PS3 Jeremy Tan

Loading CSV

```r
data_1 = read.csv("PS3.csv")
data_1 = data_1[-1] # remove ID column
data_1
```

```
##          y  x01  x02  x03  x04  x05  x06 x07 x08 x09  x10    x11    x12
## 1  25.3209 1.61 1.15 0.53 0.89 1.93 2.66  56  47  53 3.51 10.490  4.940
## 2  11.1852 1.61 0.76 0.53 0.89 0.98 2.20  56   3  76 3.51  0.535 22.590
## 3  16.2945 1.61 1.02 0.53 0.89 0.93 2.66  56  11  53 3.51  0.968  4.940
## 4  14.8078 1.61 1.02 0.53 0.89 0.93 2.20  56  11  76 3.51  0.968 22.590
## 5   9.0212 1.34 0.76 0.53 1.00 0.98 2.20  20   3  76 1.55  0.535 22.590
## 6   9.0514 1.34 0.76 0.53 1.00 0.98 1.90  20   3  75 1.55  0.535 21.020
## 7   9.2891 1.44 0.76 0.53 0.95 0.98 1.90  38   3  75 2.63  0.535 21.020
## 8  11.7087 1.44 1.02 0.53 0.95 0.93 2.20  38  11  76 2.63  0.968 22.590
## 9  32.4948 1.61 1.03 0.64 0.89 1.90 1.50  56  83  73 3.51  9.780 16.650
## 10 45.3982 1.61 1.01 0.78 0.89 1.12 1.50  56  58  91 3.51  6.689 15.370
## 11 23.6934 1.61 0.91 0.64 0.89 1.22 1.60  56  66  41 3.51  8.551  8.570
## 12 41.0378 1.61 0.91 0.78 0.89 1.22 1.50  56  66  91 3.51  8.551 15.370
## 13 22.2316 1.61 0.89 0.64 0.89 1.24 1.60  56  68  41 3.51  9.066  8.570
## 14 38.6801 1.61 0.89 0.78 0.89 1.24 1.50  56  68  91 3.51  9.066 15.370
## 15 15.7121 1.61 0.89 0.57 0.89 1.24 2.20  56  68  44 3.51  9.066 12.370
## 16 21.8867 1.61 0.89 0.64 0.89 1.24 1.50  56  68  73 3.51  9.066 16.650
## 17 27.1412 1.61 0.95 0.64 0.89 1.12 1.60  56  63  41 3.51  5.244  8.570
## 18 43.5963 1.61 0.95 0.78 0.89 1.12 1.50  56  63  91 3.51  5.244 15.370
## 19 10.3569 1.61 0.65 0.61 0.89 1.83 2.16  56  26  42 3.51  7.874 10.280
## 20 10.0140 1.61 0.65 0.58 0.89 1.83 1.90  56  26  75 3.51  7.874 21.020
## 21 42.7278 1.61 0.94 0.78 0.89 1.20 1.50  56  64  91 3.51  7.901 15.370
## 22 22.5850 1.61 0.94 0.58 0.89 1.20 1.90  56  64  75 3.51  7.901 21.020
## 23 23.3137 1.61 0.90 0.64 0.89 1.23 1.60  56  67  41 3.51  8.795  8.570
## 24 40.2304 1.61 0.90 0.78 0.89 1.23 1.50  56  67  91 3.51  8.795 15.370
## 25 14.3286 1.61 0.80 0.64 0.89 1.78 1.60  56  49  41 3.51  7.310  8.570
## 26 13.5628 1.61 0.80 0.58 0.89 1.78 2.20  56  49  76 3.51  7.310 22.590
## 27 14.0126 1.61 0.80 0.58 0.89 1.78 1.90  56  49  75 3.51  7.310 21.020
## 28 14.1693 1.61 0.80 0.60 0.89 1.78 2.05  56  49  51 3.51  7.310  6.697
## 29 29.1140 1.61 0.80 0.76 0.89 1.78 1.70  56  49  92 3.51  7.310 19.050
## 30 48.3337 1.61 1.03 0.78 0.89 1.10 1.50  56  57  91 3.51  6.146 15.370
## 31 17.6175 1.61 0.86 0.64 0.89 1.00 1.60  56  71  41 3.51  9.841  8.570
## 32 34.4563 1.61 0.86 0.78 0.89 1.00 1.50  56  71  91 3.51  9.841 15.370
## 33 12.6617 1.61 0.65 0.58 0.89 1.55 1.90  56  25  75 3.51  7.470 21.020
## 34 30.1899 1.61 0.98 0.64 0.89 1.14 1.60  56  60  41 3.51  7.010  8.570
## 35 27.6160 1.61 0.98 0.58 0.89 1.14 1.90  56  60  75 3.51  7.010 21.020
## 36 31.8921 1.61 0.98 0.64 0.89 1.14 1.50  56  60  73 3.51  7.010 16.650
## 37 12.5410 1.61 0.75 0.64 0.89 2.28 1.60  56  45  41 3.51 12.450  8.570
## 38 13.8465 1.61 0.75 0.64 0.89 1.36 1.60  56  21  41 3.51  2.985  8.570
```

```
## 39  31.3089 1.61 0.75 0.78 0.89 1.36 1.50  56  21  91  3.51  2.985 15.370
## 40  12.3209 1.61 0.75 0.58 0.89 1.36 1.90  56  21  75  3.51  2.985 21.020
## 41  13.7049 1.61 0.75 0.64 0.89 1.36 1.50  56  21  73  3.51  2.985 16.650
## 42  25.7778 1.61 0.75 0.76 0.89 1.36 1.70  56  21  92  3.51  2.985 19.050
## 43  44.4866 1.61 0.96 0.78 0.89 1.17 1.50  56  62  91  3.51  7.353 15.370
## 44  28.6045 1.61 0.96 0.64 0.89 1.17 1.50  56  62  73  3.51  7.353 16.650
## 45  19.1320 1.61 0.89 0.60 0.89 1.80 2.05  56  81  51  3.51 11.850  6.697
## 46  21.5513 1.61 0.89 0.64 0.89 1.80 1.50  56  81  73  3.51 11.850 16.650
## 47  37.9163 1.61 0.88 0.78 0.89 1.25 1.50  56  69  91  3.51  9.320 15.370
## 48  20.8969 1.61 0.88 0.64 0.89 1.25 1.50  56  69  73  3.51  9.320 16.650
## 49  39.4406 1.61 0.90 0.78 0.89 1.22 1.50  56  39  91  3.51  4.472 15.370
## 50  17.8597 1.61 0.90 0.58 0.89 1.22 1.90  56  39  75  3.51  4.472 21.020
## 51  37.1927 1.61 0.90 0.76 0.89 1.22 1.70  56  39  92  3.51  4.472 19.050
## 52  18.3482 1.61 0.87 0.64 0.89 1.21 1.60  56  70  41  3.51  6.570  8.570
## 53  19.9783 1.61 0.87 0.64 0.89 1.21 1.50  56  70  73  3.51  6.570 16.650
## 54  11.9974 1.49 0.75 0.64 1.80 1.36 1.50  82  21  73 11.34  2.985 16.650
## 55   8.8962 1.44 0.54 0.64 0.95 1.61 1.50  38  13  73  2.63  2.700 16.650
## 56   9.3718 1.44 0.61 0.60 0.95 1.88 2.05  38  27  51  2.63  8.900  6.697
## 57   9.0923 1.44 0.62 0.58 0.95 1.66 2.20  38  24  76  2.63  7.190 22.590
## 58   8.9531 1.44 0.62 0.62 0.95 1.66 1.70  38  24  74  2.63  7.190 19.250
## 59   8.9818 1.44 0.62 0.58 0.95 1.81 2.20  38  31  76  2.63  5.904 22.590
## 60   9.1595 1.44 0.62 0.58 0.95 1.81 1.90  38  31  75  2.63  5.904 21.020
## 61  10.2965 1.44 0.80 0.58 0.95 1.78 1.90  38  49  75  2.63  7.310 21.020
## 62  16.4962 1.44 0.80 0.76 0.95 1.78 1.70  38  49  92  2.63  7.310 19.050
## 63  12.7771 1.44 0.75 0.76 0.95 1.36 1.90  38  21  83  2.63  2.985  9.780
## 64   9.7971 1.44 0.75 0.58 0.95 1.36 2.20  38  21  76  2.63  2.985 22.590
## 65   9.5073 1.44 0.67 0.64 0.95 2.28 1.50  38  45  73  2.63 12.450 16.650
## 66   9.3250 1.44 0.62 0.64 0.95 1.66 1.60  38  24  41  2.63  7.190  8.570
## 67  18.8656 1.61 1.00 0.59 0.89 1.00 2.16  56  20  42  3.51  1.550 10.280
## 68  17.3870 1.61 1.00 0.55 0.89 1.00 2.20  56  20  76  3.51  1.550 22.590
## 69  20.2747 1.61 1.00 0.56 0.89 1.00 2.10  56  20  52  3.51  1.550  6.240
## 70  35.1132 1.61 1.00 0.73 0.89 1.00 1.70  56  20  92  3.51  1.550 19.050
## 71  15.9001 1.61 0.95 0.59 0.89 1.69 2.16  56  48  42  3.51  8.650 10.280
## 72  16.0929 1.61 0.95 0.55 0.89 1.69 2.20  56  48  76  3.51  8.650 22.590
## 73  10.6717 1.61 0.75 0.59 0.89 1.88 2.16  56  27  42  3.51  8.900 10.280
## 74  10.7372 1.61 0.75 0.55 0.89 1.88 1.90  56  27  75  3.51  8.900 21.020
## 75  11.3461 1.61 0.75 0.60 0.89 1.88 1.70  56  27  74  3.51  8.900 19.250
## 76  15.1521 1.61 0.80 0.73 0.89 1.66 1.70  56  24  92  3.51  7.190 19.050
## 77  15.3407 1.61 0.78 0.73 0.89 1.83 1.70  56  26  92  3.51  7.874 19.050
## 78  10.6059 1.61 0.72 0.59 0.89 1.31 2.16  56  12  42  3.51  1.738 10.280
## 79  10.5416 1.61 0.72 0.55 0.89 1.31 1.90  56  12  75  3.51  1.738 21.020
## 80  11.7993 1.61 0.72 0.56 0.89 1.31 2.10  56  12  52  3.51  1.738  6.240
## 81  11.0280 1.61 0.72 0.60 0.89 1.31 1.70  56  12  74  3.51  1.738 19.250
## 82  12.4311 1.61 0.83 0.59 0.89 1.55 2.16  56  25  42  3.51  7.470 10.280
## 83  29.6353 1.61 0.83 0.73 0.89 1.55 1.70  56  25  92  3.51  7.470 19.050
## 84   9.9032 1.61 0.69 0.59 0.89 1.91 2.16  56  28  42  3.51  8.908 10.280
## 85  16.7089 1.61 0.69 0.73 0.89 1.91 1.70  56  28  92  3.51  8.908 19.050
## 86  10.4146 1.61 0.69 0.60 0.89 1.91 1.70  56  28  42  3.51  8.908 19.250
## 87  10.9527 1.61 0.74 0.55 0.89 1.65 2.20  56  30  76  3.51  7.140 22.590
## 88  11.2642 1.61 0.74 0.55 0.89 1.65 1.90  56  30  75  3.51  7.140 21.020
## 89  11.5202 1.61 0.74 0.60 0.89 1.65 1.70  56  30  42  3.51  7.140 19.250
## 90   9.6905 1.34 1.00 0.60 1.00 1.00 1.70  20  20  42  1.55  1.550 19.250
## 91  10.0638 1.49 0.78 0.60 1.80 1.83 1.70  82  26  42 11.34  7.874 19.250
## 92   9.6387 1.49 0.72 0.56 1.80 1.31 2.10  82  12  52 11.34  1.738  6.240
```

```
## 93   13.4170 1.44 1.00 0.55 0.95 1.00 2.20  38  20  76  2.63  1.550 22.590
## 94   12.9015 1.44 0.75 0.73 0.95 1.88 1.70  38  27  92  2.63  8.900 19.050
## 95    9.1999 1.44 0.78 0.55 0.95 1.83 2.20  38  26  76  2.63  7.874 22.590
## 96   11.4301 1.44 0.78 0.73 0.95 1.83 1.70  38  26  92  2.63  7.874 19.050
## 97   13.0207 1.44 0.72 0.73 0.95 1.31 1.70  38  12  92  2.63  1.738 19.050
## 98   14.4828 1.44 0.83 0.73 0.95 1.55 1.70  38  25  92  2.63  7.470 19.050
## 99   11.6083 1.61 0.76 0.58 0.89 0.98 1.90  56   3  75  3.51  0.535 21.020
## 100  14.9819 1.61 1.02 0.58 0.89 0.93 1.90  56  11  75  3.51  0.968 21.020
## 101   9.2440 1.44 0.76 0.53 0.95 0.98 2.20  38   3  76  2.63  0.535 22.590
## 102  11.8922 1.44 1.02 0.58 0.95 0.93 1.90  38  11  75  2.63  0.968 21.020
## 103  10.8043 1.61 0.75 0.58 0.89 1.88 1.90  56  27  75  3.51  8.900 21.020
## 104  30.7430 1.61 0.91 0.64 0.89 1.22 1.50  56  66  73  3.51  8.551 16.650
## 105  16.9389 1.61 0.89 0.58 0.89 1.24 1.90  56  68  75  3.51  9.066 21.020
## 106  35.7826 1.61 0.89 0.89 0.89 1.24 1.38  56  68  92  3.51  9.066 19.050
## 107  26.6801 1.61 0.95 0.95 0.89 1.12 1.50  56  63  73  3.51  5.244 16.650
## 108  26.2214 1.61 0.94 0.64 0.89 1.20 1.60  56  64  41  3.51  7.901  8.570
## 109  24.0826 1.61 0.94 0.60 0.89 1.20 2.05  56  64  51  3.51  7.901  6.697
## 110  24.8866 1.61 0.90 0.95 0.89 1.23 1.50  56  67  73  3.51  8.795 16.650
## 111  33.7496 1.61 0.80 0.78 0.89 1.78 1.50  56  49  91  3.51  7.310 15.370
## 112  14.6398 1.61 0.80 0.95 0.89 1.78 1.50  56  49  73  3.51  7.310 16.650
## 113  33.1125 1.61 1.03 0.53 0.89 1.10 1.90  56  57  75  3.51  6.146 21.020
## 114  18.0999 1.61 0.86 0.95 0.89 1.00 1.50  56  71  73  3.51  9.841 16.650
## 115  46.3647 1.61 0.98 0.78 0.89 1.14 1.50  56  60  91  3.51  7.010 15.370
## 116  47.3367 1.61 0.99 0.78 0.89 1.13 1.50  56  59  91  3.51  6.640 15.370
## 117  12.2072 1.61 0.75 0.53 0.89 1.36 2.20  56  21  76  3.51  2.985 22.590
## 118  13.1500 1.61 0.75 0.60 0.89 1.36 2.05  56  21  51  3.51  2.985  6.697
## 119  28.1076 1.61 0.96 0.64 0.89 1.17 1.60  56  62  41  3.51  7.353  8.570
## 120  41.8685 1.61 0.92 0.78 0.89 1.10 1.50  56  65  91  3.51  8.219 15.370
## 121  21.2151 1.61 0.88 0.64 0.89 1.25 1.60  56  69  41  3.51  9.320  8.570
## 122  24.4796 1.61 0.90 0.64 0.89 1.22 1.60  56  39  41  3.51  4.472  8.570
## 123  22.9453 1.61 0.90 0.64 0.89 1.22 1.50  56  39  73  3.51  4.472 16.650
## 124  36.4770 1.61 0.87 0.78 0.89 1.21 1.50  56  70  91  3.51  6.570 15.370
## 125   8.8639 1.44 0.54 0.64 0.95 1.61 1.60  38  13  41  2.63  2.700  8.570
## 126   9.1240 1.44 0.62 0.59 0.95 1.66 2.16  38  24  42  2.63  7.190 10.280
## 127  10.2336 1.44 0.78 0.76 0.95 1.83 1.90  38  26  83  2.63  7.874  9.780
## 128  10.1770 1.44 0.80 0.58 0.95 1.78 2.20  38  49  76  2.63  7.310 22.590
## 129   9.8421 1.44 0.75 0.53 0.95 1.36 1.90  38  21  75  2.63  2.985 21.020
## 130   9.4436 1.44 0.75 0.64 0.95 2.28 1.60  38  45  41  2.63 12.450  8.570
## 131  49.3373 1.61 1.61 0.76 0.89 0.89 1.38  56  56  92  3.51  3.510 19.050
## 132  17.1582 1.61 1.34 0.58 0.89 1.00 1.90  56  20  75  3.51  1.550 21.020
## 133  19.6902 1.61 1.34 0.62 0.89 1.00 2.36  56  20  74  3.51  1.550 19.250
## 134  15.5286 1.61 0.95 0.58 0.89 1.69 1.90  56  48  75  3.51  8.650 21.020
## 135  18.6007 1.61 0.75 0.76 0.89 1.88 1.38  56  27  92  3.51  8.900 19.050
## 136  10.1147 1.61 0.78 0.58 0.89 1.83 1.90  56  26  75  3.51  7.874 21.020
## 137  10.4789 1.61 0.72 0.58 0.89 1.31 2.20  56  12  76  3.51  1.738 22.590
## 138  19.4042 1.61 0.72 0.76 0.89 1.31 1.38  56  12  92  3.51  1.738 19.050
## 139  13.2774 1.61 0.83 0.62 0.89 1.55 2.36  56  25  25  3.51  7.470 19.250
## 140   9.9555 1.61 0.69 0.58 0.89 1.91 1.90  56  28  75  3.51  8.908 21.020
## 141  11.1072 1.61 0.74 0.59 0.89 1.65 2.16  56  30  42  3.51  7.140 10.280
## 142  20.5795 1.61 0.74 0.76 0.89 1.65 1.38  56  30  92  3.51  7.140 19.050
## 143   8.7513 1.34 0.72 0.62 1.00 1.31 2.36  20  12  74  1.55  1.738 19.250
## 144   9.7393 1.49 0.72 0.62 1.80 1.31 2.36  82  12  74  1.34  1.738 19.250
## 145  10.8771 1.44 0.62 0.76 0.95 1.66 1.38  38  24  92  2.63  7.190 19.050
## 146   9.5480 1.44 0.72 0.56 0.95 1.31 2.10  38  12  52  2.63  1.738  6.240
```

```
## 147 12.1051 1.44 0.69 0.76 0.95 1.91 1.38  38  28  92  2.63  8.908 19.050
```

1.

a. Fit a linear model to the given data

Full Model All Variables

```r
model1 = summary(lm(y ~., data = data_1))
model1
```

```
##
## Call:
## lm(formula = y ~ ., data = data_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.4850  -2.7802  -0.2502   2.9391  12.6975
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -139.58970   72.51657  -1.925  0.05636 .
## x01           91.57240   48.09504   1.904  0.05906 .
## x02           17.34218    3.77987   4.588 1.02e-05 ***
## x03            9.52072    6.90665   1.378  0.17035
## x04           24.49670   17.02610   1.439  0.15255
## x05           -5.41615    2.42357  -2.235  0.02709 *
## x06           -4.27759    2.04530  -2.091  0.03838 *
## x07           -0.51718    0.36820  -1.405  0.16244
## x08            0.15565    0.04409   3.530  0.00057 ***
## x09            0.27336    0.03823   7.151 5.04e-11 ***
## x10           -0.02014    0.57582  -0.035  0.97215
## x11           -0.08652    0.34406  -0.251  0.80184
## x12           -0.61904    0.11823  -5.236 6.20e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.824 on 134 degrees of freedom
## Multiple R-squared:  0.8163, Adjusted R-squared:  0.7999
## F-statistic: 49.63 on 12 and 134 DF,  p-value: < 2.2e-16
```

```r
model_1 = lm(y ~., data = data_1)
```

The regression coefficients

```r
model1$coefficients
```

```
##                 Estimate  Std. Error     t value     Pr(>|t|)
## (Intercept) -139.58969648 72.51657066 -1.92493516 5.635654e-02
## x01           91.57240283 48.09503584  1.90398866 5.905559e-02
## x02           17.34218119  3.77986594  4.58804134 1.016677e-05
## x03            9.52071607  6.90664708  1.37848597 1.703504e-01
```

```
## x04           24.49670330 17.02610039  1.43877357 1.525464e-01
## x05           -5.41614994  2.42357139 -2.23478044 2.708761e-02
## x06           -4.27758804  2.04529869 -2.09142462 3.837785e-02
## x07           -0.51718427  0.36819569 -1.40464510 1.624406e-01
## x08            0.15564832  0.04408910  3.53031293 5.695137e-04
## x09            0.27335642  0.03822564  7.15112676 5.042774e-11
## x10           -0.02014261  0.57581516 -0.03498104 9.721469e-01
## x11           -0.08651679  0.34405695 -0.25146066 8.018432e-01
## x12           -0.61904349  0.11823439 -5.23573122 6.198813e-07
```

b. Check the fit of the model, and identify which coefficients are significant.

Given alpha (level of significance) = 0.05, x02,x05,x06,x08,x09,x12 are the significant coefficients since their p-value is less than alpha.

The fit of the model is measured by the Rsquared value, the coefficient of determination. This is the proportion of the total sum of squares due to regression. Since later on we will be using reduced models with fewer variables, we are going to use adjusted R squared which takes into account the number of independent variables in the model.

```
model1$adj.r.squared
```

```
## [1] 0.7998815
```

This means that our model can explain 80% of the total variation of y around its mean. Since our adjusted R^2 is between 0.75 to 1.00, our model is a good fit for the data.

c. Perform model diagnostics by identifying which variables should be included, analyzing the resulting residuals, and testing for multicollinearity.

Variable Selection

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.2.3
```

```
nvar = ncol(data_1) - 1 #id col, and y-int are not included
variableselection = regsubsets(y~., data = data_1, nvmax= nvar)
variableselection.res = summary(variableselection)
variableselection.res
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ ., data = data_1, nvmax = nvar)
## 12 Variables  (and intercept)
##     Forced in Forced out
## x01     FALSE      FALSE
## x02     FALSE      FALSE
## x03     FALSE      FALSE
## x04     FALSE      FALSE
## x05     FALSE      FALSE
## x06     FALSE      FALSE
```

```
## x07     FALSE      FALSE
## x08     FALSE      FALSE
## x09     FALSE      FALSE
## x10     FALSE      FALSE
## x11     FALSE      FALSE
## x12     FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
##           x01 x02 x03 x04 x05 x06 x07 x08 x09 x10 x11 x12
## 1  ( 1 )  " " " " " " " " " " " " " " "*" " " " " " " " "
## 2  ( 1 )  " " "*" "*" " " " " " " " " " " " " " " " " " "
## 3  ( 1 )  " " "*" "*" " " " " " " " " "*" " " " " " " " "
## 4  ( 1 )  " " "*" " " " " " " " " " " "*" "*" " " " " "*"
## 5  ( 1 )  "*" " " " " " " "*" " " " " "*" "*" " " " " "*"
## 6  ( 1 )  "*" "*" " " " " "*" " " " " "*" "*" " " " " "*"
## 7  ( 1 )  "*" "*" " " " " "*" "*" " " "*" "*" " " " " "*"
## 8  ( 1 )  "*" "*" "*" " " "*" "*" " " "*" "*" " " " " "*"
## 9  ( 1 )  "*" "*" " " "*" "*" "*" "*" "*" "*" " " " " "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" " " " " "*"
## 11  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" " " "*" "*"
## 12  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

```r
names(variableselection.res)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

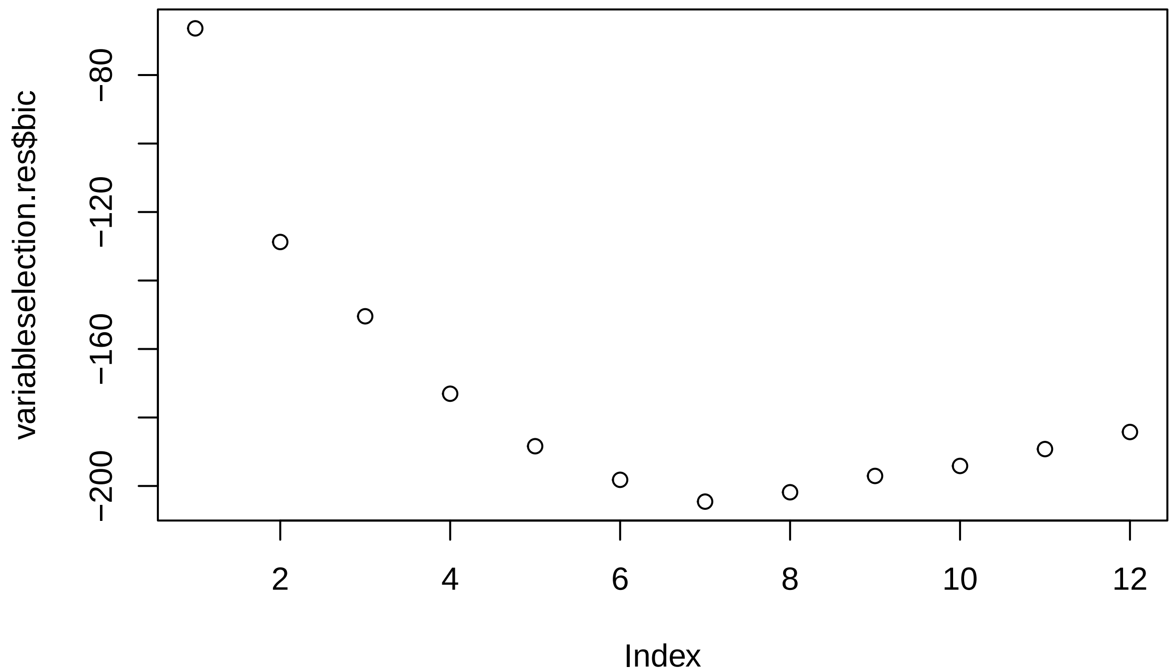
We will only be considering 1 metric, BIC or the Bayesian Information Criterion. The model with the lowest BIC is usually preferred, as it strikes a balance between goodness of fit and model complexity.

```r
variableselection.metric = cbind(1:nvar, variableselection.res$bic)
colnames(variableselection.metric) = c("No. of Variables", "BIC")
variableselection.metric
```

```
##       No. of Variables        BIC
##  [1,]                1  -66.37373
##  [2,]                2 -128.75107
##  [3,]                3 -150.44266
##  [4,]                4 -173.05911
##  [5,]                5 -188.38451
##  [6,]                6 -198.19422
##  [7,]                7 -204.55991
##  [8,]                8 -201.83125
##  [9,]                9 -197.06974
## [10,]               10 -194.14157
## [11,]               11 -189.22147
## [12,]               12 -184.23238
```

We plot the BIC against the number of variables in the model

```r
plot(variableselection.res$bic) #BIC
```



We find the variables which will result in the lowest BIC value.

```
variableselection.res$which[which.min(variableselection.res$bic),] #BIC
```

```
## (Intercept)         x01         x02         x03         x04         x05
##        TRUE        TRUE        TRUE       FALSE       FALSE        TRUE
##         x06         x07         x08         x09         x10         x11
##        TRUE       FALSE        TRUE        TRUE       FALSE       FALSE
##         x12
##        TRUE
```

The variables which will result in the lowest BIC are x01,x02,x05,x06,x08,x09,x12.

We will select these 7 variables: x01,x02,x05,x06,x08,x09,x12 for our reduced model.

```
reduced_model_1 = lm(y ~ x01 + x02 + x05 + x06 + x08 + x09 + x12, data = data_1)
reduced_model1 = summary(reduced_model_1)
reduced_model1
```

```
##
## Call:
## lm(formula = y ~ x01 + x02 + x05 + x06 + x08 + x09 + x12, data = data_1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.5840  -2.7843  -0.5895   2.7654  14.3002
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.47514    9.96275  -2.858  0.00492 **
## x01          23.60338    5.72485   4.123 6.40e-05 ***
## x02          17.47304    3.74160   4.670 7.03e-06 ***
## x05          -6.35633    1.45473  -4.369 2.42e-05 ***
## x06          -5.72048    1.71209  -3.341  0.00107 **
## x08           0.15021    0.02577   5.828 3.73e-08 ***
## x09           0.29431    0.03124   9.420  < 2e-16 ***
## x12          -0.66556    0.10573  -6.295 3.76e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.811 on 139 degrees of freedom
## Multiple R-squared:  0.8105, Adjusted R-squared:  0.8009
## F-statistic: 84.91 on 7 and 139 DF,  p-value: < 2.2e-16
```

Comparison of Fit between the full and reduced models

```
model1$adj.r.squared
```

```
## [1] 0.7998815
```

```
reduced_model1$adj.r.squared
```

```
## [1] 0.8009142
```

```
anova(model_1, reduced_model_1)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ x01 + x02 + x03 + x04 + x05 + x06 + x07 + x08 + x09 + x10 +
##     x11 + x12
## Model 2: y ~ x01 + x02 + x05 + x06 + x08 + x09 + x12
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    134 3118.3
## 2    139 3217.9 -5    -99.66 0.8565 0.5123
```

The reduced model is slightly better fitting compared to the full model; however since the p value in the anova test is 0.5123, which is greater then 0.05, the difference is not significant. We can conclude that the predictive power of both models is roughly the same.

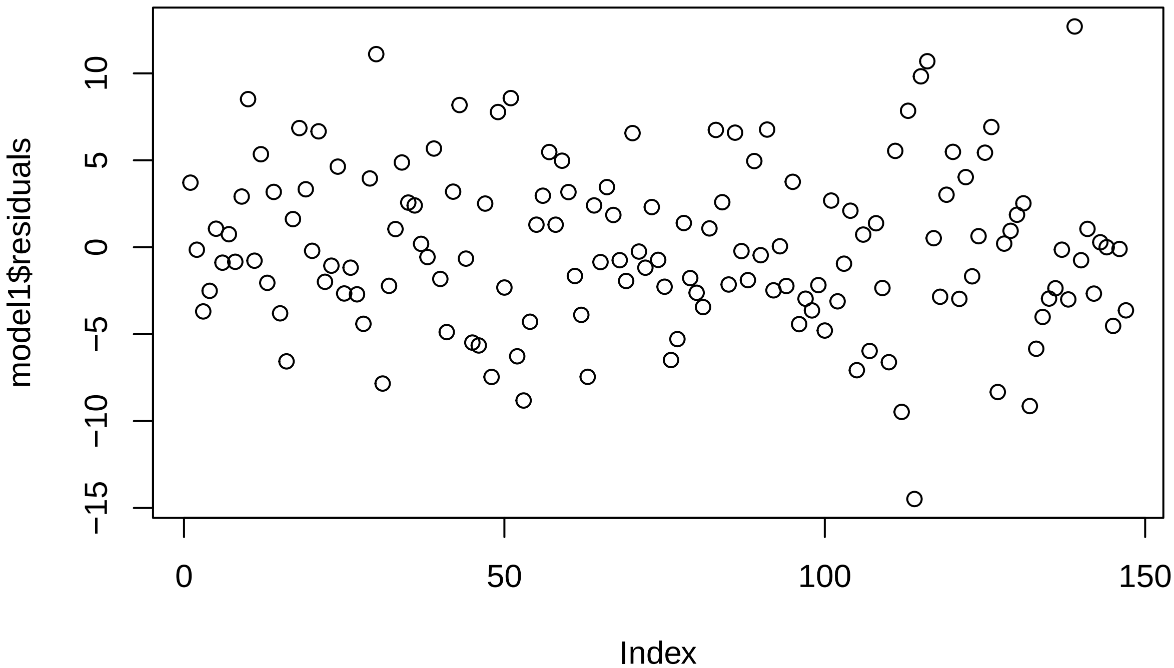
Residual Analysis

We have to check if the residuals are uncorrelated and if they are normalized as this is one of the assumptions of a multiple linear regression model.

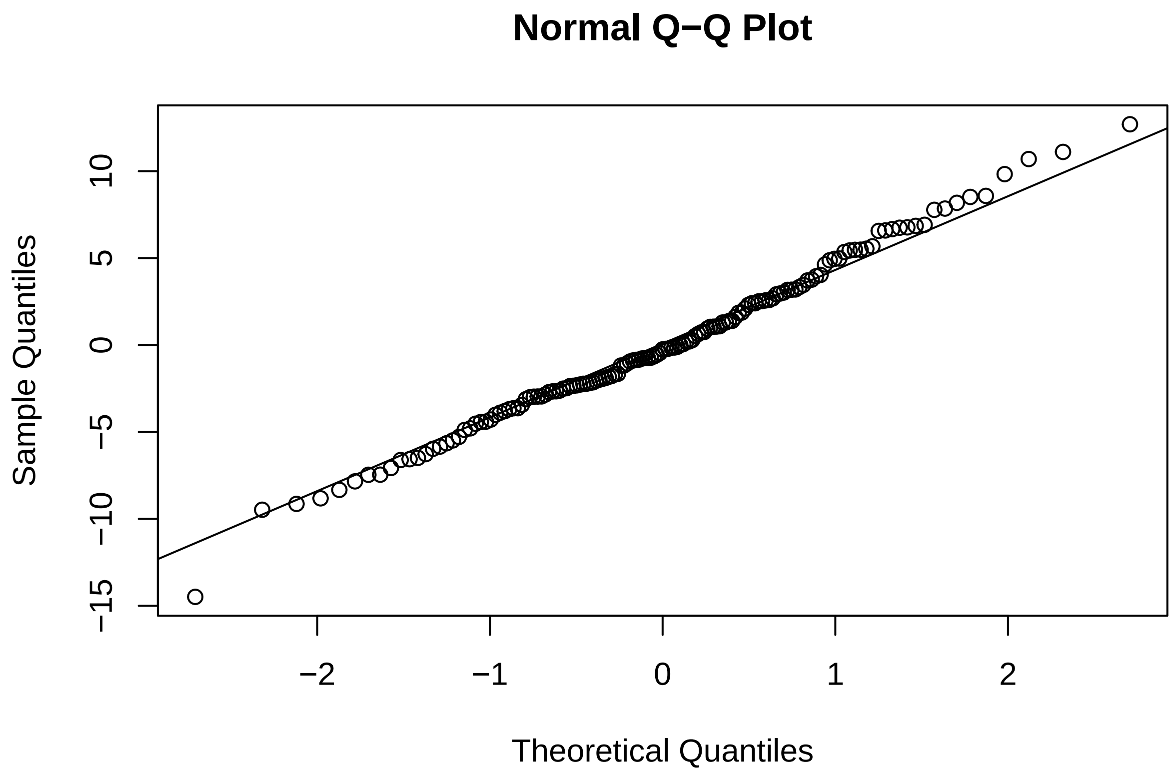
Normal QQ Plot

First, we check the residuals of model1 or the complete model.

```
library(nortest)
plot(model1$residuals)
```

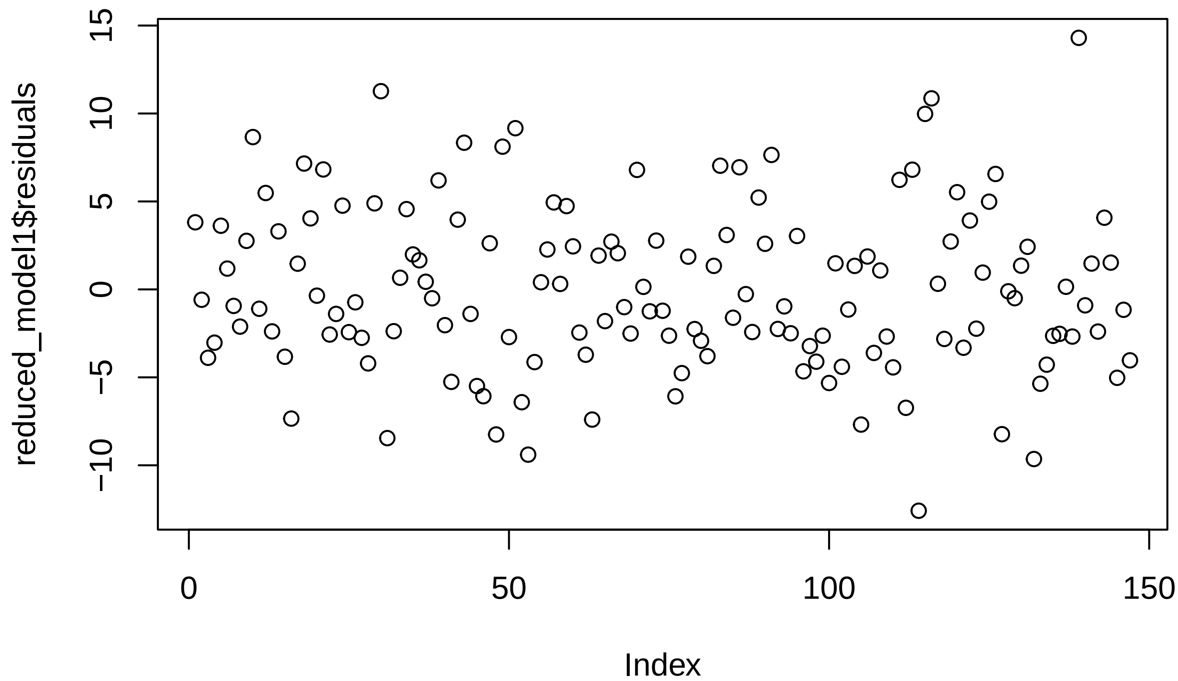



```
qqnorm(model1$residuals)
qqline(model1$residuals)
```

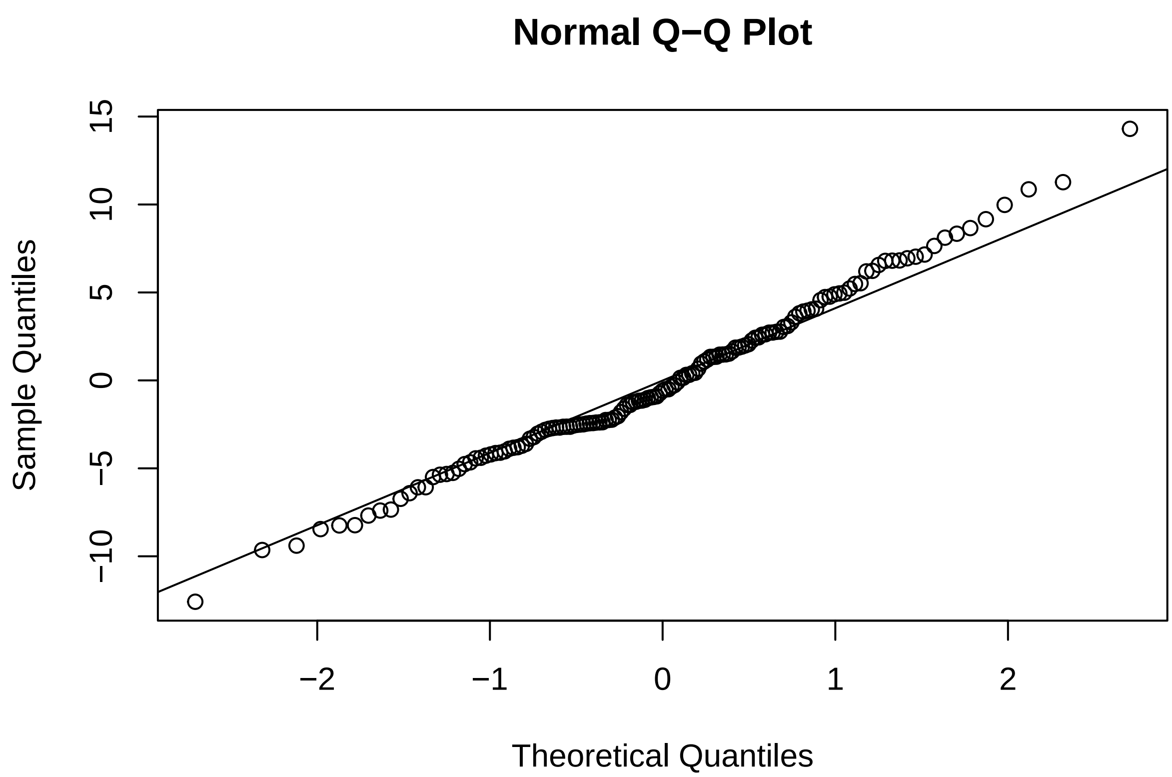


Residuals of the reduced model

```
library(nortest)
plot(reduced_model1$residuals)
```




```
qqnorm(reduced_model1$residuals)
qqline(reduced_model1$residuals)
```



For both the complete and reduced models, the residuals don't form a distinct line and are randomly distributed, meaning that the residuals are not correlated.

The quantiles of the residuals almost perfectly fall on the normal QQ line. This means that the residuals for both the full and reduced models follow a normal distribution.

```
cat("Full Model")
```

```
## Full Model
```

```
ad.test(model1$residuals) #Anderson-Darling
```

```
##
##  Anderson-Darling normality test
##
## data:  model1$residuals
## A = 0.36558, p-value = 0.4312
```

```
shapiro.test(model1$residuals) #Shapiro-Wilk
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model1$residuals
## W = 0.99368, p-value = 0.7691
```

```r
cat("Reduced Model")
```

```
## Reduced Model
```

```r
ad.test(reduced_model1$residuals) #Anderson-Darling
```

```
##
##  Anderson-Darling normality test
##
## data:  reduced_model1$residuals
## A = 0.60152, p-value = 0.1163
```

```r
shapiro.test(reduced_model1$residuals) #Shapiro-Wilk
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reduced_model1$residuals
## W = 0.99078, p-value = 0.4518
```

This assumption of normality is verified by the AD and SW tests. The null hypothesis of both the AD and SW tests is that the residuals follow a normal distribution. In both tests for the full and reduced models, the p value > alpha = 0.05 so we fail to reject the null hypothesis that the residuals follow a normal distribution.

Testing Multicollinearity

We retain the variable if its variance inflation factor is less than 5. The VIF (variance inflation factor) measures how reliably a variable is predicted by the other variables.

First, we check the full model.

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.2.3
```

```r
vif(model_1)
```

```
##       x01       x02       x03       x04       x05       x06       x07       x08
## 92.221080  1.987640  2.833045 40.266695  4.203389  2.317247 90.295026  5.697418
##       x09       x10       x11       x12
##  3.131424  3.230250  7.286726  2.438326
```

The variables: x01, x04, x07, x08, and x11 all have significant multicollinearity, which means that these variables have a significant correlation with the other variables.

In particular, x01, x04, x07 are highly correlated with a VIF greater than 10. This means that we can remove two out of the three variables since these variables essentially have the same relationship with the dependent variable. In our case, we only keep x01 in the reduced model.

VIF of the reduced model

```r
vif(reduced_model_1)
```

```
##      x01      x02      x05      x06      x08      x09      x12
## 1.313426 1.957701 1.522306 1.632148 1.956929 2.102893 1.959864
```

In the reduced model, all the variables' VIF are below 5 so none of the variables are significantly correlated or can be reliably predicted by the other variables. There is little multicollinearity between the variables.

2. In some cases, data transformations are necessary to improve the fit of the linear model. Consider the transformation z = ln y for the data given in PS3.csv.

Transforming the data

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::recode() masks car::recode()
## x purrr::some()   masks car::some()
```

```r
data_2 = data_1
data_2 = transform(data_2, y = log(y))
data_2 = rename(data_2, z = y)
data_2
```

```
##            z  x01  x02  x03  x04  x05  x06 x07 x08 x09  x10    x11    x12
## 1   3.231630 1.61 1.15 0.53 0.89 1.93 2.66  56  47  53 3.51 10.490  4.940
## 2   2.414591 1.61 0.76 0.53 0.89 0.98 2.20  56   3  76 3.51  0.535 22.590
## 3   2.790828 1.61 1.02 0.53 0.89 0.93 2.66  56  11  53 3.51  0.968  4.940
## 4   2.695154 1.61 1.02 0.53 0.89 0.93 2.20  56  11  76 3.51  0.968 22.590
## 5   2.199577 1.34 0.76 0.53 1.00 0.98 2.20  20   3  76 1.55  0.535 22.590
## 6   2.202919 1.34 0.76 0.53 1.00 0.98 1.90  20   3  75 1.55  0.535 21.020
## 7   2.228842 1.44 0.76 0.53 0.95 0.98 1.90  38   3  75 2.63  0.535 21.020
## 8   2.460332 1.44 1.02 0.53 0.95 0.93 2.20  38  11  76 2.63  0.968 22.590
## 9   3.481080 1.61 1.03 0.64 0.89 1.90 1.50  56  83  73 3.51  9.780 16.650
## 10  3.815472 1.61 1.01 0.78 0.89 1.12 1.50  56  58  91 3.51  6.689 15.370
## 11  3.165197 1.61 0.91 0.64 0.89 1.22 1.60  56  66  41 3.51  8.551  8.570
## 12  3.714494 1.61 0.91 0.78 0.89 1.22 1.50  56  66  91 3.51  8.551 15.370
## 13  3.101515 1.61 0.89 0.64 0.89 1.24 1.60  56  68  41 3.51  9.066  8.570
## 14  3.655325 1.61 0.89 0.78 0.89 1.24 1.50  56  68  91 3.51  9.066 15.370
## 15  2.754431 1.61 0.89 0.57 0.89 1.24 2.20  56  68  44 3.51  9.066 12.370
## 16  3.085879 1.61 0.89 0.64 0.89 1.24 1.50  56  68  73 3.51  9.066 16.650
## 17  3.301053 1.61 0.95 0.64 0.89 1.12 1.60  56  63  41 3.51  5.244  8.570
## 18  3.774972 1.61 0.95 0.78 0.89 1.12 1.50  56  63  91 3.51  5.244 15.370
## 19  2.337653 1.61 0.65 0.61 0.89 1.83 2.16  56  26  42 3.51  7.874 10.280
```

```
## 20  2.303984 1.61 0.65 0.58 0.89 1.83 1.90  56  26  75  3.51  7.874 21.020
## 21  3.754850 1.61 0.94 0.78 0.89 1.20 1.50  56  64  91  3.51  7.901 15.370
## 22  3.117286 1.61 0.94 0.58 0.89 1.20 1.90  56  64  75  3.51  7.901 21.020
## 23  3.149041 1.61 0.90 0.64 0.89 1.23 1.60  56  67  41  3.51  8.795  8.570
## 24  3.694623 1.61 0.90 0.78 0.89 1.23 1.50  56  67  91  3.51  8.795 15.370
## 25  2.662258 1.61 0.80 0.64 0.89 1.78 1.60  56  49  41  3.51  7.310  8.570
## 26  2.607331 1.61 0.80 0.58 0.89 1.78 2.20  56  49  76  3.51  7.310 22.590
## 27  2.639957 1.61 0.80 0.58 0.89 1.78 1.90  56  49  75  3.51  7.310 21.020
## 28  2.651078 1.61 0.80 0.60 0.89 1.78 2.05  56  49  51  3.51  7.310  6.697
## 29  3.371219 1.61 0.80 0.76 0.89 1.78 1.70  56  49  92  3.51  7.310 19.050
## 30  3.878129 1.61 1.03 0.78 0.89 1.10 1.50  56  57  91  3.51  6.146 15.370
## 31  2.868893 1.61 0.86 0.64 0.89 1.00 1.60  56  71  41  3.51  9.841  8.570
## 32  3.539692 1.61 0.86 0.78 0.89 1.00 1.50  56  71  91  3.51  9.841 15.370
## 33  2.538582 1.61 0.65 0.58 0.89 1.55 1.90  56  25  75  3.51  7.470 21.020
## 34  3.407507 1.61 0.98 0.64 0.89 1.14 1.60  56  60  41  3.51  7.010  8.570
## 35  3.318395 1.61 0.98 0.58 0.89 1.14 1.90  56  60  75  3.51  7.010 21.020
## 36  3.462358 1.61 0.98 0.64 0.89 1.14 1.50  56  60  73  3.51  7.010 16.650
## 37  2.529003 1.61 0.75 0.64 0.89 2.28 1.60  56  45  41  3.51 12.450  8.570
## 38  2.628032 1.61 0.75 0.64 0.89 1.36 1.60  56  21  41  3.51  2.985  8.570
## 39  3.443902 1.61 0.75 0.78 0.89 1.36 1.50  56  21  91  3.51  2.985 15.370
## 40  2.511297 1.61 0.75 0.58 0.89 1.36 1.90  56  21  75  3.51  2.985 21.020
## 41  2.617753 1.61 0.75 0.64 0.89 1.36 1.50  56  21  73  3.51  2.985 16.650
## 42  3.249514 1.61 0.75 0.76 0.89 1.36 1.70  56  21  92  3.51  2.985 19.050
## 43  3.795188 1.61 0.96 0.78 0.89 1.17 1.50  56  62  91  3.51  7.353 15.370
## 44  3.353564 1.61 0.96 0.64 0.89 1.17 1.50  56  62  73  3.51  7.353 16.650
## 45  2.951362 1.61 0.89 0.60 0.89 1.80 2.05  56  81  51  3.51 11.850  6.697
## 46  3.070436 1.61 0.89 0.64 0.89 1.80 1.50  56  81  73  3.51 11.850 16.650
## 47  3.635381 1.61 0.88 0.78 0.89 1.25 1.50  56  69  91  3.51  9.320 15.370
## 48  3.039601 1.61 0.88 0.64 0.89 1.25 1.50  56  69  73  3.51  9.320 16.650
## 49  3.674796 1.61 0.90 0.78 0.89 1.22 1.50  56  39  91  3.51  4.472 15.370
## 50  2.882547 1.61 0.90 0.58 0.89 1.22 1.90  56  39  75  3.51  4.472 21.020
## 51  3.616113 1.61 0.90 0.76 0.89 1.22 1.70  56  39  92  3.51  4.472 19.050
## 52  2.909531 1.61 0.87 0.64 0.89 1.21 1.60  56  70  41  3.51  6.570  8.570
## 53  2.994647 1.61 0.87 0.64 0.89 1.21 1.50  56  70  73  3.51  6.570 16.650
## 54  2.484690 1.49 0.75 0.64 1.80 1.36 1.50  82  21  73 11.34  2.985 16.650
## 55  2.185624 1.44 0.54 0.64 0.95 1.61 1.50  38  13  73  2.63  2.700 16.650
## 56  2.237705 1.44 0.61 0.60 0.95 1.88 2.05  38  27  51  2.63  8.900  6.697
## 57  2.207428 1.44 0.62 0.58 0.95 1.66 2.20  38  24  76  2.63  7.190 22.590
## 58  2.192000 1.44 0.62 0.62 0.95 1.66 1.70  38  24  74  2.63  7.190 19.250
## 59  2.195200 1.44 0.62 0.58 0.95 1.81 2.20  38  31  76  2.63  5.904 22.590
## 60  2.214792 1.44 0.62 0.58 0.95 1.81 1.90  38  31  75  2.63  5.904 21.020
## 61  2.331804 1.44 0.80 0.58 0.95 1.78 1.90  38  49  75  2.63  7.310 21.020
## 62  2.803130 1.44 0.80 0.76 0.95 1.78 1.70  38  49  92  2.63  7.310 19.050
## 63  2.547655 1.44 0.75 0.76 0.95 1.36 1.90  38  21  83  2.63  2.985  9.780
## 64  2.282086 1.44 0.75 0.58 0.95 1.36 2.20  38  21  76  2.63  2.985 22.590
## 65  2.252060 1.44 0.67 0.64 0.95 2.28 1.50  38  45  73  2.63 12.450 16.650
## 66  2.232699 1.44 0.62 0.64 0.95 1.66 1.60  38  24  41  2.63  7.190  8.570
## 67  2.937340 1.61 1.00 0.59 0.89 1.00 2.16  56  20  42  3.51  1.550 10.280
## 68  2.855723 1.61 1.00 0.55 0.89 1.00 2.20  56  20  76  3.51  1.550 22.590
## 69  3.009374 1.61 1.00 0.56 0.89 1.00 2.10  56  20  52  3.51  1.550  6.240
## 70  3.558577 1.61 1.00 0.73 0.89 1.00 1.70  56  20  92  3.51  1.550 19.050
## 71  2.766325 1.61 0.95 0.59 0.89 1.69 2.16  56  48  42  3.51  8.650 10.280
## 72  2.778378 1.61 0.95 0.55 0.89 1.69 2.20  56  48  76  3.51  8.650 22.590
## 73  2.367595 1.61 0.75 0.59 0.89 1.88 2.16  56  27  42  3.51  8.900 10.280
```

```
## 74  2.373714 1.61 0.75 0.55 0.89 1.88 1.90  56  27  75  3.51  8.900 21.020
## 75  2.428874 1.61 0.75 0.60 0.89 1.88 1.70  56  27  74  3.51  8.900 19.250
## 76  2.718139 1.61 0.80 0.73 0.89 1.66 1.70  56  24  92  3.51  7.190 19.050
## 77  2.730509 1.61 0.78 0.73 0.89 1.83 1.70  56  26  92  3.51  7.874 19.050
## 78  2.361410 1.61 0.72 0.59 0.89 1.31 2.16  56  12  42  3.51  1.738 10.280
## 79  2.355329 1.61 0.72 0.55 0.89 1.31 1.90  56  12  75  3.51  1.738 21.020
## 80  2.468040 1.61 0.72 0.56 0.89 1.31 2.10  56  12  52  3.51  1.738  6.240
## 81  2.400437 1.61 0.72 0.60 0.89 1.31 1.70  56  12  74  3.51  1.738 19.250
## 82  2.520201 1.61 0.83 0.59 0.89 1.55 2.16  56  25  42  3.51  7.470 10.280
## 83  3.388966 1.61 0.83 0.73 0.89 1.55 1.70  56  25  92  3.51  7.470 19.050
## 84  2.292858 1.61 0.69 0.59 0.89 1.91 2.16  56  28  42  3.51  8.908 10.280
## 85  2.815942 1.61 0.69 0.73 0.89 1.91 1.70  56  28  92  3.51  8.908 19.050
## 86  2.343209 1.61 0.69 0.60 0.89 1.91 1.70  56  28  42  3.51  8.908 19.250
## 87  2.393586 1.61 0.74 0.55 0.89 1.65 2.20  56  30  76  3.51  7.140 22.590
## 88  2.421630 1.61 0.74 0.55 0.89 1.65 1.90  56  30  75  3.51  7.140 21.020
## 89  2.444102 1.61 0.74 0.60 0.89 1.65 1.70  56  30  42  3.51  7.140 19.250
## 90  2.271146 1.34 1.00 0.60 1.00 1.00 1.70  20  20  42  1.55  1.550 19.250
## 91  2.308945 1.49 0.78 0.60 1.80 1.83 1.70  82  26  42 11.34  7.874 19.250
## 92  2.265786 1.49 0.72 0.56 1.80 1.31 2.10  82  12  52 11.34  1.738  6.240
## 93  2.596523 1.44 1.00 0.55 0.95 1.00 2.20  38  20  76  2.63  1.550 22.590
## 94  2.557344 1.44 0.75 0.73 0.95 1.88 1.70  38  27  92  2.63  8.900 19.050
## 95  2.219193 1.44 0.78 0.55 0.95 1.83 2.20  38  26  76  2.63  7.874 22.590
## 96  2.436250 1.44 0.78 0.73 0.95 1.83 1.70  38  26  92  2.63  7.874 19.050
## 97  2.566540 1.44 0.72 0.73 0.95 1.31 1.70  38  12  92  2.63  1.738 19.050
## 98  2.672962 1.44 0.83 0.73 0.95 1.55 1.70  38  25  92  2.63  7.470 19.050
## 99  2.451720 1.61 0.76 0.58 0.89 0.98 1.90  56   3  75  3.51  0.535 21.020
## 100 2.706843 1.61 1.02 0.58 0.89 0.93 1.90  56  11  75  3.51  0.968 21.020
## 101 2.223975 1.44 0.76 0.53 0.95 0.98 2.20  38   3  76  2.63  0.535 22.590
## 102 2.475883 1.44 1.02 0.58 0.95 0.93 1.90  38  11  75  2.63  0.968 21.020
## 103 2.379944 1.61 0.75 0.58 0.89 1.88 1.90  56  27  75  3.51  8.900 21.020
## 104 3.425662 1.61 0.91 0.64 0.89 1.22 1.50  56  66  73  3.51  8.551 16.650
## 105 2.829613 1.61 0.89 0.58 0.89 1.24 1.90  56  68  75  3.51  9.066 21.020
## 106 3.577462 1.61 0.89 0.89 0.89 1.24 1.38  56  68  92  3.51  9.066 19.050
## 107 3.283918 1.61 0.95 0.95 0.89 1.12 1.50  56  63  73  3.51  5.244 16.650
## 108 3.266576 1.61 0.94 0.64 0.89 1.20 1.60  56  64  41  3.51  7.901  8.570
## 109 3.181490 1.61 0.94 0.60 0.89 1.20 2.05  56  64  51  3.51  7.901  6.697
## 110 3.214330 1.61 0.90 0.95 0.89 1.23 1.50  56  67  73  3.51  8.795 16.650
## 111 3.518969 1.61 0.80 0.78 0.89 1.78 1.50  56  49  91  3.51  7.310 15.370
## 112 2.683744 1.61 0.80 0.95 0.89 1.78 1.50  56  49  73  3.51  7.310 16.650
## 113 3.499911 1.61 1.03 0.53 0.89 1.10 1.90  56  57  75  3.51  6.146 21.020
## 114 2.895906 1.61 0.86 0.95 0.89 1.00 1.50  56  71  73  3.51  9.841 16.650
## 115 3.836538 1.61 0.98 0.78 0.89 1.14 1.50  56  60  91  3.51  7.010 15.370
## 116 3.857286 1.61 0.99 0.78 0.89 1.13 1.50  56  59  91  3.51  6.640 15.370
## 117 2.502026 1.61 0.75 0.53 0.89 1.36 2.20  56  21  76  3.51  2.985 22.590
## 118 2.576422 1.61 0.75 0.60 0.89 1.36 2.05  56  21  51  3.51  2.985  6.697
## 119 3.336040 1.61 0.96 0.64 0.89 1.17 1.60  56  62  41  3.51  7.353  8.570
## 120 3.734534 1.61 0.92 0.78 0.89 1.10 1.50  56  65  91  3.51  8.219 15.370
## 121 3.054713 1.61 0.88 0.64 0.89 1.25 1.60  56  69  41  3.51  9.320  8.570
## 122 3.197840 1.61 0.90 0.64 0.89 1.22 1.60  56  39  41  3.51  4.472  8.570
## 123 3.133113 1.61 0.90 0.64 0.89 1.22 1.50  56  39  73  3.51  4.472 16.650
## 124 3.596682 1.61 0.87 0.78 0.89 1.21 1.50  56  70  91  3.51  6.570 15.370
## 125 2.181987 1.44 0.54 0.64 0.95 1.61 1.60  38  13  41  2.63  2.700  8.570
## 126 2.210908 1.44 0.62 0.59 0.95 1.66 2.16  38  24  42  2.63  7.190 10.280
## 127 2.325676 1.44 0.78 0.76 0.95 1.83 1.90  38  26  83  2.63  7.874  9.780
```

```
## 128 2.320130 1.44 0.80 0.58 0.95 1.78 2.20  38  49  76  2.63  7.310 22.590
## 129 2.286669 1.44 0.75 0.53 0.95 1.36 1.90  38  21  75  2.63  2.985 21.020
## 130 2.245337 1.44 0.75 0.64 0.95 2.28 1.60  38  45  41  2.63 12.450  8.570
## 131 3.898680 1.61 1.61 0.76 0.89 0.89 1.38  56  56  92  3.51  3.510 19.050
## 132 2.842476 1.61 1.34 0.58 0.89 1.00 1.90  56  20  75  3.51  1.550 21.020
## 133 2.980121 1.61 1.34 0.62 0.89 1.00 2.36  56  20  74  3.51  1.550 19.250
## 134 2.742683 1.61 0.95 0.58 0.89 1.69 1.90  56  48  75  3.51  8.650 21.020
## 135 2.923199 1.61 0.75 0.76 0.89 1.88 1.38  56  27  92  3.51  8.900 19.050
## 136 2.313990 1.61 0.78 0.58 0.89 1.83 1.90  56  26  75  3.51  7.874 21.020
## 137 2.349364 1.61 0.72 0.58 0.89 1.31 2.20  56  12  76  3.51  1.738 22.590
## 138 2.965490 1.61 0.72 0.76 0.89 1.31 1.38  56  12  92  3.51  1.738 19.050
## 139 2.586063 1.61 0.83 0.62 0.89 1.55 2.36  56  25  25  3.51  7.470 19.250
## 140 2.298125 1.61 0.69 0.58 0.89 1.91 1.90  56  28  75  3.51  8.908 21.020
## 141 2.407594 1.61 0.74 0.59 0.89 1.65 2.16  56  30  42  3.51  7.140 10.280
## 142 3.024295 1.61 0.74 0.76 0.89 1.65 1.38  56  30  92  3.51  7.140 19.050
## 143 2.169202 1.34 0.72 0.62 1.00 1.31 2.36  20  12  74  1.55  1.738 19.250
## 144 2.276169 1.49 0.72 0.62 1.80 1.31 2.36  82  12  74  1.34  1.738 19.250
## 145 2.386660 1.44 0.62 0.76 0.95 1.66 1.38  38  24  92  2.63  7.190 19.050
## 146 2.256332 1.44 0.72 0.56 0.95 1.31 2.10  38  12  52  2.63  1.738  6.240
## 147 2.493627 1.44 0.69 0.76 0.95 1.91 1.38  38  28  92  2.63  8.908 19.050
```

(a) Fit a linear model to the given data.

```r
model2 = summary(lm(z ~., data = data_2))
model2
```

```
##
## Call:
## lm(formula = z ~ ., data = data_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5837 -0.1117  0.0000  0.1110  0.5363
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.378075   2.681674  -1.633  0.10490
## x01          4.126398   1.778562   2.320  0.02185 *
## x02          0.859452   0.139780   6.149 8.40e-09 ***
## x03          0.483531   0.255409   1.893  0.06049 .
## x04          0.860407   0.629628   1.367  0.17406
## x05         -0.246589   0.089624  -2.751  0.00676 **
## x06         -0.189448   0.075635  -2.505  0.01345 *
## x07         -0.019324   0.013616  -1.419  0.15817
## x08          0.007767   0.001630   4.764 4.88e-06 ***
## x09          0.011124   0.001414   7.869 1.06e-12 ***
## x10          0.001597   0.021294   0.075  0.94033
## x11         -0.006227   0.012723  -0.489  0.62534
## x12         -0.026724   0.004372  -6.112 1.00e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1784 on 134 degrees of freedom
```

```
## Multiple R-squared:  0.886,  Adjusted R-squared:  0.8758
## F-statistic: 86.77 on 12 and 134 DF,  p-value: < 2.2e-16
```

```
model_2 = lm(z~., data = data_2 )
```

(b) Check the fit of the model, and identify which coefficients are significant.

The fit of the model is given by the adjusted R^2 value, coefficient of determination. This metric penalizes excess independent variables which don't improve the fit of the model.

```
model2$adj.r.squared
```

```
## [1] 0.8757667
```

The adjusted R squared value of our transformed model is 0.8758, which means it is a very good fit for the data.

The significant coefficients are those coefficients whose p value is less than alpha, 0.05.

```
model2$coefficients
```

```
##                 Estimate  Std. Error     t value     Pr(>|t|)
## (Intercept) -4.378075492 2.681674442 -1.63259023 1.049029e-01
## x01          4.126397924 1.778562158  2.32007518 2.184790e-02
## x02          0.859452105 0.139780050  6.14860349 8.398777e-09
## x03          0.483530509 0.255408919  1.89316219 6.049265e-02
## x04          0.860406879 0.629627930  1.36653226 1.740606e-01
## x05         -0.246589020 0.089624060 -2.75137079 6.756297e-03
## x06         -0.189448114 0.075635474 -2.50475211 1.345259e-02
## x07         -0.019323582 0.013615936 -1.41918868 1.581660e-01
## x08          0.007766526 0.001630422  4.76350652 4.876083e-06
## x09          0.011123858 0.001413590  7.86922283 1.064344e-12
## x10          0.001597040 0.021293737  0.07500044 9.403262e-01
## x11         -0.006227154 0.012723281 -0.48942986 6.253376e-01
## x12         -0.026723800 0.004372327 -6.11203170 1.004985e-08
```

The significant variables in our model are x01,x02,x05,x06,x08,x09,x12.

Let's create a reduced model where we only use the significant variables.

```
reduced_model2 = summary(lm(z ~x01 + x02 + x05 + x06 + x08 + x09 + x12, data = data_2))
reduced_model2
```

```
##
## Call:
## lm(formula = z ~ x01 + x02 + x05 + x06 + x08 + x09 + x12, data = data_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49025 -0.12162 -0.00619  0.10761  0.62019
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.220140   0.370726  -0.594    0.554
## x01          1.627115   0.213029   7.638 3.23e-12 ***
## x02          0.867765   0.139229   6.233 5.14e-09 ***
## x05         -0.296834   0.054132  -5.483 1.91e-07 ***
## x06         -0.262356   0.063709  -4.118 6.52e-05 ***
## x08          0.007330   0.000959   7.643 3.14e-12 ***
## x09          0.012327   0.001163  10.603  < 2e-16 ***
## x12         -0.029407   0.003934  -7.475 7.87e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.179 on 139 degrees of freedom
## Multiple R-squared:  0.8809, Adjusted R-squared:  0.8749
## F-statistic: 146.8 on 7 and 139 DF,  p-value: < 2.2e-16
```

```
reduced_model_2 = lm(z ~x01 + x02 + x05 + x06 + x08 + x09 + x12, data = data_2)
```

```
model2$adj.r.squared
```

```
## [1] 0.8757667
```

```
reduced_model2$adj.r.squared
```

```
## [1] 0.8748587
```

```
anova(model_2, reduced_model_2)
```

```
## Analysis of Variance Table
##
## Model 1: z ~ x01 + x02 + x03 + x04 + x05 + x06 + x07 + x08 + x09 + x10 +
##     x11 + x12
## Model 2: z ~ x01 + x02 + x05 + x06 + x08 + x09 + x12
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    134 4.2643
## 2    139 4.4558 -5  -0.19144 1.2032 0.3111
```

The adjusted R squared of the reduced model is 0.8749, which is slightly lower than the adj. R squared of the full model. However, since the p-value in the anova test is 0.3111, we can say the difference is insignificant, and both models have around the same predictive power.

(c) Compare this model with the previous linear model. Which of the two models is a better fit for the given data? Justify your answer.

First, we compare the two full models.

We compared the adjusted R squared value of both models.

```
model1$adj.r.squared
```

```
## [1] 0.7998815
```

```r
model2$adj.r.squared
```

```
## [1] 0.8757667
```

The higher adjusted r squared of the second model means that the model with the transformed dependent variable z is better fitting compared to the first model. It can explain more of the variation of the transformed dependent variable.

Next, we check the BIC metric of both models.

```r
# For Model 1
varsel_1 = regsubsets(y~., data=data_1, nvmax=nvar)
varsel.res_1 = summary(varsel_1)
varsel.metric_1 = cbind(1:nvar, varsel.res_1$bic)
colnames(varsel.metric_1) = c("No. of Variables", "BIC")
varsel.metric_1
```

```
##       No. of Variables        BIC
##  [1,]                1  -66.37373
##  [2,]                2 -128.75107
##  [3,]                3 -150.44266
##  [4,]                4 -173.05911
##  [5,]                5 -188.38451
##  [6,]                6 -198.19422
##  [7,]                7 -204.55991
##  [8,]                8 -201.83125
##  [9,]                9 -197.06974
## [10,]               10 -194.14157
## [11,]               11 -189.22147
## [12,]               12 -184.23238
```

```r
# For Model 2
varsel_2 = regsubsets(z~., data=data_2, nvmax=nvar)
varsel.res_2 = summary(varsel_2)
varsel.metric_2 = cbind(1:nvar, varsel.res_2$bic)
colnames(varsel.metric_2) = c("No. of Variables", "BIC")
varsel.metric_2
```

```
##       No. of Variables        BIC
##  [1,]                1  -82.40367
##  [2,]                2 -145.38595
##  [3,]                3 -183.76567
##  [4,]                4 -207.08555
##  [5,]                5 -240.61664
##  [6,]                6 -260.87950
##  [7,]                7 -272.81093
##  [8,]                8 -271.80935
##  [9,]                9 -267.07592
## [10,]               10 -264.02955
## [11,]               11 -259.29865
## [12,]               12 -254.31438
```

In terms of BIC, looking at the full model (no. of variables = 12), the second model has a significantly lower BIC, -254.314 compared to the first model, -184.232.

We can also compare the residual standard error of both models, where a lower residual means that the model is better fitting.

```
model1$sigma
```

```
## [1] 4.823953
```

```
model2$sigma
```

```
## [1] 0.1783906
```

The residual standard error of model2 is significantly less than that of model1.

Lastly, we have to check if the residuals of both models are normal and uncorrelated. We use the QQ plot, Shapiro Wilk, and Anderson Darling tests.

```
ad.test(model1$residuals) #Anderson-Darling
```

```
##
##  Anderson-Darling normality test
##
## data:  model1$residuals
## A = 0.36558, p-value = 0.4312
```

```
shapiro.test(model1$residuals) #Shapiro-Wilk
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model1$residuals
## W = 0.99368, p-value = 0.7691
```

```
ad.test(model2$residuals) #Anderson-Darling
```

```
##
##  Anderson-Darling normality test
##
## data:  model2$residuals
## A = 0.43593, p-value = 0.2945
```

```
shapiro.test(model2$residuals) #Shapiro-Wilk
```

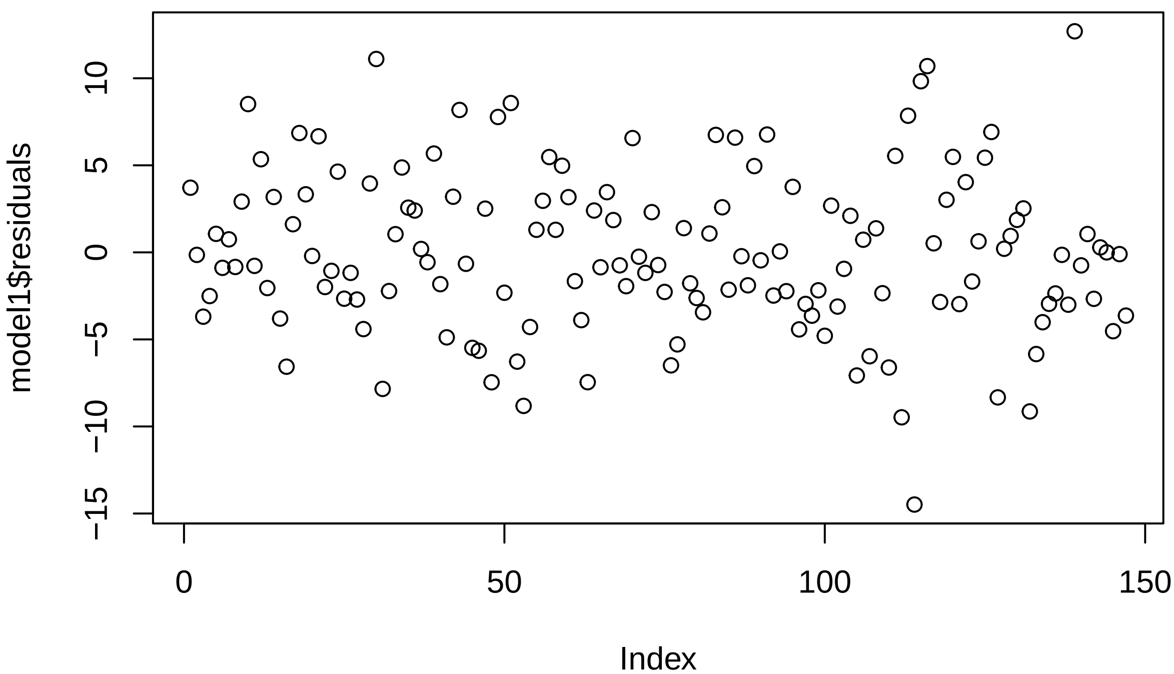
```
##
##  Shapiro-Wilk normality test
##
## data:  model2$residuals
## W = 0.99126, p-value = 0.5003
```

The p-values of the 1st and 2nd models under the AD and SW normality tests are greater than 0.05. This means that we can assume that the residuals from both models are normally distributed.

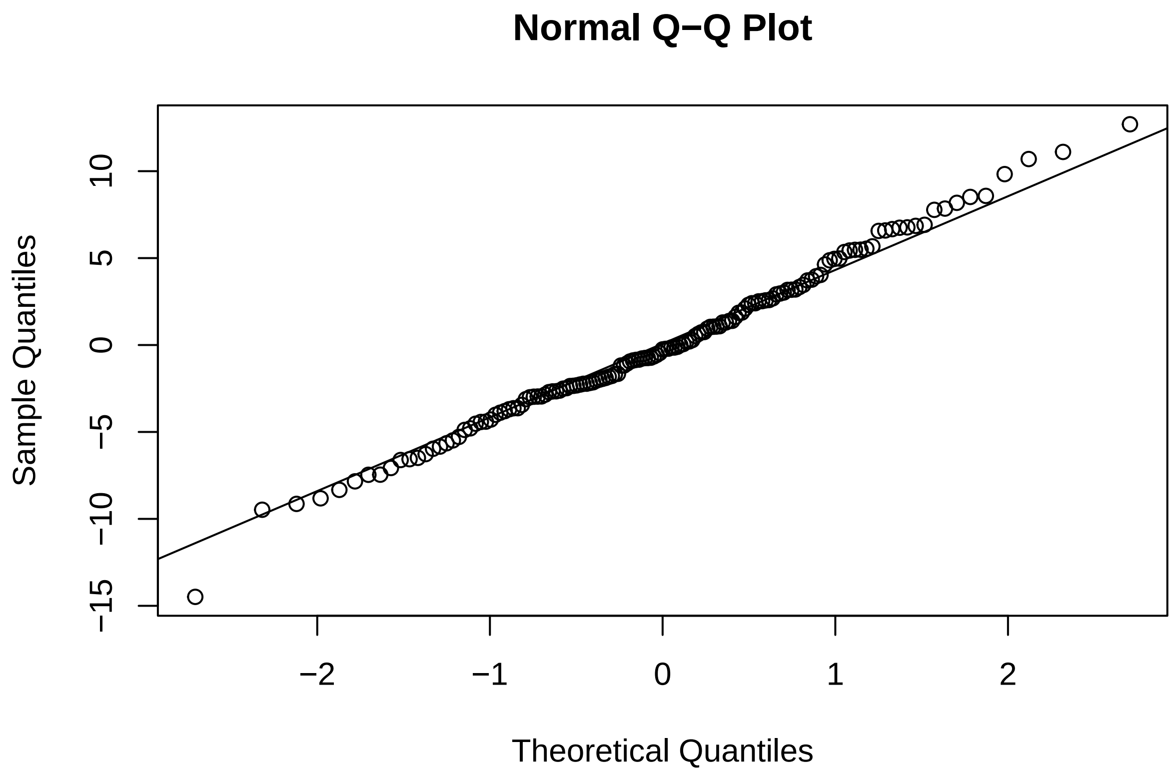
The next question is which residuals are more normal?

For Model 1

```
plot(model1$residuals)
```

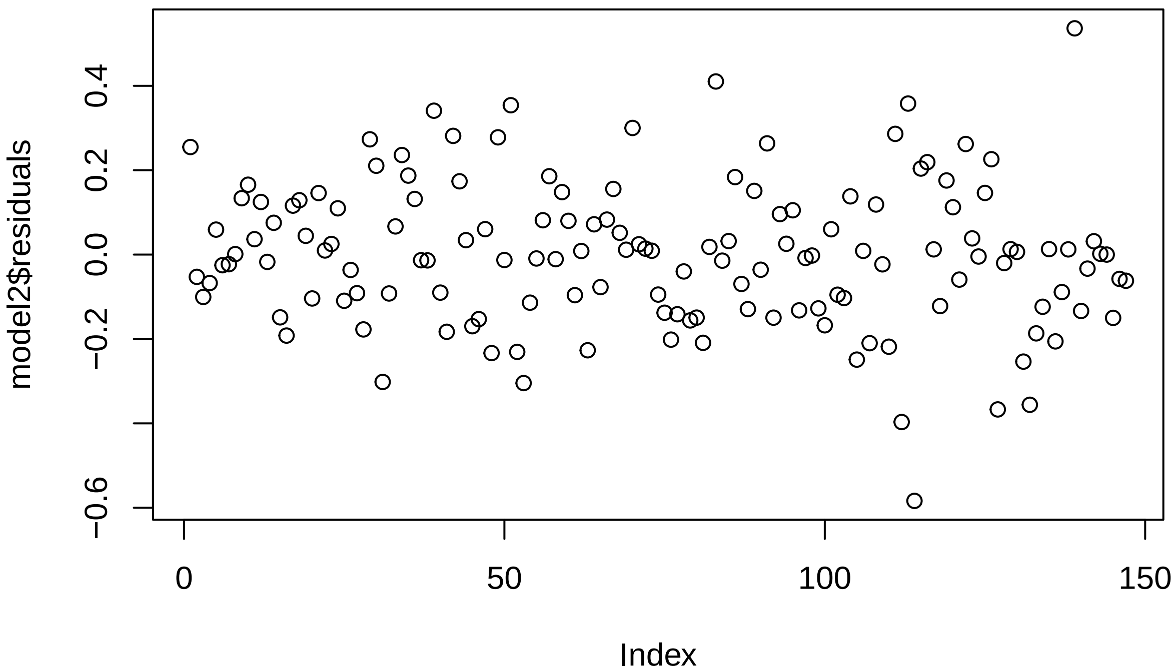



```
qqnorm(model1$residuals)
qqline(model1$residuals)
```

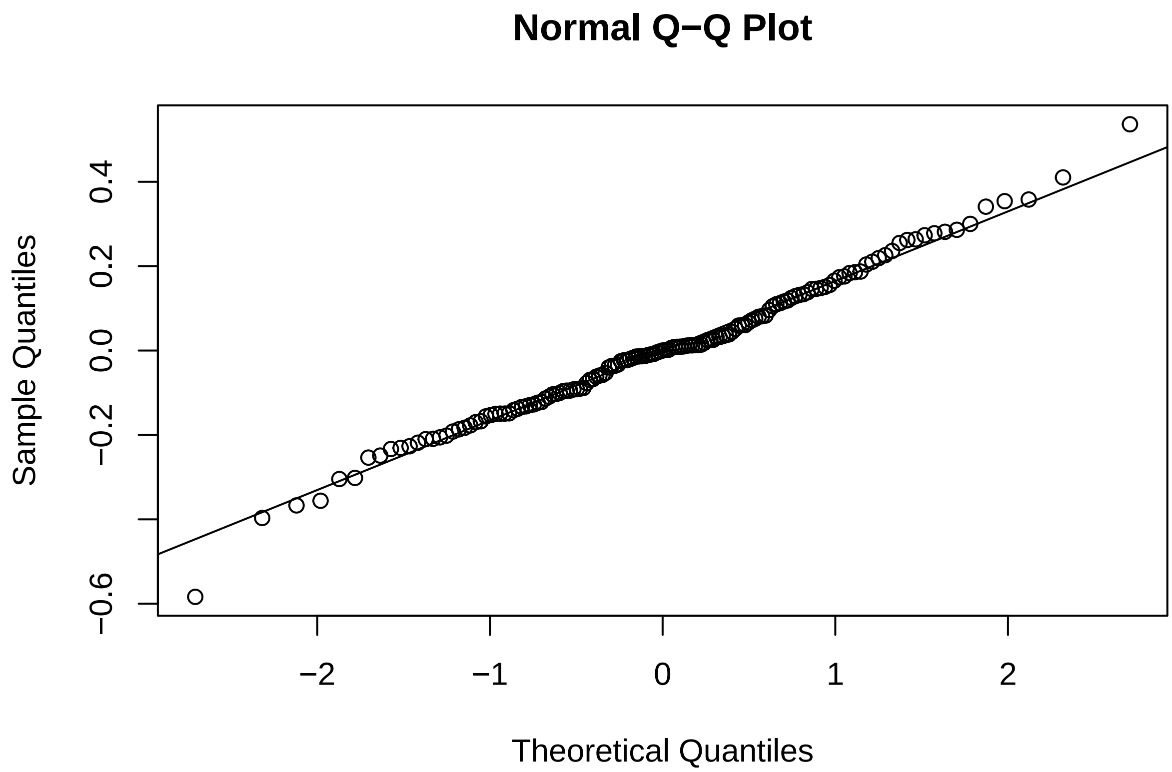


For Model 2 (transformed y)

```
plot(model2$residuals)
```




```
qqnorm(model2$residuals)
qqline(model2$residuals)
```



Qualitatively, we can see that the residuals of both models are randomly scattered and distributed, meaning that they are uncorrelated. Looking at the QQ Plot, the second model is a slightly better match on the normal line compared to the first model.

Final Comparison between normal linear model and transformed linear model

```
# Model 1
cat("Comparison of the two full models\n")
```

```
## Comparison of the two full models
```

```
cat("Model 1:", "\n", "Adjusted R squared =", model1$adj.r.squared, "\n", "BIC value = ", BIC(model_1),
```

```
## Model 1:
##  Adjusted R squared = 0.7998815
##  BIC value =  936.0594
##  Residual Standard Error =  4.823953
```

```
# Model 2
cat("Model 2:", "\n", "Adjusted R squared =", model2$adj.r.squared, "\n", "BIC value = ", BIC(model_2),
```

```
## Model 2:
##  Adjusted R squared = 0.8757667
##  BIC value =  -33.36845
##  Residual Standard Error =  0.1783906
```

Thus, we can conclude that for the full models with all 12 variables, the second model with the transformed dependent variable z is better fitting compared to the first model.