```
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlib inline
%matplotlice
%matplotlice
%matplotlice
```

```
plt.figure(figsize=(8,5))
sns.countplot(x='Education', hue='Loan', data=df[df['Education'].isin(['Undergraduate', 'Graduate', 'Postgraduate'])])
plt.title('Education vs Loan')
plt.xlabel('Education')
plt.ylabel('Count')
plt.legend(title='Loan')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Education', hue='Personal Loan', data=df[df['Education'].isin(['Undergraduate', 'Graduate', 'Postgraduate'])])
plt.title('Education vs Personal Loan')
plt.xlabel('Education')
plt.ylabel('Count')
plt.legend(title='Personal Loan')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Education', hue='Mortgage', data=df[df['Education'].isin(['Undergraduate', 'Graduate', 'Postgraduate'])])
plt.title('Education vs Mortgage')
plt.xlabel('Education')
plt.ylabel('Count')
plt.legend(title='Mortgage')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Education', hue='Securities Account', data=df[df['Education'].isin(['Undergraduate', 'Graduate', 'Postgraduate'])])
plt.title('Education vs Securities Account')
plt.xlabel('Education')
plt.ylabel('Count')
plt.legend(title='Securities Account')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Education', hue='CD Account', data=df[df['Education'].isin(['Undergraduate', 'Graduate', 'Postgraduate'])])
plt.title('Education vs CD Account')
plt.xlabel('Education')
plt.ylabel('Count')
plt.legend(title='CD Account')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Education', hue='Online', data=df[df['Education'].isin(['Undergraduate', 'Graduate', 'Postgraduate'])])
plt.title('Education vs Online')
plt.xlabel('Education')
plt.ylabel('Count')
plt.legend(title='Online')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Education', hue='CreditCard', data=df[df['Education'].isin(['Undergraduate', 'Graduate', 'Postgraduate'])])
plt.title('Education vs CreditCard')
plt.xlabel('Education')
plt.ylabel('Count')
plt.legend(title='CreditCard')
plt.tight_layout()
plt.show()
```

```
plt.figure(figsize=(8,5))
sns.countplot(x='Family', hue='Loan', data=df[df['Family'].isin([1, 2, 3, 4])])
plt.title('Family vs Loan')
plt.xlabel('Family')
plt.ylabel('Count')
plt.legend(title='Loan')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Family', hue='Personal Loan', data=df[df['Family'].isin([1, 2, 3, 4])])
plt.title('Family vs Personal Loan')
plt.xlabel('Family')
plt.ylabel('Count')
plt.legend(title='Personal Loan')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Family', hue='Mortgage', data=df[df['Family'].isin([1, 2, 3, 4])])
plt.title('Family vs Mortgage')
plt.xlabel('Family')
plt.ylabel('Count')
plt.legend(title='Mortgage')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Family', hue='Securities Account', data=df[df['Family'].isin([1, 2, 3, 4])])
plt.title('Family vs Securities Account')
plt.xlabel('Family')
plt.ylabel('Count')
plt.legend(title='Securities Account')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Family', hue='CD Account', data=df[df['Family'].isin([1, 2, 3, 4])])
plt.title('Family vs CD Account')
plt.xlabel('Family')
plt.ylabel('Count')
plt.legend(title='CD Account')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Family', hue='Online', data=df[df['Family'].isin([1, 2, 3, 4])])
plt.title('Family vs Online')
plt.xlabel('Family')
plt.ylabel('Count')
plt.legend(title='Online')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Family', hue='CreditCard', data=df[df['Family'].isin([1, 2, 3, 4])])
plt.title('Family vs CreditCard')
plt.xlabel('Family')
plt.ylabel('Count')
plt.legend(title='CreditCard')
plt.tight_layout()
plt.show()
```

```
plt.figure(figsize=(8,5))
sns.countplot(x='Zip Code', hue='Loan', data=df[df['Zip Code'].isin(df['Zip Code'].value_counts().nlargest(10).index)])
plt.title('Zip Code vs Loan')
plt.xlabel('Zip Code')
plt.ylabel('Count')
plt.legend(title='Loan')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Zip Code', hue='Personal Loan', data=df[df['Zip Code'].isin(df['Zip Code'].value_counts().nlargest(10).index)])
plt.title('Zip Code vs Personal Loan')
plt.xlabel('Zip Code')
plt.ylabel('Count')
plt.legend(title='Personal Loan')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Zip Code', hue='Mortgage', data=df[df['Zip Code'].isin(df['Zip Code'].value_counts().nlargest(10).index)])
plt.title('Zip Code vs Mortgage')
plt.xlabel('Zip Code')
plt.ylabel('Count')
plt.legend(title='Mortgage')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Zip Code', hue='Securities Account', data=df[df['Zip Code'].isin(df['Zip Code'].value_counts().nlargest(10).index)])
plt.title('Zip Code vs Securities Account')
plt.xlabel('Zip Code')
plt.ylabel('Count')
plt.legend(title='Securities Account')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Zip Code', hue='CD Account', data=df[df['Zip Code'].isin(df['Zip Code'].value_counts().nlargest(10).index)])
plt.title('Zip Code vs CD Account')
plt.xlabel('Zip Code')
plt.ylabel('Count')
plt.legend(title='CD Account')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Zip Code', hue='Online', data=df[df['Zip Code'].isin(df['Zip Code'].value_counts().nlargest(10).index)])
plt.title('Zip Code vs Online')
plt.xlabel('Zip Code')
plt.ylabel('Count')
plt.legend(title='Online')
plt.tight_layout()
plt.show()
plt.figure(figsize=(8,5))
sns.countplot(x='Zip Code', hue='CreditCard', data=df[df['Zip Code'].isin(df['Zip Code'].value_counts().nlargest(10).index)])
plt.title('Zip Code vs CreditCard')
plt.xlabel('Zip Code')
plt.ylabel('Count')
plt.legend(title='CreditCard')
plt.tight_layout()
plt.show()
```



```
LOAN = df['Loan']
ZIP = df['Zip Code']
ct = pd.crosstab(ZIP, LOAN, normalize='index')[1].sort_values(ascending=False)[:5]
print(ct)
```

```
Zip Code
90089    0.333333
90095    0.333333
90245    0.333333
90291    0.333333
92717    0.333333
Name: 1, dtype: float64
```

```
plt.pie(ct, labels=ct.index, autopct='%1.1f%%')
plt.title('Top 5 Zip Codes with Highest Loan')
plt.tight_layout()
plt.show()
```



```
plt.bar(ct.index, ct, color='skyblue')
plt.xlabel('Zip Code')
plt.ylabel('Loan')
plt.title('Top 5 Zip Codes with Highest Loan')
plt.tight_layout()
plt.show()
```



```
Zip Code
90089    0.333333
90095    0.333333
90245    0.333333
90291    0.333333
92717    0.333333
Name: 1, dtype: float64
```

```
plt.pie(ct, labels=ct.index, autopct='%1.1f%%')
plt.title('Top 5 Zip Codes with Highest Loan')
plt.tight_layout()
plt.show
```



```
plt.bar(ct.index, ct, color='skyblue')
plt.xlabel('Zip Code')
plt.ylabel('Loan')
plt.title('Top 5 Zip Codes with Highest Loan')
plt.tight_layout()
plt.show
```



```
df['Zip Code'].value_counts()[df['Zip Code'].value_counts().index[:5]]
```

```
Zip Code
92093    17
94720    169
94305    127
95616    116
90089     32
Name: count, dtype: int64
```

```
plt.pie(df['Zip Code'].value_counts()[df['Zip Code'].value_counts().index[:5]], labels=df['Zip Code'].value_counts()[df['Zip Code'].value_counts().index[:5]].index, autopct='%1.1f%%')
plt.title('Top 5 Zip Codes')
plt.tight_layout()
plt.show
```



```
plt.bar(df['Zip Code'].value_counts()[df['Zip Code'].value_counts().index[:5]].index, df['Zip Code'].value_counts()[df['Zip Code'].value_counts().index[:5]])
plt.xlabel('Zip Code')
plt.ylabel('Count')
plt.title('Top 5 Zip Codes')
plt.tight_layout()
plt.show
```



```
df[df['Zip Code'].isin(df['Zip Code'].value_counts().index[:5])].groupby('Zip Code')[['Income', 'Family', 'CCAvg', 'Education', 'Mortgage', 'Personal Loan', 'Securities Account', 'CD Account', 'Online', 'CreditCard']].mean()
```

```
                Income    Family     CCAvg  Education  Mortgage  Personal Loan  Securities Account  CD Account    Online  CreditCard
Zip Code
90089        92.312500  2.312500  2.312500   2.000000  0.000000       0.000000            0.000000    0.000000  0.687500    0.312500
92093        81.294118  2.470588  2.470588   2.000000  0.000000       0.000000            0.000000    0.000000  0.705882    0.352941
94305        93.748031  2.496063  2.496063   2.000000  0.000000       0.000000            0.000000    0.000000  0.708661    0.346457
94720        93.816568  2.508876  2.508876   2.000000  0.000000       0.000000            0.000000    0.000000  0.698225    0.349112
95616        93.681034  2.491379  2.491379   2.000000  0.000000       0.000000            0.000000    0.000000  0.698276    0.344828
```

Income of grade 1

Income of Income college

Income of Bachelors

Income of Masters

Income of Doctorate

Income of 18

Income of 19

Income of 20

Income of 21

Income of 22

Income of 23

Income of 24

Income of 25

Income of 26

Income of 27

Income of 28

Income of 29

Income of 30

Income of 31

Income of 32

Income of 33

Income of 34

Income of 35

Income of 36

Income of 37

Income of 38

Income of 39

Income of 40


```
> ggplot(data=df,aes(x=age,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of age and income")+ylab("Proportion")+xlab("Age")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=workclass,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of workclass and income")+ylab("Proportion")+xlab("workclass")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=education,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of education and income")+ylab("Proportion")+xlab("education")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=education.num,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of education.num and income")+ylab("Proportion")+xlab("education.num")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=marital.status,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of marital.status and income")+ylab("Proportion")+xlab("marital.status")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=occupation,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of occupation and income")+ylab("Proportion")+xlab("occupation")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=relationship,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of relationship and income")+ylab("Proportion")+xlab("relationship")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=race,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of race and income")+ylab("Proportion")+xlab("race")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=sex,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of sex and income")+ylab("Proportion")+xlab("sex")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=hours.per.week,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of hours.per.week and income")+ylab("Proportion")+xlab("hours.per.week")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=native.country,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of native.country and income")+ylab("Proportion")+xlab("native.country")+scale_fill_manual(values=c("#ff0000","#0000ff"))
> ggplot(data=df,aes(x=income,fill=income))+geom_bar(position="fill")+theme_minimal()+ggtitle("Relationship of income and income")+ylab("Proportion")+xlab("income")+scale_fill_manual(values=c("#ff0000","#0000ff"))
```

Income of Private

Income of grad

Income of grade grp

Income of edu-num

Income of hours grp

Income of Gender

Income of Relationship

Income of gender updated

```
> df$income[df$income=="<=50K"]<-"Income<=50K"; df$income[df$income==">50K"]<-"Income>50K"
> table(df$income)

Income<=50K Income>50K
      24720       7841
> ggplot(df,aes(x=income))+
+ geom_bar()
```

```
> ggplot(df,aes(x=age))+geom_histogram(bins=30)+
+ facet_wrap(~income,scales="free_y")+theme_minimal()
```

Select the Optimal K value

```
inertia_list = []
for k in range(1,11):
    model = KMeans(n_clusters=k,random_state=42,n_jobs=-1)
    model.fit(df_scaled)
    inertia = model.inertia_
    print("k = {:2d}, inertia = {:f}".format(k, inertia))
    inertia_list.append(inertia)
```

```
k = 1, inertia = 1000.000000
k = 2, inertia = 690.188000
k = 3, inertia = 486.168000
k = 4, inertia = 353.160000
k = 5, inertia = 293.564000
k = 6, inertia = 248.228000
k = 7, inertia = 215.004000
k = 8, inertia = 187.914000
k = 9, inertia = 165.020000
k = 10, inertia = 147.060000
```

```
plot_inertia = pd.DataFrame(inertia_list,columns=['inertia'])
plot_inertia.index = range(1,11)
plot_inertia.plot(figsize=(12,6))
```

```
from sklearn.metrics import silhouette_score, calinski_harabasz_score
```

```
for k in range(2,11):
    model = KMeans(n_clusters=k,random_state=42,n_jobs=-1)
    model.fit(df_scaled)
    labels = model.predict(df_scaled)
    sil = silhouette_score(df_scaled, labels, metric='euclidean')
    ch = calinski_harabasz_score(df_scaled, labels)
    print("For n_clusters = {:2d}, silhouette score is {:0.3f}, calinski_harabasz score is {:0.3f)".\
          format(k, sil, ch))
```

```
For n_clusters = 2, silhouette score is 0.270, calinski_harabasz score is 1.000
For n_clusters = 3, silhouette score is 0.270, calinski_harabasz score is 1.000
For n_clusters = 4, silhouette score is 0.270, calinski_harabasz score is 1.000
For n_clusters = 5, silhouette score is 0.270, calinski_harabasz score is 1.000
For n_clusters = 6, silhouette score is 0.270, calinski_harabasz score is 1.000
For n_clusters = 7, silhouette score is 0.270, calinski_harabasz score is 1.000
For n_clusters = 8, silhouette score is 0.270, calinski_harabasz score is 1.000
For n_clusters = 9, silhouette score is 0.270, calinski_harabasz score is 1.000
For n_clusters = 10, silhouette score is 0.270, calinski_harabasz score is 1.000
```

```
model = KMeans(n_clusters=4,random_state=42,n_jobs=-1)
model.fit(df_scaled)
```

```
KMeans(n_clusters=4, random_state=42, n_jobs=-1)
```

```
labels = model.predict(df_scaled)
labels
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       
```