

Vendor Risk Management Program: Management Reporting

This program uses a sample excel file to import data and perform some analysis and visualization of the program data. The goal of this file is to generate data for use in vendor risk management reports at the program level, risk assessment level and vendor level (aggregation of vendor contracts), and perform risk scoring of the vendors

Import Libraries and Dependencies

```
In [109]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pandas_profiling import ProfileReport
import seaborn as sns
import plotly.pyplot as plt
import chart_studio
import reators as sns
import matplotlib.pyplot as plt
import numpy as np
from pandas_profiling import ProfileReport
import random
```

Read Excel Source Files

```
In [49]: # Read VRM Program Data
# The program data provides a high level over of the vrm program; it shows vendor and contract specific
information, length of service, geographic information, contract value, service risk score,
contract compliance with respect to RPIs, open audit, regulatory and assessment issues, etc...
program_data = pd.read_excel('vrm_program_reporting.xlsx')
program_data.head()
```

	Vendor	Service Type	Contract ID	Service Inception	Region	Country	Contract Value	Service Risk Score	Service Risk Rating	Alternate Vendor	Reassessment Date	Reassessment Completion	Reas
0	Vend-87	Software Packages	Con-101	2019-11-01	North America	USA	32139	12.0	High	No	2020-10-31	2020-11-30	
1	Vend-61	Insurance	Con-102	2019-05-13	Europe	Germany	90685	10.5	High	No	2020-05-12	2020-06-11	
2	Vend-98	Hardware	Con-103	2019-04-06	South America	Mexico	108545	8.0	Medium	Not Required	2021-04-04	2021-05-04	Past
3	Vend-54	Mail	Con-104	2020-07-08	Asia	India	23560	8.0	Medium	Not Required	2022-07-08	2022-08-07	Not
4	Vend-70	Technical Consulting	Con-105	2019-03-13	North America	USA	10898	9.0	Medium	Not Required	2021-03-12	2021-04-11	

```
In [50]: # Read Risk Assessment Data worksheet within vrm_program_reporting.xlsx file
# This data shows the number of responses across the risk domains, total responses % of risk assessment
completed and the status of the risk assessment
# The status of risk assessment include, past due 0 - 15 days, in progress - due in 15 days, past due over 30 days, in progress between 15 and 30 days, etc...
risk_assmt_data = pd.read_excel('vrm_program_reporting.xlsx', 'risk_assmt_data')
risk_assmt_data.head()
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	To Respons
0	Vend-98	Hardware	Con-103	0	0	3	10	3	17	27	26	4
1	Vend-76	Stationary	Con-107	0	0	0	9	5	26	40	16	5
2	Vend-50	Software Packages	Con-123	3	6	2	1	53	45	23	7	1
3	Vend-91	Administrative Support	Con-131	1	6	7	1	22	14	27	6	
4	Vend-64	Stationary	Con-138	0	8	8	8	7	55	15	5	2

```
In [51]: risk_assmt_data.tail()
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	T Respon
6	Vend-79	Technical Consulting	Con-158	6	10	8	7	11	30	11	4	
7	Vend-52	Mail	Con-173	3	5	3	1	34	43	9	4	
8	Vend-86	Software Packages	Con-182	5	1	5	4	47	37	13	10	
9	Vend-75	Insurance	Con-198	5	11	7	7	57	27	34	10	
10	Total Questions	Total Questions	Total Questions	10	20	15	10	75	50	40	10	

```
In [52]: # Read Risk Scoring Data
# Shows the derivation of service risk score at the contract/service level.
# This methodology assigns 50% of the risk to strategic, reputational, financial, physical security and
information; it assigns the remaining 50% to legal/regulatory, information security and
information technology to derive the service/contract risk score
risk_scoring = pd.read_excel('vrm_program_reporting.xlsx', 'risk_scoring_data')
risk_scoring.head()
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	Service Risk Score	Se R
0	Vend-87	Software Packages	Con-101	4	3	2	2	4	2	4	3	12.0	
1	Vend-61	Insurance	Con-102	2	4	2	2	4	3	1	3	10.5	
2	Vend-98	Hardware	Con-103	1	1	2	2	3	2	1	4	8.0 Me	
3	Vend-54	Mail	Con-104	1	3	2	2	1	1	3	3	8.0 Me	
4	Vend-70	Technical Consulting	Con-105	1	4	1	2	3	3	3	1	9.0 Me	

```
In [53]: # Calibration of Service Risk Data
# This shows the ranges of the Service Risk Score and how these align to the ratings: Low, Medium, High
and Critical
# It also shows the Best Case (Lowest Risk) and Worst Case (Highest Risk) for contracts
risk_scoring.tail(2)
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	Service Risk Score	Se R
100	Best Case	Best Case	Lowest Risk	1	1	1	1	1	1	1	1	4.0	Ap
101	Worst Case	Worst Case	Highest Risk	4	4	4	4	4	4	4	4	16.0	Ap

```
In [54]: risk_calibration = pd.read_excel('vrm_program_reporting.xlsx', 'calibration')
```

```
Out [54]: risk_calibration
```

