



FINAL PROJECT OF THE BUSINESS INTELLIGENCE COURSE

Natalia Makovska



■ SQL

■ Python

■ Tableau

■ Power BI

Покажите среднюю зарплату сотрудников за каждый год.

```
select year(sl.from_date) as years,  
       round(avg(sl.salary)) as avg_emp_sal  
  from salaries as sl  
 group by year(sl.from_date)  
order by 1;
```

years	avg_emp_sal
1985	53182
1986	54085
1987	54960
1988	55862
1989	56841
1990	57839
1991	58804
1992	59759
1993	60754
1994	61728
1995	62681
1996	63619
1997	64565
1998	65540
1999	66525
2000	68556
2001	70695
2002	72684
2021	50000

Покажите среднюю зарплату сотрудников по каждому отделу.

Примечание: возьмите текущие отделы и текущую зарплату.

```
select de.dept_no,
       round(avg(sl.salary)) as avg_emp_sal
  from salaries as sl
 inner join dept_emp as de
    on sl.emp_no = de.emp_no
   where sl.to_date >= now()
     and de.to_date >= now()
 group by de.dept_no
 order by 1;
```

	dept_no	avg_emp_sal
▶	d001	80059
	d002	78560
	d003	63922
	d004	67843
	d005	67658
	d006	65442
	d007	88853
	d008	67913
	d009	67285

Покажите среднюю зарплату сотрудников по каждому отделу за каждый год.

Примечание: для средней зарплаты отдела X в году Y нам нужно взять среднее значение всех зарплат в году Y сотрудников, которые были в отделе X в году Y

```
select year(sl.to_date) as years,  
       de.dept_no,  
       round(avg(sl.salary)) as avg_emp_sal  
  from salaries as sl  
inner join dept_emp as de  
    on sl.emp_no = de.emp_no  
group by de.dept_no, year(sl.to_date)  
order by 1;
```

	years	dept_no	avg_emp_sal
▶	1985	d001	56748
	1985	d002	57369
	1985	d003	40000
	1985	d004	44326
	1985	d005	46416
	1985	d006	47720
	1985	d007	62793
	1985	d008	51367
	1985	d009	54916
	1986	d001	61673
	1986	d002	60330
	1986	d003	44970
	1986	d004	48944
	1986	d005	48728
	1986	d006	46608
	1986	d007	69987
	1986	d008	49049
	1986	d009	48106
	1987	d001	62396
	1987	d002	61055
	1987	d003	46029
	1987	d004	49890
	1987	d005	49682
	1987	d006	47463
	1987	d007	70942
	1987	d008	50038

Покажите для каждого года самый крупный отдел в этом году и его среднюю зарплату.

```
select t1.*,
       first_value(t1.cnt_emp) over(partition by t1.years order by t1.cnt_emp desc) as largest_numb_emp,
       first_value(t1.dept_name) over(partition by t1.years order by t1.cnt_emp desc) as largest_department
  from (select year(de.from_date) as years,
              de.dept_no,
              dp.dept_name,
              count(de.emp_no) as cnt_emp,
              round(avg(sl.salary)) as avg_sal
        from dept_emp as de
        inner join salaries as sl
        on de.emp_no = sl.emp_no
        inner join departments as dp
        on de.dept_no = dp.dept_no
       group by year(de.from_date), de.dept_no) as t1
 order by 1;
```

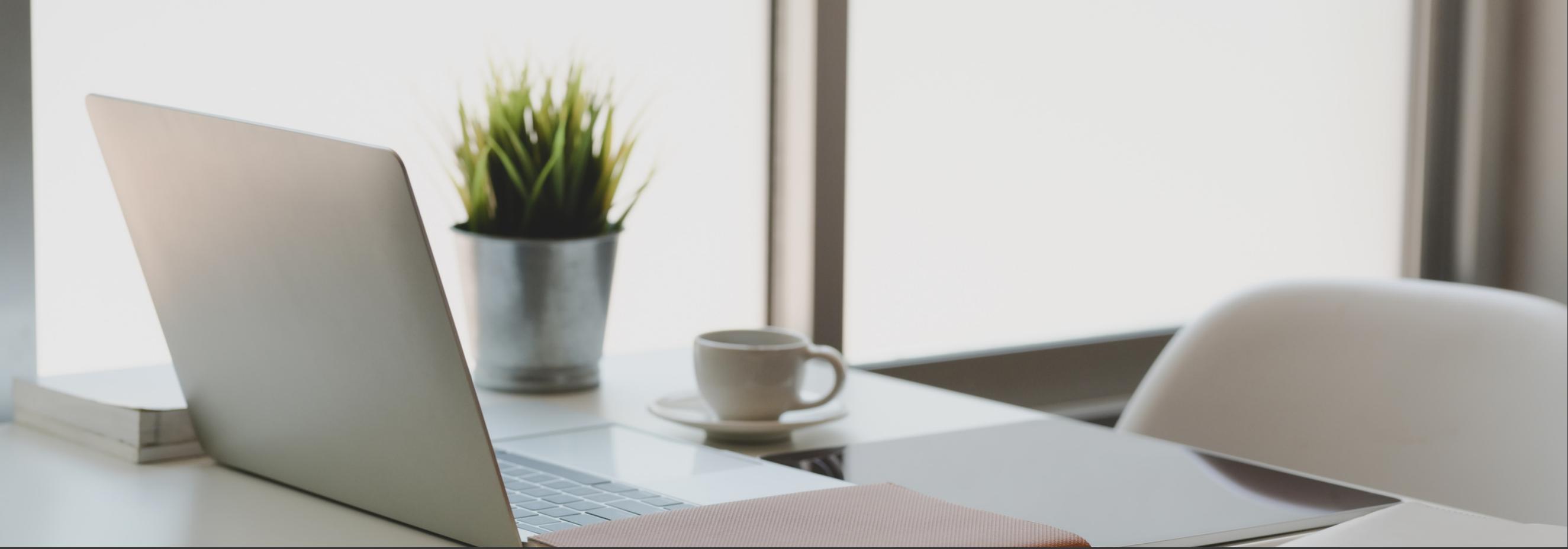
	years	dept_no	dept_name	cnt_emp	avg_sal	largest_numb_emp	largest_department
	1985	d006	Quality Management	15414	62001	80088	Development
	1986	d005	Development	81279	63905	81279	Development
	1986	d004	Production	66998	63980	81279	Development
	1986	d007	Sales	49793	85210	81279	Development
	1986	d009	Customer Service	17912	59441	81279	Development
	1986	d008	Research	17572	63977	81279	Development
	1986	d006	Quality Management	16693	61165	81279	Development
	1986	d003	Human Resources	16645	59607	81279	Development
	1986	d001	Marketing	16361	75836	81279	Development
►	1986	d002	Finance	16358	75531	81279	Development
	1987	d005	Development	78299	62727	78299	Development
	1987	d004	Production	63192	62722	78299	Development
	1987	d007	Sales	47276	84096	78299	Development
	1987	d009	Customer Service	17279	58788	78299	Development
	1987	d006	Quality Management	16764	59697	78299	Development
	1987	d003	Human Resources	16224	57948	78299	Development
	1987	d008	Research	16062	63189	78299	Development
	1987	d002	Finance	15553	74148	78299	Development
	1987	d001	Marketing	15354	74237	78299	Development
	1988	d005	Development	72037	61771	72037	Development
	1988	d004	Production	60339	61800	72037	Development
	1988	d007	Sales	44809	82965	72037	Development
	1988	d009	Customer Service	16489	57918	72037	Development
	1988	d006	Quality Management	15947	58909	72037	Development
	1988	d008	Research	15799	61034	72037	Development

Покажите подробную информацию о текущем менеджере, который дольше всех выполняет свои обязанности.

```
select em.emp_no,
       em.birth_date,
       em.first_name,
       em.last_name,
       em.gender,
       em.hire_date
  from employees as em
 where em.emp_no = (select dm.emp_no
                     from dept_manager as dm
                    where dm.from_date = (select min(dm.from_date)
                     from dept_manager as dm
                    where dm.to_date >= now()));
```

	emp_no	birth_date	first_name	last_name	gender	hire_date
▶	110114	1957-03-28	Isamu	Legleitner	F	1985-01-14
●	NULL	NULL	NULL	NULL	NULL	NULL

Python



1. Загрузите базу данных мебели ИКЕА.
2. Выполните исследовательский анализ данных для набора данных, включая описательную статистику и визуализацию. Опишите результаты.
3. Основываясь на EDA и здравом смысле, выберите две гипотезы, которые вы хотите проверить / проанализировать. Для каждой гипотезы перечислите нулевую гипотезу и другие возможные альтернативные гипотезы, разработайте тесты, чтобы различать их, и выполните их. Опишите результаты.
4. Обучите модель предсказывать цену на мебель.
 - Посоветуйте, какие столбцы не следует включать в модель и почему.
 - Создайте пайпайн кросс-валидации для обучения и оценки модели,включая (при необходимости) такие шаги, как заполнение пропущенных значений и нормализация.
 - Предложите методы повышения производительности модели.
 - Опишите результаты.

```
Ввод [5]: 1 import numpy as np  
2 import pandas as pd  
3 import matplotlib.pyplot as plt  
4 import seaborn as sns  
5 from sklearn.preprocessing import LabelEncoder
```

```
Ввод [6]: 1 link = "https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-11-03/ikea.csv"
```

```
Ввод [7]: 1 ikea_df = pd.read_csv(link)  
2 ikea_df.head()
```

Out[7]:

#	name	category	price	old_price	sellable_online	link	other_colors	short_description	designer	depth	height	width
2	FREKVEN	Bar furniture	265.0	No old price	True	https://www.ikea.com/sa/en/p/frekvens-bar-tabl...	No	Bar table, in/outdoor, 51x51 cm	Nicholai Wiig Hansen	NaN	99.0	51.0
4	NORDVIKEN	Bar furniture	995.0	No old price	False	https://www.ikea.com/sa/en/p/nordviken-bar-tab...	No	Bar table, 140x80 cm	Francis Cayouette	NaN	105.0	80.0
3	NORDVIKEN / NORDVIKEN	Bar furniture	2095.0	No old price	False	https://www.ikea.com/sa/en/p/nordviken-nordvik...	No	Bar table and 4 bar stools	Francis Cayouette	NaN	NaN	NaN
5	STIG	Bar furniture	69.0	No old price	True	https://www.ikea.com/sa/en/p/stig-bar-stool-wi...	Yes	Bar stool with backrest, 74 cm	Henrik Preutz	50.0	100.0	60.0
4	NORBERG	Bar furniture	225.0	No old price	True	https://www.ikea.com/sa/en/p/norberg-wall-moun...	No	Wall-mounted drop-leaf table, ...	Marcus Arvonen	60.0	43.0	74.0

```
1 ikea_df.describe()
```

	Unnamed: 0	item_id	price	depth	height	width
count	3694.000000	3.694000e+03	3694.000000	2231.000000	2706.000000	3105.000000
mean	1846.500000	4.863240e+07	1078.208419	54.379202	101.679970	104.470853
std	1066.510275	2.888709e+07	1374.652494	29.958351	61.097585	71.133771
min	0.000000	5.848700e+04	3.000000	1.000000	1.000000	1.000000
25%	923.250000	2.039057e+07	180.900000	38.000000	67.000000	60.000000
50%	1846.500000	4.928808e+07	544.700000	47.000000	83.000000	80.000000
75%	2769.750000	7.040357e+07	1429.500000	60.000000	124.000000	140.000000
max	3693.000000	9.993262e+07	9585.000000	257.000000	700.000000	420.000000

Аналіз кількості номенклатури за категоріями.

За результатами аналізу ми можемо побачити розподіл номенклатури за категоріями.

Визначити категорії з високим і низьким попитом

```
1 name_count=len(ikea_df_new['name'].unique())
2 print('База даних містить', name_count, 'найменувань')
```

База даних містить 607 найменувань

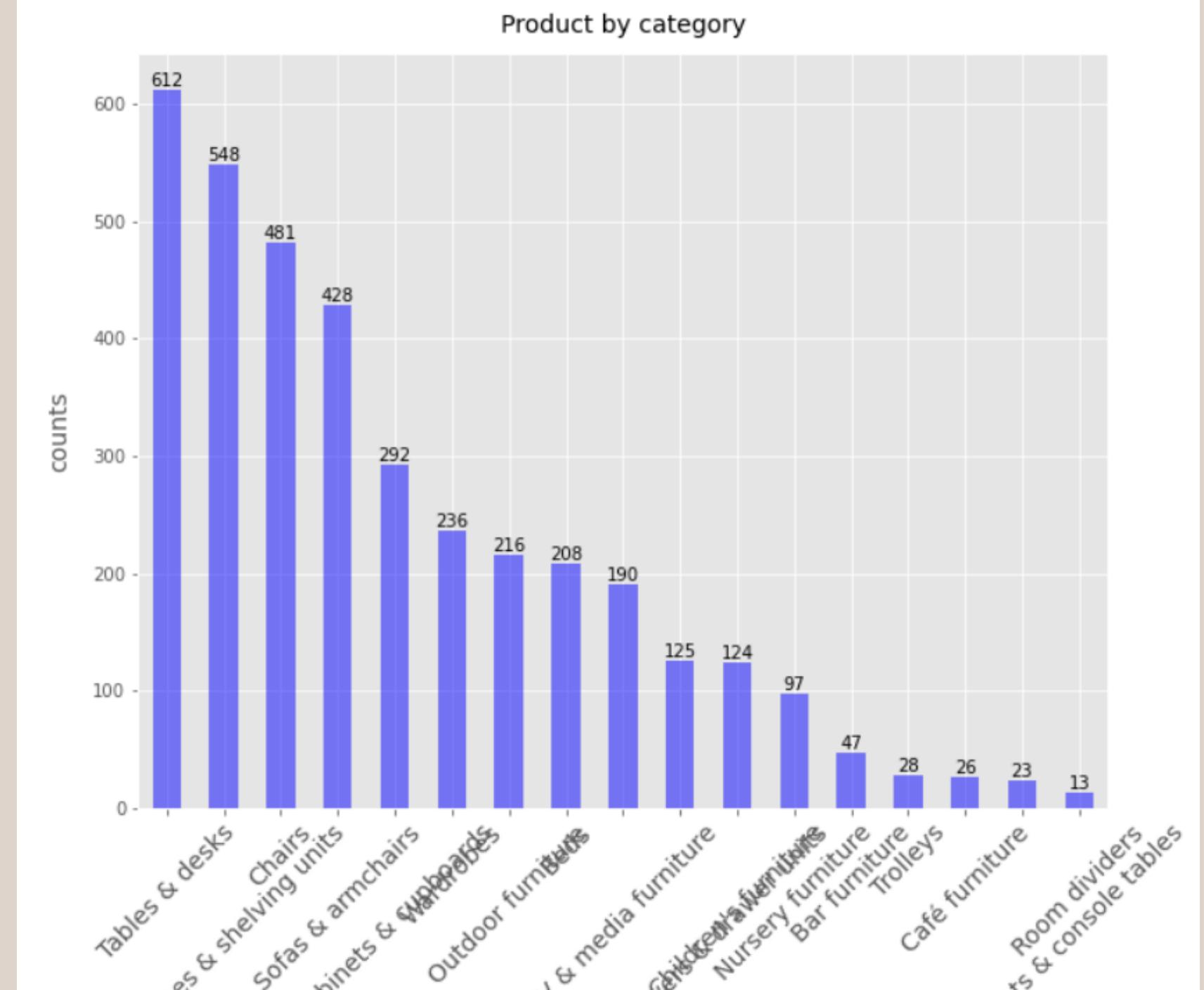
```
1 category_count=len(ikea_df_new['category'].unique())
2 print('База даних містить', category_count, 'категорій')
```

База даних містить 17 категорій

```
1 product_by_category = ikea_df_new['category'].value_counts()
2 product_by_category
```

category	counts
Tables & desks	612
Bookcases & shelving units	548
Chairs	481
Sofas & armchairs	428
Cabinets & cupboards	292
Wardrobes	236
Outdoor furniture	216
Beds	208
TV & media furniture	190
Chests of drawers & drawer units	125
Children's furniture	124
Nursery furniture	97
Bar furniture	47
Trolleys	28
Café furniture	26
Sideboards, buffets & console tables	23
Room dividers	13
Name: category, dtype: int64	

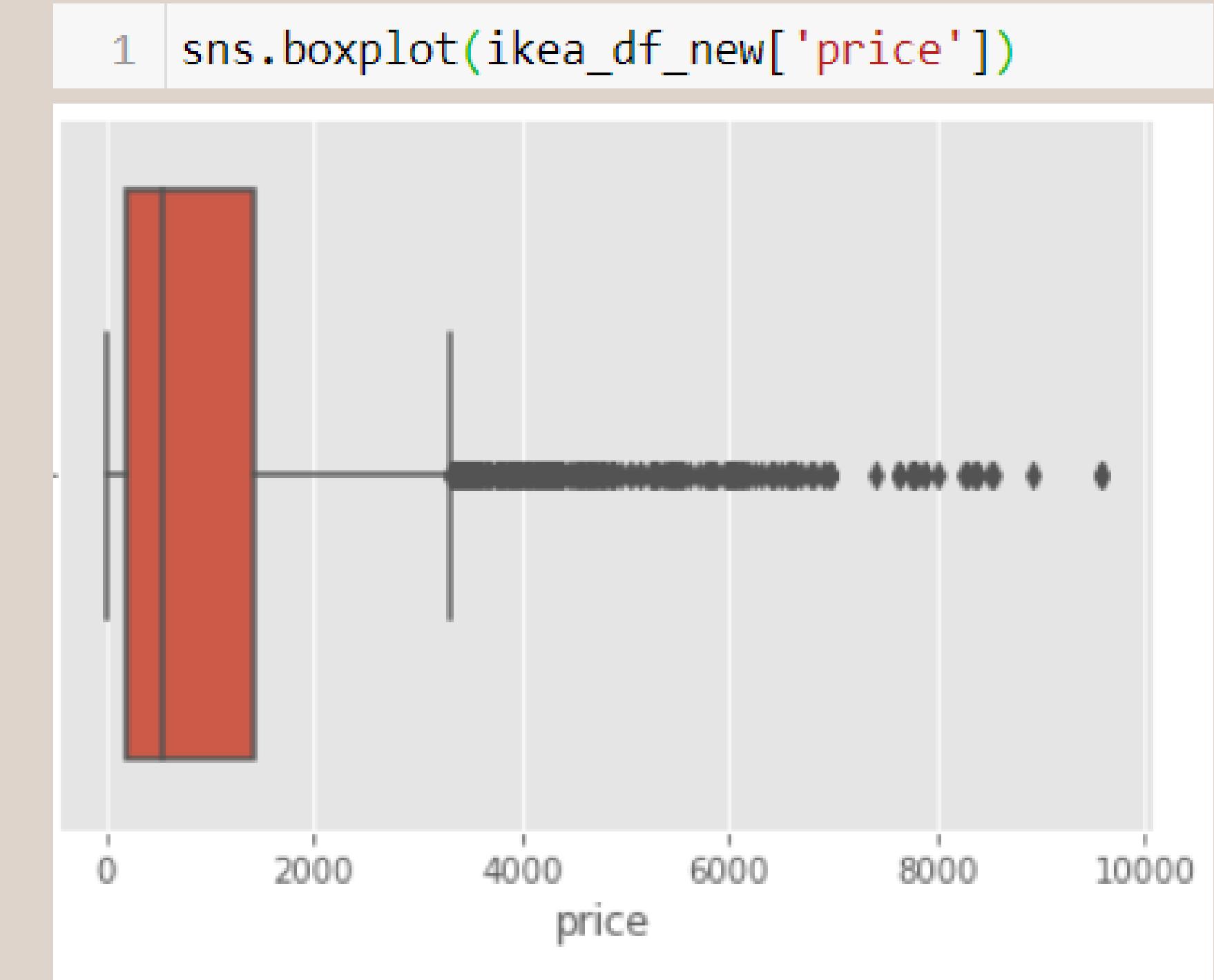
```
1 plt.style.use('ggplot')
2 product_by_category.plot(kind='bar', figsize=(10,8), alpha=0.5, rot=45, color='b')
3 plt.xticks(fontsize=14)
4 plt.xlabel('name', fontsize=14, labelpad=12)
5 plt.ylabel('counts', fontsize=14, labelpad=12)
6 plt.title('Product by category', fontsize=14, pad=12)
7
8 for i,j in zip(range(len(product_by_category)),product_by_category):
9     plt.text(i,j+0.5,j,ha='center',va='bottom',fontsize=10)
```



Співвідношення кількості продажів online (True) i offline(False).



Розподіл значення Price.



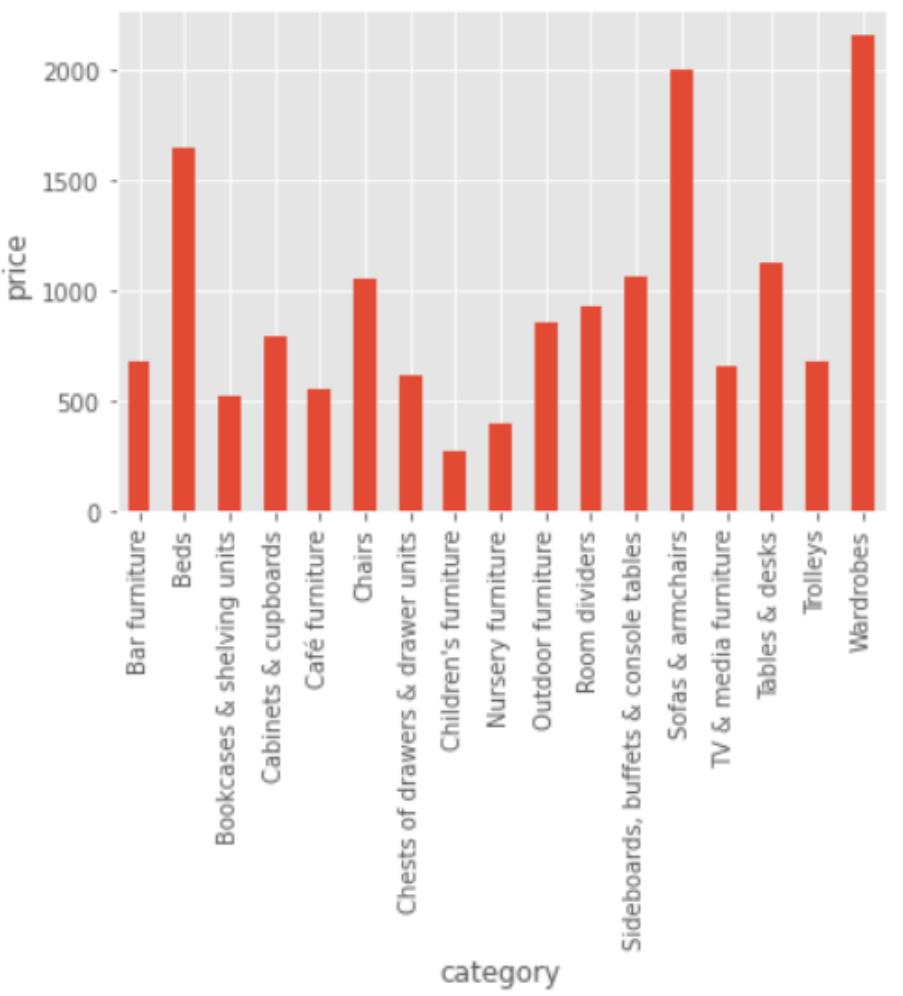
Аналіз цін за категоріями.

Максимальна, мінімальна, середня ціна та медіану за категоріями.

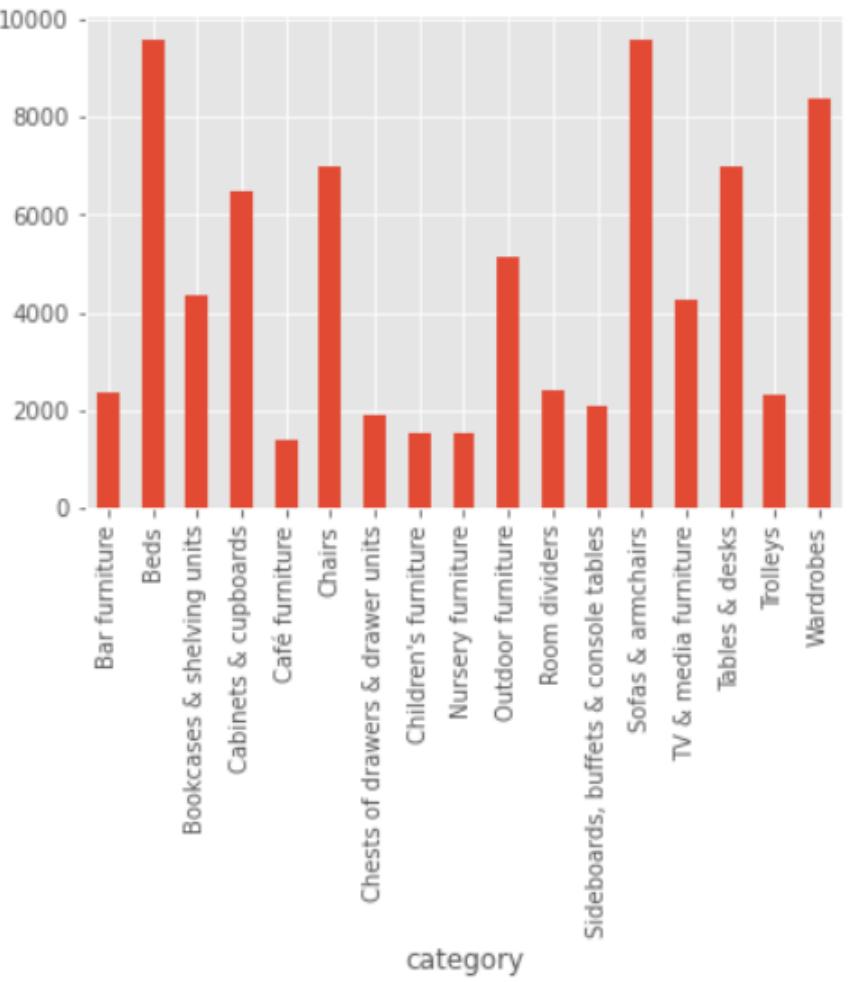
```
1 ikea_df_group = ikea_df_new.groupby(['category']).agg(max_price=('price','max'),  
2                                         min_price=('price', 'min'),  
3                                         mean_price=('price', 'mean'),  
4                                         median_price=('price', 'median'))  
5 ikea_df_group
```

category	max_price	min_price	mean_price	median_price
Bar furniture	2375.0	40.0	679.553191	445.0
Beds	9585.0	30.0	1647.432692	1093.5
Bookcases & shelving units	4335.0	3.0	519.416058	310.0
Cabinets & cupboards	6480.0	3.0	789.015753	402.0
Café furniture	1385.0	145.0	553.884615	424.5
Chairs	6965.0	11.4	1048.932640	425.0
Chests of drawers & drawer units	1880.0	49.0	612.180000	460.0
Children's furniture	1545.0	5.0	272.758065	202.5
Nursery furniture	1545.0	5.0	393.659794	350.0
Outdoor furniture	5144.0	9.0	859.826852	313.7
Room dividers	2397.0	189.0	932.000000	725.0
Sideboards, buffets & console tables	2100.0	295.0	1068.086957	965.0
Sofas & armchairs	9585.0	27.0	2004.016355	1268.5
TV & media furniture	4260.0	3.0	659.050526	175.0
Tables & desks	6965.0	9.0	1129.395425	595.0
Trolleys	2295.0	39.0	678.517857	447.5
Wardrobes	8380.0	45.0	2162.141949	1870.0

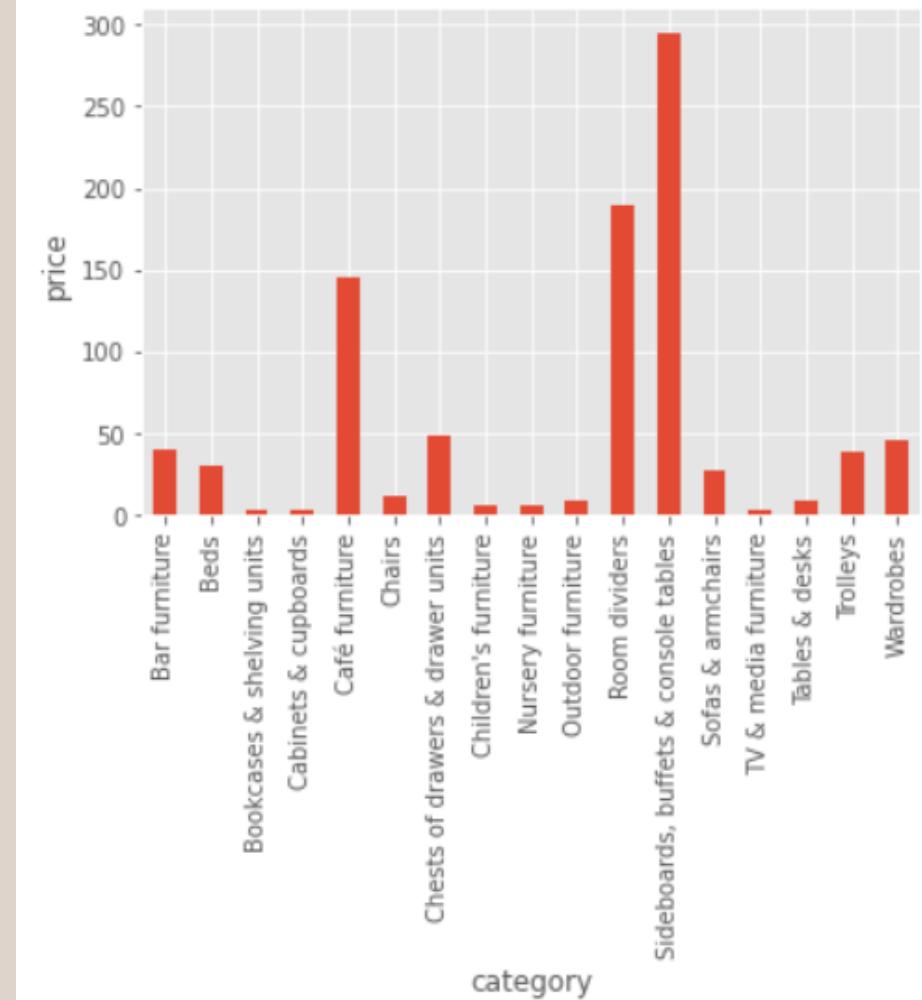
```
1 ikea_df_new.groupby('category')['price'].mean().plot(kind='bar')
2 plt.ylabel('price')
3 plt.show()
```



```
1 ikea_df_new.groupby('category')['price'].max().plot(kind='bar')
2 plt.ylabel('price')
3 plt.show()
```



```
1 ikea_df_new.groupby('category')['price'].min().plot(kind='bar')
2 plt.ylabel('price')
3 plt.show()
```

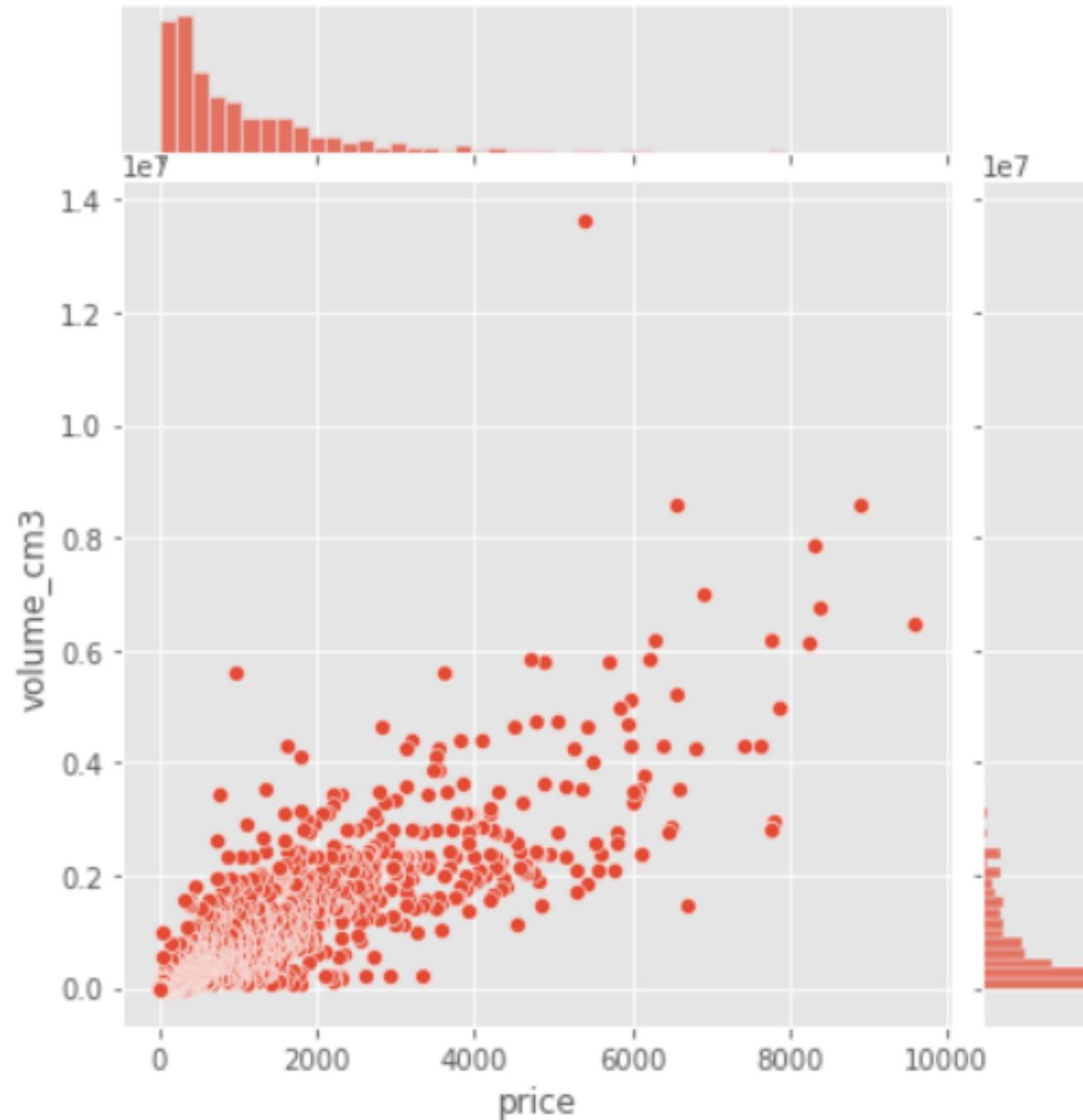


За допомогою jointplot можемо побачити чи є залежність між розміром (об'ємом) виробу та його ціною.

Попередньо необхідно видалити із датасету значення, які містять NaN.

```
1 sns.jointplot(x='price', y='volume_cm3', data=droped_ikea_df)
```

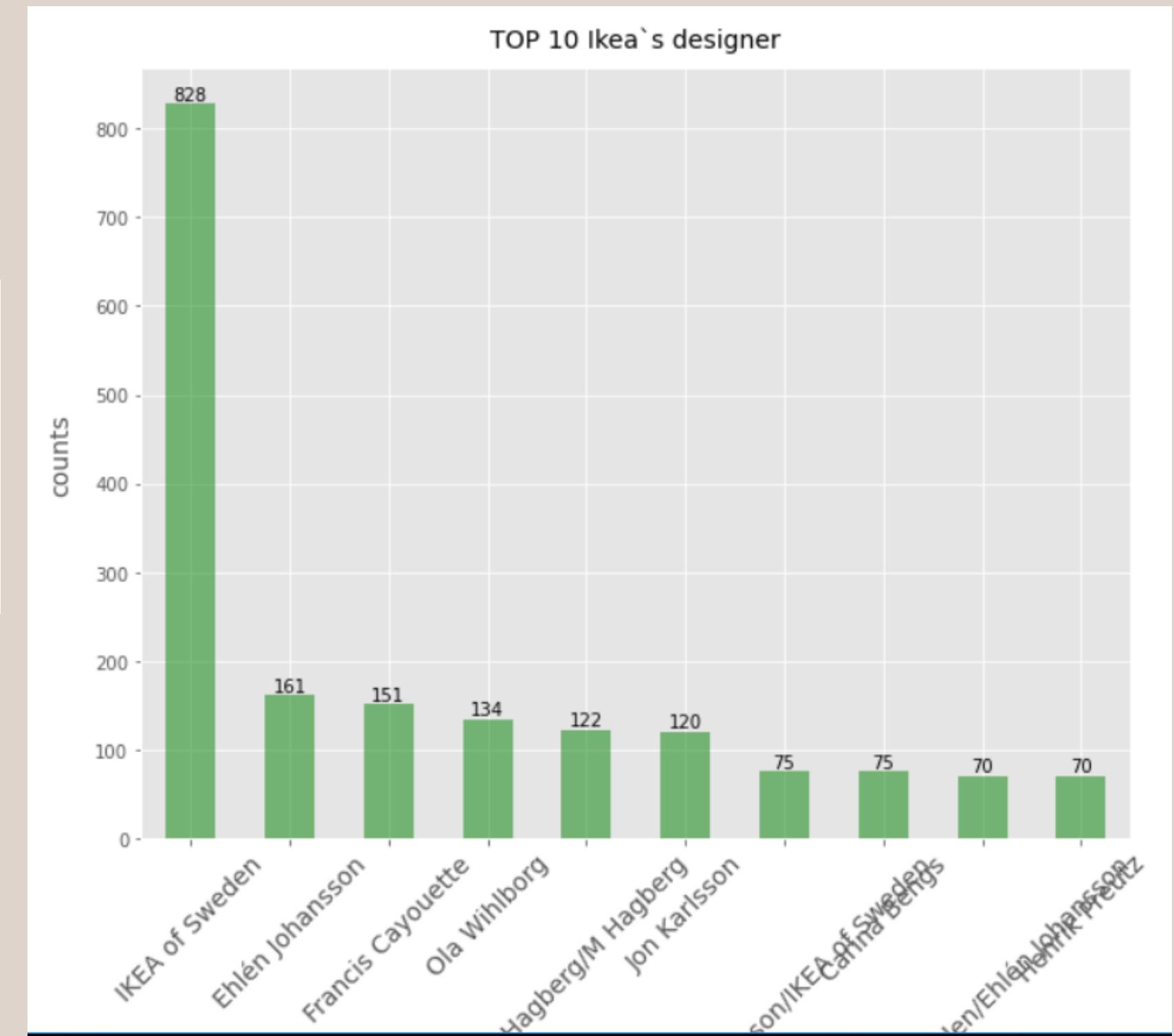
```
<seaborn.axisgrid.JointGrid at 0x293b4020580>
```



Як бачимо, в основному розмір товару (об'єм) значно не впливає на ціноутворення. Є деякі виключення.

TOP-10 дизайнерів Ikea.

```
1 plt.style.use('ggplot')
2 designer_TOP_10.plot(kind='bar',figsize=(10,8),alpha=0.5,rot=45,color='g')
3 plt.xticks(fontsize=14)
4 plt.xlabel('name',fontsize=14,labelpad=12)
5 plt.ylabel('counts',fontsize=14,labelpad=12)
6 plt.title('TOP 10 Ikea`s designer',fontsize=14,pad=12)
7
8 for i,j in zip(range(len(designer_TOP_10)),designer_TOP_10):
9     plt.text(i,j+0.5,j,ha='center',va='bottom',fontsize=10)
```



Тестування взаємозв'язку ціни та категорії, ціни та розміру товару

```
1 x = droped_ikea_df[['category']].values
2 x
array([['Sofas & armchairs'],
       ['Beds'],
       ['Sofas & armchairs'],
       ...,
       ['Nursery furniture'],
       ['Tables & desks'],
       ['Bookcases & shelving units']], dtype=object)

1 le = LabelEncoder()
2
3 for i in range(len(x[0])):
4     x[:, i] = le.fit_transform(x[:, i])
5
6 x
array([[12],
       [1],
       [12],
       ...,
       [8],
       [14],
       [2]], dtype=object)
```

```
1 y = droped_ikea_df[['price']]
2 y
   price
2344 9585.0
190 9585.0
2559 8900.0
2387 8395.0
2289 8295.0
...
325 15.0
1839 10.0
1946 10.0
2718 10.0
461 6.0
1899 rows × 1 columns
1 from scipy.stats import f_oneway
2 f_oneway(x, y).pvalue
array([2.80063806e-258])
```

```
1 a = droped_ikea_df[['volume_cm3']]
2 a
   volume_cm3
2344 6473736.0
190 6473736.0
2559 8570016.0
2387 6758109.0
2289 7849808.0
...
325 19125.0
1839 12600.0
1946 12600.0
2718 5625.0
461 40.0
1899 rows × 1 columns
```

```
1 b = droped_ikea_df[['price']]
2 b
   price
2344 9585.0
190 9585.0
2559 8900.0
2387 8395.0
2289 8295.0
...
325 15.0
1839 10.0
1946 10.0
2718 10.0
461 6.0
1899 rows × 1 columns
1 from scipy.stats import f_oneway
2 f_oneway(b, a).pvalue
array([7.34438282e-241])
```

Прогнозування ціни

```
1 from sklearn.pipeline import Pipeline
2 from sklearn.impute import SimpleImputer
3 from sklearn.preprocessing import OneHotEncoder, StandardScaler
4 from sklearn.neighbors import KNeighborsClassifier, KNeighborsRegressor
5 from sklearn.model_selection import train_test_split
6 from sklearn.metrics import accuracy_score, mean_squared_error
7 from sklearn.compose import ColumnTransformer
8 from sklearn.tree import DecisionTreeRegressor
```



```
1 X = droped_ikea_df[['depth', 'width', 'height', 'category']]
2 y = droped_ikea_df['price']
```



```
1 numeric_pipeline = Pipeline(steps=[
2     ('impute', SimpleImputer(strategy='most_frequent')),
3     ('scale', StandardScaler())])
4
5 categorical_pipeline = Pipeline(steps=[
6     ('impute', SimpleImputer(strategy='most_frequent')),
7     ('onehot', OneHotEncoder(handle_unknown = 'ignore'))])
8
9 preprocessing = ColumnTransformer(transformers=[
10    ('numeric_pipeline', numeric_pipeline, ['depth', 'width', 'height']),
11    ('categorical_pipeline', categorical_pipeline, ['category'])])
12
13 model = Pipeline(steps=[
14     ('preprocessing', preprocessing),
15     ('regression', DecisionTreeRegressor()),])
16
17 results = []
18
19 for _ in range(20):
20     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=.1)
21     model.fit(X_train, y_train)
22
23     model.score(X_test, y_test)
24     y_predict = model.predict(X_test)
25     print(mean_squared_error(y_train, model.predict(X_train)))
26     result = mean_squared_error(y_test, y_predict)
27     results.append(result)
28
29 print('Mean:', np.mean(results))
```

55484.4709568006
60597.90168700716
59780.88785477444
44597.538611490905
62685.48866163094
58349.07140153292
60565.64696582139
47905.75234422297
49395.09282217392
60007.93632431882
48897.12150828107
57764.10585698386
60002.4315098048
62519.36182986694
57631.5040028725
62067.983246422846
60679.46131976317
49004.21189027278
59461.065182038554
59234.32940069844
Mean: 407134.7876878847

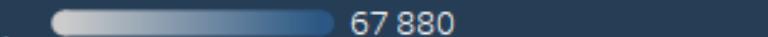
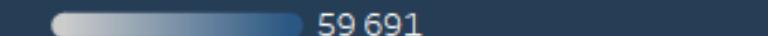
Tableau



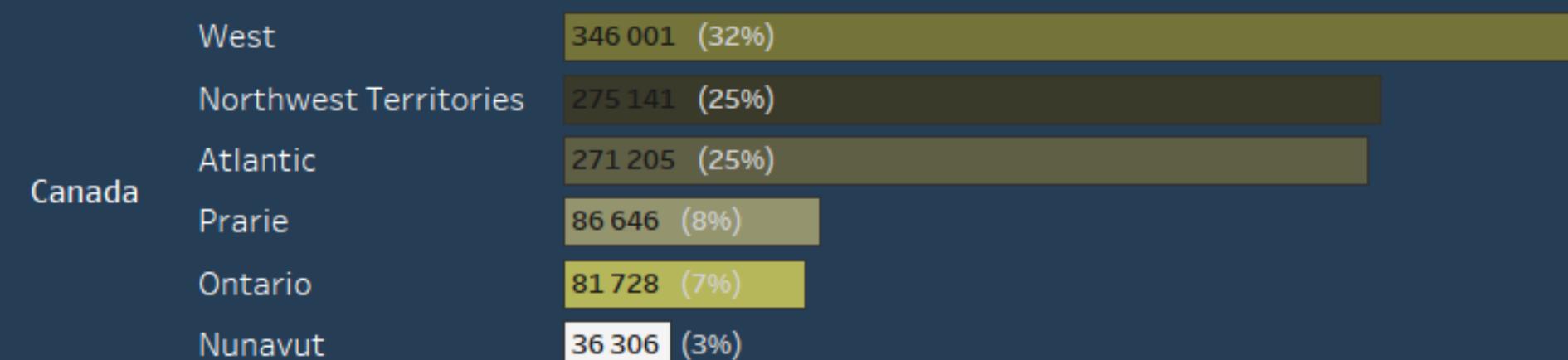
GENERAL DASHBOARD

SALES

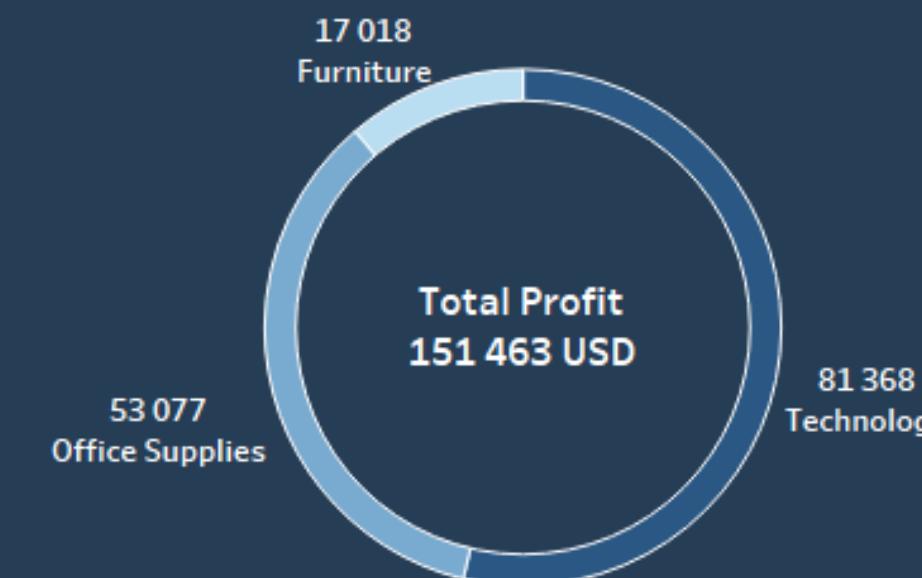
Profitable products

Corporate	Telephones ..		182 473
	Appliances		73 170
	Binders and ..		67 880
	Office Machi..		59 691
	Computer P..		43 218
	Copiers and ..		40 855
	Office Furni..		40 404
	Chairs & Cha..		39 406
	Storage & O..		28 247
	Paper		18 790
	Envelopes		14 857
	Tables		6 693
	Labels		2 620
Consumer	Telephones ..		33 087
	Office Machi..		17 681
	Binders and ..		14 490
	Office Furni..		13 380
	Appliances		12 004
	Paper		11 118
	Storage & O..		5 204
	Computer P..		3 125
	Envelopes		1 146
	Labels		556
Home Office	Telephones ..		9 227
	Storage & O..		8 468
	Binders and ..		7 609
	Office Furni..		5 704
	Chairs & Cha..		4 897
	Paper		3 144
	Envelopes		2 833
	Appliances		2 746
	Labels		199
Small Business	Telephones ..		11 696
	Copiers and ..		4 430
	Storage & O..		3 555
	Appliances		2 564
	Paper		2 157
	Office Furni..		2 137
	Computer P..		940
	Binders and ..		917

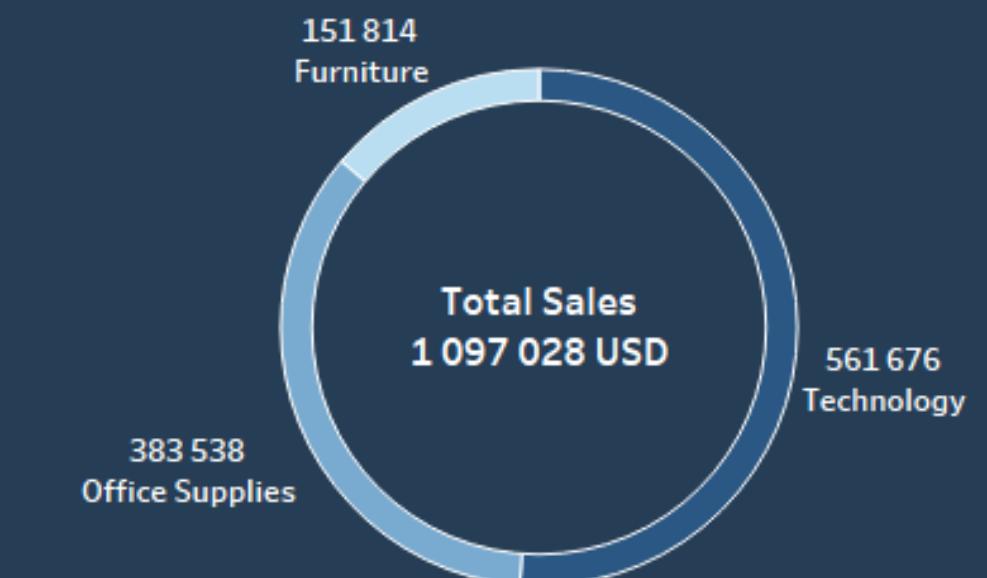
Rating of Canada's regions by sales



Profit by category



Sales by category



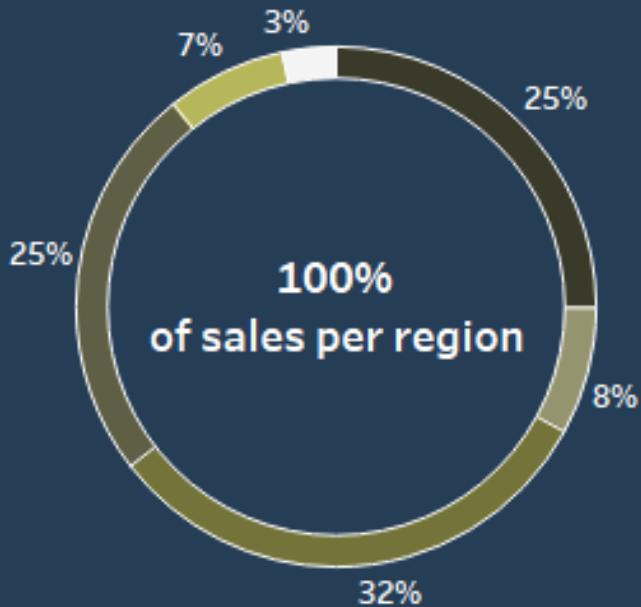
Sales, profit and quantity dynamics



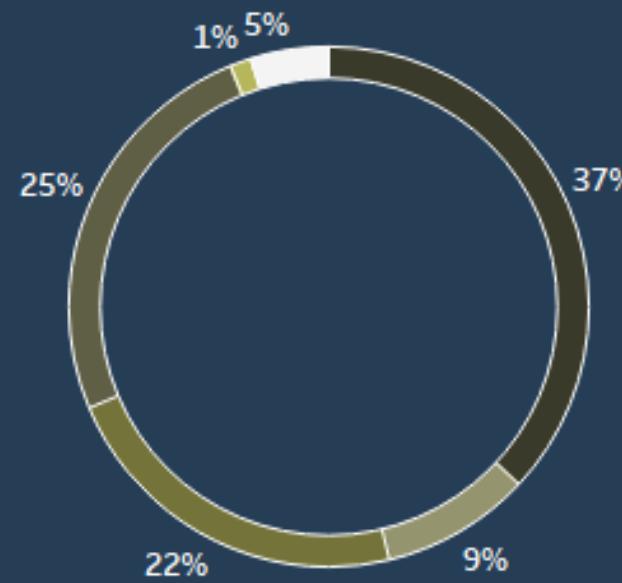
SALES DASHBOARD

GENERAL

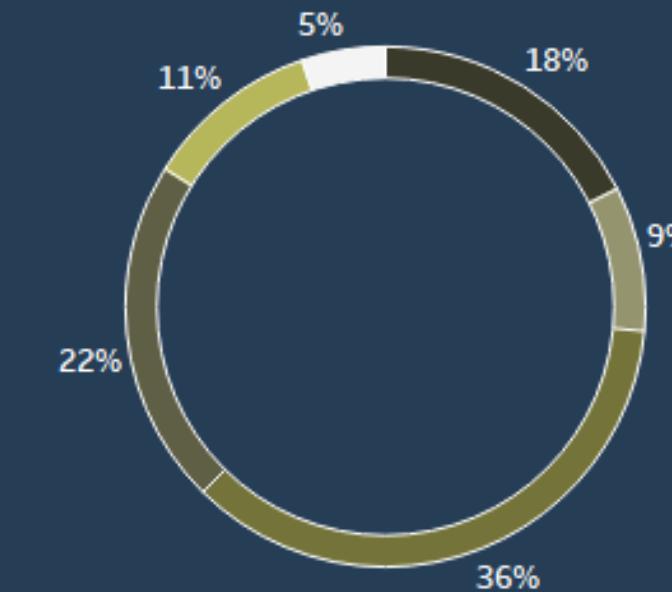
Sales per region



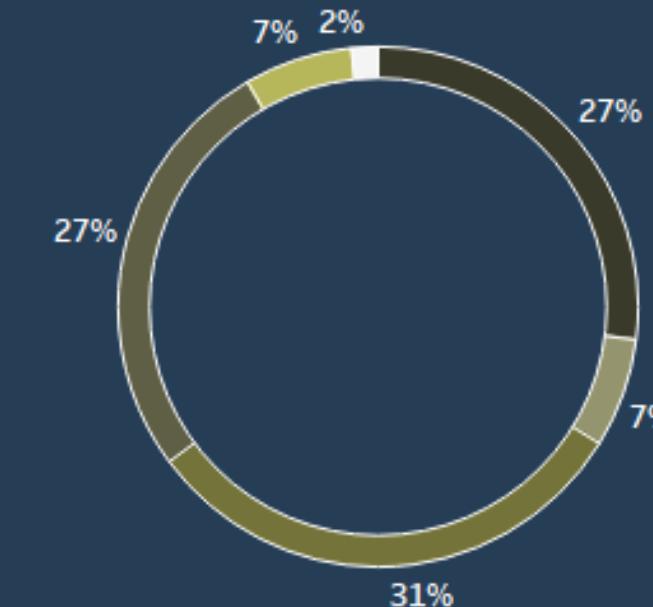
Sales by Furniture



Sales by Office Supplies



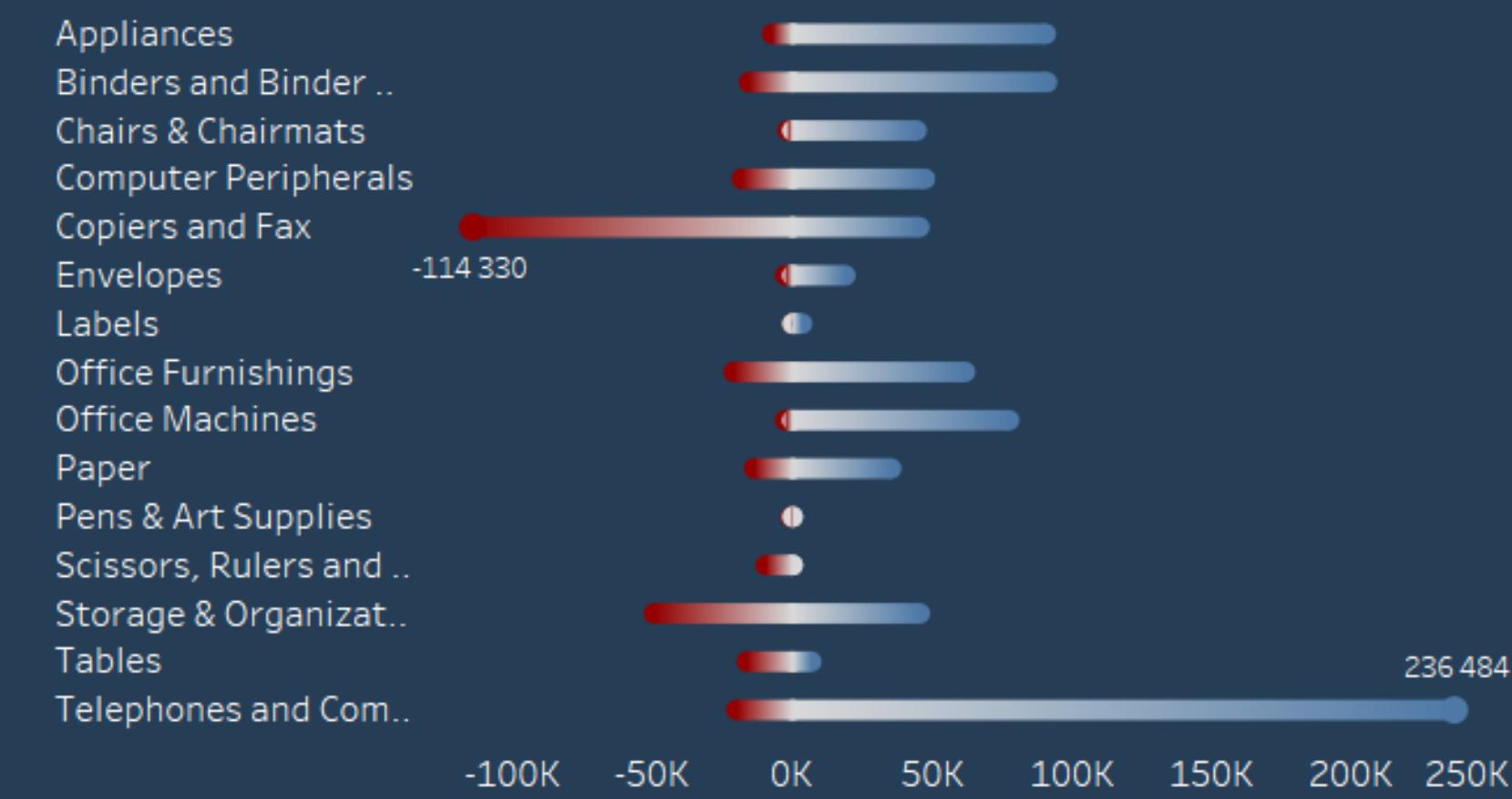
Sales by Technology



Average sum of sales per each users

Region	User Type	Average Sales
Atlantic	Corporate	4 899
	Home Office	2 223
	Small Business	884
	Consumer	2 555
Northwest Territories	Corporate	7 703
	Home Office	150
	Small Business	489
	Consumer	2 636
Nunavut	Corporate	4 092
	Home Office	987
	Small Business	1 602
	Consumer	5 895
Ontario	Corporate	5 895
	Home Office	1 602
	Small Business	489
	Consumer	2 636
Prairie	Corporate	4 092
	Home Office	987
	Small Business	1 602
	Consumer	5 895
West	Corporate	2 555
	Home Office	884
	Small Business	2 223
	Consumer	4 899

Sales with positive and negative profit



Power BI



2017 2018 2019

Sales fact in UAH/USD



City

All

UAH

USD

Analysis of sales and profit by region/shops

Город	Sales_Fact	Operating cost (fact)	Cost	Net Profit	% Sales by city and shops
Днепр					
Днепр	<u>4 563 367,73</u>	<u>1 538 960</u>	<u>2 574 552,45</u>	<u>449 855,38</u>	<u>17,19%</u>
Киев	<u>8 911 891,29</u>	<u>3 748 035</u>	<u>5 001 679,38</u>	<u>162 176,86</u>	<u>33,57%</u>
Киев №4 Позняки	2 321 174,13	977 912	1 299 522,95	43 739,00	26,05%
Киев №5 ЦУМ	2 036 925,68	864 360	1 130 490,81	42 075,02	22,86%
Киев №6 Троєщина	2 275 531,64	883 139	1 285 005,68	107 387,14	25,53%
Киев №7 КПИ	2 278 259,84	1 022 624	1 286 659,94	-31 024,31	25,56%
Львов	<u>6 357 609,05</u>	<u>2 950 161</u>	<u>3 570 394,42</u>	<u>-162 946,59</u>	<u>23,95%</u>
Львов №1 Плазма	2 049 829,40	817 789	1 163 459,79	68 580,37	32,24%
Львов №2 Софиевка	2 259 163,30	1 076 024	1 259 683,60	-76 543,83	35,53%
Львов №3 Бандери	2 048 616,36	1 056 348	1 147 251,03	-154 983,13	32,22%
Харьков	<u>6 711 764,70</u>	<u>2 601 956</u>	<u>3 749 723,79</u>	<u>360 085,01</u>	<u>25,28%</u>
Харьков №10 Плаза	2 344 946,59	952 674	1 315 332,94	76 940,09	34,94%
Харьков №11 Таирова	2 114 053,95	1 010 993	1 177 514,10	-74 453,39	31,50%
Харьков №12 Небесной Сотни	2 252 764,16	638 289	1 256 876,75	357 598,30	33,56%
Total	26 544 632,78	10 839 112	14 896 350,04	809 170,65	100,00%

Execution of sales plan

Город	Sales_Plan	Sales_Fact	Execution of sales plan %
Днепр			
Днепр	<u>3 511 380,00</u>	<u>4 563 367,73</u>	<u>129,96%</u>
Киев	<u>6 288 160,00</u>	<u>8 911 891,29</u>	<u>141,72%</u>
Киев №4 Позняки	1 567 080,00	2 321 174,13	148,12%
Киев №5 ЦУМ	1 378 950,00	2 036 925,68	147,72%
Киев №6 Троєщина	1 652 070,00	2 275 531,64	137,74%
Киев №7 КПИ	1 690 060,00	2 278 259,84	134,80%
Львов	<u>4 608 230,00</u>	<u>6 357 609,05</u>	<u>137,96%</u>
Львов №1 Плазма	1 521 400,00	2 049 829,40	134,73%
Львов №2 Софиевка	1 527 230,00	2 259 163,30	147,93%
Львов №3 Бандери	1 559 600,00	2 048 616,36	131,36%
Харьков	<u>4 929 320,00</u>	<u>6 711 764,70</u>	<u>136,16%</u>
Total	19 337 090,00	26 544 632,78	137,27%

2017 2018 2019

January February March April May June July August September October November December

11,82%

Sales increase compared to the last month

-5,02%

Sales increase compared to the same period last year

Sales_Fact and Sales_Plan

534,52K



Cumulative YTD_Fact and Cumulative YTD_Plan

6 263,49K

9 590.14K

City

All



Sales increase compared to the last month by Category

Економ

22,94%

Стандарт

19,32%

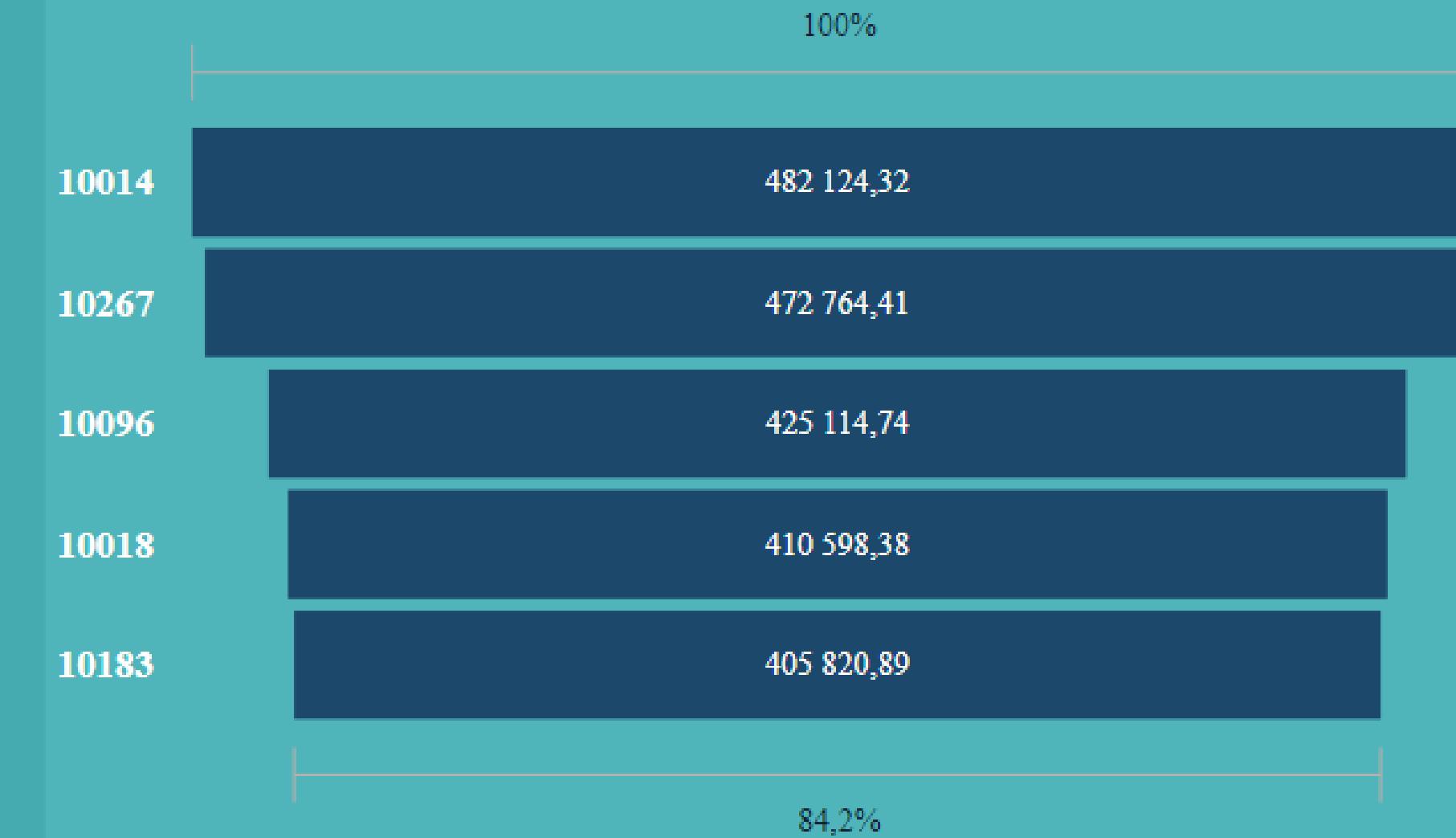
Люкс

11,65%

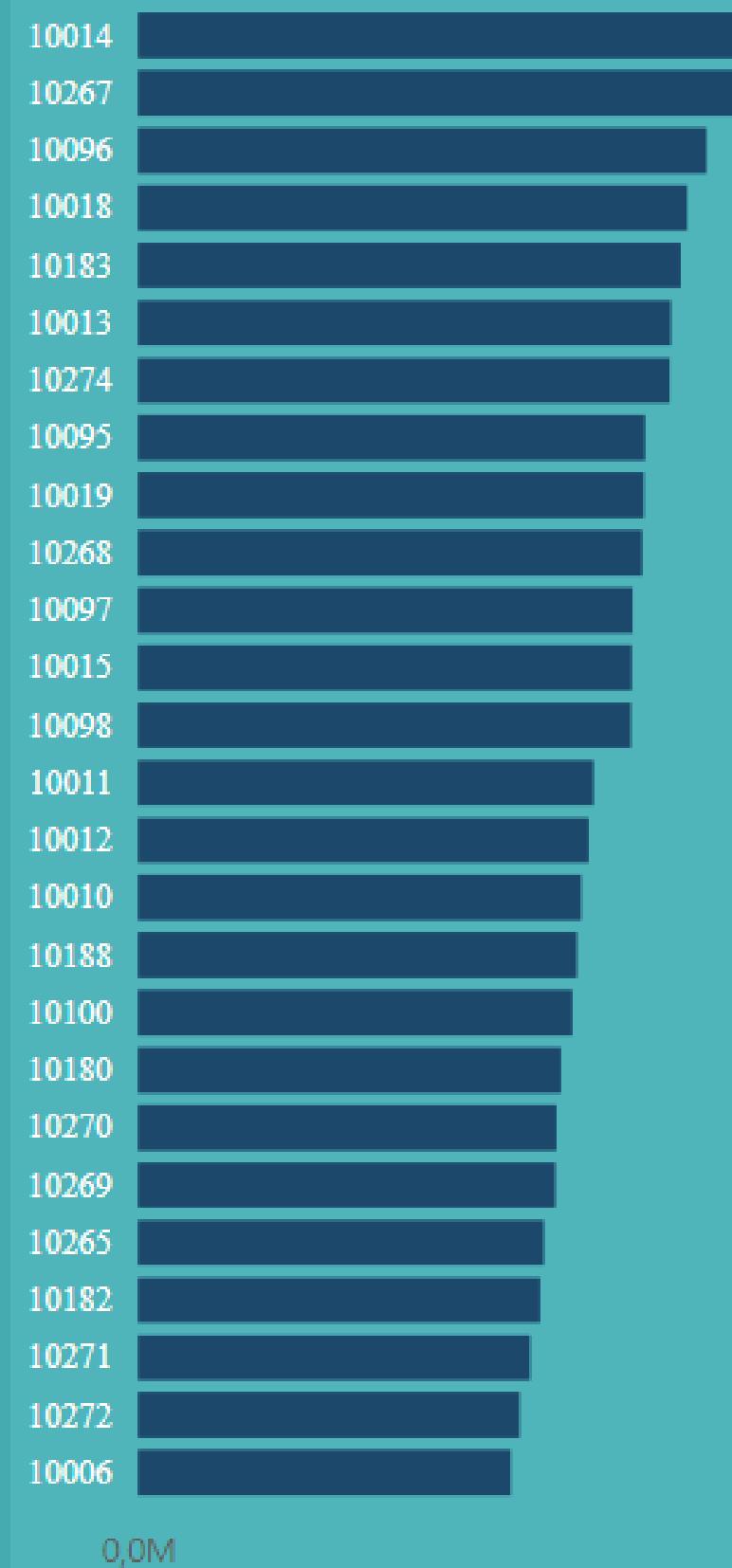
Премиум

10,67%

TOP 5 Products (Sales, UAH)



Products's sales (fact), UAH



Chart

Table

Products report

● Економ ● Люкс ● Премиум ● Сервис ● Стандарт

2K

1K

0K

группа 1

группа 2

группа 3

группа 7

группа 4

группа 5

группа 6

City

All

Column

Группа

Категория

Размер

Цвет

Row

Группа

Категория

Размер

Цвет

Price group (Quantity sales, pcs)

● 0-1000 ● 1000-1500 ● 1500-2000 ● 2000+

Економ

98,35%

Люкс

15,91%

9,37%

69,08%

Премиум

56,84%

20,44%

15,31%

7,41%

Сервис

100,00%

Стандарт

99,61%

4123

The balance in the warehouse, pcs



10096

First ID новый

ЛЮКС

First Категория

77

Quantity Purchases

группа 2

First Группа

65

Quantity Sales

бирюзов...

First Цвет

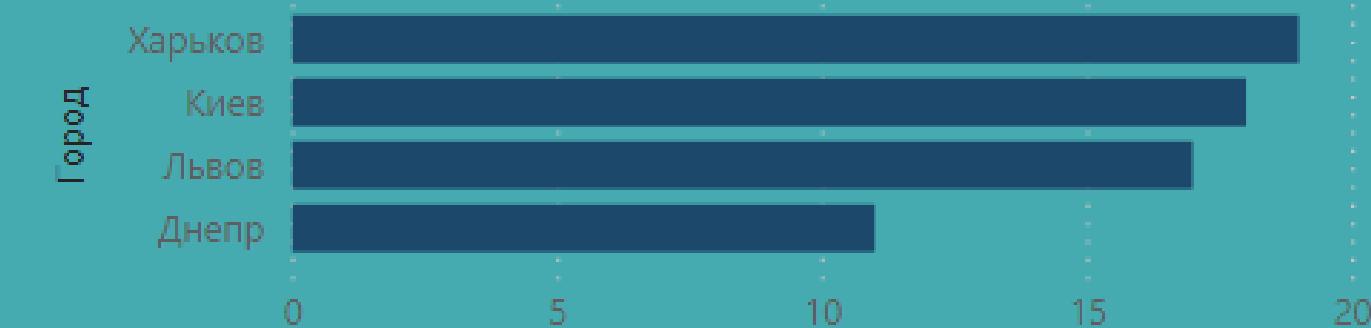
12

The balance in the warehouse, pcs

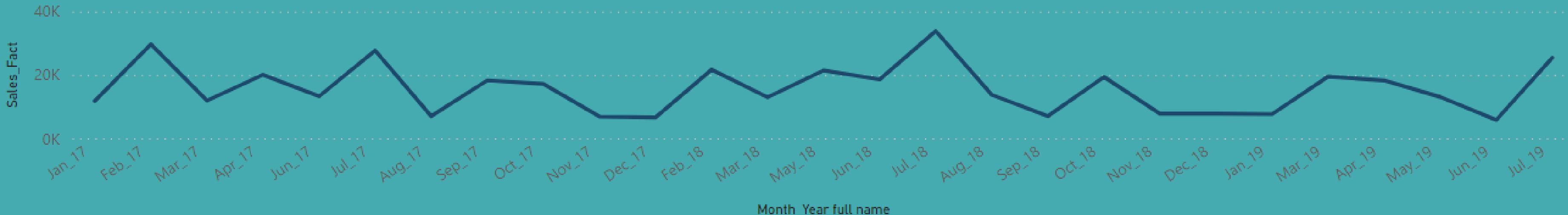
Поставщик
Авиас

First Поставщик

Quantity Sales by Город



Sales_Fact by Month_Year full name



believe in new beginnings