FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION

OF HIGHER EDUCATION

ITMO UNIVERSITY

Project report

Performed by:

Victor Voronin (J41321c)

Andrey Antonov (J41332c)

Nelubin Daniil (J41325c)

Saint-Petersburg

2021

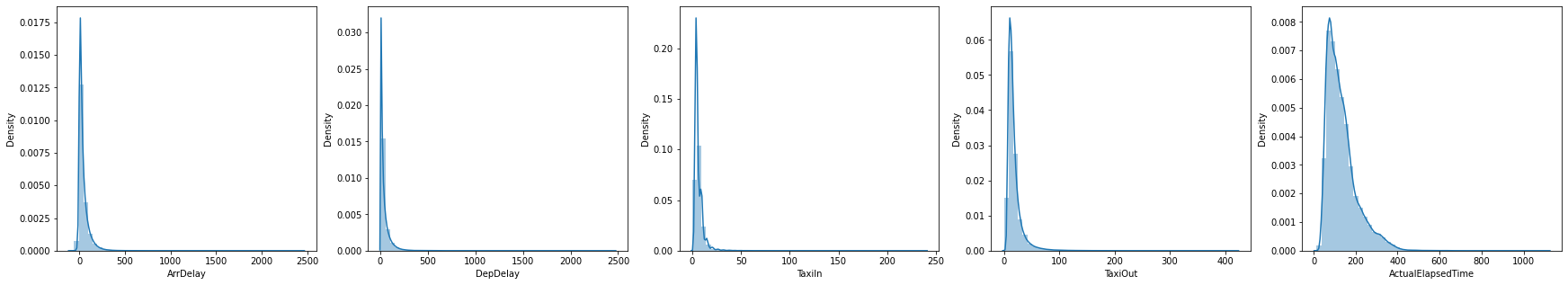
Analysis of univariate random variables

# Dataset

For our experiments we used follow dataset [AirlinesDelay](https://www.kaggle.com/giovamata/airlinedelaycauses?select=DelayedFlights.csv) and follow columns for univariable analysis:

|  |  |  |
| --- | --- | --- |
| Variable | Description | Type |
| ArrDelay | arrival delay | Continuous |
| DepDelay | departure delay | Continuous |
| TaxiIn | taxi time from wheels down to arrival at the gate | Continuous |
| TaxiOut | taxi time from departure from the gate to wheels up | Continuous |
| ActualElapsedTime | actual elapsed time of the flight | Continuous |

# Histograms and kde



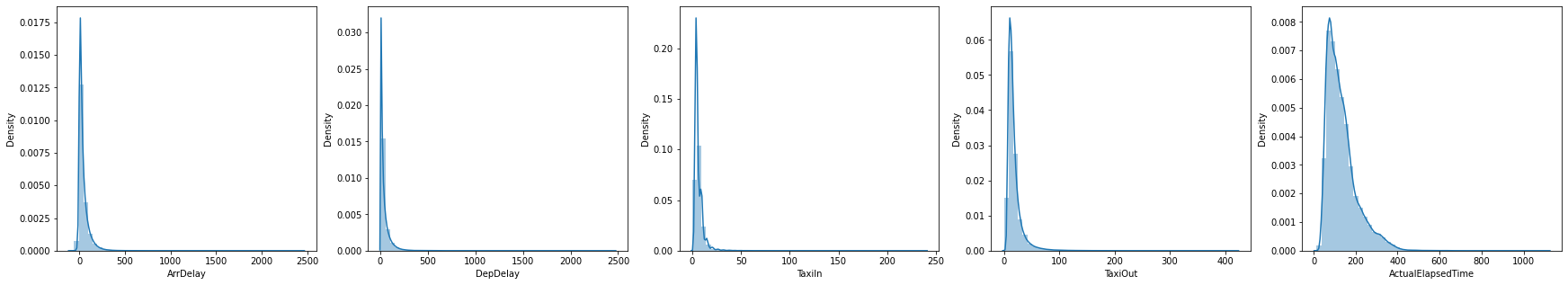
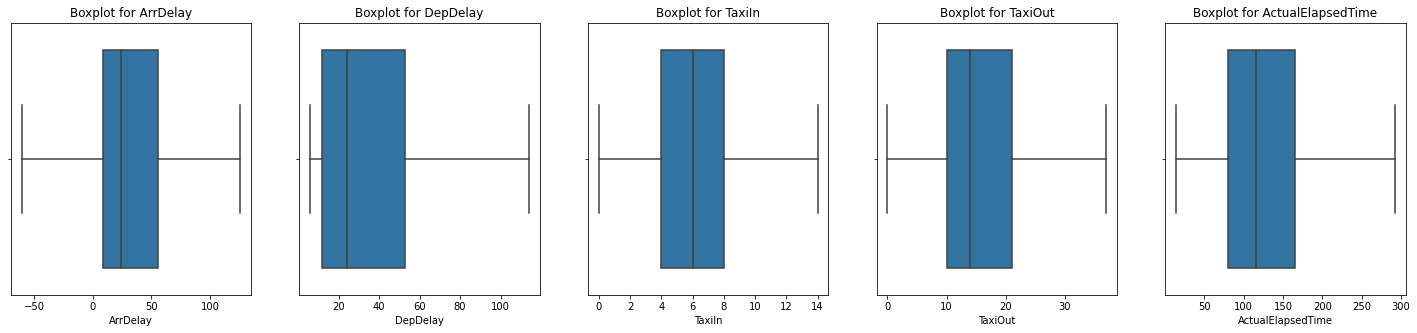


Figure 1 Histograms of chosen variables

# “Box with whiskers” plot



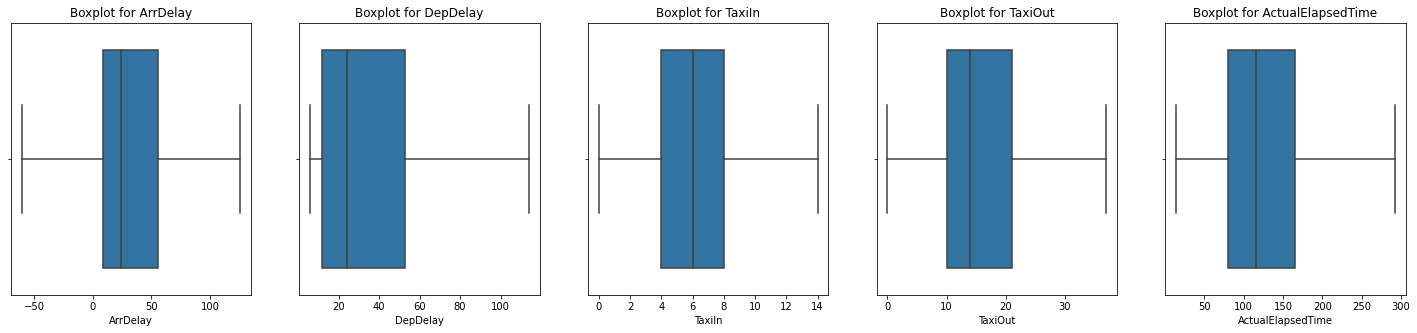


Figure 2 “Box with whiskers” plots

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | min | median | max |
| ArrDelay | -109.0 | 24.0 | 2461.0 |
| DepDelay | 6.0 | 24.0 | 2467.0 |
| TaxiIn | 0.0 | 6.0 | 240.0 |
| TaxiOut | 0 | 14.0 | 422.0 |
| ActualElapsedTime | 14.0 | 116.0 | 1114.0 |

# Estimation of distribution parameters using MLE and LS methods

Method best\_fit\_distribution selects the best theoretical distribution from a given set in the MLE sense to the observed distribution. The method returns the name of the best fit distribution and its parameters. The parameters are estimated with attribute “.fit” in scipy.stats.rv\_continuous.fit. The perform of this method for our data usually takes about 9 minutes.

For implementation of least square method, we use scipy.optimize.curve\_fit. It has as a optimizing function the function of theoretical distribution.

## ArrDelay

The call of best\_fit\_distribution for arrDelay says that arrDelay has exponnorm distribution and returns its parameters.

Probability density function (pdf) of observed random variable as well as theoretical ones are showed in figure 3 where “mle” means theoretical distribution with the parameters which were fitted by maximum likelihood methods and “ls” means theoretical distribution with the parameters which were fitted by least square method.

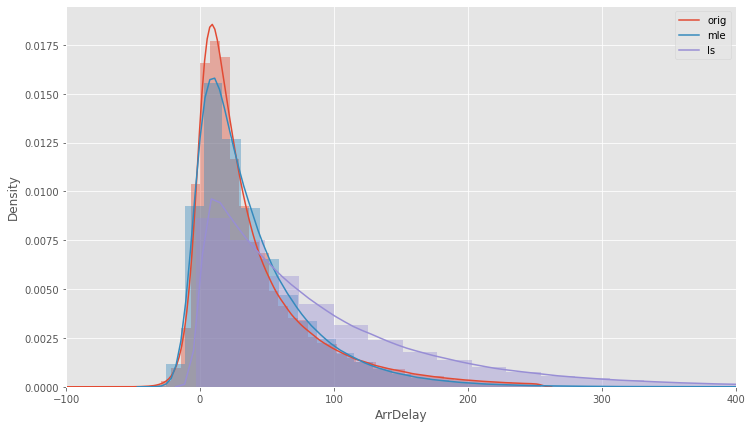


Figure 3 PDF of observed distribution and theoretical ones

We used Kolmogorov-Smirnov test and Cramér-von Mises like statistical tests for a check of correctness our assumption about theoretical distribution and its parameters. We obtained follow result (table 1):

Table 1 ‒ Statistical tests for arrDelay

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Null hypothesis | Test | Statistical | Pvalue | Conclusion |
| ArrDelay described by exponnorm distribution | Kolmogorov-Smirnov | 0.044993 | 0.981982 | do not reject the null hypothesis |
| Cramér-von Mises | 0.041901 | 0.924031 | do not reject the null hypothesis |

Draw qq-plot of observed distribution and theoretical one (mle) (figure 4)

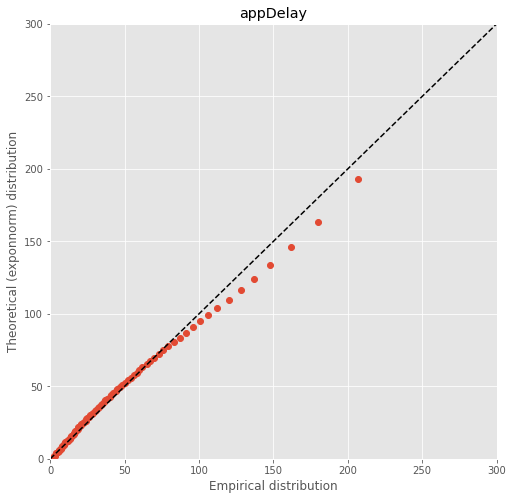


Figure 4 QQ-plot for observed distribution and theoretical one

## DepDelay

The call of best\_fit\_distribution for depDelay says that depDelay has gamma distribution and returns its parameters.

Probability density function of observed random variable as well as theoretical ones are showed in figure 5 where “mle” means theoretical distribution with the parameters which were fitted by maximum likelihood methods and “ls” means theoretical distribution with the parameters which were fitted by least square method.

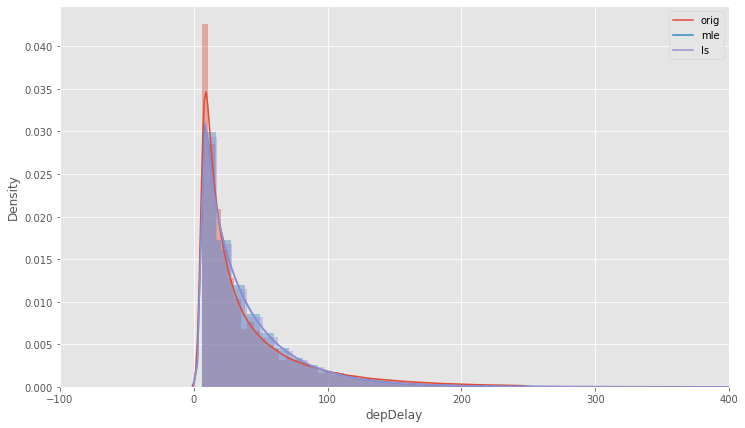


Figure 5 PDF of observed distribution and theoretical ones

We used Kolmogorov-Smirnov test and Cramér-von Mises like statistical tests for a check of correctness our assumption about theoretical distribution and its parameters. We obtained follow result (table 2):

Table 2 ‒ Statistical tests for depDelay

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Null hypothesis | Test | Statistical | Pvalue | Conclusion |
| depDelay described by gamma distribution | Kolmogorov-Smirnov | 0.08444 | 0.995138 | do not reject the null hypothesis |
| Cramér-von Mises | 0.035604 | 0.957899 | do not reject the null hypothesis |

Draw qq-plot of observed distribution and theoretical one (mle) (figure 6)

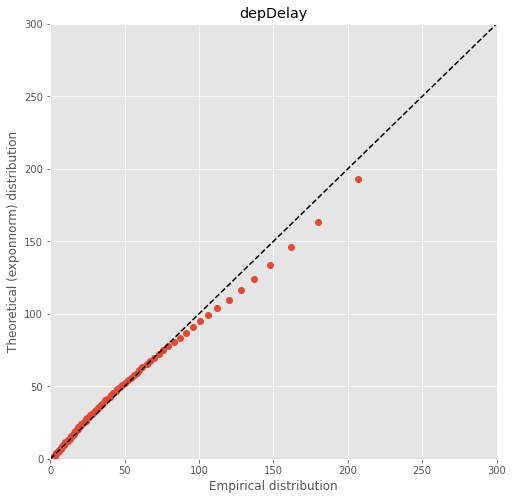


Figure 6 QQ-plot for observed distribution and theoretical one

## TaxiIn

The call of best\_fit\_distribution for arrDelay says that taxiIn has exponnorm distribution and returns its parameters.

Probability density function of observed random variable as well as theoretical ones are showed in figure 7 where “mle” means theoretical distribution with the parameters which were fitted by maximum likelihood methods and “ls” means theoretical distribution with the parameters which were fitted by least square method.

