**Veri Analizi Aşamaları**

1. **Veri erişimi** 
   1. **PROC IMPORT ile CSV/XLSX İçe Aktarma**

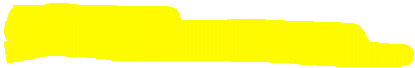
**Libname:** kütüphane tanımlar

* libname pg1 "/home/uXXXXXX/data";

**proc import**: Excel gibi dış kaynaklardan veri okuma.

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

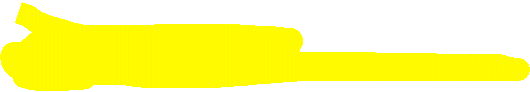
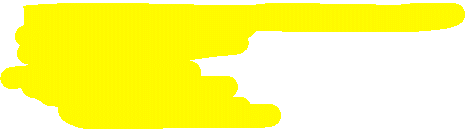


* **datafile**: Kaynak dosyanın tam yolu.
* **out**: Hedef SAS veri seti.
* **dbms**: Dosya formatı.
* **replace**: Yazma izni.
* **guessingrows**: Karakter/numerik ayrımı için örnek satır sayısı.

**Ne Zaman:** Yapılandırılmış (tablo) veriler CSV veya Excel formatındaysa

* 1. metin, ekran görüntüsü, yazı tipi içeren bir resim

     Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**DATA ve INFILE/INPUT ile Düz Dosya Okuma**



* **:format**\*\*.\*\*: Uzunluk belirterek okuyucu (informat).
* **anydtdte.**: Farklı tarih formatlarını otomatik algılar.

**Ne Zaman:** Düz metin dosyaları (txt, csv) içinde özel ayraçlar veya karmaşık yapılar varsa.

**proc contents**: Veri seti yapısını görüntüleme.

1. **Veri Keşfi**

**2.1-İçerik ve Yapı İnceleme**

**proc contents data=output.trafficRaw varnum;**

* Değişkenleri oluşturulma sırasına göre listeler

**run;**

**2.2- İlk Gözlemler**

**ekran görüntüsü, metin, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**



**2.3-Kategorial Dağılım**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**



**2.4-Sayısal Özet**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**



**Ne Zaman:** Veri setindeki sayısal sütunların merkezi eğilimi ve dağılım bilgisini hızlıca görmek istendiğinde.

**proc print**: Veri görüntüleme.

**proc means**: Sayısal değişkenlerin özet istatistikleri.

**proc freq**: Kategorik değişkenlerin frekansı.

* **One-way**: Tek değişken
* **Two-way**: Çapraz tablo

**proc univariate**: Ayrıntılı dağılım analizi.

**where** koşulu: Koşullu filtreleme.

**substr, like**: Karakter verisi filtreleme.

**macro değişkeni**:

%let T=TS; where Type="&T";

1. **Veri ön hazırlık (temizlik)**

**data adımı**: Yeni veri seti oluşturma, dönüşüm, koşullu işlemler.

**3.1-Eksik Değer Filtreleme**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**



**3.2-Değişken Seçme/Düşürme**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**



**3.3-String/Fonksiyon Uygulamaları**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**



**3.4-Sıralama ve Yinelenen Kayıtlar**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**



* **nodupkey**: Yinelenen satırları atla.
* **dupout**: Atılanları ayrı bir tabloya kaydet.

1. **Analiz ve Raporlama**

**4.1Aylık Trafik Özet Tablosu**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.



**4.2-SQL ile Karmaşık Sorgular**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**



**4.3- ODS ile PDF Raporu Oluşturma**

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**5- Dışa Aktarım**

**5.1- PROC EXPORT ile CSV Yazma**

**metin, yazı tipi, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

**5.2-ODS EXCEL ile Excel Dosyası**

**metin, yazı tipi, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulan içerik yanlış olabilir.**

* **DERS kod örneklerli ve kullanımları**

data myclass;

Length Name $9 Grade $4;**/\* bu satır olmazsa string ifadeler outputta gözükmüyor\*/**

infile "/home/u64155505/EPG1V2/data/class\_birthdate.csv"

dlm=',' firstobs=2**;/\* ilkini göstermeden 2.den başlıyor\*/**

input Name Grade Age Height; **/\* Stringlerden sonra $ kullanılır.\*/**

run;

**/\*temizleme işlemi\*/**

data storm\_2019;

infile"/home/u64155505/EPG1V2/output/storm\_final\_output.csv"

dlm=',' missover **/\*hatalı gelen değer için \*/** firstobs=2;

input Season Name $ Basin:$2. BasinName $ OceanCode $ Ocean $ StormType

$ StartDate : anydtde. EndDate:anydtde. MinPressure:coma. MaxWind ;

if Name= "26F" then

Name=" ";

format StartDate EndDate date9.;

run;

 infile: Dışarıdan bir dosya (CSV) okuma işlemini başlatır.

 "/home/u64155505/EPG1V2/output/storm\_final\_output.csv": Okunacak dosyanın tam yolu.

 dlm=',': Dosyada değerler virgül (',') ile ayrılmış. Yani bir CSV dosyası.

 missover: Eğer satırdaki bazı değişken değerleri eksikse, SAS bir sonraki satıra geçmeden bunları boş olarak alır. Bu veri kaymasını engeller.

 firstobs=2: Dosyanın ilk satırı başlık olduğundan dolayı okumayı ikinci satırdan başlatır.

**/\*Veri erişimi adımı\*/**

**/\*Verilerin saklanacağı kütüphane\*/**

libname parks "/home/u64155505/EPG1V2/output";

**/\*Değişken adlarının SAS veri setine uygun hale gelmesi\*/**

options validvarname=v7**; /\* Geçersiz karakter içeren değişken adlarını SAS uyumlu hale getirir \*/**

options nodate nonumber**; /\* Raporlarda tarih ve sayfa numarasını göstermeyi kapatır \*/**

**/\*CSV Dosyasının SAS veri setine çevir\*/**

proc import datafile="/home/u64155505/EPG1V2/data/np\_traffic.csv"

dbms=csv

out=parks.Trafficraw

replace;

guessingrows=100;

run;

 proc import: CSV dosyasını okur.

 dbms=csv: CSV formatında olduğunu belirtir.

 out=parks.trafficraw: Çıktı veri seti olarak parks kütüphanesine Trafficraw adıyla kaydeder.

 replace: Aynı isimde veri varsa üzerine yazar.

 guessingrows=100: Değişken tipini belirlerken 100 satır göz önünde bulundurulsun.

**/\*Veri keşfi\*/**

**/\*İLK 10 SATIR VERİYİ GÖRÜNTÜLEME \*/**

title "10 satır veri";

proc print data=parks.TRAFFICRAW (obs=10);

run;

**/\*Region, parktype ve UnitCode sütunlarının dağılımını incele\*/**

proc freq data=parks.trafficraw;

tables Region parktype UnitCode /nocum nopercent;

run;

 proc freq: Frekans tablosu oluşturur.

 /nocum nopercent: Kümülatif sayı ve yüzdelik gösterimini kapat

**/\*Veri ön işleme \*/**

**/\* TrafficCount değeri: boş olmayan veriler\*/**

data parks.trafficValid;

set parks.trafficraw;

**if not missing**(TrafficCount);

run;

**/\*Sıralama işlemi\*/**

proc sort data=parks.trafficValid nodupkey out=parks.trafficNoDups;

by \_all\_;

run;

 nodupkey: Tüm değişkenler aynıysa yinelenen kayıtları siler.

 by \_all\_;: Tüm sütunları karşılaştırmak için kullanılır.

**/\*Analiz etme\*/**

**/\*Her kayıt için ay bilgisini almak\*/**

data parks.trafficMonth;

set parks.TRAFFICNODUPS;

Month = month(ReportingDate);

run;

**/\*Aylara göre toplam trafik sayısını hesapla\*/**

proc means data=parks.trafficMonth;

class Month;

var TrafficCount;

output out=parks.MonthlySummary sum=TotalTraffic;

run;

**/\* Hesaplanan toplam trafik sayısı değerini ekrana yazalım\*/**

title "Aylara göre toplam trafic sayısı";

proc print data=parks.monthlysummary;

var Month TotalTraffic;

run;

**/\*Park türlerine göre toplam tarfik sayısı hesaplama\*/**

proc means data=parks.trafficnodups;

class ParkType;

var TrafficCount;

output out=parks.ParkTypeSummary sum=TotalTraffic;

run;

**/\*Park türlerine göre trafik verini yazdır\*/**

title "Park türlerine göre Trafik";

proc print data=parks.parktypesummary noobs;

var ParkType TotalTraffic;

run;

**/\*Veriyi dışarı aktarım\*/**

**/\*PDF çıktısı olarak alınsın, stil"journal" olsun\*/**

ods pdf file="/home/u64155505/EPG1V2/output/ParkTrafficReport.pdf"

style=journal;

**/\*Park türlerine göre trafik verisini rapora ekle\*/**

title "Park turune gore trafik sayisi";

proc print data=parks.parktypesummary noobs;

run;

**/\*Aylık trafik özeti rapora ekle\*/**

proc print data=parks.monthlysummary noobs;

run;

**/\*Pdf dosyasını tamamla ve kapat\*/**

ods pdf close;

**ARTILAR**

**250 beygir gücünden yüksek araçları filtreleme**

data horsepower\_over250;

set sashelp.cars;

where horsePower>250;

run;

 sashelp.cars: SAS’in yerleşik otomobil veri setidir.

 data horsepower\_over250;: Yeni bir veri seti oluşturuyoruz.

 where horsepower > 250;: Sadece 250'den büyük beygir gücüne sahip araçlar seçiliyor.

 set: Var olan veri setinden veri alını

**Bu araçların hangi tiplerde yoğunlaştığını analiz etme**

proc freq data=horsepower\_over250;

tables Type;

run;

 proc freq: Kategorik değişkenlerin frekans analizini yapar.

 tables Type;: Araç tiplerine göre kaç adet araç olduğunu gösterir (örneğin: SUV, Sedan vs.).

**MPG\_city değişkeninin orijine göre ortalaması**

proc means data=horsepower\_over250;

var MPG\_city;

class origin;

run;

 proc means: Sayısal değişkenlerin ortalama, std sapma gibi istatistiklerini verir.

 class origin;: Orijin (menşe) bazında gruplayarak MPG\_city ortalamalarını verir.

**MPG\_city ve origin ilişkisinin frekansı**

proc freq data=horsepower\_over250;

tables MPG\_city \* Origin;

run;

 MPG\_city değerleri ile Origin arasında bir çapraz tablo oluşturur.

 Sayısal değişken olan MPG\_city, çok fazla kategoriye bölüneceği için genelde sınıflandırılarak kullanılır (bu haliyle yorumlaması zor olabilir).

**MPG\_Highway ve origin frekans ilişkisi**

proc freq data=horsepower\_over250;

tables MPG\_Highway \* Origin;

run;

* Aynı şekilde şehir dışı yakıt tüketimi (MPG\_Highway) ile Origin arasındaki frekans dağılımı gösterilir.

**Korelasyon analizi**

proc corr data=horsepower\_over250;

var Invoice MPG\_city Horsepower;

run;

 proc corr: Sayısal değişkenler arasındaki ilişkiyi gösterir.

 Invoice: Araç fiyatı, MPG\_city: şehir içi yakıt verimliliği, Horsepower: beygir gücü.

 Bu analiz ile fiyat, yakıt verimliliği ve beygir gücü arasındaki korelasyonlar hesaplanır.