 0):
    G_Match2019 = "1"
else:
    G_Match2019 = "0"
print(G_Match2019)
```

```
0
```

[212]:
```
if (match_gen2020 > 0):
    G_Match2020 = "1"
else:
    G_Match2020 = "0"
print(G_Match2020)
```

```
0
```

[ ]:
```

```

[213]:
```
# Key Performance Indicators on Work Environment
```

[214]:
```
HSE_KPI = re.compile("(number|total|reported|reportable)\s+(?:
↪\w+\s+){0,8}(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sick
↪absence|sick leave)\s+(?:\w+\s+){0,8}(\d+|no|none)", flags=re.I)
```

[215]:
```
HSE_KPI1 = re.compile("(\d+|no|none)\s+(?:
↪\w+\s+){0,8}(number|total|reported|reportable)\s+(?:
↪\w+\s+){0,8}(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sick
↪absence|sick leave)", flags=re.I)
```

[216]:
```
HSE_KPI2 = re.
↪compile("(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sicknes
↪absence|sick leave)\s+(?:\w+\s+){0,8}(\d+|no|none)", flags=re.I)
```

[217]:
```
HSE_KPI3 = re.compile("(number|total|reported|reportable)\s+(?:
↪\w+\s+){0,8}(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sick
↪absence|sick leave)", flags=re.I)
```

[218]:
```
HSE_KPI4 = re.
↪compile("(injuries|injury|accidents|incidents|accident|incident|fatalities|fatality|sicknes
↪absence|sick leave)\s+(?:\w+\s+){0,8}(number|total|reported|reportable)",␣
↪flags=re.I)
```

[219]:
```
HSE_KPI5 = re.
↪compile("(TRI|OSHA|TCIR|TRIR|TRIFR|TRCF|AIFR|AFR|LTI|LTIF|LTIR|LTIFR|LWDR)\s+(?
↪:\w+\s+){0,8}(\d+)", flags=re.I)
```

[ ]:
```

```

[220]:
```
# Printing matches (can be adjusted for year and pattern)
```

[221]:
```
print(HSE_KPI.findall(new_text2010))
```

```
[]
```

[ ]:
```

```

[222]:
```
# Aggregating all matches
```

[223]:
```
match_HSE2010 = len(HSE_KPI5.findall(new_text2010)) + len(HSE_KPI4.
↪findall(new_text2010)) + len(HSE_KPI3.findall(new_text2010)) + len(HSE_KPI2.
↪findall(new_text2010)) + len(HSE_KPI1.findall(new_text2010)) + len(HSE_KPI.
↪findall(new_text2010))
```

[224]:
```
match_HSE2011 = len(HSE_KPI5.findall(new_text2011)) + len(HSE_KPI4.
↪findall(new_text2011)) + len(HSE_KPI3.findall(new_text2011)) + len(HSE_KPI2.
↪findall(new_text2011)) + len(HSE_KPI1.findall(new_text2011)) + len(HSE_KPI.
↪findall(new_text2011))
```

[225]:
```
match_HSE2012 = len(HSE_KPI5.findall(new_text2012)) + len(HSE_KPI4.
↪findall(new_text2012)) + len(HSE_KPI3.findall(new_text2012)) + len(HSE_KPI2.
↪findall(new_text2012)) + len(HSE_KPI1.findall(new_text2012)) + len(HSE_KPI.
↪findall(new_text2012))
```

[226]:
```
match_HSE2013 = len(HSE_KPI5.findall(new_text2013)) + len(HSE_KPI4.
↪findall(new_text2013)) + len(HSE_KPI3.findall(new_text2013)) + len(HSE_KPI2.
↪findall(new_text2013)) + len(HSE_KPI1.findall(new_text2013)) + len(HSE_KPI.
↪findall(new_text2013))
```

[227]:
```
match_HSE2014 = len(HSE_KPI5.findall(new_text2014)) + len(HSE_KPI4.
↪findall(new_text2014)) + len(HSE_KPI3.findall(new_text2014)) + len(HSE_KPI2.
↪findall(new_text2014)) + len(HSE_KPI1.findall(new_text2014)) + len(HSE_KPI.
↪findall(new_text2014))
```

[228]:
```
match_HSE2015 = len(HSE_KPI5.findall(new_text2015)) + len(HSE_KPI4.
↪findall(new_text2015)) + len(HSE_KPI3.findall(new_text2015)) + len(HSE_KPI2.
↪findall(new_text2015)) + len(HSE_KPI1.findall(new_text2015)) + len(HSE_KPI.
↪findall(new_text2015))
```

[229]:
```

```

```
match_HSE2016 = len(HSE_KPI5.findall(new_text2016)) + len(HSE_KPI4.
↪findall(new_text2016)) + len(HSE_KPI3.findall(new_text2016)) + len(HSE_KPI2.
↪findall(new_text2016)) + len(HSE_KPI1.findall(new_text2016)) + len(HSE_KPI.
↪findall(new_text2016))
```

[230]:
```
match_HSE2017 = len(HSE_KPI5.findall(new_text2017)) + len(HSE_KPI4.
↪findall(new_text2017)) + len(HSE_KPI3.findall(new_text2017)) + len(HSE_KPI2.
↪findall(new_text2017)) + len(HSE_KPI1.findall(new_text2017)) + len(HSE_KPI.
↪findall(new_text2017))
```

[231]:
```
match_HSE2018 = len(HSE_KPI5.findall(new_text2018)) + len(HSE_KPI4.
↪findall(new_text2018)) + len(HSE_KPI3.findall(new_text2018)) + len(HSE_KPI2.
↪findall(new_text2018)) + len(HSE_KPI1.findall(new_text2018)) + len(HSE_KPI.
↪findall(new_text2018))
```

[232]:
```
match_HSE2019 = len(HSE_KPI5.findall(new_text2019)) + len(HSE_KPI4.
↪findall(new_text2019)) + len(HSE_KPI3.findall(new_text2019)) + len(HSE_KPI2.
↪findall(new_text2019)) + len(HSE_KPI1.findall(new_text2019)) + len(HSE_KPI.
↪findall(new_text2019))
```

[233]:
```
match_HSE2020 = len(HSE_KPI5.findall(new_text2020)) + len(HSE_KPI4.
↪findall(new_text2020)) + len(HSE_KPI3.findall(new_text2020)) + len(HSE_KPI2.
↪findall(new_text2020)) + len(HSE_KPI1.findall(new_text2020)) + len(HSE_KPI.
↪findall(new_text2020))
```

[ ]:
```

```

[234]:
```
# Creating a binary variable
```

[235]:
```
if (match_HSE2010 > 0):
    H_Match2010 = "1"
else:
    H_Match2010 = "0"
print(H_Match2010)
```

```
0
```

[236]:
```
if (match_HSE2011 > 0):
    H_Match2011 = "1"
else:
    H_Match2011 = "0"
print(H_Match2011)
```

```
0
```

[237]:
```
if (match_HSE2012 > 0):
    H_Match2012 = "1"
else:
```

```
    H_Match2012 = "0"
print(H_Match2012)
```

```
0
```

[238]:
```
if (match_HSE2013 > 0):
    H_Match2013 = "1"
else:
    H_Match2013 = "0"
print(H_Match2013)
```

```
0
```

[239]:
```
if (match_HSE2014 > 0):
    H_Match2014 = "1"
else:
    H_Match2014 = "0"
print(H_Match2014)
```

```
0
```

[240]:
```
if (match_HSE2015 > 0):
    H_Match2015 = "1"
else:
    H_Match2015 = "0"
print(H_Match2015)
```

```
0
```

[241]:
```
if (match_HSE2016 > 0):
    H_Match2016 = "1"
else:
    H_Match2016 = "0"
print(H_Match2016)
```

```
0
```

[242]:
```
if (match_HSE2017 > 0):
    H_Match2017 = "1"
else:
    H_Match2017 = "0"
print(H_Match2017)
```

```
1
```

[243]:
```
if (match_HSE2018 > 0):
    H_Match2018 = "1"
else:
    H_Match2018 = "0"
print(H_Match2018)
```

```
1
```

[244]:
```
if (match_HSE2019 > 0):
    H_Match2019 = "1"
else:
    H_Match2019 = "0"
print(H_Match2019)
```

```
1
```

[245]:
```
if (match_HSE2020 > 0):
    H_Match2020 = "1"
else:
    H_Match2020 = "0"
print(H_Match2020)
```

```
1
```

[ ]:
```

```

[246]:
```
## Narrative Disclosure Measures
```

[ ]:
```

```

[247]:
```
# Outside-in Impact
```

[248]:
```
E_Out = re.compile("(climate|planet|transition|physical)\s+(?:
↪\w+\s+){0,8}(risk|risks|impact|impacts)\s+(?:\w+\s+\D){0,5}", flags=re.I)
```

[ ]:
```

```

[249]:
```
# Counting number of matches (converted to binary variable later)
```

[250]:
```
OutIn2010 = len(E_Out.findall(new_text2010))
```

[251]:
```
OutIn2011 = len(E_Out.findall(new_text2011))
```

[252]:
```
OutIn2012 = len(E_Out.findall(new_text2012))
```

[253]:
```
OutIn2013 = len(E_Out.findall(new_text2013))
```

[254]:
```
OutIn2014 = len(E_Out.findall(new_text2014))
```

[255]:
```
OutIn2015 = len(E_Out.findall(new_text2015))
```

[256]:
```
OutIn2016 = len(E_Out.findall(new_text2016))
```

[257]:
```
OutIn2017 = len(E_Out.findall(new_text2017))
```

[258]:
```
OutIn2018 = len(E_Out.findall(new_text2018))
```

[259]:
```
OutIn2019 = len(E_Out.findall(new_text2019))
```

[260]:
```
OutIn2020 = len(E_Out.findall(new_text2020))
```

[ ]:
```

```

[261]:
```
# Governance Dimension
```

[262]:
```
Nar_G = re.
 ↪compile("(governance|conduct|misconduct|audit|audited|auditing|control|controls|oversee|ove
 ↪structure|board member|board␣
 ↪members|composition|independence|independent|succession|tenure|vacancies|vacancy|nomination
 ↪ethics|business ethic|ethics|ethical)", flags = re.I)
```

[ ]:
```

```

[263]:
```
# Printing all matches
```

[264]:
```
print(Nar_G.findall(new_text2014), len(Nar_G.findall(new_text2014)))
```

```
['disclosure', 'proxy', 'voting', 'voting', 'proxy', 'disclosure', 'disclosure',
'disclosure', 'control', 'governance', 'compensation', 'independence',
'control', 'talent', 'election', 'vest', 'vest', 'recruit', 'talent', 'talent',
'recruit', 'recruit', 'independent', 'salary', 'vest', 'attract', 'recruit',
'recruit', 'recruit', 'audit', 'disclose', 'control', 'audit', 'control',
'ethics', 'conduct', 'governance', 'investor', 'governance', 'disclose',
'conduct', 'recruit', 'control', 'control', 'evaluate', 'evaluate', 'fraud',
'disclosure', 'vest', 'compensation', 'award', 'compliance', 'recruit',
'conduct', 'control', 'governance', 'disclosure', 'compliance', 'disclosure',
'governance', 'governance', 'disclosure', 'comply', 'compliance', 'comply',
'control', 'independent', 'audit', 'audit', 'control', 'audit', 'audit',
'evaluate', 'shareholder', 'shareholder', 'vest', 'proposals', 'control',
'control', 'control', 'control', 'control', 'fraud', 'control', 'control',
'control', 'control', 'disclosure', 'compensation', 'vest', 'disclosure',
'compliance', 'compensation', 'award', 'shareholder', 'vest', 'audit', 'audit',
'shareholder', 'control', 'talent', 'salary', 'fraud', 'incentive',
'compensation', 'incentive', 'compensation', 'compensation', 'incentive',
'compensation', 'disclosure', 'control', 'control', 'compensation', 'evaluate',
'disclosure', 'compensation', 'vest', 'compensation', 'compensation',
'compensation', 'fraud', 'compensation', 'electing', 'compensation',
'incentive', 'compensation', 'incentive', 'compensation', 'vesting',
'compensation', 'vesting', 'payout', 'compliance', 'vest', 'disclosure',
'disclosure', 'vest', 'disclosure', 'compensation', 'compensation', 'recruit',
'compensation', 'compensation', 'compensation', 'compensation', 'disclosure',
'conduct', 'independent', 'shareholder', 'audit', 'shareholder', 'audit',
'conduct', 'audit', 'audit', 'audit', 'disclosure', 'audit', 'audit', 'audit',
```

'control', 'control', 'control', 'pension', 'vest', 'shareholder',
'compensation', 'compensation', 'shareholder', 'shareholder', 'shareholder',
'compensation', 'compensation', 'compensation', 'vesting', 'compensation',
'vest', 'vesting', 'compensation', 'shareholder', 'shareholder', 'compensation',
'compensation', 'compensation', 'pension', 'compensation', 'compensation',
'compensation', 'pension', 'shareholder', 'compensation', 'compensation',
'compensation', 'pension', 'disclosure', 'vest', 'salary', 'vest', 'evaluate',
'reviewed', 'evaluate', 'compensation', 'compensation', 'compensation', 'award',
'compensation', 'award', 'award', 'compensation', 'compensation', 'award',
'vest', 'compensation', 'vest', 'pension', 'pension', 'compensation', 'vest',
'disclosure', 'fraud', 'vest', 'vest', 'compensation', 'vest', 'compliance',
'compensation', 'vest', 'incentive', 'compensation', 'compensation',
'compensation', 'pension', 'pension', 'pension', 'pension', 'pension', 'salary',
'pension', 'compensation', 'pension', 'vest', 'vest', 'vesting', 'pension',
'compensation', 'vest', 'vest', 'vest', 'voting', 'election', 'compensation',
'compensation', 'voting', 'election', 'shareholder', 'compensation',
'shareholder', 'compensation', 'compensation', 'compensation', 'compensation',
'compensation', 'award', 'award', 'compensation', 'compensation', 'award',
'vest', 'compensation', 'compensation', 'compensation', 'compensation',
'shareholder', 'award', 'award', 'award', 'compensation', 'vesting', 'award',
'award', 'award', 'shareholder', 'award', 'award', 'award', 'compensation',
'vesting', 'award', 'award', 'vest', 'shareholder', 'vesting', 'vest',
'vesting', 'vesting', 'vest', 'compensation', 'independent', 'vest', 'vote',
'award', 'vest', 'vest', 'compensation', 'vest', 'compensation', 'audit',
'audit', 'audit', 'audit', 'disclosure', 'disclose', 'audit', 'disclosure',
'control', 'disclosure', 'control', 'evaluate', 'disclosure', 'control',
'disclosure', 'control', 'control', 'control', 'control', 'control', 'control',
'control', 'conduct', 'control', 'control', 'control', 'audit', 'control',
'independent', 'control', 'independent', 'shareholder', 'audit', 'control',
'control', 'control', 'control', 'control', 'control', 'audit', 'conduct',
'audit', 'audit', 'control', 'audit', 'control', 'control', 'audit', 'audit',
'control', 'control', 'control', 'control', 'compliance', 'control', 'control',
'audit', 'shareholder', 'control', 'control', 'evaluate', 'control', 'control',
'control', 'compensation', 'reviewed', 'compensation', 'compensation',
'compensation', 'control', 'control', 'vest', 'award', 'control', 'control',
'control', 'salary', 'incentive', 'control', 'voting', 'election', 'board
member', 'vote', 'board member', 'board member', 'governance', 'election',
'compliance', 'audit', 'governance', 'nomination', 'governance', 'proxy',
'proxy', 'compensation', 'compensation', 'compensation', 'proxy', 'proxy',
'compensation', 'compensation', 'compensation', 'award', 'award',
'compensation', 'award', 'independence', 'audit', 'independence', 'proxy',
'audit', 'proxy', 'independent', 'shareholder', 'independent', 'independent',
'shareholder', 'shareholder', 'audit', 'audit', 'audit', 'compensation',
'compensation', 'compensation', 'compensation', 'compensation', 'award',
'compensation', 'control', 'control', 'proxy', 'compensation', 'incentive',
'incentive', 'compensation', 'award', 'compensation', 'control', 'ethics',
'proxy', 'shareholder', 'proxy', 'shareholder', 'proxy', 'shareholder', 'proxy',
'governance'] 454

```
[ ]:

[265]: # Computing the Ratio ESG Dictionary (G) / Report Length

[266]: NarGDS2010 = (len(Nar_G.findall(new_text2010)) / len(filt_text2010)) * 100

[267]: NarGDS2011 = (len(Nar_G.findall(new_text2011)) / len(filt_text2011)) * 100

[268]: NarGDS2012 = (len(Nar_G.findall(new_text2012)) / len(filt_text2012)) * 100

[269]: NarGDS2013 = (len(Nar_G.findall(new_text2013)) / len(filt_text2013)) * 100

[270]: NarGDS2014 = (len(Nar_G.findall(new_text2014)) / len(filt_text2014)) * 100

[271]: NarGDS2015 = (len(Nar_G.findall(new_text2015)) / len(filt_text2015)) * 100

[272]: NarGDS2016 = (len(Nar_G.findall(new_text2016)) / len(filt_text2016)) * 100

[273]: NarGDS2017 = (len(Nar_G.findall(new_text2017)) / len(filt_text2017)) * 100

[274]: NarGDS2018 = (len(Nar_G.findall(new_text2018)) / len(filt_text2018)) * 100

[275]: NarGDS2019 = (len(Nar_G.findall(new_text2019)) / len(filt_text2019)) * 100

[276]: NarGDS2020 = (len(Nar_G.findall(new_text2020)) / len(filt_text2020)) * 100

[ ]:

[277]: # Environmental Dimension

[278]: Nar_E = re.compile("(sustainable|sustainability|corporate␣
       ↪responsibility|renewable|ESG|global␣
       ↪warming|ecology|environment|environmental|paris agreement|kyoto␣
       ↪protocol|biodiversity|biofuel|biofuels|deforestation|reforestation|emission|emissions|recyc
       ↪development goals|ozone depletion|contamination|radioactive|hazardous|CO2|CO␣
       ↪2|GHG|nitrogen|asbestos|CSR|carbon|greenhouse|climate|net-zero|net zero|zero␣
       ↪waste|sectoral decarbonization|SDA|GRI|TCFD|circular economy|taxonomy|scope␣
       ↪1|scope 2|scope 3|footprint|un environment␣
       ↪programme|divestment|transition|toxic)", flags=re.I)

[ ]:

[279]: # Printing all matches

[280]: print(Nar_E.findall(new_text2020), len(Nar_E.findall(new_text2020)))

       ['energy', 'energy', 'energy', 'transition', 'transition', 'sda', 'transition',
       'environment', 'ghg', 'sustainable', 'environment', 'ghg', 'ghg', 'energy',
       'energy', 'energy', 'energy', 'air', 'climate', 'gri', 'environment',
```

```
'environment', 'environment', 'climate', 'climate', 'climate', 'air', 'sda',
'air', 'air', 'air', 'air', 'environment', 'energy', 'energy', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'air', 'air', 'climate', 'air', 'transition', 'transition', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'environment', 'energy', 'energy', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'transition', 'transition', 'air', 'air', 'air', 'air', 'air', 'air',
'transition', 'transition', 'air', 'air', 'air', 'air', 'environment', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'transition', 'transition', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'energy', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air',
'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'air', 'transition',
'sda', 'sda', 'air', 'air', 'air', 'air', 'air', 'sda', 'sda', 'air', 'air',
'taxonomy', 'taxonomy', 'taxonomy', 'taxonomy', 'taxonomy', 'air'] 227
```

[ ]:

[281]:
```
# Computing the Ratio ESG Dictionary (E) / Report Length
```

[282]:
```
NarEDS2010 = (len(Nar_E.findall(new_text2010)) / len(filt_text2010)) * 100
```

[283]:
```
NarEDS2011 = (len(Nar_E.findall(new_text2011)) / len(filt_text2011)) * 100
```

[284]:
```
NarEDS2012 = (len(Nar_E.findall(new_text2012)) / len(filt_text2012)) * 100
```

[285]:
```
NarEDS2013 = (len(Nar_E.findall(new_text2013)) / len(filt_text2013)) * 100
```

[286]:
```
NarEDS2014 = (len(Nar_E.findall(new_text2014)) / len(filt_text2014)) * 100
```

[287]:
```
NarEDS2015 = (len(Nar_E.findall(new_text2015)) / len(filt_text2015)) * 100
```

[288]:
```
NarEDS2016 = (len(Nar_E.findall(new_text2016)) / len(filt_text2016)) * 100
```

[289]:
```
NarEDS2017 = (len(Nar_E.findall(new_text2017)) / len(filt_text2017)) * 100
```

[290]:
```
NarEDS2018 = (len(Nar_E.findall(new_text2018)) / len(filt_text2018)) * 100
```

[291]:
```
NarEDS2019 = (len(Nar_E.findall(new_text2019)) / len(filt_text2019)) * 100
```

[292]:
```
NarEDS2020 = (len(Nar_E.findall(new_text2020)) / len(filt_text2020)) * 100
```

[ ]:

[293]:
```
# Social Dimension
```

[294]:
```
Nar_S = re.
↪compile("(gender|genders|transgender|female|woman|women|sex|ethnicity|ethnic|ethnicities|mi
↪rights|labor rights|equality|equal pay|pay gap|wage␣
↪gap|discrimination|discriminate|discriminated|harassment|harass|charity|donate|donation|don
↪absence|sick leave|fairtrade|fair trade)", flags=re.I)
```

[ ]:

[295]:
```
# Printing all matches
```

[296]:
```
print(Nar_S.findall(new_text2020), len(Nar_S.findall(new_text2020)))
```

```
['health', 'health', 'health', 'safety', 'health', 'health', 'workplace',
'workplace', 'workplace', 'health', 'health', 'health', 'health', 'health',
'health', 'health', 'safety', 'communities', 'safety', 'health', 'health',
'safety', 'communities', 'health', 'health', 'illness', 'health', 'health',
'health', 'health', 'health', 'health', 'health', 'health', 'health', 'health',
'health', 'health', 'health', 'health', 'safety', 'health', 'health', 'health',
'health', 'health', 'health', 'health', 'health', 'health', 'health', 'health',
'health', 'health', 'health', 'health', 'health', 'health', 'wage', 'health',
'health', 'health', 'health', 'health', 'health', 'health', 'health', 'health',
'health', 'health', 'health', 'health', 'health', 'health', 'health', 'health',
'health', 'health', 'health', 'health', 'health', 'society', 'health', 'health',
'diverse', 'health', 'inclusion', 'health', 'health', 'disability',
'disability', 'accident', 'accident', 'disability', 'disability', 'disability',
'disability', 'accident', 'accident', 'accident', 'health', 'disability',
'accident', 'accident', 'disability', 'accident', 'accident', 'disability',
'accident', 'accident', 'overtime', 'disability', 'disability', 'disability',
'disability', 'health', 'disability'] 117
```

[ ]:

[297]:
```
# Computing the Ratio ESG Dictionary (S) / Report Length
```

[298]:
```
NarSDS2010 = (len(Nar_S.findall(new_text2010)) / len(filt_text2010)) * 100
```

[299]:
```
NarSDS2011 = (len(Nar_S.findall(new_text2011)) / len(filt_text2011)) * 100
```

[300]:
```
NarSDS2012 = (len(Nar_S.findall(new_text2012)) / len(filt_text2012)) * 100
```

[301]:
```
NarSDS2013 = (len(Nar_S.findall(new_text2013)) / len(filt_text2013)) * 100
```

[302]:
```
NarSDS2014 = (len(Nar_S.findall(new_text2014)) / len(filt_text2014)) * 100
```

[303]:
```
NarSDS2015 = (len(Nar_S.findall(new_text2015)) / len(filt_text2015)) * 100
```

[304]:
```
NarSDS2016 = (len(Nar_S.findall(new_text2016)) / len(filt_text2016)) * 100
```

[305]:
```
NarSDS2017 = (len(Nar_S.findall(new_text2017)) / len(filt_text2017)) * 100
```

[306]:
```
NarSDS2018 = (len(Nar_S.findall(new_text2018)) / len(filt_text2018)) * 100
```

[307]:
```
NarSDS2019 = (len(Nar_S.findall(new_text2019)) / len(filt_text2019)) * 100
```

[308]:
```
NarSDS2020 = (len(Nar_S.findall(new_text2020)) / len(filt_text2020)) * 100
```

[ ]:
```

```

[309]:
```
# Computing the Aggregate Narrative Sustainability Disclosure
```

[ ]:
```

```

[310]:
```
NarDS2010 = ( (len(Nar_E.findall(new_text2010)) + len(Nar_S.
↪findall(new_text2010)) + len(Nar_G.findall(new_text2010))) /␣
↪len(filt_text2010) ) * 100
```

[311]:
```
NarDS2011 = ( (len(Nar_E.findall(new_text2011)) + len(Nar_S.
↪findall(new_text2011)) + len(Nar_G.findall(new_text2011))) /␣
↪len(filt_text2011) ) * 100
```

[312]:
```
NarDS2012 = ( (len(Nar_E.findall(new_text2012)) + len(Nar_S.
↪findall(new_text2012)) + len(Nar_G.findall(new_text2012))) /␣
↪len(filt_text2012) ) * 100
```

[313]:
```
NarDS2013 = ( (len(Nar_E.findall(new_text2013)) + len(Nar_S.
↪findall(new_text2013)) + len(Nar_G.findall(new_text2013))) /␣
↪len(filt_text2013) ) * 100
```

[314]:
```
NarDS2014 = ( (len(Nar_E.findall(new_text2014)) + len(Nar_S.
↪findall(new_text2014)) + len(Nar_G.findall(new_text2014))) /␣
↪len(filt_text2014) ) * 100
```

[315]:
```
NarDS2015 = ( (len(Nar_E.findall(new_text2015)) + len(Nar_S.
↪findall(new_text2015)) + len(Nar_G.findall(new_text2015))) /␣
↪len(filt_text2015) ) * 100
```

[316]:
```
NarDS2016 = ( (len(Nar_E.findall(new_text2016)) + len(Nar_S.
↪findall(new_text2016)) + len(Nar_G.findall(new_text2016))) /␣
↪len(filt_text2016) ) * 100
```

[317]:
```

```

```
NarDS2017 = ( (len(Nar_E.findall(new_text2017)) + len(Nar_S.
↪findall(new_text2017)) + len(Nar_G.findall(new_text2017))) /␣
↪len(filt_text2017) ) * 100
```

[318]:
```
NarDS2018 = ( (len(Nar_E.findall(new_text2018)) + len(Nar_S.
↪findall(new_text2018)) + len(Nar_G.findall(new_text2018))) /␣
↪len(filt_text2018) ) * 100
```

[319]:
```
NarDS2019 = ( (len(Nar_E.findall(new_text2019)) + len(Nar_S.
↪findall(new_text2019)) + len(Nar_G.findall(new_text2019))) /␣
↪len(filt_text2019) ) * 100
```

[320]:
```
NarDS2020 = ( (len(Nar_E.findall(new_text2020)) + len(Nar_S.
↪findall(new_text2020)) + len(Nar_G.findall(new_text2020))) /␣
↪len(filt_text2020) ) * 100
```

[ ]:
```

```

[321]:
```
output_env = [[E_Match2010, E_Match2011, E_Match2012, E_Match2013, E_Match2014,␣
↪E_Match2015, E_Match2016, E_Match2017, E_Match2018, E_Match2019,␣
↪E_Match2020]]
```

[322]:
```
print("="*100)
print(tabulate(output_env, headers=["E2010", "E2011", "E2012", "E2013",␣
↪"E2014", "E2015", "E2016", "E2017", "E2018", "E2019", "E2020"]))
print("-"*100)
```

