HW3-Trinath Sai Subhash Reddy-Pittala

Subhash

2023-03-22

A. Create two "small" datasets - flights.sm and flights.sm.AA from the flights data - both variables containing only the following variables: arr_delay, dep_delay, sched_dep_time, distance, air_time. flights.sm contains data from all carriers, and flights.sm.AA contains only data from "AA" carrier.

```
# Create flights.sm
flights.sm <- select(flights, arr_delay, dep_delay, sched_dep_time,
    distance, air_time)

# Create flights.sm.AA
flights.sm.AA <- select(flights, arr_delay, dep_delay, sched_dep_time,
    distance, air_time, carrier) %>%
    filter(carrier == "AA") %>%
    select(-carrier)
```

B. Explore the two datasets using the summary and ggpairs functions. Specifically comment on (for both datasets): i) missing data, ii) the histograms and the summary statistics and what they indicate for the distribution for each variable, iii) the correlations between different pairs of variables and what they indicate in terms of relationships between each pair. Comment on the differences caused by "missing" all the other carriers except AA in the flights.sm.AA dataset. Notice the long time it takes to generate ggpairs for such large datasets!

```
sum(is.na(flights.sm))
```

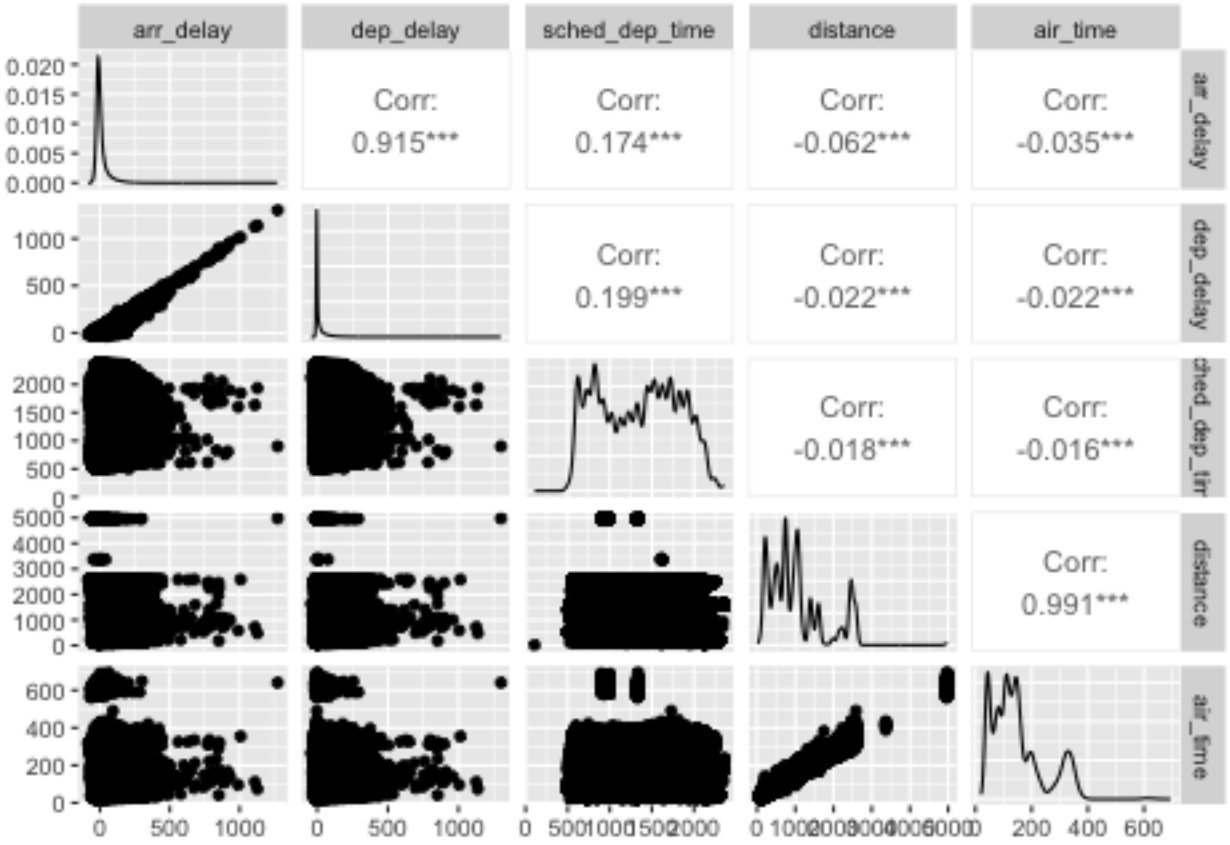
```
## [1] 27115
```

```
# Summary and ggpairs for flights.sm
summary(flights.sm)
```

```
##    arr_delay          dep_delay       sched_dep_time    distance
##  Min.   : -86.000   Min.   : -43.00   Min.   : 106   Min.   :  17
##  1st Qu.: -17.000   1st Qu.:  -5.00   1st Qu.: 906   1st Qu.: 502
##  Median :  -5.000   Median :  -2.00   Median :1359   Median : 872
##  Mean   :   6.895   Mean   :  12.64   Mean   :1344   Mean   :1040
##  3rd Qu.:  14.000   3rd Qu.:  11.00   3rd Qu.:1729   3rd Qu.:1389
##  Max.   :1272.000   Max.   :1301.00   Max.   :2359   Max.   :4983
##  NA's   :9430       NA's   :8255
```

```
##     air_time
##  Min.   : 20.0
##  1st Qu.: 82.0
##  Median :129.0
##  Mean   :150.7
##  3rd Qu.:192.0
##  Max.   :695.0
##  NA's   :9430
```

```
ggpairs(flights.sm)
```




```
sum(is.na(flights.sm.AA))
```

