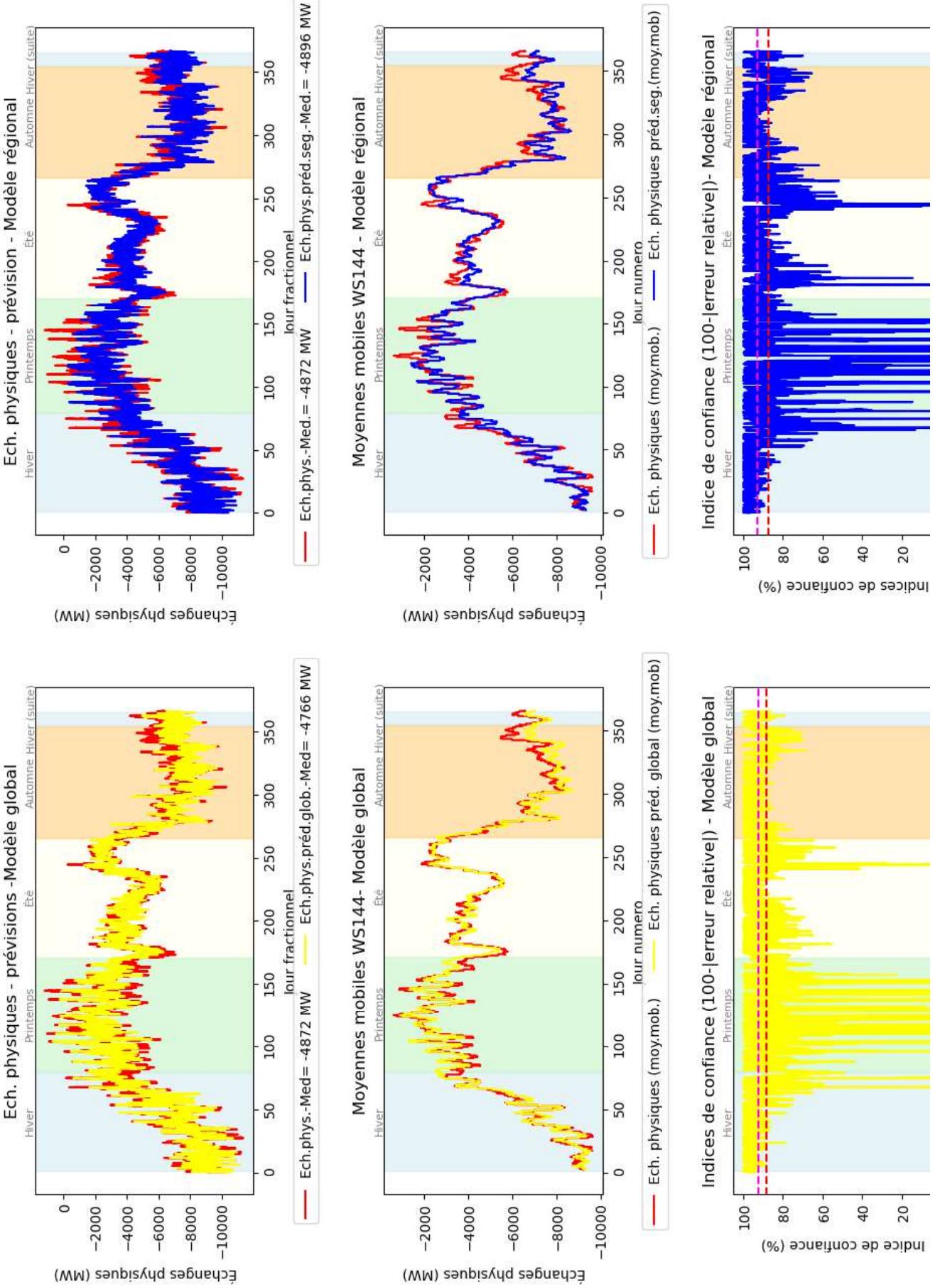
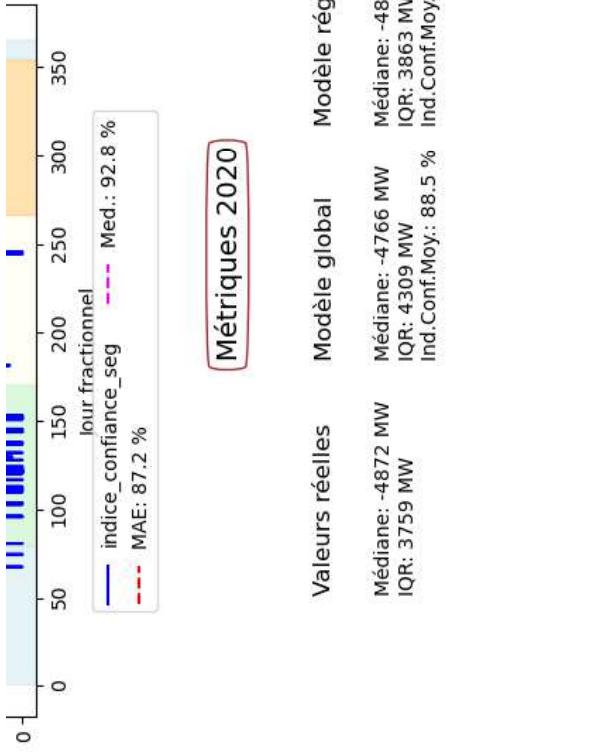
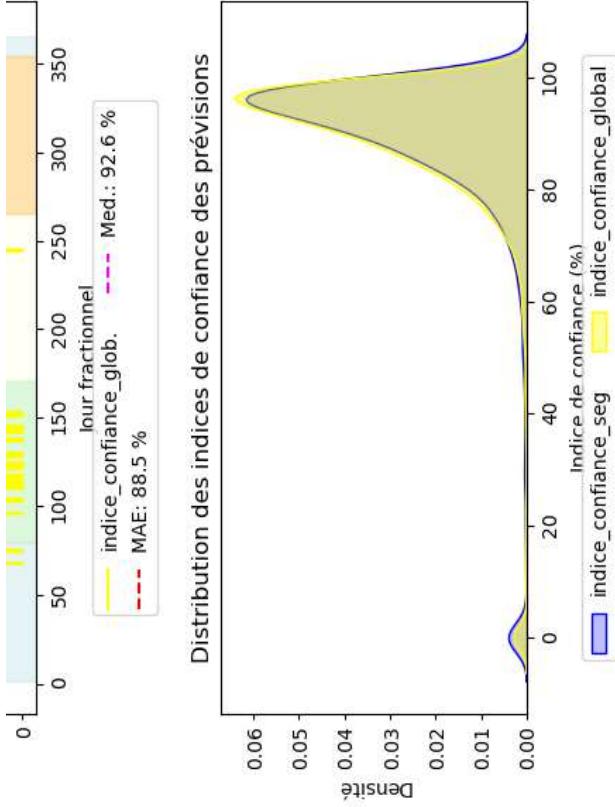


XGBoost\*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

# Prévisions des échanges physiques par XGBoost sans transformation cible

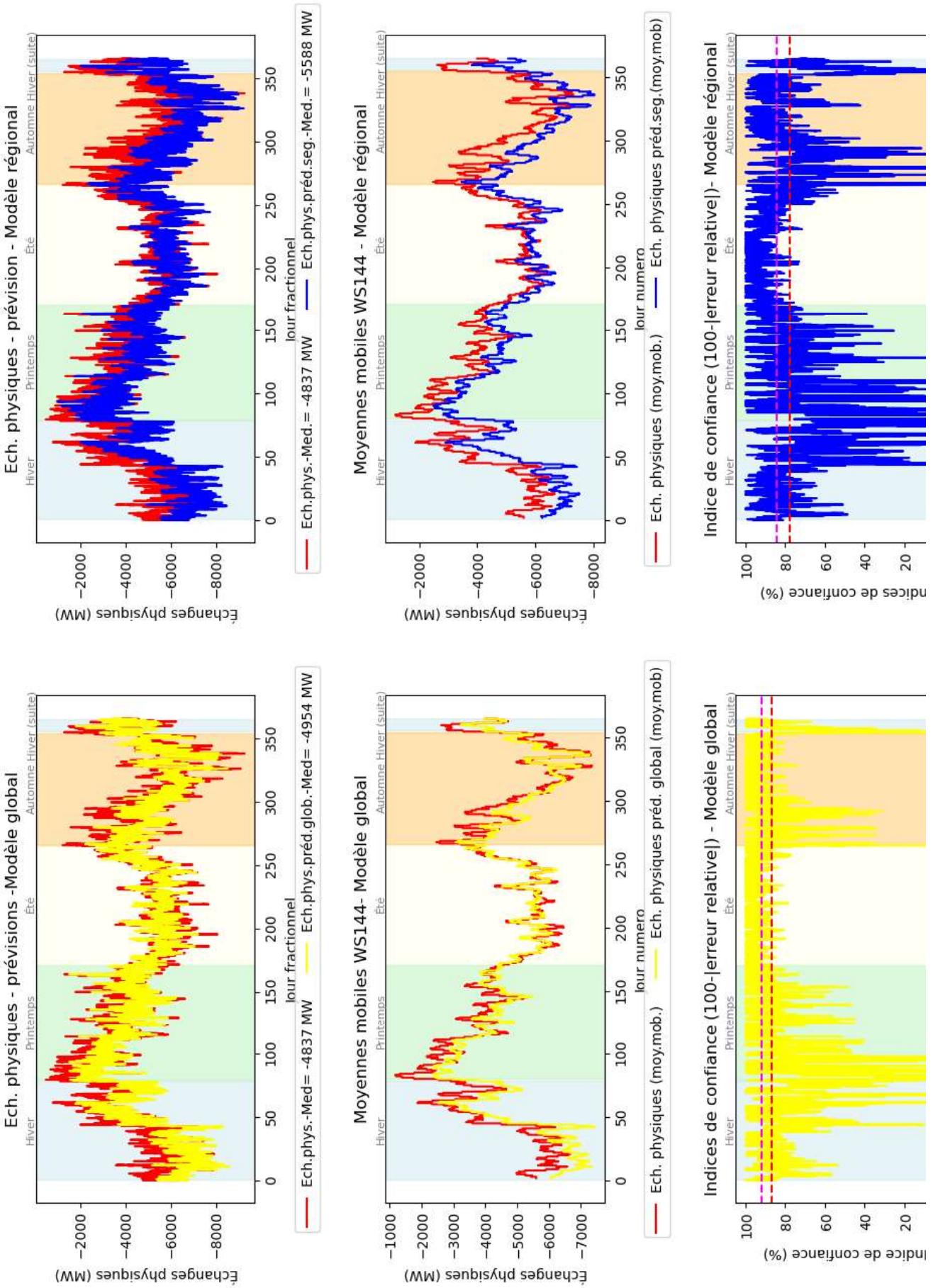
## GRAND EST, 2020

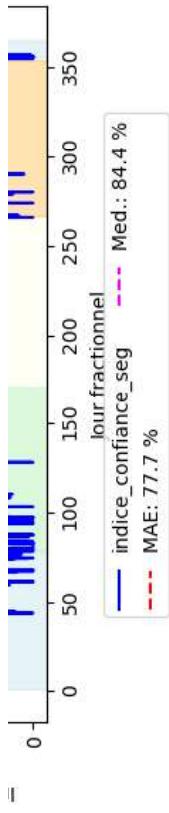
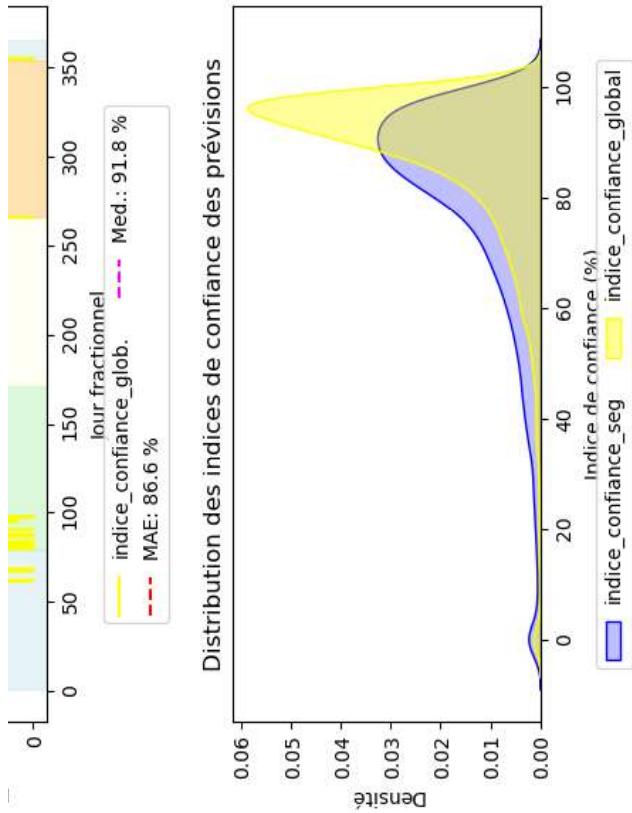




XGBoost \*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

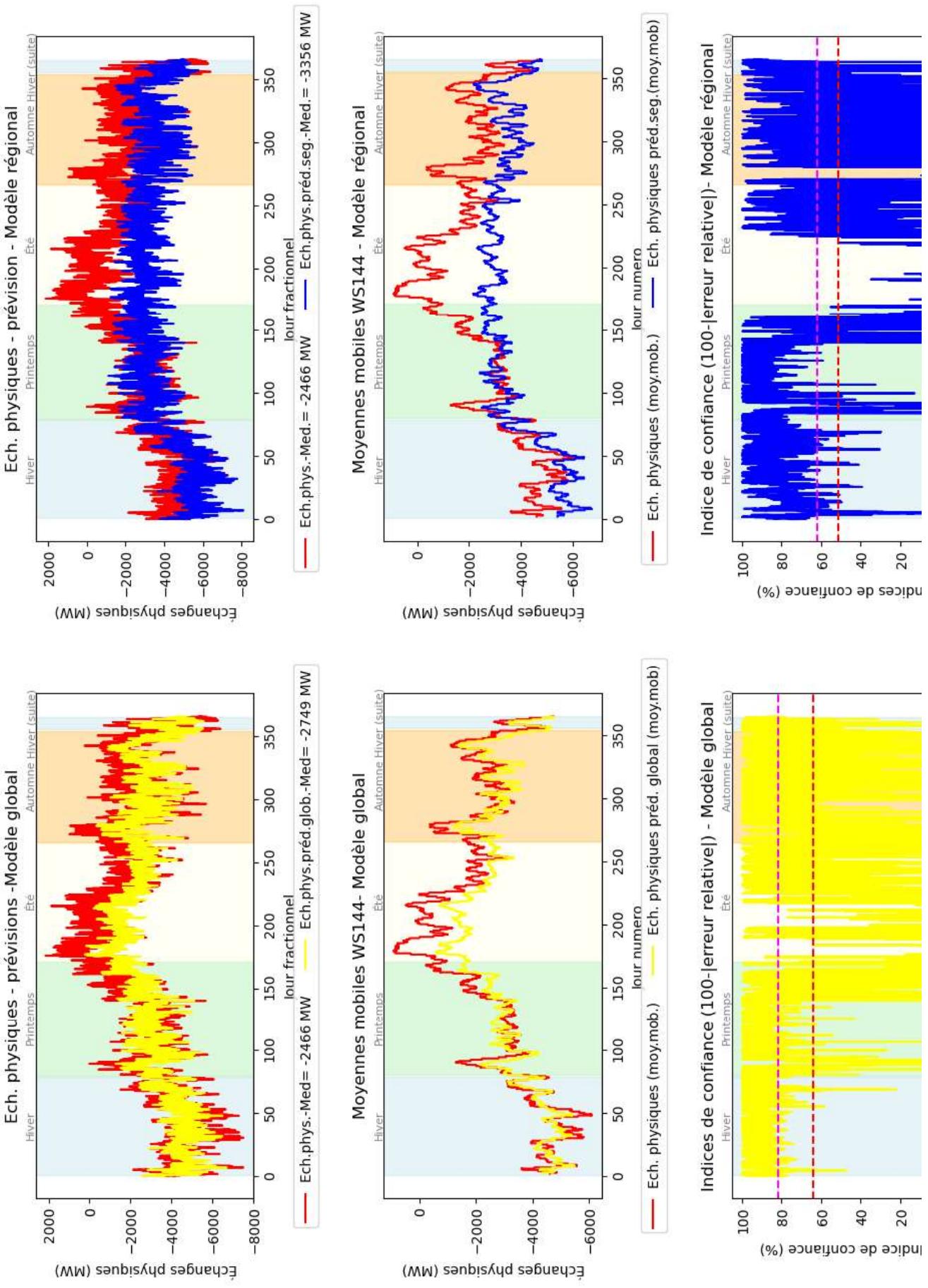
Prévisions des échanges physiques par XGBoost sans transformation cible  
GRAND EST, 2021

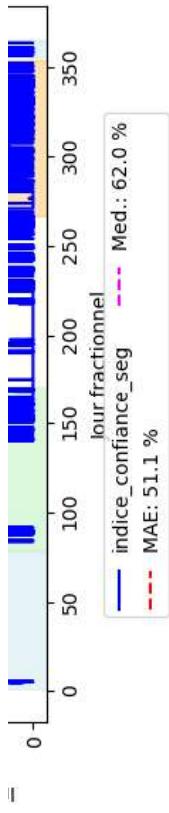
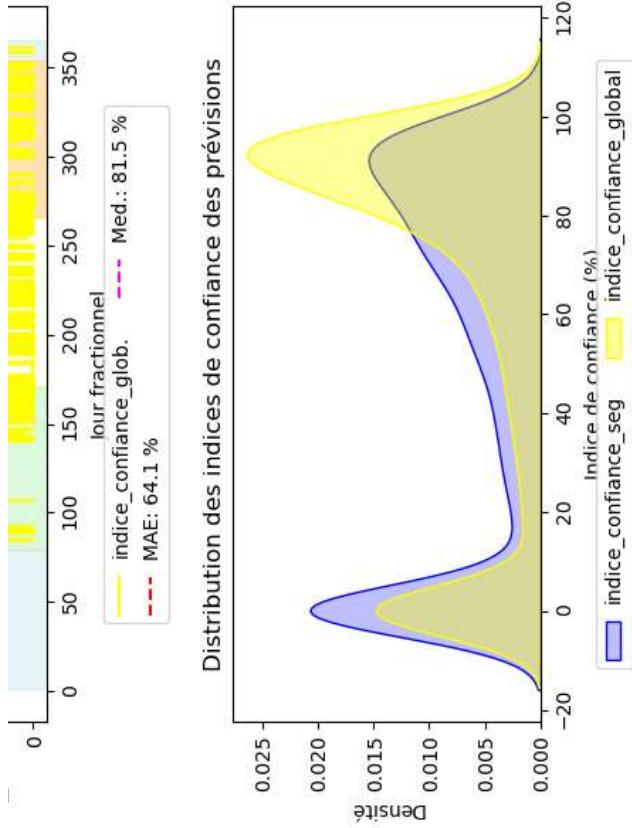




XGBoost\*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

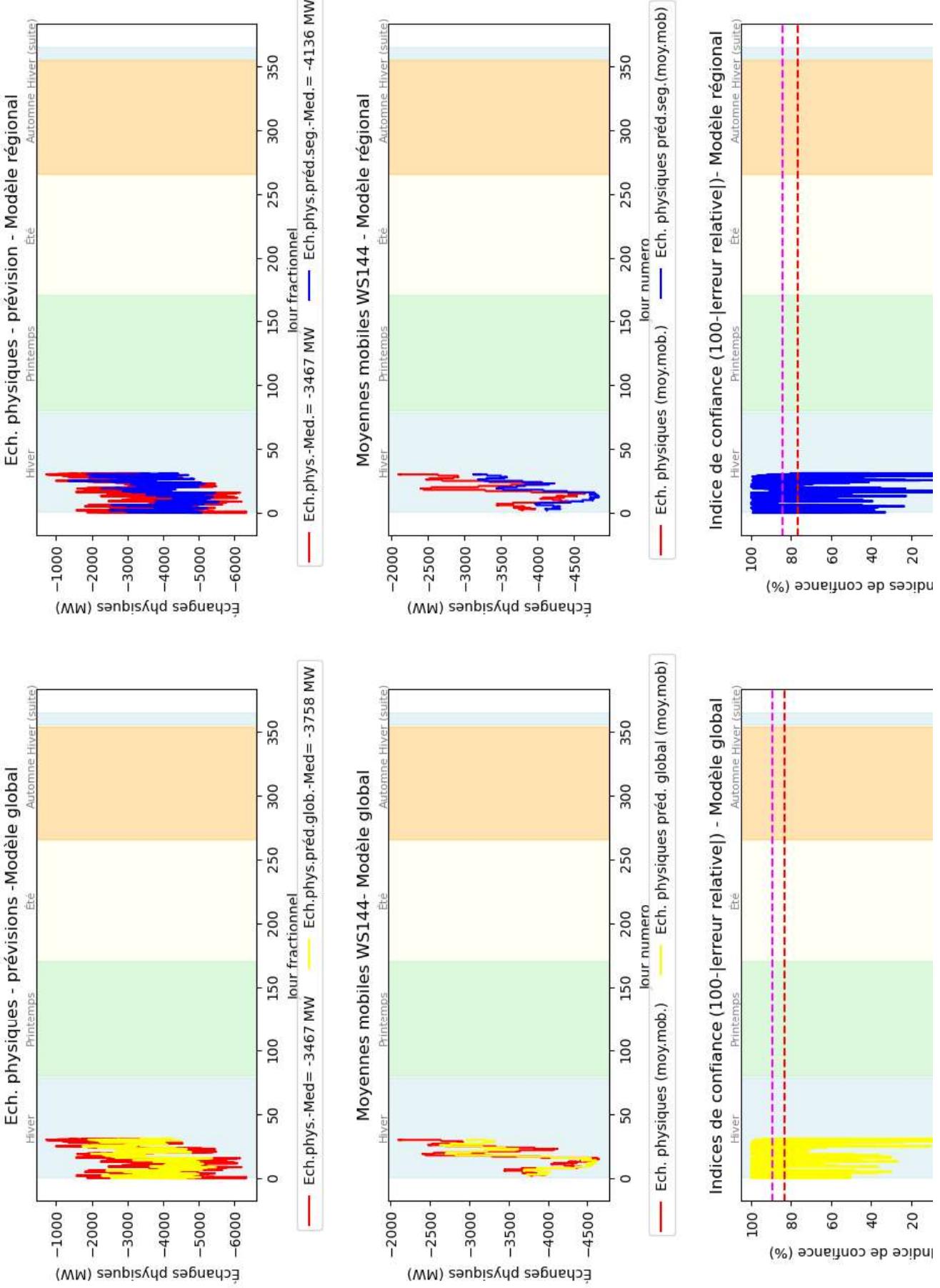
Prévisions des échanges physiques par XGBoost sans transformation cible  
GRAND EST, 2022

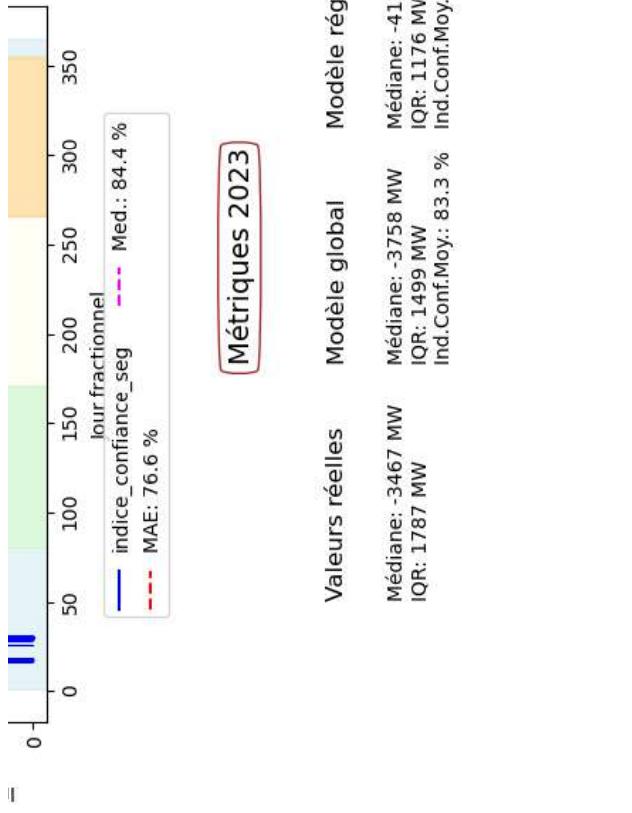
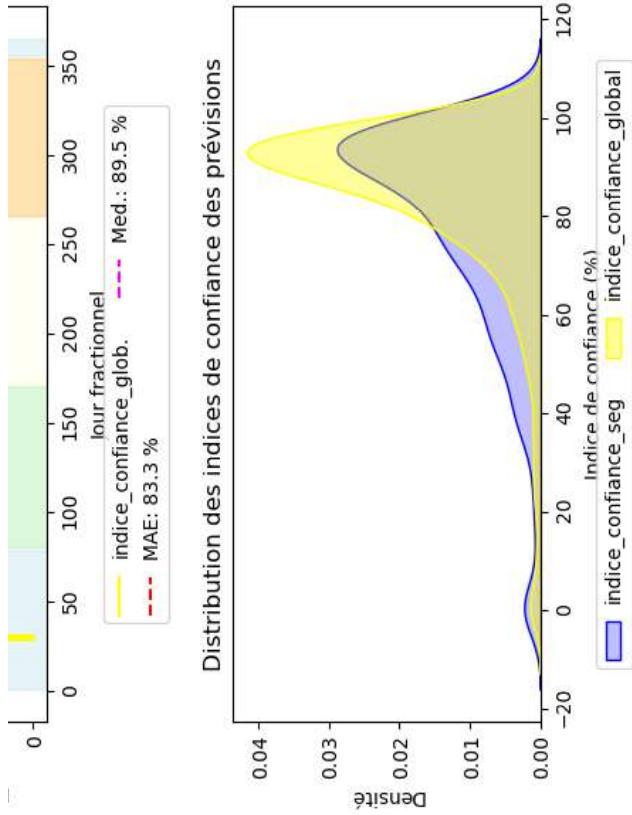




XGBoost\*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

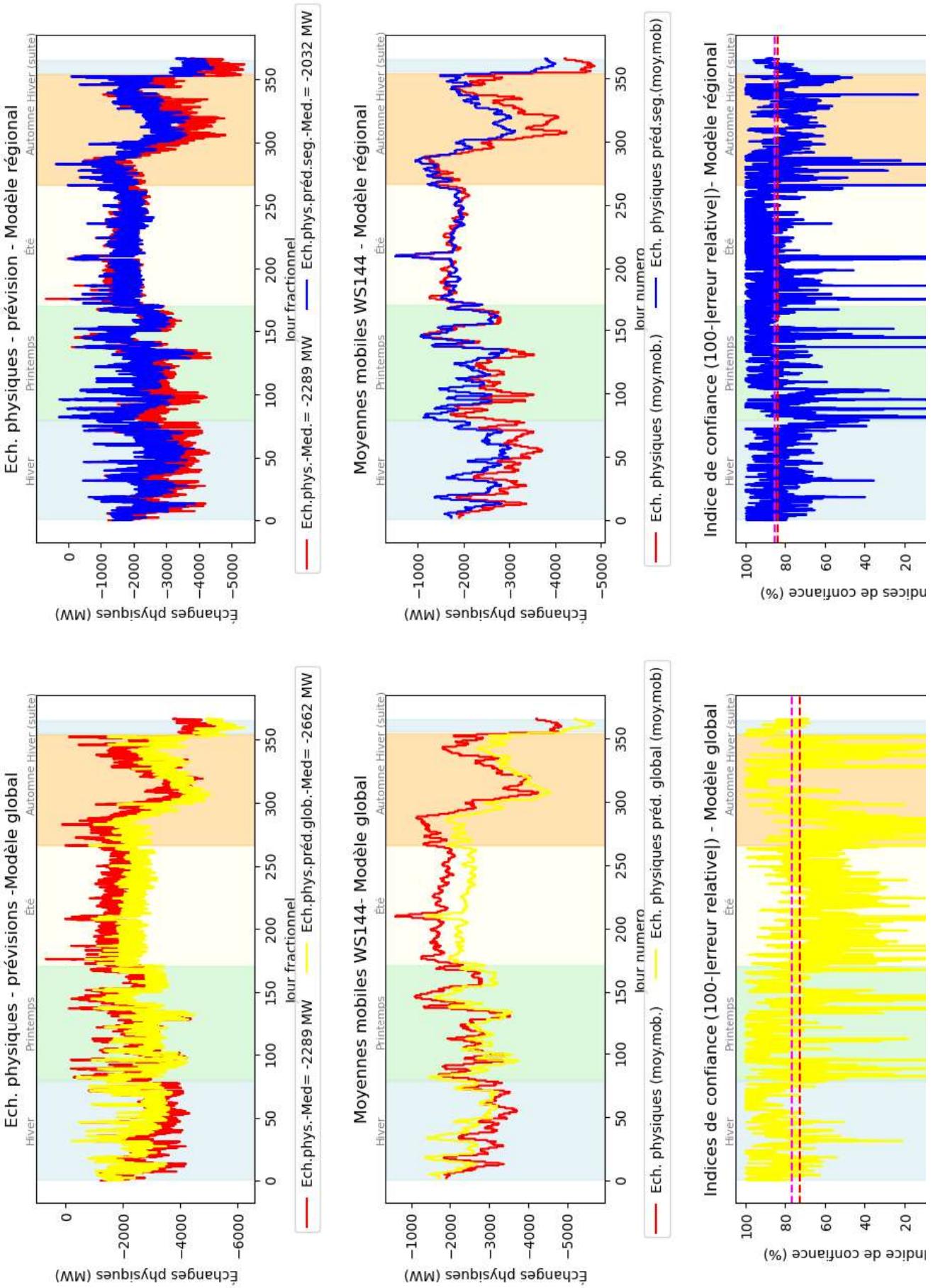
# Prévisions des échanges physiques par XGBoost sans transformation cible GRAND EST, 2023

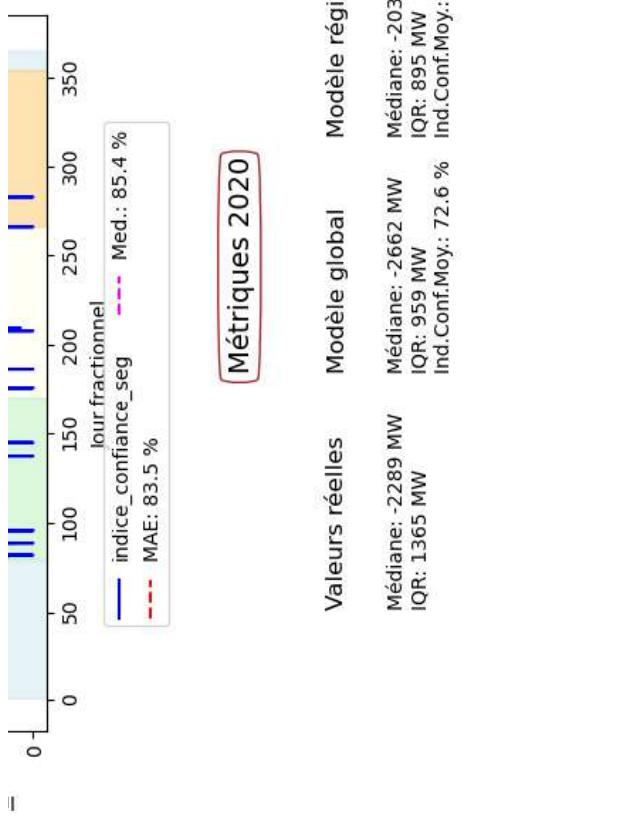
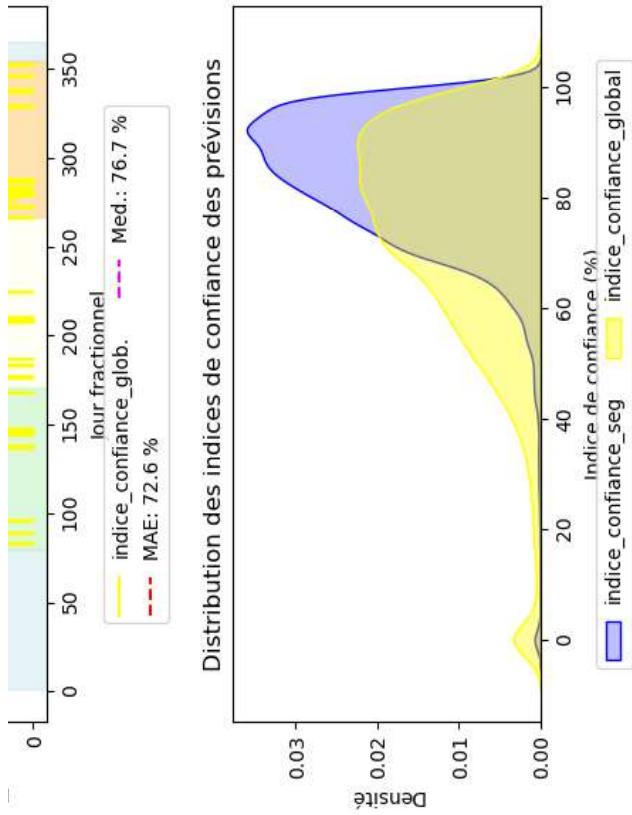




XGBoost \*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

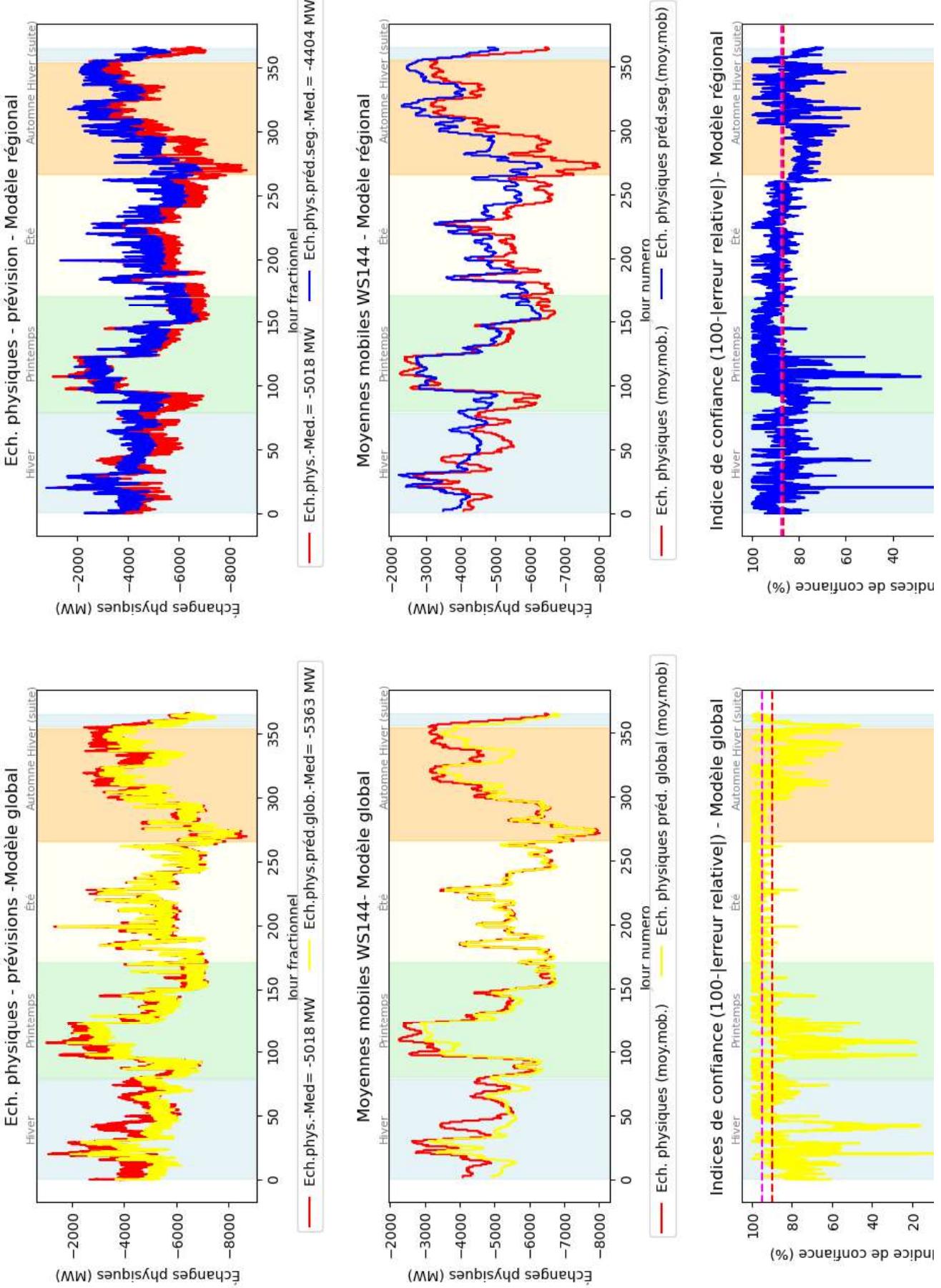
Prévisions des échanges physiques par XGBoost sans transformation cible  
NORMANDIE, 2020

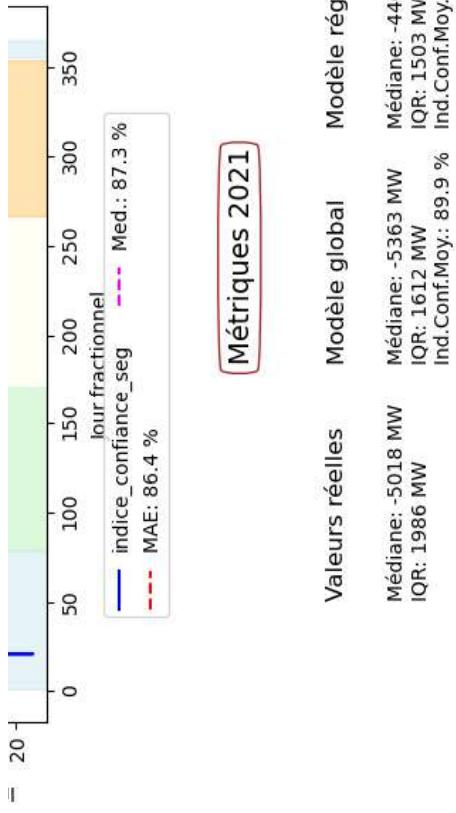
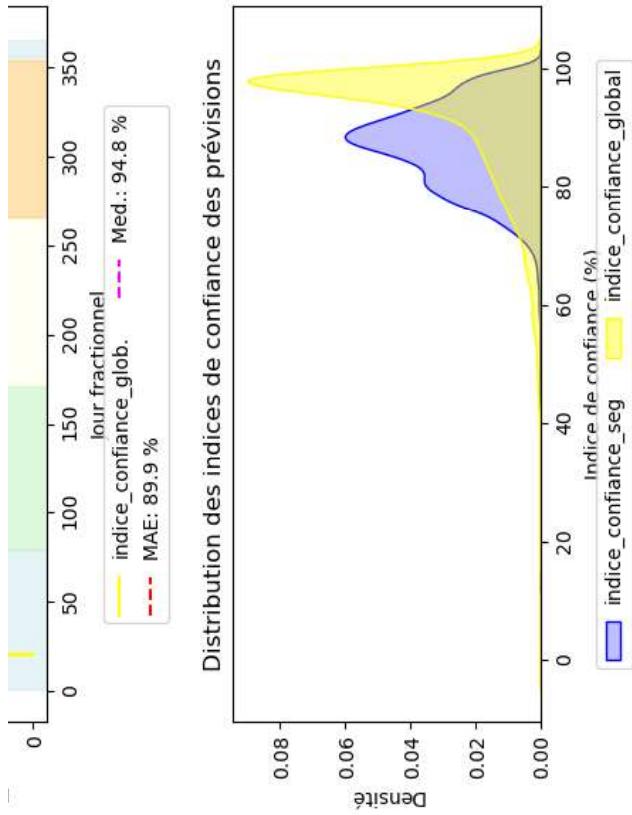




XGBoost \*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

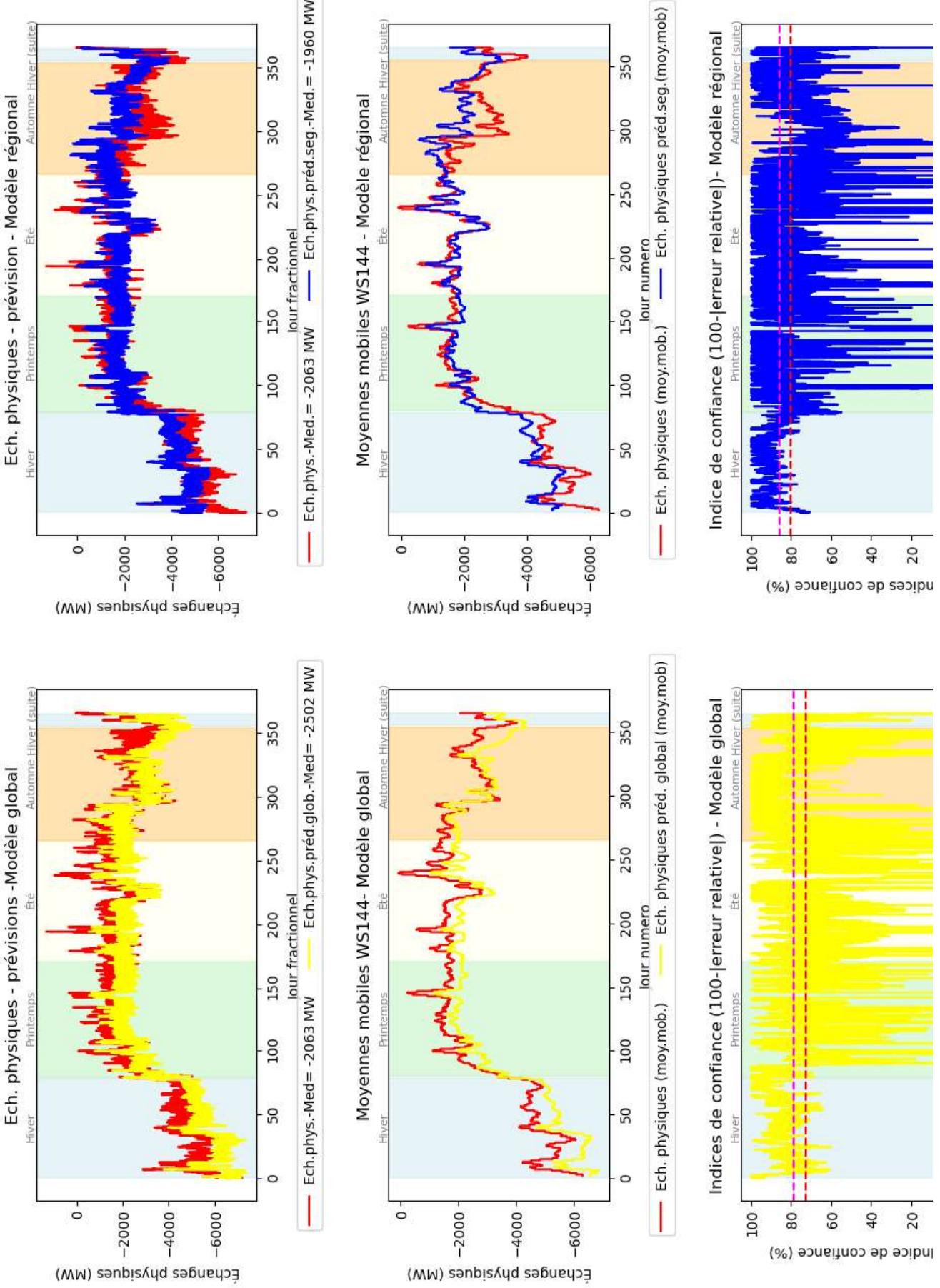
## Prévisions des échanges physiques par XGBoost sans transformation cible NORMANDIE, 2021

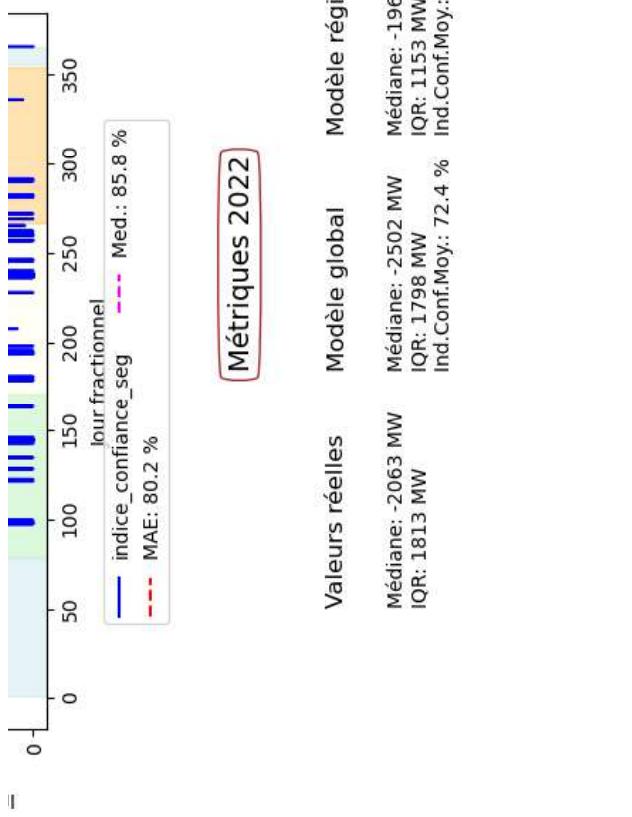
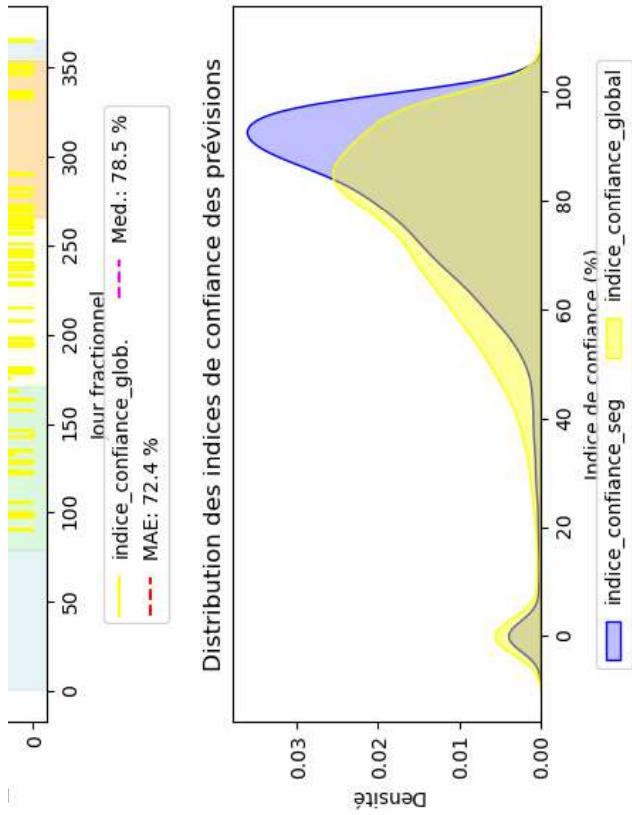




XGBoost\*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

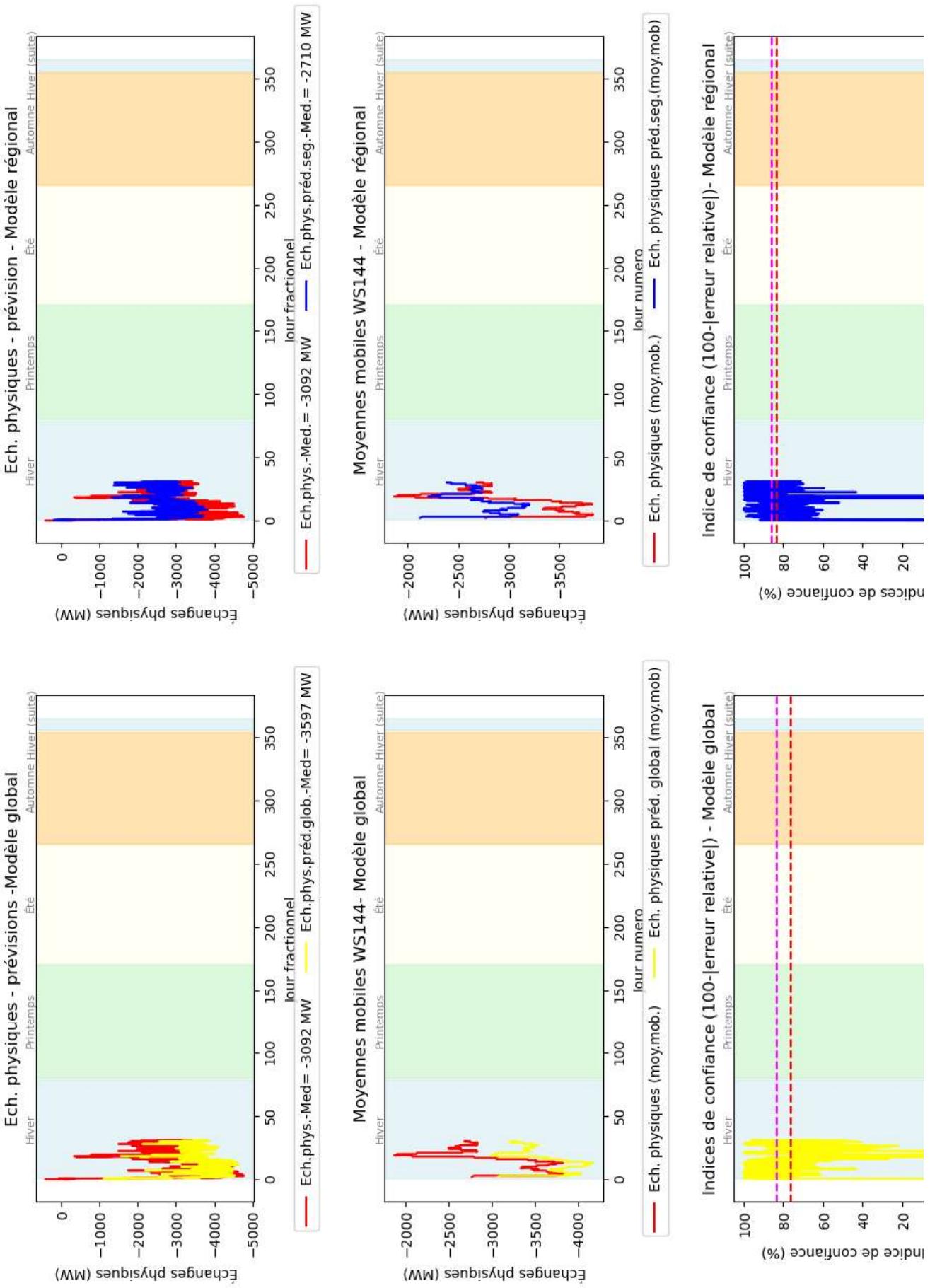
## Prévisions des échanges physiques par XGBoost sans transformation cible NORMANDIE, 2022

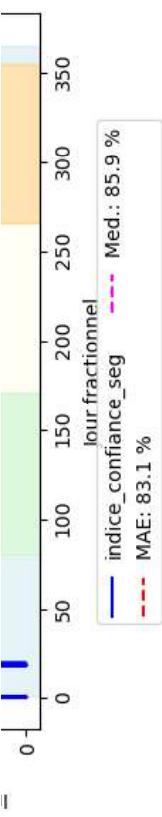
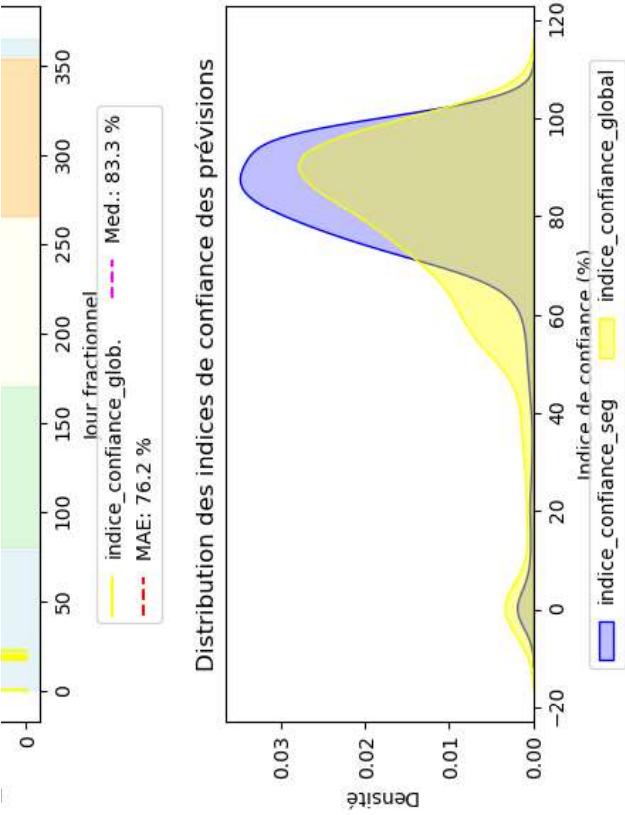




XGBoost \*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

Prévisions des échanges physiques par XGBoost sans transformation cible  
NORMANDIE, 2023



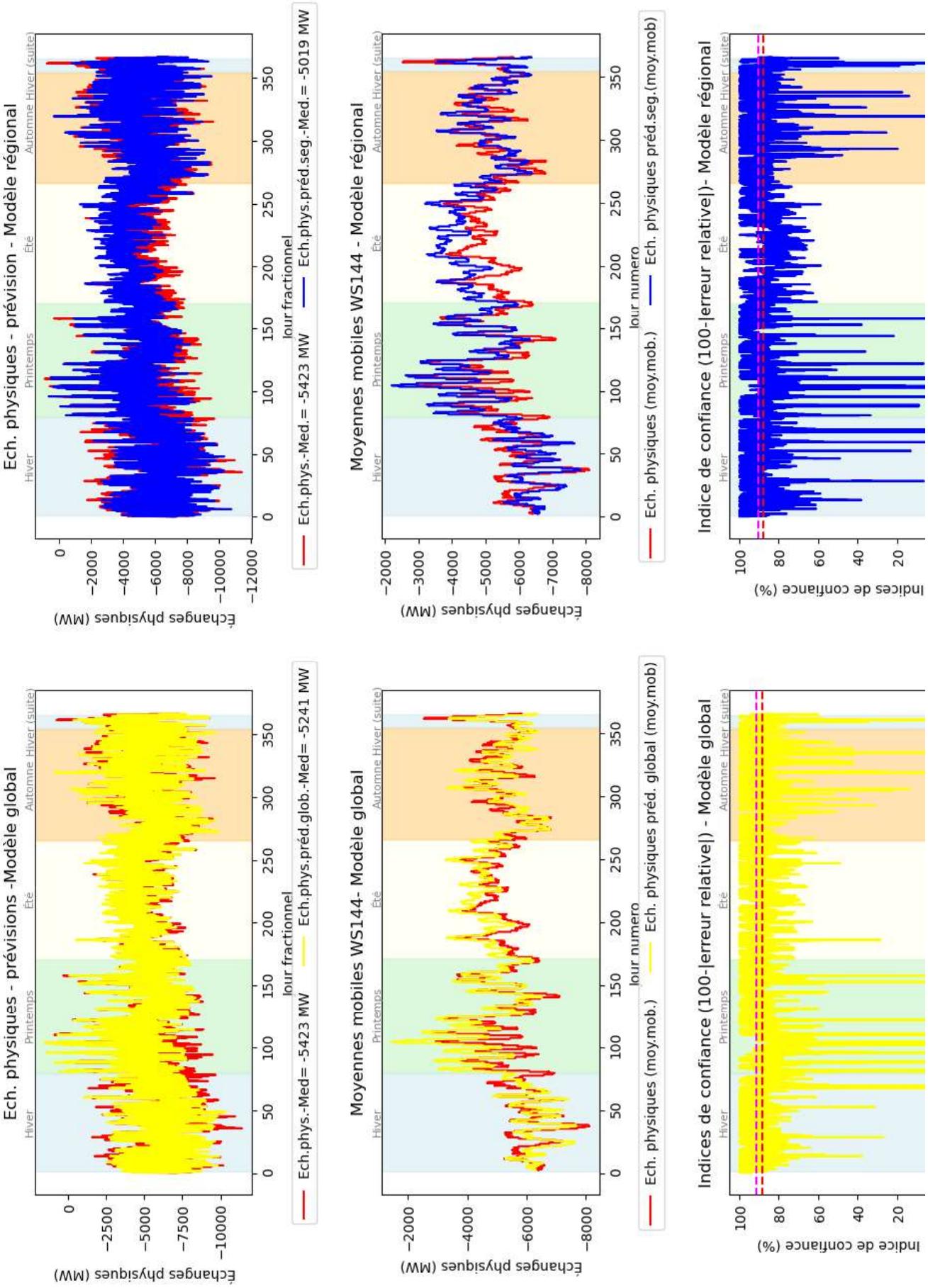


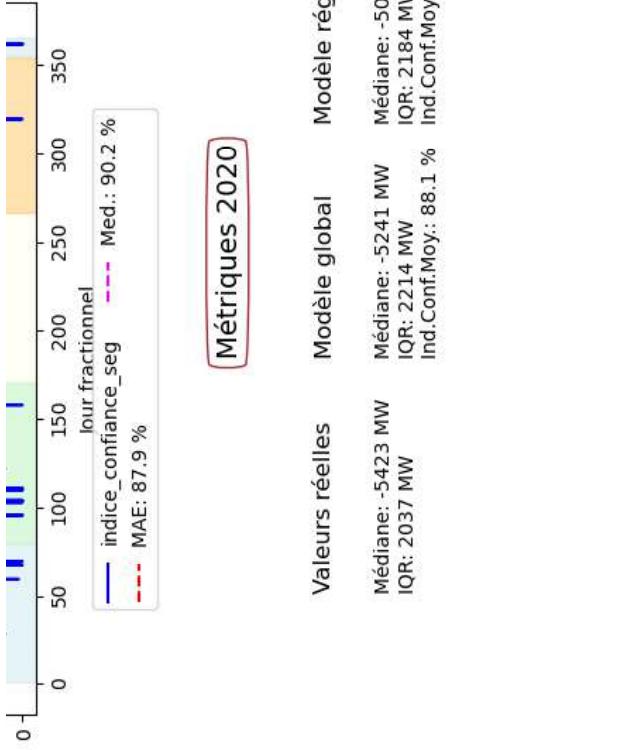
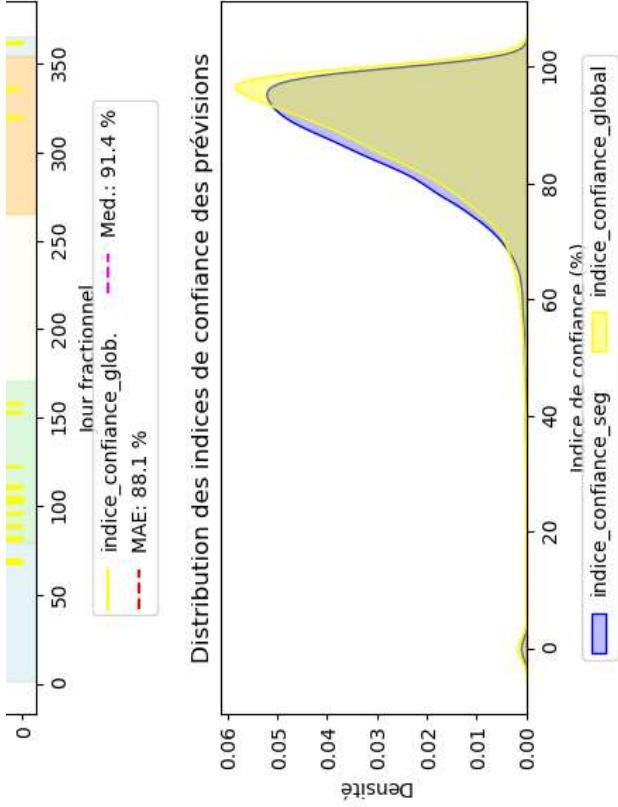
	Valeurs réelles	Modèle global	Modèle régional
	Médiane: -3092 MW IQR: 1061 MW	Médiane: -3597 MW IQR: 700 MW	Médiane: -2710 MW IQR: 636 MW

