

Análise demográfica e clínica da obesidade

Hipótesis

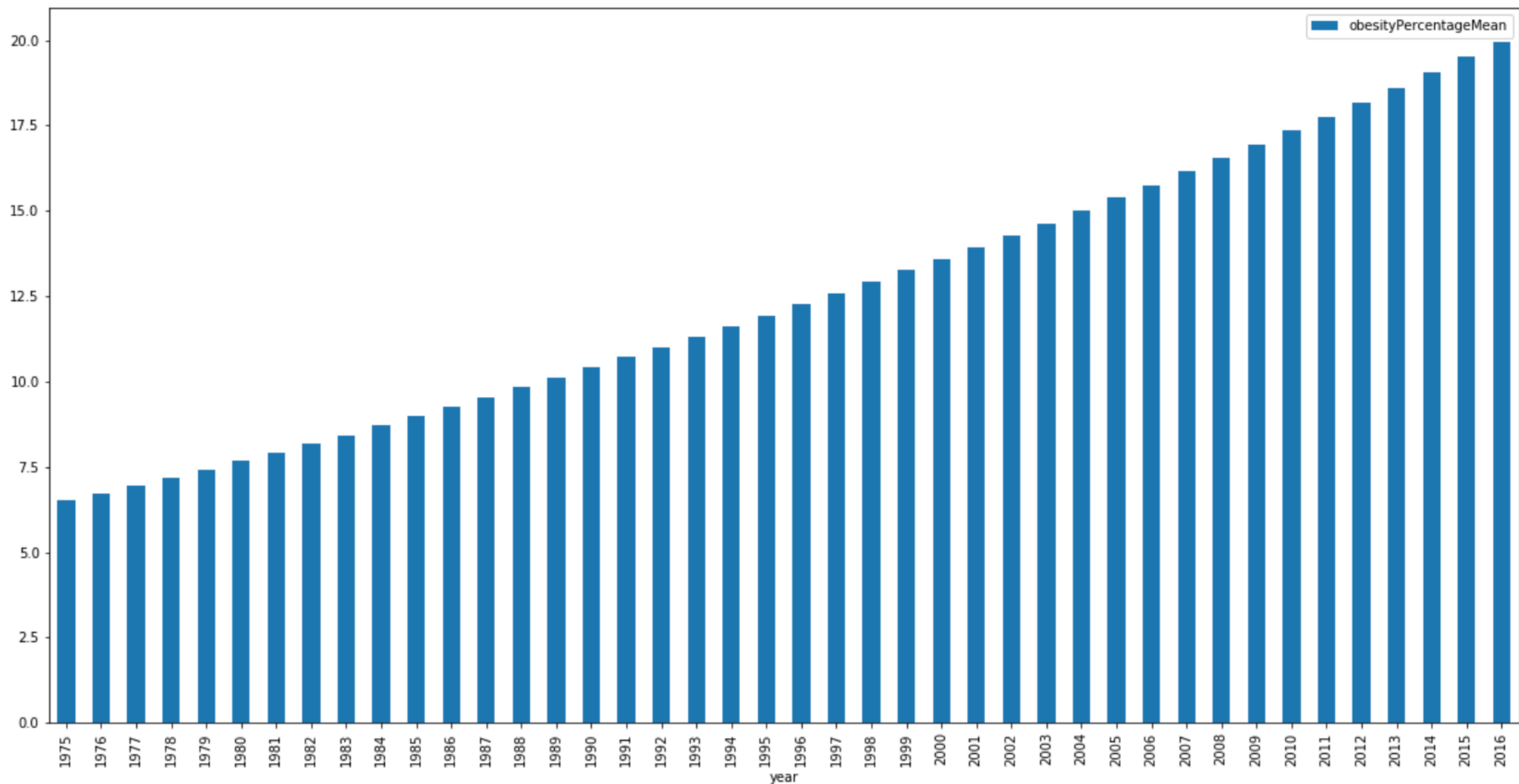
O crescimento da obesidade está diretamente relacionado ao crescimento da diabetes.

Quanto mais alto o IDH de um país, maior o índice de obesidade

A quantidade de vegetais ingeridos influencia diretamente na taxa de obesidade do país

Características clínicas que não o peso são suficientes para classificar uma pessoa como obesa

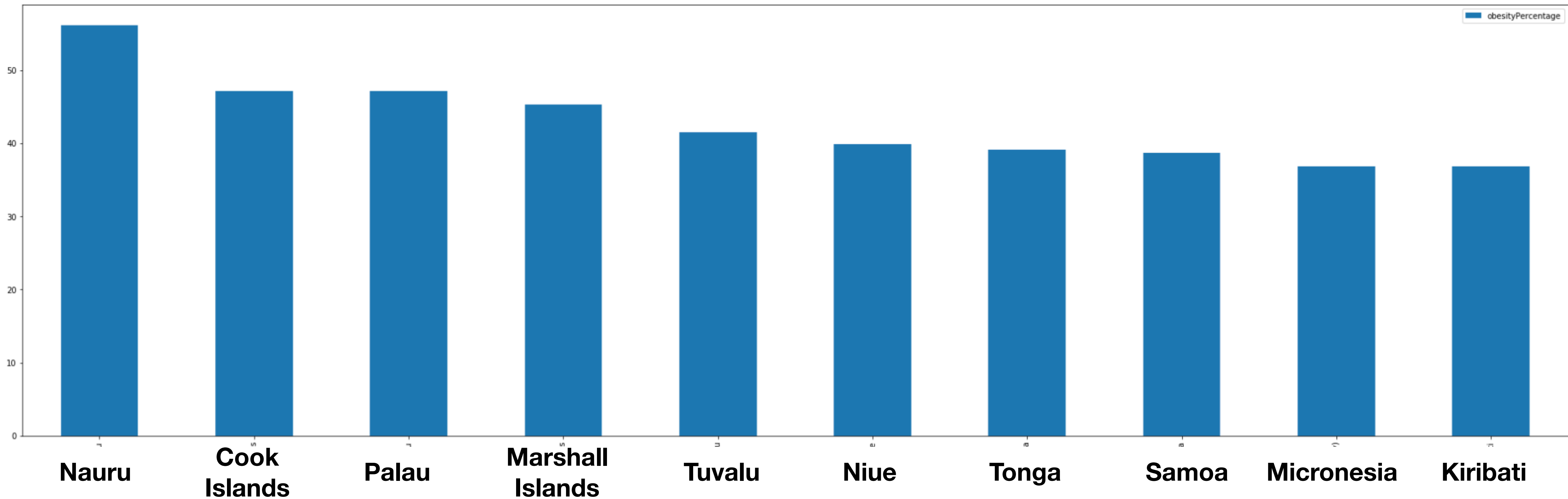
Evolução do percentual de obesos no mundo a cada ano



Top 10 países com maior percentual de obesos

Percentual médio desde 1990

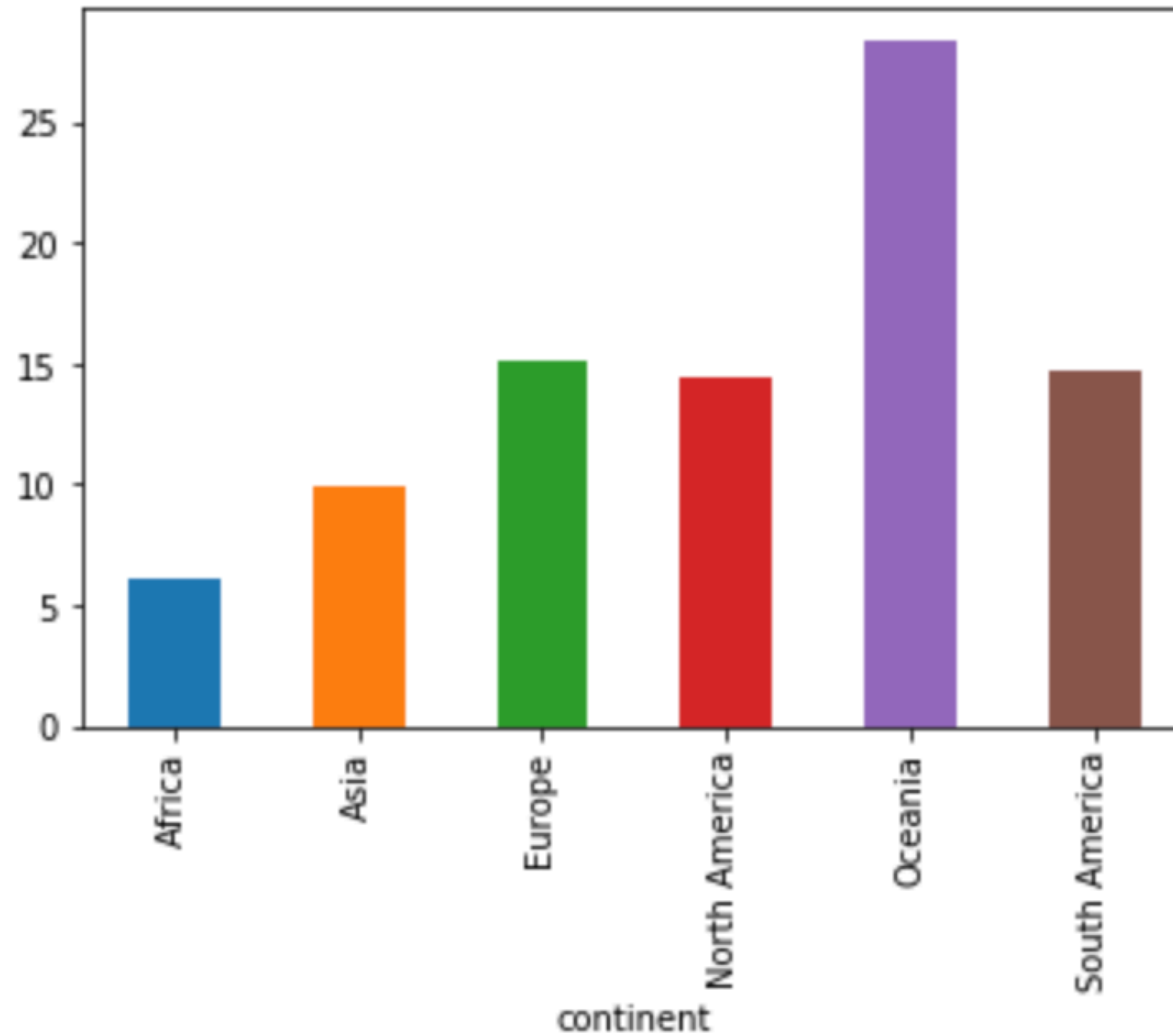
60%



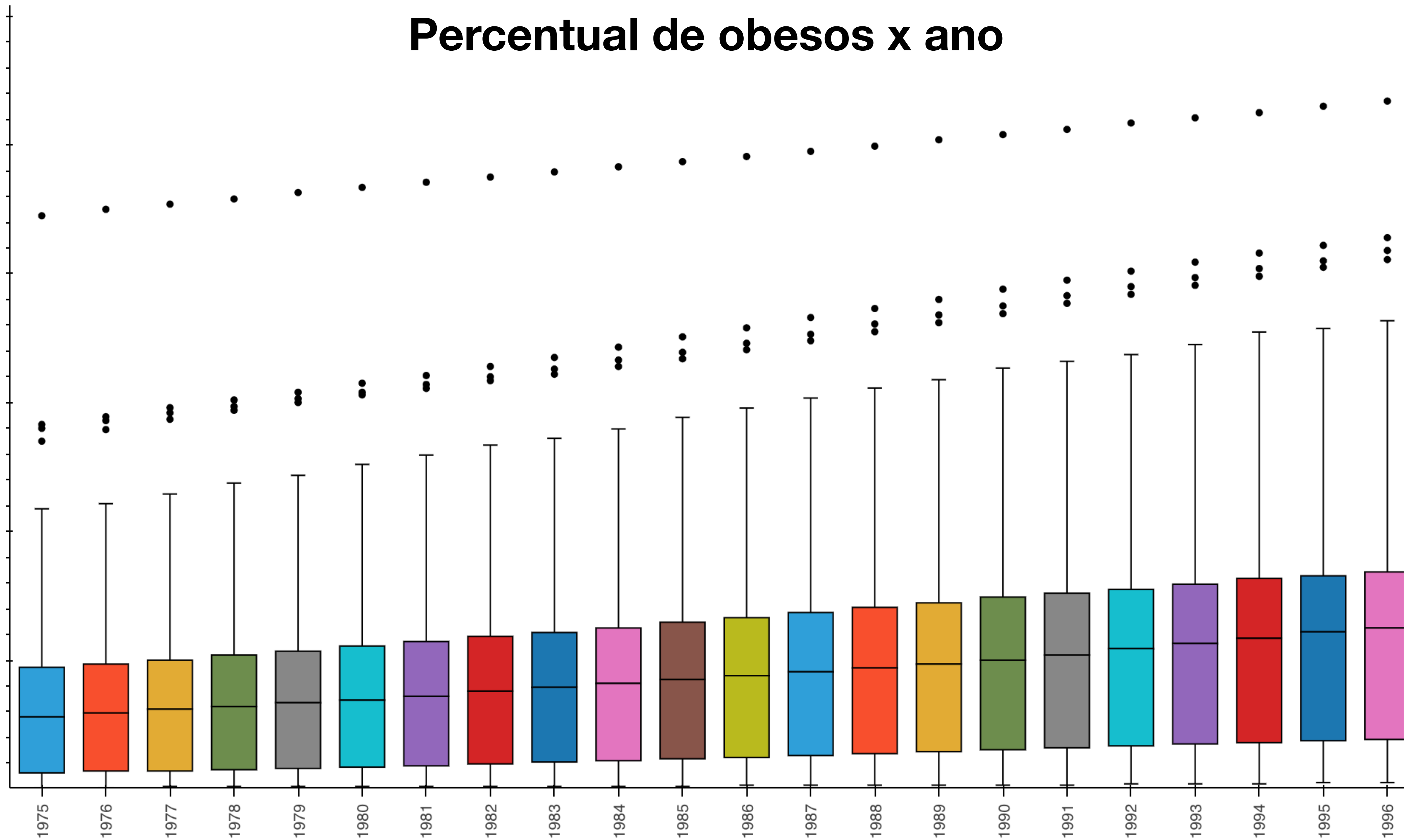
Nauru



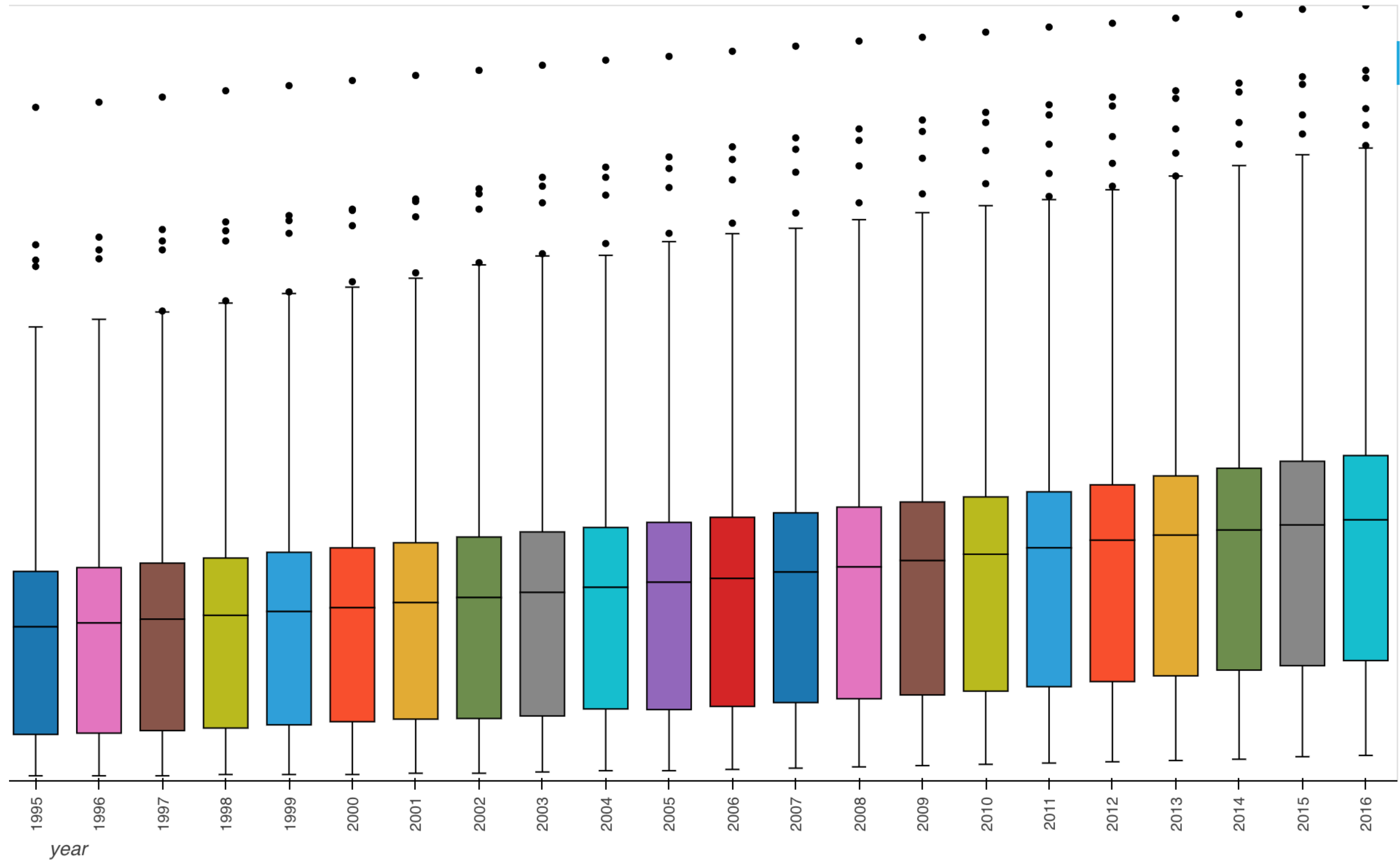
Percentual médio de obesos por continente



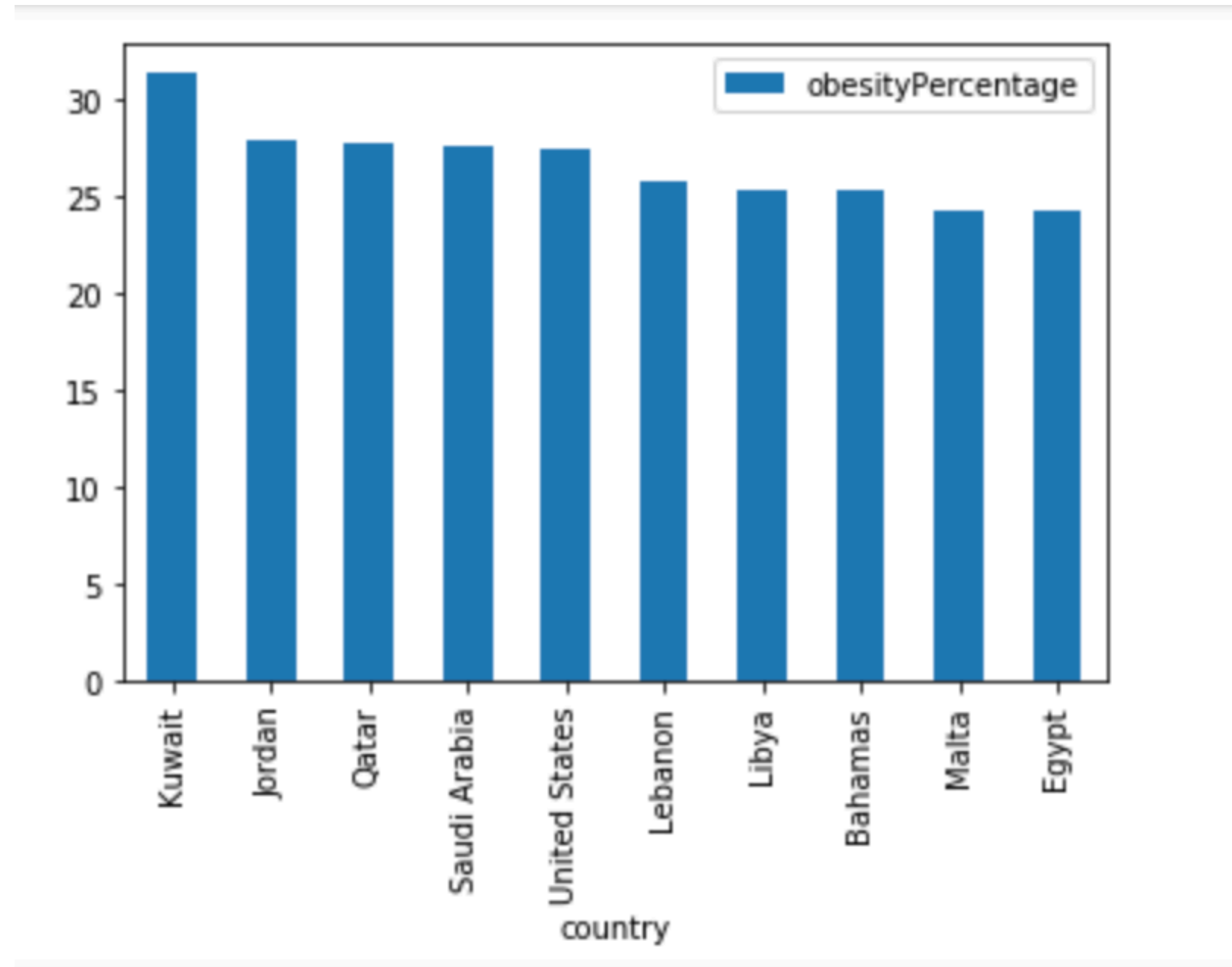
Percentual de obesos x ano



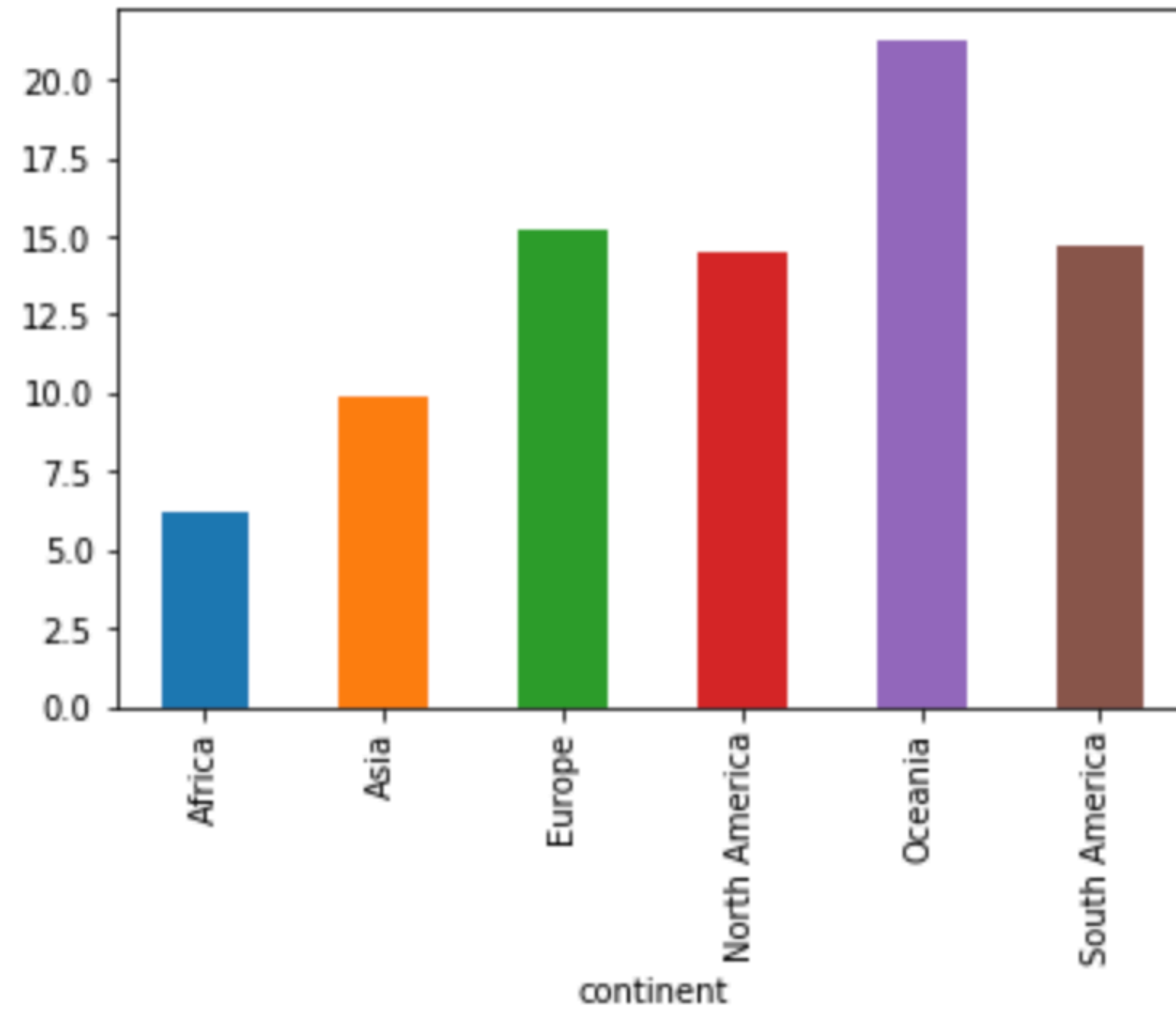
Percentual de obesos x ano



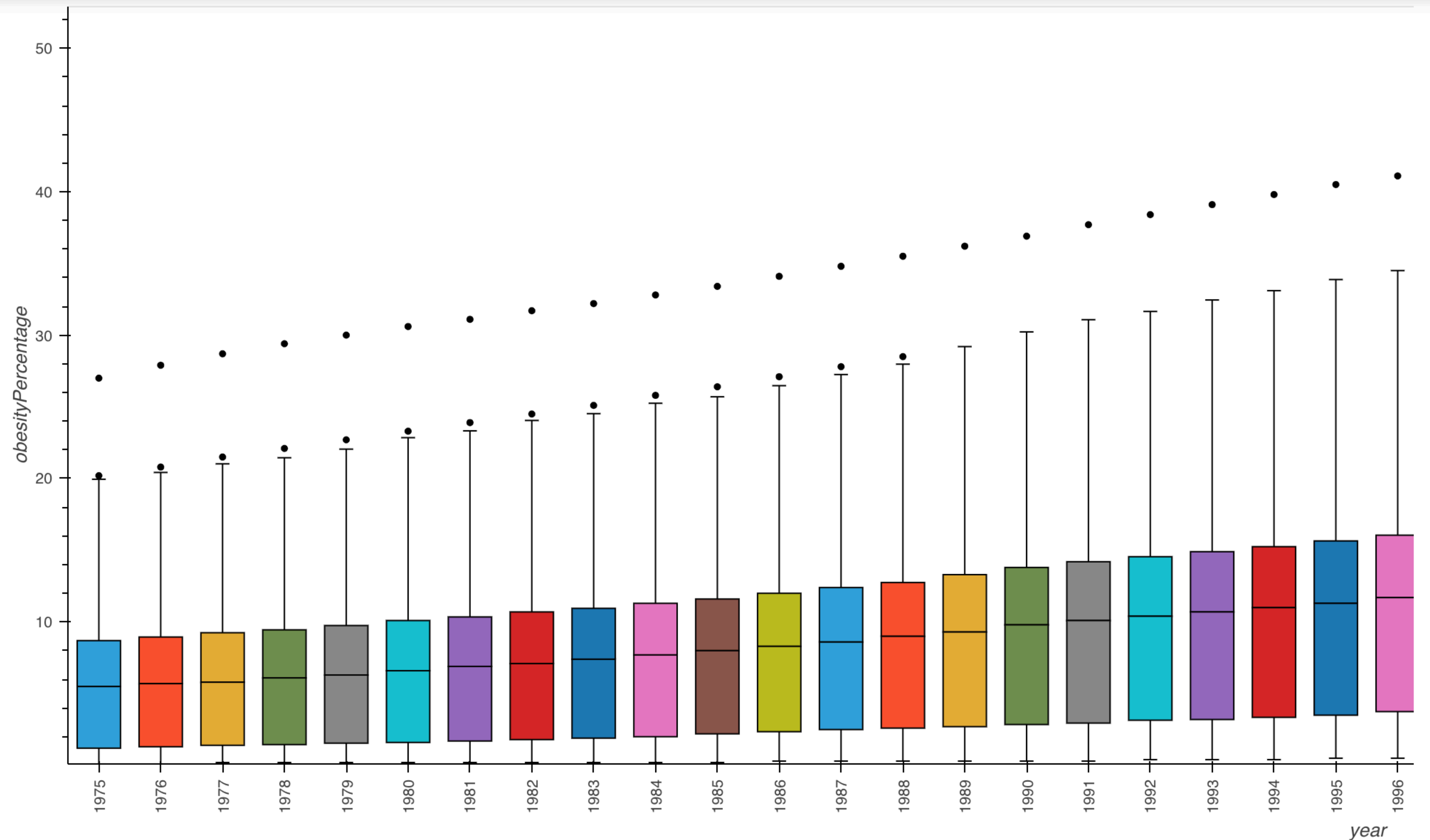
Top 10 países com maior percentual de obesos após remover outliers



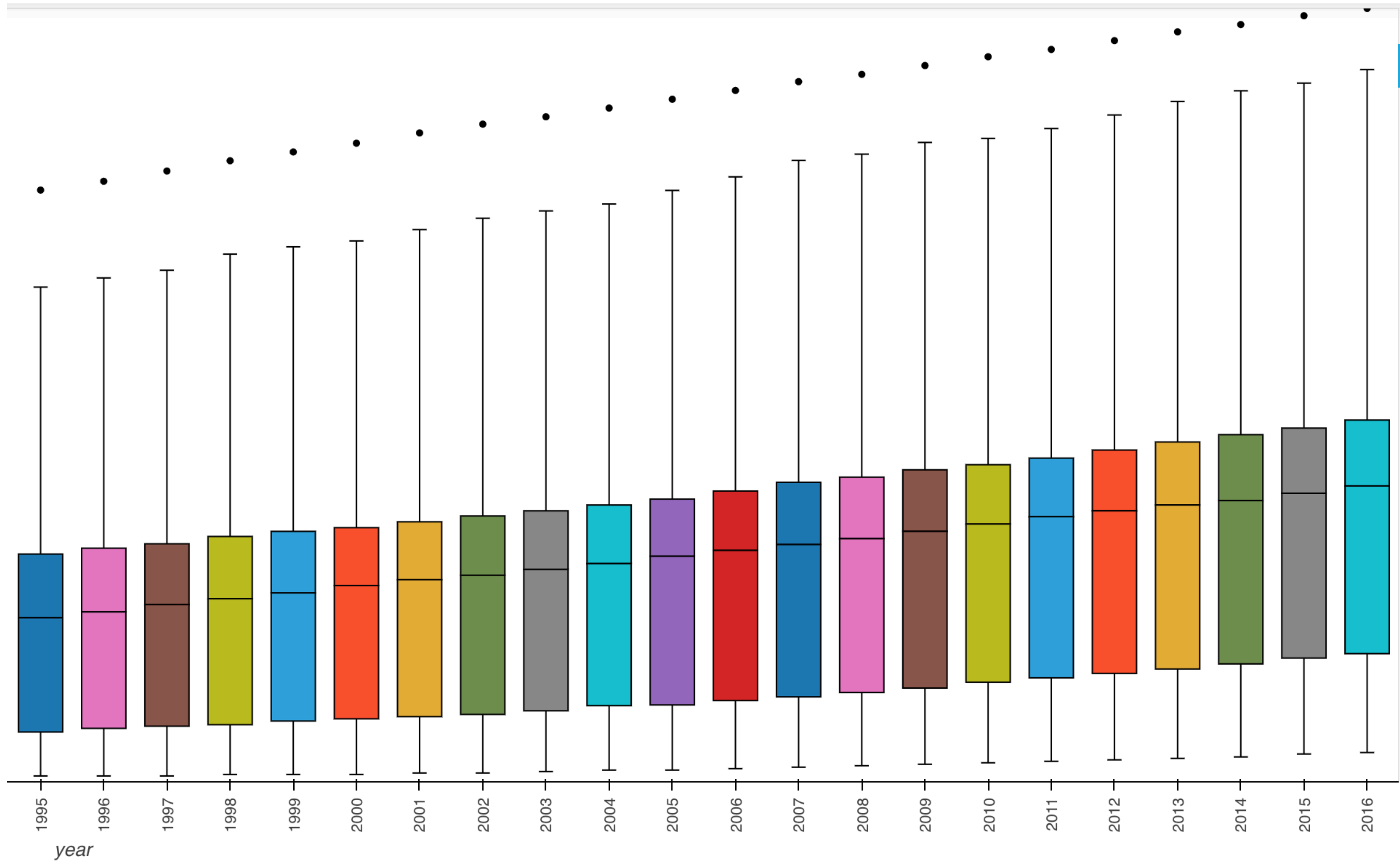
Analise de obesos por continente após remover outliers



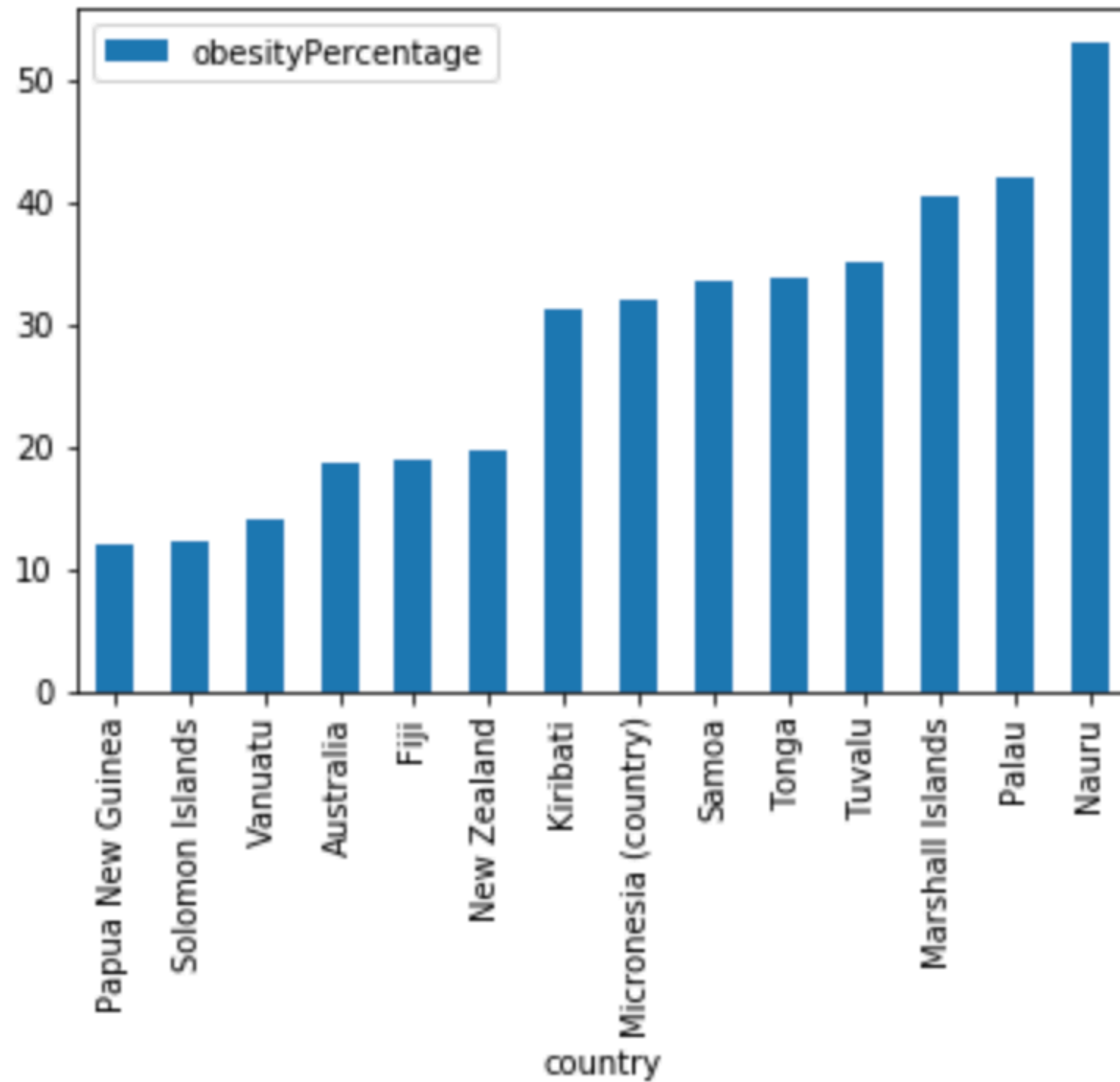
Percentual de obesidade por ano após remover outliers



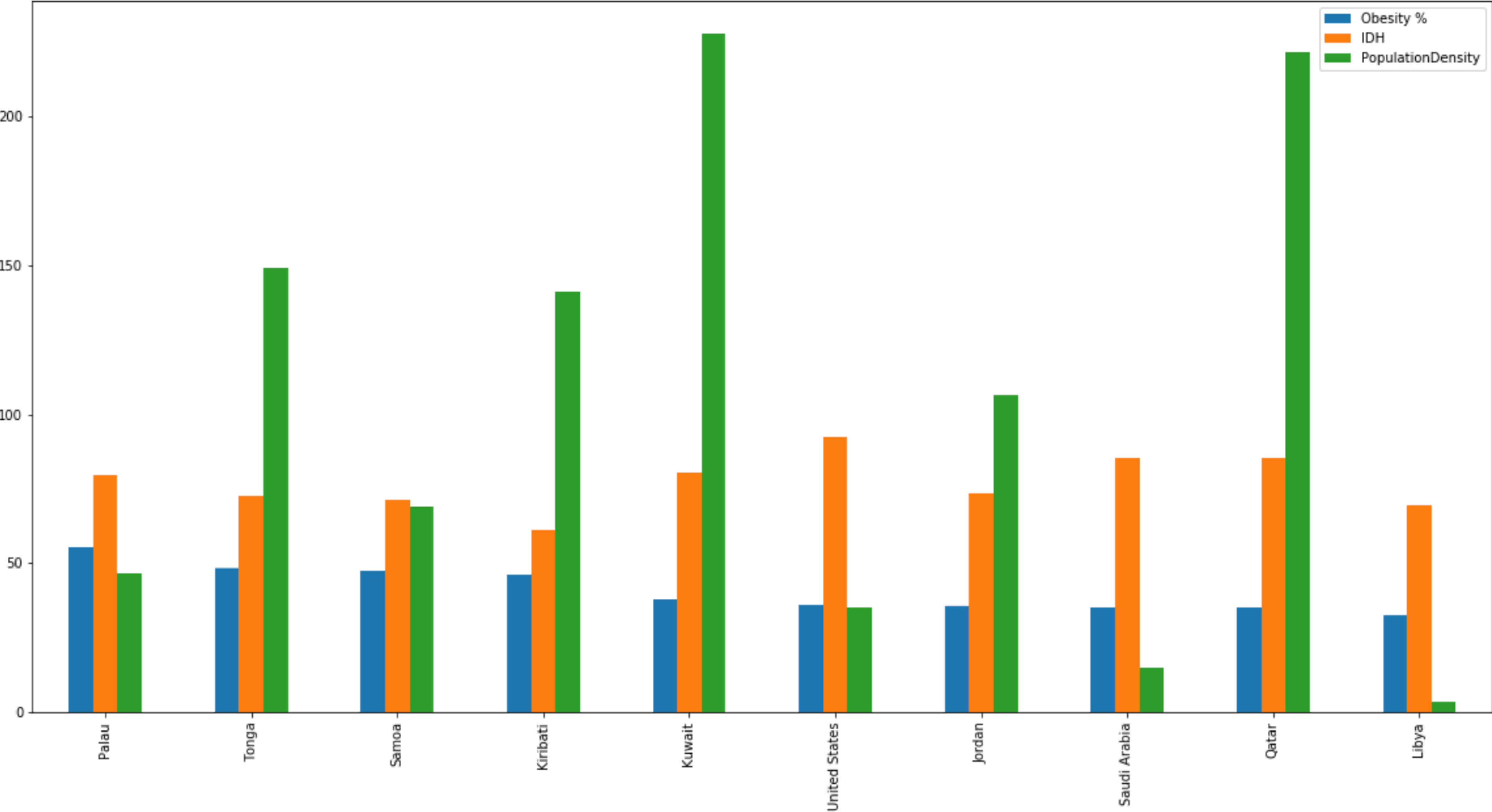
Percentual de obesidade por ano após remover outliers



Análise mais detalhada da Oceania



Visualização relação percentual de obesos, IDH e densidade populacional



Correlação entre percentual de obesos e densidade populacional

	Obesity %	IDH	PopulationDensity
Obesity %	1.000000	0.562613	-0.119254
IDH	0.562613	1.000000	0.138122
PopulationDensity	-0.119254	0.138122	1.000000

Correlação consumo de vegetais e percentual de obesos em um país

	year	obesityPercentage	avgVegIntake
year	1.000000	0.311797	0.117562
obesityPercentage	0.311797	1.000000	0.394922
avgVegIntake	0.117562	0.394922	1.000000

Modelo de aprendizado para classificar uma pessoa como obesa

```
In [99]: gnb.fit(X_train_res, y_train_res)
resultGNB = gnb.score(X_test, y_test)
print("Score Naive-Bayes: %.3f"%(resultGNB))
```

Score Naive-Bayes: 0.861

```
In [92]: matrixConfusionGNB = confusion_matrix(y_test, gnb.predict(X_test))
pd.DataFrame(matrixConfusionGNB, index=['True', 'False'], columns= ['True', 'False'])
```

Out[92]:

	True	False
True	76	14
False	0	11

Precision 0.844

Recall: 0.874

Modelo de aprendizado para classificar uma pessoa como obesa

KNN

```
In [74]: knn = KNeighborsClassifier()
```

```
In [100]: knn.fit(X_train_res, y_train_res)
print("Score KNN: %.3f"%(knn.score(X_test, y_test)))
```

Score KNN: 0.911

```
In [97]: matrixConfusionKNN = confusion_matrix(y_test, knn.predict(X_test))
pd.DataFrame(matrixConfusionKNN, index=['True', 'False'], columns= ['True', 'False'])
```

Out[97]:

	True	False
True	83	7
False	2	9

Precision 0.922

Recall: 0.902

Modelo de aprendizado para classificar uma pessoa como obesa

```
In [101]: rfc1.fit(X_train_res, y_train_res)
          rfc2.fit(X_train_res, y_train_res)
          print("Score Random Forest 1: %.3f\nScore Random Forest 2: %.3f"
```

1) Precision 0.956
Recall: 0.896

2) Precision 1.000
Recall: 0.891

Score Random Forest 1: 0.970
Score Random Forest 2: 1.000

```
In [94]: matrixConfusionRFC1 = confusion_matrix(y_test, rfc1.predict(X_test))
          pd.DataFrame(matrixConfusionRFC1, index=['True', 'False'], columns=['True', 'False'])
```

Out[94]:

	True	False
True	86	4
False	1	10

Modelo de aprendizado para classificar uma pessoa como obesa

```
In [96]: matrixConfusionRFC2 = confusion_
pd.DataFrame(matrixConfusionRFC2
```

Out[96]:

	True	False
True	90	0
False	0	11

```
In [ ]:
```