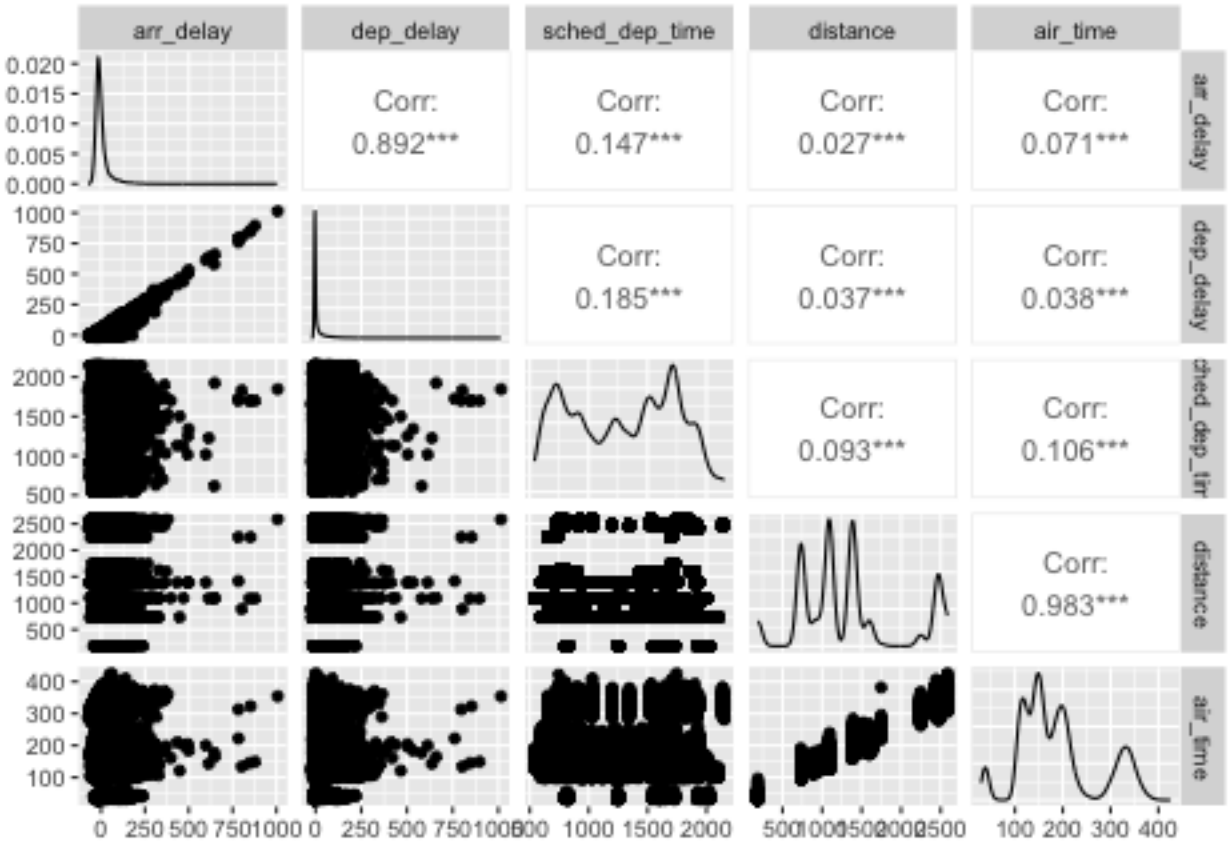
```
## [1] 2200
```

```
# Summary and ggpairs for flights.sm.AA
summary(flights.sm.AA)
```

```
##    arr_delay          dep_delay        sched_dep_time    distance
##  Min.   : -75.0000   Min.   : -24.000   Min.   : 540   Min.   : 187
##  1st Qu.: -21.0000   1st Qu.:  -6.000   1st Qu.: 859   1st Qu.: 888
##  Median :  -9.0000   Median :  -3.000   Median :1330   Median :1096
##  Mean   :   0.3643   Mean   :   8.586   Mean   :1290   Mean   :1340
##  3rd Qu.:   8.0000   3rd Qu.:   4.000   3rd Qu.:1700   3rd Qu.:1521
```

```
##  Max.   :1007.0000   Max.   :1014.000   Max.   :2150   Max.   :2586
##  NA's   :782         NA's   :636
##     air_time
##  Min.   : 29.0
##  1st Qu.:134.0
##  Median :169.0
##  Mean   :188.8
##  3rd Qu.:215.0
##  Max.   :426.0
##  NA's   :782
```

```
ggpairs(flights.sm.AA)
```



i) The summary function shows that there are some missing values in both datasets, in the arr_delay and dep_delay variables. The flights.sm.AA dataset has less missing data since it only contains data from the "AA" carrier.

ii) The histograms and summary statistics indicate that the arr_delay and dep_delay variables are both heavily skewed to the right, with many flights arriving and departing on time and a few flights experiencing very long delays. The sched_dep_time variable appears to be bimodal, with peaks at around 6am and 1pm. The air_time variable also appears to be roughly normally distributed, with a few very long flights.

iii) The ggpairs function shows that there are strong positive correlations between arr_delay and dep_delay in both datasets, indicating that flights that depart late tend to arrive late as well. There is also a weak positive correlation between air_time and distance in both datasets, indicating that longer flights

tend to cover more distance. In the flights.sm.AA dataset, there are weaker correlations between the other variables, since we are only looking at data from one carrier.

