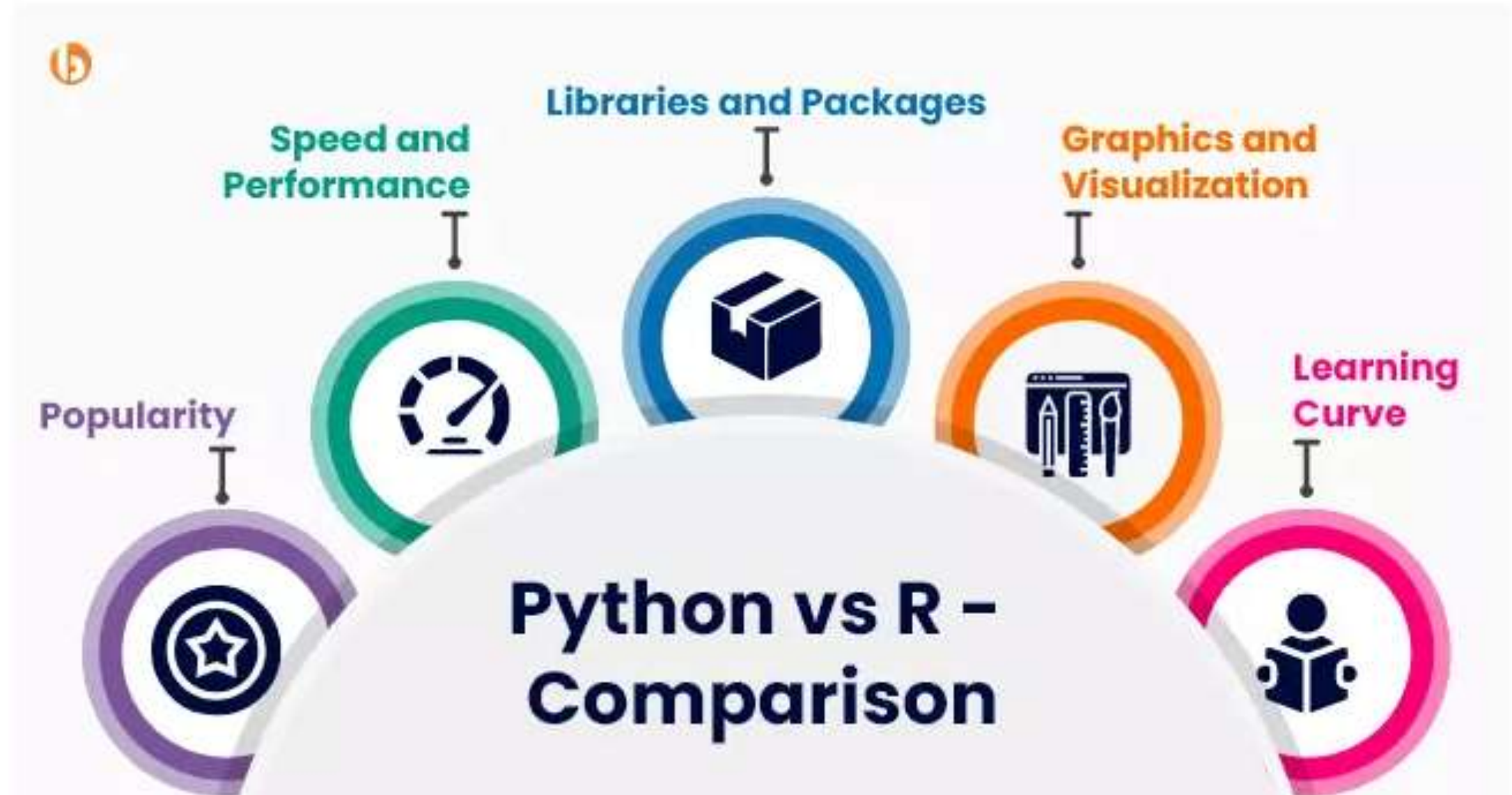


Statistik metodlar (R, Python). Python programı vasitəsi ilə statistika çıxarılması



Populyar Statistik Təhlil Proqramları

Müxtəlif sənaye sahələrində və tədqiqat sahələrində bir çox statistik proqram paketlərindən geniş şəkildə istifadə olunur. Ən populyar seçimlərdən bəziləri bunlardır:

R: R statistik hesablamalar və qrafiklər üçün pulsuz, açıq mənbəli bir proqramlaşdırma dili və proqram mühitidir. Məlumat emalı, vizuallaşdırma və təhlil üçün geniş bir paket ekosistemi təqdim edir və bu da onu statistiklər arasında populyar seçmə çevirir.

Python: Python, məlumat təhlili və elmi hesablamalar üçün NumPy, SciPy və pandas kimi güclü kitabxanalara malik çox yönlü bir proqramlaşdırma dilidir. Python-un sadəliyi və çevikliyi, xüsusilə proqramlaşdırma təcrübəsi olan istifadəçilər üçün statistik təhlil üçün cəlbədicə bir seçim halına gətirir.

SPSS: SPSS (Sosial Elmlər üçün Statistik Paket), sosial elmlər tədqiqatlarında, marketinqdə və səhiyyə xidmətlərində geniş istifadə olunan hərtərəfli statistik proqram paketidir. İstifadəçi dostu interfeysə və məlumatların təhlili və hesabatı üçün geniş statistik prosedurlara malikdir.

SAS: SAS (Statistik Analiz Sistemi), məlumatların idarə olunması, inkişaf etmiş analizlər və proqnozlaşdırıcı modelləşdirmə üçün istifadə olunan güclü statistik proqram paketidir. SAS səhiyyə, maliyyə və dövlət kimi sahələrdə məlumat əsaslı qərarvermə prosesində geniş şəkildə istifadə olunur.

Stata: Stata, məlumatların təhlili, emalı və vizuallaşdırılması üçün alətlər təqdim edən statistik proqram paketidir. Güclü statistik imkanları və istifadəsinin asanlıığı səbəbilə akademik tədqiqatlar, iqtisadiyyat və sosial elmlər sahəsində populyardır.

MATLAB: MATLAB, sayısal hesablamalar və vizuallaşdırmalar üçün yüksək səviyyəli proqramlaşdırma dili və mühitidir. Statistik təhlil, maşın öyrənməsi və signal emalı üçün daxili funksiyalar və alət qutuları təklif edir.

Python vasitəsilə Statistik Metodlar

1. Statistikanın əsas anlayışları

•Descriptive Statistics (Təsviri Statistika):

- Orta (Mean)
- Median
- Mod (Mode)
- Standart Sapma (Standard Deviation)
- Varsians
- Min, Max, Kvartillər

•Inferential Statistics (Çıxarıcı Statistika):

- Hipotez testləri
- p-qiyəti
- t-testi, chi-square testi
- Regresiya analizi

2. Python-da Statistik Kitabxanalar

Python dilində statistik analiz üçün əsas kitabxanalar:

Kitabxana	Təyinatı
NumPy	Əsas riyazi əməliyyatlar
Pandas	Verilərin saxlanması və emalı
SciPy	Statistik və elmi funksiyalar
Statsmodels	Statistik modellər (regressiya, hipotez testləri)
Scikit-learn	Maşın öyrənməsi və statistik modellər
Matplotlib / Seaborn	Vizualizasiya

R Statistik Kitabxanalar (Packages)

Kitabxana

stats (*əsas paketdir*)

car

MASS

lmtest

nortest

psych

Hmisc

e1071

boot

lme4

multcomp

survival

BayesFactor

İstifadə sahəsi

T-test, ANOVA, regresiya, ehtimal paylanmaları və s.

Regresiya analizləri, diagnostik testlər, ANOVA və dəyişənlər arasındakı əlaqə.

Modern statistik üsullar – logistika regresiyası, diskriminant analiz və s.

Linear modellərin testləri (heteroskedastiklik, autocorrelation və s.).

Normallıq testləri (Anderson-Darling, Lilliefors, Shapiro-Francia və s.).

Psixometriya, təsviri statistika, faktor analizi və s.

Statistik məlumatların özetlənməsi, korelyasiya və təsviri analiz.

Statistika və maşın öyrənməsi – xüsusilə skewness, kurtosis və SVM.

Bootstrap analizləri.

Qarışıq effektli modellər (mixed models).

Çoxlu müqayisə testləri (post-hoc analizlər).

Sağ qalma (survival) analizləri.

Bayes statistikasını və Bayes faktorlarının hesablanması.

R Vizuallaşdırmalar

Github-da R-dən istifadə etməklə məlumatların vizuallaşdırılması üçün nəzərdə tutulmuş paketlərdən ibarət 2000-ə yaxın repozitoriya var. Aşağıda hər birinin qısa təsviri ilə birlikdə istifadə edilən ən ümumi paketlərdən dördü verilmişdir:

1. **Ggplot2** - estetik baxımdan xoşagələn qrafiklərin asan yaradılması üçün nəzərdə tutulmuş diaqram və qrafiklərin yaradılması üçün ən çox tövsiyə edilən paket
2. **Plotly** - JavaScript ilə yanaşı interaktiv veb-əsaslı qrafiklər, xətt qrafikləri, səpələnmə qrafikləri və diaqramlar yaratmaq üçün istifadə olunur (plotly.js istifadə edərək)
3. **Şəbəkə** - baza qrafikasının təkmilləşdirilməsi kimi qurulmuş və çoxvariantlı məlumatları qəbul etmək üçün dəyişdirilmiş yüksək səviyyəli məlumat vizualizasiyası, çoxsaylı kiçik qrafiklərin asan yaradılmasına imkan verir.
4. **RGL** - interaktiv 3 ölçülü süjetlər

Python Vizuallaşdırmaları

Python-da verilənlərin vizuallaşdırılması üçün minlərlə kitabxana və onların modifikasiyaları mövcuddur. Aşağıda bu məqsədlə ən çox istifadə olunan dörd Python məlumat vizuallaşdırma kitabxanası təqdim olunur:

- Matplotlib** – səpələnmə qrafikləri, xətt qrafikləri, histoqramlar və s. yaratmaq üçün istifadə olunan aşağı səviyyəli interfeysdir.
- Seaborn** – vizual cəhətdən cəlbedici üslublar və rəng palitraları ilə fərdiləşdirmə imkanı təqdim edən, matplotlib üzərində qurulmuş yüksək səviyyəli interfeysdir.
- Bokeh** – yüksək interaktivlik təmin etmək üçün HTML və JavaScript-dən istifadə edərək, müasir veb brauzerlərdə interaktiv qrafiklər yaratmaq üçün istifadə olunur.
- Plotly** – matplotlib-ə bənzər plan qurmaq üçün istifadə edilir, lakin kənar göstəriciləri və anomaliyaları vurğulamaq, daha geniş qrafik fərdiləşdirməsi və estetik dizayn komponentləri təqdim etmək üçün daxili alətlərə malikdir.



Vizualizasiya üçün hansı proqramlaşdırma dili daha üstündür?

Vizuallaşdırma sahəsində R dili hələ də Python üzərində müəyyən üstünlüyə malikdir. Hər iki dil həm aşağı səviyyəli, həm də yüksək detallı vizualizasiya üçün zəngin paket və kitabxanalar təqdim edir.

Lakin R dili bu sahədə daha yetkin hesab olunur və daha inkişaf etmiş vizualizasiya imkanları ilə seçilir.

Python və R: Maşın Öyrənməsi və Dərin Öyrənmədə Hansı Daha Güclüdür?

Son dövrlərdə maşın öyrənməsi (machine learning) və dərin öyrənmə (deep learning) sahələrinə olan maraq və tələbat sürətlə artmaqdadır. Aşağıda bu texnologiyaların nə olduğunu və Python ilə R proqramlaşdırma dillərinin bu sahələrdə necə istifadə olunduğunu nəzərdən keçirək.

Maşın Öyrənməsi və Dərin Öyrənmə nədir?

Maşın öyrənməsi — əvvəlki məlumatlardan istifadə edərək müəyyən nümunələri müəyyənləşdirməyə və bu nümunələr əsasında gələcək nəticələri proqnozlaşdırmağa çalışan süni intellekt texnologiyasıdır. Məsələn, axtarış sistemləri, bank əməliyyatlarında fırıldaqçılıq aşkarlanması, nitqin mətnə çevrilməsi, GPS əsasında gediş vaxtının təxmini və marşrutun optimallaşdırılması maşın öyrənməsinin gündəlik həyatda geniş tətbiq sahələridir.

Dərin öyrənmə isə insan beyninin məlumatları qavrama və öyrənmə modelinə əsaslanaraq yaradılmış xüsusi bir maşın öyrənmə alt sahəsidir. O, klassik maşın öyrənmədən fərqli olaraq, çox böyük və kompleks verilənlər toplusunu təhlil edə bilir və əvvəlcədən müəyyən edilmiş xüsusiyyətlərə əsaslanmadan, xam (strukturlaşdırılmamış) məlumatlar üzərindən nümunələri özü müəyyənləşdirə bilir.

Dərin öyrənmənin real həyatdakı tətbiqlərinə özünü idarə edən avtomobillər, sosial media alqoritmləri, Siri və digər virtual köməkçilər aiddir.

R ilə Maşın Öyrənmə

R dili maşın öyrənmə sahəsində geniş istifadə olunur və bu məqsədlə xüsusi hazırlanmış bir sıra güclü paketlər mövcuddur. Aşağıda R dilində maşın öyrənmə prosesini həyata keçirmək üçün tövsiyə olunan əsas paketlər təqdim olunur:

- **Caret (Classification And REgression Training)** - Bu paket təsnifat və reqressiya modellərinin qurulmasını sadələşdirmək üçün nəzərdə tutulub. Məlumatların bölünməsi, ilkin emalı, modelin tənzimlənməsi və digər addımlar üçün müxtəlif funksiyalar təqdim edir. Maşın öyrənmə üçün ən geniş istifadə olunan paketlərdən biridir.
- **Dplyr** - Verilənlərin manipulyasiyası üçün istifadə olunur. Həm yaddaşda olan, həm də yaddaşda olmayan məlumat çərçivələri ilə işləməyi asanlaşdırır. Maşın öyrənmədən əvvəl məlumatların təmizlənməsi və hazırlanmasında faydalıdır.
- **Random Forest (Təsadüfi Meşə)** - Bu alqoritm çoxlu sayda qərar ağacından ibarət modeli quraraq, hər bir ağacın nəticəsini toplayaraq yekun qərar verir. Təsnifat və reqressiya problemlərinin həllində yüksək dəqiqlik təqdim edir.
- **E1071** - Bu paket SVM (Dəstək Vektor Maşınları), Naive Bayes təsnifatı, fuzzy (qeyri-səlis) klasterləşdirmə, qısa yol hesablamaları və digər statistik və maşın öyrənmə metodlarını əhatə edir.
- **Rpart (Recursive Partitioning and Regression Trees)** - Bu paket təsnifat və reqressiya ağacları yaratmaq üçün istifadə olunur. İki mərhələli rekursiv bölmə prosesi ilə verilənləri analiz edir və nəticəni proqnozlaşdırmağa kömək

R ilə Maşın Öyrənmənin Müsbət və Mənfi Tərəfləri

Müsbət cəhətləri:

- Zəngin paket seçimi:** R dili maşın öyrənmə üçün xüsusi hazırlanmış çoxsaylı və funksional paketlərə malikdir, bu da müxtəlif alqoritmləri asanlıqla tətbiq etməyə imkan verir.
- Geniş icma dəstəyi:** Aktiv istifadəçi və inkişaf etdirici icması sayəsində qarşılaşdığınız problemlərə tez bir zamanda cavab tapa bilərsiniz.
- Böyük verilənlər dəstləri üçün uyğunluq:** R, böyük miqdarda məlumatı emal etmək və təhlil aparmaq üçün nəzərdə tutulmuşdur.

Mənfi cəhətləri:

- Yüksək yaddaş tələbatı:** Xüsusilə böyük verilənlər üzərində işləyərkən R çoxlu RAM istifadə edə bilər ki, bu da sistem performansını azalda bilər.
- Uyğunsuzluq və öyrənmə ayrısı:** Açıq mənbəli paketlərin müxtəlifliyinə görə alqoritmlərin implementasiyası paketdən pakete dəyişə bilər. Hər yeni paket üçün ayrıca öyrənmə prosesi tələb oluna bilər.

Python ilə Maşın Öyrənmə

Python həm faydalılığı, həm də maşın və dərin öyrənmə üçün geniş kitabxana dəstəyi ilə tanınır. Aşağıda bu sahədə ən çox istifadə olunan bəzi mühüm Python kitabxanaları verilmişdir:

- NumPy** – Böyük çoxölçülü massivlər və matrislər üzərində hesablama aparmaq üçün nəzərdə tutulmuşdur. Eyni zamanda, yüksək səviyyəli riyazi funksiyaların geniş çeşidinə malikdir.
- SciPy** – Xətti cəbr, inteqrasiya, optimallaşdırma və statistik analiz üçün modullar təqdim edir. Xüsusilə təsvir emalında və elmi hesablama proseslərində faydalıdır.
- Scikit-learn** – NumPy və SciPy üzərində qurulmuşdur və Python-da ən məşhur maşın öyrənmə kitabxanalarından biridir. Təsnifat, reqressiya, klasterləşdirmə və digər əsas ML alqoritmləri ilə yanaşı, məlumatların hazırlanması və təhlili üçün də istifadə olunur.
- Theano** – Riyazi ifadələrin tərif, qiymətləndirilməsi və optimallaşdırılması üçün istifadə olunur. Çoxölçülü massivlər üzərində yüksək performanslı hesablamalar aparmağa imkan verir. Səhvlərin aşkarlanması və doğrulama prosesləri üçün də əlverişlidir.
- TensorFlow** – Google Brain tərəfindən dərin öyrənmə və neyron şəbəkələri üçün hazırlanmış açıq mənbə kitabxanadır. Hesablama proseslərini bir neçə CPU və ya GPU arasında bölüşdürmək imkanına malikdir. Bununla belə, bu funksiyalar əsasən icma dəstəyi ilə təmin olunur.

Müsbət cəhətləri:

- Maşın öyrənmə sahəsi getdikcə populyarlaşdığı üçün bir çox alqoritm artıq mövcud kitabxanalarda hazır şəkildə təqdim olunur, əl ilə kod yazmağa ehtiyac qalmır.
- Python-un sadə və oxunaqlı sintaksisi proqramlaşdırma prosesini asanlaşdırır və nəticədə maşın öyrənmə layihələrinin inkişafı daha sürətli baş verir.
- Python həm obyekt yönlü (OOP), həm də prosedur yanaşmanı dəstəklədiyindən, kodların yenidən istifadəsi və real dünya problemlərinin modelləşdirilməsi daha rahat həyata keçirilir.

Mənfi cəhətləri:

- Python-da çox iş parçacığı (multi-threading) ilə bağlı məhdudiyyətlər var, bu da xüsusilə maşın öyrənmə zamanı paralel emalın effektivliyini azaldır.
- Python sətir-sətir icra olunan interpretasiya olunan dil olduğundan, bəzi digər dillərlə müqayisədə daha yavaşıdır.
- Mobil tətbiqlərlə integrasiya imkanları məhduddur və mobil mühitlərdə istifadə üçün geniş dəstək yoxdur.
- Verilənlər bazası ilə integrasiya üçün vasitələr mövcuddur, lakin onlar hələ də tam inkişaf etməmişdir, bu isə Python-u çox böyük məlumat dəstələri ilə işləmək üçün ideal seçim etmir.

Python-da Məlumat Analizi üçün Ən Geniş Yayılmış Statistik Yanaşmalar

Təsviri Statistikalar

Təsviri statistika, məlumatlarınızın əsas xüsusiyyətlərini ümumiləşdirmək və vizual şəkildə təqdim etmək üçün istifadə olunur. Buraya **ortalama, mediana, mod, standart sapma, dəyişmə aralığı, tezlik** və **paylanma** kimi göstəricilər daxildir.

Python-da bu cür analizlər aparmaq üçün **pandas** kitabxanası geniş istifadə olunur. Pandas ilə **məlumat çərçivələri** (dataframe) yaradaraq məlumatlarınızı sətir və sütunlar şəklində təşkil edə bilərsiniz. Bu kitabxana aşağıdakı statistik göstəriciləri hesablamaq üçün müxtəlif funksiyalar təklif edir:

- .describe() – ümumi statistik xülasə
- .mean() – orta qiymət
- .median() – mediana
- .mode() – mod
- .std() – standart sapma
- .min() və .max() – minimum və maksimum dəyərlər
- .count() – elementlərin sayı
- .value_counts() – dəyərlərin tezliyi

Əlavə olaraq, məlumatların paylanmasını və dəyişənlər arasındakı əlaqəni vizuallaşdırmaq üçün **seaborn** kitabxanasından istifadə olunur.

Bu məqsədlə **histoqram, qutu diaqramı** (box plot) və **səpilmə qrafiki** (scatter plot) kimi təsviri qrafiklər qurmaq mümkündür.

Çıxarışlı statistik analizlər

Çıxarışlı statistik analizlər, bir nümunə məlumatına əsaslanaraq bütöv əhali haqqında ümumiləşdirmələr və təxminlər etmək üçün istifadə olunur. Bu təxminlərin nə qədər etibarlı və dəqiq olduğunu ölçmək də çıxarım sal statistikaların əsas məqsədidir. Məsələn, hipotezlərin yoxlanılması, etibar intervalları və əlaqə analizləri kimi üsullar bu kateqoriyaya daxildir.

Python-da bu işləri görmək üçün scipy kitabxanasının scipy.stats modulundan istifadə etmək olar. Burada t-testləri, ANOVA, ki-kvadrat testləri və Pearson korrelyasiya testləri kimi analizlər mövcuddur. Bundan əlavə, asılı dəyişən ilə bir və ya bir neçə müstəqil dəyişən arasındakı əlaqəni modelləşdirmək üçün statsmodels kitabxanası ilə reqressiya analizi aparmaq mümkündür.

Sadə desək, çıxarışlı statistik analizlər nümunədən əldə olunan nəticələri bütöv əhali üçün etibarlı şəkildə ümumiləşdirməyə kömək edir və Python kitabxanaları bunun üçün güclü alətlər təqdim edir.

Çox dəyişkənli statistikalar

Çox dəyişkənli statistikalar, iki və ya daha çox dəyişkəni əhatə edən məlumatları təhlil etmək, onların arasındakı qarşılıqlı əlaqələri və asılılıqları tapmaq üçün istifadə olunur.

Python-da bu növ analizləri həyata keçirmək üçün sklearn kitabxanasından yararlanmaq olar. Məsələn, əsas komponentlər təhlili (PCA), faktor təhlili (FA), kümə təhlili (CA) və ayırıcı təhlil (DA) kimi metodlarla çox dəyişkənli statistikalar aparılır.

Bu üsullar sayəsində məlumatların ölçüsünü azalda, gizli faktorları və ya qrupları müəyyən edə, həmçinin məlumatları fərqli kateqoriyalara və ya segmentlərə ayıra bilərsiniz.

Sadə desək, çox dəyişkənli statistikalar mürəkkəb məlumatlarda gizlənmiş nümunələri və əlaqələri tapmaq üçün güclü vasitədir.

Bayes statistikasi

Bayes statistikasi, Bayes teoremi adlı riyazi qaydaya əsaslanır və yeni sübutlar və ya verilənlər gələndə bir parametr və ya hipotez haqqında inancımızı yeniləmək fikrini ifadə edir.

Bayes statistikasi aparmaq üçün Python-da pymc3 kitabxanasından istifadə edə bilərsiniz. Bu kitabxana sizə ehtimallı modellər yaratmaq, əvvəlki (prior) və ehtimal (likelihood) paylanmalarını təyin etmək və Markov zənciri Monte Karlo (MCMC) üsulu ilə posterior paylanmadan nümunələr götürmək imkanı verir.

Nəticələri analiz etmək və vizuallaşdırmaq üçün isə arviz kitabxanası çox faydalıdır. Burada iz qrafikləri, posterior paylanma qrafikləri və orman qrafikləri kimi müxtəlif təsvirlərdən istifadə etməklə Bayes modellərinizin nəticələrini daha yaxşı anlaya bilərsiniz.

Qısaca desək, Bayes statistikasi məlumatlara əsaslanaraq inanc və ehtimallarımızı davamlı yeniləmək üçün güclü bir yanaşmadır.

Maşın öyrənməsi (ML)

Maşın öyrənməsi, verilərdən öyrənə bilən və gələcəkdə proqnozlar verə bilən və ya qərarlar qəbul edə bilən alqoritmlər və modellər yaratmaqla məşğul olan süni intellekt sahəsidir.

Python-da müxtəlif maşın öyrənməsi tapşırıqlarını yerinə yetirmək üçün sklearn kitabxanasından istifadə etmək olar. Buraya təsnifat (classification), reqressiya (regression), kümələmə (clustering), ölçü azaldılması (dimensionality reduction) və xüsusiyyət seçimi (feature selection) kimi metodlar daxildir.

Həmçinin, keras kitabxanası vasitəsilə sinir şəbəkələri yarada və təlim verə bilərsiniz. Sinir şəbəkələri verilərdəki mürəkkəb nümunələri və funksiyaları öyrənməkdə çox güclüdür.

Bundan əlavə, böyük və yüksək ölçülü veriləri — məsələn, şəkillər, mətn və səs kimi — emal etmək üçün istifadə olunan sinir şəbəkələrinin daha inkişaf etmiş növü olan dərin öyrənmə (deep learning) üçün tensorflow kitabxanasından yararlanmaq mümkündür.

Qısacası, maşın öyrənməsi verilərdən nümunələr çıxarmaq və ağıllı qərarlar vermək üçün geniş istifadə olunan texnologiyadır.

Zaman sırası təhlili

Zaman sırası təhlili, vaxt üzrə toplanan və zamansal datalar, meyllər, dövrlər və ya mövsümi təsirlər göstərən məlumatların analiz üsuludur.

Zaman sırası məlumatlarını işləmək və çevirmək üçün Python-da pandas kitabxanasından istifadə edə bilərsiniz.

Məsələn, yenidən nümunələmə (resampling), sürüşdürmə (shifting), yuvarlama (rolling) və toplama (aggregation) əməliyyatları burada mümkündür.

Müxtəlif zaman sırası təhlilləri — yumşaltma (smoothing), parçalama (decomposition), avtokorrelyasiya (autocorrelation), stasionarlıq testi, proqnozlaşdırma və ARIMA modelləri kimi — statsmodels kitabxanası ilə həyata keçirilə bilər.

Bundan əlavə, qeyri-xətti meylləri, dəyişiklik nöqtələrini, anomaliyaları və bayram effektlərini nəzərə alan zaman sırası modellərini yaratmaq və tətbiq etmək üçün fbprophet kitabxanası da geniş istifadə olunur.

Sadə desək, zaman sırası təhlili vaxtla dəyişən məlumatlardakı gizli nümunələri tapmaq və gələcəkdəki vəziyyəti proqnozlaşdırmaq üçün vacib alətdir.

Python ilə Əsas Statistikaya Giriş

Python-da əsas statistikanı öyrənmək üçün NumPy, SciPy, Pandas, Matplotlib və daxili statistics kitabxanalarından istifadə edə bilərsiniz. Bu vasitələr sizə məlumatları vizual şəkildə təsvir etmək, ümumiləşdirmək və təqdim etmək imkanı verir.

Müasir dünyada hər şey məlumatlara əsaslanır. Hər saniyə yaradılan məlumatların həcmi terabaytlarla ölçülür və buna görə də məlumatları anlamaq, analiz etmək və vizuallaşdırmaq çox önəmlidir. Statistik elmi məhz məhz bu ehtiyacdən doğur — məlumatlardan doğru nəticələr çıxarmaq və onları səmərəli təqdim etmək üçün istifadə olunur.

Python-un güclü kitabxanaları ilə siz məlumatlarınızı daha asan idarə edə, təhlil edə və başqalarına anlaşılır şəkildə göstərə bilərsiniz. Bu, həm tədqiqat, həm biznes, həm də digər sahələrdə uğur üçün vacib bacarıqdır.

Python Statistika Kitabxanalarının Seçilməsi

Python-da çoxlu statistik kitabxanalar mövcuddur, amma bu dərslikdə ən populyar və geniş istifadə olunan bir neçəsi ilə tanış olacaqsınız:

•**Python statistics**

Python-un daxili statistik kitabxanasıdır. Əgər məlumatlarınız çox böyük deyilsə və ya əlavə kitabxanaları idxal etmək istəmirsinizsə, bu kitabxanadan istifadə edə bilərsiniz.

•**NumPy**

Ədədi hesablamalar üçün optimallaşdırılmış üçüncü tərəf kitabxanasıdır. Bir və çoxölçülü massivlər (ndarray) ilə işləmək üçün geniş funksiyalar təqdim edir və statistik təhlil üçün çoxlu rutinlərə malikdir.

•**SciPy**

NumPy əsasında qurulmuş elmi hesablamalar kitabxanasıdır. NumPy-dən əlavə funksionallıqlar təklif edir, xüsusilə **scipy.stats** modulu statistik analiz üçün istifadə olunur.

•**Pandas**

NumPy üzərində qurulmuş rəqəmsal hesablamalar üçün kitabxanadır. Etiketlənmiş verilənlər ilə işləmək üçün uyğundur: birölçülü məlumatlar üçün **Series**, ikiölçülü məlumatlar üçün **DataFrame** obyektlərini təqdim edir.

•**Matplotlib**

Məlumatların vizuallaşdırılması üçün geniş istifadə olunan üçüncü tərəf kitabxanasıdır. NumPy, SciPy və Pandas ilə yaxşı inteqrasiya olunur və qrafiklər yaratmaq üçün əlverişlidir.

Statistikada dəyişənlər — müxtəlif qiymətlər ala bilən xüsusiyyətlər və ya parametrlərdir. İki əsas növə bölünürlər:

- Müstəqil dəyişənlər:** Təcrübədə idarə olunan və ya dəyişdirilən dəyişənlərdir.
- Asılı dəyişənlər:** Müstəqil dəyişənlərdəki dəyişikliklərə cavab olaraq ölçülən və ya müşahidə olunan dəyişənlərdir.
- Parametrlər və Statistikalar:**

Parametrlər populyasiyanı təsvir edən rəqəmsal ölçülərdir, məsələn, populyasiya orta qiyməti (μ - mu).

Statistikalar isə nümunəni təsvir edən rəqəmsal ölçülərdir, məsələn, nümunə orta qiyməti (\bar{x} - x bar).

Çıxarımsal Statistikalar

Çıxarımsal statistikalar tədqiqatçılara nümunə məlumatlarına əsaslanaraq ümumi əhali (populyasiya) haqqında nəticələr çıxarmağa və ya proqnozlar verməyə imkan yaradır. Bu metodlar müşahidə edilən məlumatların həddlərindən kənara çıxaraq ümumiləşdirmələr etməyə şərait yaradır. Çıxarımsal statistikada istifadə olunan əsas texnikalar bunlardır:

Hipotezlərin Sınanması (Test edilməsi):

Bu yanaşma nümunə məlumatları əsasında populyasiya parametrləri haqqında qərarlar qəbul etməyə kömək edir. Prosesi aşağıdakı addımları əhatə edir:

- Sıfır Hipotezi (H_0):** Populyasiya arasında əhəmiyyətli bir fərqin və ya əlaqənin olmadığını irəli sürür.
- Alternativ Hipotez (H_1):** Əhəmiyyətli bir fərq və ya əlaqənin mövcud olduğunu iddia edir.
- Test statistikas** seçilir, **əhəmiyyətlik səviyyəsi** (məsələn, $\alpha = 0.05$) müəyyən edilir və nəticələr şərh olunur.

❖ Etibarlılıq Aralıqları (Güvən Aralıqları):

Bu aralıqlar populyasiya parametrlərinin ehtimal olunan dəyərlər aralığını göstərir. Nümunədən əldə edilən təxminlərin nə qədər dəqiq olduğunu və həmin təxminlərə nə dərəcədə güvənilə biləcəyini əks etdirir.

❖ Reqressiya Analizi:

Reqressiya analizi bir və ya bir neçə müstəqil dəyişən ilə asılı dəyişən arasındakı əlaqəni öyrənir. Bu model müstəqil dəyişənlərə əsaslanaraq asılı dəyişənin dəyərini proqnozlaşdırmağa imkan verir.

❖ Nümunə Seçmə Metodları:

Sadə təsadüfi nümunə, təbəqəli nümunə və klaster nümunəsi kimi üsullar populyasiyanı dəqiq təmsil edən nümunələr toplamaq üçün istifadə olunur. Bu metodlar qərəzləri (bias) azaltmağa və nəticələrin ümumiləşdirilə bilməsini artırmağa kömək edir.

Statistik Təhlil Növləri

Statistik təhlil, müxtəlif məlumat növləri və tədqiqat suallarına uyğun olan müxtəlif metod və yanaşmaları əhatə edir. Hansı statistik texnikanın analiziniz üçün daha uyğun olduğunu müəyyən etmək üçün bu təhlil növlərini başa düşmək vacibdir. Aşağıda geniş yayılmış statistik analiz növlərinə dair əsas fərqləri təqdim edirik:

Parametrik və Parametrik Olmayan Təhlil

Statistik analizlər əsasən iki böyük kateqoriyaya bölünür: **parametrik** və **parametrik olmayan** üsullar. Hər iki yanaşmanın özünəməxsus fərziyyələri və tətbiq sahələri vardır.

Parametrik Təhlil

Parametrik metodlar məlumatların müəyyən bir ehtimal paylanması – çox vaxt **normal paylanması** – izlədiyini fərz edir. Bu metodlar populyasiya parametrlərini (məsələn, ortalama, dispersiya) qiymətləndirmək üçün istifadə olunur. Parametrik testlər adətən daha yüksək statistik gücə malikdir, lakin daha sərt fərziyyələrə əsaslanır.

Məşhur parametrik testlər:

- **t-testləri** (müstəqil və cütləşdirilmiş qruplar üçün)
- **ANOVA** (bir neçə qrup ortalamasının müqayisəsi)
- **Doğrusal reqressiya** (iki və ya daha çox dəyişən arasındakı əlaqənin modelləşdirilməsi)

Parametrik Olmayan Təhlil

Bu metodlar məlumatların konkret bir paylanma formasına sahib olduğunu fərz etmir. Parametrik olmayan testlər, çox zaman **sıralamalara** və ya paylanmadan asılı olmayan metodlara əsaslanır. Bu üsullar, məlumatlar parametrik fərziyyələrə cavab vermədikdə və ya **normal paylanmayan, ordinal** (sıralı) məlumatlarla işləyərkən tətbiq olunur.

Məşhur parametrik olmayan testlər:

- **Wilcoxon sıralama cəmi testi** (iki əlaqəli qrupun müqayisəsi)
- **Kruskal-Wallis testi** (bir neçə qrupun müqayisəsi üçün)
- **Spearman korrelyasiyası** (iki dəyişən arasında sıralı əlaqənin ölçülməsi)

Regressiya Təhlili

Regressiya analizi, **bir və ya bir neçə müstəqil dəyişən** ilə **asılı dəyişən** arasındakı əlaqəni araşdırmaq üçün istifadə olunan statistik üsuldur. Bu analiz növü, **proqnozlaşdırma modelləri** qurmaq və dəyişənlərin nəticələr üzərindəki təsirini başa düşmək üçün geniş tətbiq olunur.

Regressiya analizinin əsas növləri aşağıdakılardır:

❖ Sadə Xətti Regressiya

Sadə xətti regressiya, **bir müstəqil dəyişən** ilə **bir asılı dəyişən** arasında **doğrusal əlaqəni** təhlil edir. Məqsəd, bir dəyişənin digərini nə dərəcədə proqnozlaşdırma bildiyini müəyyən etməkdir. Məsələn, temperaturun dondurma satışlarına təsiri.

❖ Çoxsaylı Xətti Regressiya

Çoxsaylı (çox dəyişənli) xətti regressiya, sadə regressiyanın genişləndirilməsidir və **bir neçə müstəqil dəyişənlə bir asılı dəyişən** arasındakı əlaqəni qiymətləndirir. Bu yanaşma, kompleks səbəb-nəticə əlaqələrini daha dəqiq modelləşdirməyə imkan verir.

❖ Loqistik Regressiya

Loqistik regressiya, **ikili nəticələrin** (məsələn, "xəstədir/xəstə deyil", "bəli/xeyr") proqnozlaşdırılması üçün istifadə olunur. Burada nəticə dəyişəni kəmiyyət deyil, **kateqorial** olur. Modellər, hadisənin başvermə ehtimalını hesablamaq üçün istifadə olunur.

Dispersiya Analizi (ANOVA)

Dispersiya analizi və ya **ANOVA (Analysis of Variance)**, iki və ya daha çox qrupun ortalamaları arasında fərqləri müqayisə etmək üçün istifadə olunan statistik üsuldur. Bu metod, **ümumi dəyişkənliyi** müxtəlif mənbələrə — yəni **qruplar arası fərqlər** və **qrup daxilindəki dəyişkənlik** — bölərək təhlil edir. ANOVA, xüsusilə **eksperimental tədqiqatlarda** və **hipotez testlərində** geniş tətbiq olunur. Məsələn, müxtəlif dərmanların təsirinin müqayisəsi zamanı istifadə edilə bilər.

Zaman Sıralı Analiz (Zaman Seriyası Analizi)

Zaman sıralı analiz, müəyyən aralıqlarla toplanmış və ya qeydə alınmış məlumatların təhlili ilə məşğuldur. Bu analiz növü, məlumatlardakı **naxışları (pattern)**, **ümumi meyilləri (trendləri)** və **müntəzəm dövrlük (mövsümlilik)** kimi xüsusiyyətləri aşkar etməyə kömək edir. Aşağıdakı üsullar zaman seriyası analizində geniş istifadə olunur:

- **Trend Analizi:** Zamanla məlumatlarda baş verən uzunmüddətli dəyişiklikləri müəyyən edir.
- **Mövsümi Ayrılma:** Məlumatları **mövsümi**, **trend** və **qalıq** komponentlərinə ayıraraq təhlil edir.
- **Proqnozlaşdırma:** Tarixi məlumatlara əsaslanaraq **gələcək göstəriciləri təxmin** etməyə çalışır.

Sağ qalma analizi (Həyatda qalma analizi)

Sağ qalma analizi, bir fərdin və ya obyektin **ölüm, uğursuzluq və ya müəyyən bir hadisənin baş verməsi** kimi bir nəticəyə qədər olan **vaxt müddətini** təhlil etmək üçün istifadə olunan statistik üsuldur. Bu metod, **tibb sahəsində, mühəndislikdə** və **sosial elmlərdə** hadisənin baş vermə ehtimalını və **təhlükə (risk) səviyyəsini zamanla** qiymətləndirmək üçün geniş tətbiq olunur. Məsələn, bir dərmanın xəstənin sağ qalma müddətinə təsiri bu metodla öyrənilə bilər.

❖ Faktor analizi

Faktor analizi, müşahidə edilən dəyişənlər arasındakı **korrelyasiya (əlaqə) modellərini** izah edən **əsas faktorları** və ya **gizli dəyişənləri** müəyyən etmək üçün istifadə olunan statistik analiz üsuludur. Bu metod vasitəsilə, çoxlu sayda dəyişən arasında əlaqəli olan **ümumi strukturlar** üzə çıxarılır.

Faktor analizi xüsusilə **psixologiya, sosiologiya** və **bazar araşdırmaları** kimi sahələrdə, **alt ölçüləri** və ya **konseptual quruluşları** müəyyənləşdirmək məqsədilə geniş istifadə olunur.

Klasterləşdirmə Analizi (Kümələmə analizi)

❖ **Klasterləşdirmə analizi**, oxşar xüsusiyyətlərə malik olan obyektləri və ya müşahidələri **qruplara (klasterlərə)** ayıran çoxdəyişənli statistik bir metoddur. Bu üsul, **bənzər elementlərin bir-birinə yaxın, fərqli olanların isə uzaq yerləşdirildiyi** qruplaşdırma sistemini qurmağa imkan verir.

Klasterləşdirmə metodu **bazar segmentasiyası, rəqəmsal görüntülərin analizi və bioloji təsnifat** kimi sahələrdə geniş şəkildə tətbiq olunur.

❖ **Əsas Komponentlər Analizi (Principal Component Analysis – PCA)**

Əsas komponentlər analizi (PCA), yüksək ölçülü məlumatlarda olan dəyişkənliyin əsas hissəsini qoruyaraq, həmin məlumatları **daha aşağı ölçülü bir sahəyə proyeksiya etmək** üçün istifadə olunan ölçü azaldıcı statistik bir üsuldur. PCA, verilmiş məlumatdakı maksimum dəyişkənliyi tutan **ortoqonal oxları** (əsas komponentləri) müəyyən edir.

Bu metod, xüsusilə **məlumatların vizuallaşdırılması, xüsusiyyət seçimi (feature selection) və məlumatların sıxlaşdırılması** məqsədilə effektiv şəkildə istifadə olunur.

Doğru Statistik Analiz Metodunun Seçilməsi

Məlumatlar üzərində aparılan təhlildən doğru və mənalı nəticələr əldə etmək üçün **uyğun statistik analiz metodunun seçilməsi** olduqca vacibdir.

Məlumat Növlərini və Paylanmanı Anlamaq

Hər hansı statistik üsulu tətbiq etməzdən əvvəl, işlədiyiniz **məlumat növünü və onun paylanma formasını** başa düşmək çox önəmlidir. Müxtəlif statistik metodlar müxtəlif məlumat tipləri üçün uyğundur:

• **Davamlı və Kateqorik Məlumatlar:** Əvvəlcə məlumatlarınızın **davamlı** (məsələn, boy, çəki) və ya **kateqorik** (məsələn, cins, irq) olub olmadığını müəyyənləşdirin.

- **Davamlı məlumatlar** üçün, adətən **parametrik üsullar** (məsələn, t-testləri, regressiya analizi) istifadə olunur.
- **Kateqorik məlumatlar** üçün isə **parametrik olmayan metodlar**, məsələn **chi-kvadrat (χ^2) testi**, daha uyğun olur.

• **Normallıq Paylanması:** Məlumatlarınızın **normal paylanmaya** uyğun olub olmadığını qiymətləndirin. Çünki **parametrik analiz metodları** çox zaman normal paylanmanı tələb edir.

Əgər məlumatlarınız **normal paylanmadan kənardırsa**, bu halda **parametrik olmayan metodlardan** istifadə etmək daha düzgün olar.

Bayes Statistik Metodları

Bayes statistikas, əvvəlki bilikləri (aprior məlumatları) nəzərə alaraq, yeni məlumatlar əldə edildikcə ehtimalları yeniləyən yanaşmadır. Bu metod qeyri-müəyyənlik və məhdud məlumat şəraitində xüsusilə faydalıdır.

•**İyerarxik Bayes modelləri** — İç-içə və mürəkkəb quruluşa malik məlumat strukturlarının təhlili üçün istifadə olunur. Bu modellər, xüsusilə klinik tədqiqatlarda, müxtəlif xəstə alt qruplarında müalicə təsirini qiymətləndirmək üçün tətbiq olunur.

•**Bayes şəbəkələri** — Dəyişənlər arasında ehtimal əsaslı əlaqələri modelləşdirərək səbəb-nəticə münasibətlərini anlamağa imkan verir.

•**Markov Zənciri Monte Karlo (MCMC) üsulları** — Çətin analitik həllərin mümkün olmadığı hallarda mürəkkəb ehtimal paylanmalarını simulyasiya etməyə imkan verir. Bu üsullar fizika, mühəndislik və maliyyə kimi sahələrdə modellərin qurulmasında geniş istifadə olunur.

Qabaqcıl Statistik Alətlərin İnteqrasiyası

Müasir statistik proqram təminatları, bu qabaqcıl metodları istifadəçi dostu interfeyslərlə birləşdirərək, geniş proqramlaşdırma biliklərinə malik olmayan istifadəçilər üçün də əlçatan edir.

- Bulud hesablama platformaları** — Böyük həcmli məlumatların təhlili üçün genişlənə bilən və elastik hesablama resursları təqdim edir. Bu platformalar, təşkilatlara mürəkkəb statistik analizləri səmərəli və sürətli şəkildə həyata keçirməyə şərait yaradır.
- Real vaxt analitikası** — Axın məlumatlarını emal edərək statistik metodların inteqrasiyasını təmin edir. Bu texnologiya sayəsində təşkilatlar əməliyyatları davamlı izləyə, anomaliyaları vaxtında aşkar edə və dinamik şəraitə sürətli reaksiya verə bilirlər.

Məlumatların Təhlili üçün Populyar Statistik Proqram Alətləri

Statistik proqram paketləri məlumatların təhlili və tədqiqi prosesində mühüm rol oynayan effektiv platformalardır. Bu alətlər sadə statistik hesablamalardan tutmuş qabaqcıl analitik metodlara qədər geniş imkanlar təqdim edir.

SPSS: Sosial Elmlər üçün Statistik Paket

SPSS proqramı sosial elm tədqiqatçıları və biznes analitikləri arasında geniş istifadə olunur. İntuitiv və istifadəçi dostu interfeysi sayəsində, proqram kliklərlə və nöqtə ilə idarə etmə funksiyalarını qabaqcıl statistik analiz alətləri ilə birləşdirir.

SPSS-in verilənlər redaktoru böyük həcmli məlumatları səmərəli şəkildə idarə etməyə imkan verir. Bu proqram vasitəsilə istifadəçilər geniş proqramlaşdırma biliklərinə ehtiyac duymadan məlumatları təmizləyə, dəyişdirə və analiz üçün hazırlaya bilərlər.

Nəticələr SPSS-də təşkil olunmuş cədvəllər və qrafiklər şəklində çıxarılır ki, bu da tədqiqatçıların tapıntılarını rahatlıqla sənədləşdirməsinə, eləcə də peşəkar hesabatlar və təqdimatlar hazırlamasına şərait yaradır.

R Proqramlaşdırma Mühiti

R, statistik təhlil üçün güclü və açıq mənbə kodlu proqramlaşdırma mühitidir. Bu platforma tədqiqatçıların və statistik mütəxəssislərin istifadə etdiyi minlərlə ixtisaslaşmış paket vasitəsilə geniş fərdiləşdirmə imkanları təqdim edir.

R-də ən məşhur paketlərdən biri olan **ggplot2** yüksək keyfiyyətli və nəşr standartlarına uyğun qrafiklər yaratmağa imkan verir. Tədqiqatçılar mürəkkəb statistik nəticələri fərdiləşdirilə bilən qrafiklər və vizual hekayələr vasitəsilə daha effektiv şəkildə təqdim edirlər.

RStudio isə R proqramlaşdırma mühitini daha rahat və istifadəsi asan hala gətirən inteqrasiya olunmuş inkişaf mühitidir. Bu interfeys həm təcrübəli istifadəçilər üçün geniş funksionallıq təmin edir, həm də yeni başlayanların R-i daha rahat mənimsəməsinə kömək edir.

Statistik Analiz üçün Python

Python, xüsusilə **NumPy** və **SciPy** kimi kitabxanalar vasitəsilə çox yönlü və güclü statistik alət kimi tanınır. Bu paketlər ədədi hesablamalar və elmi təhlillər üçün geniş imkanlar təqdim edir.

Pandas kitabxanası isə verilənlərin manipulyasiyası və təhlilində xüsusi üstünlük təşkil edir. Analitiklər bu alətdən strukturlaşdırılmış məlumatları səmərəli idarə etmək və müxtəlif statistik əməliyyatları həyata keçirmək üçün istifadə edirlər.

Scikit-learn kitabxanası isə statistik analiz üçün maşın öyrənmə alqoritmlərini təmin edir. Bu vasitə tədqiqatçılara proqnozlaşdırıcı modellər qurmaq və qabaqcıl statistik prosedurları effektiv şəkildə tətbiq etmək imkanı verir.

SAS: Statistik Analiz Sistemi

SAS irimiqyaslı məlumatların təhlili üçün müəssisə səviyyəsində güclü statistik alətlər təqdim edir. Təşkilatlar SAS-ın sabitliyinə və böyük məlumat dəstlərini səmərəli idarə etmək qabiliyyətinə etibar edirlər.

SAS Enterprise Guide istifadəçilərə qrafik interfeys vasitəsilə statistik təhlil aparmaq imkanı verir. Bu xüsusiyyət, müxtəlif texniki bilik səviyyəsinə malik olan şəxslər üçün qabaqcıl analitik metodları əlçatan edir.

SAS Visual Analytics interaktiv tablolar və hesabatlar hazırlamağa imkan verir. Biznes analitikləri bu alətdən istifadə edərək təşkilatlar arasında fikirləri daha effektiv şəkildə çatdırırlar.

Stata Statistik Proqramı

Stata statistik təhlili və məlumatların idarə edilməsini bir arada təqdim edən güclü proqramdır. Xüsusilə iqtisadiyyat, biostatistika və epidemiologiya sahəsində aparılan tədqiqatlarda geniş istifadə olunur.

Proqramın komanda xətti interfeysi analitik prosedurlar üzərində dəqiq nəzarət imkanı verir. Tədqiqatçılar təhlillərin təkrar olunmasını və metodologiyaların düzgün sənədləşdirilməsini təmin etmək baxımından bu xüsusiyyəti yüksək qiymətləndirirlər.

Stata-nın qrafik redaktoru isə fərdiləşdirilə bilən vizuallaşdırmalar yaratmağa imkan verir. Bu alətlər tədqiqatçıların nəşr və təqdimat üçün hazır, peşəkar görünüşlü qrafiklər hazırlamasına kömək edir.

R Proqramlaşdırma Dili

R, məlumat elmi sahəsində ən güclü və ixtisaslaşmış proqramlaşdırma dillərindən biri hesab olunur. Python ilə müqayisədə daha çox statistik təhlil və məlumatların emalı üzrə ixtisaslaşmışdır. Bu xüsusiyyət onu məlumat alimləri (Data Scientist) üçün, xüsusilə də statistik analiz və nəticələrin vizuallaşdırılması ilə məşğul olan mütəxəssislər üçün son dərəcə faydalı edir. R böyük həcmli məlumat dəstlərini və mürəkkəb emal proseslərini idarə edə bilir. İstifadəçilərə məlumatların təmizlənməsi, çevrilməsi, modelləşdirilməsi və vizuallaşdırılması imkanlarını təqdim edir. Açıq mənbəli olması, müxtəlif platformalarda işləməsi və geniş kitabxana dəstəyinə malik olması onun əsas üstünlüklərindəndir.

R eyni zamanda paralel və paylanmış hesablama imkanlarına malikdir, bu da onu iri həcmli məlumatların emalında effektiv edir. ggplot2, dplyr, shiny və tidyverse kimi məşhur paketlər vasitəsilə yüksək keyfiyyətli qrafiklər və interaktiv vizual təqdimatlar yaratmaq mümkündür.

R kodlarının icrası üçün kompilyatora ehtiyac yoxdur və digər proqramlaşdırma dilləri ilə inteqrasiyada çevikdir. O, maşın öyrənməsi alqoritmlərinin tətbiqi üçün də geniş imkanlar təklif edir və RStudio kimi inkişaf etmiş proqramlaşdırma mühitləri ilə birgə istifadə oluna bilər.

Bununla belə, R dilinin bəzi məhdudiyyətləri də mövcuddur. Digər proqramlaşdırma dilləri ilə müqayisədə öyrənilməsi bir qədər çətin ola bilər. İri həcmli məlumatlarla işləyərkən sürət baxımından yavaşlıq və yüksək yaddaş istifadəsi müşahidə oluna bilər. R-in təhlükəsizlik baxımından zəifliyi və rəsmi texniki dəstək komandasının olmaması da bəzi istifadəçilər üçün problem yarada bilər. Üstəlik, standart kodlaşdırma qaydalarının olmaması proqramların oxunaqlılığını və saxlanmasını çətinləşdirə bilər.

Məlumatın Vizuallaşdırılması

“Sadə bir qrafik, məlumat analitiki üçün istənilən digər alətdən daha çox informasiya təqdim edə bilər.”

— Con Taker

Bu fəsildə siz R proqramlaşdırma mühitində **ggplot2** paketi vasitəsilə məlumatların necə vizuallaşdırıldığını öyrənəcəksiniz.

R dilində bir neçə fərqli qrafik qurma sistemi mövcuddur, lakin **ggplot2** həm funksionallığı, həm də estetik baxımdan ən güclü və çox yönlü sistemlərdən biridir.

ggplot2, "qrafik qrammatikası" (grammar of graphics) adlı yanaşmaya əsaslanır. Bu sistem vasitəsilə istifadəçilər qrafiklərin strukturunu daha dərinlən anlayır və bu biliyi müxtəlif kontekstlərdə tətbiq etməyi öyrənirlər. Beləliklə, bir dəfə öyrəndikdən sonra ggplot2-nin təqdim etdiyi imkanlardan geniş şəkildə istifadə etmək mümkündür.

Analizlərdə **ggplot2** paketi istifadə olunur. Bu paket **tidyverse** adlı daha geniş bir kolleksiyanın tərkib hissəsidir və məlumatların təmizlənməsi, vizuallaşdırılması və analizində çox faydalıdır.

Əgər sisteminizdə tidyverse paketi artıq quraşdırılmayıbsa, əvvəlcə onu aşağıdakı kimi quraşdırmaq lazımdır:

```
install.packages("tidyverse")
```

Quraşdırma tamamlandıqdan sonra isə paketi yükləmək üçün bu kodu işlədilir:

```
library(tidyverse)
```

Bu paketi yüklədikdə, tidyverse daxilindəki əsas funksiyalar digər R paketləri ilə toqquşa bilər. Sistem bu barədə sizi xəbərdar edəcək – yəni hansı funksiyanın hansı paketdən gəldiyini bildirəcək.

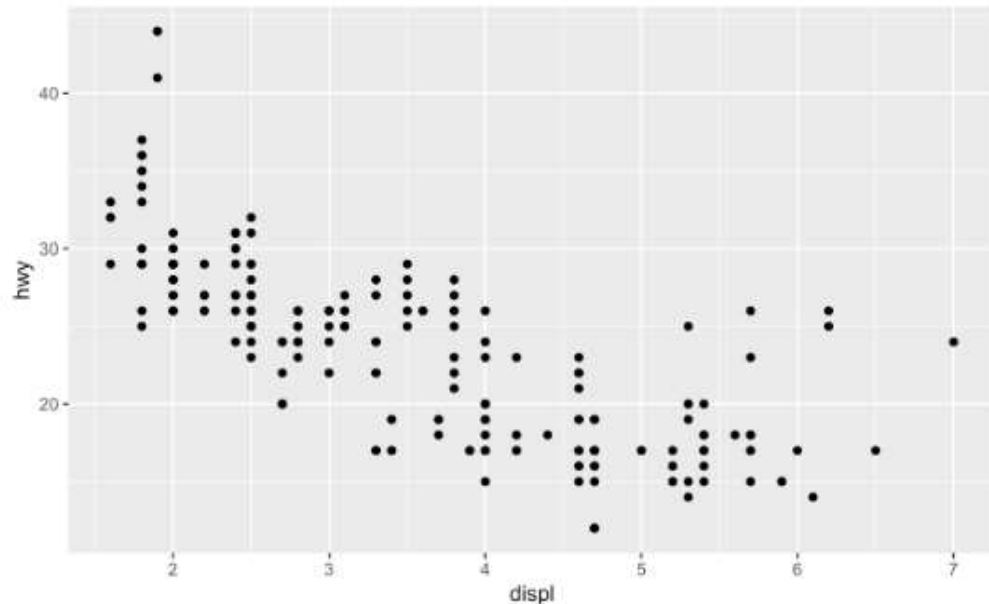
Əgər “tidyverse adlı paket yoxdur” kimi bir səhv çıxarsa, deməli, paket quraşdırılmayıb. Belə olan halda yuxarıdakı `install.packages()` sətirini mütləq icra etməliyik, sonra isə `library()` ilə yükləmək olar.

ggplot() funksiyası nə edir?

ggplot2 paketindəki ggplot() funksiyası vizuallaşdırmaya başlamaq üçün ilk addımdır. Bu funksiya bir növ **koordinat sistemi yaradır**, daha sonra bu sistemin üzərinə müxtəlif **qatlar (layers)** əlavə edə bilirik.

Məsələn, ggplot(data = mpg) sadəcə mpg adlı məlumat çərçivəsi üzərində boş bir qrafik qurur. Bu özü-özlüyündə hələ maraqlı bir nəticə göstərmir. Ona görə də bu mərhələdə həmin qrafiki göstərmək hələ lazım deyil — çünki o, vizual baxımdan hələ tam formalaşmayıb.

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```



Qrafikə qatlar əlavə etmək

ggplot() funksiyası təkcə başlanğıcdır. Əsl qrafiki yaratmaq üçün ona **qatlar (layers)** əlavə etmək lazımdır. Bu qatlar vasitəsilə məlumatları vizual formada təqdim edə bilərik. Məsələn, geom_point() funksiyası **səpələnmə (nöqtə) qrafiki** yaratmaq üçün istifadə olunur. Bu funksiya qrafikə **nöqtə qatı** əlavə edir.

ggplot2 paketində müxtəlif **geom funksiyaları** mövcuddur və hər biri qrafikə fərqli vizual qatlar əlavə etmək üçün istifadə olunur. Bu fəsildə həmin geom_ funksiyaları barədə daha ətraflı öyrənəcəksiniz.

mapping nədir?

ggplot2 paketində istifadə olunan hər bir geom_ funksiyası **mapping** adlı bir parametr tələb edir. mapping funksiyası qrafikdə hansı **dəyişənin hansı vizual xüsusiyyətə** (məsələn, x və ya y oxuna) uyğunlaşdırılacağını təyin edir.

Bu uyğunlaşdırma **aes()** adlı funksiya vasitəsilə verilir. aes() funksiyası daxilində **x və y oxları üçün hansı dəyişənlərin istifadə olunacağını** göstəririk.

```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy))
```

burada:

- data = mpg — istifadə etdiyimiz məlumat çərçivəsini göstərir,
 - geom_point() — nöqtə qrafiki yaradır,
 - mapping = aes(x = displ, y = hwy) — displ dəyişəni x oxuna, hwy isə y oxuna uyğunlaşdırılır.
- Beləliklə, bu kod bizə mühərrik həcmi ilə yanacaq sərfiyyəti arasındakı əlaqəni göstərən səpələnmə qrafiki təqdim edir.

R proqramlaşdırma dilində birbaşa işləyə və ya qrafik interfeysə malik olan **RStudio** proqramının zəngin imkanlarından yararlanı bilərsiniz. **RStudio** istifadəsi rahat olan və tətbiqləri asanlaşdıran **entegre edilmiş inkişaf mühiti (Integrated Development Environment - IDE)** kimi tanınır. Bu proqram **Windows, Mac** və **Linux** əməliyyat sistemlərində problemsiz işləyir.

RStudio-nu faydalı edən əsas xüsusiyyətlərdən bəziləri aşağıdakılardır:

- **Skript pəncərəsi** – kodlarınızı yadda saxlayaraq layihə formasında işləməyə imkan verir.
- **Kod tarixçəsi** – daha əvvəl yazdığınız kodlara qayıda bilərsiniz.
- **Güclü qrafik altyapısı** – vizuallaşdırma üçün mükəmməl vasitələr təqdim edir.
- **Cheatsheet-lər** – tez-tez istifadə olunan funksiyalar və sintaksis üçün qısa bələdçilər.
- **Dəyişən və funksiya tamamlaması** – yazarkən avtomatik tamamlanma ilə kodlaşdırmanı asanlaşdırır.

Bu imkanlar sayəsində RStudio həm yeni başlayanlar, həm də təcrübəli istifadəçilər üçün güclü və rahat bir mühit təqdim edir.

Aritmetik Əməliyyatlar

R proqramı ən sadə halda kalkulyator kimi istifadə edilə bilər. Toplama üçün +, çıxma üçün -, vurma üçün *, bölmə üçün isə / operatorlarından istifadə olunur. Eyni sətrdə bir neçə riyazi əməliyyat da yerinə yetirmək mümkündür.

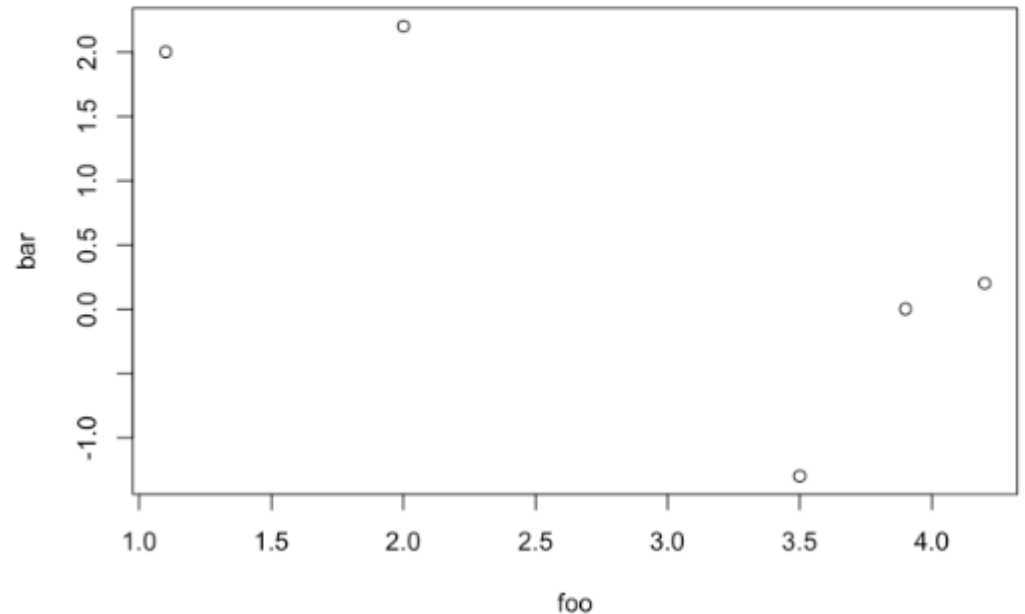
```
5+4
```

```
## [1] 9
```


Qrafiklər

R proqramının ən vacib xüsusiyyətlərindən biri verilənlərin vizuallaşdırılmasıdır. Ona görə də R çox zəngin qrafik imkanlarına malikdir. Qrafiklərin görünüşünü fərqləndirmək üçün müxtəlif parametrlərdən istifadə etmək mümkündür. Bu parametrlər nöqtələrin rənglənməsi, oxların üzərində etiketlərin əlavə olunması kimi sadə vizual təkmilləşdirmələri təmin edir, həmçinin diaqramın texniki xüsusiyyətlərinə də nəzarət etməyə imkan verir.

```
foo <- c(1.1,2,3.5,3.9,4.2)
bar <- c(2,2.2,-1.3,0,0.2)
plot(foo,bar)
```



Tip: R-ə koordinatları necə düzüləcəyini göstərir (məsələn, yalnız nöqtələr, yalnız xətlər və ya həm nöqtələr, həm də xətlər kimi).

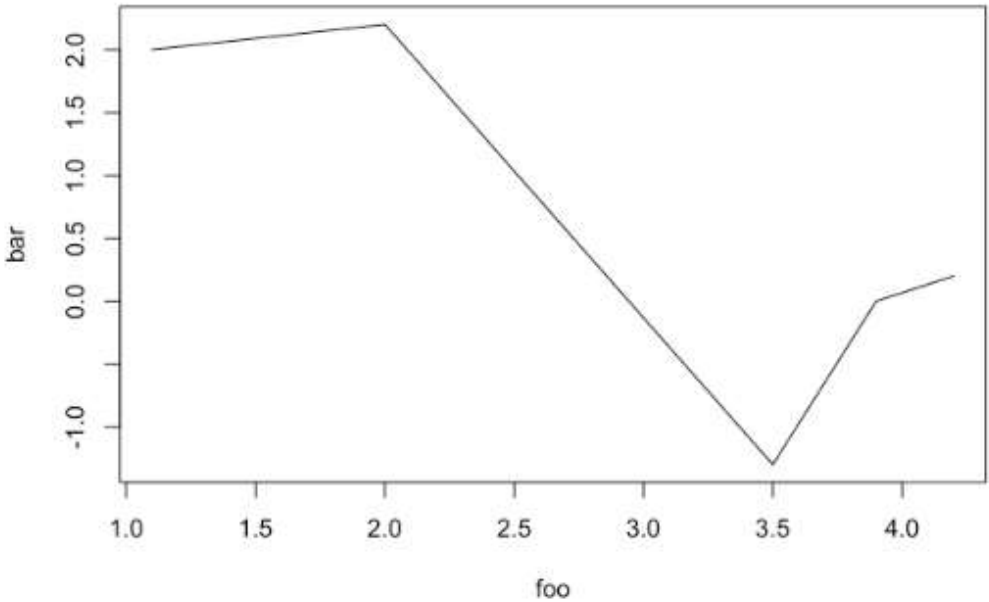
Tip funksiyası R proqramında qrafik yaradarkən, verilən nöqtələrin necə göstəriləcəyini müəyyən edir. Məsələn, müstəqil nöqtələr kimi (scatter plot), sadəcə xətlər kimi (line plot) və ya həm nöqtələr, həm də xətlər (point and line plot) şəklində təqdim etmək mümkündür. Bu seçim qrafikin vizual təsirini və məlumatların təqdim olunma tərzini təsir edir.

Misal üçün, `geom_point()` funksiyası məlumatları nöqtələr şəklində göstərir, `geom_line()` isə nöqtələri birləşdirən xətlər yaradır.

`geom_point()` və `geom_line()` birlikdə istifadə olunarsa, həm nöqtələr, həm də onları birləşdirən xətlər göstərilmiş olur. Bu cür tərtibat məlumatların təhlilində fərqli nüansları daha yaxşı anlamağa kömək edir.

```
plot(foo, bar, type="l")
```

Yəni, “tip” burada R-ə bildirir ki, qrafikdə məlumat nöqtələri necə təsvir olunacaq — sadəcə nöqtə, xətt, yoxsa hər ikisi. Bu seçim vizuallaşdırmanın məqsədindən və məlumatların növündən asılı olaraq dəyişə bilər.



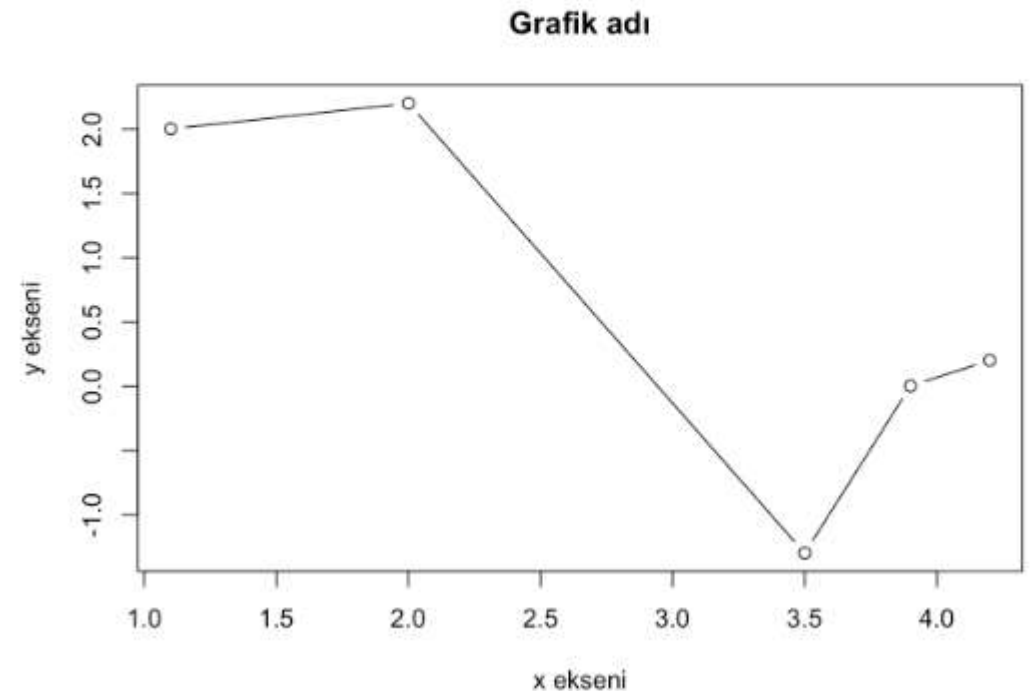
main, **xlab**, və **ylab** parametrləri R-də qrafiklər yaradarkən qrafikin başlığını və oxların etiketlərini təyin etmək üçün istifadə olunur. Onların funksiyaları və əhəmiyyəti aşağıdakı kimidir:

- **main** – Qrafikin əsas başlığını göstərir. Bu başlıq qrafikin yuxarısında yerləşir və qrafikin mövzusunun və ya göstərdiyi məlumatları ümumi şəkildə izah edir. Məsələn, “Mühərrik Ölçüsü və Yanacaq Səmərəliliyi” kimi bir başlıq qrafikin nəyi əks etdirdiyini aydınlaşdırır.

- **xlab** – Üfüqi oxun (x oxu) etiketidir. Bu, həmin oxda göstərilən məlumatın nə olduğunu bildirir. Məsələn, əgər x oxu mühərrik həcmi göstərsə, bu parametrlə “Mühərrik Həcmi (litr)” kimi bir yazı əlavə olunur ki, qrafiki izləyən şəxs bu oxun mənasını dərhal başa düşsün.

- **ylab** – Şaquli oxun (y oxu) etiketidir. Bu da y oxunda verilən məlumatın təsvirini verir. Məsələn, “Yanacaq Səmərəliliyi (mpg)” kimi etiket y oxundakı dəyərlərin nəyi ifadə etdiyini açıqlayır.

```
plot(foo, bar, type="b", main="Grafik adı", xlab="x ekseni",  
      ylab="y ekseni")
```



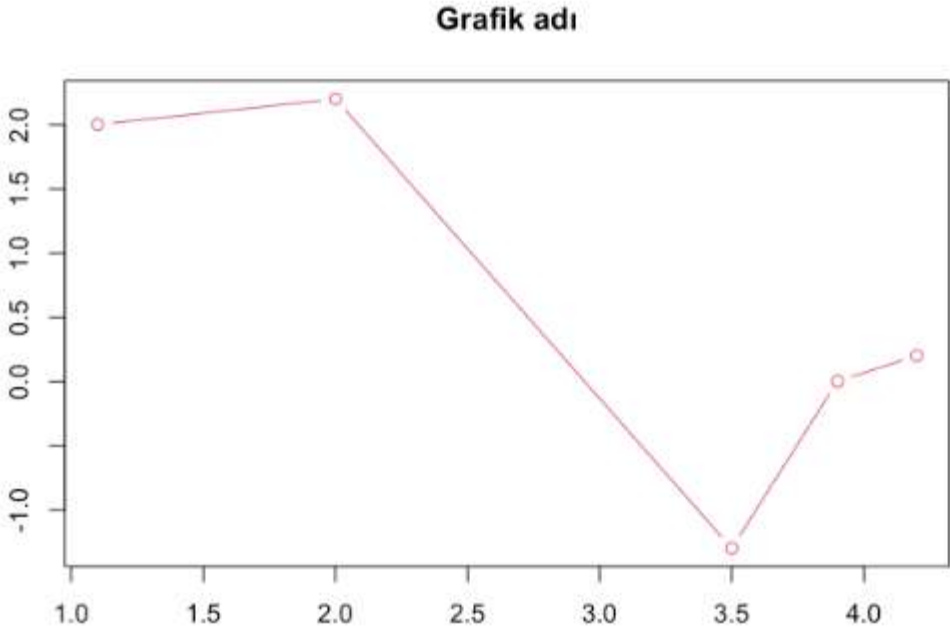
col parametri R-də qrafiklər yaratarkən nöqtələrin, xətlərin, çubuqların və digər vizual elementlərin rəngini təyin etmək üçün istifadə olunur. Bu parametr qrafikin görünüşünü fərdiləşdirmək və məlumatların daha aydın və təsirli təqdim olunması üçün vacibdir.

col-un əsas xüsusiyyətləri və istifadəsi:

•Rənglərin seçimi:

col arqumentinə rəng adları (məsələn, "red", "blue", "green"), rəng kodları (məsələn, "#FF0000", "#0000FF") və ya rəqəmlərlə göstərilən rəng indeksləri verilə bilər. Bu, qrafikdə istədiyiniz rəngi seçməyə imkan verir.

```
plot(foo,bar,type="b",main="Grafik adı",xlab="",ylab="",col=2)
```



•**Məlumatların qruplaşdırılması və rəngləndirilməsi:**

Əgər verilənlərinizdə fərqli kateqoriyalar varsa, col parametri vasitəsilə hər bir qrup üçün fərqli rəng təyin edə bilərsiniz.

Bu, vizual müqayisəni asanlaşdırır və qrafiki daha informativ edir.

•**Daha yaxşı görünüş və vurğu:**

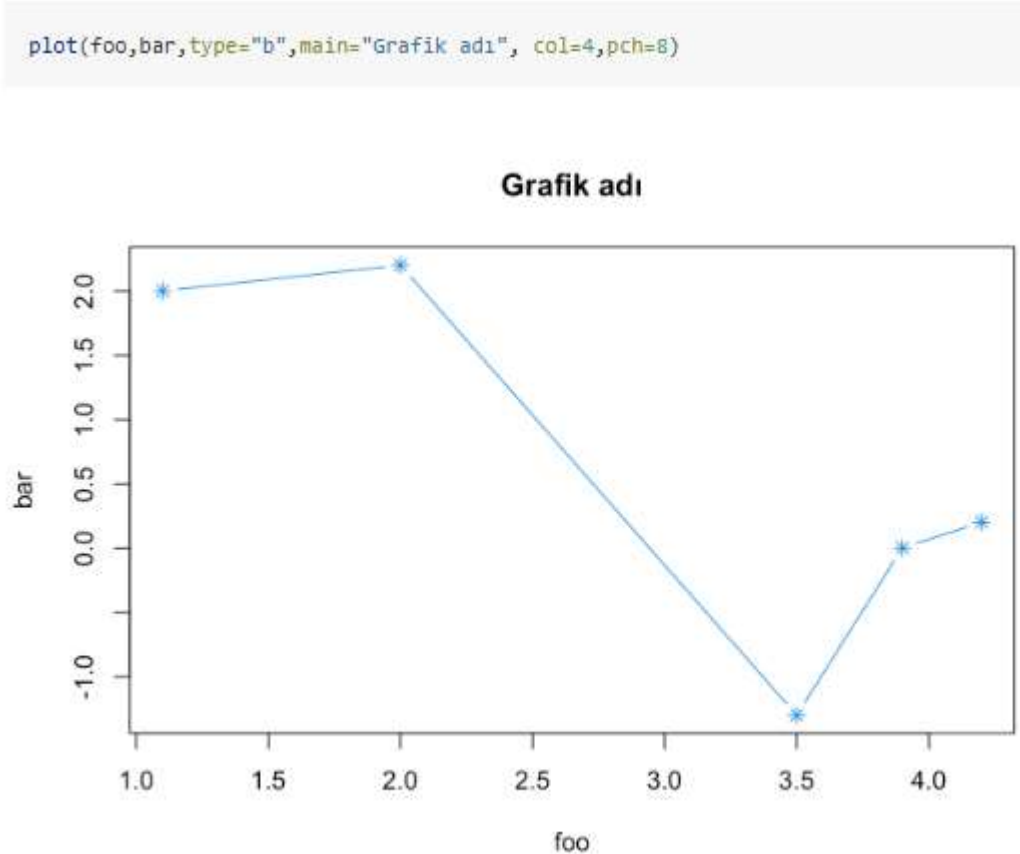
Rənglərin düzgün seçilməsi qrafikin daha cəlbedici olmasını təmin edir və məlumatların əsas məqamlarını vurğulamağa kömək edir. Məsələn, vacib nöqtələri qırmızı rənglə göstərmək, digər məlumatlardan fərqləndirir.

Nəticə olaraq, col parametri qrafikdəki elementlərin rəngini təyin etməklə məlumatların daha aydın, fərqli və vizual cəhətdən cəlbedici təqdim edilməsini təmin edir. Rənglərin məqsədyönlü və düzgün seçilməsi qrafiklərin effektivliyini artırır və izləyicinin diqqətini əsas nöqtələrə yönəltməyə kömək edir.

pch parametri R-də qrafiklərdə nöqtələrin formasını müəyyən etmək üçün istifadə olunur. Bu parametr xüsusilə səpələnmiş qrafikləri (scatter plots) və digər nöqtə əsaslı qrafiklərdə nöqtələrin necə göründüyünü tənzimləmək üçün çox faydalıdır.

pch nədir və nə üçün istifadə olunur?

- **pch** — "plotting character" sözlərinin qısaltmasıdır və nöqtələrin, simvolların qrafikdə necə göründüyünü təyin edir.
- Bu parametr nöqtənin şəklini seçməyə imkan verir: dairə, kvadrat, üçbucaq, ulduz, plus işarəsi və s.
- Hər bir pch dəyəri müəyyən bir simvolu ifadə edir. Məsələn, pch = 1 dairə, pch = 2 üçbucaq, pch = 3 plus işarəsi və s.
- **R**-də 0-dan 25-ə qədər standart pch dəyərləri mövcuddur və hər biri fərqli bir simvol təmsil edir.



pch-in istifadəsi və üstünlükləri:

- Nöqtələri müxtəlif kateqoriyalara görə fərqləndirmək üçün fərqli pch dəyərləri təyin edilə bilər. Məsələn, fərqli qrupları fərqli simvollarla göstərərək məlumatların qrafikdə daha aydın görünməsini təmin etmək olur.
- Qrafikdə nöqtələrin vizual olaraq daha cəlbedici və anlaşılan olmasına kömək edir.
- Rənglərlə birlikdə istifadə edildikdə, həm rəng, həm də forma ilə qrupların fərqləndirilməsi mümkün olur.

Nəticə:

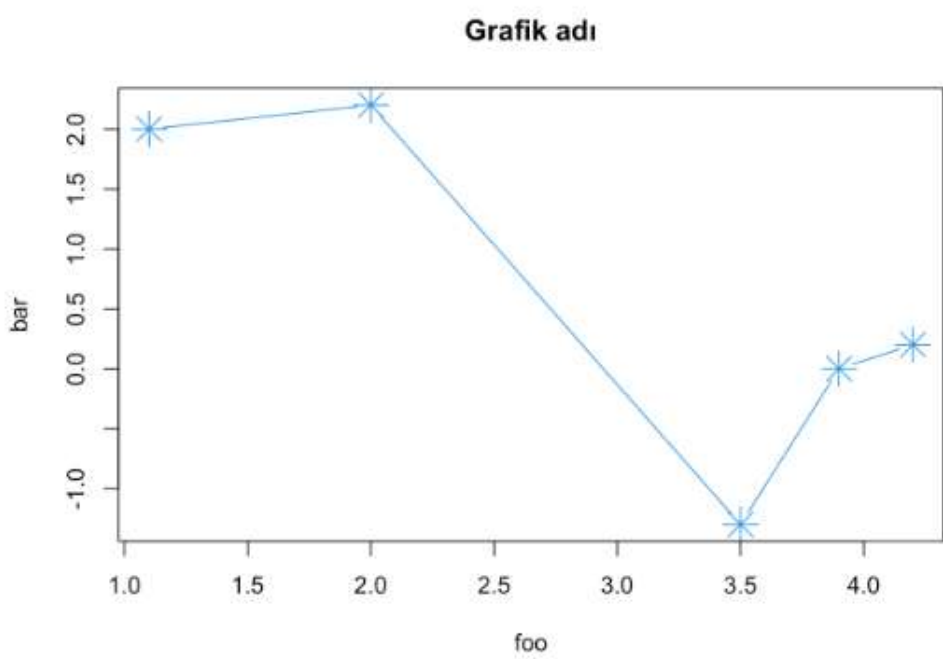
pch parametri qrafiklərdə nöqtələrin formasını seçmək üçün istifadə olunur və məlumatların qrafik üzərində daha fərqli və vizual cəhətdən anlaşılan təqdimatı üçün çox önəmlidir. Bu parametr sayəsində, müxtəlif kateqoriyalar və məlumat qrupları fərqli simvollarla göstərilərək qrafikin interpretasiyası asanlaşır.

cex parametri R-də qrafiklərdə nöqtə, mətn və digər simvolların ölçüsünü tənzimləmək üçün istifadə olunur. Bu parametr qrafik elementlərinin vizual olaraq nə qədər böyük və ya kiçik görünəcəyini müəyyən edir.

cex nədir və necə işləyir?

- **cex** — "character expansion" sözlərinin qısaltmasıdır, yəni simvolun ölçüsünü böyütmək və ya kiçiltmək deməkdir.
- Əsas ölçü dəyəri **1** olaraq qəbul edilir. Məsələn, $cex = 1$ standart ölçüdür, $cex = 2$ simvolu iki dəfə böyük göstərir, $cex = 0.5$ isə yarı ölçüdə göstərir.
- **cex** həm nöqtələr, həm də mətnlər, ox etiketləri və digər qrafik elementlərinin ölçüsünü dəyişmək üçün istifadə oluna bilər.

```
plot(foo,bar,type="b",main="Grafik adı", col=4,pch=8, cex=2.3)
```



cex-in istifadəsi və üstünlükləri:

- Qrafikdəki nöqtələri və digər elementləri daha aydın və diqqət çəkici etmək üçün ölçüsünü böyütmək mümkündür.
- Kiçik ölçülü nöqtələr və simvollar qrafiki daha sıx və az vizual yüklü göstərir.
- Müxtəlif qruplar üçün fərqli ölçülərdə nöqtələrdən istifadə etməklə qrafikin daha zəngin və informativ olması təmin edilir.
- Mətnlərdə və başlıqlarda ölçünün tənzimlənməsi oxunurluğu artırır.

Nəticə:

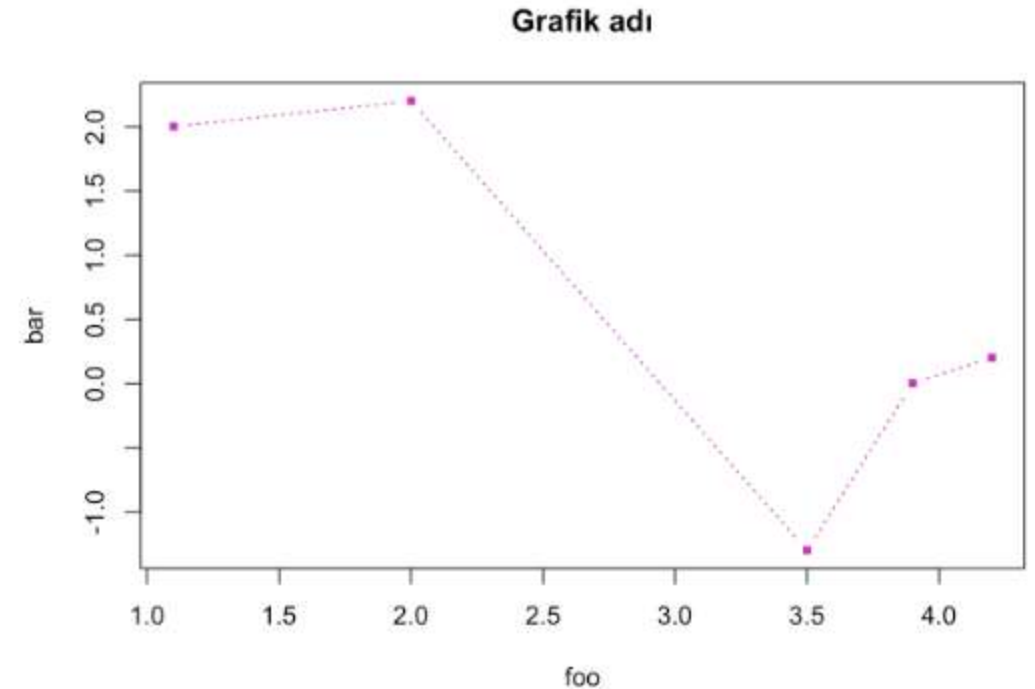
cex parametri R qrafiklərində nöqtə və digər simvolların ölçüsünü idarə etməyə imkan verir. Bu, qrafikin oxunaqlılığını və vizual təsirini artırmaq üçün vacib bir xüsusiyyətdir. İstifadəçi **cex** dəyərini dəyişərək qrafikin müxtəlif hissələrini (nöqtələr, mətnlər, başlıqlar) öz ehtiyacına uyğun şəkildə tənzimləyə bilər.

lty R qrafiklərində xəttin növünü, yəni xəttin necə görünəcəyini müəyyən etmək üçün istifadə olunan parametredir. Bu parametr xəttin kəsik-kəsik, nöqtəli, qatı və ya digər müxtəlif formalarını seçməyə imkan verir.

lty nədir və necə işləyir?

- **lty** — "line type" sözlərinin qısaltmasıdır və xətt növünü bildirir.
- Xətt növü qrafikdəki xəttin vizual görünüşünü dəyişir.
- Standart olaraq, **lty = 1** qatı (düz) xətti göstərir.

```
plot(foo,bar,type="b",main="Grafik adı", col=6,pch=15,cex=0.7, lty=3)
```



İty parametrinin istifadəsi və faydaları:

- Qrafikdə müxtəlif xətt növləri istifadə etməklə məlumatların qruplarını və ya fərqli kateqoriyaları vizual olaraq ayırmaq mümkündür.
- Xəttin növünü dəyişərək qrafik daha anlaşılan və oxunaqlı olur.
- Nöqtəli və zolaqlı xətlər məlumatlarda xüsusi vurğular yaratmaq və ya fərqli təbəqələri göstərmək üçün faydalıdır.
- Çap edilən və ya rəngsiz qrafiklərdə xətt növləri fərqli məlumatların ayırd edilməsində əsas vasitədir.

Nəticə:

İty parametri qrafiklərdə xəttin formasını təyin edir və vizual fərqlilik yaratmaq üçün çox önəmlidir. İstifadəçi qrafikin daha aydın və peşəkar görünməsi üçün xətt növlərini rahatlıqla dəyişə bilər. Bu, xüsusilə çoxsaylı xətlərin eyni qrafikdə göstərildiyi hallarda informasiyanın daha yaxşı başa düşülməsinə kömək edir.

lwd (İngiliscə: *line width* sözlərinin qısaltması) — **R proqramlaşdırma dilində** və bəzi statistik qrafik alətlərində **qrafik üzərində çəkilən xətlərin qalınlığını** (enini) təyin etmək üçün istifadə olunan bir parametrdir.

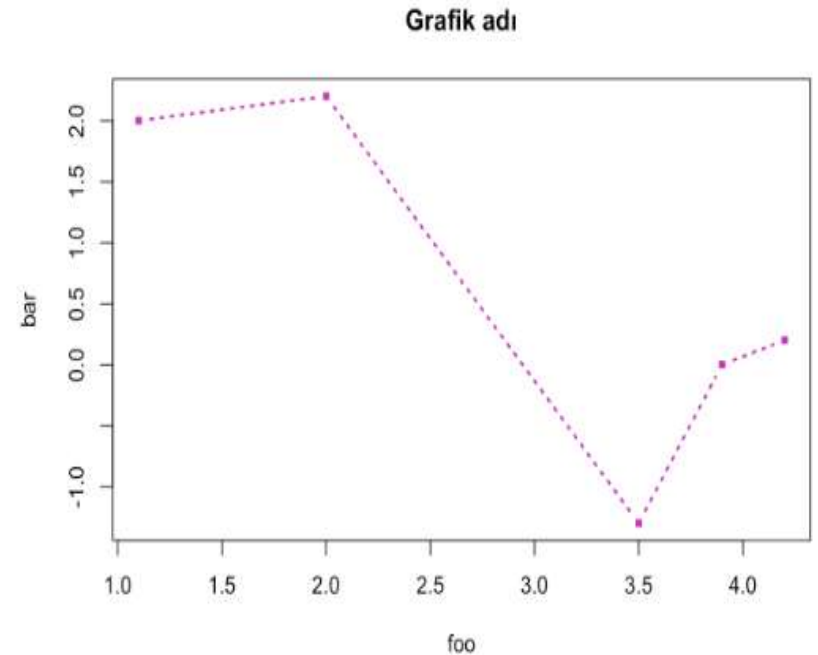
lwd parametri adətən **plot()**, **lines()**, **abline()**, **points()** və s. funksiyalarla birlikdə istifadə olunur. Bu parametr vasitəsilə xəttlərin daha qalın və ya incə görünməsini təmin edə bilərsiniz.

```
plot(foo,bar,type="b",main="Grafik adı",  
col=6,pch=15,lty=3,cex=0.7,lwd=2)
```

***lwd* üçün dəyərlər:**

- $lwd = 1$ — Ən sadə (default) qalınlıq.
- $lwd = 2, 3, 4$ və s. — Daha qalın xətlər.
- $lwd = 0.5$ — Daha incə xətt.

Əlavə olaraq, lwd dəyəri real ədədlərlə tənzimlənir və 0 olarsa xətt görünməz ola bilər.



Nəticə:

lwd — **qrafiklərdə xəttlərin qalınlığını təyin etməyə imkan verən vizual nəzarət vasitəsidir**. Qrafikdəki vacib xətləri ön plana çıxarmaq və ya məlumatları daha yaxşı vurğulamaq üçün istifadə olunur.

ərməyə bəzi.

Məsələn, `ylim` — Qrafikdə şaquli oxun (yəni **y oxunun**) başlanğıc və son nöqtəsini təyin edir. **aradan qaldırmaq** istəyə bilərsiniz.

❖ **xlim (x-axis limits)** — Qrafikdə üfüqi oxun (yəni **x oxunun**) başlanğıc və son nöqtəsini təyin edir.

Bunun üçün `xlim` və `ylim` parametrləri **el ilə təyin olunur**, beləliklə nəticə daha dəqiq və idarəolunan olur.

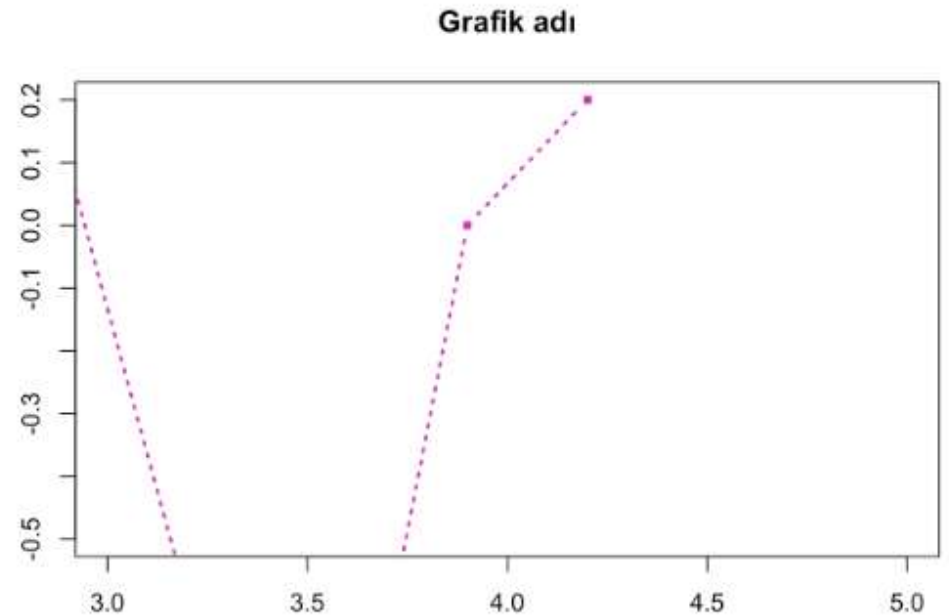
Bu parametr sayəsində `x` oxunun hansı aralıqda göstəriləcəyini istədiyiniz kimi seçə bilərsiniz.

❖ **ylim nədir?**

ylim (y-axis limits) — Qrafikdə **şaquli oxun** (yəni **y oxunun**) başlanğıc və son nöqtəsini təyin edir.

Bu parametr qrafikin dikey (şaquli) sahəsini idarə etməyə imkan verir.

```
plot(foo,bar,type="b",main="Grafik adı",xlab="",ylab="",
      col=6,pch=15,lty=3,cex=0.7,lwd=2,xlim=c(3,5),ylim=c(-0.5,0.2))
```



Qrafik çəkərkən, bəzən avtomatik olaraq təyin olunan ox limitləri (sərhədləri) **istədiyiniz vizual nəticəni verməyə bilər**.

Məsələn, bəzi dəyərləri xüsusi olaraq **böyüdülmüş miqyasda** göstərmək və ya **gərəksiz boşluqları aradan qaldırmaq** istəyə bilərsiniz.

Bunun üçün xlim və ylim parametrləri **əl ilə təyin olunur**, beləliklə **nəticə daha dəqiq və idarəolunan** olur.

- ❖ Vizual təhlil zamanı müəyyən bir aralığa fokuslanmaq üçün xlim və ylim istifadə etmək çox vacibdir.
- ❖ Bəzi hallarda, **həddindən artıq böyük və ya kiçik dəyərlər** avtomatik miqyası pozur — bu zaman öz sərhədlərinizi təyin etməklə qrafiki daha **idarəolunan və oxunaqlı** hala gətirə bilərsiniz.

lattice — R proqramlaşdırma dilində **çoxölçülü (multi-dimensional)** və **strukturlaşdırılmış** qrafiklər qurmaq üçün istifadə olunan bir vizualizasiya paketidir.

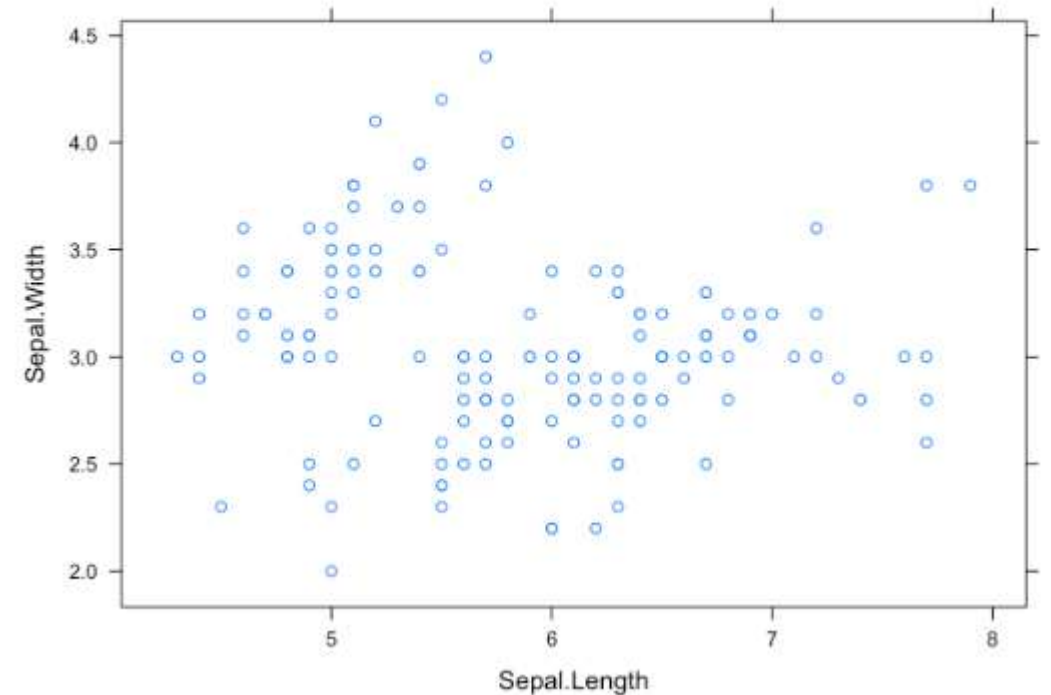
Bu paket **kompleks qrafikləri asanlıqla yaratmağa**, xüsusilə **kateqorik dəyişənlərə görə qruplaşdırılmış** təsvirlər təqdim etməyə imkan verir.

Niyə lattice istifadə olunur?

- Dəyişənlərin qarşılıqlı əlaqəsini vizual göstərmək.**
- Qruplara görə (kateqorik dəyişənlər əsasında) fərqli qrafiklər yaratmaq.**
- Fərqli panel (şəkil) strukturunda qrafiklər çəkmək** (hər kateqoriya üçün ayrıca panel).

- **lattice paketi** — R dilində kompleks, strukturlaşdırılmış və çoxölçülü qrafiklər yaratmaq üçün ideal vasitədir.
- **Kateqorik dəyişənlərə görə qrafikləri fərqli panellərdə** göstərmək bu paketin əsas üstünlüklərindəndir.
- iris verilənlərindəki Species kateqoriyası əsasında **bütün ölçüləri ayrı-ayrı paneldə müqayisə etmək mümkündür.**

```
lattice::xyplot(Sepal.Width ~ Sepal.Length, data = iris)
```



R-də daxili olan **iris** verilənləri çiçəklərin morfoloji ölçülərini ehtiva edir. Buradakı Species dəyişəni **kateqorik dəyişəndir** və 3 növü əhatə edir:

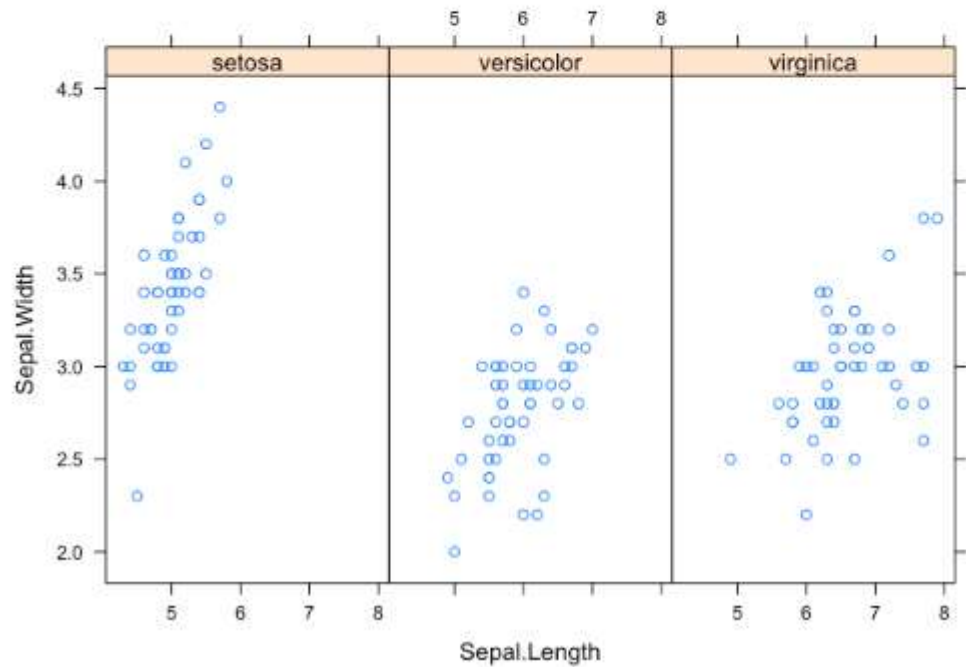
- setosa
- versicolor
- virginica

Bu növlərə görə fərqli qrafiklər qurmaq mümkündür.

```
levels(iris$Species)

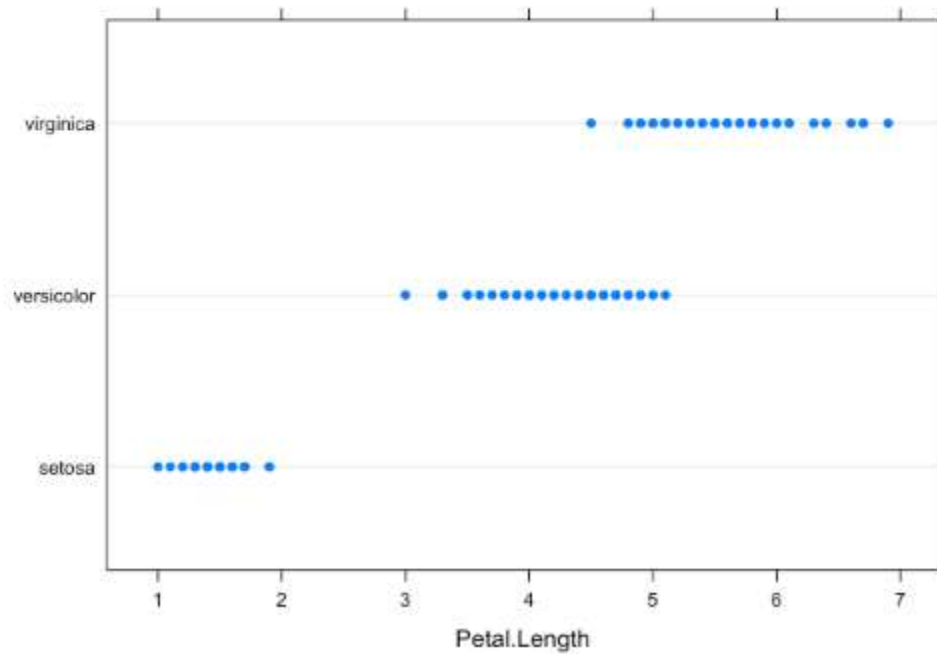
## [1] "setosa"      "versicolor" "virginica"

lattice::xyplot(Sepal.Width ~ Sepal.Length | Species, data = iris)
```



Paketin əsas məqsədi tək bir funksiyanın köməyi ilə mürəkkəb qrafiklər çəkə bilməkdir, ona görə də müxtəlif qrafik növləri aşağıda nümunə göstərilmişdir.

```
lattice::dotplot(Species ~ Petal.Length, data = iris)
```



lattice funksiyaları:

Funksiya

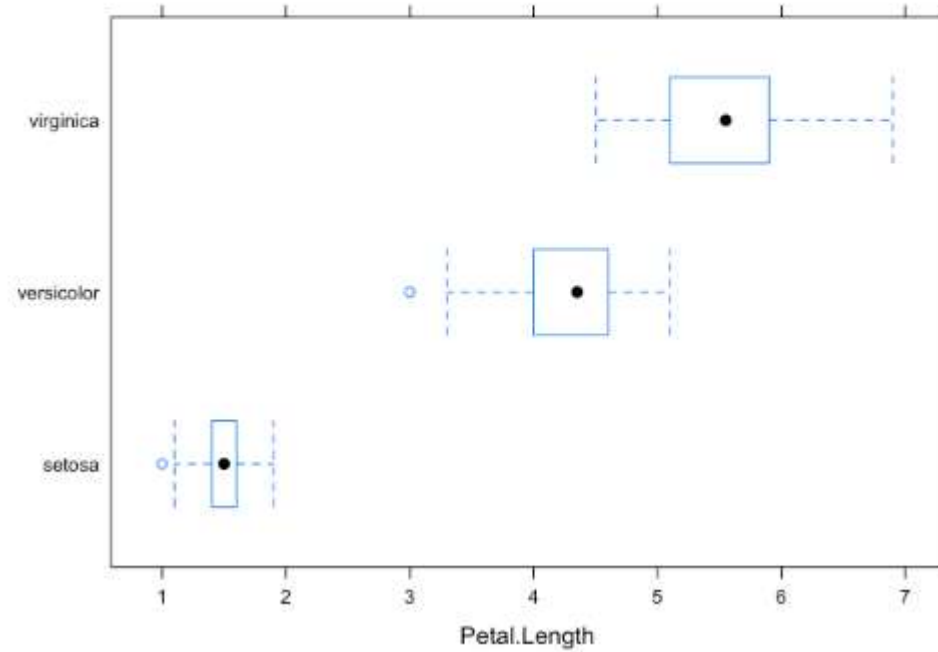
bwplot()

densityplot()

histogram()

dotplot()

```
lattice::bwplot(Species ~ Petal.Length, data = iris)
```



Təsviri

Boxplot (qutu qrafiki) yaratmaq üçün

Sıxlıq funksiyası (distribution) göstərmək

Histogram yaratmaq üçün

Kateqorik dəyişənlərlə nöqtə qrafiki

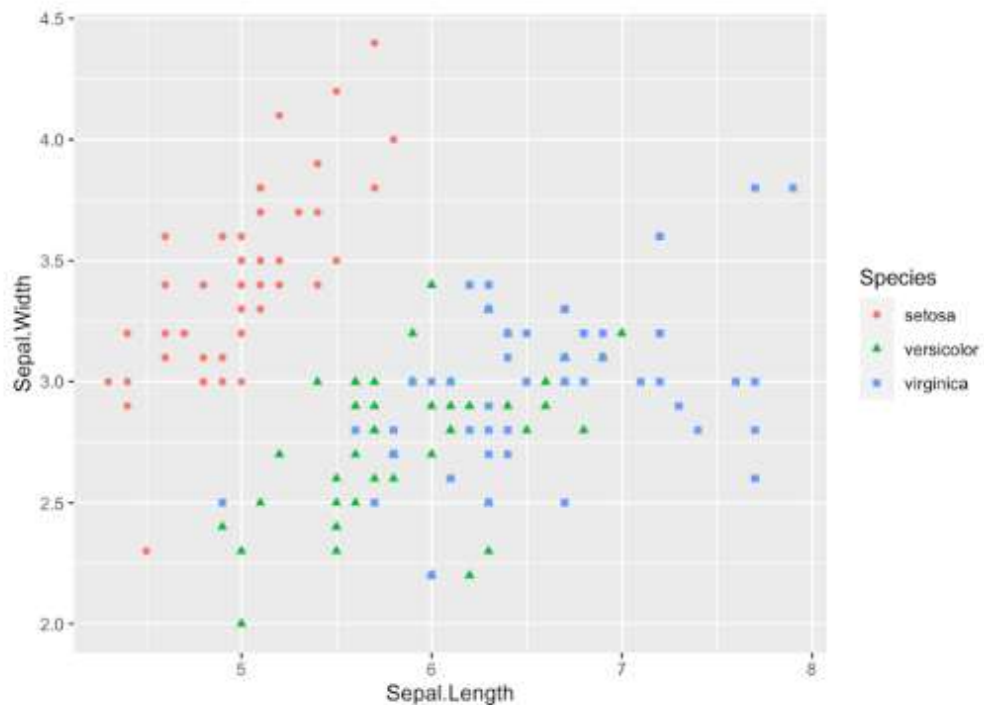
ggplot2



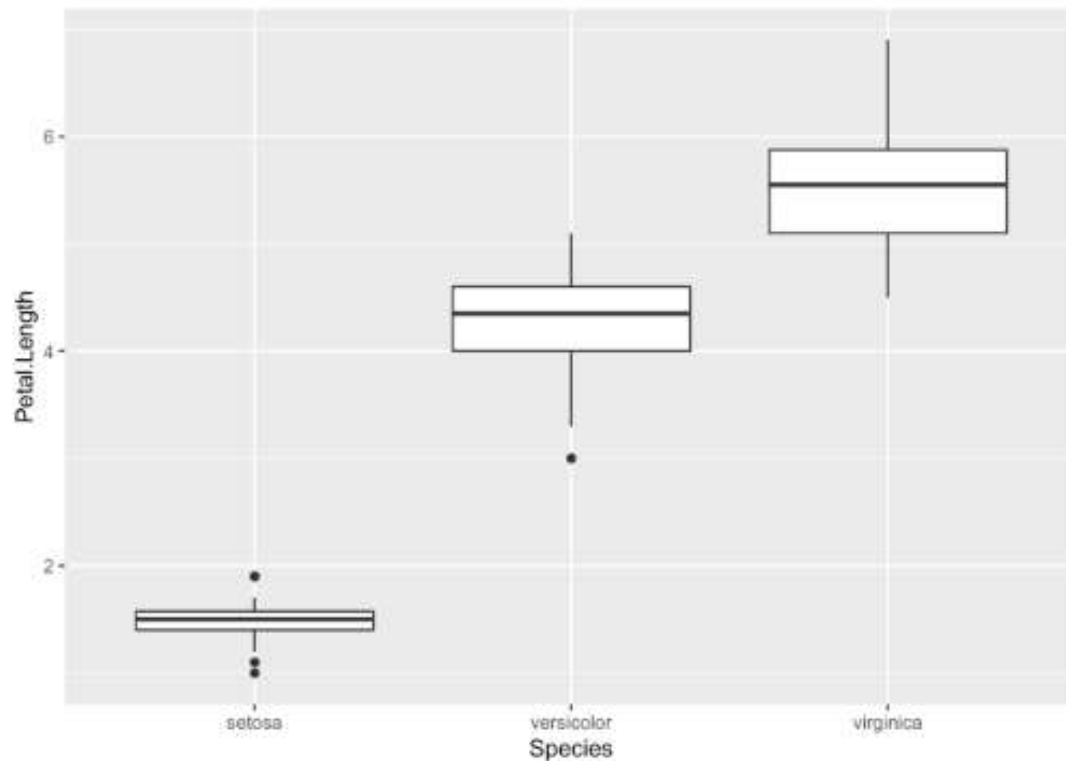
ggplot2 — R proqramlaşdırma dilində **müasir, estetik baxımdan zəngin və elmi vizualizasiyalar** yaratmaq üçün ən çox istifadə olunan paketlərdən biridir.

- **qplot()** – Sadə, ilkin və sürətli vizualizasiya üçün əladır.
- **ggplot()** – Daha professional, kompleks və fərdiləşdirilmiş qrafiklər üçün güclü alətdir.

```
library(ggplot2)
attach(iris)
qplot(Sepal.Length, Sepal.Width, color = Species, shape = Species)
```



```
qplot(Species, Petal.Length, geom = "boxplot")
```



qplot() funksiyası — Sürətli və sadə qrafik yaratmaq üçün

qplot() funksiyası ggplot2 paketinin bir hissəsidir və adı "quick plot" (sürətli qrafik) sözlərinin qısaltmasıdır.

Bu funksiya **ən sadə şəkildə və sürətli formada qrafik yaratmaq** üçün istifadə olunur.

Əgər əvvəlcə **əsas vizualı tez yaratmaq**, sonra isə üzərində dəyişiklik etmək istəyirsinizsə, qplot() idealdır.

❖ qplot() – “Quick Plot” (Sürətli Qrafik)

- **Sadə və tez bir şəkildə qrafik çəkmək** üçün hazırlanmışdır.
- Sintaksisi plot() funksiyasına oxşardır, ona görə də yeni başlayanlar üçün daha rahatdır.
- Daha az kodla tez nəticə verir.
- Çox qatlı (layered) və kompleks qrafiklər yaratmaqda **məhdudiyyətləri var**.

❖ ggplot() – Qrafiklərin strukturlaşdırılmış quruluşu

- **Qrafikləri mərhələ-mərhələ qurmağa** imkan verir.
- Daha **modul, çoxqatlı, və çevik qrafiklər** yaratmaq üçün idealdır.
- Daha çox **nəzarət və fərdiləşdirmə imkanı** verir.
- Sintaksis bir qədər mürəkkəbdir, amma öyrənildikdən sonra daha güclüdür.

ggplot(), ggplot2 paketində ən əsas və güclü funksiyadır.

Bu funksiya vasitəsilə **məlumatları mərhələli şəkildə vizuallaşdırmaq, fərqli qatlar (layers) əlavə etmək və çox yönlü qrafiklər** qurmaq mümkündür.

ggplot() funksiyası qrafik qurmaq üçün sadəcə bir başlanğıc nöqtəsidir — yəni, onu istifadə edərkən vizualın hansı dəyişənlərə, necə və nə cür göstəriləcəyinə dair məlumatları bir neçə strukturla təyin etmək lazımdır.

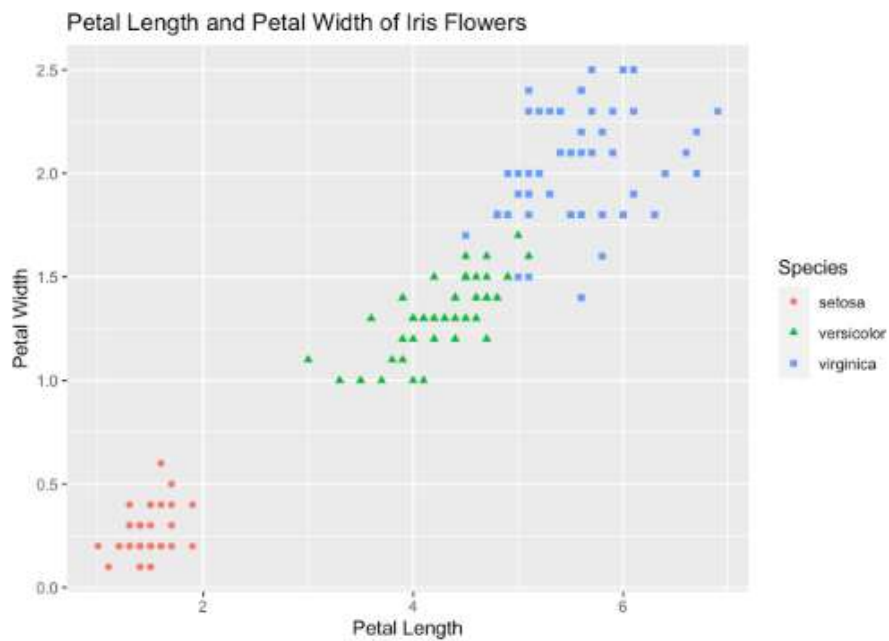
❖ **ggplot() funksiyasının əsas komponentləri**

1. Estetik ayarlar – aes() funksiyası

aes() — qrafikdə **nəyin hansı dəyişənə uyğun göstəriləcəyini** müəyyən edir.

Burada əsasən **x və y oxları**, həmçinin **rəng, forma, ölçü** kimi vizual xüsusiyyətlər təyin olunur.

```
p <- ggplot(iris) +  
  geom_point(aes(x = Petal.Length, y = Petal.Width, shape = Species, color = Species)) +  
  xlab("Petal Length") + # Add axis Labels  
  ylab("Petal Width") +  
  ggtitle("Petal Length and Petal Width of Iris Flowers")  
  
print(p)
```



2. Geometrik obyektlər (qrafik tipləri) – geom_* funksiyaları

Estetik ayarlarla müəyyən etdiyimiz məlumatları **vizual olaraq göstərmək** üçün geom_* adlı funksiyalardan istifadə edirik.

Bu funksiyalar qrafikin **formasını** təyin edir – nöqtə, xətt, sütun, qutu və s.

Məşhur geom funksiyaları:

- geom_point() – nöqtə qrafiki (scatter plot)
 - geom_line() – xətt qrafiki (line plot)
 - geom_bar() – bar qrafiki
 - geom_boxplot() – qutu diaqramı
- ggplot() funksiyası ilə peşəkar və çox qatlı qrafiklər yaratmaq üçün:*
- aes()* ilə vizual dəyişənləri təyin edirik,
 - geom_**() funksiyaları ilə bu dəyişənləri qrafikdə göstəririk,
 - stat_**() funksiyaları ilə məlumatlara statistik baxış əlavə edirik.

3. Statistik təhlil və ümumiləşdirmə – stat_* funksiyaları

Bəzən məlumatları **ümumiləşdirmək**, **trendləri göstərmək**, yaxud **statistik nəticələr əlavə etmək** istəyə bilərik. Bu zaman stat_* funksiyalarından istifadə olunur.

Məsələn:

- stat_summary() – ortalama, median və s. statistik göstəriciləri göstərir
- stat_smooth() – trendləri göstərən hamarlanmış xətlər əlavə edir
- stat_quantile() – kvartillərə görə bölmələr təqdim edir

❖ plotly nədir?

plotly, R dilində interaktiv və HTML formatında qrafiklər yaratmaq üçün istifadə edilən geniş imkanlara malik bir paketdir.

Bu paket vasitəsilə qurulan diaqramlar:

- vurma-böyütmə (zoom)
- məlumat baloncuqları (tooltip)
- sürüşdürülə bilən oxlar
- filtrləmə imkanları

kimisi funksiyaları dəstəkləyir.

Bütün bu xüsusiyyətlər, plotly-ni **vəb əsaslı hesabatlar, dashboardlar və təqdimatlar** üçün çox əlverişli edir.

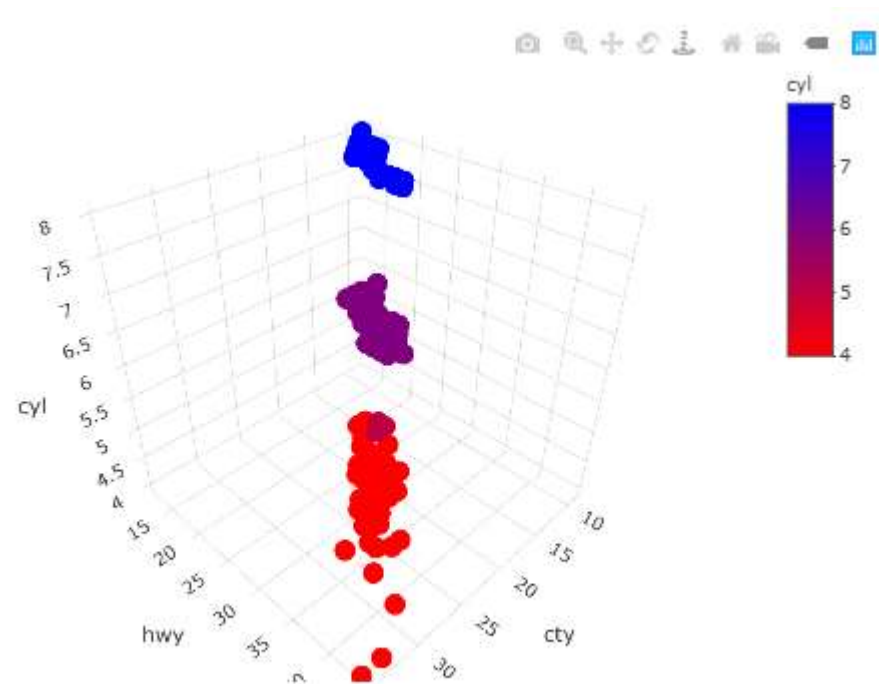
```
library(plotly)
head(mtcars)
```

```
##           mpg cyl  disp  hp  drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160  110 3.90 2.620 16.46  0  1   4    4
## Mazda RX4 Wag  21.0   6  160  110 3.90 2.875 17.02  0  1   4    4
## Datsun 710      22.8   4  108   93 3.85 2.320 18.61  1  1   4    1
## Hornet 4 Drive  21.4   6  258  110 3.08 3.215 19.44  1  0   3    1
## Hornet Sportabout 18.7   8  360  175 3.15 3.440 17.02  0  0   3    2
## Valiant         18.1   6  225  105 2.76 3.460 20.22  1  0   3    1
```

```
col1 <- colorRamp(c("red", "blue"))

p<-plot_ly(mpg, x = ~cty, y = ~hwy, z = ~cyl)

add_markers(p,color = ~cyl, colors = col1)
```



Harada istifadə olunur?

- İnteraktiv **təqdimatlar** (məsələn, R Markdown, Shiny tətbiqləri)
- Dashboard** qurmaq
- Onlayn hesabatlar** hazırlamaq
- Analitik **nəticələrin daha dinamik təqdimatı**

plotly R istifadəçilərinə **qrafiklərlə qarşılıqlı əlaqə qurmaq** imkanı verir. Əgər məqsəd yalnız şəkil olaraq qrafik hazırlamaqdan daha artıqdırsa – yəni, istifadəçinin məlumatla qarşılıqlı işləməsini təmin etməkdirsə – bu zaman plotly çox güclü bir alətə çevrilir.

Təşəkkürlər