In [301…
```
# Importer les librairies nécessaires
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
color = ['teal', 'orchid','#a7a7a7','#aec7e8','#9467bd','#EDE862']
```

Traitement de données

In [302…
```
# Decouvertes de données
dossier_actuel = os.getcwd()
chemin_fichier = os.path.join(dossier_actuel, 'HR-employee-db.csv')
data = pd.read_csv(chemin_fichier) # Lire le fichier CSV
```

In [303…
```
# Afficher la table
data.head()
```

Out[303]:

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Educat
0	41	Yes	Travel_Rarely	1102	Sales	1	
1	49	No	Travel_Frequently	279	Research & Development	8	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	
4	27	No	Travel_Rarely	591	Research & Development	2	

5 rows × 35 columns

Suppression des colonnes indésirables

In [304…
```
data.drop(['EmployeeCount','EmployeeNumber','StandardHours','Over18'],axis=1,inp
```

In [305…
```
# Comprendre les détails statistiques de notre base de données
data.describe().style.background_gradient(cmap='viridis')
```

Out[305]:

	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisfact
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.000
mean	36.923810	802.485714	9.192517	2.912925	2.721
std	9.135373	403.509100	8.106864	1.024165	1.093
min	18.000000	102.000000	1.000000	1.000000	1.000
25%	30.000000	465.000000	2.000000	2.000000	2.000
50%	36.000000	802.000000	7.000000	3.000000	3.000
75%	43.000000	1157.000000	14.000000	4.000000	4.000
max	60.000000	1499.000000	29.000000	5.000000	4.000

In [323…

```
# Verification des colones numériques
numeric = data.select_dtypes(exclude='O')
numeric.head()
```

Out[323]:

	Age	DailyRate	DistanceFromHome	Education	EnvironmentSatisfaction	HourlyRate
0	41	1102	1	2	2	94
1	49	279	8	1	3	61
2	37	1373	2	2	4	92
3	33	1392	3	4	4	56
4	27	591	2	1	1	40

5 rows × 23 columns

In [324…

```
# Verification des colonnes types objects
obj = data.select_dtypes(include='O')
obj.head()
```

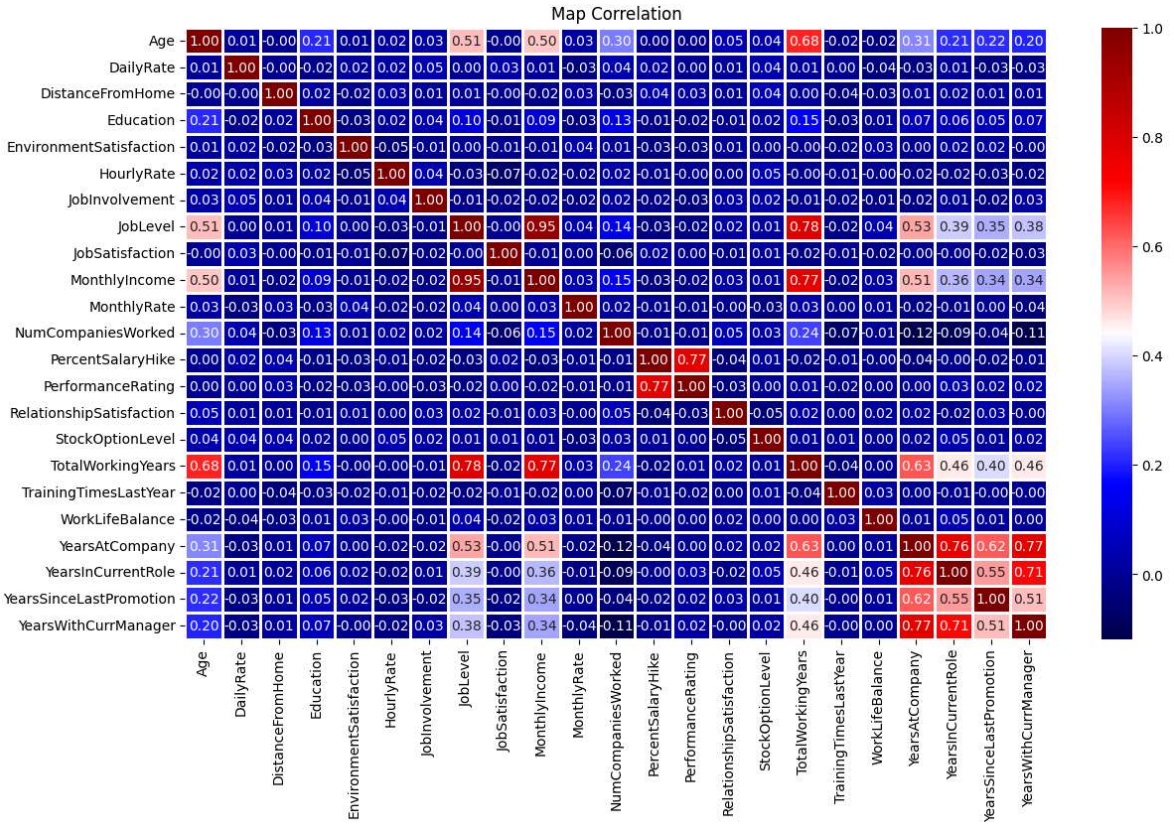
Out[324]:

	Attrition	BusinessTravel	Department	EducationField	Gender	JobRole	Marital
0	Yes	Travel_Rarely	Sales	Life Sciences	Female	Sales Executive	
1	No	Travel_Frequently	Research & Development	Life Sciences	Male	Research Scientist	M
2	Yes	Travel_Rarely	Research & Development	Other	Male	Laboratory Technician	
3	No	Travel_Frequently	Research & Development	Life Sciences	Female	Research Scientist	M
4	No	Travel_Rarely	Research & Development	Medical	Male	Laboratory Technician	M

In [325…
```
# Affiche les structures des tables
print(data.shape)
print(numeric.shape)
print(obj.shape)
```

```
(1470, 31)
(1470, 23)
(1470, 8)
```

In [327…
```
# Verification des corrélation matrix
plt.figure(figsize=(14,8))
sns.heatmap(numeric.corr(),annot=True,cmap='seismic',fmt='.2f',linewidths=1)
plt.title('Map Correlation')
plt.show()
```



Nettoyage de données

Vérification des valeurs null dans le datasets

In [330…
```
# Verification des valeurs null
print(data.shape)
print(data.duplicated().sum()) # Verifier les doublons

table_info = pd.DataFrame({
    'Unique':data.nunique(),
    'Null':data.isna().sum(),
    'NullPercent':data.isna().sum() / len(data),
    'Type':data.dtypes.values
})
print(table_info)
```

```
(1470, 31)
0
```

	Unique	Null	NullPercent	Type
Age	43	0	0.0	int64
Attrition	2	0	0.0	object
BusinessTravel	3	0	0.0	object
DailyRate	886	0	0.0	int64
Department	3	0	0.0	object
DistanceFromHome	29	0	0.0	int64
Education	5	0	0.0	int64
EducationField	6	0	0.0	object
EnvironmentSatisfaction	4	0	0.0	int64
Gender	2	0	0.0	object
HourlyRate	71	0	0.0	int64
JobInvolvement	4	0	0.0	int64
JobLevel	5	0	0.0	int64
JobRole	9	0	0.0	object
JobSatisfaction	4	0	0.0	int64
MaritalStatus	3	0	0.0	object
MonthlyIncome	1349	0	0.0	int64
MonthlyRate	1427	0	0.0	int64
NumCompaniesWorked	10	0	0.0	int64
OverTime	2	0	0.0	object
PercentSalaryHike	15	0	0.0	int64
PerformanceRating	2	0	0.0	int64
RelationshipSatisfaction	4	0	0.0	int64
StockOptionLevel	4	0	0.0	int64
TotalWorkingYears	40	0	0.0	int64
TrainingTimesLastYear	7	0	0.0	int64
WorkLifeBalance	4	0	0.0	int64
YearsAtCompany	37	0	0.0	int64
YearsInCurrentRole	19	0	0.0	int64
YearsSinceLastPromotion	16	0	0.0	int64
YearsWithCurrManager	18	0	0.0	int64

In [328…
```
# Autres methodes pour verifier les valeurs null
valeurs_nul=data.isna().sum()
total=np.product(data.shape)
total_valeur_manquant=null_values.sum()
pourcentage_valeurs_manquant=(total_valeur_manquant/total)*100
print(f'Notre données contients en {percentage_missing_values} % de valeurs null
```

```
Notre données contients en 0.0 % de valeurs null
```

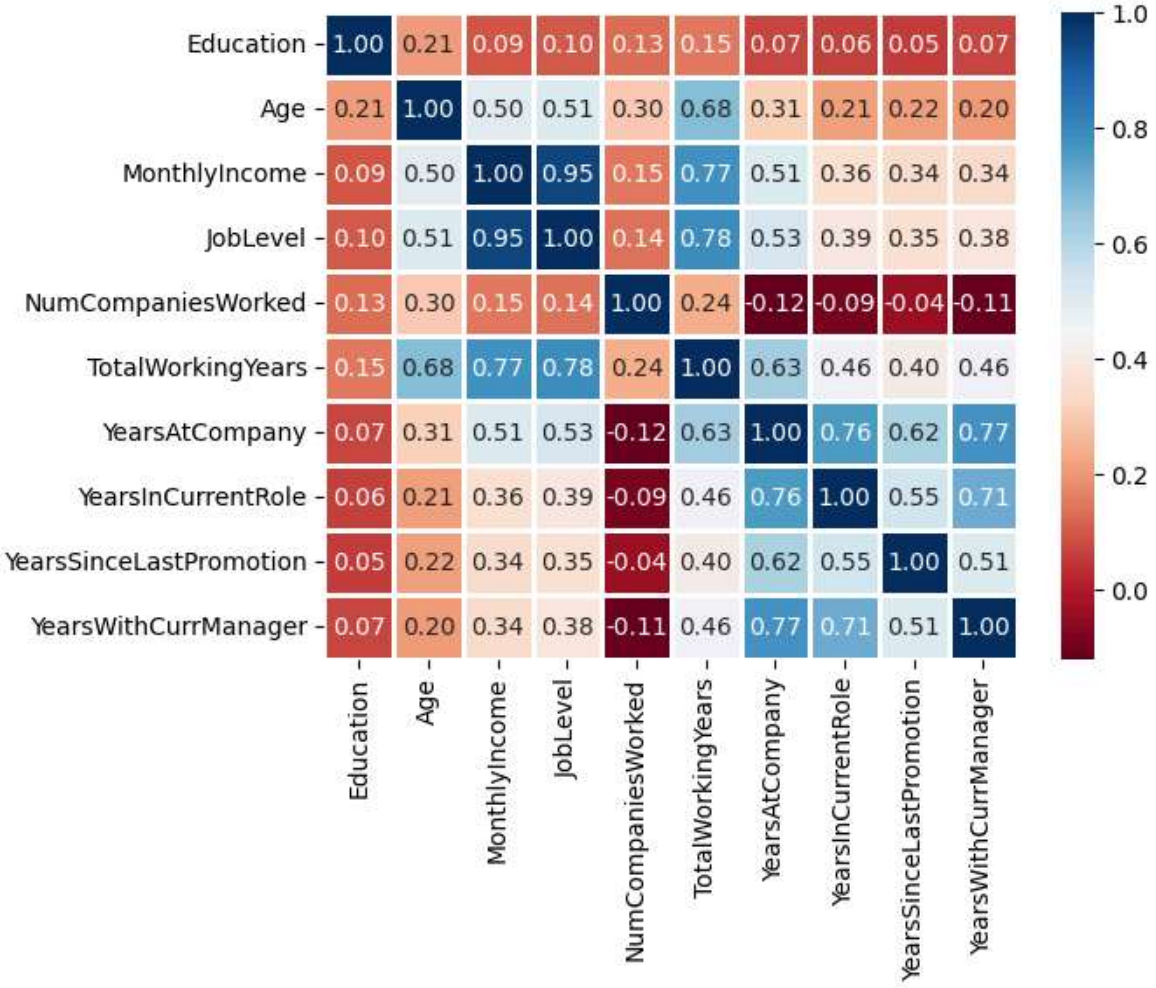
In [329…
```
# Autres methodes pour verifier les doublons
doublons=data.duplicated().sum()
print(f'Ils y a {doublons} velurs dupliqués et nous allons les suprimmer')
```

```
Ils y a 0 velurs dupliqués et nous allons les suprimmer
```

In [ ]:
```
# Au cas ou il en aurra
data=data.drop.duplicated()
data_clean=data.duplicated().sum()
print(f'Ils y a {duplicate} velurs dupliqués et nous allons les suprimmer')
```

In [313…
```
# Verification des corrélation entre les colonnes
colonnes = ['Education','Age','MonthlyIncome','JobLevel','NumCompaniesWorked','T
       'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager']
sns.heatmap(data[colonnes].corr(),annot=True,cmap='RdBu',fmt='.2f',linewidths=1)
```

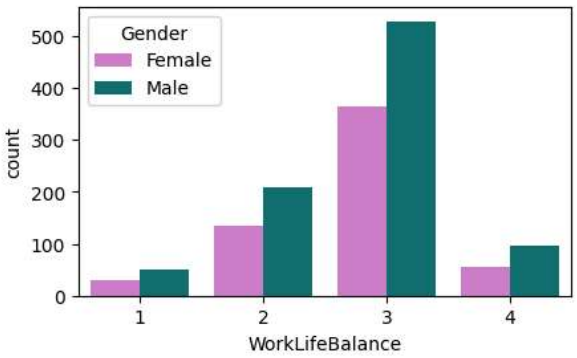
Out[313]: <Axes: >

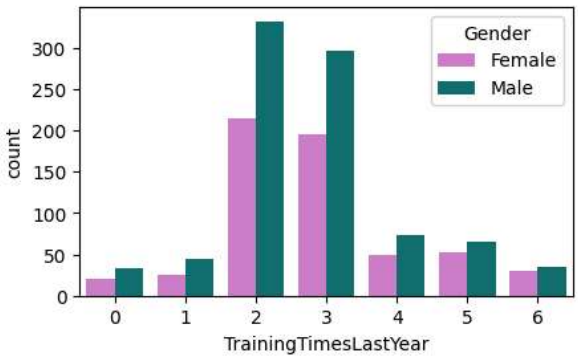


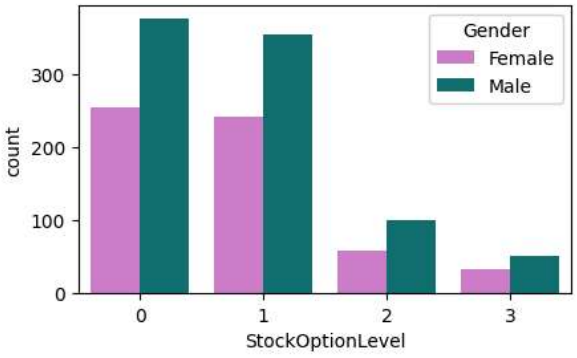
Analyse Exploratoire de Données

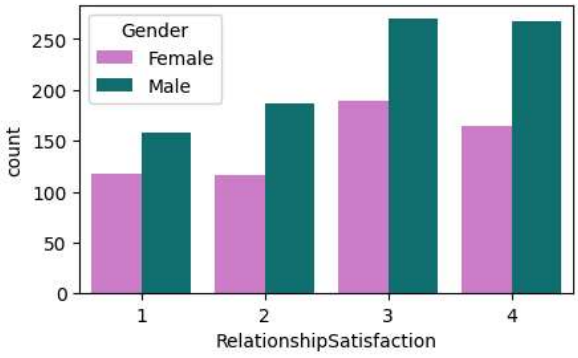
In [314…

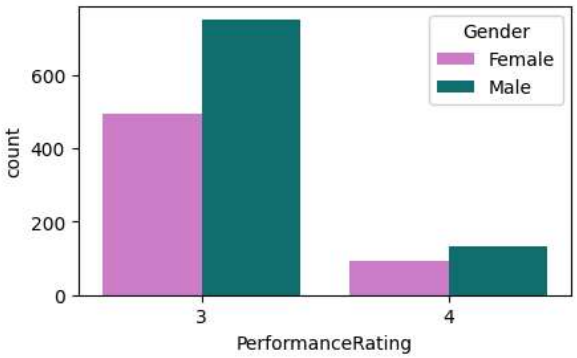
```
colonnes = ['WorkLifeBalance','TrainingTimesLastYear','StockOptionLevel',
    'RelationshipSatisfaction','PerformanceRating','NumCompaniesWorked',
   'JobInvolvement', 'JobLevel', 'JobSatisfaction',
   'EnvironmentSatisfaction','Education']
plt.figure(figsize=(9,36))
for i,col in enumerate(colonnes):
    axes = plt.subplot(13,2, i + 1)
    sns.countplot(x=data[col], hue=data['Gender'], palette=['orchid', 'teal'])
plt.tight_layout()
plt.show()
```

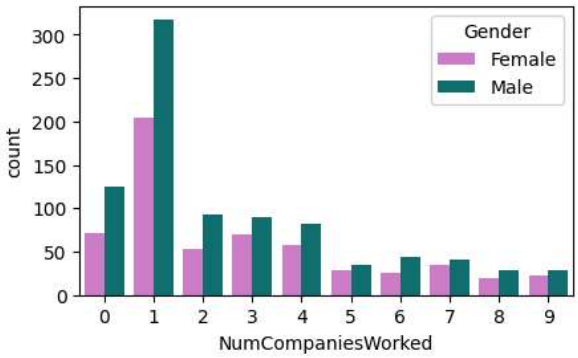

Gender
Female
Male
count
WorkLifeBalance

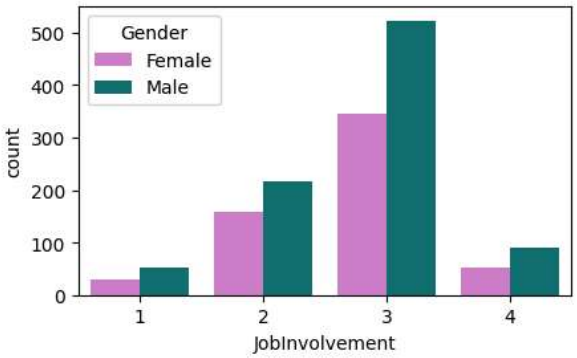

Gender
Female
Male
count
TrainingTimesLastYear

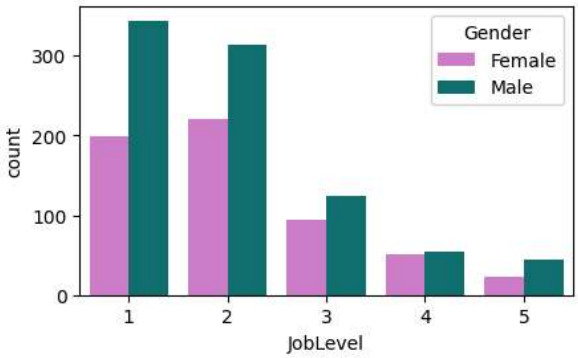

Gender
Female
Male
count
StockOptionLevel

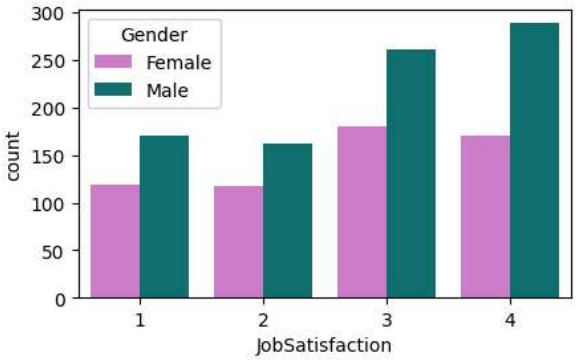

Gender
Female
Male
count
RelationshipSatisfaction

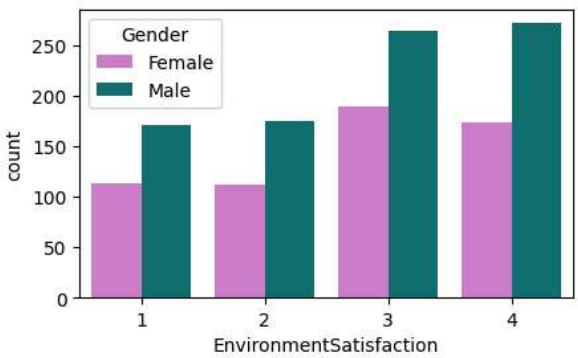

Gender
Female
Male
count
PerformanceRating

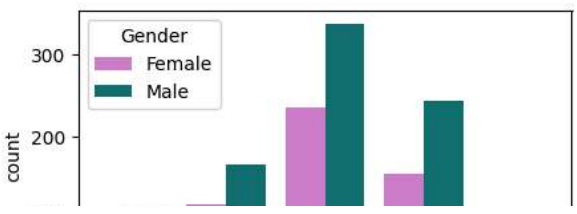

Gender
Female
Male
count
NumCompaniesWorked

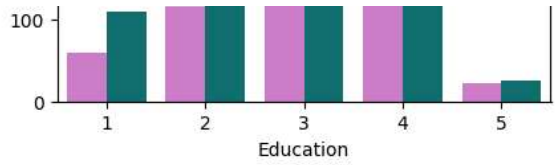

Gender
Female
Male
count
JobInvolvement


Gender
Female
Male
count
JobLevel


Gender
Female
Male
count
JobSatisfaction


Gender
Female
Male
count
EnvironmentSatisfaction


Gender
Female
Male
count



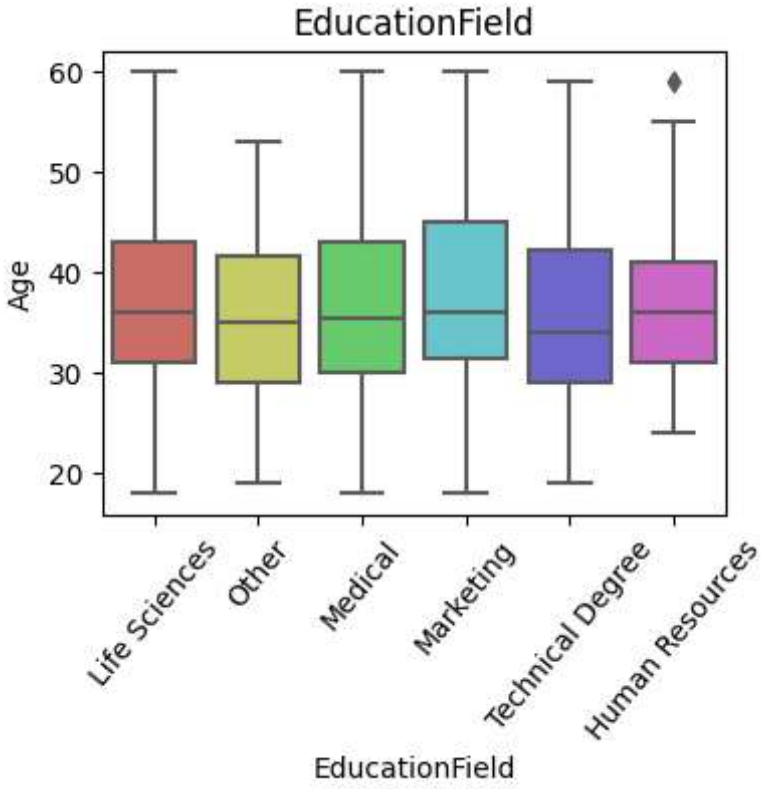
In [315…

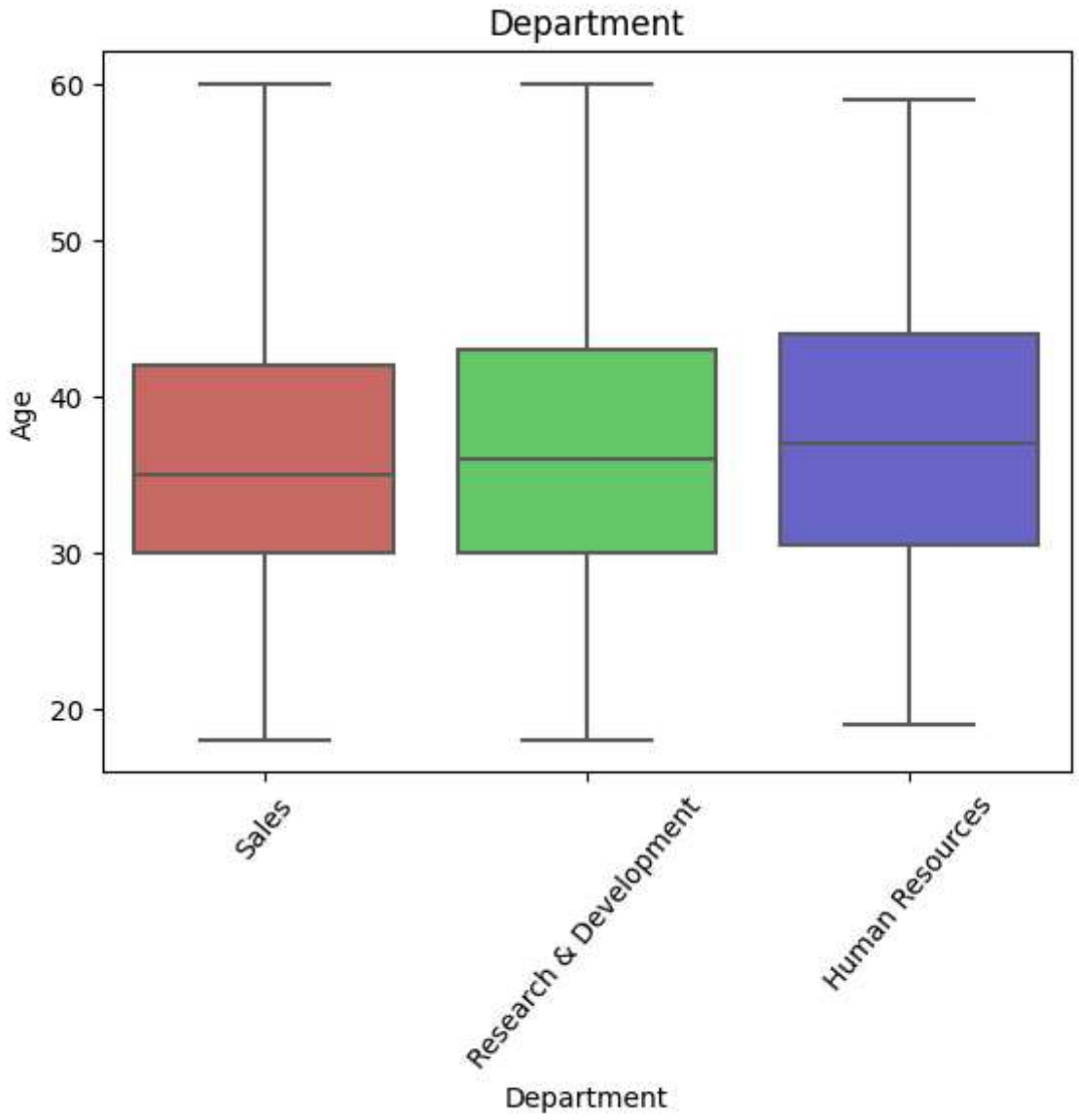
```
plt.figure(figsize=(18,35))
for i,col in enumerate(cat):
    axes = plt.subplot(8,2, i + 1)
    data[col].value_counts().plot.pie(autopct='%.1f%%', colors=color,textprops={
    plt.title(col)
plt.tight_layout()
plt.show()
```

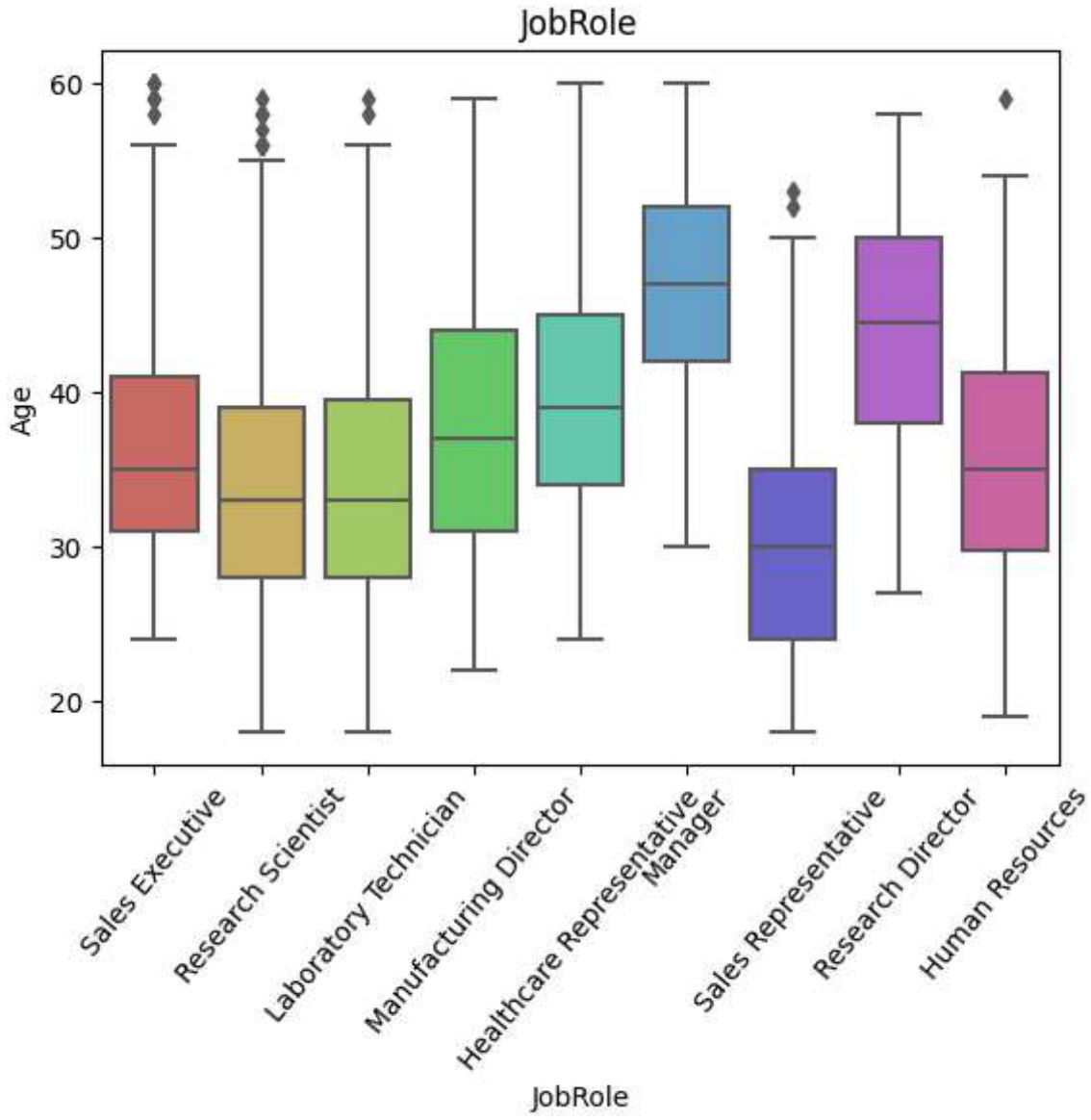
In [331…

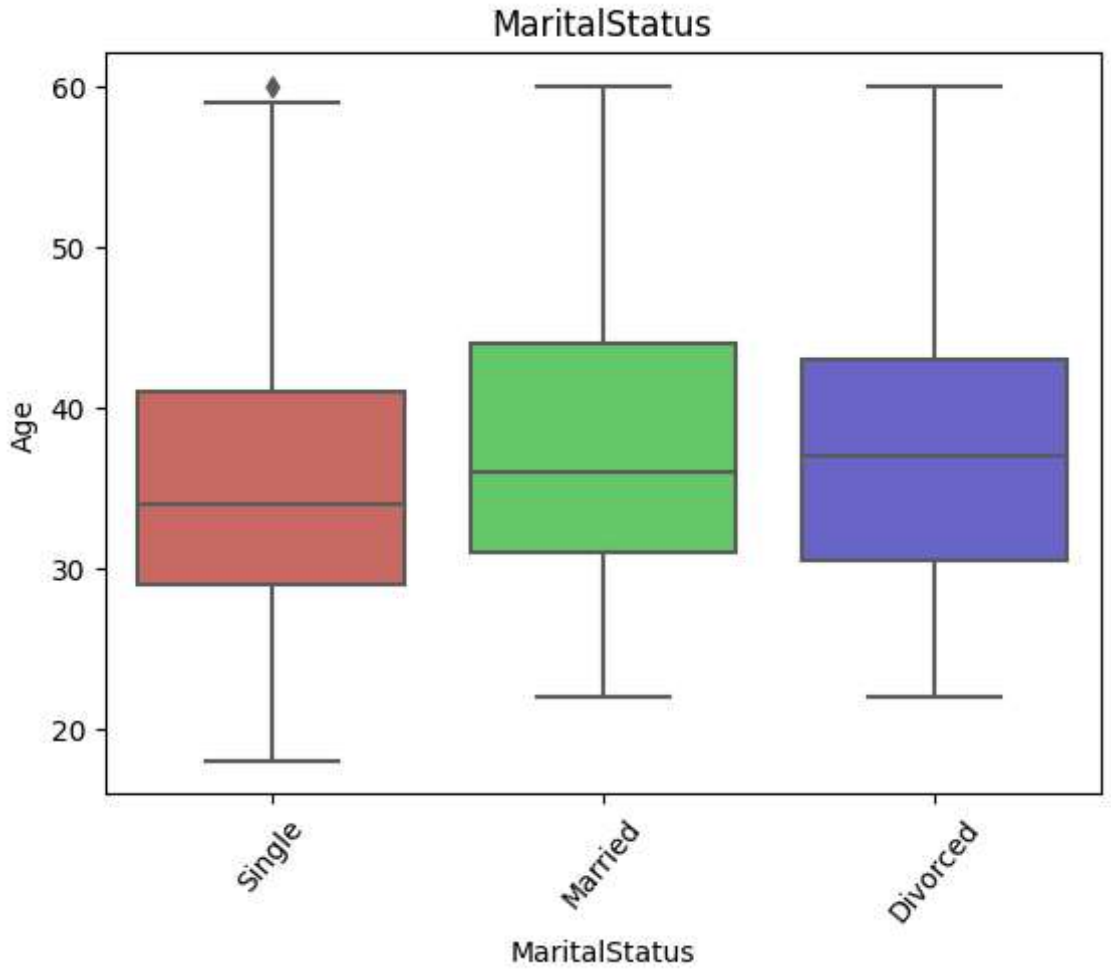
```
colonnes_1 = ['EducationField', 'Department', 'JobRole', 'MaritalStatus']
plt.figure(figsize=(4,3))
for i, col in enumerate(colonnes_1):
    sns.boxplot(data=data,x=data[col],y='Age',palette='hls')
    plt.title(col)
    plt.xlabel(col)
```

```
    plt.xticks(rotation=50)
    plt.ylabel('Age')
    plt.show()
```



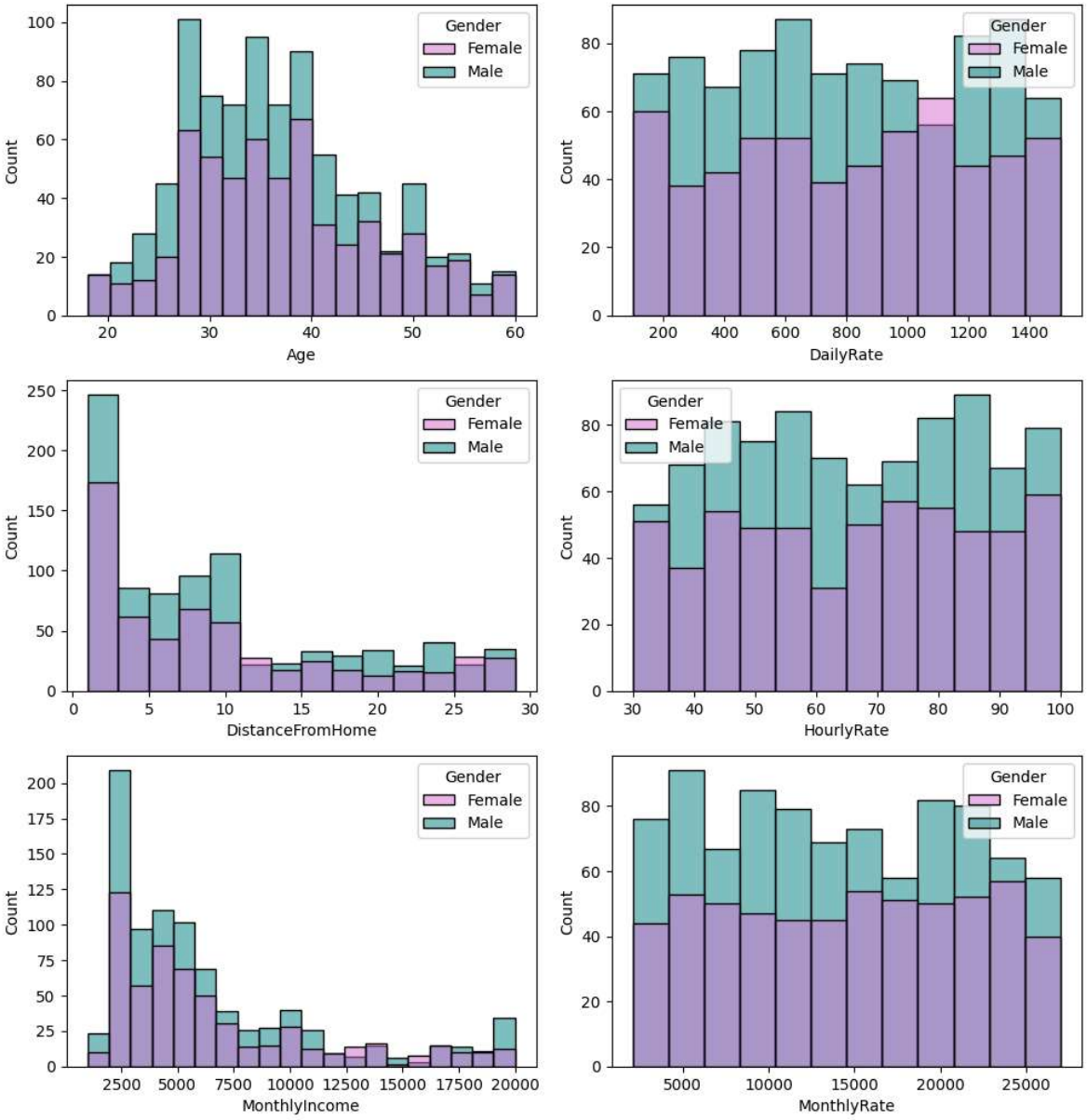

Department
60
50
40
30
20
Age
Sales
Research & Development
Human Resources
Department


JobRole
60
50
40
30
20
Age
Sales Executive
Research Scientist
Laboratory Technician
Manufacturing Director
Healthcare Representative
Manager
Sales Representative
Research Director
Human Resources
JobRole



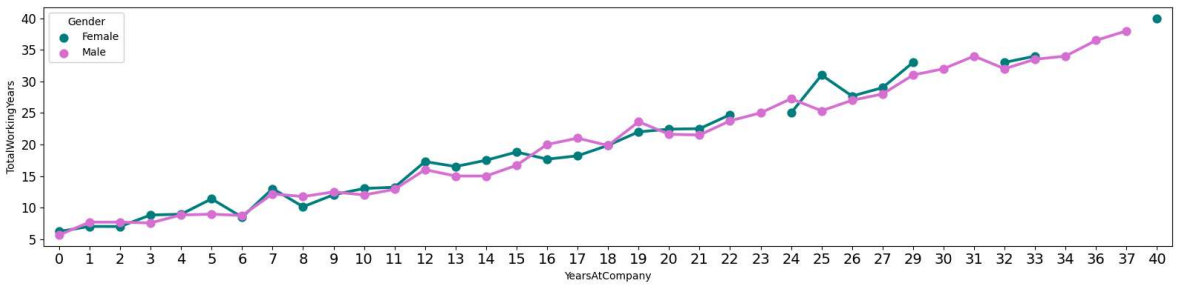
In [320…

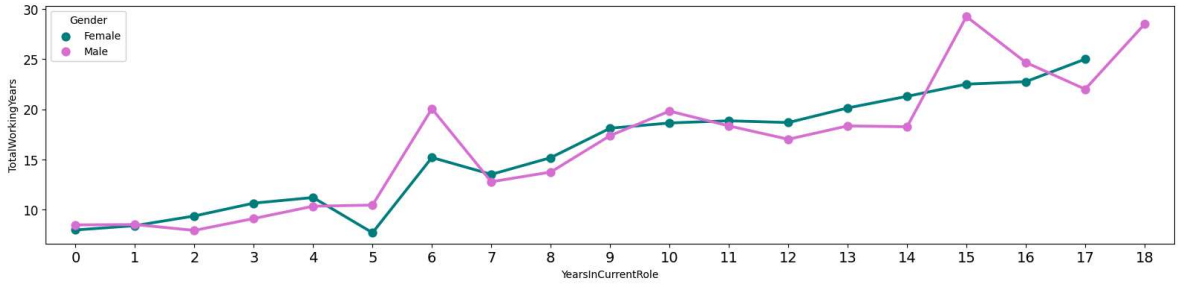
```
histogrm = ['Age', 'DailyRate','DistanceFromHome','HourlyRate','MonthlyIncome',
            'MonthlyRate']

plt.figure(figsize=(10,20))
for i,col in enumerate(histogrm):
    axes = plt.subplot(6,2, i + 1)
    sns.histplot(x=data[col], hue=data['Gender'], palette=['orchid', 'teal'])
plt.tight_layout()
plt.show()
```

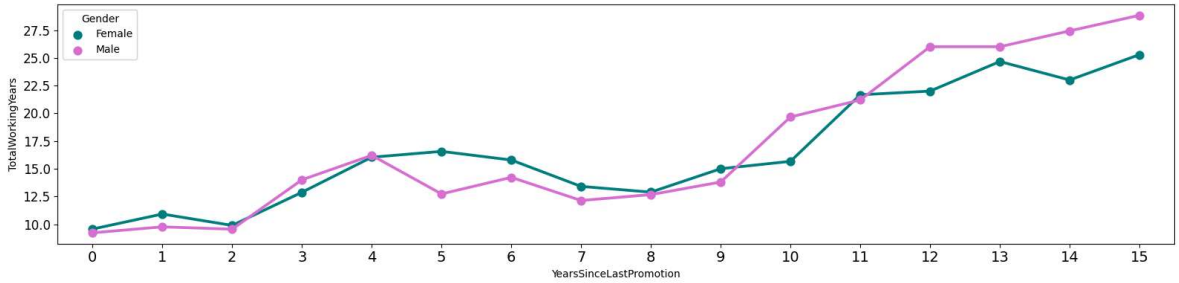


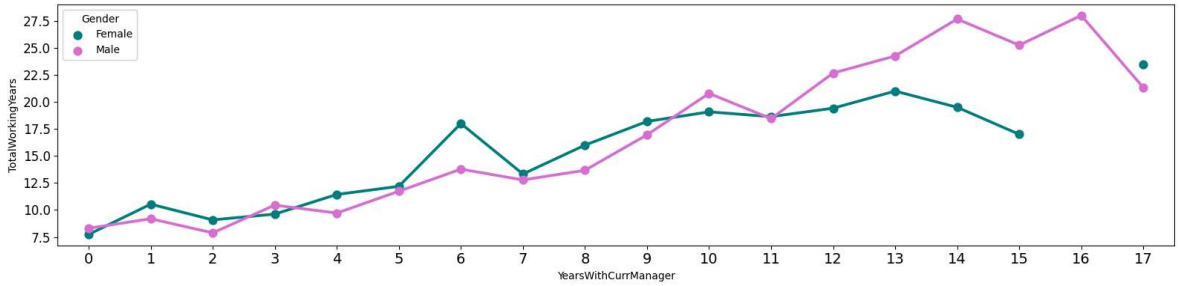
In [318…

```
# Correlation: Total des années de travail
colonnes = ['YearsAtCompany','YearsInCurrentRole', 'YearsSinceLastPromotion','Ye

for i in colonnes :
    fig, axes = plt.subplots(figsize=(16,4))
    sns.pointplot(x=data[i], y=data['TotalWorkingYears'], hue=data['Gender'], pa
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=12)
    plt.tight_layout()
    plt.show()
```



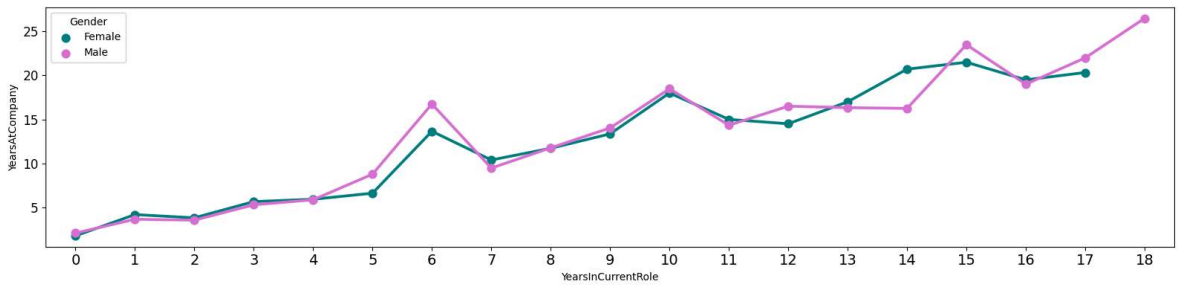


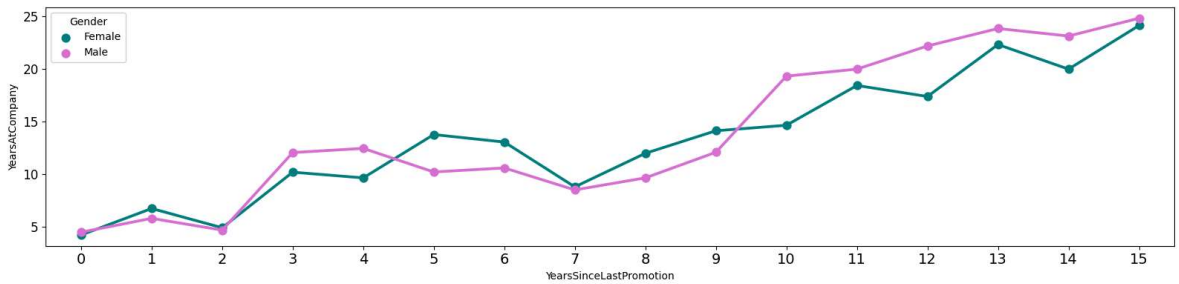


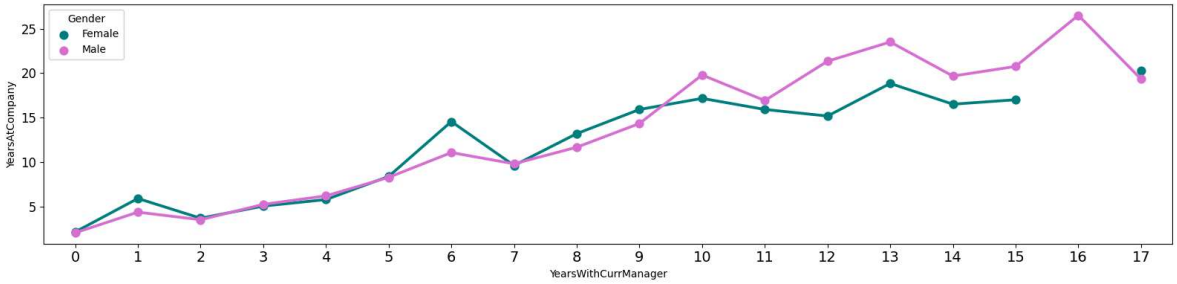


In [319…

```
# Correlation: Années dans l'entreprise
colonnes = ['YearsInCurrentRole', 'YearsSinceLastPromotion','YearsWithCurrManage

for i in colonnes:
    fig, axes = plt.subplots(figsize=(16,4))
    sns.pointplot(x=data[i], y=data['YearsAtCompany'], hue=data['Gender'], palet
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=12)

    plt.tight_layout()
    plt.show()
```



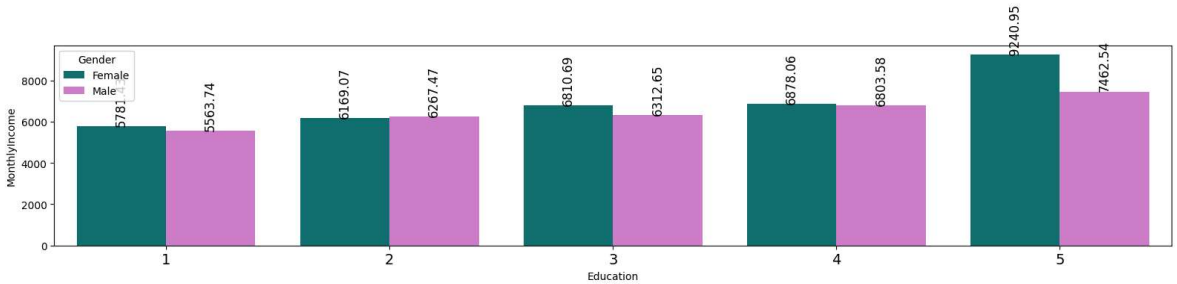


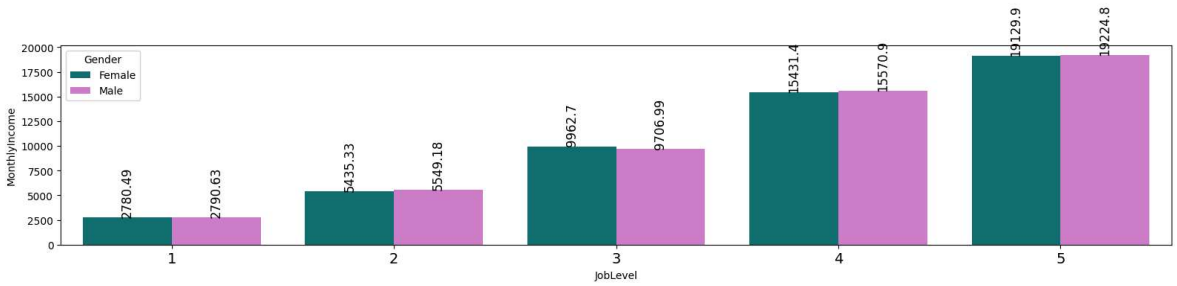


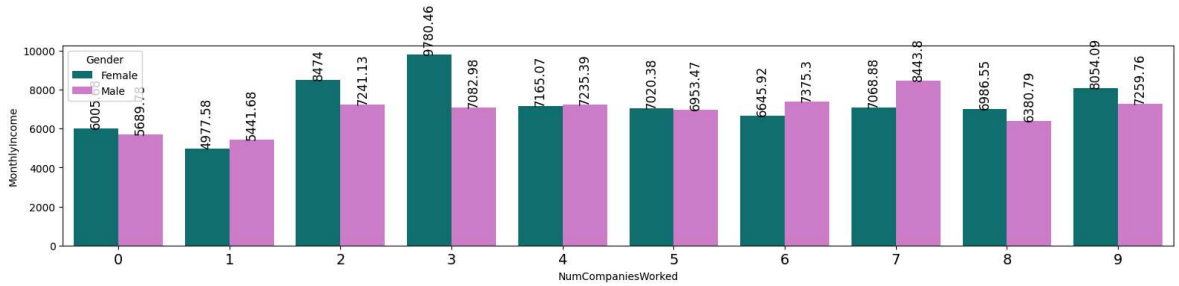
Le graphique linéaire monte une corrélattion claire entre l'ancienété des employées et le revenu mensuels. A mesure que les employées restent plus longtemps dans l'organisation, leurs revenus mensuales augmentent régulierement, ce qui indique une trajectoir enrichichant en matière de rénumaration financières.

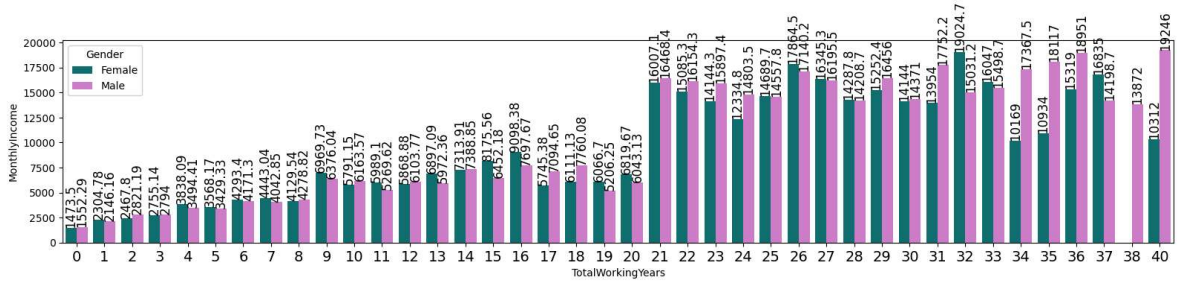
In [ ]:
```
# Correlation: Nombres d'années sur le travail actuel
colonnes = ['YearsSinceLastPromotion','YearsWithCurrManager']

for i in colonnes :
    fig, axes = plt.subplots(figsize=(16,4))
    sns.pointplot(x=data[i], y=data['YearsInCurrentRole'], hue=data['Gender'], p
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=12)

    plt.tight_layout()
    plt.show()
```

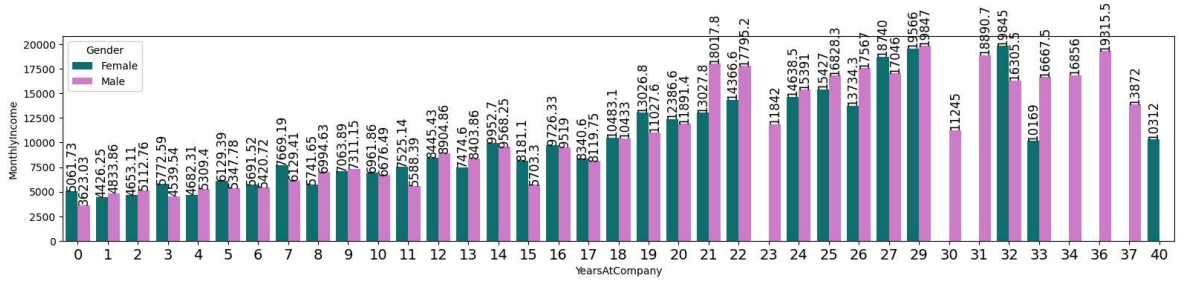
In [322…
```
# Revenu mensuel pour les années de travail
colonnes = ['Education','JobLevel','NumCompaniesWorked','TotalWorkingYears','Yea
            'YearsInCurrentRole', 'YearsSinceLastPromotion',
            'YearsWithCurrManager']
for i in colonnes:
    fig, axes = plt.subplots(figsize=(16,4))
    sns.barplot(x=data[i], y=data['MonthlyIncome'], hue=data['Gender'], palette=
    plt.xticks(fontsize=14)
    for cont in axes.containers:
        axes.bar_label(cont,rotation=90, fontsize=12)
    plt.tight_layout()
    plt.show()
```

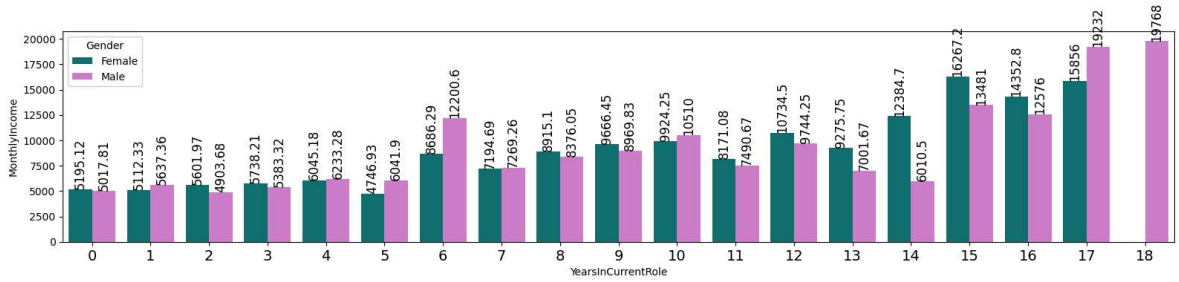


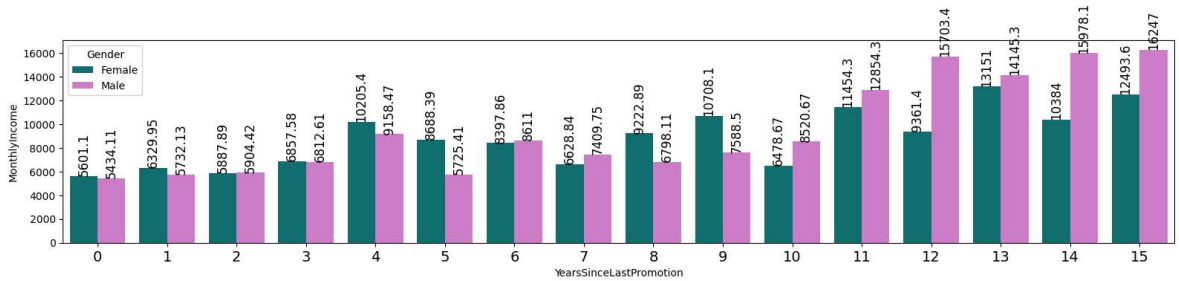


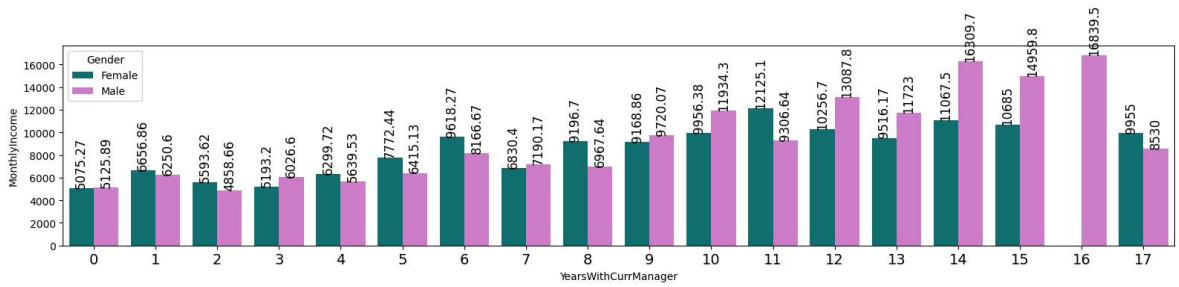












In [ ]:
```
# Salaire mensuel par sexe
colonnes = ['Department','EducationField','Gender','MaritalStatus','BusinessTrav

for i in colonnes :
    fig, axes = plt.subplots(figsize=(16,6))
    sns.barplot(x=data[i], y=data['MonthlyIncome'], hue=data['Gender'], palette=
```

```
    plt.xticks(rotation=50,fontsize=14)
    for cont in axes.containers:
        axes.bar_label(cont,rotation=30, fontsize=14)
    plt.tight_layout()
    plt.show()
```

In [ ]:
```
# Salaire mensuel par situation matrimoniale
colonnes = ['Gender','Department','EducationField','Gender','BusinessTravel','Jc
for i in colonnes:
    fig, axes = plt.subplots(figsize=(16,6))
    sns.barplot(x=data[i], y=data['MonthlyIncome'], hue=data['MaritalStatus'], p
    plt.xticks(rotation=50,fontsize=14)
    for cont in axes.containers:
        axes.bar_label(cont,rotation=30,fontsize=14)
    plt.tight_layout()
    plt.show()
```

In [ ]:
```

```