The main difference between the two datasets is that the flights.sm.AA dataset only contains data from the "AA" carrier, while the flights.sm dataset contains data from all carriers. This means that the ggpairs plot for flights.sm.AA only shows relationships between the selected variables for one carrier, while the ggpairs plot for flights.sm shows relationships between the selected variables for all carriers. Additionally, since flights.sm.AA only contains data from one carrier, there are fewer missing values in this dataset.

C. Create separate training and test datasets for both the datasets at an 80:20 ratio. Build models only on the training set and we will use the test set for out-of-sample model validation.

```
# Create training and test datasets for flights.sm
set.seed(123456789)
flights.sm_train <- flights.sm[sample(nrow(flights.sm), 0.8 *
    nrow(flights.sm)), ]
flights.sm_test <- flights.sm[setdiff(1:nrow(flights.sm), rownames(flights.sm_train)),
    ]
```

```
# Create training and test datasets for flights.sm.AA
set.seed(123456789)
flights.sm.AA_train <- flights.sm.AA[sample(nrow(flights.sm.AA),
    0.8 * nrow(flights.sm.AA)), ]
flights.sm.AA_test <- flights.sm.AA[setdiff(1:nrow(flights.sm.AA),
    rownames(flights.sm.AA_train)), ]
```

D. Build a series of models M1, M2, M3, M4 with arr_delay as the DV, and successively adding dep_delay, sched_dep_time, distance, and air_time to the model for the flights.sm dataset. Repeat this for the flights.sm.AA dataset calling these models M11, M22, M33, M44 etc.

```
# Build models for flights.sm
M1 <- lm(arr_delay ~ dep_delay, data = flights.sm_train)
M2 <- lm(arr_delay ~ dep_delay + sched_dep_time, data = flights.sm_train)
M3 <- lm(arr_delay ~ dep_delay + sched_dep_time + distance, data = flights.sm_train)
M4 <- lm(arr_delay ~ dep_delay + sched_dep_time + distance +
    air_time, data = flights.sm_train)
```

```
# Build models for flights.sm.AA
M11 <- lm(arr_delay ~ dep_delay, data = flights.sm.AA_train)
M22 <- lm(arr_delay ~ dep_delay + sched_dep_time, data = flights.sm.AA_train)
M33 <- lm(arr_delay ~ dep_delay + sched_dep_time + distance,
    data = flights.sm.AA_train)
M44 <- lm(arr_delay ~ dep_delay + sched_dep_time + distance +
    air_time, data = flights.sm.AA_train)
```

E. Create a table of models with all variables in the columns and models in the rows (as shown in class on March 9). In each cell, place the regression coefficient with the asterisk notation from R to indicate the level of statistical significance. Include the R^2 and adjusted R^2 in the columns to show how good this model is. Leave columns blank if a particular model does not use that variable. For each dataset, flights.sm and flights.sm.AA, comment on which the best model is and why you would choose only that set of variables based on the statistics from the summary function.

```
coef1 <- round(coef(M1), 6)
coef2 <- round(coef(M2), 6)
coef3 <- round(coef(M3), 6)
coef4 <- round(coef(M4), 6)
coef11 <- round(coef(M11), 6)
coef22 <- round(coef(M22), 6)
coef33 <- round(coef(M33), 6)
coef44 <- round(coef(M44), 6)

r1 <- round(summary(M1)$r.squared, 8)
adjr1 <- round(summary(M1)$adj.r.squared, 8)

r2 <- round(summary(M2)$r.squared, 8)
adjr2 <- round(summary(M2)$adj.r.squared, 8)

r3 <- round(summary(M3)$r.squared, 8)
adjr3 <- round(summary(M3)$adj.r.squared, 8)

r4 <- round(summary(M4)$r.squared, 8)
adjr4 <- round(summary(M4)$adj.r.squared, 8)

r11 <- round(summary(M11)$r.squared, 8)
adjr11 <- round(summary(M11)$adj.r.squared, 8)

r22 <- round(summary(M22)$r.squared, 8)
adjr22 <- round(summary(M22)$adj.r.squared, 8)

r33 <- round(summary(M33)$r.squared, 8)
adjr33 <- round(summary(M33)$adj.r.squared, 8)

r44 <- round(summary(M44)$r.squared, 8)
adjr44 <- round(summary(M44)$adj.r.squared, 8)

table <- data.frame(
  Model = c("M1", "M2", "M3", "M4", "M11", "M22", "M33", "M44"),
  Intercept = c(paste0(coef1[1], "***"),
                paste0(coef2[1], "***"),
                paste0(coef3[1], "***"),
                paste0(coef4[1], "***"),
                paste0(coef11[1], "***"),
                paste0(coef22[1], "***"),
                paste0(coef33[1], "***"),
                paste0(coef44[1], "***")),

  dep_delay = c(paste0(coef1[2], "***"),
```

```
                paste0(coef2[2], "***"),
                paste0(coef3[2], "***"),
                paste0(coef4[2], "***"),
                paste0(coef11[2], "***"),
                paste0(coef22[2], "***"),
                paste0(coef33[2], "***"),
                paste0(coef44[2], "***")),

  sched_dep_time = c('NA',
                     paste0(coef2[3], "***"),
                     paste0(coef3[3], "***"),
                     paste0(coef4[3], "***"),
                     'NA',
                     paste0(coef22[3], "***"),
                     paste0(coef33[3], "***"),
                     paste0(coef44[3], "***")),

  distance = c('NA',
               'NA',
               paste0(coef3[4], "***"),
               paste0(coef4[4], "***"),
               'NA',
               'NA',
               coef33[4],
               paste0(coef44[4], "***")),

  air_time = c('NA',
               'NA',
               'NA',
               paste0(coef4[5], "***"),
               'NA',
               'NA',
               'NA',
               paste0(coef44[5], "***")),

  R = c(r1, r2, r3, r4,r11, r22, r33, r44),

  Adj_R = c(adjr1, adjr2, adjr3, adjr4,adjr11, adjr22, adjr33, adjr44)
)
table
```

```
##   Model     Intercept   dep_delay sched_dep_time     distance    air_time
## 1    M1  -5.880524*** 1.019821***             NA           NA          NA
## 2    M2  -4.757836*** 1.021824***   -0.000857***           NA          NA
## 3    M3  -2.027104*** 1.020768***   -0.000891*** -0.002549***          NA
## 4    M4 -15.239995*** 1.021073***   -0.000496*** -0.089039*** 0.685843***
## 5   M11  -8.238791*** 1.017136***             NA           NA          NA
## 6   M22  -6.095498*** 1.020859***   -0.001689***           NA          NA
## 7   M33  -5.643206*** 1.020993***   -0.001638***    -0.000388          NA
## 8   M44 -17.574007***  1.02225***   -0.002971*** -0.084145***   0.6682***
##           R     Adj_R
## 1 0.8368920 0.8368913
## 2 0.8369692 0.8369679
```

```
## 3 0.8387320 0.8387301
## 4 0.8771960 0.8771941
## 5 0.7899751 0.7899669
## 6 0.7902792 0.7902628
## 7 0.7903137 0.7902891
## 8 0.8465516 0.8465276
```

