1-eda

June 18, 2024

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import warnings

warnings.filterwarnings("ignore")

%matplotlib inline
```

[3]:
```
df =pd.read_csv("Visadataset.csv")
```

[4]:
```
df.head()
```

[4]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_of_employees	yr_of_estab	region_of_employment	prevailing_wage	unit_of_wage	full_time_position	case_status
0	EZYV01	Asia	High School	N	N	14513	2007	West	592.2029	Hour	Y	Denied
1	EZYV02	Asia	Master's	Y	N	2412	2002	Northeast	83425.6500	Year	Y	Certified
2	EZYV03	Asia	Bachelor's	N	Y	44444	2008	West	122996.8600	Year	Y	Denied
3	EZYV04	Asia	Bachelor's	N	N	98	1897	West	83434.0300	Year	Y	Denied
4	EZYV05	Africa	Master's	Y	N	1082	2005	South	149907.3900	Year	Y	Certified

[5]:
```
df.shape
```

[5]: (25480, 12)

[6]:
```
df.describe()
```

[6]:

	no_of_employees	yr_of_estab	prevailing_wage
count	25480.000000	25480.000000	25480.000000
mean	5667.043210	1979.409929	74455.814592
std	22877.928848	42.366929	52815.942327
min	-26.000000	1800.000000	2.136700
25%	1022.000000	1976.000000	34015.480000
50%	2109.000000	1997.000000	70308.210000
75%	3504.000000	2005.000000	107735.512500
max	602069.000000	2016.000000	319210.270000

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25480 entries, 0 to 25479
Data columns (total 12 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   case_id                25480 non-null  object
 1   continent              25480 non-null  object
 2   education_of_employee  25480 non-null  object
 3   has_job_experience     25480 non-null  object
 4   requires_job_training  25480 non-null  object
 5   no_of_employees        25480 non-null  int64
 6   yr_of_estab            25480 non-null  int64
 7   region_of_employment   25480 non-null  object
 8   prevailing_wage        25480 non-null  float64
 9   unit_of_wage           25480 non-null  object
 10  full_time_position     25480 non-null  object
 11  case_status            25480 non-null  object
dtypes: float64(1), int64(2), object(9)
memory usage: 2.3+ MB
```

[8]:
```
df.columns
```

[8]:
```
Index(['case_id', 'continent', 'education_of_employee', 'has_job_experience',
       'requires_job_training', 'no_of_employees', 'yr_of_estab',
       'region_of_employment', 'prevailing_wage', 'unit_of_wage',
       'full_time_position', 'case_status'],
      dtype='object')
```

[9]:
```
row = df.iterrows
row
```

[9]:
```
<bound method DataFrame.iterrows of          case_id continent
education_of_employee has_job_experience  \
0         EZYV01      Asia           High School                  N
1         EZYV02      Asia             Master's                   Y
2         EZYV03      Asia            Bachelor's                  N
3         EZYV04      Asia            Bachelor's                  N
4         EZYV05    Africa             Master's                   Y
…           …       …                    …                  …
25475  EZYV25476      Asia            Bachelor's                  Y
25476  EZYV25477      Asia           High School                  Y
25477  EZYV25478      Asia             Master's                   Y
25478  EZYV25479      Asia             Master's                   Y
25479  EZYV25480      Asia            Bachelor's                  Y

      requires_job_training  no_of_employees  yr_of_estab  \
0                         N            14513         2007
1                         N             2412         2002
2                         Y            44444         2008
3                         N               98         1897
4                         N             1082         2005
…                       …              …           …
25475                     Y             2601         2008
25476                     N             3274         2006
25477                     N             1121         1910
25478                     Y             1918         1887
25479                     N             3195         1960

      region_of_employment  prevailing_wage unit_of_wage full_time_position  \
0                     West         592.2029         Hour                  Y
1                Northeast       83425.6500         Year                  Y
2                     West      122996.8600         Year                  Y
3                     West       83434.0300         Year                  Y
4                    South      149907.3900         Year                  Y
…                    …              …           …                 …
25475                South       77092.5700         Year                  Y
25476           Northeast      279174.7900         Year                  Y
25477                South      146298.8500         Year                  N
25478                 West       86154.7700         Year                  Y
25479              Midwest       70876.9100         Year                  Y

      case_status
0          Denied
1       Certified
2          Denied
3          Denied
4       Certified
…           …
```

```
25475   Certified
25476   Certified
25477   Certified
25478   Certified
25479   Certified

[25480 rows x 12 columns]>
```

0.1 Exploring Data

[10]:
```
# define numerical & categorical columns
numeric_features = [feature for feature in df.columns if df[feature].dtype !=␣
 ↪'object']
categorical_features = [feature for feature in df.columns if df[feature].dtype␣
 ↪== 'object']

# print columns
print('We have {} numerical features : {}'.format(len(numeric_features),␣
 ↪numeric_features))
print('\nWe have {} categorical features : {}'.
 ↪format(len(categorical_features), categorical_features))
```

```
We have 3 numerical features : ['no_of_employees', 'yr_of_estab',
'prevailing_wage']

We have 9 categorical features : ['case_id', 'continent',
'education_of_employee', 'has_job_experience', 'requires_job_training',
'region_of_employment', 'unit_of_wage', 'full_time_position', 'case_status']
```

[11]:
```
# proportion of count data on categorical columns
for col in categorical_features:
    print(df[col].value_counts(normalize=True) * 100)
    print('---------------------------')
```

```
case_id
EZYV01       0.003925
EZYV16995    0.003925
EZYV16993    0.003925
EZYV16992    0.003925
EZYV16991    0.003925
               …
EZYV8492     0.003925
EZYV8491     0.003925
EZYV8490     0.003925
EZYV8489     0.003925
EZYV25480    0.003925
Name: proportion, Length: 25480, dtype: float64
```

```
--------------------------
continent
Asia             66.173469
Europe           14.646782
North America    12.919937
South America     3.343799
Africa            2.162480
Oceania           0.753532
Name: proportion, dtype: float64
--------------------------
education_of_employee
Bachelor's     40.164835
Master's       37.810047
High School    13.422292
Doctorate       8.602826
Name: proportion, dtype: float64
--------------------------
has_job_experience
Y    58.092622
N    41.907378
Name: proportion, dtype: float64
--------------------------
requires_job_training
N    88.402669
Y    11.597331
Name: proportion, dtype: float64
--------------------------
region_of_employment
Northeast    28.237834
South        27.539246
West         25.847724
Midwest      16.903454
Island        1.471743
Name: proportion, dtype: float64
--------------------------
unit_of_wage
Year     90.117739
Hour      8.465463
Week      1.067504
Month     0.349294
Name: proportion, dtype: float64
--------------------------
full_time_position
Y    89.375981
N    10.624019
Name: proportion, dtype: float64
--------------------------
case_status
```

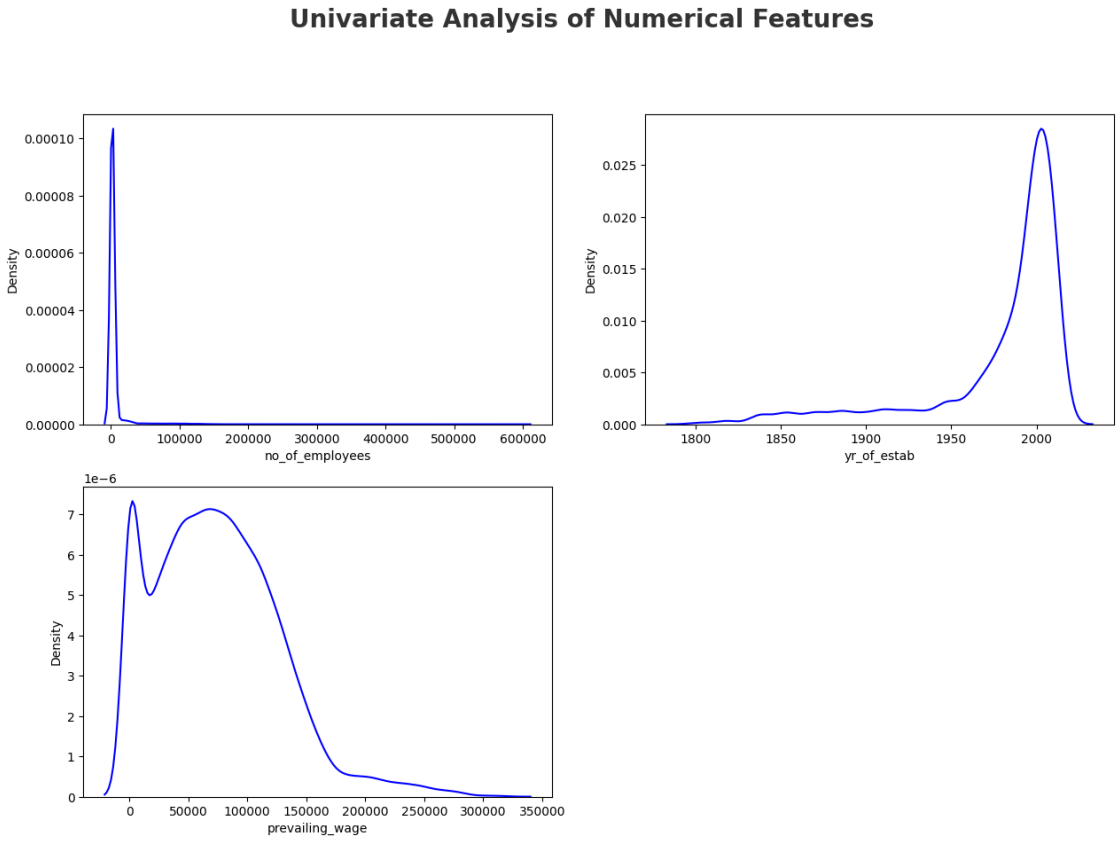
```
Certified    66.789639
Denied       33.210361
Name: proportion, dtype: float64
---------------------------
```

