SEABORN

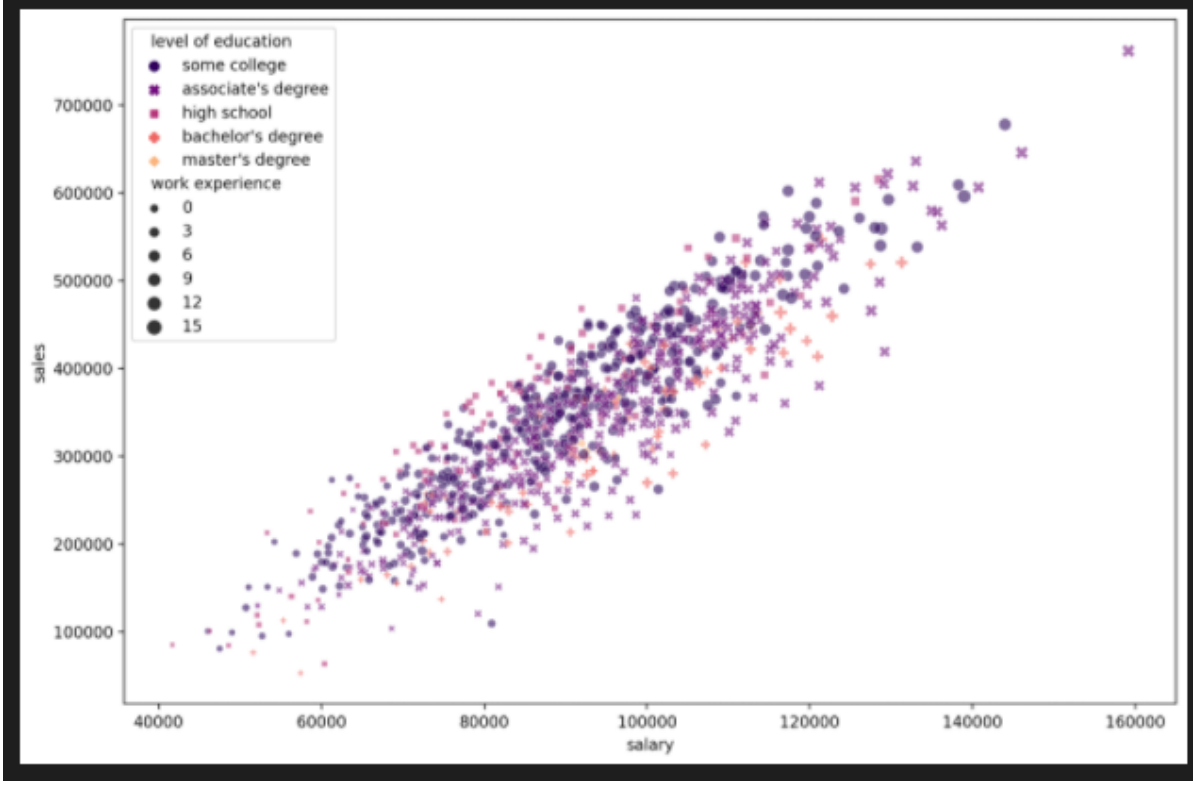
▼ Scatter Plot

- hue=”kolon adı” : değerleri kolon farklı bir kolon başlığına göre gruplandırır
  size="work experience” : büyük sayıyı büyük nokta yapar
  palette="magma” : noktaların renk paleti Choosing Colormaps in Matplotlib — Matplotlib 3.5.1 documentation
  s=100 : noktaların boyutu(hepsi)
  alpha=0.6 : noktaların şeffaflığı
  style="level of education” : noktaları şekillere göre gruplandırır

```
1  plt.figure(figsize=(12,8), dpi=200)
2  sns.scatterplot(x="salary",y="sales",
3   data=df, hue="level of education", size="work experience",
4   palette="magma", alpha=0.6, style="level of education")
```

✓ 0.6s Pytho

```
<AxesSubplot:xlabel='salary', ylabel='sales'>
```



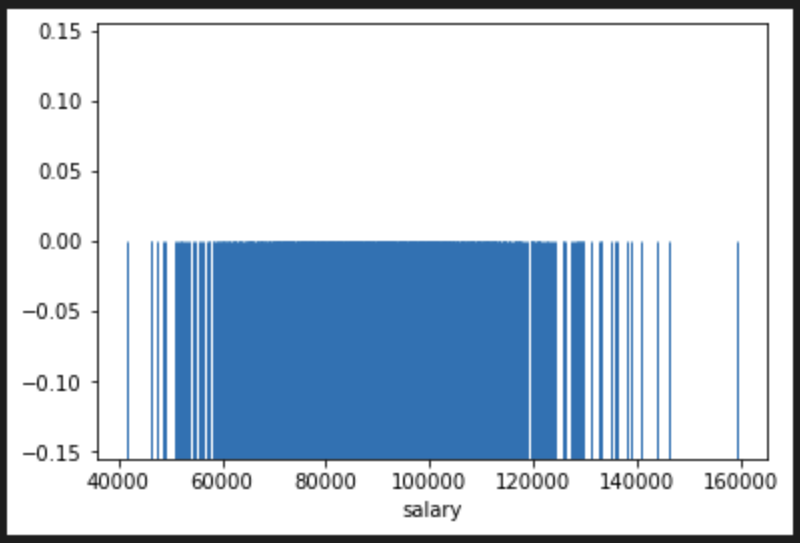
▼ Distribution Plots

- Rugplot

```
1  sns.rugplot(x='salary', data=df, height=0.5)
```

✓ 0.4s

```
<AxesSubplot:xlabel='salary'>
```

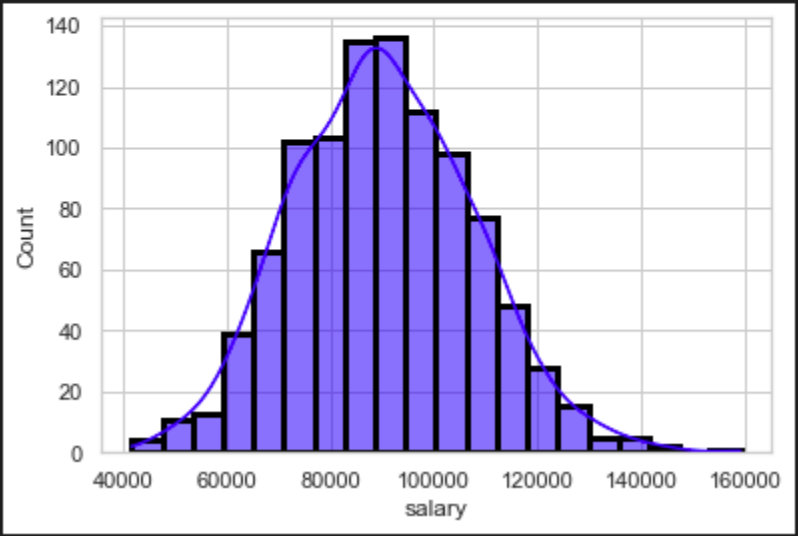


- Histogram
  bins=20 : kutu sayısı
  color="blue", edgecolor="black” : kutu rengi ve kenarlık rengi
  linewidth=3 : Kenarlık kalınlığı
  kde=True : kde dağılım eğrisi

```
1  sns.set(style="whitegrid")
2  sns.histplot(data=df, x="salary", bins=20,
3  color="blue", edgecolor="black", linewidth=3, kde=True)
```