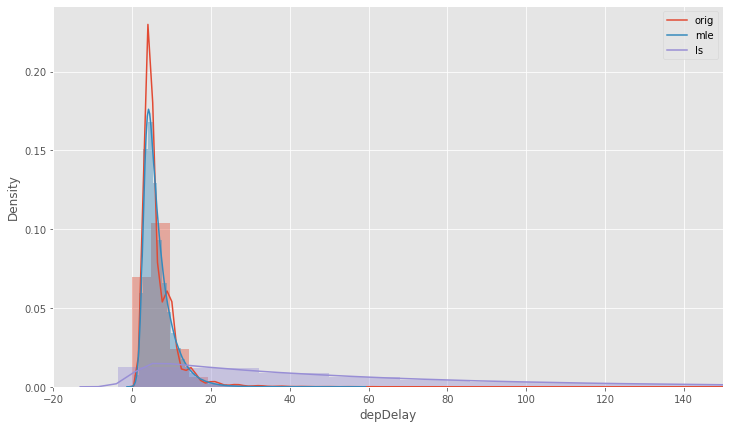


Figure 7 PDF of observed distribution and theoretical ones

We used Kolmogorov-Smirnov test and Cramér-von Mises like statistical tests for a check of correctness our assumption about theoretical distribution and its parameters. We obtained follow result (table 3):

Table 3 ‒ Statistical tests for taxiIn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Null hypothesis | Test | Statistical | Pvalue | Conclusion |
| taxiIn described by exponnorm distribution | Kolmogorov-Smirnov | 0.090159 | 0.989473 | do not reject the null hypothesis |
| Cramér-von Mises | 0.028307 | 0.983621 | do not reject the null hypothesis |

Draw qq-plot of observed distribution and theoretical one (mle) (figure 8)

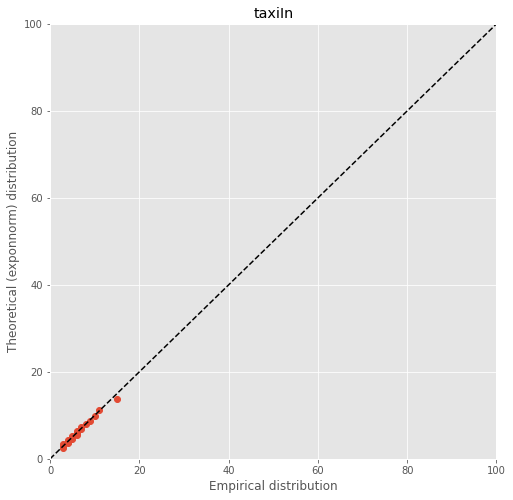


Figure 8 QQ-plot for observed distribution and theoretical one

## TaxiOut

The call of best\_fit\_distribution for arrDelay says that taxiOut has exponnorm distribution and returns its parameters.

Probability density function of observed random variable as well as theoretical ones are showed in figure 9 where “mle” means theoretical distribution with the parameters which were fitted by maximum likelihood methods and “ls” means theoretical distribution with the parameters which were fitted by least square method.

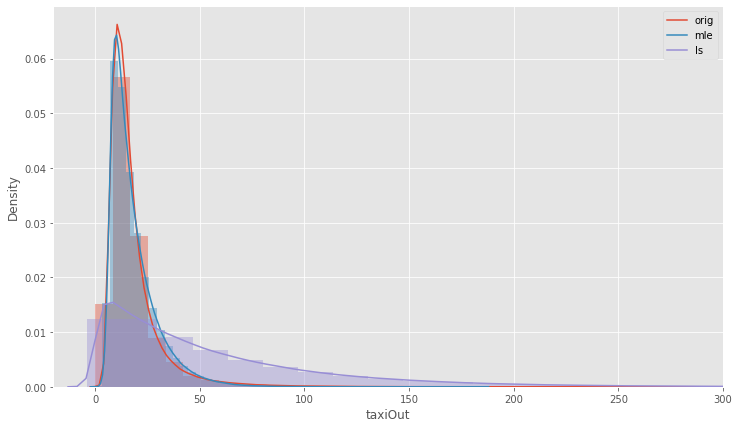


Figure 9 PDF of observed distribution and theoretical ones

We used Kolmogorov-Smirnov test and Cramér-von Mises like statistical tests for a check of correctness our assumption about theoretical distribution and its parameters. We obtained follow result (table 4):

Table 4 ‒ Statistical tests for taxiOut

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Null hypothesis | Test | Statistical | Pvalue | Conclusion |
| taxiOut described by exponnorm distribution | Kolmogorov-Smirnov | 0.073737 | 0.99931 | do not reject the null hypothesis |
| Cramér-von Mises | 0.02132 | 0.99649 | do not reject the null hypothesis |

Draw qq-plot of observed distribution and theoretical one (mle) (figure 10)

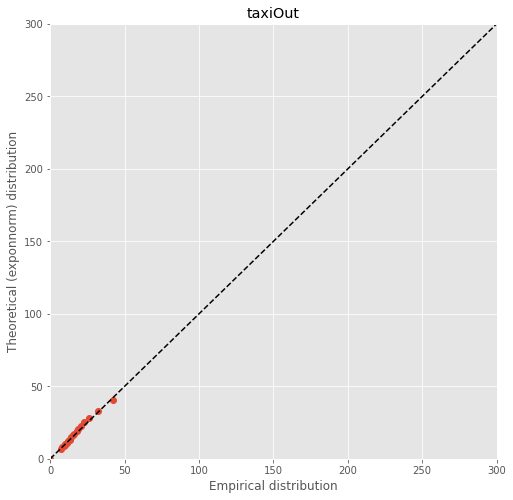


Figure 10 QQ-plot for observed distribution and theoretical one

It is quite expected that TaxiIn and TaxiOut have the same distribution. It because there is strict procedure for flight crew and all what could have gone wrong it would have to finish on the previous stages. The return of the airplane from taxiing or run-up is costly to the reputation of the company.

## ActualElapsedTime

The call of best\_fit\_distribution for arrDelay says that actualTimeElapsed has gamma distribution and returns its parameters.

Probability density function of observed random variable as well as theoretical ones are showed in figure 11 where “mle” means theoretical distribution with the parameters which were fitted by maximum likelihood methods and “ls” means theoretical distribution with the parameters which were fitted by least square method.

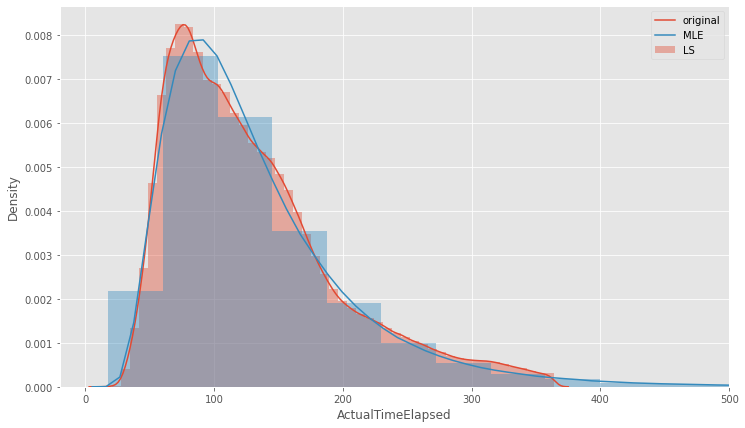


Figure 11 PDF of observed distribution and theoretical ones

There are two strategies for planning the route grid in the world. Point-to-point and hub ones. The first one is characterized by many short, direct routes. Very convenient for passengers, very ruinous for airlines. Hub strategy implies the existence of large airports through which almost all flights go. This is the situation like in Russia with Moscow's airports. The same strategy in USA and we can see the realization of that on the plot. Mean elapsed time 116 minutes per flight, that is about two hours, that is there are not many many short route, there are quite long route to nearest hub.

If we try to draw the same plot for Europe airlines we see that peak of the pdf lies to the left. Because Europe use direct route strategy. Someone can say “it’s just because Europe is smaller!”. Yes, it’s true. that is why there is a direct route system. Being determines consciousness as Marx said.

We used Kolmogorov-Smirnov test and Cramér-von Mises like statistical tests for a check of correctness our assumption about theoretical distribution and its parameters. We obtained follow result (table 5):

Table 2 ‒ Statistical tests for ActualTime

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Null hypothesis | Test | Statistical | Pvalue | Conclusion |
| ActualTime described by lognorm distribution | Kolmogorov-Smirnov | 0.028464 | 0.999991 | do not reject the null hypothesis |
| Cramér-von Mises | 0.009865 | 0.999995 | do not reject the null hypothesis |

Draw qq-plot of observed distribution and theoretical one (mle) (figure 12)

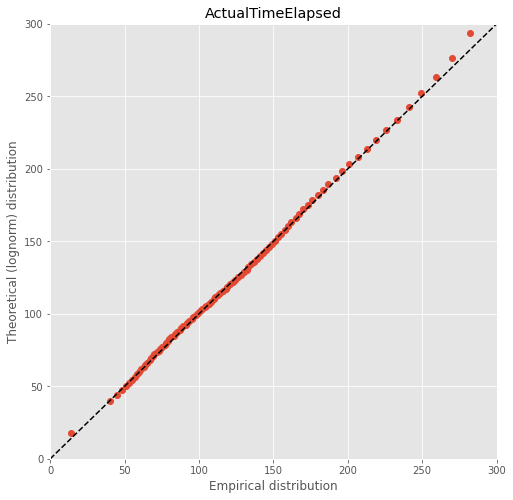


Figure 12 QQ-plot for observed distribution and theoretical one

Analysis of univariate random variables

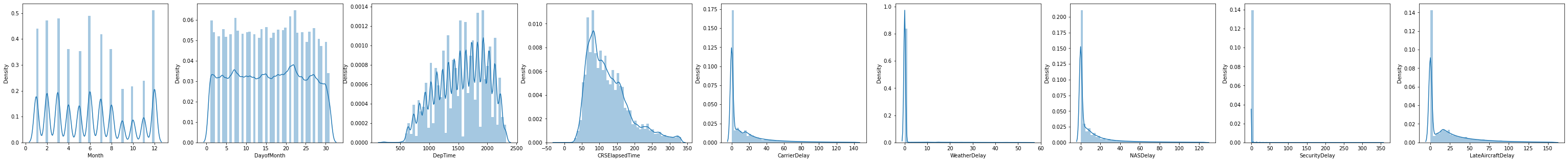
# Dataset

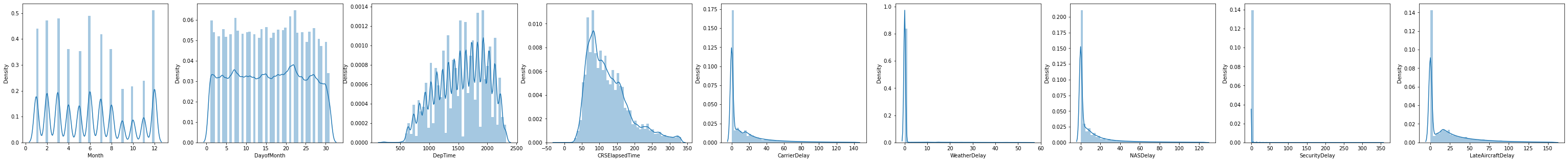
For our experiments we used follow dataset [AirlinesDelay](https://www.kaggle.com/giovamata/airlinedelaycauses?select=DelayedFlights.csv) and follow columns for univariable analysis:

Table 1 ‒ Description variables

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Type |
| ArrDelay | arrival delay | continuous |
| Month | month of the flight | discrete |
| DayofMonth | day of the month (1 to 31) | discrete |
| DepTime | actual departure time | continuous |
| CRSElapsedTime | scheduled elapsed time of the flight | continuous |
| CarrierDelay | delay, in minutes, attributable to the carrier | continuous |
| WeatherDelay | delay, in minutes, attributable to weather factors | continuous |
| NASDelay | delay, in minutes, attributable to the National Aviation System | continuous |
| SecurityDelay | delay, in minutes, attributable to security factors | continuous |
| LateAircraftDelay | delay, in minutes, attributable to late-arriving aircraft | continuous |

# Histograms and kde





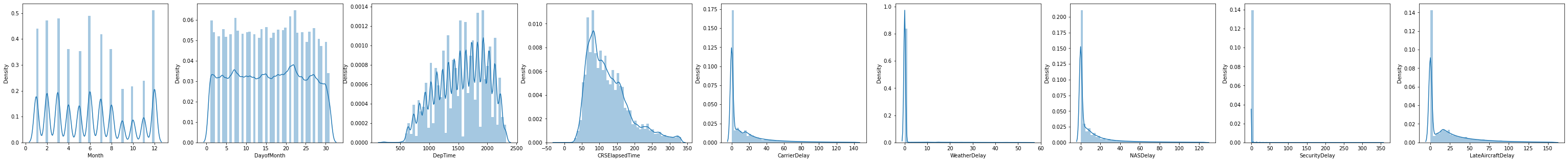


Figure 13 Histograms of chosen variables

Abovementioned histograms were created from cut initial random variables. The truncating was performed with follow bound:

* 0.11 percentile < DepTime <0.99 percentile
* CRSElapsedTime < 0.98 percentile
* CarrierDelay < 0.98 percentile
* WeatherDelay < 0.98 percentile
* NASDelay < 0.98 percentile
* SecurityDelay without truncating
* LateAircraftDelay < 0.98 percentile

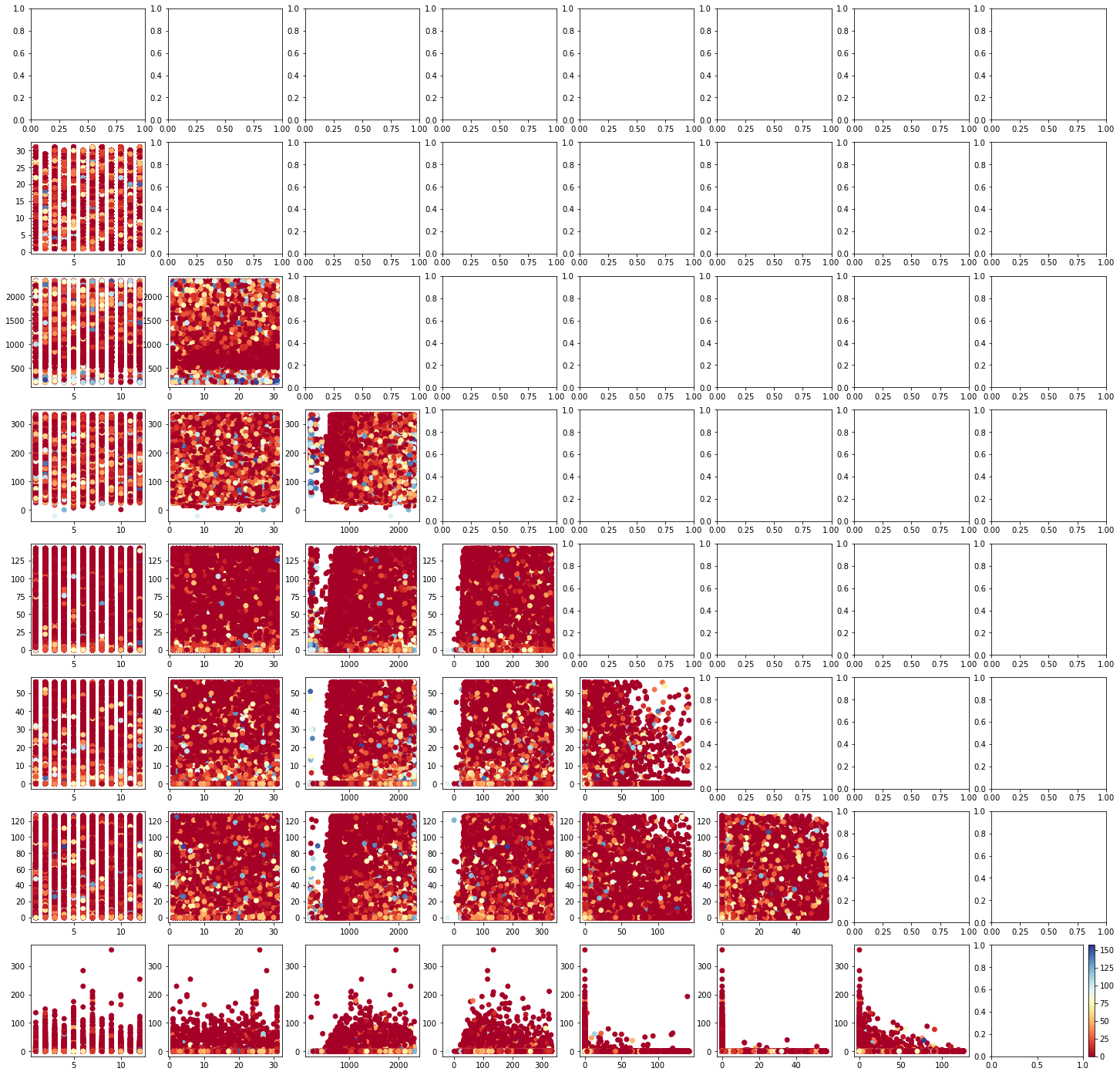
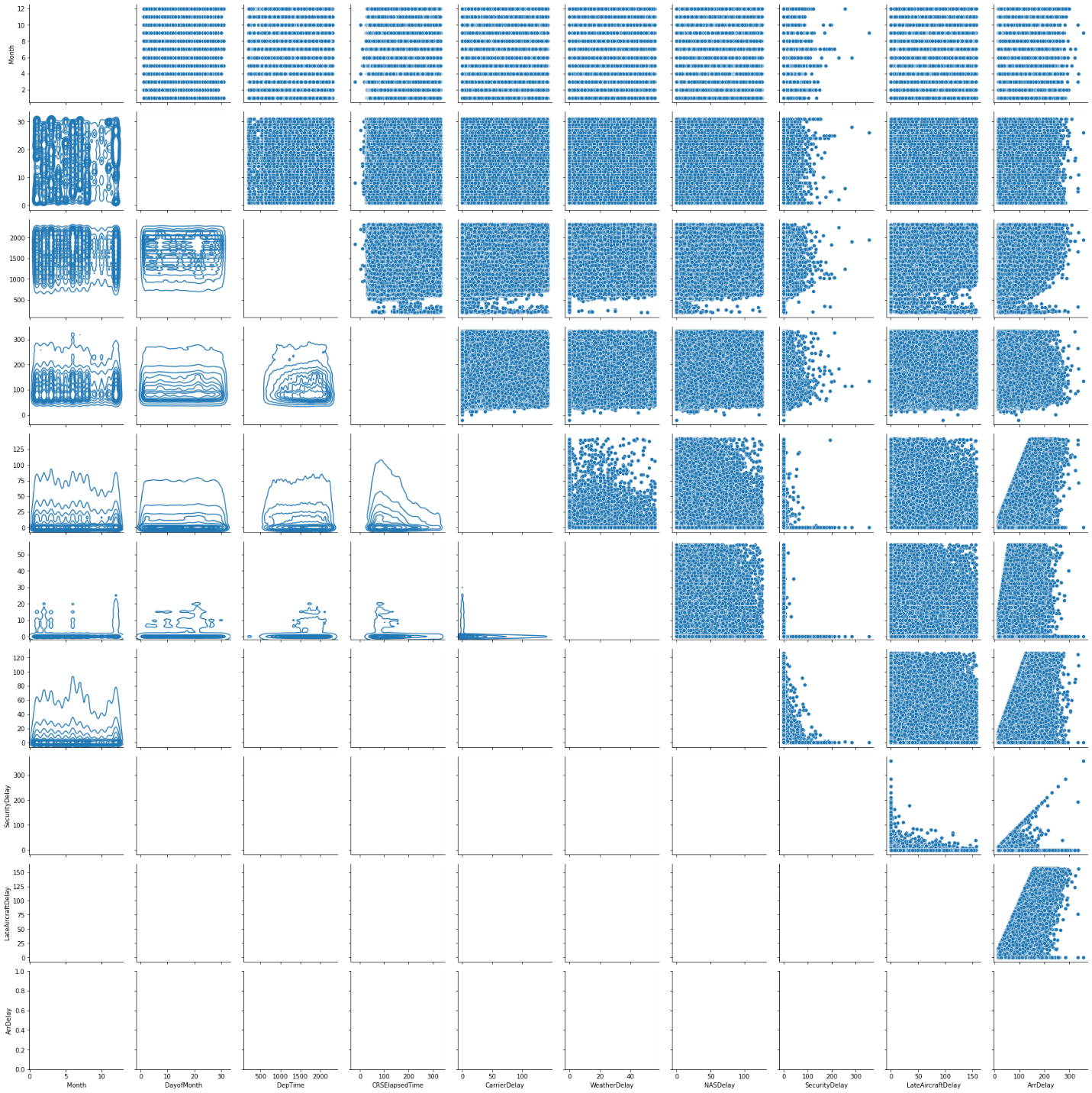


Figure 14 Joint distribution for each pair variables with arrDelay colorbar



# Estimation of the expected value and variance

Table 2 ‒ Estimation of the expected value and variance

|  |  |  |
| --- | --- | --- |
| Variable | Mean | Variance |
| Month | 6 | 12 |
| DayofMonth | 15 | 77 |
| DepTime | 1558.832 (19:58:05) | 206415.606 (14:32:38) |
| CRSElapsedTime | 131.764 (18:20:10) | 4834.697 16:43:41 |
| CarrierDelay | 19.179 | 1896.273 |
| WeatherDelay | 3.703 | 461.886 |
| NASDelay | 15.022 | 1144.676 |
| SecurityDelay | 0.09 | 4.091 |
| LateAircraftDelay | 25.296 | 1768.612 |

# Estimation of the conditional expected value and variance

We try to predict arrive delay so we interesting to consider conditional distribution, mean and average under condition that arrive delay have become the certain value. To realize that at the first we choose from the dataset only the entries the predictor with the first value of arrive delay. Calculate mean and variance for obtained series. After that fix next value of arrive delay and choose from the dataset all values of a predictor with this value of target variable. Repeating this cycle, finally we have got the table with a number of rows that equal to a number of unique value in arrive delay and a number of columns that equal to a number of the predictors. Fragment of the table is showed below:

Table 3 ‒ Conditional mean and variance

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ArrrDelay** | **Month** | **DayofMonth** | **DepTime** | **CRSElapsedTime** | **CarrierDelay** | **WeatherDelay** | **NASDelay** | **SecurityDelay** | **LateAircraftDelay** |
| **15.0** | 6.081177 | 15.695619 | 1475.370881 | 122.797911 | 6.384579 | 0.426682 | 2.446377 | 0.102607 | 5.639755 |
| **16.0** | 6.085253 | 15.711426 | 1476.812365 | 123.129947 | 6.649504 | 0.466562 | 2.700827 | 0.099790 | 6.083317 |
| **17.0** | 6.060566 | 15.737445 | 1478.528964 | 123.673937 | 6.990092 | 0.495857 | 3.002536 | 0.096209 | 6.415306 |
| **18.0** | 6.053121 | 15.692020 | 1483.892040 | 123.146693 | 7.169047 | 0.561157 | 3.323913 | 0.097521 | 6.848362 |
| **19.0** | 6.080368 | 15.737161 | 1486.680383 | 124.099996 | 7.510831 | 0.572856 | 3.640127 | 0.090932 | 7.185254 |

289 rows × 9 columns

# Estimation of pair correlation coefficients, confidence intervals

Month - DayofMonth

p: 0.000

corr: 0.065

Conf interval: 0.063 0.067

Month - DepTime

p: 0.000

corr: -0.011

Conf interval: -0.012 -0.009

Month - CRSElapsedTime

p: 0.000

corr: 0.004

Conf interval: 0.002 0.005

Month - CarrierDelay

p: 0.000

corr: -0.003

Conf interval: -0.005 -0.001

Month - WeatherDelay

p: 0.000

corr: 0.006

Conf interval: 0.005 0.008

Month - NASDelay

p: 0.000

corr: 0.018

Conf interval: 0.017 0.020

Month - SecurityDelay

p: 0.000

corr: -0.003

Conf interval: -0.005 -0.002

Month - LateAircraftDelay

p: 0.326

corr: -0.001

Conf interval: -0.003 0.001

Month - ArrDelay

p: 0.000

corr: 0.009

Conf interval: 0.007 0.011

DayofMonth - DepTime

p: 0.000

corr: 0.003

Conf interval: 0.002 0.005

DayofMonth - CRSElapsedTime

p: 0.846

corr: -0.000

Conf interval: -0.002 0.002

DayofMonth - CarrierDelay

p: 0.358

corr: -0.001

Conf interval: -0.003 0.001

DayofMonth - WeatherDelay

p: 0.000

corr: 0.005

Conf interval: 0.003 0.007

DayofMonth - NASDelay

p: 0.720

corr: 0.000

Conf interval: -0.002 0.002

DayofMonth - SecurityDelay

p: 0.374

corr: -0.001

Conf interval: -0.003 0.001

DayofMonth - LateAircraftDelay

p: 0.000

corr: 0.004

Conf interval: 0.002 0.006

DayofMonth - ArrDelay

p: 0.000

corr: 0.004

Conf interval: 0.002 0.006

DepTime - CRSElapsedTime

p: 0.000

corr: -0.039

Conf interval: -0.041 -0.038

DepTime - CarrierDelay

p: 0.000

corr: -0.067

Conf interval: -0.069 -0.066

DepTime - WeatherDelay

p: 0.000

corr: -0.007

Conf interval: -0.009 -0.005

DepTime - NASDelay

p: 0.000

corr: -0.025

Conf interval: -0.027 -0.023

DepTime - SecurityDelay

p: 0.000

corr: -0.018

Conf interval: -0.020 -0.016

DepTime - LateAircraftDelay

p: 0.000

corr: 0.259

Conf interval: 0.264 0.267

DepTime - ArrDelay

p: 0.000

corr: 0.168

Conf interval: 0.168 0.171

CRSElapsedTime - CarrierDelay

p: 0.000

corr: 0.009

Conf interval: 0.007 0.010

CRSElapsedTime - WeatherDelay

p: 0.059

corr: -0.002

Conf interval: -0.004 0.000

CRSElapsedTime - NASDelay

p: 0.000

corr: 0.079

Conf interval: 0.077 0.081

CRSElapsedTime - SecurityDelay

p: 0.000

corr: 0.005

Conf interval: 0.003 0.007

CRSElapsedTime - LateAircraftDelay

p: 0.000

corr: -0.031

Conf interval: -0.033 -0.029

CRSElapsedTime - ArrDelay

p: 0.000

corr: 0.028

Conf interval: 0.026 0.030

CarrierDelay - WeatherDelay

p: 0.000

corr: -0.116

Conf interval: -0.119 -0.115

CarrierDelay - NASDelay

p: 0.000

corr: -0.180

Conf interval: -0.183 -0.180

CarrierDelay - SecurityDelay

p: 0.000

corr: -0.027

Conf interval: -0.029 -0.025

CarrierDelay - LateAircraftDelay

p: 0.000

corr: -0.305

Conf interval: -0.317 -0.313

CarrierDelay - ArrDelay

p: 0.000

corr: 0.318

Conf interval: 0.327 0.331

WeatherDelay - NASDelay

p: 0.000

corr: 0.008

Conf interval: 0.006 0.010

WeatherDelay - SecurityDelay

p: 0.000

corr: -0.010

Conf interval: -0.011 -0.008

WeatherDelay - LateAircraftDelay

p: 0.000

corr: -0.055

Conf interval: -0.057 -0.053

WeatherDelay - ArrDelay

p: 0.000

corr: 0.051

Conf interval: 0.049 0.053

NASDelay - SecurityDelay

p: 0.000

corr: -0.013

Conf interval: -0.015 -0.011

NASDelay - LateAircraftDelay

p: 0.000

corr: -0.141

Conf interval: -0.144 -0.140

NASDelay - ArrDelay

p: 0.000

corr: 0.368

Conf interval: 0.384 0.388

SecurityDelay - LateAircraftDelay

p: 0.000

corr: -0.024

Conf interval: -0.026 -0.022

SecurityDelay - ArrDelay

p: 0.000

corr: 0.008

Conf interval: 0.006 0.010

LateAircraftDelay - ArrDelay

p: 0.000

corr: 0.583

Conf interval: 0.666 0.669

Correlation is square matrix with a number of rows that equals to a number of a variables. In our case it is 10x10. Correlation matrix is symmetrical one in relate to the diagonal. So ,we create down-corner matrix mask and calculate correlation between all variables (fig. 3)

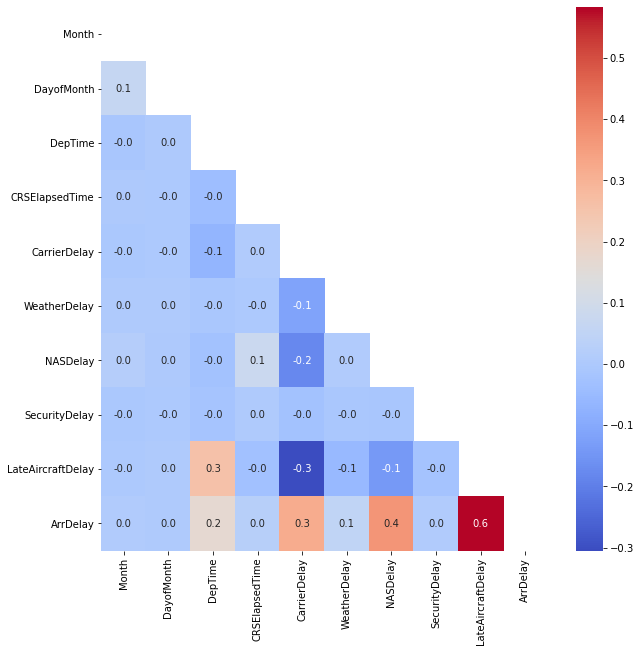


Figure 15 Correlation matrix

As we can see from correlation matrix the significant correlation between in example ArrDelay and LateAircraftDelay. However, pvalue of these correlation coefficients is low. We have extracted every non-zero p-value from the last point and have got follow:

Month – LateAircraftDelay p: 0.326

DayofMonth – CRSElapsedTime p: 0.846

DayofMonth – CarrierDelay p: 0.358

DayofMonth – NASDelay p: 0.720

DayofMonth – SecurityDelay p: 0.374

WeatherDelay – CRSElapsedTime p: 0.059

In other words, of course, we have come strong correlation between predictors, but they all are not statistically significant. Those who are such ones have near-zero correlation coefficient.