0.2 Numerical Features

[16]:
```
plt.figure(figsize=(15, 10))
plt.suptitle('Univariate Analysis of Numerical Features', fontsize=20,␣
 ↪fontweight='bold', alpha=0.8, y=1.)

for i in range(0, len(numeric_features)):
    plt.subplot(2, 2, i+1)
    sns.kdeplot(x=df[numeric_features[i]], color='blue')
    plt.xlabel(numeric_features[i])
    plt.tight_layout()

# save plot
# plt.savefig('./images/Univariate_Num.png')
```

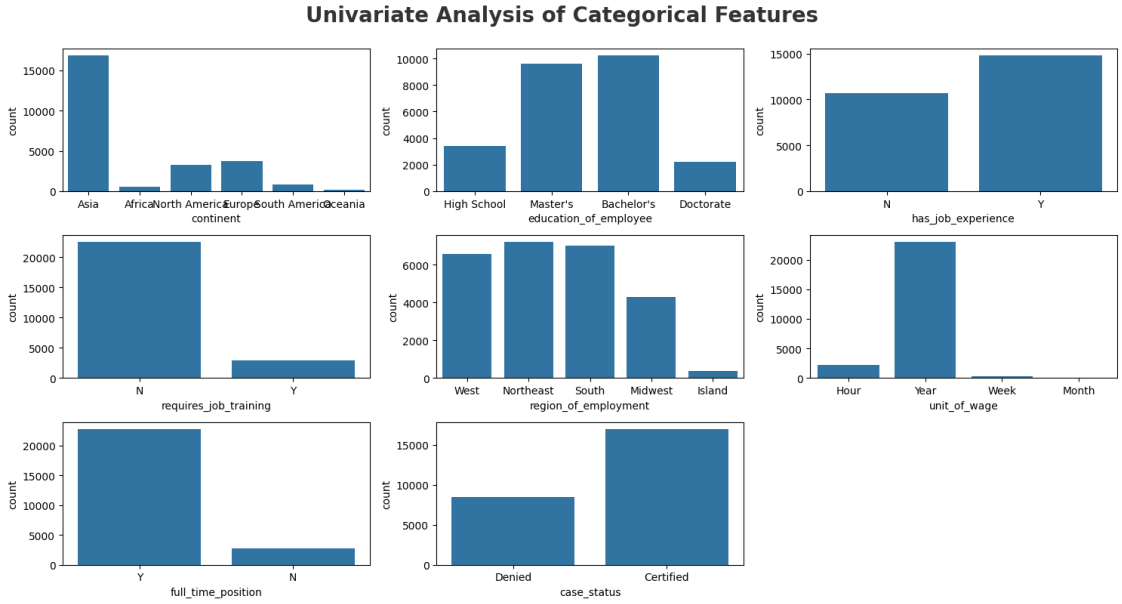


[13]:
```
## Categorical Features
```

[17]:
```
# categorical columns
categorical_features.remove('case_id')
plt.figure(figsize=(15, 8))
plt.suptitle('Univariate Analysis of Categorical Features', fontsize=20,␣
 ↪fontweight='bold', alpha=0.8, y=1.)

for i in range(0, len(categorical_features)):
    plt.subplot(3, 3, i+1)
    sns.countplot(x=df[categorical_features[i]])
    plt.xlabel(categorical_features[i])
    plt.tight_layout()

# save plot
# plt.savefig('./images/Univariate_Cat.png')
```



[18]:
```
discrete_features=[feature for feature in numeric_features if len(df[feature].
 ↪unique())<=25]

continuous_features=[feature for feature in numeric_features if len(df[feature].
 ↪unique()) > 25]

print('We have {} discrete features : {}'.format(len(discrete_features),␣
 ↪discrete_features))
print('\nWe have {} continuous_features : {}'.format(len(continuous_features),␣
 ↪continuous_features))
```

```
We have 0 discrete features : []
```

```
We have 3 continuous_features : ['no_of_employees', 'yr_of_estab',
'prevailing_wage']
```

[19]:
```
df.head()
```

[19]:
```
  case_id continent education_of_employee has_job_experience  \
0  EZYV01      Asia           High School                  N
1  EZYV02      Asia             Master's                   Y
2  EZYV03      Asia            Bachelor's                  N
3  EZYV04      Asia            Bachelor's                  N
4  EZYV05    Africa             Master's                   Y

  requires_job_training  no_of_employees  yr_of_estab region_of_employment  \
0                     N            14513         2007                 West
1                     N             2412         2002            Northeast
2                     Y            44444         2008                 West
3                     N               98         1897                 West
4                     N             1082         2005                South

   prevailing_wage unit_of_wage full_time_position case_status
0         592.2029         Hour                  Y      Denied
1       83425.6500         Year                  Y   Certified
2      122996.8600         Year                  Y      Denied
3       83434.0300         Year                  Y      Denied
4      149907.3900         Year                  Y   Certified
```

Check Multicollinearity for Categorical features

- A chi-squared test (also chi-square or χ^2 test) is a statistical hypothesis test that is valid to perform when the test statistic is chi-squared distributed under the null hypothesis, specifically Pearson's chi-squared test
- A chi-square statistic is one way to show a relationship between two categorical variables.
- Here we test correlation of Categorical columns with Target column i.e case_status

Null Hypothesis ( ): The Feature is independent of target column (No-Correlation)

Alternative Hypothesis ( ): The Feature and Target column are not independent (Correalted)

[17]:
```
from scipy.stats import chi2_contingency
chi2_test = []
for feature in categorical_features:
    if chi2_contingency(pd.crosstab(df['case_status'], df[feature]))[1] < 0.05:
        chi2_test.append('Reject Null Hypothesis')
    else:
        chi2_test.append('Fail to Reject Null Hypothesis')
result = pd.DataFrame(data=[categorical_features, chi2_test]).T
result.columns = ['Column', 'Hypothesis Result']
```

```
result
```

[17]:

	Column	Hypothesis Result
0	continent	Reject Null Hypothesis
1	education_of_employee	Reject Null Hypothesis
2	has_job_experience	Reject Null Hypothesis
3	requires_job_training	Fail to Reject Null Hypothesis
4	region_of_employment	Reject Null Hypothesis
5	unit_of_wage	Reject Null Hypothesis
6	full_time_position	Reject Null Hypothesis
7	case_status	Reject Null Hypothesis

[18]:
```
df.isnull().sum()
```

[18]:
```
case_id                 0
continent               0
education_of_employee   0
has_job_experience      0
requires_job_training   0
no_of_employees         0
yr_of_estab             0
region_of_employment    0
prevailing_wage         0
unit_of_wage            0
full_time_position      0
case_status             0
dtype: int64
```

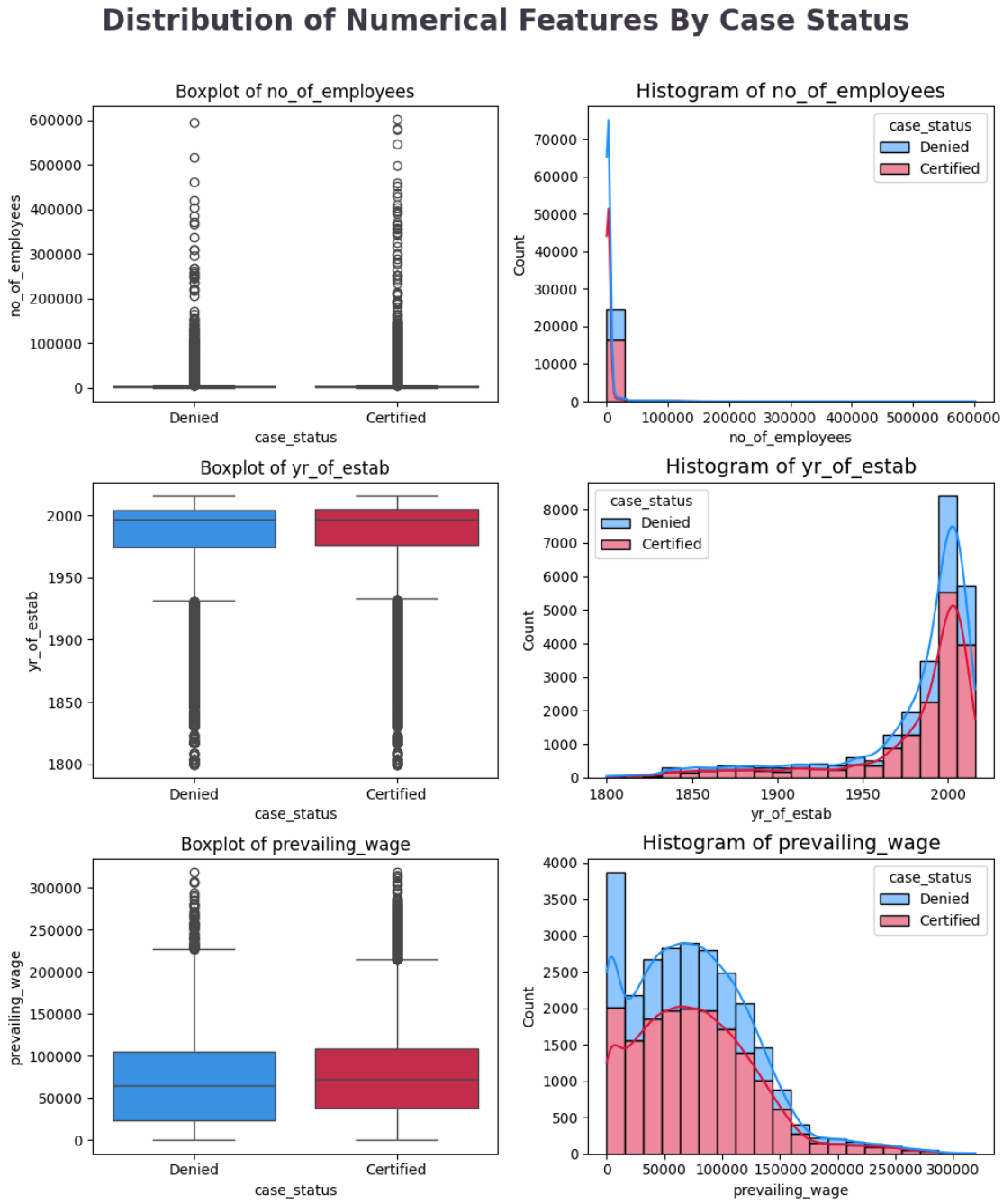
[19]:
```
continues_features=[feature for feature in numeric_features if len(df[feature].
 ↪unique())>=10]
print('Num of continues features :',continues_features)
```

```
Num of continues features : ['no_of_employees', 'yr_of_estab',
'prevailing_wage']
```

