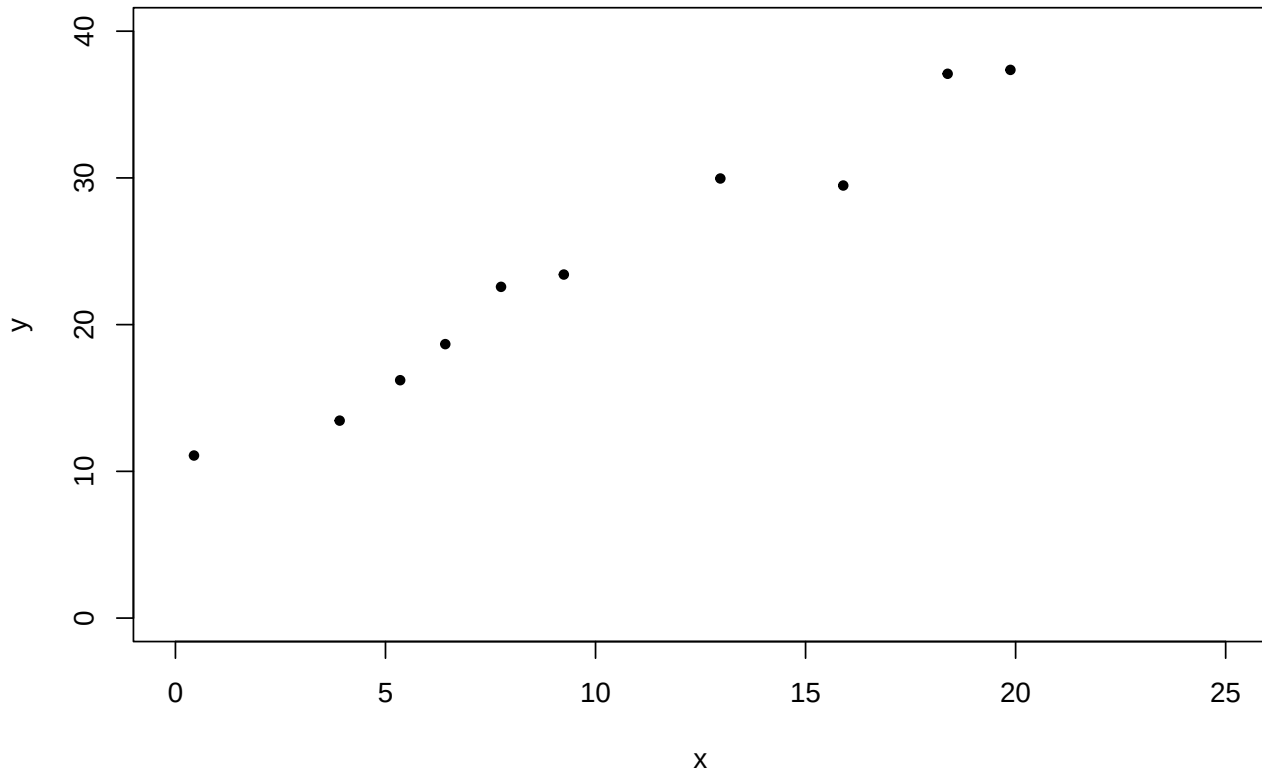


MATH 204: PRINCIPLES OF STATISTICS 2  
WINTER 2026  
COMPLEXITY AND MODEL SELECTION

Consider the following data set with  $n = 10$ , where we have a single predictor  $x$  for response  $y$



We might consider that a straight line model might be appropriate, and can obtain the parameter estimates using the `lm` function.

```
> fit1<-lm(y~x)
> summary(fit1)

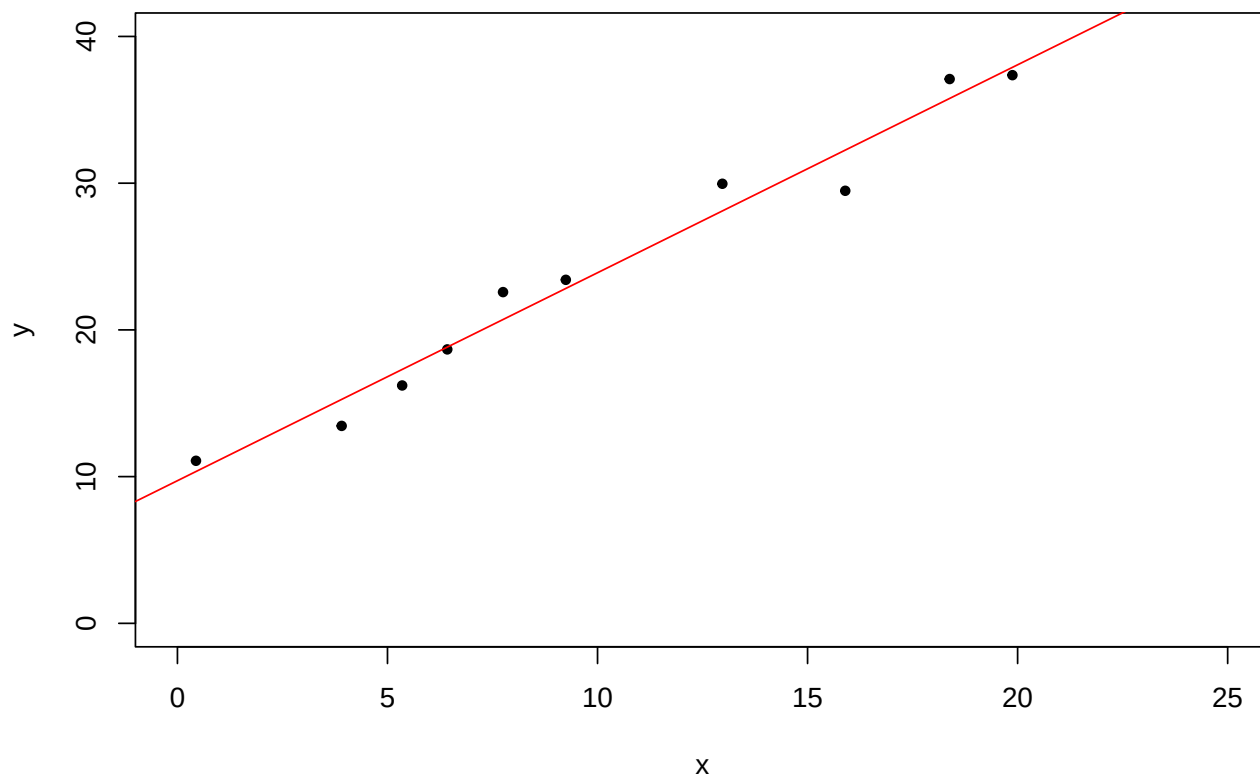
+
+ Call:
+ lm(formula = y ~ x)
+
+ Residuals:
+    Min       1Q   Median       3Q      Max
+ -2.772 -0.953  0.216  1.174  1.867
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)   9.7226     1.0011    9.71  1.1e-05 ***
+ x             1.4173     0.0851   16.66  1.7e-07 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 1.66 on 8 degrees of freedom
```

```

+ Multiple R-squared:  0.972, Adjusted R-squared:  0.968
+ F-statistic:  278 on 1 and 8 DF,  p-value: 1.7e-07

> par(mar=c(4,4,2,2))
> plot(x,y,pch=19,cex=0.7,xlim=range(0,25), ylim=range(0,40))
> abline(coef(fit1),col='red')

```



The line of best-fit yields an *SSE* value of 21.9957.

However we might not be satisfied that the straight-line model is adequate, and attempt a fit of a quadratic model.

```

> fit2<-lm(y~x+I(x^2))
> summary(fit2)

+
+ Call:
+ lm(formula = y ~ x + I(x^2))
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -2.8287 -0.9324  0.0737  1.3361  1.7171
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)   9.34251    1.65761    5.64  0.00079 ***
+ x             1.52462    0.37023    4.12  0.00447 **
+ I(x^2)       -0.00502    0.01681   -0.30  0.77367
+
+
+

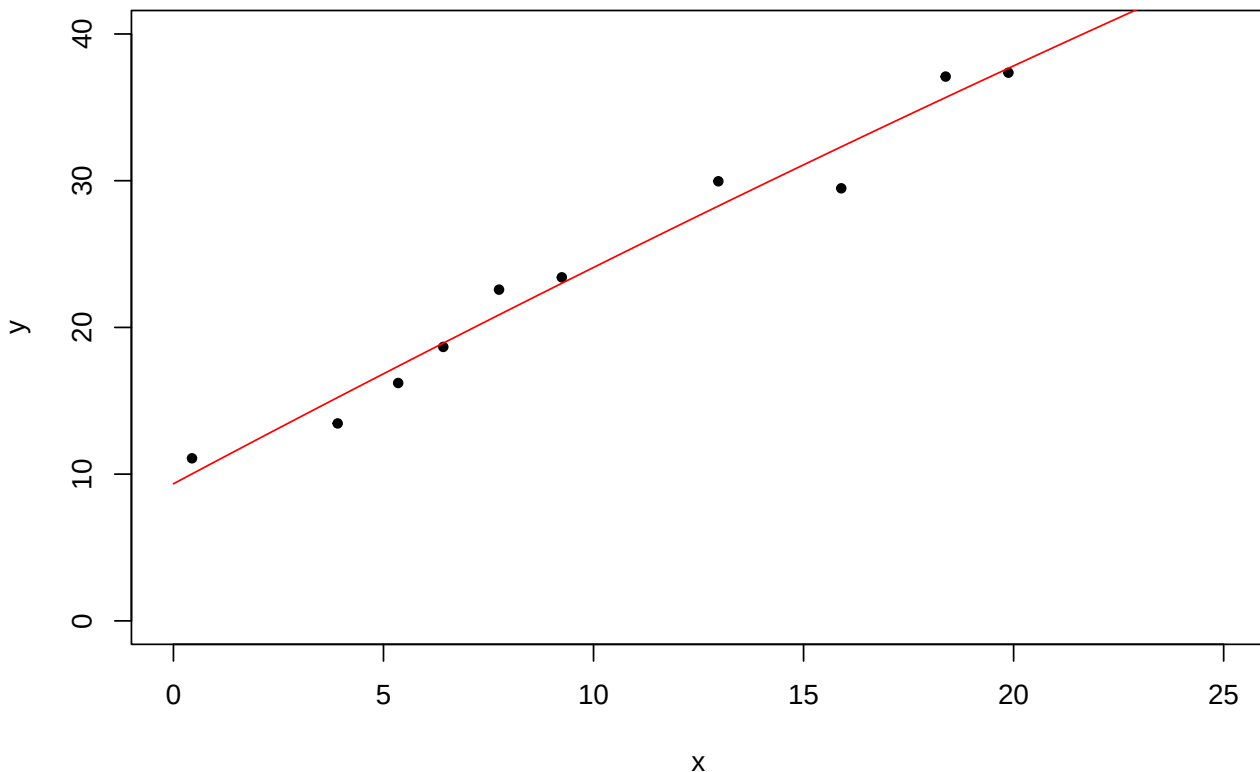
```

```

+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 1.76 on 7 degrees of freedom
+ Multiple R-squared:  0.972, Adjusted R-squared:  0.964
+ F-statistic: 123 on 2 and 7 DF,  p-value: 3.52e-06

> par(mar=c(4,4,2,2))
> plot(x,y,pch=19,cex=0.7,xlim=range(0,25), ylim=range(0,40))
> xv<-seq(0,25,by=0.01)
> yv<-predict(fit2,newdata=data.frame(x=xv))
> lines(xv,yv,col='red')

```



The quadratic fit yields an  $SSE$  value of 21.7184, which is lower than the  $SSE$  for the straight-line model.

We can go further and fit higher-order polynomials, and examine the  $SSE$  quantities.

```

> fit3<-lm(y~x+I(x^2)+I(x^3))
> fit4<-lm(y~x+I(x^2)+I(x^3)+I(x^4))
> fit5<-lm(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5))
> fit6<-lm(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6))
>
> sse1<-sum(fit1$residuals^2)
> sse2<-sum(fit2$residuals^2)
> sse3<-sum(fit3$residuals^2)
> sse4<-sum(fit4$residuals^2)
> sse5<-sum(fit5$residuals^2)
> sse6<-sum(fit6$residuals^2)

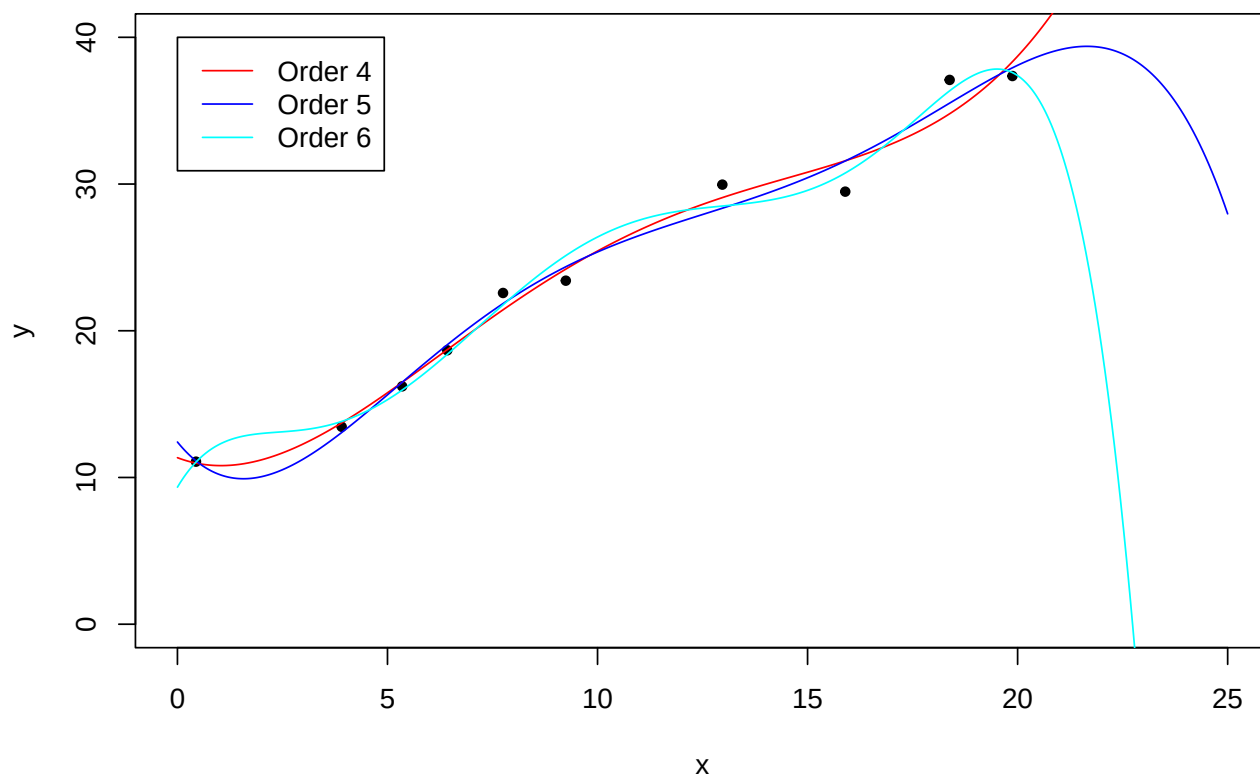
```

The results are as follows:

| Order | $SSE$   |
|-------|---------|
| 1     | 21.9957 |
| 2     | 21.7184 |
| 3     | 21.2019 |
| 4     | 13.5600 |
| 5     | 11.5724 |
| 6     | 8.1929  |

We see that the  $SSE$  drops considerably when we move from cubic (order equal to 3) to quartic (order equal to 4) polynomial, and then continues to decrease. We therefore examine the curves of best fit for those cases:

```
> par(mar=c(4, 4, 2, 2))
> plot(x, y, pch=19, cex=0.7, xlim=range(0, 25), ylim=range(0, 40))
> xv<-seq(0, 25, by=0.01)
> yv4<-predict(fit4, newdata=data.frame(x=xv))
> lines(xv, yv4, col='red')
> yv5<-predict(fit5, newdata=data.frame(x=xv))
> lines(xv, yv5, col='blue')
> yv6<-predict(fit6, newdata=data.frame(x=xv))
> lines(xv, yv6, col='cyan')
> legend(0, 40, c('Order 4', 'Order 5', 'Order 6'),
+       col=c('red', 'blue', 'cyan'), lty=1)
```

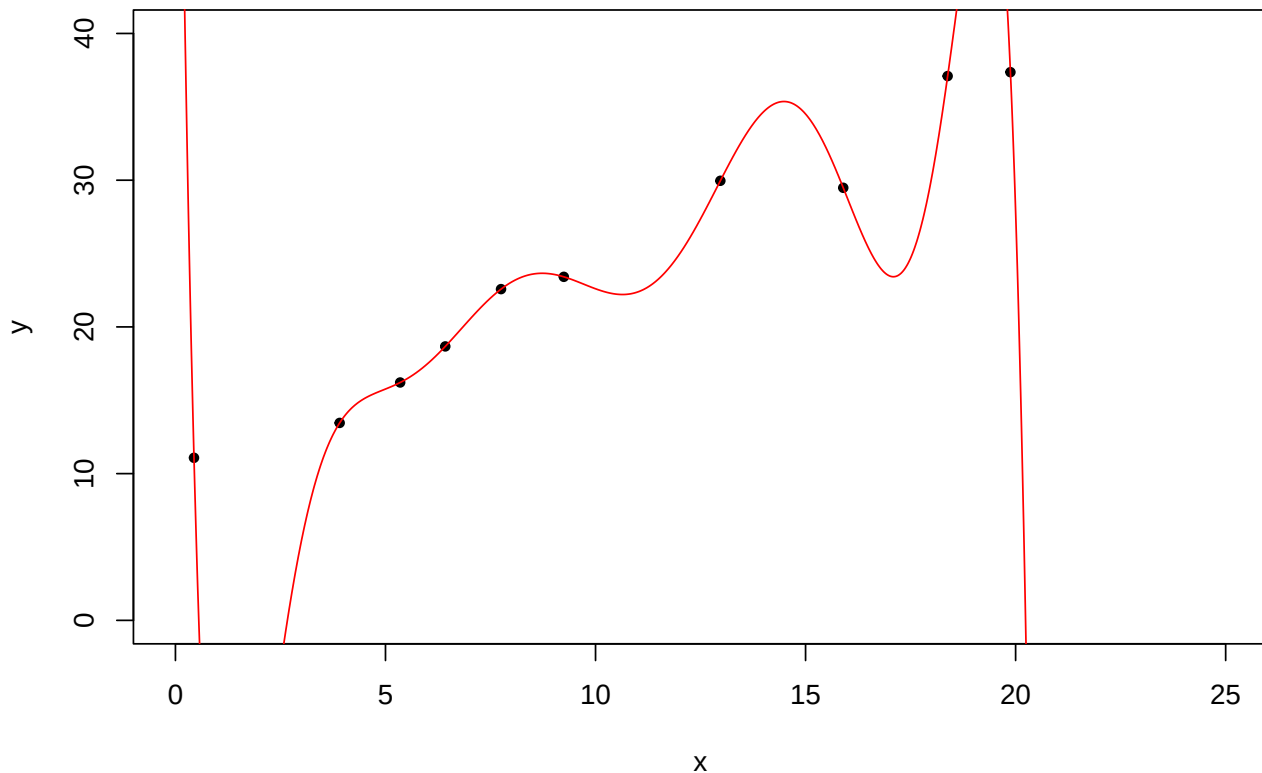


These results might encourage us to try an even higher polynomial, say up to order 9.

```
> fit9<-lm(y~x+I(x^2)+I(x^3)+I(x^4)+I(x^5)+I(x^6)+I(x^7)+I(x^8)+I(x^9))
```

The yields an  $SSE$  value of 0.0000 ! We now inspect the best fitting curve:

```
> par(mar=c(4,4,2,2))
> plot(x,y,pch=19,cex=0.7,xlim=range(0,25),ylim=range(0,40))
> xv<-seq(0,25,by=0.01)
> yv9<-predict(fit9,newdata=data.frame(x=xv))
> lines(xv,yv9,col='red')
```



We have been able to interpolate the data exactly. However, the resulting model is not that useful for interpretation: recall that our goal is to understand the relationship between  $x$  and  $y$  with the goal of explaining the underlying mechanism (the **systematic** variation), accounting for the assumption that the data are subject to random residual error (the **random** variation). The 9th order polynomial fit does not do this; it instead interprets **all** of the observed variation as **systematic** variation.

We must find a way to trade off **fidelity** to the data (as measured by  $SSE$ ) and model **complexity** (as measured by the number of terms in the model).