	Scale	Inequality	numerator	percentage	Rating	Explanation
0	(0,7]	7 <= Score < 7	7.0	0.4375	Low	Less than 7 (or 43.75%)
1	(7,10]	7 <= Score < 10	10.0	0.6250	Medium	Between 7 and 10 (0.4375% and 62.5%)
2	(10,14]	10 <= Score < 14	14.0	0.8750	High	Between 10 and 14 (or 62.5% and 87.5%)
3	>=14	Score >= 14	NaN	NaN	Critical	Greater than 14 (or 87.5%)

```
In [55]: risk_scoring.tail()
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	Service Risk Score	Se R
97	Vend-75	Insurance	Con-198	4	2	3	4	2	3	2	2	11.0	
98	Vend-84	Administrative Support	Con-199	3	2	4	1	2	3	1	2	9.0	
99	Vend-64	Engineering	Con-200	4	3	4	1	1	3	4	2	11.0	
100	Best Case	Best Case	Lowest Risk	1	1	1	1	1	1	1	1	4.0	
101	Worst Case	Worst Case	Highest Risk	4	4	4	4	4	4	4	4	16.0	

```
In [56]: # Drop the last two rows of risk_scoring dataframe
risk_scoring.drop(risk_scoring.tail(2).index, inplace = True)
risk_scoring.head()
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	Service Risk Score	Se R
0	Vend-87	Software Packages	Con-101	4	3	2	2	4	2	4	3	12.0	
1	Vend-61	Insurance	Con-102	2	4	2	2	4	3	1	3	10.5	
2	Vend-98	Hardware	Con-103	1	1	2	2	3	2	1	4	8.0 Me	
3	Vend-54	Mail	Con-104	1	3	2	2	1	1	3	3	8.0 Me	
4	Vend-70	Technical Consulting	Con-105	1	4	1	2	3	3	3	1	9.0 Me	

```
In [57]: risk_scoring
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	Service Risk Score	Se R
0	Vend-87	Software Packages	Con-101	4	3	2	2	4	2	4	3	12.0	
1	Vend-61	Insurance	Con-102	2	4	2	2	4	3	1	3	10.5	
2	Vend-98	Hardware	Con-103	1	1	2	2	3	2	1	4	8.0	
3	Vend-54	Mail	Con-104	1	3	2	2	1	1	3	3	8.0	
4	Vend-70	Technical Consulting	Con-105	1	4	1	2	3	3	3	1	9.0	

```
Out [56]: risk_scoring
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	Service Risk Score	Se R
0	Vend-87	Software Packages	Con-101	4	3	2	2	4	2	4	3	12.0	
1	Vend-61	Insurance	Con-102	2	4	2	2	4	3	1	3	10.5	
2	Vend-98	Hardware	Con-103	1	1	2	2	3	2	1	4	8.0	
3	Vend-54	Mail	Con-104	1	3	2	2	1	1	3	3	8.0	
4	Vend-70	Technical Consulting	Con-105	1	4	1	2	3	3	3	1	9.0	

```
Out [57]: risk_scoring
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	Service Risk Score	Se R
0	Vend-87	Software Packages	Con-101	4	3	2	2	4	2	4	3	12.0	
1	Vend-61	Insurance	Con-102	2	4	2	2	4	3	1	3	10.5	
2	Vend-98	Hardware	Con-103	1	1	2	2	3	2	1	4	8.0	
3	Vend-54	Mail	Con-104	1	3	2	2	1	1	3	3	8.0	
4	Vend-70	Technical Consulting	Con-105	1	4	1	2	3	3	3	1	9.0	

```
Out [57]: risk_scoring
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	Service Risk Score	Se R
95	Vend-84	Administrative Support	Con-196	2	1	2	3	4	4	4	4	12.0	
96	Vend-79	Mail	Con-197	2	4	1	1	4	3	2	2	9.5	
97	Vend-75	Insurance	Con-198	4	2	3	4	2	3	2	2	11.0	
98	Vend-84	Administrative Support	Con-199	3	2	4	1	2	3	1	2	9.0	
99	Vend-64	Engineering	Con-200	4	3	4	1	1	3	4	2	11.0	

```
Out [58]: risk_scoring.tail()
```

	Vendor	Service Type	Contract ID	Strategic Risk	Legal and Regulatory Compliance	Financial Risk	Reputational	Information Security	Information Technology	Physical Security	Insurance Coverage	Service Risk Score	Se R
95	Vend-84	Administrative Support	Con-196	2	1	2	3	4	4	4	4	12.0	
96	Vend-79	Mail	Con-197	2	4	1	1	4	3	2	2	9.5	
97	Vend-75	Insurance	Con-198	4	2	3	4	2	3	2	2	11.0	
98	Vend-84	Administrative Support	Con-199	3	2	4	1	2	3	1	2	9.0	
99	Vend-64	Engineering	Con-200	4	3	4	1	1	3	4	2	11.0	

Data Cleaning and Reprocessing

```
In [59]: # Datatypes
program_data.dtypes
```

```
Out [59]: Vendor                object
Service Type                object
Contract ID                object
Service Inception          datetime64[ns]
Region                    object
Country                   object
Contract Value             int64
Service Risk Score         float64
Service Risk Rating        object
Alternate Vendor           object
Reassessment Date          datetime64[ns]
Reassessment Completion    datetime64[ns]
Reassessment Status        object
Days Until Reassessment    object
RPIs Measured              object
Contract Compliance Risk   object
Open Regulatory Findings   int64
Open Audit Findings        int64
Unresolved Assessment Findings int64
dtype: object
```

```
In [60]: risk_assmt_data.dtypes
```

```
Out [60]: Vendor                object
Service Type                object
Contract ID                object
Strategic Risk             int64
Legal and Regulatory Compliance int64
Financial Risk             int64
Reputational              int64
Information Security       int64
Information Technology     int64
Physical Security         int64
Insurance Coverage        int64
Total Responses           float64
Completion %              float64
Reassessment Status        object
dtype: object
```

```
In [61]: risk_scoring.dtypes
```

```
Out [61]: Vendor                object
Service Type                object
Contract ID                object
Strategic Risk             int64
Legal and Regulatory Compliance int64
Financial Risk             int64
Reputational              int64
Information Security       int64
Information Technology     int64
Physical Security         int64
Insurance Coverage        int64
Service Risk Score         float64
Service Risk Rating        object
Normalized Score          float64
Reassessment Status        object
dtype: object
```

```
In [62]: # Check for missing data
program_data.isna().sum()
```

```
Out [62]: Vendor                0
Service Type                0
Contract ID                0
Service Inception          0
Region                    0
Country                   0
Contract Value             0
Service Risk Score         0
Service Risk Rating        0
Alternate Vendor           0
Reassessment Date          0
Reassessment Completion    0
Reassessment Status        0
Days Until Reassessment    0
RPIs Measured              0
Contract Compliance Risk   0
Open Regulatory Findings   0
Open Audit Findings        0
Unresolved Assessment Findings 0
dtype: int64
```

```
In [63]: # Check for duplicates
program_data.duplicated().sum()
```

```
Out [63]: 0
```

Data Visualization of VRM Program Data

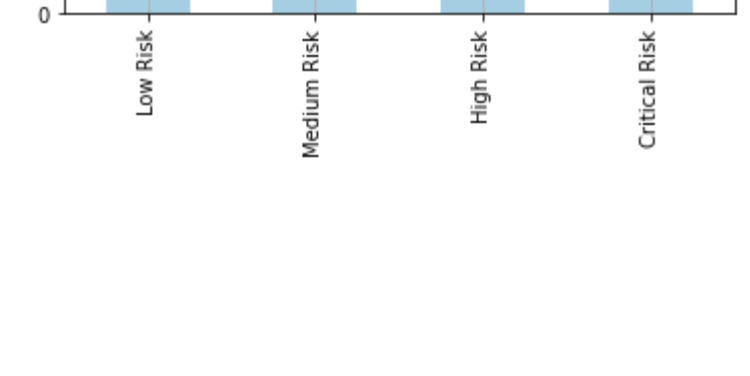
```
In [64]: # Descriptive Statistics Data
program_data.describe()
```

```
Out [64]:
```

	Contract Value	Service Risk Score	Open Regulatory Findings	Open Audit Findings	Unresolved Assessment Findings
count	100.000000	100.000000	100.000000	100.000000	100.000000
mean	123381.020000	9.920000	2.420000	2.650000	2.070000
std	67822.374588	1.70815	1.810073	1.783057	1.584362
min	10898.000000	6.000000	0.000000	0.000000	0.000000
25%	67565.000000	8.87500	1.000000	1.000000	0.750000
50%	126764.500000	10.00000	2.000000	3.000000	2.000000
75%	171904.750000	11.00000	4.000000	4.000000	3.000000
max	244779.000000	15.00000	6.000000	5.000000	5.000000

```
In [65]: # Distribution of Service Risk Score
# The Average Service Risk Score is 9.92 (from the summary statistics table); the bell curve approximates to 10
sns.distplot(program_data['Service Risk Score'])
plt.axvline(x=np.mean(program_data['Service Risk Score']), c='red', ls='--', label='mean')
plt.axvline(x=np.percentile(program_data['Service Risk Score'], 25), c='green', ls='--', label='25-75th percentile')
plt.axvline(x=np.percentile(program_data['Service Risk Score'], 75), c='green', ls='--')
plt.legend()
```

```
Out [65]: <matplotlib.legend.Legend at 0x2ae3e4254c8>
```



```
In [66]: # Distribution of contracts across service risk ratings: Low, Medium, High and Low
sns.countplot(program_data['Service Risk Rating'])
```

```
Out [66]: <AxesSubplot: xlabel='Service Risk Rating', ylabel='count'>
```



```
In [67]: # Distribution of Contract Value
# The average contract value is $123,381
sns.distplot(program_data['Contract Value'], c='red', ls='--', label='mean')
plt.axvline(x=np.mean(program_data['Contract Value']), c='red', ls='--', label='mean')
plt.axvline(x=np.percentile(program_data['Contract Value'], 25), c='green', ls='--', label='25-75th percentile')
plt.axvline(x=np.percentile(program_data['Contract Value'], 75), c='green', ls='--')
plt.legend()
```

```
Out [67]: <matplotlib.legend.Legend at 0x2ae3e4f1808>
```



```
In [68]: # Display all Service Types in the VRM Program file
uniqueServiceType = program_data['Service Type'].unique()
```

```
Out [68]: array(['Software Packages', 'Insurance', 'Hardware', 'Mail',
       'Technical Consulting', 'Engineering', 'Stationary',
       'Professional Services', 'Legal', 'Information Technology',
       'Travel', 'Administrative Support', 'Printing', 'Human Resources'],
      dtype=object)
```

```
In [69]: # Determine the concentration of contracts across the service types
insurance = 0
hardware = 0
mail = 0
technical_consulting = 0
engineering = 0
stationary = 0
prof_services = 0
legal = 0
info_tech = 0
travel = 0
admin_support = 0
printing = 0
hr = 0

for service_type in program_data['Service Type']:
    if service_type == 'Software Packages':
        software += 1
    elif service_type == 'Insurance':
        insurance += 1
    elif service_type == 'Hardware':
        hardware += 1
    elif service_type == 'Mail':
        mail += 1
    elif service_type == 'Technical Consulting':
        technical_consulting += 1
    elif service_type == 'Engineering':
        engineering += 1
    elif service_type == 'Stationary':
        stationary += 1
    elif service_type == 'Professional Services':
        prof_services += 1
    elif service_type == 'Legal':
        legal += 1
    elif service_type == 'Information Technology':
        info_tech += 1
    elif service_type == 'Travel':
        travel += 1
    elif service_type == 'Administrative Support':
        admin_support += 1
    elif service_type == 'Printing':
        printing += 1
    elif service_type == 'Human Resources':
        hr += 1
total_services = software + insurance + hardware + mail + technical_consulting + engineering + stationary + prof_services + legal + info_tech + travel + admin_support + printing + hr
total_services
```

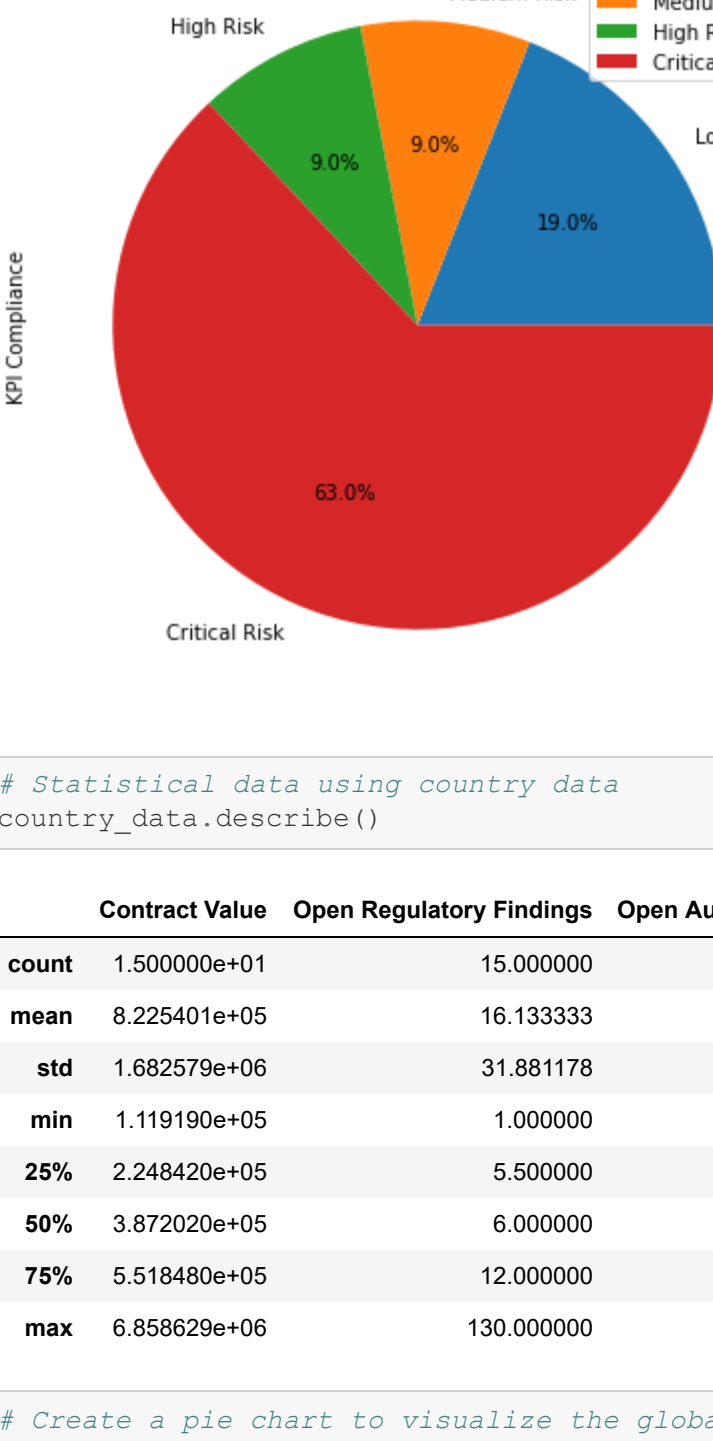
```
Out [69]: 100
Out [70]: engineering
Out [70]: 5
```

```
In [71]: # Visualize all data in the program data file
program_data.hist(figsize = (10, 10))
```

```
Out [71]: array([[<AxesSubplot: titles='center': '
```


In [110]: # Create a pie chart to visualize the KPI Compliance for all contracts in the VRM program as Low, Medium, High, and Critical compliance_dataframe.transpose(['KPI Compliance']).plot.pie(legend = True, autopct = '%.1f%%', figsize = (14, 7))

Out[110]: <AxesSubplot:ylabel='KPI Compliance'>



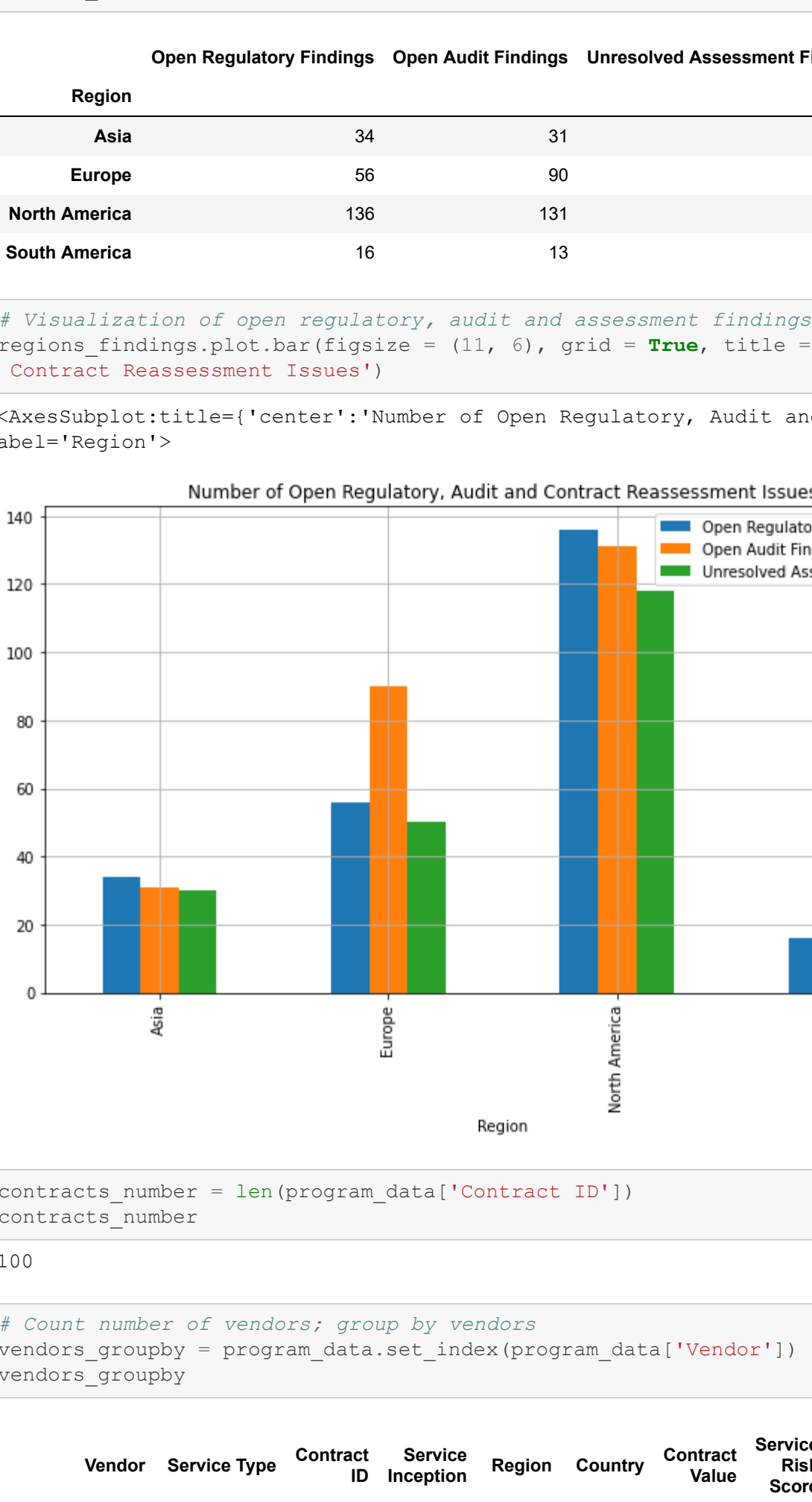
In [85]: # Statistical data using country data country_data.describe()

Out[85]:

	Contract Value	Open Regulatory Findings	Open Audit Findings	Unresolved Assessment Findings
count	1.500000e+01	15.000000	15.000000	15.000000
mean	8.225401e+05	16.133333	17.666667	13.800000
std	1.682579e+06	31.881178	30.651663	26.630273
min	1.119190e+05	1.000000	3.000000	2.000000
25%	2.248420e+05	5.500000	6.000000	4.000000
50%	3.872020e+05	6.000000	9.000000	7.000000
75%	5.518480e+05	12.000000	10.500000	10.500000
max	6.558629e+06	130.000000	126.000000	109.000000

In [86]: # Create a pie chart to visualize the global distribution of contract value in percentages regions_findings = regions_groupby(['Open Regulatory Findings', 'Open Audit Findings', 'Unresolved Assessment Findings']).sum()

Out[86]: <AxesSubplot:ylabel='Contract Value'>



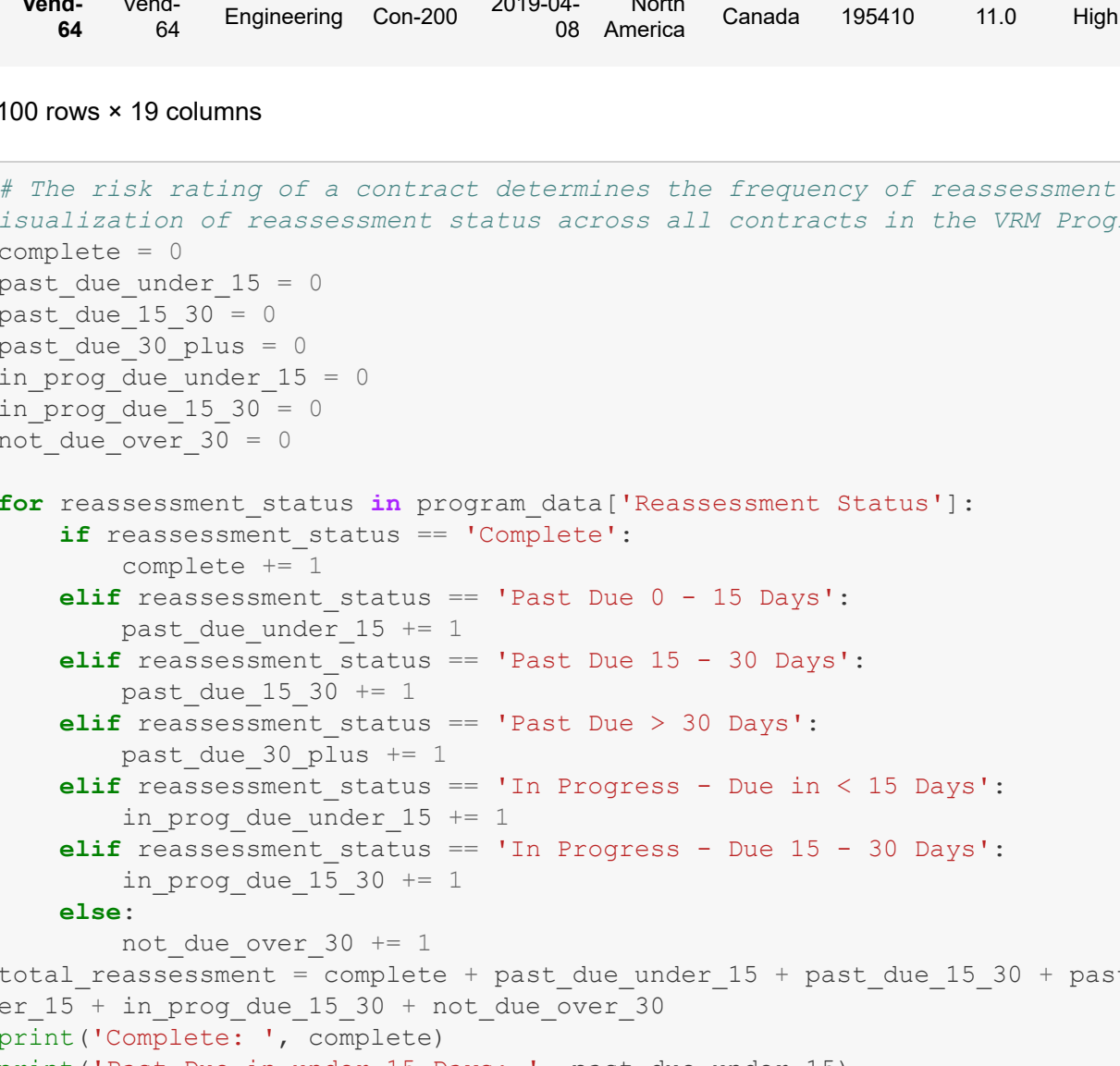
In [87]: # Open Regulatory, Audit and Assessment Findings by Region regions_groupby = regions_groupby(['Open Regulatory Findings', 'Open Audit Findings', 'Unresolved Assessment Findings']).sum()

Out[87]:

	Open Regulatory Findings	Open Audit Findings	Unresolved Assessment Findings
Region			
Asia	34	31	30
Europe	56	90	50
North America	136	131	118
South America	16	13	9

In [88]: # Visualization of open regulatory, audit and assessment findings across all regions regions_findings.plot.bar(figsize = (11, 6), grid = True, title = 'Number of Open Regulatory, Audit and Contract Reassessment Issues')

Out[88]: <AxesSubplot:title='center': 'Number of Open Regulatory, Audit and Contract Reassessment Issues', xlabel='Region'>



In [89]: contracts_number = len(program_data['Contract ID']) contracts_number

Out[89]: 100

In [90]: # Count number of vendors; group by vendors vendors_groupby = program_data.set_index(program_data['Vendor'])

Out[90]:

	Vendor	Service Type	Contract ID	Service Inception	Region	Country	Contract Value	Service Risk Score	Service Risk Rating	Alternate Vendor	Reassessment Date	Reassessment Complete
Vend-87	Vend-87	Software Packages	Con-101	2019-11-01	North America	USA	32139	12.0	High	No	2020-10-31	2020-11-30
Vend-61	Vend-61	Insurance	Con-102	2019-05-13	Europe	Germany	90695	10.5	High	No	2020-05-12	2020-06-1
Vend-98	Vend-98	Hardware	Con-103	2019-04-05	South America	Mexico	108545	8.0	Medium	Not Required	2021-04-04	2021-05-0
Vend-54	Vend-54	Mail	Con-104	2020-07-08	Asia	India	23560	8.0	Medium	Not Required	2021-03-12	2021-04-0
Vend-70	Vend-70	Technical Consulting	Con-105	2019-03-13	North America	USA	10898	9.0	Medium	Not Required	2021-03-12	2021-04-1
Vend-89	Vend-89	Administrative Support	Con-196	2021-01-20	North America	USA	241033	12.0	High	Yes	2022-01-20	2022-02-1
Vend-79	Vend-79	Mail	Con-197	2020-11-08	Europe	England	150585	9.5	Medium	Not Required	2022-11-08	2022-12-0
Vend-75	Vend-75	Insurance	Con-198	2020-04-16	Europe	Germany	45322	11.0	High	Yes	2021-04-16	2021-05-1
Vend-84	Vend-84	Administrative Support	Con-199	2020-10-19	North America	USA	78963	9.0	Medium	Not Required	2022-10-19	2022-11-1
Vend-64	Vend-64	Engineering	Con-200	2019-04-08	North America	Canada	195410	11.0	High	Yes	2020-04-07	2020-05-0

100 rows x 19 columns

In [91]: # The risk rating of a contract determines the frequency of reassessment; below is an aggregation and visualization of reassessment status across all contracts in the VRM Program

```
complete = 0
past_due_under_15 = 0
past_due_15_30 = 0
in_prog_due_under_15 = 0
in_prog_due_15_30 = 0
not_due_over_30 = 0

for reassessment_status in program_data['Reassessment Status']:
    if reassessment_status == 'Complete':
        complete += 1
    elif reassessment_status == 'Past Due 0 - 15 Days':
        past_due_under_15 += 1
    elif reassessment_status == 'Past Due 15 - 30 Days':
        past_due_15_30 += 1
    elif reassessment_status == 'In Progress - Due in < 15 Days':
        in_prog_due_under_15 += 1
    elif reassessment_status == 'In Progress - Due 15 - 30 Days':
        in_prog_due_15_30 += 1
    else:
        not_due_over_30 += 1

total_reassessment = complete + past_due_under_15 + past_due_15_30 + past_due_30_plus + in_prog_due_under_15 + in_prog_due_15_30 + not_due_over_30
print('Complete: ', complete)
print('Past Due in under 15 Days: ', past_due_under_15)
print('Past Due 15 to 30 Days: ', past_due_15_30)
print('Past Due over 30 Days: ', past_due_30_plus)
print('In Progress - Due in 15 Days or Less: ', in_prog_due_under_15)
print('In Progress - Due 15 - 30 Days: ', in_prog_due_15_30)
print('Not Due over 30 Days: ', not_due_over_30)
print('Total Contracts: ', total_reassessment)
```

Complete: 30
Past Due in under 15 Days: 3
Past Due 15 to 30 Days: 1
Past Due over 30 Days: 4
In Progress - Due in 15 Days or Less: 2
In Progress - Due 15 - 30 Days: 0
Not Due over 30 Days: 60
Total Contracts: 100

In [115]: # Put the sum of reassessment status into a dataframe file structure

```
reassessment_data = {'Complete': complete,
                     'Past Due in under 15 Days': past_due_under_15,
                     'Past Due 15 to 30 Days': past_due_15_30,
                     'Past Due over 30 Days': past_due_30_plus,
                     'In Progress - Due in 15 Days or Less': in_prog_due_under_15,
                     'In Progress - Due 15 - 30 Days': in_prog_due_15_30,
                     'Not Due over 30 Days': not_due_over_30}

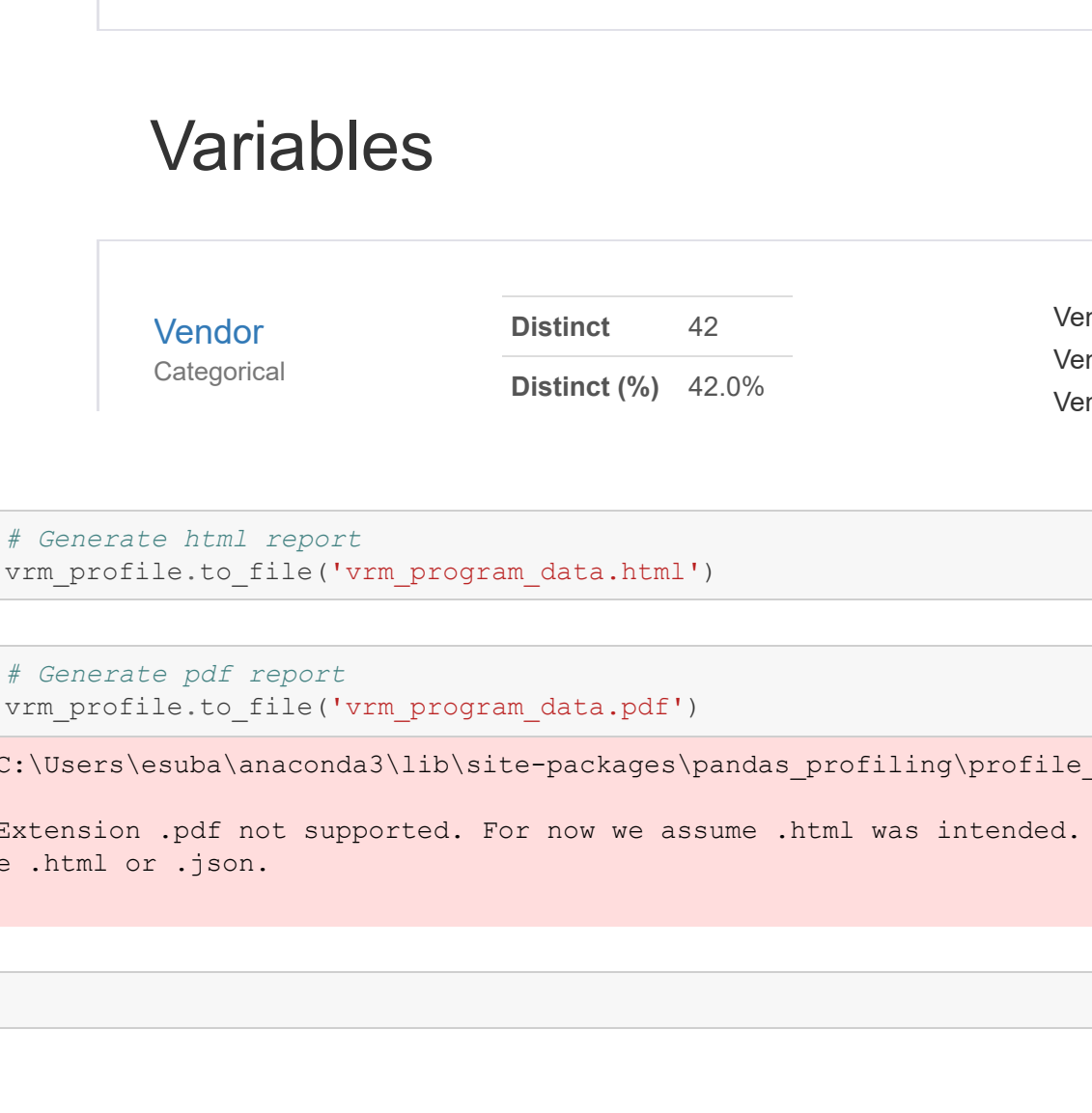
reassessment_dataframe = pd.DataFrame(reassessment_data, index = ['Reassessment Status'])
reassessment_dataframe.transpose()
reassessment_dataframe.transpose()
```

Out[115]:

	Reassessment Status
Complete	30
Past Due in under 15 Days	3
Past Due 15 to 30 Days	1
Past Due over 30 Days	4
In Progress - Due in 15 Days or Less	2
In Progress - Due 15 - 30 Days	0
Not Due over 30 Days	60

In [116]: # Bar Chart of reassessment status reassessment_dataframe.transpose().plot.bar(legend = True, grid = True, figsize = (10, 5), title = 'Status of Contracts Reassessment')

Out[116]: <AxesSubplot:title='center': 'Status of Contracts Reassessment'>



How Do We Determine Vendor Risk?

This is left out on purpose

In []:

In []:

In [117]: # Using pandas profiling package to streamline the data vrm_dataset = pd.read_excel('vrm_program_reporting.xlsx')

vrm_profile = ProfileReport(vrm_dataset)

vrm_profile

Pandas Profiling Report

Overview Variables Interactions Correlations Missing values Sample

Overview

Overview Warnings (16) Reproduction

Dataset statistics

Number of variables 19

Number of observations 100

Missing cells 0

Missing cells (%) 0.0%

Duplicate rows 0

Duplicate rows (%) 0.0%

Total size in memory 15.0 KiB

Average record size in memory 153.3 B

Variable types

Categorical 9

DateTime 3

Numeric 5

Unsupported 1

Boolean 1

Variables

Vendor

Distinct 42

Distinct (%) 42.0%

Vend-61 6

Vend-59 5

Vend-67 5

Out[117]:

In [118]: # Generate html report vrm_profile.to_file('vrm_program_data.html')

In [119]: # Generate pdf report vrm_profile.to_file('vrm_program_data.pdf')

C:\Users\vesuba\anaconda3\lib\site-packages\pandas_profiling\profile_report.py:293: UserWarning:

Extension .pdf not supported. For now we assume .html was intended. To remove this warning, please use .html or .json.

In []: