# Demografik Veri Analizi

Bu uygulamada Pandas kütüphanesini kullanarak veri analizi yapacağız. Aşağıda 1994 Census veri tabanından çıkarılan bir demografik veri seti verilmiştir.

| index | age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | salary |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 39 | State-gov | 77516 | Bachelors | 13 | Never-married | Adm-clerical | Not-in-family | White | Male | 2174 | 0 | 40 | United-States | <=50K |
| 1 | 50 | Self-emp-not-inc | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 13 | United-States | <=50K |
| 2 | 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-family | White | Male | 0 | 0 | 40 | United-States | <=50K |
| 3 | 53 | Private | 234721 | 11th | 7 | Married-civ-spouse | Handlers-cleaners | Husband | Black | Male | 0 | 0 | 40 | United-States | <=50K |
| 4 | 28 | Private | 338409 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Wife | Black | Female | 0 | 0 | 40 | Cuba | <=50K |
| 5 | 37 | Private | 284582 | Masters | 14 | Married-civ-spouse | Exec-managerial | Wife | White | Female | 0 | 0 | 40 | United-States | <=50K |
| 6 | 49 | Private | 160187 | 9th | 5 | Married-spouse-absent | Other-service | Not-in-family | Black | Female | 0 | 0 | 16 | Jamaica | <=50K |
| 7 | 52 | Self-emp-not-inc | 209642 | HS-grad | 9 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 45 | United-States | >50K |
| 8 | 31 | Private | 45781 | Masters | 14 | Never-married | Prof-specialty | Not-in-family | White | Female | 14084 | 0 | 50 | United-States | >50K |
| 9 | 42 | Private | 159449 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 5178 | 0 | 40 | United-States | >50K |

Show 10 per page

1 2 10 100 1000 1900 1990 2000

Aşağıdaki soruları Pandas kütüphanesini kullanarak yanıtlayınız.

- Bu veri setinde hangi ırktan kaç kişi vardır? Her ırkın sayıyı gözükmelidir.
- Erkeklerin yaş ortalaması nedir?
- Lisans derecesine sahip kişilerin yüzdesi nedir?
- Lisans, Yüksek Lisans veya Doktora yapanlar arasından yüzde kaçı 50.000'den fazla kazanıyor?
- Lisans, Yüksek Lisans veya Doktora yapmayanlar arasından yüzde kaçı 50.000'den fazla kazanıyor?
- Bir kişinin haftada çalıştığı minimum saat sayısı nedir?
- Haftada asgari saat çalışan insanların yüzde kaçının maaşı 50 binden fazladır?
- 50 binden fazla kazanan en yüksek yüzdeye sahip ülke hangisidir ve bu yüzde nedir?
- Hindistan'da 50 binden fazla kazananlar için en popüler meslek nedir?

## Çözüm:

İlk olarak Pandas kütüphanemizi import ediyoruz.

```
import pandas as pd
```

Mevcut veri setini okuyarak data adlı değişkene atayalım. Böylece hesaplamalırımızı data değişkeninde kolayca yapabileceğiz.

```
data = pd.read_csv(".../adult.data.csv") #... yerine veri setinin dosya konumunu yazmalısınız.
```

Veri setindeki ırk sayısını hesaplamak için *value_counts()* fonksiyonunu kullanacağız. Bu fonksiyon kategorik verileri saymamızı sağlar.

```
race_count = data['race'].value_counts()
output:
White                27816
Black                 3124
Asian-Pac-Islander    1039
Amer-Indian-Eskimo     311
Other                  271
Name: race, dtype: int64
```

*mean()* fonksiyonu istenen sutünların ortalamasını döndürür. Bizden erkeklerin yaş ortalaması isteniyor. Cinsiyet kontrolü yapmak için loc fonksiyonundan yararlanacağız. Loc satırlardaki ve sütunlardaki belirli değerlere erişmemizi sağlar. Nasıl mı? Örneğe bakalım :)

```
men = data.loc[ data['sex']=="Male"]
```

output:

| | age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | salary |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 39 | State-gov | 77516 | Bachelors | 13 | Never-married | Adm-clerical | Not-in-family | White | Male | 2174 | 0 | 40 | United-States | <=50K |
| 1 | 50 | Self-emp-not-inc | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 13 | United-States | <=50K |
| 2 | 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-family | White | Male | 0 | 0 | 40 | United-States | <=50K |
| 3 | 53 | Private | 234721 | 11th | 7 | Married-civ-spouse | Handlers-cleaners | Husband | Black | Male | 0 | 0 | 40 | United-States | <=50K |

*men* değişkeninde sadece cinsiyeti erkek olanları verileri mevcuttur. Her satırı kontrol ederek *sex* sütununda *Male*'e sahip olanları değişkene ekliyor. Erkeklerin sadece yaş verisini görmek için:

```
age_men = data.loc[ data['sex']=="Male",'age']
output:
0         39
1         50
2         38
3         53
7         52
          ..
32553     32
```

Artık mean() fonksiyonunu çağırarak erkeklerin yaş ortalamasını hesaplayabiliriz.

```
average_age_men = data.loc[ data['sex']=="Male",'age'].mean()
output:
39.43354749885268
```

***Ekstra** Hem kadınların hem erkeklerin yaş ortalaması hesaplamak istiyorsak cinsiyete göre gruplama yaptığımızda yaş ortalamalarını hesaplayabiliriz.

```
average_age = data[["sex", "age"]].groupby("sex").mean()
output:
```

| sex | age |
|---|---|
| **Female** | 36.858230 |
| **Male** | 39.433547 |

Yukarıda ırkların sayısını hesapladığımız gibi eğitim derecelerinin sayılarını da hesaplayalım.

```
percentage_bachelors_count = data['education'].value_counts()
output:
HS-grad         10501
Some-college     7291
Bachelors        5355
Masters          1723
Assoc-voc        1382
11th             1175
Assoc-acdm       1067
10th              933
```

```
7th-8th           646
Prof-school       576
9th               514
12th              433
Doctorate         413
5th-6th           333
1st-4th           168
Preschool          51
Name: education, dtype: int64
```

Hangi eğitim düzeyinde kaç kişinin olduğunu görebiliyoruz. Lisans derecesine sahip kişilerin oranlarını hesaplamak için tüm sayısı lisans öğrenci sayısına böleceğiz. Sadece sayı bize gerektiği için *len* fonksiyonu ile elde edebiliriz. Yüzdesel olarak görmek için bölünen oranı 100 ile çarpıyoruz.

```
bachelors_count = len( data.loc[ data["education"] == "Bachelors"].value_counts() )
education_count = len(data['education'])
percentage_bachelors = (bachelors_count / education_count) * 100
print(percentage_bachelors)
output:
16.43991277909155
```

loc fonksiyonu ile sadece belirli değerler erişebilmiştik. Örneğin *"Male"* veya *"Bachelors"* gibi. Peki hem lisans hem de yüksek lisans öğrencilerine erişmek istiyorsak ne yapmalıyız? Yine aynı yöntem :) Bulmak istediğimiz değerleri bir liste olarak tutuyoruz.

```
array_higher_edu = ["Bachelors", "Masters", "Doctorate" ]
```

Şimdi loc fonksiyonu tüm satırlara bakacak. Diyeceğiz ki *isin* fonksiyonunu kullanarak sana liste olarak verdiğimiz değerlerden hangisi mevcut kontrol et. isin DataFrame'deki her öğenin değerleri içerilip içerilmediğine bakar.

```
array_higher_edu = ["Bachelors", "Masters", "Doctorate" ]
higher_education = data.loc[data["education"].isin(array_higher_edu)]
output:
```

| | age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | salary |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 39 | State-gov | 77516 | Bachelors | 13 | Never-married | Adm-clerical | Not-in-family | White | Male | 2174 | 0 | 40 | United-States | <=50K |
| 1 | 50 | Self-emp-not-inc | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 13 | United-States | <=50K |
| 4 | 28 | Private | 338409 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Wife | Black | Female | 0 | 0 | 40 | Cuba | <=50K |
| 5 | 37 | Private | 284582 | Masters | 14 | Married-civ-spouse | Exec-managerial | Wife | White | Female | 0 | 0 | 40 | United-States | <=50K |
| 8 | 31 | Private | 45781 | Masters | 14 | Never-married | Prof-specialty | Not-in-family | White | Female | 14084 | 0 | 50 | United-States | >50K |

Eğer DataFrame'deki her öğenin yukarıdaki listedeki değerleri içermeyenleri görmek istiyorsak ufak bir *not in* (~) işareti ekleyeceğiz.

```
lower_education = data.loc[~data["education"].isin(array_higher_edu)]
lower_education
output:
```

| | age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | salary |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-family | White | Male | 0 | 0 | 40 | United-States | <=50K |
| 3 | 53 | Private | 234721 | 11th | 7 | Married-civ-spouse | Handlers-cleaners | Husband | Black | Male | 0 | 0 | 40 | United-States | <=50K |
| 6 | 49 | Private | 160187 | 9th | 5 | Married-spouse-absent | Other-service | Not-in-family | Black | Female | 0 | 0 | 16 | Jamaica | <=50K |
| 7 | 52 | Self-emp-not-inc | 209642 | HS-grad | 9 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 45 | United-States | >50K |

Lisans, Yüksek Lisans veya Doktora yapanları ve yapmayanları iki ayrı değişkene atadık. İki grup arasında 50 binden fazla kazananların yüzdesel oranını hesaplayacağız.

Lisans, Yüksek Lisans veya Doktora yapanlar:

```
higher_education_salary50k_count = len(
   higher_education[higher_education['salary'] == '>50K']
   )
higher_education_rich = (
   higher_education_salary50k_count/len(higher_education)
   ) *100
print(higher_education_rich)
output:
46.535843011613935
```

Lisans, Yüksek Lisans veya Doktora yapmayanlar:

```
lower_education_salary50k_count = len(
   lower_education[lower_education['salary'] == '>50K']
   )
lower_education_rich = (
   lower_education_salary50k_count/len(lower_education)
   ) *100
print(lower_education_rich)
output:
17.3713601914639
```

Dataframe'de herhangi bir verinin minimum değerini bulmak için min fonksiyonunu kullanırız. Bir kişinin haftada çalıştığı minimum saat sayısı:

```
min_work_hours = data['hours-per-week'].min()
print(min_work_hours)
output:
1
```

Haftada asgari saat çalışanların ve 50 binden fazla alanları bulmak için yine loc fonksiyonundan yaralanacağız. Haftalık asgari çalışma saatini 45 olarak alabiliriz. İki durumunda True olması için ve operatörünü kullanıyoruz.

```
num_min_workers = data.loc[
                          (data['salary'] == ">50K") &
                          (data["hours-per-week"]<45)
                          ]
num_min_workers
output:
```

| | age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | salary |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **9** | 42 | Private | 159449 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 5178 | 0 | 40 | United-States | >50K |
| **11** | 30 | State-gov | 141297 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Husband | Asian-Pac-Islander | Male | 0 | 0 | 40 | India | >50K |
| **14** | 40 | Private | 121772 | Assoc-voc | 11 | Married-civ-spouse | Craft-repair | Husband | Asian-Pac-Islander | Male | 0 | 0 | 40 | ? | >50K |
| **25** | 56 | Local-gov | 216851 | Bachelors | 13 | Married-civ-spouse | Tech-support | Husband | White | Male | 0 | 0 | 40 | United-States | >50K |
| **38** | 31 | Private | 84154 | Some-college | 10 | Married-civ-spouse | Sales | Husband | White | Male | 0 | 0 | 38 | ? | >50K |

Bizden bu çalışanların yüzdesi isteniyor.

```
rich_percentage = (
   len(num_min_workers) / len(data['hours-per-week'])
   ) *100
print(rich_percentage)
```

Ülkelere göre 50 binden fazla kazanan çalışanların sayısını hesaplamak için value_counts fonksiyonundan yararlanacağız. Hep yaptığımız gibi :) Kategorik verileri sayısını bize döndürüyor.

```
highest_salary_country_count = data.loc[
                                    data['salary'] == '>50K',
                                    'native-country'].value_counts()
output:
United-States          7171
?                       146
Philippines              61
.
.
.
Nicaragua                 2
Honduras                  1
Name: native-country, dtype: int64
```

Ülkelerin sahip olduğu çalışanları yüzdesel olarak hesaplamamız lazım. Toplam ülke sayısını 50 binden fazla alan çalışan sayısına sahip ülkelere böleceğiz. Bizden sadece en yüksek yüzdeye sahip ülke isteniyor. İstenen sütun üzerindeki maximum değerini *idxmax* fonksiyonu döndürür.

```
highest_earning_country = (
   (highest_salary_country_count / country_count) * 100).idxmax()
print(highest_earning_country)
output:
Iran
```

Peki İran'ın yüzdesi nedir?

```
highest_earning_country_percentage = ((highest_salary_country_count /
country_count) * 100).max()
print(highest_earning_country_percentage)
output:
41.86046511627907
```

Son sorumuzda Hindistan'da 50 binden fazla kazandıran en popüler meslek nedir? İlk aşamada Hindistan'da olup 50 binden fazla kazananları bulalım.

```
highest_salary_IN = data[
                        (data['salary'] == '>50K') &
                        (data['native-country'] == 'India')
                        ]
```

output:

| | age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | salary |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **11** | 30 | State-gov | 141297 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Husband | Asian-Pac-Islander | Male | 0 | 0 | 40 | India | >50K |
| **968** | 48 | Private | 164966 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | Asian-Pac-Islander | Male | 0 | 0 | 40 | India | >50K |
| **1327** | 52 | Private | 168381 | HS-grad | 9 | Widowed | Other-service | Unmarried | Asian-Pac-Islander | Female | 0 | 0 | 40 | India | >50K |

Şimdi buradaki mesleklerin sayısını bulalım. En yüksek sayıya sahip meslek en popüler meslek olacak

```
top_IN_occupation = highest_salary_IN['occupation'].value_counts()
output:
Prof-specialty      25
Exec-managerial      8
Tech-support         2
Other-service        2
Sales                1
Transport-moving     1
Adm-clerical         1
Name: occupation, dtype: int64
```

Geriye sadece maximum değere erişmek kaldı. Bunun çözümü tabi ki de idxmax :)

```
top_IN_occupation = highest_salary_IN['occupation'].value_counts().idxmax()
print(top_IN_occupation)
Prof-specialty
```

Öğrenme azminiz için çok teşekkür ederim. Bol pratikli günler :)

Github: @burakakbulut