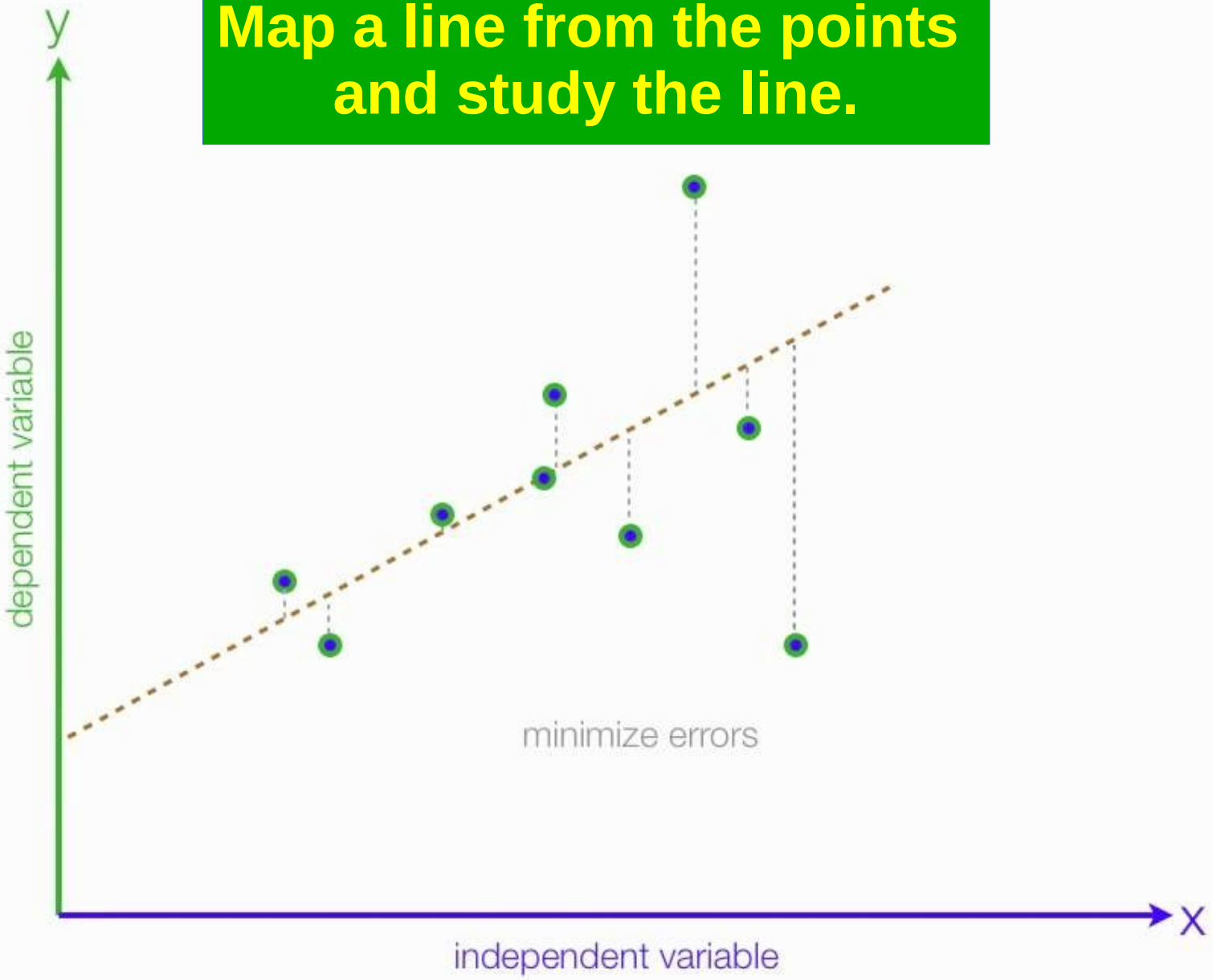
Data Analytics
CS301
Intro to Linear Models

Week 8: 4th March
Spring 2020

Oliver BONHAM-CARTER

Linear Regression

Map a line from the points and study the line.



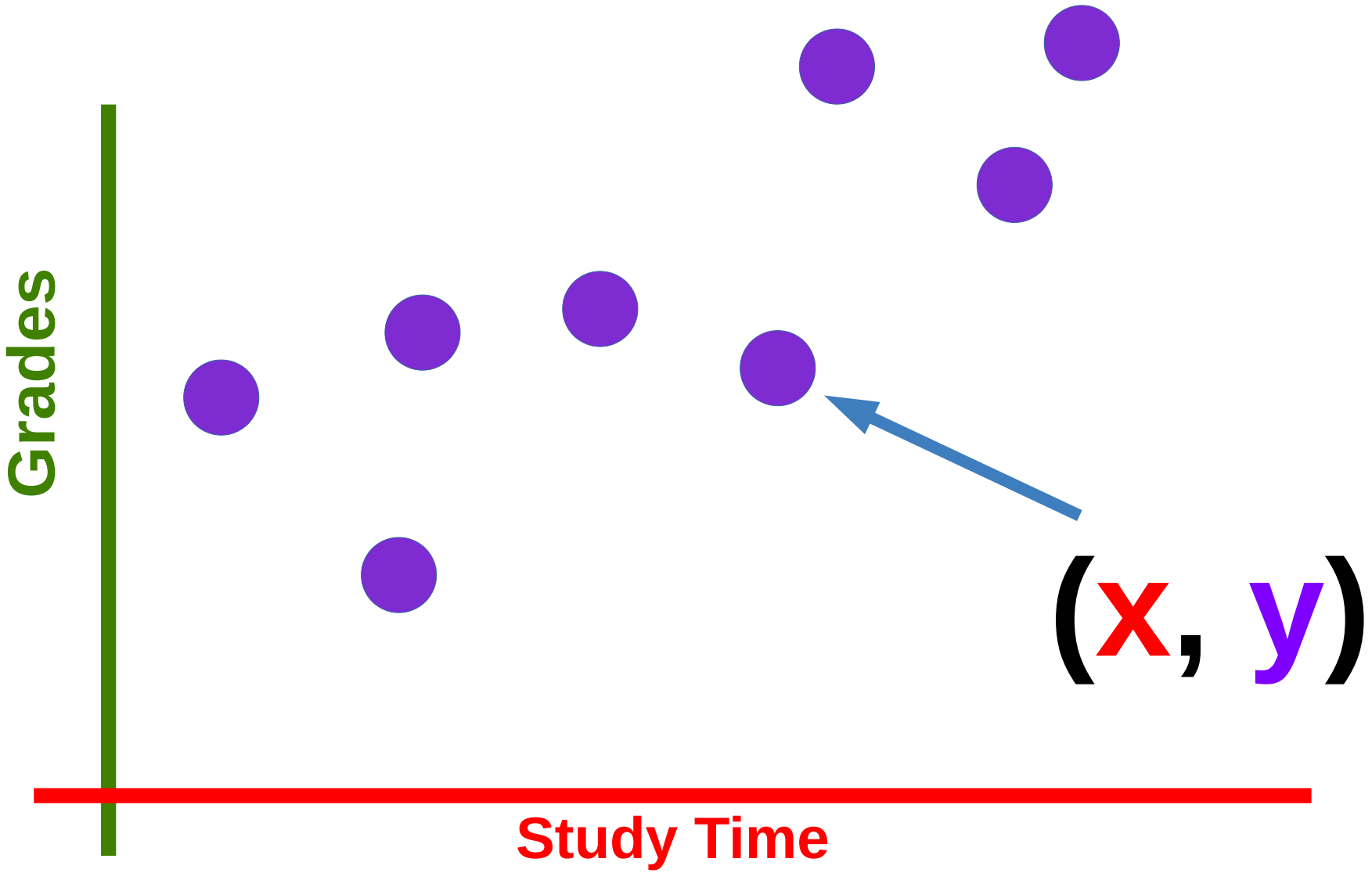
Linear Regression

- Is one thing able to influence another thing?
- A linear approach for modeling the relationship between a scalar dependent variable ***y*** and one or more explanatory variables, or independent variables, denoted by ***x***.
- *Simple linear regression*: Single explanatory variable; **models x and y**
- *Multiple linear regression*: More than one explanatory variable (***y's***); **models x and y1, y2**

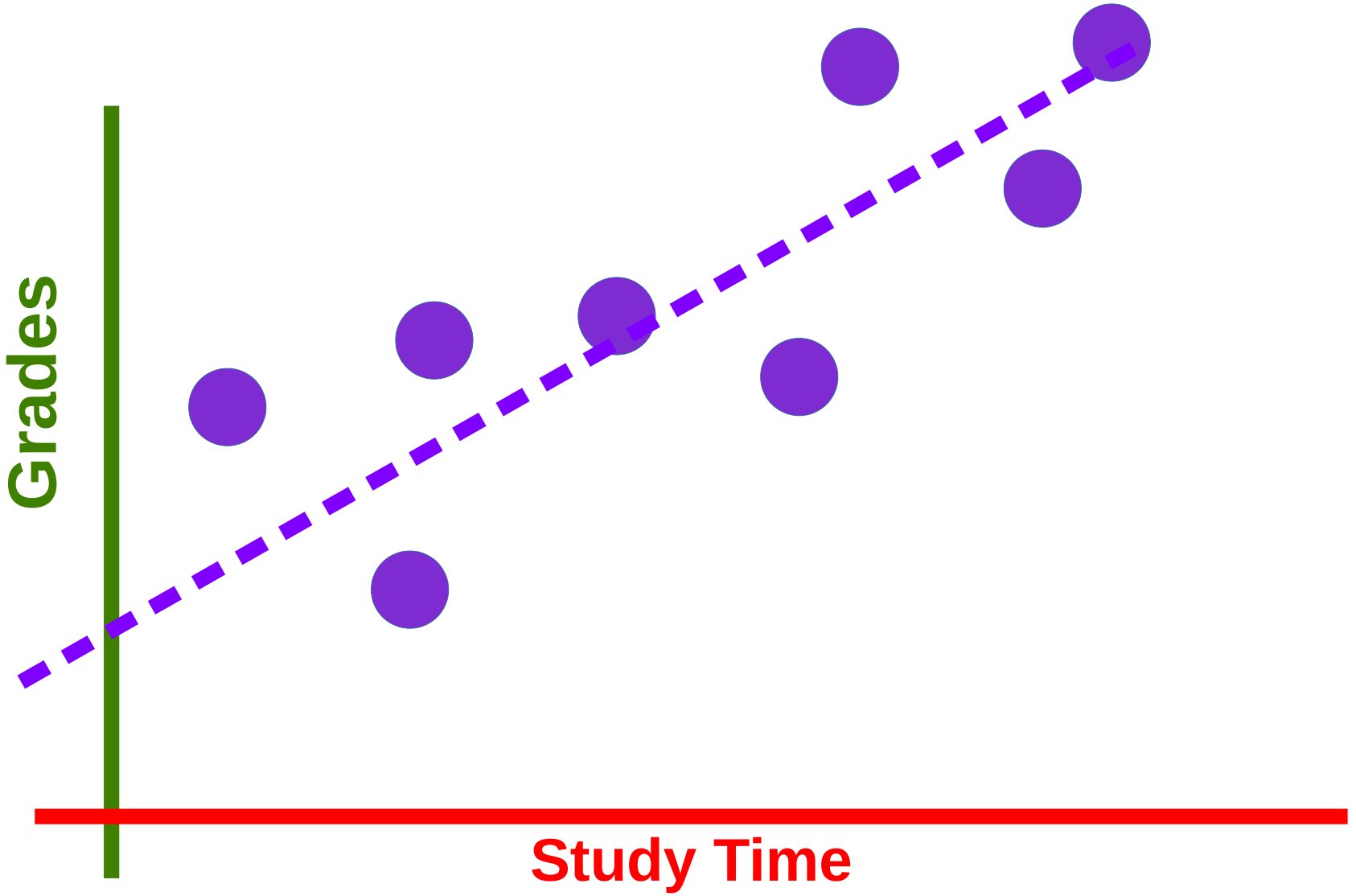
Linear Regression

- A straight line is drawn through a dot cloud.
- As the independent variable progresses, what is the dependent variable doing? Is there a relationship?
- The line has a y-intercept and a slope and can be used to determine the positive or negative relationship

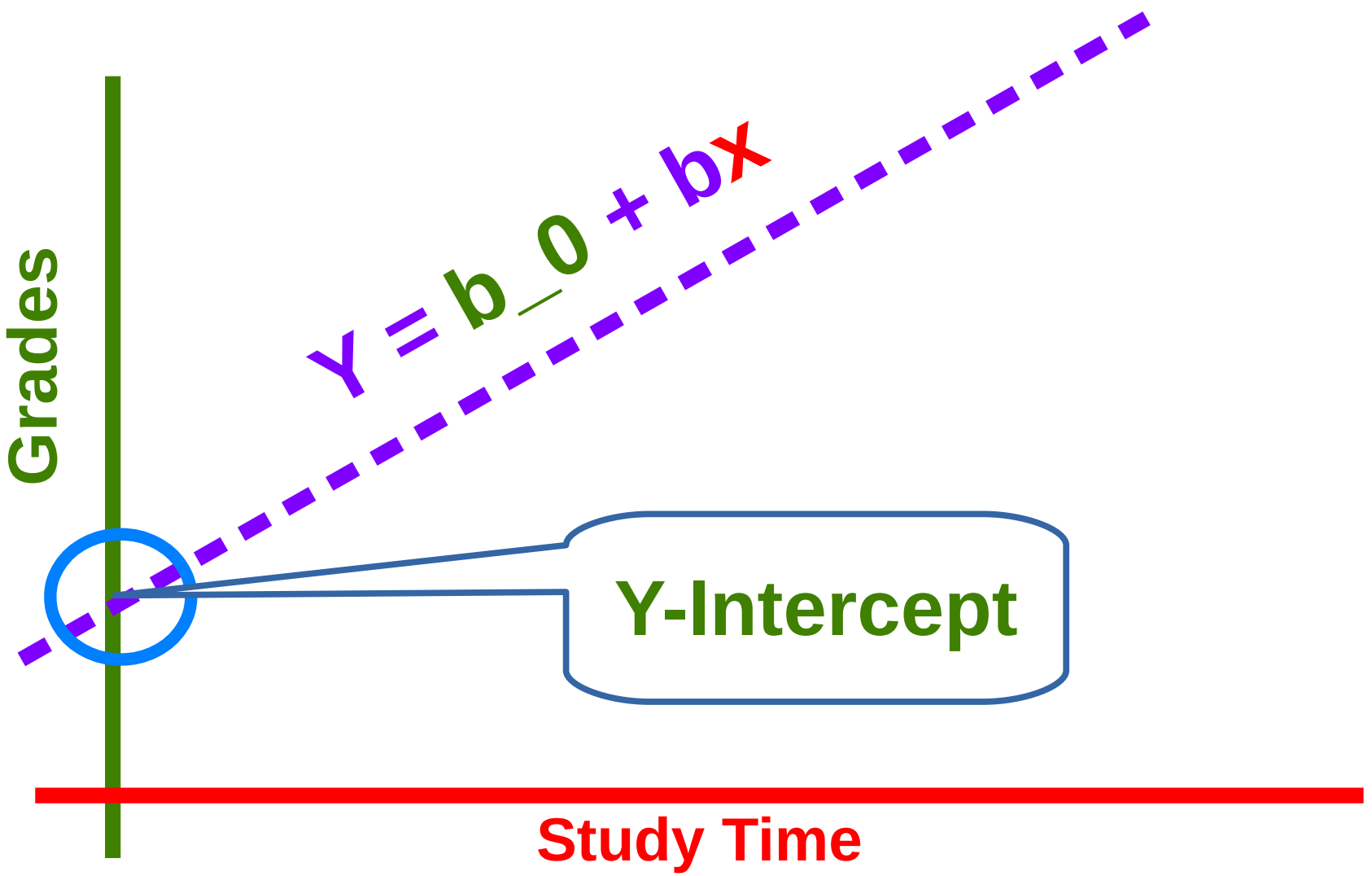
Plot *Study Time* to *Grades* Points



Draw Line Through Points



Intercept and Slope: Positive Relationship



Linear Model: `lm()`

- Function: `Lm()` is a linear model function, similar to *linear regression analysis*.

- Syntax: `lm(formula, data, subset, weights, ...)`
- Formula: model description, such as x ~ y
- Data: optional, variables in the model
- Subset: optional, a subset vector of observations to be used in the fitting process
- Weights: optional, a vector of weights to be used in the fitting process

Linear Model: Example

```
rm(list = ls())

height <- c(176, 154, 138, 196, 132, 176, 181, 169, 150,
175)

bodymass <- c(82, 49, 53, 112, 47, 69, 77, 71, 62, 78)

plot(bodymass, height)

hb <- lm(height ~ bodymass)

summary(hb)

# hb_noIntercept <- lm(height ~ bodymass – 1) # omitting
intercept

summary(hb_noIntercept)
```

There are two linear models to create from the above code of the *height* and *bodymass* sets.
Here, we are studying the `model`, **hb**

Linear Model: Example

```
Call:
lm(formula = height ~ bodymass)

Residuals:
    Min      1Q  Median      3Q     Max
-10.786  -8.307   1.272   7.818  12.253

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  98.0054    11.7053   8.373 3.14e-05 ***
bodymass      0.9528     0.1618   5.889 0.000366 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.358 on 8 degrees of freedom
Multiple R-squared:  0.8126,    Adjusted R-squared:  0.7891
F-statistic: 34.68 on 1 and 8 DF,  p-value: 0.0003662
```

Model: **hb**

Linear Model: Example

```
rm(list = ls())

height <- c(176, 154, 138, 196, 132, 176, 181, 169, 150,
175)

bodymass <- c(82, 49, 53, 112, 47, 69, 77, 71, 62, 78)

plot(bodymass, height)

# hb <- lm(height ~ bodymass)

summary(hb)

hb_noIntercept <- lm(height ~ bodymass – 1) # omitting
intercept

summary(hb_noIntercept)
```

There are two linear models to create from the above code of the *height* and *bodymass* sets.
Here, we are studying the `model`, **`hb_noIntercept`**

Linear Model: Example

```
Call:
lm(formula = height ~ bodymass - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-57.497   0.524   8.986  19.381  43.095

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
bodymass   2.2634     0.1205   18.78 1.58e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 27.57 on 9 degrees of freedom
Multiple R-squared:  0.9751,    Adjusted R-squared:  0.9724
F-statistic: 352.8 on 1 and 9 DF,  p-value: 1.578e-08
```

Model: **`hb_noIntercept`**

Linear Model: *p*-Values

H_0: (Null Hyp) there is no relationship between vars, m = 0
H_a: (Alt Hyp) There is a relationship between vars, m!= 0
 # Check the p-value:
If p-val =< alpha = 0.05: reject H0.
If p-val > alpha = 0.05: do not reject H0.

Why Two Models?!

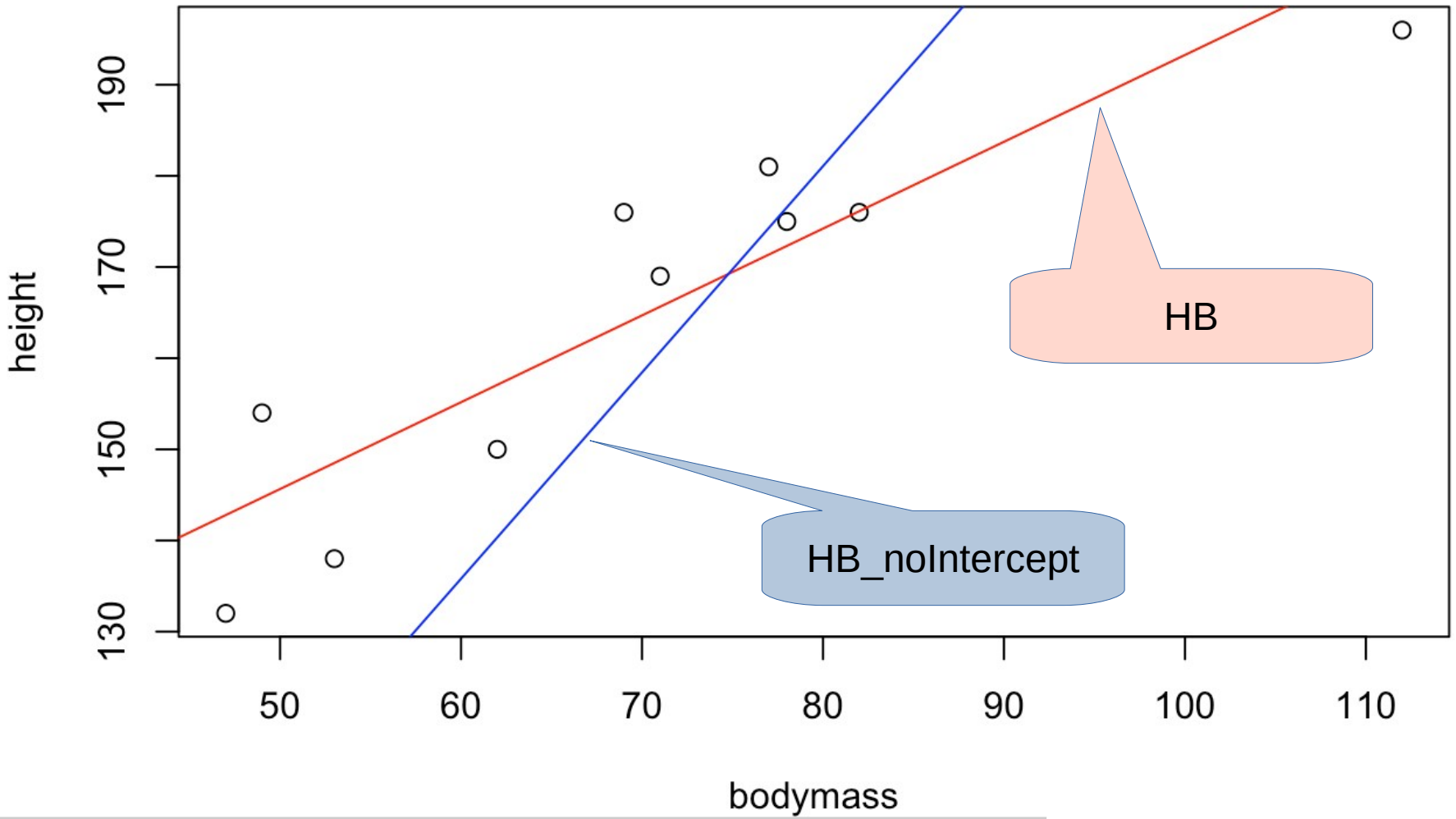
Keep the intercept	Omit the intercept
Model: hb	**Model: hb_noIntercept**
***P*-value: 0.0003662**	***P*-value: 1.578e-08**
Line Coefficients	**Line Coefficients**
(Intercept) bodymass **98.0054393 0.9527794**	**bodymass** **2.263363**

Line equation:

$y = m*x + b$

$y = \textbf{bodymass} * x + \textbf{intercept}$

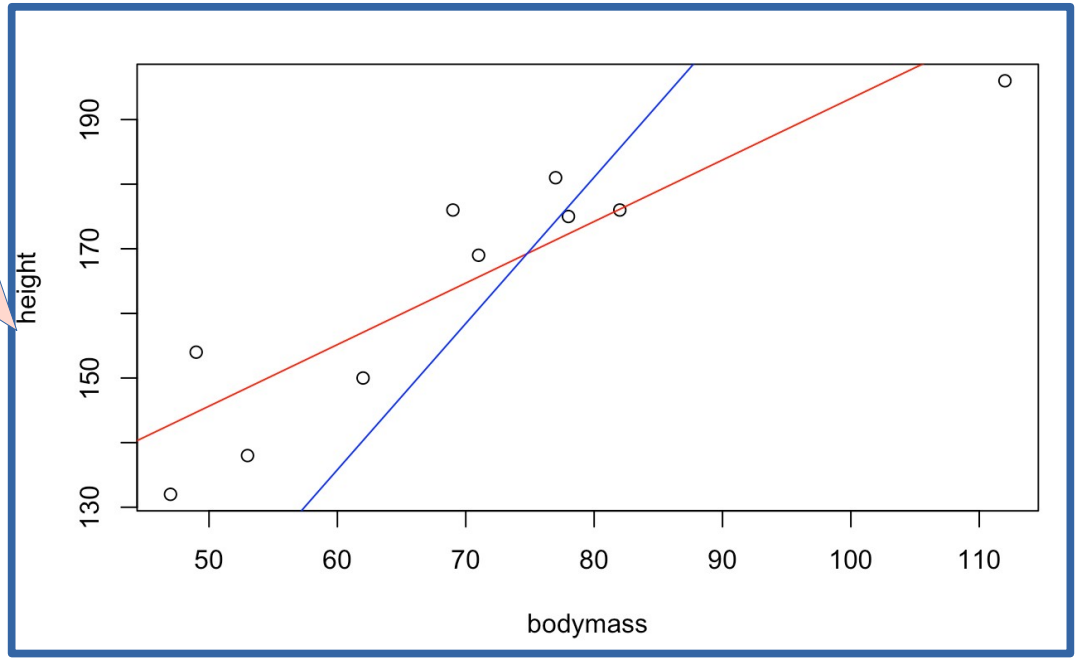
Lines With and Without Intercept




```
plot(bodymass, height)
abline(hb, col = "red")
abline(hb_noIntercept, col = "blue")
```

Lines With and Without Intercept

When we keep the intercept, outliers may distort our model. Often we keep the intercept unless we know that the data is linear. Examine both scenarios.



- To remove an intercept from a regression model is to set it equal to 0, rather than using it to estimate data.
- The model's fitted line estimates that intercept passes through most of the actual data.
- Setting the intercept to 0 makes line continue through the origin and it will never fit as well as a line whose intercept is estimated from the data.

Control and Training groups

Another lm() Example

ALLEGHENY COLLEGE

```
Ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
Trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
group <- gl(2, 10, 20, labels = c("Ctl","Trt"))
weight <- c(Ctl, Trt)
lm.D9 <- lm(weight ~ group)
lm.D90 <- lm(weight ~ group - 1) # omitting intercept
summary(lm.D9)
```

- H_0: (Null Hyp) there is no relationship between vars, m = 0
- H_a: (Alt Hyp) There is a relationship between vars, m!= 0

  # Check the p-value:

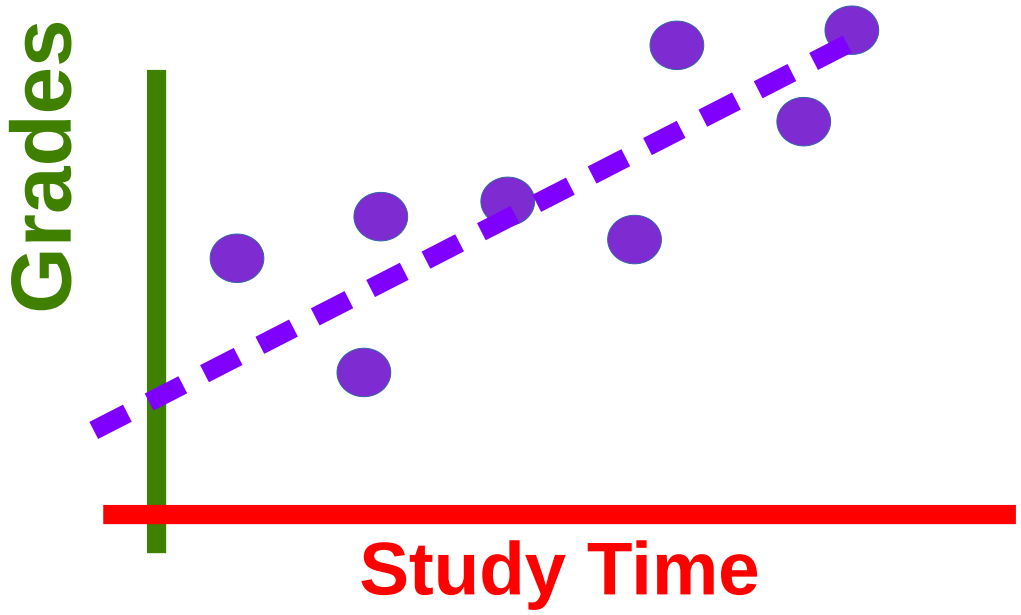
  - If p-val =< alpha = 0.05: reject H0.
  - If p-val > alpha = 0.05: do not reject H0.

Regression Assumptions

- The regression has five key assumptions:
  - Linear relationship
  - Multivariate normality
  - No or little multicollinearity
  - No auto-correlation
  - Homoscedasticity

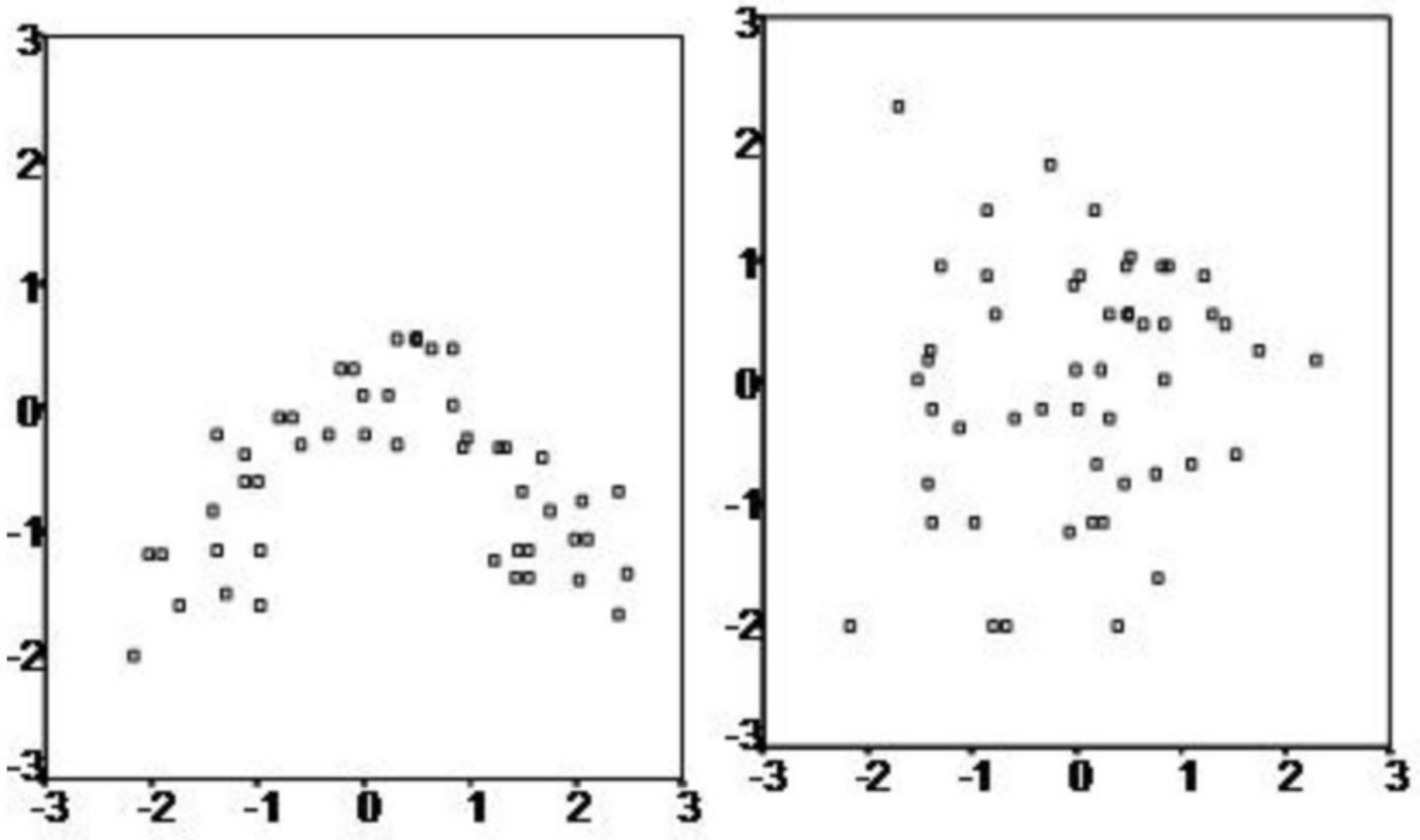
Linear Relationship

- Linear regression needs the relationship between the independent and dependent variables to be *linear*.
- Check for outliers linear regression is sensitive to outlier effects.



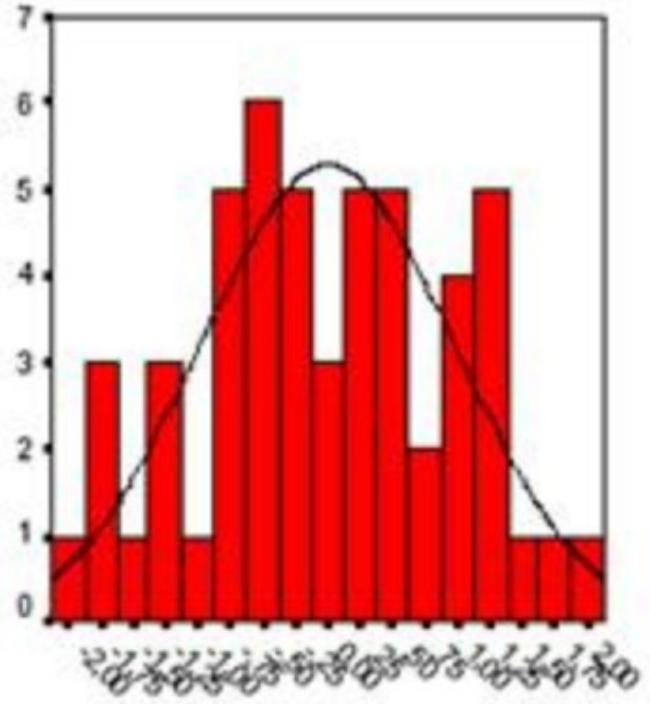
Linear Relationship

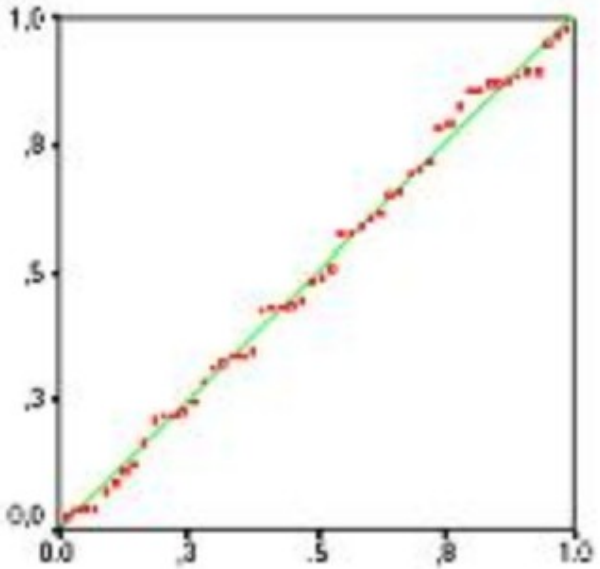
- Scatter plots: See where no and little linearity is present.



Multivariate Normality

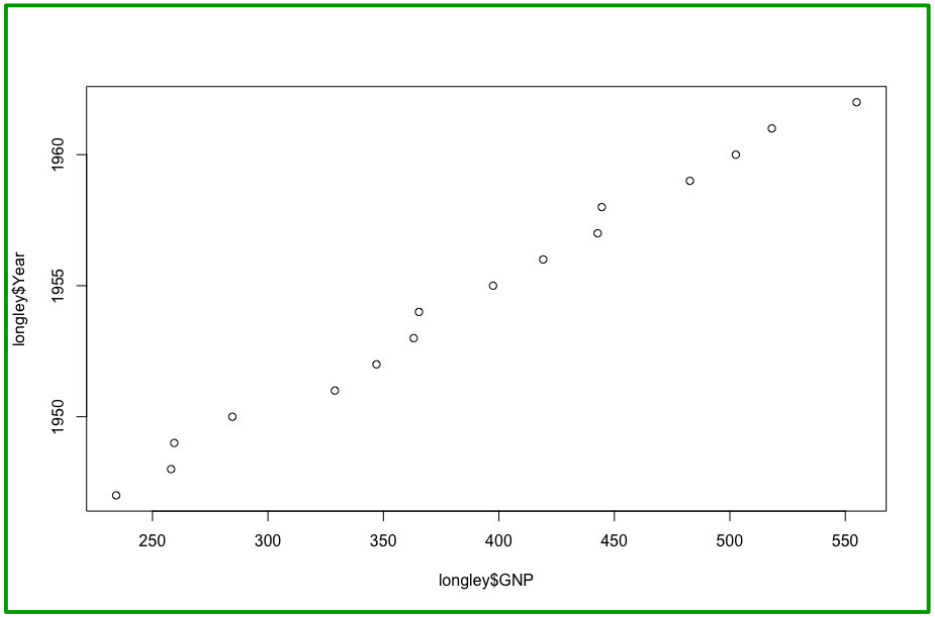
- The data must be of a *normal* distribution
- Check this with a QQ-plot

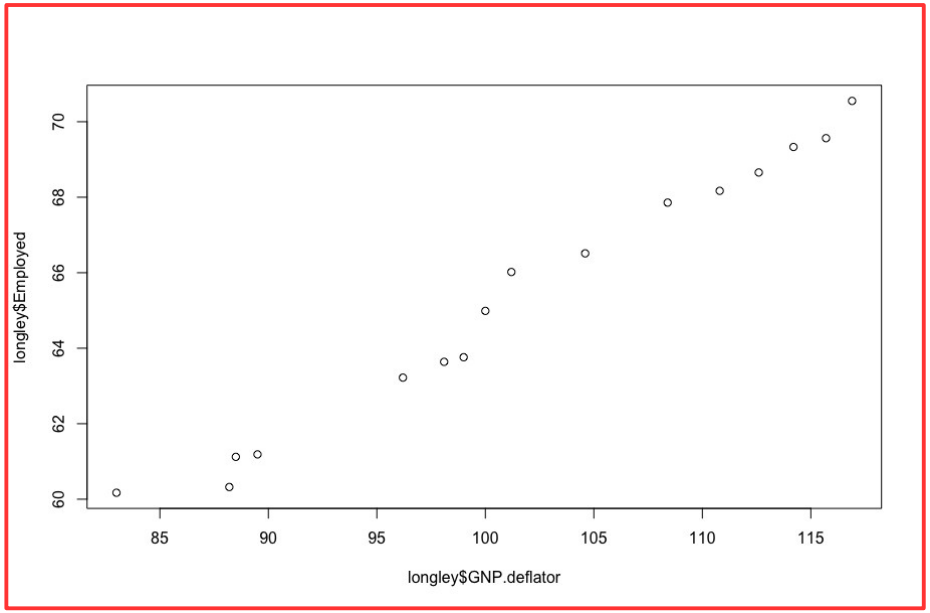




Multivariate Normality

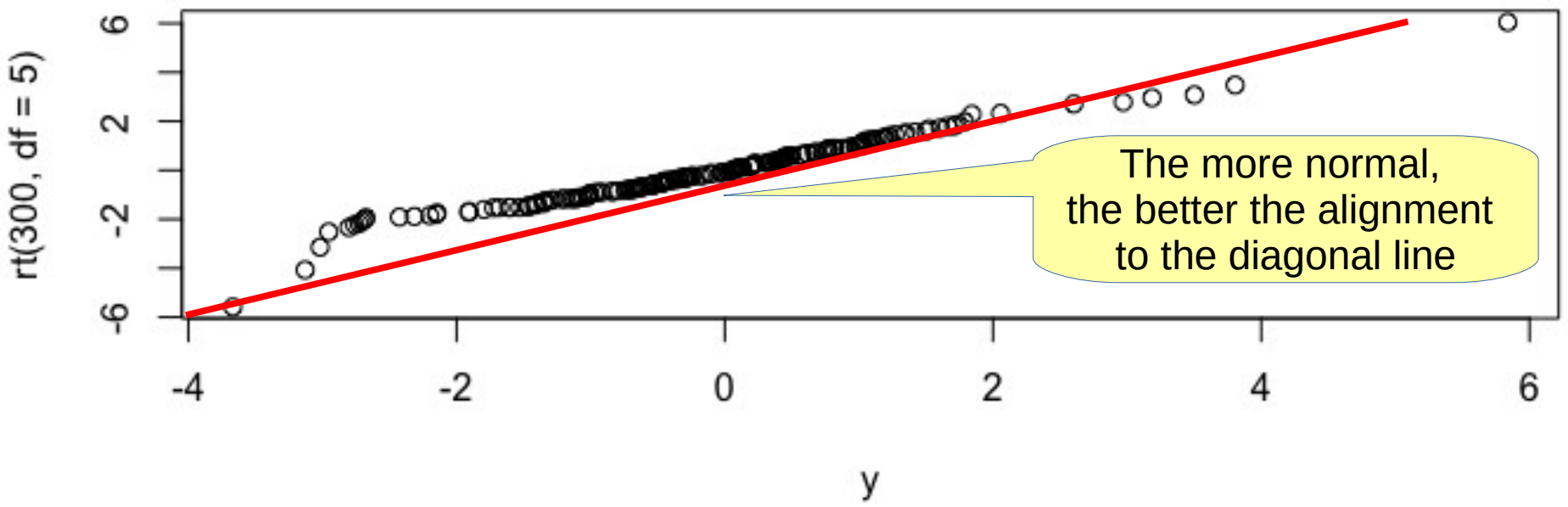
```
# Good
qqplot(x = longley$GNP, y = longley$Year)
# Not so good
qqplot(x = longley$GNP.deflator, y =
longley$Employed)
```





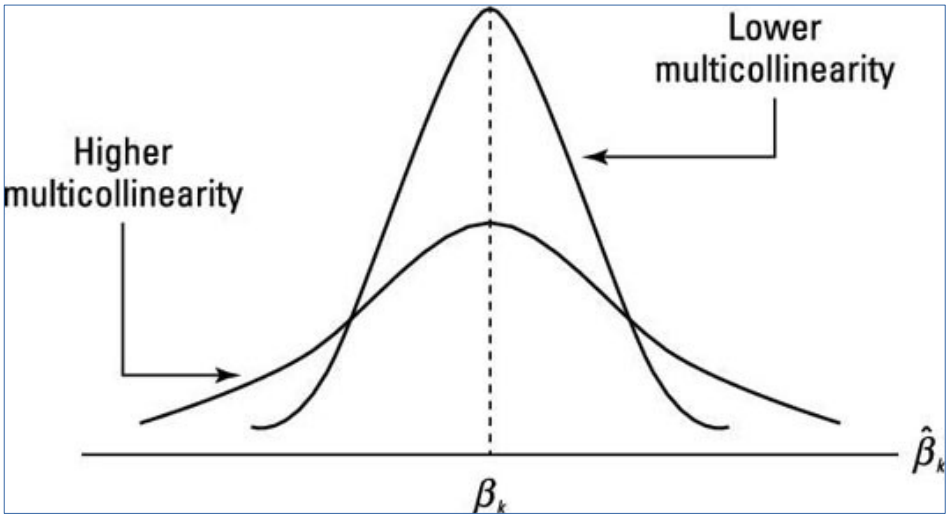
Detecting Normality: QQ-Plot

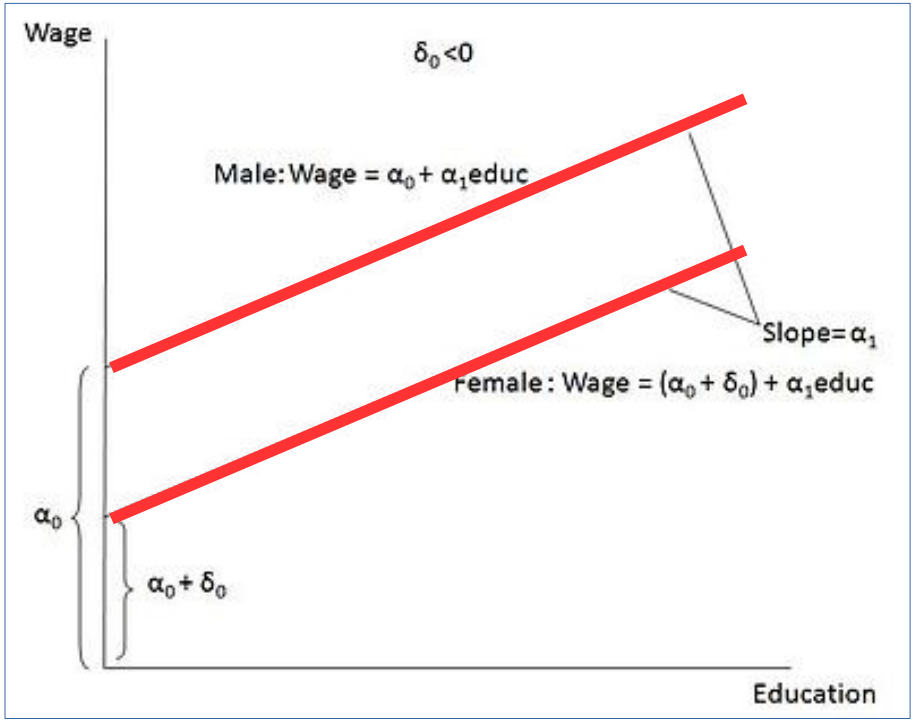
```
y <- rt(200, df = 5) #random
qqnorm(y); qqline(y, col = 2)
qqplot(y, rt(300, df = 5))
```



Multicollinearity

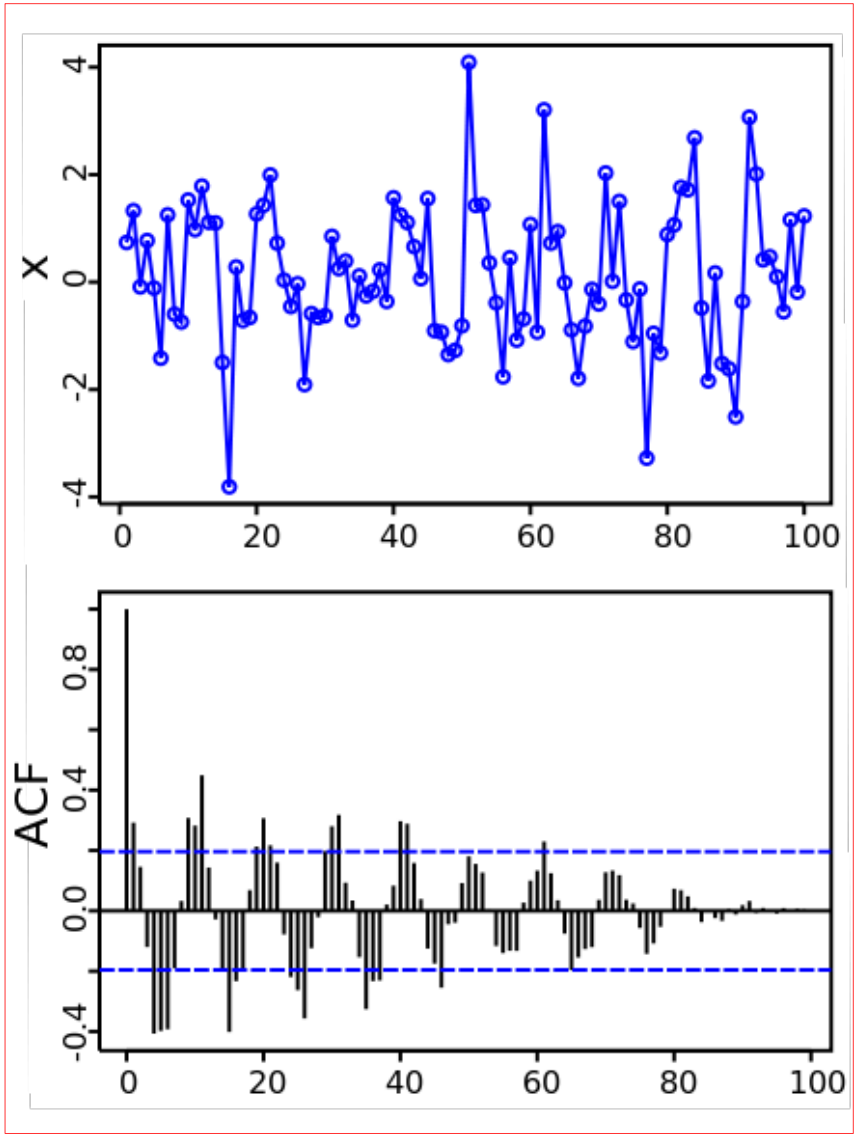
- A phenomenon in which one predictor variable in a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy.
- Same slope; same line





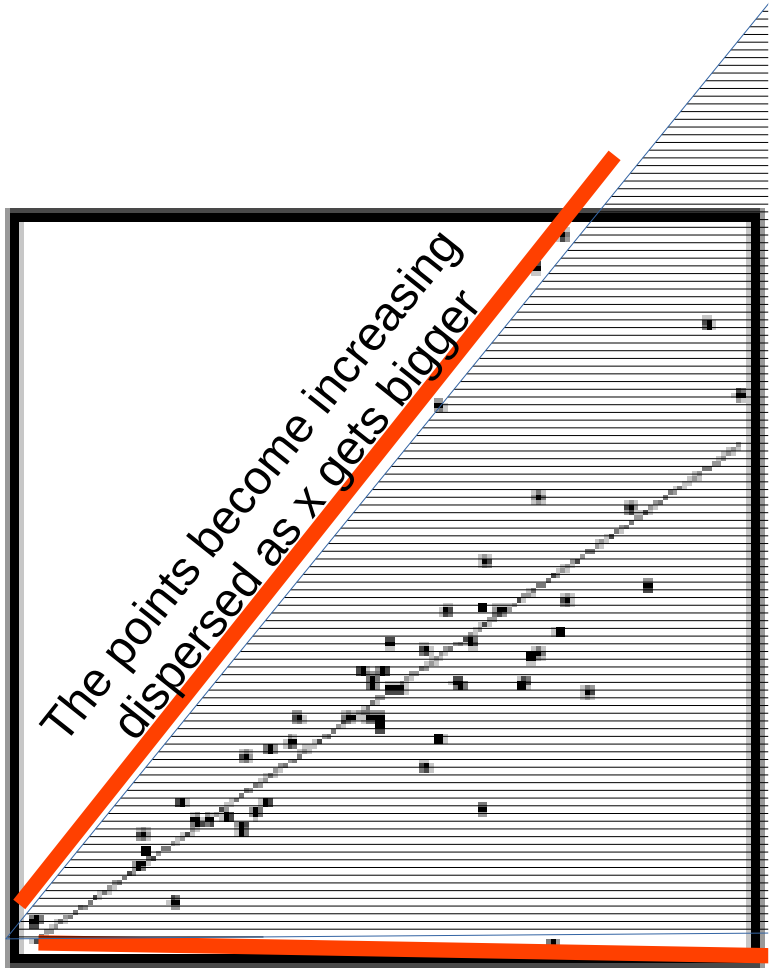
No Auto-correlation

- The correlation of a signal with a delayed copy of itself as a function of delay
- Ex: A plot of a series of 100 random numbers concealing a sine function. The sine function revealed in a correlogram produced by autocorrelation.
- Result: Non random output



Must Have Homoscedasticity

- Data sets in the regression must have the same variance (same quality of being different or divergent)
- This assumption means that the variance around the regression line is the same for all values of the predictor variable (*X*).
- The plot shows a violation of this assumption. For the lower values on the X-axis, the points are all very near the regression line.



Must Have Homoscedasticity

- Heteroscedasticity examples below
- Differing variance is bad for regression models.