Models (/github/ramizallahverdiyev/Models/tree/main)
/ Linear_Regression.ipynb (/github/ramizallahverdiyev/Models/tree/main/Linear_Regression.ipynb)

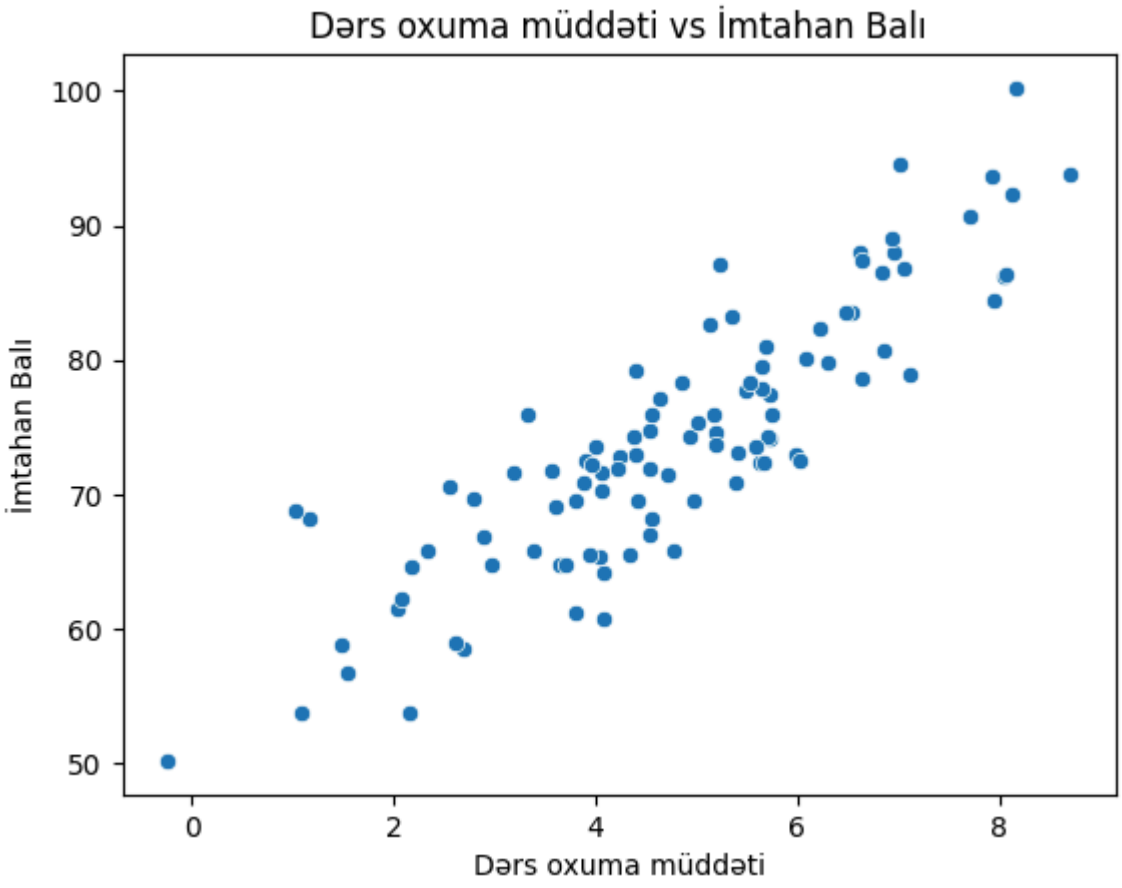
📘 1. ƏSAS ANLAYIŞLAR – XƏTTİ REQRESSİYAYA GİRİŞ

1.1. Asılı və Sərbəst Dəyişənlər

Xətti reqressiyanın əsas məqsədi, bir **asılı dəyişəni** (yəni nəticəni), bir və ya bir neçə **sərbəst dəyişən** (giriş, amil, səbəb) vasitəsilə izah etməkdir.

- **Asılı dəyişən (dependent variable)**: Təxmin edilən dəyişəndir. Adətən y ilə göstərilir.
- **Sərbəst dəyişənlər (independent variables)**: y-ə təsir edən dəyişənlərdir. $x_1, x_2, \ldots, x_n$ kimi ifadə edilir.

In [ ]:



📌 Nümunə: Bir tələbənin imtahan balını təxmin etmək istəyirik.

- Asılı dəyişən: İmtahan balı
- Sərbəst dəyişənlər: Dərs oxuma müddəti, yuxu rejimi, əvvəlki imtahan nəticələri

Xətti reqressiya bu dəyişənlər arasındakı əlaqəni modelləşdirərək y-nin dəyərini təxmin etməyə çalışır.

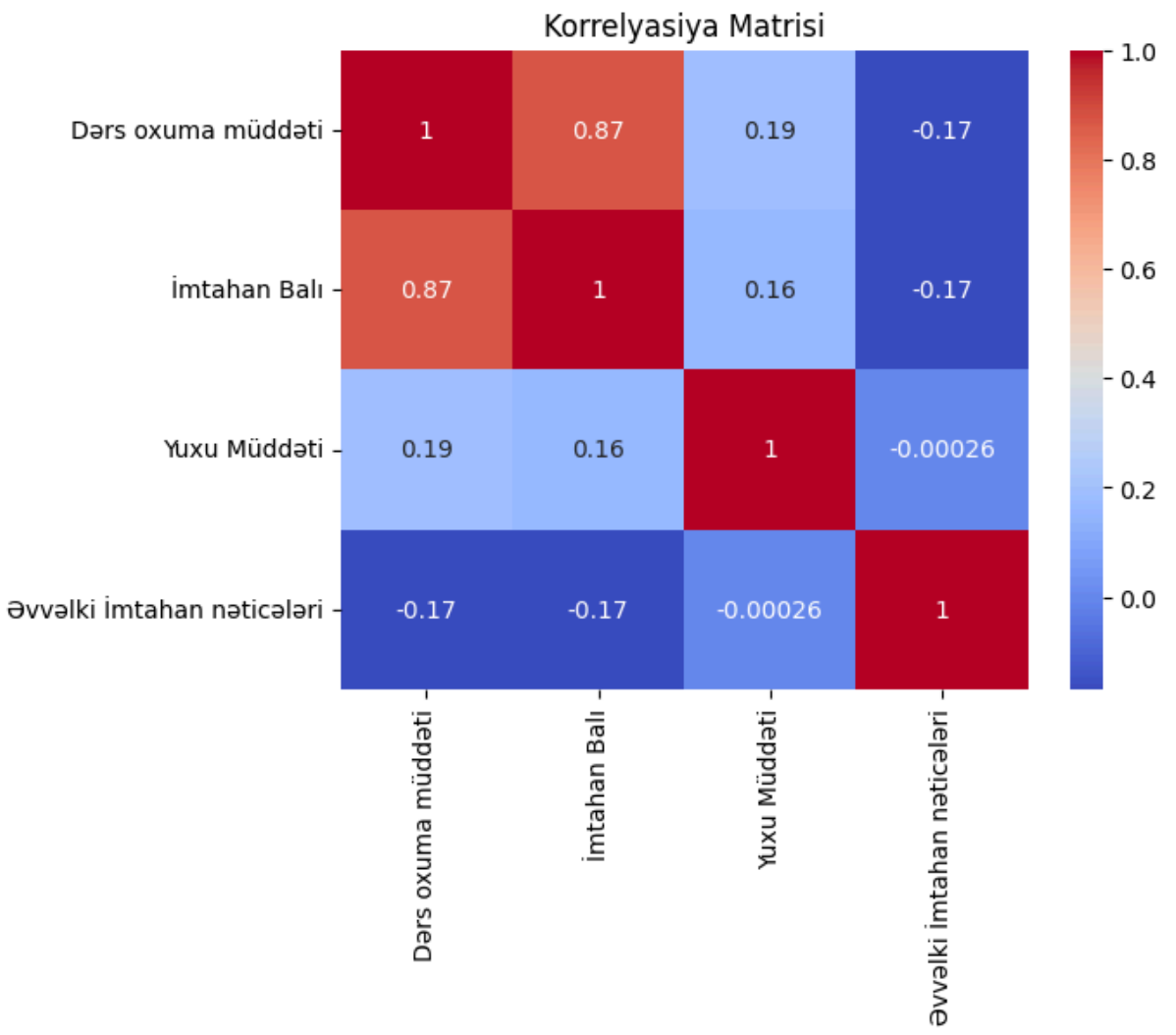
1.2. Reqressiya və Korrelyasiya Arasındakı Fərq

Bu iki anlayış tez-tez qarışdırılır, lakin fərqlidir:

- **Korrelyasiya**: İki dəyişənin birlikdə necə hərəkət etdiyini göstərir.
  - Məsələn, biri artdıqda digəri də artırmı?
  - Pearson korrelyasiya əmsalı -1 ilə $+1$ arasında bir dəyər alır.
  - Lakin korrelyasiya **səbəb-nəticə əlaqəsi** vermir.
- **Reqressiya**: Dəyişənlər arasında **səbəb-nəticə əlaqəsini modelləşdirmək** məqsədi daşıyır.
  - Reqressiya analizində bir dəyişənin (məsələn, dərs oxuma müddəti) başqa bir dəyişənə (məsələn, imtahan balı) necə təsir etdiyi təxmin edilir.
  - Reqressiya, dəyişənlər arasındakı əlaqənin istiqamətini və böyüklüyünü verir.

📌 Korrelyasiya varsa reqressiya aparıla bilər, amma korrelyasiyanın yüksək olması reqressiyanın doğru işləyəcəyi mənasını vermir.

In [ ]:



1.3. Sadə və Çoxsaylı Xətti Reqressiya

Xətti reqressiyanın iki əsas növü var:

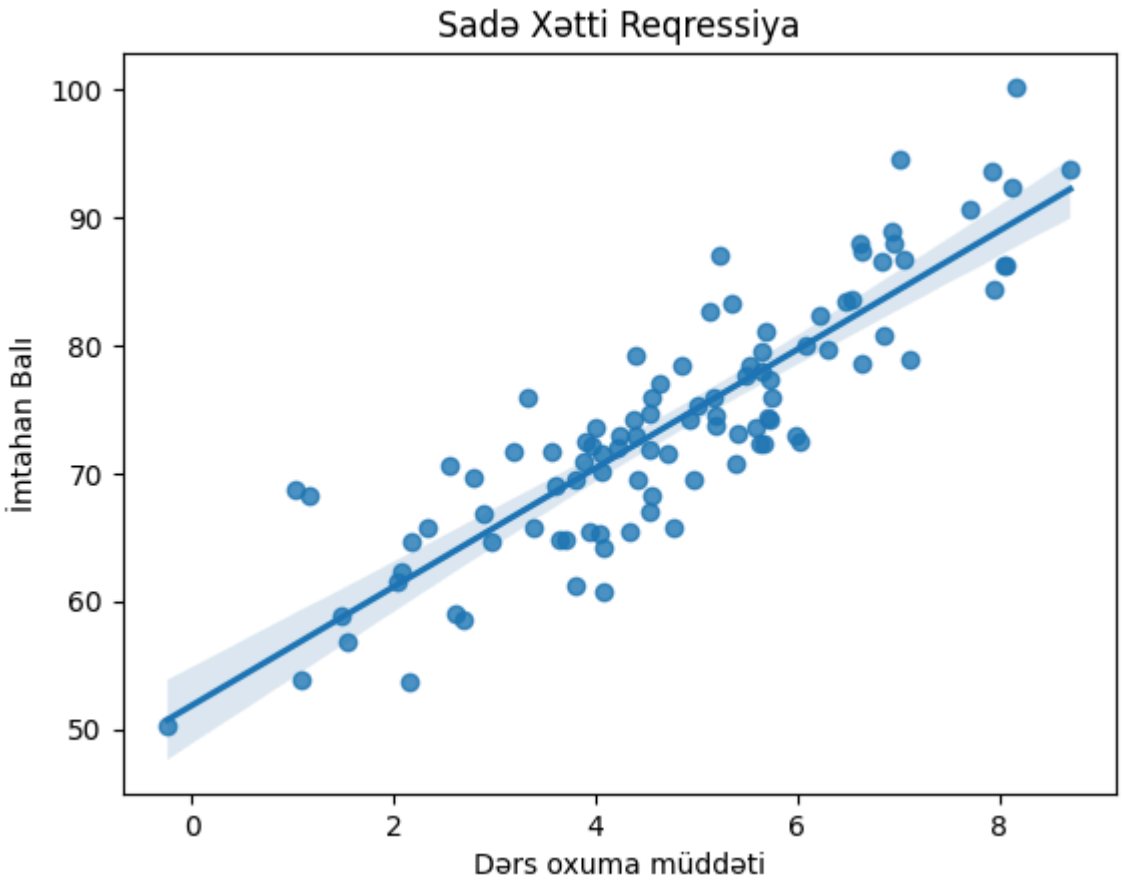
- **Sadə (Simple) Xətti Reqressiya**:\

Yalnız bir sərbəst dəyişən ehtiva edir.

$$y = \beta_0 + \beta_1 x + \epsilon$$

Burada β_0 sabit həddi (intercept), β_1 isə meyl əmsalıdır.

In [ ]:



- **Çoxsaylı (Multiple) Xətti Reqressiya**:\

Birdən çox sərbəst dəyişəndən istifadə edilir.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

Bu modeldə hər sərbəst dəyişənin y üzərindəki təsiri ayrı-ayrılıqda hesablanır.

📌 Real həyatdakı tətbiqlərin əksəriyyəti çoxsaylı reqressiyadır, çünki hadisələr adətən birdən çox amildən asılıdır.

In [ ]:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           İmtahan Balı   R-squared:                       0.762
Model:                            OLS   Adj. R-squared:                  0.754
Method:                 Least Squares   F-statistic:                     102.2
Date:                Sat, 02 Aug 2025   Prob (F-statistic):           9.03e-30
Time:                        08:45:09   Log-Likelihood:                -296.56
No. Observations:                 100   AIC:                             601.1
Df Residuals:                      96   BIC:                             611.5
Df Model:                           3
Covariance Type:            nonrobust
===============================================================================
===========
                                 coef    std err          t      P>|t|      [0.025
0.975]
-------------------------------------------------------------------------------
-----------
const                         53.5671      4.774     11.220      0.000      44.090
63.044
Dərs oxuma müddəti             4.6280      0.274     16.872      0.000       4.084
5.173
Yuxu Müddəti                  -0.0422      0.453     -0.093      0.926      -0.941
0.857
Əvvəlki İmtahan nəticələri    -0.0225      0.055     -0.407      0.685      -0.132
0.087
==============================================================================
Omnibus:                        1.802   Durbin-Watson:                   2.174
Prob(Omnibus):                  0.406   Jarque-Bera (JB):                1.775
Skew:                           0.314   Prob(JB):                        0.412
Kurtosis:                       2.821   Cond. No.                         622.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
ecified.
```

1.4. Xəttilik Nə Deməkdir?

Xətti reqressiyadakı “xətti” ifadəsi çox vaxt yanlış başa düşülür.

- Xəttilik, **dəyişənlərlə deyil, əmsallarla olan xətti əlaqəni** ifadə edir.
- Modelin xətti olması o deməkdir ki:\

Əmsalların (β-ların) modelə xətti olaraq daxil olması deməkdir.

📌 Bu səbəbdən aşağıdakı nümunələrin hamısı xətti reqressiyaya aiddir:

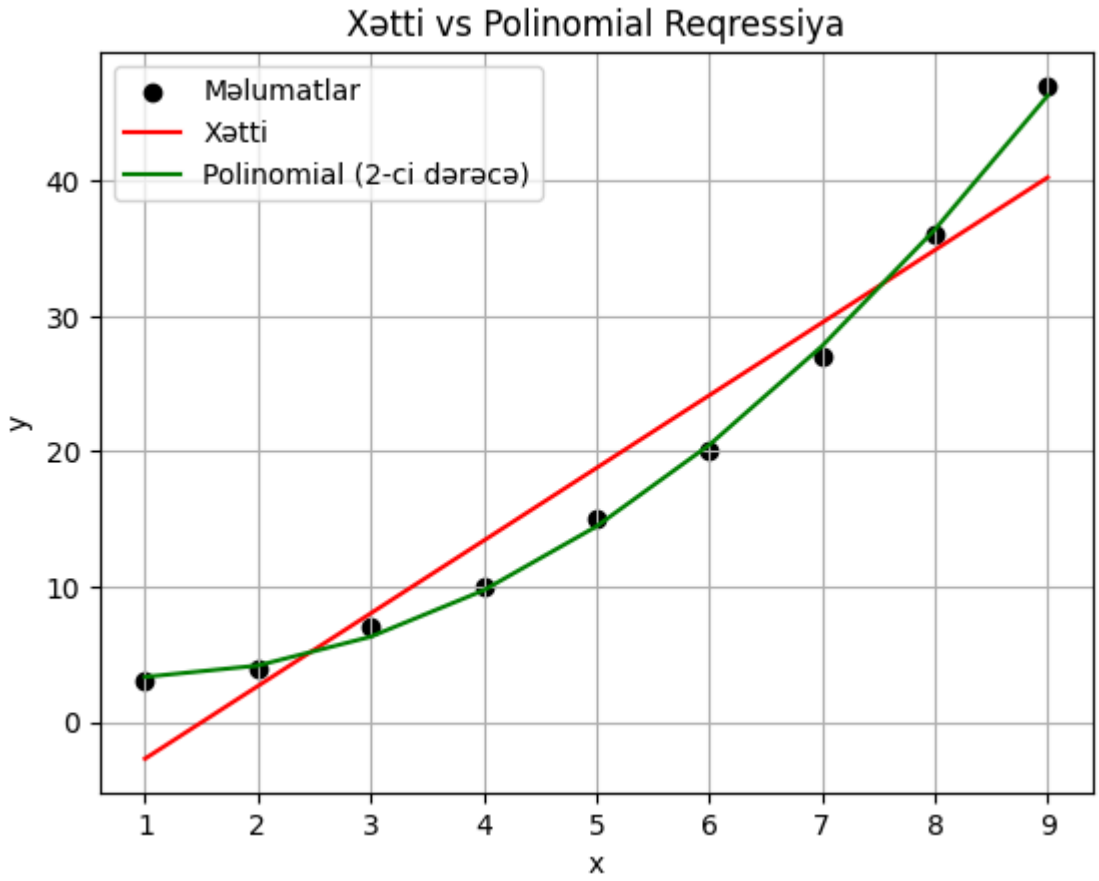
- $y = \beta_0 + \beta_1 x$
- $y = \beta_0 + \beta_1 x + \beta_2 x^2$
- $y = \beta_0 + \beta_1 \log(x)$

Hamısı xəttidir, çünki əmsallar (β) birbaşa və xətti şəkildə yer alır.

Lakin aşağıdakı kimi bir tənlik **xətti deyil**:

- $y = \beta_0 + \beta_1^2 x$
- $y = e^{\beta_1 x}$

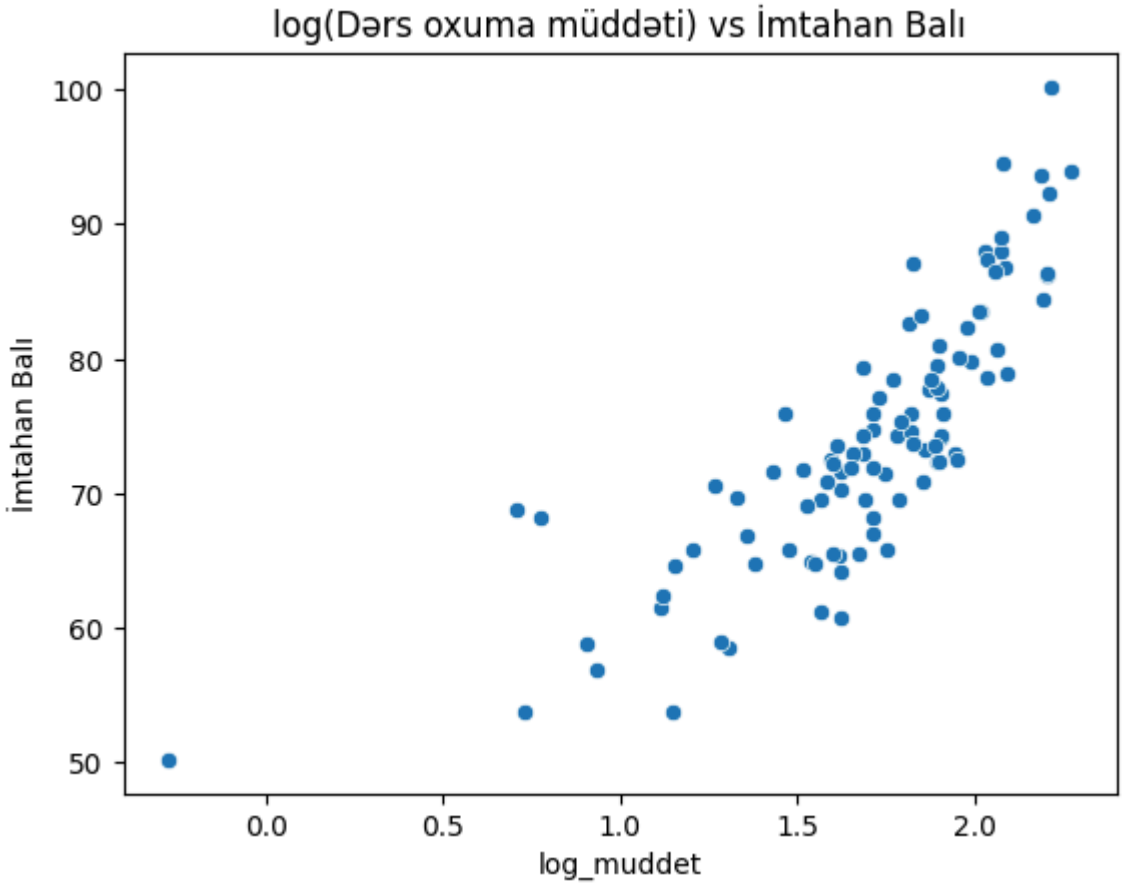
In [ ]:



📝 Ümumi Qeydlər:

- Sərbəst dəyişənlər modelə təsirlərini **xətti olaraq əlavə etmək** məcburiyyətindədir. Lakin dəyişənin özü transformasiyaya məruz qala bilər (məsələn x^2, $\log(x)$, $\sqrt{x}$, və s.).
- Xətti reqressiyanın gücü, həm sadə, həm də izahlı olmasından irəli gəlir.
- Sadə modellərlə başlayıb, lazım olduqda mürəkkəbliyi addım-addım artırmaq doğru yanaşmadır.

In [ ]:



📘 2. MODELİN RİYAZİ ƏSASLARI

Xətti reqressiyanın arxasındakı əsas riyazi struktur, modelin necə qurulduğunu, necə həll edildiyini və parametrlərin nə anlama gəldiyini anlamağımızı təmin edir.

2.1. Reqressiya Tənliyi

Xətti reqressiya modeli ümumiyyətlə belə yazılır:

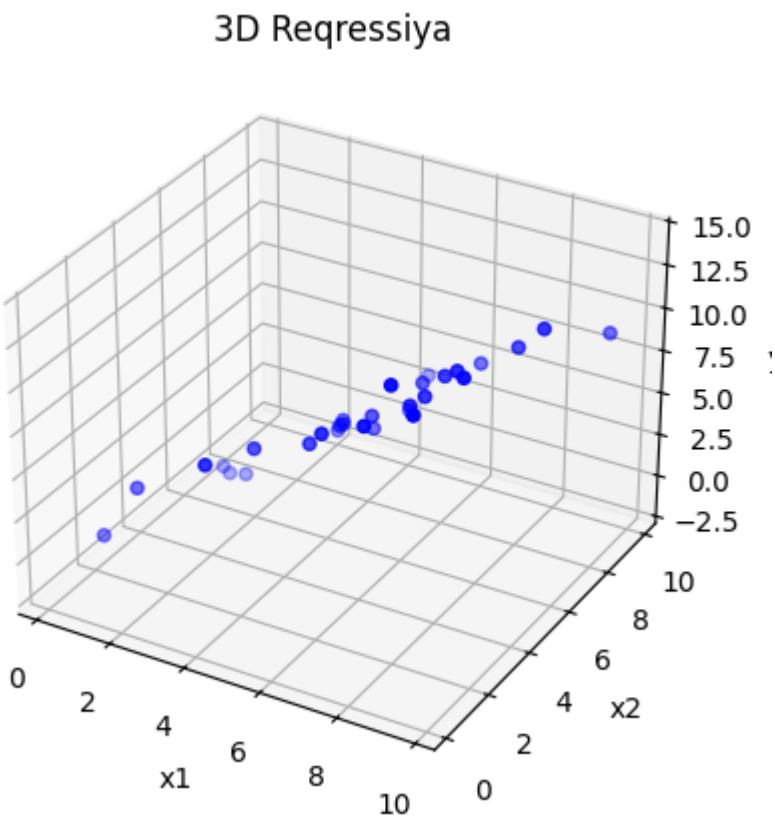
$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon$$

Burada:

- y: Asılı dəyişən (təxmin edilən)
- $x_1, x_2, \ldots, x_n$: Sərbəst dəyişənlər (girdilər)
- β_0: Sabit həddi (intercept)
- $\beta_1, \beta_2, \ldots, \beta_n$: Reqressiya əmsalları (hər bir dəyişənin təsiri)
- ϵ: Xəta həddi (modelin izah edə bilmədiyi fərqlər)

Modelin məqsədi, β əmsallarını elə seçməkdir ki, model təxminləri real verilənlərə ən yaxın nəticəni versin.

In [ ]:



2.2. Əmsalların Mənası

Hər β_i əmsalı, digər dəyişənlər sabit qaldıqda x_i-dəki bir vahidlik artımın y üzərindəki təsirini ifadə edir.

- Müsbət β: x artdıqca y artar
- Mənfi β: x artdıqca y azalar
- $\beta = 0$: x-in y üzərində əhəmiyyətli bir təsiri yoxdur

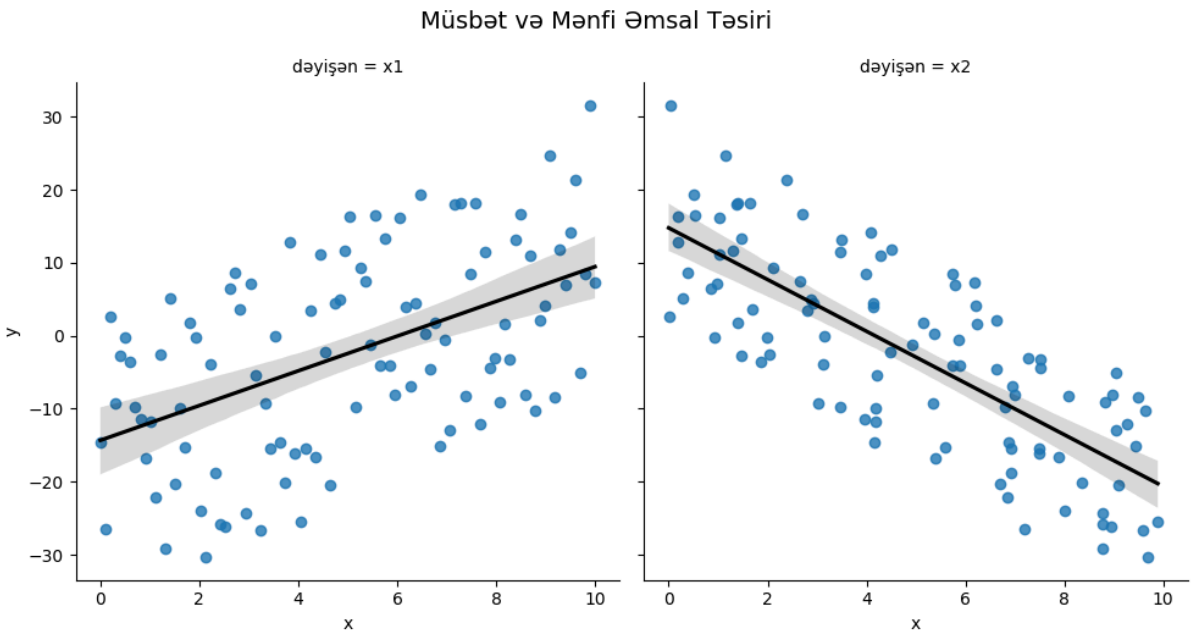
Nümunə tənlik:

$$y = 2 + 3x_1 - 4x_2$$

Şərh:

- x_1 hər 1 vahid artdıqda y orta hesabla 3 vahid artar
- x_2 hər 1 vahid artdıqda y orta hesabla 4 vahid azalar

In [ ]:



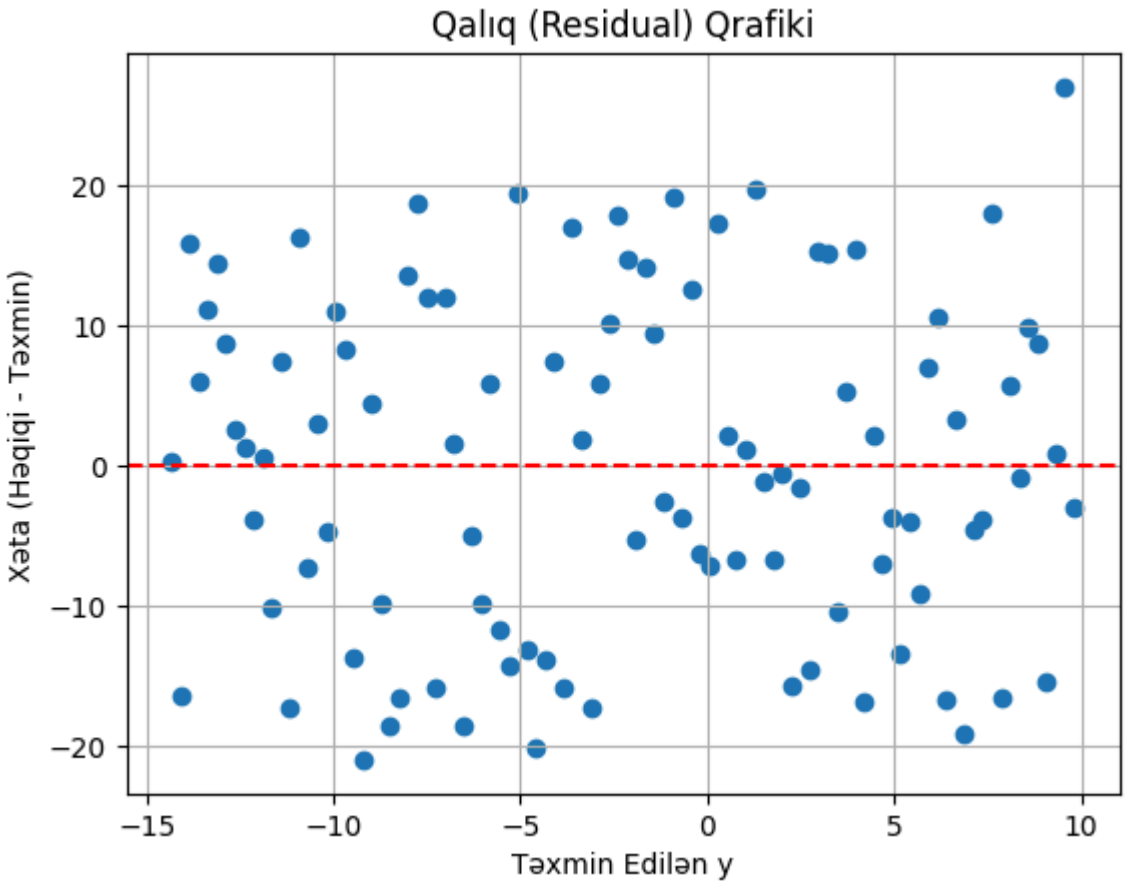
2.3. Xəta Həddi (ϵ)

Real verilənlər mükəmməl bir şəkildə model tərəfindən təxmin edilə bilməz. Bu fərqlərə xəta (residual) deyilir.

- ϵ: Müşahidə edilən və təxmin edilən y arasındakı fərq
- Xətaların ortalaması sıfır qəbul edilir
- Xətaların normal paylanması gözlənilir

Bu hədd, modelin xarici amilləri nəzərə ala bilməməsini təmsil edir.

In [ ]:



2.4. Ən Kiçik Kvadratlar Üsulu (Ordinary Least Squares - OLS)

Modelin parametrləri adətən Ən Kiçik Kvadratlar (OLS) üsulu ilə hesablanır.

Məqsəd: Xəta kvadratlarının cəmini minimuma endirmək:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y} * i)^2 = \sum *i = 1^n (y_i - \beta_0 - \beta_1 x_{1i} - \cdots - \beta_n x_{ni})^2$$

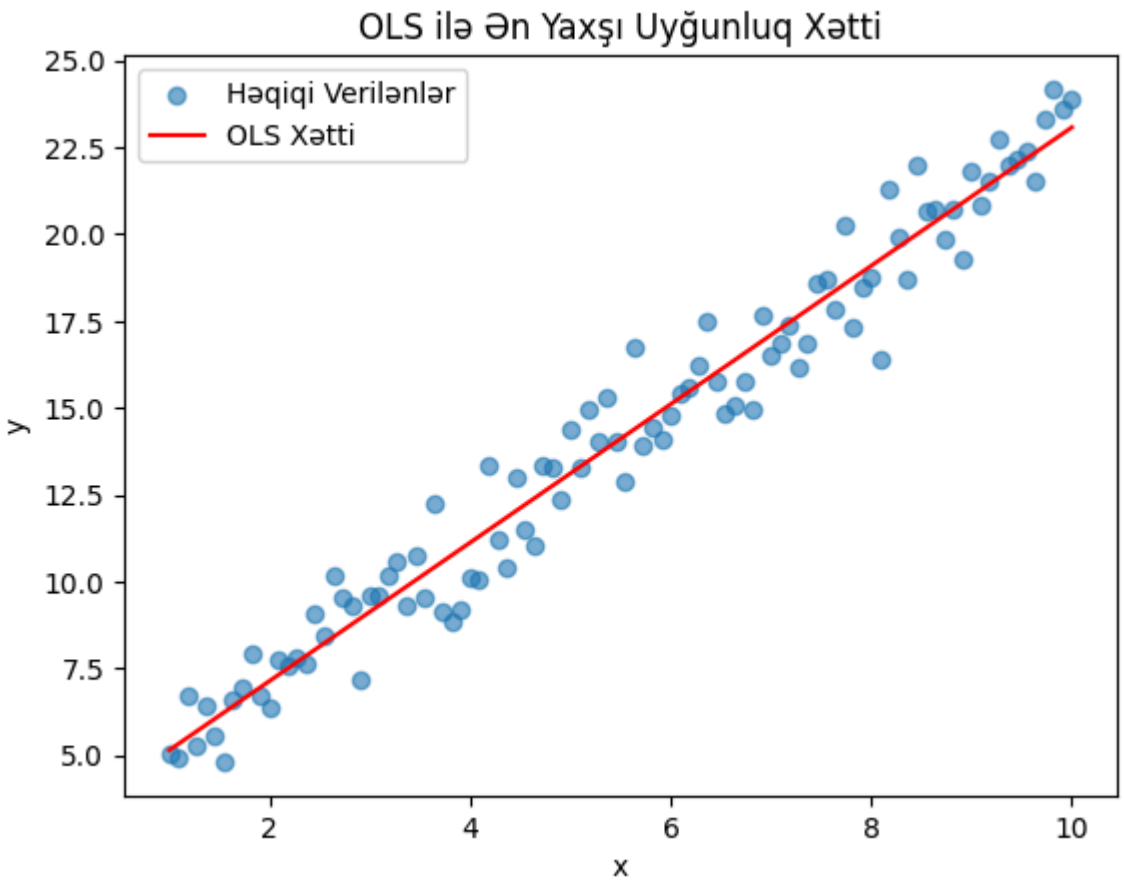
Bu problemin analitik həlli aşağıdakı kimidir:

$$\boldsymbol{\beta} = (X^T X)^{-1} X^T y$$

- X: Girdi (sərbəst dəyişən) matrisi
- y: Müşahidə edilmiş çıxış vektoru
- β: Əmsallar vektoru

Bu həll, matris cəbri ilə model parametrlərinin ən uyğun təxminini verir.

In [ ]:



📌 Xülasə

- Xətti reqressiya, dəyişənlər arasındakı xətti əlaqəni modelləşdirmək üçün qurulur.
- Modeldeki β əmsalları, hər dəyişənin y üzərindəki töhfəsini göstərir.
- Xəta həddi, modelin təxmin edə bilmədiyi sapmaları təmsil edir.
- Ən kiçik kvadratlar üsulu, parametrləri təxmin etmək üçün istifadə olunur və qapalı formada həll edilə bilər.

📘 3. XƏTTİ REQRESSİYANIN FƏRZİYYƏLƏRİ

Xətti reqressiya güclü bir modeldir, ancaq bəzi **nəzəri fərziyyələrin** təmin edilməsi lazımdır. Bu fərziyyələr pozularsa, modelin təxmin gücü azalar, nəticələr yanıltıcı olar və statistik testlərin mənası qalmaz.

Xətti reqressiya 5 əsas fərziyyəyə əsaslanır:

3.1. Xəttilik Fərziyyəsi (Linearity)

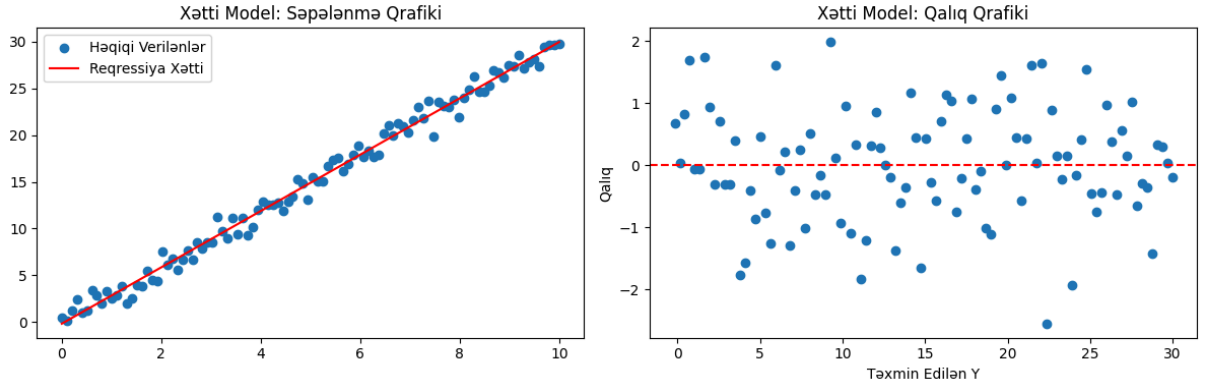
Model, asılı dəyişənlə sərbəst dəyişənlər arasında xətti bir əlaqə olduğunu fərz edir.

- Bu əlaqə, düsturun **əmsallarla xətti olması** deməkdir.
- Ancaq bu, dəyişənlərin düz olmasını tələb etmir. Məsələn x^2, $\log(x)$ kimi çevrilmələr aparıldıqda da model xətti qala bilər (çünki əmsallar hələ də xətti daxil olur).

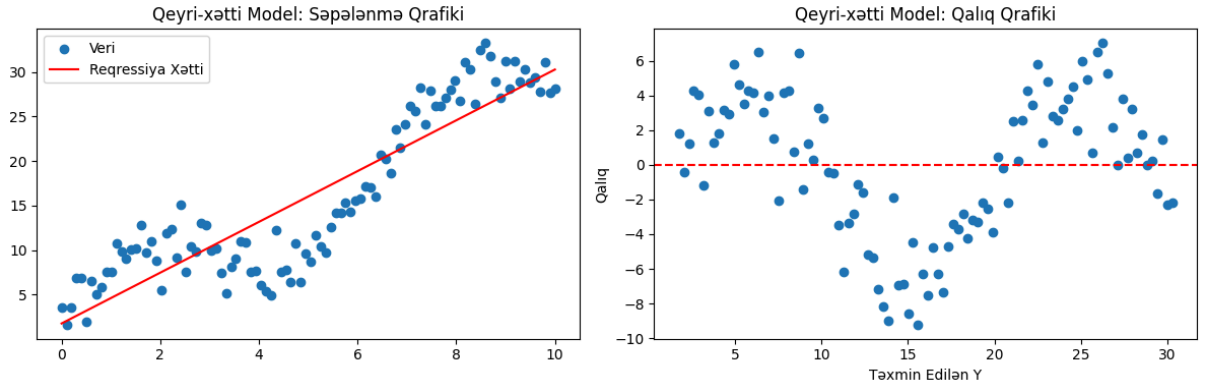
📌 **Necə yoxlanılır:**

- Səpələnmə qrafikləri (scatter plots) və ya qalıq qrafikləri (residual plots) ilə. Xətalar təsadüfi paylanmalı, müəyyən bir naxış görünməməlidir.

In [ ]:



In [ ]:



3.2. Müstəqillik Fərziyyəsi (Independence)

Modeldəki müşahidələrin bir-birindən asılı olmadığı fərz edilir.

- Bu, xüsusilə zaman seriyası kimi məlumatlarda kritikdir.\

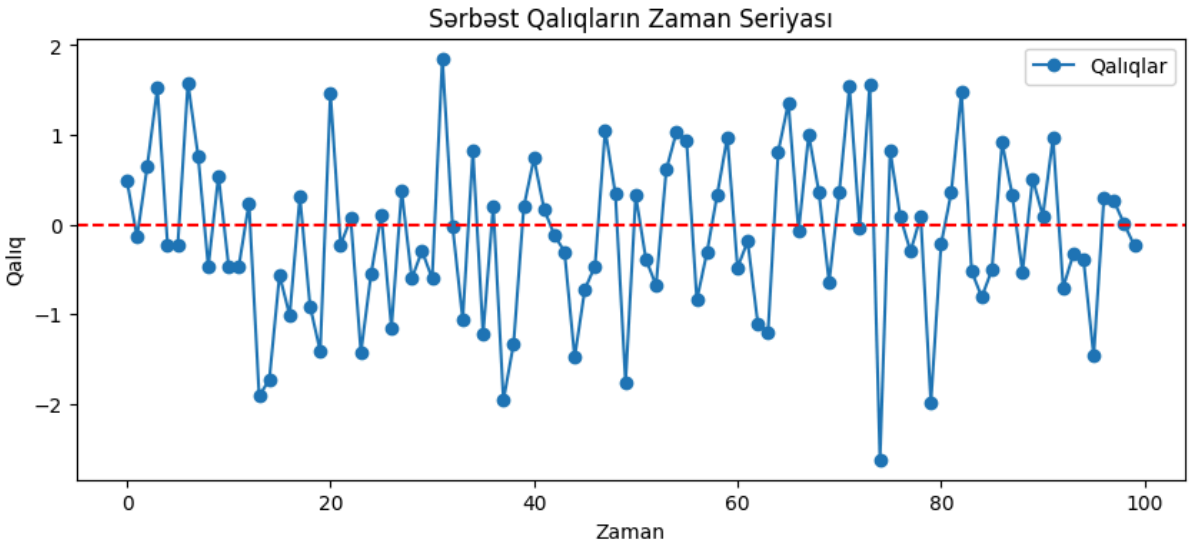
Məsələn: Bir gündə satılan məhsul sayı, əvvəlki günlə əlaqəlidirsə, sərbəstlik pozulmuşdur.

📌 **Necə yoxlanılır:**

- Durbin-Watson testi kimi avtokorrelyasiya testləri ilə.
- Zaman seriyası varsa: Qalıqların ardıcıl sərbəstliyi yoxlanılmalıdır.

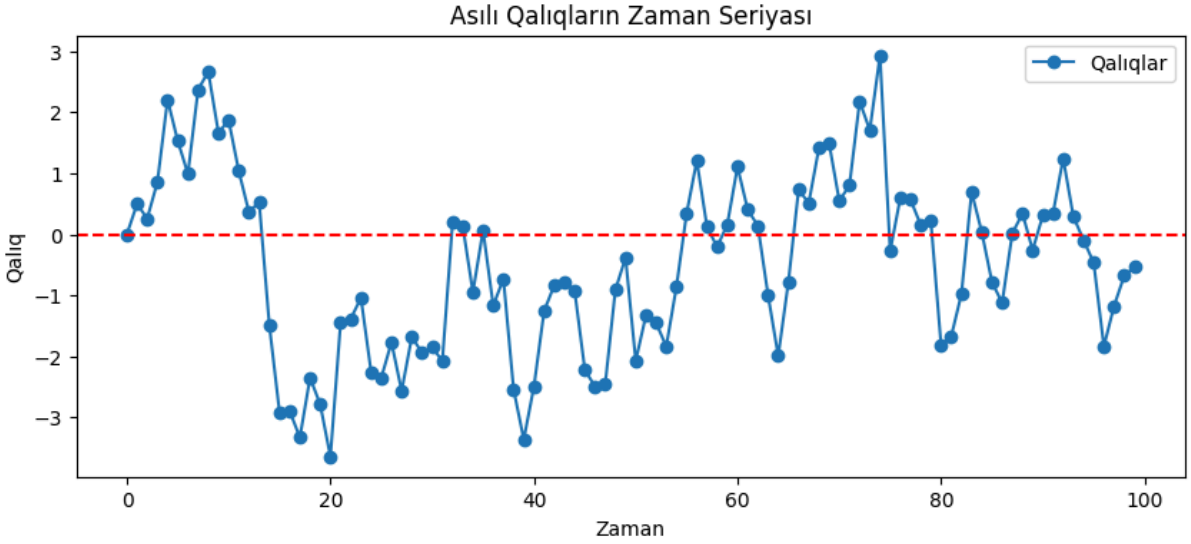
In [ ]:

```
Sərbəst Durbin-Watson: 2.01
```



In [ ]:

```
Asılı Durbin-Watson: 0.39
```



3.3. Homoskedastiklik (Sabit Dispersiya)

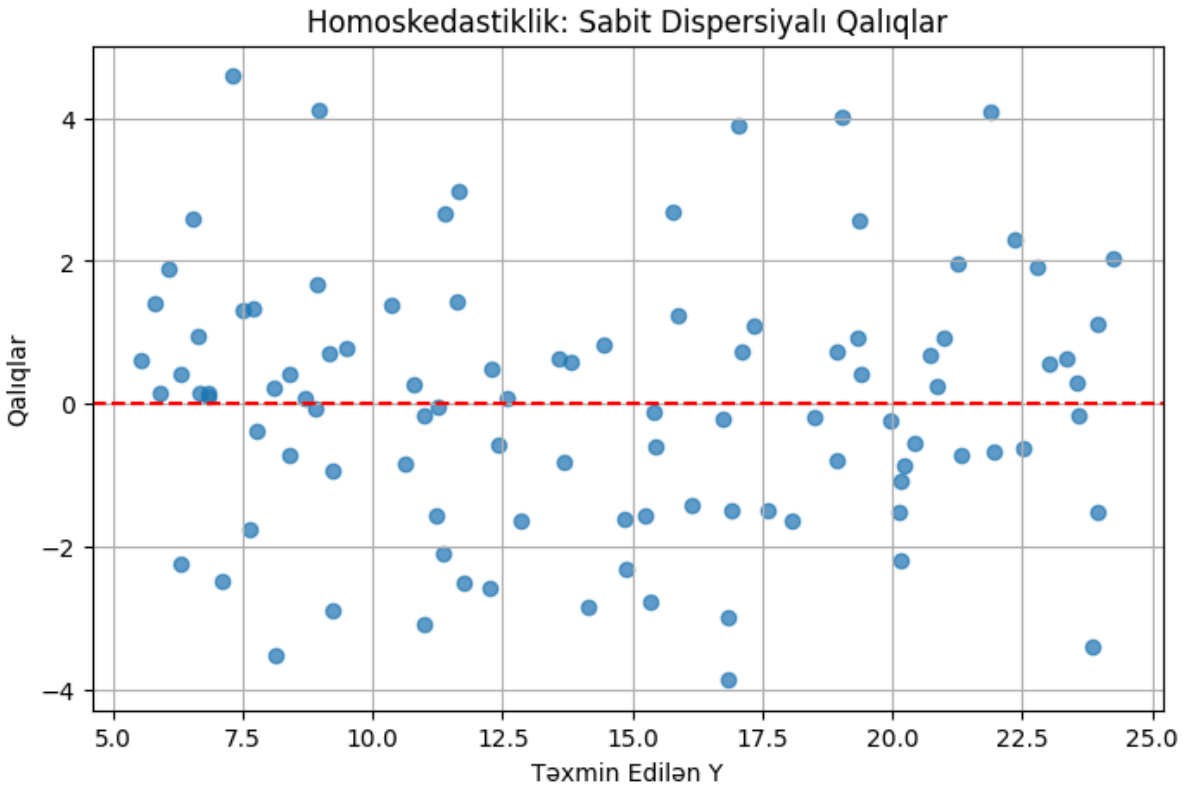
Modeldəki xəta hədlərinin dispersiyası sabit olmalıdır.

- Buna “homoskedastiklik” deyilir.
- Əgər bəzi müşahidələr daha böyük/sistemli xətalar ehtiva edirsə, buna “heteroskedastiklik” deyilir və model etibarsız olur.

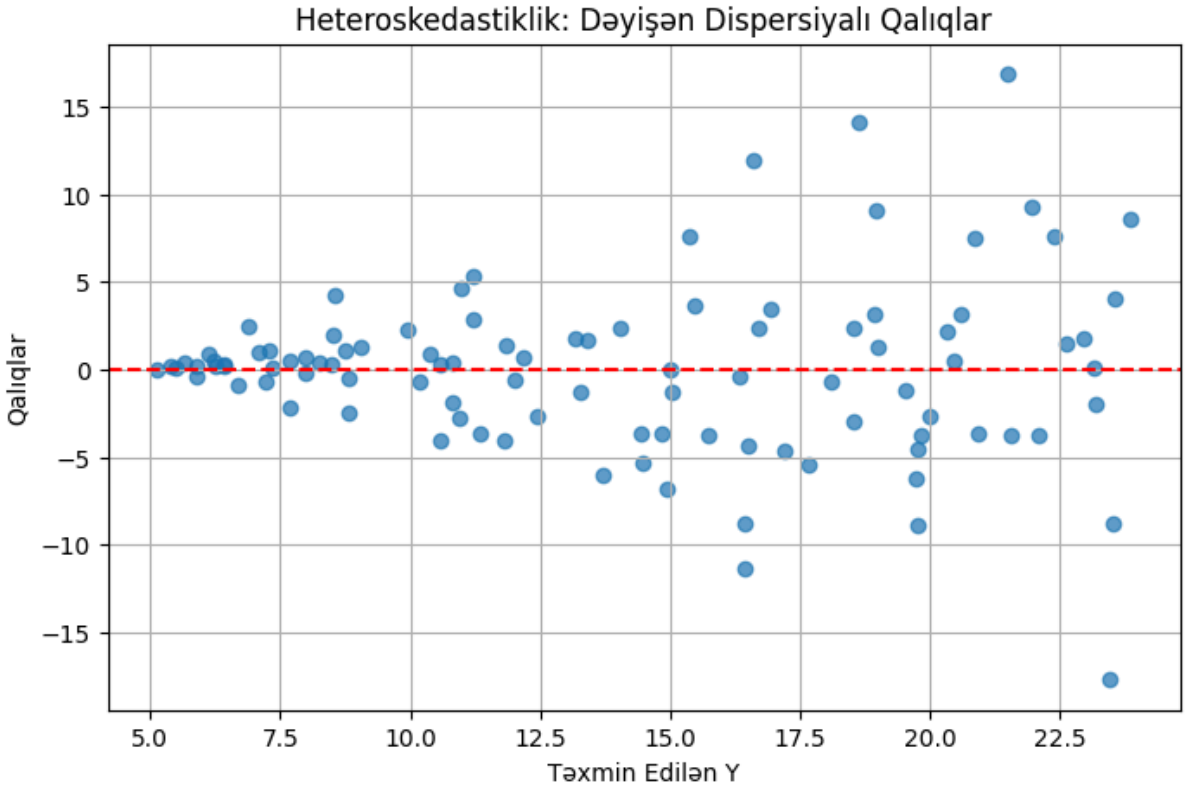
📌 **Necə yoxlanılır:**

- Qalıq vs. təxmin qrafiki (Residual vs. predicted plot): Əgər xətalar artan/azalan şəklində açılırsa, heteroskedastiklik var deməkdir.
- Breusch-Pagan testi, White testi kimi statistik testlər də tətbiq edilə bilər.

In [ ]:


Homoskedastiklik: Sabit Dispersiyalı Qalıqlar
Qalıqlar
Təxmin Edilən Y
4
2
0
−2
−4
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
25.0

In [ ]:


Heteroskedastiklik: Dəyişən Dispersiyalı Qalıqlar
Qalıqlar
Təxmin Edilən Y
15
10
5
0
−5
−10
−15
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5

In [ ]:

```
=== Breusch-Pagan Testi Nəticələri ===
Lagrange multiplier statistic: 16.8008
p-value: 0.0000
f-value: 19.7896
f p-value: 0.0000


=== White Testi Nəticələri ===
Test statistic: 17.4680
p-value: 0.0002
F statistic: 10.2651
F p-value: 0.0001
```

3.4. Normal Paylanma Fərziyyəsi (Normality of Errors)

Xəta hədləri normal paylanmaya sahib olmalıdır. Bu xüsusilə:

- p-dəyəri hesablamalarında
- Əmsallar üçün etibar aralıqları yaradarkən\

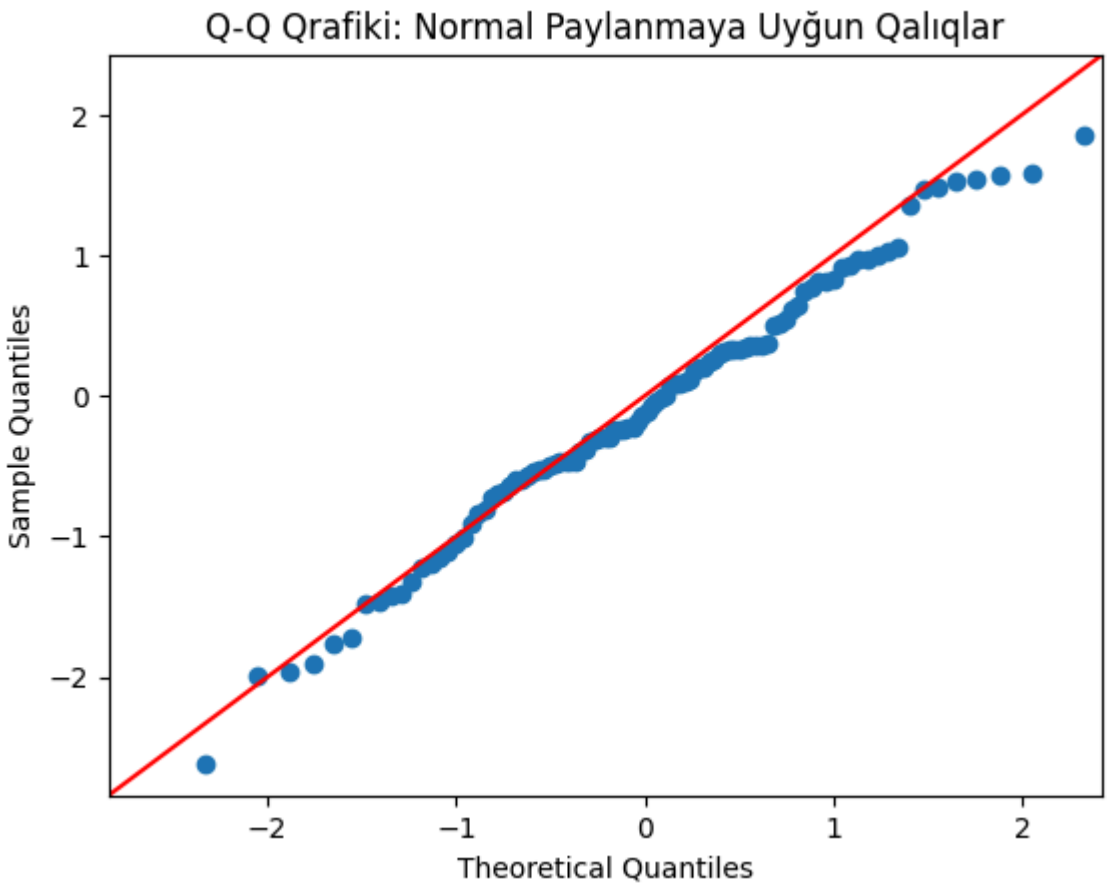
vacibdir.

📌 **Necə yoxlanılır:**

- Q-Q (quantile-quantile) qrafiki: Xətaların normal paylanmaya nə qədər uyğun olduğunu vizuallaşdırır.
- Kolmogorov-Smirnov və ya Shapiro-Wilk kimi normal paylanma testləri istifadə edilə bilər.

🧠 **Qeyd:** Bu fərziyyə təxmin doğruluğu üçün deyil, **statistik nəticə** üçün vacibdir. Təxmin etmək üçün çox kritik deyil.

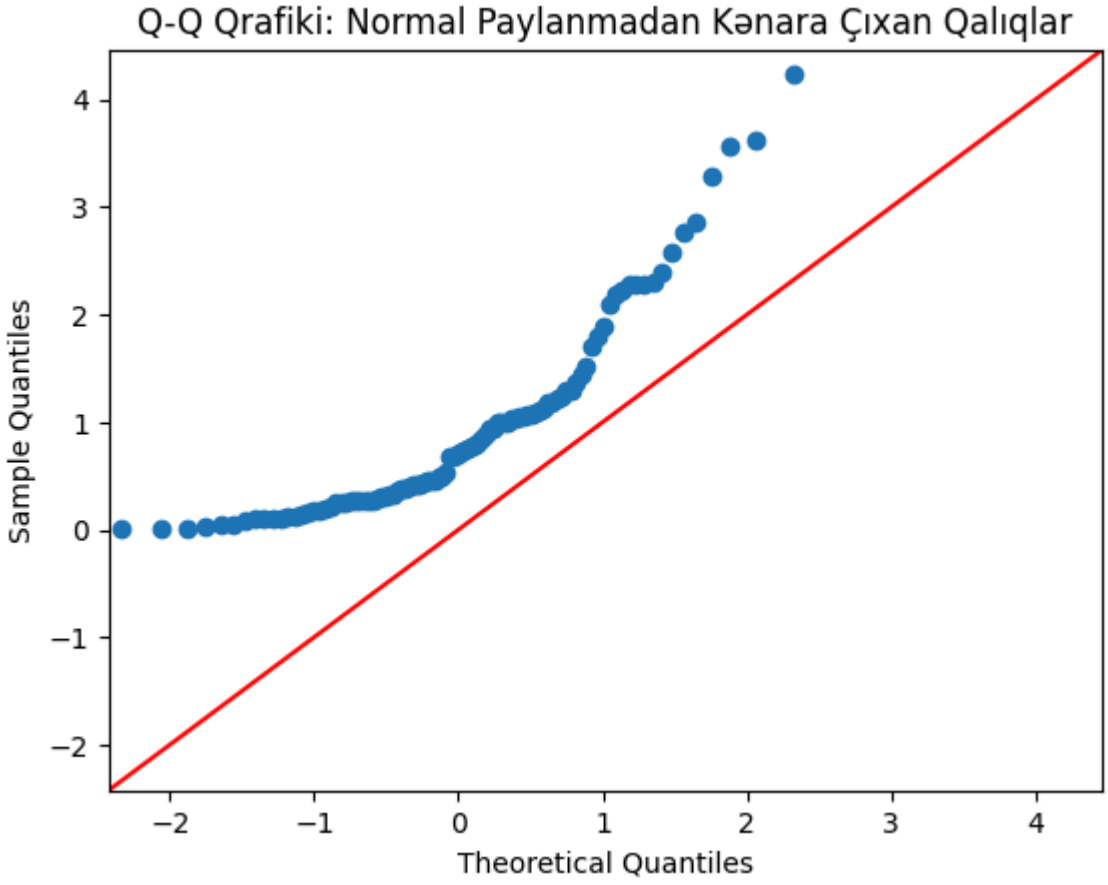
In [ ]:



In [ ]:

```
Shapiro-Wilk Testi Statistikası: 0.9899, p-dəyəri: 0.6552
Kolmogorov-Smirnov Testi Statistikası: 0.0508, p-dəyəri: 0.9467
```

In [ ]:



In [ ]:

```
Shapiro-Wilk Testi (əyrilik): Statistika: 0.8406, p-dəyəri: 0.0000
Kolmogorov-Smirnov Testi (əyrilik): Statistika: 0.1497, p-dəyəri: 0.0202
```

Shapiro-Wilk Testi Hipotezləri

Sıfır Hipotezi (H_0): Verilənlər dəsti normal paylanmaya uyğundur. (Qalıqlar normal paylanır.)

Alternativ Hipotez (H_1): Verilənlər dəsti normal paylanmaya uyğun deyil. (Qalıqlar normal paylanmır.)

Kolmogorov-Smirnov Testi Hipotezləri

Sıfır Hipotezi (H_0): Verilənlər dəsti verilən paylanmaya (məsələn, standart normal) uyğundur.

Alternativ Hipotez (H_1): Verilənlər dəsti verilən paylanmaya uyğun deyil.

3.5. Çoxlu Xətti Asılılığın Olmaması (No Multicollinearity)

Sərbəst dəyişənlər bir-biri ilə yüksək dərəcədə əlaqədə olmamalıdır.

- Əgər dəyişənlər bir-birinə çox bənzəyirsə, model bu dəyişənlərin təsirini ayırd edə bilməz.
- Bu vəziyyət əmsal təxminlərini qeyri-sabit edir.

📌 **Necə yoxlanılır:**

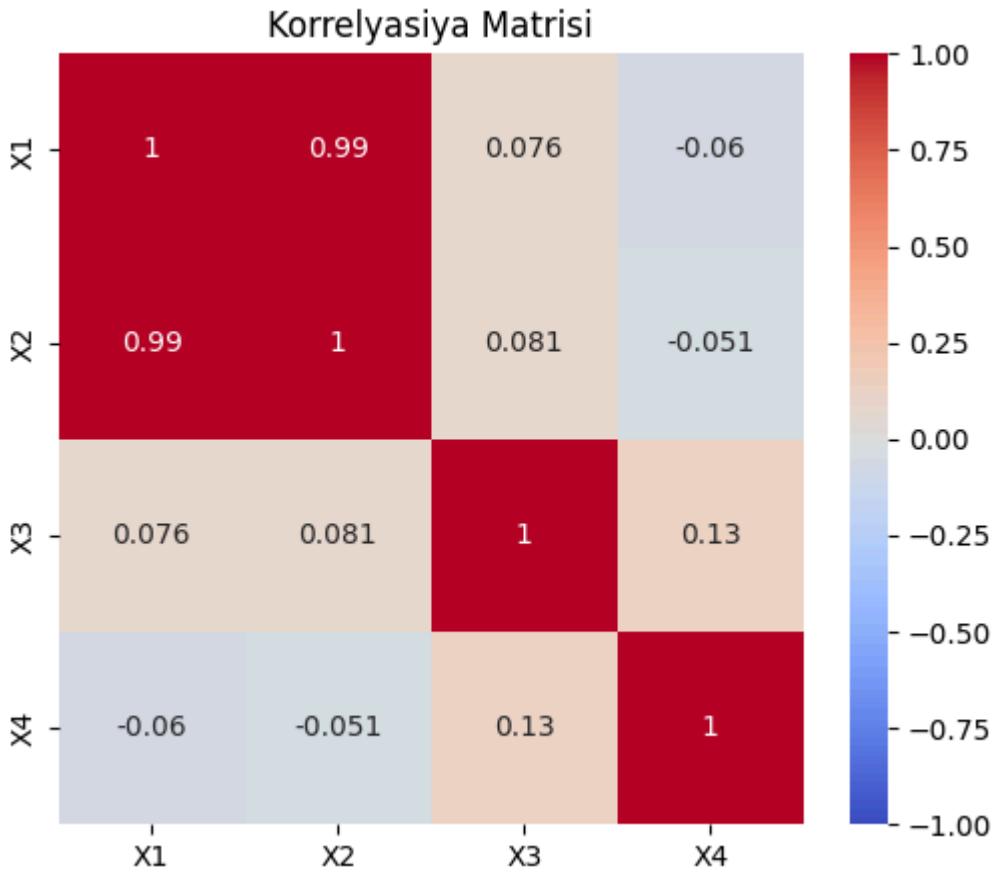
- **VIF (Variance Inflation Factor)** hesablanır.
  - VIF > 5 və ya 10 olarsa, təhlükə siqnalı verir.
- Korrelyasiya matrisinə baxıla bilər.

🧠 **Qeyd:** Əgər çoxlu xətti asılılıq varsa:

- Bir dəyişəni modeldən çıxarmaq
- PCA kimi ölçü azaltma üsulları
- Ridge və ya Lasso kimi requlyarizasiya üsulları nəzərdən keçirilə bilər

In [ ]:

In [ ]:

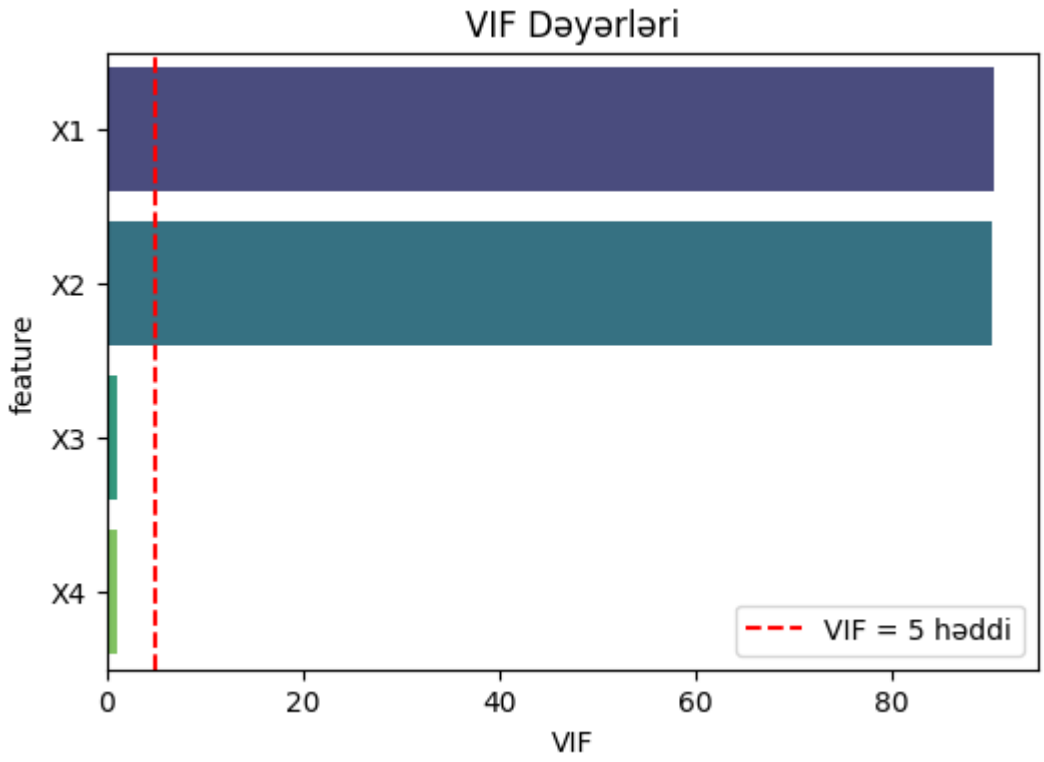


In [ ]:

```
/tmp/ipython-input-1230236099.py:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(data=vif_df, x="VIF", y="feature", palette="viridis")
```



📘 4. MODELİN DƏYƏRLƏNDİRİLMƏSİ

Bir xətti reqressiya modelini qurmaq qədər, o modelin **nə qədər yaxşı işlədiyini qiymətləndirmək** də çox vacibdir. Bu mərhələ, modelin həm təxmin gücünü, həm də statistik etibarlılığını yoxlayır.

4.1. R-Kvadrat (R^2) – İzah Edilən Dispersiya

Tərif:
R^2, asılı dəyişəndəki ümumi dispersiya nə qədərinin model tərəfindən izah edildiyini göstərir.

- Dəyər aralığı: 0 ilə 1 arasında kimi görünsə də **0-dan kiçik ola bilər** (aşağıda izah ediləcək).
- R^2 = 0: Model heç bir şeyi izah etmir
- R^2 = 1: Model bütün variasiyanı izah edir (mükəmməl uyğunluq)

Düstur:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

- SS_{res}: Xətaların kvadratları cəmi (residual sum of squares)
- SS_{tot}: Ümumi kvadratlar cəmi (total sum of squares)

📌 R^2 nə qədər yüksəkdirsə model o qədər yaxşıdır — amma diqqət: yüksək R^2 hər zaman yaxşı model demək deyil.

R^2 Niyə 0-dan Kiçik Ola Bilər?

Əgər model, sabit ortalamanı təxmin etməkdən belə daha pis performans göstərirsə, $SS_{res} > SS_{tot}$ olur və:

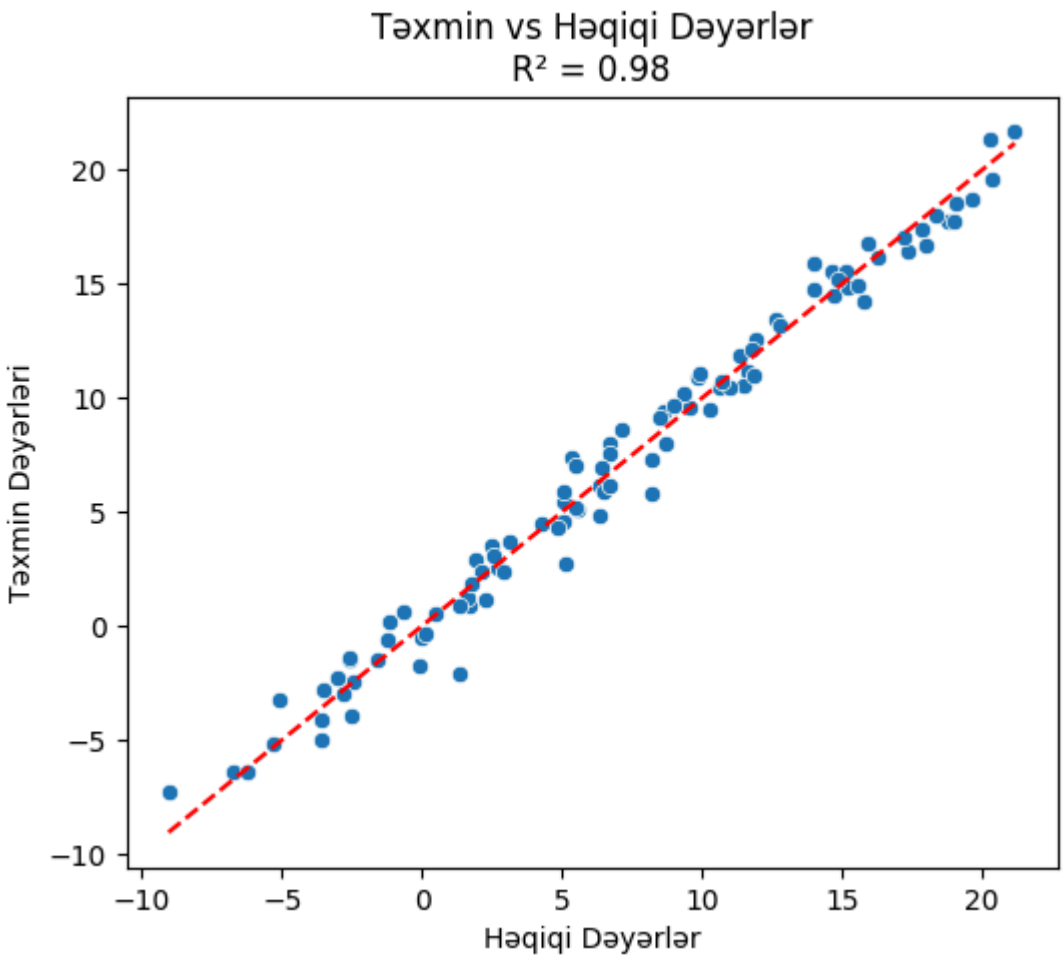
$$R^2 < 0$$

Bu, modelin məlumatlarla demək olar ki, heç bir uyğunluq göstərmədiyini, hətta təsadüfi təxmindən belə pis olduğunu göstərir.

In [ ]:

In [ ]:

```
R²: 0.9824
```



4.2. Düzəldilmiş R^2 (Adjusted R^2)

Problem:
Modelə daha çox dəyişən əlavə edildikcə R^2 təbii olaraq artır. Bu, modelin şişməsinə (overfitting) səbəb ola bilər.

Həll:
Düzəldilmiş R^2, dəyişən sayına görə R^2-ni cəzalandırır.

Düstur:

$$\text{Düzəldilmiş } R^2 = 1 - \left(\frac{(1 - R^2)(n - 1)}{n - p - 1} \right)$$

- n: Müşahidə sayı
- p: Sərbəst dəyişən sayı
- R^2: Klassik R^2

📌 Faydalı olmayan dəyişənlər modelə daxil edildikdə Düzəldilmiş R^2 azalır. Xüsusilə **çoxsaylı reqressiya** modellərində istifadə olunur.

In [ ]:

```
Model 1: (Sadə)
R²:           0.9824
Düzəldilmiş R²:  0.9821

Model 2: (Lazımsız dəyişənlərlə)
R²:           0.9829
Düzəldilmiş R²:  0.9820
```

4.3. Xəta Ölçümləri (MSE, RMSE, MAE)

Modelin təxminlərinin nə qədər yanıldığını ölçmək üçün:

- **MSE (Orta Kvadratik Xəta):**

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **RMSE (Kökaltı Orta Kvadratik Xəta):** $RMSE = \sqrt{MSE}$
- **MAE (Orta Mütləq Xəta):** Ortalama mütləq xəta

📌 Ümumi fərqlər:

- RMSE böyük meyllənmələri daha çox cəzalandırır
- MAE kənarlaşmalara daha davamlıdır
- MSE törəməsi alına bildiyi üçün riyazi əməliyyatlarda üstünlük verilir

In [ ]:

```
MSE: 0.9457
RMSE: 0.9725
MAE: 0.7748
```

4.4. p-Dəyərləri və Əmsalların mənalılığı

- p-dəyəri: Əmsalın sıfırdan fərqli olma ehtimalıdır
- Adətən $p < 0.05$ mənalı qəbul edilir
- Anlamsız əmsallar hər zaman çıxarılmaq məcburiyyətində deyil — kontekstə görə dəyərləndirilir

In [ ]:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.982
Model:                            OLS   Adj. R-squared:                  0.982
Method:                 Least Squares   F-statistic:                     2711.
Date:                Sat, 02 Aug 2025   Prob (F-statistic):           7.60e-86
Time:                        11:23:44   Log-Likelihood:                -139.10
No. Observations:                 100   AIC:                             284.2
Df Residuals:                      97   BIC:                             292.0
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          4.9106      0.254     19.318      0.000       4.406       5.415
X1             1.9658      0.033     58.898      0.000       1.900       2.032
X2            -1.4281      0.034    -42.156      0.000      -1.495      -1.361
==============================================================================
Omnibus:                        6.139   Durbin-Watson:                   2.073
Prob(Omnibus):                  0.046   Jarque-Bera (JB):                5.737
Skew:                           0.456   Prob(JB):                       0.0568
Kurtosis:                       3.738   Cond. No.                         19.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
ecified.
```

4.5. F-Statistikası – Modelin Ümumi mənalılığı

F-statistikası, modelin ümumilikdə mənalı olub-olmaması sualına cavab verir.

- H_0: Bütün β'lar = 0
- H_1: Ən az bir $\beta \neq 0$

Əgər p-dəyəri < 0.05 olarsa, model ümumilikdə mənalıdır.

In [ ]:

```
F-statistikası: 2710.6901
F-testi p-dəyəri: 0.0000
```

4.6. t-Testləri – Fərdi Əmsal Testi

Hər β əmsalı üçün ayrı-ayrılıqda t-testi aparılır:

$$t = \frac{\hat{\beta}}{SE(\hat{\beta})}$$

- SE: Əmsalın standart xətası
- t dəyəri böyüdükcə p-dəyəri kiçilir → mənalılıq artır

In [ ]:

```
const: t = 19.3177, p = 0.0000
X1: t = 58.8977, p = 0.0000
X2: t = -42.1564, p = 0.0000
```

```
/tmp/ipython-input-4233422888.py:20: FutureWarning: Series.__getitem__ treating key
s as positions is deprecated. In a future version, integer keys will always be trea
ted as labels (consistent with DataFrame behavior). To access a value by position,
use `ser.iloc[pos]`
  t_value = model.tvalues[i]
/tmp/ipython-input-4233422888.py:21: FutureWarning: Series.__getitem__ treating key
s as positions is deprecated. In a future version, integer keys will always be trea
ted as labels (consistent with DataFrame behavior). To access a value by position,
use `ser.iloc[pos]`
  p_value = model.pvalues[i]
/tmp/ipython-input-4233422888.py:20: FutureWarning: Series.__getitem__ treating key
s as positions is deprecated. In a future version, integer keys will always be trea
ted as labels (consistent with DataFrame behavior). To access a value by position,
use `ser.iloc[pos]`
  t_value = model.tvalues[i]
```

4.7. VIF (Variance Inflation Factor) – Çoxlu Xətti Asılılıq Yoxlanışı

- VIF, bir sərbəst dəyişənin digər dəyişənlərlə olan üst-üstə düşməsini ölçür.
- $VIF > 5$ (və ya 10) olması çoxlu xətti asılılığa işarə edir.

📌 Yüksək VIF dəyərləri varsa:

- Dəyişəni çıxarmaq
- PCA istifadə etmək
- Ridge/Lasso kimi requlyarizasiya üsullarına üstünlük verilə bilər

In [ ]:

```
  feature       VIF
0      X1  9.654774
1      X2  1.048221
2      X3  9.660076
3      X4  1.002491
```

📘 5. MODEL DİAQNOSTİKASI (XƏTA VƏ QALIQ ANALİZİ)

Xətti reqressiya modelini qurduqdan sonra, modelin təxminləri ilə real müşahidələr arasındakı fərqləri araşdıraraq modelin necə performans göstərdiyini anlamağa çalışırıq. Bu fərqlərə **qalıq (residual)** deyilir.

5.1. Qalıq (Residual) Nədir? Niyə Vacibdir?

Bir müşahidədə həqiqi dəyər y_i ilə modelimizin təxmini $\hat{y}_i$ arasındakı fərq:

$$e_i = y_i - \hat{y}_i$$

Bu fərq, modelin o müşahidədəki xətasını göstərir.

- **Kiçik qalıqlar**, modelin o nöqtədə uğurlu olduğunu,
- **Böyük qalıqlar**, modelin o nöqtədə yaxşı təxmin edə bilmədiyini göstərir.

Qalıqlar, modelin harada güclü və ya zəif olduğunu göstərən ən vacib göstəricilərdir.

5.2. Qalıqların Məqsədi: Model Fərziyyələrini Test Etmək

Xətti reqressiyanın doğru işləməsi üçün bəzi fərziyyələr var idi (xəttilik, homoskedastiklik, normal paylanma, müstəqillik). Qalıqlar vasitəsilə bu fərziyyələrin təmin edilib-edilmədiyini yoxlayırıq.

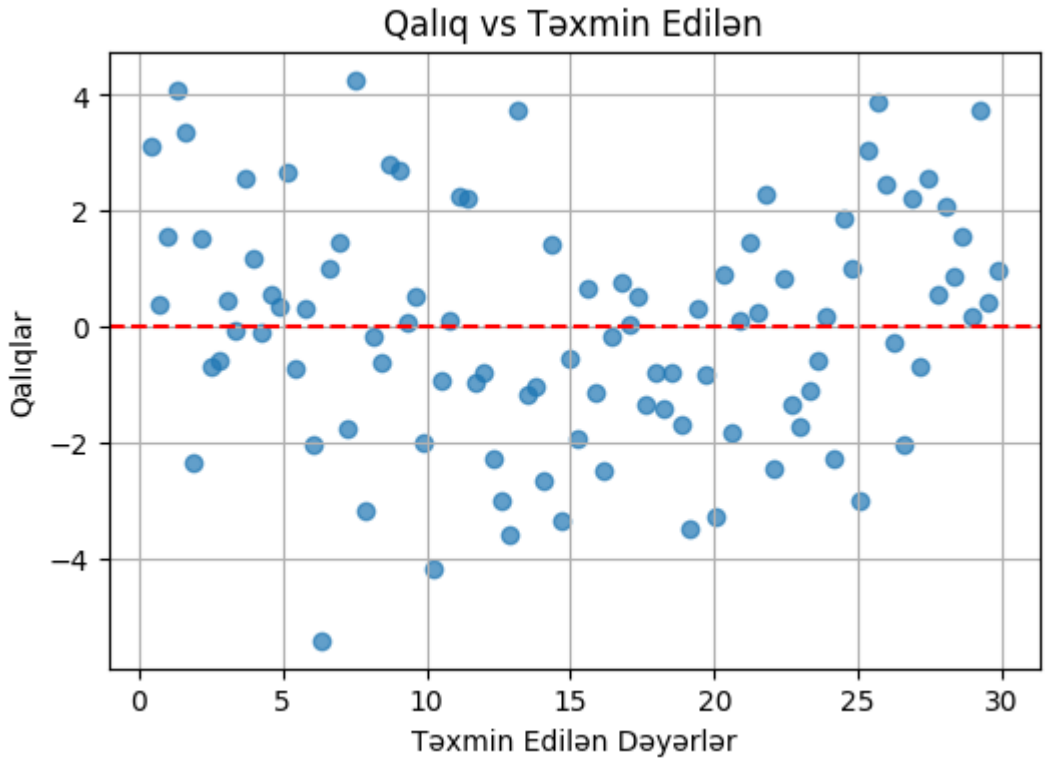
- Qalıqların təsadüfi paylanması → yaxşı model
- Qalıqlarda sistematik bir naxış → problem var

5.3. Qalıq Qrafiklərindən İstifadə Edərək Yoxlama

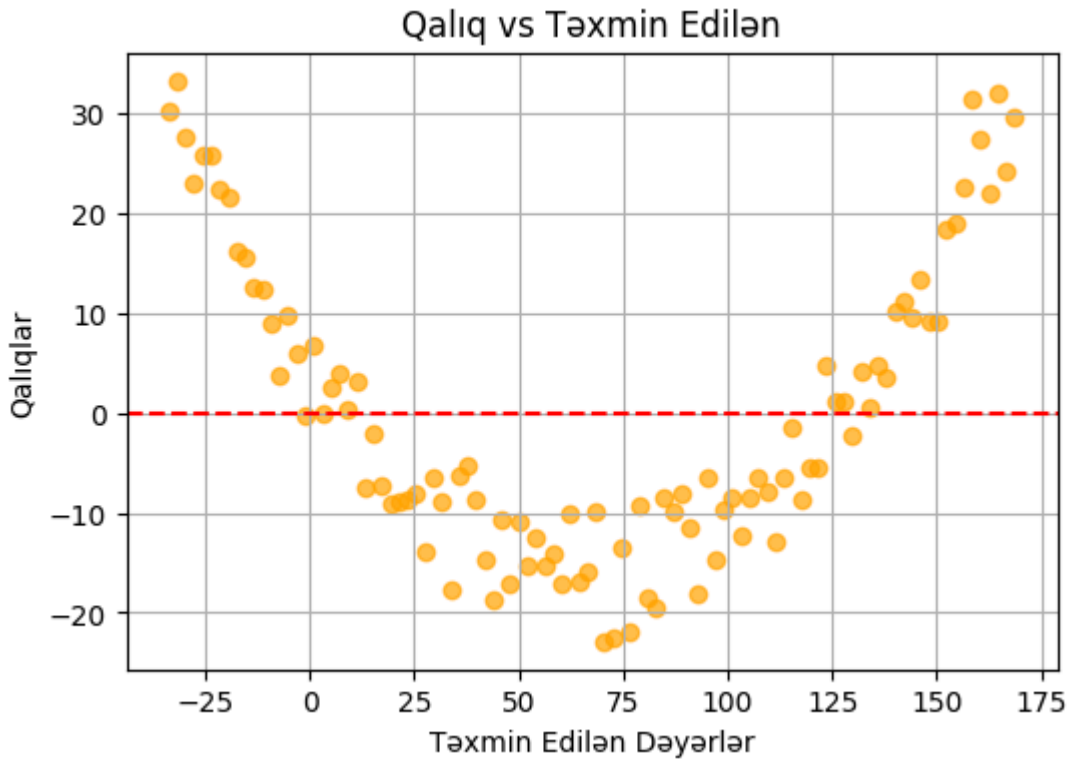
1. Qalıq vs Təxmin Edilən Dəyərlər Qrafiki

- Burada $\hat{y}$ oxunda qalıqlar göstərilir.
- **Gözlənilən:** Qalıqlar üfüqi bir zolaq şəklində və təsadüfi paylanmalıdır.
- **Problemli Vəziyyətlər:**
  - Qalıqlar müəyyən bir əyri əmələ gətirirsə → model xətti deyil (xəttilik pozuntusu)
  - Qalıqlar "yelpazə" şəklində yayılırsa → heteroskedastiklik (sabit olmayan dispersiya) var.

In [ ]:



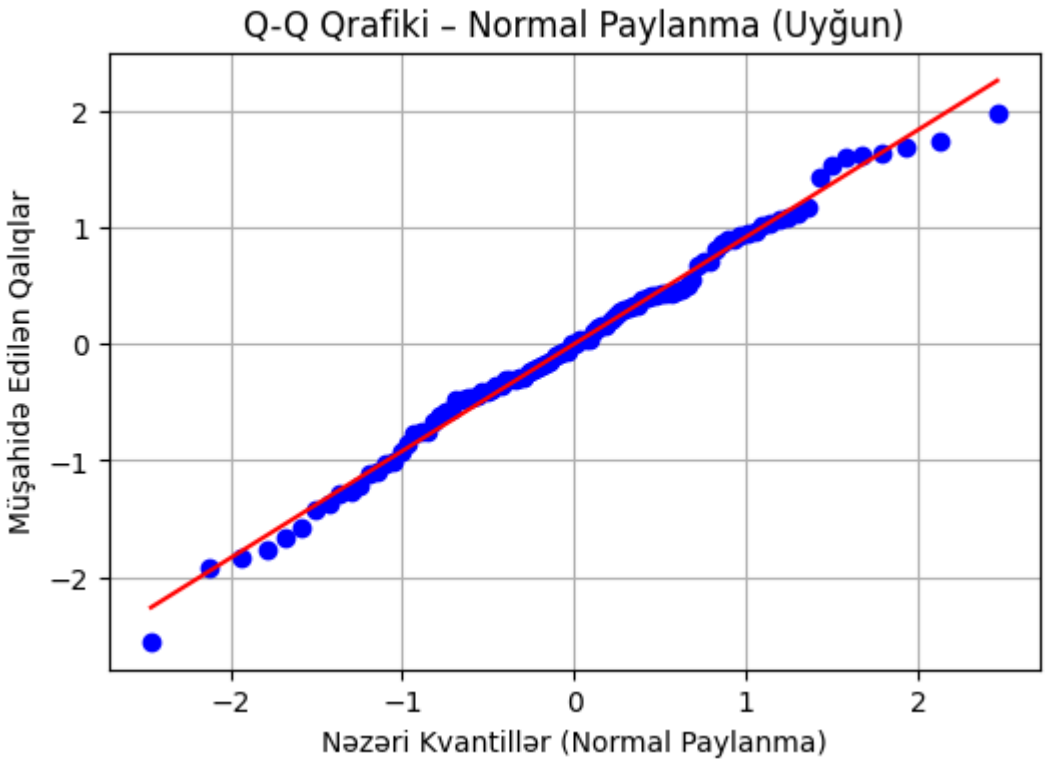
In [ ]:



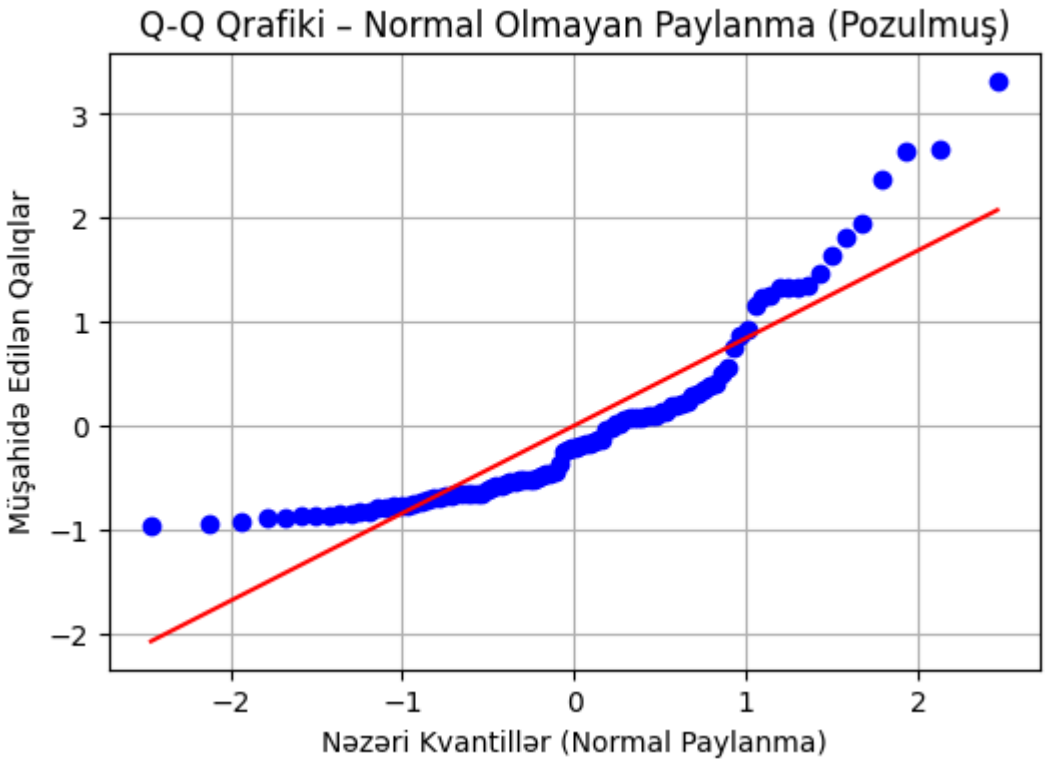
2. Q-Q (Quantile-Quantile) Qrafiki

- Qalıqların normal paylanmaya uyğun olub-olmadığını görməyə imkan verir.
- Qalıqlar düz bir xətt üzərində sıralanırsa → normal paylanma fərziyyəsi təmin edilir.
- Əyrilik və ya kənaraçıxma varsa → normal paylanma pozuntusu.

In [ ]:



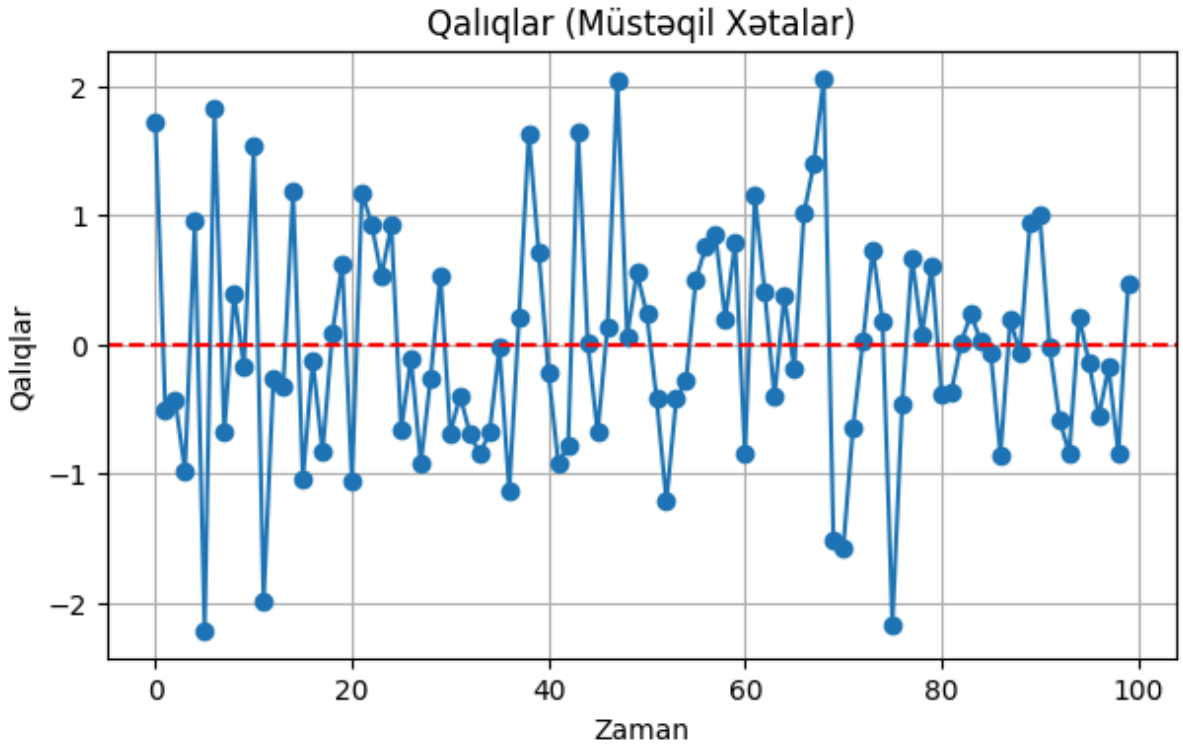
In [ ]:



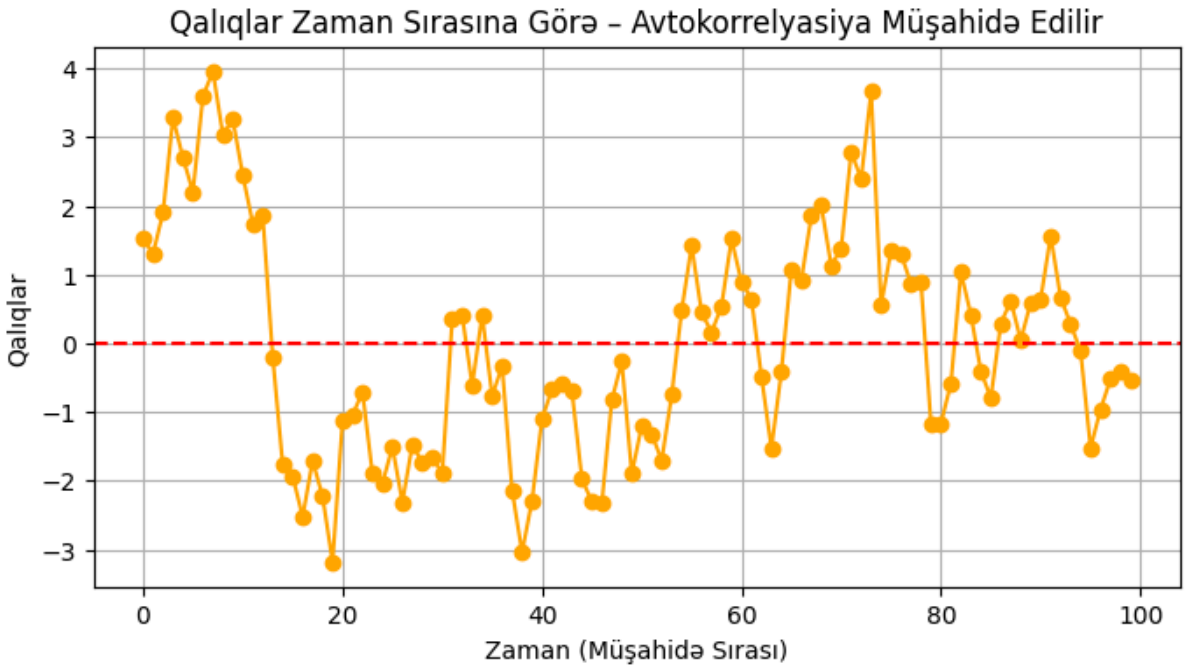
3. Qalıqların Zaman Sırasına Görə Qrafiki

- Xüsusilə zaman seriyası analizlərində qalıqların bir-birindən asılı olmaması lazımdır.
- Qalıqlar ardıcıl bir naxış göstərirsə → müstəqillik fərziyyəsi pozulur.

In [ ]:



In [ ]:



5.4. Kənar Nöqtələr (Outliers) və Təsirli Müşahidələr (Leverage Points)

Bəzi müşahidələr, modelə gözlənilməz dərəcədə böyük təsir göstərə bilər. Bunları aşkar etmək üçün müxtəlif metrikalar istifadə olunur.

5.4.1. Leverage (Ling)

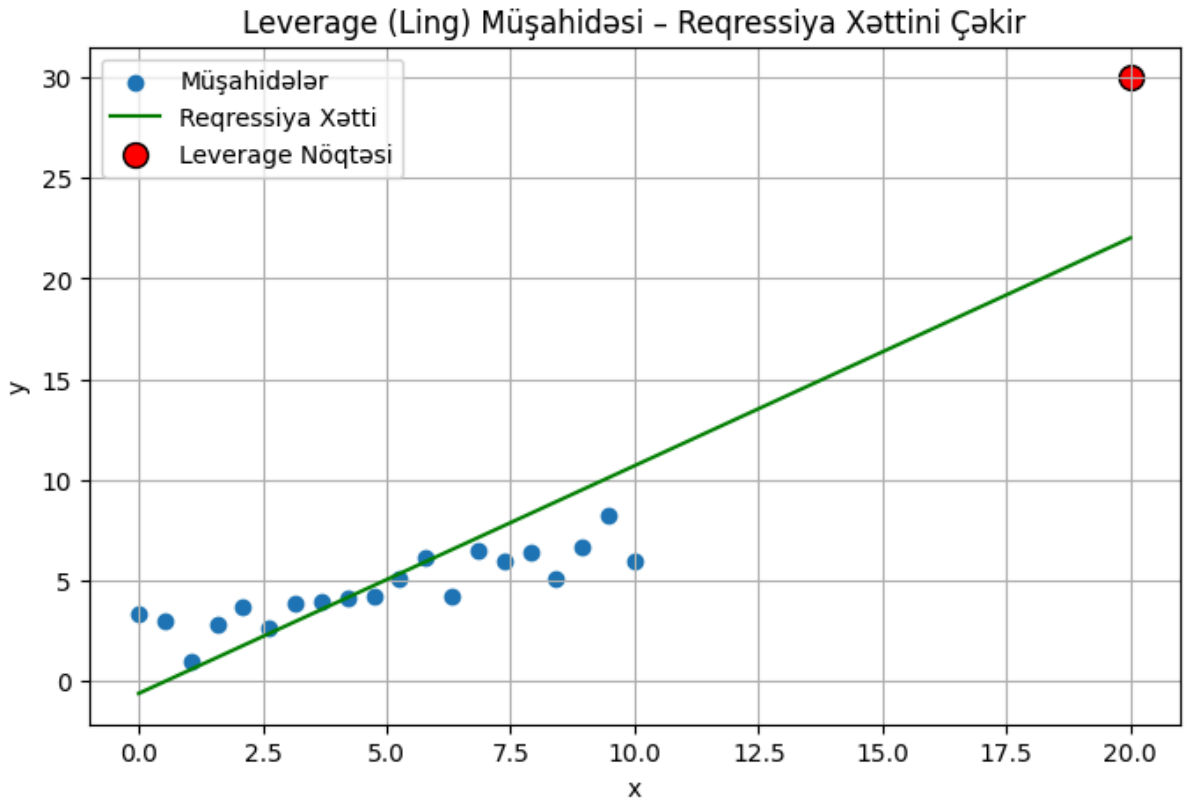
Nə edir?
Leverage, bir müşahidənin müstəqil dəyişənlər (X-lər) baxımından nə qədər "kənarda" olduğunu ölçür.

- Əgər bir müşahidə X fəzasında çox fərqli, qeyri-adi (kənar) bir nöqtədirsə, yüksək leverage dəyərinə malikdir.
- Bu o deməkdir ki, bu müşahidə modelin xəttini və ya müstəvisini daha çox təsir edə bilər.

Konkret Nümunə:
Ev qiymətlərini təxmin edən modeldə, əksər evlər 100-200 m^2 olduğu halda, 1000 m^2-lik bir ev varsa, bu müşahidə yüksək leverage-ə malikdir.

In [ ]:

In [ ]:



5.4.2. Cook Məsafəsi (Cook's Distance)

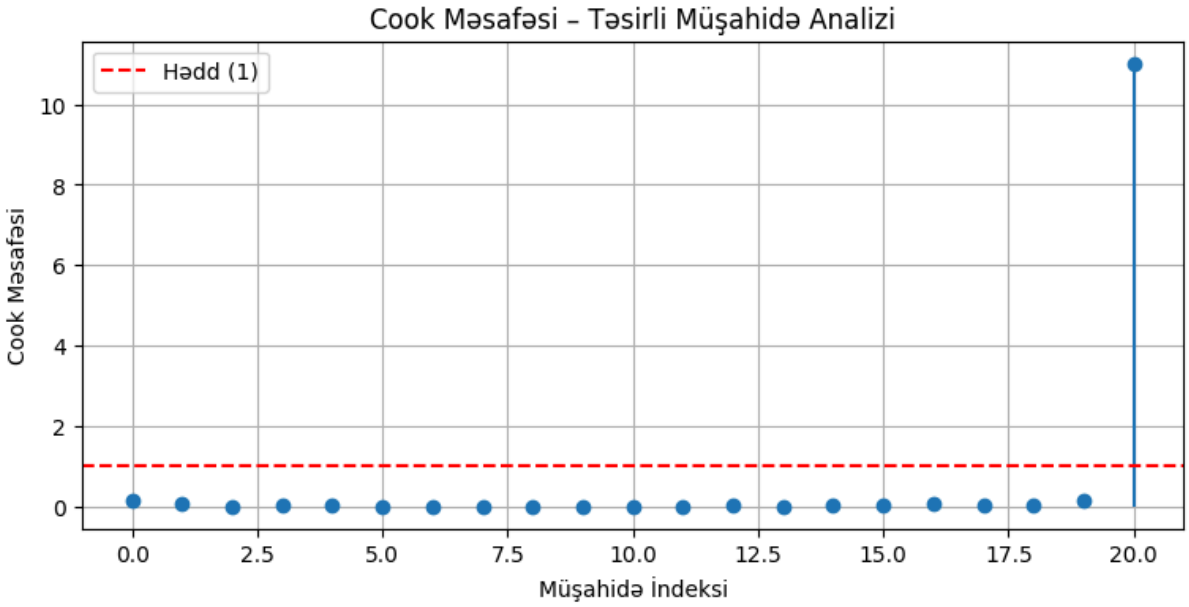
Nə edir?
Cook Məsafəsi, hər bir müşahidənin model təxminləri üzərindəki ümumi təsirini ölçür.

- Yəni, bu müşahidə olmadan model nə qədər dəyişərdi, bunu hesablayır.
- Həm leverage, həm də qalığı birlikdə qiymətləndirir.

Konkret Nümunə:
Deyək ki, modeldə bir müşahidə var; yüksək leverage-ə malikdir və eyni zamanda model tərəfindən pis təxmin edilir (böyük qalıq). Bu müşahidə Cook Məsafəsi baxımından da yüksək çıxacaq.

Yüksək Cook Məsafəsi adətən 1-dən yuxarıdır və bu halda müşahidə model parametrlərini əhəmiyyətli dərəcədə təsir edir.

In [ ]:



5.4.3. DFBETAS

Nə edir?

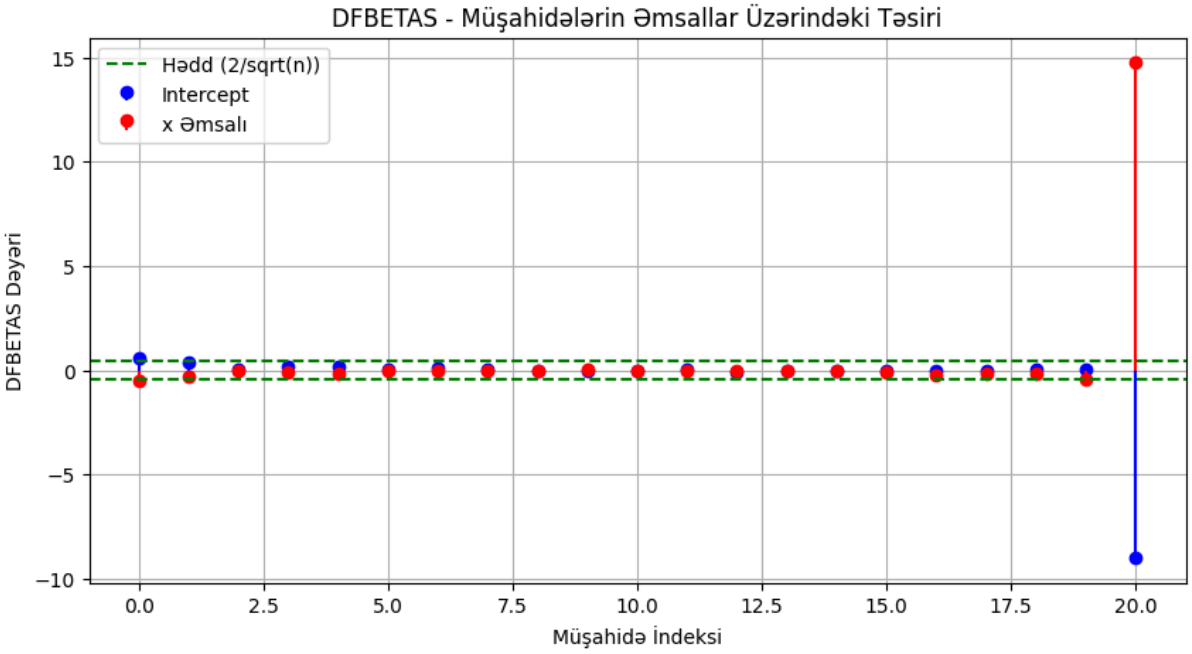
DFBETAS, hər bir müşahidənin, modeldəki hər bir əmsal üzərindəki təsirini ayrı-ayrılıqda ölçür.

- Hər müşahidəni çıxarıb, əmsallarda nə qədər dəyişiklik olduğunu hesablayır.
- Böyük dəyişikliklər varsa, o müşahidə "təsirli" hesab olunur.

Konkret Nümunə:

Əgər bir müşahidə çıxarıldıqda, β_2 əmsalı %30 dəyişirsə, bu müşahidə β_2 üçün əhəmiyyətli bir təsirə malikdir deməkdir.

In [ ]:

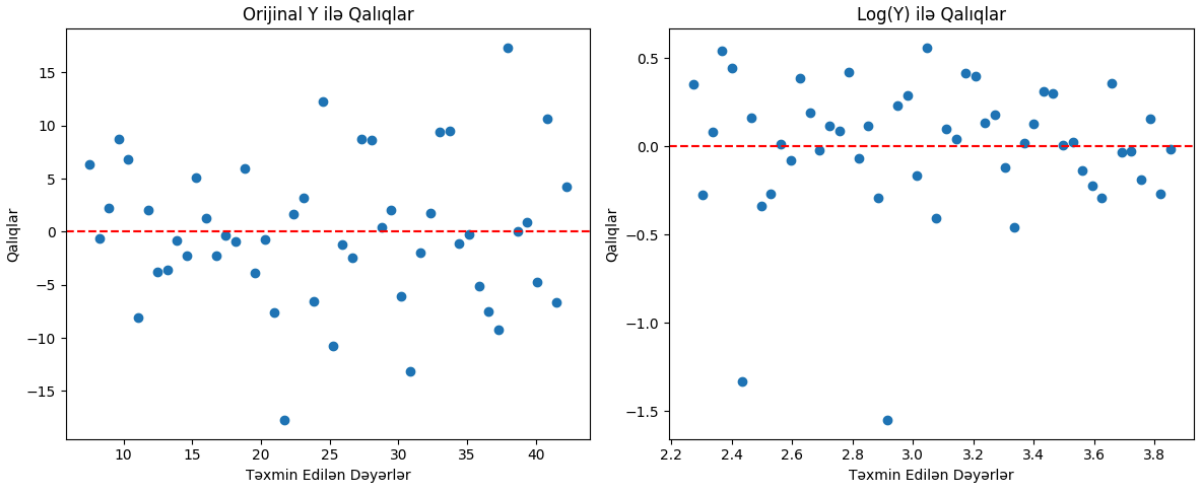


5.5. Qalıqlar Üzrə Model Təkmilləşdirmə Üsulları

Əgər qalıqlar yuxarıdakı fərziyyələrə uyğun gəlmirsə:

- **Data transformasiyaları:**
  - Asılı dəyişən və ya müstəqil dəyişənlərdə loqarifm, kvadrat kök kimi çevrilmələr aparmaq
- **Fərqli modellər:**
  - Polinomial reqressiya, (robust) reqressiya kimi fərziyyələrə daha dözümlü modellər seçmək
- **Kənar nöqtələrlə(outlier) mübarizə:**
  - Bu müşahidələri analizdən çıxarmaq və ya çəkiləndirmək
- **Daha çox məlumat toplamaq:** Xüsusilə kənar dəyərlərin təsirini azaltmaq üçün

In [ ]:



📗 6. FEATURE ENGİNEERİNG VƏ PREPROCESSİNG

Xətti reqressiya modelinin uğuru, verilənlərin keyfiyyətinə və uyğun şəkildə hazırlanmasına bağlıdır. Bu səbəbdən **xüsusiyyət mühəndisliyi (feature engineering)** və **verilənlərin ön işləməsi** çox vacibdir. Bilməli olduğunuz əsas anlayışlar bunlardır:

6.1. Dummy Dəyişənlər (Kategorik Verilənlər Üçün)

Xətti reqressiya **ədədi verilənlərlə** işləyir, buna görə də kategorik dəyişənləri ədədi formata çevirməliyik.

Nə edirik?

Kategorik dəyişənlərdəki hər bir kateqoriya üçün bir **ikili (0/1)** dəyişən yaradırıq. Bunlara dummy dəyişənlər deyilir.

Nümunə:

"Cins" dəyişəni 2 kateqoriyadan ibarətdir: Kişi, Qadın

- Kişi üçün: [1, 0]
- Qadın üçün: [0, 1]

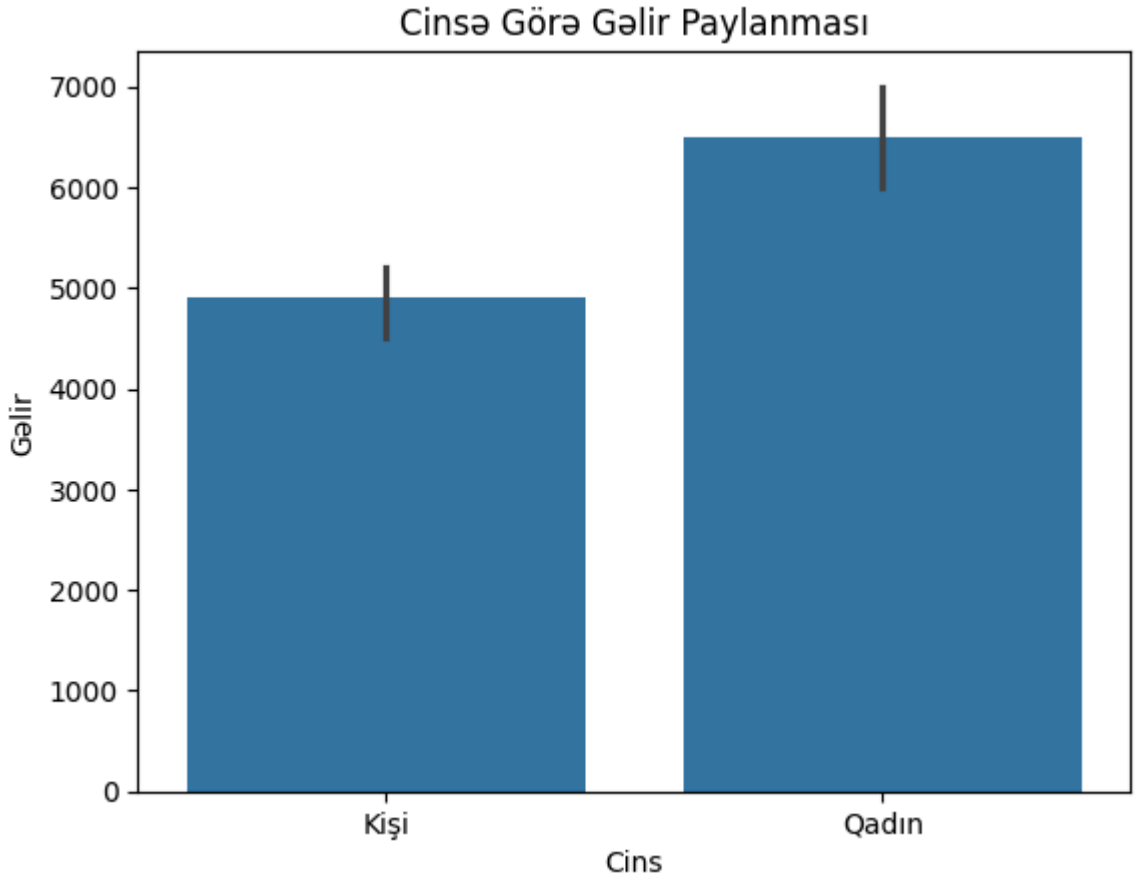
Lakin modeldə **dummy dəyişən tələsi** adlanan, yəni kategorik dəyişənlərdə artıq dummy yaratma xətasını önləmək üçün, kateqoriyalardan biri referans olaraq saxlanılır. Məsələn, yalnız bir dummy dəyişən yaradıb onun 1 və ya 0 olmasına görə digər kateqoriyanı anlayırıq.

Niyə vacibdir?

- Kategorik verilənləri ədədi formata çevirmək lazımdır.
- Model əmsalları bu dummy dəyişənlərin təsirini ölçür.

In [ ]:

```
   Gəlir  Cins_Qadın
0   5000       False
1   6000        True
2   4500       False
3   7000        True
4   5200       False
```



6.2. Xüsusiyyət Miqyaslaması (Feature Scaling)

Dəyişənlərin fərqli miqyasda olması modelin təlimini çətinləşdirə bilər.

İki əsas üsul:

a) Standardizasiya (Z-hesabı normallaşdırması)

Dəyişənləri ortalaması 0, standart kənaraçıxması 1 olacaq şəkildə çeviririk:

$$z = \frac{x - \mu}{\sigma}$$

- μ: dəyişənin ortalaması
- σ: standart kənaraçıxması

b) Min-Maksimum Normallaşdırma

Verilənləri 0 ilə 1 arasına sıxışdırırıq:

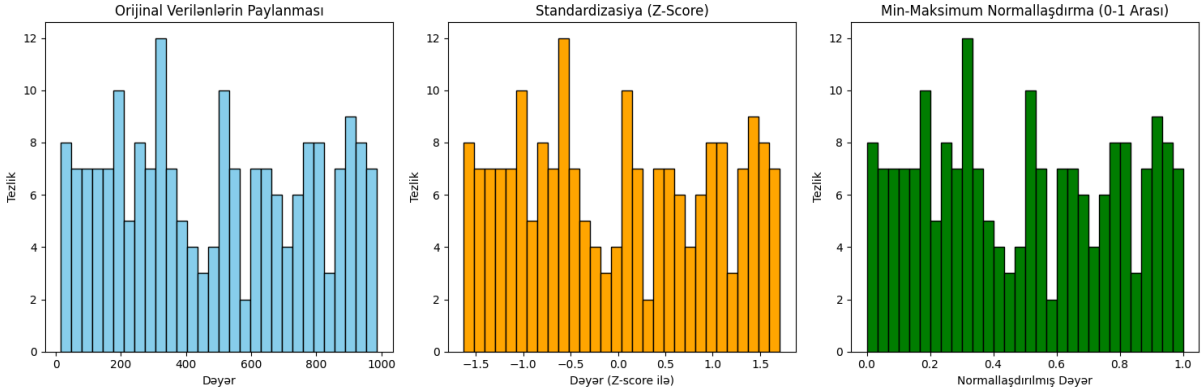
$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- x_{min}, x_{max}: dəyişənin minimum və maksimum dəyərləri

Niyə?

- Xüsusilə reqressiya əmsallarının şərhi və alqoritmlərin dayanıqlılığı üçün
- Bəzi alqoritmlər (məsələn, requlyarizasiyalı modellər) üçün şərtdir

In [ ]:

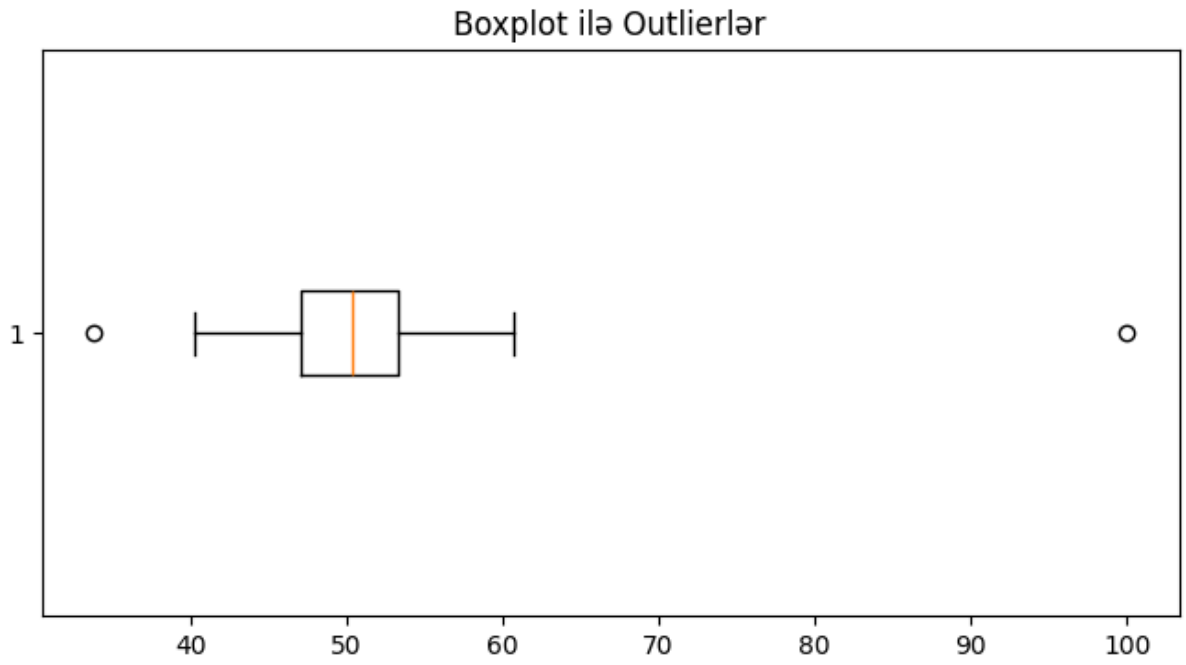


6.3. Kənar Nöqtələrin Təhlili (Outlier Analysis)

Aykırı nöqtələr, modeli yanlış istiqamətləndirə bilər.

- Əvvəlcə vizuallaşdırma (boxplot, səpələnmə qrafiki) ilə müəyyən edilir.
- Statistik üsullarla (Z-hesabı, IQR) təyin edilir.
- Modelin həssaslığına görə aykırı nöqtələr çıxarılır və ya çəkiləndirilir.

In [ ]:



6.4. Qarşılıqlı Təsir Terminləri (Interaction Terms)

Bəzən iki dəyişənin birlikdə təsiri, ayrı-ayrı təsirlərindən fərqli ola bilər.

Nümunə:

Bir məhsulun satışında həm qiymət, həm də reklam büdcəsi vacibdir, lakin bu ikisi birlikdə qarşılıqlı təsirdə ola bilər.

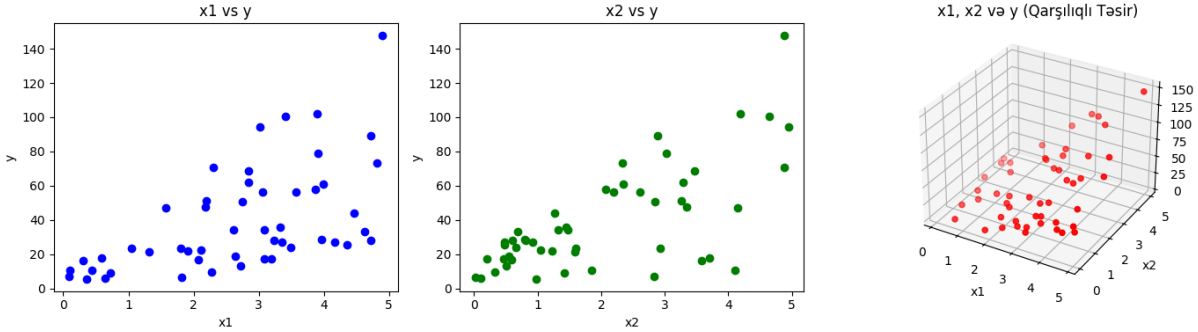
Model Təsviri:

Əgər dəyişənlər x_1 və x_2 olarsa, qarşılıqlı təsir termini:

$$x_1 \times x_2$$

Bu termin modelə əlavə olunur və əmsalı qarşılıqlı təsirin təsirini göstərir.

In [ ]:



6.5. Polinomial Reqressiya (Doğrusal Olmayanlığı Modelləşdirmək Üçün)

Bəzi əlaqələr doğrusal olmaya bilər. Bu halda polinom terminləri əlavə edirik:

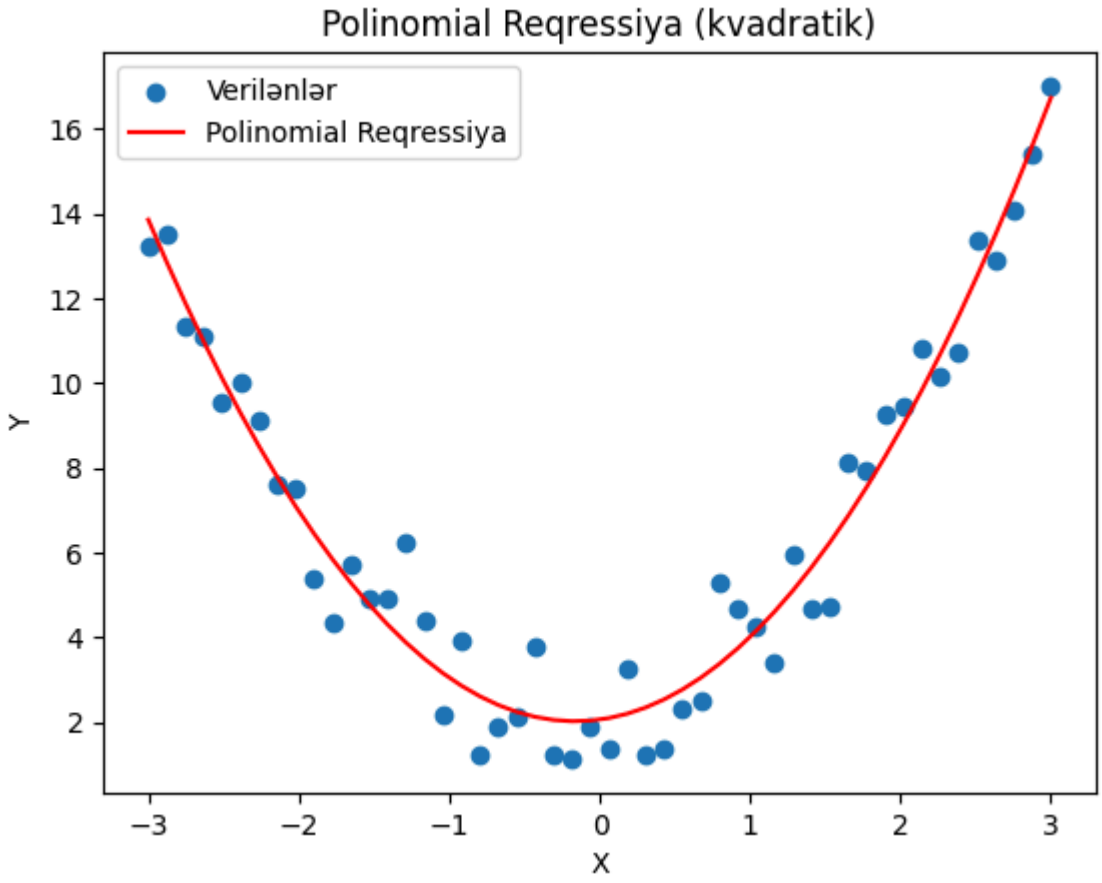
Nümunə:

Bir dəyişənin kvadratı modelə daxil edilir:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

- Beləliklə, əyri şəkildəki əlaqələr tutula bilər.
- Polinom dərəcəsi artırıla bilər, lakin həddindən artıq olarsa, həddindən artıq öyrənmə (overfitting) baş verər.

In [ ]:



📗 7. XƏTTİ REQRESSİYANIN ALTERNATİVLƏRİ

Klassik xətti reqressiya bir çox hallarda işə yarayır, ancaq bəzi problemləri vardır:

- Çoxlu xətti asılılıq (multikolinearlıq) varsa,
- Model həddindən artıq öyrənməyə (overfitting) meyllidirsə,
- Verilənlər aykırı nöqtələrdən təsirlənirsə,
- Dəyişən sayı çoxdursa,

Alternativ üsullar istifadə olunur. Bunlar adətən **requlyarizasiya** və ya **robust regression** üsullarıdır.

7.1. Ridge Regression (L2 Regularization)

Əsas fikir:

Model əmsallarının ölçüsünü cəzalandıraraq həddindən artıq öyrənmənin qarşısını almaq.

Model tənliyi:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

Burada:

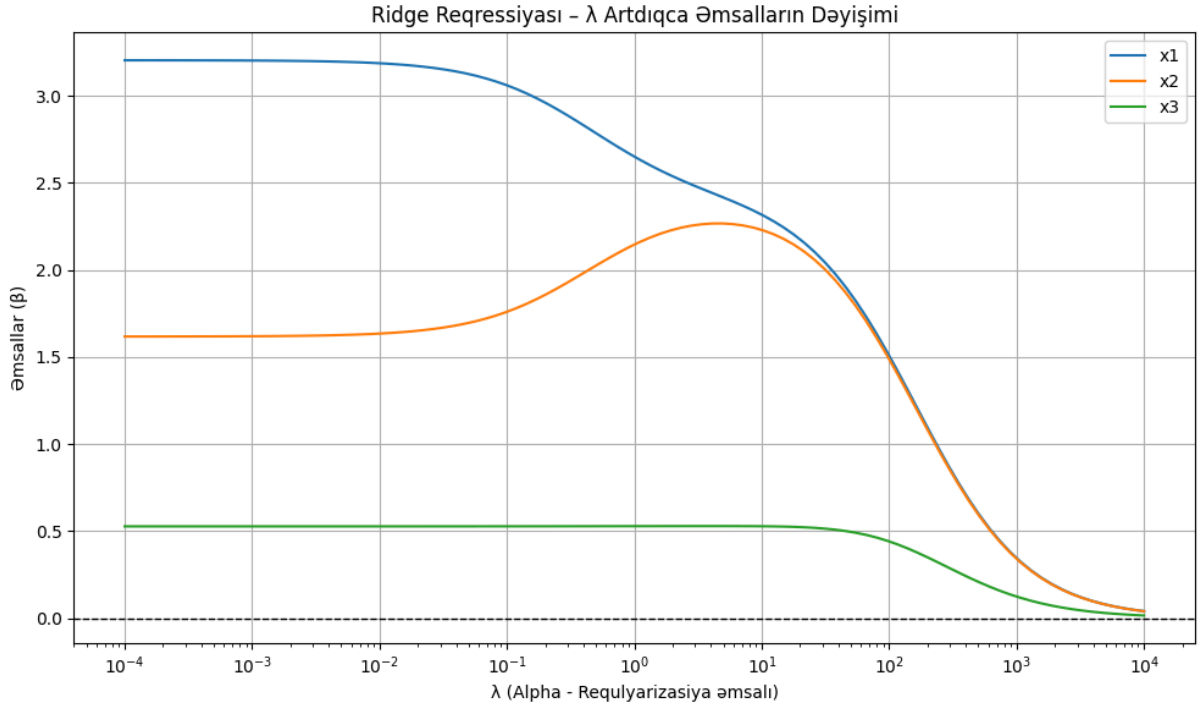
- Birinci termin klassik **RSS** (Qalıq Kvadratlarının Cəmi)
- İkinci termin isə əmsalların kvadratlarının cəmidir (L2 norması)
- $\lambda \geq 0$ isə cəza (requlyarizasiya) parametrdir

Təsir:

- λ artdıqca əmsallar kiçilir (sıfıra doğru çəkilir, amma tam sıfırlanmır).
- Bu, modeldə çoxlu xətti asılılıq problemini azaldır və həddindən artıq öyrənmənin qarşısını alır.

In [ ]:

In [ ]:



7.2. Lasso Reqressiyası (L1 Requlyarizasiyası)

Əsas fikir:

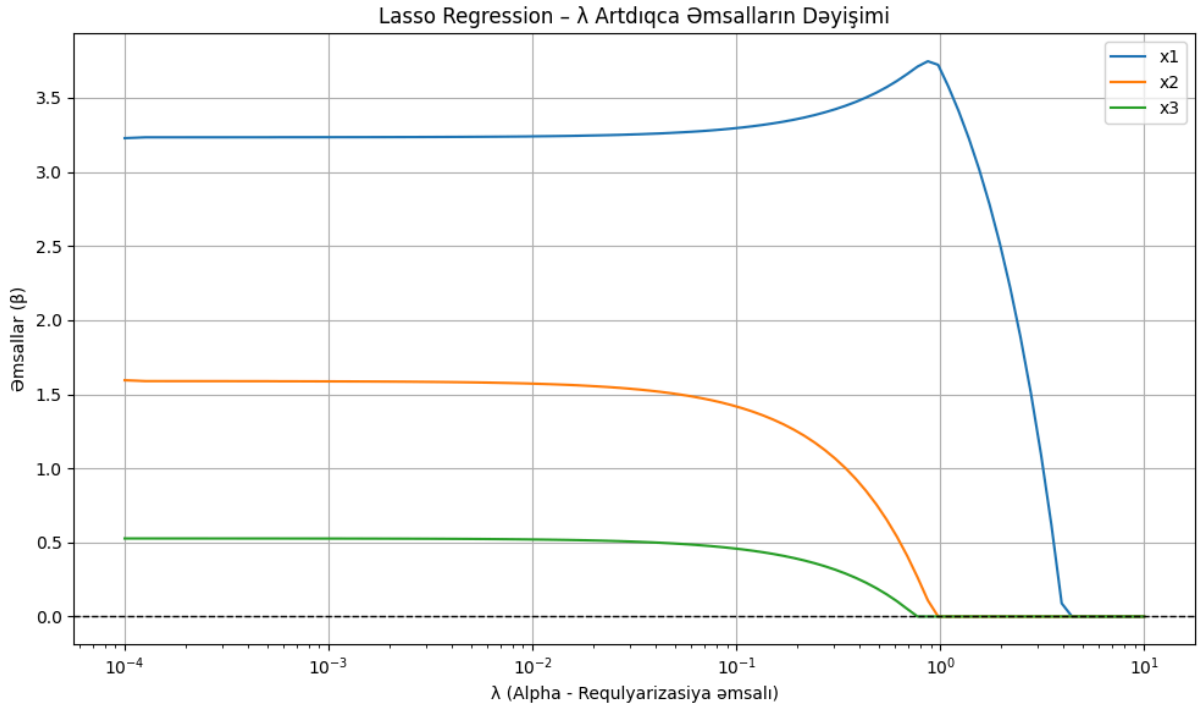
Əmsalların ümumi mütləq dəyərini cəzalandıraraq bəzi əmsalların tam sıfıra düşməsini təmin etmək.

Model tənliyi:

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- L1 normu, bəzi əmsalları tam olaraq sıfıra bərabərləşdirərək dəyişən seçimi edir.
- Bu sayədə model daha sadə olur (xüsusiyyət seçimi).

In [ ]:



7.3. Elastic Net (L1 + L2 Requlyarizasiyası)

Əsas fikir:

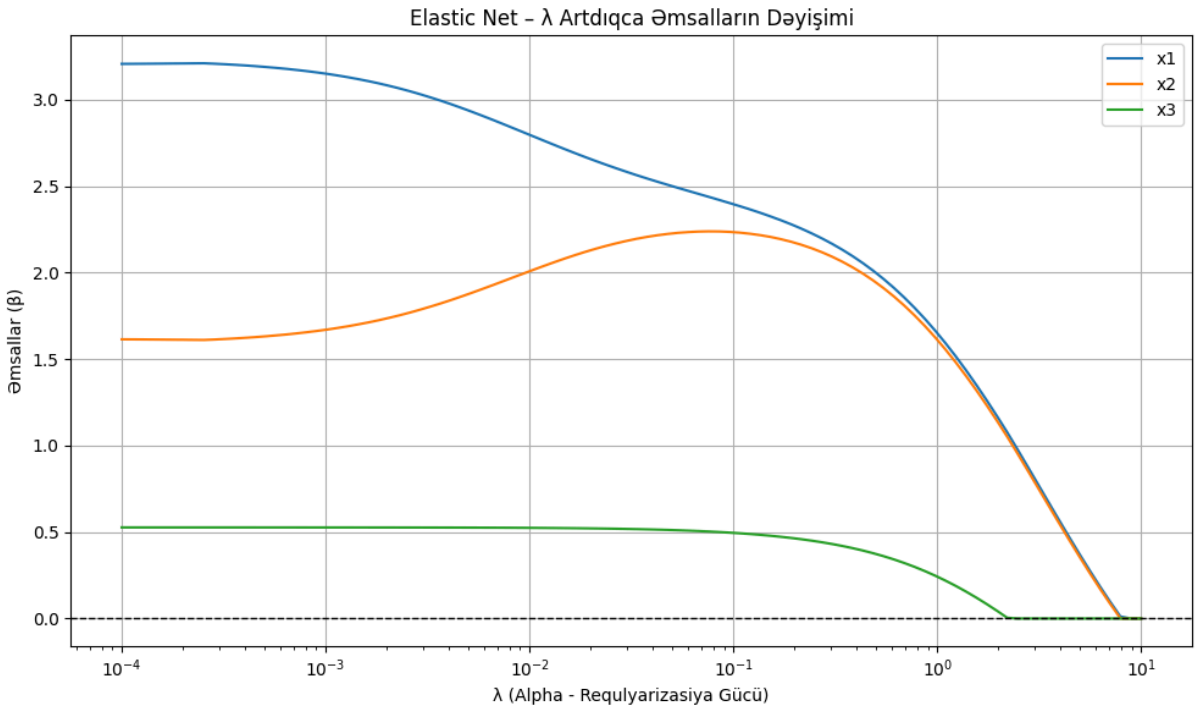
Həm Ridge, həm də Lasso cəzasını birlikdə istifadə edərək üstünlükləri birləşdirmək.

Model tənliyi:

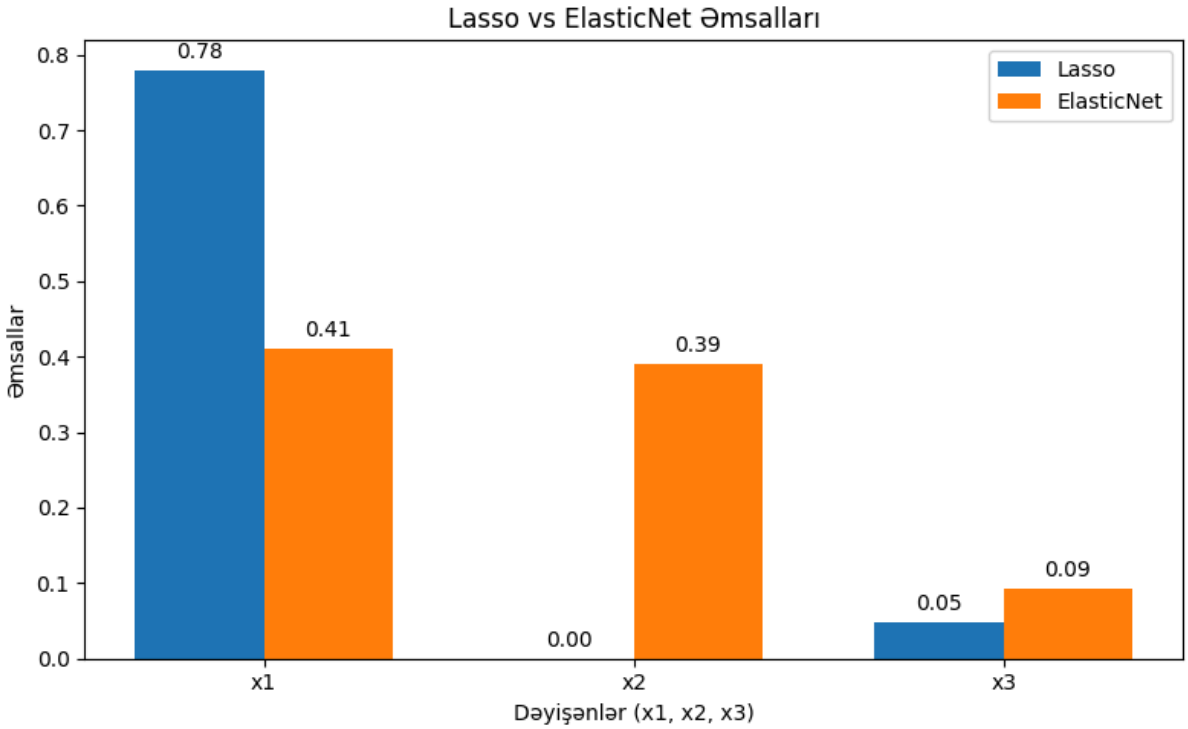
$$\min_{\beta} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \left(\alpha \sum_{j=1}^{p} |\beta_j| + (1 - \alpha) \sum_{j=1}^{p} \beta_j^2 \right)$$

- $\alpha \in [0, 1]$ parametri, L1 və L2 arasındakı tarazlığı tənzimləyir.

In [ ]:



In [ ]:



Xüsusiyyət	Lasso (L1)	ElasticNet (L1 + L2)
Feature Selection	Bəli – birbaşa sıfırlayır	Bəli – amma L2 ilə sabitlik artır
Multicollinearityə qarşı	Pis – birini seçir, digərini sıfırlar	Daha balanslı – oxşar olanları birlikdə saxlayır
Əmsalların sıfırlanma sürəti	Tez	Daha yavaş (çünki L2 qarşı çıxır)
Yalnız L1 olduğu zaman	Həddindən artıq dəyişkənlər varsa model qeyri-sabitdir	L2 sabitləşdirmə gətirir (ridge davranışı)

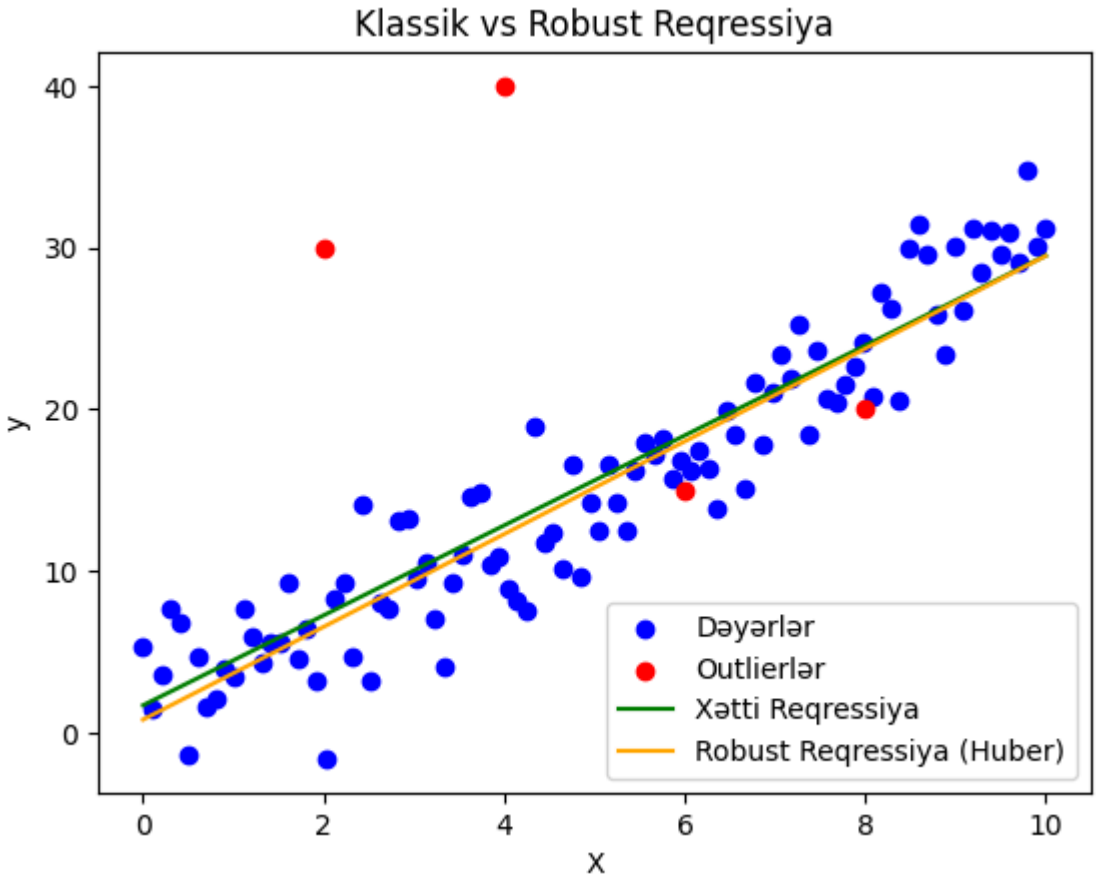
7.4. Robust Regression (Outlierlərə Dayanıqlı Reqressiya)

Əsas fikir:

Outlierlərin model təxminini çox təsir etməsinin qarşısını almaq.

- Klassik reqressiya qalıqların kvadratlarını minimallaşdırır, bu səbəbdən böyük qalıqlar modelə çox təsir edir.
- Robust Regression isə qalıqları daha az cəzalandıran funksiyalar istifadə edir (məsələn, Huber loss).

In [ ]:



📗 8. TƏTBİQ (PYTHON & SCIKIT-LEARN / STATSMODELS)

Nəzəri bilikləri praktikaya tökmək, öyrənmənin ən möhkəm yoludur. Xətti reqressiyanı anlamaq və düzgün tətbiq etmək üçün Python dünyasında ən çox seçilən iki kitabxana var: **scikit-learn** və **statsmodels**. Hər ikisi də fərqli güclü tərəflərə malikdir.

8.1. Scikit-learn ilə LinearRegression

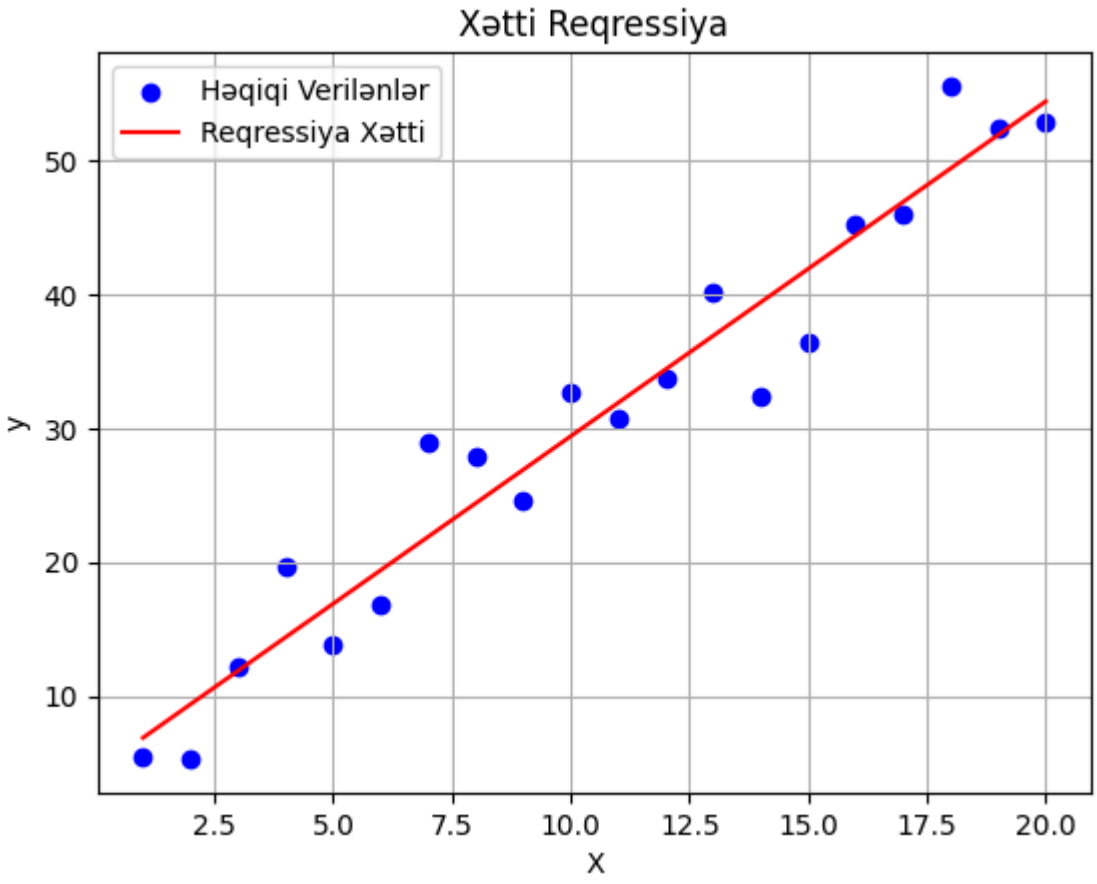
Scikit-learn, maşın öyrənməsi üçün optimallaşdırılmış, istifadəsi asan bir kitabxanadır. Xətti reqressiya modeli qurmaq, öyrətmək və təxmin etmək üçün `LinearRegression` sinifindən istifadə edirik.

- Model qurma və training sürətlidir,
- Əsas reqressiya əməliyyatları asanlıqla yerinə yetirilir,
- Lakin ətraflı statistik analiz və testlər məhduddur.

İstifadə prosesi aşağıdakı kimidir:

- Verilənlər seti train və test olaraq ayrılır,
- Model train verilənləri ilə `fit` edilir (öyrənir),
- Test verilənləri üzərində təxmin aparılır.

In [16]:



8.2. Statsmodels ilə OLS (Ordinary Least Squares)

Statsmodels daha çox statistik analiz üçün hazırlanmışdır. `ols()` funksiyası ən kiçik kvadratlar üsulu ilə reqressiyanı qurur.

- Model haqqında çox ətraflı təhlil,
- Əmsalların əhəmiyyətlilik testləri (t-test, p-dəyəri),
- İnam aralıqları (confidence interval),

- Hipotez testləri kimi ətraflı məlumatlar verir.

Bu sayədə yalnız təxmin deyil, modelin statistik gücü və etibarlılığı haqqında da ətraflı məlumat əldə etmək olar.

In [ ]:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.938
Model:                            OLS   Adj. R-squared:                  0.935
Method:                 Least Squares   F-statistic:                     274.5
Date:                Sun, 03 Aug 2025   Prob (F-statistic):           2.41e-12
Time:                        05:35:33   Log-Likelihood:                -54.515
No. Observations:                  20   AIC:                             113.0
Df Residuals:                      18   BIC:                             115.0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          4.3707      1.809      2.416      0.027       0.570       8.171
x1             2.5022      0.151     16.569      0.000       2.185       2.819
==============================================================================
Omnibus:                        0.333   Durbin-Watson:                   2.055
Prob(Omnibus):                  0.846   Jarque-Bera (JB):                0.489
Skew:                           0.195   Prob(JB):                        0.783
Kurtosis:                       2.341   Cond. No.                         25.0
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly sp
ecified.
```

8.3. Modelin Öyrənmə və Təxmin Prosesi

- İlk addımda verilənləri uyğun şəkildə (lazım gələrsə, preprocessingdən keçirərək) train və test hissələrinə ayırırıq.
- Sonra modeli train verilənləri ilə `fit` edirik.
- Öyrədilmiş model ilə test verilənləri üzərində təxminlər edirik.
- Təxmin nəticələrini qiymətləndiririk.

In [17]:

8.4. Model Qiymətləndirmə və Performans Metrikləri

Təxmin keyfiyyətini ölçmək üçün müxtəlif metriklər istifadə olunur:

- **R^2 (Determinasiya Əmsalı):** Modelin verilənləri izah etmə gücü, 0 ilə 1 arasında dəyər alır, 1 ən yaxşı nəticəni göstərir.
- **MSE (Orta Kvadratik Xəta):** Təxmin xətalarının orta kvadratı, daha kiçik dəyər daha yaxşıdır.
- **MAE (Orta Mütləq Xəta):** Xətaların mütləq ortalaması.

In [ ]:

```
R²: 0.933468340621999
MSE: 18.935690698559867
MAE: 3.791954067798572
```

8.5. Verilənlər Setini Bölmə və Yoxlama

- Modelin real həyatda nə qədər uğurlu olacağını anlamaq üçün verilənləri adətən **təlim (train)** və **test** dəstlərinə bölürük.
- `train_test_split` funksiyası bunun üçün ən çox istifadə olunan üsuldur.
- Həmçinin, modelin ümumiləşdirilməsini yoxlamaq üçün **çarpaz yoxlama (cross-validation)** aparılır; beləliklə, model fərqli alt setlərlə test edilərək performansı ortalanır.

In [22]:

```
CV R² əmsalları: [0.72326447 0.36080518 0.70287298]
Ortalama CV: 0.5956475448468986
```

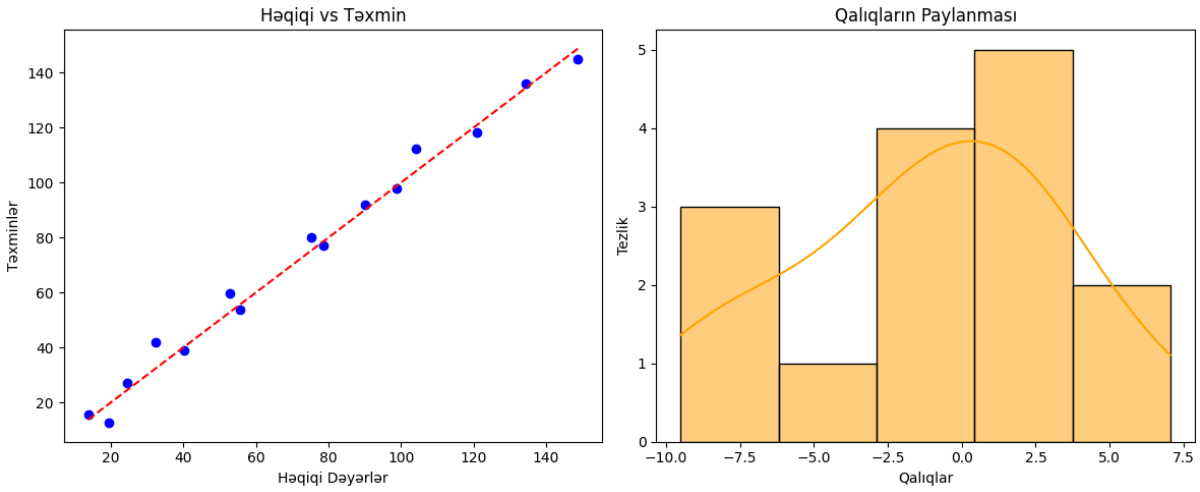
8.6. Reqressiya Nəticələrinin Vizuallaşdırılması

Sonda, model nəticələrini daha yaxşı şərh etmək üçün müxtəlif qrafiklər istifadə olunur:

- **Səpələnmə qrafiki (Scatter plot):** Həqiqi və təxmin edilən dəyərləri müqayisə etmək üçün.
- **Qalıq qrafiki (Residual plot):** Qalıqların paylanmasını görmək üçün.
- **Əmsal qrafikləri:** Xüsusiyyətlərin modelə təsirini göstərmək üçün.

Python-da bu qrafiklər üçün **matplotlib** və **seaborn** kitabxanaları geniş şəkildə istifadə olunur.

In [23]:



📗 9. REAL HƏYATDA TƏTBİQ SAHƏLƏRİ

9.1. Təxmin (Proqnozlaşdırma)

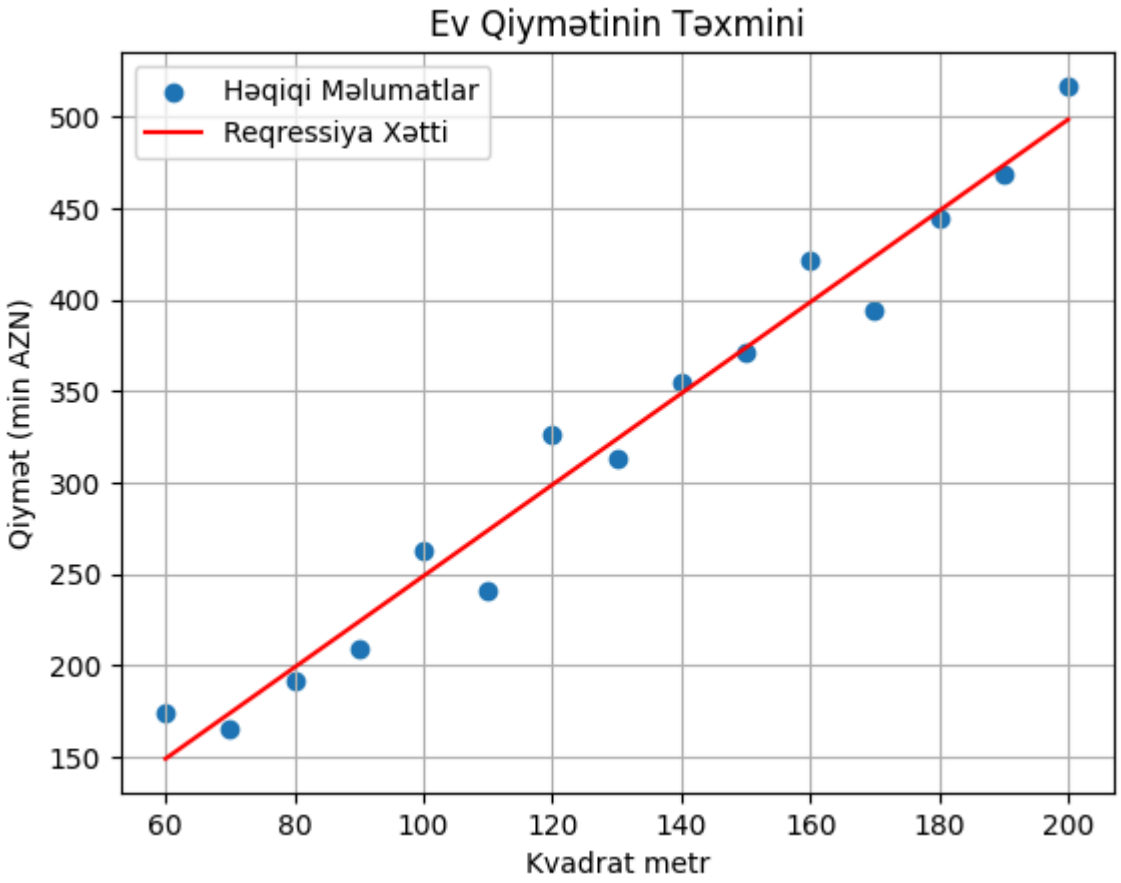
Nümunə:
Bir daşınmaz əmlak şirkəti, bir evin qiymətini təxmin etmək istəyir.

- Evlərin sahəsi (kvadrat metr), otaq sayı, yerləşdiyi rayon kimi məlumatlar toplanır.
- Bu məlumatlarla xətti reqressiya modeli qurulur.
- Model, yeni bir ev üçün təxmini qiyməti hesablayır.

Başqa bir nümunə:
Bir e-ticarət saytı, reklam xərclərinə əsasən satış proqnozu verir.

- Reklam büdcəsi artdıqca satışların necə dəyişdiyini modelləşdirərək gələcək satışları təxmin edir.

In [43]:



9.2. Səbəb-Nəticə Analizi (Causality Analysis)

Nümunə:
Bir iqtisadçı-tədqiqatçı, faiz dərəcələrinin istehlakçı xərclərinə təsirini araşdırır.

- Faiz dərəcəsi dəyişdikcə xərclərin necə dəyişdiyini ölçmək üçün xətti reqressiya qurur.
- Əldə edilən əmsal, faiz dərəcəsindəki bir vahid dəyişikliyin xərclərə təsirini göstərir.

Başqa bir nümunə:
Bir şirkət, işçi təlim proqramının performansa təsirini analiz edir.

- Təlim alan və almayan işçilərin performans məlumatları müqayisə edilir, təlimin təsiri modelləşdirilir.

In [39]:

```
əmsal: -5.462499448158006
Sabit: 154.1007922869477
```

9.3. A/B Test Nəticələrinin Analizi

Nümunə:
Bir veb sayt iki fərqli dizaynı (A və B) test edir.

- İstifadəçilərin hər bir dizaynda keçirdiyi vaxt və ya satınalma nisbətləri ölçülür.
- Xətti reqressiya, dizayn növü və digər dəyişənlərdən istifadə edərək hansı dizaynın daha uğurlu olduğunu göstərir.

Başqa bir nümunə:
Bir mobil tətbiq qiymətini iki fərqli səviyyədə (pulsuz və ödənişli) təklif edir.

- İstifadəçi yükləmə sayıları və gəlir məlumatları təhlil edilərək hansı qiymət strategiyasının daha yaxşı olduğu müəyyən edilir.

In [41]:

```
Dizayn təsiri (B - A): 22.500000000000004
```

9.4. Zaman Reqressiyaları (Trend Modelləşdirmə)

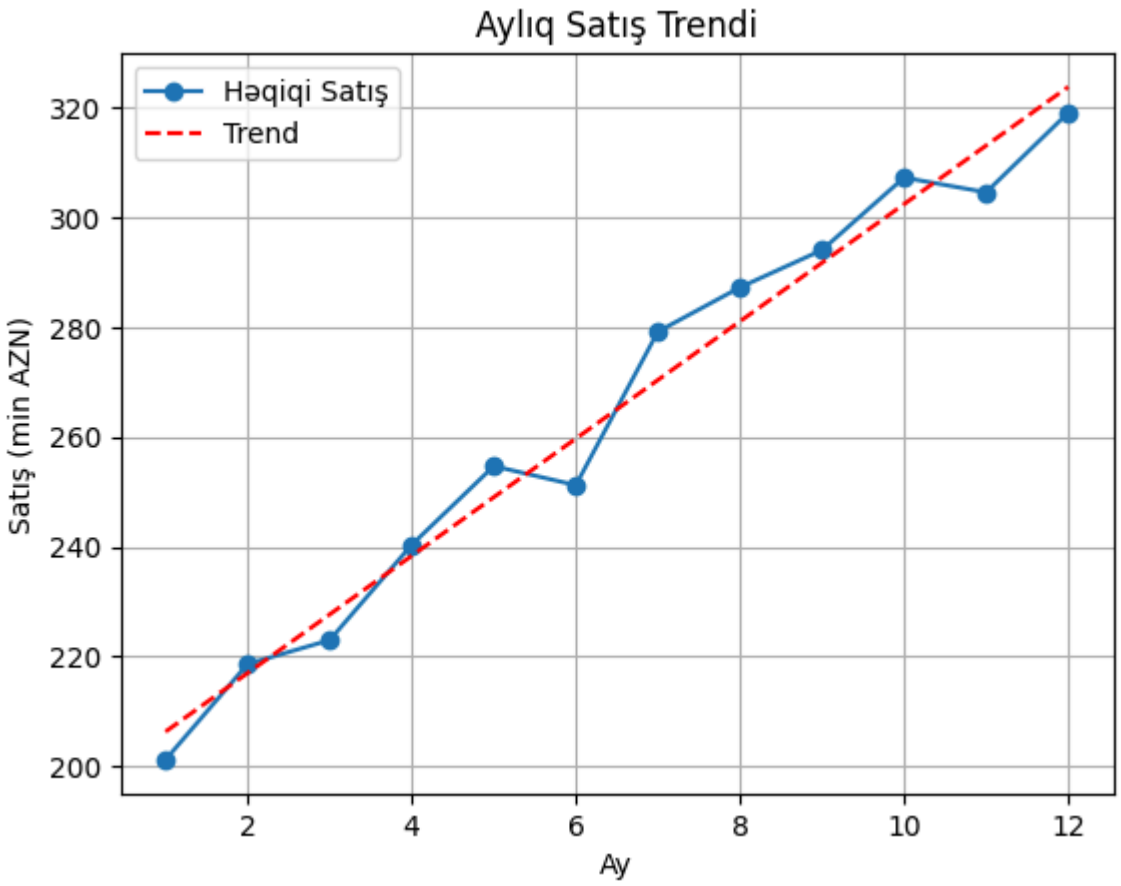
Nümunə:
Bir pərakəndə satış şəbəkəsi, aylıq satışların zamanla necə dəyişdiyini modelləşdirmək istəyir.

- Ay sayı müstəqil dəyişən, satış miqdarı isə asılı dəyişən olur.
- Model, satışlardakı artım və ya azalma trendini göstərir.

Başqa bir nümunə: Meteoroloji məlumatlarla illik orta temperatur dəyişiklikləri araşdırılır.

- İllər üzrə temperatur məlumatları ilə trend təhlil edilir, iqlim dəyişikliyinin təsiri araşdırılır.

In [44]:



📗 10. TEZ-TEZ EDİLƏN SƏHVLƏR

10.1. Kategorik Dəyişənləri Ədədi Kimi İstifadə Etmək

Nümunə:
Bir modeldə "şəhər" dəyişəni birbaşa 1 = Bakı, 2 = Gəncə, 3 = Sumqayıt olaraq verilsə, model bu rəqəmlər arasındakı böyüklük fərqini anlamağa çalışır. Halbuki bu şəhərlərin arasında bir sıralama yoxdur.
Doğrusu: "Şəhər" dəyişəni üçün üç ayrı dummy dəyişən yaradılır: Bakı (0/1), Gəncə (0/1), Sumqayıt (0/1).

In [ ]:

10.2. VIF Yoxlaması Etməmək (Çoxlu Xətti Reqressiya)

Nümunə:
Bir ev qiyməti təxminində həm evin ümumi sahəsi, həm də otaq sayı modelə daxil edilir. Çünki otaqlar artdıqca sahə də ümumiyyətlə artar, buna görə də iki dəyişən yüksək korrelyasiyaya malikdir.
Problem: Modelin əmsalları qeyri-sabit olur, hansı dəyişənin daha təsirli olduğu qeyri-müəyyənləşir.
Həll: VIF hesablanır, yüksək dəyər varsa bir dəyişən çıxarılır və ya birləşdirilir.

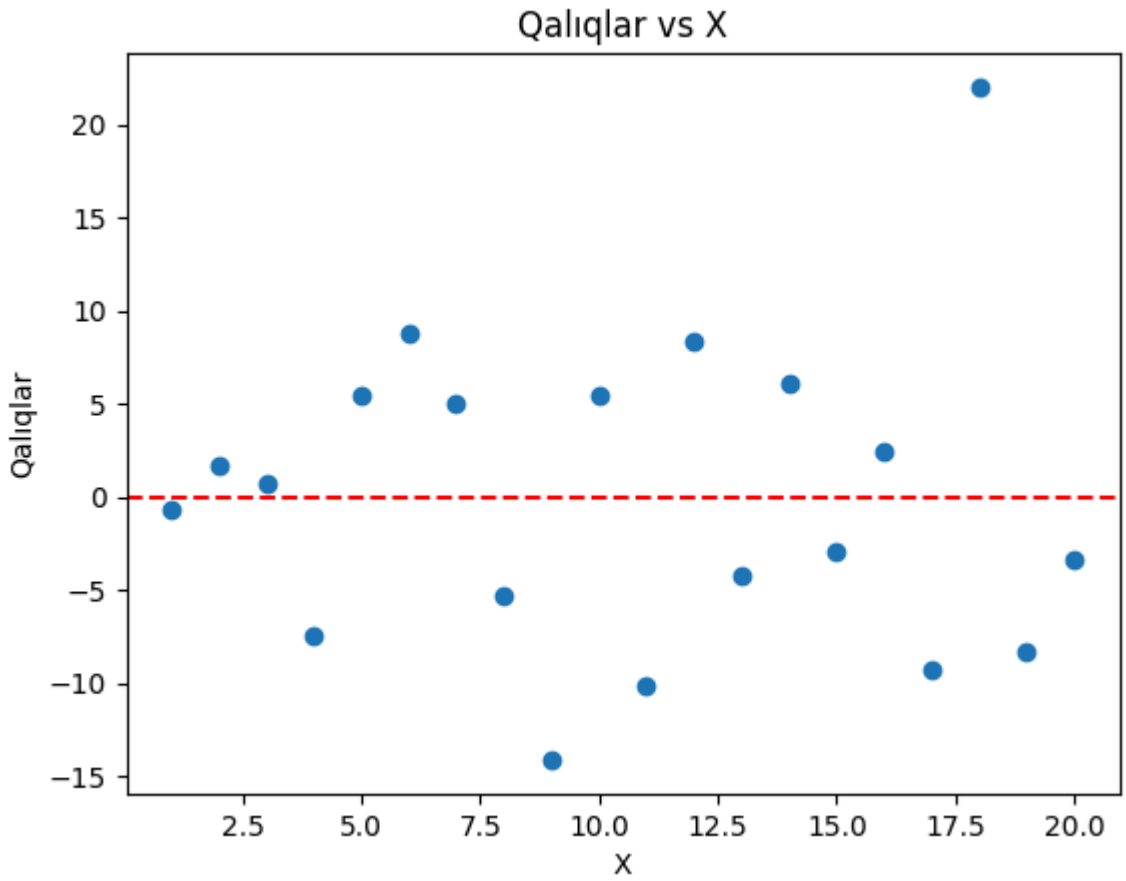
In [ ]:

```
  Dəyişən  VIF Dəyəri
0   const      967.05
1    sahə      242.26
2    otaq      242.26
```

10.3. Hipotezləri Nəzərə Almamaq

Nümunə:
Model qalıqları (təxmin xətaları) müəyyən bir nizam göstərir (məsələn, xəta böyüklüyü asılı dəyişən artdıqca artır). Bu, modelin sabit dispersiya (homoskedastiklik) fərziyyəsini pozur.
Nəticə: Model təxminləri etibarlı deyil, yanlış nəticələr çıxarıla bilər.
Həll: Qalıqların qrafiki araşdırılır, lazım gələrsə fərqli modelləmə və ya çevrilmələr aparılır.

In [69]:



10.4. Test Setindən İstifadə Etmədən Bütün Verilənləri Təlim Etmək

Nümunə:
Bir tələbə əlindəki bütün verilənləri istifadə edərək modeli train edir və performansını da eyni verilənlərdə test edir.
Nəticə: Model verilənləri əzbərlədiyi üçün çox yaxşı nəticə göstərir, ancaq yeni verilənlərdə uğursuz olur (overfitting).
Həll: Verilənlər train və test olaraq bölünür, model əvvəlcə təlimdə öyrənir, sonra testdə sınanır.

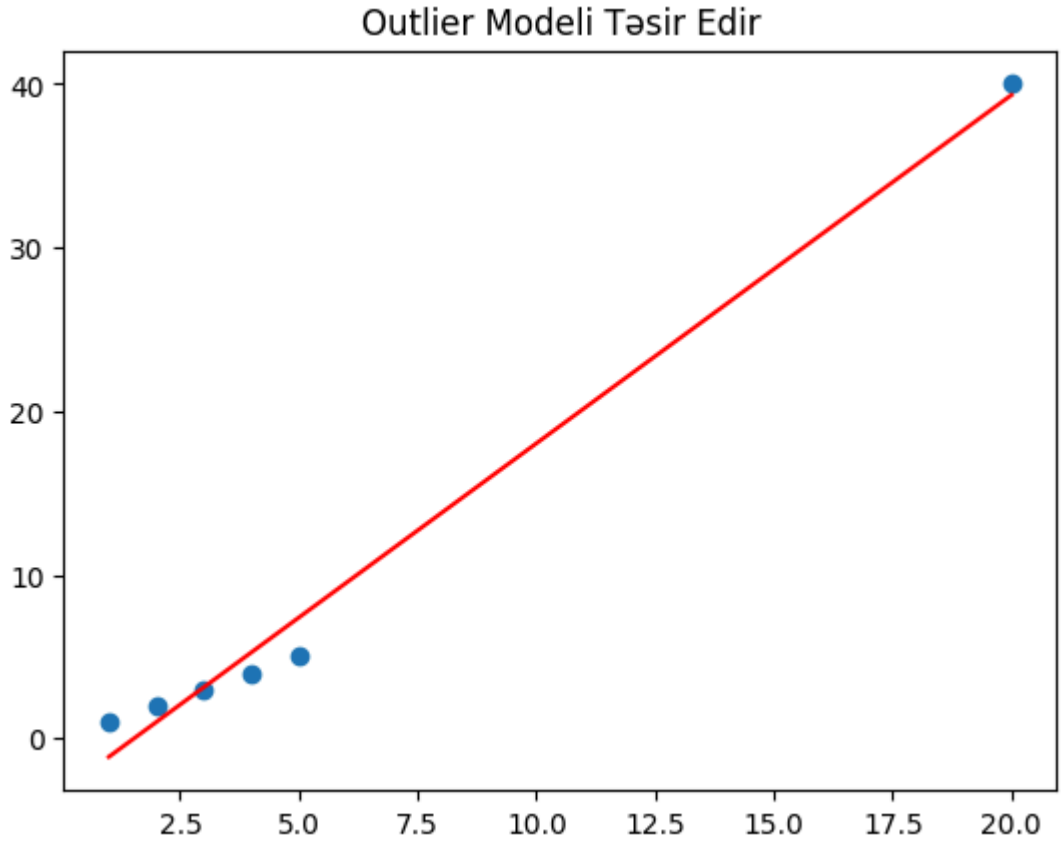
In [70]:

```
Həqiqi R² (Test seti): 0.5850338140225688
```

10.5. Outlier'ları Təmizləməmək

Nümunə:
Bir mağazanın satış verilənlərində nadir hallarda həddindən artıq yüksək satışlar (məsələn, böyük bir kampaniya günü) var.
Problem: Bu kənar dəyərlər model əmsallarını və təxminlərini həddindən artıq təsir edə bilər.
Həll: Bu kənar dəyərlər aşkar edilib, lazım gələrsə modelə xüsusi olaraq daxil edilir və ya təmizlənir.

In [ ]:



Out[ ]:

```
▾ LinearRegression i ?
(https://scikit-learn.org/1.6/modules/generated/sklearn.linear_model.LinearRegress
LinearRegression()
```

10.6. Bütün Dəyişənləri Avtomatik İstifadə Etmək

Nümunə:

Əlinizdə bir çox dəyişən var deyə hamısını modelə daxil etmək geniş yayılmış bir səhvdir.

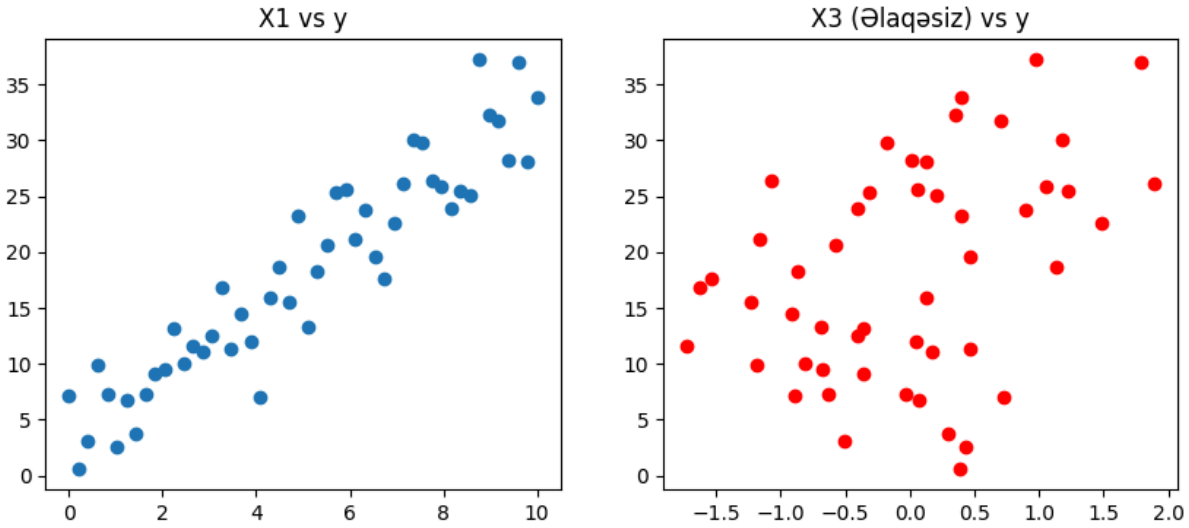
Bəzi dəyişənlər asılı dəyişənlə heç bir əlaqəsi olmaya bilər.

Lazımsız dəyişənlər modelin mürəkkəbləşməsinə və həddindən artıq öyrənməyə (overfitting) səbəb olur.

Problem: Model lazımsız dərəcədə mürəkkəb olur, şərhi çətinləşir, performans düşə bilər.

Həll: Vacib dəyişənlər seçilməli, lazımsız dəyişənlər modeldən çıxarılmalıdır.

In [ ]:



10.7. Qeyri-xətti Əlaqələri Xətti Modellə Tutmağa Çalışmaq

Nümunə:

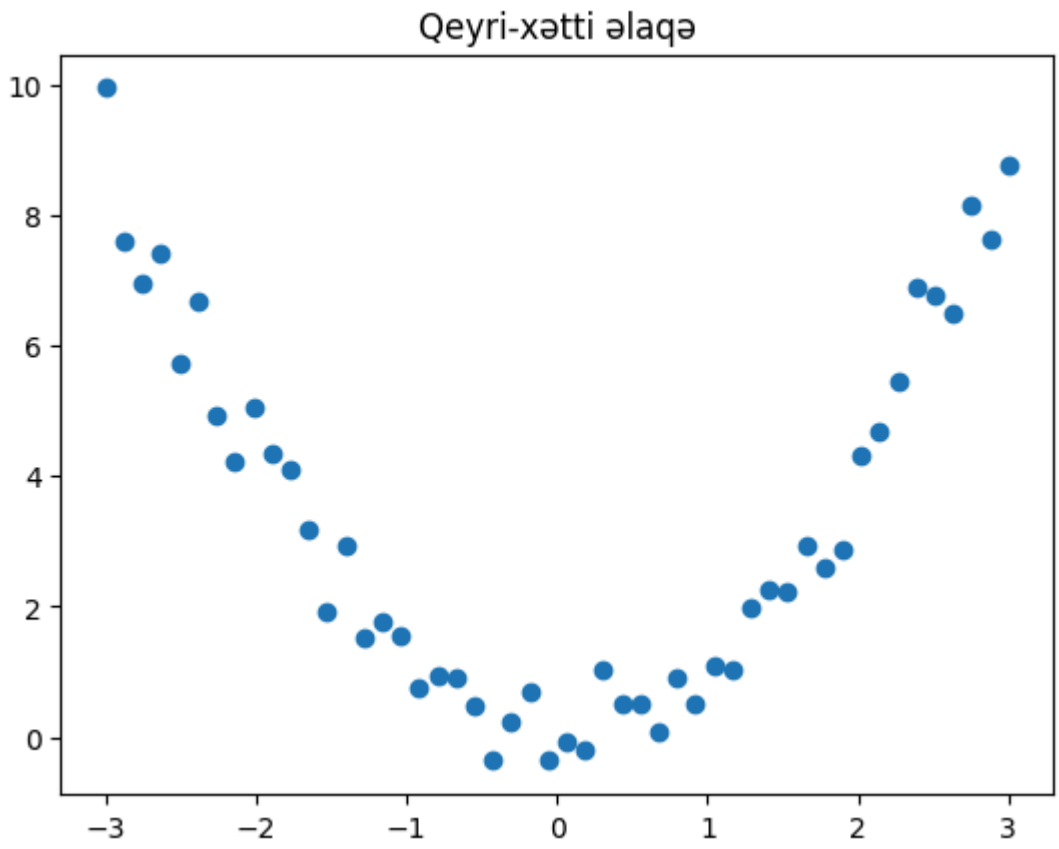
Asılı və müstəqil dəyişənlər arasında U şəklində (qeyri-xətti) bir əlaqə vardır.

Xətti reqressiya bu əlaqəni xətti xətlə təxmin etməyə çalışır, xəta yüksək olur.

Problem: Model verilənləri yaxşı izah edə bilmir, təxminlərdə yanılma olur.

Həll: Polinom terminlər əlavə etmək və ya qeyri-xətti modellər istifadə etmək lazımdır.

In [71]:



10.8. Miqyaslamanı (Scaling) Unutmaq (Requlyarizasiyalı Modellərdə)

Nümunə:

Bir dəyişən maaş kimi böyük rəqəmlərdən ibarət olarkən, digəri yaş kimi kiçik rəqəmlərdədir.

Miqyaslama aparılmasa, model böyük rəqəmlərə malik dəyişəni daha vacib hesab edəcək.

Problem: Model yanlış əmsallar öyrənir, dəyişən təsirləri qarışır.

Həll: Standartlaşdırma (standardization) və ya miqyaslama aparılmalıdır.

In [ ]:

```
Miqyaslama olmadan əmsallar: [2.97779543e+00 9.04726431e-04]
Miqyaslama ilə əmsallar: [3.33679406 0.7682216 ]
```

10.9. Zəif Mənalı Əmsalların Şərhi

Nümunə:

Bir dəyişənin p-dəyəri 0.7 kimi yüksək dəyərlər alırsa, bu dəyişənin təsiri statistik olaraq mənalı deyil.

Problem: Mənalı olmayan dəyişənlərə məna yükləmək yanıldıcı olacaq.

Həll: Yalnız statistik olaraq mənalı (məsələn, $p < 0.05$) dəyişənlər şərh edilməlidir.

10.10. Qalıqların Normal Paylanmasını Nəzərə Almamaq

Nümunə: Model qalıqları əyri və ya anormal bir paylanmaya malikdir.

Bu vəziyyət, xüsusilə inam aralıqları (confidence interval) və fərziyyə testlərinin etibarlılığını azaldır.

Problem: Yanlış nəticələrə və model xətalarına səbəb olur.

Həll: Qalıqların paylanması araşdırılmalı, lazım gələrsə transformasiya və ya alternativ modellər istifadə edilməlidir.

In [72]: