**Assignment 4**

**Simple Linear Regression**

**Name: Motbhare Subodh Uddhav**

**Roll No: 92**

**Branch & Div: Computer Engineering & C**

**Gr No: 12120132**

**Source Code**

?cars #description of data set

str(cars)

summary(cars)

head(cars)

car\_data <- cars

plot(car\_data)

library('ggplot2')

g <- ggplot(aes(x=speed, y=dist), data=car\_data)

g

g1 <- g + geom\_point(col='red')

g1

df = data.frame(beta\_0=seq(0,1,length.out=10),beta\_1=seq(0,5,length.out=10))

df

#fit a line to data using abline function with randomly selected b0 & b1

for (i in 0:10) {

g1 = g1 + geom\_abline(intercept = df$beta\_0[i], slope = df$beta\_1[i], col=i)

}

g1

#generate RSS (residual sum of squares) vector for each of the 10 lines generated above

RSS <- vector(mode='numeric')

for (i in 0:10) {

RSS[i] <- sum(((df$beta\_1[i]\*car\_data$speed+df$beta\_0[i]) - car\_data$dist)\*\*2)

}

RSS

df$RSS = RSS

head(df)

ggplot(aes(x=beta\_0, y=RSS), data = df) + geom\_point() + geom\_line()

ggplot(aes(x=beta\_1, y=RSS), data = df) + geom\_point() + geom\_line()

i = which(df$RSS == min(df$RSS)) #select index of row containing minimum RSS

min\_beta = c(df$beta\_0[i], df$beta\_1[i])

min\_beta

g2 = ggplot(aes(x=speed, y=dist), data=car\_data) + geom\_point(col='red') + geom\_abline(intercept = min\_beta[1], slope = min\_beta[2], col='green')

g2

#fitting linear model using lm() function

l\_model <- lm(dist~speed, data=car\_data)

l\_model #model coefficients

summary(l\_model) #residual : diff. betn predicted & actual data

car\_data$y\_pred = predict(l\_model, data=car\_data$speed)

View(car\_data)

#plot the regression line using base plot function

plot(car\_data$speed, car\_data$dist, xlab = 'speed', ylab = 'Distance', main = 'Simple Linear Regression')

abline(l\_model, col='blue', lwd=2)

#plot the regression line using ggplot2

ggplot(aes(x=speed, y=dist), data=car\_data) + geom\_point(col='red') +

geom\_line(aes(car\_data$`speed`, car\_data$`y\_pred`), col='green') +

ylab('Distance') + ggtitle('Simple Linear Regression')

**Output:**

?cars #description of data set

> str(cars)

'data.frame': 50 obs. of 2 variables:

$ speed: num 4 4 7 7 8 9 10 10 10 11 ...

$ dist : num 2 10 4 22 16 10 18 26 34 17 ...

> summary(cars)

speed dist

Min. : 4.0 Min. : 2.00

1st Qu.:12.0 1st Qu.: 26.00

Median :15.0 Median : 36.00

Mean :15.4 Mean : 42.98

3rd Qu.:19.0 3rd Qu.: 56.00

Max. :25.0 Max. :120.00

> head(cars)

speed dist

1 4 2

2 4 10

3 7 4

4 7 22

5 8 16

6 9 10

>

> car\_data <- cars

> plot(car\_data)

>

> library('ggplot2')

> g <- ggplot(aes(x=speed, y=dist), data=car\_data)

> g

>

> g1 <- g + geom\_point(col='red')

> g1

>

> df = data.frame(beta\_0=seq(0,1,length.out=10),beta\_1=seq(0,5,length.out=10))

> df

beta\_0 beta\_1

1 0.0000000 0.0000000

2 0.1111111 0.5555556

3 0.2222222 1.1111111

4 0.3333333 1.6666667

5 0.4444444 2.2222222

6 0.5555556 2.7777778

7 0.6666667 3.3333333

8 0.7777778 3.8888889

9 0.8888889 4.4444444

10 1.0000000 5.0000000

>

> #fit a line to data using abline function with randomly selected b0 & b1

> for (i in 0:10) {

+ g1 = g1 + geom\_abline(intercept = df$beta\_0[i], slope = df$beta\_1[i], col=i)

+ }

> g1

>

> #generate RSS (residual sum of squares) vector for each of the 10 lines generated above

> RSS <- vector(mode='numeric')

>

> for (i in 0:10) {

+ RSS[i] <- sum(((df$beta\_1[i]\*car\_data$speed+df$beta\_0[i]) - car\_data$dist)\*\*2)

+ }

> RSS

[1] 124903.00 85846.06 55145.91 32802.56 18815.99 13186.21 15913.22 26997.02

[9] 46437.62 74235.00

>

> df$RSS = RSS

> head(df)

beta\_0 beta\_1 RSS

1 0.0000000 0.0000000 124903.00

2 0.1111111 0.5555556 85846.06

3 0.2222222 1.1111111 55145.91

4 0.3333333 1.6666667 32802.56

5 0.4444444 2.2222222 18815.99

6 0.5555556 2.7777778 13186.21

> ggplot(aes(x=beta\_0, y=RSS), data = df) + geom\_point() + geom\_line()

> ggplot(aes(x=beta\_1, y=RSS), data = df) + geom\_point() + geom\_line()

>

> i = which(df$RSS == min(df$RSS)) #select index of row containing minimum RSS

> min\_beta = c(df$beta\_0[i], df$beta\_1[i])

> min\_beta

[1] 0.5555556 2.7777778

>

> g2 = ggplot(aes(x=speed, y=dist), data=car\_data) + geom\_point(col='red') + geom\_abline(intercept = min\_beta[1], slope = min\_beta[2], col='green')

> g2

>

> #fitting linear model using lm() function

> l\_model <- lm(dist~speed, data=car\_data)

> l\_model #model coefficients

Call:

lm(formula = dist ~ speed, data = car\_data)

Coefficients:

(Intercept) speed

-17.579 3.932

> summary(l\_model) #residual : diff. betn predicted & actual data

Call:

lm(formula = dist ~ speed, data = car\_data)

Residuals:

Min 1Q Median 3Q Max

-29.069 -9.525 -2.272 9.215 43.201

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -17.5791 6.7584 -2.601 0.0123 \*

speed 3.9324 0.4155 9.464 1.49e-12 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.38 on 48 degrees of freedom

Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438

F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12

>

> car\_data$y\_pred = predict(l\_model, data=car\_data$speed)

> View(car\_data)

>

> #plot the regression line using base plot function

> plot(car\_data$speed, car\_data$dist, xlab = 'speed', ylab = 'Distance', main = 'Simple Linear Regression')

> abline(l\_model, col='blue', lwd=2)

>

> #plot the regression line using ggplot2

> ggplot(aes(x=speed, y=dist), data=car\_data) + geom\_point(col='red') +

+ geom\_line(aes(car\_data$`speed`, car\_data$`y\_pred`), col='green') +

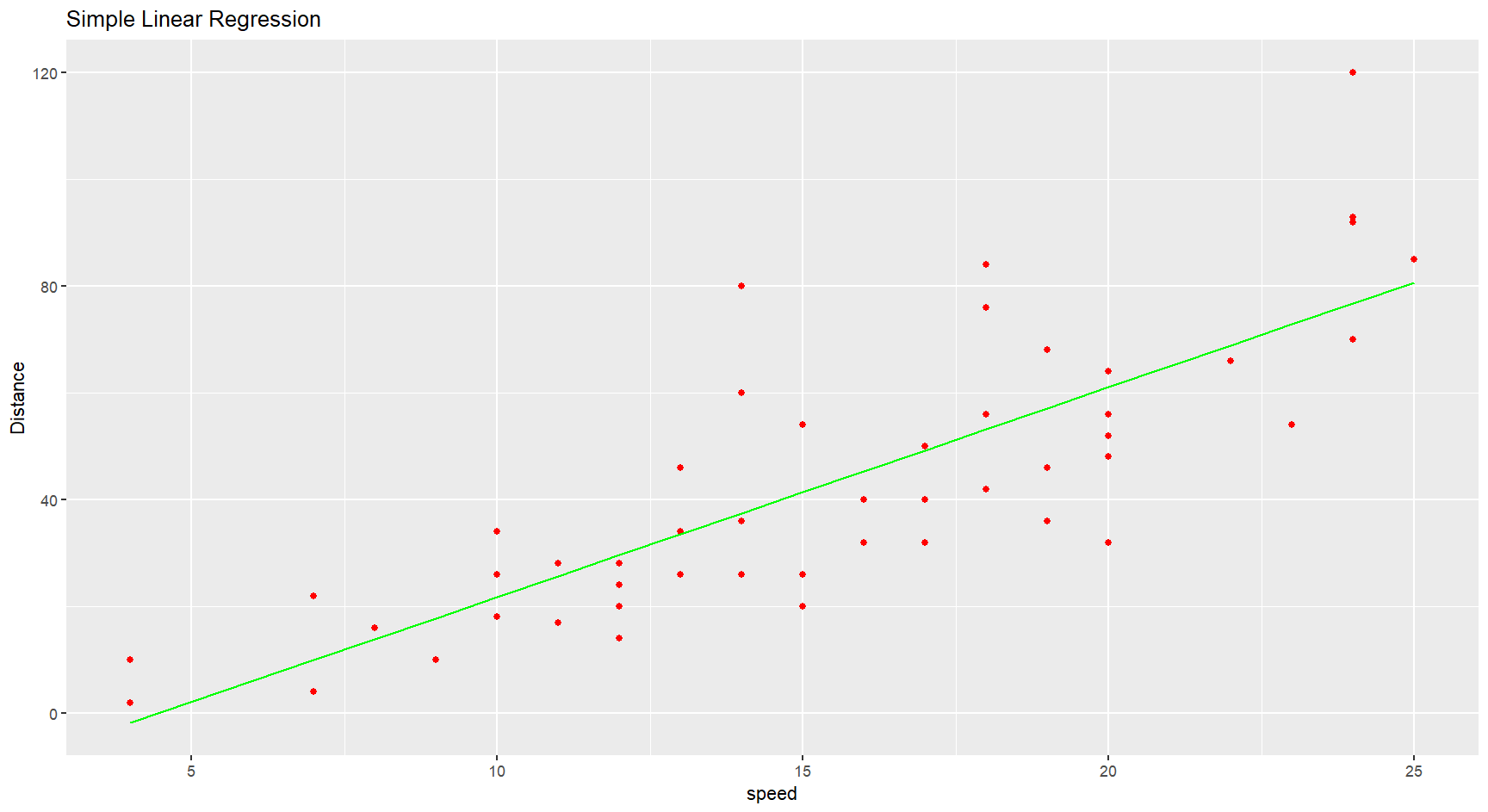
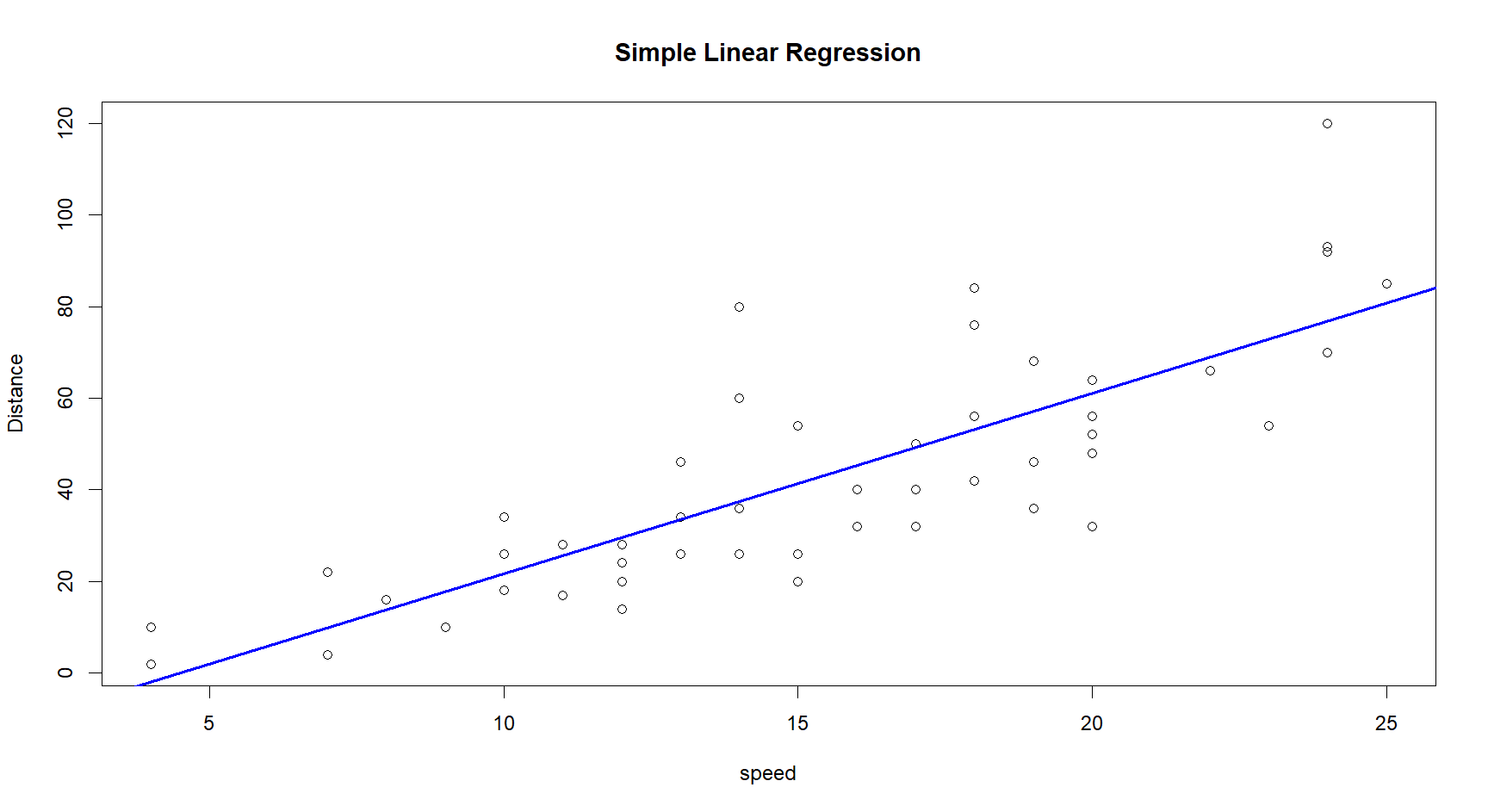
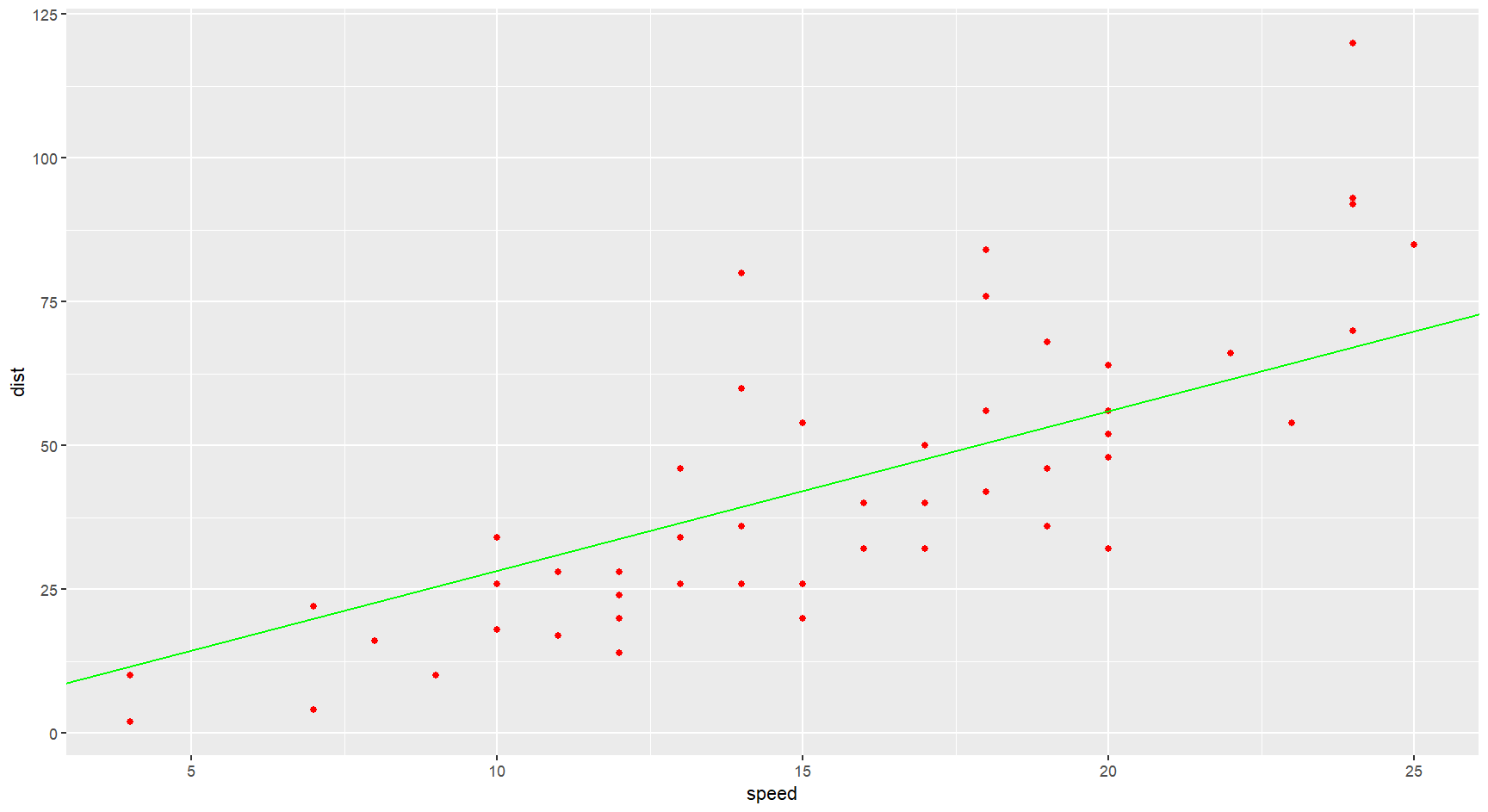
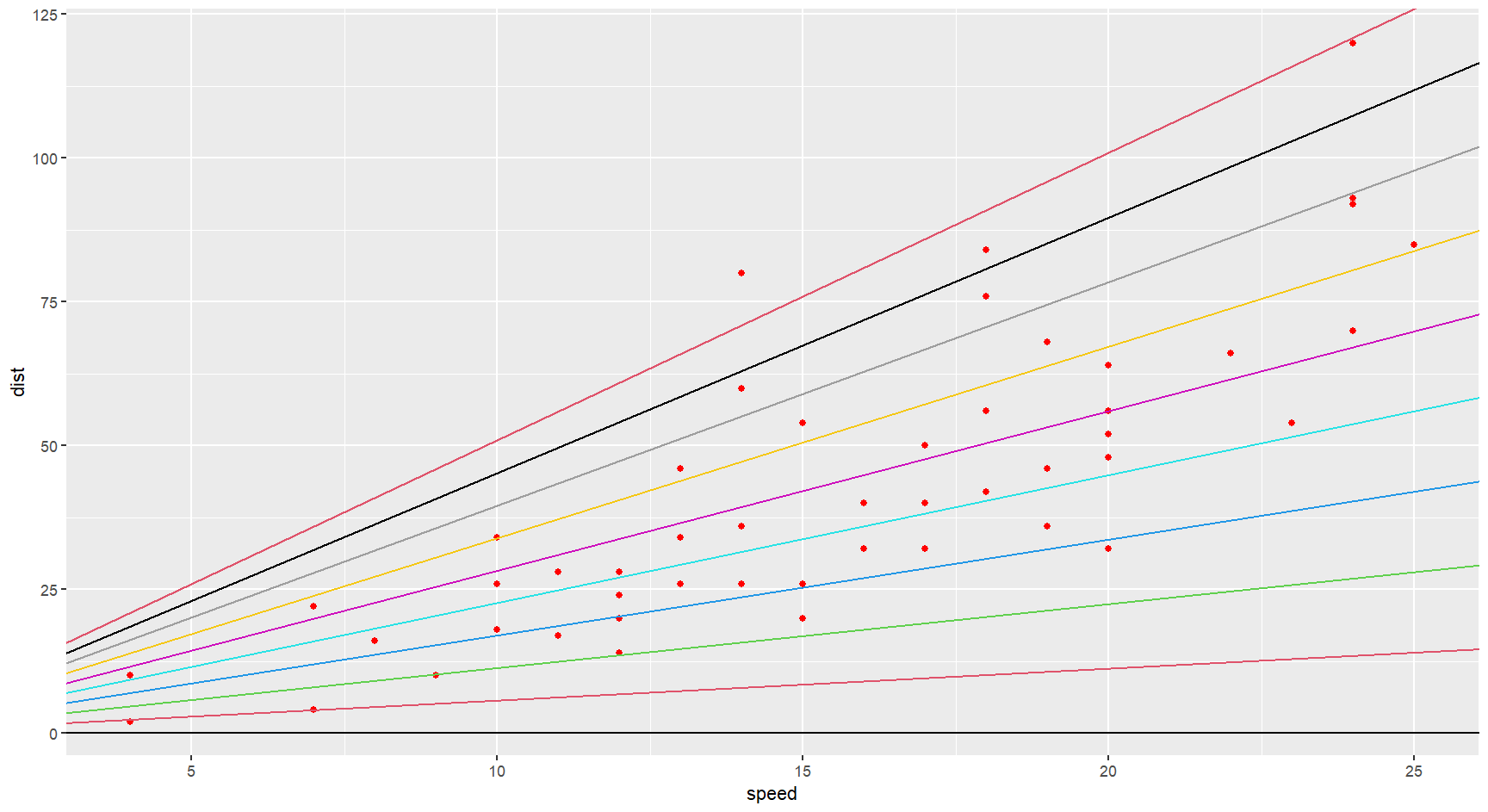
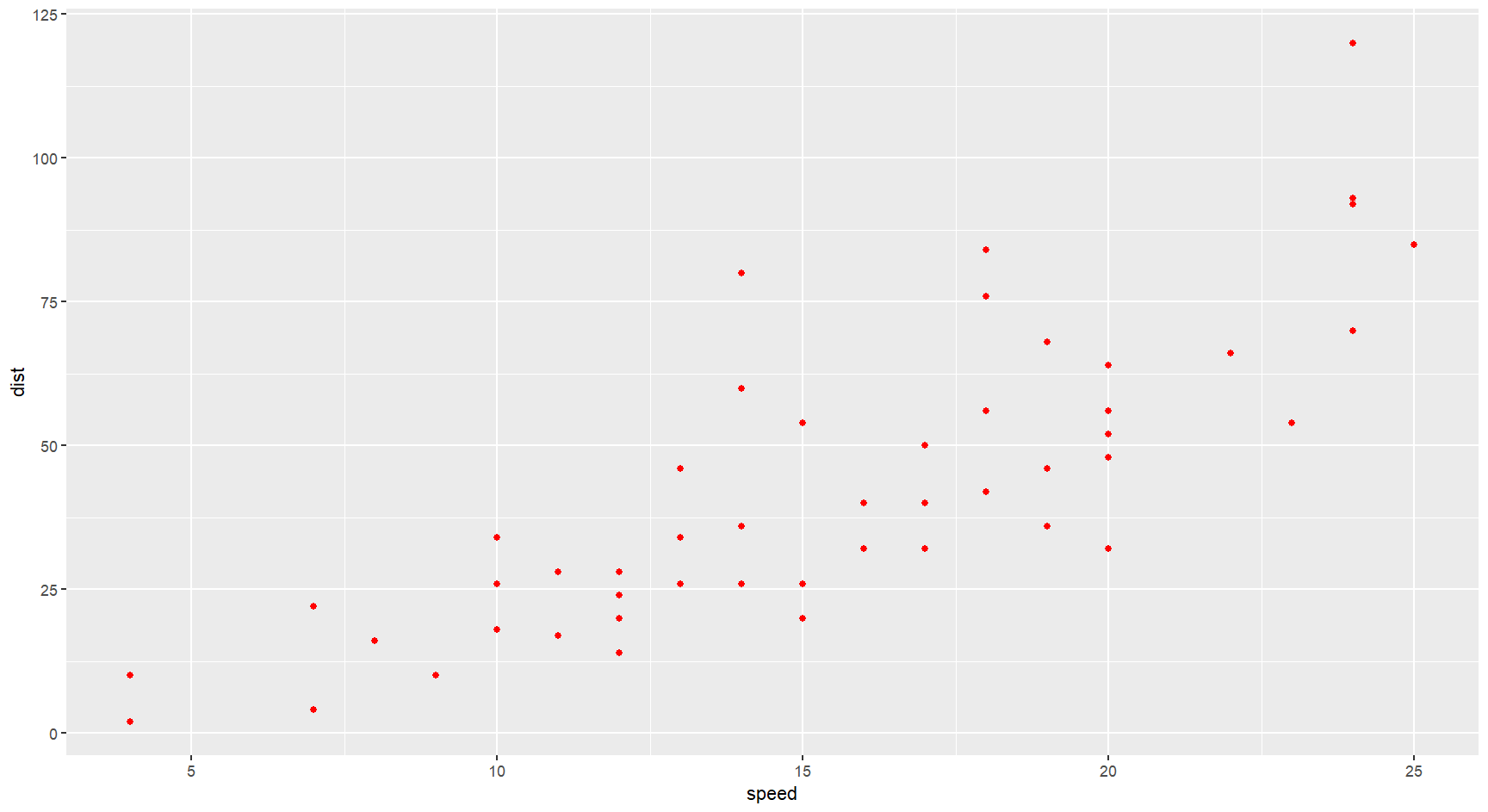
+ ylab('Distance') + ggtitle('Simple Linear Regression')

Warning messages:

1: Use of `car\_data$speed` is discouraged. Use `speed` instead.

2: Use of `car\_data$y\_pred` is discouraged. Use `y\_pred` instead.

>

****