```
====================================================================================
====================
  E2010    E2011    E2012    E2013    E2014    E2015    E2016    E2017    E2018
E2019    E2020
-------  -------  -------  -------  -------  -------  -------  -------  -------
-------  -------
      0        0        0        0        0        0        0        0        0
0        0
------------------------------------------------------------------------------------
--------------------
```

[323]:
```
output_gen = [[G_Match2010, G_Match2011, G_Match2012, G_Match2013, G_Match2014,␣
↪G_Match2015, G_Match2016, G_Match2017, G_Match2018, G_Match2019,␣
↪G_Match2020]]
```

[324]:
```
print("="*100)
print(tabulate(output_gen, headers=["G2010", "G2011", "G2012", "G2013",␣
↪"G2014", "G2015", "G2016", "G2017", "G2018", "G2019", "G2020"]))
print("-"*100)
```

```
====================================================================================
```

```
====================
  G2010    G2011    G2012    G2013    G2014    G2015    G2016    G2017    G2018
G2019    G2020
-------  -------  -------  -------  -------  -------  -------  -------  -------
-------  -------
      0        0        0        0        0        0        0        0        0
0        0
--------------------------------------------------------------------------------
--------------------
```

[325]:
```
output_hse = [[H_Match2010, H_Match2011, H_Match2012, H_Match2013, H_Match2014,␣
 ↪H_Match2015, H_Match2016, H_Match2017, H_Match2018, H_Match2019,␣
 ↪H_Match2020]]
```

[326]:
```
print("="*100)
print(tabulate(output_hse, headers=["H2010", "H2011", "H2012", "H2013",␣
 ↪"H2014", "H2015", "H2016", "H2017", "H2018", "H2019", "H2020"]))
print("-"*100)
```

```
====================================================================================
====================
  H2010    H2011    H2012    H2013    H2014    H2015    H2016    H2017    H2018
H2019    H2020
-------  -------  -------  -------  -------  -------  -------  -------  -------
-------  -------
      0        0        0        0        0        0        0        1        1
1        1
--------------------------------------------------------------------------------
--------------------
```

[327]:
```
output_nar = [[NarDS2010, NarDS2011, NarDS2012, NarDS2013, NarDS2014,␣
 ↪NarDS2015, NarDS2016, NarDS2017, NarDS2018, NarDS2019, NarDS2020]]
```

[328]:
```
print("="*100)
print(tabulate(output_nar, headers=["2010", "2011", "2012", "2013", "2014",␣
 ↪"2015", "2016", "2017", "2018", "2019", "2020"]))
print("-"*100)
```

```
====================================================================================
====================
   2010     2011     2012     2013     2014     2015     2016     2017     2018
2019     2020
-------  -------  -------  -------  -------  -------  -------  -------  -------
-------  -------
2.05654  2.07818  2.24422  2.27032  2.60184  2.44208  2.23727  3.26695  3.11057
3.15597  3.32111
--------------------------------------------------------------------------------
--------------------
```

[329]:
```
output_oi = [[OutIn2010, OutIn2011, OutIn2012, OutIn2013, OutIn2014, OutIn2015,␣
↪OutIn2016, OutIn2017, OutIn2018, OutIn2019, OutIn2020]]
```

[330]:
```
print("="*100)
print(tabulate(output_oi, headers=["2010", "2011", "2012", "2013", "2014",␣
↪"2015", "2016", "2017", "2018", "2019", "2020"]))
print("-"*100)
```

```
====================================================================================
====================
  2010    2011    2012    2013    2014    2015    2016    2017    2018    2019
2020
------  ------  ------  ------  ------  ------  ------  ------  ------  ------
------
     0       0       0       0       0       0       0       0       0       1
1
------------------------------------------------------------------------------------
--------------------
```

[ ]:

[ ]:

Difference in Difference Analysis-wControls

June 23, 2022

[1]:
```
## Difference-in-Difference Analysis and Regression
```

[ ]:
```

```

[2]:
```
# Importing neccessary packages and toolkits
```

[3]:
```
import pandas as pd
```

[4]:
```
import statsmodels.formula.api as smf
```

[5]:
```
import numpy as np
```

[6]:
```
import statistics
```

[7]:
```
import matplotlib.pyplot as plt
```

[8]:
```
from matplotlib.collections import LineCollection
```

[9]:
```
from scipy import interpolate
```

[ ]:
```

```

[10]:
```
# Importing the data
```

[11]:
```
df =pd.read_excel(r'/YOUR PATH/Doc.xlsx')
```

[12]:
```
#print(df) # Overview of the data
```

[ ]:
```

```

[13]:
```
# Computing mean values
```

[14]:
```
mean_both = df.EnvPerc.mean(), df.GenPerc.mean(), df.HSEPerc.mean(), df.NarDS.
↪mean(), df.OutInPerc.mean()
```

[ ]:
```

```

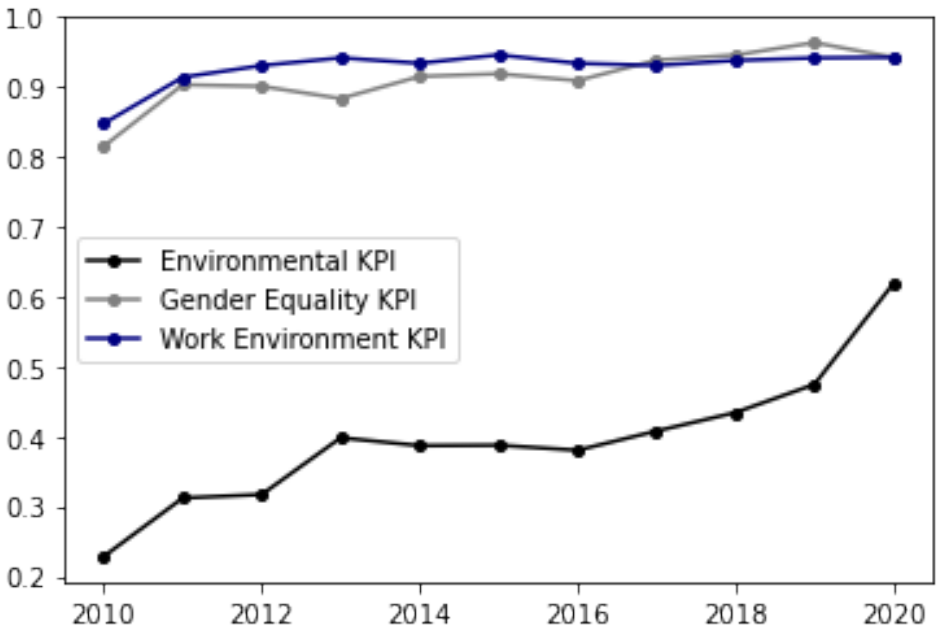
[15]:
```
# Creating a data frame with only the treatment group (domiciled = 1)
# and illustrating the measures as graphs

df_treat = df.query('Domiciled==1')
#df_treat
```

[ ]:
```
```

[16]:
```
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020]
```

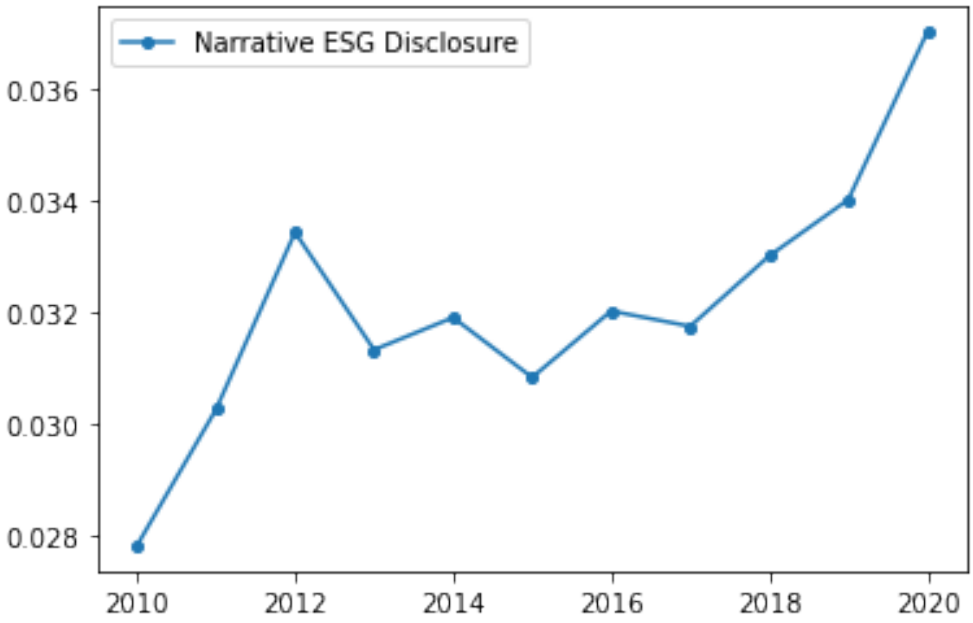
[ ]:
```
```

[17]:
```
plt.plot(year, df_treat.EnvPerc, label = 'Environmental KPI', linestyle='-',␣
 ↪color='black', marker='o', markersize=4)
plt.plot(year, df_treat.GenPerc, label = 'Gender Equality KPI', linestyle='-',␣
 ↪color='grey', marker='o', markersize=4)
plt.plot(year, df_treat.HSEPerc, label = 'Work Environment KPI', linestyle='-',␣
 ↪color='navy', marker='o', markersize=4)
plt.legend()
plt.savefig('/Users/sindreaske/Documents/Master/MASTER/NewFigures/
 ↪KPIsTreatmentNew.png')
```



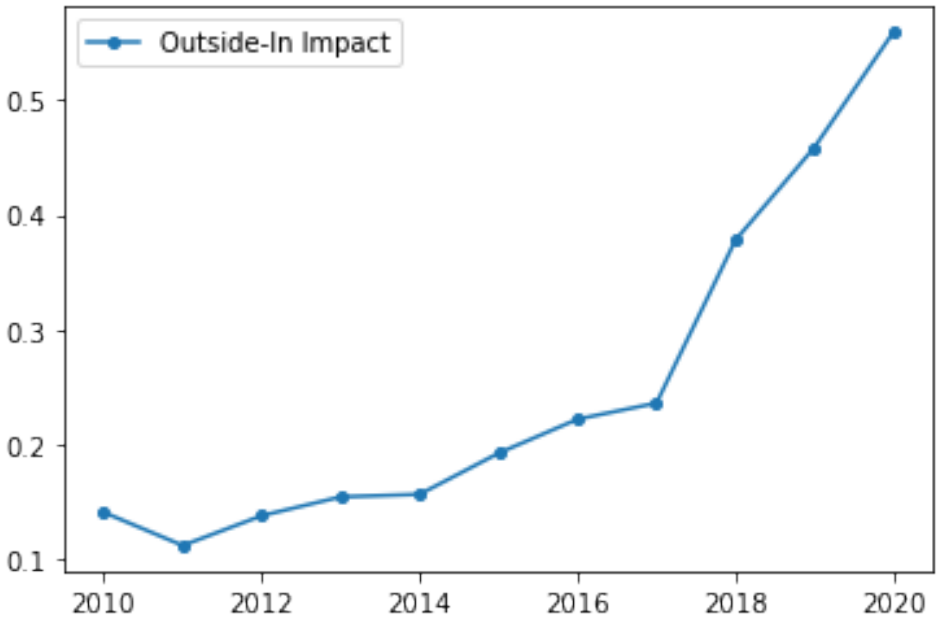
[18]:
```
plt.plot(year, df_treat.NarDS, label = 'Narrative ESG Disclosure',␣
 ↪linestyle='-', marker='o', markersize=4)
plt.legend()
```

[18]: <matplotlib.legend.Legend at 0x7f807857cc10>



[19]:
```
plt.plot(year, df_treat.OutInPerc, label = 'Outside-In Impact', linestyle='-',␣
↪marker='o', markersize=4)
plt.legend()
```

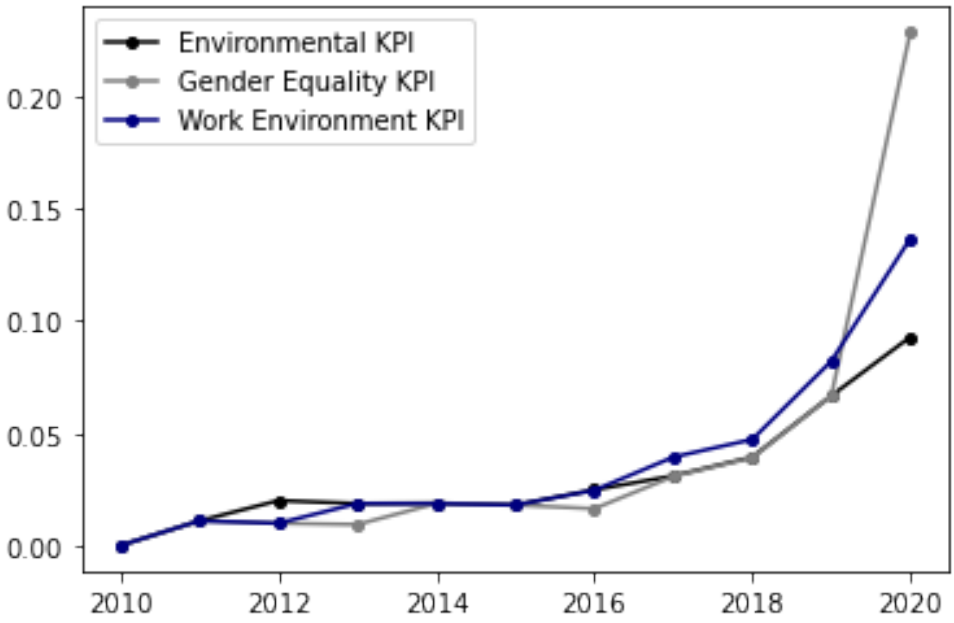
[19]: <matplotlib.legend.Legend at 0x7f807859cd90>



[ ]:

[20]:
```
# Creating a data frame with only the control group (domiciled = 0)
# and illustrating the measures as graphs

df_control = df.query('Domiciled==0')
#df_control
```

[21]:
```
plt.plot(year, df_control.EnvPerc, label = 'Environmental KPI', linestyle='-',␣
 ↪color='black', marker='o', markersize=4)
plt.plot(year, df_control.GenPerc, label = 'Gender Equality KPI',␣
 ↪linestyle='-', color='grey', marker='o', markersize=4)
plt.plot(year, df_control.HSEPerc, label = 'Work Environment KPI',␣
 ↪linestyle='-', color='navy', marker='o', markersize=4)
plt.legend()
plt.savefig('/Users/sindreaske/Documents/Master/MASTER/NewFigures/
 ↪KPIsControlNew.png')
```



[22]:
```
#plt.plot(year, df_control.NarDS, label = 'Narrative ESG Disclosure',␣
 ↪linestyle='-', marker='o', markersize=4)
#plt.legend()
```

[23]:
```
#plt.plot(year, df_control.OutInPerc, label = 'Outside-In Impact',␣
↪linestyle='-', marker='o', markersize=4)
#plt.legend()
```

[ ]:
```
```

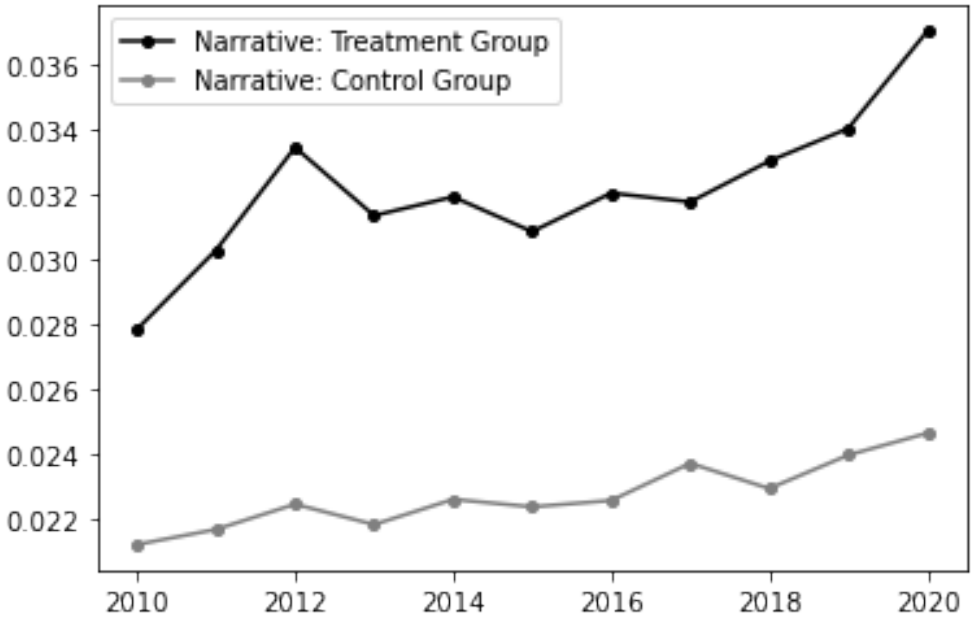
[24]:
```
# Crateating a data frame with both groups to compare the disclosure measures

df_comparison = df.query('Domiciled==1 or Domiciled==0')
```

[25]:
```
df_comparison = df_comparison[['Year', 'Domiciled', 'EnvPerc', 'GenPerc',␣
↪'HSEPerc', 'NarDS', 'OutInPerc', 'LNMVE', 'MTB', 'Leverage', 'ROA']]

#df_comparison.head()
```

[ ]:
```
```

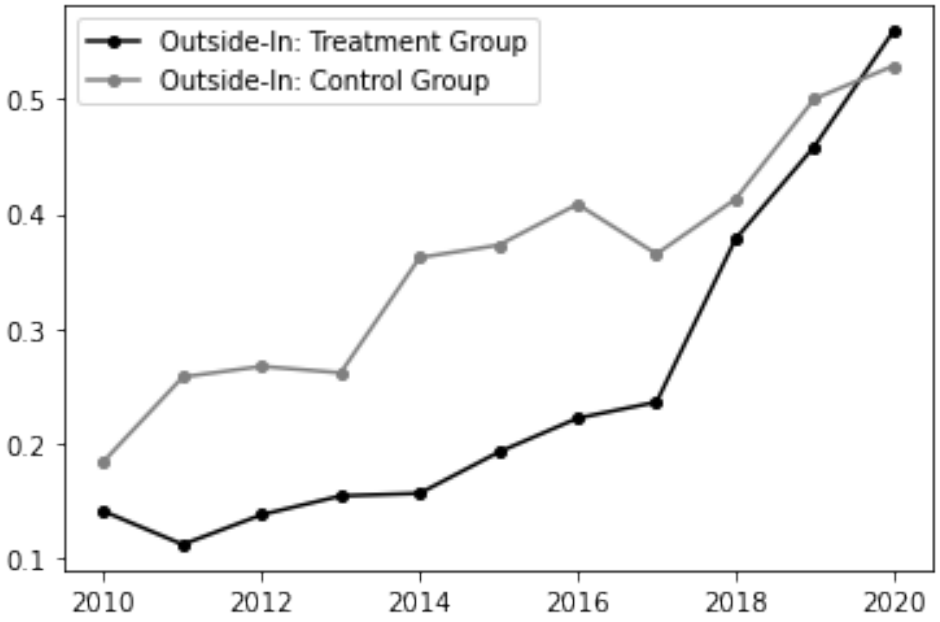
[26]:
```
# Narrative sustainability disclosure
```

[27]:
```
plt.plot(year, df_treat.NarDS, label = 'Narrative: Treatment Group',␣
↪linestyle='-', color='black', marker='o', markersize=4)
plt.plot(year, df_control.NarDS, label = 'Narrative: Control Group',␣
↪linestyle='-', color='grey', marker='o', markersize=4)
plt.legend()
plt.savefig('/Users/sindreaske/Documents/Master/MASTER/NewFigures/NarrativeNew.
↪png')
```



[28]:
```
# Outside-in measure
```

[29]:
```
plt.plot(year, df_treat.OutInPerc, label = 'Outside-In: Treatment Group',␣
 ↪linestyle='-', color='black', marker='o', markersize=4)
plt.plot(year, df_control.OutInPerc, label = 'Outside-In: Control Group',␣
 ↪linestyle='-', color='grey', marker='o', markersize=4)
plt.legend()
plt.savefig('/Users/sindreaske/Documents/Master/MASTER/NewFigures/OutsideInNew.
 ↪png')
```



[ ]:

[30]:
```
# Shaping the dataset in order to run the difference-in-difference

df_treat_and_control = df_comparison.set_index(['Year', 'Domiciled']).
 ↪unstack('Domiciled')
#df_treat_and_control
```

[ ]:

[31]:
```
# Implementing a dummy variable to distinguish between pre- and post-periods
# The dummy (copay) is changed from 2013 to 2018 to examine the effect of the
# removal of the transition rule.
```

[32]:
```
df_treat_and_control['copay'] = 0
```

[33]:
```
df_treat_and_control['copay'] = df_treat_and_control.copay.
↪where(df_treat_and_control.index<2018, 1)
```

[ ]:
```

```

[34]:
```
# Grouping by disclosure measure and computing means pre-and post treatment
```

[35]:
```
DID = df_treat_and_control.groupby('copay').mean()
```

[36]:
```
DID.columns = ['Control_Env', 'Treat_Env', 'Control_Gen', 'Treat_Gen',␣
↪'Control_HSE', 'Treat_HSE', 'Control_NarDS', 'Treat_NarDS', 'Control_OutIn',␣
↪'Treat_OutIn', 'Control_MVE', 'Treat_MVE', 'Control_MTB', 'Treat_MTB',␣
↪'Control_Lev', 'Treat_Lev', 'Control_ROA', 'Treat_ROA']
DID
```

[36]:
```
       Control_Env  Treat_Env  Control_Gen  Treat_Gen  Control_HSE  Treat_HSE  \
copay
0         0.017796   0.352127     0.014264   0.898062     0.017531   0.922351
1         0.066298   0.509089     0.111434   0.950562     0.088675   0.940679

       Control_NarDS  Treat_NarDS  Control_OutIn  Treat_OutIn  Control_MVE  \
copay
0           0.022278     0.031165       0.309988     0.169305    19.804633
1           0.023823     0.034672       0.480423     0.465141    19.723426

       Treat_MVE  Control_MTB  Treat_MTB  Control_Lev  Treat_Lev  Control_ROA  \
copay
0      18.359740     2.625937   2.245446     0.322002   0.303993    -0.023216
1      18.012151     2.714963   2.972700     0.307288   0.312973    -0.055370

       Treat_ROA
copay
0      -0.009507
1      -0.026262
```

[37]:
```
# Univariate Difference-in-Difference

# Disclosure on impact on external environment (EnvKPI)

#DID['Treat_Env'] - DID['Control_Env']
```

[38]:
```
DID_Env = (DID['Treat_Env'] - DID['Control_Env']).diff()
```

[39]:
```
 # Disclosure on gender equality

#DID['Treat_Gen'] - DID['Control_Gen']
```

[40]:
```
DID_Gen = (DID['Treat_Gen'] - DID['Control_Gen']).diff()
```

[41]:
```
 # Disclosure on work environment

#DID['Treat_HSE'] - DID['Control_HSE']
```

[42]:
```
DID_HSE = (DID['Treat_HSE'] - DID['Control_HSE']).diff()
```

[43]:
```
 # Narrative ESG Disclosure

#DID['Treat_NarDS'] - DID['Control_NarDS']
```

[44]:
```
DID_NarDS = (DID['Treat_NarDS'] - DID['Control_NarDS']).diff()
```

[45]:
```
 # Addressing outside in impact of the external environment

#DID['Treat_OutIn'] - DID['Control_OutIn']
```

[46]:
```
DID_OutIn = (DID['Treat_OutIn'] - DID['Control_OutIn']).diff()
```

[ ]:
```

```

[47]:
```
# Difference-in-Difference Regressions
```

[ ]:
```

```

[48]:
```
# Implementing the dummy variable. The dummy variable is changed between 2013␣
 ↪and 2018.

df_comparison['copay'] = 0
df_comparison['copay'] = df_comparison['copay'].where(df_comparison.Year<2018,␣
 ↪1)
```

[ ]:
```

```

[49]:
```
# Creating a dummy variable that takes on the value 1 for the treatment group
```

[50]:
```
df_comparison['Treatment'] = np.where(df_comparison.Domiciled==1, 1, 0)
```

[ ]:
```

```

[51]:
```
# Regression

# Outcome = treat dum + post treatment + treated * post treatment + year fixed

# Interaction term (treated * post treatment) where the firms are in the␣
 ↪treatment group
# and after treatment.
```

[52]:
```
# Regression 1: Disclosure on impact on external environment.

model_env = 'EnvPerc ~ Treatment + copay + copay * Treatment + LNMVE + MTB +␣
↪Leverage + ROA'
```

[53]:
```
mod_env = smf.ols(formula=model_env, data=df_comparison)
res_env = mod_env.fit()
#print(res_env.summary())
```

[ ]:
```

```

[54]:
```
# Regression 2: Disclosure on Gender Equality
```

[55]:
```
model_gen = 'GenPerc ~ Treatment + copay + copay * Treatment + LNMVE + MTB +␣
↪Leverage + ROA'
```

[56]:
```
mod_gen = smf.ols(formula=model_gen, data=df_comparison)
res_gen = mod_gen.fit()
#print(res_gen.summary())
```

[ ]:
```

```

[57]:
```
# Regression 3: Disclosure on Work Environment
```

[58]:
```
model_hse = 'HSEPerc ~ Treatment + copay + copay * Treatment + LNMVE + MTB +␣
↪Leverage + ROA'
```

[59]:
```
mod_hse = smf.ols(formula=model_hse, data=df_comparison)
res_hse = mod_hse.fit()
#print(res_hse.summary())
```

[ ]:
```

```

[60]:
```
# Regression 4: Narrative Sustainability Disclosure
```

[61]:
```
model_NarDS = 'NarDS ~ Treatment + copay + copay * Treatment + LNMVE + MTB +␣
↪Leverage + ROA'
```

[62]:
```
mod_NarDS = smf.ols(formula=model_NarDS, data=df_comparison)
res_NarDS = mod_NarDS.fit()
#print(res_NarDS.summary())
```

[ ]:
```

```

[63]:
```
# Regression 5: Outside-In Impact
```

[64]:
```
model_OutIn = 'OutInPerc ~ Treatment + copay + copay * Treatment + LNMVE + MTB␣
↪+ Leverage + ROA'
```

[65]:
```
mod_OutIn = smf.ols(formula=model_OutIn, data=df_comparison)
res_OutIn = mod_OutIn.fit()
#print(res_OutIn.summary())
```

[ ]:
```

```