# Estimate multivariate correlation

We can see that our target variable ‒ arrDelay ‒ has strong linear dependency with the late airplane feed. This kind of delay is often linked with technical problems on board. Until maintenance stuff don’t sign the certificate release to service, announcement about the boarding start is not made.

Also, arrDelay has little slower correlation with NASDelay, CarrierDelay, DepTime.

Delay that is within the control of the National Airspace System (NAS) may include: non-extreme weather conditions, airport operations, heavy traffic volume, air traffic control, etc. Delays that occur after Actual Gate Out are usually attributed to the NAS and are also reported through OPSNET.

Carrier delay is within the control of the air carrier. Examples of occurrences that may determine carrier delay are: aircraft cleaning, aircraft damage, awaiting the arrival of connecting passengers or crew, baggage, bird strike, cargo loading, catering, computer, outage-carrier equipment, crew legality (pilot or attendant rest), damage by hazardous goods, engineering inspection, fueling, handling disabled passengers, late crew, lavatory servicing, maintenance, oversales, potable water servicing, removal of unruly passenger, slow boarding or seating, stowing carry-on baggage, weight and balance delays.

At the usual level it seems that weather has strong influence into the arrDelay, hovewer, we don’t see strong dependence between arrDedlay and weather.

# Task formulation for regression

We try to predict arrive delay knowing date and time the taking off and also magnitude of delays. Pilots could try to catch up the schedule in air, but civil airplane cannot fly at the speed above then 900 km/h. So, we have to take into account average time in air and do that with CRSElapsedTime.

As we can see from correlation matrix there is weak linear dependencies between predictors. Thus, we don’t use penalties

We create train and test dataset with train\_test\_split and given test\_size = 25% of all available data. Linear regression model was trained with train dataset and we got the predict had getting for test dataset.

Table 4 ‒ Observed and predicted arrDelay

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Month | DayofMonth | CarrierDelay | LateAircraftDelay | NASDelay | WeatherDelay | ArrDelay\_pred | ArrDelay |
| 379245 | 3 | 8 | 0.0 | 29.0 | 0.0 | 0.0 | 29.175600 | 29.0 |
| 1131391 | 7 | 23 | 125.0 | 0.0 | 6.0 | 0.0 | 130.757825 | 131.0 |
| 1217640 | 7 | 12 | 21.0 | 0.0 | 0.0 | 0.0 | 21.169990 | 21.0 |
| 239699 | 2 | 26 | 0.0 | 0.0 | 0.0 | 47.0 | 46.994852 | 47.0 |
| 278300 | 2 | 1 | 0.0 | 101.0 | 0.0 | 0.0 | 100.978162 | 101.0 |

276984 rows × 8 columns

# The quality of the regression

The coefficient is defined as , where u is the residual sum of squares and v is the total sum of squares . The best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a score of 0.0.

In our case we have a score

As metrics of the regression quality we chosen RMSE, MAPE

For out test dataset we got:

RMSE = 3.988047389719877e-13

RMSE = 2.015

MAPE = 6.813394903841156e-13

MAPE = 0.007

Apart from that we can plot histogram test and predict target variable and in full agreement with metrics we see well result.

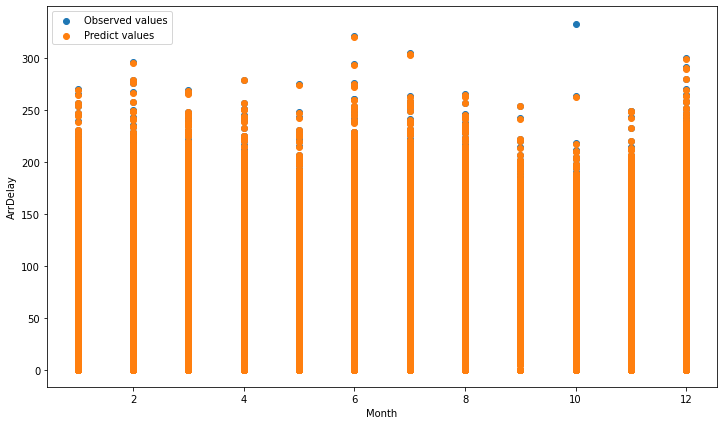


Figure 16 Observed and predicted arrDelay from test dataset

Plotting the distribution of residuals with norm\_hist = True



Figure 17 Distribution of residuals

For residual distribution:

* Mean = 0.0031
* Var = 4.414

Histogram of residual proves again good quality of regression. The most part of error fall to the first near-zero bin.

Sampling of multivariate random variables

# Dataset

For our experiments we used the following 7 predictor variables:

* ArrTime – Arrival time
* DepTime – Departure time
* CRSArrTime – Scheduled arrival time
* CRSDepTime – Scheduled departure time
* Distance – Distance traveled
* LaneAircraftDelay – Aircraft delay
* CarrierDelay – Carrier delay

and 3 target variables:

* ArrDelay – Arrival delay
* DepDelay – Departure delay
* ActualElapsedTime – Actual elapsed time of the flight

# Sampling

Here we estimated the best fit distribution from lab 1 and sampled our target variables using accept-reject and inverse transform sampling methods. We then creatd QQ biplots to visualy estimate the quality of sampling

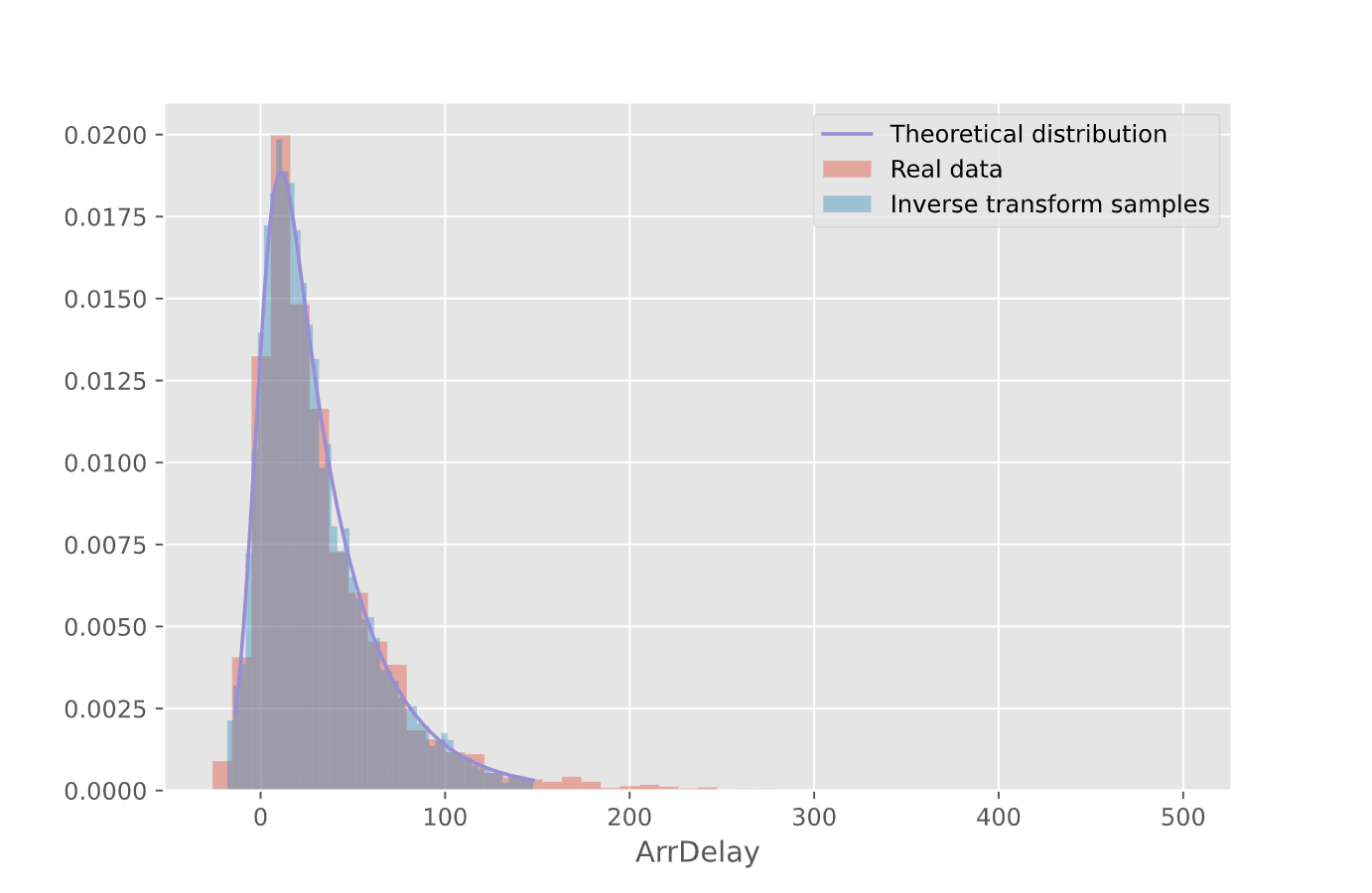


Figure 18 -- ArrDelay inverse-transform sampling

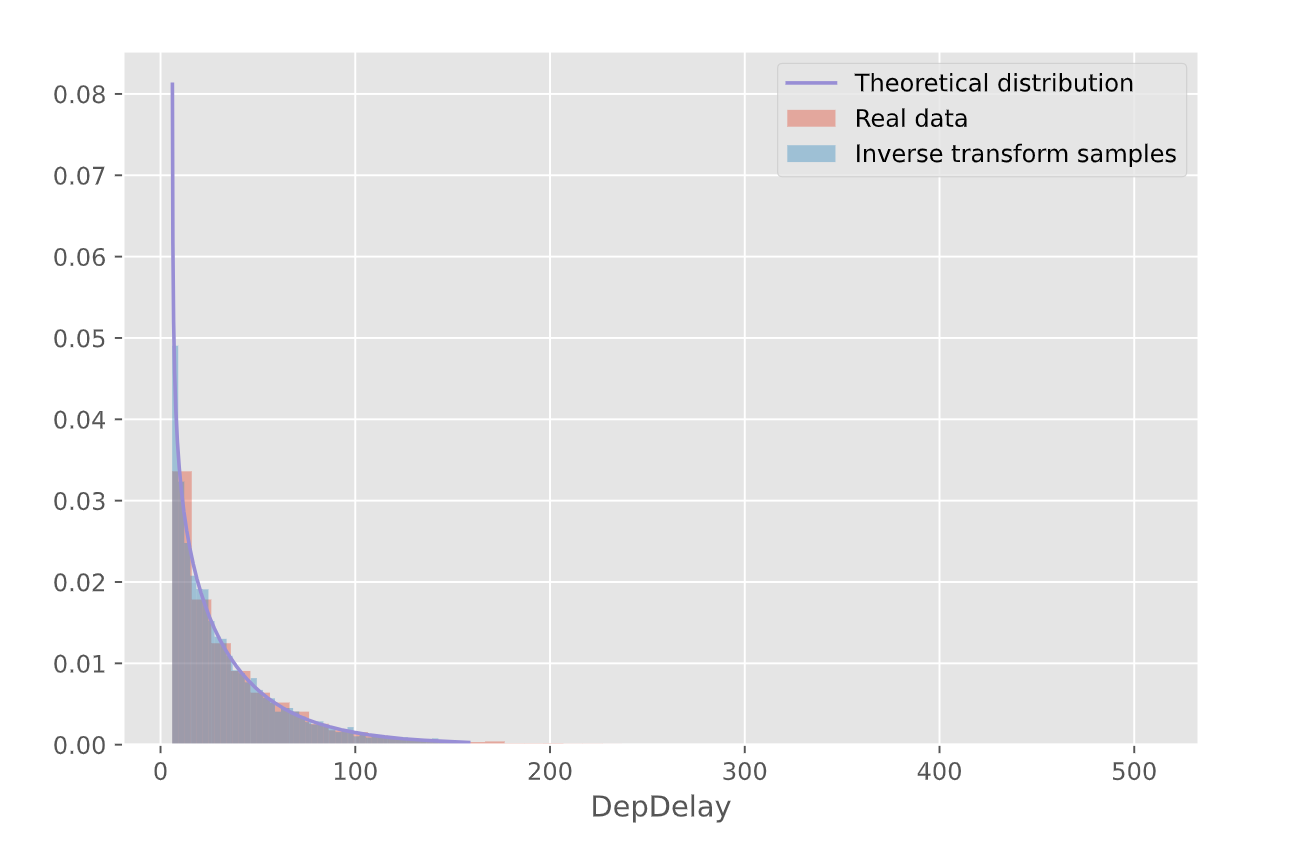


Figure 19 -- DepDelay inverse transform sampling

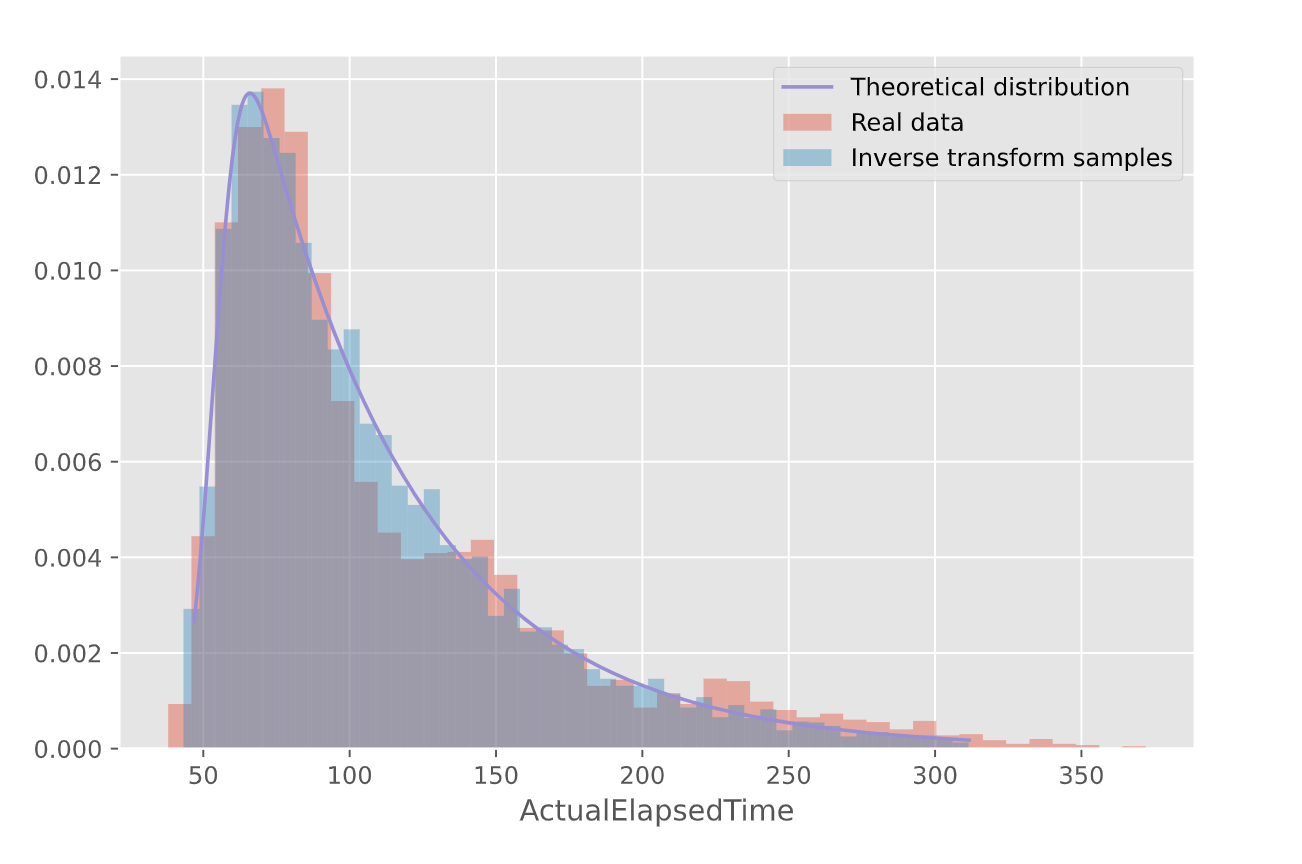


Figure 20 -- ActualElapsedTime inverse transform sampling

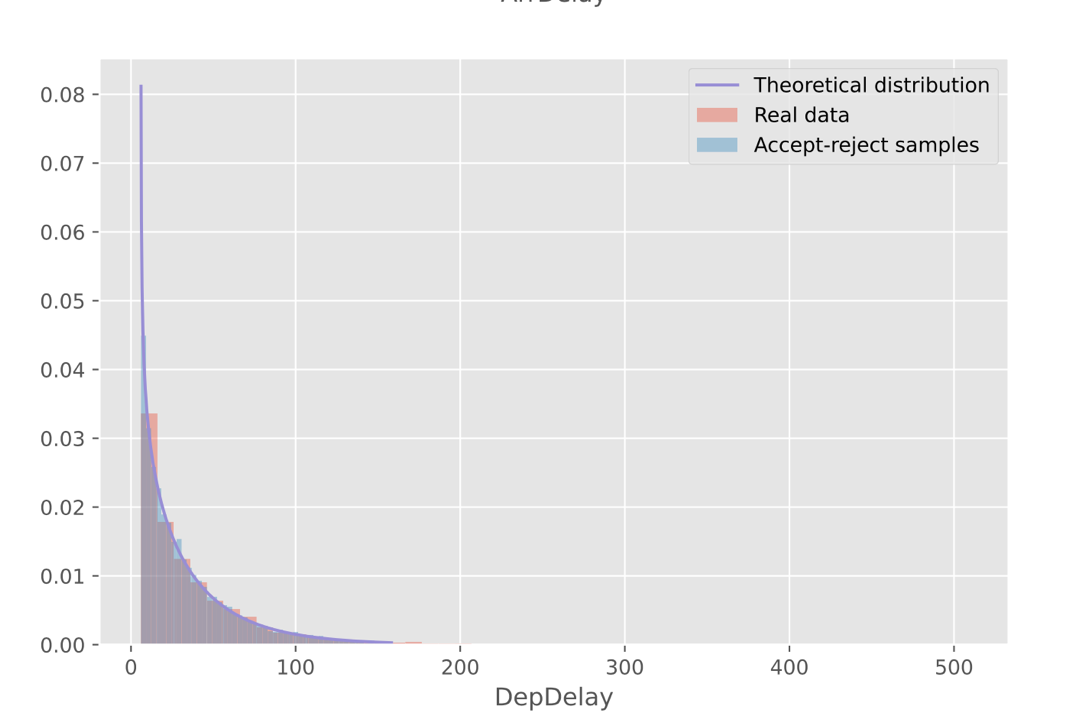


Figure 21 -- DepDelay reject sampling

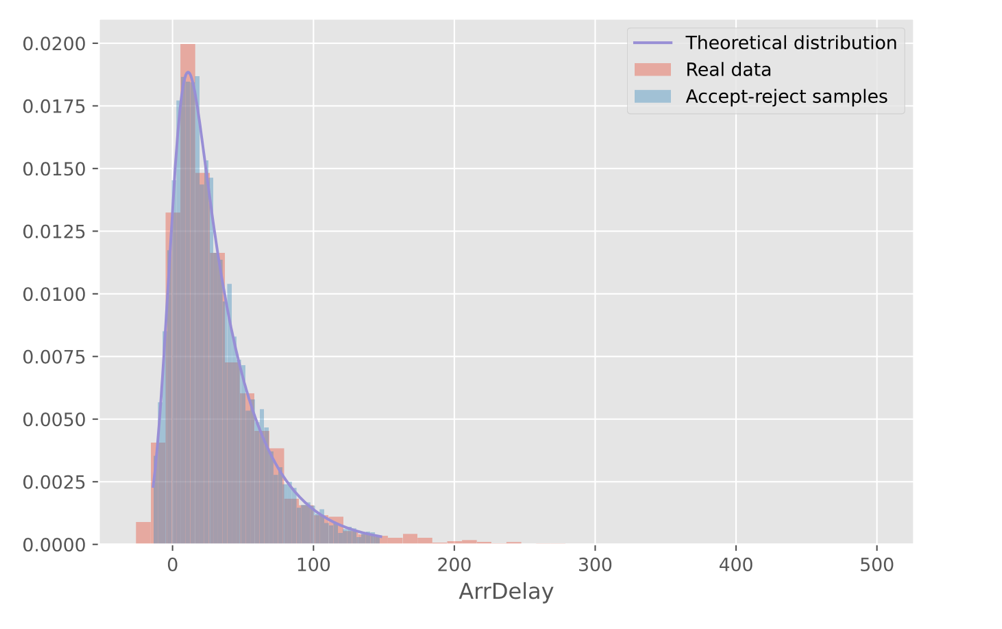


Figure 22 -- ArrDelay reject sampling

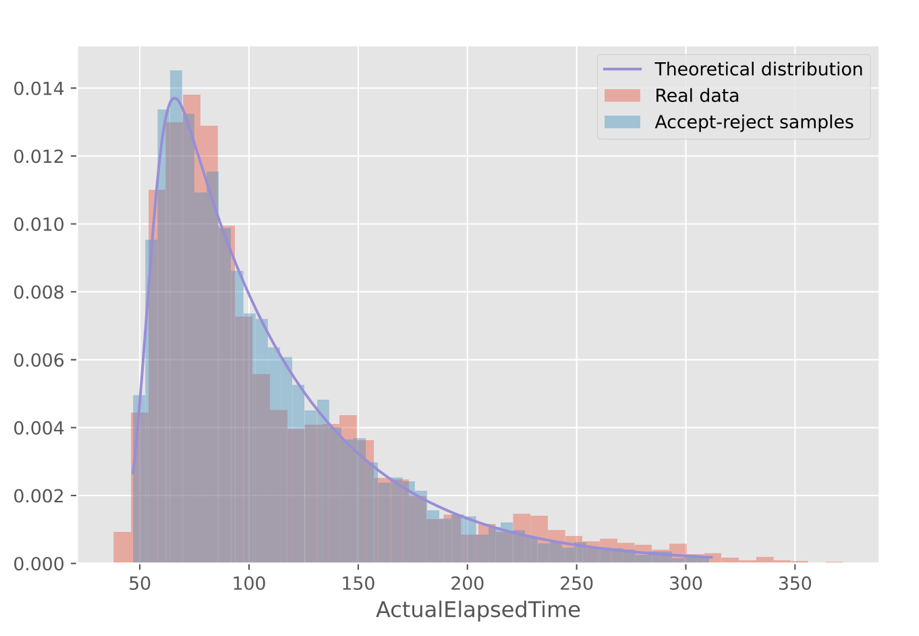
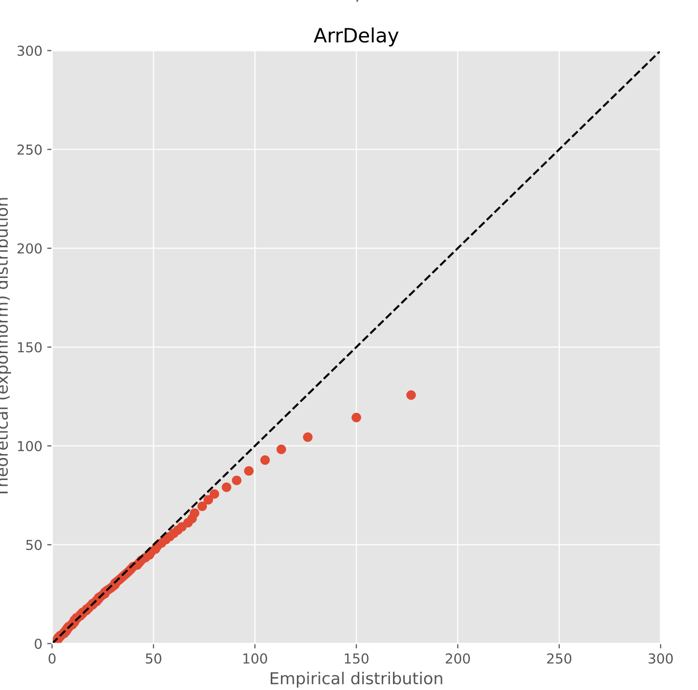
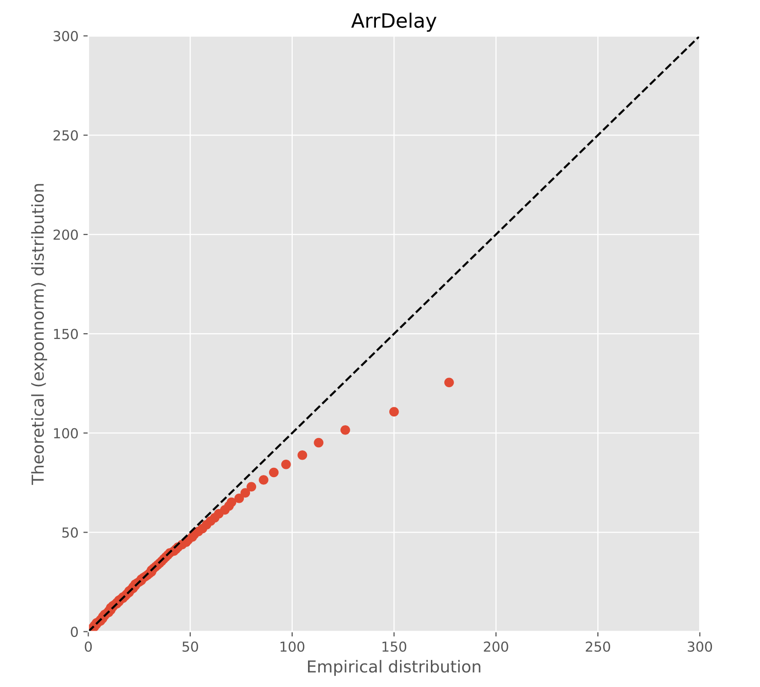


Figure 23 -- AcrualElapcedTime reject sampling



# Bayesian networks

Figure 24 -- QQ-biplots for inverse-transform sampling (on the left) and reject sampling (on the right)

First of all, we conducted an analisys of correlation between targets and predictors to find the strongest paired correlations.

# 

Figure 25 -- Correlation matrix of features

Then we build a naïve Bayesian network, modeling edges after the strongest paired correlations.

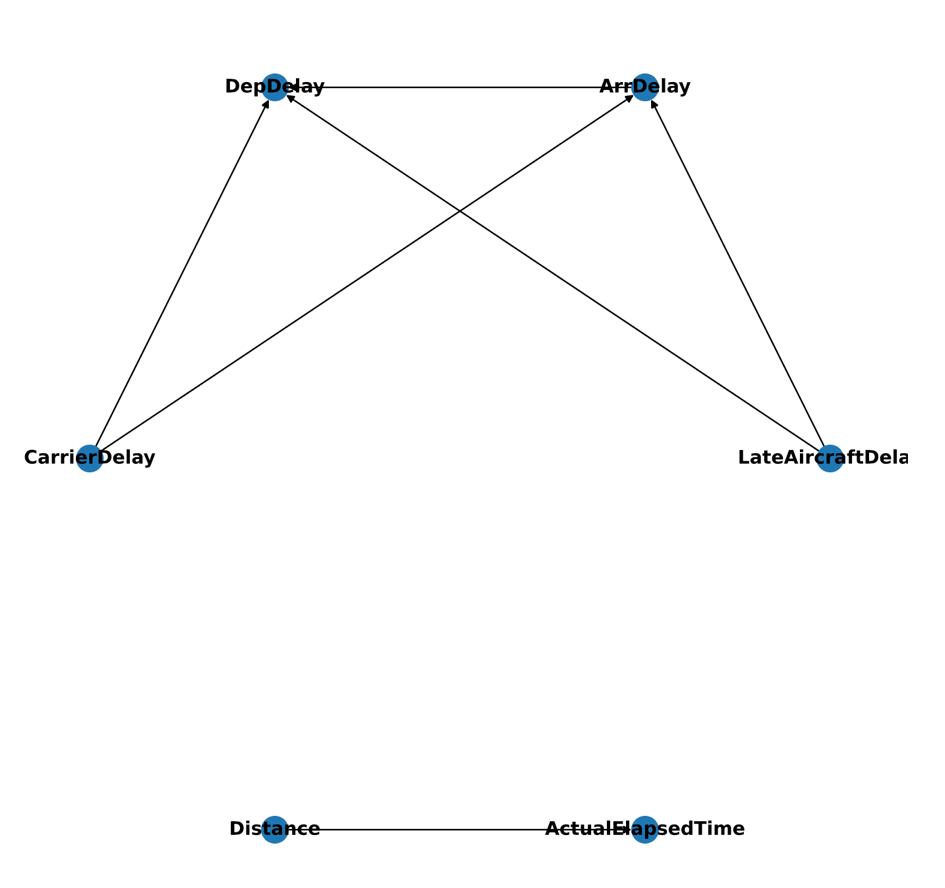


Figure 26 -- Naive Bayesian network

We also build two more networks using Hilclimb algorithm with BiScore and a Tree search algorithm.

# 

Figure 27 -- Bayesian network built with HilcLimb seach algorithm

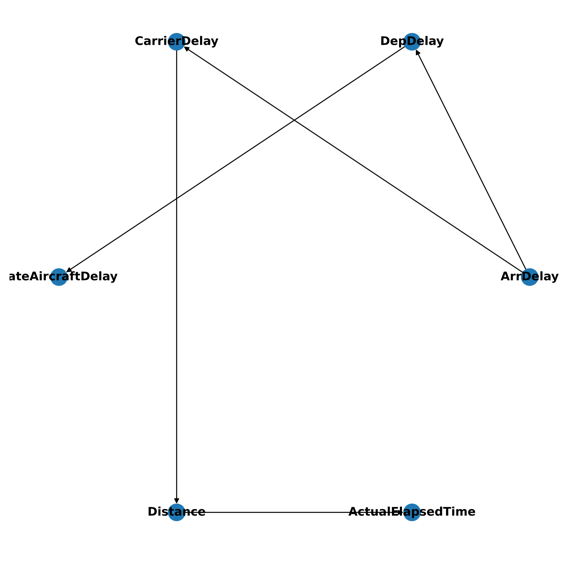
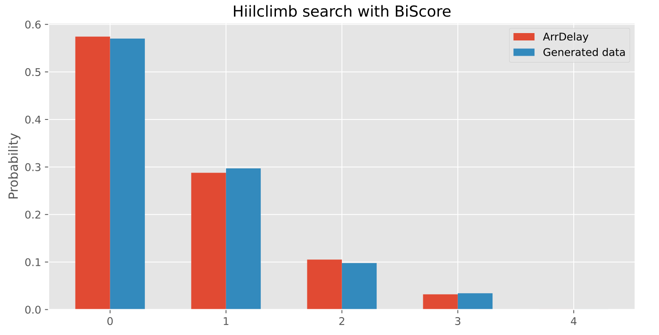
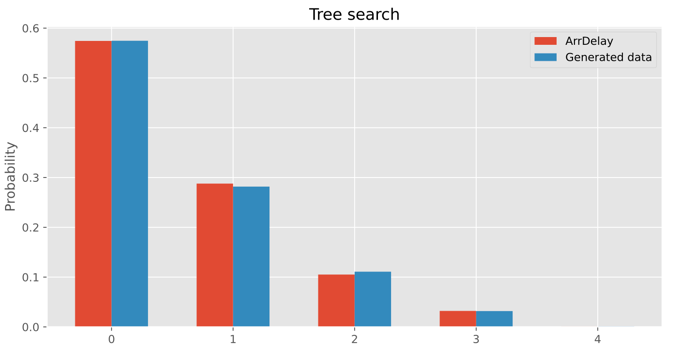
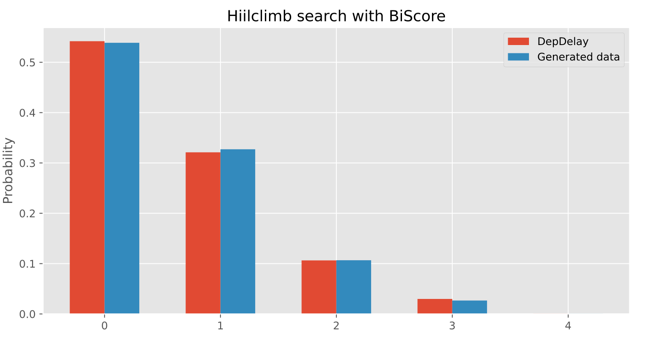
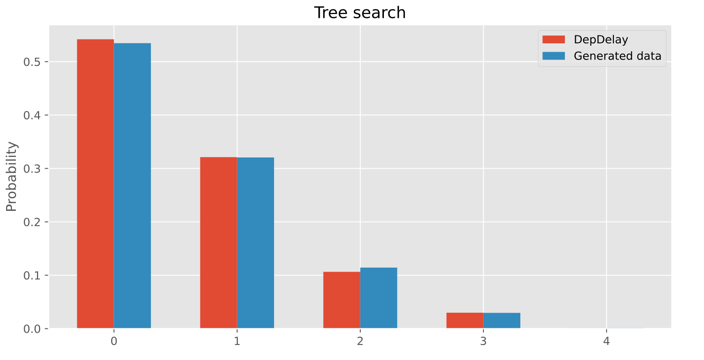


Figure 28 -- Bayesian network built withTree Search algorithm

# Training and analyzing the results

We trained all 3 networks on the target variables, then sampled the data and analyzied it’s quality using R-2 score, RMSE and MAE metrics.





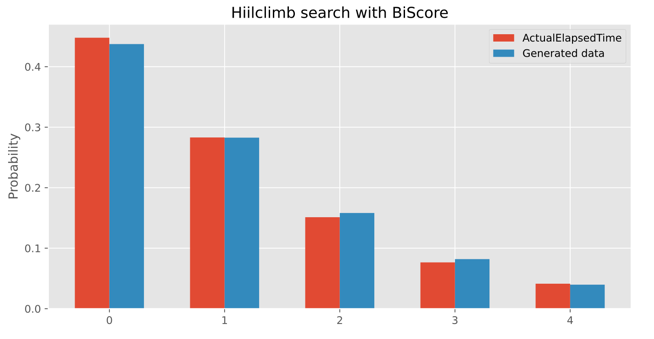
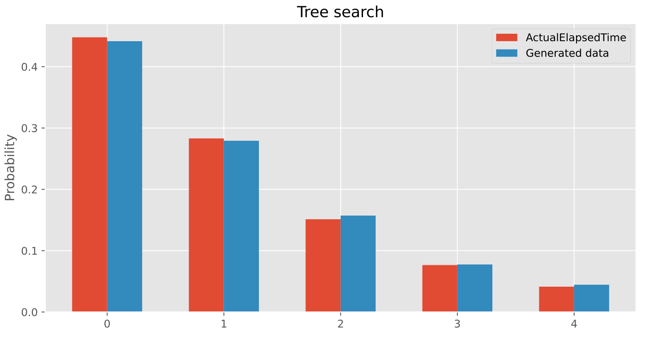


Figure 29 -- sampling from Bayesian networks

Table 1 – Manually constructed Bayesian network metrics

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | ArrDelay | DepDelay | ActualElapsedTime |
| R-2 | 0.66 | 0.91 | 0.86 |
| RMSE | 28.61 | 13.47 | 34.78 |
| MAE | 12.37 | 2.67 | 18.27 |

Table 2 -- Bayesian network build with Hillclimb search metrics

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | ArrDelay | DepDelay | ActualElapsedTime |
| R-2 | 0.88 | 0.87 | 0.85 |
| RMSE | 54.25 | 55.71 | 39.35 |
| MAE | 22.77 | 27.46 | 22.94 |

Table 3 -- Bayesian network built with Tree search metrics

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | ArrDelay | DepDelay | ActualElapsedTime |
| R-2 | 0.85 | 0.83 | 0.84 |
| RMSE | 58.87 | 63.15 | 40.20 |
| MAE | 24.53 | 30.07 | 23.82 |

As we can see from the obtained metrics, on our data a manually constructed network performed worse than the other networks, Hillclimb and Tree search methods shown roughly the same results.

Stationarity of the processes

# Dataset

For our experiments we used follow dataset [AirlinesDelay](https://www.kaggle.com/giovamata/airlinedelaycauses?select=DelayedFlights.csv) and follow columns for univariable analysis:

Table 1 ‒ Description variables

|  |  |  |
| --- | --- | --- |
| Column Name | Description | Type |
| ArrDelay | arrival delay | continuous |
| Month | month of the flight | discrete |
| DayofMonth | day of the month (1 to 31) | discrete |
| DepTime | actual departure time | continuous |
| CRSElapsedTime | scheduled elapsed time of the flight | continuous |
| CarrierDelay | delay, in minutes, attributable to the carrier | continuous |
| WeatherDelay | delay, in minutes, attributable to weather factors | continuous |
| NASDelay | delay, in minutes, attributable to the National Aviation System | continuous |
| SecurityDelay | delay, in minutes, attributable to security factors | continuous |
| LateAircraftDelay | delay, in minutes, attributable to late-arriving aircraft | continuous |

Merge columns Month and DayofMonth into one column “Date” and choose the most frequency flight from the initial dataset. We try to predict ArrDelay for two most frequency flight. These flights are 75 and 321 with 826 and 333 entries correspondently.

We also cut initial random variables. The truncating was performed with follow bound:

* 0.11 percentile < DepTime <0.99 percentile
* CRSElapsedTime < 0.98 percentile
* CarrierDelay < 0.98 percentile
* WeatherDelay < 0.98 percentile
* NASDelay < 0.98 percentile
* SecurityDelay without truncating
* LateAircraftDelay < 0.98 percentile

For FlightNum 75

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **DepTime** | **CRSElapsedTime** | **CarrierDelay** | **WeatherDelay** | **NASDelay** | **SecurityDelay** | **LateAircraftDelay** | **ArrDelay** |
| **2008-01-01** | 1740.0 | 80.0 | 4.0 | 0.0 | 21.0 | 0.0 | 16.0 | 41.0 |
| **2008-01-01** | 1214.0 | 152.0 | 53.0 | 0.0 | 6.0 | 0.0 | 0.0 | 59.0 |
| **2008-01-01** | 1959.0 | 105.0 | 0.0 | 0.0 | 4.0 | 0.0 | 60.0 | 64.0 |
| **2008-01-02** | 1932.0 | 105.0 | 28.0 | 0.0 | 0.0 | 0.0 | 0.0 | 28.0 |
| **2008-01-02** | 1401.0 | 75.0 | 4.0 | 0.0 | 16.0 | 0.0 | 12.0 | 32.0 |

816 rows × 10 columns

For FlightNum 321

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date** | **DepTime** | **CRSElapsedTime** | **CarrierDelay** | **WeatherDelay** | **NASDelay** | **SecurityDelay** | **LateAircraftDelay** | **ArrDelay** |
| **2008-01-01** | 1715.0 | 223.0 | 0.0 | 0.0 | 43.0 | 0.0 | 0.0 | 43.0 |
| **2008-01-01** | 2043.0 | 270.0 | 0.0 | 7.0 | 0.0 | 0.0 | 12.0 | 19.0 |
| **2008-01-02** | 1633.0 | 70.0 | 0.0 | 0.0 | 0.0 | 0.0 | 147.0 | 147.0 |
| **2008-01-02** | 1321.0 | 75.0 | 0.0 | 0.0 | 0.0 | 0.0 | 114.0 | 114.0 |
| **2008-01-02** | 2042.0 | 80.0 | 0.0 | 0.0 | 18.0 | 0.0 | 112.0 | 130.0 |

828 rows × 10 columns

# Stationary analysis.



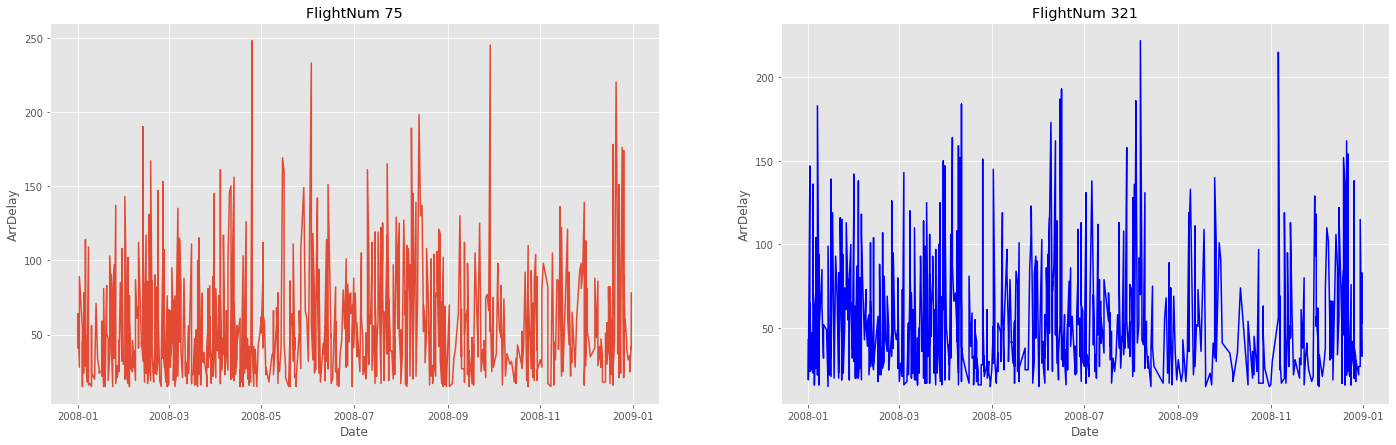


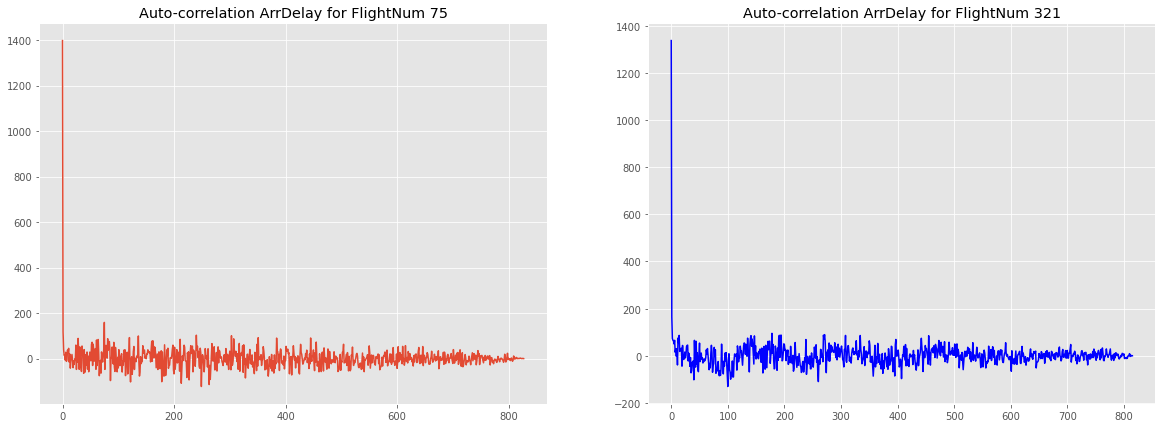
Figure 30 ArrDelay for chosen flights

We can conclude that both time series are quite stationary, but to gain confidence we perform the Dickey - Fuller test.

|  |  |  |  |
| --- | --- | --- | --- |
| Variable | Statistical value | p-value | Conclusion |
| ArrDelay | -26.413199 | 0.0 | stationary |
| DepTime | -14.900550 | 0.0 | stationary |
| CRSElapsedTime | -28.915694 | 0.0 | stationary |
| CarrierDelay | -22.915694 | 0.0 | stationary |
| WeatherDelay | -29.242249 | 0.0 | stationary |
| NASDelay | -19.131572 | 0.0 | stationary |
| SecurityDelay | -28.819522 | 0.0 | stationary |
| LateAircraftDelay | -27.852850 | 0.0 | stationary |

As we expected all our variables are stationarity.

# Auto-correlation and correlation



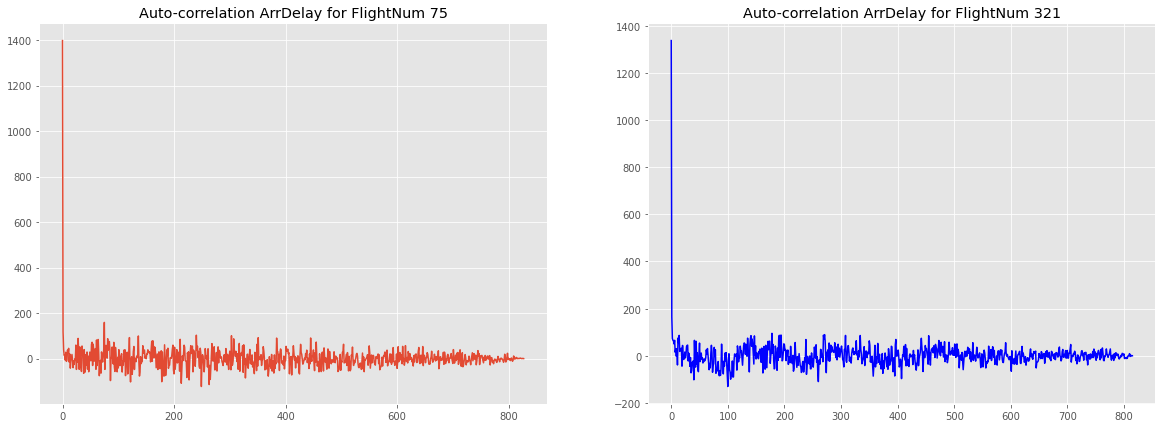


Figure 31 Autocorrelation for chosen flights

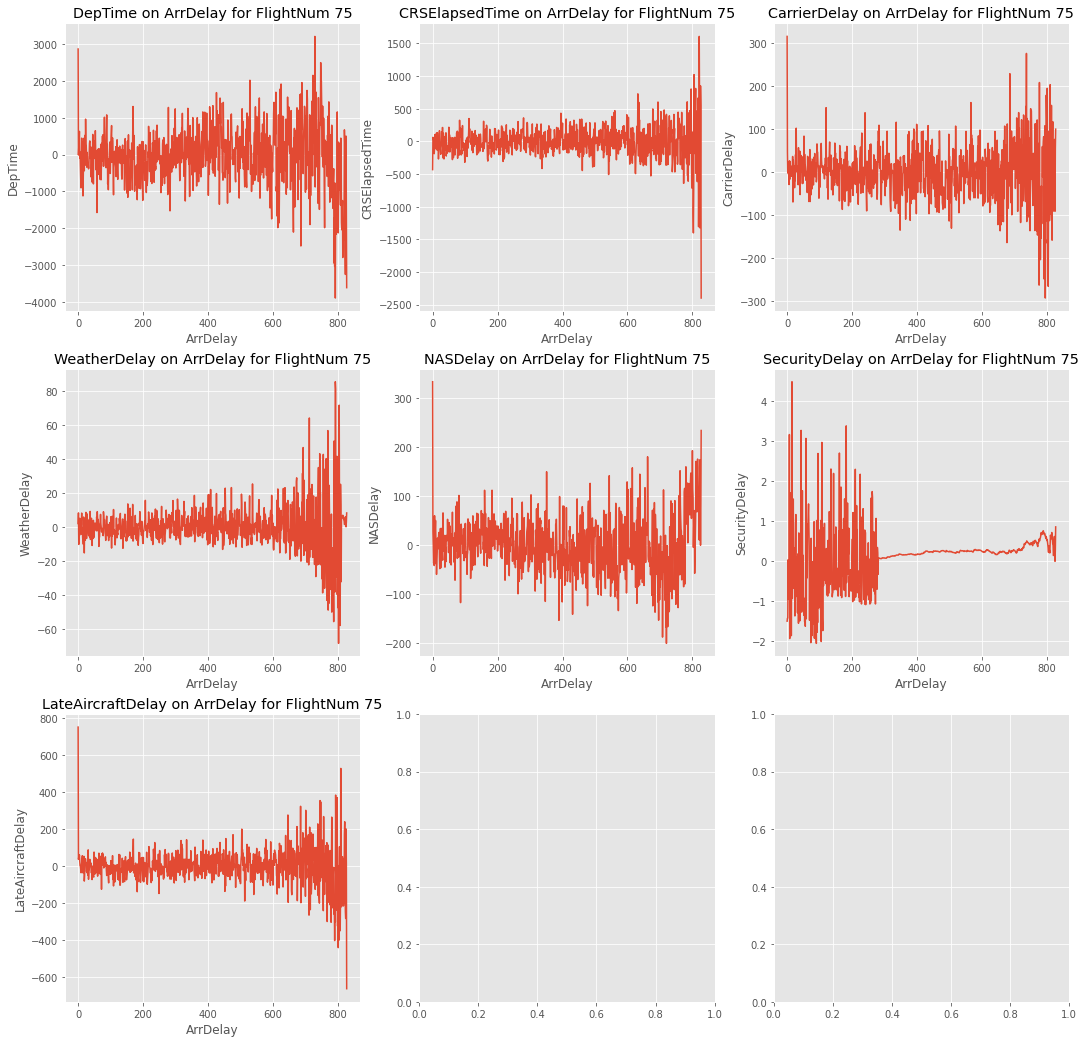


Figure 32 Correlation target variables and predictors for flightNum 75

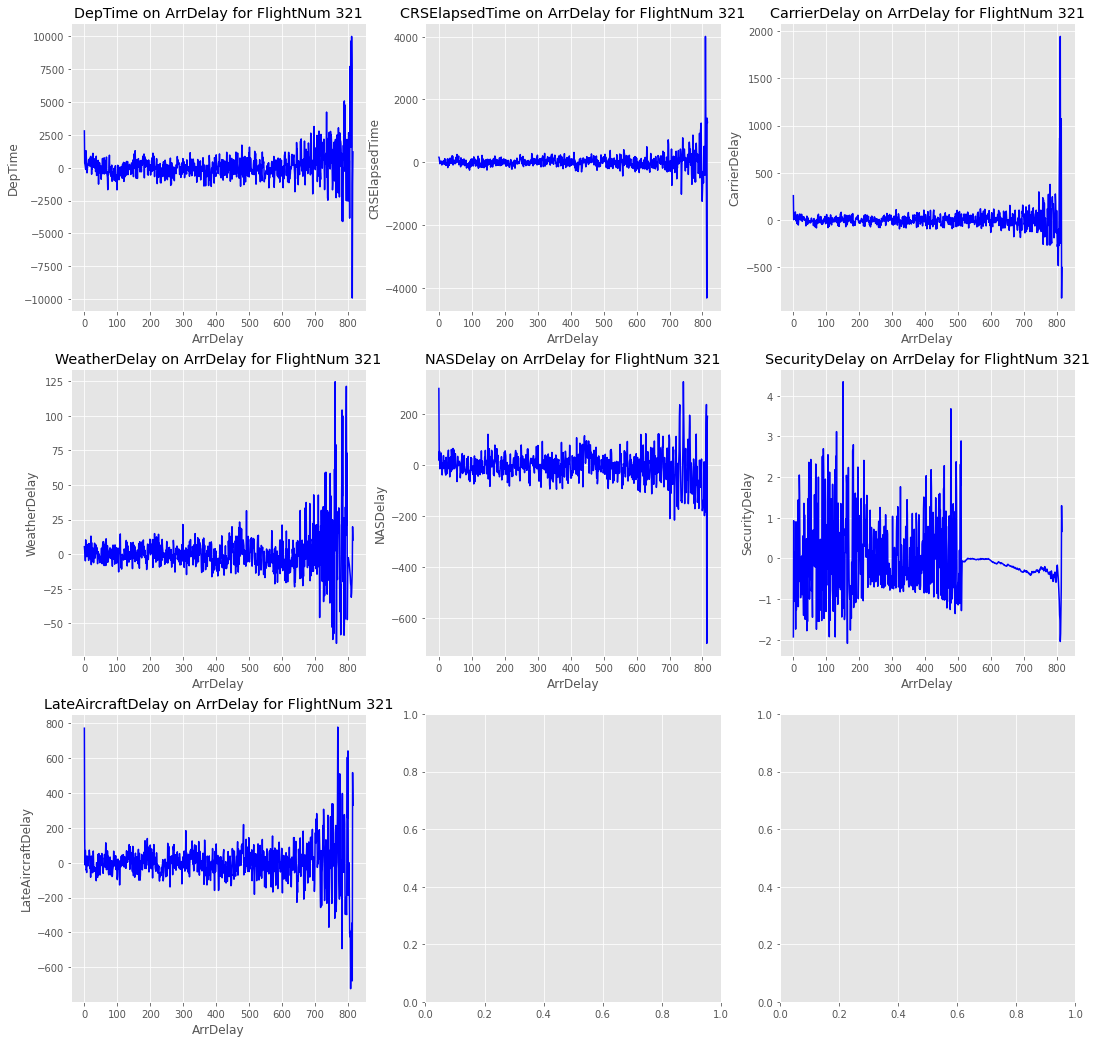


Figure 33 Correlation target variables and predictors for FlightNum 321

# Noise filtration

Noise has been filtered with Butterworth and Gaussian filters.

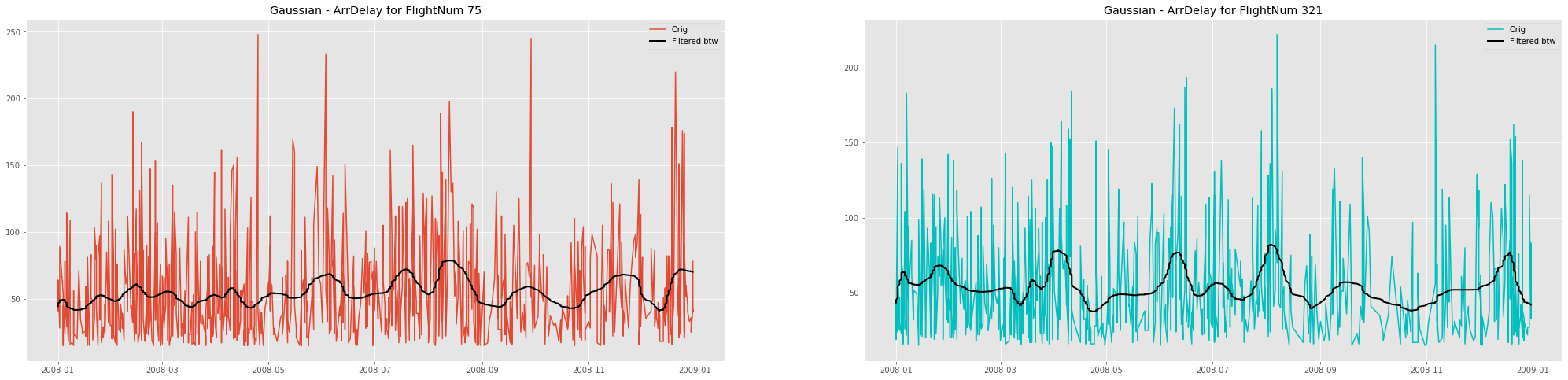


Figure 34 Filtered target variables with Butterworth filter for FlightNum 75

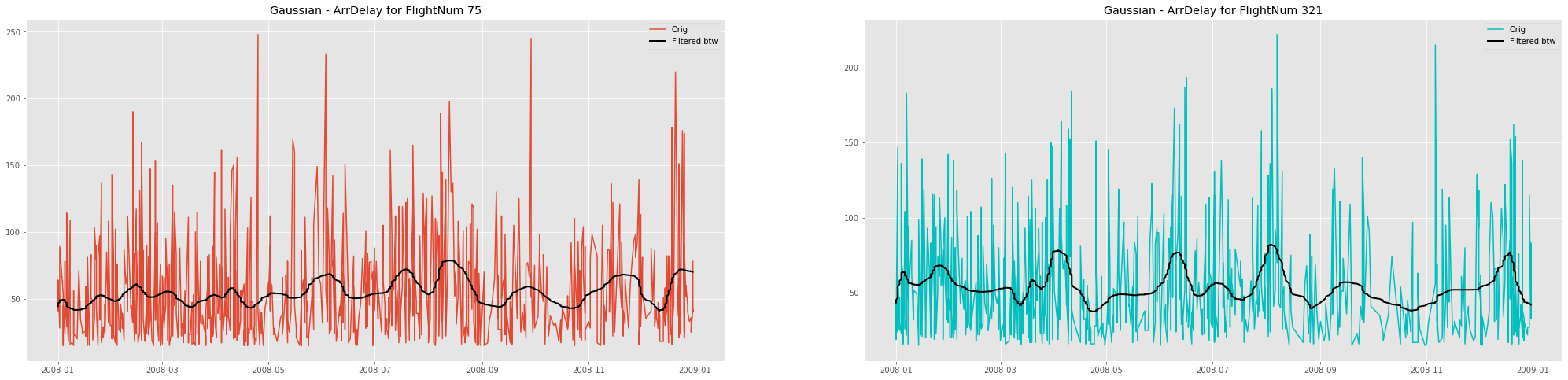


Figure 35 Filtered target variables with Butterworth filter for FlightNum 321



Figure 36 Filtered target variables with Gauss filter for FlightNum 75



Figure 37 Filtered target variables with Gauss filter for FlightNum 321

# Estimation of spectral density function

For estimation of spectral density, we use Blackman window from scipy.signal. The Blackman window is a taper formed by using the first three terms of a summation of cosines. It was designed to have close to the minimal leakage possible. It is close to optimal, only slightly worse than a Kaiser window.

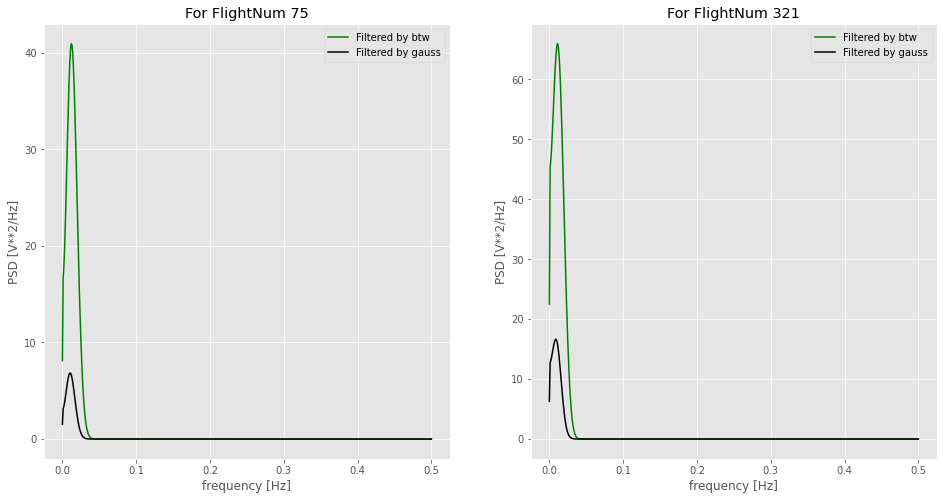


Figure 38 Estimation of spectral density function

# Auto-regression model

Define intervals of the parameters for a auto-correlation model

* ps = range(0,6)
* d=1
* qs = range(0,6)
* Ps = range(0,2)
* D=1
* Qs= range(0,1)

Using Akaike information criterion and varying parameters of the model in given intervals we select the best model in sense AIC, i.e. with the lowest value of AIC.

We determine parameters for three models: on initial data, filtered by Gaussian and filtered by Butterworth filter ones for both flights.

|  |  |  |
| --- | --- | --- |
| Series | Model | AIC |
| ArrDelay | SARIMAX(5, 1, 5)x(1, 1, 0, 12) | 8533.255 |
| ArrDelay\_75\_btw | SARIMAX(5, 1, 5)x(1, 1, 0, 12) | 5681.599 |
| ArrDelay\_75\_gauss | SARIMAX(5, 1, 5)x(1, 1, 0, 12) | 7430.193 |
| ArrDelay\_321\_btw | SARIMAX(5, 1, 5)x(1, 1, 0, 12) | 5357.291 |
| ArrDelay\_321\_gauss | SARIMAX(5, 1, 5)x(1, 1, 0, 12) | 7288.278 |

The coefficient is defined as , where u is the residual sum of squares and v is the total sum of squares . The best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a score of 0.0.

In our case we have a score

As metrics of the regression quality, we chosen RMSE, MAPE

For our models and data we got:

|  |  |  |  |
| --- | --- | --- | --- |
| FlightNum | Filter | RMSE | R2 |
| 75 | None | 44.198 | -1.874 |
| Butterworth filter | 1.81 | 0.959 |
| Gauss filter | 0.89 | 0.983 |
| 321 | None | 43.09 | -1.73 |
| Butterworth filter | 1.814 | 0.972 |
| Gauss filter | 1.241 | 0.971 |

Apart from that we can plot predicted target variable and in full agreement with metrics we see good result.

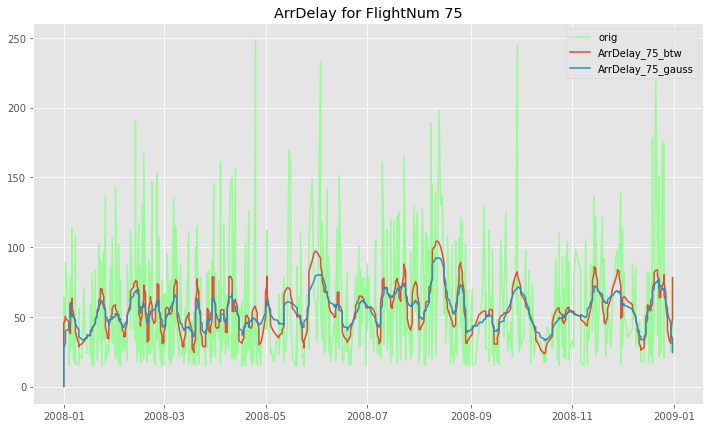


Figure 39 Simulate target variable with SARIMA for FlightNum 75

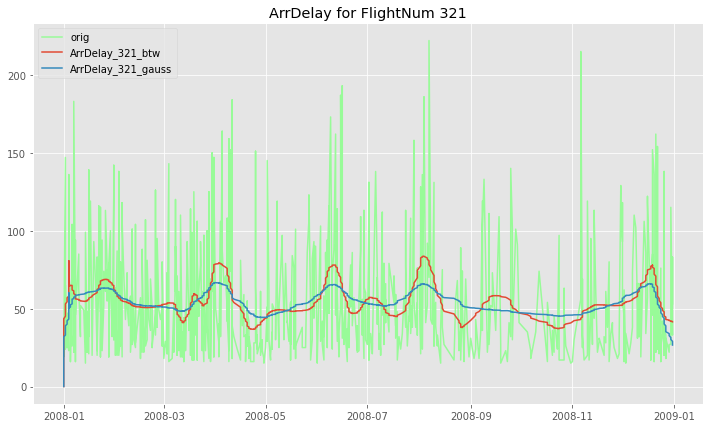


Figure 40 Simulate target variable with SARIMA for FlightNum 321

# Model in a form of linear dynamical system

As linear dynamical system we FEDOT. Install it and train model on our data for FlightNum 75 and 321.

Calculate metrics of a forecast quality:

|  |  |  |  |
| --- | --- | --- | --- |
| FlightNum | RMSE | MAE | MAPE |
| 75 | 34.449 | 30.609 | 0.944 |
| 321 | 13.886 | 10.837 | 0.274 |

As we can see metrics are not pretty well. From plots we see the same things.

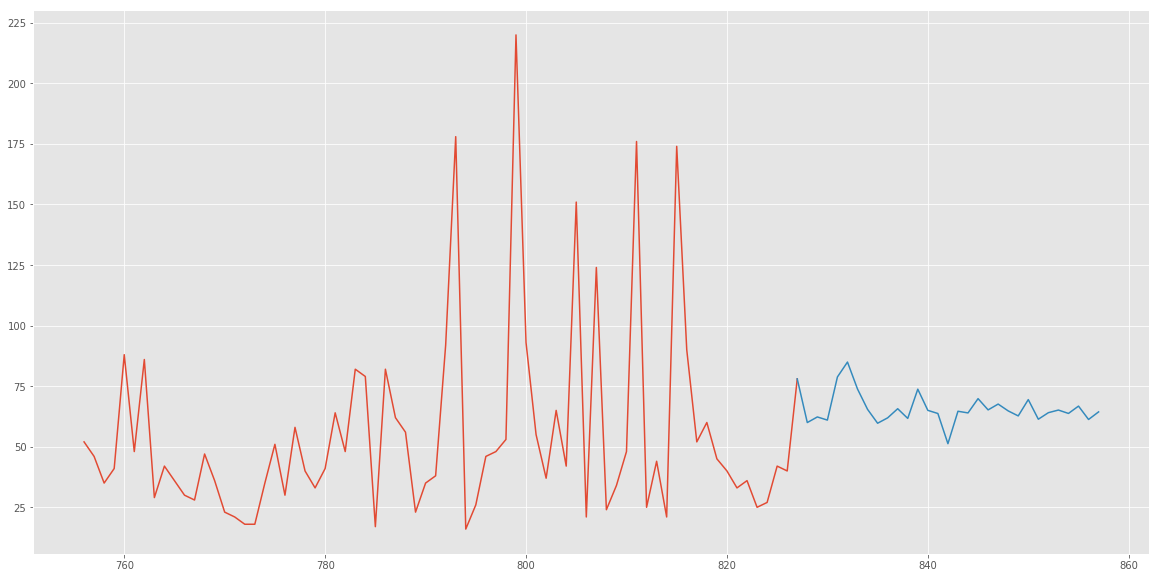


Figure 41 Forecast with FEDOT for FlightNum 75

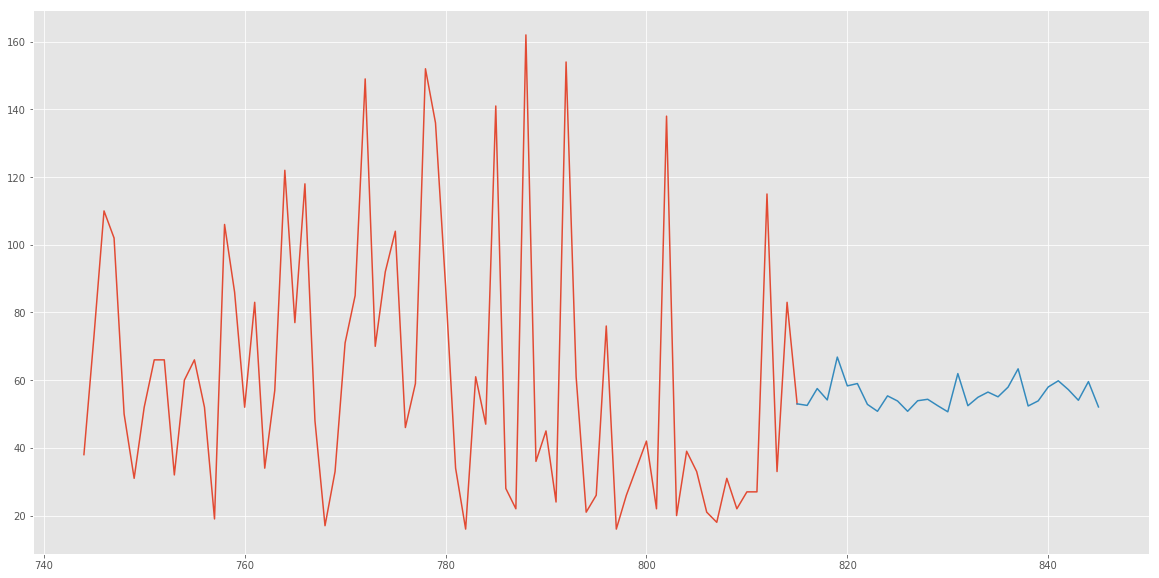


Figure 42 Forecast with FEDOT for FlightNum 321

## Residual analysis

We can see that residuals have near-zero mean and it seems that they are normal distributed.

|  |  |  |  |
| --- | --- | --- | --- |
| FlightNum | Filter | Mean | Variance |
| 75 | Butterworth | -0.241 | 1323.763 |
| Gauss | 1.0 | 1365.556 |
| 321 | Butterorth | 0.202 | 1227.552 |
| Gauss | 1.113 | 1282.334 |

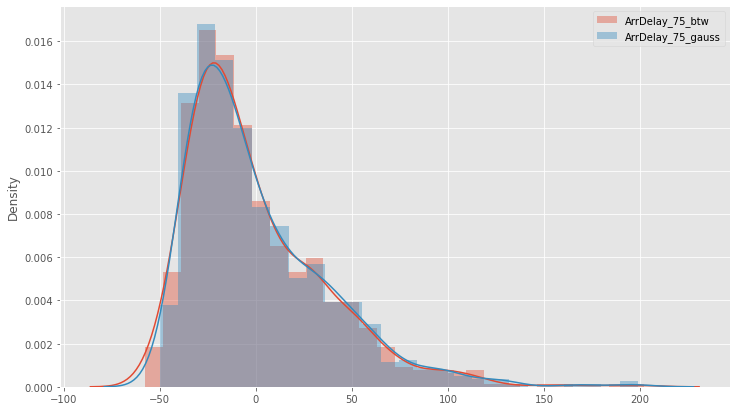


Figure 43 Residuals for ARIMA FlightNum 75 model

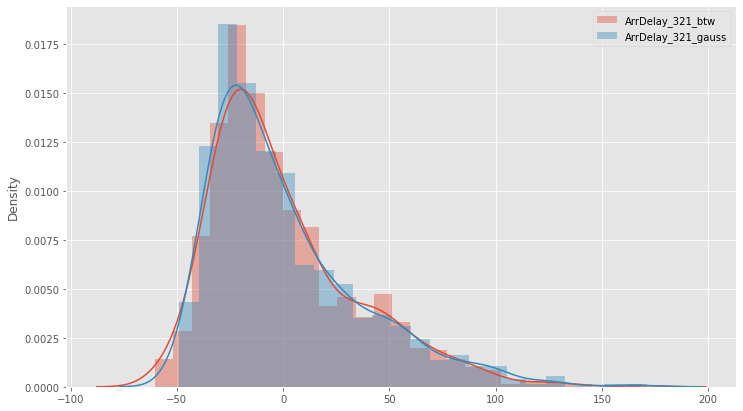


Figure 44 Residuals for ARIMA FlightNum 321 model

# Conclusion

For most cases MLE approach works well to fit the distribution on the observing data.

We have seen that the greatest contribution to the arrival delay is made by LateArrivaldelay. This delay occurs due to the late arrival of the same aircraft at a previous airport. The ripple effect of an earlier delay at downstream airports is referred to as delay propagation. We can prevent this effect by set a bigger buffer between two consequential flights.

At the same time, it is obvious that LateArrivalDelay is tip of the iceberg, because if at an origin airport there was carrierDelay due to maintenance at a destination airport we would have delay, but the root of the problem in non-optimized maintenance procedure of the carrier. LateArrivalDelay is some kind generalization of all flight problems.

If we consider the rest type of delays and those of them that we can affect, then we have understood that carrier delay is a stumbling block.

Carrier delay is within the control of the air carrier. Examples of occurrences that may determine carrier delay are: aircraft cleaning, aircraft damage, awaiting the arrival of connecting passengers or crew, baggage, bird strike, cargo loading, catering, computer, outage-carrier equipment, crew legality (pilot or attendant rest), damage by hazardous goods, engineering inspection, fueling, handling disabled passengers, late crew, lavatory servicing, maintenance, oversales, potable water servicing, removal of unruly passenger, slow boarding or seating, stowing carry-on baggage, weight and balance delays.

Most of the causes of a carrier delay are concern to maintenance of airplane as before the flight (line maintenance\ground handling) so periodical base maintenance.

Thus, it can be concluded that investments aimed at improving the quality of maintenance can help reduce the magnitude of delays.

# Source code

<https://github.com/vvoronin96/MM_MDA>