# ISyE 6414 Computer Project 2

## Kaustubh Mohite (903235166)

1. **Motivation Studies**

1.1

“Adaptive Estimation of Daily Demands with Complex Calendar Effects for Freight Transportation”

The purpose of the paper is to propose a class of exponential smoothing-based methods (called called damped trend multi-calendar (DTMC) exponential smoothing) for forecasting ‘spatial activities’ on a daily basis.

These methods can model multiple periodic and non-periodic ‘calendar effects’, are easier to apply than more sophisticated Auto-regressive Integrated Moving Average (ARIMA) methods, and actually outperform ARIMA methods in terms of forecast accuracy (3 percent lower RMSE than ARIMA; 2 percent lower than single exponential smoothing).

There are three sets of experiments in the study:

* Examining the validity of modelling multiple calendar effects (using only one day-of-the-week effect vs. multiple effects and the corresponding RMSE improvements)
* The relationship between trend-adjusted models and the need for control charts; Conclusion: using dampened trend gives better results than using full trend, and control charts give a slight performance improvement for testing sets
* Testing the effect of each of the control charts under a range of control parameters: EWMA charts give the most consistent and accurate results; Trig charts improve accuracy, but are sensitive to control parameters; Stewart charts provide no improvement

1.2

“Forecasting and Risk Analysis in Supply Chain Management”

The purpose of the paper to explore the application of autoregressive forecasting models to supply chain problems, and, in particular, generalized autoregressive conditional heteroscedasticity (GARCH), of which the writers present a Proof of Concept. The data used is from a supply chain inventory model (spare parts inventory management). The paper uses the ‘4-stage Beer Distribution Game’ simulation model to illustrate Demand Amplification or the Bullwhip effect.

The writers identify four main domains in which GARCH may minimise forecasting errors and fiscal losses:

* Cost of personnel, supply, support
* Planning, programming, budgeting
* Defense program and fiscal guidance development
* Force planning and financial program development

The writers compare the performance of the Classical Linear Regression Model (CLRM) with the Auto-regressive Moving Average (ARMA) model, and GARCH, and find that CLRM is always outperformed by the latter two.

The paper’s temporary conclusion is that GARCH and VAR‐ MGARCH techniques are promising (and often outperform ARMA), in high-volume and rich data environments that will be made feasible through real time capture (e.g. with RFID tagging and other Automatic Identification Technologies).

1. **Model Fitting**

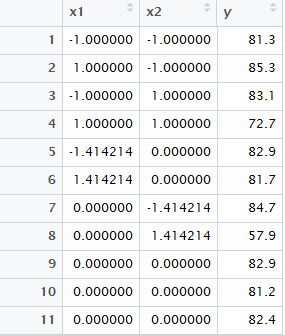
*MBT Synthesis*

**Raw Data**:

y: dependent variable

x1: [ time (in hours) – 12] / 5.6

x2 = [ Temperature (in Celsius) – 250] / 20



**Second order polynomial regression model:**

Call:

lm(formula = y ~ x1 + x2 + I(x1 \* x2) + I(x1 \* x1) + I(x2 \* x2), data = dfP2)

Residuals:

1 2 3 4 5 6 7 8 9 10 11

-0.8997 -2.0755 5.8755 4.6997 -2.7314 -1.0686 2.8908 -6.6908 0.7333 -0.9667 0.2333

Coefficients:

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

(Intercept) 82.167 2.887 28.466 1e-06 \*\*\*

x1 -1.012 1.768 -0.573 0.5917

x2 -6.088 1.768 -3.444 0.0184 \*

I(x1 \* x2) -3.600 2.500 -1.440 0.2094

I(x1 \* x1) 1.017 2.104 0.483 0.6493

I(x2 \* x2) -4.483 2.104 -2.131 0.0863 .

---

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

Residual standard error: 5 on 5 degrees of freedom

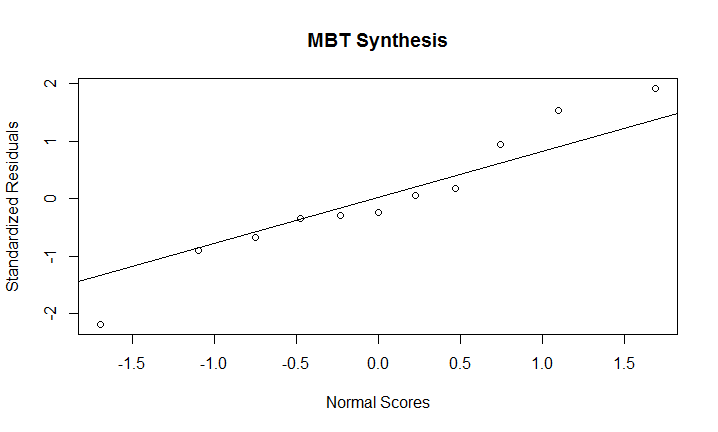
Multiple R-squared: 0.8012, Adjusted R-squared: 0.6024

F-statistic: 4.031 on 5 and 5 DF, p-value: 0.07614

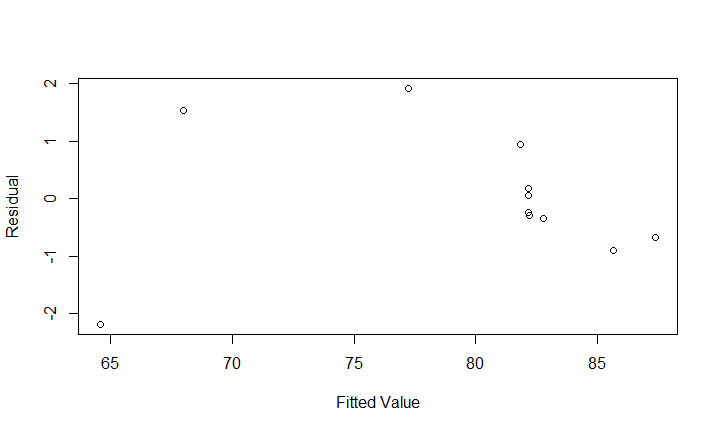
Y = -1.012(x1) -6.088(x2) - 3.6(x1\*x2) +1.017(x1\*x1) -4.483(x2\*x2)

**Diagnostics:**

1. Normal Probability plot of Standardised Residuals:

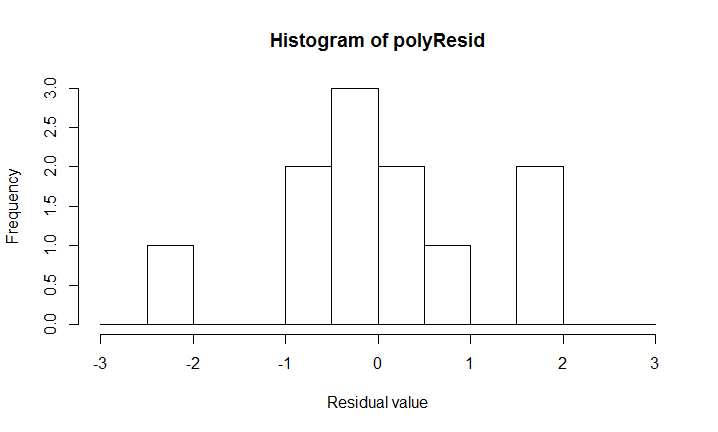


Inference: Standardised residuals approximate the normal distribution.

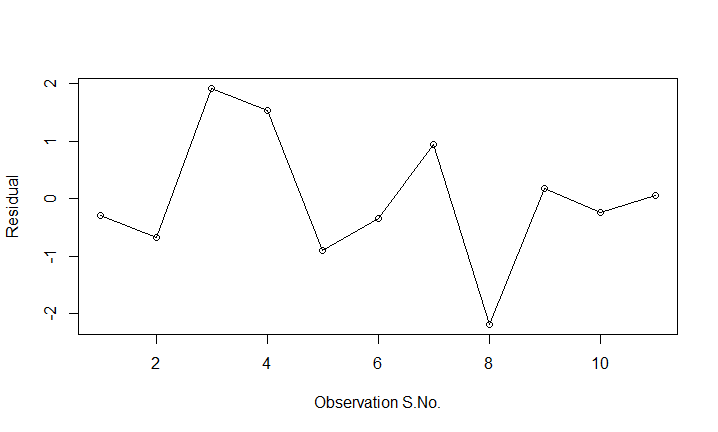
1. Standardised fitted values vs. residuals:

Inference: Residuals seem to be random w.r.t. Fitted Values

1. Histogram of residual values:



Inference: Residuals are approximately normally distributed, with mean = 0

1. Observation sequence vs. residuals:

Inference: no trend seen

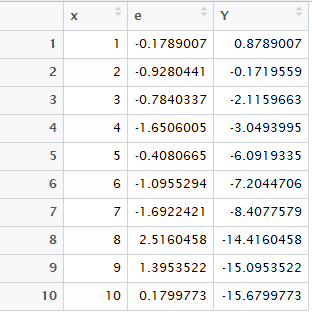
1. **Simulation Studies**

A]

**Raw Data:**

e: simulated normal distribution with mean = 0, s.d. = 1

Y = 2.5 – 1.8 x + *e*



**Simple Linear Regression Model:**

Y ~ x

Call:

lm(formula = Y ~ x)

Residuals:

Min 1Q Median 3Q Max

-2.24759 -0.70597 0.01161 0.83188 1.74747

Coefficients:

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

(Intercept) 3.9373 0.8478 4.644 0.00166 \*\*

x -2.0132 0.1366 -14.734 4.43e-07 \*\*\*

---

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

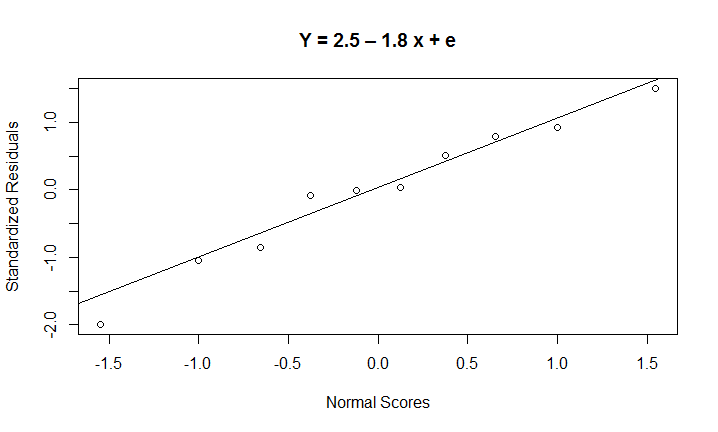
Residual standard error: 1.241 on 8 degrees of freedom

Multiple R-squared: 0.9645, Adjusted R-squared: 0.96

F-statistic: 217.1 on 1 and 8 DF, p-value: 4.426e-07

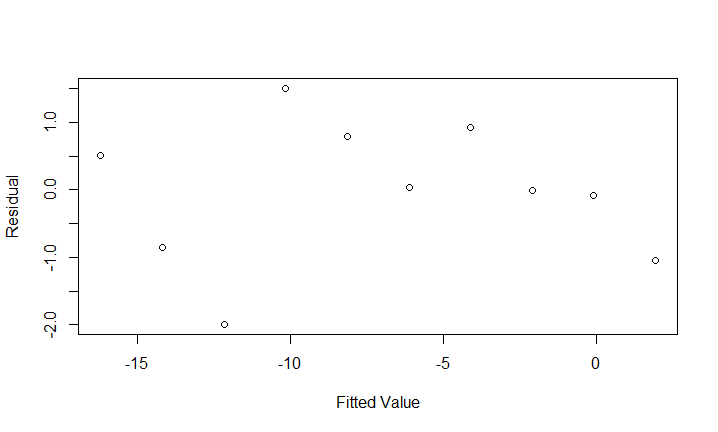
**Diagnostics:**

1. Normal Probability plot of Standardised Residuals:



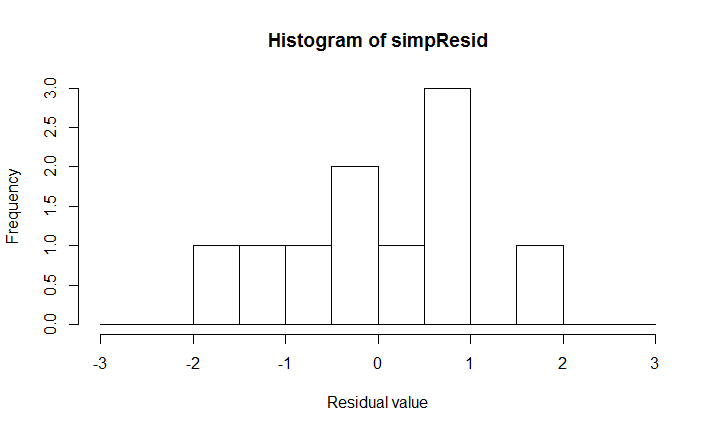
Inference: Standardised residuals approximate the normal distribution.

1. Standardised fitted values vs. residuals:



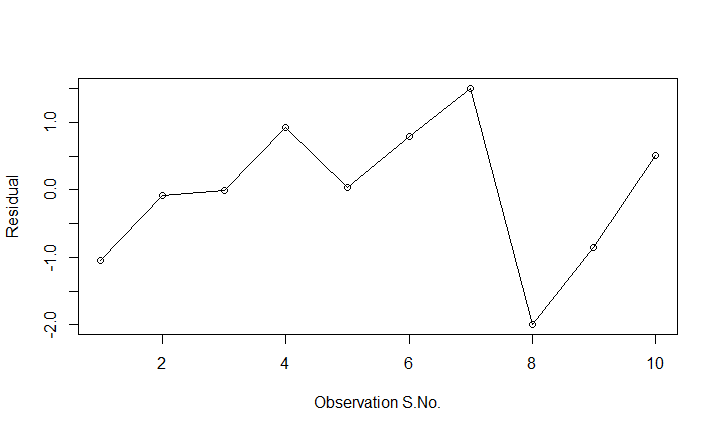
Inference: Residuals seem to be random w.r.t. Fitted Values

1. Histogram of residual values:



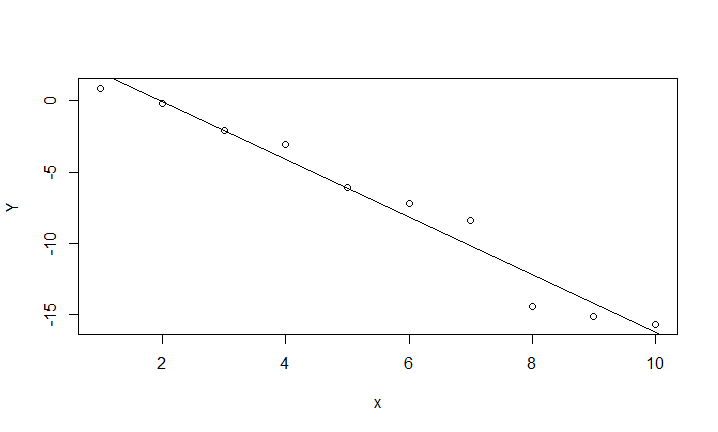
Inference: Residuals are approximately normally distributed, with mean = 0

1. Observation sequence vs. residuals:



Inference: no trend seen

**Plotting the regression line over the data**



**Comment: The regression line seems to be well-balanced, with positive as well as negative residual values (points above as well as below)**

**B] Finding alternative sigma**

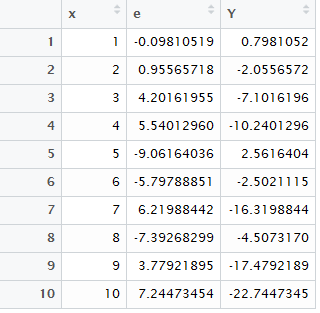
By simulating multiple data sets with different values of *sigma*, and examining the respective regression models (value of p statistic for the coefficient estimate of x) we arrive at:

*sigma = 8*

**Updated raw data**

e: simulated normal distribution with mean = 0, s.d. = 1

Y = 2.5 – 1.8 x + *e*



**Regression Model:**

Call:

lm(formula = Y ~ x)

Residuals:

Min 1Q Median 3Q Max

-17.9808 -5.6477 0.0929 6.6551 13.9798

Coefficients:

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

(Intercept) 13.999 6.782 2.064 0.0729 .

x -3.506 1.093 -3.207 0.0125 \*

---

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

Residual standard error: 9.928 on 8 degrees of freedom

Multiple R-squared: 0.5625, Adjusted R-squared: 0.5078

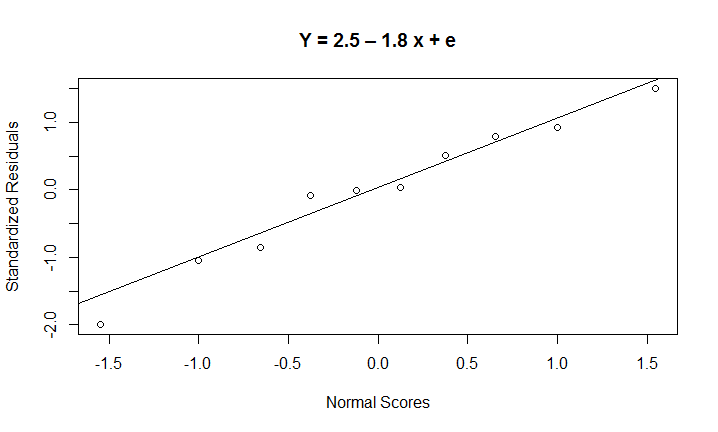
F-statistic: 10.29 on 1 and 8 DF, p-value: 0.01248

**Inference: p statistic of the estimate of the coefficient x is 0.0125**

**Hence, the estimate is statistically significant for alpha=0.1 (10%), but insignificant for alpha = 0.01 (1%)**

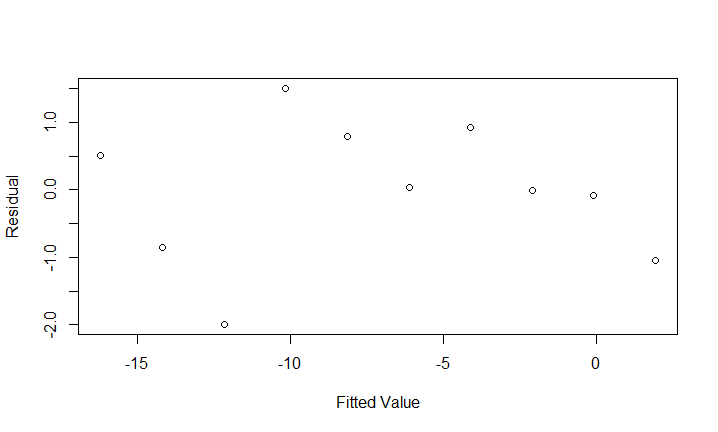
**Diagnostics:**

1. Normal Probability plot of Standardised Residuals:



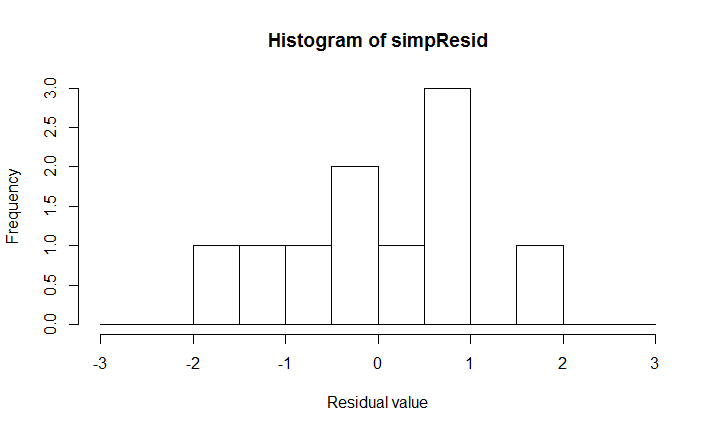
Inference: Standardised residuals approximate the normal distribution.

1. Standardised fitted values vs. residuals:



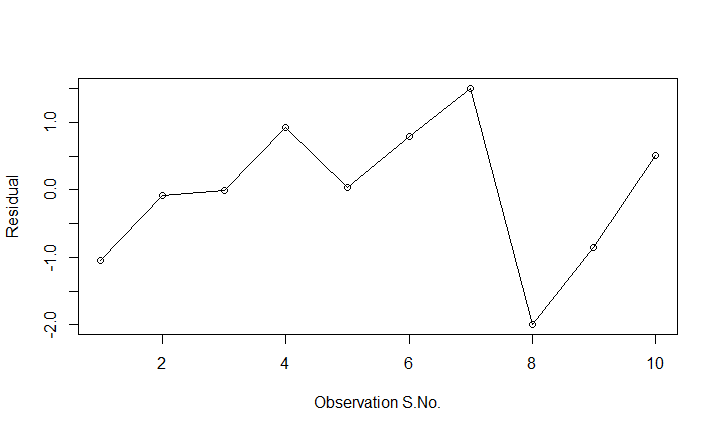
Inference: Residuals seem to be random w.r.t. Fitted Values

1. Histogram of residual values:



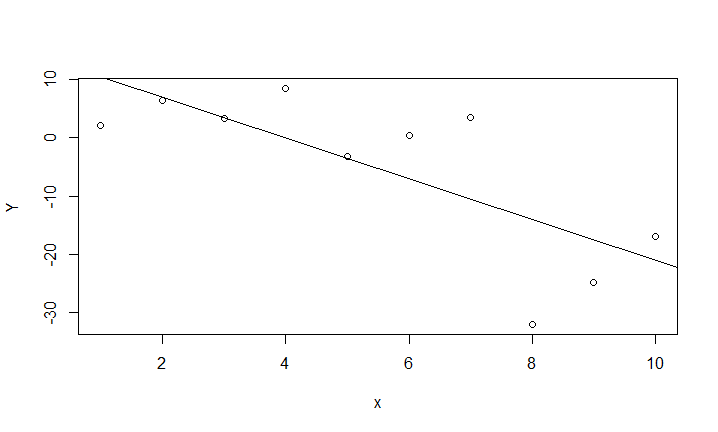
Inference: Residuals are approximately normally distributed, with mean = 0

1. Observation sequence vs. residuals:



Inference: no trend seen

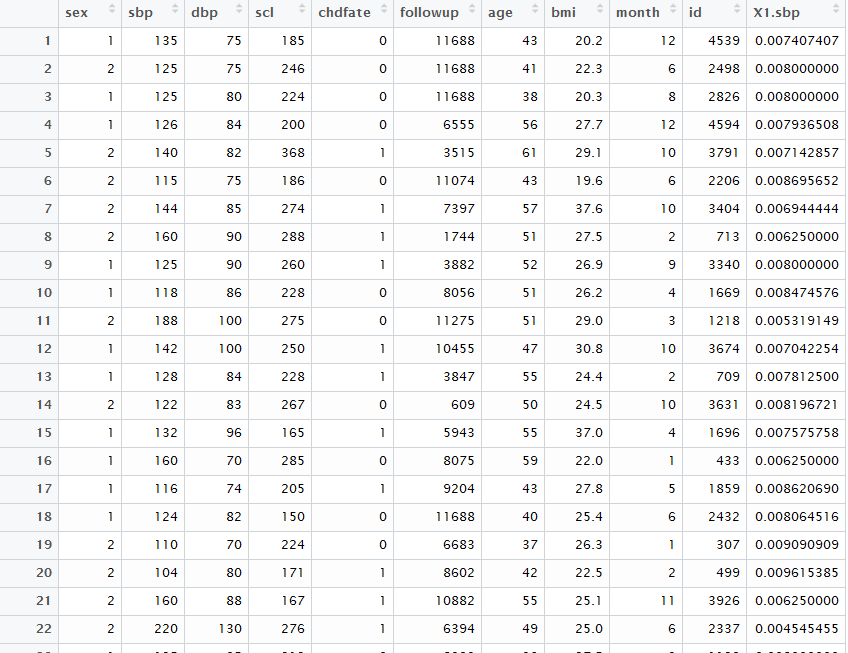
**Plotting the regression line over the data**



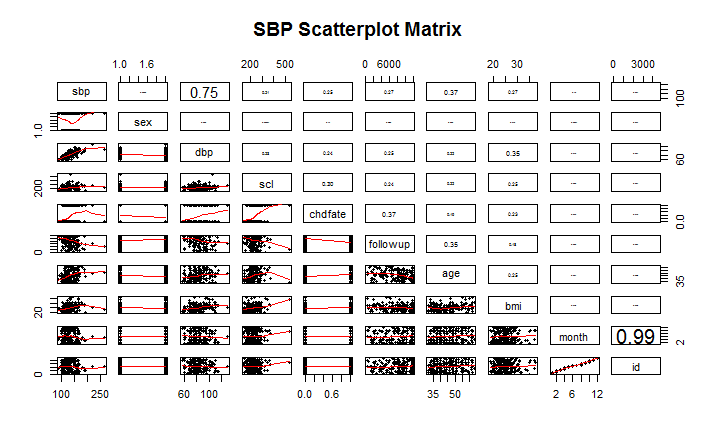
**Comment: The regression line seems to be fairly balanced, with positive as well as negative residual values (points above as well as below). The residual values are larger than the earlier case (when e: sd = 1)**

1. **Real-life Data Analysis for Variable Selections**

Sample of Raw Data:

****

**Matrix scatter plot – variable correlation:**

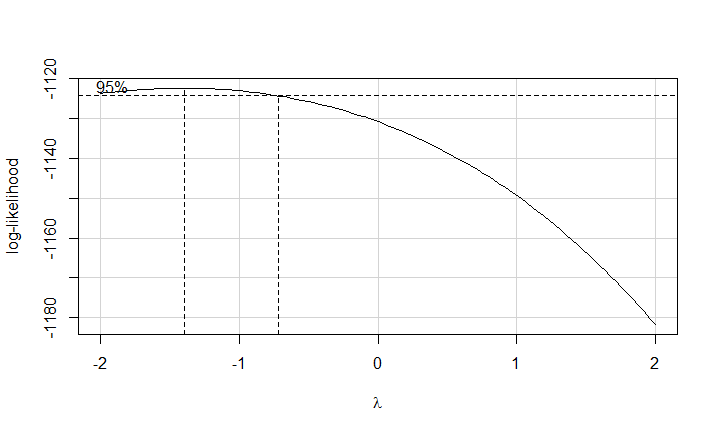


Inference: Variables *month* and *id* are almost identical: correlation coefficient = 0.99

Variables *sbp* and *dbp* are highly correlated: correlation coefficient = 0.75

Variables bmi and scl are not evenly distributed – crowded towards the left. Hence using inverse transformation.

**Box Cox Transform of dependent variable *sbp***



Inference: Based on the plot, we select lambda = -1, and take inverse of variable *sbp*

**Eliminating multicollinearity using VIF:**

We calculate VIF value for each independent variable, based on a full regression model (i.e. model involving all available independent variables)

vif(linRegFullSBP)

sex dbp scl\_inv chdfate followup age bmi\_inv month id

1.042867 1.258584 1.193355 1.295479 1.334374 1.251730 1.222040 61.268017 60.971757

We find that variables month and id have very high VIF, hence we eliminate one of them – id

vif(linRegFullSBP)

sex dbp scl\_inv chdfate followup age bmi\_inv month

1.028791 1.249625 1.192292 1.265258 1.328000 1.251716 1.215426 1.041857

**Forward Stepwise Variable Selection:**

We start with a ‘null’ regression model (output ~ 1), and use forward stepwise selection (based on AIC value).

Start: AIC=-2672.4

X1.sbp ~ 1

Df Sum of Sq RSS AIC

+ dbp 1 1.6918e-04 0.00012075 -2844.7

+ age 1 4.4971e-05 0.00024496 -2703.9

+ bmi\_inv 1 2.9537e-05 0.00026039 -2691.8

+ chdfate 1 2.3017e-05 0.00026691 -2686.9

+ scl\_inv 1 2.1798e-05 0.00026813 -2685.9

+ followup 1 2.1006e-05 0.00026892 -2685.4

<none> 0.00028993 -2672.4

+ month 1 2.4620e-06 0.00028747 -2672.1

+ id 1 1.6520e-06 0.00028828 -2671.5

+ sex 1 5.5900e-07 0.00028937 -2670.8

Step: AIC=-2844.7

X1.sbp ~ dbp

Df Sum of Sq RSS AIC

+ age 1 1.5807e-05 0.00010495 -2870.6

+ chdfate 1 3.0492e-06 0.00011770 -2847.8

+ scl\_inv 1 2.7441e-06 0.00011801 -2847.3

+ followup 1 2.0307e-06 0.00011872 -2846.1

<none> 0.00012075 -2844.7

+ bmi\_inv 1 6.7770e-07 0.00012008 -2843.8

+ month 1 3.6070e-07 0.00012039 -2843.3

+ sex 1 2.5460e-07 0.00012050 -2843.1

+ id 1 2.4670e-07 0.00012051 -2843.1

Step: AIC=-2870.62

X1.sbp ~ dbp + age

Df Sum of Sq RSS AIC

+ chdfate 1 1.3767e-06 0.00010357 -2871.2

+ month 1 1.0958e-06 0.00010385 -2870.7

<none> 0.00010495 -2870.6

+ id 1 8.8473e-07 0.00010406 -2870.3

+ scl\_inv 1 5.1993e-07 0.00010443 -2869.6

+ sex 1 3.0894e-07 0.00010464 -2869.2

+ followup 1 4.1900e-08 0.00010490 -2868.7

+ bmi\_inv 1 1.2640e-08 0.00010493 -2868.6

Step: AIC=-2871.25

X1.sbp ~ dbp + age + chdfate

Df Sum of Sq RSS AIC

+ month 1 1.1788e-06 0.00010239 -2871.5

<none> 0.00010357 -2871.2

+ id 1 9.1965e-07 0.00010265 -2871.0

+ sex 1 4.5820e-07 0.00010311 -2870.1

+ scl\_inv 1 2.3211e-07 0.00010334 -2869.7

+ followup 1 2.4570e-08 0.00010354 -2869.3

+ bmi\_inv 1 7.6000e-10 0.00010357 -2869.2

Step: AIC=-2871.52

X1.sbp ~ dbp + age + chdfate + month

Df Sum of Sq RSS AIC

<none> 0.00010239 -2871.5

+ id 1 8.0610e-07 0.00010158 -2871.1

+ sex 1 4.4451e-07 0.00010195 -2870.4

+ scl\_inv 1 2.6444e-07 0.00010213 -2870.0

+ followup 1 9.8070e-08 0.00010229 -2869.7

+ bmi\_inv 1 3.0000e-11 0.00010239 -2869.5

Call:

lm(formula = X1.sbp ~ dbp + age + chdfate + month, data = dfSBPData)

Coefficients:

(Intercept) dbp age chdfate month

1.473e-02 -6.572e-05 -3.429e-05 -1.902e-04 2.126e-05

Final model:

sbp\_inv ~ dbp + age + chdfate + month

**All subsets variable selection:**

We start with the ‘full’ regression model, and use all subsets variable selection.

Criteria: Mallow’s Cp

Subset selection object

Call: regsubsets.formula(X1.sbp ~ sex + dbp + scl + chdfate + followup +

age + bmi + month, data = dfSBPData, nbest = 10)

8 Variables (and intercept)

Forced in Forced out

sex FALSE FALSE

dbp FALSE FALSE

scl FALSE FALSE

chdfate FALSE FALSE

followup FALSE FALSE

age FALSE FALSE

bmi FALSE FALSE

month FALSE FALSE

10 subsets of each size up to 8

Selection Algorithm: exhaustive

sex dbp scl chdfate followup age bmi month

1 ( 1 ) " " "\*" " " " " " " " " " " " "

1 ( 2 ) " " " " " " " " " " "\*" " " " "

1 ( 3 ) " " " " " " " " " " " " "\*" " "

1 ( 4 ) " " " " " " "\*" " " " " " " " "

1 ( 5 ) " " " " " " " " "\*" " " " " " "

1 ( 6 ) " " " " "\*" " " " " " " " " " "

1 ( 7 ) " " " " " " " " " " " " " " "\*"

1 ( 8 ) "\*" " " " " " " " " " " " " " "

2 ( 1 ) " " "\*" " " " " " " "\*" " " " "

2 ( 2 ) " " "\*" " " "\*" " " " " " " " "

2 ( 3 ) " " "\*" " " " " "\*" " " " " " "

2 ( 4 ) " " "\*" "\*" " " " " " " " " " "

2 ( 5 ) " " "\*" " " " " " " " " "\*" " "

2 ( 6 ) " " "\*" " " " " " " " " " " "\*"

2 ( 7 ) "\*" "\*" " " " " " " " " " " " "

2 ( 8 ) " " " " " " " " " " "\*" "\*" " "

2 ( 9 ) " " " " " " "\*" " " "\*" " " " "

2 ( 10 ) " " " " "\*" " " " " "\*" " " " "

3 ( 1 ) " " "\*" " " "\*" " " "\*" " " " "

3 ( 2 ) " " "\*" " " " " " " "\*" " " "\*"

3 ( 3 ) "\*" "\*" " " " " " " "\*" " " " "

3 ( 4 ) " " "\*" "\*" " " " " "\*" " " " "

3 ( 5 ) " " "\*" " " " " "\*" "\*" " " " "

3 ( 6 ) " " "\*" " " " " " " "\*" "\*" " "

3 ( 7 ) " " "\*" " " "\*" "\*" " " " " " "

3 ( 8 ) " " "\*" "\*" "\*" " " " " " " " "

3 ( 9 ) " " "\*" " " "\*" " " " " " " "\*"

3 ( 10 ) "\*" "\*" " " "\*" " " " " " " " "

4 ( 1 ) " " "\*" " " "\*" " " "\*" " " "\*"

4 ( 2 ) "\*" "\*" " " "\*" " " "\*" " " " "

4 ( 3 ) " " "\*" "\*" "\*" " " "\*" " " " "

4 ( 4 ) " " "\*" "\*" " " " " "\*" " " "\*"

4 ( 5 ) " " "\*" " " "\*" "\*" "\*" " " " "

4 ( 6 ) "\*" "\*" " " " " " " "\*" " " "\*"

4 ( 7 ) " " "\*" " " "\*" " " "\*" "\*" " "

4 ( 8 ) " " "\*" " " " " " " "\*" "\*" "\*"

4 ( 9 ) " " "\*" " " " " "\*" "\*" " " "\*"

4 ( 10 ) "\*" "\*" "\*" " " " " "\*" " " " "

5 ( 1 ) "\*" "\*" " " "\*" " " "\*" " " "\*"

5 ( 2 ) " " "\*" " " "\*" "\*" "\*" " " "\*"

5 ( 3 ) " " "\*" "\*" "\*" " " "\*" " " "\*"

5 ( 4 ) " " "\*" " " "\*" " " "\*" "\*" "\*"

5 ( 5 ) "\*" "\*" "\*" "\*" " " "\*" " " " "

5 ( 6 ) "\*" "\*" " " "\*" "\*" "\*" " " " "

5 ( 7 ) "\*" "\*" " " "\*" " " "\*" "\*" " "

5 ( 8 ) "\*" "\*" "\*" " " " " "\*" " " "\*"

5 ( 9 ) " " "\*" "\*" "\*" "\*" "\*" " " " "

5 ( 10 ) " " "\*" "\*" "\*" " " "\*" "\*" " "

6 ( 1 ) "\*" "\*" " " "\*" "\*" "\*" " " "\*"

6 ( 2 ) "\*" "\*" "\*" "\*" " " "\*" " " "\*"

6 ( 3 ) "\*" "\*" " " "\*" " " "\*" "\*" "\*"

6 ( 4 ) " " "\*" "\*" "\*" "\*" "\*" " " "\*"

6 ( 5 ) " " "\*" " " "\*" "\*" "\*" "\*" "\*"

6 ( 6 ) " " "\*" "\*" "\*" " " "\*" "\*" "\*"

6 ( 7 ) "\*" "\*" "\*" "\*" "\*" "\*" " " " "

6 ( 8 ) "\*" "\*" "\*" "\*" " " "\*" "\*" " "

6 ( 9 ) "\*" "\*" " " "\*" "\*" "\*" "\*" " "

6 ( 10 ) "\*" "\*" "\*" " " " " "\*" "\*" "\*"

7 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*"

7 ( 2 ) "\*" "\*" " " "\*" "\*" "\*" "\*" "\*"

7 ( 3 ) "\*" "\*" "\*" "\*" " " "\*" "\*" "\*"

7 ( 4 ) " " "\*" "\*" "\*" "\*" "\*" "\*" "\*"

7 ( 5 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" " "

7 ( 6 ) "\*" "\*" "\*" " " "\*" "\*" "\*" "\*"

7 ( 7 ) "\*" "\*" "\*" "\*" "\*" " " "\*" "\*"

7 ( 8 ) "\*" " " "\*" "\*" "\*" "\*" "\*" "\*"

8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" "\*" "\*"

[1] 30.443125 262.335719 291.784542 303.322984 307.077927 314.409607 341.698211 345.251380

[9] 2.932627 26.750270 28.651770 29.709875 31.105644 31.769773 31.967827 236.744818

[17] 240.407999 250.704721 2.362348 2.886762 4.355835 4.484827 4.854393 4.917190

[25] 27.238162 27.626258 27.871986 27.875341 2.161601 3.506902 4.280291 4.315312

[33] 4.316468 4.336998 4.351641 4.862286 4.876316 5.932746 3.331706 3.978503

[41] 4.027517 4.155719 5.449584 5.455222 5.501482 5.793662 6.223080 6.260792

[49] 5.138314 5.229669 5.329457 5.812300 5.973397 6.012582 7.387964 7.438721

[57] 7.450132 7.787414 7.007572 7.136563 7.221993 7.797472 9.377081 9.787386

[65] 32.445018 217.700972 9.000000

We find that the value of Cp  is lowest for the model:

sbp\_inv ~ dbp + age + chdfate

**Final Regression Model:**

sbp \_inv ~ dbp + chdfate + age

Call:

lm(formula = X1.sbp ~ dbp + chdfate + age, data = dfSBPData)

Residuals:

Min 1Q Median 3Q Max

-2.307e-03 -4.794e-04 3.369e-05 5.120e-04 1.897e-03

Coefficients:

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

(Intercept) 1.486e-02 4.070e-04 36.512 < 2e-16 \*\*\*

dbp -6.637e-05 4.266e-06 -15.559 < 2e-16 \*\*\*

chdfate -1.846e-04 1.147e-04 -1.610 0.109

age -3.328e-05 6.451e-06 -5.159 6.09e-07 \*\*\*

---

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

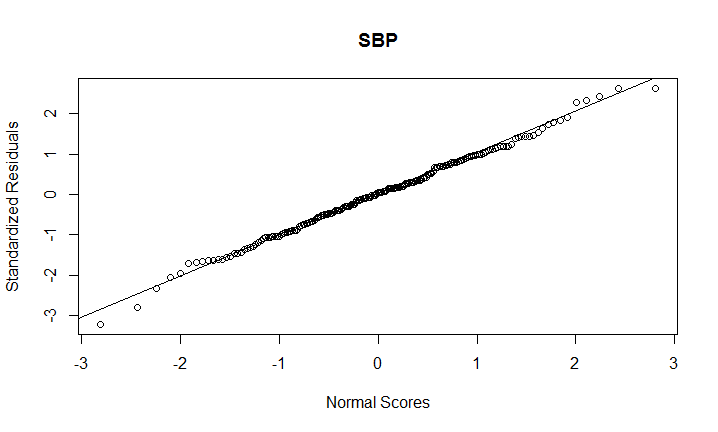
Residual standard error: 0.0007288 on 195 degrees of freedom

Multiple R-squared: 0.6428, Adjusted R-squared: 0.6373

F-statistic: 117 on 3 and 195 DF, p-value: < 2.2e-16

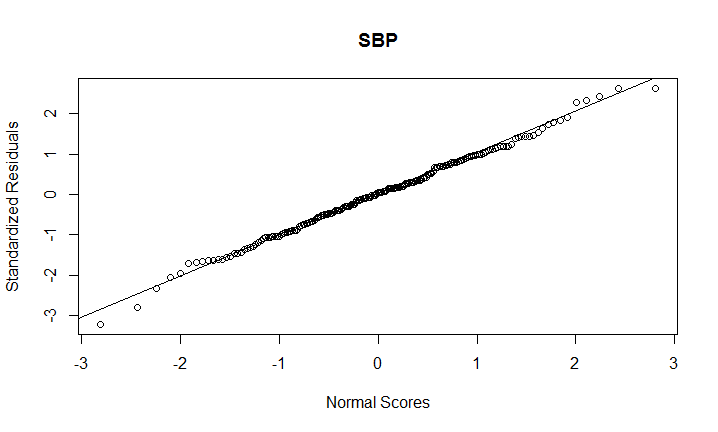
**Diagnostics:**

1. Normal Probability plot of Standardised Residuals:



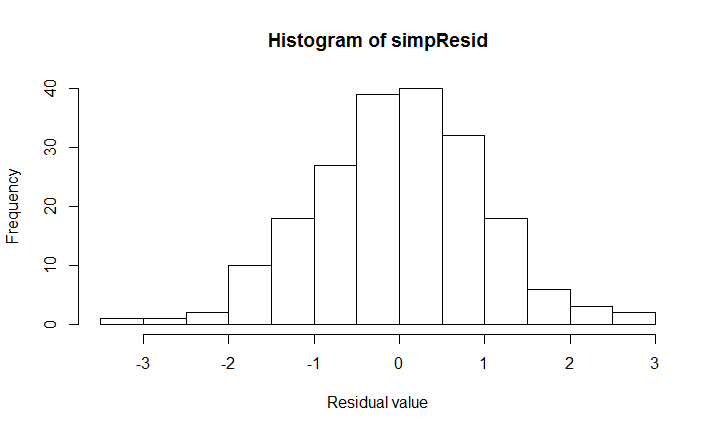
Inference: Standardised residuals approximate the normal distribution.

1. Standardised fitted values vs. residuals:



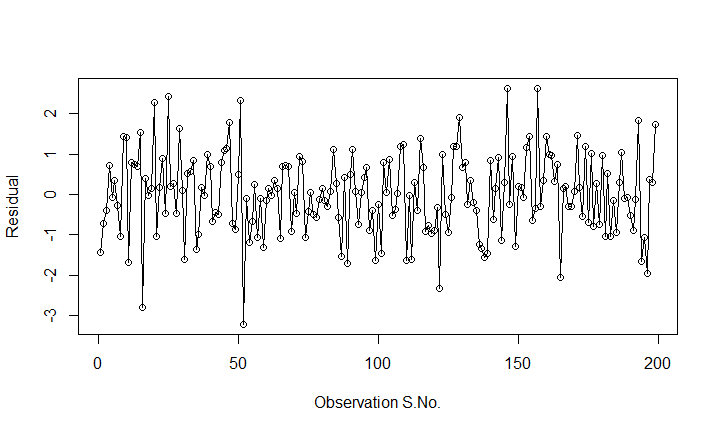
Inference: Residuals seem to be random w.r.t. Fitted Values

1. Histogram of residual values:



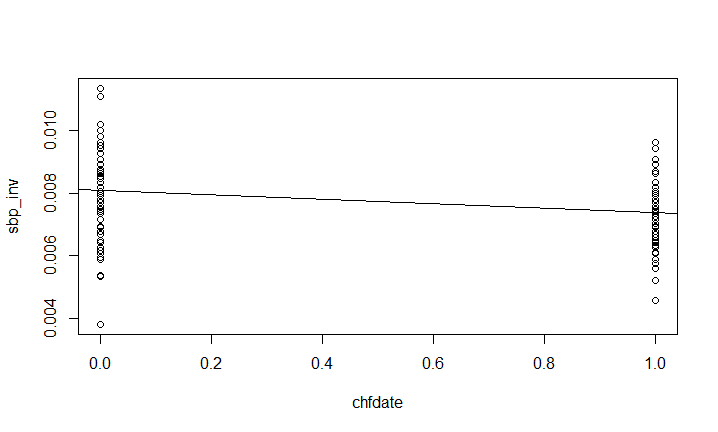
Inference: Residuals are approximately normally distributed, with mean = 0

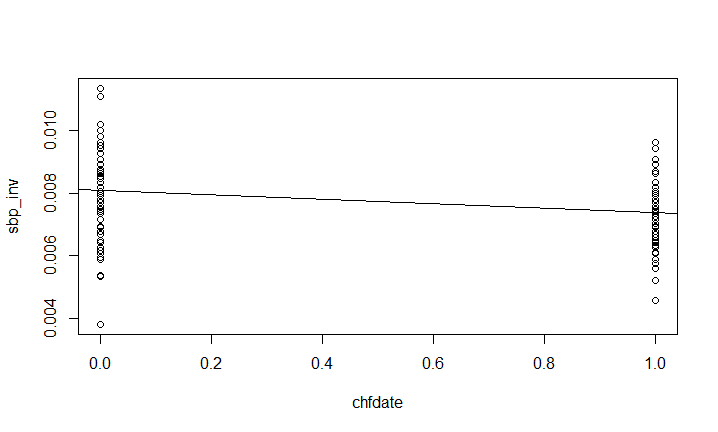
1. Observation sequence vs. residuals:

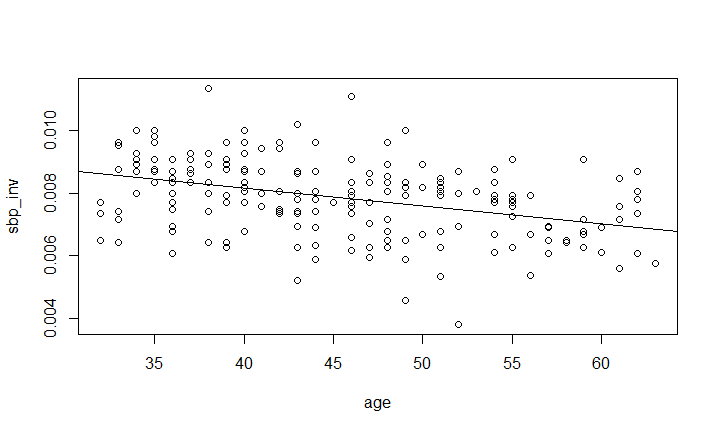


Inference: no trend seen

**Plotting the regression line over the data – one independent variable at a time:**







**Comment: The regression lines seem to be well-balanced, with positive as well as negative residual values (points above as well as below)**

**Appendix**

1. R code for Q2

****

setwd("C:/Users/Kaustubh/Documents/Regression Models/CP2")

x1 <- c(-1, 1, -1, 1, -1.4142136, 1.4142136, 0, 0, 0, 0, 0)

x2 <- c(-1, -1, 1, 1, 0, 0, -1.4142136, 1.4142136, 0, 0, 0)

y <- c(81.3, 85.3, 83.1, 72.7, 82.9, 81.7, 84.7, 57.9, 82.9, 81.2, 82.4)

dfP2 <- data.frame(x1, x2, y)

polyRegModel <- lm(y ~ x1 + x2 + I(x1\*x2) + I(x1\*x1) + I(x2\*x2), data = dfP2)

summary(polyRegModel)

#Calculate standardised residuals, and use in normal probability plot

polyResid <- rstandard(polyRegModel)

qqnorm(polyResid, ylab="Standardized Residuals", xlab="Normal Scores", main="MBT Synthesis")

qqline(polyResid)

# Standardised fitted values vs. residuals

plot(fitted.values(polyRegModel), polyResid, xlab = "Fitted Value", ylab = "Residual")

#histogram of residual values

hist(polyResid, breaks = seq(-3,3,0.5), xlab = "Residual value")

#Observation order vs. standardised residuals

plot(seq(1,11,1), polyResid, xlab = "Observation S.No.", ylab = "Residual", type = "o")

1. R code for Q3



set.seed(54321)

#Generating random errors - normal with mean 0 and sd = 1

e <- rnorm(10, 0, 1)

#Generating vector of input values

x <- seq(1,10,1)

#Calculating Y by formula: Y = 2.5 – 1.8 x + e

Y <- vector(mode = "numeric", length = 10)

Y <- 2.5 - (1.8\*x) - e

dfProb3 <- data.frame(x,e,Y)

#Fitting a simple regression model

simpRegModel <- lm(Y~x)

# examining the regression model

summary(simpRegModel)

#Calculate standardised residuals, and use in normal probability plot

simpResid <- rstandard(simpRegModel)

qqnorm(simpResid, ylab="Standardized Residuals", xlab="Normal Scores", main="Y = 2.5 – 1.8 x + e")

qqline(simpResid)

# Standardised fitted values vs. residuals

plot(fitted.values(simpRegModel), simpResid, xlab = "Fitted Value", ylab = "Residual")

#histogram of residual values

hist(simpResid, breaks = seq(-3,3,0.5), xlab = "Residual value")

#Observation order vs. standardised residuals

plot(seq(1,10,1), simpResid, xlab = "Observation S.No.", ylab = "Residual", type = "o")

#Plotting the regression line over the data

plot(x,Y)

abline(lm(Y ~ x))

###########################################

#Finding alternative sigma

###########################################

set.seed(54321)

#Generating random errors - normal with mean 0 and sd = 1

e <- rnorm(10, 0, 8)

#Generating vector of input values

x <- seq(1,10,1)

#Calculating Y by formula: Y = 2.5 – 1.8 x + e

Y <- vector(mode = "numeric", length = 10)

Y <- 2.5 - (1.8\*x) - e

dfProb3b <- data.frame(x,e,Y)

#Fitting a simple regression model

simpRegModel <- lm(Y~x)

# examining the regression model

summary(simpRegModel)

#Calculate standardised residuals, and use in normal probability plot

simpResid <- rstandard(simpRegModel)

qqnorm(simpResid, ylab="Standardized Residuals", xlab="Normal Scores", main="Y = 2.5 – 1.8 x + e")

qqline(simpResid)

# Standardised fitted values vs. residuals

plot(fitted.values(simpRegModel), simpResid, xlab = "Fitted Value", ylab = "Residual")

#histogram of residual values

hist(simpResid, breaks = seq(-3,3,0.5), xlab = "Residual value")

#Observation order vs. standardised residuals

plot(seq(1,10,1), simpResid, xlab = "Observation S.No.", ylab = "Residual", type = "o")

#Plotting the regression line over the data

plot(x,Y)

abline(lm(Y ~ x))

1. R code for Q4



#Reading the data set

dfSBPData <- read.csv("CP\_Data\_P4\_052114.csv", header = TRUE)

head(dfSBPData)

# matrix scatter plot of all possible input variables

# source: https://www.r-bloggers.com/scatterplot-matrices-in-r/

# panel.smooth function is built in.

# panel.cor puts correlation in upper panels, size proportional to correlation

panel.cor <- function(x, y, digits=2, prefix="", cex.cor, ...)

{

usr <- par("usr"); on.exit(par(usr))

par(usr = c(0, 1, 0, 1))

r <- abs(cor(x, y))

txt <- format(c(r, 0.123456789), digits=digits)[1]

txt <- paste(prefix, txt, sep="")

if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)

text(0.5, 0.5, txt, cex = cex.cor \* r)

}

# Plot #2: same as above, but add loess smoother in lower and correlation in upper

pairs(~sbp+sex+dbp+scl+chdfate+followup+age+bmi+month+id, data=dfSBPData,

lower.panel=panel.smooth, upper.panel=panel.cor,

pch=20, main="SBP Scatterplot Matrix")

#Based on the scatter plot, scl and bmi data points are crowded together respectively, hence we transform scl (inverse)

dfSBPData$scl\_inv <- (1/dfSBPData$scl)

dfSBPData$bmi\_inv <- (1/dfSBPData$bmi)

#Box Cox transformation of the dependent variable sbp

install.packages("car")

library(car)

boxCox(dfSBPData$sbp~1)

#From the graph, we get lambda = -1

#therefore, transformed sbp = 1/sbp

# fit first model - all input variables included

linRegFullSBP <- lm(X1.sbp ~ sex + dbp + scl\_inv + chdfate + followup + age + bmi\_inv + month + id, data = dfSBPData)

vif(linRegFullSBP)

# Eliminate input variable id, since VIF for month and id are high (very high correlation)

linRegFullSBP <- lm(X1.sbp ~ sex + dbp + scl\_inv + chdfate + followup + age + bmi\_inv + month, data = dfSBPData)

vif(linRegFullSBP)

# Forward Stepwise variable selection:

linRegNullSBP <- lm(X1.sbp ~ 1, data = dfSBPData)

step(linRegNullSBP, scope=list(lower=linRegNullSBP, upper = linRegFullSBP), direction="forward", trace = 10)

#All subsets variable selection:

install.packages("leaps")

library(leaps)

leaps<-regsubsets(X1.sbp ~ sex + dbp + scl + chdfate + followup + age + bmi + month, data=dfSBPData, nbest=10)

summary(leaps)

summary(leaps)$cp

#Final model

linRegFinSBP <- lm(X1.sbp ~ dbp + chdfate + age , data = dfSBPData)

summary(linRegFinSBP)

#Diagnostics

#Calculate standardised residuals, and use in normal probability plot

simpResid <- rstandard(linRegFinSBP)

qqnorm(simpResid, ylab="Standardized Residuals", xlab="Normal Scores", main="SBP")

qqline(simpResid)

# Standardised fitted values vs. residuals

plot(fitted.values(linRegFinSBP), simpResid, xlab = "Fitted Value", ylab = "Residual")

#histogram of residual values

hist(simpResid, xlab = "Residual value")

#Observation order vs. standardised residuals

plot(seq(1,199,1), simpResid, xlab = "Observation S.No.", ylab = "Residual", type = "o")

#Plotting the regression line over the data

plot(dfSBPData$chdfate,dfSBPData$X1.sbp, xlab = "chfdate", ylab = "sbp\_inv")

abline(lm(dfSBPData$X1.sbp ~ dfSBPData$chdfate))

plot(dfSBPData$dbp,dfSBPData$X1.sbp, xlab = "dbp", ylab = "sbp\_inv")

abline(lm(dfSBPData$X1.sbp ~ dfSBPData$dbp))

plot(dfSBPData$age,dfSBPData$X1.sbp, xlab = "age", ylab = "sbp\_inv")

abline(lm(dfSBPData$X1.sbp ~ dfSBPData$age))