Yuyi Zhang

May 16, 2019

Spring 2019 CPS Quarter Term A

Instructor: Steward Huang

Assignment 6: Collaborative Project

ALY 6015\_Intermediate Analytics

# **Introduction**

In this report, we are going to go through an air traffic passenger counts dataset and apply regression and time series analysis to them. The dataset is on monthly basis from 2005 to the current year. We are also going to use a weather report as a dependent dataset in regression analysis.

# **Method**

## **Descriptive and Regression**

In this part, we are going to use hist() to create histogram, density() to create density plots. As well as boxplots, normal probability plots. Then conduct a regression analysis.

## **Time Series**

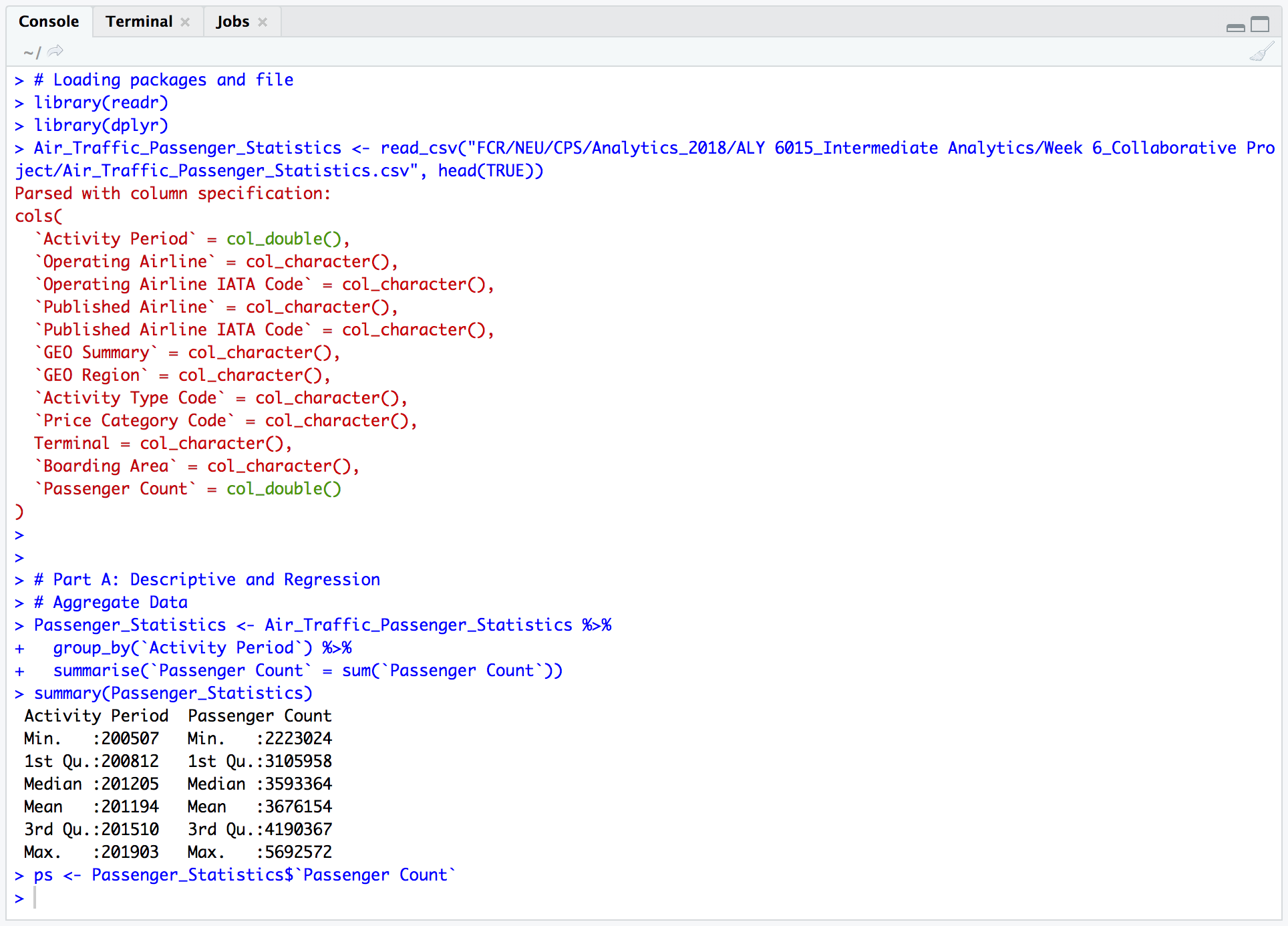
This is for seasonal data analysis. We are going to decomposing the passenger counts dataset and select a candidate for ARIMA model.

# **Analysis**

## **Part A: Descriptive and Regression**

### **Data Preparation**

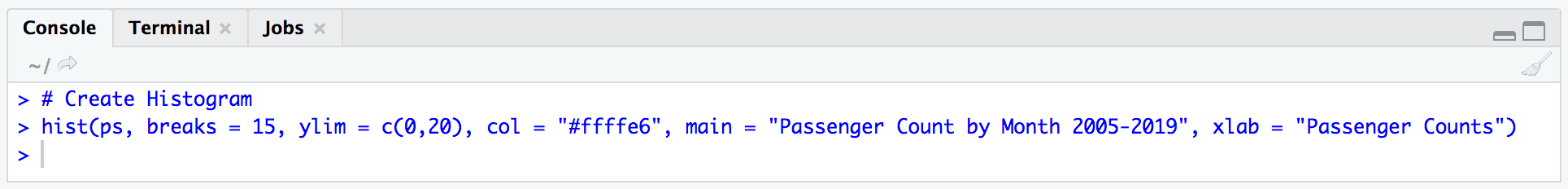
In this phase, we are going load dataset “Air Traffic Passenger Statistics” and aggregate the passenger count column by monthly activity. After aggregation, filter passenger count to a new value called “ps” for further use.



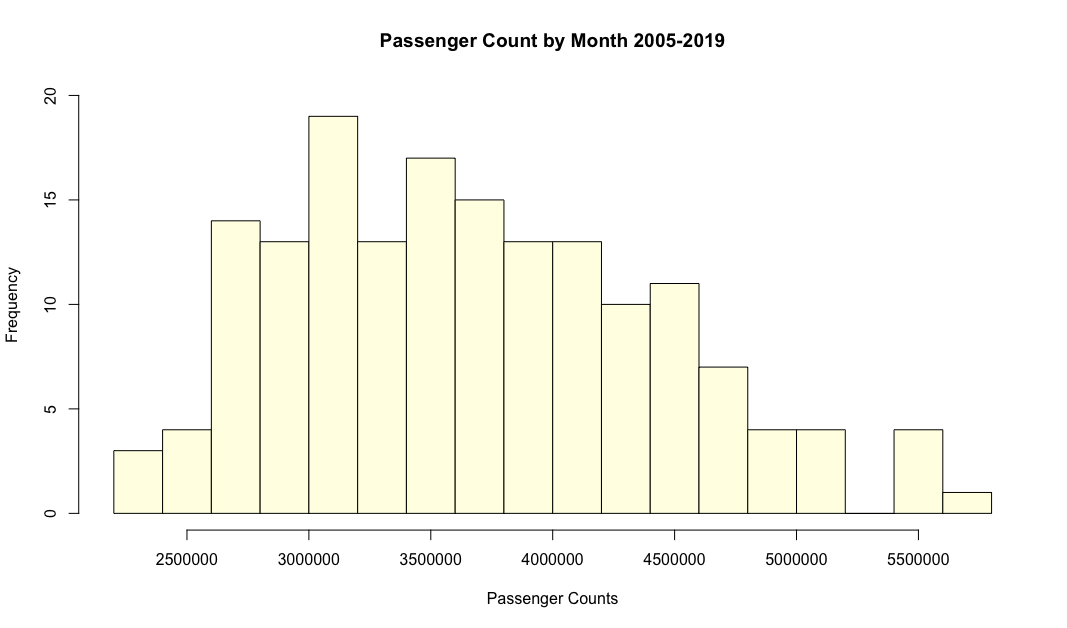
*Figure 1*. Data preparation

### **Create Histogram**

Use hist() to create a histogram. Made few adjustments for clear chart.