XGBoost\*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

# Prévisions des échanges physiques par XGBoost sans transformation cible

## AUVERGNE RHONE ALPES, 2020

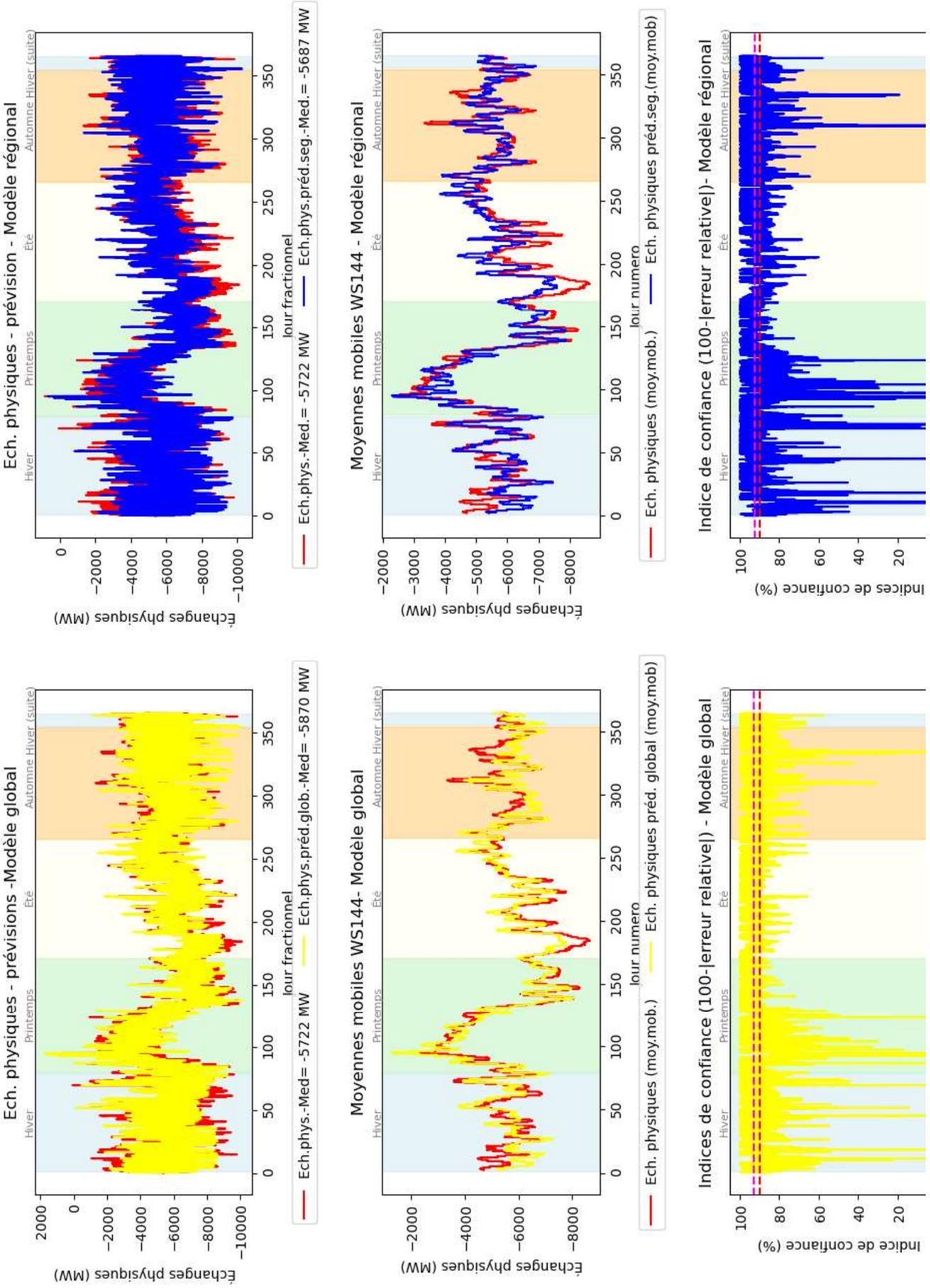


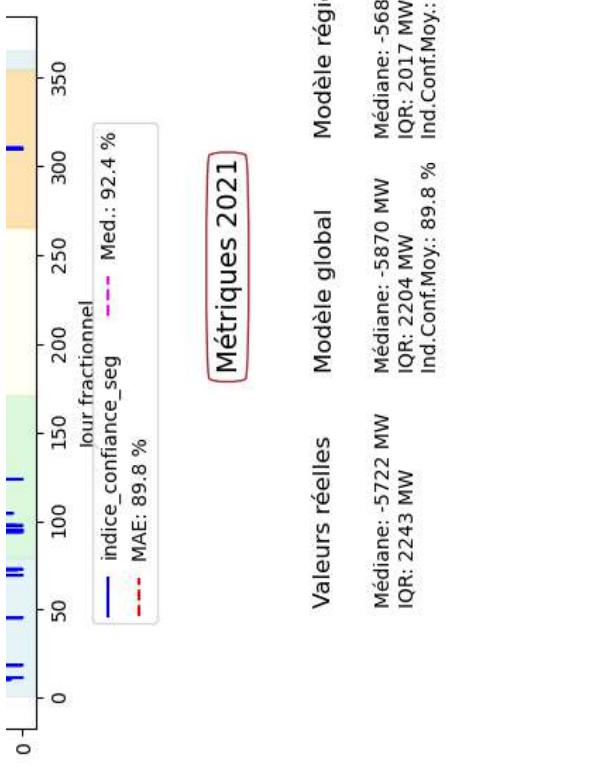
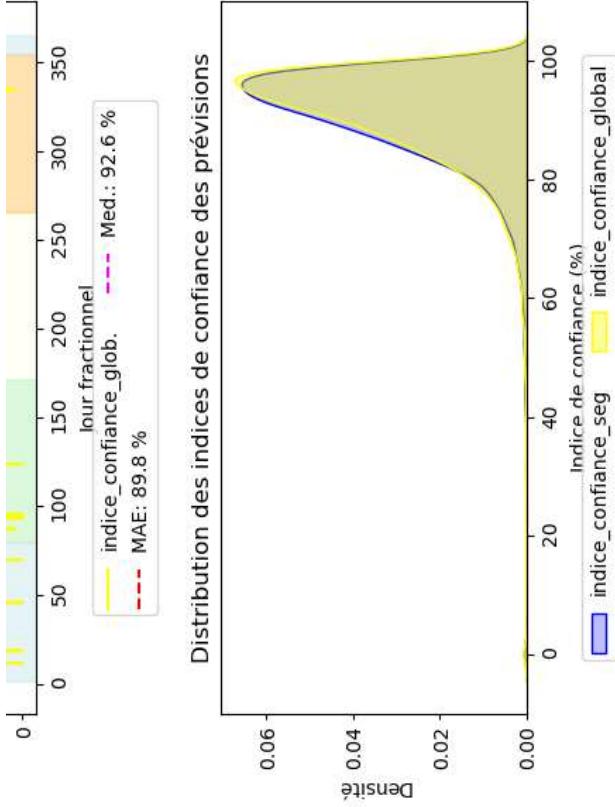


XGBoost \*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

# Prévisions des échanges physiques par XGBoost sans transformation cible

## AUVERGNE RHONE ALPES, 2021

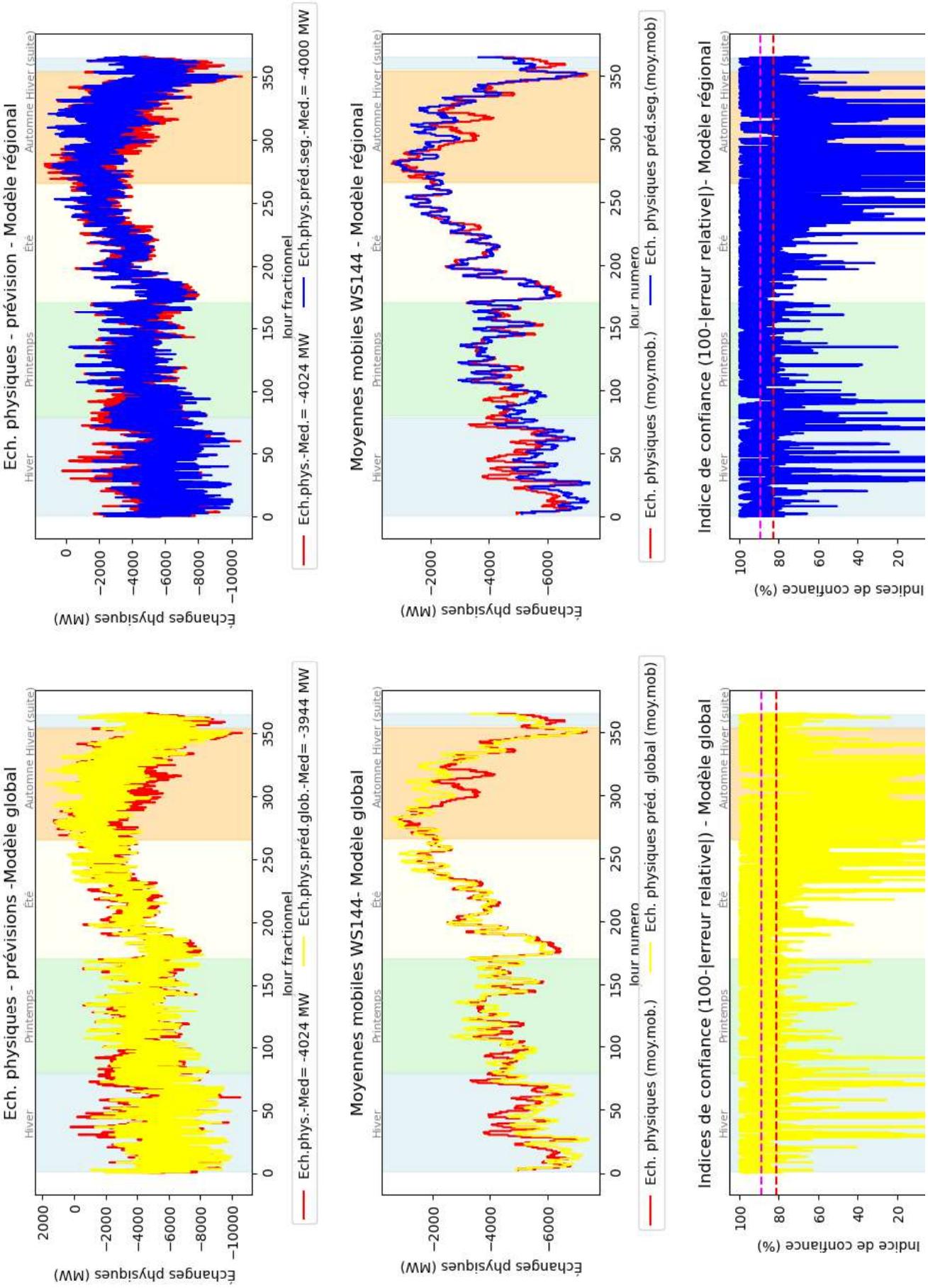


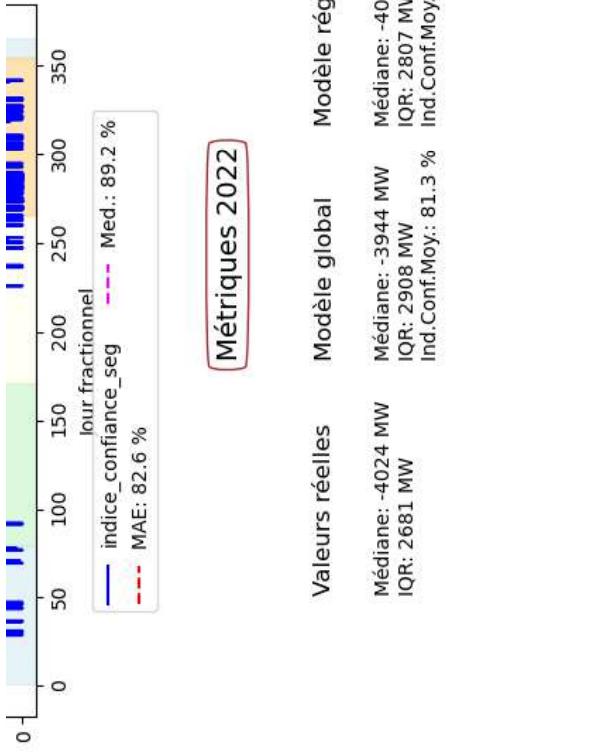
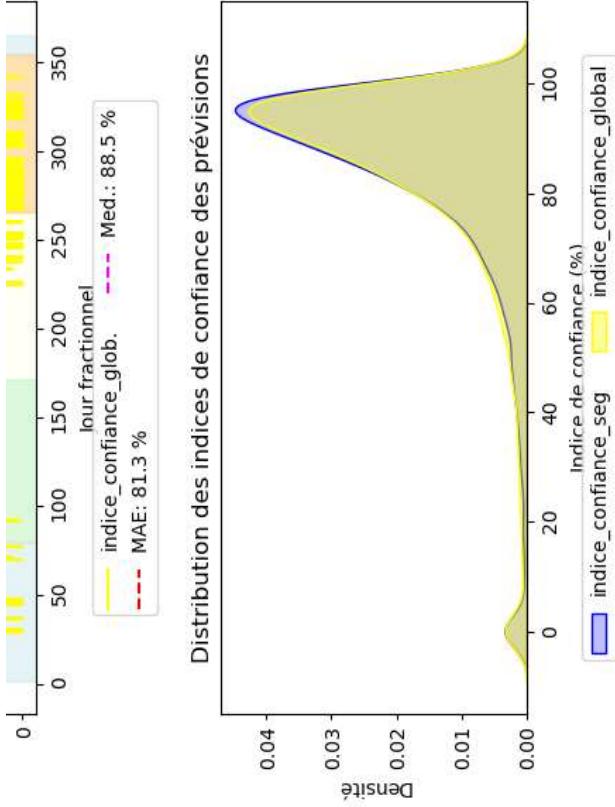


XGBoost \*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

## Prévisions des échanges physiques par XGBoost sans transformation cible

### AUVERGNE RHONE ALPES, 2022

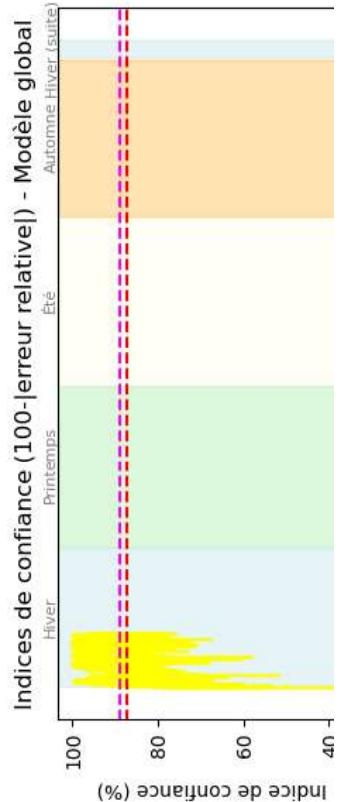
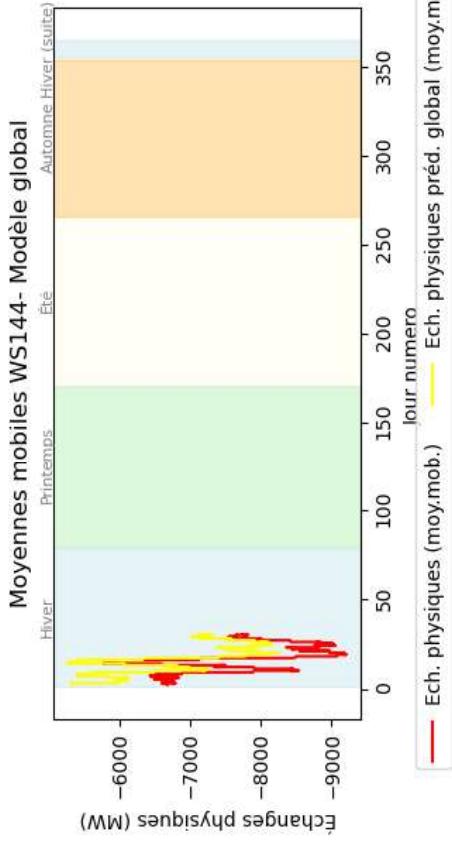
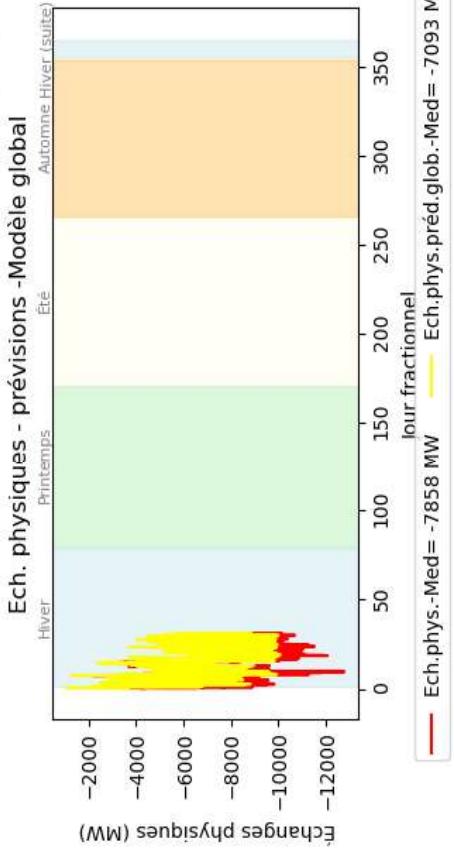




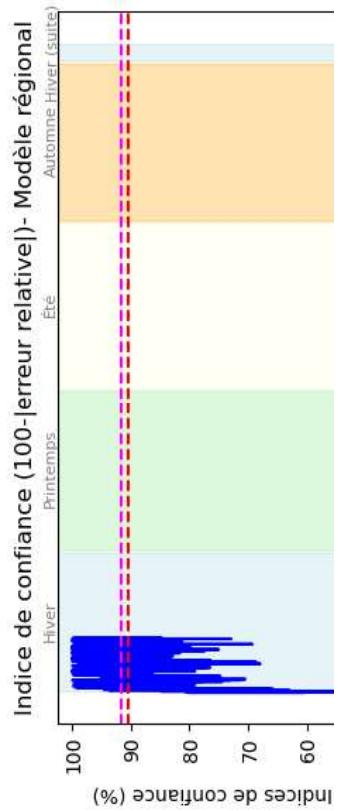
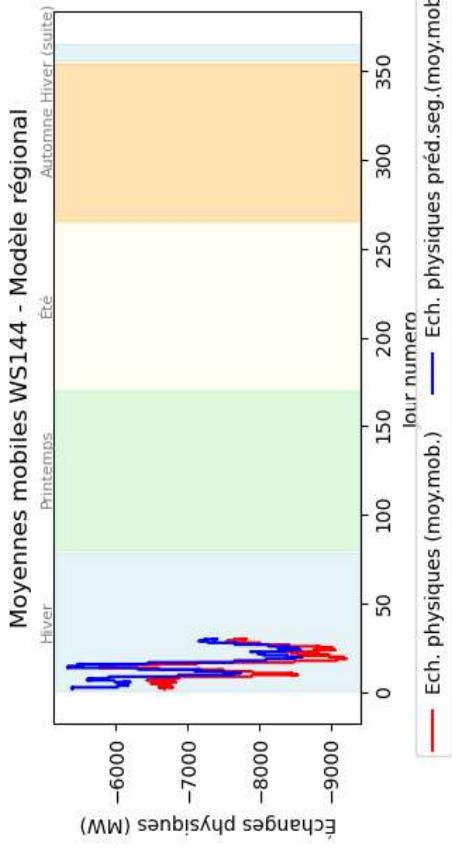
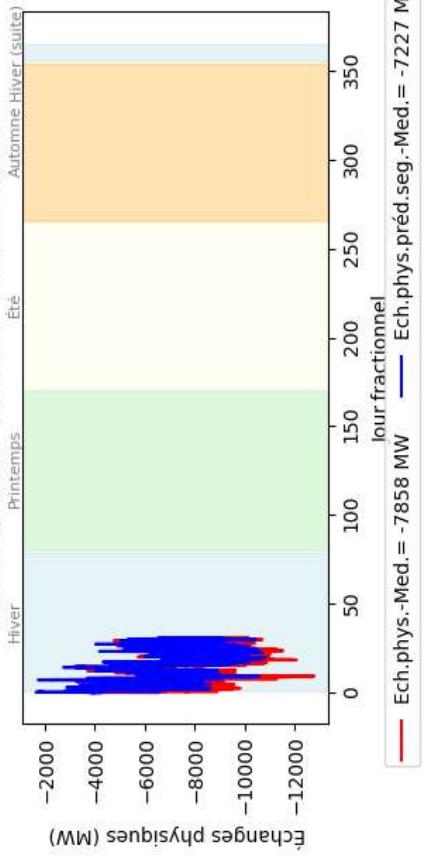
XGBoost \*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

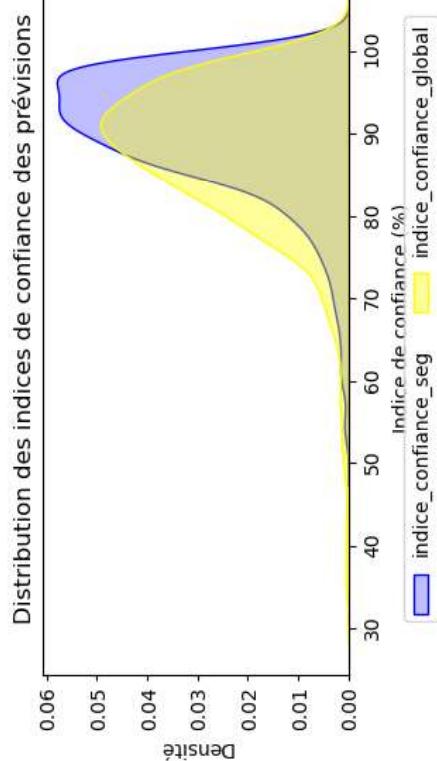
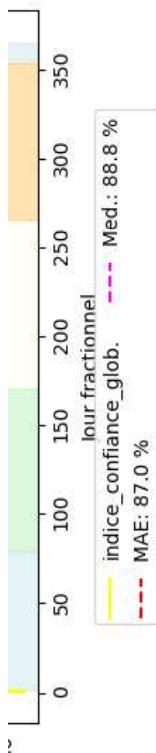
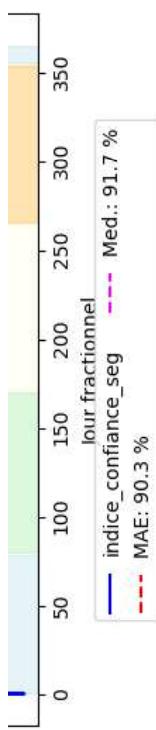
Prévisions des échanges physiques par XGBoost sans transformation cible  
AUVERGNE RHÔNE ALPES, 2023

AUVERGNE RHÔNE ALPES, 2023



Ech. physiques - prévisions - Modèle régional





XGBoost \*subsample: 0.7, n\_estimators: 400, max\_depth: 6, learning\_rate: 0.1, 'gamma': 0.2, 'colsample\_bytree': 0.7

In [36]:

Out[36] :	region	annee	MSE	RMSE	MAE	R <sup>2</sup>	mediane_ech_phys	irq_ech_phys	mediane_ech_phys_glob	irq_ech_phys_glob	...	mediane_ech_phys_seg	irq_ech
<b>0</b>	CENTRE VAL DE LOIRE	2020	235548.1481	485.333028	357.112431	0.921531	-6473.0	2939.25	-6118.0465	2557.337275	...	-6185.82780	24
<b>1</b>	CENTRE VAL DE LOIRE	2021	155054.4326	393.769517	313.251704	0.880647	-6253.0	1622.00	-6079.3962	1655.602800	...	-6038.79935	15
<b>2</b>	CENTRE VAL DE LOIRE	2022	169071.0998	411.182563	309.426938	0.930583	-5357.5	2565.50	-5060.0875	2040.789475	...	-5336.27850	21
<b>3</b>	CENTRE VAL DE LOIRE	2023	538332.2605	733.711292	670.038008	0.559392	-8440.0	1368.75	-7690.5765	939.315250	...	-7809.533385	9
<b>4</b>	PAYS DE LA LOIRE	2020	179310.6228	423.450851	341.935977	0.434895	2187.0	787.25	2297.2935	745.911950	...	2397.74500	6

5 rows × 27 columns

In [37] : df\_metric\_seg\_annee.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 27 columns):
 #   Column          Non-Null Count   Dtype  
--- 
 0   region          48 non-null      object  
 1   annnee          48 non-null      int64  
 2   MSE              48 non-null      float64
 3   RMSE             48 non-null      float64
 4   MAE              48 non-null      float64
 5   R2             48 non-null      float64
 6   mediane_ech_phys
 7   irq_ech_phys
 8   mediane_ech_phys_glob
 9   irq_ech_phys_glob
 10  mae_err_abs_glob
 11  ecart_type_err_abs_glob
 12  variance_err_abs_glob
 13  mediane_erreur_absolu_glob
 14  irq_err_abs_glob
 15  mae_indice_confiance_global
 16  mediane_indice_confiance_global
 17  mediane_ech_phys_seg
 18  irq_ech_phys_seg
 19  mae_err_abs_seg
 20  ecart_type_err_abs_seg
 21  variance_err_abs_seg
 22  mediane_erreur_absolu_seg
 23  irq_err_abs_seg
 24  mae_indice_confiance_seg
 25  mediane_indice_confiance_seg
 26  mediane_indice_confiance_glob
dtypes: float64(25), int64(1), object(1)
memory usage: 10.3+ KB

```

```

In [38]: # Export du DataFrame avec Les métriques
df_metric_seg_annee.to_excel('resultats_xgb_metrics_20_23_glob_seg_IC.xlsx', index=False)
df_metric_seg_annee.to_csv('resultats_xgb_metrics_20_23_glob_seg_IC.csv', index=False)

```

```

In [39]: # ***** Visualisation des médianes annuelles des valeurs réelles des échanges physiques en fonction des médianes des indices de confiance *****

```

```

In [40]: # Filtrage des données par année
annees = df_metric_seg_annee['annee'].unique()

for annee in annees:
    # Création de la figure
    fig, axs = plt.subplots(2, 1, figsize=(15, 24))

    df_annee = df_metric_seg_annee[df_metric_seg_annee['annee'] == annee]

```

```

# Mlavage de points des 'indice_confiance_global'
scatter1 = sns.scatterplot(ax=axes[0],x='mediane_indice_confiance_global' , y='mediane_ech_phys' , hue='region' , palette='tab10' , s=100,
                           data=df_annee , legend = 'brief')

axes[0].set_xlim(0, 110)
axes[0].set_title("prévisions par modélisation globale" , fontsize=14)
axes[0].set_xlabel("Médiane Indice Confiance Global (%)")
axes[0].set_ylabel("Médiane Échanges Physiques (MW)" )
axes[0].grid(True, which='both' , linestyle='--' , linewidth=0.5 , color='grey')
# Légende
axes[0].legend(loc='upper center' , bbox_to_anchor=(0.5, 1.3) , ncol=2, fontsize=12)

# Mlavage de points des 'indice_confiance_seg'
sns.scatterplot(ax=axes[1],x='mediane_indice_confiance_seg' , y='mediane_ech_phys' , hue='region' , palette='tab10' , s=100,
                           data=df_annee , legend = False)

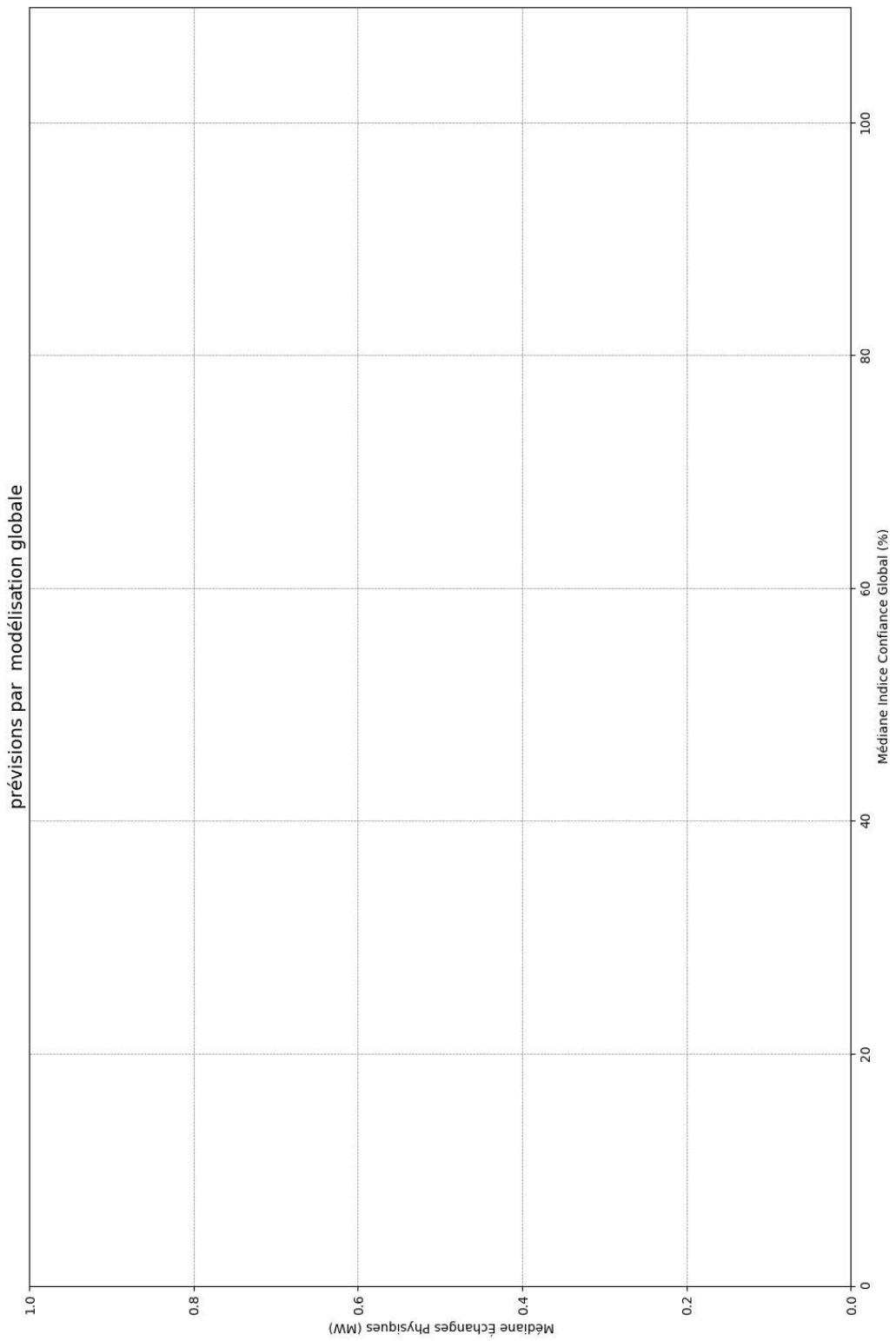
axes[1].set_xlim(0, 110)
axes[1].set_title("prévisions par modélisations régionales" , fontsize=14)
axes[1].set_xlabel("Médiane Indice Confiance Segmenté (%)")
axes[1].set_ylabel("Médiane Échanges Physiques (MW)" )
axes[1].grid(True, which='both' , linestyle='--' , linewidth=0.5 , color='grey')

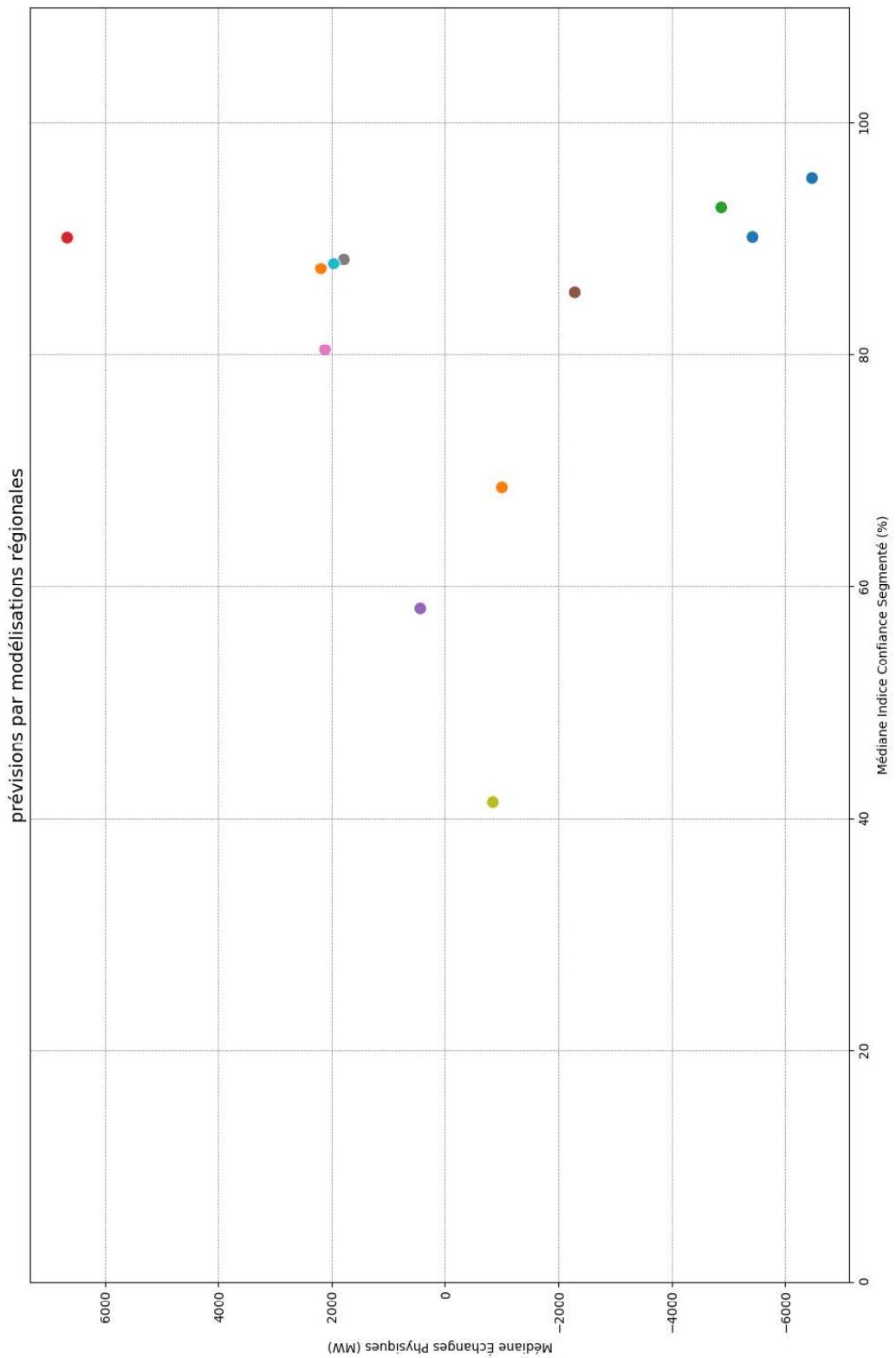
# Titre général
fig.suptitle(f"Echanges physiques (Médianes annuelles) & Indices de Confiance de modélisation (médianes) - {annee}" , fontsize=22, weight='bold' , y=0.96)

# Ajustement de la mise en page et sauvegarde de l'image
plt.tight_layout(rect=[0, 0, 1, 0.93])
plt.savefig(f"scatters_mediane_ech_phys_IC_{annee}.png" , dpi=300, bbox_inches='tight')
plt.show(fig)

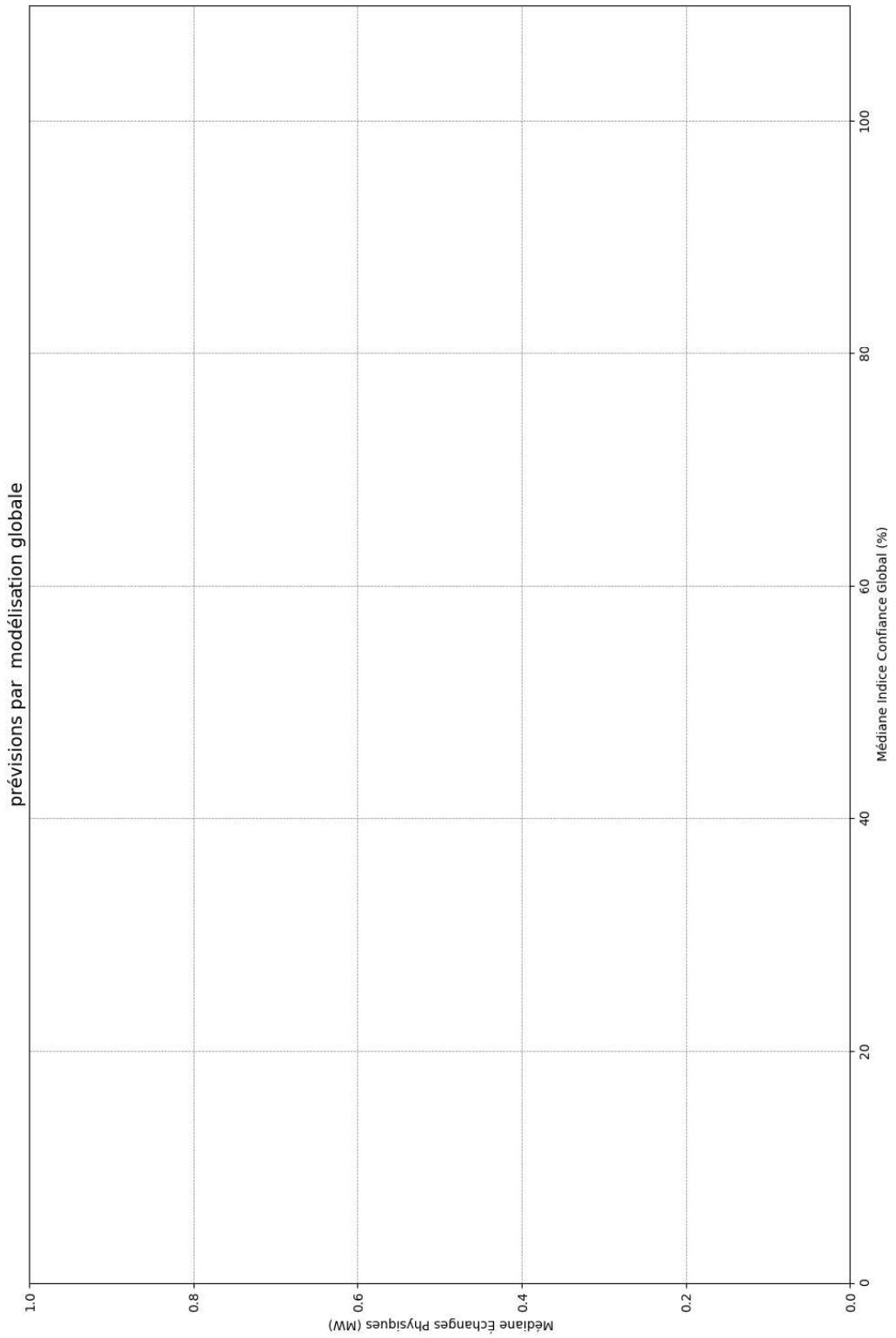
```

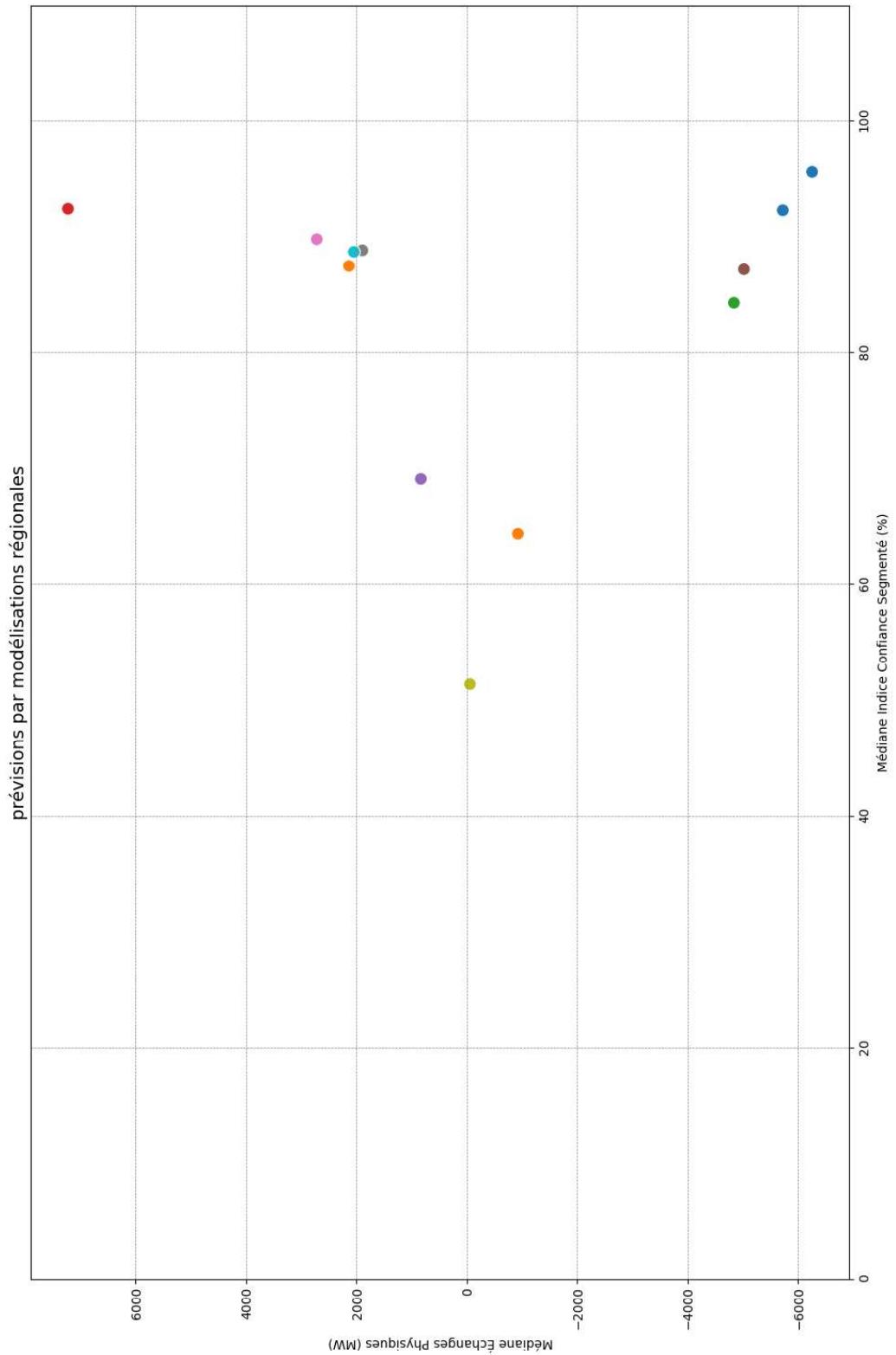
## Echanges physiques (Médianes annuelles) & Indices de Confiance de modélisation (médianes) - 2020



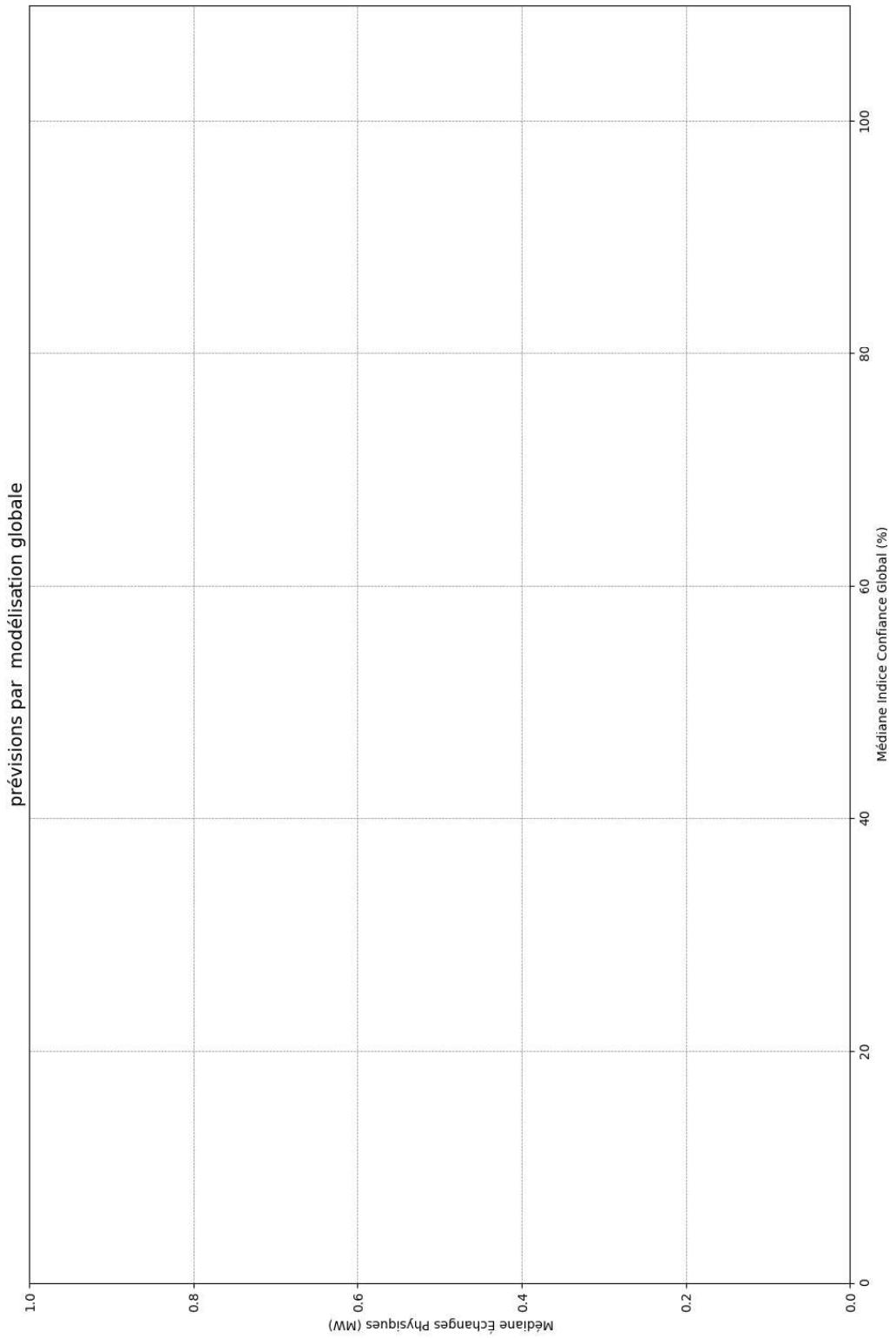


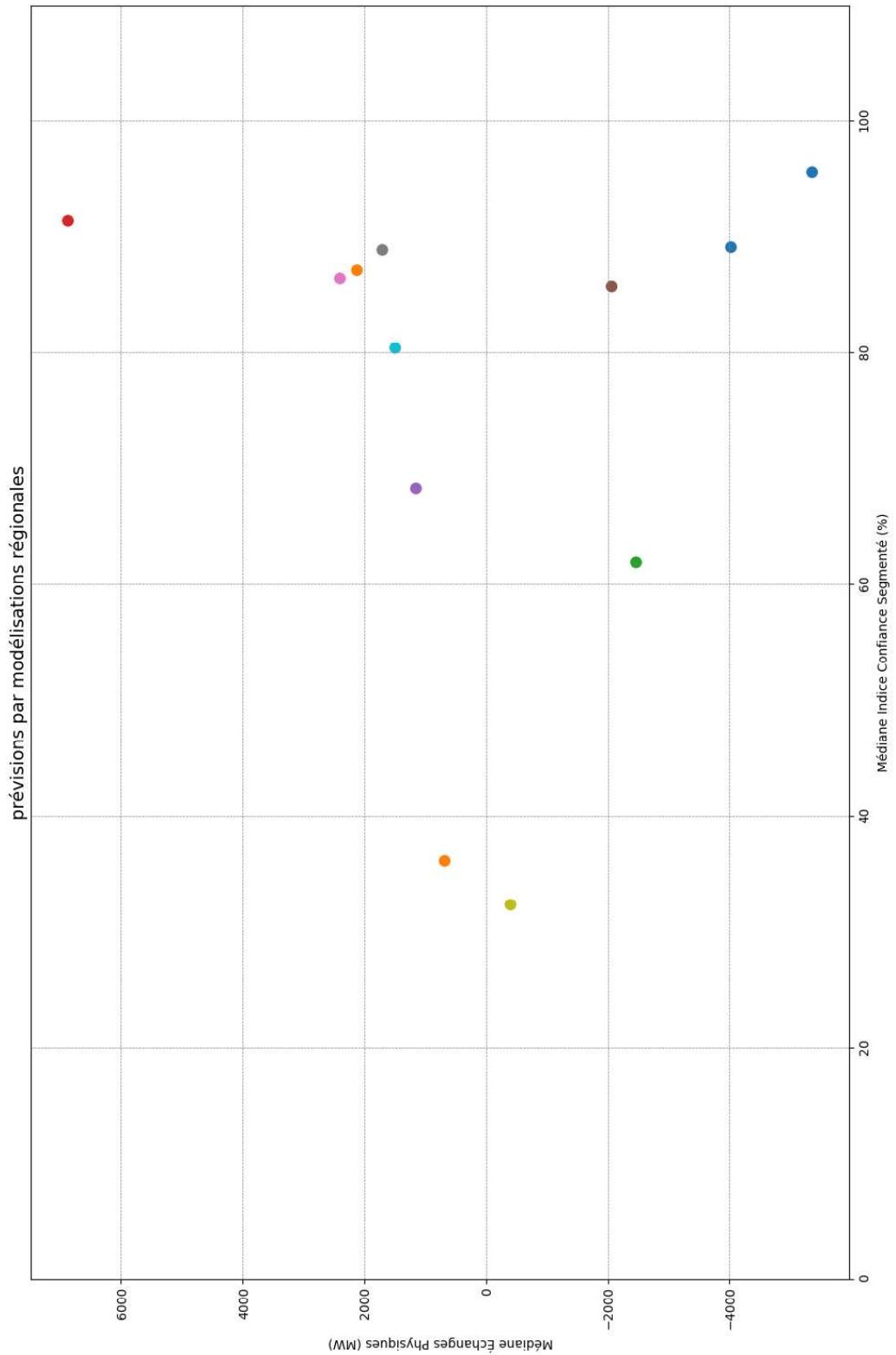
## Echanges physiques (Médianes annuelles) & Indices de Confiance de modélisation (médianes) - 2021



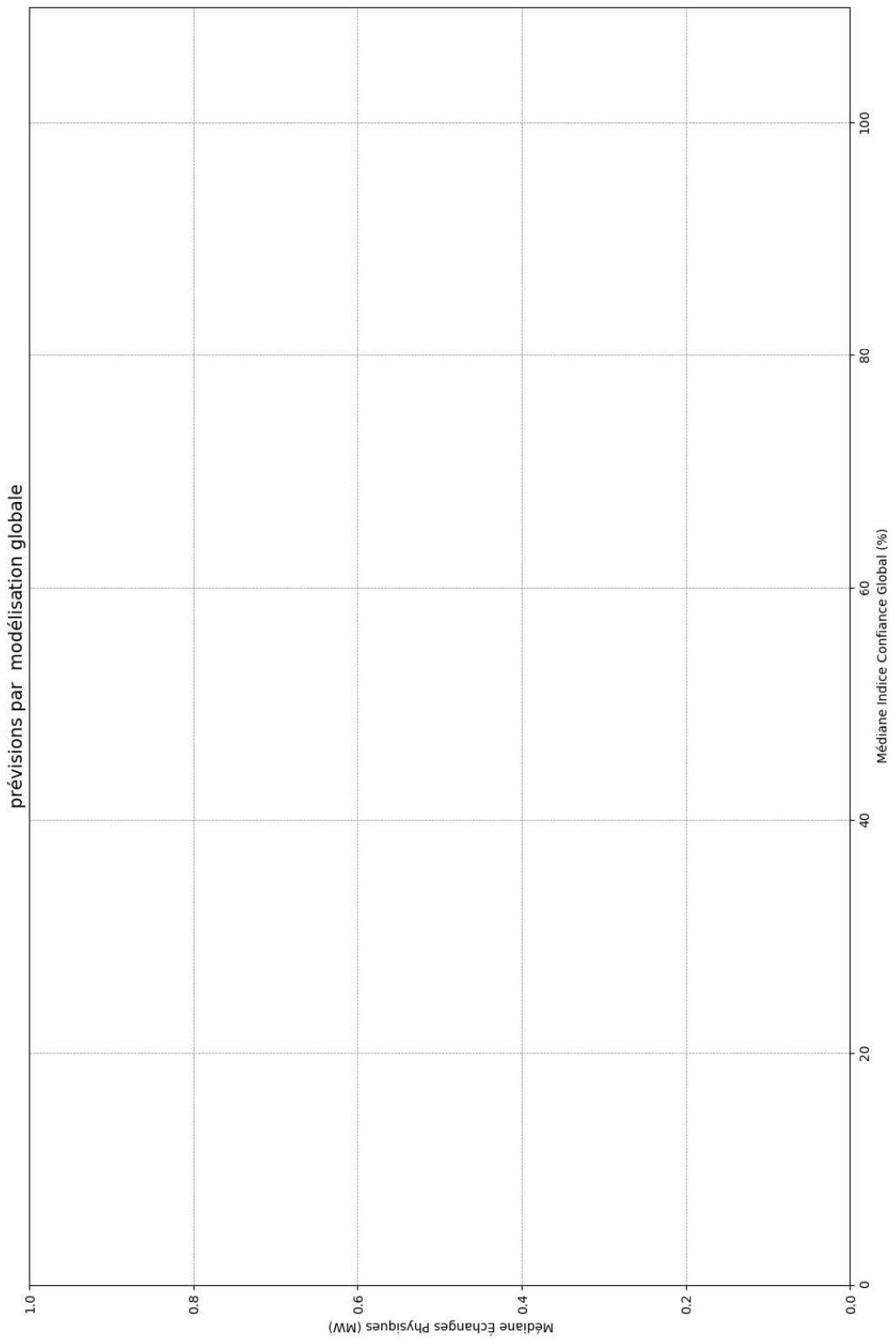


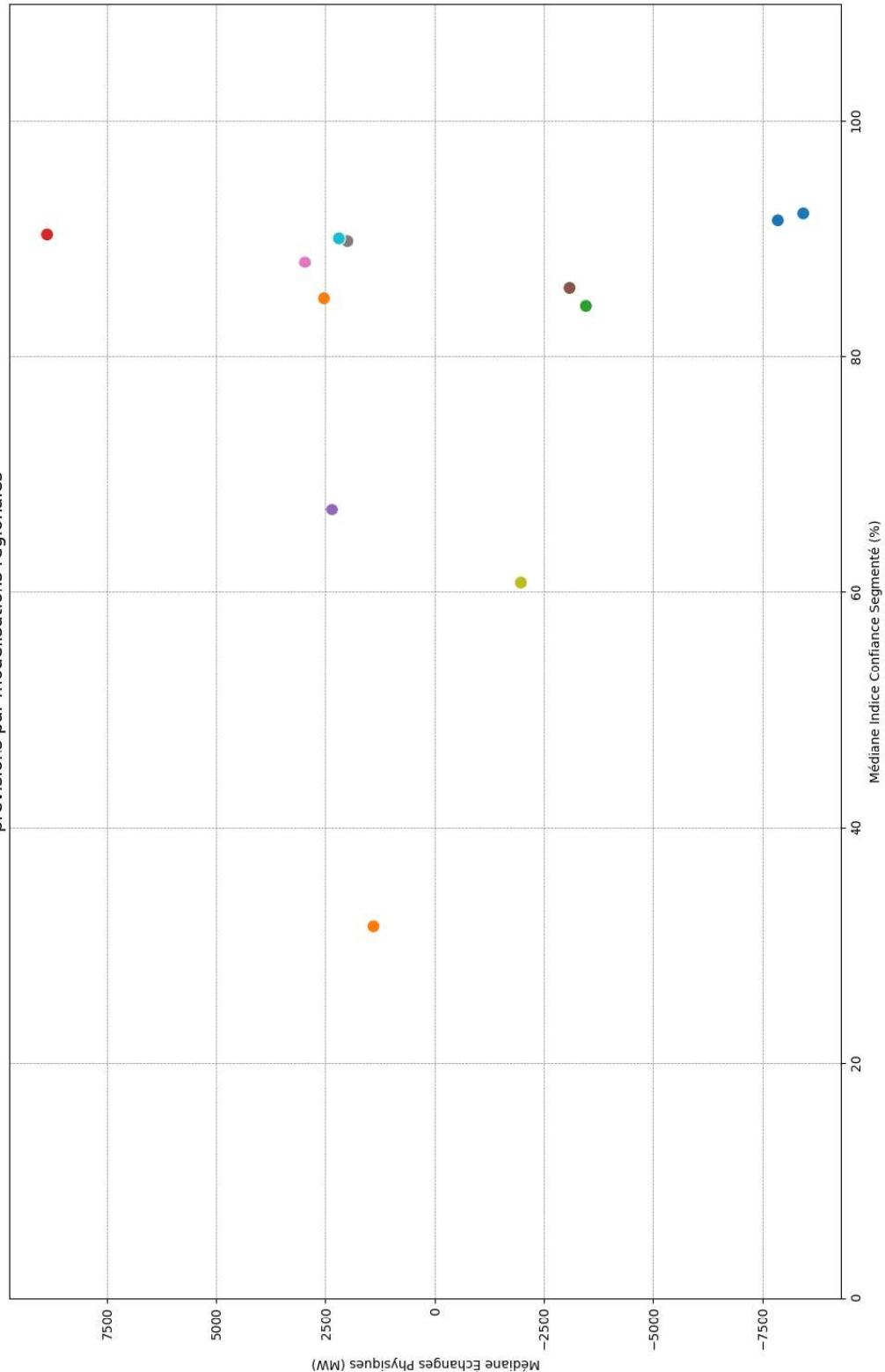
## Echanges physiques (Médianes annuelles) & Indices de Confiance de modélisation (médianes) - 2022





## Echanges physiques (Médianes annuelles) & Indices de Confiance de modélisation (médianes) - 2023





```
In [41]: # **** Nuages de points de toutes les valeurs 'échanges physiques' réelles en fonction des indices de confiance ****
```

```
# Initialisation d'une palette de couleurs selon la région
palette = sns.color_palette('tab10', n_colors=df_blackout_xgb_20_23['region'].nunique())
années = df_blackout_xgb_20_23['année'].unique()
for année in années:
    df_année = df_blackout_xgb_20_23[df_blackout_xgb_20_23['année'] == année]

# Création d'une figure avec deux sous-graphiques
fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(15, 24))
```

```

# Premier sous-graphique : ech_physiques en fonction de 'indice_confiance_glob'
sns.scatterplot(ax=axes[0], data=df_annee, x='indice_confiance_glob', y='ech_physiques',
                 hue='region', palette=palette, legend=False, s = 10)
axes[0].set_title("prévisions par modélisation globale")
axes[0].set_xlabel('Indice de Confiance glob. (%)')
axes[0].set_ylabel('Échanges Physiques réels (MW)')

# Deuxième sous-graphique : ech_physiques en fonction de 'indice_confiance_seg'
sns.scatterplot(ax=axes[1], data=df_annee, x='indice_confiance_seg', y='ech_physiques',
                 hue='region', palette=palette, legend=False, s=10)
axes[1].set_title("prévisions par modélisations régionales")
axes[1].set_xlabel('Indice de Confiance rég. (%)')
axes[1].set_ylabel('Échanges Physiques réels (MW)')

# Légende, sur le 1er sous-graphique
handles, labels = axes[0].get_legend_handles_labels()
fig.legend(handles, labels, loc='upper right', title='Région')

# Titre général
fig.suptitle(f"Échanges physiques & Indices de Confiance de modélisation- {annee}", fontsize=22, weight='bold', y=0.96)

# Ajustement de la mise en page pour éviter les chevauchements
plt.tight_layout()

# Sauvegarde du graphique
plt.savefig(f'scatters_ech_physiques_IC{annee}.png', dpi=300)

plt.show()

```

prévisions par modélisation globale

## Echanges physiques & Indices de Confiance de modélisation- 2020

10000

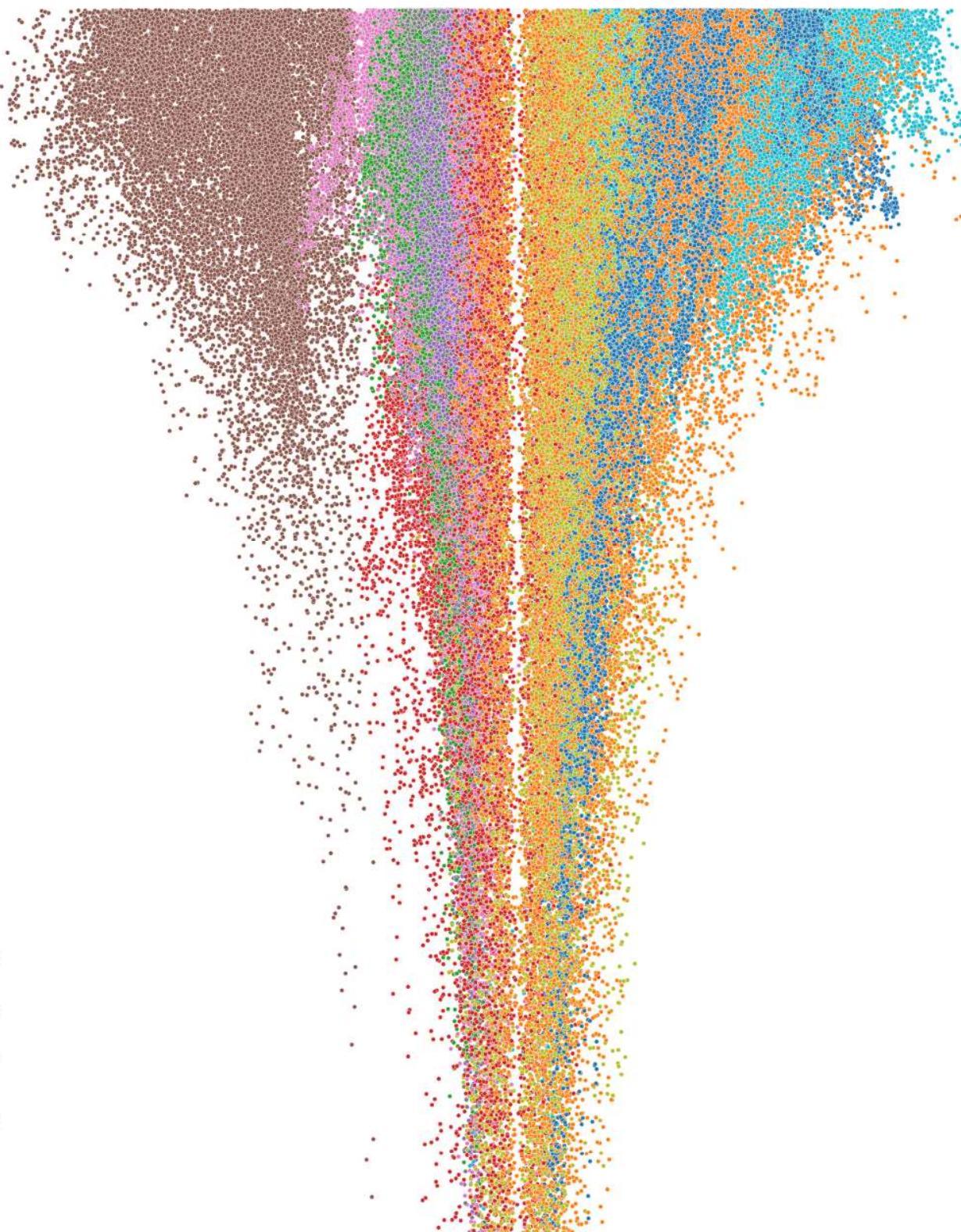
5000

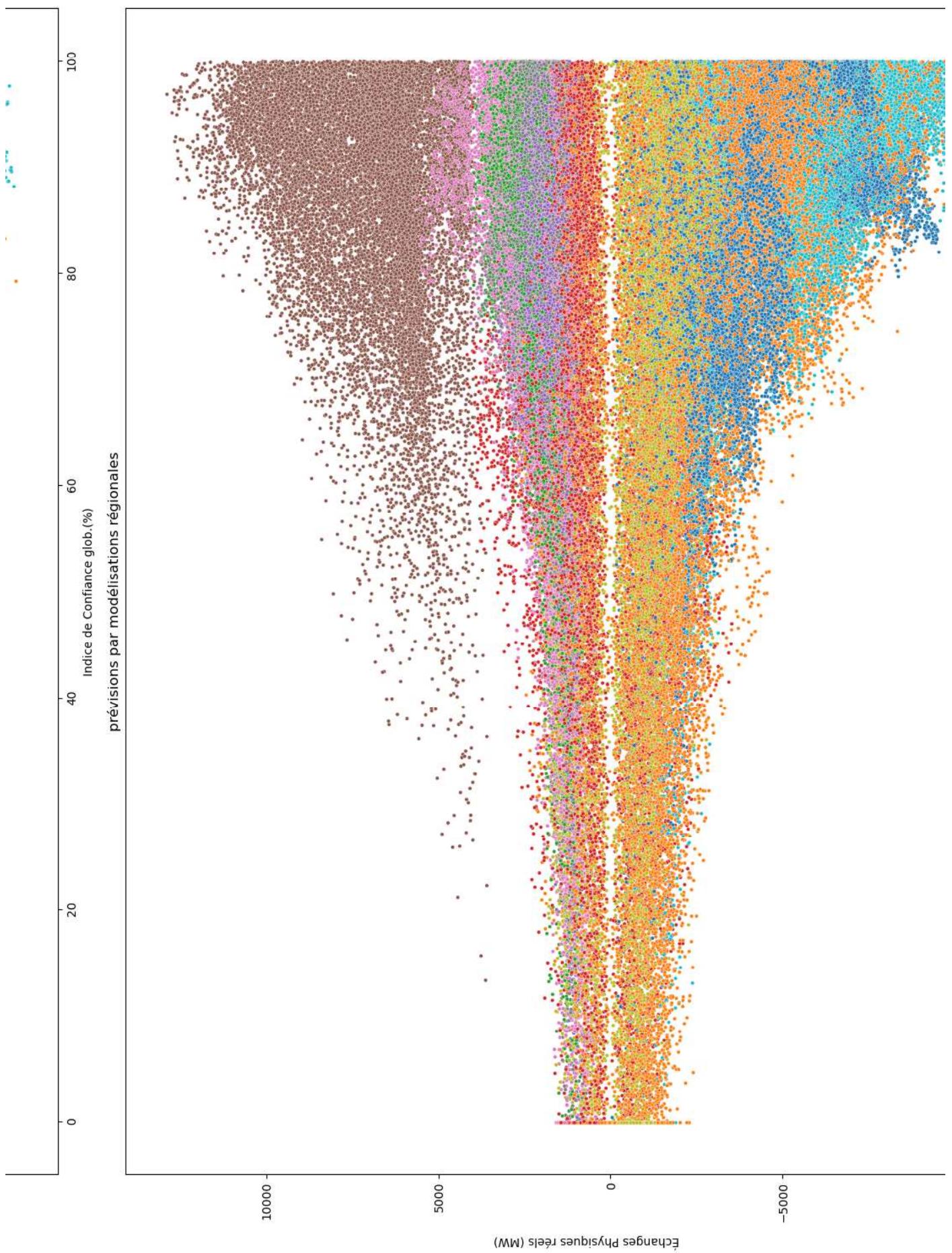
0

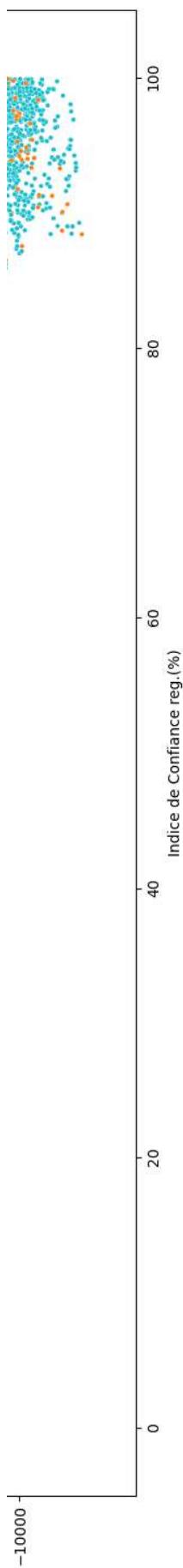
-5000

-10000

Echanges Physiques réels (MW)







prévisions par modélisation globale

## Echanges physiques & Indices de Confiance de modélisation- 2021

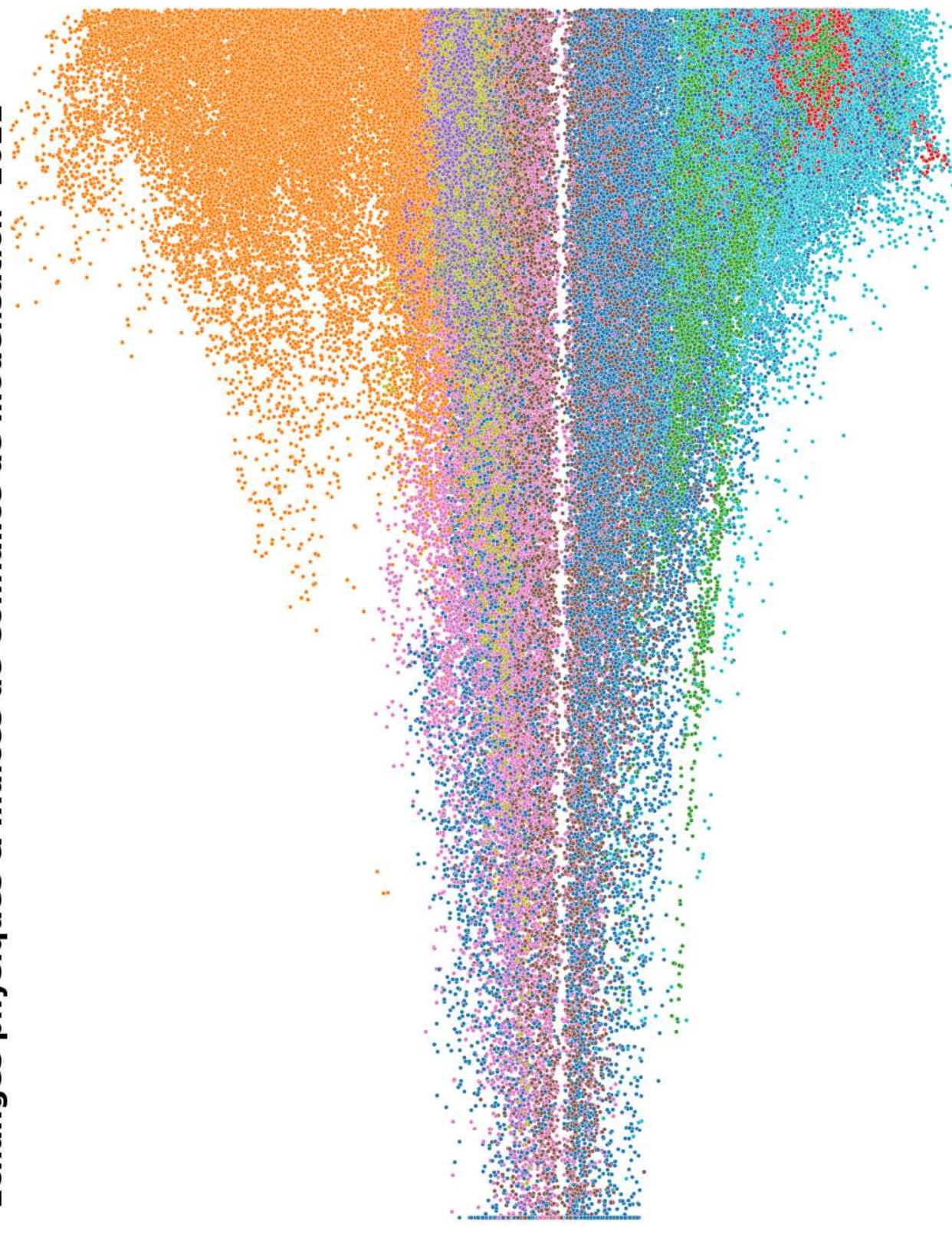
10000

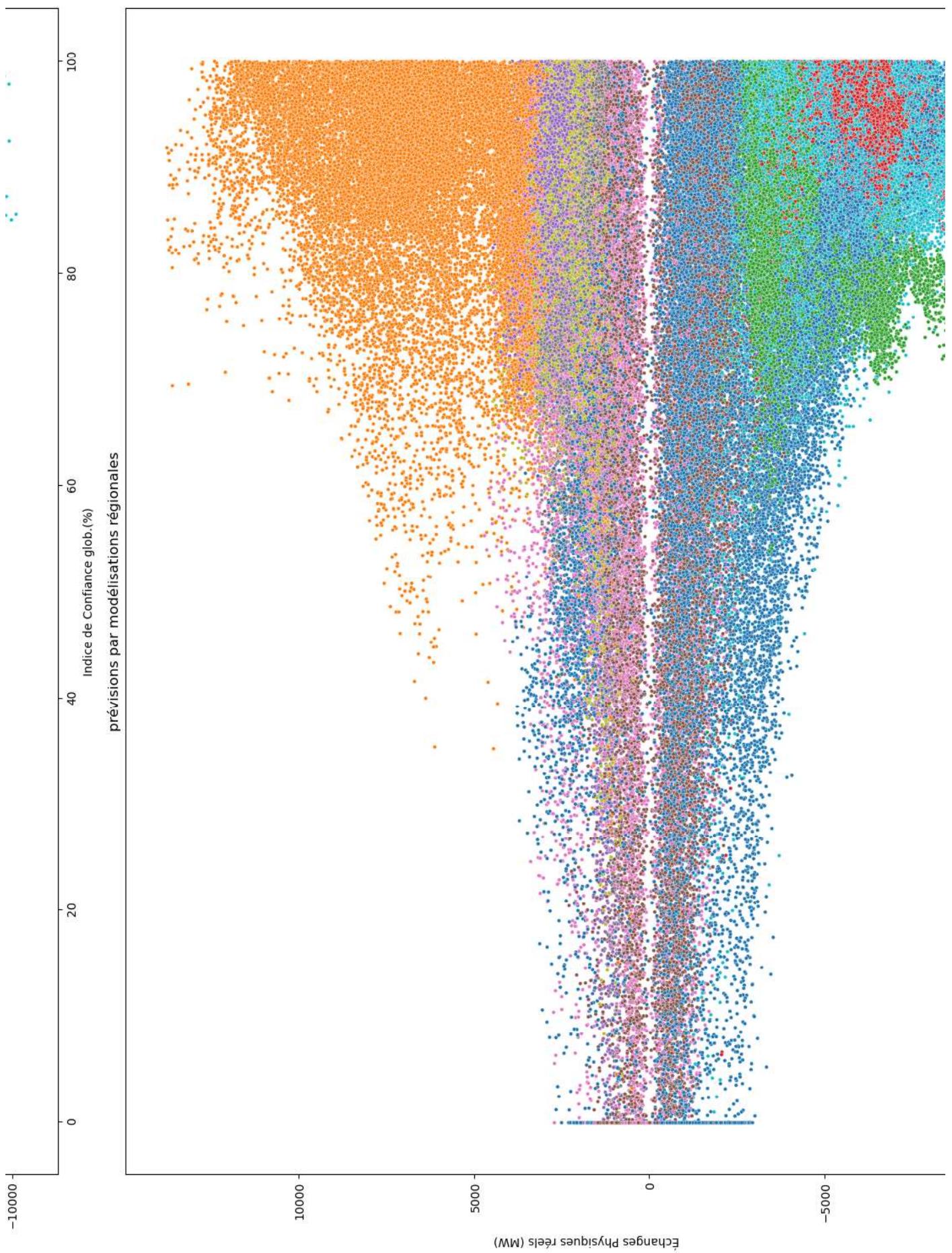
5000

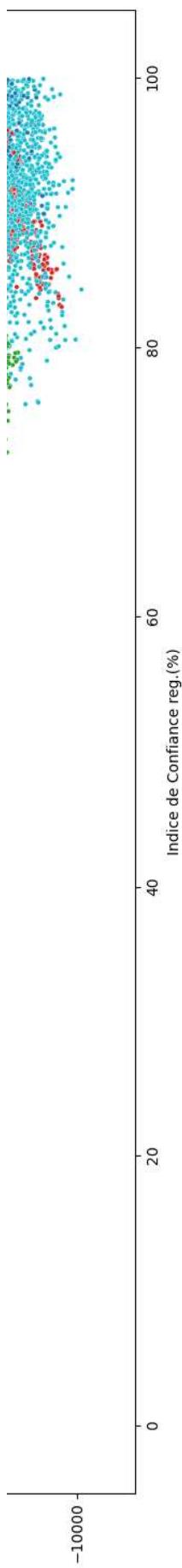
0

-5000

Echanges Physiques réels (MW)



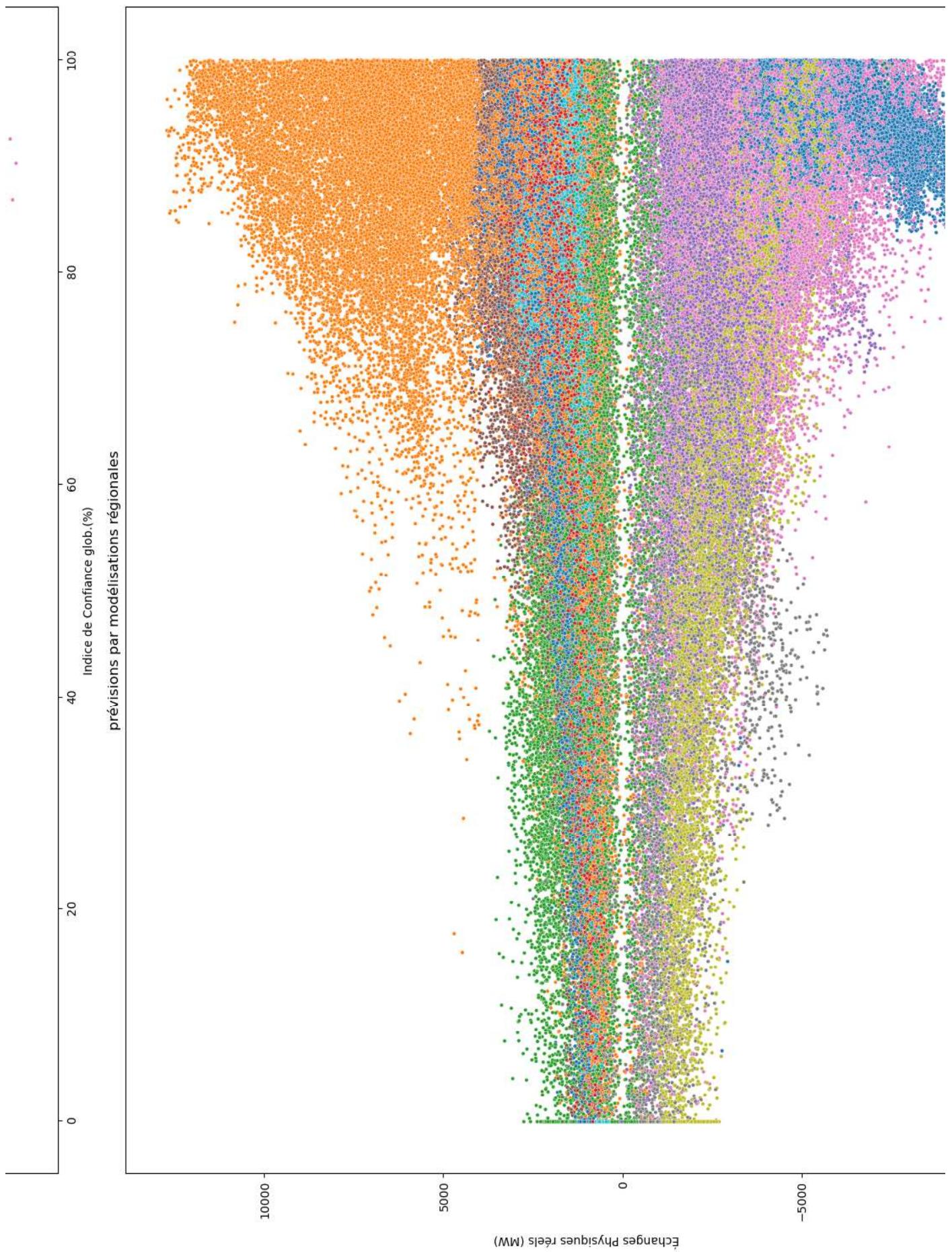


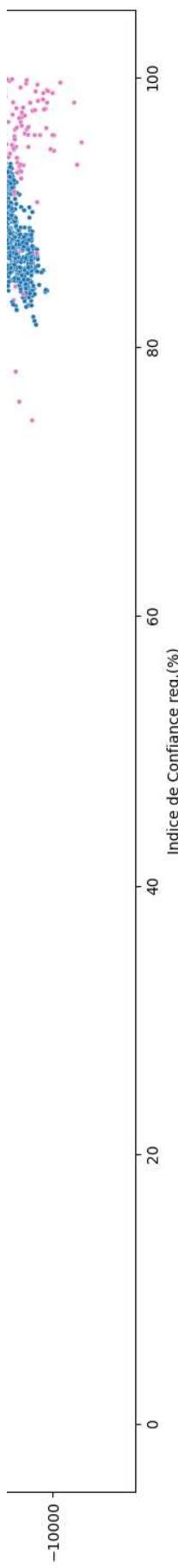


prévisions par modélisation globale

## Echanges physiques & Indices de Confiance de modélisation- 2022







prévisions par modélisation globale

## Echanges physiques & Indices de Confiance de modélisation- 2023

10000

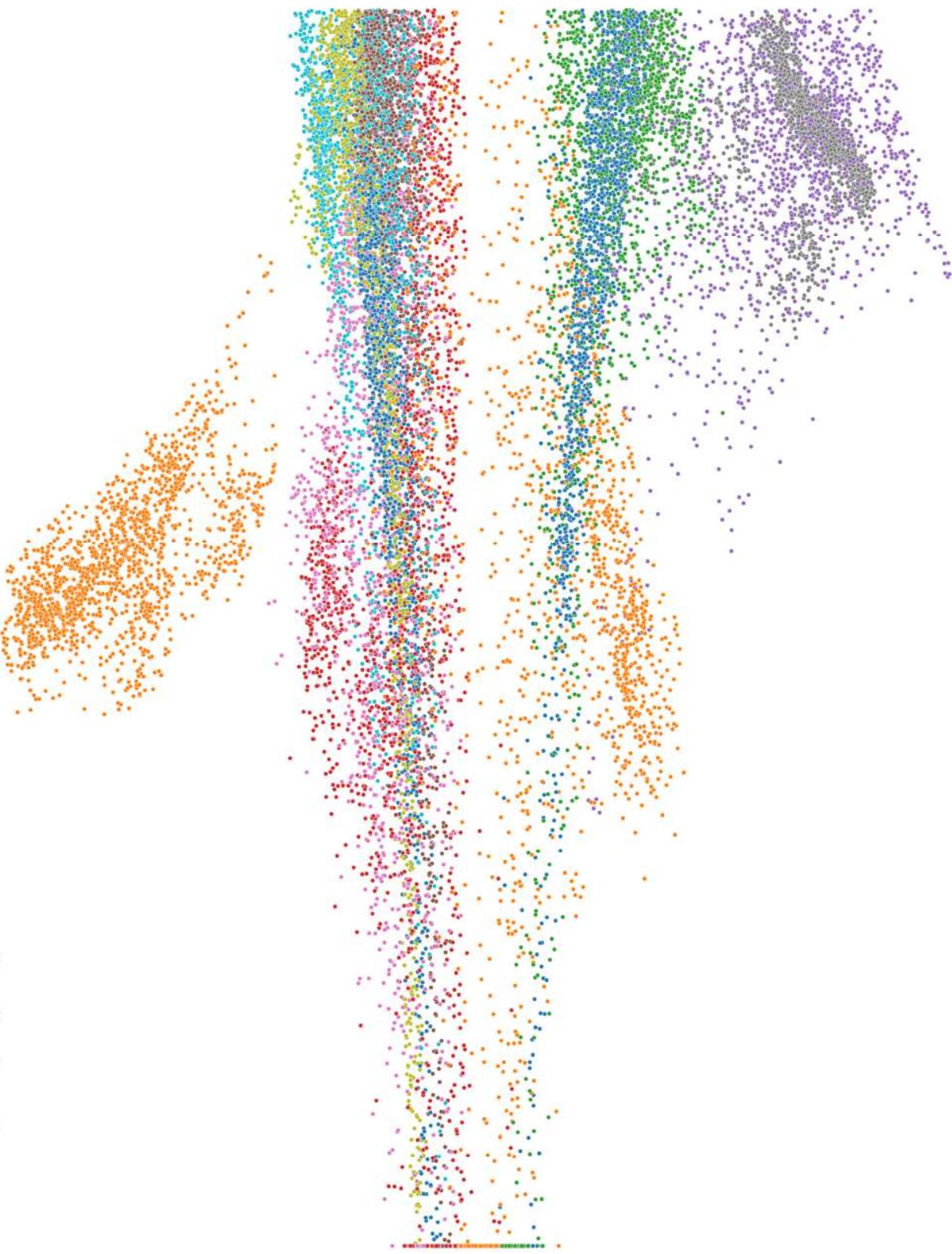
5000

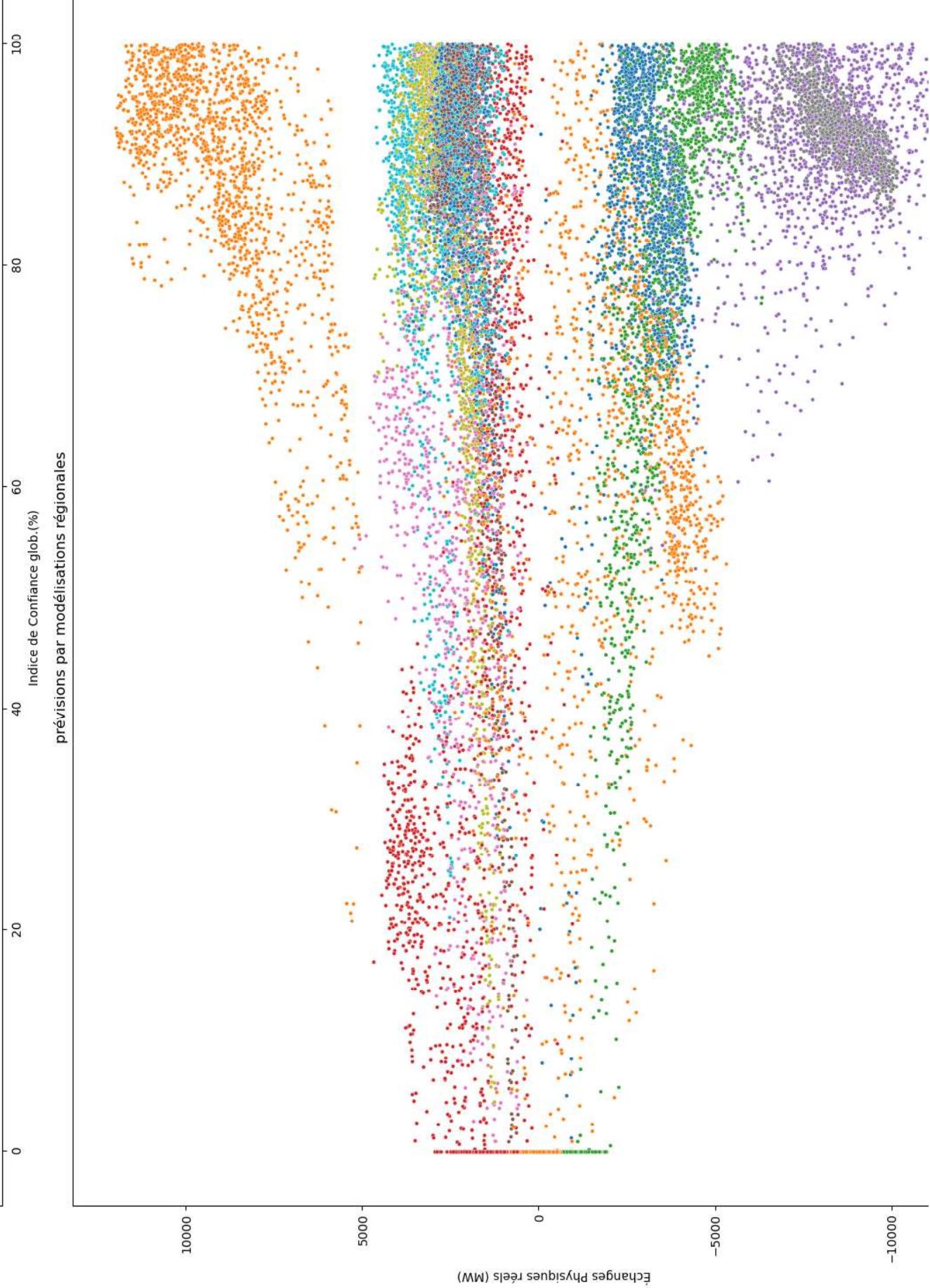
0

-5000

-10000

Échanges Physiques réels (MW)







```
# Ajout de la Légende globale à la figure entière
handles, labels = axes[0].get_legend_handles_labels()
fig.legend(handles, labels, loc='upper center', bbox_to_anchor=(0.5, 1.15), ncol=len(regions), title='Région')

# Titre général
fig.suptitle(f"Echanges physiques & Indices de Confiance de modélisation - {annee}", fontsize=22, weight='bold', y=0.96)

# Ajustement de la mise en page pour éviter les chevauchements
plt.tight_layout()

# Sauvegarde du graphique
plt.savefig(f'scatters_ech_physiques_IC_{annee}.png', dpi=300)
plt.show()
```

Région

prévisions par modélisation globale

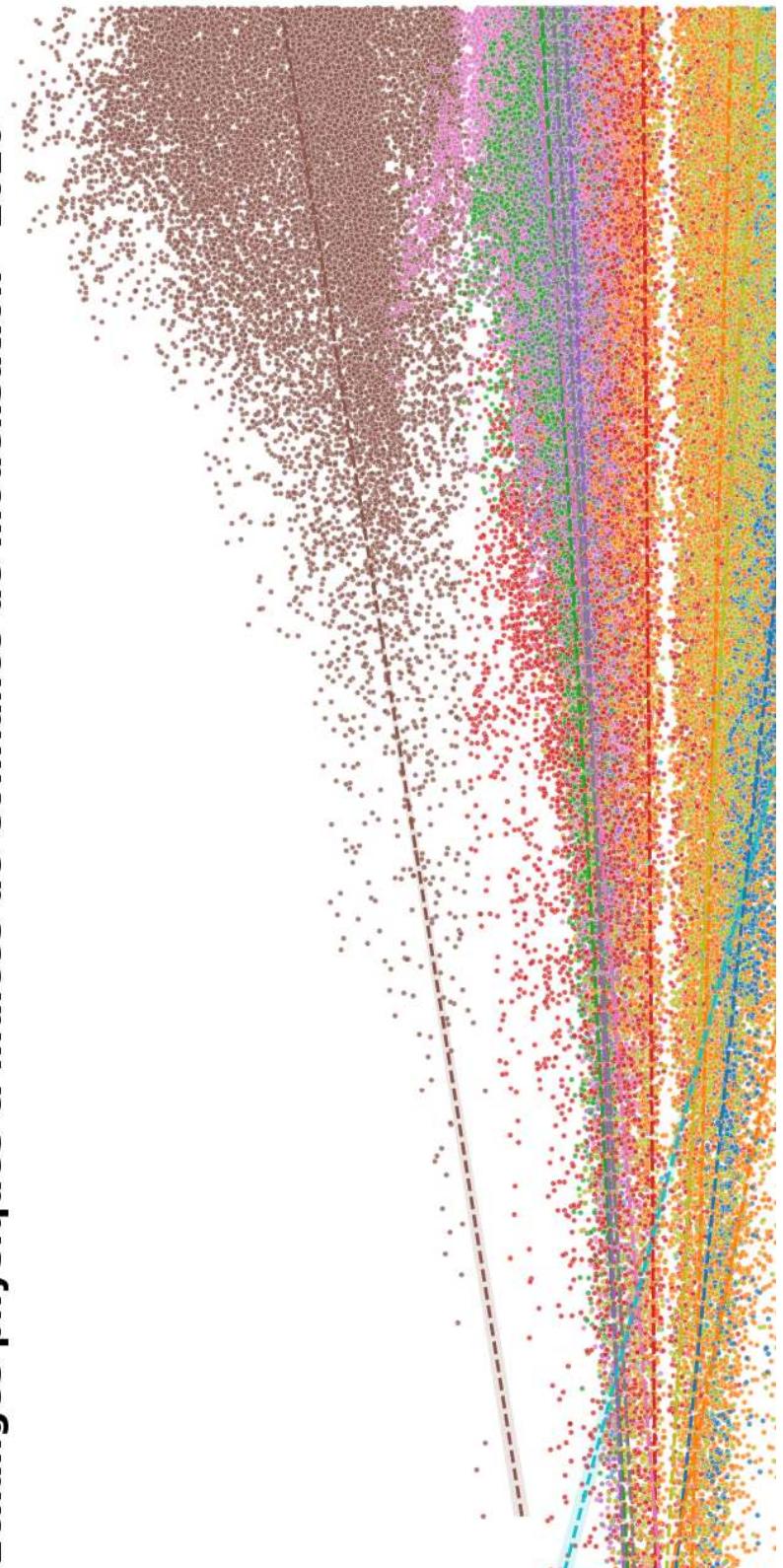
## Echanges physiques & Indices de Confiance de modélisation - 2020

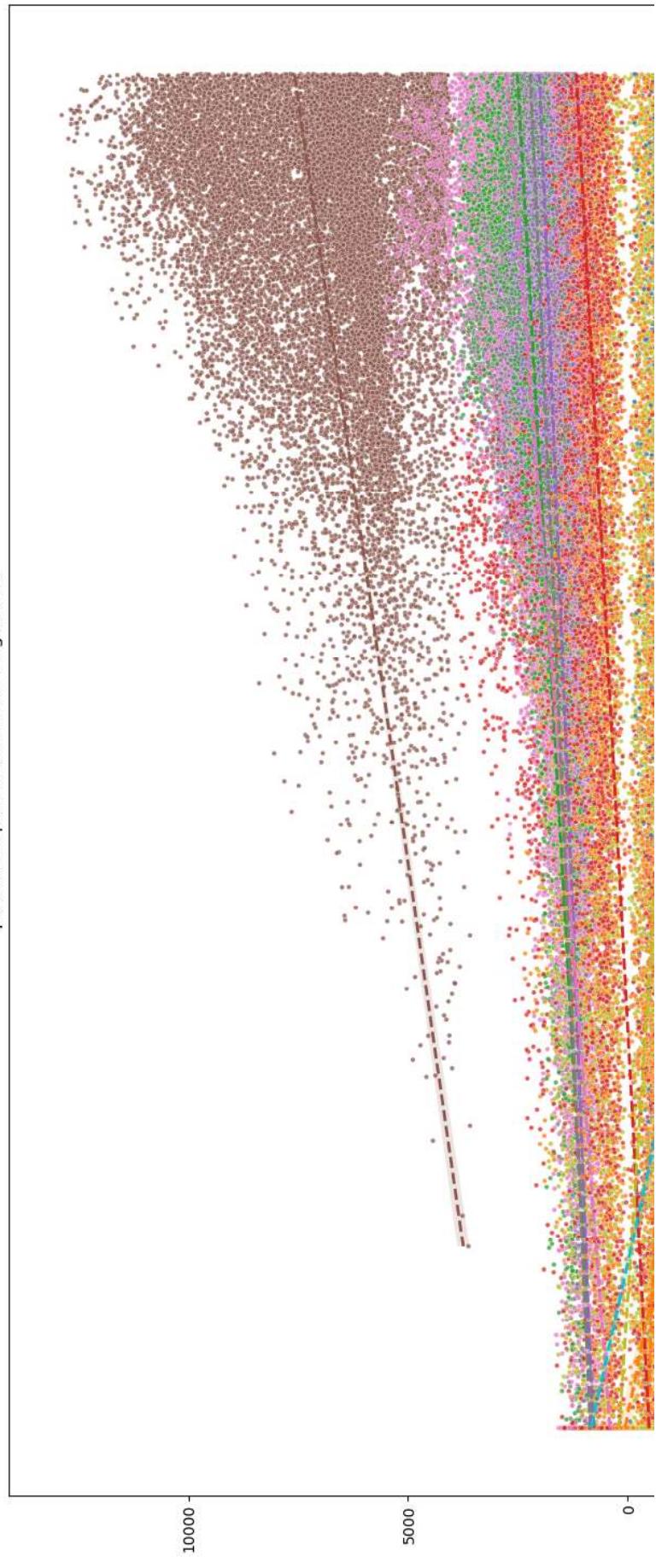
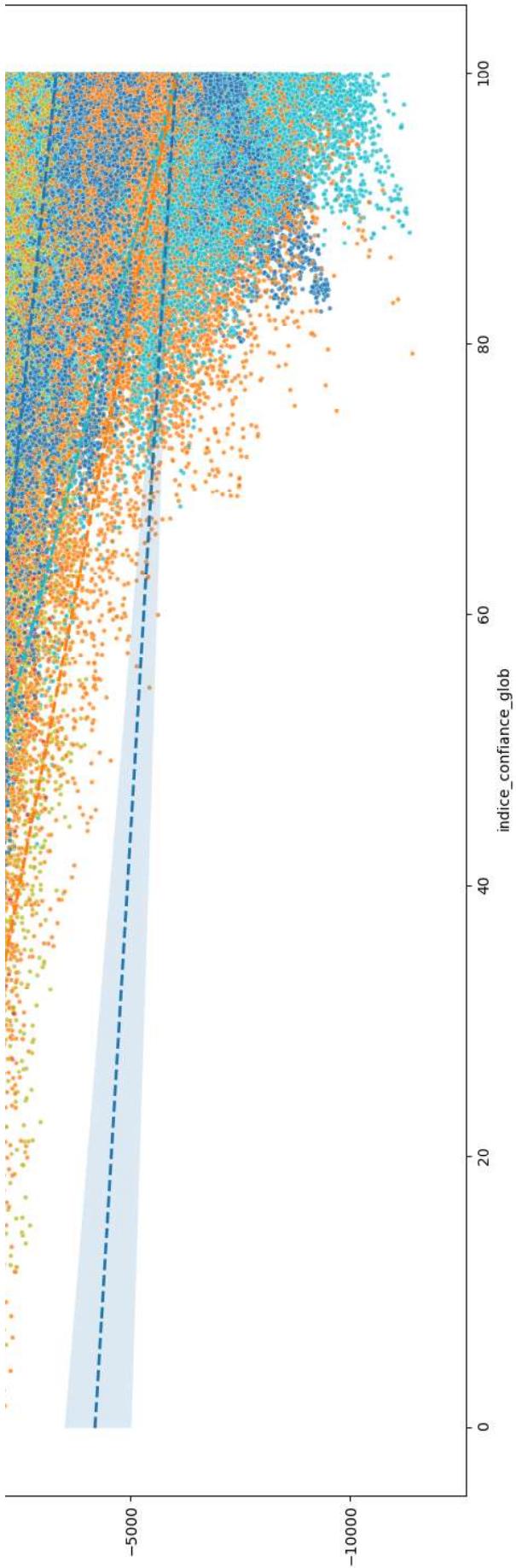
10000

5000

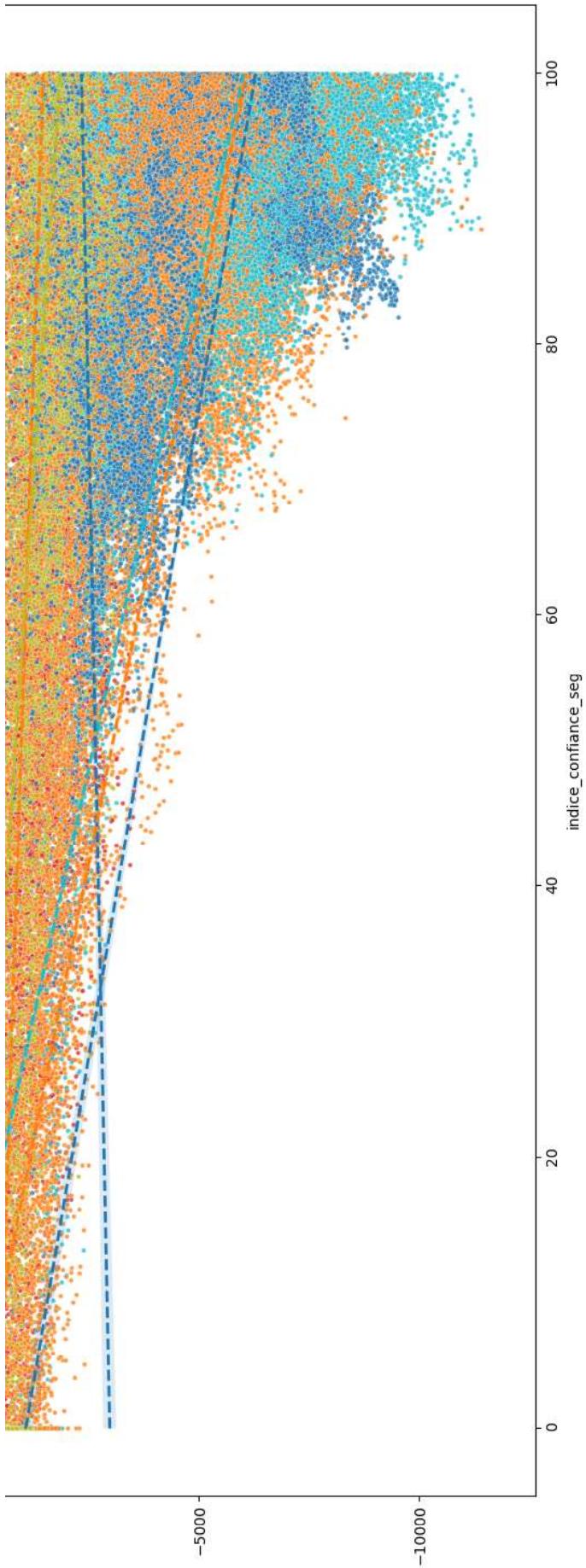
0

éch\_physiques





ech\_physiques



Région

prévisions par modélisation globale

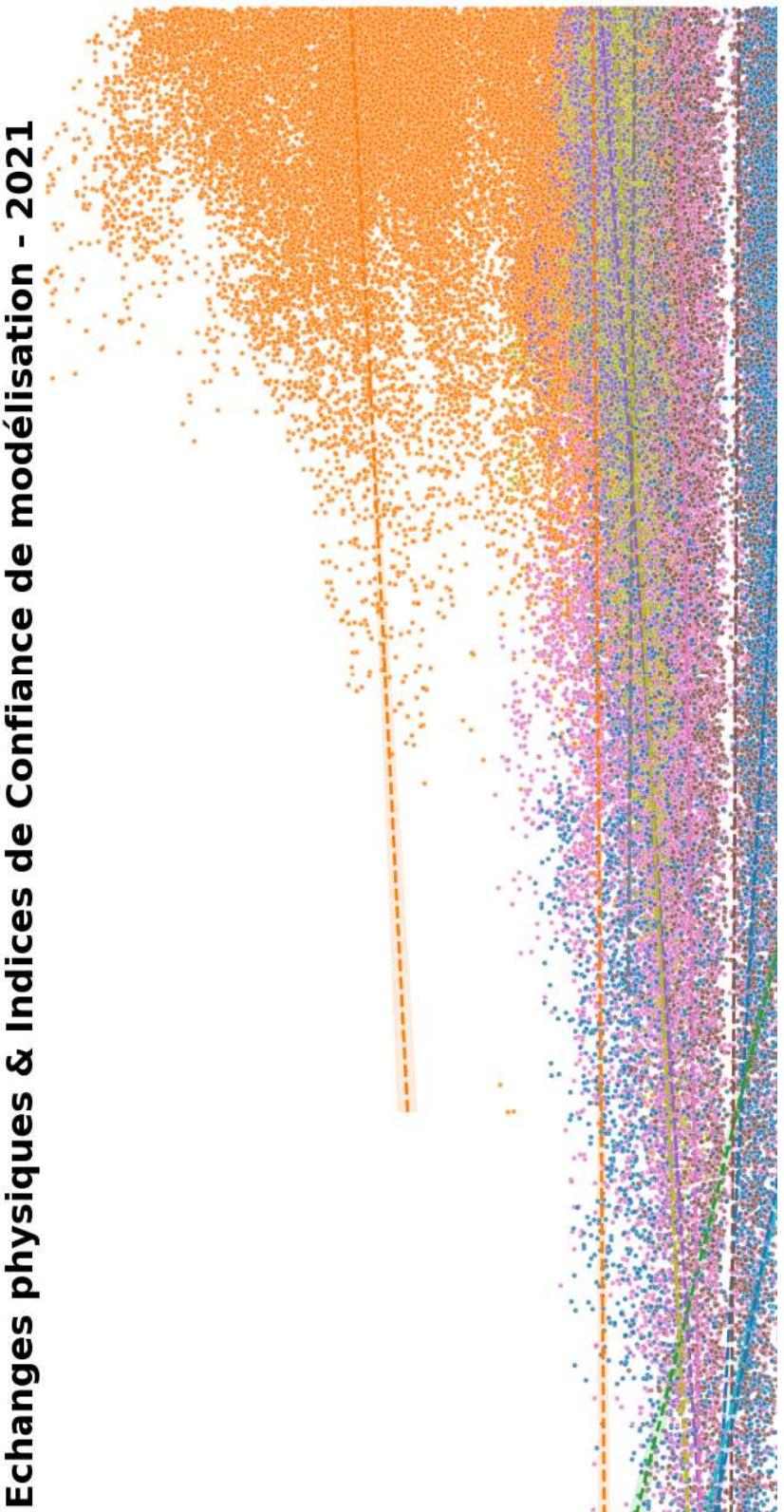
## Echanges physiques & Indices de Confiance de modélisation - 2021

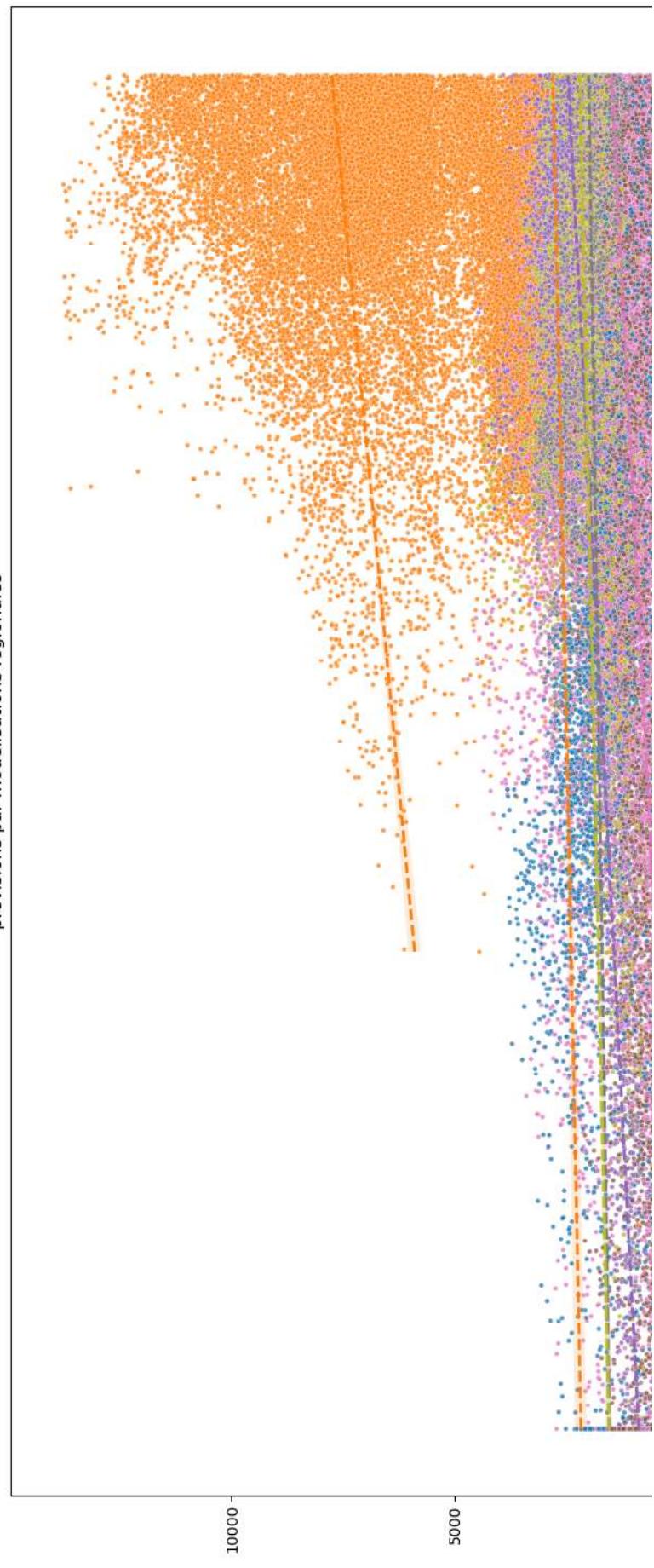
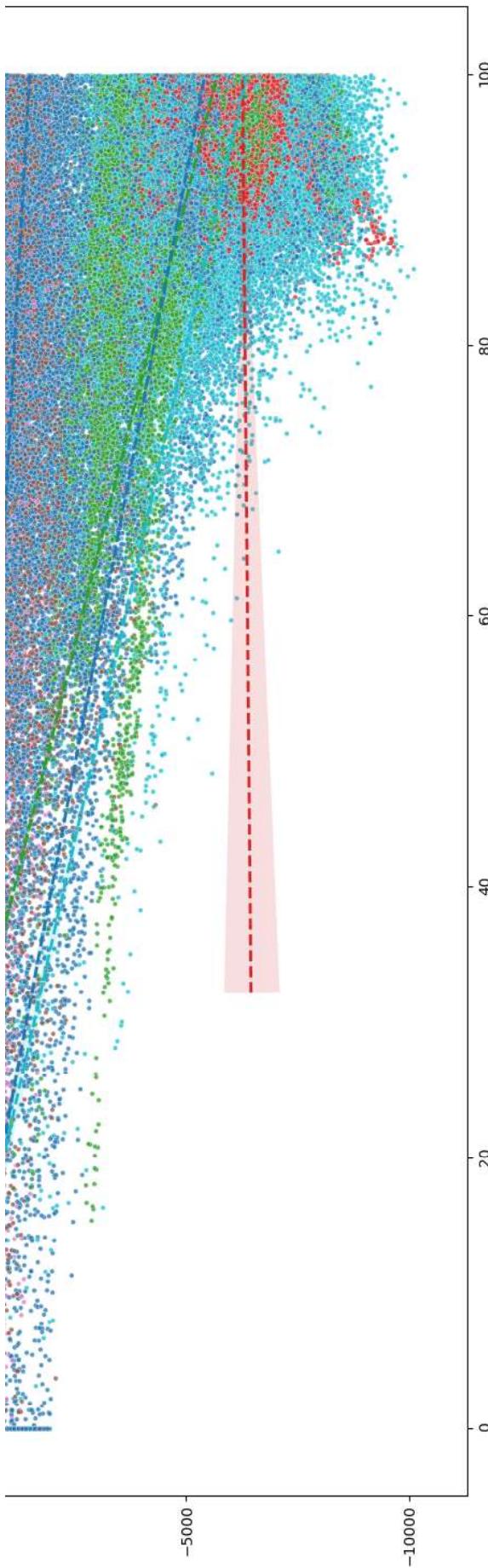
10000

5000

0

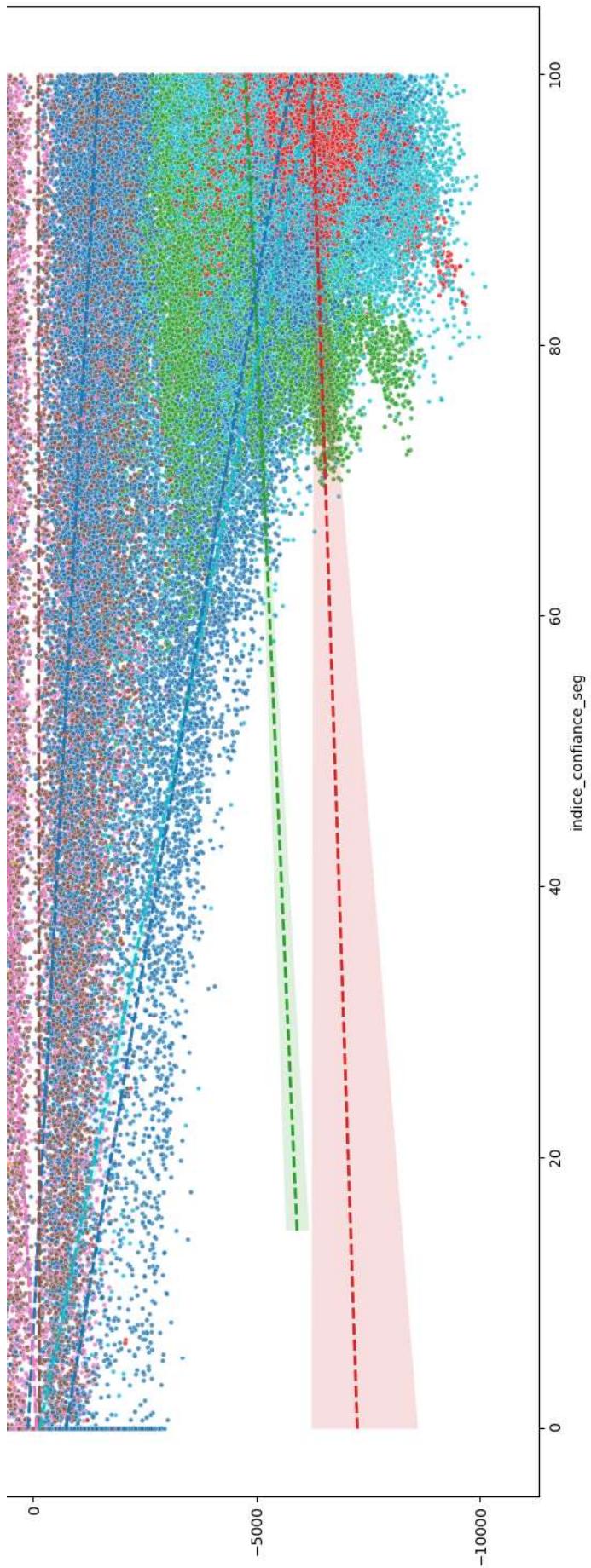
éch\_physiques





prévisions par modélisations régionales

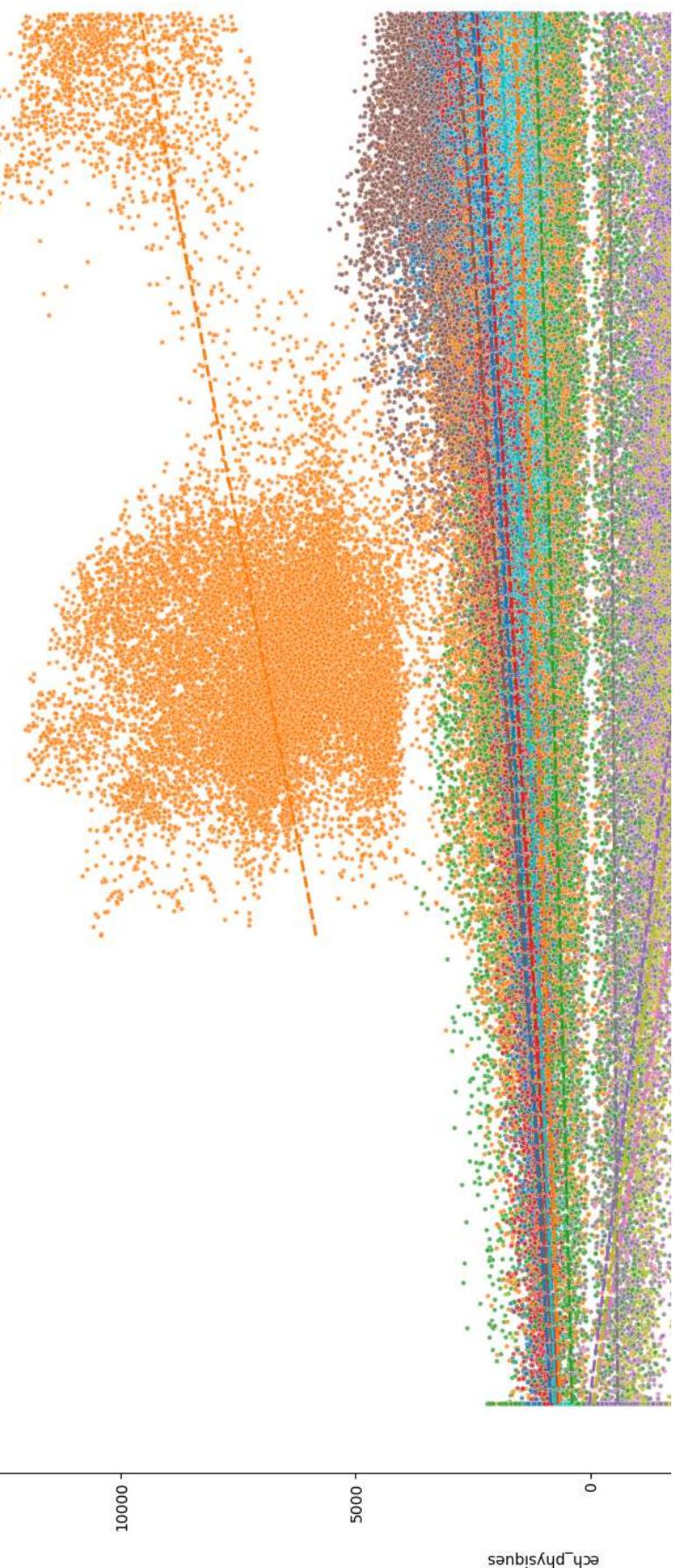
ech\_physiques

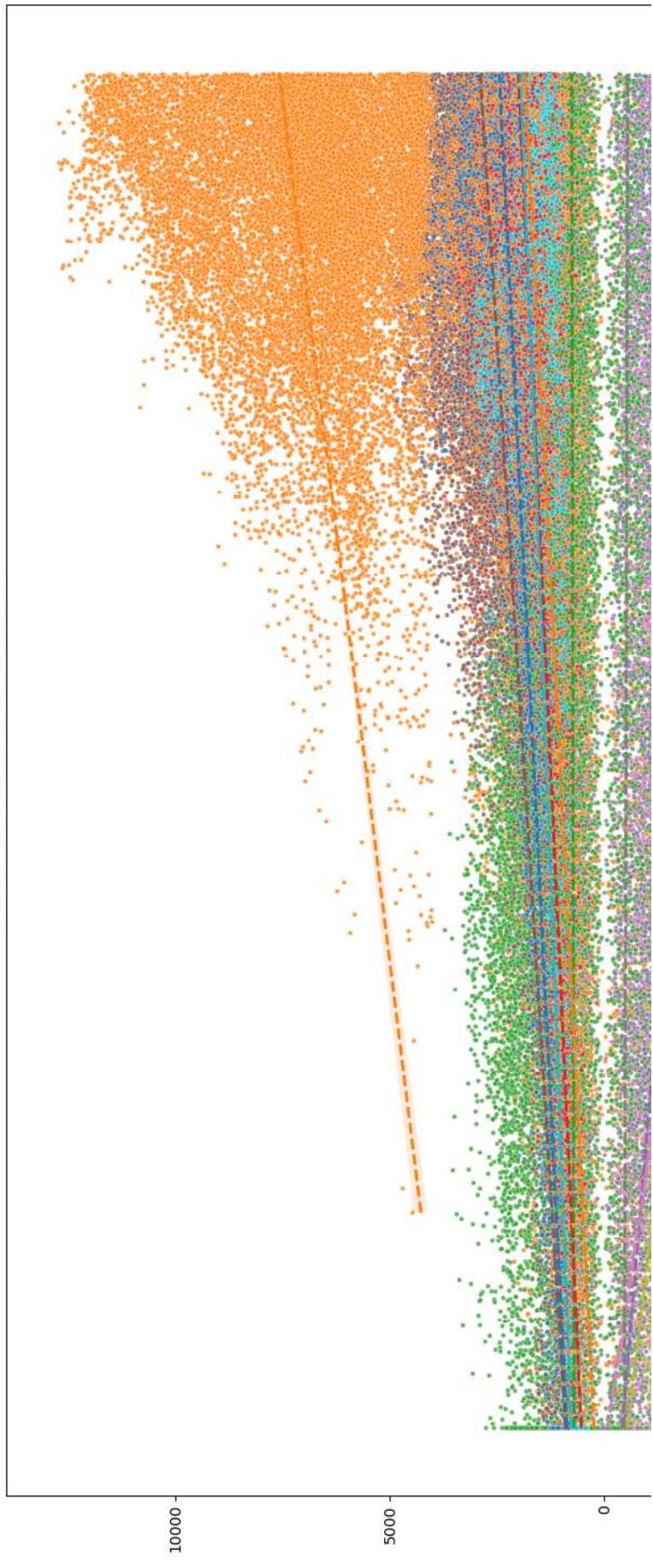
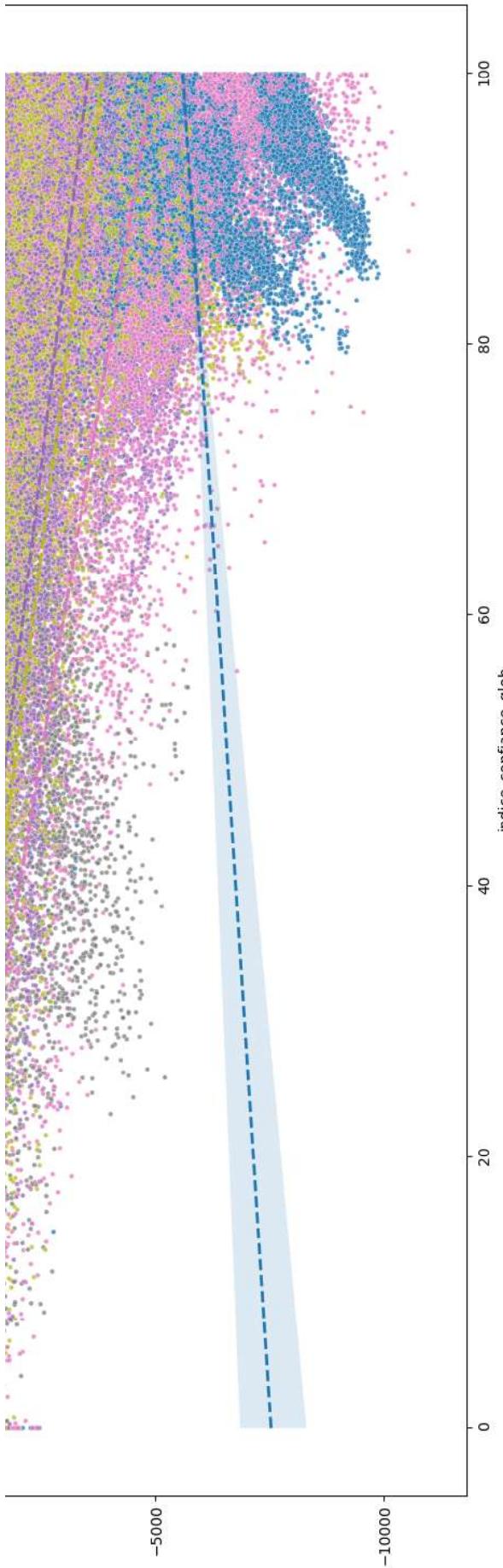


Région

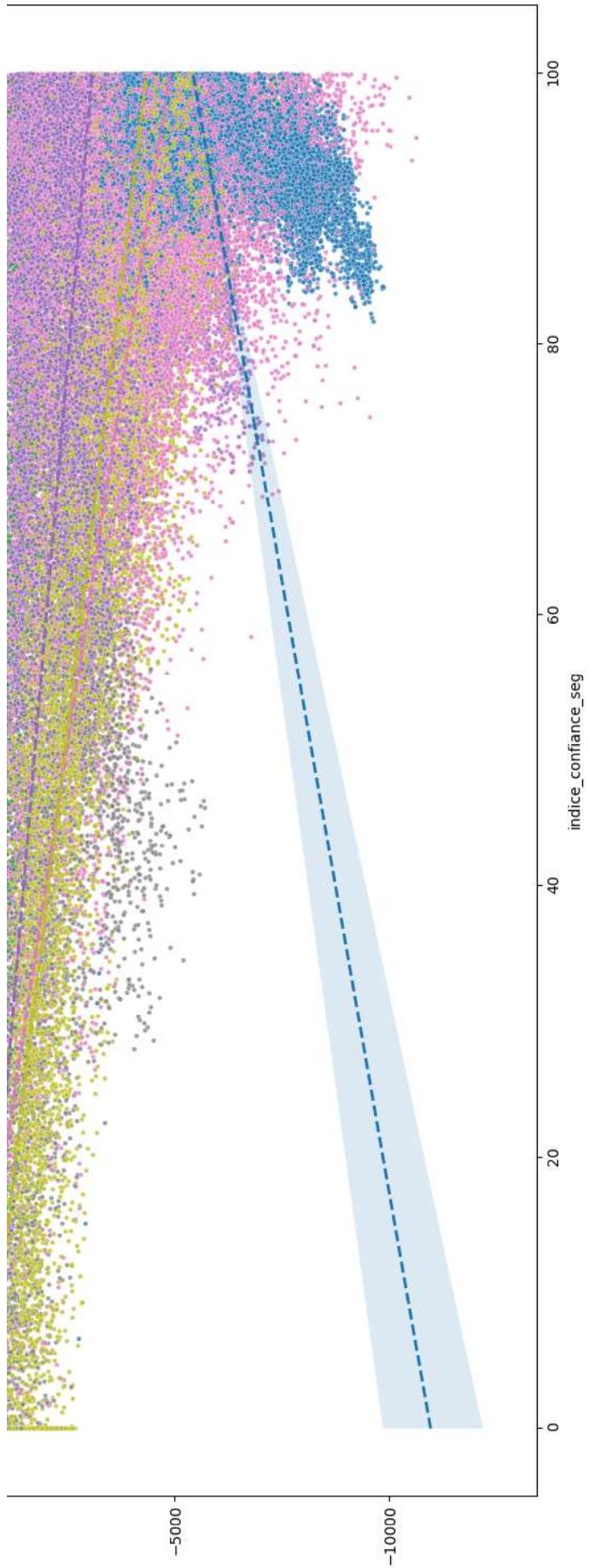
prévisions par modélisation globale

## Echanges physiques & Indices de Confiance de modélisation - 2022





ech\_physiques



Région

prévisions par modélisation globale

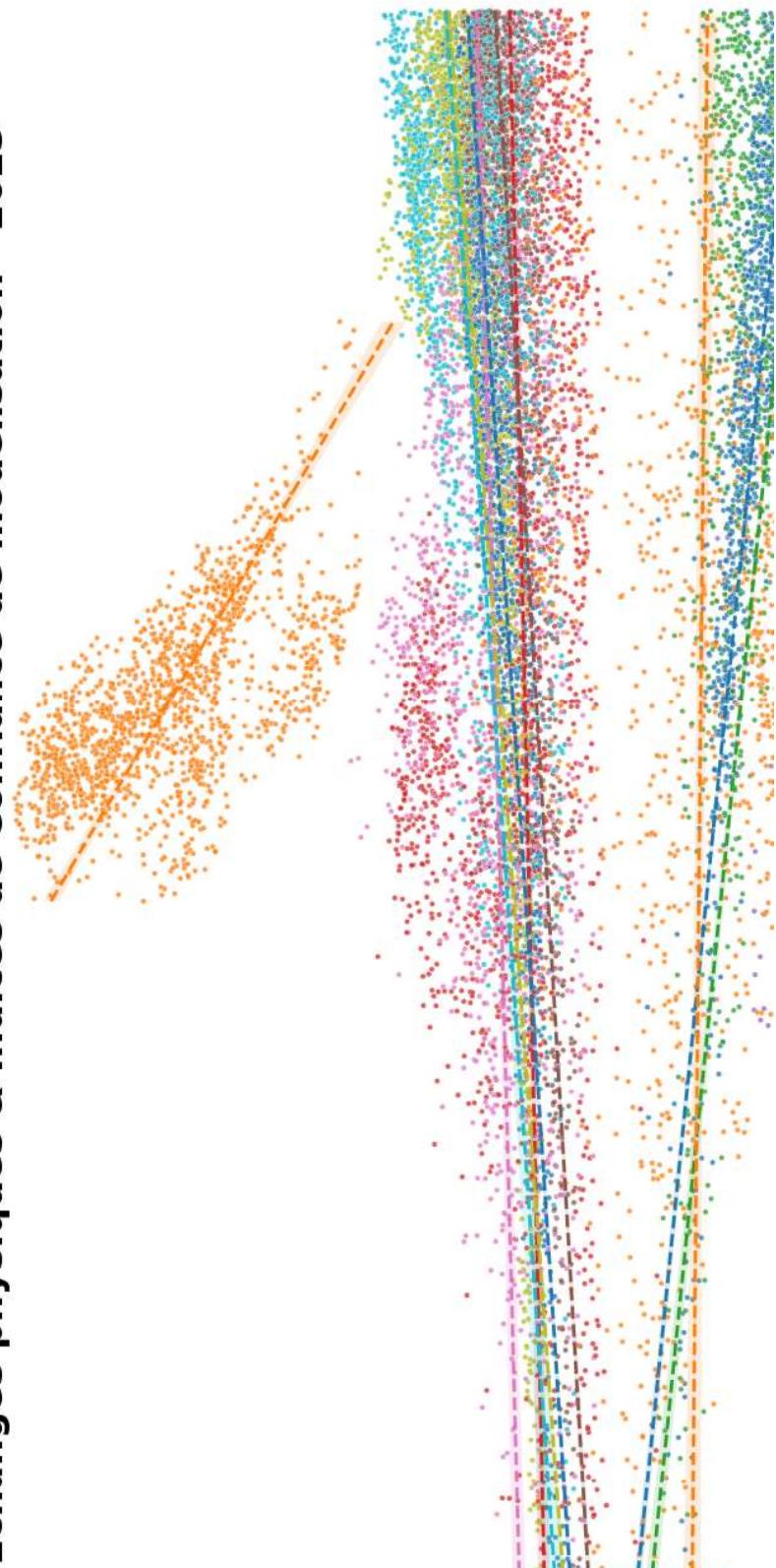
## Echanges physiques & Indices de Confiance de modélisation - 2023

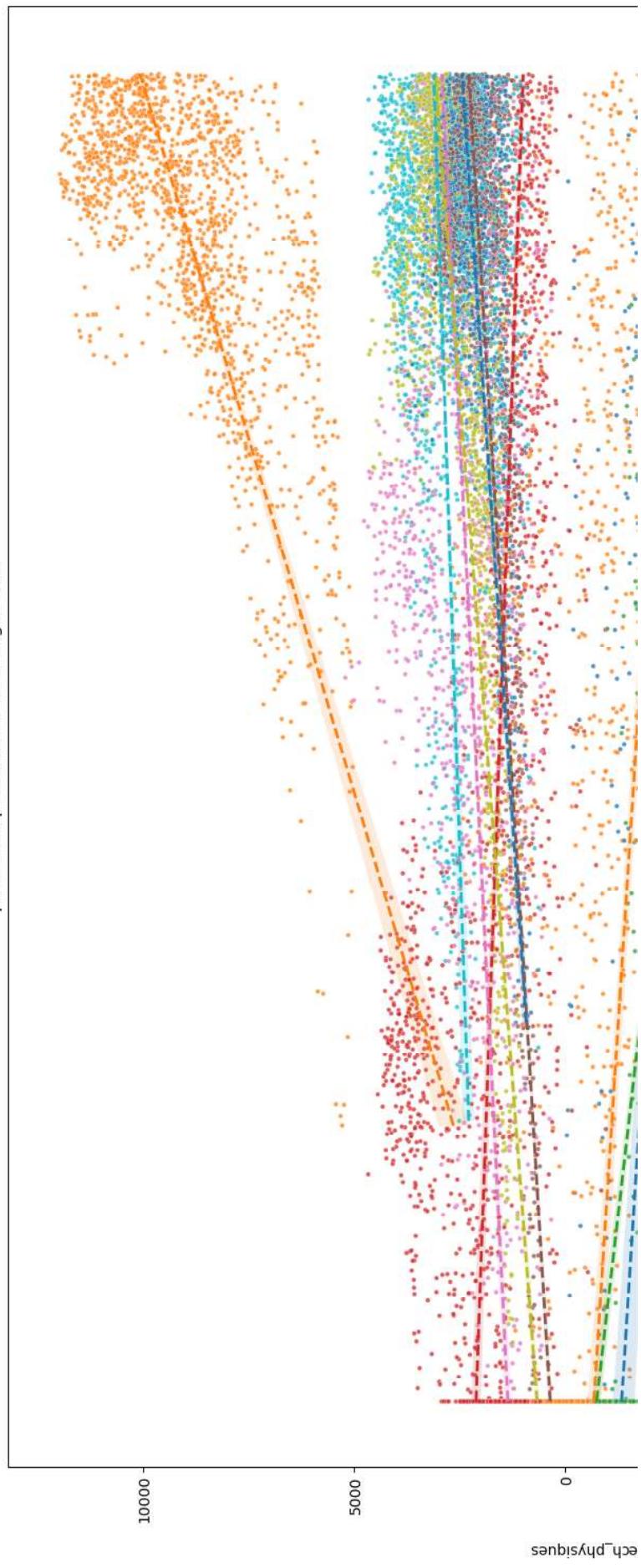
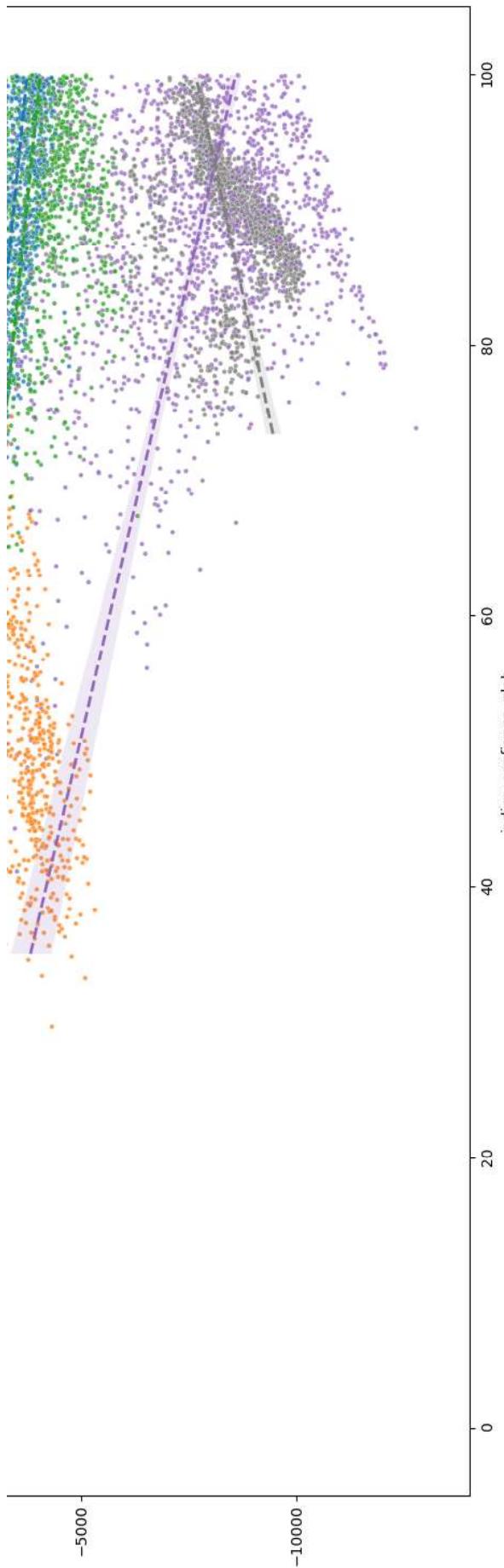
10000

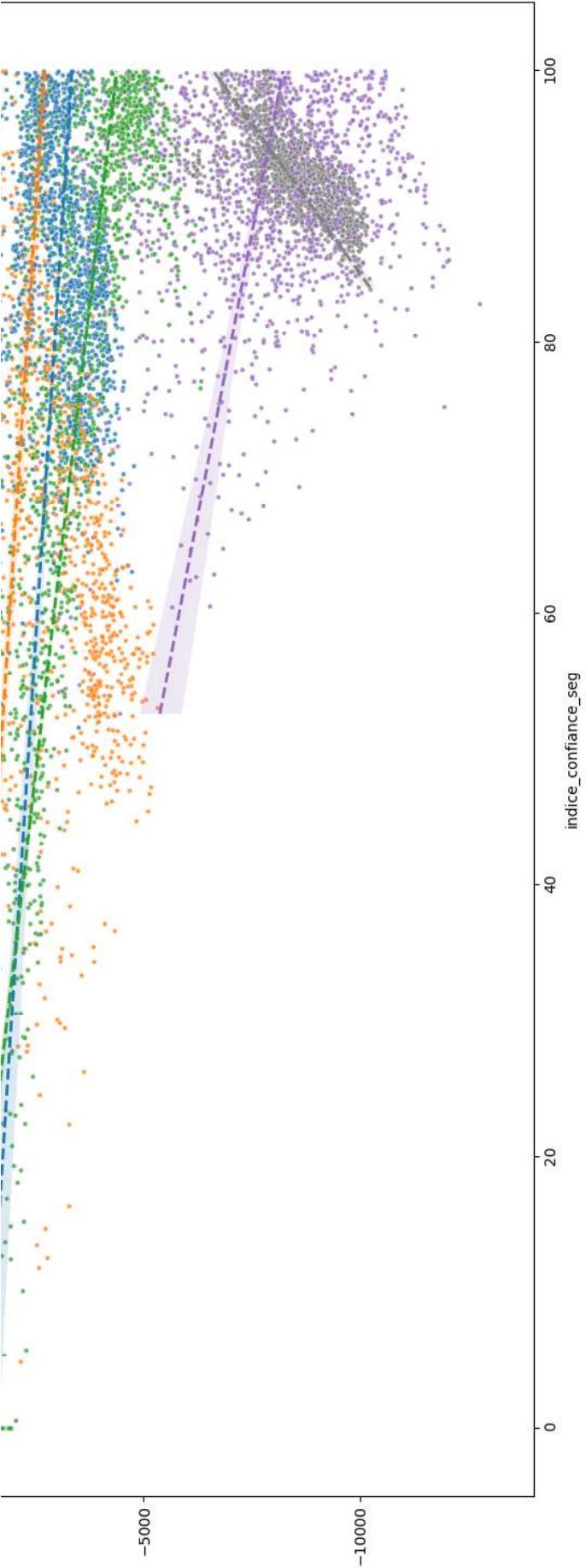
5000

0

éch\_physiques







```
In [45]: # Nuages de points de toutes les valeurs 'echanges physiques' réelles en fonction des indices de confiance, avec affichage des tendances -Lissage LOWESS
```

```
palette = sns.color_palette('tab10', n_colors=df_blackout_xgb_20_23['region'].nunique())
annees = df_blackout_xgb_20_23['annee'].unique()
```

```
for annnee in annees:
```

```
df_annee = df_blackout_xgb_20_23[df_blackout_xgb_20_23['annee'] == annnee]
```

```
fig, axes = plt.subplots(nrows=2, ncols=1, figsize=(15, 24))
```

```
regions = df_annee['region'].unique()
```

```
# Premier sous-graphique avec régression polynomiale (ordre 2)
scatter_0 = sns.scatterplot(ax=axes[0], data=df_annee, x='indice_confiance_glob', y='ech_physiques',
hue='region', palette=palette, legend=False, s=10, alpha=0.6)
axes[0].set_title("Par modélisation globale")
axes[0].set_xlabel('Indice de Confiance glob. (%)')
axes[0].set_ylabel('Echanges Physiques réels (MW)')
```

```
# Courbes de tendance polynomiale par région
for region in regions:
```

```
df_region = df_annee[df_annee['region'] == region]
sns.regplot(ax=axes[0], data=df_region, x='indice_confiance_glob', y='ech_physiques',
scatter=False, color=palette[regions.tolist().index(region)],
```

```

line_kws={'linestyle': '--', 'linewidth': 2, lowess=True) # Lissage LOWESS

# Deuxième sous-graphique avec LOWESS
scatter_1 = sns.scatterplot(ax=axes[1], data=df_annee, x='indice_confiance_seg', y='ech_physiques',
hue='region', palette=palette, legend=False, s=10, alpha=0.6)
axes[1].set_title('Par modélisations régionales')
axes[1].set_xlabel('Indice de Confiance reg. (%)')
axes[1].set_ylabel('Échanges Physiques réels (MW)')

# Courbes de tendance LOWESS par région
for region in regions:
    df_region = df_annee[df_annee['region'] == region]
    sns.regplot(ax=axes[1], data=df_region, x='indice_confiance_seg', y='ech_physiques',
    scatter=False, color=palette[regions.tolist().index(region)],
    line_kws={'linestyle': '--', 'linewidth': 2}, lowess=True) # Lissage LOWESS

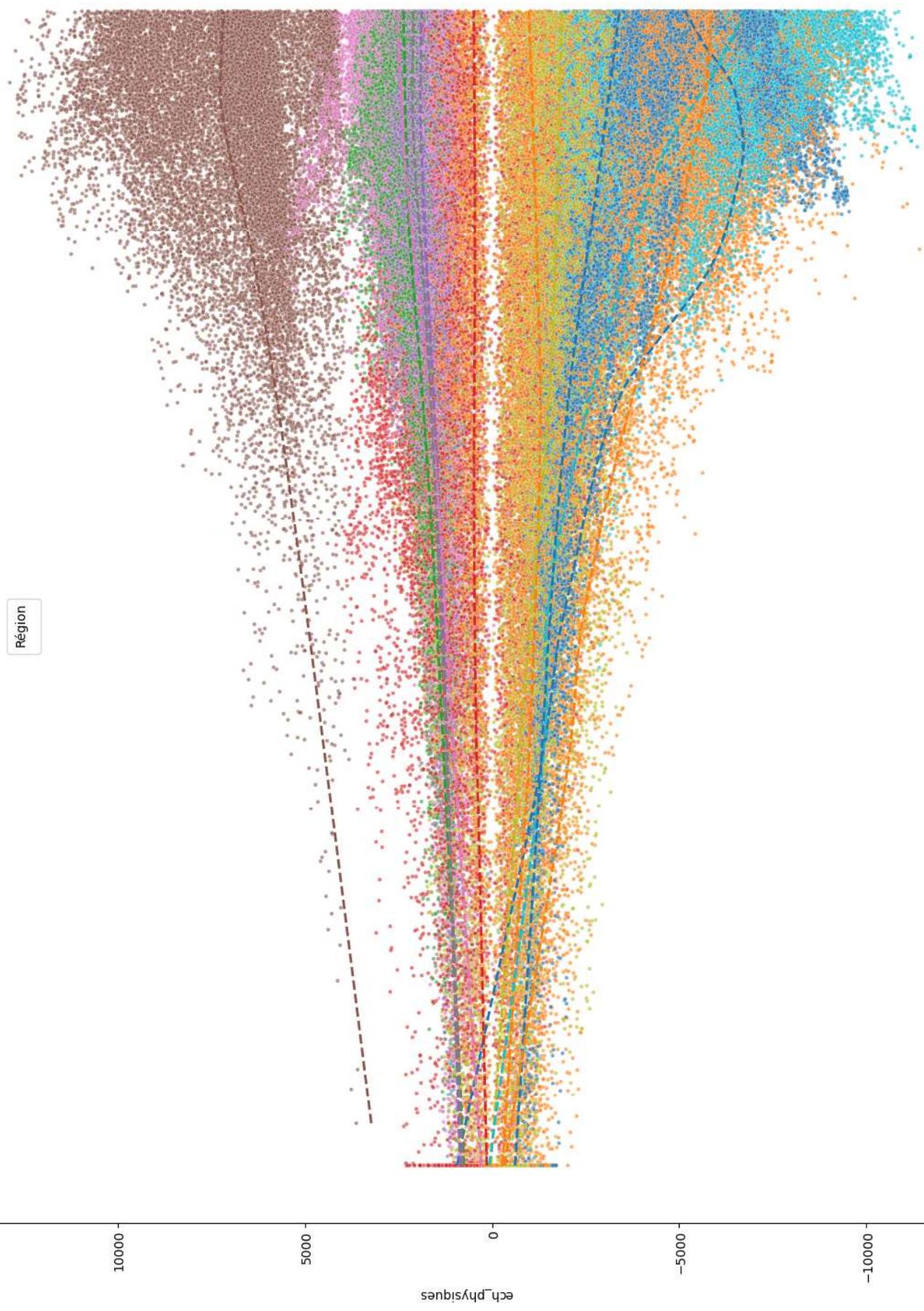
# Ajout de la légende commune en utilisant les handles et labels du premier graphique
handles, labels = scatter_0.get_legend_handles_labels()
fig.legend(handles, labels, loc='upper center', title='Région', bbox_to_anchor=(0.5, 0.95), ncol=len(regions), fontsize='large')

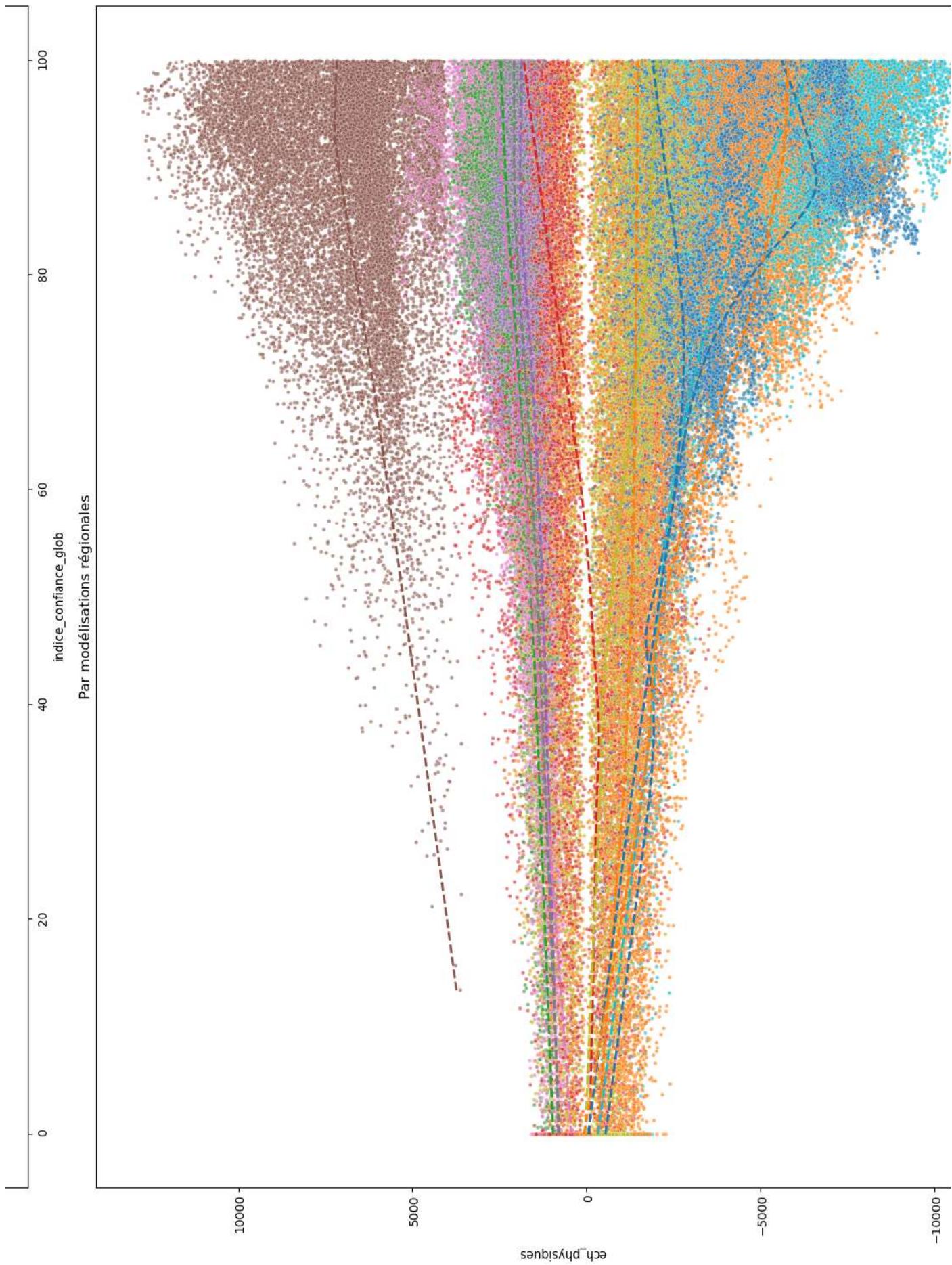
# Titre général
fig.suptitle(f"Échanges physiques & Indices de Confiance de modélisation + LOWESS - {annee}", fontsize=22, weight='bold', y=0.98)

plt.tight_layout()
plt.savefig(f'scatters_ech_physiques_IC_Tendances_LLW_{annee}.png', dpi=300)
plt.show()

```

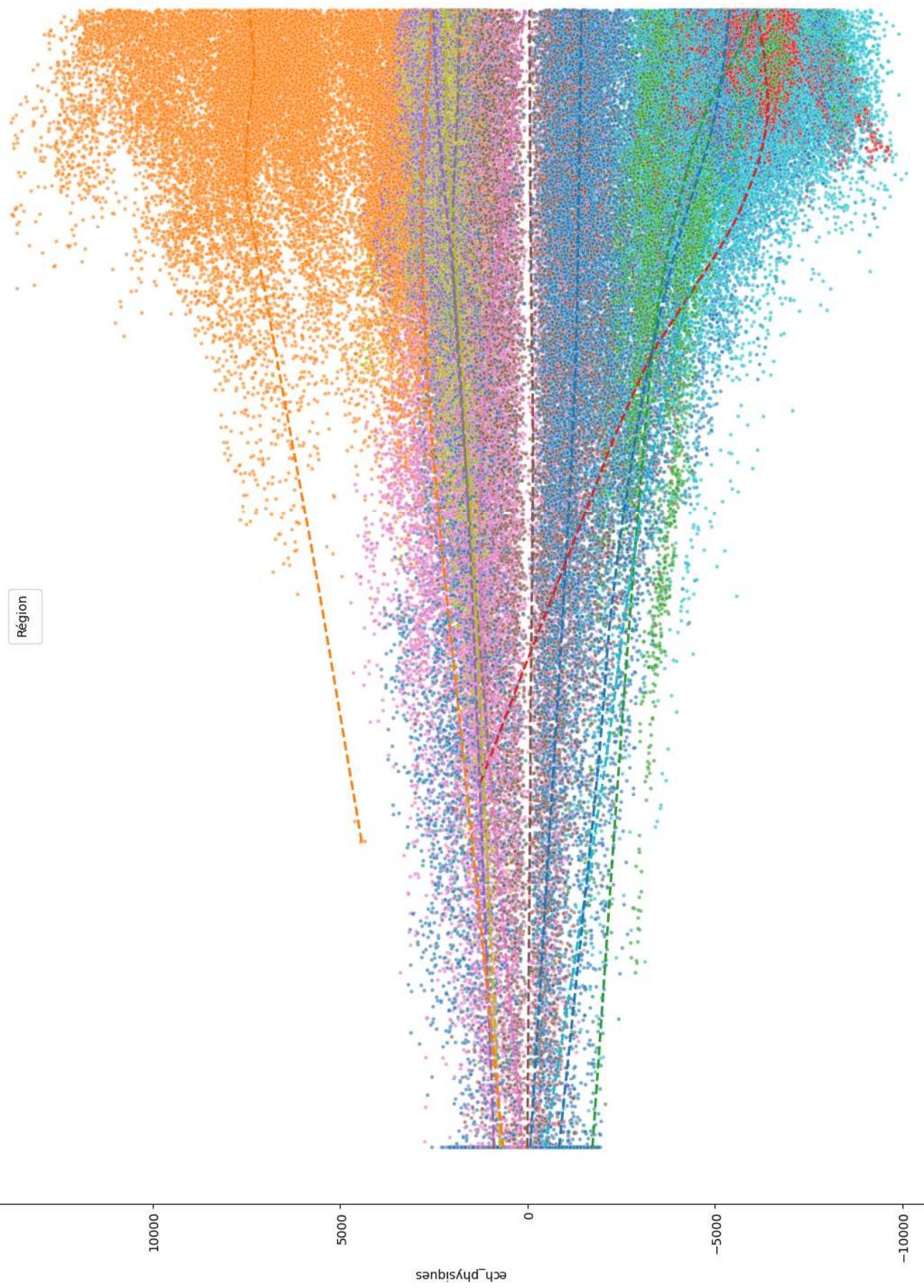
## Echanges physiques & Indices de Confiance de modélisation + LOWESS - 2020

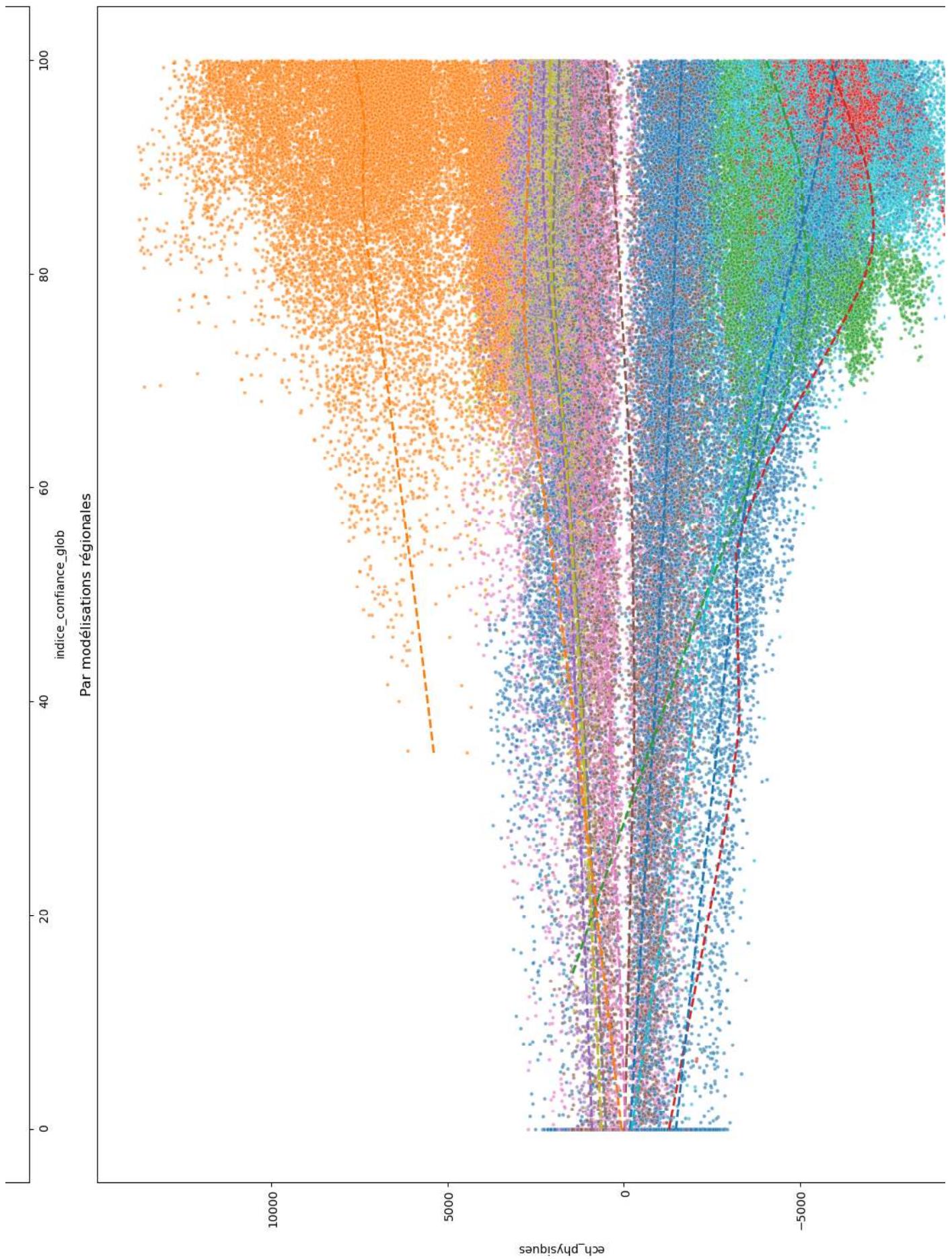






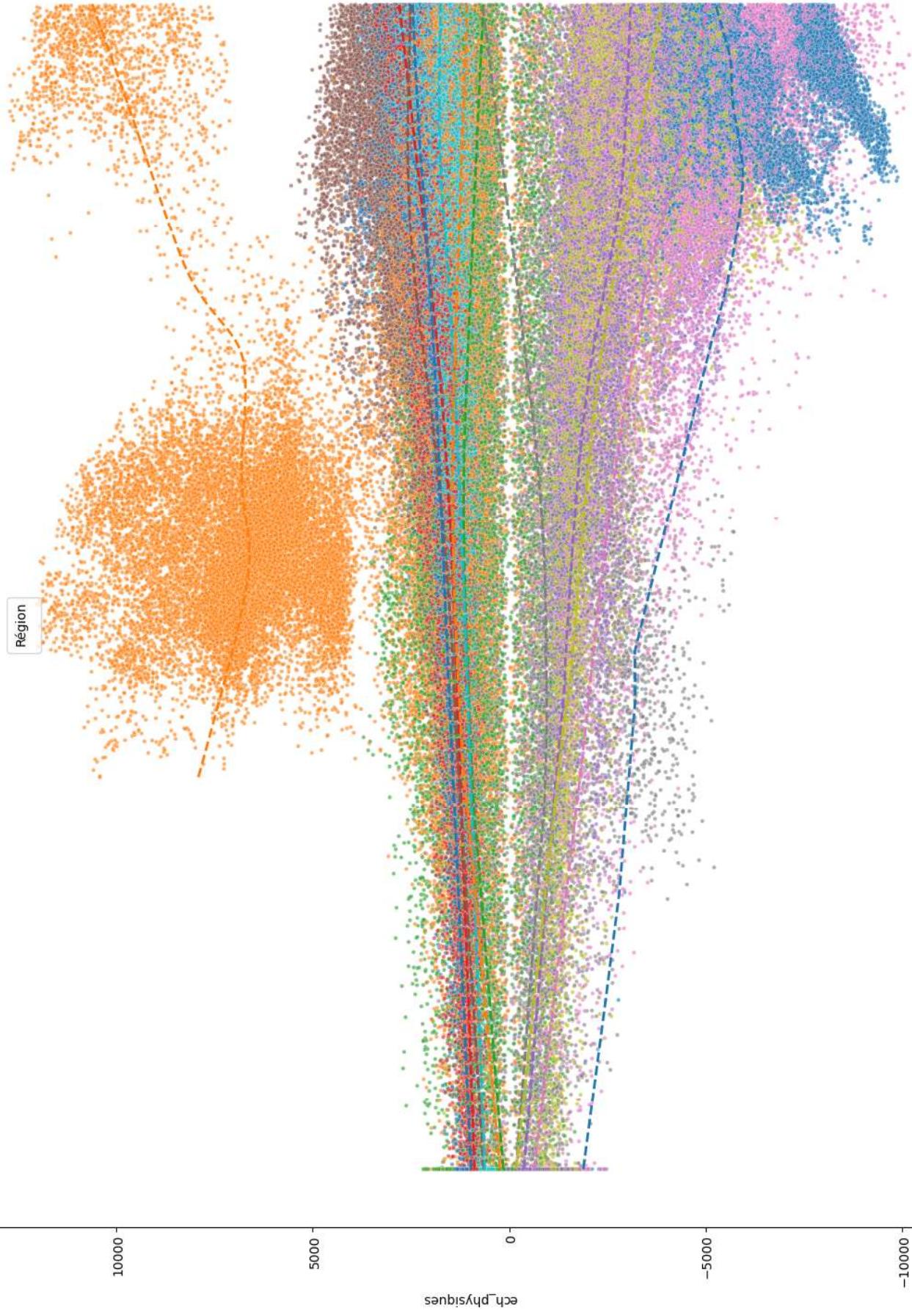
## Echanges physiques & Indices de Confiance de modélisation + LOWESS - 2021

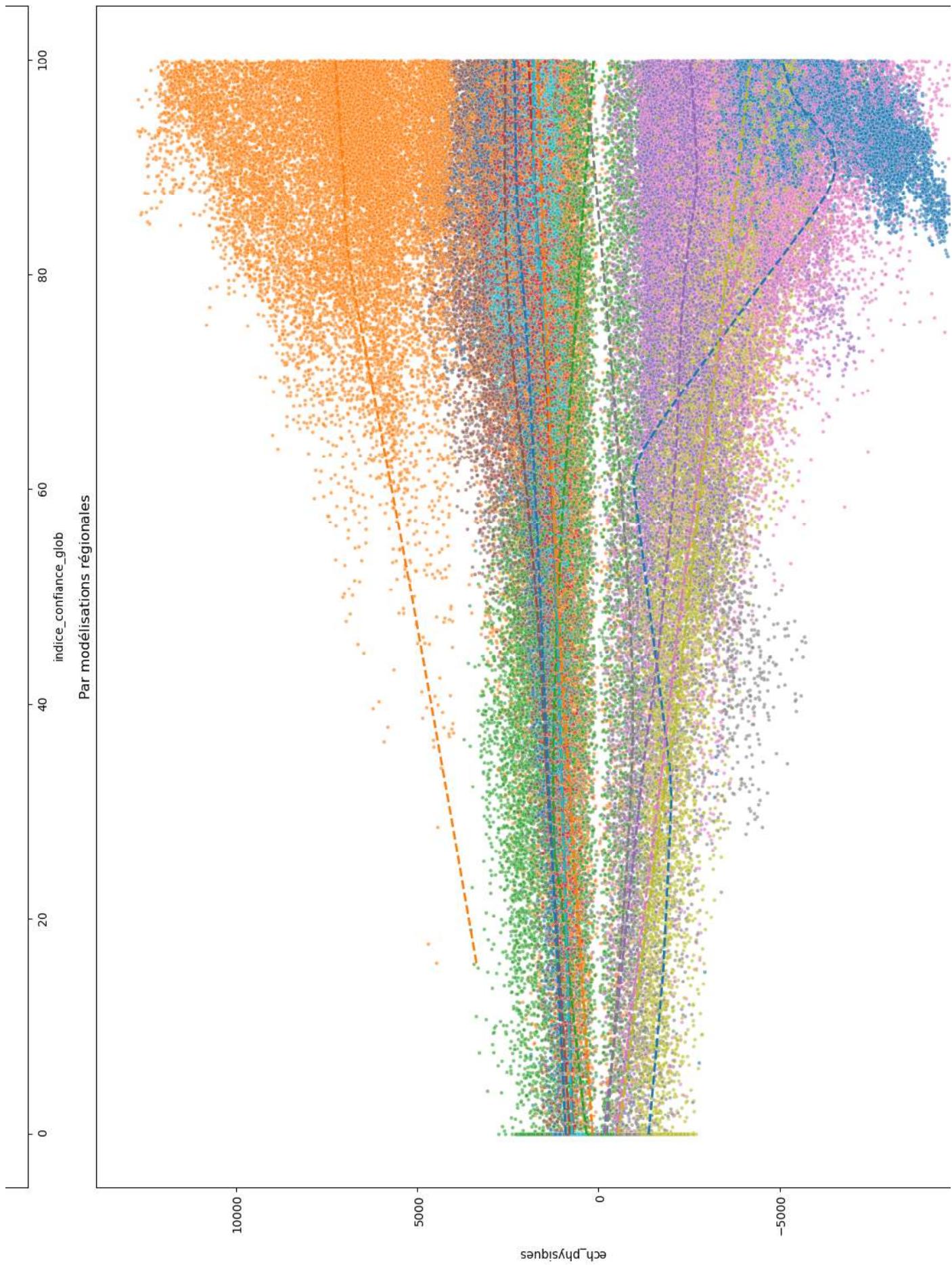






## Echanges physiques & Indices de Confiance de modélisation + LOWESS - 2022

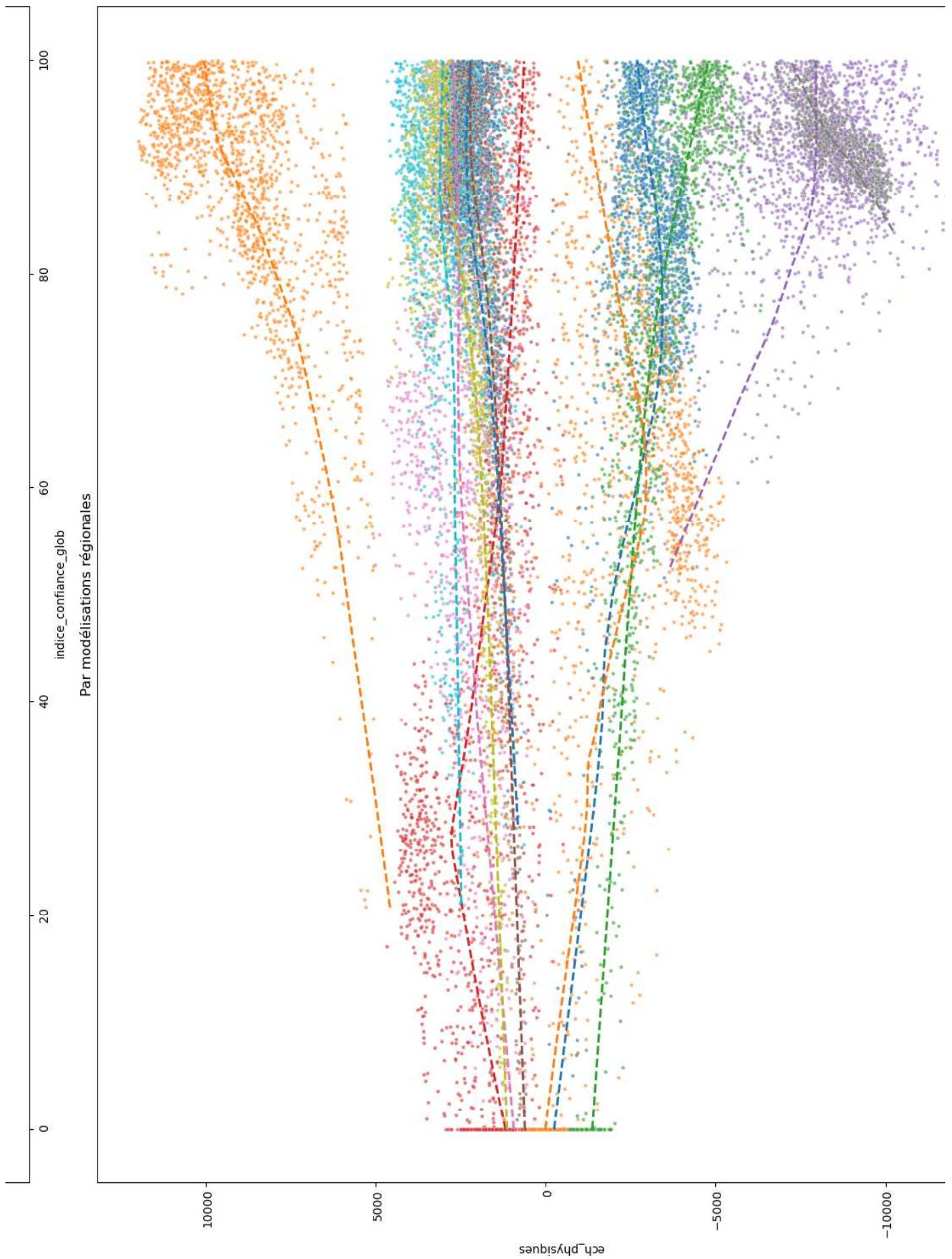






## Echanges physiques & Indices de Confiance de modélisation + LOWESS - 2023





The figure is a histogram titled "Histogrammes des bilans annuels des échanges physiques réels et prédictifs selon les 2 méthodes". The x-axis is labeled "indice\_confiance\_seg" and ranges from 0 to 100. The y-axis represents frequency. There are two main distributions: one centered around 50 for the 'ech\_physiques' method (red bars), and another centered around 70 for the 'ech\_physiques\_pred\_global' method (blue bars). A vertical dashed line is drawn at approximately 65.

```

In [47]: # ****Histogrammes des bilans annuels des échanges physiques réels et prédictifs selon les 2 méthodes ****

In [48]: # Calcul des sommes par année et région
somme_values = df_blackout_xgb_20_23.groupby(['region', 'année']).[['ech_physiques', 'ech_physiques_pred_global', 'ech_physiques_pred_seg']].sum().reset_index()

# Transformation des valeurs de MW en TWh (on divise par 2 car les valeurs sont relevées toutes les 0.5h et on divise 10E6 pour obtenir les TWh)
somme_values[['ech_physiques', 'ech_physiques_pred_global', 'ech_physiques_pred_seg']] /= 1_000_000/2

# Couleurs des barres
colors = ['red', 'yellow', 'blue']

# Iteration sur les années pour créer un graphique par année
for année in somme_values['année'].unique():

    # Filtrage des données pour l'année en cours
    df_annee = somme_values[somme_values['année'] == année]
    regions = df_annee['region'].unique()

    # Création de la figure et des axes
    fig, ax = plt.subplots(figsize=(12, 8))

    # Paramètres pour les barres
    bar_width = 0.25 # Largeur des barres
    index = np.arange(len(regions)) # Position des groupes sur l'axe y

    # Création des barres côté à côté
    ax.bart(index, df_annee[['ech_physiques']], bar_width, color=colors[0], label='Ech. physiques')
    ax.bart(index + bar_width, df_annee[['ech_physiques_pred_global']], bar_width, color=colors[1], label='Ech. physiques préd. global')
    ax.bart(index + 2 * bar_width, df_annee[['ech_physiques_pred_seg']], bar_width, color=colors[2], label='Ech. physiques préd. segmenté')

    # Ajout de traits de séparation entre les régions
    for i in range(len(regions) - 1):
        ax.axhline(y=i + 0.75, color='gray', linestyle='--', linewidth=0.5)

    # Affichage des valeurs au bout de chaque barre 'ech_physiques' avec ajustement des positions
    for i, v in enumerate(df_annee['ech_physiques']):
        # Décalage horizontal en fonction de la valeur (positif ou négatif)
        horizontal_offset = 0.7 if v > 0 else -30
        ax.text(v + horizontal_offset, i, f'{v:.0f}', color='black', va='center', fontsize=8)

# Affichage du pourcentage de différence pour 'ech_physiques_pred_global' avec décalage
for i, (v_real, v_pred_global) in enumerate(zip(df_annee['ech_physiques'], df_annee['ech_physiques_pred_global'])):
    percentage_diff_global = (v_pred_global - v_real) / v_real * 100
    horizontal_offset_global = 0.9 if v_pred_global > 0 else -38

```

```

ax.text(v_pred_global + horizontal_offset_global, i + bar_width, f'{percentage_diff_global:.1f}%', color='black', va='center', fontsize=8)

# Affichage du pourcentage de différence pour 'ech_physiques_pred_seg' avec décalage
for i, (v_real, v_pred_seg) in enumerate(zip(df_annee['ech_physiques'], df_annee['ech_physiques_pred_seg'])):
    percentage_diff_seg = (v_pred_seg - v_real) / v_real * 100
    horizontal_offset_seg = 0.9 if v_pred_seg > 0 else -38
    ax.text(v_pred_seg + horizontal_offset_seg, i + 2 * bar_width, f'{percentage_diff_seg:.1f}%', color='black', va='center', fontsize=8)

# Configuration des légendes, titres et axes
ax.set_xlabel('Echanges physiques annuels (TWh)')
ax.set_ylabel('Région')
ax.set_title(f'Comparaison des échanges physiques annuels réels / prédicts (XGBoost) par région, {annee}')
ax.set_yticks(index + bar_width)
ax.set_yticklabels(regions)

# Ajustement des limites de l'axe x pour ajouter de l'espace et tenir compte de la plage élargie
ax.set_xlim(-290, df_annee[['ech_physiques', 'ech_physiques_pred_global', 'ech_physiques_pred_seg']].max().max() * 1.3)

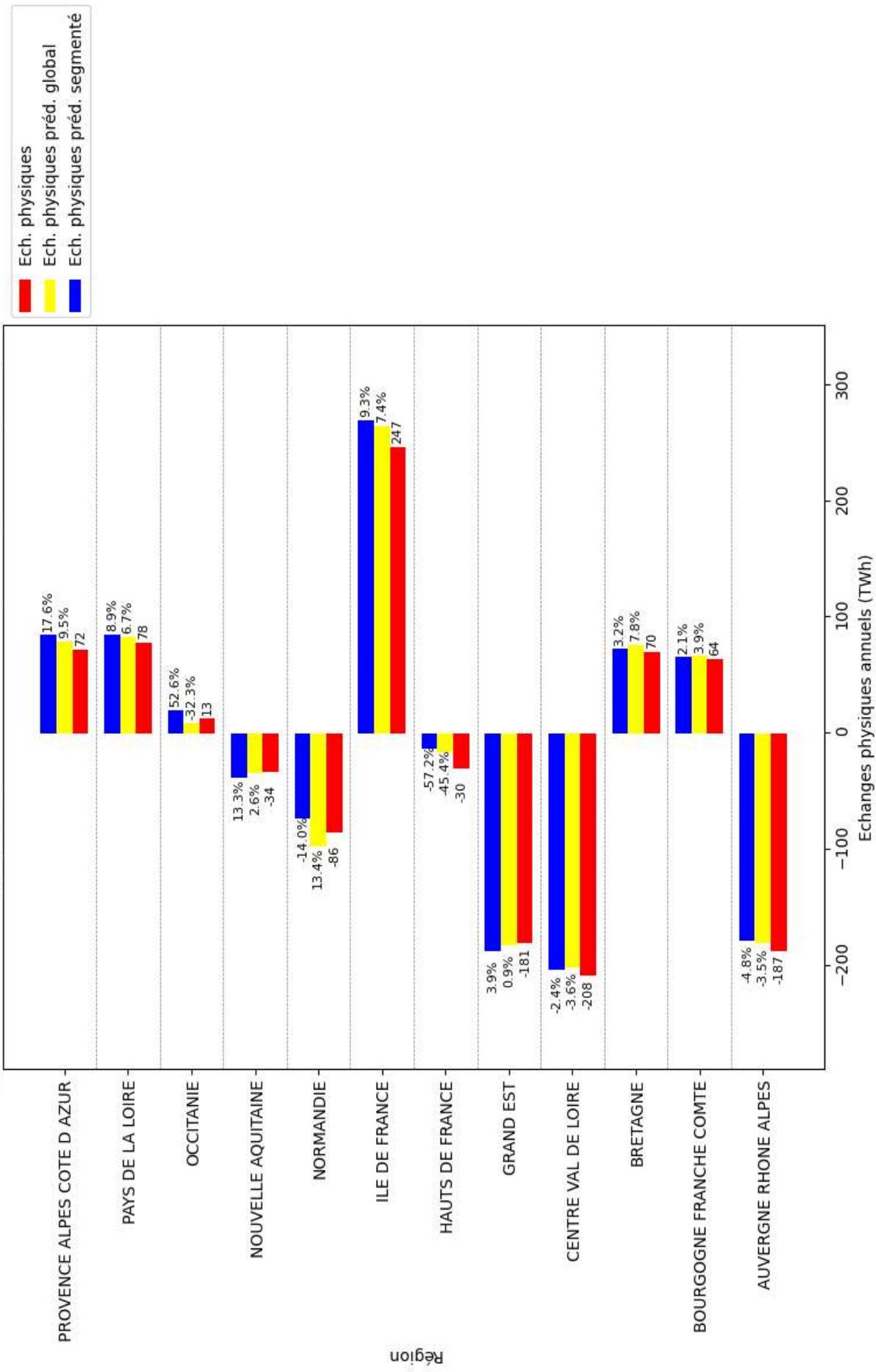
# Ajustement de la légende afin qu'elle ne recouvre pas les graphiques
ax.legend(loc='upper left', bbox_to_anchor=(1, 1))

plt.tight_layout()

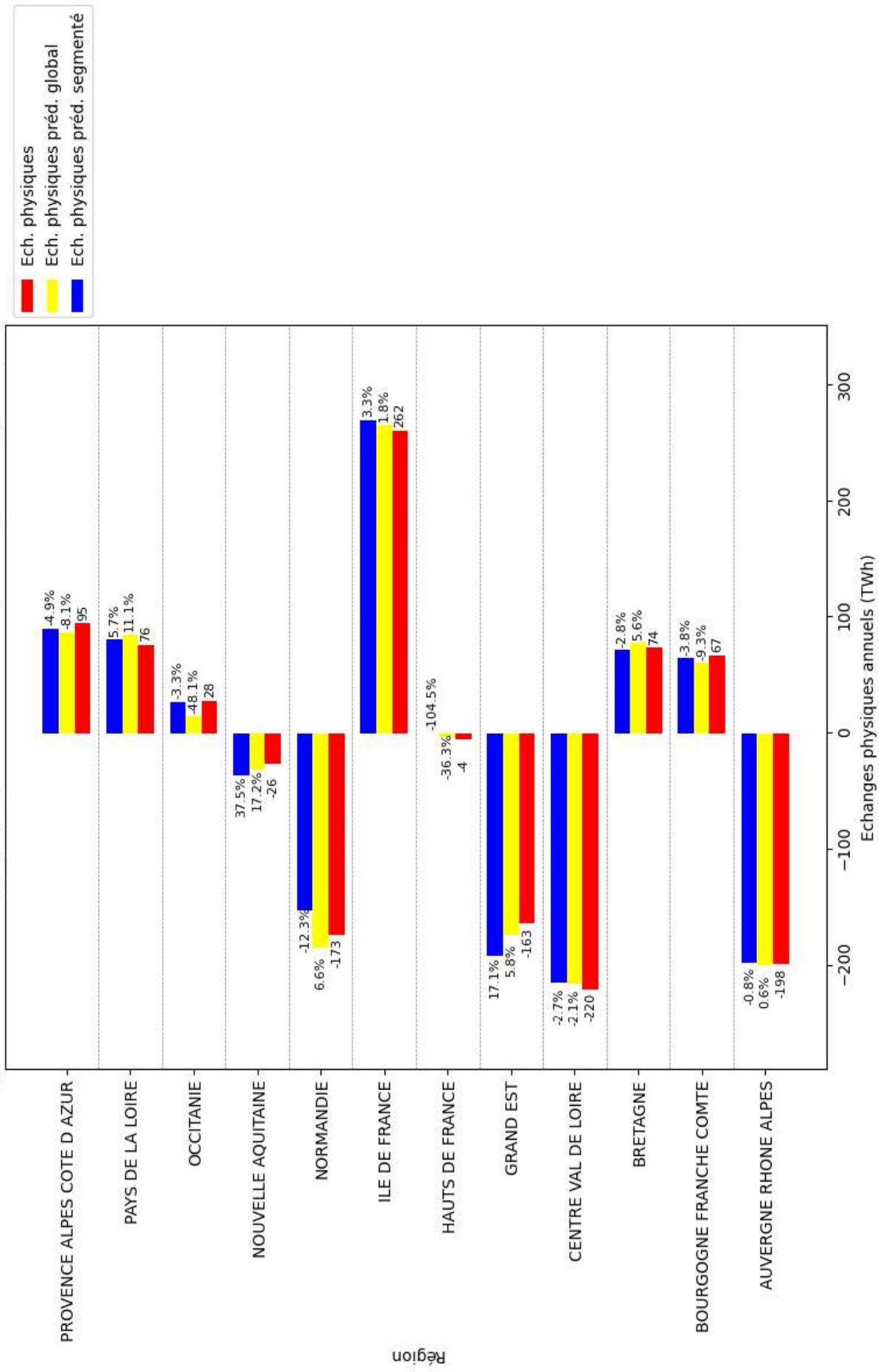
# Sauvegarde du graphique
plt.savefig(f'comparaison_valeurs_region_{annee}.png', dpi=300)
plt.show()

```

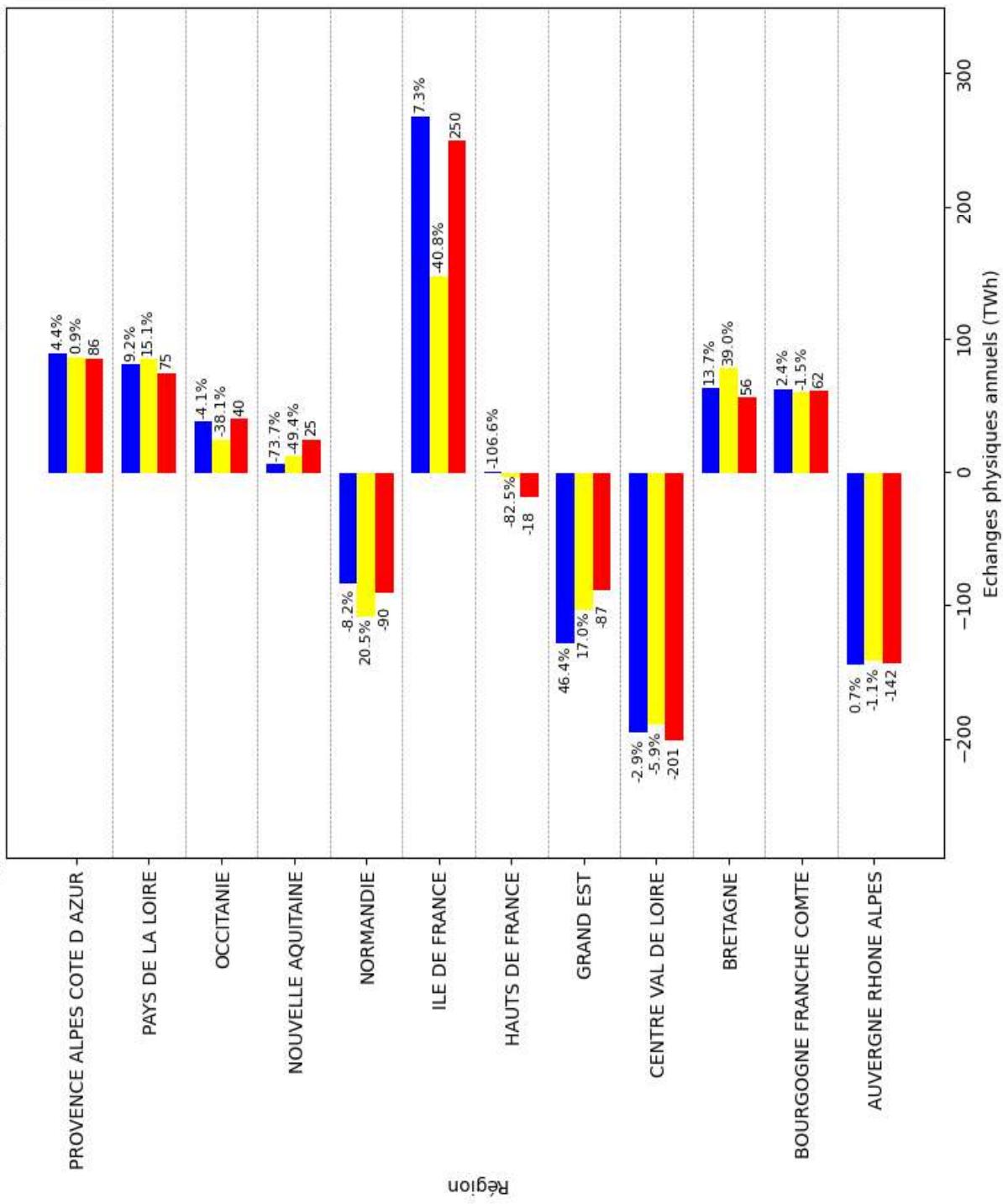
Comparaison des échanges physiques annuels réels / prédicts (XGBoost) par région, 2020



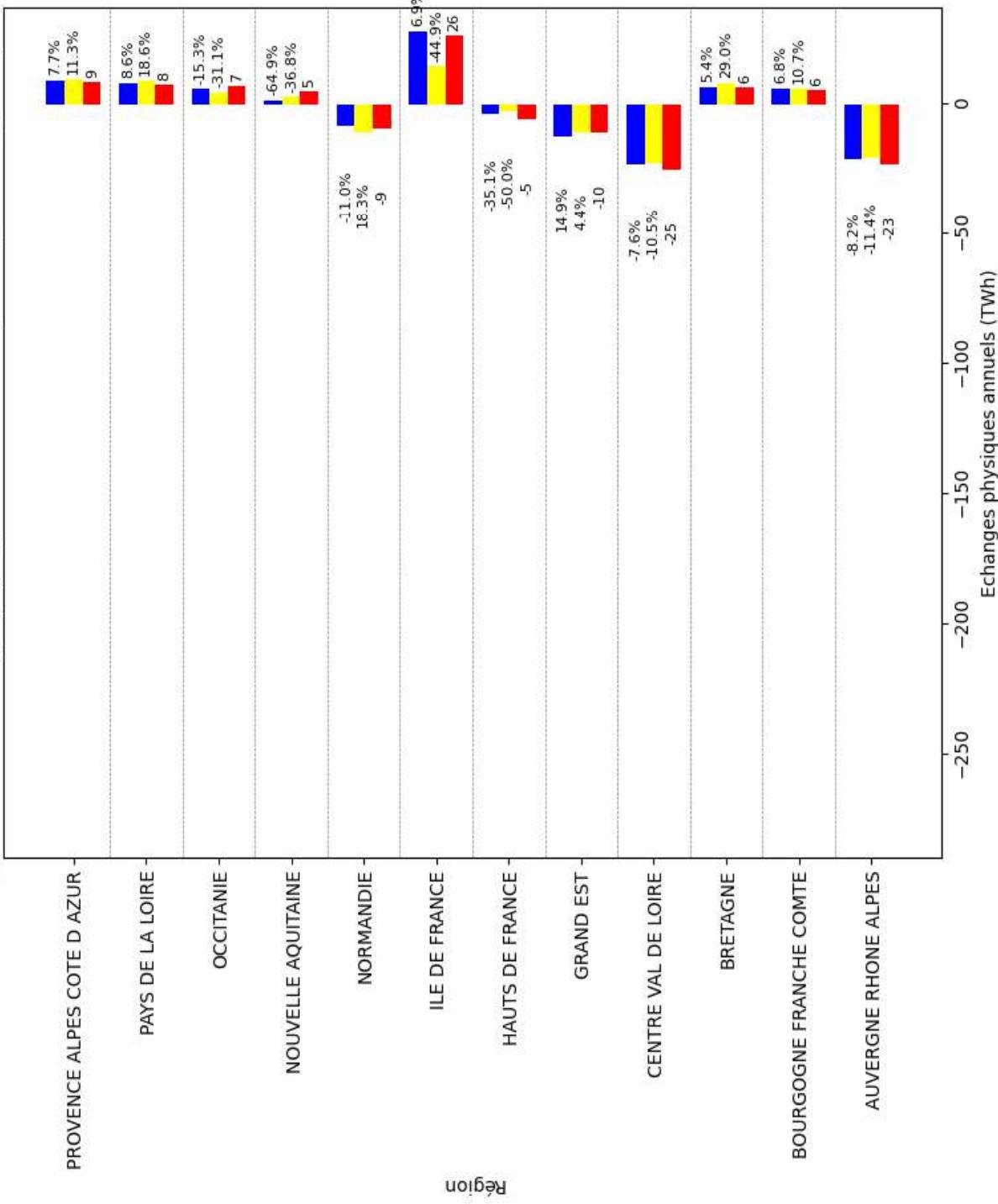
## Comparaison des échanges physiques annuels réels / prédicts (XGBoost) par région, 2021



## Comparaison des échanges physiques annuels réels / prédicts (XGBoost) par région, 2022



Comparaison des échanges physiques annuels réels / prédicts (XGBoost) par région, 2023



In [49]: print('FIN DU CODE')

FIN DU CODE