TimeTravel

Gao

2023-11-08

Load libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidymodels)
```

```
## -- Attaching packages --------------------------------------- tidymodels 1.1.1 --
## v broom        1.0.5     v rsample      1.2.0
## v dials        1.2.0     v tune         1.1.2
## v infer        1.0.5     v workflows    1.1.3
## v modeldata    1.2.0     v workflowsets 1.0.1
## v parsnip      1.1.1     v yardstick    1.2.0
## v recipes      1.0.8
## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```
library(ggforce)
library(mctest)
library(olsrr)
```

```
##
## Attaching package: 'olsrr'
```

```
##
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
library(jtools)
```

```
##
## Attaching package: 'jtools'
##
## The following object is masked from 'package:yardstick':
##
##     get_weights
```

```
library(ggcorrplot)
library(yardstick)
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(moments)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following object is masked from 'package:car':
##
##     logit
##
## The following objects are masked from 'package:scales':
##
##     alpha, rescale
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(fastDummies)
```

```
## Thank you for using fastDummies!
## To acknowledge our work, please cite the package:
## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from
```

Load data, and summarize

```
Travel <- read_csv("Travel_Times.csv") %>% as_tibble()
```

```
## New names:
## Rows: 205 Columns: 13
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (4): DayOfWeek, GoingTo, FuelEconomy, Take407All dbl (7): Observation,
## Distance, MaxSpeed, AvgSpeed, AvgMovingSpeed, TotalTi... lgl (1): ...2 time
## (1): StartTime
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...2`
```

```
summary(Travel)
```

```
##   Observation    ...2          StartTime         DayOfWeek
##  Min.   :  1   Mode:logical   Length:205        Length:205
##  1st Qu.: 52   NA's:205       Class1:hms        Class :character
##  Median :103                  Class2:difftime   Mode  :character
##  Mean   :103                  Mode  :numeric
##  3rd Qu.:154
##  Max.   :205
##    GoingTo             Distance        MaxSpeed        AvgSpeed
##  Length:205         Min.   :48.32   Min.   :112.2   Min.   : 38.10
##  Class :character   1st Qu.:50.65   1st Qu.:124.9   1st Qu.: 68.90
##  Mode  :character   Median :51.14   Median :127.4   Median : 73.60
##                     Mean   :50.98   Mean   :127.6   Mean   : 74.48
##                     3rd Qu.:51.63   3rd Qu.:129.8   3rd Qu.: 79.90
##                     Max.   :60.32   Max.   :140.9   Max.   :107.70
##  AvgMovingSpeed   FuelEconomy          TotalTime      MovingTime
##  Min.   : 50.30   Length:205         Min.   :28.2   Min.   :27.10
##  1st Qu.: 76.60   Class :character   1st Qu.:38.4   1st Qu.:35.70
##  Median : 81.40   Mode  :character   Median :41.3   Median :37.60
##  Mean   : 81.98                      Mean   :41.9   Mean   :37.87
##  3rd Qu.: 86.00                      3rd Qu.:44.4   3rd Qu.:39.90
##  Max.   :112.10                      Max.   :82.3   Max.   :62.40
##   Take407All
##  Length:205
##  Class :character
##  Mode  :character
##
##
##
```

```
cor(Travel$TotalTime, select_if(Travel, is.numeric))
```

```
##      Observation  Distance   MaxSpeed   AvgSpeed AvgMovingSpeed TotalTime
## [1,]  0.08637486 0.1972073 -0.1987747 -0.8778056     -0.8569861         1
##      MovingTime
## [1,]  0.9209345
```

```
Travel <- dummy_cols(Travel, select_columns = "Take407All", remove_first_dummy = TRUE)
Travel
```

```
## # A tibble: 205 x 14
##    Observation ...2  StartTime DayOfWeek GoingTo Distance MaxSpeed AvgSpeed
##          <dbl> <lgl> <time>    <chr>     <chr>      <dbl>    <dbl>    <dbl>
##  1           1 NA    16:37     Friday    Home        51.3     127.     78.3
##  2           2 NA    08:20     Friday    GSK         51.6     130.     81.8
##  3           3 NA    16:17     Wednesday Home        51.3     127.     82
##  4           4 NA    07:53     Wednesday GSK         49.2     132.     74.2
##  5           5 NA    18:57     Tuesday   Home        51.2     136.     83.4
##  6           6 NA    07:57     Tuesday   GSK         51.8     136.     84.5
##  7           7 NA    17:31     Monday    Home        51.4     123.     82.9
##  8           8 NA    07:34     Monday    GSK         49.0     128.     77.5
##  9           9 NA    08:01     Friday    GSK         52.9     130.     80.9
## 10          10 NA    17:19     Thursday  Home        51.2     122.     70.6
## # i 195 more rows
## # i 6 more variables: AvgMovingSpeed <dbl>, FuelEconomy <chr>, TotalTime <dbl>,
## #   MovingTime <dbl>, Take407All <chr>, Take407All_Yes <int>
```

```
model <- lm(TotalTime ~ Distance + MaxSpeed + AvgSpeed + AvgMovingSpeed + MovingTime + Take407All_Yes, d
model
```

```
##
## Call:
## lm(formula = TotalTime ~ Distance + MaxSpeed + AvgSpeed + AvgMovingSpeed +
##     MovingTime + Take407All_Yes, data = Travel)
##
## Coefficients:
##    (Intercept)        Distance        MaxSpeed        AvgSpeed  AvgMovingSpeed
##      -12.92568         0.06059         0.03559        -0.28908         0.24437
##     MovingTime  Take407All_Yes
##        1.27988         1.32496
```

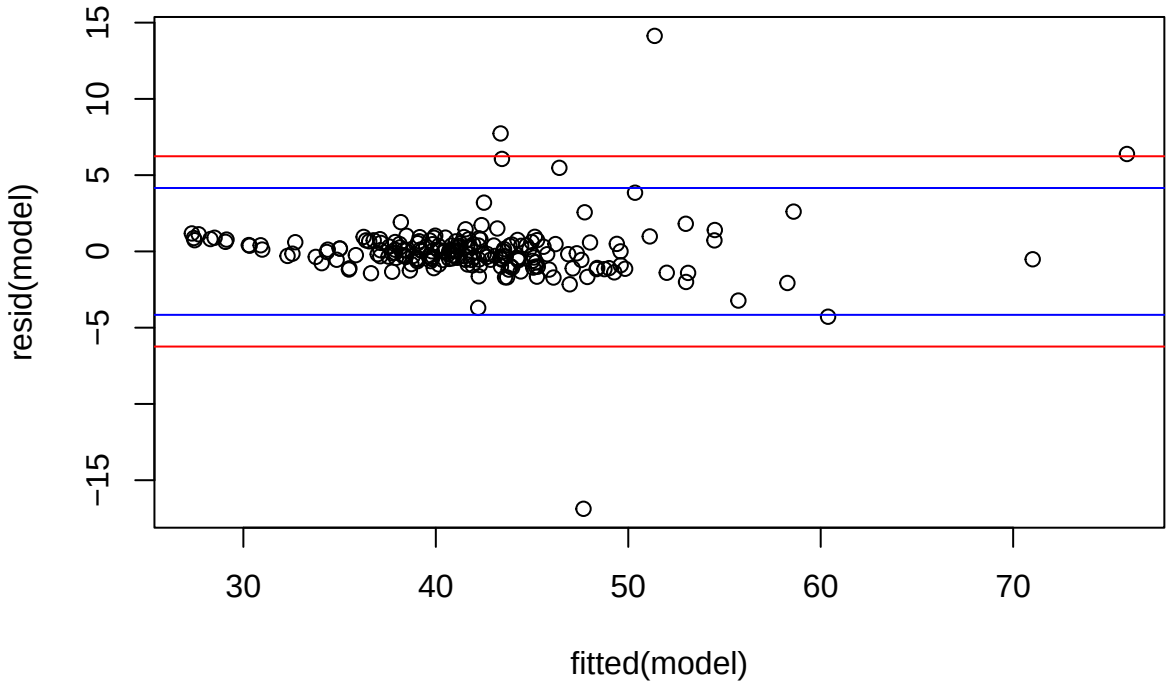
```
standard_error <- sqrt(deviance(model)/df.residual(model))
standard_error
```

```
## [1] 2.079387
```

```
2*standard_error
```

```
## [1] 4.158774
```

```r
plot(fitted(model),resid(model))
abline(h=2*standard_error, col = "blue")
abline(h=-2*standard_error, col = "blue")
abline(h=3*standard_error, col = "red")
abline(h=-3*standard_error, col = "red")
```



3 values may be outliers and 3 values here are outliers, as they lie outside of the 3SD lines. The residual plot shows how much the actual data deviates from the predicted value

```r
res_pot_outliers <- Travel %>% filter(2*standard_error <= abs(resid(model)) & abs(resid(model)) < 3*sta
print(res_pot_outliers)
```

```
## # A tibble: 3 x 14
##   Observation ...2  StartTime DayOfWeek GoingTo Distance MaxSpeed AvgSpeed
##         <dbl> <lgl> <time>    <chr>     <chr>      <dbl>    <dbl>    <dbl>
## 1          58 NA    07:19     Monday    GSK         51.7     126.     55.2
## 2         204 NA    17:51     Tuesday   Home        53.3     126.     61.6
## 3         205 NA    16:56     Monday    Home        51.7     125      62.8
## # i 6 more variables: AvgMovingSpeed <dbl>, FuelEconomy <chr>, TotalTime <dbl>,
## #   MovingTime <dbl>, Take407All <chr>, Take407All_Yes <int>
```

```r
res_outliers <- Travel %>% filter(abs(resid(model)) >= 3*standard_error)
print(res_pot_outliers)
```

```
## # A tibble: 3 x 14
```

```
##   Observation ...2  StartTime DayOfWeek GoingTo Distance MaxSpeed AvgSpeed
##         <dbl> <lgl> <time>    <chr>     <chr>      <dbl>    <dbl>    <dbl>
## 1          58 NA    07:19     Monday    GSK         51.7     126.     55.2
## 2         204 NA    17:51     Tuesday   Home        53.3     126.     61.6
## 3         205 NA    16:56     Monday    Home        51.7     125      62.8
## # i 6 more variables: AvgMovingSpeed <dbl>, FuelEconomy <chr>, TotalTime <dbl>,
## #   MovingTime <dbl>, Take407All <chr>, Take407All_Yes <int>
```

```
h <- 2*(6+1)/205
h
```

```
## [1] 0.06829268
```

```
leverage<-hatvalues(model)
sort(round(leverage,4))
```

```
##     41     81    170    192     56    122    118     51     52    100    194
## 0.0065 0.0065 0.0066 0.0066 0.0071 0.0072 0.0075 0.0076 0.0079 0.0079 0.0080
##     37     78     83     48     59    124     54     69     89     73    172
## 0.0083 0.0084 0.0084 0.0085 0.0085 0.0085 0.0089 0.0090 0.0090 0.0091 0.0091
##     28     13     43    156    114      1     47     23     72     86    160
## 0.0092 0.0093 0.0093 0.0093 0.0094 0.0095 0.0096 0.0097 0.0097 0.0097 0.0097
##    180     42    158    186     53     46     85     80    152    154     64
## 0.0097 0.0098 0.0098 0.0098 0.0099 0.0100 0.0100 0.0101 0.0101 0.0102 0.0103
##     65    110     12     19     44     35     38    102    176     27     98
## 0.0103 0.0105 0.0106 0.0106 0.0106 0.0107 0.0107 0.0107 0.0107 0.0109 0.0109
##     66     30     34    120    128    144     62    106      3     31     84
## 0.0110 0.0112 0.0115 0.0115 0.0117 0.0118 0.0120 0.0120 0.0121 0.0121 0.0124
##    182    104     20     21     88    178     15    108    164    174    188
## 0.0124 0.0125 0.0129 0.0130 0.0132 0.0134 0.0135 0.0135 0.0138 0.0138 0.0138
##     68     32     10     36     74     22     55    136    175      2     39
## 0.0141 0.0143 0.0145 0.0145 0.0145 0.0157 0.0160 0.0162 0.0162 0.0168 0.0170
##    153     33     24     67    132    161     87    146    165     70    139
## 0.0170 0.0172 0.0173 0.0175 0.0175 0.0176 0.0178 0.0179 0.0179 0.0180 0.0180
##    138     11    171    177     25     60    155    168    129     18    167
## 0.0184 0.0187 0.0188 0.0189 0.0192 0.0192 0.0192 0.0192 0.0194 0.0195 0.0195
##    145    162     71    121     17    147    189    141    169    191     29
## 0.0199 0.0199 0.0201 0.0203 0.0207 0.0207 0.0208 0.0209 0.0212 0.0213 0.0214
##    179     77     26    157     61    126      7     49      9    181      8
## 0.0215 0.0217 0.0218 0.0221 0.0225 0.0226 0.0231 0.0231 0.0235 0.0236 0.0238
##    163     79    148      4    151    185    137     57     94    183    117
## 0.0242 0.0249 0.0255 0.0269 0.0269 0.0269 0.0271 0.0281 0.0282 0.0285 0.0295
##    133    187      6      5    143    115    199    197    103    105    123
## 0.0306 0.0311 0.0314 0.0320 0.0341 0.0356 0.0357 0.0359 0.0360 0.0360 0.0384
##    200    112     76    195     14    130     63    125    131    193     16
## 0.0387 0.0391 0.0393 0.0393 0.0394 0.0400 0.0412 0.0420 0.0420 0.0446 0.0447
##    113     95    109     96    127    142    134    202     90     97    150
## 0.0460 0.0465 0.0477 0.0490 0.0493 0.0493 0.0494 0.0514 0.0516 0.0537 0.0540
##    111    173    101    116    196     75    198    140    107    166     92
## 0.0555 0.0582 0.0586 0.0591 0.0611 0.0617 0.0624 0.0636 0.0650 0.0668 0.0678
##    135     91     58    149    184    159     82    119    205    204     40
## 0.0751 0.0808 0.0814 0.0851 0.0855 0.0869 0.0875 0.1041 0.1046 0.1103 0.1133
##    203     93    201    190     45     50     99
## 0.1167 0.1242 0.1455 0.2557 0.2574 0.3438 0.5592
```

The leverage critical value is 0.06929, and any leverage value exceeding it is thus considered an outlier.

There are 18 outliers, observations 135, 91, 58, 149, 184, 159, 82, 119, 2015, 204, 40, 203, 93, 201, 190, 45, 50 and 99

```
leverage_outliers <- Travel %>% filter(leverage > h)
leverage_outliers
```

```
## # A tibble: 18 x 14
##    Observation ...2  StartTime DayOfWeek GoingTo Distance MaxSpeed AvgSpeed
##          <dbl> <lgl> <time>    <chr>     <chr>      <dbl>    <dbl>    <dbl>
##  1          40 NA    07:23     Tuesday   GSK         51.7     112.     55.3
##  2          45 NA    16:17     Wednesday Home        60.3     129.     68.9
##  3          50 NA    07:24     Monday    GSK         52.2     127.     38.1
##  4          58 NA    07:19     Monday    GSK         51.7     126.     55.2
##  5          82 NA    08:31     Friday    GSK         50.6     129     107.
##  6          91 NA    08:36     Thursday  GSK         50.7     128.    106.
##  7          93 NA    08:28     Wednesday GSK         50.6     128.     59.5
##  8          99 NA    08:22     Thursday  GSK         50.6     126.     38.5
##  9         119 NA    08:10     Tuesday   GSK         51.7     129.     70.4
## 10         135 NA    07:50     Tuesday   GSK         54.4     132.     95.1
## 11         149 NA    09:09     Thursday  GSK         50.4     134.    107.
## 12         159 NA    08:11     Thursday  GSK         52.3     138.     51.2
## 13         184 NA    20:31     Friday    Home        50.7     136.    108.
## 14         190 NA    17:15     Tuesday   Home        51.3     122.     43.7
## 15         201 NA    08:09     Monday    GSK         54.5     126.     49.9
## 16         203 NA    17:08     Wednesday Home        52.0     133.     57.5
## 17         204 NA    17:51     Tuesday   Home        53.3     126.     61.6
## 18         205 NA    16:56     Monday    Home        51.7     125      62.8
## # i 6 more variables: AvgMovingSpeed <dbl>, FuelEconomy <chr>, TotalTime <dbl>,
## #   MovingTime <dbl>, Take407All <chr>, Take407All_Yes <int>
```

```
t <- qt(df = 205 - 6 - 2, 0.95)
t
```

```
## [1] 1.652625
```

```
jackknife <- rstudent(model)
sort(round(jackknife, 4))
```

```
##       99       58      150       45       40      188       93       60
## -24.6073  -2.1684  -1.8405  -1.8078  -1.0569  -1.0451  -1.0291  -0.8312
##      142       98       27       42      128      111       92       49
##  -0.8247  -0.8221  -0.8200  -0.7989  -0.7894  -0.7072  -0.7005  -0.6801
##      148      107      132      116      135      187       33       57
##  -0.6680  -0.6617  -0.6357  -0.6196  -0.5920  -0.5812  -0.5774  -0.5659
##      173      106      166      113       46      108       62      118
##  -0.5624  -0.5593  -0.5457  -0.5446  -0.5370  -0.5299  -0.5243  -0.5135
##      110      100      120       37       72       51       71       79
##  -0.5121  -0.4814  -0.4809  -0.4780  -0.4741  -0.4740  -0.4423  -0.4400
##      165      156       21       68      134      160       88       89
##  -0.4341  -0.4143  -0.4132  -0.4008  -0.3794  -0.3410  -0.3106  -0.3054
##      122      190       52       32      114       64       63      191
```

```
##  -0.2979  -0.2870  -0.2814  -0.2739  -0.2707  -0.2705  -0.2683  -0.2651
##       41      180      170       86       70      185      152       47
##  -0.2621  -0.2602  -0.2576  -0.2567  -0.2447  -0.2393  -0.2369  -0.2367
##       56      140      164       69      139        3       38       39
##  -0.2325  -0.2231  -0.2208  -0.2191  -0.2107  -0.2053  -0.2016  -0.1939
##      153      105       54       96       15      162       12      163
##  -0.1841  -0.1758  -0.1729  -0.1687  -0.1666  -0.1651  -0.1611  -0.1594
##      179        6       73      168       36      103       81      194
##  -0.1547  -0.1535  -0.1522  -0.1498  -0.1485  -0.1440  -0.1420  -0.1324
##       19       48       65      196      129       78       44       20
##  -0.1275  -0.1264  -0.1237  -0.1208  -0.1155  -0.0950  -0.0908  -0.0894
##        5       61       83       22       90       53      146      104
##  -0.0881  -0.0876  -0.0797  -0.0783  -0.0736  -0.0600  -0.0597  -0.0441
##       95      171      178       10      117      177       28      174
##  -0.0184  -0.0168  -0.0022  -0.0015   0.0014   0.0035   0.0039   0.0148
##        1       23      138       29       13        7       59       30
##   0.0175   0.0259   0.0263   0.0274   0.0326   0.0387   0.0394   0.0419
##      182       14        9      195      115      155      124       17
##   0.0497   0.0536   0.0559   0.0565   0.0593   0.0677   0.0748   0.0890
##       26       66      167      176      158       11        2       31
##   0.0934   0.1259   0.1279   0.1298   0.1472   0.1504   0.1520   0.1547
##      189      144       25      126      123      161      181      143
##   0.1708   0.1723   0.1770   0.1820   0.1837   0.1864   0.1866   0.1888
##       24      112      125       87       34      186      147      183
##   0.1895   0.1966   0.2038   0.2181   0.2228   0.2278   0.2338   0.2342
##       55       43      154       77       85      199      130      198
##   0.2390   0.2499   0.2597   0.2696   0.2827   0.2950   0.3022   0.3045
##      109       35       67        4      102      137      131       76
##   0.3092   0.3174   0.3205   0.3236   0.3456   0.3540   0.3545   0.3546
##       94       84      184       80       97      172        8       75
##   0.3587   0.3630   0.3685   0.3765   0.3826   0.3829   0.3896   0.4014
##      175       18       16      149      101       74      121      145
##   0.4157   0.4377   0.4388   0.4394   0.4432   0.4518   0.4566   0.4634
##      200      133      136      157       91       82      127      169
##   0.4654   0.4793   0.4948   0.5002   0.5625   0.5921   0.6909   0.7048
##      141      119      197      202      151      159      193      203
##   0.7304   0.8748   0.8870   0.9468   1.2521   1.3155   1.5793   1.9840
##      204      205      192       50      201
##   2.8436   3.1504   3.8607   3.9275   8.5992
```

There are 10 potential outliers, as their jackknife values exceed +/- 1.6526

These observations are 99, 58, 150, 45, 203, 204, 205, 192, 50 and 201

```
jackknife_outliers <- Travel %>% filter(jackknife > t | jackknife < -t)
jackknife_outliers
```

```
## # A tibble: 10 x 14
##    Observation ...2  StartTime DayOfWeek GoingTo Distance MaxSpeed AvgSpeed
##          <dbl> <lgl> <time>    <chr>     <chr>      <dbl>    <dbl>    <dbl>
##  1          45 NA    16:17     Wednesday Home        60.3     129.     68.9
##  2          50 NA    07:24     Monday    GSK         52.2     127.     38.1
##  3          58 NA    07:19     Monday    GSK         51.7     126.     55.2
##  4          99 NA    08:22     Thursday  GSK         50.6     126.     38.5
```

```
##  5         150 NA    16:47     Wednesday Home        51.0     133.     79.6
##  6         192 NA    16:59     Monday    Home        51.0     127.     70.4
##  7         201 NA    08:09     Monday    GSK         54.5     126.     49.9
##  8         203 NA    17:08     Wednesday Home        52.0     133.     57.5
##  9         204 NA    17:51     Tuesday   Home        53.3     126.     61.6
## 10         205 NA    16:56     Monday    Home        51.7     125      62.8
## # i 6 more variables: AvgMovingSpeed <dbl>, FuelEconomy <chr>, TotalTime <dbl>,
## #   MovingTime <dbl>, Take407All <chr>, Take407All_Yes <int>
```

```
cookCV <- 4/205
cookCV
```

```
## [1] 0.0195122
```

```
cook <- cooks.distance(model)
sort(round(cook, 4))
```

```
##      1      5      7      9     10     12     13     14     17     19
## 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##     20     22     23     26     28     29     30     31     36     44
## 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##     48     53     54     59     61     65     66     73     78     81
## 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##     83     90     95    104    115    117    124    129    138    146
## 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##    155    158    167    171    174    176    177    178    182    194
## 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
##    195      2      3      6     11     15     24     25     34     38
## 0.0000 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001
##     39     41     43     47     52     55     56     64     69     85
## 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001
##     86     87     89    103    114    122    126    139    144    152
## 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001
##    153    154    161    162    163    164    168    170    179    180
## 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001
##    181    186    189    196     32     35     51     70     77     80
## 0.0001 0.0001 0.0001 0.0001 0.0002 0.0002 0.0002 0.0002 0.0002 0.0002
##     84     88     96    102    105    112    123    143    147    156
## 0.0002 0.0002 0.0002 0.0002 0.0002 0.0002 0.0002 0.0002 0.0002 0.0002
##    160    172    183    185    191     21     37     67     68     72
## 0.0002 0.0002 0.0002 0.0002 0.0002 0.0003 0.0003 0.0003 0.0003 0.0003
##    100    118    125      4     46     63     74    110    120    175
## 0.0003 0.0003 0.0003 0.0004 0.0004 0.0004 0.0004 0.0004 0.0004 0.0004
##      8     18     62     94    106    130    137    140    165    199
## 0.0005 0.0005 0.0005 0.0005 0.0005 0.0005 0.0005 0.0005 0.0005 0.0005
##     71    108    121    136    145     76     79    109     33    131
## 0.0006 0.0006 0.0006 0.0006 0.0006 0.0007 0.0007 0.0007 0.0008 0.0008
##    157     42    198    132    133     27     98    128    134     97
## 0.0008 0.0009 0.0009 0.0010 0.0010 0.0011 0.0011 0.0011 0.0011 0.0012
##     16     57    200     75    169     49    141    187    148    101
## 0.0013 0.0013 0.0013 0.0015 0.0015 0.0016 0.0016 0.0016 0.0017 0.0018
##    184     60    113    188    149    173    166    116    127     91
## 0.0018 0.0019 0.0020 0.0022 0.0026 0.0028 0.0031 0.0035 0.0035 0.0040
```

```
##     135     190     111     197     107      82     142      92     151     202
##  0.0041  0.0041  0.0042  0.0042  0.0044  0.0048  0.0050  0.0051  0.0062  0.0069
##     119     192     193      40      93     159     150      58     203     204
##  0.0127  0.0133  0.0165  0.0204  0.0214  0.0234  0.0273  0.0584  0.0732  0.1383
##     205      45      50     201      99
##  0.1586  0.1600  1.0762  1.3147 27.0780
```

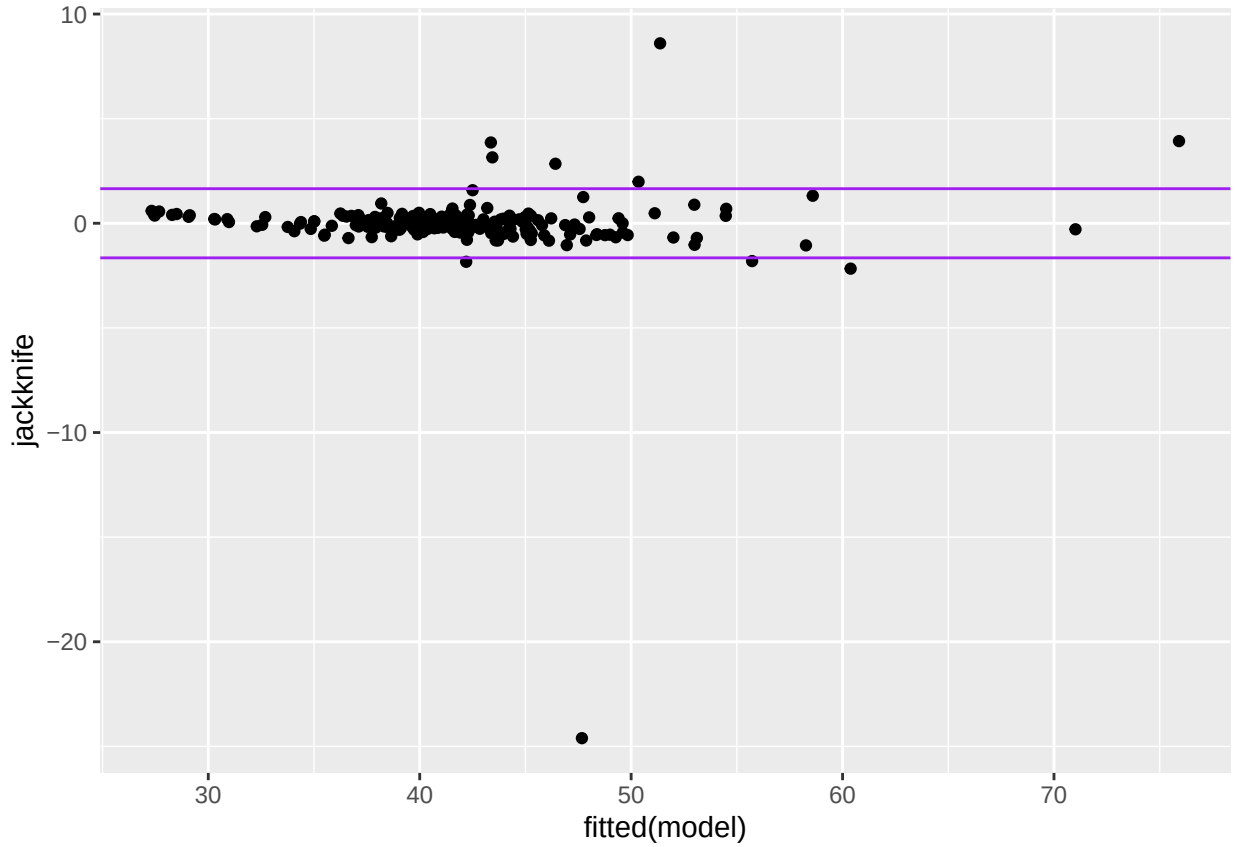
Any value that exceeds 0.0195 is considered an outlier, thus there are 12 total outliers

These are observations 40, 93, 159, 150, 58, 203, 204, 205, 45, 50, 201, 99

```
cook_outliers <- Travel %>% filter(cook > cookCV)
cook_outliers
```

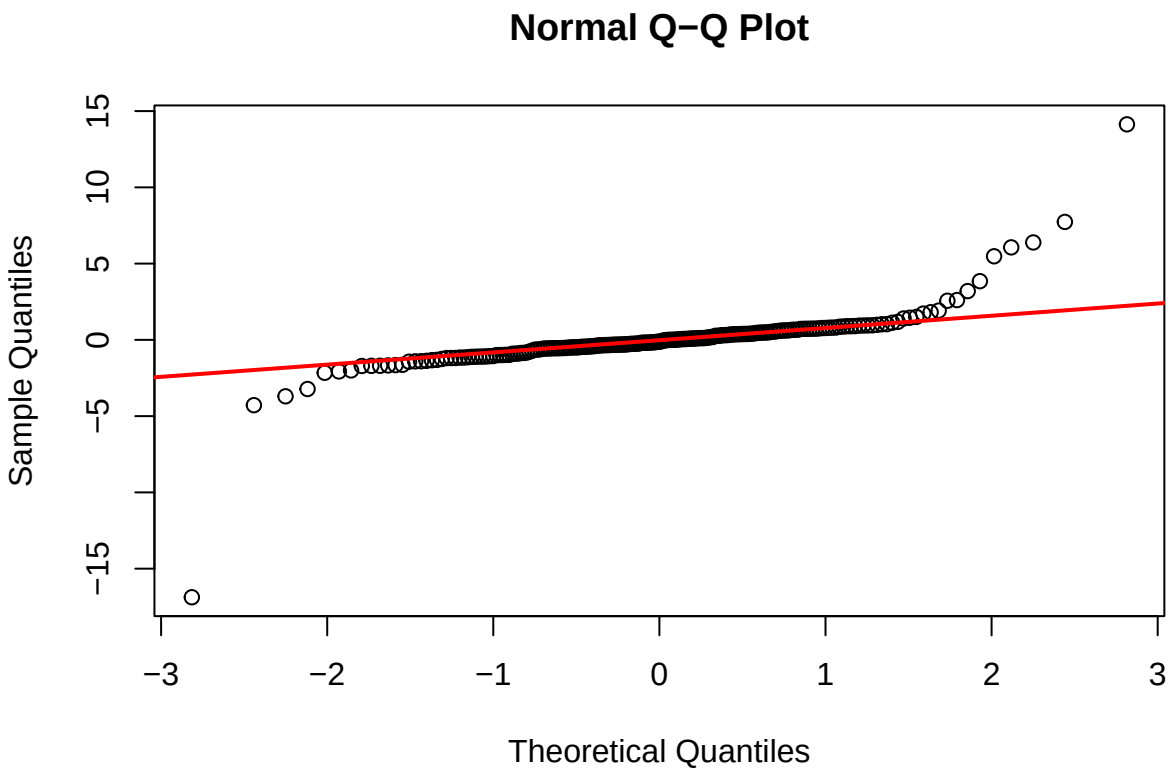
```
## # A tibble: 12 x 14
##    Observation ...2  StartTime DayOfWeek GoingTo Distance MaxSpeed AvgSpeed
##          <dbl> <lgl> <time>    <chr>     <chr>      <dbl>    <dbl>    <dbl>
##  1          40 NA    07:23     Tuesday   GSK         51.7     112.     55.3
##  2          45 NA    16:17     Wednesday Home        60.3     129.     68.9
##  3          50 NA    07:24     Monday    GSK         52.2     127.     38.1
##  4          58 NA    07:19     Monday    GSK         51.7     126.     55.2
##  5          93 NA    08:28     Wednesday GSK         50.6     128.     59.5
##  6          99 NA    08:22     Thursday  GSK         50.6     126.     38.5
##  7         150 NA    16:47     Wednesday Home        51.0     133.     79.6
##  8         159 NA    08:11     Thursday  GSK         52.3     138.     51.2
##  9         201 NA    08:09     Monday    GSK         54.5     126.     49.9
## 10         203 NA    17:08     Wednesday Home        52.0     133.     57.5
## 11         204 NA    17:51     Tuesday   Home        53.3     126.     61.6
## 12         205 NA    16:56     Monday    Home        51.7     125      62.8
## # i 6 more variables: AvgMovingSpeed <dbl>, FuelEconomy <chr>, TotalTime <dbl>,
## #   MovingTime <dbl>, Take407All <chr>, Take407All_Yes <int>
```

```
ggplot(Travel, aes(x = fitted(model), y = jackknife)) + geom_point()+ geom_hline(yintercept = t, col =
```



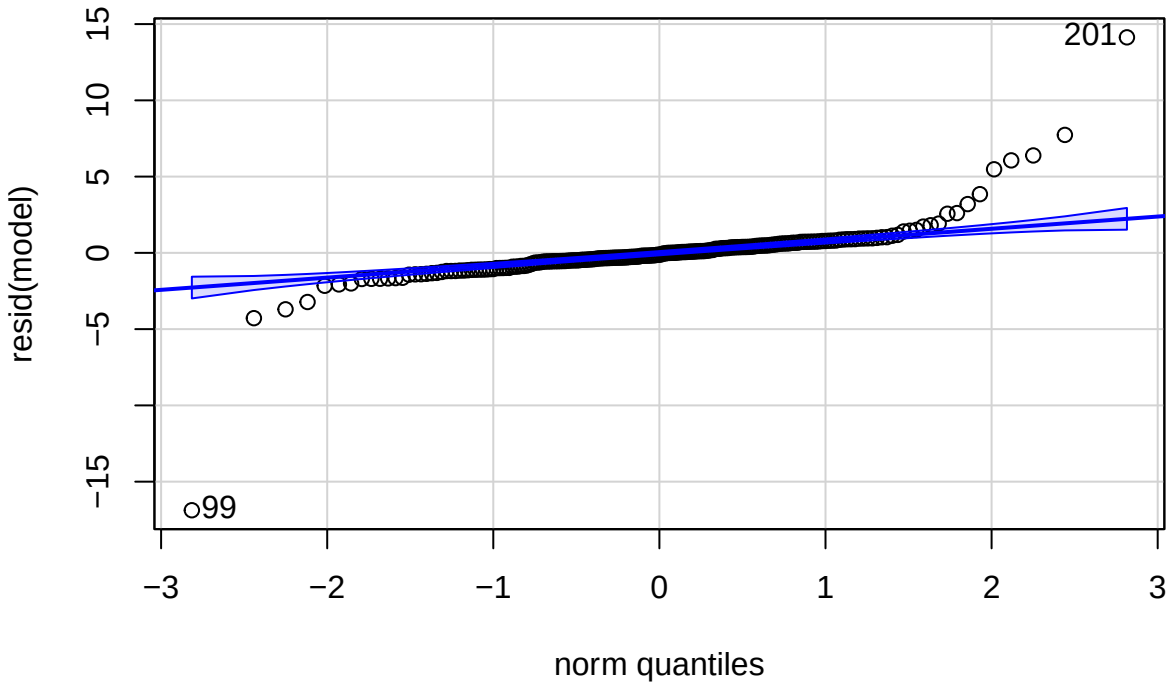
Though it is a bit hard to see, there are 10 outliers

```
qqnorm(resid(model))
qqline(resid(model), col = "red", lwd = 2)
```



The QQPlot tests the normality of the data. Our data is not normal because the plot displays curvature, especially at the +x end

```
qqPlot(resid(model))
```




```
## [1]  99 201
```

99 and 201 are the extreme outliers here

```
skewness(jackknife)
```

```
## [1] -9.211245
```

```
kurtosis(jackknife)
```

```
## [1] 125.1368
```

This shows that the normality assumption is violated, as neither value is very close to 0.

```
ols_vif_tol(model)
```

```
##        Variables  Tolerance       VIF
## 1       Distance 0.51930698  1.925643
## 2       MaxSpeed 0.90519386  1.104736
## 3       AvgSpeed 0.23072522  4.334160
## 4 AvgMovingSpeed 0.03657195 27.343357
## 5     MovingTime 0.05203649 19.217285
## 6 Take407All_Yes 0.41432793  2.413547
```

Tolerance is less than 0.1 for avgmovingspeed and movingtime, and VIF is greater than 10 for those same variables, thus we have colinearity issues

```
eigprop(model)
```

```
##
## Call:
## eigprop(mod = model)
##
##   Eigenvalues       CI (Intercept) Distance MaxSpeed AvgSpeed AvgMovingSpeed
## 1      6.1604   1.0000      0.0000   0.0000   0.0000   0.0001         0.0000
## 2      0.8068   2.7633      0.0000   0.0000   0.0000   0.0000         0.0000
## 3      0.0287  14.6566      0.0001   0.0001   0.0001   0.0653         0.0030
## 4      0.0028  46.6586      0.0028   0.0024   0.0194   0.8945         0.0529
## 5      0.0008  85.4610      0.0003   0.0483   0.8052   0.0147         0.0815
## 6      0.0003 142.3136      0.4562   0.5388   0.1114   0.0010         0.0223
## 7      0.0002 187.4948      0.5407   0.4103   0.0639   0.0244         0.8403
##   MovingTime Take407All_Yes
## 1     0.0000         0.0022
## 2     0.0000         0.3999
## 3     0.0128         0.2529
## 4     0.0411         0.0376
## 5     0.0319         0.0412
## 6     0.0217         0.0075
## 7     0.8924         0.2586
##
## ================================
## Row 6==> Distance, proportion 0.538788 >= 0.50
## Row 5==> MaxSpeed, proportion 0.805152 >= 0.50
## Row 4==> AvgSpeed, proportion 0.894520 >= 0.50
## Row 7==> AvgMovingSpeed, proportion 0.840307 >= 0.50
## Row 7==> MovingTime, proportion 0.892378 >= 0.50
```

Four of the independent variables have CI scores in excess of 30, but their eigenvalues are below 0.9.

```
ols_step_forward_p(model)
```

```
##
##                               Selection Summary
## ------------------------------------------------------------------------------------
##         Variable                        Adj.
## Step       Entered        R-Square    R-Square      C(p)        AIC        RMSE
## ------------------------------------------------------------------------------------
##    1    MovingTime          0.8481      0.8474    135.1814    989.3162    2.6759
##    2    AvgSpeed            0.8869      0.8858     51.4060    930.9299    2.3152
##    3    AvgMovingSpeed      0.9080      0.9066      6.6455    890.5558    2.0930
##    4    Take407All_Yes      0.9100      0.9082      4.1740    888.0044    2.0751
## ------------------------------------------------------------------------------------
```

```
ols_step_backward_p(model)
```

```
##
```

```
##
##                          Elimination Summary
## ----------------------------------------------------------------------
##         Variable                 Adj.
## Step    Removed     R-Square    R-Square     C(p)       AIC        RMSE
## ----------------------------------------------------------------------
##    1    Distance      0.9105      0.9082    5.1570    888.9550    2.0750
##    2    MaxSpeed       0.910      0.9082    4.1740    888.0044    2.0751
## ----------------------------------------------------------------------
```

```
ols_step_both_p(model)
```

```
##
##                                Stepwise Selection Summary
## ----------------------------------------------------------------------------------------------
##                            Added/                   Adj.
## Step       Variable       Removed     R-Square    R-Square      C(p)        AIC        RMSE
## ----------------------------------------------------------------------------------------------
##    1      MovingTime      addition       0.848       0.847    135.1810    989.3162    2.6759
##    2       AvgSpeed       addition       0.887       0.886     51.4060    930.9299    2.3152
##    3    AvgMovingSpeed    addition       0.908       0.907      6.6450    890.5558    2.0930
##    4    Take407All_Yes    addition       0.910       0.908      4.1740    888.0044    2.0751
## ----------------------------------------------------------------------------------------------
```

All three methods recommend a model with movingtime, avgspeed, avgmovingspeed, and take407all_yes, while they all excluded distance and maxspeed

```
model2 <- lm(TotalTime ~ AvgSpeed + AvgMovingSpeed + MovingTime + Take407All_Yes, data = Travel)
model2
```

```
##
## Call:
## lm(formula = TotalTime ~ AvgSpeed + AvgMovingSpeed + MovingTime +
##     Take407All_Yes, data = Travel)
##
## Coefficients:
##    (Intercept)        AvgSpeed  AvgMovingSpeed      MovingTime  Take407All_Yes
##        -9.2567         -0.2882          0.2713          1.3251          1.1861
```

```
pos <- ols_step_all_possible(model)
```

With n = 6, r^2 adjusted = 0.9078, and mallow's CP = 7 With n = 5, r^2 adjusted = 0.9082, and mallow's CP = 5.1570 With n = 4, r^2 adjusted = 0.9082, and mallow's CP = 4.1740 With n = 3, r^2 adjusted = 0.9066, and mallow's CP = 6.6455

The first and last models may have concerns with Mallow's CP, and Model n = 4 is ideal, with AvgSpeed AvgMovingSpeed MovingTime Take407All_Yes being the best model

```
ols_step_best_subset(model)
```

```
##                              Best Subsets Regression
## ---------------------------------------------------------------------------------
```

```
## Model Index    Predictors
## --------------------------------------------------------------------------------
##      1         MovingTime
##      2         AvgSpeed MovingTime
##      3         AvgSpeed AvgMovingSpeed MovingTime
##      4         AvgSpeed AvgMovingSpeed MovingTime Take407All_Yes
##      5         MaxSpeed AvgSpeed AvgMovingSpeed MovingTime Take407All_Yes
##      6         Distance MaxSpeed AvgSpeed AvgMovingSpeed MovingTime Take407All_Yes
## --------------------------------------------------------------------------------
##
##                                                    Subsets Regression Summary
## ----------------------------------------------------------------------------------------------------------
##                        Adj.        Pred
## Model    R-Square    R-Square    R-Square      C(p)        AIC         SBIC        SBC         MSEP
## ----------------------------------------------------------------------------------------------------------
##   1        0.8481      0.8474      0.8433    135.1814    989.3162    405.6860    999.2852    1467.92
##   2        0.8869      0.8858      0.8546     51.4060    930.9299    348.0113    944.2219    1098.82
##   3        0.9080      0.9066      0.7757      6.6455    890.5558    308.8441    907.1708     898.10
##   4        0.9100      0.9082      0.7772      4.1740    888.0044    306.5304    907.9425     882.79
##   5        0.9105      0.9082      0.7774      5.1570    888.9550    307.6022    912.2160     882.72
##   6        0.9105      0.9078      0.7688      7.0000    890.7925    309.5201    917.3766     886.50
## ----------------------------------------------------------------------------------------------------------
## AIC: Akaike Information Criteria
##  SBIC: Sawa's Bayesian Information Criteria
##  SBC: Schwarz Bayesian Criteria
##  MSEP: Estimated error of prediction, assuming multivariate normality
##  FPE: Final Prediction Error
##  HSP: Hocking's Sp
##  APC: Amemiya Prediction Criteria
```

The best model has the highest r^2 adjusted, while the Mallow's CP closest to n + 1, and the smallest AIC value. Thus model 4 is ideal.