In [2]:

```
import pandas as pd
import plotly.express as px
from scipy.stats import f_oneway
from dython import nominal
```

In [3]:

```
df=pd.read_csv('data/Global_Superstore2.csv', encoding = "ISO-8859-1")
```

In [4]:

```
df.head()
```

Out[4]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City	
0	32298	CA-2012-124891	31-07-2012	31-07-2012	Same Day	RH-19495	Rick Hansen	Consumer	New York City	Nev
1	26341	IN-2013-77878	05-02-2013	07-02-2013	Second Class	JR-16210	Justin Ritter	Corporate	Wollongong	New
2	25330	IN-2013-71249	17-10-2013	18-10-2013	First Class	CR-12730	Craig Reiter	Consumer	Brisbane	Queer
3	13524	ES-2013-1579342	28-01-2013	30-01-2013	First Class	KM-16375	Katherine Murray	Home Office	Berlin	
4	47221	SG-2013-4320	05-11-2013	06-11-2013	Same Day	RH-9495	Rick Hansen	Consumer	Dakar	

5 rows × 24 columns

Entfernen aller Attribute, die hier nicht untersucht werden:

In [5]:

```
def deleteColumns(pColumns):
    for i in range(0,len(pColumns)):
        del df[pColumns[i]]
```

In [6]:

```
deleteColumns(["Order ID", "Order Date", "Ship Mode", "Customer ID", "Customer Name",
"Segment", "Ship Date", "City", "State", "Country", "Market", "Region", "Postal Code",
"Sales","Quantity","Discount","Shipping Cost","Order Priority"])
```

In [7]:

```
df.head()
```

Out[7]:

	Row ID	Product ID	Category	Sub-Category	Product Name	Profit
0	32298	TEC-AC-10003033	Technology	Accessories	Plantronics CS510 - Over-the-Head monaural Wir...	762.1845
1	26341	FUR-CH-10003950	Furniture	Chairs	Novimex Executive Leather Armchair, Black	-288.7650
2	25330	TEC-PH-10004664	Technology	Phones	Nokia Smart Phone, with Caller ID	919.9710
3	13524	TEC-PH-10004583	Technology	Phones	Motorola Smart Phone, Cordless	-96.5400
4	47221	TEC-SHA-10000501	Technology	Copiers	Sharp Wireless Fax, High-Speed	311.5200

Definieren von Funktionen, die die "Clean Code Guidelines" erfüllen und im ganzen Dokument zur Analyse von Attributbeziehungen genutzt werden

In [8]:

```
def countColumn(pColumn, pColumnName, pYName):
    groups=df.groupby(pColumn)
    amount=groups.count()[["Row ID"]]
    dataset = pd.DataFrame({pColumnName: list(df.groupby(pColumn).groups.keys()), pYNam
e: amount["Row ID"]}, columns=[pColumnName, pYName])
    colour=amount["Row ID"]
    return dataset,colour
```

In [9]:

```
def dfOfAverageMeans(pColumn, pValue, pColumnName, pYName):
    means = []
    groups=df.groupby(pColumn)
    for index,group in groups:
        current = group[pValue]
        currentMean = current.mean()
        means.append(currentMean)
    dataset = pd.DataFrame({pColumnName: list(df.groupby(pColumn).groups.keys()), pYNam
e: means}, columns=[pColumnName, pYName])
    return dataset, means
```

In [10]:

```
def twoStepSunburst(pColumn1, pColumn2):
    dataframe = df.groupby(by=[pColumn1, pColumn2]).count()[["Row ID"]].rename(columns=
{"Row ID":"Anzahl"})
    dataframe = dataframe.reset_index()
    fig = px.sunburst(dataframe,path=[pColumn1, pColumn2], values="Anzahl")
    fig.show()
```

In [11]:

```
def numberOfTransactions(pColumn):
    count_r=df.groupby(by=pColumn).count()[["Row ID"]].rename(columns={"Row ID":"Number
of Transactions"})
    return count_r.sort_values(by="Number of Transactions")
```

In [12]:

```
def howOftenDoAmountsAppear(pColumn):
    counter=df.groupby(by=pColumn).count()[["Row ID"]].rename(columns={"Row ID":"Number
of Transactions"})
    counting_amounts=counter.groupby(['Number of Transactions']).size().reset_index(nam
e='counts')
    counting_amounts.sort_values(by="Number of Transactions")
    return counting_amounts
```

In [13]:

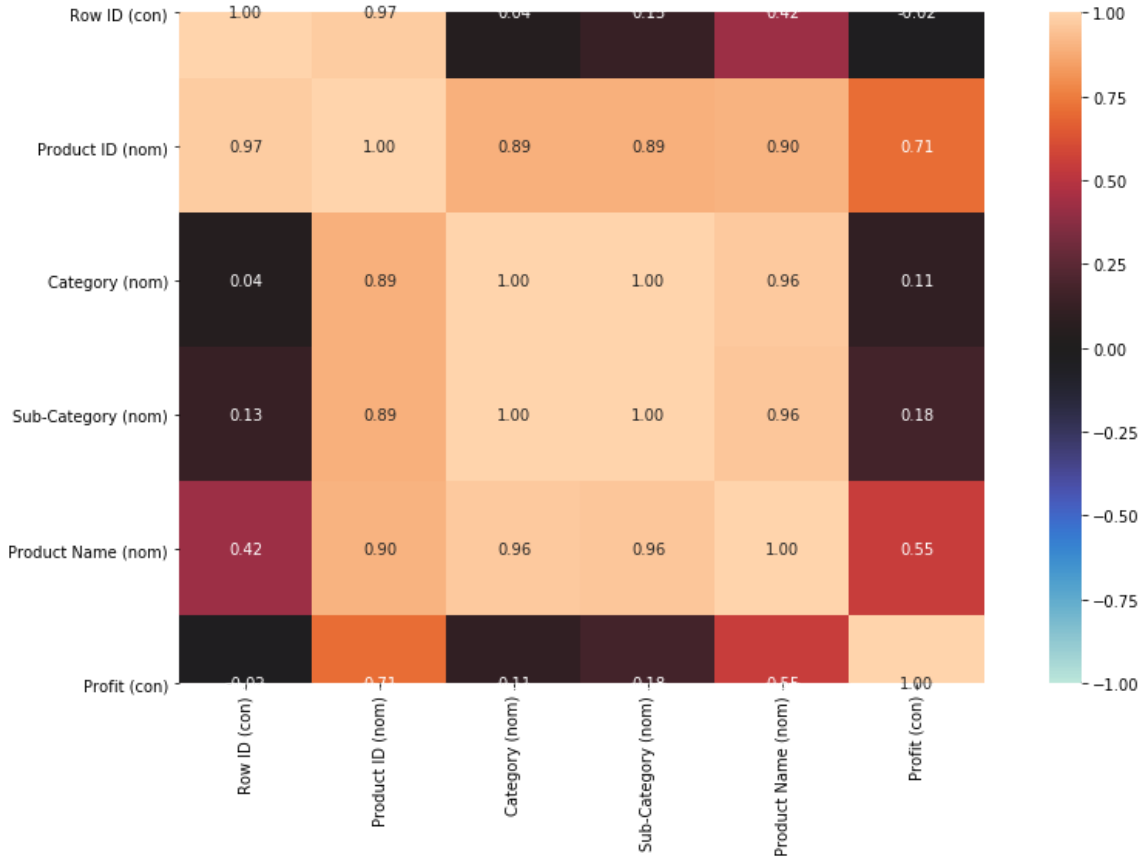
```
def calculateAnovaScore(pColumn1, pColumn2):
    CategoryGroupLists=df.groupby(pColumn1)[pColumn2].apply(list)
    AnovaResults = f_oneway(*CategoryGroupLists)
    print('P-Wert für Anova ist: ', AnovaResults[1])
    print('F Score für Anova ist: ', AnovaResults[0])
    if AnovaResults[1] < 0.05:
        print(f"Der Unterschied der {pColumn2}-Mittelwerte zwischen den verschiedenen G
ruppen von {pColumn1} ist signifikant, da der p-Wert unter 0.05 liegt.")
```

Berechnen und Erstellen der Korrelations Heatmap der relevanten Attribute

Wenn im Folgenden von Werten aus der Korrelationstabelle/-heatmap gesprochen wird, ist diese gemeint

In [14]:

```
nominal.associations(df,figsize=(15,8),mark_columns=True)
```



Out[14]:

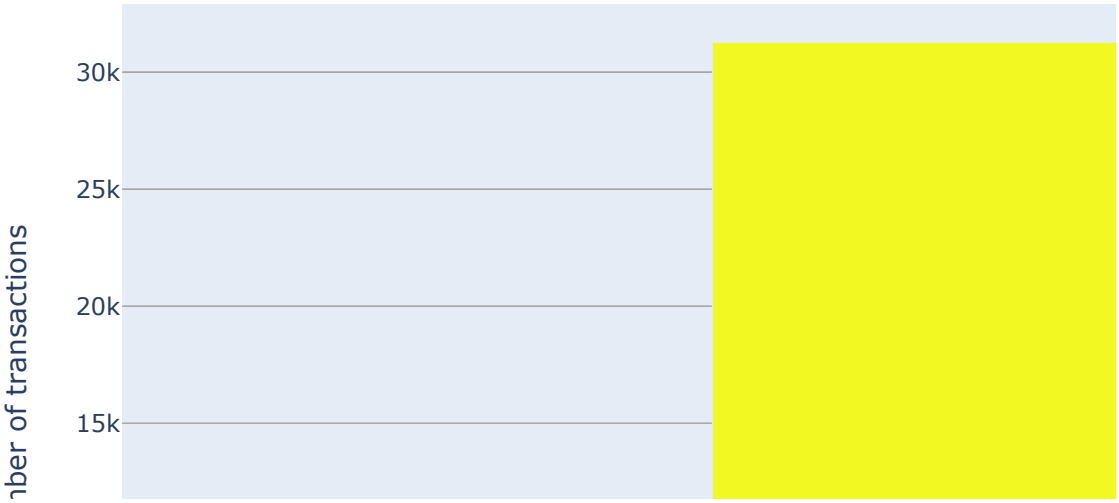
```
{'corr':                    Row ID (con)  Product ID (nom)  Category (no
m)  \
 Row ID (con)           1.000000          0.972239        0.042214
 Product ID (nom)       0.972239          1.000000        0.894083
 Category (nom)         0.042214          0.894083        1.000000
 Sub-Category (nom)     0.129397          0.894162        0.999864
 Product Name (nom)     0.422326          0.901826        0.961878
 Profit (con)          -0.019037          0.708139        0.108329

                    Sub-Category (nom)  Product Name (nom)  Profit (con)
 Row ID (con)                 0.129397            0.422326     -0.019037
 Product ID (nom)             0.894162            0.901826      0.708139
 Category (nom)               0.999864            0.961878      0.108329
 Sub-Category (nom)           1.000000            0.960318      0.177225
 Product Name (nom)           0.960318            1.000000      0.553277
 Profit (con)                 0.177225            0.553277      1.000000
,
 'ax': <matplotlib.axes._subplots.AxesSubplot at 0x23c678d5be0>}
```

1. Analyse des Attributs Category

1.1 Verteilung der Transaktionen pro Category

In [15]:

```
result,colour = countColumn("Category","All regions","Number of transactions")
fig = px.bar(result, x="All regions", y="Number of transactions", color=colour)
fig.show()
```



1.2 Prüfen, ob jede Ausprägung von Category genug Transaktionen hat, um aussagekräftig zu sein

In [16]:

```
numberOfTransactions("Category")
```

Out[16]:

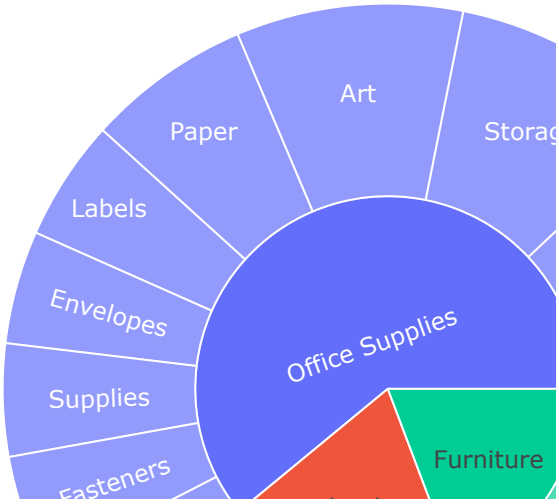
Category	Number of Transactions
Furniture	9876
Technology	10141
Office Supplies	31273

1.3 Verteilung der anderen Produkttypen innerhalb der Kategorien

1.3.1 Verteilung Sub-Categories pro Category

In [17]:

```
twoStepSunburst("Category","Sub-Category")
```



weitere Verteilungen nicht mehr visualisierbar, da Sunburst Diagramm sonst zu aufgespalten wäre

1.4 Korrelation zwischen "Category" and "Profit"

1.4.1 Berechnungen der statistischen Korrelationswerte

Wert aus der Heatmap: 0,11

In [18]:

```
calculateAnovaScore("Category","Profit")
```

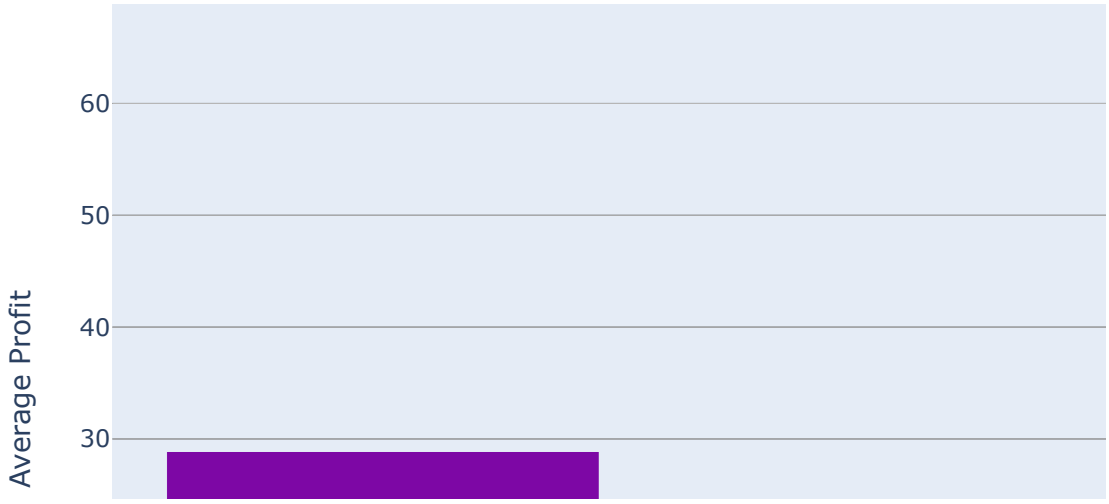
```
P-Wert für Anova ist:  3.4173111634965594e-132
F Score für Anova ist:  304.50613538510214
Der Unterschied der Profit-Mittelwerte zwischen den verschiedenen Gruppen
von Category ist signifikant, da der p-Wert unter 0.05 liegt.
```

1.4.2 Diagramme zur Prüfung der Korrelation zwischen dem durchschnittlichen Profit und Category

a) Balkendiagramm

In [19]:

```
result,means = dfOfAverageMeans("Category","Profit", "All categories", "Average Profit"
)
fig = px.bar(result, x="All categories", y="Average Profit", color=means)
fig.show()
```



Box Plot und Streudiagramm haben keine Aussagekraft, weil es zu wenige Punkte gibt

1.4.3 Fazit

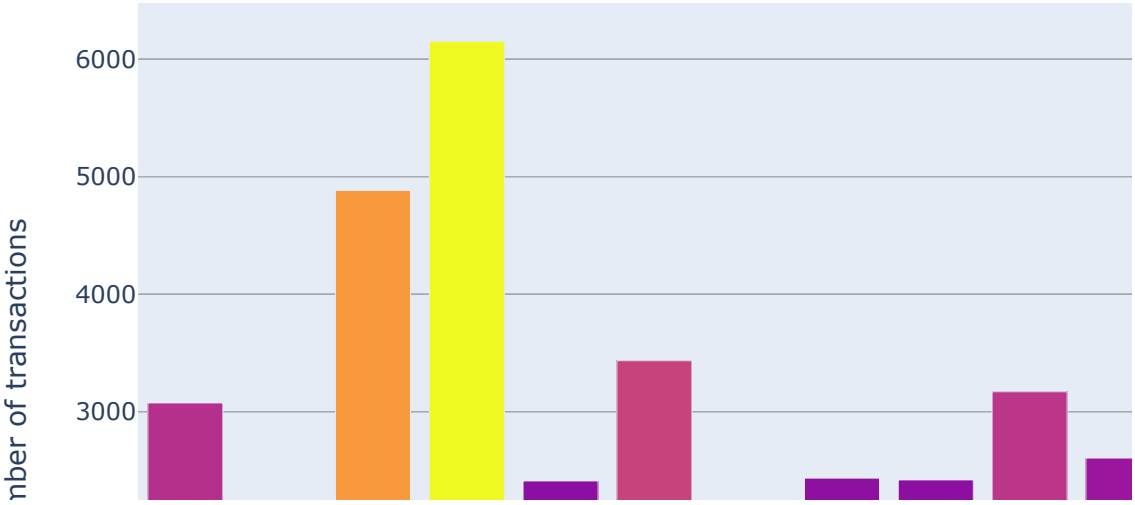
Für ein Attribut, bei dem jede Gruppe mindestens 9000 Transaktionen hat, ist der Heatmap Wert relativ hoch. Auch der hohe F-Score Wert und die Tatsache, dass sich die Profits im Balkendiagramm so stark unterscheiden, sprechen dafür, dass Category gut geeignet ist, um den Profit vorherzusagen.

2. Analyse des Attributs Sub-Category

2.1 Verteilung der Transaktionen pro Category

In [20]:

```
result,colour = countColumn("Sub-Category","All regions","Number of transactions")
fig = px.bar(result, x="All regions", y="Number of transactions", color=colour)
fig.show()
```



2.2 Prüfen ob jede Gruppe von Sub-Category genug Transaktionen hat um aussagekräftig zu sein

In [21]:

```
numberOfTransactions("Sub-Category")
```

Out[21]:

Sub-Category	Number of Transactions
Tables	861
Machines	1486
Appliances	1755
Copiers	2223
Bookcases	2411
Fasteners	2420
Supplies	2425
Envelopes	2435
Labels	2606
Accessories	3075
Furnishings	3170
Phones	3357
Chairs	3434
Paper	3538
Art	4883
Storage	5059
Binders	6152

2.3 Korrelation zwischen "Sub-Category" und "Profit"

2.3.1 Rechnerische Bestimmung der statistischen Korrelationswerte

Wert aus Heatmap: 0,18

In [22]:

```
calculateAnovaScore("Sub-Category","Profit")
```

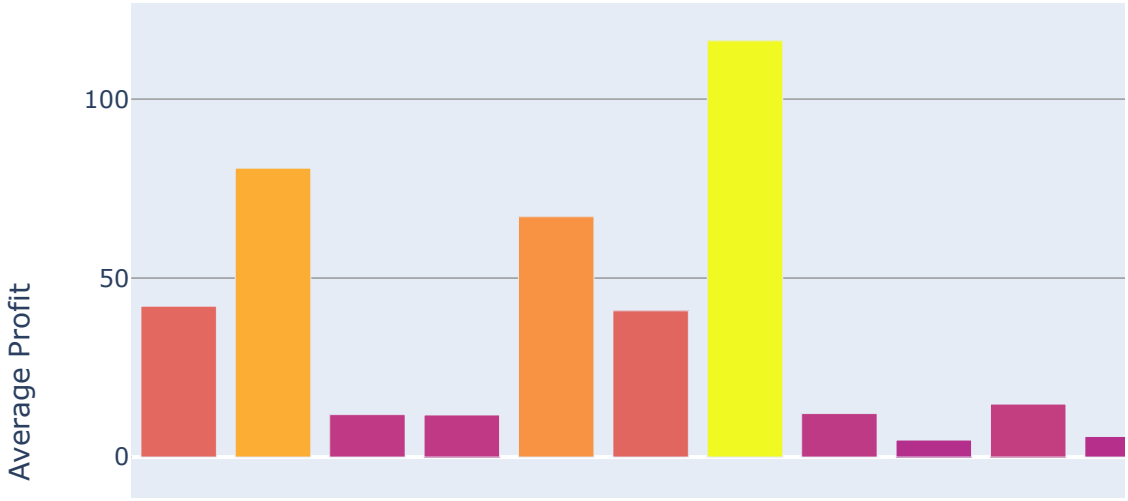
```
P-Wert für Anova ist:  0.0
F Score für Anova ist:  103.91551183097657
Der Unterschied der Profit-Mittelwerte zwischen den verschiedenen Gruppen
von Sub-Category ist signifikant, da der p-Wert unter 0.05 liegt.
```

2.3.2 Grafische Prüfung der Korrelation

a) Balkendiagramm

In [23]:

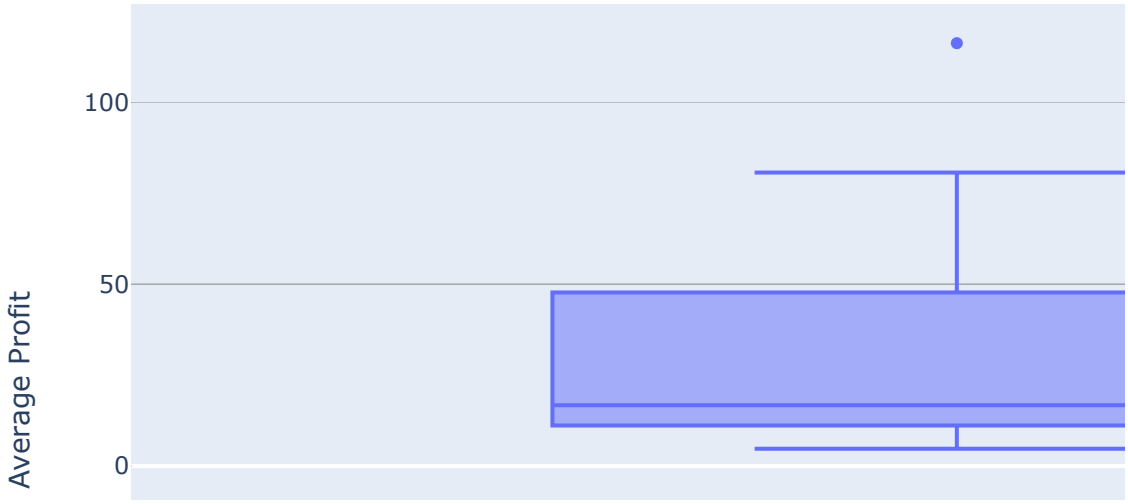
```
result,means = dfOfAverageMeans("Sub-Category","Profit", "All Sub-Categorys", "Average
 Profit")
fig = px.bar(result, x="All Sub-Categorys", y="Average Profit", color=means)
fig.show()
```



b) Box Plot

In [24]:

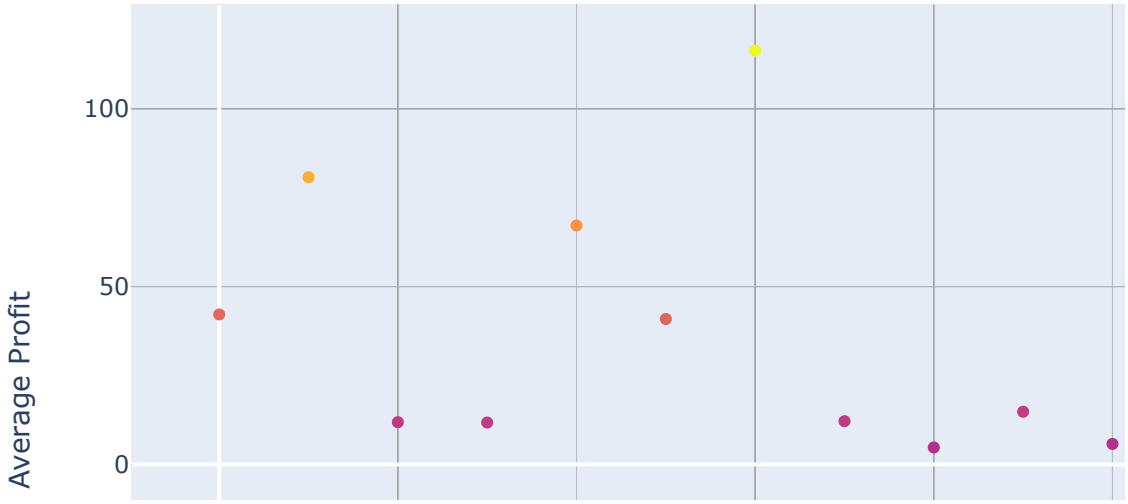
```
result,means = dfOfAverageMeans("Sub-Category","Profit", "All Sub-Categorys", "Average
 Profit")
fig = px.box(result, y="Average Profit")
fig.show()
```



c) Streudiagramm

In [25]:

```
result,means = dfOfAverageMeans("Sub-Category","Profit", "All Sub-Categorys", "Average
 Profit")
fig = px.scatter(result, y="Average Profit", color=means)
fig.show()
```



2.4 Fazit

Auch die Sub-Category scheint ein geeignetes Attribut zur Vorhersage des Profits sein. Die statistischen Werte und Diagramme deuten auf einen Zusammenhang zwischen Sub-Category und Profit hin. Jede Gruppe hat genug Werte um aussagekräftig zu sein. Die weiteren Argumentationen von Category gelten hier auch

3. Analysing the attribute Product ID

3.1 Verteilung der Transaktionen pro Product ID

In [26]:

```
# result,colour = countColumn("Product ID","All Product IDs","Number of transactions")
# fig = px.bar(result, x="All Product IDs", y="Number of transactions", color=colour)
# fig.show()
```

Diagramm nicht mehr sinnvoll darstellbar --> viel zu viele Balken.

In [27]:

```
numberOfTransactions("Product ID")
```

Out[27]:

Product ID	Number of Transactions
TEC-STA-10004927	1
OFF-FA-10004269	1
FUR-FU-10002930	1
OFF-FA-10004374	1
FUR-FU-10002874	1
...	...
FUR-CH-10003354	28
OFF-BI-10003708	30
OFF-BI-10002799	30
OFF-AR-10003829	31
OFF-AR-10003651	35

10292 rows × 1 columns

In [28]:

```
howOftenDoAmountsAppear("Product ID")
```

Out[28]:

	Number of Transactions	counts
0	1	1420
1	2	1223
2	3	1296
3	4	1357
4	5	1278
5	6	1045
6	7	756
7	8	554
8	9	374
9	10	296
10	11	159
11	12	161
12	13	124
13	14	71
14	15	58
15	16	38
16	17	22
17	18	15
18	19	13
19	20	12
20	21	1
21	22	5
22	23	3
23	24	4
24	25	1
25	27	1
26	28	1
27	30	2
28	31	1
29	35	1

Product ID hat über 10000 verschiedene Ausprägungen/Gruppen. Hierbei kommen 1420 verschiedene Product IDs nur einmal vor. Sogar die am häufigsten vorkommende Gruppe hat nur 35 Transaktionen. Dies spricht jetzt schon dafür, dass Product ID kein geeignetes Attribut ist, um den Profit vorherzusagen. Denn es gibt in jeder Gruppe viel zu wenig Vorkommnisse, um für diese spezifische Product ID generalisieren zu können. Dennoch werden zur Bestätigung dieser Annahmen die Prüfungen zur Korrelation durchgeführt, um evtl. versteckte Korrelationen zu finden.

3.2 Prüfung der Korrelation zwischen Product ID und Profit

3.2.1 rechnerische Bestimmung der Korrelationswerte

Wert aus Heatmap: 0,71

In [29]:

```
calculateAnovaScore("Product ID","Profit")
```

```
P-Wert für Anova ist:  0.0
F Score für Anova ist:  4.007224887155502
Der Unterschied der Profit-Mittelwerte zwischen den verschiedenen Gruppen
von Product ID ist signifikant, da der p-Wert unter 0.05 liegt.
```

3.2.2 Dlagramme zur Visualisierung der Zusammenhänge

a) Balkendiagramm nicht sinnvoll, da zu viele Balken

b) Box Plot

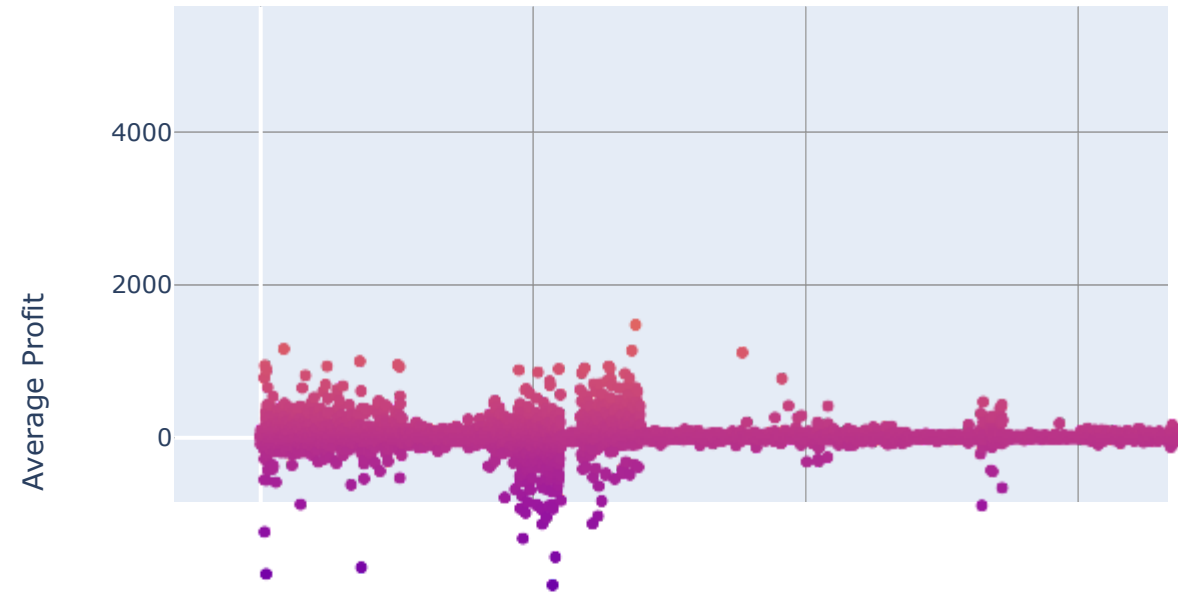
In [30]:

```
result,means = dfOfAverageMeans("Product ID","Profit", "All Product IDs", "Average Profit")
fig = px.box(result, y="Average Profit")
fig.show()
```

c) Streudiagramm

In [31]:

```
result,means = dfOfAverageMeans("Product ID","Profit", "All Product IDs", "Average Profit")
fig = px.scatter(result, y="Average Profit", color=means)
fig.show()
```



3.3 Fazit

Die zuvor getätigen Annahmen bestätigen sich in den Diagrammen. Es gibt keinen validen Zusammenhang zwischen Product ID und Profit. Das Streudiagramm zeigt nur zufällige Verhältnisse, die durch Ausreißer bedingt sind und der Box Plot zeigt wie sehr die Ausreißer das Gesamtbild verzerren. Der extrem kleine F-Score bestätigt dies. Der hohe Wert in der Heatmap kommt daher, dass innerhalb einer Gruppe die Korrelation mit Profit oft hoch ist, da es nur ein paar wenige Transaktionen pro Gruppe gibt und diese dann natürlich mit Profit korrelieren. Aus diesen Gründen, ist Product ID nicht sinnvoll zur Vorhersage des Profits nutzbar.

In [32]:

```
del df["Product ID"]
```

4. Product Name

4.1 Prüfen der Anzahl Transaktionen pro Gruppe

In [33]:

```
numberOfTransactions("Product Name")
```

Out[33]:

Product Name	Number of Transactions
Barricks Coffee Table, with Bottom Storage	1
Sanitaire Vibra Groomer IR Commercial Upright Vacuum, Replacement Belts	1
Hewlett-Packard Deskjet 5550 Printer	1
Hewlett-Packard Deskjet 3050a All-in-One Color Inkjet Printer	1
Grip Seal Envelopes	1
...	...
Ibico Index Tab, Clear	83
Rogers File Cart, Single Width	84
Eldon File Cart, Single Width	90
Cardinal Index Tab, Clear	92
Staples	227

3788 rows × 1 columns

In [34]:

```
howOftenDoAmountsAppear("Product Name")
```

Out[34]:

	Number of Transactions	counts
0	1	98
1	2	178
2	3	238
3	4	306
4	5	304
...	...	...
61	83	1
62	84	1
63	90	1
64	92	1
65	227	1

66 rows × 2 columns

Product Name hat über 3700 verschiedene Ausprägungen, von denen wieder viele viel zu wenige Transaktionen haben. Es gibt nur die Gruppe "Staples", die genügend Transaktionen hätte.

4.2 Korrelation zwischen Product Name und Profit

4.2.1 rechnerische Bestimmung der Korrelationswerte

Wert aus Heatmap: 0,55

In [35]:

```
calculateAnovaScore("Product Name", "Profit")
```

```
P-Wert für Anova ist:  0.0
F Score für Anova ist:  5.533698195429963
Der Unterschied der Profit-Mittelwerte zwischen den verschiedenen Gruppen
von Product Name ist signifikant, da der p-Wert unter 0.05 liegt.
```

4.2.2 Diagramme zur Analyse der Korrelation

a) Balkendiagramm --> wegen zu vieler Balken wieder nicht gut lesbar

In [36]:

```
# result,means = dfOfAverageMeans("Product Name","Profit", "All Sub-Categorys", "Averag
e Profit")
# fig = px.bar(result, x="All Sub-Categorys", y="Average Profit", color=means)
# fig.show()
```

b) Box Plot

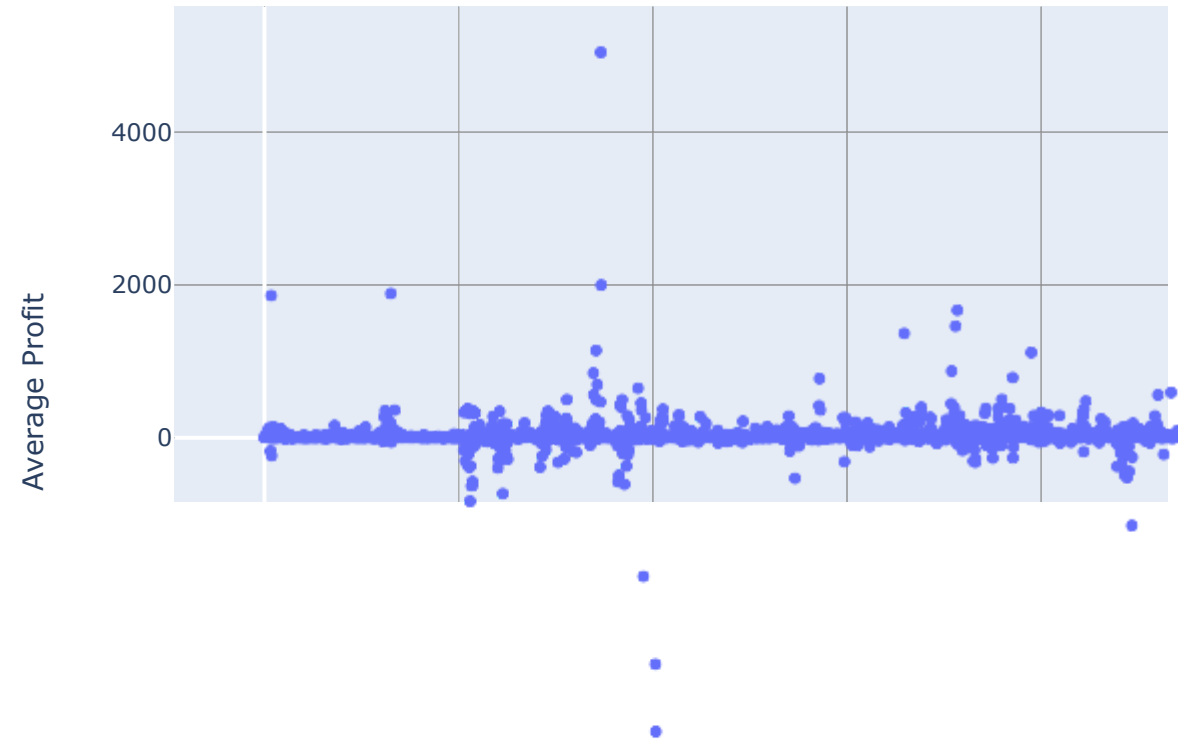
In [37]:

```
result,means = dfOfAverageMeans("Product Name","Profit", "All Product Names", "Average
 Profit")
fig = px.box(result, y="Average Profit")
fig.show()
```

c) Streudiagramm

In [38]:

```
result,means = dfOfAverageMeans("Product Name","Profit", "All Product Names", "Average
 Profit")
fig = px.scatter(result, y="Average Profit")
fig.show()
```



4.3 Fazit

Die rechnerischen Werte des Product Names gleichen denen der Product ID, was dafür spricht, dass es kein geeignetes Attrbut ist. Insgesamt ist Product Name zwar etwas aggregierter als Product ID, aber die Diagramme haben immer noch keine Aussagekraft und die Anzahl Transaktionen pro Gruppe erlaubt keine zuverlässige Prognose. Auch das Attribut Product Name wird also nicht weiter berücksichtigt.

In [39]:

```
del df["Product Name"]
```

Überarbeitetes DataFrame der produktbezogenen Attribute

In [40]:

```
df
```

Out[40]:

	Row ID	Category	Sub-Category	Profit
0	32298	Technology	Accessories	762.1845
1	26341	Furniture	Chairs	-288.7650
2	25330	Technology	Phones	919.9710
3	13524	Technology	Phones	-96.5400
4	47221	Technology	Copiers	311.5200
...	...	...	...	...
51285	29002	Office Supplies	Fasteners	4.5000
51286	35398	Office Supplies	Appliances	-1.1100
51287	40470	Office Supplies	Envelopes	11.2308
51288	9596	Office Supplies	Binders	2.4000
51289	6147	Office Supplies	Paper	1.8000

51290 rows × 4 columns