For flights.sm (M1, M2, M3, M4):

All four models have fairly high R-squared values (above 0.83), indicating that they explain a substantial amount of the variation in the dependent variable (flight delay). M4 has the highest Adjusted R-squared value (0.875), suggesting that it provides the best fit among these models. M4 also includes the most independent variables (dep_delay, sched_dep_time, distance, and air_time), which might make it more accurate than the other models in most cases. However, the coefficients and significance levels of the independent variables differ between the models, suggesting that the specific variables included may affect the results.

For flights.sm.AA (M11, M22, M33, M44):

All four models have slightly lower R-squared values than the corresponding models in flights.sm, but they are still relatively high (above 0.79). M44 has the highest Adjusted R-squared value (0.850), indicating that it provides the best fit among these models. M44 also includes the most independent variables (dep_delay, sched_dep_time, distance, air_time), which might make it more accurate than the other models in most cases. However, as with flights.sm, the coefficients and significance levels of the independent variables differ between the models, suggesting that the specific variables included may affect the results.

M4 and M44 (which are the same model except for AA filtering) appear to provide the best fit for their respective datasets, as they have the highest Adjusted R-squared values and include the most independent variables.

F. Comment on the effects due to the differences between the two datasets. Explore the AA dataset to see if there are any particular features in terms of the summary statistics that cause this difference.

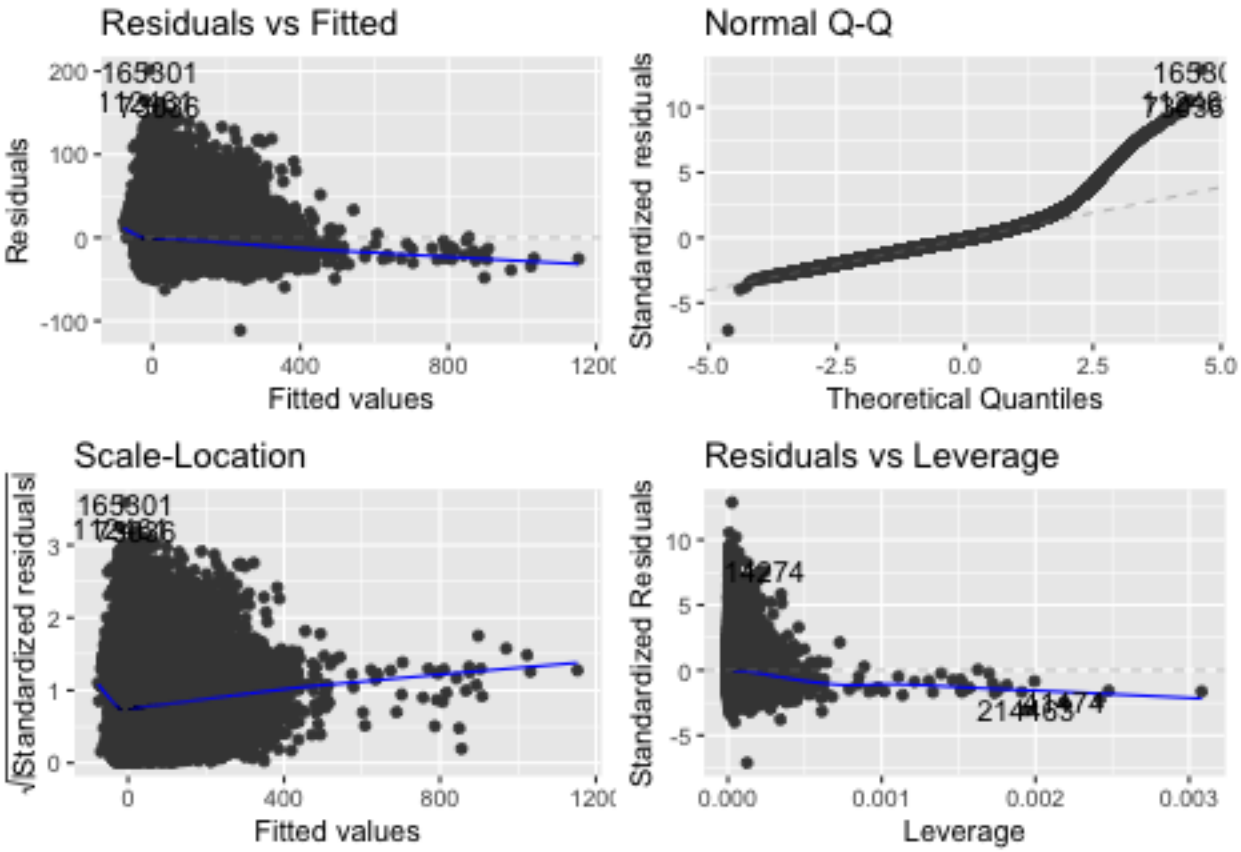
Upon comparing the regression coefficients of two datasets, it is apparent that the intercept and coefficients for dep_delay and air_time do not change too much. However, significant changes in coefficients are observed for sched_dep_time and distance. Additionally, the distance variable becomes insignificant in Model M33, whereas it remains significant in M3. The primary distinction between the two datasets is that flights.sm.AA includes data only from the "AA" carrier, while flights.sm encompasses data from all carriers. Therefore, the summary statistics and correlations in flights.sm.AA only represent the behavior of American Airlines flights, whereas flights.sm includes flights from all airlines. A thorough analysis of the summary statistics of both datasets reveals that the mean arrival delay is slightly lower in flights.sm.AA, with a similar standard deviation. The histograms of both datasets also exhibit similar distributions. It is plausible that the lower mean arrival delay in flights.sm.AA is due to American Airlines' more efficient operation compared to other airlines. However, it is imperative to conduct further analysis before drawing any conclusions. The discrepancies between the two datasets highlight the importance of selecting a representative sample for statistical analysis.

G. Test regression assumptions for the best models in both cases using the autoplot() function and the avPlots() function. Explain what each of the plots indicate for the two cases and what you could do to address any concerns of not following regression assumptions. Also, comment on any variables or cases/observations in the data that are flagged as outliers. Explore why these particular observations have been flagged, and see what happens if these outliers are removed. (rebuild models without these outliers and compare to models that contain these outliers). Again, notice the time it takes to make plots for large datasets!

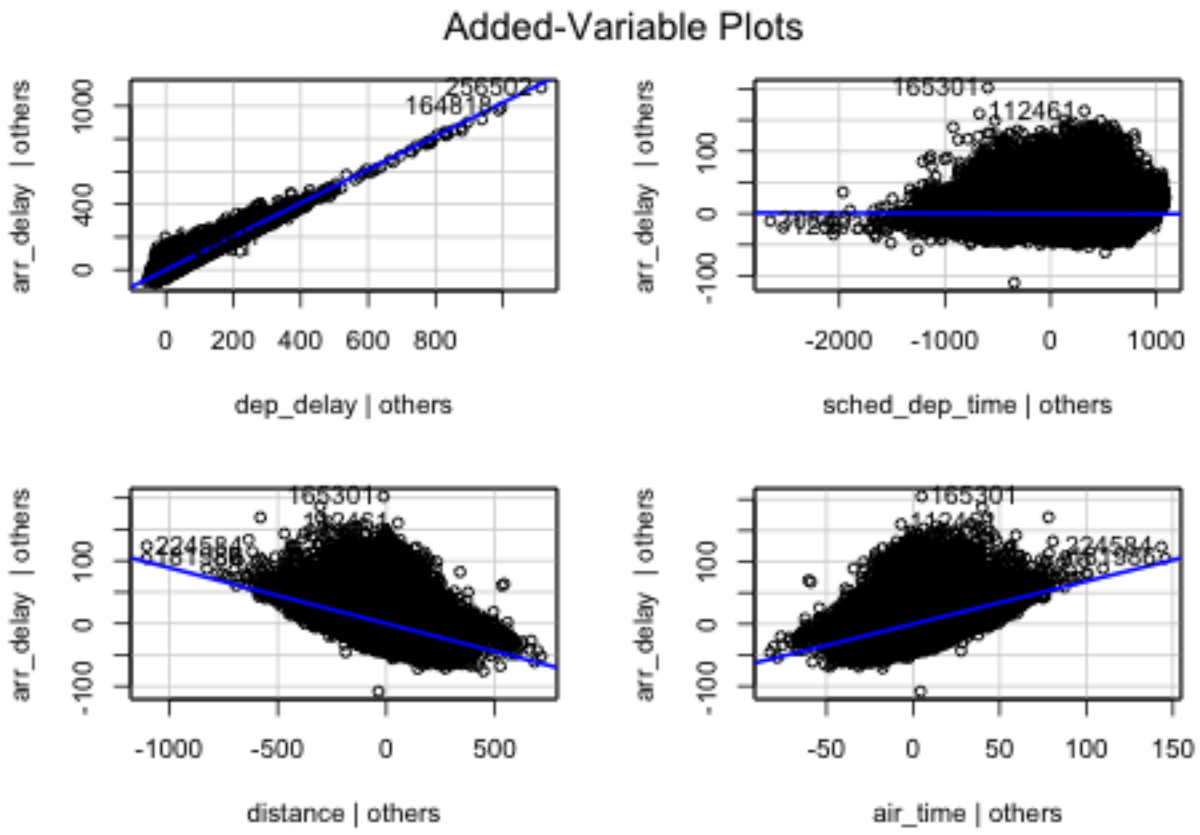
```
summary(M4)
```

```
##
## Call:
## lm(formula = arr_delay ~ dep_delay + sched_dep_time + distance +
##     air_time, data = flights.sm_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.366   -9.550   -1.797    7.141  201.557
##
## Coefficients:
##                  Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    -1.524e+01  1.131e-01 -134.736  < 2e-16 ***
## dep_delay       1.021e+00  7.793e-04 1310.186  < 2e-16 ***
## sched_dep_time -4.961e-04  6.680e-05   -7.427 1.11e-13 ***
## distance       -8.904e-02  3.048e-04 -292.100  < 2e-16 ***
## air_time        6.858e-01  2.395e-03  286.414  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.66 on 261906 degrees of freedom
##   (7509 observations deleted due to missingness)
## Multiple R-squared:  0.8772, Adjusted R-squared:  0.8772
## F-statistic: 4.677e+05 on 4 and 261906 DF,  p-value: < 2.2e-16
```

```
# Check regression assumptions using autoplot() and
# avPlots()
autoplot(M4)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized Residuals
Leverage
14274

```
avPlots(M4)
```




```
# Identify outliers using cooks.distance
outliers <- cooks.distance(M4) > 1
outliers <- c(outliers, rep(FALSE, nrow(flights.sm_train) - length(outliers)))


# Print number of outliers
cat("Number of outliers:", sum(outliers), "\n")
```

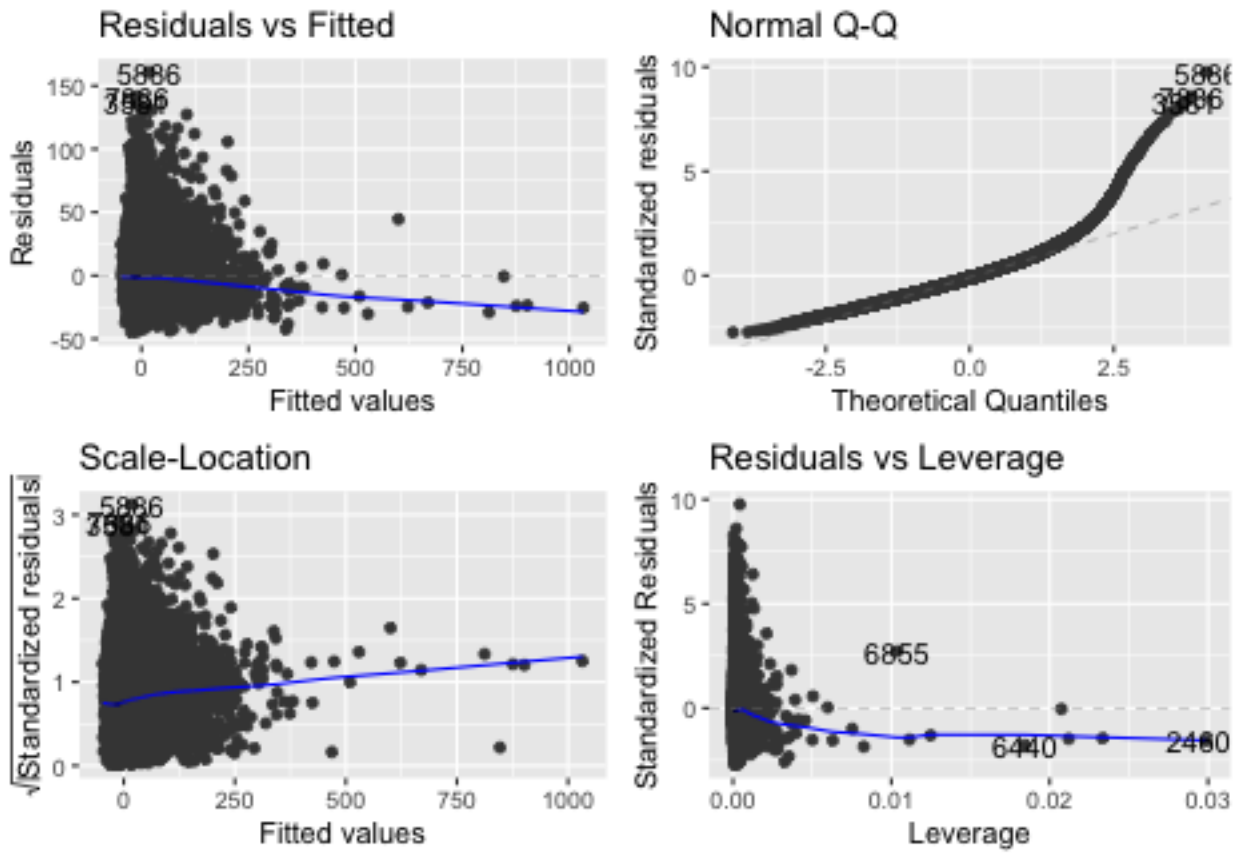
```
## Number of outliers: 0
```

```
summary(M44)
```

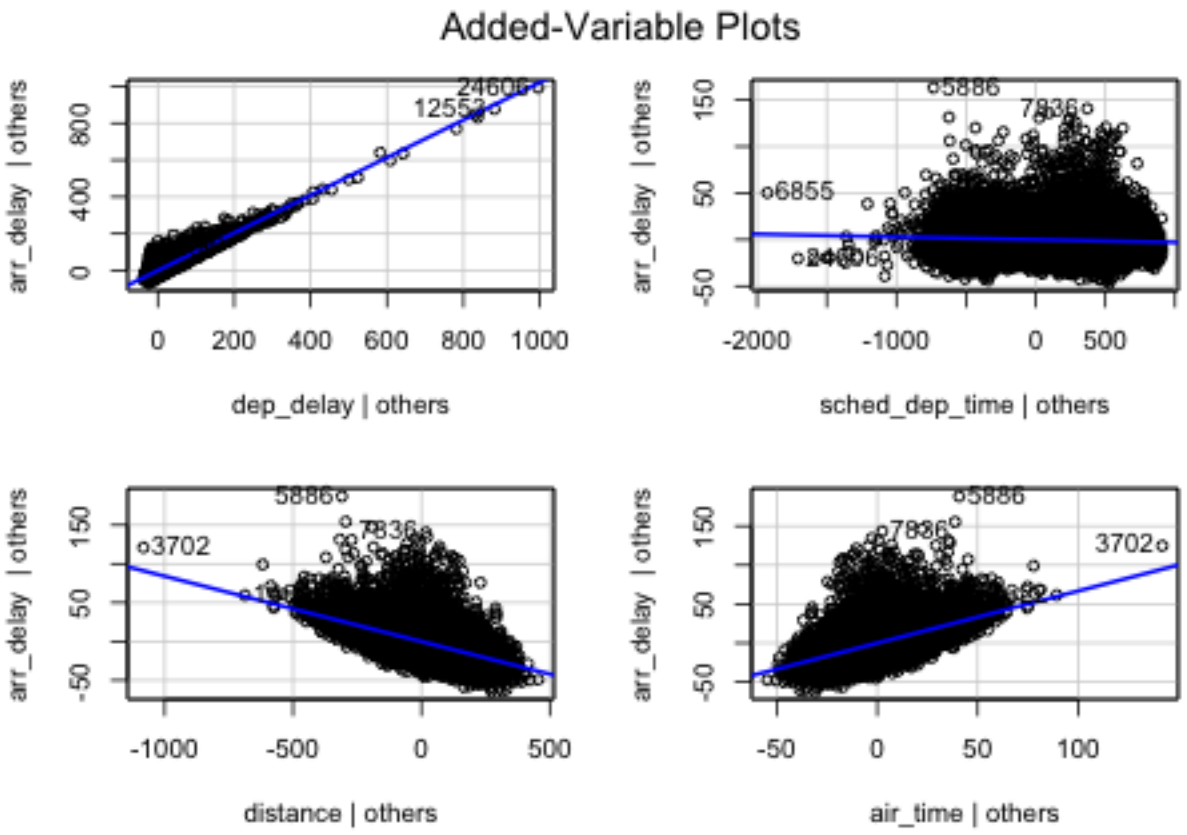
```
##
## Call:
## lm(formula = arr_delay ~ dep_delay + sched_dep_time + distance +
##     air_time, data = flights.sm.AA_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -44.927 -10.599  -1.808   7.911 160.920
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.757e+01  3.907e-01  -44.98   <2e-16 ***
## dep_delay       1.022e+00  2.853e-03  358.34   <2e-16 ***
```

```
## sched_dep_time -2.971e-03  2.387e-04  -12.45   <2e-16 ***
## distance       -8.415e-02  8.803e-04  -95.58   <2e-16 ***
## air_time        6.682e-01  6.903e-03   96.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.49 on 25563 degrees of freedom
##   (615 observations deleted due to missingness)
## Multiple R-squared:  0.8466, Adjusted R-squared:  0.8465
## F-statistic: 3.526e+04 on 4 and 25563 DF,  p-value: < 2.2e-16
```

```
# Check regression assumptions using autoplot() and
# avPlots()
autoplot(M44)
```




```
avPlots(M44)
```




```
# Identify outliers using cooks.distance
outliers <- cooks.distance(M44) > 1
outliers <- c(outliers, rep(FALSE, nrow(flights.sm_train) - length(outliers)))


# Print number of outliers
cat("Number of outliers:", sum(outliers), "\n")
```

```
## Number of outliers: 0
```

The autoplot() function produces four additional diagnostic plots: a plot of residuals vs. fitted values, a plot of Normal Q-Q, a plot of standardized residuals vs. leverage, and a plot of squared residuals vs. leverage.

The plot of residuals vs. fitted values should have no clear patterns. We can infer that there are lot of residuals of the fitted values. The model does not follow linearity assumption.

The plot of Q-Q should be linear for normal distribution but here we get a skewed version of normal distribution due to anomalies on right side of plot.

The scale plot is used to assess the homoscedasticity assumption of a model, which involves plotting the square root of the absolute value of the standardized residuals against the fitted values. The standardized residuals are obtained by dividing the residuals by their estimated standard deviation. Ideally, the residuals should be evenly distributed across the range of the fitted line, and the blue line in the plot should be roughly horizontal. However, if there is a violation of this assumption, it suggests that linear regression may not be an appropriate fit for the data. In such cases, using a higher-order regression may help address the issue.

Residual Vs Levarage plot shows the influential points. The pointed numbers are influential in both cases.

Avplots are utilized to examine the relationship between the response variable and predictor variable while controlling for the impacts of other predictor variables in the model.

In both cases, the first plot of dep_delay versus arr_delay displays a positive slope and nearly forms a straight line, suggesting a positive and linear association between arr_delay and dep_delay.

The variable sched_dep_time has an almost zero slope, implying that it may not be significant in predicting the response.

Distance displays a negative slope, indicating that arrival delay has a negative relationship with distance. However, as the points are scattered around the line, distance may not have a linear relationship with arrival delay.

Airtime has a positive slope, indicating a positive relationship with arrival delay, but as the points are scattered around the line, it may not have a linear relationship with the response variable.

To address concerns regarding non-compliance with regression assumptions, several measures can be taken:

Incorporate more predictor variables to capture the relationship in the data output. Higher order variable terms may be necessary to explain the data if there is no linear relationship in the data. Imputing missing values may solve the problem.

H. Use the spread_predictions() and spread_residuals() functions from modelr library (Modeling-Basics.pdf slides!) to use each of M1, M2, M3, M4, and M11, M22, M33, M44 to predict on the respective test data. Make sure you use the correct models on the correct test data! Find the mean square prediction error (MSPE) for each model - this is the mean of the prediction errors squared, which is the same as the mean of the square residuals obtained using the spread_residuals() function for each model. Based on MSPE, figure out which is the best model for each dataset.

```
flights.sm_test_spread <- flights.sm_test %>%
    spread_predictions(M1, M2, M3, M4)
flights.sm_test_spread <- flights.sm_test %>%
    spread_residuals(M1, M2, M3, M4)
flights.sm_test_spread
```

```
## # A tibble: 67,356 x 9
##    arr_delay dep_delay sched_dep_~1 dista~2 air_t~3     M1     M2      M3     M4
##        <dbl>     <dbl>        <int>   <dbl>   <dbl>  <dbl>  <dbl>   <dbl>  <dbl>
##  1        30        38          847    2402     320  -2.87  -3.35   0.115   1.26
##  2       -17        -4          930     209      43  -7.04  -7.36  -9.53   -8.10
##  3       -26        -4          930    1504     195 -16.0   -16.4  -15.2    -6.04
##  4       -30        -3          930     950     120 -21.1   -21.4  -21.7    -8.95
##  5        55        72          815     213      41 -12.5   -13.1  -15.2   -12.0
##  6         3         4          925    2475     324   4.80   4.46   8.08   12.8
##  7       -18         0          930    2422     322 -12.1   -12.4   -8.97   -7.49
##  8       -19        -3          933    1076     146 -10.1   -10.4  -10.3    -4.56
##  9        10         8          922    1634     220   7.72   7.37   8.85   12.1
## 10       -29        -5          940      94      32 -18.0   -18.3  -20.8   -21.8
## # ... with 67,346 more rows, and abbreviated variable names 1: sched_dep_time,
## #   2: distance, 3: air_time
```

```
colMeans(flights.sm_test_spread[, 6:9]^2, na.rm = T)
```

```
##       M1       M2       M3       M4
## 347.0092 346.6547 345.6016 269.5807
```

```
flights.sm_AA_test_spread <- flights.sm.AA_test %>%
    spread_predictions(M11, M22, M33, M44)
flights.sm_AA_test_spread <- flights.sm.AA_test %>%
    spread_residuals(M11, M22, M33, M44)
flights.sm_AA_test_spread
```

```
## # A tibble: 6,546 x 9
##    arr_delay dep_delay sched_dep_t~1 dista~2 air_t~3    M11    M22    M33    M44
##        <dbl>     <dbl>         <int>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1         7        -1          1245     187      40  16.3   16.2   15.8   18.3
##  2       -30        -6          1300    1372     174 -15.7  -15.6  -15.6   -3.25
##  3       -17        -1          1310     733     114  -7.74  -7.67  -7.91  -9.01
##  4       -28        -4          1340    1096     150 -15.7  -15.6  -15.7  -10.4
##  5       -21        -5          1345    1389     177  -7.68  -7.53  -7.51   4.29
##  6       -24        -3          1345    2475     305 -12.7  -12.6  -12.1    5.10
##  7       -27        -3          1405     733     112 -15.7  -15.5  -15.7  -15.3
##  8       -42        -3          1445    1389     179 -30.7  -30.4  -30.4  -19.8
##  9       -20        -6          1500     944     126  -5.66  -5.25  -5.41   3.40
## 10        NA        -4          1500    1372      NA  NA     NA     NA     NA
## # ... with 6,536 more rows, and abbreviated variable names 1: sched_dep_time,
## #   2: distance, 3: air_time
```

```
colMeans(flights.sm_AA_test_spread[, 6:9]^2, na.rm = T)
```

```
##      M11      M22      M33      M44
## 391.4416 389.9461 390.5425 293.6423
```

M4 and M44 exhibit the lowest MSPE values for the flights.sm and flights.sm.AA datasets, respectively, indicating that they are the superior models.