

Analysis of Superstore Data

The goal of this analysis is to identify areas that can be improved upon to enhance profitability.

In [116...

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('ggplot')
import os
```

In [117...

```
orders = pd.read_excel('Sample - Superstore.xls')
```

In [118...

```
orders.shape
```

Out[118...

(9994, 21)

In [119...

```
orders.head()
```

Out[119...

| | Row ID | Order ID | Order Date | Ship Date | Ship Mode | Customer ID | Customer Name | Segment | Country | City | ... | F |
|---|--------|----------------|------------|------------|----------------|-------------|-----------------|-----------|---------------|-----------------|-----|---|
| 0 | 1 | CA-2016-152156 | 2016-11-08 | 2016-11-11 | Second Class | CG-12520 | Claire Gute | Consumer | United States | Henderson | ... | 4 |
| 1 | 2 | CA-2016-152156 | 2016-11-08 | 2016-11-11 | Second Class | CG-12520 | Claire Gute | Consumer | United States | Henderson | ... | 4 |
| 2 | 3 | CA-2016-138688 | 2016-06-12 | 2016-06-16 | Second Class | DV-13045 | Darrin Van Huff | Corporate | United States | Los Angeles | ... | 5 |
| 3 | 4 | US-2015-108966 | 2015-10-11 | 2015-10-18 | Standard Class | SO-20335 | Sean O'Donnell | Consumer | United States | Fort Lauderdale | ... | 3 |
| 4 | 5 | US-2015-108966 | 2015-10-11 | 2015-10-18 | Standard Class | SO-20335 | Sean O'Donnell | Consumer | United States | Fort Lauderdale | ... | 3 |

5 rows × 21 columns



In [120...

```
orders.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Row ID                9994 non-null  int64
1   Order ID              9994 non-null  object
2   Order Date            9994 non-null  datetime64[ns]
3   Ship Date             9994 non-null  datetime64[ns]
4   Ship Mode             9994 non-null  object
5   Customer ID           9994 non-null  object
6   Customer Name         9994 non-null  object
7   Segment               9994 non-null  object
8   Country               9994 non-null  object
9   City                  9994 non-null  object
10  State                 9994 non-null  object
11  Postal Code           9994 non-null  int64
12  Region                9994 non-null  object
13  Product ID            9994 non-null  object
14  Category              9994 non-null  object
15  Sub-Category          9994 non-null  object
16  Product Name          9994 non-null  object
17  Sales                 9994 non-null  float64
18  Quantity              9994 non-null  int64
19  Discount              9994 non-null  float64
20  Profit                9994 non-null  float64
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB
```

In [121...

```
orders.columns
```

Out[121...

```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
      'Customer ID', 'Customer Name', 'Segment', 'Country', 'City', 'State',
      'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category',
      'Product Name', 'Sales', 'Quantity', 'Discount', 'Profit'],
      dtype='object')
```

In [122...

```
orders.nunique()
```

Out[122...

```
Row ID      9994
Order ID    5009
Order Date  1237
Ship Date   1334
Ship Mode     4
Customer ID  793
Customer Name 793
Segment      3
Country      1
City        531
State        49
Postal Code  631
Region       4
Product ID  1862
Category     3
Sub-Category 17
Product Name 1850
Sales       6144
Quantity    14
Discount    12
Profit      7545
dtype: int64
```

In [184...

```
for col in orders:
    print(orders[col].unique())
```

```
['CA-2016-152156' 'CA-2016-138688' 'US-2015-108966' ... 'CA-2014-110422'
'CA-2017-121258' 'CA-2017-119914']
['2016-11-08T00:00:00.000000000' '2016-06-12T00:00:00.000000000'
'2015-10-11T00:00:00.000000000' ... '2016-06-03T00:00:00.000000000'
'2015-04-12T00:00:00.000000000' '2014-01-21T00:00:00.000000000']
['CG-12520' 'DV-13045' 'SO-20335' 'BH-11710' 'AA-10480' 'IM-15070'
'HP-14815' 'PK-19075' 'AG-10270' 'ZD-21925' 'KB-16585' 'SF-20065'
'EB-13870' 'EH-13945' 'TB-21520' 'MA-17560' 'GH-14485' 'SN-20710'
'LC-16930' 'RA-19885' 'ES-14080' 'ON-18715' 'PO-18865' 'LH-16900'
'DP-13000' 'JM-15265' 'TB-21055' 'KM-16720' 'PS-18970' 'BS-11590'
'KD-16270' 'HM-14980' 'JE-15745' 'KB-16600' 'SC-20770' 'DN-13690'
'JC-16105' 'CS-12400' 'PG-18895' 'GM-14455' 'JS-15685' 'KB-16315'
'RB-19705' 'PN-18775' 'KD-16345' 'ER-13855' 'RB-19465' 'GZ-14470'
'LC-16870' 'JM-15250' 'PA-19060' 'CV-12805' 'CL-12565' 'RC-19960'
'DK-13090' 'GG-14650' 'SC-20725' 'AD-10180' 'PF-19165' 'TS-21610'
'LS-16975' 'DW-13585' 'LC-16885' 'JD-15895' 'SH-19975' 'SG-20080'
'HA-14920' 'MG-17680' 'JE-16165' 'TW-21025' 'SP-20650' 'NK-18490'
'DB-13060' 'NP-18670' 'TT-21070' 'EM-13960' 'RD-19900' 'MJ-17740'
'BM-11140' 'CS-12130' 'JB-15400' 'SJ-20500' 'JK-15640' 'DK-13150'
'RM-19675' 'SK-19990' 'FM-14290' 'AM-10360' 'MP-17470' 'MZ-17515'
'CB-12025' 'VM-21685' 'FH-14365' 'MB-17305' 'BS-11755' 'LC-17140'
'HK-14890' 'LE-16810' 'JH-15985' 'MS-17980' 'VW-21775' 'JH-15910'
'JB-15925' 'DS-13180' 'VD-21670' 'EA-14035' 'DB-13120' 'KL-16645'
'DW-13480' 'LH-17155' 'KC-16540' 'DL-13315' 'DR-12880' 'CC-12670'
'DL-13600' 'SB-20290' 'RC-19825' 'AH-10210' 'CB-12535' 'CA-12310'
'KH-16690' 'BB-10990' 'AG-10495' 'JO-15280' 'AH-10195' 'NZ-18565'
'KL-16555' 'AS-10225' 'CR-12625' 'SH-20395' 'BP-11185' 'TS-21205'
'AG-10525' 'SP-20860' 'NM-18445' 'FA-14230' 'GK-14620' 'DJ-13510'
'PO-18850' 'JL-15850' 'DB-13615' 'AC-10420' 'CC-12550' 'TD-20995'
'AB-10060' 'JL-15505' 'VB-21745' 'KW-16435' 'JD-16060' 'MK-17905'
'GT-14755' 'AG-10900' 'MM-18280' 'AR-10405' 'RA-19915' 'AS-10285'
'LA-16780' 'DO-13435' 'DK-13225' 'NG-18430' 'MV-18190' 'JG-15115'
'BP-11095' 'VP-21730' 'SS-20140' 'AG-10675' 'LF-17185' 'RF-19840'
'KH-16510' 'KC-16675' 'CJ-12010' 'PB-19150' 'MP-17965' 'NF-18385'
'SD-20485' 'KH-16630' 'RB-19795' 'MK-18160' 'PO-19180' 'BB-11545'
'TB-21595' 'RB-19360' 'EB-13705' 'SC-20095' 'TN-21040' 'JS-15940'
'MH-17785' 'JP-15520' 'JE-15475' 'JG-15805' 'XP-21865' 'EM-14065'
'MT-18070' 'SA-20830' 'CW-11905' 'AJ-10960' 'SS-20590' 'RO-19780'
'MD-17350' 'MY-17380' 'CM-12385' 'LS-17245' 'BN-11515' 'DB-13210'
'MC-17605' 'BD-11605' 'PH-18790' 'MG-18145' 'KB-16240' 'JC-15340'
'RL-19615' 'AA-10375' 'EP-13915' 'DK-12985' 'BD-11500' 'LM-17065'
'AS-10135' 'BD-11320' 'GT-14710' 'AJ-10945' 'OT-18730' 'LP-17080'
'CA-12775' 'JF-15490' 'FP-14320' 'EB-13840' 'JF-15415' 'SF-20200'
'TG-21640' 'CS-11950' 'CC-12145' 'DV-13465' 'BD-11725' 'ZC-21910'
'MS-17830' 'LR-16915' 'TP-21130' 'CK-12205' 'AS-10240' 'AR-10510'
'NB-18655' 'GD-14590' 'CK-12595' 'NG-18355' 'CA-12265' 'SF-20965'
'MO-17800' 'AT-10735' 'FM-14380' 'DJ-13420' 'ME-17725' 'JD-16150'
'JL-15835' 'SC-20305' 'CC-12430' 'AR-10825' 'SR-20740' 'CR-12730'
'EH-14125' 'SP-20545' 'TH-21235' 'RP-19390' 'RB-19570' 'CD-11980'
'DJ-13630' 'GT-14635' 'MC-17845' 'RA-19285' 'NP-18325' 'AB-10165'
'JO-15550' 'JK-15370' 'BN-11470' 'DP-13165' 'TH-21550' 'AP-10915'
'RS-19765' 'SV-20365' 'CK-12325' 'RD-19810' 'MR-17545' 'SC-20695'
'JF-15355' 'EG-13900' 'DS-13030' 'PO-19195' 'SS-20875' 'PB-19105'
'RF-19735' 'YC-21895' 'DC-13285' 'CP-12340' 'BF-11020' 'LH-17020'
'CS-12250' 'AJ-10795' 'BV-11245' 'DL-12865' 'BM-11785' 'LT-17110'
'JK-15730' 'ES-14020' 'RH-19495' 'CD-11920' 'HW-14935' 'MC-18130'
'GM-14440' 'PJ-19015' 'BW-11110' 'TR-21325' 'PG-18820' 'JL-15175'
'BM-11650' 'EM-14095' 'AF-10885' 'GA-14725' 'CK-12760' 'DP-13105'
'BK-11260' 'SJ-20125' 'CM-12445' 'AJ-10780' 'LS-16945' 'GP-14740'
'PK-18910' 'SM-20005' 'AG-10765' 'PM-19135' 'LL-16840' 'JS-15595']
```

| | | | | | |
|------------|------------|------------|------------|------------|------------|
| 'EL-13735' | 'PC-18745' | 'HL-15040' | 'MS-17365' | 'GB-14530' | 'JR-16210' |
| 'BE-11335' | 'SC-20050' | 'RW-19630' | 'SE-20110' | 'AH-10075' | 'JM-15535' |
| 'JJ-15760' | 'RK-19300' | 'CG-12040' | 'RP-19270' | 'KC-16255' | 'KH-16360' |
| 'GH-14665' | 'SW-20275' | 'JA-15970' | 'DL-12925' | 'LW-16990' | 'TB-21190' |
| 'BS-11800' | 'RW-19690' | 'TZ-21580' | 'AS-10630' | 'TS-21340' | 'SL-20155' |
| 'MW-18235' | 'RD-19585' | 'RA-19945' | 'MT-17815' | 'VG-21790' | 'JS-15880' |
| 'KM-16225' | 'HR-14770' | 'DE-13255' | 'AG-10390' | 'JJ-15445' | 'JH-15430' |
| 'RD-19660' | 'MO-17500' | 'NS-18640' | 'DG-13300' | 'NF-18595' | 'MG-17650' |
| 'TS-21160' | 'BD-11620' | 'CM-12160' | 'SN-20560' | 'EH-14005' | 'FO-14305' |
| 'MS-17710' | 'CC-12100' | 'DW-13540' | 'BT-11395' | 'CY-12745' | 'BT-11485' |
| 'PS-19045' | 'PV-18985' | 'NM-18520' | 'DL-13495' | 'CS-12355' | 'FH-14275' |
| 'NC-18340' | 'AA-10315' | 'LT-16765' | 'AP-10720' | 'PM-18940' | 'AT-10435' |
| 'CA-12055' | 'HR-14830' | 'BT-11530' | 'LH-16750' | 'SW-20755' | 'SP-20620' |
| 'BF-11170' | 'KT-16480' | 'BG-11695' | 'GM-14680' | 'EJ-14155' | 'NP-18700' |
| 'MH-18115' | 'JR-15700' | 'SM-20950' | 'CC-12220' | 'PF-19225' | 'DC-12850' |
| 'BD-11770' | 'GM-14500' | 'TB-21355' | 'JH-16180' | 'EB-13975' | 'QJ-19255' |
| 'TC-21535' | 'CS-12460' | 'HG-14965' | 'LW-16825' | 'MC-17575' | 'LP-17095' |
| 'EB-14170' | 'GZ-14545' | 'CP-12085' | 'FG-14260' | 'LD-17005' | 'AB-10255' |
| 'MN-17935' | 'JR-15670' | 'JF-15190' | 'CM-12115' | 'AS-10045' | 'BP-11290' |
| 'ND-18370' | 'LB-16735' | 'KT-16465' | 'HM-14860' | 'AB-10600' | 'SZ-20035' |
| 'MG-17890' | 'JK-16120' | 'PP-18955' | 'YS-21880' | 'KM-16375' | 'AB-10105' |
| 'HA-14905' | 'BT-11305' | 'SV-20815' | 'RW-19540' | 'DK-12835' | 'ST-20530' |
| 'MM-17920' | 'PW-19030' | 'SC-20440' | 'TS-21085' | 'MC-17425' | 'ME-17320' |
| 'NH-18610' | 'MB-18085' | 'KD-16495' | 'KN-16390' | 'NP-18685' | 'CS-12505' |
| 'MS-17770' | 'CM-12655' | 'Co-12640' | 'TS-21370' | 'JW-15220' | 'JD-15790' |
| 'PC-19000' | 'AR-10540' | 'AI-10855' | 'TB-21400' | 'PL-18925' | 'GH-14425' |
| 'MP-18175' | 'JM-15655' | 'CL-11890' | 'DB-13270' | 'IG-15085' | 'BO-11425' |
| 'AB-10150' | 'JW-16075' | 'EB-13750' | 'SG-20470' | 'CM-12190' | 'AW-10840' |
| 'MC-18100' | 'TT-21460' | 'VG-21805' | 'MY-18295' | 'RD-19480' | 'DP-13390' |
| 'ML-17395' | 'JC-15385' | 'JG-15160' | 'MC-17275' | 'NW-18400' | 'TB-21280' |
| 'BS-11380' | 'HH-15010' | 'CD-12280' | 'TH-21100' | 'MM-18055' | 'NS-18505' |
| 'RB-19645' | 'SW-20455' | 'EB-13930' | 'PS-18760' | 'HF-14995' | 'HZ-14950' |
| 'CD-12790' | 'JK-15205' | 'FM-14215' | 'ED-13885' | 'DA-13450' | 'JW-15955' |
| 'RM-19375' | 'ML-17755' | 'CC-12685' | 'JE-15610' | 'RP-19855' | 'TB-21175' |
| 'BE-11455' | 'JF-15565' | 'PB-19210' | 'BT-11680' | 'JL-15235' | 'CH-12070' |
| 'ND-18460' | 'BF-10975' | 'KH-16330' | 'GW-14605' | 'NC-18625' | 'ME-18010' |
| 'BP-11230' | 'JC-15775' | 'AS-10090' | 'AC-10450' | 'MD-17860' | 'DB-13660' |
| 'EH-13990' | 'EH-13765' | 'SC-20230' | 'JE-15715' | 'AC-10615' | 'JD-16015' |
| 'CB-12415' | 'JS-16030' | 'LW-17215' | 'SC-20800' | 'AM-10705' | 'RH-19510' |
| 'CT-11995' | 'MC-17590' | 'CC-12610' | 'KA-16525' | 'TC-20980' | 'BF-11080' |
| 'MM-17260' | 'AH-10120' | 'BW-11200' | 'SW-20245' | 'BS-11665' | 'RF-19345' |
| 'WB-21850' | 'TB-21625' | 'AF-10870' | 'RB-19435' | 'KF-16285' | 'JH-15820' |
| 'IL-15100' | 'PB-18805' | 'RH-19600' | 'AW-10930' | 'ML-17410' | 'DB-13555' |
| 'MH-17620' | 'DK-13375' | 'BT-11440' | 'DB-13405' | 'TG-21310' | 'BF-11005' |
| 'JM-16195' | 'MZ-17335' | 'MW-18220' | 'MV-17485' | 'SM-20320' | 'TP-21415' |
| 'JK-15625' | 'PJ-18835' | 'RS-19420' | 'SV-20935' | 'BC-11125' | 'EM-13825' |
| 'BM-11575' | 'KN-16705' | 'KW-16570' | 'SC-20260' | 'CV-12295' | 'SG-20605' |
| 'TM-21010' | 'EM-13810' | 'ML-18040' | 'CR-12580' | 'AZ-10750' | 'PW-19240' |
| 'SC-20380' | 'CM-11935' | 'GM-14695' | 'TB-21250' | 'JM-15865' | 'SC-20575' |
| 'LS-17200' | 'RR-19315' | 'DB-12910' | 'TT-21220' | 'LO-17170' | 'KD-16615' |
| 'NB-18580' | 'BD-11635' | 'CM-12235' | 'EN-13780' | 'KN-16450' | 'BO-11350' |
| 'AG-10300' | 'MC-17635' | 'TA-21385' | 'JF-15295' | 'TT-21265' | 'SB-20170' |
| 'CL-12700' | 'HG-15025' | 'NL-18310' | 'RR-19525' | 'TC-21295' | 'SV-20785' |
| 'BE-11410' | 'SC-20680' | 'DF-13135' | 'FH-14350' | 'MS-17530' | 'RH-19555' |
| 'GA-14515' | 'JP-16135' | 'Dp-13240' | 'MO-17950' | 'DL-13330' | 'MH-18025' |
| 'DR-12940' | 'DM-13015' | 'CA-11965' | 'AC-10660' | 'DM-13345' | 'VF-21715' |
| 'CC-12370' | 'BF-11275' | 'HG-14845' | 'BP-11155' | 'EM-14140' | 'MA-17995' |
| 'AY-10555' | 'GB-14575' | 'JB-16045' | 'MG-17875' | 'SR-20425' | 'JB-16000' |
| 'DM-12955' | 'TC-21475' | 'SW-20350' | 'RE-19450' | 'BF-11215' | 'KB-16405' |
| 'JG-15310' | 'EC-14050' | 'EB-14110' | 'JP-15460' | 'CS-11845' | 'GH-14410' |
| 'PT-19090' | 'JL-15130' | 'AH-10030' | 'CC-12475' | 'DW-13195' | 'SJ-20215' |
| 'BG-11740' | 'LB-16795' | 'CM-11815' | 'EH-14185' | 'TS-21505' | 'PR-18880' |
| 'LC-17050' | 'CS-12490' | 'DH-13075' | 'JO-15145' | 'AH-10690' | 'HJ-14875' |
| 'MH-17455' | 'RD-19930' | 'SC-20020' | 'SU-20665' | 'FC-14335' | 'RB-19330' |

```

'NC-18535' 'DB-12970' 'MF-17665' 'RM-19750' 'VM-21835' 'EJ-13720'
'NC-18415' 'LS-17230' 'KE-16420' 'DH-13675' 'PF-19120' 'VT-21700'
'MG-17695' 'SP-20920' 'CS-12175' 'DK-12895' 'KM-16660' 'AA-10645'
'DD-13570' 'AH-10465' 'TP-21565' 'EK-13795' 'CR-12820' 'SG-20890'
'AH-10585' 'NF-18475' 'BS-11365' 'SH-20635' 'RD-19720' 'AG-10330'
'GR-14560' 'VS-21820' 'TM-21490' 'TS-21430' 'DB-13360' 'NR-18550'
'CS-11860' 'MF-18250' 'LW-17125' 'AR-10345' 'KS-16300' 'AB-10015'
'LR-17035' 'SS-20410' 'JM-15580' 'JK-15325' 'DM-13525' 'ML-18265'
'MH-17290' 'FC-14245' 'TH-21115' 'JK-16090' 'SB-20185' 'BG-11035'
'BW-11065' 'MG-18205' 'DO-13645' 'BP-11050' 'TS-21655' 'EM-14200'
'AO-10810' 'MH-17440' 'SS-20515' 'LD-16855' 'VP-21760' 'TC-21145'
'IM-15055' 'AR-10570' 'CM-12715' 'FW-14395' 'LC-16960' 'HE-14800'
'BD-11560' 'HD-14785' 'CJ-11875' 'RS-19870' 'SC-20845' 'RE-19405'
'SM-20905']
['Claire Gute' 'Darrin Van Huff' "Sean O'Donnell" 'Brosina Hoffman'
'Andrew Allen' 'Irene Maddox' 'Harold Pawlan' 'Pete Kriz'
'Alejandro Grove' 'Zuschuss Donatelli' 'Ken Black' 'Sandra Flanagan'
'Emily Burns' 'Eric Hoffmann' 'Tracy Blumstein' 'Matt Abelman'
'Gene Hale' 'Steve Nguyen' 'Linda Cazamias' 'Ruben Ausman' 'Erin Smith'
'Odella Nelson' "Patrick O'Donnell" 'Lena Hernandez' 'Darren Powers'
'Janet Molinari' 'Ted Butterfield' 'Kunst Miller' 'Paul Stevenson'
'Brendan Sweed' 'Karen Daniels' 'Henry MacAllister' 'Joel Eaton'
'Ken Brennan' 'Stewart Carmichael' 'Duane Noonan' 'Julie Creighton'
'Christopher Schild' 'Paul Gonzalez' 'Gary Mitchum' 'Jim Sink'
'Karl Braun' 'Roger Barcio' 'Parhena Norris' 'Katherine Ducich'
'Elpida Rittenbach' 'Rick Bensley' 'Gary Zandusky' 'Lena Cacioppo'
'Janet Martin' 'Pete Armstrong' 'Cynthia Voltz' 'Clay Ludtke'
'Ryan Crowe' 'Dave Kipp' 'Greg Guthrie' 'Steven Cartwright'
'Alan Dominguez' 'Philip Fox' 'Troy Staebel' 'Lindsay Shagiari'
'Dorothy Wardle' 'Lena Creighton' 'Jonathan Doherty' 'Sally Hughesby'
'Sandra Glassco' 'Helen Andreada' 'Maureen Gastineau' 'Justin Ellison'
'Tamara Willingham' 'Stephanie Phelps' 'Neil Knudson' 'Dave Brooks'
'Nora Paige' 'Ted Trevino' 'Eric Murdock' 'Ruben Dartt' 'Max Jones'
'Becky Martin' 'Chad Sievert' 'Jennifer Braxton' 'Shirley Jackson'
'Jim Kriz' 'David Kendrick' 'Robert Marley' 'Sally Knutson'
'Frank Merwin' 'Alice McCarthy' 'Mark Packer' 'Mary Zewe'
'Cassandra Brandow' 'Valerie Mitchum' 'Fred Hopkins' 'Maria Bertelson'
'Bruce Stewart' 'Logan Currie' 'Heather Kirkland' 'Laurel Elliston'
'Joseph Holt' 'Michael Stewart' 'Victoria Wilson' 'Jonathan Howell'
'Joni Blumstein' 'David Smith' 'Valerie Dominguez' 'Erin Ashbrook'
'David Bremer' 'Ken Lonsdale' 'Dianna Wilson' 'Logan Haushalter'
'Kelly Collister' 'Delfina Latchford' 'Dan Reichenbach' 'Craig Carreira'
'Dorris liebe' 'Sean Braxton' 'Roy Collins' 'Alan Hwang'
'Claudia Bergmann' 'Christine Abelman' 'Kristen Hastings'
'Barry Blumstein' 'Andrew Gjertsen' "Jas O'Carroll" 'Alan Haines'
'Nick Zandusky' 'Kelly Lampkin' 'Alan Schoenberger' 'Corey Roper'
'Shahid Hopkins' 'Ben Peterman' 'Thomas Seio' 'Andy Gerbode' 'Sung Pak'
'Nathan Mautz' 'Frank Atkinson' 'Grace Kelly' 'Don Jones'
"Patrick O'Brill" 'John Lucas' 'Doug Bickford' 'Alyssa Crouse'
'Clay Cheatham' 'Tamara Dahlen' 'Adam Bellavance' 'Jeremy Lonsdale'
'Victoria Brennan' 'Katrina Willman' 'Julia Dunbar' 'Michael Kennedy'
'Guy Thornton' 'Arthur Gainer' 'Muhammed MacIntyre' 'Allen Rosenblatt'
'Russell Applegate' 'Alejandro Savely' 'Laura Armstrong' 'Denny Ordway'
'Dean Katz' 'Nathan Gelder' 'Mike Vittorini' 'Jack Garza' 'Bart Pistole'
'Victor Preis' 'Saphhira Shifley' 'Anna Gayman' 'Luke Foster'
'Roy Französisch' 'Keith Herrera' 'Kimberly Carter' 'Caroline Jumper'
'Philip Brown' 'Michael Paige' 'Natalie Fritzler' 'Shirley Daniels'
'Ken Heidel' 'Ross Baird' 'Mike Kennedy' 'Philisse Overcash'
'Brenda Bowman' 'Troy Blackwell' 'Raymond Buch' 'Ed Braxton'
'Sanjit Chand' 'Tanja Norvell' 'Joni Sundaresam' 'Maya Herman'
'Jeremy Pistek' 'Jeremy Ellison' 'John Grady' 'Xylona Preis' 'Erin Mull'
'Michelle Tran' 'Sue Ann Reed' 'Carl Weiss' 'Astrea Jones' 'Sonia Sunley'
"Rose O'Brian" 'Maribeth Dona' 'Maribeth Yedwab' 'Christopher Martinez'
'Lynn Smith' 'Bradley Nguyen' 'Dean Braden' 'Matt Connell' 'Brian Dahlen'

```

'Patricia Hirasaki' 'Mike Gockenbach' 'Karen Bern' 'Jasper Cacioppo'
 'Rob Lucas' 'Allen Arnold' 'Emily Phan' 'Darren Koutras'
 'Bradley Drucker' 'Liz MacKendrick' 'Adrian Shami' 'Bill Donatelli'
 'Greg Tran' 'Ashley Jarboe' 'Olvera Toch' 'Liz Pelletier'
 'Cynthia Arntzen' 'Jeremy Farry' 'Frank Preis' 'Ellis Ballard'
 'Jennifer Ferguson' 'Sarah Foster' 'Trudy Glocke' 'Carlos Soltero'
 'Charles Crestani' 'Dianna Vittorini' 'Bruce Degenhardt'
 'Zuschuss Carroll' 'Melanie Seite' 'Lena Radford' 'Theone Pippenger'
 'Chloris Kastensmidt' 'Alan Shonely' 'Andrew Roberts' 'Nona Balk'
 'Giulietta Dortch' 'Clytie Kelty' 'Nat Gilpin' 'Christina Anderson'
 'Sylvia Foulston' 'Meg O'Connel' 'Annie Thurman' 'Fred McMath'
 'Denny Joy' 'Max Engle' 'Justin Deggeller' 'John Lee' 'Sean Christensen'
 'Chuck Clark' 'Anthony Rawles' 'Steven Roelle' 'Craig Reiter'
 'Eugene Hildebrand' 'Sibella Parks' 'Tiffany House' 'Resi Pölking'
 'Rob Beeghly' 'Carol Darley' 'Doug Jacobs' 'Grant Thornton'
 'Michael Chen' 'Ralph Arnett' 'Naresj Patel' 'Alan Barnes' 'Jesus Ocampo'
 'Jay Kimmel' 'Brad Norvell' 'David Philippe' 'Tracy Hopkins'
 'Arthur Prichep' 'Roland Schwarz' 'Seth Vernon' 'Christine Kargatis'
 'Ross DeVincentis' 'Mathew Reese' 'Steve Chapman' 'Jay Fein'
 'Emily Grady' 'Darrin Sayre' 'Phillina Ober' 'Sung Shariari'
 'Peter Bühler' 'Roland Fjeld' 'Yoseph Carroll' 'Debra Catini'
 'Christine Phan' 'Barry Französisch' 'Lisa Hazard' 'Chris Selesnick'
 'Anthony Johnson' 'Benjamin Venier' 'Dan Lawera' 'Bryan Mills'
 'Liz Thompson' 'Joe Kamberova' 'Erica Smith' 'Rick Hansen' 'Carlos Daly'
 'Helen Wasserman' 'Mike Caudle' 'Gary McGarr' 'Pauline Johnson'
 'Bart Watters' 'Toby Ritter' 'Patrick Gardner' 'James Lanier'
 'Brian Moss' 'Eudokia Martin' 'Art Foster' 'Guy Armstrong' 'Cyma Kinney'
 'Dave Poirier' 'Berenike Kampe' 'Sanjit Jacobs' 'Chuck Magee'
 'Anthony Jacobs' 'Linda Southworth' 'Guy Phonely' 'Paul Knutson'
 'Sally Matthias' 'Anthony Garverick' 'Peter McVee' 'Lauren Leatherbury'
 'Jill Stevenson' 'Ed Ludwig' 'Pamela Coakley' 'Hunter Lopez'
 'Maribeth Schnelling' 'George Bell' 'Justin Ritter' 'Bill Eplett'
 'Sample Company A' 'Rob Williams' 'Sanjit Engle' 'Adam Hart'
 'Jessica Myrick' 'Joel Jenkins' 'Ralph Kennedy' 'Catherine Glotzbach'
 'Rachel Payne' 'Karen Carlisle' 'Katherine Hughes' 'Greg Hansen'
 'Scott Williamson' 'Joseph Airdo' 'Daniel Lacy' 'Lindsay Williams'
 'Thomas Brumley' 'Bryan Spruell' 'Robert Waldorf' 'Tracy Zic'
 'Ann Steele' 'Toby Swindell' 'Sara Luxemburg' 'Mitch Willingham'
 'Rob Dowd' 'Ryan Akin' 'Meg Tillman' 'Vivek Gonzalez' 'John Stevenson'
 'Kalyca Meade' 'Hallie Redmond' 'Deanra Eno' 'Allen Goldenen'
 'Jennifer Jackson' 'Jennifer Halladay' 'Robert Dilbeck' 'Mary O'Rourke'
 'Noel Staavos' 'Deirdre Greer' 'Nicole Fjeld' 'Matthew Grinstein'
 'Theresa Swint' 'Brian DeCherney' 'Charles McCrossin' 'Skye Norling'
 'Erica Hernandez' 'Frank Olsen' 'Maurice Satty' 'Chad Cunningham'
 'Don Weiss' 'Bill Tyler' 'Craig Yedwab' 'Brad Thomas' 'Penelope Sewall'
 'Paul Van Hugh' 'Neoma Murray' 'Dionis Lloyd' 'Christine Sundaresam'
 'Frank Hawley' 'Nat Carroll' 'Alex Avila' 'Larry Tron' 'Anne Pryor'
 'Paul MacIntyre' 'Alyssa Tate' 'Cathy Armstrong' 'Harold Ryan'
 'Bradley Talbott' 'Larry Hughes' 'Steven Ward' 'Stefania Perrino'
 'Ben Ferrer' 'Kean Thornton' 'Brooke Gillingham' 'Greg Matthias'
 'Eva Jacobs' 'Nora Preis' 'Mick Hernandez' 'Jocasta Rupert'
 'Suzanne McNair' 'Chris Cortes' 'Phillip Flathmann' 'Dan Campbell'
 'Bryan Davis' 'Gene McClure' 'Todd Boyes' 'Justin Hirsh' 'Erica Bern'
 'Quincy Jones' 'Tracy Collins' 'Chuck Sachs' 'Henry Goldwyn'
 'Laurel Workman' 'Matt Collins' 'Liz Preis' 'Evan Bailliet'
 'George Zrebassa' 'Cathy Prescott' 'Frank Gastineau' 'Lisa DeCherney'
 'Alejandro Ballentine' 'Michael Nguyen' 'Jim Radford' 'Jamie Frazer'
 'Chad McGuire' 'Aaron Smayling' 'Beth Paige' 'Natalie DeCherney'
 'Larry Blacks' 'Kean Takahito' 'Harry Marie' 'Ann Blume' 'Sam Zeldin'
 'Michael Granlund' 'Julie Kriz' 'Paul Prost' 'Yana Sorensen'
 'Katherine Murray' 'Adrian Barton' 'Helen Abelman' 'Beth Thompson'
 'Stuart Van' 'Rick Wilson' 'Damala Kotsonis' 'Shui Tom' 'Michael Moore'
 'Pauline Webber' 'Shaun Chance' 'Thais Sissman' 'Mark Cousins'
 'Maria Etezadi' 'Nicole Hansen' 'Mick Brown' 'Keith Dawkins'

'Katherine Nockton' 'Nora Pelletier' 'Cindy Stewart' 'Maxwell Schwartz'
 'Corinna Mitchell' 'Corey-Lock' 'Todd Sumrall' 'Jane Waco' 'John Dryer'
 'Pauline Chand' 'Andy Reiter' 'Arianne Irving' 'Tom Boeckenhauer'
 'Paul Lucas' 'Gary Hwang' 'Mike Pelletier' 'Jim Mitchum' 'Carl Ludwig'
 'Deborah Brumfield' 'Ivan Gibson' 'Bobby Odegard' 'Aimee Bixby'
 'Julia West' 'Edward Becker' 'Sheri Gordon' 'Charlotte Melton'
 'Anthony Witt' 'Mick Crebagga' 'Tonja Turnell' 'Vivek Grady'
 'Muhammed Yedwab' 'Rick Duston' 'Dennis Pardue' 'Marina Lichtenstein'
 'Jenna Caffey' 'James Galang' 'Marc Crier' 'Natalie Webber'
 'Toby Braunhardt' 'Bill Stewart' 'Hilary Holden' 'Christina DeMoss'
 'Thea Hendricks' 'Michelle Moray' 'Neola Schneider' 'Robert Barroso'
 'Shaun Weien' 'Eric Barreto' 'Pamela Stobb' 'Herbert Flentye'
 'Henia Zydlo' 'Cynthia Delaney' 'Jamie Kunitz' 'Filia McAdams'
 'Emily Ducich' 'Dianna Arnett' 'Joni Wasserman' 'Raymond Messe'
 'Max Ludwig' 'Craig Carroll' 'Jim Epp' 'Roy Phan' 'Thomas Boland'
 'Brad Eason' 'Jill Fjeld' 'Phillip Breyer' 'Brian Thompson' 'Janet Lee'
 'Cathy Hwang' 'Neil Ducich' 'Barbara Fisher' 'Katharine Harms'
 'Giulietta Weimer' 'Noah Childs' 'Michelle Ellison' 'Benjamin Patterson'
 'John Castell' 'Adam Shillingsburg' 'Amy Cox' 'Michael Dominguez'
 'Duane Benoit' 'Erica Hackney' 'Edward Hooks' 'Scot Coram' 'Joe Elijah'
 'Ann Chong' 'Joy Daniels' 'Christy Brittain' 'Joy Smith' 'Luke Weiss'
 'Stuart Calhoun' 'Anne McFarland' 'Rick Huthwaite' 'Carol Triggs'
 'Matt Collister' 'Corey Catlett' 'Kelly Andreada' 'Tamara Chand'
 'Bart Folk' 'Magdelene Morse' 'Adrian Hane' 'Ben Wallace' 'Scot Wooten'
 'Brian Stugart' 'Randy Ferguson' 'William Brown' 'Trudy Brown'
 'Art Ferguson' 'Richard Bierner' 'Karen Ferguson' 'John Huston'
 'Ivan Liston' 'Patrick Bzostek' 'Rob Haberlin' 'Arthur Wiediger'
 'Maris LaWare' 'Dorothy Badders' 'Matt Hagelstein' 'Dennis Kane'
 'Bobby Trafton' 'Denny Blanton' 'Toby Gnade' 'Barry Franz'
 'Justin MacKendrick' 'Maria Zettner' 'Mitch Webber' 'Mark Van Huff'
 'Sean Miller' 'Tom Prescott' 'Jim Karlsson' 'Patrick Jones'
 'Ricardo Sperren' 'Susan Vittorini' 'Becky Castell' 'Elizabeth Moffitt'
 'Brendan Murry' 'Kristina Nunn' 'Kelly Williams' 'Scott Cohen'
 'Christina VanderZanden' 'Speros Goranitis' 'Tamara Manning'
 'Eleni McCrary' 'Michelle Lonsdale' 'Clay Rozendal' 'Annie Zypern'
 'Pierre Wener' 'Shahid Collister' 'Carlos Meador' 'Greg Maxwell'
 'Tim Brockman' 'John Murray' 'Sonia Cooley' 'Luke Schmidt' 'Ralph Ritter'
 'Daniel Byrd' 'Thomas Thornton' 'Lori Olson' 'Ken Dana' 'Nicole Brennan'
 'Brian Derr' 'Chris McAfee' 'Edward Nazzal' 'Kean Nguyen' 'Bill Overfelt'
 'Aleksandra Gannaway' 'Matthew Clasen' 'Tom Ashbrook' 'Jason Fortune-'
 'Tim Taslimi' 'Sarah Bern' 'Craig Leslie' 'Hunter Glantz'
 'Nancy Lomonaco' 'Rick Reed' 'Toby Carlisle' 'Stewart Visinsky'
 'Bobby Elias' 'Steve Carroll' 'David Flashing' 'Fred Harton'
 'MaryBeth Skach' 'Ritsa Hightower' 'George Ashbrook' 'Julie Prescott'
 'Dean percer' 'Michael Oakman' 'Denise Leinenbach' 'Michelle Huthwaite'
 'Daniel Raglin' 'Darrin Martin' 'Carol Adams' 'Anna Chung'
 'Denise Monton' 'Vicky Freymann' 'Christopher Conant' 'Beth Fritzler'
 'Harry Greene' 'Becky Pak' 'Eugene Moren' 'Michelle Arnett' 'Andy Yotov'
 'Giulietta Baptist' 'Julia Barnett' 'Michael Grace' 'Sharelle Roach'
 'Joy Bell-' 'Dario Medina' 'Tony Chapman' 'Sean Wendt' 'Richard Eichhorn'
 'Benjamin Farhat' 'Katrina Bavinger' 'Jason Gross' 'Erin Creighton'
 'Eugene Barchas' 'Jennifer Patt' 'Cari Sayre' 'Gary Hansen'
 'Pete Takahito' 'Jack Lebron' 'Aaron Hawkins' 'Cindy Chapman'
 'David Wiener' 'Sarah Jordon' 'Bruce Geld' 'Laurel Beltran'
 'Candace McMahon' 'Evan Henry' 'Tony Sayre' 'Patrick Ryan' 'Liz Carlisle'
 'Cindy Schnelling' 'Dave Hallsten' 'Jack O'Briant' 'Anna Häberlin'
 'Heather Jas' 'Mark Hamilton' 'Russell D'Ascenzo' 'Sam Craven'
 'Stephanie Ulpright' 'Fred Chung' 'Randy Bradley' 'Nick Crebassa'
 'Darren Budd' 'Maureen Fritzler' 'Roland Murray' 'Vivian Mathis'
 'Ed Jacobs' 'Nathan Cano' 'Lycoris Saunders' 'Katrina Edelman'
 'Duane Huffman' 'Peter Fuller' 'Valerie Takahito' 'Maureen Gnade'
 'Susan Pistek' 'Charles Sheldon' 'Dana Kaydos' 'Khloe Miller'
 'Anna Andreadi' 'Dorothy Dickinson' 'Amy Hunt' 'Tracy Poddar'
 'Eileen Kiefer' 'Cyra Reiten' 'Susan Gilcrest' 'Angele Hood'

'Neil Französisch' 'Bill Shonely' 'Stefanie Holloman' 'Roger Demir'
 'Alex Grayson' 'Georgia Rosenberg' 'Vivek Sundaresam' 'Tony Molinari'
 'Tom Stivers' 'Dennis Bolton' 'Nick Radford' 'Cari Schnelling'
 'Monica Federle' 'Liz Willingham' 'Alex Russell' 'Karen Seio'
 'Aaron Bergman' 'Lisa Ryan' 'Shahid Shariari' 'Jill Matthias'
 'Jason Klamczynski' 'Don Miller' 'Muhammed Lee' 'Marc Harrigan'
 'Frank Carlisle' 'Thea Hudgings' 'Juliana Krohn' 'Sarah Brown'
 'Barry Gonzalez' 'Barry Weirich' 'Mitch Gastineau' 'Doug O'Connell'
 'Barry Pond' 'Trudy Schmidt' 'Evan Minnotte' 'Anthony O'Donnell'
 'Mark Haberlin' 'Shirley Schmidt' 'Lela Donovan' 'Victoria Pisteka'
 'Theresa Coyne' 'Ionia McGrath' 'Anemone Ratner' 'Craig Molinari'
 'Fred Wasserman' 'Lindsay Castell' 'Harold Engle' 'Brendan Dodson'
 'Harold Dahlen' 'Carl Jackson' 'Roy Skaria' 'Sung Chung'
 'Ricardo Emerson' 'Susan MacKendrick']
 ['Consumer' 'Corporate' 'Home Office']
 ['Henderson' 'Los Angeles' 'Fort Lauderdale' 'Concord' 'Seattle'
 'Fort Worth' 'Madison' 'West Jordan' 'San Francisco' 'Fremont'
 'Philadelphia' 'Orem' 'Houston' 'Richardson' 'Naperville' 'Melbourne'
 'Eagan' 'Westland' 'Dover' 'New Albany' 'New York City' 'Troy' 'Chicago'
 'Gilbert' 'Springfield' 'Jackson' 'Memphis' 'Decatur' 'Durham' 'Columbia'
 'Rochester' 'Minneapolis' 'Portland' 'Saint Paul' 'Aurora' 'Charlotte'
 'Orland Park' 'Urbandale' 'Columbus' 'Bristol' 'Wilmington' 'Bloomington'
 'Phoenix' 'Roseville' 'Independence' 'Pasadena' 'Newark' 'Franklin'
 'Scottsdale' 'San Jose' 'Edmond' 'Carlsbad' 'San Antonio' 'Monroe'
 'Fairfield' 'Grand Prairie' 'Redlands' 'Hamilton' 'Westfield' 'Akron'
 'Denver' 'Dallas' 'Whittier' 'Saginaw' 'Medina' 'Dublin' 'Detroit'
 'Tampa' 'Santa Clara' 'Lakeville' 'San Diego' 'Brentwood' 'Chapel Hill'
 'Morristown' 'Cincinnati' 'Inglewood' 'Tamarac' 'Colorado Springs'
 'Belleville' 'Taylor' 'Lakewood' 'Arlington' 'Arvada' 'Hackensack'
 'Saint Petersburg' 'Long Beach' 'Hesperia' 'Murfreesboro' 'Layton'
 'Austin' 'Lowell' 'Manchester' 'Harlingen' 'Tucson' 'Quincy'
 'Pembroke Pines' 'Des Moines' 'Peoria' 'Las Vegas' 'Warwick' 'Miami'
 'Huntington Beach' 'Richmond' 'Louisville' 'Lawrence' 'Canton'
 'New Rochelle' 'Gastonia' 'Jacksonville' 'Auburn' 'Norman' 'Park Ridge'
 'Amarillo' 'Lindenhurst' 'Huntsville' 'Fayetteville' 'Costa Mesa'
 'Parker' 'Atlanta' 'Gladstone' 'Great Falls' 'Lakeland' 'Montgomery'
 'Mesa' 'Green Bay' 'Anaheim' 'Marysville' 'Salem' 'Laredo' 'Grove City'
 'Dearborn' 'Warner Robins' 'Vallejo' 'Mission Viejo' 'Rochester Hills'
 'Plainfield' 'Sierra Vista' 'Vancouver' 'Cleveland' 'Tyler' 'Burlington'
 'Waynesboro' 'Chester' 'Cary' 'Palm Coast' 'Mount Vernon' 'Hialeah'
 'Oceanside' 'Evanston' 'Trenton' 'Cottage Grove' 'Bossier City'
 'Lancaster' 'Asheville' 'Lake Elsinore' 'Omaha' 'Edmonds' 'Santa Ana'
 'Milwaukee' 'Florence' 'Lorain' 'Linden' 'Salinas' 'New Brunswick'
 'Garland' 'Norwich' 'Alexandria' 'Toledo' 'Farmington' 'Riverside'
 'Torrance' 'Round Rock' 'Boca Raton' 'Virginia Beach' 'Murrieta'
 'Olympia' 'Washington' 'Jefferson City' 'Saint Peters' 'Rockford'
 'Brownsville' 'Yonkers' 'Oakland' 'Clinton' 'Encinitas' 'Roswell'
 'Jonesboro' 'Antioch' 'Homestead' 'La Porte' 'Lansing' 'Cuyahoga Falls'
 'Reno' 'Harrisonburg' 'Escondido' 'Royal Oak' 'Rockville' 'Coral Springs'
 'Buffalo' 'Boynton Beach' 'Gulfport' 'Fresno' 'Greenville' 'Macon'
 'Cedar Rapids' 'Providence' 'Pueblo' 'Deltona' 'Murray' 'Middletown'
 'Freeport' 'Pico Rivera' 'Provo' 'Pleasant Grove' 'Smyrna' 'Parma'
 'Mobile' 'New Bedford' 'Irving' 'Vineland' 'Glendale' 'Niagara Falls'
 'Thomasville' 'Westminster' 'Coppell' 'Pomona' 'North Las Vegas'
 'Allentown' 'Tempe' 'Laguna Niguel' 'Bridgeton' 'Everett' 'Watertown'
 'Appleton' 'Bellevue' 'Allen' 'El Paso' 'Grapevine' 'Carrollton' 'Kent'
 'Lafayette' 'Tigard' 'Skokie' 'Plano' 'Suffolk' 'Indianapolis' 'Bayonne'
 'Greensboro' 'Baltimore' 'Kenosha' 'Olathe' 'Tulsa' 'Redmond' 'Raleigh'
 'Muskogee' 'Meriden' 'Bowling Green' 'South Bend' 'Spokane' 'Keller'
 'Port Orange' 'Medford' 'Charlottesville' 'Missoula' 'Apopka' 'Reading'
 'Broomfield' 'Paterson' 'Oklahoma City' 'Chesapeake' 'Lubbock'
 'Johnson City' 'San Bernardino' 'Leominster' 'Bozeman' 'Perth Amboy'
 'Ontario' 'Rancho Cucamonga' 'Moorhead' 'Mesquite' 'Stockton'
 'Ormond Beach' 'Sunnyvale' 'York' 'College Station' 'Saint Louis'

'Manteca' 'San Angelo' 'Salt Lake City' 'Knoxville' 'Little Rock'
 'Lincoln Park' 'Marion' 'Littleton' 'Bangor' 'Southaven' 'New Castle'
 'Midland' 'Sioux Falls' 'Fort Collins' 'Clarksville' 'Sacramento'
 'Thousand Oaks' 'Malden' 'Holyoke' 'Albuquerque' 'Sparks' 'Coachella'
 'Elmhurst' 'Passaic' 'North Charleston' 'Newport News' 'Jamestown'
 'Mishawaka' 'La Quinta' 'Tallahassee' 'Nashville' 'Bellingham'
 'Woodstock' 'Haltom City' 'Wheeling' 'Summerville' 'Hot Springs'
 'Englewood' 'Las Cruces' 'Hoover' 'Frisco' 'Vacaville' 'Waukesha'
 'Bakersfield' 'Pompano Beach' 'Corpus Christi' 'Redondo Beach' 'Orlando'
 'Orange' 'Lake Charles' 'Highland Park' 'Hempstead' 'Noblesville'
 'Apple Valley' 'Mount Pleasant' 'Sterling Heights' 'Eau Claire' 'Pharr'
 'Billings' 'Gresham' 'Chattanooga' 'Meridian' 'Bolingbrook' 'Maple Grove'
 'Woodland' 'Missouri City' 'Pearland' 'San Mateo' 'Grand Rapids'
 'Visalia' 'Overland Park' 'Temecula' 'Yucaipa' 'Revere' 'Conroe'
 'Tinley Park' 'Dubuque' 'Dearborn Heights' 'Santa Fe' 'Hickory'
 'Carol Stream' 'Saint Cloud' 'North Miami' 'Plantation'
 'Port Saint Lucie' 'Rock Hill' 'Odessa' 'West Allis' 'Chula Vista'
 'Manhattan' 'Altoona' 'Thornton' 'Champaign' 'Texarkana' 'Edinburg'
 'Baytown' 'Greenwood' 'Woonsocket' 'Superior' 'Bedford' 'Covington'
 'Broken Arrow' 'Miramar' 'Hollywood' 'Deer Park' 'Wichita' 'McAllen'
 'Iowa City' 'Boise' 'Cranston' 'Port Arthur' 'Citrus Heights'
 'The Colony' 'Daytona Beach' 'Bullhead City' 'Portage' 'Fargo' 'Elkhart'
 'San Gabriel' 'Margate' 'Sandy Springs' 'Mentor' 'Lawton' 'Hampton'
 'Rome' 'La Crosse' 'Lewiston' 'Hattiesburg' 'Danville' 'Logan'
 'Waterbury' 'Athens' 'Avondale' 'Marietta' 'Yuma' 'Wausau' 'Pasco'
 'Oak Park' 'Pensacola' 'League City' 'Gaithersburg' 'Lehi' 'Tuscaloosa'
 'Moreno Valley' 'Georgetown' 'Loveland' 'Chandler' 'Helena' 'Kirkwood'
 'Waco' 'Frankfort' 'Bethlehem' 'Grand Island' 'Woodbury' 'Rogers'
 'Clovis' 'Jupiter' 'Santa Barbara' 'Cedar Hill' 'Norfolk' 'Draper'
 'Ann Arbor' 'La Mesa' 'Pocatello' 'Holland' 'Milford' 'Buffalo Grove'
 'Lake Forest' 'Redding' 'Chico' 'Utica' 'Conway' 'Cheyenne' 'Owensboro'
 'Caldwell' 'Kenner' 'Nashua' 'Bartlett' 'Redwood City' 'Lebanon'
 'Santa Maria' 'Des Plaines' 'Longview' 'Hendersonville' 'Waterloo'
 'Cambridge' 'Palatine' 'Beverly' 'Eugene' 'Oxnard' 'Renton' 'Glenview'
 'Delray Beach' 'Commerce City' 'Texas City' 'Wilson' 'Rio Rancho'
 'Goldsboro' 'Montebello' 'El Cajon' 'Beaumont' 'West Palm Beach'
 'Abilene' 'Normal' 'Saint Charles' 'Camarillo' 'Hillsboro' 'Burbank'
 'Modesto' 'Garden City' 'Atlantic City' 'Longmont' 'Davis' 'Morgan Hill'
 'Clifton' 'Sheboygan' 'East Point' 'Rapid City' 'Andover' 'Kissimmee'
 'Shelton' 'Danbury' 'Sanford' 'San Marcos' 'Greeley' 'Mansfield' 'Elyria'
 'Twin Falls' 'Coral Gables' 'Romeoville' 'Marlborough' 'Laurel' 'Bryan'
 'Pine Bluff' 'Aberdeen' 'Hagerstown' 'East Orange' 'Arlington Heights'
 'Oswego' 'Coon Rapids' 'San Clemente' 'San Luis Obispo' 'Springdale'
 'Lodi' 'Mason']
 ['Kentucky' 'California' 'Florida' 'North Carolina' 'Washington' 'Texas'
 'Wisconsin' 'Utah' 'Nebraska' 'Pennsylvania' 'Illinois' 'Minnesota'
 'Michigan' 'Delaware' 'Indiana' 'New York' 'Arizona' 'Virginia'
 'Tennessee' 'Alabama' 'South Carolina' 'Oregon' 'Colorado' 'Iowa' 'Ohio'
 'Missouri' 'Oklahoma' 'New Mexico' 'Louisiana' 'Connecticut' 'New Jersey'
 'Massachusetts' 'Georgia' 'Nevada' 'Rhode Island' 'Mississippi'
 'Arkansas' 'Montana' 'New Hampshire' 'Maryland' 'District of Columbia'
 'Kansas' 'Vermont' 'Maine' 'South Dakota' 'Idaho' 'North Dakota'
 'Wyoming' 'West Virginia']
 ['South' 'West' 'Central' 'East']
 ['FUR-BO-10001798' 'FUR-CH-10000454' 'OFF-LA-10000240' ...
 'OFF-AP-10003099' 'TEC-PH-10002645' 'OFF-ST-10001627']
 ['Furniture' 'Office Supplies' 'Technology']
 ['Bookcases' 'Chairs' 'Labels' 'Tables' 'Storage' 'Furnishings' 'Art'
 'Phones' 'Binders' 'Appliances' 'Paper' 'Accessories' 'Envelopes'
 'Fasteners' 'Supplies' 'Machines' 'Copiers']
 ['Bush Somerset Collection Bookcase'
 'Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back'
 'Self-Adhesive Address Labels for Typewriters by Universal' ...
 'Eureka Hand Vacuum, Bagless' 'LG G2'

```
'Eldon Jumbo ProFile Portable File Boxes Graphite/Black']
[261.96 731.94 14.62 ... 97.98 55.6 243.16]
[ 2  3  5  7  4  6  9  1  8 14 11 13 10 12]
[0.   0.45 0.2  0.8  0.3  0.5  0.7  0.6  0.32 0.1  0.4  0.15]
[ 41.9136 219.582   6.8714 ...   4.1028 13.32   72.948 ]
```

In [123...

```
#dropping unnecessary columns
orders = orders.drop(columns=["Row ID", "Ship Date", "Ship Mode", "Country", "Postal Co
```

In [125...

```
orders.head()
```

Out[125...

| | Order ID | Order Date | Customer ID | Customer Name | Segment | City | State | Region | Product ID | Categoror |
|---|----------------|------------|-------------|-----------------|-----------|-----------------|------------|--------|-----------------|---------------|
| 0 | CA-2016-152156 | 2016-11-08 | CG-12520 | Claire Gute | Consumer | Henderson | Kentucky | South | FUR-BO-10001798 | Furnitur |
| 1 | CA-2016-152156 | 2016-11-08 | CG-12520 | Claire Gute | Consumer | Henderson | Kentucky | South | FUR-CH-10000454 | Furnitur |
| 2 | CA-2016-138688 | 2016-06-12 | DV-13045 | Darrin Van Huff | Corporate | Los Angeles | California | West | OFF-LA-10000240 | Offic Supplie |
| 3 | US-2015-108966 | 2015-10-11 | SO-20335 | Sean O'Donnell | Consumer | Fort Lauderdale | Florida | South | FUR-TA-10000577 | Furnitur |
| 4 | US-2015-108966 | 2015-10-11 | SO-20335 | Sean O'Donnell | Consumer | Fort Lauderdale | Florida | South | OFF-ST-10000760 | Offic Supplie |

In [126...

```
orders.isnull().sum() #no null values
```

Out[126...

| | |
|---------------|---|
| Order ID | 0 |
| Order Date | 0 |
| Customer ID | 0 |
| Customer Name | 0 |
| Segment | 0 |
| City | 0 |
| State | 0 |
| Region | 0 |
| Product ID | 0 |
| Category | 0 |
| Sub-Category | 0 |
| Product Name | 0 |
| Sales | 0 |
| Quantity | 0 |
| Discount | 0 |

```
Profit          0
dtype: int64
```

```
In [130... orders.duplicated().sum() #duplicated values
```

```
Out[130... 1
```

```
In [135... orders = orders.drop_duplicates()
```

```
In [136... orders.duplicated().sum()
```

```
Out[136... 0
```

```
In [137... orders["Sub-Category"].unique()
```

```
Out[137... array(['Bookcases', 'Chairs', 'Labels', 'Tables', 'Storage',
      'Furnishings', 'Art', 'Phones', 'Binders', 'Appliances', 'Paper',
      'Accessories', 'Envelopes', 'Fasteners', 'Supplies', 'Machines',
      'Copiers'], dtype=object)
```

```
In [138... orders["Segment"].unique()
```

```
Out[138... array(['Consumer', 'Corporate', 'Home Office'], dtype=object)
```

```
In [139... orders["Region"].unique()
```

```
Out[139... array(['South', 'West', 'Central', 'East'], dtype=object)
```

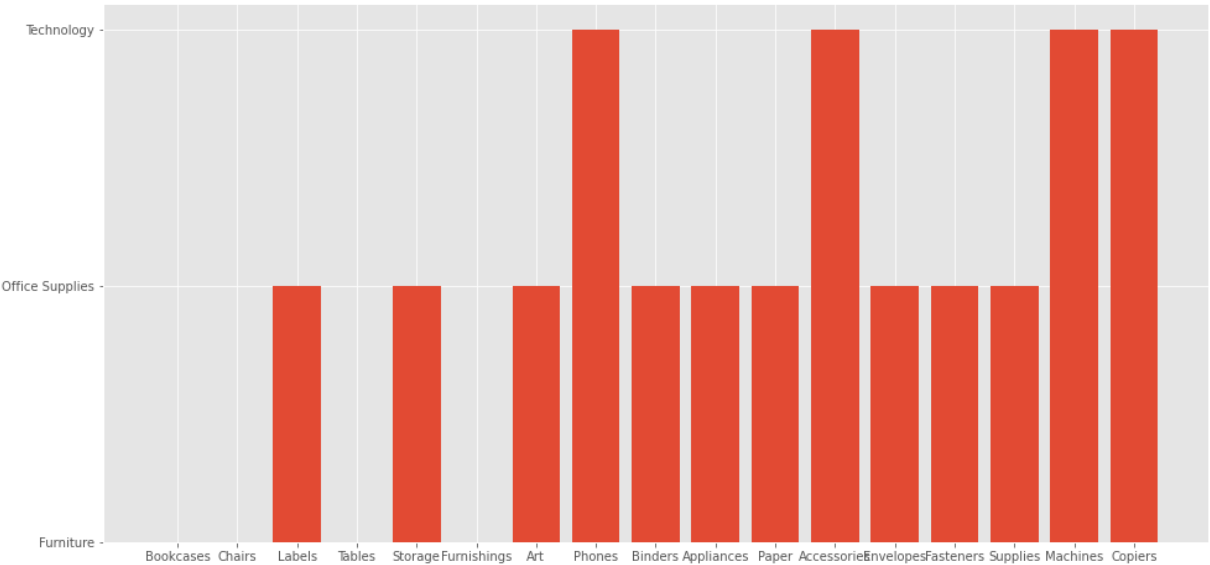
```
In [140... orders["Category"].unique()
```

```
Out[140... array(['Furniture', 'Office Supplies', 'Technology'], dtype=object)
```

```
In [181... #Sub-Categories with reference to Categories
for i in orders['Category'].unique():
    print(i, orders.loc[orders['Category'] == i, 'Sub-Category'].unique())
```

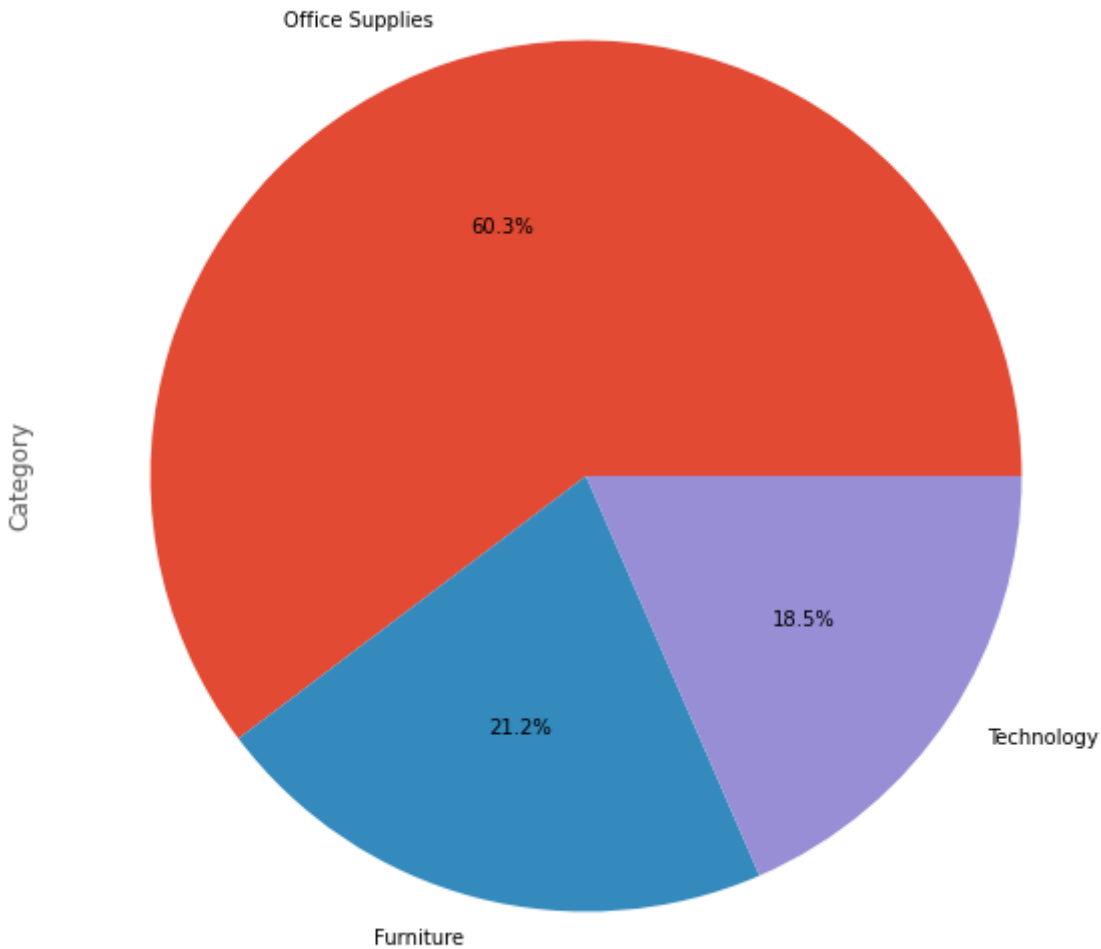
```
Furniture ['Bookcases' 'Chairs' 'Tables' 'Furnishings']
Office Supplies ['Labels' 'Storage' 'Art' 'Binders' 'Appliances' 'Paper' 'Envelopes'
 'Fasteners' 'Supplies']
Technology ['Phones' 'Accessories' 'Machines' 'Copiers']
```

```
In [142... plt.figure(figsize=(16,8))
plt.bar('Sub-Category', 'Category', data=orders)
plt.show()
```



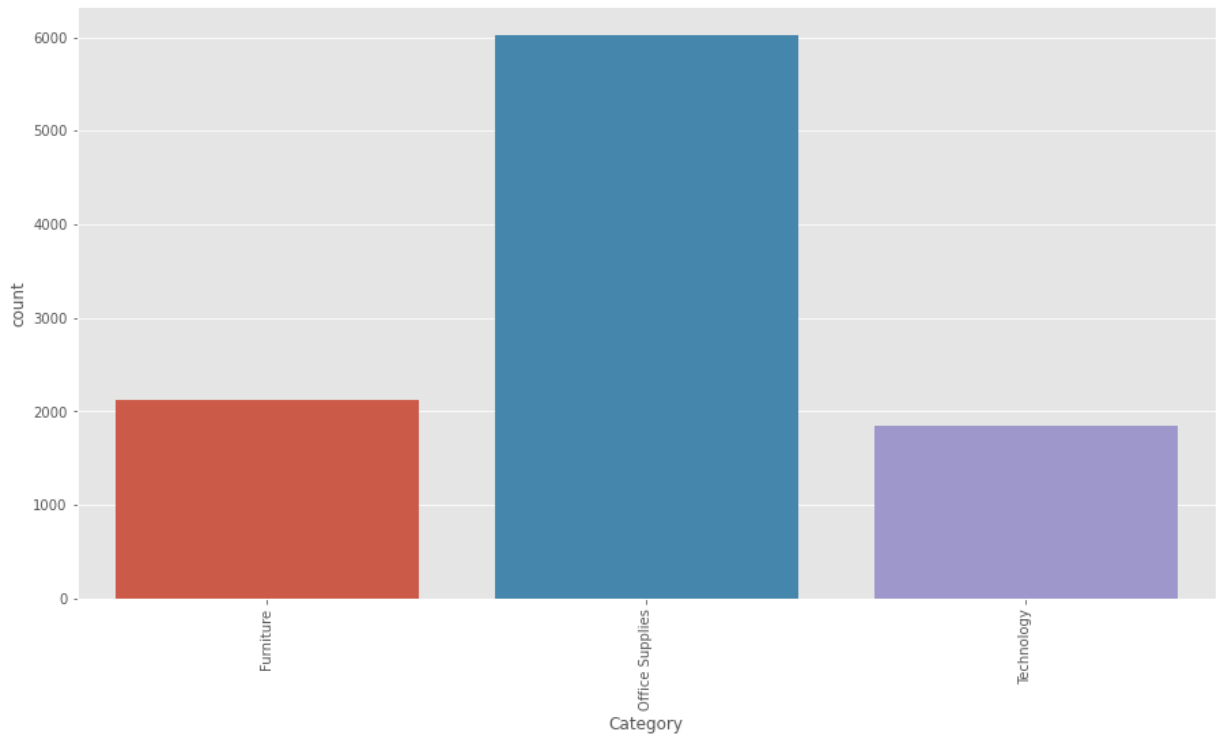
Category Analysis

```
In [49]: plt.figure(figsize=(12,10))
orders['Category'].value_counts().plot.pie(autopct="%1.1f%%")
plt.show()
#the store has a lot of office supplies
```



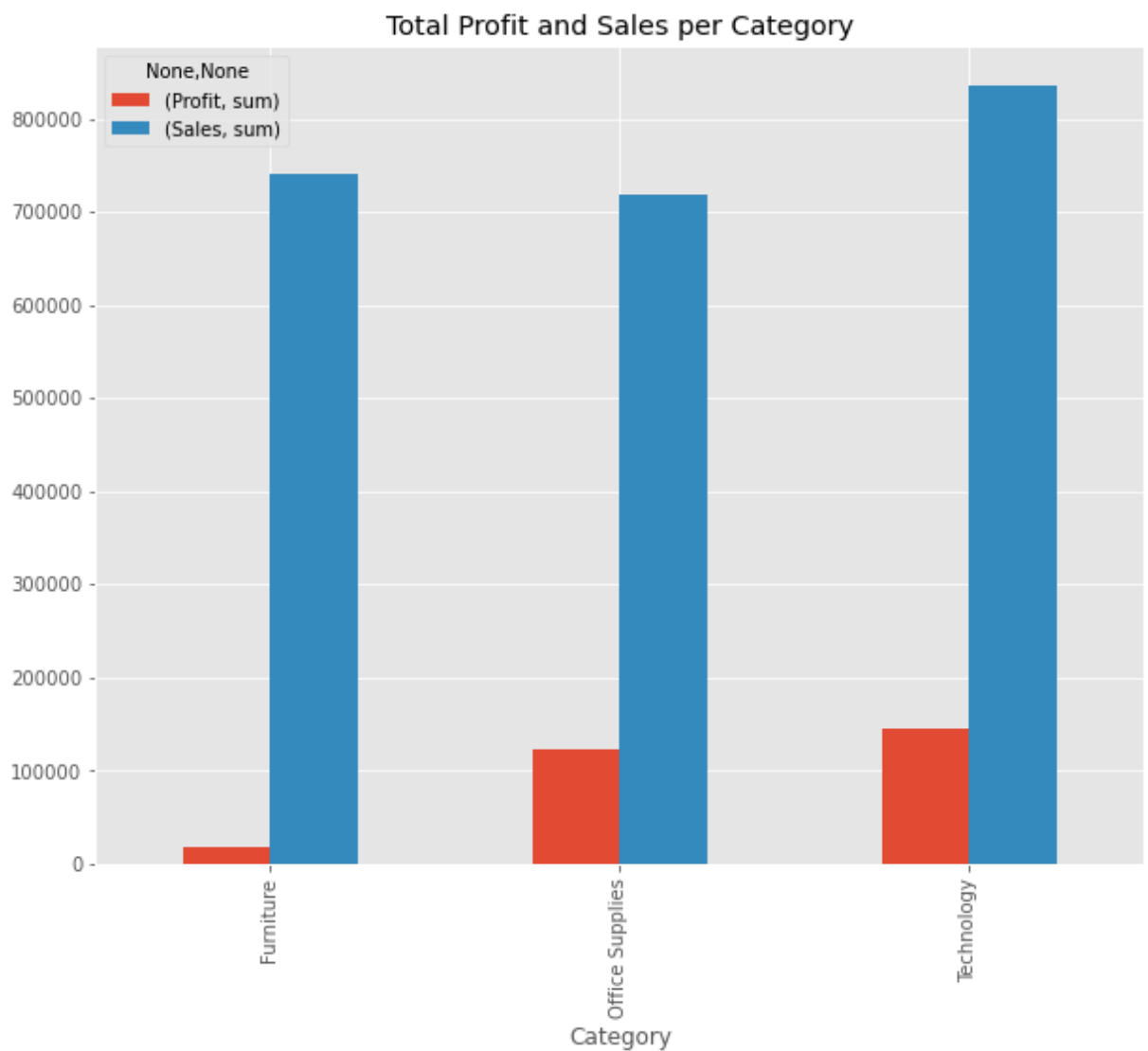
In [144...

```
plt.figure(figsize=(15,8))
sns.countplot(x=orders['Category'])
plt.xticks(rotation=90)
plt.show()
```



In [145...

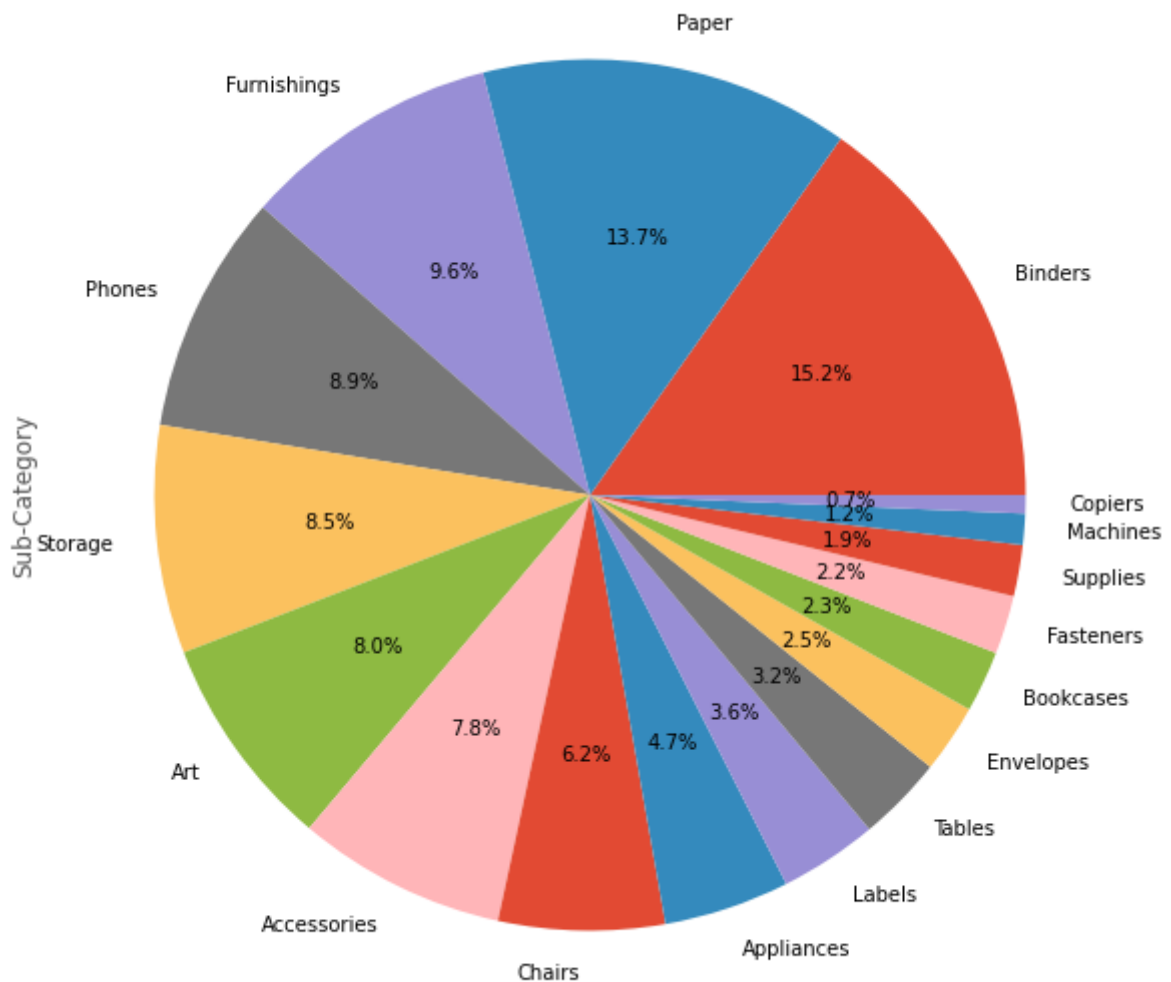
```
orders.groupby('Category')[['Profit', 'Sales']].agg(['sum']).plot.bar()
plt.title('Total Profit and Sales per Category')
plt.rcParams['figure.figsize'] = [10,8]
plt.show()
#we can see that the category technology has the most sales as well as profit
```



Sub-Category Analysis

In [146...

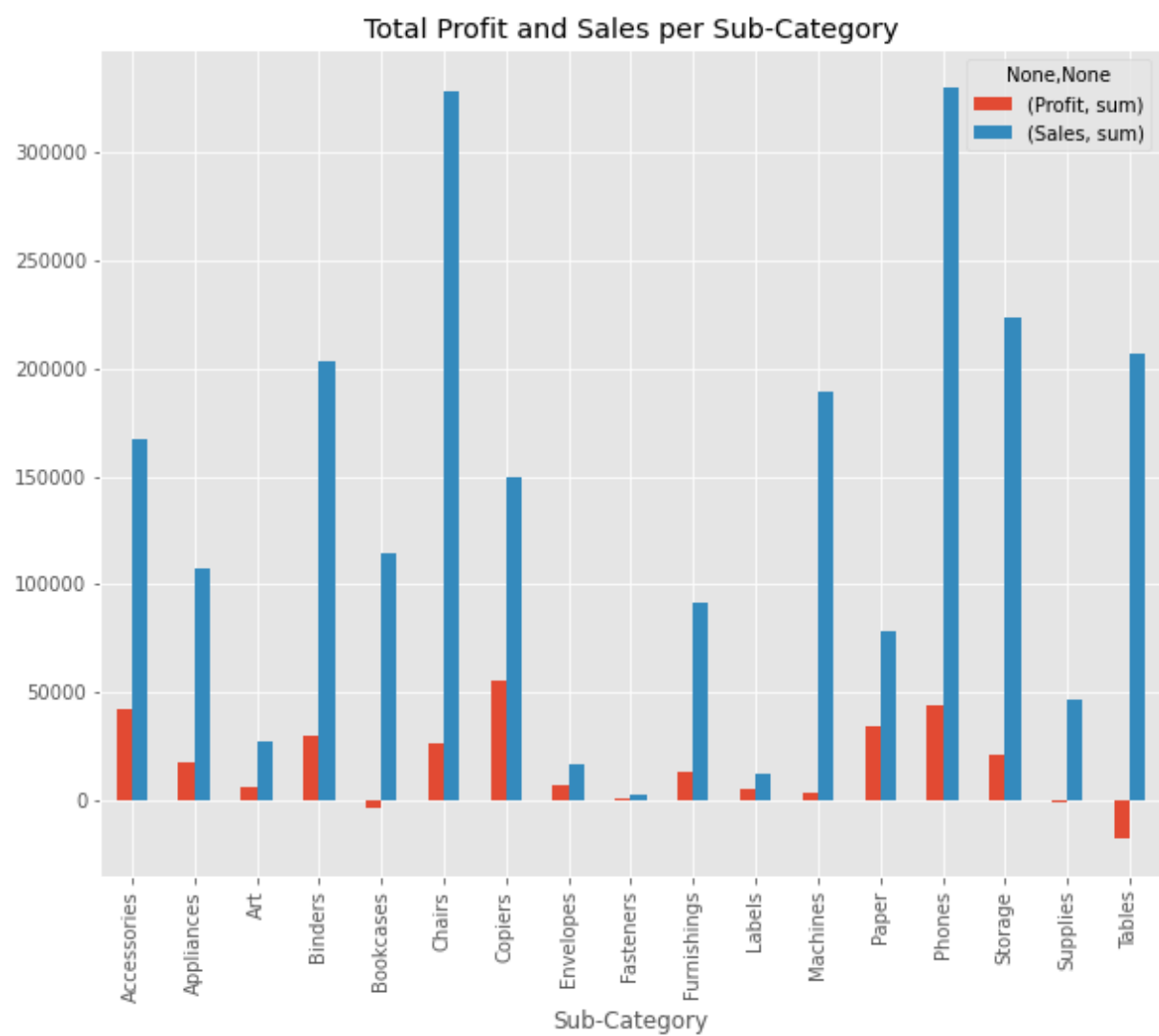
```
plt.figure(figsize=(12,10))
orders['Sub-Category'].value_counts().plot.pie(autopct="%1.1f%%")
plt.show()
#The amount of orders for Binders are the most
```



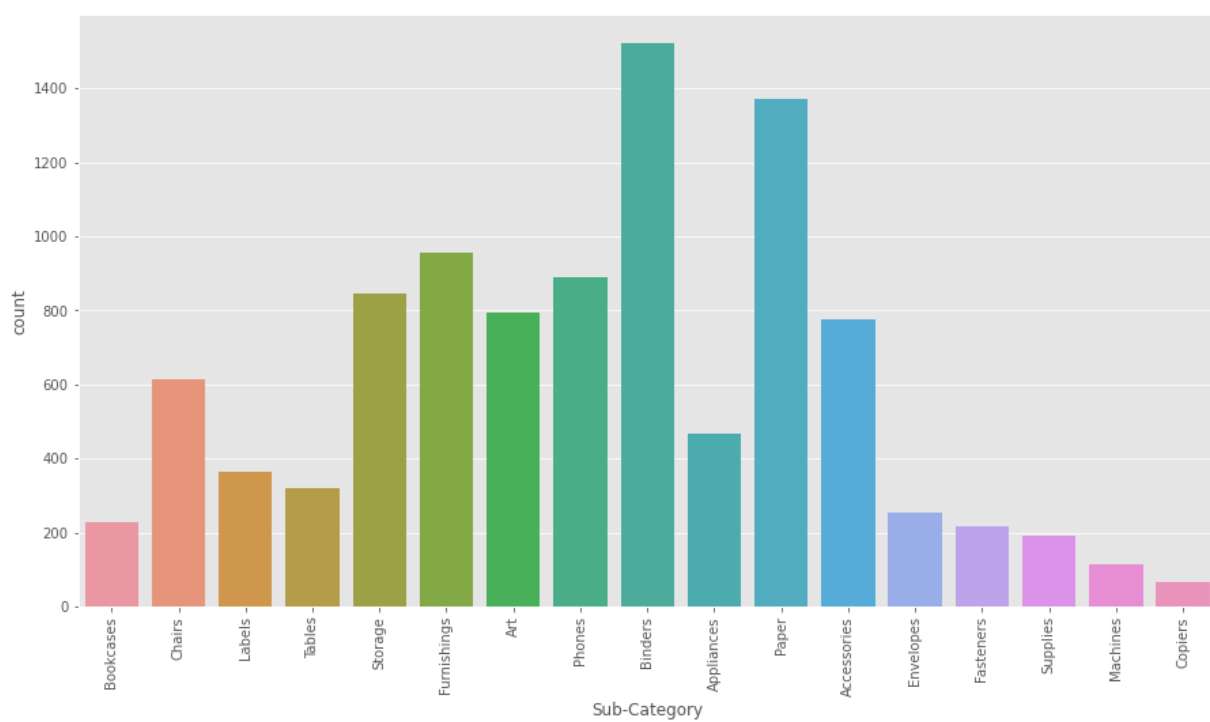
In [147...

```
orders.groupby('Sub-Category')[['Profit', 'Sales']].agg(['sum']).plot.bar()
plt.title('Total Profit and Sales per Sub-Category')
plt.rcParams['figure.figsize'] = [10,8]
plt.show()
```

#although the sales of tables and bookcases are comparatively more but we can see th
#the most profitable is copiers although the sales are not that high
#whereas for accessories and phones we can see higher sales along with high profit



```
In [148...  
plt.figure(figsize=(15,8))  
sns.countplot(x=orders['Sub-Category'])  
plt.xticks(rotation=90)  
plt.show()
```

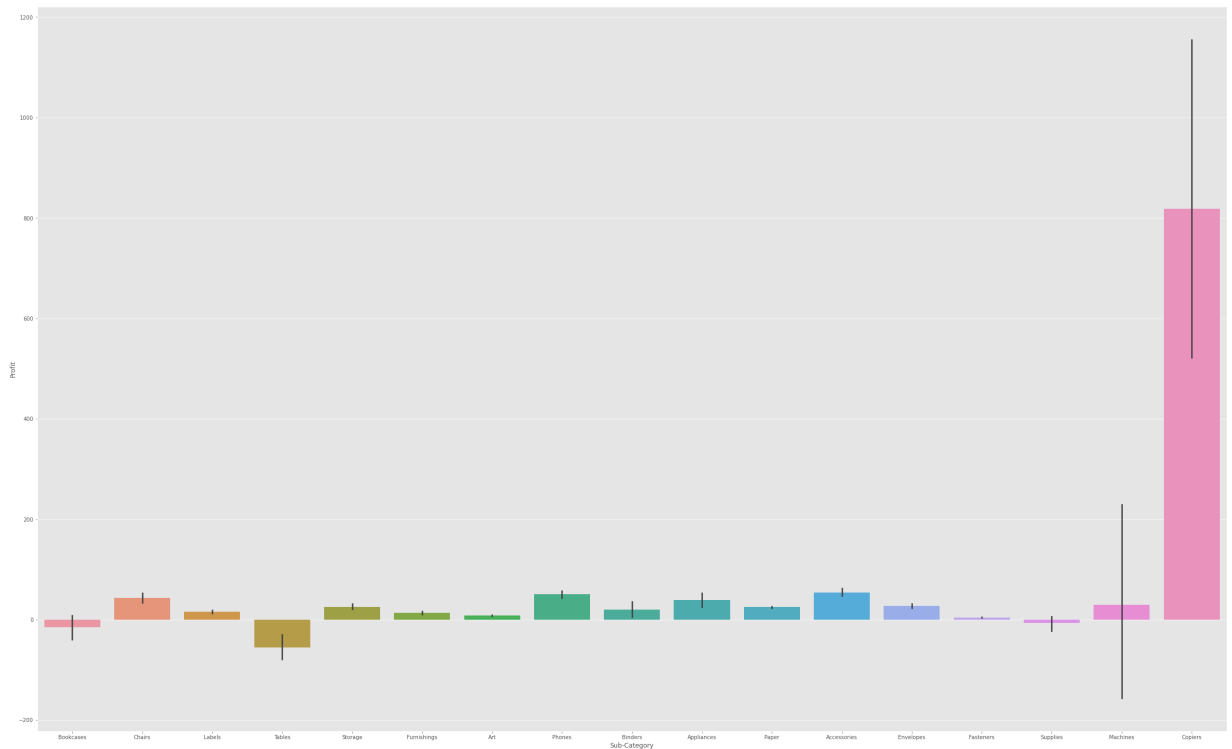


In [149...

```
plt.figure(figsize=(40,25))
sns.barplot(x=orders['Sub-Category'], y=orders['Profit'])
#Again using this barplot we can see observe the above conclusions
```

Out[149...

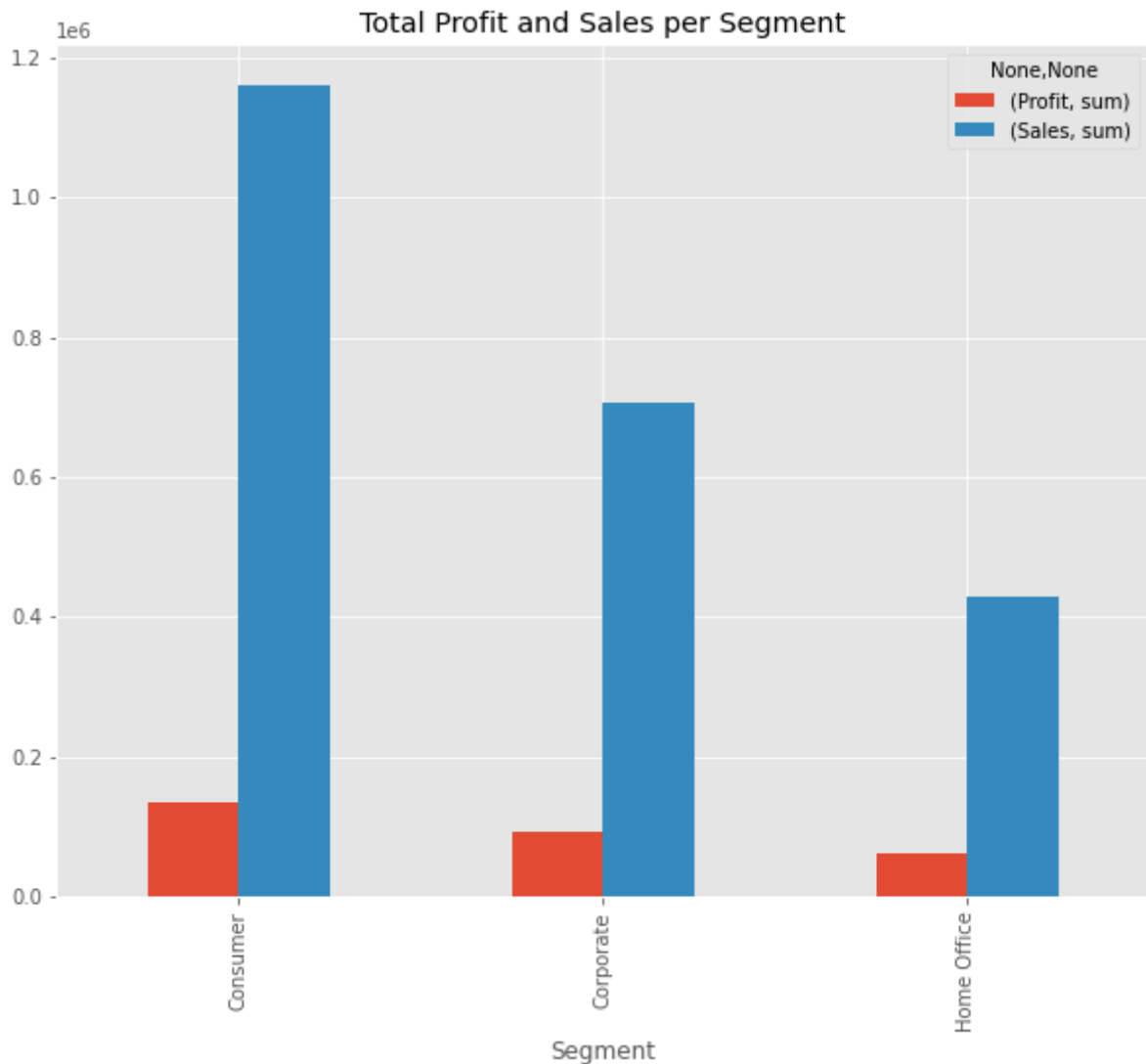
```
<AxesSubplot:xlabel='Sub-Category', ylabel='Profit'>
```



Segment Analysis

In [150...

```
orders.groupby('Segment')[['Profit', 'Sales']].agg(['sum']).plot.bar()
plt.title('Total Profit and Sales per Segment')
plt.rcParams['figure.figsize'] = [10,8]
plt.show()
#therefore we can deduce that both the sales and profit for the consumer segment is
```



In [182...

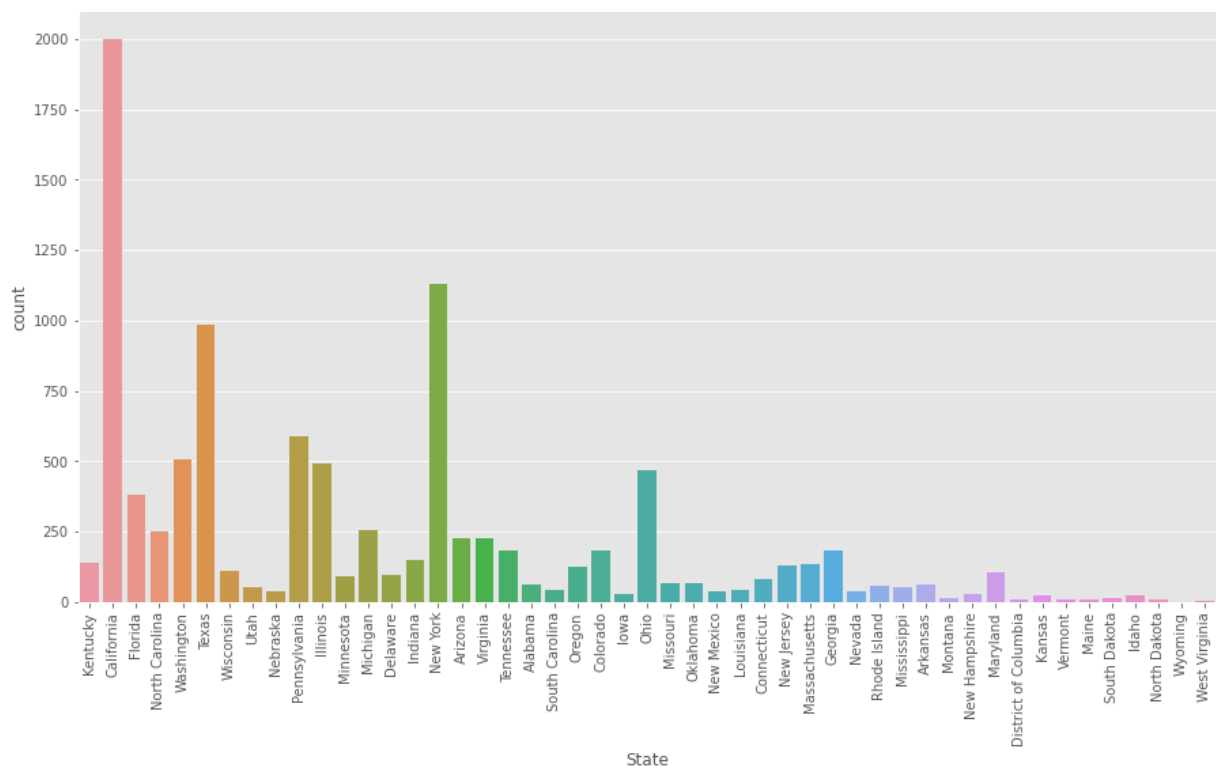
```
#We cannot deduce much from Segments as the subcategories are spread across more than
#Sub-Categories with reference to Segment
for i in orders['Segment'].unique():
    print(i, orders.loc[orders['Segment'] == i, 'Sub-Category'].unique())
#As we can see Copiers (which is the most profitable as shown above) belongs to all
```

```
Consumer ['Bookcases' 'Chairs' 'Tables' 'Storage' 'Furnishings' 'Art' 'Phones'
'Binders' 'Appliances' 'Paper' 'Accessories' 'Envelopes' 'Labels'
'Fasteners' 'Supplies' 'Machines' 'Copiers']
Corporate ['Labels' 'Art' 'Appliances' 'Phones' 'Furnishings' 'Storage'
'Accessories' 'Binders' 'Fasteners' 'Envelopes' 'Paper' 'Chairs'
'Bookcases' 'Machines' 'Tables' 'Supplies' 'Copiers']
Home Office ['Appliances' 'Binders' 'Paper' 'Envelopes' 'Bookcases' 'Chairs' 'Phones'
'Storage' 'Furnishings' 'Accessories' 'Art' 'Tables' 'Fasteners' 'Labels'
'Machines' 'Supplies' 'Copiers']
```

State Analysis

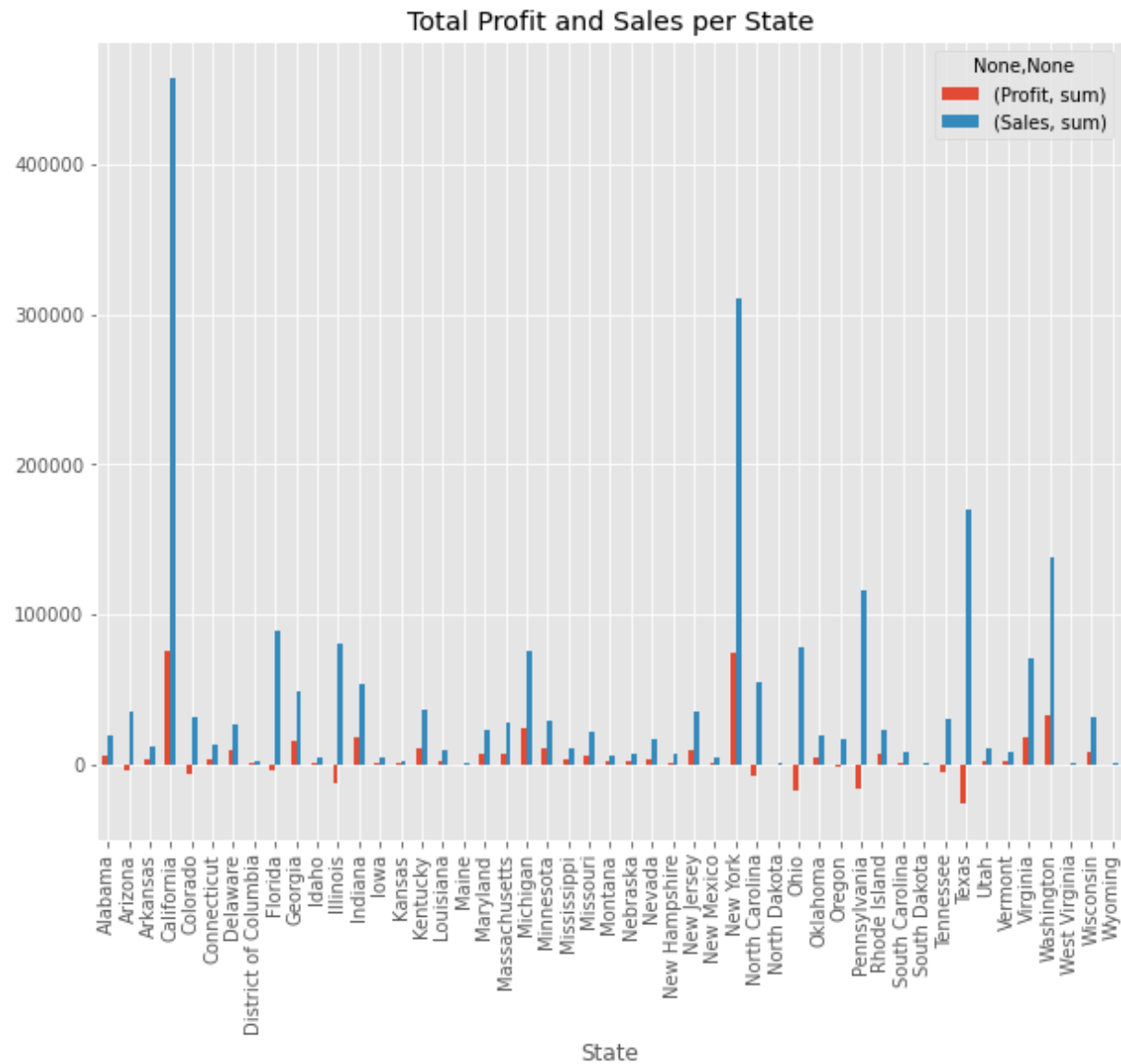
In [152...

```
plt.figure(figsize=(15,8))
sns.countplot(x=orders['State'])
plt.xticks(rotation=90)
plt.show()
#California and New York have significant number of orders compared to others
```



In [76]:

```
orders.groupby('State')[['Profit', 'Sales']].agg(['sum']).plot.bar()
plt.title('Total Profit and Sales per State')
plt.rcParams['figure.figsize'] = [10,8]
plt.show()
#We can see that California and New York have higher sales and approximately the same
```



City Analysis

In [153...

```
orders.groupby('City')['Sales'].sum().reset_index().sort_values(by='Sales', ascending=False)
```

#as for the cities we can see that New York City has the highest amount of sales fol

Out[153...

| | City | Sales |
|-----|---------------|------------|
| 329 | New York City | 256368.161 |
| 266 | Los Angeles | 175851.341 |
| 452 | Seattle | 119540.742 |
| 438 | San Francisco | 112669.092 |
| 374 | Philadelphia | 109077.013 |
| ... | ... | ... |
| 354 | Ormond Beach | 2.808 |
| 370 | Pensacola | 2.214 |
| 221 | Jupiter | 2.064 |
| 140 | Elyria | 1.824 |

| | City | Sales |
|---|---------|-------|
| 1 | Abilene | 1.392 |

531 rows × 2 columns

In [154...

```
orders.groupby('City')['Profit'].sum().reset_index().sort_values(by='Profit', ascend
#we can see the same for profit although Philadelphia seems to have incurred the hig
```

Out[154...

| | City | Profit |
|-----|---------------|-------------|
| 329 | New York City | 62036.9837 |
| 266 | Los Angeles | 30440.7579 |
| 452 | Seattle | 29156.0967 |
| 438 | San Francisco | 17507.3854 |
| 123 | Detroit | 13181.7908 |
| ... | ... | ... |
| 80 | Chicago | -6654.5688 |
| 241 | Lancaster | -7239.0684 |
| 434 | San Antonio | -7299.0502 |
| 207 | Houston | -10153.5485 |
| 374 | Philadelphia | -13837.7674 |

531 rows × 2 columns

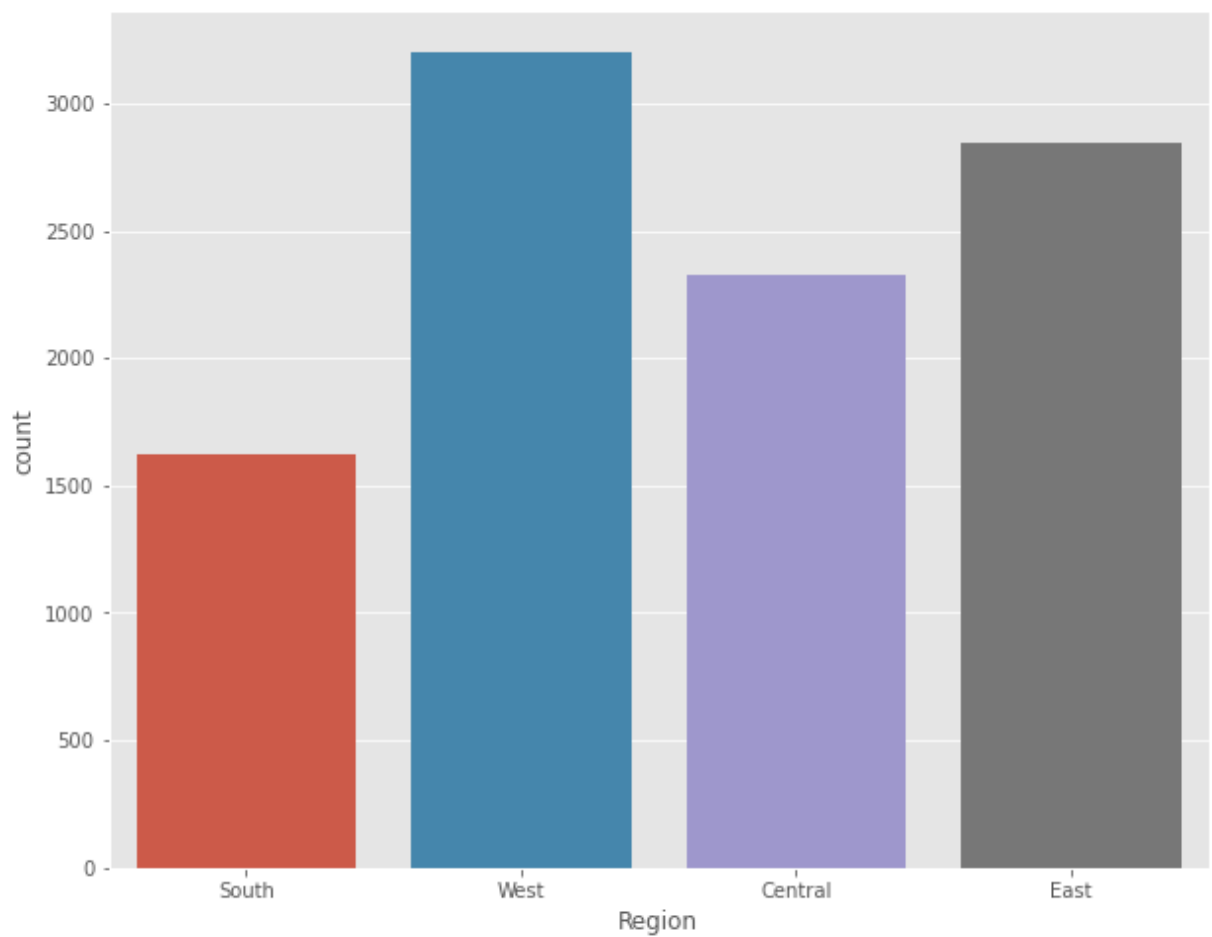
Region Analysis

In [156...

```
sns.countplot(x=orders['Region'])
#highest amount of orders are from the west
```

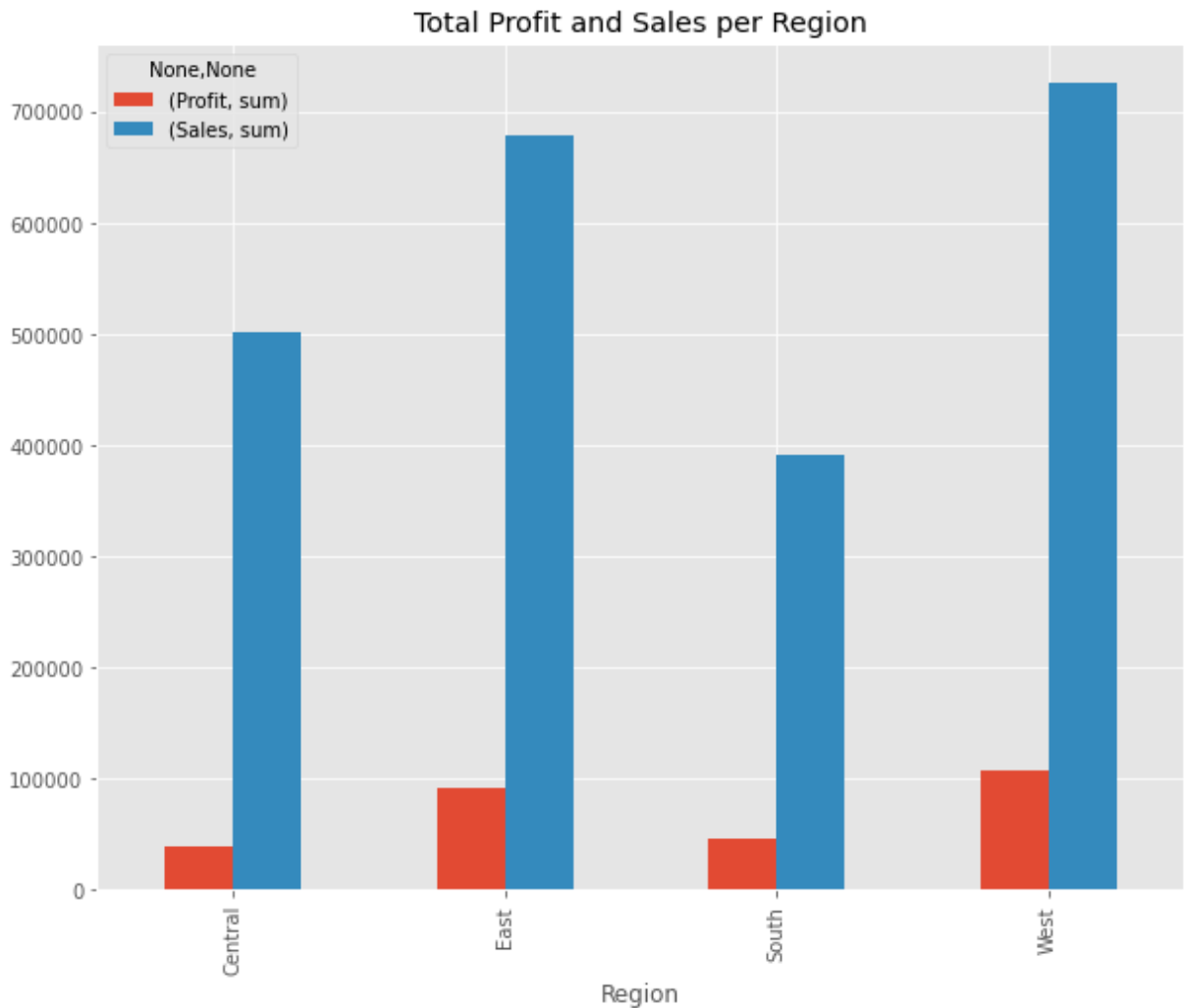
Out[156...

<AxesSubplot:xlabel='Region', ylabel='count'>



In [185...

```
orders.groupby('Region')[['Profit', 'Sales']].agg(['sum']).plot.bar()
plt.title('Total Profit and Sales per Region')
plt.rcParams['figure.figsize'] = [10,8]
plt.show()
#West and East have better Sales and Profit
```



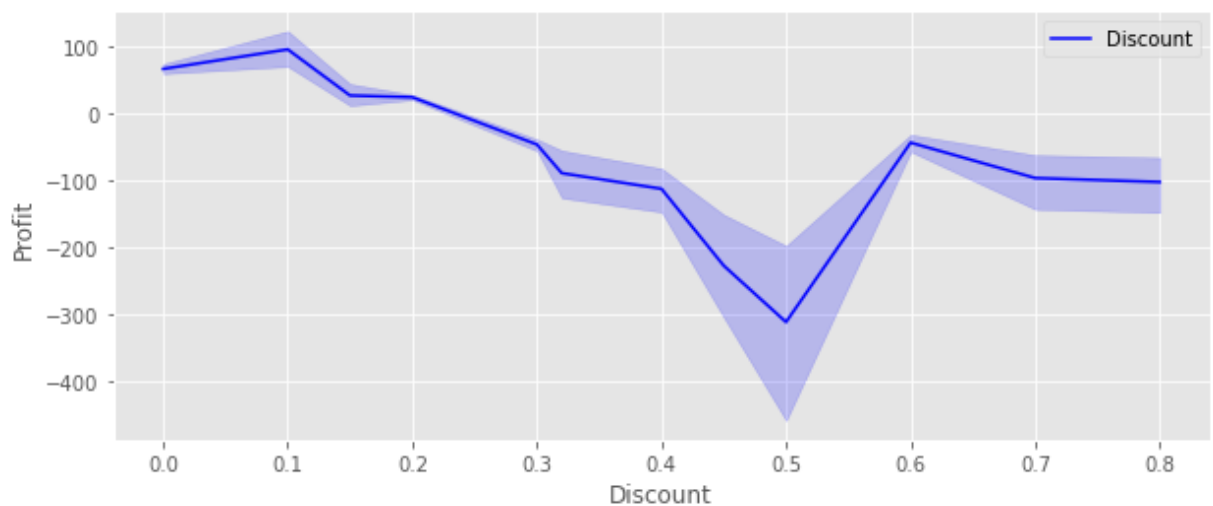
Discount Vs Profit

In [158...

```
plt.figure(figsize=(10,4))
sns.lineplot(x='Discount', y='Profit', data=orders, color='b', label='Discount')
plt.legend()
#we can see that upto 0.5 the profit declines with a little bit rise at 0.6
#so the highest amount of profit is made when no discount is given
#hence the negative correlation between discount and profit can be seen
```

Out[158...

<matplotlib.legend.Legend at 0x22e322700d0>



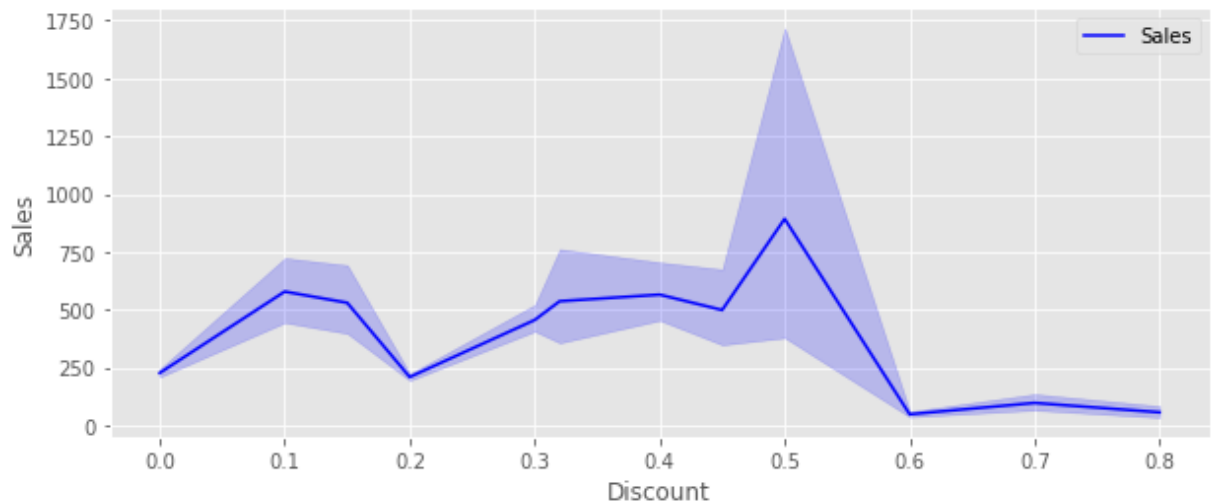
Discount Vs Sales

In [161...

```
plt.figure(figsize=(10,4))
sns.lineplot(x='Discount', y='Sales', data=orders, color='b', label='Sales')
plt.legend()
#although here we can observe an almost flipped graph
#deducing that sales increases due to discount
```

Out[161...

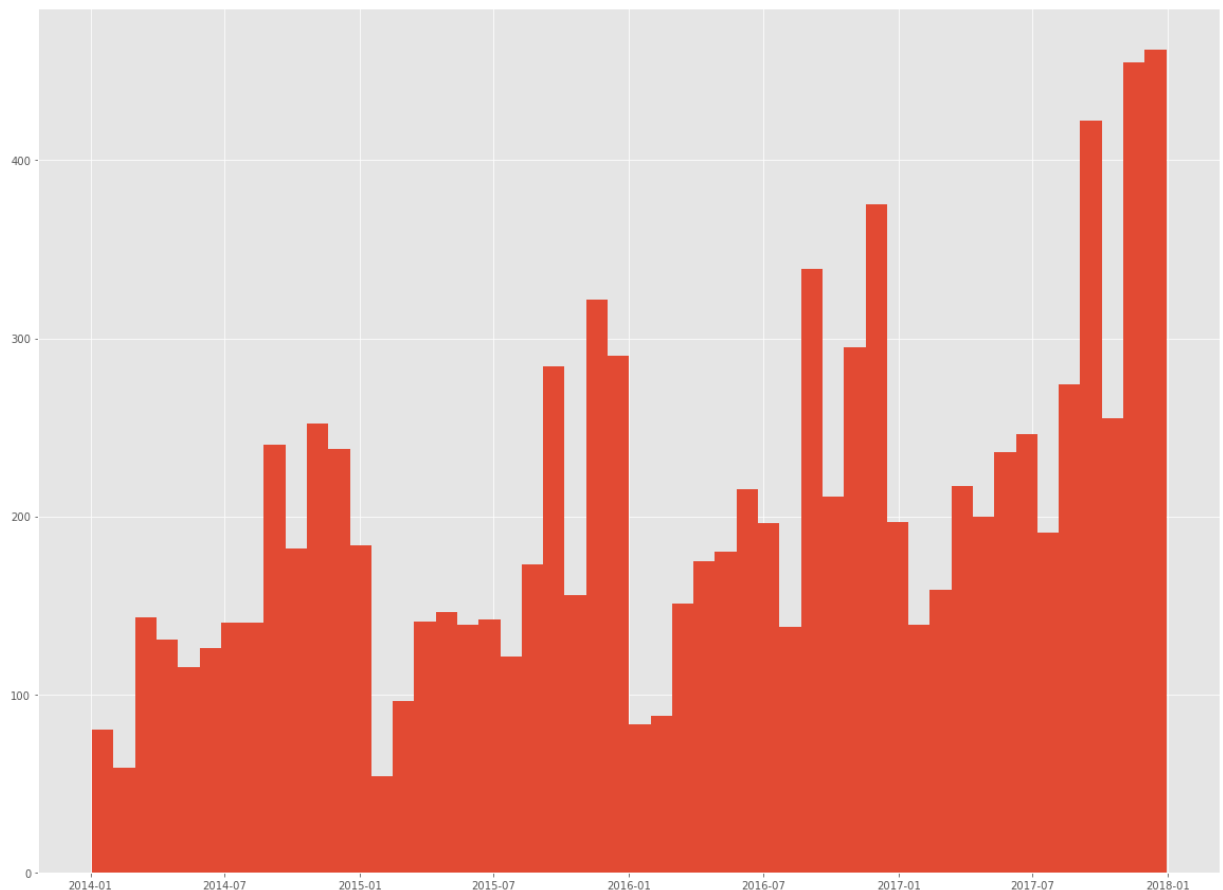
<matplotlib.legend.Legend at 0x22e327a1b20>



Distribution of Orders Over Time

In [165...

```
orders['Order Date'].hist(bins=50, figsize=(20,15))
plt.show()
```



Histogram of Data

In [183...

```
orders.hist(bins=50, figsize=(20,15))
plt.show()
```



Correlation among the Columns

In [166...

```
orders.corr()
```

Out[166...

| | Sales | Quantity | Discount | Profit |
|----------|-----------|----------|-----------|-----------|
| Sales | 1.000000 | 0.200808 | -0.028197 | 0.479067 |
| Quantity | 0.200808 | 1.000000 | 0.008680 | 0.066241 |
| Discount | -0.028197 | 0.008680 | 1.000000 | -0.219481 |
| Profit | 0.479067 | 0.066241 | -0.219481 | 1.000000 |

In [179...

```
fig, axes = plt.subplots(1, 1, figsize=(9, 6))
sns.heatmap(orders.corr(), annot=True)
plt.show()
#Profit and Discount are negatively correlated
```



In [178...

group = pd.DataFrame(orders.groupby(['Category', 'Sub-Category'])(['Quantity', 'Discount', 'Sales', 'Profit'])

Out[178...

| | Category | Sub-Category | Quantity | Discount | Sales | Profit |
|----|-----------------|--------------|----------|----------|-------------|-------------|
| 0 | Furniture | Bookcases | 868 | 48.14 | 114879.9963 | -3472.5560 |
| 1 | Furniture | Chairs | 2354 | 104.70 | 328167.7310 | 26602.2251 |
| 2 | Furniture | Furnishings | 3563 | 132.40 | 91705.1640 | 13059.1436 |
| 3 | Furniture | Tables | 1241 | 83.35 | 206965.5320 | -17725.4811 |
| 4 | Office Supplies | Appliances | 1729 | 77.60 | 107532.1610 | 18138.0054 |
| 5 | Office Supplies | Art | 3000 | 59.60 | 27118.7920 | 6527.7870 |
| 6 | Office Supplies | Binders | 5974 | 567.00 | 203412.7330 | 30221.7633 |
| 7 | Office Supplies | Envelopes | 906 | 20.40 | 16476.4020 | 6964.1767 |
| 8 | Office Supplies | Fasteners | 914 | 17.80 | 3024.2800 | 949.5182 |
| 9 | Office Supplies | Labels | 1400 | 25.00 | 12486.3120 | 5546.2540 |
| 10 | Office Supplies | Paper | 5178 | 102.60 | 78479.2060 | 34053.5693 |
| 11 | Office Supplies | Storage | 3158 | 63.20 | 223843.6080 | 21278.8264 |
| 12 | Office Supplies | Supplies | 647 | 14.60 | 46673.5380 | -1189.0995 |
| 13 | Technology | Accessories | 2976 | 60.80 | 167380.3180 | 41936.6357 |
| 14 | Technology | Copiers | 234 | 11.00 | 149528.0300 | 55617.8249 |
| 15 | Technology | Machines | 440 | 35.20 | 189238.6310 | 3384.7569 |
| 16 | Technology | Phones | 3289 | 137.40 | 330007.0540 | 44515.7306 |

In [177...

group1 = pd.DataFrame(orders.groupby(['State', 'City', 'Segment', 'Region'])(['Quantity', 'Discount', 'Sales', 'Profit'])

Out[177...

| | State | City | Segment | Region | Quantity | Discount | Sales | Profit |
|------|-----------|------------|-------------|---------|----------|----------|----------|----------|
| 0 | Alabama | Auburn | Consumer | South | 14 | 0.0 | 1763.070 | 451.5167 |
| 1 | Alabama | Auburn | Home Office | South | 2 | 0.0 | 3.760 | 1.0904 |
| 2 | Alabama | Decatur | Consumer | South | 14 | 0.0 | 435.790 | 77.4454 |
| 3 | Alabama | Decatur | Corporate | South | 38 | 0.0 | 2939.030 | 922.6427 |
| 4 | Alabama | Florence | Consumer | South | 23 | 0.0 | 1992.370 | 246.2187 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1170 | Wisconsin | Superior | Home Office | Central | 37 | 0.0 | 1299.730 | 514.0822 |
| 1171 | Wisconsin | Waukesha | Consumer | Central | 5 | 0.0 | 54.500 | 14.1700 |
| 1172 | Wisconsin | Wausau | Consumer | Central | 14 | 0.0 | 317.480 | 90.4306 |
| 1173 | Wisconsin | West Allis | Corporate | Central | 5 | 0.0 | 250.480 | 28.3708 |
| 1174 | Wyoming | Cheyenne | Home Office | West | 4 | 0.2 | 1603.136 | 100.1960 |

1175 rows × 8 columns

Conclusion

- The Category Technology is the most profitable and has the most sales whereas the profit in Furniture Category seems to be lacking.
- The most profitable among the Sub-Category is Copiers although Sales are comparatively not that high whereas for accessories and phones (Technology Category) we can see higher sales along with high profit.
- Also, the sales of tables and bookcases are comparatively more but we can see that we are dealing with overall loss.
- Consumer segment has the highest profit and sales although we cannot deduce much from Segments as the subcategories are spread across more than one segment.
- As for States, California and New York have higher sales and approximately the same amount of profit whereas Texas and Ohio has significant overall loss.
- For Cities, New York City has the highest amount of sales and overall profit followed by Los Angeles (a major reason is their higher population). Philadelphia seems to have incurred the highest amount of overall loss.
- The Region West and East have better Sales and Profit.
- Discount and Profit have a negative Correlation.

In []: