Analyses des indicateurs de résultats des lycées en occitanie

November 28, 2023

1 PARTIE 1 : Les indicateurs de résultats des lycées

1.1 Le taux de réussite constaté au baccalauréat

- Le taux de réussite au baccalauréat est la proportion, parmi les élèves présents à l'examen, de ceux qui ont obtenu le diplôme
- C'est l'indicateur traditionnel, le plus connu.
- Il rapporte le nombre d'élèves reçus au baccalauréat au nombre d'élèves présents à l'examen.

1.2 Le taux de réussite attendu pour l'établissement

- La Direction de l'évaluation, de la prospective et de la performance (DEPP) a mis au point un modèle statistique de calcul des taux attendus de chaque lycée. Il permet de simuler, pour chaque élève, sa probabilité d'obtenir le baccalauréat, en fonction de ses caractéristiques (âge, niveau scolaire à l'entrée en seconde 5 , origine sociale et sexe) et des caractéristiques du lycée dans lequel il évolue. Le calcul de cette probabilité est réalisé en considérant que l'élève est scolarisé dans un établissement ne contribuant ni plus ni moins que la moyenne des établissements à la réussite scolaire de ses élèves.
- Une fois obtenues ces probabilités pour chaque élève, il suffit ensuite d'en calculer les moyennes. Pour le taux de réussite, ces moyennes sont calculées au niveau de la série ou du domaine de spécialité.Ces moyennes de probabilités sont ce qu'on appelle les taux attendus.
- Le taux de réussite attendu pour l'établissement est ensuite obtenu en faisant la moyenne des taux de réussite attendus par série, pondérés par l'effectif d'élèves présents au baccalauréat.
- Le taux de réussite attendu pour chaque établissement est déjà calculé dans les csv disponibles sur le site data occitanie du gouvernement et que nous allons importer dans ce présent notebook pour réaliser nos analyses.

1.3 La valeur ajoutée

La question est de savoir comment évaluer l'action propre du lycée, ce qu'il a « ajouté » au niveau initial des élèves qu'il a reçus. En d'autres termes, si un lycée présente une valeur élevée pour un indicateur, est-ce dû au fait qu'il a reçu des élèves ayant de meilleures chances de succès – bons élèves dotés de bonnes méthodes de travail ayant pu obtenir le baccalauréat sans effort particulier de la part du lycée – ou bien est-ce dû au fait qu'il a su, tout au long d'une scolarité, développer chez des élèves peut-être moins bien dotés au départ, les connaissances et les capacités qui ont permis leur succès ? Il faut donc s'efforcer d'éliminer l'incidence des facteurs de réussite scolaire extérieurs au lycée pour essayer de conserver ce qui est dû à sonaction propre. Une partie des facteurs de réussite est propre à l'élève. Les facteurs individuels extérieurs que sont l'âge, l'origine

sociale, le sexe et le niveau scolaire à l'entrée au lycée de chaque élève (apprécié par la moyenne des notes aux épreuves écrites du diplôme national du brevet [DNB]) ont été retenus car ils donnent une première approximation des chances (au sens statistique du terme) d'accès et de réussite au baccalauréat d'un élève

- la valeur ajoutée de l'établissement est la différence entre le taux constaté de l'établissement et le taux attendu.

  Valeur ajoutée = taux constaté – taux attendu

- **la valeur ajoutée peut etre négative**

2 PARTIE 2 : Analyse des résultats des lycées professionels en Occitanie

- En premier temps, nous allons procéder à une analyse des résultats et de la valeur ajoutée des lycées professionels en fonction du labels numérique et nous allons expliquer en quoi consiste la valeur ajoutée.
- En deuxième temps, nous allons procéder à une ANOVA à un facteur pour répondre à la question : y a t'il un effet significatif du label numérique sur le taux de réussite des élèves des lycées professionels au baccalauréat

- En troisième temps, nous allons procéder à une ANOVA à un facteur pour répondre à la question : y a t'il un effet significatif du label numérique sur la valeur ajoutée des lycées proffessionels au baccalauréat

2.1 PARTIE 2.1 : Importer, transformer et nettoyer les données nécessaires à l'analyse

2.1.1 Dataframe des labels numériques des lycées

Import data from api

[1]:
```
#importing necessary packages
import pandas as pd
import requests
import json

# get data
#creating the url
url = 'https://data.occitanie.education.gouv.fr/api/records\
/1.0/search/?dataset=fr-en-occitanie-label-numerique-lycee\
&q=&rows=10000&facet=annee&facet=rne&facet=departement&facet=label'
#making the request
response = requests.get(url)
#converting the response to json
data = response.json()
#appending the data to the list

#creating an empty list to store the records
```

```
records = []

#looping through the records
for record in data['records']:
    #storing the records in the list
    records.append(record['fields'])

#creating a dataframe from the json
df = pd.DataFrame(records)
```

Transforming dataframe

[2]:
```
#transform the label to numeric
df["label"] = pd.to_numeric(df["label"])
df["annee"] = pd.to_numeric(df["annee"])

#sort annee by ascending order
df.sort_values(by="annee",inplace=True)

#create two new columns for longitude and latitude
df['latitude'] = df['position'].apply(lambda x: float(x[0]))
df['longitude'] = df['position'].apply(lambda x: float(x[1]))
```

2.1.2 Dataframe des résultats des lycee professionnels

Import data from api

[3]:
```
# get data
#creating the url
url = 'https://data.occitanie.education.gouv.fr\
/api/records/1.0/search\
/?dataset=fr-en-indicateurs-de-resultat-des-lycees-denseignement-professionnels\
&q=&rows=10000&facet=etablissement&facet=code_etablissement&facet=annee\
&facet=ville&facet=academie&facet=departement\
&facet=secteur_public_1_prive_2&facet=libelle_region_2016'
#making the request
response = requests.get(url)
#converting the response to json
data = response.json()

#creating an empty list to store the records
records = []

#looping through the records
for record in data['records']:
    #storing the records in the list
    records.append(record['fields'])
```

```
#creating a dataframe from the json
df2 = pd.DataFrame(records)
```

Nettoyage

[4]:
```
# nettoyage

# Le lycee "LYCEE PROFESSIONNEL CASTELNOUVEL" a un taux de réussite brut 0
# Il a aussi un effectif_presents_total_secteurs 1, ce qui errone la␣
 ↪visualisation (un point très écarté)
# ainsi qu'un taux_mentions_attendu_tous_secteurs Nan
# j'ai donc décidé d'enlever ce lycee
df2.drop(df2[df2['code_etablissement'] == "0312063Z"].index, inplace = True)
```

Transform

[5]:
```
# transformation

#transform the annee to numeric
df2["annee"] = pd.to_numeric(df2["annee"])
#sort annee by ascending order
df2.sort_values(by="annee",inplace=True)


# transform le taux en numerique
df2["taux_brut_de_reussite_total_secteurs"] = pd.
 ↪to_numeric(df2["taux_brut_de_reussite_total_secteurs"])

# keep the df of lycee professionels in a variable to later use
df_lycee_pro = df2.copy()
```

2.1.3 MERGE :

- Datarame labels numériques lycee
- Dataframe résultats scolaires des lycee professionnels

[6]:
```
# merge 2 dataframes on the uri (id de l'etablissement scolaire)
df_result = pd.merge(df, df2, left_on='rne', right_on='code_etablissement',␣
 ↪how="inner")
```

[7]:
```
# merge 2 dataframes but keep all the lycee that does not have label
df_result_all = pd.merge(df,df2, left_on='rne', right_on='code_etablissement',␣
 ↪how="right")
```

Nettoyage df result (resultat des lycées professionels labelisés)

[8]:
```
#nettoyage (supression) des colonnes ou toutes les valeures sont nulles
df_result.dropna(axis=1,how='all',inplace=True)
```

```
#nettoyage (supression) des lignes ou toutes les valeures sont nulles
df_result.dropna(axis=0,how='all',inplace=True)

# L'index 1393 a une valeure ajoutée de réusiite totale = "ND"
# par contre ce lycee a un taux_brut_de_reussite_total_secteurs = 84
# j'ai donc décidé de rempace le "ND" par 0 (neutre) pour ce lycee et pour tous␣
 ↪les lycee qui ont
# une valeure ajoutée de réussite totale = "ND" (non définir)
import re
reg_ex1 = r"ND"
reg_ex2 = r"\."
aNegliger = re.compile(f"{reg_ex1}|{reg_ex2}",re.I)
subst = "0"
def apply_reg(str_):
    '''Fonction pour remplacer les "ND" par 0 tout court dans une colonne
        - paramètres :
            - str_ : la chaine de caractère dans laquelle la fonction va␣
 ↪chercher la pattern à nettoyé
        - Return la chaine de caractère nettoyé '''
    if pd.notna(str_):
        return re.sub(aNegliger, subst, str(str_))
    else:
        return str_

df_result['va_reu_total'] = df_result['va_reu_total'].apply(apply_reg)
```

Nettoyage df result all (resultat de tous les lycées professionels)

[9]:
```
#nettoyage (supression) des colonnes ou toutes les valeures sont nulles
df_result_all.dropna(axis=1,how='all',inplace=True)

#nettoyage (supression) des lignes ou toutes les valeures sont nulles
df_result_all.dropna(axis=0,how='all',inplace=True)

# L'index 1393 a une valeure ajoutée de réusiite totale = "ND"
# par contre ce lycee a un taux_brut_de_reussite_total_secteurs = 84
# j'ai donc décidé de rempace le "ND" par 0 (neutre) pour ce lycee et pour tous␣
 ↪les lycee qui ont
# une valeure ajoutée de réussite totale = "ND" (non définir)
df_result_all['va_reu_total'] = df_result_all['va_reu_total'].apply(apply_reg)
```

Transform df qui contient les resultat des lycées labelisés seulement (df_result)

[10]:
```
# transformer les dates en numériques pour la prochaine opération
df_result["annee_x"] = pd.to_numeric(df_result["annee_x"])
df_result["annee_y"] = pd.to_numeric(df_result["annee_y"])
```

```
# ajouter une colonne qui contient boolean (resultat après obtention label ou␣
 ↪pas)
# True si oui, false sinon
# les resultats des lycee après l'obtention de leurs labels
# (annee_x pour obtention label, annee_y pour passage examens de baccalauréat)
df_result["resultat_apres_label"] = (df_result["annee_y"] >=␣
 ↪df_result["annee_x"])

# transformer la valeur ajoutée de réussites totale en numérique
df_result["va_reu_total"] = pd.to_numeric(df_result["va_reu_total"])

# transorm taux_reussite_attendu_france_total_secteurs to numeric
df_result["taux_reussite_attendu_france_total_secteurs"] = pd.
 ↪to_numeric(df_result["taux_reussite_attendu_france_total_secteurs"])
```

Transform df qui contient les résultats de tous les lycées inclus sont qui sont labelisés (df_result_all)

[11]:
```
# transformer les dates en numériques pour la prochaine opération
# transformer les dates en numériques pour la prochaine opération
df_result_all["annee_y"] = pd.to_numeric(df_result_all["annee_y"])

# Creer une colonne pour dire si le lycee est labelisé : True/False
df_result_all["label_true"] = (pd.notna(df_result_all["rne"]))

# ajouter une colonne qui contient boolean (resultat après obtention label ou␣
 ↪pas)
# True si oui, false sinon
# les resultats des lycee après l'obtention de leurs labels
# (annee_x pour obtention label, annee_y pour passage examens de baccalauréat)
df_result_all["resultat_apres_label"] = (df_result_all["annee_y"] >=␣
 ↪df_result_all["annee_x"]) & (df_result_all["label_true"] == True)

# remplacer tous les labels à 0 where examen passé avant 2017
df_result_all.loc[df_result_all["resultat_apres_label"] == False, "label"] = 0

# transformer la valeur ajoutée de réussites totale en numérique
df_result_all["va_reu_total"] = pd.to_numeric(df_result_all["va_reu_total"])

# transorm taux_reussite_attendu_france_total_secteurs to numeric
df_result_all["taux_reussite_attendu_france_total_secteurs"] = pd.
 ↪to_numeric(df_result_all["taux_reussite_attendu_france_total_secteurs"])
```

2.1.4 Transform for visualisations

[12]:
```
# re transform annee to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result["annee_x"] = df_result["annee_x"].apply(str)
df_result["annee_y"] = df_result["annee_y"].apply(str)

df_result_all["annee_x"] = df_result_all["annee_x"].apply(str)
df_result_all["annee_y"] = df_result_all["annee_y"].apply(str)
```

2.2 PARTIE 2.2 : Analyse générale des lycées professionels en Occitanie

[13]:
```
# Import necessary libraries
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import folium # for maps
import plotly.graph_objects as go
import kaleido
from IPython.display import Image
import io
from PIL import Image as ImagePIL
```

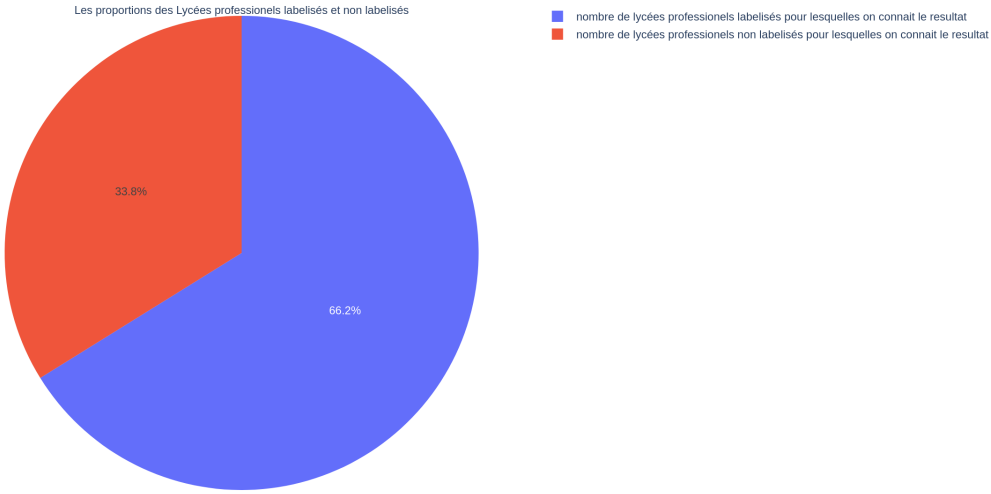
2.2.1 Les proportions des Lycées professionels labelisés et non labelisés

[14]:
```
nbre_lycee_professionnel = df_result_all['code_etablissement'].nunique()
nbre_lycee_professionel_labelise = df_result_all['rne'].nunique()
nbre_lycee_professionnel_non_labelise = nbre_lycee_professionnel -␣
 ↪nbre_lycee_professionel_labelise
values =␣
 ↪[nbre_lycee_professionnel_non_labelise,nbre_lycee_professionel_labelise]
labels = ["nombre de lycées professionels non labelisés pour lesquelles on␣
 ↪connait le resultat", "nombre de lycées professionels labelisés pour␣
 ↪lesquelles on connait le resultat"]

fig = go.Figure(data=[go.Pie(labels=labels, values=values, title = "Les␣
 ↪proportions des Lycées professionels labelisés et non labelisés")])
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[14]:



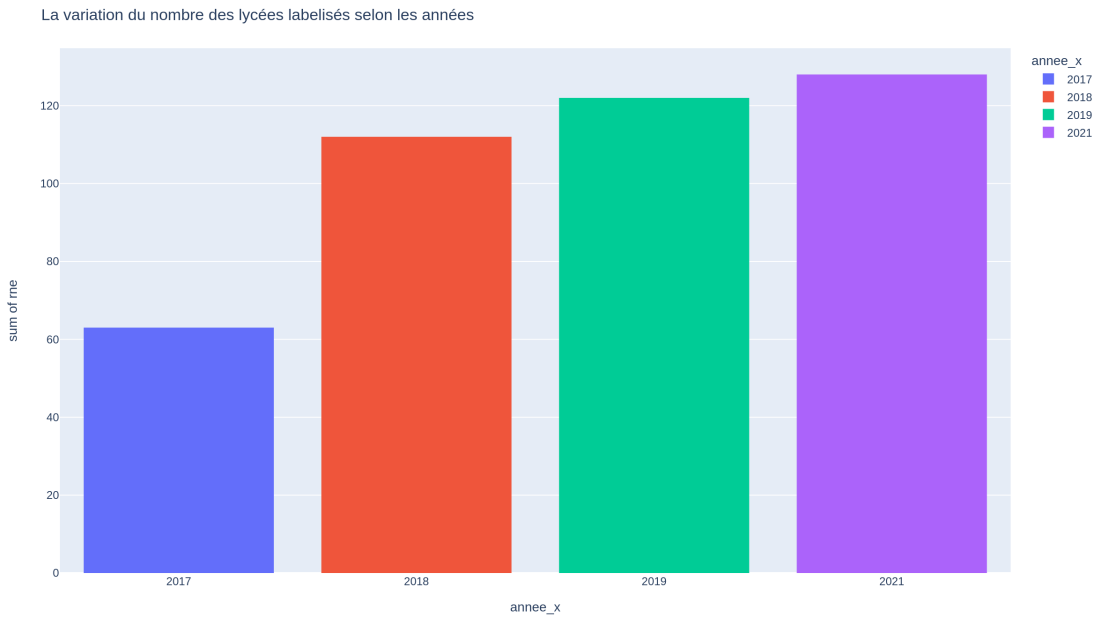
2.2.2 La variation du nombre des lycées labelisés selon les années

[15]:
```
# La variation du nombre des lycées labelisés selon les années
df_proportion_lycee_labelise_annee = df_result.groupby("annee_x")["rne"].
 ↪nunique()
df_proportion_lycee_labelise_annee = df_proportion_lycee_labelise_annee.
 ↪reset_index()

#plotting a histogram
fig = px.histogram(df_proportion_lycee_labelise_annee, x="annee_x", y="rne",
                   color="annee_x",
                      labels={"code_etablissement" : "nombre de lycee␣
 ↪labelisés"})
fig.update_layout(title_text="La variation du nombre des lycées labelisés selon␣
 ↪les années")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[15]:



- On constate que le nombre de lycées professionels labelisés a augmente de 103,17 % entre 2017 et 2021

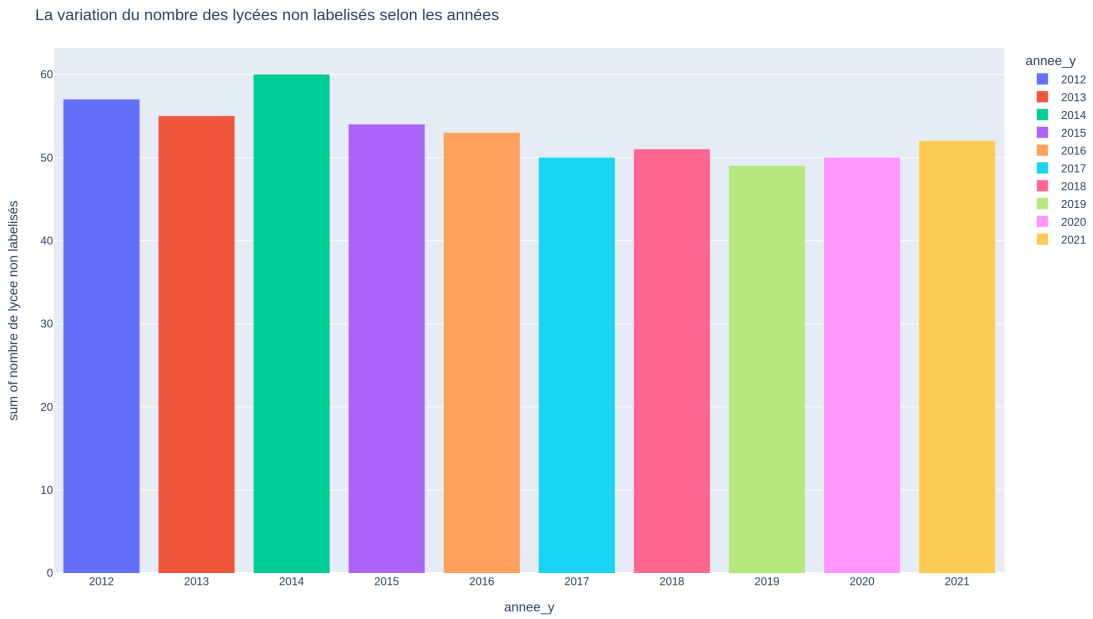
2.2.3 La variation du nombre des lycées non labelisés selon les années

[16]:
```
# La variation du nombre des lycées non labelisés selon les années
df_proportion_lycee_non_labelise_annee = df_result_all.query("label_true ==␣
 ↪False").groupby("annee_y")["code_etablissement"].nunique()
df_proportion_lycee_non_labelise_annee = df_proportion_lycee_non_labelise_annee.
 ↪reset_index()

#plotting a histogram
fig = px.histogram(df_proportion_lycee_non_labelise_annee, x="annee_y",␣
 ↪y="code_etablissement",
                   color="annee_y",
                      labels={"code_etablissement" : "nombre de lycee non␣
 ↪labelisés"})
fig.update_layout(title_text="La variation du nombre des lycées non labelisés␣
 ↪selon les années")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[16]:



- On constate que le nombre de lycées professionels non labelisés a diminué de 8.7% entre 2012 et 2021

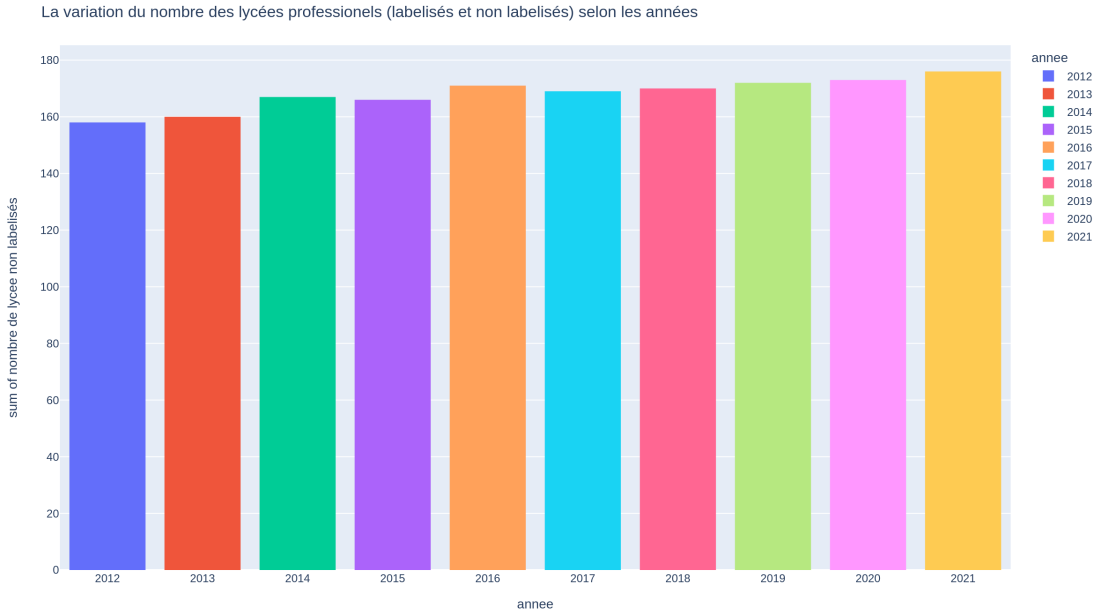
2.2.4 La variation du nombre des lycées professionels (labelisés et non labelisés) selon les années

[17]:
```
# La variation du nombre des lycées professionels (labelisé ou non) selon les années
df2["annee"] = df2["annee"].apply(str)
df_proportion_lycee_tout_annee = df2.groupby("annee")["code_etablissement"].nunique()
df_proportion_lycee_tout_annee = df_proportion_lycee_tout_annee.reset_index()
df_proportion_lycee_tout_annee

#plotting a histogram
fig = px.histogram(df_proportion_lycee_tout_annee, x="annee",y="code_etablissement",
                   color="annee",
                      labels={"code_etablissement" : "nombre de lycee non labelisés"})
fig.update_layout(title_text="La variation du nombre des lycées professionels (labelisés et non labelisés) selon les années")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[17]:



- On constate que le nombre des lycées professionels (labelisés et non labelisé) a augmenté de 11.3% entre 2012 et 2021

2.2.5 Conclusion

Au fil des années, les lycées non labelisés diminuent et obtiennent des labels et deviennent alors labelisés. De ce fait le nombre des lycées labelisés augmentent. Cependant, l'augmentation du nombre des lycées professionels labelisés (103%) entre 2017 et 2021, n'est pas seulement expliqué par la diminution du nombre des lycées non labelisés (8,7%) entre 2012 et 2021. Cette augmentation est aussi du au fait que des nouveaux lycées labelisés sont introduits chaque année

2.3 PARTIE 2.3 : Analyse du taux de réussite des élèves des lycées professionels au baccalauréat

- Il est intéréssant de voir quelle sont les moyennes de taux de réussite des écoles avant et après l'obtention de leurs labels numérique !

2.3.1 La variance du taux de réussites des élèves au baccalauréat dans les lycées professionels avant et après obtention labels

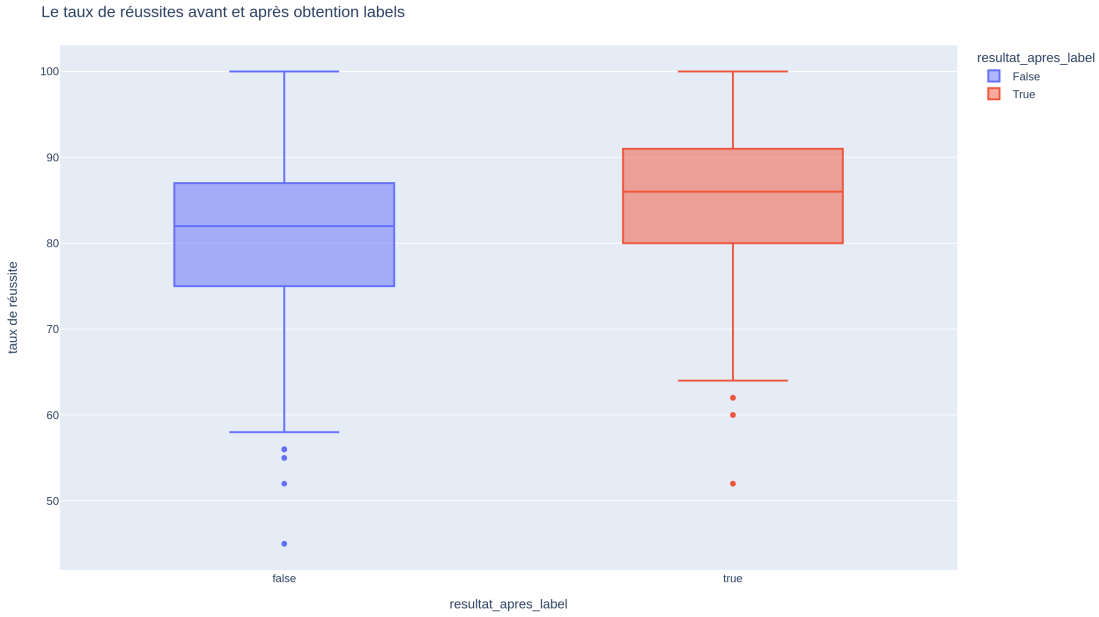
- **True = après obtention label**
- **False = avant obtention label**

[18]:
```
#plotting a boxplot
fig = px.box(df_result, x="resultat_apres_label",␣
 ↪y="taux_brut_de_reussite_total_secteurs",
                  color="resultat_apres_label",
```

```
                        labels={"taux_brut_de_reussite_total_secteurs" : "taux de␣
 ↪réussite"})
fig.update_layout(title_text="Le taux de réussites avant et après obtention␣
 ↪labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[18]:



Comparaison des taux de réussites constatés avant et après obtention des labels numériques

- Le graphique ci-dessus montre une amélioration des taux de réussite des lycées après obtention de leurs labels numérique !
- On remarque que le taux de réussite median était **82%** avant obtention du label numérique, et la médiane de ce taux a augmenté à **86%** après obtention du label numérique. Il existe donc une variance inter-colonnes (due au facteur du label numérique) intéressante. Ce qui nous ramène par la suite à faire une ANOVA. Nous allons expliquer ce point dans la suite de cette analyse.
- IL est donc intéréssant d'aller chercher pourquoi ? est ce que l'école a apporté plus de valeur pour les élèves grace au numérique ?
- C'est intéressant comme résultat, nous allons aussi voir par la suite la différence de la moyenne du taux de réussites des élèves au baccalauréat dans les lycées professionels qui ne sont pas labelisés du tout et ceux qui sont labelisés

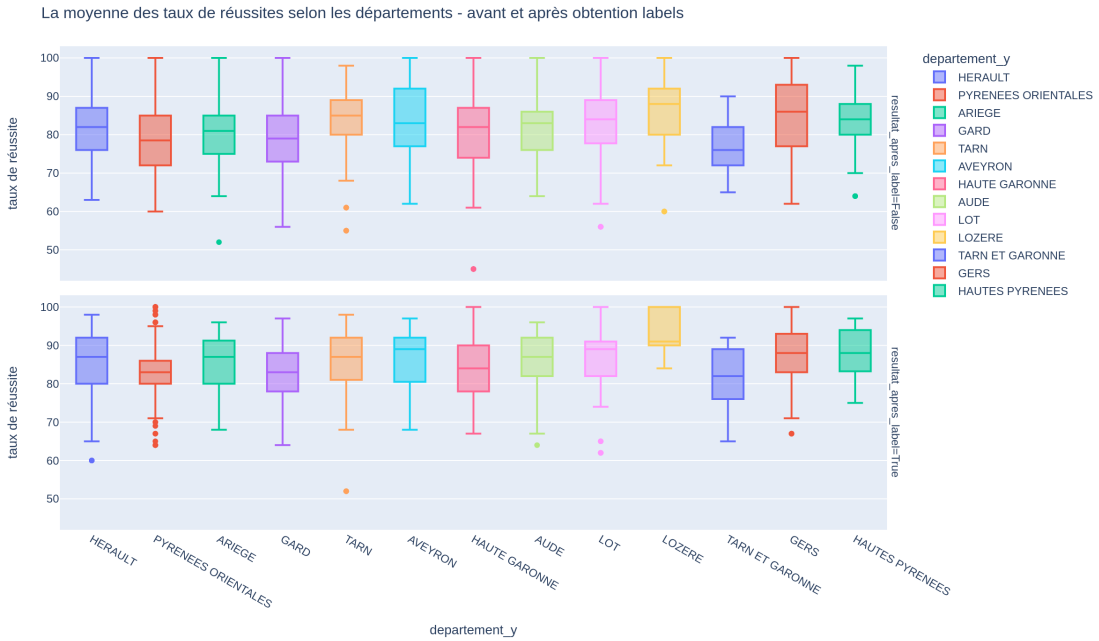
2.3.2 Boxplot: Les taux de réussites des élèves au baccalauréat dans les lycées professionels avant et après obtention labels par DÉPARTEMENT

- La partie supérieure montre le taux de réussite avant obtention du label
- La partie inférieure montre le taux de réussite après obtention du label

[19]:
```
#plotting a histogram
fig = px.box(df_result, x="departement_y",␣
 ↪y="taux_brut_de_reussite_total_secteurs", color="departement_y",
                       facet_row="resultat_apres_label",
                      labels={"taux_brut_de_reussite_total_secteurs" : "taux de␣
 ↪réussite"})
fig.update_layout(title_text="La moyenne des taux de réussites selon les␣
 ↪départements - avant et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[19]:



INTERPRÉTATION

- ON constate une amélioration du taux de réussite des lycées dans chaque département, notamment le département du **AVEYRON** qui voit la médiane de son taux de réussite augmenté de **83%** avant obtention du label à **89%** après obtention du label

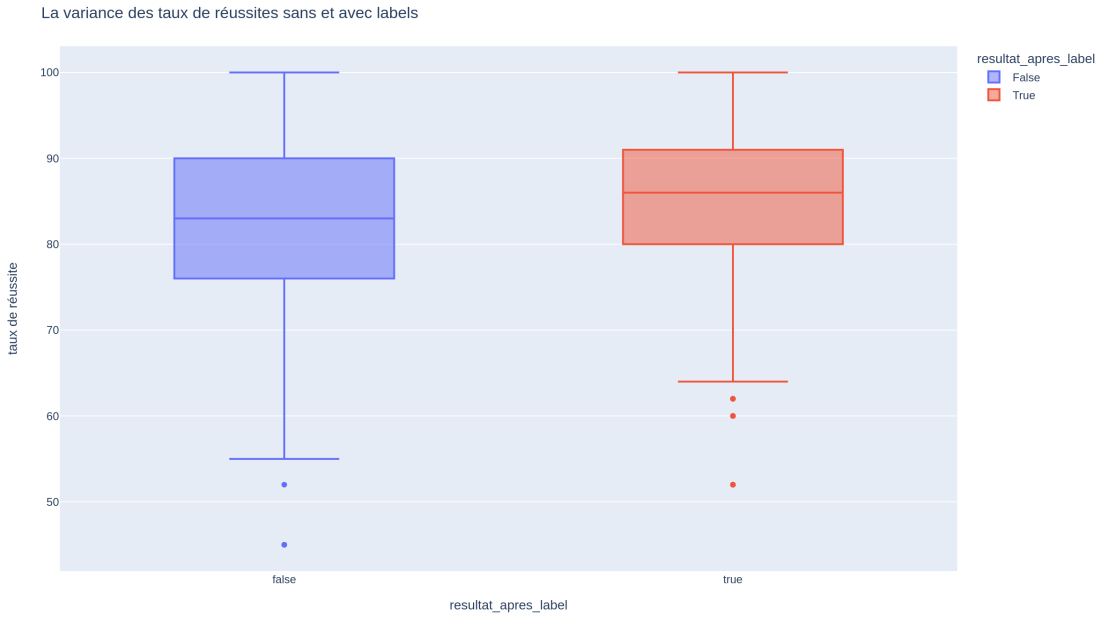
On va voir maintenant la différence du taux de réussites des élèves au baccalauréat dans les lycées professionels qui ne sont pas labelisés du tout et ceux qui sont labelisés

2.3.3 BOXPLOT : La variance des taux de réussites au baccalauréat sans et avec labels

[20]:
```
#plotting a bocplot
fig = px.box(df_result_all, x="resultat_apres_label",␣
 ↪y="taux_brut_de_reussite_total_secteurs",
             color="resultat_apres_label",
            labels={"taux_brut_de_reussite_total_secteurs" : "taux de␣
 ↪réussite"})
fig.update_layout(title_text="La variance des taux de réussites sans et avec␣
 ↪labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[20]:



Comparaison des taux de réussites constatés SANS et AVEC label numérique

- Le graphique ci-dessus montre que les lycées AVEC labels ont un meilleur taux de réussite que ceux SANS label!
- On remarque que le taux de réussite median est à **83%** SANS label numérique, et la médiane de ce taux a augmenté à **86%** pour les lycées ayant un niveau de label numérique.

2.3.4 Variance du taux de réussite selon le niveau de label numérique

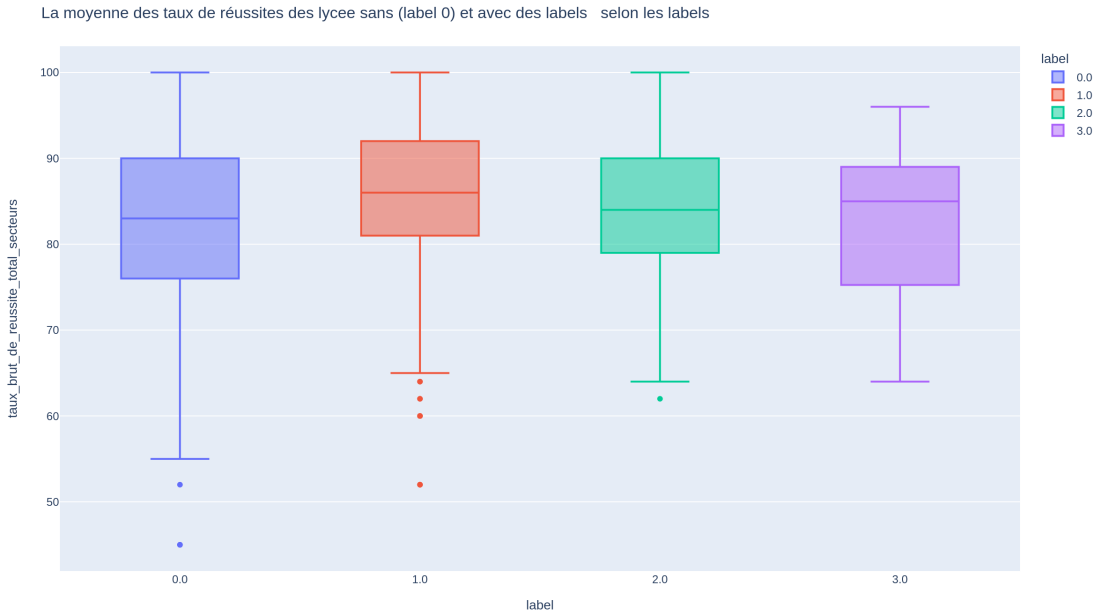
A NOTER LABEL 0 = PAS DE LABEL

[21]:
```
# transform label to numeric
df_result_all["label"] = pd.to_numeric(df_result_all["label"])
#sort label by ascending order
df_result_all.sort_values(by="label",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["label"] = df_result_all["label"].apply(str)

#plotting a histogram
fig = px.box(df_result_all, x="label",␣
 ↪y="taux_brut_de_reussite_total_secteurs", color="label")
fig.update_layout(title_text="La moyenne des taux de réussites des lycee sans␣
 ↪(label 0) et avec des labels   selon les labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[21]:



INTERPRÉTATION :

- On remarque que le niveau de label 1 a la médiane du taux de réussite la plus élevé par parmis les niveaux de labels
- Le niveau de **label 0 (c'est à dire aucun label)** à la médiane du taux de réussite la plus basse

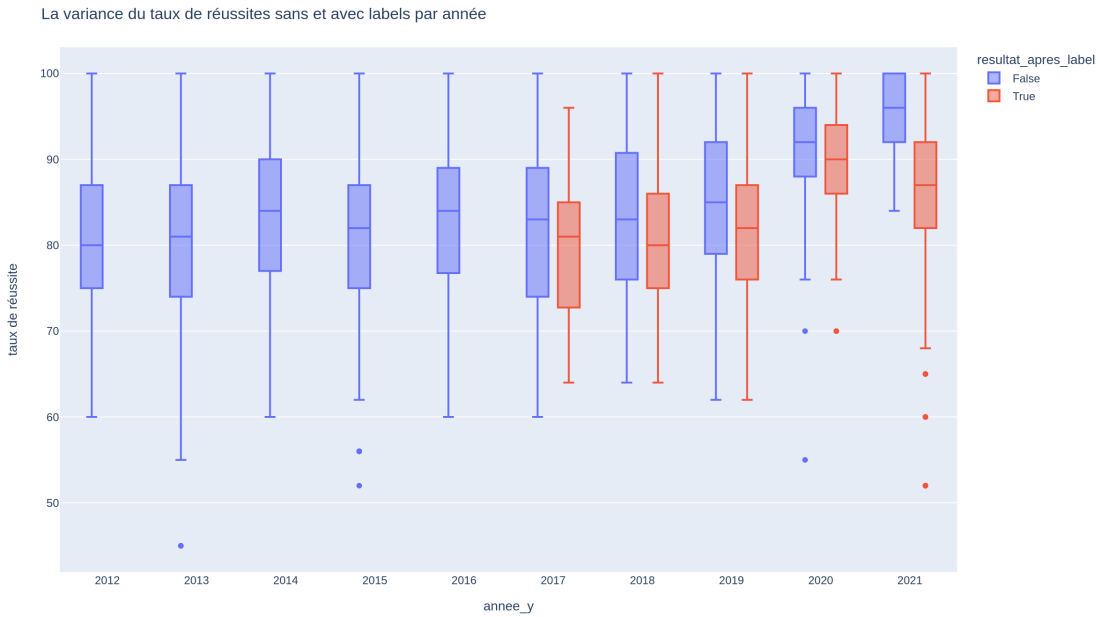
2.3.5 BOXPLOT : La variance du taux de réussites sans et avec labels par année

[22]:
```
# transform label to numeric
df_result_all["annee_y"] = pd.to_numeric(df_result_all["annee_y"])
#sort label by ascending order
df_result_all.sort_values(by="annee_y",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["annee_y"] = df_result_all["annee_y"].apply(str)

#plotting a bocplot
fig = px.box(df_result_all, x="annee_y",␣
 ↪y="taux_brut_de_reussite_total_secteurs",
             color="resultat_apres_label",
            labels={"taux_brut_de_reussite_total_secteurs" : "taux de␣
 ↪réussite"})
fig.update_layout(title_text="La variance du taux de réussites sans et avec␣
 ↪labels par année")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[22]:



2.3.6 Interprétation : On pose l'hypothèse H0

On constate que les lycées professionels labelisés ont un meilleur résultat que ceux non labelisés (toutes années confondues)

- On se demande pourquoi ?
- On sait que parmis les lycées professionels, 33% ne sont pas labelisés, et 67% sont labelisés.
- Alors nous souhaitons répondre à la question suivante : y'a t'il un effet de la labélisation sur le taux de réussite des lycées professionels ?
- Pour cela on pose les hypothèses suivantes :
  - **H0 : La labélisation d'un lycee professionel n'a pas d'effet sur son taux de réussite**
  - **H1 : La labélisation d'un lycee professionel a un effet sur son taux de réussite**

2.3.7 Covid

Nous constatons un pic de taux de réussite des lycées professionels en 2020, puis une diminution de ce taux en 2021

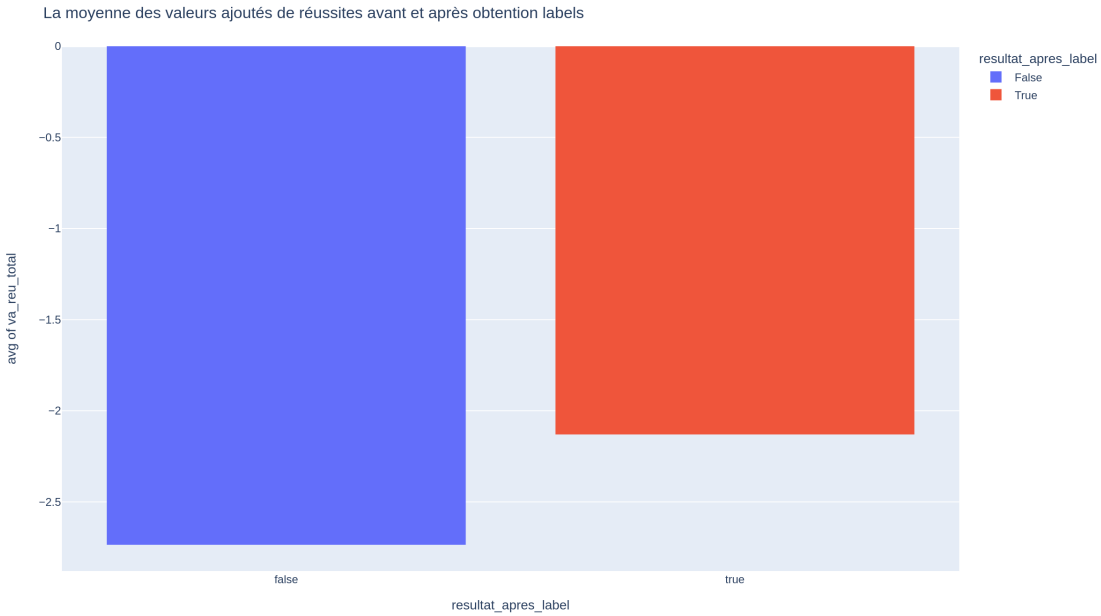
2.4 PARTIE 2.4 : Analyse de la valeur ajoutée de réussite

2.4.1 Histogramme : La moyenne des valeurs ajoutés de réussites avant et après obtention labels

[23]:
```
#plotting a histogram
fig = px.histogram(df_result, x="resultat_apres_label", y="va_reu_total",␣
 ↪nbins=10, histfunc="avg", color="resultat_apres_label")
fig.update_layout(title_text="La moyenne des valeurs ajoutés de réussites avant␣
 ↪et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[23]:



Intérprétation

- On constate une amélioration de la moyenne de la valeur ajoutée des lycées professionels après obtention de leurs labels numériques
- On remarque que la moyenne de la valeur ajoutée a aumenté de **-2,735** avant obtention label à **-2,129** après obtention label, ce qui est non négligeable à l'échelle de la valeur ajoutée

2.4.2 Histogramme : La moyenne des valeurs ajoutés de réussite selon les départements avant et après obtention labels

- La partie supérieure montre le taux de réussite avant obtention du label
- La partie inférieure montre le taux de réussite après obtention du label

[24]:
```
#plotting a histogram
fig = px.histogram(df_result, x="departement_y", y="va_reu_total", nbins=10,␣
 ↪histfunc="avg", color="departement_y", facet_row="resultat_apres_label")
fig.update_layout(title_text="La moyenne des valeurs ajoutés de réussites selon␣
 ↪les départements avant et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[24]:

Comparaison des valeurs ajoutées des écoles avant et après obtention des labels numériques

- le graphique ci-dessous montre une véritables amélioration des valeurs ajoutée des écoles après obtention de labels. Surtout pour le département de LOZERE pour lequel la valeur ajoutée est passée du **-2,5** à **+2,04** !

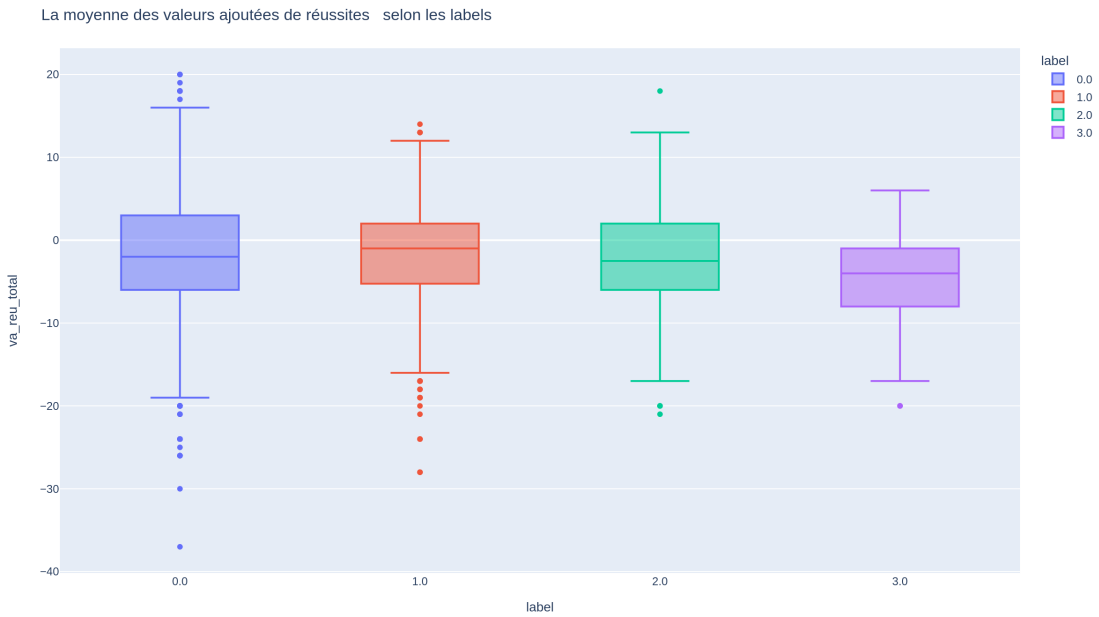
2.4.3 Boxplot : Variance de la valeure ajoutée selon le label (0 = pas de label)

[25]:
```
# transform label to numeric
df_result_all["label"] = pd.to_numeric(df_result_all["label"])
#sort label by ascending order
df_result_all.sort_values(by="label",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
↪(discrete value)
df_result_all["label"] = df_result_all["label"].apply(str)

#plotting a histogram
fig = px.box(df_result_all, x="label", y="va_reu_total", color="label")
fig.update_layout(title_text="La moyenne des valeurs ajoutées de réussites ␣
↪selon les labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[25]:



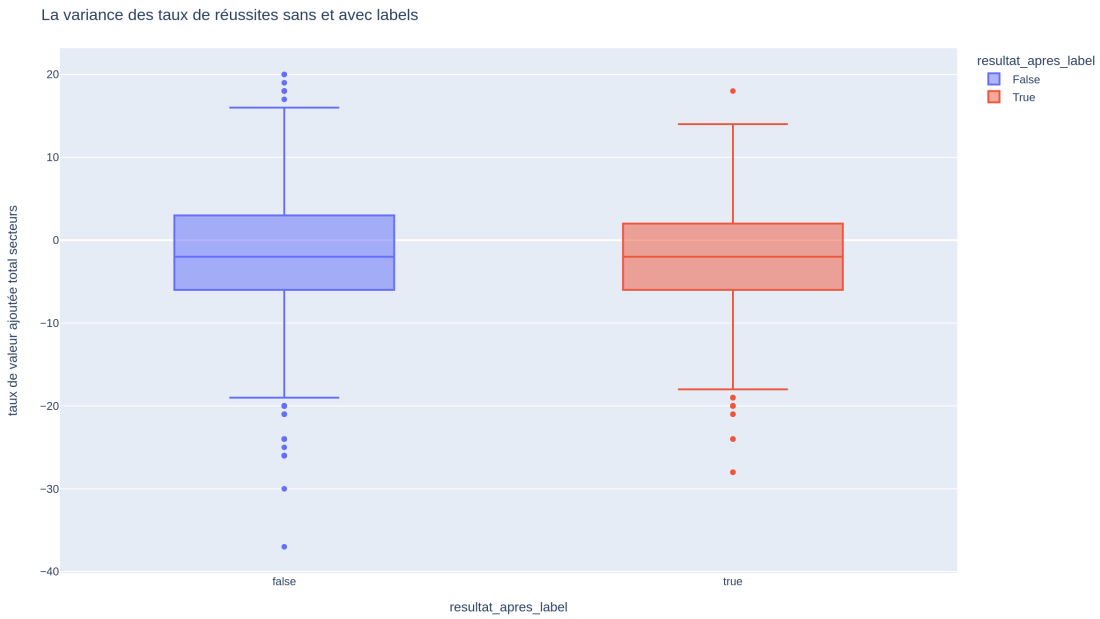
2.4.4 Boxplot : la variance des valeurs ajoutés de réussites des lycées professionnels sans et avec labels

- True = avec label
- False = sans label

[26]:
```
#plotting a bocplot
fig = px.box(df_result_all, x="resultat_apres_label", y="va_reu_total",
             color="resultat_apres_label",
            labels={"va_reu_total" : "taux de valeur ajoutée total secteurs"})
fig.update_layout(title_text="La variance des taux de réussites sans et avec␣
 ↪labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[26]:



On remarque que la médiane est de la valeur ajoutée est à **-2** que ça soit sans ou avec label

2.5 PARTIE 2.5 : Analyse du taux de réussite attendu des lycées professionels

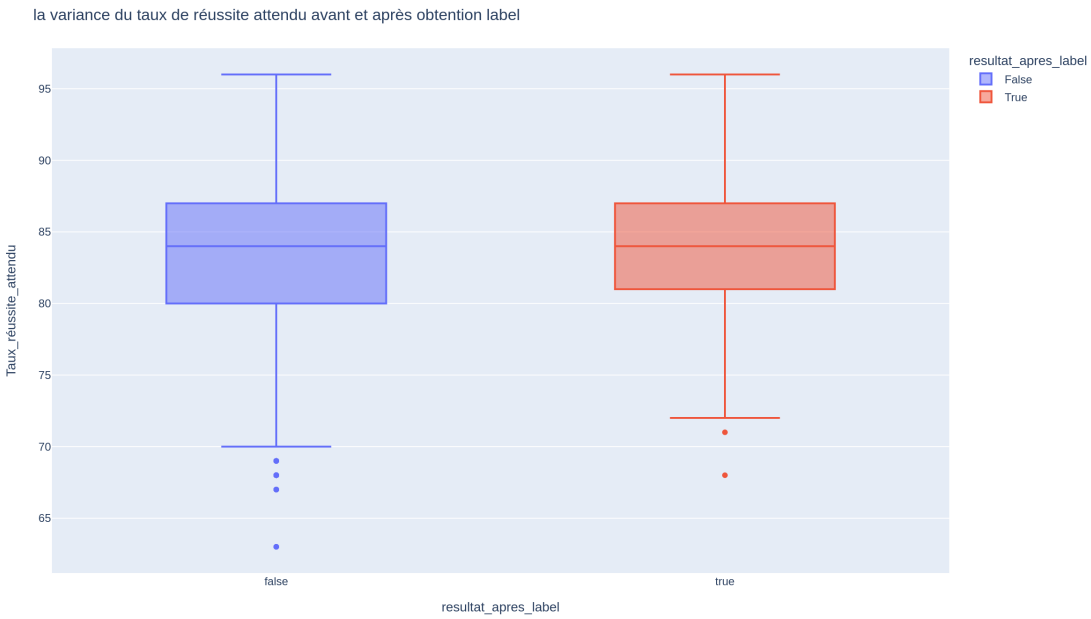
2.5.1 La variance du taux de réussite attendu avant et après obtention label

- True = après label
- False = avant label

[27]:
```
#plotting a histogram
fig = px.box(df_result, x="resultat_apres_label",␣
 ↪y="taux_reussite_attendu_france_total_secteurs",
             color="resultat_apres_label",
                      labels={"taux_reussite_attendu_france_total_secteurs" :␣
 ↪"Taux_réussite_attendu"})
fig.update_layout(title_text="la variance du taux de réussite attendu avant et␣
 ↪après obtention label")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[27]:



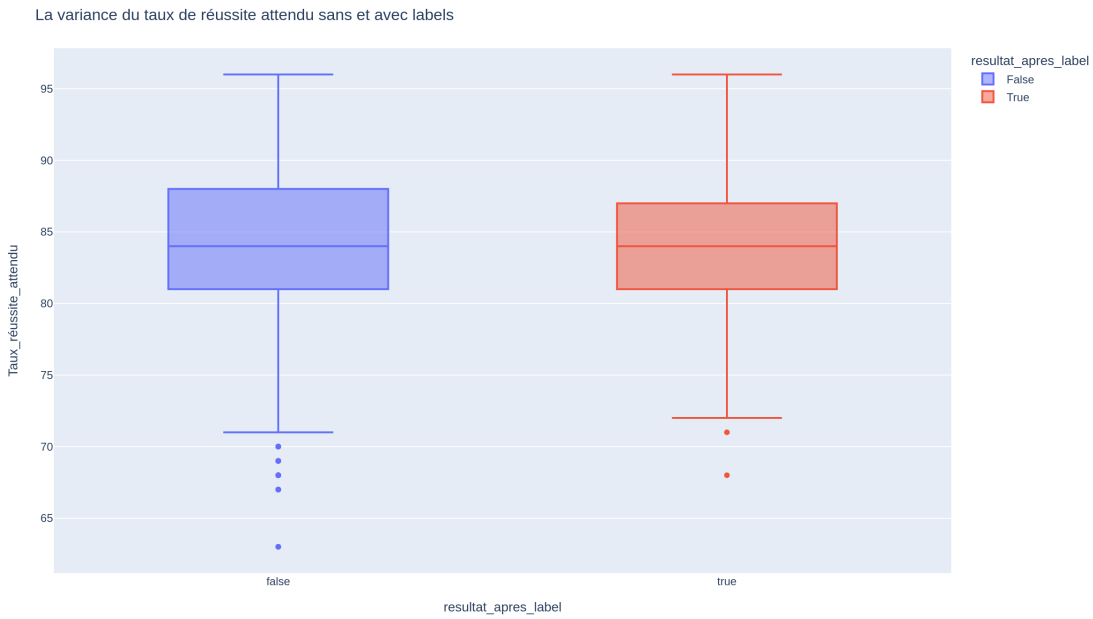
2.5.2 La variance du taux de réussite attendu sans et avec labels

- True = avec label
- False = sans label

[28]:
```
#plotting a histogram
fig = px.box(df_result_all, x="resultat_apres_label",␣
 ↪y="taux_reussite_attendu_france_total_secteurs",␣
 ↪color="resultat_apres_label",
                     labels={"taux_reussite_attendu_france_total_secteurs" :␣
 ↪"Taux_réussite_attendu"})
fig.update_layout(title_text="La variance du taux de réussite attendu sans et␣
 ↪avec labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[28]:



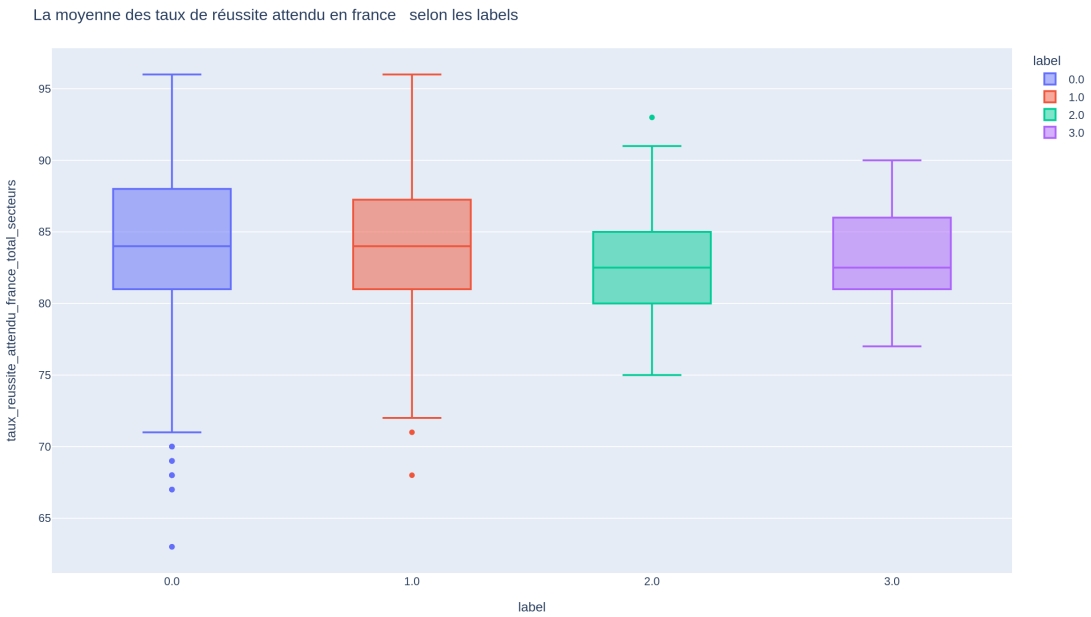
2.5.3 La variance du taux de réussite attendu selon le niveau du label

[29]:
```
# transform label to numeric
df_result_all["label"] = pd.to_numeric(df_result_all["label"])
#sort label by ascending order
df_result_all.sort_values(by="label",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["label"] = df_result_all["label"].apply(str)

#plotting a histogram
fig = px.box(df_result_all, x="label",␣
 ↪y="taux_reussite_attendu_france_total_secteurs", color="label")
fig.update_layout(title_text="La moyenne des taux de réussite attendu en france␣
 ↪  selon les labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[29]:



2.6 PARTIE 2.6 : ANOVA

2.6.1 Effet de la labélisation sur le taux de réussite des lycées professionels

[30]:
```
import statsmodels.api as sa
import statsmodels.formula.api as sfa
import scikit_posthocs as sp
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import scipy.stats as stats
```

Résultat de l'anova

[31]:
```
lm = sfa.ols('taux_brut_de_reussite_total_secteurs ~ C(label)',␣
 ↪data=df_result_all).fit()
anova = sa.stats.anova_lm(lm)
anova
```

[31]:

	df	sum_sq	mean_sq	F	PR(>F)
C(label)	3.0	4843.154856	1614.384952	19.516795	1.452482e-12
Residual	4284.0	354362.742299	82.717727	NaN	NaN

Analyse de l'anova P_value < alpha (0,05), donc on rejette H0 et on conclut d'une manière significative un effet du label numérique sur le taux de réussite des lycées professionels

Test de Tukey

- permet de préciser quelles modalités de la variable qualitative label a provoqué ce rejet

[32]:
```
# perform Tukey's test
tukey =␣
 ↪pairwise_tukeyhsd(endog=df_result_all['taux_brut_de_reussite_total_secteurs'],
                          groups=df_result_all['label'],
                          alpha=0.05)

#display results
print(tukey)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1 group2 meandiff p-adj   lower   upper  reject
-----------------------------------------------------
   0.0    1.0   2.5613    0.0  1.6883  3.4343   True
   0.0    2.0   1.0238 0.3992 -0.6581  2.7057  False
   0.0    3.0  -0.7712 0.9246 -3.9509  2.4084  False
   1.0    2.0  -1.5375 0.1245 -3.3369  0.2619  False
   1.0    3.0  -3.3325 0.0413 -6.5759 -0.0892   True
   2.0    3.0  -1.7951 0.5627 -5.3427  1.7526  False
-----------------------------------------------------
```

Analyse de test de tukey **Le test de tukey est utilisé pour comparer les moyennes de plusieurs groupes. Le test est utilisé pour déterminer s'il existe des différences significatives entre les moyennes de différents groupes. L'output montre les résultats du test pour chaque comparaison à paires de groupes. Les colonnes de l'output incluent:**

- group1 et group2: les groupes étant comparés
- meandiff: la différence des moyennes entre les deux groupes
- p-adj: la valeur p ajustée pour la comparaison
- lower et upper: les limites inférieure et supérieure de l'intervalle de confiance de 95% pour la différence des moyennes
- reject: si oui ou non l'hypothèse nulle (que les moyennes sont égales) peut être rejetée en fonction de la valeur p et du niveau alpha choisi (0,05 dans ce cas)

Ce test compare les moyennes de quatre groupes: 0.0, 1.0, 2.0 et 3.0.

- Pour le groupe 0.0 et 1.0, la valeur p-adj est de 0.0, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05.
- Pour le groupe 1.0 et 3.0, la valeur p-adj est de 0.0413, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05

Shapiro test : Tester l'hypothèse de normalité

- H0 : Les échatillons sont gaussiens

[33]:
```
# Split the data
x = df_result_all.groupby('label')['taux_brut_de_reussite_total_secteurs'].
 ↪apply(list)
```

[34]:
```
# Perform Shapiro-Wilk test
stat, p = stats.shapiro(x[0])
print("Shapiro-Wilk test: statistic=%f, p-value=%f" % (stat, p))
if p > 0.05:
    print("The data is likely normal")
else:
    print("The data is likely not normal")
```

```
Shapiro-Wilk test: statistic=0.987115, p-value=0.000000
The data is likely not normal
```

p value < 0.05 donc on rejette H0 et on conclut que les échantillons ne sont plutot pas gaussiens

Levene's test : Tester l'hypothèse d'homoscédasticité

- H0 : Les variances sont égales

L'hyphothèse de normalité n'est pas validé, donc on réalise un test de Levene pour tester l'hypothèse d'homoscédasticité

[35]:
```
# Perform Levene's test
stat, p = stats.levene(x[0], x[1], x[2], x[3])
print("Levene's test: statistic=%.3f, p-value=%.3f" % (stat, p))
if p > 0.05:
    print("The variances of the samples are likely similar")
else:
    print("The variances of the samples are likely different")
```

```
Levene's test: statistic=10.186, p-value=0.000
The variances of the samples are likely different
```

p value < 0,05 donc on rejette H0 et on conclut que les variances des labels ne sont plutot pas égales

2.6.2 Effet de la labélisation sur la valeur ajoutée des lycées professionels

Résultat de l'anova

[36]:
```
lm = sfa.ols('va_reu_total ~ C(label)', data=df_result_all).fit()
anova = sa.stats.anova_lm(lm)
anova
```

[36]:

	df	sum_sq	mean_sq	F	PR(>F)
C(label)	3.0	504.957296	168.319099	3.647392	0.012113
Residual	4284.0	197697.131323	46.147790	NaN	NaN

Analyse de l'anova P_value < alpha (0,05), donc on rejette H0 et on conclut d'une manière significative un effet du label numérique sur la valeur ajoutée de réussite des lycées professionels

Test de Tukey

[37]:
```
# perform Tukey's test
tukey = pairwise_tukeyhsd(endog=df_result_all['va_reu_total'],
                          groups=df_result_all['label'],
                          alpha=0.05)

#display results
print(tukey)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1 group2 meandiff p-adj   lower   upper  reject
-----------------------------------------------------
   0.0    1.0   0.1205 0.9646 -0.5316  0.7726  False
   0.0    2.0  -0.3897 0.8557 -1.6459  0.8666  False
   0.0    3.0  -2.9043 0.0091 -5.2793 -0.5293   True
   1.0    2.0  -0.5102 0.7634 -1.8542  0.8339  False
   1.0    3.0  -3.0248 0.0073 -5.4474 -0.6023   True
   2.0    3.0  -2.5147 0.0701 -5.1645  0.1352  False
-----------------------------------------------------
```

Analyse de test de tukey Ce test compare les moyennes de quatre groupes: 0.0, 1.0, 2.0 et 3.0.

- Pour le groupe 0.0 et 3.0, la valeur p-adj est de 0.0091, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05.
- Pour le groupe 1.0 et 3.0, la valeur p-adj est de 0.0073, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05

Shapiro test : Tester l'hypothèse de normalité

- H0 : Les échatillons sont gaussiens

[38]:
```
# Split the data
x = df_result_all.groupby('label')['va_reu_total'].apply(list)
```

[39]:
```
# Perform Shapiro-Wilk test
stat, p = stats.shapiro(x[0])
print("Shapiro-Wilk test: statistic=%f, p-value=%f" % (stat, p))
if p > 0.05:
    print("The data is likely normal")
else:
    print("The data is likely not normal")
```

```
Shapiro-Wilk test: statistic=0.992140, p-value=0.000000
The data is likely not normal
```

Levene's test : Tester l'hypothèse d'homoscédasticité

- H0 : Les variances sont égales

[40]:
```
# Perform Levene's test
stat, p = stats.levene(x[0], x[1], x[2], x[3])
print("Levene's test: statistic=%.3f, p-value=%.3f" % (stat, p))
if p > 0.05:
    print("The variances of the samples are likely similar")
else:
    print("The variances of the samples are likely different")
```

```
Levene's test: statistic=6.494, p-value=0.000
The variances of the samples are likely different
```

3 PARTIE 3 : Analyse des résultats des lycées d'enseignement général et technologique

- En premier temps, nous allons procéder à une analyse des résultats et de la valeur ajoutée des lycées d'enseignement général et technologique en fonction du labels numérique et nous allons expliquer en quoi consiste la valeur ajoutée.
- En deuxième temps, nous allons procéder à une ANOVA à un facteur pour répondre à la question : y a t'il un effet significatif du label numérique sur le taux de réussite des élèves des lycées d'enseignement général et technologique au baccalauréat

- En troisième temps, nous allons procéder à une ANOVA à un facteur pour répondre à la question : y a t'il un effet significatif du label numérique sur la valeur ajoutée des lycées proffessionels au baccalauréat

3.1 PARTIE 3.1 : Importer, transformer et nettoyer les données nécessaires à l'analyse

3.1.1 Dataframe des résultats des lycées d'enseignement Général et technologique

import data from csv

[41]:
```
df2 = pd.read_csv("../data/lycee/csv/
  ↪fr-en-indicateurs-de-resultat-des-lycees-denseignement-general-et-technologique.
  ↪csv", sep=";")
```

```
/tmp/ipykernel_42746/3805400727.py:1: DtypeWarning:

Columns (47,48,49,51,52,53,55,56,57,58,59,71,72,73,74,75,76,77,78,79,80,81,93,95
,98,100,111,113,115,126,129,130,131,134,136) have mixed types. Specify dtype
option on import or set low_memory=False.
```

Nettoyage

[42]:
```
# nettoyage
# filtrer les données sur occitanie
```

```
df2 = df2.query("Region == 'OCCITANIE'")
```

Transform

[43]:
```
# transformation

# keep df lycee general in this variable for later use
df_lycee_gen = df2.copy()

#RENOMMAGE
df2.rename(columns={
    "Annee" : "annee",
    "Taux de reussite - Toutes series" : "taux_reussite_toutes_series",
    "Valeur ajoutee du taux de reussite - Toutes series" :␣
 ↪"va_taux_reussite_toutes_series",
    "Taux de reussite attendu france - Toutes series" :␣
 ↪"taux_reussite_attendu_toutes_series",
    "Departement" : "departement"
},inplace=True)


#transform the annee to numeric
df2["annee"] = pd.to_numeric(df2["annee"])
#sort annee by ascending order
df2.sort_values(by="annee",inplace=True)


# transform le taux en numerique
df2["taux_reussite_toutes_series"] = pd.
 ↪to_numeric(df2["taux_reussite_toutes_series"])
```

3.1.2 MERGE :

- Dataframe labels numériques lycee
- Dataframe résultats scolaires des lycee générals et technologiques

[44]:
```
# merge 2 dataframes on the uri (id de l'etablissement scolaire)
df_result = pd.merge(df, df2, left_on='rne', right_on='UAI', how="inner")
```

[45]:
```
# merge 2 dataframes but keep all the lycee that does not have label
df_result_all = pd.merge(df,df2, left_on='rne', right_on='UAI', how="right")
```

Nettoyage df result (resultat des lycées d'enseignement générales et technologiques numériques)

[46]:
```
#nettoyage (supression) des colonnes ou toutes les valeures sont nulles
df_result.dropna(axis=1,how='all',inplace=True)
```

```
#nettoyage (supression) des lignes ou toutes les valeures sont nulles
df_result.dropna(axis=0,how='all',inplace=True)

# L'index 1393 a une valeure ajoutée de réusiite totale = "ND"
# par contre ce lycee a un taux_reussite_toutes_series = 84
# j'ai donc décidé de rempace le "ND" par 0 (neutre) pour ce lycee et pour tous␣
 ↪les lycee qui ont
# une valeure ajoutée de réussite totale = "ND" (non définir)
import re
reg_ex1 = r"ND"
reg_ex2 = r"\."
aNegliger = re.compile(f"{reg_ex1}|{reg_ex2}",re.I)
subst = "0"
def apply_reg(str_):
    '''Fonction pour remplacer les "ND" par 0 tout court dans une colonne
        - paramètres :
            - str_ : la chaine de caractère dans laquelle la fonction va␣
 ↪chercher la pattern à nettoyé
        - Return la chaine de caractère nettoyé '''
    if pd.notna(str_):
        return re.sub(aNegliger, subst, str(str_))
    else:
        return str_

df_result['va_taux_reussite_toutes_series'] =␣
 ↪df_result['va_taux_reussite_toutes_series'].apply(apply_reg)
```

Nettoyage df result all (resultat de tous les lycées d'enseignement générales et technologiques)

[47]:
```
#nettoyage (supression) des colonnes ou toutes les valeures sont nulles
df_result_all.dropna(axis=1,how='all',inplace=True)

#nettoyage (supression) des lignes ou toutes les valeures sont nulles
df_result_all.dropna(axis=0,how='all',inplace=True)

# L'index 1393 a une valeure ajoutée de réusiite totale = "ND"
# par contre ce lycee a un taux_reussite_toutes_series = 84
# j'ai donc décidé de rempace le "ND" par 0 (neutre) pour ce lycee et pour tous␣
 ↪les lycee qui ont
# une valeure ajoutée de réussite totale = "ND" (non définir)
df_result_all['va_taux_reussite_toutes_series'] =␣
 ↪df_result_all['va_taux_reussite_toutes_series'].apply(apply_reg)
```

Transform df qui contient les resultat des lycées labelisés seulement (df_result)

[48]:
```
# transformer les dates en numériques pour la prochaine opération
df_result["annee_x"] = pd.to_numeric(df_result["annee_x"])
df_result["annee_y"] = pd.to_numeric(df_result["annee_y"])

# ajouter une colonne qui contient boolean (resultat après obtention label ou␣
 ↪pas)
# True si oui, false sinon
# les resultats des lycee après l'obtention de leurs labels
# (annee_x pour obtention label, annee_y pour passage examens de baccalauréat)
df_result["resultat_apres_label"] = (df_result["annee_y"] >=␣
 ↪df_result["annee_x"])

# transformer la valeur ajoutée de réussites totale en numérique
df_result["va_taux_reussite_toutes_series"] = pd.
 ↪to_numeric(df_result["va_taux_reussite_toutes_series"])

# transorm taux_reussite_attendu_toutes_series to numeric
df_result["taux_reussite_attendu_toutes_series"] = pd.
 ↪to_numeric(df_result["taux_reussite_attendu_toutes_series"])
```

Transform df qui contient les résultats de tous les lycées généralsinclus sont qui sont labelisés (df_result_all)

[49]:
```
# transformer les dates en numériques pour la prochaine opération
# transformer les dates en numériques pour la prochaine opération
df_result_all["annee_y"] = pd.to_numeric(df_result_all["annee_y"])

# Creer une colonne pour dire si le lycee est labelisé : True/False
df_result_all["label_true"] = (pd.notna(df_result_all["rne"]))

# ajouter une colonne qui contient boolean (resultat après obtention label ou␣
 ↪pas)
# True si oui, false sinon
# les resultats des lycee après l'obtention de leurs labels
# (annee_x pour obtention label, annee_y pour passage examens de baccalauréat)
df_result_all["resultat_apres_label"] = (df_result_all["annee_y"] >=␣
 ↪df_result_all["annee_x"]) & (df_result_all["label_true"] == True)

# remplacer tous les labels à 0 where examen passé avant 2017
df_result_all.loc[df_result_all["resultat_apres_label"] == False, "label"] = 0

# transformer la valeur ajoutée de réussites totale en numérique
df_result_all["va_taux_reussite_toutes_series"] = pd.
 ↪to_numeric(df_result_all["va_taux_reussite_toutes_series"])

# transorm taux_reussite_attendu_toutes_series to numeric
```

```
df_result_all["taux_reussite_attendu_toutes_series"] = pd.
 ↪to_numeric(df_result_all["taux_reussite_attendu_toutes_series"])
```

3.1.3 Transform for visualisations

[50]:
```
# re transform annee to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result["annee_x"] = df_result["annee_x"].apply(str)
df_result["annee_y"] = df_result["annee_y"].apply(str)

df_result_all["annee_x"] = df_result_all["annee_x"].apply(str)
df_result_all["annee_y"] = df_result_all["annee_y"].apply(str)
```

3.2 PARTIE 3.2 : Analyse générale des lycées d'enseignement générales et technologiques en Occitanie

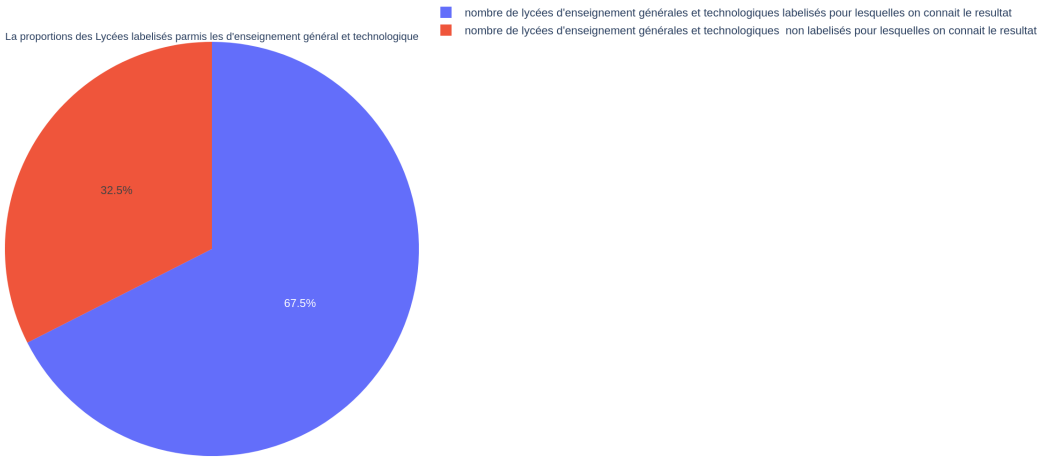
3.2.1 Les proportions des Lycées d'enseignement générales et technologiques labelisés et non labelisés

[51]:
```
nbre_lycee_professionnel = df_result_all['UAI'].nunique()
nbre_lycee_professionel_labelise = df_result_all['rne'].nunique()
nbre_lycee_professionnel_non_labelise = nbre_lycee_professionnel -␣
 ↪nbre_lycee_professionel_labelise
values =␣
 ↪[nbre_lycee_professionnel_non_labelise,nbre_lycee_professionel_labelise]
labels = ["nombre de lycées d'enseignement générales et technologiques  non␣
 ↪labelisés pour lesquelles on connait le resultat", "nombre de lycées␣
 ↪d'enseignement générales et technologiques labelisés pour lesquelles on␣
 ↪connait le resultat"]

fig = go.Figure(data=[go.Pie(labels=labels, values=values, title = "La␣
 ↪proportions des Lycées labelisés parmis les d'enseignement général et␣
 ↪technologique")])
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[51]:



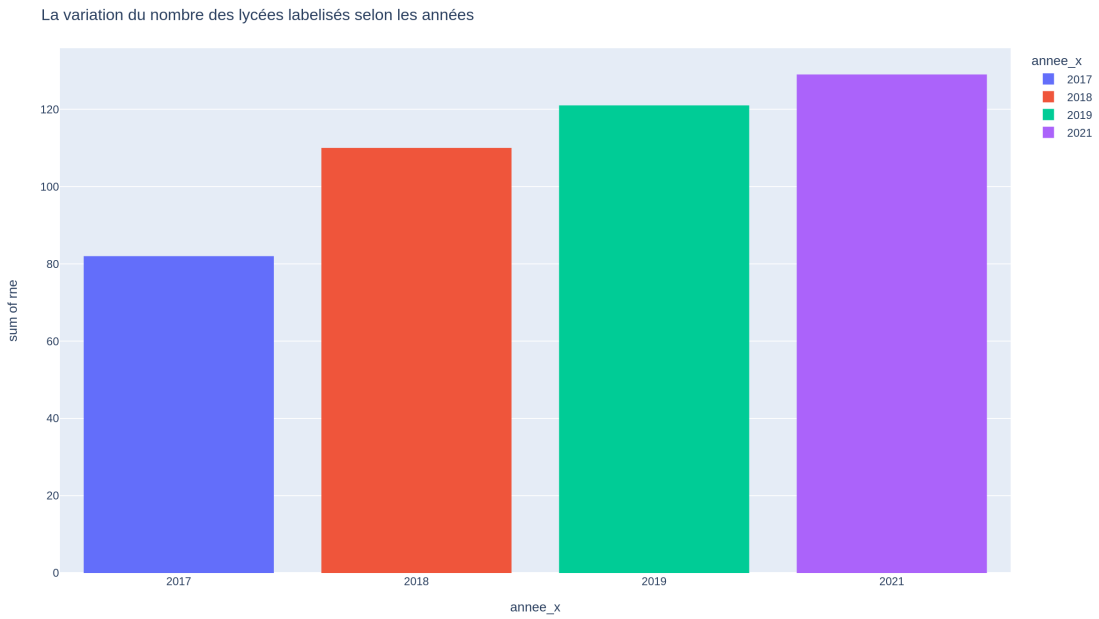
3.2.2 La variation du nombre des lycées labelisés selon les années

[52]:
```
# La variation du nombre des lycées labelisés selon les années
df_proportion_lycee_labelise_annee = df_result.groupby("annee_x")["rne"].
 ↪nunique()
df_proportion_lycee_labelise_annee = df_proportion_lycee_labelise_annee.
 ↪reset_index()

#plotting a histogram
fig = px.histogram(df_proportion_lycee_labelise_annee, x="annee_x", y="rne",
                   color="annee_x",
                      labels={"UAI" : "nombre de lycee labelisés"})
fig.update_layout(title_text="La variation du nombre des lycées labelisés selon␣
 ↪les années")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[52]:



- On constate que le nombre de lycées d'enseignement générales et technologiques labelisés a augmente de 57,31 % entre 2017 et 2021

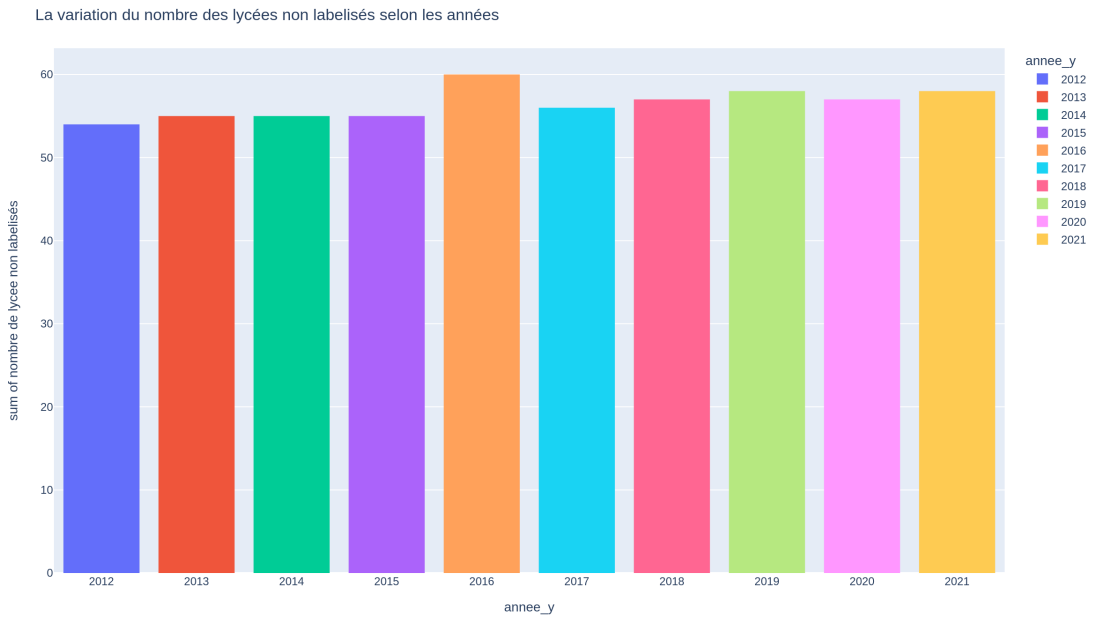
3.2.3 La variation du nombre des lycées non labelisés selon les années

[53]:
```
# La variation du nombre des lycées non labelisés selon les années
df_proportion_lycee_non_labelise_annee = df_result_all.query("label_true ==␣
 ↪False").groupby("annee_y")["UAI"].nunique()
df_proportion_lycee_non_labelise_annee = df_proportion_lycee_non_labelise_annee.
 ↪reset_index()

#plotting a histogram
fig = px.histogram(df_proportion_lycee_non_labelise_annee, x="annee_y", y="UAI",
                   color="annee_y",
                      labels={"UAI" : "nombre de lycee non labelisés"})
fig.update_layout(title_text="La variation du nombre des lycées non labelisés␣
 ↪selon les années")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[53]:



- On constate que le nombre de lycées d'enseignement générales et technologiques non labelisés a augmenté de 7.4% entre 2012 et 2021

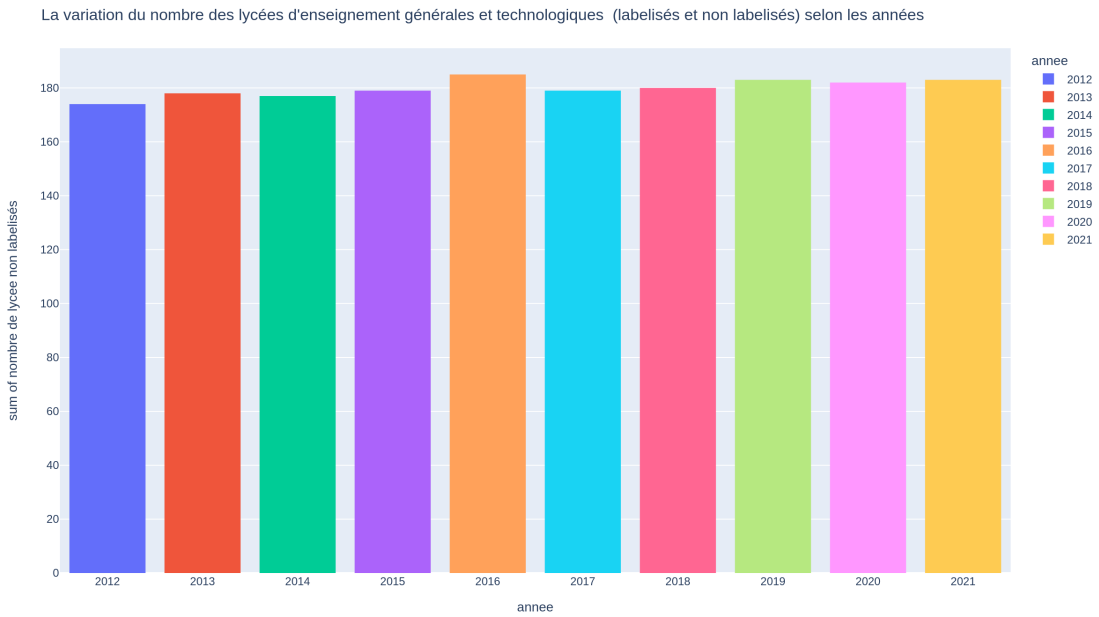
3.2.4 La variation du nombre des lycées d'enseignement générales et technologiques (labelisés et non labelisés) selon les années

[54]:
```
# La variation du nombre des lycées d'enseignement générales et technologiques (labelisé ou non) selon les années
df2["annee"] = df2["annee"].apply(str)
df_proportion_lycee_tout_annee = df2.groupby("annee")["UAI"].nunique()
df_proportion_lycee_tout_annee = df_proportion_lycee_tout_annee.reset_index()
df_proportion_lycee_tout_annee

#plotting a histogram
fig = px.histogram(df_proportion_lycee_tout_annee, x="annee", y="UAI",
                   color="annee",
                      labels={"UAI" : "nombre de lycee non labelisés"})
fig.update_layout(title_text="La variation du nombre des lycées d'enseignement générales et technologiques  (labelisés et non labelisés) selon les années")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[54]:



On constate que le nombre des lycées d'enseignement générales et technologiques (labelisés et non labelisé) a augmenté de 5.17% entre 2012 et 2021

3.2.5 Conclusion

Au fil des années, les lycées d'enseignement générales et technologiques non labelisés et labelisés augmentent

3.3 PARTIE 3.3 : Analyse du taux de réussite constaté des élèves des lycées d'enseignement générales et technologiques au baccalauréat

- Il est intéréssant de voir quelle sont les moyennes de taux de réussite des écoles avant et après l'obtention de leurs labels numérique !

3.3.1 La variance du taux de réussites des élèves au baccalauréat dans les lycées d'enseignement général et technologique (toutes séries) avant et après obtention labels

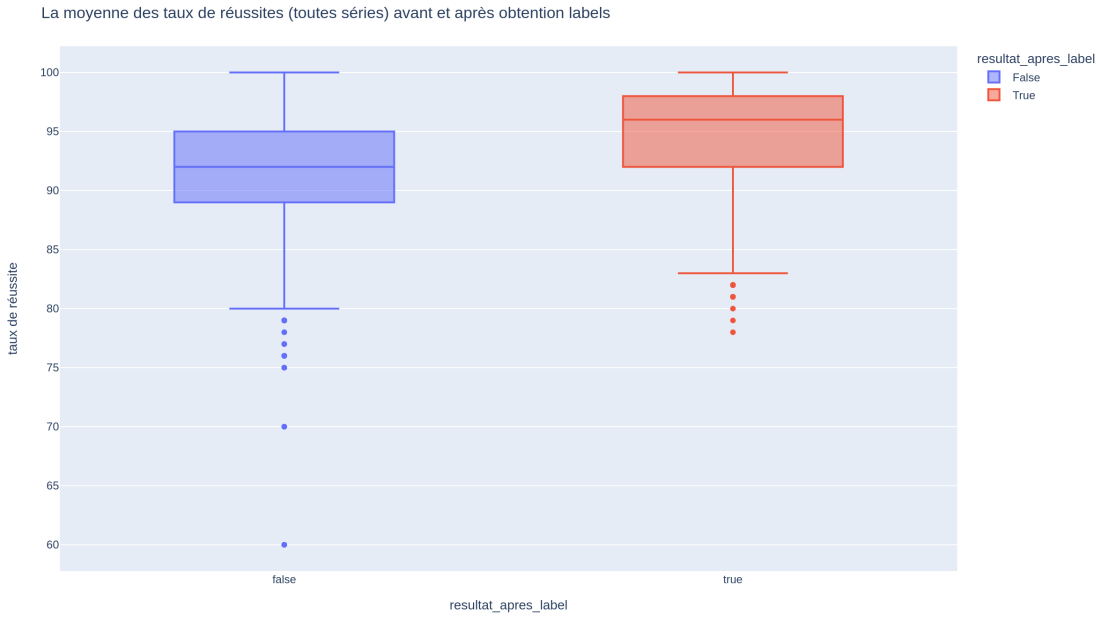
- **True = après obtention label**
- **False = avant obtention label**

[55]:
```
#plotting a histogram
fig = px.box(df_result, x="resultat_apres_label",␣
 ↪y="taux_reussite_toutes_series", color="resultat_apres_label",
                      labels={"taux_reussite_toutes_series" : "taux de␣
 ↪réussite"})
```

```
fig.update_layout(title_text="La moyenne des taux de réussites (toutes séries) avant et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[55]:



Comparaison des taux de réussites constatés avant et après obtention des labels numériques

- Le graphique ci-dessus montre une amélioration des taux de réussite des lycées après obtention de leurs labels numérique !
- On remarque que le taux de réussite median était **92%** avant obtention du label numérique, et la médiane de ce taux a augmenté à **96%** après obtention du label numérique. Il existe donc une variance inter-colonnes (due au facteur du label numérique) intéressante. Ce qui nous ramène par la suite à faire une ANOVA. Nous allons expliquer ce point dans la suite de cette analyse.
- IL est donc intéréssant d'aller chercher pourquoi ? est ce que l'école a apporté plus de valeur pour les élèves grace au numérique ?
- C'est intéressant comme résultat, nous allons aussi voir par la suite la différence de la moyenne du taux de réussites des élèves au baccalauréat dans les lycées d'enseignement général et technologique qui ne sont pas labelisés du tout et ceux qui sont labelisés

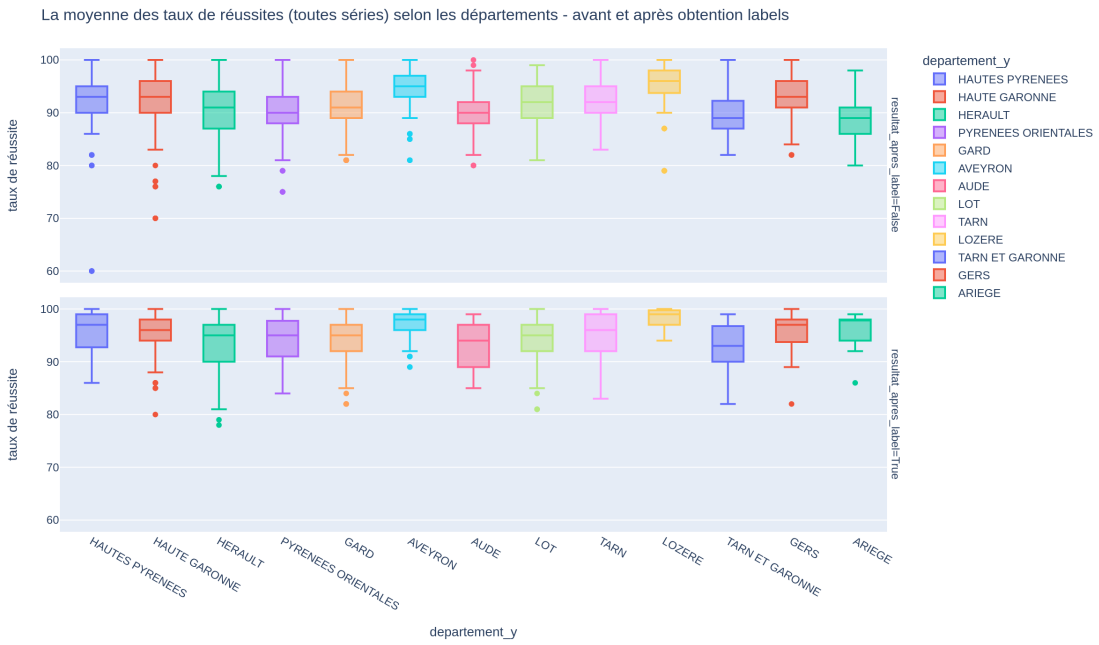
3.3.2 Boxplot: Les taux de réussites des élèves au baccalauréat dans les lycées d'enseignement général et technologique avant et après obtention labels par DÉPARTEMENT

- La partie supérieure montre le taux de réussite avant obtention du label
- La partie inférieure montre le taux de réussite après obtention du label

[56]:
```
#plotting a histogram
fig = px.box(df_result, x="departement_y", y="taux_reussite_toutes_series",
             color="departement_y",
                      facet_row="resultat_apres_label",
                     labels={"taux_reussite_toutes_series" : "taux de␣
 ↪réussite"})
fig.update_layout(title_text="La moyenne des taux de réussites (toutes séries)␣
 ↪selon les départements - avant et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[56]:



INTERPRÉTATION

- ON constate une amélioration du taux de réussite des lycées dans chaque département, notamment le département du **ARIEGE** qui voit la médiane de son taux de réussite augmenté de **89%** avant obtention du label à **98%** après obtention du label

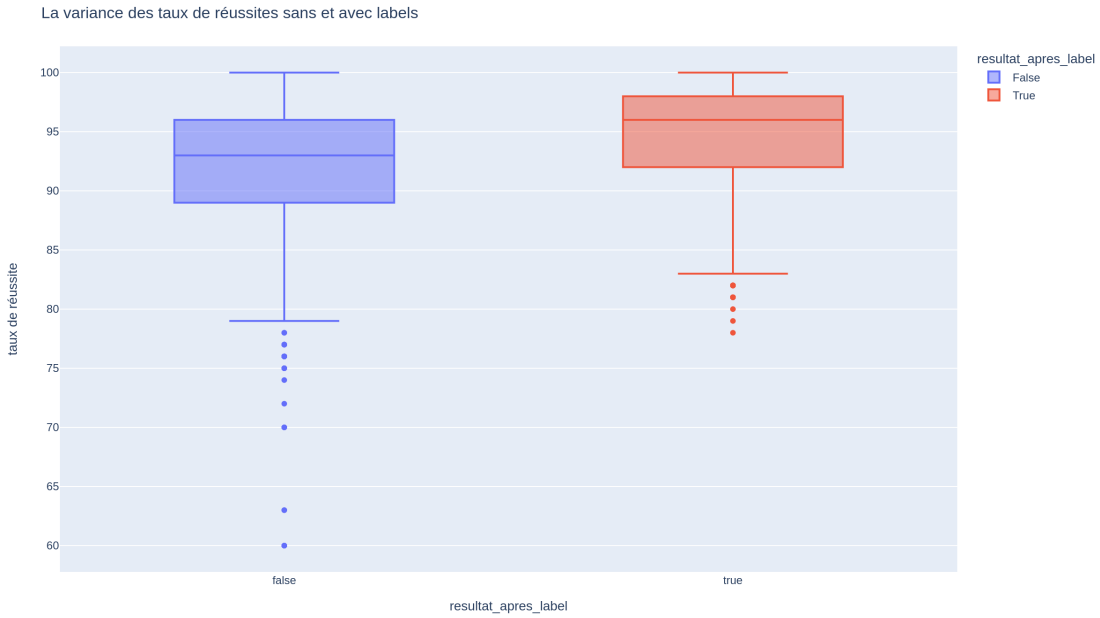
On va voir maintenant la différence du taux de réussites des élèves au baccalauréat dans les lycées d'enseignement général et technologique qui ne sont pas labelisés du tout et ceux qui sont labelisés

3.3.3 BOXPLOT : La variance des taux de réussites au baccalauréat sans et avec labels

[57]:
```
#plotting a bocplot
fig = px.box(df_result_all, x="resultat_apres_label",␣
 ↪y="taux_reussite_toutes_series",
             color="resultat_apres_label",
            labels={"taux_reussite_toutes_series" : "taux de réussite"})
fig.update_layout(title_text="La variance des taux de réussites sans et avec␣
 ↪labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[57]:



Comparaison des taux de réussites constatés SANS et AVEC label numérique

- Le graphique ci-dessus montre que les lycées AVEC labels ont un meilleur taux de réussite que ceux SANS label!
- On remarque que le taux de réussite median est à **93%** SANS label numérique, et la médiane de ce taux a augmenté à **96%** pour les lycées ayant un niveau de label numérique.

3.3.4 Variance du taux de réussite selon le niveau de label numérique

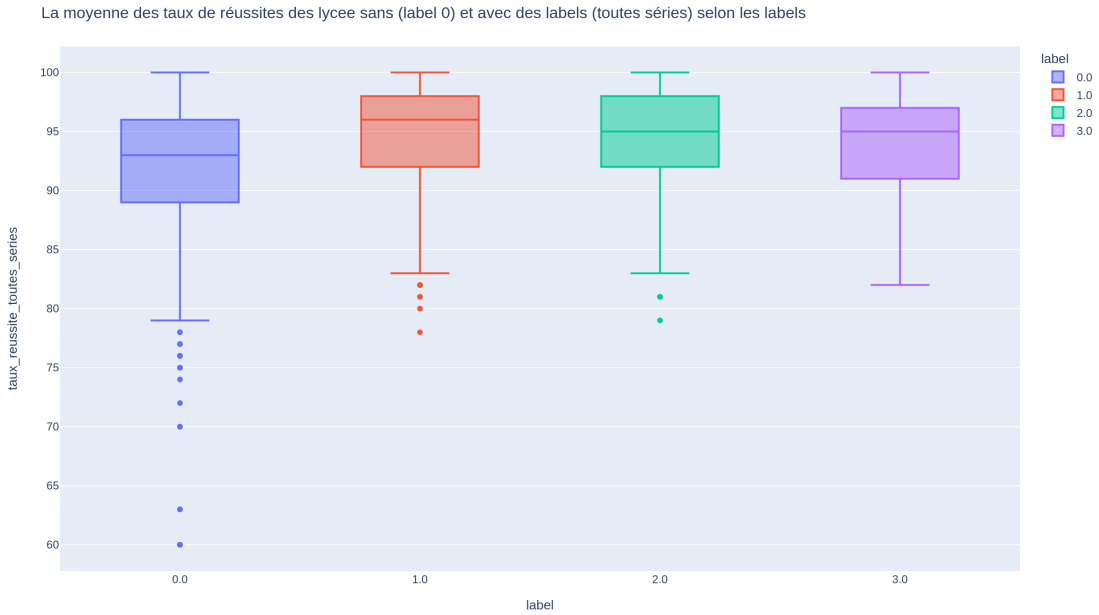
A NOTER LABEL 0 = PAS DE LABEL

[58]:
```
# transform label to numeric
df_result_all["label"] = pd.to_numeric(df_result_all["label"])
#sort label by ascending order
df_result_all.sort_values(by="label",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["label"] = df_result_all["label"].apply(str)

#plotting a histogram
fig = px.box(df_result_all, x="label", y="taux_reussite_toutes_series",␣
 ↪color="label")
fig.update_layout(title_text="La moyenne des taux de réussites des lycee sans␣
 ↪(label 0) et avec des labels (toutes séries) selon les labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[58]:



INTERPRÉTATION :

- On remarque que le niveau de label 1 a la médiane du taux de réussite la plus élevé par parmis les niveaux de labels
- Le niveau de **label 0 (c'est à dire aucun label)** à la médiane du taux de réussite la plus basse

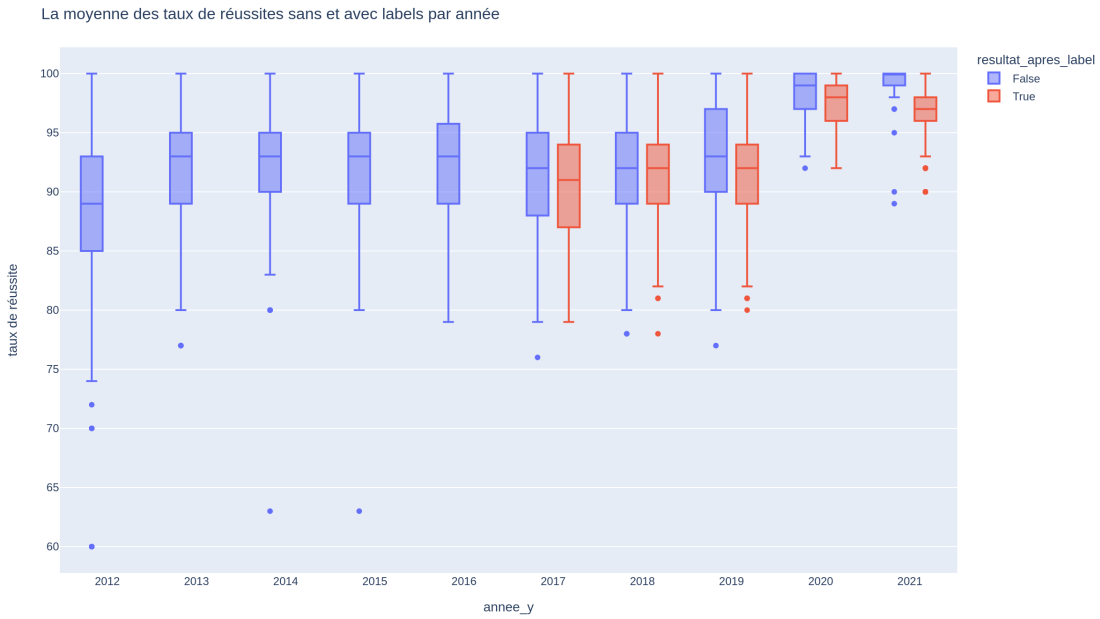
3.3.5 BOXPLOT : La variance du taux de réussites sans et avec labels par année

[59]:
```
# transform label to numeric
df_result_all["annee_y"] = pd.to_numeric(df_result_all["annee_y"])
#sort label by ascending order
df_result_all.sort_values(by="annee_y",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["annee_y"] = df_result_all["annee_y"].apply(str)

#plotting a bocplot
fig = px.box(df_result_all, x="annee_y", y="taux_reussite_toutes_series",
             color="resultat_apres_label",
            labels={"taux_reussite_toutes_series" : "taux de réussite"})
fig.update_layout(title_text="La moyenne des taux de réussites sans et avec␣
 ↪labels par année")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[59]:



3.3.6 Interprétation : On pose l'hypothèse H0

On constate que les lycées d'enseignement générales et technologiques labelisés ont un meilleur résultat que ceux non labelisés (toutes années confondues)

- On se demande pourquoi ?

- On sait que parmis les lycées d'enseignement générales et technologiques , 32,5% ne sont pas labelisés, et 67,5% sont labelisés.
- Alors nous souhaitons répondre à la question suivante : y'a t'il un effet de la labélisation sur le taux de réussite des lycées d'enseignement générales et technologiques ?
- Pour cela on pose les hypothèses suivantes :
  - H0 : La labélisation d'un lycee professionel n'a pas d'effet sur son taux de réussite toutes séries
  - H1 : La labélisation d'un lycee professionel a un effet sur son taux de réussite toutes séries

3.3.7 Covid

Nous constatons un pic de taux de réussite des lycées d'enseignement générales et technologiques en 2020, puis une diminution de ce taux en 2021

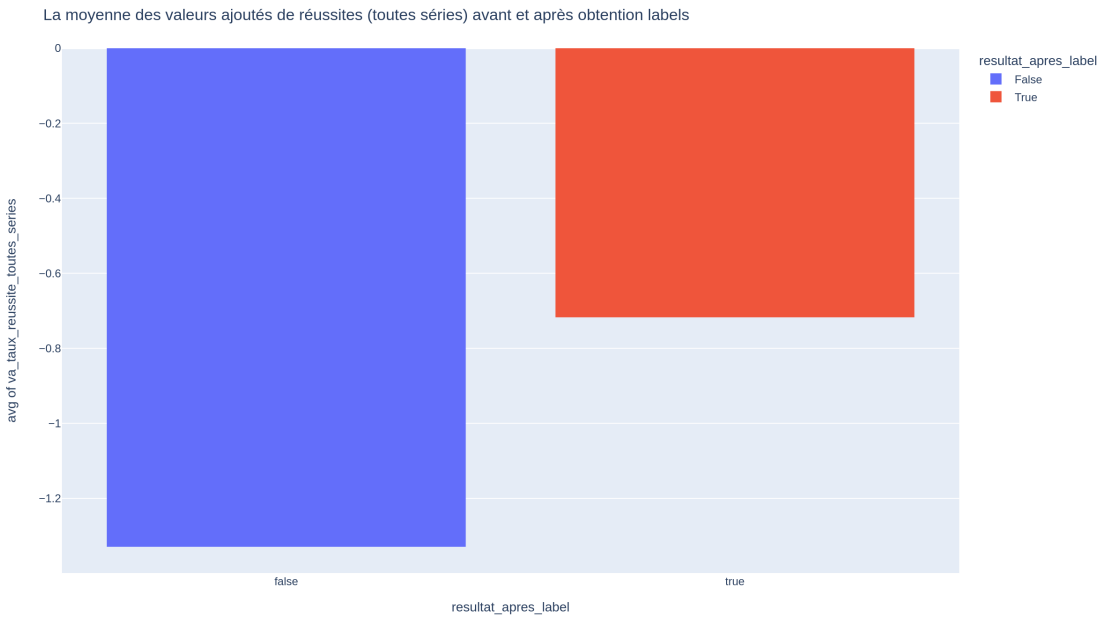
3.4 PARTIE 3.4 : Analyse de la valeur ajoutée de réussite

3.4.1 Histogramme : La moyenne des valeurs ajoutés de réussites avant et après obtention labels

[60]:
```
#plotting a histogram
fig = px.histogram(df_result, x="resultat_apres_label",␣
 ↪y="va_taux_reussite_toutes_series", nbins=10, histfunc="avg",␣
 ↪color="resultat_apres_label")
fig.update_layout(title_text="La moyenne des valeurs ajoutés de réussites␣
 ↪(toutes séries) avant et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[60]:



Intérprétation

- On constate une amélioration de la moyenne de la valeur ajoutée des lycées d'enseignement général et technologique après obtention de leurs labels numériques
- On remarque que la moyenne de la valeur ajoutée a aumenté de **-1,3** avant obtention label à **-0,7** après obtention label, ce qui est non négligeable à l'échelle de la valeur ajoutée

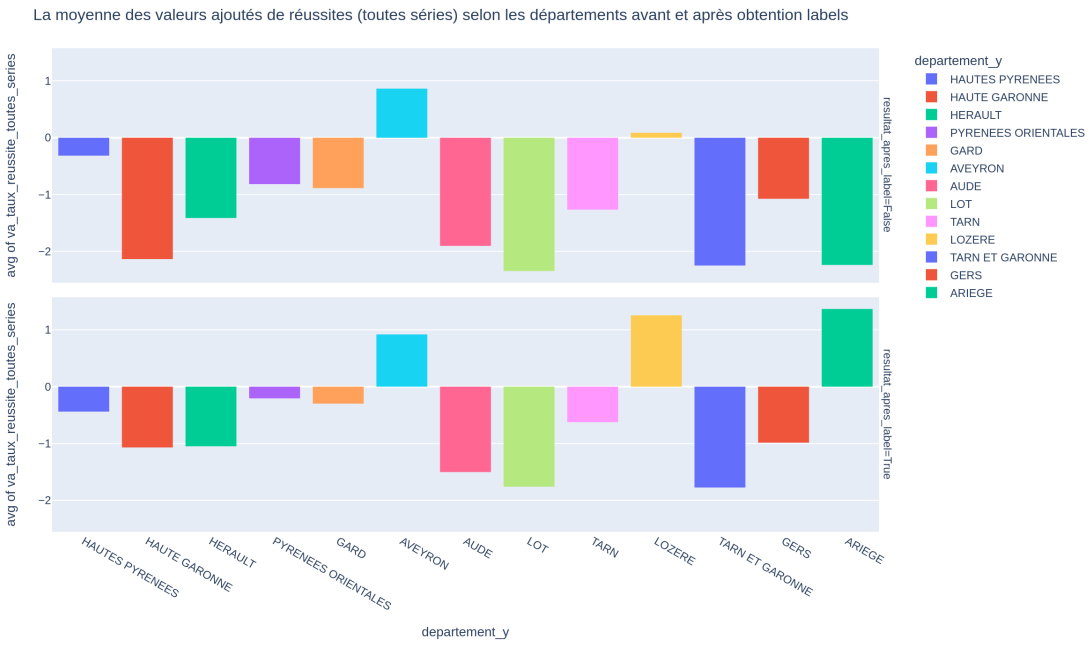
3.4.2 Histogramme : La moyenne des valeurs ajoutés de réussite selon les départements avant et après obtention labels

- La partie supérieure montre le taux de réussite avant obtention du label
- La partie inférieure montre le taux de réussite après obtention du label

[61]:
```
#plotting a histogram
fig = px.histogram(df_result, x="departement_y",␣
 ↪y="va_taux_reussite_toutes_series", nbins=10, histfunc="avg",␣
 ↪color="departement_y", facet_row="resultat_apres_label")
fig.update_layout(title_text="La moyenne des valeurs ajoutés de réussites␣
 ↪(toutes séries) selon les départements avant et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[61]:



Comparaison des valeurs ajoutées des écoles avant et après obtention des labels numériques

- le graphique ci-dessous montre une véritables amélioration des valeurs ajoutée des écoles après obtention de labels. Surtout pour le département de ARIEGE pour lequel la valeur ajoutée est passée du **-2,2** à **+1,3** !

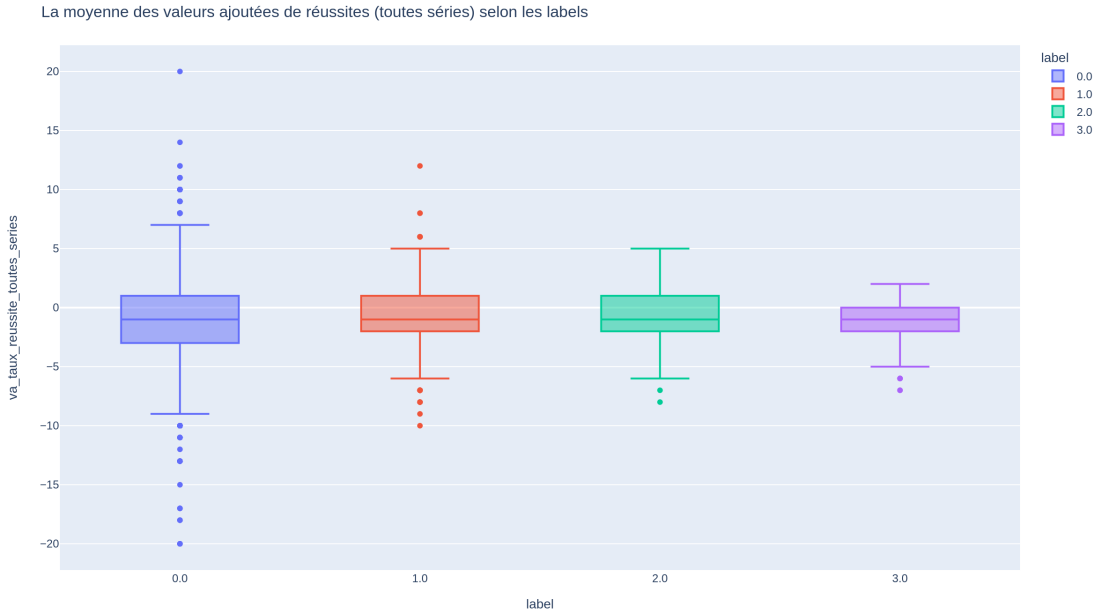
3.4.3 Boxplot : Variance de la valeure ajoutée selon le label (0 = pas de label)

[62]:
```
# transform label to numeric
df_result_all["label"] = pd.to_numeric(df_result_all["label"])
#sort label by ascending order
df_result_all.sort_values(by="label",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["label"] = df_result_all["label"].apply(str)

#plotting a histogram
fig = px.box(df_result_all, x="label", y="va_taux_reussite_toutes_series",␣
 ↪color="label")
fig.update_layout(title_text="La moyenne des valeurs ajoutées de réussites␣
 ↪(toutes séries) selon les labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[62]:



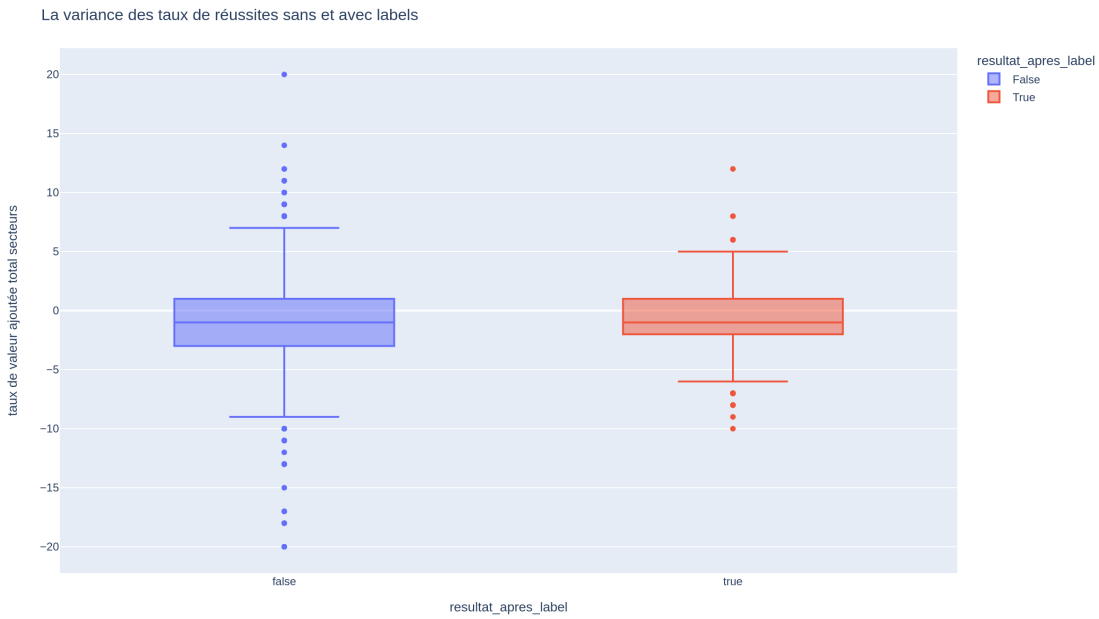
3.4.4 Boxplot : la variance des valeurs ajoutés de réussites des lycées professionnels sans et avec labels

- True = avec label
- False = sans label

[63]:
```
#plotting a bocplot
fig = px.box(df_result_all, x="resultat_apres_label",␣
 ↪y="va_taux_reussite_toutes_series",
             color="resultat_apres_label",
            labels={"va_taux_reussite_toutes_series" : "taux de valeur ajoutée␣
 ↪total secteurs"})
fig.update_layout(title_text="La variance des taux de réussites sans et avec␣
 ↪labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[63]:



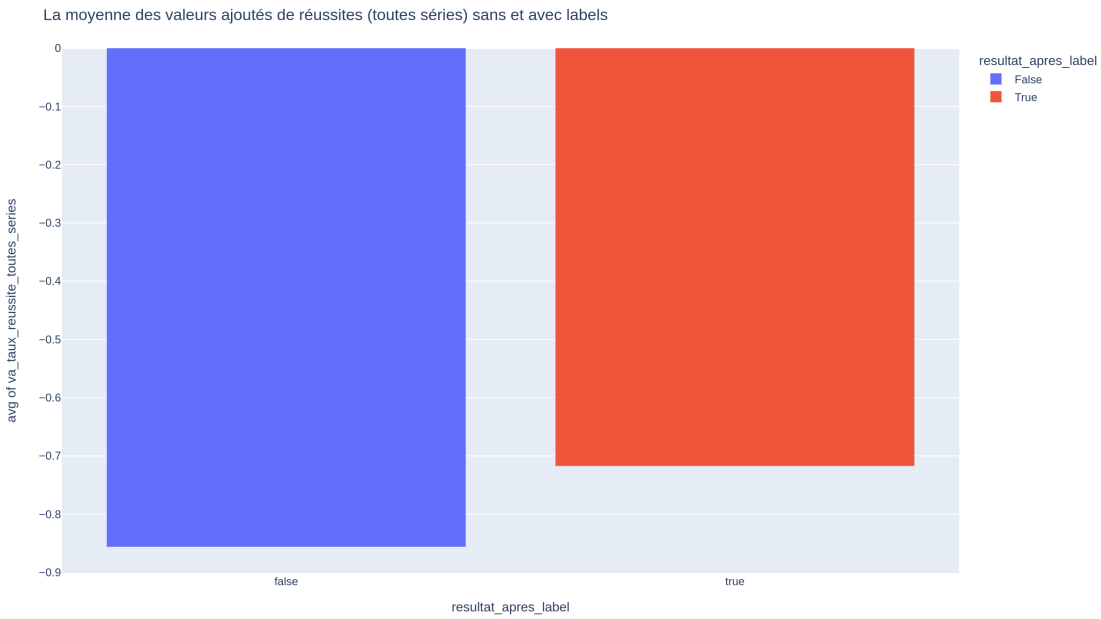
On remarque que la médiane est de la valeur ajoutée est à **-1** que ça soit sans ou avec label

3.4.5 Histogramme : Comparaison de la moyenne des valeurs ajoutés de réussites (toutes séries) des lycées d'enseignement générales et technologiques sans et avec labels

[64]:
```
#plotting a histogram
fig = px.histogram(df_result_all, x="resultat_apres_label",␣
 ↪y="va_taux_reussite_toutes_series", nbins=10, histfunc="avg",␣
 ↪color="resultat_apres_label")
fig.update_layout(title_text="La moyenne des valeurs ajoutés de réussites␣
 ↪(toutes séries) sans et avec labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[64]:



3.5 Partie 3.5 : Analyse du taux de réussite attendu des l'établissement

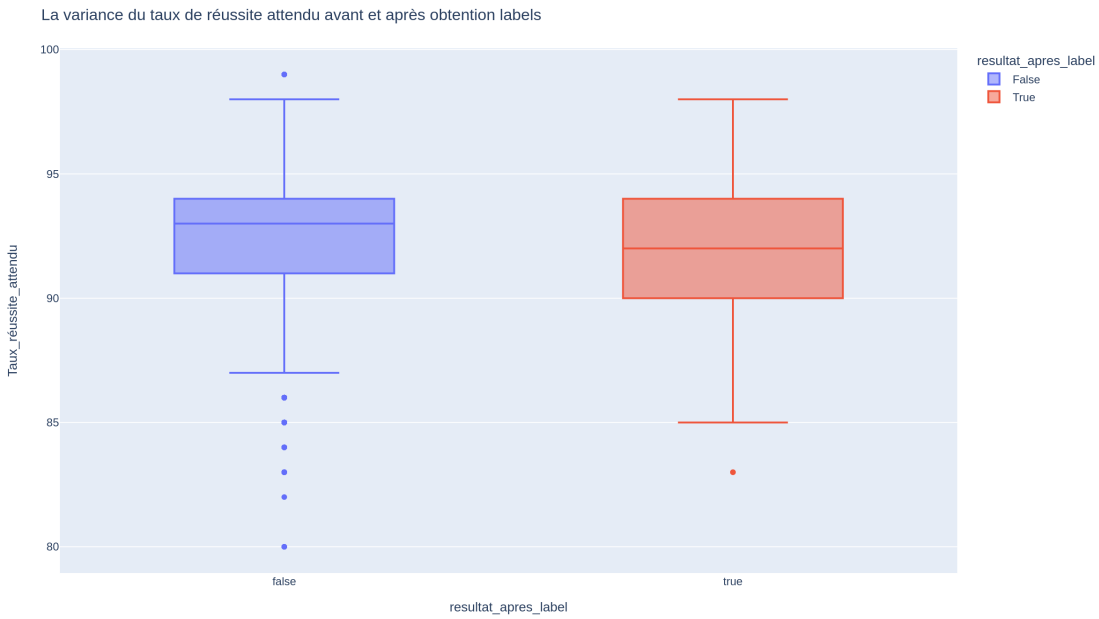
3.5.1 La variance du taux de réussite attendu avant et après obtention label

- True = après label
- False = avant label

[65]:
```
#plotting a histogram
fig = px.box(df_result, x="resultat_apres_label",␣
 ↪y="taux_reussite_attendu_toutes_series", color="resultat_apres_label",
                      labels={"taux_reussite_attendu_toutes_series" :␣
 ↪"Taux_réussite_attendu"})
fig.update_layout(title_text="La variance du taux de réussite attendu avant et␣
 ↪après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[65]:



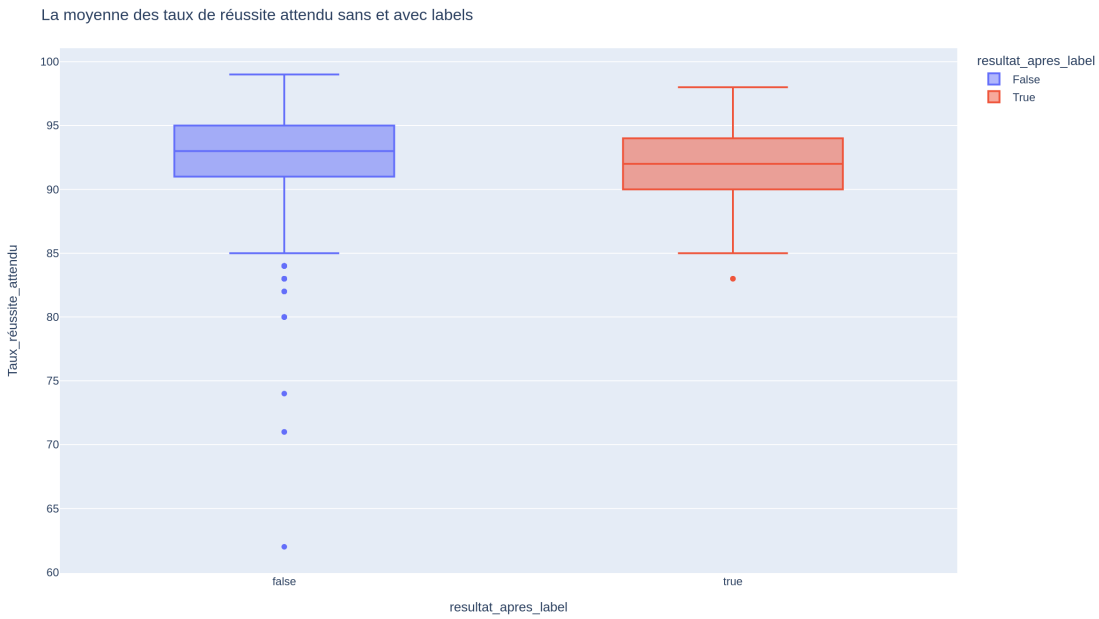
3.5.2 La variance du taux de réussite attendu sans et avec labels

- True = avec label
- False = sans label

[66]:
```
#plotting a histogram
fig = px.box(df_result_all, x="resultat_apres_label",␣
 ↪y="taux_reussite_attendu_toutes_series", color="resultat_apres_label",
                      labels={"taux_reussite_attendu_toutes_series" :␣
 ↪"Taux_réussite_attendu"})
fig.update_layout(title_text="La moyenne des taux de réussite attendu sans et␣
 ↪avec labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[66]:



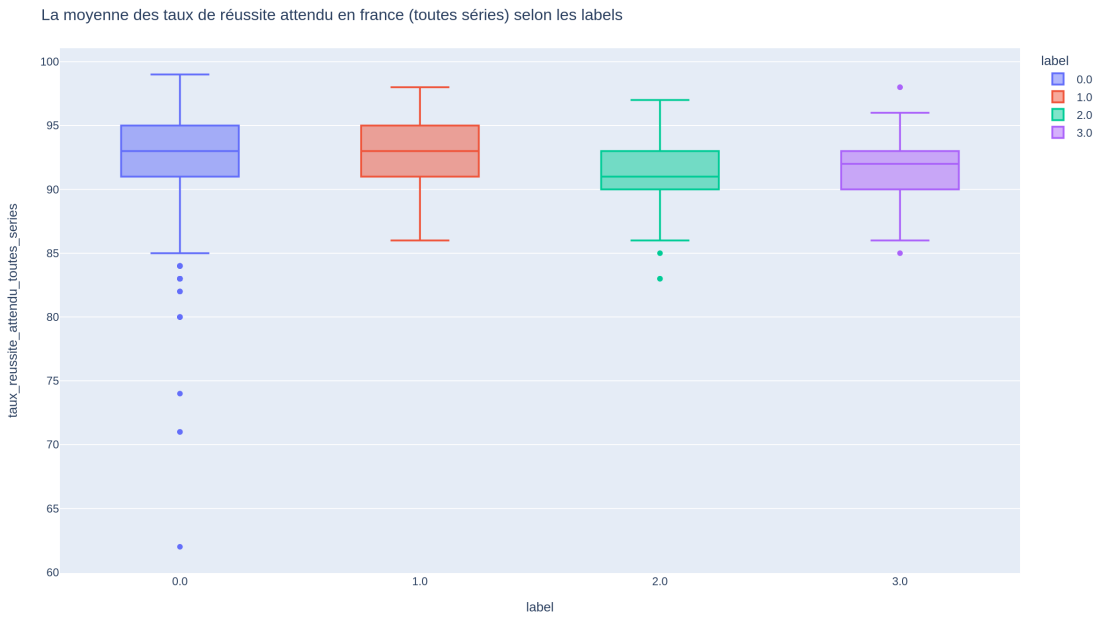
3.5.3 La variance du taux de réussite attendu selon le niveau du label

[67]:
```
# transform label to numeric
df_result_all["label"] = pd.to_numeric(df_result_all["label"])
#sort label by ascending order
df_result_all.sort_values(by="label",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["label"] = df_result_all["label"].apply(str)

#plotting a histogram
fig = px.box(df_result_all, x="label", y="taux_reussite_attendu_toutes_series",␣
 ↪color="label")
fig.update_layout(title_text="La moyenne des taux de réussite attendu en france␣
 ↪(toutes séries) selon les labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[67]:



3.6 PARTIE 3.6 ANOVA

3.6.1 Effet de la labélisation sur le taux de réussite des lycées d'enseignement générales et technologiques toutes séries

[68]:
```
import statsmodels.api as sa
import statsmodels.formula.api as sfa
import scikit_posthocs as sp
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import scipy.stats as stats
```

Résultat de l'anova

[69]:
```
lm = sfa.ols('taux_reussite_toutes_series ~ C(label)', data=df_result_all).fit()
anova = sa.stats.anova_lm(lm)
anova
```

[69]:

	df	sum_sq	mean_sq	F	PR(>F)
C(label)	3.0	5745.308058	1915.102686	82.611728	4.199653e-52
Residual	4775.0	110693.912281	23.181971	NaN	NaN

Analyse de l'anova P_value < alpha (0,05), donc on rejette H0 et on conclut d'une manière significative un effet du label numérique sur le taux de réussite des lycées d'enseignement générales et technologiques toutes séries

Test de Tukey

- permet de préciser quelles modalités de la variable qualitative label a provoqué ce rejet

[70]:
```
# perform Tukey's test
tukey = pairwise_tukeyhsd(endog=df_result_all['taux_reussite_toutes_series'],
                          groups=df_result_all['label'],
                          alpha=0.05)

#display results
print(tukey)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1 group2 meandiff p-adj   lower  upper  reject
-----------------------------------------------------
   0.0    1.0   2.6784    0.0  2.2044 3.1523   True
   0.0    2.0   2.0858    0.0  1.3578 2.8137   True
   0.0    3.0   1.6905 0.0006  0.5685 2.8126   True
   1.0    2.0  -0.5926 0.2433 -1.4092  0.224  False
   1.0    3.0  -0.9879  0.138 -2.1693 0.1936  False
   2.0    3.0  -0.3952 0.8641 -1.6995 0.9091  False
-----------------------------------------------------
```

Analyse de test de tukey Le test de tukey est utilisé pour comparer les moyennes de plusieurs groupes. Le test est utilisé pour déterminer s'il existe des différences significatives entre les moyennes de différents groupes. L'output montre les résultats du test pour chaque comparaison à paires de groupes. Les colonnes de l'output incluent:

- group1 et group2: les groupes étant comparés
- meandiff: la différence des moyennes entre les deux groupes
- p-adj: la valeur p ajustée pour la comparaison
- lower et upper: les limites inférieure et supérieure de l'intervalle de confiance de 95% pour la différence des moyennes
- reject: si oui ou non l'hypothèse nulle (que les moyennes sont égales) peut être rejetée en fonction de la valeur p et du niveau alpha choisi (0,05 dans ce cas)

Ce test compare les moyennes de quatre groupes: 0.0, 1.0, 2.0 et 3.0.

- Pour le groupe 0.0 et 1.0, la valeur p-adj est de 0.0, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05.
- Pour le groupe 0.0 et 2.0, la valeur p-adj est de 0.0, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05
- Pour le groupe 0.0 et 3.0, la valeur p-adj est de 0.0, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05

Shapiro test : Tester l'hypothèse de normalité

- H0 : Les échatillons sont gaussiens

[71]:
```
# Split the data
x = df_result_all.groupby('label')['taux_reussite_toutes_series'].apply(list)
```

[72]:
```
# Perform Shapiro-Wilk test
stat, p = stats.shapiro(x[0])
print("Shapiro-Wilk test: statistic=%f, p-value=%f" % (stat, p))
if p > 0.05:
    print("The data is likely normal")
else:
    print("The data is likely not normal")
```

```
Shapiro-Wilk test: statistic=0.953409, p-value=0.000000
The data is likely not normal
```

p value < 0.05 donc on rejette H0 et on conclut que les échantillons ne sont plutot pas gaussiens

Levene's test : Tester l'hypothèse d'homoscédasticité

- H0 : Les variances sont égales

L'hyphothèse de normalité n'est pas validé, donc on réalise un test de Levene pour tester l'hypothèse d'homoscédasticité

[73]:
```
# Perform Levene's test
stat, p = stats.levene(x[0], x[1], x[2], x[3])
print("Levene's test: statistic=%.3f, p-value=%.3f" % (stat, p))
if p > 0.05:
    print("The variances of the samples are likely similar")
else:
    print("The variances of the samples are likely different")
```

```
Levene's test: statistic=13.646, p-value=0.000
The variances of the samples are likely different
```

p value < 0,05 donc on rejette H0 et on conclut que les variances des labels ne sont plutot pas égales

3.6.2 Effet de la labélisation sur la valeur ajoutée des lycées d'enseignement générales et technologiques toutes séries

Résultat de l'anova

[74]:
```
lm = sfa.ols('va_taux_reussite_toutes_series ~ C(label)', data=df_result_all).
↪fit()
anova = sa.stats.anova_lm(lm)
anova
```

[74]:

	df	sum_sq	mean_sq	F	PR(>F)
C(label)	3.0	99.515442	33.171814	2.931535	0.032255
Residual	4775.0	54031.557167	11.315509	NaN	NaN

Analyse de l'anova P_value < alpha (0,05), donc on rejette H0 et on conclut d'une manière significative un effet du label numérique sur la valeur ajoutée de réussite des lycées d'enseignement générales et technologiques toutes séries

Test de Tukey

[75]:
```
# perform Tukey's test
tukey = pairwise_tukeyhsd(endog=df_result_all['va_taux_reussite_toutes_series'],
                          groups=df_result_all['label'],
                          alpha=0.05)

#display results
print(tukey)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1 group2 meandiff p-adj   lower   upper  reject
-----------------------------------------------------
   0.0    1.0   0.2387 0.2488 -0.0924  0.5698  False
   0.0    2.0   0.1737 0.8165 -0.3349  0.6822  False
   0.0    3.0    -0.62 0.1762 -1.4039  0.1639  False
   1.0    2.0  -0.0651 0.9912 -0.6356  0.5054  False
   1.0    3.0  -0.8587 0.0378 -1.6842 -0.0333   True
   2.0    3.0  -0.7937 0.1132 -1.7049  0.1176  False
-----------------------------------------------------
```

Analyse de test de tukey Ce test compare les moyennes de quatre groupes: 0.0, 1.0, 2.0 et 3.0.

- Pour le groupe 1.0 et 3.0, la valeur p-adj est de 0.0378, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05

Shapiro test : Tester l'hypothèse de normalité

- H0 : Les échatillons sont gaussiens

[76]:
```
# Split the data
x = df_result_all.groupby('label')['va_taux_reussite_toutes_series'].apply(list)
```

[77]:
```
# Perform Shapiro-Wilk test
stat, p = stats.shapiro(x[0])
print("Shapiro-Wilk test: statistic=%f, p-value=%f" % (stat, p))
if p > 0.05:
    print("The data is likely normal")
else:
    print("The data is likely not normal")
```

```
Shapiro-Wilk test: statistic=0.975870, p-value=0.000000
The data is likely not normal
```

Levene's test : Tester l'hypothèse d'homoscédasticité

- H0 : Les variances sont égales

[78]:
```
# Perform Levene's test
stat, p = stats.levene(x[0], x[1], x[2], x[3])
print("Levene's test: statistic=%.3f, p-value=%.3f" % (stat, p))
if p > 0.05:
    print("The variances of the samples are likely similar")
else:
    print("The variances of the samples are likely different")
```

```
Levene's test: statistic=72.948, p-value=0.000
The variances of the samples are likely different
```

4 PARTIE 4 : Analyse conjointe des résultats de tous les lycées Occitanie

- En premier temps, nous allons procéder à une analyse des résultats et de la valeur ajoutée des lycées en fonction du labels numérique et nous allons expliquer en quoi consiste la valeur ajoutée.
- En deuxième temps, nous allons procéder à une ANOVA à un facteur pour répondre à la question : y a t'il un effet significatif du label numérique sur le taux de réussite des élèves des lycées au baccalauréat

- En troisième temps, nous allons procéder à une ANOVA à un facteur pour répondre à la question : y a t'il un effet significatif du label numérique sur la valeur ajoutée des lycées proffessionels au baccalauréat

4.1 PARTIE 4.1 : Importer, transformer et nettoyer les données nécessaires à l'analyse

4.1.1 Dataframe des résultats des lycee professionnels

Transform

[79]:
```
# transformation

# Renommage des colonnes dans df_lycee_pro des lycées professionels pour␣
 ↪pouvoir le concatener après après les lycées générales
df_lycee_pro.rename(columns={
  'code_etablissement' : 'UAI',
  'taux_brut_de_reussite_total_secteurs' : 'taux_reussite_total',
  'va_reu_total' : 'valeur_ajoute_totale',
  'taux_reussite_attendu_france_total_secteurs' : 'taux_reussite_attendu_total'
}, inplace=True)

df_lycee_pro =␣
 ↪df_lycee_pro[["UAI","taux_reussite_total","valeur_ajoute_totale","taux_reussite_attendu_tot
```

4.1.2 Dataframe des résultats des lycées d'enseignement Général et technologique

Transform

[80]:
```
# transformation

#RENOMMAGE
df_lycee_gen.rename(columns={
  "Annee" : "annee",
  "Taux de reussite - Toutes series" : "taux_reussite_total",
  "Valeur ajoutee du taux de reussite - Toutes series" : "valeur_ajoute_totale",
  "Taux de reussite attendu france - Toutes series" :␣
 ↪"taux_reussite_attendu_total",
  "Departement" : "departement"
},inplace=True)


#transform the annee to numeric
df_lycee_gen["annee"] = pd.to_numeric(df_lycee_gen["annee"])
#sort annee by ascending order
df_lycee_gen.sort_values(by="annee",inplace=True)


# transform le taux en numerique
df_lycee_gen["taux_reussite_total"] = pd.
 ↪to_numeric(df_lycee_gen["taux_reussite_total"])

df_lycee_gen =␣
 ↪df_lycee_gen[["UAI","taux_reussite_total","valeur_ajoute_totale","taux_reussite_attendu_tot
```

4.1.3 Concatenation des dataframe lycées professionels et lycées générals

- dataframe df_lycee_pro
- dataframe df_lycee_gen

[81]:
```
df_lycee_all = pd.concat([df_lycee_pro,df_lycee_gen])
```

4.1.4 MERGE :

- dataframe labels numériques lycee
- dataframe résultats scolaires des lycée

[82]:
```
# merge 2 dataframes on the uri (id de l'etablissement scolaire)
df_result = pd.merge(df, df_lycee_all, left_on='rne', right_on='UAI',␣
 ↪how="inner")
```

[83]:
```
# merge 2 dataframes but keep all the lycee that does not have label
df_result_all = pd.merge(df,df_lycee_all, left_on='rne', right_on='UAI',␣
 ↪how="right")
```

Nettoyage df result (resultat des lycées labelisés)

[84]:
```
#nettoyage (supression) des colonnes ou toutes les valeures sont nulles
df_result.dropna(axis=1,how='all',inplace=True)

#nettoyage (supression) des lignes ou toutes les valeures sont nulles
df_result.dropna(axis=0,how='all',inplace=True)

# L'index 1393 a une valeure ajoutée de réusiite totale = "ND"
# par contre ce lycee a un taux_reussite_total = 84
# j'ai donc décidé de rempace le "ND" par 0 (neutre) pour ce lycee et pour tous␣
 ↪les lycee qui ont
# une valeure ajoutée de réussite totale = "ND" (non définir)
import re
reg_ex1 = r"ND"
reg_ex2 = r"\."
aNegliger = re.compile(f"{reg_ex1}|{reg_ex2}",re.I)
subst = "0"
def apply_reg(str_):
  '''Fonction pour remplacer les "ND" par 0 tout court dans une colonne
    - paramètres :
      - str_ : la chaine de caractère dans laquelle la fonction va chercher la␣
 ↪pattern à nettoyé
    - Return la chaine de caractère nettoyé '''
  if pd.notna(str_):
    return re.sub(aNegliger, subst, str(str_))
  else:
    return str_

df_result['valeur_ajoute_totale'] = df_result['valeur_ajoute_totale'].
 ↪apply(apply_reg)
```

Nettoyage df result all (resultat de tous les lycées)

[85]:
```
#nettoyage (supression) des colonnes ou toutes les valeures sont nulles
df_result_all.dropna(axis=1,how='all',inplace=True)

#nettoyage (supression) des lignes ou toutes les valeures sont nulles
df_result_all.dropna(axis=0,how='all',inplace=True)

# L'index 1393 a une valeure ajoutée de réusiite totale = "ND"
# par contre ce lycee a un taux_reussite_total = 84
# j'ai donc décidé de rempace le "ND" par 0 (neutre) pour ce lycee et pour tous␣
 ↪les lycee qui ont
# une valeure ajoutée de réussite totale = "ND" (non définir)
df_result_all['valeur_ajoute_totale'] = df_result_all['valeur_ajoute_totale'].
 ↪apply(apply_reg)
```

Transform df qui contient les resultat des lycées labelisés seulement (df_result)

[86]:
```
# transformer les dates en numériques pour la prochaine opération
df_result["annee_x"] = pd.to_numeric(df_result["annee_x"])
df_result["annee_y"] = pd.to_numeric(df_result["annee_y"])

# ajouter une colonne qui contient boolean (resultat après obtention label ou␣
 ↪pas)
# True si oui, false sinon
# les resultats des lycee après l'obtention de leurs labels
# (annee_x pour obtention label, annee_y pour passage examens de baccalauréat)
df_result["resultat_apres_label"] = (df_result["annee_y"] >=␣
 ↪df_result["annee_x"])

# transformer la valeur ajoutée de réussites totale en numérique
df_result["valeur_ajoute_totale"] = pd.
 ↪to_numeric(df_result["valeur_ajoute_totale"])

# transorm taux_reussite_attendu_total to numeric
df_result["taux_reussite_attendu_total"] = pd.
 ↪to_numeric(df_result["taux_reussite_attendu_total"])
```

Transform df qui contient les résultats de tous les lycées inclus sont qui sont labelisés (df_result_all)

[87]:
```
# transformer les dates en numériques pour la prochaine opération
# transformer les dates en numériques pour la prochaine opération
df_result_all["annee_y"] = pd.to_numeric(df_result_all["annee_y"])

# Creer une colonne pour dire si le lycee est labelisé : True/False
df_result_all["label_true"] = (pd.notna(df_result_all["rne"]))

# ajouter une colonne qui contient boolean (resultat après obtention label ou␣
 ↪pas)
# True si oui, false sinon
# les resultats des lycee après l'obtention de leurs labels
# (annee_x pour obtention label, annee_y pour passage examens de baccalauréat)
df_result_all["resultat_apres_label"] = (df_result_all["annee_y"] >=␣
 ↪df_result_all["annee_x"]) & (df_result_all["label_true"] == True)

# remplacer tous les labels à 0 where examen passé avant 2017
df_result_all.loc[df_result_all["resultat_apres_label"] == False, "label"] = 0

# transformer la valeur ajoutée de réussites totale en numérique
df_result_all["valeur_ajoute_totale"] = pd.
 ↪to_numeric(df_result_all["valeur_ajoute_totale"])

# transorm taux_reussite_attendu_total to numeric
```

```
df_result_all["taux_reussite_attendu_total"] = pd.
 ↪to_numeric(df_result_all["taux_reussite_attendu_total"])
```

4.1.5 Transform for visualisations

[88]:
```
# re transform annee to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result["annee_x"] = df_result["annee_x"].apply(str)
df_result["annee_y"] = df_result["annee_y"].apply(str)

df_result_all["annee_x"] = df_result_all["annee_x"].apply(str)
df_result_all["annee_y"] = df_result_all["annee_y"].apply(str)
```

4.2 PARTIE 4.2 : Analyse conjointe du taux de réussite constaté des élèves des lycées au baccalauréat

- Il est intéréssant de voir quelle sont les moyennes de taux de réussite des écoles avant et après l'obtention de leurs labels numérique !

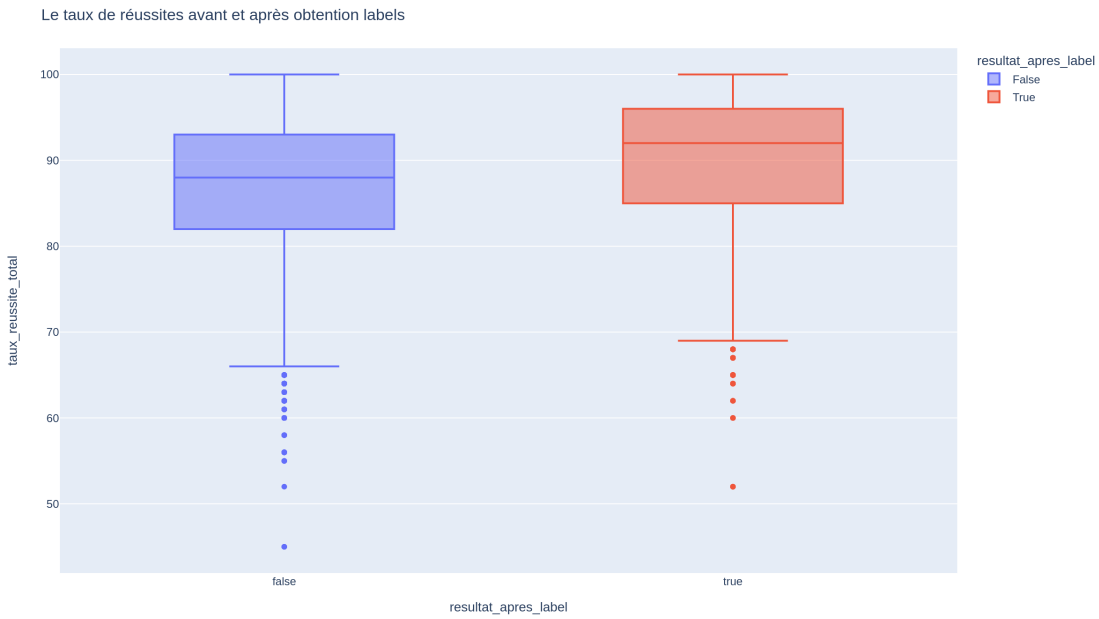
4.2.1 La variance du taux de réussites des élèves au baccalauréat dans les lycées avant et après obtention labels

- **True = après obtention label**
- **False = avant obtention label**

[89]:
```
#plotting a boxplot
fig = px.box(df_result, x="resultat_apres_label", y="taux_reussite_total",
                   color="resultat_apres_label",
                      labels={"taux_brut_de_reussite_total_secteurs" : "taux de␣
 ↪réussite"})
fig.update_layout(title_text="Le taux de réussites avant et après obtention␣
 ↪labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[89]:



Comparaison des taux de réussites constatés avant et après obtention des labels numériques

- Le graphique ci-dessus montre une amélioration des taux de réussite des lycées après obtention de leurs labels numérique !
- On remarque que le taux de réussite median était **88%** avant obtention du label numérique, et la médiane de ce taux a augmenté à **92%** après obtention du label numérique.

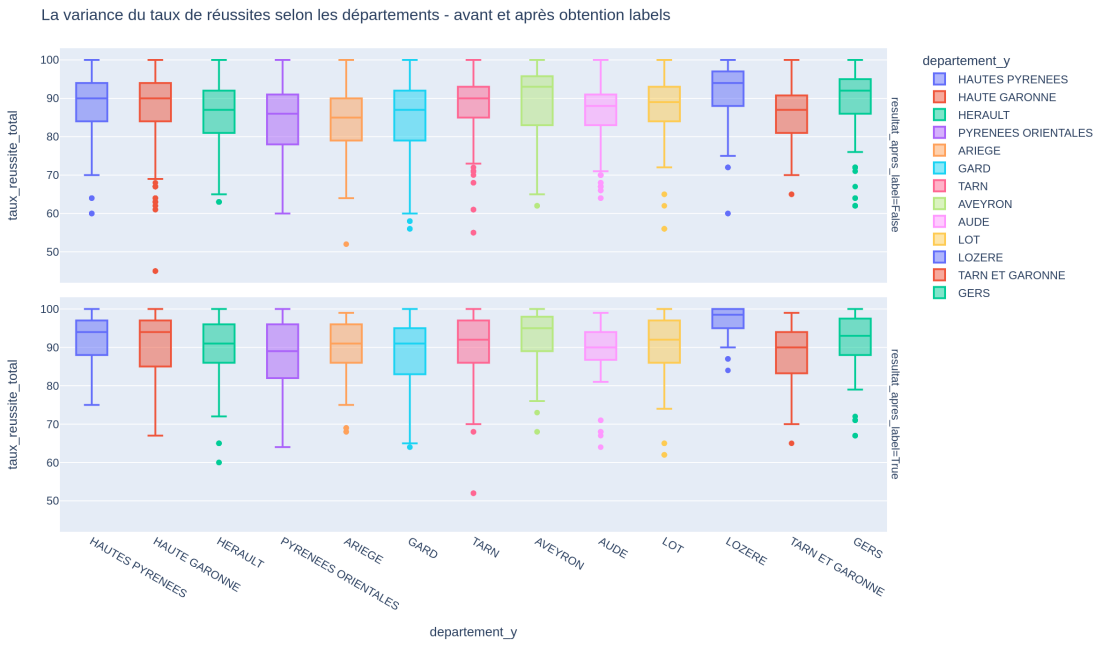
4.2.2 Boxplot: Les taux de réussites des élèves au baccalauréat dans les lycées avant et après obtention labels par DÉPARTEMENT

- La partie supérieure montre le taux de réussite avant obtention du label
- La partie inférieure montre le taux de réussite après obtention du label

[90]:
```
#plotting a histogram
fig = px.box(df_result, x="departement_y", y="taux_reussite_total",␣
 ↪color="departement_y",
                       facet_row="resultat_apres_label",
                      labels={"taux_brut_de_reussite_total_secteurs" : "taux de␣
 ↪réussite"})
fig.update_layout(title_text="La variance du taux de réussites selon les␣
 ↪départements - avant et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[90]:



INTERPRÉTATION

- ON constate une amélioration du taux de réussite des lycées dans chaque département, notamment le département du **HAUTES PURENEES** qui voit la médiane de son taux de réussite augmenté de **90%** avant obtention du label à **94%** après obtention du label

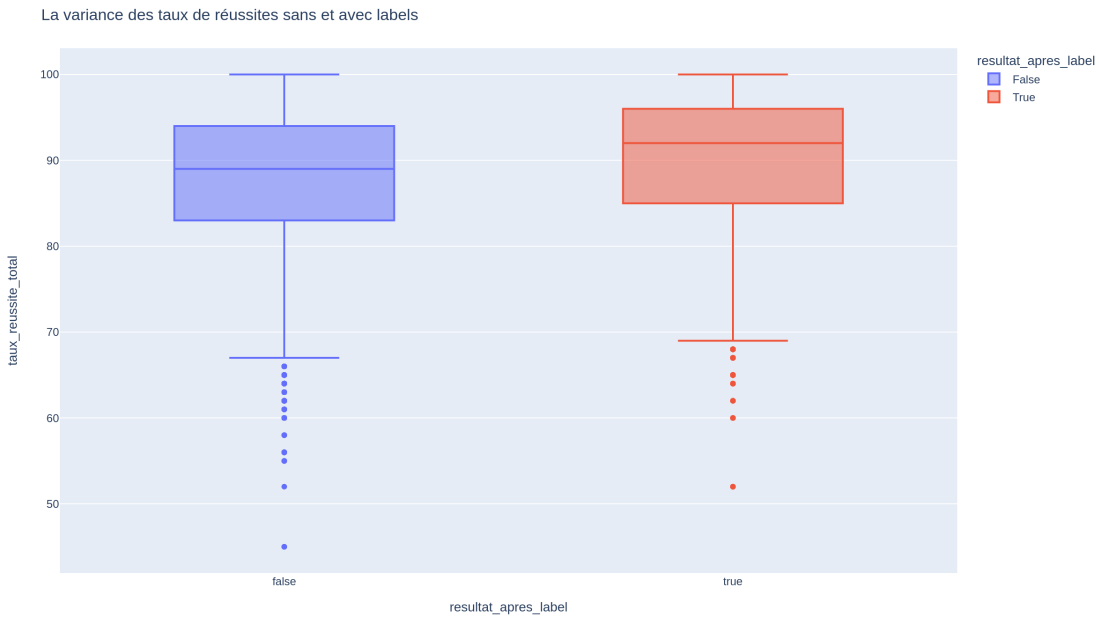
On va voir maintenant la différence du taux de réussites des élèves au baccalauréat dans les lycées qui ne sont pas labelisés du tout et ceux qui sont labelisés

4.2.3 BOXPLOT : La variance des taux de réussites au baccalauréat sans et avec labels

[91]:
```
#plotting a bocplot
fig = px.box(df_result_all, x="resultat_apres_label", y="taux_reussite_total",
             color="resultat_apres_label",
            labels={"taux_brut_de_reussite_total_secteurs" : "taux de␣
 ↪réussite"})
fig.update_layout(title_text="La variance des taux de réussites sans et avec␣
 ↪labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[91]:



Comparaison des taux de réussites constatés SANS et AVEC label numérique

- Le graphique ci-dessus montre que les lycées AVEC labels ont un meilleur taux de réussite que ceux SANS label!
- On remarque que le taux de réussite median est à **89%** SANS label numérique, et la médiane de ce taux a augmenté à **92%** pour les lycées ayant un niveau de label numérique.

4.2.4 Variance du taux de réussite selon le niveau de label numérique

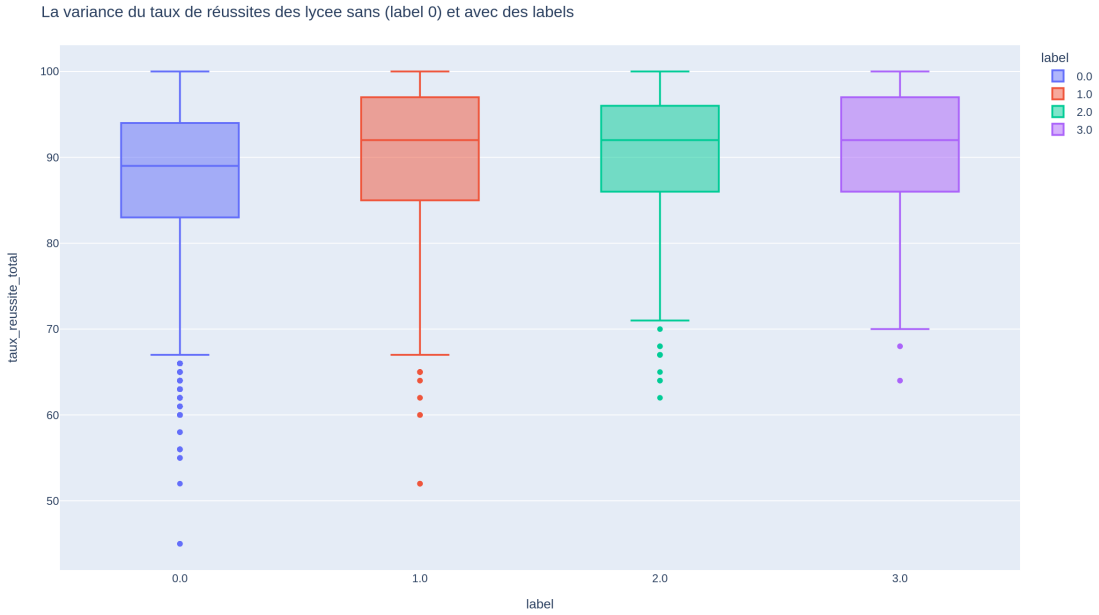
A NOTER LABEL 0 = PAS DE LABEL

[92]:
```
# transform label to numeric
df_result_all["label"] = pd.to_numeric(df_result_all["label"])
#sort label by ascending order
df_result_all.sort_values(by="label",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["label"] = df_result_all["label"].apply(str)

#plotting a histogram
fig = px.box(df_result_all, x="label", y="taux_reussite_total", color="label")
fig.update_layout(title_text="La variance du taux de réussites des lycee sans␣
 ↪(label 0) et avec des labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[92]:



INTERPRÉTATION :

- On remarque que les niveaux de label 1, 2 et 3 ont la meme médiane du taux de réussite (92%)
- Le niveau de **label 0 (c'est à dire aucun label)** à la médiane du taux de réussite la plus basse (89%)

4.2.5 BOXPLOT : La variance du taux de réussites sans et avec labels par année

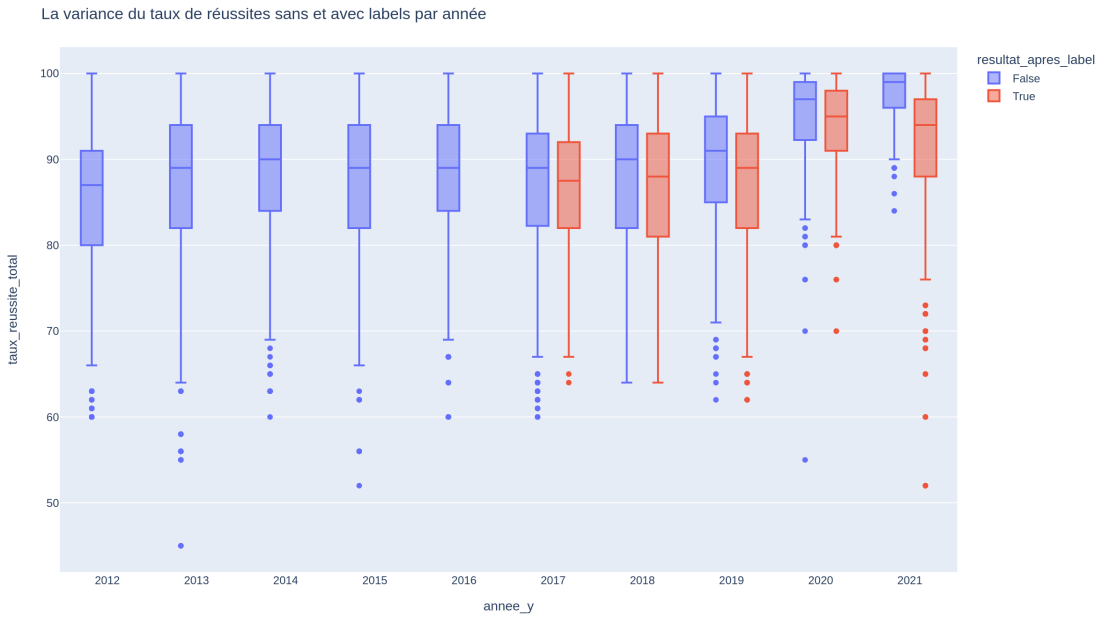
[93]:

```
# transform label to numeric
df_result_all["annee_y"] = pd.to_numeric(df_result_all["annee_y"])
#sort label by ascending order
df_result_all.sort_values(by="annee_y",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["annee_y"] = df_result_all["annee_y"].apply(str)

#plotting a bocplot
fig = px.box(df_result_all, x="annee_y", y="taux_reussite_total",
             color="resultat_apres_label",
            labels={"taux_brut_de_reussite_total_secteurs" : "taux de␣
 ↪réussite"})
fig.update_layout(title_text="La variance du taux de réussites sans et avec␣
 ↪labels par année")
#fig.show()
```

```
image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[93]:



4.2.6 Interprétation : On pose l'hypothèse H0

On constate que les lycées labelisés ont un meilleur résultat que ceux non labelisés (toutes années confondues)

- On se demande pourquoi ?
- On sait que parmis les lycées professionels, 33% ne sont pas labelisés, et 67% sont labelisés.
- Alors nous souhaitons répondre à la question suivante : y'a t'il un effet de la labélisation sur le taux de réussite des lycées professionels ?
- Pour cela on pose les hypothèses suivantes :
  - **H0 : La labélisation d'un lycee n'a pas d'effet sur son taux de réussite**
  - **H1 : La labélisation d'un lycee a un effet sur son taux de réussite**

4.2.7 Covid

Nous constatons un pic de taux de réussite des lycées en 2020, puis une diminution de ce taux en 2021

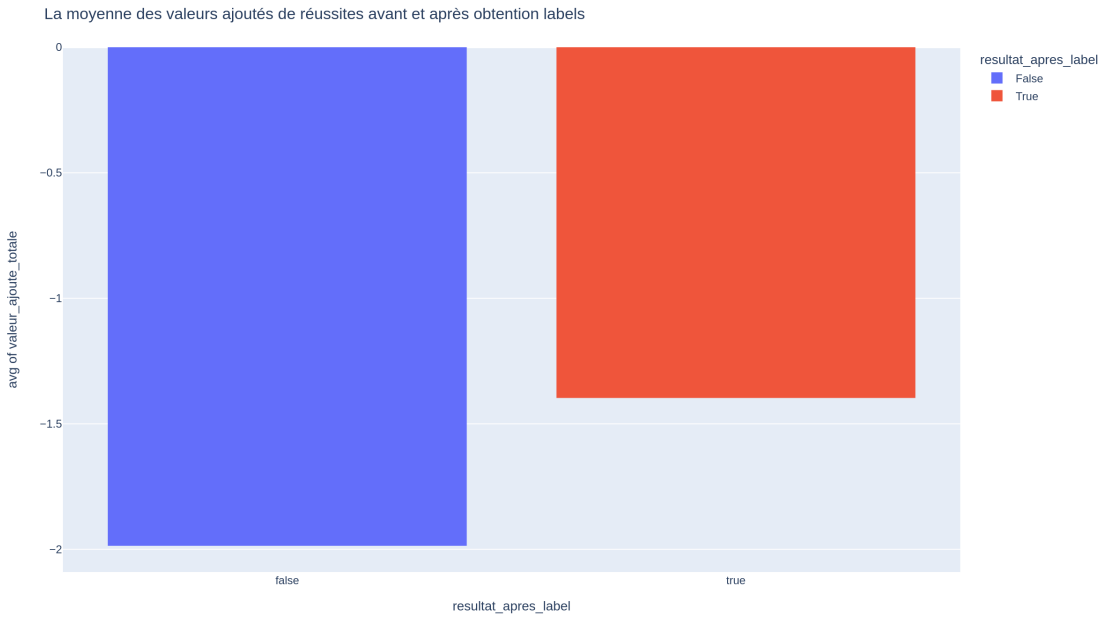
4.3 PARTIE 4.3 : Analyse de la valeur ajoutée des lycées

4.3.1 Histogramme : La moyenne des valeurs ajoutés de réussites avant et après obtention labels

[94]:
```
#plotting a histogram
fig = px.histogram(df_result, x="resultat_apres_label",␣
 ↪y="valeur_ajoute_totale", nbins=10, histfunc="avg",␣
 ↪color="resultat_apres_label")
fig.update_layout(title_text="La moyenne des valeurs ajoutés de réussites avant␣
 ↪et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[94]:



Intérprétation

- On constate une amélioration de la moyenne de la valeur ajoutée des lycées après obtention de leurs labels numériques
- On remarque que la moyenne de la valeur ajoutée a aumenté de **-1.98** avant obtention label à **-1.39** après obtention label, ce qui est non négligeable à l'échelle de la valeur ajoutée

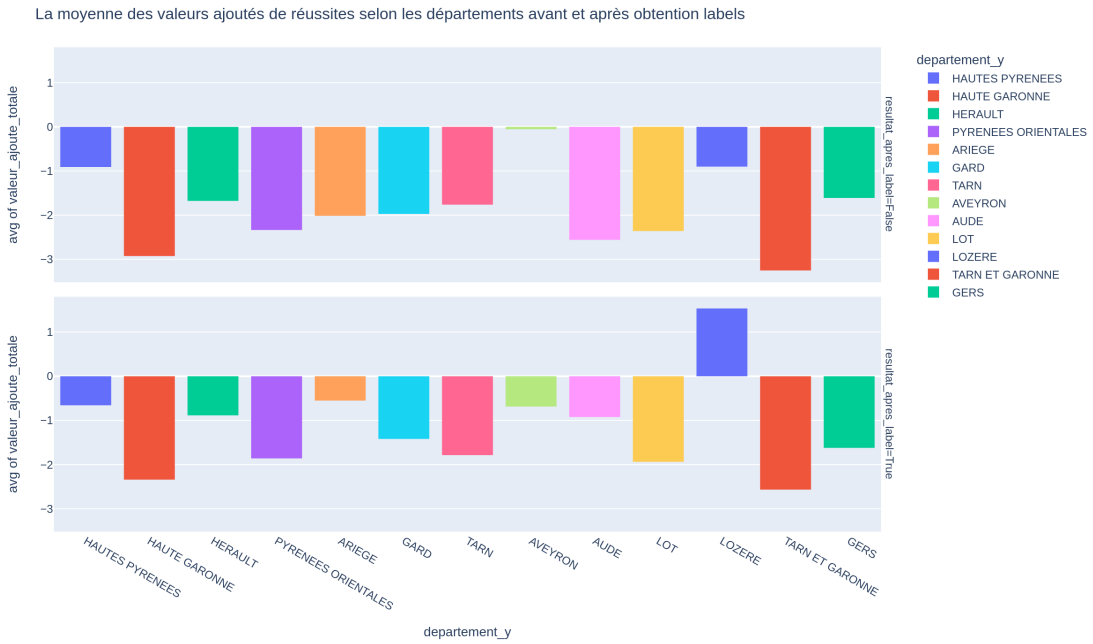
4.3.2 Histogramme : La moyenne des valeurs ajoutés de réussite selon les départements avant et après obtention labels

- La partie supérieure montre le taux de réussite avant obtention du label
- La partie inférieure montre le taux de réussite après obtention du label

[95]:
```
#plotting a histogram
fig = px.histogram(df_result, x="departement_y", y="valeur_ajoute_totale",␣
 ↪nbins=10, histfunc="avg", color="departement_y",␣
 ↪facet_row="resultat_apres_label")
fig.update_layout(title_text="La moyenne des valeurs ajoutés de réussites selon␣
 ↪les départements avant et après obtention labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[95]:



Comparaison des valeurs ajoutées des écoles avant et après obtention des labels numériques

- le graphique ci-dessous montre une véritables amélioration des valeurs ajoutée des écoles après obtention de labels. Surtout pour le département de AUDE pour lequel la valeur ajoutée est passée du **-2,5** à **-0.92** !

4.3.3 HISTOGRAMME : Variance de la valeure ajoutée selon le label (0 = pas de label)

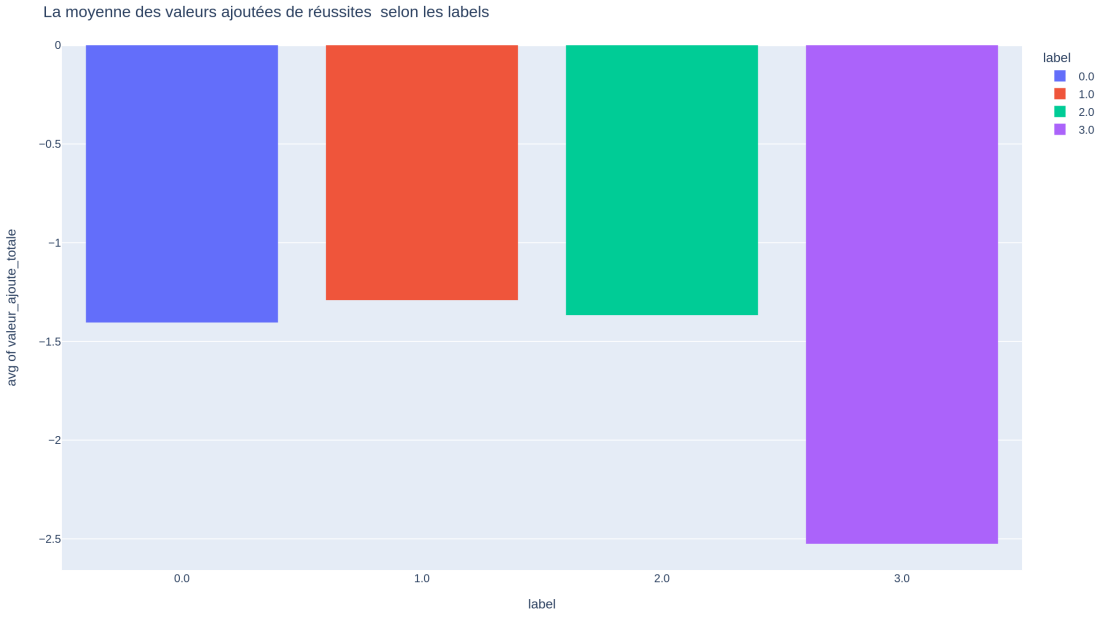
[96]:
```
# transform label to numeric
df_result_all["label"] = pd.to_numeric(df_result_all["label"])
#sort label by ascending order
df_result_all.sort_values(by="label",inplace=True)
```

```
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["label"] = df_result_all["label"].apply(str)

#plotting a histogram
fig = px.histogram(df_result_all, x="label", y="valeur_ajoute_totale",␣
 ↪nbins=10, histfunc="avg", color="label")
fig.update_layout(title_text="La moyenne des valeurs ajoutées de réussites ␣
 ↪selon les labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[96]:



4.3.4 Boxplot : la variance des valeurs ajoutés de réussites des lycées professionnels sans et avec labels

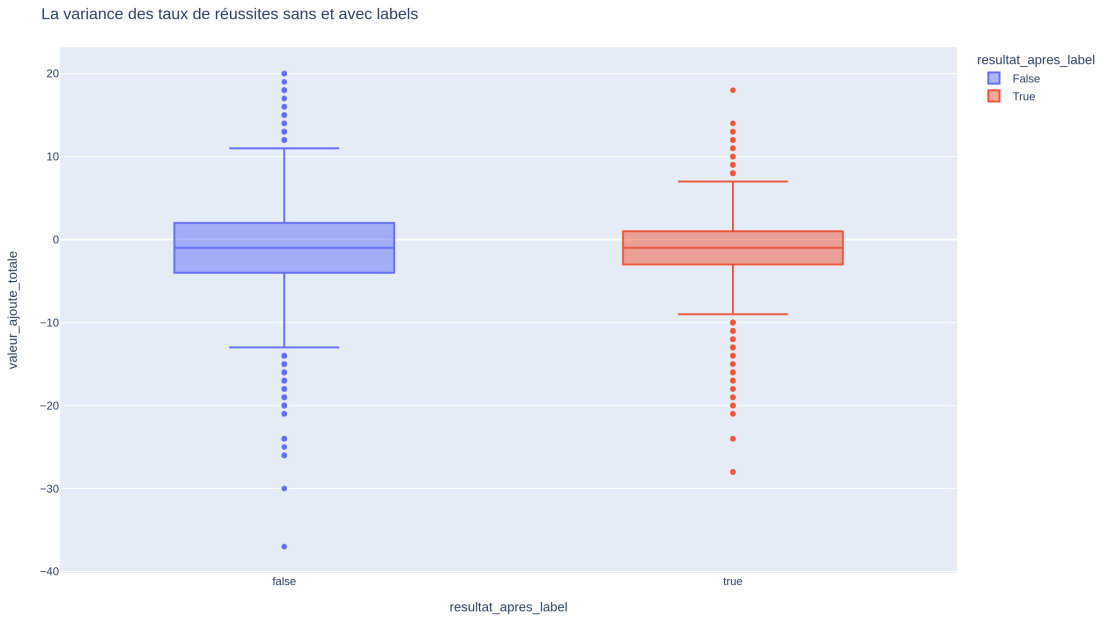
- True = avec label
- False = sans label

[97]:
```
#plotting a bocplot
fig = px.box(df_result_all, x="resultat_apres_label", y="valeur_ajoute_totale",
             color="resultat_apres_label",
            labels={"va_reu_total" : "taux de valeur ajoutée total secteurs"})
fig.update_layout(title_text="La variance des taux de réussites sans et avec␣
 ↪labels")
```

```
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[97]:



On remarque que la médiane est de la valeur ajoutée est à **-1** que ça soit sans ou avec label

4.4 PARTIE 4.4 : Analyse conjointe du taux de réussite attendu des lycées

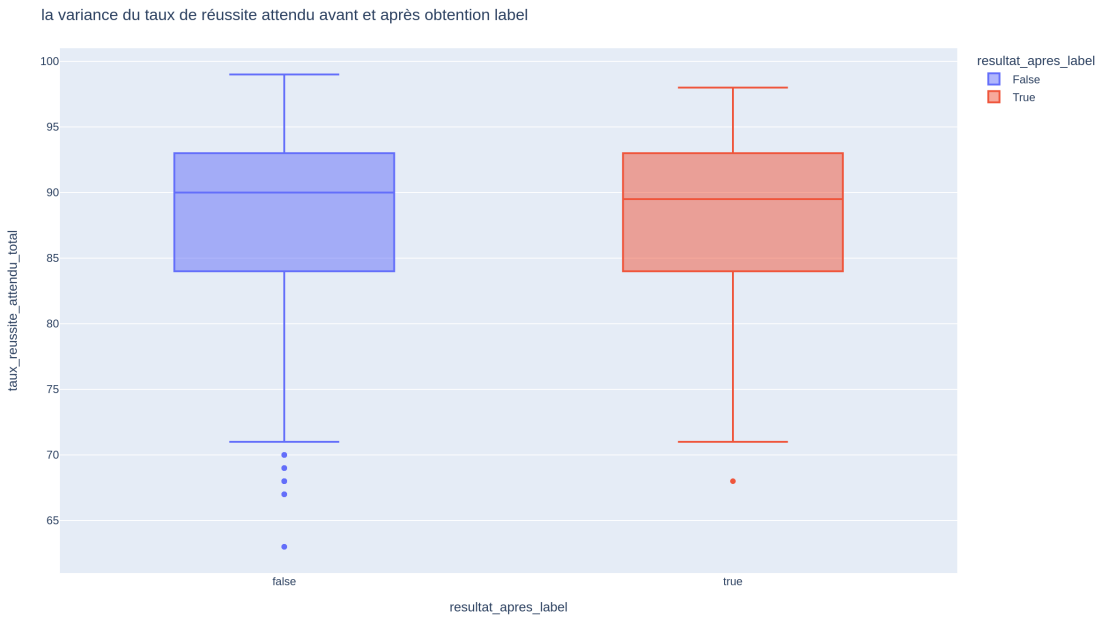
4.4.1 La variance du taux de réussite attendu avant et après obtention label

- True = après label
- False = avant label

[98]:
```
#plotting a histogram
fig = px.box(df_result, x="resultat_apres_label",␣
 ↪y="taux_reussite_attendu_total",
             color="resultat_apres_label")
fig.update_layout(title_text="la variance du taux de réussite attendu avant et␣
 ↪après obtention label")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[98]:



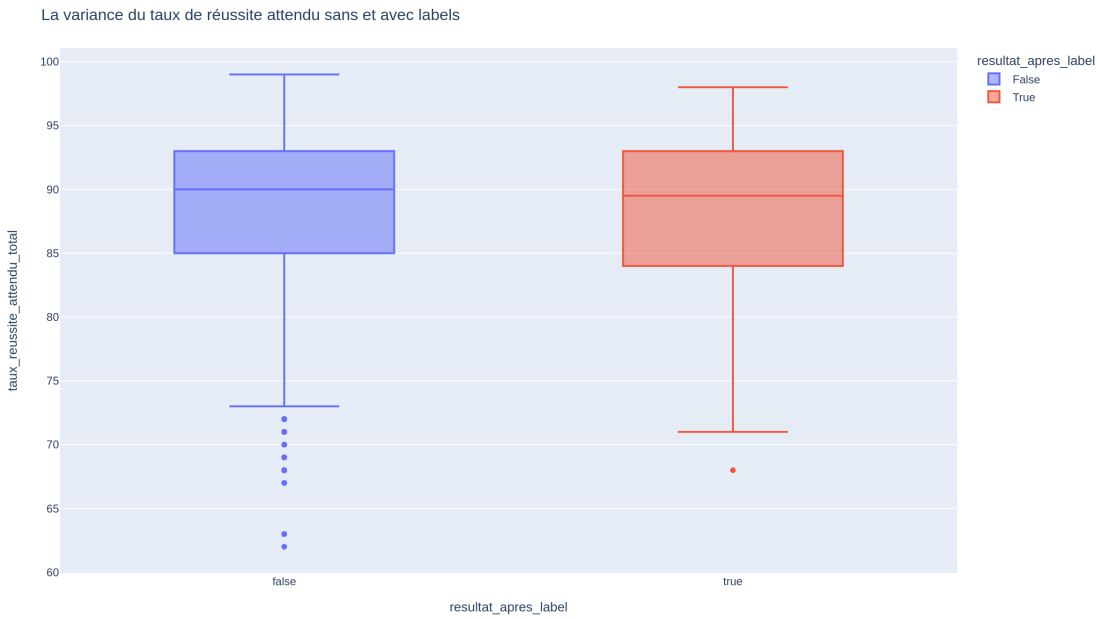
4.4.2 La variance du taux de réussite attendu sans et avec labels

- True = avec label
- False = sans label

[99]:
```
#plotting a histogram
fig = px.box(df_result_all, x="resultat_apres_label",␣
 ↪y="taux_reussite_attendu_total", color="resultat_apres_label",
                      labels={"taux_reussite_attendu_france_total_secteurs" :␣
 ↪"Taux_réussite_attendu"})
fig.update_layout(title_text="La variance du taux de réussite attendu sans et␣
 ↪avec labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[99]:



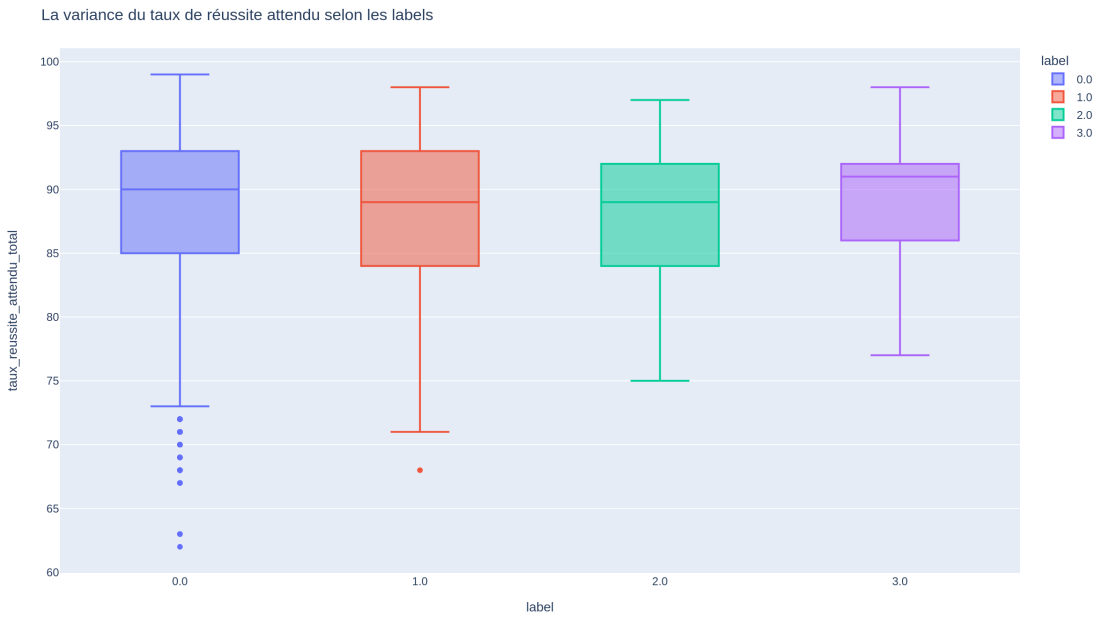
4.4.3 La variance du taux de réussite attendu selon le niveau du label

[100]:
```
# transform label to numeric
df_result_all["label"] = pd.to_numeric(df_result_all["label"])
#sort label by ascending order
df_result_all.sort_values(by="label",inplace=True)
# re transform label to str so it can be a discrete value for visualisations␣
 ↪(discrete value)
df_result_all["label"] = df_result_all["label"].apply(str)

#plotting a histogram
fig = px.box(df_result_all, x="label", y="taux_reussite_attendu_total",␣
 ↪color="label")
fig.update_layout(title_text="La variance du taux de réussite attendu selon les␣
 ↪labels")
#fig.show()

image = fig.to_image(format='png',width=1200, height=700, scale=2)
Image(image)
```

[100]:



4.5 PARTIE 4.6 : ANOVA

4.5.1 Effet de la labélisation sur le taux de réussite des lycées

Résultat de l'anova

[101]:
```
lm = sfa.ols('taux_reussite_total ~ C(label)', data=df_result_all).fit()
anova = sa.stats.anova_lm(lm)
anova
```

[101]:

	df	sum_sq	mean_sq	F	PR(>F)
C(label)	3.0	8696.448132	2898.816044	39.609113	2.017794e-25
Residual	9063.0	663280.943728	73.185584	NaN	NaN

Analyse de l'anova P_value < alpha (0,05), donc on rejette H0 et on conclut d'une manière significative un effet du label numérique sur le taux de réussite des lycées

Test de Tukey

- permet de préciser quelles modalités de la variable qualitative label a provoqué ce rejet

[102]:
```
# perform Tukey's test
tukey = pairwise_tukeyhsd(endog=df_result_all['taux_reussite_total'],
              groups=df_result_all['label'],
              alpha=0.05)

#display results
```

```
print(tukey)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
===================================================
group1 group2 meandiff p-adj   lower  upper  reject
---------------------------------------------------
   0.0    1.0   2.1142   -0.0  1.5269 2.7014   True
   0.0    2.0   2.3554    0.0   1.355 3.3558   True
   0.0    3.0   2.4795 0.0007  0.8234 4.1357   True
   1.0    2.0   0.2412 0.9421 -0.8537 1.3362  False
   1.0    3.0   0.3654 0.9473 -1.3496 2.0803  False
   2.0    3.0   0.1241 0.9983 -1.7724 2.0207  False
---------------------------------------------------
```

Analyse de test de tukey **Le test de tukey est utilisé pour comparer les moyennes de plusieurs groupes. Le test est utilisé pour déterminer s'il existe des différences significatives entre les moyennes de différents groupes. L'output montre les résultats du test pour chaque comparaison à paires de groupes. Les colonnes de l'output incluent:**

- group1 et group2: les groupes étant comparés
- meandiff: la différence des moyennes entre les deux groupes
- p-adj: la valeur p ajustée pour la comparaison
- lower et upper: les limites inférieure et supérieure de l'intervalle de confiance de 95% pour la différence des moyennes
- reject: si oui ou non l'hypothèse nulle (que les moyennes sont égales) peut être rejetée en fonction de la valeur p et du niveau alpha choisi (0,05 dans ce cas)

Ce test compare les moyennes de quatre groupes: 0.0, 1.0, 2.0 et 3.0.

- Pour le groupe 0.0 et 1.0, la valeur p-adj est de 0.0, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05.
- Pour le groupe 0.0 et 2.0, la valeur p-adj est de 0.0, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05.
- Pour le groupe 1.0 et 3.0, la valeur p-adj est de 0.0, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05

Shapiro test : Tester l'hypothèse de normalité

- H0 : Les échatillons sont gaussiens

[103]:
```
# Split the data
x = df_result_all.groupby('label')['taux_reussite_total'].apply(list)
```

[104]:
```
# Perform Shapiro-Wilk test
stat, p = stats.shapiro(x[0])
print("Shapiro-Wilk test: statistic=%f, p-value=%f" % (stat, p))
if p > 0.05:
  print("The data is likely normal")
else:
```

```
  print("The data is likely not normal")
```

```
Shapiro-Wilk test: statistic=0.937840, p-value=0.000000
The data is likely not normal
```

```
/home/ubuntu/anaconda3/lib/python3.9/site-
packages/scipy/stats/morestats.py:1760: UserWarning:

p-value may not be accurate for N > 5000.
```

p value < 0.05 donc on rejette H0 et on conclut que les échantillons ne sont plutot pas gaussiens

Levene's test : Tester l'hypothèse d'homoscédasticité

- H0 : Les variances sont égales

L'hyphothèse de normalité n'est pas validé, donc on réalise un test de Levene pour tester l'hypothèse d'homoscédasticité

[105]:
```
# Perform Levene's test
stat, p = stats.levene(x[0], x[1], x[2], x[3])
print("Levene's test: statistic=%.3f, p-value=%.3f" % (stat, p))
if p > 0.05:
  print("The variances of the samples are likely similar")
else:
  print("The variances of the samples are likely different")
```

```
Levene's test: statistic=5.533, p-value=0.001
The variances of the samples are likely different
```

p value < 0,05 donc on rejette H0 et on conclut que les variances des labels ne sont plutot pas égales

4.5.2 Effet de la labélisation sur la valeur ajoutée des lycées

Résultat de l'anova

[106]:
```
lm = sfa.ols('valeur_ajoute_totale ~ C(label)', data=df_result_all).fit()
anova = sa.stats.anova_lm(lm)
anova
```

[106]:

	df	sum_sq	mean_sq	F	PR(>F)
C(label)	3.0	250.824958	83.608319	2.965451	0.030755
Residual	9063.0	255523.445474	28.194135	NaN	NaN

Analyse de l'anova P_value < alpha (0,05), donc on rejette H0 et on conclut d'une manière significative un effet du label numérique sur la valeur ajoutée de réussite des lycées

Test de Tukey

[107]:
```
# perform Tukey's test
tukey = pairwise_tukeyhsd(endog=df_result_all['valeur_ajoute_totale'],
              groups=df_result_all['label'],
              alpha=0.05)

#display results
print(tukey)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====================================================
group1 group2 meandiff p-adj   lower   upper  reject
-----------------------------------------------------
   0.0    1.0   0.1134 0.8547 -0.2511  0.4779  False
   0.0    2.0   0.0378 0.9986 -0.5831  0.6587  False
   0.0    3.0  -1.1205 0.0263 -2.1484 -0.0925   True
   1.0    2.0  -0.0756 0.9919 -0.7552   0.604  False
   1.0    3.0  -1.2339 0.0154 -2.2983 -0.1694   True
   2.0    3.0  -1.1583 0.0557 -2.3354  0.0189  False
-----------------------------------------------------
```

Analyse de test de tukey Ce test compare les moyennes de quatre groupes: 0.0, 1.0, 2.0 et 3.0.

- Pour le groupe 0.0 et 3.0, la valeur p-adj est de 0.02, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05.
- Pour le groupe 1.0 et 3.0, la valeur p-adj est de 0.01, ce qui indique que la différence de moyenne est statistiquement significative au niveau de 0.05

Shapiro test : Tester l'hypothèse de normalité

- H0 : Les échatillons sont gaussiens

[108]:
```
# Split the data
x = df_result_all.groupby('label')['valeur_ajoute_totale'].apply(list)
```

[109]:
```
# Perform Shapiro-Wilk test
stat, p = stats.shapiro(x[0])
print("Shapiro-Wilk test: statistic=%f, p-value=%f" % (stat, p))
if p > 0.05:
  print("The data is likely normal")
else:
  print("The data is likely not normal")
```

```
Shapiro-Wilk test: statistic=0.972493, p-value=0.000000
The data is likely not normal
```

```
/home/ubuntu/anaconda3/lib/python3.9/site-
packages/scipy/stats/morestats.py:1760: UserWarning:
```

```
p-value may not be accurate for N > 5000.
```

Levene's test : Tester l'hypothèse d'homoscédasticité

- H0 : Les variances sont égales

[110]:
```python
# Perform Levene's test
stat, p = stats.levene(x[0], x[1], x[2], x[3])
print("Levene's test: statistic=%.3f, p-value=%.3f" % (stat, p))
if p > 0.05:
  print("The variances of the samples are likely similar")
else:
  print("The variances of the samples are likely different")
```

```
Levene's test: statistic=32.003, p-value=0.000
The variances of the samples are likely different
```

[ ]:
```python

```