✓ 0.4s

```
<AxesSubplot:xlabel='salary', ylabel='Count'>
```

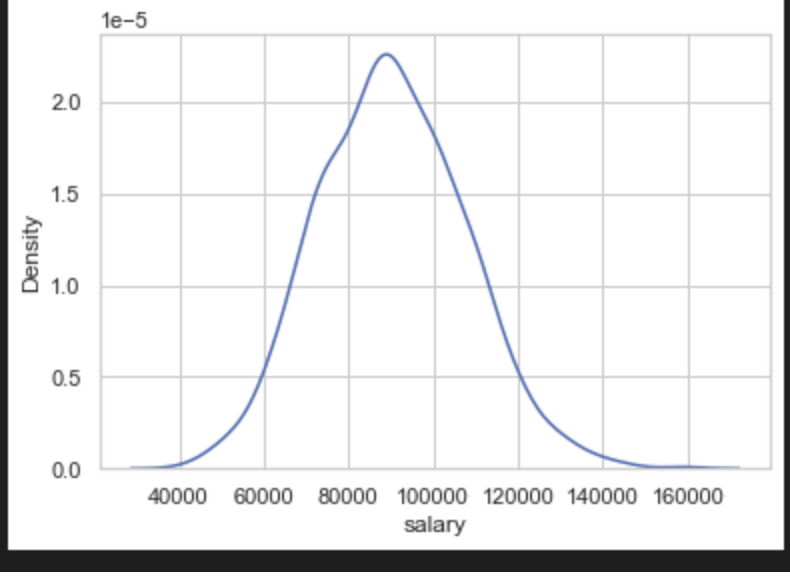


- KDE plot

```
1  sns.kdeplot(data=df, x="salary")
```

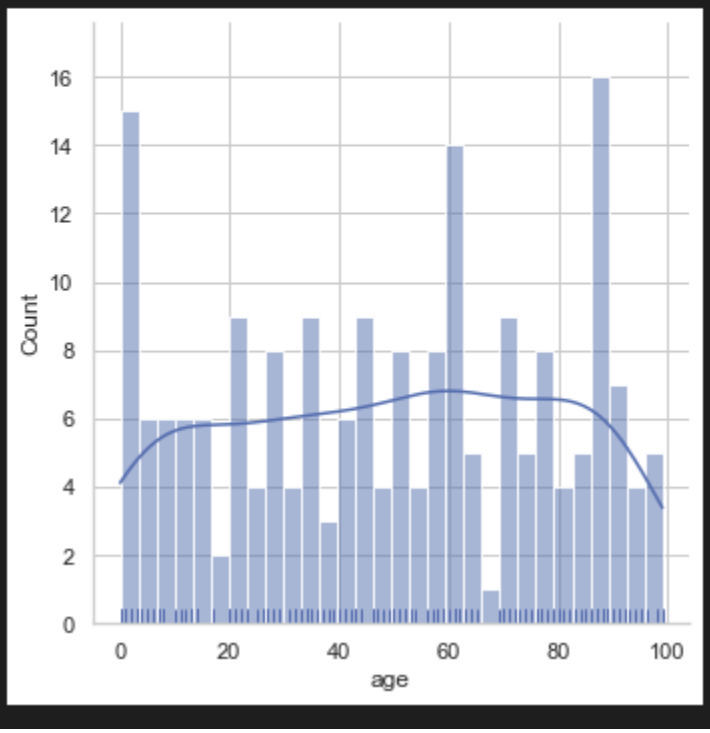
✓ 0.2s

```
<AxesSubplot:xlabel='salary', ylabel='Density'>
```



- Displot

```
1  sns.displot(data=sample_ages, x="age", rug=True,
2  |bins=30, kde=True)
✓ 0.6s
<seaborn.axisgrid.FacetGrid at 0x27b6d867970>
```

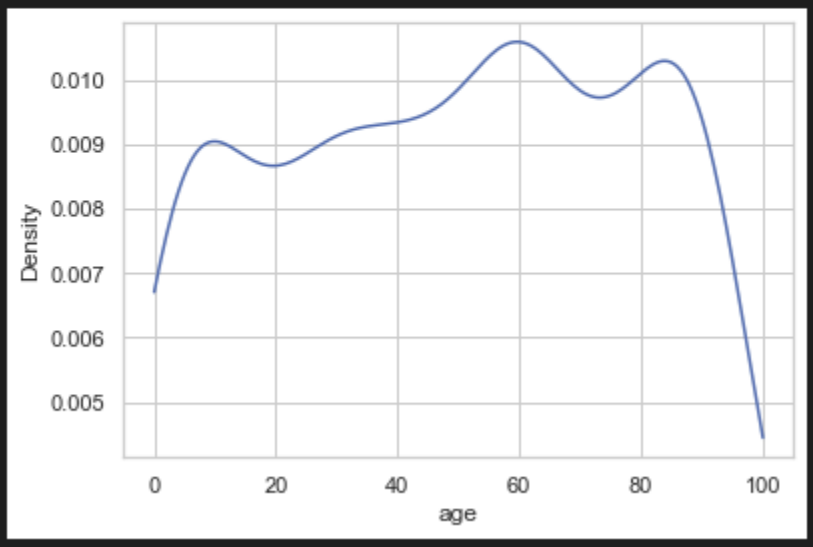


- clip=[0,100] : KDE'yi keser 0, 100 arasında
  bw_adjust=0.8 : bandwidth adjust : KDE'nin hassasiyetini ayarlar

```
1  sns.kdeplot(data=sample_ages, x="age",
2  clip=[0,100], bw_adjust=0.8)
```

