MBAN 5110: PREDICTIVE MODELING

SESSION 3: LINEAR REGRESSION

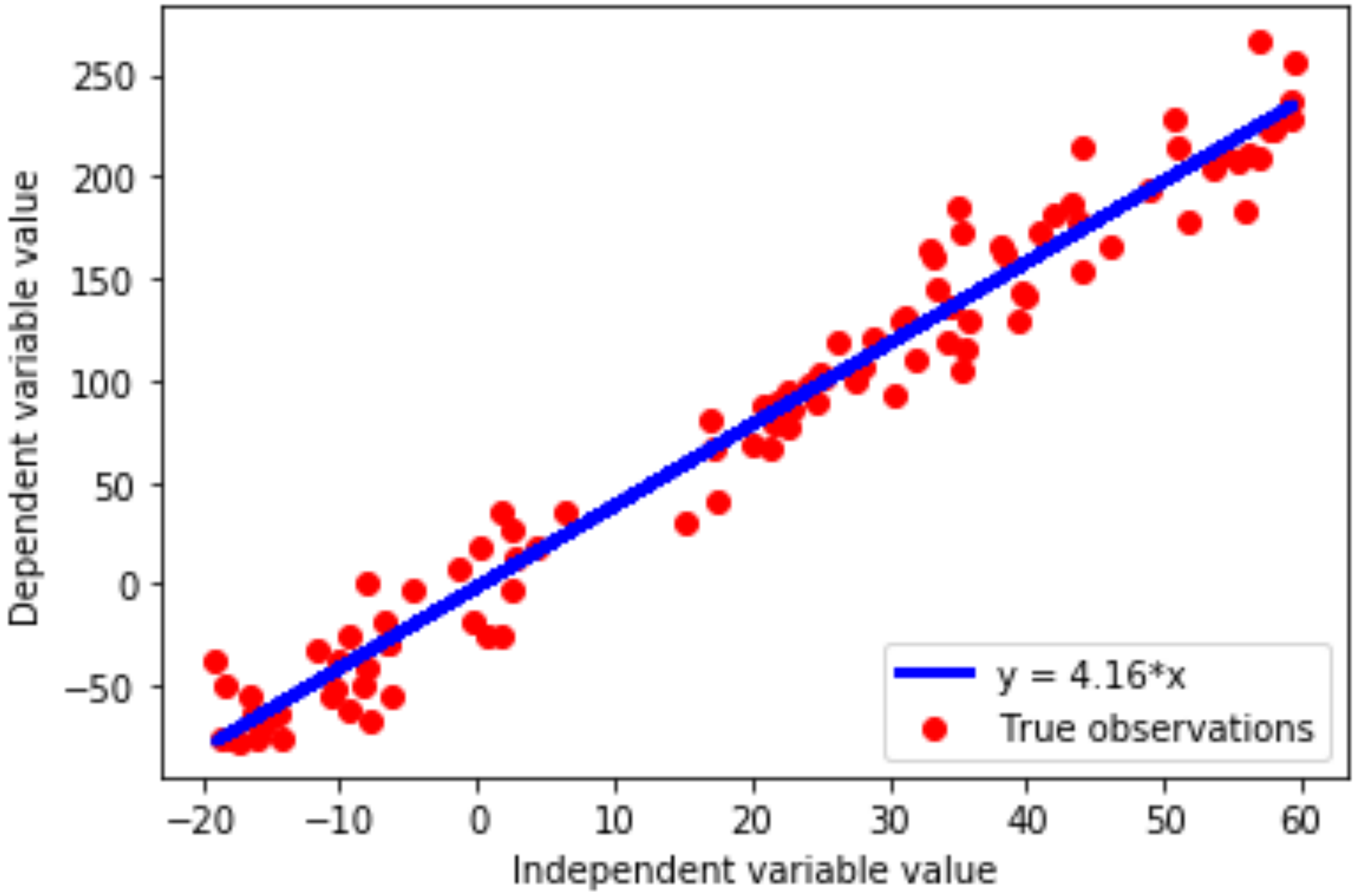
DR. ISIK BICER

Schulich School of Business | YORK UNIVERSITÉ UNIVERSITY

TODAY'S AGENDA

- Linear Regression: Model and Assumptions
- Maximum Likelihood Estimation
- Ordinary Least Squares
- Generalized Least Squares

LINEAR MODELS



ANALYTICAL REPRESENTATION

- y : Dependent variable (m observations)
- x : Independent variables
- ϵ : Error terms
- β : Coefficients

$$
\begin{gathered}
y_1 = \beta_0 + \beta_1 x_{11} + \beta_2 x_{12} + \beta_3 x_{13} + \cdots + \beta_n x_{1n} + \epsilon_1 \\
y_2 = \beta_0 + \beta_1 x_{21} + \beta_2 x_{22} + \beta_3 x_{23} + \cdots + \beta_n x_{2n} + \epsilon_2 \\
\vdots \\
y_m = \beta_0 + \beta_1 x_{m1} + \beta_2 x_{m2} + \beta_3 x_{m3} + \cdots + \beta_n x_{mn} + \epsilon_m
\end{gathered}
$$

ASSUMPTIONS

- There is a linear relationship between y and x
- The error term ϵ follows a normal distribution with zero mean and a standard deviation of σ
- The explanatory variables (i.e., x terms) are independent from the error term so they are also referred to as independent variables
- The error term is homoscedastic such that it has a fixed variance

MAXIMUM LIKELIHOOD ESTIMATION

- Suppose $y = f(x) + \epsilon$ is a linear model
  - $\epsilon = y - f(x) \sim N(0, \sigma)$
  - Normal dist. pdf: $\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\epsilon^2}{2\sigma^2}}$
  - The likelihood for a single observation i is:

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - f(x_i))^2}{2\sigma^2}}$$

  - The likelihood for all observations

$$L(y, x|\sigma) = \prod_{i=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - f(x_i))^2}{2\sigma^2}}$$

MAXIMUM LIKELIHOOD ESTIMATION

- The common practice in MLE is to transform the likelihood function to log-likelihood for exponential family functions:

$$l(y,x|\sigma) = \ln(L(y,x|\sigma) = -m\ln(\sigma) - \frac{m}{2}\ln(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{m}(y_i - f(x_i))^2$$

- While fitting the coefficients for $f(x)$, we aim to minimize the term: $\sum_{i=1}^{m}(y_i - f(x_i))^2$

MATRIX FORM

$$Y = \begin{vmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{vmatrix}; \quad \mathrm{B} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}; \quad X = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1n} \\ 1 & X_{21} & X_{22} & & X_{2n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix}; \mathrm{E} = \begin{vmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{vmatrix}$$

- Transpose of a matrix:

$$\mathrm{E}^T = |\epsilon_1, \epsilon_2, \dots, \epsilon_m|$$

$$X^T = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ X_{11} & X_{21} & X_{31} & & X_{m1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ X_{1n} & X_{2n} & X_{3n} & \cdots & X_{mn} \end{bmatrix}$$

ORDINARY LEAST SQUARES (OLS) ESTIMATION

Schulich
School of Business
York University

$$Y = \begin{vmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{vmatrix}; \quad \mathrm{B} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}; \quad X = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1n} \\ 1 & X_{21} & X_{22} & & X_{2n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix}; \mathrm{E} = \begin{vmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_m \end{vmatrix}$$

- $Y = X\mathrm{B} + \mathrm{E}$
- Sum of square of errors:

$$\mathrm{E}^T\mathrm{E} = (Y - X\mathrm{B})^T(Y - X\mathrm{B}) = Y^TY - Y^TX\mathrm{B} - (X\mathrm{B})^TY + (X\mathrm{B})^TX\mathrm{B}$$

$$\mathrm{E}^T\mathrm{E} = Y^TY - 2Y^TX\mathrm{B} + \mathrm{B}^TX^TX\mathrm{B}$$

$$\frac{\partial \mathrm{E}^T\mathrm{E}}{\partial \mathrm{B}} = -2X^TY + 2X^TX\mathrm{B} = 0$$

$$\mathrm{B} = (X^TX)^{-1}X^TY$$

CONSISTENCY OF OLS ESTIMATION

- X and E should be independent
  - Error terms should be orthogonal to independent variables
  - In mathematical terms:

$$X^T \mathrm{E} = 0$$

$$X^T (Y - X\mathrm{B}) = 0$$

$$X^T Y - X^T X B = 0$$

$$X^T Y = X^T X B$$

$$(X^T X)^{-1} X^T Y = (X^T X)^{-1} X^T X B$$

$$\mathrm{B} = (X^T X)^{-1} X^T Y$$

- So, we reach the same result from the orthogonality condition.

ENDOGENEITY PROBLEMS

- The endogeneity problems occur when X and E are independent:

$$X^T\mathrm{E} = \delta$$

$$X^T(Y - X\mathrm{B}) = \delta$$

$$X^TY - X^TXB = \delta$$

$$X^TY - \delta = X^TXB$$

$$(X^TX)^{-1}(X^TY - \delta) = (X^TX)^{-1}X^TXB$$

$$\mathrm{B} = (X^TX)^{-1}X^TY - (X^TX)^{-1}\delta$$

- The OLS estimation is then biased by $(X^TX)^{-1}\delta$

ENDOGENEITY PROBLEMS

- Omitted variables also cause endogeneity problems
- True model: $Y = X\mathrm{B} + \theta Z + \mathrm{E}$
- θ is constant. It is the coefficient of Z
- The omitted variable: $Z = \begin{vmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{vmatrix}$

ENDOGENEITY PROBLEMS

$$X^T \mathrm{E} = 0$$

$$X^T(Y - X\mathrm{B} - \theta Z) = 0$$

$$X^T Y - X^T X B - X^T \theta Z = 0$$

$$X^T Y - X^T \theta Z = X^T X B$$

$$(X^T X)^{-1}(X^T Y - X^T \theta Z) = (X^T X)^{-1} X^T X B$$

$$\mathrm{B} = (X^T X)^{-1} X^T Y - (X^T X)^{-1} X^T \theta Z$$

- The OLS estimation is biased by $(X^T X)^{-1} X^T \theta Z$

VARIANCE OF COEFFICIENTS

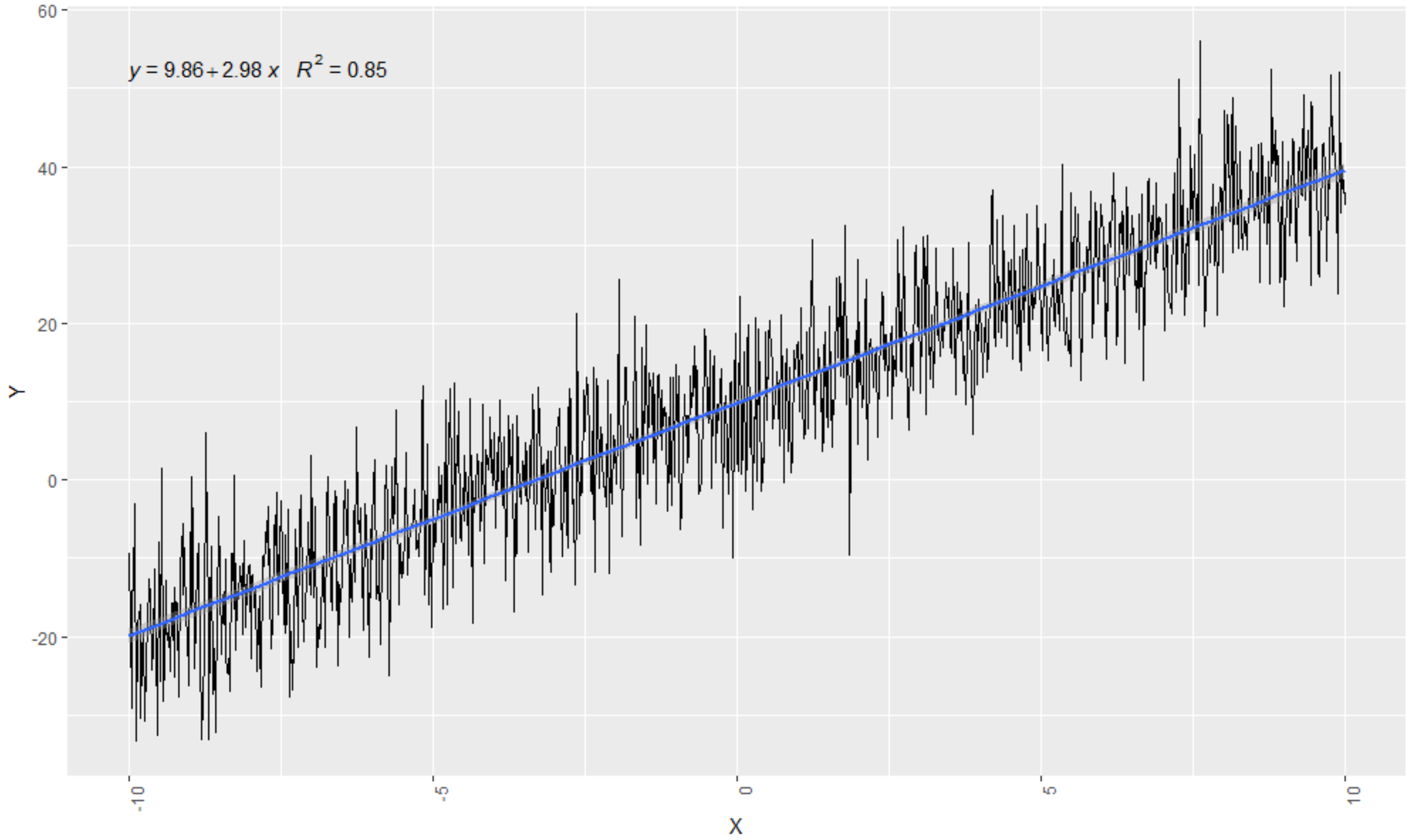
$$Var(\mathrm{B}) = (X^T X)^{-1} X^T Var(\mathrm{E}) X (X^T X)^{-1}$$

- **Assumption:** $Var(\mathrm{E}) = \sigma^2$ such that the variance of residuals do not change for varying X values. Then,

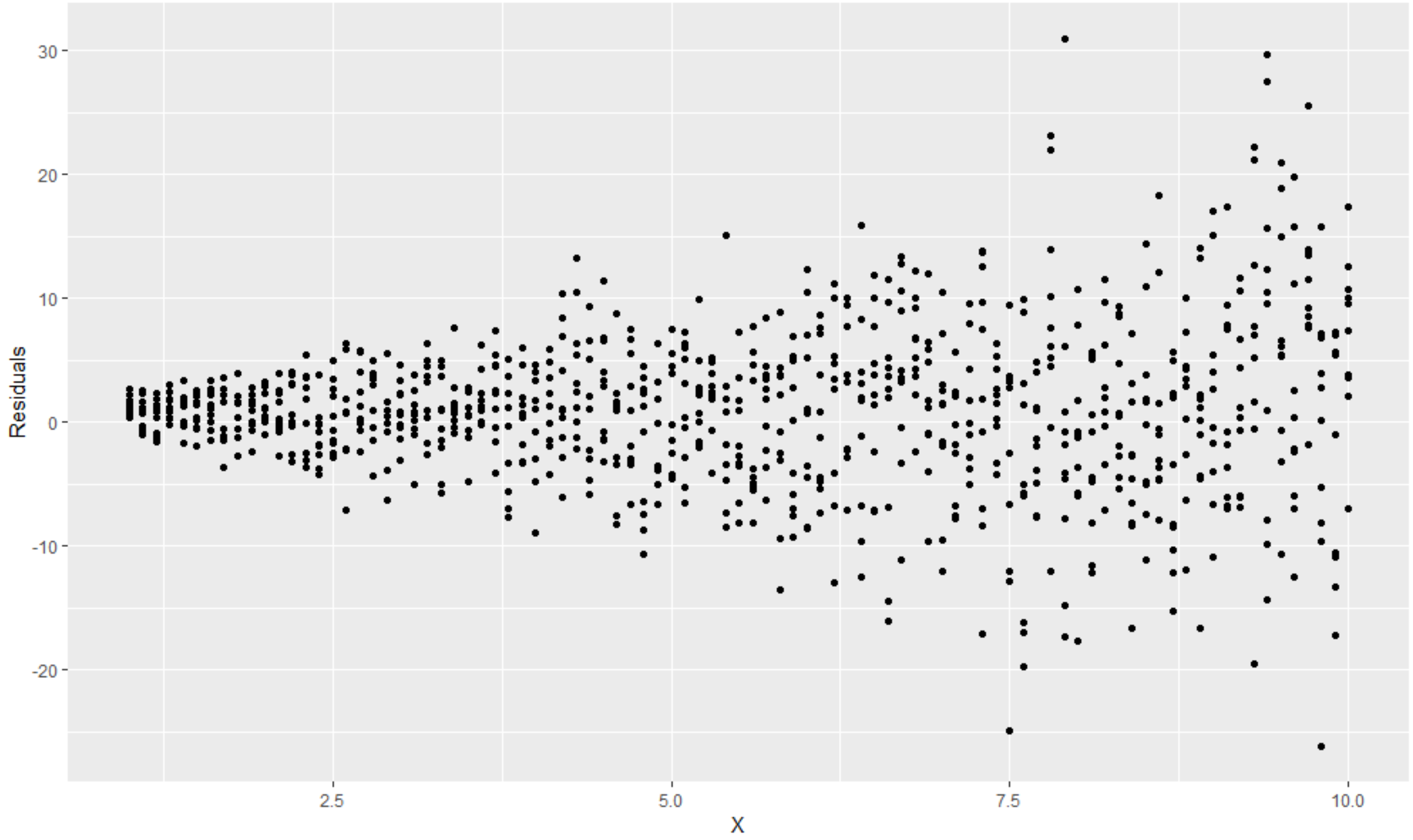
$$Var(\mathrm{B}^T) = \sigma^2 (X^T X)^{-1}$$

- This assumption is the condition of homoscedasticity.

HOMOSCEDASTICITY



HETEROSCEDASTICITY



- Assumption of Linear Regression based on OLS
  - Homescedasticity: $Var(\mathrm{E}) = \sigma^2$
- Violated→ Heteroscedasticity : $Var(\mathrm{E}) = \sigma^2 f(X)$

TESTING FOR HETEROSCEDASTICITY

- Breusch-Pagan (BP) test

$$\mathrm{E}^T\mathrm{E} = AX + \Upsilon$$

- If $\mathrm{A} \neq 0$, variance of residuals depends on X
  - Then, the model is heteroscedastic.
- BP test checks if $\mathrm{A} \neq 0$.
  - Null hypothesis: $\mathrm{A} = 0$ so the model is homoskedastic
  - *If p-value is above 0,05, we cannot reject the null hypothesis and **rely on OLS estimates***
- In Python: statsmodels library
  - statsmodels.stats.api.breuschpagan()

AUTOCORRELATION

- Standard correlation defined as

$$\rho_{zq} = \frac{Cov(z,q)}{\sigma_z \sigma_q}$$

- When we measure a parameter over time (panel data), the parameter may have an autocorrelation at different lags.

$$\hat{\rho}_{\epsilon_i \epsilon_{i-p}} = \frac{Cov\left(\epsilon_i, \epsilon_{i-p}\right)}{\sigma_{\epsilon_i}^2} = \frac{\sum_{i=p+1}^{m} (\epsilon_i \epsilon_{i-p})}{\sum_{i=1}^{m} (\epsilon_i \epsilon_i)}$$

AUTOCORRELATION

- Residuals sometimes have autocorrelation
  - For example, ϵ_i may correlate with ϵ_{i-p} for some p values.
- Remember:

$$Var(\mathrm{B}) = (X^T X)^{-1} X^T Var(\mathrm{E}) X (X^T X)^{-1}$$

- If there is autocorrelation,

$$Var(\mathrm{E}) \neq \sigma^2$$

- Covariance matrix of residuals:

$$Var(\mathrm{E}) = \Psi = \sigma^2 \begin{bmatrix} \rho_0 & \rho_1 & \rho_2 & \cdots & \rho_m \\ \rho_1 & \rho_0 & \rho_1 & & \rho_{m-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho_m & \rho_{m-1} & \rho_{m-2} & \cdots & \rho_0 \end{bmatrix}$$

GENERALIZED LEAST SQUARES

- Suppose $Y = XB + \mathrm{E}$ and the variance of error terms is not constant:
  - $Var(\mathrm{E}) = \sigma^2\Omega$
  - Ω is a symmetric matrix such that $\omega^T\omega = \Omega$
  - ω is also symmetric: $\omega^T = \omega$
- The basic idea of GLS
  - Multiply both sides of the regression equation with ω^{-1}
  - Then, $Var(\omega^{-1}\mathrm{E}) = \sigma^2 I$

GENERALIZED LEAST SQUARES

$$\omega^{-1}Y = \omega^{-1}X\mathrm{B} + \omega^{-1}\mathrm{E}$$

$$(\omega^{-1}\mathrm{E})^T\omega^{-1}\mathrm{E} = (\omega^{-1}Y - \omega^{-1}X\mathrm{B})^T(\omega^{-1}Y - \omega^{-1}X\mathrm{B})$$

$$= Y^T\omega^{-1}\omega^{-1}Y - 2Y^T\omega^{-1}\omega^{-1}X\mathrm{B} + \mathrm{B}^TX^T\omega^{-1}\omega^{-1}X\mathrm{B}$$

- Taking the derivative of the last expression and making it equal to zero, we obtain:

$$0 = -2(X^T\Omega^{-1}Y) + 2(X^T\Omega^{-1}X)\mathrm{B}$$

$$\mathrm{B} = (X^T\Omega^{-1}X)^{-1}X^T\Omega^{-1}Y$$

OLS VS. GLS

OLS	GLS
$\mathrm{B} = (X^T X)^{-1} X^T Y$	$\mathrm{B} = (X^T \Omega^{-1} X)^{-1} X^T \Omega^{-1} Y$
$Var(\mathrm{B}) = \sigma^2 (X^T X)^{-1}$	$Var(\mathrm{B}) = \sigma^2 (X^T \Omega^{-1} X)^{-1}$

OLS IN PYTHON

- The statmodels library includes packages for various statistical models (www.statsmodels.org)
- A non-exhaustive list
  - OLS
  - GLS
  - Time series
  - Discrete choice models
  - State space models

OLS IN PYTHON

- Generating the variables and constructing the model

$$y_i = 3 + 4x_i + \epsilon_i$$

```
import numpy as np
import pandas as pd
import matplotlib as mp
import statsmodels.api as sm
```

```
mu, sigma = 0, 5 # mean and standard deviation of normal distribution for the error term
x = np.random.uniform(40,80,100)
epsilon = np.random.normal(mu,sigma,100)
y = 3 + 4*x + epsilon
```

OLS IN PYTHON

```
model_reg = sm.OLS(y,x).fit()
model_reg.summary()
```

OLS Regression Results

Dep. Variable:	y	**R-squared (uncentered):**	1.000
Model:	OLS	**Adj. R-squared (uncentered):**	1.000
Method:	Least Squares	**F-statistic:**	2.157e+05
Date:	Sun, 25 Sep 2022	**Prob (F-statistic):**	4.45e-167
Time:	10:31:54	**Log-Likelihood:**	-308.07
No. Observations:	100	**AIC:**	618.1
Df Residuals:	99	**BIC:**	620.7
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
x1	4.0379	0.009	464.475	0.000	4.021	4.055

No intercept

Omnibus:	0.299	**Durbin-Watson:**	1.771
Prob(Omnibus):	0.861	**Jarque-Bera (JB):**	0.102
Skew:	-0.072	**Prob(JB):**	0.950
Kurtosis:	3.058	**Cond. No.**	1.00

OLS IN PYTHON

- Python works with the matrix principle

$$X = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1n} \\ 1 & X_{21} & X_{22} & & X_{2n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & X_{m1} & X_{m2} & \cdots & X_{mn} \end{bmatrix}$$

Remember the column with ones

OLS IN PYTHON

```
x_updated = sm.add_constant(x)
x_updated
```

```
array([[ 1.        , 55.69095773],
       [ 1.        , 59.9960433 ],
       [ 1.        , 62.60411587],
       [ 1.        , 46.2488649 ],
       [ 1.        , 55.61509709],
       [ 1.        , 79.92455652],
       [ 1.        , 44.61432769],
       [ 1.        , 42.81059865],
       [ 1.        , 58.88386557],
       [ 1.        , 69.55063262],
       [ 1.        , 62.87108642],
       [ 1.        , 75.12183828],
       [ 1.        , 64.29944042],
       [ 1.        , 49.85570667],
       [ 1.        , 40.89602389],
       [ 1.        , 77.76653301],
```

OLS IN PYTHON

```
x_updated = sm.add_constant(x)
model_updated = sm.OLS(y,x_updated).fit()
model_updated.summary()
```

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.987
Model:	OLS	**Adj. R-squared:**	0.987
Method:	Least Squares	**F-statistic:**	7643.
Date:	Sun, 25 Sep 2022	**Prob (F-statistic):**	8.45e-95
Time:	10:38:20	**Log-Likelihood:**	-307.73
No. Observations:	100	**AIC:**	619.5
Df Residuals:	98	**BIC:**	624.7
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.2556	2.788	0.809	0.420	-3.277	7.788
x1	4.0015	0.046	87.425	0.000	3.911	4.092

GLS IN PYTHON

- Let's first make the error term autocorrelated

```
# We now generate autocorrelated error terms
epsilon[0] = np.random.normal(mu,sigma,1)
for i in range(0,99):
    epsilon[i+1]=0.4*epsilon[i]+0.6*np.random.normal(mu,sigma,1)
```

```
y = 3 + 4*x + epsilon
```

- Each error term has an autocorrelation with the previous one

GLS IN PYTHON

- What if we use OLS instead of GLS

```
x_updated = sm.add_constant(x)
model_OLS = sm.OLS(y,x_updated).fit()
model_OLS.summary()
```

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.994
Model:	OLS	**Adj. R-squared:**	0.994
Method:	Least Squares	**F-statistic:**	1.668e+04
Date:	Sun, 25 Sep 2022	**Prob (F-statistic):**	2.92e-111
Time:	10:52:07	**Log-Likelihood:**	-268.82
No. Observations:	100	**AIC:**	541.6
Df Residuals:	98	**BIC:**	546.8
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	3.1394	1.889	1.662	0.100	-0.609	6.888
x1	4.0055	0.031	129.146	0.000	3.944	4.067

GLS IN PYTHON

- The covariance matrix of residuals:

$$\Psi = \sigma^2 \begin{bmatrix} \rho_0 & \rho_1 & \rho_2 & \cdots & \rho_m \\ \rho_1 & \rho_0 & \rho_1 & & \rho_{m-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho_m & \rho_{m-1} & \rho_{m-2} & \cdots & \rho_0 \end{bmatrix}$$

- It is symmetric and has a special structure
- Its structure is the same as Toeplitz matrix
- Python needs the second part as input:

$$\begin{bmatrix} \rho_0 & \rho_1 & \rho_2 & \cdots & \rho_m \\ \rho_1 & \rho_0 & \rho_1 & & \rho_{m-1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho_m & \rho_{m-1} & \rho_{m-2} & \cdots & \rho_0 \end{bmatrix}$$

GLS IN PYTHON

```
from scipy.linalg import toeplitz
toeplitz(np.array([1,0.5,0,0,0,0,0,0]))
```

```
array([[1. , 0.5, 0. , 0. , 0. , 0. , 0. , 0. ],
       [0.5, 1. , 0.5, 0. , 0. , 0. , 0. , 0. ],
       [0. , 0.5, 1. , 0.5, 0. , 0. , 0. , 0. ],
       [0. , 0. , 0.5, 1. , 0.5, 0. , 0. , 0. ],
       [0. , 0. , 0. , 0.5, 1. , 0.5, 0. , 0. ],
       [0. , 0. , 0. , 0. , 0.5, 1. , 0.5, 0. ],
       [0. , 0. , 0. , 0. , 0. , 0.5, 1. , 0.5],
       [0. , 0. , 0. , 0. , 0. , 0. , 0.5, 1. ]])
```

GLS IN PYTHON

```
rho = 0.4
cov_matrix = sigma**2*toeplitz(np.append([1,rho],np.zeros(98)))
sm.GLS(y,x_updated,cov_matrix).fit().summary()
```

GLS Regression Results

Dep. Variable:	y	**R-squared:**	0.997
Model:	GLS	**Adj. R-squared:**	0.997
Method:	Least Squares	**F-statistic:**	3.188e+04
Date:	Sun, 25 Sep 2022	**Prob (F-statistic):**	5.51e-125
Time:	11:12:50	**Log-Likelihood:**	-253.63
No. Observations:	100	**AIC:**	511.3
Df Residuals:	98	**BIC:**	516.5
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.1123	1.409	3.628	0.000	2.316	7.909
x1	3.9723	0.022	178.538	0.000	3.928	4.016

GLS VS OLS

```
rho = 0.4
cov_matrix = sigma**2*toeplitz(np.append([1,rho],np.zeros(98)))
sm.GLS(y,x_updated,cov_matrix).fit().summary()
```

GLS Regression Results

Dep. Variable:	y	**R-squared:**	0.997
Model:	GLS	**Adj. R-squared:**	0.997
Method:	Least Squares	**F-statistic:**	3.188e+04
Date:	Sun, 25 Sep 2022	**Prob (F-statistic):**	5.51e-125
Time:	11:12:50	**Log-Likelihood:**	-253.63
No. Observations:	100	**AIC:**	511.3
Df Residuals:	98	**BIC:**	516.5
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	5.1123	1.409	3.628	0.000	2.316	7.909
x1	3.9723	0.022	178.538	0.000	3.928	4.016

```
x_updated = sm.add_constant(x)
model_OLS = sm.OLS(y,x_updated).fit()
model_OLS.summary()
```

OLS Regression Results

Dep. Variable:	y	**R-squared:**	0.994
Model:	OLS	**Adj. R-squared:**	0.994
Method:	Least Squares	**F-statistic:**	1.668e+04
Date:	Sun, 25 Sep 2022	**Prob (F-statistic):**	2.92e-111
Time:	10:52:07	**Log-Likelihood:**	-268.82
No. Observations:	100	**AIC:**	541.6
Df Residuals:	98	**BIC:**	546.8
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	3.1394	1.889	1.662	0.100	-0.609	6.888
x1	4.0055	0.031	129.146	0.000	3.944	4.067