

▼ Data Analysis Code for BU Sustainability Project

▼ Part 1: Loading data

```
# Import Python functions
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from google.colab import drive
import datetime
%matplotlib inline
```

```
# Mount drive from Google Drive
drive.mount('/content/drive')
```

↳ Drive already mounted at /content/drive; to attempt to forcibly remount, call

```
# Loading University wide purchasing data as "df"
df = pd.read_excel('/content/drive/Shareddrives/Spark! and Sustainable Purchasing/I
df
```

	Requisition ID	Requisitioning Date	Supplier - ERP Supplier	Description	Requester - Shopper ID	Part Suppl: Part Numl
0	PR8853	2020-09-08	WB Mason Company	952 Ink Cartridges - Cyan, Magenta, Yellow, 3 ...	AMBERFW	HEWN9K27
1	PR32062	2021-01-05	WB Mason Company	61XL Ink Cartridge, Black (CH563WN)	TWARE	HEWCH563
2	PR12682	2020-09-24	WB Mason Company	63 Ink Cartridges - Black, Tri-color, 2 Cartri...	SARAHBF	HEWL0R46
3	PR31013	2020-12-21	WB Mason	62 Ink Cartridge,	JIGGOMF7	HFWC2P04

```
# Loading Department spending data as "df1"
df1 = pd.read_excel('/content/drive/Shareddrives/Spark! and Sustainable Purchasing')
df1 = df1.drop(columns=['Without\nPurchase Order', 'Unnamed: 19', 'Unnamed: 20']).c
df1
```

	ARAVO Vendor ID	Vendor	Vendor Name	Type of Business	PO Creation Date	Product P/N (Vendor)	Shop
0	10572031.0	10000800.0	WB Mason Company	OFFICE SUPPLIES & COMMERCIAL S	2016-01-06	GMT6505	TIAF
1	10572031.0	10000800.0	WB Mason Company	OFFICE SUPPLIES & COMMERCIAL S	2016-01-06	NES35110BX	TIAF
2	10572031.0	10000800.0	WB Mason Company	OFFICE SUPPLIES & COMMERCIAL S	2016-01-11	GMT4050	MIKE
3	10572031.0	10000800.0	WB Mason Company	OFFICE SUPPLIES & COMMERCIAL S	2016-01-11	ITD102042	MLMORI
4	10572031.0	10000800.0	WB Mason Company	OFFICE SUPPLIES & COMMERCIAL S	2016-01-12	HEWC9731A	AMACI
...
61047	10572031.0	10000800.0	WB Mason Company	OFFICE SUPPLIES & COMMERCIAL S	2020-06-04	GMT6003	ADASII
61048	10572031.0	10000800.0	WB Mason Company	OFFICE SUPPLIES & COMMERCIAL S	2020-06-04	GMT6520	ADASII
61049	10572031.0	10000800.0	WB Mason	OFFICE SUPPLIES & COMMERCIAL	2020-06-04	GMT6570	ADASII

```
# Loading paper purchasing data as "df2"
excel_reader=pd.ExcelFile('/content/drive/Shareddrives/Spark! and Sustainable Purchasing Data')
sheet_names = excel_reader.sheet_names
df2 = excel_reader.parse(sheet_name=sheet_names[1])
df2
```

	Requisition ID	Requisitioning Date	Supplier - ERP Supplier	Description	Request - Sho
0	SH449754	2018-11-26	WB Mason Company	59301110 5821035628 Astrobrights Color Paper...	C- ANDRE
1	SH449839	2018-11-26	WB Mason Company	C-59311433 5821081290 78A, (CE278A) Black Orig...	JMC
2	SH455528	2018-12-17	WB Mason Company	59869171 5823461205 Continuous Paper Label T...	C- CAN
3	SH452999	2018-12-06	WB Mason Company	C-59620403 5822419340 12A, (Q2612A) Black Orig...	JMALZ
4	SH453242	2018-12-06	WB Mason Company	59637773 5822492854 70C1HM0 (LEX-701HM) High...	C- JANE
...
7846	PR13081	2020-09-28	WB Mason Company	30% Recycled Copy Paper, 92 Bright, 20 lb., 8 ...	SAR/
7847	PR7995	2020-09-03	WB Mason Company	78 Extra Bright AND Heavy Copy Paper, 98 Bright...	CHR
7848	PR8853	2020-09-08	WB Mason	952 Ink Cartridge, Black (F6UJ15AN)	AMBE

▼ Part 2: Data Processing

Company

ууру : уру, уе у...

- ▼ Snapshot of the first 10 items in coffee and toner

Company

7851 rows × 16 columns

```
coffee_req, toner_req = df[df.Category=='Coffee'], df[df.Category=='Toner']
coffee_req[:10], toner_req[:10]
```

```
(      Requisition ID Requisitioning Date ... Part - Supplier Part Number Category
16228      SH218419    2016-01-06 ...          GMT6505   ( 
16229      SH218687    2016-01-06 ...          GMT6505   ( 
16230      SH218692    2016-01-06 ...          GMT6505   ( 
16231      SH218425    2016-01-06 ...          GMT6505   ( 
16232      SH219652    2016-01-11 ...          GMT4050   ( 
16233      SH219657    2016-01-11 ...          ITD102042  ( 
16234      SH219699    2016-01-11 ...          ITD102042  ( 
16235      SH219791    2016-01-12 ...          TWG08762   ( 
16236      SH220241    2016-01-13 ...          GMT6792   ( 
16237      SH220177    2016-01-13 ...          GMT4061   ( 

[10 rows x 7 columns],
      Requisition ID Requisitioning Date ... Part - Supplier Part Number Category
0      PR8853        2020-09-08 ...          HEWN9K27AN  Tone
1      PR32062       2021-01-05 ...          HEWCH563WN  Tone
2      PR12682       2020-09-24 ...          HEWL0R46AN  Tone
3      PR31013       2020-12-21 ...          HEWC2P04AN  Tone
4      PR28465       2020-12-09 ...          HEW3JA03AN  Tone
5      PR22162       2020-11-10 ...          HEWCN045AN  Tone
6      PR37770       2021-01-28 ...          HEWCC644WN  Tone
7      PR25272       2020-11-23 ...          HEWL0R46AN  Tone
8      PR24179-V2    2020-11-18 ...          HEWC2P05AN  Tone
9      PR22042       2020-11-09 ...          CNM2075C006  Tone

[10 rows x 7 columns])
```

▼ List of all unique items in coffee categorize

```
coffee_req_cat = coffee_req['Description'].str.split(',', expand=True)
coffee_req_cat[0].unique(), coffee_req_cat[1].unique(), coffee_req_cat[2].unique()

(array(['Tea K-Cups Sampler', 'C-19323837|5725967831|Tea K-Cups Sampler',
       'C-19324327|5725970348|Tea K-Cups Sampler', ...,
       'Coffee machine service to re-program and change water filter. The f',
       'Laminate Computer Table Desk with Drawer',
       'Caffe Espresso Ground Coffee'], dtype=object),
array([' Assorted Flavors', ' 24/Box', ' .375oz', ' 24/BX', ' 22/Box',
       ' Pike Place', ' Breakfast Blend', ' Paper', ' 8/10-Cup Size',
       ' Regular Ground', ' 24/ Box', ' None', ' Hazelnut', ' Country Roast',
       ' House Blend', ' Singles', ' Original', ' French Roast',
       ' Arabica', ' Cafe Verona', ' Fits 26 Pods',
       ' Classic Roast Ground', ' 25/BX', ' 25 Pods/Box',
       ' 5 2/5 x 11 x 12 3/5', ' Dark Roast', ' Sweetened Original',
       ' Fractional Packs', ' Black Silk', ' 12 Cup', ' 12-Cups',
       ' Signature-Level 3', ' French Vanilla', ' Cinnamon Vanilla',
```

' Pike Place Decaf', ' Caramel', ' 16 oz Canister', ' 2.2oz',
' 96/Carton', ' 1.5 Gallon Urn Style', ' 12-Cup Size',
' Caf Mocha', ' Hazelnut Creme', ' 2/BX', ' 2.2 oz. bags',
' Classic Roast', ' Verona', ' Light Roast', ' 8 Compartments',
' 10 to 12-Cups', ' 16/Box', ' Breakfast Blend Decaf',
' Vernanda Blend', ' Lemon', ' Ground', ' Medium Roast', ' Peach',
' Italian Sweet Creme', ' French Vanilla Flavor .375 oz.',
' Vanilla Caramel', ' Colombian', ' Pre-Measured', ' 25 Per Box',
' French Caramel', ' Pike Place Roast', ' Decaf', ' Whole Bean',
' 12 oz', ' Coffee Bean', ' Regular House Blend', ' 7 oz Bottle',
' Coffee', ' Sugar Free', ' Breakfast Blend-Level 2', ' Raspberry',
' Kona Blend', ' 42/CT', ' 2.2 OZ', ' 14 oz.', ' 12 Cups',
' 2 Ounces', ' Decaf Signature-Level 3', ' Mini Cups',
' 18 1/4 x 6 5/8 x 9 7/8', ' 5 1/2" Long', ' 2.5oz',
' 13 23/50 x 12 87/100 x 2 18/25', ' 22/BX',
' Raspberry Chocolate Truffle', ' Premium Blend',
' Ultimate Roast', ' Vienna', ' 9/BX', ' 16 oz. Bottle', ' 160/PK',
' Decaffeinated Ground Coffee', ' Red/White', ' 12oz Bottle',
' Brown', ' Classic Roast Decaffeinated', ' K-Cup Pods',
' Half Caff', ' Espresso', ' Espresso Roast', ' Get Up Stand Up',
' 1.5oz', ' 6 Gallon Urn Style', ' 100% Colombian Supremo',
' .44 oz.', ' Fraction Pack', ' 3-Gallon Urn Style',
' Colombian Decaf', ' Decaf Pike Place Roast', ' 100% Colombian',
' Morning Blend', ' 70/BX', ' Coffee Design',
' Swiss Hazelnut Coffee', ' 100% Colombian Coffee',
' 100% Colombian Decaf', ' 1.5 Gallon Brewer', ' 2.2 oz.',
' Black', ' Simmer Down Swiss Water Decaf',
' 100% Colombian Regular', ' 16oz', ' 2.5 oz.',
' Salted Caramel Chocolate', ' Caf Domingo',
" Major Dickason's Blend", ' 2.2lb Bag', ' Full City',
" Dunkin' Dark", ' Original Blend', ' Vanilla Hazelnut Decaf',
' Fits 72 Pods', ' Vanilla Hazelnut', ' Caf Domingo Blend', ' KCUP',
' Dulce De Leche', ' Sumatra', ' 3/25oz',
' Fair Trade Organic House Blend', ' Colombian Supremo',
' Reserve', ' Privateer Dark', ' Firefly Decaf',
' Donut Shop Original', ' Colombia Las Hermosas', ' 96/CT',
' Vermont Country Blend', ' 1.75 oz.', ' Naturally Decaffeinated',
' Colombian Blend', ' Level 4', ' Chocolate Coconut Cake',
' Insulated', ' Classic Bold', ' Guatemala', ' Vienna Espresso',
' Hazelnut Decaf', ' Smooth AND Rich Ground Coffee',
' Simply Smooth', ' Decaf Green Tea', ' 4 Compartments',
' Raspberry Chocolate Lava', ' Peppermint Bark',
' Jamaican Me Crazy', ' 10 Gallon Urn Style', ' 100 Count',
' Natural Cinnamon', ' Pumpkin Spice', ' 24 x 11 4/5 x 12 1/2',
' 0 38 oz Single-Serve Cups' ' 0 3 oz Single-Serve Cups'

▼ List of all unique items in toner categorize

```
toner_req_cat = toner_req['Description'].str.split(',', expand=True)
toner_req_cat[0].unique(), toner_req_cat[1].unique(), toner_req_cat[2].unique()
```

```
(array(['952 Ink Cartridges - Cyan', '61XL Ink Cartridge',
       '63 Ink Cartridges - Black', ..., 'C-29404873|5772035762|130A',
       'C-22196188|5738506362|X264H11G High-Yield Toner',
       'C-19635261|5727182732|HP 78A'], dtype=object),
array(['Magenta', 'Black (CH563WN)', 'Tri-color', 'Black (C2P04AN)',
       'Black (3JA03AN)', 'Black (CN045AN)', 'Tri-color (CC644WN)',
       'Black (C2P05AN)', 'Black', 'Black (F6U19AN)', 'Paper',
       'Black (T6M14AN)', 'Black (3YM56AN)', 'Black (N9K04AN)',
       'Black (F6U64AN)', 'Tri-color (N9J89AN)', 'Black (N9J90AN)',
       '2 Cartridges (3YN96AN)', 'Black (F6U62AN)',
       'Tri-color (F6U63AN)', 'Tri-color (N9K03AN)', 'Black (CH561WN)',
       'Black (F6U15AN)', '2 Cartridges (3JB40AN)',
       'Tri-color (CH562WN)',
       '(C9370A) Photo Black Original Ink Cartridge', 'Black (3YL65AN)',
       None, 'Magenta (CN055AN)', 'Black (CN049AN)', 'Cyan (CN054AN)',
       '(C9449A) Photo Black Original Ink Cartridge', 'Black (CN053AN)',
       'Yellow (CN056AN)', '(C9374A) Gray Original Ink Cartridge',
       '(C9372A) Magenta Original Ink Cartridge',
       '(C9373A) Yellow Original Ink Cartridge',
       '(C9403A) Matte Black Original Ink Cartridge', 'Black (C8767WN)',
       'Cyan (L0S61AN)', 'Yellow (L0S67AN)', 'Magenta (L0S64AN)',
       'Tri-color (F6U61AN)', 'Black (N9K02AN)', 'Tri-color (N9K01AN)',
       'Magenta (CN047AN)', 'Cyan (CN046AN)', 'Yellow (CN048AN)',
       '2 Cartridges (3YP22AN)', 'Tri-color (C2P06AN)',
       'Tri-color (CC656AN)', 'Black (CB336WN)', 'Black (CC641WN)',
       'Black (CC640WN)', 'Tri-color (C2P07AN)', 'Black (C4906AN)',
       'Cyan', 'Black (CC654AN)', '2 Cartridges (L0S28AN)',
       'Black (L0R39AN)', '2 Cartridges (3YP21AN)', 'Black (CN057AN)',
       '(C4912A) Magenta Original Ink Cartridge',
       '(N9H64FN) 2-pack Black/Tri-Color Original Ink Cartridges',
       '(CN069FN) 2-pack Black/Tri-color Original Ink Cartridges',
       '(CR314FN) 3-pack Cyan/Magenta/Yellow Original Ink Cartridges',
       '(C8766WN) Tri-color Original Ink Cartridge',
       '(CN045AN) High Yield Black Original Ink Cartridge',
       '(F6U62AN) Black Original Ink Cartridge',
       '(C8765WN) Black Original Ink Cartridge',
       '(C2P07AN) High Yield Tri-color Original Ink Cartridge',
       '(CB324WN) High Yield Magenta Original Ink Cartridge', '8oz',
       '(CD973AN) High Yield Magenta Original Ink Cartridge',
       '(C2P05AN) High Yield Black Original Ink Cartridge',
       '(CN060AN) Yellow Original Ink Cartridge',
       '(CN059AN) Magenta Original Ink Cartridge',
       '(CB322WN) High Yield Photo Black Original Ink Cartridge',
       '(CD974AN) High Yield Yellow Original Ink Cartridge',
       '(CD972AN) High Yield Cyan Original Ink Cartridge',
       '(CH561WN) Black Original Ink Cartridge', 'Assorted Inks',
       '(CN065FN) 3-pack Cyan/Magenta/Yellow Original Ink Cartridges',
       '(C8721WN) Black Original Ink Cartridge',
       '(C8773WN) Yellow Original Ink Cartridge',
       '(C8772WN) Magenta Original Ink Cartridge',
       '(C8771WN) Cyan Original Ink Cartridge', '1',
       '(CC644WN) High Yield Tri-color Original Ink Cartridge',
       'L0S67AN High Yield Yellow Original Ink Cartridge', '2',
       '(CN684WN) High Yield Black Original Ink Cartridge',
```

```
' L0S61AN High Yield Cyan Original Ink Cartridge',
' (CN049AN) Black Original Ink Cartridge',
' (CN052AN) Yellow Original Ink Cartridge',
' (CN050AN) Cyan Original Ink Cartridge',
' (CC610WN1) Black Original Ink Cartridge'
```

▼ List of all unique item in paper categorize

```
paper_df = df2[df2['Description'].str.contains('Ream')]
paper_df['Description'].unique()
```

```
array(['C-56417606|5808667711|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Bott
      'C-59489757|5821852877|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Crac
      'C-56050265|5806950473|BriteHue Multipurpose Colored Paper, 24lb, 8-1
      'C-59410909|5821512163|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Bott
      'C-56050265|5806951262|Recycled Colored Paper, 20lb, 11 x 17, Blue, E
      'C-58927034|5819472813|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Bott
      'C-58231079|5816514390|FIREWORX Colored Paper, 20lb, 8-1/2 x 14, Flas
      'C-58261715|5816646430|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Golc
      'C-59961900|5823845086|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Popr
      'C-59959747|5823843507|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Bott
      'C-59959747|5823843748|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Popr
      'C-59961900|5823845638|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Bott
      'C-59959525|5823834848|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Crac
      'C-59959525|5823834847|FIREWORX Colored Paper, 20lb, 11 x 17, Bottle
      'C-59959525|5823834849|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Popr
      'C-59959747|5823843515|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Crac
      'C-59961900|5823845621|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Crac
      'C-57185176|5812034967|BriteHue Multipurpose Colored Paper, 60lb, 8 1
      'C-57172990|5811982401|BriteHue Multipurpose Colored Paper, 60lb, 8 1
      'C-56952512|5811009660|FIREWORX Colored Paper, 20lb, 8-1/2 x 14, Flas
      'C-56955322|5811018063|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Popr
      'C-57442604|5813167373|Fore MP Multipurpose Paper, 96 Brightness, 201
      'C-58020470|5815608185|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Bott
      'C-58020470|5815608164|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Crac
      'C-58020470|5815608428|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Light
      "C-58020470|5815608489|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Jamn
      'C-58360343|5817123996|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Bott
      'C-57370661|5812861262|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Powc
      'C-57370661|5812861696|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Popr
      'C-57370661|5812860339|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Bott
      'C-58439448|5817398426|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Bott
      "C-58432650|5817371213|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Jamn
      'C-58562561|5817910900|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Lemc
      'C-58562561|5817919828|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Comk
      'C-58578304|5817977515|Great White 30 Recycled Copy Paper, 92 Brightr
      'C-58556570|5817884475|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Crac
      'C-58562561|5817910153|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Light
      'C-57140724|5811853505|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Smok
      'C-57140724|5811853511|BriteHue Multipurpose Colored Paper, 60lb, 8 1
      'C-57138468|5811827263|BriteHue Multipurpose Colored Paper, 60lb, 8 1
```

```
'C-57138468|5811827262|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Smok
'C-57151945|5811888964|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Smok
'C-57150838|5811884324|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Smok
'C-57151945|5811889410|BriteHue Multipurpose Colored Paper, 60lb, 8 1
'C-57150838|5811884336|BriteHue Multipurpose Colored Paper, 60lb, 8 1
'C-57721664|5814343584|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Aerj
'C-58927034|5819472960|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Crac
'C-59758808|5822995695|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Crac
'C-57126863|5811778885|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Aerj
'C-57126863|5811779764|FIREWORX Colored Paper, 24lb, 8-1/2 x 11, Emer
'C-59477035|5821797525|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Crac
'C-57482824|5813342261|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Bott
'C-59845569|5823364414|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Popr
'C-57482824|5813341639|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Popr
'C-57044317|5811413772|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Bott
'C-57993918|5815491823|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Powc
'C-58081646|5815876595|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Powc
'C-56063401|5807007812|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Bott
'C-56063401|5807007821|FIREWORX Colored Paper, 20lb, 8-1/2 x 11, Lumi
```

▼ Data cleaning for first excel data

```
coffee = coffee_req[~coffee_req['Part - Supplier Part Number'].str.contains("Unclas
toner = toner_req[~toner_req['Part - Supplier Part Number'].str.contains("Unclassifi
coffee.describe()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: FutureWarning:
  This is separate from the ipykernel package so we can avoid doing imports ur
```

	Requisition ID	Requisitioning Date	Supplier - ERP Supplier	Description	Requester - Shopper ID	Part - Supplier ID	Supplier Part Number
count	45193	45193	45193	45193	45193	45193	45193
unique	16285	1129	1	44166	798	573	
top	SH312986	2017-03-13 00:00:00	WB Mason Company	Original Liquid Coffee Creamer, 0.38 oz. Singl...	LSKINNER	GMT6520	

```
toner.describe()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: FutureWarning:  
    """Entry point for launching an IPython kernel.
```

	Requisition ID	Requisitioning Date	Supplier - ERP Supplier	Description	Requester - Shopper ID	Part Supplier Part Number
count	16221	16221	16221	16221	16221	16221
unique	8916	1088	1	15961	977	9
top	SH247682	2017-03-06 00:00:00	WB Mason Company	62XL Ink Cartridge, Black (C2P05AN)	TWARE	HEWCE50!

- ▼ List of all department University wide

```

df1['Unit name'].unique()

array(['ADMINISTRATIVE SERVICES', 'INFORMATION SERVICES & TECHNOLOGY',
       'SCHOOL OF MEDICINE (BUSM)', 'THE BU ACADEMY',
       'GENERAL ED SUPPORT & ACADEMIC INITIATIVES',
       'BUMC FACILITIES MANAGEMENT',
       'HENRY M GOLDMAN SCH OF DENTAL MED (GSDM)',
       'PHYSICAL DEVELOPMENT PROGRAMS & ATHLETIC', 'PROVOST',
       'QUESTROM SCHOOL OF BUSINESS (QST)',
       'COLLEGE OF ARTS AND SCIS (CAS)', 'CENTRAL ADMINISTRATION',
       'GLOBAL PROGRAMS', 'MET & EXTENDED EDUCATION ADMINISTRATION',
       'VICE PRESIDENT FINANCE', 'COLLEGE OF ENGINEERING (ENG)',
       'DEVELOPMENT & ALUMNI RELATIONS', 'METROPOLITAN COLLEGE (MET)',
       'NATL EMERGING INFECTIOUS DIS LAB (NEIDL)',
       'ENROLLMENT & STUDENT AFFAIRS', 'SDM DENTAL CLINICS',
       'SARGENT COLLEGE OF HEALTH & REHAB. SCI.',
       'SCHOOL OF PUBLIC HEALTH (SPH)', 'UNIV PROFESSORS-HONORS COLLEGE',
       'CRC FACILITIES MANAGEMENT & PLANNING',
       'COLLEGE OF FINE ARTS (CFA)', 'GRADUATE MEDICAL SCIENCES (GMS)',
       'ACADEMIC INSTITUTES & CENTERS', 'COLLEGE OF COMMUNICATION (COM)',
       'PARDEE SCHOOL OF GLOBAL STUDIES', 'UNIVERSITY LIBRARIES',
       'SCHOOL OF SOCIAL WORK (SSW)', 'FACULTY AND STAFF BENEFITS',
       'MET SPECIAL PROGRAMS NON-CREDIT (MSP)',
       'PHYSICAL EDUCATION, RECREATION & DANCE',
       'MEDICAL CAMPUS ADMINISTRATION (BUMC)', 'SCHOOL OF THEOLOGY (STH)',
       'SCHOOL OF HOSPITALITY (SHA)', 'OTHER OPS & AUXILIARY',
       'SCHOOL OF LAW (LAW)', 'WHEELOCK COLLEGE OF EDUC & HUMAN DEVELOP',
       'COLLEGE OF GENERAL STUDIES (CGS)', 'STUDENT LIFE',
       'NBL ACADEMIC PLANT', 'STUDENT HEALTH SERVICES',
       'HARRY AGGANIS ARENA', 'HUMAN RESOURCES', 'CRC RENTAL PROPERTIES',
       'CRC RESIDENCE LIFE', 'EVENTS & CONFERENCES',
       'CRC RESIDENCE HALLS', 'WBUR RADIO', 'GENERAL INSTITUTIONAL EXP',
       'CRC PARKING', 'MARKETING AND COMMUNICATION', 'SUMMER TERM (SUM)',
       'BUMC RENTAL PROPERTIES', 'MED CLINICAL TRIALS',
       'COMPUTING AND DATA SCIENCES'], dtype=object)

```

▼ Total request number for both coffee and toner during 2016-2021,
categorized by department

```
df1['Unit name'].value_counts()
```

SCHOOL OF MEDICINE (BUSM)	6616
HENRY M GOLDMAN SCH OF DENTAL MED (GSDM)	5721
COLLEGE OF ARTS AND SCIS (CAS)	5421
GENERAL ED SUPPORT & ACADEMIC INITIATIVES	3611
SCHOOL OF PUBLIC HEALTH (SPH)	2538
ADMINISTRATIVE SERVICES	2200
INFORMATION SERVICES & TECHNOLOGY	1973
METROPOLITAN COLLEGE (MET)	1757

ENROLLMENT & STUDENT AFFAIRS	1590
SCHOOL OF LAW (LAW)	1572
COLLEGE OF ENGINEERING (ENG)	1533
CRC FACILITIES MANAGEMENT & PLANNING	1521
SARGENT COLLEGE OF HEALTH & REHAB. SCI.	1511
VICE PRESIDENT FINANCE	1506
ACADEMIC INSTITUTES & CENTERS	1450
QUESTROM SCHOOL OF BUSINESS (QST)	1349
PROVOST	1299
PHYSICAL DEVELOPMENT PROGRAMS & ATHLETIC	1262
GRADUATE MEDICAL SCIENCES (GMS)	1101
COLLEGE OF FINE ARTS (CFA)	1095
MEDICAL CAMPUS ADMINISTRATION (BUMC)	1011
UNIVERSITY LIBRARIES	845
GLOBAL PROGRAMS	833
UNIV PROFESSORS-HONORS COLLEGE	811
WHEELOCK COLLEGE OF EDUC & HUMAN DEVELOP	799
SDM DENTAL CLINICS	625
MET & EXTENDED EDUCATION ADMINISTRATION	622
HUMAN RESOURCES	609
STUDENT LIFE	586
SCHOOL OF SOCIAL WORK (SSW)	563
PARDEE SCHOOL OF GLOBAL STUDIES	556
NBL ACADEMIC PLANT	540
SCHOOL OF THEOLOGY (STH)	513
STUDENT HEALTH SERVICES	502
EVENTS & CONFERENCES	499
CRC RENTAL PROPERTIES	492
NATL EMERGING INFECTIOUS DIS LAB (NEIDL)	462
CENTRAL ADMINISTRATION	432
WBUR RADIO	403
DEVELOPMENT & ALUMNI RELATIONS	338
COLLEGE OF COMMUNICATION (COM)	320
CRC RESIDENCE LIFE	280
FACULTY AND STAFF BENEFITS	226
COLLEGE OF GENERAL STUDIES (CGS)	193
THE BU ACADEMY	174
MARKETING AND COMMUNICATION	168
BUMC FACILITIES MANAGEMENT	167
MET SPECIAL PROGRAMS NON-CREDIT (MSP)	159
PHYSICAL EDUCATION, RECREATION & DANCE	149
HARRY AGGANIS ARENA	121
SUMMER TERM (SUM)	96
SCHOOL OF HOSPITALITY (SHA)	93
CRC RESIDENCE HALLS	69
OTHER OPS & AUXILIARY	57
GENERAL INSTITUTIONAL EXP	55
CRC PARKING	47
BUMC RENTAL PROPERTIES	5
COMPUTING AND DATA SCIENCES	1
MED CLINICAL TRIALS	1

Name Unit Name dvnum intfa

Total request number for coffee during 2016-2021, categorized by department

```

coffee_req1, toner_req1 = df1[df1.Category=='Coffee'], df1[df1.Category=='Toner']
s_date = datetime.datetime.strptime('20160101', '%Y%m%d')
e_date = datetime.datetime.strptime('20161231', '%Y%m%d')
y2016_req = df1[(df1['P0 Creation Date']>=s_date) & (df1['P0 Creation Date'] <= e_date)]

s_date2 = datetime.datetime.strptime('20170101', '%Y%m%d')
e_date2 = datetime.datetime.strptime('20171231', '%Y%m%d')
y2017_req = df1[(df1['P0 Creation Date']>=s_date2) & (df1['P0 Creation Date'] <= e_date2)]

s_date3 = datetime.datetime.strptime('20180101', '%Y%m%d')
e_date3 = datetime.datetime.strptime('20181231', '%Y%m%d')
y2018_req = df1[(df1['P0 Creation Date']>=s_date3) & (df1['P0 Creation Date'] <= e_date3)]

s_date4 = datetime.datetime.strptime('20190101', '%Y%m%d')
e_date4 = datetime.datetime.strptime('20191231', '%Y%m%d')
y2019_req = df1[(df1['P0 Creation Date']>=s_date4) & (df1['P0 Creation Date'] <= e_date4)]

s_date5 = datetime.datetime.strptime('20200101', '%Y%m%d')
e_date5 = datetime.datetime.strptime('20201231', '%Y%m%d')
y2020_req = df1[(df1['P0 Creation Date']>=s_date5) & (df1['P0 Creation Date'] <= e_date5)]

test1 = pd.Series(coffee_req1['Unit name'].value_counts(), name="total")
test2 = pd.Series(y2016_req['Unit name'].value_counts(), name="2016")
test3 = pd.Series(y2017_req['Unit name'].value_counts(), name="2017")
test4 = pd.Series(y2018_req['Unit name'].value_counts(), name="2018")
test5 = pd.Series(y2019_req['Unit name'].value_counts(), name="2019")
test6 = pd.Series(y2020_req['Unit name'].value_counts(), name="2020")
df3 = pd.concat([test1, test2, test3, test4, test5], axis=1).fillna(0)
df3

```

	total	2016	2017	2018	2019
HENRY M GOLDMAN SCH OF DENTAL MED (GSDM)	4062	962.0	992.0	965.0	895.0
SCHOOL OF MEDICINE (BUSM)	3722	825.0	939.0	897.0	843.0
COLLEGE OF ARTS AND SCIS (CAS)	3366	727.0	835.0	895.0	752.0
GENERAL ED SUPPORT & ACADEMIC INITIATIVES	2696	591.0	636.0	579.0	674.0
ADMINISTRATIVE SERVICES	2014	471.0	470.0	503.0	478.0
INFORMATION SERVICES & TECHNOLOGY	1812	355.0	515.0	600.0	280.0
SCHOOL OF PUBLIC HEALTH (SPH)	1769	469.0	452.0	375.0	391.0

ENROLLMENT & STUDENT AFFAIRS	1558	293.0	319.0	421.0	442.0
METROPOLITAN COLLEGE (MET)	1512	346.0	371.0	372.0	354.0
VICE PRESIDENT FINANCE	1385	349.0	324.0	274.0	359.0
CRC FACILITIES MANAGEMENT & PLANNING	1360	255.0	341.0	345.0	258.0
ACADEMIC INSTITUTES & CENTERS	1247	205.0	263.0	399.0	308.0
PROVOST	1161	156.0	288.0	311.0	327.0
SCHOOL OF LAW (LAW)	1154	230.0	288.0	278.0	275.0
SARGENT COLLEGE OF HEALTH & REHAB. SCI.	1107	213.0	266.0	334.0	259.0
PHYSICAL DEVELOPMENT PROGRAMS & ATHLETIC	1042	249.0	262.0	257.0	215.0
QUESTROM SCHOOL OF BUSINESS (QST)	984	241.0	279.0	209.0	205.0
COLLEGE OF ENGINEERING (ENG)	956	105.0	177.0	272.0	319.0
GRADUATE MEDICAL SCIENCES (GMS)	916	315.0	280.0	143.0	148.0
UNIV PROFESSORS-HONORS COLLEGE	805	192.0	259.0	289.0	58.0
MEDICAL CAMPUS ADMINISTRATION (BUMC)	767	187.0	172.0	174.0	184.0
UNIVERSITY LIBRARIES	715	152.0	189.0	199.0	153.0
COLLEGE OF FINE ARTS (CFA)	665	140.0	202.0	139.0	138.0
GLOBAL PROGRAMS	539	86.0	119.0	183.0	121.0
STUDENT LIFE	524	75.0	108.0	157.0	157.0
WHEELOCK COLLEGE OF EDUC & HUMAN DEVELOP	506	122.0	151.0	122.0	81.0
NBL ACADEMIC PLANT	505	74.0	97.0	113.0	136.0
HUMAN RESOURCES	505	108.0	139.0	116.0	123.0
PARDEE SCHOOL OF GLOBAL STUDIES	505	106.0	107.0	123.0	144.0
MET & EXTENDED EDUCATION ADMINISTRATION	498	121.0	119.0	117.0	115.0
CRC RENTAL PROPERTIES	485	98.0	145.0	130.0	91.0
EVENTS & CONFERENCES	474	107.0	115.0	105.0	105.0

- ▼ Total request number for toner during 2016-2021, categorized by department

STUDFNT HF&I TH SFRVICS	431	30.0	88.0	134.0	139.0
------------------------------------	-----	------	------	-------	-------

```
s_date = datetime.datetime.strptime('20160101', '%Y%m%d')
e_date = datetime.datetime.strptime('20161231', '%Y%m%d')
y2016_req = df1[(df1['PO Creation Date']>=s_date) & (df1['PO Creation Date'] <= e_date)]
```

```

s_date2 = datetime.datetime.strptime('20170101', '%Y%m%d')
e_date2 = datetime.datetime.strptime('20171231', '%Y%m%d')
y2017_req = df1[(df1['PO Creation Date']>=s_date2) & (df1['PO Creation Date'] <= e_]

s_date3 = datetime.datetime.strptime('20180101', '%Y%m%d')
e_date3 = datetime.datetime.strptime('20181231', '%Y%m%d')
y2018_req = df1[(df1['PO Creation Date']>=s_date3) & (df1['PO Creation Date'] <= e_]

s_date4 = datetime.datetime.strptime('20190101', '%Y%m%d')
e_date4 = datetime.datetime.strptime('20191231', '%Y%m%d')
y2019_req = df1[(df1['PO Creation Date']>=s_date4) & (df1['PO Creation Date'] <= e_]

s_date5 = datetime.datetime.strptime('20200101', '%Y%m%d')
e_date5 = datetime.datetime.strptime('20201231', '%Y%m%d')
y2020_req = df1[(df1['PO Creation Date']>=s_date5) & (df1['PO Creation Date'] <= e_]

test1 = pd.Series(toner_req1['Unit name'].value_counts(), name="total")
test2 = pd.Series(y2016_req['Unit name'].value_counts(), name="2016")
test3 = pd.Series(y2017_req['Unit name'].value_counts(), name="2017")
test4 = pd.Series(y2018_req['Unit name'].value_counts(), name="2018")
test5 = pd.Series(y2019_req['Unit name'].value_counts(), name="2019")
test6 = pd.Series(y2020_req['Unit name'].value_counts(), name="2020")
df4 = pd.concat([test1, test2, test3, test4, test5], axis=1).fillna(0)
df4

```

	total	2016	2017	2018	2019
SCHOOL OF MEDICINE (BUSM)	2894	666.0	755.0	683.0	627.0
COLLEGE OF ARTS AND SCIS (CAS)	2055	522.0	549.0	442.0	433.0
HENRY M GOLDMAN SCH OF DENTAL MED (GSDM)	1659	386.0	414.0	466.0	303.0
GENERAL ED SUPPORT & ACADEMIC INITIATIVES	915	255.0	233.0	179.0	205.0
SCHOOL OF PUBLIC HEALTH (SPH)	769	199.0	211.0	169.0	164.0
COLLEGE OF ENGINEERING (ENG)	577	163.0	129.0	156.0	116.0
COLLEGE OF FINE ARTS (CFA)	430	107.0	124.0	100.0	80.0
SCHOOL OF LAW (LAW)	418	99.0	86.0	105.0	101.0
SARGENT COLLEGE OF HEALTH & REHAB. SCI.	404	98.0	106.0	92.0	89.0
QUESTROM SCHOOL OF BUSINESS (QST)	365	81.0	74.0	101.0	82.0
GLOBAL PROGRAMS	294	51.0	67.0	94.0	60.0
WHEELOCK COLLEGE OF EDUC & HUMAN DEVELOP	293	75.0	46.0	82.0	75.0
SCHOOL OF SOCIAL WORK (SSW)	281	115.0	75.0	56.0	30.0

METROPOLITAN COLLEGE (MET)	245	82.0	81.0	46.0	32.0
MEDICAL CAMPUS ADMINISTRATION (BUMC)	244	61.0	60.0	60.0	53.0
WBUR RADIO	239	18.0	49.0	67.0	77.0
PHYSICAL DEVELOPMENT PROGRAMS & ATHLETIC	220	69.0	43.0	42.0	54.0
ACADEMIC INSTITUTES & CENTERS	203	33.0	38.0	67.0	51.0
ADMINISTRATIVE SERVICES	186	40.0	46.0	52.0	43.0
GRADUATE MEDICAL SCIENCES (GMS)	185	69.0	46.0	29.0	32.0
CRC FACILITIES MANAGEMENT & PLANNING	161	23.0	28.0	45.0	40.0
INFORMATION SERVICES & TECHNOLOGY	161	56.0	56.0	35.0	12.0
SDM DENTAL CLINICS	160	40.0	57.0	29.0	26.0
PROVOST	138	47.0	45.0	19.0	22.0
UNIVERSITY LIBRARIES	130	26.0	22.0	39.0	36.0
MET & EXTENDED EDUCATION ADMINISTRATION	124	26.0	40.0	34.0	23.0
VICE PRESIDENT FINANCE	121	27.0	31.0	29.0	23.0
SCHOOL OF THEOLOGY (STH)	115	34.0	32.0	25.0	23.0
NATL EMERGING INFECTIOUS DIS LAB (NEIDL)	111	28.0	23.0	32.0	26.0
HUMAN RESOURCES	104	32.0	33.0	20.0	14.0
COLLEGE OF COMMUNICATION (COM)	104	39.0	14.0	25.0	19.0
THEATRE DEPARTMENT	88	88.0	11.0	10.0	11.0

Part 3: Visual Demonstration

DEVELOPMENT & ALUMNI RELATIONS 73 7.0 3.0 15.0 30.0

- Order request counts for coffee request among the whole university, categorized by date

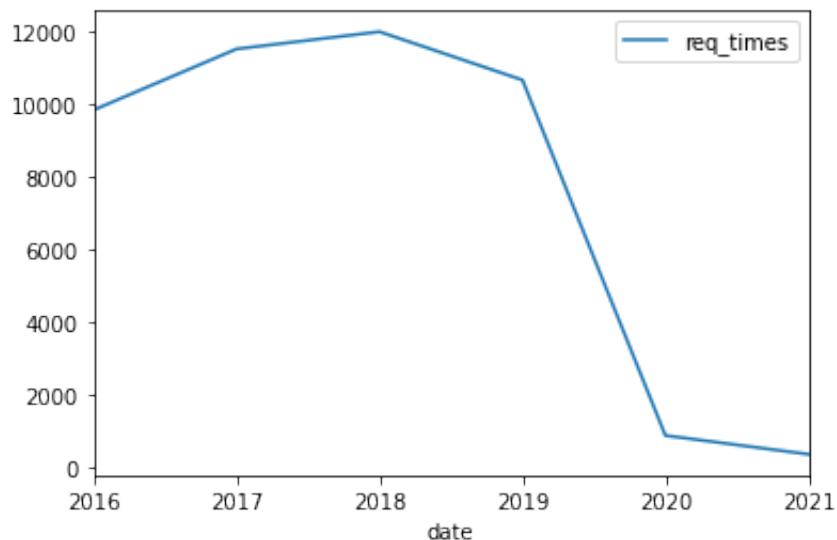
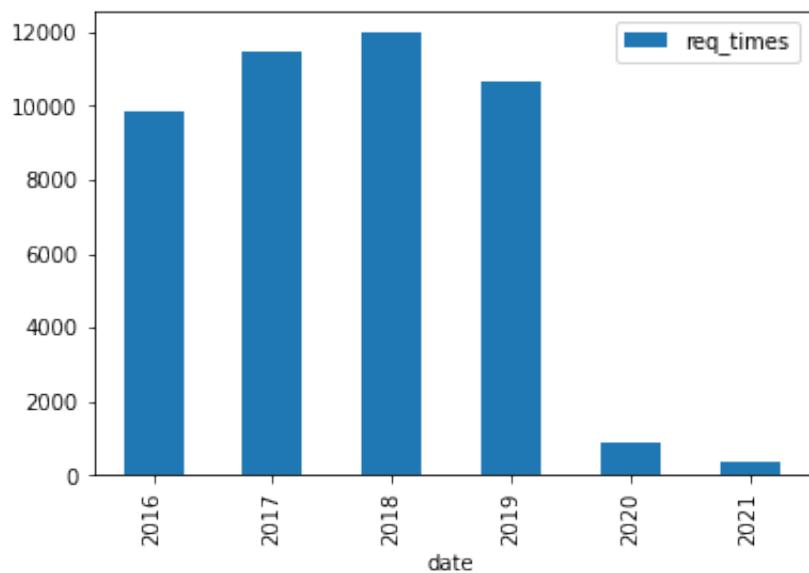
```
## Now processing the data from coffee_req
coffee_req_num_by_date = coffee_req.groupby(['Requisitioning Date','Category']).size()
coffee_req_num_by_date.columns = ['date','req_times']
cqd = coffee_req_num_by_date.set_index('date')
cqm = cqd.resample('MS').sum().to_period('M')
cqy = cqd.resample('YS').sum().to_period('Y')
cqy.plot(kind='bar')
cqy.plot()
```

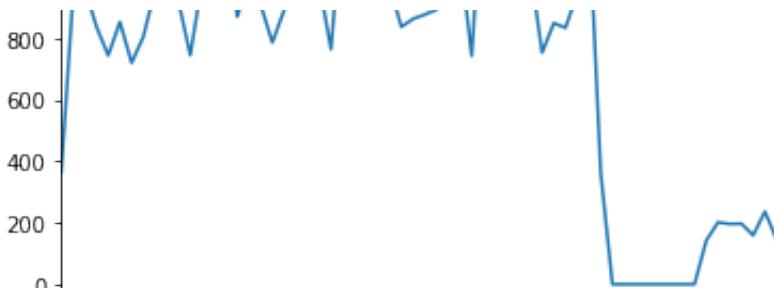
```
ggm.plot()
```

req_times

date

date	req_times
2016	9820
2017	11494
2018	11967
2019	10644
2020	895
2021	382

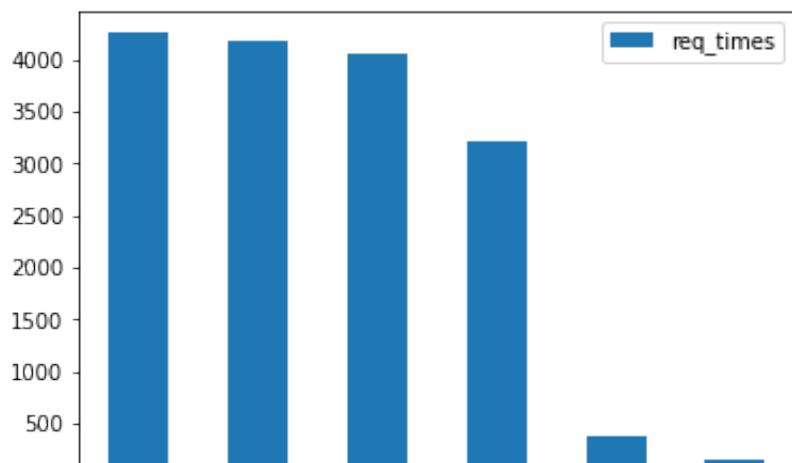


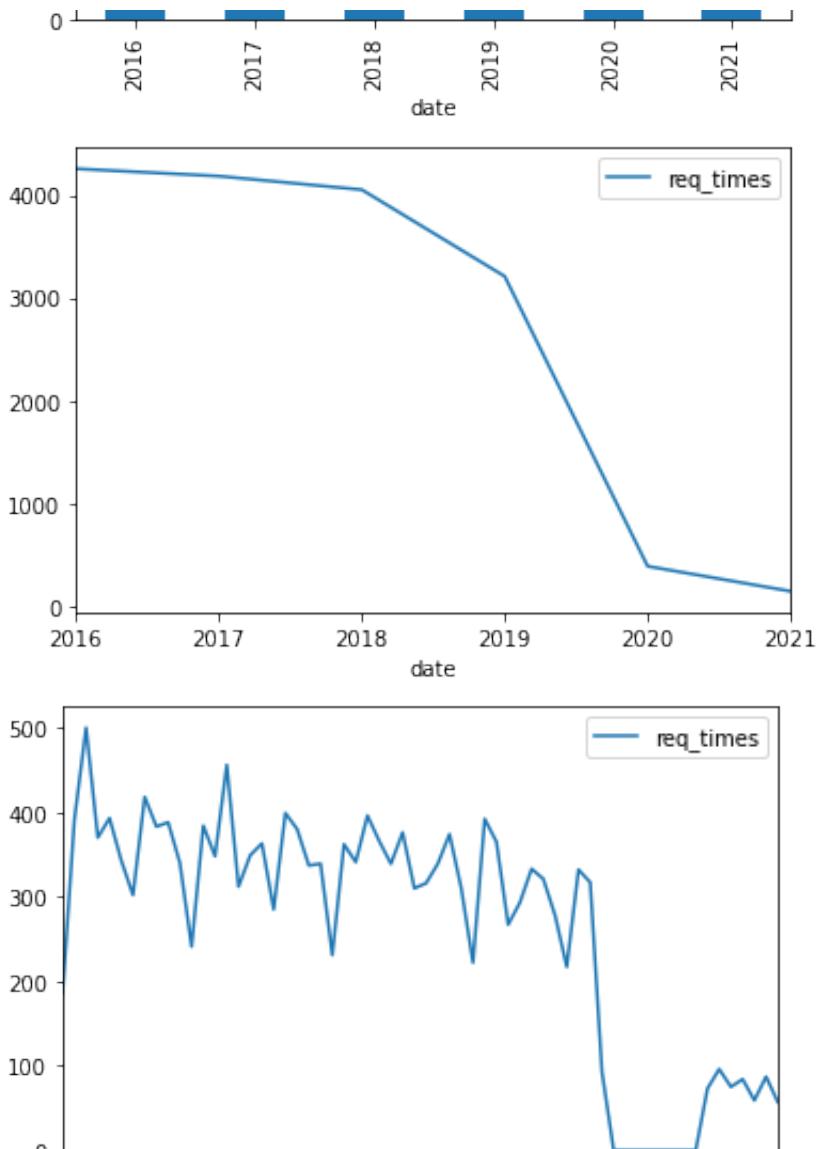


Order request counts for toner request among the whole university,
categorized by date

```
## Now processing the data from toner_req
toner_req_num_by_date = toner_req.groupby(['Requisitioning Date','Category']).size()
toner_req_num_by_date.columns = ['date','req_times']
tqd = toner_req_num_by_date.set_index('date')
tqm = tqd.resample('MS').sum().to_period('M')
tqy = tqd.resample('YS').sum().to_period('Y')
tqy.plot(kind='bar')
tqy.plot()
tqm.plot()
tqy
```

req_times	
date	req_times
2016	4255
2017	4183
2018	4051
2019	3208
2020	387
2021	144





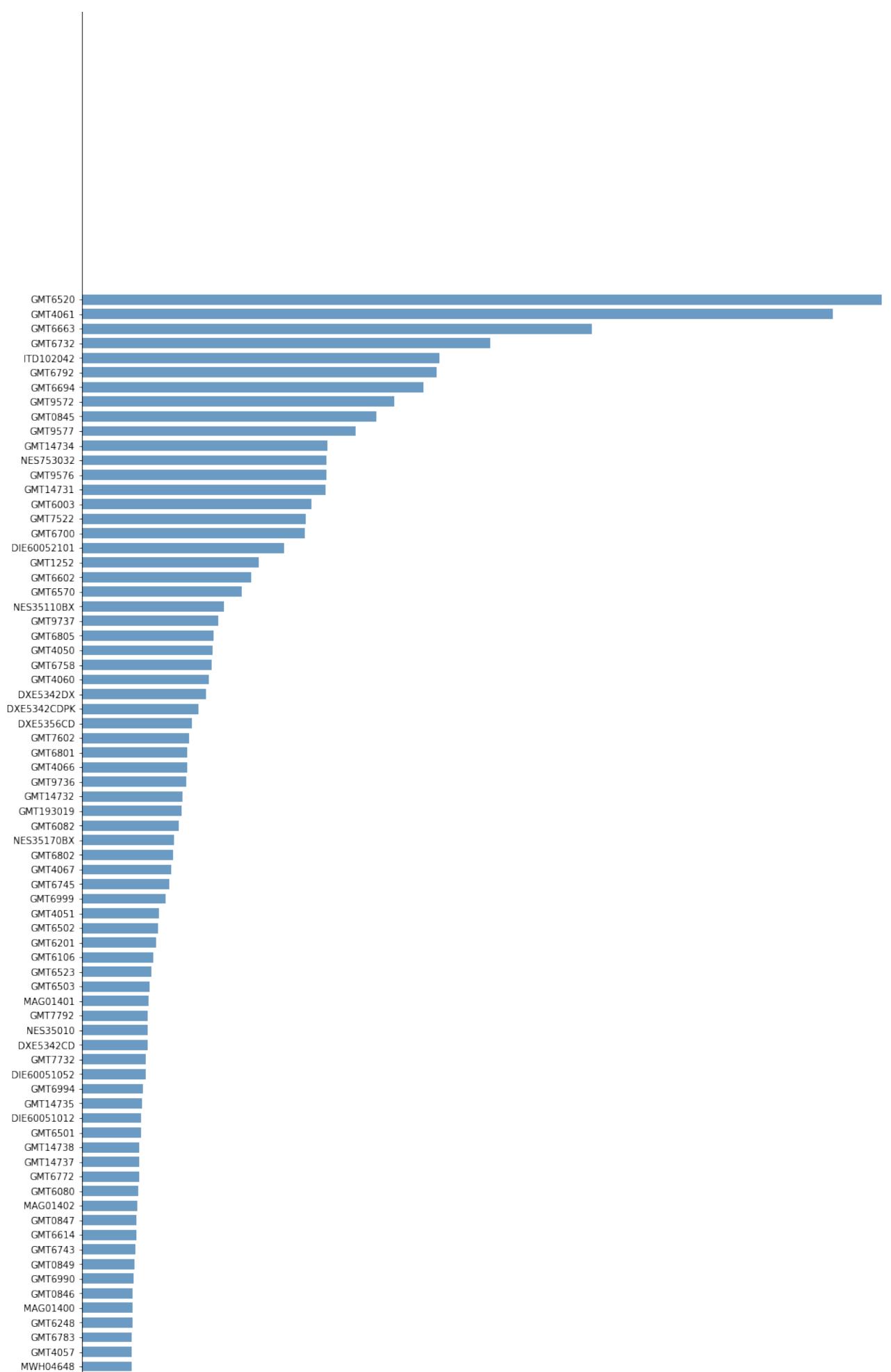
▼ All requested coffee products and their rank by order requests count

```

items, counts = np.unique(sorted(coffee['Part - Supplier Part Number']), return_counts=True)
temp = pd.DataFrame(data={'items': items, 'counts': counts}).sort_values(by=['counts'], ascending=False)
items, counts = temp['items'], temp['counts']

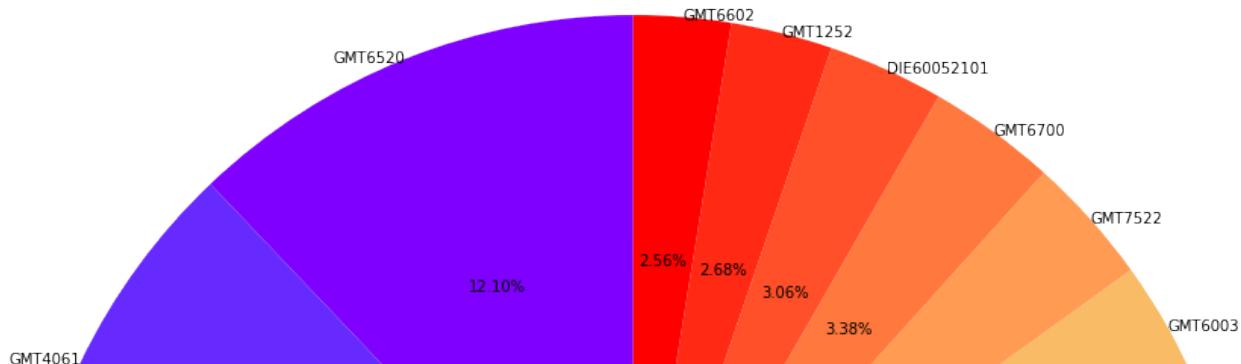
plt.figure(figsize=(16,180))
plt.barh(range(len(counts)), counts, height=0.7, color='steelblue', alpha=0.8)
plt.yticks(range(len(counts)), items)
plt.show()

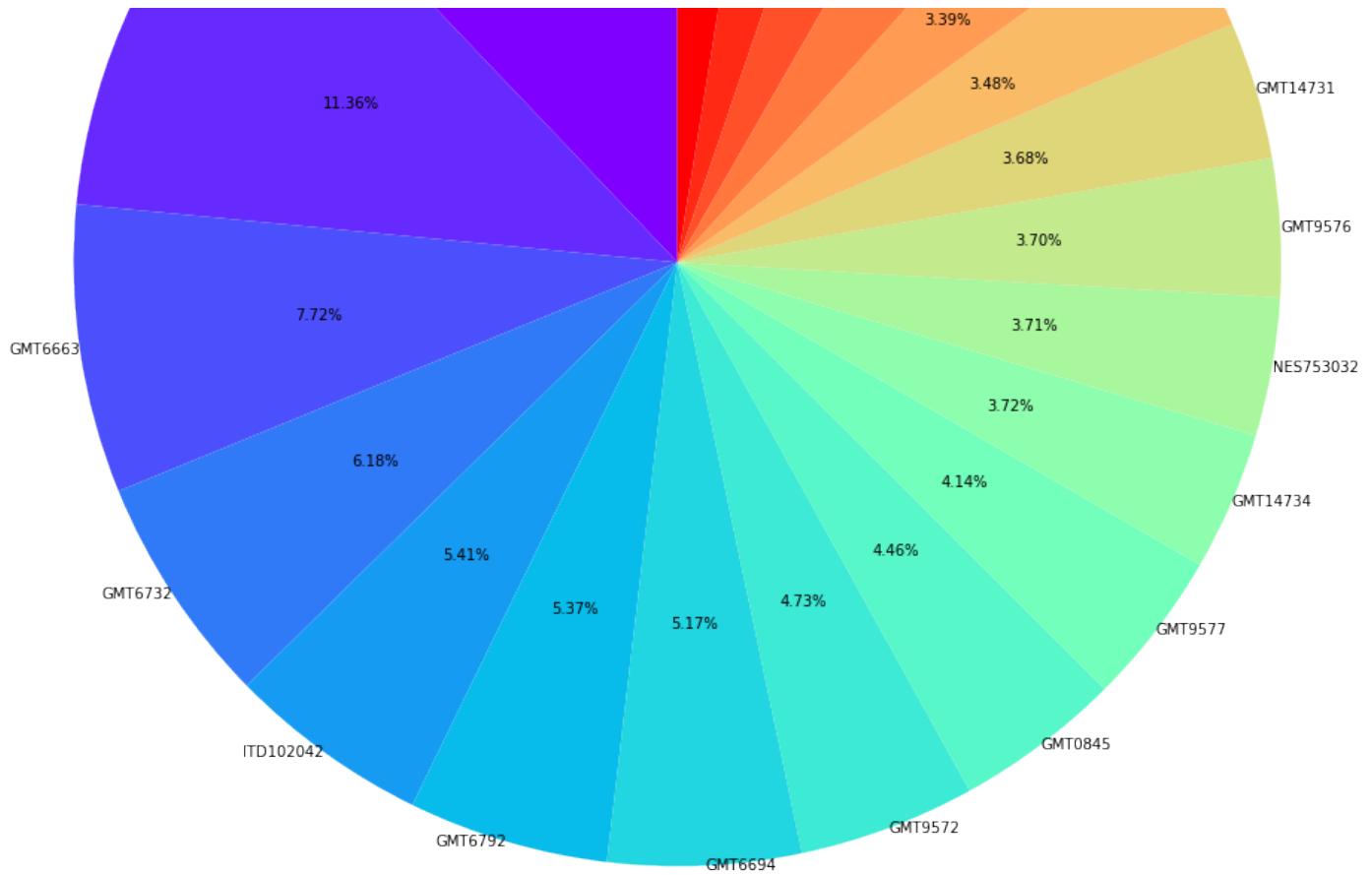
```



▼ Top 20 requested coffee products and their proportions

```
items, counts = np.unique(sorted(coffee['Part – Supplier Part Number']), return_cour  
temp = pd.DataFrame(data={'items': items,'counts': counts}).sort_values(by=['counts'])  
items, counts = temp['items'][-20:][::-1],temp ['counts'][-20:][::-1]  
  
plt.figure(figsize=(16,24)) #resize the figure  
labels = items #define the labels as Supplier P  
sizes = counts #define the size corresponds to  
colors = cm.rainbow(np.linspace(0, 1, len(labels)))#define the color for each secto  
explode = (0,)*len(sizes) #make distance between sectors  
  
patches,text1,text2 = plt.pie(sizes, #no shadow  
                           explode=explode, #counter-clockwise  
                           labels=labels, #value distance from the center  
                           colors=colors,  
                           labeldistance = 1,  
                           autopct = '%3.2f%%',  
                           shadow = False,  
                           startangle =90,  
                           pctdistance = 0.6)  
  
plt.axis('equal')  
plt.show()
```





▼ All requested toner products and their rank by order requests count

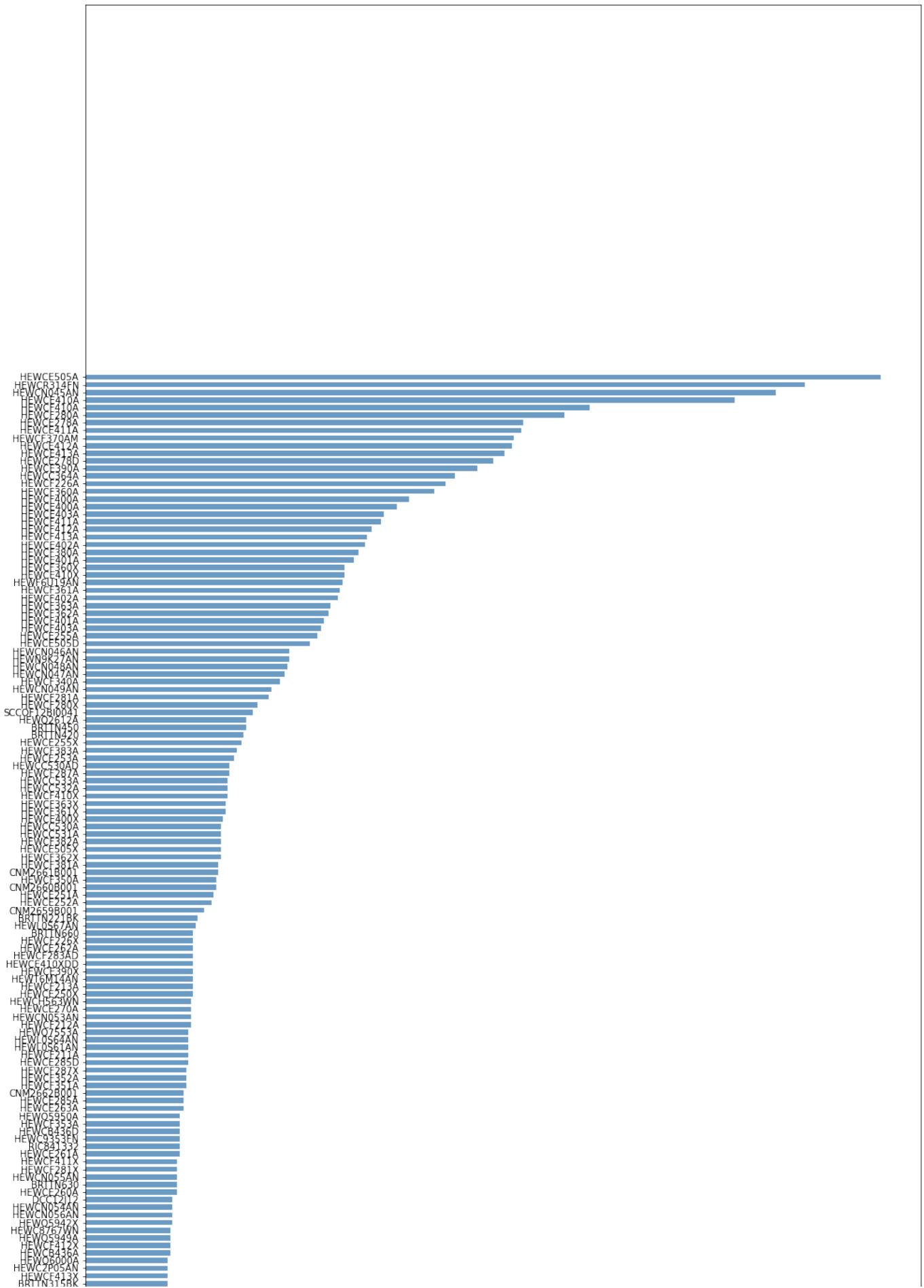
```

items, counts =np.unique(sorted(toner['Part - Supplier Part Number']), return_count
temp = pd.DataFrame(data={'items': items,'counts': counts}).sort_values(by=['counts'])
items, counts = temp['items'],temp ['counts']

plt.figure(figsize=(16,160))
plt.barh(range(len(counts)), counts, height=0.7, color='steelblue', alpha=0.8)
plt.yticks(range(len(counts)), items)

```

```
plt.show()
```





▼ Top 20 requested toner products and their proportions

```

HEWCE505A [  ] |  

HEWCF400X [  ] |  

HEWC1250A [  ] |  

items, counts = np.unique(sorted(toner['Part - Supplier Part Number']), return_count=True)  

temp = pd.DataFrame(data={'items': items, 'counts': counts}).sort_values(by=['counts'], ascending=False)  

items, counts = temp['items'][-20:][::-1], temp ['counts'][-20:][::-1]  

plt.figure(figsize=(16,24)) #resize the figure  

labels = items #define the labels as Supplier F  

sizes = counts #define the size corresponds to  

colors = cm.rainbow(np.linspace(0, 1, len(labels)))#define the color for each sector  

explode = (0,)*len(sizes) #make distance between sectors  

patches, text1, text2 = plt.pie(sizes, #no shadow  

                               explode=explode, #counter-clockwise  

                               labels=labels, #value distance from the center  

                               colors=colors,  

                               labeldistance = 1,  

                               autopct = '%3.2f%%',  

                               shadow = False,  

                               startangle =90,  

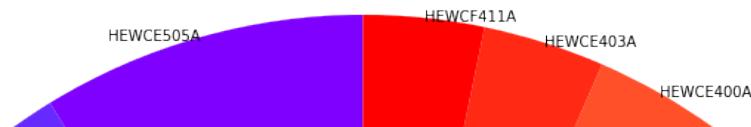
                               pctdistance = 0.6)  

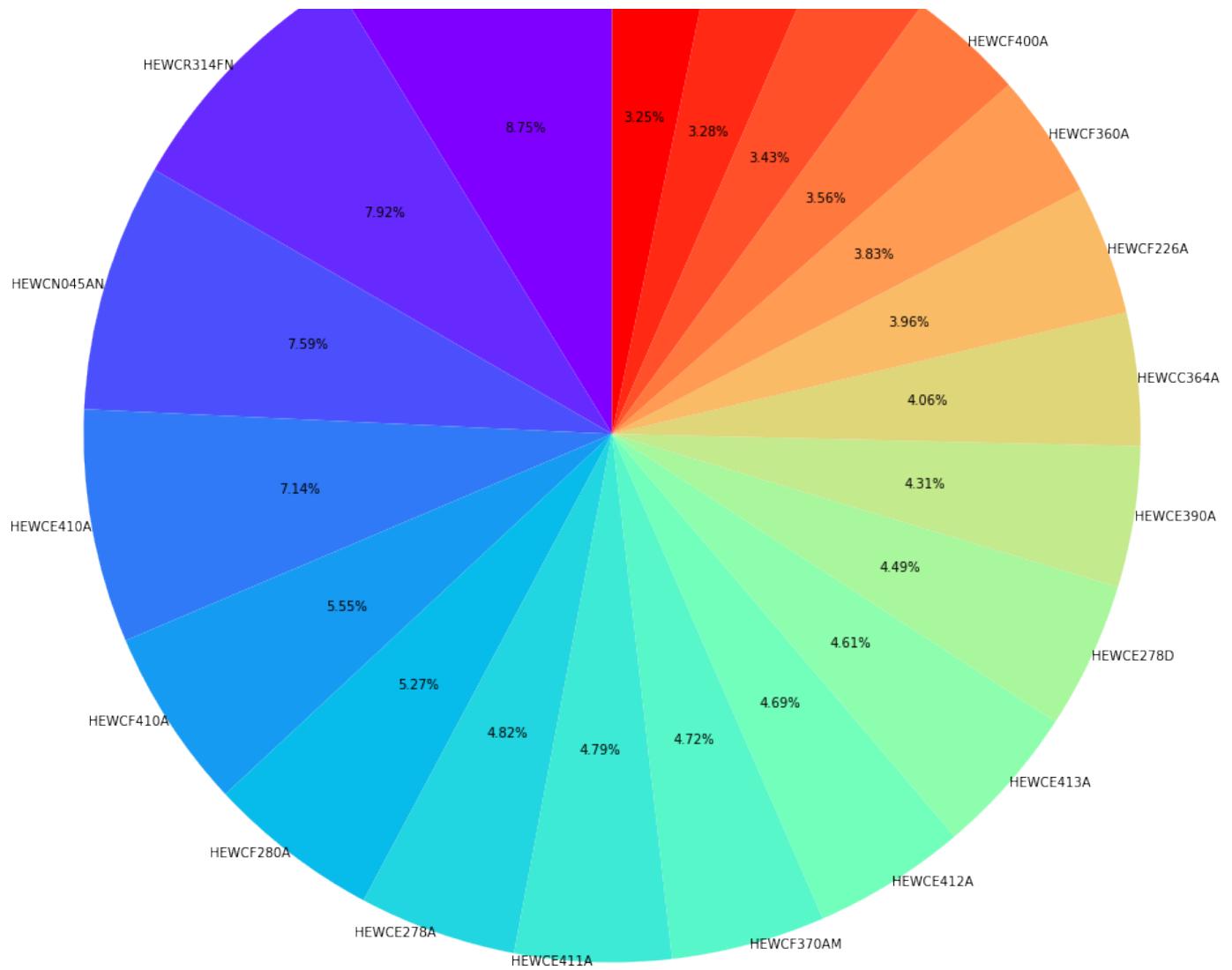
  

plt.axis('equal')  

plt.show()

```





Components for a single unit of plastic coffee pod

total: 3.0213 g

Shell total: 1.7908 g + 0.7841 g (ring) = 2.5749 g [Polypropylene]

Lid total: 0.2462 g = Aluminum (24%) PET coating (12%) LDPE (50%) Printing ink (14%)

Filter Total: 0.2002 g (6.63%) = Kraft paper, bleached (79%) + LDPE (21%)

reference: <https://www.nature.com/articles/s41598-020-65058-1/tables/3>

NEWC9426A

Waste weight of the top 18 consumed k-cup pods university wide during year 2016-2021

HEWC9388AN

```
items, counts = np.unique(sorted(coffee['Part - Supplier Part Number']), return_counts=True)
temp = pd.DataFrame(data={'items': items, 'counts': counts}).sort_values(by=['counts'], ascending=False)
items, counts = temp['items'][-20:][::-1], temp['counts'][-20:][::-1]
total_number = sum([i * 24 for i in counts]) - counts.tolist()[4] - counts.tolist()[-1]
print("the total number of eighteen most used K-cup university wide during year 2011 is", total_number)
```

the total number of eighteen most used K-cup university wide during year 2016-

LEX50F1000
HEWCB1632AN
HEWB87Y1A
IVRFEX7
HEWC5040AM
LEWXCD11H9
HEWCFA5G3D
CNM7621A001LA
HEWC482XZ
HEWCF7470A
IVRES50XSE1
VRB376FC
CNM1890B0U
HEWL055SAN
HEWL0527AN
HEWNKG03AN
VRBN121M
VRBN122N
HEWC2075PN
IVRF226X
VRBN1362
HEWB259IA
HEWB259IA
CNM34798001
DXED954001
HEWCF480X
HWEV480X
DPDSPC785AP
HEW3VPO0AN
HEWT0A53AN
VRBD3130
LEXC944ATYG
HEWC400XD
HEWC371AM
HEWMV62020
LEXC540H1TKG
CPC01903C1
CNMFXM3
ONM93435B0V
HEWV1000
LEXC54011M0
LEXB0C1DM0
XER106R01410
HEW5982EN
HECWG5982EN
HEWB6Y15A
HEWD0611A
XER106R01596
XER108R01850
XER108R01850
XER108R01854
HEWC5352AN
IVRCS51ASEL
IVR104
ivr83053596
HEWB6Y16A
IVRC5264EN
HEWF9345WEN

```

(Polypropylene,Aluminum,PET,LDPE,Printing,paper) = (2.5749 * total_number,
                                                    0.0591 * total_number,
                                                    0.0295 * total_number,
                                                    ( 0.1231 + 0.042 )* total_number,
                                                    0.0345 * total_number,
                                                    0.1582 * total_number)

print("the total waste weight of the top 18 consumed k-cup pods university wide dur
print("total weight: ", 3.0213 * total_number, " g")
print("total weight of Polypropylene: ", 2.5749 * total_number, " g")
print("total weight of Aluminum:", 0.0591 * total_number, "g")

print("total weight of PET coating: ", 0.0295 * total_number, " g")
print("total weight of LDPE: ",( 0.1231 + 0.042 )* total_number, " g")
print("total weight of Printing ink: ", 0.0345 * total_number, " g")
print("total weight of bleached Kraft paper: ", 0.1582 * total_number, " g")

```

the total waste weight of the top 18 consumed k-cup pods university wide dur
total weight: 1512184.8204 g
total weight of Polypropylene: 1288758.0492 g
total weight of Aluminum: 29580.0228 g
total weight of PET coating: 14764.985999999999 g
total weight of LDPE: 82633.8708 g
total weight of Printing ink: 17267.526 g
total weight of bleached Kraft paper: 79180.3656 g

```

>MIKMA21INITA
>IWRMS310M
HEW6625A
HEW10594059
SASCTIM508L
>VBTN450L
HEW105241A8
HEW105241A8
RIC841718
DELL3108092
>VRD1720
SAC1579B
HEW51579B
HEWCN0666N
NRD16660
HEWB6619A
HEWB6619A
DNSSCRH1000REM
HEWC3903A
EPS3004153
DLINV319
DXF914LS0D
EPST252XBCS
DELT1HD47M
DPSDB336B
CH18294
HEW31840AN
CNM63848002
BRTTN3398K
HEWB6720A
BRTTN318K
BRTTN431M
BRTTN431Y
CNM3028C001
CNM3023C001
CNM3022C001
BRTTN5000PF
BRTTN570
BRTTN500
CNM1828B001
BRTTN2278K
CNM6385B002
HEW3YM56AN
HEW3YM57AN
HEW3YM58AN
HEW3YM59AN
HEW3YM52AN
CNM7814AA003AA
HEWC1092A
CNM6391B002
HEW62455AN
CNM6388B002
CNM6388B002
BRTTN223C
CNM6387B002
BRTTN223Y
CNM6386B002
DELL31072
nc841724
HEW411823
PA1G32
RIC406042
RIC406046
RIC406046
RIC406475
RIC407656
BRC841420
BRC841357
BRC841357

```

```

def draw(lable, size):
    from matplotlib import font_manager as fm
    plt.figure(figsize=(12,18)) #resize the figure
    labels = lable #define the labels as Supplier
    sizes = size #define the size corresponds to
    colors = cm.rainbow(np.linspace(0.3, 1, len(labels)))#define the color for each s
    explode = (0,)*len(sizes) #make distance between sectors

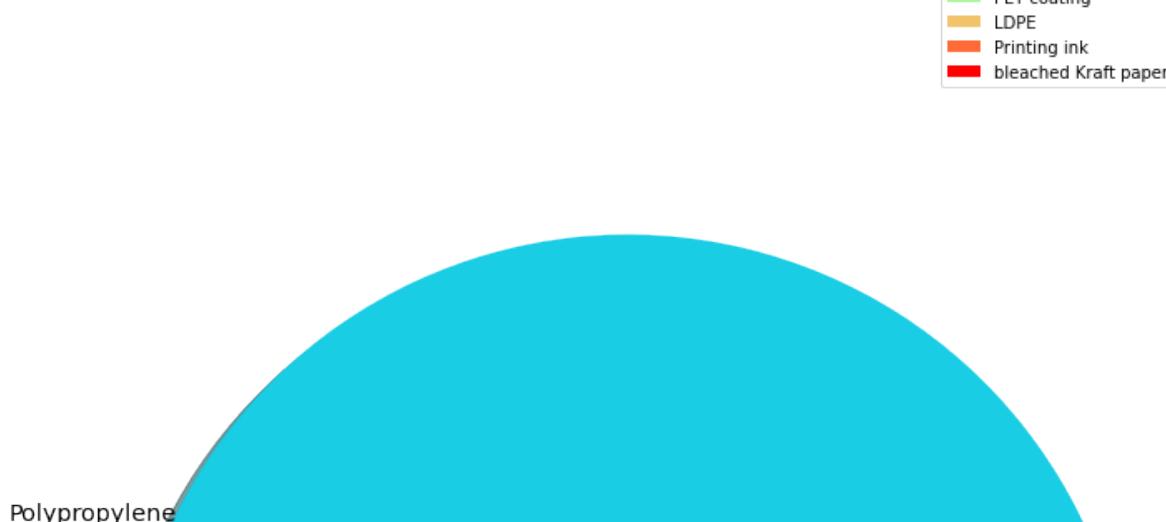
    patches, text1, text2 = plt.pie(sizes,
                                    explode=explode,
                                    labels=labels,
                                    colors=colors,
                                    labeldistance = 1,
                                    autopct = '%3.2f%%',
                                    shadow = True, #no shadow
                                    startangle =0, #counter-clockwise
                                    pctdistance = 0.6) #value distance from the center

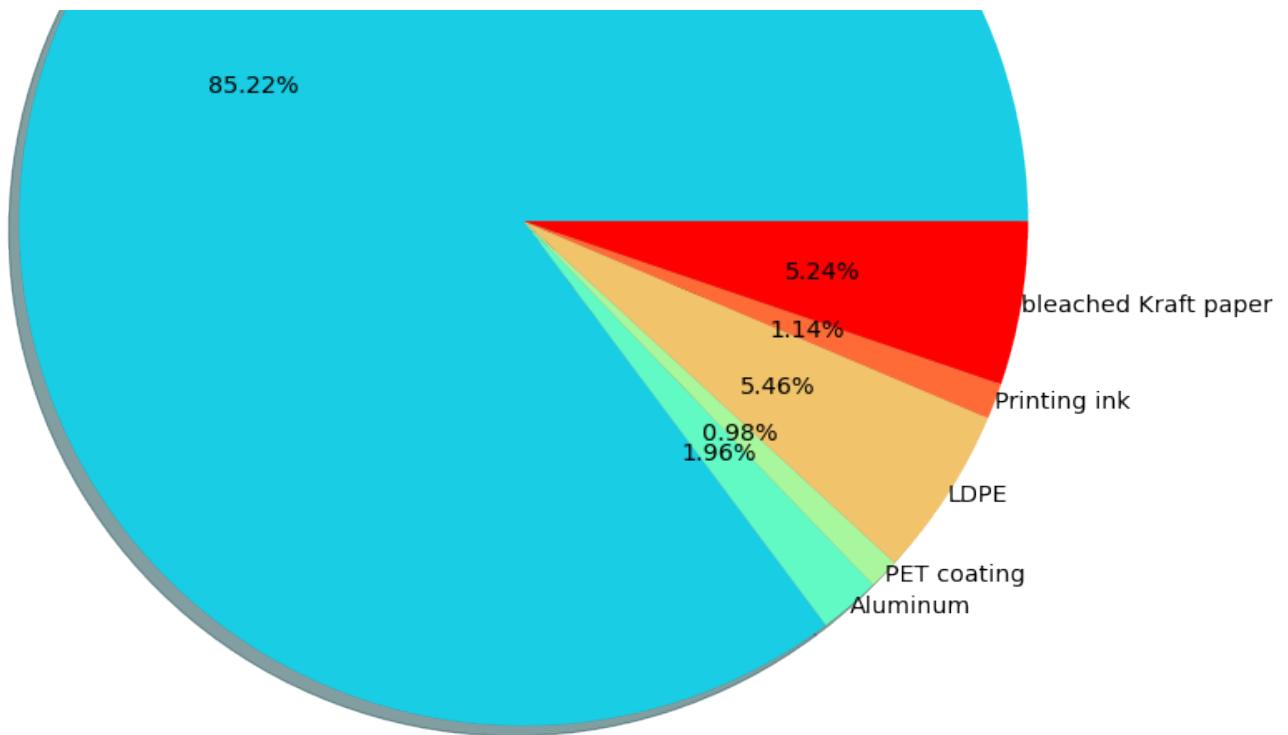
    proptease = fm.FontProperties()
    proptease.set_size('x-large')
    # font size include: 'xx-small','x-small','small','medium','large','x-large','xx-large'
    plt.setp(text1, fontproperties=proptease)
    plt.setp(text2, fontproperties=proptease)

    plt.axis('equal')
    plt.legend()
    plt.show()

HEWECE2338 #
coffee_comp = ['Polypropylene', 'Aluminum', 'PET coating', 'LDPE', 'Printing ink', 'bleached Kraft paper']
coffee_comp_size = [Polypropylene, Aluminum, PET, LDPE, Printing, paper]
draw(coffee_comp, coffee_comp_size)

```





Components for a single unit of plastic coffee pod

total: 2.3467 g

Shell total: 1.3909 g + 0.6090 g (ring) = 1.999 g [Polypropylene]

Lid total: 0.1912 g = Aluminum (24%) PET coating (12%) LDPE (50%) Printing ink (14%)

Filter Total: 0.1555 g (6.63%) = Kraft paper, bleached (79%) + LDPE (21%)

reference: <https://www.nature.com/articles/s41598-020-65058-1/tables/3>

Waste weight of the top 2 consumed coffee creamer pods university wide during year 2016-2021

```
print("total waste for the top two creams: ITD102042 + NES753032")
total2 = (counts.tolist()[4] + counts.tolist()[11]) * 2.3467
print("total weight: ", total2 , " g")

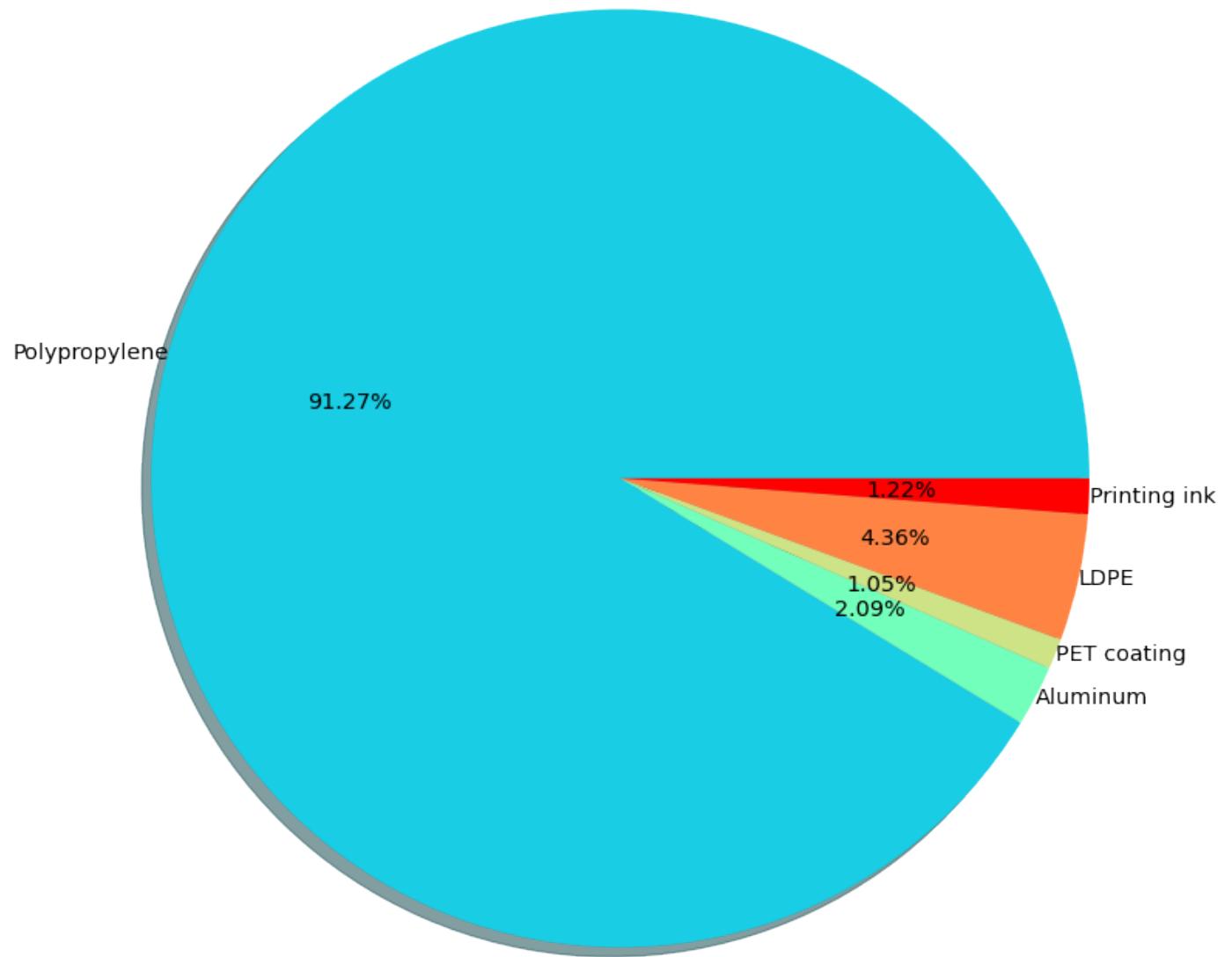
print("total weight of Polypropylene: ", 2.5749 / 3.0213 * total2, " g")
print("total weight of Aluminum:", 0.0591 / 3.0213 * total2, "g")
print("total weight of Printing ink: ", 0.0345/ 3.0213 * total2, " g")
print("total weight of PET coating: ", 0.0295/3.0213 * total2, " g")
print("total weight of LDPE: ",0.1231 / 3.0213 * total2, " g")

(Polypropylene,Aluminum,PET,LDPE,Printing) = (2.5749 / 3.0213 * total2,
                                              0.0591 / 3.0213 * total2,
                                              0.0295/3.0213 * total2,
                                              0.1231 / 3.0213 * total2,
                                              0.0345/ 3.0213 * total2)
```

```
total waste for the top two creams: ITD102042 + NES753032
total weight: 4477.5036 g
total weight of Polypropylene: 3815.9481083109918 g
total weight of Aluminum: 87.58496764968721 g
total weight of Printing ink: 51.12828060768544 g
total weight of PET coating: 43.71838486744116 g
total weight of LDPE: 182.4316331248138 g
```

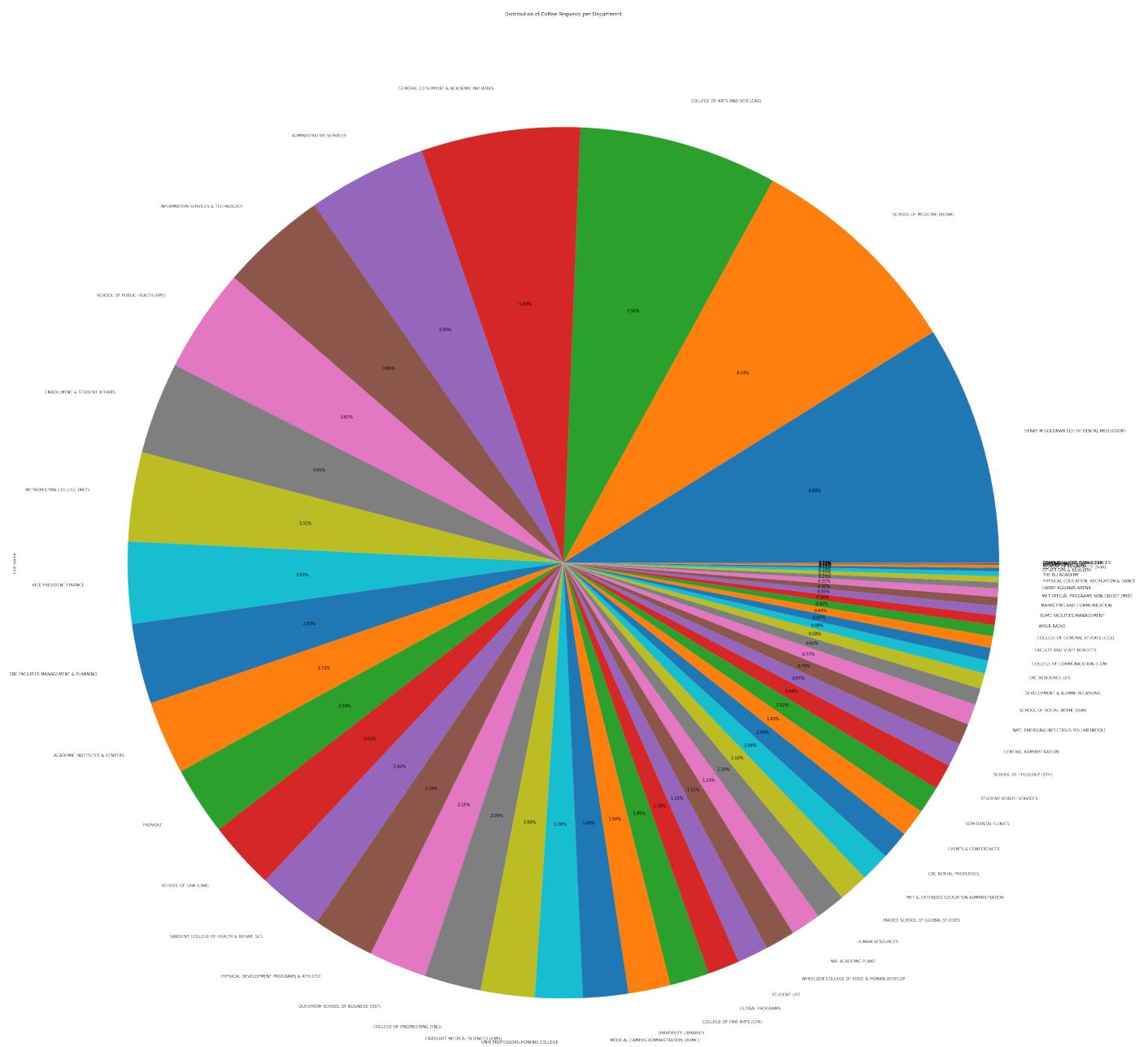
```
cream_comp = ['Polypropylene','Aluminum','PET coating','LDPE','Printing ink']
cream_comp_size = [Polypropylene,Aluminum,PET,LDPE,Printing]
draw(cream_comp,cream_comp_size)
```





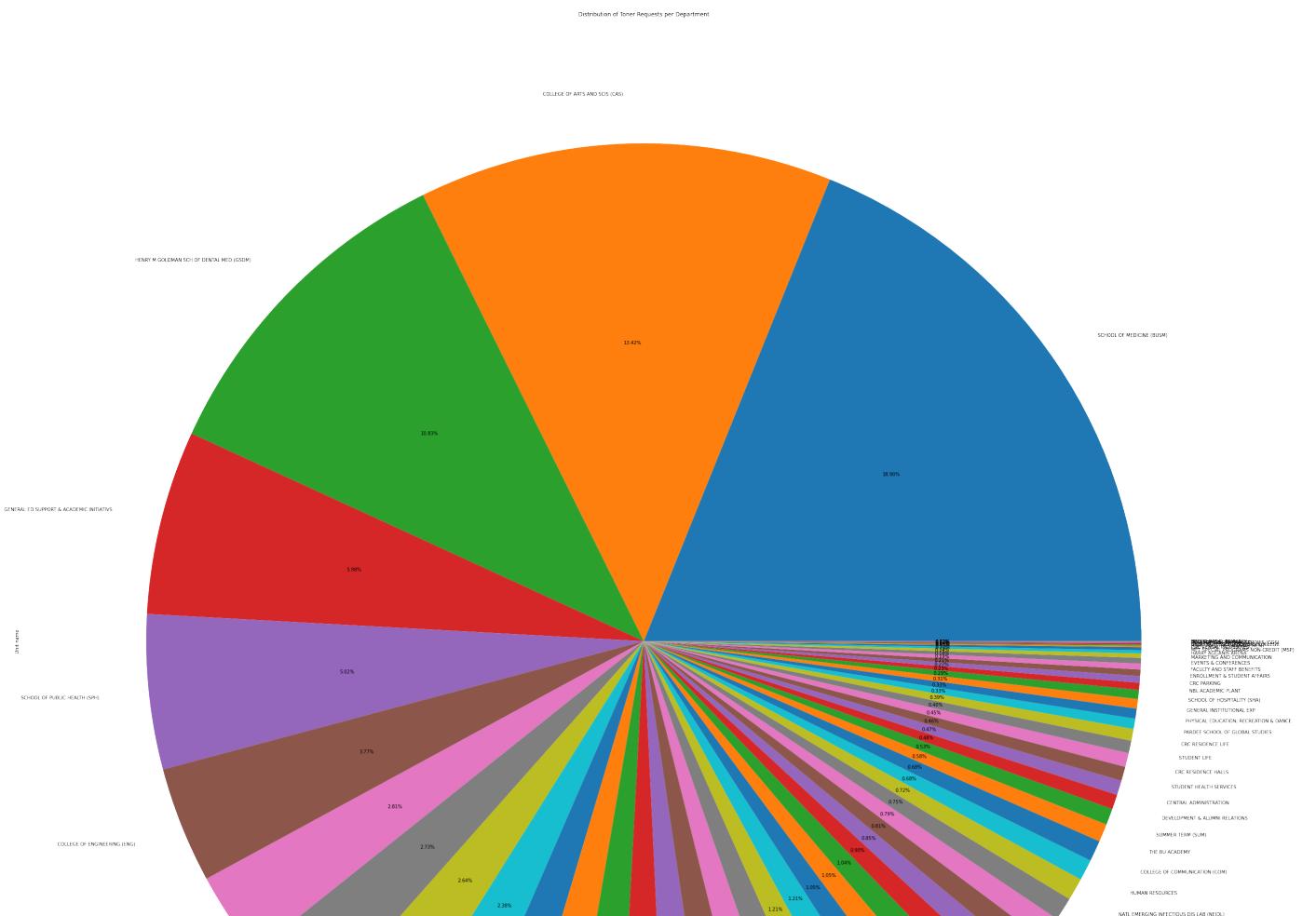
Total request number for coffee during 2016-2021, categorized by department and their proportions

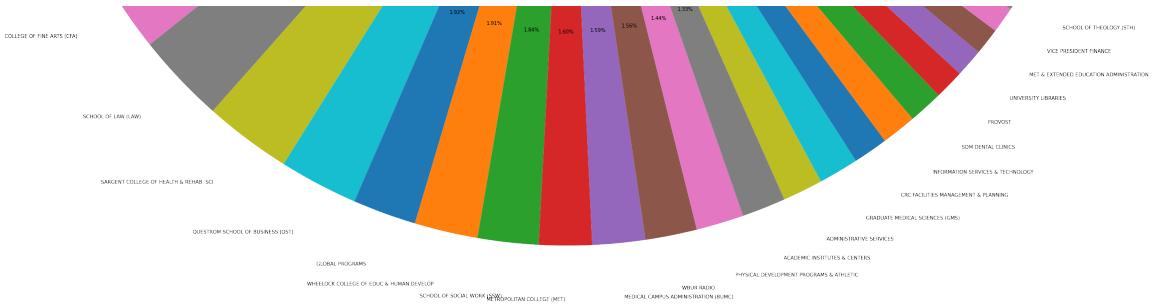
```
coffee_req1['Unit name'].value_counts().plot.pie(figsize=(48,72),title="Distribution of coffee units", autopct='%.1f%%', startangle=90)  
plt.gca().set_aspect('equal')
```



Total request number for toner during 2016-2021, categorized by department and their proportions

```
toner_req1['Unit name'].value_counts().plot.pie(figsize=(48,72),title="Distribution of Toner Requests per Department")
plt.gca().set_aspect('equal')
```





Top 10 consumed coffee product and its order interval during 2016 - 2021,
university wide

```
top10coffee=coffee_req1['upper part'].value_counts()[:10]._stat_axis.values.tolist()

def find_thecoffee_gap(name):
    for h in name:
        print('{0}'.format(h))
    a1=coffee_req1[coffee_req1['Product P/N (Vendor)'] == h]['PO Creation Date'].to
    # print(a1)
    a1 = sorted(a1)

    a11 = []
    for i in range(len(a1)-1):
        a11.append((a1[i+1]-a1[i]).days)
```

```

# print(a11)
arr_mean=(np.mean(a11))
arr_std=(np.std(a11))

print("The mean is : %f" % arr_mean)

print("The standard div is : %f" % arr_std)
plt.plot(a1[1:], a11, '-o')
plt.rcParams["figure.figsize"] = (40,30)
plt.xlabel('Date', fontsize=18)
plt.ylabel('Order interval (Days)', fontsize=16)
plt.show()

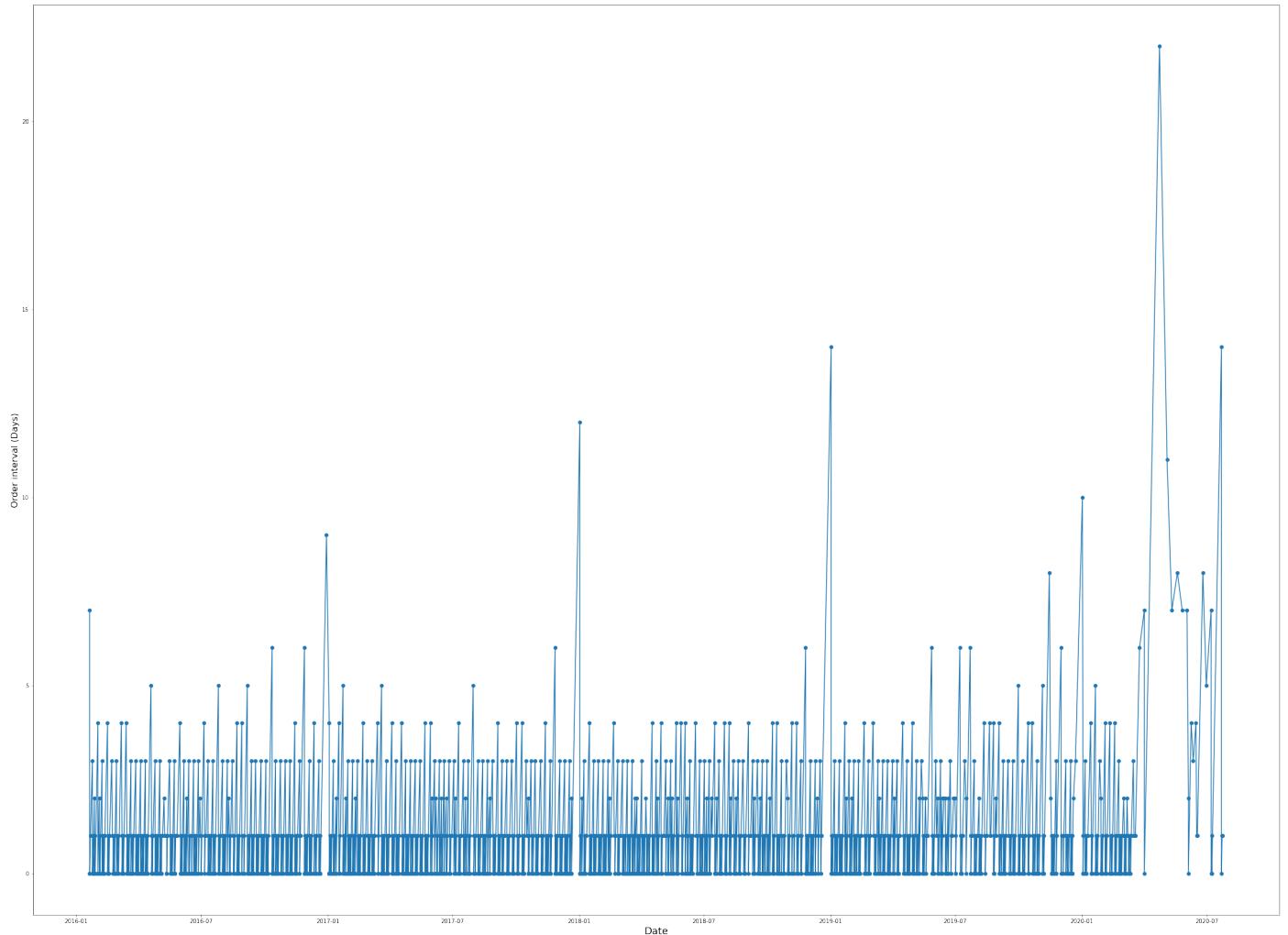
```

find_thecoffee_gap(top10coffee)

"GMT6520"

The mean is : 0.686462

The standard div is : 1.322605

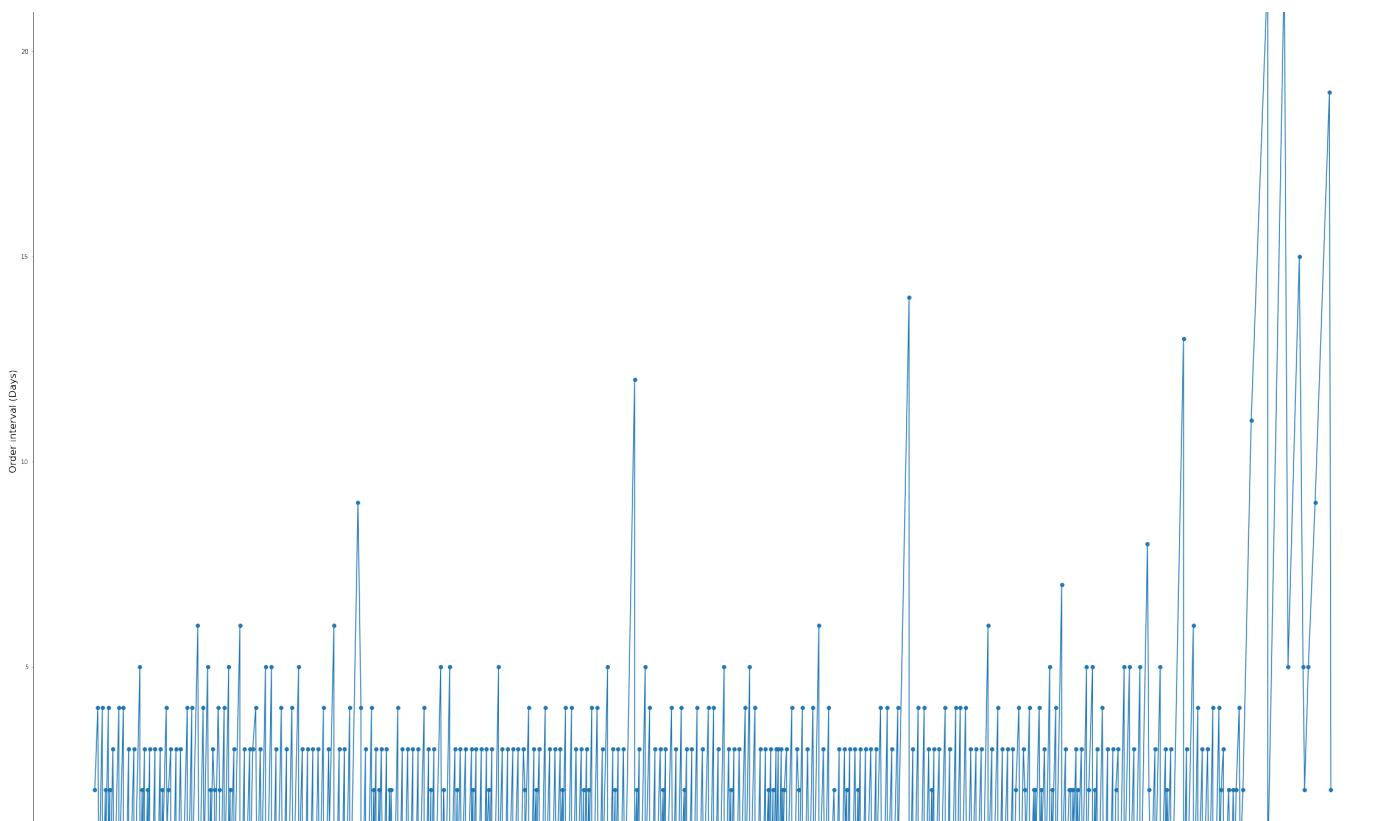


"GMT4061"

The mean is : 0.720665

The standard div is : 1.468255





Top 10 consumed toner product and its order interval during 2016 - 2021, university wide

```

    The standard div is : 1.750251
top10toner=toner_req1['upper part'].value_counts()[:10]._stat_axis.values.tolist()

def find_thetoner_gap(name):
    for h in name:
        print('{0}'.format(h))
        a1=toner_req1[toner_req1['Product P/N (Vendor)'] == h]['P0 Creation Date'].tol
        # print(a1)
        a1 = sorted(a1)
        a11 = []
        for i in range(len(a1)-1):
            a11.append((a1[i+1]-a1[i]).days)
        arr_mean=(np.mean(a11))
        arr_std=(np.std(a11))

        print("The mean is : %f" % arr_mean)

        print("The standard div is : %f" % arr_std)
        plt.plot(a1[1:], a11, '-o')
        plt.rcParams["figure.figsize"] = (40,30)
        plt.xlabel('Date', fontsize=18)
        plt.ylabel('Order interval (Days)', fontsize=16)
        plt.show()

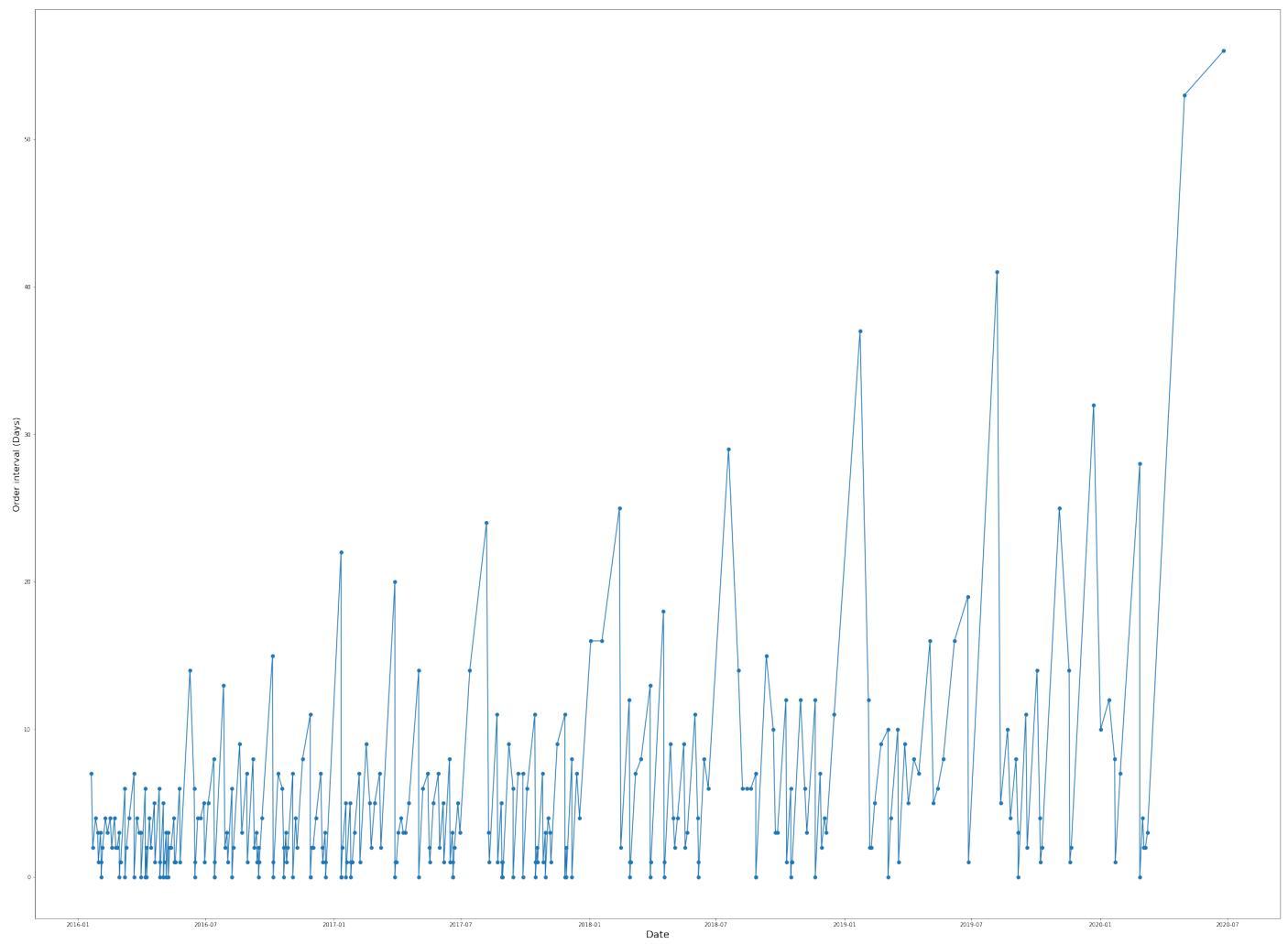
```

`find_thetoner_gap(top10toner)`

"HEWCE505A"

The mean is : 5.145570

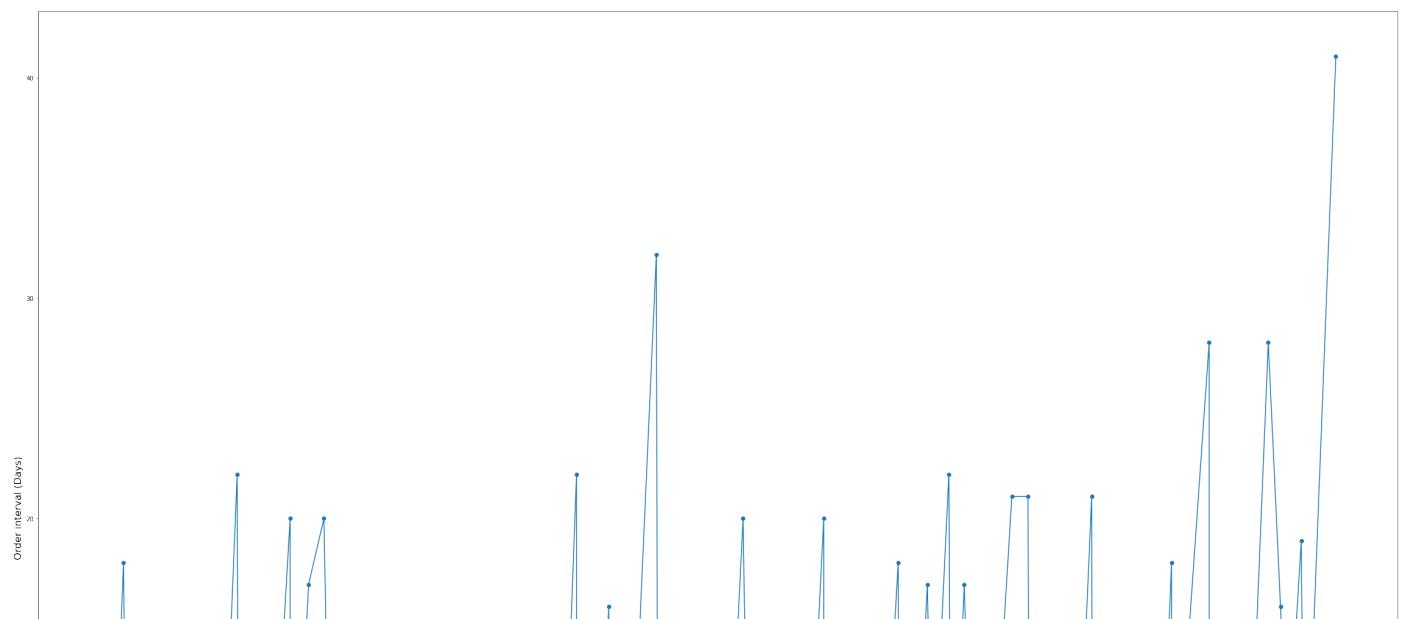
The standard div is : 7.125748

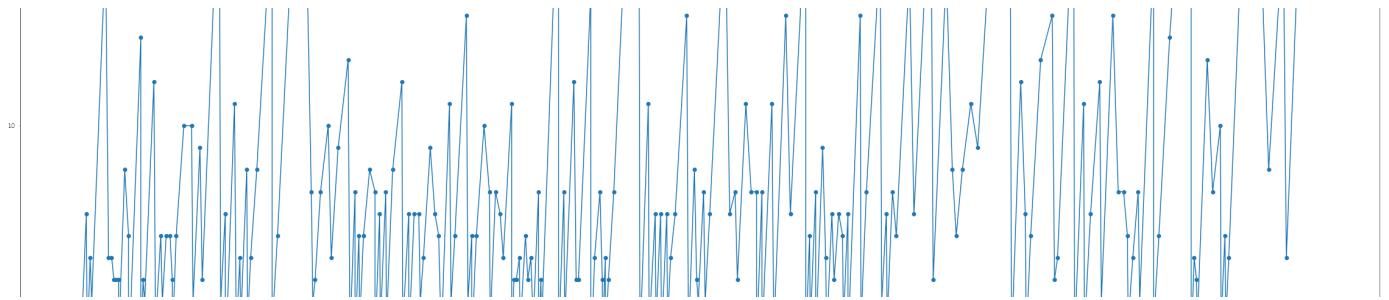


"HEWCN045AN"

The mean is : 5.672535

The standard div is : 6.151991





Top 5 consumed coffee product and its order interval during 2016 - 2021, categorized by top 5 most frequent ordered department

```

The standard div is : 6.262906
top10coffee=coffee_req1['upper part'].value_counts()[:5]._stat_axis.values.tolist()
top10toner=toner_req1['upper part'].value_counts()[:5]._stat_axis.values.tolist()

top10dep=coffee_req1['Unit name'].value_counts()[:5]._stat_axis.values.tolist()
top10dep2=toner_req1['Unit name'].value_counts()[:5]._stat_axis.values.tolist()

mark=['-o','.-','-v','-s','-x']

def find_thecoffee_gap2(name,dep):
    for h in name:
        print("{0}".format(h))
    for l in range(len(dep)):
        print("{0}".format(dep[l]))
        temp = coffee_req1[coffee_req1['Unit name'] == dep[l]]
        a1=temp[temp['Product P/N (Vendor)'] == h]['PO Creation Date'].tolist()
        # print(a1)
        a1 = sorted(a1)

        a11 = []
        for i in range(len(a1)-1):
            a11.append((a1[i+1]-a1[i]).days)
        # print(a11)

        arr_mean=(np.mean(a11))
        arr_std=(np.std(a11))

        print("The mean is : %f" % arr_mean)

        print("The standard div is : %f" % arr_std)
        plt.plot(a1[1:], a11, mark[l])
        plt.rcParams["figure.figsize"] = (40,30)
        plt.legend(dep,prop={'size': 20})
        plt.xlabel('Date', fontsize=18)
        plt.ylabel('Order interval (Days)', fontsize=16)

```

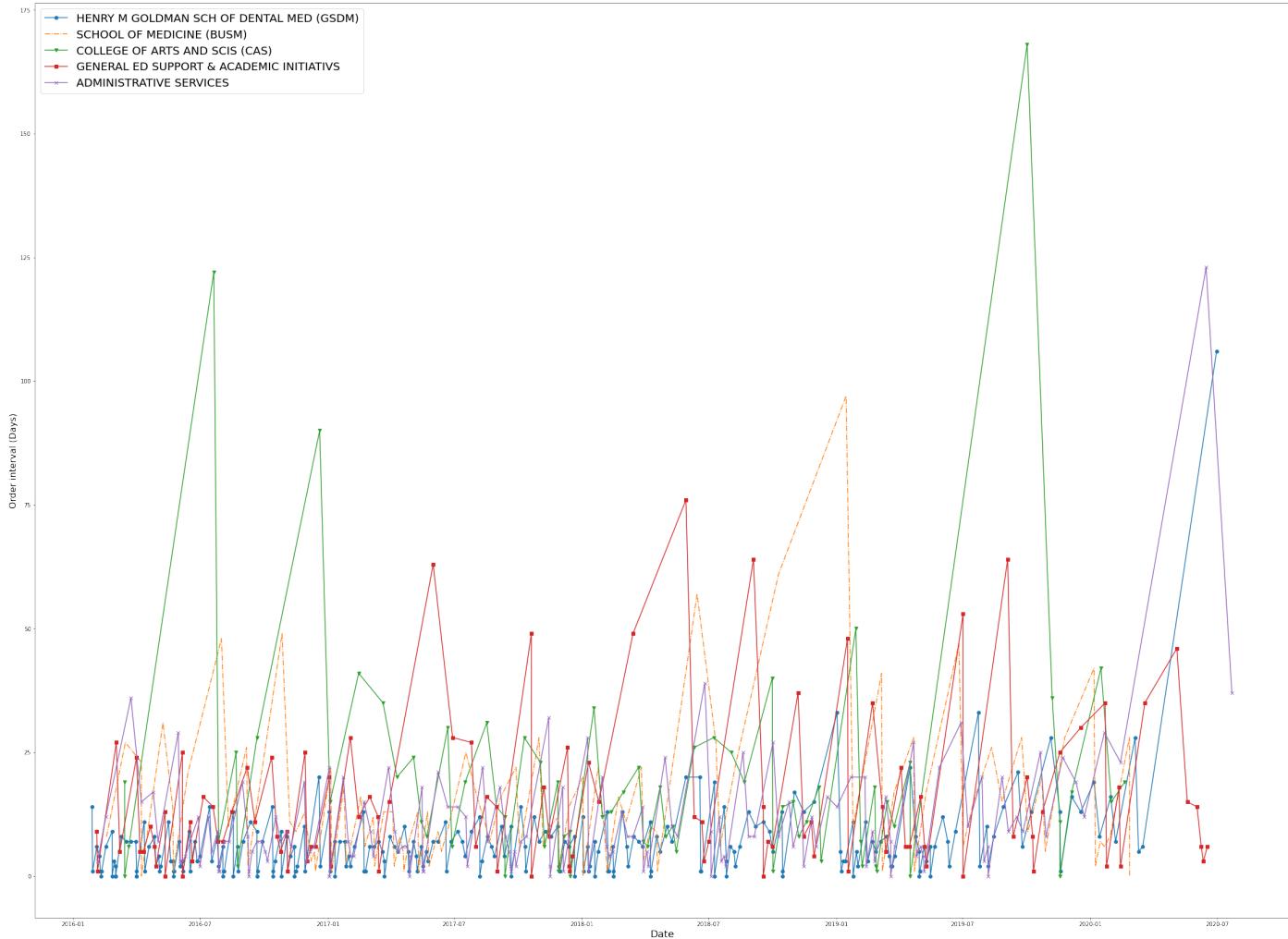
```

plt.show()

find_thecoffee_gap2(top10coffee,top10dep)

"GMT6520"
"HENRY M GOLDMAN SCH OF DENTAL MED (GSDM)"
The mean is : 6.388235
The standard div is : 8.479402
"SCHOOL OF MEDICINE (BUSM)"
The mean is : 14.770000
The standard div is : 15.306766
"COLLEGE OF ARTS AND SCIS (CAS)"
The mean is : 20.208333
The standard div is : 25.827385
"GENERAL ED SUPPORT & ACADEMIC INITIATIVS"
The mean is : 15.423077
The standard div is : 15.905788
"ADMINISTRATIVE SERVICES"
The mean is : 10.953020
The standard div is : 12.672613

```



```

"GMT4061"
"HENRY M GOLDMAN SCH OF DENTAL MED (GSDM)"
The mean is : 9.078788
The standard div is : 12.439423
"SCHOOL OF MEDICINE (BUSM)"
The mean is : 12.571429

```

```
The standard div is : 9.750304
"COLLEGE OF ARTS AND SCIS (CAS)"
The mean is : 12.944444
The standard div is : 15.865659
"GENERAL ED SUPPORT & ACADEMIC INITIATIVES"
The mean is : 7.597156
The standard div is : 6.700342
"ADMINISTRATIVE SERVICES"
The mean is : 18.592593
```

Top 5 consumed toner product and its order interval during 2016 - 2021,
categorized by top 5 most frequent ordered department

```
| | |
def find_thetoner_gap2(name,dep):
    for h in name:
        print("{0}".format(h))
        for l in range(len(dep)):
            print("{0}".format(dep[l]))
            temp = toner_req1[toner_req1['Unit name'] == dep[l]]
            a1=temp[temp['Product P/N (Vendor)'] == h]['PO Creation Date'].tolist()
            # print(a1)
            a1 = sorted(a1)

    a11 = []
    for i in range(len(a1)-1):
        a11.append((a1[i+1]-a1[i]).days)
    # print(a11)

    arr_mean=(np.mean(a11))
    arr_std=(np.std(a11))

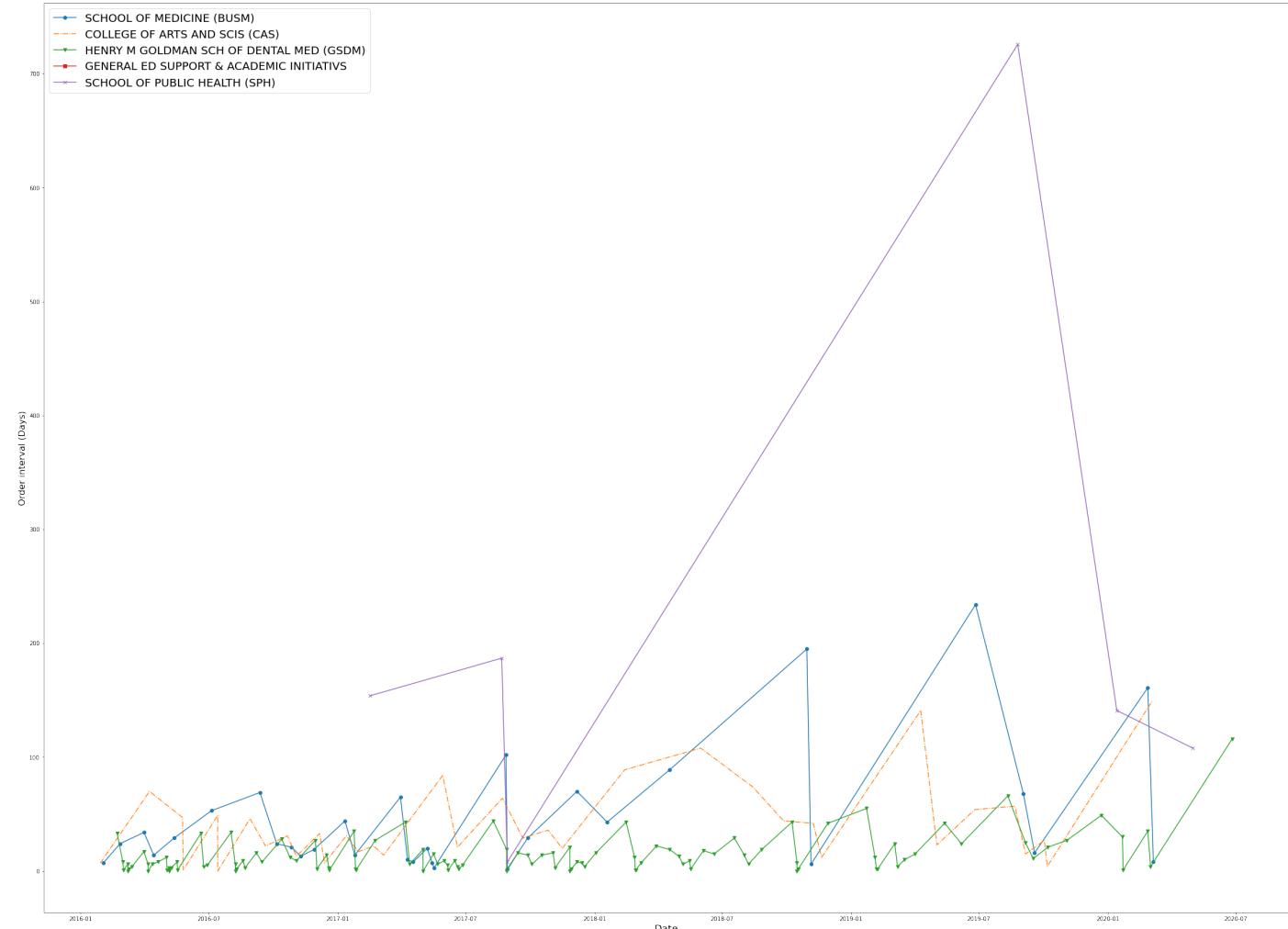
    print("The mean is : %f" % arr_mean)

    print("The standard div is : %f" % arr_std)
    plt.plot(a1[1:], a11, mark[l])
    plt.rcParams["figure.figsize"] = (40,30)
    plt.legend(dep,prop={'size': 20})
    plt.xlabel('Date', fontsize=18)
    plt.ylabel('Order interval (Days)', fontsize=16)
    plt.show()

find_thetoner_gap2(top10toner,top10dep2)

"HEWCE505A"
"SCOOTER OR XEROX TONER (DRUM)"
```

SCHOOL OF MEDICINE (BUSM)
The mean is : 46.906250
The standard div is : 55.370547
"COLLEGE OF ARTS AND SCIS (CAS)"
The mean is : 41.750000
The standard div is : 35.896746
"HENRY M GOLDMAN SCH OF DENTAL MED (GSDM)"
The mean is : 14.201754
The standard div is : 16.718086
"GENERAL ED SUPPORT & ACADEMIC INITIATIVES"
The mean is : nan
The standard div is : nan
"SCHOOL OF PUBLIC HEALTH (SPH)"
The mean is : 220.666667
The standard div is : 232.811559
/usr/local/lib/python3.7/dist-packages/numpy/core/fromnumeric.py:3373: RuntimeWarning: divide by zero encountered in double division
out=out, **kwargs)
/usr/local/lib/python3.7/dist-packages/numpy/core/_methods.py:170: RuntimeWarning: invalid value encountered in double division
ret = ret.dtype.type(ret / rcount)
/usr/local/lib/python3.7/dist-packages/numpy/core/_methods.py:234: RuntimeWarning: invalid value encountered in double division
keepdims=keepdims)
/usr/local/lib/python3.7/dist-packages/numpy/core/_methods.py:195: RuntimeWarning: invalid value encountered in double division
arrmean, rcount, out=arrmean, casting='unsafe', subok=False)
/usr/local/lib/python3.7/dist-packages/numpy/core/_methods.py:226: RuntimeWarning: invalid value encountered in double division
ret = ret.dtype.type(ret / rcount)



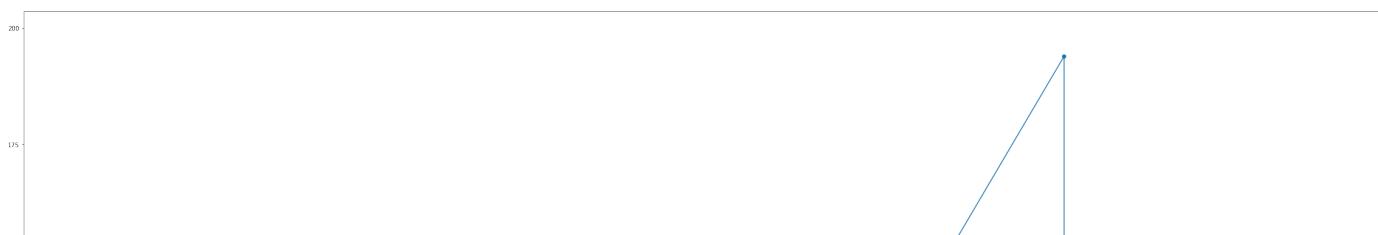
"HEWCN045AN"

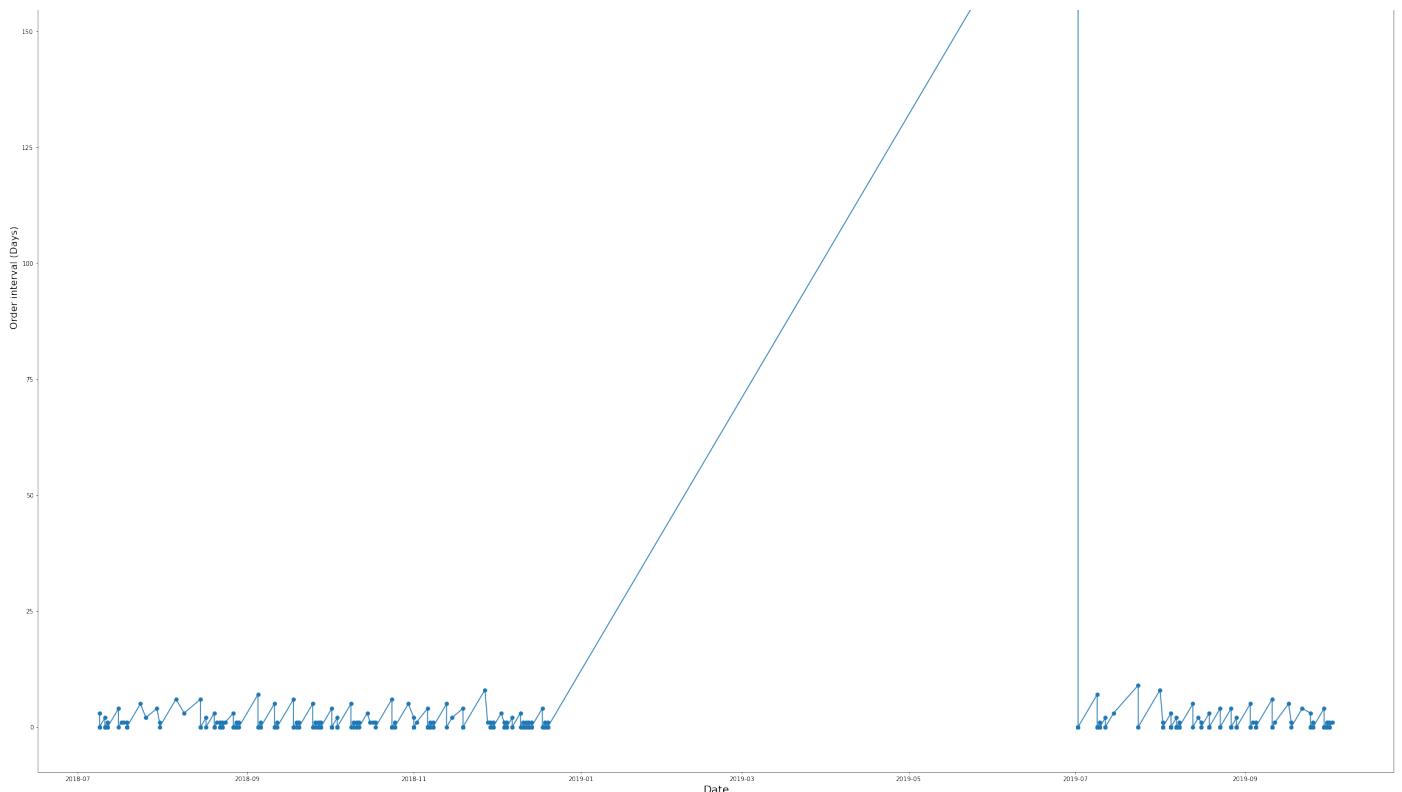
```
"SCHOOL OF MEDICINE (BUSM)"  
The mean is : 27.169492  
The standard div is : 28.006145  
"COLLEGE OF ARTS AND SCIS (CAS)"
```

Top 10 consumed paper product and its order interval during 2016 - 2021, university wide

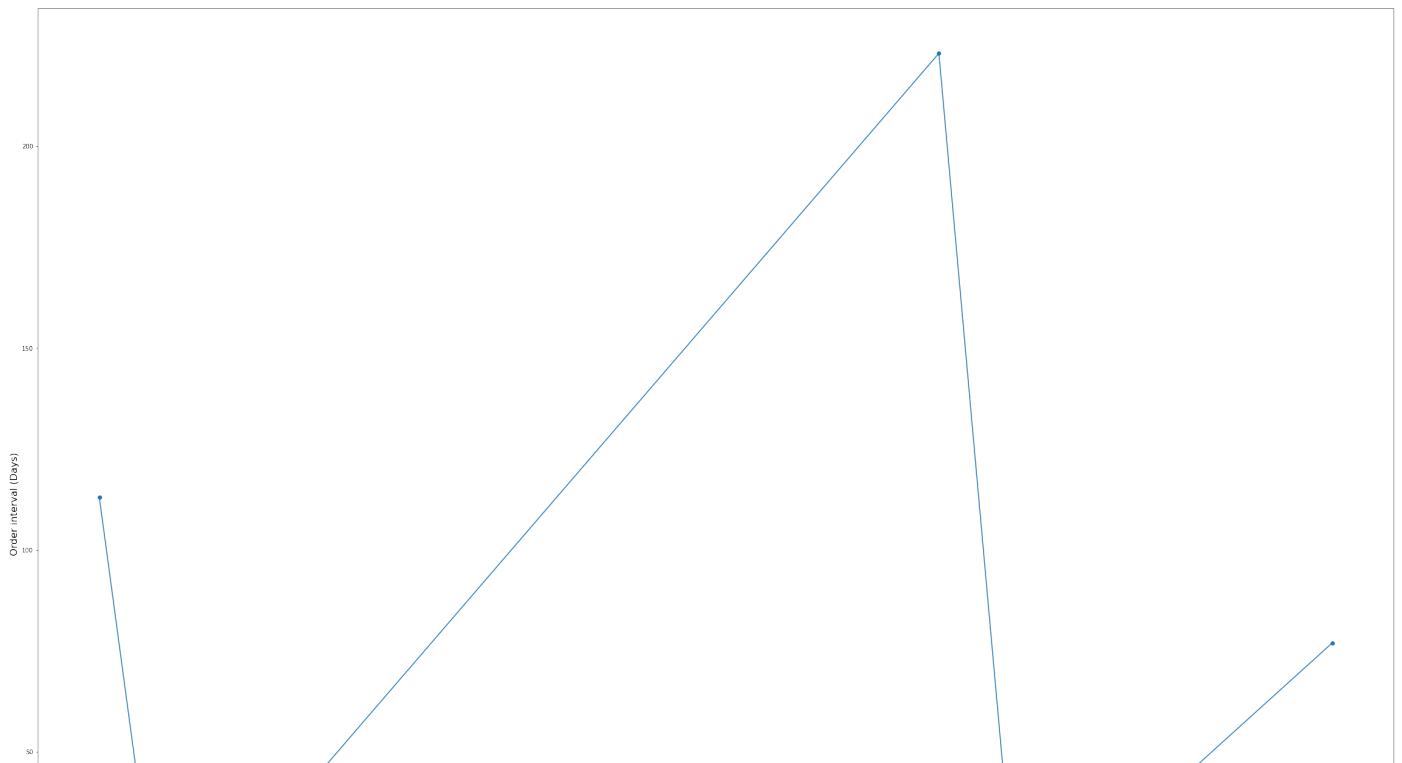
```
"GENERAL ED SUPPORT & ACADEMIC INITIATIVES"  
paper_name = paper_df['Description'].str.split('|', expand=True)[2].str.split(',',  
  
def find_thepaper_gap(name):  
    for h in name:  
        print('{0}'.format(h))  
    a1=paper_df[paper_df['Description'].str.contains(h)]['Requisitioning Date'].tol  
    print(type(a1))  
    a1 = sorted(a1)  
  
    a11 = []  
    for i in range(len(a1)-1):  
        a11.append((a1[i+1]-a1[i]).days)  
    print(set(a11))  
  
    arr_mean=(np.mean(a11))  
    arr_std=(np.std(a11))  
  
    print("The mean is : %f" % arr_mean)  
  
    print("The standard div is : %f" % arr_std)  
    plt.plot(a1[1:], a11, '-o')  
    plt.rcParams["figure.figsize"] = (40,30)  
    plt.xlabel('Date', fontsize=18)  
    plt.ylabel('Order interval (Days)', fontsize=16)  
    plt.show()  
  
find_thepaper_gap(paper_name[:-3])
```

```
"FIREWORX Colored Paper"  
<class 'list'>  
{0, 1, 2, 3, 4, 5, 6, 7, 8, 194, 9}  
The mean is : 1.115479  
The standard div is : 9.687446
```





```
"Copy Plus Copy Paper"  
<class 'list'>  
{0, 4, 11, 77, 16, 113, 20, 22, 223}  
The mean is : 54.000000  
The standard div is : 69.490207
```



Most consumed paper product and its order interval during 2016 - 2021,
categorized by top 5 most frequent ordered department

```

top5fc = paper_df[paper_df['fc name'] != 'no match to bus']['fc name'].value_counts()
top5dep = paper_df[paper_df['Unit name'] != 'no match to bus']['Unit name'].value_counts()

mark=['-o','.-','-v','-s','-x']

def find_thepaper_gap2(name, dep, sub):
    for h in name:
        print('{0}'.format(h))
    for l in range(len(dep)):
        print('{0}'.format(dep[l]))
        temp = paper_df[paper_df['Unit name'] == dep[l]]
        if sub:
            temp = paper_df[paper_df['fc name'] == dep[l]]

    a1=temp[temp['Description'].str.contains(h)]['Requisitioning Date'].tolist()
    # print(type(a1))
    a1 = sorted(a1)

    a11 = []
    for i in range(len(a1)-1):
        a11.append((a1[i+1]-a1[i]).days)
    print(set(a11))

    arr_mean=(np.mean(a11))
    arr_std=(np.std(a11))

    print("The mean is : %f" % arr_mean)

    print("The standard div is : %f" % arr_std)
    plt.plot(a1[1:], a11, mark[l])
    plt.rcParams["figure.figsize"] = (40,30)
    plt.legend(dep,prop={'size': 20})
    plt.xlabel('Date', fontsize=18)
    plt.ylabel('Order interval (Days)', fontsize=16)
    plt.show()

find_thepaper_gap2(paper_name[:1], top5dep, False)
find_thepaper_gap2(paper_name[:1], top5fc, True)

```

```

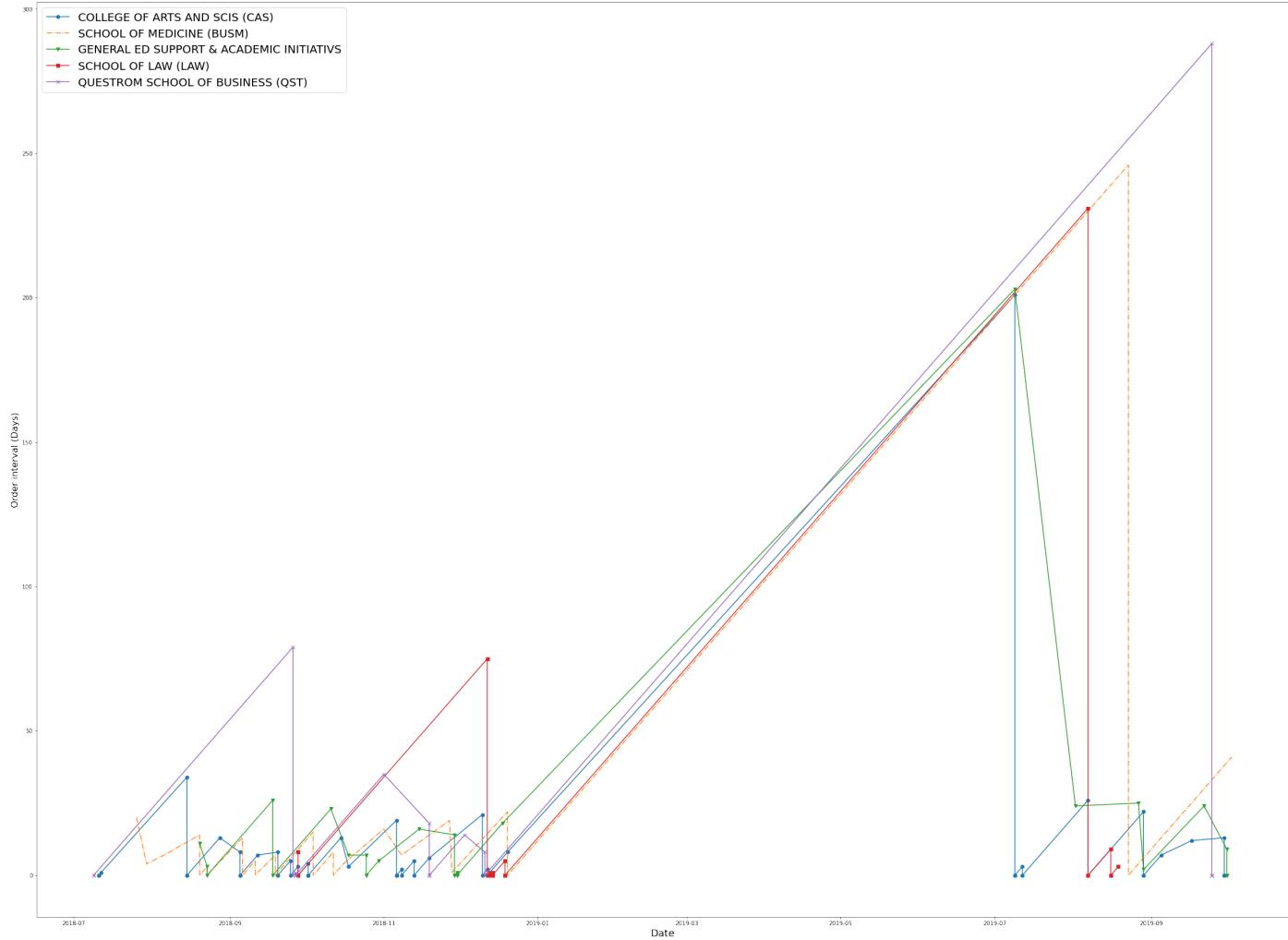
"FIREWORX Colored Paper"
"COLLEGE OF ARTS AND SCIS (CAS)"
{0, 1, 34, 3, 4, 5, 2, 7, 8, 6, 201, 12, 13, 19, 21, 22, 26}
The mean is : 5.309524
The standard div is : 22.421809
"SCHOOL OF MEDICINE (BUSM)"
{0, 1, 3, 4, 5, 7, 8, 41, 13, 14, 15, 16, 19, 20, 22, 246}

```

```

The mean is : 12.611111
The standard div is : 40.429141
"GENERAL ED SUPPORT & ACADEMIC INITIATIVES"
{0, 1, 2, 3, 5, 7, 9, 11, 203, 14, 16, 18, 23, 24, 25, 26}
The mean is : 13.062500
The standard div is : 35.287159
"SCHOOL OF LAW (LAW)"
{0, 1, 3, 5, 231, 8, 9, 75}
The mean is : 11.482759
The standard div is : 43.685096
"QUESTROM SCHOOL OF BUSINESS (QST)"
{0, 1, 288, 35, 8, 14, 79, 18}
The mean is : 17.720000
The standard div is : 57.669763

```



```

"FIREWORX Colored Paper"
"OFFICE OF THE UNIV CHAPLAIN"
{0, 5, 9, 14, 16, 49, 23, 26, 29, 222}
The mean is : 21.421053
The standard div is : 49.079223
"QST Copy Center"
{0, 288, 35, 8, 14, 80, 18}
The mean is : 19.260870
The standard div is : 59.934905
"CAS PHYSICS"
{0, 5, 8, 41, 204, 80, 29}
The mean is : 21.588235

```

```
the mean is : 21.500255
The standard div is : 50.119925
"CAS ECONOMICS"
{0}
The mean is : 0.000000
The standard div is : 0.000000
"ATH WOMEN'S SOCCER"
{0, 27, 7}
The mean is : 2.615385
The standard div is : 7.280516
```

