In [1]:

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

In [2]:

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

In [3]:

```
df.head()
```

Out[3]:

	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

In [ ]:

```

```

Entfernen der Attribute, die hier nicht untersucht werden

In [4]:

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

In [5]:

```
deleteColumns(["Order ID", "Order Date", "Ship Mode", "Customer ID", "Customer Name",
"Segment", "Ship Date", "Product ID", "Product Name", "Category", "Sub-Category","Sale
s","Quantity","Discount","Shipping Cost","Order Priority"])
```

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

In [6]:

```
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 [7]:

```
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 [8]:

```
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 [9]:

```
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 [10]:

```
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 [11]:

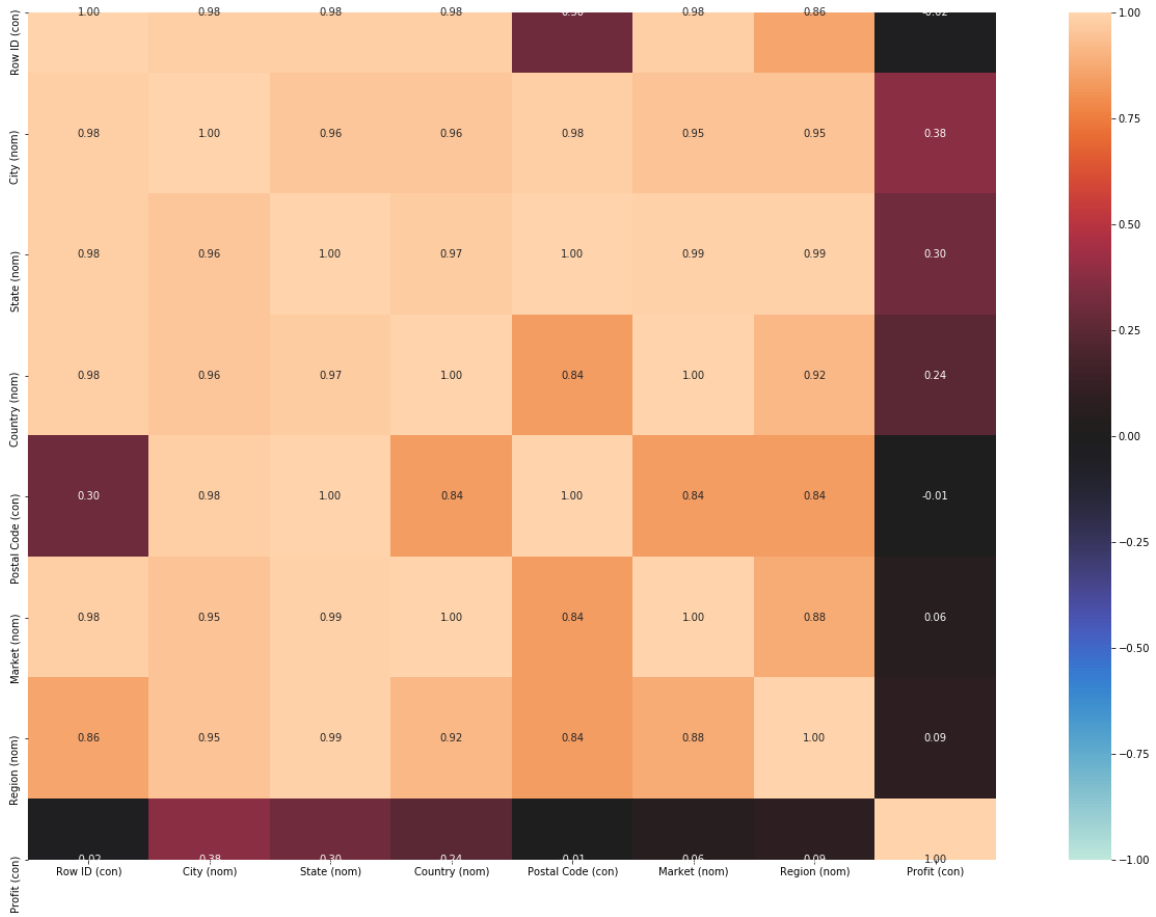
```
def calculateAnovaScore(pColumn1, pColumn2):
    CategoryGroupLists=df.groupby(pColumn1)[pColumn2].apply(list)
    AnovaResults = f_oneway(*CategoryGroupLists)
    print('P-Wert für Anova ist: ', AnovaResults[1])
    print('F Score für Anova ist: ', AnovaResults[0])
    if AnovaResults[1] < 0.05:
        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 generellen Korrelations Heatmap

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

In [12]:

```
nominal.associations(df,figsize=(25,15),mark_columns=True)
```


Row ID (con)
City (nom)
State (nom)
Country (nom)
Postal Code (con)
Market (nom)
Region (nom)
Profit (con)
1.00 0.98 0.98 0.98 0.98 0.86
0.98 1.00 0.96 0.96 0.98 0.95 0.95 0.38
0.98 0.96 1.00 0.97 1.00 0.99 0.99 0.30
0.98 0.96 0.97 1.00 0.84 1.00 0.92 0.24
0.30 0.98 1.00 0.84 1.00 0.84 0.84 -0.01
0.98 0.95 0.99 1.00 0.84 1.00 0.88 0.06
0.86 0.95 0.99 0.92 0.84 0.88 1.00 0.09
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Out[12]:

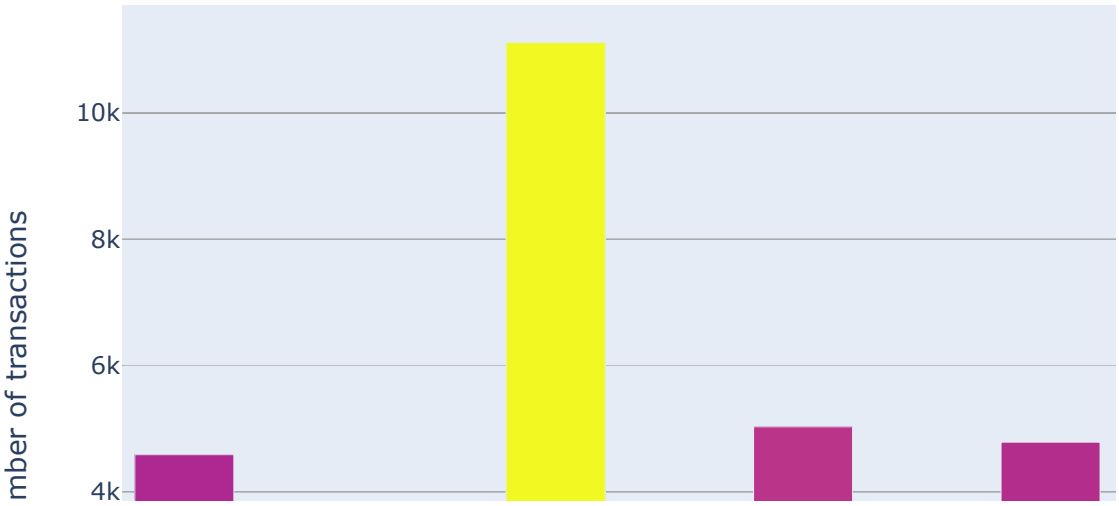
```
{'corr':                  Row ID (con)  City (nom)  State (nom)  Country
(nom)  \
 Row ID (con)          1.000000    0.976968     0.977397       0.978541
 City (nom)            0.976968    1.000000     0.959648       0.959955
 State (nom)           0.977397    0.959648     1.000000       0.974738
 Country (nom)         0.978541    0.959955     0.974738       1.000000
 Postal Code (con)     0.297431    0.982825     0.999829       0.839429
 Market (nom)          0.979708    0.948484     0.986421       0.997369
 Region (nom)          0.862192    0.945334     0.985398       0.921685
 Profit (con)         -0.019037    0.376437     0.304979       0.244427

                  Postal Code (con)  Market (nom)  Region (nom)  Profit
(con)
 Row ID (con)              0.297431      0.979708      0.862192     -0.0
19037
 City (nom)                0.982825      0.948484      0.945334      0.3
76437
 State (nom)               0.999829      0.986421      0.985398      0.3
04979
 Country (nom)             0.839429      0.997369      0.921685      0.2
44427
 Postal Code (con)         1.000000      0.839429      0.838106     -0.0
09549
 Market (nom)              0.839429      1.000000      0.882198      0.0
57222
 Region (nom)              0.838106      0.882198      1.000000      0.0
89186
 Profit (con)             -0.009549      0.057222      0.089186      1.0
00000  ,
 'ax': <matplotlib.axes._subplots.AxesSubplot at 0x20636d30dd8>}
```

1. Analyse des Attributs "Region"

1.1 Verteilung welche Region wie viele Transaktionen hat

In [13]:

```
result,colour = countColumn("Region","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 Region genug Transaktionen hat, um aussagekräftig zu sein

In [14]:

```
numberOfTransactions("Region")
```

Out[14]:

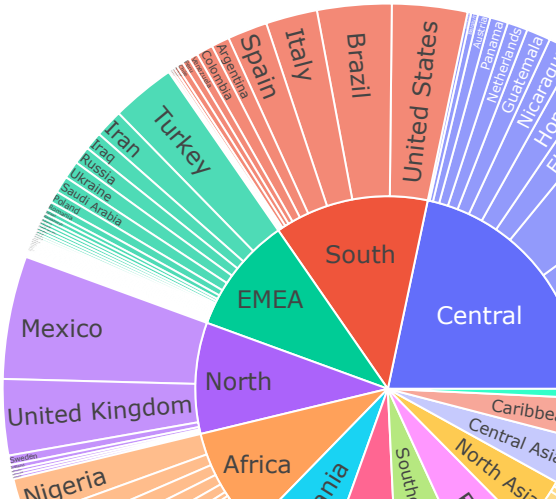
Region	Number of Transactions
Canada	384
Caribbean	1690
Central Asia	2048
North Asia	2338
East	2848
Southeast Asia	3129
West	3203
Oceania	3487
Africa	4587
North	4785
EMEA	5029
South	6645
Central	11117

1.3 Verteilung der anderen geografischen Attribute in der Region

1.3.1 Verteilung der Länder pro Region

In [15]:

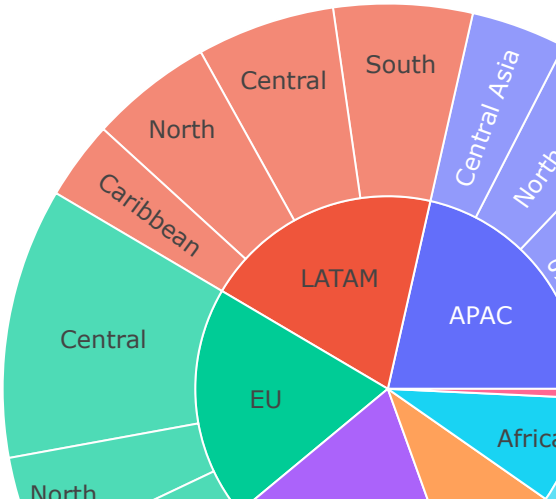
```
twoStepSunburst("Region","Country")
```



1.3.2 Verteilung der Regionen pro Markt

In [16]:

```
twoStepSunburst("Market","Region")
```



1.4 Korrelationen zwischen "Region" und "Profit"

1.4.1 Berechnungen der statistischen Korrelationswerte

Wert aus der Heatmap: 0,09

In [17]:

```
calculateAnovaScore("Region","Profit")
```

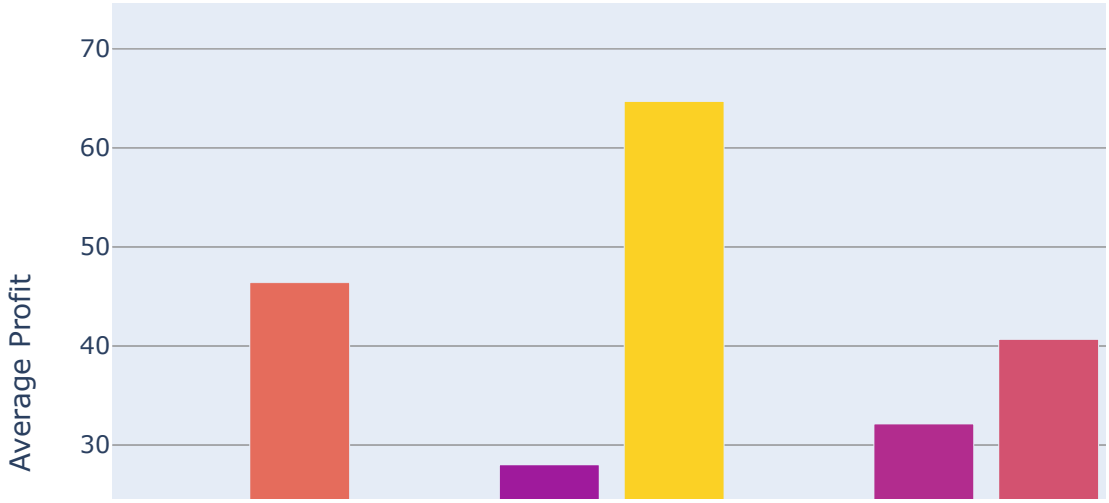
```
P-Wert für Anova ist:  3.617471983600004e-80
F Score für Anova ist:  34.261215682356564
Der Unterschied der Profit-Mittelwerte zwischen den verschiedenen Gruppen
von Region ist signifikant, da der p-Wert unter 0.05 liegt.
```

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

a) Balkendiagramm

In [18]:

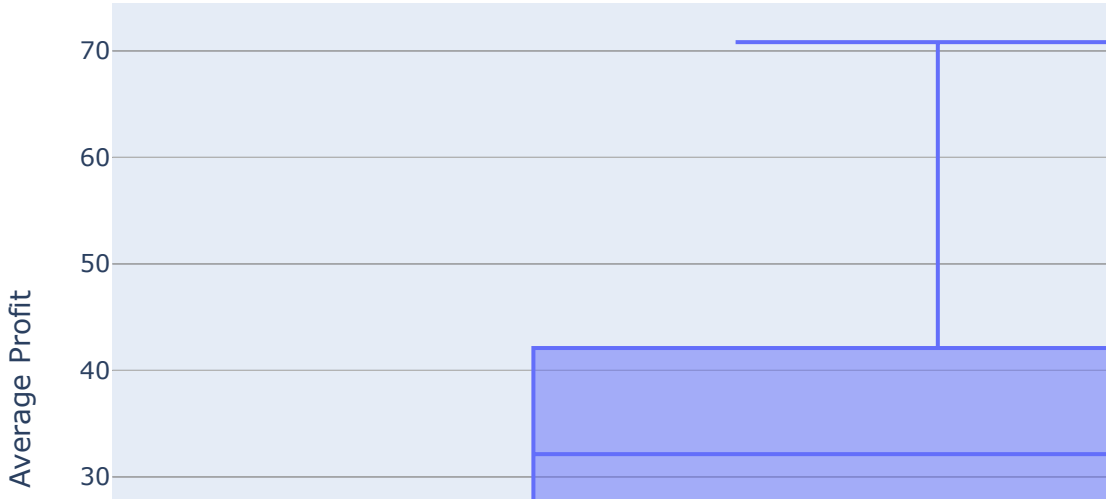
```
result,means = dfOfAverageMeans("Region","Profit", "All regions", "Average Profit")
fig = px.bar(result, x="All regions", y="Average Profit", color=means)
fig.show()
```



b) Box Plot

In [19]:

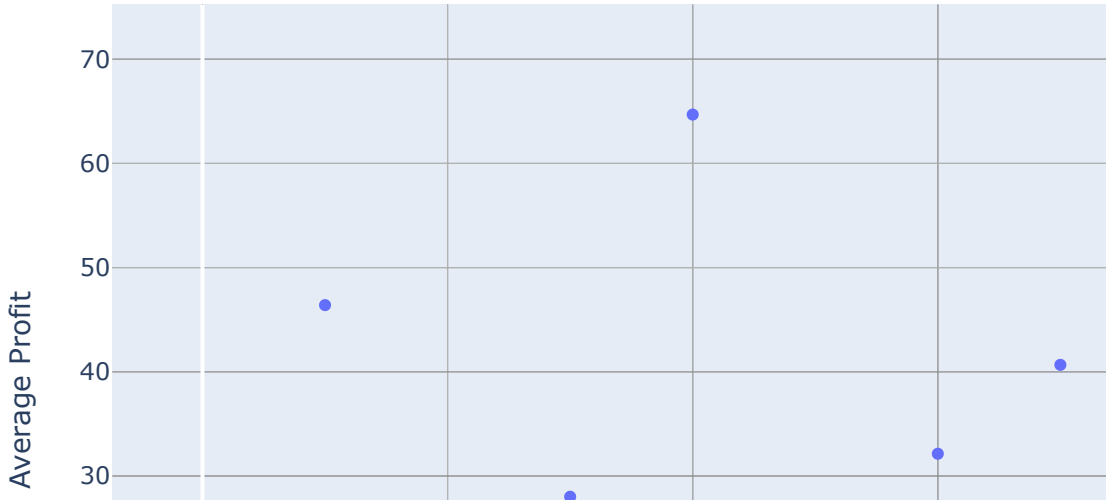
```
result,means = dfOfAverageMeans("Region","Profit", "All regions", "Average Profit")
fig = px.box(result, y="Average Profit")
fig.show()
```



c) Streudiagramm

In [20]:

```
result,means = dfOfAverageMeans("Region","Profit", "All regions", "Average Profit")
fig = px.scatter(result, y="Average Profit")
fig.show()
```



1.4.3 Fazit

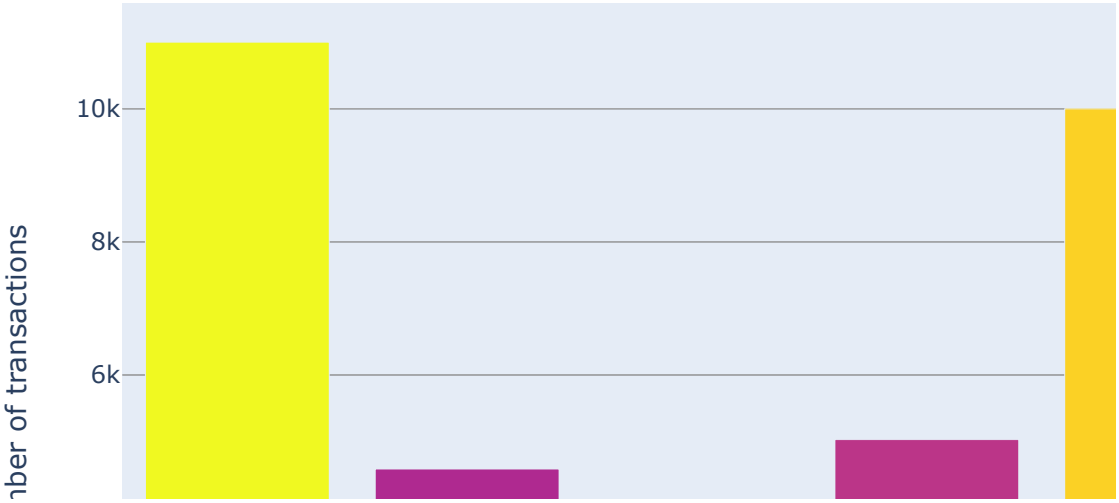
Die Ergebnisse des Anova-Tests, die hohe Anzahl an Ausprägungen bei jeder Gruppe und die in den Diagrammen erkennbare unterschiedliche Verteilung zwischen den Gruppen deutet darauf hin, dass das Attribut Region zur Vorhersage des Profits genutzt werden kann.

2. Analyse des Attributs "Market"

2.1 Verteilung welcher Market wie viele Transaktionen hat

In [21]:

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



2.2 Prüfen, ob jede Ausprägung von Market genug Transaktionen hat, um aussagekräftig zu sein

In [22]:

```
numberOfTransactions("Market")
```

Out[22]:

Market	Number of Transactions
Canada	384
Africa	4587
EMEA	5029
US	9994
EU	10000
LATAM	10294
APAC	11002

2.3 Korrelation zwischen Market und Profit

2.3.1 Berechnungen der statistschen Korrelationswerte

Wert aus der Heatmap: 0.06

In [23]:

```
calculateAnovaScore("Market","Profit")
```

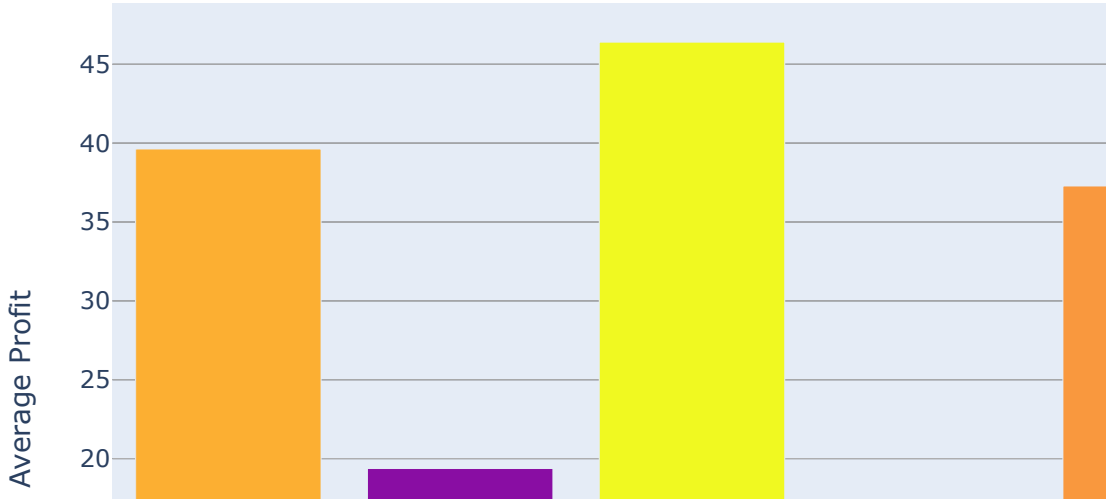
```
P-Wert für Anova ist:  1.0834601137634787e-33
F Score für Anova ist:  28.078188749595615
Der Unterschied der Profit-Mittelwerte zwischen den verschiedenen Gruppen
von Market ist signifikant, da der p-Wert unter 0.05 liegt.
```

2.3.2 Diagramme zur Prüfung der Korrelation zwischen dem durchschnittlichen Profit und Market

a) Balkendiagramm

In [24]:

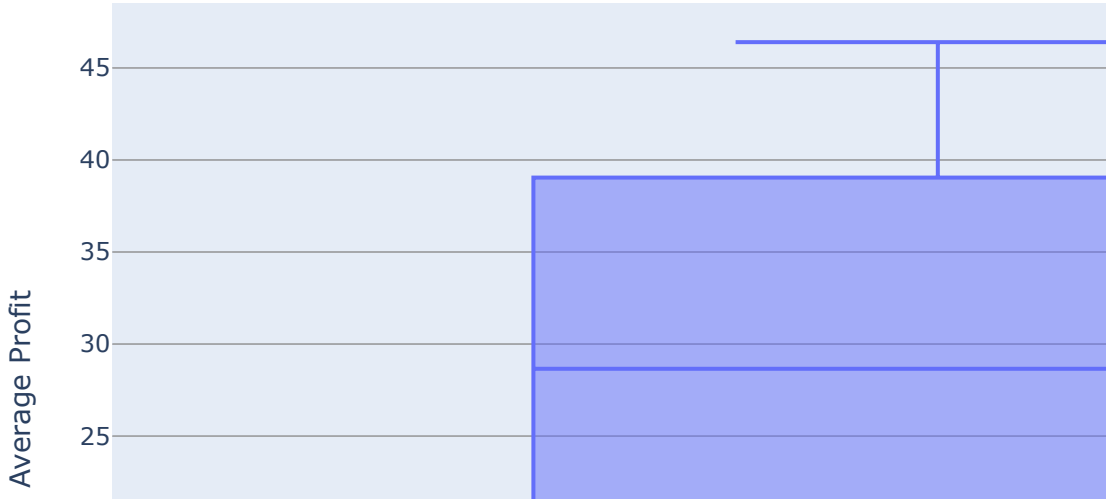
```
result,means = dfOfAverageMeans("Market","Profit", "All markets", "Average Profit")
fig = px.bar(result, x="All markets", y="Average Profit", color=means)
fig.show()
```



b) Box Plot

In [25]:

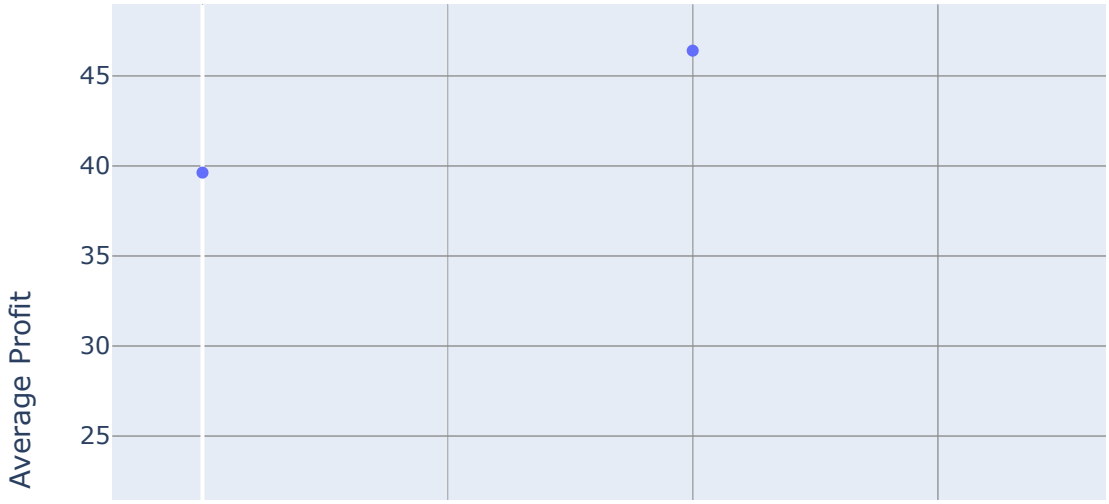
```
result,means = dfOfAverageMeans("Market","Profit", "All markets", "Average Profit")
fig = px.box(result, y="Average Profit")
fig.show()
```



c) Streudiagramm

In [26]:

```
result,means = dfOfAverageMeans("Market","Profit", "All market", "Average Profit")
fig = px.scatter(result, y="Average Profit")
fig.show()
```



2.3.3 Fazit

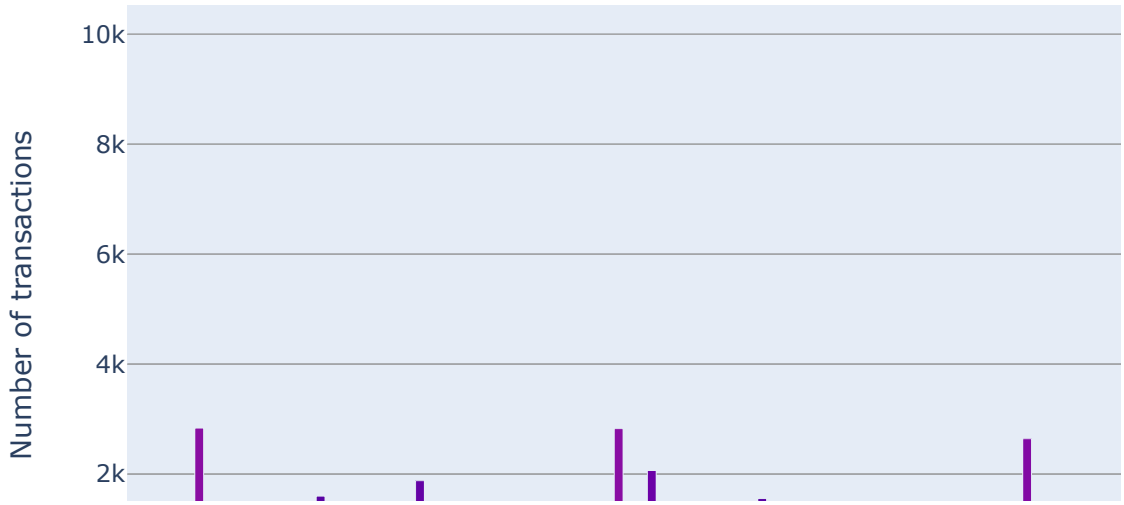
Der P-Wert ist etwas größer und der F-Score etwas kleiner als bei Region --> Tendiert also eher dazu die Null Hypothese abzulehnen. Dennoch ist der Wert sehr klein und auch der F Score ist noch relativ hoch. Die statistisch signifikant unterschiedlichen MIttelwerte in Verbindung mit den Diagrammen , die eine starke Unterscheidung zwischen den Gruppen zeigen und der Tatsache, dass auch hier jede Gruppe eine Mindestmenge von Beispielen hat, spricht dafür, dass auch Market zur Vorhersage des Profits genutzt werden kann.

3. Analyse Country

3.1 Verteilung welches country wie viele Transaktionen hat

In [27]:

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



Es macht den Eindruck, dass die Verteilung der Transaktionen sehr ungleich ist. Es gibt also viele countries, die zu selten vorkommen, um eine verlässliche Vorhersage spielen zu können.

3.2 Prüfen wie oft es vorkommt, dass ein country nur wenige Transaktionen hat

In [28]:

```
howOftenDoAmountsAppear("Country")
```

Out[28]:

	Number of Transactions	counts
0	2	6
1	3	4
2	4	2
3	6	2
4	7	3
...	...	...
108	2065	1
109	2644	1
110	2827	1
111	2837	1
112	9994	1

113 rows × 2 columns

counts zählt hier wie oft die Häufigkeit eines bestimmten country vorkommt. Es gibt also 6 countries, die nur 2 mal vorkommen, 4 countries, die nur 3 mal vorkommen und so weiter. Dies zeigt, dass ggf. das country keine zuverlässige Vorhersage erlaubt, da viele Ausprägungen von country zu selten vorkommen.

3.3 Korrelation zwischen Country und Profit

3.3.1 Berechnungen der statistischen Korrelationswerte

Wert aus der Heatmap: 0.24

In [29]:

```
calculateAnovaScore("Country","Profit")
```

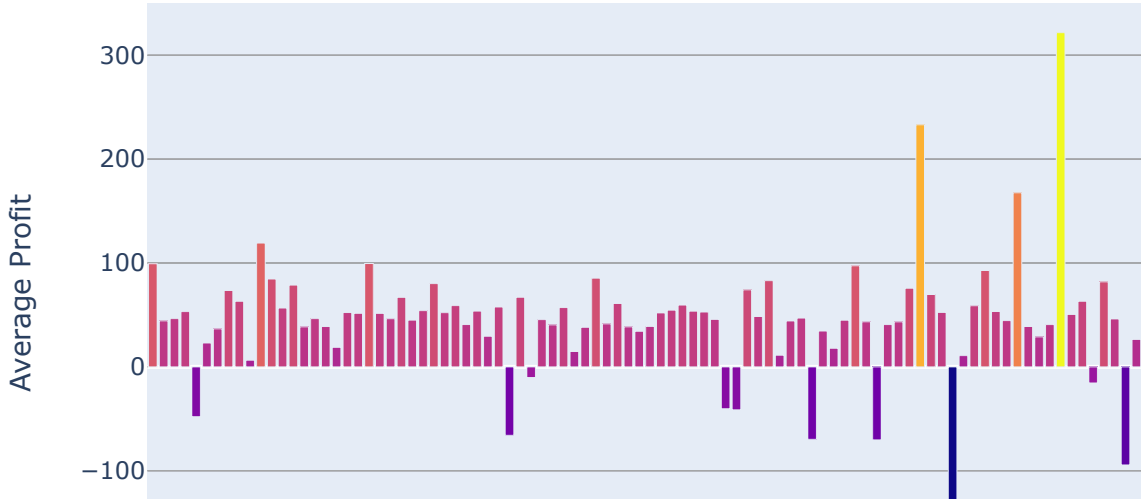
```
P-Wert für Anova ist:  0.0
F Score für Anova ist:  22.25802770920223
Der Unterschied der Profit-Mittelwerte zwischen den verschiedenen Gruppen
von Country ist signifikant, da der p-Wert unter 0.05 liegt.
```

3.3.2 Diagramme zur grafischen Verbindung zwischen Country und und dem durchschnittlichen Profit

a) Balkendiagramm

In [30]:

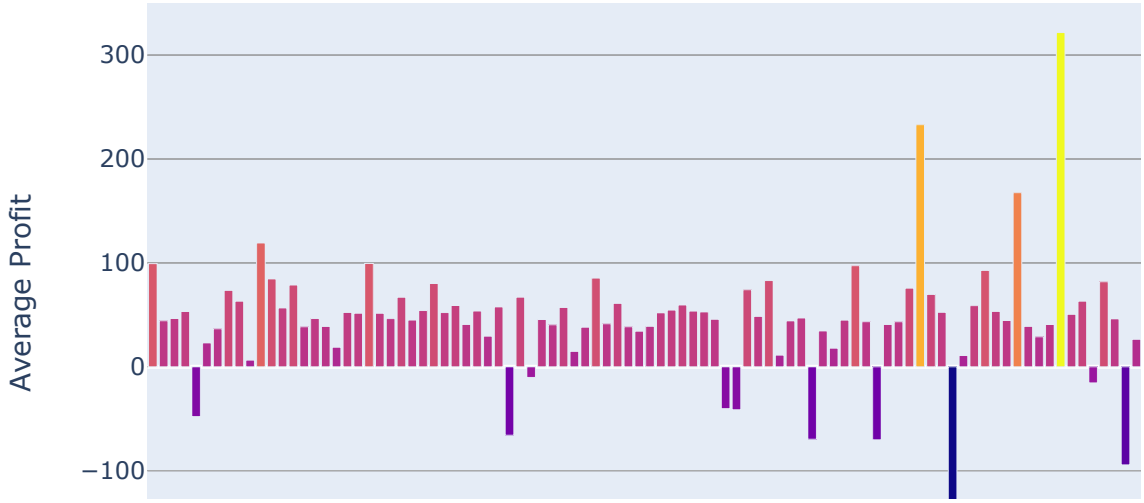
```
result,means = dfOfAverageMeans("Country","Profit", "All countries", "Average Profit")
fig = px.bar(result, x="All countries", y="Average Profit", color=means)
fig.show()
```



b) Box Plot

In [31]:

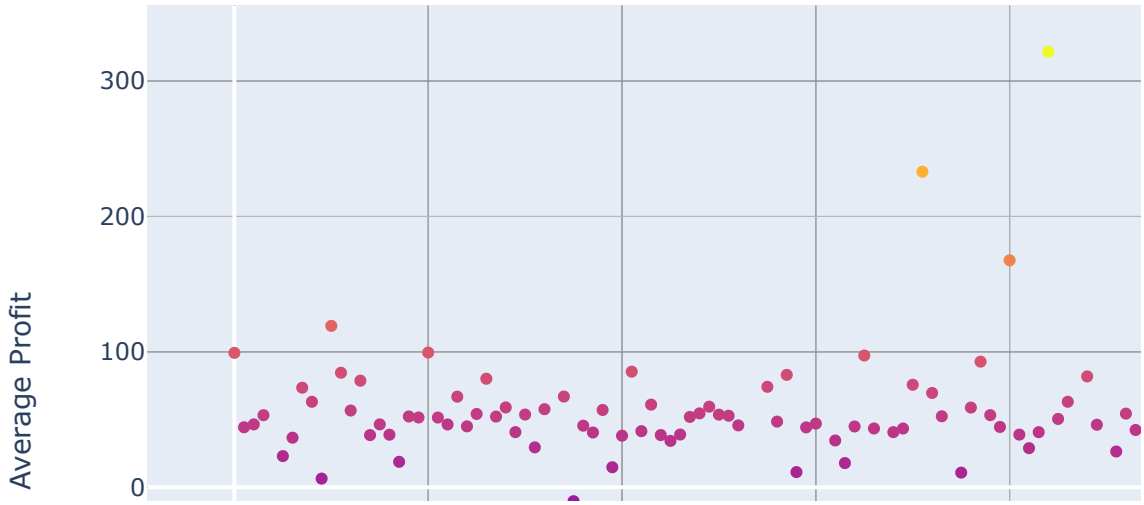
```
result,means = dfOfAverageMeans("Country","Profit", "All countries", "Average Profit")
fig = px.bar(result, x="All countries", y="Average Profit", color=means)
fig.show()
```



c) Streudiagramm

In [32]:

```
result,means = dfOfAverageMeans("Country","Profit", "All countries", "Average Profit")
fig = px.scatter(result, y="Average Profit", color=means)
fig.show()
```



3.3.3 Fazit

Der P-Wert des Anova Tests ist so klein, dass er einfach nur 0 ist. Die kommt daher, dass man so verschiedene Mittelwerte hat, dass die Gruppen in nahezu keiner Beziehung mehr stehen. Der etwas höhere Wert aus der Heatmap kommt daher, dass es viele Gruppe mit wenig Werten gibt. Bei diesen ist innerhalb der Gruppe die Korrelaton groß, weshalb der Gesamtwert höher ist. Außerdem gibt es das Problem, dass viele Ausprägungen von country zu wenig Transaktionen haben und daher keine Aussagekraft haben. Dennoch gibt es Werte, die eine hohe Aussagekraft haben. Country ist also ein 50/50. --> Da es im Sachzusammenhang aber relevant und interessant zu betrachten ist, wird es mit aufgenommen.

4. Analyse des Attributs State

4.1 Anzahl Transaktionen pro State

In [33]:

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

Diagramm schon zu breit wegen zu vieler verschiedener Balken --> Darstellung im DataFrame:

In [34]:

```
numberOfTransactions("State")
```

Out[34]:

State	Number of Transactions
Rize	1
Iringa	1
Pernik	1
Pleven	1
Inhambane	1
...	...
Ile-de-France	981
Texas	985
New York	1128
England	1499
California	2001

1094 rows × 1 columns

Es gibt über 1000 verschiedene Grppen und scheinbar haben viele zu wenige Transaktionen. Prüfung:

In [35]:

```
howOftenDoAmountsAppear("State")
```

Out[35]:

	Number of Transactions	counts
0	1	64
1	2	74
2	3	55
3	4	58
4	5	42
...	...	...
178	981	1
179	985	1
180	1128	1
181	1499	1
182	2001	1

183 rows × 2 columns

64 mal kommt ein state nut einmal vor. 74 mal nur zweifach, 55 mal nur dreifach... Das Attribut state scheint zu viele verschiedene Gruppen zu haben, bei denen viele nur selten vorkommen. Die Verteilung ist in zu viele kleine Stücke aufgeteilt. Damit besthet eine hohe Gefahr der Verzerrung durch Ausreißer

4.2 Korrelation zwischen State und Profit

4.2.1 Rechnerische Bestimmung der Korrelationswerte

Wert aus der Heatmap: 0,3

In [36]:

```
calculateAnovaScore("State","Profit")
```

```
P-Wert für Anova ist:  0.0
F Score für Anova ist:  4.709630866402647
Der Unterschied der Profit-Mittelwerte zwischen den verschiedenen Gruppen
von State ist signifikant, da der p-Wert unter 0.05 liegt.
```

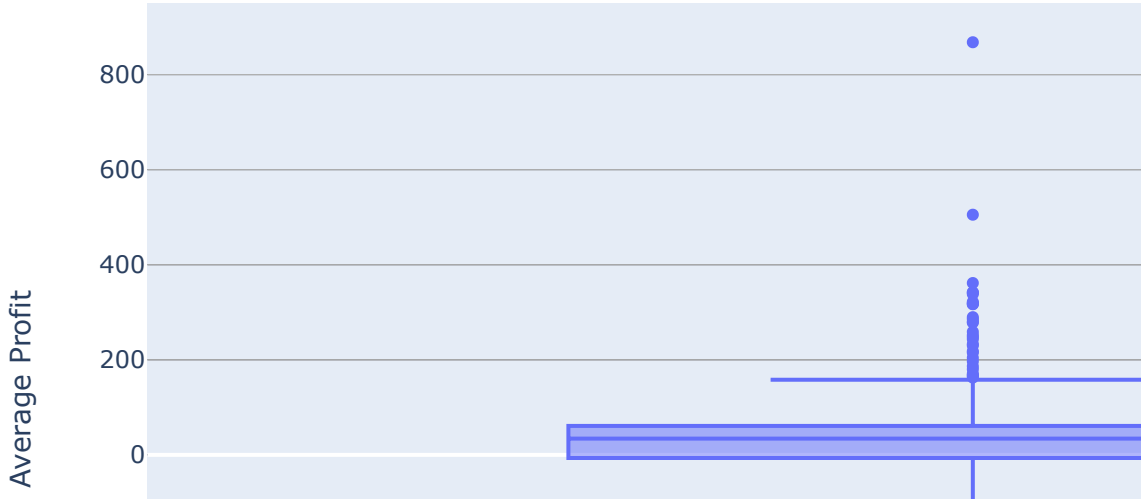
4.2.2 Diaramme zur grafischen Bestimmung von Korrelationen zwischen State und dem durchschnittlichen Profit

a) Balkendiagramm: zu viele einzelne Balken --> Diagramm nicht lesbar

b) Box Plot

In [37]:

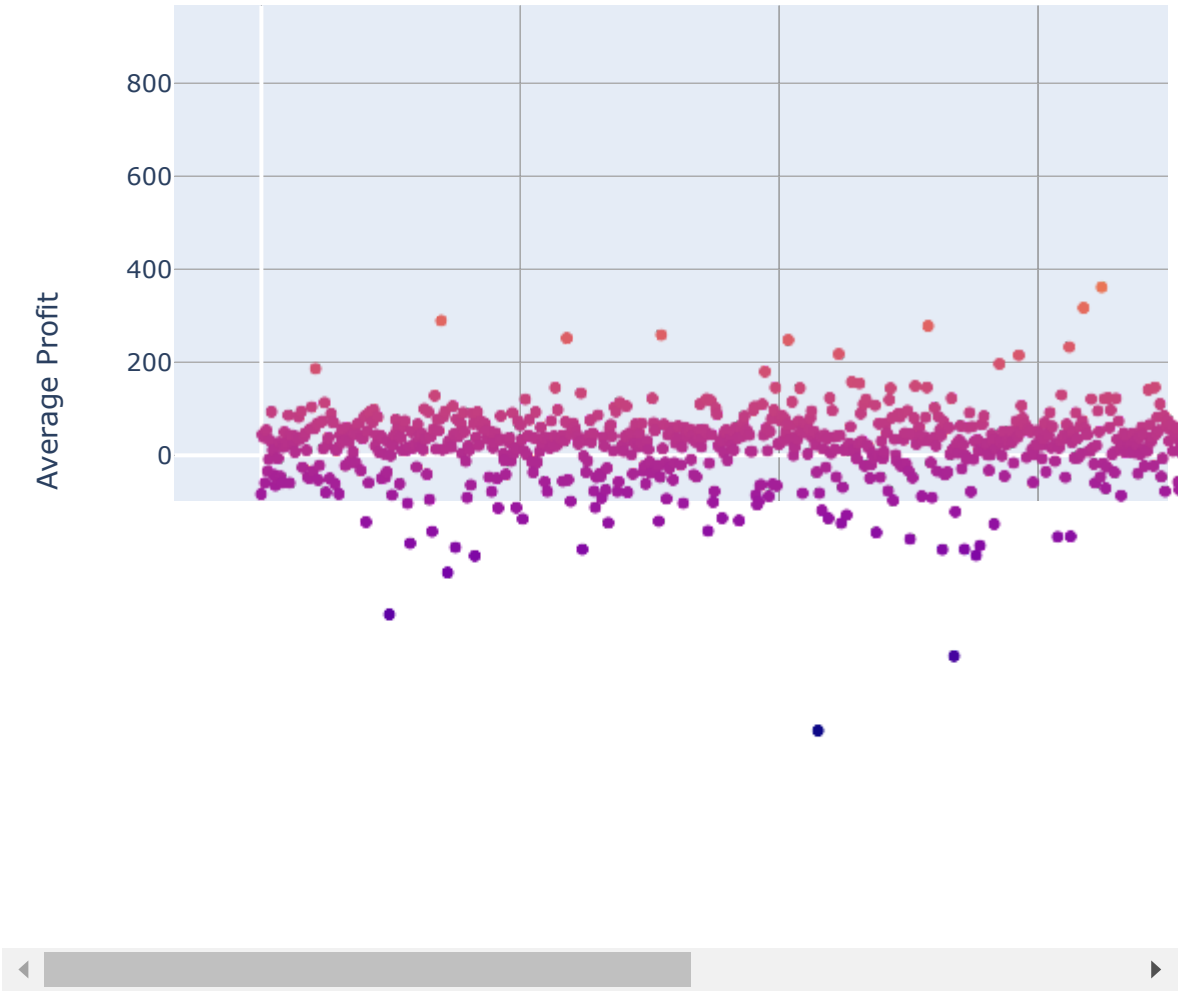
```
result,means = dfOfAverageMeans("State","Profit", "All states", "Average Profit")
fig = px.box(result, y="Average Profit")
fig.show()
```



c) Streudiagramm

In [38]:

```
result,means = dfOfAverageMeans("State","Profit", "All states", "Average Profit")
fig = px.scatter(result, y="Average Profit", color=means)
fig.show()
```



4.3 Fazit

Das Attribut State hat viele Gruppen, die sehr wenige Ausprägungen haben und ist daher für die Vorhersage des Profits nicht geeignet, da die Wahrscheinlichkeit von Ausreißerwerten zu hoch ist. Problem der Ausreißer: Wenn es nur einmal den State Iringa gibt und bei diesem der Profit 100000 wäre, würde der Algorithmus einen hohen Profit garantieren, sobald der State Iringa ist. Da dieser Wert aber nur ein Ausreißer ist, kann nicht mit Gewissheit davon ausgegangen werden, dass jede andere Transaktion in diesem State zum ähnlichen Profit führt. Die Berechnungen der statistischen Abhängigkeiten bestätigen diese Annahmen. Daher wird das Attribut State entfernt

In [39]:

```
del df["State"]
```

5. Analyse des Attributs city

5.1 Anzahl der Transaktionen pro city

In [40]:

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

In [41]:

```
numberOfTransactions("City")
```

Out[41]:

City	Number of Transactions
Río Bravo	1
Montería	1
Sikasso	1
Chengjiang	1
Cheonan	1
...	...
Santo Domingo	443
San Francisco	510
Philadelphia	537
Los Angeles	747
New York City	915

3636 rows × 1 columns

In [42]:

```
howOftenDoAmountsAppear("City")
```

Out[42]:

	Number of Transactions	counts
0	1	488
1	2	419
2	3	349
3	4	293
4	5	234
...	...	...
151	443	1
152	510	1
153	537	1
154	747	1
155	915	1

156 rows × 2 columns

Es gibt über 3600 verschiedene Ausprägungen von City, bei denen 488 nur einmal vorkommen, 419 nur zweimal vorkommen und so weiter. Dies deutet darauf hin, dass City ein ungeeignetes Attribut ist (wegen zu großer Streuung der Gruppen)

5.2 Korrelationen zwischen State und Profit

5.2.1 Rechnerische Bestimmung der Korrelation

Wert aus Heatmap: 0,38

In [43]:

```
calculateAnovaScore("City","Profit")
```

```
P-Wert für Anova ist:  1.9283854766021634e-276
F Score für Anova ist:  2.164422870379695
Der Unterschied der Profit-Mittelwerte zwischen den verschiedenen Gruppen
von City ist signifikant, da der p-Wert unter 0.05 liegt.
```

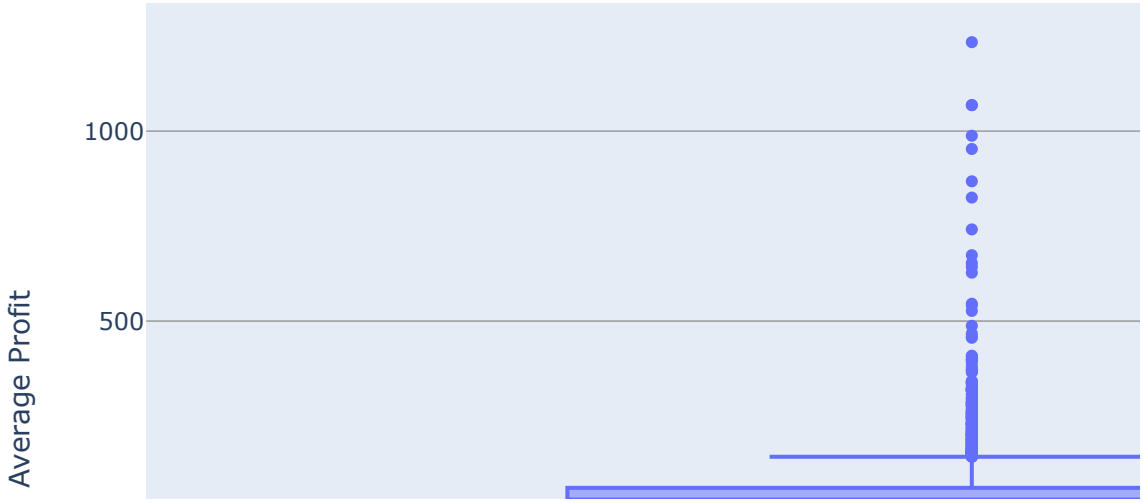
5.2.2 Grafische Prüfung der Korrelation

a) Balkendiagramm: zu viele Balken --> nicht mehr erkennbar

b) Box Plot

In [44]:

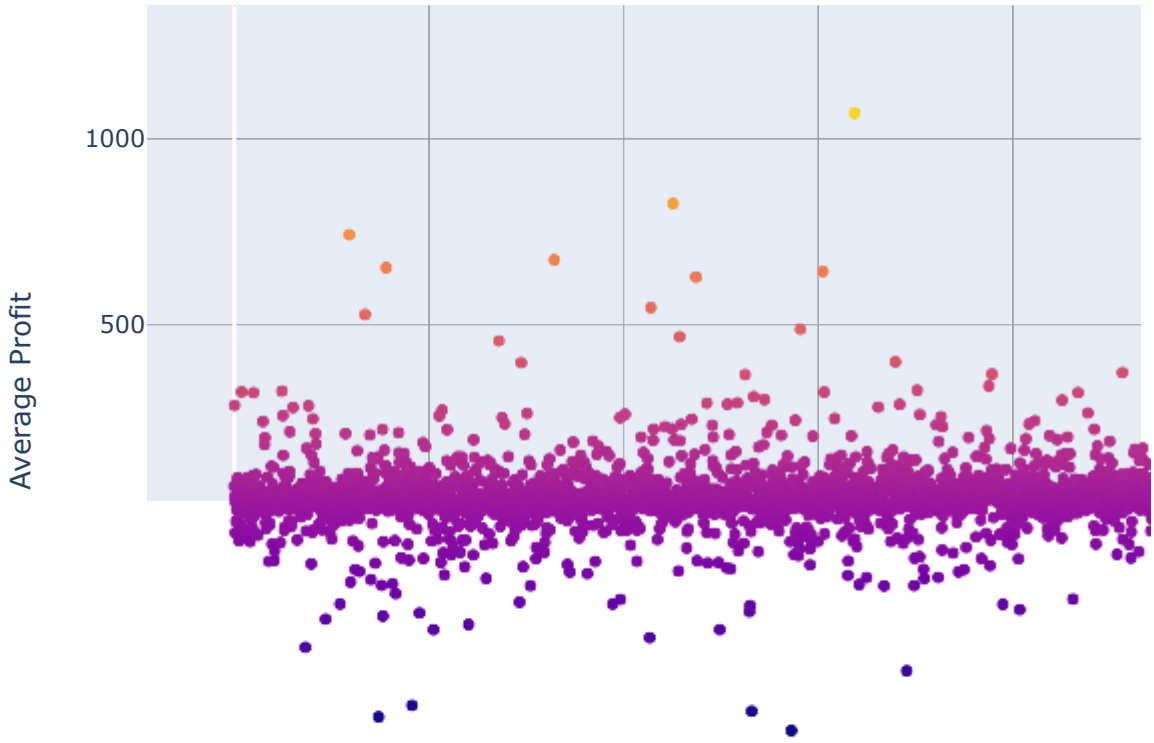
```
result,means = dfOfAverageMeans("City","Profit", "All cities", "Average Profit")
fig = px.box(result, y="Average Profit")
fig.show()
```



c) Streudiagramm

In [45]:

```
result,means = dfOfAverageMeans("City","Profit", "All cities", "Average Profit")
fig = px.scatter(result, y="Average Profit", color=means)
fig.show()
```



5.3 Fazit

Auch das Attribut City hat (genau wie State) zu viele verschiedene Gruppen mit jeweils viel zu wenigen Transaktionen pro Gruppe. Dieser Zusammenhang zeigt sich auch in den Korrelationswerten und Diagrammen (die Streuung ist zu groß um verlässliche Vrhersagen zu treffen). Die Argumentation zur Löschung von State gilt auch hier.

In [46]:

```
del df["City"]
```

6. Analyse des Attributs Postal Code

In [47]:

```
df['Postal Code']
```

Out[47]:

```
0        10024.0
1            NaN
2            NaN
3            NaN
4            NaN
          ...
51285        NaN
51286    77095.0
51287    93030.0
51288        NaN
51289        NaN
Name: Postal Code, Length: 51290, dtype: float64
```

In [48]:

```
df['Postal Code'].isna().sum()
```

Out[48]:

```
41296
```

Von den 51290 haben 41296 Transaktionen keinen Wert bei Postal Code. Da dies die Aussagekraft enorm mindert, wird es entfernt

In [49]:

```
del df["Postal Code"]
```

Überarbeitetes DataFrame der geografischen Attribute:

In [50]:

```
df
```

Out[50]:

	Row ID	Country	Market	Region	Profit
0	32298	United States	US	East	762.1845
1	26341	Australia	APAC	Oceania	-288.7650
2	25330	Australia	APAC	Oceania	919.9710
3	13524	Germany	EU	Central	-96.5400
4	47221	Senegal	Africa	Africa	311.5200
...	...	...	...	...	...
51285	29002	Japan	APAC	North Asia	4.5000
51286	35398	United States	US	Central	-1.1100
51287	40470	United States	US	West	11.2308
51288	9596	Brazil	LATAM	South	2.4000
51289	6147	Nicaragua	LATAM	Central	1.8000

51290 rows × 5 columns