> library(readr)

> del\_time <- read.csv(file.choose())

> View(del\_time)

> del\_time

Delivery.Time Sorting.Time

1 21.00 10

2 13.50 4

3 19.75 6

4 24.00 9

5 29.00 10

6 15.35 6

7 19.00 7

8 9.50 3

9 17.90 10

10 18.75 9

11 19.83 8

12 10.75 4

13 16.68 7

14 11.50 3

15 12.03 3

16 14.88 4

17 13.75 6

18 18.11 7

19 8.00 2

20 17.83 7

21 21.50 5

> # Exploratory data analysis #

> summary(del\_time)

Delivery.Time Sorting.Time

Min. : 8.00 Min. : 2.00

1st Qu.:13.50 1st Qu.: 4.00

Median :17.83 Median : 6.00

Mean :16.79 Mean : 6.19

3rd Qu.:19.75 3rd Qu.: 8.00

Max. :29.00 Max. :10.00

> # Scatter plot #

> plot(del\_time$Delivery.Time, del\_time$Sorting.Time) # plot(X,Y)

> attach(del\_time)

> # Correlation Coefficient (r) #

> cor(Delivery.Time, Sorting.Time)

[1] 0.8259973

> # Simple Linear Regression Model #

> reg <- lm(Delivery.Time ~ Sorting.Time) #lm(Y ~ X)

> summary(reg)

Call:

lm(formula = Delivery.Time ~ Sorting.Time)

Residuals:

Min 1Q Median 3Q Max

-5.1729 -2.0298 -0.0298 0.8741 6.6722

Coefficients:

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

(Intercept) 6.5827 1.7217 3.823 0.00115 \*\*

Sorting.Time 1.6490 0.2582 6.387 3.98e-06 \*\*\*

---

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

Residual standard error: 2.935 on 19 degrees of freedom

**Multiple R-squared: 0.6823, Adjusted R-squared: 0.6655**

F-statistic: 40.8 on 1 and 19 DF, p-value: 3.983e-06

> pred <- predict(reg)

> pred

1 2 3 4 5 6 7 8 9

23.072933 13.178814 16.476853 21.423913 23.072933 16.476853 18.125873 11.529794 23.072933

10 11 12 13 14 15 16 17 18

21.423913 19.774893 13.178814 18.125873 11.529794 11.529794 13.178814 16.476853 18.125873

19 20 21

9.880774 18.125873 14.827833

> reg$residuals

1 2 3 4 5 6 7

-2.07293294 0.32118644 3.27314665 2.57608696 5.92706706 -1.12685335 0.87412675

8 9 10 11 12 13 14

-2.02979366 -5.17293294 -2.67391304 0.05510685 -2.42881356 -1.44587325 -0.02979366

15 16 17 18 19 20 21

0.50020634 1.70118644 -2.72685335 -0.01587325 -1.88077377 -0.29587325 6.67216654

> sum(reg$residuals)

[1] 9.992007e-16

> mean(reg$residuals)

[1] 4.758099e-17

> sqrt(sum(reg$residuals^2)/nrow(del\_time)) #RMSE

[1] 2.79165

> sqrt(mean(reg$residuals^2))

[1] 2.79165

> confint(reg,level = 0.95)

2.5 % 97.5 %

(Intercept) 2.979134 10.186334

Sorting.Time 1.108673 2.189367

> predict(reg,interval="predict")

fit lwr upr

1 23.072933 16.457161 29.68870

2 13.178814 6.780993 19.57663

3 16.476853 10.188630 22.76508

4 21.423913 14.955850 27.89198

5 23.072933 16.457161 29.68870

6 16.476853 10.188630 22.76508

7 18.125873 11.823294 24.42845

8 11.529794 5.010345 18.04924

9 23.072933 16.457161 29.68870

10 21.423913 14.955850 27.89198

11 19.774893 13.411938 26.13785

12 13.178814 6.780993 19.57663

13 18.125873 11.823294 24.42845

14 11.529794 5.010345 18.04924

15 11.529794 5.010345 18.04924

16 13.178814 6.780993 19.57663

17 16.476853 10.188630 22.76508

18 18.125873 11.823294 24.42845

19 9.880774 3.198090 16.56346

20 18.125873 11.823294 24.42845

21 14.827833 8.507631 21.14804

Warning message:

In predict.lm(reg, interval = "predict") :

predictions on current data refer to \_future\_ responses

> predict

standardGeneric for "predict" defined from package "stats"

function (object, ...)

standardGeneric("predict")

<environment: 0x000001923d025518>

Methods may be defined for arguments: object

Use showMethods("predict") for currently available ones.

> ggplot(data = del\_time, aes(x = Delivery.Time, y = Sorting.Time)) +

+ geom\_point(color='blue') +

+ geom\_line(color='red',data = del\_time, aes(x=Delivery.Time, y=pred))

> ## Logrithamic Model/ Transformation ##

> # x = log(Delivery.Time); y = Sorting.Time

> plot(log(Delivery.Time), Sorting.Time)

> cor(log(Delivery.Time), Sorting.Time)

[1] 0.8431773

> reg\_log <- lm(Sorting.Time ~ log(Delivery.Time)) ## lm(Y ~ X)

> summary(reg\_log)

Call:

lm(formula = Sorting.Time ~ log(Delivery.Time))

Residuals:

Min 1Q Median 3Q Max

-3.1658 -0.9513 0.0949 0.5029 3.0685

Coefficients:

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

(Intercept) -12.4992 2.7510 -4.543 0.000222 \*\*\*

log(Delivery.Time) 6.7355 0.9853 6.836 1.59e-06 \*\*\*

---

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

Residual standard error: 1.402 on 19 degrees of freedom

**Multiple R-squared: 0.7109, Adjusted R-squared: 0.6957**

F-statistic: 46.73 on 1 and 19 DF, p-value: 1.593e-06

> predict(reg\_log)

1 2 3 4 5 6 7 8 9

8.007296 5.031310 7.593942 8.906703 10.181351 5.896328 7.333179 2.664452 6.931482

10 11 12 13 14 15 16 17 18

7.243965 7.621170 3.497060 6.456018 3.951314 4.254794 5.686869 5.154901 7.010043

19 20 21

1.506947 6.905090 8.165786

> reg\_log$residuals

1 2 3 4 5 6 7

1.99270440 -1.03130970 -1.59394209 0.09329724 -0.18135141 0.10367218 -0.33317862

8 9 10 11 12 13 14

0.33554777 3.06851781 1.75603525 0.37882979 0.50293999 0.54398230 -0.95131402

15 16 17 18 19 20 21

-1.25479422 -1.68686948 0.84509878 -0.01004265 0.49305350 0.09490958 -3.16578640

> sqrt(sum(reg\_log$residuals^2)/nrow(del\_time)) ###RMSE

[1] 1.333748

> confint(reg\_log, level = 0.95)

2.5 % 97.5 %

(Intercept) -18.257239 -6.741226

log(Delivery.Time) 4.673307 8.797790

> predict(reg\_log,interval = "confidence")

fit lwr upr

1 8.007296 7.15901765 8.855574

2 5.031310 4.29911624 5.763503

3 7.593942 6.82271303 8.365171

4 8.906703 7.85705245 9.956353

5 10.181351 8.80179248 11.560910

6 5.896328 5.24959729 6.543058

7 7.333179 6.60341521 8.062942

8 2.664452 1.40921259 3.919692

9 6.931482 6.25205434 7.610910

10 7.243965 6.52689590 7.961034

11 7.621170 6.84526544 8.397075

12 3.497060 2.45293434 4.541186

13 6.456018 5.81044858 7.101587

14 3.951314 3.01314746 4.889481

15 4.254794 3.38221913 5.127369

16 5.686869 5.02814017 6.345599

17 5.154901 4.44028266 5.869520

18 7.010043 6.32220919 7.697876

19 1.506947 -0.06353677 3.077430

20 6.905090 6.22831795 7.581863

21 8.165786 7.28492549 9.046647

> ## Exponential Transformation ##

> ## x =Delivery.Time and y = log(Sorting.Time)

> plot(Delivery.Time, log(Sorting.Time))

> cor(Delivery.Time, log(Sorting.Time))

[1] 0.8339325

> reg\_exp <- lm(log(Sorting.Time) ~ Delivery.Time) ## lm(log(Y) ~ X)

> summary(reg\_exp)

Call:

lm(formula = log(Sorting.Time) ~ Delivery.Time)

Residuals:

Min 1Q Median 3Q Max

-0.48116 -0.19522 0.04756 0.17410 0.48883

Coefficients:

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

(Intercept) 0.43724 0.20438 2.139 0.0456 \*

Delivery.Time 0.07690 0.01167 6.587 2.64e-06 \*\*\*

---

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

Residual standard error: 0.265 on 19 degrees of freedom

**Multiple R-squared: 0.6954, Adjusted R-squared: 0.6794**

F-statistic: 43.39 on 1 and 19 DF, p-value: 2.642e-06

> reg\_exp$residuals

1 2 3 4 5 6 7

0.25043702 -0.08909959 -0.16426292 -0.08562514 -0.36476737 0.17409950 0.04756317

8 9 10 11 12 13 14

-0.06917947 0.48882872 0.31810274 0.11726711 0.12237692 0.22597245 -0.22298057

15 16 17 18 19 20 21

-0.26373786 -0.19522235 0.29714038 0.11600466 -0.35929375 0.13753682 -0.48116044

> sqrt(mean(reg\_exp$residuals^2))

[1] 0.2520376

> logat <- predict(reg\_exp)

> logat

1 2 3 4 5 6 7 8 9 10

2.052148 1.475394 1.956022 2.282850 2.667352 1.617660 1.898347 1.167792 1.813756 1.879122

11 12 13 14 15 16 17 18 19 20

1.962174 1.263917 1.719938 1.321593 1.362350 1.581517 1.494619 1.829905 1.052441 1.808373

21

2.090598

> at <- exp(logat)

> error = del\_time$Sorting.Time -Sorting.Time

> error

[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

> sqrt(sum(error^2)/nrow(del\_time))

[1] 0

> confint(reg\_exp,level = 0.95)

2.5 % 97.5 %

(Intercept) 0.00945716 0.8650159

Delivery.Time 0.05246456 0.1013365

> predict(reg\_exp,interval = "confidence")

fit lwr upr

1 2.052148 1.8933249 2.210971

2 1.475394 1.3300900 1.620698

3 1.956022 1.8150452 2.097000

4 2.282850 2.0691241 2.496575

5 2.667352 2.3454005 2.989304

6 1.617660 1.4916201 1.743700

7 1.898347 1.7658325 2.030861

8 1.167792 0.9524135 1.383170

9 1.813756 1.6897375 1.937775

10 1.879122 1.7489762 2.009267

11 1.962174 1.8201847 2.104164

12 1.263917 1.0730329 1.454802

13 1.719938 1.5988858 1.840990

14 1.321593 1.1444996 1.498686

15 1.362350 1.1944785 1.530222

16 1.581517 1.4517988 1.711235

17 1.494619 1.3526050 1.636633

18 1.829905 1.7046652 1.955146

19 1.052441 0.8058807 1.299001

20 1.808373 1.6847170 1.932030

21 2.090598 1.9236032 2.257594

> plot(Delivery.Time, Sorting.Time)

> plot(Delivery.Time\*Delivery.Time, Sorting.Time)

> cor(Delivery.Time\*Delivery.Time, Sorting.Time)

[1] 0.7763201

> plot(Delivery.Time\*Delivery.Time, log(Sorting.Time))

> cor(Delivery.Time, log(Sorting.Time))

[1] 0.8339325

> cor(Delivery.Time\*Delivery.Time, log(Sorting.Time))

[1] 0.7597441

> # lm(Y ~ X + I(X\*X) +....+ I(X\*X\*X))

> reg2degree <- lm(log(Sorting.Time) ~ Delivery.Time + I(Delivery.Time\*Delivery.Time))

> summary(reg2degree)

Call:

lm(formula = log(Sorting.Time) ~ Delivery.Time + I(Delivery.Time \*

Delivery.Time))

Residuals:

Min 1Q Median 3Q Max

-0.52375 -0.12486 0.03022 0.12204 0.38167

Coefficients:

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

(Intercept) -0.778048 0.449624 -1.730 0.100659

Delivery.Time 0.227222 0.052283 4.346 0.000389 \*\*\*

I(Delivery.Time \* Delivery.Time) -0.004271 0.001459 -2.928 0.008988 \*\*

---

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

Residual standard error: 0.2241 on 18 degrees of freedom

**Multiple R-squared: 0.7937, Adjusted R-squared: 0.7708**

F-statistic: 34.62 on 2 and 18 DF, p-value: 6.771e-07

> logpol <- predict(reg2degree)

> expy <- exp(logpol)

> expy

1 2 3 4 5 6 7 8 9 10

8.250905 4.531944 7.720077 9.165723 9.206382 5.493602 7.370749 2.705128 6.827202 7.250110

11 12 13 14 15 16 17 18 19 20

7.756102 3.225473 6.195767 3.561679 3.809137 5.245998 4.659292 6.933311 2.152041 6.791625

21

8.441717

> err = del\_time$Sorting.Time -expy

> err

1 2 3 4 5 6 7

1.74909450 -0.53194353 -1.72007652 -0.16572344 0.79361768 0.50639805 -0.37074885

8 9 10 11 12 13 14

0.29487227 3.17279834 1.74988962 0.24389783 0.77452719 0.80423276 -0.56167890

15 16 17 18 19 20 21

-0.80913710 -1.24599819 1.34070765 0.06668872 -0.15204121 0.20837459 -3.44171704

> sqrt(sum(err^2)/nrow(del\_time)) #RMSE

[1] 1.347142

> confint(reg2degree, level = 0.95)

2.5 % 97.5 %

(Intercept) -1.722672486 0.166577416

Delivery.Time 0.117379511 0.337064681

I(Delivery.Time \* Delivery.Time) -0.007334905 -0.001206109

> predict(reg2degree,interval = "confidence")

fit lwr upr

1 2.1103230 1.9692002 2.251446

2 1.5111509 1.3851779 1.637124

3 2.0438243 1.9085912 2.179057

4 2.2154708 2.0277304 2.403211

5 2.2198970 1.7982698 2.641524

6 1.7035841 1.5801073 1.827061

7 1.9975193 1.8644207 2.130618

8 0.9951491 0.7743170 1.215981

9 1.9209149 1.7905555 2.051274

10 1.9810167 1.8485446 2.113489

11 2.0484799 1.9129798 2.183980

12 1.1710795 0.9958979 1.346261

13 1.8238664 1.6969070 1.950826

14 1.2702320 1.1154651 1.424999

15 1.3374027 1.1937950 1.481010

16 1.6574655 1.5346128 1.780318

17 1.5388636 1.4142127 1.663514

18 1.9363375 1.8054444 2.067231

19 0.7664168 0.4732926 1.059541

20 1.9156903 1.7855121 2.045869

21 2.1331857 1.9881853 2.278186

> ggplot(data = del\_time, aes(x = Delivery.Time + I(Delivery.Time^2), y = log(Sorting.Time))) +

+ geom\_point(color='blue') +

+ geom\_line(color='red',data = del\_time, aes(x=Delivery.Time+I(Delivery.Time^2), y=logpol))

> reg3degree <- lm(log(Sorting.Time)~Delivery.Time + I(Delivery.Time\*Delivery.Time) + I(Delivery.Time\*Delivery.Time\*Delivery.Time))

> summary(reg3degree)

Call:

lm(formula = log(Sorting.Time) ~ Delivery.Time + I(Delivery.Time \*

Delivery.Time) + I(Delivery.Time \* Delivery.Time \* Delivery.Time))

Residuals:

Min 1Q Median 3Q Max

-0.48903 -0.15701 0.02729 0.12037 0.37952

Coefficients:

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

(Intercept) -1.5556893 1.4308664 -1.087 0.292

Delivery.Time 0.3747311 0.2625857 1.427 0.172

I(Delivery.Time \* Delivery.Time) -0.0128287 0.0149916 -0.856 0.404

I(Delivery.Time \* Delivery.Time \* Delivery.Time) 0.0001537 0.0002679 0.574 0.574

Residual standard error: 0.2284 on 17 degrees of freedom

**Multiple R-squared: 0.7976, Adjusted R-squared: 0.7619**

F-statistic: 22.33 on 3 and 17 DF, p-value: 3.934e-06

> logpol3 <- predict(reg3degree)

> expy3 <- exp(logpol3)

> ggplot(data = del\_time, aes(x = Delivery.Time + I(Delivery.Time^2) + I(Delivery.Time^3), y= Sorting.Time))+

+ geom\_point(color='blue') +

+ geom\_line(color='red', data = del\_time, aes(x=Delivery.Time+I(Delivery.Time^2)+I(Delivery.Time^3), y=expy3))