0.2.1 Distribution of Numerical Features By Case Status

[20]:
```
clr1 = ['#1E90FF', '#DC143C']
fig, ax = plt.subplots(3, 2, figsize=(10,12))
fig.suptitle('Distribution of Numerical Features By Case Status',␣
 ↪color='#3C3744',
             fontsize=20, fontweight='bold', ha='center')
for i, col in enumerate(continues_features):
    sns.boxplot(data=df, x='case_status', y=col,  palette=clr1, ax=ax[i,0])
    ax[i,0].set_title(f'Boxplot of {col}', fontsize=12)
    sns.histplot(data=df, x=col, hue='case_status', bins=20, kde=True,
                 multiple='stack', palette=clr1, ax=ax[i,1])
```

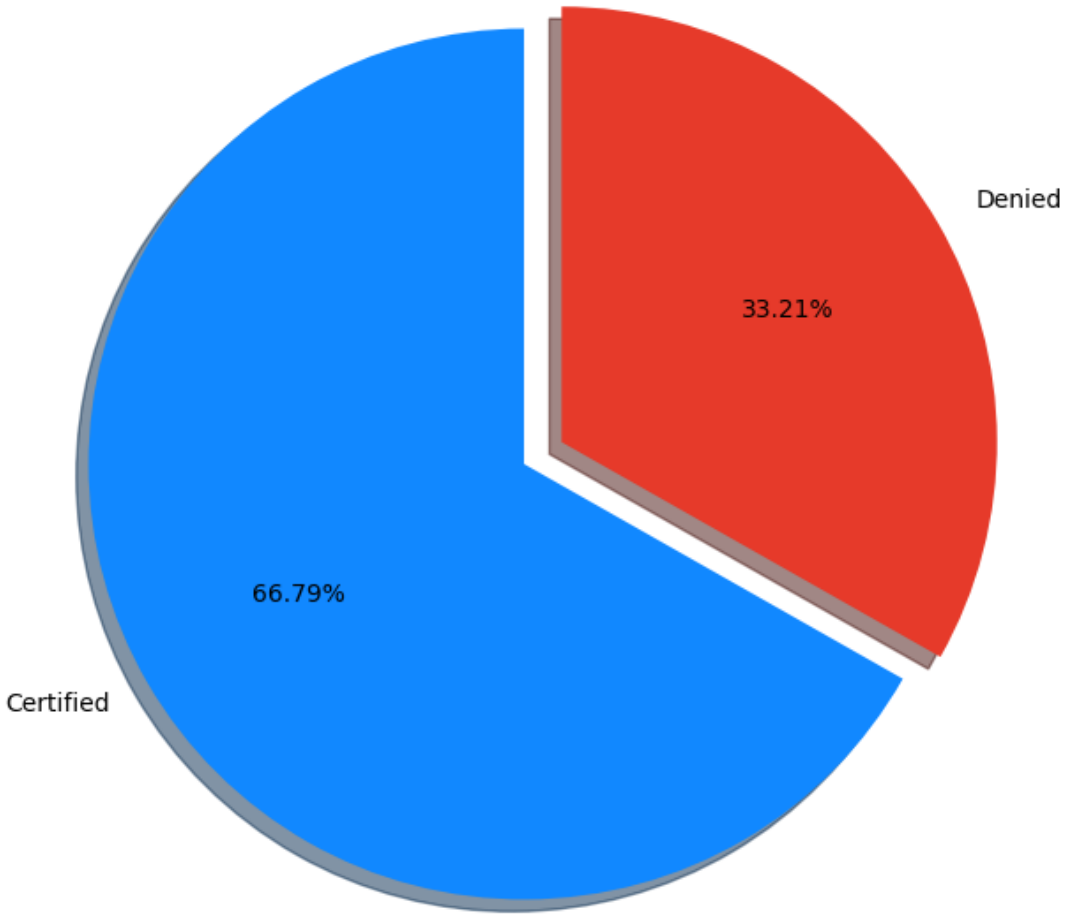
```
    ax[i,1].set_title(f'Histogram of {col}', fontsize=14)
fig.tight_layout()
fig.subplots_adjust(top=0.90)
# plt.savefig('images/multivariate_num.png')
```



0.2.2 Visualize the Target Feature

[21]:
```
percentage = df.case_status.value_counts(normalize=True)*100
labels = ["Certified","Denied"]

# Plot PieChart with Ploty library
fig, ax = plt.subplots(figsize =(15, 8))
explode = (0, 0.1)
colors = ['#1188ff','#e63a2a']
ax.pie(percentage, labels = labels, startangle = 90,
       autopct='%1.2f%%',explode=explode, shadow=True, colors=colors)
plt.show()
```



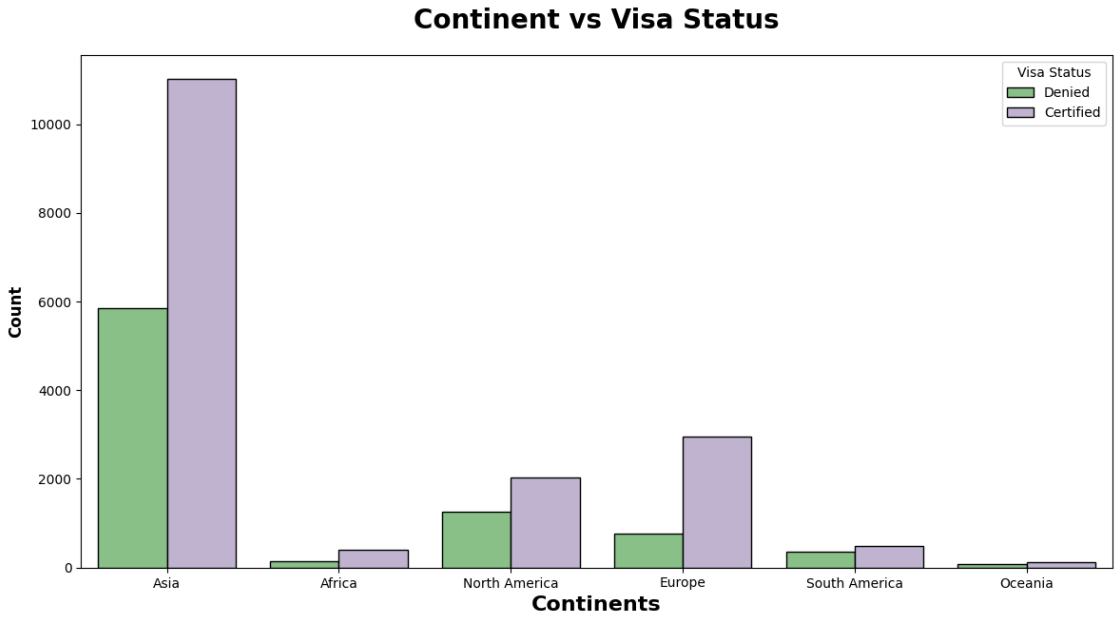
0.2.3 applicant Continent has any impact on Visa status

[22]:
```
#group data by continent and their count of case_status
df.groupby('continent')['case_status'].value_counts(normalize=True).
 ↪to_frame()*100
```

[22]:

continent	case_status	proportion
Africa	Certified	72.050817
	Denied	27.949183
Asia	Certified	65.310480
	Denied	34.689520
Europe	Certified	79.233655
	Denied	20.766345
North America	Certified	61.877278
	Denied	38.122722
Oceania	Certified	63.541667
	Denied	36.458333
South America	Certified	57.863850
	Denied	42.136150

[23]:
```
plt.subplots(figsize=(14,7))
sns.countplot(x="continent",hue="case_status", data=df, ec =␣
 ↪"black",palette="Accent")
plt.title("Continent vs Visa Status", weight="bold",fontsize=20, pad=20)
plt.ylabel("Count", weight="bold", fontsize=12)
plt.xlabel("Continents", weight="bold", fontsize=16)
plt.legend(title="Visa Status", fancybox=True)
plt.show()
```

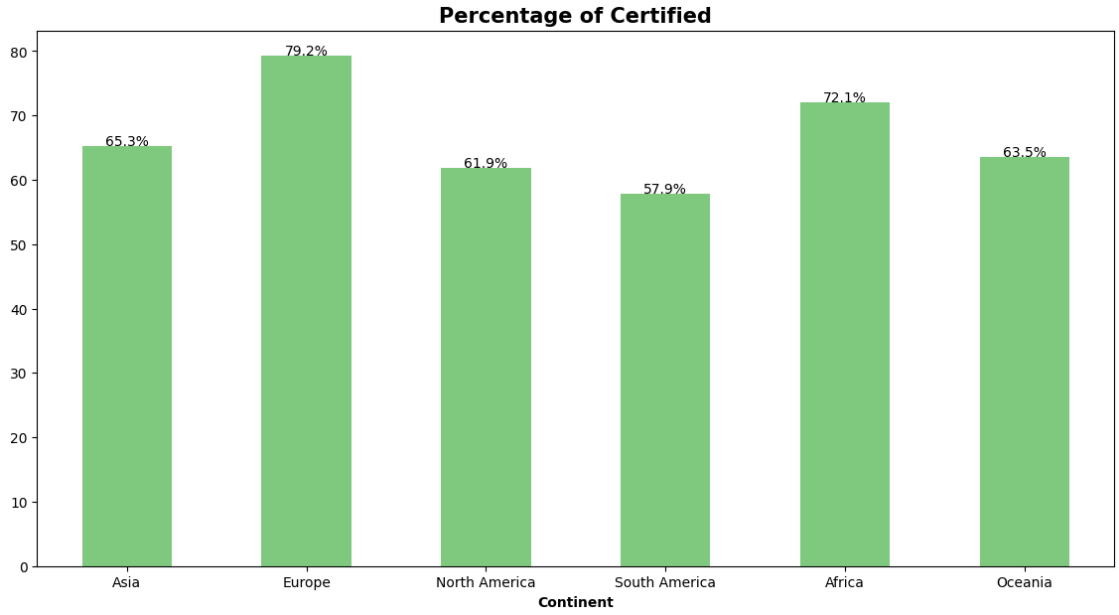


[24]:
```
df2 = df.copy()

plt.figure(figsize=[14,7])

(100*df2[df2["case_status"].isin(['Certified'])]['continent'].value_counts()/
 ↪df2['continent'].value_counts()).plot(
    kind='bar',stacked=True , colormap='Accent')
plt.title("Percentage of Certified" , fontsize = 15, fontweight ='bold'  )
order1 = (100*df2[df2["case_status"].isin(['Certified'])]['continent'].
 ↪value_counts()/df2['continent'].value_counts())
for n in range(order1.shape[0]):
    count = order1[n]
    strt='{:0.1f}%'.format(count)
    plt.text(n,count+0.1,strt,ha='center')

plt.xlabel('Continent' , fontweight ='bold')
plt.xticks(rotation=0)
plt.show()
```



Report

- As per the Chart Asia applicants applied more than other continents.
- 43% of Certified applications are from Asia.
- This is followed by Europe with 11% of Certified applications.
- Highest chance of getting certified if you are from Europe and followed by Africa

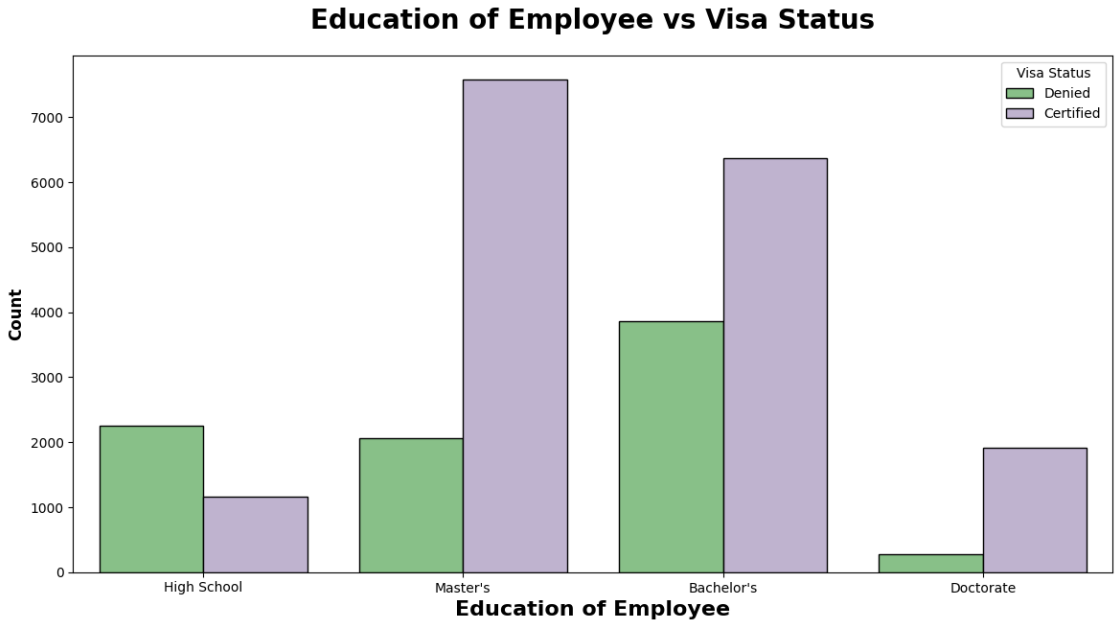
0.2.4 Impact of Education on Visa Status

[25]:
```
#group data by Education and their count of case_status
df.groupby('education_of_employee')['case_status'].value_counts(normalize=True).
 ↪to_frame()*100
```

[25]:

education_of_employee	case_status	proportion
Bachelor's	Certified	62.214188
	Denied	37.785812
Doctorate	Certified	87.226277
	Denied	12.773723
High School	Denied	65.964912
	Certified	34.035088
Master's	Certified	78.627777
	Denied	21.372223

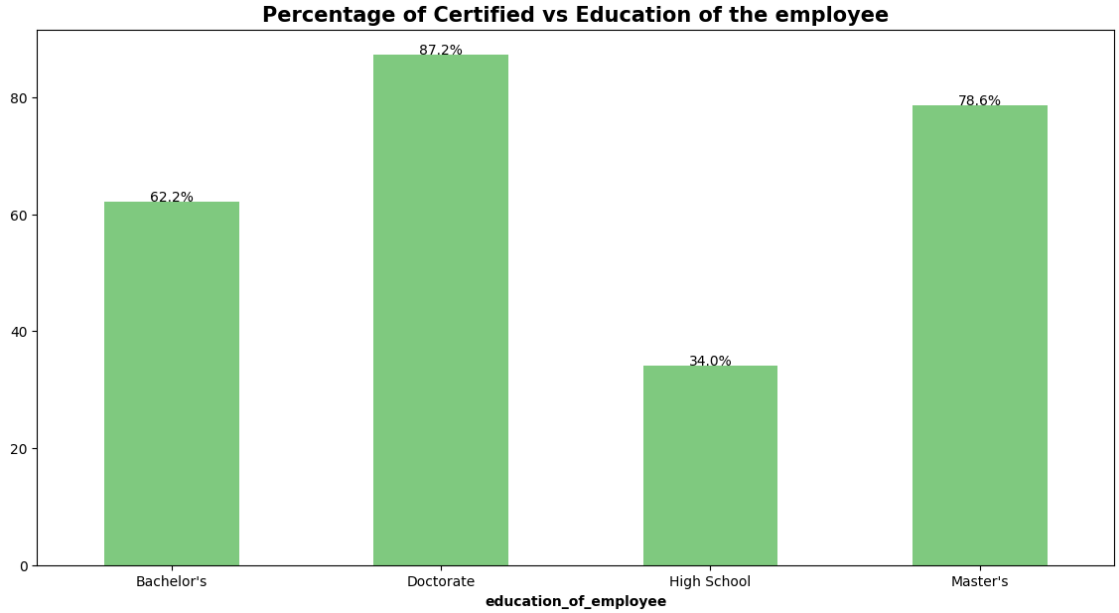
[26]:
```
plt.subplots(figsize=(14,7))
sns.countplot(x="education_of_employee",hue="case_status", data=df, ec =␣
 ↪"black",palette="Accent")
plt.title("Education of Employee vs Visa Status", weight="bold",fontsize=20,␣
 ↪pad=20)
plt.ylabel("Count", weight="bold", fontsize=12)
plt.xlabel("Education of Employee", weight="bold", fontsize=16)
plt.legend(title="Visa Status", fancybox=True)
plt.show()
```



[27]:
```
plt.figure(figsize=[14,7])

(100*df2[df2["case_status"].isin(['Certified'])]['education_of_employee'].
 ↪value_counts()/df2['education_of_employee'].value_counts()).plot(
    kind='bar',stacked=True , colormap='Accent')
plt.title("Percentage of Certified vs Education of the employee" , fontsize =␣
 ↪15, fontweight ='bold'  )
order1 = (100*df2[df2["case_status"].
 ↪isin(['Certified'])]['education_of_employee'].value_counts()/
 ↪df2['education_of_employee'].value_counts())
for n in range(order1.shape[0]):
    count = order1[n]
    strt='{:0.1f}%'.format(count)
    plt.text(n,count+0.1,strt,ha='center')

plt.xlabel('education_of_employee' , fontweight ='bold')
plt.xticks(rotation=0)
plt.show()
```



0.2.5 Impact of Has_job_experience on Visa Status

[28]:
```
#group data by has_job_experience and count case_status
df.groupby('has_job_experience')['case_status'].value_counts(normalize=True).
 ↪to_frame()*100
```

[28]:

		proportion
has_job_experience	case_status	
N	Certified	56.134108
	Denied	43.865892
Y	Certified	74.476422
	Denied	25.523578