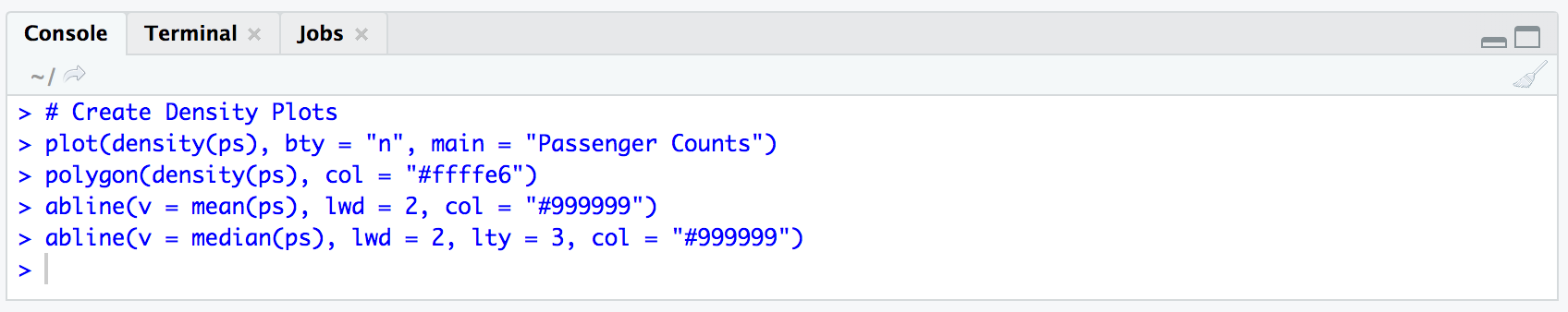
*Figure 2*. Create Histogram

*Figure 3*. Histogram

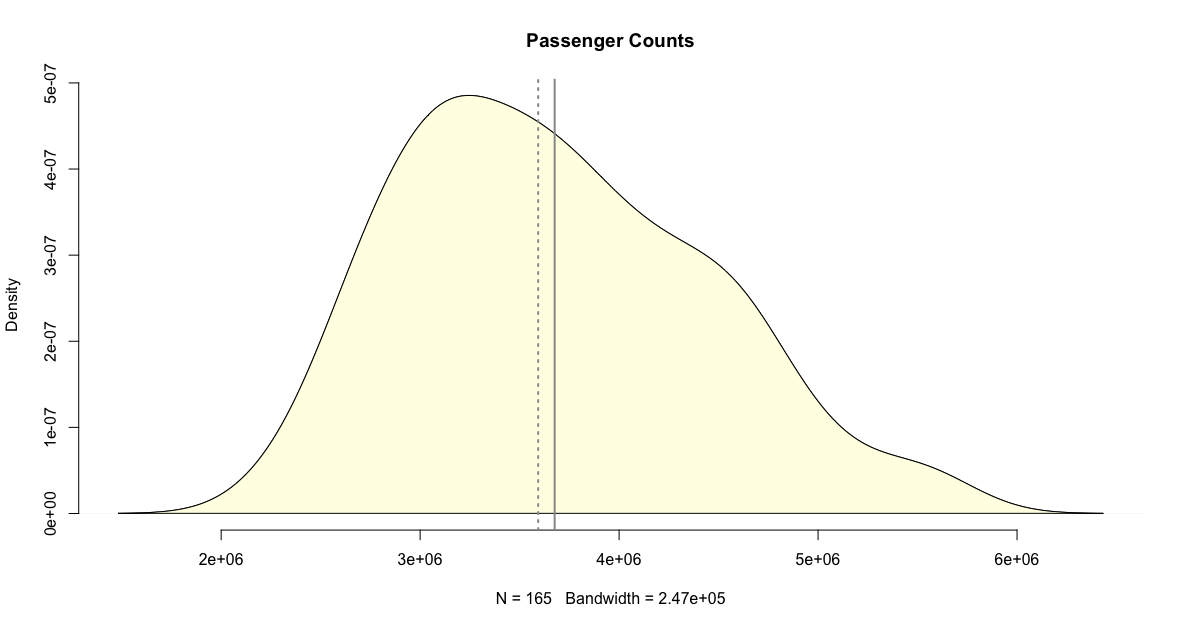
This distribution is unimodal with only one main cluster. There are 20 samples on horizontal axis, while middle 10 bars are higher than the rest. The highest relative frequency is 19. This chart is skewed to the right with only 2 outliers in it. In another word, the normal passenger load would be 3 million to 4.25 million per month. Occasionally, the passenger counts would be less than 2.5 million are higher than 5.25 million.

### **Create Density Plots**

Use density() to create Kernel Density Plots.



*Figure 4*. Create Density Plots

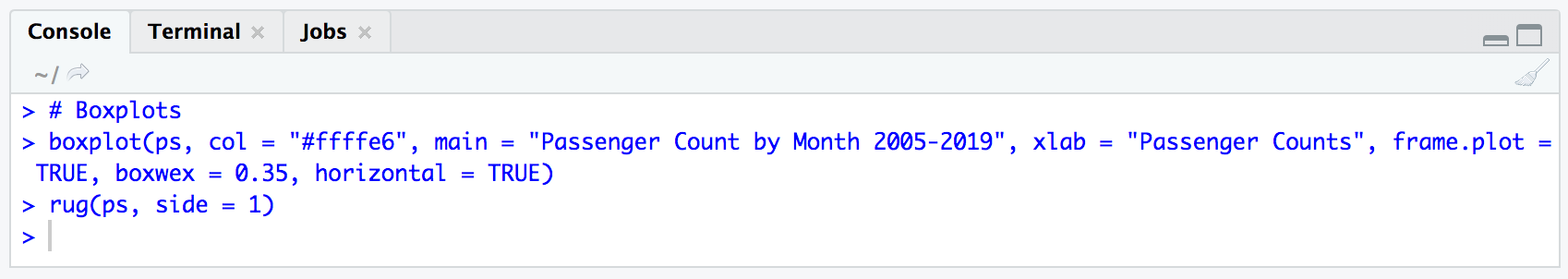


*Figure 5*. Density Plots

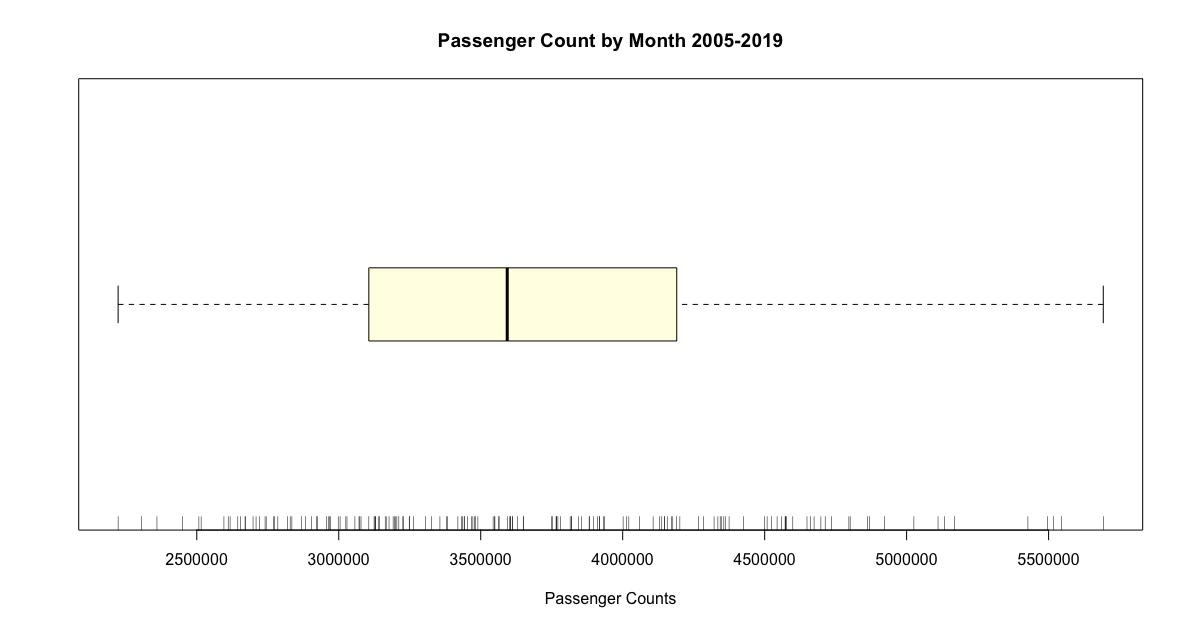
As we can see that density plots are much smoother than histogram. In these plots we also added mean bar (gray bar) and medium bar (dash dots) to get a better idea of the basic distribution. Because the mean bar is to the right of the median, so this distribution has a longer right-hand tail.

### **Create Box Plot**

Use boxplot() to create a box plot in R. Then use rug() to make enhancement.



*Figure 6*. Code for Box Plots

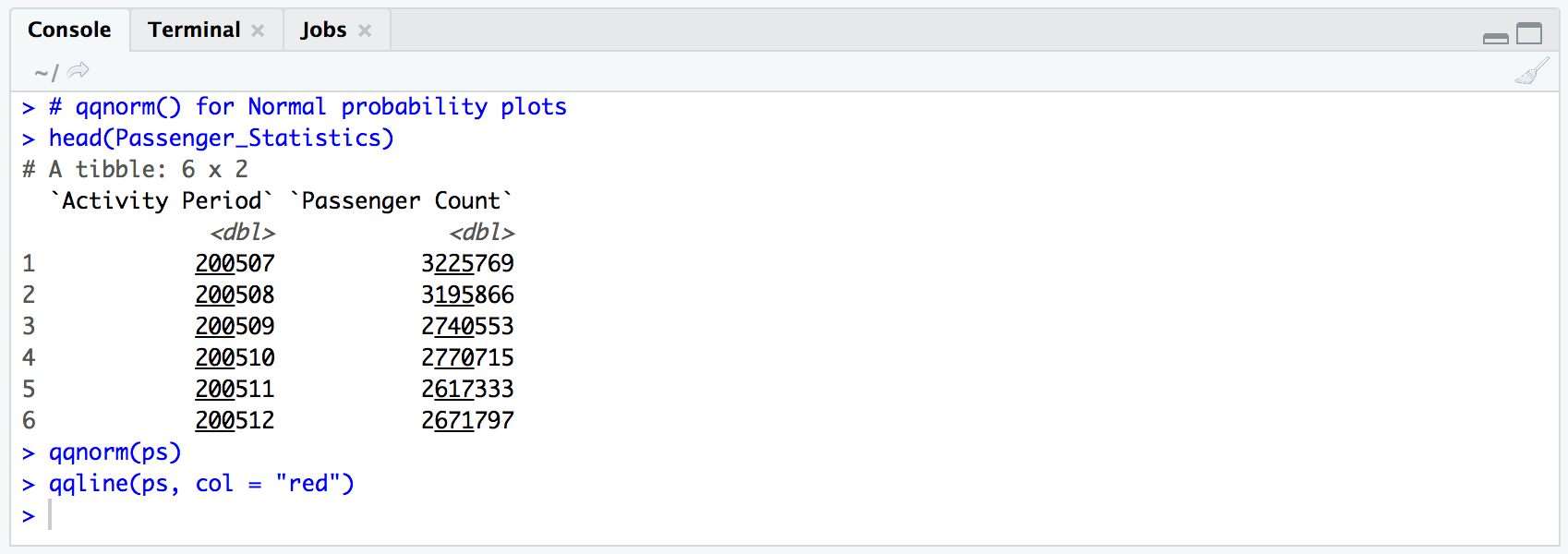


*Figure 7*. Box Plots

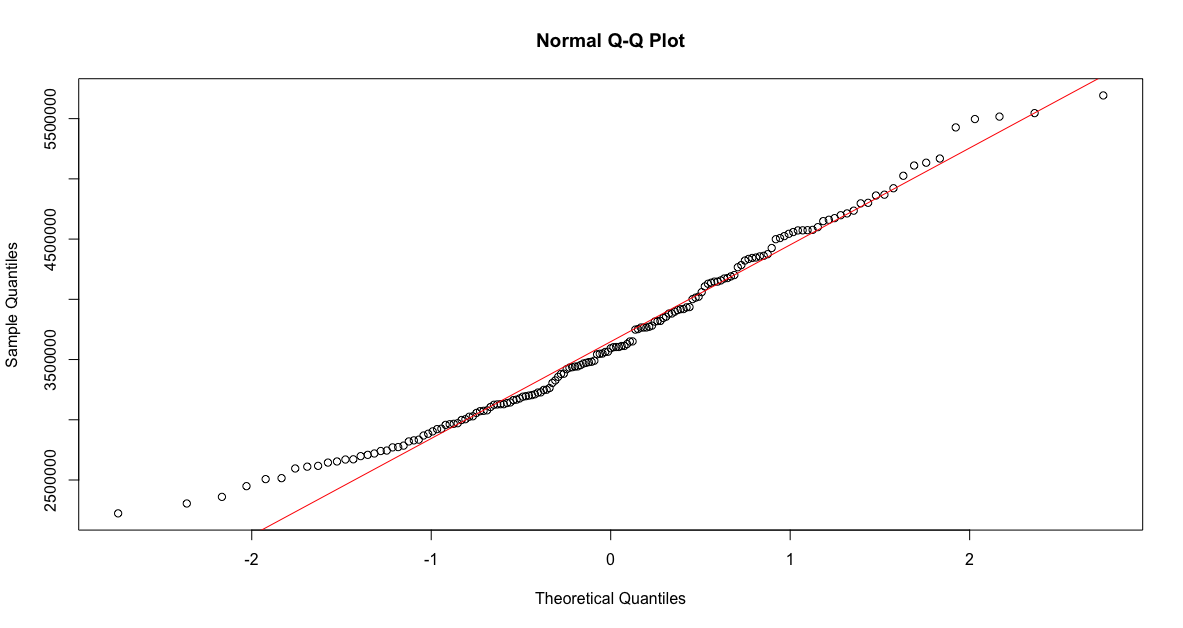
In the box plot we could get a better idea of where the median is. In the box we could see that interquartile range is about 3,200,000 to 4,250,000. As the idea that we’ve got from the first plots, the histogram plots that, this is a skewed to the right. Better than that, we could see a roughly data allocation throughout the rug in below. But if we want to see the frequency of each range, histogram would be the best chart that we are looking for.

### **Create Normal Probability Plots**

Use qqnorm() to create normal probability plots. Quantile-Quantile plots are used to determine if data can be approximated by a statistical distribution. In this case, we want to know if the ps(passenger counts of passenger statistics) was normally distributed.



*Figure 8*. Code for Normal Probability Plots (QQ plot)

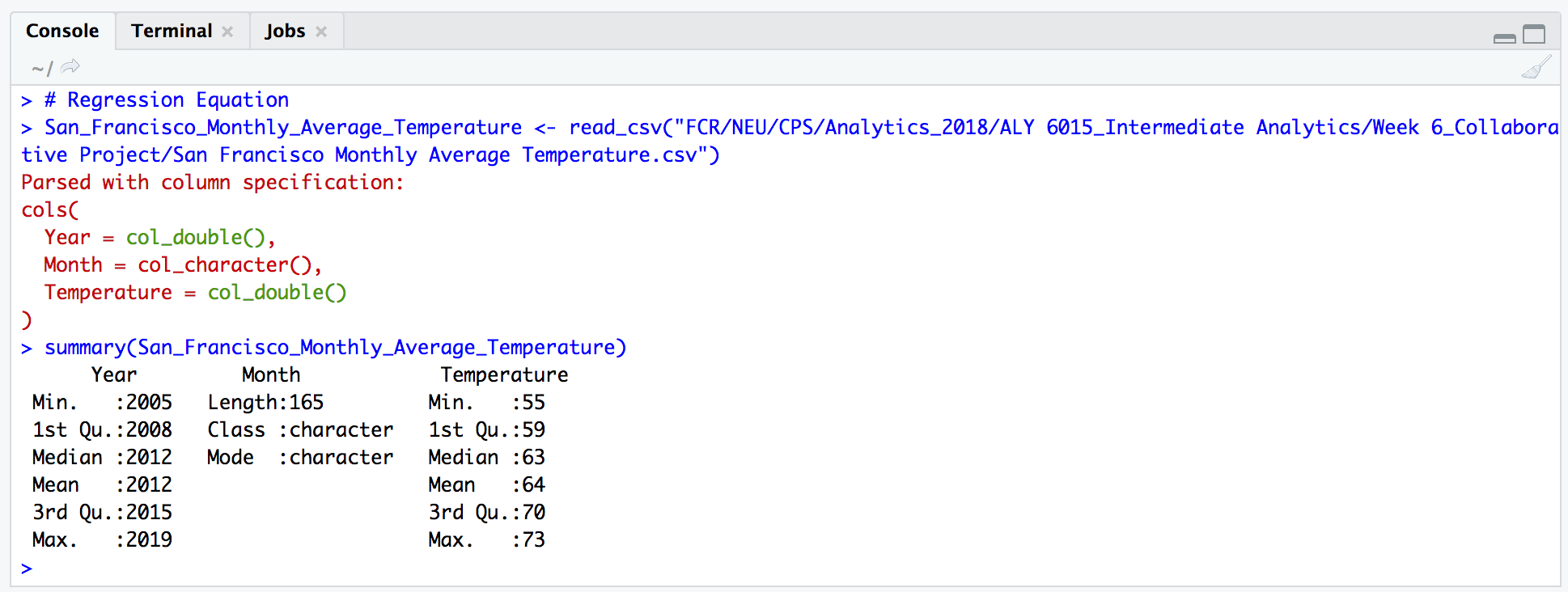


*Figure 9*. Normal Probability Plots

As we can see from the qq plots, the ps quantiles vs. standard normal distributed quantiles are not perfectly fit in a straight line. Therefore, ps is not very much normally distributed. But in from the 1 standard distribution part, it looks fit better than the ones besides.

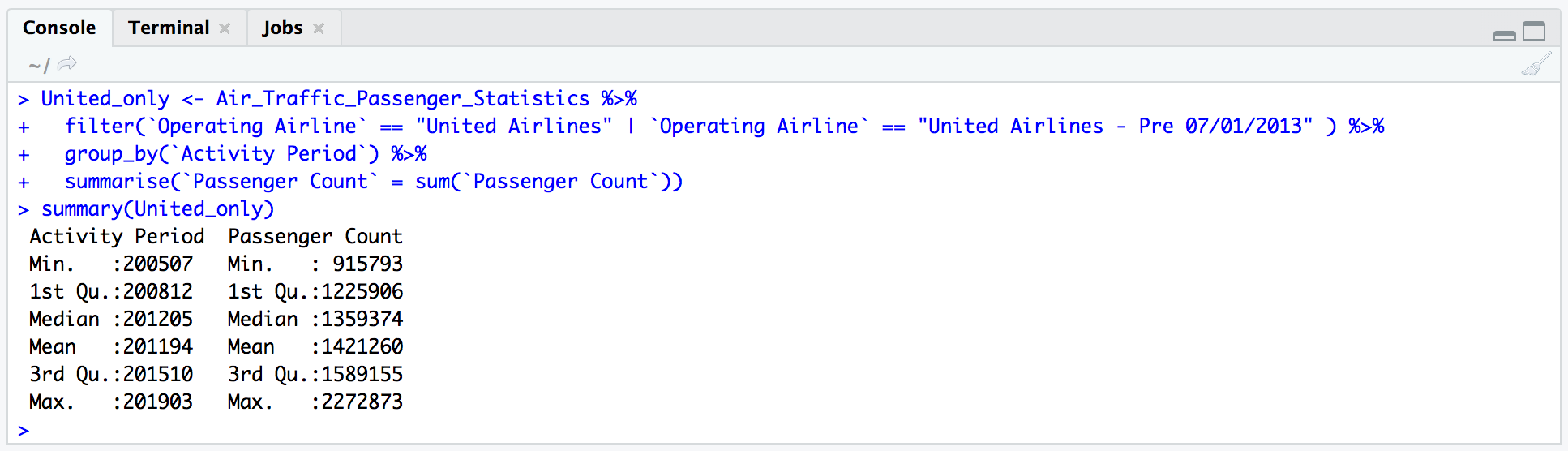
### **Linear Regression**

In this part, we are going to generate a linear regression test. First step, we are going to pull out the dependent data that we are going to use for the test. The first dataset would be monthly average temperature in San Francesco. Read the csv file and use the summary() to get a general idea of the data shown as below.



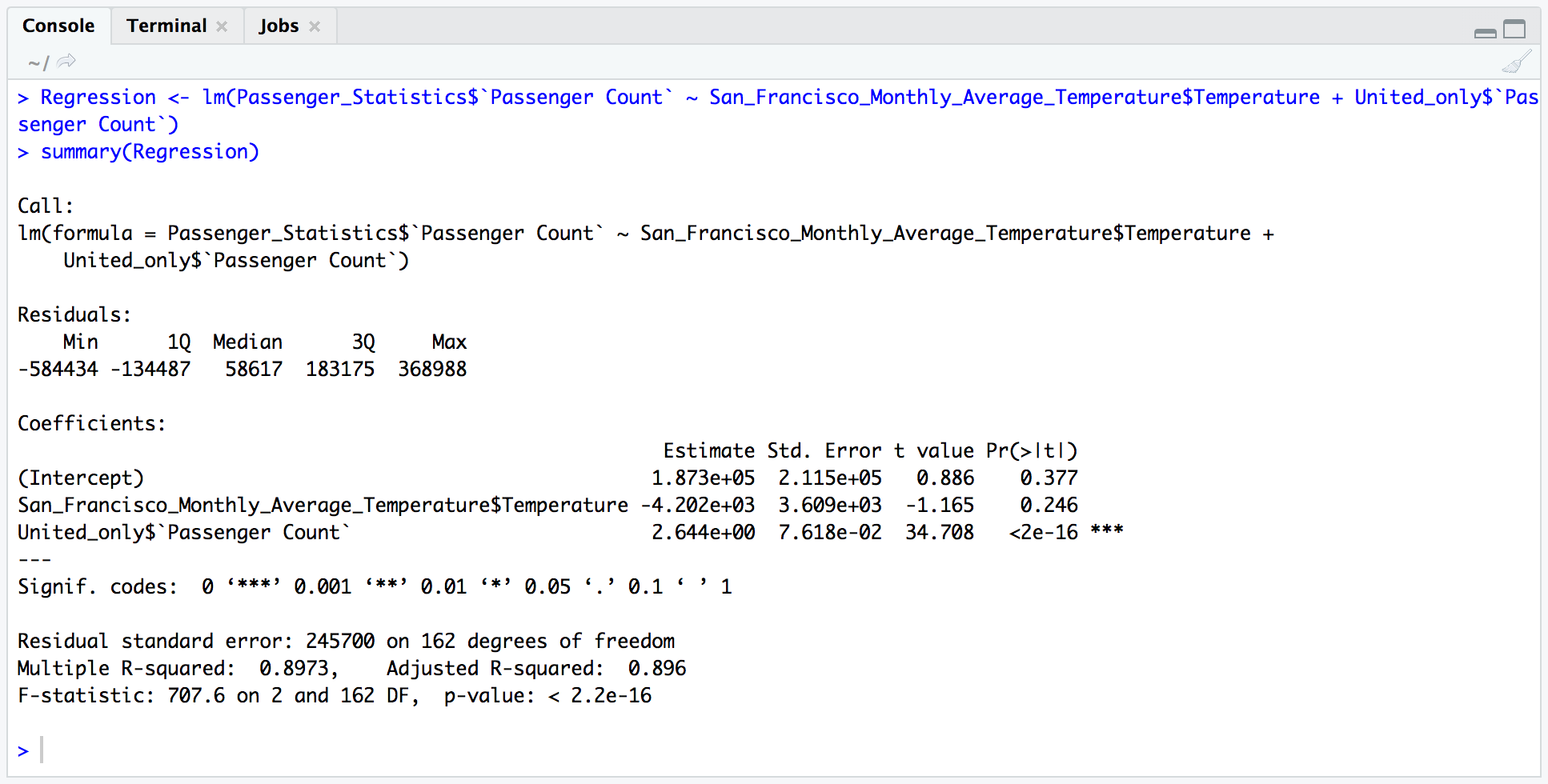
*Figure 10*. Loading file

The second dataset that we are going to use is actually in the original file that we’ve been using the whole time: the united passenger counts for each month. So we used filter() to aggregate what we want and use summary() to get a general idea. Code is shown as below.



*Figure 11*. Data Preparation

After all data have been prepared, we use lm() to generate linear regression. We set the total passenger counts as independent variable, the monthly average temperature in San Francisco and United Airline passenger counts as dependent variables. Then we use summary() to get the summary report of the regression shown as below.



*Figure 12*. Linear Regression

In this report, the residual section shows all the residual quantile information (the distance from the data to the fitted line). Ideally, they should be symmetrically distributed around the line. That is to say, ideally the minimum value and the maximum value would be the same distance from zero, as well as 1Q and 3Q. But in our data, the absolute value of min and max are quite different.

In the coefficients section, we got the least-squares estimates for the fitted line. In this case, we’ve got our equation as:

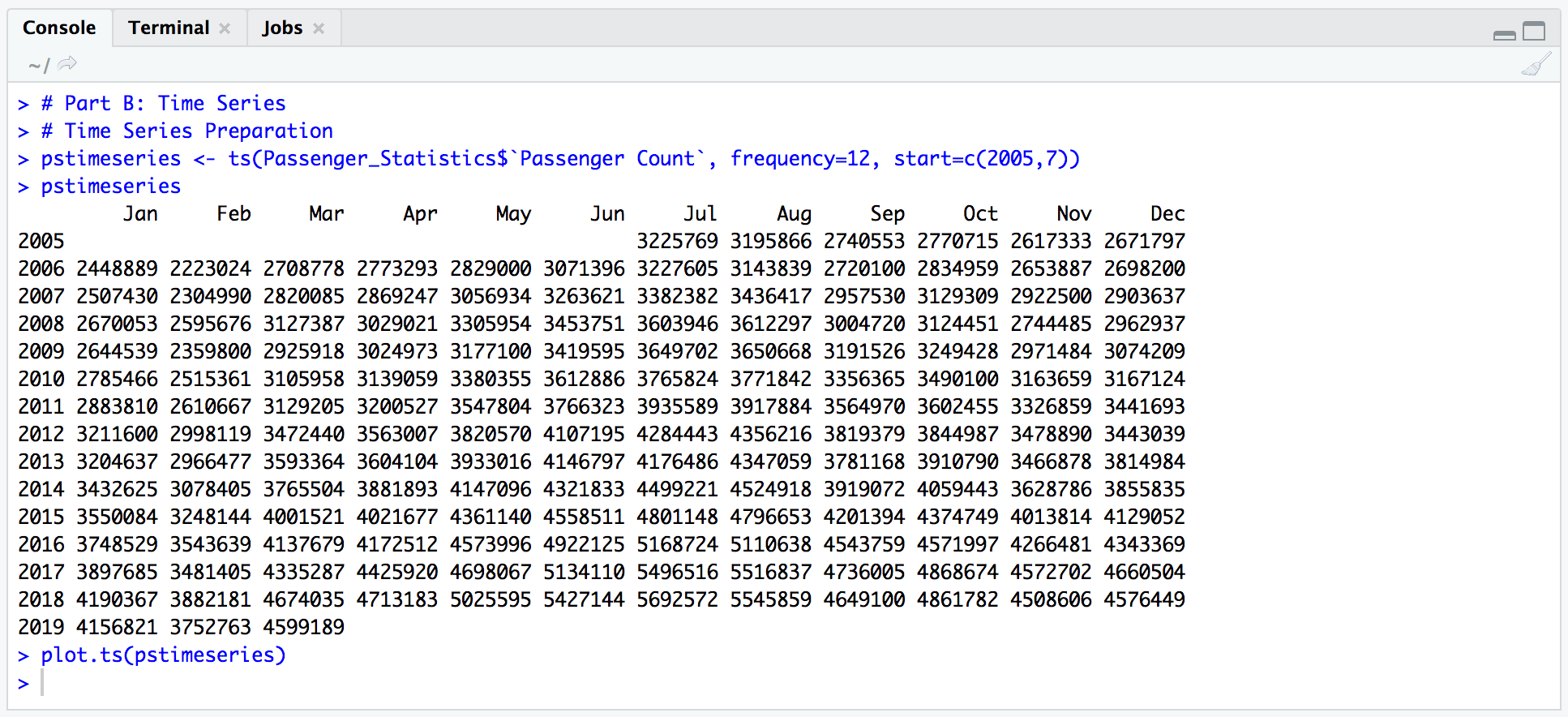
The std. error and t value are provided to show how the p-value were calculated. If the value is equal to zero, it means that the variable doesn’t have much use in the model. But in this case, we’ve got a significant t-value of United Airline passenger counts, 34.708. Moreover, the p-value is way much smaller than 0.05, which is statistically significant.

In the last section we see the r-squared value is 0.8973. This means that United airline passenger counts explain 89.73% of the variation in total passenger counts in San Francisco airport. The p-value of R-squared is 2.2e-16. This means, again, the United Airline passenger counts give out a reliable estimate for total passenger count in San Francisco airport.

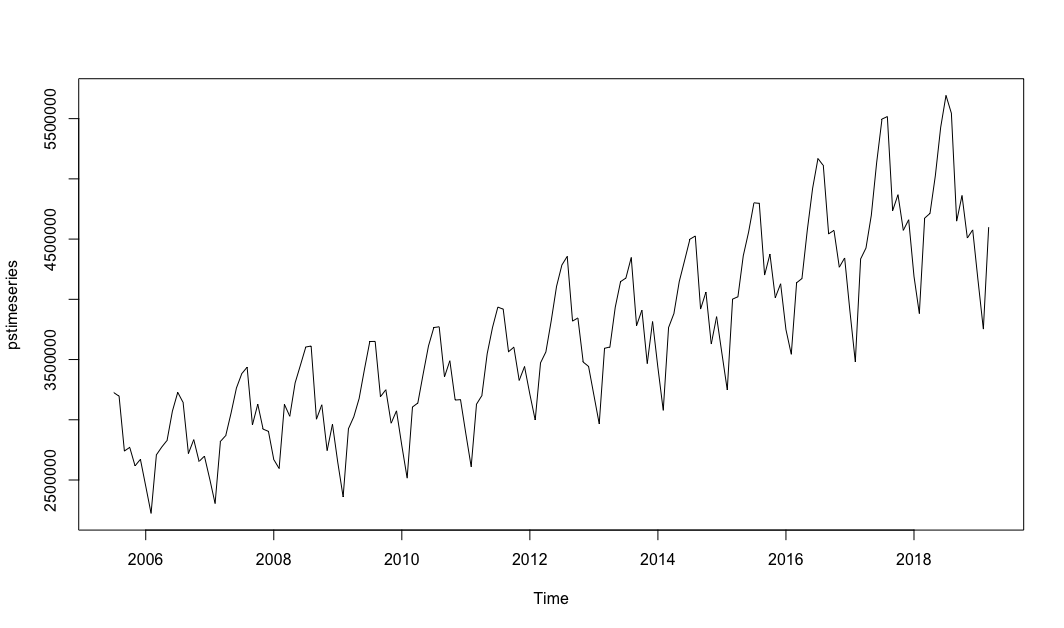
## **Part B: Time Series**

### **Data preparation**

In part B, we are going to conduct time series analysis. First of all, convert ps data to time series format.



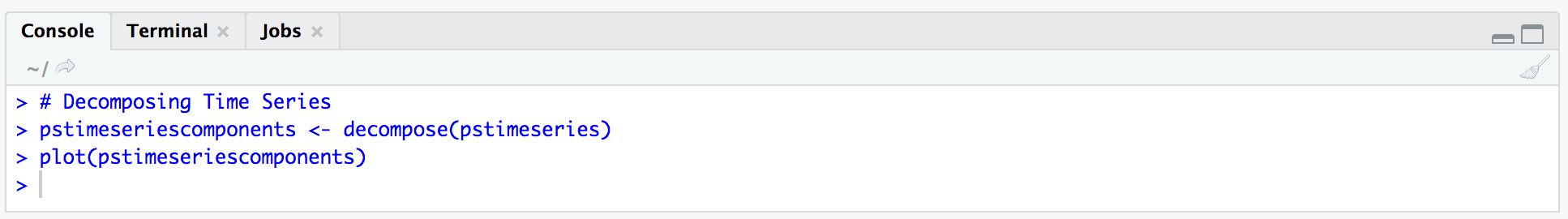
*Figure 13*. Code for Data Preparation



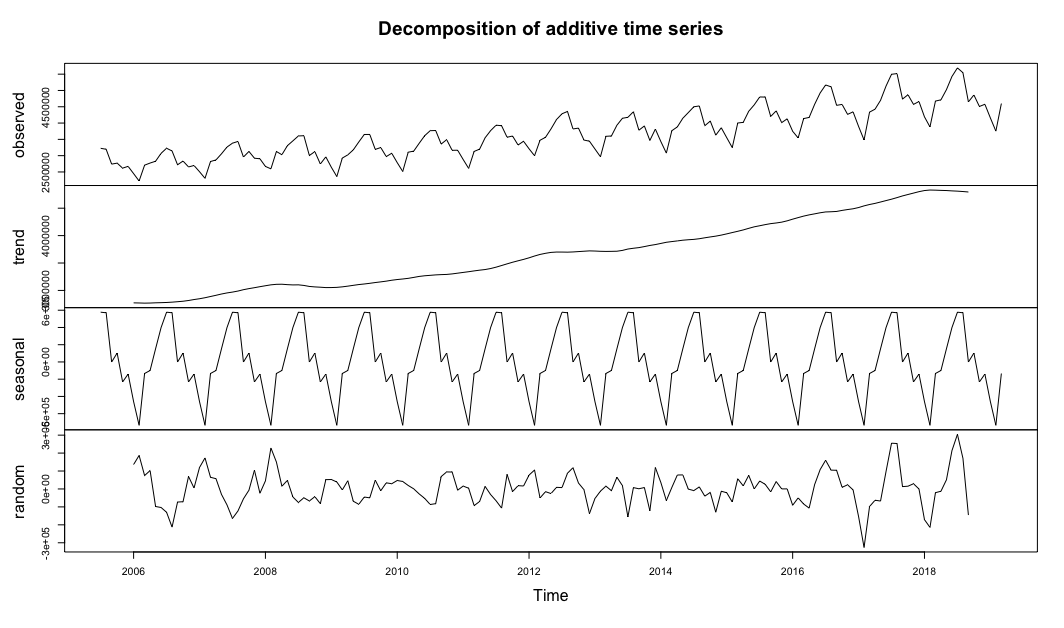
*Figure 14*. Original Data Plot

### **Decomposing**

Use decompose() and then draw plots to get a visual illustration of data decomposition. This dataset is a seasonal time series consists of a trend component, a seasonal component and an irregular component. So in this process, we are going to separate the time series into three components: trend, seasonal and irregular.



*Figure 15*. Code for Decomposing

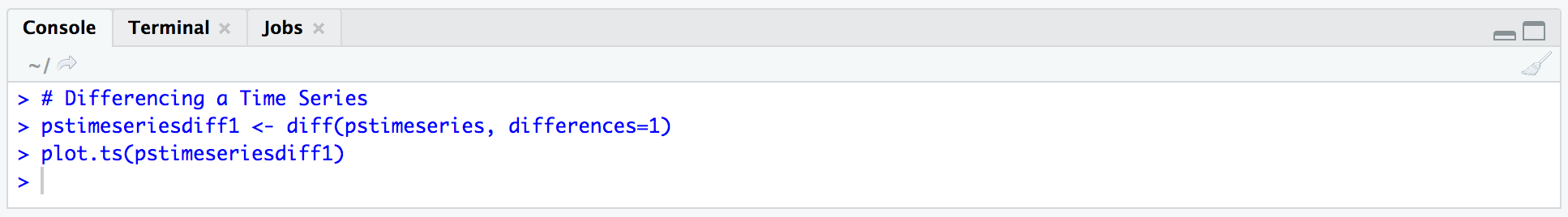


*Figure 16*. Decomposition Plots

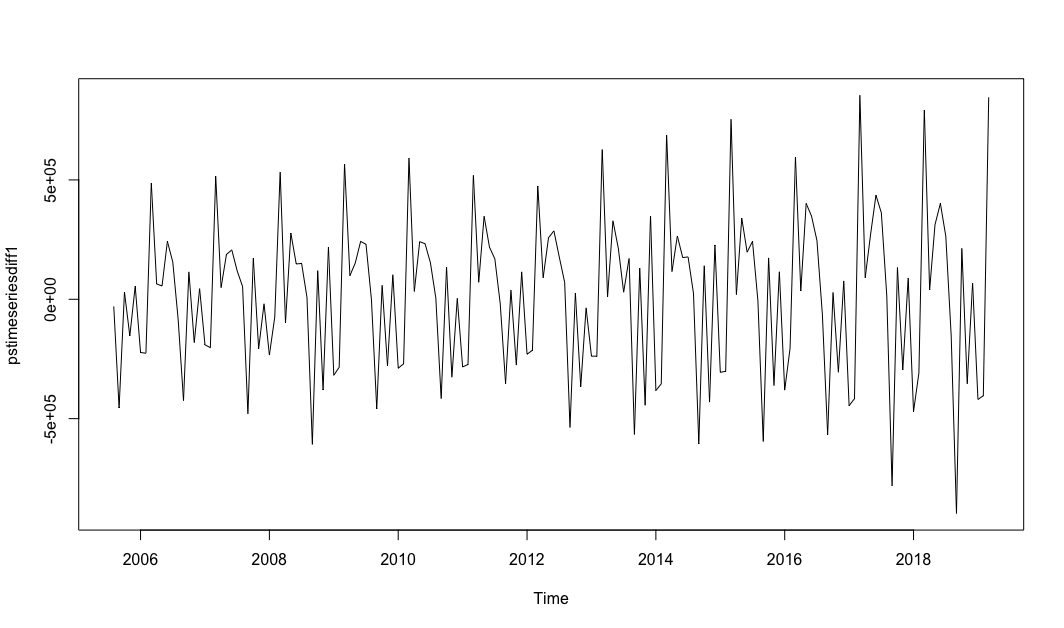
This plot shows the original time series and three components we mentioned before. As expected, the trend shows overall increase. There is a highly consistent seasonal cycle as shown in the third plot from top. Irregular noise shown in the bottom.

### **Differencing**

Containing of trends and seasonality define a time series data as being non-stationary. Stationary datasets are those that have a stable mean and variance and are turn much easier to be modeled. Differencing is a method to transform time series data stationary.



*Figure 17*. Code for differencing

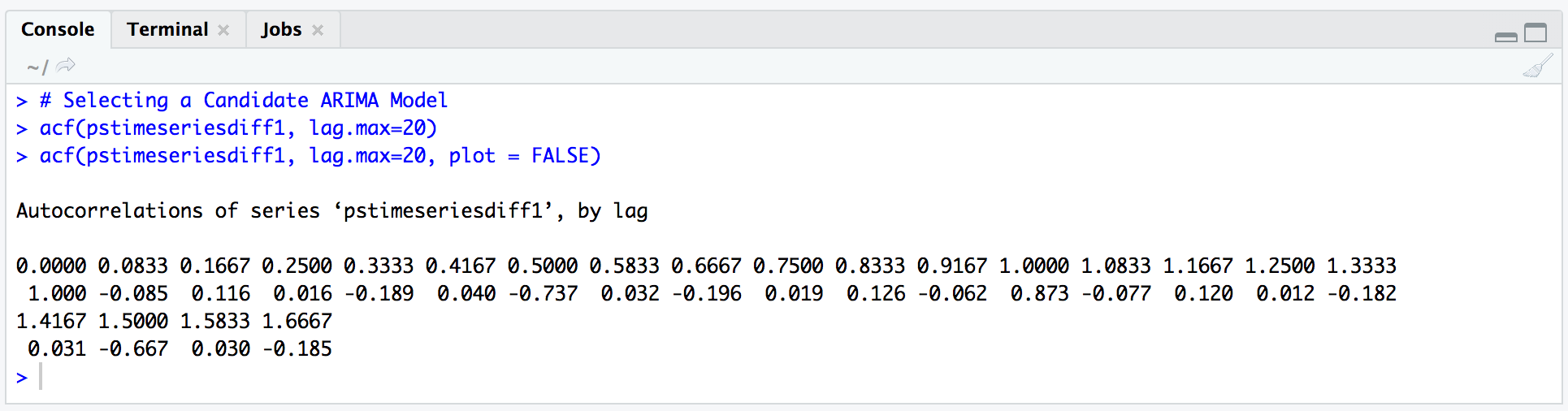


*Figure 18*. Differencing plot

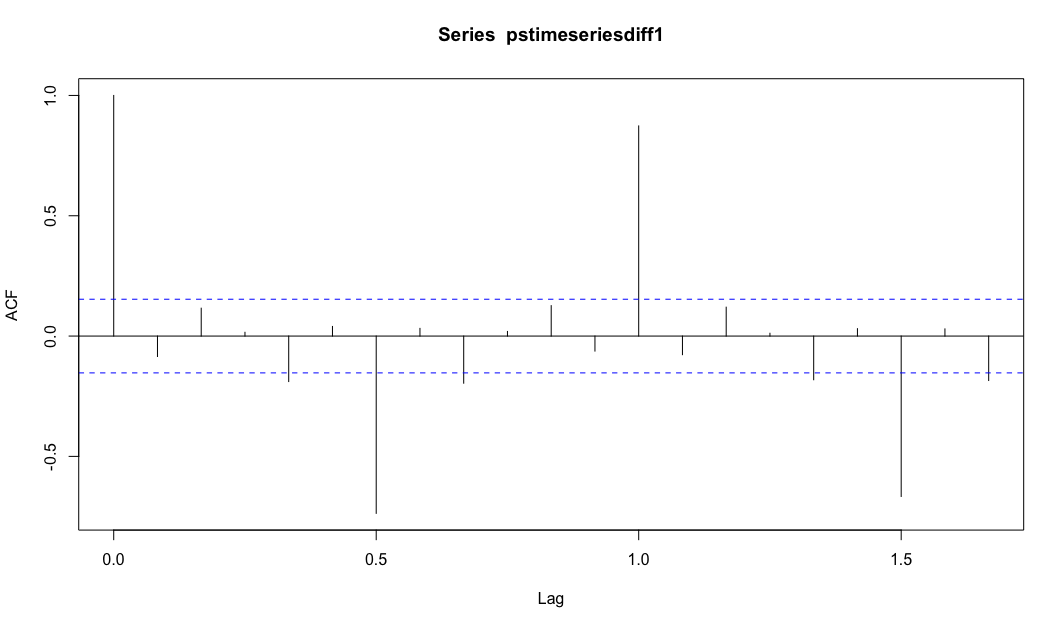
In this case, the difference order would be 1, which means performing a lag-1 differencing operation that would transform to remove a trend.

### **Selecting a Candidate ARIMA Model manually**

To select candidates for ARIMA model, we are going to use two ways: manually and automatically. We need to conduct residual analysis and find the appropriate values of p, d, q representing the AR order, the degrees of differencing and the MA order, respectively. After the previous step, we know that the d value is 1. To manually selective p and q values, we need to do ACF and PACF analysis.

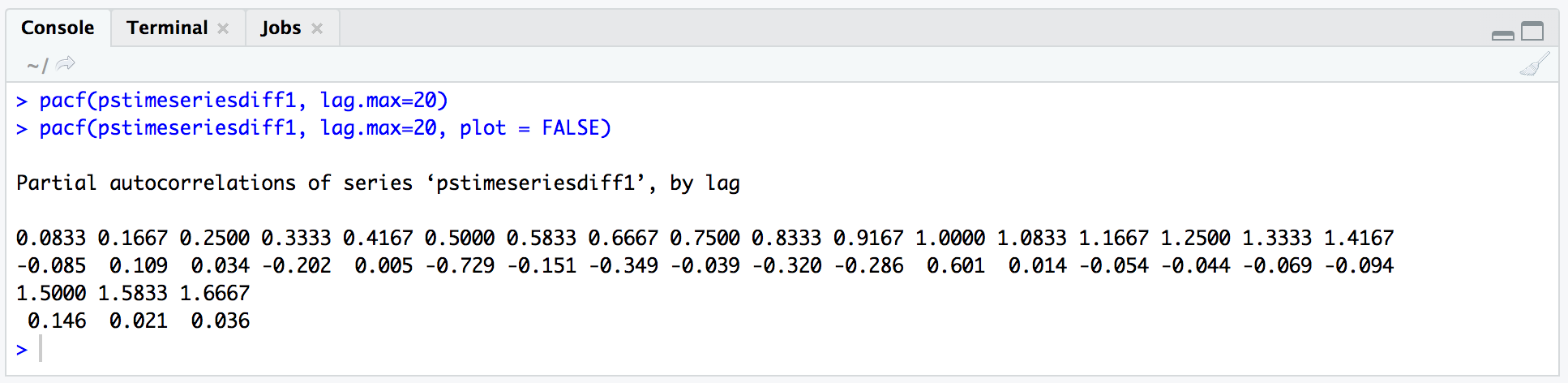


*Figure 19*. Code for ACF

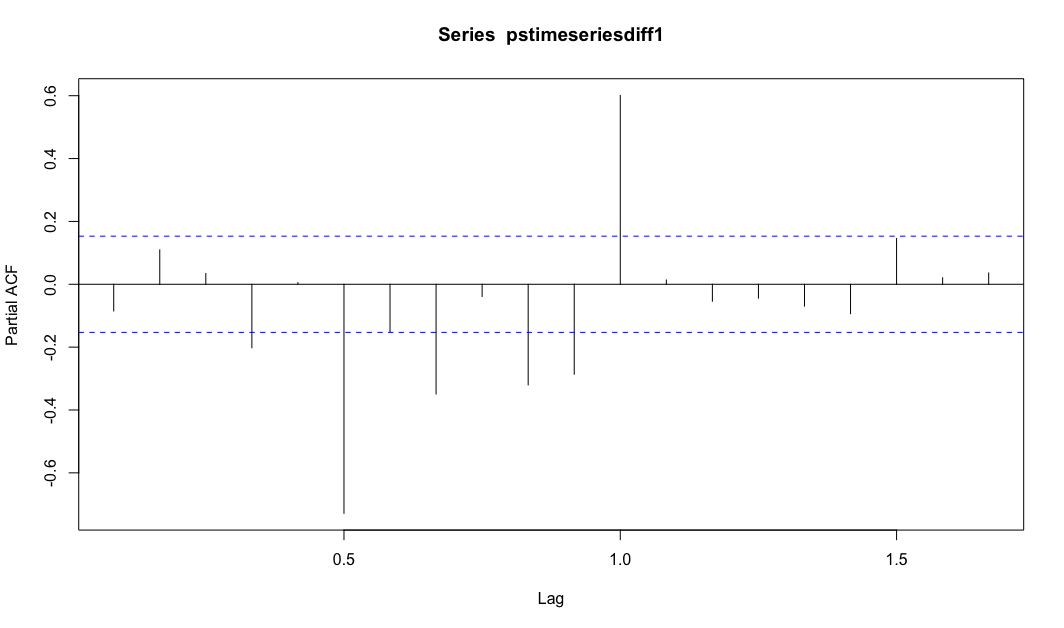


*Figure 20*. ACF plot

The blue dash lines indicate bounds for statistical significance. The scale is from -1 to 1 because it is the correlation coefficient. As we can see from the ACF plot, lag 0 exceed the significance bounds. But lag 1 is a negative value. Therefore, the q value would be 1.



*Figure 21*. Code for PACF

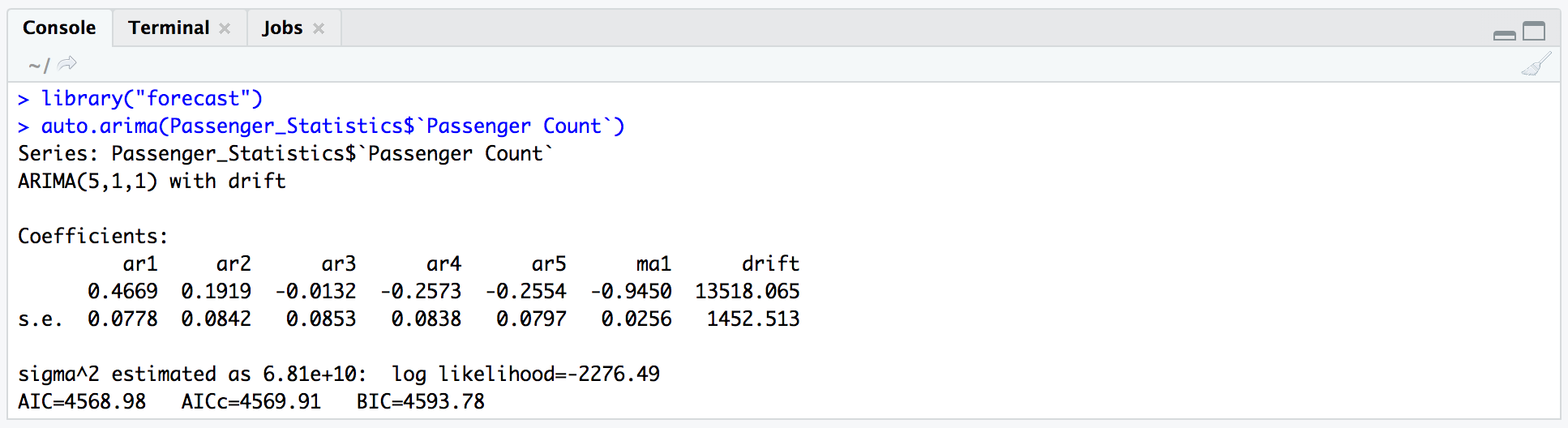


*Figure 22*. PACF plot

In this step, we could see that the partial autocorrelation at lag 5 is way exceed the significance bounds. Therefore the p value of ARIMA model is 5. The way to final decide the most appropriate values for ARIMA model is to consider the fewest parameter is the best. In this case, ARIMA (5,1,1) wins.

### **Selecting a Candidate ARIMA Model automatically**

Use auto.arima() to calculate q, d, p values for the model.



*Figure 22*. Code for auto.arima()

As shown above, we’ve got ARIMA(5,1,1) with drift model. Therefore if we apply Arima() function, we are going to set include.drift = TRUE to alow drift in ARIMA model.

# **Conclusion**

In this report we conduct two analysis: regression analysis and time series analysis. In part A, we’ve learned that the dataset is almost normally distributed and skewed to the right. After the regression we known that the United Airline passenger counts explain 89.73% of the variation in total passenger counts in San Francisco airport.

In the ARIMA analysis, we’ve conducted decomposing and differencing process. Then we selected candidates of the model as (5, 1, 1) both by manually and automatically.

Reference

1. City of San Francisco (May 10, 2019). *Air Traffic Passenger Statistics*. Retrieved from <https://catalog.data.gov/dataset/air-traffic-passenger-statistics>
2. Brownlee, J. (2017). *How to remove Trends and Seasonality with a Difference Transform in Python*. Retrieved from <https://machinelearningmastery.com/remove-trends-seasonality-difference-transform-python/>
3. Elprince, N. (2014). *Air Passengers Forecast*. Retrieved from <https://rpubs.com/nohaelprince/47545>