✓ 0.3s

```
<AxesSubplot:xlabel='age', ylabel='Density'>
```



▼ Categorical Plots

- Categorical plot

```
1  df["division"].value_counts()
```

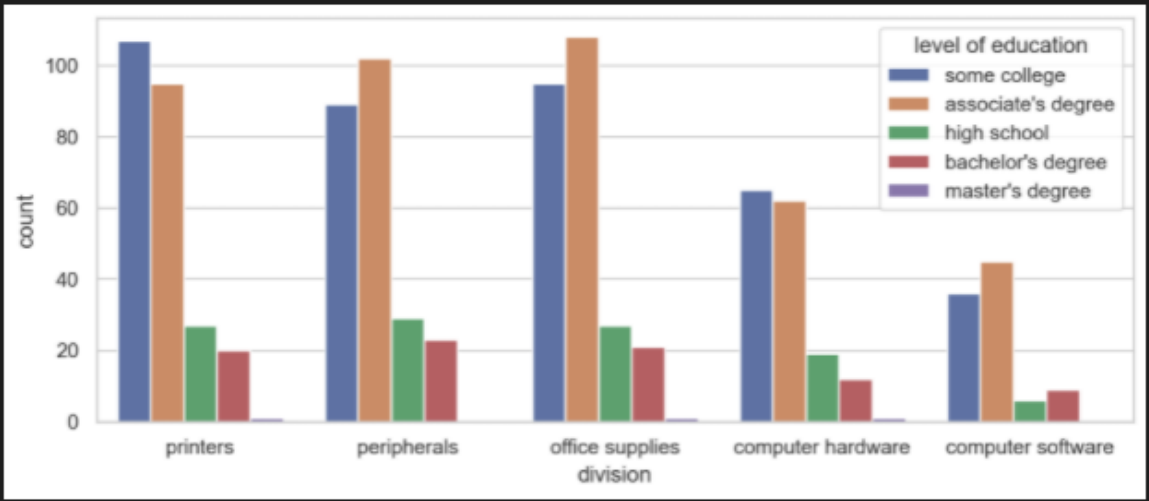
✓ 0.5s Python

```
office supplies      252
printers             250
peripherals          243
computer hardware    159
computer software     96
Name: division, dtype: int64
```

```
1  plt.figure(figsize=(10,4), dpi=200)
2  sns.countplot(data=df, x="division", hue="level of education")
3  # plt.ylim(90,260)
```

✓ 0.4s Python

```
<AxesSubplot:xlabel='division', ylabel='count'>
```

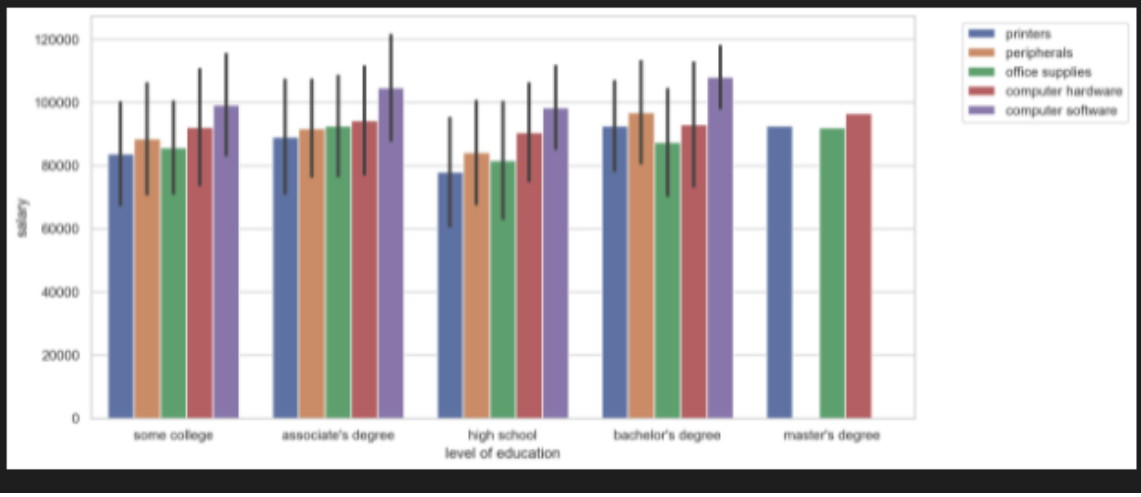


- estimator=np.mean : ortalama değerlere göre siyah çubukları çekiyor
  ci="sd" : confidence interval, “sd” standart deviation kadar yukarı aşağı oynar
  plt.legend(bbox_to_anchor=(1.05,1)) : legend yeri değiştirir

```
1  plt.figure(figsize=(12,6), dpi=200)
2  sns.barplot(data=df, x="level of education",
3  y="salary", estimator=np.mean, ci="sd", hue="division")
4  plt.legend(bbox_to_anchor=(1.05,1))
```

✓ 0.7s Python

```
<matplotlib.legend.Legend at 0x27b6d821a30>
```

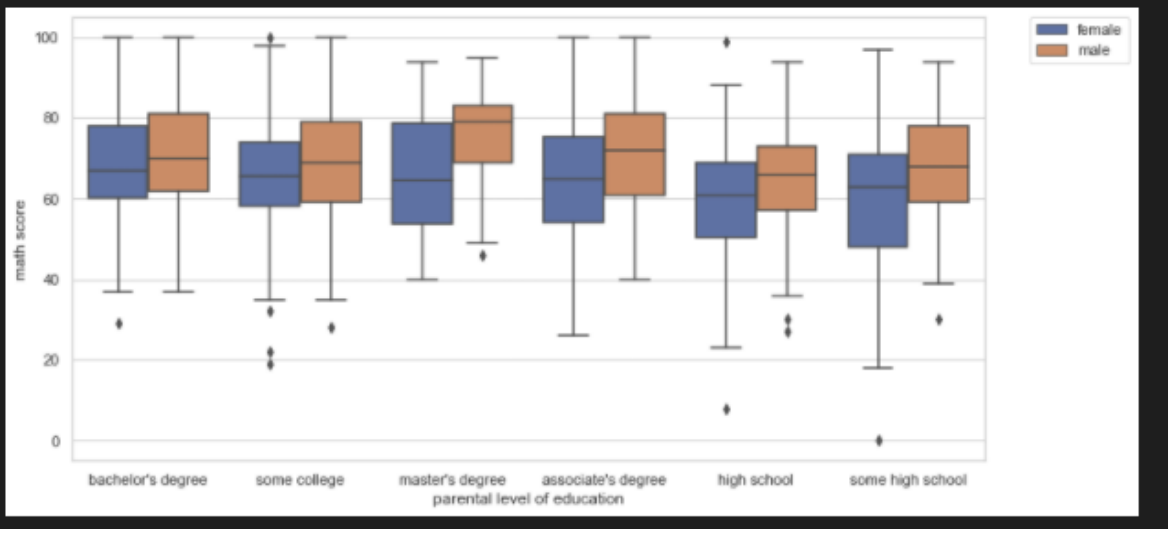


- Box Plot

```python
plt.figure(figsize=(12,6))
sns.boxplot(x='parental level of education',
    y='math score',data=df,hue='gender')
# Optional move the legend outside
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
```

0.4s Python

```
<matplotlib.legend.Legend at 0x27b6fac4070>
```

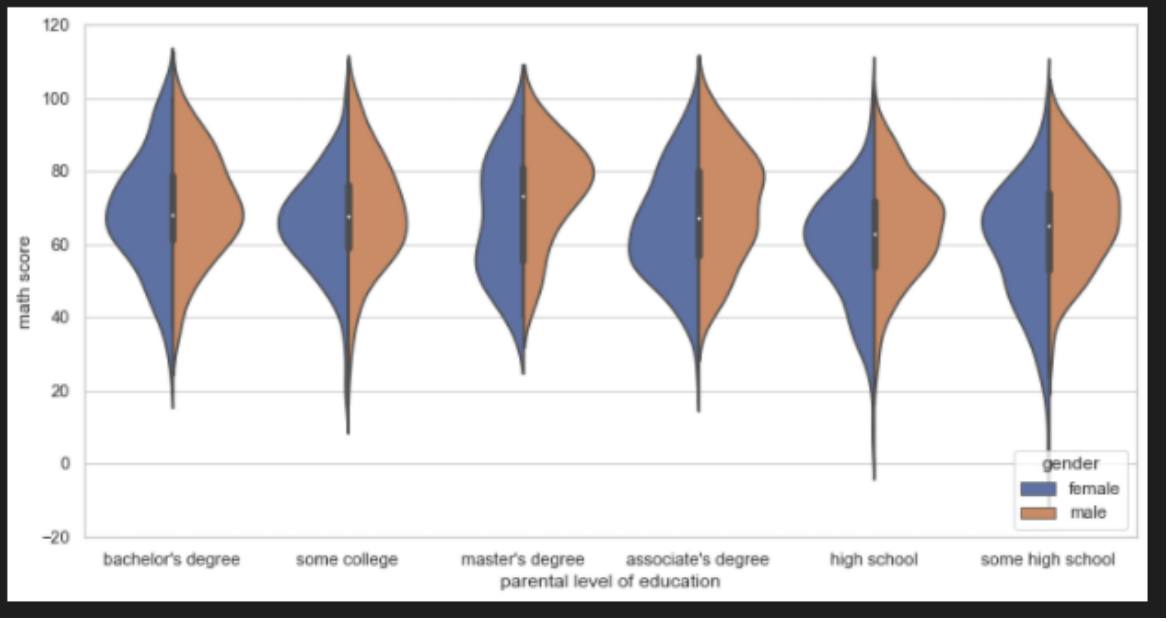


- Violin Plot
  split=True sağ sol farklı grafik
  bw=0.3 : bandwidth detay. 0.1 detaylı 1 çok kaba

```
1  plt.figure(figsize=(12,6))
2  sns.violinplot(x='parental level of education',
3  |  y='math score',data=df,hue='gender',split=True)
```

✓ 0.3s Python

```
<AxesSubplot:xlabel='parental level of education', ylabel='math score'>
```

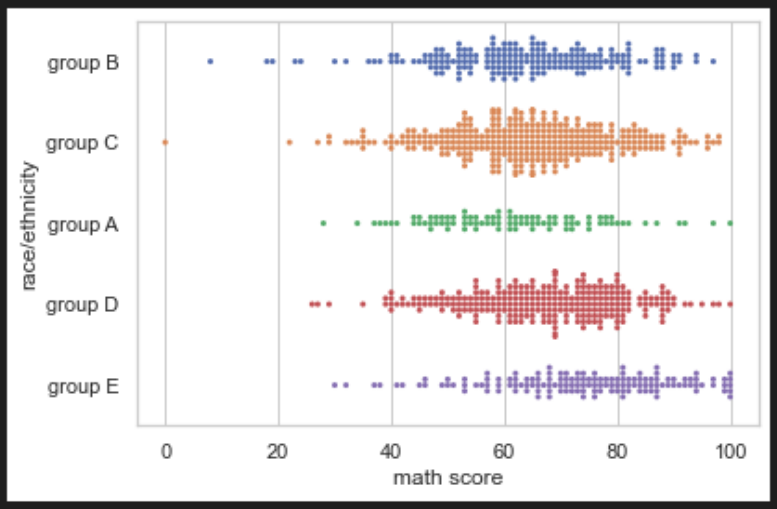


- Swarm Plot

```
1  sns.swarmplot(x='math score',y='race/ethnicity',data=df,size=3)
```

✓ 0.4s

```
<AxesSubplot:xlabel='math score', ylabel='race/ethnicity'>
```

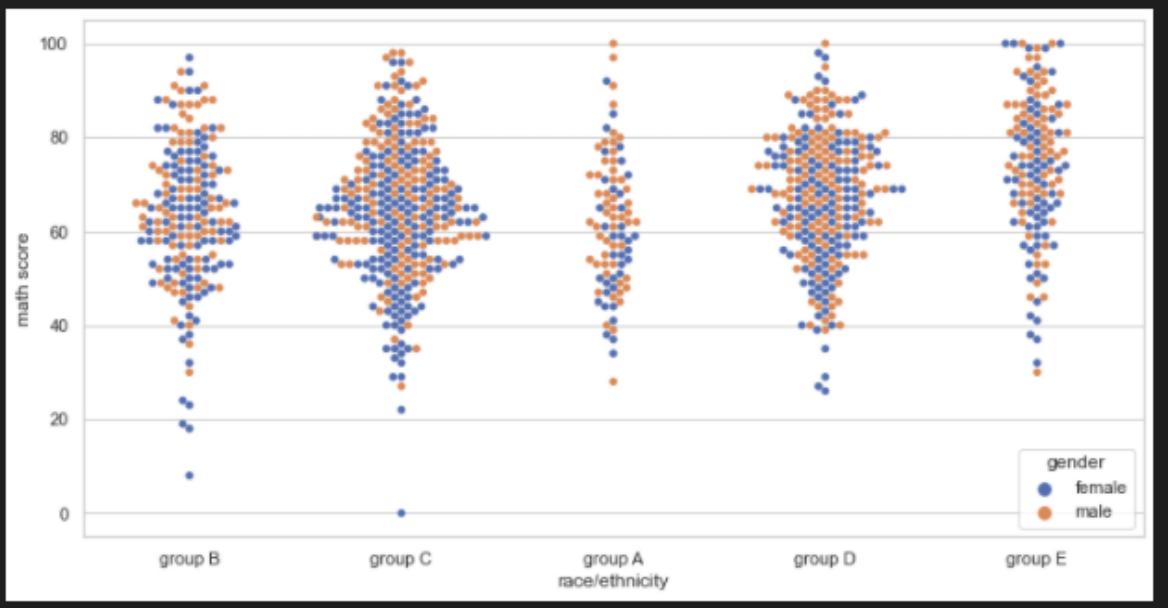


- Swarm Plot

```
1  plt.figure(figsize=(12,6))
2  sns.swarmplot(x='race/ethnicity',
3  | y='math score',data=df,hue='gender')
```

✓ 0.7s Python

```
<AxesSubplot:xlabel='race/ethnicity', ylabel='math score'>
```



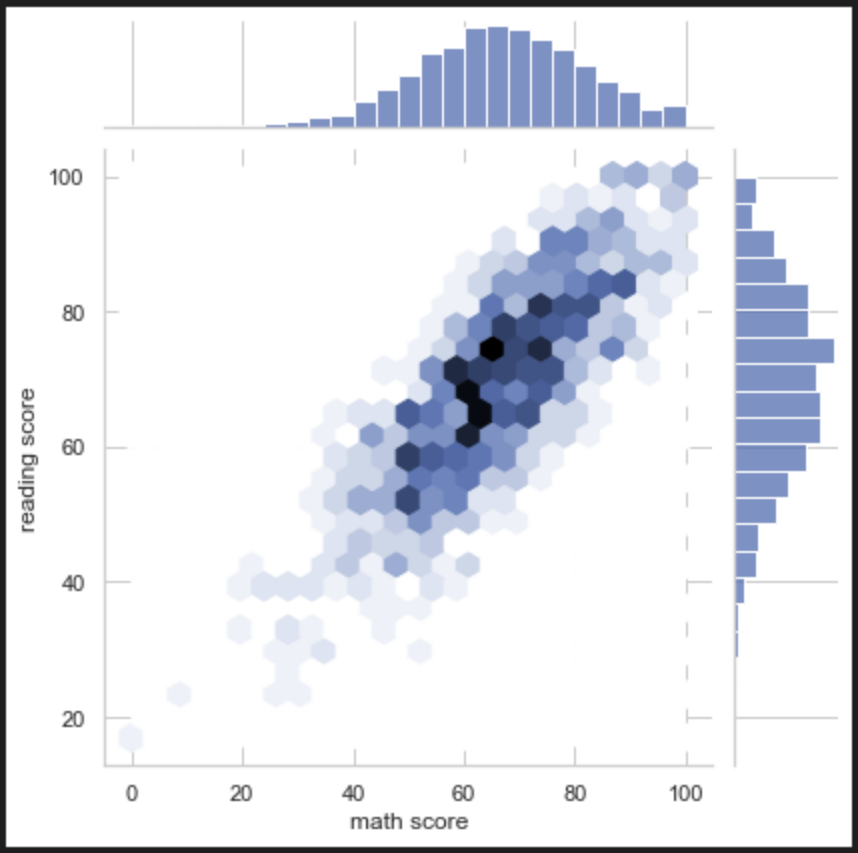
▼ Comparison Plots

- Joint plot, Hex

```
1  sns.jointplot(data=df, x="math score",y="reading score",
2  kind="hex")
```

✓ 0.9s

```
<seaborn.axisgrid.JointGrid at 0x27b70c0bfa0>
```

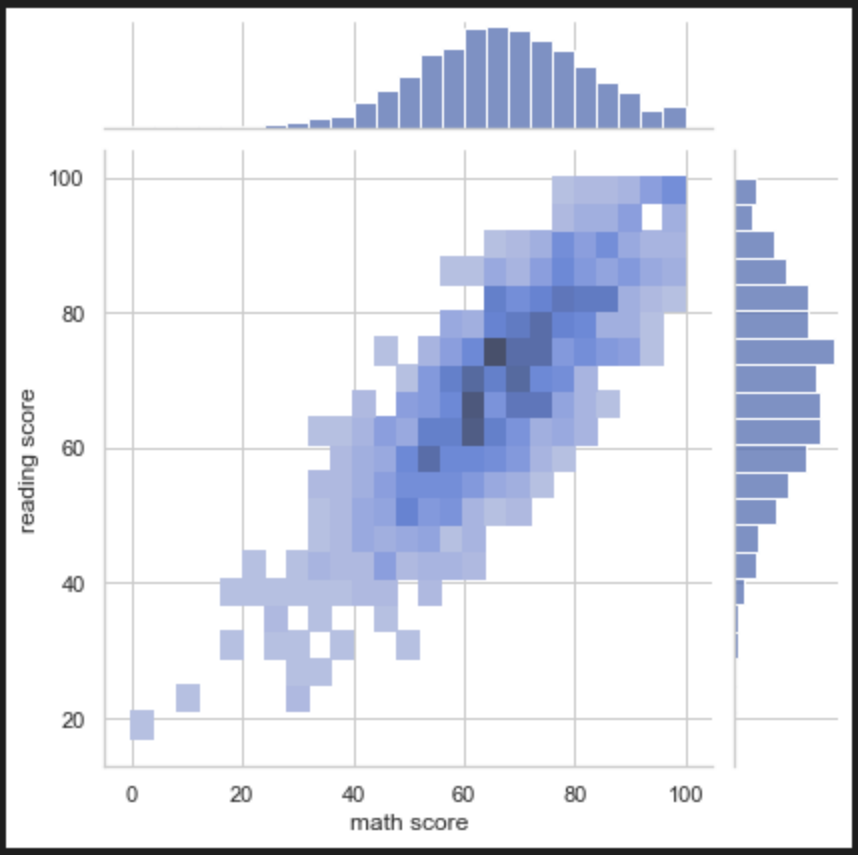


- joint plot, Histogram

```
1  sns.jointplot(data=df, x="math score",y="reading score",
2  kind="hist")
```

✓ 0.6s

```
<seaborn.axisgrid.JointGrid at 0x27b6fa6d100>
```

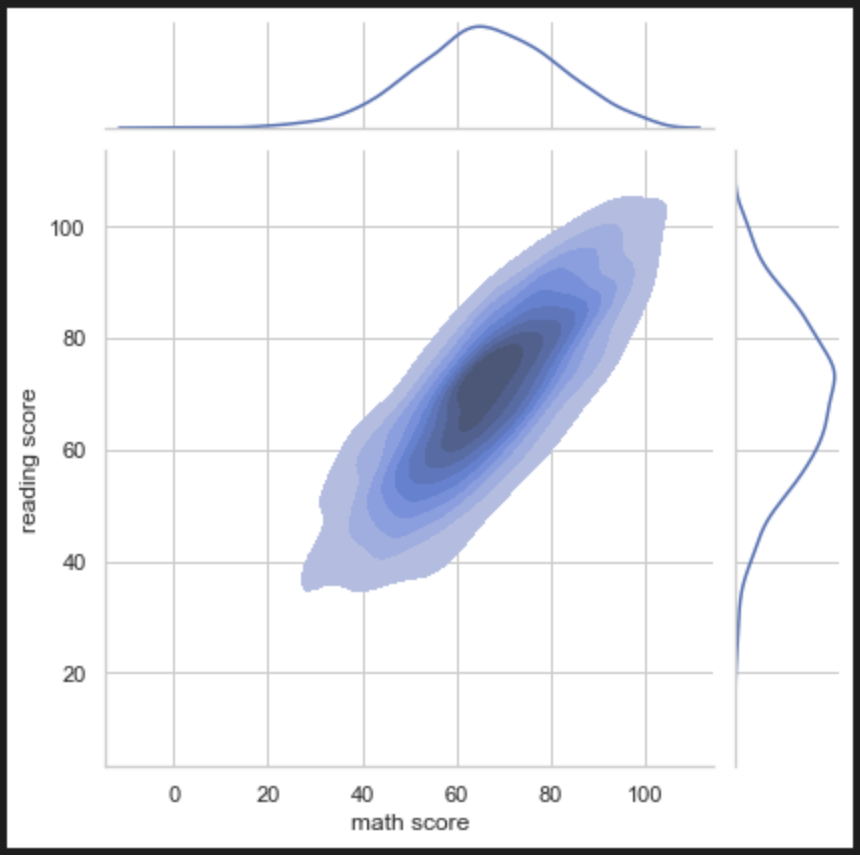


- Joint plot, KDE
  shade=True : izohipsleri koyulaşacak şekilde yapar

```
1  sns.jointplot(data=df, x="math score",y="reading score",
2  kind="kde", shade=True)
```

✓ 1.4s

```
<seaborn.axisgrid.JointGrid at 0x27b75131850>
```

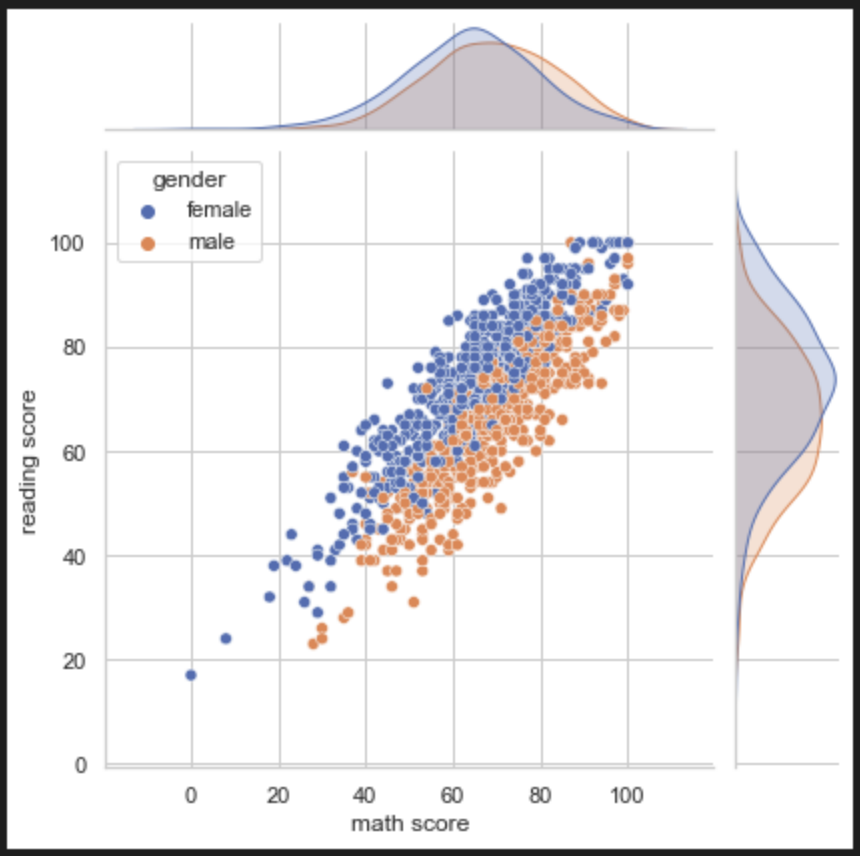


- Joint plot, Scatter

```
1 sns.jointplot(data=df, x="math score",y="reading score",
2 | hue="gender")
```

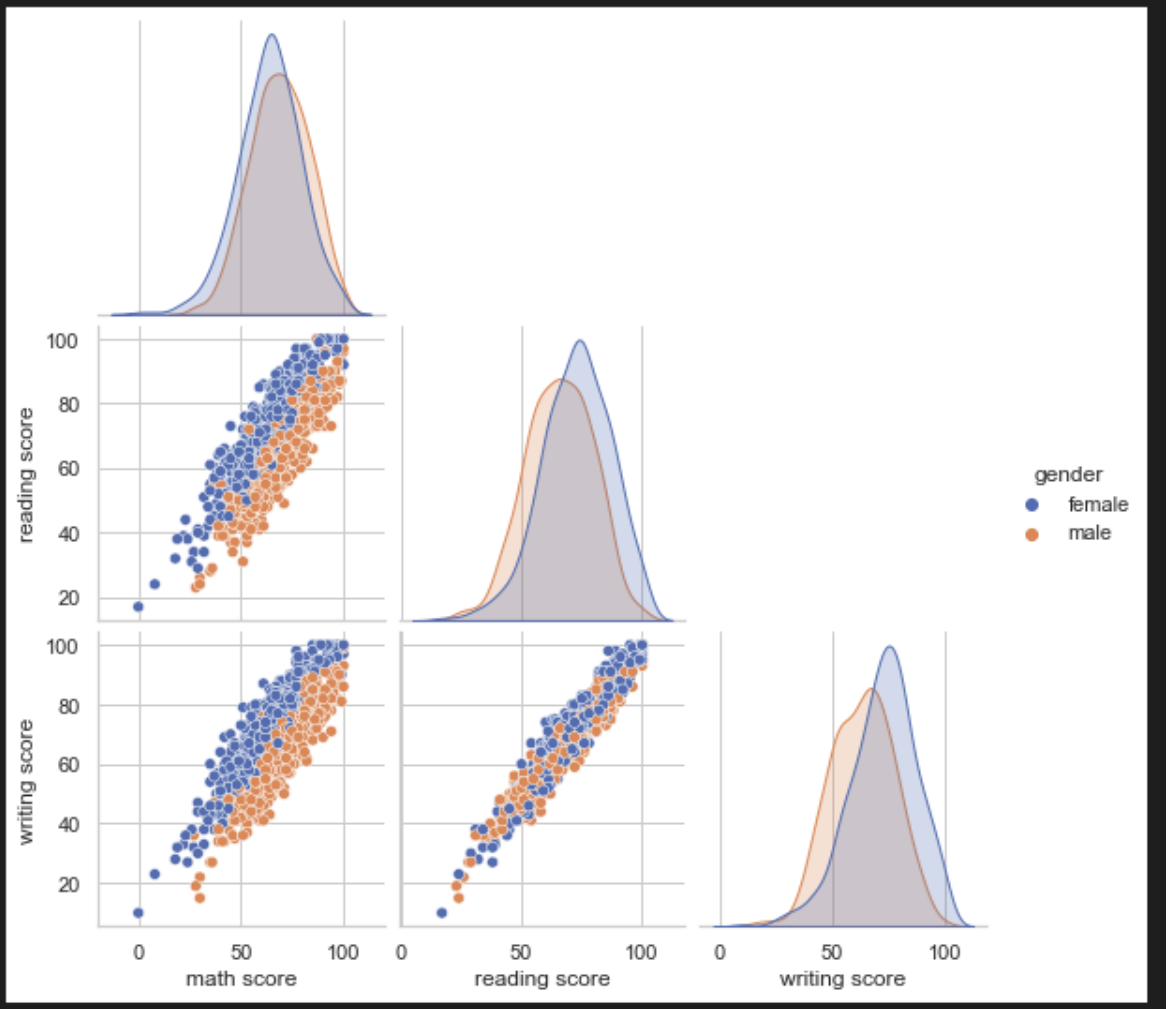
✓ 0.6s

```
<seaborn.axisgrid.JointGrid at 0x27b7502ff40>
```



- pairplot,
  corner=True : Simertik eksendeki aynı grafikleri kaldırır

```
1  sns.pairplot(data=df, hue="gender", corner=True)
✓ 2.8s
<seaborn.axisgrid.PairGrid at 0x27b75bc4fd0>
```



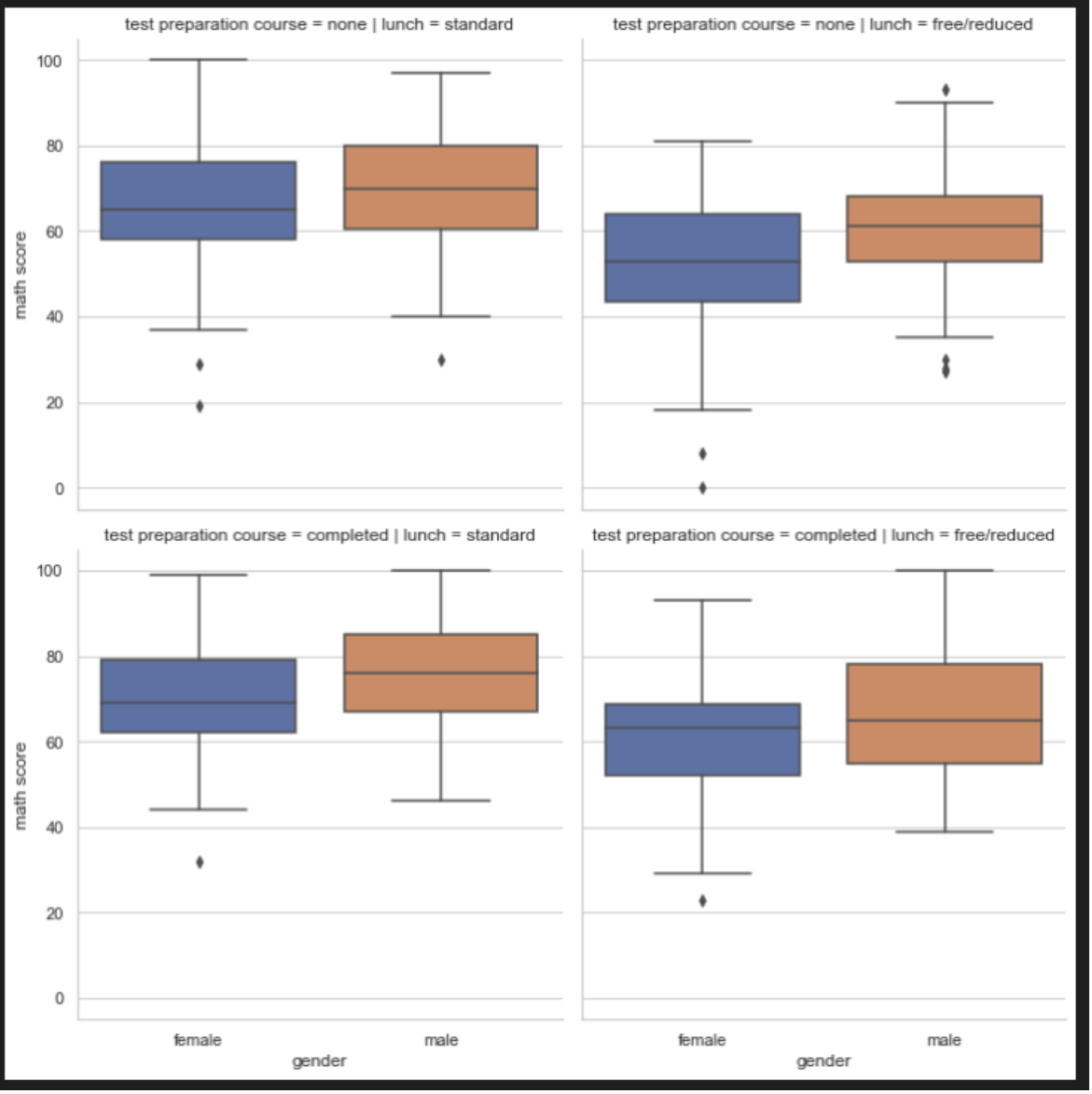
▼ Grids

- Grid Grafik
  satır ve sütubnlara verilen değerlere göre farklı grafikler döker.
  Grafik türü değiştirilebilir.

```
1 sns.catplot(data=df, x="gender",y="math score",
2 kind="box", col="lunch", row="test preparation course")
```

✓ 1.4s Python

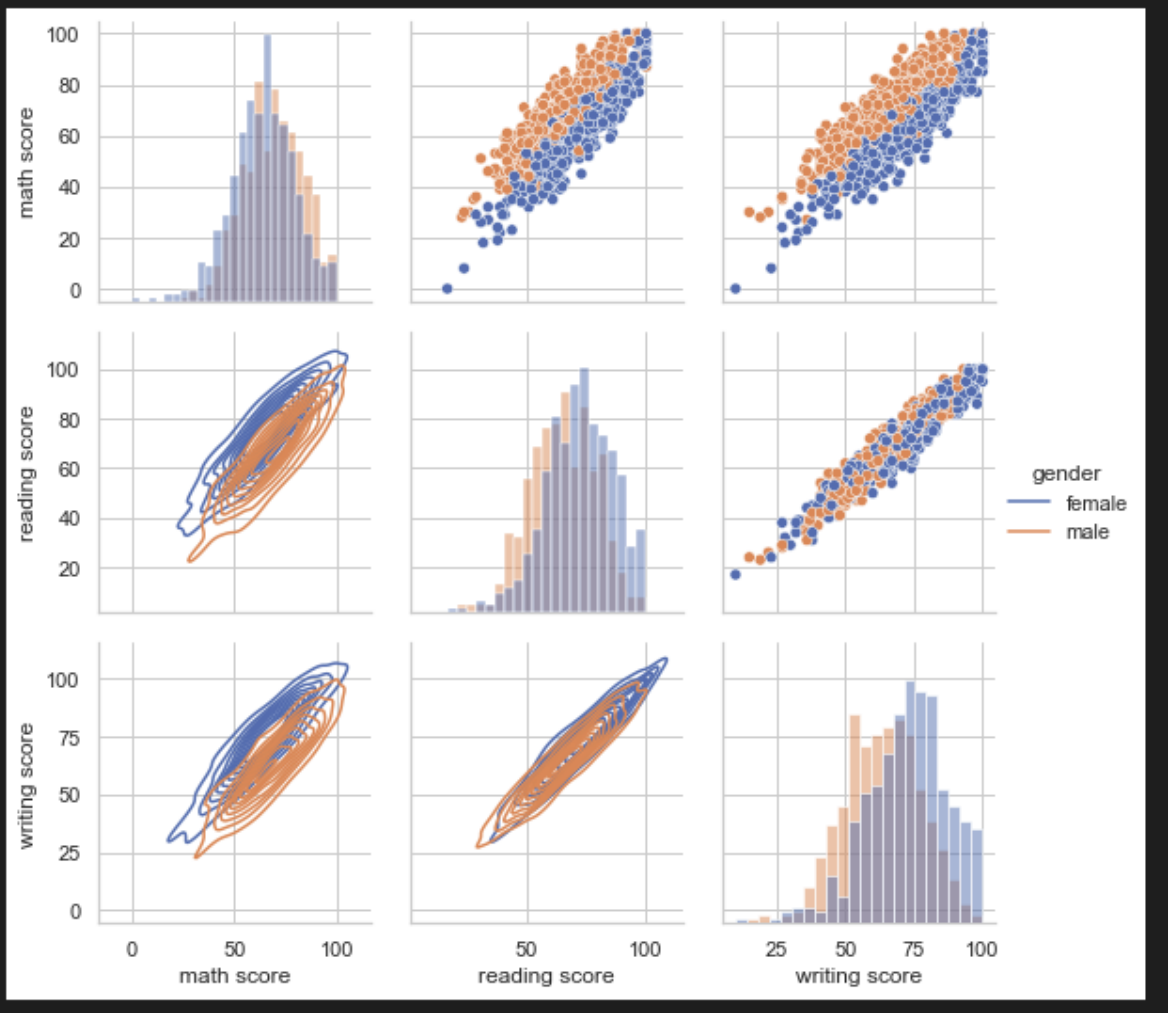
```
<seaborn.axisgrid.FacetGrid at 0x27b785f7e80>
```



- PairGrid
  grid = grid.map_upper(sns.scatterplot) : Üst çapraz değişir
  grid = grid.map_lower(sns.kdeplot) : Alt Çağraz değişir
  grid = grid.map_diag(sns.histplot) : Diagonal değişir

```
1  grid = sns.PairGrid(df, hue="gender")
2  grid = grid.map_upper(sns.scatterplot)
3  grid = grid.map_lower(sns.kdeplot)
4  grid = grid.map_diag(sns.histplot)
5  grid = grid.add_legend()
```

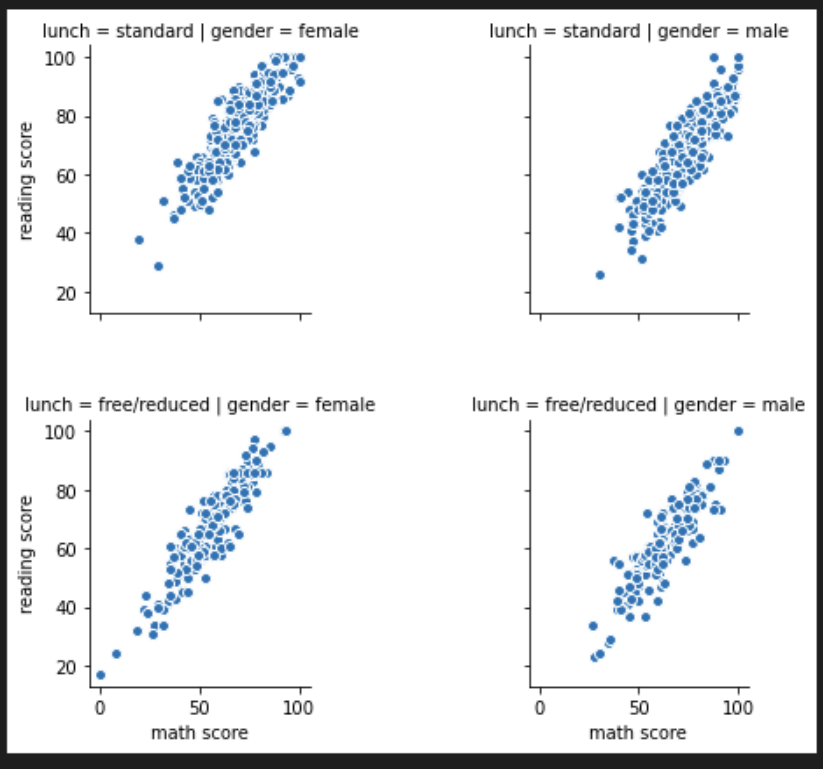
✓ 6.3s



- FacetGrid

```python
g = sns.FacetGrid(data=df,col='gender',row='lunch')
g = g.map(plt.scatter, "math score", "reading score", edgecolor="w")
g.add_legend()

plt.subplots_adjust(hspace=0.4, wspace=1)
```



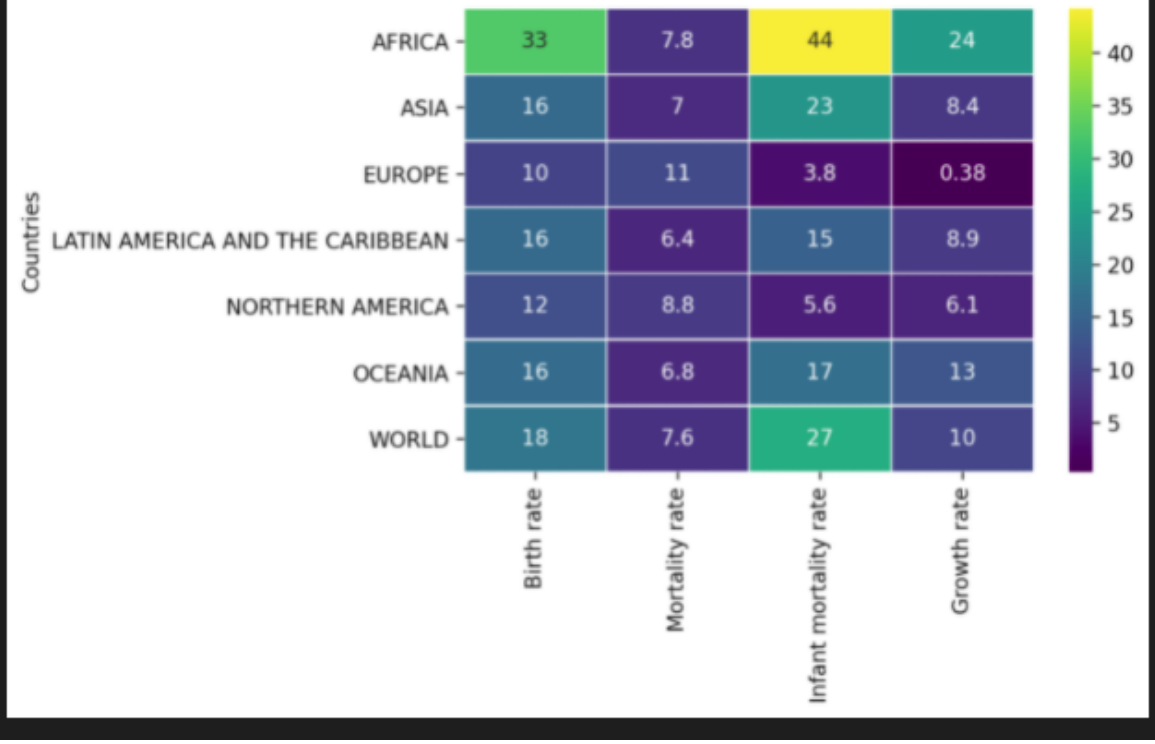
▼ Matrix Plot

- Heat map
  lw= 0.5 : kutular arası boşluklar (linewidth= şeklinde de yazılabilir)
  annot=True : Değerleri kutulara ekler
  cmap="viridis" : color map

```
1  plt.figure(dpi=200)
2  sns.heatmap(df.drop("Life expectancy", axis=1),
3   lw= 0.5, annot=True, cmap="viridis")
```

✓ 0.8s Python

```
<AxesSubplot:ylabel='Countries'>
```



- Cluster map : bağlantılı kolonları seçer
  col_cluster=False : kolonlar arasındaki bağlantıyı kapatır

```python
plt.figure(dpi=200)
sns.clustermap(df.drop("Life expectancy", axis=1),
lw= 0.5, annot=True, cmap="viridis", col_cluster=False)
```

✓ 0.8s Python

```
<seaborn.matrix.ClusterGrid at 0x2a54847f400>

<Figure size 1200x800 with 0 Axes>
```