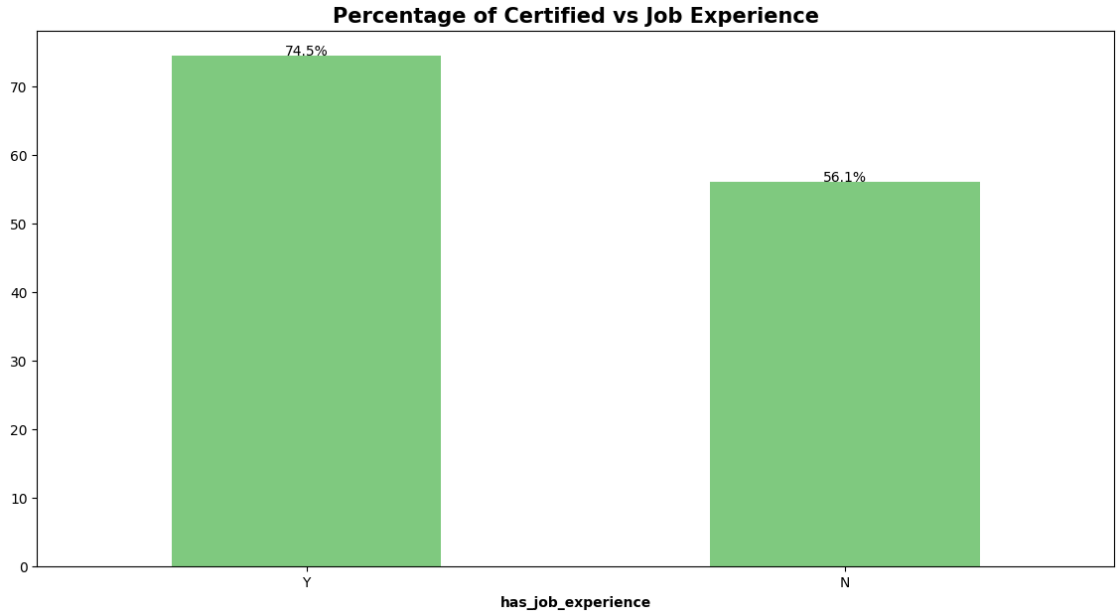
[29]:
```
plt.subplots(figsize=(13,7))
sns.countplot(x="has_job_experience",hue="case_status", data=df, ec="black",palette="Accent")
plt.title("Previous Work Experience of Employee vs Visa Status",weight="bold",fontsize=20, pad=20)
plt.ylabel("Count", weight="bold", fontsize=12)
plt.xlabel("Work Experience of Employee", weight="bold", fontsize=16)
plt.legend(title="Visa Status", fancybox=True)
plt.show()
```

[30]:
```
plt.figure(figsize=[14,7])

(100*df2[df2["case_status"].isin(['Certified'])]['has_job_experience'].value_counts()/df2['has_job_experience'].value_counts()).plot(
    kind='bar',stacked=True , colormap='Accent')
plt.title("Percentage of Certified vs Job Experience" , fontsize = 15,fontweight ='bold'  )
```

```
order1 = (100*df2[df2["case_status"].isin(['Certified'])]['has_job_experience'].
  ↪value_counts()/df2['has_job_experience'].value_counts())
for n in range(order1.shape[0]):
    count = order1[n]
    strt='{:0.1f}%'.format(count)
    plt.text(n,count+0.1,strt,ha='center')

plt.xlabel('has_job_experience' , fontweight ='bold')
plt.xticks(rotation=0)
plt.show()
```



0.2.6 Impact of requires_job_training on Visa Status

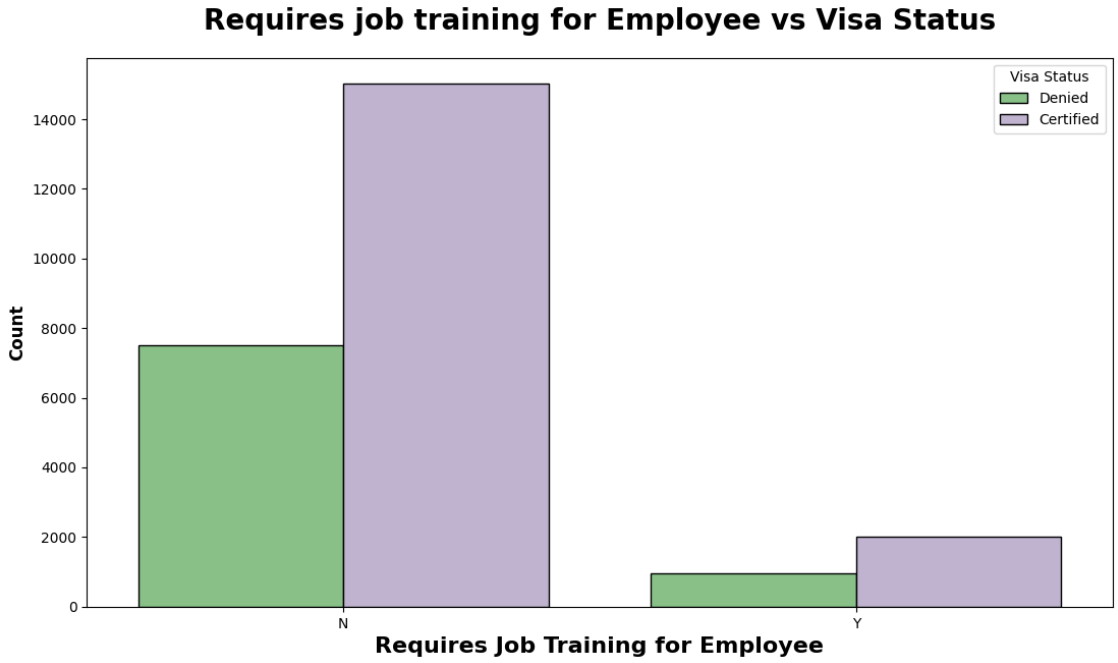
[31]:
```
#group data by requires_job_training and count case_status
df.groupby('requires_job_training')['case_status'].value_counts(normalize=True).
  ↪to_frame()*100
```

[31]:

		proportion
requires_job_training	case_status	
N	Certified	66.645949
	Denied	33.354051
Y	Certified	67.884941
	Denied	32.115059

[32]:
```
plt.subplots(figsize=(13,7))
```

```
sns.countplot(x="requires_job_training",hue="case_status", data=df, ec="black",palette="Accent")
plt.title("Requires job training for Employee vs Visa Status",weight="bold",fontsize=20, pad=20)
plt.ylabel("Count", weight="bold", fontsize=12)
plt.xlabel("Requires Job Training for Employee", weight="bold", fontsize=16)
plt.legend(title="Visa Status", fancybox=True)
plt.show()
```



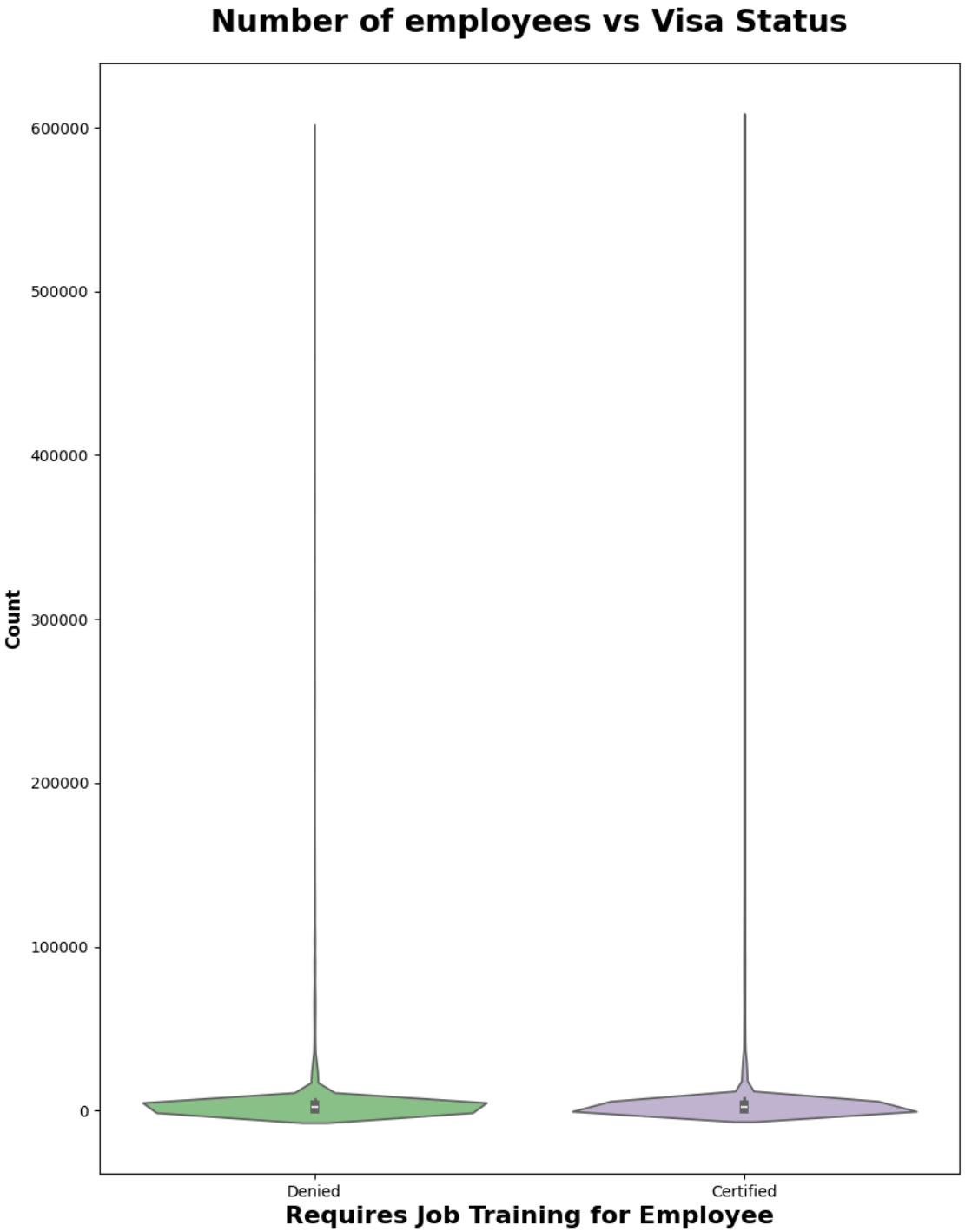
[33]:
```
plt.figure(figsize=[14,7])

(100*df2[df2["case_status"].isin(['Certified'])]['requires_job_training'].value_counts()/df2['requires_job_training'].value_counts()).plot(
    kind='bar',stacked=True , colormap='Accent')
plt.title("Percentage of Certified vs Required training" , fontsize = 15,fontweight ='bold'  )
order1 = (100*df2[df2["case_status"].isin(['Certified'])]['requires_job_training'].value_counts()/df2['requires_job_training'].value_counts())
for n in range(order1.shape[0]):
    count = order1[n]
    strt='{:0.1f}%'.format(count)
    plt.text(n,count+0.1,strt,ha='center')

plt.xlabel('requires_job_training' , fontweight ='bold')
```

```
plt.xticks(rotation=0)
plt.show()
```

[34]:
```
plt.subplots(figsize=(10,13))
sns.violinplot(x="case_status",y='no_of_employees', data=df, palette="Accent")
plt.title("Number of employees vs Visa Status", weight="bold",fontsize=20,␣
↪pad=20)
plt.ylabel("Count", weight="bold", fontsize=12)
plt.xlabel("Requires Job Training for Employee", weight="bold", fontsize=16)
plt.ylim()
plt.show()
```

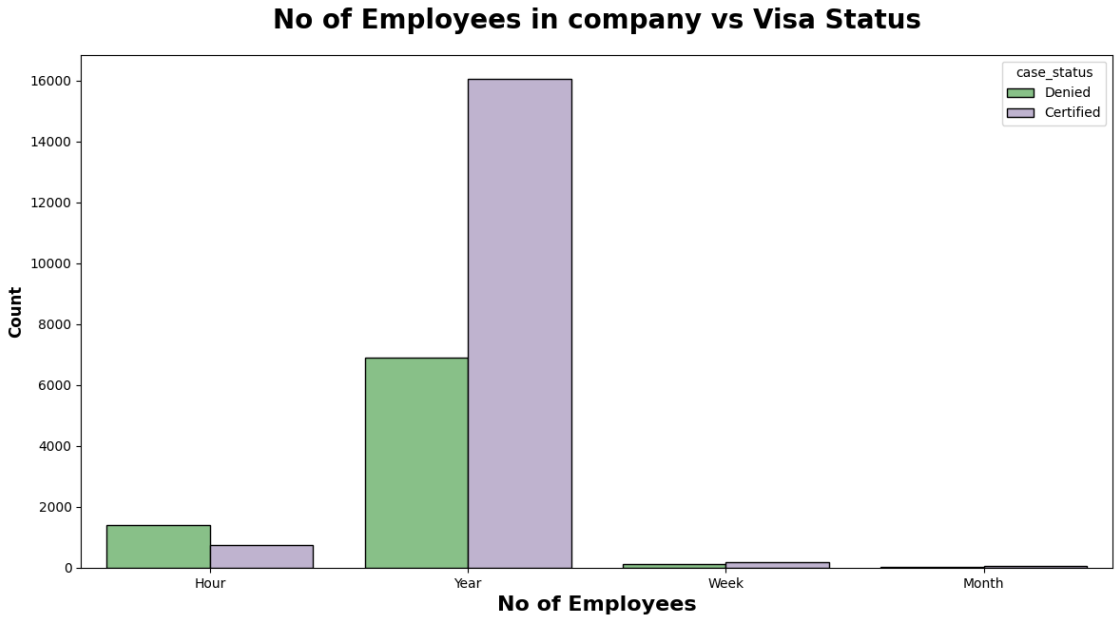

Number of employees vs Visa Status
600000
500000
400000
300000
200000
100000
0
Count
Denied
Certified
Requires Job Training for Employee

0.2.7 Impact of unit_of_wage on Visa status

[35]:
```
#group data by unit_of_wage and count the case_status
df.groupby('unit_of_wage')['case_status'].value_counts(normalize=True)
```

[35]:
```
unit_of_wage  case_status
Hour          Denied         0.653686
              Certified      0.346314
Month         Certified      0.617978
              Denied         0.382022
Week          Certified      0.621324
              Denied         0.378676
Year          Certified      0.698850
              Denied         0.301150
Name: proportion, dtype: float64
```

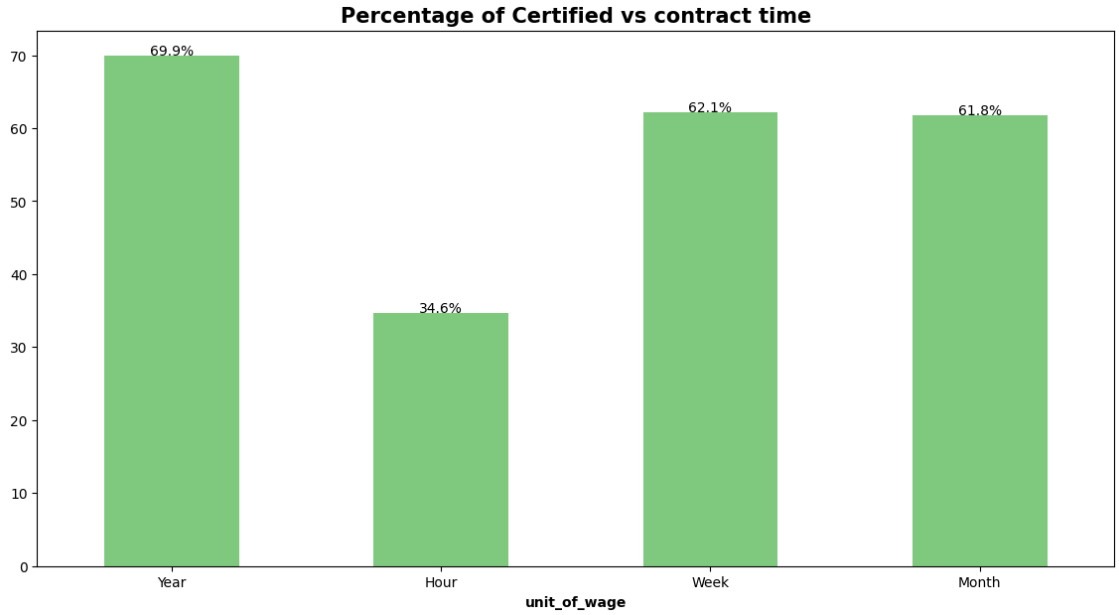
[36]:
```
plt.subplots(figsize=(14,7))
sns.countplot(x="unit_of_wage",hue="case_status", data=df,ec='k',␣
 ↪palette='Accent')
plt.title("No of Employees in company vs Visa Status",␣
 ↪weight="bold",fontsize=20, pad=20)
plt.xlabel("No of Employees", weight="bold", fontsize=16)
plt.ylabel("Count", weight="bold", fontsize=12)
plt.show()
```



[37]:
```
plt.figure(figsize=[14,7])
```

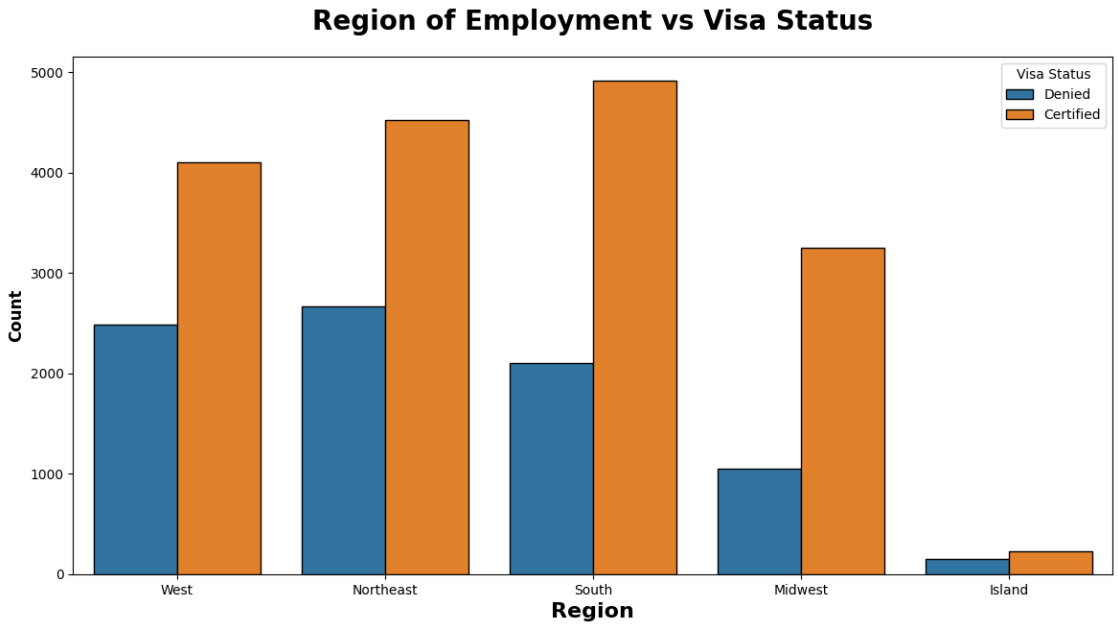
```
(100*df2[df2["case_status"].isin(['Certified'])]['unit_of_wage'].value_counts()/
 ↪df2['unit_of_wage'].value_counts()).plot(
    kind='bar',stacked=True , colormap='Accent')
plt.title("Percentage of Certified vs contract time" , fontsize = 15,␣
 ↪fontweight ='bold'  )
order1 = (100*df2[df2["case_status"].isin(['Certified'])]['unit_of_wage'].
 ↪value_counts()/df2['unit_of_wage'].value_counts())
for n in range(order1.shape[0]):
    count = order1[n]
    strt='{:0.1f}%'.format(count)
    plt.text(n,count+0.1,strt,ha='center')

plt.xlabel('unit_of_wage' , fontweight ='bold')
plt.xticks(rotation=0)
plt.show()
```



0.2.8 4.7 Does Region of employment has impact on Visa status ?

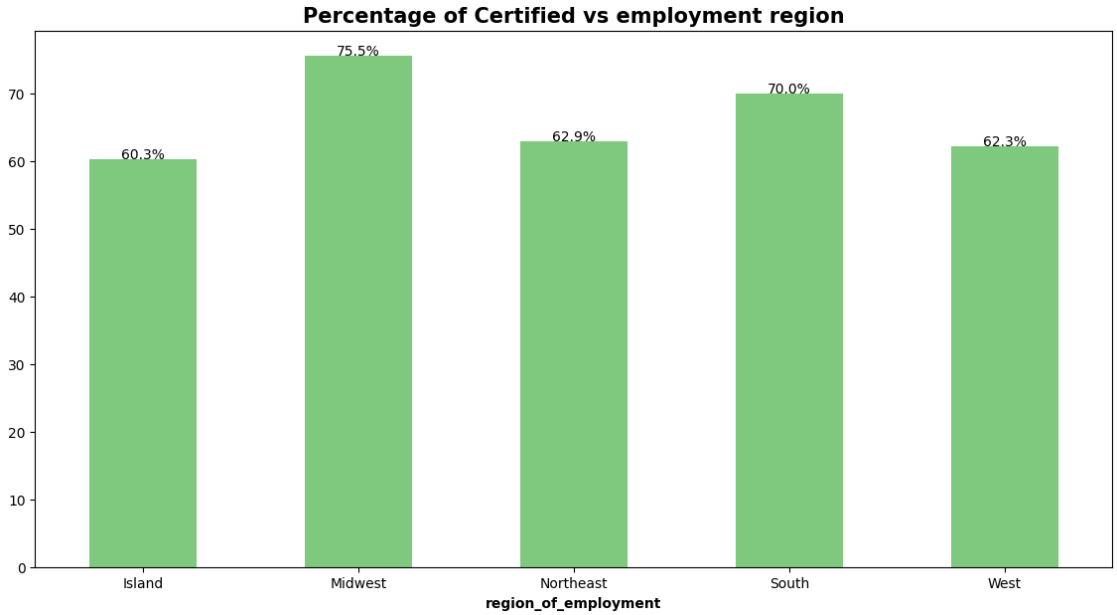
[38]:
```
plt.subplots(figsize=(14,7))
sns.countplot(x="region_of_employment",hue="case_status", data=df,ec='k')
plt.title("Region of Employment vs Visa Status", weight="bold",fontsize=20,␣
 ↪pad=20)
plt.xlabel("Region", weight="bold", fontsize=16)
plt.ylabel("Count", weight="bold", fontsize=12)
plt.legend(title="Visa Status", fancybox=True)
plt.show()
```



[39]:
```
plt.figure(figsize=[14,7])

(100*df2[df2["case_status"].isin(['Certified'])]['region_of_employment'].
 ↪value_counts()/df2['region_of_employment'].value_counts()).plot(
    kind='bar',stacked=True , colormap='Accent')
plt.title("Percentage of Certified vs employment region" , fontsize = 15,␣
 ↪fontweight ='bold'  )
order1 = (100*df2[df2["case_status"].
 ↪isin(['Certified'])]['region_of_employment'].value_counts()/
 ↪df2['region_of_employment'].value_counts())
for n in range(order1.shape[0]):
    count = order1[n]
    strt='{:0.1f}%'.format(count)
    plt.text(n,count+0.1,strt,ha='center')

plt.xlabel('region_of_employment' , fontweight ='bold')
plt.xticks(rotation=0)
plt.show()
```

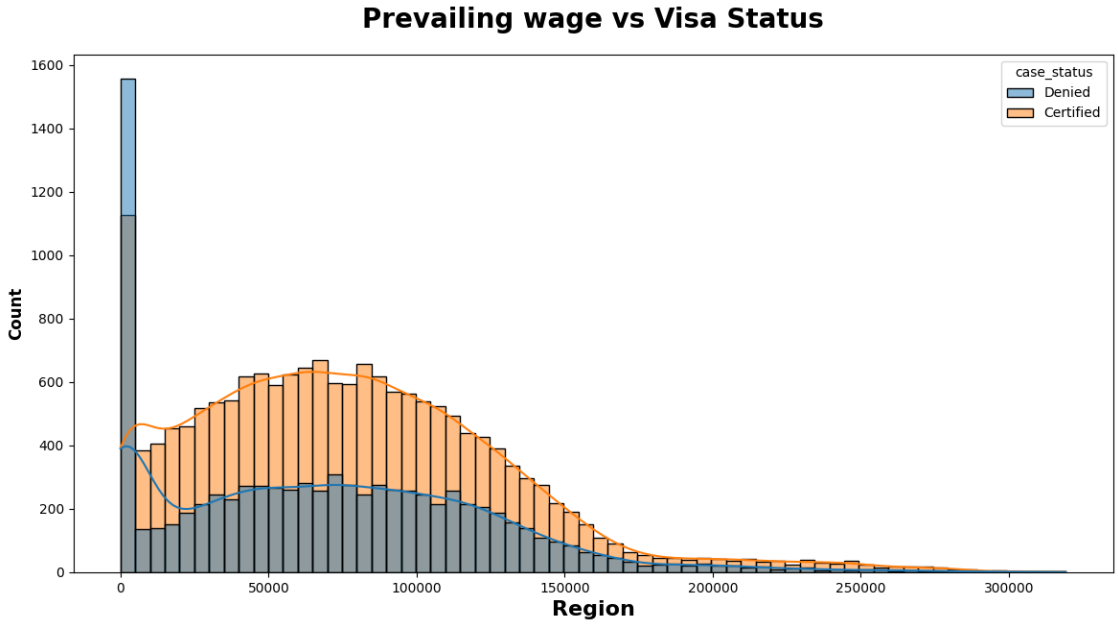


0.2.9 4.8 Does Prevailing wage has any impact on Visa status ?

[40]:
```
df.groupby('prevailing_wage').case_status.value_counts()
```

[40]:
```
prevailing_wage  case_status
2.1367           Certified      1
2.9561           Denied         1
3.0031           Denied         1
3.3188           Denied         1
3.4889           Certified      1
                                ..
311734.4900      Certified      1
314156.0600      Certified      1
315497.6000      Certified      1
318446.0500      Certified      1
319210.2700      Denied         1
Name: count, Length: 25462, dtype: int64
```

[41]:
```
plt.subplots(figsize=(14,7))
sns.histplot(x="prevailing_wage",hue="case_status", data=df, kde=True)
plt.title("Prevailing wage vs Visa Status", weight="bold",fontsize=20, pad=20)
plt.xlabel("Region", weight="bold", fontsize=16)
plt.ylabel("Count", weight="bold", fontsize=12)
plt.show()
```



4.8.1 Prevailing wage based on Education

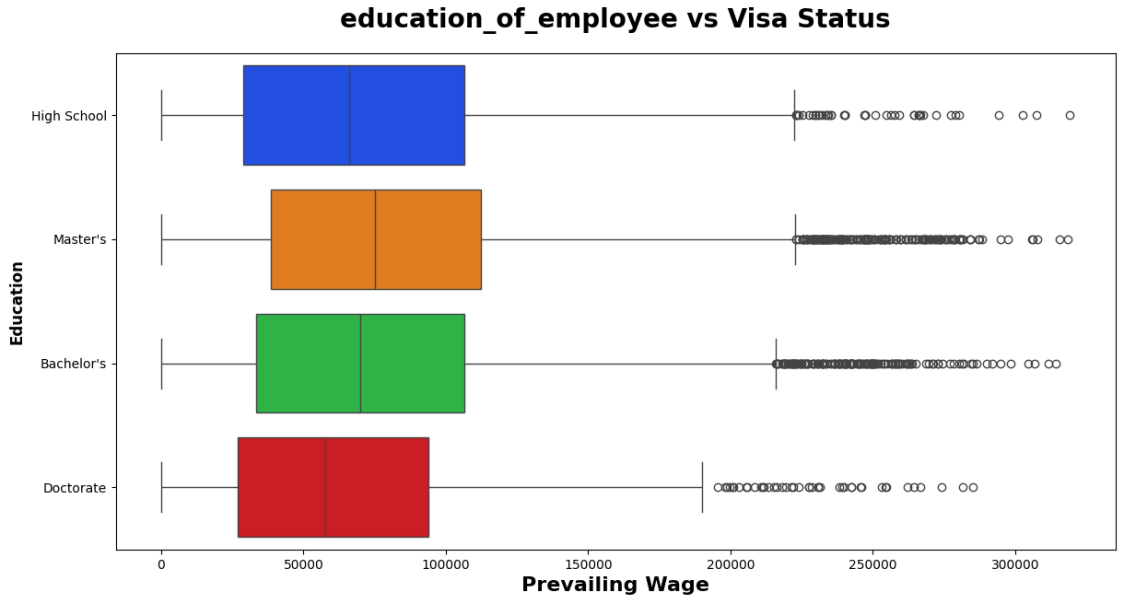
[42]:
```
print('Average Prevailing wage based on Employee education')
df.groupby('education_of_employee')['prevailing_wage'].mean().to_frame().
  ↪sort_values(by='prevailing_wage',ascending=False)
```

```
Average Prevailing wage based on Employee education
```

[42]:

education_of_employee	prevailing_wage
Master's	78843.057843
Bachelor's	73405.443735
High School	71582.147756
Doctorate	64561.076657

[43]:
```
plt.subplots(figsize=(14,7))
sns.boxplot(y="education_of_employee",x = "prevailing_wage",␣
  ↪data=df,palette='bright')
plt.title("education_of_employee vs Visa Status", weight="bold",fontsize=20,␣
  ↪pad=20)
plt.xlabel("Prevailing Wage", weight="bold", fontsize=16)
plt.ylabel("Education", weight="bold", fontsize=12)
plt.show()
```



4.8.2 Prevailing wage based on Job experience

[44]:
```
print('Median Prevailing wage based on Job experience')
df.groupby('has_job_experience')['prevailing_wage'].median().to_frame().
  ↪sort_values(by='prevailing_wage',ascending=False)
```

```
Median Prevailing wage based on Job experience
```

[44]:

has_job_experience	prevailing_wage
N	72602.290
Y	69033.665

4.8.3 Prevailing wage based on Continent

[45]:
```
print('Average Prevailing wage based on Continent')
df.groupby('continent')['prevailing_wage'].mean().to_frame().
  ↪sort_values(by='prevailing_wage',ascending=False)
```

