NYCFlights: Arrival Delay Regression Model

Christopher Brown

R Sys.Date()

NYCFlights Model

Using the rectangular data that you created last week and following the *Predicting Medical Expenses* example from the text (*MLwR*), create a model for `arr_delay`. Follow *MLwR* structure for building a model. Describe/Explain each of the steps and show all work in codeblocks below.

Rubric

Points (10)

1 Provided knitted documentas HTML or PDF 1 Clearly laid out a notebook/Rmarkdown/Project Template 1 Clearly articulates the target/response value that is being modeled and specified a *Performance Metric* for evaluating the model 1 Suggested a Naive Model # Performed EDA on: 1 Specified response and ploted it 1 Specified predictors and ploted them 1 Checked/Tested for Co-linearity of predictors 1 Measured Performance of the Naive Model 1 Fit lm using lm function 1 Evaluated model performance using *Performance Metric*

1 Created an advanced feature

STEP 0: Read the data

```
YX <- readRDS("nycflights_joined.rds")
yx <- YX %>% sample_n(10000)
```

Step 1: Define the problem

This document provides a model(s) for `arr_delay` from the NYCFlights data set. The model(s) are regression models evaluated and compared by **RMSE**.

The most basic model that we can think of is the mean of the arr_delay. It is: NA

Our naive guess is: NA Our naive rmse is: NA

Step 2: Explore Data Analysis

The first thing that we should probably do is say something about the variables, both the response and the predictors ### Response;

- arr_delay : right skewed, some negative
- Hypothesis: flight delays are a function of weather, mechanical problems and other factors:

Predictors

This is reasonable list of variables that can be used

- fligths
- month **
- dep_delay (?) ***
- carrier / flight *
- origin

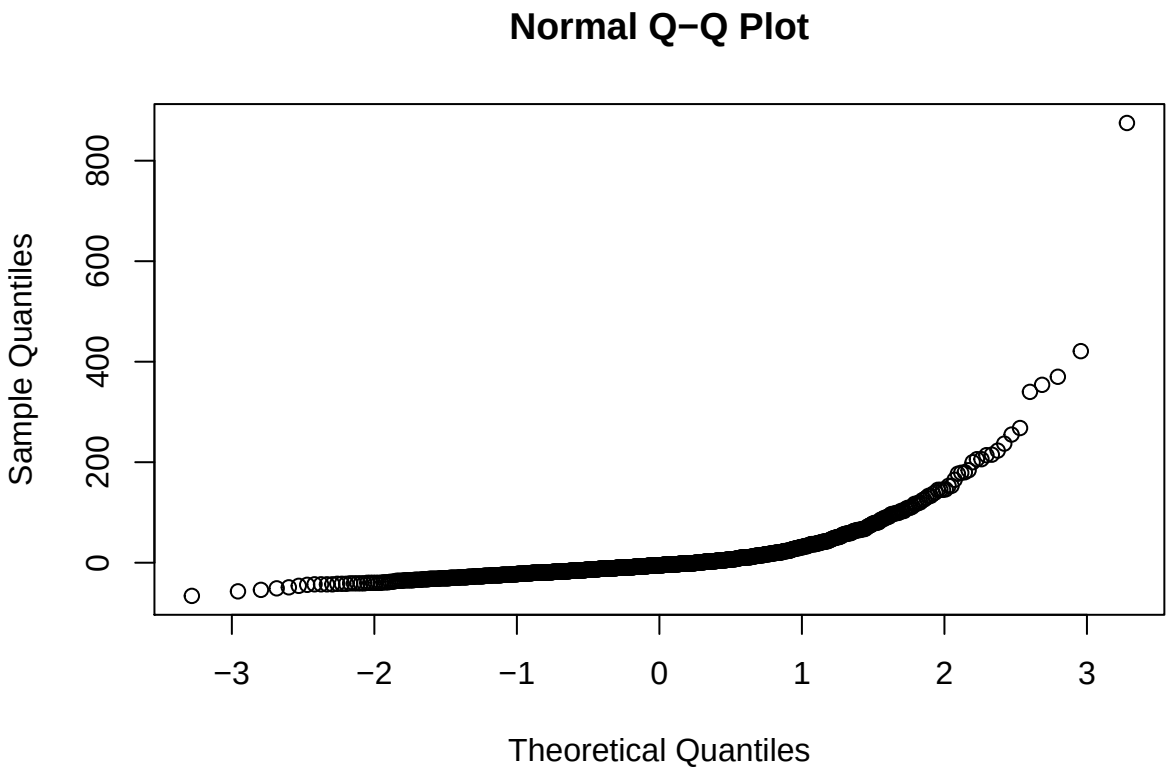
- dest
- air_time **
- distance ** (can make up time in the air)
- air time / hour / minute not known at end of flight
- planes
- year **
- type **
- manufacture
- engines (not much variation)
- seats
- engine **
- speed (not well populated)
- weather
- temp

- dewp
- wind_dir **+
- wind_gust **+
- precip **
- pressure **
- visib **
- airports (origin and dest)
- faa or name
- lat ** dest(only)
- lon ** dest(only)
- alt
- DERIVED:
- bearing relate to orgin and dest

- cos(bearing, wind_dir)

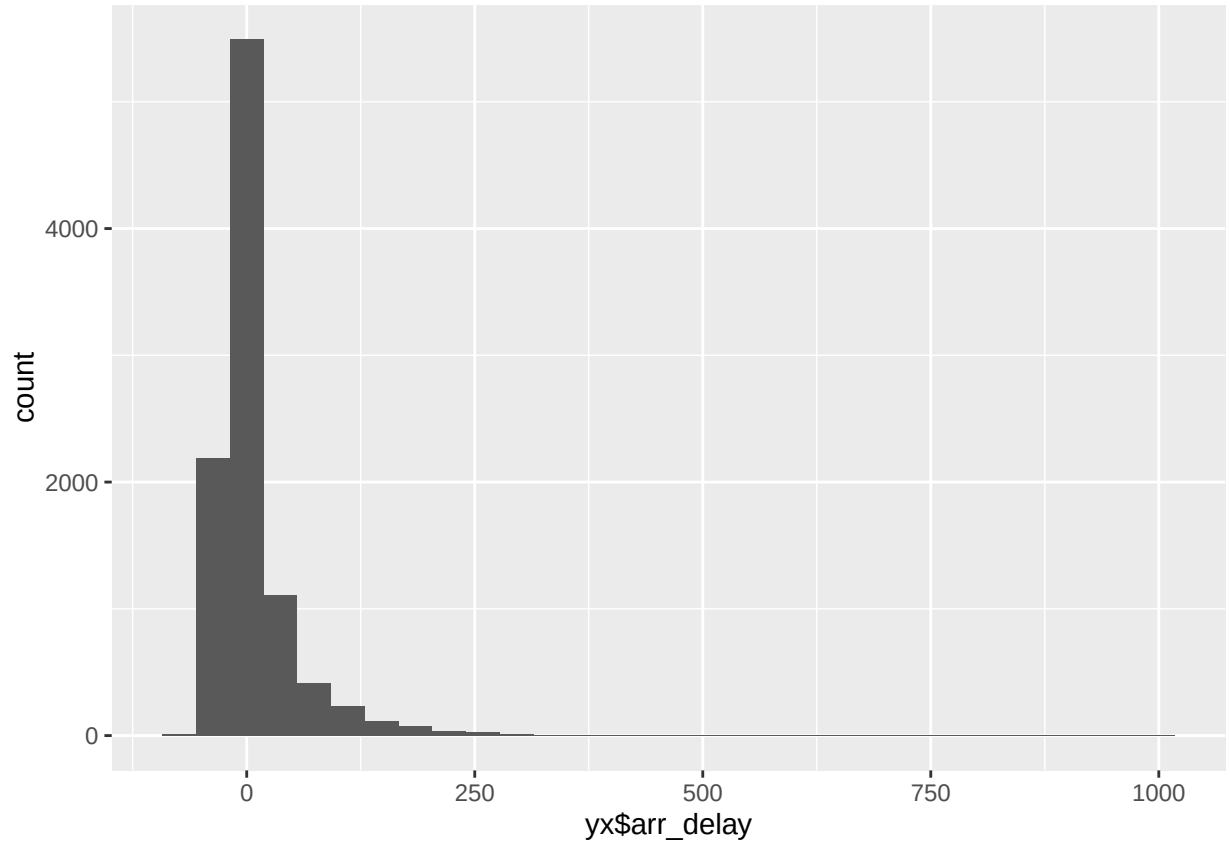
Exploration

```
y <- "arr_delay"
xs <- c(
   'month','dep_delay','carrier','air_time','distance'
  ,'year.pl','type','engine'
  ,'wind_dir','wind_speed','wind_gust','precip','pressure','visib'
  ,'lat','lon','lat.dest','lon.dest'
)


yx <- yx[ , c(y,xs), with=FALSE ]

qqnorm(yx  %>% sample_n(1000) %>% extract2('arr_delay') )
```



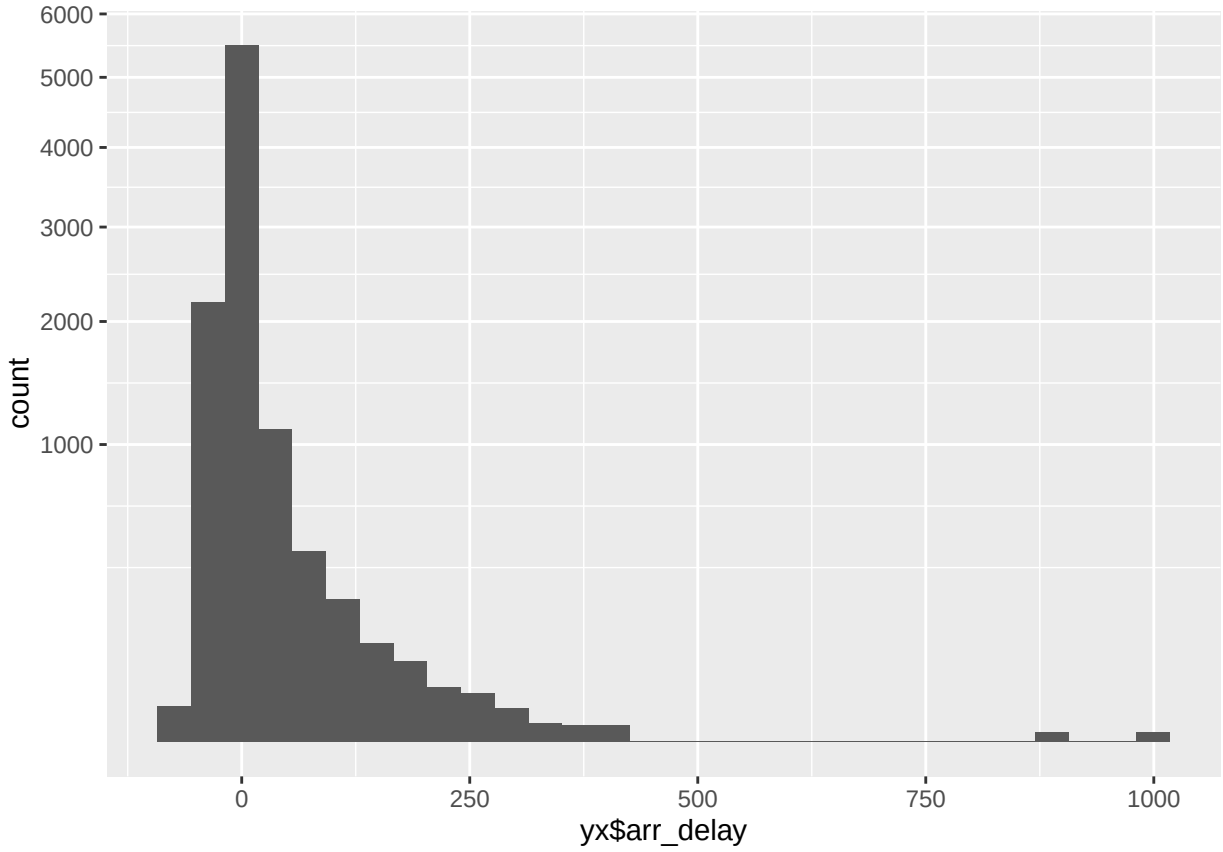

```
qplot(yx$arr_delay)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 286 rows containing non-finite values (stat_bin).
```



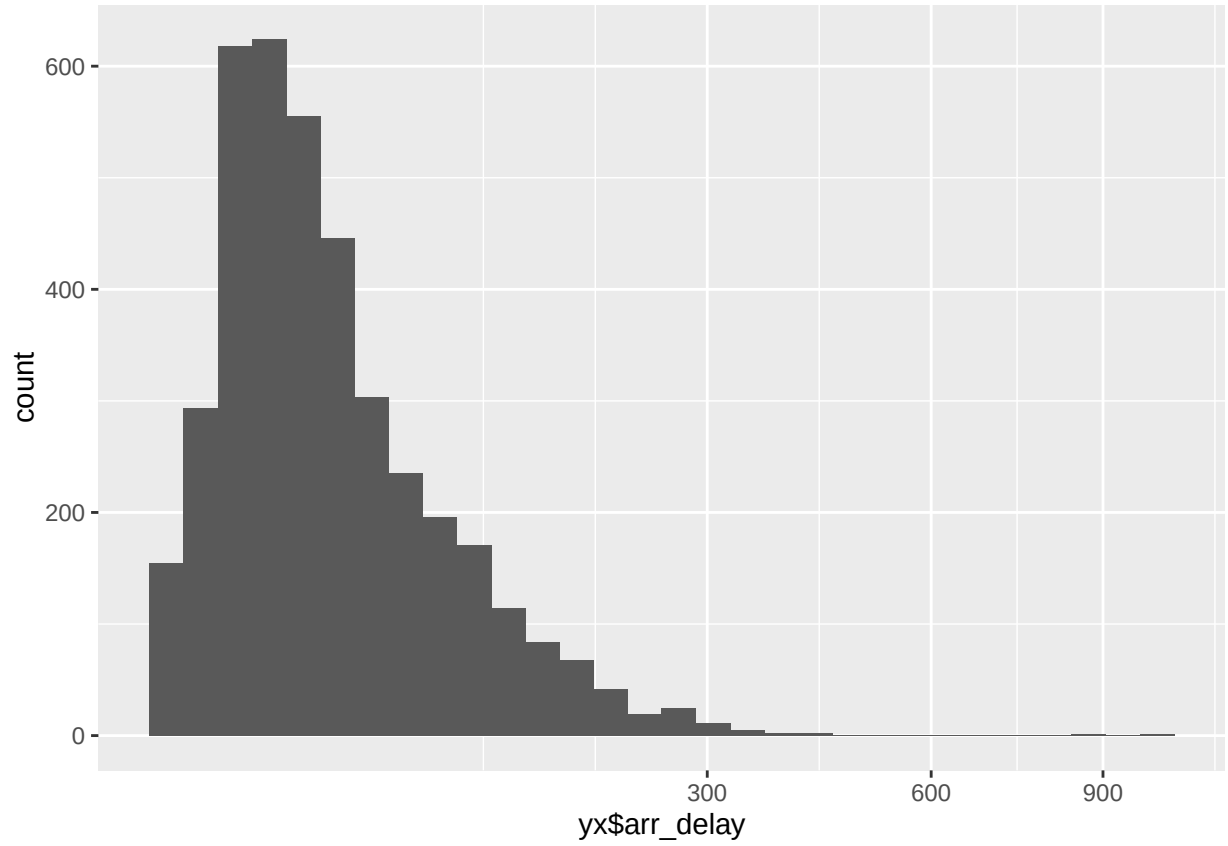

```
qplot(yx$arr_delay) + scale_y_sqrt()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 286 rows containing non-finite values (stat_bin).
```




```
qplot(yx$arr_delay) + scale_x_sqrt()
```

```
## Warning in self$trans$transform(x): NaNs produced
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 6029 rows containing non-finite values (stat_bin).
```




```
# Fix the response (arr_delay)
# Remove Outliers

 yx <- yx[ arr_delay >= quantile(arr_delay,0.01,na.rm=TRUE) &
           arr_delay <= quantile(arr_delay,0.96,na.rm=TRUE) ]

 # Consider: log, sqrt transformation (?)
 #  This will make the calculation of errors trickier
 yx[ , target := arr_delay %>% add( min(arr_delay, na.rm=TRUE ) ) %>% log  ]
```

```
## Warning in log(.): NaNs produced
```

```
##       arr_delay month dep_delay carrier air_time distance year.pl
##    1:        22     1         3      UA      123      733    2002
##    2:        17    12        27      EV      143      946    2003
##    3:        -5    11         2      HA      622     4983    2013
##    4:       -25     4        -7      UA      298     2133    1994
##    5:       -14    12        -7      B6      149     1005    2006
##   ---
## 9232:        11     9        15      MQ       67      431      NA
## 9233:        -6     9        -7      EV       41      209    1999
## 9234:        28     1        38      EV       49      277    2005
## 9235:       -32     5        -6      EV       75      529    2001
## 9236:        -6    12        -2      9E       78      427    2007
##                          type    engine wind_dir wind_speed wind_gust
##    1: Fixed wing multi engine Turbo-fan      310   17.26170  19.86440
##    2: Fixed wing multi engine Turbo-fan       20    4.60312   5.29718
```

```
##    3: Fixed wing multi engine Turbo-fan      330   16.11090  18.54010
##    4: Fixed wing multi engine Turbo-fan       NA    6.90468   7.94577
##    5: Fixed wing multi engine Turbo-fan      140    4.60312   5.29718
##   ---
## 9232:                       NA        NA       50   10.35700  11.91870
## 9233: Fixed wing multi engine Turbo-fan      200    8.05546   9.27006
## 9234: Fixed wing multi engine Turbo-fan      250   21.86480  25.16160
## 9235: Fixed wing multi engine Turbo-fan      210    4.60312   5.29718
## 9236: Fixed wing multi engine Turbo-fan      330   10.35700  11.91870
##       precip pressure visib     lat      lon lat.dest  lon.dest target
##    1:      0   1012.7    10 40.7772 -73.8726  41.9786  -87.9048    NaN
##    2:      0   1017.7     6 40.6925 -74.1687  35.0424  -89.9767    NaN
##    3:      0   1026.9    10 40.6398 -73.7789  21.3187 -157.9220    NaN
##    4:      0   1024.6    10 40.6925 -74.1687  33.4343 -112.0120    NaN
##    5:      0   1019.9    10 40.6398 -73.7789  27.9755  -82.5332    NaN
##   ---
## 9232:      0   1021.9    10 40.7772 -73.8726  35.8776  -78.7875    NaN
## 9233:      0   1010.9    10 40.6925 -74.1687  42.9326  -71.4357    NaN
## 9234:      0   1014.7    10 40.6925 -74.1687  37.5052  -77.3197    NaN
## 9235:      0   1023.7    10 40.6925 -74.1687  35.2140  -80.9431    NaN
## 9236:      0   1034.8    10 40.6398 -73.7789  35.8776  -78.7875    NaN
```

```
# Examine predictors

# COR
numerics <- yx   %>% sapply( is.numeric )  %>% which   %>% names()  # numerics
yx[ , numerics, with = FALSE ] %>% sample_n(1e3) %>% cor( use="pairwise.complete.obs" )
```

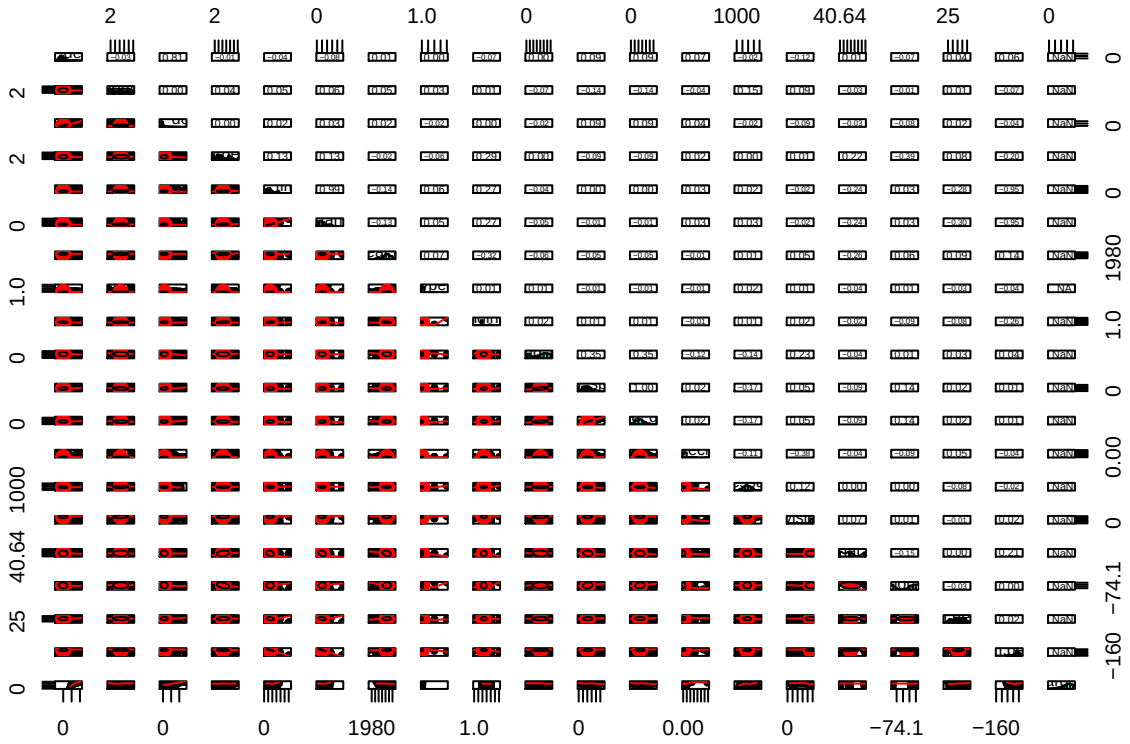
```
##               arr_delay       month    dep_delay     air_time     distance
## arr_delay   1.000000000 -0.03057793  0.797497251  0.042817049  0.001890901
## month      -0.030577930  1.00000000 -0.032008065  0.017677055  0.021188057
## dep_delay   0.797497251 -0.03200807  1.000000000  0.058577759  0.059804837
## air_time    0.042817049  0.01767705  0.058577759  1.000000000  0.990264518
## distance    0.001890901  0.02118806  0.059804837  0.990264518  1.000000000
## year.pl     0.025271878 -0.02164858  0.015631691 -0.110533706 -0.112401937
## wind_dir    0.011979949 -0.05173401  0.005204924 -0.008176723 -0.020960138
## wind_speed  0.027127389 -0.07943933  0.069551898  0.041985398  0.041749859
## wind_gust   0.027127345 -0.07943919  0.069551914  0.041985696  0.041750168
## precip      0.012161459 -0.02186717 -0.008590817  0.139151277  0.142011038
## pressure   -0.093818325  0.10194587 -0.050633536  0.048157106  0.047607236
## visib      -0.059259357  0.04381376 -0.008963412 -0.068882940 -0.066621544
## lat        -0.032770454  0.01360476 -0.065661106 -0.278629299 -0.289296923
## lon        -0.103876097 -0.02327998 -0.093902507  0.060495648  0.067242898
## lat.dest   -0.027306277  0.03978721 -0.031844851 -0.204876750 -0.222972984
## lon.dest   -0.002136802 -0.03141454 -0.058552559 -0.944723146 -0.945676431
## target              NaN         NaN          NaN          NaN          NaN
##                  year.pl      wind_dir  wind_speed   wind_gust
## arr_delay   0.0252718783  0.0119799490  0.02712739  0.02712734
## month      -0.0216485792 -0.0517340092 -0.07943933 -0.07943919
## dep_delay   0.0156316914  0.0052049237  0.06955190  0.06955191
## air_time   -0.1105337061 -0.0081767231  0.04198540  0.04198570
## distance   -0.1124019372 -0.0209601382  0.04174986  0.04175017
## year.pl     1.0000000000 -0.0007250522  0.01237424  0.01237405
## wind_dir   -0.0007250522  1.0000000000  0.39405508  0.39405537
## wind_speed  0.0123742437  0.3940550847  1.00000000  1.00000000
```

```
## wind_gust   0.0123740491  0.3940553741  1.00000000  1.00000000
## precip     -0.0386268076 -0.0849838857 -0.03198190 -0.03198196
## pressure    0.0043095356 -0.2470337514 -0.15067702 -0.15067743
## visib       0.0752112769  0.2436713724  0.06894920  0.06894906
## lat        -0.2330478955  0.0496893912 -0.05076295 -0.05076283
## lon         0.0344922214  0.0777277303  0.18356216  0.18356227
## lat.dest    0.1305279556  0.0141536463 -0.03381384 -0.03381400
## lon.dest    0.1010279672  0.0109757038 -0.02650110 -0.02650142
## target              NaN           NaN         NaN         NaN
##                  precip     pressure        visib          lat
## arr_delay   0.012161459 -0.093818325 -0.059259357 -0.032770454
## month      -0.021867173  0.101945871  0.043813764  0.013604761
## dep_delay  -0.008590817 -0.050633536 -0.008963412 -0.065661106
## air_time    0.139151277  0.048157106 -0.068882940 -0.278629299
## distance    0.142011038  0.047607236 -0.066621544 -0.289296923
## year.pl    -0.038626808  0.004309536  0.075211277 -0.233047895
## wind_dir   -0.084983886 -0.247033751  0.243671372  0.049689391
## wind_speed -0.031981902 -0.150677016  0.068949198 -0.050762955
## wind_gust  -0.031981962 -0.150677435  0.068949062 -0.050762829
## precip      1.000000000  0.008556841 -0.284297928 -0.034031179
## pressure    0.008556841  1.000000000  0.100849734 -0.033206568
## visib      -0.284297928  0.100849734  1.000000000  0.047433284
## lat        -0.034031179 -0.033206568  0.047433284  1.000000000
## lon        -0.038572609 -0.005152698 -0.012336407 -0.075533424
## lat.dest   -0.065898435 -0.040456796  0.042933996 -0.005498497
## lon.dest   -0.123694955 -0.028233642  0.057523446  0.254309894
## target              NaN           NaN         NaN         NaN
##                     lon     lat.dest     lon.dest target
## arr_delay  -0.103876097 -0.027306277 -0.002136802    NaN
## month      -0.023279980  0.039787209 -0.031414541    NaN
## dep_delay  -0.093902507 -0.031844851 -0.058552559    NaN
## air_time    0.060495648 -0.204876750 -0.944723146    NaN
## distance    0.067242898 -0.222972984 -0.945676431    NaN
## year.pl     0.034492221  0.130527956  0.101027967    NaN
## wind_dir    0.077727730  0.014153646  0.010975704    NaN
## wind_speed  0.183562156 -0.033813841 -0.026501099    NaN
## wind_gust   0.183562274 -0.033814001 -0.026501424    NaN
## precip     -0.038572609 -0.065898435 -0.123694955    NaN
## pressure   -0.005152698 -0.040456796 -0.028233642    NaN
## visib      -0.012336407  0.042933996  0.057523446    NaN
## lat        -0.075533424 -0.005498497  0.254309894    NaN
## lon         1.000000000  0.039296487 -0.041659837    NaN
## lat.dest    0.039296487  1.000000000 -0.053197507    NaN
## lon.dest   -0.041659837 -0.053197507  1.000000000    NaN
## target              NaN           NaN         NaN    NaN
```

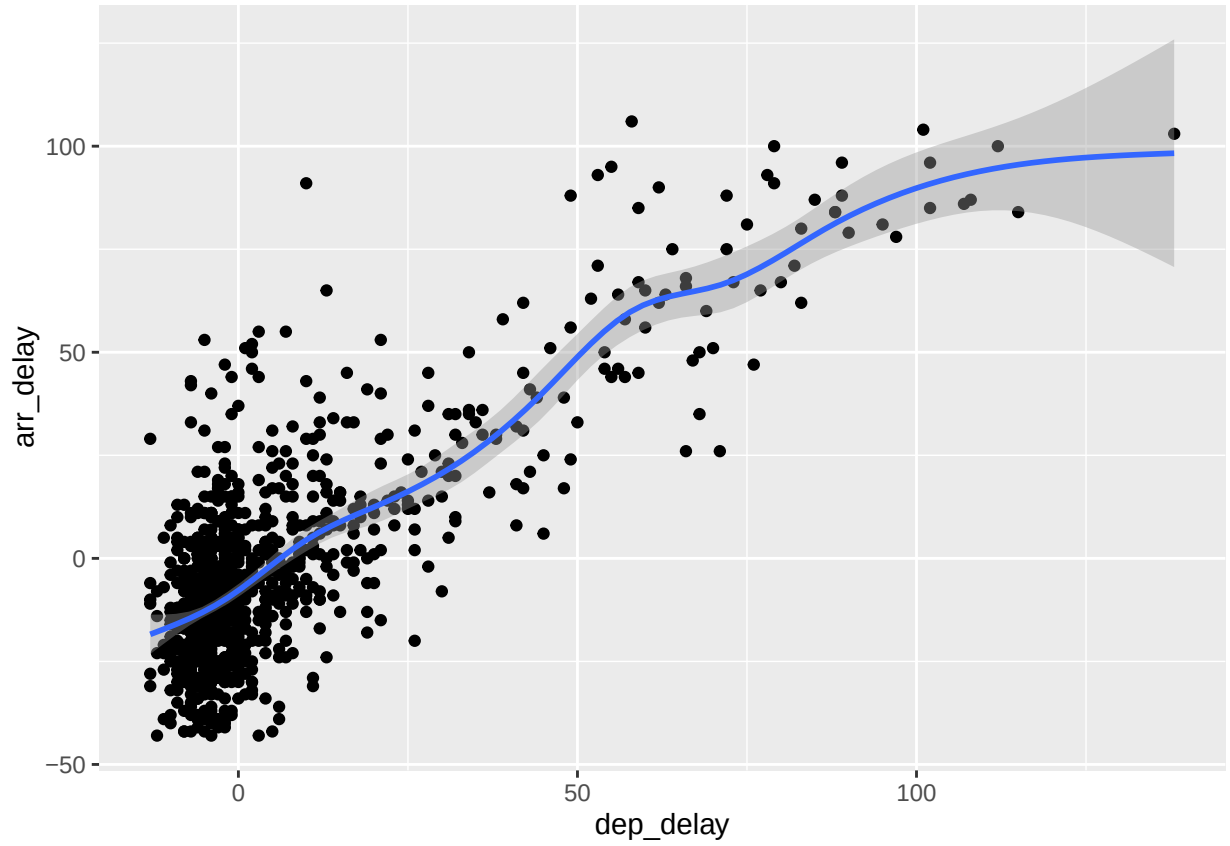
```
# pairs.panels
yx  %>% sample_n(1000)  %>% psych::pairs.panels()
```

```
## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero
```

```
## Warning in cor(x, y, use = "pairwise", method = method): the standard
## deviation is zero
```



```
# ggplot2
( yx  %>% sample_n(1000)  %>% .[ , qplot(y=arr_delay, x=dep_delay )] ) + geom_smooth() + coord_cartesi
```




```
# HANDLE NA VALUES
yx %>% sapply( . %>% is.na %>% sum )
```

```
##  arr_delay      month  dep_delay    carrier   air_time   distance
##          0          0          0          0          0          0
##    year.pl       type     engine   wind_dir wind_speed  wind_gust
##       1542       1389       1389        216         27         27
##     precip   pressure      visib        lat        lon   lat.dest
##         25        943         25          0          0        211
##   lon.dest     target
##        211       8489
```

FIX MISSING VALUES

There might be missing values; these are a drag on the model. Here we identify missing values.

```
yx  %>% sapply( .  %>% is.na  %>% sum )
```

```
##  arr_delay      month  dep_delay    carrier   air_time   distance
##          0          0          0          0          0          0
##    year.pl       type     engine   wind_dir wind_speed  wind_gust
##       1542       1389       1389        216         27         27
##     precip   pressure      visib        lat        lon   lat.dest
##         25        943         25          0          0        211
##   lon.dest     target
##        211       8489
```

Step 3: Train The Model

```
# START WITH A SMALL MODEL
set.seed(1234)
form <- arr_delay ~ dep_delay
fit <- lm( form, yx %>% sample_n(1000) )
fit$resid %>%  .^2 %>% mean %>% sqrt
```

```
## [1] 16.43438
```

```
set.seed(1234)
fit <- lm( arr_delay ~ dep_delay + month + air_time + distance + carrier + lat.dest + lon.dest, yx )
fit$resid %>%  .^2 %>% mean %>% sqrt
```

```
## [1] 13.72917
```

```
summary(fit)
```

```
##
## Call:
## lm(formula = arr_delay ~ dep_delay + month + air_time + distance +
##     carrier + lat.dest + lon.dest, data = yx)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.848  -8.731  -1.503   6.610 112.076
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.154899   3.327155  -3.052  0.00228 **
## dep_delay     0.992714   0.006614 150.088  < 2e-16 ***
## month         0.250770   0.042627   5.883 4.17e-09 ***
## air_time      0.705492   0.012093  58.340  < 2e-16 ***
## distance     -0.088353   0.001975 -44.734  < 2e-16 ***
## carrierAA     1.084227   0.827893   1.310  0.19036
## carrierAS     3.932326   3.918185   1.004  0.31559
## carrierB6     6.638739   0.779285   8.519  < 2e-16 ***
## carrierDL     3.676327   0.768744   4.782 1.76e-06 ***
## carrierEV     5.637177   0.724110   7.785 7.74e-15 ***
## carrierF9     9.581490   3.405844   2.813  0.00491 **
## carrierFL    12.116658   1.545961   7.838 5.11e-15 ***
## carrierHA    25.497411   4.799732   5.312 1.11e-07 ***
## carrierMQ    10.119406   0.806943  12.540  < 2e-16 ***
## carrierUA     1.683207   0.773821   2.175  0.02964 *
## carrierUS     8.634109   0.860793  10.030  < 2e-16 ***
## carrierVX     0.283816   1.382539   0.205  0.83735
## carrierWN    -0.279041   0.960133  -0.291  0.77134
## carrierYV     4.886428   3.728309   1.311  0.19002
## lat.dest     -0.145808   0.050307  -2.898  0.00376 **
## lon.dest      0.121507   0.066942   1.815  0.06954 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.75 on 9004 degrees of freedom
##   (211 observations deleted due to missingness)
```

```
## Multiple R-squared:  0.7423, Adjusted R-squared:  0.7417
## F-statistic:  1297 on 20 and 9004 DF,  p-value: < 2.2e-16
```

```
fit.step <- stepAIC(fit, scope=list(upper=. ~ ., lower = . ~ 1 ), trace = 2 )
```

```
## Start:  AIC=47324.39
## arr_delay ~ dep_delay + month + air_time + distance + carrier +
##     lat.dest + lon.dest
##
##             Df Sum of Sq     RSS   AIC
## <none>                   1701124 47324
## - lon.dest   1       622 1701747 47326
## - lat.dest   1      1587 1702711 47331
## - month      1      6539 1707663 47357
## - carrier   14     85699 1786823 47740
## - distance   1    378081 2079205 49134
## - air_time   1    643032 2344156 50216
## - dep_delay  1   4255934 5957058 58633
```

```
fit.step$resid %>%  .^2 %>% mean %>% sqrt
```

```
## [1] 13.72917
```

```
summary(fit)
```

```
##
## Call:
## lm(formula = arr_delay ~ dep_delay + month + air_time + distance +
##     carrier + lat.dest + lon.dest, data = yx)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.848  -8.731  -1.503   6.610 112.076
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.154899   3.327155  -3.052  0.00228 **
## dep_delay     0.992714   0.006614 150.088  < 2e-16 ***
## month         0.250770   0.042627   5.883 4.17e-09 ***
## air_time      0.705492   0.012093  58.340  < 2e-16 ***
## distance     -0.088353   0.001975 -44.734  < 2e-16 ***
## carrierAA     1.084227   0.827893   1.310  0.19036
## carrierAS     3.932326   3.918185   1.004  0.31559
## carrierB6     6.638739   0.779285   8.519  < 2e-16 ***
## carrierDL     3.676327   0.768744   4.782 1.76e-06 ***
## carrierEV     5.637177   0.724110   7.785 7.74e-15 ***
## carrierF9     9.581490   3.405844   2.813  0.00491 **
## carrierFL    12.116658   1.545961   7.838 5.11e-15 ***
## carrierHA    25.497411   4.799732   5.312 1.11e-07 ***
## carrierMQ    10.119406   0.806943  12.540  < 2e-16 ***
## carrierUA     1.683207   0.773821   2.175  0.02964 *
## carrierUS     8.634109   0.860793  10.030  < 2e-16 ***
## carrierVX     0.283816   1.382539   0.205  0.83735
## carrierWN    -0.279041   0.960133  -0.291  0.77134
## carrierYV     4.886428   3.728309   1.311  0.19002
## lat.dest     -0.145808   0.050307  -2.898  0.00376 **
```

```
## lon.dest      0.121507   0.066942   1.815  0.06954 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.75 on 9004 degrees of freedom
##   (211 observations deleted due to missingness)
## Multiple R-squared:  0.7423, Adjusted R-squared:  0.7417
## F-statistic:  1297 on 20 and 9004 DF,  p-value: < 2.2e-16
```

Step 4: Evaluate Performance

Evaluating performance is a crutial step, that has yet to be treated well by the MLwR. Think about some of the tools that you have for evaluating performance. Choose one and articulate why you have chosen it.

```
fit.step %>% resid %>% .^2 %>% mean %>% sqrt
```

```
## [1] 13.72917
```

Step 5: Improve Performance

Show some steps for improving model performance.

```
 # ....
```

Question:

Is this a good model? (Write your answer here.)

It is decent based on the little effort we applied.