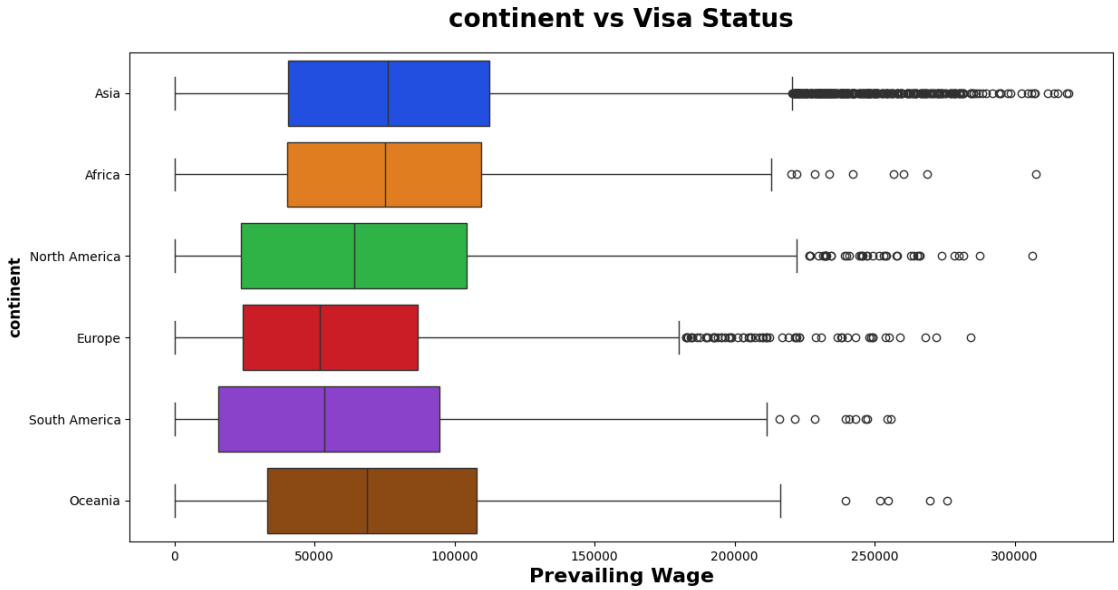
```
Average Prevailing wage based on Continent
```

[45]:

continent	prevailing_wage
Asia	79543.021780
Africa	77425.923450
Oceania	75994.276719
North America	68066.319257
South America	60209.575314

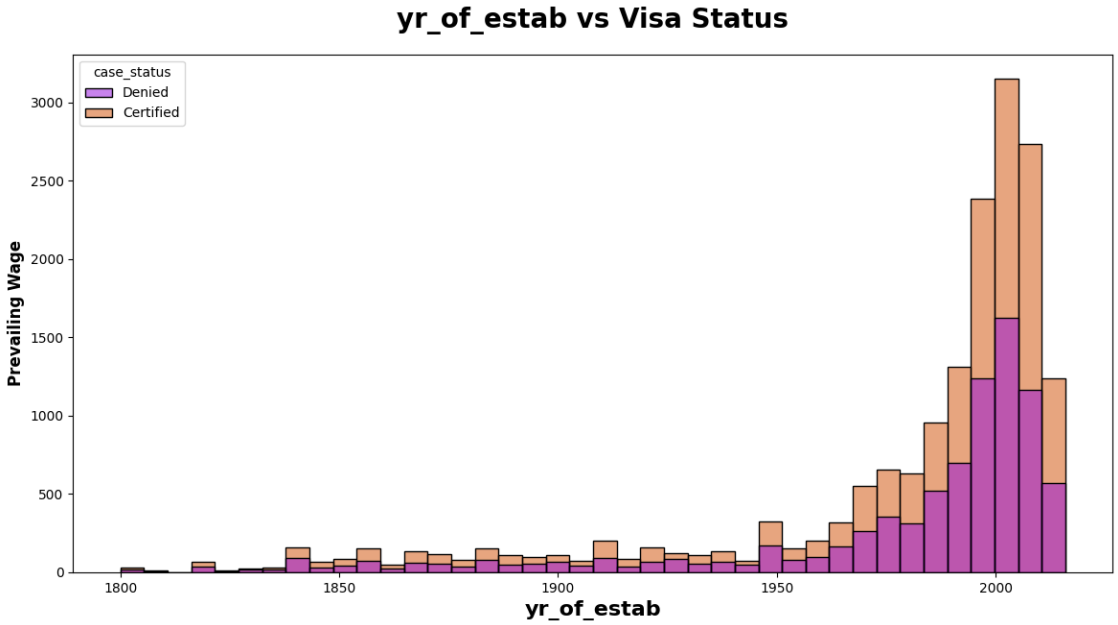
```
Europe              59842.925899
```

[46]:
```
plt.subplots(figsize=(14,7))
sns.boxplot(y="continent",x = "prevailing_wage", data=df,palette='bright')
plt.title("continent vs Visa Status", weight="bold",fontsize=20, pad=20)
plt.xlabel("Prevailing Wage", weight="bold", fontsize=16)
plt.ylabel("continent", weight="bold", fontsize=12)
plt.show()
```



0.2.10 4.9 Year of Establishment

[47]:
```
plt.subplots(figsize=(14,7))
sns.histplot(x = "yr_of_estab", data=df,palette='gnuplot', bins=40,␣
 ↪hue='case_status')
plt.title("yr_of_estab vs Visa Status", weight="bold",fontsize=20, pad=20)
plt.xlabel("yr_of_estab", weight="bold", fontsize=16)
plt.ylabel("Prevailing Wage", weight="bold", fontsize=12)
plt.show()
```



1 Final Report

- case_id column can be dropped as it is an ID.
- requires_job_training column can be dropped as it doesn't have much impact on target variable, Proved in visualization and chi2 test.
- no_of_employees, prevailing_wage columns have outllier which should be handled.
- continent columns has few unique values with very less count, which can be made as others
- Target column case_status is imbalanced can be handled before model building.

[48]:
```
df.loc[1,:]
```

[48]:
```
case_id                    EZYV02
continent                    Asia
education_of_employee     Master's
has_job_experience               Y
requires_job_training            N
no_of_employees               2412
yr_of_estab                   2002
region_of_employment     Northeast
prevailing_wage           83425.65
unit_of_wage                 Year
full_time_position               Y
case_status              Certified
Name: 1, dtype: object
```

[49]:
```
df.loc[1]
```

[49]:
```
case_id                     EZYV02
continent                     Asia
education_of_employee     Master's
has_job_experience               Y
requires_job_training            N
no_of_employees               2412
yr_of_estab                   2002
region_of_employment     Northeast
prevailing_wage           83425.65
unit_of_wage                  Year
full_time_position               Y
case_status              Certified
Name: 1, dtype: object
```

[50]:
```
nf = [f for f in df.columns if df[f].dtype != "object"]

nf
len(nf)
```

[50]:
```
3
```

[51]:
```
print(" there are {} numerical features: {}".format( len(nf), nf))
```

```
 there are 3 numerical features: ['no_of_employees', 'yr_of_estab',
'prevailing_wage']
```

[52]:
```
cf = [ f for f in df.columns if df[f].dtype == "object"]
cf
```

[52]:
```
['case_id',
 'continent',
 'education_of_employee',
 'has_job_experience',
 'requires_job_training',
 'region_of_employment',
 'unit_of_wage',
 'full_time_position',
 'case_status']
```

[53]:
```
len(cf)
```

[53]:
```
9
```

[54]:
```
for c in cf:
    print(df[c].value_counts())
```

```
case_id
EZYV01       1
```

```
EZYV16995    1
EZYV16993    1
EZYV16992    1
EZYV16991    1
            ..
EZYV8492     1
EZYV8491     1
EZYV8490     1
EZYV8489     1
EZYV25480    1
Name: count, Length: 25480, dtype: int64
continent
Asia             16861
Europe            3732
North America     3292
South America      852
Africa             551
Oceania            192
Name: count, dtype: int64
education_of_employee
Bachelor's     10234
Master's        9634
High School     3420
Doctorate       2192
Name: count, dtype: int64
has_job_experience
Y    14802
N    10678
Name: count, dtype: int64
requires_job_training
N    22525
Y     2955
Name: count, dtype: int64
region_of_employment
Northeast    7195
South        7017
West         6586
Midwest      4307
Island        375
Name: count, dtype: int64
unit_of_wage
Year     22962
Hour      2157
Week       272
Month       89
Name: count, dtype: int64
full_time_position
Y    22773
```

```
N     2707
Name: count, dtype: int64
case_status
Certified    17018
Denied        8462
Name: count, dtype: int64
```

[55]:
```
for c in cf:
    print(df[c].value_counts(normalize=True)* 100 )
```

```
case_id
EZYV01       0.003925
EZYV16995    0.003925
EZYV16993    0.003925
EZYV16992    0.003925
EZYV16991    0.003925
               …
EZYV8492     0.003925
EZYV8491     0.003925
EZYV8490     0.003925
EZYV8489     0.003925
EZYV25480    0.003925
Name: proportion, Length: 25480, dtype: float64
continent
Asia             66.173469
Europe           14.646782
North America    12.919937
South America     3.343799
Africa            2.162480
Oceania           0.753532
Name: proportion, dtype: float64
education_of_employee
Bachelor's     40.164835
Master's       37.810047
High School    13.422292
Doctorate       8.602826
Name: proportion, dtype: float64
has_job_experience
Y    58.092622
N    41.907378
Name: proportion, dtype: float64
requires_job_training
N    88.402669
Y    11.597331
Name: proportion, dtype: float64
region_of_employment
Northeast    28.237834
South        27.539246
```

```
West         25.847724
Midwest      16.903454
Island        1.471743
Name: proportion, dtype: float64
unit_of_wage
Year     90.117739
Hour      8.465463
Week      1.067504
Month     0.349294
Name: proportion, dtype: float64
full_time_position
Y    89.375981
N    10.624019
Name: proportion, dtype: float64
case_status
Certified    66.789639
Denied       33.210361
Name: proportion, dtype: float64
```

[ ]: